Disentangled Sparse Representations for Concept-Separated Diffusion Unlearning

Hyeonjin Kim*
Yonsei University
hyeonjin@yonsei.ac.kr

Hangyeol Jung*
KAIST
han.jung@kaist.ac.kr

Heechan Yun
Yonsei University
jason0295@yonsei.ac.kr

Sungjun Yun
Yonsei University
sungjun.yun@yonsei.ac.kr

Dong-Jun Han
Yonsei University
djh@yonsei.ac.kr

Abstract

Unlearning specific concepts in text-to-image diffusion models has become increasingly important for preventing undesirable content generation. Among prior approaches, sparse autoencoder (SAE)-based methods have attracted attention due to their ability to suppress target concepts through lightweight manipulation of latent features, without modifying model parameters. However, SAEs trained primarily for sparse reconstruction do not explicitly enforce concept-wise separation, allowing semantically related concepts to share latent features. To address this, we propose **SAEParate**, which explicitly separates semantically overlapping concepts in the SAE latent space through concept-aware contrastive learning, supported by cross-attention-guided spatial aggregation that emphasizes concept-relevant regions. In addition, we enhance the encoder with a nonlinear transformation to increase its expressive capacity under this separation objective, enabling more concept-discriminative latent representations. Experiments on UnlearnCanvas demonstrate state-of-the-art performance, with particularly strong gains in joint style-object unlearning, a challenging setting where existing methods suffer from severe interference between target and non-target concepts.

1 Introduction

Diffusion-based text-to-image (T2I) models have recently achieved remarkable progress in image synthesis, producing highly realistic and diverse visual content [35, 38]. However, these models can also generate undesirable content, including violent or sexually explicit imagery [36], as well as content that raises copyright or privacy concerns. This has motivated growing interest in machine unlearning [5], which aims to remove the influence of specific data or concepts from trained models.

T2I diffusion unlearning methods can be broadly categorized into two approaches, namely modifying model parameters [12, 14, 30, 42] and suppressing the generation of target concepts without directly editing the model [10, 28]. Among the latter, SAeUron [10] employs sparse autoencoders (SAEs) [9] as auxiliary modules to extract interpretable latent features [22] and suppresses target concepts by ablating selected features. This provides a lightweight and modular alternative to parameter-modifying approaches, enabling concept removal while keeping the base diffusion model unchanged.

However, the sparse reconstruction objective of conventional SAEs is not necessarily aligned with fine-grained diffusion unlearning. Accurate sparse reconstruction does not guarantee concept-level

*Equal Contribution.
Code available at https://github.com/khyeonjin/SAEParate

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

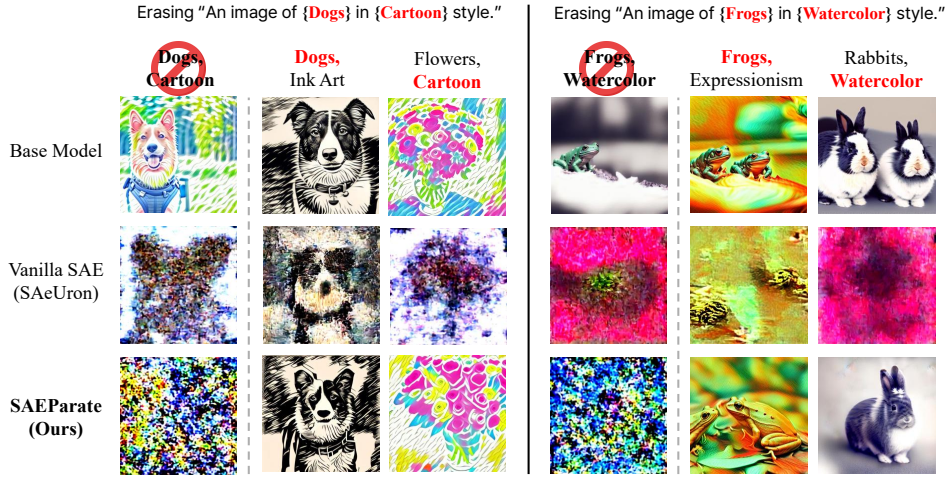


Figure 1: **Qualitative comparison of joint combination unlearning results.** Unlearning results for the target style–object combinations *Dogs in Cartoon style* (left) and *Frogs in Watercolor style* (right). First column: target combinations to be removed. Other columns: non-target combinations sharing either the same object or style. SAEParate selectively erases the target while preserving non-target concepts, while SAeUron degrades non-target outputs due to latent space interference.

separation, and semantically related concepts may therefore share latent features. Ablating target-associated features can consequently interfere with retained concepts that rely on the same features. This issue is particularly problematic in joint style–object unlearning, where concepts share partial semantics. As shown in Figure 1, removing *"Dogs in Cartoon style"* with SAeUron also degrades related non-target concepts such as *"Flowers in Cartoon style"*. This raises a key question:

Can we better separate semantically overlapping concepts in the SAE latent space so that target concepts can be removed with minimal impact on non-target ones?

In this work, we propose **SAEParate**, which explicitly learns concept-separated sparse representations for diffusion model unlearning. The key idea is to organize latent activation patterns into concept-specific clusters via a latent-space contrastive objective computed over concept-relevant regions identified by cross-attention, enabling more precise suppression of target concepts with minimal interference to non-target ones. To further improve this separation, we enhance the encoder with a nonlinear transformation, increasing its expressive capacity to learn more discriminative and disentangled features. As shown in Figure 1, SAEParate effectively erases the target style–object combination *"Dogs in Cartoon style"*, with minimal impact on non-target concepts that share partial attributes (e.g., *"Dogs in Ink Art style"* or *"Flowers in Cartoon style"*). This leads to more reliable concept removal in challenging settings where existing approaches suffer from significant interference, highlighting the importance of concept separation.

We evaluate our method on UnlearnCanvas [44], a standard benchmark for diffusion model unlearning. SAEParate outperforms existing baselines on both single-attribute unlearning (style or object) and joint style-object unlearning while preserving the generation quality of non-target concepts. Our main contributions are summarized as follows:

1. We identify concept-level sharing of SAE latent features as a key cause of unintended interference in SAE-based diffusion unlearning, and characterize this phenomenon through overlapping activation patterns in the latent space.
2. We propose SAEParate, a unified concept-separation framework that explicitly separates semantically overlapping concepts in the SAE latent space, enabling selective feature intervention with reduced interference to retained concepts.
3. We demonstrate that SAEParate consistently improves unlearning across benchmark, unseen-concept, and safety-critical settings, with the largest benefits arising when target and retained concepts are semantically similar and highly entangled in the latent space.

Following prior work [7, 10], we use the term "unlearning" to refer to the suppression of target concepts in model outputs. SAEParate achieves this suppression through SAE-based steering, without claiming permanent removal of the underlying knowledge.

2 Related Work

Concept Unlearning in Diffusion Models. Machine unlearning [5] aims to remove specific data or knowledge from a trained model. In text-to-image diffusion models, this is commonly formulated as concept erasure while preserving unrelated generation capabilities [44]. Existing approaches intervene at different levels: ESD [13] and EDiff [42] modify model behavior through optimization-based objectives; SalUn [12], SHS [41], and FMN [43] restrict updates to selected parameters or layers; and CA [26] and SA [19] reformulate the target distribution for selective forgetting. Parameter-efficient or training-free alternatives include SPM [30], UCE [14], and SEOT [28].

SAE-based approaches take a different route by intervening on intermediate activations through an auxiliary sparse representation rather than encoding the forgetting behavior into the diffusion model parameters. This modular formulation leaves the pretrained generator itself untouched and allows the intervention to be removed when no longer needed. Moreover, sparsity provides a direct view of the latent components being modified during unlearning. SAeUron [10] suppresses target-associated SAE features. SAEmnesia [7] instead assigns each labeled concept to a designated latent, enabling explicit concept localization, but this one-to-one mapping can be restrictive when fine-grained or compositional semantics are distributed across multiple features. PER [6] addresses an orthogonal intervention-level issue by replacing target patches to mitigate out-of-distribution activations induced by steering. In contrast, SAEParate retains distributed feature usage and explicitly organizes sparse activation patterns according to concept identity, promoting concept-level separation across the full latent space. This design is particularly suited to fine-grained settings where target and retained concepts share substantial semantic attributes, such as closely related objects or style–object combinations.

Sparse Autoencoders (SAEs). SAEs are feature learning methods that expand input representations into a higher-dimensional latent space, trained to reconstruct inputs using only a sparse subset of activated features [9]. They have become a widely used tool for mechanistic interpretability, decomposing dense representations into more interpretable feature dictionaries [2, 22]. Sparsity can be enforced through different mechanisms: ReLU-based SAEs apply a threshold but are prone to dead neurons, TopK SAEs [31] fix the number of active features per forward pass, and BatchTopK SAEs [3] relax this constraint at the batch level for more flexible per-sample activation. Beyond standard sparsification, recent SAE variants explore alternative objectives for improving feature organization, including Matryoshka SAE [4] and OrtSAE [25]. Complementary to these approaches, SAEParate introduces a concept-level training signal that structures the latent space by concept identity.

Contrastive Learning. The core idea of contrastive learning is to shape the representation space based on relative similarity, drawing semantically related samples closer while pushing distinct samples apart. This objective has proven effective in self-supervised visual representation learning [8, 17] and vision–language alignment [34]. With label supervision, supervised contrastive learning [23] treats samples from the same class as positives, yielding more compact intra-class representations and clearer inter-class separation. Recent work has also extended contrastive objectives beyond instance-level representation learning. For example, T-SAE [1] leverages temporal relationships between neighboring language tokens to encourage semantically consistent latent activations. Our work introduces a contrastive objective in the SAE latent space to explicitly enforce concept-level separation, extending contrastive learning to structured concept separation for diffusion unlearning.

3 Problem Background

3.1 Problem Setup

We study concept unlearning for a pre-trained text-to-image diffusion model, where the goal is to suppress a specified target concept while preserving the model's ability to generate retained concepts. We evaluate this objective across style, object, and joint style–object unlearning, with the joint setting targeting a specific style–object combination [44]. For instance, given the joint target *"Dogs in Cartoon style"*, the model should suppress this specific combination while still generating related non-target concepts such as *"Flowers in Cartoon style"*. We focus on SAE-based unlearning, in which an auxiliary SAE is trained on intermediate diffusion activations and target-associated sparse features are subsequently identified and suppressed at inference time. This setting requires the learned

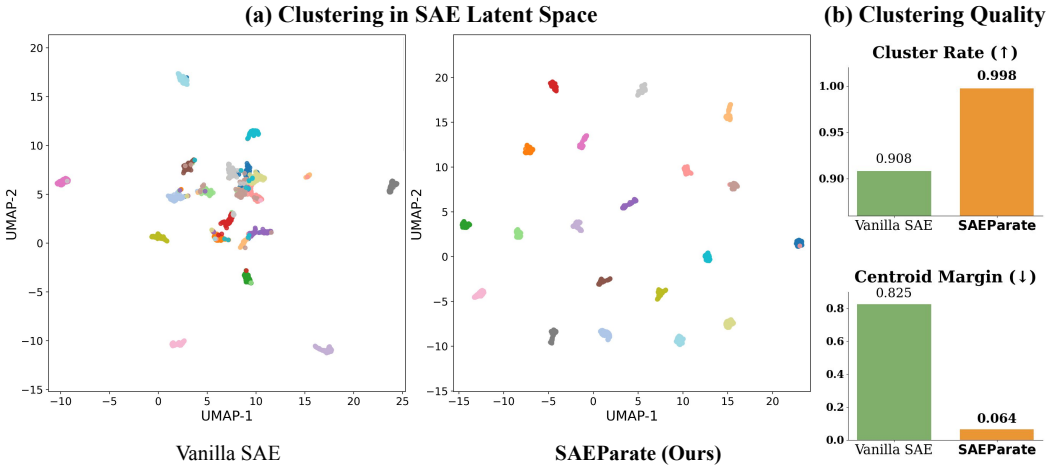


Figure 2: Visualization of SAE latent representations. (a) UMAP [33] projections of Vanilla SAE (SAeUron) and SAEParate on UnlearnCanvas [44], with each color representing a different concept. (b) Quantitative comparison of cluster rate and centroid margin, which measure centroid-based class assignment and inter-centroid similarity, respectively. Both results demonstrate that SAEParate produces clearer concept-wise clusters with reduced overlap, indicating improved concept separation.

sparse representation to support fine-grained interventions even when target and retained concepts share semantic attributes.

3.2 Preliminaries: SAE-based Diffusion Unlearning

SAeUron [10] instantiates SAE-based unlearning by training an SAE to reconstruct intermediate diffusion activations. Given an input activation $x \in \mathbb{R}^d$, a ReLU SAE with TopK [2, 15, 31] encodes it into a latent space as

$$z = \mathrm{ReLU}(W_{\mathrm{enc}}(x - b_{\mathrm{pre}}) + b_{\mathrm{enc}}), \tag{1}$$

where $W_{\mathrm{enc}} \in \mathbb{R}^{n \times d}$ and $W_{\mathrm{dec}} \in \mathbb{R}^{d \times n}$ denote the encoder and decoder weight matrices, respectively. From z, we then obtain the sparsified latent

$$\hat{z} = \mathrm{TopK}(z, k) \tag{2}$$

by retaining only the top-k activated features, and reconstruct the input as $\hat{x} = W_{\mathrm{dec}}\hat{z} + b_{\mathrm{pre}}$. Here, $b_{\mathrm{enc}} \in \mathbb{R}^n$ and $b_{\mathrm{pre}} \in \mathbb{R}^d$ are learnable bias terms, where b_{pre} centers the input before encoding and is added back after decoding.

The SAE is trained by minimizing the following objective:

$$\mathcal{L}_{\mathrm{SAE}}(x) = \|x - \hat{x}\|_2^2 + \alpha \mathcal{L}_{\mathrm{aux}}, \tag{3}$$

where the first term is the reconstruction loss, $\mathcal{L}_{\mathrm{aux}}$ is an auxiliary term that mitigates inactive latent units, and α controls its contribution.

Given a trained SAE, a set of concept-relevant SAE features, $\mathcal{F}_c$, is identified for each target concept c based on feature-importance scores after training. During inference, activations from the corresponding diffusion layer are encoded into the SAE latent space, where the selected features in $\mathcal{F}_c$ are suppressed with negative multipliers γ_c. The modified features are then reconstructed through the SAE decoder. Further details on SAE-based diffusion unlearning are provided in Section 4.3.

3.3 Motivation: Latent Entanglement in SAE-based Unlearning

However, SAE-based unlearning inherits a key limitation of SAE training, whose objective primarily encourages sparse reconstruction but does not explicitly enforce concept-wise separation. As a result, recent studies suggest that sparse autoencoders often learn shared or compositional features, where correlated attributes are encoded by shared latents to preserve reconstruction quality with fewer active features [4, 27]. Figure 2(a) illustrates this behavior in our setting, where representations of different concepts substantially overlap in the latent space. Suppressing concept-relevant features $\mathcal{F}_c$ can therefore also affect retained concepts that rely on overlapping features, degrading the unlearning–retention trade-off (Section 5). This motivates explicitly structuring SAE representations according to concept identity.

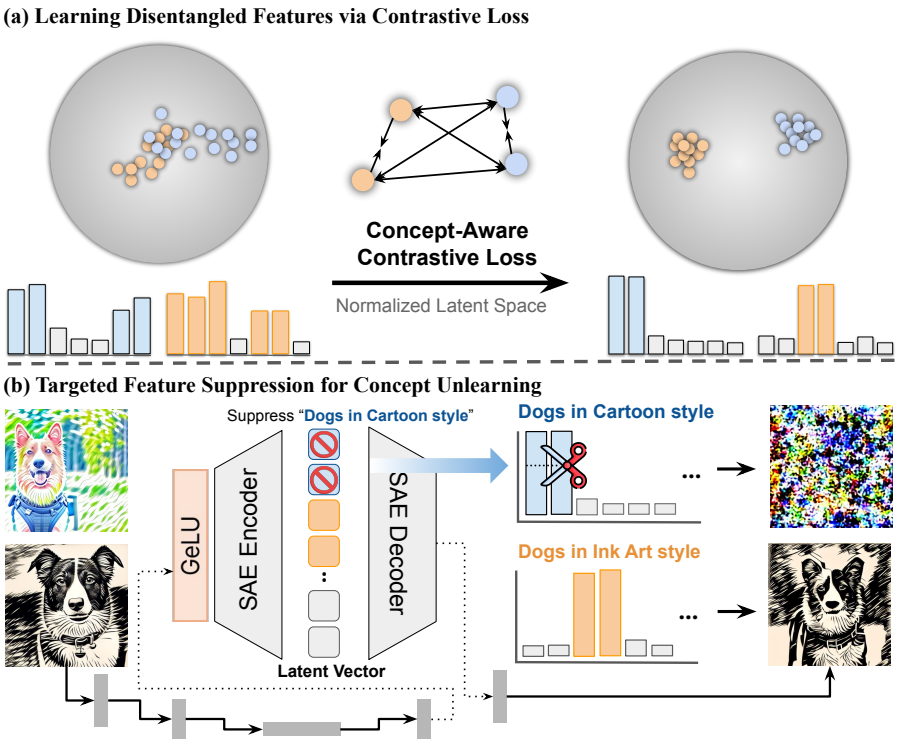


Figure 3: **Overview of SAEParate.** We perform concept unlearning in text-to-image diffusion models by suppressing concept-specific features in a Sparse Autoencoder (SAE). **(a) Training.** A concept-aware contrastive loss is applied to normalized SAE latents, pulling together representations of samples with the same concept label and pushing apart those with different labels, yielding well-separated concept-wise activation patterns. **(b) Inference-time unlearning.** Given a target concept (e.g., *Dogs in Cartoon style*), we identify and suppress the corresponding SAE features in the latent vector before decoding, erasing the target combination while leaving non-target concepts (e.g., *Dogs in Ink Art style*) intact.

4 Method

We propose **SAEParate**, which trains the SAE so that concept identity is explicitly encoded in its latent space, allowing semantically overlapping concepts to be separated at the representation level. Section 4.1 introduces concept-aware spatial aggregation, which focuses supervision on concept-relevant positions, and a latent-space contrastive objective that structures the aggregated representations by concept. Section 4.2 enhances the SAE encoder with an additional nonlinear transformation, providing the expressivity needed to reorganize entangled activations under this objective. Section 4.3 describes how the resulting representation is used for inference-time unlearning. An overview of SAEParate is shown in Figure 3.

4.1 Concept-Discriminative Sparse Representation Learning

We introduce a *concept-aware contrastive objective in the SAE latent space*, designed to explicitly separate concept-level activation patterns. While inspired by supervised contrastive (SupCon) learning [23], our formulation departs from the standard setting by operating on concept-conditioned representations aggregated from SAE latent activations and using diffusion timesteps to construct multiple views.

Multiviewed Batch. Let $x(t)^c \in \mathbb{R}^{S\times d}$ denote the diffusion activation of concept c at timestep t, where $S = h \times w$ is the number of spatial positions in the feature map and d is the dimensionality of each spatial activation. Unlike the standard SupCon framework, which constructs positive pairs using random augmentations of the same sample, our setting naturally provides multiple views through the temporal structure of the diffusion process. Specifically, for each concept c, we sort activations by timestep and construct adjacent-timestep pairs $\{(x(t_0)^c, x(t_1)^c), (x(t_2)^c, x(t_3)^c), \dots\}$, where each pair is treated as a positive pair because both activations share the same concept label and arise

from neighboring denoising steps. Following this construction, each activation in a training batch is accompanied by its adjacent-timestep counterpart, resulting in a multiviewed batch [23]. As a result, a batch of N samples forms M positive pairs with $N = 2M$. This adjacent-timestep construction ensures that each sample has at least one same-concept positive view, enabling stable and effective concept-aware contrastive learning.

Concept-Aware Spatial Aggregation. Given the resulting multi-view batch, each diffusion activation is denoted by $x_i \in \mathbb{R}^{S\times d}$ for $i \in \mathcal{I} \equiv \{1, \ldots, N\}$. For each spatial position $p \in \{1, \ldots, S\}$, we map $x_{i,p}$ into an n-dimensional latent representation using the SAE encoder f_θ:

$$z_{i,p} = f_\theta(x_{i,p}), \tag{4}$$

where f_θ denotes the nonlinearity-enhanced encoder introduced in Section 4.2, and $z_{i,p}$ is taken before the TopK sparsification. To obtain a sample-level representation, we aggregate the spatial latents $\{z_{i,p}\}_{p=1}^{S}$. Since many spatial positions correspond to concept-irrelevant regions (e.g., background or unrelated objects), uniformly pooling all positions would introduce noise. We therefore weight each position by its cross-attention relevance to the concept:

$$\alpha_{i,p} = \frac{\sum_{k\in\mathcal{C}_i}\left(\frac{1}{H}\sum_{h=1}^{H} A_{i,h,p,k}\right)}{\sum_{p'=1}^{S}\sum_{k\in\mathcal{C}_i}\left(\frac{1}{H}\sum_{h=1}^{H} A_{i,h,p',k}\right)}, \tag{5}$$

where $A_{i,h,p,k}$ denotes the cross-attention probability from spatial position p to text token k at head h, H is the number of attention heads, and $\mathcal{C}_i$ is the set of prompt tokens corresponding to the concept of sample i. The weights satisfy $\sum_{p=1}^{S} \alpha_{i,p} = 1$, and the sample-level representation is

$$r_i = \sum_{p=1}^{S} \alpha_{i,p} z_{i,p}. \tag{6}$$

Concept-wise visualizations of Eq. 5 and an ablation on cross-attention weighted pooling are provided in Appendix A.11.

Latent-Space Contrastive Objective. Let y_i denote the label of sample x_i, and define the positive set for anchor i as $\mathcal{P}(i) = \{j \neq i \mid y_j = y_i\}$. For example, in joint style–object unlearning, two activations generated from the same style–object concept (e.g., *Dogs in Cartoon style*) form a positive pair, whereas activations from different concepts (e.g., *Dogs in Ink Art style*, *Flowers in Cartoon style*) form negative pairs, even when they share partial attributes. We ℓ_2-normalize each sample-level representation as $u_i = r_i/\|r_i\|_2$, so that $u_i^\top u_j$ is the cosine similarity between two samples. The latent-space contrastive loss $\mathcal{L}_{\mathrm{LSC}}$ is then defined as

$$\mathcal{L}_{\mathrm{LSC}} = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\frac{1}{|\mathcal{P}(i)|}\sum_{j\in\mathcal{P}(i)}\log\frac{\exp(u_i^\top u_j/\tau)}{\sum_{k\neq i}\exp(u_i^\top u_k/\tau)}, \tag{7}$$

where $\tau > 0$ denotes the temperature parameter. This objective encourages activation patterns from the same concept to form compact clusters while remaining well separated from those of other concepts. We make minor extensions to Eq. 7 for joint style–object unlearning in Section 5.1.

Training Objective. The final training objective combines the SAE objective from Eq. 3 with the proposed latent-space contrastive regularization of Eq. 7, where λ controls the contribution of the contrastive term:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{SAE}} + \lambda\mathcal{L}_{\mathrm{LSC}}. \tag{8}$$

4.2 Nonlinearity-Enhanced SAE Encoder

The latent representations of the SAE encoder in Eq. 1 are obtained by applying a ReLU to a single affine transformation. Although this formulation is simple and effective for reconstruction, its representational capacity remains limited by its shallow structure, making it insufficient to fully accommodate the discriminative pressure imposed by the latent-space contrastive loss.

To increase the expressive power of the encoder, we enhance it with an additional hidden layer followed by a nonlinear activation σ, introducing a deeper nonlinear transformation before sparse feature selection. The modified version of Eq. 1 becomes

$$z = \mathrm{ReLU}(W_{\mathrm{enc2}}\,\sigma(W_{\mathrm{enc1}}(x - b_{\mathrm{pre}}) + b_{\mathrm{enc1}}) + b_{\mathrm{enc2}})\,, \tag{9}$$

where $W_{\text{enc1}} \in \mathbb{R}^{d\times d}$ and $W_{\text{enc2}} \in \mathbb{R}^{n\times d}$ denote the weights of the additional hidden layer and the final encoder projection, and $b_{\text{enc1}} \in \mathbb{R}^d$ and $b_{\text{enc2}} \in \mathbb{R}^n$ are the corresponding biases. We then obtain the sparsified latent $\hat{z}$ by applying the TopK operation in Eq. 2, and use this encoder consistently during both training and inference. The decoder remains linear, adding expressivity only to the encoder while preserving sparsification and inference-time ablation.

For σ, we adopt GeLU [18], defined as $\mathrm{GeLU}(x) = x\Phi(x)$ with Φ the standard Gaussian CDF. Appendix A.4 compares linear, ReLU, and GeLU hidden layers, showing that GeLU further improves target-concept removal while maintaining comparable retention.

4.3 Unlearning Procedure

After completing the training procedure of SAEParate, we perform concept unlearning via an inference-time intervention in the concept-separated latent space learned by the SAE. For each target concept c, we compute feature importance scores using feature activations from target and non-target samples, and select the top-τ_c concept-relevant features as $\mathcal{F}_c$. For example, for the concept *Dogs in Cartoon style*, this step identifies features that are strongly activated by this concept compared to non-target concepts. During inference, we encode the diffusion activation from the layer on which the SAE was trained into the SAEParate latent space, and suppress the selected features in $\mathcal{F}_c$ by multiplying their activations with a negative multiplier γ_c. The main cost of SAEParate therefore comes from offline SAE training and feature selection, while unlearning itself only requires an inference-time intervention. Further details of the unlearning procedure are provided in Appendix A.7.

5 Experiments

We evaluate our method under the 50-style, 20-object evaluation setting of UnlearnCanvas [44], using its fine-tuned Vision Transformer-based [11] classifiers for evaluation. The benchmark also provides a Stable Diffusion v1.5 [38] checkpoint fine-tuned on UnlearnCanvas, which we use as the basis for unlearning evaluation.

Baselines. We compare SAEParate against nine existing T2I diffusion unlearning methods from the UnlearnCanvas benchmark [44]: ESD [13], FMN [43], UCE [14], CA [26], SalUn [12], SEOT [28], SPM [30], EDiff [42], and SHS [41]. We additionally include two SAE-based methods closely related to ours, SAeUron [10] and SAEmnesia [7].

Performance Metrics. Following the UnlearnCanvas [44] protocol, we report Unlearning Accuracy (**UA**), In-domain Retain Accuracy (**IRA**), Cross-domain Retain Accuracy (**CRA**), and FID [21] for single-attribute unlearning. For joint style–object unlearning, in addition to UA, we report Style Consistency (**SC**), Object Consistency (**OC**), and Unrelated Prompting (**UP**), where UP measures generation quality on non-target (style, object) combinations.

Implementation Details. For fair comparison with SAeUron [10], we keep the SAE application location, training data construction, and feature-calibration pipeline unchanged, and modify only the SAE training objective and encoder architecture. Following SAeUron, we apply the SAE to `up.1.2` for style unlearning and `up.1.1` for object unlearning. For joint unlearning, we uniformly use `up.1.1` for all SAEs. We also use the same anchor-prompt-based training and calibration setup, while keeping the training prompts disjoint from the UnlearnCanvas evaluation prompts. Specifically, we use 80 one-sentence anchor prompts for each of the 20 object concepts, appending *"in {style} style."* so that the SAE is exposed to benchmark styles during training. For calibration, we use 20 prompts per style for style unlearning, 80 prompts per object for object unlearning, and 80 prompts per (object, style) combination for joint unlearning. The contrastive objective for joint unlearning and the feature scoring/suppression pipeline are detailed in Appendices A.8 and A.7, respectively.

5.1 Main Experimental Results

Single-Attribute Concept Unlearning. We first perform single-attribute concept unlearning by removing all samples from either a target class or a target style (i.e., one attribute at a time). We use two hyperparameters for each target concept c: τ_c, the number of selected features in the ablation set $\mathcal{F}_c$, and γ_c, a negative multiplier controlling the suppression strength of each selected feature. Following the SAeUron setup, we fix $\tau_c = 1$ and $\gamma_c = -0.1$ for all style concepts, while tuning them on a validation set for object unlearning. The selected values are reported in Appendix A.9.

As shown in Table 1, compared with the baseline results reported in the UnlearnCanvas benchmark, SAEParate achieves the best average performance and outperforms prior methods on most metrics

Table 1: Comparison of diffusion unlearning methods on style, object, and joint unlearning benchmarks. The best and second-best results in each column are highlighted in green and underlined, respectively. Notably, SAEParate demonstrates strong unlearning performance across all unlearning settings, with particularly pronounced gains on object and joint unlearning.

Method	Style Unlearning			Object Unlearning			Avg. (↑)	FID (↓)	Joint Unlearning					**Total Avg.** (↑)
	UA (↑)	IRA (↑)	CRA (↑)	UA (↑)	IRA (↑)	CRA (↑)			UA (↑)	SC (↑)	OC (↑)	UP (↑)	Avg. (↑)	
ESD	98.58%	80.97%	93.96%	92.15%	55.78%	44.23%	77.61%	65.55	91.42%	4.88%	14.72%	84.38%	48.85%	66.11%
FMN	88.48%	56.77%	46.60%	45.64%	90.63%	73.46%	66.93%	131.37	45.37%	68.73%	62.74%	83.25%	65.02%	66.17%
UCE	98.40%	60.22%	47.71%	94.31%	39.35%	34.67%	62.44%	182.01	75.97%	4.53%	5.72%	35.42%	30.41%	49.63%
CA	60.82%	96.01%	92.70%	46.67%	90.11%	81.97%	78.05%	**54.21**	47.92%	10.08%	56.35%	81.54%	48.97%	66.42%
SalUn	86.26%	90.39%	95.08%	86.91%	**96.35%**	**99.59%**	92.43%	61.05	42.21%	62.45%	70.93%	87.28%	65.72%	81.75%
SEOT	56.90%	94.68%	84.31%	23.25%	95.57%	82.71%	72.90%	62.38	29.32%	45.31%	53.64%	85.45%	53.43%	65.11%
SPM	60.94%	92.39%	84.33%	71.25%	90.79%	81.65%	80.23%	59.79	45.72%	41.34%	36.32%	67.82%	47.80%	67.26%
EDiff	92.42%	73.91%	98.93%	86.67%	94.03%	48.48%	82.41%	81.42	71.33%	35.23%	26.32%	51.52%	46.10%	67.89%
SHS	95.84%	80.42%	43.27%	80.73%	81.15%	67.99%	74.90%	119.34	55.32%	14.34%	24.32%	83.95%	44.48%	62.73%
SAeUron	95.80%	99.10%	**99.40%**	78.82%	95.47%	95.58%	94.03%	62.15	63.00%	16.98%	75.84%	71.90%	56.93%	79.19%
SAEmnesia†	96.60%	98.67%	99.30%	94.65%	91.39%	88.48%	94.85%	56.15	81.00%	59.58%	11.18%	54.60%	51.59%	77.55%
SAEParate	**99.60%**	**99.24%**	98.30%	**95.20%**	95.91%	94.26%	**97.09%**	63.98	**94.11%**	**93.49%**	**91.66%**	**98.82%**	**94.52%**	**96.06%**

† SAEmnesia does not originally consider joint style–object unlearning; we adapt it to this setting as detailed in Appendix A.1.

for both style and object unlearning. Notably, it preserves non-target concepts well, as reflected in IRA and CRA, while improving UA. The competitive FID further indicates that retain-set image quality is preserved at the distributional level. These results suggest that SAEParate learns a concept-discriminative latent space that mitigates redundant feature selection and enables more precise suppression of target-relevant features. SAEParate also improves efficiency over other baselines while remaining competitive with SAeUron, as shown in Table 2. Additional details are in Appendix A.6.

Joint Style-Object Concept Unlearning. We further evaluate on the joint style-object unlearning setting from UnlearnCanvas, where the target is a specific style-object pair (e.g., *Dogs in Cartoon style*), while partially matched prompts (e.g., the same object in different styles or different objects in the same style) must be preserved. This requires finer-grained selectivity than single-attribute removal, providing a stricter test of whether the learned SAE representations are sufficiently concept-specific. For this setting, we retain the SAEParate framework and adapt Eq. 7 to the style-object level: positives are defined at the style-object combination level, while partially matched samples are treated as hard negatives which get more separation pressure. Further details are provided in Appendix A.8.

As shown in Table 1, SAEParate achieves the best performance across all four joint unlearning metrics, improving the average score from 65.72% to 94.52% over the strongest baseline. SAEParate achieves effective UA while maintaining OC and SC, demonstrating that our method performs surgical rather than destructive intervention.

Table 2: Efficiency comparison. Memory is measured with a batch size of 4096 following SAeUron. Storage denotes additional method-specific files.

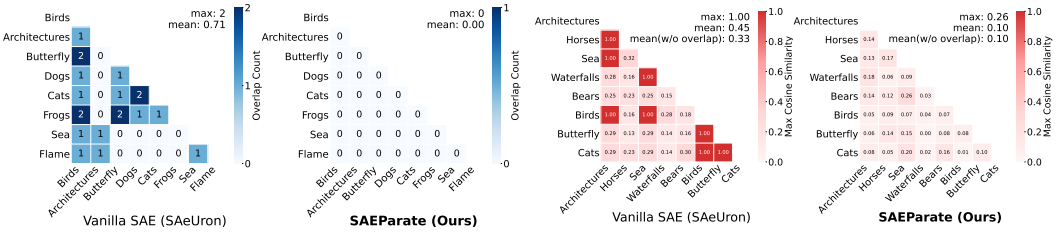
Method	Memory (GB)↓	Storage (GB)↓
ESD	17.8	4.3
FMN	17.9	4.2
UCE	5.1	1.7
CA	10.1	4.2
SalUn	30.8	4.0
SEOT	7.34	0.0
SPM	6.9	0.0
EDiff	27.8	4.0
SHS	31.2	4.0
SAeUron	2.8	0.2
SAEParate	3.4	0.2

Figure 1 further provides qualitative comparisons with SAeUron, showing cleaner removal of the target combination while better preserving partially matched prompts. These results suggest that the effectiveness of SAEParate extends beyond single-concept removal to finer-grained unlearning settings that require more selective and concept-specific representations. We provide more qualitative results in Appendix C.3.

5.2 Further Analyses

Overlap Analysis. To directly assess whether the proposed method mitigates redundant feature selection, we perform a pairwise analysis of class-specific selected feature groups. We measure both the number of overlapping selected features and the maximum cosine similarity between feature groups, capturing overlap in feature identity as well as similarity in their decoder directions. SAEParate reduces feature overlap to zero and substantially lowers pairwise similarity (Figure 4), indicating more distinct intervention features and a reduced risk of affecting retained concepts during target suppression. We provide the full class-wise overlap analysis, including the more challenging joint style–object setting, in Appendix B.3.

Concept-wise k-sparse Probing. To assess how compactly concept-discriminative information is encoded in the SAE latent space, we perform concept-wise k-sparse probing [16]. For each concept, we select the top-k latents according to the feature-importance score used for unlearning, and train a



(a) Pairwise overlap counts between selected class-specific feature sets.

(b) Pairwise maximum cosine similarity between selected class-specific feature groups.

Figure 4: Class-wise overlap analysis comparing Vanilla SAE (left) and SAEParate (right).

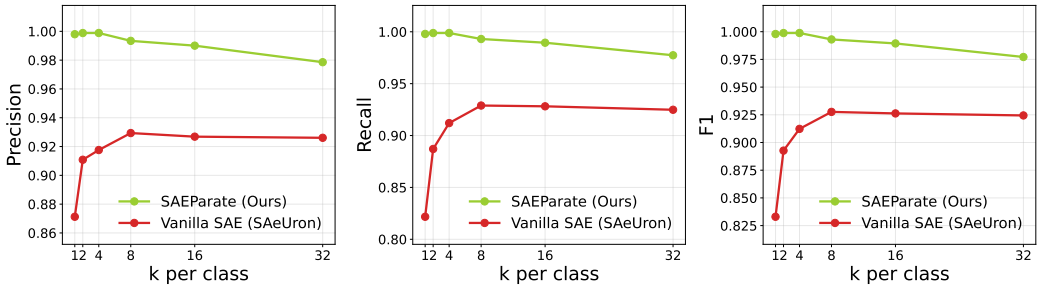


Figure 5: Concept-wise k-sparse probing comparison on UnlearnCanvas object concepts.

linear probe with cross-entropy loss to predict concept labels using only the union of these selected latents.

As shown in Figure 5, SAEParate achieves strong classification scores even with $k = 1$ or $k = 2$, whereas SAeUron that uses Vanilla SAE requires larger k and still underperforms. These results indicate that SAEParate concentrates concept-discriminative information into a small number of latent features, reducing redundancy and improving controllability.

Table 3: Quantitative ablation results on object unlearning.

Variants	Unlearning Performance			Avg. (↑)	Cluster Quality	
	UA (↑)	IRA (↑)	CRA (↑)		Cluster Rate (↑)	Centroid Margin (↓)
Vanilla SAE	58.14%	96.74%	97.45%	84.11%	0.93	0.73
only-LSC	83.82%	95.23%	91.55%	90.20%	0.98	0.71
only-GeLU	47.55%	96.69%	97.53%	80.59%	0.94	0.64
SAEParate	95.59%	96.48%	93.46%	95.18%	1.00	0.06

Ablation Study. To analyze the contributions of the contrastive objective and nonlinear encoder, we conduct an ablation study on contrastive learning (Section 4.1) and the nonlinearity-enhanced encoder (Section 4.2) under the subset setting (Appendix A.2). We evaluate two ablated variants, *only*-LSC and *only*-GeLU, together with the full SAEParate model. The quantitative results in Table 3 show that the full model achieves the best overall performance on object unlearning. Moreover, the cluster quality indicates that the clusters remain only partially separated when using either *only*-LSC or *only*-GeLU, whereas combining both components yields substantially clearer concept-wise clusters.

Nudity Unlearning on I2P Benchmark. We further evaluate SAEParate on safety-critical nudity unlearning using the I2P benchmark [39], targeting *naked woman* and *naked man* as distinct concepts. SAEParate achieves strong gender-specific unlearning accuracy while retaining the opposite-gender detections, demonstrating its ability to separate fine-grained concepts. Full results are in Appendix B.5.

Generalization to Unseen Concepts and Sequential Unlearning. SAEParate also generalizes beyond the standard setup. When the SAE is trained on only half of the styles, it improves unlearning on unseen styles and reduces cross-domain interference. In a sequential setting where five objects are removed consecutively, it avoids the retention collapse observed in SAeUron. Full results are provided in Appendices B.7 and B.8.

Other Results. Additional appendix experiments demonstrate robustness to concept-restoration attacks and prompt variations (Appendix B.4), compatibility with Matryoshka SAEs (Appendix B.9), and additional qualitative generation results (Appendix C.3). SAEParate also generalizes beyond the standard benchmark setting, enabling selective *Golden Retriever* removal under coarse binary forget/retain supervision (Appendix B.6) and transferring to the MM-DiT-based SD3.5 Medium

Turbo under a preliminary adaptation, where it outperforms SAeUron in UA and overall average (Appendix B.10).

6 Conclusion

We proposed **SAEParate**, which learns concept-discriminative representations for diffusion unlearning through concept-aware spatial aggregation, contrastive representation structuring, and a nonlinearity-enhanced encoder. SAEParate achieves state-of-the-art performance on UnlearnCanvas, with particularly strong gains in fine-grained joint unlearning, and demonstrates effective gender-distinct nudity unlearning on I2P. Further analyses show substantially reduced overlap among concept-relevant latent features, suggesting that concept-discriminative sparse representations provide an effective direction for reliable diffusion unlearning.

Limitations and Future Directions. Our method requires at least group-level supervision, inherits SAE-wide issues such as feature splitting and absorption that may incompletely capture target concepts, and is currently evaluated only on image generation. Our compatibility experiments with Matryoshka SAEs (Appendix B.9) suggest that advances in SAE design could help alleviate limitations arising from feature splitting and absorption. Exploring such combinations, unsupervised training, and extension to video generation are promising directions for future work.

Acknowledgments

This work was supported in part by the "Advanced GPU Utilization Support Program" funded by the Government of the Republic of Korea (Ministry of Science and ICT), the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) in 2026 (Grant No. RS-2026-25612350, Project Title: Development of an End-to-End Design and Validation Framework for Mathematically Principled Next-Generation AI Architectures and Learning Algorithms), the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2026-RS-2026-25544188, Leading Generative AI Human Resources Development), the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (RS-2026-25470983). Dong-Jun Han is the corresponding author.

References

[1] Usha Bhalla, Alex Oesterling, Claudio Mayrink Verdun, Hima Lakkaraju, and Flavio Calmon. Temporal sparse autoencoders: Leveraging the sequential nature of language for interpretability. In *International Conference on Learning Representations*, volume 2026, pages 141147–141175, 2026.

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

[3] Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. URL `https://openreview.net/forum?id=d4dpOCqybL`.

[4] Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=m25T5rAy43`.

[5] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy*, SP '15, page 463–480, USA, 2015. IEEE Computer Society. ISBN 9781467369497. doi: 10.1109/SP.2015.35. URL `https://doi.org/10.1109/SP.2015.35`.

[6] Enrico Cassano, Riccardo Renzulli, Rayyan Ahmed, Marco Grangetto, and Stephan Alaniz. Look but don't touch with sparse autoencoders for unlearning in diffusion models. *arXiv preprint arXiv:2606.31699*, 2026.

[7] Enrico Cassano, Riccardo Renzulli, Marco Nurisso, Mirko Zaffaroni, Alan Perotti, and Marco Grangetto. SAEmnesia: Erasing concepts in diffusion models with supervised sparse autoencoders. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Pa6EoOViOq.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

[9] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/coates11a.html.

[10] Bartosz Cywiński and Kamil Deja. Saeuron: Interpretable concept unlearning in diffusion models with sparse autoencoders. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=6N0GxaKdX9.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[12] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gn0mIhQGNM.

[13] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2426–2436, 2023.

[14] Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5111–5120, 2024.

[15] Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tcsZt9ZNKD.

[16] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=JYs1R9IMJr.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[18] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[19] Alvin Heng and Harold Soh. Selective amnesia: A continual learning approach to forgetting in deep generative models. *Advances in Neural Information Processing Systems*, 36:17170–17194, 2023.

[20] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 7514–7528, 2021.

[21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[22] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

[23] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] Anton Korznikov, Andrey V. Galichin, Alexey Dontsov, Oleg Rogov, Ivan Oseledets, and Elena Tutubalina. OrtSAE: Orthogonal sparse autoencoders uncover atomic features. In *ICLR 2026 Workshop on Representational Alignment (Re^4-Align)*, 2026. URL https://openreview.net/forum?id=f0xiIBVbNk.

[26] Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22691–22702, 2023.

[27] Patrick Leask, Bart Bussmann, Michael Pearce, Joseph Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis, 2025. URL https://arxiv.org/abs/2502.04878.

[28] Senmao Li, Joost van de Weijer, taihang Hu, Fahad Khan, Qibin Hou, Yaxing Wang, and jian Yang. 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*, 2024. URL https://openreview.net/forum?id=zpVPhvVKXk.

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[30] Mengyao Lyu, Yuhong Yang, Haiwen Hong, Hui Chen, Xuan Jin, Yuan He, Hui Xue, Jungong Han, and Guiguang Ding. 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*, pages 7559–7568, 2024.

[31] Alireza Makhzani and Brendan J. Frey. k-sparse autoencoders. *CoRR*, abs/1312.5663, 2013. URL https://api.semanticscholar.org/CorpusID:14850799.

[32] Mansi, Avinash Kori, Francesca Toni, and Soteris Demetriou. Selective fine-tuning for targeted and robust concept unlearning. *arXiv preprint arXiv:2602.07919*, 2026.

[33] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29), 2018.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[35] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[36] Javier Rando, Daniel Paleka, David Lindner, Lennart Heim, and Florian Tramer. Red-teaming the stable diffusion safety filter. In *NeurIPS ML Safety Workshop*, 2022. URL https://openreview.net/forum?id=zhDO3F35Uc.

[37] Junyu Ren and Lek-Heng Lim. Low-dimensional topology of deep neural networks. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=yZAbtq7Srt.

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

[39] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22522–22531, 2023.

[40] Yu-Lin Tsai, Chia-Yi Hsu, Chulin Xie, Chih-Hsun Lin, Jia You Chen, Bo Li, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Ring-a-bell! how reliable are concept removal methods for diffusion models? In *International Conference on Learning Representations*, volume 2024, pages 41543–41554, 2024.

[41] Jing Wu and Mehrtash Harandi. Scissorhands: Scrub data influence via connection sensitivity in networks. In *European Conference on Computer Vision*, pages 367–384. Springer, 2024.

[42] Jing Wu, Trung Le, Munawar Hayat, and Mehrtash Harandi. Erasing undesirable influence in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28263–28273, June 2025.

[43] Gong Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. 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*, pages 1755–1764, 2024.

[44] Yihua Zhang, Chongyu Fan, Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jiancheng Liu, Gaoyuan Zhang, Gaowen Liu, Ramana Rao Kompella, Xiaoming Liu, and Sijia Liu. 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. URL https://openreview.net/forum?id=t9aThFL1lE.

[45] Yimeng Zhang, Jinghan Jia, Xin Chen, Aochuan Chen, Yihua Zhang, Jiancheng Liu, Ke Ding, and Sijia Liu. To generate or not? safety-driven unlearned diffusion models are still easy to generate unsafe images... for now. In *European Conference on Computer Vision*, pages 385–403. Springer, 2024.

Appendix

A Implementation Details **15**

A.1 SAE Training Details 15

A.2 Subset Training Setting 15

A.3 Sensitivity Analysis on the Contrastive Loss Weight 16

A.4 Encoder Nonlinearity Ablation 16

A.5 Effect of SAE Layer Selection 17

A.6 Computational Efficiency and Storage Overheads 17

A.7 Unlearning Procedure Details 19

A.7.1 Feature Importance Scoring 19

A.7.2 Inference-Time Suppression 19

A.8 Joint Unlearning Formulation 19

A.9 Unlearning Hyperparameters 21

A.9.1 Sensitivity to Suppression Hyperparameters 23

A.10 Prompt Details 24

A.11 Cross-attention Weighted Pooling Details 26

B Further Experiments and Analyses **28**

B.1 SAeUron with Matched Training Epochs 28

B.2 Reconstruction–Separation Trade-off 29

B.3 Extended Overlap Analysis 30

B.4 Robustness to Concept Restoration and Prompt Variations 32

B.5 Gender-Distinct Nudity Unlearning 33

B.6 Fine-Grained Unlearning with Binary Forget/Retain Supervision 34

B.7 Generalization to Unseen Concepts 35

B.8 Object Sequential Unlearning 36

B.9 Compatibility with Alternative SAE Training Objectives 40

B.10 Preliminary Extension to an MM-DiT Backbone 41

C Additional Visualizations **42**

C.1 Additional Latent Space Visualizations 42

C.2 Detailed Performance Heatmaps 45

C.3 Qualitative Comparisons 49

D Broader Impacts **49**

A Implementation Details

A.1 SAE Training Details

We follow the SAE training setup of SAeUron [10] for our object and style blocks, training BatchTopK sparse autoencoders with $k = 32$ and an expansion factor of 16. For the joint block, we use a larger $k = 64$ to accommodate the increased granularity of joint (style–object) features. Optimization uses Adam [24] with a learning rate of 4×10^{-4} and a linear scheduler without warmup. We use an effective batch size of 131,072 and unit-normalize decoder weights after each training step. We set k_{aux} to a power of two close to $n/2$ and $\alpha = 1/32$. A latent is considered dead if it has not activated over the last 10M training samples.

We jointly optimize the standard reconstruction objective with a latent-space contrastive loss from scratch. The contrastive loss is applied to unit-normalized SAE latents and uses temperature $\tau = 0.07$ across all settings. The contrastive loss weight λ is set to 0.1 for the object SAE, 0.01 for the style SAE, and 1.0 for the joint SAE. For the joint SAE, we additionally apply a hard negative weighting factor of 4 to up-weight semantically close negatives that share either the object or the style with the anchor.

We train SAEs for the object, style, and joint settings. For object and style unlearning, we follow the empirically selected cross-attention blocks identified by SAeUron, using `up.1.1` for object unlearning and `up.1.2` for style unlearning. For joint unlearning, we apply SAEParate to the `up.1.1` block. Specifically, we train the object SAE for 40 epochs, the style SAE for 20 epochs, and the joint SAE for 40 epochs, using Stable Diffusion v1.5 [38] as the base model. Table 4 summarizes the key training hyperparameters and metrics. Training takes approximately 46, 20, and 69 hours on a single NVIDIA B200 GPU for the object, style, and joint SAEs, respectively.

Table 4: Summary of SAE training for SAEParate.

Block	**# Latents** n	k	α	λ	τ	**Hard Neg.**	**LR**	**Batch Size**	**Epochs**	**GPU-hours**
`up.1.1` (object)	20480	32	$\frac{1}{32}$	0.1	0.07	–	0.0004	131072	40	46h 21min
`up.1.2` (style)	20480	32	$\frac{1}{32}$	0.01	0.07	–	0.0004	131072	20	20h 18min
`up.1.1` (joint)	20480	64	$\frac{1}{32}$	1.0	0.07	4	0.0004	131072	40	68h 35min

SAEmnesia Adaptation for Joint Unlearning. Because SAEmnesia [7] does not originally support joint style–object unlearning, we treat each style–object pair as a single atomic concept and assign it to a distinct latent, preserving its one-to-one concept–latent design. The model is trained from scratch on the same `up.1.1` activation data as our joint SAE, with all other training settings following the official implementation except that the object–style decorrelation term is disabled. At inference, the assigned latent is suppressed and γ_c is tuned per target pair using the same validation and evaluation protocol as SAEParate.

A.2 Subset Training Setting

For ablations and additional analyses, we train SAEs on one-eighth of the full UnlearnCanvas training prompt set, keeping all other settings identical to Table 4. Within each experiment, all compared variants, including Vanilla SAE (SAeUron), are trained under this same setting, so comparisons remain controlled. However, absolute numbers are not directly comparable to the main results in Table 1; for example, Vanilla SAE attains lower UA on object unlearning in this setting (58.14%) than its reported full-data result (78.82%). This setting is used for the component ablation (Table 3), the cross-attention weighted pooling ablation (Table 13), the reconstruction–separation trade-off (Table 15), compatibility with Matryoshka SAE (Table 23), and prompt variations in joint unlearning (Table 17).

A.3 Sensitivity Analysis on the Contrastive Loss Weight

We analyze the sensitivity of SAEParate to the contrastive loss weight λ in the overall training objective,

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{SAE}} + \lambda \mathcal{L}_{\text{LSC}}. \tag{10}$$

The hyperparameter λ controls the trade-off between preserving the original diffusion activations through sparse reconstruction and encouraging concept-wise separation in the SAE latent space.

We conduct the ablation in the object-unlearning setting by varying $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ while keeping all other training and inference hyperparameters fixed. We report Unlearning Accuracy (UA), In-domain Retain Accuracy (IRA), Cross-domain Retain Accuracy (CRA), and their arithmetic mean.

Table 5: Sensitivity analysis of the contrastive loss weight λ on object unlearning. All other training and inference hyperparameters are held fixed. The best result in each column is highlighted in bold.

λ	UA (↑)	IRA (↑)	CRA (↑)	Avg. (↑)
0.1	**95.20%**	95.91%	94.26%	95.12%
0.2	94.61%	**97.09%**	94.79%	**95.50%**
0.3	93.33%	96.47%	94.53%	94.78%
0.4	92.65%	96.39%	**95.12%**	94.72%
0.5	94.61%	96.97%	94.37%	95.31%

As shown in Table 5, SAEParate remains relatively stable across the evaluated range of λ, with the average performance varying by approximately one percentage point. Nevertheless, the effect of increasing λ is not monotonic. While a stronger contrastive objective can encourage concept-wise separation, placing excessive emphasis on it does not consistently improve downstream unlearning performance.

A.4 Encoder Nonlinearity Ablation

Section 4.2 introduces a two-layer nonlinear encoder to provide additional flexibility for reshaping overlapping concept representations before sparsification. In particular, we adopt GeLU [18], whose smooth and mildly nonmonotonic response differs from the coordinate-wise monotonic ReLU. We hypothesize that this property enables more flexible local transformations under the concept-separation objective. Relatedly, Ren and Lim [37] show that, under certain assumptions, nonmonotonic activations can induce local folding of representations that is unavailable to coordinate-wise monotone activations. We regard this as a possible geometric explanation rather than a formal characterization of our encoder.

To examine the effect of this design, we compare otherwise identical two-layer encoders using Linear, ReLU, or GeLU as the intermediate activation σ in Eq. 9. The SAE width, hidden dimensionality, sparsity level, training objective, and unlearning hyperparameters are fixed across all variants. The Linear variant contains no intermediate nonlinearity and is therefore expressively equivalent to a single affine transformation before the final ReLU, allowing us to isolate the contribution of nonlinear transformation.

Table 6: Ablation of the intermediate encoder activation on object unlearning. All variants use the same two-layer architecture and differ only in the intermediate activation. The best result in each column is bolded.

Activation	UA (↑)	IRA (↑)	CRA (↑)	Avg. (↑)
Linear	74.02%	**97.34%**	**96.49%**	89.28%
ReLU	90.88%	96.77%	95.31%	94.32%
GeLU	**95.20%**	95.91%	94.26%	**95.12%**

Introducing an intermediate nonlinearity yields most of the improvement: ReLU increases UA from 74.02% to 90.88% and the average from 89.28% to 94.32%. GeLU further raises UA to 95.20% and achieves the best average of 95.12%, while maintaining retention performance comparable to ReLU. These results suggest that under the LSC objective, introducing the nonlinear hidden transformation

accounts for most of the improvement over the linear encoder, with GeLU providing a further benefit for target-concept removal.

A.5 Effect of SAE Layer Selection

SAeUron [10] uses `up.1.1` for object unlearning and `up.1.2` for style unlearning, and we follow the same layer choices in the corresponding settings for a fair comparison. For joint style–object unlearning, which requires representing both types of concepts within a single SAE, we use `up.1.1` as the default configuration in our main experiments. To examine the sensitivity of SAEParate to this layer choice, we additionally evaluate both SAEParate and SAeUron using `up.1.2`. Table 7 compares the two candidate layers under the same joint unlearning protocol.

Table 7: Effect of SAE insertion layer on joint style–object unlearning. Results are reported for both `up.1.1` and `up.1.2`.

Method	Layer	UA (↑)	SC (↑)	OC (↑)	UP (↑)	Avg. (↑)
Vanilla SAE (SAeUron)	`up.1.1`	63.00%	16.98%	75.84%	71.90%	56.93%
SAEParate	`up.1.1`	**94.11%**	**93.49%**	**91.66%**	**98.82%**	**94.52%**
Vanilla SAE (SAeUron)	`up.1.2`	93.90%	0.89%	55.73%	54.63%	51.29%
SAEParate	`up.1.2`	**94.00%**	**93.88%**	**91.60%**	**96.90%**	**94.10%**

SAEParate consistently maintains strong joint unlearning performance across the two candidate layers, indicating that its effectiveness is not tied to a specific SAE insertion point. The comparison also suggests that the improvement over the reconstruction-oriented SAE baseline persists across layer choices, supporting the robustness of the proposed representation-learning approach.

A.6 Computational Efficiency and Storage Overheads

We also conduct an efficiency comparison with existing methods, including SAeUron. For a fair comparison, we follow the SAE training batch size of 4096 reported in Appendix E of SAeUron [10]. As shown in Table 8, since SAEParate introduces supervised contrastive regularization only as an additional training objective, it does not substantially increase the training memory budget under the same batch size. Moreover, although SAEParate employs a lightweight enhanced SAE encoder, the resulting SAE checkpoint size remains nearly unchanged compared to SAeUron.

Table 8: Efficiency comparison with existing diffusion unlearning methods. Memory is measured under the same SAE training batch size of 4096 following SAeUron Appendix E. Storage denotes additional method-specific files required for unlearning or inference.

Method	Efficiency	
	Memory (GB) (↓)	Storage (GB) (↓)
ESD [13]	17.8	4.3
FMN [43]	17.9	4.2
UCE [14]	5.1	1.7
CA [26]	10.1	4.2
SalUn [12]	30.8	4.0
SEOT [28]	7.3	0.0
SPM [30]	6.9	0.0
EDiff [42]	27.8	4.0
SHS [41]	31.2	4.0
SAeUron [10]	2.8	0.2
SAEParate	3.4	0.2

Inference-Time Overhead. We additionally measure end-to-end image-generation wall-clock time on a single NVIDIA RTX 4090 under identical generation settings. As shown in Table 9, both SAE-based methods introduce only a small inference overhead relative to the original Stable Diffusion backbone.

Table 9: Inference-time efficiency measured on a single NVIDIA RTX 4090. Relative inference time is normalized by the Stable Diffusion backbone.

Variant	Relative Inference Time	Inference Time (s)
SD Backbone	1.00×	2.9252
Vanilla SAE (SAeUron)	1.02×	2.9776
SAEParate	1.02×	2.9907

SAEParate processes the spatial activations through a single batched SAE forward pass at one U-Net hook point per denoising step, resulting in approximately 2% inference-time overhead relative to the backbone. The additional storage requirement is also modest: we train one SAE per unlearning setting, and each FP32 SAE checkpoint occupies approximately 200 MB.

A.7 Unlearning Procedure Details

We perform concept unlearning by suppressing a small set of concept-specific SAE features at inference time. Both the feature importance score function and the suppression mechanism are taken directly from Sections 4.1 and 4.2 of SAeUron [10] without modification; we describe them here for completeness and specify the implementation used in our experiments.

A.7.1 Feature Importance Scoring

Following SAeUron [10], we identify SAE features that strongly and exclusively correspond to a target concept c at each denoising timestep t. Given a dataset of activations $\mathcal{D} = \mathcal{D}_c \cup \mathcal{D}_{\neg c}$ partitioned by concept presence, the importance of the i-th feature for concept c at timestep t is defined as

$$\text{score}(i, t, c, \mathcal{D}) = \frac{\mu(i, t, \mathcal{D}_c)}{\sum_{j=1}^{n} \mu(j, t, \mathcal{D}_c) + \delta} - \frac{\mu(i, t, \mathcal{D}_{\neg c})}{\sum_{j=1}^{n} \mu(j, t, \mathcal{D}_{\neg c}) + \delta}, \tag{11}$$

where $\mu(i, t, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} f_i(x_t)$ is the average activation of feature i at timestep t, n is the SAE width, and δ is a small constant for numerical stability. Normalization by the per-subset feature mass prevents broadly active features from dominating the score, so high-scoring features are those activating strongly on $\mathcal{D}_c$ while remaining weak on $\mathcal{D}_{\neg c}$.

A.7.2 Inference-Time Suppression

For each concept c, we form the suppression set $\mathcal{F}_c$ by selecting the top-τ_c features by score, after filtering out dead features and those whose activation frequency exceeds the 99th percentile of the feature density distribution. For features in $\mathcal{F}_c$, the activation is modified as

$$\tilde{f}_i(x) = \begin{cases} \gamma_c \, \mu(i, t, \mathcal{D}_c) \, f_i(x), & \text{if } i \in \mathcal{F}_c \text{ and } f_i(x) > \mu(i, t, \mathcal{D}), \\ f_i(x), & \text{otherwise,} \end{cases} \tag{12}$$

where the condition $f_i(x) > \mu(i, t, \mathcal{D})$ prevents random ablation when activations are uniformly low. The modified activations are decoded back through the SAE decoder while preserving the reconstruction error. The percentile is selected via grid search over the score distribution; the chosen values for each experimental setup are reported in Appendix A.9.

A.8 Joint Unlearning Formulation

To enable precise unlearning in the style–object combination setup, we extend the SupCon objective in Eq. 7 along three axes that exploit the compositional structure of the target: (i) the positive set $\mathcal{P}(i)$ is restricted to samples matching on *both* attributes, (ii) the negatives in the partition function of Eq. 7 are reweighted to emphasize partial-match neighbors, and (iii) the cross-attention weights $\alpha_{i,p}$ in Eq. 5 are computed from prompts specifying the joint concept. Throughout this section we reuse the per-sample representation $u_i = r_i / \|r_i\|_2$ defined in Section 4.1.

Joint positive set. Each target combination consists of two attributes, an object and a style. Let y_i^{cls} and y_i^{thm} denote the object and style labels of sample i, and let $\mathcal{V}$ be the set of target (object, style) combinations. We replace the single-label positive set $\mathcal{P}(i) = \{j \neq i \mid y_j = y_i\}$ from Eq. 7 with one that requires agreement on both attributes within $\mathcal{V}$:

$$\mathcal{P}^{+}(i) = \{j \neq i \mid y_j^{cls} = y_i^{cls},\ y_j^{thm} = y_i^{thm},\ (y_i^{cls}, y_i^{thm}) \in \mathcal{V}\}. \tag{13}$$

Reducing to a single-attribute (i.e., $y_i \equiv y_i^{cls}$ or $y_i \equiv y_i^{thm}$) recovers Eq. 7 exactly.

Hard negative reweighting. A central difficulty specific to the joint setting is that samples sharing exactly one attribute sit in an ambiguous region of representation space. For instance, if the unlearning target is *Dogs in Cartoon style*, then *Flowers in Cartoon style* (matching only the style) and *Dogs in Ink Art style* (matching only the object) are partial-match samples that remain close to the anchor under $u_i^{\top} u_j$ yet belong to distinct concepts. The standard SupCon partition function in Eq. 7 treats these as ordinary negatives, providing an insufficiently strong separation signal. We therefore define the hard negative set

$$\mathcal{H}(i) = \{j \neq i \mid (y_j^{cls} = y_i^{cls}) \oplus (y_j^{thm} = y_i^{thm})\}, \tag{14}$$

where $\oplus$ denotes exclusive-or, i.e., the condition holds iff exactly one of the two equalities is satisfied. Rather than altering the cosine similarity itself, we modify the per-pair logit $u_i^\top u_j/\tau$ used inside Eq. 7 by an additive bias:

$$\tilde{s}_{ij} = \frac{u_i^\top u_j}{\tau} + \log w_{ij}, \qquad w_{ij} = \begin{cases} \omega_{\mathrm{HN}} & \text{if } j \in \mathcal{H}(i), \\ 1 & \text{otherwise.} \end{cases} \tag{15}$$

Inflating the logit of a hard negative by $\log \omega_{\mathrm{HN}}$ increases its mass in the partition function, yielding a stronger push away from partial-match neighbors while leaving non-hard negatives at their original SupCon weighting. The joint SupCon loss is then a direct generalization of Eq. 7:

$$\mathcal{L}_{\text{Joint}} = -\frac{1}{|\mathcal{I}^+|} \sum_{i \in \mathcal{I}^+} \frac{1}{|\mathcal{P}^+(i)|} \sum_{j \in \mathcal{P}^+(i)} \left(\tilde{s}_{ij} - \log \sum_{k \neq i} \exp(\tilde{s}_{ik}) \right), \tag{16}$$

where $\mathcal{I}^+ = \{i \in \mathcal{I} \mid |\mathcal{P}^+(i)| > 0\}$. Setting $\omega_{\mathrm{HN}} = 1$ and replacing $\mathcal{P}^+$ with $\mathcal{P}$ recovers Eq. 7, making explicit that $\mathcal{L}_{\text{Joint}}$ adds two compositional ingredients on top of the base objective.

Joint cross-attention weighting and target-biased sampling. The spatial weighting in Eq. 5 is concept-conditioned through the prompt token set $\mathcal{C}_i$. For joint unlearning we set $\mathcal{C}_i$ to include both the target object and style tokens, so that $\alpha_{i,p}$ reflects spatial relevance to both attributes of the joint concept; the aggregated representation

$$r_i = \sum_{p=1}^{S} \alpha_{i,p}\, z_{i,p} \tag{17}$$

from Eq. 6 is otherwise unchanged. Finally, to ensure that target combinations receive sufficient gradient signal, we enforce that at least one fifth of each mini-batch consists of samples drawn from $\mathcal{V}$. Together with $\mathcal{P}^+$ and $\mathcal{H}$, this prevents $\mathcal{L}_{\text{Joint}}$ from being dominated by non-target pairs and provides consistent exposure to the fine-grained distinctions the SAE must learn, encouraging tight, well-separated clusters for each (object, style) combination.

A.9 Unlearning Hyperparameters

For each concept c, τ_c denotes the number of selected features used to suppress the target concept, and γ_c denotes the negative multiplier controlling the suppression strength. Following SAeUron, which selects these hyperparameters through validation-based search, we tune τ_c and γ_c on validation prompts for object and joint unlearning, while fixing $\tau_c = 1$ and $\gamma_c = -0.1$ for style unlearning. This yields one hyperparameter pair per target concept, resulting in 20 pairs for object unlearning and 50 pairs for joint unlearning. The selected hyperparameters are reported in Tables 10 and 11, respectively.

Table 10: Hyperparameters of our method for object unlearning.

Object	**Selected features** τ_c	**Multiplier** γ_c
Architectures	3	-0.1
Bears	2	-0.1
Birds	3	-0.25
Butterfly	2	-0.1
Cats	2	-0.1
Dogs	2	-0.1
Fishes	1	-0.1
Flame	3	-5.0
Flowers	3	-1.0
Frogs	2	-0.1
Horses	3	-0.75
Human	4	-0.25
Jellyfish	1	-10.0
Rabbits	2	-1.0
Sandwiches	3	-0.5
Sea	3	-1.0
Statues	1	-0.1
Towers	2	-0.1
Trees	3	-0.25
Waterfalls	3	-0.25

Table 11: Hyperparameters of our method for joint (object, style) unlearning.

Object	**Style**	τ_c	γ_c	**Object**	**Style**	τ_c	γ_c
Architectures	Abstractionism	2	-1.0	Dogs	Magic_Cube	2	-9.0
Bears	Artist_Sketch	3	-9.0	Fishes	Meta_Physics	2	-7.0
Birds	Blossom_Season	4	-11.0	Flame	Meteor_Shower	2	-5.0
Butterfly	Bricks	4	-7.0	Flowers	Monet	4	-11.0
Cats	Byzantine	3	-1.0	Frogs	Mosaic	3	-9.0
Dogs	Cartoon	2	-3.0	Horses	Neon_Lines	3	-3.0
Fishes	Cold_Warm	3	-11.0	Human	On_Fire	4	-7.0
Flame	Color_Fantasy	4	-9.0	Jellyfish	Pastel	2	-5.0
Flowers	Comic_Etch	2	-1.0	Rabbits	Pencil_Drawing	5	-5.0
Frogs	Crayon	13	-11.0	Sandwiches	Picasso	2	-5.0
Horses	Cubism	2	-7.0	Sea	Pop_Art	3	-11.0
Human	Dadaism	3	-5.0	Statues	Red_Blue_Ink	3	-9.0
Jellyfish	Dapple	2	-5.0	Towers	Rust	2	-1.0
Rabbits	Defoliation	11	-11.0	Waterfalls	Sketch	2	-5.0
Sandwiches	Early_Autumn	2	-11.0	Architectures	Sponge_Dabbed	3	-9.0
Sea	Expressionism	3	-9.0	Bears	Structuralism	3	-9.0
Statues	Fauvism	2	-7.0	Birds	Superstring	4	-11.0
Towers	French	2	-1.0	Butterfly	Surrealism	3	-11.0
Trees	Glowing_Sunset	3	-9.0	Cats	Ukiyoe	12	-7.0
Waterfalls	Gorgeous_Love	2	-7.0	Dogs	Van_Gogh	3	-1.0
Architectures	Greenfield	2	-9.0	Fishes	Vibrant_Flow	3	-7.0
Bears	Impressionism	4	-5.0	Flame	Warm_Love	3	-7.0
Birds	Ink_Art	4	-5.0	Flowers	Warm_Smear	4	-5.0
Butterfly	Joy	2	-11.0	Frogs	Watercolor	2	-11.0
Cats	Liquid_Dreams	2	-9.0	Horses	Winter	2	-3.0

A.9.1 Sensitivity to Suppression Hyperparameters

We further analyze the sensitivity of SAEParate to the two hyperparameters governing inference-time suppression: the multiplier γ_c, which controls the strength of suppression, and the percentile used to determine τ_c. We vary $\gamma_c \in \{-1, -3, -5, -7, -9, -11\}$ and the percentile for $\tau_c \in \{99.95, 99.99, 99.995\}$ while keeping all other training and inference settings fixed.

We conduct this analysis in the joint style–object unlearning setting and report results averaged across the evaluated target concepts. We measure Unlearning Accuracy (UA), Style Consistency (SC), Object Consistency (OC), and their arithmetic mean (Avg.).

Table 12: Sensitivity to inference-time suppression hyperparameters in joint unlearning. The percentile is used to determine τ_c. Results are averaged across the evaluated joint concepts.

γ_c	**Percentile for τ_c**	**UA (↑)**	**SC (↑)**	**OC (↑)**	**Avg. (↑)**
-1	99.95	56.4	87.9	88.0	77.4
	99.99	57.0	88.0	88.0	77.7
	99.995	61.4	88.0	88.0	79.1
-3	99.95	81.0	86.0	86.4	84.5
	99.99	86.8	86.2	86.4	86.5
	99.995	86.6	86.4	86.4	86.5
-5	99.95	87.6	84.9	85.3	85.9
	99.99	91.6	85.1	85.3	87.3
	99.995	92.8	85.3	85.3	87.8
-7	99.95	92.4	84.0	84.6	87.0
	99.99	94.2	84.5	84.6	87.8
	99.995	94.4	84.6	84.6	87.9
-9	99.95	94.2	83.1	84.0	87.1
	99.99	94.4	83.6	84.0	87.3
	99.995	95.6	84.0	84.0	87.9
-11	99.95	96.2	82.5	83.5	87.4
	99.99	96.4	83.0	83.5	87.6
	99.995	96.8	83.5	83.5	87.9

We perform the sensitivity analysis under the joint unlearning setting. As shown in Table 12, increasing the magnitude of γ_c consistently improves UA while gradually reducing retention. Consequently, the overall average remains stable at approximately 87–88% across a broad range of moderate-to-strong suppression settings. The percentile used to determine τ_c has a comparatively smaller effect, indicating that SAEParate does not rely on a narrowly tuned suppression configuration.

A.10 Prompt Details

For training the SAE in concept unlearning tasks, we collect internal diffusion activations across 50 denoising steps using the prompts provided in UnlearnCanvas [44]. Specifically, we use 80 simple one-sentence *anchor prompts* for each object class, originally employed in CA [26]. To train style-related representations, we append the postfix "`in {style} style.`" to each anchor prompt. UnlearnCanvas provides 20 object classes and 50 style themes.

The object classes are:

- *Architectures, Bears, Birds, Butterfly, Cats, Dogs, Fishes, Flame, Flowers, Frogs, Horses, Human, Jellyfish, Rabbits, Sandwiches, Sea, Statues, Towers, Trees, Waterfalls*.

The style themes are:

- *Abstractionism, Artist Sketch, Blossom Season, Bricks, Byzantine, Cartoon, Cold Warm, Color Fantasy, Comic Etch, Crayon, Cubism, Dadaism, Dapple, Defoliation, Early Autumn, Expressionism, Fauvism, French, Glowing Sunset, Gorgeous Love, Greenfield, Impressionism, Ink Art, Joy, Liquid Dreams, Magic Cube, Meta Physics, Meteor Shower, Monet, Mosaic, Neon Lines, On Fire, Pastel, Pencil Drawing, Picasso, Pop Art, Red Blue Ink, Rust, Sketch, Sponge Dabbed, Structuralism, Superstring, Surrealism, Ukiyoe, Van Gogh, Vibrant Flow, Warm Love, Warm Smear, Watercolor, Winter*.

The joint combinations follow the style–object Combination Unlearning setup in Appendix B.5 of UnlearnCanvas [44].

- *(Architectures, Abstractionism), (Bears, Artist Sketch), (Birds, Blossom Season), (Butterfly, Bricks), (Cats, Byzantine), (Dogs, Cartoon), (Fishes, Cold Warm), (Flame, Color Fantasy), (Flowers, Comic Etch), (Frogs, Crayon), (Horses, Cubism), (Human, Dadaism), (Jellyfish, Dapple), (Rabbits, Defoliation), (Sandwiches, Early Autumn), (Sea, Expressionism), (Statues, Fauvism), (Towers, French), (Trees, Glowing Sunset), (Waterfalls, Gorgeous Love), (Architectures, Greenfield), (Bears, Impressionism), (Birds, Ink Art), (Butterfly, Joy), (Cats, Liquid Dreams), (Dogs, Magic Cube), (Fishes, Meta Physics), (Flame, Meteor Shower), (Flowers, Monet), (Frogs, Mosaic), (Horses, Neon Lines), (Human, On Fire), (Jellyfish, Pastel), (Rabbits, Pencil Drawing), (Sandwiches, Picasso), (Sea, Pop Art), (Statues, Red Blue Ink), (Towers, Rust), (Waterfalls, Sketch), (Architectures, Sponge Dabbed), (Bears, Structuralism), (Birds, Superstring), (Butterfly, Surrealism), (Cats, Ukiyoe), (Dogs, Van Gogh), (Fishes, Vibrant Flow), (Flame, Warm Love), (Flowers, Warm Smear), (Frogs, Watercolor), (Horses, Winter)*.

For feature score calculation, which is used to identify concept-specific SAE features, we use a validation set based on the *anchor prompts* same as SAeUron [10]. For style unlearning, we use 20 selected *anchor prompts*, each appended with the postfix "`in {style} style.`" for every style theme. The 20 prompts are:

1. "Gothic cathedral with flying buttresses and stained glass windows."
2. "Bear dressed as a medieval knight in armor."
3. "Bird with iridescent, oil-slick-like feathers."
4. "Butterfly emerging from a jeweled cocoon."
5. "Cat wearing a superhero cape, leaping between buildings."
6. "Dog wearing aviator goggles, piloting an airplane."
7. "Goldfish swimming in a crystal-clear bowl."
8. "Candle flame flickering in an old mysterious library."
9. "Flower blooming in a snow-covered landscape."
10. "Frog whose croak sounds like a jazz trumpet."
11. "Wild horse galloping across the prairie at sunrise."
12. "Man hiking through a dense forest."

13. “Jellyfish floating in deep blue water.”
14. “Rabbit peering out from a burrow.”
15. “Classic BLT sandwich on toasted bread.”
16. “Sea waves crashing over ancient coastal ruins.”
17. “Forgotten hero statue covered in ivy.”
18. “Tower soaring above the clouds.”
19. “Majestic oak tree in a serene forest.”
20. “Moonlit waterfall in a serene forest.”

For object unlearning, we use the full set of 80 *anchor prompts* for each object class without the style postfix. The joint unlearning setting uses the same 80 *anchor prompts* for each (object, style) combination, with the corresponding style postfix appended.

At evaluation time, we follow the UnlearnCanvas prompt format and generate style–object prompts using the template

‘‘`An image of {object} in {style} style.`’’

For example, ‘‘`An image of Architectures in Abstractionism style.`’’.

A.11 Cross-attention Weighted Pooling Details

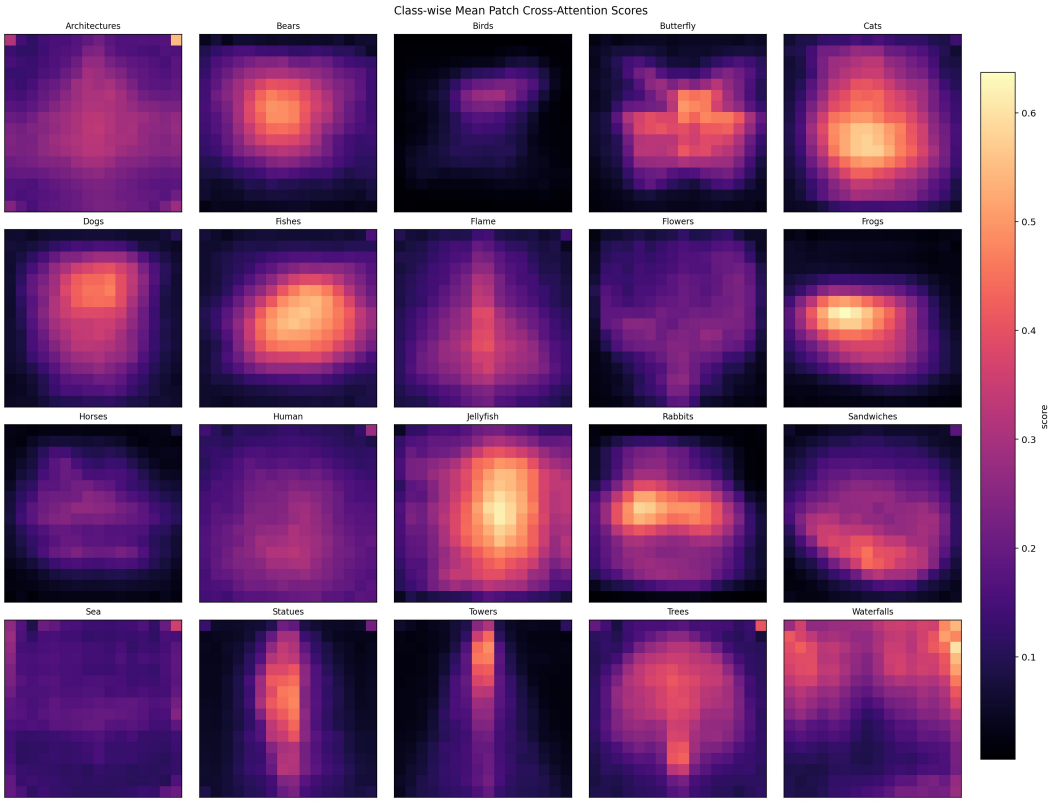


Figure 6: Cross-attention score visualization of object classes.

We compute the cross-attention-guided weights in Eq. 5 by reusing the spatial grounding signal already present in the diffusion model's cross-attention maps. For each sample, we identify the prompt tokens corresponding to the target concept, denoted by $\mathcal{C}_i$. This set contains all tokenizer units matched to the concept phrase, since object names or style phrases may be split into multiple tokens.

We extract attention probabilities from the cross-attention layer in the same UNet block where the SAE is trained. We attach a forward hook to this layer and access the image hidden states and text encoder hidden states passed through it during the diffusion process. Since this layer performs text-conditioned cross-attention, image hidden states are used as queries, while text encoder hidden states provide the keys and values. The resulting attention probabilities therefore indicate how each spatial position in the image feature map attends to each prompt token.

Within the hook, we reproduce the layer's internal attention computation using its own projection weights to compute the query–key attention probabilities. Hence, the attention maps are extracted directly from the diffusion UNet's cross-attention operation at the SAE training block, rather than from an external localization or segmentation module.

Given the resulting attention probabilities $A_{i,h,p,k}$ over attention head h, spatial position p, and text token k, we compute a concept-level relevance score for each spatial position by averaging over heads and aggregating the attention mass assigned to tokens in $\mathcal{C}_i$. The scores are then normalized across spatial positions to obtain $\alpha_{i,p}$, where $\sum_p \alpha_{i,p} = 1$. These weights are soft pooling coefficients rather than binary masks.

The weights are computed independently for each denoising timestep and used to construct the sample-level representation for supervised contrastive learning:

$$r_i = \sum_{p=1}^{S} \alpha_{i,p} z_{i,p}. \tag{18}$$

Thus, concept-relevant spatial regions contribute more strongly to the contrastive representation, while background or concept-irrelevant regions are down-weighted. Figure 6 visualizes these stored spatial weights averaged over samples of the same concept.

Table 13: Cross-attention weighted pooling ablation results on object unlearning.

Variants	Unlearning Performance			Avg. (↑)
	UA (↑)	IRA (↑)	CRA (↑)	
w/o-CAWP	79.41%	97.28%	95.56%	90.75%
SAEParate	95.59%	96.48%	93.46%	95.18%

To assess the contribution of cross-attention weighted pooling, we conduct an ablation study under the subset setting (Appendix A.2). The *w/o*-CAWP variant constructs the representation for the contrastive loss by mean-pooling over all spatial positions, i.e., replacing $\alpha_{i,p}$ in Eq. 6 with uniform weights $1/S$. We compare this variant with our full method, which uses cross-attention weighted pooling to emphasize concept-relevant spatial positions. As shown in Table 13, *w/o*-CAWP exhibits a weaker trade-off between unlearning and retention, yielding lower unlearning accuracy.

These results indicate that treating all spatial positions equally can introduce concept-irrelevant evidence into the contrastive representation, such as background regions or co-occurring non-target objects. This weakens the concept-specific contrastive signal, leading to less discriminative feature separation and a weaker unlearning–retention trade-off.

B Further Experiments and Analyses

B.1 SAeUron with Matched Training Epochs

For the main comparison, we report the published SAeUron results obtained under its original training configuration. To examine whether the performance gap could be attributed to differences in SAE training duration, we additionally retrain the official SAeUron implementation using the same number of epochs as SAEParate. As shown in Table 14, extending SAeUron training to 40 epochs for object unlearning and 20 epochs for style unlearning yields comparable or slightly worse performance than its original 5- and 10-epoch schedules, respectively. This controlled comparison indicates that the gains of SAEParate are not simply due to longer SAE training.

Table 14: Comparison with SAeUron under matched SAE training epochs. We retrain SAeUron with the same number of epochs used by our method (40 for object, 20 for style) and compare against its original training schedule (5 / 10 epochs) and our method. Best results in **bold**.

Setting	Method	UA (↑)	IRA (↑)	CRA (↑)	Avg. (↑)
Object	SAeUron (5 epochs, original)	77.35%	95.68%	**95.20%**	89.41%
	SAeUron (40 epochs, matched)	75.98%	95.49%	94.45%	88.64%
	SAEParate (40 epochs)	**95.20%**	**95.91%**	94.26%	**95.12%**
Style	SAeUron (10 epochs, original)	96.30%	98.86%	97.40%	97.52%
	SAeUron (20 epochs, matched)	96.90%	98.76%	97.80%	97.82%
	SAEParate (20 epochs)	**99.60%**	**99.24%**	**98.30%**	**99.05%**

B.2 Reconstruction–Separation Trade-off

Since SAEParate introduces an explicit concept-separation objective in addition to sparse reconstruction, we examine whether the improved concept discrimination comes at the cost of reconstruction fidelity. We compare SAEParate with a Vanilla SAE (SAeUron), both trained under the subset setting (Appendix A.2) with a matched budget of 20K optimization steps.

We measure reconstruction quality using the fraction of variance unexplained (FVU). For each microbatch b, we compute

$$\mathrm{FVU}_b = \frac{\sum_{i \in b} \|x_i - \hat{x}_i\|_2^2}{\sum_{i \in b} \|x_i - \bar{x}_b\|_2^2}, \tag{19}$$

where x_i and $\hat{x}_i$ denote the original and reconstructed activations, respectively, and $\bar{x}_b$ is the mean activation within microbatch b. We report the average of the microbatch-level ratios over the corresponding logging interval:

$$\mathrm{FVU} = \frac{1}{B} \sum_{b=1}^{B} \mathrm{FVU}_b, \tag{20}$$

where B denotes the number of microbatches in the interval. Lower FVU indicates better reconstruction fidelity.

As shown in Table 15, SAEParate exhibits a moderate increase in reconstruction error compared with the Vanilla SAE, with FVU increasing from 0.172 to 0.217. This trade-off is expected because SAEParate is not optimized solely for reconstruction: the additional concept-aware contrastive objective explicitly reorganizes the latent space to promote concept discrimination.

Importantly, the reduction in reconstruction fidelity is accompanied by a substantial improvement in latent-space separation. SAEParate increases the cluster rate from 0.93 to 1.00 and reduces the centroid margin from 0.73 to 0.06, indicating markedly clearer concept-wise separation. At the same time, CLIPScore remains comparable and is slightly higher for SAEParate (29.323 vs. 29.158), suggesting that the moderate reconstruction trade-off does not translate into a clear degradation in generated-image quality. These results support our design objective: for selective diffusion unlearning, preserving reconstruction fidelity alone is not sufficient, and allowing a modest reconstruction trade-off can yield substantially more concept-discriminative sparse representations.

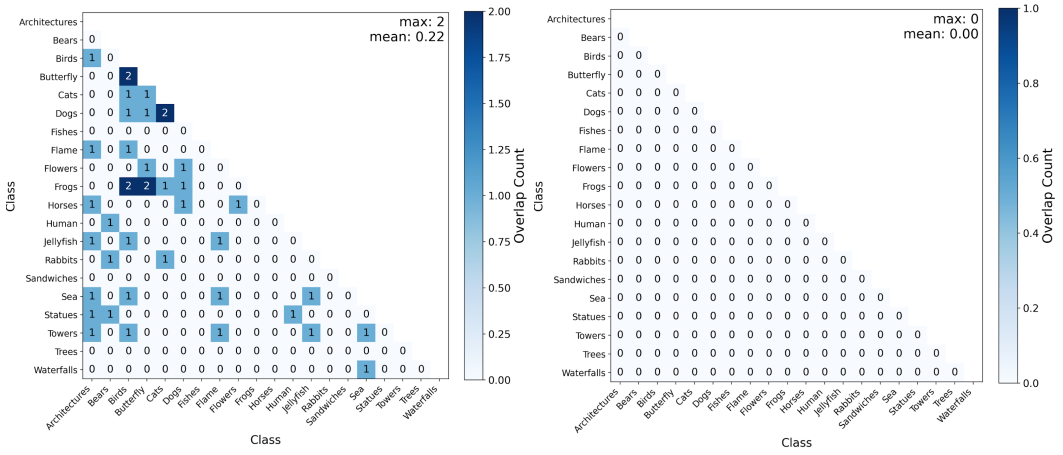
Table 15: Reconstruction–separation trade-off between Vanilla SAE (SAeUron) and SAEParate under the subset setting (Appendix A.2), with a matched budget of 20K steps. Lower FVU and centroid margin are better, whereas higher cluster rate and CLIPScore are better.

Method	FVU (↓)	Cluster Rate (↑)	Centroid Margin (↓)	CLIPScore (↑)
Vanilla SAE (SAeUron)	**0.172**	0.93	0.73	29.158
SAEParate	0.217	**1.00**	**0.06**	**29.323**

B.3 Extended Overlap Analysis

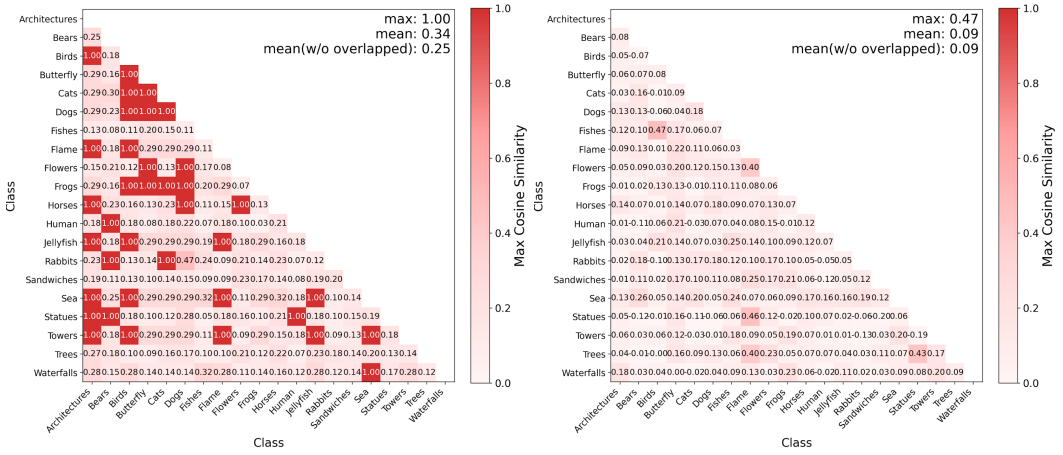
We provide the full class-wise overlap analysis for both object and joint style–object unlearning, extending the subset visualization presented in Figure 4. Specifically, we measure (i) the pairwise overlap counts between concept-specific selected feature groups and (ii) the pairwise maximum cosine similarity between these groups.

For object unlearning, Figure 7 shows that the Vanilla SAE (SAeUron) exhibits substantial redundancy in selected feature groups across different concepts, whereas SAEParate considerably reduces such overlap. Similarly, Figure 8 shows lower pairwise maximum cosine similarity under SAEParate, indicating that concept-specific feature groups become more distinct in the learned latent space.



Vanilla SAE (SAeUron) **SAEParate (Ours)**

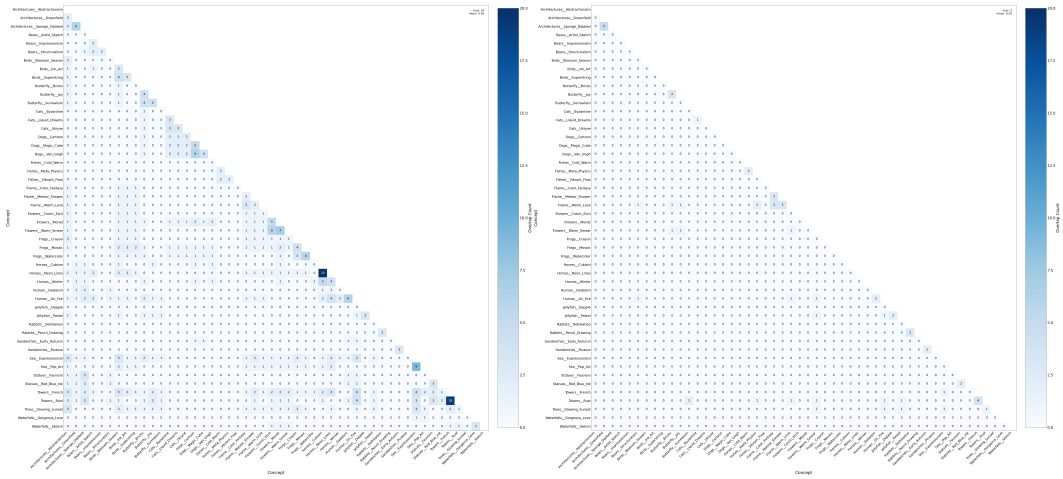
Figure 7: Full class-wise overlap counts of selected feature groups for object unlearning.



Vanilla SAE (SAeUron) **SAEParate (Ours)**

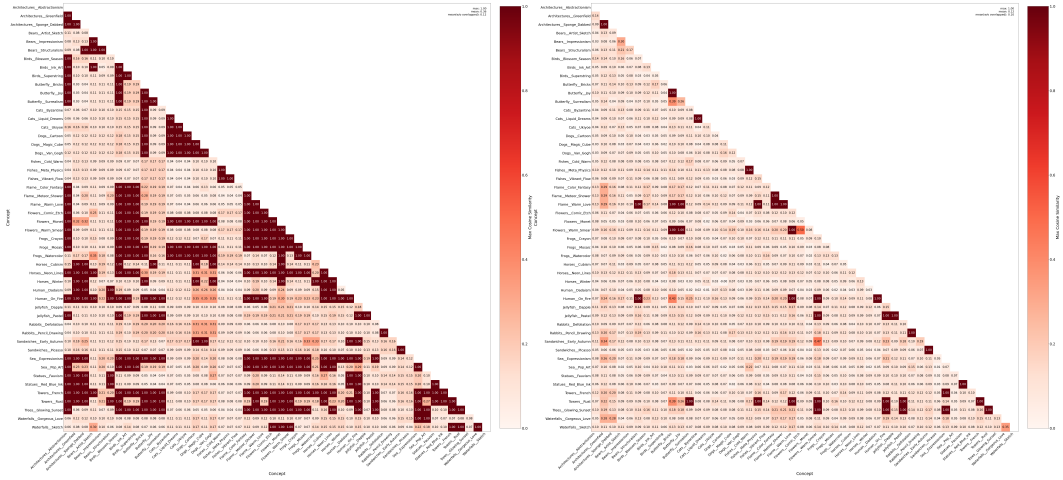
Figure 8: Full class-wise maximum cosine similarity between selected feature groups for object unlearning.

We further extend the analysis to joint style–object unlearning. Figures 9 and 10 show that SAEParate reduces both selected-feature overlap and pairwise maximum cosine similarity, indicating improved separation even among concepts sharing an object or style.



Vanilla SAE (SAeUron) **SAEParate (Ours)**

Figure 9: Full pairwise overlap counts of selected feature groups for joint style–object unlearning. SAEParate exhibits substantially less feature sharing across different style–object concepts.



Vanilla SAE (SAeUron) **SAEParate (Ours)**

Figure 10: Full pairwise maximum cosine similarity between selected feature groups for joint style–object unlearning. Lower off-diagonal similarities indicate improved separation between concept-specific feature groups.

B.4 Robustness to Concept Restoration and Prompt Variations

Concept-Restoration Attacks. We first examine whether removed concepts can be restored through adversarial prompting. Specifically, we consider Ring-A-Bell [40] and UnlearnDiffAtk [45] under object unlearning, and compare SAEParate with SAeUron. For Ring-A-Bell, we extend the attack to object concepts in UnlearnCanvas by constructing concept-present/concept-absent prompt pairs and estimating the corresponding empirical concept vectors. We also evaluate UnlearnDiffAtk, which iteratively optimizes the input prompt using gradients from the target diffusion model.

Table 16: Robustness against Ring-A-Bell and UnlearnDiffAtk under object unlearning. We report unlearning accuracy (UA) before and after each attack.

	UnlearnDiffAtk		Ring-A-Bell	
Method	Before (↑)	After (↑)	Before (↑)	After (↑)
Vanilla SAE (SAeUron)	83.7	34.2	82.0	79.9
SAEParate	**94.9**	**49.7**	**93.6**	**92.8**

As shown in Table 16, SAEParate achieves higher post-attack UA than SAeUron under both attacks, indicating that its unlearning effect is better preserved under concept-restoration attempts.

Prompt Variations. We next examine whether SAEParate generalizes beyond the fixed prompt template used during training and evaluation. For joint style–object unlearning, we consider three types of prompt variations: template paraphrasing, object synonyms, and natural-language reformulations of style descriptions.

Table 17: Robustness of SAEParate to prompt variations under joint style–object unlearning.

Setting	UA (↑)	SC (↑)	OC (↑)	Avg. (↑)
Original prompts	**86.2%**	81.8%	**91.4%**	**86.5%**
Reformulated prompts	81.3%	**82.0%**	91.3%	84.9%

As shown in Table 17, obtained under the subset setting (Appendix A.2), reformulating the prompts lowers UA from 86.2% to 81.3%, whereas SC and OC remain essentially unchanged (81.8% to 82.0% and 91.4% to 91.3%). The average therefore decreases only modestly, from 86.5% to 84.9%, and the drop is confined to target removal rather than retention, suggesting that the learned representations are not tied to the exact prompt template used during training.

B.5 Gender-Distinct Nudity Unlearning

For real-world adaptation of our method, we consider gender-specific nudity unlearning scenarios. We collect internal activations from a pre-trained Stable Diffusion v1.4 model [38] using randomly selected 5K captions from COCO [29] train2014, supplemented with a small number of prompts containing *"naked man"* and *"naked woman"*. We train and apply the SAE on the `up.1.1` cross-attention block following the object unlearning setup.

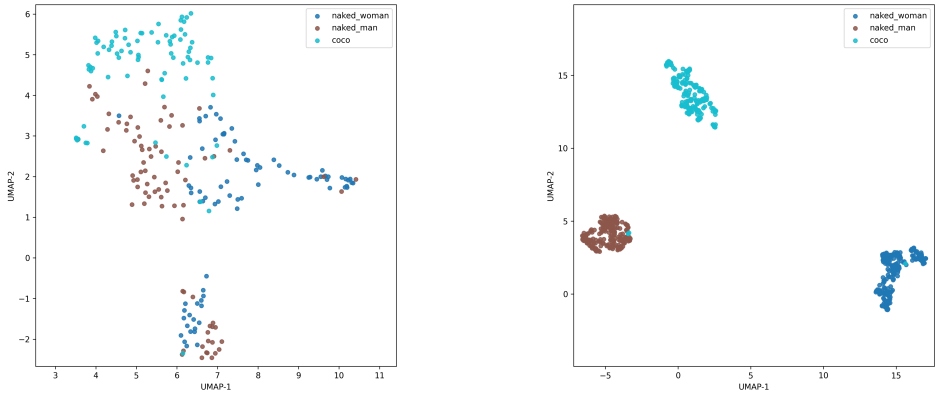
While prior works, including SAeUron, mainly treat *"nudity"* as a single target concept, we decompose it into two more fine-grained concepts, *"naked woman"* and *"naked man"*, to evaluate whether SAEParate can separate closely related real-world concepts.

After training the SAE, we collect SAE latent representations and identify target-specific feature groups using the feature importance score described in Appendix A.7.1. We use the simple prompts *"naked woman"* and *"naked man"* during this feature selection stage. We then perform a concept-wise hyperparameter search for each target concept using 20 validation prompts related to *"naked woman"* and *"naked man"*. For evaluation, we select sexually related prompts from the I2P benchmark [39] and use the NudeNet detector to classify generated images, which provides gender-specific exposed nudity detection results. We additionally report CLIPScore [20] and FID to measure the general image generation quality, using images generated from 1K COCO 2014 validation captions.

Table 18: Gender-specific nudity evaluation on I2P prompts. Unlearning and Retention Accuracy are computed relative to the base model and clipped to $[0, 100]\%$.

Method	Concept	Breasts (F)	Genitalia (F)	Breasts (M)	Genitalia (M)	UA. (↑)	RA. (↑)	CLIPScore (↑)	FID (↓)
Vanilla SAE (SAeUron)	Naked Woman	242	11	17	2	25.6%	100.0%	30.58	72.28
	Naked Man	234	8	14	1	0.0%	71.2%	30.53	72.18
SAEParate (Ours)	Naked Woman	85	4	13	1	73.8%	100.0%	30.70	70.12
	Naked Man	250	16	0	0	100.0%	78.2%	30.70	69.98
SD v1.4 (Base)	–	316	24	11	3	–	–	31.06	71.63

As shown in Table 18, SAEParate achieves high unlearning accuracy for each target concept while maintaining or improving retention of the non-target concept, e.g., preserving *"naked woman"* when unlearning *"naked man"*, compared to SAeUron. Interestingly, Figure 11 shows that SAEParate learns a concept-discriminative SAE latent space that separates fine-grained nudity concepts from each other as well as from normal COCO captions. These results demonstrate that our method can robustly separate desired concepts across diverse scenarios, suggesting a promising extension to label-available downstream unlearning tasks.



Vanilla SAE (SAeUron)

Cluster Rate (↑): 0.96, Centroid Margin (↓): 0.60

SAEParate (Ours)

Cluster Rate (↑): **1.00**, Centroid Margin (↓): **0.27**

Figure 11: UMAP visualization of sparse SAE latent representations for nudity unlearning: Vanilla SAE (SAeUron) (left) and SAEParate (right).

B.6 Fine-Grained Unlearning with Binary Forget/Retain Supervision

SAEParate requires supervision to define positive and negative relationships during training, but it does not require a fine-grained taxonomy with a separate label for every retained subclass. To evaluate this reduced-label setting directly, we consider a binary forget/retain formulation for fine-grained dog-breed unlearning. Golden Retriever samples form the *forget* group, while 13 non-target dog breeds are jointly assigned to a single *retain* group, without individual breed labels. This setting is particularly challenging because the target is a subordinate category of the broader *dog* concept, requiring breed-specific evidence to be suppressed while shared canine features and other dog breeds are preserved.

In the standard multi-class setting, positive pairs are defined as

$$\mathcal{P}(i) = \{j \neq i \mid y_j = y_i\}.$$

Under binary forget/retain supervision, y_i instead indicates whether a sample belongs to the forget or retain group. Positive pairs are constructed within each binary group, while negative pairs are formed across the two groups. Therefore, the same contrastive formulation can be applied without requiring a separate label for every retained subclass.

Table 19: Fine-grained Golden Retriever unlearning using only binary forget/retain supervision. All 13 non-target dog breeds are assigned to a single retain group.

Method	UA (↑)	RA (↑)	Dog FID (↓)	COCO CLIPScore (↑)
Vanilla SAE (SAeUron)	75.0%	33.1%	214.15	**28.56**
SAEParate	**100.0%**	**77.2%**	**112.34**	28.39

As shown in Table 19, SAEParate achieves substantially stronger Golden Retriever unlearning while preserving the 13 non-target dog breeds more effectively than SAeUron. It also reduces Dog FID from 214.15 to 112.34, while maintaining a comparable COCO CLIPScore, indicating that the improvement does not come from a substantial degradation in general generation quality.

Comparison with TRUST. We additionally compare SAEParate with TRUST [32], which also aims to reduce interference with semantically related retained concepts but does so through selective fine-tuning of concept-relevant cross-attention parameters. We evaluate both methods on the same Golden Retriever forget/retain split.

Table 20: Comparison with TRUST on fine-grained Golden Retriever unlearning. Both methods are evaluated using the same forget/retain split.

Method	UA (↑)	RA (↑)
TRUST	**100.0%**	68.8%
SAEParate	**100.0%**	**77.2%**

As shown in Table 20, both methods achieve 100.0% UA, while SAEParate achieves higher retention accuracy on the 13 non-target dog breeds (77.2% vs. 68.8%).

Together, these results show that SAEParate does not require a fine-grained taxonomy over all retained concepts. With only binary forget/retain supervision, it achieves a strong unlearning–retention trade-off against SAeUron and preserves closely related non-target breeds more effectively than TRUST at the same unlearning accuracy. Thus, coarse group-level supervision can be sufficient when the objective is to distinguish a target concept from a broader retained group.

B.7 Generalization to Unseen Concepts

For measuring the generalization ability of our method, we train the SAE using the first 25 style themes from the list (out of 50 total themes), following the setup of SAeUron Appendix F. All training configurations are kept identical to the style setup described in Appendix A.1. For unlearning, we set $\gamma_c = -1.0$ for all concepts, while keeping the remaining unlearning procedure identical to our main method. Table 21 compares our method with SAeUron under three evaluation setups. *In-distribution* denotes evaluation restricted to the 25 style themes used for SAE training, while *out-of-distribution* (OOD) denotes evaluation restricted to the remaining 25 unseen style themes. *All data* reports performance on the combined set of seen and unseen style themes. We report UA, IRA, and CRA for each setup to assess whether the learned SAE features generalize beyond the concepts observed during training.

Table 21: Generalization performance when the SAE is trained on half of the style themes.

Method	**Setup**	**UA (↑)**	**IRA (↑)**	**CRA (↑)**	**Avg. (↑)**
Vanilla SAE (SAeUron)	All data	75.00%	90.18%	80.74%	81.97%
	In-distribution	99.76%	99.23%	98.48%	99.16%
	Out of distribution	51.00%	67.38%	98.36%	72.25%
SAEParate (Ours)	All data	84.80%	83.42%	97.40%	88.54%
	In-distribution	97.00%	98.35%	96.40%	97.25%
	Out of distribution	72.60%	68.49%	98.40%	79.83%

Notably, SAEParate substantially improves OOD unlearning accuracy (72.60% vs. 51.00%). On the full set, SAEParate also achieves much higher CRA (97.40% vs. 80.74%), indicating that unlearning a style causes less collateral damage to images of other styles, including those unseen during SAE training. This comes at the cost of lower in-domain retention on the full set (83.42% vs. 90.18%) and slightly lower in-distribution performance. We attribute the OOD and CRA gains to the concept-separated latent space, which reduces feature sharing across styles.

B.8 Object Sequential Unlearning

We conduct an object sequential unlearning task, which assumes multiple concept-level unlearning requests and better reflects real-world deployment scenarios. Since UnlearnCanvas provides only a style sequential unlearning setting, we extend the evaluation to object concepts, where Vanilla SAE (SAeUron) tends to struggle more with balancing unlearning and retention.

We sequentially unlearn the following object concepts:

- $\mathcal{T}_1$: Architectures
- $\mathcal{T}_2$: Bears
- $\mathcal{T}_3$: Birds
- $\mathcal{T}_4$: Butterfly
- $\mathcal{T}_5$: Cats

At each step, we add a new target concept to the set of unlearning requests and evaluate performance using UA, IRA, and CRA. We define RA as the average of IRA and CRA, i.e., $\mathrm{RA} = (\mathrm{IRA} + \mathrm{CRA})/2$.

As shown in Table 22, SAEParate achieves a better balance between sequential unlearning and retention compared to SAeUron. In particular, SAEParate avoids the severe retention collapse observed in SAeUron when the number of unlearned object concepts increases.

Table 22: Object sequential unlearning results. Each column denotes a new unlearning request, where $\mathcal{T}_1$ is the oldest request. Each UA row reports the unlearning accuracy for a specific target object. RA is computed as the average of IRA and CRA, averaged over the active requests in each column. Results indicating catastrophic retaining failure are highlighted in red.

Method: Vanilla SAE (SAeUron)

Metrics		$\mathcal{T}_1$	$\mathcal{T}_1 \sim \mathcal{T}_2$	$\mathcal{T}_1 \sim \mathcal{T}_3$	$\mathcal{T}_1 \sim \mathcal{T}_4$	$\mathcal{T}_1 \sim \mathcal{T}_5$
		Unlearning Request				
UA	$\mathcal{T}_1$	33.33%	35.29%	84.31%	100.00%	100.00%
	$\mathcal{T}_2$	-	65.69%	92.16%	100.00%	100.00%
	$\mathcal{T}_3$	-	-	91.50%	100.00%	100.00%
	$\mathcal{T}_4$	-	-	-	93.63%	97.55%
	$\mathcal{T}_5$	-	-	-	-	98.04%
RA		97.91%	93.27%	87.32%	18.40%	4.96%

Method: SAEParate

Metrics		$\mathcal{T}_1$	$\mathcal{T}_1 \sim \mathcal{T}_2$	$\mathcal{T}_1 \sim \mathcal{T}_3$	$\mathcal{T}_1 \sim \mathcal{T}_4$	$\mathcal{T}_1 \sim \mathcal{T}_5$
		Unlearning Request				
UA	$\mathcal{T}_1$	96.08%	70.59%	92.16%	68.63%	98.04%
	$\mathcal{T}_2$	-	85.29%	92.16%	84.31%	89.22%
	$\mathcal{T}_3$	-	-	88.24%	80.39%	86.93%
	$\mathcal{T}_4$	-	-	-	70.59%	66.67%
	$\mathcal{T}_5$	-	-	-	-	73.33%
RA		97.11%	93.94%	92.51%	88.83%	87.67%

We also provide qualitative results in Figures 13 and 14. SAEParate robustly suppresses multiple target concepts while largely preserving non-target concepts, whereas Vanilla SAE fails to maintain retention after several sequential unlearning requests. These results further suggest that SAEParate reduces redundant feature selection by directly separating latent representations, allowing target concepts to be isolated in more concept-specific features.

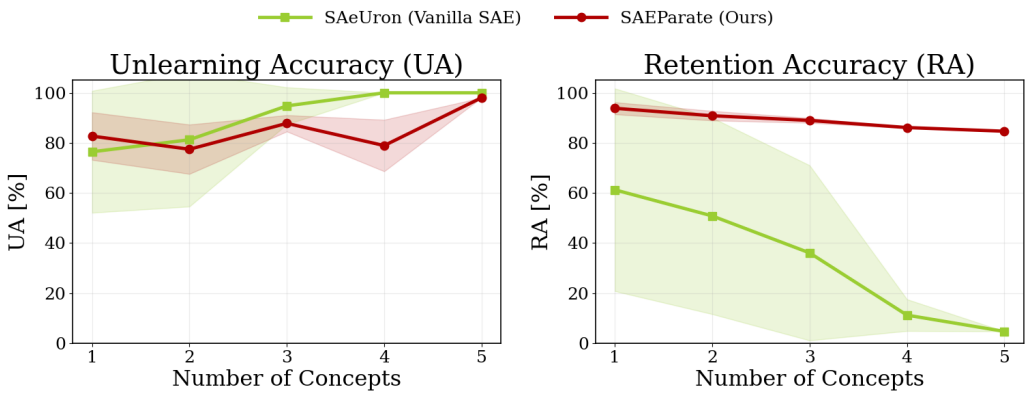


Figure 12: Sequential object unlearning results across increasing numbers of target concepts. SAEParate preserves retention accuracy substantially better than Vanilla SAE (SAeUron) under multiple unlearning requests.

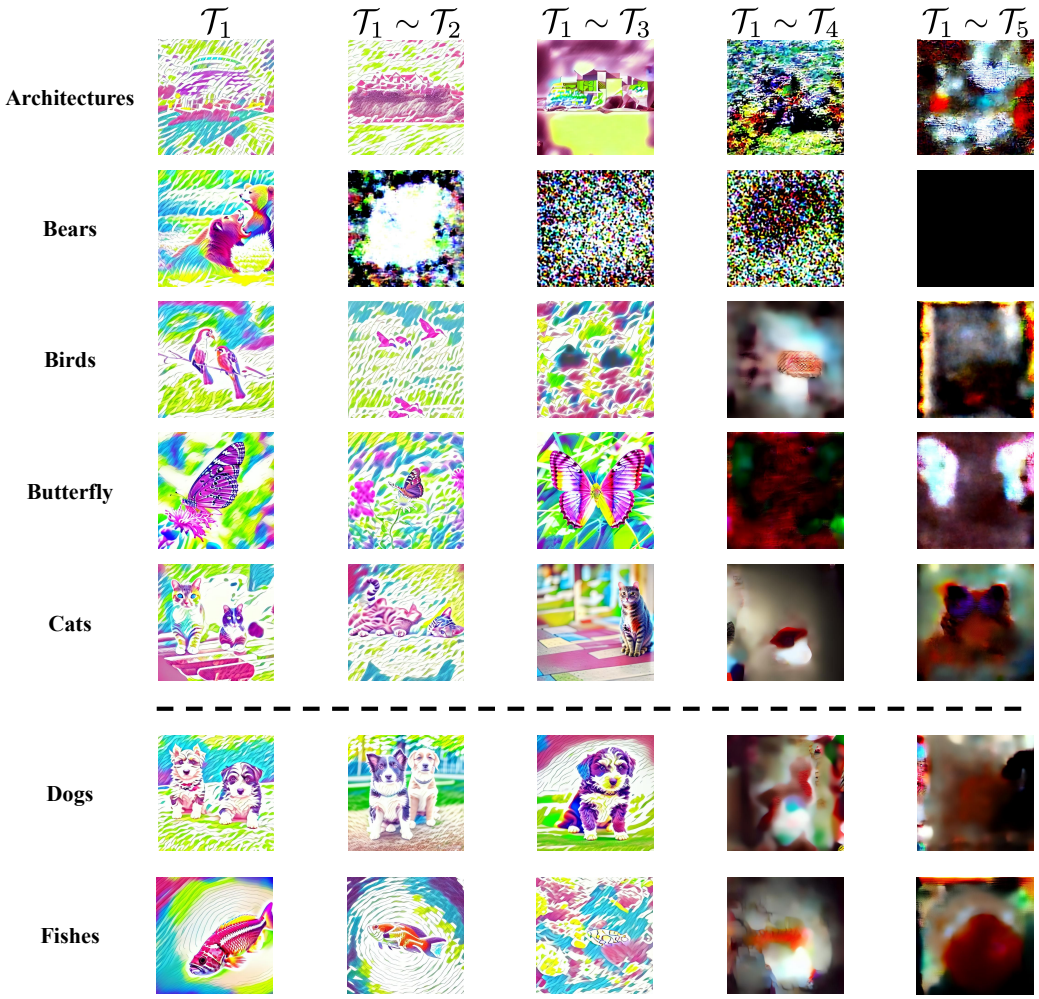


Figure 13: Qualitative visualization of object sequential unlearning for Vanilla SAE (SAeUron). The first five rows correspond to the target object classes $\mathcal{T}_1$ through $\mathcal{T}_5$, while the columns show cumulative unlearning requests. The "Dogs" and "Fishes" rows are non-target object classes and are used to evaluate retention.

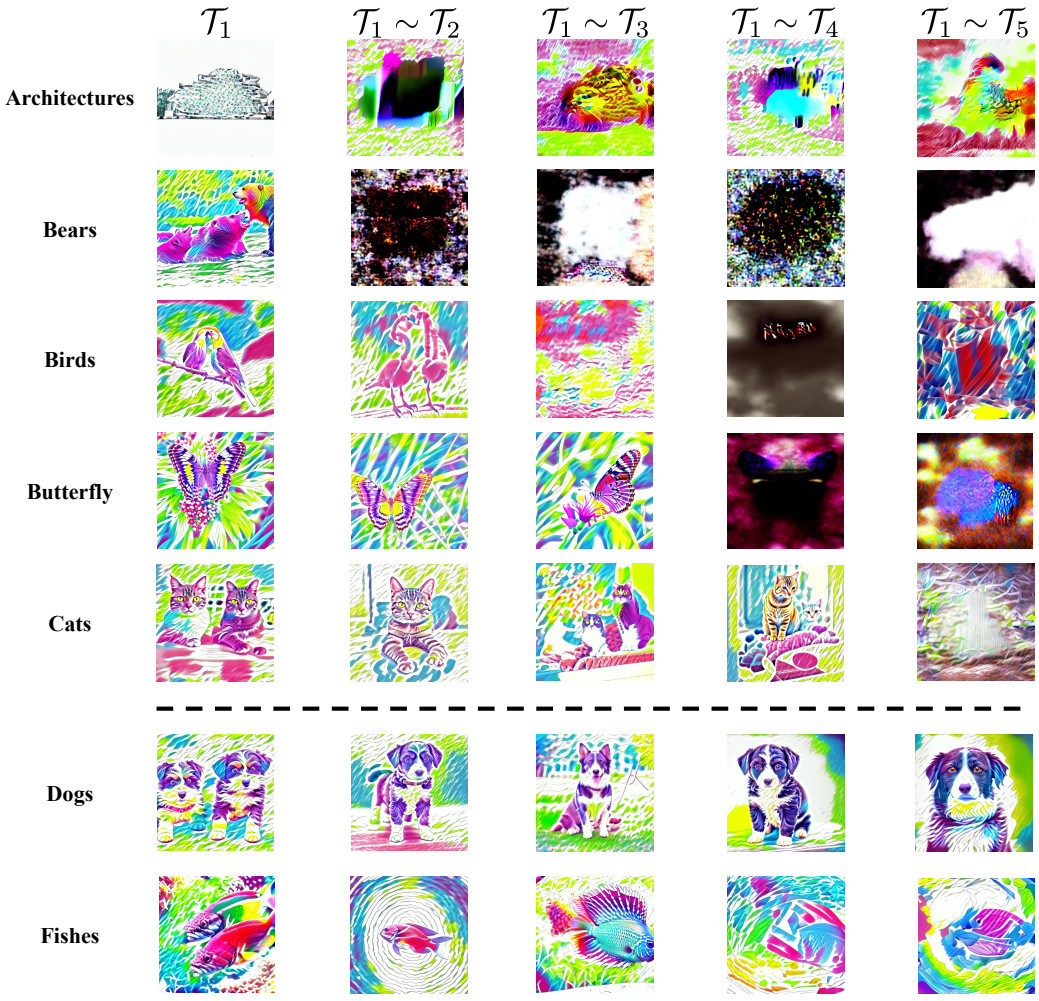


Figure 14: Qualitative visualization of object sequential unlearning for SAEParate (Ours). The first five rows correspond to the target object classes $\mathcal{T}_1$ through $\mathcal{T}_5$, while the columns show cumulative unlearning requests. The "Dogs" and "Fishes" rows are non-target object classes and are used to evaluate retention.

B.9 Compatibility with Alternative SAE Training Objectives

We further examine whether the concept-separation objective of SAEParate can be combined with alternative SAE training schemes. Specifically, we incorporate the fixed-prefix, multi-scale training objective of Matryoshka SAE [4] into the SAEParate pipeline and evaluate the resulting model under the subset setting (Appendix A.2) in the object-unlearning setting.

Table 23: Compatibility of SAEParate with the Matryoshka SAE training objective. All results are obtained under the subset setting (Appendix A.2).

Method	UA (↑)	IRA (↑)	CRA (↑)	Avg. (↑)	Cluster Rate (↑)	Centroid Margin (↓)
Vanilla SAE (SAeUron)	58.14%	96.74%	**97.45%**	84.11%	0.93	0.73
Matryoshka SAE	81.18%	95.61%	91.06%	89.28%	0.94	0.30
SAEParate + Matryoshka	**93.14%**	**97.43%**	94.39%	**94.99%**	**0.99**	**0.02**

Combining SAEParate with Matryoshka SAE substantially improves both unlearning performance and latent-space separation over Matryoshka SAE alone, increasing the average accuracy from 89.28% to 94.99% while improving the Cluster Rate from 0.94 to 0.99 and reducing the Centroid Margin. These results suggest that the concept-separation objective of SAEParate is complementary to alternative SAE training objectives rather than being tied to a specific SAE formulation. More broadly, this compatibility indicates that SAEParate can benefit from advances in SAE design, providing a potential avenue for alleviating SAE-specific limitations such as feature splitting and absorption without changing the core concept-separation framework.

B.10 Preliminary Extension to an MM-DiT Backbone

Extending SAEParate beyond the U-Net architecture used in our main experiments to MM-DiT-based diffusion models requires several architecture-specific considerations. In particular, MM-DiT jointly updates image and text tokens across transformer blocks, so target-concept information retained in the text-token stream may be reintroduced into the image stream after an intervention. Moreover, standalone cross-attention is replaced by joint attention over image and text tokens, requiring the cross-modal relevance signal used by SAEParate to be reformulated.

One possible adaptation is to intervene on the post-block image-token residual stream, which preserves image-token structure while incorporating both attention and feed-forward updates. Likewise, the image-query-to-text-key portion of joint attention can serve as an analogue of the cross-attention signal used in our U-Net implementation for estimating concept relevance. The remaining components of SAEParate, including contrastive representation learning, feature calibration and suppression, and the nonlinearity-enhanced encoder, are not inherently tied to U-Net-specific operators. A systematic treatment of these architecture-specific design choices is beyond the scope of this work and remains an important direction for future study. Nevertheless, we conduct a preliminary experiment to examine whether a simplified adaptation can retain the benefits of SAEParate on an MM-DiT backbone.

Preliminary Experiments on SD3.5-Medium-Turbo. We evaluate this simplified adaptation on Stable Diffusion 3.5 Medium Turbo, released by TensorArt Studios. We insert the SAE into the image-token stream of MM-DiT block 8 out of 24. The insertion depth is selected using a 20-way class-separability probe across blocks; separability remains relatively stable across early and middle blocks and begins to degrade beyond approximately block 14.

Table 24: Preliminary object-unlearning results on the MM-DiT-based Stable Diffusion 3.5 Medium Turbo backbone. RA denotes object-level retention accuracy, and Cluster Rate measures class-wise separation in the learned SAE latent space.

Method	UA (↑)	RA (↑)	Avg. (↑)	Cluster Rate (↑)
Vanilla SAE (SAeUron)	43.3%	**70.8%**	57.1%	0.435
SAEParate	**71.7%**	60.3%	**66.0%**	**0.498**

Unlike UnlearnCanvas, this backbone does not provide a model specifically fine-tuned for the predefined object and style concepts. We therefore restrict retention evaluation to object-level RA. The absolute UA and RA values should also be interpreted with caution because the UnlearnCanvas ViT classifiers were fine-tuned on UnlearnCanvas images, which may differ in distribution from SD3.5 Medium Turbo outputs. Nevertheless, the relative comparison remains informative because both methods are evaluated using the same backbone, prompts, generation settings, and classifiers.

As shown in Table 24, the simplified SAEParate adaptation improves UA from 43.3% to 71.7% and the overall average from 57.1% to 66.0%, while also increasing the Cluster Rate from 0.435 to 0.498. These results demonstrate that the concept-separation objective can also improve unlearning performance on an MM-DiT-based backbone, despite differences in architecture and evaluation settings.

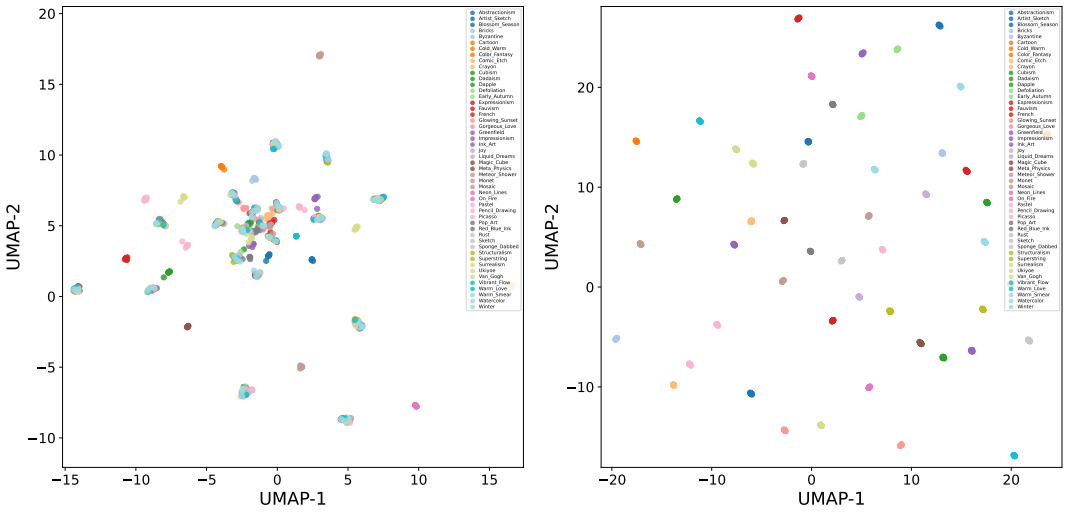
C Additional Visualizations

C.1 Additional Latent Space Visualizations

To further analyze the structure of the learned latent space, we visualize sparse latent representations using UMAP [33] and measure their concept-wise clustering behavior. The analysis is conducted on the validation set used for computing feature importance scores, which contains 20 prompts per target style, 80 prompts per target object and per target style–object combination for the joint setting. In each UMAP visualization, different colors denote different concepts, following the class and theme categories defined in UnlearnCanvas.

Using the same visualization protocol as the object-level analysis in Figure 2, we first provide the style-level latent visualization in Figure 15. We then visualize the latent space of SAEs trained for joint style–object concepts in Figure 16. Finally, we additionally report the latent space visualizations of the *only*-LSC and *only*-GeLU variants discussed in the ablation study in Section 5.2 with Figure 17.

Across these visualizations, Vanilla SAE exhibits noticeably entangled sparse representations, whereas SAEParate learns a more concept-discriminative latent space with clearer class-wise separation. This suggests that SAEParate reduces co-activation of shared latent features across concepts, thereby mitigating interference during the unlearning process.



Vanilla SAE (SAeUron)

Cluster Rate (↑): 0.98, Centroid Margin (↓): 0.76

SAEParate (Ours)

Cluster Rate (↑): **1.00**, Centroid Margin (↓): **0.08**

Figure 15: UMAP visualization of sparse SAE latent representations for style unlearning: Vanilla SAE (SAeUron) (left) and SAEParate (right). Each color denotes a different style concept from UnlearnCanvas.

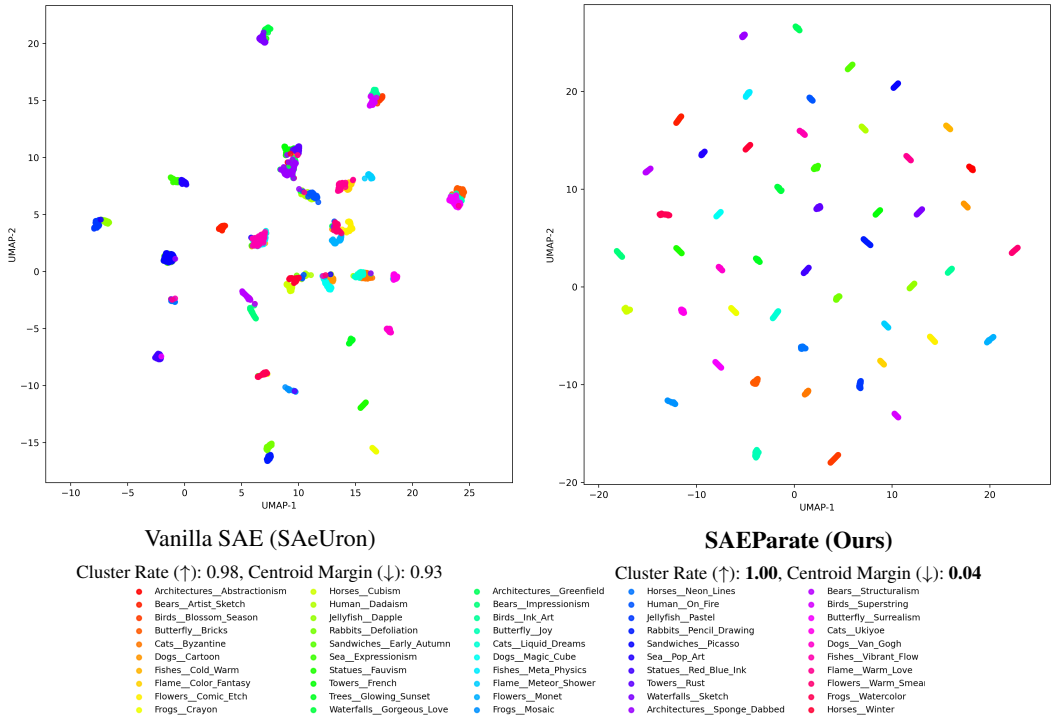
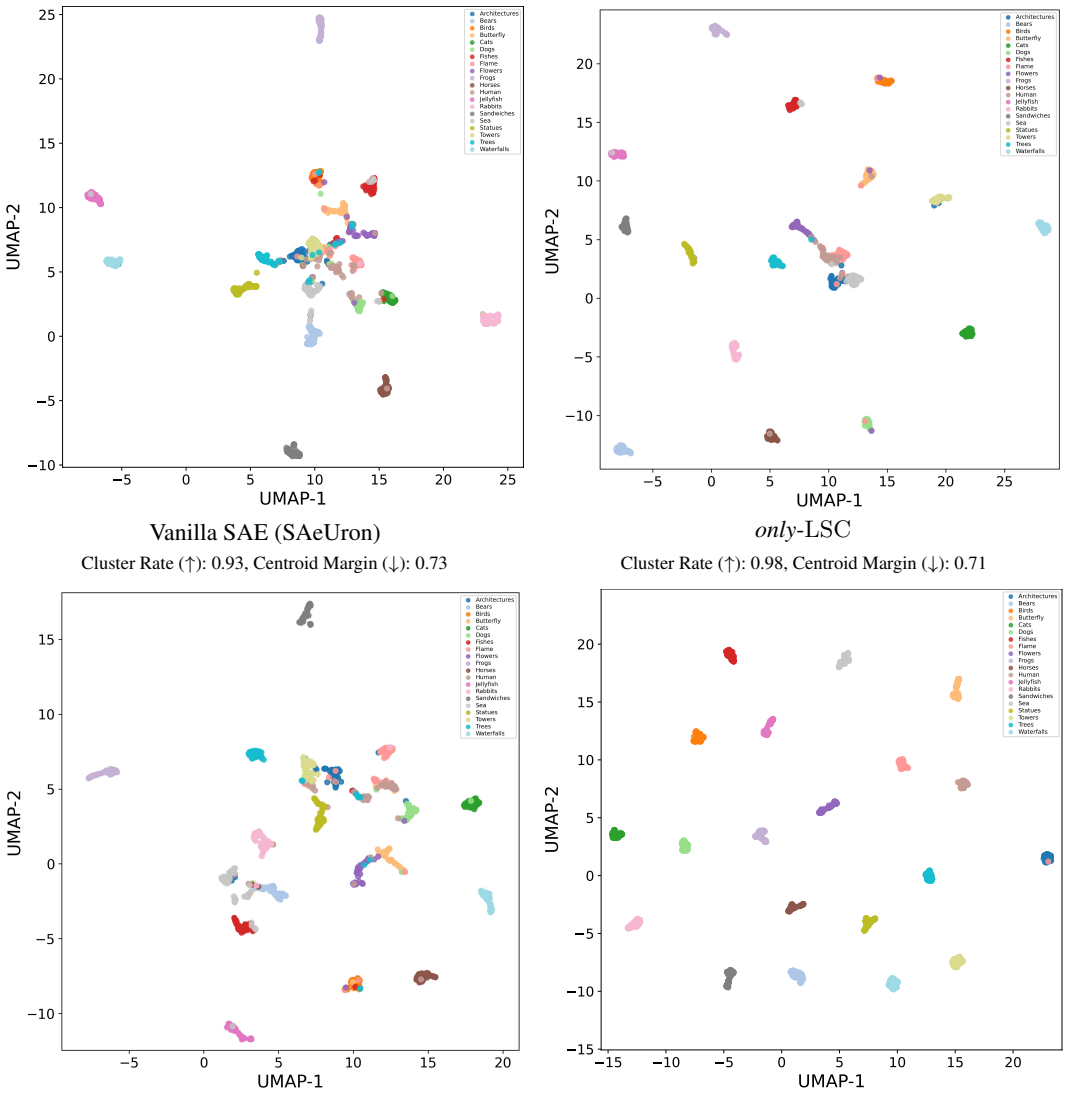


Figure 16: UMAP visualization of sparse SAE latent representations for style–object joint unlearning: Vanilla SAE (SAeUron) (left), SAEParate (right). Each color denotes a different style–object concept from UnlearnCanvas.



Vanilla SAE (SAeUron)

Cluster Rate (↑): 0.93, Centroid Margin (↓): 0.73

only-LSC

Cluster Rate (↑): 0.98, Centroid Margin (↓): 0.71

only-GeLU

Cluster Rate (↑): 0.94, Centroid Margin (↓): 0.64

SAEParate (Ours)

Cluster Rate (↑): **1.00**, Centroid Margin (↓): **0.06**

Figure 17: UMAP visualization of sparse SAE latent representations for Vanilla SAE (SAeUron) (top-left), *only*-LSC (top-right), *only*-GeLU (bottom-left), and SAEParate (Ours) (bottom-right). Each color denotes a different concept class from UnlearnCanvas.

C.2 Detailed Performance Heatmaps

For measuring the class-wise impact of unlearning, we report classification-accuracy heatmaps over all UnlearnCanvas concepts, including the target concept itself. We provide object-unlearning heatmaps for SAeUron and SAEParate in Figures 18 and 19, respectively, and style-unlearning heatmaps in Figures 20 and 21. As shown in these heatmaps, SAEParate achieves near-perfect unlearning accuracy along the target-concept diagonal while preserving high accuracy for non-target concepts.

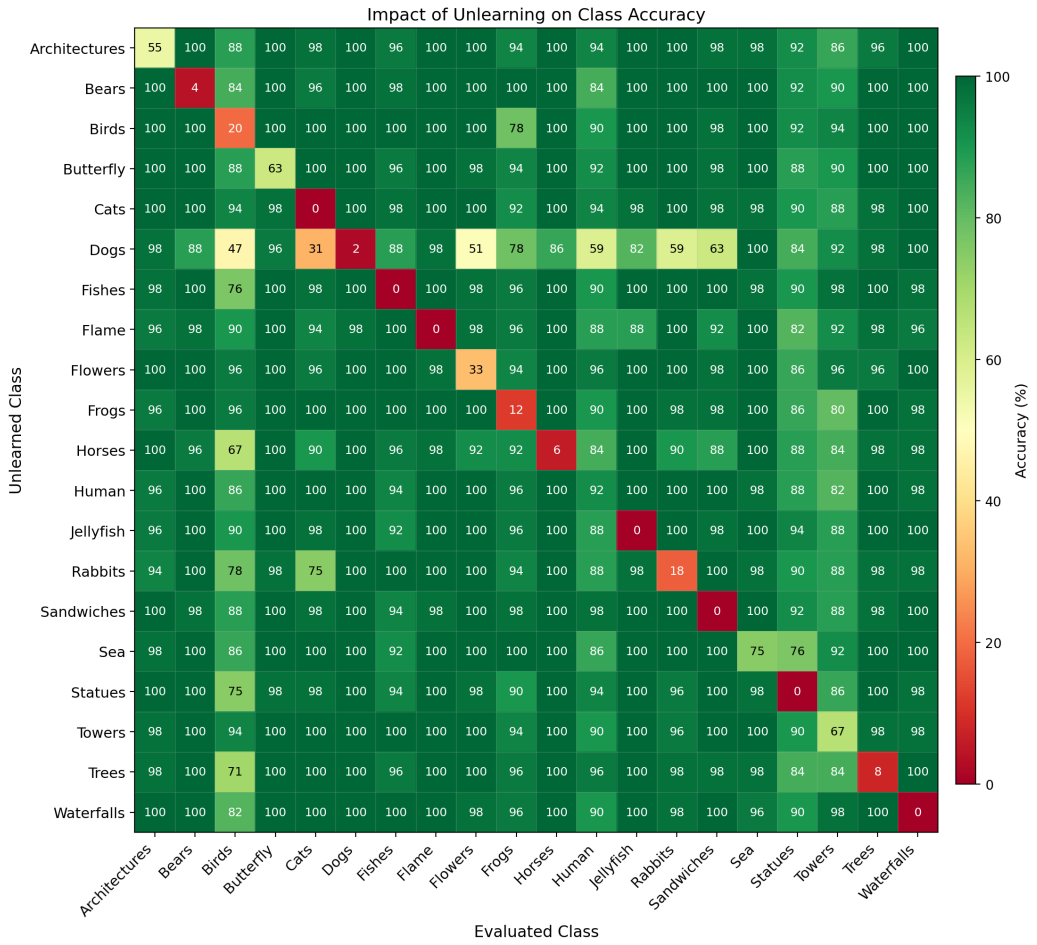


Figure 18: Detailed performance heatmap of SAeUron in object unlearning.

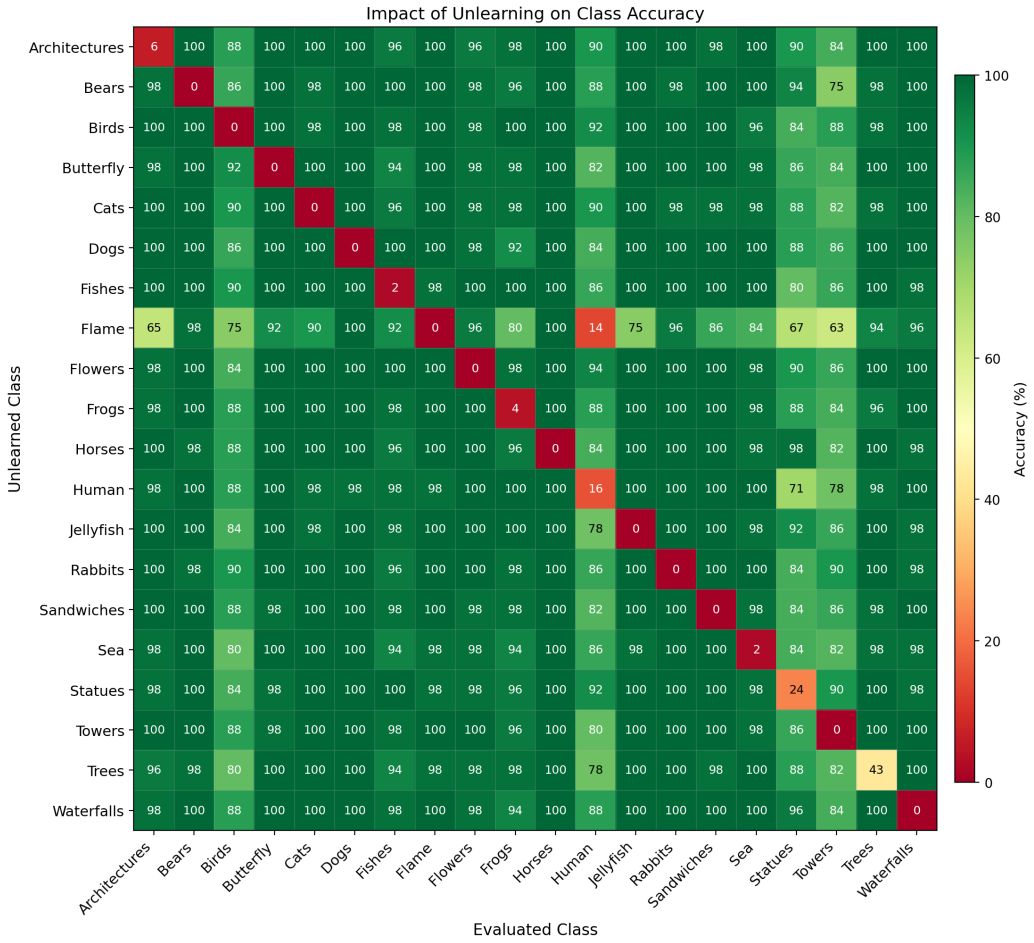


Figure 19: Detailed performance heatmap of SAEParate in object unlearning.

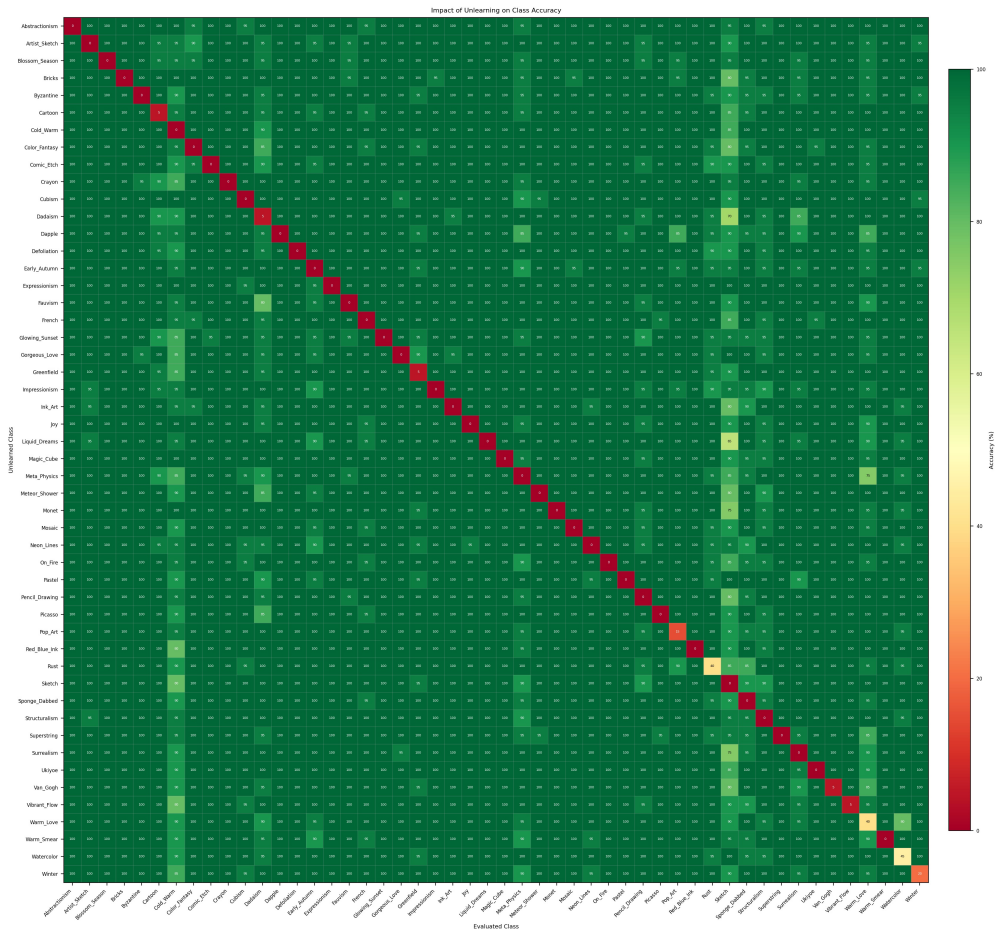


Figure 20: Detailed performance heatmap of SAeUron in style unlearning.

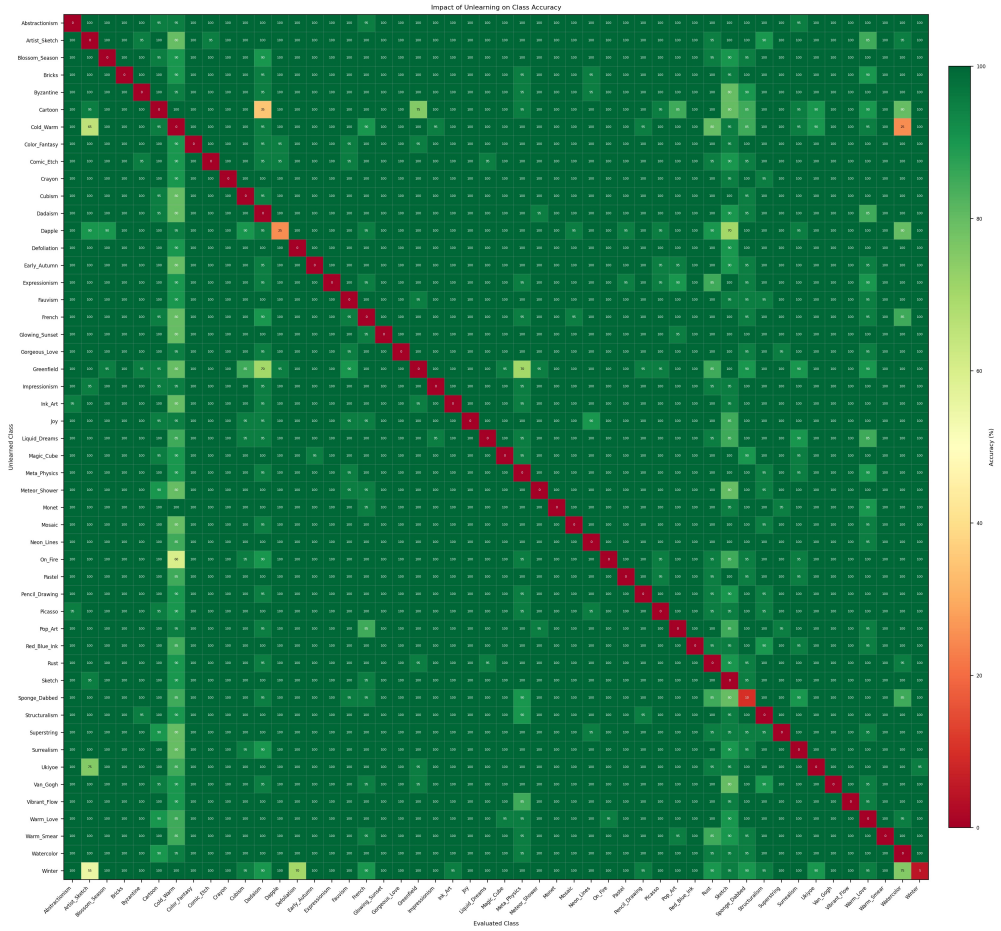


Figure 21: Detailed performance heatmap of SAEParate in style unlearning.

C.3 Qualitative Comparisons

We provide qualitative comparisons for object, style, and joint unlearning scenarios, including images generated by the base model, SAeUron, and SAEParate. The results are shown in Figures 22, 23, 24, and 25. Notably, SAEParate preserves non-target concepts more faithfully than SAeUron, with particularly strong retention in the joint unlearning examples shown in Figures 24 and 25.

D Broader Impacts

This work aims to advance machine learning by improving selective concept unlearning in text-to-image diffusion models. While our method is designed to remove unwanted, biased, or harmful content while preserving non-target concepts, it is inherently dual-use and could be misused for censorship, unfair content suppression, or undesirable manipulation of model behavior. Therefore, real-world deployment should require transparent target definitions, careful evaluation, and appropriate human oversight.

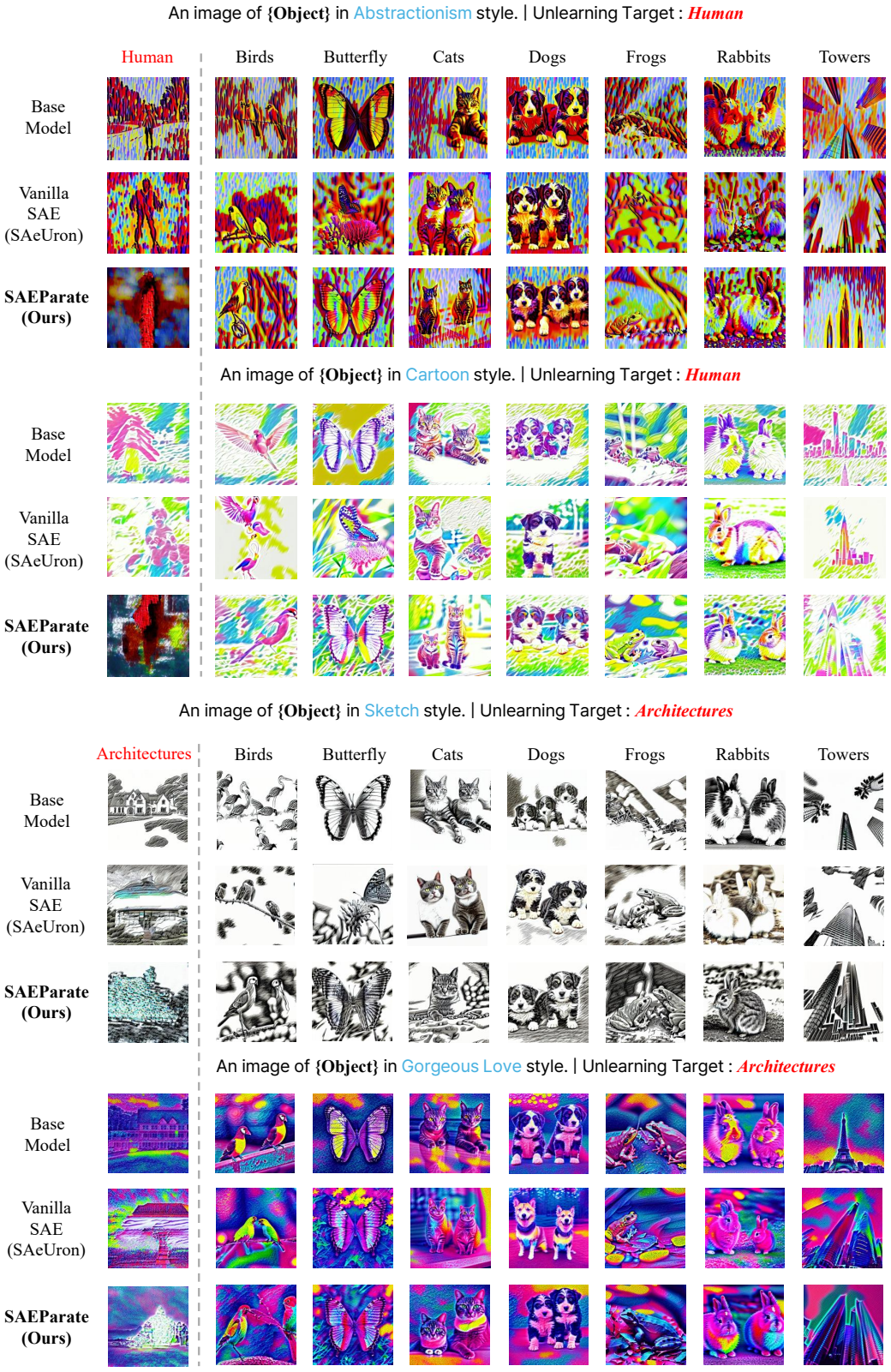


Figure 22: Qualitative comparison with Vanilla SAE (SAeUron) on object unlearning.

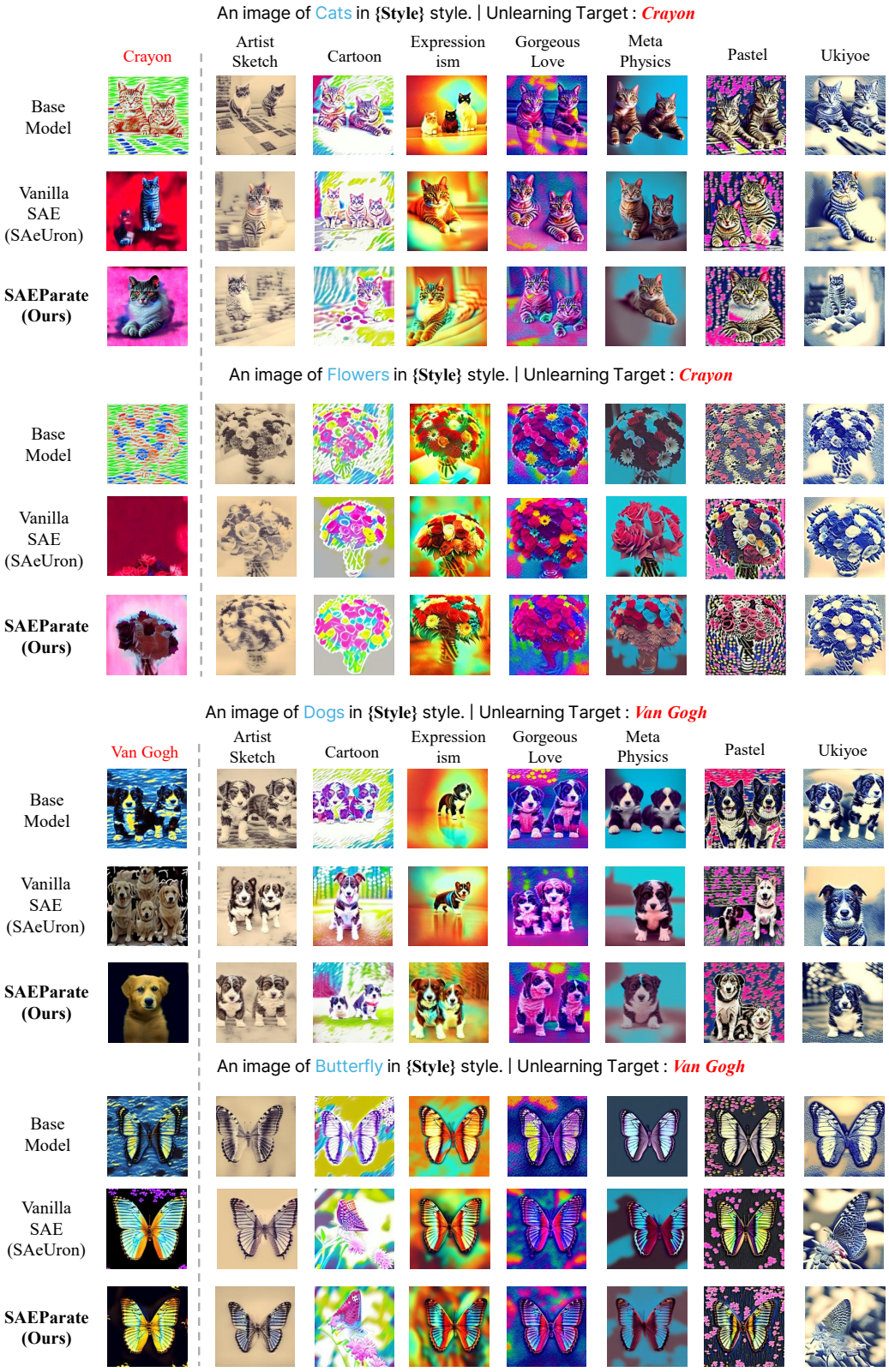


Figure 23: Qualitative comparison with Vanilla SAE (SAeUron) on style unlearning.

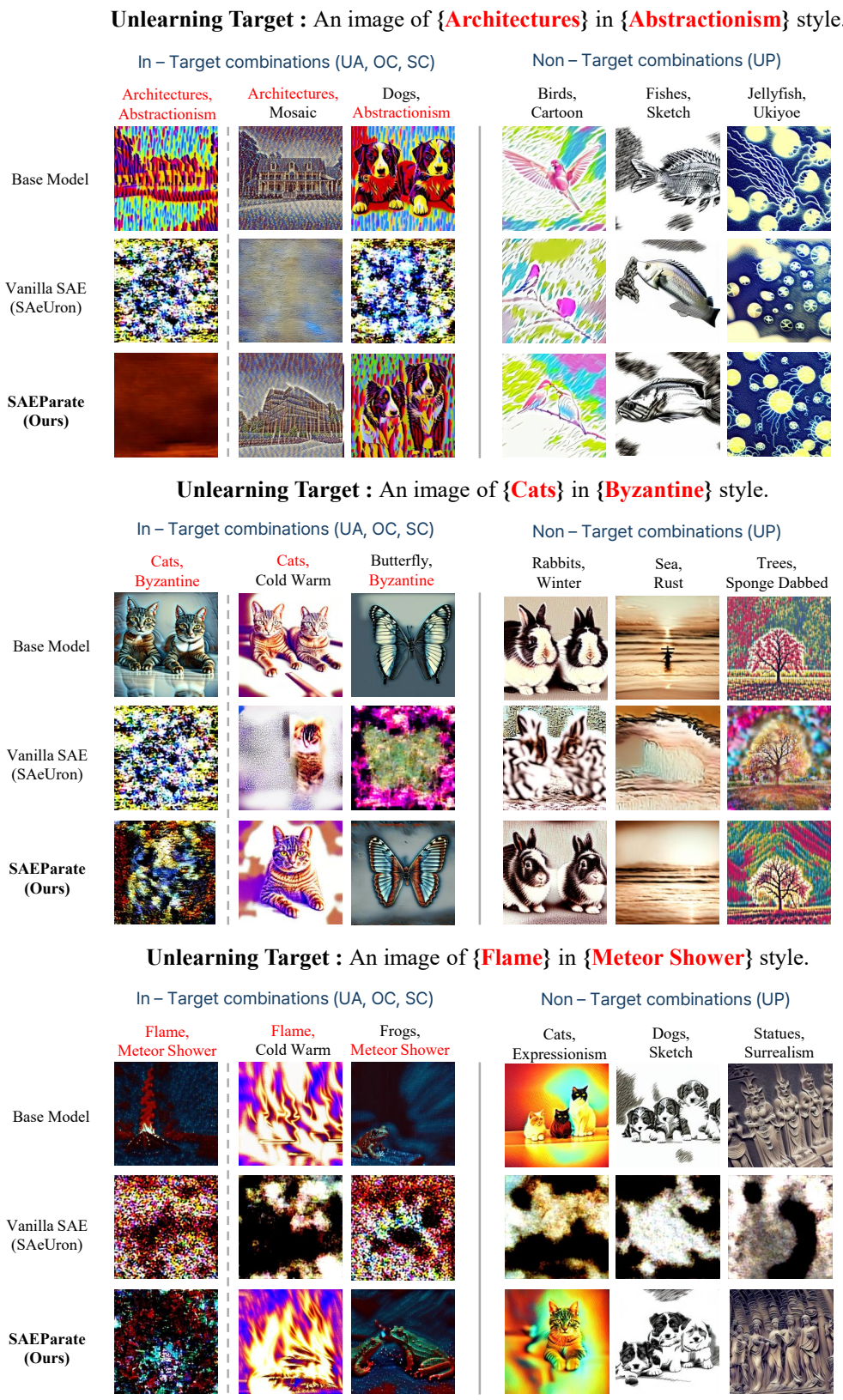


Figure 24: Qualitative comparison with Vanilla SAE (SAeUron) on style–object joint unlearning.



Figure 25: Qualitative comparison with Vanilla SAE (SAeUron) on style–object joint unlearning.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction make three main claims: (1) SAE latent spaces exhibit concept entanglement, (2) concept-aware contrastive learning separates SAE latent activation patterns, and (3) our method improves performance on the UnlearnCanvas benchmark, particularly in joint unlearning scenarios. These claims are supported in Section 3, Section 4, and Section 5, respectively.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work, including the need for concept labels and evaluation limited to image generation, along with future directions in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper is empirical in nature and does not present theoretical results or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide detailed descriptions of our method in Sections 4.1–4.3, the full set of SAE training hyperparameters and configurations in Appendix A.1 and Table 4, the joint unlearning formulation in Appendix A.8, the inference-time suppression procedure in Appendix A.7, and per-target hyperparameter values in Appendix A.9. We also provide implementation code for reproducing the main experiments. We use the publicly available UnlearnCanvas benchmark and build on the SAeUron training pipeline, with all modifications described in the paper.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide source code for the main training, unlearning, and evaluation procedures as a supplementary zip archive, including scripts and configuration files needed to reproduce the main experimental results. The code includes a README with setup instructions, environment requirements, and example commands for running each experiment. We use the publicly available UnlearnCanvas benchmark and follow the same checkpoint setup as SAeUron [10], with download instructions provided in the README.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide training and evaluation details in Section 5, key hyperparameters in Appendix A.1 and Table 4, and per-concept hyperparameters with the grid search procedure in Appendix A.9 and Appendix A.7.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Following the standard evaluation protocol of the UnlearnCanvas benchmark and prior unlearning baselines (e.g., SAeUron, SalUn, ESD), we report deterministic evaluation results without error bars. Running multiple seeds across all unlearning targets and baselines would be prohibitively expensive given the cost of large-scale image generation, but the consistency of our gains across diverse target concepts supports the reliability of our findings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: SAE training was conducted on a single NVIDIA B200 GPU, with training times of approximately 46, 20, and 69 hours for the object, style, and joint blocks, respectively, as reported in Appendix A.1 (Table 4). Memory and storage usage are compared against existing baselines in Table 2 (main paper) and Appendix A.6 (Table 8).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our research conforms to it. Our work focuses on machine unlearning to suppress undesirable concepts in text-to-image diffusion models, contributing to the responsible deployment of generative models, and does not involve human subjects or sensitive personal data.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impact of our work in Appendix D, including both its potential benefits for safer concept unlearning and risks such as misuse, censorship, and unfair content suppression.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We release code and model checkpoints for research purposes. To mitigate potential misuse, we rely on publicly available datasets and pretrained models that follow existing safety guidelines. We encourage responsible use of our assets and adhere to the usage policies of the original models and datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available datasets, models, and code, and properly cite their original sources, including UnlearnCanvas [44], Stable Diffusion [38], and SAeUron [10]. All assets are used in accordance with their respective licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not introduce new datasets, benchmarks, software packages, or other assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research does not involve crowdsourcing or human subjects. All evaluations are performed using automated metrics and pretrained classifiers, so no participant instructions or compensation are applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This research does not involve human subjects or crowdsourcing. All experiments are conducted on benchmark datasets and evaluated using automated metrics and pretrained classifiers.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used in the methodology. SAEParate is based on sparse autoencoders applied to diffusion model features and does not involve any LLM components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.