Appendix: Interpretable and Steerable Concept Bottleneck Sparse Autoencoders

Akshay Kulkarni[1] Tsui-Wei Weng[1] Vivek Narayanaswamy[2]
Shusen Liu[2] Wesam A. Sakla[2] Kowshik Thopalli[2]
[1]University of California, San Diego [2]Lawrence Livermore National Laboratory
{a2kulkarni,lweng}@ucsd.edu {narayanaswam1,liu42,sakla1,thopalli1}@llnl.gov

In this appendix, we present full implementation details along with additional analyses. To support reproducibility, we will also release our codebase and pretrained models. The appendix is organized as follows:

A. Limitations and Future Work

We acknowledge that the efficacy of our approach depends on the reliability of CLIP-Dissect in assigning accurate neuron-level concepts; however, continued advances in vision-language models are likely to enhance its performance. Extending and exploring hybrid approaches that combine the strengths of other unsupervised concept discovery methods such as transcoders [14] with user-specified concept control methods constitute our future work.

Recent work on LLM SAEs [2] showed "feature-splitting" where hierarchical features split into fine-grained features (*e.g.* "math" may be represented in the SAE by "algebra" and "geometry" neurons). It will be interesting to investigate if this phenomenon is connected to SAEs having limited concept coverage that we highlighted in this paper.

B. Implementation Details

CLIP-Dissect [9]. Consider a probing dataset of N images $\mathcal{D} = \{x_i \in \mathbb{X}\}_{i=1}^{N}$ where $\mathbb{X}$ is the space of images, a concept set $\mathcal{C} = \{c_k\}_{k=1}^{M}$ with M concepts in text form, and let layer l of model f being explained be denoted by f_l. CLIP-Dissect uses the probing set and a multimodal model, *e.g.* CLIP [15] with an image and text encoder E_I, E_T to identify concepts from $\mathcal{C}$ for individual neurons at the output of f_l.

The probing set $\mathcal{D}$ is passed through the CLIP image encoder E_I to obtain corresponding set of image embeddings $\{A_i = E_I(x_i)\}_{i=1}^{N}$. The concept set is passed through the CLIP text encoder E_T to obtain text embeddings $\{E_T(c_k)\}_{k=1}^{M}$. Next, a matrix $P \in \mathbb{R}^{N \times M}$ is computed as the inner product of the image-text embeddings with entries $P_{ik} = A_i^\top E_T(c_k)$, as CLIP image and text encoders have the same embedding dimensions. The layer l activations of a neuron j for the same probing set are denoted by $q_j = [f_l(x_1)_j, f_l(x_2)_j, \cdots, f_l(x_N)_j]$. Finally, each neuron j can be identified to have the concept $\arg\max_k \text{sim}(P_{:,k}, q_j)$ where $P_{:,k}$ is the k^{th} column of P. In other words, we compare each neuron's activations over the probing set with the corresponding activations of the CLIP model for each concept, and select the concept with the highest similarity. The maximum similarity itself (averaged across all neurons) is used as our *interpretability score*. The similarity function sim is soft weighted pointwise mutual information (soft-WPMI) following [9]. Please refer to the original paper [9] for more details.

Cosine-cubed similarity loss [10] $\mathcal{L}_{\text{int}}$. As discussed in Sec. 5.2 (main paper), we use a cosine-cubed similarity loss $\mathcal{L}_{\text{int}}$ to train the CB encoder E_{cb} to produce concept predictions c that match with CLIP zero-shot classifier predictions $\hat{y}$ for the same concept set $\mathcal{C}$. Concretely,

$$\mathcal{L}_{\text{int}}(c, \hat{y}) = \sum_{k=1}^{|\mathcal{C}|} -\frac{c_k^3 \cdot \hat{y}_k^3}{\|c_k^3\|_2 \|\hat{y}_k^3\|_2} \tag{1}$$

Here, c_k is the k^{th} concept prediction for the current mini-batch and $\hat{y}_k$ is the zero-shot CLIP prediction for concept k with the same mini-batch. Following [10], we also normalize both vectors $c_k, \hat{y}_k \; \forall \; k$ before raising them to the third power (element-wise) and computing the cosine similarity. The third power is used to make the loss more sensitive to highly activating inputs. And we minimize the negative similarity which is equivalent to maximizing the similarity.

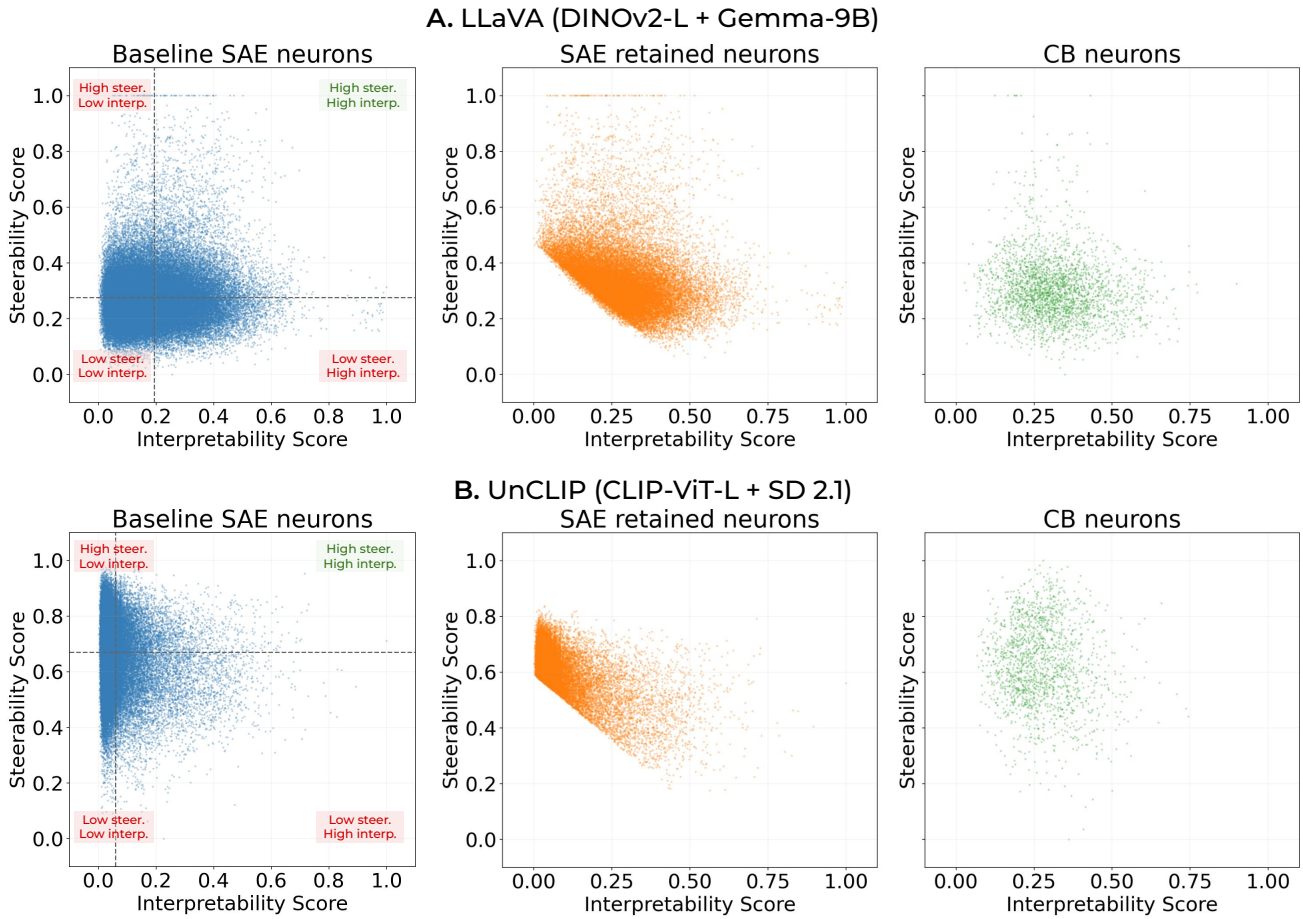

Figure 8. We analyze the interpretability and steerability of SAE and CB-SAE neurons for LLaVA with DINOv2 and Gemma2 as well as for UnCLIP with CLIP-ViT-L and Stable Diffusion 2.1. The dashed lines in the baseline SAE plots indicate the average scores along each axis.

C. Experiments

C.1. Experimental Setup

Downstream model details. We experiment with SAEs/CB-SAEs trained on vision encoders for downstream models like LLaVA [7] and UnCLIP [16]. LLaVA models are large vision-language models that take an image and a text prompt as input and output a text-based answer (Fig. 2A, main paper). Specifically, we used LLaVA-1.5-7B [8] which uses a CLIP-ViT-L-14-336 [15] vision encoder, a 2-layer MLP projector (not shown in Fig. 2A for simplicity), and an instruction-finetuned Vicuna-7B LLM [3]. We also use LLaVA-MORE [4] with DINOv2-Large [11] vision encoder, a 2-layer MLP projector, and an instruction-finetuned Gemma2-9B LLM [17]. On the other hand, UnCLIP is an image-to-image generative model that uses a CLIP-ViT-L [15] vision encoder and a finetuned Stable Diffusion 2.1 [16] as the image generator (Fig. 2B, main paper).

Miscellaneous details. We implement our CB-SAE in PyTorch [13] building on the SAE codebase from Pach et al. [12]. Following the baseline SAE training [12], we train the CB-SAE for 110k iterations with batch size 4096 and learning rate 2e−4 on a single 80GB Nvidia H100 GPU.

C.2. Interpretability vs Steerability in SAEs

We extend our analysis from Sec. 4 (main paper) on an SAE from LLaVA with CLIP image encoder to SAEs from LLaVA with DINOv2 image encoder and UnCLIP image-to-image generation model with CLIP image encoder in Fig. 8 (left). We report our observations (repeating those from Sec. 4):

- LLaVA (CLIP-ViT-L + Vicuna-7B, Fig. 3, main paper):
  - Low interpretability, low steerability: 36.26% (23763)
  - High interpretability, low steerability: 19.87% (13022)
  - Low interpretability, high steerability: 25.03% (16403)
  - High interpretability, high steerability: 18.84% (12348)
- LLaVA (DINOv2-L + Gemma-9B, Fig. 8A):
  - Low interpretability, low steerability: 33.07% (21675)
  - High interpretability, low steerability: 23.35% (15304)
  - Low interpretability, high steerability: 23.75% (15565)
  - High interpretability, high steerability: 19.82% (12992)

Table 3. Sensitivity to type of SAE.

SAE type	Interpretability			Steerability	
	CD	MS	Dead Neurons	Unit-Vec	White Image
Top-k SAE	**0.162**	**0.548**	52965 / 65536	**0.228**	**0.241**
Batch Top-k SAE	0.158	0.540	56899 / 65536	0.226	0.231
Matryoshka SAE	0.154	0.517	**4 / 65536**	0.198	0.203
Top-k CB-SAE	0.264	0.556	**0 / 32167**	**0.315**	**0.317**
Batch Top-k CB-SAE	**0.265**	**0.564**	0 / 32162	0.307	0.299
Matryoshka CB-SAE	0.244	0.556	4 / 32169	0.261	0.250

Table 4. Ablation to quantify the usefulness of SAE in CB-SAE.

	Interpretability	Steerability	
	CLIP-Dissect score	Unit-Vec	White Image
SAE [12]	0.154	0.198	0.203
CB-AE (w/o SAE)	**0.308**	0.238	0.232
CB-SAE (Ours)	0.244	**0.261**	**0.250**

- UnCLIP (CLIP-ViT-L + Stable Diffusion 2.1, Fig. 8B):
  - Low interpretability, low steerability: 30.84% (20209)
  - High interpretability, low steerability: 14.53% (9517)
  - Low interpretability, high steerability: 42.76% (28022)
  - High interpretability, high steerability: 11.88% (7788)

Note that the average steerability score for UnCLIP is higher than for LLaVA since the scores are computed in image embedding space and text embedding space respectively. Across both types of models, we consistently find that only a small portion of neurons (12-20%) are useful for both interpretability and steerability. And a majority of neurons (30-36%) are unsuitable for both interpreting new inputs and steering outputs.

We also show the retained SAE neurons and CB neurons in Fig. 8 (right) similar to Fig. 6 (main paper). We find CB neurons are similar to retained SAE neurons while being significantly better than the discarded SAE neurons (also shown quantitatively in Table 1, 2, main paper). We emphasize that CB neurons have to incorporate relatively more difficult concepts due to our concept set selection (Sec. 5.2, main paper) which excludes already discovered (and relatively easier to learn) concepts present in the retained SAE. Hence, it is more difficult for CB neurons to always outperform the retained SAE neurons.

Analysis using null baseline for steering scores. In the previous analysis of interpretability vs steerability in SAEs, we used the average (or mean) score as the threshold for both metrics. Another choice could be a null baseline for steerability, which would be the average steerability score of the original model's neurons without an SAE. We found this steerability score of CLIP layer 22 neurons (which are the input to LLaVA SAE/CB-SAE) to be 0.173, while the average steerability score in Fig. 3 (main paper) was 0.232. Using 0.173 as the steerability threshold, the proportions of neurons change as follows:

- LLaVA (CLIP-ViT-L + Vicuna-7B, Fig. 3, main paper):
  - Low interp., low steer.: 36.26% $\rightarrow$ 17.86%
  - High interp., low steer.: 19.87% $\rightarrow$ 9.58%
  - Low interp., high steer.: 25.03% $\rightarrow$ 43.42%
  - High interp., high steer.: 18.84% $\rightarrow$ 29.14%

However, our original claim of SAEs having low proportion (29%) of high utility neurons is still valid.

Traditional SAE metrics for 4 neuron classes. Reconstruction loss $||v - \hat{v}||_2^2$ (Sec. 3, main paper) does not directly involve SAE latent neurons. To isolate and quantify their impact, we individually zero each SAE neuron and compute the reconstruction loss change w.r.t. the original SAE. Then, the average loss changes over Interpretability/Steerability neuron classes (from Fig. 3) are: low/low (1.6e-5), high/low (1.4e-6), low/high (1.1e-5), high/high (3.9e-6). We observe low interpretability neurons contribute more to reconstruction, while steerability is relatively less influential.

C.3. Extended Analysis of our CB-SAE

Sensitivity to type of SAE. In Table 3, we evaluate the sensitivity of our CB-SAE to the type of pretrained SAE used. We consider Top-k and Batch Top-k SAEs in addition to the Matryoshka SAEs already compared in the main paper. We find that our CB-SAE can provide consistent improvements regardless of the type of pretrained SAE used. Interestingly, Top-k and Batch Top-k SAEs have better interpretability and steerability than Matryoshka SAEs, but also feature a very high number of dead neurons, *i.e.* SAE neurons which do not activate for any inputs. This makes sense since Matryoshka SAEs were proposed to overcome the dead neurons limitation. Further, our CB-SAE can also resolve the dead neurons problem by eliminating frequently activating SAE neurons.

Ablation of SAE from CB-SAE. In Table 4, we evaluate a CB-SAE model without using any SAE, *i.e.* a CB-AE [6] where all the user-defined concepts are directly used in the concept bottleneck. We find that CB-AE has higher interpretability score than CB-SAE since all concepts are now explicitly optimized for it. On the other hand, CB-SAE achieves better steerability since the retained SAE neurons

Table 5. **Sensitivity to choice of metrics for SAE pruning.**

Scores for SAE pruning	Reconstruction evaluation	Interpretability evaluation		Steerability evaluation	
	Zero-shot ImageNet Acc. (%)	CLIP-Dissect	Monosemanticity	Unit Vector	White Image
None (SAE baseline) [12]	**74.07**	0.154	0.517	0.198	0.203
Interpretability score only	73.39	<u>0.233</u>	**0.566**	0.216	0.220
Steerability score only	70.99	0.167	0.520	**0.288**	**0.269**
Both scores	<u>73.78</u>	**0.244**	<u>0.556</u>	<u>0.261</u>	<u>0.250</u>

Table 6. Sensitivity of interpretability evaluation with CLIP-Dissect to choice of CLIP-like model used.

CLIP-like model for evaluation		Interpretability Score	
Model	Architecture	SAE	CB-SAE (*Ours*)
CLIP [15]	ViT-B-16	0.198	**0.307**
CLIP [15]	ViT-L-14-336	0.154	**0.244**
SigLIP [19]	ViT-SO400M-14-384	0.189	**0.289**
SigLIP2 [18]	ViT-gopt-16-384	0.188	**0.290**
SigLIP2 [18]	ViT-SO400M-16-384	0.176	**0.272**
DFN [5]	ViT-H-14-378	0.220	**0.347**
PE-core [1]	BigG-14-448	0.207	**0.312**

have high steerability based on our analysis.

Sensitivity to scores used for SAE pruning. We extend our sensitivity analysis from Fig. 5A (main paper) in Table 5 to additionally include monosemanticity score [12] (interpretability evaluation) and zero-shot ImageNet-1k accuracy (reconstruction evaluation) when using the SAE/CB-SAE reconstructed latents. We observe that using either the interpretability score or both scores yields similar reconstruction as the baseline SAE, while steerability-based pruning leads to significantly worse reconstruction. Similarly, using either the interpretability score or both scores improves the monosemanticity significantly w.r.t. the baseline, while steerability-based pruning provides only a marginal gain over the baseline.

Sensitivity to CLIP model in interpretability evaluation. We evaluate the sensitivity of our interpretability evaluation with CLIP-Dissect by varying the CLIP-like model used, in Table 6. While our evaluation used a stronger CLIP-ViT-L-14-336 [15] model w.r.t. the smaller CLIP-ViT-B-16 used for training the CB-SAE, we now evaluate with even stronger models including SigLIP [19], SigLIP2 [18], Data Filtering Networks (DFN) [5] and Perception Encoder (PE) [1]. Across all CLIP-like models, our CB-SAE achieves consistent gains over the baseline SAE for LLaVA with CLIP-ViT-L encoder, validating that our choice of CLIP-like model for interpretability score does not affect our evaluation.

Sensitivity to k in σ_{cb}. In Fig. 9, we analyze the sensitivity of our CB-SAE to the choice of k in the top-k activation function used in the CB decoder. Here, we define reconstruction score as the zero-shot ImageNet-1k accuracy of CLIP when using SAE/CB-SAE reconstructed latents. We also report the white image steerability score of only the CB neurons to understand the impact of k on steerability. Note that we do not consider interpretability score here since σ_{cb} is only applied in the CB decoder while interpretability evaluation only considers the CB encoder, *i.e.* interpretability score does not change when varying k. We observe that reconstruction score improves as k increases, but it is already very close to the baseline even at $k = 3$ to $k = 5$. The steerability score first increases with k and then decreases for $k > 30$. This is because with higher k, steering might be less successful as the selected concept contends with many other concepts to be combined into the final reconstructed latent. On the other hand, if k is too low, then the reconstruction might not be good enough for the downstream model to produce the appropriate response. However, across all values of k, our CB-SAE is able to outperform the discarded SAE neurons while being worse than the retained SAE neurons. Hence, future work can develop more steerability-focused training objectives to further improve steerability.

Correlation between LLaVA and UnCLIP steerability. We used the same training method for CB-SAEs in both LLaVA and UnCLIP, but existing pretrained LLaVA and UnCLIP models do not use the same CLIP model (and even use different layers of CLIP and different number of tokens). However, we computed the Pearson correlation between the steerability scores of 1329 common CB concepts from LLaVA and UnCLIP CB-SAEs, which is 0.06 with a p-value 0.035. Hence, steerability seems to be largely downstream task/model-dependent.

C.4. Extended Qualitative Results

We provide qualitative examples of white image steering of UnCLIP with SAE/CB-SAE in Fig. 10. Similar to our results in Fig. 7 (main paper), we find steering CB-SAE neurons produces higher quality images while SAE neurons tend to produce more noisy images.

References

[1] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Abdul Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Shang-Wen Li, Piotr Dollar, and Christoph Feichtenhofer. Perception

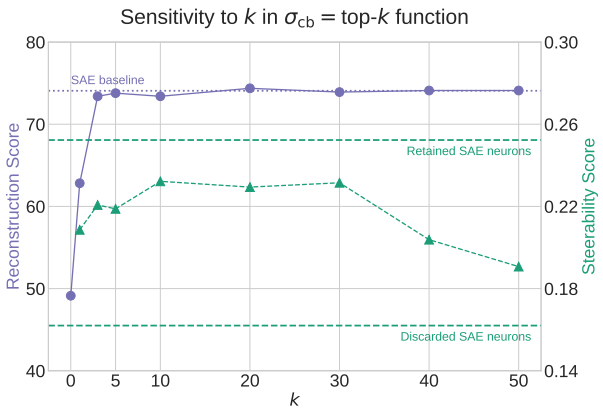

Figure 9. Sensitivity analysis of CB-SAE in LLaVA to k in top-k activation function used in the CB decoder. Steerability score here is computed only for CB neurons, reconstruction score is zero-shot accuracy when using SAE/CB-SAE reconstructions of CLIP latents on ImageNet-1k.

encoder: The best visual embeddings are not at the output of the network. In *NeurIPS*, 2025. 4

[2] David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Isaac Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. In *NeurIPS*, 2025. 1

[3] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality, 2023. 2

[4] Federico Cocchi, Nicholas Moratelli, Davide Caffagni, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. LLaVA-MORE: A Comparative Study of LLMs and Visual Backbones for Enhanced Visual Instruction Tuning. In *ICCVW*, 2025. 2

[5] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T Toshev, and Vaishaal Shankar. Data filtering networks. In *ICLR*, 2024. 4

[6] Akshay Kulkarni, Ge Yan, Chung-En Sun, Tuomas Oikarinen, and Tsui-Wei Weng. Interpretable generative models through post-hoc concept bottlenecks. In *CVPR*, 2025. 3

[7] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *NeurIPS*, 2023. 2

[8] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 2

[9] Tuomas Oikarinen and Tsui-Wei Weng. CLIP-Dissect: Automatic description of neuron representations in deep vision networks. In *ICLR*, 2023. 1

[10] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *ICLR*, 2023. 1

[11] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 2

[12] Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. In *NeurIPS*, 2025. 2, 3, 4

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 2

[14] Gonçalo Paulo, Stepan Shabalin, and Nora Belrose. Transcoders beat sparse autoencoders for interpretability. *arXiv preprint arXiv:2501.18823*, 2025. 1

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 4

[16] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2

[17] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving Open Language Models at a Practical Size. *arXiv preprint arXiv:2408.00118*, 2024. 2

[18] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features. *arXiv preprint arXiv:2502.14786*, 2025. 4

[19] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *ICCV*, 2023. 4

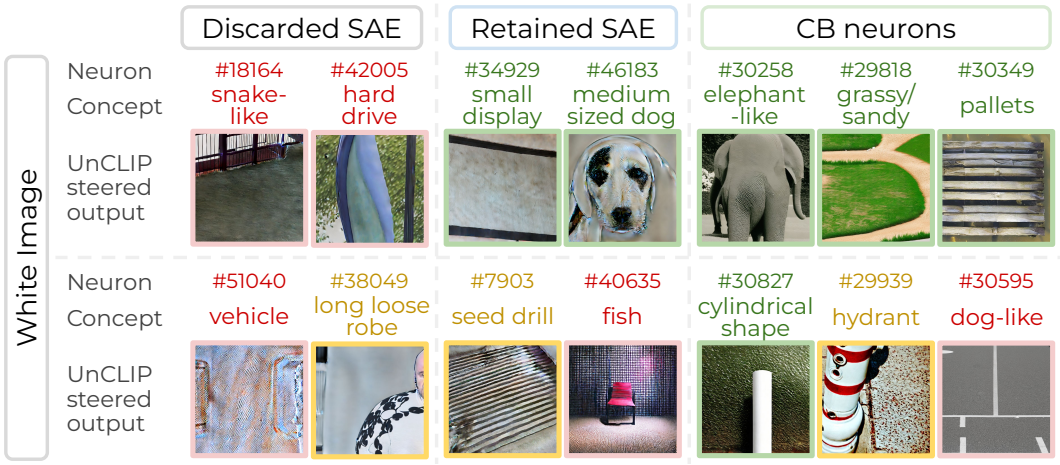

Figure 10. Qualitative examples of steering UnCLIP. Green indicates successful steering, yellow indicates partial success, and red indicates failure cases.