

## 477 A Feature Editor Application

478 We host our SAE-based feature editing app for you to try adding and subtracting SAE features during  
 479 a forward pass. To find interesting features to try, you can read the rest of the supplementary material  
 480 or you can try some of the ones of the ones we list on the app landing page. Alternatively, what works  
 481 best if you want to explore the features yourself is to generate images for which you believe your  
 482 features of interest should be on and have a look at their activation masks in the “Generate” tab. For

483 **SDXL Turbo App:** [https://huggingface.co/spaces/anonymous-author-129/  
 484 sdxlturbosae](https://huggingface.co/spaces/anonymous-author-129/sdxlturbosae)

485 **SDXL App:** <https://huggingface.co/spaces/anonymous-author-129/sdxlsae>

486 Notice that while we trained on SDXL Turbo 1-step mode the same features without additional  
 487 training also work for 4 steps and even for the base model with, e.g., 25 steps.

## 488 B Flux

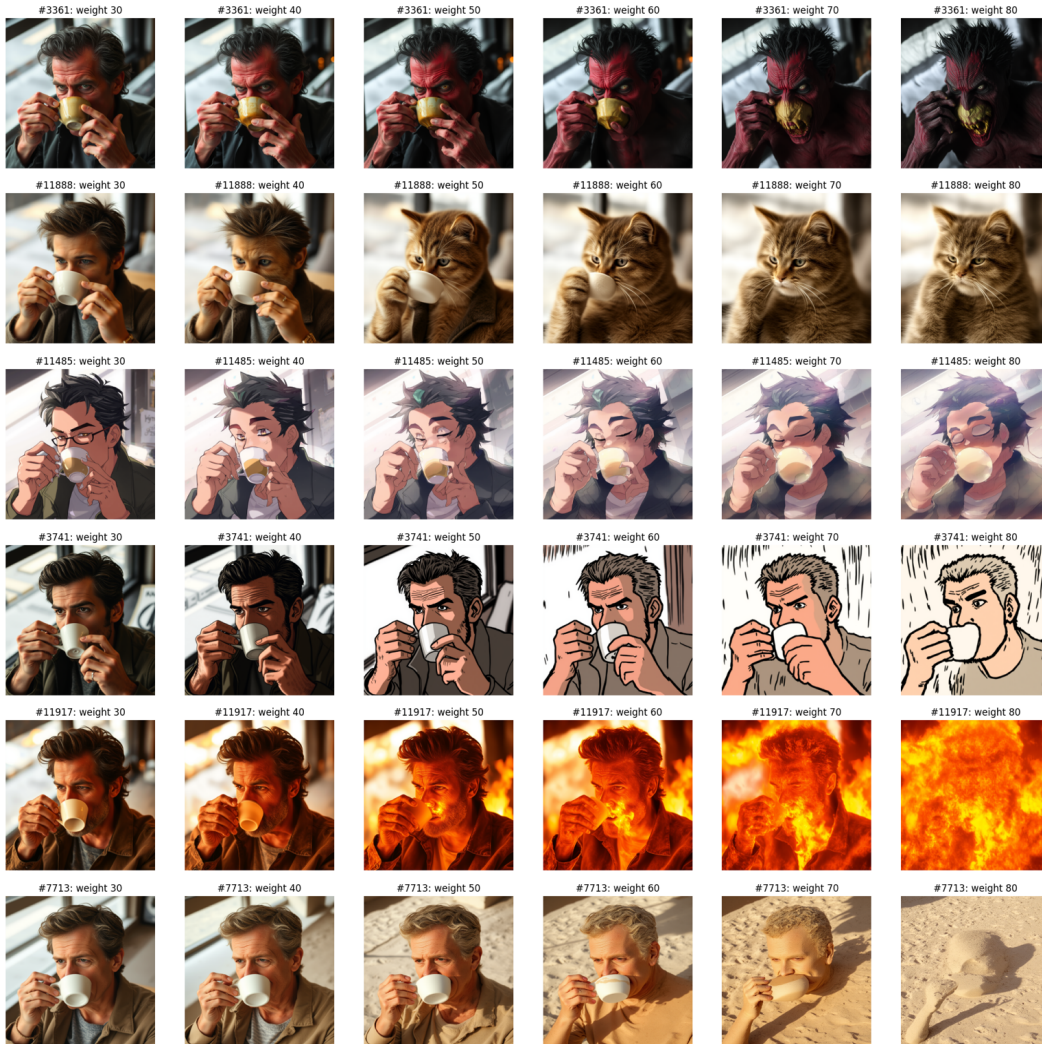


Figure 7: Feature injections on Flux-schnell 4-steps generations.

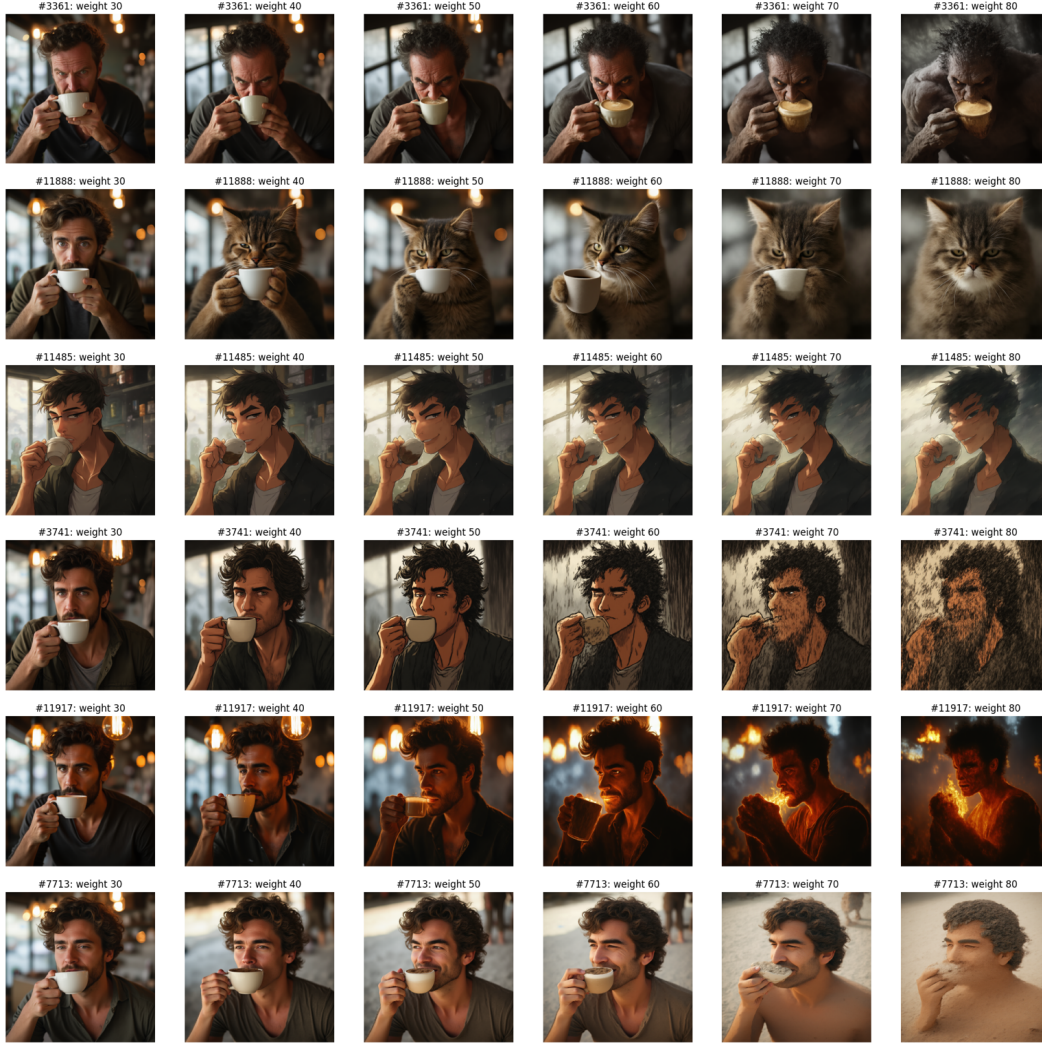


Figure 8: Feature injections on Flux-dev 25-steps generations.

**Training settings** We train a SAE on layer 18 activations of Flux-schnell 1-step. We choose layer 18 because we empirically find that its activations have higher norms than other layers. Additionally, all other exploratory experiments that we tried on FLUX, e.g., ablating layers, patching activations, simple activation steering, all consistently showed that layer 18 is a high impact layer. To train the SAE, we sample 1 million prompts from LAION-5B [52] and input them to Flux-schnell, then we randomly sample 10% of the activations (output - input) in the image stream of layer 18 (so for each prompt we get  $[64 \times 64 \times 0.1]$  3072-dimensional vectors).

The trained SAE has an expansion factor of 4 (thus its hidden dimension is 12288) and  $k = 20$ . All other hyperparameters and the training loss are detailed in [J].

**Features injection** Features learned on Flux-schnell 1-step can be used on Flux-schnell 4-steps as well as Flux-dev (we report examples with 25 steps). To inject a feature in a new generation, we simply rescale it by a strength factor and add it to the output of every layer starting from layer 18, finding that this way we can achieve high-quality results. Figures [7] and [8] show some examples of feature injections with varying strengths.



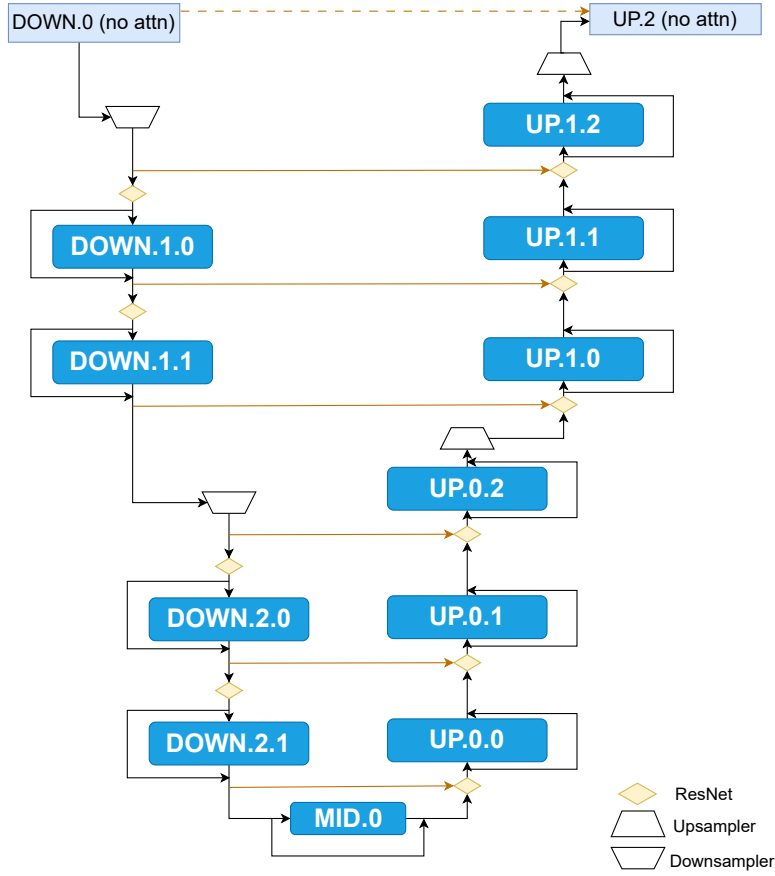


Figure 9: Cross-attention transformer blocks in SDLX’s U-net.

## C Finding Causally Influential Transformer Blocks

We narrow down design space of the 11 cross-attention transformer blocks (see Fig. 9) to those with the highest causal impact on the output. In order to assess their causal impact on the output we qualitatively study the effect of individually ablating each of them (see Fig. 10). As can be seen in Fig. 10 each of the middle blocks down.2.1, mid.0, up.0.0, up.0.1 have a relatively high impact on the output respectively. In particular, the blocks down.2.1 and up.0.1 stand out. It seems like most colors and textures are added in up.0.1, which in the community is already known as “style” block [55]. Ablating down.2.1, which is also already known in the community as “composition” block, impacts the entire image composition, including object sizes, orientations and framing. The effects of ablating other blocks such as mid.0 and up.0.0 are more subtle. For mid.0 it is difficult to describe in words and up.0.0 seems to add local details to the image while leaving the overall composition mostly intact.

## D Interventions in the Multi-Step Setting

In addition to our quantitative analysis from the main paper showing that features stabilize fast and are relatively shared across timesteps, here, we performed a series of experiments to also qualitatively assess the impact of performing interventions across multiple timesteps and also on subsets of timesteps. See Fig. 11, 17, 18, 19, 20, 21, 22, and 23.

Broadly, these results are aligned with what one would expect. Intervening from the beginning to the end leads to big perturbations of the original generation. Starting the interventions at later

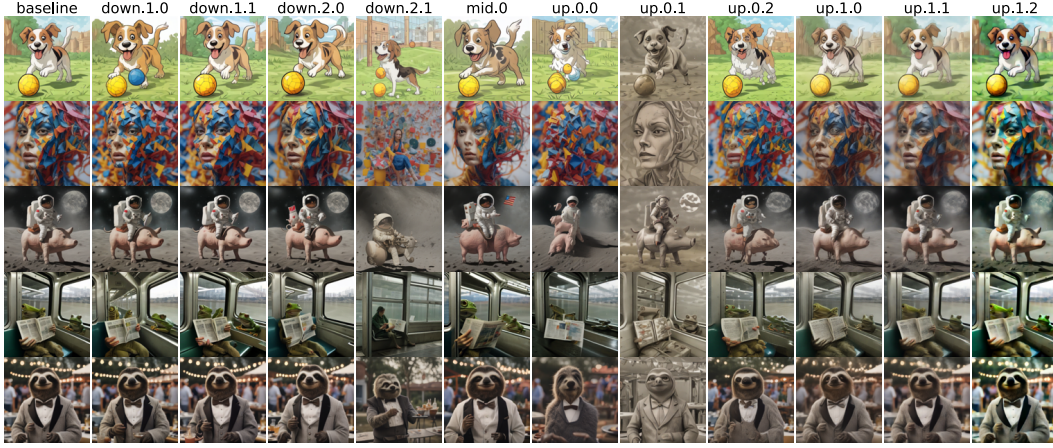


Figure 10: We generate images for the prompts “A dog playing with a ball cartoon.”, “A photo of a colorful model.”, “An astronaut riding on a pig on the moon.”, “A photograph of the inside of a subway train. There are frogs sitting on the seats. One of them is reading a newspaper. The window shows the river in the background.” and “A cinematic shot of a professor sloth wearing a tuxedo at a BBQ party.” while ablating the updates performed by different cross-attention layers (indicated by the titles). The title “baseline” corresponds to the generation without interventions.

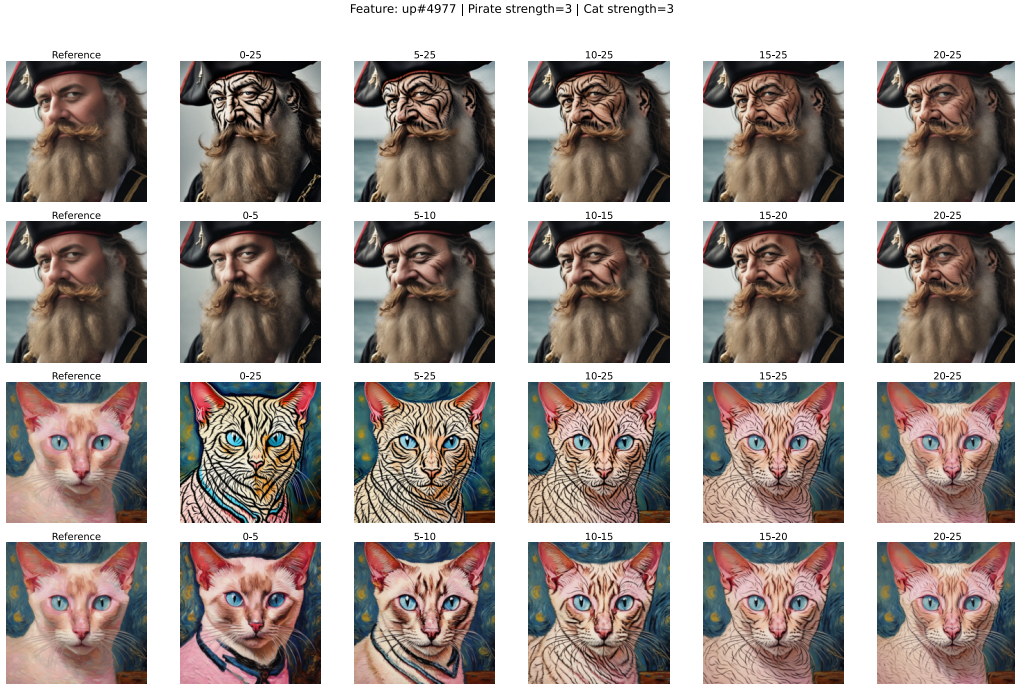


Figure 11: Performing interventions across different time intervals. For each prompt there are two rows, the first row contains ranges 0-25, 5-25, 10-25, 15-25, 20-25 and the second one 0-5, 5-10, 10-15, 15-20, 20-25. We would describe this feature as “tiger texture feature”. We intervened with this feature across the entire face of the pirate and across the entire cat except its ears. *These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ . This is a preview, please find the remaining multi-step intervention figures at the bottom of the document after the text.*



denoising steps keeps more of the original generated image intact. Interestingly, the sliding window of interventions shows that the different transformer blocks can have different effective ranges, e.g., up .0 .1 features start working later than down .2 .1 features.

## E Feature Transport Interventions

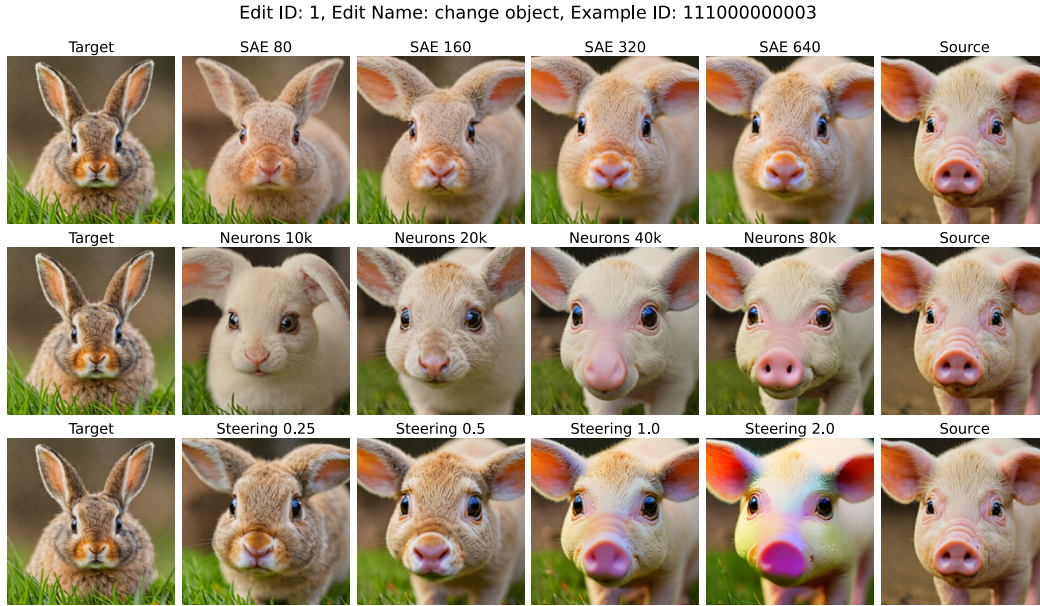
For each of our PIEBench adaptation’s edit categories, we implement corresponding feature transport interventions. We always add the top features from the source forward pass into the target forward pass (top and bottom features are determined via our feature importance criterion in equation 9) and for some categories also subtract the bottom features within the source forward pass. Before selecting features we aggregate them across spatial positions and timesteps by simply taking the mean across these dimensions. The different interventions for the different categories mainly differ in where the features are collected and where they are inserted, whether they are inserted in the target mask or within the source mask and if they are inserted in the source mask whether the spatial information is kept or not. When inserting in the target mask, the spatial information from the source mask cannot be kept.

1. **Change object:** We add the top features using the source mask while maintaining their spatial locations and we subtract the bottom features within the target mask. See Fig. 24 and Fig. 12
2. **Add object:** In this edit category we use the source mask in both images to collect the relevant feature coefficients to contrast with our criterion. Then, we add the top features using the source mask while maintaining their spatial locations and subtract the bottom ones also using the source mask. See Fig. 25.
3. **Delete object:** Here, we within the target (= original) image we collect features within the object and then again using the inverted mask (i.e., usually the background). Then we subtract the top features and add the bottom ones within the target mask. See Fig. 26
4. **Change content:** We add the top features using the source mask while maintaining their spatial locations and we subtract the bottom features within the target mask. See Fig. 27
5. **Change pose:** We add the top features using the source mask while maintaining their spatial locations and we subtract the bottom features within the target mask. See Fig. 28
6. **Change color:** We add the top features using the target mask while using their 95% quantile value across spatial locations and we subtract the bottom features within the target mask. See Fig. 29
7. **Change material:** We add the top features using the target mask while using their 95% quantile value across spatial locations and we subtract the bottom features within the target mask. See Fig. 30.
8. **Change background:** We add the top features using the target mask while using their 95% quantile value across spatial locations and we subtract the bottom features within the target mask. See Fig. 31
9. **Change style:** For this category we collect features across all spatial locations and subtract the bottom ones in the target image while adding the 95% quantile value of the top ones. See Fig. 32

### E.1 Feature Visualization Techniques

We introduce our methods used for feature visualization used in Fig. 13. Informally, given a feature, *spatial activations* (denoted by  $\text{hmap}$ ) highlight the regions of an image where the feature activates during generation process. *Activation modulation* (A. columns) refers to the intervention process in which the feature activations are enhanced or diminished. This technique is used to demonstrate how the manipulation of a feature’s value affects the generated image. Finally, *empty-prompt interventions* (B. column) illustrate the isolated role of the feature by disabling all other features during generation conditioned on an empty prompt. In the remainder of this section, we provide formal definitions and details.

**Spatial activations.** We visualize a sparse feature map  $S^\rho \in \mathbb{R}^{h \times w}$  containing activations of a feature  $\rho$  across the spatial locations by up-scaling it to the size of the generated images and overlaying it as



(a) Row 1 and row 2: varying number of SAE features / neurons transported; Row 3: steering with different strengths.



(b) Row 1 and row 2: strength 1 and strength 2 for SAE interventions. Row 3 and row 4. strength 1 and strength 2 for neuron interventions.

Figure 12: Example for edit category 1: “change object”. Original prompt (target): “a cute little bunny with big eyes”, edit prompt (source): “a cute little pig with big eyes”. Source and target refers to from where we extract features (source) and where we insert them (target). Grounded SAM2 masks used to collect the features are not shown but in this example they would select the entire foreground objects respectively. **This is a preview, please find the figures for the remaining edit categories at the bottom of this document after the text.**



572 a heatmap over the generated images. In the heatmap, red indicates the highest feature activation, and  
 573 blue represents the lowest non-zero one.

574 **Top dataset examples.** For a given feature  $\rho$ , we sort dataset examples according to their average  
 575 spatial activation

$$a_\rho = \frac{1}{wh} \sum_{i=1}^h \sum_{j=1}^w S_{ij}^\rho \in \mathbb{R}. \quad (10)$$

576 We use equation 10 to define the top dataset examples and to sample from the top 5% quantile of the  
 577 activating examples ( $a_\rho > 0$ ). We will refer to them as top 5% images for a feature  $\rho$ .

578 Note that  $S_{ij}^\rho$  always depends on an embedding of the input prompt  $c$  and input noise  $z_1$ , via  
 579  $S_{ij}(c, z_1) = \text{ENC}(\Delta D_{ij}(c, z_1))$ , which we usually omit for ease of notation. As a result,  $a_\rho$  also  
 580 depends on  $c$  and  $z_1$ . When we refer to the top dataset examples, we mean our  $(c, z_1)$  pairs with the  
 581 largest values for  $a_\rho(c, z_1)$ .

582 **Activation modulation.** We design interventions that allow us to modulate the strength of the  $\rho$ th  
 583 feature. Specifically, we achieve this by adding or subtracting a multiple of the feature  $\rho$  on all of the  
 584 spatial locations  $i, j$  proportional to its original activation  $S_{ij}^\rho$

$$\Delta D'_{ij} = \Delta D_{ij} + \beta S_{ij}^\rho \mathbf{f}_\rho, \quad (11)$$

585 in which  $\Delta D_{ij}$  is the update performed by the transformer block before and  $\Delta D'_{ij}$  after the interven-  
 586 tion,  $\beta \in \mathbb{R}$  is a modulation factor, and  $\mathbf{f}_\rho$  is the  $\rho$ th learned feature vector. In the following, we will  
 587 refer to this intervention as *activation modulation intervention*.

588 Note that  $S_{ij}^\rho$  can be also freely defined allowing for the application of sparse features to arbitrary  
 589 images and spatial positions (refer to Fig. 1 for examples).

590 **Activation on empty context.** Another way of visualizing the causal effect of features is to activate  
 591 them while doing a forward pass on the empty prompt  $c(\text{" "})$ . To do so, we turn off all other features  
 592 at the transformer block  $\ell$  of intervention and turn on the target feature  $\rho$ . Formally, we modify the  
 593 forward pass by setting

$$D_{ij}^{\text{out}'} = D_{ij}^{\text{in}} + \gamma k \mu_\rho \mathbf{f}_\rho, \quad (12)$$

594 in which  $D_{ij}^{\text{out}'}$  replaces residual stream plus transformer block update,  $D_{ij}^{\text{in}}$  is the input to the block,  
 595  $\mathbf{f}_\rho$  is the  $\rho$ th learned feature vector,  $\gamma \in \mathbb{R}$  is a hyperparameter to adjust the intervention strength,  
 596 and  $\mu_\rho$  is a feature-dependent multiplier obtained by taking the average activation across positive  
 597 activations of  $\rho$  (collected over a subset of 50,000 dataset examples). Multiplying it by  $k$  aims to  
 598 recover the coefficients lost by setting the other features to zero. Further in the text, we will refer to  
 599 this intervention as *empty-prompt intervention*, and the images generated using this method with  $\gamma$   
 600 set to 1, as *empty-prompt intervention images*.

601 Note that we directly added/subtracted feature vectors to the dense vectors for both intervention types  
 602 instead of encoding, manipulating sparse features, and decoding. This approach helps mitigate side  
 603 effects caused due to reconstruction loss (see App. K).

## 604 F Case Study: Most Active Features on a Prompt

605 Combining all our feature visualization techniques, in Fig. 13 we depict the features with the highest  
 606 average activation when processing the prompt: “A cinematic shot of a professor sloth wearing a  
 607 tuxedo at a BBQ party”. We discuss the transformer blocks in order of decreasing interpretability.

608 **Down.2.1** seems to contribute towards the image composition. Several features seem to relate directly  
 609 to phrases of the prompt: 4539 “professor sloth”, 4751, 1226, “wearing a tuxedo”, 2881, 567, 3119,  
 610 2345 “party”.

611 Turning off features (A. -6.0 column) removes elements and changes elements in the scene in ways  
 612 that align with heatmap (hmap column) and the top examples (C columns): 1674 *removes* the light  
 613 chains in the back, 4608 the umbrellas/tents, 4539 the 3D animation-like sloth face, 567 people in the  
 614 background, 3119, 2345 some of the light chains, and, 4751 *changes* the type of suit, 1226 the shirt.



Figure 13: The top 9 features of down.2.1 (a), up.0.1 (b), up.0.0 (c) and mid.0 (d) for the prompt: “A cinematic shot of a professor sloth wearing a tuxedo at a BBQ party.” Each row represents a feature. The first column depicts a feature heatmap (highest activation red and lowest nonzero one blue). The column titles containing “A” show feature modulation interventions, the ones containing “B” the intervention of turning on the feature on the empty prompt, and the ones containing “C” depict top dataset examples. Floating point values in the title denote  $\beta$  and  $\gamma$  values. *These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ .*

615 Similarly, enhancing the same features (A. 6.0 column) enhances the corresponding elements and  
 616 sometimes changes them.

617 Activating the features on the empty prompt often creates related elements. Note that, for the fixed  
 618 random seed we use, the empty prompt itself looks like a painting of a piece of nature with a lot of  
 619 green and brown. Therefore, while the prompt is empty the features active during the forward pass  
 620 are not and due to the layers that we don’t intervene on still contribute to the images.



While top dataset examples (C.0, C.1 columns) and also empty-prompt intervention (B. column) mostly agree with the feature activation heatmaps (hmap column), some of them add additional insight, e.g., 2881, which activates on the suit, seems to correspond to (masqueraded) characters in a (festive) scene, 3119 seems to be about party decorations in general and not just light chains, 2345 seems to react to other celebration backgrounds as well.

**Up.0.1** transformer block indeed seems to contribute substantially to the style of the image. They are hard to relate directly to phrases in the prompt, yet indirectly they do relate. E.g., the illumination (2727) and shadow (500, 1700) effects probably have something to do with “a cinematic shot” and the animal hair texture (2314) with “sloth”. Beyond that several features seem to mainly contribute to the glowing lights in the background (1295, 4238, 2341).

Interestingly, turning on the up.0.1 features on the entire empty prompt (B. column) results in texture-like images. In contrast, when activating them locally (A. columns) their contribution to the output is highly localized and keeps most of the remaining image largely unchanged. For the up.0.1 we find it remarkable that often the ablation and amplification are counterparts: 500 (light, shadow), 2727 (shadow, light), 3936 (blue, orange), 2314 (less grey hair, more brown hair).

**Up.0.0.** First, we observe that up.0.0 features act very locally and we think that it often requires relevant other features from the previous and subsequent transformer blocks effectively influence the image. For the empty prompt, activating these features results in abstract looking images, which are hard to relate to the other columns. Thus, we excluded this visualization technique and instead added one more example.

Most top dataset examples and their activations (C columns) are highly interpretable: 3603 party decoration, 5005 upper part of tent, 775 buttons on suit, 153 lower animal jaw, 1550 collars, 2648 pavilions, 1604 right part of the image, 564 bootie. Many of the features have a expected causal effect on the generation when ablating/enhancing (B. columns): 3603, 5005, 775, 153, 1550, 564, but not all: 2221, 2648, 1604. To sum up, this transformer block seems to mostly add local details to the generation and when interventions are performed locally they are effective.

**Mid.0.** Again to the best of our knowledge, mid.0’s role is also not well understood. We find it harder to interpret because most interventions on the mid.0 have very subtle effects. Similar to up.0.0, we did not include the results of empty-prompt interventions.

While effects of interventions are subtle, dataset examples (C. columns) and heatmap (hmap column) all mostly agree with each other and are specific enough to be interpretable: 4755 bottom right part of faces, 4235 left part of (animal) faces, 1388 people in the background, 1935 is active on chests, 473 mostly active on the image border, 2322 again seems to have to do with backgrounds that also contain people, 3067 active on the neck or neck accessories, and, 5102 outlines the left border of the main object in the scene. The feature 3018 is difficult to interpret.

Our observations indicate that mid.0’s features encode more abstract concepts. Particularly, some of them are activated at specific spatial locations within images<sup>8</sup> and other features potentially signify how image objects relate to each other.

## G Case Study: Random Features

In this case study, we explore the learned features independently of any specific prompt. *We moved the many and large figures corresponding to this section to the end of the supplementary material.* In Fig. 33 and Fig. 34 we demonstrate the first 5 and last 5 learned features for each transformer block (since SAEs were initialized randomly before training, we can treat these features as a random sample). As SAEs are randomly initialized before the training process, these sets can be considered as random samples of features. Each feature visualization consists of 3 images of top 5% images for this feature, and their perturbations with activation modulation interventions. For down.2.1 and up.0.1, we also include the empty-prompt intervention images. Additionally, we provide visualizations of several selected features in App. G Fig. 35 and demonstrate the effects of their forced activation on unrelated prompts in App. G Fig. 36.

<sup>8</sup>SDXL Turbo does not utilize positional encodings for the spatial locations in the feature maps. Therefore, we did a brief sanity check and trained linear probes to detect  $i, j$  given  $D_{ij}^m$ . These probes achieved high accuracy on a holdout set: 97.9%, 98.48%, 99.44%, 95.57% for down.2.1, mid.0, up.0.0, up.0.1.

#### Active interventions

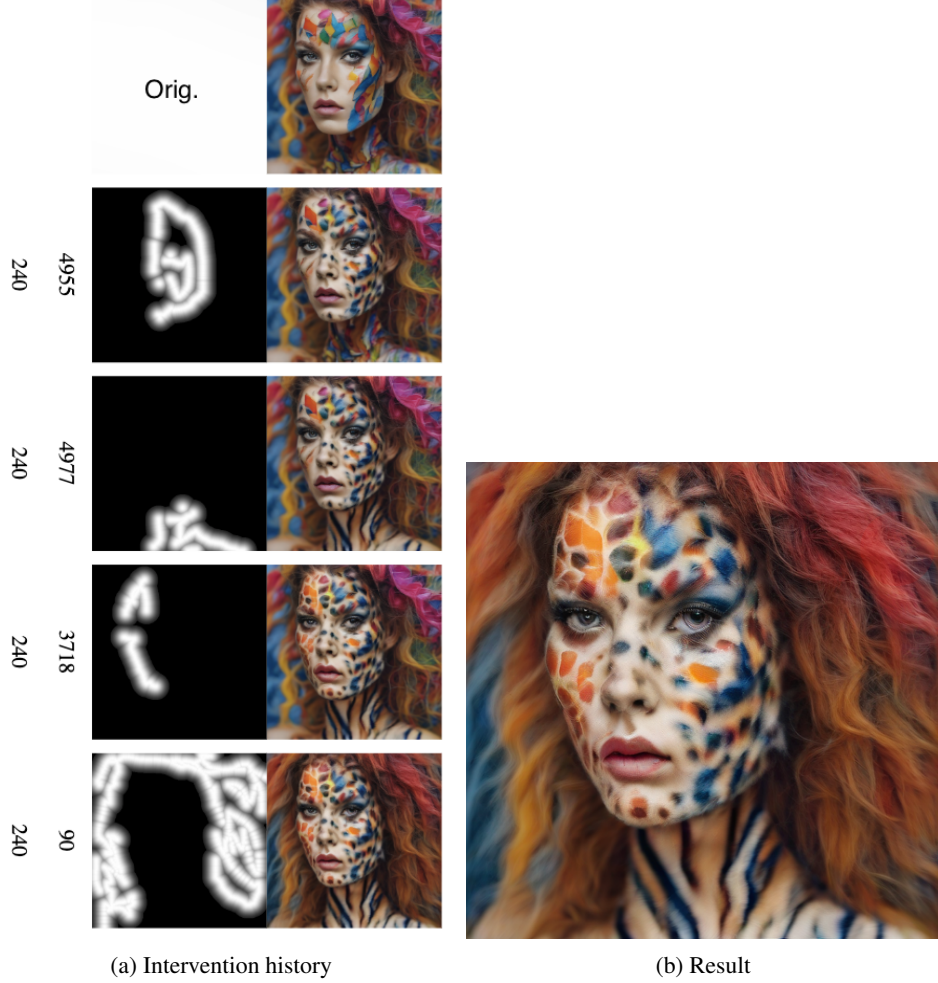


Figure 14: Local edits showcase up.0.1’s ability to locally change textures in the image without affecting the remaining image. Multiple consecutive interventions are possible (a). The first in (a) row depicts the original image and each subsequent row we add an intervention by drawing a heatmap with a brush tool and then turning on the feature labelling the row only on that area. The other number (240) is the absolute feature strength of the edit. Figure (b) shows the final result in full resolution (512x512). *These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ .*

**Feature plots.** We provide the same plots as in Fig. 33 but for the last six feature indices of each transformer block in Fig. 34 and the corresponding prompts in Table 6. Additionally, provide some selected features for down.2.1 and up.0.1 in Fig. 35 and the corresponding prompts in Table 7.

**Intervention plots.** Additionally, we provide plots in which we turn on features from Fig. 35 but in unrelated prompts (as opposed to top dataset example prompts that already activate the features by themselves). For simplicity here we simply turn on the features across all spatial locations, which does not seem to be a well suitable strategy for up.0.1, which usually acts locally. To showcase, the difference we created one example image in Fig. 14 in which we manually draw localized masks to turn on the corresponding features.

## H Quantitative Evaluation of the Roles of the Blocks

In this section, we follow up on qualitative insights by collecting quantitative evidence.



## H.1 Annotation Pipeline

Feature annotation with an LLM followed by further evaluation is a common way to assess feature properties such as specificity, sensitivity, and causality [7]. We found it applicable to the features learned by the `down.2.1` transformer block, which have a strong effect on the generation. Thus, they are amenable to automatic annotation using visual language models (VLMs) such as GPT-4o [37]. In contrast, for the features of other blocks with more subtle effects, we found VLM-generated captions to be unsatisfactory. In order to caption the features of `down.2.1`, we prompt GPT-4o with a sequence of 14 images. The first five images are irrelevant to the feature (i.e., the feature was inactive during the generation of the images), followed by a progression of 4 images with increasing average activation values, and finished by five images with the highest average activation values. The last nine images are provided alongside their so-called “coldmaps”: a version of an image with weakly active and inactive regions being faded and concealed. The prompt template and examples of the captions can be found in the App. I.

## H.2 Experimental Details

We perform a series of experiments to get statistical insights into the features. We report the majority of the experimental scores in the format  $M(S)$ . When the score is reported in the context of a SDXL Turbo transformer block, it means that we computed the score for each feature of the block and set  $M$  and  $S$  to mean and standard deviation across the feature scores. Note that  $S$  does not represent the error margin of  $M$ , as the actual error margin is much lower<sup>9</sup>. Therefore, almost all the differences in the reported means are statistically significant. For the baselines, we calculate the mean and standard deviation across the scores of a 100-element sample.

Table 1: Specificity, texture score, and color activation for different blocks and baselines.

Block	Specificity	Texture	Color
Down.2.1 SAE	0.76 (0.10)	0.16 (0.02)	86.2 (14.9)
Down.2.1 Neurons	0.65 (0.09)		
Down.2.1 PCA all	0.58 (0.06)		
Down.2.1 PCA 50	0.66 (0.08)		
Down.2.1 PCA 100	0.64 (0.08)		
Down.2.1 PCA 500	0.61 (0.07)		
Mid SAE	0.70 (0.10)	0.14 (0.01)	84.7 (16.3)
Mid Neurons	0.67 (0.07)		
Up.0.0 SAE	0.74 (0.10)	0.18 (0.03)	86.3 (16.5)
Up.0.0 Neurons	0.67 (0.07)		
Up.0.1 SAE	0.73 (0.09)	0.20 (0.02)	73.8 (20.6)
Up.0.1 Neurons	0.66 (0.08)		
Random	0.57 (0.09)	0.13 (0.02)	90.7 (54.9)
Same Prompt	0.89 (0.06)	–	–
Textures	–	0.18 (0.02)	–

**Interpretability.** Features are usually considered interpretable if they are sufficiently specific, i.e., images exhibiting the feature share some commonality. In order to measure this property, we compute the similarity between images on which the feature is active. High similarity between these images is a proxy for high specificity. For each feature, we collect 10 random images among top 5% images for this feature and calculate their average pairwise CLIP similarity [43, 9]. This value reflects how semantically similar the contexts are in which the feature is most active. We display the results in the first column of Table I, which shows that the CLIP similarity between images with the feature active is significantly higher than the random baseline (CLIP similarity between random images) for all transformer blocks. This suggests that the generated images share similarities when a feature is active.

For `down.2.1` we compute an additional *interpretability* score by comparing how well the generated annotations align with the top 5% images. The resulting CLIP similarity score is 0.21 (0.03) and significantly higher than the random baseline (average CLIP similarity with random images) 0.12 (0.02). To obtain an upper bound on this score we also compute the CLIP similarity to an image generated from the feature annotation, which is 0.25 (0.03).

<sup>9</sup>Given that  $M$  is computed over a sample of 1280 elements, the confidence interval of  $M$  can be estimated as  $M \pm S \cdot 0.055$ .

**Causality.** We can use the feature annotations to measure a feature’s causal strength by comparing the empty prompt intervention images with the caption<sup>10</sup>. The CLIP similarity between intervention images and feature caption is 0.19 (0.04) and almost matches the annotation-based interpretability score of 0.21 (0.03). This suggests that feature annotations effectively describe to the corresponding empty-prompt intervention images. Notably, the annotation pipeline did not use empty-prompt intervention images to generate captions. This fact speaks for the high causal strength of the features learned on `down.2.1`.

**Sensitivity.** A feature is considered sensitive when activated in its relevant context. As a proxy for the context, we have chosen the feature annotations obtained with the auto-annotation pipeline. For each learned feature, we collected the 100 prompts from a 1.5M sample of LAION-COCO with the highest sentence similarity based on sentence transformer embeddings of all-MiniLM-L6-v2 [47]. Next, we run SDXL Turbo on these prompts and count the proportion of generated images in which the feature is active on more than 0%, 10%, 30% of the image area, resulting in 0.60 (0.32), 0.40 (0.34), 0.27 (0.30) respectively, which is much higher than the random baseline, which is at 0.06 (0.09), 0.003 (0.006), 0.001 (0.003). However, the average scores are  $< 1$  and thus not perfect. This may be caused by incorrect or imprecise annotations for subtle features and, therefore, hard to annotate with a VLM and SDXL Turbo failing to comply with some prompts.

**Relatedness to texture.** In Fig. 13 the empty prompt interventions of the `up.0.1` features resulted in texture-like pictures. To quantify whether this consistently happens, we design a simple texture score by computing CLIP similarity between an image and the word “texture”. Using this score, we compare empty-prompt interventions of the different transformer blocks with each other and real-world texture images. The results are in the second column of Table 1 and suggest that empty-prompt intervention images of `up.0.1` and `up.0.0` resemble textures and some of the `down.2.1` images look like textures as well. For `up.0.0`, we did not observe any connection of these images to the top activating images. Interestingly, the score of `up.0.1` is higher than the one of the real-world textures dataset (Cimpoi et al. [10]).

**Color sensitivity.** In our qualitative analysis, we suggested that the features learned on `up.0.1` relate to texture and color. If this holds, the image regions that activate a feature should not differ significantly in color on average. To test that, we calculate the “average” color for each feature: this is a weighted average of pixel colors with the feature activation values as weights. To determine the average color of a each feature we compute it over a sample of 10 images of the feature’s top 5% images. Then, we calculate Manhattan distances between the colors of the pixels and the “average” color on the same images (the highest possible distance is  $3 \cdot 255 = 765$ ). Finally, we take a weighted average of the Manhattan distances using the same weights. We report these distances for different transformer blocks and for the images generated on random prompts from LAION-COCO. We present the results in the third column of Table 1. The average distance for the `up.0.1` transformer block is, in fact, the lowest.

Table 2: Manhattan distances between original and intervened images at varying intervention strengths outside/inside of the feature’s activation map.

Block	-10	-5	5	10
Down.2.1	148.2 / 116.0	124.2 / 94.4	101.4 / 78.7	128.9 / 105.60
Mid	69.2 / 32.2	39.4 / 18.5	33.2 / 15.2	59.9 / 29.82
Up.0.0	105.3 / 38.4	77.7 / 23.7	63.6 / 23.3	88.6 / 37.08
Up.0.1	125.0 / 26.8	73.1 / 16.4	68.6 / 21.9	98.9 / 34.74

**Intervention locality.** We suggested that features learned on `up.0.0` and `up.0.1` primarily influence local regions of the generation, with minimal effect outside the active areas. To test this, we measure changes in the top 5% images inside and outside the active regions while performing activation modulation interventions. To exclude weak activation regions from consideration, a pixel is considered inside the active area if the corresponding patch has an activation value larger than 50% of the image patches, and it is outside the active area if the corresponding patch has zero activation. Table 2 reports Manhattan distances between the original images and the intervened images outside and inside the active areas for activation modulation intervention strengths -10, -5, 5, 10. The features for `up.0.0` and `up.0.1` have a stronger effect inside the active area than outside, unlike `down.2.1` where the difference is smaller.

<sup>10</sup>We require feature captions for the causality and sensitivity analyses, we only have them for `down.2.1`.



## I Annotation Pipeline Details

We used GPT-4o to caption learned features on down.2.1. For each feature, the model was shown a series of 5 unrelated images, a progression of 9 images, the  $i$ -th of those corresponds to  $\sim i \cdot 10\%$  average activation value of the maximum. Finally, we show 5 images corresponding to the highest average activations. Since some features are active on particular parts of images, the last 9 images are provided alongside their so-called “coldmaps”: a version of an image with weakly active and inactive regions being faded and concealed.

The images were generated by 1-step SDXL Turbo diffusion process on 50'000 random prompts of LAION-COCO dataset.

### I.1 Textual Prompt Template

Here is the prompt template for the VLM.

**System.** You are an experienced mechanistic interpretability researcher that is labeling features from the hidden representations of an image generation model.

**User.** You will be shown a series of images generated by a machine learning model. These images were selected because they trigger a specific feature of a sparse auto-encoder, trained to detect hidden activations within the model. This feature can be associated with a particular object, pattern, concept, or a place on an image. The process will unfold in three stages:

1. **Reference Images:** First, you'll see several images *unrelated* to the feature. These will serve as a reference for comparison.

2. **Feature-Activating Images:** Next, you'll view images that activate the feature with varying strengths. Each of these images will be shown alongside a version where non-activated regions are masked out, highlighting the areas linked to the feature.

3. **Strongest Activators:** Finally, you'll be presented with the images that most strongly activate this feature, again with corresponding masked versions to emphasize the activated regions.

Your task is to carefully examine all the images and identify the thing or concept represented by the feature. Here's how to provide your response:

- **Reasoning:** Between '`<thinking>`' and '`</thinking>`' tags, write up to 400 words explaining your reasoning. Describe the visual patterns, objects, or concepts that seem to be consistently present in the feature-activating images but not in the reference images.

- **Expression:** Afterward, between '`<answer>`' and '`</answer>`' tags, write a concise phrase (no more than 15 words) that best captures the common thing or concept across the majority of feature-activating images.

Note that not all feature-activating images may perfectly align with the concept you're describing, but the images with stronger activations should give you the clearest clues. Also pay attention to the masked versions, as they highlight the regions most relevant to the feature.

**User.** These images are not related to the feature: {Reference Images}

**User.** This is a row of 9 images, each illustrating increasing levels of feature activation. From left to right, each image shows a progressively higher activation, starting with the image on the far left where the feature is activated at 10% relative to the image that activates it the most, all the way to the far right, where the feature activates at 90% relative to the image that activates it the most. This gradual transition highlights the feature's growing importance across the series. {Feature-Activating Images}

**User.** This row consists of 9 masked versions of the original images. Each masked image corresponds to the respective image in the activation row. Areas where the feature is not activated are completely concealed by a white mask, while regions with activation remain visible.) {Feature-Activating Images Coldmaps}

**User.** These images activate the feature most strongly. {Strongest Activators}

Table 3: down.2.1 first 10 and last 10 feature captions.

Block	Feature	Caption
down.2.1	0	Organizational/storage items for documents and office supplies
	1	Luxury kitchen interiors and designs
	2	Architectural Landmarks and Monumental Buildings
	3	Upper body clothing and attire
	4	Rustic or Natural Wooden Textures or Surfaces
	5	Intricately designed and ornamental brooches
	6	Technical diagrams and instructional content
	7	Feature predominantly activated by visual representations of dresses
	8	Home decor textiles focusing on cushions and pillows
	9	Eyewear: glasses and sunglasses
	5110	Concept of containment or organized enclosure
	5111	Groups of people in collective settings
	5112	Modern minimalist interior design
	5113	Indoor plants and greenery
	5114	Feature sensitivity focused on sneakers
	5115	Handling or manipulating various objects
	5116	Athletic outerwear, particularly zippered sporty jackets
	5117	Spectator Seating in Sporting Venues
	5118	Textiles and clothing materials, focus on textures and folds
	5119	Yarn and Knitting Textiles

817 **User.** These masked images highlight the activated regions of the images that  
818 activate the feature most strongly. The masked images correspond to the images  
819 above. The unmasked regions are the ones that activate the feature. {Strongest  
820 Activators Coldmaps}

## 821 I.2 Example of Prompt Images

822 The images used to annotate feature 0 are shown in Fig. 15

## 823 I.3 Examples of Generated Captions

824 We present the captions generated by GPT-4o for the first and last 10 features in Table 3

## 825 J Sparse Autoencoders and Superposition

826 This is an extended version of Sparse Autoencoders subsection of background section.

827 Let  $h(x) \in \mathbb{R}^d$  be some intermediate result of a forward pass of a neural network on the input  $x$ . In a  
828 fully connected neural network, the components  $h(x)$  could correspond to neurons. In transformers,  
829 which are residual neural networks with attention and fully connected layers,  $h(x)$  usually either  
830 refers to the content of the residual stream after some layer, an update to the residual stream by some  
831 layer, or the neurons within a fully connected block. In general,  $h(x)$  could refer to anything, e.g.,  
832 also keys, queries, and values. It has been shown [59][11][5] that in many neural networks, especially  
833 LLMs, intermediate representations can be well approximated by sparse sums of  $n_f \in \mathbb{N}$  learned  
834 feature vectors, i.e.,

$$h(x) \approx \sum_{\rho=1}^{n_f} s_{\rho}(x) \mathbf{f}_{\rho}, \quad (13)$$

835 where  $s_{\rho}(x)$  are the input-dependent<sup>11</sup> coefficients most of which are equal to zero and  $\mathbf{f}_1, \dots, \mathbf{f}_{n_f} \in$   
836  $\mathbb{R}^d$  is a learned dictionary of feature vectors.

837 Importantly, these learned features are usually highly *interpretable* (specific), *sensitive* (fire on the  
838 relevant contexts), *causal* (change the output in expected ways in intervention) and usually do not  
839 correspond directly to individual neurons. There are also some preliminary results on the universality  
840 of these learned features, i.e., that different training runs on similar data result in the corresponding  
841 models picking up largely the same features [5].

<sup>11</sup>In the literature this input dependence is usually omitted.

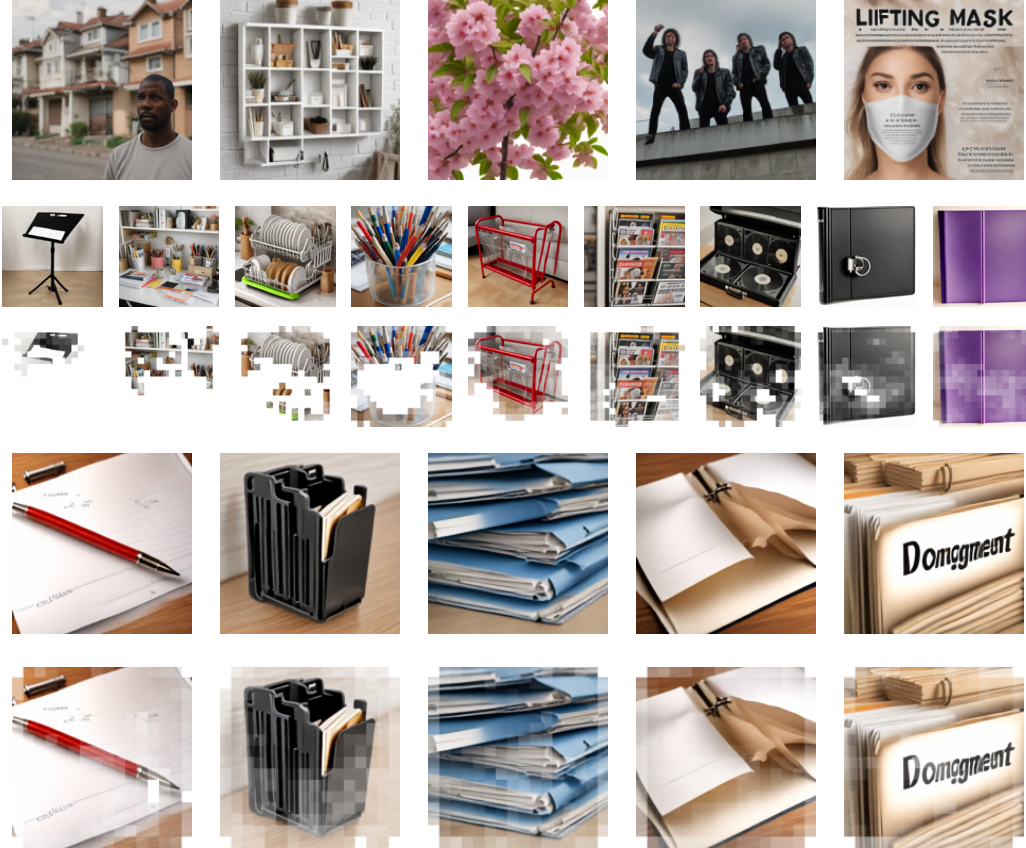


Figure 15: The images used by GPT-4o to generate captions for feature 0. From top to bottom: irrelevant images to feature 0; image progression from left to right, showing increasing activation of SAE feature 0, with low activation on the left and high activation on the right; “Coldmaps” representing the image progression; images corresponding to the highest activation of feature 0; “Coldmaps” corresponding to these highest activation images.

**Superposition.** By associating task-relevant features with directions in  $\mathbb{R}^d$  instead of individual components of  $h(x) \in \mathbb{R}^d$ , it is possible to represent many more features than there are components, i.e.,  $n_f \gg d$ . As a result, in this case, the learned dictionary vectors  $\mathbf{f}_1, \dots, \mathbf{f}_{n_f}$  cannot be orthogonal to each other, which can lead to interference when too many features are on (thus the sparsity requirement). However, it would be theoretically possible to have exponentially (in  $d$ ) many almost orthogonal directions embedded in  $\mathbb{R}^d$ <sup>12</sup>.

Using representations like this, the optimization process during training can trade off the benefits of being able to represent more features than there are components in  $h$  with the costs of features interfering with each other. Such representations are especially effective if the real features underlying the data do not co-occur with each other too much, that is, they are sparse. In other words, in order to represent a single input (“Michael Jordan”) only a small subset of the features (“person”, ..., “played basketball”) is required [17, 5].

The phenomenon of neural networks that exploit representations with more features than there are components (or neurons) is called superposition [17]. Superposition can explain the presence of polysemantic neurons. The neurons, in this case, are simply at the wrong level of abstraction. The closest feature vector can change when varying a neuron, resulting in the neuron seemingly reacting to or steering semantically unrelated things.

<sup>12</sup>It follows from the Johnson-Lindenstrauss Lemma [24] that one can find at least  $\exp(d\epsilon^2/8)$  unit vectors in  $\mathbb{R}^d$  with the dot product between any two not larger than  $\epsilon$ .



**Sparse autoencoders.** In order to implement the sparse decomposition from equation [13] the vector  $s$  containing the  $n_f$  coefficients of the sparse sum is parameterized by a single linear layer followed by an activation function, called the *encoder*,

$$s = \text{ENC}(h) = (W^{\text{ENC}}(h - b_{\text{pre}}) + b_{\text{act}}), \quad (14)$$

in which  $h \in \mathbb{R}^d$  is the latent that we aim to decompose,  $\sigma(\cdot)$  is an activation function,  $W^{\text{ENC}} \in \mathbb{R}^{n_f \times d}$  is a learnable weight matrix and  $b_{\text{pre}}$  and  $b_{\text{act}}$  are learnable bias terms. We omitted the dependencies  $h = h(x)$  and  $s = s(h)$  that are clear from context.

Similarly, the learnable features are parametrized by a single linear layer, called *decoder*,

$$h' = \text{DEC}(s) = W^{\text{DEC}}s + b_{\text{pre}}, \quad (15)$$

in which  $W^{\text{DEC}} = (\mathbf{f}_1 | \dots | \mathbf{f}_{n_f}) \in \mathbb{R}^{d \times n_f}$  is a learnable matrix of whose columns take the role of learnable features and  $b_{\text{pre}}$  is a learnable bias term.

**Training.** The pair ENC and DEC are trained in a way that ensures that  $h'$  is a sparse sum of feature vectors. Given a dataset of latents  $h_1, \dots, h_n$ , both encoder and decoder are trained jointly to minimize a proxy to the loss

$$\min_{\substack{W^{\text{ENC}}, W^{\text{DEC}} \\ b_{\text{pre}}, b_{\text{act}}}} \sum_{i=1}^n \|h'_i - h_i\|_2^2 + \lambda \|s_i\|_0, \quad (16)$$

where  $h_i = h(x_i)$ ,  $s_i = \text{ENC}(h(x_i))$  (when we refer to components of  $s$  we use  $s_\rho$  instead), the  $\|h'_i - h_i\|_2^2$  is a reconstruction loss,  $\|s_i\|_0$  a regularization term ensuring the sparsity of the activations and  $\lambda$  the corresponding trade-off term.

In practice,  $\|s_i\|_0$  cannot be efficiently optimized directly, which is why it is usually replaced with  $\|s_i\|_1$  or other proxy objectives.

**Technical details.** In our work, we make use of the top- $k$  formulation from [19], in which  $\|s_i\|_0 \leq k$  is ensured by introducing the a top- $k$  function TopK into the encoder:

$$s = \text{ENC}(h) = \text{RELU}(\text{TopK}(W^{\text{ENC}}(h - b_{\text{pre}}) + b_{\text{act}})). \quad (17)$$

As the name suggests, TopK returns a vector that sets all components except the top  $k$  ones to zero.

In addition [19] use an auxiliary loss to handle dead features. During training, a sparse feature  $\rho$  is considered *dead* if  $s_\rho$  remains zero over the last 10M training examples.

The resulting training loss is composed of two terms: the  $L_2$ -reconstruction loss and the top-auxiliary  $L_2$ -reconstruction loss for dead feature reconstruction. For a single latent  $h$ , the loss is defined

$$L(h, h') = \|h - h'\|_2^2 + \alpha \|h - h'_{\text{aux}}\|_2^2 \quad (18)$$

In this equation, the  $h'_{\text{aux}}$  is the reconstruction based on the top  $k_{\text{aux}}$  dead features. This auxiliary loss is introduced to mitigate the issue of dead features. After the end of the training process, we observed none of them. Following [19], we set  $\alpha = \frac{1}{32}$  and  $k_{\text{aux}} = 256$ , performed tied initialization of encoder and decoder, normalized decoder rows after each training step. The number of learned features  $n_f$  is set to 5120, which is four times the length of the input vector. The value of  $k$  is set to 10 as a good trade-off between sparsity and reconstruction quality. Other training hyperparameters are batch size: 4096, optimizer: Adam with learning rate:  $10^{-4}$  and betas: (0.9, 0.999).

## K SAE Training Results

We trained several SAEs with different sparsity levels and sparse layer sizes and observed no dead features. To assess reconstruction quality, we processed 100 random LAION-COCO prompts through a one-step SDXL Turbo process, replacing the additive component of the corresponding transformer block with its SAE reconstruction.

The explained variance ratio and the output effects caused by reconstruction are shown in Table 4. Fig. 16 presents random examples of reconstructions from an SAE with the following hyperparameters:  $k = 10$ ,  $n_f = 5120$ , trained on down. 2.1. The reconstruction causes minor deviations in the images, and the fairly low LPIPS [60] and pixel distance scores also support these findings. However, to prevent these minor reconstruction errors from affecting our analysis of interventions, we decided to directly add or subtract learned directions from dense feature maps.

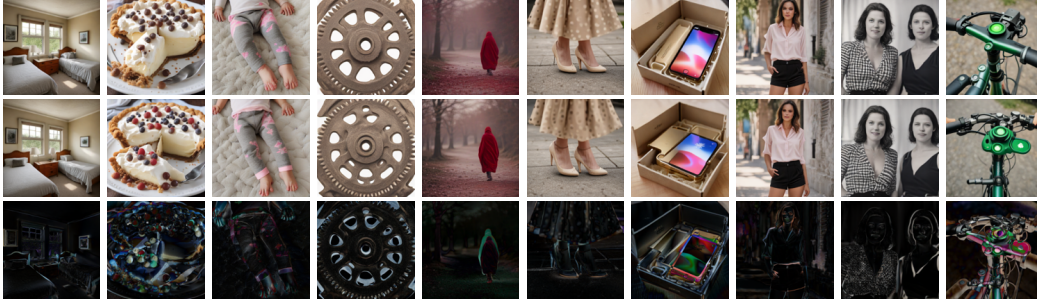


Figure 16: Images generated from 10 random prompts taken from the LAION-COCO dataset are shown in the first row. In the second row, down.2.1 updates are replaced by their SAE reconstructions ( $k = 10, n_f = 5120$ ). The third row visualizes the differences between the original and reconstructed images.

Table 4: Distances and explained variance ratio in generated images. “Mean” represents the average pixel Manhattan distance between original and reconstruction-intervened images, with a maximum possible value of 765. “Median” represents the median Manhattan distance per pixel, averaged over all images. ‘LPIPS’ refers to the average LPIPS score, measuring perceptual similarity. “Explained variance ratio” denotes the ratio of variance explained by the trained SAEs to the total variance.

$k$	$n_f$	Configuration	Mean	Median	LPIPS	EV (%)
5	640	down.2.1	83.29	50.04	0.3383	56.0
		mid.0	52.64	26.82	0.2032	43.4
		up.0.0	55.89	30.69	0.2276	44.8
		up.0.1	52.67	34.53	0.2073	50.3
	5120	down.2.1	74.68	41.49	0.3036	67.8
		mid.0	48.82	24.60	0.1845	50.8
		up.0.0	49.19	25.86	0.1969	57.2
		up.0.1	47.50	31.11	0.1775	59.5
10	640	down.2.1	73.65	41.79	0.2893	62.8
		mid.0	46.80	23.10	0.1772	51.5
		up.0.0	48.43	25.80	0.1908	52.5
		up.0.1	43.06	26.85	0.1638	58.7
	5120	down.2.1	64.97	34.77	0.2582	73.7
		mid.0	44.02	21.72	0.1627	58.8
		up.0.0	42.08	21.54	0.1624	64.2
		up.0.1	39.77	24.84	0.1453	67.1
20	640	down.2.1	59.29	31.47	0.2291	69.9
		mid.0	39.95	19.44	0.1459	60.0
		up.0.0	40.15	21.06	0.1499	60.9
		up.0.1	31.97	18.15	0.1196	66.7
	5120	down.2.1	56.37	29.04	0.2190	78.8
		mid.0	37.28	17.82	0.1328	66.5
		up.0.0	35.73	18.03	0.1302	70.6
		up.0.1	30.31	17.22	0.1104	74.2

Feature: down#2301 | Pirate strength=10 | Cat strength=5

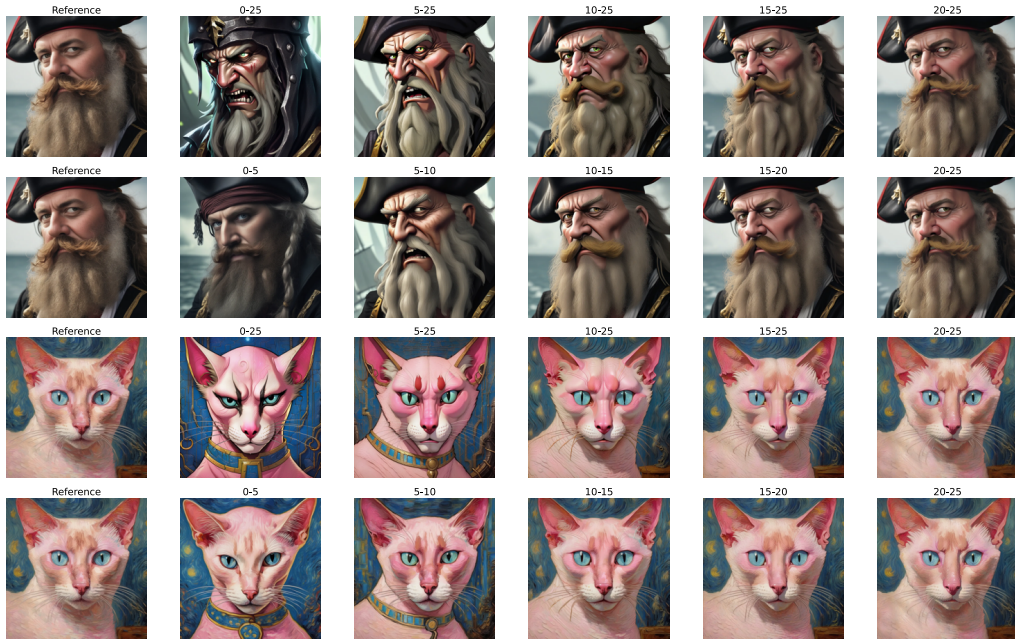


Figure 17: Performing interventions across different time intervals. For each prompt there are two rows, the first row contains ranges 0-25, 5-25, 10-25, 15-25, 20-25 and the second one 0-5, 5-10, 10-15, 15-20, 20-25. We would describe this feature as “evil feature”. We intervened with this feature across the entire spatial grid. *These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ .*



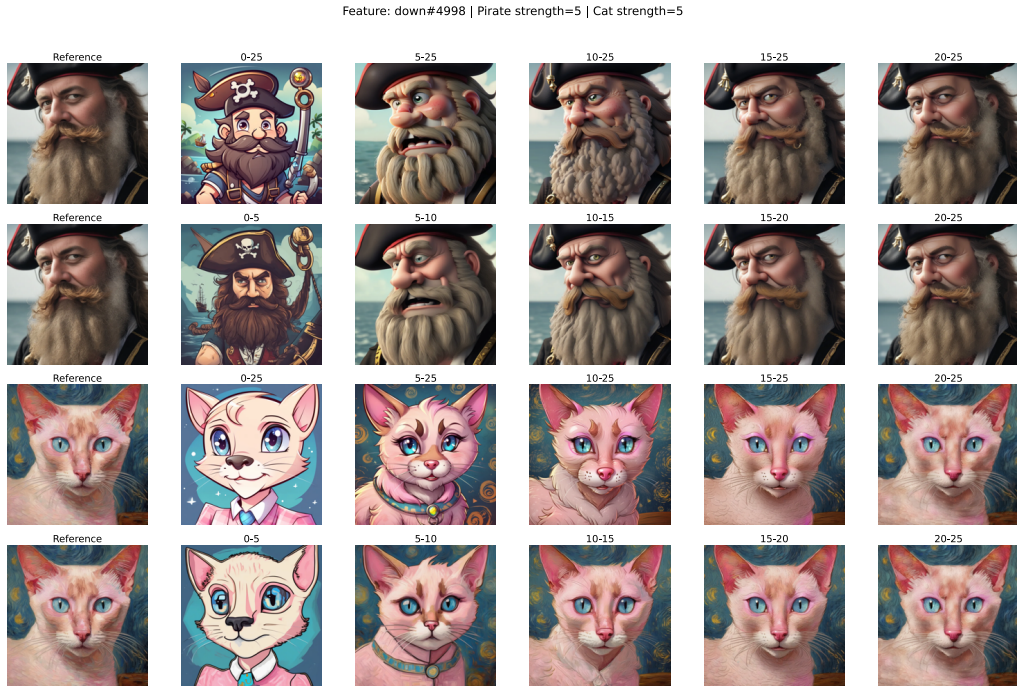


Figure 18: Performing interventions across different time intervals. For each prompt there are two rows, the first row contains ranges 0-25, 5-25, 10-25, 15-25, 20-25 and the second one 0-5, 5-10, 10-15, 15-20, 20-25. We would describe this feature as “cartoon feature”. We intervened with this feature across the entire spatial grid. *These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ .*

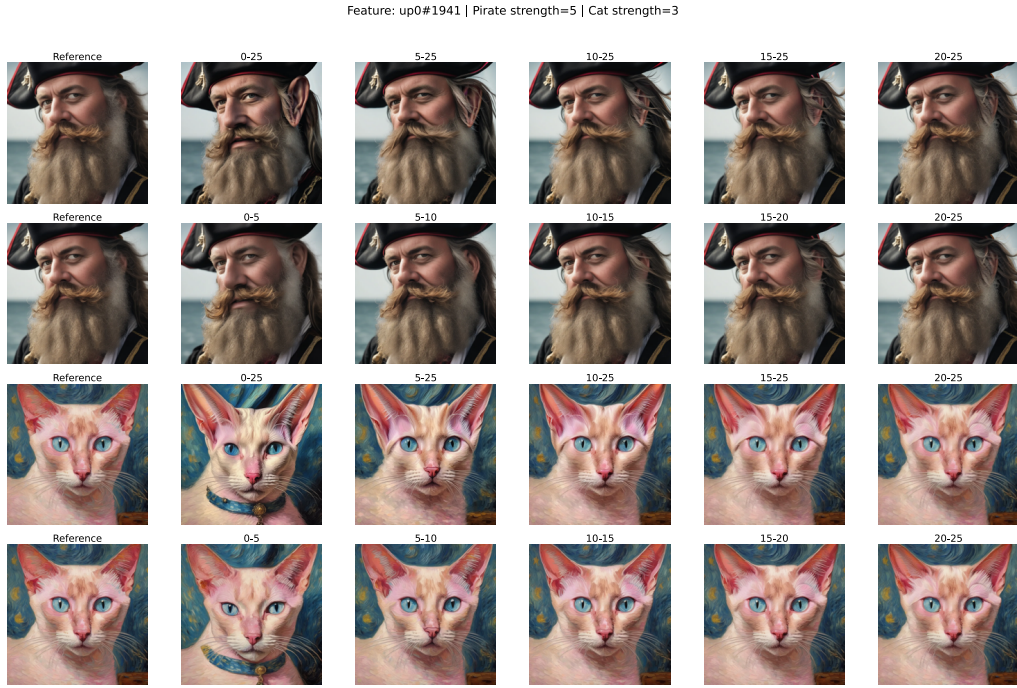


Figure 19: Performing interventions across different time intervals. For each prompt there are two rows, the first row contains ranges 0-25, 5-25, 10-25, 15-25, 20-25 and the second one 0-5, 5-10, 10-15, 15-20, 20-25. We would describe this feature as “ear feature”. We intervened with this feature on the ears. *These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ .*

Feature: up0#3742 | Pirate strength=-5 | Cat strength=3

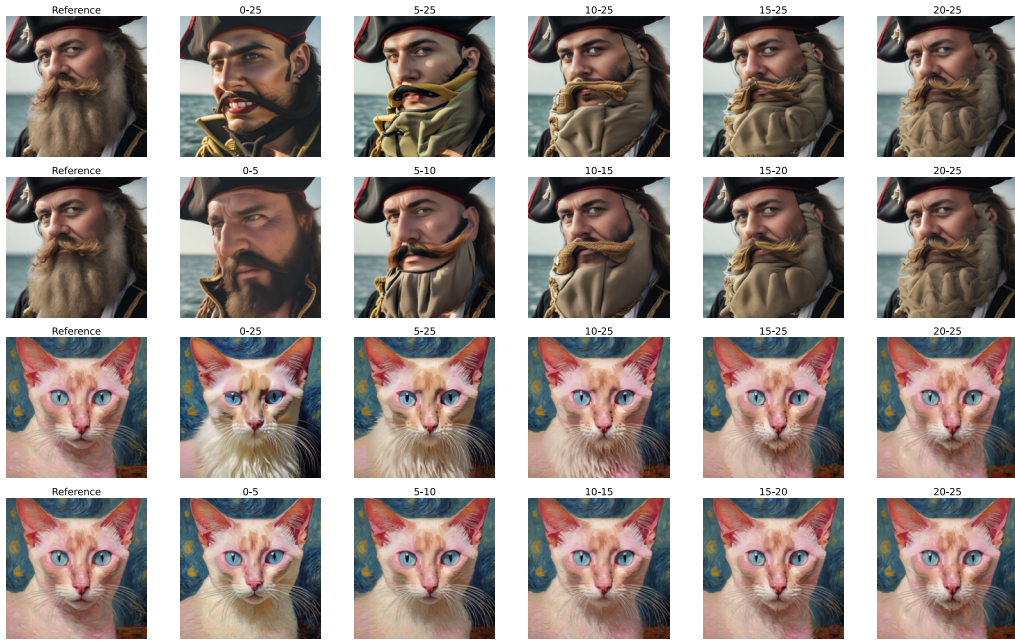


Figure 20: Performing interventions across different time intervals. For each prompt there are two rows, the first row contains ranges 0-25, 5-25, 10-25, 15-25, 20-25 and the second one 0-5, 5-10, 10-15, 15-20, 20-25. We would describe this feature as “beard feature”. We intervened with this feature on the chin/beard area. In the pirate we subtracted this feature. *These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ .*



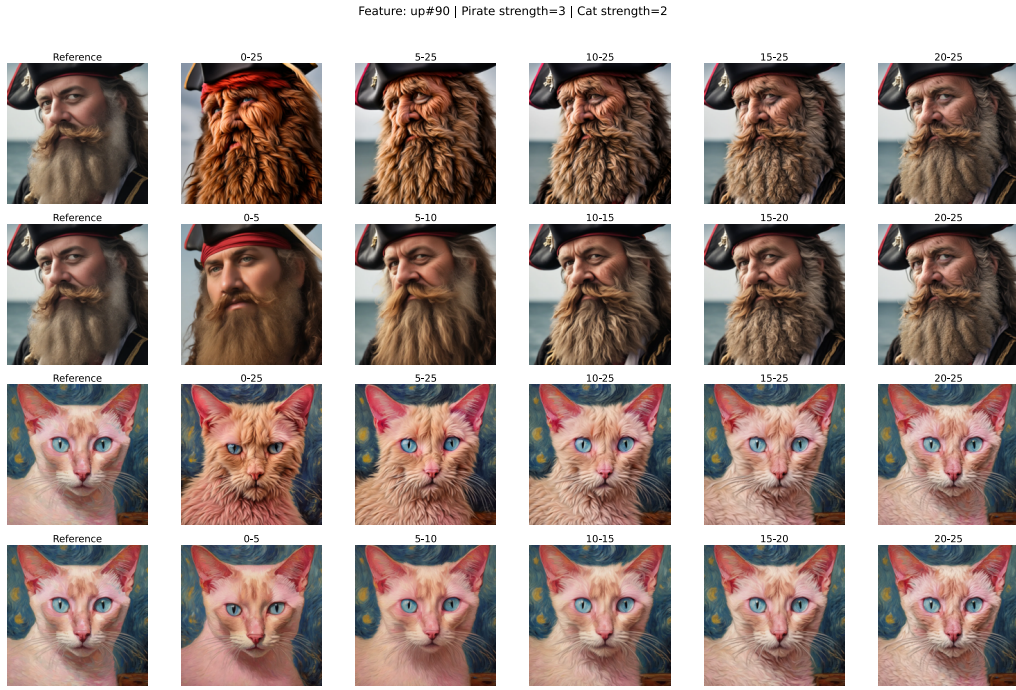


Figure 21: Performing interventions across different time intervals. For each prompt there are two rows, the first row contains ranges 0-25, 5-25, 10-25, 15-25, 20-25 and the second one 0-5, 5-10, 10-15, 15-20, 20-25. We would describe this feature as “furry feature”. We intervened with this feature across the entire beard and face of the pirate and across the entire cat except its ears. *These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ .*

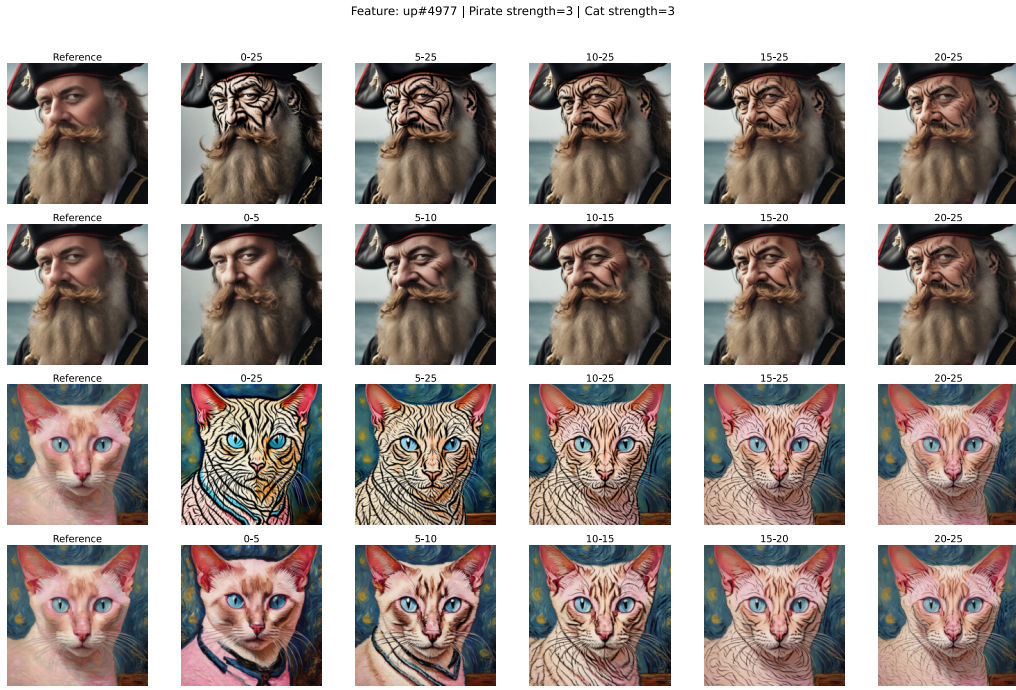


Figure 22: Performing interventions across different time intervals. For each prompt there are two rows, the first row contains ranges 0-25, 5-25, 10-25, 15-25, 20-25 and the second one 0-5, 5-10, 10-15, 15-20, 20-25. We would describe this feature as “tiger texture feature”. We intervened with this feature across the entire face of the pirate and across the entire cat except its ears. *These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ .*

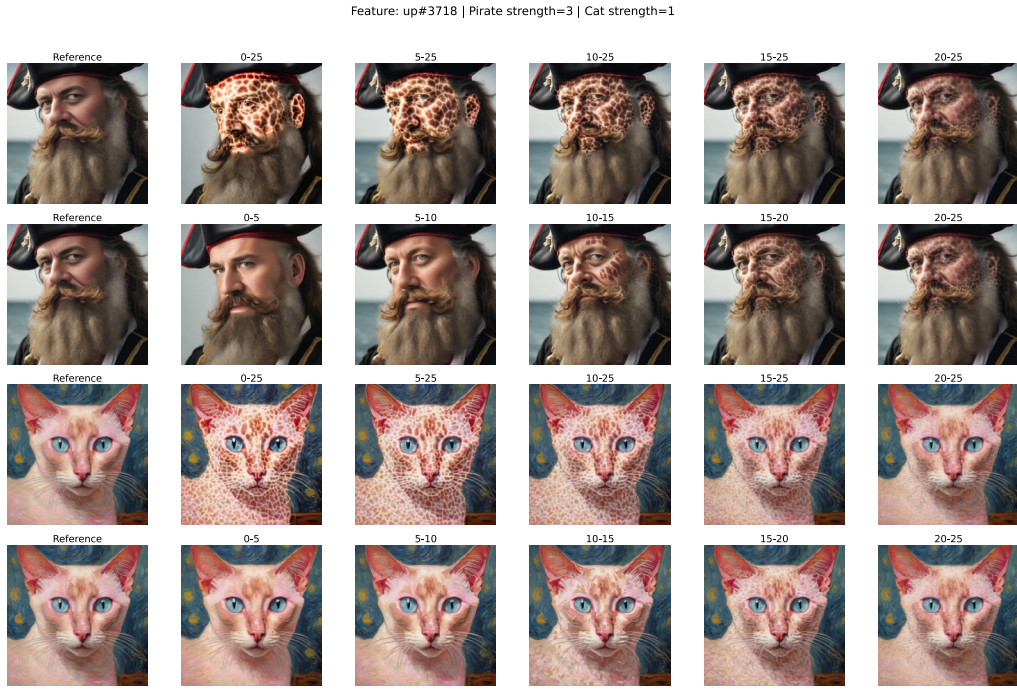
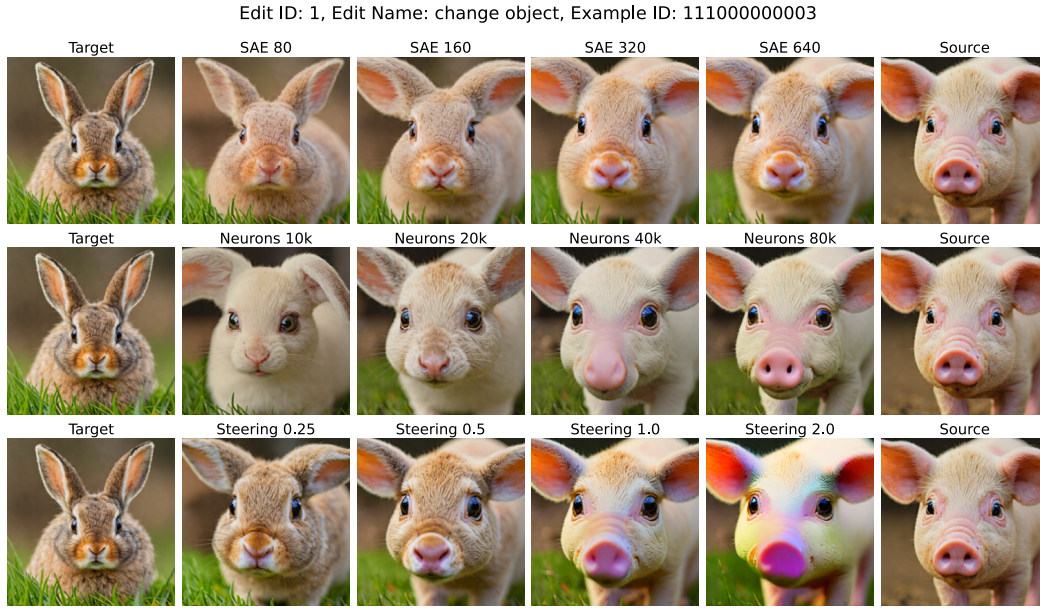


Figure 23: Performing interventions across different time intervals. For each prompt there are two rows, the first row contains ranges 0-25, 5-25, 10-25, 15-25, 20-25 and the second one 0-5, 5-10, 10-15, 15-20, 20-25. We would describe this feature as “giraffe pattern feature”. We intervened with this feature across the entire face of the pirate and across the entire cat except its ears. *These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ .*



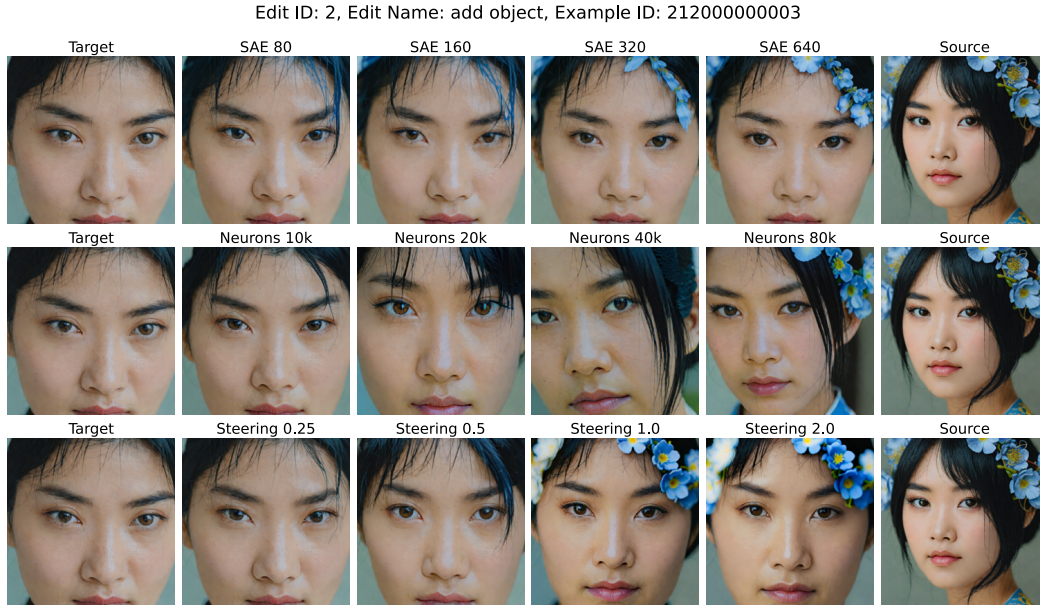


(a) Row 1 and row 2: varying number of SAE features / neurons transported; Row 3: steering with different strengths.



(b) Row 1 and row 2: strength 1 and strength 2 for SAE interventions. Row 3 and row 4. strength 1 and strength 2 for neuron interventions.

Figure 24: Example for edit category 1: “change object”. Original prompt (target): “a cute little bunny with big eyes”, edit prompt (source): “a cute little pig with big eyes”. Source and target refers to from where we extract features (source) and where we insert them (target). Grounded SAM2 masks used to collect the features are not shown but in this example they would select the entire foreground objects respectively.



(a) Row 1 and row 2: varying number of SAE features / neurons transported; Row 3: steering with different strengths.



(b) Row 1 and row 2: strength 1 and strength 2 for SAE interventions. Row 3 and row 4. strength 1 and strength 2 for neuron interventions.

Figure 25: Example for edit category 2: “add object”. Original prompt (target): “an Asian woman with blue thick-lashed eyes and black hair”, edit prompt (source): “an Asian woman with blue thick-lashed eyes and flowers on her black hair”. Source and target refers to from where we extract features (source) and where we insert them (target). Grounded SAM2 masks used to collect the features are not shown but in this example they would select the woman’s hair in the target and the flowers in the source forward pass.





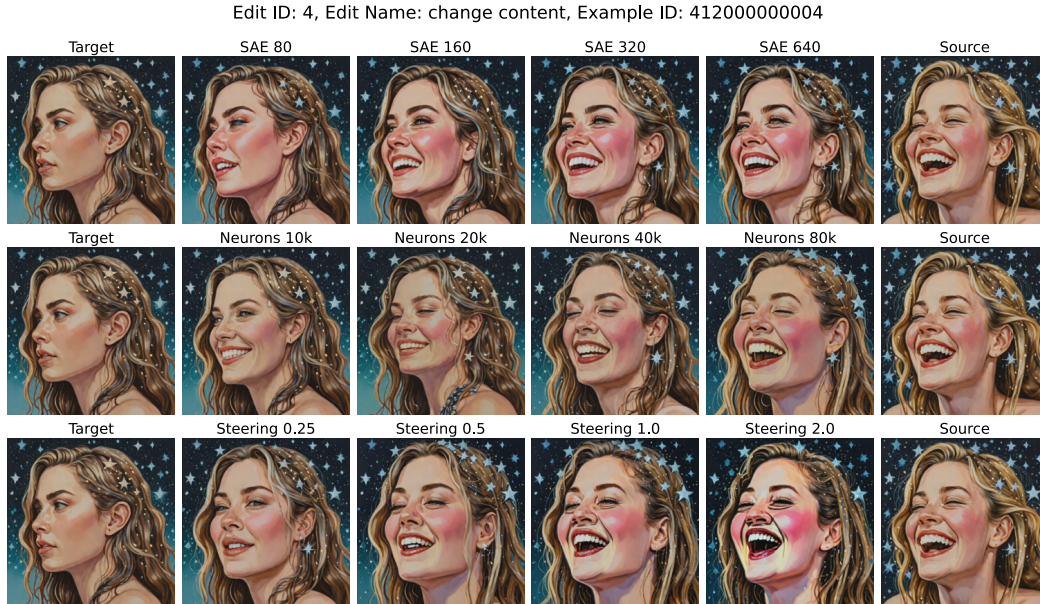
(a) Row 1 and row 2: varying number of SAE features / neurons transported; Row 3: steering with different strengths.



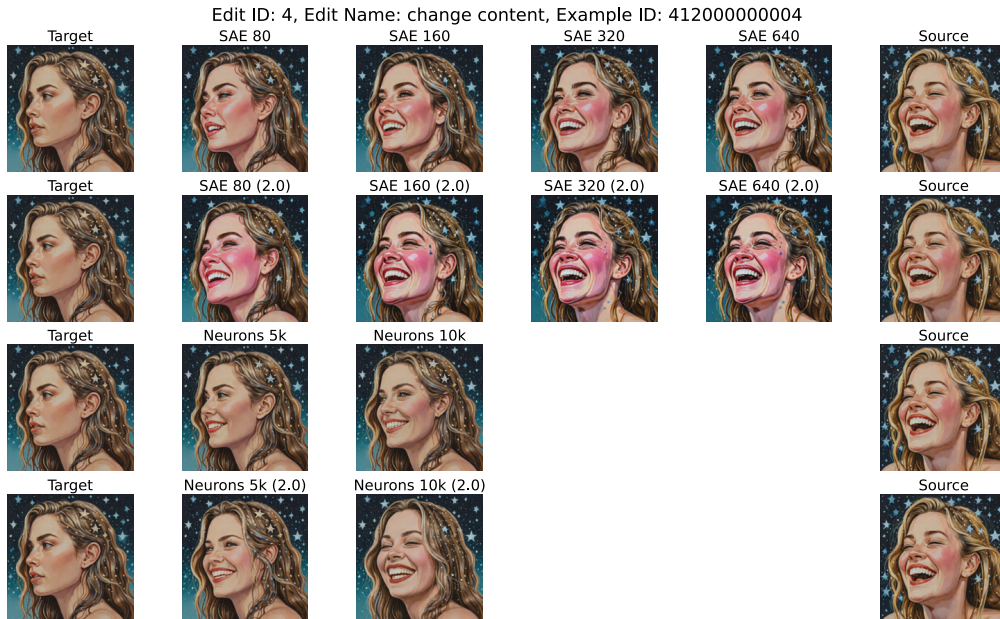
(b) Row 1 and row 2: strength 1 and strength 2 for SAE interventions. Row 3 and row 4. strength 1 and strength 2 for neuron interventions.

Figure 26: Example for edit category 3: “delete object”. Original prompt (target): “a lion in a suit sitting at a table with a laptop”, edit prompt (source): “a lion in a suit sitting at a table”. Source and target refers to from where we extract features (source) and where we insert them (target). Grounded SAM2 masks used to collect the features are not shown but in this example they would select the table in the source forward pass and the laptop in the target one. This example showcases a frequent failure mode of our intervention where the deleted object (re)appears in a different location in the image.



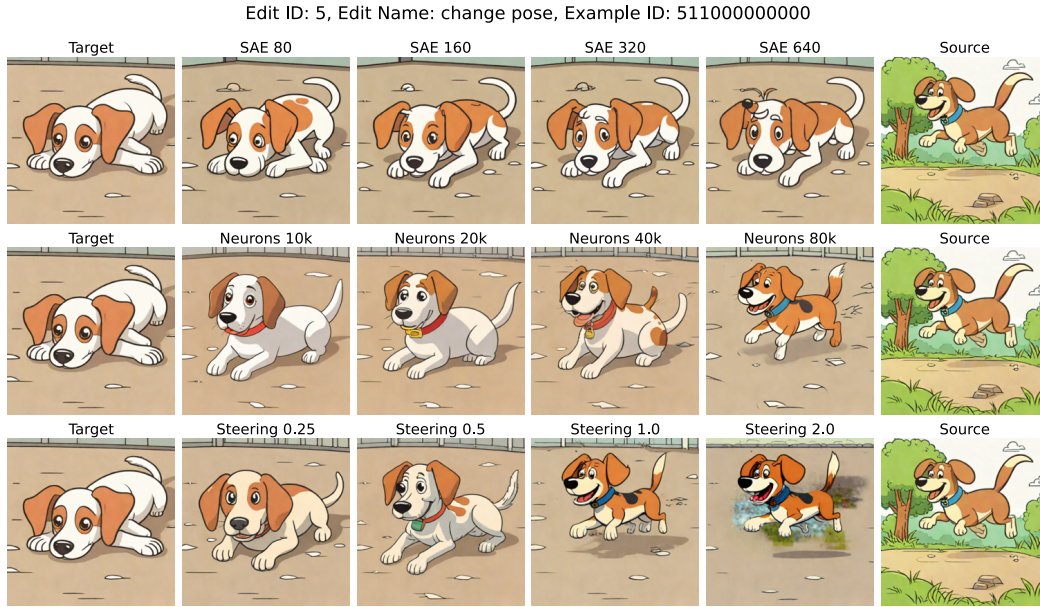


(a) Row 1 and row 2: varying number of SAE features / neurons transported; Row 3: steering with different strengths.



(b) Row 1 and row 2: strength 1 and strength 2 for SAE interventions. Row 3 and row 4. strength 1 and strength 2 for neuron interventions.

Figure 27: Example for edit category 4: “change content”. Original prompt (target): “a detailed oil painting of a calm beautiful woman with stars in her hair”, edit prompt (source): “a detailed oil painting of a laughing beautiful woman with stars in her hair”. Source and target refers to from where we extract features (source) and where we insert them (target). Grounded SAM2 masks used to collect the features are not shown but in this example they would select the woman’s face in both forward passes.



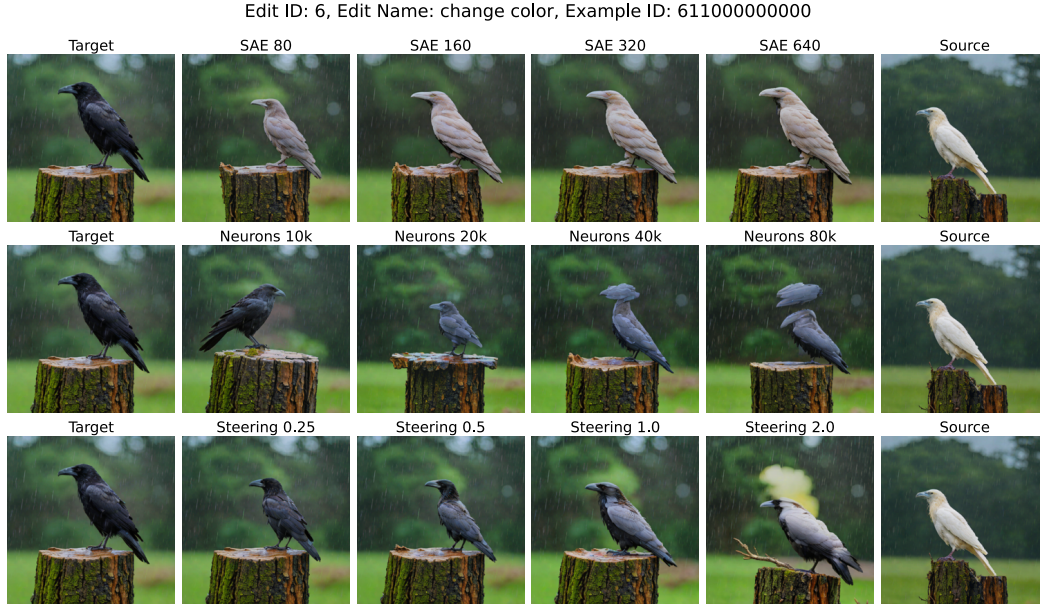
(a) Row 1 and row 2: varying number of SAE features / neurons transported; Row 3: steering with different strengths.



(b) Row 1 and row 2: strength 1 and strength 2 for SAE interventions. Row 3 and row 4. strength 1 and strength 2 for neuron interventions.

Figure 28: Example for edit category 5: “change pose”. Original prompt (target): “a cartoon dog laying down on the ground”, edit prompt (source): “a cartoon dog jumping up from the ground”. Source and target refers to from where we extract features (source) and where we insert them (target). Grounded SAM2 masks used to collect the features are not shown but in this example they would select the dogs in both forward passes.



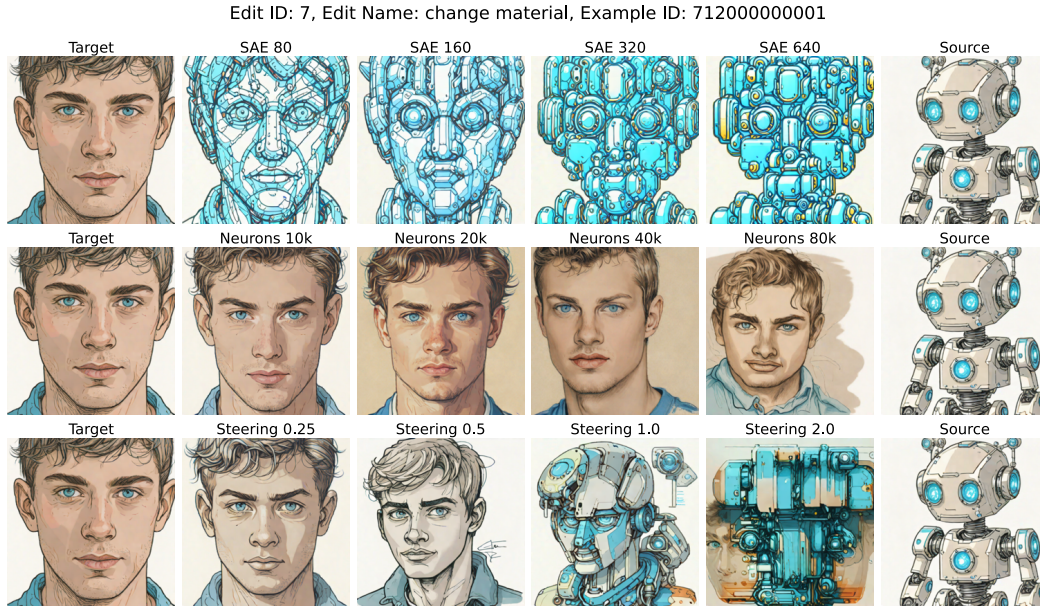


(a) Row 1 and row 2: varying number of SAE features / neurons transported; Row 3: steering with different strengths.

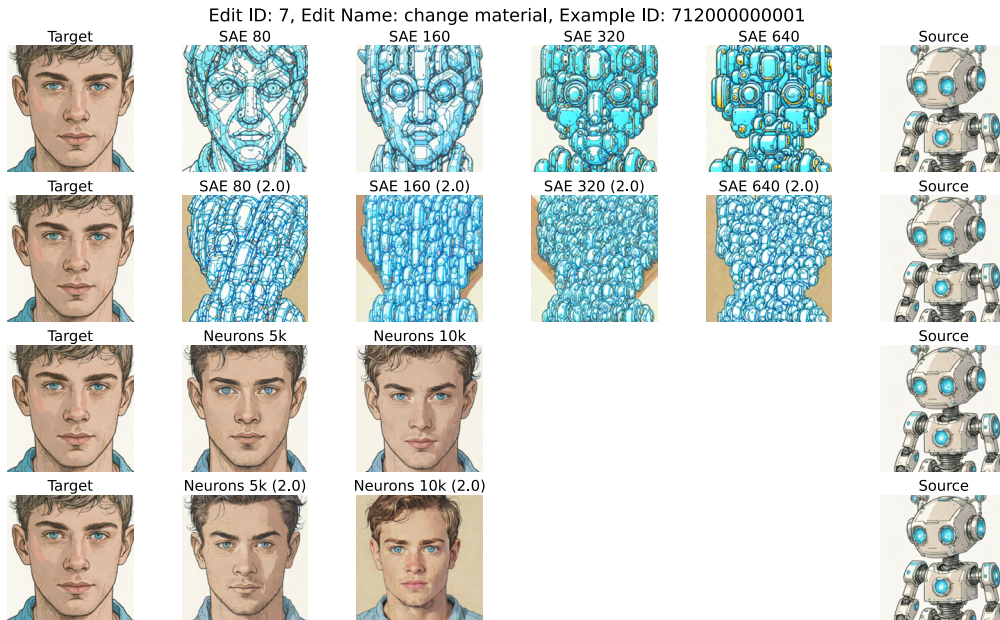


(b) Row 1 and row 2: strength 1 and strength 2 for SAE interventions. Row 3 and row 4. strength 1 and strength 2 for neuron interventions.

Figure 29: Example for edit category 6: “change color”. Original prompt (target): “a black raven sits on a tree stump in the rain”, edit prompt (source): “a white raven sits on a tree stump in the rain”. Source and target refers to from where we extract features (source) and where we insert them (target). Grounded SAM2 masks used to collect the features are not shown but in this example they would select the ravens in both forward passes. In this edit category we used the target mask to insert the features from source. Thus, we lose spatial information from source (the masks don’t agree so we have to aggregate).



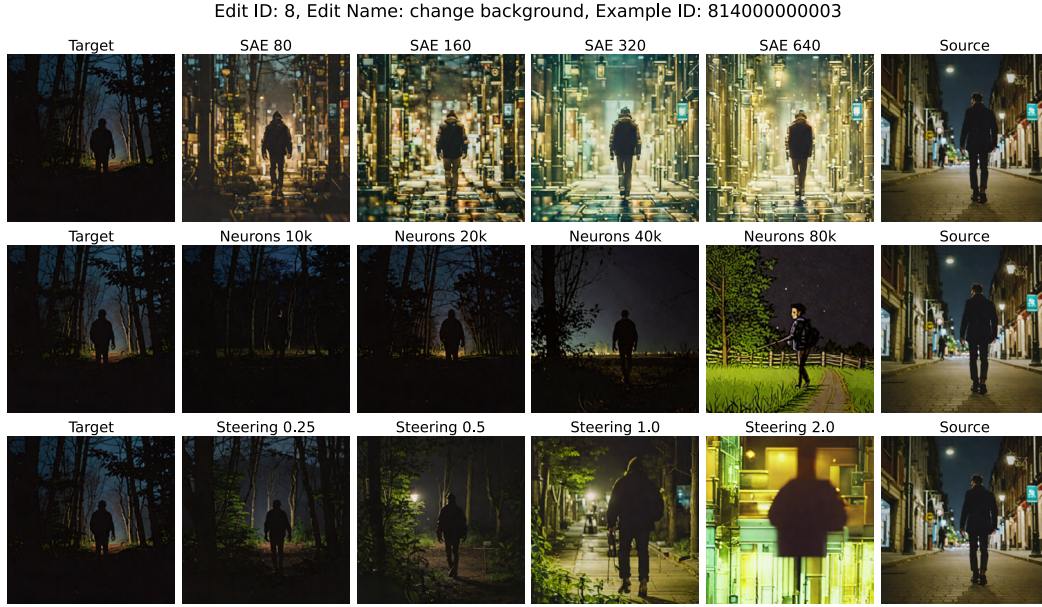
(a) Row 1 and row 2: varying number of SAE features / neurons transported; Row 3: steering with different strengths.



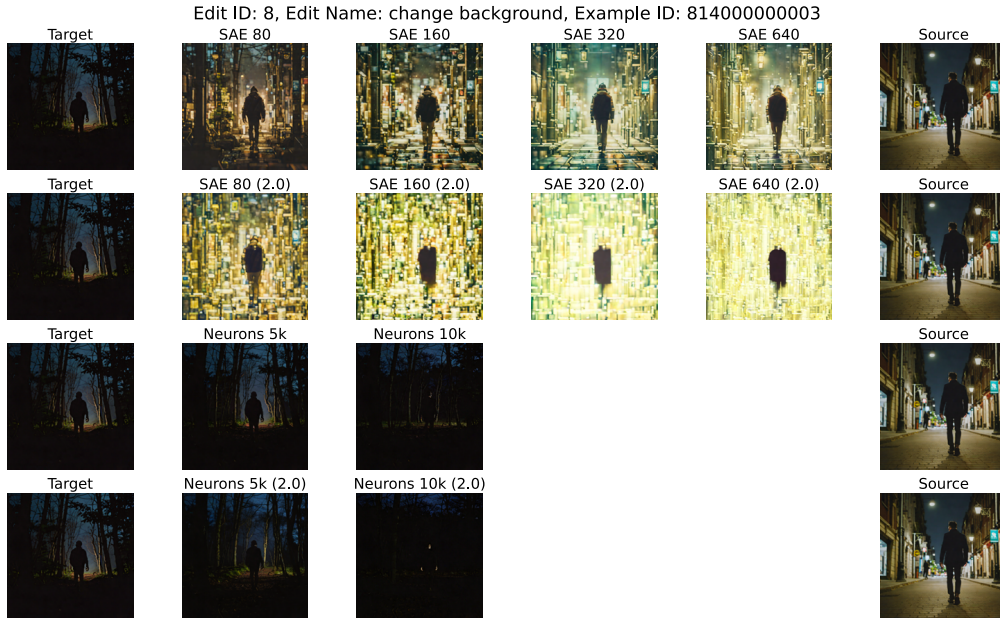
(b) Row 1 and row 2: strength 1 and strength 2 for SAE interventions. Row 3 and row 4. strength 1 and strength 2 for neuron interventions.

Figure 30: Example for edit category 7: “change material”. Original prompt (target): “a drawing of a young man with blue eyes”, edit prompt (source): “a drawing of a young robot with blue eyes”. Source and target refers to from where we extract features (source) and where we insert them (target). Grounded SAM2 masks used to collect the features are not shown but in this example they would select the face of the man in the target and the robot in the source forward pass. In this edit category we used the target mask to insert the features from source. Thus, we lose spatial information from source (the masks don’t agree so we have to aggregate).





(a) Row 1 and row 2: varying number of SAE features / neurons transported; Row 3: steering with different strengths.

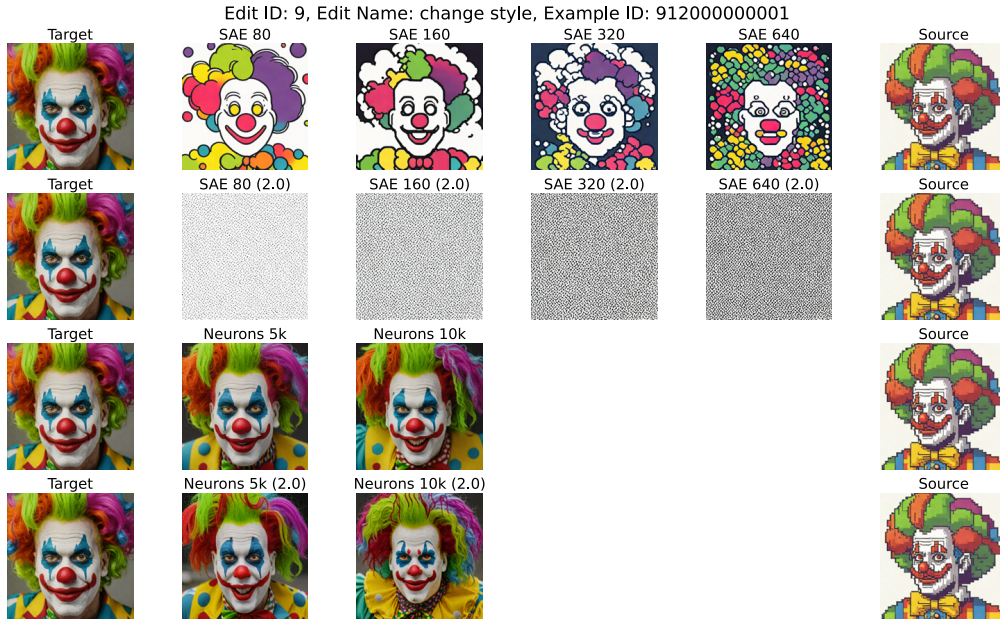


(b) Row 1 and row 2: strength 1 and strength 2 for SAE interventions. Row 3 and row 4. strength 1 and strength 2 for neuron interventions.

Figure 31: Example for edit category 8: “change background”. Original prompt (target): “a man walking in the woods at night”, edit prompt (source): “a man walking in the city at night”. Source and target refers to from where we extract features (source) and where we insert them (target). Grounded SAM2 masks used to collect the features are not shown but in this example they would select the backgrounds in both forward passes. In this edit category we used the target mask to insert the features from source. Thus, we lose spatial information from source (the masks don’t agree so we have to aggregate).



(a) Row 1 and row 2: varying number of SAE features / neurons transported; Row 3: steering with different strengths.



(b) Row 1 and row 2: strength 1 and strength 2 for SAE interventions. Row 3 and row 4. strength 1 and strength 2 for neuron interventions.

Figure 32: Example for edit category 9: “change style”. Original prompt (target): “a photograph a clown with colorful hair”, edit prompt (source): “a clown in pixel art style with colorful hair”. Source and target refers to from where we extract features (source) and where we insert them (target). In this edit category we used features from the entire spatial grid. However, we don’t keep the spatial information from source and instead set the features the the same value across the grid. From this example it becomes clear that for this edit category we should select fewer features and probably a lower strength.



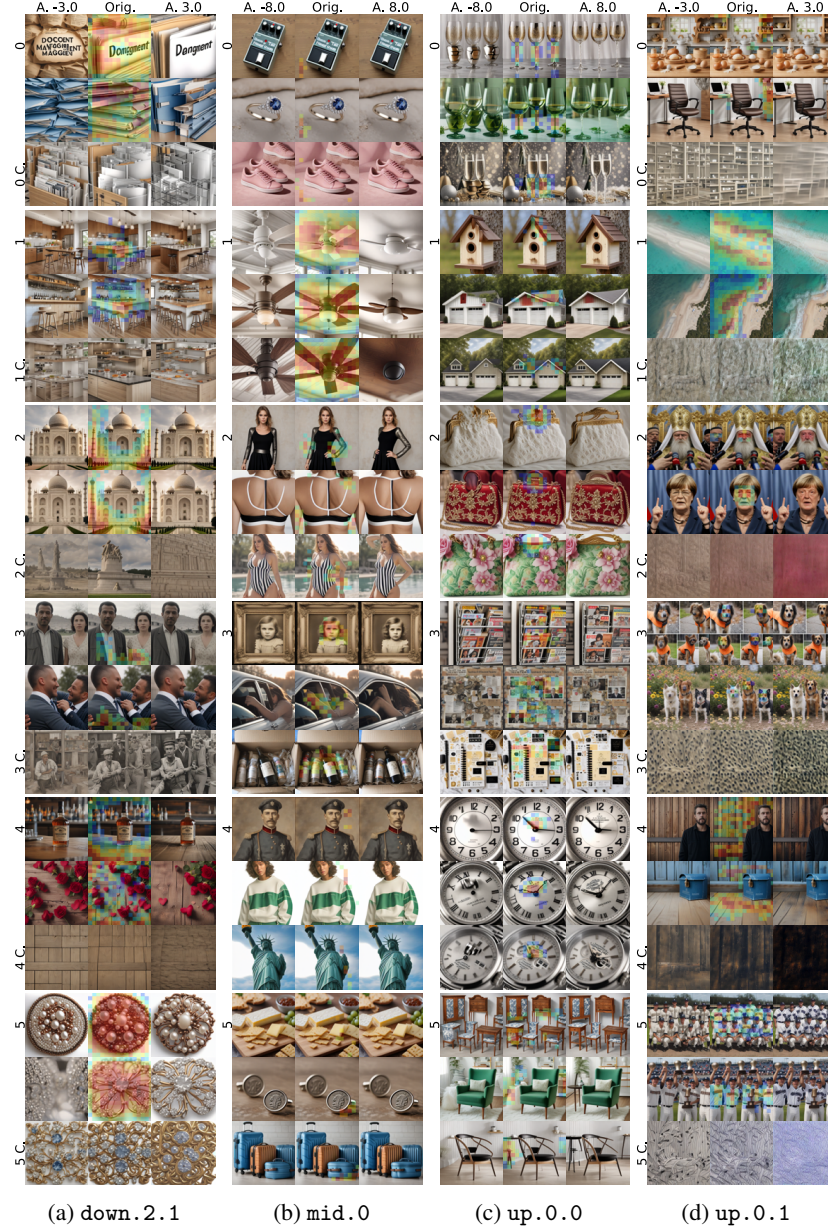


Figure 33: We visualize 6 features for down.2.1 (a), mid.0 (b), up.0.0, and up.0.1. We use three columns for each transformer block and three rows for each feature. For down.2.1 and up.0.1 we visualize the two samples from the top 5% quantile of activating dataset examples (middle) together a feature ablation (left) and a feature enhancement (right), and, activate the feature on the empty prompt with  $\gamma = 0.5, 1, 2$  from left to right. For mid.0 and up.0.0 we display three samples with ablation and enhancement. Captions are in Table 5. These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ .

Table 5: Prompts for the top 5% quantile examples in Fig. 33

Block	Feature	Prompt
down.2.1	0	A file folder with the word document management on it.
	0	Two blue folders filled with dividers.
	1	A kitchen with an island and bar stools.
	1	An unfinished bar with stools and a wood counter.
	2	The Taj Mahal, or a white marble building in India.
	2	The Taj Mahal, or a white marble building in India.
	3	A man and woman standing next to each other.
	3	Two men in suits hugging each other outside.
	4	An old Forester whiskey bottle sitting on top of a wooden table.
	4	Red roses and hearts on a wooden table.
	5	A beaded brooch with pearls and copper.
	5	An image of a brooch with diamonds.
mid.0	0	The Boss TS-3W pedal has an electronic tuner.
	0	An engagement ring with blue sapphire and diamonds.
	0	The women's pink sneaker is shown.
	1	A white ceiling fan with three blades.
	1	A ceiling fan with three blades and a light.
	1	The ceiling fan is dark brown and has two wooden blades.
	2	The black dress is made from knit and has metallic sleeves.
	2	The back view of a woman wearing a black and white sports bra.
	2	The woman is wearing a striped swimsuit.
	3	An old-fashioned photo frame with a little girl on it.
	3	The woman is sitting in her car with her head down.
	3	The contents of an empty bottle in a box.
	4	An old painting of a man in uniform.
	4	The model wears an off-white sweatshirt with green panel.
	4	The Statue of Liberty stands tall in front of a blue sky.
	5	Cheese and crackers on a cutting board.
	5	Two cufflinks with coins on them.
	5	Three pieces of luggage are shown in blue.
up.0.0	0	Three wine glasses with gold and silver designs.
	0	Three green wine glasses sitting next to each other.
	0	New Year's Eve with champagne, gold, and silver.
	1	The birdhouse is made from wood and has a brown roof.
	1	The garage is white with red shutters.
	1	Two garages with one attached porch and the other on either side.
	2	An elegant white lace purse with gold clasp.
	2	The red handbag has gold and silver designs.
	2	A pink and green floral-colored purse.
	3	A magazine rack with magazines on it.
	3	The year-in-review page for this digital scrap.
	3	The planner sticker kit is shown with gold and black accessories.
	4	A clock with numbers on the face.
	4	A silver watch with roman numerals on the face.
	4	An automatic watch with a silver dial.
	5	Four pieces of wooden furniture with blue and white designs.
	5	The green chair is in front of a white rug.
	5	The wish chair with a black seat.
up.0.1	0	The wooden toy kitchen set includes bread, eggs, and flour.
	0	The office chair is brown and black.
	1	An aerial view of the white sand and turquoise water.
	1	An aerial view of the beach and ocean.
	2	The patriarch of Ukraine is shown speaking to reporters.
	2	German Chancellor Merkel gestures as she speaks to the media.
	3	Four pictures showing dogs wearing orange vests.
	3	Two dogs are standing on the ground next to flowers.
	4	A man standing in front of a wooden wall.
	4	A blue mailbox sitting on top of a wooden floor.
	5	The baseball players are posing for a team photo.
	5	The baseball players are holding up their trophies.



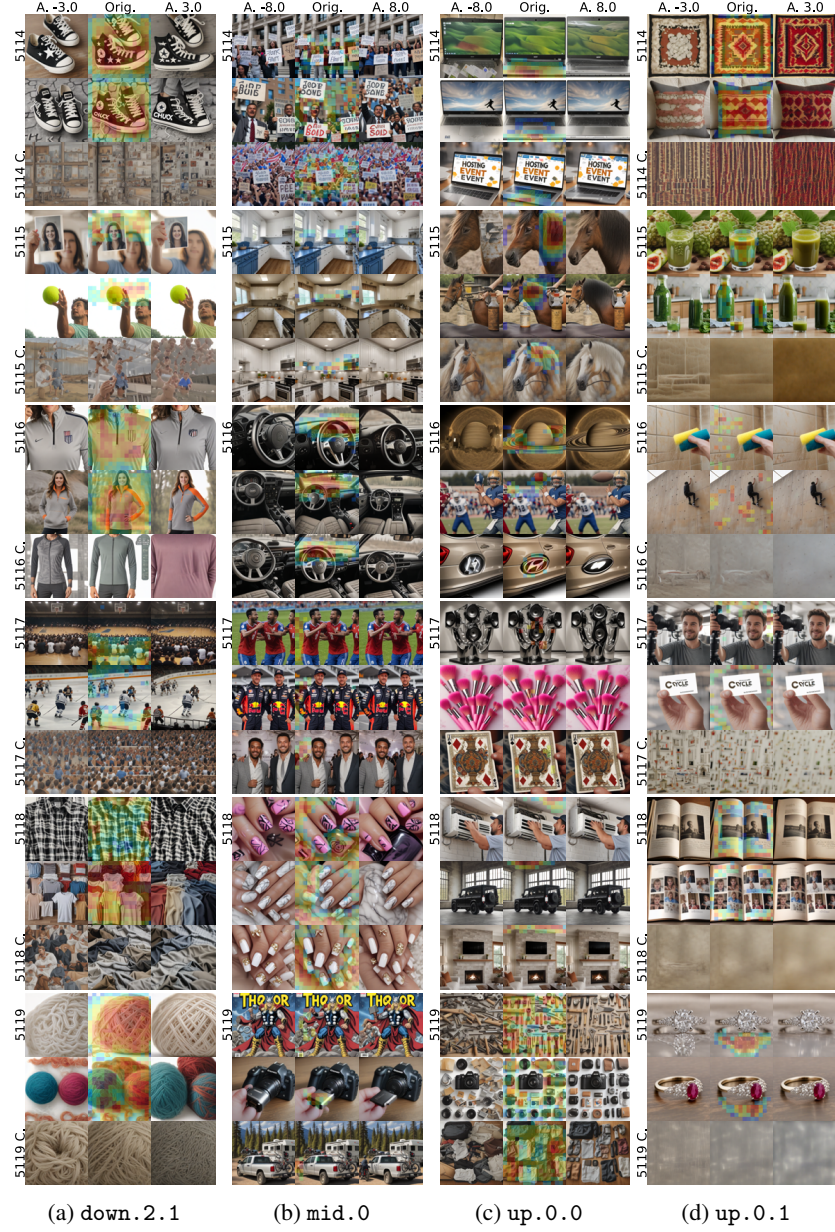


Figure 34: We visualize last 6 features for down.2.1 (a), mid.0 (b), up.0.0, and up.0.1. We use three columns for each transformer block and three rows for each feature. For down.2.1 and up.0.1 we visualize two samples from the top 5% quantile of activating dataset examples (middle) together a feature ablation (left) and a feature enhancement (right), and, activate the feature on the empty prompt with  $\gamma = 0.5, 1, 2$  from left to right. For mid.0 and up.0.0 we display three samples with ablation and enhancement. Captions are in Table 6. These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ .

Table 6: Prompts for the top 5 % quantile examples in Fig. 34

Block	Feature	Prompt
down.2.1	5114	Black and white Converse sneakers with the word black star.
	5114	Black and white Converse sneakers with the word Chuck.
	5115	A woman holding up a photo of herself.
	5115	A man holding up a tennis ball in the air.
	5116	The Nike Women's U.S. Soccer Team DRI-Fit 1/4 Zip Top.
	5116	The women's gray and orange half-zip sweatshirt.
	5117	A large group of people sitting in front of a basketball court.
	5117	Hockey players are playing in an arena with spectators.
	5118	The black and white plaid shirt is shown.
	5118	The different colors and sizes of t-shirts.
	5119	A ball of yarn on a white background.
	5119	Two balls of colored wool are on the white surface.
mid.0	5114	People holding signs in front of a building.
	5114	Two men dressed in suits and ties are holding up signs.
	5114	A large group of people holding flags and signs.
	5115	A kitchen with white cabinets and a blue stove.
	5115	The kitchen is clean and ready for us to use.
	5115	A kitchen with white cabinets and stainless steel appliances.
	5116	The steering wheel and dashboard in a car.
	5116	The interior of a car with dashboard controls.
	5116	The dashboard and steering wheel in a car.
	5117	Three men are celebrating a goal on the field.
	5117	Two men in Red Bull racing gear standing next to each other.
	5117	Two men are posing for the camera at an event.
	5118	Someone is holding up their nail polish with pink and black designs.
	5118	The nail is very cute and looks great with marble.
	5118	White stily nails with gold and diamonds.
	5119	The Mighty Thor comic book.
	5119	The camera is showing its flash drive.
	5119	A truck with bikes on the back parked next to a camper.
up.0.0	5114	The Acer laptop is open and ready to use.
	5114	The Lenovo S13 laptop is open and has an image of a person jumping off the keyboard.
	5114	A laptop with the words Hosting Event on it.
	5115	A horse with a black nose and brown mane.
	5115	The horse leather oil is being used to protect horses.
	5115	An oil painting on a canvas of a horse.
	5116	The sun is shining brightly over Saturn.
	5116	A football player throws the ball to another team.
	5116	Car door light logo sticker for Hyundai.
	5117	An artistic black and silver sculpture with speakers.
	5117	The pink brushes are sitting on top of each other.
	5117	Four kings playing cards in the hand.
	5118	A man is fixing an air conditioner.
	5118	The black Land Rover is parked in front of a large window.
	5118	A flat screen TV mounted on the wall above a fireplace.
	5119	A table with many different tools on it.
	5119	A camera with many different items including flash cards, lenses, and other accessories.
	5119	The contents of an open suitcase and some clothes.
up.0.1	5114	An old Navajo rug with multicolored designs.
	5114	The pillow is made from an old kilim.
	5115	An image of noni juice with some fruits.
	5115	A bottle and glass on the counter with green juice.
	5116	Someone cleaning the shower with a sponge.
	5116	A man on a skateboard climbing a wall with ropes.
	5117	A man taking a selfie in front of some camera equipment.
	5117	A person holding up a business card with the words cycle transportation.
	5118	Two photos are placed on top of an open book.
	5118	An open book with pictures of children and their parents.
	5119	An engagement ring with diamonds on top.
	5119	An oval ruby and diamond ring.

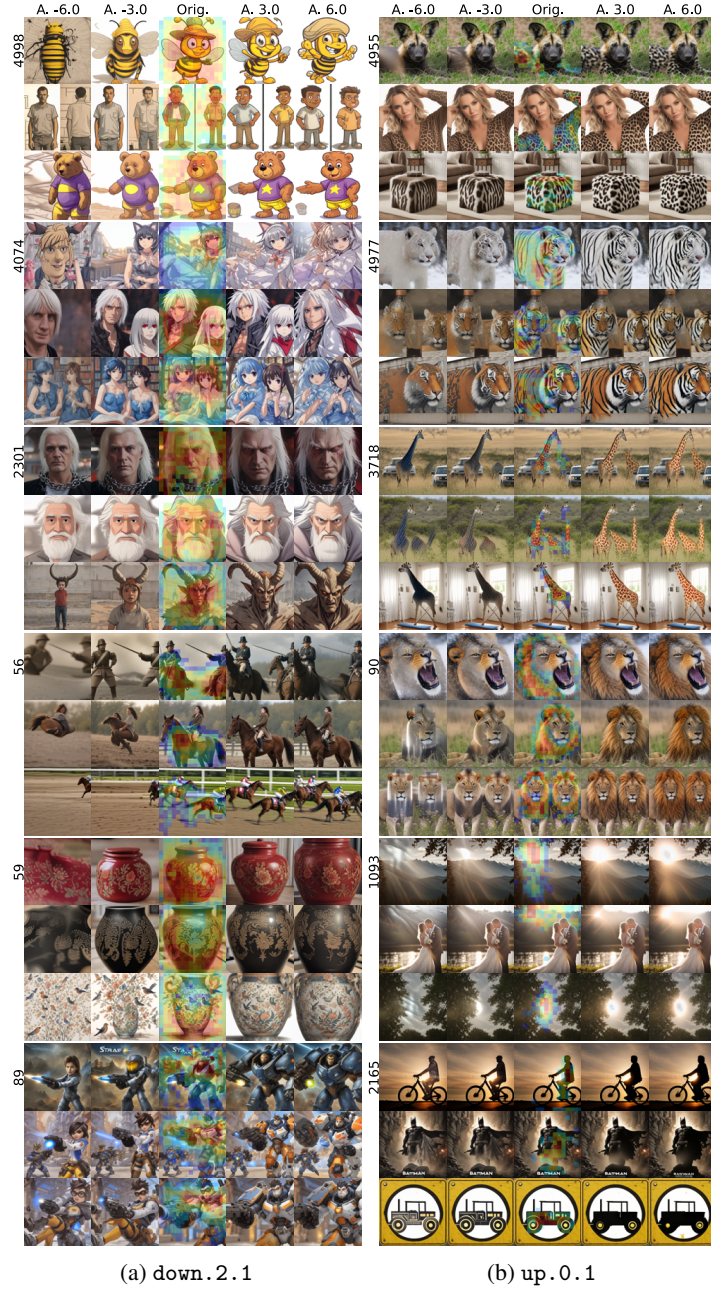


Figure 35: We visualize 6 features for down.2.1 (a) and up.0.1 (b). For each feature, we use 5 columns showing ablations (left), activating examples (middle), enhancements (right) and 3 rows with different samples from the top 5% quantile of activating examples. Captions are in Table 7. These results are from our first working SAE's with  $k = 10$  and  $n_f = 5120$ .



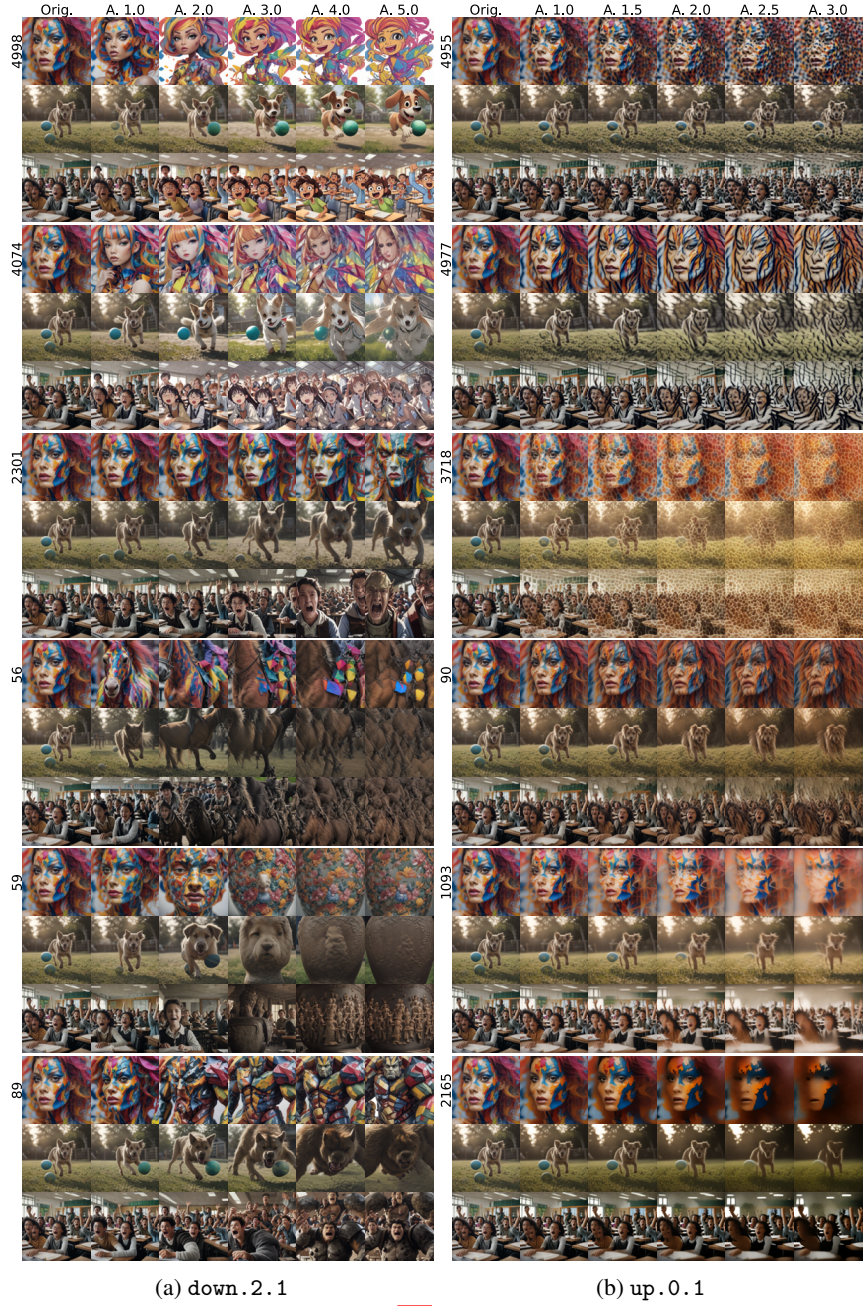


Figure 36: We turn on the features from Fig. 35 on three unrelated prompts “a photo of a colorful model”, “a cinematic shot of a dog playing with a ball”, and “a cinematic shot of a classroom with excited students”. *These results are from our first working SAE’s with  $k = 10$  and  $n_f = 5120$ .*



Table 7: Prompts for the top 5% quantile examples in Fig. 35

Block	Feature	Prompt
down.2.1	4998	A cartoon bee wearing a hat and holding something.
	4998	Two cartoon pictures of the same man with his hands in his pockets.
	4998	A cartoon bear with a purple shirt and yellow shorts.
	4074	An anime character with cat ears and a dress.
	4074	Two anime characters, one with white hair and the other with red eyes.
	4074	An anime book with two women in blue dresses.
	2301	A man with white hair and red eyes holding a chain.
	2301	An animated man with white hair and a beard.
	2301	The character is standing with horns on his head.
	56	Two men in uniforms riding horses with swords.
	56	A woman riding on the back of a brown horse.
	56	Two jockeys on horses racing down the track.
	59	A red jar with floral designs on it.
	59	An old black vase with some design on it.
	59	A vase with birds and flowers on it.
	89	StarCraft 2 is coming to the Nintendo Wii.
	89	Overwatch is coming to Xbox and PS3.
	89	The hero in Overwatch is holding his weapon.
up.0.1	4955	An African wild dog laying in the grass.
	4955	The woman is posing for a photo in her leopard print top.
	4955	An animal print cube ottoman with brown and white fur.
	4977	A white tiger with blue eyes standing in the snow.
	4977	A bottle and tiger are shown next to each other.
	4977	A mural on the side of a building with a tiger.
	3718	Giraffes are standing in the grass near a vehicle.
	3718	Two giraffes standing next to each other in the grass.
	3718	A giraffe standing next to an ironing board.
	90	A lion is roaring its teeth in the snow.
	90	A lion sitting in the grass looking off into the distance.
	90	Two lions with flowers on their backs.
	1093	The sun is shining over mountains and trees.
	1093	Bride and groom in front of a lake with sun flare.
	1093	The milky sun is shining brightly over the trees.
	2165	The silhouette of a person riding a bike at sunset.
	2165	The Dark Knight rises from his cave in Batman's poster.
	2165	A yellow sign with black design depicting a tractor.