Emergence and Evolution of Interpretable Concepts in Diffusion Models Through the Lens of Sparse Autoencoders

Berk Tinaz* Zalan Fabian* Mahdi Soltanolkotabi
Dept. of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA
{tinaz, zfabian, soltanol}@usc.edu

Abstract

Diffusion models (DMs) can generate diverse images with exceptional visual quality aligned with the input natural language prompt. However, the inner workings of DMs, especially the evolution of internal representations throughout the generative process is still largely a mystery. Mechanistic interpretability techniques, such as Sparse Autoencoders (SAEs), aim at uncovering the fundamental operating principles of models through granular analysis of their features, and have been successful in understanding and steering the behavior of large language models at scale. In this work, we leverage the SAE framework to probe the inner workings of text-to-image DMs, and uncover a variety of human-interpretable concepts in their activations. We find that even before the first reverse diffusion step is completed, the final composition of the scene can be predicted surprisingly well by looking at the spatial distribution of activated concepts. We find that while image composition is mostly finalized by the middle of the reverse process, image style is still subject to change. Finally, we design SAE-based interventions that control the layout and style of the generated image.

1. Introduction

Diffusion models [10, 29] have revolutionized the field of generative modeling and have established state-of-the-art in image [4, 11, 19, 24, 26], audio [15], and video generation [12]. Text-conditioning in DMs [24, 25], i.e. guiding the generation process via text prompts, enables careful customization of generated samples while simultaneously maintaining exceptional sample quality.

While DMs excel at producing images of exceptional quality, the internal mechanisms by which they ground textual concepts in visual features that govern generation remain opaque. The time-evolution of internal representations through the generative process, from pure noise to high-quality images, renders the understanding of DMs even more challenging compared to other deep learning models. A particular blind spot is the early, 'chaotic' stage [33] of diffusion, where noise dominates the generative process.

Recently, a flurry of research has emerged towards demystifying the inner workings of DMs [1, 2, 5, 6, 8, 16, 21, 22, 31]. However, most existing techniques are aimed at addressing particular editing tasks and are not wide enough in scope to provide a more holistic interpretation on the internal representations of diffusion models. Mechanistic interpretability (MI) [20] is focused on addressing the above challenges via uncovering causal mechanisms from inputs to outputs that reveal how neural networks process information internally. Sparse autoencoders have emerged within MI as powerful tools to discover highly interpretable features (or *concepts*) within large models at scale [3]. These learned features enable direct interventions to steer model behavior in a controlled manner. Despite their success in understanding language models, the application of SAEs to diffusion models remains largely unexplored. Recent work Surkov et al. [30] leverages SAEs and discovers highly interpretable concepts in the activations of a distilled DM [27]. While the results are promising, the paper focuses on a single-step diffusion model, and thus the time-evolution of visual features, a key characteristic and major source of intrigue around the inner workings of DMs, is not captured in their work.

In this paper, we aim to bridge this gap by addressing the following key questions:

- What kind of visual concepts are present in the early, 'chaotic' stage of the generative process?
- How do visual representations evolve through various stages of the generative process?
- Can we harness the uncovered concepts to steer the generative process in an interpretable way?

We perform extensive experiments on the features of a large-scale text-to-image diffusion model, Stable Diffusion

*Equal contribution.

Figure 1. General scene layout emerges during the very first generation step in diffusion models.

v1.4 [24], and extract thousands of interpretable concepts via SAEs. Strikingly, we find that the general composition of the image emerges *even before the first reverse diffusion step*, at which stage the model output carries no identifiable visual information (see Figure 1). While image composition is mostly finalized by the middle of the reverse process, we find that image style is still subject to change. Moreover, we demonstrate that intervening on the discovered concepts has interpretable, causal effect on the generated output image. We can manipulate image layout at early time steps, and influence image style at middle time steps. Our work sheds light on the evolution of visual representations in text-to-image diffusion models and opens the door to powerful, time-adaptive editing techniques.

2. Method

Training Sparse Autoencoders– We opt for k-sparse autoencoders with TopK activation given their success with GPT-4 [7] and SDXL Turbo [30]. Exact parametrization of the SAE and training objective can be found in Appendix A.

We focus on Stable Diffusion v1.4 (SDv1.4) [24] as our diffusion model due to its widespread use. To extract activations for SAE training, we sample 1.5M training prompts from the LAION-COCO dataset [28] and store updates made by the cross-attention transformer blocks to the residual stream while generating the corresponding images.

Throughout this paper, we assume that the diffusion process is parameterized by a continuous variable $t \in [0, 1]$, where $t = 1$ corresponds to pure noise distribution and $t = 0$ corresponds to the distribution of clean images. To capture the time-evolution of concepts, we collect activations at time steps corresponding to $t \in [0, 0.5, 1.0]$ and analyze *late* ($t = 0$), *middle*, and *early* ($t = 1.0$) diffusion dynamics respectively. For each time step t, we target 3 different cross-attention blocks in the denoising model of SDv1.4 , which we refer to as `down_block, mid_block, up_block`. We specifically include the `mid_block` or the bottleneck layer of the U-Net since earlier work found interpretable editing directions here [16]. Other blocks are chosen to be the closest to the bottleneck layer in the encoder and decoder paths. Note that, in SDv1.4 the performance of the text guidance is improved through Classifier-Free Guidance (CFG) [9], where the score is modified as $\tilde{\varepsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c}) = \varepsilon_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c}) + \omega\left(\varepsilon_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c}) - \varepsilon_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \varnothing)\right)$ where ω denotes the guidance scale, $\boldsymbol{c}$ is the conditioning input and $\varnothing$ is the null-text prompt. Therefore, we collect both the text-conditioned diffusion features (called `cond`) and null-text-conditioned features (denoted with `uncond`).

We train separate SAEs for different `block`, `conditioning`, `timestep` combinations to reconstruct individual feature vectors along the spatial dimension. Training results are in Appendix B. We focus on the `cond` features, as we hypothesize that such features may be more aligned with human-interpretable concepts due to the direct influence of language guidance through cross-attention (more on this in Appendix E).

Interpreting SAE features– We represent each concept with an associated list of objects, constituting a *concept dictionary*. The keys are unique concept identifiers (CIDs) assigned to each of the concept vectors of the SAE. The values correspond to objects that commonly occur in areas where the concept is activated. We leverage a zero-shot vision pipeline to annotate generated images with semantic segmentation masks, and associate concepts with a given object label if the concept's activation pattern sufficiently overlaps with the object's mask. In the interest of space, we detail how we build the concept dictionary in Appendix C.

Predicting image composition from SAE features– Suppose that we would like to predict the location of a particular object in the final generated image, but before the reverse diffusion process is completed. First, given SAE features from a given intermediate time step, we extract the top activating concepts for each spatial location. Next, we create a *conceptual map* of the image by assigning a word embedding to each spatial location based on our curated concept dictionary. Given a concept we would like to localize, such as an object from the input prompt, we produce a target word embedding and compare its similarity to each spatial location in the conceptual map. Finally, we assign the target concept to spatial locations with high similarity. This technique can be applied to each object present in the input prompt (or to any concepts of interest) to predict the composition of the final generated image. A visual overview of the method is included in Appendix D.

3. Experiments

We provide experimental details on curating the concept dictionary in Appendix C.1. A qualitative analysis of discovered concept categories is deferred to Appendix G.

Emergence of image composition– We investigate how image composition emerges and evolves in the internal representations of the diffusion model. We sample $5k$ LAION-COCO test prompts, and generate corresponding images with SDv1.4. Then, we follow the methodology described in Section 2 to predict a segmentation mask for every noun

in the input prompt using SAE features at various stages of diffusion. We evaluate the mean IoU between the predicted masks and ground truth annotations from our labeling pipeline at various time steps. Numerical results are summarized in Figure 2a.

First, we surprisingly find that the image composition emerges during the very first reverse diffusion step (even before the first complete forward pass!), as we are able to predict the rough layout of the final scene with $IoU \approx 0.26$ from `mid_block` SAE activations. As Figure 1 demonstrates, the general location of objects from the input prompt is already determined at this stage, even though the model output (posterior mean prediction) does not contain any visual clues about the final generated scene yet. More examples can be seen in the second column of Figure 2b.

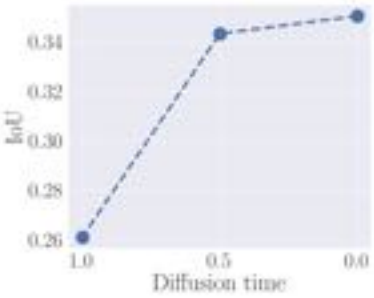
Second, we observe that the image composition and layout is mostly finalized by the middle of the reverse diffusion process ($t = 0.5$), which is supported by the saturation in the accuracy of predicted masks. Visually, predicted masks for $t = 0.5$ and $t = 0.0$ look similar, however we see indications of increasing semantic granularity in represented concepts. For instance, the second row in Figure 2b depicts predicted segmentation masks for the noun *church*. Even though the masks for $t = 0.5$ and $t = 0.0$ are overall similar, the mask in the final time step excludes doors and windows on the building, suggesting that those regions are assigned more specific concepts, such as *door* and *window*. We note that segmentation IoU is evaluated with respect to our zero-shot annotations, which are often *less accurate* than our predicted masks for $t = 0.0$, and thus the reported IoU is bottlenecked by the quality of our annotations.

Finally, we find that image composition can be extracted from any of the investigated blocks, and thus we do not observe strong specialization between these layers for composition-related information. However, `up_block` provides more accurate segmentations than `down_block`, and `mid_block` provides the lowest due to the lower spatial resolution (more details in Appendix E).

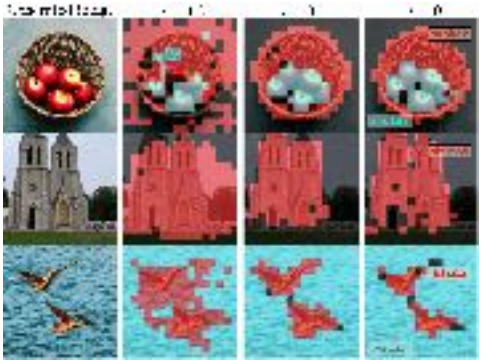
Causal interventions – Dataset examples that activate a particular concept provide only *correlational* interpretation. Here, we investigate *causal* effects on generated images by observing the results of directly manipulating SAE concepts.

Image composition. To assess whether one can control the composition of the generated image using our discovered concepts, we propose a simple task: enforce a specific object to appear only in a designated quadrant (e.g., top-left) of the image. To achieve this, we intercept `mid_block` activations of the diffusion model and edit them in the latent space of the SAE by amplifying the strength of the target concept in the first 40% of reverse diffusion. An overview of this intervention can be seen in Figure 3. More details can be found in Appendix F.1.

In Figure 4, we consider *bee*, *book*, and *dog* as the objects



(a) Evolution of predicted image composition accuracy (in terms of IoU) over the reverse diffusion process (`mid_block`).



(b) Visualization of segmentation maps predicted from extracted concepts across reverse diffusion steps (`up_block`).

Figure 2. Evolution of predicted image composition during the reverse diffusion process, shown through segmentation accuracy (top) and visualizations (bottom).

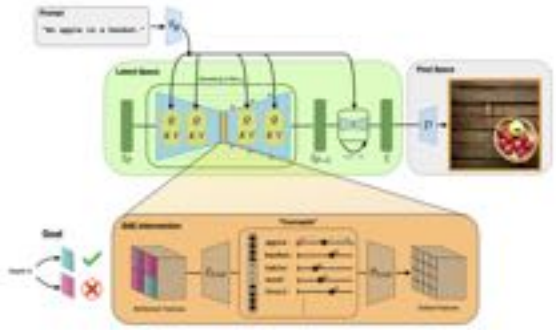


Figure 3. Overview of our SAE interventions.

of interest. We show the generated images after the intervention for four different quadrants: `top-left`, `top-right`, `bottom-left`, and `bottom-right`. In order to find the CIDs, we sweep through the concept dictionary and collect all the CIDs where the word of interest appears. We observe that the objects of interest are successfully guided to their respective locations. Moreover, the concepts that we do not intervene on, such as the flower in the first row are preserved.

Image style. Next, we investigate whether SAE features can be altered to influence the style of the image. To this end, given a CID related to the style of interest,

Figure 4. Intervening with the spatial composition of images. We restrict all the concepts related to the object/noun to appear only in a particular quadrant of the image. We use the SAE trained on `mid_block` activations at $t = 1.0$.

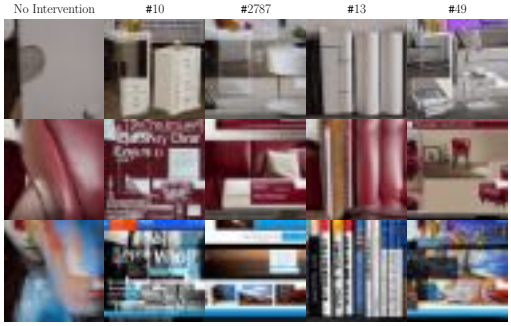


Figure 5. Example interventions at the early stages of the reverse diffusion. We intervene on activations at $t = 1.0$.

we modify the activation at each spatial location during the first 40% of reverse diffusion (details in Appendix F.2). Through our concept dictionary and visual inspection of top dataset examples, we identify four CIDs that seem to have a consistent style: `#10 - comic book cover, poster`, `#2787 - computer screen`, `#13 - books`, and `#49 - game art/style`. Top activating images for each of these concepts can be found in Appendix I. We depict examples of the generated images after intervention in Figure 5. We crucially observe that for a fixed style c, the generated images have similar patterns of lines, directions of edges, image gradients, etc. rather than *artistic styles*. Therefore, we hypothesize that during the earlier stages of reverse diffusion, concepts are more aligned with low-level image features rather than high-level semantics. For instance, `#13 - books` may be more related to *book-looking* objects rather than images of actual books.

In an effort to control artistic style, we turn our attention to the middle stages ($t \approx 0.5$) of diffusion, and intervene on activations for $t \in [0.3, 0.6]$ using our SAE trained on $t = 0.5$ features. Through our concept dictionary and visual inspection of top dataset examples, we identify CIDs

Figure 6. Example interventions at the middle stages of the reverse diffusion. We intervene on activations at $t = 0.5$.

#1314 that controls the *cartoon* look of images, #524 appears mostly with beach images where *sea* and *sand* are visible together, and #2137 activates the most on *paintings* (top activating images can be found in Appendix I). We provide examples of the generated images in Figure 6. Interestingly, we observe that the interventions do not alter the layout of the image as before. Instead, we observe changes in the texture (cartoon look, sandy texture, smooth straight lines, etc.) and local edits more aligned with *artistic* styles. Contrasting this with $t = 1.0$ edits, we conclude that middle time steps are responsible for more high-level artistic and textural edits where the image layout is already determined in the earlier time steps (also evident from our semantic segmentation experiments).

4. Conclusions and limitations

In this paper, we take a step towards demystifying the inner workings of text-to-image diffusion models under the lens of mechanistic interpretability, with an emphasis on understanding how visual representations evolve over the generative process. We show that the semantic layout of the image emerges as early as the first reverse diffusion step and can be predicted surprisingly well from our learned features, even though no coherent visual cues can be identified in the model outputs at this stage yet. As reverse diffusion progresses, the semantic layout becomes progressively more refined, and the image composition is largely finalized by the middle of the reverse trajectory. Furthermore, we conduct feature intervention experiments to demonstrate that the learned SAE features can be leveraged to control the generation process. Our experiments suggest that concepts discovered at early stages of diffusion are related to image composition and low-level visual features, whereas the middle stages are responsible for higher-level concepts such as artistic style. Developing editing techniques that adapt to the evolving nature of diffusion representations is a promising direction for future work.

5. Acknowledgements

We would like to thank Microsoft for an Accelerating Foundation Models Research grant that provided the OpenAI credits enabling this work. This research is also in part supported by AWS credits through an Amazon Faculty research award and a NAIRR Pilot award. M. Soltanolkotabi is also supported by the Packard Fellowship in Science and Engineering, a Sloan Research Fellowship in Mathematics, an NSF-CAREER under award #1846369, and NSF-CIF awards #1813877 and #2008443. and NIH DP2LM014564-01.

References

[1] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5343–5353, 2024.

[2] Siyi Chen, Huijie Zhang, Minzhe Guo, Yifu Lu, Peng Wang, and Qing Qu. Exploring low-dimensional subspaces in diffusion models for controllable image editing, 2024.

[3] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023.

[4] Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. *arXiv preprint arXiv:2105.05233*, 2021.

[5] Dave Epstein, Allan Jabri, Ben Poole, Alexei A. Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation, 2023.

[6] Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models, 2024.

[7] 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, 2024.

[8] René Haas, Inbar Huberman-Spiegelglas, Rotem Mulayoff, Stella Graßhof, Sami S. Brandt, and Tomer Michaeli. Discovering interpretable directions in the semantic latent space of diffusion models, 2024.

[9] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *arXiv preprint arXiv:2006.11239*, 2020.

[11] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23(47):1–33, 2022.

[12] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

[13] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's" up" with vision-language models? investigating their struggle with spatial reasoning. *arXiv preprint arXiv:2310.19785*, 2023.

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the*

IEEE/CVF international conference on computer vision, pages 4015–4026, 2023.

[15] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

[16] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space, 2023.

[17] Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13872–13882, 2024.

[18] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.

[19] Alex Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. *arXiv preprint arXiv:2102.09672*, 2021.

[20] Chris Olah. Mechanistic interpretability, variables, and the importance of interpretable bases. `https://www.transformer-circuits.pub/2022/mech-interp-essay`, 2022. Accessed: 2025-04-14.

[21] Hadas Orgad, Bahjat Kawar, and Yonatan Belinkov. Editing implicit assumptions in text-to-image diffusion models, 2023.

[22] Yong-Hyun Park, Mingi Kwon, Jaewoong Choi, Junghyo Jo, and Youngjung Uh. Understanding the latent space of diffusion models through the lens of riemannian geometry, 2023.

[23] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[25] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image Super-Resolution via Iterative Refinement. *arXiv:2104.07636 [cs, eess]*, 2021.

[26] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[27] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024.

[28] Christoph Schuhmann, Andreas Kopf, Richard Vencu, Theo Coombes, Romain Beaumont, and Benjamin Trom. Laion coco: 600m synthetic captions from laion2b-en.

[29] Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. *arXiv:1907.05600 [cs, stat]*, 2020.

[30] Viacheslav Surkov, Chris Wendler, Mikhail Terekhov, Justin Deschenaux, Robert West, and Caglar Gulcehre. Unpacking sdxl turbo: Interpreting text-to-image models with sparse autoencoders, 2024.

[31] Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the daam: Interpreting stable diffusion using cross attention, 2022.

[32] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578, 2024.

[33] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23174–23184, 2023.

[34] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1724–1732, 2024.

Appendix

A. SAE architecture and loss

Let $\boldsymbol{x} \in \mathbb{R}^d$ denote the input to the autoencoder that we want to decompose into a sparse combination of features. Then, we obtain the latent $\boldsymbol{z} \in \mathbb{R}^{n_f}$ by encoding $\boldsymbol{x}$ as

$$\boldsymbol{z} = \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \text{TopK}\left(\text{ReLU}\left(\boldsymbol{W}_{enc}\left(\boldsymbol{x} - \boldsymbol{b}\right)\right)\right),$$

where $\boldsymbol{W}_{enc} \in \mathbb{R}^{n_f \times d}$ denotes the learnable weights of the encoder, $\boldsymbol{b} \in \mathbb{R}^d$ is a learnable bias term, and TopK function keeps the top k highest activations and sets the remaining ones to 0. Note, that due to the superposition hypothesis, we want the encoding to be expansive and therefore $n_f >> d$. Then, a decoder is trained to reconstruct the input from the latent $\boldsymbol{z}$ in the form

$$\hat{\boldsymbol{x}} = \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{z}) = \boldsymbol{W}_{dec}\boldsymbol{z} + \boldsymbol{b},$$

where $\boldsymbol{W}_{dec} \in \mathbb{R}^{d \times n_f}$ represents the learnable weights of the decoder. Note that the bias term is shared between the encoder and the decoder. We refer to $\boldsymbol{f}_i = \boldsymbol{W}_{dec}[:, i]$ columns of $\boldsymbol{W}_{dec}$ as *concept vectors*. We obtain the learnable parameters by optimizing the reconstruction error

$$\mathcal{L}_{rec}\left(\boldsymbol{W}_{enc}, \boldsymbol{W}_{dec}, \boldsymbol{b}\right) = \mathcal{L}_{rec}\left(\boldsymbol{\theta}\right) = \left\|\boldsymbol{x} - \hat{\boldsymbol{x}}\right\|_2^2.$$

In practice, training only based on the reconstruction error is insufficient due to the emergence of dead features. Dead features are defined as directions in the latent space that are not activated for some specified number of training iterations resulting in wasted model capacity and compute. To resolve this issue, Gao et al. [7] proposes an auxiliary loss AuxK that models the reconstruction error of the SAE using the top-k_{aux} dead features. To be specific, define the reconstruction error as $\boldsymbol{e} = \boldsymbol{x} - \hat{\boldsymbol{x}}$, then the auxiliary loss takes the form,

$$\mathcal{L}_{aux}\left(\boldsymbol{\theta}\right) = \left\|\boldsymbol{e} - \hat{\boldsymbol{e}}\right\|_2^2$$

where $\hat{\boldsymbol{e}} = \boldsymbol{W}_{dec}\boldsymbol{z}$ approximates the reconstruction error with the top-k_{aux} dead latents. The combined loss for the SAE training becomes:

$$\mathcal{L}\left(\boldsymbol{\theta}\right) = \mathcal{L}_{rec}\left(\boldsymbol{\theta}\right) + \alpha\mathcal{L}_{aux}\left(\boldsymbol{\theta}\right)$$

where α is a small mixing coefficient. We set $\alpha = \frac{1}{32}$ and $k_{aux} = 256$ in our experiments. We set the latent expansion factor as 4, i.e. $n_f = 4d$.

B. Results on SAE training

We train SAEs on the residual updates in the diffusion U-Net blocks `down_blocks.2.attentions.1`, `mid_block.attentions.0`, `up_blocks.1.attentions.0`, referred to as `down_block`, `mid_block`, `up_block`. We keep track of normalized mean-squared error (MSE) and explained variance of the SAE reconstructions. In Tab. 1 we provide the complete set of training metrics for all combinations of `block`, `conditioning`, `timestep`, and k.

C. Building the concept dictionary

Multiple work on automatic labeling of SAE features resort to LLM pipelines where the captions corresponding to top activating dataset examples are collected and the LLM is prompted to summarize them. However, these approaches come with severe shortcomings. First, they may incorporate the biases and limitations of the language model into the concept labels, including failures in spatial reasoning [13], object counting, identifying structural characteristics and appearance [32] and object hallucinations [17]. Second, they are sensitive to the prompt format and phrasing, and the instructions may bias or limit the extracted concept labels. Last but not least, it is computationally infeasible to scale LLM-based concept summarization to a large number of images, limiting the reliability of extracted concepts. For instance, [30] only leverages a few dozens of images to define each concept. Therefore, we opt for designing a scalable approach that obviates the need for LLM-based labeling and instead use a vision-based pipeline to label our extracted SAE features.

Table 1. Performance metrics for different block types, timesteps, conditioning and k values. Best metrics for each conditioning block are underlined. Best overall metrics are **bold**.

Conditioning	Block	Timestep (t)	k	Scaled MSE	Explained Variance (%)
cond	down_block	0	10	0.6293	36.6
			20	0.5466	44.8
		0.5	10	0.6275	37.6
			20	0.5510	45.1
		1.0	10	0.4617	51.7
			20	0.3767	60.5
	mid_block	0	10	0.4817	50.5
			20	0.4133	57.3
		0.5	10	0.4802	50.9
			20	0.4194	57.0
		1.0	10	0.4182	56.4
			20	0.3503	63.3
	up_block	0	10	0.5540	44.0
			20	0.4698	52.5
		0.5	10	0.5414	45.3
			20	0.4648	52.9
		1.0	10	0.4177	57.7
			20	0.3424	65.3
uncond	down_block	0	10	0.6306	36.4
			20	0.5477	44.6
		0.5	10	0.6364	36.9
			20	0.5580	44.5
		1.0	10	0.3874	58.6
			20	0.3081	66.9
	mid_block	0	10	0.4852	50.7
			20	0.4161	57.6
		0.5	10	0.4909	50.8
			20	0.4277	57.0
		1.0	10	0.3286	65.7
			20	0.2613	72.6
	up_block	0	10	0.5550	44.0
			20	0.4701	52.5
		0.5	10	0.5436	45.3
			20	0.4653	53.3
		1.0	10	0.2724	71.4
			20	**0.2115**	**77.7**

In particular, we represent each concept by an associated list of objects, constituting a *concept dictionary*. The keys are unique concept identifiers (CIDs) assigned to each of the concept vectors of the SAE. The values correspond to objects that commonly occur in areas where the concept is activated. To build the concept dictionary (Figure 7), we first sample a set of text prompts, generate the corresponding images using a diffusion model and extract the SAE activations for each CID during generation. We obtain ground truth annotations for each generated image using a zero-shot pipeline, that combines image tagging, object detection and semantic segmentation, resulting in a mask and label for each object in generated images. Finally, we evaluate the alignment between our ground truth masks and the SAE activations for each CID, and assign the

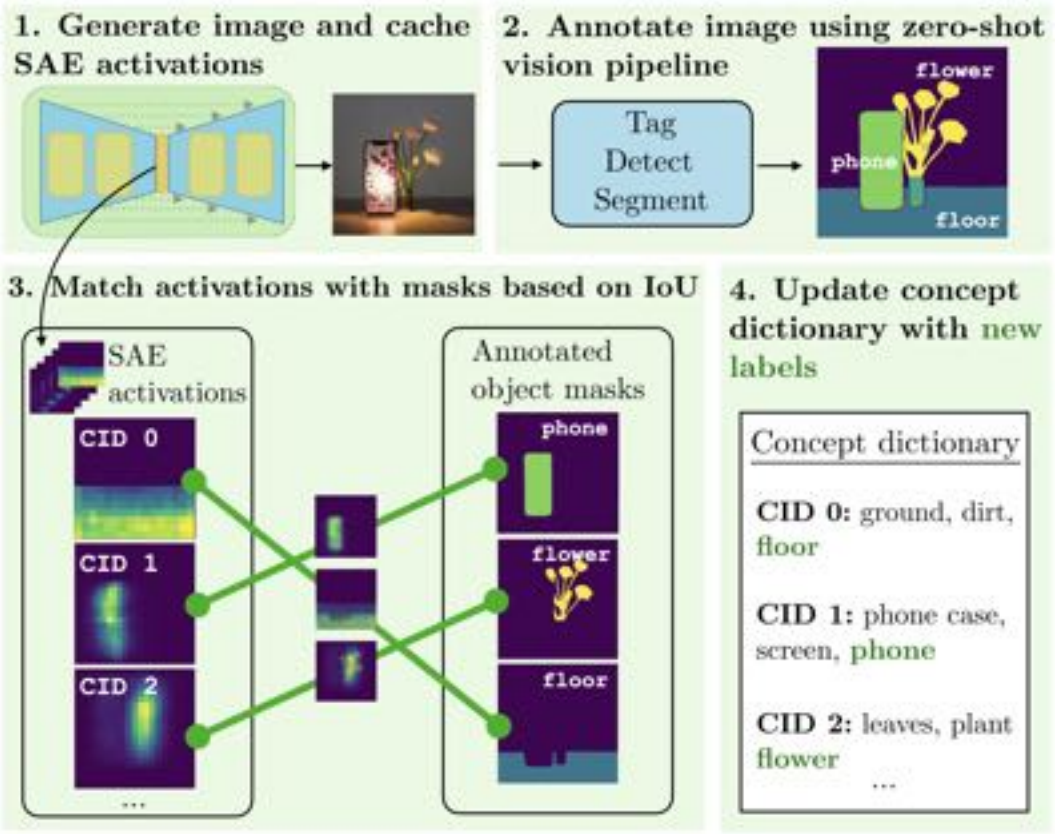


Figure 7. Curating the concept dictionary: 1) We cache SAE activations for various time steps and blocks during image generation. 2) We leverage a pipeline of image tagging, open-set object detection and promptable segmentation to annotate the generated image with segmentation masks and corresponding object labels. 3) We find SAE activations that sufficiently overlap with the object masks. 4) We add the overlapping object's label to the concept dictionary under the matching SAE activation's CID.

corresponding label to the CID only if there is sufficient overlap.

The concept dictionary represents each concept with a list of objects. In order to provide a more concise summary that incorporates semantic information, we assign an embedding vector to each concept. In general, we could use any model that provides robust natural language embeddings, such as an LLM, however we opt for a simple approach by assigning the mean Word2Vec embedding of object names activating the given concept.

C.1. Experimental details

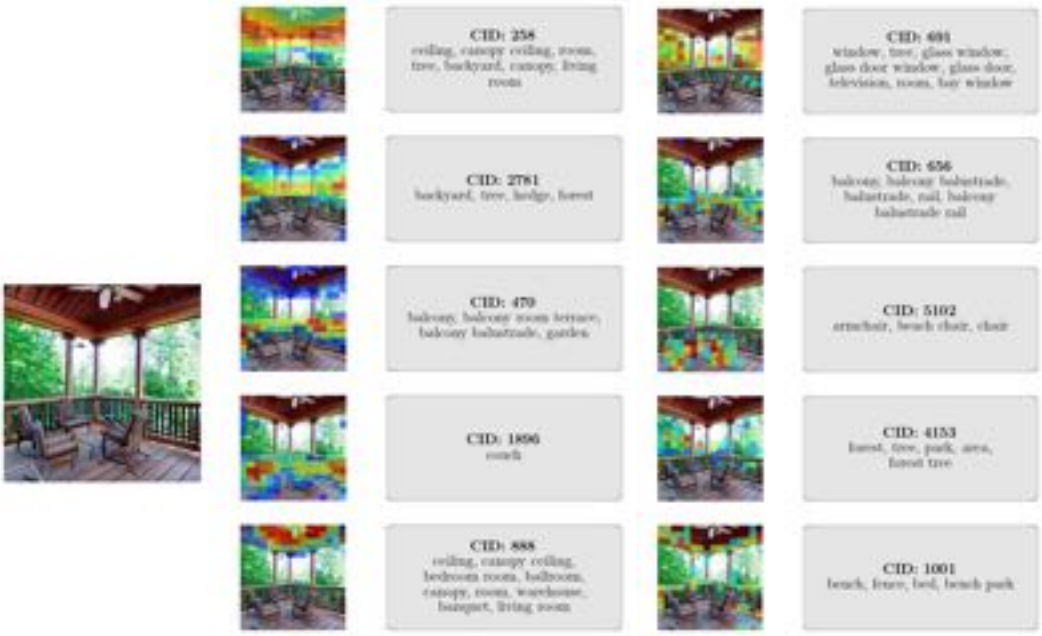
In all experiments, we leverage RAM [34] for image tagging, Grounding DINO [18] for open-set object detection and SAM [14] for segmentation in our zero-shot annotation pipeline, following [23].

We sample $40k$ prompts from the LAION-COCO dataset from a split that has not been used to train the SAEs. We assign a label to a specific CID if the IoU between the corresponding annotated mask and activation is greater than 0.5. We binarize the activation map for the IoU calculation by first normalizing to $[0, 1]$ range, then thresholding at 0.1.

We depict top 5 activating concepts, extracted from `up_blocks.1.attentions.0`, for generated images and their corresponding concept dictionary entries in Figures 8 - 10.

D. Predicting final image composition

Leveraging the concept dictionary, we predict the final image composition based on SAE features at any time step (Figure 11), allowing us to gain invaluable insight into the evolution of image representations in diffusion models. Suppose that we would like to predict the location of a particular object in the final generated image, but before the reverse diffusion process is completed. First, given SAE features from a given intermediate time step, we extract the top activating concepts for each



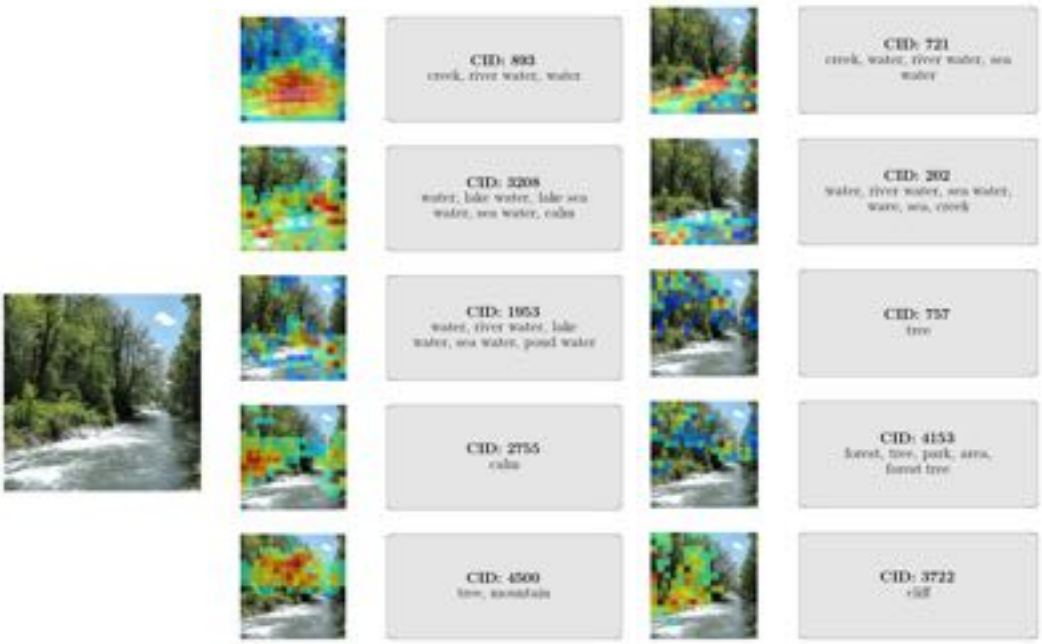
(a) $t = 1.0$ (b) $t = 0.0$

Figure 8. Concept dictionary and visualization of the activation map for the top 5 activating concepts. Sample ID: `2000031`

(a) $t = 1.0$ (b) $t = 0.0$

Figure 9. Concept dictionary and visualization of the activation map for the top 5 activating concepts. Sample ID: `2000061`

spatial location. Next, we create a *conceptual map* of the image by assigning a word embedding to each spatial location based on our curated concept dictionary. This conceptual map shows how image semantics, described by localized word embeddings, vary spatially across the image. Given a concept we would like to localize, such as an object from the input prompt, we produce a target word embedding and compare its similarity to each spatial location in the conceptual map. To produce a predicted segmentation map, we assign the target concept to spatial locations with high similarity, based on a pre-defined



(a) $t = 1.0$ (b) $t = 0.0$

Figure 10. Concept dictionary and visualization of the activation map for the top 5 activating concepts. Sample ID: `2000062`

threshold value. This technique can be applied to each object present in the input prompt (or to any concepts of interest) to predict the composition of the final generated image.

E. Additional results on segmentation accuracy

We provide a comprehensive overview of the accuracy of predicted segmentations across different architectural blocks in SDv1.4 in Figure 12.

We find that coarse image composition can be extracted from any of the investigated blocks, and from both `cond` and `uncond` features even in the first reverse diffusion step. We consistently observe saturation by the middle of the reverse diffusion trajectory. We note that the saturation is partially due to imperfect ground truth masks from our annotation pipeline that can be *less* accurate than the masks obtain from the SAE features at late time steps. Overall, `up_block` provides the most accurate, and `mid_block` the least accurate segmentations (due to the lower spatial resolution in the bottleneck). We observe consistently lower segmentation accuracy based on `uncond` features. We hypothesize that `uncond` features may encode more low-level visual information, whereas `cond` features are directly influenced by the text conditioning and therefore represent more high-level semantic information.

F. Details of causal interventions

F.1. Manipulating image composition

We intercept `mid_block` activations of the diffusion model and edit them in the latent space of the SAE. Recall that the contribution of the bottleneck transformer block at time t is given by $\Delta_{mid,t} \in \mathbb{R}^{H \times W \times d}$. Let $\boldsymbol{Z}_{mid,t} \in \mathbb{R}^{H \times W \times n_f}$ denote the latents after encoding the activations with the SAE encoder $\mathcal{E}_{\boldsymbol{\theta}}$. Let S denote the set of coordinates to which we would like to restrict the object. Let C_o be the set of CIDs that are relevant to object o. We wish to modifty the latents as follows:

$$\forall c \in C_o, \;\; \tilde{\boldsymbol{Z}}_{mid,t}[i,j,c] = \begin{cases} \beta, & \text{if } (i,j) \in S \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where β is our intervention strength. However, decoding the modified latents directly is suboptimal as the SAE cannot reconstruct the input perfectly. Instead, we modify the activations directly using the concept vectors. The modification in

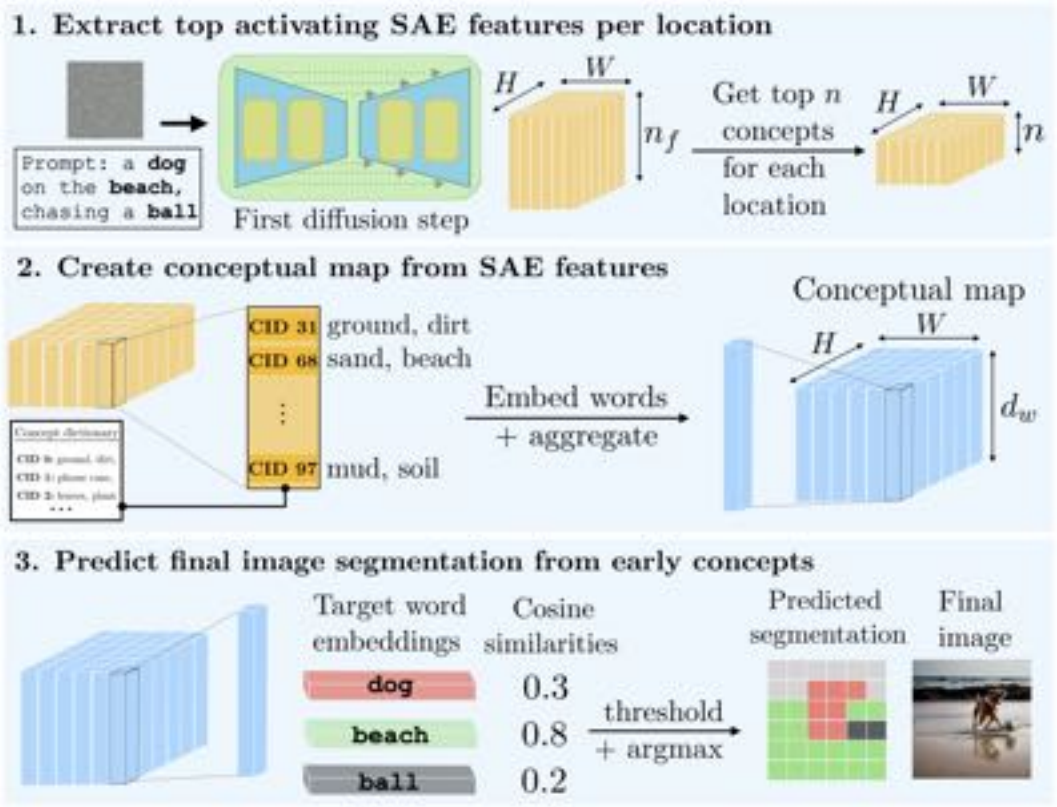
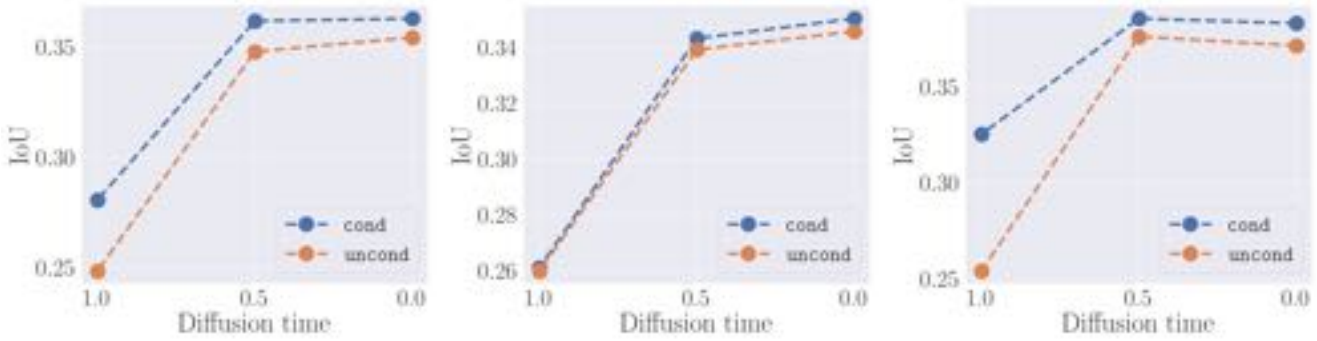


Figure 11. Predicting image composition: 1) We cache SAE activations during the *very first* diffusion step (or other time step of interest) and extract top activated concepts for each spatial location. 2) For each spatial location, we fetch the associated objects from the concept dictionary and produce a conceptual embedding via Word2Vec. 3) We compare the conceptual embedding at each location to the target word embeddings from the input prompt and predict a segmentation map based on cosine similarity.



(a) `down_block` (b) `mid_block` (c) `up_block`

Figure 12. Accuracy of predicted segmentations based on SAE features from different architectural blocks. `cond` stands for text-conditioned diffusion features, and `uncond` denotes null-prompt conditioning.

Equation (1) can be equivalently written as:

$$\tilde{\Delta}_{mid,t}[i,j] = \begin{cases} \Delta_{mid,t}[i,j] + \beta \sum_{m \in C_o} \boldsymbol{f}_m & \text{if } (i,j) \in S \\ \Delta_{mid,t}[i,j] - \sum_{m \in C_o} \boldsymbol{f}_m, & \text{otherwise} \end{cases}. \tag{2}$$

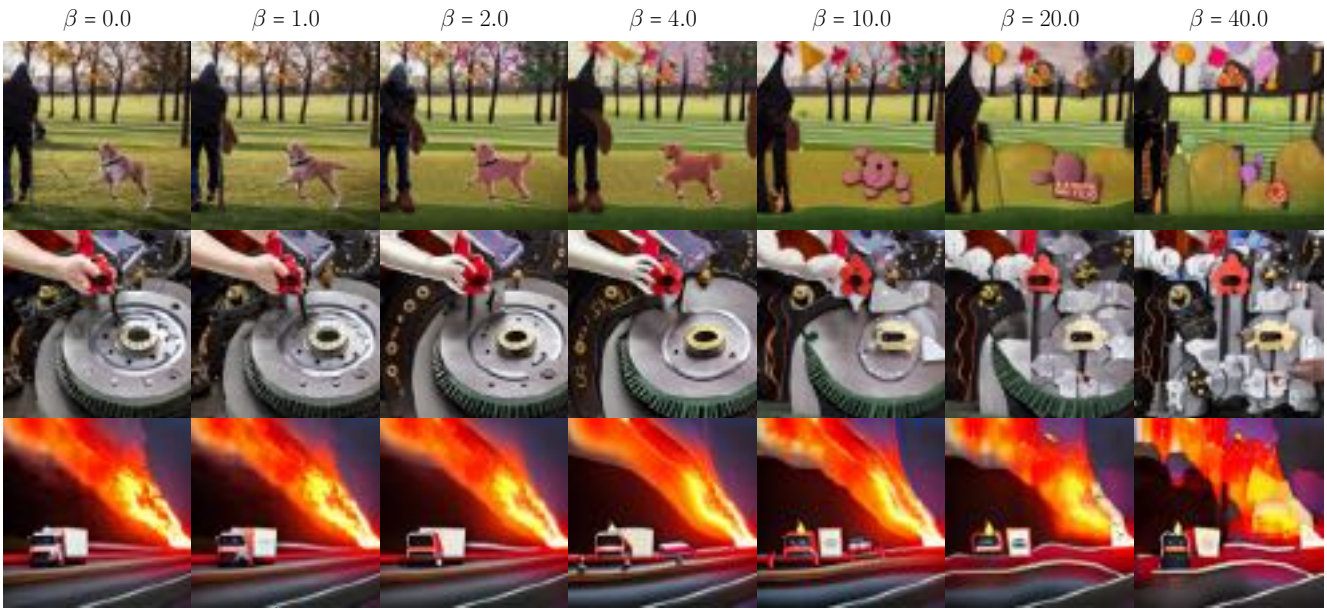


Figure 13. Effect of the intervention strength β. We intervene the concept #1314 (controlling *cartoon* look) of the SAE trained on `mid_block` activations at $t = 0.5$. Prompts corresponding to images are drawn randomly from the validation split. From top to bottom prompts are: `'A man playing with his dog in the park.', 'The flywheel is being used to make an automatic clutch.', 'Cars and trucks driving on the highway with flames in the background.'`

In our experiments, we observe that the same intervention strength β does not work well across different objects o. To solve this, we introduce a normalization where the intervention at a spatial coordinate (i, j) is proportional to the norm of the latent at that coordinate $\|\boldsymbol{Z}_{mid,t}[i, j]\|$. Therefore, the effective intervention strength is $\beta_{ij} = \beta \|\boldsymbol{Z}_{mid,t}[i, j]\|$. We apply Equation (2) for 40% of the reverse diffusion ($t \in [0.6, 1.0]$) using the SAE trained on `cond` activations of `mid_block` at $t = 1.0$. Although the SAE is trained only on a particular timestep, we observe that it generalizes to other timesteps in the vicinity. We empirically observe that a significantly large value of $\beta = 4000$ is needed to successfully control the spatial composition consistently. We hypothesize that the skip connections in the U-Net architecture and the features from the `null-text` conditioning in classifier-free guidance reduce the effect of our interventions, as they provide paths that bypass the intervention. Thus, a larger value of intervention strength is needed to mask the leakage effects.

F.2. Manipulating image style

Given a CID c related to the style of interest, we modify the activation at each spatial location as follows:

$$\tilde{\Delta}_{mid,t}[i, j] = \Delta_{mid,t}[i, j] + \beta \boldsymbol{f}_c. \tag{3}$$

Similar to Appendix F.1, we find that a standardization is necessary for β to work well across different choice of styles. We let β to be adaptive to spatial locations and modify them as $\tilde{\beta}_{ij} = \frac{\|\boldsymbol{Z}_{mid,t}[i,j]\|}{\sum_{i,j}\|\boldsymbol{Z}_{mid,t}[i,j]\|}\beta$ to alleviate this.

First, we target intervening during the first 40% of the reverse diffusion trajectory that corresponds to $t \in [0.6, 1.0]$. With the new standardization, we find $\beta = 8.0$ to work well after a grid search. Next, we shift the intervention interval to the middle stages of diffusion ($t \in [0.3, 0.6]$). We find $\beta = 10.0$ to work well in general. Figure 13 provides more intuition on the effect of intervention strength β from Eq. (3) on controlling image style

G. Qualitative assessment of activations

We visualize the activation maps for top 10 (in terms of mean activation across the spatial dimensions) activating concepts for generated samples in Figures 14 - 16 for various time steps and blocks. Based on our empirical observations, the activations can be grouped in the following categories:

- **Local semantics** – Most concepts fire in semantically homogeneous regions, producing a semantic segmentation mask for a particular concept. Examples include the segmentation of the pavement, buildings and people in Figure 14, the plate, food

items and background in Figure 15 and the face, hat, suit and background in Figure 16. We observe that these semantic concepts can be redundant in the sense that multiple concepts often fire in the same region (e.g. see Fig. 15, second row with multiple concepts focused on the food in the bowl). We hypothesize that these duplicates may add different conceptual layers to the same region (e.g. *food* and *round* in the previous example). In terms of diffusion time, we observe that the segmentation masks are increasingly more accurate with respect to the final generated image, which is expected as the final scene progressively stabilizes during the diffusion process. This observation is more thoroughly verified in Section **??** and Figure 2a. In terms of different U-Net blocks, we observe that `up_blocks.1.attentions.0` provides the most accurate segmentation of the final scene, especially at earlier time steps.

- **Global semantics (style)** – We find concepts that activate more or less uniformly in the image. We hypothesize that these concepts capture global information about the image, such as artistic style, setting or ambiance. We observe such concepts across all studied diffusion steps and architectural blocks.
- **Context-free** – We observe that some concepts fire exclusively in specific, structured regions of the image, such as particular corners or bordering edges of the image,irrespective of semantics (see e.g. the last activation in the first row of Figure 14). We hypothesize that these concepts may be a result of optimization artifacts, and are leveraged as semantic-independent knobs for the SAE to reduce reconstruction error. Specifically, if the SAE is unable to "find" k meaningful concepts in the image, as encouraged by the training objective, it may compensate for the missing signal energy in these context-free directions.

H. Context-free activations

We observe the emergence of feature directions in the representation space of the SAE that are localized to particular, structured regions in the image (corners, vertical or horizontal lines) independent of high-level image semantics. We visualize examples in Figures 17 - 18. Specifically, we find concept IDs for which the variance of activations averaged across spatial dimensions is minimal over a validation split. We depict the mean and variance of such activations and showcase generated samples that activate the particular concept. We observe that these localized activation patterns appear throughout the generative process (both at $t = 1.0$ in Figure 17 and at $t = 0.0$ in Figure 18). Moreover, the retrieved activating samples typically do not share common semantic or low-level visual features, as demonstrated by the sample images. We hypothesize that these feature directions may be used by the SAE as "registers" for context-independent information.

I. Visualization of top dataset examples

Top dataset examples for a concept ID c is determined by sorting images based on their average concept intensity γ_c where the averaging is over spatial dimensions. Formally (definition is taken from [30]), for a transformer block ℓ and timestep t, we define γ_c as:

$$\gamma_c = \frac{1}{HW} \sum_{i,j} \boldsymbol{Z}_{\ell,t}[i, j, c].$$

In Figures 19 - 29, we provide top activated images for various concept IDs and for various timestep t's.

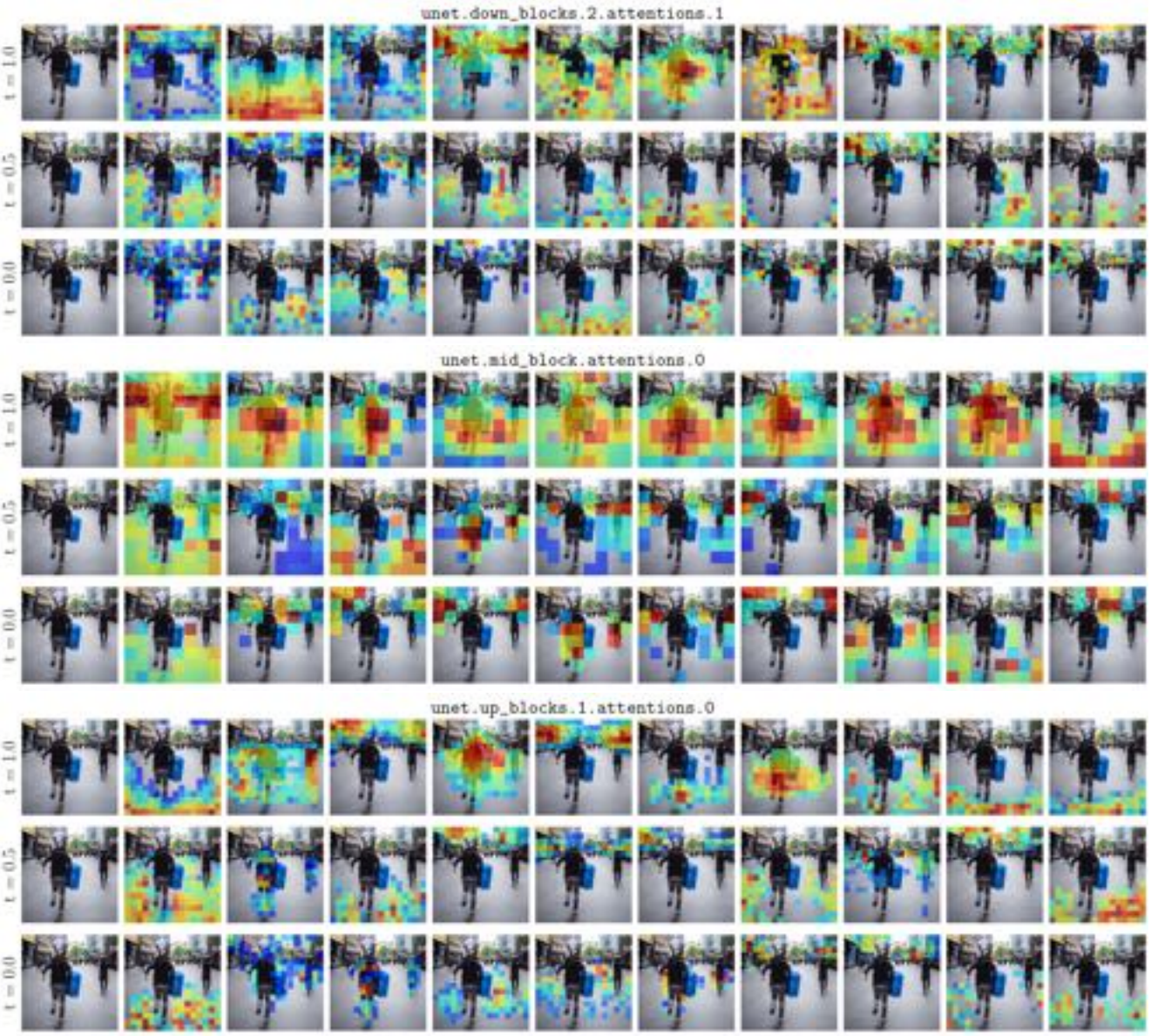


Figure 14. Visualization of top activating concepts in a generated sample. Concepts are sorted by mean activation across spatial locations and top 10 activation maps are shown. Each row depicts a different snapshot along the reverse diffusion trajectory starting from pure noise ($t = 1.0$) and terminating with the generated final image ($t = 0.0$). Note that each row within the same column may belong to a different concept, as concepts are not directly comparable across different diffusion time indices (separate SAE is trained for each individual timestep). Sample ID: `2000018`.

Figure 15. Visualization of top activating concepts in a generated sample. Concepts are sorted by mean activation across spatial locations and top 10 activation maps are shown. Each row depicts a different snapshot along the reverse diffusion trajectory starting from pure noise ($t = 1.0$) and terminating with the generated final image ($t = 0.0$). Note that each row within the same column may belong to a different concept, as concepts are not directly comparable across different diffusion time indices (separate SAE is trained for each individual timestep). Sample ID: 2000035.

Figure 16. Visualization of top activating concepts in a generated sample. Concepts are sorted by mean activation across spatial locations and top 10 activation maps are shown. Each row depicts a different snapshot along the reverse diffusion trajectory starting from pure noise ($t = 1.0$) and terminating with the generated final image ($t = 0.0$). Note that each row within the same column may belong to a different concept, as concepts are not directly comparable across different diffusion time indices (separate SAE is trained for each individual timestep). Sample ID: `2000042`.

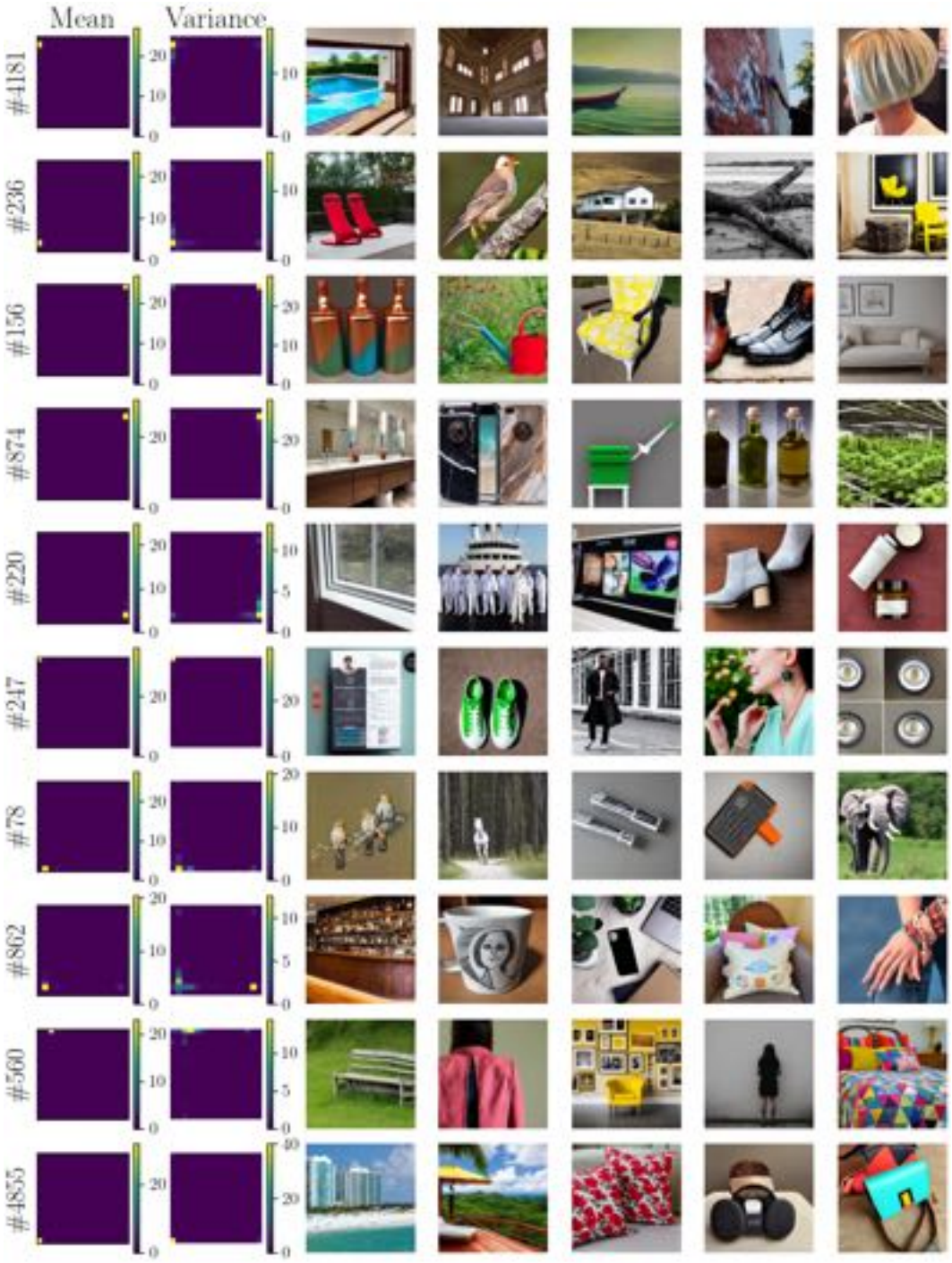


Figure 17. We plot the mean and variance of activations, extracted at $t = 1.0$, for concepts with lowest average variance across spatial locations. We find concepts that fire exclusively at specific spatial locations. We depict generated samples that maximally activate for the given concept.

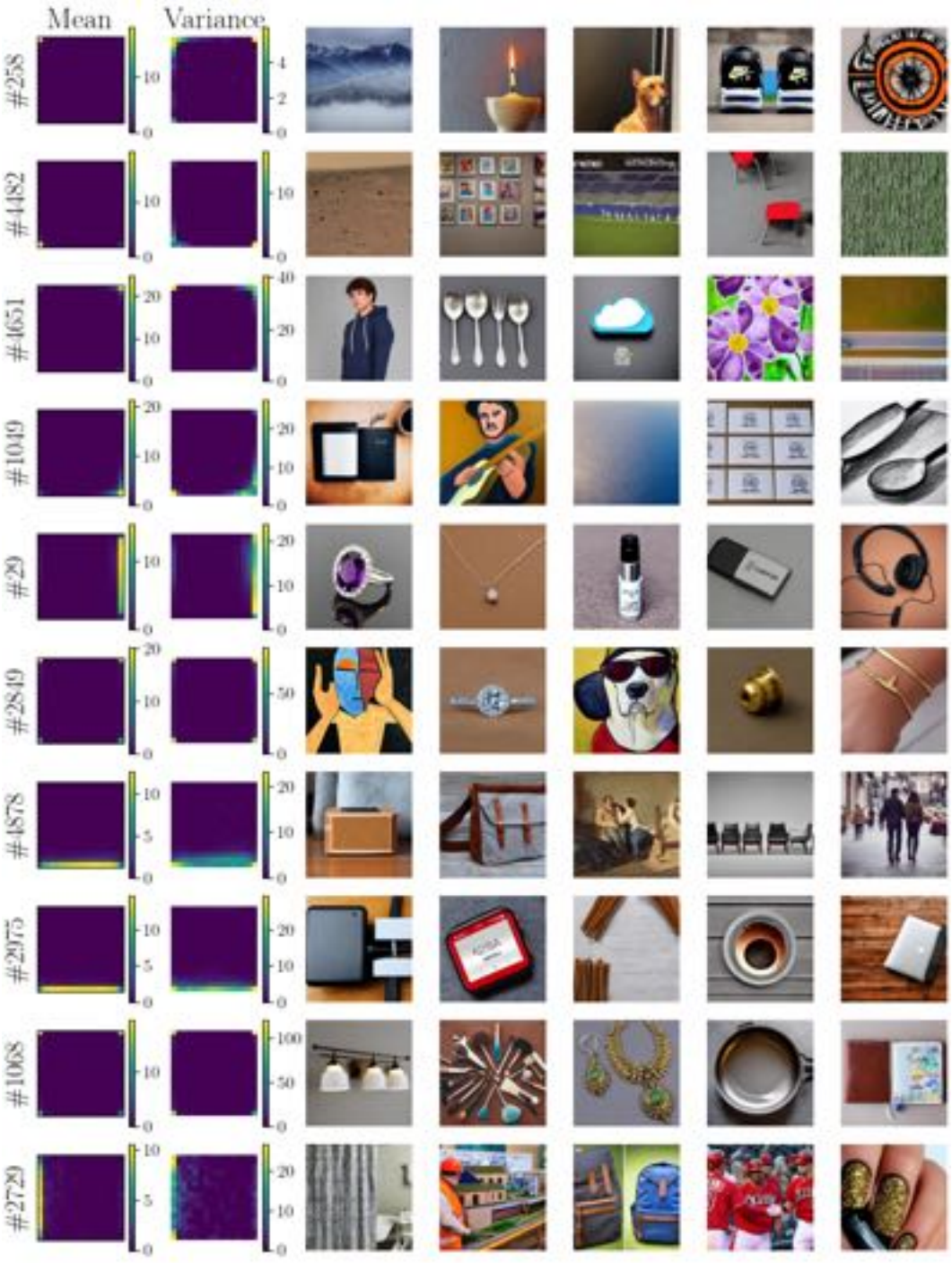


Figure 18. We plot the mean and variance of activations, extracted at $t = 0.0$, for concepts with lowest average variance across spatial locations. We find concepts that fire exclusively at specific spatial locations. We depict generated samples that maximally activate for the given concept.

Figure 19. Top activating dataset examples for the concept ID 10 belonging to the SAE trained on the `cond` activation of `mid_block` at $t = 1.0$.

Figure 20. Top activating dataset examples for the concept ID 13 belonging to the SAE trained on the `cond` activation of `mid_block` at $t = 1.0$.

Figure 21. Top activating dataset examples for the concept ID 49 belonging to the SAE trained on the `cond` activation of `mid_block` at $t = 1.0$.

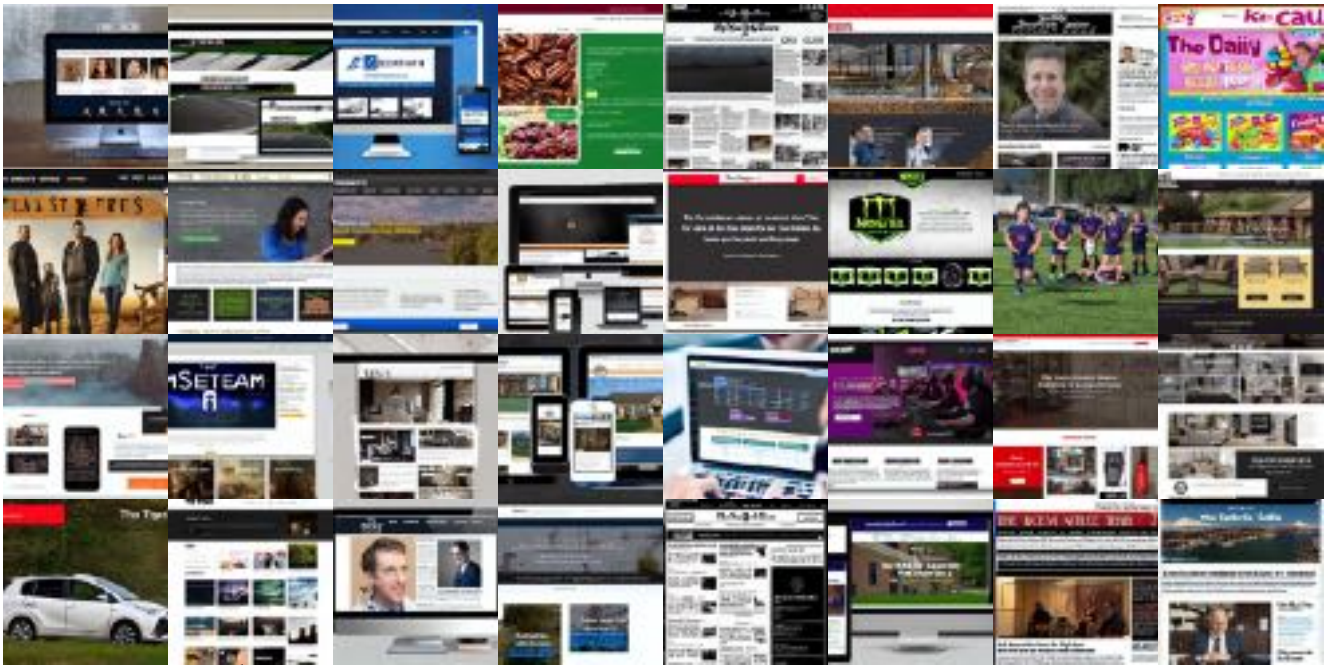


Figure 22. Top activating dataset examples for the concept ID 2787 belonging to the SAE trained on the `cond` activation of `mid_block` at $t = 1.0$.

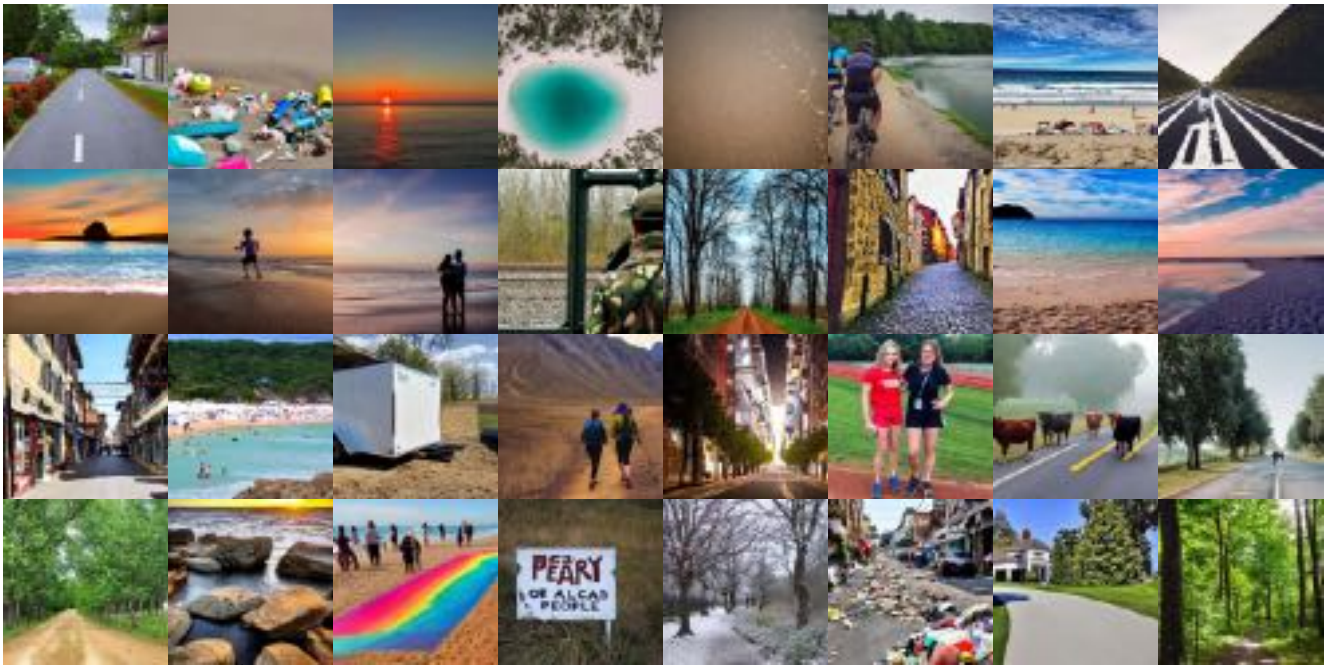


Figure 23. Top activating dataset examples for the concept ID 524 belonging to the SAE trained on the `cond` activation of `mid_block` at $t = 0.5$.

Figure 24. Top activating dataset examples for the concept ID 1314 belonging to the SAE trained on the `cond` activation of `mid_block` at $t = 0.5$.

Figure 25. Top activating dataset examples for the concept ID 2137 belonging to the SAE trained on the `cond` activation of `mid_block` at $t = 0.5$.

Figure 26. Top activating dataset examples for the concept ID 4972 belonging to the SAE trained on the `cond` activation of `mid_block` at $t = 0.0$.

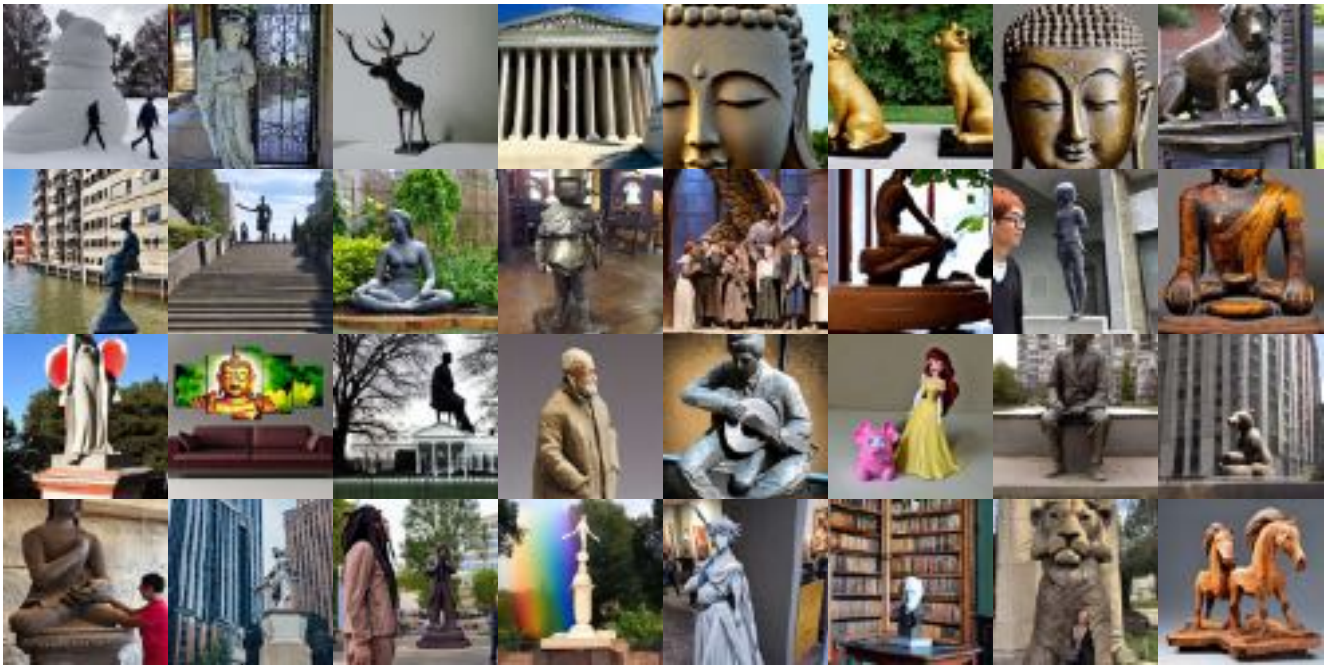


Figure 27. Top activating dataset examples for the concept ID 0 belonging to the SAE trained on the `cond` activation of `mid_block` at $t = 0.0$.

Figure 28. Top activating dataset examples for the concept ID 4979 belonging to the SAE trained on the `cond` activation of `mid_block` at $t = 0.0$.

Figure 29. Top activating dataset examples for the concept ID 86 belonging to the SAE trained on the `cond` activation of `down_block` at $t = 0.0$.