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

Abstract

Sparse autoencoders (SAEs) promise a unified approach for mechanistic interpretability, concept discovery, and model steering in LLMs and LVLMs. However, realizing this potential requires learned features to be both interpretable and steerable. To that end, we introduce two new computationally inexpensive interpretability and steerability metrics, and conduct a systematic analysis on LVLMs. We find that (i) a majority of SAE neurons exhibit either low interpretability or low steerability or both, rendering them ineffective for downstream use; and (ii) user-desired concepts are often absent in the learned dictionary, thus limiting their practical utility. To address these limitations, we propose Concept Bottleneck Sparse Autoencoders (CB-SAE)—a novel post-hoc framework that prunes low-utility neurons and adds a lightweight concept bottleneck aligned to a user-defined concept set. The resulting CB-SAE improves interpretability (+32.1%) and steerability (+14.5%) across LVLMs and image generation tasks. We will make our code and model weights available.

1. Introduction

Sparse autoencoders (SAEs) [3, 15, 34] have emerged as a foundational tool for mechanistic interpretability, mapping dense polysemantic activations into sparse monosemantic latents in large language models (LLMs) [19], vision models [37], and large vision-language models (LVLMs) [29]. However, realizing this promise requires SAE features to be semantically meaningful and causally effective, *i.e. interpretable* and *steerable* respectively.

Prior work [1, 43] shows interpretability does not guarantee steerability in LLM SAEs, but its implications in vision encoders remain unexplored. Empirically, we find only ∼19% SAE neurons exhibit both high interpretability and steerability. Further, despite large dictionary sizes (∼65k), SAEs fail to represent 27-45% of user-defined concepts. Hence, we identify two key limitations of SAEs: (i) a large proportion of low utility SAE neurons, and (ii) incomplete coverage of semantically meaningful concepts.

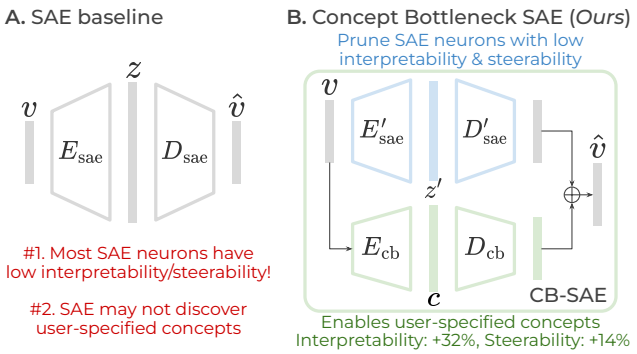

Figure 1. **A.** Majority of SAE neurons have low interpretability/steerability, with no guarantee of discovering user-specified concepts. **B.** Our CB-SAE addresses both limitations by pruning low utility SAE neurons, replacing them with a user-specified concept bottleneck that improves interpretability and steerability.

To overcome these limitations, we propose Concept Bottleneck Sparse Autoencoders (CB-SAE) – combining the unsupervised discovery capabilities of SAEs with the controllability of concept bottlenecks [16, 27, 36, 47]. We begin by pruning SAE features that lack interpretability and steerability, and then augment the resulting latent space with a lightweight CB autoencoder [17], trained to align with a user-specified concept set (Fig. 1B).

We evaluate CB-SAE on two challenging downstream tasks: controlled text generation via LVLMs like LLaVA [9, 20] and controlled image synthesis using UnCLIP [35]. CB-SAE consistently outperforms standard SAEs (+32.1% in interpretability and +14.5% in steerability) across all models and metrics. To our knowledge, CB-SAE is the first framework to unify sparse autoencoders with concept bottleneck models, enabling robust interpretation and control of vision representations across modalities and architectures.

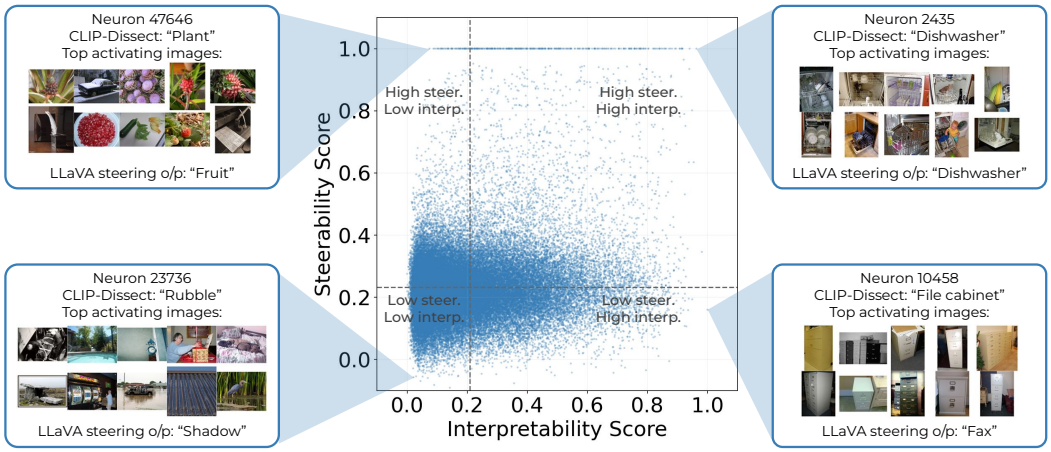

Figure 2. We analyze the interpretability and steerability of 65,536 neurons of an SAE trained for a CLIP image encoder. We also visualize the CLIP-Dissect assigned concept, top-activating images, and LLaVA steering outputs for some characteristic neurons. The dashed lines indicate the average scores along each axis, and we observe that most SAE neurons have either low interpretability, low steerability, or both.

2. Interpretability vs Steerability in SAEs

Measuring SAE interpretability. We assign each SAE neuron j to a concept c_j using CLIP-Dissect [26], which matches neurons to the closest concept in a user-specified concept set $\mathcal{C}$ via a vision-language model like CLIP [33]. Hence, we measure interpretability as the maximum similarity score from CLIP-Dissect averaged across all SAE neurons (see Appendix D for more details).

Measuring SAE steerability. Steerability measures how well overwriting an SAE neuron j's activation causally shifts model towards its concept c_j. Since our base model is a vision encoder, we use a downstream LLaVA [20] model to evaluate interventions. Following [29], we use a white image with the prompt *"What is shown in this image? Use exactly one word!"*, overwrite neuron j's activation to α across all tokens, and measure the cosine similarity between the steered LLaVA output and c_j in sentence-transformer embedding space. Unlike [29], our metric is grounded in CLIP-Dissect concepts instead of image-space similarity.

Expt. 1: Are all SAE neurons interpretable & steerable? Setup. We analyze a Matryoshka Batch Top-k SAE trained on CLIP-ViT-L/14-336 activations following [29], using the Broden concept set [2] ($|\mathcal{C}|$=1197) for CLIP-Dissect. Results with other models are presented in Appendix E.2.

Observations. Fig. 2 shows interpretability and steerability scores for 65,536 SAE neurons, revealing four quadrants. They are split based on interpretability/steerability: low/low (36.26%), high/low (19.87%), low/high (25.03%), and high/high (18.84%). This indicates that over 80% of neurons are unsuitable for downstream use.

Expt. 2: Can SAEs represent all user-specified concepts? Although the SAE contains 65,536 neurons—far exceeding the size of standard concept sets—its ability to represent concepts varies considerably with the diversity and complexity of the set. Using CLIP-Dissect, we evaluate the coverage of unique concepts across multiple concept sets. Concept coverage ranges from 96.3% on the smaller Broden set [2] to just 28.0% on 20k English words [27]. Notably, the SAE misses 27-45% of ImageNet-related concepts from VLG-CBM [36] and DECIDER [38] despite being trained on ImageNet.

3. Our Approach: CB-SAE

We propose a novel concept bottleneck sparse autoencoder (CB-SAE) based on our analysis to address two limitations of sparse autoencoders namely low interpretability/steerability and the lack of support for user-specified concepts.

3.1. Pruning SAE neurons

Step 1 (Fig. 3) We begin with training an SAE on layer l activations from the vision model f (Appendix B).

Step 2 (Fig. 3) As in our analysis experiments, we compute interpretability and steerability scores for each sparse neuron in the trained SAE denoted by $I \in [0,1]^\omega$ and $S \in [0,1]^\omega$.

Step 3 (Fig. 3) We prune the SAE weights $E_{\text{sae}}, D_{\text{sae}}$ to remove the M least interpretable and steerable SAE neurons as they are unsuitable for downstream applications. Concretely, the set of M SAE neurons to be pruned is $\mathcal{P} = \{m \mid I_m + S_m < \tau,\ m \in [\omega]\}$ where τ is the threshold that determines $|\mathcal{P}| = M$ and $[\omega] = \{1, 2, \cdots, \omega\}$. We prune $\mathcal{P}$ by deleting corresponding rows/columns in E_{sae} and D_{sae} respectively, keeping the bias $b \in \mathbb{R}^d$ unchanged.

$$E'_{\text{sae}} = E_{\text{sae}}[[\omega] \setminus \mathcal{P}, :] \quad (1)$$

$$D'_{\text{sae}} = D_{\text{sae}}[:, [\omega] \setminus \mathcal{P}] \quad (2)$$

Pruning degrades reconstruction fidelity, which we recover by introducing a concept bottleneck below.

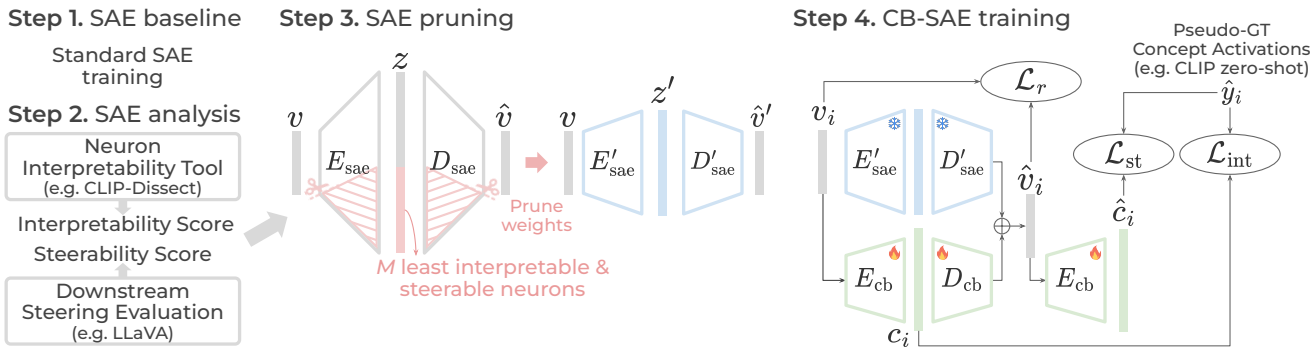

Figure 3. **Pipeline for CB-SAE**. **Step 1.** A baseline SAE is trained and **Step 2.** evaluated with CLIP-Dissect and downstream steering to obtain interpretability and steerability scores per SAE neuron. **Step 3.** M least interpretable and steerable neurons are pruned by deleting the corresponding SAE weights. **Step 4.** We train CB-SAE with frozen, pruned SAE weights to recover the reconstruction ability lost by pruning using $\mathcal{L}_r$, incorporate the user-specified concept set with $\mathcal{L}_{\text{int}}$, and promote steerability with a cyclic reconstruction loss $\mathcal{L}_{\text{st}}$.

3.2. Training CB-SAE

Step 4 (Fig. 3) We introduce a linear concept bottleneck autoencoder [17], $E_{\text{cb}} \in \mathbb{R}^{|\mathcal{C}| \times d}$, $D_{\text{cb}} \in \mathbb{R}^{d \times |\mathcal{C}|}$ alongside the retained SAE, with a top-k activation function σ_{cb} for sparsity, where $\mathcal{C}$ is a pre-defined concept set. Reconstruction of $\hat{v}'$ is done by,

$$z' = \sigma_{\text{sae}}(E'_{\text{sae}}(v - b)) \tag{3}$$

$$c = E_{\text{cb}}(v - b) \tag{4}$$

$$\hat{v}' = D'_{\text{sae}} z' + b + D_{\text{cb}}\sigma_{\text{cb}}(c) \tag{5}$$

To avoid redundancy, the CB concept set excludes concepts already captured by the retained SAE: $\mathcal{C} = \mathcal{C}_{\text{user}} \setminus \mathcal{C}_{\text{rsae}}$, where $\mathcal{C}_{\text{rsae}}$ are concepts present in the retained SAE.

Training Objectives. Retained SAE weights are frozen to preserve their reconstruction, interpretability, and steerability, and only $E_{\text{cb}}, D_{\text{cb}}$ are trained.

Objective A: Reconstruction $\mathcal{L}_{\mathbf{r}}$ (Fig. 3, Step 4). We optimize the mean-squared error $\mathcal{L}_r$ between v and $\hat{v}'$ from Eq. (5), with top-k sparsity instead of ℓ_1 regularization,

$$\min_{E_{\text{cb}}, D_{\text{cb}}} [\mathcal{L}_r(v, \hat{v}')] \tag{6}$$

Objective B: Interpretability $\mathcal{L}_{\text{int}}$ (Fig. 3, Step 4). A CLIP zero-shot classifier [33] M produces pseudo-ground-truth concept activations [17, 27] $\hat{y} = \mathcal{M}(x, \mathcal{C}) \in \mathbb{R}^{|\mathcal{C}|}$. We use a cosine-cubed similarity loss [27] $\mathcal{L}_{\text{int}}$ between $c = E_{\text{cb}}(v)$ and $\hat{y}$. Note that σ_{cb} is applied only after the decoder in Eq. 5, allowing E_{cb} to interpret all concepts.

$$\min_{E_{\text{cb}}} [\mathcal{L}_{\text{int}}(c, \hat{y})] \tag{7}$$

Objective C: Steerability $\mathcal{L}_{\text{st}}$ (Fig. 3, Step 4). We propose a task-agnostic cyclic loss: reconstruction $\hat{v}'$ is passed back through E_{cb} to get $\hat{c} = E_{\text{cb}}(\hat{v}' - b)$, and we optimize the same loss as Eq. (7) between $\hat{c}$ and pseudo-GT concepts $\hat{y}$. Only D_{cb} is updated to preserve the interpretability of E_{cb}.

$$\min_{D_{\text{cb}}} [\mathcal{L}_{\text{st}}(\hat{c}, \hat{y})] \tag{8}$$

4. Experiments

We extensively evaluate our CB-SAE w.r.t. interpretability and steerability on two downstream tasks, (image,text)-to-text generation and image-to-image generation.

Baseline SAE and CB-SAE. We follow [29], retaining $\omega - M = 30$k SAE neurons after pruning, with $k = 5$ in σ_{cb}, and use the VLG-CBM ImageNet concept set [27, 36] for CB neurons (full details in Appendix E.1).

Evaluation Metrics. We report CLIP-Dissect and monosemanticity [29] scores for interpretability (using a stronger CLIP-ViT-L/14 for evaluation than ViT-B/16 used for training). Our steerability score is computed in two settings: **Unit Vector** (neuron j set to $\alpha = 50$, all others to 0) and **White Image** (neuron j set to $\alpha = 50$, all others to neuron values for white image as input). For text outputs (LLaVA), steered text is compared to the CLIP-Dissect concept in sentence-transformer space. For image outputs (UnCLIP), we use DINOv2 similarity to top-16 activating images [29].

4.1. Quantitative Comparison

Table 1 shows CB-SAE consistently outperforms SAE baseline across all models and tasks, with gains of +32.1% intepretability and +14.5% steerability. To our knowledge, we are the first to show an SAE (and CB-SAE) trained with the same method can steer different downstream tasks.

4.2. Analysis of our CB-SAE

Effect of CB neurons. Table 2 shows CB neurons achieve the highest interpretability but lower steerability than re-

Table 1. **Interpretability and Steerability Evaluation with LLaVA and UnCLIP.** All four metrics are in 0-1 range (higher is better), CD indicates CLIP-Dissect score and MS indicates monosemanticity score.

Downstream Task	Steered Model	Method	Interpretability		Steerability	
			CD	MS	Unit-Vec	White Image
Image + Text → Text Generation	LLaVA-1.5-7B [21] (CLIP-ViT-L + Vicuna-7B)	SAE [29]	0.154	0.517	0.198	0.203
		CB-SAE (*Ours*)	**0.244**	**0.556**	**0.261**	**0.250**
	LLaVA-MORE [9] (DINOv2-L + Gemma2-9B)	SAE [29]	0.194	0.553	0.179	0.177
		CB-SAE (*Ours*)	**0.291**	**0.598**	**0.192**	**0.189**
Image → Image Generation	UnCLIP [35] (CLIP-ViT-L + SD-2.1)	SAE [29]	0.058	0.540	0.642	0.654
		CB-SAE (*Ours*)	**0.092**	**0.594**	**0.659**	**0.664**

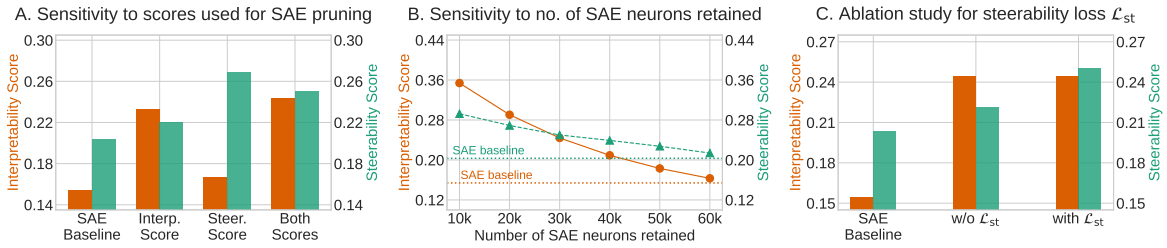

Figure 4. **A.** Sensitivity of CB-SAE to the choice of scores used for SAE pruning. **B.** Sensitivity of CB-SAE to the number of SAE neurons retained. **C.** Ablation study for our proposed steerability objective $\mathcal{L}_{st}$ from Eq. (8).

Table 2. Evaluating interpretability and steerability of discarded SAE neurons, retained SAE neurons, and CB neurons separately.

Set of Neurons	Interpretability	Steerability	
	CLIP-Dissect	Unit-Vec	White Image
All SAE neurons	0.154	0.198	0.203
Discarded SAE neurons	0.084	0.144	0.162
Retained SAE neurons	0.238	**0.263**	**0.252**
CB neurons	**0.323**	0.231	0.219
All CB-SAE neurons	0.244	0.261	0.250

tained SAE neurons. The latter is expected since steerability was used as a pruning criterion.

Sensitivity to scores used for SAE pruning. Using both interpretability & steerability scores gives balanced performance; using only either sacrifices the other (Fig. 4A).

Sensitivity to no. of SAE neurons retained. Retaining fewer SAE neurons improves scores but hurts reconstruction; $\omega - M = 30$k gives the best trade-off (Fig. 4B).

Ablation study for steerability loss $\mathcal{L}_{st}$. Our steerability loss $\mathcal{L}_{st}$ improves steerability by 2.9% without affecting interpretability, validating its usefulness (Fig. 4C).

Qualitative examples of steering (Fig. 5). CB and retained SAE neurons consistently produce coherent outputs, while discarded neurons fail. CB neurons produce notably cleaner images in UnCLIP due to explicit concept supervision (additional examples in Appendix E.4).

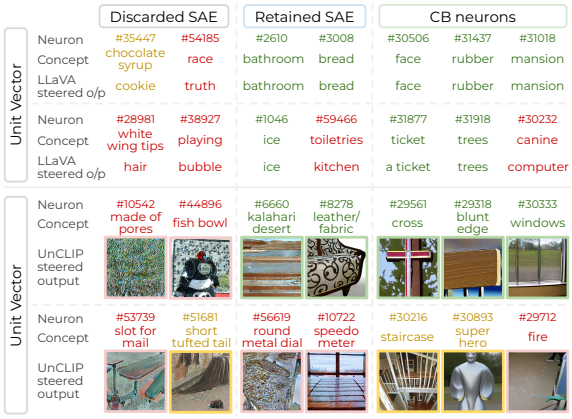

Figure 5. Qualitative examples of steering UnCLIP and LLaVA. Green indicates successful steering, yellow indicates partial success, and red indicates failure cases. See Appendix for more results.

5. Conclusion

In this work, we made the first attempt to unify two complementary paradigms - SAEs for unsupervised concept discovery and CBMs for interpretable control - into a single unified framework, CB-SAE. Motivated by insights derived from our comprehensive analysis of SAEs in LVLMs, we first pruned low-utility neurons and their corresponding weights in the SAE. We then introduced a lightweight CB module trained alongside the frozen, retained SAE using three principled objectives.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344, Lawrence Livermore National Security, LLC. and was supported by the LLNL-LDRD Program under Project No. 25-SI-001 and DOE ECRP 51917/SCW1885. LLNL-CONF-2013863. A. Kulkarni and T-W. Weng are also partially supported by National Science Foundation under Grant No. 2313105, 2430539, Hellman Fellowship, ARL Award, and Intel Rising Star Faculty Award.

References

[1] Dana Arad, Aaron Mueller, and Yonatan Belinkov. Saes are good for steering–if you select the right features. *arXiv preprint arXiv:2505.20063*, 2025. 1, 7

[2] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, 2017. 2

[3] Joseph Bloom, Curt Tigges, Anthony Duong, and David Chanin. SAELens: An open-source library for training and analyzing sparse autoencoders, 2024. 1

[4] 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 encoder: The best visual embeddings are not at the output of the network. In *NeurIPS*, 2025. 10

[5] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. 7

[6] Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. In *NeurIPSW*, 2024. 7

[7] Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. In *ICML*, 2025. 7

[8] 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. 8

[9] 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. 1, 4, 8

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 8

[11] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T Toshev, and Vaishaal Shankar. Data filtering networks. In *ICLR*, 2024. 10

[12] Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *ICLR*, 2025. 7

[13] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *ICLR*, 2024. 7

[14] Aya Abdelsalam Ismail, Julius Adebayo, Hector Corrada Bravo, Stephen Ra, and Kyunghyun Cho. Concept bottleneck generative models. In *ICLR*, 2024. 7

[15] Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *ICML*, 2025. 1, 7

[16] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *ICML*, 2020. 1, 7

[17] Akshay Kulkarni, Ge Yan, Chung-En Sun, Tuomas Oikarinen, and Tsui-Wei Weng. Interpretable generative models through post-hoc concept bottlenecks. In *CVPR*, 2025. 1, 3, 7, 10

[18] Jogendra Nath Kundu, Maharshi Gor, and R Venkatesh Babu. Bihmp-gan: Bidirectional 3d human motion prediction gan. In *AAAI*, 2019. 8

[19] Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, 2024. 1, 7

[20] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *NeurIPS*, 2023. 1, 2, 8

[21] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 4, 8

[22] Hantao Lou, Changye Li, Jiaming Ji, and Yaodong Yang. SAE-V: Interpreting multimodal models for enhanced alignment. In *ICML*, 2025. 7

[23] Alireza Makhzani and Brendan Frey. k-Sparse autoencoders. In *ICLR*, 2014. 7

[24] Ali Nasiri-Sarvi, Hassan Rivaz, and Mahdi S Hosseini. Sparc: Concept-aligned sparse autoencoders for cross-model and cross-modal interpretability. *arXiv preprint arXiv:2507.06265*, 2025. 7

[25] Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. Towards interpreting visual information processing in vision-language models. In *ICLR*, 2025. 7

[26] Tuomas Oikarinen and Tsui-Wei Weng. CLIP-Dissect: Automatic description of neuron representations in deep vision networks. In *ICLR*, 2023. 2, 7, 8

[27] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *ICLR*, 2023. 1, 2, 3, 7, 8

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *TMLR*, 2024. 8, 9

[29] Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. In *NeurIPS*, 2025. 1, 2, 3, 4, 7, 8, 9, 10, 11

[30] Isabel Papadimitriou, Huangyuan Su, Thomas Fel, Sham Kakade, and Stephanie Gil. Interpreting the linear structure of vision-language model embedding spaces. In *COLM*, 2025. 7

[31] 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. 9

[32] Gonçalo Paulo, Stepan Shabalin, and Nora Belrose. Transcoders beat sparse autoencoders for interpretability. *arXiv preprint arXiv:2501.18823*, 2025. 7

[33] 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. 2, 3, 7, 8, 10

[34] Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024. 1, 7

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 4, 8, 9

[36] Divyansh Srivastava, Ge Yan, and Lily Weng. VLG-CBM: Training concept bottleneck models with vision-language guidance. In *NeurIPS*, 2024. 1, 2, 3, 7, 8

[37] Samuel Stevens, Wei-Lun Chao, Tanya Berger-Wolf, and Yu Su. Sparse autoencoders for scientifically rigorous interpretation of vision models. *arXiv preprint arXiv:2502.06755*, 2025. 1, 7

[38] Rakshith Subramanyam, Kowshik Thopalli, Vivek Narayanaswamy, and Jayaraman J Thiagarajan. Decider: Leveraging foundation model priors for improved model failure detection and explanation. In *ECCV*, 2024. 2

[39] Chung-En Sun, Tuomas Oikarinen, Berk Ustun, and Tsui-Wei Weng. Concept bottleneck large language models. In *ICLR*, 2025. 7

[40] 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. 8

[41] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features. *arXiv preprint arXiv:2502.14786*, 2025. 10

[42] Constantin Venhoff, Ashkan Khakzar, Sonia Joseph, Philip Torr, and Neel Nanda. How visual representations map to language feature space in multimodal llms. In *CVPRW*, 2025. 7

[43] Xu Wang, Yan Hu, Benyou Wang, and Difan Zou. Does higher interpretability imply better utility? a pairwise analysis on sparse autoencoders. In *NeurIPSW*, 2025. 1

[44] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition. In *ICCV*, 2023. 7

[45] Minglai Yang, Xinyu Guo, Mihai Surdeanu, and Liangming Pan. Alignsae: Concept-aligned sparse autoencoders. *arXiv preprint arXiv:2512.02004*, 2025. 7

[46] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *CVPR*, 2023. 7

[47] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR*, 2023. 1, 7

[48] Vladimir Zaigrajew, Hubert Baniecki, and Przemyslaw Biecek. Interpreting CLIP with hierarchical sparse autoencoders. In *ICML*, 2025. 7, 8

[49] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *ICCV*, 2023. 10

Appendix

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 [32] with user-specified concept control methods constitute our future work.

B. Extended Background

SAE preliminaries. Let $v = f_l(x) \in \mathbb{R}^d$ denote the dense activations from layer l of a deep pre-trained vision model (e.g., CLIP image encoder [33]) f for an input image $x \in \mathbb{X}$. Here d denotes the activation dimension and $\mathbb{X}$ corresponds to the space of images. SAEs decomposes the polysemantic activations v into sparse, overcomplete latent representations $z \in \mathbb{R}^\omega$ ($\frac{\omega}{d} >> 1$) with the aim of associating every unit in z to distinct, interpretable concepts. Here $\frac{\omega}{d}$ corresponds to the expansion factor of the SAE [12]. Formally, an SAE is parameterized by a linear encoder $E_{\text{sae}} \in \mathbb{R}^{\omega \times d}$, a linear decoder $D_{\text{sae}} \in \mathbb{R}^{d \times \omega}$, a shared bias term $b \in \mathbb{R}^d$, and a non-linear activation function $\sigma_{\text{sae}} : \mathbb{R}^\omega \to \mathbb{R}^\omega$:

$$z = \sigma_{\text{sae}}(E_{\text{sae}}(v - b)) \tag{9}$$

$$\hat{v} = D_{\text{sae}} z + b \tag{10}$$

The SAE training objective is given by $\mathcal{L}_r = ||v - \hat{v}||_2^2 + \lambda||z||_1$, where $\lambda \geq 0$ balances reconstruction fidelity and sparsity, where $\hat{v}$ represents the SAE reconstruction. In addition to standard ℓ_1 regularization, sparsity can be enforced directly via the activation function $\sigma_{\text{sae}}(\cdot)$, such as top-$k$ [12], batch top-k [6], or ReLU with a learnable threshold [19, 34].

C. Related Work

Sparse Autoencoders. SAEs aim to discover interpretable features in neural networks by learning overcomplete decompositions of activations [23]. Recent work [5, 13] showed SAEs can decompose LLM representations into monosemantic features. Various architectural innovations improved SAEs, like Batch-Top-k sparsity [6], JumpReLU [34], and Matryoshka SAEs [7] with multi-level feature hierarchies. Large-scale efforts trained LLM SAEs across multiple layers and models [12, 19], with systematic benchmarks [15]. However, we uncover two key limitations of SAEs: their unsupervised training does not guarantee the discovery of user-desired concepts, and many SAE neurons exhibit low interpretability or utility in downstream steering [1].

Concept Bottleneck Models. CBMs [16, 47] provide a framework for building interpretable models by constraining predictions through a human-understandable concept layer, enabling both interpretation and steering. This approach has been extended to label-free settings [27], enhanced with vision-language guidance [36, 44, 46], applied to image generative models [14, 17] as well as LLMs [39]. Our work bridges SAEs and CBMs into our novel CB-SAE, combining the expressiveness of overcomplete feature decomposition with user-specified concepts, steerability, and interpretability of concept-guided learning. A concurrent work, AlignSAE [45], independently devised a similar approach to introduce supervised concepts in SAEs. They attempt to disentangle the supervised concepts from the unsupervised SAE neurons with an orthogonality loss, while our approach explicitly prunes the low utility SAE neurons and only introduces the supervised concepts absent from the retained SAE neurons. Further, AlignSAE focuses on text-based LLM SAEs while we focus on vision SAEs for multimodal LLMs and image-to-image generative models.

SAEs for Vision and Vision-Language Models. Recent work showed that SAEs can learn interpretable, monosemantic features in vision models [37] as well as vision-language models [29, 48]. Another line of work [25, 30, 42] investigated how visual information maps to language feature spaces via SAEs for cross-modal interpretability [22, 24]. However, these approaches typically neither address the challenges of ensuring discovered features are both interpretable and steerable, nor do they guarantee the discovery of user-specified concepts. Our CB-SAE addresses both limitations through post-hoc pruning and concept-bottleneck training.

D. Implementation Details

Intepretability Score. We define our CLIP-Dissect-based interpretability score as the similarity score obtained from CLIP-Dissect, averaged across all SAE/CB-SAE neurons.

CLIP-Dissect [26]. 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

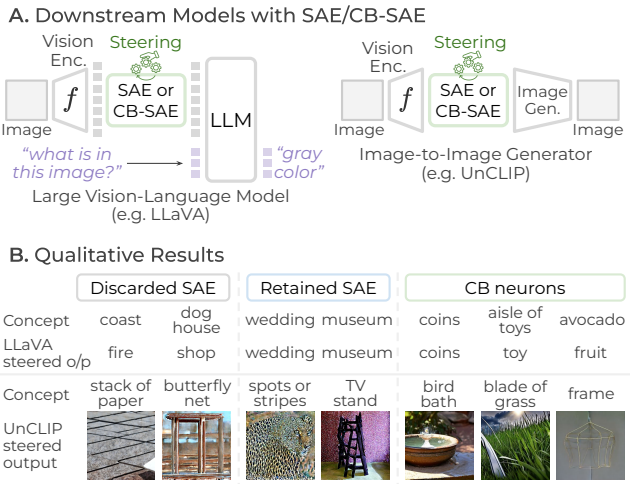

Figure 6. **A.** Our CB-SAE and baseline SAE can steer multiple downstream models like large vision-language models (LLaVA [21]) or image generative models (UnCLIP [35]). **B.** Examples of steering LLaVA and UnCLIP when using unit vector steering (zeroing out all SAE/CB-SAE neurons except the selected 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 [33] 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 [26]. Please refer to the original paper [26] for more details.

Cosine-cubed similarity loss [27] $\mathcal{L}_{\text{int}}$. 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,

$$\mathcal{L}_{\text{int}}(c, \hat{y}) = \sum_{k=1}^{|\mathcal{C}|} -\frac{c_k^3 \cdot \hat{y}_k^3}{\|c_k^3\|_2 \|\hat{y}_k^3\|_2} \tag{11}$$

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 [27], 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.

Instead of loss weighting hyperparameters, we train by alternately minimizing the objectives via separate Adam optimizers which adaptively scale weight updates [18].

E. Experiments

E.1. Experimental Setup

Baseline SAE and CB-SAE. We follow Pach et al. [29] and train a Matryoshka Batch Top-k SAE [48] with expansion factor $\frac{\omega}{d} = 64$ as the baseline SAE on the ImageNet-1k [10] dataset. Our CB-SAE is also trained on the same intermediate activations as the baseline SAE for a fair comparison. We retain $\omega - M = 30$k neurons in the SAE pruning and use a top-k function as σ_{cb} with $k = 5$ in our CB-SAE. We use the VLG-CBM ImageNet concept set [27, 36] for the CB neurons. In our training, we use a CLIP-ViT-B/16 [33] model for obtaining the pseudo-ground-truth concept activations.

Downstream model details. We experiment with SAEs/CB-SAEs trained on vision encoders for downstream models like LLaVA [20] and UnCLIP [35]. 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 [21] which uses a CLIP-ViT-L-14-336 [33] vision encoder, a 2-layer MLP projector (not shown in Fig. 2A for simplicity), and an instruction-finetuned Vicuna-7B LLM [8]. We also use LLaVA-MORE [9] with DINOv2-Large [28] vision encoder, a 2-layer MLP projector, and an instruction-finetuned Gemma2-9B LLM [40]. On the other hand, UnCLIP is an image-to-image generative model that uses a CLIP-ViT-L [33] vision encoder and a finetuned Stable Diffusion 2.1 [35] as the image generator (Fig. 2B, main paper).

Evaluation Metrics. To evaluate interpretability, we use the CLIP-Dissect interpretability score introduced in Sec. 2 and the monosemanticity score from [29] using the ImageNet validation set. To ensure a fair evaluation, we use a stronger CLIP-ViT-L/14 model (w.r.t. smaller ViT-B/16 used for training CB-SAE). To evaluate steerability, we use our proposed steerability score (Sec. 2). Concretely, we evaluate the steerability of each CB/SAE neuron in two ways:

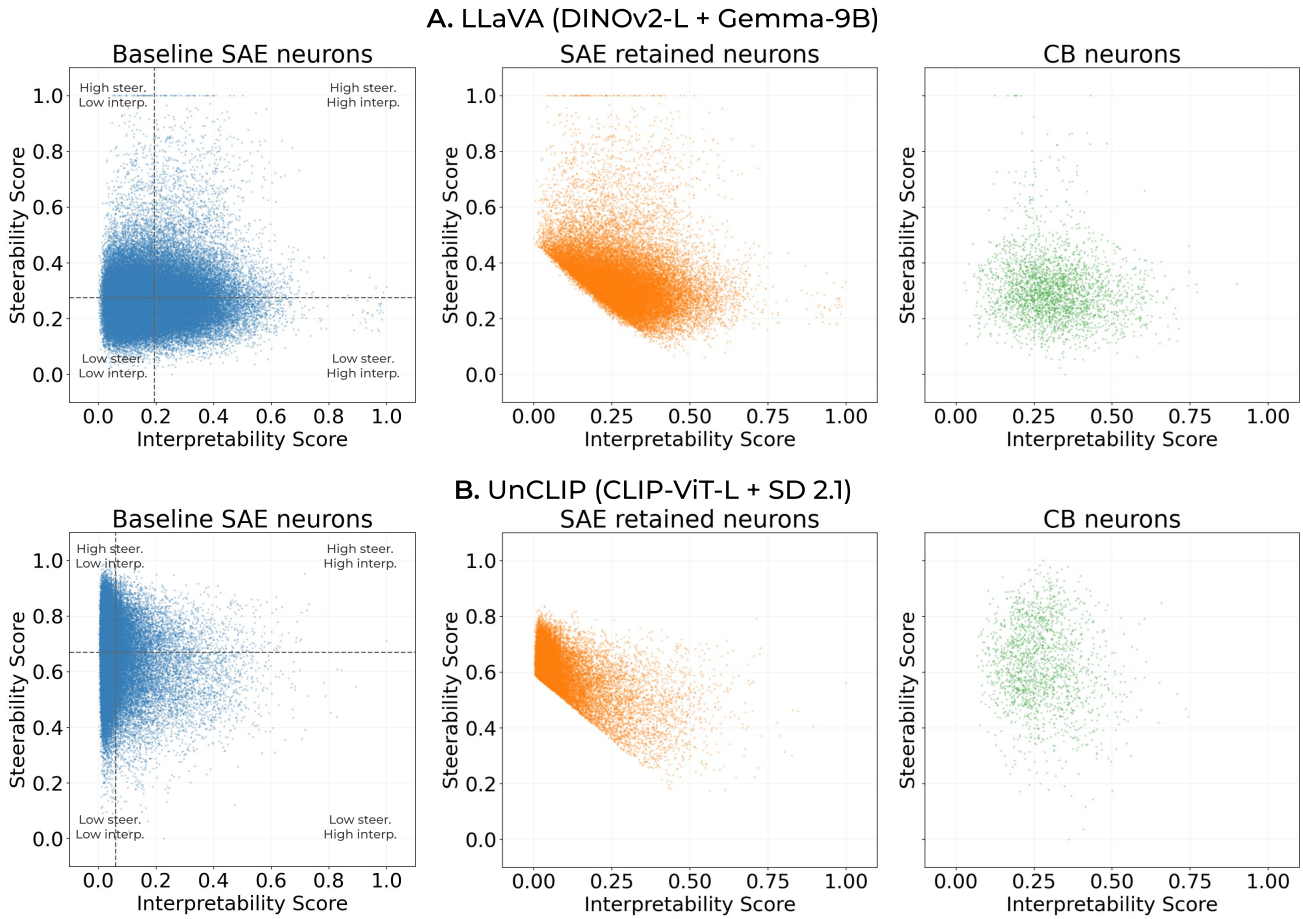

Figure 7. We analyze the interpretability and steerability of SAE and CB-SAE neurons for LLaVA with DINOv2 and Gemma2 as well as for UnCLIP with CLIP-ViT-L and Stable Diffusion 2.1. The dashed lines in the baseline SAE plots indicate the average scores along each axis.

- Unit Vector: The selected neuron is activated to a high value $\alpha = 50$ (as in [29]) & all other neurons are set to 0.
- White Image: The selected neuron is activated to a high value $\alpha = 50$ and all other neurons have the values predicted when using an empty white image (following [29]) as input, instead of 0 like in unit vector steering.

The interpretability and steerability scores of individual neurons are averaged to obtain the overall scores. For experiments where the steered output is text, we compare the similarity between the steered text and the CLIP-Dissect assigned concept for the selected neuron in a sentence transformer embedding space (as in Sec. 2). For experiments where steered output is an image, we compute the average similarity between the steered image and top-16 highly activating images for the selected neuron in the DINOv2 [28] embedding space. This is because the diffusion model being steered (UnCLIP [35]) may rarely return partially or completely noisy images after steering, which cannot not be properly evaluated with an image-text similarity score (*e.g.* CLIP) that expects clean images. All metrics are normalized in 0-1 range and higher values indicate better performance.

Miscellaneous details. We implement our CB-SAE in PyTorch [31] building on the SAE codebase from Pach et al. [29]. Following the baseline SAE training [29], we train the CB-SAE for 110k iterations with batch size 4096 and learning rate 2e−4 on a single 80GB Nvidia H100 GPU.

E.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. 7 (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. 7A):
  - Low interpretability, low steerability: 33.07% (21675)

Table 3. **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) [29]	**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 4. 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 [33]	ViT-B-16	0.198	**0.307**
CLIP [33]	ViT-L-14-336	0.154	**0.244**
SigLIP [49]	ViT-SO400M-14-384	0.189	**0.289**
SigLIP2 [41]	ViT-gopt-16-384	0.188	**0.290**
SigLIP2 [41]	ViT-SO400M-16-384	0.176	**0.272**
DFN [11]	ViT-H-14-378	0.220	**0.347**
PE-core [4]	BigG-14-448	0.207	**0.312**

  - High interpretability, low steerability: 23.35% (15304)
  - Low interpretability, high steerability: 23.75% (15565)
  - High interpretability, high steerability: 19.82% (12992)
- UnCLIP (CLIP-ViT-L + Stable Diffusion 2.1, Fig. 7B):
  - 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. 7 (right) and Fig. 8 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.

E.3. Extended Analysis of our CB-SAE

Sensitivity to type of SAE. In Table 5, 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 6, we evaluate a CB-SAE model without using any SAE, *i.e*. a CB-AE [17] 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 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 3 to additionally include monosemanticity score [29] (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 4. While our evaluation used a stronger CLIP-ViT-L-14-336 [33] 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 [49], SigLIP2 [41], Data Filtering Networks (DFN) [11] and Perception Encoder (PE) [4]. 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.

Table 5. 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 6. Ablation to quantify the usefulness of SAE in CB-SAE.

	Interpretability	Steerability	
	CLIP-Dissect score	Unit-Vec	White Image
SAE [29]	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**

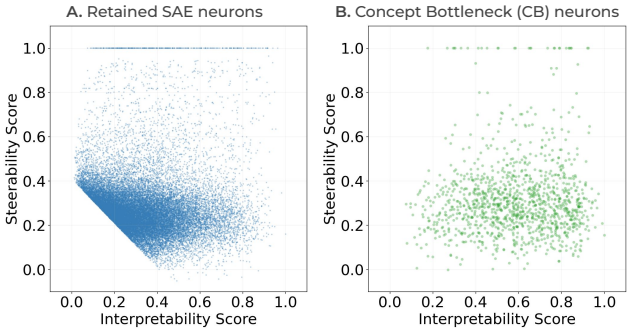

Figure 8. Visualizing the interpretability and steerability of retained SAE neurons and CB neurons for LLaVA with CLIP-ViT encoder, similar to Fig. 2.

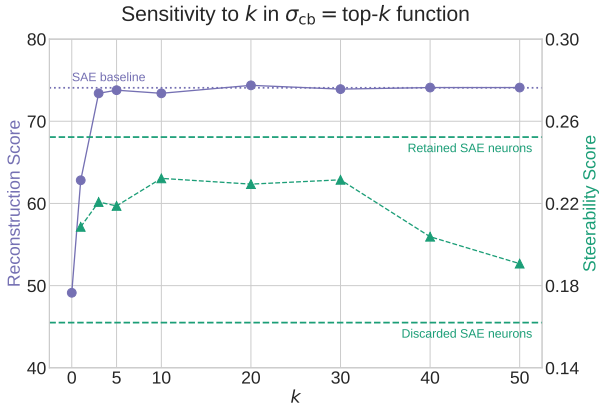

Figure 9. Sensitivity analysis of CB-SAE in LLaVA to k in top-k activation function used in the CB decoder. Steerability score here is computed only for CB neurons, reconstruction score is zero-shot accuracy when using SAE/CB-SAE reconstructions of CLIP latents on ImageNet-1k.

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.

E.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.

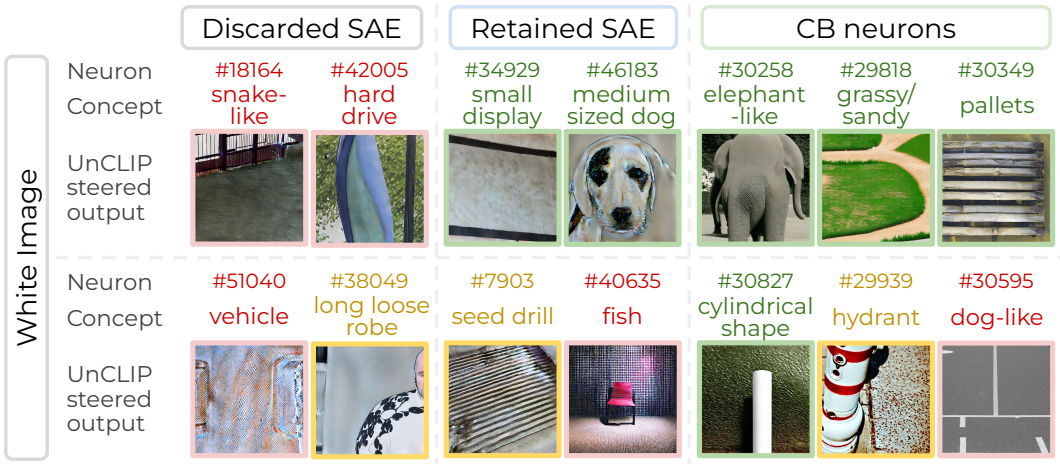

Figure 10. Qualitative examples of steering UnCLIP. Green indicates successful steering, yellow indicates partial success, and red indicates failure cases.