

000 TRACE: TRANSCODER-BASED CONCEPT EDITING

001
002
003 **Anonymous authors**

004 Paper under double-blind review

005 006 ABSTRACT

007 Image generation with diffusion and autoregressive models can inadvertently output
 008 undesirable content, such as copyrighted characters, harmful images, unwanted
 009 objects, or protected artistic styles. Therefore, trustworthy content moderation
 010 remains a major challenge: retraining for the removal of each of these concepts is
 011 infeasible, while existing post-hoc interventions are either easy to bypass or come
 012 at the cost of image quality. We introduce a white-box, model-agnostic framework
 013 that uses *Transcoders* as an integrated, surgical intervention layer that allows
 014 precise, in-place suppression of targeted concepts without retraining the generative
 015 model. Because our approach modifies the model’s backbone and not just external
 016 modules, it is robust against circumvention and preserves overall generation quality.
 017 Empirically, our method achieves new state-of-the-art results for both diffusion and
 018 autoregressive image generative models, remaining robust even against adversarial
 019 prompts and throughout sequential, diverse concept removal requests. Thereby,
 020 our approach sets the practical foundation for trustworthy image generation in
 021 real-world scenarios.

022 1 INTRODUCTION

023 Image generative models, such as diffusion models (DMs) (Esser et al., 2024b; StabilityAI, 2023;
 024 Black Forest Labs, 2024) and image autoregressive models (IARs) (Han et al., 2025; Tang et al.,
 025 2024), have revolutionized the creation of realistic, detailed, and aesthetic content. Despite their
 026 capabilities, these models often raise concerns, as they can inadvertently generate undesirable content,
 027 such as copyrighted characters (Gary Marcus, 2024), Not Safe For Work (NSFW) images, e.g.,
 028 depictions of violence or nudity (Qu et al., 2023; Rando et al., 2022; Yang et al., 2024b), unwanted
 029 objects (Wei et al., 2025), or copyrighted artistic styles (Aspertì et al., 2025).

030 Numerous works have addressed the challenge of preventing models from generating these con-
 031 cepts. Overall, there are two types of approaches. The first one adds additional safety modules
 032 to the model (Poppi et al., 2024; Cywiński & Deja, 2025) or the generation pipeline during the
 033 inference (Schramowski et al., 2023; Li et al., 2024) but leaves the original model unaltered. Since
 034 such approaches can be easily bypassed or removed in local deployments, they are limited to API-
 035 based deployments. The second type alters the model in-place, either through training or through
 036 direct closed-form model edits, which makes them more robust to removal. Training-based ap-
 037 proaches (Zhang et al., 2025; Gandikota et al., 2023; Gao et al., 2025; Zhang et al., 2024a; Kumari
 038 et al., 2023; Wu et al., 2025; Fan et al., 2024; Wu & Harandi, 2024; Heng & Soh, 2023; Lu et al.,
 039 2024) suffer from high computational costs, especially as the models grow in size. Additionally,
 040 they usually rely on objective functions tied to a specific training paradigm, which restricts their
 041 applicability across models. Closed-form edits (Orgad et al., 2023; Gandikota et al., 2024; Basu et al.,
 042 2024a; Gong et al., 2024) are computationally efficient and can be applied across models but tend to
 043 degrade generation quality, especially under sequential concept removal requests, which is limiting in
 044 real-word deployment where multiple undesired concepts can be identified over time.

045 To address these shortcoming, we introduce **TRAnsco**der-based Concept Editing (TRACE), a
 046 new framework for concept removal in text-to-image generative models. Originally developed to
 047 disentangle features in large language models (Dunefsky et al., 2024), transcoders provide sparse
 048 and interpretable representations that make it possible to isolate individual concepts, a prerequisite
 049 for targeted concept removal. In text-to-image architectures, the interface between the text encoder
 050 and the generative backbone typically passes through one or two narrow transformation layers (Esser
 051 et al., 2024a; Han et al., 2025), which act as a bottleneck. Replacing these layers with a transcoder



Figure 1: TRACE removes diverse concepts, including NSFW, Style, objects, or IP Characters.

provides a natural intervention point where unwanted features can be blocked before they propagate through the model. Concept removal is then achieved by identifying the transcoder-latents associated with a target concept and redirecting them toward a neutral representation, such as the empty token. Importantly, unlike add-on safety modules (Poppi et al., 2024; Cywiński & Deja, 2025; Schramowski et al., 2023; Li et al., 2024) that can be detached or bypassed, our method integrates the transcoder directly into the generative model’s backbone where it remains persistent. Notably, our method also eliminates the need for backpropagation through the full model as the transcoder can be trained in isolation at low computational costs. This enables our method to scale to modern, large diffusion and autoregressive architectures with billions of parameters.

We evaluate our method on state-of-the-art (SOTA) DMs, namely Stable Diffusion 3.5 Large (SD3.5) (Esser et al., 2024a) and FLUX.1-dev (Flux) (Labs, 2024), as well as on IARs, namely Infinity-2B and Infinity-8B (Han et al., 2025). We focus mainly on the tasks of **style removal** and **object removal** using the UnlearnCanvas benchmark (Zhang et al., 2024b). Our results highlight that TRACE achieves SOTA performance in concept removal across both paradigms, DMs and IARs. For example, on our largest model Flux (12B parameters) it surpasses the baselines by up to 21 percentage points in style removal accuracy, while also achieving significantly higher in- and cross-domain retain accuracy. This trend is consistent over all tested models, confirming our method’s flexibility and scalability. In Figure 1, we also provide quantitative results for Infinity-8B which demonstrate that our method effectively removes diverse concepts while maintaining high visual quality. We further show that unlike prior weight-editing methods that degrade under sequential edits, TRACE remains effective when removing concepts sequentially. Finally, TRACE maintains robust concept removal also under targeted attacks, highlighting its robustness. Together, these results establish TRACE as a practical and general solution for safe deployment of generative models.

In summary, we make the following contributions:

- We introduce TRACE, a framework for concept removal in text-to-image generative models, based on integrated transcoders, which does not require retraining the generative model.
- Our method is broadly applicable: it removes a wide range of target concepts across both SOTA diffusion and image autoregressive architectures.
- We demonstrate that our approach enables sequential removal of multiple concepts, a setting where existing methods degrade dramatically or simply fail.
- Unlike prior add-on modules, our intervention yields persistent concept removal: it is robust to both local tampering (removal/circumvention) and dedicated attacks.
- Extensive experiments show that our method maintains high visual fidelity and competitive performance across diverse benchmarks and concept types.

108 **2 RELATED WORK**

109

110 **Text-to-Image Models.** Currently, there are two dominant paradigms for text-to-image modeling,
 111 namely Diffusion Models (DM) and Image AutoRegressive (IAR) models. DM learn the gradient field
 112 between Gaussian noise and the image distribution, originally through stochastic processes (Rombach
 113 et al., 2022), with recent work favoring deterministic formulations (Esser et al., 2024a). IAR models
 114 instead factorize the joint distribution into conditional probabilities, generating tokens sequentially.
 115 While early IARs were restricted to class-conditional generation (Van Den Oord et al., 2016; Ramesh
 116 et al., 2021), recently, models like Infinity (Han et al., 2025) are text-conditional and generate images
 117 at SOTA quality based on the principle of next-scale prediction. See Appendix A.1 for more details
 118 on both paradigms. Our proposed TRACE method applies to both DMs and IARs.

119 **Concept Removal** aims to prevent a model from generating content related to a *target concept*.
 120 Approaches can be grouped into two categories: **internal** methods, which directly modify model
 121 weights or structure, and **external** methods, which leave the model frozen and operate externally, for
 122 example by adding separate, removable modules (Cywiński & Deja, 2025) or modifying generation
 123 at inference time (Schramowski et al., 2023; Li et al., 2024). Those external methods are, therefore,
 124 solely suitable for API-based deployments but can be bypassed in local settings. Internal methods,
 125 such as our proposed TRACE, in contrast, produce persistent changes and are therefore more suitable
 126 for open-source models that can be deployed locally. Existing methods can be further divided into
 127 two main families: closed-form editing and training-based editing. We present a full overview in
 128 Table 4 and detailed method descriptions in Appendix A.2.2.

129 **Training-based Editing (Internal)** methods retrain or fine-tune the target model to suppress a
 130 concept. Safe-CLIP (Poppi et al., 2024) fine-tunes CLIP (Radford et al., 2021) encoders to reduce
 131 sensitivity to unsafe concepts, but is limited to DMs that rely solely on CLIP-based encoders. SOTA
 132 DMs (Esser et al., 2024b; Labs, 2024) and image autoregressive models (Han et al., 2025) employ
 133 alternative text encoders such as T5 (Raffel et al., 2020), making this approach less general. Other
 134 methods update the generative model parameters directly. EDiff (Wu et al., 2025) formulates erasure
 135 as a bi-level optimization problem, Erased Stable Diffusion (ESD) (Gandikota et al., 2023) defines
 136 an objective derived from the classifier-free guidance formulation, SA (Heng & Soh, 2023) uses a
 137 surrogate objective based on Bayesian modeling, and Forget-Me-Not (Zhang et al., 2024a) introduces
 138 a re-steering loss applied to attention layers. More recent work includes EraseAnything (Gao et al.,
 139 2025), which combines LoRA-based adapters with a bi-level min–max optimization, and Minimalist
 140 Concept Erasure (Zhang et al., 2025), which performs end-to-end optimization through the entire
 141 generation process with learnable neuron masks. Several approaches refine which parameters are
 142 updated rather than modifying all weights: SalUn (Fan et al., 2024) and SHS (Wu & Harandi,
 143 2024) use saliency or sensitivity, CA (Kumari et al., 2023) extends this idea to anchor concepts,
 144 and SPM (Lyu et al., 2024) inserts lightweight adapters after each linear and convolutional layer to
 145 block the flow of unwanted information. While these approaches can be effective, they suffer from
 146 high computational cost, dependence on curated data, and limited flexibility across architectures.
 147 Importantly, most have been evaluated only on U-Net-based DMs. Our method closes this important
 148 gap, generalizes over both paradigms and to SOTA transformer-based IARs and DMs.

149 **Closed-form Editing (Internal)** methods update model parameters through analytic projections
 150 without retraining. TIME (Orgad et al., 2023) modifies all cross-attention weights via a projection
 151 constrained by an ℓ_2 norm, and UCE (Gandikota et al., 2024) replaces the constraint with an unrelated
 152 knowledge projection. RECE (Gong et al., 2024) combines both constraints while refining text
 153 embeddings, but does not introduce a fundamentally different mechanism beyond TIME and UCE.
 154 LOCOEDIT (Basu et al., 2024a) applies an ℓ_2 norm constraint, and operates locally by identifying
 155 the most relevant layers through concept and neutral prompts. MACE (Lu et al., 2024) augments
 156 UCE with segmentation masks and trains LoRA adapters on key and value projections inside the
 157 denoising process. Its formulation depends on diffusion time steps and attention over noisy latents,
 158 making it inapplicable to IARs. Closed-form edits are computationally efficient but exhibit trade-offs:
 159 weak edits leave traces of the target concept, while strong edits degrade image quality. Because they
 160 modify parameters throughout the model without accounting for neuron polysemy (Elhage
 161 et al., 2022), they struggle to isolate the target concept while preserving others. Sequential edits are
 162 also unstable, since cumulative projections on the same matrices cause spectral drift and degraded
 163 alignment. In contrast, our method enables precise suppression at the level of semantic features,
 164 resulting in improved trade-offs and robustness across sequential removals.

Overall, our methods combines the advantages of targeted removal from training-based editing with computational costs closer to the closed-form editing methods, making it a practical approach for content moderation.

Sparse Autoencoder (SAE)-based Editing (External) pursue a similar approach to our work by sparsifying the internal embeddings in the model and filtering the concepts that should be removed. These methods either apply an SAE in the image stream to unlearn visual features (Cywiński & Deja, 2025) or operate in the text stream (Kim & Ghadiyaram, 2025; Tian et al., 2025). Both Cywiński & Deja (2025) and Tian et al. (2025) compute a feature importance score for a given concept using two sets of inputs concept and non concept prompts and then intervene on the corresponding SAE features. Specifically, Cywiński & Deja (2025) apply a multiplicative factor to the selected columns, whereas Tian et al. (2025) downscale the identified activations and compare the base layer output with the SAE reconstruction: if the reconstruction error exceeds the scaling threshold, the SAE output is used instead. Kim & Ghadiyaram (2025), in contrast, steer the representation toward a direction derived from SAE features associated with a concept token. A key limitation of these methods is that, in contrast to our Transcoder-based approach, the SAE remains external to the main model architecture. This makes the intervention easily removable under white-box model access. Our approach remains robust also under white-box model access and, therefore, combines the advantages of sparsity-based approaches and internal concept removal techniques.

3 BACKGROUND AND NOTATION

Transcoders are neural networks designed to learn a *sparsified* approximation of multi-layer perceptron (MLP) layers. They promote sparsity by adding an ℓ_1 penalty on latent activations to the reconstruction loss, yielding a more interpretable latent representation.

Formally, let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ denote an input–output pair of an MLP layer (*i.e.*, $\mathbf{y} = \text{MLP}(\mathbf{x})$). A transcoder maps the input \mathbf{x} into a higher-dimensional latent representation $\mathbf{z} \in \mathbb{R}^{m \cdot d}$, where m is the expansion factor, from which it reconstructs the output $\hat{\mathbf{y}} \in \mathbb{R}^d$. The architecture of the transcoder can be formalized as:

$$\begin{aligned} \mathbf{z} &= \text{ReLU}(W_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}}), \\ \hat{\mathbf{y}} &= W_{\text{dec}}\mathbf{z} + \mathbf{b}_{\text{dec}}, \end{aligned} \quad (1)$$

where $W_{\text{enc}} \in \mathbb{R}^{m \cdot d \times d}$ and $W_{\text{dec}} \in \mathbb{R}^{d \times m \cdot d}$ are encoder and decoder weight matrices, respectively. The $\mathbf{b}_{\text{enc}} \in \mathbb{R}^{m \cdot d}$ and $\mathbf{b}_{\text{dec}} \in \mathbb{R}^d$ are learnable bias terms. Through this formulation, each feature in the transcoder is associated with two vectors, namely the i -th row in W_{enc} (encoder feature vector) and the i -th column of W_{dec} (decoder feature vector). Intuitively, for every feature, the encoder vector indicates how much the feature should activate and the decoder vector is scaled by this amount. The resulting weighted sum of the decoder vectors represents the transcoder’s output. Transcoder training minimizes the reconstruction objective of the original MLP with an ℓ_1 regularization on the latent:

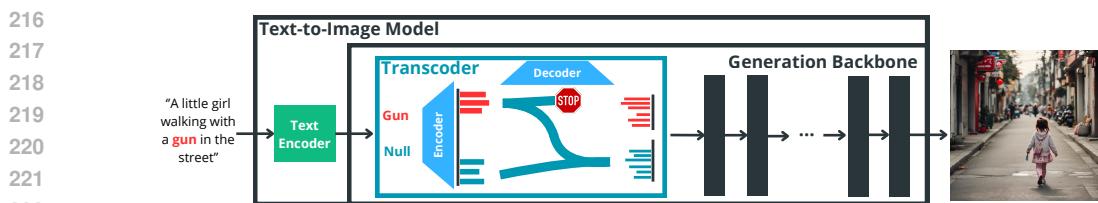
$$\mathcal{L}_t = \underbrace{\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2}_{\text{faithfulness loss}} + \underbrace{\lambda \|\mathbf{z}\|_1}_{\text{sparsity penalty}}, \quad (2)$$

where $\lambda > 0$ controls the trade-off between faithfulness and sparsity (Dunefsky et al., 2024).

Notation. We denote by t a target concept to be removed and by p a prompt used to generate an image with the text-to-image model \mathcal{M} . The prompt consists of P tokens $\{p_1, \dots, p_P\}$, and we denote the tokens that encode concept t by the subset $p_t = \{p_{t1}, \dots, p_{tk}\}$, where k is the number of tokens for concept t . For example, in the prompt $p = \text{"An image of a cat in Van Gogh style"}$, with concept $t = \text{"Van Gogh"}$, $k = 2$ and $p_{t1} = \text{"Van"}$, and $p_{t2} = \text{"Gogh"}$.

4 TRACE: CONCEPT REMOVAL BASED ON TRANSCODERS

In this section, we present TRACE, the first application of transcoders for concept removal in text-to-image generative models. While transcoders have previously enabled analysis of MLP circuits in large language models (Dunefsky et al., 2024), their use for concept removal, in particular for text-to-image architectures, remains unexplored. We leverage transcoders’ ability to sparsely approximate transformation layers to selectively suppress undesired concepts with minimal side effects. We first provide a high-level overview of our framework, then describe transcoder training and our targeted intervention for concept removal.



216
217
218
219
220
221
222
223 **Figure 2: Overview of our TRACE.** We detail our transcoder-based concept removal framework.
224
225

226 4.1 OUR TRANSCODER-BASED CONCEPT REMOVAL FRAMEWORK 227

228 We propose a model-agnostic framework for concept removal in text-to-image generative models.
229 SOTA architectures, including both DMs and IARs, rely on one or more text encoders that generate
230 embeddings to guide image generation. These embeddings are typically injected into the image-
231 generative backbone via transformations such as projection layers or MLPs applied to pooled text
232 features, depending on the specific model (see Appendix D.1 for details). In all cases, we can
233 uniformly replace this transformation with a transcoder that yields condensed, monosemantic features
234 from text representations, enabling precise suppression of targeted content. We present a schematic
235 overview of our approach in Figure 2.

236 By intervening at this early stage of the model, we prevent unwanted features from propagating
237 through the model. Moreover, since our intervention is *permanently* integrated into the generative
238 backbone, rather than *non-permanently* operating externally on the text encoder as in Poppi et al.
239 (2024), simply replacing the text encoder will not undo concept removal with our method. Unlike
240 stand-alone SAE-based modules like SAEuron (Cywiński & Deja, 2025), our approach embeds the
241 Transcoder as a permanent, non-removable, integral component, ensuring persistent concept removal
242 and robustness, even in local model deployments. Additionally, unlike SAE-based methods that
243 intervene in the middle of the network and require extensive inference over multiple DM generation
244 time-steps to collect training samples, our approach operates at the model input, where training data
245 can be obtained directly from the encoding stage with minimal computation, both for DMs and IARs.

246 Finally, despite being located early in the model architecture, our method does not need a computa-
247 tionally expensive backpropagation through the large generative model. Instead, it can be trained
248 stand-alone and separately from the image-generative-model to learn sparse monosemantic features,
249 and then simply "plugged in". During the intervention, we only need to identify the features respon-
250 sible for our target concept t , and redirect them to prevent the generation of t . We detail the training
251 and the intervention for concept removal in the next sections.

252 4.2 TRANSCODER TRAINING 253

254 In the training phase, the transcoder is optimized to replicate the behavior of the original transforma-
255 tion while imposing a monosemantic decomposition of concepts.

256 **Training Data Curation.** The curation of the transcoder's training data is lightweight and simple. It
257 requires a set of prompts, for example, taken from open source datasets. These prompts are forwarded
258 through the model's initial text encoder to yield the training data inputs for the transcoder. The ground
259 truth outputs for the transcoder's training are obtained by inferring these embeddings through the
260 model's original text transformation, *e.g.*, the MLP layer in DMs. Crucially, unlike SAEuron, our
261 approach avoids the need for expensive full-model inference to gather activations. We detail our
262 concrete instantiation for DMs and IARs in Appendix D.2. Based on our generated input-output
263 pairs, the transcoder is then trained to approximate the original transformation.

264 **Training Objective.** The ℓ_0 pseudo-norm measures the number of non-zero elements in a vector,
265 which ideally we would like to minimize for the latent representation $\mathbf{z} \in \mathbb{R}^{m \cdot d}$ in a transcoder
266 to enforce sparsity, but direct optimization with respect to this norm is challenging due to its non-
267 convexity. Therefore, transcoder employs the ℓ_1 norm as a convex surrogate, which promotes sparsity
268 while enabling more efficient optimization. We observed that the original transcoder training with
269 ℓ_1 regularization introduces two key issues for effective concept removal. 1) First, ℓ_1 is only an
imperfect surrogate for the desired ℓ_0 constraint and it only biases positive activations toward zero.

The shrinkage of activations in the ℓ_1 norm instead of zero-ing them out as in ℓ_0 norm, prevents a clean separation between active and inactive units. 2) Second, the formulation often results in many "dead" latents that never activate, leading to under-utilization of the available latent space. For concept removal, both limitations are critical: the lack of a sharp boundary between concept-specific latents hinders precise isolation, thereby affecting both the target concept and unrelated concepts alike, leading to sub-optimal trade-offs. At the same time, the presence of dead latents reduces the model's capacity to encode distinct concepts, resulting in less removal capacity.

As a solution, we replace the ReLU with a TopK activation function, which enforces sparsity by retaining only the k largest entries of the encoded representation. This provides a direct ℓ_0 constraint rather than a surrogate. Our modified encoder-decoder mapping is given by

$$\begin{aligned} \mathbf{z} &= \text{TopK}(W_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}}), \\ \hat{\mathbf{y}} &= W_{\text{dec}}\mathbf{z} + \mathbf{b}_{\text{dec}}. \end{aligned} \quad (3)$$

Since the TopK operator is non-differentiable, we employ a straight-through estimator to propagate gradients during backpropagation. Additional implementation and optimization details are provided in Appendix E.1.

Inspired by work on training SAEs (Gao et al., 2024), we then train with a composite loss \mathcal{L} consisting of three components:

$$\mathcal{L} = \mathcal{L}_{\text{fidelity}} + \alpha_1 \mathcal{L}_{\text{multi-topK}} + \alpha_2 \mathcal{L}_{\text{aux}}, \quad (4)$$

where $\mathcal{L}_{\text{fidelity}} = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$ enforces alignment between the original transformation and the transcoder output. The multi-TopK loss, $\mathcal{L}_{\text{multi-topK}} = \|W_{\text{dec}}\mathbf{z}^{(4k)} + \mathbf{b}_{\text{dec}} - \mathbf{y}\|_2^2$, applies the same reconstruction objective with an expanded TopK budget (typically $4k$) to encourage progressive code utilization. Finally, the auxiliary loss, $\mathcal{L}_{\text{aux}} = \|W_{\text{dec}}\mathbf{z}^{(\text{aux})} + \mathbf{b}_{\text{dec}} - \mathbf{y}\|_2^2$, computes the fidelity objective using only inactive latents, thereby mitigating the problem of dead units. Each term is normalized by the variance of the target activations, so all losses represent fractions of unexplained variance. A detailed description of the loss components is provided in Appendix E.2.

4.3 TRANSCODER-BASED INTERVENTION FOR CONCEPT REMOVAL

We perform the intervention by identifying the features responsible for one or multiple target concepts and redirecting them to prevent the visual generation of the specified concepts.

Feature Identification. Formally, let $\mathbf{z}(p_i) \in \mathbb{R}^{m \cdot d}$ denote the latent vector obtained after applying the TopK encoder to a token p_i . We define the activation indicator

$$\delta_i(p_i) = \mathbf{1}\{z^{(i)}(p_i) \neq 0\}, \quad i \in \{1, \dots, m \cdot d\},$$

which specifies whether latent i is active for token p_i . For a target concept t with tokens $\{p_{t1}, \dots, p_{tk}\}$, $\mathcal{A}_t = \bigcup_{i=1}^k \mathcal{A}_{ti}$ denotes the latent subset that encodes t , *i.e.*, the indices for removal:

$$\mathcal{A}_{ti} = \{i \mid \delta_i(p_{ti}) = 1\}.$$

Concept Removal. Once the subset of latents associated with t has been identified, the goal is to block their contribution to the text representation. Since each latent corresponds to a column of the decoder weight matrix W_{dec} , suppression can be implemented by directly modifying these columns.

We modify W_{dec} such that the activations of \mathcal{A}_t are redirected to reproduce the output of the empty token \emptyset . Let $\mathbf{z}_t \in \mathbb{R}^{md}$ and $\mathbf{z}_\emptyset \in \mathbb{R}^{md}$ denote the latent activations for the target concept and the empty token, respectively. Because the TopK operator produces exactly k active latents per token, each \mathcal{A}_{ti} and \mathcal{A}_\emptyset have the same cardinality. We therefore define a pairing

$$\pi : \mathcal{A}_{ti} \rightarrow \mathcal{A}_\emptyset$$

as a bijection between these sets. Any bijection leads to the same result. In our implementation, we adopt the simplest strategy: latents are paired in order, so the first latent of \mathcal{A}_{ti} is matched to the first of \mathcal{A}_\emptyset , the second to the second, and so on. The decoder is then modified as

$$\forall i \in \{1, \dots, md\}, \quad W_{\text{dec}}^{(i)} = \begin{cases} \frac{z_\emptyset^{(\pi(i))}}{z_t^{(i)} + \varepsilon} W_{\text{dec}}^{(\pi(i))} & \text{if } i \in \mathcal{A}_t, \\ W_{\text{dec}}^{(i)} & \text{otherwise,} \end{cases} \quad (5)$$

324 with $\varepsilon > 0$ a small constant for numerical stability. Under this substitution, the decoder contribution
 325 for \mathbf{z}_t becomes
 326

$$\begin{aligned} 327 \quad W_{\text{dec}} \mathbf{z}_t &= \sum_{i \notin \mathcal{A}_t} W_{\text{dec}}^{(i)} z_t^{(i)} + \sum_{i \in \mathcal{A}_t} \frac{z_\emptyset^{(\pi(i))}}{z_t^{(i)} + \varepsilon} W_{\text{dec}}^{(\pi(i))} z_t^{(i)} = \sum_{i \in \mathcal{A}_t} \frac{z_\emptyset^{(\pi(i))}}{z_t^{(i)} + \varepsilon} W_{\text{dec}}^{(\pi(i))} z_t^{(i)} \\ 328 \quad &\approx \sum_{i \in \mathcal{A}_t} W_{\text{dec}}^{(\pi(i))} z_\emptyset^{(\pi(i))} = W_{\text{dec}} \mathbf{z}_\emptyset, \\ 329 \end{aligned} \quad (6)$$

333 showing that the contribution of \mathbf{z}_t is redirected to reproduce the output of the empty token, thereby
 334 removing the target concept while preserving a coherent representation.
 335

336 5 EMPIRICAL EVALUATION

337 5.1 EXPERIMENTAL SETUP

340 **Models.** We mainly evaluate our method on SOTA text-to-image generative models from two
 341 major architectures, namely DMs and IARs. For **DMs**, we consider SD3.5 (Esser et al., 2024a),
 342 Flux (Labs, 2024). For **IARs**, we analyze Infinity-2B and Infinity-8B (Han et al., 2025). This
 343 selection ensures coverage of both architectures and multiple model scales to highlight our method’s
 344 generality. **To compare further with prior work that was tailored to and evaluated on older models,
 345 we also experiment with SD1.5.**

346 **Concept Removal Tasks and Benchmarks.** We evaluate our method on **style removal** and **object**
 347 **removal** from UnlearnCanvas (Zhang et al., 2024b). UnlearnCanvas provides a benchmark for
 348 object and style unlearning. Yet, the benchmark’s style classifier is trained on SD1.5 generations and
 349 generalizes poorly to modern architectures such as SD3.5, Flux, and Infinity (below 6% accuracy,
 350 see Table 7). Therefore, we replace it with a LLaVA-1.6-Vicuna-7B (Liu et al., 2023) as a unified
 351 zero-shot classifier (see Appendix E.4 for details). With this external classifier, we focus on ten
 352 visually distinct styles (Cartoon, Cubism, Winter, Pop Art, Ukiyoe, Impressionism, Byzantine, Van
 353 Gogh, Bricks, Watercolor), while retaining the object categories defined in the original benchmark.

354 **Metrics.** We assess concept removal using classification-based metrics together with distributional
 355 quality measures. For a target concept c , we report **Unlearning Accuracy (UA)** as the fraction of
 356 images generated with prompts containing c that are not classified as c . To measure preservation
 357 of unrelated content, we report two retention metrics: **In-domain Retain Accuracy (IRA)**, the
 358 classification accuracy on non-target concepts from the same domain (*e.g.*, other objects when
 359 unlearning one object), and **Cross-domain Retain Accuracy (CRA)**, the accuracy on concepts
 360 from a different domain (*e.g.*, style accuracy when unlearning an object). Image quality is evaluated
 361 with the Fréchet Inception Distance (FID) (Heusel et al., 2017), **HPSv3** (Ma et al., 2025b), and
 362 **Aesthetics v2.5**¹ computed between intervened and original generations after excluding samples of
 363 the removed concept, and CLIPScore (Hessel et al., 2021) under the same exclusion, as concept
 364 removal is supposed to break the alignment between prompts containing the target concept and their
 365 generated image.

366 **Transcoder.** We initialize the transcoder with an expansion factor of 16 and a TopK budget of
 367 $k = 32$. We train it for 100 epochs on the curated dataset introduced by Cywiński & Deja (2025). A
 368 complete description of hyperparameters and training settings is provided in Appendix E.1.

369 **Baselines.** Overall, we compare our approach against 12 state-of-the-art baselines. For the state-
 370 of-the-art models, namely SD3.5, Flux, Infinity-2B, and Infinity-8B, we compare TRACE against
 371 UCE (Gandikota et al., 2024) and LOCOEDIT (Basu et al., 2024b), the most relevant white-box
 372 editing methods that are directly applicable to both DMs and IARs. UCE has demonstrated SOTA
 373 unlearning accuracy (UA) in Zhang et al. (2024b), while LOCOEDIT is designed to perform localized
 374 interventions that aim to preserve model utility by restricting edits to concept-relevant layers. For
 375 SD3.5 and Flux, which use dynamic text representations that are incompatible with LOCOEDIT’s
 376 original localization step, we adapt the method using zero-ablation as described in Appendix B.2.
 377 **For SD1.5, we extend the comparison by adding MACE (Lu et al., 2024), SPM (Lyu et al., 2024),**

¹<https://github.com/discuss0434/aesthetic-predictor-v2-5>

378 **Table 1: Style and object removal performance of the proposed method and SOTA baselines.**
 379 FID, CLIP, HPSv3, and Aesthetic v2.5 scores are reported on non-target data.
 380

Model	Method	Style Removal			Object Removal			FID (↓)	CLIP (↑)	HPSv3 (↑)	Aesthetic v2.5 (↑)
		UA (↑)	IRA (↑)	CRA (↑)	UA (↑)	IRA (↑)	CRA (↑)				
SD3.5	LOCOEDIT	41.49	59.23	84.57	37.87	85.78	58.89	54.66	0.3246	7.46	6.02
	UCE	61.78	61.89	83.47	84.45	82.23	60.93	62.34	0.3205	6.19	6.11
	Ours	69.60	67.30	92.60	68.00	93.00	67.50	51.89	0.3376	8.31	6.15
Flux	LOCOEDIT	66.45	33.23	83.44	35.75	91.23	35.76	55.56	0.2911	8.13	6.13
	UCE	67.43	34.78	76.56	87.45	82.34	36.43	58.90	0.2996	7.54	6.34
	Ours	88.60	36.10	96.40	93.20	96.61	38.20	51.67	0.3059	8.49	6.49
Infinity-2B	LOCOEDIT	81.27	37.65	62.45	85.72	68.09	34.52	91.55	0.3007	6.57	5.27
	UCE	68.59	35.09	74.29	72.48	69.34	31.43	77.56	0.3078	6.38	5.54
	Ours	86.80	39.60	90.30	91.20	84.50	38.30	90.92	0.3093	6.98	5.61
Infinity-8B	LOCOEDIT	80.07	59.61	63.47	88.67	70.54	57.44	56.21	0.3225	7.87	6.47
	UCE	76.75	58.76	68.67	75.35	69.41	56.56	62.34	0.3289	7.96	6.44
	Ours	85.40	63.70	95.50	90.20	95.70	61.40	53.93	0.3338	8.54	6.51

391
 392 and ANT (Li et al., 2025) for sequential unlearning (Section 5.3). We also extend the comparison
 393 for SD1.5 on UnlearnCanvas benchmark to ESD (Gandikota et al., 2023), FMN (Zhang et al.,
 394 2024a), UCE (Gandikota et al., 2024), CA (Kumari et al., 2023), SalUn (Fan et al., 2024), SEOT (Li
 395 et al., 2024), SPM (Lyu et al., 2024), EDiff (Wu et al., 2025), SHS (Wu & Harandi, 2024) and
 396 SAeUron (Cywiński & Deja, 2025).
 397

398 5.2 OUR METHOD YIELDS EFFECTIVE CONCEPT REMOVAL

400 **State-of-the-Art IARs and DMs.** We first compare our method to the baselines on state-of-the-art
 401 models in Table 1. The results highlight that our method outperforms the baselines over all models.
 402 The results in Table 1 show that our method delivers consistent improvements in concept removal
 403 across both style and object domains. On SD3.5, our approach achieves 69.6% UA, 67.3% IRA, and
 404 92.6% CRA for style removal, outperforming LOCOEDIT and UCE by margins of 8-28%. For object
 405 removal on SD3.5, UCE attains a strong 84.5% UA, but our method achieves higher in-domain and
 406 cross-domain retention (93.0% IRA and 67.5% CRA), demonstrating that it more reliably removes
 407 the target concept while preserving fidelity to the original domain and maintaining transferability
 408 across domains.

409 On Flux, our method’s improvements are even more pronounced. It reaches 88.6% UA and 96.4%
 410 CRA for style removal, with over 93% UA and 96.6% IRA for object removal, surpassing LOCOEDIT
 411 and UCE by wide margins. These results indicate that our approach is particularly effective at
 412 disentangling style attributes, which are often more diffuse and challenging to isolate, while still
 413 retaining high accuracy within and across domains.

414 For Infinity-2B, LOCOEDIT performs reasonably well on object removal, yet our method still
 415 achieves the highest overall results, with 90.3% CRA for style removal and 91.2% UA for object
 416 removal. Compared to UCE, which struggles on this model, our method provides clear advantages
 417 across all three metrics. On the larger Infinity-8B model, our method again delivers the best
 418 performance, reaching 95.5% CRA for style removal and 95.7% IRA for object removal, confirming
 419 that our approach scales effectively with model size while ensuring both in-domain and cross-domain
 420 consistency across novel SOTA models and training paradigms.

421 In addition to removal accuracy, our approach preserves generative quality, as reflected by lower
 422 FID and higher CLIP scores across most settings. This balance between strong concept suppression
 423 and high-fidelity image synthesis highlights the effectiveness of our method compared to existing
 424 baselines, which often face trade-offs between removal strength and image quality. We complement
 425 these quantitative results with qualitative examples in Appendix C. Furthermore, the comparison of
 426 training time, storage, and memory requirements in Appendix F.6 demonstrates a clear advantage of
 427 our method over training-based approaches.

428 **Full Baseline Comparison against SD1.5.** We further compare our method against additional
 429 baselines Section 5.2 on the UnlearnCanvas benchmark (Zhang et al., 2024b) using the less capable
 430 SD1.5 model. Since this model lacks an explicit bottleneck between the text and image streams, we
 431 introduce a virtual identity layer between the CLIP text encoder and the cross-attention text input,
 which we then replace with our transcoder. We use an expansion factor of 32 and $k = 32$.

432
 433 **Table 2: Evaluation of style and object removal against state-of-the-art methods on style and**
object unlearning. We evaluate on SD 1.5 finetune model of unlearn canvas. The best result for each
 434 metric is highlighted in bold, and the second-best is underlined.
 435

Method	Style Removal			Object Removal			Mean (\uparrow)
	UA (\uparrow)	IRA (\uparrow)	CRA (\uparrow)	UA (\uparrow)	IRA (\uparrow)	CRA (\uparrow)	
ESD	98.58%	80.97%	93.96%	92.15%	55.78%	44.23%	77.28%
FMN	88.48%	<u>56.77%</u>	46.60%	45.64%	90.63%	73.46%	66.60%
UCE	98.40%	60.22%	47.71%	<u>94.31%</u>	39.35%	34.67%	62.78%
CA	60.82%	<u>96.01%</u>	92.70%	<u>46.67%</u>	90.11%	<u>81.97%</u>	77.38%
SaiUn	86.26%	90.39%	95.08%	<u>86.91%</u>	96.35%	99.59%	<u>92.43%</u>
SEOT	56.90%	94.68%	84.31%	23.25%	<u>95.57%</u>	82.71%	72.23%
SPM	60.94%	92.39%	84.33%	71.25%	90.79%	81.65%	80.56%
EDiff	92.42%	73.91%	<u>98.93%</u>	86.67%	94.03%	48.48%	82.07%
SHS	95.84%	80.42%	43.27%	80.73%	81.15%	67.99%	74.24%
SAeUron	95.80%	99.10%	99.40%	78.82%	95.47%	95.58%	94.03%
Ours	95.02%	93.84%	86.22%	83.88%	91.82%	97.87%	91.11%

448
 449
 450 Note that not all these baselines operate in the same setup and threat space as our TRACE. Therefore,
 451 the results only provide a context for TRACE’s performance in the overall landscape of concept
 452 removal methods. Our results in Table 2 highlight that also on SD1.5, our TRACE remains competitive
 453 to the baselines. While ESD outperforms TRACE in terms of UA, this comes at the cost of strong
 454 reductions in IRA, which is undesirable. In contrast, TRACE maintains a more careful balance
 455 between UA and IRA. SAeUron overall achieves slightly better trade-offs than ours. Note, however,
 456 as discussed in Section 2 that this method is an *external* editing method which can be removed
 457 without affecting the model’s generation performance while ours, at least on the state-of-the art
 458 models evaluated above, is integrated *internally* to the model, making it a part that cannot be removed
 459 without harming model performance. This makes TRACE applicable in white-box model access
 460 setups where SAeUron fails.
 461

462 5.3 MULTI-CONCEPT REMOVAL

464 We assess our method’s ability to remove multiple concepts in comparison to the baselines. Con-
 465 cretely, we distinguish between two setups. **1) Simultaneous Removal** removes N concepts by
 466 identifying the corresponding latent subsets in the transcoder and jointly redirecting their decoder
 467 contributions to the empty token, ensuring that none of the N concepts are expressed during the
 468 generation. **2) Sequential Removal** removes one concept at the time until reaching N . For our
 469 method, both setups are functionally identical since after N sequential steps the same set of latents
 470 has been redirected as in the simultaneous case. In contrast, for baselines LOCOEDIT and UCE,
 471 sequential removal requires iterative weight edits, which accumulate across steps and lead to de-
 472 graded performance compared to the simultaneous setting. For simultaneous removal, we re-run the
 473 experiment with different $N \in \{1, \dots, 10\}$, for sequential removal, we continue the experiment from
 474 the removal of $N = 1$, and then add another one at the time, until we also reach $N = 10$.

475 We present the results in Figure 3, where the y -axis reports the average score defined as $(UA + (IRA +$
 476 $CRA)/2)/2$, following Cywiński & Deja (2025). This score combines the unlearning and retain
 477 capabilities into a single metric for simpler comparison. We report the mean computed 5 random
 478 seeds. Across all models and both setups, our method consistently achieves higher average scores
 479 than LOCOEDIT and UCE. In the sequential setting, the performance of the baselines drops sharply
 480 as more concepts are removed, reflecting the accumulation of errors from repeated weight edits. In
 481 contrast, our method remains stable, showing only a gradual decline even when ten concepts are
 482 removed. In the simultaneous setting, all methods perform better at small N , but the gap between our
 483 method and the baselines widens as N increases, with our approach maintaining substantially higher
 484 scores. The advantage is especially pronounced for larger models such as SD3.5 and Infinity-8B,
 485 where the baselines degrade most quickly. These results demonstrate that our TRACE provide robust
 486 multi-concept removal that scales reliably with N , avoiding the compounding errors of sequential
 487 editing while maintaining strong retention across both setups. While the results in Figure 3 are for

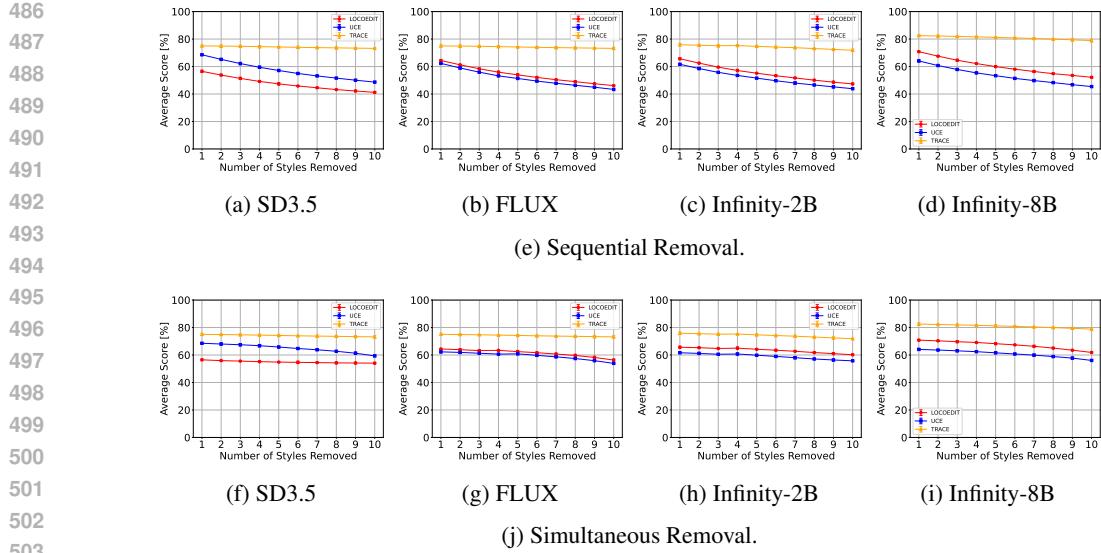


Figure 3: Multi-Concept Style Removal of Our Method and SOTA Baselines.

style removal, the same trends can be observed for object removal as we show in Appendix F.1. To broaden the evaluation for style removal, we also include results for SD1.5 in Appendix F.2.

5.4 ROBUSTNESS EVALUATION

To evaluate robustness of our method under attacks, we test the effectiveness of Ring-A-Bell (Tsai et al., 2024), an adversarial prompting framework originally proposed against DMs where it was designed to bypass concept removal methods by leveraging adversarially optimized input prompts. It constructs adversarial queries by extracting semantic representations of sensitive concepts (*e.g.*, nudity, violence, or artistic styles) from text encoders (CLIP), and then applies discrete optimization to generate prompts that reintroduce these concepts even when models have been explicitly trained to suppress them. Importantly, the attack operates without requiring access to model internals. Table 3 shows Ring-A-Bell attack success rates (%) across models, with lower values indicating greater robustness. The original models are the most vulnerable, reaching 58.08% on SD3.5 and 56.34% on Infinity-8B. Both LOCOEDIT and UCE reduce these rates, but their protection remains partial, especially on stronger models like SD3.5. Our approach achieves the lowest success rates across all four models, cutting attack success by 33 percentage points on SD3.5 (from 58.08% to 25.21%) and by over 36 points on Infinity-8B (from 56.34% to 19.78%). Even on Flux and Infinity-2B, where baseline vulnerabilities are lower, our method still outperforms by clear margins. These results demonstrate that our TRACE provides reliable concept removal even under attacks. Additional diverse adversarial attack benchmarks are presented in Appendix F.3.

6 CONCLUSION

We introduced **TRAnscoder-based Concept Editing (TRACE)**, a practical and scalable framework for robust concept removal in SOTA text-to-image generative models. By integrating concept removal directly into the model backbone via transcoders, TRACE achieves permanent and efficient interventions. Our approach yields strong performance in removing unwanted concepts while preserving image fidelity and robustness to adversarial attacks, and it generalizes across training paradigms and multi-concept settings. These advances position TRACE as a foundation for more controllable, safe, and responsible deployment of generative models.

Table 3: Attack Success Rates.

Method	SD3.5	Flux	Inf-2B	Inf-8B
Original	58.08	28.43	36.46	56.34
LOCOEDIT	57.36	26.32	24.67	22.61
UCE	43.44	24.45	25.31	24.43
Ours	25.21	15.62	21.23	19.78

540 REFERENCES
541

- 542 Andrea Asperti, Franky George, Tiberio Marras, Razvan Ciprian Stricescu, and Fabio Zanotti. A
543 critical assessment of modern generative models’ ability to replicate artistic styles. *Big Data and*
544 *Cognitive Computing*, 9(9):231, 2025.
- 545 Samyadeep Basu, Keivan Rezaei, Priyatham Kattakinda, Vlad Morariu, Nanxuan Zhao, R_A Rossi,
546 Varun Manjunatha, and Soheil Feizi. On mechanistic knowledge localization in text-to-image
547 generative models. International Conference on Machine Learning, 2024a.
- 548 Samyadeep Basu, Nanxuan Zhao, Vlad Morariu, Soheil Feizi, and Varun Manjunatha. Localizing
549 and editing knowledge in text-to-image generative models. 2024b.
- 550 Black Forest Labs. Flux.1, 2024. URL <https://blackforestlabs.ai/announcing-black-forest-labs/>.
- 551 Bartosz Cywiński and Kamil Deja. Saeuron: Interpretable concept unlearning in diffusion models
552 with sparse autoencoders. *International Conference on Machine Learning*, 2025.
- 553 Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable llm feature
554 circuits. *Advances in Neural Information Processing Systems*, 37:24375–24410, 2024.
- 555 Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec,
556 Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition.
557 *arXiv preprint arXiv:2209.10652*, 2022.
- 558 Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam
559 Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers
560 for high-resolution image synthesis. In *Forty-first international conference on machine learning*,
561 2024a.
- 562 Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam
563 Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for
564 high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*,
565 2024b.
- 566 Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Em-
567 powering machine unlearning via gradient-based weight saliency in both image classification and
568 generation. *The Twelfth International Conference on Learning Representations*, 2024.
- 569 Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts
570 from diffusion models. In *Proceedings of the IEEE/CVF international conference on computer*
571 *vision*, pp. 2426–2436, 2023.
- 572 Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified
573 concept editing in diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on*
574 *Applications of Computer Vision*, pp. 5111–5120, 2024.
- 575 Daiheng Gao, Shilin Lu, Wenbo Zhou, Jiaming Chu, Jie Zhang, Mengxi Jia, Bang Zhang, Zhaoxin
576 Fan, and Weiming Zhang. Eraseanything: Enabling concept erasure in rectified flow transformers.
577 In *Forty-second International Conference on Machine Learning*, 2025.
- 578 Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya
579 Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint*
580 *arXiv:2406.04093*, 2024.
- 581 Reid Southen Gary Marcus. Generative ai has a visual plagiarism problem: Experiments with
582 midjourney and dall-e 3 show a copyright minefield, 2024. URL <https://spectrum.ieee.org/midjourney-copyright>. Accessed: 2024-01-06.
- 583 Chao Gong, Kai Chen, Zhipeng Wei, Jingjing Chen, and Yu-Gang Jiang. Reliable and efficient
584 concept erasure of text-to-image diffusion models. In *European Conference on Computer Vision*,
585 pp. 73–88. Springer, 2024.

- 594 Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing
 595 Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In
 596 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15733–15744, 2025.
 597
- 598 Alvin Heng and Harold Soh. Selective amnesia: A continual learning approach to forgetting in deep
 599 generative models. *Advances in Neural Information Processing Systems*, 36:17170–17194, 2023.
- 600 Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-
 601 free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical
 602 Methods in Natural Language Processing*, pp. 7514–7528, 2021.
 603
- 604 Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans
 605 trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural
 606 information processing systems*, 30, 2017.
- 607 Chi-Pin Huang, Kai-Po Chang, Chung-Ting Tsai, Yung-Hsuan Lai, and Yu-Chiang Frank Wang.
 608 Receler: Reliable concept erasing of text-to-image diffusion models via lightweight erasers.
 609 *European Conference on Computer Vision*, 2023. doi: 10.48550/arXiv.2311.17717.
 610
- 611 Dahye Kim and Deepti Ghadiyaram. Concept steerers: Leveraging k-sparse autoencoders for
 612 controllable generations. *arXiv preprint arXiv:2501.19066*, 2025.
 613
- 614 Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan
 615 Zhu. Ablating concepts in text-to-image diffusion models. In *Proceedings of the IEEE/CVF
 616 International Conference on Computer Vision*, pp. 22691–22702, 2023.
- 617 Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024.
 618
- 619 Leyang Li, Shilin Lu, Yan Ren, and Adams Wai-Kin Kong. Set you straight: Auto-steering denois-
 620 ing trajectories to sidestep unwanted concepts. In *Proceedings of the 33rd ACM International
 621 Conference on Multimedia*, MM ’25, pp. 9257–9266, New York, NY, USA, 2025. Associa-
 622 tion for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3754546. URL
 623 <https://doi.org/10.1145/3746027.3754546>.
- 624 Senmao Li, Joost van de Weijer, Taihang Hu, Fahad Shahbaz Khan, Qibin Hou, Yaxing Wang, and
 625 Jian Yang. Get what you want, not what you don’t: Image content suppression for text-to-image
 626 diffusion models. *The Twelfth International Conference on Learning Representations*, 2024.
 627
- 628 Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro
 629 Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects
 630 in context. *arXiv preprint arXiv: 1405.0312*, 2014.
- 631 Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*,
 632 2023.
 633
- 634 Shilin Lu, Zilan Wang, Leyang Li, Yanzhu Liu, and Adams Wai-Kin Kong. Mace: Mass concept
 635 erasure in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and
 636 Pattern Recognition*, pp. 6430–6440, 2024.
- 637 Mengyao Lyu, Yuhong Yang, Haiwen Hong, Hui Chen, Xuan Jin, Yuan He, Hui Xue, Jungong Han,
 638 and Guiguang Ding. One-dimensional adapter to rule them all: Concepts diffusion models and
 639 erasing applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
 640 Recognition*, pp. 7559–7568, 2024.
- 642 Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human
 643 preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision
 644 (ICCV)*, pp. 15086–15095, October 2025a.
 645
- 646 Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human
 647 preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*,
 pp. 15086–15095, 2025b.

- 648 Hadas Orgad, Bahjat Kawar, and Yonatan Belinkov. Editing implicit assumptions in text-to-image
 649 diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*,
 650 pp. 7053–7061, 2023.
- 651 William S. Peebles and Saining Xie. Scalable diffusion models with transformers. *IEEE International
 652 Conference on Computer Vision*, 2022. doi: 10.1109/ICCV51070.2023.00387.
- 653 Samuele Poppi, Tobia Poppi, Federico Cocchi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara.
 654 Safe-clip: Removing nsfw concepts from vision-and-language models. In *European Conference
 655 on Computer Vision*, pp. 340–356. Springer, 2024.
- 656 Yiting Qu, Xinyue Shen, Xinlei He, Michael Backes, Savvas Zannettou, and Yang Zhang. Unsafe
 657 diffusion: On the generation of unsafe images and hateful memes from text-to-image models. In
 658 *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*,
 659 pp. 3403–3417, 2023.
- 660 Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish
 661 Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and I. Sutskever. Learning
 662 transferable visual models from natural language supervision. *International Conference on Machine
 663 Learning*, 2021.
- 664 Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi
 665 Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text
 666 transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
- 667 Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen,
 668 and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine
 669 learning*, pp. 8821–8831. Pmlr, 2021.
- 670 Javier Rando, Daniel Paleka, David Lindner, Lennart Heim, and Florian Tramèr. Red-teaming the
 671 stable diffusion safety filter. *arXiv preprint arXiv:2210.04610*, 2022.
- 672 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-
 673 resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF confer-
 674 ence on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- 675 Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion:
 676 Mitigating inappropriate degeneration in diffusion models. In *Proceedings of the IEEE/CVF
 677 Conference on Computer Vision and Pattern Recognition*, pp. 22522–22531, 2023.
- 678 StabilityAI. DeepFloyd IF: a novel state-of-the-art open-source text-to-image model with a high
 679 degree of photorealism and language understanding. [https://github.com/deep-floyd/
 680 IF](https://github.com/deep-floyd/IF), 2023. Last accessed on 2025-01-17.
- 681 Haotian Tang, Yecheng Wu, Shang Yang, Enze Xie, Junsong Chen, Junyu Chen, Zhuoyang Zhang,
 682 Han Cai, Yao Lu, and Song Han. Hart: Efficient visual generation with hybrid autoregressive
 683 transformer. *arXiv preprint arXiv:2410.10812*, 2024.
- 684 Zhihua Tian, Sirun Nan, Ming Xu, Shengfang Zhai, Wenjie Qu, Jian Liu, Ruoxi Jia, and Jiaheng
 685 Zhang. Sparse autoencoder as a zero-shot classifier for concept erasing in text-to-image diffusion
 686 models. *arXiv preprint arXiv:2503.09446*, 2025.
- 687 Yu-Lin Tsai, Chia-Yi Hsu, Chulin Xie, Chih-Hsun Lin, Jia-You Chen, Bo Li, Pin-Yu Chen, Chia-Mu
 688 Yu, and Chun-Ying Huang. Ring-a-bell! how reliable are concept removal methods for diffusion
 689 models? In *The Twelfth International Conference on Learning Representations, ICLR 2024,
 690 Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL [https://openreview.net/
 691 forum?id=1m7MRcsFis](https://openreview.net/forum?id=1m7MRcsFis).
- 692 Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks.
 693 In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016.
- 694 Runpu Wei, Zijin Yin, Shuo Zhang, Lanxiang Zhou, Xueyi Wang, Chao Ban, Tianwei Cao, Hao Sun,
 695 Zhongjiang He, Kongming Liang, et al. Omnieraser: Remove objects and their effects in images
 696 with paired video-frame data. *arXiv preprint arXiv:2501.07397*, 2025.

- 702 Jing Wu and Mehrtash Harandi. Scissorhands: Scrub data influence via connection sensitivity in
 703 networks. In *European Conference on Computer Vision*, pp. 367–384. Springer, 2024.
 704
- 705 Jing Wu, Trung Le, Munawar Hayat, and Mehrtash Harandi. Erasing undesirable influence in
 706 diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp.
 707 28263–28273, 2025.
- 708 Yijun Yang, Ruiyuan Gao, Xiaosen Wang, Tsung-Yi Ho, Nan Xu, and Qiang Xu. MMA-Diffusion:
 709 MultiModal Attack on Diffusion Models. In *Proceedings of the IEEE Conference on Computer*
 710 *Vision and Pattern Recognition (CVPR)*, 2024a.
- 711 Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. Sneakyprompt: Jailbreaking
 712 text-to-image generative models. In *2024 IEEE symposium on security and privacy (SP)*, pp.
 713 897–912. IEEE, 2024b.
- 714 Jaehong Yoon, Shoubin Yu, Vaidehi Patil, Huaxiu Yao, and Mohit Bansal. Safree: Training-free
 715 and adaptive guard for safe text-to-image and video generation. *The Thirteenth International*
 716 *Conference on Learning Representations*, 2025.
- 717 Gong Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not:
 718 Learning to forget in text-to-image diffusion models. In *Proceedings of the IEEE/CVF conference*
 719 *on computer vision and pattern recognition*, pp. 1755–1764, 2024a.
- 720
- 721 Yang Zhang, Er Jin, Yanfei Dong, Yixuan Wu, Philip Torr, Ashkan Khakzar, Johannes Stegmaier,
 722 and Kenji Kawaguchi. Minimalist concept erasure in generative models. *International Conference*
 723 *on Machine Learning*, 2025.
- 724
- 725 Yihua Zhang, Chongyu Fan, Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jiancheng Liu, Gaoyuan
 726 Zhang, Gaowen Liu, Ramana Rao Kompella, Xiaoming Liu, and Sijia Liu. Unlearnncanvas: Stylized
 727 image dataset for enhanced machine unlearning evaluation in diffusion models. In *The Thirty-eight*
 728 *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b.
 729 URL <https://openreview.net/forum?id=t9aThFL11E>.
- 730
- 731 Yimeng Zhang, Jinghan Jia, Xin Chen, Aochuan Chen, Yihua Zhang, Jiancheng Liu, Ke Ding, and
 732 Sijia Liu. To generate or not? safety-driven unlearned diffusion models are still easy to generate
 733 unsafe images... for now. *European Conference on Computer Vision (ECCV)*, 2024c.
- 734
- 735 Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and video tokenization with binary
 736 spherical quantization. *arXiv preprint arXiv: 2406.07548*, 2024.
- 737
- 738
- 739
- 740
- 741
- 742
- 743
- 744
- 745
- 746
- 747
- 748
- 749
- 750
- 751
- 752
- 753
- 754
- 755

756 **A EXTENDED BACKGROUND**

757 **A.1 DIFFUSION AND IMAGE AUTOREGRESSIVE MODELS**

760 **DMs.** DMs generate images by learning to invert a forward noising process. In the forward process,
 761 an image \mathbf{x}_0 is gradually perturbed into Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(0, I)$ using a variance schedule
 762 $\{\beta_t\}_{t=1}^T$. A neural network is trained to approximate the reverse conditionals $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$, typically
 763 parameterized as noise prediction. At inference, generation begins from random noise and applies
 764 the learned reverse steps until an image is reconstructed. Large-scale text-to-image systems such as
 765 Stable Diffusion (Rombach et al., 2022) adopted U-Net backbones as the denoising network. These
 766 models operate in a latent space obtained from an autoencoder and rely on cross-attention to inject
 767 conditioning. A separate text encoder produces embeddings once at the start of sampling. These
 768 text embeddings remain fixed for the entire denoising trajectory, and the U-Net attends to them as
 769 static keys and values at multiple layers. This design achieves a good performance but restricts
 770 text conditioning to a fixed representation. Recent models replace U-Nets with transformer-based
 771 denoisers. Diffusion Transformers (DiTs) (Peebles & Xie, 2022) represent images as sequences of
 772 latent patches and model dependencies with standard transformer blocks. The Multi-Modal DiT
 773 (MMDiT) (Esser et al., 2024a) extends this formulation to text-to-image generation by processing
 774 image and text tokens jointly in the same transformer stack. In architectures such as SD3.5, and
 775 Flux, text embeddings are updated dynamically at every layer through multimodal attention, rather
 776 than being fixed once at the beginning. This design integrates conditioning more deeply and scales
 777 effectively with model size. Modern DMs therefore fall into two main categories: U-Net-based
 778 systems with static text embeddings, and DiT-based systems with dynamic multimodal conditioning.
 779 Our method is compatible with both families, independent of their choice of backbone or conditioning
 780 strategy.

780 **Image AutoRegressive Models** IARs model the joint distribution of an image by decomposing
 781 it autoregressively into a product of conditional probabilities over discrete tokens. An image \mathbf{x} is
 782 first quantized into a sequence $\mathbf{z} = (z_1, \dots, z_N)$ using a tokenzier, and the conditional distribution is
 783 expressed as

$$784 \quad p_\theta(\mathbf{z} | \mathbf{y}) = \prod_{i=1}^N p_\theta(z_i | z_{<i}, \mathbf{y}),$$

787 where \mathbf{y} denotes a text prompt. During training, the model observes the ground-truth tokens $\{z_i\}_{i=1}^N$
 788 from the tokenizer and learns to maximize their likelihood under the predicted conditional distribu-
 789 tions. The training objective is therefore the negative log-likelihood

$$790 \quad \mathcal{L}_{\text{IAR}} = - \sum_{i=1}^N \log p_\theta(z_i | z_{<i}, \mathbf{y}).$$

793 Infinity (Han et al., 2025) extends autoregressive image generation by replacing fixed-size token vo-
 794 cabularies with a bitwise multi-scale residual quantizer (Zhao et al., 2024). An image is hierarchically
 795 decomposed into binary variables across scales, and the model predicts each bit autoregressively
 796 given the previously generated bits and the text prompt. The training loss is

$$798 \quad \mathcal{L}_{\text{Infinity}} = - \sum_{s=1}^S \sum_{b=1}^B \log p_\theta(z_{s,b} | z_{<s,<b}, \mathbf{y}),$$

801 where s indexes scales and b indexes bits within each scale. Text conditioning is provided by
 802 embeddings from a Flan-T5 encoder. These embeddings are computed once and remain fixed
 803 throughout generation, and are injected into the transformer backbone via cross-attention at every
 804 layer so that image tokens repeatedly attend to the same text features. Infinity therefore trains under a
 805 maximum likelihood objective while scaling autoregressive modeling efficiently to high-resolution
 806 text-to-image generation.

807 **A.2 CONCEPT EDITING/REMOVAL**

809 We present a taxonomy on concept removal approaches in Table 4.

Table 4: **Taxonomy of concept removal methods.** Properties are marked with ✓ (yes) or ✗ (no).

Method	Permanent	Paradigm-Agnostic	Fine-Tuning Free
FMN (Zhang et al., 2024a)	✗	✗	✗
MACE (Lu et al., 2024)	✓	✗	✗
MCE (Zhang et al., 2025)	✓	✗	✗
ESD (Gandikota et al., 2023)	✓	✗	✗
EA (Gao et al., 2025)	✓	✗	✗
CA (Kumari et al., 2023)	✓	✗	✗
EDiff (Wu et al., 2025)	✓	✗	✗
RECELER (Huang et al., 2023)	✓	✗	✗
SalUn (Fan et al., 2024)	✓	✓	✗
SHS (Wu & Harandi, 2024)	✓	✓	✗
SLD (Schramowski et al., 2023)	✗	✓	✓
SAFREE (Yoon et al., 2025)	✗	✓	✓
SEOT (Li et al., 2024)	✗	✓	✗
SAEuron (Cywiński & Deja, 2025)	✗	✓	✓
SPM (Lyu et al., 2024)	✓	✗	✗
TIME (Orgad et al., 2023)	✓	✓	✓
RECE (Gong et al., 2024)	✓	✓	✓
LOCO EDIT (Basu et al., 2024a)	✓	✓	✓
UCE (Gandikota et al., 2024)	✓	✓	✓
TRACE (Ours)	✓	✓	✓

A.2.1 EXTERNAL METHODS

Several methods suppress target concepts without modifying model weights, intervening only externally at inference time.

SLD (Schramowski et al., 2023) (Safe Latent Diffusion) modifies the classifier-free guidance (CFG) formulation applied during denoising. In the standard setup, CFG interpolates between unconditional and conditional predictions. SLD augments this process with additional “safe” and “unsafe” guidance terms and a custom dynamic. This approach is appealing because it requires no retraining, but it incurs computational overhead due to the extra predictions. Moreover, constraining the CFG direction often reduces fidelity and prompt adherence. As an inference-time intervention, it can also be trivially disabled in local deployments.

SAEuron (Cywiński & Deja, 2025) introduces sparse autoencoders (SAEs) trained to disentangle semantic features in the residual stream of frozen DMs. The SAEs identify features corresponding to unwanted concepts, which can then be suppressed by blocking their activations before they re-enter the model. This approach provides finer control than global steering, but it relies on auxiliary modules trained outside the model. Because the underlying weights remain unchanged, the suppression can be undone simply by removing the SAE modules, making the intervention non-persistent.

SEOT (Li et al., 2024) suppresses unwanted content by directly optimizing text embeddings at inference. It first applies soft-weighted regularization to reduce redundant encodings of the negative concept in the embedding space. Then, during generation, it performs embedding optimization with two explicit losses: a suppression loss that discourages attention to the target concept and a preservation loss that encourages fidelity to the original prompt. This optimization is repeated for every input, which increases latency, and because it operates entirely in embedding space, the generative model itself remains unmodified.

SAFREE (Yoon et al., 2025) introduces a black-box filtering pipeline for unsafe content. It first identifies a toxic subspace in the text embedding representation, then projects prompt tokens orthogonally to that subspace, suppressing unsafe semantics before they reach the model. To maintain image quality, SAFREE further applies self-validating filtering and latent re-attention during denoising. While effective at filtering NSFW content, the method is entirely training-free and external to the model, which again makes it easy to bypass in local deployments.

Receler (Huang et al., 2023) introduces lightweight eraser modules that are attached to the cross-attention layers of frozen DMs. Instead of updating the model weights, Receler trains these add-on erasers with two objectives: a locality regularization that restricts modifications to features aligned with the target concept, and an adversarial prompt loss that improves robustness to paraphrased or adversarially crafted prompts. This design enables concept suppression while preserving unrelated content more effectively than global steering methods. However, because the erasers are external components and the base model remains unchanged, the intervention is not persistent: users can simply remove the eraser modules to restore the original generation behavior. Furthermore, training erasers for each new concept still requires additional fine-tuning overhead, making the method less practical for multi-concept or sequential removal scenarios.

Overall, external methods avoid changing model weights, which makes them attractive when persistence is not required or when direct editing is impractical. However, because the base models remain unchanged, these interventions can be trivially removed, offering no lasting guarantee of concept removal. Some approaches operate purely at inference and add latency (SLD, SEOT, SAFREE), while others attach auxiliary modules that require additional training (SAeUron, Receler). In both cases, scalability to multi-concept or sequential removal is limited. By contrast, our method directly edits the model at the level of semantic features, providing persistent, interpretable, and generalizable concept removal across architectures.

A.2.2 INTERNAL METHODS

A variety of model-editing methods have been proposed for concept removal. These approaches fall into two categories. *Training-based methods* adapt model parameters through fine-tuning or re-optimization. *Closed-form methods* instead modify attention weights through constrained projections. While many of these methods achieve strong results in specific settings, we do not include them as baselines. They rely on costly retraining, assume diffusion-specific architectures, or fail to generalize to autoregressive models. We now describe the most prominent approaches in detail.

Forget-Me-Not (Zhang et al., 2024a) proposes attention re-steering to suppress target concepts. It modifies cross-attention maps in U-Net-based DMs by fine-tuning them such that attention weights corresponding to a concept of interest are minimized, while unrelated tokens are preserved. Although Forget-Me-Not is relatively lightweight compared to full retraining, it still requires model fine-tuning for every new concept, which limits scalability to multi-concept and sequential removal scenarios. More critically, the method is inherently tied to the cross-attention mechanisms of U-Net denoisers and cannot transfer to architectures without these components, such as Transformer-based DMs (e.g., SD3.5 (Esser et al., 2024b)) or image autoregressive models such as Infinity (Han et al., 2025).

Erased Stable Diffusion (ESD) (Gandikota et al., 2023) erases concepts by leveraging classifier-free guidance. The method retrains the diffusion backbone so that generations conditioned on a target concept are pushed toward the generations of an “empty” prompt, thereby erasing the target concept during denoising. While effective for certain DMs, ESD is tightly coupled to the classifier-free guidance mechanism and the iterative denoising schedule of U-Net-based pipelines. ESD requires retraining for each concept, which is computationally expensive and unsuitable for large-scale or sequential removal tasks.

EDiff (Wu et al., 2025) formulates concept erasure as a bi-level optimization problem. At the inner level, it perturbs latent features associated with the target concept, while at the outer level it minimizes reconstruction losses to preserve unrelated content. This process requires gradient-based fine-tuning of the diffusion backbone, with optimization closely tied to the denoising schedules of U-Net-based DMs. Although effective within this scope, the reliance on iterative denoising and the computational demands of bi-level optimization make EDiff impractical for broader deployment. It cannot be applied to autoregressive models, and its high per-concept training cost prevents scalable or sequential removal.

Safe-CLIP (Poppi et al., 2024) fine-tunes CLIP encoders to reduce sensitivity to unsafe or undesired concepts. It trains on synthetic quadruplet datasets, which pair safe and unsafe samples along with positive and negative augmentations, in order to redirect unsafe embeddings into safer regions of the representation space while preserving semantic structure. This approach, however, requires large-scale curated training data and costly retraining for every new concept. More importantly, it is only applicable to models that rely on CLIP as their text encoder, such as early versions of Stable

918 Diffusion. Modern SOTA DMs (e.g., SD3.5) and autoregressive models (e.g., Infinity) instead rely
 919 on encoders such as T5 in addition to CLIP, rendering Safe-CLIP incompatible with our setting.
 920

921 **EraseAnything** (Gao et al., 2025) extends training-based erasure to rectified flow and Flux-style
 922 models by formulating the task as a bi-level optimization problem. At the inner level, LoRA adapters
 923 are optimized to suppress the target concept using erasure losses similar to those of ESD along
 924 with attention suppression. At the outer level, a reverse self-contrastive loss enforces preservation
 925 of unrelated concepts, yielding a min–max formulation that balances removal and fidelity. While
 926 effective, EraseAnything requires gradient-based fine-tuning with LoRA modules for every new
 927 concept and is tied to the specific dynamics of rectified flow architectures, making it computationally
 928 expensive and unsuitable for autoregressive models.

929 **Minimalist Concept Erasure** (Zhang et al., 2025) formulates concept erasure as a direct optimiza-
 930 tion over generated outputs. The method minimizes the distributional distance between samples
 931 conditioned on a target concept and those from a neutral prompt, while simultaneously preserving
 932 unrelated content. Unlike prior approaches that intervene only at specific layers, Minimalist Concept
 933 Erasure backpropagates through the entire generation process. It further introduces learnable neuron
 934 masks that are optimized jointly with the model to suppress target concepts more robustly. Although
 935 effective, this approach requires end-to-end fine-tuning for each concept and is tied to diffusion-style
 936 generative dynamics, making it computationally costly and inapplicable to autoregressive models.

937 In summary, the above methods illustrate diverse strategies for training-based erasure but share
 938 key limitations that make them unsuitable as baselines for our study. Training-based approaches
 939 such as Forget-Me-Not, ESD, Safe-CLIP, EraseAnything, and Minimalist Concept Erasure require
 940 costly fine-tuning or retraining for each concept, which prevents scalable evaluation across multiple
 941 or sequentially added concepts. Some are further restricted to diffusion-specific architectures or
 942 text encoders, rendering them incompatible with modern DMs and autoregressive models such as
 943 Infinity. For these reasons, we do not include them as baselines in our evaluation. Instead, we focus
 944 on closed-form approaches, which directly edit cross-attention weights and are applicable across
 945 architectures without retraining.

946 A.2.3 CLOSED-FORM EDITING METHODS

947 Closed-form editing methods modify the linear projections inside cross-attention without any gradient-
 948 based retraining of the base model. They specify constraints for a set of edited concepts and a set of
 949 preserved concepts, then solve for new projection weights in closed form.

950 **UCE** (Gandikota et al., 2023) edits the key and value projection matrices W_k and W_v in cross-
 951 attention by enforcing linear constraints on specific text embeddings. Let $E = \{c_i\}$ be text embed-
 952 dings of concepts to edit and let $P = \{c_j\}$ be embeddings of concepts to preserve. For each $c_i \in E$,
 953 a destination embedding c_i^* is chosen and the desired target output is $v_i^* = W_{\text{old}} c_i^*$ for the projection
 954 under edit. UCE solves

$$955 \min_W \sum_{c_i \in E} \|W c_i - v_i^*\|_2^2 + \sum_{c_j \in P} \|W c_j - W_{\text{old}} c_j\|_2^2,$$

956 with closed-form solution

$$957 W = \left(\sum_{c_i \in E} v_i^* c_i^\top + \sum_{c_j \in P} W_{\text{old}} c_j c_j^\top \right) \left(\sum_{c_i \in E} c_i c_i^\top + \sum_{c_j \in P} c_j c_j^\top \right)^{-1}.$$

958 This update is applied to W_k and W_v in the chosen cross-attention layers. The method supports
 959 simultaneous multi-concept edits by stacking constraints for many c_i , and it ensures invertibility by
 960 augmenting the preservation set with canonical basis directions when needed. Erasure is implemented
 961 by choosing c_i^* that maps to a neutral destination such as a generic token. The same formulation
 962 also covers debiasing and moderation by suitable choices of c_i^* . UCE generalizes TIME, and it is
 963 applicable to any architecture that uses linear cross-attention projections.

964 **LOCOEDIT** (Basu et al., 2024a) performs localized cross-attention edits by restricting updates
 965 to the layers and heads most responsible for representing a target concept. These are identified by
 966 contrasting activations from concept prompts against neutral prompts, which highlights the parameters
 967 most sensitive to the concept. For the selected W_k and W_v matrices, LOCOEDIT solves a regularized

972 least-squares problem

$$973 \quad \min_W \sum_{c_i \in E} \|W c_i - v_i^*\|_2^2 + \lambda \|W - W_{\text{old}}\|_2^2,$$

974
975 with closed-form solution

$$976 \quad W = \left(\sum_{c_i \in E} v_i^* c_i^\top + \lambda W_{\text{old}} \right) \left(\sum_{c_i \in E} c_i c_i^\top + \lambda I \right)^{-1}.$$

977 Here, E is the set of embeddings for concepts to edit, v_i^* are their desired target projections, and
978 λ controls the strength of the update relative to the original weights W_{old} . By confining edits to
979 concept-relevant regions, LOCOEDIT reduces interference with unrelated concepts while preserving
980 training-free efficiency. Like UCE, it applies to any architecture with linear cross-attention projections
981 and requires no retraining.

982 **TIME** (Orgad et al., 2023) and **RECE** (Gong et al., 2024) are variants of this formulation. TIME
983 applies a global Frobenius-norm regularization to constrain the edits, while RECE combines the
984 constraints of TIME and UCE with additional embedding refinements. Both are less localized than
985 LOCOEDIT and do not introduce fundamentally new mechanisms.

986 **MACE** (Lu et al., 2024) extends UCE to large-scale concept removal by training LoRA adapters
987 for individual concepts and fusing them through a closed-form integration step. This hybrid de-
988 sign enables simultaneous erasure of up to 100 concepts but requires diffusion-specific denoising
989 schedules and per-concept LoRA training, making it computationally expensive and inapplicable to
990 autoregressive models.

991 In summary, closed-form editing methods offer efficient, retraining-free updates but vary in their
992 ability to isolate target concepts without harming unrelated features. We include UCE and LOCOEDIT
993 as baselines, since they are closed-form, architecture-agnostic, and applicable without retraining.
994 Other methods, such as TIME, RECE, and MACE, are not included as baselines due to either
995 redundancy, diffusion-specific assumptions, or reliance on additional training.

1000 B BASELINE SETUP

1001 B.1 UCE

1002 We implement UCE (Gandikota et al., 2023) following its original closed-form formulation. For each
1003 target concept, we construct the *edit set* E from text embeddings of the corresponding concept tokens.
1004 The *preservation set* P is sampled from unrelated tokens in the same vocabulary, ensuring that both
1005 object and style categories are represented. As in our method, the destination embedding c_i^* is set to
1006 the embedding of the empty token, such that edited projections for target concepts map to a neutral
1007 representation.

1008 The update is applied to the key and value projection matrices W_k and W_v in all cross-attention
1009 layers. Multi-concept removal is handled in two ways. For **simultaneous removal**, constraints for
1010 multiple concepts are stacked to compute a single joint update. For **sequential removal**, UCE is
1011 re-applied iteratively: after editing one concept, the updated weights are used as W_{old} for the next
1012 edit, accumulating changes across concepts.

1013 This setup is consistent across all models, including SD3.5, Flux, and Infinity, since the UCE
1014 formulation does not require architecture-specific retraining or model internals beyond cross-attention
1015 projections.

1016 B.2 LOCOEDIT

1017 LOCOEDIT (Basu et al., 2024a) localizes concept-relevant layers by contrasting concept and neutral
1018 prompts. This procedure is well-suited to U-Net-based DMs, where the text embeddings remain fixed
1019 across all cross attention layers during denoising. In more recent architectures such as SD3.5 (Esser
1020 et al., 2024a) and FLUX (Labs, 2024) however, text and image features are updated jointly inside the
1021 multimodal transformer blocks, so the assumption of static text embeddings no longer holds. As a
1022 result, LOCOEDIT’s original localization step is not directly applicable.

1026 To adapt the method, we replaced LOCOEDIT’s contrastive localization step with a zero-ablation
 1027 procedure. We sampled 100 random prompts from the MS-COCO training set (Lin et al., 2014). For
 1028 each prompt, we performed zero-ablation of the key and value projections across a sliding window
 1029 of three consecutive cross-attention layers and measured the resulting CLIP score. The influence of
 1030 each layer was quantified as the mean score drop over all prompts. We then selected the three layers
 1031 with the largest drop as the localized set used for editing. For SD3.5, this setup identified attention
 1032 layers 18, 19, and 20 as the most influential, while for Flux the most influential attention layers were
 1033 1, 2, and 3. These sets were then used in place of the original localization step for LOCOEDIT.

1034 C QUALITATIVE RESULTS

1035 We provide qualitative examples illustrating the effect of concept removal across models and baselines.
 1036 For each category, we report the prompts and seed used to generate the images.

1037 Representative generations are shown in Figures 4 and 5, demonstrating the suppression of target
 1038 styles and objects while retaining unrelated content.

1043 D EXTENDED INSIGHTS INTO OUR TRACE METHOD

1044 D.1 MODEL-ARCHITECTURE-DEPENDENT SUBTLETIES

1045 Across the considered architectures, text representations enter the generative backbone through two
 1046 possible network gates.

1047 The first, present in all models, is a linear projection layer that maps the text encoder outputs to the
 1048 model dimension.

1049 In SD3.5 and Flux, there is an additional gate given by an MLP applied to a pooled text representation.
 1050 Its output is combined with that of another MLP processing the noise schedule, and the sum is used
 1051 for modulation throughout the network. Despite operating on a pooled representation, we found it
 1052 necessary to intervene at this layer as well to ensure effective suppression of targeted concepts.

1053 In Infinity models, a special first token is constructed from a pooled embedding of the text representa-
 1054 tion. This token is only used at the very beginning of the model and is not essential for downstream
 1055 generation. We therefore omit it from our intervention, as experiments confirmed that transcoder
 1056 substitution at the standard projection layer is sufficient.

1061 D.2 TRANSCODER TRAINING DATA

1062 We use the curated dataset from Cywiński & Deja (2025). It contains 80 prompts describing each
 1063 of 20 objects. To these, we append the 10 selected styles (chosen based on LLaVA performance),
 1064 and additionally include a variant without style, yielding a total of 17,600 prompts. Each prompt is
 1065 tokenized, with padded and special tokens (EOS/BOS) removed since they do not convey semantic
 1066 meaning and would otherwise be overrepresented. The resulting token sequences are processed by
 1067 the projection layer, producing input–output pairs that form the training dataset. The exact number of
 1068 pairs varies depending on the tokenizer used by each model.

1069 For the modulation MLP, pooling collapses each prompt into a single representation. Using the
 1070 17,600 curated prompts would therefore yield only 17,600 data points, which we found insufficient
 1071 to capture the behavior of the MLP in SD3.5 and Flux. To address this, we instead sample 30,000
 1072 prompts from the MS-COCO 2014 training set (Lin et al., 2014). As above, we append style variants
 1073 and a no-style variant to each prompt, yielding a total of 330,000 prompts. Each pooled embedding is
 1074 processed to form one input–output pair, resulting in 330,000 training examples for the transcoder.

1077 E EXTENDED EXPERIMENTAL SETUP

1078 We present further details of the experimental setup.

1080
1081

E.1 TRANSCODER TRAINING DETAILS

1082
1083
1084
1085
1086

Hyperparameters. We set the sparsity level to $k = 32$ and the expansion factor to 16. Training uses the fused Adam optimizer with learning rate $\eta = 10^{-4}$, parameters $(\beta_1, \beta_2) = (0.9, 0.999)$, and stability constant $\epsilon = 6.25 \times 10^{-10}$. We apply an exponential moving average (EMA) of parameters with decay 0.999. Gradients are clipped to an ℓ_2 -norm of 1. We train each transcoder for 100 epochs, using a batch size of 16384 for projection-layer surrogates and 1024 for MLP-layer surrogates.

1087
1088
1089
1090
1091

Initialization. When the input and output dimensions are equal, the decoder matrix is initialized as the transpose of the encoder. If they differ, we perform a QR decomposition of the encoder weights to obtain an orthogonal basis. If the resulting basis is smaller than the output dimension, we augment it with gaussian random vectors, apply QR again, and then truncate or pad to initialize the decoder. This procedure ensures orthogonality and prevents degenerate initialization.

1092
1093
1094
1095

Efficiency. We implement sparse–dense multiplication using the Triton kernel of Gao et al. (2024), reducing the forward-pass cost by approximately a factor of two. Each forward pass therefore requires only one large dense matrix multiplication, making training tractable even with an expansion factor of 16.

1096
1097

All training runs were conducted on NVIDIA A100 GPUs with 40GB of memory.

1098
1099

E.2 TRANSCODER LOSS DETAILS

1100
1101

Let (\mathbf{x}, \mathbf{y}) denote an input–output pair of the original transformation. The encoder–decoder mapping is defined as

$$\begin{aligned} \mathbf{z} &= \text{TopK}(W_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}}), \\ \hat{\mathbf{y}} &= W_{\text{dec}}\mathbf{z} + \mathbf{b}_{\text{dec}}. \end{aligned}$$

1102
1103
1104

The composite training loss consists of three components:

1105
1106
1107

Fidelity loss. This term enforces alignment between the transcoder output and the target activations of the original transformation:

$$\mathcal{L}_{\text{fidelity}} = \frac{\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2}{\text{Var}(\mathbf{y})}. \quad (7)$$

1108

Multi-TopK loss. To promote a smoother and more progressive allocation of importance across latent units, the fidelity objective is recomputed with an expanded TopK budget, typically $4k$. Let $\mathbf{z}^{(4k)}$ denote the latent vector obtained under this setting:

1109
1110
1111
1112
1113

$$\mathcal{L}_{\text{multi-topK}} = \frac{\|W_{\text{dec}}\mathbf{z}^{(4k)} + \mathbf{b}_{\text{dec}} - \mathbf{y}\|_2^2}{\text{Var}(\mathbf{y})}. \quad (8)$$

1114

Auxiliary loss. To mitigate the problem of *dead latents*—units that fail to activate—we keep track of activations using the multi-TopK setting. A latent is declared dead if it has not fired in the last 10^7 forward passes. If at least one dead latent exists, an auxiliary loss is applied. Specifically, the fidelity objective is recomputed with a restricted TopK operator that is allowed to select only from the set of dead latents. The budget k_{aux} is set to the minimum between the number of dead latents and half of the input dimension. Let $\mathbf{z}^{(\text{aux})}$ denote the resulting latent vector. The auxiliary loss is then

1115
1116
1117
1118
1119
1120
1121
1122
1123

$$\mathcal{L}_{\text{aux}} = \frac{\|W_{\text{dec}}\mathbf{z}^{(\text{aux})} + \mathbf{b}_{\text{dec}} - \mathbf{y}\|_2^2}{\text{Var}(\mathbf{y})}. \quad (9)$$

1124
1125
1126

In summary, the composite loss in Eq. (4) balances fidelity to the original transformation, progressive code utilization, and prevention of dead latents, with $\alpha_1 = \frac{1}{8}$ and $\alpha_2 = \frac{1}{32}$ in all our experiments.

1127

E.3 TRANSCODER LOSS ABLATION

1128
1129
1130
1131
1132
1133

We conduct an ablation study on the loss components of our transcoder training. Specifically, we evaluate style unlearning performance on Infinity-2B under three configurations: **(i)** fidelity loss only,

1134 Table 5: **Ablation of loss components on Infinity-2B for class unlearning.** We report UA, IRA,
 1135 and CRA on the I2P benchmark, comparing different training configurations of our transcoder.
 1136

Setting	Style Removal			Mean (\uparrow)
	UA (\uparrow)	IRA (\uparrow)	CRA (\uparrow)	
Infinity-2B (Fidelity only)	88.60	25.40	59.87	57.96
Infinity-2B (Fidelity + Top- k)	86.80	26.57	74.72	62.69
Infinity-2B (Fidelity + Aux)	85.00	25.10	67.88	59.33
Infinity-2B (Full)	86.80	39.60	90.30	72.23

1143
 1144 (ii) fidelity + multi–Top- k loss, and (iii) fidelity + auxiliary loss. We do not evaluate the combination
 1145 of multi–Top- k and auxiliary loss, as it is effectively equivalent to the fidelity + auxiliary formulation
 1146 with a larger effective k . The results are reported in Table 5.
 1147

E.4 LLAVA CLASSIFIER

1150 To evaluate concept removal performance, we require classifiers that can detect whether a generated
 1151 image still exhibits the target concept. Following prior work, we adopt LLaVA-1.6-Vicuna-7B (Liu
 1152 et al., 2023) as a zero-shot classifier, but adapt its usage to our benchmark settings.
 1153

1154 **Style Classification.** For the style removal experiments, we prompt LLaVA with a fixed instruction
 1155 asking it to assign an image to exactly one artistic style from a predefined list. UnlearnCanvas
 1156 constructed a custom dataset by combining real images with stylization techniques and trained both
 1157 Stable Diffusion 1.5 and a dedicated style classifier on it. However, this setup is tailored to SD 1.5
 1158 and its classifier, making it unsuitable for evaluating other architectures such as SD 3.5, Flux, and
 1159 Infinity (Table 6). To ensure consistency across models, we instead employ LLaVA as a unified
 1160 zero-shot classifier. We reduce the style set from 50 (as in UnlearnCanvas) to 10 categories to balance
 1161 evaluation cost while maintaining coverage of diverse artistic modes. The exact LLaVA prompt used
 1162 for style classification is shown in Figure 6.
 1163

1164 **Object Classification.** For object removal experiments, we follow a similar procedure. LLaVA is
 1165 instructed to classify each generated image into exactly one of 20 object categories, corresponding to
 1166 the object set from UnlearnCanvas. This ensures a fair evaluation of whether removed objects still
 1167 appear in generated samples, while also verifying that unrelated objects are retained. The full prompt
 1168 for object classification is shown in Figure 7.
 1169

1170 **Evaluation.** For both style and object removal, LLaVA returns an integer corresponding to the
 1171 predicted class. Given a target concept c , we report:
 1172

- **Unlearning Accuracy (UA):** Fraction of images generated with prompts containing c that are not
 1173 classified as c .
- **In-domain Retain Accuracy (IRA):** Accuracy on non-target concepts from the same domain (e.g.,
 1174 other objects when unlearning one object).
- **Cross-domain Retain Accuracy (CRA):** Accuracy on concepts from a different domain (e.g.,
 1175 style accuracy when unlearning an object).

1176 This setup ensures that classification is consistent across all generative models considered, without
 1177 requiring domain-specific classifiers or retraining. Table 7 reports the classification accuracy of the
 1178 unmodified baseline models. Consistent with the findings of Zhang et al. (2025), we observe that
 1179 Flux struggles to render artistic styles accurately.
 1180

Model	Style Acc.	Object Acc.
Infinity 2B	38.8%	95.6%
Infinity 8B	63.2%	96.4%
SD 3.5	62.1%	94.3%
Flux	31.8%	95.3%

1181 Table 7: Style and object classification accuracies across models.
 1182

Table 6: Baseline performance of style and object classifiers across different models.

Model	Style Accuracy (%)	Class Accuracy (%)
SD 1.5 Finetuned	99.76	98.42
Flux	2.12	90.64
SD 3.5 Large	3.96	93.58
Infinity 2B	5.32	95.32
Infinity 8B	6.06	95.04

Table 8: Sequential concept removal for SD1.5(%).

N	ANT	SPM	MACE	TRACE
1	88.22	61.32	84.21	91.43
2	88.10	63.68	83.62	90.09
3	87.56	57.79	83.45	89.23
4	86.78	58.27	83.77	87.24
5	86.34	56.80	82.54	86.43
6	85.73	54.67	82.68	86.28
7	84.55	51.60	81.89	85.90
8	82.49	50.89	81.15	85.48
9	82.08	43.54	80.30	85.13
10	81.24	39.24	81.42	84.89

F ADDITIONAL EMPIRICAL RESULTS

F.1 MULTI-CONCEPT REMOVAL: OBJECTS

Across both sequential and simultaneous setups, our method consistently outperforms LOCOEDIT and UCE. In the sequential case, baselines degrade rapidly as more objects are removed. On SD3.5 and Infinity-8B, LOCOEDIT falls by more than 30 points as N increases, and UCE shows a steady decline as well. Flux exhibits a similar trend, with both baselines losing around 15–20 points by the tenth removal. In contrast, our method shows only a gradual decrease across all models and sustains average scores above 70% even at $N = 10$, demonstrating robustness to accumulated interventions.

In the simultaneous case, where all concepts are removed in one step, performance for all methods is initially higher, but the gap widens with increasing N . On Flux and Infinity-2B, our approach maintains almost flat performance curves, while LOCOEDIT and UCE steadily lose accuracy. On the larger models, SD3.5 and Infinity-8B, the baselines degrade most quickly, whereas our method remains stable above 75%.

These findings confirm that transcoders provide reliable multi-concept removal across both diffusion and autoregressive models, scaling to larger N without the sharp performance drops observed in prior methods.

F.2 MULTI-CONCEPT REMOVAL: SD1.5

The results in Table 8 show that ANT (Li et al., 2025) and MACE (Lu et al., 2024) remain strong across the sequence of removals on SD1.5. Both methods keep stable values for small and medium values of N and decline gradually as more styles are removed. SPM drops more quickly, which suggests that its updates accumulate across steps. TRACE achieves the highest values for all ten removals and shows only a mild reduction as N increases. The experiment removes the ten styles presented in Figure 6 in a fixed order: Van Gogh, Picasso, Cartoon, Cubism, Winter, Pop Art, Ukiyoe, Impressionism, Byzantine, and Bricks. Each step is evaluated using the average score ($UA + (IRA + CRA)/2\right)/2$, following Cywiński & Deja (2025). These results confirm that TRACE maintains stable behavior even when several styles are removed in sequence, while ANT and MACE provide strong baselines with slightly lower values.

1242

1243

Table 9: MMA Diffusion and UnlearnDiff Attack Success Rates (%).

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

Method	SD1.4		SD1.5		SD3.5		Flux	
	MMA	UnlearnDiff	MMA	UnlearnDiff	MMA	UnlearnDiff	MMA	UnlearnDiff
LOCOEDIT	81.44	84.56	79.21	87.67	60.03	61.22	28.63	23.32
UCE	49.67	59.28	53.32	63.21	55.44	52.76	29.91	26.11
Ours	36.21	27.77	41.76	37.81	37.21	34.69	18.89	12.32

Method	Armpits	Belly	Buttocks	Feet	Breasts (F)	Genitalia (F)	Breasts (M)	Genitalia (M)	Total	FID (\downarrow)
FMN	43	117	12	59	155	17	19	2	424	13.52
CA	153	180	45	66	298	22	67	7	838	16.25
AdvUn	8	0	0	13	1	1	0	0	28	17.18
Reeler	48	32	3	35	20	0	17	5	160	15.32
MACE	17	19	2	39	16	0	9	7	111	13.42
UCE	29	62	7	29	35	5	11	4	182	14.07
SLD-M	47	72	3	21	39	1	26	3	212	16.34
ESD-x	59	73	12	39	100	6	18	8	315	14.41
ESD-u	32	30	2	19	27	3	8	2	123	15.10
SAeUron	7	1	3	2	4	0	0	1	18	14.37
Ours	52	80	6	10	132	3	17	4	304	16.04
SD v1.4	148	170	29	63	266	18	42	7	743	14.04

Table 10: Nudity unlearning evaluation on the I2P benchmark. The best result for each metric is highlighted in bold.

F.3 ROBUSTNESS

To evaluate TRACE under a broader range of conditions, we include additional adversarial attack benchmarks based on MMA Diffusion (Yang et al., 2024a) and UnlearnDiff (Zhang et al., 2024c). The adversarial attack results in Table 9 show clear differences across methods and models. Our approach consistently reaches the lowest values for both MMA Diffusion and UnlearnDiff, which indicates stronger resistance to targeted attacks. LOCOEDIT produces higher scores across all models, and UCE shows a similar pattern, suggesting that both are more vulnerable when exposed to these attacks. The trend holds across SD1.4, SD1.5, SD3.5, and Flux, with the strongest gains appearing on Flux where the gap between our method and the baselines is the largest. These results show that our approach provides improved robustness under adversarial prompts across model sizes and architectures.

F.4 NSFW REMOVAL

To evaluate in a more practical scenario, we use the I2P benchmark (Schramowski et al., 2023) to assess nudity removal, employing NudeNet with a filtering factor of 0.6, consistent with prior work Appendix F.1. We additionally report the FID on 30k COCO 2014 validation images. For this experiment, we use SD-1.4 with an expansion factor of 32 and a TopK hyperparameter of 128.

F.5 ADDITIONAL QUALITY METRICS

To further assess the perceptual quality of edited images, we compute HPSv3 (Ma et al., 2025a) and Aesthetic v2.5² scores for all editing methods across the five model families evaluated in the main paper. For each model, we generate edited outputs using LOCOEDIT, UCE, and TRACE under the same prompts used in our core experiments. We also include a No Edit reference, representing the original model output with no concept removal applied.

HPSv3 is used to measure prompt conditioned human preference quality, capturing realism, semantic alignment, and overall perceptual fidelity. Aesthetic v2.5 evaluates only visual appeal and stylistic quality, independent of prompt correctness. Each metric is computed directly on the generated images without additional filtering or post processing. All models are run with their recommended inference settings to ensure consistency and fairness in comparison.

²<https://github.com/discus0434/aesthetic-predictor-v2-5>

Table 11 reports the full set of scores. TRACE consistently remains closest to the No Edit reference across both metrics and all architectures, indicating that it achieves concept removal while retaining perceptual quality more effectively than LOCOEDIT and UCE. This behavior is also reflected in the qualitative examples shown in Figure 4 of the appendix.

Table 11: HPSv3 and Aesthetic v2.5 scores for all editing methods across the five model families. TRACE remains closest to the No Edit reference on both metrics, indicating stronger preservation of perceptual and aesthetic quality.

Method	HPSv3					Aesthetic v2.5				
	SD1.5	SD3.5	FLUX	Infinity-8B	Infinity-2B	SD1.5	SD3.5	FLUX	Infinity-8B	Infinity-2B
LOCOEDIT	3.81	7.46	8.13	7.87	6.57	2.58	6.02	6.13	6.47	5.27
UCE	3.54	6.19	7.54	7.96	6.38	2.43	6.11	6.34	6.44	5.54
TRACE	4.32	8.31	8.49	8.54	6.98	3.07	6.15	6.49	6.51	5.61
No Edit	4.59	8.63	8.87	8.61	7.13	3.54	6.23	6.77	6.58	5.88

F.6 EFFICIENCY

We evaluate the efficiency of our method in Table 12. Memory denotes the peak increase over the baseline model during execution. Baseline peak inference memory (batch size 1 with classifier-free guidance) is 34.51 GB for Infinity-2B, 58.94 GB for Infinity-8B, 76.52 GB for Flux, and 64.54 GB for SD3.5. Storage refers to the additional disk space introduced by our method. The base model weights occupy 20.6 GB for Infinity-2B, 44 GB for Infinity-8B, 54 GB for Flux, and 67 GB for SD3.5. The upfront cost corresponds to the one-time expense of activation collection and transcoder training. The per-target unlearning time captures the additional cost of suppressing a new concept once the transcoder is trained. Finally, the per-image overhead reflects the extra inference time incurred when the intervention is active. All measurements were obtained on a single NVIDIA A100-SXM4-80GB GPU (CUDA 12.4, driver 550.90.07).

As shown in Table 12, our method incurs only a low computational cost across all models. The higher upfront cost for diffusion models compared to Infinity stems from activation collection at the MLP layer. Since this layer operates on pooled prompt representations, it requires substantially more training data and therefore many more prompt encodings. For reference, 16,800 prompts are used for the projection layer, whereas 330,000 prompts are required for the MLP layer, as described in Appendix D.2.

G LIMITATIONS

Our approach inherits several intrinsic limitations rooted in the capabilities of the underlying generative model. First, the method is constrained by the model’s internal representation of the target concept. If a concept is rare or not sufficiently represented in the activations used during transcoder training, the model may not have learned a disentangled or identifiable representation of it. In such cases, the transcoder cannot reliably suppress the concept, leading to partial or ineffective removal. This represents a fundamental failure mode: a concept cannot be unlearned if it was never meaningfully learned in the first place. We highlight these limitations to clarify the scope of applicability of our method and to motivate future work on broader concept generalization and disentanglement.

H THE USE OF LARGE LANGUAGE MODELS (LLMs)

Large language models (LLMs) were used in the preparation of this paper exclusively as a writing assistant. Specifically, they supported the reformulation of sentences, correction of grammar, and improvements in clarity and style. All technical contributions, research design, implementation, experiments, and analysis were conceived and carried out by the authors. The LLMs did not contribute to the generation of novel research ideas, experiments, or results.

1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1370
 1371
 1372

Model	Memory (GB)	Storage (GB)	Upfront Cost (s)	Per-target Unlearning (s)	Per-image Overhead (s)
SD3.5	2.4	2.15	6060	0.661	0.60
Flux	1.9	1.93	5428	1.02	0.43
Infinity-2B	0.67	0.512	648	0.323	0.17
Infinity-8B	0.75	0.704	753	0.317	0.20

1380
 1381 Table 12: **Efficiency comparison.** Memory and storage requirements, and time costs (in seconds)
 1382 split into upfront cost, per-target unlearning, and per-image overhead.
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1390
 1391
 1392
 1393
 1394
 1395
 1396
 1397
 1398
 1399
 1400
 1401
 1402
 1403

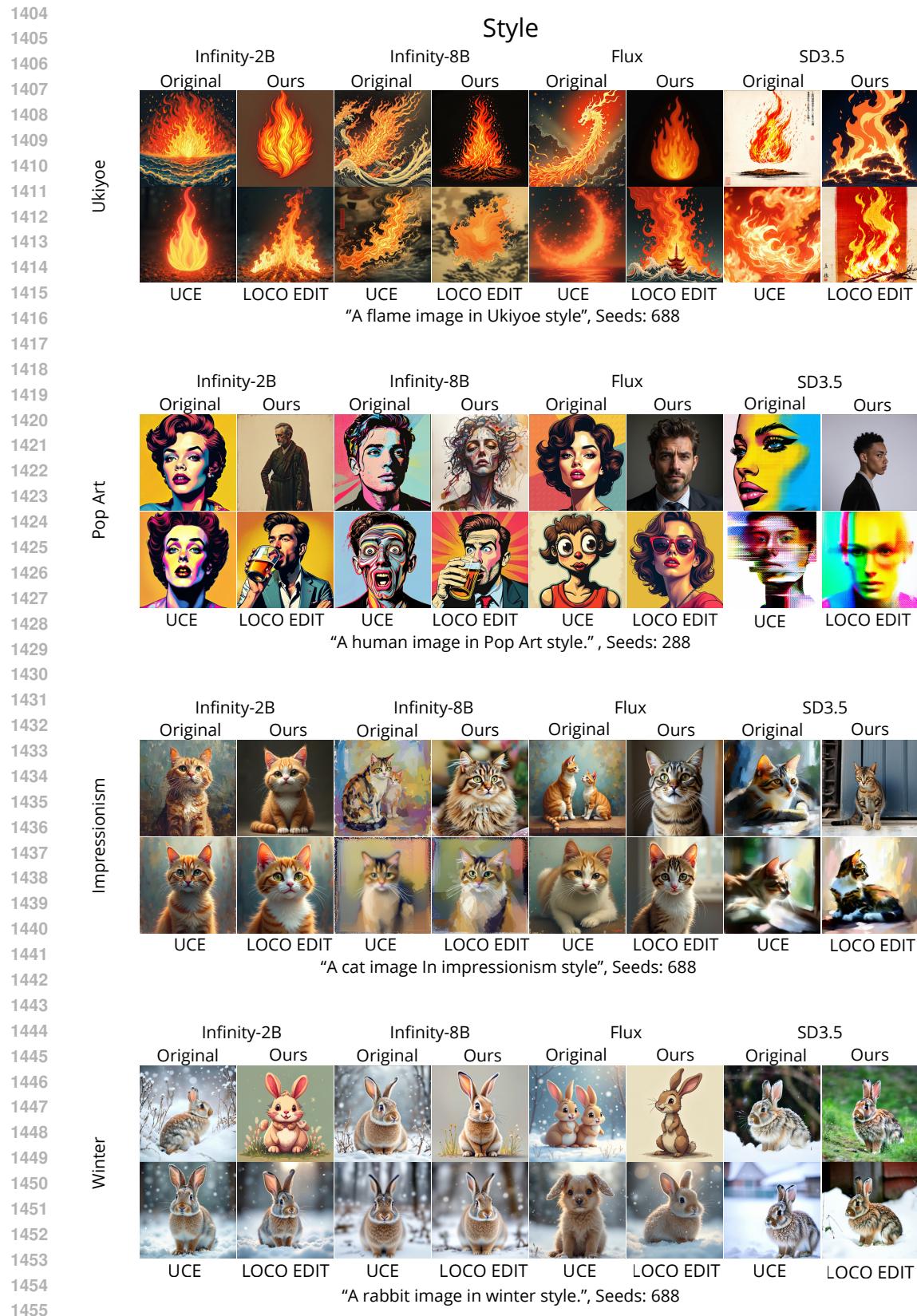


Figure 4: Qualitative results for style removal across models and baselines. Prompts and seeds are listed below each image.

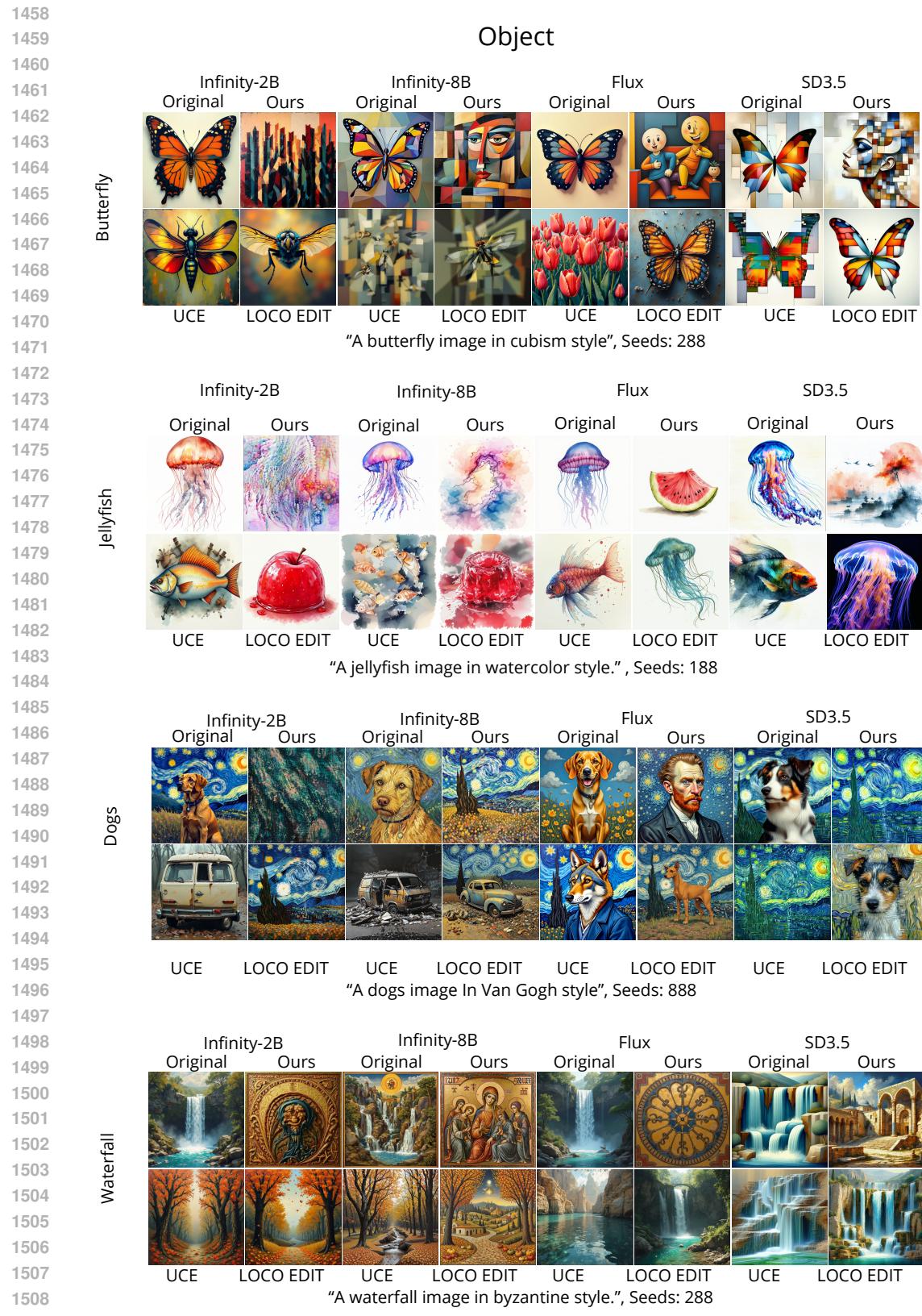


Figure 5: Qualitative results for object removal across models and baselines. Prompts and seeds are listed below each image.

1512

1513

1514

1515

1516

1517

1518

1519

1520

1521

1522

1523

1524

1525

You are an image classifier. Classify the artistic style of the given image.
Instruction: Choose exactly **one** option from the numbered list below. Respond with **only the number**.

Options:

- | | |
|-------------|------------------|
| 1. Van Gogh | 6. Pop Art |
| 2. Picasso | 7. Ukiyoe |
| 3. Cartoon | 8. Impressionism |
| 4. Cubism | 9. Byzantine |
| 5. Winter | 10. Bricks |

Figure 6: Prompt used for LLaVA-based style classification in our evaluation. The model must select exactly one label, ensuring consistent evaluation across generated samples.

Classify the object depicted in this image.
Choose exactly one option from the numbered list.
Respond with only the number.

Object categories:

- | | | |
|-----------------|---------------|----------------|
| 1. Architecture | 8. Flame | 15. Sandwich |
| 2. Bear | 9. Flowers | 16. Sea |
| 3. Bird | 10. Frog | 17. Statue |
| 4. Butterfly | 11. Horse | 18. Tower |
| 5. Cat | 12. Human | 19. Tree |
| 6. Dog | 13. Jellyfish | 20. Waterfalls |
| 7. Fish | 14. Rabbits | |

Figure 7: Prompt used for object classification in our LLaVA-based evaluation.

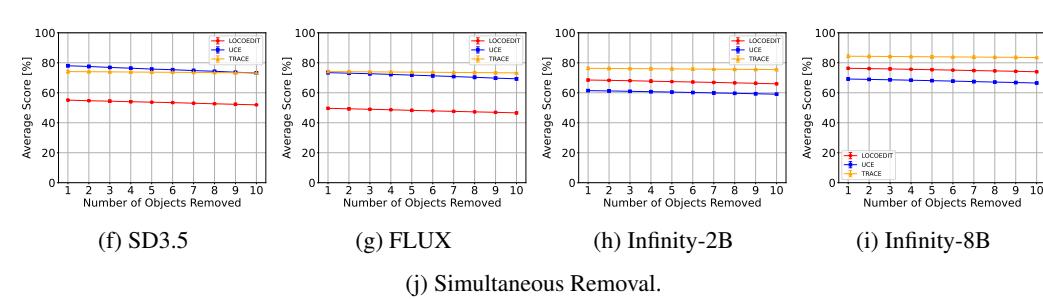
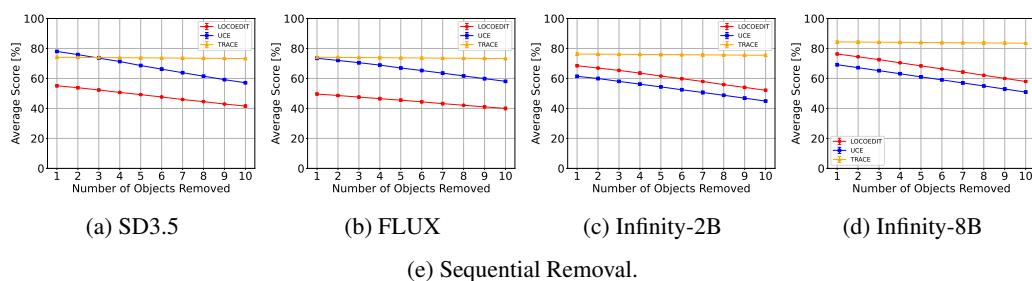


Figure 8: Multi-Concept Removal of Our Method and SOTA Baselines for Objects..