

Look But Don't Touch with Sparse Autoencoders for Unlearning in Diffusion Models

Enrico Cassano¹, Riccardo Renzulli¹, Rayyan Ahmed², Marco Grangetto¹, and Stephan Alaniz²

¹ University of Turin, Computer Science Department, Italy
`{name.surname}@unito.it`

² LTCI, Télécom Paris, Institut Polytechnique de Paris, France
`{name.surname}@telecom-paris.fr`

Abstract. Sparse autoencoders (SAEs) have recently been proposed as interpretable tools for concept-level manipulation, under the assumption that isolated features can serve as controllable intervention points. In this work, we systematically evaluate this assumption in the context of object erasure and steering in diffusion models. We show that while SAEs reliably detect and localize semantic concepts within diffusion model activations, direct intervention in their latent space frequently induces out-of-distribution activations, resulting in severe visual artifacts. To disentangle detection from intervention, we use SAE activations purely as semantic detectors to identify image regions containing the target object, and replace those patch embeddings with the ones that do not contain it. This detection-based replacement preserves the diffusion model's activation statistics and produces significantly cleaner erasure results than latent steering. Our findings reveal a fundamental gap between concept detection and concept intervention in diffusion models: monosemantic or sparse features are not inherently suitable as control knobs for steering. These results position SAEs as powerful interpretability tools for analyzing generative models, but highlight important limitations when used for direct manipulation, such as unlearning. We release the code at <https://eidoslab.github.io/PER/>.

Keywords: Unlearning · Sparse autoencoders · Diffusion models

1 Introduction

Controlling content generation in text-to-image models [18] has become increasingly critical as these systems are deployed at scale. When models produce explicit content, copyrighted works, or other undesirable outputs, the ability to selectively erase specific concepts, ensuring the model can no longer represent or generate them, becomes essential [26]. A core challenge in machine unlearning is identifying where and how concepts are represented inside diffusion models (DMs), particularly given the phenomenon of *polysemanticity*, where each neuron can encode multiple unrelated concepts simultaneously [2]. Sparse Autoencoders (SAEs) decompose dense neural activations into sparse, *monosemantic* latents

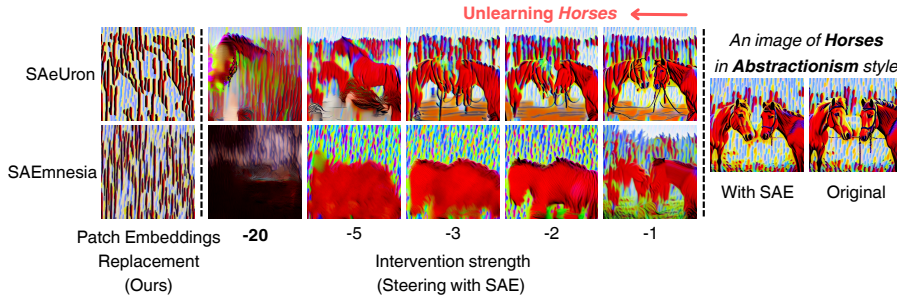


Fig. 1: Effect of SAE-based activation steering under varying intervention strengths and feature patch replacement when unlearning “Horses”. Both SAEUron (top) and SAEEmnesia (bottom) exhibit severe visual artifacts at large negative multipliers, while weaker interventions fail to fully erase the concept. Multipliers highlighted in bold are the values selected by each method for horse unlearning.

that activate almost exclusively for specific concepts and serve as the fundamental units of neural network representations [1]. Recent work has explored SAEs as a tool for interpretable concept erasure [3, 4] and editing [11, 20, 21] in DMs. These methods all follow a similar pipeline: (i) encode DM activations through the SAE encoder and (ii) steer activations with the SAE latents that represent the concepts to erase or edit. This approach offers interpretability advantages, since we can precisely identify which latents represent which concepts. However, we can observe in Fig. 1 that steering with SAEs produces systematic artifacts when unlearning the concept “Horses” from Stable Diffusion (SD) v1.5 [18] fine-tuned on UnlearnCanvas [28], which is a stylized image dataset for benchmarking unlearning methods in DMs. In fact, concept erasure via SAE reconstruction frequently produces activations that lie outside the DM’s in-distribution (ID) manifold, resulting in visual artifacts. Nonetheless, SAE-based approaches excel at detecting the presence of a concept, an important step for unlearning, and thus present a promising alternative to traditional unlearning methods based on fine-tuning [7, 12, 27] or closed-form weight modifications [8]. While prior work has shown that SAEs excel at detection but underperform at steering in language models [25], we hereby systematically characterize a concrete failure mode in SAE-based unlearning for DMs and provide a concrete alternative. We show that using SAEs only for concept detection rather than direct manipulation avoids out-of-distribution (OOD) activations. For each representation of the DM containing the unwanted concept, we replace the concept-containing embeddings with concept-free ones drawn from the same feature map. By doing so, we remove unwanted concepts while maintaining visual coherence. Moreover, every activation in our processed output originates from the DM’s forward pass, guaranteeing that all activations are ID within the DM. Our key contributions can be summarized as follows: ❶ We show that SAE reconstruction under latent intervention produces activations outside the DM’s manifold (Sec. 4.2).

② We introduce a detection-based framework for SAE-guided concept removal through patch embeddings replacement (PER), avoiding the distributional mismatch (Sec. 3.5). ③ Our method improves the quality of the unlearned images, eliminates the costly grid search for steering intervention strength and scores up to an average of 95.33% on UnlearnCanvas, which is an improvement of 3.82% over state-of-the-art for SAE-based object unlearning (Sec. 4). We also achieve promising results in NSFW removal and adversarial attacks.

2 Related work

Concept erasure in DMs. Machine unlearning aims to remove the influence of particular data or concepts from a trained model without necessitating re-training, ensuring that the resulting model works as if the target information were never encountered during training [26]. We follow the established convention in which unlearning is framed as concept erasure [6], aiming to prevent the generation of specific concepts while maintaining overall generation quality and preserving unrelated concepts. The distributed and entangled representation of concepts in DMs makes this problem especially challenging. Recent work involving full model fine-tuning to unlearn concepts has explored various approaches: ESD [7] and CA [12] eliminate anchor concepts through fine-tuning with negative guidance; EDiff [24] formulates data forgetting as a constrained optimization problem; SA [10] substitutes unwanted data distributions with surrogate distributions. Fine-tuning-free methods include SalUn [5] and SHS [23], which identify parameters to modify using saliency maps or connection sensitivity; FMN [27], which introduces a re-steering loss applied exclusively to attention layers; SPM [14], which incorporates linear adapters to directly prevent unwanted content propagation; SEOT [13], which eliminates unwanted content from text embeddings; UCE [8], which modifies cross-attention weights via closed-form solutions. In contrast to these approaches, our method leverages SAEs to achieve interpretable concept erasure at inference time without modifying model weights, ensuring that activations remain within the DM’s training distribution.

Erasing concepts with SAEs in DMs. In recent years, the exploitation of SAEs for preventing the generation of specific concepts has become an emerging topic. Kim *et al.* [11] leverage SAEs in the latent space of text embeddings to precisely steer the generation away from a given concept. Cywiński and Deja [4] proposed SAeUron, which applies unsupervised SAEs after cross-attention layer in UNet-based DMs. While SAeUron achieves promising results when unlearning styles in UnlearnCanvas, its effectiveness when unlearning objects is lower. With unsupervised SAEs, concepts can still be distributed across many neurons. Therefore, Cassano *et al.* [3] introduced SAEmnesia, which trains SAEs with supervision to achieve higher feature centralization, binding each concept to a single, interpretable neuron. Similarly, Härle *et al.* [9] proposed Guided Sparse Autoencoders (G-SAEs), a method that conditions latent representations on labeled concepts to better understand and influence the internal dynamics of large language models (LLMs). In this work, we adapt G-SAEs to DMs. Surkov

et al. [20] use SAEs trained SDXL Turbo in its 1-step setting for editing images, but they struggle in the delete object task. In contrast to these works, our method uses SAEs exclusively for concept detection, identifying patches containing unwanted concepts and replacing them with concept-free alternatives from the same feature map, thereby ensuring that all activations remain within the DM’s training distribution while avoiding the distributional mismatch inherent in reconstruction-based approaches.

Limitations of SAE-based steering. While SAEs excel at decomposing neural activations into interpretable features, recent work reveals systematic limitations when using these features for steering. Wu *et al.* [25] introduce a comprehensive benchmark evaluating SAE-based steering against simple baselines across concept detection and model steering tasks. Their findings show how SAEs significantly underperform even basic methods and fail to approach the effectiveness of standard prompting or fine-tuning. They demonstrate that supervised dictionary learning methods consistently outperform unsupervised SAEs for both detection and intervention tasks. O’Brien *et al.* [17] show that editing SAE latent values to control refusal behavior in language models improves safety metrics but causes widespread performance degradation on unrelated benchmarks. Critically, they find this degradation occurs regardless of which features are manipulated, suggesting fundamental limitations in latents-based steering. Mayne *et al.* [16] investigate why SAE decompositions of steering vectors fail to preserve steering properties, identifying two core issues: steering vectors exhibit distributional properties incompatible with SAE training assumptions, and meaningful interventions require negative feature coefficients which SAEs cannot represent. Building on this prior evidence of SAE-based steering in language models, we identify an analogous limitation in DMs unlearning. To our knowledge, we are the first to systematically characterize this failure mode in SAE-based interventions for DMs, wherein steering via SAE features generates OOD activations that degrade both visual fidelity and unlearning effectiveness. We show that our methodology, PER, avoids this distributional drift by preserving the model’s learned activation manifold throughout the unlearning.

3 Unlearning with SAEs

Fig. 2 provides an overview of our approach. Instead of steering with SAE latents, we replace detected concept-containing patch embeddings with concept-free ones sampled from the same feature map, this removes the concept while keeping activations ID.

3.1 Training SAEs in DMs

In this work, we apply SAEs to the residual stream of cross-attention layers on the denoising network of DMs. At timestep t , we extract output feature maps $\mathbf{X}_t \in \mathbb{R}^{H \times W \times D}$, where H and W are the height and width of the feature map, and D is the hidden dimension. Each spatial position within the feature map

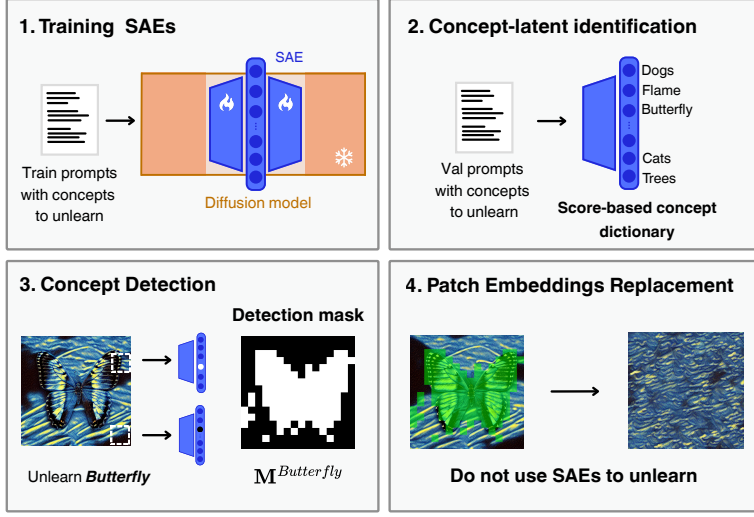


Fig. 2: Overview of the proposed pipeline. First, SAEs are trained on DM activations using prompts containing the concepts to be removed. Next, a score-based analysis identifies the SAE latents associated with each concept, forming a concept-latent dictionary. During inference, these latents are used to detect concept-containing patches, producing a spatial detection mask. Finally, instead of steering with SAEs, detected patch embeddings are replaced with embeddings sampled from non-detected locations, removing the concept while keeping activations in-distribution.

corresponds to a patch in the input image. We denote by $\mathbf{x} \in \mathbb{R}^D$ a generic patch embedding vector given as input to the SAE, which is composed of an encoder and a decoder structure. The encoder expands the dimension D to dimension N by a fixed factor. The encoding and decoding operations are formulated as [2]:

$$\mathbf{v} = \text{ReLU}(\mathbf{W}_{\text{enc}}(\mathbf{x} - \mathbf{b}_{\text{pre}}) + \mathbf{b}_{\text{enc}}), \quad \hat{\mathbf{x}} = \mathbf{W}_{\text{dec}}\mathbf{v} + \mathbf{b}_{\text{pre}}, \quad (1)$$

where $\mathbf{v}$ denotes sparse latent codes, $\hat{\mathbf{x}}$ represents the reconstructed input, and weight matrices $\mathbf{W}_{\text{enc}} \in \mathbb{R}^{N \times D}$ and $\mathbf{W}_{\text{dec}} \in \mathbb{R}^{D \times N}$ parameterize the encoder and decoder, while $\mathbf{b}_{\text{pre}} \in \mathbb{R}^D$ and $\mathbf{b}_{\text{enc}} \in \mathbb{R}^N$ are learnable bias terms. Instead of vanilla ReLU SAEs, we train TopK SAEs [15] that enforce sparsity by retaining only the k most strongly activated latents. This mechanism zeroes out all but the top k pre-activation values, maintaining the dominant features while guaranteeing a fixed sparsity level:

$$\mathbf{z} = \text{TopK}(\mathbf{v}), \quad \hat{\mathbf{x}} = \mathbf{W}_{\text{dec}}\mathbf{z} + \mathbf{b}_{\text{pre}}. \quad (2)$$

The vector $\mathbf{v}$ contains pre-TopK activations, while $\mathbf{z}$ holds the sparsified representation after TopK. Training TopK SAEs requires balancing reconstruction fidelity against latent usage. For a batch of B samples, the objective is:

$$\mathcal{L}_{\text{unsupSAE}} = \frac{1}{B} \sum_{b=1}^B \|\mathbf{x}^{(b)} - \hat{\mathbf{x}}^{(b)}\|_2^2 + \alpha \mathcal{L}_{\text{aux}}, \quad (3)$$

where the first term measures reconstruction quality and $\mathcal{L}_{\text{aux}}$ [4] addresses dead latents, *i.e.* features that rarely activate during training, weighted by scalar α . Additionally, when employing SAE_{mnesia} and G-SAEs, we also compute the corresponding proposed supervised losses to enforce concept-latent mapping.

3.2 Concept-latent identification

To identify which SAE latents correspond to a given concept, we follow Cywiński and Deja [4] and use a score function that measures the correspondence between a latent (*i.e.* neuron) and a concept. Given a dataset of activations $\mathcal{D} = \mathcal{D}_c \cup \mathcal{D}_{\neg c}$, where $\mathcal{D}_c$ are the activations containing the concept c and $\mathcal{D}_{\neg c}$ are the activations without it, the score for latent n at denoising timestep t is:

$$\text{score}(n, t, c, \mathcal{D}) = \frac{\mu(n, t, \mathcal{D}_c)}{\sum_{j=1}^N \mu(j, t, \mathcal{D}_c) + \delta} - \frac{\mu(n, t, \mathcal{D}_{\neg c})}{\sum_{j=1}^N \mu(j, t, \mathcal{D}_{\neg c}) + \delta} \quad (4)$$

where δ is a small constant to prevent division by zero and $\mu(n, t, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} z_n$ denotes the mean activation of the n -th latent over dataset $\mathcal{D}$ at timestep t (t is omitted from z_n for brevity). Intuitively, this score ranks all latents for a given concept c : latents with high scores activate strongly in the presence of c and weakly otherwise. We denote the set of top-scoring latents for concept c as:

$$\mathcal{F}_c = \{n : n \text{ is a top-scoring latent selected for concept } c\} \quad (5)$$

The size $|\mathcal{F}_c|$ depends on the underlying pipeline.

3.3 Concept detection

Let $\mathbf{Z} \in \mathbb{R}^{H \times W \times N}$ denote the SAE latent activations for all spatial locations, where $Z_{h,w,n}$ is the activation of latent n at position (h, w) . For a concept c , associated with a set of latents $\mathcal{F}_c$, we define a binary detection mask $\mathbf{M}^c \in \{0, 1\}^{H \times W}$ as

$$M_{h,w}^c = \mathbf{1}(\exists n \in \mathcal{F}_c : Z_{h,w,n} > \mu(n, t, \mathcal{D})), \quad (6)$$

where $\mu(n, t, \mathcal{D})$ denotes the average activation of latent n at timestep t over the validation set $\mathcal{D}$, and $\mathbf{1}(\cdot)$ is the indicator function. A spatial location is marked as concept-containing if at least one concept-associated latent exceeds its dataset mean activation. We employ this detection mask when steering G-SAEs [9] and SAEs on SDXL Turbo [20] and with PER.

3.4 Steering with SAEs

SAE-based steering modifies the contribution of concept-associated latents and re-injects the modified activation into the denoising network. We adopt two suppression strategies based on the SAE. The first strategy performs a direct additive displacement in activation space. As mentioned before, for G-SAEs and

SAEs trained on SDXL Turbo, we modify their original steering formulation with the binary detection mask as

$$\tilde{\mathbf{X}}_{h,w} = \mathbf{X}_{h,w} - A_{h,w} \mathbf{W}_{\text{dec}}[:, c] \quad (7)$$

where $\mathbf{A} = \gamma_c \mathbf{M}^c$ and γ_c is a negative multiplier, namely the intervention strength. For simplicity, we omit the timestep t in the notation and use c as the index of the decoder column responsible for concept c . This formulation applies a linear perturbation aligned with the learned feature directions. SAEUron and SAEemnesia employ an alternative suppression strategy that acts directly on $\mathbf{Z}$, replacing the activation with its SAE reconstruction. We define a latent-level gating mask as

$$G_{h,w,n} = \mathbf{1}(n \in \mathcal{F}_c \wedge Z_{h,w,n} > \mu(n, t, \mathcal{D})). \quad (8)$$

Selected latents are scaled by a negative multiplier γ_c , normalized by the average activation on concept samples $\mu(n, t, \mathcal{D}_c)$:

$$Z'_{h,w,n} = (1 - G_{h,w,n}) Z_{h,w,n} + G_{h,w,n} \gamma_c \mu(n, t, \mathcal{D}_c) Z_{h,w,n}. \quad (9)$$

The steered activation is then reconstructed through the decoder:

$$\tilde{\mathbf{X}}_{h,w} = \mathbf{W}_{\text{dec}} \mathbf{Z}'_{h,w,:} + \mathbf{b}_{\text{pre}}, \quad (10)$$

and replaces the original activation at the spatial position (h, w) . In both cases, steering relies on the decoder dictionary to define semantically meaningful directions in representation space. The key difference lies in whether the intervention applies a localized displacement directly to the denoising network's original activation or steers it in the SAE latent space.

3.5 Patch Embedding Replacement (PER)

We observed that current SAE-based unlearning steering techniques exhibit degenerate failure modes characterized by: 1) the generation of OOD activations, and 2) systematic visual artifacts resulting from these perturbations. We analyze these effects in detail in Sec. 4.2. Moreover, prior approaches require a grid search over the intervention strength γ_c for each concept c , introducing additional complexity and instability. In contrast, we proposed a simple Patch Embedding Replacement (PER) method that removes multiplier-based latent manipulation. Given the binary detection mask $\mathbf{M}^c \in \{0, 1\}^{H \times W}$ defined in Sec. 3.3, we erase concept c by replacing detected patch embeddings at locations (h, w) where $\mathbf{M}_{h,w}^c = 1$ with activations drawn from non-detected locations:

$$\begin{aligned} \tilde{\mathbf{X}}_{h,w} &= (1 - \mathbf{M}_{h,w}^c) \mathbf{X}_{h,w} + \mathbf{M}_{h,w}^c \mathbf{X}_{h',w'} \\ \text{where } (h', w') &\sim \mathcal{U}(\{(i, j) \mid \mathbf{M}_{i,j}^c = 0\}). \end{aligned} \quad (11)$$

Here, the spatial location (h', w') is uniformly sampled from positions not activating the concept. Since concept information may propagate to neighboring

patches through spatial correlations, we optionally dilate the detection mask using a padding factor $p \geq 0$. Specifically, for each detected location, all patches within a spatial distance p are also marked as concept-containing, yielding a dilated mask $\mathbf{M}^c(p)$. When $p = 0$, no dilation is applied. In the rare case where all spatial locations are detected (*i.e.*, $\mathbf{M}^c = \mathbf{1}$), the method falls back to Gaussian noise replacement. Other replacement strategies are reported in Appendix A.4. Importantly, all replacement activations originate from the same forward pass of the denoising network. As a result, $\tilde{\mathbf{X}}$ remains within the model’s native activation distribution, avoiding the distributional shift and the corresponding visual artifacts introduced by multiplier-based latent interventions.

4 Experiments and Results

Our experiments reveal the OOD behavior of existing SAE unlearning approaches, resulting in generated artifacts. Furthermore, we show that PER compares favorably across multiple dimensions: computational efficiency, unlearning effectiveness on the UnlearnCanvas benchmark and adversarial robustness. We apply PER on top of four existing SAE-based pipelines, SAeUron [4], SAEmnesia [3], G-SAE [9] and SAE trained on SDXL Turbo [20]. For a fair comparison, we compare PER in the experimental setups of each pipeline.

4.1 Experimental setup

SAEs and DMs. Although our pipeline includes SAE training as first step, we can directly adopt the pre-trained SAE models released by previous pipelines, when available. Note that, to the best of our knowledge, G-SAEs have not been applied to DMs unlearning before our work. We evaluate PER across different DM architectures: SD v1.5 [18], which serves as the primary testbed given that most existing erasure techniques are developed and benchmarked on this architecture, and the more recent SDXL Turbo, assessing the generality of our approach. For SD v1.5, we use the SAE on UNet block `up.1.1`, consistent with prior work [3, 4], while for SDXL Turbo we use block `up.0.1` [20].

Dataset and evaluation metrics. Our primary evaluation uses the UnlearnCanvas benchmark [28], which consists of 20 object classes and 50 artistic styles, with prompts of the form `An image of {object} in {style} style`. Vision Transformer-based classifiers are employed to measure three key metrics: 1) Unlearning Accuracy (UA), which quantifies the proportion of samples from target concept prompts that are misclassified (*i.e.* successful unlearning); 2) In-domain Retain Accuracy (IRA), measuring classification accuracy on retained concepts within the same domain; 3) Cross-domain Retain Accuracy (CRA), assessing accuracy on concepts from different domains. To quantify visual artifacts, we employ Qwen2-VL-7B-Instruct [22] as an automated evaluator to measure the Artifacts Rate (AR) across generated images on which unlearning was applied. Comparisons with other unlearning methodologies can be found in Sec. A.1 of the Appendix, and additional experiments performed for NSFW removal on the I2P benchmark [19] on SD v1.4 are reported in Sec. A.3 of the Appendix.

4.2 The brittleness of SAE-based Unlearning

We find that current SAE-based unlearning techniques suffer from degenerate failure modes and identify two primary issues: the generation of OOD activations and the resulting systematic visual artifacts.

Activation distribution analysis. To quantify the effect of SAE-based unlearning through negative multipliers, we collect patch-level activations with and without intervention. Following the setting of UnlearnCanvas, we generate one image for every object class c using the template `An image of {c}` and record the activations for every patch embedding at the SAE layer where the SAE detects the concept c . Further, we repeat the process and record the embeddings after SAE unlearning of concept c as described in Sec. 3. Since the negative multiplier γ_c is typically different for every concept c , we collect embeddings for the minimum, median and maximum value of γ on UnlearnCanvas to capture the range of the manipulated embedding space distributions. We plot the per-dimension activation distributions (Fig. 3 top) and find that using the median negative multiplier produces a wider and flatter activation profile, with 27.6%, 28%, 18.8%, and 83.7% of values falling outside the original distributions for SAeUron, SAEmnesia, G-SAE, and SDXL Turbo respectively. This distri-

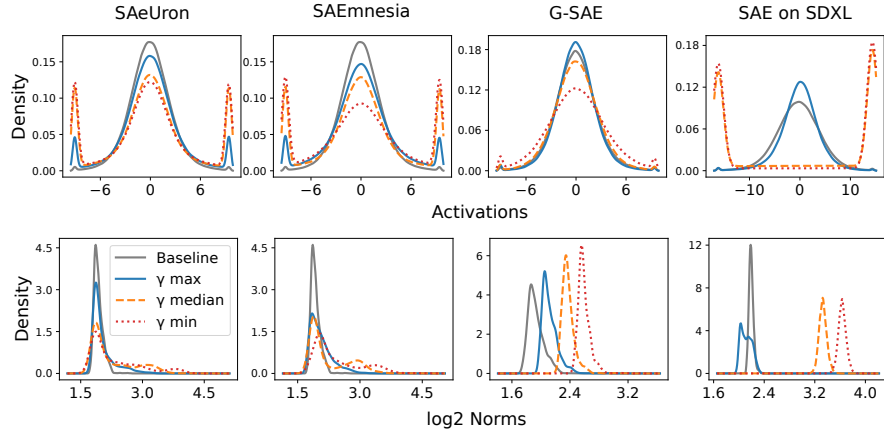


Fig. 3: Distributions at the hooked layer comparing original and SAE-steered activations across four models (SAeUron, SAEmnesia, GSAE, SDXL Turbo). Top row: Activation distributions; bottom row: L2 norm distributions (log scale). γ max, γ median and γ min are the maximum, median and minimum values, respectively, of the γ ranges of the presented methods.

bution shift is particularly extreme for SDXL Turbo where half activations lie outside the original activation distribution. These results show that multiplier-based unlearning pushes reconstructed activations outside the DM’s training distribution, especially with larger multipliers, rather than performing a targeted removal that preserves the activation manifold. In Fig. 3 (bottom), we

show the L2 norm distributions, revealing that, while original norms are concentrated around 10^2 , steered activations exhibit a heavy tail reaching 10^3 . We see that 51.3%, 44.7%, 99.5%, and 100% of activations generated with the median negative multipliers do not overlap with the norm distribution for SAeUron, SAEmnesia, G-SAE, and SDXL Turbo respectively. Both of these results support our claim that multiplier-based unlearning pushes the reconstructed activations outside the activation distribution of the DM.

Artifacts Rate. A consequence of the OOD activations from multiplier-based interventions is the presence of artifacts in the generated images. We use Qwen2-VL-7B-Instruct to measure the AR across images generated with and without unlearning. It is prompted to detect artifacts in outputs of SAE-based unlearning on UnlearnCanvas concepts (employed prompt reported in Sec. A.2 of the Appendix). We define the AR as the percentage of images that the VLM marks as containing artifacts out of the total images submitted. As shown in Fig. 4, when no unlearning is applied, the AR is 2% for SD 1.5 and 1% for SDXL Turbo, both computed over 1020 generated images. Multiplier-based methods have higher

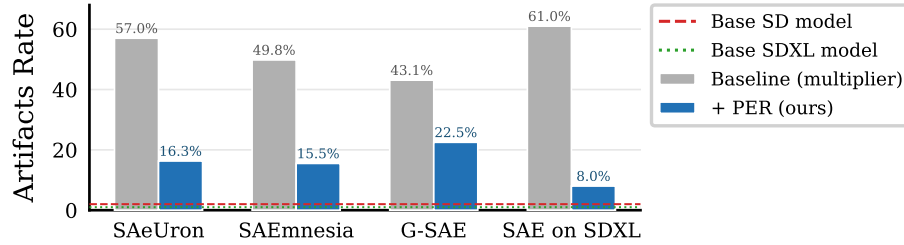


Fig. 4: AR (%) for SAeUron, SAEmnesia, G-SAE, and SAE on SDXL Turbo on UnlearnCanvas. Applying PER to different base SAEs consistently improves AR.

ARs: SAeUron produces artifacts in 57% of generated images, SAEmnesia in 49.8%, G-SAE in 43.1%, and the SAE trained on SDXL Turbo in 61%. When we apply PER, the AR is lower across all baselines: from 57% to 16.3% for SAeUron, from 49.8% to 15.5% for SAEmnesia, from 43.1% to 22.5% for G-SAE and from 61% to 8% for SAE on SDXL Turbo. These results support our hypothesis: replacing concept-containing patches with in-distribution patches mitigates the distributional mismatch (we analyze the distribution of the replacement patches in Sec. A.9 of the Appendix) and artifacts. Fig. 5 provides a per-concept breakdown for SAeUron (the same analysis performed for SAEmnesia is reported in Sec. A.8 of the Appendix). Concepts with high baseline AR improve the most: “Birds” drops from 98.0% to 3.9%, “Cats” from 88.2% to 27.5%, and “Bears” from 74.5% to 5.9%. Notably, “Dogs” and “Humans” show limited or no improvement (56.9% $\rightarrow$ 52.9% and 31.4% $\rightarrow$ 31.4%).

Human Evaluation on AR. We conducted two user studies to validate our AR results. The first compares the AR measured by Qwen2-VL-7B-Instruct

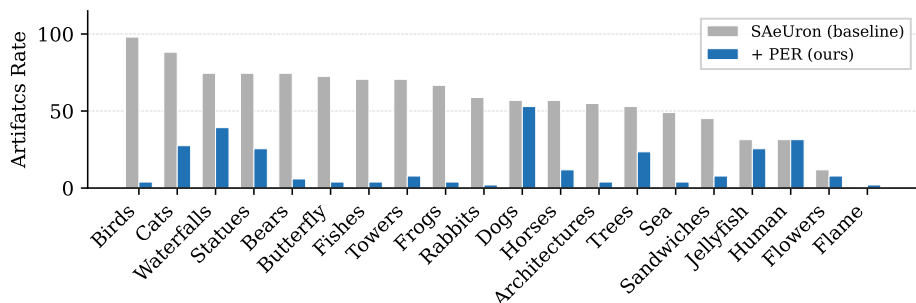


Fig. 5: Per-concept AR (%) on UnlearnCanvas for SAeUron with and without PER. Concepts are sorted by baseline AR (descending). PER reduces the overall AR from 57.0% to 16.3%, with the largest improvements on concepts where intervention strength is most disruptive (*e.g.*, “Birds”, “Cats” and “Bears”).

against humans on a 150-image subset. As reported in Tab. 1, the 18 annotators reproduce the ordering $\text{AR}(\text{SD}) < \text{AR}(+\text{PER}) \ll \text{AR}(\text{SAEmnesia})$ with Krippendorff’s $\alpha = 0.76$, denoting substantial agreement. This confirms that

	Base SD	SAEmnesia	+PER
Humans	0.0	90.7	10.0
VLM	10.0	39.5	14.0

Table 1: AR (%) ($\downarrow$ better) on a 150-image subset: human majority vote (18 annotators) vs. Qwen2-VL.

the VLM-AR in Fig. 4 is a conservative lower bound on PER’s improvements. The second study collects pairwise preferences between SAEmnesia and +PER outputs on 47 evaluation pairs among 26 retained annotators. The results are reported in Tab. 2. Out of 1183 valid responses, 927 were non-tie (exactly one of SAEmnesia or +PER was preferred) and 256 were ties. Among the non-tie responses, +PER is preferred over SAEmnesia in 885 of 927 cases, or 95.5%; this margin is highly significant ($p < 10^{-4}$), with a 95% confidence interval ranging from 93.9 to 96.6%. Ties between SAEmnesia and +PER account for 21.6% of responses, of which 231 were rated “both bad” and only 25 “both good”, indicating that tied pairs reflect joint failure rather than joint success. SAEmnesia is preferred on only 2 pairs, both in the Architectures concept.

4.3 Quantitative results

Computational efficiency. PER can be applied to any SAE-based unlearning pipeline, requiring no model fine-tuning, no additional SAE training, and no γ_c search. In contrast, existing methods require a grid search on a validation set over γ_c values per concept: SAeUron, SAEmnesia, G-SAE, and SAE trained on SDXL Turbo must evaluate 7, 7, 6, and 9 values of γ , respectively (ranges are reported in Sec. A.7). PER removes this costly search across all methods.

Table 2: Pairwise human preference between +PER and SAE_{mnesia}. We report the results for 26 retained annotators across 47 evaluation pairs (1,183 valid judgments). “% of total” is over all judgments; “% of group” is within each split. Among non-tie judgments, +PER is preferred in 95.5% of cases.

Split	Outcome	Judgments	% of total	% of group
Non-tie	+ PER preferred	885	74.8	95.5
	SAE _{mnesia} preferred	42	3.6	4.5
	<i>Subtotal</i>	927	78.4	100.0
Tie	Both bad	231	19.5	90.2
	Both good	25	2.1	9.8
	<i>Subtotal</i>	256	21.6	100.0
Total		1183	100.0	—

UnlearnCanvas. Tab. 3 presents the performance of PER on the UnlearnCanvas benchmark for object unlearning. The base metrics reported by UnlearnCanvas do not account for the visual quality of generated images when unlearning concepts. In fact, a method could achieve high unlearning scores while produc-

Table 3: Evaluation metrics (%) on object concept unlearning using the UnlearnCanvas benchmark. Comparing different SAE-based unlearning methods to PER with the same SAE. The best result for each metric is highlighted in bold.

Method	UA (↑)	IRA (↑)	CRA (↑)	Avg. (↑)	AR (↓)	GA (↑)
SAeUron	87.16	85.57	74.14	82.29	57.0	72.47
+ PER	85.37	81.14	86.55	84.35	16.3	84.19
SAE _{mnesia}	94.65	91.39	88.48	91.51	49.8	81.18
+ PER	91.37	91.92	97.97	93.45	15.5	91.44
G-SAE	78.14	96.14	95.56	89.94	43.1	81.69
+ PER	94.02	96.11	95.87	95.33	22.5	90.88
SDXL Turbo SAE	13.92	85.55	-	49.74	61.0	46.15
+ PER	14.41	86.06	-	50.24	8.0	64.16

ing images with high AR. To capture this dimension, we incorporate $(100 - \text{AR})$ in the average score computation and name it Global Average (GA). This ensures that the final aggregate metric rewards both effective unlearning *and* the ability to generate clean, high-quality images. When applied to SAeUron, PER reaches GA of 84.19%, a 11.72% improvement over the multiplier-based baseline. Notably, the best improvement is on the CRA metric, where our method scores 86.55% compared to 74.14%. When applied to SAE_{mnesia}, PER achieves 91.44% GA score, improving over the 81.18% baseline by 10.26%. We again achieve substantial gains on CRA (97.97% vs. 88.48%) while keeping IRA slightly above the baseline (91.92% vs. 91.39%). The consistent CRA improvements across both pipelines confirm that replacing concept-containing patches with in-distribution activations preserves unrelated concepts more effectively than multiplier-based

intervention. PER also significantly improves the GA when applied to G-SAE, yielding a 9.19% gain. In this case, the highest gain is on the UA metric, mainly due to the already high CRA. For SAE trained on SDXL Turbo, the CRA could not be computed fairly since, unlike the other models, SDXL Turbo was not fine-tuned on UnlearnCanvas. We still evaluated the unlearning on objects, showing a slight improvement in UA, IRA, and AR, which brings the GA from 46.15% to 64.16%. Additional results with different padding values are in Sec. A.6.

Adversarial Attacks. We follow the experimental setup of SAeUron and SAEEmnesia. By applying adversarial attacks to PER, we go beyond surface-level unlearning comparisons and provide a more complete picture of our approach’s behavior. We follow the UnlearnDiffAtk benchmark [29] on the objects of UnlearnCanvas, optimizing 5-token adversarial prefixes for 40 iterations with a learning rate of 0.01. Results are reported in Tab. 4. When applied to the SAeUron

Table 4: Adversarial robustness (UnlearnDiffAtk). Object UA (%) before and after attack, and attack effectiveness.

Pipeline	Method	Before ($\uparrow$)	After ($\uparrow$)	Eff. ($\downarrow$)
SAeUron	Baseline	83.70	34.20	49.50
	+ PER	84.60	28.30	56.30
SAEEmnesia	Baseline	97.60	57.50	40.10
	+ PER	91.10	56.20	34.90

pipeline, PER achieves comparable pre-attack UA (84.60%) but shows a larger drop after the adversarial attack, with an effectiveness value of 56.30 compared to 49.50 for the baseline. When applied to the SAEEmnesia pipeline, the trend reverses. PER achieves the lowest attack effectiveness at 34.90, improving over the SAEEmnesia baseline of 40.10. While the pre-attack UA is lower (91.10% vs. 97.60%), the post-attack UA remains competitive (56.20% vs. 57.50%), resulting in a smaller overall degradation. Additional results are reported in Sec. A.5.

Padding Analysis. We first motivate the choice of $p = 1$ through the experimental results reported in Sec. A.6. Setting $p = 0$ leads to a clear drop in unlearning effectiveness: with SAeUron, $p = 0$ yields a UA of only 49.12% compared to 85.37% at $p = 1$, a drop of over 36 pp, despite achieving higher CRA (97.95% vs. 86.55%), suggesting that an overly sparse intervention fails to suppress the target concept reliably. Padding beyond $p = 1$ does not consistently improve results: while $p = 2$ achieves a slightly higher UA under SAeUron (88.82% vs. 85.37%), it underperforms $p = 1$ on the overall average under SAEEmnesia (91.70% vs. 93.45%), and also yields a worse adversarial robustness effectiveness score (38.60% vs. 34.90%). This suggests that $p = 1$ is the most reliable choice. Fig. 6 further illustrates this trade-off by showing the fraction of patches marked for replacement at each denoising timestep for increasing padding factors. At $p = 0$, only the patches directly identified as concept-containing are replaced, resulting in a sparse intervention concentrated at later timesteps. As p increases, a larger portion of the feature map is marked.

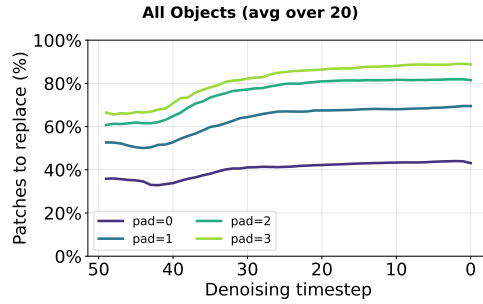


Fig. 6: Ratio of SAEEmnesia patches replaced across timesteps for padding values $p \in \{0, 1, 2, 3\}$.

4.4 Qualitative results

Fig. 7 shows the patches detected as concept containing by the SAE latents alone ($p = 0$), overlaid on the feature maps at selected timesteps. The green regions correspond to patches where the Eq. 11 is satisfied. We can see that SAE latents localize the target concept with remarkable precision. At early denoising

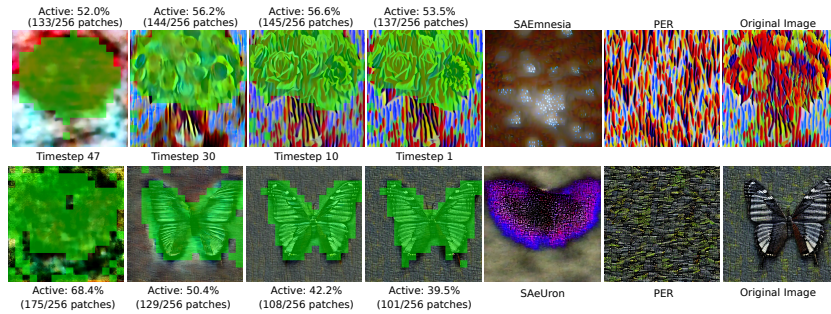


Fig. 7: Concept detection masks ($p = 0$) overlaid on intermediate timesteps for SAEUron and SAEEmnesia. As generation progresses, the mask tightens around the objects.

timesteps (high t), the feature map is still noisy and the detected patches are more scattered; as denoising progresses toward $t = 1$, the concept becomes increasingly localized and the mask sharpens around the target object. Concept information in a contiguous feature map does not have hard spatial boundaries: the influence of a concept naturally bleeds into neighboring patches, which the $p = 0$ mask leaves unmarked. Expanding the detected region by a padding factor $p > 0$ is therefore a natural choice to increase coverage. In Fig. 8, we observe that unlearning with PER produces artifact-free results, preserving original style and improving visual quality.

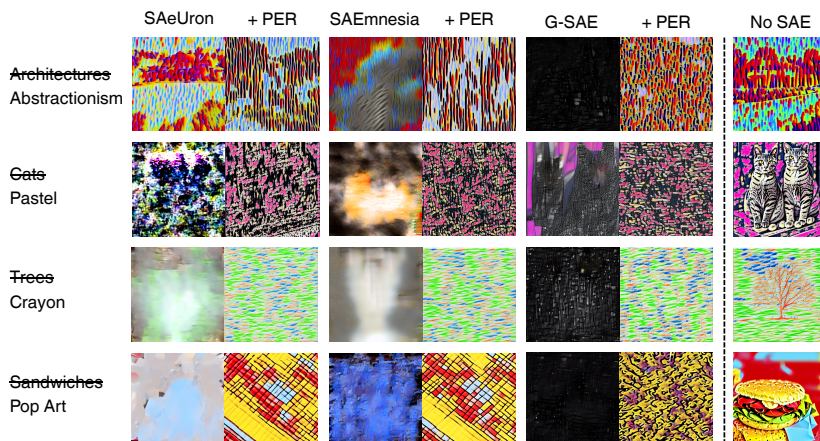


Fig. 8: Qualitative comparison of SAE-based unlearning methods with and without PER. Existing SAE-based steering methods can produce visual artifacts or corrupted images. In contrast, PER removes the object while preserving coherent style, demonstrating that detection-based patch replacement produces cleaner results.

5 Conclusion

We studied SAEs for unlearning in DMs and identified a concrete failure mode. While SAE latents can reliably detect concepts, directly steering them with multiplier-based interventions often produces OOD activations, leading to artifacts. We introduced PER, a detection-based framework that leverages SAEs only for concept localization, avoiding latent manipulation. By replacing concept-containing patch embeddings with ID activations of the same feature map, PER preserves the DM’s activation statistics while removing the target concept. Experiments across multiple SAEs show that PER consistently reduces artifacts, improves cross-domain retention, and achieves higher performance on Unlearn-Canvas, NSFW removal, and adversarial attacks without intervention strength tuning. Our work shows the importance of separating detection from intervention when designing interpretable control mechanisms for generative models.

Acknowledgements

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support. The ‘Mechanistically-Grounded Adaptive AI’ action has received funding from the European Union, via the oc4-2025-TES-02 issued and implemented by the ENFIELD project, under the grant agreement No 101120657. This work is also supported by Hi! PARIS and ANR/France 2030 program (ANR-23-IACL-0005).

References

1. Bereska, L., Gavves, S.: Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research* (2024), survey Certification, Expert Certification
2. Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J.E., Hume, T., Carter, S., Henighan, T., Olah, C.: Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread* (2023), <https://transformer-circuits.pub/2023/monosemantic-features/index.html>
3. Cassano, E., Renzulli, R., Nurisso, M., Zaffaroni, M., Perotti, A., Grangetto, M.: SAEmmesia: Erasing concepts in diffusion models with supervised sparse autoencoders. In: *Forty-third International Conference on Machine Learning* (2026)
4. Cywiński, B., Deja, K.: SAEUron: Interpretable concept unlearning in diffusion models with sparse autoencoders. In: *Forty-second International Conference on Machine Learning* (2025)
5. Fan, C., Liu, J., Zhang, Y., Wong, E., Wei, D., Liu, S.: Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In: *The Twelfth International Conference on Learning Representations* (2024)
6. Feng, X., Zhang, J., Yu, F., Wang, C., Zhang, L., Li, K., Li, Y., Chen, C., Yin, J.: A survey on generative model unlearning: Fundamentals, taxonomy, evaluation, and future direction. *arXiv preprint arXiv:2507.19894* (2025)
7. Gandikota, R., Materzynska, J., Fiotto-Kaufman, J., Bau, D.: Erasing concepts from diffusion models. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 2426–2436 (2023)
8. Gandikota, R., Orgad, H., Belinkov, Y., Materzyńska, J., Bau, D.: Unified concept editing in diffusion models. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. pp. 5111–5120 (2024)
9. Härle, R., Friedrich, F., Brack, M., Deiseroth, B., Waeldchen, S., Schramowski, P., Kersting, K.: Measuring and guiding monosemanticity. In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems* (2025)
10. Heng, A., Soh, H.: Selective amnesia: A continual learning approach to forgetting in deep generative models. *Advances in Neural Information Processing Systems* **36**, 17170–17194 (2023)
11. Kim, D., Ghadiyaram, D.: Concept steerers: Leveraging k-sparse autoencoders for controllable generations. *arXiv preprint arXiv:2501.19066* (2025)
12. Kumari, N., Zhang, B., Wang, S.Y., Shechtman, E., Zhang, R., Zhu, J.Y.: Ablating concepts in text-to-image diffusion models. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 22691–22702 (2023)
13. Li, S., van de Weijer, J., Khan, F., Hou, Q., Wang, Y., et al.: Get what you want, not what you don’t: Image content suppression for text-to-image diffusion models. In: *The Twelfth International Conference on Learning Representations* (2023)
14. Lyu, M., Yang, Y., Hong, H., Chen, H., Jin, X., He, Y., Xue, H., Han, J., Ding, G.: One-dimensional adapter to rule them all: Concepts diffusion models and erasing applications. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 7559–7568 (2024)
15. Makhzani, A., Frey, B.J.: k-sparse autoencoders. *CoRR* **abs/1312.5663** (2013)

16. Mayne, H., Yang, Y., Mahdi, A.: Can sparse autoencoders be used to decompose and interpret steering vectors? In: MINT: Foundation Model Interventions (2024)
17. O'Brien, K., Majercak, D., Fernandes, X., Edgar, R.G., Bullwinkel, B., Chen, J., Nori, H., Carignan, D., Horvitz, E., Poursabzi-Sangdeh, F.: Steering language model refusal with sparse autoencoders. In: ICML 2025 Workshop on Reliable and Responsible Foundation Models (2025)
18. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
19. Schramowski, P., Brack, M., Deiseroth, B., Kersting, K.: Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22522–22531 (2023)
20. Surkov, V., Wendler, C., Mari, A., Terekhov, M., Deschenaux, J., West, R., Gulcehre, C., Bau, D.: One-step is enough: Sparse autoencoders for text-to-image diffusion models. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
21. Tinaz, B., Fabian, Z., Soltanolkotabi, M.: Emergence and evolution of interpretable concepts in diffusion models through the lens of sparse autoencoders. In: Second Workshop on Visual Concepts (2025)
22. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., Lin, J.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution (2024)
23. Wu, J., Harandi, M.: Scissorhands: Scrub data influence via connection sensitivity in networks. In: European Conference on Computer Vision. pp. 367–384. Springer (2024)
24. Wu, J., Le, T., Hayat, M., Harandi, M.: Erasediff: Erasing data influence in diffusion models (2024)
25. Wu, Z., Arora, A., Geiger, A., Wang, Z., Huang, J., Jurafsky, D., Manning, C.D., Potts, C.: Axbench: Steering LLMs? even simple baselines outperform sparse autoencoders. In: Forty-second International Conference on Machine Learning (2025)
26. Xu, H., Zhu, T., Zhang, L., Zhou, W., Yu, P.S.: Machine unlearning: A survey. *ACM Comput. Surv.* **56**(1) (Aug 2023). <https://doi.org/10.1145/3603620>
27. Zhang, G., Wang, K., Xu, X., Wang, Z., Shi, H.: Forget-me-not: Learning to forget in text-to-image diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1755–1764 (2024)
28. Zhang, Y., Fan, C., Zhang, Y., Yao, Y., Jia, J., Liu, J., Zhang, G., Liu, G., Kompella, R.R., Liu, X., Liu, S.: Unlearncanvas: Stylized image dataset for enhanced machine unlearning evaluation in diffusion models. In: The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2024)
29. Zhang, Y., Jia, J., Chen, X., Chen, A., Zhang, Y., Liu, J., Ding, K., Liu, S.: To generate or not? safety-driven unlearned diffusion models are still easy to generate unsafe images... for now. In: European Conference on Computer Vision. pp. 385–403. Springer (2024)