BEYOND THE LINEAR REPRESENTATION HYPOTHESIS: NON-LINEAR ACTIVATION STEERING IN TEXT-TO-IMAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Mechanistic interpretability often relies on the Linear Representation Hypothesis (LRH), which assumes that high-level concepts are encoded as linear directions in activation space. Yet a natural visual concept does not necessarily require a linear visual transition: between sunny and stormy lies an intermediate weather state such as a sky with a few white clouds, not simply a weaker storm; between a caterpillar and a butterfly, the progression is not a caterpillar with continuously growing wings. This raises the question of whether such true intermediate states are also represented nonlinearly by the model. Indeed, when we prompt text-to-image models directly for intermediate attributes, their activations rarely fall along the straight direction connecting the endpoints. Therefore, we propose KANSteer, which models concept traversal as a curve passing through its intermediate states. Seeking a representation that is both simple and interpretable, we propose to use Kolmogorov-Arnold Networks (KANs), which provide a one-dimensional coordinate whose learned functions define the trajectory. This allows the steering direction to vary along the concept while preserving an interpretable representation. Across several concepts and text-to-image diffusion transformers, we find that their activation trajectories substantially deviate from straight lines, and that KANSteer provide a closer fit and smoother traversal of intermediate attributes than linear steering.

1 INTRODUCTION

In deep learning, many semantic concepts admit an ordered progression, yet their representations need not change in a uniform, linear fashion. For example, the transition between seasons, the metamorphosis of an insect, or the progression of human aging is not simply a matter of scaling a single visual feature. Instead, these concepts evolve through meaningful intermediate states, where different attributes emerge, change, and disappear at different stages of the progression. Despite the complexity of these transitions, mechanistic interpretability has predominantly relied on the Linear Representation Hypothesis (LRH) to understand how models encode them. LRH posits that high-level concepts are represented by straight directions within a model's activation space. In text-to-image diffusion models, this perspective has motivated methods for precise control over generated content (Surkov et al., 2024), model debiasing (Modell, 2026) or unlearning (Gaintseva et al., 2025; Cywiński & Deja, 2025). However, recent research in Large Language Models (LLMs) has highlighted a critical limitation of the LRH: while concepts may be linearly identifiable, they are not always linearly traversable. When navigating between related states–such as moving from Thursday to Friday or adjusting a character count-the underlying geometric path forms a curved manifold rather than a straight line (Raval et al., 2026; Wurgaft et al., 2026).

This distinction becomes particularly important in visual domain. For a discrete structure such as days of the week, the goal is ultimately to reach one of a finite number of states; an imperfect trajectory can still place the representation close to the intended state. Visual concepts that are inherently continuous, offer no such simplification. Between two endpoints lie a sequence of meaningful intermediate states, and these states are themselves part of the concept. If they do not lie on a straight path, linear steering can therefore misrepresent the concept throughout the traversal, even when its endpoints are correctly identified. We directly test this by examining the activation tra-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

jectories induced by ordered visual attributes in text-to-image diffusion models. For each concept, we prompt the model with a sequence of progressively varying attribute levels and measure how their corresponding representations evolve in activation space. Across a range of visual concepts, these intermediate representations consistently deviate from straight line connecting the endpoints, revealing that visual concepts are not always linearly traversable.

This suggests that a concept can still have a simple 1D representation without following a fixed linear path for its intermediate states. We therefore model concept traversal as a one-dimensional curve through activation space. Using a Kolmogorov-Arnold Network (KAN) (Liu et al., 2024), we parameterize this curve directly from the observed intermediate states, allowing steering direction to vary along trajectory while retaining a single interpretable coordinate. In this way, the representation preserves the simplicity of linear steering without imposing a straight path on a concept whose geometry may be inherently curved. We evaluate this approach across a range of ordered visual concepts and diffusion-model configurations. For each concept, we compare nonlinear steering against a linear trajectory under the same set of observed attribute states. While both approaches recover the endpoints, their behavior between them differs substantially: the nonlinear trajectory more closely follows the intermediate states in activation space and produces images that better match both the corresponding text and their prompt-generated visual references. These improvements are reflected consistently across directional and image-based similarity measures, indicating that the learned trajectory captures the progression between attributes rather than merely connecting its endpoints. We further test whether the learned trajectories can extend beyond the range of attributes used to fit them. By withholding later stages of each ordered concept during fitting, we evaluate whether the resulting trajectories can recover them. The nonlinear trajectories provide substantially more faithful extrapolations, both qualitatively and quantitatively, whereas linear steering remains constrained to a fixed direction and increasingly departs from the underlying progression.

Our contributions can be summarized as follows:

- We demonstrate that visual concepts in T2I diffusion models are not linearly traversable: intermediate attributes deviate from straight directions defined by their endpoints.
- We introduce KANSteer, a 1D-nonlinear steering framework that replaces the fixed steering direction with a learned activation-space trajectory, preserving an interpretable scalar concept coordinate while allowing the steering direction to vary along the trajectory.
- Given the same supervision, KANSteer improves over the linear direction on every metric and all three model configurations, raising directional CLIP from 0.108 to 0.170 and lowering FID from 47.9 to 28.6 on FLUX.1-schnell (1-step). Notably, it transfers unchanged from the FLUX.1-schnell 1-step activations it is fitted on to FLUX.1-schnell 4-step and FLUX.1-dev generation. Furthermore, we show that KANSteer can extrapolate beyond its fitted range while generating realistic samples.

2 Related Work

Linear representation hypothesis (LRH). The LRH proposes that a high-level concept is represented by a direction in activation space. This idea has been formalized for language models by Park et al. (2023; 2024); Jiang et al. (2024), and used to control properties such as truthfulness (Li et al., 2023), sentiment (Tigges et al., 2023), and refusal (Arditi et al., 2024) by adding a fixed steering vector. An analogous picture has emerged in vision, where diffusion models exhibit surprisingly linear internal representations. Simple linear probes can extract scene properties–such as depth and figure-ground structure–from intermediate activations well before denoising is complete (Chen et al., 2023). Similarly, linear projections of attention layers in diffusion transformers yield sharp concept saliency maps (Helbling et al., 2025). Recent work highlights that specific knowledge localizes to just a few blocks (Basu et al., 2024a;b; Zarei et al., 2025), and sparse autoencoders can decompose these activations into interpretable, causal features (Surkov et al., 2024; Cywiński & Deja, 2025).

These structural discoveries naturally motivated direct interventions. Early methods identified the U-Net bottleneck as a semantic latent space that is largely homogeneous and linear, where scaling a single vector gradually alters image attributes (Kwon et al., 2023). Subsequent approaches extracted these directions unsupervised (Dalva & Yanardag, 2023) or used them to guide attribute distributions (Parihar et al., 2024). A second line of research directly ported activation steering from

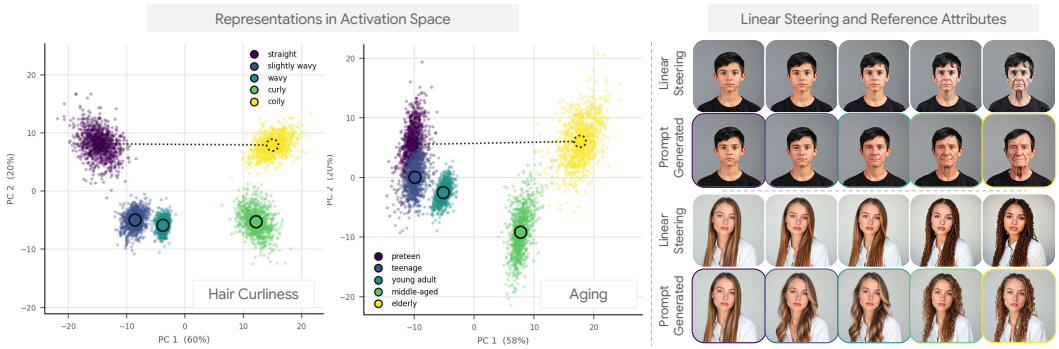

Figure 1: Many concepts do not vary linearly in activation space. **Left:** PCA projection of block-q delta activations for three concepts (see more in Appendix 7); each point is one seed, with attribute centroids, and the two endpoints outlined. The intermediate attributes bend away from straight line between the endpoints, so a fixed direction cannot pass through these attributes. **Right:** Sweeping closed-form linear direction (*Linear Steering*) holds the source attribute and then jumps to the target, while prompt-generated attributes it is fitted on (*Prompt Generated*) traverse the concept smoothly.

language models to image generators. In these methods, steering directions are computed via optimal transport (Rodriguez et al., 2025a;b) or by isolating the mean difference between prompts with and without a specific concept (Gaintseva et al., 2025). Other approaches apply this linear logic elsewhere, calculating directions in the text embedding space (Kim & Ghadiyaram, 2025; Baumann et al., 2024) or applying continuous attribute control via learned low-rank updates (Gandikota et al., 2024; 2025). Yet, despite the variety of these activation-space methods, underlying intervention mechanism remains identical: they all steer by updating $x' = x + \alpha v$ for a fixed vector v.

Extending linear representation hypothesis In LLMs, several works argue that the linear manifold does not hold for all concepts. The clearest example is the representation of weekdays, where each day is identifiable as a separate direction but the seven form a circle rather than a line, so moving from Monday to Tuesday requires a different displacement than moving from Thursday to Friday (Engels et al., 2024). Character counts likewise lie on curved low-dimensional manifolds (Gurnee et al., 2026). A concept can therefore be *linearly identifiable* without being *linearly traversable*. This has motivated steering methods that replace a fixed direction with a trajectory, learning a concept manifold with a supervised probe (Modell, 2026), obtaining curved trajectories with polynomial kernel PCA (Raval et al., 2026), or fitting cubic splines through concept centroids, which keeps steering on trajectories already present in the model where a linear direction passes through off-manifold regions and produces unnatural outputs (Wurgaft et al., 2026). Engels et al. (2024) note that such representations are "mostly discontinuous", as the model does not naturally represent states between adjacent weekdays or months. Visual attributes differ, since meaningful intermediates are expected even when only a few are specified, and in T2I diffusion models they can be generated and observed directly. Unlike prior approaches that recover trajectories from activation geometry or concept centroids, we fit a parametric trajectory with a KAN to the intermediate states themselves, which gives an explicit scalar coordinate and a channel-wise decomposition, and makes the trajectory define the steering rule rather than being fitted separately from the intervention.

3 Concept Trajectories are not Linear in Activation Space

Before moving to the proposed method, we first ask a simple question: if we learn a linear steering direction between two endpoints, can this direction reproduce the intermediate states of the concept? A common approach to activation steering in T2I models is to define a concept using two attributes and learn a direction between them (Gaintseva et al., 2025; Rodriguez et al., 2025a). Under the linear representation hypothesis (LRH), moving along this direction should then traverse the concept from one endpoint to the other, passing through the corresponding intermediate states.

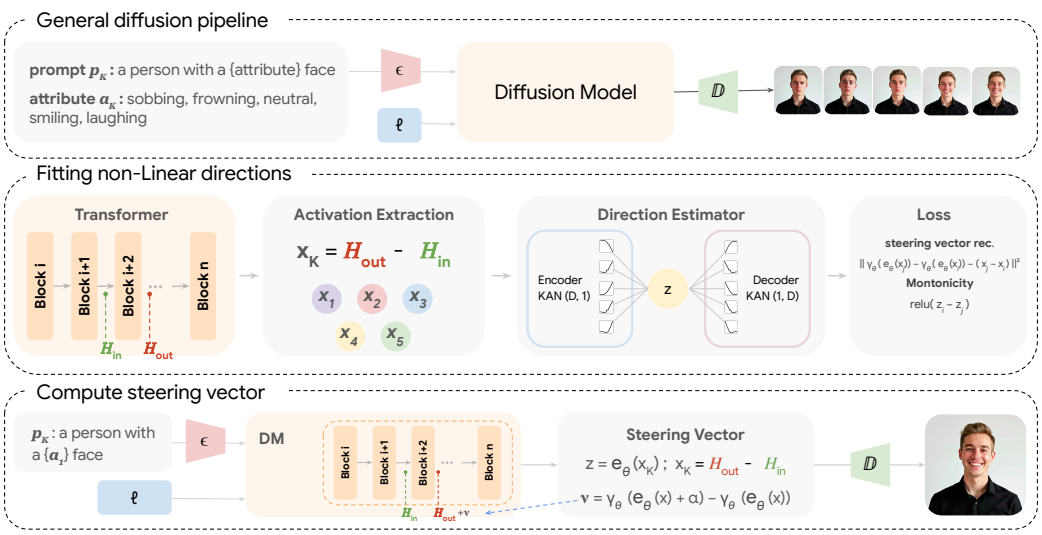

Figure 2: Overview of KANsteer. **Top:** Each concept is instantiated at $K = 5$ ordered attributes $(a_1, .., a_k)$, and generated all from same initial noise. **Middle:** At block i, we compute residual activation from the image stream. Every attribute serves as a source x_i and later one as its target x_j, giving source–target combinations, each a training example with source $X = x_i$ and difference $Y_{ij} = x_j - x_i$. A 1D autoencoder maps x_i to a coordinate $z_k = e_\theta(x_i)$ and recovers the direction as a difference of decoded positions, fitted with $\mathcal{L}_{\text{rec}}$ and $\lambda \sum_{i<j} \max(0, z_i - z_j)$ to ensure reconstruction and monotonicity of the attributes. **Bottom:** At inference, the coordinate is and shifted by α, so the steering vector computed $\nu_\alpha(x) = \gamma_\theta\left(e_\theta(x) + \alpha\right) - \gamma_\theta\left(e_\theta(x)\right)$.

We test this directly by constructing intermediate states independent of steering direction. For each concept, we specify five ordered attributes $a_1 < a_2 < \cdots < a_K$, with first and last serving as the steering endpoints and the remaining three defining intermediate targets. For example, for hair curliness we use *straight*, *slightly wavy*, *wavy*, *curly*, and *coily*. For aging, we use *preteen*, *teenage*, *young adult*, *middle-aged*, and *elderly*. We then generate images for all five attributes using same prompt template, and same seed, changing only attribute phrase (See Section 5.1 for more details). These prompt-generated representations give us an explicit trajectory of intermediate states against which to evaluate linear steering. For each concept, we first learn the linear steering direction using only the two endpoint attributes. We then sweep the steering coefficient between these endpoints and compare the resulting representations to those obtained from the three intermediate prompts.

To assess whether the ordered attributes of a concept lie on a straight path, here we analyze the geometry of the attributes in the activation space. For each concept we take the delta activation of every {attribute, seed} pair, and since all K attributes of a seed are generated from the same initial noise, we subtract the per-seed mean across attributes. This removes the shared initial noise and leaves the variation which the attributes induce. We then project the centered activations onto their leading principal components. If the attributes are linearly encoded, the K attribute clusters would fall on a single line in this projection, and their ordering would run straight along it. However, the results in Figure 1 reveal that across the concepts, the intermediate attributes sit off the straight path between the two endpoint attributes. A quantitative analysis confirms that intermediate attributes deviate substantially from the straight path between the endpoints (Appendix A.1).

4 Method

In Section 3, we establish that the attributes of many concepts do not lie on a line in activation space, so a single direction cannot carry a concept through its intermediate states. This motivates replacing fixed steering direction used by linear activation steering with a 1D trajectory. To do so, we fit a model of how activations change across all K attributes of a concept and derive the steering rule from that model. Steering then amounts to moving the activation along this learned trajectory.

4.1 A ONE-DIMENSIONAL TRAJECTORY MODEL

We model the activations induced by the ordered attributes as points along a shared *concept trajectory* in activation space. For each seed, we generate one image for each attribute $a_1, \ldots, a_K$ from the same initial noise, changing only the attribute phrase. Because the resulting images share their initial noise, their activations retain much of the same image-specific content, such as layout, identity, and background. We therefore view the activation under attribute a_k as

$$x_k \approx b + \gamma(t_k), \tag{1}$$

where $b \in \mathbb{R}^D$ is the content shared across the attributes of a given seed and patch, and $\gamma : \mathbb{R} \to \mathbb{R}^D$ is the *concept trajectory*. The trajectory describes how the representation changes as the attribute progresses, while the scalar t_k specifies the position of attribute a_k along that trajectory.

A displacement from x_i to x_j can be expressed as $\gamma(t_j) - \gamma(t_i)$. Thus, steering requires two components: a model of the trajectory γ, and a way to assign an activation its coordinate on that trajectory.

4.2 FITTING THE TRAJECTORY

Given this formulation, we jointly learn trajectory and the coordinate assigned to each activation. Let $e_\theta : \mathbb{R}^D \to \mathbb{R}$ map an activation to its coordinate on the trajectory, and let $\gamma_\theta : \mathbb{R} \to \mathbb{R}^D$ map a coordinate to the corresponding point in activation space. For every seed and patch, each ordered pair of attributes provides a transition that the trajectory must explain. For a pair (i, j) with $i < j$, the observed transition is $Y_{ij} = x_j - x_i$. The model predicts the corresponding displacement as,

$$\gamma_\theta\left(e_\theta(x_j)\right) - \gamma_\theta\left(e_\theta(x_i)\right). \tag{2}$$

We therefore minimize,

$$\mathcal{L}_{\text{rec}} = \sum_{i<j} \left|\gamma_\theta\left(e_\theta(x_j)\right) - \gamma_\theta\left(e_\theta(x_i)\right) - (x_j - x_i)\right|_2^2, \tag{3}$$

over all source–target pairs. Because the reconstruction loss depends only on differences along the trajectory, it does not by itself enforce that the learned coordinates follow the semantic ordering of the attributes. We therefore add a monotonicity penalty for the violations of the desired ordering.

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda \sum_{i<j} \max(0, z_i - z_j), \qquad z_k = e_\theta(x_k), \tag{4}$$

4.3 STEERING ALONG THE LEARNED TRAJECTORY

Once the encoder and trajectory have been learned, steering requires no separate vector estimation. Given an activation x, its current position on the trajectory is $z = e_\theta(x)$. A steering strength α specifies a displacement in this one-dimensional coordinate. We therefore move from z to $z + \alpha$ and apply the corresponding displacement in activation space:

$$\nu_\alpha(x) = \gamma_\theta\left(e_\theta(x) + \alpha\right) - \gamma_\theta\left(e_\theta(x)\right), \qquad x' = x + \nu_\alpha(x). \tag{5}$$

The steering vector $\nu_\alpha(x)$ depends on current position $e_\theta(x)$. Thus, same coordinate shift can produce different activation-space displacements at different points along the concept. This allows the intervention to follow a curved trajectory while retaining a single scalar notion of concept strength.

Linear steering is recovered as a special case. If the trajectory is affine, $\gamma_\theta(z) = c + zw$, then

$$\nu_\alpha(x) = [c + (z + \alpha)w] - [c + zw] = \alpha w, \tag{6}$$

which is independent of x. Thus, the proposed formulation generalizes fixed-vector steering: a straight trajectory recovers the standard linear intervention, while a curved trajectory produces position-dependent steering vectors.

4.4 KAN PARAMETERIZATION

The trajectory and encoder can be parameterized by many nonlinear one-dimensional models. We use Kolmogorov–Arnold Networks (KANs) (Liu et al., 2024) because they preserve an explicit scalar coordinate while allowing the trajectory to be nonlinear.

The encoder $e_\theta : \mathbb{R}^D \to \mathbb{R}$ is implemented as a one-dimensional KAN that maps the activation to the scalar coordinate z. The decoder $\gamma_\theta : \mathbb{R} \to \mathbb{R}^D$ is implemented as a KAN layer of width $[1, D]$. In this form, each output coordinate is a learnable univariate function of the same scalar input:

$$\gamma_\theta(z) = [\phi_1(z), \ldots, \phi_D(z)], \tag{7}$$

with each function parameterized as

$$\phi_d(z) = w_{b,d}, \mathrm{SiLU}(z) + w_{s,d}, \mathrm{Spline}_d(z). \tag{8}$$

The spline term provides flexibility, while SiLU provides a smooth base function. This allows to deviate from linearity and represent curved activation-space trajectories observed in Section 3.

5 Experiments

5.1 Experiment Setup

Models. We evaluate KANSteer on three distinct FLUX.1 configurations. FLUX.1 employs a transformer-based architecture comprising 19 dual-stream and 38 single-stream blocks, conditioned on T5-XXL and CLIP text encoders. We contrast FLUX.1-schnell at both 1 and 4 sampling steps against FLUX.1-dev at its native high-step regime. This design isolates the variable of denoising-step budget while varying the distillation objective between the schnell and dev variants.

Delta activations at a residual block. We analyze a text-to-image diffusion transformer where the backbone is a stack of residual blocks operating on a stream of P image patches of width D. Let $H^\ell \in \mathbb{R}^{P\times D}$ denote the image stream entering block ℓ, and $\mathcal{B}_\ell$ the transformation it applies under text conditioning c. Each block writes an additive update to the stream,

$$H^{\ell+1} \;=\; H^\ell \;+\; \mathcal{B}_\ell\big(H^\ell, c\big)\,, \tag{9}$$

and our unit of analysis is that update at a single patch p,

$$x_p^\ell \;:=\; \mathcal{B}_\ell\big(H^\ell, c\big)_p \;=\; H_p^{\ell+1} - H_p^\ell \;\in\; \mathbb{R}^D, \tag{10}$$

which isolates what the block contributes from what the residual stream already carried. We refer to x_p^ℓ as a *delta activation* and drop the block index hereafter. All analysis is performed at a single block q, selected as the most consequential for the attributes we study; the selection procedure, based on activation norms and activation patching, is detailed in Appendix A.2.

Implementation Details. We generate at 1024^2, giving $P = 4096$ image patches of width $D = 3072$. All fitting is done at block 18, the last dual-stream block, which we identify as the most influential for these attributes by activation norm and causal patching (Appendix A.2), where we extract the image-stream delta activation $x = H_{out} - H_{in} \in \mathbb{R}^{P\times D}$. Following (Surkov et al., 2024), we fit on a random 10% of the patches per seed, using the same patch indices across the K attributes of a seed so that the pairing in equation 3 is preserved, with 800 training and 100 validation seeds. Since KAN layers place their spline grid over $[-1, 1]$, we divide every activation by the mean patch norm before fitting, and set the grid to 5 intervals with order-3 splines. A few patches carry massive norm but no concept information and would dominate a squared-error fit, so we drop any patch whose norm exceeds the 0.99 quantile of the per-seed patch norms, taken as the maximum over the K attributes (Appendix A.2). At inference we add the steering vector at every block from 18 onward (Surkov et al., 2024), rescaled by the mean patch norm to each block's activation scale.

Evaluation Metrics. We evaluate on a held-out test set of 600 seeds per attribute, 3000 per concept. All metrics are reported at a per-attribute strength α_k chosen on a calibration split, as the value that maximizes SigLIP2 (Tschannen et al., 2025) similarity between steered image and attribute prompt, subject to the attribute order $a_1 < \cdots < a_K$. **Directional-CLIP** (Gal et al., 2022), $\cos\big(\phi(I_{\alpha_k}) - \phi(I_0),\, t_k - t_0\big)$, asks whether steering moves the image in the intended direction; **CLIP-Image**, $\cos\big(\phi(I_{\alpha_k}), \phi(\mathrm{nat}_k)\big)$, is its alignment with direct prompting for the same attribute and seed; and **CLIP-T2I**, $\cos\big(\phi(I_{\alpha_k}), t_k\big)$, its agreement with attribute prompt. **LPIPS** (Zhang et al.,

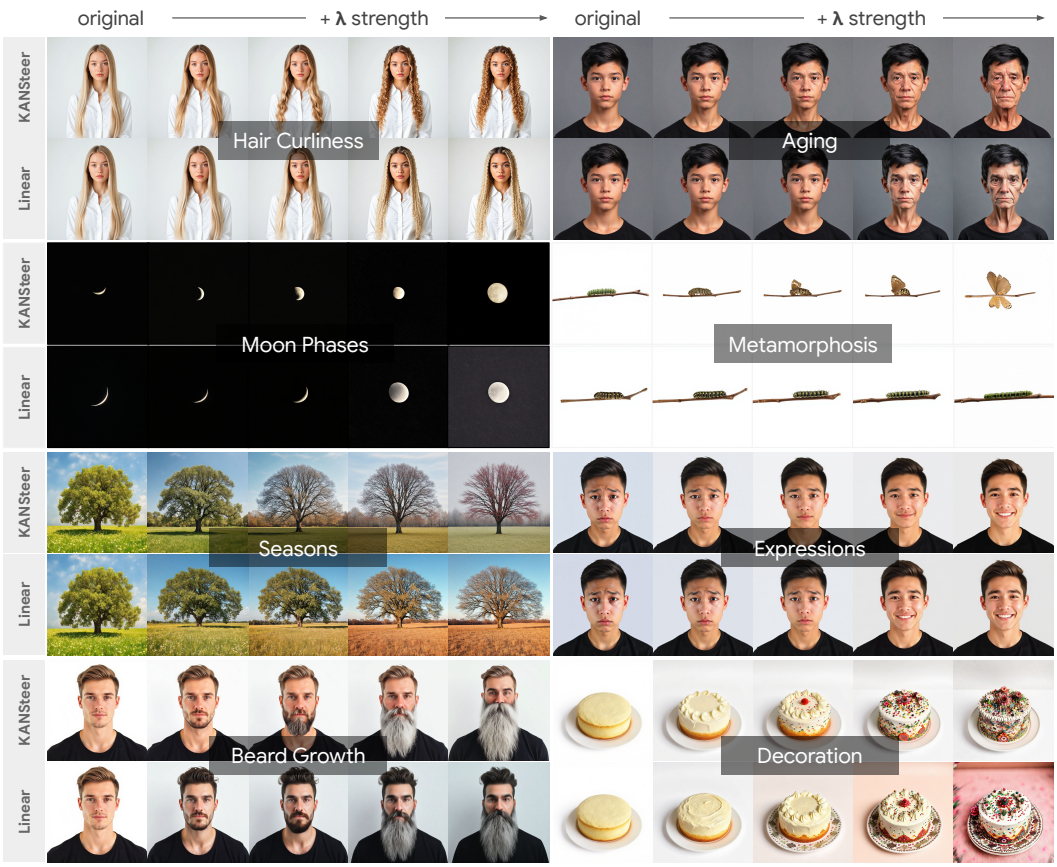

Figure 3: Qualitative comparison on FLUX.1-schnell. We sweep the steering strength from unsteered generation, comparing the fixed direction against KANsteer learned trajectory. *KANSteer traverses the attribute gradually, with intermediate columns showing intermediate attributes, while the linear direction holds the source attribute and then transitions abruptly.*

2018) is the perceptual distance to I_0, capturing change beyond the target attribute, and **FID** (Heusel et al., 2017), pooled over attributes against the prompt-generated distribution, reports if steering stays on the natural manifold. **FVU** reports how much of the model's own attribute displacement the fitted curve fails to reconstruct on held-out activations.

5.2 Results

We demonstrate results in figure 3. The fixed direction holds source attribute the most and then switches abruptly to the target. For instance, hair stays straight and then curls, and a young face becomes elderly. KANSteer advances through the attribute continuously and recovers intermediate states, giving wavy hair before curly, and a face that passes through middle age. The same pattern holds across the concepts (Appendix A.5). Where the attributes lie close to a line i.e., human age, the linear direction still reaches the target and the both methods differ mainly in continuous traversal. Where the trajectory bends, as for facial expression, hair curliness and butterfly metamorphosis, it skips intermediate attributes and forces the endpoint at the cost of leaving reference distribution.

Table 1 reports the metrics averaged over the concepts, with per-concept results in Appendix A.3. KANSteer improves on every metric at every setting. It follows the target direction more closely (directional CLIP $0.108 \rightarrow 0.170$ at 1-step and $0.053 \rightarrow 0.110$ on dev) and stays nearer the prompt-generated distribution (FID $47.9 \rightarrow 28.6$ and $46.7 \rightarrow 33.8$). The FID gap is the quantitative counterpart of the sweeps, where the linear direction reaches the target attribute only by driving the image off the natural distribution. The advantage is not specific to single-step sampling, as directional CLIP rises from 0.086 to 0.149 and FID falls from 38.2 to 28.3 at 4 steps. KANSteer also reaches

Table 1: Comparison of Linear steering and KANSteer on FLUX.1-schnell (1- and 4-step) and FLUX.1-dev, under two fitting regimes. *In-range*: the curve is fitted and steered on all ordered attributes. *Extrapolation*: the curve is fit on the first three attributes and steered over all five; Directional CLIP, CLIP-Image, and CLIP-T2I measure alignment with the target attribute; LPIPS and FID measure fidelity to the unsteered image and to the prompt generated distribution; FVU measures how well the fitted curve reconstructs the model's own attribute displacement in activation space.

Model	Method	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑	FID ↓	FVU ↓
In-range (fit: 5 attributes, steer: 5)							
FLUX.1 Schnell (1-step)	Linear	0.1079	0.8563	0.2434	0.5367	47.86	0.6187
	KANSteer	0.1703	0.9017	0.2275	0.5592	28.56	0.4674
FLUX.1 Schnell (4-step)	Linear	0.0855	0.8147	0.2243	0.5056	38.17	0.4906
	KANSteer	0.1493	0.8625	0.1932	0.5337	28.25	0.4531
FLUX.1 dev	Linear	0.0534	0.7932	0.2121	0.5006	46.73	0.5437
	KANSteer	0.1102	0.8375	0.1766	0.5320	33.79	0.4811
Extrapolation (fit: attributes 1–3, steer: 1–5)							
FLUX.1 Schnell	Linear	0.0501	0.6114	0.3754	0.4469	89.12	0.6938
	KANSteer	0.1104	0.741	0.2971	0.482	61.74	0.547

these attributes with less collateral change to the image, lowering LPIPS from 0.243 to 0.228 at 1-step and from 0.212 to 0.177 on dev, so the gain does not come from perturbing the activation more strongly. We fit the trajectory once on 1-step activations and reuse it unchanged at 4 steps and on FLUX.1-dev, which takes 50 steps, and in activation space the curve leaves less of the model's own displacement unexplained than the best line at every setting, 0.467 against 0.619 at 1-step and 0.481 against 0.544 on dev. Per concept, the two methods coincide at the source attribute and separate as steering progresses. Human age on FLUX.1-dev is the clearest case, where at the middle attribute KANSteer lowers LPIPS from 0.493 to 0.142 and raises directional CLIP from 0.137 to 0.241. On facial expression the linear direction is even slightly negative at the neutral attribute (-0.010 against 0.093), moving the image away from the state it should pass through. The concepts that gain most are those whose trajectories bend, matching the qualitative split.

5.3 Steering beyond the fitted range

The in-range results fit the trajectory on all five attributes. A natural follow-up is whether a trajectory fitted on a partial range of attributes can extrapolate to unseen ones. For instance, if a curve is fitted on first three attributes of a concept, can steering recover the remaining which are not observed during training? We train both models on a_1, a_2, a_3 and evaluate over $a_1, \ldots, a_5$, using same held-out seeds and metrics as the in-range setting. For the linear model, extrapolation simply extends the fitted line; the direction is unchanged, so the same hold-and-jump failure that Section 3 diagnosed persists at the same steering strengths. For the KANSteer, the B-spline grid covers only the coordinate range seen during training; beyond it the decoder reduces to its SiLU base, $w_b \,\mathrm{silu}(z)$, which provides a smooth monotone continuation rather than a flat cutoff. Figure 4 shows that on hair curliness, KANSteer trained on straight through wavy still produces curly and coily hair at higher steering strengths, while the linear direction overshoots into artifacts. On decoration, same pattern holds where KANSteer advances gradually from plain to elaborately decorated, whereas the linear direction drives the image off the natural distribution. The quantitative comparison in Table 1 also

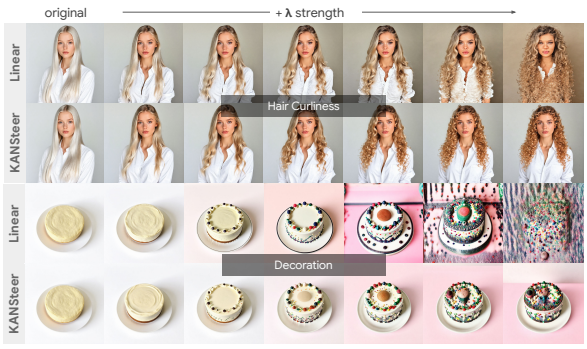

Figure 4: Extrapolating linear and KANSteer directions, both are fitted on first three attributes (a_1, a_2, a_3) and steered over all five attributes.

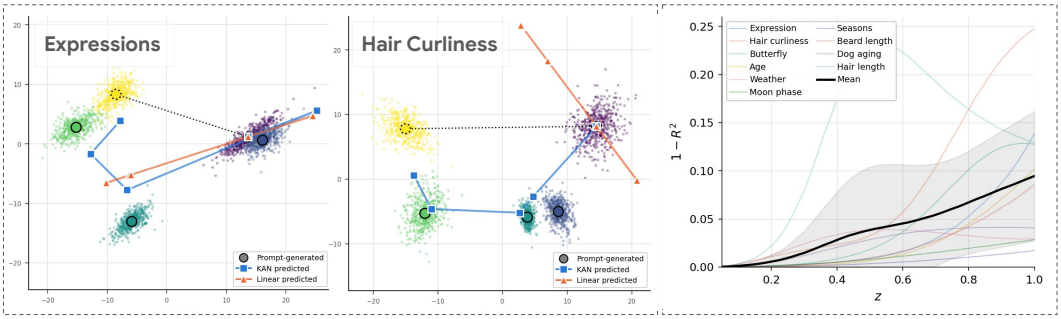

Figure 5: **Left:** PCA of block-18 activation, with per-seed scatter (dots), and model predicted centroids overlaid. *KANSteer predictions follow the curved centroid path whereas the linear predictions collapse onto a single direction that misses the curvature.* **Right:** Non-linearity $(1 - R^2)$ of KANSteer decoder curve as a function of the z interval, measured by the fraction of decoder variance not captured by the best linear fit. *Over the full attribute range the non-linear share rises to* $10 - 25\%$ *on concepts such as hair curliness and facial expression.*

confirms that KANSteer retains its advantage on unseen attributes, indicating that the fitted curve captures a continuous structure in activation space rather than memorizing training points.

5.4 Does nonlinear steering better predict the concept trajectory?

The previous sections show that KANSteer outperforms linear steering on reconstruction and perceptual quality. Here we ask a further question that does the nonlinearity in the KAN decoder actually capture the geometry of the concept trajectory? To realize this, we first measure the nonlinearity of the learned deocder curve $\gamma(z)$. The decoder function of $\phi_c(z)$ is evaluated over the empirical z-range and decomposed against orthonormal polynomials; the nonlinear share $1 - R^2$ is the fraction of centered energy above degree one. Figure 5(Right) demonstrates $1 - R^2$ as a function of sub-interval width which shows that over the full attribute range, the nonlinear share grows monotonically, reaching 25% on hair curliness and 14% on facial expression. This is consistent with a trajectory that bends gradually over the course. Secondly, we test whether this curvature predicts where concepts actually land in activation space. For each held-out seed we encode all five attribute activations through the trained model, compute the predicted displacement $\gamma_\theta\left(e_\theta(x_j)\right) - \gamma_\theta\left(e_\theta(x_i)\right)$, and average over seeds and patches to obtain centroid-level predictions. We then measure the cosine similarity between each model's predicted centroid displacement and the actual displacement. Across the concepts, KANSteer achieves a mean cosine alignment of 0.55 against 0.37 for the linear model. The gap is largest on hair curliness (0.88 vs. 0.20), where the trajectory curves the most. Figure 5(Left) shows per-seed patch-mean activations projected onto first two principal components. For each held-out seed, KANSteer and the linear model predict per-patch displacements across all five attributes; averaging these predictions over seeds and patches yields predicted centroids which shows that KANSteer follow the curved path traced by the actual prompt-generated distribution (circles), while the linear-direction collapse onto a single direction that diverges from the true trajectory.

6 Conclusion

In this paper, we analyze whether ordered visual concepts in T2I diffusion models can be traversed along a single direction, and find that in general they cannot. We therefore describe the traversal as a 1D curve and propose KANSteer, which fits it across all pairwise attribute transitions, keeping a single scalar coordinate while letting the steering vector depend on where the image already sits along the concept. Linear steering is the special case of a straight curve, so the difference between the two measures how far a concept is from a line. That difference is consistent across three FLUX.1 configurations, holds for attributes withheld from fitting, and is largest on the concepts whose trajectories bend the most. Linearity is therefore a local approximation rather than a description of a concept over its full range, and we hope that treating the traversal as a curve offers a useful starting point for interpreting and controlling non-linear concepts in generative models.

AI USE STATEMENT

We used a large language model solely to aid in polishing the writing and improving the clarity of the manuscript. The model was not involved in ideation, data analysis, or deriving any of the scientific contributions presented in this work.

ETHICS STATEMENT

We acknowledge the potential ethical implications of deploying generative models, including issues related to privacy, data misuse, and the propagation of biases. All models used in this paper are publicly available. We will release the codes to reproduce the results of this paper. We also want to point out the potential role of steering approaches in the generation of fake content, and we encourage and support responsible usage.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we will make the entire source code and scripts needed to replicate all results presented in this paper available after the peer review period. We conducted all experiments using publicly accessible models. Details of all experiments have been provided in the Appendices.

ACKNOWLEDGMENTS

REFERENCES

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Samyadeep Basu, Keivan Rezaei, Priyatham Kattakinda, Vlad I Morariu, Nanxuan Zhao, Ryan A Rossi, Varun Manjunatha, and Soheil Feizi. On mechanistic knowledge localization in text-to-image generative models. In *Forty-first International Conference on Machine Learning*, 2024a.

Samyadeep Basu, Nanxuan Zhao, Vlad I. Morariu, Soheil Feizi, and Varun Manjunatha. Localizing and editing knowledge in text-to-image generative models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b. URL https://openreview.net/forum?id=Qmw9ne6SOQ.

Stefan Andreas Baumann, Felix Krause, Michael Neumayr, Nick Stracke, Vincent Tao Hu, and Bjorn Ommer. Continuous, subject-specific attribute control in t2i models by identifying semantic directions. *Computer Vision and Pattern Recognition*, 2024. doi: 10.1109/CVPR52734.2025.01235.

Yida Chen, Fernanda Viégas, and Martin Wattenberg. Beyond surface statistics: Scene representations in a latent diffusion model. *arXiv preprint arXiv: 2306.05720*, 2023.

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

Yusuf Dalva and Pinar Yanardag. Noiseclr: A contrastive learning approach for unsupervised discovery of interpretable directions in diffusion models. *Computer Vision and Pattern Recognition*, 2023. doi: 10.1109/CVPR52733.2024.02285.

Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. *arXiv preprint arXiv: 2405.14860*, 2024.

Tatiana Gaintseva, Chengcheng Ma, Ziquan Liu, Martin Benning, Gregory Slabaugh, Jiankang Deng, and Ismail Elezi. Casteer: Steering diffusion models for controllable generation. *arXiv e-prints*, pp. arXiv–2503, 2025.

Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (ToG)*, 41(4):1–13, 2022.

Rohit Gandikota, Joanna Materzynska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adaptors for precise control in diffusion models. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XL*, volume 15098 of *Lecture Notes in Computer Science*, pp. 172–188. Springer, 2024. doi: 10.1007/978-3-031-73661-2_10. URL https://doi.org/10.1007/978-3-031-73661-2_10.

Rohit Gandikota, Zongze Wu, Richard Zhang, David Bau, Eli Shechtman, and Nick Kolkin. Sliderspace: Decomposing the visual capabilities of diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15994–16003, October 2025.

Wes Gurnee, Emmanuel Ameisen, Isaac Kauvar, Julius Tarng, Adam Pearce, Chris Olah, and Joshua Batson. When models manipulate manifolds: The geometry of a counting task. *arXiv preprint arXiv: 2601.04480*, 2026.

Alec Helbling, Tuna Han Salih Meral, Ben Hoover, Pinar Yanardag, and Duen Horng Chau. Conceptattention: Diffusion transformers learn highly interpretable features. *International Conference on Machine Learning*, 2025. doi: 10.48550/arXiv.2502.04320.

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

Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, Bryon Aragam, and Victor Veitch. On the origins of linear representations in large language models. *arXiv preprint arXiv: 2403.03867*, 2024.

Dahye Kim and Deepti Ghadiyaram. Concept steerers: Leveraging k-sparse autoencoders for test-time controllable generations. *arXiv preprint arXiv: 2501.19066*, 2025.

Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have A semantic latent space. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=pd1P2eUBVfq.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *arXiv preprint arXiv: 2306.03341*, 2023.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv: 2404.19756*, 2024.

Alexander Modell. Probing for representation manifolds in superposition. *arXiv preprint arXiv: 2605.18537*, 2026.

Rishubh Parihar, Abhijnya Bhat, Abhipsa Basu, Saswat Mallick, Jogendra Nath Kundu, and R. Venkatesh Babu. Balancing act: Distribution-guided debiasing in diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 6668–6678. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00637. URL https://doi.org/10.1109/CVPR52733.2024.00637.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *International Conference on Machine Learning*, 2023. doi: 10.48550/arXiv.2311.03658.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2406.01506.

Shivam Raval, Hae Jin Song, Linlin Wu, Abir Harrasse, Jeff M. Phillips, Fazl Barez, and Amirali Abdullah. Curveball steering: The right direction to steer isn't always linear. *arXiv preprint arXiv: 2603.09313*, 2026.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Marco Cuturi, and Xavier Suau. Controlling language and diffusion models by transporting activations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025a.

Pau Rodriguez, Michal Klein, Eleonora Gualdoni, Valentino Maiorca, Arno Blaas, Luca Zappella, Xavier Suau, et al. Lineas: End-to-end learning of activation steering with a distributional loss. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Viacheslav Surkov, Chris Wendler, Antonio Mari, Mikhail Terekhov, Justin Deschenaux, Robert West, Caglar Gulcehre, and David Bau. One-step is enough: Sparse autoencoders for text-to-image diffusion models. *arXiv preprint arXiv:2410.22366*, 2024.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear representations of sentiment in large language models. *arXiv preprint arXiv: 2310.15154*, 2023.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Evelyn Turri, Davide Bucciarelli, Sara Sarto, Lorenzo Baraldi, and Marcella Cornia. Few channels draw the whole picture: Revealing massive activations in diffusion transformers. *arXiv preprint arXiv:2605.13974*, 2026.

Daniel Wurgaft, Can Rager, Matthew Kowal, Vasudev Shyam, Sheridan Feucht, Usha Bhalla, Tal Haklay, Eric Bigelow, Raphael Sarfati, Thomas McGrath, Owen Lewis, Jack Merullo, Noah Goodman, Thomas Fel, Atticus Geiger, and Ekdeep Singh Lubana. Manifold steering reveals the shared geometry of neural network representation and behavior. *arXiv preprint arXiv: 2605.05115*, 2026.

Arman Zarei, Samyadeep Basu, Keivan Rezaei, Zihao Lin, Sayan Nag, and Soheil Feizi. Localizing knowledge in diffusion transformers. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 14584–14612. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/157a6107a95883f6ccc0e38e80a24295-Paper-Conference.pdf.

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

A Appendix

A.1 Most of the Concepts are not linear in activation space

Figure 1 shows PCA projections for three concepts; Figure 7 extends this to all remaining concepts in our evaluation set. In every case the intermediate attribute centroids deviate from the straight line between the two endpoints, confirming that the nonlinearity is not concept-specific but a general property of how diffusion models encodes ordered attributes.

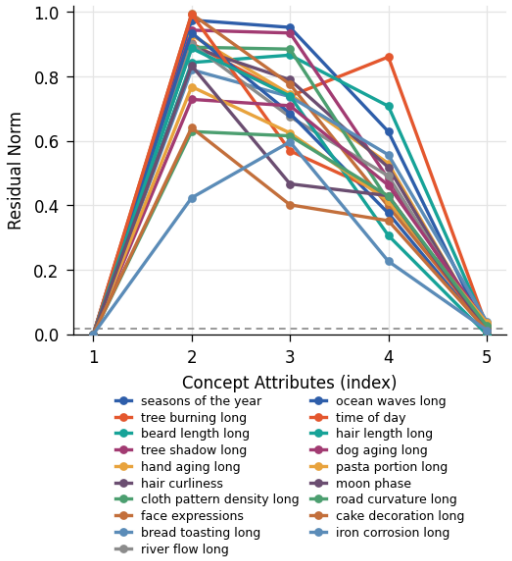

Figure 6: Residual norm ρ_k per concept. $\rho_k = 0$ means the displacement to attribute k lies exactly on the endpoint direction; $\rho_k = 1$ means it is entirely orthogonal. The dashed line marks the held-out endpoint average. Intermediate attributes (indices 2-4) are far from the line for every concept.

To quantify how far each intermediate attribute sits from the best possible linear trajectory, we compute the noise-corrected residual norm ρ_k (Eq. in Sec. 3). The steering direction $v = \mu_K - \mu_1$ is fitted on a held-out split of 700 seeds, and ρ_k is evaluated on the remaining seeds, so even the endpoint level is a genuine test rather than an identity. Centroid noise is corrected via a half-split estimator where the energy of the noise residual is subtracted from both numerator and denominator so that a perfectly collinear concept returns $\rho_k = 0$ at any sample size.

Figure 6 reports ρ_k for all concepts. At every intermediate attribute (indices 2-4) the residual norm exceeds 0.4 and typically exceeds 0.7, meaning that at best 30% of the displacement to an intermediate level lies along the endpoint direction. This is the best case for linear steering where no choice of scaling α can reduce ρ_k further. The consistently high values across concepts confirm that the nonlinearity observed in the PCA projections is quantitatively large, which ultimately indicates towards the need of a non-linear trajectory fitting method.

A.2 Localizing Activation Steering Block

We run three interventions on the block-18 update $x_p = H_p^{19} - H_p^{18}$ (Eq. 10). Each sets x_p to zero for a chosen set of patches S, so that $H_p^{19} = H_p^{18}$ for $p \in S$, and lets the remaining blocks run unchanged. All generations share the seed and prompt template within a concept.

(i) Block 18 adds the concept. Setting S to all $P = 4096$ patches removes the block's entire contribution to the image stream. The resulting images (Fig. 8, column *w/o Block-18 Output*) no longer follow the attribute prompt: the attribute-dependent appearance of the prompt-generated images (curl of the hair, expression, season) is not recovered by the later blocks. The residual stream entering block 18 is therefore not sufficient to specify the attribute, and the update that block 18

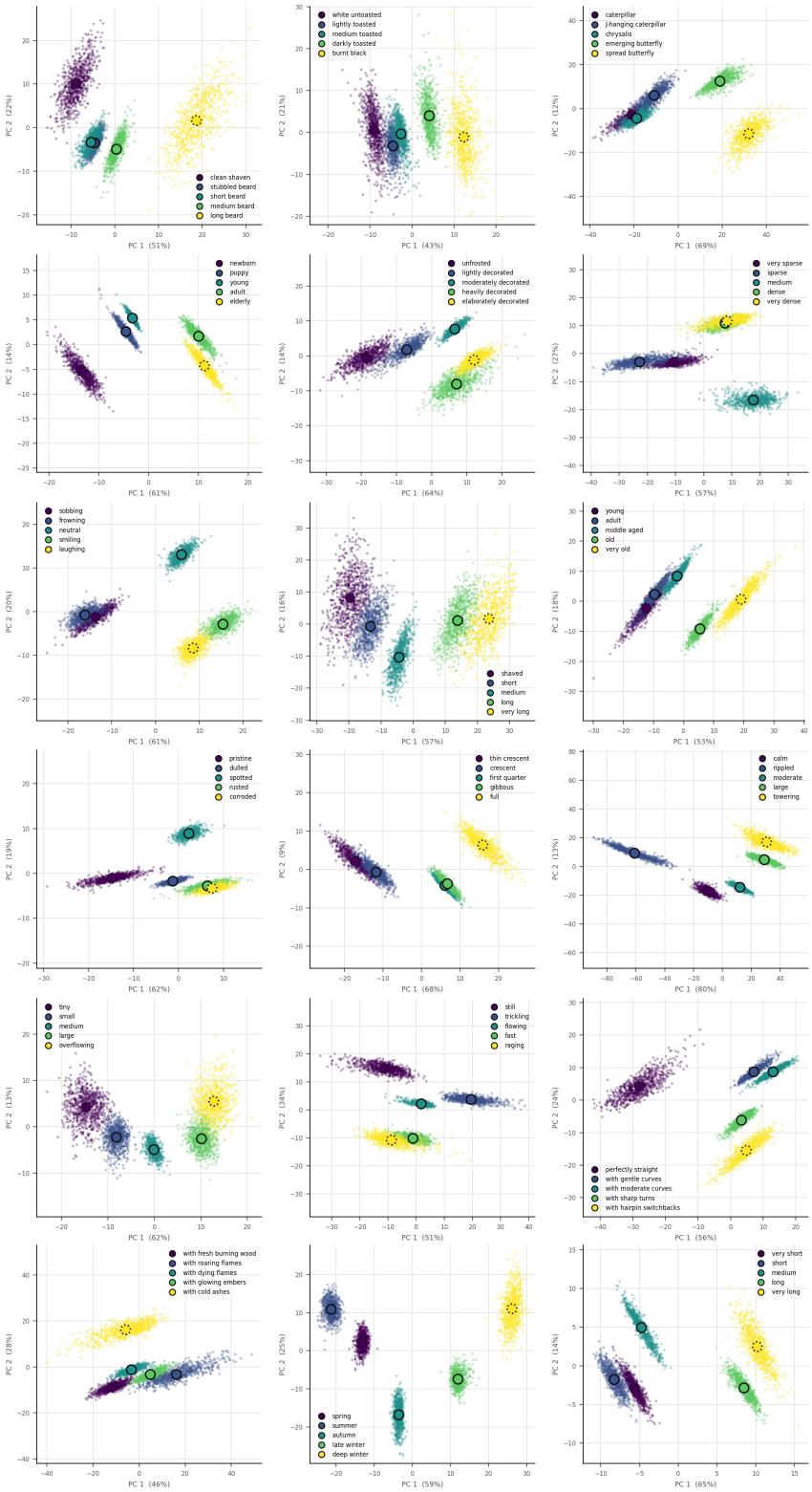

Figure 7: PCA projections of per-seed, per-attribute delta activations. Each point is one seed; large markers are attribute centroids; the dotted line connects the two endpoints. Intermediate attributes consistently bend away from the endpoint line.

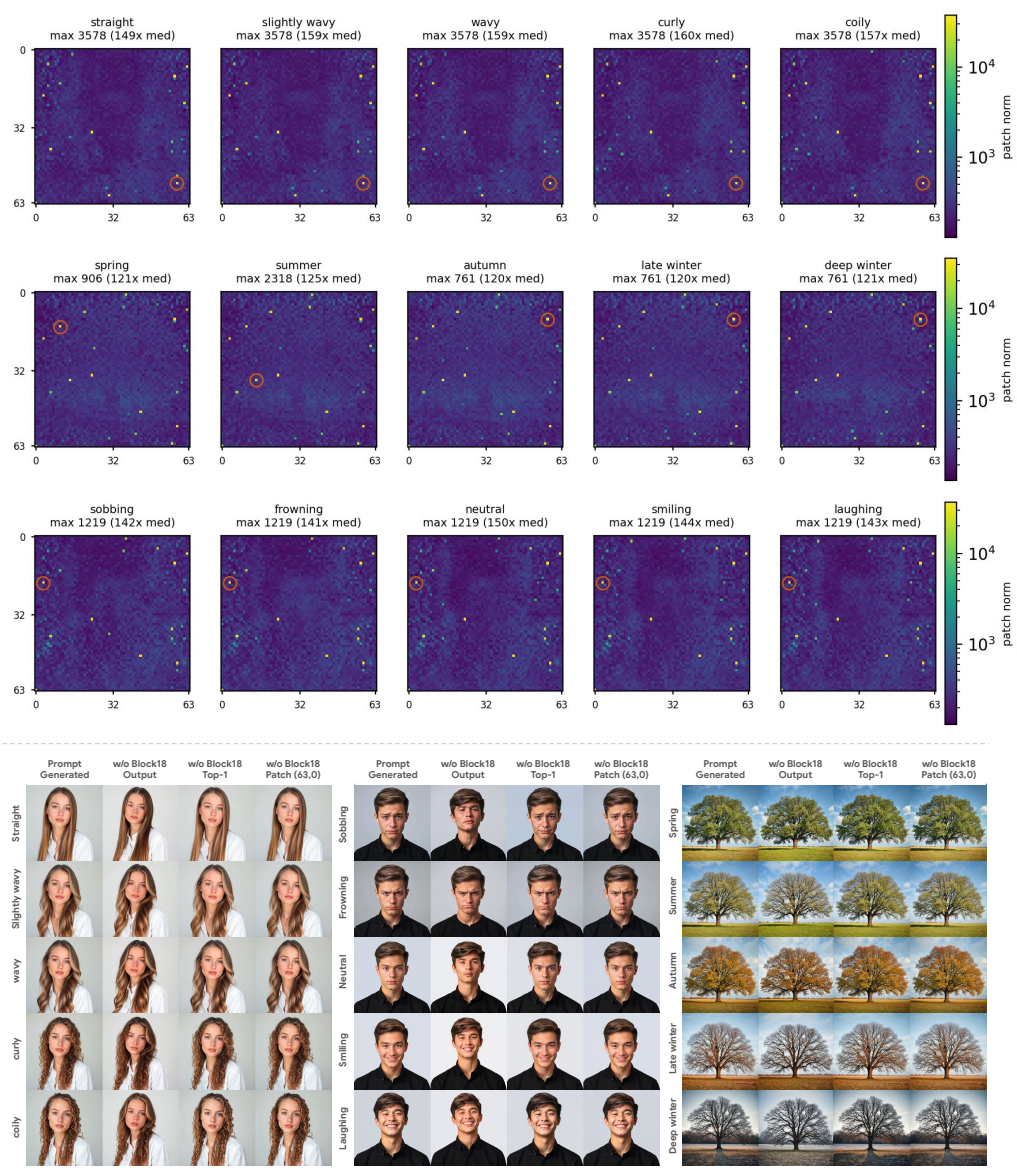

Figure 8: **Block 18 of FLUX.1 writes the concept; its massive-norm patches do not. Top:** Patch-norm maps $\|x_p\|_2$ of the block-18 image-stream delta activations for the five ordered attributes of hair curliness, facial expression and season. A handful of patches sit two orders of magnitude above the median, and for a given concept they occupy nearly the same location across attributes. **Bottom:** Generations for the same seed and prompt template when (col. 1) nothing is removed, (col. 2) the entire block-18 update is removed, (col. 3) only the top-1% patches of the update is removed, (col. 4) only the update at patch $(63, 0)$ is removed. Removing the block erases the attribute, whereas removing the massive patches leaves the generation, and the attribute, unchanged.

writes is where the attribute-specific signal is added. This is the causal counterpart of the norm-based localization and is why we fit and steer at this block.

(ii) A few patches carry very large norm, and they are causally inert. The norm maps in Fig. 8 (top) are far from uniform. The maximal patch norm is 120–160$\times$ the median patch norm in every panel, and the extreme patches are isolated single positions on the grid rather than regions. Their locations are governed by the image content and the seed, not by the attribute: for hair curliness and facial expression the top-1 position is identical across all five attributes, and for the season

concept it is identical across the last three. One position, $(63, 0)$ (flat index 4032), is special in that it is flagged as an extreme patch in every seed and every concept we inspected, unlike all other positions, which move with the image. Consistent with the register and sink tokens reported in vision transformers (Turri et al., 2026), these patches do not lie on the object but at positions where the model stores global computation. To test whether they carry anything the generation depends on, we set $S = \{p^\star\}$ with $p^\star = \arg\max_p \|x_p\|$ (*w/o Top-1*), and $S = \{(63, 0)\}$ (*w/o Patch* $(63, 0)$). In both cases the generated images are visually indistinguishable from the unablated ones, and in particular the attribute is preserved (Fig. 8, columns 3 and 4). Contrast this with (i): removing one of 4096 patches has no visible effect, whereas removing all of them removes the concept.

(iii) Removing the top 1% is safe, and necessary for fitting. The objective in Eq. 3 is a squared error, so a patch contributes in proportion to $\|x_p\|^2$, i.e. 10^4 to $2.5 \cdot 10^4$ times a median patch at the ratios above. As an order-of-magnitude estimate, if a fraction $\epsilon = 0.5\%$ of patches has norm $r = 140\times$ the median, their share of the total squared norm is $\epsilon r^2/(\epsilon r^2 + 1 - \epsilon) \approx 99\%$. Unfiltered, the fit is thus determined by a few dozen patches per image, whose content we show above to be irrelevant to the attribute, and the remainder of the patches, which do carry the concept, contribute little to the loss. We therefore drop, per image, every patch whose norm exceeds the 99th percentile of the patch norms ($\approx$ 41 of 4096 patches). To check that this discards no concept information, we generate with the update zeroed on exactly this set, $S = \{p : \|x_p\| > q_{0.99}\}$ (*w/o Top-1%*). Combined with (ii), this shows that the filtered patches are inert with respect to the concept and that excluding them from the fit removes a numerical artifact of the loss, not signal. In the fit, the percentile is taken on the input norm as the maximum over the K attributes, so a position flagged under any attribute is excluded from all K (Sec. 5.1).

A.3 Detailed Quantitative Results

A.4 Prompts for generating the concepts

In this section, we list the concept along with the attributes, and the prompt templates used to compute steering vectors for our experiments. Each concept is specified as a template $T(\cdot)$ into which one attribute is substituted, together with an ordered attribute set $A = (a_1, \ldots, a_5)$; the prompt for attribute k is $T(a_k)$, so that consecutive prompts differ only in the substituted attribute. The fixed text in each template holds the subject, pose, framing and background constant across the five attributes.

Facial emotion. $T(\cdot)$ = "a young man with a $\{\cdot\}$ face, wearing black shirt, facing the camera, straight posture, plain background".
A = (`sobbing`, `frowning`, `neutral`, `smiling`, `laughing`).

Hair curliness. $T(\cdot)$ = "a young white woman with long $\{\cdot\}$ hair, wearing white shirt, facing the camera, straight posture, plain background".
A = (`straight`, `slightly wavy`, `wavy`, `curly`, `coily`).

Tree across seasons. $T(\cdot)$ = "an oak tree in an open field in $\{\cdot\}$, centered, wide shot".
A = (`spring`, `summer`, `autumn`, `late winter`, `deep winter`).

Time of day. $T(\cdot)$ = "$\{\cdot\}$ photograph of a straight two-lane asphalt highway cutting through an empty flat plain".
A = (`bluehour`, `sunrise`, `midday`, `evening`, `night`).

Moon phase. $T(\cdot)$ = "a telephoto photo of a $\{\cdot\}$ moon, lit from the right side, centered in a black night sky".
A = (`thin crescent`, `crescent`, `first quarter`, `gibbous`, `full`).

Human age. $T(\cdot)$ = "$\{\cdot\}$ male, short black hair, wearing a black t-shirt, facing the camera, straight posture, plain grey background".
A = (`preteen`, `teenage`, `young adult`, `middle-aged`, `elderly`).

Table 2: FLUX.1 Schnell — Comparison of linear and KAN directions across ten semantic concepts.

(a) Hair

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Straight	Linear	0	0.990	0	0.4909
	KAN	0	0.999	0	0.4909
Slightly wavy	Linear	0.083	0.889	0.252	0.4779
	KAN	**0.192**	**0.923**	**0.260**	**0.5098**
Wavy	Linear	0.124	0.869	0.311	0.5081
	KAN	**0.209**	**0.911**	**0.313**	**0.5352**
Curly	Linear	0.187	0.880	0.360	0.5213
	KAN	**0.271**	**0.931**	**0.355**	**0.5567**
Coily	Linear	0.139	0.803	0.493	0.4819
	KAN	**0.283**	**0.915**	**0.386**	**0.5604**

(b) Moon Phase

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Thin crescent	Linear	0	0.990	0	0.6584
	KAN	0	0.999	0	0.6581
Crescent	Linear	0.015	0.914	0.005	0.6541
	KAN	**0.019**	**0.973**	**0**	**0.659**
First quarter	Linear	0.001	0.868	0.024	0.6169
	KAN	**0.018**	**0.889**	**0.009**	**0.6214**
Gibbous	Linear	0.146	0.849	0.079	0.5491
	KAN	**0.162**	**0.891**	**0.031**	**0.5732**
Full	Linear	0.195	0.854	0.229	0.5547
	KAN	**0.227**	**0.901**	**0.041**	**0.5661**

(c) Age

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Preteen	Linear	0	0.960	0	0.582
	KAN	0	0.996	0	0.5709
Teenage	Linear	0.014	0.816	0.121	0.5108
	KAN	**0.070**	**0.900**	**0.022**	**0.5206**
Young adult	Linear	0.056	0.726	0.171	0.4151
	KAN	**0.084**	**0.852**	**0.041**	**0.4283**
Middle-aged	Linear	0.194	0.619	0.223	0.4584
	KAN	**0.270**	**0.724**	**0.149**	**0.4756**
Elderly	Linear	0.256	0.694	0.274	0.4992
	KAN	**0.314**	**0.784**	**0.230**	**0.5446**

(d) Expression

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Sobbing	Linear	0	0.990	0	0.4743
	KAN	0	0.999	0	0.4774
Frowning	Linear	0.011	0.882	0.100	0.4885
	KAN	**0.075**	**0.884**	**0.068**	**0.4952**
Neutral	Linear	-0.010	0.852	0.048	0.4859
	KAN	**0.093**	**0.883**	**0.007**	**0.4898**
Smiling	Linear	0.160	0.884	0.188	0.4695
	KAN	**0.184**	**0.896**	**0.246**	**0.4715**
Laughing	Linear	0.177	0.861	0.219	0.4341
	KAN	**0.195**	**0.881**	**0.294**	**0.4569**

(e) Weather

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Bright sunny	Linear	0	0.986	0	0.662
	KAN	0	1.000	0	0.6598
Pale hazy	Linear	0.045	0.897	0.093	0.5583
	KAN	**0.097**	**0.911**	**0.062**	**0.5656**
Uniform overcast	Linear	0.209	0.867	0.476	0.6487
	KAN	**0.281**	**0.898**	**0.437**	**0.6492**
Dark leaden	Linear	0.312	0.881	0.559	0.6727
	KAN	**0.353**	**0.929**	**0.527**	**0.6938**
Black storm	Linear	0.249	0.724	0.742	0.6496
	KAN	**0.374**	**0.934**	**0.538**	**0.7148**

(f) Season

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Spring	Linear	0	0.974	0	0.6401
	KAN	0	0.972	0	0.6425
Summer	Linear	0.041	0.981	0.231	0.6343
	KAN	**0.033**	**0.974**	**0.256**	**0.6351**
Autumn	Linear	0.155	0.969	0.193	0.5734
	KAN	**0.098**	**0.986**	**0.342**	**0.5784**
Late winter	Linear	0.184	0.910	0.303	0.5612
	KAN	**0.215**	**0.965**	**0.414**	**0.5690**
Deep winter	Linear	0.194	0.890	0.370	0.5338
	KAN	**0.303**	**0.934**	**0.558**	**0.5878**

(g) Butterfly

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Caterpillar	Linear	0	0.958	0	0.5815
	KAN	0	0.961	0	0.5805
J-hanging caterpillar	Linear	0.0007	0.875	0.086	0.5702
	KAN	**0.008**	**0.886**	**0.052**	**0.5930**
Chrysalis	Linear	0.031	0.832	0.054	0.5203
	KAN	**0.054**	**0.863**	0.197	**0.5340**
Emerging butterfly	Linear	0.019	0.796	0.062	0.4769
	KAN	**0.077**	**0.817**	0.206	**0.5003**
Spread butterfly	Linear	0.012	0.788	0.068	0.4662
	KAN	**0.108**	**0.822**	0.229	**0.4821**

(h) Beard

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Clean shaven	Linear	0	0.927	0	0.4551
	KAN	0	0.907	0	0.4532
Stubbled	Linear	0.113	0.848	0.212	0.5073
	KAN	**0.119**	**0.883**	**0.195**	**0.5353**
Short beard	Linear	0.092	0.846	0.276	0.5286
	KAN	**0.185**	**0.893**	**0.221**	**0.5452**
Medium beard	Linear	0.129	0.864	0.350	0.5969
	KAN	**0.222**	**0.913**	**0.316**	**0.6077**
Long beard	Linear	0.139	0.880	0.422	0.5781
	KAN	**0.251**	**0.937**	**0.390**	**0.6367**

(i) Dog Age

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Newborn	Linear	0	0.945	0	0.6371
	KAN	0	0.939	0	0.6338
Puppy	Linear	-0.009	0.970	0.170	0.6328
	KAN	**0.007**	**0.990**	**0.090**	**0.6378**
Young	Linear	0.093	0.983	0.290	0.6614
	KAN	**0.134**	**0.989**	**0.170**	**0.6664**
Adult	Linear	0.164	0.946	0.476	0.6121
	KAN	**0.257**	**0.973**	**0.360**	**0.6362**
Elderly	Linear	0.274	0.917	0.635	0.6283
	KAN	**0.310**	**0.956**	**0.435**	**0.6412**

(j) Hair Length

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Shaved	Linear	0	0.955	0	0.5709
	KAN	0	0.952	0	0.5760
Short	Linear	-0.030	0.834	0.243	0.4614
	KAN	**0.067**	**0.875**	**0.022**	**0.4714**
Medium	Linear	0.046	0.862	0.086	0.4766
	KAN	**0.200**	**0.918**	**0.068**	**0.5236**
Long	Linear	0.038	0.837	0.110	0.4056
	KAN	**0.176**	**0.868**	**0.275**	**0.4637**
Very long	Linear	0.029	0.796	0.134	0.3885
	KAN	**0.219**	**0.819**	**0.289**	**0.4383**

Table 3: FLUX.1 Schnell 4 steps — Comparison of linear and KAN directions across ten semantic concepts.

(a) Hair

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Straight	Linear	0	0.939	0	0.4867
	KAN	0	0.946	0	0.4742
Slightly wavy	Linear	-0.009	0.861	0.263	0.4289
	KAN	**0.11**	**0.865**	**0.168**	**0.4713**
Wavy	Linear	0.025	0.833	0.318	0.4519
	KAN	**0.13**	**0.848**	**0.216**	**0.4929**
Curly	Linear	0.083	0.746	0.379	0.4524
	KAN	**0.186**	**0.855**	**0.36**	**0.5162**
Coily	Linear	0.079	0.638	0.579	0.4181
	KAN	**0.207**	**0.673**	**0.456**	**0.4463**

(b) Moon Phase

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Thin crescent	Linear	0	0.957	0	0.6537
	KAN	0	0.937	0	0.6469
Crescent	Linear	0.019	0.943	0.008	0.6501
	KAN	**0.021**	**0.961**	**0.005**	**0.6645**
First quarter	Linear	0.025	0.888	0.019	0.6209
	KAN	**0.7**	**0.901**	**0.01**	**0.6426**
Gibbous	Linear	0.195	0.86	0.083	0.5745
	KAN	**0.209**	**0.899**	**0.032**	**0.5747**
Full	Linear	0.227	0.818	0.25	0.555
	KAN	**0.26**	**0.927**	**0.041**	**0.5746**

(c) Age

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Preteen	Linear	0	0.96	0	0.582
	KAN	0	0.944	0	0.5705
Teenage	Linear	0.035	0.856	0.075	0.4912
	KAN	**0.1**	**0.873**	**0.025**	**0.5021**
Young adult	Linear	0.103	0.776	0.129	0.3673
	KAN	**0.112**	**0.802**	**0.047**	**0.3792**
Middle-aged	Linear	0.137	0.69	0.109	0.4688
	KAN	**0.253**	**0.74**	**0.083**	**0.4735**
Elderly	Linear	0.267	0.719	0.214	0.5693
	KAN	**0.309**	**0.736**	**0.235**	**0.5788**

(d) Expression

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Sobbing	Linear	0	0.828	0	0.4778
	KAN	0	0.858	0	0.4702
Frowning	Linear	-0.003	0.857	0.108	0.483
	KAN	**0.001**	**0.862**	**0.072**	**0.4849**
Neutral	Linear	-0.029	0.816	0.053	0.5089
	KAN	**0.017**	**0.828**	**0.033**	**0.5139**
Smiling	Linear	0.094	0.825	0.228	0.4784
	KAN	**0.128**	**0.845**	**0.194**	**0.4921**
Laughing	Linear	0.146	0.816	0.293	0.4475
	KAN	**0.12**	**0.834**	**0.224**	**0.4587**

(e) Weather

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Bright sunny	Linear	0	0.941	0	0.6735
	KAN	0	0.942	0	0.6802
Pale hazy	Linear	-0.027	0.85	0.156	0.5629
	KAN	**0.157**	**0.86**	**0.12**	**0.5696**
Uniform overcast	Linear	0.12	0.833	0.556	0.6252
	KAN	**0.201**	**0.851**	**0.491**	**0.6391**
Dark leaden	Linear	0.172	0.772	0.727	0.6309
	KAN	**0.269**	**0.857**	**0.638**	**0.6646**
Black storm	Linear	0.206	0.692	0.958	0.6665
	KAN	**0.262**	**0.786**	**0.73**	**0.7083**

(f) Season

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Spring	Linear	0	0.945	0	0.6221
	KAN	0	0.951	0	0.6252
Summer	Linear	0.078	0.953	0.346	0.6278
	KAN	**0.079**	**0.958**	**0.306**	**0.6346**
Autumn	Linear	0.092	0.946	0.239	0.5596
	KAN	**0.122**	**0.95**	**0.2**	**0.5662**
Late winter	Linear	0.123	0.886	0.522	0.5346
	KAN	**0.163**	**0.945**	**0.339**	**0.5423**
Deep winter	Linear	0.113	0.801	0.634	0.5504
	KAN	**0.169**	**0.82**	**0.673**	**0.582**

(g) Butterfly

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Caterpillar	Linear	0	0.929	0	0.5713
	KAN	0	0.935	0	0.5835
J-hanging caterpillar	Linear	-0.009	0.873	0.039	0.5611
	KAN	**0.005**	**0.883**	**0**	**0.5724**
Chrysalis	Linear	0.009	0.798	0.048	0.5067
	KAN	**0.016**	**0.826**	**0.033**	**0.5149**
Emerging butterfly	Linear	-0.009	0.741	0.057	0.4722
	KAN	**0.024**	**0.789**	**0.068**	**0.4865**
Spread butterfly	Linear	0.013	0.691	0.07	0.4374
	KAN	**0.029**	**0.775**	**0.097**	**0.4448**

(h) Beard

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Clean shaven	Linear	0	0.912	0	0.4479
	KAN	0	0.899	0	0.4397
Stubbled	Linear	0.009	0.814	0.194	0.4937
	KAN	**0.114**	**0.867**	**0.176**	**0.5081**
Short beard	Linear	0.087	0.824	0.241	0.5140
	KAN	**0.173**	**0.873**	**0.199**	**0.5318**
Medium beard	Linear	0.122	0.841	0.315	0.5867
	KAN	**0.218**	**0.908**	**0.284**	**0.5941**
Long beard	Linear	0.143	0.860	0.397	0.5619
	KAN	**0.245**	**0.922**	**0.354**	**0.6235**

(i) Dog Age

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Newborn	Linear	0	0.939	0	0.6190
	KAN	0	0.944	0	0.6200
Puppy	Linear	-0.003	0.964	0.210	0.6499
	KAN	**0.004**	**0.971**	**0.130**	**0.6571**
Young	Linear	0.086	0.967	0.350	0.6770
	KAN	**0.149**	**0.979**	**0.210**	**0.6896**
Adult	Linear	0.157	0.938	0.492	0.6203
	KAN	**0.239**	**0.960**	**0.395**	**0.6289**
Elderly	Linear	0.284	0.921	0.655	0.6178
	KAN	**0.326**	**0.968**	**0.469**	**0.6565**

(j) Hair Length

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Shaved	Linear	0	0.954	0	0.5514
	KAN	0	0.959	0	0.5558
Short	Linear	0.005	0.811	0.219	0.4584
	KAN	**0.078**	**0.879**	**0.196**	**0.4625**
Medium	Linear	0.034	0.854	0.071	0.4649
	KAN	**0.214**	**0.901**	**0.049**	**0.5074**
Long	Linear	0.049	0.829	0.160	0.3849
	KAN	**0.156**	**0.844**	**0.282**	**0.4512**
Very long	Linear	0.063	0.762	0.156	0.3975
	KAN	**0.228**	**0.801**	**0.293**	**0.4358**

Table 4: FLUX.1 Dev — Comparison of linear and KAN directions across ten semantic concepts.

(a) Hair

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Straight	Linear	0	0.857	0	0.4041
	KAN	0	0.869	0	0.4327
Slightly wavy	Linear	0.046	0.826	0.209	0.4428
	KAN	**0.063**	**0.84**	**0.16**	**0.4652**
Wavy	Linear	0.06	0.814	0.256	0.4395
	KAN	**0.097**	**0.845**	**0.209**	**0.4684**
Curly	Linear	0.06	0.79	0.299	0.4502
	KAN	**0.14**	**0.869**	**0.255**	**0.4784**
Coily	Linear	0.105	0.772	0.33	0.4443
	KAN	**0.17**	**0.798**	**0.296**	**0.4923**

(b) Moon Phase

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Thin crescent	Linear	0	0.882	0	0.6389
	KAN	0	0.888	0	0.6518
Crescent	Linear	-0.014	0.85	0.019	0.6157
	KAN	**0.006**	**0.909**	**0.011**	**0.6479**
First quarter	Linear	0.012	0.734	0.05	0.506
	KAN	**0.017**	**0.858**	**0.021**	**0.6058**
Gibbous	Linear	0.019	0.707	0.026	0.4632
	KAN	**0.067**	**0.847**	**0.199**	**0.5243**
Full	Linear	0.093	0.669	0.092	0.4442
	KAN	**0.128**	**0.83**	**1.071**	**0.4846**

(c) Age

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Preteen	Linear	0	0.849	0	0.5415
	KAN	0	0.849	0	0.5415
Teenage	Linear	0.014	0.764	0.123	0.4499
	KAN	**0.038**	**0.79**	**0.088**	**0.4596**
Young adult	Linear	0.037	0.71	0.14	0.3561
	KAN	**0.084**	**0.732**	**0.055**	**0.3653**
Middle-aged	Linear	0.137	0.548	0.493	0.4475
	KAN	**0.241**	**0.637**	**0.142**	**0.4661**
Elderly	Linear	0.259	0.593	0.634	0.5587
	KAN	**0.306**	**0.694**	**0.285**	**0.5644**

(d) Expression

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Sobbing	Linear	0	0.66	0	0.4506
	KAN	0	0.66	**0**	0.455
Frowning	Linear	0.005	0.696	0.246	0.452
	KAN	**0.007**	**0.753**	**0.141**	**0.4685**
Neutral	Linear	0.018	0.733	0.101	0.4601
	KAN	**0.031**	**0.761**	**0.037**	**0.4795**
Smiling	Linear	0.115	0.807	0.175	0.474
	KAN	**0.138**	**0.813**	**0.142**	**0.4874**
Laughing	Linear	0.129	0.811	0.184	0.4403
	KAN	**0.174**	**0.814**	**0.159**	**0.464**

(e) Weather

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Bright sunny	Linear	0	0.899	0	0.6638
	KAN	0	0.899	0	0.6694
Pale hazy	Linear	0.014	0.828	0.049	0.5988
	KAN	**0.104**	**0.84**	**0.027**	**0.6243**
Uniform overcast	Linear	0.044	0.791	0.308	0.5292
	KAN	**0.153**	**0.826**	**0.12**	**0.629**
Dark leaden	Linear	0.041	0.798	0.424	0.4556
	KAN	**0.211**	**0.842**	**0.135**	**0.6109**
Black storm	Linear	0.038	0.735	0.614	0.4714
	KAN	**0.203**	**0.804**	**0.355**	**0.6207**

(f) Season

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Spring	Linear	0	0.829	0	0.5913
	KAN	0	0.824	0	0.5904
Summer	Linear	0.009	0.921	0.109	0.6142
	KAN	**0.037**	**0.927**	**0.114**	**0.6289**
Autumn	Linear	0.069	0.909	0.387	0.5482
	KAN	**0.087**	**0.915**	**0.15**	**0.5572**
Late winter	Linear	0.133	0.89	0.372	0.5305
	KAN	**0.207**	**0.911**	**0.424**	**0.5422**
Deep winter	Linear	0.12	0.73	0.577	0.5434
	KAN	**0.223**	**0.88**	**0.678**	**0.5441**

(g) Butterfly

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Caterpillar	Linear	0	0.861	0	0.5811
	KAN	0	0.867	0	0.5819
J-hanging caterpillar	Linear	0.003	0.84	0.084	0.5654
	KAN	**0.009**	**0.855**	**0.074**	**0.5676**
Chrysalis	Linear	0.012	0.764	0.127	0.4911
	KAN	**0.017**	**0.774**	**0.115**	**0.5094**
Emerging butterfly	Linear	0.061	0.753	0.149	0.4293
	KAN	**0.147**	**0.774**	0.243	**0.4831**
Spread butterfly	Linear	0.066	0.718	0.166	0.374
	KAN	**0.198**	**0.746**	0.268	**0.4309**

(h) Beard

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Clean shaven	Linear	0	0.870	0	0.4230
	KAN	0	0.869	0	0.4250
Stubbled	Linear	0.013	0.809	0.215	0.4500
	KAN	**0.085**	**0.842**	**0.195**	**0.4561**
Short beard	Linear	0.056	0.741	0.239	0.4710
	KAN	**0.139**	**0.833**	**0.197**	**0.4793**
Medium beard	Linear	0.097	0.795	0.298	0.4712
	KAN	**0.188**	**0.859**	**0.216**	**0.4950**
Long beard	Linear	0.123	0.836	0.357	0.5188
	KAN	**0.195**	**0.891**	**0.228**	**0.5329**

(i) Dog Age

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Newborn	Linear	0	0.922	0	0.5910
	KAN	0	0.927	0	0.5890
Puppy	Linear	-0.008	0.895	0.175	0.5764
	KAN	**0.013**	**0.900**	**0.113**	**0.5813**
Young	Linear	0.012	0.870	0.236	0.6180
	KAN	**0.109**	**0.891**	**0.179**	**0.6425**
Adult	Linear	0.086	0.913	0.297	0.6187
	KAN	**0.178**	**0.955**	**0.234**	**0.6610**
Elderly	Linear	0.162	0.901	0.378	0.6248
	KAN	**0.244**	**0.946**	**0.245**	**0.6890**

(j) Hair Length

Attribute	Model	dir-CLIP ↑	CLIP-I ↑	LPIPS ↓	CLIP-T2I ↑
Shaved	Linear	0	0.942	0	0.5620
	KAN	0	0.940	0	0.5600
Short	Linear	0.009	0.792	0.175	0.4312
	KAN	**0.067**	**0.816**	**0.143**	**0.4358**
Medium	Linear	0.027	0.784	0.196	0.4470
	KAN	**0.160**	**0.843**	**0.149**	**0.4600**
Long	Linear	0.081	0.790	0.254	0.4010
	KAN	**0.187**	**0.861**	**0.185**	**0.4870**
Very long	Linear	0.114	0.734	0.270	0.4250
	KAN	**0.199**	**0.872**	**0.203**	**0.4892**

Weather severity. $T(\cdot)$ = "$\{\cdot\}$ skies over a featureless flat grass plain meeting a straight unbroken horizon, no trees, no hills, no buildings, eye-level wide shot, sky filling the upper two thirds of the frame".
A = (`bright sunny`, `pale hazy`, `uniform overcast`, `dark leaden`, `black storm`).

Butterfly metamorphosis. $T(\cdot)$ = "a $\{\cdot\}$, on a single horizontal bare twig, centered and filling the frame, macro photo, plain white background, no leaves, no flowers".
A = (`caterpillar`, `J-hanging caterpillar`, `chrysalis`, `emerging butterfly`, `spread butterfly`).

Dog aging. $T(\cdot)$ = "a side-view photo of a $\{\cdot\}$ golden retriever standing on a plain grey floor, plain white background, even lighting".
A = (`newborn`, `puppy`, `young`, `adult`, `elderly`).

Hair length. $T(\cdot)$ = "a frontal studio portrait of a young man with $\{\cdot\}$ hair, black t-shirt, shoulders square to the camera, plain grey background".
A = (`shaved`, `short`, `medium`, `long`, `very-long`).

Beard length. $T(\cdot)$ = "a frontal studio portrait of a young man with a $\{\cdot\}$ beard, black t-shirt, shoulders square to the camera, plain grey background".
A = (`shaven`, `stubbled`, `short`, `full`, `bushy`).

Tree shadow. $T(\cdot)$ = "a wide photo of a single oak tree on an open grass field casting a $\{\cdot\}$ shadow, plain sky, fixed viewpoint".
A = (`faint`, `soft`, `clear`, `long`, `deep`).

Hand aging. $T(\cdot)$ = "a close-up photo of a single $\{\cdot\}$ human hand held palm-down against a plain grey surface, even lighting".
A = (`teenage`, `young`, `adult`, `aging`, `elderly`).

Iron corrosion. $T(\cdot)$ = "a front-view photo of a flat iron plate filling the frame, the metal $\{\cdot\}$, overcast daylight, plain background".
A = (`pristine`, `dulled`, `spotted`, `rusted`, `corroded`).

Cake decoration. $T(\cdot)$ = "a front-view photo of a single round cake on a white plate, the cake $\{\cdot\}$, plain grey background, even lighting".
A = (`plain`, `lightly-decorated`, `decorated`, `heavily-decorated`, `ornate`).

Tree burning. $T(\cdot)$ = "a wide photo of a single tree on a plain field, the tree $\{\cdot\}$, plain sky, fixed viewpoint".
A = (`green`, `wilting`, `burning`, `charred`, `ashen`).

Cloth pattern density. $T(\cdot)$ = "a flat top-view photo of a square fabric swatch filling the frame, printed with a $\{\cdot\}$ pattern, even lighting".
A = (`sparse`, `light`, `moderate`, `dense`, `packed`).

Pasta portion. $T(\cdot)$ = "a top-view photo of a white round plate on a plain grey table holding a $\{\cdot\}$ portion of spaghetti, even lighting".
A = (`tiny`, `small`, `medium`, `large`, `overflowing`).

River flow. $T(\cdot)$ = "a wide photo of a river running through a plain rocky bank, the water $\{\cdot\}$, overcast daylight, fixed viewpoint".
A = (`still`, `trickling`, `flowing`, `fast`, `raging`).

Road curvature. $T(\cdot)$ = "an aerial photo of a single asphalt road crossing an empty green plain, the road $\{\cdot\}$, overcast daylight, fixed viewpoint".
A = (`straight`, `gently-curved`, `curved`, `winding`, `hairpin`).

Ocean waves. $T(\cdot)$ = "a wide photo of the open ocean surface under an overcast sky, the water showing $\{\cdot\}$ waves, fixed viewpoint, horizon centered".
A = (`calm`, `rippled`, `moderate`, `large`, `towering`).

Bread toasting. $T(\cdot)$ = "a top-view photo of a single slice of plain white bread on a white plate, the bread $\{\cdot\}$, even lighting, plain background".
A = (`fresh`, `lightly-toasted`, `golden`, `dark-toasted`, `burnt`).

A.5 Qualitative Results

Figures 9–20 show steering sweeps for the concepts on FLUX.1 Schnell. The linear direction and KANSteer are compared under the same conditions, so that within a figure both share the noise, the prompt template and the grid of steering strengths, and the leftmost column is the unsteered generation at $\alpha = 0$, which is identical for the two methods by construction. Every curve is fit once on 1-step block-18 activations (Sec. 5.1) and is applied here without refitting.



Figure 9: **Cake decoration (plain → decorated):** Comparison the Linear and KANSteer steering on FLUX.1 Schnell.

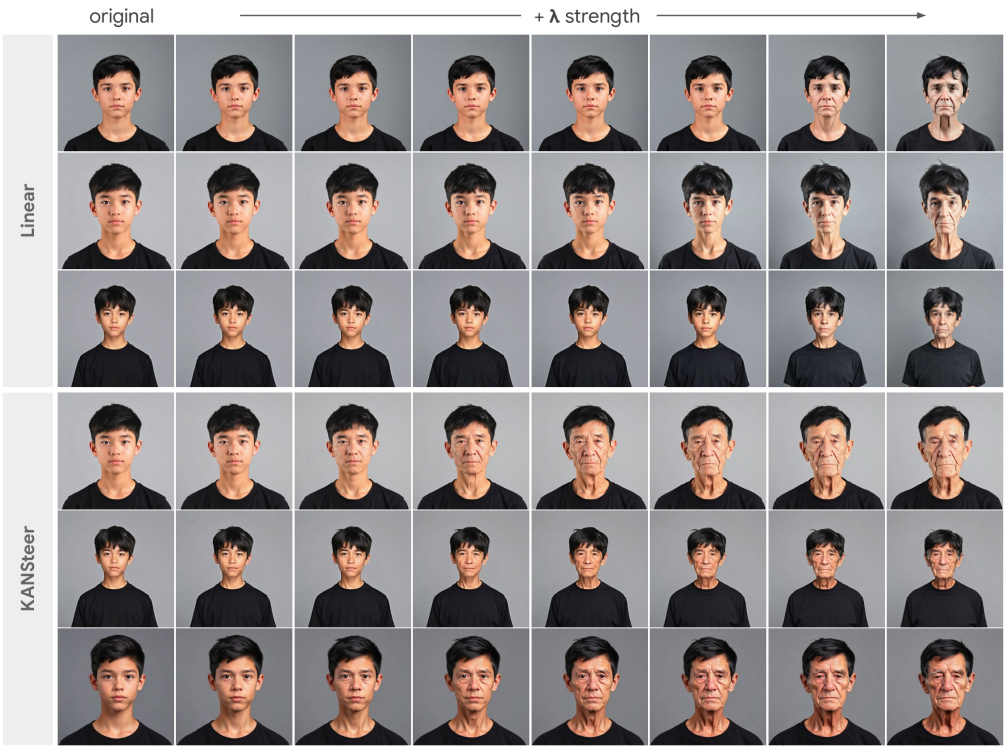

Figure 10: **Human Aging (preteen → elderly):** Comparison the Linear and KANSteer steering on FLUX.1 Schnell.



Figure 11: **Portion Size (tiny → overflowing):** Comparison the Linear and KANSteer steering on FLUX.1 Schnell.

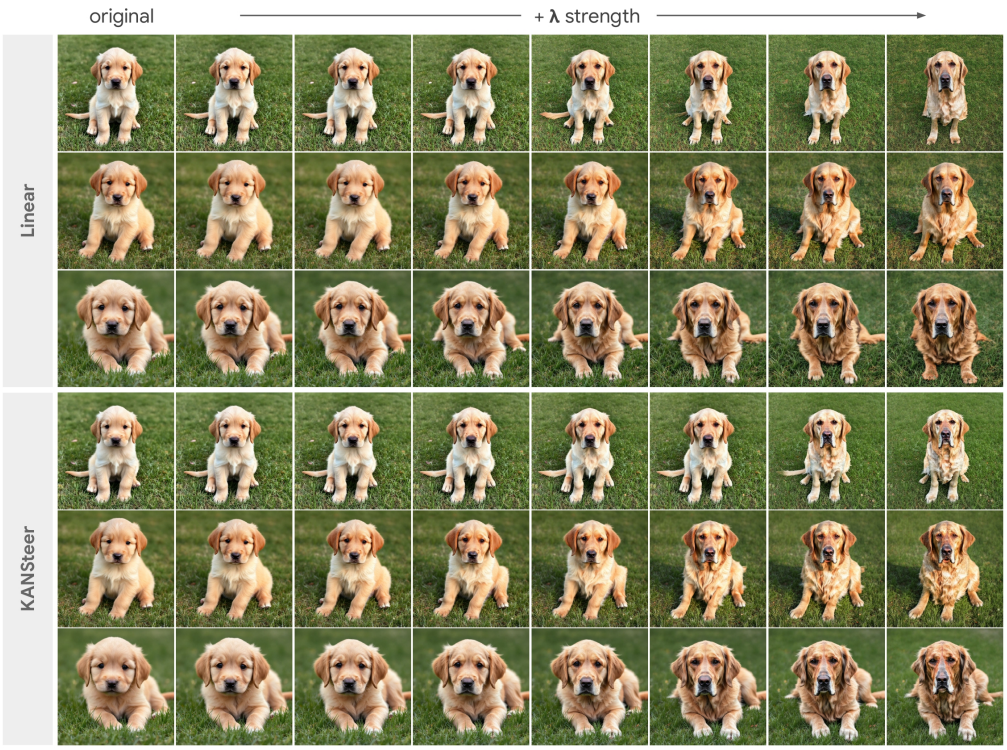

Figure 12: **Dog Aging (preteen → elderly):** Comparison the Linear and KANSteer steering on FLUX.1 Schnell.



Figure 13: **Beard Growth (clean shaven → long beard):** Comparison the Linear and KANSteer steering on FLUX.1 Schnell.



Figure 14: **Metamorphosis (caterpillar → spread butterfly):** Comparison the Linear and KANSteer steering on FLUX.1 Schnell.

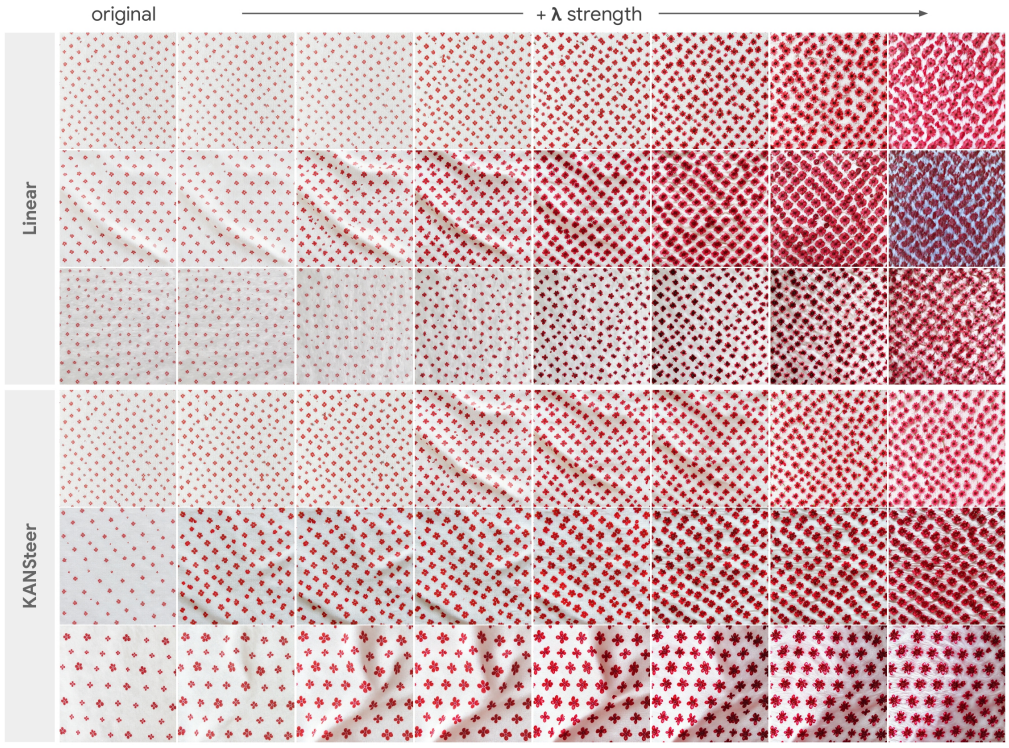

Figure 15: **Pattern Density (very sparse → very dense):** Comparison the Linear and KANSteer steering on FLUX.1 Schnell.



Figure 16: **Facial Emotions (sobbing → laughing):** Comparison the Linear and KANSteer steering on FLUX.1 Schnell.

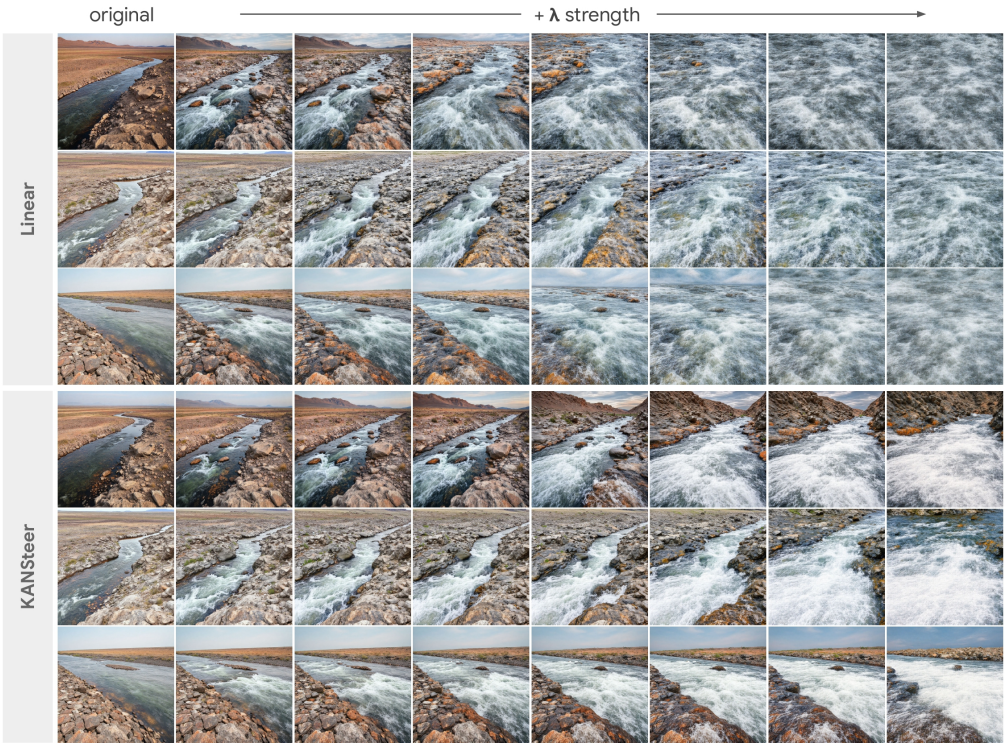

Figure 17: **River flow (still → raging):** Comparison the Linear and KANSteer steering on FLUX.1 Schnell.



Figure 18: **Seasons (spring → deep winter):** Comparison the Linear and KANSteer steering on FLUX.1 Schnell.

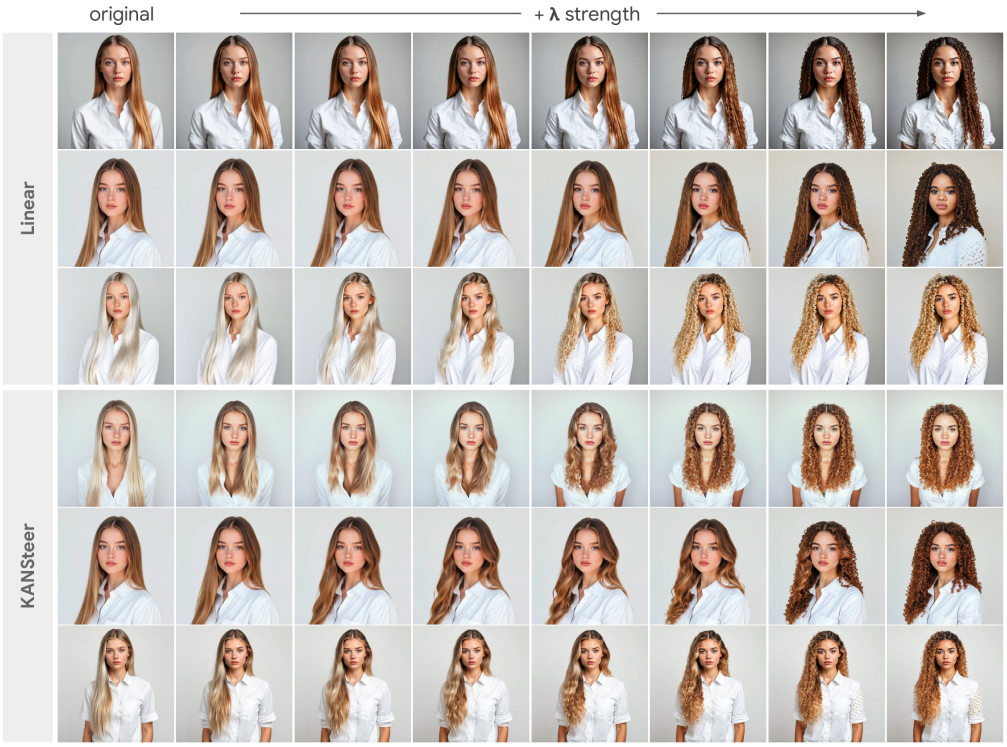

Figure 19: **Hair Curliness (straight → coily):** Comparison the Linear and KANSteer steering on FLUX.1 Schnell.

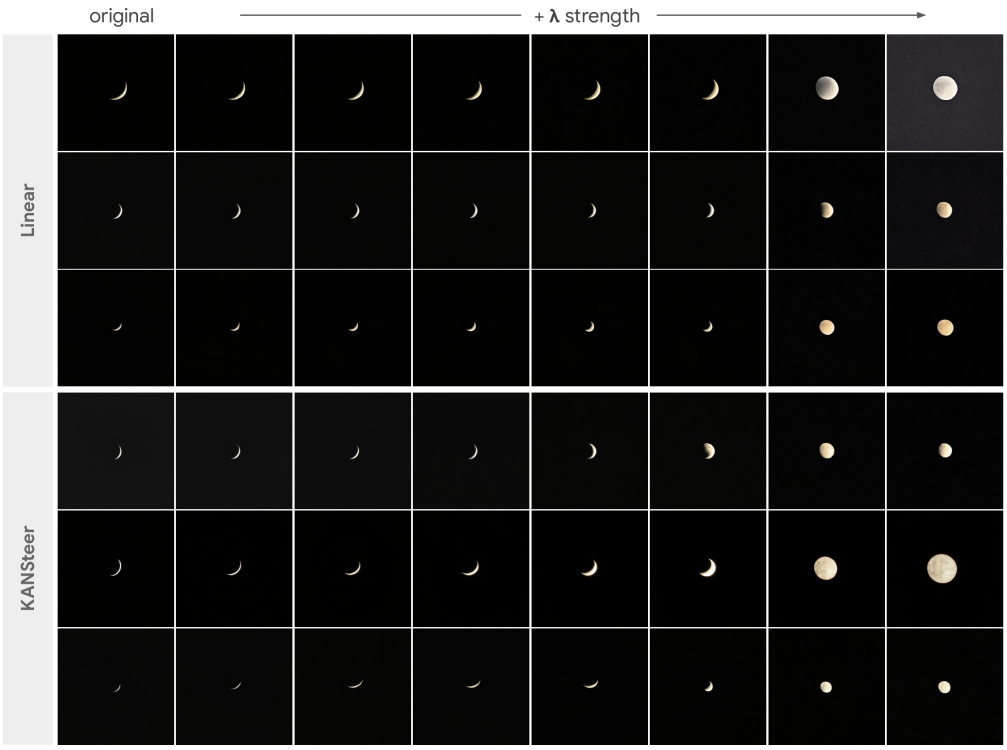

Figure 20: **Moon Phases (thin crescent → full):** Comparison the Linear and KANSteer steering on FLUX.1 Schnell.