CASL: Concept-Aligned Sparse Latents for Interpreting Diffusion Models

Zhenghao He **Guangzhi Xiong** **Boyang Wang** **Sanchit Sinha** **Aidong Zhang**
Department of Computer Science, University of Virginia
{zhenghao, aidong}@virginia.edu

Abstract

Internal activations of diffusion models encode rich semantic information, but interpreting such representations remains challenging. While Sparse Autoencoders (SAEs) have shown promise in disentangling latent representations, existing SAE-based methods for diffusion model understanding rely on unsupervised approaches that fail to align sparse features with human-understandable concepts. This limits their ability to provide reliable semantic control over generated images. We introduce **CASL** (Concept-Aligned Sparse Latents), a supervised framework that aligns sparse latent dimensions of U-Net-based diffusion models with semantic concepts. We focus on unconditional diffusion models, where the h-space encodes stable and unconfounded semantic structure, providing a controlled setting for mechanistic interpretability analysis. CASL first trains an SAE on frozen U-Net activations to obtain disentangled latent representations, and then learns a lightweight linear mapping that associates each concept with a small set of relevant latent dimensions. To validate the semantic meaning of these aligned directions, we propose **CASL-Steer**, a controlled latent intervention that shifts activations along the learned concept axis. Unlike editing methods, CASL-Steer is used solely as a causal probe to reveal how concept-aligned latents influence generated content. We further introduce the **Editing Precision Ratio (EPR)**, a metric that jointly measures concept specificity and the preservation of unrelated attributes. Experiments on CelebA-HQ, FFHQ, LSUN-Church, and AFHQ-Dog demonstrate that concept-aligned sparse directions yield more precise and disentangled causal effects than existing activation-space editing methods. Classification probing further confirms that a compact subset of aligned latent units carries strong concept-specific signals. These results validate supervised alignment as an effective approach to interpreting diffusion model representations. Code and models are available at https://github.com/Zhenghao-He/CASL.

1 Introduction

Understanding how generative models internally represent semantic concepts is a central challenge in mechanistic interpretability. Diffusion models are a compelling subject for this inquiry: recent studies [21, 32, 15, 28] have revealed that their internal activations already exhibit rich semantic properties, with high-level concepts implicitly encoded within feature spaces. In NLP, Sparse Autoencoders (SAEs) have proven effective in uncovering interpretable features from internal activations [4], and recent works have begun adapting this approach to diffusion models [30, 13, 14, 26]. These efforts show promise in revealing meaningful structures within high-dimensional hidden states.

However, extending sparse interpretability to vision diffusion models remains challenging. Existing attempts rely on unsupervised SAEs that attribute semantics to each latent unit by identifying the input that maximally activates it. While this heuristic is effective in NLP, where discrete tokens naturally serve as semantic anchors, it breaks down in vision settings. Image representations are

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

continuous, spatially distributed, and heavily entangled, meaning that high activations often reflect a mixture of correlated attributes rather than a single interpretable concept. For instance, the notion of "smiling" co-occurs with changes in the mouth, cheeks, eyes, and global facial geometry, making it unclear which latent unit should be associated with the concept itself. As a result, existing unsupervised interpretation procedures struggle to reliably identify and verify concept-specific directions within vision SAE representations. The difficulty lies less in learning sparse factors than in *semantically identifying* them: without discrete anchors, it is unclear which latent corresponds to a predefined concept, limiting their ability to support meaningful interpretation or controlled intervention. As illustrated in Fig. 1, scaling an entangled semantic direction may strengthen the target concept while simultaneously amplifying correlated components, leading to unintended changes in other semantic attributes. This raises a more fundamental question: *is there a principled way to verify whether a discovered sparse feature truly corresponds to a human-understandable concept, rather than reflecting a statistical artifact or a mixture of correlated attributes?*

To overcome these limitations, we introduce CASL, a framework that learns concept-aligned sparse latents in U-Net-based diffusion models. We focus on unconditional diffusion models, where the bottleneck h-space provides stable and unconfounded semantic representations, offering a controlled setting for mechanistic interpretability analysis. In text-conditional models, the same h-space is heavily influenced by text conditioning, introducing additional confounding factors that complicate concept-level analysis. CASL first trains a Sparse Autoencoder on frozen U-Net activations to obtain a disentangled latent space. Then, instead of relying on unsupervised heuristics, we incorporate weak supervision to learn a lightweight linear mapping that associates each concept with its most relevant sparse dimensions. This yields a set of concept-aligned latent directions that are suitable for semantic interpretation. To verify their semantic effect, we further apply CASL-Steer, a controlled latent intervention that perturbs activations along the aligned direction to evaluate how each concept influences the model's generative behavior.

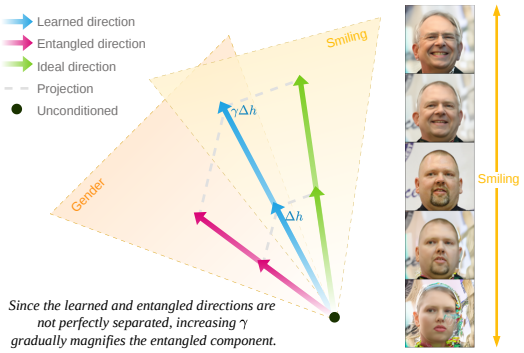


Figure 1: Demonstration of semantic directions in the activation space. Left: Geometric illustration of ideal, entangled, and learned semantic directions in the activation space. Right: scaling the Asyrp direction $\Delta h^{(\text{smile})}$ from $\gamma = -2$ to $\gamma = 2$ increases smiling intensity, but also introduces entangled changes in identity, hairstyle, and gender.

Our main contributions are summarized as follows:

- We introduce **CASL**, a supervised framework that aligns SAE latent dimensions in U-Net-based diffusion models with human-defined semantic concepts, enabling direct verification of whether sparse features encode interpretable semantics.
- We propose **CASL-Steer**, a controlled latent intervention used to validate the semantic effects of the aligned latent directions. CASL-Steer serves as a causal probe for interpretability rather than an image editing method.
- We design the **Editing Precision Ratio (EPR)**, a quantitative metric that jointly measures semantic specificity and the preservation of unrelated attributes, enabling systematic evaluation of concept alignment in sparse latent spaces.

We focus on U-Net-based architectures as a controlled testbed for validating our framework against established h-space methods, and show that CASL also transfers to text-conditioned SDXL and to a flow-matching DiT (Sec. 5.4, App. I).

2 Related Work

Intervening in Diffusion Representations. Recent work has shown that diffusion-model representations can be directly manipulated to alter semantic attributes of generated images. Methods such as Prompt-to-Prompt [9], Asyrp [21], and Concept Sliders [8] intervene in intermediate U-Net activations of frozen diffusion models to steer generation without full retraining. These approaches show

that semantic information resides in internal feature spaces and that localized activation perturbations can induce controllable output changes. However, their goal is primarily behavioral steering rather than understanding the semantic structure of the representation space. The manipulated dimensions remain largely uninterpreted, and their correspondence to human-interpretable concepts is unclear.

Mechanistic Interpretability in Diffusion Models. Sparse Autoencoders (SAEs) are powerful tools for uncovering sparse and partially disentangled features from frozen neural representations. In NLP, SAEs trained on transformer activations have revealed interpretable linguistic features such as sentiment, negation, and syntactic structures [4, 1]. Recent work has extended this paradigm to diffusion models. Surkov et al. [26] and Ijishakin et al. [14] train SAEs on residual or bottleneck activations and show that manipulating individual latent units can causally influence generation. Tinaz et al. [27] analyze the temporal evolution of interpretable concepts across diffusion timesteps, while Kim et al. [18] use k-sparse autoencoders to study semantic granularity across layers and timesteps. These studies show that interpretable features can emerge in diffusion representations, but most rely on unsupervised discovery and post-hoc interpretation. As a result, the learned features may capture entangled visual or temporal factors, and their alignment with predefined human concepts is not explicitly enforced.

Concept Alignment in Generative Representations. Although prior work extracts sparse and partially disentangled features from diffusion models, relatively little work explicitly aligns these latent dimensions with predefined semantic concepts. Without concept-level supervision, discovered features may not consistently represent stable semantic attributes, limiting interpretability and controllability. In contrast, our approach uses explicit semantic supervision to align sparse latent units with human-interpretable concepts, enforcing concept-level consistency and enabling faithful interpretation and structured control.

3 Problem Formulation

U-Net Activations (*h*-space). Following [21, 32, 30, 14], we similarly denote the U-Net intermediate activation (h-space) at a specific layer as $h \in \mathbb{R}^{C\times H\times W}$, where C, H, and W denote the number of channels, height, and width, respectively. Prior works [32, 21] have shown that h encodes rich semantic information and can be directly edited to control high-level concepts.

DDIM Sampling. We adopt the DDIM sampling schedule [24] for deterministic denoising, which enables applying controlled interventions in h-space without introducing stochastic variation. Kwon et al. [21] introduces a learnable shift Δh_t in the U-Net bottleneck, modifying only the P_t term of the DDIM update:

$$x_{t-1} = \sqrt{\alpha_{t-1}}\, P_t(\epsilon_\theta(x_t \mid \Delta h_t)) + D_t(\epsilon_\theta(x_t)). \tag{1}$$

This asymmetric activation-space injection enables controlled semantic manipulation without altering the overall generative trajectory.

Structured Latent Hypothesis. Empirical activation-space editing often exhibits entanglement: increasing the edit scale γ_c strengthens the target attribute but simultaneously alters identity, background, or lighting. We argue that this arises because the learned direction $\Delta h^{(c)}$ is not a pure concept vector, but instead a mixture of multiple semantic components. Small edits suppress these mixed factors, while larger edits amplify them (see Fig. 1).

To explain this behavior, we hypothesize that each concept direction $\Delta h^{(c)}$ admits a *sparse expansion* over a shared set of latent semantic bases. Specifically, there exist basis elements $\{b_i\}_{i=1}^{K} \subset \mathbb{R}^{C\times H\times W}$ such that

$$\Delta h^{(c)} = \sum_{i=1}^{K} \beta_i^{(c)}\, b_i, \qquad \|\boldsymbol{\beta}^{(c)}\|_0 \ll K. \tag{2}$$

Here, $\{b_i\}$ represent underlying semantic directions, and $\boldsymbol{\beta}^{(c)}$ encodes the concept's coordinates in this basis. Our goal is to learn this structured latent space and align selected basis directions with human-defined concepts.

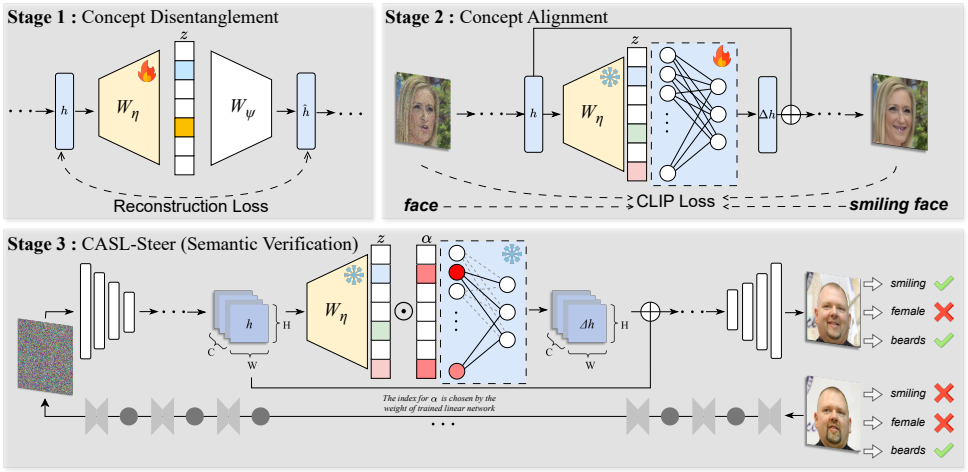


Figure 2: Overview of our proposed CASL framework. **Stage 1 (Concept Disentanglement):** A Sparse Autoencoder is trained on U-Net activations to obtain a structured sparse latent representation. **Stage 2 (Concept Alignment):** A lightweight linear mapping aligns selected latent dimensions with human-defined semantic concepts, producing concept-aligned directions. **Stage 3 (CASL-Steer):** A controlled latent intervention is applied along the aligned direction to *verify* its semantic effect, serving as a probing mechanism.

4 Method

Our framework consists of three main stages: (1) We train a sparse autoencoder (SAE) to encode U-Net activations into a structured and interpretable latent space (Sec. 4.1). (2) We align selected latent dimensions with human-defined concepts by learning a lightweight linear mapping under semantic supervision (Sec. 4.2). (3) We verify the semantic effect of the aligned directions through a controlled latent intervention in activation space (Sec. 4.3). In addition, we propose a metric to measure the effectiveness of aligning the latent representation with the semantic concepts (Sec. 4.4). Fig. 2 illustrates the main components of our framework and their relationships.

4.1 Concept Disentanglement (Stage 1)

Given an input activation $h \in \mathbb{R}^{C\times H\times W}$ from a diffusion model, we reshape h into $N = H \times W$ tokens of dimension C, i.e., $h_{\text{seq}} \in \mathbb{R}^{N\times C}$. Following [13, 26], each spatial location is treated as a token, and SAE maps h_{seq} to sparse latent $z \in \mathbb{R}^{N\times K}$, where $K \gg C$. The encoding and decoding processes are defined as:

$$z^{(t)} = \phi\left(W_\eta(h^{(t)} + e(t) - b_{\text{pre}}) + b_\eta\right) \tag{3}$$

$$\hat{h}^{(t)} = W_\psi z^{(t)} + b_{\text{pre}} \tag{4}$$

where $h^{(t)} \in \mathbb{R}^{N\times C}$ is the U-Net activation at timestep t, and $z^{(t)} \in \mathbb{R}^{N\times K}$ is its sparse latent representation. The term $e(t) \in \mathbb{R}^C$ is a learnable timestep embedding added to $h^{(t)}$ before encoding. The encoder is a linear transformation $W_\eta \in \mathbb{R}^{K\times C}$ without bias, followed by a latent bias $b_\eta \in \mathbb{R}^K$ and an element-wise activation function $\phi(\cdot)$. The decoder is also linear, $W_\psi \in \mathbb{R}^{C\times K}$, without bias. A learnable pre-bias vector $b_{\text{pre}} \in \mathbb{R}^C$ is subtracted from the input before encoding and added back after decoding to restore the original activation space.

The overall objective for training the SAE is:

$$\mathcal{L}_{\text{SAE}} = \underbrace{\|h - \hat{h}\|_2^2}_{\text{Reconstruction Loss}} + \underbrace{\lambda_{\text{sparse}}\|z\|_1}_{\text{Sparsity}} \tag{5}$$

where λ_{sparse} controls the strength of sparsity regularization.

4.2 Concept Alignment (Stage 2)

After training the SAE, we freeze its encoder W_η and perform inversion to obtain the disentangled latent representation $z = W_\eta(h)$ for each input activation h.

To align specific latent dimensions z_i with human-defined concepts (e.g., *smiling*), we introduce a lightweight linear mapping without any activation function that predicts the concept editing direction in the activation space:

$$\Delta h = W_\Delta z + b_\Delta \tag{6}$$

where $W_\Delta \in \mathbb{R}^{C\times K}$ and $b_\Delta \in \mathbb{R}^C$ are learnable parameters.

The edited activation is computed as $h' = h + \Delta h$, which is injected into the U-Net to compute the modified noise prediction. We then use this modified output in the DDIM inversion formula to obtain the edited image:

$$\hat{x}_0^{\text{edit}} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \sqrt{1-\alpha_t}\cdot \varepsilon_\theta(x_t \mid h + \Delta h)\right) \tag{7}$$

To ensure that Δh effectively induces the desired concept change (e.g., adding a smile), we follow [21] and combine the DiffusionCLIP loss [19] with an L1 regularization term:

$$\begin{aligned}\mathcal{L} = \lambda_{\text{CLIP}}\, \mathcal{L}_{\text{DiffusionCLIP}}(\hat{x}_0^{\text{edit}}, y^{\text{ref}}; x_0^{\text{origin}}, y^{\text{origin}}) \\ + \lambda_{\text{recon}}\, \|\hat{x}_0^{\text{edit}} - x_0^{\text{origin}}\|_1\end{aligned} \tag{8}$$

where $\hat{x}_0^{\text{edit}}$ denotes the generated image after applying the edited latent representation, and x_0^{origin} is the input image without editing. y^{ref} and y^{origin} are the corresponding CLIP text embeddings of the target and original semantic descriptions, respectively.

Through this process, the model learns to associate certain latent dimensions in z with explicit semantic concepts, enabling controllable and interpretable concept editing.

4.3 CASL-Steer: Semantic Verification (Stage 3)

Building on the concept alignment in Section 4.2, We aim to verify whether the learned sparse latent directions correspond to distinct, human-interpretable semantic concepts in the activation space, rather than focusing on image editing. To this end, we construct explicit mathematical formulations that enable principled and quantitative evaluation of concept alignment and disentanglement.

Firstly, we construct a sparse latent coordinate vector $\alpha \in \mathbb{R}^K$, where each entry specifies the traversal strength along a particular semantic direction. For a given concept c, we first select the top-k latent dimensions as:

$$\mathcal{I}_c = \text{TopK}_k\left(\left|W_\Delta^{(c)}\right|\right) \tag{9}$$

where $\left|W_\Delta^{(c)}\right|$ denotes the element-wise absolute value of the learned editing weights for concept c.

We then define the editing coordinate as

$$\alpha_i = \begin{cases}\alpha, & \text{if } i \in \mathcal{I}_c \\ 0, & \text{otherwise}\end{cases} \tag{10}$$

where α controls the editing intensity. This construction ensures that editing is only applied along concept-specific, disentangled semantic directions.

Given a disentangled semantic basis z for the current activation (obtained from the SAE encoder), the concept-wise editing direction in activation space is computed as:

$$\Delta h_c = W_\Delta\big(\underbrace{\alpha}_{\substack{\text{coordinate in}\\ \text{semantic basis}}} \odot \underbrace{z}_{\substack{\text{sparse}\\ \text{semantic basis}}}\big) \tag{11}$$

where $\odot$ denotes element-wise multiplication, z represents the semantic basis vectors in the activation space, and α specifies the editing coordinate in this basis. The resulting activation shift Δh_c is then added to the original activation, and the edited image $\hat{x}_0$ is obtained following the same DDIM inversion process as described in Section 4.2.

4.4 Editing Precision Ratio (EPR)

A desirable causal probing metric should induce significant changes in the target attribute while minimizing unintended modifications to other (non-target) attributes. Existing metrics either focus on edit strength alone (e.g., CLIP-Score) or global preservation (e.g., LPIPS, ArcFace). AugCLIP [20] constructs an ideal editing direction by minimizing the norm of the shift in CLIP space, but does not explicitly penalize correlated changes in non-target attributes, as CLIP directions can be entangled and non-orthogonal. To address this, we propose the *Editing Precision Ratio* (EPR), which formulates editing quality as a semantic signal-to-interference ratio: it directly measures per-attribute side effects and penalizes collateral change across all non-target attributes, providing a more fine-grained assessment of concept specificity.

Given N pairs of original and edited images $\{x_i, x'_i\}_{i=1}^N$, let $f_{\text{target}}(\cdot)$ denote the classifier score for the target attribute, and $f_j(\cdot)$ denote the classification score for the j-th non-target attribute (for $j = 1, \ldots, L$).

Average Target Attribute Change (Editing Effectiveness). This term captures how much the editing on the target attribute shifts the score of the target attribute:

$$\Delta_{\text{target}} = \frac{1}{N} \sum_{i=1}^{N} |f_{\text{target}}(x'_i) - f_{\text{target}}(x_i)| \tag{12}$$

Average Non-Target Attribute Change (Side Effects). This term reflects the average change across all unrelated attributes, measuring how much collateral change the edit induces on all the non-target attributes:

$$\Delta_{\text{non_target}} = \frac{1}{L} \sum_{j=1}^{L} \left(\frac{1}{N} \sum_{i=1}^{N} |f_j(x'_i) - f_j(x_i)| \right) \tag{13}$$

Editing Precision Ratio. The final score is defined as the ratio between the average target attribute change and the average non-target attribute change:

$$\text{EPR} = \frac{\Delta_{\text{target}}}{\Delta_{\text{non_target}} + \epsilon} \tag{14}$$

where ϵ is a small constant added for numerical stability.

A higher EPR indicates that the editing method achieves greater changes in the target attribute with less unintended change to non-target attributes, thus providing a quantitative measure of both editing effectiveness on the target attribute and side effects on the non target attributes. In practice, f_j denotes the classifier logits rather than probabilities, as logits preserve relative semantic strength without saturation. We set $\epsilon = 10^{-8}$ for numerical stability.

5 Experiments

5.1 Implementation Details

We evaluate CASL on multiple datasets, including FFHQ [17], CelebA-HQ [16], LSUN-Church [29] and AFHQ [3]. For each dataset, we adopt the corresponding diffusion backbone: DDPM++ [25] for FFHQ and CelebA-HQ, DDPM [11] for LSUN and iDDPM [23] for AFHQ. All diffusion backbones are official pretrained checkpoints and remain frozen. We uniformly sample 50 timesteps between $t = 999$ and $t = 500$ and extract the associated U-Net activations. The Sparse Autoencoder is trained on bottleneck activations, after which we learn a lightweight linear mapping for each concept using 1,000 real images. We evaluate SAE reconstruction quality and sparsity across expansion ratios $\gamma_{\text{sae}} \in \{16, 32, 64, 128, 256, 512\}$; full results are provided in Appendix D. We adopt $\gamma_{\text{sae}} = 128$ and $\lambda_{\text{sparse}} = 32$ as default based on this analysis. All experiments are run on 1 NVIDIA RTX A100 GPU (80GB memory). More implementation details can be found in Appendix C.

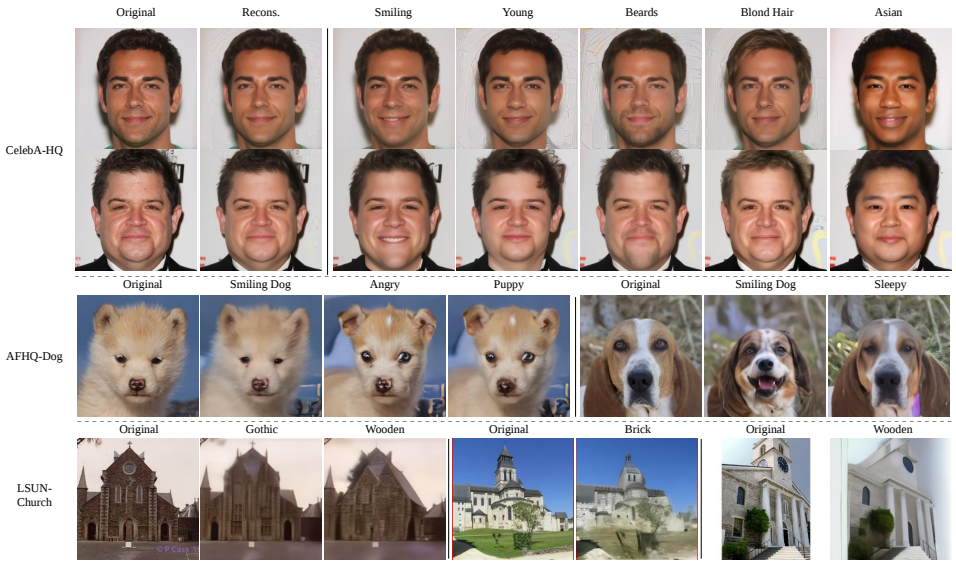


Figure 3: **Attribute editing results of concept-aligned editing.** Our method enables concept-aligned editing across diverse attributes and domains. Each column shows a specific attribute (*Smiling*, *Young*, *Puppy*, etc.) with consistent edits across different identities. Results are obtained by traversing sparse latent dimensions learned through supervised alignment.

5.2 Quantitative Evaluation

Our goal is not to introduce a new editing algorithm, but to use semantic editing as a **causal probe** for verifying whether a supervised sparse latent direction truly corresponds to a human-defined concept. We therefore compare CASL-Steer against representative attribute-editing methods that operate through activation-space or directional manipulations, including BoundaryDiffusion [32], Asyrp [21], Concept Slider [8], MasaCtrl [2], and SwiftEdit [22]. We report four metrics commonly used in facial attribute editing: CLIP-Score [10] for semantic alignment, LPIPS [31] for perceptual distortion, ArcFace similarity [6] for identity preservation, and our proposed EPR (Sec. 4.4) for measuring the trade-off between attribute strength and unintended changes.

Table 1 shows results on CelebA-HQ. For each attribute, we randomly sample 32 images from the test split and use only the top-1 aligned latent dimension from CASL. Images are not filtered or ranked ("top-1" refers only to the latent), and the same images are used for all methods (160 paired edits per method). CASL-Steer applies the concept-aligned shift during the selected DDIM denoising steps, while all baseline methods follow their standard configurations. Additional results with more datasets and a broader range of concepts are provided in the Table 20, Table 21, and Table 22 (Appendix).

The reconstruction baseline passes the input through DDIM inversion and bottleneck reconstruction without applying any concept direction. Its non-zero EPR reflects attribute drift from reconstruction alone, providing a reference for separating inversion artifacts from concept-specific effects.

CASL-Steer achieves the highest EPR across all evaluated attributes while maintaining competitive CLIP-Score, lower LPIPS, and stronger ArcFace similarity. This indicates that the aligned sparse directions induce the target semantic change with fewer collateral modifications than activation-space or black-box editing baselines, especially for entangled attributes such as Young and Blond Hair. This conclusion is robust: CASL-Steer keeps the highest average EPR when EPR is averaged over four classifier backbones (App. F.2), improves over Asyrp on AugCLIP [20], which also penalizes background drift, on all five concepts (App. G.2), and is not surpassed by the baselines at other strengths (App. C.11).

Figure 3 qualitatively validates the semantic effect of CASL-aligned latent directions. Across CelebA-HQ, AFHQ-Dog, and LSUN-Church, the selected directions induce the intended attributes, such as Smiling, Young, Puppy, Gothic, and Wooden, while largely preserving identity, viewpoint, geometry, and background. These results suggest that CASL captures concept-specific factors rather than broad correlated changes. Additional examples are provided in the Appendix.

Table 1: Our explainable editing framework achieves superior attribute manipulation performance without degrading quality or identity consistency. We evaluate five facial attributes using EPR, CLIP-Score, LPIPS, and ArcFace across all methods. We highlight the best value in **green** with **bold text** and the second best value in red.

Concept	Metric	Recon.	Editing Methods					Explainable
			Boundary [32]	Asyrp [21]	Slider [8]	MasaCtrl [2]	SwiftEdit [22]	CASL-Steer
Smiling	EPR (↑)	0.679±0.289	1.231±0.976	3.199±2.182	0.949±0.479	1.685±1.375	3.359±2.064	**4.465**±2.319
	CLIP-Score [10] (↑)	0.126±0.014	0.166±0.012	0.192±0.014	0.178±0.010	**0.204**±0.013	0.171±0.015	0.192±0.012
	LPIPS [31] (↓)	0.169±0.057	0.251±0.049	0.358±0.086	0.487±0.076	0.678±0.246	0.251±0.052	**0.243**±0.085
	ArcFace [6] (↑)	0.943±0.037	0.564±0.144	0.238±0.144	0.540±0.061	0.193±0.184	0.358±0.123	**0.566**±0.210
Big Nose	EPR (↑)	0.207±0.280	0.882±0.539	0.938±0.608	1.035±0.775	0.793±0.600	0.867±0.627	**1.060**±0.608
	CLIP-Score (↑)	0.140±0.019	0.148±0.014	0.173±0.017	0.168±0.012	**0.200**±0.029	0.157±0.014	0.159±0.012
	LPIPS (↓)	0.169±0.057	0.270±0.055	0.275±0.082	0.546±0.078	0.699±0.231	0.208±0.048	**0.191**±0.050
	ArcFace (↑)	0.943±0.037	0.545±0.112	0.463±0.182	0.487±0.084	0.200±0.191	0.412±0.121	**0.761**±0.059
Young	EPR (↑)	0.773±0.619	1.411±0.909	1.685±1.220	0.937±0.770	1.427±0.709	1.536±0.989	**1.817**±1.117
	CLIP-Score (↑)	0.113±0.013	0.180±0.014	**0.211**±0.014	0.184±0.013	0.204±0.017	0.184±0.013	0.206±0.013
	LPIPS (↓)	0.169±0.057	**0.258**±0.065	0.343±0.082	0.567±0.082	0.678±0.246	0.309±0.052	0.305±0.082
	ArcFace (↑)	0.943±0.037	0.546±0.139	0.306±0.139	**0.646**±0.095	0.266±0.191	0.357±0.128	0.387±0.142
Beards	EPR (↑)	0.763±0.669	1.165±0.974	1.739±1.268	0.919±0.383	2.059±1.541	1.255±0.884	**2.161**±1.080
	CLIP-Score (↑)	0.112±0.033	0.134±0.038	0.206±0.047	0.132±0.032	0.204±0.022	0.141±0.031	**0.211**±0.043
	LPIPS (↓)	0.169±0.057	0.274±0.054	0.353±0.090	0.490±0.073	0.741±0.216	0.257±0.053	**0.233**±0.071
	ArcFace (↑)	0.943±0.037	0.550±0.116	0.294±0.164	0.542±0.058	0.171±0.191	0.422±0.136	**0.562**±0.260
Blond Hair	EPR (↑)	0.110±0.032	0.876±0.794	1.173±1.002	0.760±0.548	1.173±0.678	1.302±1.014	**1.386**±1.213
	CLIP-Score (↑)	0.143±0.029	0.153±0.030	**0.186**±0.039	0.143±0.027	0.185±0.021	0.154±0.033	0.170±0.036
	LPIPS (↓)	0.169±0.057	0.262±0.052	0.303±0.082	0.491±0.076	0.713±0.208	0.227±0.103	**0.221**±0.072
	ArcFace (↑)	0.943±0.037	0.547±0.117	0.409±0.210	0.544±0.053	0.169±0.128	0.387±0.141	**0.686**±0.155

5.3 Empirical Characterization of Concept-Associated Latents

To distinguish sparse representation learning from semantic alignment, we examine whether the SAE dimensions selected by Stage 2 already exhibit concept-relevant activation patterns in the frozen SAE space; since the SAE is frozen before Stage 2, the CLIP-supervised mapping only selects and names existing factors.

For each concept-associated latent dimension, we randomly sample 1000 images from the evaluation split of FFHQ, AFHQ, and Lsun-Church datasets and compute a scalar activation score per image. Let $z^{(t)} \in \mathbb{R}^{N\times K}$ denote the SAE latent at timestep t, where $N = H \times W$ is the number of spatial tokens. For a fixed latent dimension k, we define the spatial activation at timestep t as

$$a_k^{(t)}(x) = \max_{i\in\{1,\dots,N\}} z_{i,k}^{(t)}(x). \tag{15}$$

We adopt spatial max-pooling to capture the peak activation across spatial locations, as mean-pooling may dilute localized concept signals. We then average activations across intervention timesteps $S = \{999, 949, \dots, 549\}$.

$$a_k(x) = \frac{1}{|S|}\sum_{t\in S} \max_{i\in\{1,\dots,N\}} z_{i,k}^{(t)}(x). \tag{16}$$

Images are ranked by activation magnitude, and we obtain the top-50 highest-activating samples. Fig. 4 shows a subset of these highest-activating examples for visualization purposes.

As shown in Fig. 4, the top-ranked samples exhibit clear semantic consistency with the aligned concept across datasets. Notably, the SAE itself is trained without any semantic supervision; CLIP is used only in Stage 2 to identify which pre-existing sparse dimensions are most relevant to each concept. The semantic coherence observed here therefore reflects structure already present in the unsupervised sparse latent space, rather than an artifact introduced by the CLIP supervision signal.

5.4 Concept Alignment Evaluation

To evaluate whether CASL successfully aligns specific latent dimensions z_i to concepts, we perform a classification-based probing experiment. For each concept c, we select the top-k latent units according to $\mathcal{I}_c$ (Eq. 9) and train a linear SVM classifier using these k-dimensional features.

As is shown in Table 3, the experiments use $\gamma_{\text{sae}} = 128$ and $\lambda_{\text{sparse}} = 32$, resulting in a high-dimensional latent space of $512 \times 128 = 65,536$ dimensions. The SVM classifiers are trained and tested on balanced datasets containing 1,000 positive and 1,000 neg-

ative samples each. During the inversion process, activation values $h^{(t)}$ are uniformly sampled at 50 time steps between $t = 999$ and $t = 500$, which serve as inputs to the classifiers. The dataset is split into training and testing sets with an 8:2 ratio.

Results. Even the single highest-ranked latent unit (top-1) achieves accuracy vastly above the random baseline (50%), confirming that individual CASL-aligned dimensions carry strong concept-specific signals. Increasing k further improves accuracy, indicating that additional aligned units contribute complementary semantic cues. Using only 16 latent units per concept, all classifiers approach near-perfect accuracy (> 0.93). We interpret these results narrowly as evidence of *attribute predictability*: they show that the selected latents carry information about the target concept, but, because CelebA attributes are strongly correlated, classification accuracy alone is not evidence of disentanglement. The same latent sets are substantially more predictive of their target than of non-target attributes (average target/non-target AUROC $0.852/0.741$; App. D.2). We therefore assess representation disentanglement with dedicated metrics below, and use EPR only for the selectivity of actual edits.



Figure 4: Images with Highest Aggregated Activation over Intervention Timesteps for Sparse Latent Dimensions (Subset of Top-50)

Label-free disentanglement (OMES). EPR depends on concept labels and on the capacity of the attribute classifier (App. A). We therefore complement it with OMES [5], which measures the modularity (OS) and compactness (MES) of a representation from controlled image pairs, without per-image annotations or trained classifiers. On 4,096 CelebA-HQ images, choosing k latents per factor by Stage-2 weights (CASL) gives the highest OMES at every budget, compared with *Random* and *High-act.* (largest mean activation) selection (Table 2(a)). LSUN-Church results, where EPR is unavailable, and label-based DCI [7], where CASL is also best, are in App. D.1.

Table 2: (a) OMES on CelebA-HQ (higher is better) for k latents per factor. (b) SAE-only ablation on CelebA-HQ: the same frozen Stage-1 SAE without Stage-2 alignment, intervening on the latent with the largest positive/negative activation difference.

(a) OMES ↑

k	Random	High-act.	CASL
1	0.536	0.489	**0.621**
2	0.469	0.491	**0.515**
4	0.421	0.438	**0.491**
8	0.392	0.414	**0.445**
16	0.377	0.400	**0.430**

(b) SAE-only ablation

Method	Δ_{tgt} ↑	$\Delta_{\text{non-tgt}}$ ↓	EPR↑
Asyrp	0.167	0.126	1.702
SAE-only	0.002	0.069	0.666
CASL	**0.302**	0.155	**1.832**

Is alignment necessary? CASL builds on h-space SAEs [14] and activation-space editing [21], but aligns SAE latents with *predefined* concepts instead of interpreting unlabeled features post hoc. Table 2(b) shows that alignment is essential: selecting a latent from the same frozen SAE by activation difference yields almost no target change (EPR 0.666 vs. 1.832).

Beyond pixel-space U-Nets. On the frozen text-conditioned SDXL latent U-Net (mid-block, new SAE, no fine-tuning), CASL improves Smiling EPR from 2.354 to 4.016 over a dense h-space intervention at matched target strength while halving LPIPS ($0.113 \rightarrow 0.060$). On a frozen flow-matching DiT-L/2, intervening in the pre-patchification u-space [12] with Euler sampling yields positive target changes for all five concepts (macro EPR 1.466). CASL is thus not specific to pixel-space U-Nets or DDIM, although a new backbone requires a suitable intervention space (App. I).

5.5 Hyperparameter Analysis

To better understand the behavior and robustness of our method, we conduct ablation studies of two key hyperparameters: the number of top-k latent dimensions for editing (Eq. 9), and the editing intensity coefficient α (Eq. 11).

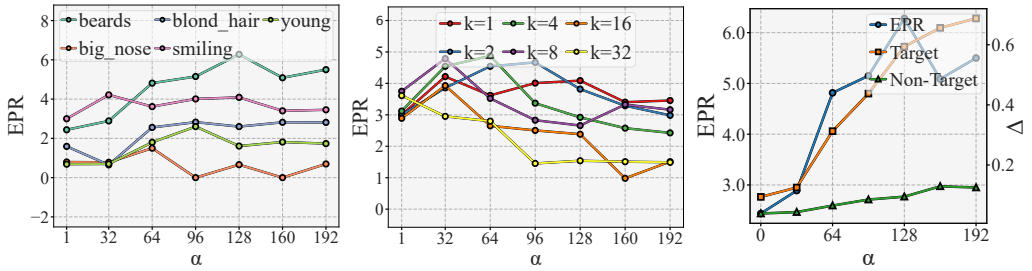
Our results are shown in Fig 5.**(a)**. When only the top-1 latent dimension is edited (Fig. 5a), EPR remains stable across a wide range of α for most concepts. This indicates that our SAE successfully extracts disentangled and interpretable latent factors, amplifying a single direction increases the target attribute without introducing unrelated concepts, even under strong editing.

Table 3: SVM classification accuracy for top-k latent dimensions across concepts.

Concepts	top-1	top-2	top-4	top-8	top-16
Smiling	0.671	0.755	0.851	0.948	0.993
Big Nose	0.675	0.694	0.736	0.771	0.818
Young	0.655	0.702	0.757	0.797	0.855
No Beard	0.757	0.825	0.851	0.891	0.929
Blond Hair	0.677	0.750	0.843	0.883	0.938

(b) For the *smiling* attribute (Fig. 5b), we observe that EPR is robust to changes in α only for top-k=1. When $k > 1$, EPR decreases significantly as α increases, suggesting that editing multiple latent dimensions simultaneously inevitably entangles additional semantic information and reduces editing precision. **(c)** Figure 5c further decomposes the editing behavior for *beards* ($k = 1$). As α increases, both the Δ_{target} (Eq. 12) and $\Delta_{\text{non_target}}$ (Eq. 13) changes rise, but EPR remains nearly constant. This implies that, although overall attribute changes are magnified by stronger editing, the *relative* specificity of the edit (i.e., the ratio between target and side effect) is well preserved due to the high semantic purity of the selected latent dimensions.

Overall, our method yields purified and accurately aligned latent features, allowing precise semantic editing with minimal change. Editing more latent dimensions, however, introduces entanglement and diminishes editing effectiveness. More results can be found in Appendix H.



(a) EPR vs α (Concepts) (b) EPR vs α (Top-k) (c) EPR on beards (Top-k=1)

Figure 5: Hyperparameter analysis of editing intensity α and top-k dimension selection. (a) Single-dimension editing shows stable EPR across concepts. (b) Multi-dimension editing reduces precision with increasing α. (c) Target and non-target changes scale proportionally for $k = 1$.

6 Conclusion

We introduced CASL, a supervised framework for aligning sparse latent dimensions in U-Net diffusion models with human-defined concepts. By combining SAE-based representation learning, concept-level alignment, and causal latent intervention, CASL exposes compact concept-relevant directions in diffusion h-space. We further proposed EPR to quantify the trade-off between target semantic change and unintended attribute shifts. Experiments across face, animal, and scene domains show that CASL yields more precise and interpretable concept directions than existing activation-space editing methods. While we focus on unconditional U-Net diffusion models as a controlled testbed, additional results on SDXL and a flow-matching DiT show that the mechanism extends to large-scale latent diffusion and to flow-matching objectives.

Acknowledgments and Disclosure of Funding

This work is supported in part by the US National Science Foundation (NSF) and National Institutes of Health (NIH) under grants 1R01HL184168-01, 2530655, 2616033, 2538206, 2529378, 2217071, and 2213700. Any recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NIH or NSF.

References

[1] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, and Adam et al. Jermyn. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL `https://transformer-circuits.pub/2023/monosemantic-features/index.html`.

[2] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22560–22570, 2023.

[3] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

[4] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[5] Jacopo Dapueto, Nicoletta Noceti, and Francesca Odone. Transferring disentangled representations: bridging the gap between synthetic and real images. In *Advances in Neural Information Processing Systems*, 2024.

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[7] Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

[8] Rohit Gandikota, Joanna Materzyńska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adaptors for precise control in diffusion models. In *European Conference on Computer Vision*, pages 172–188. Springer, 2024.

[9] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control.(2022). *URL https://arxiv.org/abs/2208.01626*, 3, 2022.

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

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[12] Vincent Tao Hu, David W. Zhang, Pascal Mettes, Meng Tang, Deli Zhao, and Cees G. M. Snoek. Latent space editing in transformer-based flow matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[13] Victor Shea-Jay Huang, Le Zhuo, Yi Xin, Zhaokai Wang, Peng Gao, and Hongsheng Li. Tide: Temporal-aware sparse autoencoders for interpretable diffusion transformers in image generation. *arXiv preprint arXiv:2503.07050*, 2025.

[14] Ayodeji Ijishakin, Ming Liang Ang, Levente Baljer, Daniel Chee Hian Tan, Hugo Laurence Fry, Ahmed Abdulaal, Aengus Lynch, and James H Cole. H-space sparse autoencoders. In *Neurips Safe Generative AI Workshop 2024*, 2024.

[15] Jaeseok Jeong, Mingi Kwon, and Youngjung Uh. Training-free content injection using h-space in diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5151–5161, 2024.

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[18] Dahye Kim, Xavier Thomas, and Deepti Ghadiyaram. Interpreting and leveraging semantic information in diffusion models. *CoRR*, 2024.

[19] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2426–2435, 2022.

[20] Yoonjeon Kim, Soohyun Ryu, Yeonsung Jung, Hyunkoo Lee, Joowon Kim, June Yong Yang, Jaeryong Hwang, and Eunho Yang. Preserve or modify? context-aware evaluation for balancing preservation and modification in text-guided image editing, 2025. URL `https://arxiv.org/abs/2410.11374`.

[21] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. *arXiv preprint arXiv:2210.10960*, 2022.

[22] Trong-Tung Nguyen, Quang Nguyen, Khoi Nguyen, Anh Tran, and Cuong Pham. Swiftedit: Lightning fast text-guided image editing via one-step diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 21492–21501, June 2025.

[23] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[24] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL `https://arxiv.org/abs/2010.02502`.

[25] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[26] Viacheslav Surkov, Chris Wendler, Mikhail Terekhov, Justin Deschenaux, Robert West, and Caglar Gulcehre. Unpacking sdxl turbo: Interpreting text-to-image models with sparse autoencoders. In *Mechanistic Interpretability for Vision at CVPR 2025 (Non-proceedings Track)*, 2025.

[27] Berk Tinaz, Zalan Fabian, and Mahdi Soltanolkotabi. Emergence and evolution of interpretable concepts in diffusion models, 2026. URL `https://arxiv.org/abs/2504.15473`.

[28] Zizheng Yang, Hu Yu, Bing Li, Jinghao Zhang, Jie Huang, and Feng Zhao. Unleashing the potential of the semantic latent space in diffusion models for image dehazing. In *European Conference on Computer Vision*, pages 371–389. Springer, 2024.

[29] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[30] Zhenbo Yu, Jian Jin, Jinhan Zhao, Zhenyong Fu, and Jian Yang. Ttfdiffusion: Training-free and text-free image editing in diffusion models with structural and semantic disentanglement. *Neurocomputing*, 619:129159, 2025.

[31] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[32] Ye Zhu, Yu Wu, Zhiwei Deng, Olga Russakovsky, and Yan Yan. Boundary guided learning-free semantic control with diffusion models. *Advances in Neural Information Processing Systems*, 36:78319–78346, 2023.

A Limitations

Our work focuses on concept-aligned sparse latents in U-Net-based diffusion models, using unconditional generation as a controlled setting for studying internal semantic structure. This design helps isolate representation-level effects, but it does not cover all sources of variation in modern text-conditional diffusion systems, where prompt conditioning and cross-attention may introduce additional semantic factors. Our additional experiments on SDXL and a flow-matching DiT (App. I) show that CASL transfers to large-scale text-conditioned latent diffusion and to flow-matching objectives, but these results cover fewer concepts, and applying CASL to a new generator still requires choosing an appropriate intervention space and propagation rule; CASL is therefore not fully generator-independent.

Our experiments are conducted on several commonly used visual domains, including faces, animals, and churches. These domains provide clear semantic attributes and allow controlled evaluation of concept alignment. Future work could further test CASL on more diverse and compositional concepts, such as object relations, scene-level attributes, and fine-grained spatial properties.

Finally, our quantitative evaluation relies on pretrained attribute classifiers and CLIP-based semantic signals. These tools provide practical proxies for measuring target and non-target semantic changes, but they may not capture all perceptual or human-preferred aspects of concept editing. Complementing these automatic metrics with human evaluation or stronger concept detectors could provide a more complete assessment. In particular, EPR requires per-attribute annotations and its absolute values depend on the classifier (App. F.2); we mitigate this with classifier-free OMES (App. D.1), AugCLIP (App. G.2), and an aggregated user study (App. E). We also observe two recurring failure modes. First, because the intervention acts on the global U-Net bottleneck, an edit can occasionally change the background or global color statistics together with the target attribute, a change that EPR does not penalize (we therefore also report AugCLIP, App. G.2). Second, at large α the edit can degrade identity or image fidelity, consistent with the monotonic increase of LPIPS with α (Table 7); this motivates selecting α under a fixed preservation budget (App. C.11).

B Broader Impact

This work contributes to the interpretability of generative models by providing a way to identify and verify concept-aligned sparse latent directions in diffusion representations. A better understanding of how semantic concepts are represented internally may help improve model auditing, debugging, and controllable generation, especially when undesired entanglement between target and non-target attributes needs to be diagnosed.

At the same time, methods that improve semantic control over generated images may also be misused for deceptive or harmful image manipulation. Our work is intended as an interpretability and causal analysis framework rather than a deployed editing system, and we evaluate it on existing public datasets and pretrained models. We do not introduce a new large-scale generative model or a new dataset of sensitive content. Future applications of concept-level intervention methods should therefore consider appropriate safeguards, provenance tracking, and responsible-use constraints when used in practical image editing systems.

C Implementation Details

C.1 Code

We provide our code in the supplementary material. Please check.

C.2 Model and Architecture Settings

SAE Training Details. We train the sparse autoencoder (SAE) on cached U-Net bottleneck activations extracted at a fixed timestep t_0. Each activation $h \in \mathbb{R}^{(HW)\times C}$ is reshaped into $N = HW$ spatial tokens of dimension C. Following prior work on token-level interpretability, we adopt a *token-wise* training scheme: a minibatch of 64 images yields $64 \times N$ training tokens per optimization

step, allowing the SAE to fully leverage spatial diversity and substantially increase the number of effective training samples.

We keep both the sparsity weight α fixed during training; no annealing schedule is applied. We did not observe training instability that would require warm-up or staged sparsification.

Architecture. The SAE is a linear encoder–decoder with learnable pre-bias and latent bias:

$$z = \phi(W_\eta(h - b_{\text{pre}}) + b_z)\,, \qquad \hat{h} = W_\psi z + b_{\text{pre}},$$

where $W_\eta \in \mathbb{R}^{K\times C}$ and $W_\psi \in \mathbb{R}^{C\times K}$. We use *untied* weights ($W_\psi \neq W_\eta^\top$), as tying the decoder noticeably reduces reconstruction fidelity at large expansion ratios. The latent dimension is set to $K = C \cdot \text{scale}$ with scale $= 32$. We adopt `ReLU` in our main experiments due to its training stability and smoother gradients.

Timestep Embedding. We keep the timestep embedding enabled (`ignore_timesteps = False`) so that the SAE can capture mild temporal variation in U-Net activations between denoising timesteps $t \in [500, 999]$. Although this temporal variation is limited, we find that including timestep information slightly improves reconstruction accuracy without affecting sparsity or alignment.

Objective. The training objective follows:

$$\mathcal{L} = \left\|\hat{h} - h\right\|_2^2 + \lambda_{\text{sparse}} \cdot \|z\|_1,$$

where the sparsity loss is implemented as the mean absolute activation of z. We tune λ_{sparse} to balance reconstruction fidelity and latent sparsity.

Optimization. The SAE is trained for 100 epochs using Adam (learning rate `lr_sae` $= 5 \times 10^{-4}$, weight decay disabled). We use a batch size of 64 images (expanded to $64 \times HW$ latent tokens per step). Gradient updates are performed on all tokens jointly, and model checkpoints are saved every 10 epochs. This token-wise design enables efficient large-scale SAE training while maintaining high reconstruction accuracy and strong sparsity.

C.3 Stage-2: Concept Alignment MLP

To map the sparse latent representation to the activation direction for semantic editing, we use a simple linear transformation implemented as a single-layer perceptron:

$$\Delta h = W_\Delta z + b_\Delta$$

where z is the latent vector, W_Δ and b_Δ are learnable parameters, and Δh is the predicted direction in the activation space.

This operation is implemented in PyTorch as:

```
import torch.nn as nn

direction_decoder = nn.Linear(latent_dim, out_dim)
delta_h = direction_decoder(z)
```

Linear vs. non-linear alignment. We replace the linear map with a parameter-matched two-layer SiLU MLP (33.55M vs. 33.82M parameters), using the same frozen SAE, 8 training and 8 held-out images, five concepts, inversion latents, CLIP/reconstruction losses, and optimization budget. The MLP gives no consistent improvement (Table 4): its target-change difference is not significant ($+0.003$, 95% CI $[-0.008, 0.016]$), its EPR is slightly lower, and its CLIP-Score decreases for all five concepts. We keep the linear map because its weights directly measure the relevance of each SAE latent to a concept, which we use to rank and select concept latents; an MLP makes this relationship input-dependent.

C.4 Steering Pipeline.

We apply semantic intervention during the denoising trajectory of the U-Net. Let $t_{\text{edit}} = 500$ denote the editing threshold. Following DDIM inversion, we obtain the noisy trajectory $\{x_t\}_{t=999}^{0}$ and at each denoising step we extract the bottleneck activation h_t from the middle block.

Table 4: Linear vs. MLP alignment, averaged over five CelebA-HQ concepts.

Mapping	Target $\Delta \uparrow$	EPR $\uparrow$	CLIP-Score $\uparrow$	LPIPS $\downarrow$
Linear	0.003	**1.113**	**0.189**	0.129
MLP	**0.006**	1.093	0.184	**0.112**

Timestep-wise intervention. Editing is applied not at a single timestep, but repeatedly across the range $t \in [500, 999]$. At every denoising step satisfying $t \geq t_{\text{edit}}$, we modify the bottleneck activation through a concept-aligned shift:

$$h'_t = h_t + \gamma\, \Delta h_c, \qquad \Delta h_c = W_\Delta\, z_t,$$

where z_t is the sparse latent representation produced by the SAE encoder, and W_Δ is the concept-mapping linear layer from Stage 2. Only this middle-block activation is modified; all other layers and all timesteps $t < 500$ remain unchanged.

Continuation of sampling. The modified activation h'_t is injected via a forward hook and the remaining DDIM update is computed normally:

$$x_{t-1} = \sqrt{\alpha_{t-1}}\, P_t(\epsilon_\theta(x_t \mid h'_t)) + D_t(\epsilon_\theta(x_t))\,.$$

Since the intervention is applied across roughly 50 denoising steps (using a step size of 10 from $t = 999$ to $t = 500$), the influence of the concept-aligned shift accumulates smoothly along the trajectory, enabling controlled semantic modification without altering the early generative dynamics.

C.5 Layer Selection for *h*-space in U-Net

Following prior work [21], we define the *h*-space as the intermediate feature representation at the bottleneck layer of the U-Net architecture. Specifically, we select the 8th layer (with spatial resolution 8×8 and channel dimension 512) as our *h*-space. This layer is not directly influenced by skip connections, which ensures that the activations encode high-level and abstract semantics, making it suitable for both sparse autoencoding and concept editing.

Our choice is motivated by the comprehensive empirical analysis in [21], where the authors systematically evaluated semantic editing performance at different layers of the U-Net. Their experiments demonstrated that the bottleneck layer (layer 8) contains the richest and most disentangled semantic information, and editing this layer yields the most effective and interpretable results for semantic manipulation tasks.

C.6 Hyperparameter Selection for Training Loss.

The selection of hyperparameters for CLIP loss (Eq. 8) is crucial for achieving optimal semantic alignment and faithful reconstruction in semantic editing tasks. As shown in Table 5, we empirically select the weighting factors for the CLIP loss and reconstruction loss ($\lambda_{\text{CLIP}} : \lambda_{\text{recons}}$) based on experiments and prior work, tailoring the settings to different datasets and target concepts. For each dataset (e.g., CelebA-HQ, FFHQ, LSUN-Church, AFHQ-Dog), we consider multiple representative concepts and adjust the loss weights accordingly to balance the semantic editing strength and image fidelity. Table 5 summarizes the datasets, concepts, target descriptions, and corresponding hyperparameter configurations used in our experiments.

C.7 Datasets and Preprocessing

AFHQ. We use the AFHQ–Dog subset following prior diffusion editing literature. Images are loaded from the official train/test split (`train/dog/*.png`, `test/dog/*.png`). Each image is resized to $s \times s$ (where s is the target resolution for each backbone, e.g., 256 or 512), converted to a float tensor in $[0, 1]$, and normalized to $[-1, 1]$ using a standard `ToTensor` + `Normalize((0.5,0.5,0.5),(0.5,0.5,0.5))` transform. No data augmentation is applied to preserve the deterministic nature of DDIM inversion.

CelebA–HQ. CelebA–HQ is stored in LMDB format for efficient random access. Each sample is retrieved via a resolution-specific LMDB key "`<res>-<index>`", decoded into a JPEG/PNG image,

and processed with the same normalization described above. The LMDB files contain the official train and test partitions. All images are resized to the target model resolution and transformed with `ToTensor` and `Normalize` without augmentation.

LSUN-Church. For LSUN, we adopt the official LMDB release and follow the conventional train/validation split (`church_outdoor_train`, `church_outdoor_val`). The loader retrieves each image from LMDB, converts it to RGB, applies `Resize` and `CenterCrop` to match the desired resolution, and then normalizes it to the $[-1, 1]$ range. The same preprocessing is applied to both training and test images.

Normalization. Across all datasets, images are scaled to $[-1, 1]$, which matches the input range expected by the U-Net denoisers used in DDPM/DM/SDXL models. We avoid augmentation because diffusion inversion and semantic editing require consistent reconstruction of the original image.

C.8 Metrics

Editing Precision Ratio (EPR). For EPR, we use the classifier *logits* rather than probabilities, as logits preserve relative semantic strength without saturation effects. We set $\epsilon = 10^{-8}$ purely for numerical stability; this does not affect the scale or ranking of the metric. For each target attribute c, all remaining annotated attributes in the dataset are treated as non-target attributes. This design evaluates unintended changes across the full attribute space, and datasets with rich attribute annotations (e.g., CelebA-HQ with 40+ labels) provide more reliable and discriminative EPR estimates.

Learned Perceptual Image Patch Similarity (LPIPS). We evaluate perceptual distortion using LPIPS [31]. Following the official implementation, we adopt the AlexNet-based LPIPS model (`lpips.LPIPS(net='alex')` in our code). All images are resized to the target resolution and normalized to the $[-1, 1]$ range before evaluation. Given an original image x and its edited counterpart $\hat{x}$, LPIPS computes the L_2 distance between deep features extracted from a pretrained perceptual network. Lower LPIPS indicates better preservation of non-target visual details.

CLIP-Score. To measure the semantic alignment between the edited images and the target attribute, we compute CLIP-Score using the ViT-L/14 CLIP model. For a text prompt t describing the target attribute, the CLIP encoder extracts a normalized text representation f_t, and each edited image $\hat{x}$ is mapped to a normalized visual feature $f_{\hat{x}}$. The CLIP-Score is the cosine similarity

$$\mathrm{CLIP}(\hat{x}, t) = \langle f_{\hat{x}},\, f_t \rangle\,,$$

where higher scores indicate stronger semantic consistency with the target attribute. All images are normalized using the default CLIP preprocessing pipeline, and scores are averaged over the entire evaluation set.

ArcFace Similarity. To quantify identity preservation in face-editing experiments, we compute the cosine similarity between ArcFace embeddings of the original and edited images. Following the standard InsightFace implementation [6], each face is detected and aligned using the SCRFD detector, and a 512-dimensional normalized embedding is extracted by the `buffalo_l` model (ResNet-50 backbone).

Given two embeddings $e, \hat{e} \in \mathbb{R}^{512}$, identity similarity is

$$\mathrm{Sim}(e, \hat{e}) = \frac{\langle e, \hat{e} \rangle}{\|e\|_2\, \|\hat{e}\|_2} \in [0, 1],$$

where higher values indicate stronger identity consistency. Images are evaluated after automatic face alignment, and pairs with undetected faces are excluded, following common practice in prior work.

C.9 Baseline Configuration

C.9.1 Concept Sliders

We implement Concept Sliders [8] as our baseline method for controllable image editing. The baseline configuration follows the original implementation with the following specifications:

Model Architecture. We use Stable Diffusion v1.4 as the base diffusion model with LoRA adaptation for concept manipulation. The LoRA network employs a `c3lier` architecture targeting both linear and convolutional layers in the UNet, with rank $r = 4$ and scaling factor $\alpha = 1.0$. Training is performed using the `noxattn` method, excluding cross-attention layers from adaptation.

Training Configuration. The model is trained for 1,000 iterations using the AdamW optimizer with a constant learning rate of 2×10^{-4}. We use DDIM scheduling with 50 denoising steps and classifier-free guidance scale of 4.0. All training is conducted at 512×512 resolution with bfloat16 precision.

Concept Definitions. For each target attribute, we define concept pairs using natural language prompts:

- **Smiling**: "smiling person" vs. "sad person"
- **Big Nose**: "person with big nose" vs. "person with small nose"
- **Beards**: "person with beards" vs. "person without beards"
- **Blonde Hair**: "person with blonde hair" vs. "person without blonde hair"
- **Young**: "very young person" vs. "very old person"

All concepts use "person" as the neutral anchor point for conditioning.

Inference Protocol. During inference, we apply the trained LoRA weights with editing strength $\alpha \in \{1, 32, 64, 96, 128\}$ using DDIM sampling with 50 steps. The LoRA scaling is activated after timestep 500 to preserve image fidelity while enabling concept manipulation. We use classifier-free guidance with scale 7.5 and the prompt "a photo of a person" for all evaluations.

Implementation Details. Models are implemented using the Diffusers library and trained on NVIDIA A100 GPUs. LoRA weights are saved every 500 iterations, with the final checkpoint used for evaluation. All images are processed at 512×512 resolution with standard normalization.

C.9.2 Asyrp

We implement Asyrp [21] as another baseline method for semantic image editing. Asyrp performs editing by learning delta blocks that modify intermediate features in the diffusion denoising process at specific timesteps.

Model Architecture. The method uses a pre-trained diffusion model on CelebA-HQ with a simple UNet architecture. The UNet consists of 128 base channels with channel multipliers $[1, 1, 2, 2, 4, 4]$, 2 residual blocks per resolution, and self-attention at 16×16 resolution. All processing is performed at 256×256 resolution with 3 input/output channels.

Diffusion Configuration. The diffusion process employs a linear noise schedule with $\beta_{\text{start}} = 10^{-4}$, $\beta_{\text{end}} = 0.02$, and $T = 1000$ timesteps. Sampling uses DDIM with 50 denoising steps during both training and inference.

Training Protocol. Delta blocks are trained using 100 training images and 32 test images over 5 iterations with batch size 1. The learning rate is set to 0.5 with a combined loss function:

$$\mathcal{L} = \lambda_{\text{CLIP}} \mathcal{L}_{\text{CLIP}} + \lambda_{L1} \mathcal{L}_{L1} \tag{17}$$

where $\lambda_{\text{CLIP}} = 1.0$ and $\lambda_{L1} = 3.0$. Training employs 50 DDIM inversion steps (n_{inv}) and 50 generation steps (n_{train}).

Temporal Control. The method uses adaptive timestep selection based on LPIPS distance thresholds:

- **Edit timestep** $t_{\text{edit}} = 500$: when delta blocks begin activation
- **LPIPS thresholds**: $\tau_{\text{edit}} = 0.33$, $\tau_{\text{noise}} = 0.1$ for automatic timestep determination

Concept Definitions. Semantic concepts are defined using text prompt pairs from source to target:

- **Smiling**: "face" $\rightarrow$ "smiling face"
- **Big Nose**: "person" $\rightarrow$ "person with big nose"
- **Beards**: "person" $\rightarrow$ "person with beards"
- **Blonde Hair**: "person" $\rightarrow$ "person with blond hair"
- **Age**: "person" $\rightarrow$ "young person" / "old person"

Inference Procedure. During inference, the method follows a three-stage process: (1) DDIM inversion with 50 steps to obtain latent codes, (2) forward diffusion with delta block injection starting at t_{edit}, and (3) optional noise injection from t_{noise} using the `add_noise_from_xt` strategy. The editing strength is controlled by the coefficient $\alpha = 1.0$.

C.9.3 BoundaryDiffusion.

We include BoundaryDiffusion [32] as a representative classifier-guided editing baseline. For each semantic attribute, we use the official CLIP-based boundaries provided by the authors, which contain both activation-space boundaries (W_h) and latent-space boundaries (W_z). During editing, the U-Net activation at timestep $t = 500$ is perturbed along the attribute direction by traversing the boundary over a fixed range ($[-500, 500]$ in our implementation), following the protocol in the original work. All methods use the same DDIM inversion setup ($n_{\text{inv}} = 40$ steps) and the same number of forward editing steps ($n_{\text{edit}} = 40$). The boundary is applied only once per denoising step, and all other hyperparameters are kept identical to the official implementation for fair comparison.

C.9.4 MasaCtrl

We employ MasaCtrl [2] as a baseline method. This approach achieves structure-preserving image editing by controlling the mutual self-attention mechanism within the U-Net architecture of diffusion models. Our configuration uses Stable Diffusion v1.4 as the pre-trained foundation model with a DDIM scheduler for the denoising process, setting beta_start=0.00085, beta_end=0.012, and beta_schedule="scaled_linear". The MasaCtrl control parameters are configured as follows: start_step=4, start_layer=10, total inference steps=50, and guidance_scale=7.5. The method first performs DDIM inversion to map the source image into the latent space, then applies mutual self-attention control during specified denoising steps (4-50) and U-Net layers (10-16) to achieve semantic editing while preserving structural coherence. Image preprocessing includes resizing input images to 512×512 resolution and normalizing pixel values to the [-1,1] range. To ensure reproducibility, we set fixed random seeds for each image during the editing process.

C.9.5 SwiftEdit:

We employ SwiftEdit as a baseline method for lightning-fast text-guided image editing via one-step diffusion. SwiftEdit utilizes a pre-trained diffusion model with single-step inversion and sampling to achieve significant efficiency improvements. Our configuration includes: image resolution resized to 512×512 pixels, middle timestep $t_{mid} = 500$, and final timestep $t_{final} = 999$. The mask generation parameters are set as follows: text hidden state scaling factor $\alpha_{ta} = 1$, editing region scaling factor $\alpha_{edit} = 0.2$, non-editing region scaling factor $\alpha_{non-edit} = 1$, mask threshold $\tau_{mask} = 0.5$, and clamp rate $\gamma_{clamp} = 3.0$. The method first predicts inverted noise through the inverse U-Net, then estimates editing masks based on the difference between source and edit prompts, and finally applies mask controllers at specified U-Net layers ("mid_blocks" and "up_blocks") for localized editing.

C.10 Choice of α in Different Concepts

The hyperparameter α determines the editing intensity along each semantic direction. In our experiments, the value of α for each concept is primarily chosen based on visual inspection of the editing results on the validation set. Specifically, we adjust α to achieve a balance between perceptible attribute change and preservation of image realism. Notably, for attributes that are abundant in the dataset (e.g., *smiling* in CelebA-HQ), a smaller α is typically sufficient to induce clear semantic changes, whereas less common or more subtle attributes often require a larger value of α. The final

Table 5: Hyperparameter settings for CLIP loss and reconstruction loss for different datasets and concepts.

Dataset	**Concept**	y^{origin}	y^{ref}	$\lambda_{\text{CLIP}} : \lambda_{\text{recons}}$
CelebA-HQ	Smiling	face	smiling face	3:1
	Big Nose	person	person with big nose	3:1
	Blond Hair	person	person with blond hair	3:1
	Beards	person	person with beards	3:1
	Eyeglasses	person	person wearing eyeglasses	3:1
	Narrow eyes	face	face with narrow eyes	3:1
	arched eyebrow	face	face with arched eyebrows	3:1
	Young	person	young person	3:1
FFHQ	Beards	person	person with beards	3:1
	Young	person	young person	3:1
	tanned	face	tanned face	3:1
	angry	face	angry face	3:1
	sad	face	sad face	3:1
LSUN-Church	Gothic	Church	Gothic Church	2.5:1
	Red Brick	Church	Red brick wall Church	2.5:1
	Snowy	Church	snowy day	2.5:1
	wooden	Church	Wooden House	2.5:1
AFHQ-Dog	Angry	Dog	Angry Dog	2.2:1
	Sleepy	Dog	Sleepy Dog	2.2:1
	Smiling	Dog	Smiling Dog	2.2:1
	puppy	Dog	Puppy	2.2:1

α settings for all datasets and concepts are summarized in Table 6. A quantitative sweep and a concept-independent selection rule that does not require visual inspection are given in App. C.11.

We do not rely on the Editing Precision Ratio (EPR) metric for the selection of α due to two primary limitations. First, EPR can only be computed on datasets with explicit attribute annotations (e.g., CelebA-HQ), restricting its applicability to many datasets or concepts without such labels. Second, EPR is inherently limited to the predefined set of annotated attributes, and therefore fails to capture editing quality for concepts or semantic directions that are not covered by existing labels. In contrast, visual inspection enables a more comprehensive and flexible assessment of editing performance across diverse and potentially unannotated attributes.

C.11 Quantitative Sensitivity to the Editing Strength

Sweep of α. To complement the visual selection above, we measure EPR, CLIP-Score, and LPIPS as α varies. For each of the five concepts in Table 1, we use the same eight held-out CelebA-HQ images and sweep $\alpha \in \{0, 32, 64, 96, 128, 160, 192\}$ while keeping the selected latent, inversion trajectory, intervention timesteps, and all other hyperparameters fixed. EPR is averaged over ResNet-18, VGG-16, MobileNetV2, and ViT-B/16; CLIP-Score uses CLIP ViT-L/14; LPIPS uses AlexNet. Edit-induced metrics are measured relative to the α=0 inversion reconstruction (Table 7).

LPIPS increases consistently with α, while CLIP-Score generally improves at moderate strengths before saturating or decreasing. EPR often favors a different operating point: for Smiling and Beards it peaks at small α, although semantic alignment continues to improve at moderate strengths. No single metric is therefore sufficient: EPR measures edit selectivity, whereas CLIP-Score and LPIPS capture semantic strength and content preservation.

A concept-independent selection rule. As an alternative to visual inspection, α can be selected on a separate validation set by maximizing CLIP-Score under a fixed preservation budget of LPIPS ≤ 0.40, applied identically to all concepts. This rule yields $\alpha = 96/64/128/64/160$ for Smiling/Young/Big Nose/Beards/Blond Hair, with EPR $2.208/0.929/1.084/2.849/2.225$, CLIP-Score $0.237/0.208/0.193/0.225/0.226$, and LPIPS $0.383/0.255/0.190/0.308/0.393$.

Table 6: Hyperparameter settings for α for different datasets and concepts.

Dataset	Concept	α
CelebA-HQ	Smiling	128
	Big Nose	64
	Blond Hair	128
	Beards	96
	Eyeglasses	96
	Narrow eyes	160
	arched eyebrow	160
	Young	128
FFHQ	Beards	96
	Young	128
	tanned	80
	angry	80
	sad	64
LSUN-Church	Gothic	16
	Red Brick	16
	Snowy	16
	wooden	16
AFHQ-Dog	Angry	32
	Sleepy	64
	Smiling	31
	puppy	64

Table 7: Metrics as α varies. Each cell lists $\alpha = 0/32/64/96/128/160/192$.

Concept	EPR ↑	CLIP-Score ↑	LPIPS ↓
Smiling	0.000/3.056/2.609/2.208/2.254/2.017/1.943	0.197/0.215/0.225/0.237/0.233/0.248/0.238	0.000/0.187/0.267/0.383/0.456/0.545/0.585
Young	0.000/0.962/0.929/0.772/0.818/1.063/1.058	0.201/0.203/0.208/0.203/0.211/0.202/0.200	0.000/0.171/0.255/0.332/0.408/0.502/0.583
Big Nose	0.000/0.919/0.998/1.141/1.084/0.907/0.890	0.177/0.182/0.183/0.189/0.193/0.192/0.186	0.000/0.078/0.104/0.159/0.190/0.218/0.264
Beards	0.000/2.856/2.849/2.388/2.472/2.230/1.903	0.165/0.210/0.225/0.237/0.252/0.231/0.242	0.000/0.224/0.308/0.408/0.489/0.559/0.601
Blond Hair	0.000/0.923/1.155/1.488/1.982/2.225/2.306	0.202/0.204/0.203/0.211/0.213/0.226/0.224	0.000/0.120/0.202/0.291/0.348/0.393/0.434

Strength sweep across methods. We also sweep the native strength control of CASL, Asyrp, and BoundaryDiffusion on the same eight images and five concepts (Table 8). The coefficients are method-specific and not directly comparable. Methods such as MasaCtrl are excluded because their controls are not analogous to a one-dimensional strength coefficient. The baselines do not outperform CASL at any tested setting; Table 1 uses the default settings of the official implementations.

Table 8: Macro EPR and signed target change under different strength settings.

Method	Settings	Macro EPR ↑	Signed target change ↑
CASL	32/64/128/192	1.743/1.708/1.722/1.620	0.152/0.224/0.242/0.154
Asyrp	0.5/1.0/1.5/2.0	1.454/1.554/1.385/1.334	0.103/0.171/0.126/0.089
BoundaryDiffusion	5/9/15/19	1.100/1.120/1.403/1.289	0.063/0.000/0.026/0.117

D Evaluation of Sparse Feature Representations

We evaluate the reconstruction quality and sparsity of the sparse autoencoder (SAE) representations as indicators of interpretability, using activations from CelebA-HQ. Reconstruction is measured by mean squared error (MSE) and cosine similarity between original and reconstructed activations. Sparsity is quantified by the dimension activation ratio (DAR):

$$\mathrm{DAR}(z) = \frac{1}{K} \sum_{j=1}^{K} \mathbb{I} \left[\frac{1}{N} \sum_{i=1}^{N} z_{ij} > \tau \right] \tag{18}$$

where $z \in \mathbb{R}^{N \times K}$ is the latent activation matrix and τ is a fixed threshold. Here, K is determined by a predefined expansion ratio γ_{sae} multiplied by the input channel dimension C. Lower DAR indicates a sparser representation.

Table 9: Reconstruction quality and sparsity of the SAE across different expansion ratios γ_{sae}. MSE and DAR values are scaled by 10^{-2} for readability.

Params	**MSE**($\times 10^{-2}$) $\downarrow$	**Cosine Sim.** $\uparrow$	**DAR** ($\times 10^{-2}$) $\downarrow$
$\gamma_{\text{sae}} = 16$	9.27	0.965	6.32
$\gamma_{\text{sae}} = 32$	5.39	0.980	3.54
$\gamma_{\text{sae}} = 64$	3.19	0.988	1.87
$\gamma_{\text{sae}} = 128$	1.91	0.993	1.08
$\gamma_{\text{sae}} = 256$	0.94	0.996	0.70
$\gamma_{\text{sae}} = 512$	0.56	0.998	0.39

Table 9 reports SAE performance across different expansion ratios γ_{sae} on CelebA-HQ, with input activations $h^{(t)}$ sampled uniformly from 50 timesteps between $t = 999$ and $t = 500$ during the DDIM inversion process. All configurations achieve low reconstruction error while maintaining strong sparsity, demonstrating the effectiveness of our design. We adopt $\gamma_{\text{sae}} = 128$ and sparsity weight $\lambda_{\text{sparse}} = 32$ as default for balancing performance and efficiency.

To validate the generalizability of our sparse autoencoder, we further conduct ablation studies on the **FFHQ dataset**. As shown in Table 10, we report the reconstruction quality (MSE, Cosine Similarity) and sparsity (DAR, $\tau = 0.01$) under different sparsity loss weights λ_{sparse}, with the latent dimension γ_{sae} fixed at 128. Compared to the results on CelebA-HQ, the trends on FFHQ are consistent, demonstrating the robustness of our method across datasets.

Table 10: Reconstruction quality and sparsity of the SAE trained on FFHQ across different sparsity loss weight λ_{sparse}.

Params	**MSE** $\downarrow$	**Cosine Sim.** $\uparrow$	**DAR** ($\tau = 0.01$) $\downarrow$
$\lambda_{\text{sparse}} = 0.5$	2.34×10^{-2}	0.993	2.72×10^{-2}
$\lambda_{\text{sparse}} = 1$	2.33×10^{-2}	0.994	2.50×10^{-2}
$\lambda_{\text{sparse}} = 2$	2.18×10^{-2}	0.994	2.23×10^{-2}
$\lambda_{\text{sparse}} = 4$	2.07×10^{-2}	0.994	2.03×10^{-2}
$\lambda_{\text{sparse}} = 8$	2.30×10^{-2}	0.993	1.80×10^{-2}
$\lambda_{\text{sparse}} = 16$	2.77×10^{-1}	0.987	4.81×10^{-2}
$\lambda_{\text{sparse}} = 32$	3.69×10^{-1}	0.986	1.72×10^{-2}

D.1 Label-free and Label-based Disentanglement Metrics

OMES. We encode fixed-timestep U-Net h-space activations with the frozen 512-to-65,536 SAE and spatially pool the latents. OMES [5] combines the Overlap Score (OS), measuring whether different factors share latents, and the Multiple Encoding Score (MES), measuring whether one factor is spread over many latents. All selection strategies use identical representations, factor pairs, active-latent pools, and budgets: *Random* samples k active latents (1,000 trials), *High-act.* selects those with the largest mean activation, and *CASL* selects latents by Stage-2 alignment-weight magnitude. On CelebA-HQ we use 4,096 independent images and factor-controlled centroid pairs for five attributes; on LSUN-Church we use 32 controlled source/intervention pairs for each of four concepts, constructed without CASL outputs. Pairs can likewise be obtained from controlled generation or interventions for any unannotated dataset, so OMES can be used as a classifier-free evaluation module without changing CASL training.

CASL achieves the best OMES at every budget on CelebA-HQ and for $k \geq 4$ on LSUN-Church (Table 11), indicating that Stage 2 selects more modular and compact concept latents than random or activation-based heuristics.

DCI. We further evaluate DCI [7] on an independent set of 4,096 CelebA-HQ images using the same frozen SAE and five factors. Gradient-boosted classifiers predict the factors from the selected latents; their importance matrix gives Disentanglement (D) and Completeness (C), and held-out

Table 11: OS / MES / OMES (all higher is better) for k selected latents.

k	Random			High-act.			CASL		
	OS	MES	OMES	OS	MES	OMES	OS	MES	OMES
CelebA-HQ									
1	0.395	0.676	0.536	0.532	0.447	0.489	0.550	0.691	**0.621**
2	0.398	0.539	0.469	0.398	0.584	0.491	0.475	0.555	**0.515**
4	0.401	0.441	0.421	0.410	0.465	0.438	0.482	0.499	**0.491**
8	0.401	0.383	0.392	0.419	0.409	0.414	0.448	0.443	**0.445**
16	0.401	0.354	0.377	0.415	0.386	0.400	0.431	0.429	**0.430**
LSUN-Church									
1	0.658	0.183	**0.421**	0.688	0.123	0.406	0.693	0.135	0.414
2	0.656	0.515	0.585	0.693	0.504	**0.598**	0.665	0.505	0.585
4	0.656	0.671	0.664	0.661	0.682	0.671	0.665	0.686	**0.676**
8	0.655	0.746	0.700	0.666	0.762	0.714	0.668	0.771	**0.720**
16	0.655	0.782	0.719	0.658	0.785	0.721	0.655	0.803	**0.729**

accuracy gives Informativeness (I). Random is averaged over 100 subspaces. CASL achieves the highest D, C, and I at every budget (Table 12).

Table 12: DCI on CelebA-HQ. Each entry reports D/C/I (higher is better).

k	Random	High-activation	CASL
4	0.080/0.073/0.748	0.089/0.090/0.755	**0.114/0.104/0.766**
8	0.148/0.111/0.766	0.185/0.144/0.775	**0.192/0.153/0.792**
16	0.233/0.151/0.782	0.261/0.167/0.787	**0.276/0.174/0.794**

D.2 Target vs. Non-target Attribute Predictability

Using the same top-k units as in Table 3 (k=8) and the same SVM protocol, we train one classifier for the target attribute and one for each remaining attribute, averaging non-target results (Table 13). The selected latents are consistently more predictive of their target (average gaps of 0.152 in accuracy and 0.111 in AUROC). Non-target AUROC remains above chance, which is expected given the strong correlations among CelebA attributes: a non-target attribute can be predicted indirectly even when a latent primarily encodes one concept. We therefore treat these results only as a diagnostic of attribute predictability, and rely on OMES and DCI for representation disentanglement and on EPR for edit selectivity.

Table 13: Target vs. non-target predictability of the selected latent sets.

Concept	Target Acc.	Non-target Acc.	Target AUROC	Non-target AUROC
Beards	0.891	0.722	0.810	0.756
Blond Hair	0.883	0.706	0.814	0.751
Smiling	0.948	0.728	0.910	0.760
Young	0.797	0.715	0.849	0.770
Big Nose	0.771	0.657	0.878	0.667
Average	0.858	0.706	0.852	0.741

E User Study Design

To further evaluate the effectiveness of our Editing Precision Ratio (EPR) metric, we conducted a user study to assess whether the metric aligns with human perception of editing quality.

E.1 Experimental Setup

In this experiment, we invited **10** participants to rate a collection of edited images generated by different semantic editing methods, including ours and baseline approaches. Each test sample consists of the following:

- **Original Image:** The unedited reference image.
- **Target Description:** The semantic attribute to be edited (e.g., "smiling").
- **Edited Image:** The result of semantic editing according to the target attribute.

For each sample, participants were instructed to evaluate the edited image on two aspects:

1. **Target Change:** How much does the edited image match the target description?
   (1 = No visible change, 10 = Completely matches the target description)
2. **Identity Preservation:** How well does the edited image preserve the identity of the original image?
   (1 = Perfectly preserves identity, 10 = Looks like a completely different person/image)

Participants were asked to provide their ratings in the format: **Target/Non-target** (e.g., 3/2).

E.2 Procedure

The method names and editing settings were anonymized to ensure a fair assessment. Prior to the formal study, a brief tutorial and several examples were given to each participant.

E.3 Participants

A total of **10** participants were recruited for the user study. (Details of participant demographics are omitted for anonymity.)

E.4 Results

Detailed results are provided in the supplementary file `Human_Evaluation.pdf`. In the supplementary tables, `edited1` corresponds to our method (CASL-Steer), `edited2` refers to Asyrp, `edited3` denotes BoundaryDiffusion, and `edited4` represents the reconstruction baseline without editing. We observe that the trends in human evaluation are largely consistent with those measured by the EPR metric, further supporting the rationality and practical effectiveness of our approach.

E.5 Summary

This user study aims to assess the subjective perception of editing precision and identity preservation, and to validate the consistency between the EPR metric and human judgment.

E.6 Aggregated Results

We aggregate the raw ratings above to make the comparison explicit. For each method, we first average the five images rated by each participant, then report the mean and standard error across the 10 participants. Identity change is converted to identity preservation as $S_{\text{preserve}} = 11 - S_{\text{identity change}}$, so both scores are higher-is-better (Table 14). CASL-Steer achieves target changes comparable to Asyrp while substantially improving identity preservation. Across the 16 method–concept combinations, EPR is strongly correlated with human-rated target change (Spearman $\rho = 0.946$, $p < 10^{-7}$), quantitatively supporting the consistency between the two.

Table 14: Aggregated user-study ratings (mean $\pm$ s.e. over 10 participants).

Method	Target Change ↑	Identity Preservation ↑	Mean EPR ↑
CASL-Steer	6.36 ± 0.24	7.61 ± 0.35	**2.457**
Asyrp	$\mathbf{6.59 \pm 0.44}$	5.02 ± 0.45	1.949
BoundaryDiffusion	3.88 ± 0.62	8.65 ± 0.28	1.171
Reconstruction	3.34 ± 0.82	$\mathbf{9.83 \pm 0.06}$	0.581

F Performance of the Attribute Classifier

To ensure fair and robust evaluation, we independently trained four attribute classifiers, ResNet18, VGG16, MobileNetV2, and ViT-B/16, using identical settings on the **cropped CelebA** dataset. All classifiers are fine-tuned from ImageNet-pretrained weights.

Following the standard protocol, we split the dataset as follows: images with indices `image_n` $<$ `162771` are used for training, `162771` $\leq$ `image_n` $<$ `182638` for validation, and `image_n` $\geq$ `182638` for testing.

- **ResNet18** is a classical convolutional neural network (CNN) featuring residual connections, which enable effective training of deep models by mitigating the vanishing gradient problem.
- **VGG16** is a deep CNN characterized by its uniform use of 3×3 convolutional layers and simple, sequential architecture, widely adopted as a strong baseline for visual tasks.
- **MobileNetV2** is a lightweight CNN designed for efficient inference on mobile and embedded devices, employing depthwise separable convolutions and inverted residual blocks to reduce computational cost while maintaining competitive performance.
- **ViT-B/16** is a vision transformer model that processes images as sequences of non-overlapping patches and leverages self-attention mechanisms to capture global dependencies, marking a departure from traditional CNN-based designs.

The tables below(table. 15, table. 16, table. 17, table. 18) reports their performance on the test set. As shown, all classifiers achieve consistently high accuracy, demonstrating stable and reliable attribute prediction across different initializations.

F.1 Ablation Study on EPR Sensitivity

To better understand the behavior and limitations of the Editing Precision Ratio (EPR), we conduct an ablation study using different attribute classifiers, including ResNet18, MobileNetV2, VGG16, and ViT-B/16. The results are summarized in Table 20, which reports EPR for five facial attributes across all editing methods.

Overall, EPR values exhibit substantial variation across classifier backbones. For example, the same edited images can yield noticeably different EPR scores when evaluated by ResNet18 versus VGG16, and ViT-B/16 shows an even larger degree of inconsistency, often producing unstable or uninformative scores. This divergence indicates that EPR is highly sensitive to the choice of attribute classifier: models with different architectures, feature embeddings, and decision boundaries respond differently to subtle attribute shifts introduced during editing.

While EPR provides a convenient quantitative measure of editing specificity under supervised attribute labels, this ablation highlights several inherent limitations. First, EPR fundamentally depends on high-quality, concept-aligned labels, and cannot be reliably computed when labels are noisy, ambiguous, or unavailable. Second, its dependence on a particular classifier introduces model-induced bias, making absolute EPR values difficult to compare across backbones or datasets. Finally, concepts with broader visual variation or weak label definitions tend to produce more volatile EPR scores, limiting the metric's robustness in open-domain or attribute-sparse settings.

Taken together, these findings suggest that EPR should be interpreted with caution and primarily as a relative metric within a fixed classifier setting, rather than as an absolute measure of editing quality. This motivates the need for complementary evaluation metrics, such as CLIP-Score and LPIPS, that do not rely on explicit attribute labels or classifier decision boundaries.

F.2 EPR Aggregated over Classifier Backbones

EPR in Table 1 is computed with a single ResNet-18 classifier. To reduce dependence on one evaluator, we average EPR over four independently trained backbones (ResNet-18, VGG-16, MobileNetV2, ViT-B/16) with equal weights (Table 19). Although absolute values change with the classifier, CASL-Steer ranks first on three concepts and second on the other two, and obtains the highest average.

Table 15: Performance of ResNet-18 attribute classifier on CelebA (test set).

Attribute	Accuracy	Balanced Acc	F1	Precision	Recall
5_o_Clock_Shadow	94.823	86.954	74.849	72.718	77.121
Arched_Eyebrows	84.358	80.833	72.543	72.449	72.656
Attractive	83.173	83.163	82.846	83.766	81.962
Bags_Under_Eyes	85.504	78.772	65.342	63.375	67.451
Bald	99.067	86.849	77.095	80.376	74.090
Bangs	96.204	91.944	87.557	89.435	85.757
Big_Lips	72.832	62.683	44.509	67.068	33.346
Big_Nose	84.297	76.823	63.289	62.761	63.847
Black_Hair	90.484	86.472	81.599	85.930	77.687
Blond_Hair	96.182	91.368	85.547	86.308	84.805
Blurry	96.380	73.442	57.232	71.188	47.921
Brown_Hair	89.496	83.691	71.858	69.291	74.631
Bushy_Eyebrows	93.038	80.330	70.158	78.885	63.179
Chubby	95.942	75.738	58.131	64.195	53.138
Double_Chin	96.470	72.727	54.690	66.250	46.594
Eyeglasses	99.684	98.401	97.541	98.162	96.928
Goatee	97.593	88.127	74.747	72.023	77.705
Gray_Hair	98.296	84.933	72.536	74.555	70.660
Heavy_Makeup	92.040	91.481	90.008	91.535	88.538
High_Cheekbones	88.085	87.996	87.370	89.293	85.529
Male	98.373	98.214	97.887	98.261	97.517
Mouth_Slightly_Open	94.339	94.332	94.241	94.938	93.555
Mustache	97.112	72.738	55.364	68.804	46.321
Narrow_Eyes	87.750	64.538	43.325	69.442	31.503
No_Beard	96.516	93.404	97.956	98.109	97.804
Oval_Face	76.107	63.041	43.454	72.329	31.079
Pale_Skin	97.200	75.297	60.674	74.176	51.381
Pointy_Nose	77.812	68.238	54.168	66.095	45.898
Receding_Hairline	94.025	74.910	59.570	69.943	51.889
Rosy_Cheeks	95.408	77.371	63.751	73.512	56.313
Sideburns	97.952	89.587	78.453	76.631	80.367
Smiling	93.365	93.366	93.268	94.721	91.859
Straight_Hair	84.999	74.687	61.427	66.734	56.912
Wavy_Hair	85.321	82.027	77.616	87.235	69.911
Wearing_Earrings	90.660	86.182	77.657	76.787	78.550
Wearing_Hat	99.176	94.328	90.086	91.165	89.035
Wearing_Lipstick	94.172	94.236	94.322	95.931	92.768
Wearing_Necklace	88.140	65.181	43.759	63.258	33.476
Wearing_Necktie	97.088	87.067	78.399	81.664	75.411
Young	89.021	82.631	92.913	90.866	95.056
Cumulative avg	91.712	86.739	81.218	85.327	77.487
Attr avg mean	91.712	82.353	73.193	78.754	69.754

G Additional Results

G.1 Quantitative Results

To further examine the generalization ability of our method beyond human facial attributes, we additionally evaluate it on two structurally different datasets: LSUN-Church and AFHQ-Dog. For LSUN-Church, we consider four architectural concepts (Gothic, Red Brick, Snowy, Wooden), while AFHQ-Dog includes four canine appearance and expression traits (Angry, Sleepy, Smiling, Puppy-like). The corresponding quantitative comparisons are presented in Table 21 and Table 22. Across both datasets, our approach produces stable and semantically aligned edits while maintaining distortion levels comparable to, or lower than, existing editing baselines. These observations suggest

Table 16: Performance of ViT-B/16 attribute classifier on CelebA (test set).

Attribute	Accuracy	Balanced Acc	F1	Precision	Recall
5_o_Clock_Shadow	91.705	78.454	63.491	66.067	61.109
Arched_Eyebrows	80.717	75.203	63.096	62.417	63.790
Attractive	77.224	77.149	78.298	77.602	79.007
Bags_Under_Eyes	80.717	68.110	50.046	54.087	46.566
Bald	98.374	73.924	55.201	64.194	48.418
Bangs	94.166	87.406	79.656	81.560	77.839
Big_Lips	78.140	60.898	33.563	31.406	36.038
Big_Nose	79.061	68.338	52.759	60.135	46.996
Black_Hair	89.681	85.422	75.950	73.904	78.113
Blond_Hair	94.458	88.492	81.598	83.396	79.876
Blurry	94.956	64.091	36.015	45.048	30.000
Brown_Hair	83.843	75.279	63.688	69.558	58.731
Bushy_Eyebrows	90.985	78.912	66.227	71.036	62.028
Chubby	94.196	66.582	42.551	53.982	35.115
Double_Chin	95.520	68.609	45.930	56.334	38.769
Eyeglasses	98.102	90.551	85.714	90.048	81.779
Goatee	95.012	78.792	63.845	68.520	59.768
Gray_Hair	97.448	82.861	71.786	77.711	66.701
Heavy_Makeup	89.435	88.979	86.520	86.138	86.905
High_Cheekbones	85.755	85.560	84.065	84.503	83.632
Male	97.443	97.404	97.001	96.864	97.139
Mouth_Slightly_Open	91.307	91.292	90.969	91.079	90.860
Mustache	95.435	68.332	45.786	57.164	38.185
Narrow_Eyes	91.101	68.001	40.790	40.763	40.818
No_Beard	93.104	87.510	95.823	95.462	96.187
Oval_Face	71.702	61.514	43.155	49.341	38.347
Pale_Skin	96.341	71.982	51.630	59.969	45.327
Pointy_Nose	72.769	63.140	46.030	52.864	40.760
Receding_Hairline	93.321	71.419	49.678	54.222	45.836
Rosy_Cheeks	94.438	73.303	54.545	61.789	48.822
Sideburns	95.616	78.372	64.694	72.545	58.376
Smiling	90.915	90.872	90.504	91.443	89.586
Straight_Hair	79.856	66.129	46.640	51.215	42.815
Wavy_Hair	80.284	77.555	66.718	62.576	71.447
Wearing_Earrings	84.945	73.431	58.139	61.889	54.817
Wearing_Hat	98.294	89.652	81.626	83.204	80.106
Wearing_Lipstick	91.136	91.412	90.435	87.164	93.962
Wearing_Necklace	86.420	58.067	26.883	38.331	20.701
Wearing_Necktie	95.319	79.017	65.038	71.076	59.945
Young	84.728	76.719	90.088	87.388	92.961
Cumulative avg	89.349	83.781	75.443	77.189	73.774
Attr avg mean	89.349	76.968	64.404	68.100	61.704

that the sparse latent directions learned by our method capture concept-level semantics that transfer reasonably well across diverse visual domains.

It is important to note that we do not report EPR or ArcFace on these datasets. EPR is specifically designed for settings where each concept is associated with well-defined attribute labels; however, LSUN-Church and AFHQ-Dog do not provide reliable per-image attribute annotations, making EPR ill-defined in these domains. Furthermore, ArcFace is a face recognition model and does not apply to non-human subjects such as buildings or animals. We therefore limit our evaluation to CLIP-Score and LPIPS, which remain applicable and provide a consistent measure of semantic alignment and perceptual distortion across a broader range of concepts.

Table 17: Performance of MobileNetV2 attribute classifier on CelebA (test set).

Attribute	Accuracy	Balanced Acc	F1	Precision	Recall
5_o_Clock_Shadow	94.357	84.224	74.803	79.087	70.959
Arched_Eyebrows	86.339	82.280	73.650	73.422	73.880
Attractive	81.985	81.930	82.786	82.283	83.295
Bags_Under_Eyes	84.240	73.585	59.311	63.850	55.375
Bald	99.018	83.899	74.172	81.395	68.127
Bangs	95.898	91.404	85.883	86.739	85.043
Big_Lips	82.020	65.167	41.056	41.247	40.867
Big_Nose	83.480	74.608	63.173	70.925	56.949
Black_Hair	91.710	87.876	80.358	79.439	81.298
Blond_Hair	95.777	91.226	86.047	87.487	84.653
Blurry	96.653	74.839	58.925	70.250	50.745
Brown_Hair	86.067	78.005	68.373	75.575	62.424
Bushy_Eyebrows	92.742	81.615	72.173	79.541	66.054
Chubby	95.767	74.029	58.754	72.783	49.260
Double_Chin	96.703	74.484	59.742	74.540	49.846
Eyeglasses	99.577	97.967	96.937	97.792	96.095
Goatee	96.869	86.584	77.817	81.418	74.522
Gray_Hair	97.966	85.636	77.506	83.957	71.975
Heavy_Makeup	92.576	92.321	90.549	89.944	91.162
High_Cheekbones	88.312	88.107	86.874	87.677	86.086
Male	98.812	98.791	98.606	98.559	98.652
Mouth_Slightly_Open	94.151	94.138	93.922	94.060	93.785
Mustache	96.275	73.778	56.927	68.392	48.754
Narrow_Eyes	93.467	74.822	54.868	57.009	52.882
No_Beard	96.215	93.322	97.701	97.594	97.809
Oval_Face	75.940	64.326	46.889	61.426	37.916
Pale_Skin	96.909	72.726	56.330	72.000	46.262
Pointy_Nose	77.324	68.244	54.222	63.812	47.138
Receding_Hairline	94.604	76.338	59.455	64.691	55.003
Rosy_Cheeks	95.248	74.830	59.554	71.209	51.178
Sideburns	97.347	87.059	79.582	84.596	75.128
Smiling	93.270	93.230	92.967	93.921	92.033
Straight_Hair	85.423	73.915	60.534	68.275	54.370
Wavy_Hair	86.661	85.155	77.230	73.156	81.783
Wearing_Earrings	91.800	84.555	77.214	82.143	72.842
Wearing_Hat	99.034	94.286	89.711	90.389	89.043
Wearing_Lipstick	92.948	93.290	92.424	88.716	96.456
Wearing_Necklace	89.545	65.786	44.301	61.965	34.474
Wearing_Necktie	96.879	85.543	77.088	82.581	72.280
Young	88.237	81.305	92.369	89.559	95.361
Cumulative avg	91.954	87.354	81.342	83.727	79.088
Attr avg mean	91.954	82.131	73.270	78.085	69.794

While EPR offers a unified way to quantify the balance between editing strength and unintended attribute changes, its reliance on explicit attribute labels is a fundamental limitation. In label-scarce or open-domain scenarios, such as LSUN-Church and AFHQ-Dog, EPR cannot be meaningfully computed. These results highlight both the usefulness and the scope constraints of label-based metrics, and motivate future work toward developing label-free editing precision metrics that remain valid across heterogeneous visual domains.

G.2 AugCLIP Evaluation

EPR does not penalize global changes outside the attribute set, such as background drift. We therefore additionally report AugCLIP [20], which jointly measures target modification and global preservation,

Table 18: Performance of VGG16 attribute classifier on CelebA (test set).

Attribute	Accuracy	Balanced Acc	F1	Precision	Recall
5_o_Clock_Shadow	94.378	86.009	75.911	76.789	75.053
Arched_Eyebrows	86.636	83.007	74.488	73.507	75.497
Attractive	81.980	81.878	82.973	81.569	84.427
Bags_Under_Eyes	84.633	74.648	60.854	64.519	57.583
Bald	99.039	88.911	77.126	75.943	78.346
Bangs	96.099	92.587	86.826	86.051	87.616
Big_Lips	84.799	64.574	41.654	50.563	35.414
Big_Nose	83.500	74.060	62.503	71.914	55.270
Black_Hair	91.856	89.017	81.169	78.395	84.146
Blond_Hair	95.666	92.138	86.070	85.120	87.042
Blurry	96.688	77.385	61.565	68.264	56.064
Brown_Hair	86.228	79.327	69.808	74.092	65.992
Bushy_Eyebrows	92.898	83.223	73.661	78.108	69.693
Chubby	95.938	75.927	61.553	73.160	53.125
Double_Chin	96.814	77.071	62.961	73.297	55.179
Eyeglasses	99.562	98.159	96.844	97.162	96.529
Goatee	96.920	88.906	79.184	78.862	79.508
Gray_Hair	98.117	88.561	80.128	82.404	77.973
Heavy_Makeup	92.767	92.692	90.878	89.453	92.349
High_Cheekbones	88.544	88.360	87.160	87.784	86.545
Male	98.822	98.806	98.618	98.536	98.699
Mouth_Slightly_Open	94.403	94.382	94.170	94.527	93.816
Mustache	96.371	76.141	59.878	67.758	53.639
Narrow_Eyes	93.532	78.983	58.959	56.315	61.863
No_Beard	96.421	93.647	97.827	97.704	97.950
Oval_Face	76.408	64.036	46.021	64.080	35.903
Pale_Skin	96.950	75.369	59.383	69.654	51.752
Pointy_Nose	77.425	68.240	54.202	64.215	46.890
Receding_Hairline	94.770	77.525	61.217	65.600	57.383
Rosy_Cheeks	95.138	77.466	61.575	66.955	56.996
Sideburns	97.297	89.369	80.322	80.470	80.176
Smiling	93.628	93.586	93.336	94.359	92.335
Straight_Hair	85.458	75.652	62.524	66.501	58.996
Wavy_Hair	86.772	85.850	77.797	72.607	83.785
Wearing_Earrings	92.203	85.752	78.655	82.295	75.323
Wearing_Hat	99.099	95.382	90.554	89.843	91.277
Wearing_Lipstick	92.676	93.035	92.147	88.292	96.354
Wearing_Necklace	89.480	63.894	40.894	63.421	30.175
Wearing_Necktie	96.703	85.544	76.156	80.215	72.488
Young	88.473	81.470	92.533	89.595	95.672
Cumulative avg	92.127	87.838	81.865	83.678	80.130
Attr avg mean	92.127	83.264	74.252	77.497	72.121

on 32 paired CelebA-HQ images per concept (Table 23). CASL improves over Asyrp on all five concepts, with an average gain of 0.065.

Table 19: EPR averaged over four classifier backbones on CelebA-HQ.

Concept	Boundary	Asyrp	Slider	MasaCtrl	SwiftEdit	CASL-Steer
Smiling	1.3195	3.1993	1.2888	1.5540	**3.7462**	3.4253
Big Nose	0.9405	0.9037	1.1455	0.8992	0.9383	**1.1990**
Young	1.3985	1.6133	1.0853	1.3722	1.5455	**1.6162**
Beards	1.2210	1.6465	1.1080	**1.8950**	1.7980	1.8475
Blond Hair	0.9380	1.1917	0.7170	1.0195	1.2065	**1.3590**
Average	1.1635	1.7109	1.0689	1.3480	1.8469	**1.8894**

Table 20: Editing Precision Ratio (EPR) scores for different methods and attribute-classifier combinations.

Model	**Concept**	**recon.**	**Boundary** [32]	**Asyrp** [21]	**Slider** [8]	**MasaCtrl** [2]	**SwiftEdit** [22]	**CASL-Steer**
ResNet18	Smiling	0.679±0.289	1.231±0.976	3.199±2.182	0.949±0.479	1.685±1.375	3.359±2.064	4.465±2.319
	Big Nose	0.207±0.280	0.882±0.539	0.938±0.608	1.035±0.775	0.793±0.600	0.867±0.627	1.060±0.608
	Young	0.773±0.619	1.411±0.909	1.685±1.220	0.937±0.770	1.427±0.709	1.536±0.989	1.817±1.117
	Beards	0.763±0.669	1.165±0.974	1.739±1.268	0.919±0.383	2.059±1.541	1.255±0.884	2.161±1.080
	Blond Hair	0.110±0.032	0.876±0.794	1.173±1.002	0.760±0.548	1.173±0.678	1.302±1.014	1.386±1.213
MobileNetV2	Smiling	0.848±0.375	1.292±1.029	3.490±3.346	1.326±0.965	1.668±1.312	4.516±1.814	3.237±1.803
	Big Nose	0.169±0.081	0.995±0.423	0.872±0.292	1.381±0.820	0.759±0.555	1.045±0.749	1.261±0.650
	Young	0.799±0.571	1.532±1.099	1.808±1.757	1.070±0.540	1.407±0.379	1.797±1.058	1.799±1.256
	Beards	0.375±0.332	1.175±0.626	1.735±1.448	1.108±0.477	1.837±0.897	2.167±2.019	1.952±1.521
	Blond Hair	0.457±0.338	1.010±0.502	1.324±1.022	0.736±0.236	0.916±0.545	1.524±1.113	1.528±1.168
VGG16	Smiling	0.891±0.615	1.454±1.276	3.742±2.238	1.240±0.851	1.459±1.165	4.684±2.039	3.628±2.069
	Big Nose	0.907±0.248	0.669±0.609	0.734±0.464	0.896±0.721	0.766±0.528	0.855±0.583	0.902±0.482
	Young	0.830±0.747	1.493±0.856	1.718±0.991	1.003±0.760	1.396±0.721	1.634±0.916	1.653±1.067
	Beards	0.873±0.563	1.193±0.703	1.729±1.301	1.234±0.787	2.095±0.831	2.222±1.809	2.101±1.263
	Blond Hair	0.251±0.606	0.923±0.621	1.132±0.803	0.619±0.399	0.920±0.681	1.092± 0.834	1.461±1.245
ViT-B/16	Smiling	1.144±0.848	1.301±0.894	2.366±1.388	1.640±1.161	1.404±0.976	2.426±1.402	2.371±1.375
	Big Nose	0.577±0.419	1.216±1.093	1.071±0.704	1.270±0.796	1.279±0.943	0.986±0.693	1.573±1.039
	Young	0.236±0.135	1.158±0.841	1.242±0.735	1.331±1.181	1.259±0.974	1.215±0.939	1.196±0.997
	Beards	0.248±0.189	1.351±1.199	1.383±0.702	1.171±0.806	1.589±1.083	1.548±0.999	1.176±1.016
	Blond Hair	0.763±0.113	0.943±0.709	1.138±0.905	0.753±0.511	1.069±1.031	0.908±0.680	1.061±0.717

Table 21: Evaluation on LSUN-Church dataset.

Concept	**Metric**	**Recon.**	**Editing Methods**					**Explainable**
			Boundary	**Asyrp**	**Slider**	**MasaCtrl**	**SwiftEdit**	**CASL-Steer**
Gothic	CLIP-Score (↑)	0.103±0.014	0.129±0.021	0.158±0.118	0.167±0.014	0.193±0.039	0.184±0.091	0.172±0.031
	LPIPS (↓)	0.184±0.028	0.268±0.052	0.302±0.143	0.256±0.035	0.332±0.071	0.315±0.152	0.247±0.040
Red Brick	CLIP-Score (↑)	0.118±0.016	0.145±0.024	0.174±0.132	0.183±0.016	0.196±0.042	0.211±0.097	0.203±0.037
	LPIPS (↓)	0.198±0.030	0.279±0.060	0.310±0.149	0.276±0.038	0.338±0.078	0.327±0.162	0.251±0.044
Snowy	CLIP-Score (↑)	0.132±0.019	0.154±0.027	0.182±0.146	0.189±0.018	0.219±0.049	0.226±0.124	0.209±0.034
	LPIPS (↓)	0.213±0.033	0.291±0.067	0.326±0.164	0.284±0.041	0.353±0.083	0.341±0.174	0.258±0.048
Wooden	CLIP-Score (↑)	0.111±0.015	0.139±0.023	0.164±0.128	0.174±0.017	0.204±0.047	0.198±0.102	0.218±0.052
	LPIPS (↓)	0.191±0.031	0.273±0.058	0.298±0.156	0.268±0.036	0.329±0.080	0.316±0.170	0.244±0.043

Table 22: Evaluation on the AFHQ-Dog dataset.

Concept	Metric	Recon.	Editing Methods					Explainable
			Boundary	Asyrp	Slider	MasaCtrl	SwiftEdit	CASL-Steer
Angry	CLIP-Score (↑)	0.115±0.017	0.142±0.024	0.171±0.148	0.182±0.016	0.201±0.045	0.219±0.134	0.197±0.039
	LPIPS (↓)	0.204±0.031	0.284±0.062	0.318±0.173	0.276±0.039	0.347±0.087	0.333±0.182	0.253±0.046
Sleepy	CLIP-Score (↑)	0.108±0.015	0.131±0.020	0.159±0.136	0.167±0.014	0.185±0.041	0.193±0.121	0.176±0.033
	LPIPS (↓)	0.196±0.030	0.273±0.057	0.301±0.159	0.268±0.036	0.329±0.082	0.317±0.168	0.243±0.045
Smiling	CLIP-Score (↑)	0.121±0.018	0.148±0.025	0.179±0.155	0.188±0.017	0.208±0.047	0.217±0.145	0.204±0.041
	LPIPS (↓)	0.207±0.033	0.289±0.064	0.324±0.166	0.277±0.040	0.346±0.089	0.335±0.178	0.257±0.048
Puppy-like	CLIP-Score (↑)	0.134±0.019	0.156±0.027	0.188±0.162	0.197±0.018	0.221±0.052	0.226±0.152	0.233±0.056
	LPIPS (↓)	0.215±0.034	0.296±0.069	0.332±0.178	0.283±0.041	0.359±0.093	0.341±0.185	0.249±0.050

Table 23: AugCLIP on CelebA-HQ (higher is better).

Method	Smiling	Big Nose	Young	Beards	Blond Hair	Avg.
Asyrp	0.674	0.765	0.648	0.610	0.713	0.682
CASL	**0.729**	**0.850**	**0.673**	**0.697**	**0.789**	**0.748**

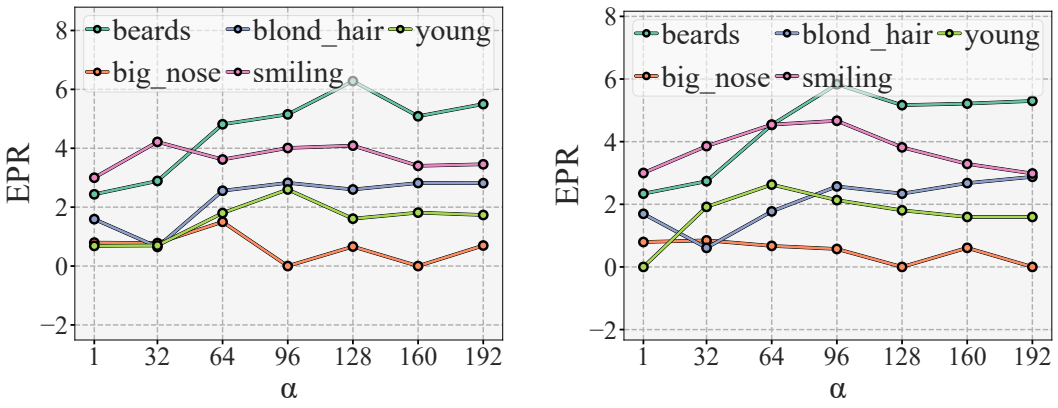
H Full Hyperparameter Analysis Results

This section reports the complete hyperparameter sweeps summarized in Sec. 5.5. Fig. 7 plots EPR against α for all concepts at each top-k; Fig. 6 plots EPR against α for each top-k per concept; and Fig. 8 plots EPR against top-k at fixed α.

Effect of α. With the single highest-ranked latent (top-k=1), EPR remains stable over a wide range of α for most concepts: amplifying one aligned direction strengthens the target attribute without recruiting unrelated attributes. The quantitative sweep in App. C.11 shows the complementary view: LPIPS grows monotonically with α, so large strengths trade preservation for semantic strength.

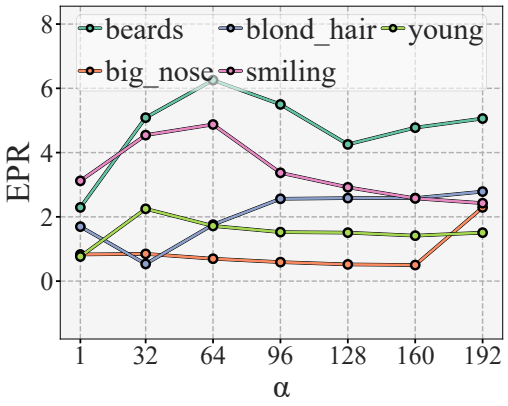
Effect of top-k. Adding more latents increases the available semantic signal (cf. the classification results in Table 3) but also mixes in latents that are only partially concept-specific.

Interaction. The two hyperparameters interact: for $k > 1$, EPR decreases markedly as α increases (e.g., for Smiling), because amplifying a combination of partially entangled latents also amplifies their correlated components. Small k is therefore the most robust choice when strong edits are required, and moderate α is preferable when several latents are combined.

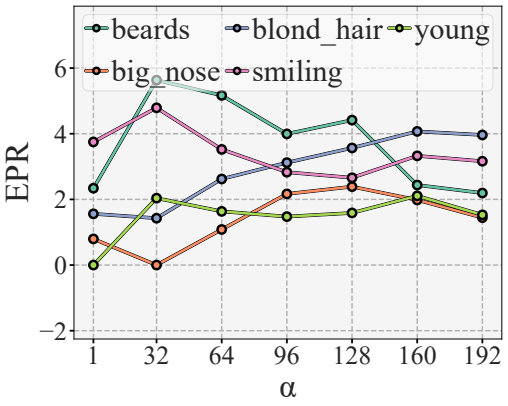


(a) top-$k = 1$

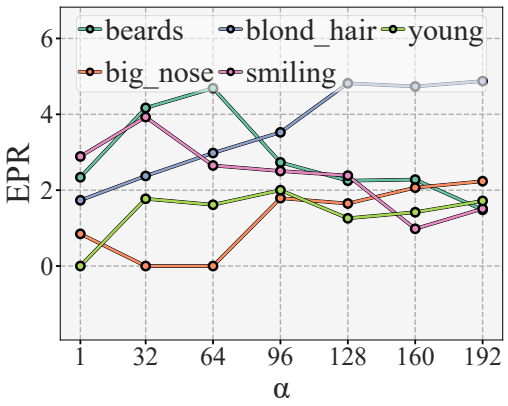
(b) top-$k = 2$



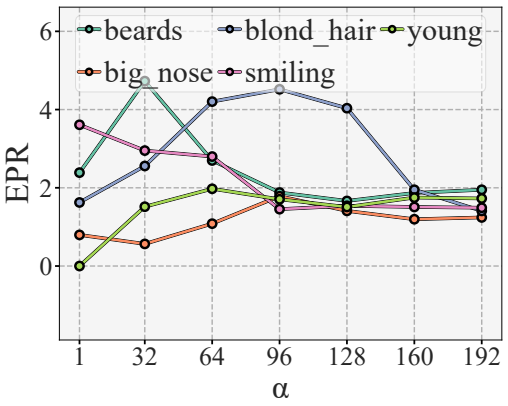
(c) top-$k = 4$



(d) top-$k = 8$

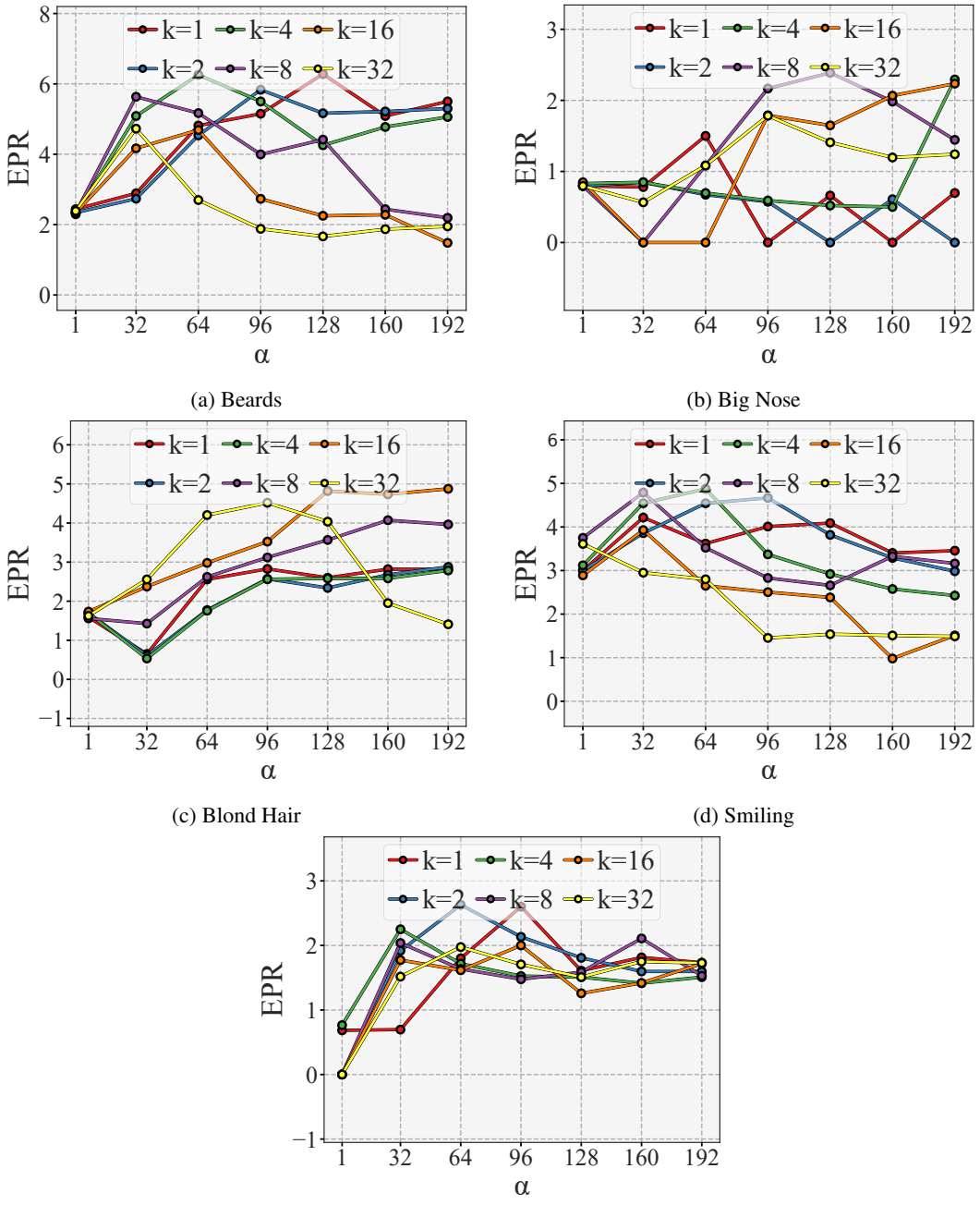


(e) top-$k = 16$



(f) top-$k = 32$

Figure 6: EPR vs. α across all top-k settings (all concepts).



(a) Beards

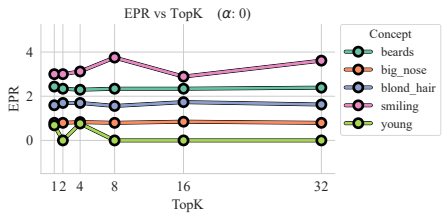
(b) Big Nose

(c) Blond Hair

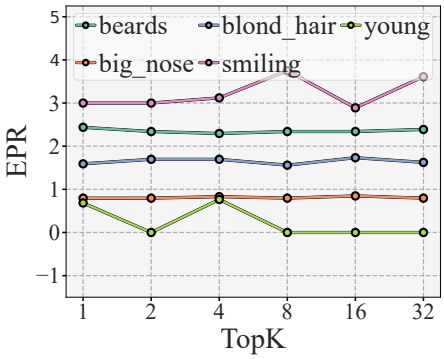
(d) Smiling

(e) Young

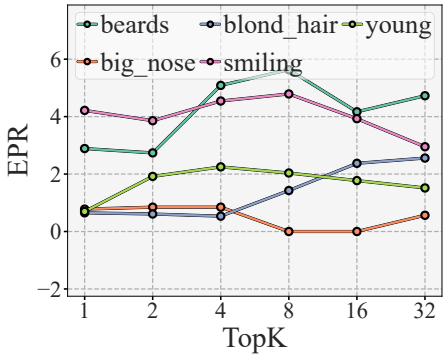
Figure 7: EPR vs. α per concept (all top-k settings).



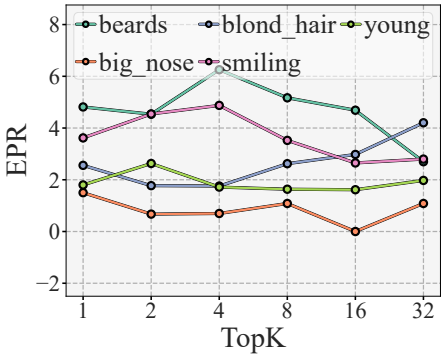
(a) $\alpha = 0$



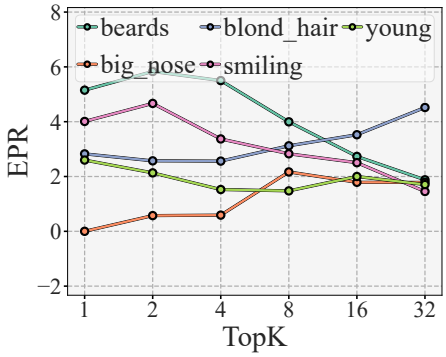
(b) $\alpha = 1$



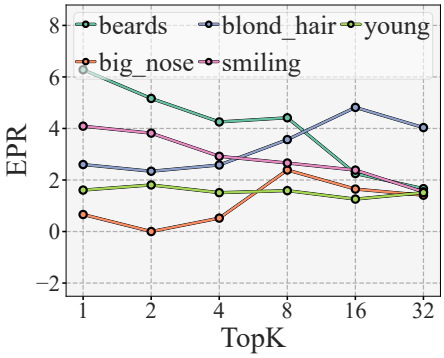
(c) $\alpha = 32$



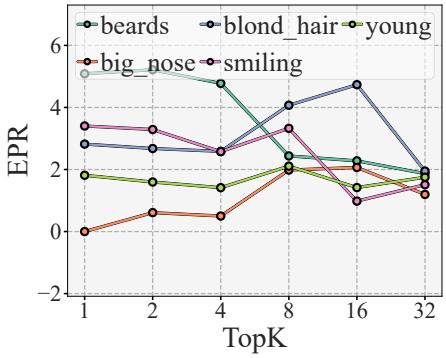
(d) $\alpha = 64$



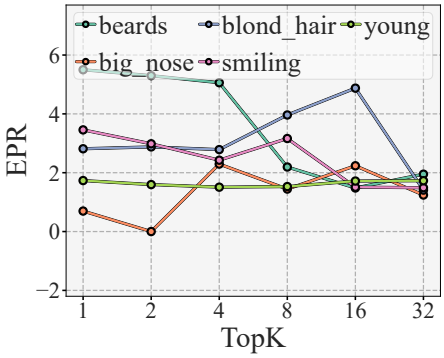
(e) $\alpha = 96$



(f) $\alpha = 128$



(g) $\alpha = 160$



(h) $\alpha = 192$

Figure 8: EPR vs. top-k across different α values (all concepts).

I Generalization to Large-Scale Latent Diffusion and Flow Matching

I.1 SDXL

We evaluate CASL on the frozen, text-conditioned SDXL-base-1.0 latent U-Net using 32 real CelebA-HQ images, 1024×1024 generation, a fixed neutral prompt, and 50-step DDIM inversion. A new SAE with 10,240 sparse features is trained on 4,096 target-domain images at the SDXL U-Net mid-block ($1280 \times 32 \times 32$); Stage 2 is trained on 64 images. The SDXL backbone is never trained or fine-tuned on FFHQ or CelebA. We explored 11 attributes and report the three concepts confirmed on 32 images in Table 24, comparing CASL with a dense h-space intervention at matched target strength. CASL gives the clearest advantage on Smiling, improving both specificity and preservation; Young shows successful but mixed editing; Blond Hair shows a favorable preservation and specificity trend under a small target change. Inversion and zero-strength tests confirm that the SDXL editing pipeline is stable.

Table 24: SDXL results, CASL / matched dense h-space intervention.

Concept	Target ↑	Off-target ↓	LPIPS ↓	ArcFace ↑	EPR ↑
Smiling	0.283 / 0.276	0.078 / 0.109	0.060 / 0.113	0.527 / 0.407	4.016 / 2.354
Young	0.082 / 0.083	0.068 / 0.063	0.049 / 0.053	0.697 / 0.720	1.483 / 1.624
Blond Hair	0.0006 / 0.0022	0.0236 / 0.0261	0.0128 / 0.0190	0.896 / 0.890	0.788 / 0.747

I.2 Flow-Matching DiT in u-space

Following [12], we replace the preliminary block-20 intervention of App. J with the pre-patchification u-space of a frozen latent DiT-L/2 trained by flow matching, $u_t = x_t \in \mathbb{R}^{4\times32\times32}$ with $\mathrm{d}x_t/\mathrm{d}t = v_\theta(x_t, t)$. We treat the 32×32 spatial locations as tokens and share a $4 \to 128 \to 4$ TopK SAE across them (a $32\times$ channel expansion per token), retraining the SAE and the CLIP-supervised Stage-2 map while keeping the flow backbone frozen. Let E_{SAE} be the token-wise encoder, P_{K_c} retain the top-k coordinates for concept c, and M_c map the sparse code back to the four u-space channels:

$$z_{t,p} = E_{\mathrm{SAE}}(u_{t,p}), \qquad \Delta u^c_{t,p} = M_c\big(P_{K_c} z_{t,p}\big), \tag{19}$$

where p indexes spatial location. M_c is trained with the directional CLIP loss of CASL, and its squared column norms rank the coordinates selected by P_{K_c}. We inject the shift before patchification and keep the original Euler solver:

$$\tilde{v}_t = v_\theta\big(x_t + \gamma_t \alpha \Delta u^c_t, t\big), \qquad \tilde{x}_{k+1} = \tilde{x}_k + \Delta t_k\, \tilde{v}_{t_k}. \tag{20}$$

Locally, $\tilde{v}_t - v_t \approx J_{v_\theta,x}\, \gamma_t \alpha \Delta u^c_t$, so the sparse displacement perturbs the frozen velocity field without changing the field or the integrator.

We evaluate real-image steering with 100-step Euler inversion and sampling on five concepts, using 32 target-attribute-absent CelebA-HQ images per concept; intervention settings are selected on a separate eight-image validation set. Edit metrics use the Euler reconstruction as reference, while LPIPS and ArcFace compare the reconstruction with the original image (the Euler reconstruction itself achieves LPIPS 0.133 and ArcFace 0.783). All four evaluation classifiers report positive target changes for every concept (Table 25).

Table 25: CASL on a flow-matching DiT-L/2 in u-space.

Concept	Target change ↑	EPR ↑	LPIPS ↓	ArcFace ↑
Smiling	0.036	1.469	0.072	0.921
Young	0.139	1.338	0.163	0.677
Beard	0.034	1.165	0.221	0.647
Blond Hair	0.110	1.550	0.144	0.761
Male	0.119	1.806	0.246	0.644
Macro	0.087	1.466	0.169	0.730

These results show that CASL is not specific to pixel-space diffusion, U-Nets, noise prediction, or DDIM sampling. We do not claim complete generator independence: applying CASL to a substantially different generator still requires an appropriate intervention space and propagation rule.

J Preliminary DiT Study: Concept-Associated Sparse Latents in Transformer Backbones

Note. This study uses an intermediate block (block 20) and is a representation-level sanity check only. A more principled intervention in the pre-patchification u-space of a flow-matching DiT, with full real-image steering results, is given in App. I.2.

Goal. This section provides a preliminary *representation-level* validation that our SAE+alignment pipeline is not specific to convolutional U-Net backbones. We test whether intermediate DiT representations can (i) be decomposed into sparse latents via SAE, and (ii) exhibit concept-specific linear separability and semantically coherent high-response samples. This experiment is intended as a sanity check rather than a full DiT editing evaluation.

J.1 Setup

Backbone and data. We use DiT-XL/2 (256×256) and extract activations from block 20 at diffusion timestep $t = 250$. Images are sampled from CelebA-HQ (1000 images, seed=42).

Activation pooling. The raw hidden representation at the hooked block has shape $[B, T, D]$. We apply mean pooling over tokens:

$$h = \frac{1}{T} \sum_{j=1}^{T} \text{hidden}_j \in \mathbb{R}^D, \tag{21}$$

yielding pooled activations of dimension $D = 1152$.

J.2 SAE Modeling and Linear Separability

We train an SAE on the pooled DiT activations with latent multiplier 32, producing sparse representations in $\mathbb{R}^{36864}$. The SAE converges and yields stable sparse codes.

For each CelebA-HQ attribute (e.g., Smiling, Male, Eyeglasses, Mustache, Young), we identify candidate latent dimensions via high average activation on positive samples and train a linear probe using the top-100 latent features.

Figure 9 shows the probe performance and AUC as a function of top-k features. Concept information remains linearly separable in the sparse latent space, suggesting that DiT representations admit concept-aligned sparse structure.

J.3 Overlap Analysis: Raw vs. Probe-Selected Latents

To assess whether supervised alignment improves concept specificity in DiT representations, we compare latent overlap under two selection strategies: (i) raw top-100 high-activation latents selected purely by average activation magnitude, and (ii) probe-selected top-10 latents ranked by absolute probe weight.

Figure 10 visualizes the Jaccard overlap among latent sets for different attributes. As shown in Fig. 10(a), raw high-activation latent sets exhibit substantial cross-concept overlap, indicating that unsupervised activation-based selection tends to capture shared facial factors rather than concept-specific structure. In contrast, Fig. 10(b) shows that probe-selected latent sets display markedly reduced overlap across concepts. This suggests that supervised selection helps isolate attribute-specific sparse dimensions within DiT representations, mirroring the behavior observed in U-Net backbones.

J.4 Qualitative High-Response Samples

To qualitatively validate semantic coherence, we compute an image-level score using the probe-selected top-1 latent dimension:

$$s(x) = w_k \, z_k(x), \tag{22}$$

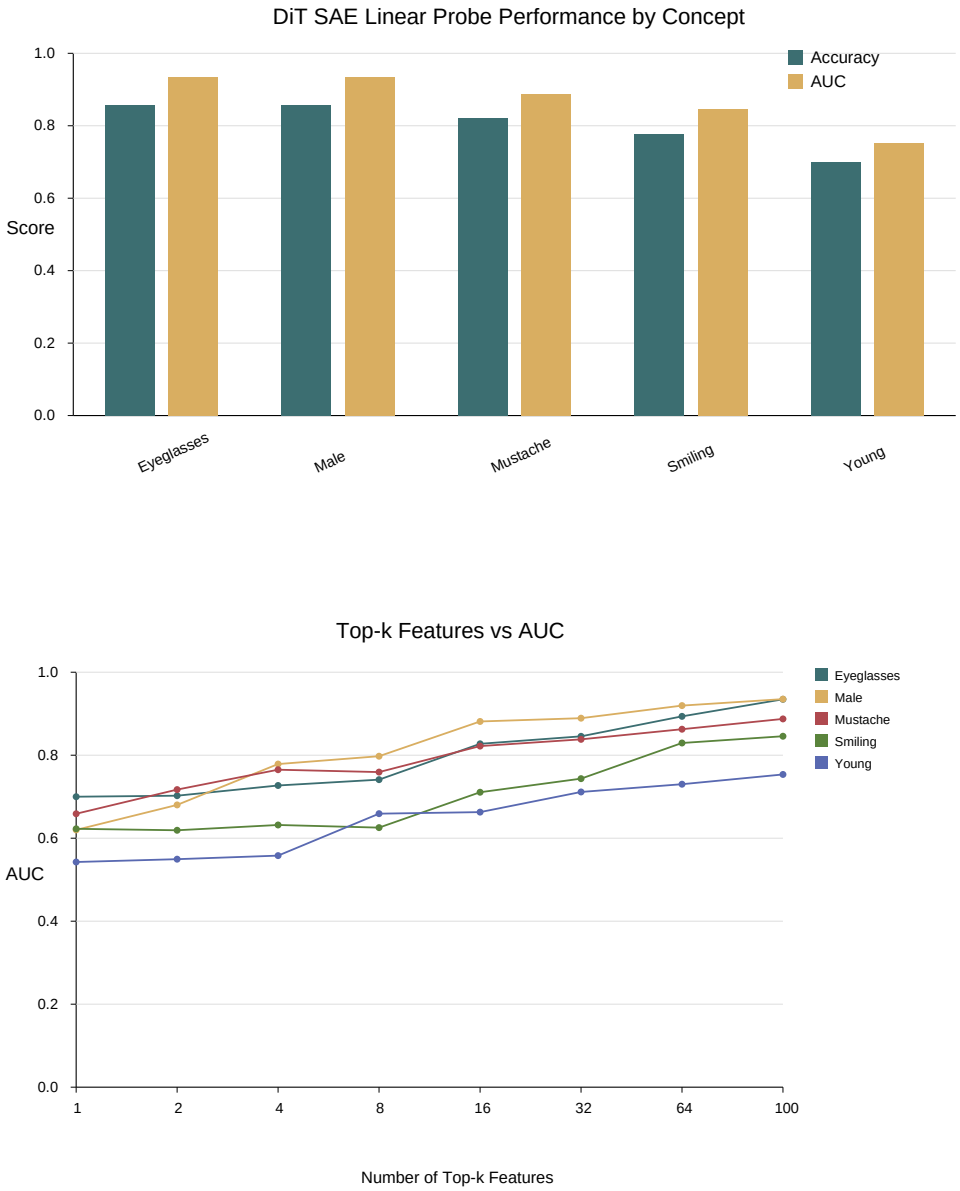
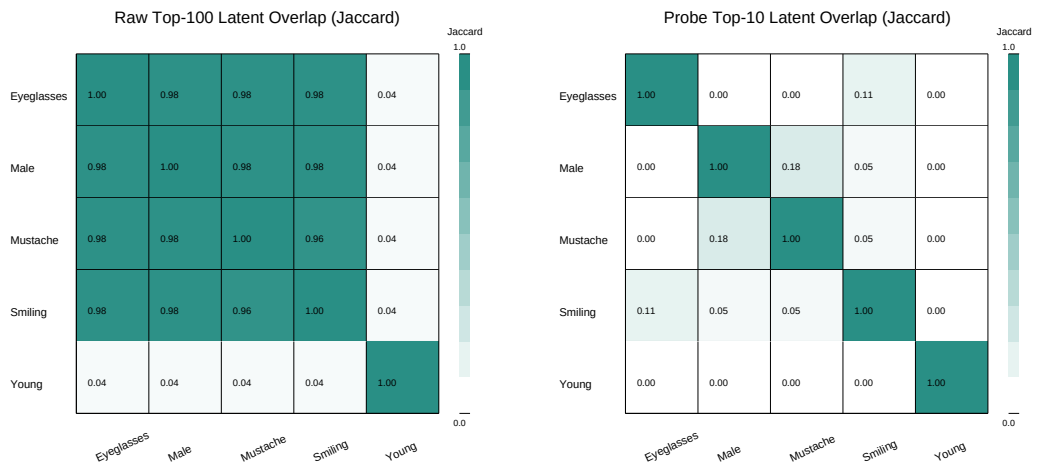


Figure 9: Linear probe performance on DiT SAE latents. Top: Accuracy and AUC across attributes. Bottom: AUC as a function of top-k probe-selected features.

where k denotes the latent dimension with the largest absolute probe weight and w_k its corresponding coefficient.

Images are ranked by $s(x)$, and for each concept we retrieve the top-32 highest-scoring samples. Figure 11 visualizes six representative examples from this top-ranked set.

As shown in Fig. 11, the highest-scoring images exhibit clear visual consistency with the target attribute (e.g., prominent eyewear for *Eyeglasses*, visible facial hair for *Mustache*, and youthful facial features for *Young*). This qualitative evidence supports the claim that probe-selected sparse latents in DiT representations capture semantically meaningful concept directions.



(a) Raw top-100 latents

(b) Probe-selected top-10 latents

Figure 10: Jaccard overlap among concept-specific latent sets. Unsupervised high-activation latents exhibit substantial cross-concept overlap, whereas probe-selected latents show markedly reduced overlap, indicating improved concept specificity under supervised alignment.

Summary. These results indicate that DiT intermediate representations admit sparse latent decomposition and contain concept-relevant directions that are linearly separable and visually coherent. This suggests that the core representation-level mechanism behind CASL is not specific to U-Net backbones.

K Visualization Results

Visualization results for the main editing tasks and qualitative comparisons can be found in `Human_Evaluation.pdf` included in the supplementary material.

L Computational Cost

CASL requires training the sparse autoencoder once on U-Net activations. On a single A100 GPU, SAE training takes approximately 6 hours with a memory footprint below 32 GB. The subsequent concept-mapping step is a lightweight linear regression and completes within minutes. CASL-Steer itself adds negligible cost at inference, since it injects a single latent shift during DDIM sampling. Overall, CASL introduces only a one-time offline training cost, and its per-edit overhead is comparable to standard DDIM generation.

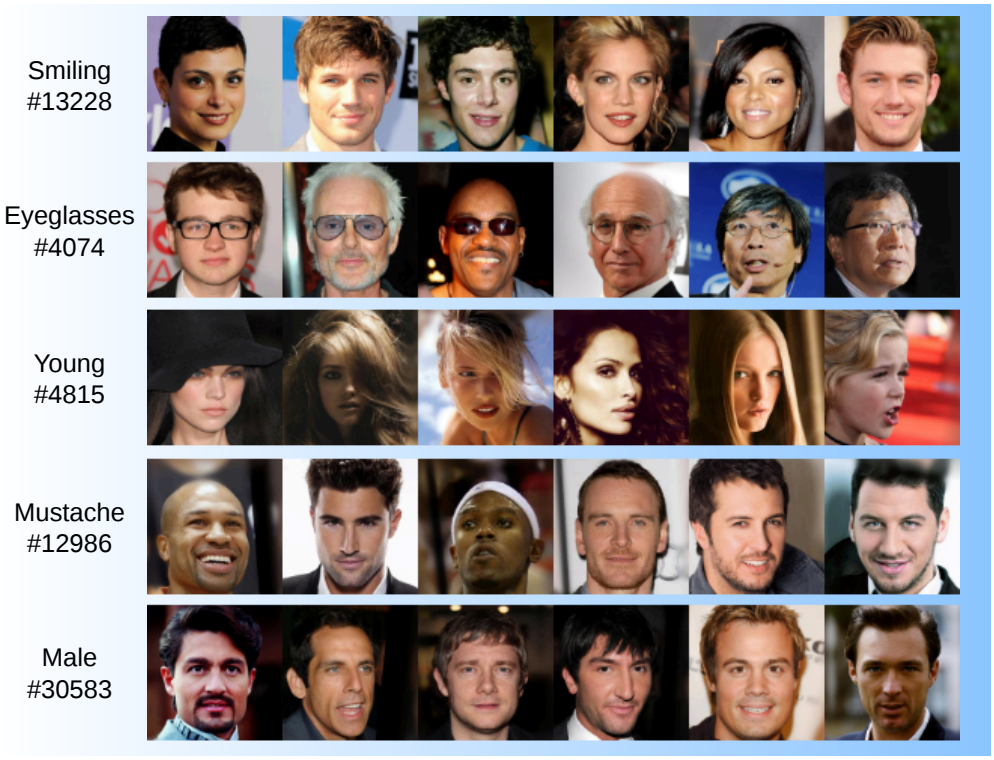


Figure 11: Representative high-response images selected from the top-32 samples ranked by the probe-selected top-1 latent for each concept.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately state the paper's main contributions: CASL aligns sparse latent dimensions in U-Net diffusion models with semantic concepts, uses CASL-Steer as a causal probe, and introduces EPR for precision-aware evaluation. The scope is also stated as U-Net-based unconditional diffusion models, with DiT results presented only as preliminary evidence.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed in App. A. CASL is primarily evaluated on unconditional U-Net diffusion models as a controlled setting; additional results on SDXL and a flow-matching DiT (App. I) cover fewer concepts, and applying CASL to a new generator still requires choosing a suitable intervention space. We also discuss the dependence of EPR on attribute labels and classifiers, and recurring failure modes such as background drift and identity degradation at large editing strengths.

   Guidelines:

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

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proofs. The mathematical formulations are used to define the method, latent intervention, and evaluation metric.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper specifies the datasets, diffusion backbones, activation extraction procedure, SAE training setup, concept-alignment procedure, evaluation metrics, and main hyperparameters needed to reproduce the main experimental results. Additional implementation details are provided in the appendix.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

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

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

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

   Answer: [Yes]

   Justification: Code is publicly released at the repository linked in the abstract, with instructions for reproducing the main experiments. All experiments use publicly available datasets and pretrained diffusion backbones cited in the paper (Sec. C).

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper reports the datasets, train/test usage, pretrained backbones, sampled timesteps, activation layer, SAE expansion ratio, sparsity coefficient, optimization details, evaluation metrics, and baseline methods. Additional training and implementation details are included in the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: The main quantitative results report mean and standard deviation for the evaluated metrics across sampled images. These error bars quantify variability across evaluation samples under the same experimental setting.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper reports the main hardware used for the experiments, including one NVIDIA A100 GPU with 80GB memory. The appendix provides additional implementation details needed to estimate the computational cost of training and evaluation.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research follows the NeurIPS Code of Ethics. It uses public datasets and pretrained models for interpretability analysis and does not involve deceptive data collection, human-subject experiments, or deployment on users.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The work may have positive impact by improving interpretability and auditing of generative models. It may also lower the barrier to more precise image manipulation, so misuse for deceptive or harmful image editing should be considered.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release a new high-risk pretrained generative model or a scraped dataset. The method is evaluated on existing public diffusion backbones and datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper cites the original sources for all datasets, pretrained diffusion backbones, and baseline methods used in the experiments. We use these assets according to their intended research-use settings and provide asset details in the appendix.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The new asset is our code, publicly released at the repository linked in the abstract and documented with instructions for training the SAE, learning the concept alignment, and reproducing the main results.

    Guidelines:

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

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

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

    Answer: [Yes]

    Justification: The paper includes a small user study with 10 volunteer participants who rated edited images (App. E). The appendix describes the instructions given to participants, the presentation of each sample (original image, target attribute, edited image), and the two rating criteria; the raw ratings are provided in the supplementary material. Participants were recruited on a voluntary basis and received no compensation.

Guidelines:

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

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

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

    Answer: [No]

    Justification: Formal IRB approval was not obtained for the user study. The study posed minimal risk: participants voluntarily rated edited images from public face datasets, and no personal or identifying information about the participants was collected.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not used as an important, original, or non-standard component of the core methodology. The method is based on sparse autoencoders, diffusion-model activations, CLIP-based semantic supervision, and classifier-based evaluation.

    Guidelines:

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