Latent Cluster Analysis for Vision-Language-Action Models

Theodor Wulff[1,*], Sergio Lanza[2,*], Tamara Bila[3,*],
Angelo Cangelosi[1], Stefan Wermter[2], Igor Farkas[3]
[1]The University of Manchester, [2]Universität Hamburg, [3]Comenius University Bratislava
Correspondence: theodor.wulff@manchester.ac.uk, sergio.lanza@uni-hamburg.de, tamara.bila@fmph.uniba.sk
*equal contribution

Abstract

Vision-Language-Action (VLA) Models are increasingly used in robotics for their ability to ground language and perception into action, yet the internal representations driving their behaviour remain poorly understood. We propose LAVLA, a framework for latent cluster analysis of VLA models, and conduct a layer-wise study of the state-of-the-art GR00T N1.5 model, with particular focus on its action decoder. To better characterise the latent space during action diffusion, we introduce a cross-attention-based embedding-weighting method that amplifies relevant features while suppressing less informative ones. Quantitative evaluation shows that weighted clustering consistently outperforms the baseline. To improve interpretability, we extract human-interpretable concepts for each cluster, linking latent representations to semantic descriptions. Our analysis shows that latent clusters progressively disentangle spatiotemporal and kinematic features, with representations becoming more refined in the middle layers and stabilising toward the output. As such, LAVLA advances the interpretability of language-driven robotic systems.

1 Introduction

Vision-Language-Action Models (VLAs) are gaining popularity in robotic research due to their promising generalisation capabilities. Conventionally, VLAs are trained on large-scale collections of robot data across several environments, tasks, and embodiments, enabling strong performance in different scenarios and architectures (Kim et al., 2025; NVIDIA et al., 2025). While performance continues to improve as the number of environments and considered tasks increases, only a limited amount of research is concerned with explaining the inner reasoning of these models (Häon et al., 2025). Extending the understanding of these models is a key factor in robotics, where the range of possible out-of-distribution scenarios is vast, and unverified behaviour can have dire consequences. Gaining insight into how models encode multimodal information improves user trust and facilitates the identification of reasoning failures (Ma et al., 2025; Zang et al., 2025). Based on these premises, we introduce a novel framework for **L**atent Cluster **A**nalysis for **V**ision-**L**anguage-**A**ction Models, abbreviated **LAVLA**. We investigate the latent space of a Vision-Language-Action Model, focusing on the action decoder and clustering embeddings in these layers. Specifically, we collect model embeddings during forward passes from the two main modules of the VLA: the final layer of the Vision-Language Model (VLM) backbone and all intermediate cross-attention layers of the Diffusion Transformer (DiT) (Peebles and Xie, 2023) action decoder. Furthermore, we introduce an embedding-weighting module to amplify the features of important visuolinguistic tokens from the corresponding VLM backbone embeddings used in the cross-attention layers of the action decoder. We demonstrate the effectiveness of the LAVLA framework by applying it to NVIDIA's GR00T N1.5 model (NVIDIA et al., 2025) using a subset of its original training data (O'Neill et al., 2024). Finally, we extract a human-interpretable concept from the resulting clusters based on the top-n closest samples to each centroid, assisted by an external VLM. Our results underscore the pivotal role of the embedding-weighting module in guiding the action denoising process.

Our contributions can be summarised as follows:

- We propose LAVLA, a novel cluster-based framework for analysing the embeddings in VLAs and demonstrate its effectiveness by analysing the latent space of the NVIDIA GR00T N1.5 model.
- We introduce a cross-attention-based weighting module designed to amplify the influence of tokens prioritised by the action decoder.

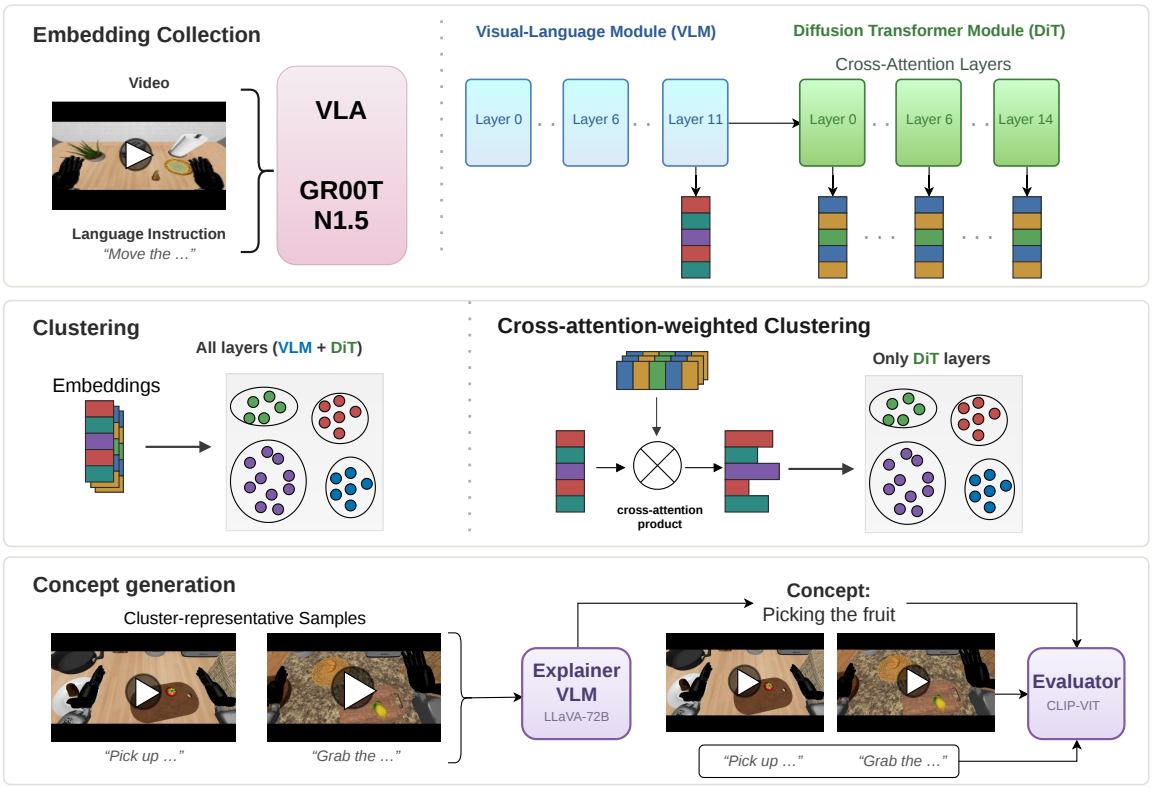


Figure 1: Overview of our framework. We collect embeddings (coloured bars) from the last VLM backbone layer and the DiT action decoder. After the collection, we compute two different cluster typologies: baseline clustering applied to all embeddings from each layer; our embedding-weighting module steers the VLM backbone tokens according to their influence in the cross-attention layers. Finally, we generate concepts based on the top-n nearest samples for each cluster and evaluate each of them with CLIP.

This technique performs consistent improvements across unsupervised clustering benchmarks; specifically, we observe a 62.1% increase in Silhouette scores and an 8% improvement in the Davies-Bouldin (DB) Index.

- We include a concept extraction pipeline that assigns a human-interpretable concept to each cluster, derived from the top-n nearest samples to the cluster centroid, labelled using an external VLM and evaluated using CLIP.

- Our analysis reveals that cross-attention layers consistently focus on specific tokens from the VLM backbone representations across different diffusion timesteps. The resulting clusters exhibit a clear disentanglement of spatiotemporal and kinematic features, suggesting that the cross-attention mechanism effectively incorporates the visuolinguistic context in the action diffusion process.

2 Related Work

2.1 Vision-Language-Action Models

Recent works on Vision Language Action Models (Kim et al., 2025; Ghosh et al., 2024; Black et al., 2025; Belkhale et al., 2024; Shi et al., 2025; NVIDIA et al., 2025) aim to create universal models capable of generating robot actions or trajectories across different embodiments. The training paradigms often follow those of recent LLMs: large datasets are collected that span a wide range of trajectories and scenarios to enable generalisation (O'Neill et al., 2024). Commonly, VLAs map language instructions and a visual observation to robotic actions. Generally, the visual input is first processed into vision token embeddings, which are then concatenated with the embedded language tokens and fed into the Large Language Model (LLM) backbone. The backbone LLM then produces tokens directly, which map to specific robotic actions (Kim et al., 2025; Belkhale et al., 2024), or returns the embeddings of intermediate

to final layers, which are then further processed by specific action decoder heads. These action decoder heads utilise different strategies to generate multi-dimensional continuous actions, including flow matching (NVIDIA et al., 2025; Black et al., 2025; Shi et al., 2025) and diffusion decoding strategies (Ghosh et al., 2024), or simple linear transformations (Belkhale et al., 2024). Finally, to generalise to long-horizon tasks, some methods explicitly model hierarchical sub-tasks (Belkhale et al., 2024; Shi et al., 2025). Closest to our approach, Häon et al. (2025) use clustering of FFN activations projected into token space to identify semantically meaningful directions for steering VLA behaviour. However, their clustering does not probe the action decoder. We close this research gap by specifically investigating the action decoding process.

2.2 Latent Analysis in Foundation Models

Latent analysis methods reduce high-dimensional embeddings into more compact representations that are generally more interpretable than the original embedding, allowing them to be linked with human-interpretable concepts. Sparse AutoEncoders (SAEs) are prominent approaches (Huben et al., 2024) that integrate an autoencoder into a layer of a pretrained model. SAEs contain a single hidden layer trained to maintain sparse neuron activations for each reconstructed token. According to the superposition theory (Elhage et al., 2022), these new neurons encode the most salient information needed to rebuild the original embedding. Although SAEs have typically been applied to LLMs, some researchers have also extended their usage to VLAs and VLMs (Grant et al., 2026; Zhang et al., 2025; Pach et al., 2025). However, SAEs require large amounts of training data, making them slow to adapt to the fast pace of robotics development. A different line of research relies on clustering techniques to group similar embeddings. Unlike SAEs, clustering does not require large amounts of training data, making it more suitable for robotic tasks. Fel et al. (2023) introduced a unifying theoretical framework that reframes concept extraction as a form of dictionary learning, while Hawasly et al. (2024) evaluated the quality of discovered concepts to compare clustering algorithms.

3 Methodology

A visual overview of the LAVLA framework is presented in Fig. 1. The first step in the LAVLA process is to collect the embeddings of selected layers during forward passes of a VLA. These embeddings are then clustered layer-wise using an unsupervised clustering algorithm. Optionally, a cross-attention-based embedding-weighting module and a PCA-based dimensionality reduction can be applied before the clustering; when both are utilised, embedding-weighting precedes dimensionality reduction. Finally, for each cluster, we select the top-n representative samples, generate a concept label using an external VLM, and evaluate it using CLIPSIM (Liu et al., 2024b).

3.1 Embedding Collection

Our investigation of the inner workings of VLAs focuses on NVIDIA's GR00T N1.5 model (NVIDIA et al., 2025), a state-of-the-art, 3B parameter VLA. GR00T N1.5 consists of an Eagle VLM (Li et al., 2025) backbone to process vision and language inputs into multimodal representations and an action decoder head, which produces the actual output command. The action decoder head is a diffusion transformer module that alternates between self-attention and cross-attention blocks, which denoise trajectories into continuous action vectors. The action decoder head receives noisy action tokens and the encoded robot state as input, while the fused vision-language embeddings from the Eagle backbone are exclusively integrated in the cross-attention layers. For each sample in our data selection, we collect the embeddings resulting from each cross-attention layer in the action decoder head of GR00T N1.5 as well as the 12th layer of the Eagle backbone, the last VLM layer before visuolingual embeddings are passed to the action decoder. The action decoder performs four iterative denoising steps; consequently, each layer is traversed four times to progressively refine the initially noisy action tokens. We store the embeddings of the action decoder with respect to the corresponding diffusion timestep. We use a part of the Open-X-Embodiment dataset[1] for our experiments. More details on the data preparation can be found in Appendix A.

[1]The data is publicly available at: https://huggingface.co/datasets/NVIDIA/PhysicalAI-Robotics-GR00T-X-Embodiment-Sim

3.2 Clustering

Once the embeddings have been collected, we cluster the embeddings of all cross-attention layers of the diffusion transformer action decoder head and the last layer of the GR00T Vision-Language-Action Model using a hierarchical clustering method. Specifically, we use agglomerative hierarchical clustering. Initially, we explored DBSCAN and k-means but ultimately selected agglomerative clustering based on its superior early-stage performance. The algorithm initialises by assigning each sample to its own cluster, then recursively merges the closest clusters based on a predefined distance threshold and linkage method. This process continues until either no more merges are possible or an alternate stopping criterion is satisfied. We perform agglomerative clustering until no further merges are possible.

3.2.1 Cross-Attention-Based Embedding-Weighting

Early in the action denoising process of the VLA diffusion head, the embeddings contain mostly noise. Only in the later layers and after several diffusion steps, the model generates the final output from a more defined latent space. In the case of the GR00T model, cross-attention layers are used during the denoising process, which incorporate the visuolingual outputs of the backbone VLM at each timestep. We want to measure the focus allocated to each visuolingual source token S_{VL} during the forward pass in GR00T's action diffusion head, to gain insight into what matters to the cross-attention layer at a specific time and reduce the impact of less impactful tokens. This also eliminates the disturbance of the clustering due to a noisy latent space early in the denoising process.

We extract the attention weight matrix α directly from the cross-attention layers. This matrix has dimensions $a \times n$, with a referring to the length of the action token sequence and n referring to the length of the visuolanguage token sequence. In the case of cross-attention in GR00T's action decoder, the query matrix Q_A originates from encoding the previous self-attention layer, while the key matrix K_{VL} is projected from the injected vision-language backbone embeddings. Following standard practice in scaled dot-product attention, the scores are scaled by the factor $\sqrt{d_k}$ to mitigate softmax saturation.

$$\alpha^{a \times n} = \text{softmax}\left(\frac{Q_A K_{VL}^T}{\sqrt{d_k}}\right) \tag{1}$$

We average the weights in α for each source token j across all target tokens i to receive a vector of impact scores $M = [m_1, \ldots, m_n]$, with m_j defined as:

$$m_j = \frac{1}{n} \sum_{i=1}^{n} \alpha_{ij} \tag{2}$$

Next, we reduce across all heads in multi-head attention by using the mean before applying the normalisation. We receive the normalised weight per source token by applying the softmax function to the vector of impact scores M. This results in the normalised weight vector $W = [w_1, \ldots, w_n]$, with:

$$w_j = \frac{\exp(m_j)}{\sum_{k=1}^{n} \exp(m_k)} \tag{3}$$

Finally, we apply the Hadamard product $\odot$ between the normalised weight vector W and the visuolingual tokens $S'_{VL} = S_{VL} \odot W$. This reduces the impact of less influential tokens while boosting the more important ones in the resulting weighted vision-language token sequence S'_{VL}. The embeddings S'_{VL} are specific to each cross-attention layer and diffusion timestep, allowing us to analyse the importance of tokens in the backbone vision-language sequence during the action denoising process.

3.2.2 Cluster Evaluation

To evaluate layer-wise cluster quality, we assess the spatial density and separation of the identified clusters using three standard metrics: the Silhouette Coefficient (Rousseeuw, 1987), the Caliński-Harabasz (CH) Index (Caliński and Harabasz, 1974), and the Davies-Bouldin (DB) Index (Davies and Bouldin, 1979). These metrics are applied across all layers of interest as defined in Section 3.1. Higher CH Index, lower DB Index, and higher Silhouette score indicate better separation of clusters.

To further investigate the flow of individual samples through the network, we calculate the Overlap Coefficient (OC) (Verma and Aggarwal, 2020) and Jaccard Index (Verma and Aggarwal, 2020) between all possible pairs of clusters A and B. See Appendix C for the exact formulations. The Overlap Coefficient highlights containment of the smaller set within the larger, and reaches its maximum value of 1 when the smaller set is perfectly contained within the larger set. On the other hand, the Jaccard index balances the amount of shared elements and elements only contained in one of the two sets. The maximum value of 1 is only reached

if both sets are identical. To compare cluster assignments of different experiments and layers, we perform an additional alignment step as detailed in Appendix B.

3.3 Concept Generation

After completing the cluster computation, we generate a representative concept for each cluster. Based on prior work (Zhang et al., 2025; Liu et al., 2024b), we select the top-n cluster samples nearest to the centroid and query a VLM to identify the human-interpretable pattern shared across them. Each sample consists of two elements: the textual command and the action video, which is decomposed into a sequence of frames, 10 total frames for a 5-second clip. The VLM is then prompted to generate a short sentence that summarises each sample separately. These resulting sentences are further passed together to the VLM a second time to synthesise a single concept label for the cluster. Finally, we evaluate the overall performance using the CLIPSIM metric (Liu et al., 2024b), computed as the CLIP similarity between the final concept and the single frames of each cluster sample.

4 Experiment Setup

Embeddings for our analysis were collected on a single NVIDIA A100 GPU using a fully frozen GR00T N1.5 model. We collect 1,000 embeddings per task from the 24 tasks in the dataset in ca. 4 hours. Consecutive samples of the same task and episode in the resulting 24,000 embeddings are offset by 10 frames. We cluster the embeddings using agglomerative clustering with Euclidean distance as metric and average linkage. The latent representations change between individual layers, hence there is no one-size-fits-all distance measure for all layers. We notice, however, that it yields the best results within certain ranges within the same components. We present an exhaustive list of hyperparameters in Appendix F. For the concept generation part, we adopt LLaVA 72B (Liu et al., 2024a) as the generator, and set top-n=5 due to computational constraints. For the same reason, we limited our analysis to the 10 most populous clusters for each layer/cluster technique pair. This threshold is specific to non-PCA-cluster methods, since these techniques create, on average, 7 clusters, as shown in Table 1. The concept generation pipeline was run on two NVIDIA A100 GPUs for ca. 3 hours.

5 Results

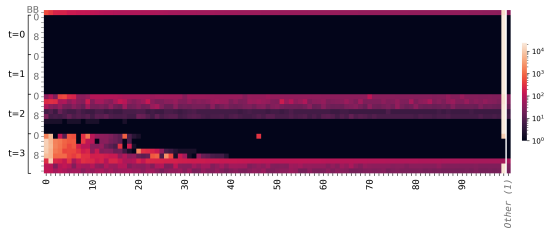
To evaluate the effectiveness of the LAVLA framework and investigate the latent space of the GR00T action decoding process, we investigate how clusters evolve during the diffusion forward passes, highlight patterns visible in the cluster contents, and present quantitative metrics on the effectiveness of different LAVLA configurations. We present our results across individual layers. Cross-attention layer weights are reused during the 4-step diffusion process (beginning with step 0, ending at step 3). Accordingly, we further differentiate embeddings from cross-attention layers by their diffusion timestep t, i.e. the cross-attention layer 2 with t=1 and cross-attention layer 2 with t=3 use the same weights, but the embeddings have been extracted at diffusion timestep t=1 and t=3 respectively. Results referencing the embeddings of the backbone layer are annotated as "BB", while we provide the layer index (starting at 0) to differentiate cross-attention layers (for example, see the y-axis of Figure 2a).

5.1 Clustering Results

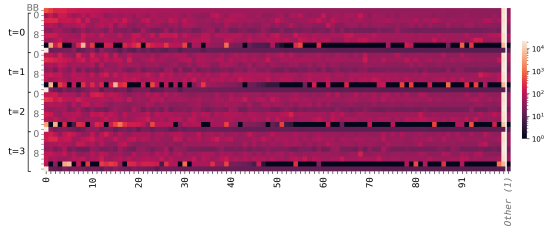
Cross-attention-based embedding-weighting leads to measurable gains in clustering performance. We compare the general performance of different LAVLA configurations using common unsupervised cluster-evaluation metrics, and display the detailed metrics in Table 1. Adding the embedding-weighting mechanism has an overall positive effect, especially in the case of unreduced embeddings. It improves the Silhouette scores by 62.1% in the scenario without PCA and by 24.0% in the scenario with PCA. DB Index scores improve in each scenario by ca. 8%. This is likely due to the embedding-weighting mechanism not relying on noisy action tokens but rather on differently amplified backbone embeddings, which bear semantic meaning over the full process. While the action token embeddings start as noise, they only become semantically informative towards the end of the diffusion process. CH Index scores only improve in the case in which PCA is also applied. This is likely a result of the extremely high number of clusters in the unweighted case, which leads to dense clusters with only a single sample, inflating the resulting score. Based on these results, we hypothesise that the embedding-weighting mechanism would also positively impact other interpretability frameworks

Table 1: Clustering performance across configurations

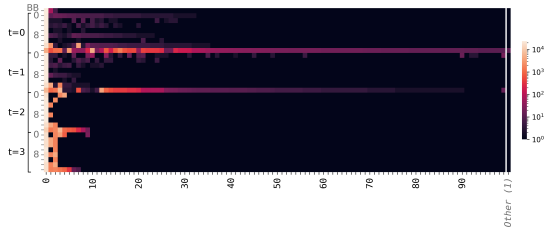
LAVLA	Silhouette Score↑	DB Index↓	CH Index↑	N clusters
Baseline	0.0837 ± 0.05	1.4734 ± 0.18	402.12 ± 358.6	15703.8 ± 10282.3
w/ weighted	0.1357 ± 0.03	**1.3617 ± 0.27**	56.41 ± 21.08	3835.2 ± 3479.0
w/ PCA	0.1324 ± 0.06	1.5858 ± 0.24	1063.9 ± 481.4	25.93 ± 72.1
w/ PCA w/ weighted	**0.1642 ± 0.06**	1.4682 ± 0.15	**1651.0 ± 775.8**	7.1515 ± 2.41



(a) Baseline clusters through layers



(b) Weighted Clustering clusters through layers



(c) PCA Clustering clusters through layers

(d) Weighted PCA Clustering clusters through layers

Figure 2: Heatmaps of cluster sizes across different layers and LAVLA configurations.

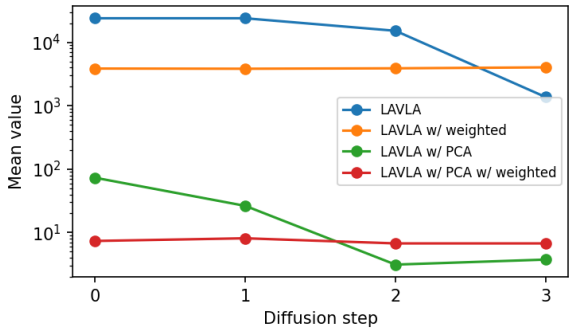


Figure 3: Mean number of clusters over diffusion timesteps

that investigate model latent spaces, e.g., SAEs, and suggest their inclusion in future work.

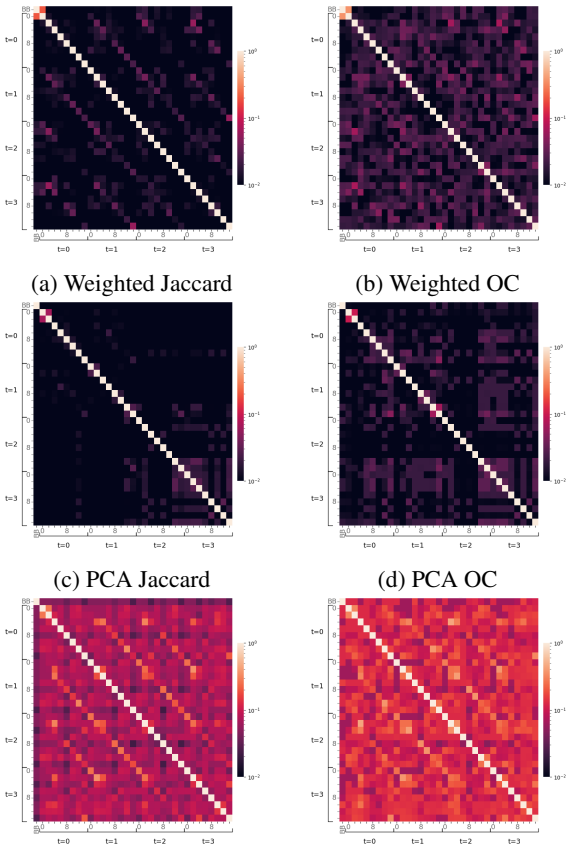
Latent representations exhibit increased structural coherence at deeper architectural layers and later diffusion timesteps. The unreduced latent embeddings of the GR00T model are represented as a $(n \times d)$ matrix, with n referring to the sequence length of the vision and language input tokens, and d being the internal dimension of the per-token embeddings. The vision-language token sequence length varies slightly with the number of tokens in the language instruction; the number of extracted vision tokens remains constant. Averaging across the sequence length results in a set of d-dimensional embedding vectors that may have been amplified by the embedding-weighting process depending on the experiment configuration. Influenced by the curse of dimensionality (Beyer et al., 1999), applying agglomerative clustering on these high-dimensional sets of embedding vectors results in a large number of clusters, with only a few select clusters containing more than a few samples. This is highlighted by the heatmaps in Figures 2a to 2d which display the largest 100 clusters over layers and diffusion timesteps.

As shown in Figure 3, the latent representations become increasingly structured as the diffusion process nears completion. This progression toward a more organised state leads to a lower cluster count, suggesting that disparate embeddings are being consolidated into cohesive groups. Applying PCA for dimensionality reduction reduces the embedding dimension to 100 per embedding. Consequently, the overall number of extracted clusters drops significantly. We observe the same trend as with the uncompressed embeddings, where the number of clusters becomes lower towards later diffusion timesteps. A visual inspection of the latent space under PCA visualisations reveals the same

organising pattern in Appendix E.

However, this same trend is less prominent when incorporating the embedding-weighting mechanism. This likely stems from the weighted backbone embeddings being directed towards certain clusters during the cross-attention mechanism throughout the whole process. In the absence of this weighting mechanism, the noisy action embeddings require additional denoising iterations before the action decoder layers can resolve the latent space into a structured state. The UMAP visualisations in Appendix D further highlight the effect of the weighting mechanism.

5.2 Cluster Consistency



(a) Weighted Jaccard (b) Weighted OC

(c) PCA Jaccard (d) PCA OC

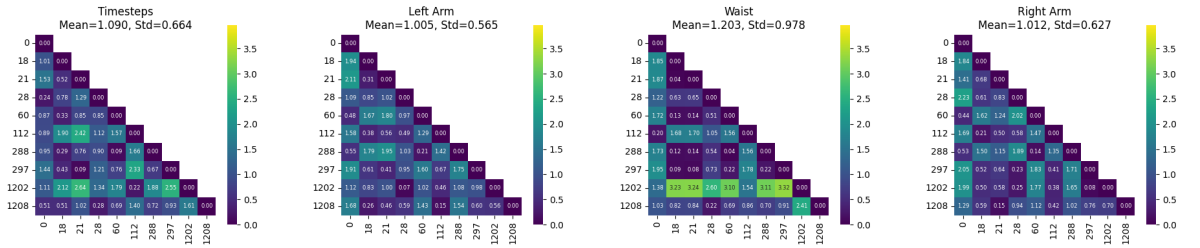
(e) Weighted + PCA Jaccard (f) Weighted + PCA OC

Figure 4: Layer-wise cluster overlap matrices for different LAVLA configurations.

Cluster assignments are stable across diffusion timesteps within the same layer. To investigate whether the cluster assignments remain consistent across different layers, we computed the pairwise Jaccard and Overlap Coefficient metrics of the largest 100 clusters between layers and diffusion steps. The results are displayed in the heatmaps in Figures 4b to 4f. We did not include the heatmaps for the case without either PCA or the embedding-weighting mechanism, as the cluster algorithm was unable to find meaningful assignments during early diffusion timesteps (see Figure 3) and assigns each sample to its own cluster. As a result, this inflates the corresponding metrics in the early layers. The heatmaps 4c and 4d reveal mainly two things: first, cluster assignments stay more consistent closer to the final outputs. This is largely a consequence of the diffusion process, which leads to less noisy embeddings over time, and the model having to decide on a much smaller number of possible action vectors compared to the vast possibilities of language-vision-state inputs. Second, the repeating patterns in the heatmaps suggest that cluster assignments between the same layer but different diffusion timesteps remain consistent. They are especially highlighted by the diagonal lines visible in Figures 4b, 4e, and 4f. We hypothesise that this is due to the model using different layers to pay special attention to certain features in the backbone embeddings. E.g., one layer might consider colour representations more important while another might specifically incorporate object-centric features. Figures 4c and 4d further illustrate how the latent space becomes more ordered towards the end of the diffusion process, likely resulting in more consistent clusters (see patterns in bottom right corner). Our LAVLA framework cannot reveal which features precisely are important for different layers, and as such we leave this investigation for future work on VLAs.

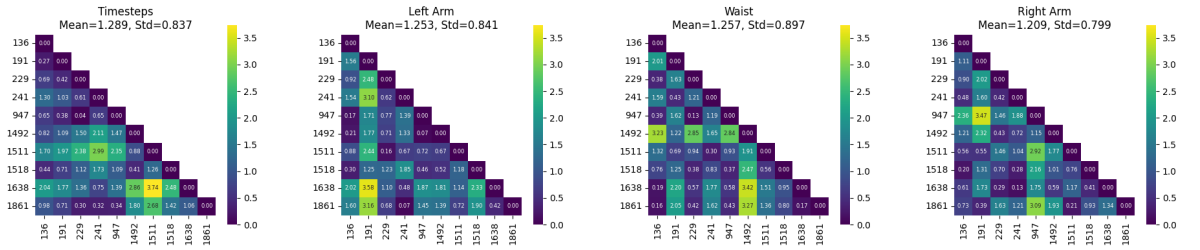
5.3 Latent Space Organisation

Clusters exhibit distinct temporal characteristics. As shown in Figures 5a and 6a, the latent space of the action diffusion head exhibits higher temporal separability than the VLM backbone. Besides a visible increase in separability between clusters, this observation is supported by the increased mean within the displayed 10 largest clusters. This suggests that the model discriminates between generalisable states commonly encountered at specific intervals within different episodes, implying spatiotemporal awareness. Furthermore, even if this generalisation were driven solely by visual features common to specific episode timesteps, it would nonetheless indicate that the model has developed a structured spatial representation of the task environment. Further experiments are required to fully comprehend the temporal understanding capabilities of the model.



(a) Timesteps (b) Left arm displacement (c) Waist displacement (d) Right arm displacement

Figure 5: Wasserstein distances between feature distributions of top clusters within VL backbone



(a) Timesteps (b) Left arm displacement (c) Waist displacement (d) Right arm displacement

Figure 6: Wasserstein distances between feature distributions of top clusters of cross-attention layer 14 at final diffusion timestep (t=3).

Table 2: Mean ± std CLIPSIM↑ (video vs. concept) per configuration and layer.

Layer	Backbone L11	CA0	CA6	CA12	CA14
unweighted	0.214 ± 0.014	**0.229** ± 0.009	0.228 ± 0.011	0.226 ± 0.014	0.218 ± 0.016
weighted	0.214 ± 0.014	0.225 ± 0.016	0.216 ± 0.015	0.206 ± 0.019	0.213 ± 0.024
pca	**0.219** ± 0.013	0.223 ± 0.015	**0.233** ± 0.018	**0.233** ± 0.022	**0.221** ± 0.020
weighted_pca	**0.219** ± 0.013	0.223 ± 0.015	0.209 ± 0.024	0.225 ± 0.010	**0.221** ± 0.014

Clusters separate body part usage. The plots in Figures 5b to 5d, and 6b to 6d reveal the same trend with a ca. 20% increase in mean intra-cluster Wasserstein distance for the arm joints observed in Figures 5b and 6b for the left arm, and Figures 5d and 6d for the right arm. While less prominent, it can still be observed in the waist embeddings in Figures 5c and 6c.

5.4 Concept Generation Results

Table 2 reports CLIPSIM results across the four main configurations. PCA-based and weighted variants tend to yield slightly more faithful concepts, with PCA consistently improving performance across layers. However, CLIPSIM scores should be interpreted with caution: robotic manipulation datasets typically exhibit low visual diversity and the input clips are short, which likely imposes a ceiling on achievable performance regardless of the clustering method. For the same reason, the final concepts identified are semantically close to each other, mostly involving picking objects and returning them into a container. Nevertheless, we believe that since the DiT is a denoising process, the concepts may persist or evolve across layers rather than remain fixed, while still focusing on the same macro task. This experiment demonstrates the feasibility of concept construction, though concept quality remains a limitation to address in future work. We present some generated concept examples in Appendix H.

6 Conclusion

We present LAVLA, a framework for analysing the latent spaces of Vision-Language-Action (VLA) Models through clustering. By amplifying the tokens prioritised during cross-attention, LAVLA elucidates the connection between backbone feature extraction and the iterative denoising process of the action decoder. Through comparative experiments, we demonstrate that our embedding-weighting mechanism significantly enhances rep-

resentational clarity. Our results indicate that task-specific representations are distributed across distinct cross-attention layers and maintain high spatiotemporal consistency across the diffusion trajectory. By making VLA latent spaces more transparent, LAVLA moves us closer to trustworthy, language-driven robotic systems.

Ethical Considerations

The presented work aims at improving the understanding of the inner workings of VLAs. The LAVLA framework poses no inherent risk to aspects such as data privacy, as models and data used in this work are publicly available. Investigating the latent space of pretrained models also does not introduce risks regarding inclusivity and biases, but helps in detecting these. Deploying VLAs in real-world scenarios on a variety of embodiments could lead to catastrophic consequences if the model is not tested and thoroughly understood. With increased understanding of the VLA latent space, LAVLA improves the safety of VLAs.

Limitations

Our experiments are limited to a single dataset and model; results may not generalise to other datasets or architectures, which we leave for future work. Concept generation and its unsupervised semantic verification remain challenging without ground-truth annotations, and since concepts are derived from static images, temporal or dynamic aspects of scenes are more difficult to capture. Our analysis is bounded by the VLA's input horizon, which only covers short clips, so our findings may not extend to longer-horizon behaviours. Finally, our clustering analysis relies on agglomerative clustering, whose assumptions may not fully capture the structure of complex, non-spherical manifolds, potentially influencing the resulting concept groupings.

References

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quan Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. 2024. RT-H: Action Hierarchies using Language. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands. RSS.

Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. 1999. When Is "Nearest Neighbor" Meaningful? In *Database Theory — ICDT'99*, pages 217–235, Berlin, Heidelberg. Springer.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, and 16 others. 2025. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In *9th Annual Conference on Robot Learning*.

T. Caliński and J Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27.

David L. Davies and Donald W. Bouldin. 1979. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, arXiv:2209.10652.

Thomas Fel, Victor Boutin, Mazda Moayeri, Rémi Cadène, Louis Bethune, Léo Andéol, Mathieu Chalvidal, and Thomas Serre. 2023. A holistic approach to unifying automatic concept extraction and concept importance estimation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 54805–54818, Red Hook, NY, USA. Curran Associates Inc.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. 2024. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands. RSS.

Bryce Grant, Xijia Zhao, and Peng Wang. 2026. Not All Features Are Created Equal: A Mechanistic Study of Vision-Language-Action Models. *Preprint*, arXiv:2603.19233.

Bear Häon, Kaylene Caswell Stocking, Ian Chuang, and Claire Tomlin. 2025. Mechanistic Interpretability for Steering Vision-Language-Action Models. In *9th Annual Conference on Robot Learning*.

Majd Hawasly, Fahim Dalvi, and Nadir Durrani. 2024. Scaling up Discovery of Latent Concepts in Deep NLP Models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–806, St. Julian's, Malta. Association for Computational Linguistics.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse Autoencoders Find Highly Interpretable Features in Language Models. In *The Twelfth International Conference on Learning Representations*, pages 7827–7845, Vienna, Austria. ICLR.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. 2025. OpenVLA: An Open-Source Vision-Language-Action Model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713, Munich, Germany. PMLR.

Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

Zhiqi Li, Guo Chen, Shilong Liu, Shihao Wang, Vibashan VS, Yishen Ji, Shiyi Lan, Hao Zhang, Yilin Zhao, Subhashree Radhakrishnan, Nadine Chang, Karan Sapra, Amala Sanjay Deshmukh, Tuomas Rintamaki, Matthieu Le, Ilia Karmanov, Lukas Voegtle, Philipp Fischer, De-An Huang, and 8 others. 2025. Eagle 2: Building Post-Training Data Strategies from Scratch for Frontier Vision-Language Models. *arXiv:2501.14818*.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. 2024b. EvalCrafter: Benchmarking and Evaluating Large Video Generation Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22139–22149.

Pingchuan Ma, Lennart Rietdorf, Dmytro Kotovenko, Vincent Tao Hu, and Björn Ommer. 2025. Does VLM classification benefit from LLM description semantics? In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25, Philadelphia, Pennsylvania, US. AAAI Press.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, 3(29):861.

NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, and 24 others. 2025. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots. *Preprint*, arXiv:2503.14734.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, and 260 others. 2024. Open X-Embodiment: Robotic Learning Datasets and RT-X Models : Open X-Embodiment Collaboration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, Yokohama, Japan. IEEE.

Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. 2025. Sparse autoencoders learn monosemantic Features in vision-language models. *Poster on Neural Information Processing Systems (NeurIPS 2025)*.

William Peebles and Saining Xie. 2023. Scalable Diffusion Models with Transformers . In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182, Los Alamitos, CA, USA. IEEE Computer Society.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Lucy Xiaoyang Shi, Brian Ichter, Michael Robert Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. 2025. Hi Robot: Open-Ended Instruction Following with Hierarchical Vision-Language-Action Models. In *Forty-Second International Conference on Machine Learning*.

Matthew Stephens. 2000. Dealing With Label Switching in Mixture Models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 62(4):795–809.

Vijay Verma and Rajesh Kumar Aggarwal. 2020. A comparative analysis of similarity measures akin to the Jaccard index in collaborative recommendations: Empirical and theoretical perspective. *Social Network Analysis and Mining*, 10(1):43.

Yuan Zang, Tian Yun, Hao Tan, Trung Bui, and Chen Sun. 2025. Pre-trained Vision-Language Models Learn Discoverable Visual Concepts. *Transactions on Machine Learning Research*, 2025.

Kaichen Zhang, Yifei Shen, Bo Li, and Ziwei Liu. 2025. Large Multi-modal Models Can Interpret Features in Large Multi-modal Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3650–3661, Honolulu, Hawaii, US. IEEE.

A Data

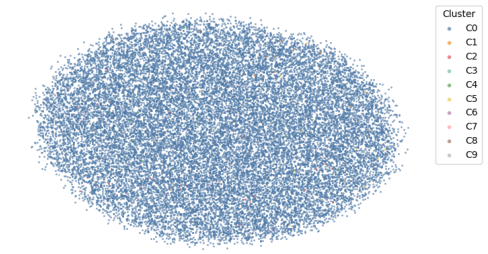
We collect embeddings from a part of the Open-X-Embodiment dataset featuring a GR1 robot performing humanoid robot tabletop manipulation. In total, we have 240k trajectories across 24 tasks at our disposal with visual observations from the ego perspective of the GR1 robot [2]. Due to memory constraints, we load 1000 samples per task, resulting in a final dataset of 24k observation-action pairs. These samples are used to collect embeddings via the GR00T model. Each task of the original dataset contains 10,000 samples, distributed across episodes of varying lengths. To spread our selection evenly across different episodes we include every 10th sample in our selection.

Consistent with standard VLA architectures (NVIDIA et al., 2025; Kim et al., 2025), the observation consists of the current frame, the task instruction in natural language, and a robot state vector containing the current joint configuration. The ground-truth action is normally used for behaviour cloning during training of VLAs, and not required in our investigation. The state information contains the joint configuration of the hands, arms, neck, and waist joints. The tasks span a variety of tabletop manipulation tasks involving the picking up and placing down of different objects. We include all 24 tasks in our investigation.

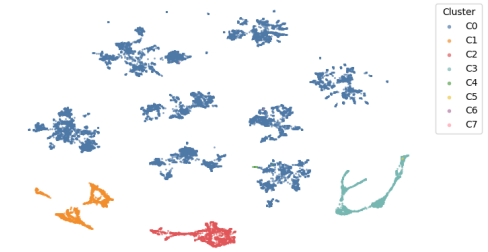
B Cluster Alignment

The clusters are computed independently for each layer and diffusion timestep combination (if applicable). This results in clusters between layers possibly containing the same or similar samples but having different cluster identifiers assigned. This phenomenon is known as label switching (Stephens, 2000). To ensure consistent assignment of cluster identifiers, an additional alignment step is required. We resolve the label switching occurring between cluster assignments of different layers and diffusion timesteps by performing optimal bipartite matching with the Hungarian algorithm (Kuhn, 1955). By using the degree of sample overlap between clusters to define the cost matrix, we can consistently trace the evolution of cluster memberships as data flows through the model.

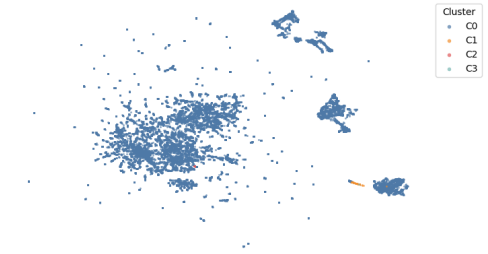
[2] Fourier GR-1 robot: https://www.fftai.com/products-gr1



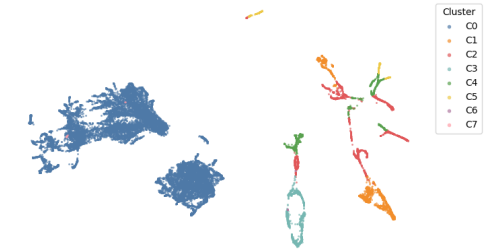
(a) UMAP of PCA compressed embeddings in layer 0 at diffusion timestep 0.



(b) UMAP of weighted and PCA compressed embeddings in layer 0 at diffusion timestep 0.



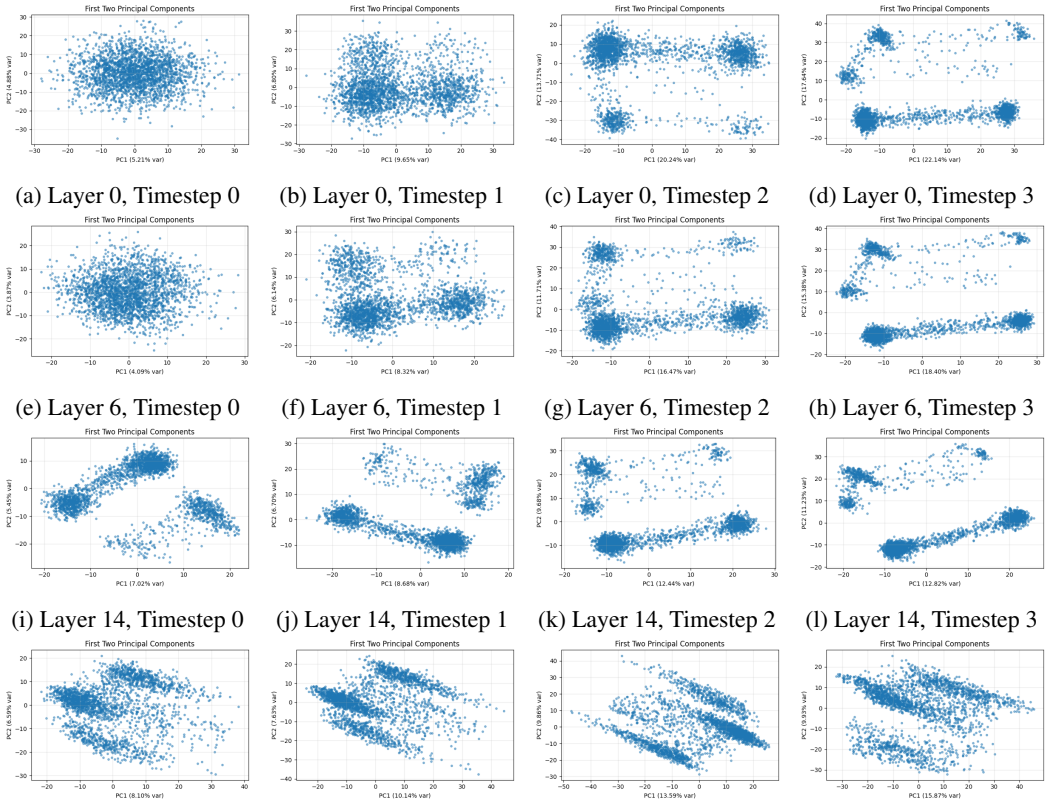
(c) UMAP of embeddings from VLM backbone layer 12 (no diffusion timestep).



(d) UMAP of PCA compressed embeddings in layer 14 at diffusion timestep 3.

(e) UMAP of weighted and PCA compressed embeddings in layer 14 at diffusion timestep 3.

Figure 7: Comparative UMAP visualisations of the latent space at different layers and diffusion timesteps using PCA-compressed embeddings without and with a weighting mechanism.



(a) Layer 0, Timestep 0 (b) Layer 0, Timestep 1 (c) Layer 0, Timestep 2 (d) Layer 0, Timestep 3

(e) Layer 6, Timestep 0 (f) Layer 6, Timestep 1 (g) Layer 6, Timestep 2 (h) Layer 6, Timestep 3

(i) Layer 14, Timestep 0 (j) Layer 14, Timestep 1 (k) Layer 14, Timestep 2 (l) Layer 14, Timestep 3

(m) Layer 15, Timestep 0 (n) Layer 15, Timestep 1 (o) Layer 15, Timestep 2 (p) Layer 15, Timestep 3

Figure 8: Visualisation of the first two principal components of the latent space across layers (rows) and timesteps (columns). Each subplot shows the latent space structure for a specific layer–timestep combination, illustrating how latent representations evolve through the network.

C Formulas

The Overlap Coefficient $O(A, B)$ is computed as follows:

$$O(A,B) = \begin{cases} 0 & \text{if } \min(|A|,|B|) = 0 \\ \frac{|A \cap B|}{\min(|A|,|B|)} & else \end{cases} \tag{4}$$

Whereas the Jaccard Index $J(A, B)$ is computed as displayed in equation 5:

$$J(A,B) = \begin{cases} 0 & \text{if } |A \cup B| = 0 \\ \frac{|A \cap B|}{|A \cup B|} & \text{else} \end{cases} \tag{5}$$

This gives us a quantifiable way of measuring how much the internal representations of the model change over different layers and diffusion steps.

D Cluster Visualisation

The embedding-weighting module uncovers structural patterns during the initial stages of the diffusion process. The UMAP (McInnes et al., 2018) visualisations in Figures 7a through 7e demonstrate how the weighting mechanism structures the VLA latent space. This effect is particularly evident in the early layers, where the weighted embeddings form significantly more coherent clusters than their unweighted counterparts. The first cross-attention layer at the first diffusion timestep takes as input barely processed noisy action token representations, which leads to the unstructured, circular-shaped latent space in Figure 7a. In contrast, the UMAP visualisation of the embeddings of the last layer of the VLM display a more ordered latent space. As our embedding-weighting mechanism directly incorporates the influence of the VLM embeddings, we are able to improve the ordering of the latent space in early noisy embeddings, based on the actual attention of the model.

E PCA Analysis

We investigate the principal components of GR00T's internal representations across 10,000 samples from all tasks by visualising the first two principal components and investigating the explained variance of individual components. Fig-

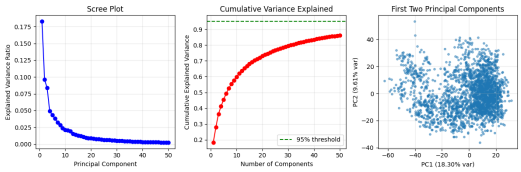


Figure 9: First two principal components of vision-language features produced by VLM backbone.

ure 9 visualises the first two principal components of the vision-language embeddings. The Scree plot (left) reveals that the first ten principal components seem to carry the highest amount of explained variance. However, the plot of the explained cumulative variance (middle) barely exceeds 60% of explained variance. This suggests that individual components are responsible only for a small portion of the total information encoded by the single embedding. The visualisation of the first two components (right) provides some insight into the embedding space, which already provides some recognisable structures.

As can be observed by the visualizations of the first two principal components of the VLA's latent space in Figure 8, the noisy action tokens already show the first signs of clear separability within the first diffusion timestep (see Figure 8 (a),(e), and (i)). Within the same layer, stronger separation occurs over time, which is visible especially across layer 6 (see Figure 8 (e) - (h)) where a single noisy cluster splits over time into 5 visually distinct clusters. Accordingly, the variance of the principal components increases over time.

F Hyperparameters

Table 3 contains the hyperparameters yielding the best results after sweeping over different distance threshold ranges.

G Violin plots of top clusters

The Wasserstein distances in Figures 5 and 6, provide a quantitative comparison of the different distributions of selected features in the top clusters. Figures 10 and 11 provide an additional visualisation of these clusters, highlighting the model's latent space organisation at different layers in the model.

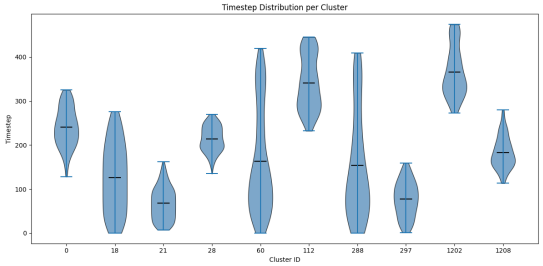
H Concept generation examples

Figures 12a and 12b show two example cluster sample sets used in the concept-generation process.

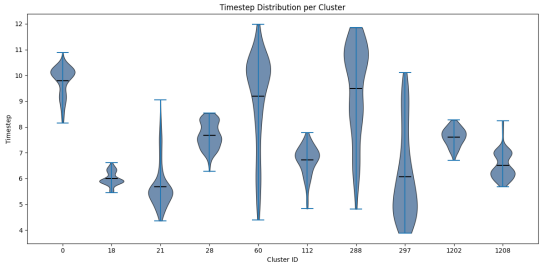
Table 3: Exhaustive list of distance thresholds per layer.

Layer	Distance Threshold
LAVLA (baseline)	
VLM Backbone	9.0383
Cross-Attention 0	109.1253
Cross-Attention 2	109.9187
Cross-Attention 4	108.6459
Cross-Attention 6	103.7606
Cross-Attention 8	108.2924
Cross-Attention 10	90.0302
Cross-Attention 12	86.9423
Cross-Attention 14	85.0988
LAVLA w/ weighted	
VLM Backbone	9.0383
Cross-Attention 0	2.6785
Cross-Attention 2	2.7405
Cross-Attention 4	2.9376
Cross-Attention 6	2.8599
Cross-Attention 8	3.6135
Cross-Attention 10	4.1296
Cross-Attention 12	2.8211
Cross-Attention 14	3.2981
LAVLA w/ PCA	
VLM Backbone	79.9869
Cross-Attention 0	60.5738
Cross-Attention 2	60.9242
Cross-Attention 4	59.8374
Cross-Attention 6	61.5705
Cross-Attention 8	56.9619
Cross-Attention 10	59.1317
Cross-Attention 12	54.3040
Cross-Attention 14	47.7121
LAVLA w/ PCA w/ weighted	
VLM Backbone	79.9869
Cross-Attention 0	61.3591
Cross-Attention 2	59.0093
Cross-Attention 4	56.0642
Cross-Attention 6	66.1785
Cross-Attention 8	59.5633
Cross-Attention 10	62.7542
Cross-Attention 12	65.4799
Cross-Attention 14	55.7679

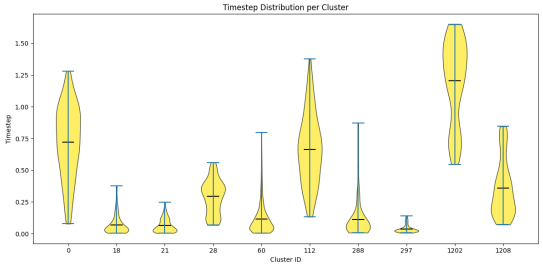
Each concept is derived from 5 videos, each decoded into 10 sequential frames and passed to the VLM (LLaVA-72B) alongside the corresponding task command, shown as the white caption below each image row.



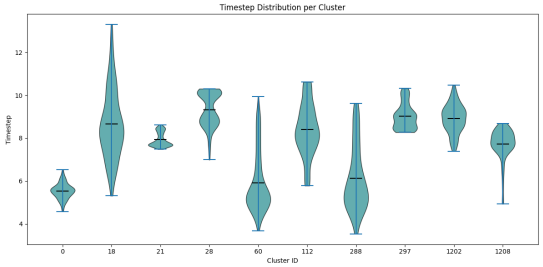
(a) Timesteps



(b) Left arm displacement

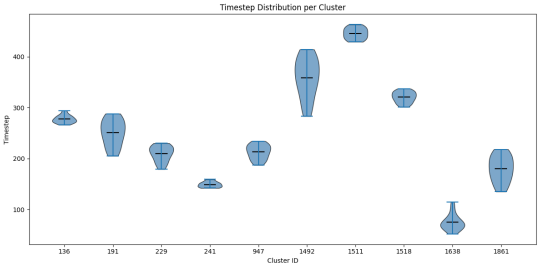


(c) Waist displacement

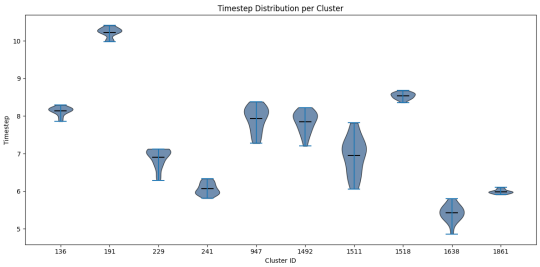


(d) Right arm displacement

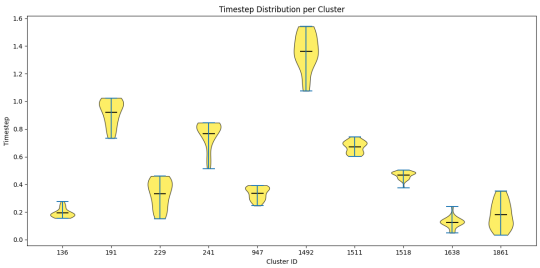
Figure 10: Violin plots of feature distributions of top clusters within VL backbone



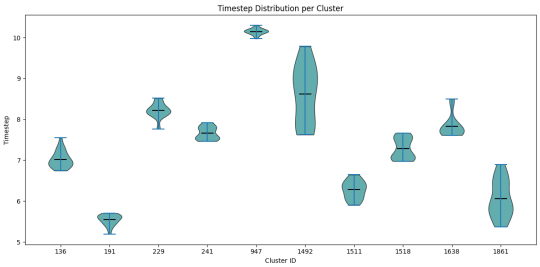
(a) Timesteps



(b) Left arm displacement



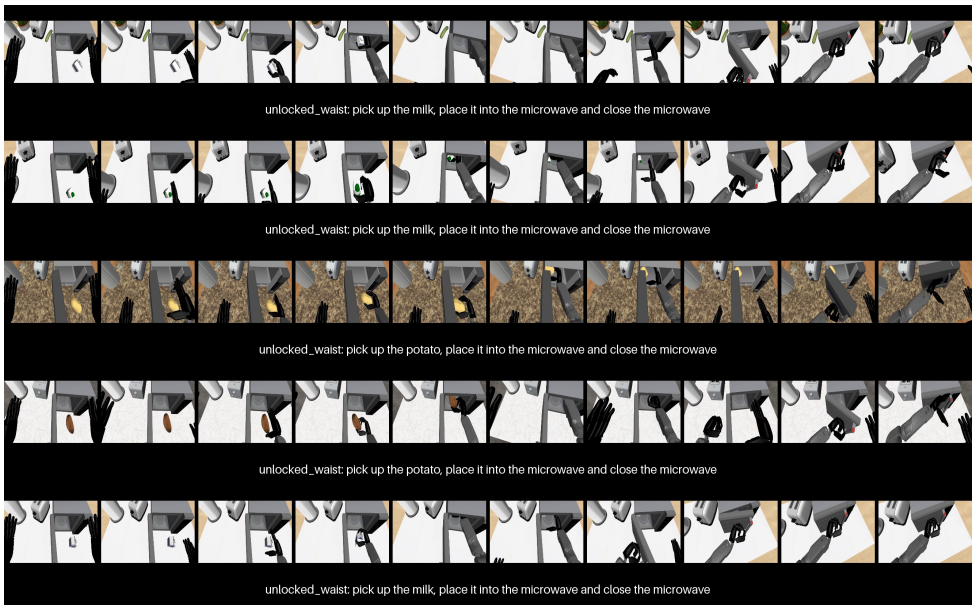
(c) Waist displacement



(d) Right arm displacement

Figure 11: Violin plots of feature distributions of top clusters of cross-attention layer 14 at final diffusion timestep (t=3).

(a) Final concept: The robot's hand picks up an object and places it into a container.



(b) Final concept: The robot picks up an object from a surface and places it into a container.

Figure 12: Examples of concept extraction for cross-attention layer 0 using both PCA and weighted clustering.