Improved Vision-Language Alignment via Text-Conditioned Image Embeddings using Sparse Autoencoders

Sweta Mahajan*,[1], Sukrut Rao*,[1], Jiahao Xie[1], Alexander Koller[1,2], Bernt Schiele[1]
[1]Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
[2]Department of Language Science and Technology, Saarland University, Saarbrücken, Germany
{sweta.mahajan,sukrut.rao,schiele}@mpi-inf.mpg.de

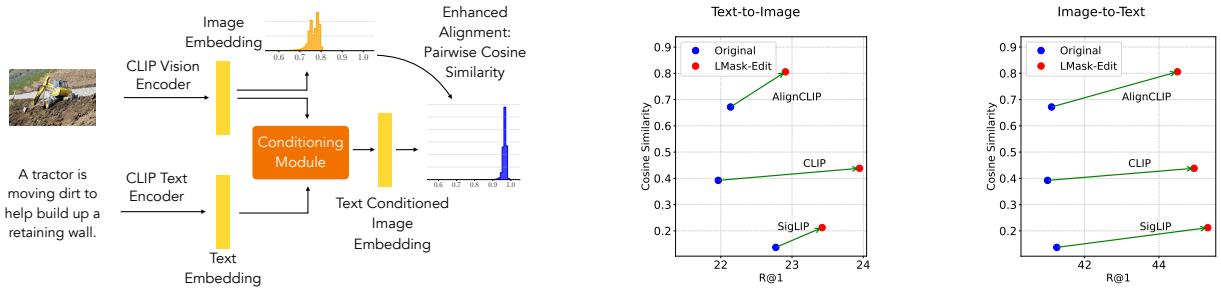


Fig. 1. **LMask-Edit: Editing image embeddings by conditioning on text.** *Left:* An overview of our approach. We train a conditioning module that learns to edit embeddings from the CLIP vision encoder conditioned on a text, improving cross-modal alignment. For details, see Fig. 2. *Right:* LMask-Edit helps improve cross-modal alignment as well as downstream retrieval performance. We report mean performance across datasets, for full results see Tab. 1 and Secs. C.3 and C.5.

Abstract

Vision-language models such as CLIP are highly useful for diverse tasks due to their shared image-text embedding space. Despite this, they often suffer from a modality gap where image and text embeddings are poorly aligned, affecting downstream performance. Recent work has shown that the modality gap can be attributed to an information imbalance between the two modalities. In this work, we propose LMask-Edit, a framework that explicitly models the information imbalance and addresses it by editing image embeddings conditioned on text. Specifically, we use sparse autoencoders to disentangle image embeddings and train a masking module to selectively reconstruct the embedding based on the text conditioning. Using a controlled setup, we show that LMask-Edit is effective at conditioning and improves cross-modal alignment. By applying LMask-Edit to CLIP models trained on natural images, we further achieve improved retrieval performance across coarse (MS COCO, Flickr) and fine-grained (IIW, DOCCI) benchmarks, as well robust retrieval on the RoCOCO benchmark, demonstrating its promise for improving learned representations.

1. Introduction

Vision-language models such as the CLIP family [12, 26, 34, 39] are trained using a contrastive loss to align images and text to a shared embedding space. Such models have shown impressive performance on a variety of tasks such as zero-/few-shot classification and cross-modal retrieval, and help encode images for text-to-image diffusion models [30] and large multimodal models (LMMs) [17]. However, the learnt embedding space suffers from a modality gap [15]—despite being trained for alignment, image and text embeddings lie in different regions of the embedding space, often leading to poor downstream performance [7, 15, 31].

Recent work [31] hypothesizes that this is caused by an *information imbalance*—the fact that images contain more information than their corresponding captions, which forces models to push their embeddings apart when minimizing the training loss—and shows evidence for this hypothesis via controlled experiments. Motivated by this, to reduce the imbalance, we propose to modify CLIP image embeddings to only capture what is described by its corresponding text. To do this, we use sparse autoencoders (SAEs) [1, 4] to decompose image embeddings into constituent concepts, and train a conditioning module that selects SAE latents to be used for reconstruction based on

*Equal contribution.

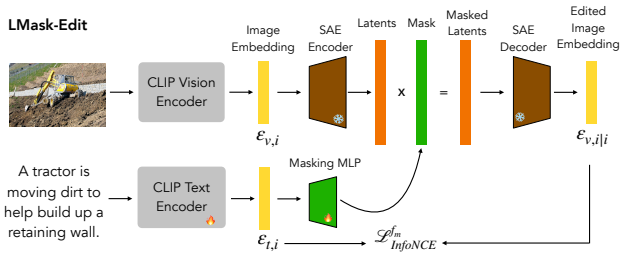


Fig. 2. **Our proposed LMask-Edit for obtaining text-conditioned image embeddings.** We train a TopK SAE [9] over CLIP image embeddings, and then use an MLP trained using the InfoNCE loss to learn a mask over the SAE latents to obtain conditioned image embeddings. For details, see Secs. 3 and 4.

text conditioning. We show that our proposed approach, **LMask-Edit** (Fig. 1, left), can preserve attributes present in the text while discarding information about other attributes, and improve cross-modal alignment without losing performance. We then apply LMask-Edit to CLIP [12, 26], SigLIP [39], and AlignCLIP [7] models trained on natural images, and show that it improves (Fig. 1, right) image-to-text and text-to-image retrieval performance, both for coarse-grained (MS COCO [16], Flickr [25]) and fine-grained retrieval (DOCCI [21], IIW [10]). Using the RoCOCO [23] benchmark, we also show that LMask-Edit leads to strong gains in robust retrieval. In summary, our **contributions** are:

- CLIP-Guided SAEs (CG-SAEs), a controlled setup with latents that encode predefined concepts, to study their utility for preserving targeted concepts.
- LMask-Edit, a framework to enhance cross-modal alignment via SAEs by directly controlling the information imbalance in a *post hoc* manner, by learning to edit image embeddings conditioned on text.

We use CG-SAEs to a show a proof-of-concept of our approach through controlled experiments on MAD [31], and show that LMask-Edit leads to improved retrieval, robust retrieval, and cross-modal alignment across diverse real-world datasets. We discuss related work in Sec. A.

2. Using SAEs to edit representations

In this section, we motivate our text-conditioned image editing approach. We then present LMask-Edit in Sec. 3.

CLIP [26]. Let $\mathcal{D} = \{(v_i, t_i)\}_{i=1}^N$ be a paired dataset of images v_i and their corresponding texts t_i. A CLIP model $M = (f_v, f_t)$ consists of a vision and text encoder respectively which provide corresponding embeddings $(\varepsilon_{v,i}, \varepsilon_{v,t})$, i.e. $\varepsilon_{v,i} = f_v(v_i) \in \mathbb{R}^d$ and $\varepsilon_{t,i} = f_t(t_i) \in \mathbb{R}^d$. The model is then trained with a contrastive InfoNCE loss [22]:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{2N}\sum_{i=1}^{N}\sum_{m\in\{v,t\}}\log\frac{e^{\hat{\varepsilon}_{m,i}\cdot\hat{\varepsilon}_{\bar{m},i}/\tau}}{\sum_{j=1}^{N}e^{\hat{\varepsilon}_{m,i}\cdot\hat{\varepsilon}_{\bar{m},j}/\tau}}. \quad (1)$$

where $\bar{m}$ denotes the opposite modality ($\bar{v} = t$, $\bar{t} = v$), τ is a learnable temperature and $\hat{x} = \frac{x}{\|x\|_2}$. Eq. (1) pulls embeddings of corresponding (positive) image-text pairs $(\varepsilon_{v,i}, \varepsilon_{t,i})$ close to each other and pushes embeddings of other (negative) image-text pairs $(\varepsilon_{v,i}, \varepsilon_{t,j}), i \neq j$ away to learn a shared semantic embedding space.

Sparse autoencoders [1, 9] consist of a linear encoder $W_E \in \mathbb{R}^{d\times d_1}$ and a linear decoder $W_D \in \mathbb{R}^{d_1\times d}$ where typically $d_1 \gg d$. The encoder maps an input $x \in \mathbb{R}^d$ to latents $z = \text{ReLU}(W_E^T(x - b_{pre}) + b_E)$, where $b_{pre} \in \mathbb{R}^d, b_E \in \mathbb{R}^{d_1}$ are learnt biases and z is a sparse disentangled concept representation of x. The decoder then reconstructs x using z, i.e. $\tilde{x} = W_D^T z + b_{pre}$. The SAE is trained with a combination of a L_2 reconstruction loss and a sparsity loss $\mathcal{L}_{\text{SAE}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{SAE}}\mathcal{L}_{\text{sparse}}$, where λ_{SAE} is a hyperparameter. Specifically, we use the TopK SAE [9] where, for sparsity, we select the top K activated latents in z for hyperparameter K.

Next, we use disentangled SAE latents to improve image-text alignment, by only preserving information in image that is present in a conditioned text. We validate this by controlled tests which we describe in Sec. 2.1.

2.1. Controlled setup for representation editing

To understand whether subselecting SAE latents before reconstruction can effectively edit embeddings, we construct a controlled setup using synthetic data with known attributes, where we induce each SAE latent to represent a predefined attribute. In contrast to a typical setup where SAEs automatically learn to disentangle concepts, this is designed to allow for targeted editing, and we refer to it as CLIP-guided SAE (CG-SAE). Let $C = \{c_j\}_{j=1}^S$ be a set of text concepts, and let (W_E, W_D) be an SAE to be trained on image embeddings of CLIP. Then, for an embedding $\varepsilon_{v,i}$, $z_i = \text{TopK}(\text{ReLU}(W_E^T(\varepsilon_{v,i} - b_{pre}) + b_E))$. where $z_i = [z_{i,r}]_{r=1}^{d_1}$ such that elements in z_i outside the top K elements are zeros.

Recent work [29] showed that latents $z_{i,r}$ could be assigned meaningful concept names in C post hoc by selecting the text embedding that is closest to their corresponding decoder weight vector $W_{D,r}$, i.e. $q = \arg\max_j \cos(W_{D,r}; c_j)$. Inspired by this, for our conditioning, we propose the opposite—given C, we fix the rows of the decoder weights W_D to be text embeddings of concepts $c_j \in C$, i.e. $W_{D,r} = \hat{\varepsilon}_{t,c_r}$, where $1 \leq r \leq S$ and $d_1 = S$, and then only train the SAE encoder.

Once trained, given an attribute set $C_A = \{c_q\} \subseteq C$, we select latents *not* present in the text, i.e. $\bar{C} = C \setminus C_A$ and set their activations to zero, i.e. $z_{p|q} = [z_p]_{z_{p,c_{q'}}=\vec{0}},\ \forall c_{q'} \in \bar{C}$. The edited image embedding of an image v_p conditioned on text t_q is then given by $\varepsilon_{v,p|q}$, where $\varepsilon_{v,p|q} = W_D^T z_{p|q} + b_{\text{pre}}$. We evaluate the following: **(R1)** if each CG-SAE la-

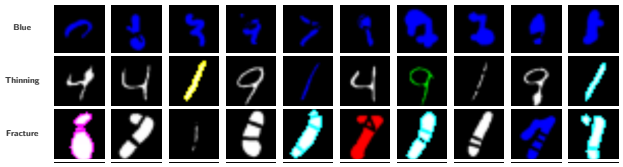


Fig. 3. **Qualitative examples of top activating images when CG-SAE is trained with fixed latents.**

tent indeed encodes the concept assigned to it, and **(R2)** if masking out specific attributes removes information about that attribute from the reconstructed embedding from the SAE.

Experimental setup. We use the synthetic MAD dataset [31], which consists of MNIST digits with colors and morphological transforms. Every image is characterized by six attributes—digit, color, thickthinning, swelling, scaling, and fracture—each of which can be one among a set of predefined options, which are 26 in total. For example, the 'thickthinning' attribute for an image could be 'thickening', 'thinning' or 'nothickthinning', depending on the transform applied to the image. For each image, we create captions that contain the digit and three randomly selected attributes out of the remaining five to simulate information imbalance, and we train small CLIP models [12, 26] following [31]. Then, we train a TopK SAE [9] on CLIP image embeddings with a latent dimension of 26 (equal to the number of attributes for the dataset), with each of its decoder weights assigned as the normalized text embedding of one of the attribute values. See Sec. B.1 for details.

Concept disentanglement (R1). We evaluate if our CG-SAE learns to disentangle image embeddings into the fixed set of predefined concepts C. We show qualitative examples of top activating images for a selection of latents in Fig. 3, and find that they are highly consistent, e.g. for the 'blue' latent, the top activating images are all digits of the color blue, while also being diverse in all other attributes. We provide a quantitative evaluation in Sec. B.2.

Editing image representations (R2). We evaluate if masking latents corresponding to a single attribute discards information about that attribute from the reconstructed representations. To do this, we pick a single attribute 'thickthinning', and set the three latents corresponding to it ('thickening', 'thinning', 'nothickthinning') to zero. We then perform classification on all six attributes using the cosine similarity with the text embeddings for attribute-specific classes. For example, for 'Thickthinning', given an edited embedding $\varepsilon_{v,p|q}$, $\text{Pred}_{\text{thickthinning}} = \arg\max \cos(\varepsilon_{v,p|q}; \varepsilon_{t,j})$ where $t_j \in \{\text{thickening}, \text{thinning}, \text{nothickthinning}\}$. We find (Fig. 5) that the classification accuracy drops to near random chance while that of all other attributes remains close to that with the original embedding. This shows that our conditioning can be an effective way to remove information present in

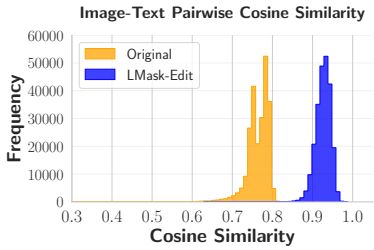


Fig. 4. **Vision-language alignment.** Improved Pairwise image-text similarities between positive pairs after applying LMask-Edit.

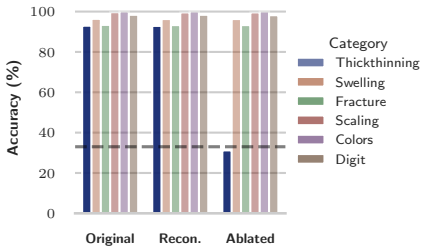


Fig. 5. **Accuracy after ablating a single latent.** We find that the the accuracy for 'Thickthinning' drops to random chance (dotted line) with our edited image embeddings (right group), while the other attributes continue to maintain high accuracy. See Sec. 2.

the image embeddings that is not in the text. We further evaluate the pairwise image-text cosine similarities between positive image-text pairs before and after conditioning in Fig. 4, and find that the alignment increases significantly.

3. LMask-Edit: learning to mask latents based on conditioned text

A predefined set of concepts as used in Sec. 2.1 is typically not available. In this section, we relax this assumption and propose LMask-Edit, an approach to learn a mask over SAE latents z to obtain the text-conditioned image embeddings. An overview of our approach is shown in Fig. 2.

We keep the CLIP vision encoder and a trained SAE frozen, and train a small multi-layer perceptron (MLP) network consisting of linear transforms with ReLU $f_m : \mathbb{R}^d \to \mathbb{R}^{d_1}$ that maps text embeddings $\varepsilon_{t,i}$ to a mask $m_i \in [0,1]^{d_1}$, i.e. $m_i = \sigma(f_m(\varepsilon_{t,i}))$, where $\sigma(\cdot)$ is the sigmoid function. Then, the edited image embedding of an image v_p conditioned on text t_q is given by $\varepsilon_{v,p|q}$, where $\varepsilon_{v,p|q} = W_D^T(z_p \odot m_q) + b_{\text{pre}}$, where $\odot$ is the element-wise product. We train f_m and fine-tune the CLIP text encoder f_t using the InfoNCE loss (Eq. (1)) to pull the edited embeddings towards their conditioning texts and away from other texts:

$$\mathcal{L}^{f_m}_{\text{InfoNCE}} = -\frac{1}{2N}\sum_{i=1}^{N}\left[\log\frac{e^{\hat{\varepsilon}_{v,i|i}\cdot\hat{\varepsilon}_{t,i}/\tau}}{\sum_{j=1}^{N} e^{\hat{\varepsilon}_{v,i|i}\cdot\hat{\varepsilon}_{t,j}/\tau}} + \log\frac{e^{\hat{\varepsilon}_{v,i|i}\cdot\hat{\varepsilon}_{t,i}/\tau}}{\sum_{j=1}^{N} e^{\hat{\varepsilon}_{v,j|j}\cdot\hat{\varepsilon}_{t,i}/\tau}}\right]. \quad (2)$$

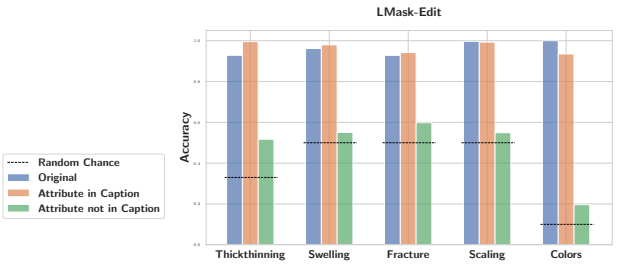


Fig. 6. **Effectiveness of conditioning.** We find that attributes present in the conditioning text are preserved in the edited embedding, while attributes that are absent are classified at close to random chance accuracy. See Sec. 3 for details.

Experimental setup. We follow the setup from Sec. 2; however, we keep the decoder weights of SAE learnable. We use 3-layer MLPs for the masking module.

Effectiveness of conditioning. We evaluate for attribute-wise classification controlling for captions. We bin the attributes for each image depending on whether the caption contains it (Sec. 2), and report the classification accuracy in Fig. 6. We find that the accuracy for attributes *present* in the caption (orange) remains similar to the accuracy from the original embeddings (blue), while the accuracy for attributes *absent* from the caption (green) reaches near random chance (dotted black line). This shows that LMask-Edit is effective in performing conditioning such that the edited embedding preserves information present in the text while discarding the remaining attributes.

Vision-language alignment. We show the pairwise image-text cosine similarities between positive image-text pairs before and after conditioning in Fig. 4, and find that they increase significantly after conditioning (blue) as compared to before (orange).

4. Using LMask-Edit for cross-modal retrieval

In this section, we extend to CLIP models trained on natural images for improving alignment and retrieval performance. Despite its effectiveness in the synthetic setup, the objective from Eq. (2)) only uses edited image embeddings $\hat{\varepsilon}_{v,i|i}$ conditioned on *their own* corresponding positive captions. However, image-to-text and text-to-image retrieval involve selecting from a set of candidates containing both positive and negative captions, which would require conditioning both positive and negative pairs. So, we modify the denominator in Eq. (2) to include negative conditioning, i.e. $e^{\hat{\varepsilon}_{v,i|i}}$ becomes $e^{\hat{\varepsilon}_{v,i|j}}$ in the first term and $e^{\hat{\varepsilon}_{v,j|j}}$ becomes $e^{\hat{\varepsilon}_{v,j|i}}$ in the second term. See Sec. C.2 for more details.

Experimental setup. Following previous works [7, 11, 14, 20], we use the CC12M dataset [3] to train a CLIP [26] ViT-B/16 [6], CLIP ViT-L/14, and a SigLIP [39] ViT-B/16 model. Full details are provided in Sec. C.1.

Retrieval performance and post-hoc applicability. We evaluate for text-to-image and image-to-text retrieval across coarse (MS COCO [16], Flickr30k [25]) and fine-grained (DOCCI [21], IIW [10]) retrieval benchmarks. We find (Tab. 1 and Sec. C.3) that LMask-Edit consistently improves retrieval performance across all datasets and models. Besides our CLIP and SigLIP models, we also compare against AlignCLIP [7], which recently proposed using intra-modal separation objectives during CLIP training to improve cross-modal alignment. We find that our post hoc approach can also help AlignCLIP, which shows its versatility and utility for use in tandem with such methods to improve model performance.

Table 1. **Coarse-grained retrieval performance on Flickr30k [25] and DOCCI [21].** Model: ViT-B/16, Dataset: CC12M.

Model	Flickr30k [16]				DOCCI [25]			
	I → T		T → I		I → T		T → I	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
CLIP	59.66	83.73	42.46	70.33	20.38	42.36	7.16	16.96
+Ours	**64.20**	**85.70**	**44.75**	**72.10**	**24.20**	**48.68**	**8.55**	**19.88**
SigLIP	62.23	85.40	**44.69**	70.61	20.68	41.84	7.41	17.17
+Ours	**62.82**	**86.19**	43.77	**71.56**	**24.52**	**48.60**	**8.47**	**20.02**
AlignCLIP	57.49	82.35	41.91	70.01	20.32	41.88	**7.39**	**17.43**
+Ours	**61.24**	**83.53**	**42.31**	**70.33**	**23.18**	**47.70**	8.32	19.76

Table 2. **Robust retrieval on the RoCOCO benchmark [23].**

Model	Diff-concept			Danger		
	R@1 (↑)	drop rate (↓)	RSMS (↓)	R@1 (↑)	drop rate (↓)	RSMS (↓)
CLIP	22.12	11.14	43.90	22.92	10.34	43.72
+Ours	**25.60**	**10.40**	**35.70**	**27.12**	**8.88**	**31.00**

Robust retrieval performance. We additionally evaluate LMask-Edit for robust image-to-text retrieval, using the RoCOCO [23] benchmark. This augments the caption set of MS COCO [16] with perturbed captions containing irrelevant concepts that alter their meaning, and should not be retrieved by the model. Following [23], we report (Tab. 2 and Sec. C.4) the drop rate (percentage drop in retrieval after augmenting captions), and Recall Score of Manipulated Samples (RSMS) (fraction of data where a perturbed entity was retrieved on top). We find that LMask-Edit provides consistent improvements across these metrics on all data splits, showing its promise for improving model safety.

5. Conclusion

In this work, we explored the poor cross-modal alignment in contrastive VLMs by introducing LMask-Edit, a framework that conditions image embeddings on text. This method refines image representations to align more closely with textual data, and improves retrieval performance. Our experiments on both synthetic and real-world datasets demonstrate its effectiveness and its ability to complement existing training approaches. Future work will focus on scaling these results to large-scale CLIP models.

Acknowledgements

Funded in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - GRK 2853/1 “Neuroexplicit Models of Language, Vision, and Action” - project number 471607914.

References

[1] 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. 1, 2, 8

[2] Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning Multi-Level Features with Matryoshka Sparse Autoencoders. *arXiv preprint arXiv:2503.17547*, 2025. 8

[3] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing Web-Scale Image-Text Pre-training to Recognize Long-Tail Visual Concepts. In *CVPR*, pages 3558–3568, 2021. 4

[4] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023. 1, 8

[5] Bartosz Cywiński and Kamil Deja. SAeUron: Interpretable Concept Unlearning in Diffusion Models with Sparse Autoencoders. *arXiv preprint arXiv:2501.18052*, 2025. 8

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021. 4, 9

[7] Sedigheh Eslami and Gerard de Melo. Mitigate the gap: Investigating approaches for improving cross-modal alignment in clip. In *ICLR*, 2025. 1, 2, 4, 7, 8, 16, 19, 22

[8] Eoin Farrell, Yeu-Tong Lau, and Arthur Conmy. Applying sparse autoencoders to unlearn knowledge in language models. *arXiv preprint arXiv:2410.19278*, 2024. 8

[9] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024. 2, 3, 8, 9, 16

[10] Roopal Garg, Andrea Burns, Burcu Karagol-Ayan, Yonatan Bitton, Ceslee Montgomery, Yasumasa Onoe, Andrew Bunner, Ranjay Krishna, Jason Michael Baldridge, and Radu Soricut. ImageInWords: Unlocking Hyper-Detailed Image Descriptions. In *EMNLP*, pages 93–127, 2024. 2, 4, 17, 18, 19, 20

[11] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining. *Advances in Neural Information Processing Systems*, 35:6704–6719, 2022. 4

[12] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. 1, 2, 3, 8, 9, 16

[13] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning with Noisy Text Supervision. In *ICML*, pages 4904–4916. PMLR, 2021. 8

[14] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021. 4

[15] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *NeurIPS*, 2022. 1, 8, 22

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2, 4, 17, 18, 19, 20

[17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 8

[18] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019. 9, 16

[19] Marco Mistretta, Alberto Baldrati, Lorenzo Agnolucci, Marco Bertini, and Andrew D Bagdanov. Cross the Gap: Exposing the Intra-modal Misalignment in CLIP via Modality Inversion. In *ICLR*, 2025. 8

[20] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European conference on computer vision*, pages 529–544. Springer, 2022. 4

[21] Yasumasa Onoe, Sunayana Rane, Zachary Berger, Yonatan Bitton, Jaemin Cho, Roopal Garg, Alexander Ku, Zarana Parekh, Jordi Pont-Tuset, Garrett Tanzer, et al. DOCCI: Descriptions of Connected and Contrasting Images. In *ECCV*, pages 291–309. Springer, 2024. 2, 4, 17, 18, 19, 20

[22] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[23] Seulki Park, Daeho Um, Hajung Yoon, Sanghyuk Chun, and Sangdoo Yun. RoCOCO: Robustness Benchmark of MS-COCO to Stress-Test Image-Text Matching Models. In *ECCV*, pages 71–91. Springer, 2024. 2, 4, 16, 17

[24] 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, 16

[25] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. In *ICCV*, pages 2641–2649, 2015. 2, 4, 17, 18, 19, 20

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models from Natural Language Supervision. In *ICML*, pages 8748–8763, 2021. 1, 2, 3, 4, 8

[27] Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving dictionary learning with gated sparse autoencoders. In *NeurIPS*, 2024. 8

[28] 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. 8

[29] Sukrut Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-Name: Task-Agnostic Concept Bottlenecks via Automated Concept Discovery. In *ECCV*, 2024. 2, 8

[30] Anton Razzhigaev, Arseniy Shakhmatov, Anastasia Maltseva, Vladimir Arkhipkin, Igor Pavlov, Ilya Ryabov, Angelina Kuts, Alexander Panchenko, Andrey Kuznetsov, and Denis Dimitrov. Kandinsky: An Improved Text-to-Image Synthesis with Image Prior and Latent Diffusion. In *EMNLP (Demos)*, 2023. 1, 8

[31] Simon Schrodi, David T Hoffmann, Max Argus, Volker Fischer, and Thomas Brox. Two Effects, One Trigger: On the Modality Gap, Object Bias, and Information Imbalance in Contrastive Vision-Language Models. In *ICLR*, 2025. 1, 2, 3, 8, 9

[32] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset for Automatic Image Captioning. In *ACL*, pages 2556–2565, 2018. 16

[33] Peiyang Shi, Michael C Welle, Mårten Björkman, and Danica Kragic. Towards understanding the modality gap in clip. In *ICLR 2023 workshop on multimodal representation learning: perks and pitfalls*, 2023. 8

[34] 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. 1, 8

[35] Rui Xiao, Sanghwan Kim, Mariana-Iuliana Georgescu, Zeynep Akata, and Stephan Alaniz. FLAIR: VLM with Fine-grained Language-informed Image Representations. *arXiv preprint arXiv:2412.03561*, 2024. 8

[36] Shaoan Xie, Lingjing Lingjing, Yujia Zheng, Yu Yao, Zeyu Tang, Eric P Xing, Guangyi Chen, and Kun Zhang. SmartCLIP: Modular Vision-language Alignment with Identification Guarantees. In *CVPR*, pages 29780–29790, 2025. 8, 20

[37] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive Captioners are Image-Text Foundation Models. *TMLR*, 2022. 8

[38] Vladimir Zaigrajew, Hubert Baniecki, and Przemyslaw Biecek. Interpreting CLIP with Hierarchical Sparse Autoencoders. *arXiv preprint arXiv:2502.20578*, 2025. 8

[39] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *ICCV*, pages 11975–11986, 2023. 1, 2, 4, 8

Improved Vision-Language Alignment via Text-Conditioned Image Embeddings

Appendix

In this supplement, we provide implementation details, additional results, and a discussion on the limitations and broader impact of our work. In Sec. B, we provide implementation details and additional results on our controlled synthetic setup. Then, in Sec. C, we provide implementation details and further results on evaluating LMask-Edit on CLIP models trained with natural images, including results with SharedCLIP [7]. In Sec. D, we briefly discuss limitations and broader impact.

A. Related work

Vision-language models (VLMs) [12, 13, 26, 34, 37, 39] learn a joint aligned embedding space between images and texts. They typically consist of separate unimodal image and text encoders that each provide an embedding, and are trained using contrastive losses so that embeddings of similar image-text pairs are placed close to each other and dissimilar pairs are placed far apart. Such models are useful for a diverse set of multimodal tasks such as image-to-text and text-to-image retrieval and zero-shot classification. Their embeddings are also used as a bridge between vision and language for text-to-image diffusion models [30] and large multimodal models (LMMs) [17]. In this work, we focus on CLIP [12, 26] and propose an approach to obtain edited image embeddings by conditioning the original image embeddings on text, so that it only preserves information presented in the conditioning text.

Modality gap [7, 15, 19, 31, 33] is a phenomenon observed in trained VLMs, where, despite the contrastive training objective, image and text embeddings lie in different regions of the embedding space. While initially attributed to the "cone effect" at initialization [15], recent work [31] showed that a likely cause is the information imbalance between the two modalities, i.e., images contain more information than is described in their corresponding caption, which forces the model to push apart their embeddings to reduce the contrastive loss. Existing post hoc approaches to reduce the modality gap [15, 31] have been shown to come at the cost of degraded performance. Recently, AlignCLIP [7] proposed to use an intra-modal separation loss during CLIP training to improve alignment and downstream performance. In our work, we propose a pipeline to explicitly control the information imbalance post hoc by editing image embeddings conditioned on a given text embedding, in order to only preserve information in the image that is also present in the text, and show that this can also benefit methods such as AlignCLIP. More recently, FLAIR [35] proposes using text-conditioned attention pooling to improve fine-grained retrieval, but in contrast to our work, trains models from scratch. Further, SmartCLIP [36] also similarly proposed masking specific dimensions of learnt CLIP embeddings. In contrast, we use sparse autoencoders to first disentangle embeddings before applying a soft mask.

Sparse autoencoders (SAEs) [1, 2, 4, 9, 27, 28] are a popular mechanistic interpretability tool to disentangle activations learnt by a deep network into constituent human understandable concepts. While originally used in the context of LLMs [1, 4], SAEs have recently been used to decompose concepts from CLIP vision embeddings [29, 38] for use in downstream tasks such as constructing concept bottleneck models [29]. SAEs have also been used as a tool for steering models by performing edits to their latents in the context of large language models [8] and text-to-image diffusion models [5]. In our work, we use SAEs in our pipeline to edit CLIP embeddings, by editing their latents in order to only preserve image concepts presented in the corresponding conditioning text and remove other concepts.

B. Results on the MAD Dataset

B.1. Implementation details

Dataset. The MAD dataset [31] is a synthetic dataset that consists of images of digits with colors and morphological transforms. Specifically, it consists of the following attributes: 'Digit': {0,1,2,3,4,5,6,7,8,9}, 'Thickthinning': {thickening, thinning, no thickthinning}, 'Scaling': {large, small}, 'Fracture': {fracture, no fracture}, 'Swelling': {swelling, no swelling}, and 'Color': {gray, red, green, blue, cyan, magenta, yellow}, which gives a total of 6 attribute categories and an aggregate of 26 attribute values. The training dataset consists of 1.44 million images, and the test data consists of 240,000 images. For training CLIP models, captions are generated by using the digit and a random sample of three out of the remaining five attribute categories in the image, placed in a random order with a '-' separator.

CLIP models. Following [31], we train CLIP models consisting of a 6-layer ViT [6] as the vision encoder and a 6-layer transformer as the text encoder, with a shared embedding dimension of 18, for 200 epochs with a batch size of 128 and a weight decay of 0.1 using the AdamW [18] optimizer. Given the 26 possible attributes per image, the text encoder uses a tokenizer with a vocabulary of size 28, after accounting for the start and end tokens. We sweep over learning rates of $\{10^{-5}, 5 \times 10^{-4}, 5 \times 10^{-5}\}$ and pick the learning rate 5×10^{-4} and the final checkpoint with the lowest loss. We use the cosine annealing for the learning rate. For finetuning the text encoder along with the learnt mask, we sweep over learning rates $\{10^{-2}, 10^{-3}, 10^{-4}, 5 \times 10^{-4}, 10^{-5}, 10^{-6}\}$ and pick the learning rate 10^{-6}. Our code is based on the implementation from OpenCLIP [12] using PyTorch [24].

SAE models. We use the TopK SAE implementation of [9]. This particular type of SAE chooses the top few ($= K$) SAE latents to reconstruct the original CLIP embeddings and uses an auxiliary loss that approximates the reconstruction error using the top few (=auxK) dead latents. For our setup, we use K as 12, auxK as 18, after sweeping over these hyperparameters and we train the SAE for 200 epochs. We choose the SAE configuration using with the lowest reconstruction error. We sweep over learning rates $\{10^{-1}, 10^{-2}\}$ and expansion factor $1, 2, 4$ and pick 10^{-2}, 1 respectively for both the SAEs used for the CG-SAE and LMask-Edit method.

Learnt masks. We use a 3-layer MLP with a hidden dimension of 256 and ReLU activations between linear layers. This MLP predicts values to mask the SAE latents. We sweep over learning rates 0.1, 0.01, and 0.001, and choose 0.01 as the optimal learning rate. We train for 25 epochs with 5 warmup epochs using the AdamW optimizer [18].

B.2. Additional results for concept disentanglement with CG-SAE

In this section, we provide full results for concept disentanglement from our CG-SAEs, as discussed in Sec. 2. In Figs. B1 and B2, we show top activating images for each latent, expanding on Fig. 3. Both quantitatively and qualitatively, we find that our CG-SAE effectively disentangles concepts and each latent activates highly only on the attribute value assigned to it.

In Fig. B3, we quantitatively validate this by showing the area under the receiver operating characteristic (ROC) curve (AUC) across images in the dataset for the two latents assigned to the 'swelling' attribute. We find that the latents are able to disentangle the attributes well, with the AUC being close to 1 for the attribute value the latent is assigned to, 0.5 for unrelated attributes, and 0 for attribute values anti-correlated to the assigned attribute (i.e. 'no swelling', for the 'swelling' latent). Figs. B4 and B5 show AUC ROC plots for other attribute categories.

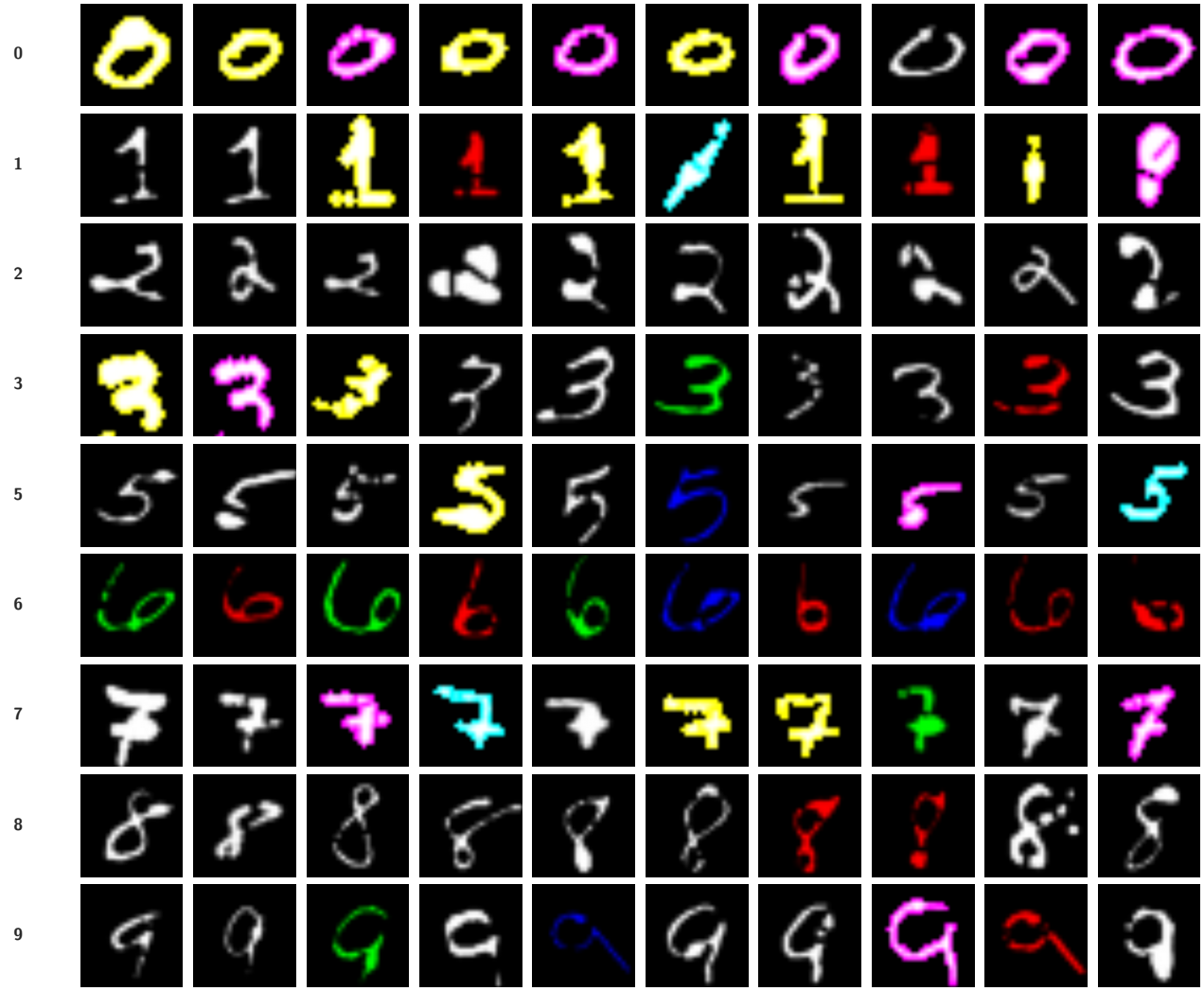


Fig. B1. **Qualitative examples of top activating images for all the digits of the CG-SAE latents.** Each row corresponds to a CG-SAE latent and is labelled with the predefined concept that is assigned to that latent (Sec. 2.1). The columns show examples of images that maximally activate these latents. We find that the CG-SAE learns to disentangle the CLIP image features into concepts as specified by the fixed weight of the latent.

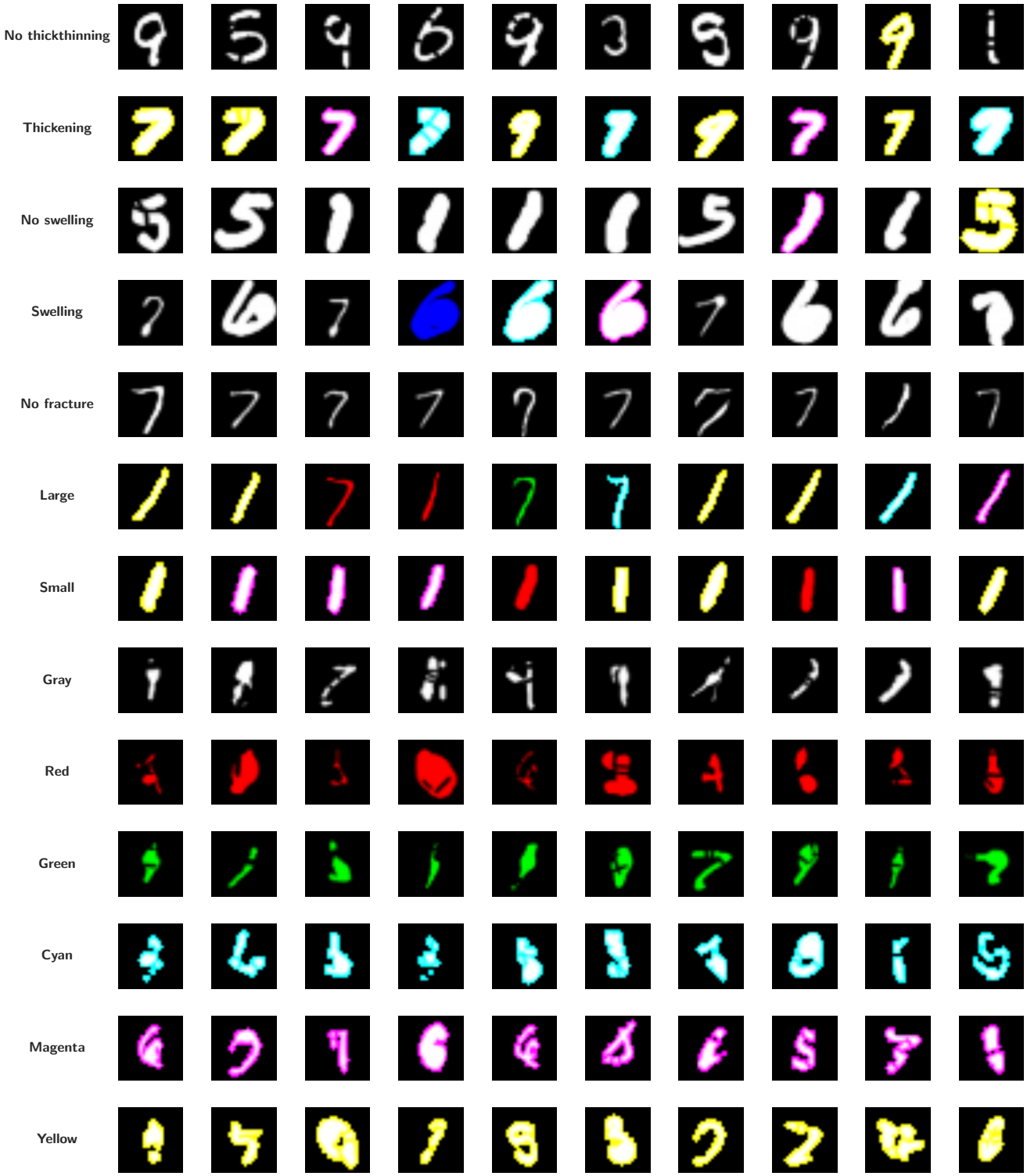


Fig. B2. **Qualitative examples of top activating images of the remaining CG-SAE latents.** Each row corresponds to a CG-SAE latent and is labelled with the predefined concept that is assigned to that latent (Sec. 2.1). The columns show examples of images that maximally activate these latents. We find that the CG-SAE learns to disentangle the CLIP image features into concepts as specified by the fixed weight of the latent.

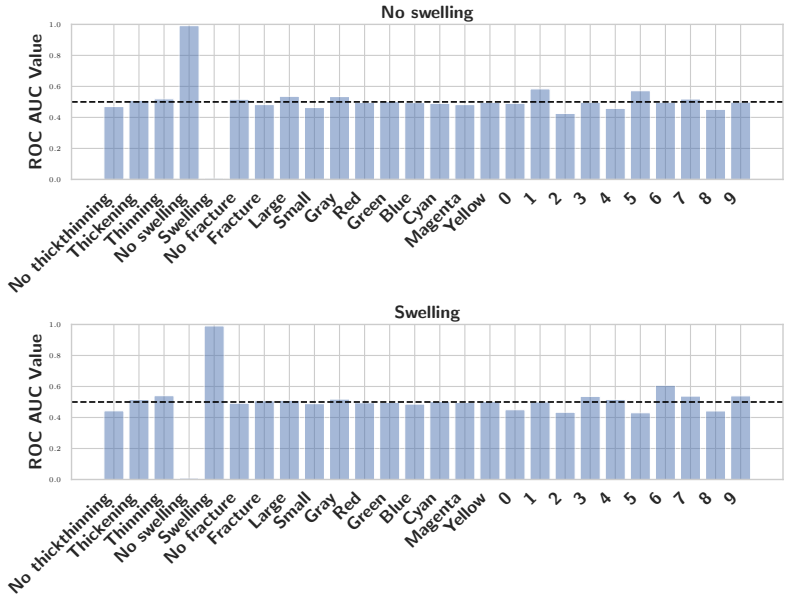


Fig. B3. **Attribute Specificity of CG-SAE latents.** We plot the area under the receiver operating characteristic (ROC) curve (AUC) for CG-SAE latents corresponding to the attribute values 'No swelling' and 'Swelling'. We find that the latents are highly attribute specific, with the AUC being close to 1 for the attribute the latent is assigned to, and 0 for unrelated attributes. This shows that our CG-SAE latents are highly disentangled, despite being assigned to a predefined concept. Interestingly, as can be expected, the AUC of attributes anti-correlated to the attribute assigned to the latent (e.g. 'Swelling' for the 'No swelling' latent) are close to 0. Results for the other latents are provided in the supplement.

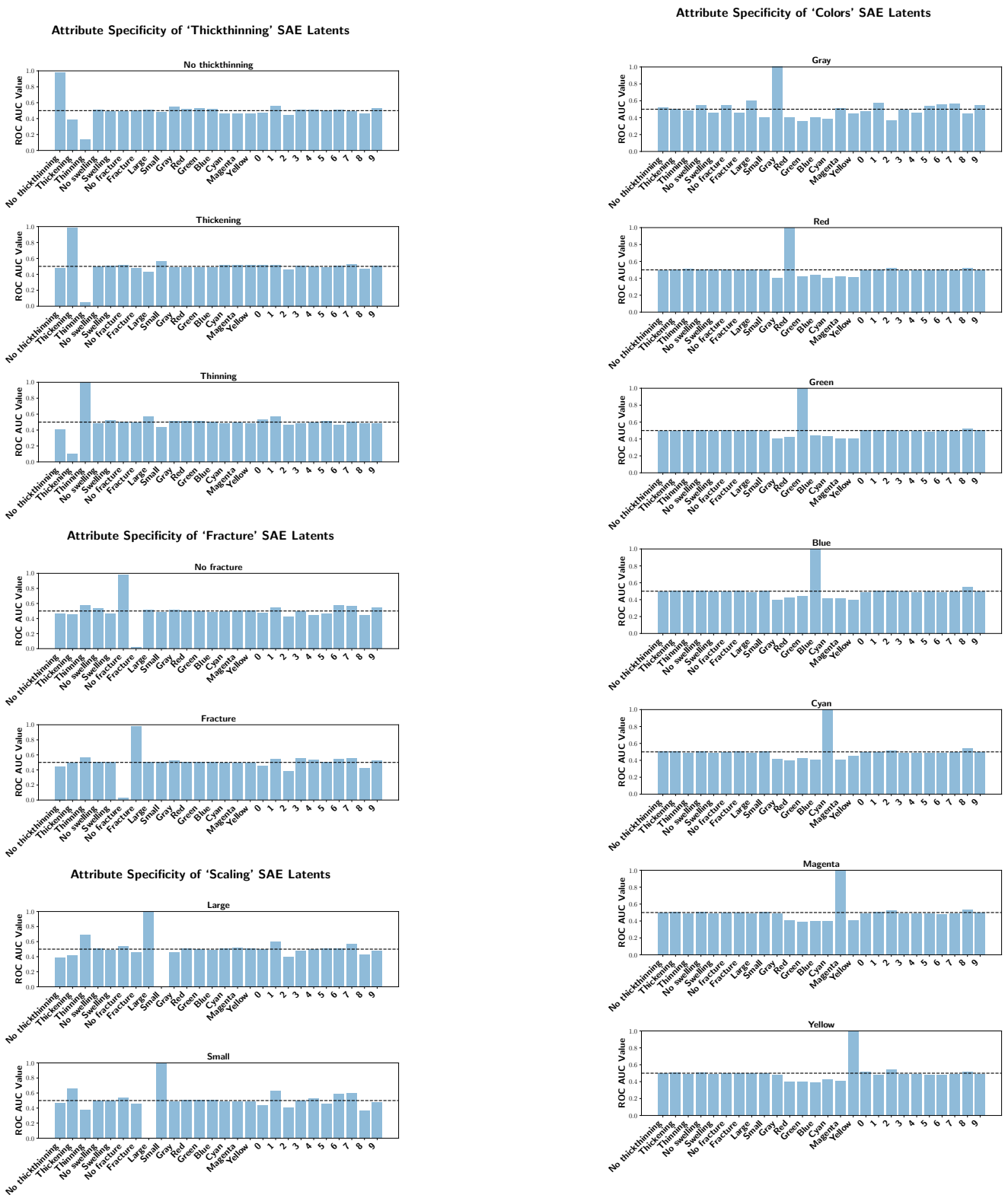


Fig. B4. **Attribute specificity of CG-SAE latents for attribute categories 'Thickthinning', 'Fracture', 'Scaling', and 'Color'.** We plot the area under the receiver operating characteristic (ROC) curve (AUC) for CG-SAE latents corresponding to the attribute values of each attribute category. We find that the latents are highly attribute-specific, with the AUC being close to 1 for the attribute the latent is assigned to, and 0.5 for unrelated attributes. For results for 'Digit', see Fig. B5.

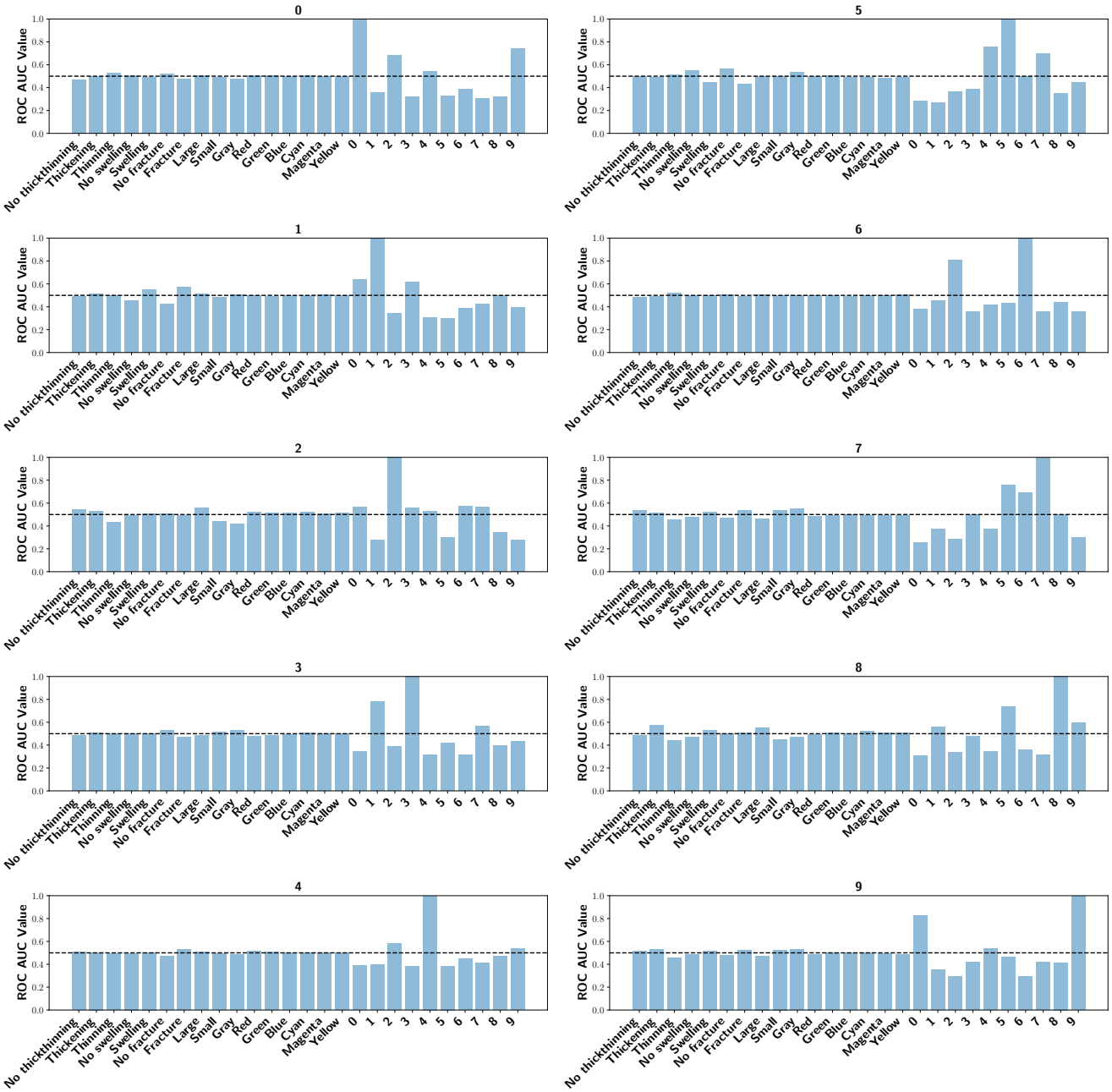


Fig. B5. **Attribute specificity of CG-SAE latents for attribute category 'Digit'.** We plot the area under the receiver operating characteristic (ROC) curve (AUC) for CG-SAE latents corresponding to each digit attribute. We find that the latents are fairly attribute-specific, with the AUC being close to 1 for the attribute the latent is assigned to, and close to 0.5 for unrelated attributes. For results for other attribute categories, see Fig. B4.

B.3. Additional results for effectiveness of conditioning

In this section, we provide additional results on our method's effectiveness of conditioning (Tabs. B1 and B2). Specifically, for robustness, we report results averaged across five runs, and find that the trends observed in Figs. 4 and 6 continue to hold.

Table B1. **Effectiveness of conditioning with CG-SAE for the MAD dataset.** For each attribute category (rows), we report the accuracy (Sec. 2) with the original embeddings (col. 1) and conditioned embeddings, where the attribute of that category is present in the caption (col. 2) and is absent from the caption (col. 3), averaged over five runs. We see that when the attribute is present in the caption, the accuracy is at par with the original embeddings. However, when the attribute is absent from the caption, the accuracy reaches close to random chance (col. 4), which shows that conditioning is effective in only preserving information about attributes present in the caption.

	Original (↑)	Attribute in Caption (↑)	Attribute not in Caption	Random Chance
Thickthinning	92.9	97.8 ± 1.3	35.4 ± 2.1	33.3
Swelling	96.2	96.6 ± 6.8	50.2 ± 0.1	30.0
Fracture	92.8	97.5 ± 3.0	50.5 ± 0.7	50.0
Scaling	99.7	99.9 ± 0.1	60.7 ± 3.4	50.0
Color	100.0	100.0 ± 0.0	34.2 ± 1.7	14.3

Table B2. **Effectiveness of conditioning with LMask-Edit for the MAD dataset.** For each attribute category (rows), we report the accuracy (Sec. 2) with the original embeddings (col. 1) and conditioned embeddings, where the attribute of that category is present in the caption (col. 2) and is absent from the caption (col. 3), averaged over five runs. We see that when the attribute is present in the caption, the accuracy is at par with the original embeddings. However, when the attribute is absent from the caption, the accuracy reaches close to random chance (col. 4), which shows that conditioning is effective in only preserving information about attributes present in the caption.

	Original (↑)	Attribute in Caption (↑)	Attribute not in Caption	Random Chance
Thickthinning	92.9	99.1 ± 0.7	54.3 ± 2.4	33.3
Swelling	96.2	98.3 ± 0.2	57.0 ± 2.7	30.0
Fracture	92.8	93.5 ± 0.9	59.1 ± 1.5	50.0
Scaling	99.7	95.2 ± 2.7	57.1 ± 3.5	50.0
Color	100.0	96.3 ± 1.9	22.6 ± 1.9	14.3

C. Evaluation on CLIP models trained with natural images

C.1. Implementation details

CLIP models. We train CLIP models using the AdamW [18] optimizer for 30 epochs with a batch size of 2048, weight decay of 0.1, and 10,000 warmup steps, similar to [7]. We sweep over learning rates of $\{10^{-3}, 5 \times 10^{-4}\}$ and pick the learning rate 10^{-3} and the final checkpoint with the lowest loss. We use cosine annealing for the learning rate. For fine-tuning the text encoder along with training the maskig module, we sweep over learning rates $\{10^{-4}, 10^{-5}, 10^{-6}\}$. Our code is based on the implementation from OpenCLIP [7, 12] using PyTorch [24].

SAE models. We use the TopK SAE implementation of [9]. This particular type of SAE chooses the top few ($= K$) SAE latents to reconstruct the original CLIP embeddings and uses an auxiliary loss that approximates the reconstruction error using the top few (=auxK) dead latents. For our setup, we use K as 128, auxK as 256, after sweeping over these hyperparameters and we train the SAE for 50 epochs. We sweep over learning rates $\{10^{-3}, 5 \times 10^{-3}, 10^{-4}, 5 \times 10^{-4}\}$, and use an expansion factor of 32 for ViT-B/16 backbones and 16 for ViT-L/14 backbones. We choose the SAE configuration using with the lowest reconstruction error, lowest number of dead nodes, and best classification accuracy on the reconstructed features across different setups.

Learnt masks. We use a 3-layer MLP and ReLU activations between linear layers. This MLP predicts values to mask the SAE latents. We sweep over learning rates $\{10^{-4}, 10^{-5}, 10^{-6}\}$ and select the configuration that is best for retrieval on the validation split of CC3M [32], to ensure generalization to different datasets.. We train for 3 epochs using the AdamW optimizer [18].

Baselines. For the results on SharedCLIP and AlignCLIP in tables Secs. C.5 and C.7 and Tab. 1, we take the checkpoints given by the authors [7] and apply LMask-Edit on them. Note that these checkpoints were trained with a batch size of 512, and in contrast we train our own CLIP models with a batch size of 2048. As a result, the performance of AlignCLIP/SharedCLIP versus our CLIP models cannot be directly compared.

Robust retrieval. Following [23], in Tab. 2, we report the drop rate and the RSMS metric in addition to the retrieval metrics. Drop rate is calculated as $\frac{(R@1 - Rp@1)}{R@1}$, where Rp is the retrieval score under the perturbed setting. RSMS calculates the percentage of newly added semantically perturbed captions that are retrieved in the first spot by the model. We note that, due to that use of SAE to decompose CLIP image features into concepts, our method is unable to work with altered images as given by the benchmark, where the original image is superimposed with a random patch from a different image. As a result, we only report image-to-text retrieval performance in Tab. 2 and Sec. C.4.

C.2. Optimization objective

In Eq. (C.1), we provide the full optimization objective discussed in Sec. 4.

$$\mathcal{L}^{f_m}_{\text{InfoNCE}} = -\frac{1}{2N}\sum_{i=1}^{N}\left[\log\frac{e^{\hat{\epsilon}_{v,i|i}\cdot\hat{\epsilon}_{t,i}/\tau}}{\sum_{j=1}^{N} e^{\hat{\epsilon}_{v,i|j}\cdot\hat{\epsilon}_{t,j}/\tau}} + \log\frac{e^{\hat{\epsilon}_{v,i|i}\cdot\hat{\epsilon}_{t,i}/\tau}}{\sum_{j=1}^{N} e^{\hat{\epsilon}_{v,j|i}\cdot\hat{\epsilon}_{t,i}/\tau}}\right] \tag{C.1}$$

C.3. Additional retrieval results

In Tabs. C1 and C2, we provide full retrieval results on MS COCO [16], Flickr30k [25], DOCCI [21], and IIW [10], including on a CLIP ViT-L/14 model. As discussed in Sec. 4, we find that LMask-Edit improves retrieval performance in most settings.

Table C1. **Coarse-grained retrieval performance on MSCOCO [16] and Flickr30k [25].** All the models are trained on CC12M dataset. We find that LMask-Edit improves retrieval performance across models and datasets.

Model	MS COCO [16]						Flickr30k [25]					
	I → T			T → I			I → T			T → I		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
CLIP ViT-B/16	32.98	59.02	70.04	21.38	45.30	57.03	59.66	83.73	90.14	42.46	70.33	79.31
+LMask-Edit	**35.66**	**61.82**	**72.94**	**23.12**	**47.03**	**58.67**	**64.20**	**85.70**	**91.12**	**44.75**	**72.10**	**81.22**
SigLIP ViT-B/16	33.88	60.30	70.62	22.06	46.21	57.70	62.23	85.40	91.42	**44.69**	70.61	79.33
+LMask-Edit	**36.24**	**61.20**	**72.82**	**22.37**	**46.60**	**58.66**	**62.82**	**86.19**	**91.62**	43.77	**71.56**	**80.73**
AlignCLIP ViT-B/16	32.70	58.92	70.46	**21.79**	**44.55**	**56.41**	57.49	82.35	**89.94**	41.91	70.01	**78.97**
+LMask-Edit	**34.42**	**60.30**	**71.48**	21.35	44.39	55.72	**61.24**	**83.53**	89.15	**42.31**	**70.33**	78.92

Table C2. **Fine-grained retrieval performance on DOCCI [21] and IIW [10].** All the models are trained on CC12M dataset. We report R@1, R@5 and R@10 for both image-to-text and text-to-image retrieval. We see that LMask-Edit improves retrieval performance across models and datasets.

Model	DOCCI [21]						IIW [10]					
	I → T			T → I			I → T			T → I		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
CLIP ViT-B/16	20.38	42.36	53.52	7.16	16.96	22.86	50.98	77.94	**88.89**	16.88	32.66	41.46
+LMask-Edit	**24.20**	**48.68**	**60.06**	**8.55**	**19.98**	**26.80**	**55.72**	**81.21**	88.72	**19.37**	**36.44**	**45.47**
CLIP ViT-L/14	23.60	46.68	57.06	8.21	18.70	24.91	53.92	81.37	89.21	18.05	35.59	44.49
+LMask-Edit	**26.06**	**51.50**	**62.20**	**9.26**	**21.51**	**27.99**	**58.33**	**85.46**	**90.85**	**19.41**	**37.80**	**46.53**
SigLIP ViT-B/16	20.68	41.84	53.70	7.41	17.17	22.77	48.20	76.14	85.13	16.94	33.07	41.42
+LMask-Edit	**24.52**	**48.60**	**59.58**	**8.47**	**20.02**	**26.47**	**57.68**	**81.86**	**88.89**	**19.08**	**36.88**	**46.61**
AlignCLIP ViT-B/16	20.32	41.88	53.38	7.39	17.43	23.69	53.92	82.03	87.58	17.46	34.27	43.57
+LMask-Edit	**23.18**	**47.70**	**58.90**	**8.32**	**19.76**	**26.34**	**59.15**	**84.31**	**91.99**	**19.66**	**36.38**	**45.58**

C.4. Additional robust retrieval results

In Tab. C3, we provide results across all splits for image-to-text retrieval on the RoCOCO benchmark [23]. As discussed in Sec. 4, LMask-Edit improves robust retrieval across settings.

Table C3. **Robust retrieval performance of on the RoCOCO benchmark [23]**. We find that LMask-Edit improves performance over the baseline across settings.

Model	COCO R@1	Rand-voca			Same-concept			Diff-concept			Danger		
		R@1 (↑)	drop rate (↓)	RSMS (↓)	R@1 (↑)	drop rate (↓)	RSMS (↓)	R@1 (↑)	drop rate (↓)	RSMS (↓)	R@1 (↑)	drop rate (↓)	RSMS (↓)
CLIP	33.26	21.26	12.00	46.90	21.74	11.52	43.36	22.12	11.14	43.90	22.92	10.34	43.72
+LMask-Edit	**36.00**	**26.14**	**9.86**	**33.98**	**25.74**	**10.26**	**37.12**	**25.60**	**10.40**	**35.70**	**27.12**	**8.88**	**31.00**

C.5. Cross-modal alignment

In Figs. C1 to C3, we show cross-modal alignment for CLIP, SharedCLIP, and AlignCLIP across four downstream datasets before and after applying LMask-Edit, and find that it improves cross-modal alignment.

Fig. C1. **Cross-modal alignment** on (1) MSCOCO [16], (2) Flickr30k [25], (3) DOCCI [21], (4) IIW [10] on CLIP ViT-B/16. The alignment is measured by the cosine similarity between the positive image-text pairs, and the y-axis denotes the number of data points for each alignment score. We see the distribution after applying LMask-Edit (violet) shifts to the right as compared to the baseline (orange), showing improved alignment.

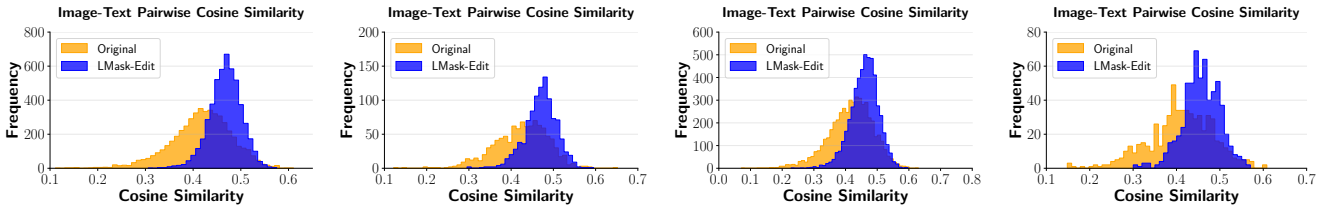


Fig. C2. **Cross-modal alignment with AlignCLIP** on (1) MSCOCO [16], (2) Flickr30k [25], (3) DOCCI [21], (4) IIW [10]. The alignment is measured by the cosine similarity between the positive image-text pairs, and the y-axis denotes the number of data points for each alignment score. We see the distribution after applying LMask-Edit (violet) shifts to the right as compared to the baseline (orange), showing improved alignment.

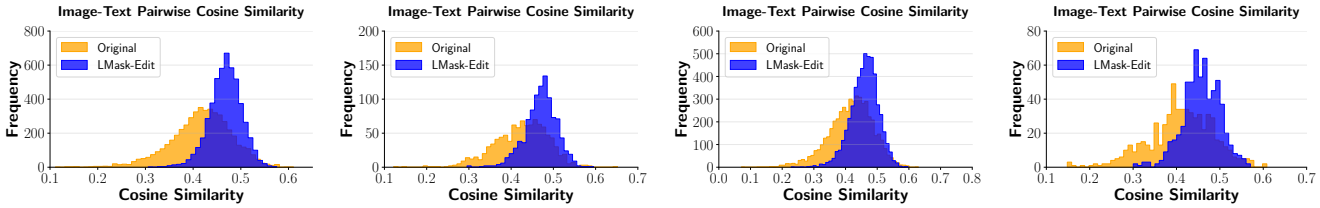
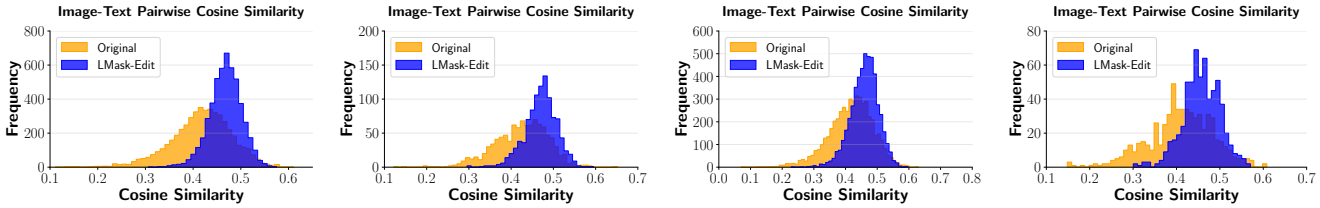


Fig. C3. **Cross-modal alignment with SharedCLIP** on (1) MSCOCO [16], (2) Flickr30k [25], (3) DOCCI [21], (4) IIW [10]. The alignment is measured by the cosine similarity between the positive image-text pairs, and the y-axis denotes the number of data points for each alignment score. We see the distribution after applying LMask-Edit (violet) shifts to the right as compared to the baseline (orange), showing improved alignment.



C.6. Baseline fine-tuning comparison

Our LMask-Edit involves training a masking module for a small number of extra epochs while also fine-tuning the text encoder. For a fairer comparison accounting for this extra training budget, we report in Tabs. C4 and C5 the performance of CLIP models where the text encoder is fine-tuned for the same number of extra epochs. We find that LMask-Edit nevertheless shows improved retrieval performance across datasets.

Table C4. **Coarse-grained retrieval performance on MSCOCO [16] and Flickr30k [25] as compared to fine-tuning CLIP.** All the models are trained on CC12M dataset. We report R@1, R@5 and R@10 for both image-to-text and text-to-image retrieval. We see that LMask-Edit improves retrieval performance across models and datasets in comparison to baseline finetuning as well.

Model	MS COCO [16]						Flickr30k [25]					
	I → T			T → I			I → T			T → I		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
CLIP ViT-B/16	32.98	59.02	70.04	21.38	45.30	57.03	59.66	83.73	90.14	42.46	70.33	79.31
CLIP ViT-B/16 Fine-tuned	35.60	60.78	71.62	22.50	46.40	58.13	63.21	**86.29**	90.93	44.12	71.50	80.47
+LMask-Edit	**35.66**	**61.82**	**72.94**	**23.12**	**47.03**	**58.67**	**64.20**	85.70	**91.12**	**44.75**	**72.10**	**81.22**

Table C5. **Fine-grained retrieval performance on DOCCI [21] and IIW [10] as compared to fine-tuning CLIP.** All the models are trained on CC12M dataset. We report R@1, R@5 and R@10 for both image-to-text and text-to-image retrieval. We see that LMask-Edit improves retrieval performance across models and datasets in comparison to baseline finetuning as well.

Model	DOCCI [21]						IIW [10]					
	I → T			T → I			I → T			T → I		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
CLIP ViT-B/16	20.38	42.36	53.52	7.16	16.96	22.86	50.98	77.94	**88.89**	16.88	32.66	41.46
CLIP ViT-B/16 Fine-tuned	23.54	46.48	56.92	7.79	18.57	25.00	53.10	80.56	88.23	18.34	35.99	45.20
+LMask-Edit	**24.20**	**48.68**	**60.06**	**8.55**	**19.98**	**26.80**	**55.72**	**81.21**	88.72	**19.37**	**36.44**	**45.47**

C.7. Retrieval results on SharedCLIP

Similar to our evaluation with AlignCLIP (Tab. 1 and Sec. C.3), in Tabs. C6 and C7 we report results on applying LMask-Edit to SharedCLIP [7] and find similar gains in performance.

Table C6. **Coarse-grained retrieval performance of LMask-Edit with SharedCLIP on MSCOCO [16] and Flickr30k [25].** All the models are trained on CC12M dataset. We report R@1, R@5 and R@10 for image-to-text and text-to-image retrieval.

Model	MS COCO [16]						Flickr30k [25]					
	I → T			T → I			I → T			T → I		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
SharedCLIP	32.62	58.88	69.94	21.54	44.81	56.87	**60.75**	**84.22**	**89.64**	43.31	69.53	78.86
+LMask-Edit	**33.82**	**59.86**	**70.90**	**22.13**	**45.95**	**57.38**	59.37	83.33	89.05	**44.30**	**70.97**	**79.53**

Table C7. **Fine-grained retrieval performance of LMask-Edit with SharedCLIP on DOCCI [21] and IIW [10].** All the models are trained on CC12M dataset. We report R@1, R@5 and R@10 for image-to-text and text-to-image retrieval.

Model	DOCCI [21]						IIW [10]					
	I → T			T → I			I → T			T → I		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
SharedCLIP	19.02	41.06	52.68	7.16	17.52	23.88	51.14	79.08	87.09	17.10	33.78	43.04
+LMask-Edit	**22.54**	**45.54**	**57.10**	**7.95**	**18.62**	**25.31**	**57.51**	**81.54**	**89.05**	**18.17**	**34.86**	**44.02**

C.8. Comparison against SmartCLIP

In Tabs. C8 and C9, we compare our approach against SmartCLIP [36], applied on the same CLIP models trained on CC12M. We see that LMask-Edit performs comparably on MS COCO and Flickr30k and outperforms on the fine-grained DOCCI and IIW datasets, possibly due to explicit disentanglement from SAEs.

Table C8. **Fine-grained retrieval performance on DOCCI [21] and IIW [10] as compared to SmartCLIP.** All the models are trained on CC12M dataset. We report R@1, R@5 and R@10 for both image-to-text and text-to-image retrieval.

Model	DOCCI [21]						IIW [10]					
	I → T			T → I			I → T			T → I		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
CLIP ViT-B/16	20.38	42.36	53.52	7.16	16.96	22.86	50.98	77.94	88.89	16.88	32.66	41.46
SmartCLIP	21.76	45.04	55.64	7.47	17.83	23.80	54.74	**82.35**	**89.05**	17.62	33.90	42.15
LMask-Edit	**24.20**	**48.68**	**60.06**	**8.55**	**19.98**	**26.80**	**55.72**	81.21	88.72	**19.37**	**36.44**	**45.47**

Table C9. **Coarse-grained retrieval performance on MSCOCO [16] and Flickr30k [25] as compared to SmartCLIP.** All the models are trained on CC12M dataset. We report R@1, R@5 and R@10 for both image-to-text and text-to-image retrieval.

Model	MS COCO [16]						Flickr30k [25]					
	I → T			T → I			I → T			T → I		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
CLIP ViT-B/16	32.98	59.02	70.04	21.38	45.30	57.03	59.66	83.73	90.14	42.46	70.33	79.31
SmartCLIP	35.54	**61.98**	72.90	**23.89**	**48.24**	**59.93**	61.05	85.11	**91.32**	**45.29**	**72.78**	**81.24**
LMask-Edit	**35.66**	61.82	**72.94**	23.12	47.03	58.67	**64.20**	**85.70**	91.12	44.75	72.10	81.22

C.9. Examples of SAE concepts

For the masking mechanism of LMask-Edit to work, it is essential that the trained SAE effectively decomposes the image features into individual concepts. We provide qualitative examples of top activating images for a random selection of SAE latents in Fig. C4, and find that they encode highly consistent concept

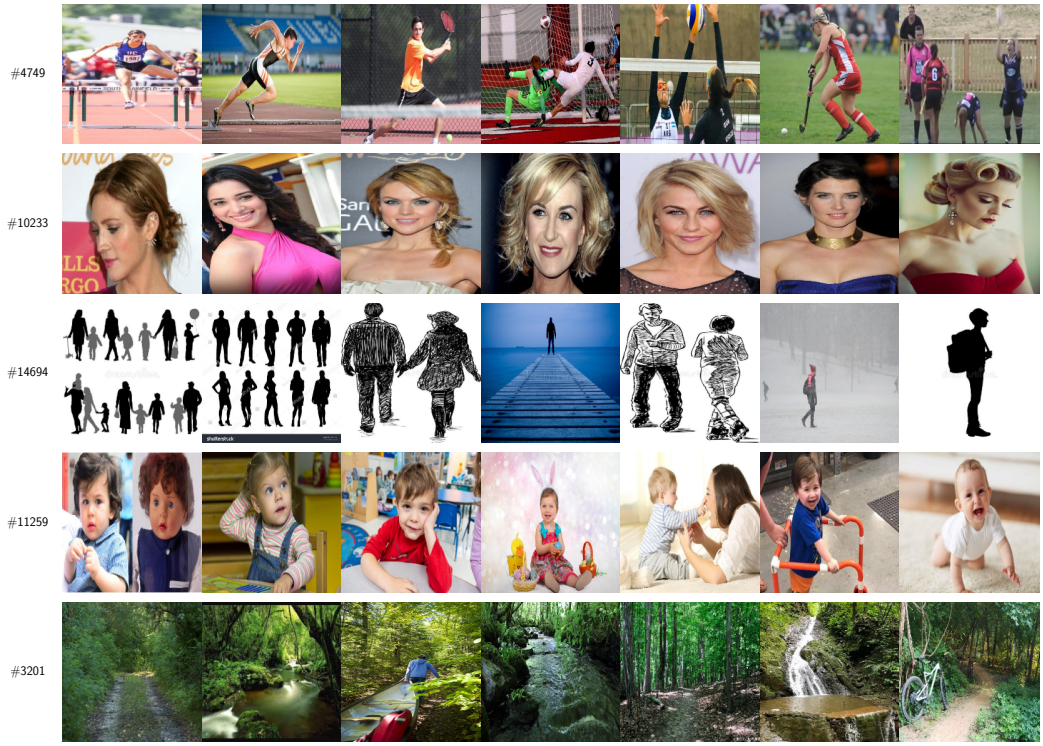


Fig. C4. **Qualitative examples of top activating images of the SAE trained on top of CLIP image embeddings.** Each row corresponds to an SAE latent. The columns show images from CC12M that maximally activate these latents. We find that the SAE is trained well and is able to disentangle the CLIP image features into concepts, which later becomes useful for masking.

D. Limitations and broader impact

In this section, we discuss the limitations (Sec. D.1) and broader impacts (Sec. D.2) of our work.

D.1. Limitations

In our work, we tackle the information imbalance problem that gives rise to a modality gap in vision-language models by proposing a scheme to explicitly control for it. To do this, we make use of sparse autoencoders to edit image embeddings conditioned on text. However, while our proposed LMask-Edit shows strong benefits for a variety of CLIP models trained on CC12M, extending to large-scale models trained on billions of data points remains a challenge, likely owing to the complexity of training sufficiently large and diverse SAEs. Nevertheless, our work intends to provide a clear proof of concept and scaling to models trained on large-scale data would be a fruitful direction for future research.

D.2. Broader impact

The use of vision-language models for multimodal tasks is widespread, from tasks like image retrieval to visual question answering to image generation. This makes it increasingly important that such models work reliably. Lack of proper semantic alignment can lead to biased outputs [15] or poor robustness to distribution shifts [7]. Improved visual-textual alignment can enhance downstream performance of such models and help alleviate the above mentioned drawbacks.