TA-SAE: Untangling Temporal Polysemanticity in Dictionary Learning for Rectified Flow Transformers

Anonymous Author(s)
Affiliation
Address
email

Abstract

As text-to-image rectified flow models become increasingly capable, understanding their internal representations is essential for interpretability, control, and model safety. While sparse autoencoders (SAEs) have emerged as a promising tool for extracting interpretable features, existing works do not shed light on how individual concepts evolve over time internally, due to a reliance on timestep-specific SAEs. We argue that the reliance on timestep-dependent SAEs is in large part due to a lack of any inductive bias in SAEs towards the generation trajectory, crippling their ability to be used across long trajectories. We show that this creates failure modes missed by standard SAE diagnostics - a feature can reconstruct well while changing semantic identity across denoising time, and a concept can be handed off across different features at different denoising stages. We explore each of these phenomena, which we refer to as temporal feature absorption and temporal feature splitting respectively. In addition, to combat them, we introduce trajectory-aware SAEs (TA-SAEs), which learn a single dictionary over denoising time while assigning each feature a learned persistence coefficient, regularizing persistent features to remain stable across adjacent timesteps while allowing transient features to capture time-local denoising computations. On FLUX.1-dev block-18 activations, we show that TA-SAE preserves standard SAE quality, reduces absorption and splitting, and improves temporal semantic stability. To our knowledge, we are the first to formalize temporal feature absorption and temporal feature splitting as feature–concept identity failures in diffusion SAEs, and propose an alternative to traditional SAEs to combat them.

1 Introduction

Text-to-image rectified flow models have rapidly become a foundation for visual content creation, image editing, design workflows, and synthetic data generation, making their behavior increasingly consequential beyond benchmark image quality. These models generate images by iteratively denoising an initial noisy latent into a final image under conditioning of a text prompt [Ho et al., 2020, Rombach et al., 2022, Lipman et al., 2023, Black Forest Labs, 2024].

As they become more widely used, understanding how they represent and manipulate visual concepts has shifted from being important to an urgent necessity for reliability, controllability, and safe deployment. A natural approach to interpreting these activations is dictionary learning. Under the linear representation hypothesis, model activations can often be understood as sparse superpositions of feature directions, where individual directions correspond to meaningful concepts or computations [Elhage et al., 2022, Bricken et al., 2023, Park et al., 2024]. Sparse autoencoders (SAEs) operationalize this idea by learning an overcomplete dictionary of features together with sparse codes that reconstruct model activations [Bricken et al., 2023, Cunningham et al., 2024, Gao et al.,

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

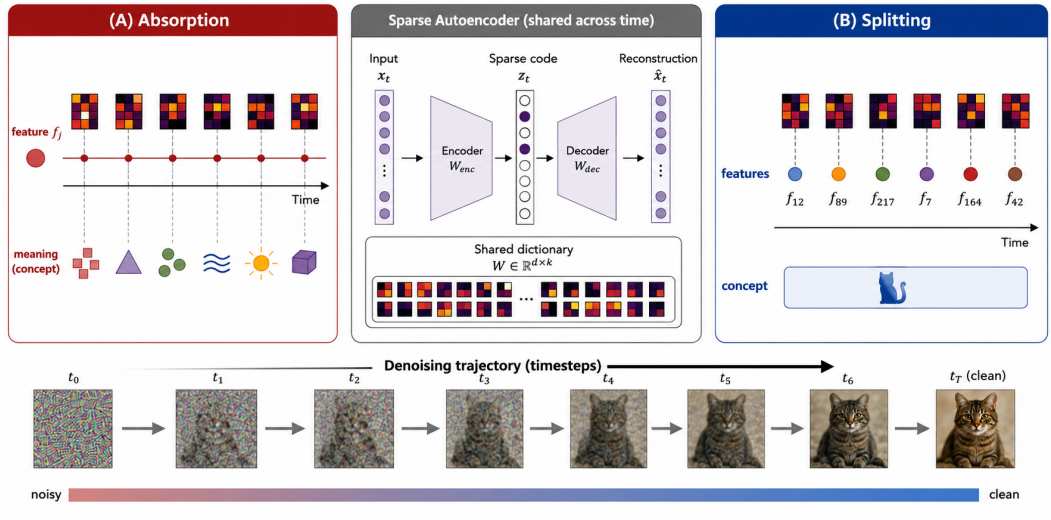


Figure 1: Forms of temporal polysemanticity in flow matching SAEs. Feature absorption involves a singular feature representing different concepts at different points in time, feature splitting involves a singular concept being represented by different features at different points in time.

2025]. In language models, this framework has enabled the discovery of interpretable features and has become a central tool in mechanistic interpretability. Recent work has begun to extend SAEs and related dictionary-learning tools to text-to-image diffusion models, revealing features useful for image editing, representation analysis, and block- or timestep-level understanding [Surkov et al., 2025, Tinaz et al., 2025, Kim et al., 2025].

However, an activation in a diffusion model is not only indexed by layer, but also by denoising time. Activations across timesteps are therefore not independent samples, but states along the same reverse trajectory, coupled through the evolving latent and shared conditioning signal. Recent work supports this, suggesting that semantic information in diffusion models is neither uniformly represented nor static across the generation trajectory, making understanding how concepts evolve over time increasingly important [Basu et al., 2024, Zarei et al., 2025, Kim et al., 2025]. Prior diffusion SAE analyses study how semantic information varies across layers and timesteps, yet are unable to do so at a concept level because of reliance on per-timestep SAEs, and as a consequence, are unequipped to answer how specific concepts change during generation. We posit that this is a core bottleneck for both understanding and control - without features that are aligned across denoising time, interpretability is restricted to static timestep snapshots, rather than the evolving trajectory over which a concept emerges, persists, transforms, or disappears.

We argue that a critical bottleneck to SAEs being applied to entire generation trajectories is *temporal polysemanticity*, an extension of the broader idea of polysemanticity in deep learning. To combat this, we introduce *trajectory-aware SAEs*, a variant of SAEs designed to learn a single dictionary over the denoising trajectory and regularize persistence continuously, learning to separate consistent semantic structure from transient denoising structure at feature-level granularity. We study our variant and compare with leading variants on FLUX.1-dev [Black Forest Labs, 2024], training SAEs on block 18 contributions, and compare them against state-of-the-art dictionary baselines. Our evaluation suite measures temporal semantic consistency, feature splitting and absorption, and cross-timestep probing transfer. Across these analyses, trajectory-aware objectives better distinguish stable semantic features from timestep-specific denoising structure, supporting the view that diffusion SAE features should be interpreted as time-conditioned objects rather than time-independent dictionary elements. Our contributions are as follows:

1. We identify *temporal polysemanticity* as a core failure mode for SAEs trained on multi-step diffusion activations, formalize temporal feature absorption and temporal feature splitting as two complementary forms of this failure, and propose novel methods to detect these phenomena at multiple hierarchical semantic levels.

2. We propose TA-SAE, a shared-dictionary SAE with learned feature-level persistence coefficients that regularize adjacent-step pre-sparsification activations, separating persistent semantic structure from transient denoising computations.
3. On FLUX.1-dev block-18 activations, we show that TA-SAE preserves reconstruction quality, reduces measured temporal absorption and splitting, and improves object-label decodability from SAE codes.

2 Related Work

Sparse autoencoders. SAEs extract interpretable features from superposed activations in language models [Elhage et al., 2022, Bricken et al., 2023, Cunningham et al., 2024, Templeton et al., 2024, Gao et al., 2025, Rajamanoharan et al., 2024a,b, Lieberum et al., 2024] and, more recently, vision and vision-language models [Fel et al., 2025, Lou et al., 2025, Lim et al., 2025, Dhimoïla et al., 2026, Papadimitriou et al., 2025]. These settings treat activations as exchangeable samples; our setting differs because diffusion activations are ordered along a denoising trajectory, so feature identity may depend on time. Most closely related is the Temporal SAE (T-SAE) of Bhalla et al. [2026b], which partitions language-model features into high-level (temporally consistent) and low-level (token-specific) groups via a contrastive objective. We adapt the temporal decomposition idea to diffusion, where the "sequence" is the denoising trajectory rather than a token sequence, and focus on diagnosing and measuring the failure modes that arise when this decomposition is absent. Feature splitting and absorption are known to complicate monosemanticity claims in the static setting [Cui et al., 2026, Pacela et al., 2026, Karvonen et al., 2025]; we study their temporal analogues.

SAEs for diffusion models. Surkov et al. [2025] train SAEs on single-step distilled diffusion residual updates, finding features that generalize across multi-step sampling and enable fine-grained editing. Tinaz et al. [2025] train per-timestep SAEs on Stable Diffusion, revealing a temporal hierarchy of concept emergence but fragmenting concepts across independent dictionaries. Kim et al. [2025] show that representation granularity varies non-linearly with depth and timestep across diffusion architectures. However, all of these train timestep-dependent SAEs, meaning the ability to understand how concepts evolve over time is severely stunted. Huang et al. [2026] introduce TIDE, which adds timestep-conditioned modulation to handle distributional shift across denoising steps. Our work is complementary: TIDE improves reconstruction across timesteps, while we ask whether individual features preserve their semantic identity along the trajectory.

Temporal structure in diffusion representations. Prior work localizes semantic information to particular layers, attention modules, and denoising stages [Basu et al., 2024,?, Zarei et al., 2025,?], and shows that diffusion representations support downstream tasks through structured low-dimensional dynamics [Wang et al., 2024, Li et al., 2025]. We build on this temporal view but shift the question from *where* semantic information lives to whether SAE features *stably represent* it across denoising time.

3 Methodology

3.1 Temporal concepts in flow matching

Text-to-image flow matching models generate images by learning a velocity field in latent space. Let $z_0 \sim p_0$ be a Gaussian noise latent, $z_1 \sim p_{\text{data}}$ an image latent, and c the text condition. Rectified flow uses the linear path

$$z_s = (1-s)z_0 + sz_1, \qquad s \in [0,1], \tag{1}$$

whose target velocity is $z_1 - z_0$. The model is trained with

$$\mathcal{L}_{\text{RF}}(\theta) = \mathbb{E}_{z_0, z_1, s, c}\left[\|v_\theta(z_s, s, c) - (z_1 - z_0)\|_2^2\right], \tag{2}$$

and sampling integrates the rectified flow ODE

$$\frac{dz_s}{ds} = v_\theta(z_s, s, c) \tag{3}$$

from noise toward the data endpoint, where v_θ is the learned velocity field and z_s the latent at flow time s.

However, although the target displacement $z_1 - z_0$ is constant for a fixed training pair, the model is not given that pair at inference - it only observes the intermediate latent z_s, the time s, and the condition c. Thus the optimal predictor is a posterior mean over all endpoints compatible with (z_s, s, c). Because $z_s = (1-s)z_0 + sz_1$ has a time-dependent signal-to-noise ratio, the posterior over possible endpoints changes along the trajectory. Early in generation, the estimate is dominated by prompt-conditioned and global semantic priors; later, as z_s contains more endpoint information, local image evidence such as color, texture, material, and fine spatial detail can become more predictive of the remaining velocity. Thus, different latent concepts can be useful at different steps during denoising, and prior work supports this, showing that early steps contain features that assist with coarse object structure and scene composition, while later layers focus on fine-grained details such as color and texture. This motivates our key assumption - that each endpoint latent–prompt pair is governed by a set of underlying semantic and visual factors, such as object identity, pose, layout, color, material, texture, and style, and that these concepts are not equally relevant for denoising throughout the trajectory.

We use "concept" to refer to these underlying visual properties and reserve "feature" for learned SAE dictionary elements. Under this view, an SAE feature is an imperfect learned proxy for one or more concepts. A single feature may absorb multiple concepts that are relevant at different times, or a single concept may split across multiple features active at different denoising stages. Thus, in order to correctly model the underlying concepts from this perspective, we must treat SAE features as time-dependent objects rather than static dictionary elements.

3.2 FLUX residual-stream activations

FLUX parameterizes the velocity field v_θ with a multimodal diffusion transformer. We use i to index a sample, t to index a solver step, $p \in [N]$ to index an image token, $r \in [M_i]$ to index a text token, and ℓ to index a transformer block. At solver step t, the current image latent for sample i is patchified into N image tokens, and the prompt is encoded into M_i text tokens. We do not introduce separate notation for the raw token embeddings, since our analysis is defined on image-token residual-stream updates.

FLUX first processes image and text tokens in multimodal double-stream blocks, then processes the concatenated sequence in single-stream transformer blocks. In both cases, attention allows text-token information to condition the image-token residual stream.

For block ℓ, sample i, solver step t, and image token p, let

$$h_{i,t,p}^{(\ell),\mathrm{in}}, \qquad h_{i,t,p}^{(\ell),\mathrm{out}} \in \mathbb{R}^d$$

denote the image-token residual state before and after the block. For double-stream blocks, this is the image-stream residual state; for single-stream blocks, it is the image-token slice of the joint residual stream. We analyze the block residual update

$$a_{i,t,p}^{(\ell)} = h_{i,t,p}^{(\ell),\mathrm{out}} - h_{i,t,p}^{(\ell),\mathrm{in}} \in \mathbb{R}^d. \tag{4}$$

This update is the computation added by block ℓ to image token p at solver step t. All SAE activations and temporal statistics below are indexed by this same tuple (i, t, p, ℓ).

3.3 Sparse autoencoders on residual updates

Before defining temporal absorption and splitting, we fix the SAE notation used throughout the paper. For a chosen block ℓ, we train SAEs on residual updates $a_{i,t,p}^{(\ell)} \in \mathbb{R}^D$ with F learned features. A TopK SAE encodes and reconstructs each residual update as

$$\begin{aligned} u_{i,t,p}^{(\ell)} &= \mathrm{ReLU}\left(W_{\mathrm{enc}}^{(\ell)}\left(a_{i,t,p}^{(\ell)} - b_{\mathrm{pre}}^{(\ell)}\right) + b_{\mathrm{enc}}^{(\ell)}\right), \\ z_{i,t,p}^{(\ell)} &= \mathrm{BatchTopK}\left(u_{i,t,p}^{(\ell)}\right), \\ \hat{a}_{i,t,p}^{(\ell)} &= W_{\mathrm{dec}}^{(\ell)} z_{i,t,p}^{(\ell)} + b_{\mathrm{pre}}^{(\ell)}. \end{aligned} \tag{5}$$

Here $u^{(\ell)}_{i,t,p} \in \mathbb{R}^F_{\geq 0}$ denotes the pre-sparsification feature activations, $z^{(\ell)}_{i,t,p} \in \mathbb{R}^F_{\geq 0}$ denotes the sparse SAE code, and the columns of $W^{(\ell)}_{\text{dec}}$ are SAE dictionary directions. $W^{(\ell)}_{\text{enc}}$ is the encoder weight, $b^{(\ell)}_{\text{enc}}$ is the encoder bias, and $b^{(\ell)}_{\text{pre}}$ is a learned pre-bias subtracted from the input and added back to the reconstruction.

Equivalently,

$$\hat{a}^{(\ell)}_{i,t,p} = \sum_{f=1}^{F} z^{(\ell)}_{i,t,p,f} d^{(\ell)}_f + b^{(\ell)}_{\text{pre}}, \tag{6}$$

where $d^{(\ell)}_f$ is the decoder direction for feature f, and $z^{(\ell)}_{i,t,p,f}$ is its activation. We write

$$\mathcal{L}_{\text{SAE}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{aux}}$$

for the standard SAE objective, consisting of normalized reconstruction loss and a dead-feature recovery term. Implementation details are given in the appendix.

3.4 Measuring temporal feature–concept identity

To measure how SAE semantics change over denoising time, we associate feature activations with token-level visual concepts in coarse temporal bins. Let $\mathcal{B} = \{B_1, \ldots, B_m\}$ partition the solver trajectory into temporal bands, where each band is a set of denoising steps pooled for analysis (e.g., early, middle, and late denoising). Let $\mathcal{Y}$ denote a concept basis: the vocabulary of visual labels used to assign semantic mass to image tokens. In our experiments, $\mathcal{Y}$ is given by COCO panoptic object categories, DINOv3 clusters labeled by k-means, or DINOv3 clusters labeled by their dominant COCO panoptic category. Details are given in Appendix A.3.

For sample i, image token p, and concept $q \in \mathcal{Y}$, let $\pi^{\mathcal{Y}}_{i,p,q} \geq 0$ be the amount of concept q assigned to token p. This is an indicator for hard labels and patch-overlap mass for panoptic masks; unlabeled tokens have zero mass for all q. Let $z^{(\ell)}_{i,t,p,f}$ be the activation of SAE feature f at block ℓ, solver step t, and token p. We accumulate the activation-weighted concept count

$$M^{(\ell),\mathcal{Y}}_{f,q,b} = \frac{1}{|B_b|} \sum_i \sum_{t \in B_b} \sum_p z^{(\ell)}_{i,t,p,f} \pi^{\mathcal{Y}}_{i,p,q}. \tag{7}$$

For a fixed block ℓ, we suppress ℓ in the notation and normalize this mass in two ways:

$$P_{\mathcal{Y}}(q \mid f, b) = \frac{M^{\mathcal{Y}}_{f,q,b}}{\sum_{q' \in \mathcal{Y}} M^{\mathcal{Y}}_{f,q',b}}, \qquad P_{\mathcal{Y}}(f \mid q, b) = \frac{M^{\mathcal{Y}}_{f,q,b}}{\sum_{f'} M^{\mathcal{Y}}_{f',q,b}}, \tag{8}$$

whenever the corresponding denominator is nonzero. The first distribution measures where feature f's activation mass falls in concept space during band b; the second measures which features carry activation mass for concept q during band b.

We now introduce temporal feature absorption and temporal feature splitting, both stemming from more general failure modes in SAEs. Temporal feature absorption refers to feature-side temporal polysemanticity - a stable feature has similar distributions $P_{\mathcal{Y}}(\cdot \mid f, b)$ across bands, whereas a temporally absorbed feature shifts its concept distribution over time, indicating that a single SAE feature carries multiple temporally separated factors. Temporal feature splitting is concept-side temporal polysemanticity - a stable concept is carried by similar feature distributions $P_{\mathcal{Y}}(\cdot \mid q, b)$ across bands, whereas a split concept undergoes temporal handoff - different SAE feature subsets carry the same concept at different denoising times.

Section 4 uses these conditional distributions to compare absorption and splitting across SAE variants, blocks, and concept bases. Scalar summaries, thresholds, and implementation details are given in Appendix A.

3.5 TA-SAE

The temporal-concept view suggests that an SAE should distinguish features that persist across much of the denoising trajectory from features that are useful only within a local temporal window.

Prior work [Bhalla et al., 2026b] has accomplished this analogously in large language models by explicitly allocating a fixed fraction of features to high-level, persistent, and semantic structure and the remainder to transient, low-level, and local structure. However, we argue that in image generation, underlying concepts are not just discretely persistent or transient, but rather lie on a spectrum between the two. An example could be a concept of fur on a dog - while it might only be active in later timesteps, it might still be active over an extended sequence of steps in the latter section of the trajectory, making it neither completely persistent nor completely transient. Thus, we propose treating the severity of temporal regularization as a parameter per-feature that the model learns.

To model this, TA-SAE keeps the SAE backbone from Section 3.3 and adds a learned persistence coefficient for each feature:

$$\tau_f^{(\ell)} = \sigma(\rho_f^{(\ell)}) \in (0,1), \qquad f = 1, \ldots, F.$$

The coefficient $\tau_f^{(\ell)}$ does not gate the code or decoder during the forward pass. Instead, it controls how strongly feature f is encouraged to fire consistently across adjacent solver steps.

Using the pre-sparsification activation $u_{i,t,p,f}^{(\ell)}$, TA-SAE adds a normalized adjacent-step change penalty:

$$\mathcal{L}_{\text{TA-SAE}} = \mathcal{L}_{\text{SAE}} + \lambda_{\text{traj}} \mathbb{E}_{i,p,t,f} \left[\tau_f^{(\ell)} \frac{\left| u_{i,t+1,p,f}^{(\ell)} - u_{i,t,p,f}^{(\ell)} \right|}{\left| u_{i,t+1,p,f}^{(\ell)} \right| + \left| u_{i,t,p,f}^{(\ell)} \right| + \epsilon} \right] + \lambda_{\text{prior}} \frac{1}{F} \sum_{f=1}^{F} -\log(\tau_f^{(\ell)} + \epsilon).$$

The trajectory term pressures high-$\tau_f^{(\ell)}$ features to remain stable on the same image token across neighboring solver steps, while the prior prevents the trivial solution where all features set $\tau_f^{(\ell)}$ near zero. This allows us to place features on a spectrum between persistent and transient, rather than denote features as entirely persistent and entirely transient.

4 Experiments

To understand whether trajectory-aware sparse autoencoders reduce temporal polysemanticity, we conduct experiments surrounding temporal feature absorption, temporal feature splitting, and probing. Full experiment methodology and results are deferred to Appendix A.

4.1 Training

We train three main SAE variants on the residual-stream of block-18 in FLUX, which we find to give substantially better reconstruction metrics after a sweep over all layers, with reconstruction quality severely degrading at later layers. We train over FLUX generations from 100k randomly sampled prompts from LAION-COCO-Aesthetic [Schuhmann et al., 2022]. The three main variants we choose are as follows: **BatchTopK**, **Matryoshka** (Matryoshka SAE with 2 levels, a higher level with 9216 features, and a lower level with 3072 features), and **TA-SAE**. All SAEs use K=32 with a d=12,288 dimension dictionary unless noted. Table 1 reports the standard SAE quality diagnostics together with the headline temporal-stability and probing metrics that the rest of Sec. 4 dissects. We find that TA-SAE performs at baseline or better on all core SAE metrics, meaning that it does not sacrifice core SAE performance to improve temporal consistency. In addition, we find that TA-SAE is interestingly able to significantly raise the number of alive features.

We also include two training ablations – a sparsity sweep over K and a trajectory-loss strength sweep over λ_{traj} – that confirm TA-SAE tracks the same reconstruction–sparsity frontier as the baselines and that reconstruction stays essentially flat for $\lambda_{\text{traj}} \leq 1$. Full curves and tables are deferred to Appendix A.2; these ablations are used only to characterize training tradeoffs and all semantic conclusions below use the main trained variants.

Because TA-SAE explicitly regularizes adjacent-step feature activations, a natural concern is that its gains might come from simply suppressing temporal variation. We therefore compare trajectory length in two spaces. First, we measure arc length in the raw SAE code space, which captures how much the sparse feature representation changes across denoising bands. Second, for each feature f and denoising band b, we compute a DINOv3 centroid $\gamma_f(b)$ by weighting each patch the feature

Table 1: Reconstruction and temporal-identity summary on FLUX.1-dev block-18 activations. Absorption is the fraction of eligible features whose dominant concept changes across valid denoising bands; Split JS is the temporal Jensen–Shannon distance of concept-conditioned feature distributions. Lower is better except for R^2 and alive fraction. Reported $\pm$ values are 95% bootstrap CIs (5,000 resamples; features for absorption columns, concepts for Split JS). R^2, alive fraction, and COCO Abs are point estimates only; the first two require per-image SSE/alive masks for CIs and the third requires the per-feature-eligibility flag in the per-(image) breakdown.

Variant	R^2 ↑	Alive Frac.↑	COCO Abs.↓	DINO→COCO Abs.↓	DINO KMeans Abs.↓	COCO Split JS↓
BatchTopK	0.944	0.489	0.558	0.356 $\pm$0.011	0.119 $\pm$0.007	0.115 $\pm$0.001
Matryoshka	0.946	0.637	0.600	0.324 $\pm$0.011	0.122 $\pm$0.007	**0.103** $\pm$0.001
TA-SAE	**0.949**	**0.734**	**0.436**	**0.233** $\pm$0.008	**0.096** $\pm$0.005	**0.103** $\pm$0.001

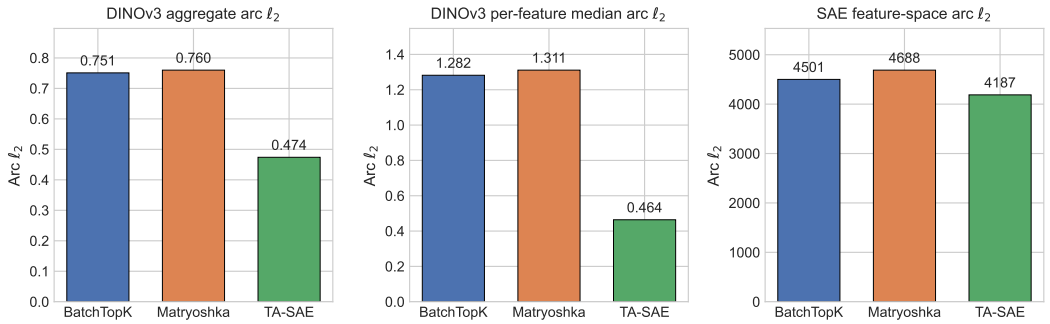


Figure 2: Trajectory arc length across the three main variants under three measurements. In the left chart, we average $\gamma_f(b)$ over features f before calculating ℓ_2 distance between adjacent bands. In the middle, we calculate ℓ_2 of ℓ_2 between adjacent bands per feature, average over bands, and then take the median over all 12288 scalars. In the right figure, we take the raw l2 between latent codes of adjacent bands and sum over bands.

fires on by its activation strength, and measure the arc length of this trajectory in DINOv3 space. The raw SAE-code arc length remains comparable across variants, indicating that TA-SAE does not merely make all feature activations constant. In contrast, DINOv3-space arc length decreases sharply for TA-SAE. Thus, TA-SAE selectively reduces movement in an external semantic representation - features can still vary over denoising time, but the visual concepts they track drift less. This supports our central claim that trajectory-aware regularization improves temporal semantic stability rather than only optimizing a smoothness surrogate.

4.2 Temporal polysemanticity

We evaluate feature-side temporal absorption using $P_{\mathcal{Y}}(q \mid f, b)$ from Section 3.4. While traditionally concept labeling for visual features has relied on object-level labels, we argue that the concept space for visual features in image generation is not fully spanned by object labels, completely ignoring concepts like texture, color, brightness, edges, and spatial layout. Thus, we instantiate the concept basis $\mathcal{Y}$ at three different hierarchies to encapsulate a larger range of visual concepts. We first use object-level panoptic segmentation masks by projecting the mask onto the patch grid: each patch receives COCO object mass from its mask overlap, with hard labels obtained by majority vote and void patches excluded. Feature activations then weight these patch labels to estimate $p(\text{object} \mid f, b)$ for each feature f and denoising band b. In order to further capture visual concepts beyond object-level labels, we also calculate DINOv3 [Simeoni et al., 2025] embeddings for every patch token, assign each patch to one of $K = 512$ DINOv3 k-means centroids, and measure absorption over the resulting unsupervised visual clusters. Finally, since raw k-means clusters can be jagged and are not

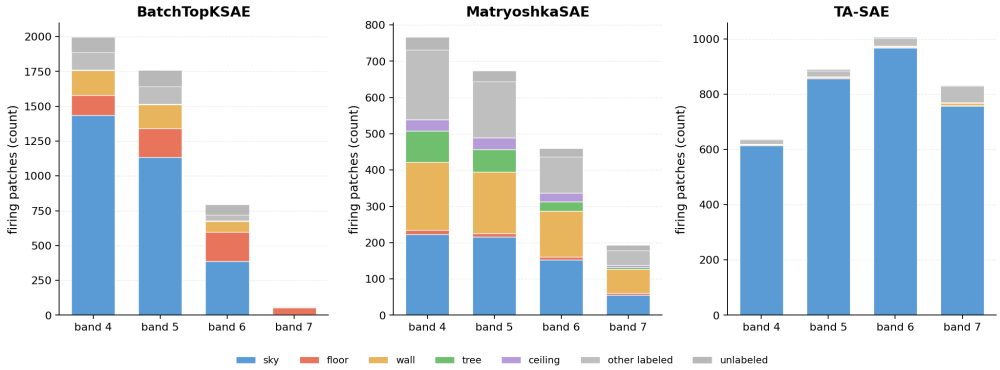


Figure 3: Qualitative "sky" case study. Each row shows the top sky-aligned feature of one variant (BatchTopK #2195, Matryoshka #2570, TA-SAE #466) across denoising bands on the validation set, with patch-level activations overlaid on COCO sky masks. BatchTopK and Matryoshka peak at band 4 and then drop off both in magnitude and in agreement with the sky mask. TA-SAE #466 fires more smoothly across the trajectory and stays concentrated on sky-labeled patches throughout, illustrating the temporal stability gain of the persistence prior.

guaranteed to correspond to interpretable concepts, we add a third basis by clustering each patch into a COCO panoptic object and recomputing absorption in this DINOv3→COCO basis.

Table 1 reports the three corresponding metrics: COCO Absorption, COCO Cluster Absorption, and DINOv3 KMeans Absorption. Across these views, we find that TA-SAE shows strict improvements across all 3 concept bases we evaluate, indicating that TA-SAE is able to not only increase temporal stability for features that correspond with human-interpretable objects, but also help stabilize features across the full visual hierarchy. We also notice that MatryoshkaSAE is able to reduce feature absorption in DINOv3-based concept bases compared to BatchTopK, but not on COCO's object concept base.

We surface a qualitative example in Figure 3, highlighting BatchTopKSAE feature 2195, MatryoshkaSAE feature 2570, and TA-SAE feature 466. Each feature fires strongly at band 4 on validation-set patches labeled as sky by COCO. However, as denoising progresses, the features not only fire significantly less, but stop firing on patches labeled sky. In contrast, TA-SAE continues to fire at smoother levels over the trajectory, and more importantly, consistently fires at a high majority on patches associated with the sky.

In addition, we evaluate whether TA-SAE helps reduce temporal feature splitting. To test this, we compute concept-side splitting over the same three concept bases from Section 4.2. For each SAE, concept q, and denoising band b, we form the alive feature distribution $P_{\mathcal{Y}}(\cdot \mid q, b)$ from weighted top-k patch activations. For each concept q, we calculate the pairwise Jensen-Shannon distance over all denoising bands b, and average over each concept, giving us a single metric that tells us how much a single concept's distribution of features changes as the generation trajectory progresses, displayed in Table 1. TA-SAE reduces temporal splitting relative to BatchTopK and matches Matryoshka on this metric. Overall, TA-SAE consistently reduces our measured forms of temporal polysemanticity without sacrificing, and in some cases improving, standard SAE diagnostics.

Finally, we find that the learned persistence coefficients are also aligned with external semantic stability. Among TA-SAE features that pass the trajectory-geometry eligibility filter, τ_f is negatively correlated with DINOv3 trajectory length and curvature, as shown in Appendix A.8. Thus, features that TA-SAE learns to treat as persistent are also the features whose activation-weighted visual centroids move less in an external representation space.

4.3 SAE-code probing

We end with a lightweight probe from SAE codes to COCO panoptic labels. Unlike the absorption metrics, which directly measure temporal identity changes, the probe asks whether object information remains linearly decodable within and across denoising bands. Figure 4 reports the top-class

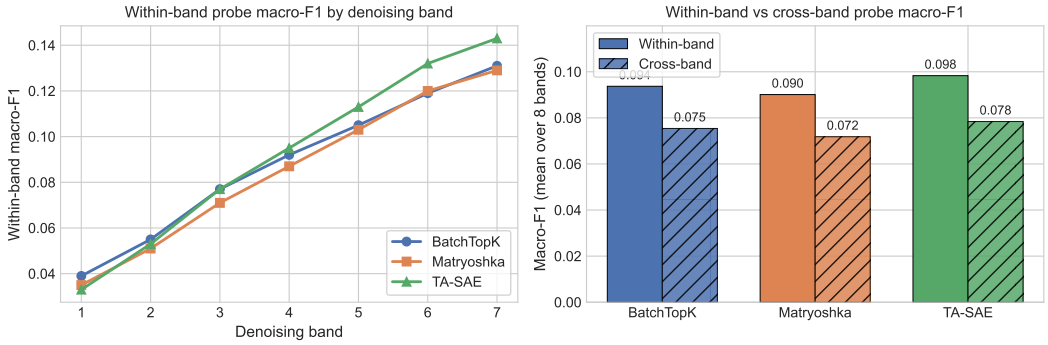


Figure 4: SAE-code linear probe macro-F1 (top-30 COCO panoptic). **Left**: within-band macro-F1 across denoising bands 1–7 (probe trained AND evaluated at the same band); all three variants improve monotonically and TA-SAE leads from band 3 onward. **Right**: aggregate within-band (solid) vs cross-band (hatched) macro-F1 per variant, averaged over all 7 train/eval band pairs. TA-SAE leads on both: cross-band transfer retains ∼80% of the within-band signal for all three variants, indicating that the codes retain decodable object information even when individual features drift in temporal identity.

probe results. TA-SAE gives the strongest probe performance, with both MatryoshkaSAE and BatchTopKSAE close behind. Cross-band transfer retains most of the within-band signal, consistent with the view that the codes retain object information even when individual features differ in temporal stability. Figure 4 plots the per-band trajectory: macro-F1 grows monotonically with band index for all three variants (later bands carry more object structure), and TA-SAE pulls ahead from band 3 onward, with the largest margin at the latest bands where the object signal is strongest.

Taken together, the experiments give a consistent picture. TA-SAE most reliably improves feature trajectory smoothness, reduces feature-level object absorption, stabilizes distributed concept ensembles, and preserves broad object-feature allocation. Cluster-level absorption provides complementary evidence that temporal regularization improves object stability beyond the COCO feature-labeling pipeline.

5 Limitations and Future Work

Our study has several limitations. First, the main experiments focus on FLUX.1-dev block 18, chosen because it provided strong reconstruction quality while still containing substantial semantic signal. We therefore view the results as evidence that temporal feature identity failures occur in flow matching SAEs, not as a complete layerwise characterization of FLUX. Second, our concept bases are proxies: COCO panoptic labels are interpretable but omit many visual factors, while DINOv3 clusters capture broader visual structure but are less directly human-readable. We address this by reporting all three bases rather than relying on a single labeling scheme. Third, our evaluations are primarily diagnostic rather than causal. A natural next step is to test whether temporally stable features provide better editing and steering handles across solver windows.

6 Conclusion

Diffusion SAE features should not be assumed to have time-independent semantics. We identified two feature–concept identity failures that can arise when a shared SAE is trained over denoising time: temporal feature absorption, where a feature's associated concepts change across denoising stages, and temporal feature splitting, where a concept is carried by different feature subsets at different stages.

We introduced TA-SAE, a shared-dictionary SAE with learned feature-level persistence coefficients that regularize adjacent-step pre-sparsification activations. On FLUX.1-dev block-18 residual updates, TA-SAE preserves standard reconstruction quality while reducing measured absorption and splitting

across COCO- and DINOv3-based concept bases, and modestly improves object-label decodability from SAE codes.

These results support a temporal view of monosemanticity for flow-matching models: a feature that is interpretable at one solver step may not represent the same concept later in the trajectory. Future dictionary-learning methods for generative models should therefore evaluate not only whether features are interpretable locally, but whether their semantics persist over the denoising trajectory they are intended to explain.

References

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Henighan, Tom Brown, Dario Amodei, Sam McCandlish, Jack Clark, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

Black Forest Labs. FLUX.1-dev: A 12-billion parameter rectified flow transformer. Technical report, 2024.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W. Gordon, Ross Wightman, Mehdi Cherti, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, et al. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations (ICLR)*, 2025.

Dahye Kim, Xavier Thomas, and Deepti Ghadiyaram. Revelio: Interpreting and leveraging semantic information in diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, et al. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

Oriane Simeoni, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Viacheslav Surkov, Chris Wendler, Antonio Mari, Mikhail Terekhov, Justin Deschenaux, Robert West, et al. One-step is enough: Sparse autoencoders for text-to-image diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Berk Tinaz, Zalan Fabian, and Mahdi Soltanolkotabi. Emergence and evolution of interpretable concepts in diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Spotlight.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Isabel Papadimitriou, Huangyuan Su, Thomas Fel, Sham Kakade, and Stephanie Gil. Interpreting the linear structure of vision–language model embedding spaces. In *Conference on Language Modeling (COLM)*, 2025.

Thomas Fel, Ekdeep Singh Lubana, Jacob S. Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, et al. Archetypal SAE: Adaptive and stable dictionary learning for concept extraction in large vision models. In *International Conference on Machine Learning (ICML)*, 2025.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Grégoire Dhimoïla, Thomas Fel, Victor Boutin, and Agustin Picard. Cross-modal redundancy and the geometry of vision–language embeddings. In *International Conference on Learning Representations (ICLR)*, 2026.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, et al. Toy models of superposition. *Transformer Circuits Thread*, 2022.

Xiao Li, Zekai Zhang, Xiang Li, Siyi Chen, Zhihui Zhu, Peng Wang, et al. Understanding representation dynamics of diffusion models via low-dimensional modeling. *arXiv preprint arXiv:2502.05743*, 2025.

Hyesu Lim, Jinho Choi, Jaegul Choo, and Steffen Schneider. Sparse autoencoders reveal selective remapping of visual concepts during adaptation. In *International Conference on Learning Representations (ICLR)*, 2025.

Jingyi Cui, Qi Zhang, Yifei Wang, and Yisen Wang. On the limits of sparse autoencoders: A theoretical framework and reweighted remedy. In *International Conference on Learning Representations (ICLR)*, 2026.

Samyadeep Basu, Nanxuan Zhao, Vlad Morariu, Soheil Feizi, and Varun Manjunatha. Localizing and editing knowledge in text-to-image generative models. In *International Conference on Learning Representations (ICLR)*, 2024.

Arman Zarei, Samyadeep Basu, Keivan Rezaei, Zihao Lin, Sayan Nag, and Soheil Feizi. Localizing knowledge in diffusion transformers. *arXiv preprint arXiv:2505.18832*, 2025.

Hantao Lou, Changye Li, Jiaming Ji, and Yaodong Yang. SAE-V: Interpreting multimodal models for enhanced alignment. *arXiv preprint arXiv:2502.17514*, 2025.

Samyadeep Basu, Keivan Rezaei, Priyatham Kattakinda, Vlad Morariu, Nanxuan Zhao, Ryan A. Rossi, et al. On mechanistic knowledge localization in text-to-image generative models. In *International Conference on Machine Learning (ICML)*, 2024.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, et al. Gemma scope: Open sparse autoencoders everywhere all at once on Gemma 2. *arXiv preprint arXiv:2408.05147*, 2024.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*, 2024.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, et al. Jumping ahead: Improving reconstruction fidelity with JumpReLU sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, et al SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *International Conference on Machine Learning (ICML)*, 2025.

Arman Zarei, Samyadeep Basu, Mobina Pournemat, Sayan Nag, Ryan Rossi, and Soheil Feizi. SliderEdit: Continuous image editing with fine-grained instruction control. *arXiv preprint arXiv:2511.09715*, 2025.

Vitória Barin Pacela, Shruti Joshi, Isabela Camacho, Simon Lacoste-Julien, and David Klindt. Stop probing, start coding: Why linear probes and sparse autoencoders fail at compositional generalisation. *arXiv preprint arXiv:2603.28744*, 2026.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, et al. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024.

Usha Bhalla, Alex Oesterling, Claudio Mayrink Verdun, Himabindu Lakkaraju, and Flavio P. Calmon. Temporal sparse autoencoders: Leveraging the sequential nature of language for interpretability. In *International Conference on Learning Representations (ICLR)*, 2026.

Victor Shea-Jay Huang, Le Zhuo, Yi Xin, Zhaokai Wang, Fu-Yun Wang, Yuchi Wang, Renrui Zhang, Peng Gao, and Hongsheng Li. TIDE: Temporal-aware sparse autoencoders for interpretable diffusion transformers in image generation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

Peng Wang, Huijie Zhang, Zekai Zhang, Siyi Chen, Yi Ma, and Qing Qu. Diffusion models learn low-dimensional distributions via subspace clustering. *arXiv preprint arXiv:2409.02426*, 2024.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning (ICML)*, 2024.

A Appendix

This appendix records the experimental setup, evaluation methodology, and numerical results used in the paper. Unless otherwise stated, reported values use the TEST split of the COCO panoptic probe set.

A.1 Common setup

Activations and denoising schedule. All sparse autoencoders operate on the *residual update* produced by transformer block 18 of FLUX.1-dev, defined as block output minus block input ($a^{(18)} = h^{(18),\text{out}} - h^{(18),\text{in}}$, matching Section 3). The hook captures this per-block residual contribution on the joint multimodal stream and is restricted to the 1,024 image-token positions. Each token is 3072-dimensional and is captured at bfloat16. Sampling uses the FlowMatchEulerDiscrete scheduler with 28 denoising timesteps and the production guidance scalar of 3.5. The 28 timesteps are partitioned into 7 contiguous denoising bands, giving the band sizes $(4, 4, 4, 4, 4, 4, 4)$. Band 1 contains the highest-σ steps and band 7 contains the lowest-σ steps. The mapping from steps to bands together with the σ summary is shown in Table 2.

Band	Steps	Sigma range
1	0–3	$[0.938, 1.000]$
2	4–7	$[0.843, 0.915]$
3	8–11	$[0.732, 0.817]$
4	12–15	$[0.601, 0.702]$
5	16–19	$[0.442, 0.564]$
6	20–23	$[0.248, 0.398]$
7	24–27	$[0.002, 0.192]$

Table 2: Seven-band partition of the 28-step FlowMatchEulerDiscrete schedule.

Per-step running statistics. Each SAE is paired with per-step running statistics computed at training time. For each step index $t \in \{0, \ldots, 27\}$, the running statistics record a per-feature mean vector and a single scalar standard deviation. At evaluation time the captured activation h_t is normalized as $x_t = (h_t - \mu_t)/\sigma_t$ before being fed to the SAE encoder, and the SAE reconstruction is reported in this normalized space.

Evaluation. We do all evaluation and post-training experiments on the COCO 2017 validation set. The set contains 2,000 images drawn from the COCO 2017 panoptic validation pool, partitioned into TRAIN, VAL, and TEST splits with a 60/20/20 ratio yielding 1,200/400/400 images respectively. Each image is rendered at 512×512 pixels and produces a 32×32 patch grid of 1,024 patches. Per-patch panoptic labels are obtained by decoding the COCO panoptic PNG into per-pixel category ids, taking the majority category over each patch's 16×16 pixel block as the hard label, and tracking pixels with category id zero as a separate void mass. Soft per-patch labels record the fractional

pixel area covered by each panoptic category in each patch and are used by the COCO-panoptic absorption pipeline. The label space is the 133-category COCO panoptic ontology. In order to get activations, for each of 2,000 COCO 2017 val images [Lin et al., 2014] with panoptic masks, we VAE-encode the clean image latent z_{img} and at each of 28 canonical FLUX-dev noise levels σ_k add scheduler-consistent noise $z_k = z_{\text{img}} + \sigma_k \, \varepsilon$ to perform a single FLUX transformer forward pass with an empty prompt, capturing the block-18 residual update (output $-$ input) per patch. SAE activations are then aggregated over (variant, feature, denoising band, COCO-panoptic category) using a 16-pixel patch-overlap with the panoptic mask.

DINOv3 patch embeddings. Each probe image is independently encoded by a DINOv3 ViT-H/16+ backbone at the same 512×512 resolution, producing patch embeddings of shape $(1024, 1280)$ stored at fp16. These embeddings are the substrate for the $K = 512$ DINO-cluster construction and for trajectory geometry in DINO patch-embedding space.

Variants. The variants reported in this appendix are listed in Table 3. Unless explicitly noted in an ablation, every variant shares dictionary width $d = 12{,}288$ and BatchTopK target sparsity $K = 32$.

Reported name	Description
BatchTopKSAE	Pooled BatchTopK baseline. Encoder–decoder with a pre-bias subtraction, a ReLU–linear encoder, BatchTopK selection of the top $B \cdot K$ activations across the batch, and a linear decoder. No trajectory loss.
TA-SAE	BatchTopK SAE augmented with a per-feature persistence logit ρ_f, $\tau_f = \sigma(\rho_f) \in [0, 1]$. The trajectory loss penalizes the per-feature normalized adjacent-step absolute change $\lvert u_{t+1,f} - u_{t,f}\rvert / (\lvert u_{t+1,f}\rvert + \lvert u_{t,f}\rvert + \epsilon)$ of pre-TopK logits, weighted per feature by τ_f, and a $-\log \tau$ prior pulls τ_f toward 1.
MatryoshkaSAE	Matryoshka-style SAE with two nested dictionaries. The persistent slice contains 9,216 features and the transient slice contains 3,072 features. The variant is trained with the matryoshka reconstruction objective only; no temporal persistence prior is added.
JumpReLU	JumpReLU baseline.
per_timestep_bin{0,1,2,3}	Four separate BatchTopK SAEs, each trained from scratch on activations from a single contiguous denoising-bin range. Used as a non-trajectory specialization control.

Table 3: Variants reported in this appendix.

Compute requirements. All experiments were run on a single 4×NVIDIA A100 node. Approximate wall-clock budgets per stage of the main pipeline: image generation (∼12 h), SAE training across the three main variants (∼12 h), evaluation over the 2,000-image COCO panoptic probe set (∼6 h), the SAE-code linear probe sweep (∼1 h), and DINOv3 cluster labeling (∼1 h). Ablation runs reuse the trained variants and add only the eval cost.

A.2 SAE training and reconstruction

Training data. SAEs are trained on FLUX.1-dev block-18 activations generated from 100,000 LAION-aesthetic captions. Each caption seeds one 28-step FlowMatchEulerDiscrete trajectory at 512×512 resolution. Per (image, step) pair, 102 image-token positions are kept by an activation-mass filter, giving approximately 285M (image, step, patch) training rows of dimensionality 3072 at bfloat16. The trajectory pairing required by TA-SAE consumes consecutive timesteps $(t, t+1)$ from the same image. The held-out evaluation set used for reconstruction metrics consists of 10,000 disjoint LAION-aesthetic captions, processed through the same pipeline.

Loss functions. All variants share a normalized mean-squared reconstruction loss $\mathcal{L}_{\text{recon}} = \|x - \hat{x}\|^2 / \|x\|^2$ averaged per batch. BatchTopKSAE adds an OpenAI-style auxiliary reconstruction term $\mathcal{L}_{\text{aux}}$ with coefficient $\alpha_{\text{aux}} = 1/32$, which reconstructs the residual using the top 256 inactive pre-TopK features. MatryoshkaSAE adds a second normalized MSE term that reconstructs x from only the persistent slice of z (the first 9,216 features), scaled by an $\alpha_{\text{matryoshka}}$ coefficient. TA-SAE

additionally adds two terms: a persistence prior $\mathcal{L}_{\text{prior}} = -\lambda_{\text{prior}} \overline{\log(\tau + \varepsilon)}$ with $\lambda_{\text{prior}} = 0.01$, and a trajectory-consistency term that, for paired pre-TopK logits $u_{t,f}$ and $u_{t+1,f}$ at steps t and $t{+}1$, penalizes $\lambda_{\text{traj}}\, \mathbb{E}_{i,p,t,f}\left[\tau_f \cdot |u_{t+1,f} - u_{t,f}| / (|u_{t+1,f}| + |u_{t,f}| + \epsilon)\right]$, identical to the formula in Section 3. The trajectory loss is computed on *pre-TopK* ReLU logits and not on the BatchTopK-sparsified codes.

Optimization. All variants are trained for 10,000 optimizer steps with a peak learning rate of 1×10^{-3} at single-epoch passes through the chunk dataset. Decoder columns are unit-norm projected after each step. The default operating point reports $\lambda_{\text{traj}} = 0.5$ for TA-SAE.

Reconstruction metrics. Reconstruction quality is reported by three quantities. The variance-normalized R^2 is computed as $1 - \|x - \hat{x}\|^2 / \sum \|x\|^2$ aggregated over the held-out evaluation set. The reconstruction cosine $\overline{\cos(x, \hat{x})}$ is computed per row in the per-step normalized space and averaged across rows. The fraction-alive statistic is the fraction of the $d = 12{,}288$ dictionary features whose batchwise fire count over the held-out set is strictly positive. Per-band reconstruction cosine breakdowns use the seven bands of Table 2.

Variant	R^2	**Mean cos**$(x, \hat{x})$	**Std cos**	**Fraction alive**
BatchTopKSAE	0.944	0.836	0.045	0.489
MatryoshkaSAE	0.946	0.831	0.044	0.637
TA-SAE	0.949	0.837	0.041	0.734

Table 4: Held-out reconstruction summary for the three main variants.

Variant	**Band 1**	**Band 2**	**Band 3**	**Band 4**	**Band 5**	**Band 6**	**Band 7**
BatchTopKSAE	0.816	0.826	0.836	0.840	0.841	0.835	0.869
MatryoshkaSAE	0.813	0.823	0.832	0.836	0.837	0.833	0.865
TA-SAE	0.824	0.832	0.838	0.841	0.841	0.836	0.866

Table 5: Per-band mean reconstruction cosine. Band index follows Table 2.

Sparsity ablation. The K ablation varies BatchTopK's target sparsity at fixed dictionary width $d = 12{,}288$ and reports held-out reconstruction R^2. For TA-SAE, λ_{traj} is held at the default 0.5.

Variant	$K{=}16$	$K{=}32$	$K{=}64$	$K{=}128$
BatchTopKSAE	0.913	0.944	0.975	0.985
MatryoshkaSAE	0.914	0.946	0.976	0.987
TA-SAE	0.914	0.949	0.975	0.984

Table 6: Held-out R^2 as a function of BatchTopK target sparsity K.

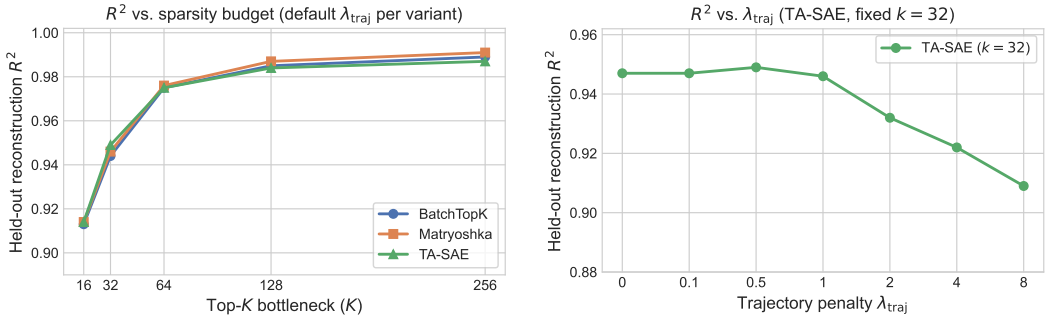
Trajectory-loss strength ablation. The λ_{traj} ablation varies the trajectory-consistency coefficient for TA-SAE at fixed $K = 32$ and reports held-out reconstruction R^2. The $\lambda_{\text{traj}} = 0.5$ row is the default operating point used elsewhere in the paper.

A.3 COCO panoptic per-band object alignment

Inputs. For each probe image, each patch is annotated with a soft COCO panoptic mass vector by overlaying the panoptic mask onto the patch grid. The soft mass for category c in patch p equals the fractional area of p covered by category c. A separate void share is recorded for the share covered by the unannotated background, and a multi-object share is recorded for patches whose mask contains more than one labeled category. Every probe image is evaluated under the FLUX schedule at all 28 timesteps, with the SAE encoder applied at each step.

λ_{traj}	0.0	0.1	0.5	1.0	2.0	4.0	8.0
R^2	0.947	0.947	0.949	0.946	0.932	0.922	0.909

Table 7: Held-out R^2 for TA-SAE as a function of λ_{traj} at $K = 32$.



(a) R^2 vs. sparsity level K at the default λ_{traj} for each variant. All three variants track the same reconstruction–sparsity frontier.

(b) R^2 vs. λ_{traj} for TA-SAE at fixed K=32. Reconstruction is essentially flat for $\lambda_{\text{traj}} \leq 1$ and degrades only for the largest penalties.

Figure 5: Training ablations for the trajectory-aware objective. Both panels report held-out reconstruction R^2 on block-18 FLUX activations. Numbers also reproduced in Tables 6 and 7.

Per-band aggregation. For each (variant, feature, band) triple, activation-weighted soft category counts are accumulated over all probe images, all eligible patches, and all timesteps belonging to the band. Specifically, the weighted count for (f, b, c) is the sum over (image, step in band b, patch) of the post-ReLU activation z_f multiplied by the soft mass of category c at that patch. The total inside mass is the sum over c of the resulting weighted count, the dominant category $c^\star(f, b) = \arg\max_c$ is the COCO category with the largest weighted count, and the label-mass purity is the share of inside mass assigned to $c^\star(f, b)$. Void share, multi-object share, and labeled-patch share are computed analogously.

A.4 Feature-object absorption

Method. Feature-object absorption asks whether the dominant COCO panoptic label of a fixed feature changes across denoising bands. The concept basis is the 133-category COCO panoptic label space and the temporal axis is the 7-band partition of Table 2. For each (variant, feature, band) cell, the dominant COCO object label is taken from Appendix A.3.

We determine a band as *valid* for a feature when its fires on at least 30% of patches labeled as the concept. A feature is *eligible* for the absorption rate denominator when it has at least two valid bands. For each eligible feature, the object absorption index is the number of distinct dominant COCO object labels observed across its valid bands. A feature is counted as *absorbed* when this index is at least two. The absorption rate is the fraction of eligible features that are absorbed. The soft absorption statistic for a feature is the mean pairwise Jensen–Shannon distance between the COCO object distributions $P_{\mathcal{Y}}(\cdot \mid f, b)$ across the feature's valid band pairs, where the JS distance is $\sqrt{\text{JSD}}$ in nats.

Variant	Absorption rate
TA-SAE	0.436
BatchTopKSAE	0.558
MatryoshkaSAE	0.600
JumpReLU	0.870

Table 8: TEST-split feature-object absorption summary.

Variant	Mean soft JS	Mean absorption index
TA-SAE	0.448	1.691
MatryoshkaSAE	0.480	1.847
BatchTopKSAE	0.482	1.784
JumpReLU	0.645	2.665

Table 9: Additional TEST-split feature-object absorption summary fields.

A.5 DINO-cluster temporal absorption

Cluster construction. The DINO cluster basis is fitted by sampling $1{,}000{,}000$ rows from a 286.7M-row DINOv3 ViT-H/16+ patch-embedding cache built from $10{,}000$ held-out images at all 28 denoising timesteps with 1024 patches per (image, step). MiniBatchKMeans with $K = 512$, batch size $10{,}240$, $n_{\text{init}} = 3$, $\max_{\text{iter}} = 200$ and seed 0 is fit on the float-32 sample. The resulting centroid matrix has shape $(512, 1280)$ and is the canonical DINO concept basis used here.

Per-feature, per-timestep aggregation. For every probe image, every (variant, step, patch) activation is paired with the index of the nearest centroid (cosine top-1 against L2-normalized centroids). For each (variant, feature, step), the activation mass is aggregated over patches and assigned to the corresponding cluster id, yielding a discrete distribution $p(c \mid f, t)$ over the $K = 512$ clusters at each timestep. The dominant concept at (f, t) is the cluster receiving the largest mass, the purity at (f, t) is its mass share, and the entropy at (f, t) is computed on the same distribution. Absorption metrics are reported over the full dictionary of alive features.

Per-feature absorption flag. For each (variant, feature), the active timestep set is the set of t at which the feature has nonzero mass. Concept switch is the number of distinct dominant clusters across active timesteps. Mean purity is the average dominant-cluster mass share over active timesteps. Time coverage is the fraction of timesteps that are active. Semantic drift is the mean adjacent-step Jensen–Shannon drift in $p(c \mid f, t)$ over active timesteps. The temporal absorption scalar is the product of mean purity, time coverage, and semantic drift. A feature is counted as *absorbed* when concept switch is at least two, mean purity is at least 0.40, and time coverage is at least 0.20. The absorbed fraction reported is the fraction of alive features in the dictionary satisfying this conjunction.

Temporal activation index. The temporal activation index of a feature is the variance across timesteps of its mean image-level maximum activation, divided by the total variance of the corresponding feature activation matrix. The bimodality scalar reported is the BIC difference between one-component and two-component Gaussian mixture fits to the feature population's temporal activation index distribution.

Variant	Percent absorbed	Mean switch P	TAI BIC
TA-SAE	11.99%	12.42	18,147
MatryoshkaSAE	12.24%	11.36	8,611
BatchTopKSAE	17.99%	9.56	8,498

Table 10: DINO-cluster temporal absorption. Values are computed on the persistence-weighted feature subset used for the original v3.5 metric; the JumpReLU population is degenerate.

A.6 COCO-relabeled DINO-cluster object absorption

Cluster→object assignment. Each of the $K = 512$ DINO clusters is annotated with a hard top-1 COCO panoptic object label. For every probe image and every patch, the patch's DINO cluster id (cosine top-1 against the centroids) and its panoptic majority label are recorded. For each cluster, the dominant panoptic category is taken to be the most frequent panoptic label over all observed patches in that cluster. Of the 512 DINO clusters, 493 have at least one labeled patch and therefore receive a usable COCO panoptic top-1 mapping; the remaining 19 are excluded from the relabeled basis.

Per-feature absorption. For each (variant, feature, branch, timestep), the activation mass at every active patch is reassigned from its cluster id to the cluster's COCO panoptic top-1 label. This yields a per-(feature, timestep) distribution over COCO panoptic categories. Multiple clusters mapping to the same category have their masses summed. Dominant object, mean purity, time coverage, and the absorbed flag are computed using the same purity and coverage thresholds as Appendix A.5 ($\tau_{\text{purity}} = 0.40$, $\tau_{\text{cov}} = 0.20$, concept switch ≥ 2). The relabeled per-feature table on the principal snapshot contains 43,085 rows.

Variant	n features	Absorbed fraction
TA-SAE	12,192	**23.30%**
MatryoshkaSAE	7,823	32.41%
BatchTopKSAE	8,080	35.57%
JumpReLU	297	24.25%†

Table 11: COCO-relabeled DINO-cluster object absorption over the full dictionary. **Bold** = best. †JumpReLU's alive-feature population is dominated by an irregular split (1 vs. 296 features) and the resulting fraction is degenerate.

A.7 Temporal feature splitting

Concept bases. Splitting is computed in three concept bases: the COCO panoptic ontology (Appendix A.3), the raw $K = 512$ DINO clusters (Appendix A.5), and the COCO-relabeled DINO clusters (Appendix A.6, with the additional cluster filter that the cluster's top-1 mass share is at least 0.40).

Concept-conditioned feature distribution. Splitting fixes a concept and asks how its dominant SAE feature support changes across denoising bands. For each (variant, concept c, band b) cell, the activation-weighted patch contribution is aggregated across probe images, eligible patches, and timesteps in the band, and normalized over alive features to give $p(f \mid c, b)$. A cell is included when its activation-active patch count is at least 5. For the DINO-cluster basis, concepts are further required to have at least 50 activation-active patches summed across bands.

Per-cell statistics. For each (variant, concept, band) cell, n_{80} is the smallest k such that the top-k features cover at least 80% of activation mass, the entropy is the Shannon entropy of $p(f \mid c, b)$ in nats, and the effective feature count is its exponential.

Per-concept temporal statistics. For each (variant, concept) pair with at least two band cells, the temporal Jensen–Shannon distance is the mean pairwise $\sqrt{\text{JSD}}$ between $p(f \mid c, \cdot)$ across present band pairs; the conditional mutual information $I(F; B \mid C{=}c)$ is computed from the same joint. Define the per-band argmax feature $f^{\star}(b) = \arg\max_f p(f \mid c, b)$. The top-feature switch rate is the fraction of consecutive present-band pairs at which $f^{\star}(b) \neq f^{\star}(b{+}1)$. The top-feature count is $|\{f^{\star}(b) : b \text{ present}\}|$. Mean band n_{80} is the average of per-band n_{80} across the bands present. Pooled n_{80} is n_{80} of the pooled distribution $\overline{p}(f \mid c) = \text{mean}_b\, p(f \mid c, b)$. Handoff n_{80} is $\max(0, \text{pooled} - \text{mean band})$ and is clamped at zero to guard against the rare step-function inversion that arises for highly concentrated SAEs.

Variant	$\mathbf{n_{80}}$	Entropy	$\mathbf{n_{eff}}$	Active feats.	Temporal JS	$\mathbf{I(F; B \mid C)}$
JumpReLU	60	4.43	84	169	0.013	0.000
MatryoshkaSAE	173	5.55	258	594	0.103	0.024
TA-SAE	223	5.75	313	733	0.103	0.024
BatchTopKSAE	208	5.70	299	597	0.115	0.034

Table 12: Per-variant medians on the COCO panoptic concept basis. MatryoshkaSAE values are taken from the 2026-05-05 backfill ($1{,}064$ per-(concept, band) cells $\times$ 133 panoptic categories).

Variant	n concepts	Temp. JS	Switch rate	Handoff n_{80}	Mean n_{80}	Top-1 count
JumpReLU	424	0.019	0.000	0.000	25.0	1.0
MatryoshkaSAE	424	0.121	0.143	1.750	68.0	2.0
TA-SAE	424	0.120	0.000	2.500	88.6	1.0
BatchTopKSAE	424	0.133	0.143	3.000	82.8	2.0
per_timestep_bin0	289	0.176	0.143	5.625	82.4	2.0

Table 13: Per-variant medians on the raw DINO-cluster basis (TEST split).

Variant	n concepts	Temp. JS	Switch rate	Handoff n_{80}	Mean n_{80}	Top-1 count
JumpReLU	28	0.018	0.000	0.000	25.0	1.0
TA-SAE	28	0.112	0.000	2.000	90.5	1.0
MatryoshkaSAE	28	0.114	0.000	1.562	68.9	1.0
BatchTopKSAE	28	0.125	0.071	2.438	84.9	1.5
per_timestep_bin0	18	0.126	0.143	3.250	87.9	2.0

Table 14: Per-variant medians on the COCO-relabeled DINO-cluster basis (TEST split).

A.8 Trajectory smoothness and feature geometry

Per-band feature centroid. For each (variant, feature, band), the feature's DINOv3 patch-embedding centroid $\gamma_f(b) \in \mathbb{R}^{1280}$ is the activation-weighted mean of DINOv3 patch embeddings over all active patches in the band, taken over the probe images. A band is valid for a feature when at least three timesteps in the band have at least one active patch and the activation-mass denominator is positive. A feature is included in the geometry summary when it has at least three valid bands.

Trajectory geometry statistics. With γ_f defined on the present bands, the L2 arc length is $\sum_b \|\gamma_f(b{+}1) - \gamma_f(b)\|_2$ summed over consecutive valid bands. The cosine arc length sums per-edge cosine distance $(1 - \cos)$ between centroids. Endpoint distance is $\|\gamma_f(b_{\max}) - \gamma_f(b_{\min})\|_2$ over the first and last valid bands. Curvature is the L2 arc length divided by endpoint distance. The mean and max turning angles are computed from successive centroid displacement vectors.

Variant	n features	Arc L2	Arc cos	Endpoint	Curvature	Max turn
TA-SAE	336	0.464	0.021	0.230	1.728	1.592
BatchTopKSAE	1,100	1.282	0.336	0.671	1.919	1.946
MatryoshkaSAE	1,201	1.311	0.408	0.722	1.943	1.961
JumpReLU	260	2.807	1.143	0.832	3.366	2.195

Table 15: TEST-split feature-geometry medians in DINOv3 patch-embedding space.

Per-feature τ and trajectory-smoothness correlation. For TA-SAE, the per-feature persistence parameter is $\tau_k = \sigma(\rho_k)$ with ρ_k the learned raw logit. The τ vector is extracted from the trained checkpoint and joined onto the per-feature geometry table by feature id. The correlation analysis is restricted to features that survive the geometry eligibility filter (≥ 3 valid bands), aggregating across all three splits ($n = 1{,}016$). Spearman and Pearson correlations are reported between τ_k and each geometry quantity, with two-sided p-values.

A.9 Distributed Concept Temporal Stability

Concept signature. For each (variant, concept, band, feature) tuple, the positive mean activation μ_+ is the mean post-ReLU SAE activation over patches in the band whose hard panoptic label equals the concept; μ_- is the analogous mean over patches whose label differs from the concept. The concept signature in band b is $s_{c,b}(f) = \mathrm{relu}\left(\log \frac{\mu_+ + \varepsilon}{\mu_- + \varepsilon}\right) / Z_b$, where Z_b normalizes the positive part across features. The concept set is the top 30 panoptic concepts of the probe set augmented with

Geometry quantity	Pearson r	Pearson p	Spearman ρ	Spearman p
Arc L2	-0.133	1.98×10^{-5}	-0.470	4.99×10^{-57}
Arc cos	$+0.038$	0.227	-0.431	3.42×10^{-47}
Endpoint distance	$+0.035$	0.271	-0.389	4.93×10^{-38}
Curvature	-0.340	6.84×10^{-29}	-0.498	7.43×10^{-65}
Mean turning angle	-0.127	4.98×10^{-5}	-0.301	9.78×10^{-23}
Max turning angle	-0.081	9.84×10^{-3}	-0.230	1.16×10^{-13}

Table 16: Correlations between TA-SAEs per-feature τ_k and per-feature trajectory-geometry quantities. Aggregated across TRAIN, VAL and TEST.

11 forced case-study labels; the per-variant summary contains 30 valid concepts after applying the activation-support filter.

Stability statistics. For each (variant, concept) pair with at least two valid bands, temporal JS is the mean pairwise $\sqrt{\mathrm{JSD}}$ between concept signatures across all band pairs. Adjacent JS uses only consecutive band pairs. Top-M overlap is the average Jaccard overlap of the top M features across consecutive bands for $M \in \{20, 50, 100\}$. Rank correlation is the average Spearman correlation between feature rankings of $s_{c,b}$ and $s_{c,b+1}$. Effective ensemble size is $\exp$ of the entropy of the average concept signature across bands, and n_{80} is the smallest k such that the top-k features of the average signature cover 80% of mass.

Variant	n	Temp JS	Adj JS	Overlap@50	Rank corr.	$\mathbf{n_{eff}}$
JumpReLU	30	0.486	0.412	0.852	0.545	13
TA-SAE	30	0.635	0.573	0.238	0.312	59
BatchTopKSAE	30	0.764	0.728	0.196	0.239	23
MatryoshkaSAE	30	0.769	0.729	0.210	0.265	19

Table 17: Per-variant DCTS summary on the VAL split. n is the number of valid concepts.

Variant	$\mathbf{n_{80}}$
JumpReLU	10
TA-SAE	44
per_timestep_bin3	19
per_timestep_bin2	18
BatchTopKSAE	18
per_timestep_bin0	15
per_timestep_bin1	16
MatryoshkaSAE	14

Table 18: Per-variant DCTS n_{80} summary.

A.10 SAE sparse-code object probe

Probe setup. For each variant, a multinomial logistic regression probe is trained from the SAE sparse code at the patch level to a 30-class COCO panoptic label space. The label space is restricted to the 30 panoptic categories with the largest per-patch support on the probe set; the random macro-F1 baseline is $1/30 \approx 0.033$. For each variant, a 7×7 grid of probes is fit using TRAIN-band activations and evaluated on each of the seven bands. Each probe uses an ℓ_2 regularizer chosen from a logarithmic sweep with the best VAL macro-F1 selecting the operating point. Per-(train band, eval band) cells record macro-F1 and accuracy on the TEST split.

Aggregate statistics. The within-band macro-F1 mean is the mean of the seven diagonal cells of the train/eval grid. The within-band max is the maximum of those diagonal cells. The cross-band

macro-F1 mean is the mean of the 56 off-diagonal cells. The cross-band drop is the within-band mean minus the cross-band mean. Within- and cross-band accuracy means are computed analogously.

Variant	Within-band mean	Within-band max	Cross-band mean	Cross-band drop
TA-SAE	0.0983	0.143	0.0784	0.020
BatchTopKSAE	0.0937	0.132	0.0754	0.018
MatryoshkaSAE	0.0901	0.129	0.0718	0.018
JumpReLU	0.0695	0.100	0.0603	0.009

Table 19: Per-variant macro-F1 on the TEST split, aggregated across the 7×7 band grid. Within-band is the mean of the seven diagonal cells (train-band equals eval-band); cross-band is the mean of the 42 off-diagonal cells; cross-band drop is within-band mean minus cross-band mean.

Variant / baseline	Peak within-band macro-F1	Mean within-band macro-F1
TA-SAE	0.143	0.098
BatchTopKSAE	0.132	0.094
MatryoshkaSAE	0.128	0.090
JumpReLU	0.100	0.069
random baseline	–	0.033

Table 20: Probe peak and mean macro-F1 for the three main variants and the JumpReLU baseline.

f

A.11 Broader societal impact

This work is foundational interpretability research on text-to-image diffusion models: we train sparse autoencoders on FLUX.1-dev residual updates and study whether their features carry stable visual semantics across denoising time. We do not release new generative models, do not train on or release any user-collected data, and do not propose new generation capabilities. The released artifacts are SAE weights and analysis scripts that operate on a publicly available pretrained model. We discuss the most plausible positive and negative societal effects below. In terms of positive impacts, there's a lot - this incentivizes mechanistic understanding of a model, allowing for better control and better safety mitigations, and prevention of misuse. In terms of negative impacts, better control of diffusion models means an adversary with internal access can generate harmful images easier, or remove watermarks easier. Overall, we believe the marginal increase in interpretability of an already-public T2I model is net positive, especially for safety auditors and researchers working on bias, memorization, and provenance.

A.12 Existing assets and licenses

This work uses only publicly released models, datasets, and software. We did not collect, scrape, or release any new datasets. The pretrained generative model (FLUX.1-dev) and the DINOv3 vision encoder are used unmodified for activation extraction and patch labeling; no derivative model weights are redistributed. Table 21 lists each external asset together with its source citation, version, and license. URLs point to the official release pages.

We use COCO 2017 panoptic exclusively for evaluation: the validation split's panoptic masks are projected onto FLUX patches to derive the per-patch object labels described in Appendix A.3. We do not redistribute COCO images. LAION-COCO-Aesthetic captions are used as text prompts for FLUX; we never download or train on the underlying LAION images. FLUX.1-dev's non-commercial license is respected by limiting our analysis to research use; we release SAE weights and analysis code, not the FLUX weights or any derivative finetune. DINOv3 is used as a frozen feature extractor for cluster labeling, consistent with its research-use license.

Asset	Citation	Version / URL	License
Models			
FLUX.1-dev (text-to-image diffusion transformer)	Black Forest Labs [2024]	`black-forest-labs/FLUX.1-dev` on Hugging Face	FLUX.1 [dev] Non-Commercial License
DINOv3 (ViT-L/16, vision encoder)	Simeoni et al. [2025]	`facebookresearch/dinov3`	DINOv3 License (research use); code Apache-2.0
Datasets			
COCO 2017 panoptic	Lin et al. [2014]	`val2017` panoptic split, `cocodataset.org`	Annotations: CC BY 4.0; images: Flickr terms (per-image)
LAION-COCO-Aesthetic captions	Schuhmann et al. [2022]	Public Hugging Face mirror	CC BY 4.0 (metadata; underlying images not redistributed by us)
Software			
PyTorch	—	v2.4	BSD 3-Clause
scikit-learn (linear probes)	—	v1.5	BSD 3-Clause
NumPy / pandas / pyarrow	—	current PyPI	BSD 3-Clause
matplotlib (figures)	—	v3.9	matplotlib license (PSF-compatible)

Table 21: External assets used in this paper, with citations, versions, and licenses. All assets were used in accordance with their respective terms of use; no asset license was modified or relicensed.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: All of our claims say exactly what our data says - we reduce splitting and absorption, we only run on block18 of Flux, and they stay competitive with baseline SAEs.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

Justification: We thoroughly discuss the limitations in our limitations section.

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: We don't have a theoretical result.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We provide all experiment details 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: [No]

Justification: We plan to release code but have not been able to get it done yet, as we have only recently finished the paper. We use majority data from public datasets, except for the images we generate.

Guidelines:

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

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide all necessary details in the appendix to the point where a reader can understand the entirety of what we did.

   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: We provide confidence intervals for all core results.

   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: We detail our compute usage in the appendix for all experiments that required a decent amount of compute.

   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: I have read through the code and we follow it fully.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: We have a section in the appendix for this.

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: We don't release data or models.

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: Yes, we only use public datasets and cite everything.

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A]

    Justification: No new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: Not crowdsourced or research with human subjects.

Guidelines:

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

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

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

Answer: [N/A]

Justification: No usage of crowdsourcing or research with human subjects.

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: The core method development in this research does not involve llms in any non-standard way.

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.