ON COLOR ALIGNMENT IN VAE LATENT SPACES AND ITS APPLICATIONS

Anonymous authors
Paper under double-blind review

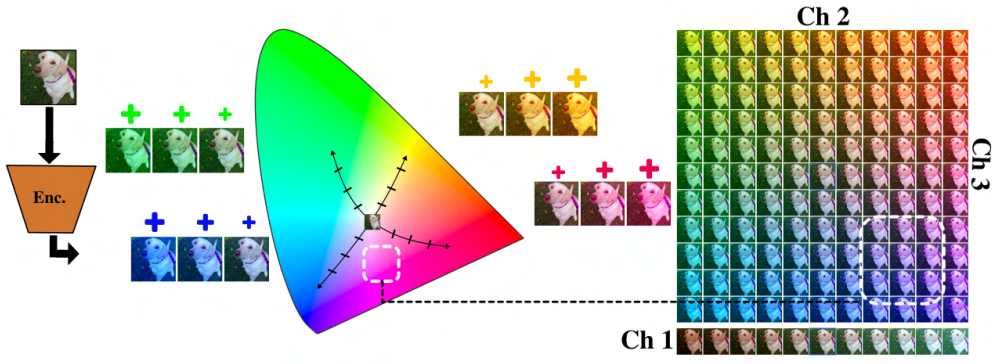


Figure 1: A color space hides in plain sight in the latent space of an autoencoder. Train a VAE with three latent channels on natural images, encode an image (left), add a constant offset δ to one channel, and decode: the image changes color while preserving its content. Sweeping two channels (right; ch2 across columns, ch3 across rows) spans an opponent-color plane and all intermediate mixtures, while the remaining channel (bottom row) controls brightness.

ABSTRACT

Variational autoencoders (VAEs) are a key part of modern text-to-image models, which generate images within their latent space. VAEs are known to disentangle the main factors of variation in the data, and color is known to be one of the most structured of these in natural images: decorrelating it yields one luminance axis and two opponent-color axes. Color should therefore be expected to emerge as a distinct factor in the VAE latent space. Yet how these latent spaces represent color remains largely unexplored. In this work, we show that the VAEs of text-to-image models share a color subspace aligned with brightness and opponent-colors. Through a linear approximation of the encoder and targeted latent steering, we find this subspace consistently across a broad range of VAEs, from SD1.5 to FLUX.2 and Z-Image. Building on this characterization, we propose three applications: *ColorTuning* , which achieves state-of-the-art in precise numerical color generation on the fine-grained CSS3/X11 system of GenColorBench, *saturation control*, to adjust the global chromatic intensity, and *color transfer*, to change the palette to match a reference.

1 INTRODUCTION

Natural images are far from random: they occupy a small, highly structured subset of the space of all possible pixel arrays, with strong spatial and spectral correlations (Simoncelli & Olshausen, 2001). Efficient coding theory holds that a visual system should exploit this redundancy by mapping its inputs to a compact, decorrelated representation (Barlow, 1961; Malo, 2020). For color, this principle gives a remarkably consistent answer: decorrelating the cone responses to natural scenes yields one achromatic luminance axis and two chromatic opponent axes, namely red–green and blue–yellow (Buchsbaum & Gottschalk, 1983; Ruderman et al., 1998). These axes closely match the cardinal

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.

directions of early human color vision (Derrington et al., 1984) and are mirrored in perceptual color spaces such as CIELAB. Luminance and opponent colors are therefore not an arbitrary convention but a property of natural images, one that any efficient encoder should be expected to recover.

In fact, the evidence suggests that artificial systems trained on natural images develop this color representation too. Unsupervised methods related to independent component analysis, applied to color image patches, learn separate achromatic and color-opponent features (Hoyer & Hyvärinen, 2000; Gutmann et al., 2014). Deep networks behave similarly: the first-layer of CNNs trained for classification display achromatic and color-opponent units (Krizhevsky et al., 2012; Flachot & Gegenfurtner, 2018), and CNNs trained for low-level vision tasks show human-like color axes and behavior, including susceptibility to color illusions (Gomez-Villa et al., 2020), and distinctive spatial bandwidth for color channels (Li et al., 2022). Finally, vision foundation models, trained on natural images at a far larger scale, may therefore retain this color structure, and recent evidence supports this expectation (Akbarinia, 2025; Wickramanayaka & Oizumi, 2025; Gomez-Villa et al., 2025).

Most modern text-to-image models are latent generators: rather than operating on pixels, they generate within the representation learned by a VAE and decode the result back to image space (Rombach et al., 2022). VAEs are known to organize this representation around the factors of variation in the data: they tend to disentangle generative attributes (Higgins et al., 2017; Burgess et al., 2018) and to align their latent axes with the principal directions of the data (Rolinek et al., 2019; Zietlow et al., 2021). Yet, although color is one of the most thoroughly studied and highly structured properties of natural images, how it is represented in these latent spaces has only recently begun to be explored (Arias et al., 2025; Pach et al., 2026). Meanwhile, text-to-image models still fail to bind colors to objects, render named colors faithfully, or follow numerical specifications such as HEX or RGB codes (Butt et al., 2026a). The color structure may already be present in the latent space; but the generators simply do not use it.

Recent studies have begun to probe color in these latent spaces with uniform images, recovering opponent hue axes in the Stable Diffusion VAE (Arias et al., 2025) and an HSL-like subspace in the FLUX VAE (Pach et al., 2026). Such inputs reveal where colors are mapped, but not how color is encoded alongside the spatial content of real images. We instead analyze the response to natural images and show that the VAEs of modern text-to-image models share a color subspace aligned with brightness and opponent-color directions (Fig. 1). On the encoder side, the singular vectors of a linear approximation of the encoder are achromatic and opponent-color patterns; on the decoder side, traversing individual latent channels shifts the color of the decoded image. We find this structure consistently in the VAEs of SD1.5, SDXL, SD3, FLUX, and FLUX.2, with 4 to 32 latent channels, and show that it is only partially disentangled from image structure. Building on it, we propose ColorTuning , which steers generation to follow numerical color specifications and achieves state-of-the-art accuracy on the CSS3/X11 system of GenColorBench (81.20% vs. 51.96%), and show that the same color basis also enables saturation control and color transfer.

In summary, our contributions are:

- We empirically show that the VAEs of modern text-to-image models share a color subspace aligned with brightness and nonlinear opponent-color axes, consistent across model families and latent sizes from 4 to 32 channels, and grounded on natural image statistics.
- We characterize this subspace through a linear approximation of the encoder and targeted latent steering, and quantify its partial entanglement with image structure.
- We propose a number of applications based on our characterization: *ColorTuning* achieves state-of-the-art in precise numerical color generation on the fine-grained CSS3/X11 system of GenColorBench, *saturation control* adjusts the global chromatic intensity, and *color transfer* changes the palette to match a reference.

2 Related Work

What VAEs disentangle. Unsupervised disentanglement is impossible without inductive biases on both model and data (Locatello et al., 2019), yet VAEs often disentangle in practice (Higgins et al., 2017; Burgess et al., 2018). Rolinek et al. (2019) traced this to the diagonal posterior, which aligns the latent axes with the local principal directions of the data, and Zietlow et al. (2021) showed that this yields disentanglement only when the variance structure of the data matches its generative

factors. This line of work has been evaluated almost exclusively on synthetic benchmarks. We test it on color, whose natural-image statistics decorrelate into one luminance and two opponent axes (Buchsbaum & Gottschalk, 1983; Ruderman et al., 1998).

Color in vision foundation models. The color representations of deep networks have been compared with human perception, with limited alignment (Wickramanayaka & Oizumi, 2025; Ehab et al., 2026). For text-to-image VAEs, Arias et al. (2025) recovered opponent hue axes in the Stable Diffusion VAE and Pach et al. (2026) an HSL-like subspace in the FLUX VAE, both by probing with uniformly colored patches. We instead analyze the response to natural images, across five VAEs with 4 to 32 latent channels.

Precise color control. Existing methods either act on the text conditioning, by learning color tokens (Butt et al., 2024) or grounding HEX codes and RGB values in a CIELAB-aligned ColorBook (NumColor; Butt et al., 2026b), or intervene at inference time, either by rewiring cross-attention (ColorWave; Laria et al., 2026) or by optimizing the latents (Shomer-Chai et al., 2026). Closest to ours, Pach et al. (2026) recolor objects through an HSL mapping interpolated from eight anchor colors, but this approch is only specifical for FLUX. ColorTuning instead recovers the color basis of any pretrained VAE and learns the displacement toward the target from measured CIELAB outcomes. An extended discussion is given in Appendix A.2.

2.1 Background

2.1.1 Variational autoencoders

Variational autoencoders (VAEs) (Kingma & Welling, 2014; Rezende et al., 2014) learn a probabilistic mapping between images $\mathbf{x} \in \mathbb{R}^{d_x}$ and a lower-dimensional latent representation $z \in \mathbb{R}^{d_z}$. An encoder E parameterizes an approximate posterior $q_\phi(z|\mathbf{x})$, typically a Gaussian with diagonal covariance, and a decoder D parameterizes the likelihood $p_\theta(\mathbf{x}|z)$. Both are trained jointly by maximizing the Evidence Lower Bound (ELBO), which in its β-weighted form (Higgins et al., 2017) is defined by :

$$\mathcal{L}_\beta = \mathbb{E}_{q_\phi(z|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|z)\right] - \beta\, D_{\mathrm{KL}}\left(q_\phi(z|\mathbf{x}) \,\|\, p(z)\right), \tag{1}$$

where the first term encourages faithful reconstruction and the second regularizes the posterior toward an isotropic Gaussian prior $p(z) = \mathcal{N}(\mathbf{0}, I)$ via the Kullback-Leibler divergence (D_{KL}). Setting $\beta > 1$ strengthens this regularization and is known to promote disentangled latent factors (Higgins et al., 2017; Burgess et al., 2018).

The VAEs of modern latent generative models operate in the opposite regime. Following Rombach et al. (2022), they are trained with $\beta \ll 1$ (on the order of 10^{-6}), together with perceptual and adversarial losses (Esser et al., 2021), which prioritizes reconstruction quality and keeps the latent space only mildly regularized. These VAEs are trained once and then kept frozen. The encoder compresses an image of size $H \times W \times 3$ into a spatial latent of size $\frac{H}{f} \times \frac{W}{f} \times C$, where f is the spatial downsampling factor and C is the number of latent channels, and the generator operates entirely in this latent space.

2.1.2 Latent Diffusion and Flow Matching

Latent generative models learn to transform Gaussian noise into VAE latents $z_0 = E(\mathbf{x})$ through an iterative sampling process. Two formulations dominate current text-to-image models. Diffusion models (Ho et al., 2020) define a forward process that progressively corrupts z_0 with Gaussian noise, $z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$, where $\bar{\alpha}_t$ decreases monotonically from 1 to 0 over time and train a network $\epsilon_\theta(z_t, t)$ to predict the added noise. Flow matching models (Lipman et al., 2023; Liu et al., 2023), adopted by SD3 and FLUX (Esser et al., 2024), instead connect data and noise through the linear path $z_t = (1-\sigma_t)\, z_0 + \sigma_t\, \epsilon$, where σ_t controls the noise magnitude, and train a network $v_\theta(z_t, t)$ to regress the velocity $\epsilon - z_0$ along it. Generation then reverses these processes, starting from pure noise and integrating toward a clean latent that is lated decoded by D into an image.

In both formulations, the clean latent can be estimated in closed form at any intermediate step t from the current network prediction, without completing the sampling trajectory (Song et al., 2021):

$$\hat{z}_0 = \frac{z_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(z_t, t)}{\sqrt{\bar{\alpha}_t}} \quad \text{(diffusion)}, \qquad \hat{z}_0 = z_t - \sigma_t\, v_\theta(z_t, t) \quad \text{(flow matching)}. \tag{2}$$

Decoding $\hat{z}_0$ gives a provides an intermediate estimate of the final image, which we use to measure the color of the target object during generation (Sec. 4.2).

3 Color in Autoencoder Latent Space

We study how the latent spaces of autoencoders represent color in two steps. We first show empirically that color is directly readable from the raw latent channels: traversing a single channel moves the decoded image along a consistent color direction, within a limited range before its spatial content also changes (Sec. 3.1). We then explain this behavior from the input side through a linear approximation of the encoder, whose singular vectors are opponent-colored patterns (Sec. 3.2).

3.1 Latent Channels Encode Color

In classical VAEs, individual latent coordinates often capture a single factor of variation of the data: traversing one coordinate while keeping the others fixed changes, for example, only the azimuth of a face (Higgins et al., 2017; Rolinek et al., 2019). We apply this same probe, a latent traversal, to the raw latent space of autoencoders, focusing on color. Since these latents are spatial, we refer to each latent dimension as a channel, and traverse channel c by adding a constant offset δ at every spatial position, $z_c \to z_c + \delta$, before decoding. As shown in Fig. 2, modifying δ progressively shifts the color of the decoded image along a consistent direction. We observe this across deterministic autoencoders (AEs), VAEs, and VQ-VAEs with 3, 4, 8, and 32 channels trained from scratch, as well as in the pretrained VAEs of SD1.5, SDXL, SD3, FLUX, and FLUX.2 (See the Fig. 2 and Appendix A.3). Color is therefore directly readable from the raw latent channels, without any learned probe.

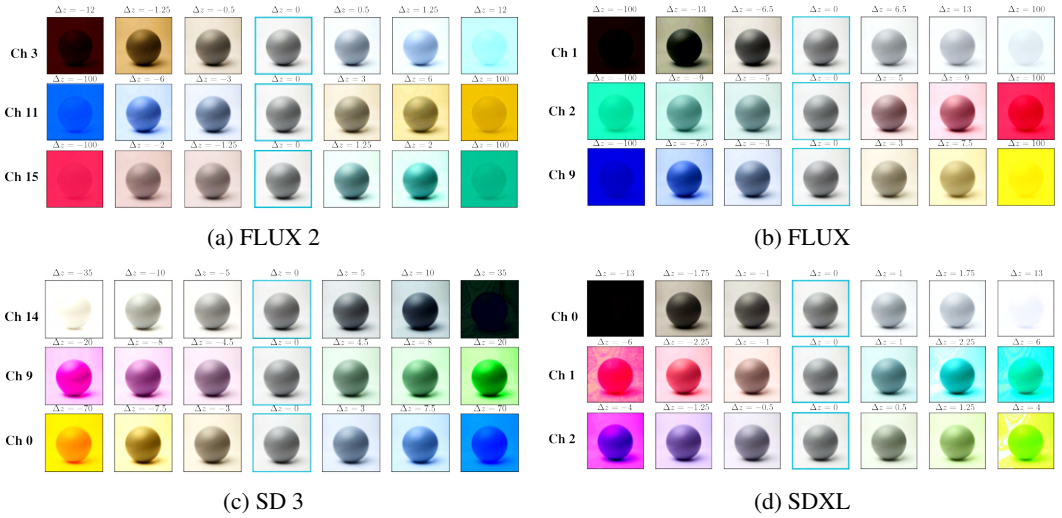
Color is, however, spatio-chromatically entangled: the same channels that carry color also carry spatial information. For small offsets, traversing the channel modifies only the color of the decoded image; but beyond a certain magnitude, it also begins to alter its spatial content. This behavior is visible in the edges and textures in the outermost columns of Fig. 2 (see Appendix A.3 for more examples). We quantify this in Fig. 3 as a function of δ. We measure color change with ΔE_{00} (CIEDE2000) relative to the unperturbed reconstruction, and spatial change with MS-SSIM computed on grayscale images using only its contrast and structure terms, so that the intended color and brightness shifts are not penalized. As $|\delta|$ increases, ΔE_{00} grows steadily, while MS-SSIM stays close to 1 for small offsets and then drops. Color can therefore be edited by moving along the latent channels, but only within a limited range before the edit starts to change the spatial content of the image.

In general, we find that individual channels do not uniquely map one-to-one onto specific colors: a single channel typically moves the image along a mixture of them, and several channels can move it in similar directions (e.g., two different channels may both turn the image green). A notable exception arises when there are only 3 latent channels. In this case, both the VAE and the AE align closely with the opponent color space: one channel controls brightness, and the other two control the opponent axes (see Appendix A.3). For the VAE, this is the behavior predicted by Rolinek et al. (2019). For the AE, it is not: its objective has no mechanism that favors a particular orientation (Rolinek et al., 2019). However, opponent color directions do appear in plain autoencoders (Gomez-Villa et al., 2020; Li et al., 2022), and, as shown below, they consistently emerge in modern autoencoders.

In the general case, where color is spread across correlated channels, the color subspace can still be recovered with a PCA decomposition of the channels' color responses, which yields one luminance and two opponent-color directions. This is the mechanism we exploit in our applications (Sec. 4).

3.2 Linear Approximation of the VAE

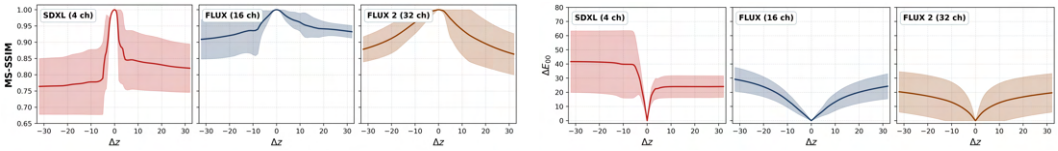
The traversals in section 3.1 show that color is readable from the latent channels, but not why. To answer this we turn to the input side and ask which image patterns the encoder is most sensitive



(a) FLUX 2 (b) FLUX

(c) SD 3 (d) SDXL

Figure 2: Image modifications through traverse channel variations across generative models: (a) Flux, (b) Flux 2, (c) Stable Diffusion 1.5, (d) SDXL, and (e) Stable Diffusion 3.



(a) Texture distortion (b) Color variation

Figure 3: Texture-color variations through traverse latent channel modifications in different VAEs.

to. Because a VAE is trained to reconstruct its input, its response stays close to the identity and is well described to first order, which lets us replace the encoder by a matrix whose singular vectors are precisely those patterns, ordered by how strongly they drive the latent. If the color structure observed above originates in the encoder, it should appear here as achromatic and opponent-color patterns among the leading singular vectors.

Let $X \in \mathbb{R}^{d_x \times N}$ represent a matrix containing N input images. Let $Z \in \mathbb{R}^{d_z \times N}$ represent their corresponding low-dimensional latent representations, obtained via the VAE's encoder, $Z = E(X)$. Subsequently, let $\hat{X} = D(Z)$ represent the reconstructed output images generated by the VAE's decoder. The application of the model (encoder + decoder) is denoted here as: $\hat{\mathbf{x}} = m(\mathbf{x})$.

Following Gomez-Villa et al. (2020) and Li et al. (2022), we analyze the behavior of the network through the linear approximation of its response. Considering any image $\mathbf{x}$ as a deviation from the average image (flat gray image, $\mathbf{0}$), the first-order approximation of the autoencoder is:

$$\hat{\mathbf{x}} = m(\mathbf{x}) \approx m(\mathbf{0}) + \nabla_{\mathbf{x}}\, m(\mathbf{0}) \cdot \mathbf{x} = M_{vae} \cdot \mathbf{x} \tag{3}$$

where the response to the average image is re-calibrated as zero, and $M_{vae} \in \mathbb{R}^{d_x \times d_x}$ is thus the Jacobian of the network response w.r.t. the input at 0. For VAE architectures, this assumption is justified by the ELBO objective, where the expected log-likelihood reconstruction term, $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$, forces the latent space to preserve sufficient information for data recovery. Furthermore, M_{vae} can be decomposed as the product of the decoder and encoder matrices, such that $M_{vae} = M_{dec} M_{enc}$, where $M_{dec} \in \mathbb{R}^{d_x \times d_z}$ and $M_{enc} \in \mathbb{R}^{d_z \times d_x}$. We select the simplest method to compute M_{enc} from the three approaches considered in Li et al. (2022). In particular, here $M_{enc} = Z \cdot X^{\dagger}$, and $M_{dec} = \hat{X} \cdot Z^{\dagger}$, where † denotes the pseudoinverse.

To analyze the structure of the latent space, we perform Singular Value Decomposition (SVD) on the encoder matrix, $M_{enc} = U \Sigma V^T$.

In Fig. 4, we visualize the singular vectors derived via SVD for the VAEs of text-to-image models. In addition, we plot the chromatic coordinates corresponding to the components with the largest singular values and highest chromatic variance on the CIE xy chromaticity diagram. Across all eval-

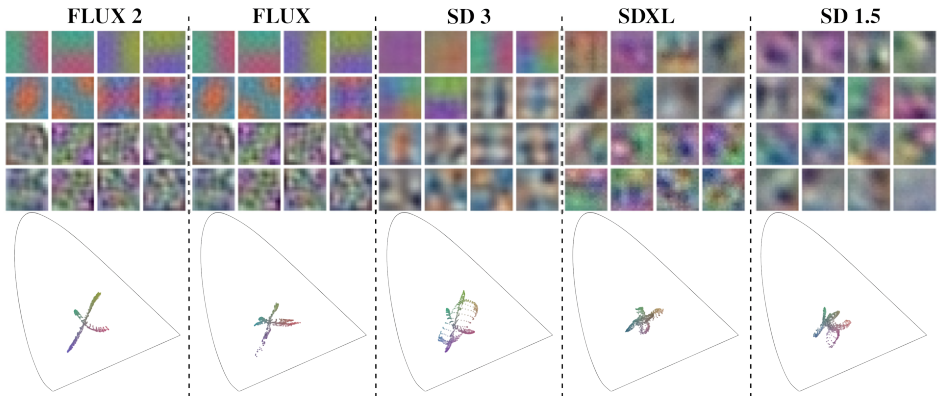


Figure 4: The first singular vectors of the linear approximation of VAEs and their corresponding CIE chromaticity directions across generative models: Flux.dev 2, Flux.dev 1, (c) SD 1.5, (d) SDXL, and SD 3. Each subplot visualizes the singular vectors ordered by their chromatic variance alongside their color content in the CIE chromatic diagram. We used 10^6 16×16 patches from *Imagenette*. Appendix A.4 shows that this kind of vectors and opponent colors also emerge in other databases.

uated models, two principal opponent chromatic directions consistently emerge. This demonstrates that the latent spaces of these VAEs align with opponent color channels.

Faithfulness of the linear approximation. This reading of the encoder is only as informative as the linear approximation behind it. Appendix A.5 shows that linear approximations of E and D are good descriptions of their behavior because (i) they explain a large fraction of the energy of the signal generated by the (nonlinear) autoencoder, and (ii) the errors wrt the original signal are small.

4 Applications

Sec. 3 shows that VAE latents carry a color structure inherited from natural image statistics, but spread across correlated channels and only partially disentangled from spatial content. We therefore recover an explicit color basis for each VAE, computed once (Sec. 4.1), and use it for three applications in T2I generation, handling the residual entanglement at inference time: numerical color precision (Sec. 4.2), saturation control, and color transfer (Sec. 4.3).

4.1 Characterizing the Color Basis of a VAE

Our goal is to find a small set of disentangled directions in the latent space. Specifically, we seek directions that move the color of an image along the luminance axis or the two opponent color axes independently. We extract these principal color directions through the following steps.

Step 1: Controlled perturbation. We select seven reference objects spanning the primary axes of the CIELAB color space. For each object, we perturb one latent channel at a time. We add a scalar perturbation $z_{c,m} \to z_c + m$ inside the region of interest for a set of magnitudes m. We repeat this process for every channel c in the latent space and measure how much the perceived color of the object shifts in the CIELAB space $(\Delta L^*_{c,m}, \Delta a^*_{c,m}, \Delta b^*_{c,m})$.

Step 2: Measuring sensitivity. For each channel, we compute the slope of color change per unit of perturbation:

$$s_{c,m,L} = \frac{\Delta L^*_{c,m}}{\Delta z_c}, \qquad s_{c,m,a} = \frac{\Delta a^*_{c,m}}{\Delta z_c}, \qquad s_{c,m,b} = \frac{\Delta b^*_{c,m}}{\Delta z_c} \tag{4}$$

We then stack all values for different m and c as rows to obtain a global sensitivity matrix $S \in \mathbb{R}^{Cm \times 3}$.

Step 3: Isolating principal color directions. We apply SVD directly to $S = U\Sigma V^\top$. The resulting three columns of $U \in \mathbb{R}^{(C \times m) \times 3}$ provide a basis that isolates the latent color shifts.

To use the principal color directions discovered for generation, we localize each perturbation to the target object using spatial masks from SAM 3 (Carion et al., 2026). To prevent abrupt deviations

Table 1: Numerical color accuracy (%) on GenColorBench, on (a) the mini subset and (b) the full benchmark. The number of latent channels is shown in parentheses. **Bold**: best result for each backbone.

(a) GenColorBench-mini

Model	ISCC-L3	CSS3/X11	Overall
SDXL (4)	1.84	7.37	5.53
+ ColorWave	11.15	28.05	22.42
+ ColorTuning	**30.78**	**63.97**	**52.90**
Z-Image (16)	16.92	12.45	13.94
+ ColorTuning	**44.91**	**81.11**	**59.22**
FLUX.2 (32)	**39.29**	42.67	41.54
+ ColorTuning	38.91	**81.90**	**64.57**

(b) GenColorBench (full)

Model	ISCC-L3	CSS3/X11	Overall
SD3 (16)	5.92	4.63	5.06
+ NumColor	**39.46**	47.17	**44.32**
+ ColorTuning	12.63	**54.74**	39.02
SD3.5M (16)	7.53	5.78	6.36
+ NumColor	**43.35**	46.87	45.57
+ ColorTuning	40.65	**75.79**	**62.65**
FLUX (16)	7.74	5.86	6.49
+ NumColor	**48.02**	51.96	50.50
+ ColorTuning	43.15	**81.20**	**64.13**

from the natural generation path, we inject these perturbations only during the final generation steps, linearly decaying the steering weight to zero.

4.2 Numerical Color Precision

Numerical colour precision refers to the ability to generate an object with the exact RGB value or HEX code specified in the text prompt. State-of-the-art T2I models still struggle with this task, exhibiting a range of failure cases (see Fig.15 and Butt et al. (2026a;b)). In this subsection, we present ColorTuning , which exploits the discovered directions to steer the generation towards the exact colour specified in the user prompt.

4.2.1 Steering the Generator

Our approach is based on two key properties of the discovered color directions. First, diffusion and flow-matching models resolve low spatial frequencies first Biroli et al. (2024): the coarse color of a region is settled well before sampling ends, so decoding the predicted clean latent $\hat{z}_0$ (Eq. 2 in Appendix 2.1) at an intermediate step already reveals the color the object will have in the final image. Second, the characterization of Sec. 4.1 gives the directions along which that color can be moved. Measuring the current color and applying the corresponding displacement over the remaining steps is therefore enough to reach the target (see Fig. 16 in Appendix A.7).

Training. The basis of Sec. 4.1 fixes the directions, but not the magnitude: how far a given displacement moves the object in CIELAB depends on the color it starts from, since the decoder is nonlinear. We therefore learn this relation empirically. For a set of prompts containing a color name, we generate one image without intervention and, for each displacement $m \in \mathcal{M}$ along the basis, a second image with the perturbation applied to the target object. Measuring the color of the object in both images gives a triplet: the color before the perturbation, the color it reached, and the displacement that produced it. On this data we train a small MLP that takes an initial and a target color, $C_i = (L_i, a_i, b_i)$ and $C_t = (L_t, a_t, b_t)$, and predicts the displacement (m_1, m_2, m_3) along the basis that carries C_i to C_t (Alg. 1).

Inference. The prompt still has to place the object in the scene, so we never pass the numerical code to the text encoder. We replace it with a proxy color name: the nearest entry in ISCC-NBS Level 2 to the requested value, measured in CIELAB. A prompt such as *"a car of color #7B3F00"* becomes *"a brown car"*. The generator renders names far more reliably than codes, so the trajectory starts in the neighborhood of the target color and the steering only has to cover the residual difference. At a gate step s, we predict the clean latent $\hat{z}_0$ (Eq. 2), decode it, locate the object with SAM 3 and measure its mean color in CIELAB. The MLP maps this color and the target to a displacement along the color basis, which we add to the latent inside the object mask at each remaining step, with a weight that decays linearly to zero so that the final steps re-integrate the perturbation into the image structure (Alg. 2; hyperparameters in Appendix A.7).

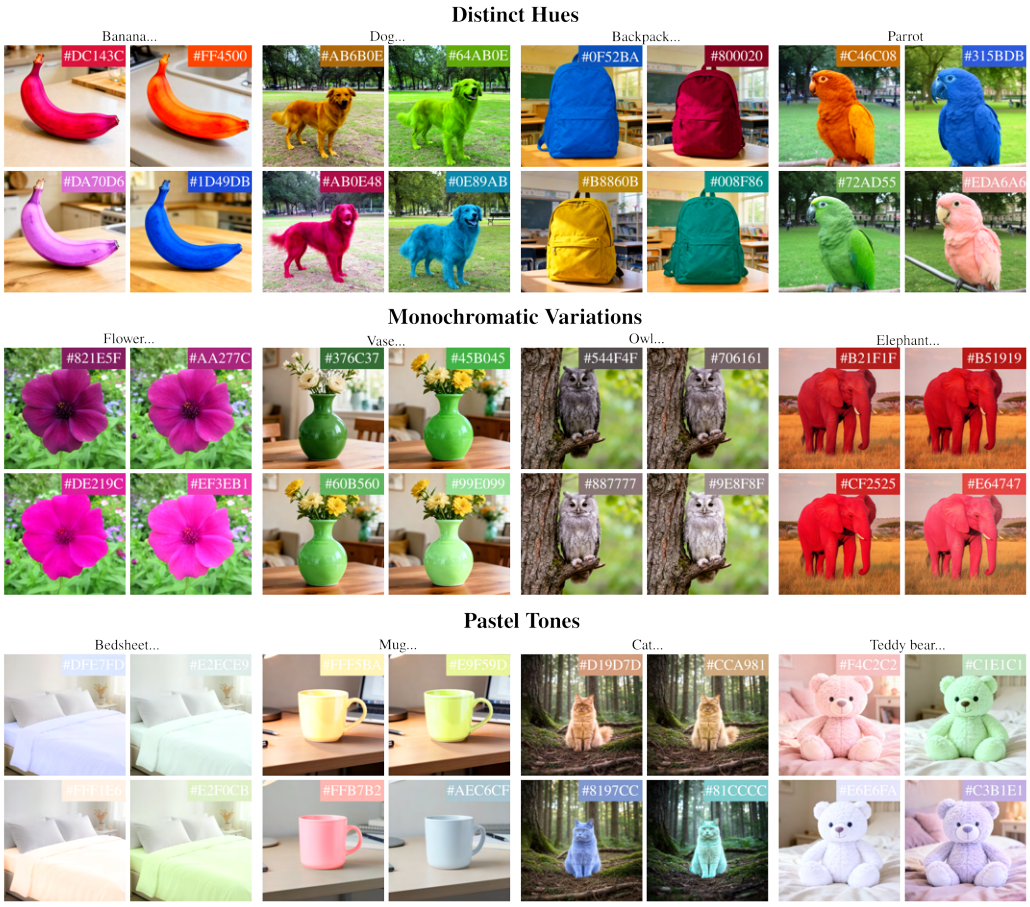


Figure 5: Qualitative results of ColorTuning . The generated examples demonstrate precise color control across three scenarios: applying distinct hues (top), generating subtle monochromatic variations within the same color family (middle), and pastel tones (bottom).

4.2.2 Results of ColorTuning

Setup. We evaluate ColorTuning on GenColorBench (Butt et al., 2026a), a benchmark for precise color generation in which prompts specify object colors as HEX codes and RGB values. We report accuracy on its two color systems, ISCC-L3 and CSS3/X11, under both formats. We apply ColorTuning to six generators covering the three latent sizes analyzed in Sec. 3: SDXL (4 channels), SD3, SD3.5M, FLUX and Z-Image (16), and FLUX.2 (32). We compare against the vanilla models and against two color-control methods, each on the backbones for which it is available: ColorWave on SDXL and NumColor on SD3, SD3.5M and FLUX.

Quantitative results. On the full benchmark (Table 1b), vanilla models stay below 8% on both systems. On CSS3/X11, NumColor reaches 47–52%, while ColorTuning achieves the best accuracy on every backbone, up to 81.20% on FLUX. This is remarkable because, while NumColor remains ahead on ISCC-L3, the CSS3/X11 on which ColorTuning dominates is the one on which every model evaluated in GenColorBench performs worst (Butt et al., 2026a). On the mini subset (Table 1a), ColorTuning brings FLUX.2 and Z-Image to about 81% on CSS3/X11 and more than doubles ColorWave on SDXL (63.97% vs. 28.05%). A 2AFC user study with 22 observers with normal color vision confirms these gains perceptually: ColorTuning is preferred over both NumColor (FLUX) and ColorWave (SDXL), with z-scores (computed using Thurstone's model) of 2.62 ± 0.30 and 1.67 ± 0.30, respectively (95% CI; 0 indicates no preference). Details are in Appendix A.9.

Qualitative results. Fig. 5 presents qualitative results obtained by applying ColorTuning to FLUX.2. The first two rows illustrate the method's behavior across the full color spectrum, showing that the same concept can be rendered in substantially different hues while preserving its structure and appearance. The third and fourth rows evaluate fine-grained numerical color control, where

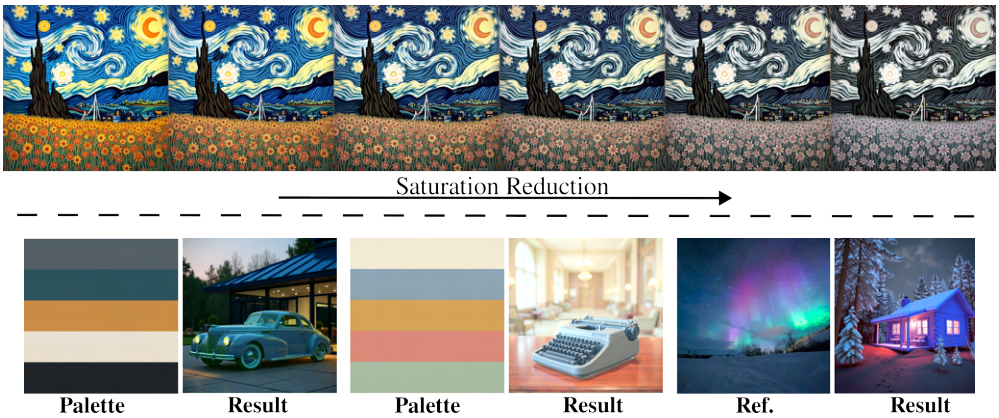


Figure 6: Further color control with the same color basis. **Top:** saturation reduction. From left to right, we intervene during generation to reduce the saturation of the final image. **Bottom:** color transfer, three examples. In each pair, the left image is the color reference and the right image is the generated result, whose colors are steered to match it.

target colors differ only subtly. The final two rows focus on pastel colors, which combine high luminance with low saturation and are particularly challenging for contemporary generative models. Even in this demanding setting, ColorTuning faithfully reproduces the requested colors.

4.3 Further Color Applications During Generation

The same characterization naturally extends to other forms of color control at generation time, without requiring any modification to the procedure introduced in Sec. 4.1. Fig. 6 (top) illustrates saturation control: by traversing the corresponding latent directions, our method can continuously modulate the saturation of the generated image (in the figure from 100% to 50% at 10% intervals). Such saturation-aware generation could facilitate the synthesis of images within narrower color gamuts, potentially reducing the need for post-processing gamut mapping techniques (Morovič, 2008).

Fig. 6 (bottom) shows a second application: color transfer during generation. In this setting, the input consists of both a text prompt and a reference image whose color distribution should be reproduced. Our method leverages the discovered latent directions to match the color distribution of the reference image while simultaneously generating the content specified by the prompt. Full details of both applications are given in Appendix A.8.

5 Conclusions

We have shown that color structure is already present in the raw latents of autoencoders trained on natural images: in AEs, VAEs, and VQ-VAEs alike, traversing a single latent channel shifts the color of the decoded image. This observation can be explained by a linear approximation of the encoder, whose leading singular vectors correspond to the achromatic and opponent-color patterns predicted by natural image statistics. Recovering these axes as an explicit basis turns the observation into a tool for color control during generation. Our main use of it is ColorTuning , which achieves state-of-the-art accuracy in numerical color generation on the fine-grained CSS3/X11 system of GenColorBench; saturation control and color transfer further show, qualitatively, that the same basis supports other forms of color control.

Limitations. Existing methods for encouraging precise color synthesis in generative models typically rely on learning new concepts that enforce specific color relationships. In contrast, our approach leverages the inherent color alignment already present in the VAE latent space to dynamically guide the generation process. However, this strategy remains sensitive to large steering magnitudes. As shown in our analysis, the extracted latent color channels are not fully disentangled from spatial and structural features. Consequently, when the MLP predicts a large displacement from the source color toward the target one, the resulting latent perturbation affects not only color but also image content, leading to structural distortions in the final output.

REFERENCES

Arash Akbarinia. Exploring the categorical nature of colour perception: Insights from artificial networks. *Neural Networks*, 181:106758, 2025.

Guillem Arias, Ariadna Solà, Martí Armengod, and Maria Vanrell. Color encoding in latent space of stable diffusion models. In *Color and Imaging Conference Final Program*, 2025.

Horace B. Barlow. Possible principles underlying the transformation of sensory messages. In *Sensory Communication*, pp. 217–234. MIT Press, 1961.

Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, 2024. doi: 10.1038/s41467-024-54281-3.

Gershon Buchsbaum and Allan Gottschalk. Trichromacy, opponent colours coding and optimum colour information transmission in the retina. *Proceedings of the royal society of London. Series B, Biological sciences*, pp. 89–113, 1983.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Muhammad Atif Butt, Kai Wang, Javier Vazquez-Corral, and Joost Van de Weijer. Colorpeel: Color prompt learning with diffusion models via color and shape disentanglement. In *European Conference on Computer Vision*, pp. 456–472. Springer, 2024.

Muhammad Atif Butt, Alexandra Gomez-Villa, Tao Wu, Javier Vazquez-Corral, Joost Van De Weijer, and Kai Wang. Gencolorbench: A color evaluation benchmark for text-to-image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36638–36648, 2026a.

Muhammad Atif Butt, Diego Hernández, Alexandra Gomez-Villa, Kai Wang, Javier Vazquez-Corral, and Joost Van De Weijer. Numcolor: Precise numeric color control in text-to-image generation. In *European Conference on Computer Vision (ECCV)*, pp. 149–166, 2026b.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, RISHI HAZRA, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts. In *International Conference on Learning Representations (ICLR)*, pp. 138846–138923, 2026.

Andrew M. Derrington, John Krauskopf, and Peter Lennie. Chromatic mechanisms in lateral geniculate nucleus of macaque. *The Journal of Physiology*, 357(1):241–265, 1984.

Engy Ehab, Pablo Hernández-Cámara, Nahla Belal, Jesús Malo, Javier Vazquez-Corral, and Alexandra Gomez-Villa. Do vision encoders exhibit human-like color thresholds? *arXiv preprint arXiv:2607.16540*, 2026.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12873–12883, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

Alban Flachot and Karl R. Gegenfurtner. Processing of chromatic information in a deep convolutional neural network. *Journal of the Optical Society of America A*, 35(4):B334–B346, 2018.

Alex Gomez-Villa, Adrian Martín, Javier Vazquez-Corral, Marcelo Bertalmío, and Jesús Malo. Color illusions also deceive cnns for low-level vision tasks: Analysis and implications. *Vision Research*, 176:156–174, 2020.

Alexandra Gomez-Villa, Kai Wang, C Alejandro Parraga, Bartłomiej Twardowski, Jesus Malo, Javier Vazquez-Corral, and Joost van den Weijer. The art of deception: Color visual illusions and diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18642–18652, 2025.

Michael U. Gutmann, Valero Laparra, Aapo Hyvärinen, and Jesús Malo. Spatio-chromatic adaptation via higher-order canonical correlation analysis of natural images. *PLoS ONE*, 9(2):e86481, 2014. doi: 10.1371/journal.pone.0086481.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 6840–6851, 2020.

Patrik O. Hoyer and Aapo Hyvärinen. Independent component analysis applied to feature extraction from colour and stereo images. *Network: Computation in Neural Systems*, 11(3):191–210, 2000.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvärinen. Variational autoencoders and nonlinear ICA: A unifying framework. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 2207–2217, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Héctor Laria, Alexandra Gomez-Villa, Jiang Qin, Muhammad Atif Butt, Bogdan Raducanu, Javier Vazquez-Corral, Joost Van De Weijer, and Kai Wang. Leveraging semantic attribute binding for free-lunch color control in diffusion models. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7689–7698, 2026.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. Technical report, Stanford University, 2015. URL `http://stanford.edu`. CS231n Project Report.

Qiang Li, Alex Gomez-Villa, Marcelo Bertalmío, and Jesús Malo. Contrast sensitivity functions in autoencoders. *Journal of Vision*, 22(6):8–8, 05 2022. doi: 10.1167/jov.22.6.8.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning (ICML)*, pp. 4114–4124, 2019.

Jesús Malo. Spatio-chromatic information available from different neural layers via gaussianization. *J. Math. Neurosci.*, 10:18, 2020. doi: 10.1186/s13408-020-00095-8.

Ethan D Montag. Empirical formula for creating error bars for the method of paired comparison. *Journal of Electronic Imaging*, 15(1):010502–010502, 2006.

Ján Morovič. *Color Gamut Mapping*. The Wiley-IS&T Series in Imaging Science and Technology. John Wiley & Sons, Chichester, UK, 2008. ISBN 978-0-470-03032-5.

Mateusz Pach, Jessica Bader, Quentin Bouniot, Serge Belongie, and Zeynep Akata. The latent color subspace: Emergent order in high-dimensional chaos. In *International Conference on Machine Learning (ICML)*, 2026.

Patrik Reizinger, Luigi Gresele, Jack Brady, Julius Von Kügelgen, Dominik Zietlow, Bernhard Schölkopf, Georg Martius, Wieland Brendel, and Michel Besserve. Embrace the gap: Vaes perform independent mechanism analysis. In *Advances in Neural Information Processing Systems (NeuriPs)*, pp. 12040–12057, 2022.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, pp. 1278–1286, 2014.

Michal Rolinek, Dominik Zietlow, and Georg Martius. Variational autoencoders pursue pca directions (by accident). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12398–12407, 2019.

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)*, pp. 10674–10685, 2022.

Daniel L. Ruderman, Thomas W. Cronin, and Chuan-Chin Chiao. Statistics of cone responses to natural images: implications for visual coding. *Journal of the Optical Society of America A*, 15 (8):2036–2045, 1998.

Shay Shomer-Chai, Wenxuan Peng, Bharath Hariharan, and Hadar Averbuch-Elor. Color bind: Exploring color perception in text-to-image models. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1916–1925. IEEE, 2026.

Eero P. Simoncelli and Bruno A. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1216, 2001.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

Nipun Ravindu Wickramanayaka and Masafumi Oizumi. Systematic comparison of color representations between humans and deep neural networks: towards predicting human color perception in a vast color space. *bioRxiv*, 2025. doi: 10.64898/2025.12.10.693611.

Xiaolin Wu. Efficient statistical computations for optimal color quantization. In James Arvo (ed.), *Graphics Gems II*, pp. 126–133. Academic Press, 1991.

Dominik Zietlow, Michal Rolinek, and Georg Martius. Demystifying inductive biases for (beta-)vae based architectures. In *International Conference on Machine Learning (ICML)*, pp. 12945–12954, 2021.

A Appendices

A.1 LLM usage statement

LLMs were used for grammar correction and to rewrite and refine parts of the manuscript. They also assisted with code development.

A.2 Related Work

A.2.1 Representation Learning and Disentanglement in VAEs

Whether a VAE recovers the true factors of variation in its data is a question of identifiability. Locatello et al. (2019) showed that unsupervised disentanglement is impossible without inductive biases on both model and data, and identifiability has since been proven only under additional assumptions such as auxiliary variables (Khemakhem et al., 2020). Yet VAEs often disentangle in practice (Higgins et al., 2017; Burgess et al., 2018). Rolinek et al. (2019) traced this to the diagonal posterior, which makes the decoder Jacobian locally orthogonal and aligns the latent axes with the local principal directions of the data. Relatedly, Reizinger et al. (2022) connected the ELBO to independent mechanism analysis in the near-deterministic regime. Zietlow et al. (2021) further showed that this bias yields disentanglement only when the variance structure of the dataset matches its generative factors. What a VAE disentangles is therefore dictated by the principal directions of its training data.

Color is a natural test of this view: it is one of the most structured properties of natural images, whose statistics decorrelate into one luminance and two opponent-color axes (Buchsbaum & Gottschalk, 1983; Ruderman et al., 1998). Yet this line of work has been evaluated almost exclusively on synthetic benchmarks, where color appears at most as a discrete hue label, and has not examined how VAEs trained on natural images represent it. Only recently have studies begun to probe color in the latent spaces of text-to-image VAEs (Arias et al., 2025; Pach et al., 2026), and none relates the structure they find to the statistics of the training data. We instead analyze the encoder's response to natural images and show that, across model families and latent sizes, these latents share a color subspace aligned with the luminance and opponent axes that natural image statistics would predict.

A.2.2 Color Representations in Vision Foundation Models

Recent work has begun to characterize how color is encoded in the representations of vision foundation models. Wickramanayaka & Oizumi (2025) compared the internal color representations of 16 deep networks with human similarity judgments over 93 colors, and found that only CLIP models reached fine-grained structural agreement with human perception. Ehab et al. (2026) probed more than 50 vision encoders, including the SD3 VAE, and compared their chromatic discrimination thresholds with human discrimination ellipses, finding weak alignment across all model families. For generative models, Arias et al. (2025) encoded uniformly colored patches with the Stable Diffusion VAE and applied PCA to their spatially averaged latents, recovering an intensity axis and two opponent hue axes. When probing grayscale synthetic patterns separately, they found that one of these chromatic channels also carries shape information. More recently, Pach et al. (2026) identified an HSL-like color subspace in the VAEs of FLUX, also by probing them with uniformly colored images, and used it to steer the color of generated objects. These studies characterize color through flat stimuli, which reveal where colors are mapped but not how color is encoded alongside the spatial content of real images. Moreover, each examines only one or two VAEs, leaving open whether the structure they find is specific to a particular model or shared across text-to-image VAEs with different architectures and latent sizes.

A.2.3 Precise Color Control in Text-based Image Models

Methods for precise color control in text-to-image models intervene at two points of the generative pipeline. A first family acts on the text conditioning. ColorPeel (Butt et al., 2024) learns color prompt tokens from synthetic stimuli that disentangle color from shape, but covers only a few colors per training session and is only able to interpolate between nearby colors. Targeting numerical specifications, NumColor (Butt et al., 2026b) addresses the fragmentation of hex and RGB codes into meaningless subword tokens by aggregating them into a single unit and grounding it with a ColorBook of learnable embeddings aligned with CIELAB.

A second family intervenes at inference without retraining the backbone, either by rewiring the cross-attention of IP-Adapter to bind a color descriptor to a patch of the target color (Laria et al., 2026), or by optimizing the latents with attention and color losses (Shomer-Chai et al., 2026). Closest to our work, Pach et al. (2026) recolor objects by shifting latents within the color subspace discussed above, through a mapping to HSL interpolated from eight anchor colors. Their interpretation is derived for FLUX alone, from flat color patches. We instead build on a general property: the latents of VAEs trained on natural images align with luminance and opponent-color axes, a structure inherited from natural image statistics (Sec. 3). This lets us design a characterization procedure that recovers these directions in any pretrained VAE, along which ColorTuning learns the displacement toward the target from measured CIELAB outcomes rather than from a fixed geometric mapping.

A.3 Latent Channel Shifts

A.3.1 Custom Baselines

We trained three autoencoder families from scratch on Tiny ImageNet-200 (Le & Yang, 2015) at its native 64×64 resolution: deterministic autoencoders (**AE**), continuous variational autoencoders (**KL-VAE**), and vector-quantized autoencoders (**VQ-VAE**). All models share a unified convolutional backbone (Rombach et al., 2022) with a spatial downsampling factor of $f = 4$ and were evaluated across bottleneck widths $C \in \{3, 4, 8, 16\}$. Training was performed for 40 epochs using Adam ($\beta_1 = 0.5, \beta_2 = 0.9$) with a learning rate of 4.5×10^{-6}.

To examine how chromatic information is encoded across the latent space, we performed latent shift interventions on encoded grayscale test images. The resulting channel-wise response grids are shown in Figure 9 for the AE, Figure 10 for KL-VAE, and Figure 11 for VQ-VAE.

A.3.2 Latent Diffusion Models

Building upon our local baselines, we extend our latent perturbation analysis to the pretrained autoencoders commonly utilized in LDMs (Rombach et al., 2022). Specifically, we isolate and shift individual latent channels across both KL-VAE (Figure 10) and VQ-VAE (Figure 11) architectures.

A.3.3 Foundation Models

Finally, we investigate the latent space architectures of foundation models. We apply our latent perturbation analysis to the autoencoders of SD1.5, SDXL, SD3, FLUX, and FLUX2, as illustrated in Figure 12 and Figure 13.

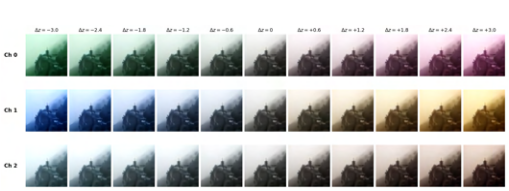
A.4 Similar color subspaces emerge in different datasets.

Fig. 14 shows the singular vectors and color content of the latent space of M_{enc} obtained from a set of images with substantially different resolutions: 10^5 images of size 64×64. In this case, for a cleaner visualization, a standard variance minimization algorithm (Wu, 1991) has been applied to select the colors shown. Qualitative similarity of these results with those of Fig. 4 suggests that color opponent subspaces emerge consistently for different sets of natural images and resolutions.

A.5 Faithfulness of the linear approximations.

In this Appendix, we quantify the validity of the linear approximation of VAE's encoder and decoder according to the methods proposed by Gomez-Villa et al. (2020) for regular AEs. In particular, here we compute three metrics for each VAE: *Fract. Lin. Resp.* is the fraction of the energy of the network response explained by the linear approximation. It is computed from the RMSE between the linear and the actual (nonlinear) responses, given in relative units with regard to the square root of the energy of the actual response. *Error NonLin.* and *Error Linear* are the corresponding reconstruction errors of the VAE and of its linear approximation, each expressed as the fraction of the energy of the original image.

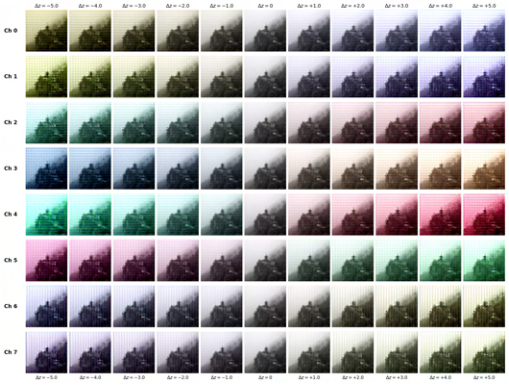
Table 2 shows such metrics and the average across all VAEs. Results show that on average, the linear approximations explain a large fraction (about 65%) of the nonlinear behavior. Moreover, the



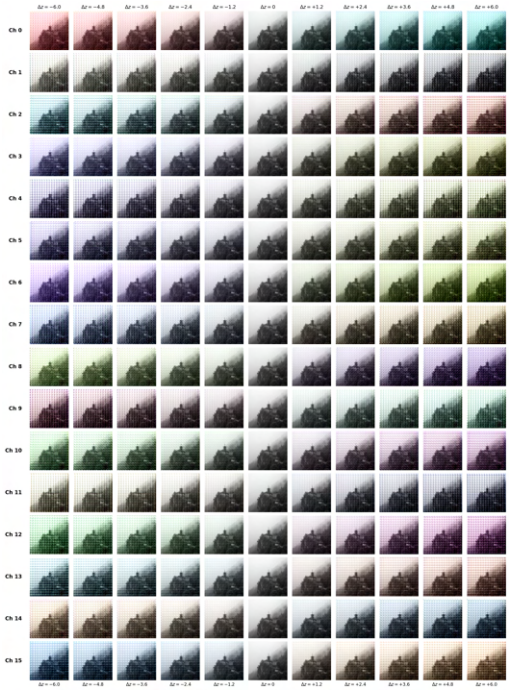
(a) 3-channel AE



(b) 4-channel AE

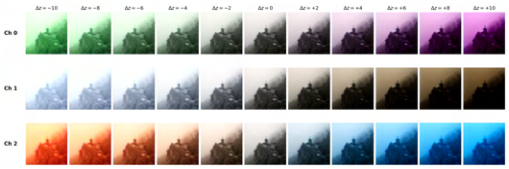


(c) 8-channel AE

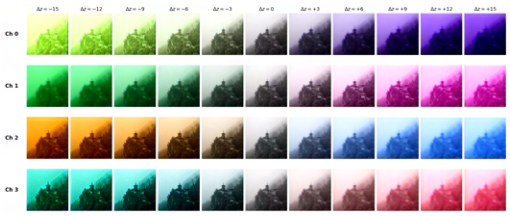


(d) 16-channel AE

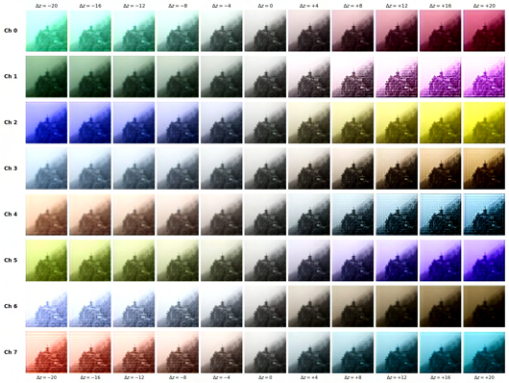
Figure 7: Visual impact of individual latent channel shifts (Δz) across different AE.



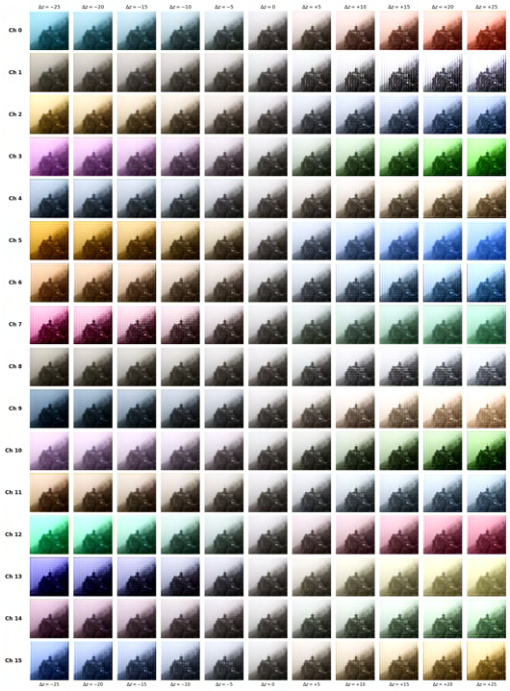
(a) 3-channel KL-VAE



(b) 4-channel KL-VAE

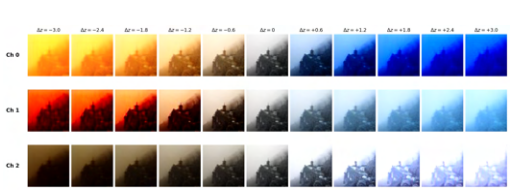


(c) 8-channel KL-VAE

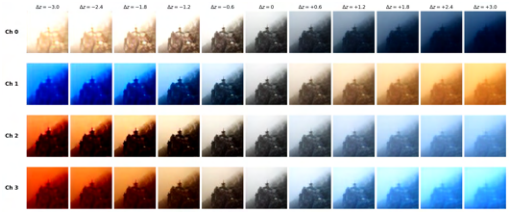


(d) 16-channel KL-VAE

Figure 8: Visual impact of individual latent channel shifts (Δz) across different KL-VAE.



(a) 3-channel VQ-VAE



(b) 4-channel VQ-VAE

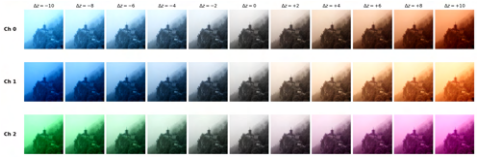


(c) 8-channel VQ-VAE

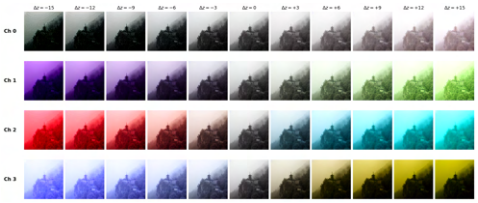


(d) 16-channel VQ-VAE

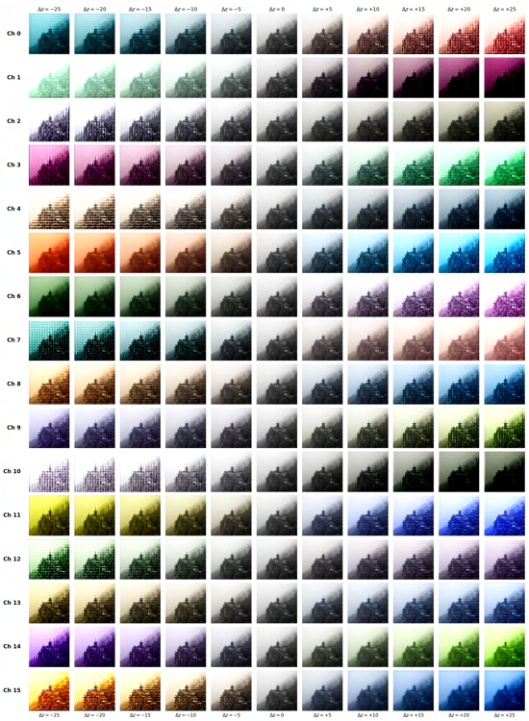
Figure 9: Visual impact of individual latent channel shifts (Δz) across different VQ-VAE.



(a) KL-f4 (3-channel)

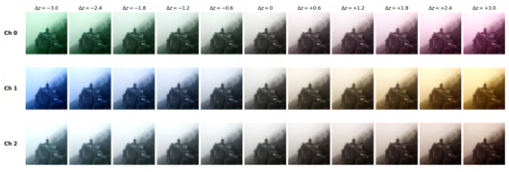


(b) KL-f8 (4-channel)

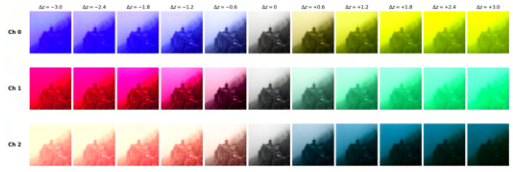


(c) KL-f16 (16-channel)

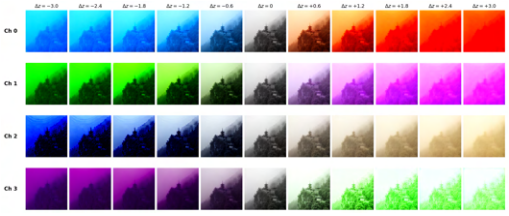
Figure 10: Latent channel shifts applied to KL-regularized VAEs with varying downsampling factors (f4, f8, f16).



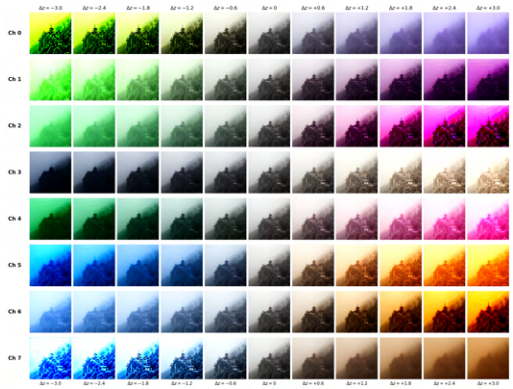
(a) VQ-f4 (3-channel)



(b) VQ-f4-noattn (3-channel)

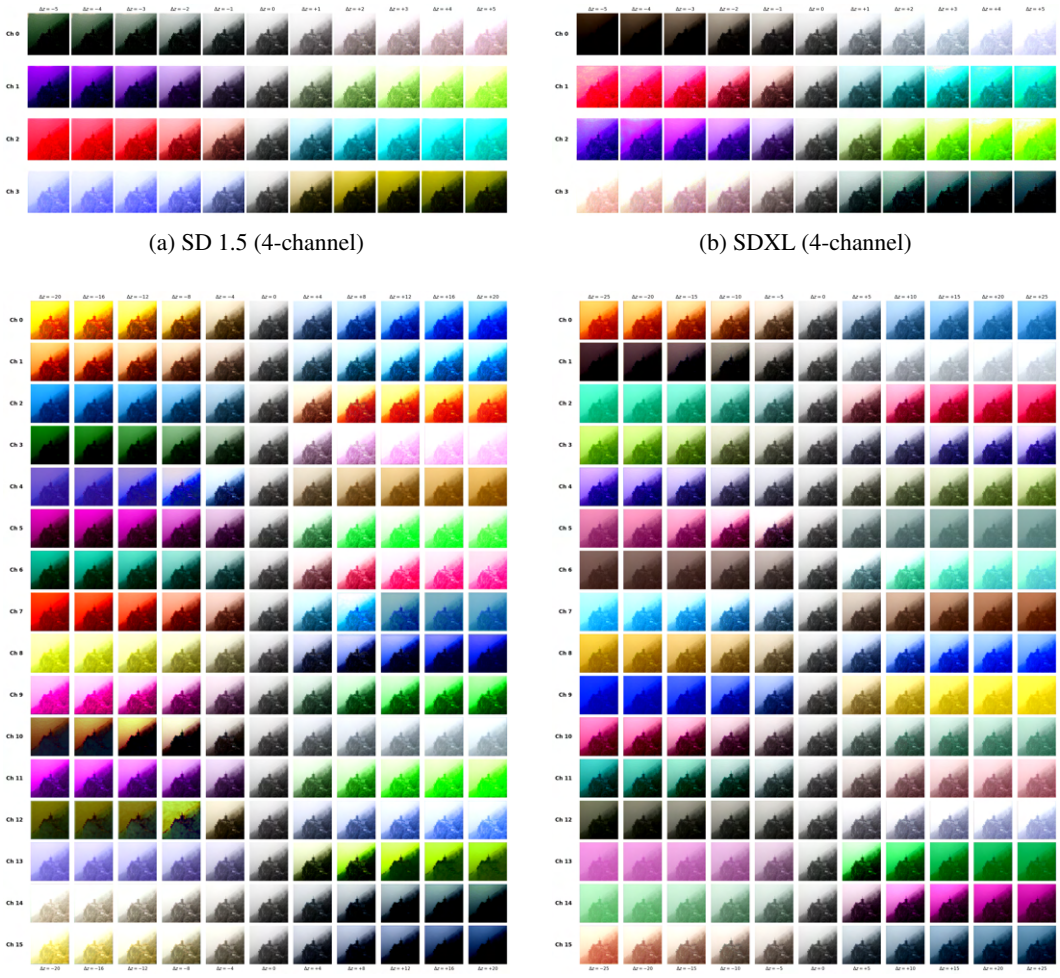


(c) VQ-f8 (4-channel)



(d) VQ-f16 (8-channel)

Figure 11: Latent channel shifts across Vector-Quantized (VQ) VAEs.



(a) SD 1.5 (4-channel)

(b) SDXL (4-channel)

(c) SD 3 (16-channel)

(d) FLUX (16-channel)

Figure 12: Latent channel shifts in state-of-the-art foundation models. The top row compares the 4-channel autoencoders of SD1.5 and SDXL. The bottom row illustrates the expanded 16-channel latent spaces of SD3 and FLUX.1.

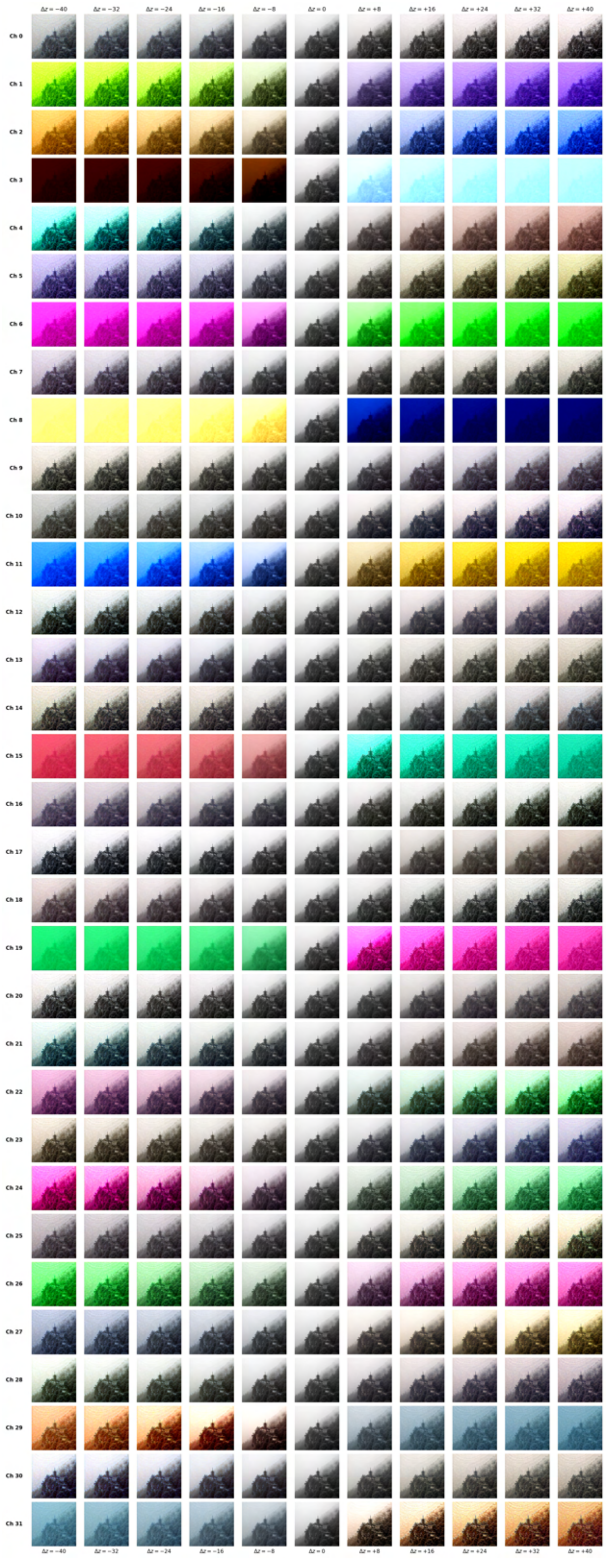


Figure 13: Extended latent channel shifts for the 32-channel FLUX2 autoencoder.

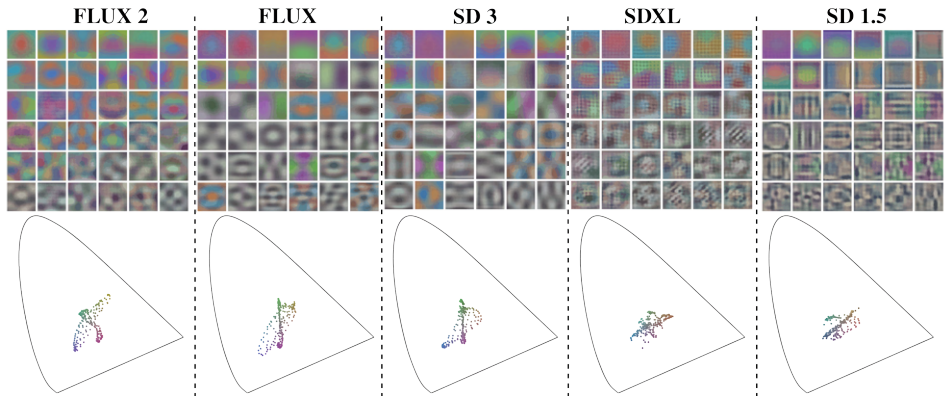


Figure 14: The first singular vectors of the linear approximation of VAEs and their corresponding CIE chromaticity directions across generative models used with 64×64 images: Flux.dev 2, Flux.dev 1, (c) SD 1.5, (d) SDXL, and SD 3. Each subplot visualizes the singular vectors ordered by their chromatic variance alongside their color content in the CIE chromatic diagram.

64×64 images

	FLUX 2	FLUX	SD3	SDXL	SD1.5	**AVERAGE**
Lin. Enc. + Lin. Dec						
Fract. Lin. Resp.	0.67	0.73	0.72	0.71	0.74	**0.71** $\pm$ **0.02**
Error No Linear	0.02	0.03	0.05	0.08	0.08	**0.05** $\pm$ **0.02**
Error Linear	0.11	0.09	0.09	0.09	0.09	**0.09** $\pm$ **0.01**
Lin. Enc. + NonLin. Dec						
Fract. Lin. Resp.	0.67	0.74	0.58	0.61	0.69	**0.66** $\pm$ **0.06**
Error No Linear	0.02	0.03	0.05	0.08	0.08	**0.05** $\pm$ **0.02**
Error Linear	0.11	0.08	0.13	0.13	0.10	**0.11** $\pm$ **0.02**
NonLin. Enc. + Lin. Dec						
Fract. Lin. Resp.	0.70	0.76	0.75	0.71	0.72	**0.73** $\pm$ **0.02**
Error No Linear	0.02	0.03	0.05	0.08	0.08	**0.05** $\pm$ **0.02**
Error Linear	0.10	0.08	0.08	0.09	0.09	**0.09** $\pm$ **0.01**

16×16 images

	FLUX 2	FLUX	SD3	SDXL	SD1.5	**AVERAGE**
Lin. Enc. + Lin. Dec						
Fract. Lin. Resp.	0.83	0.80	0.69	0.55	0.73	**0.72** $\pm$ **0.10**
Error No Linear	0.05	0.09	0.37	0.27	0.19	**0.19** $\pm$ **0.10**
Error Linear	0.05	0.06	0.09	0.14	0.08	**0.08** $\pm$ **0.03**
Lin. Enc. + NonLin. Dec						
Fract. Lin. Resp.	0.67	0.68	-0.03	0.23	0.45	**0.40** $\pm$ **0.30**
Error No Linear	0.05	0.09	0.37	0.27	0.19	**0.19** $\pm$ **0.10**
Error Linear	0.09	0.10	0.31	0.23	0.17	**0.18** $\pm$ **0.09**
NonLin. Enc. + Lin. Dec						
Fract. Lin. Resp.	0.78	0.75	0.47	0.41	0.60	**0.60** $\pm$ **0.20**
Error No Linear	0.05	0.09	0.37	0.27	0.19	**0.19** $\pm$ **0.10**
Error Linear	0.07	0.08	0.20	0.21	0.12	**0.14** $\pm$ **0.04**

Table 2: Linear approximation metrics across all models, configurations and datasets.

magnitudes of the reconstruction error by the VAEs and by their linear approximations are similar (about 12% of the energy of the original signal).

These results imply that the singular vectors shown in this work are a meaningful description of what the VAEs are sensitive to, both in terms of texture and color directions.

A.6 T2I MODELS FAIL AT NUMERICAL COLOR PRECISION

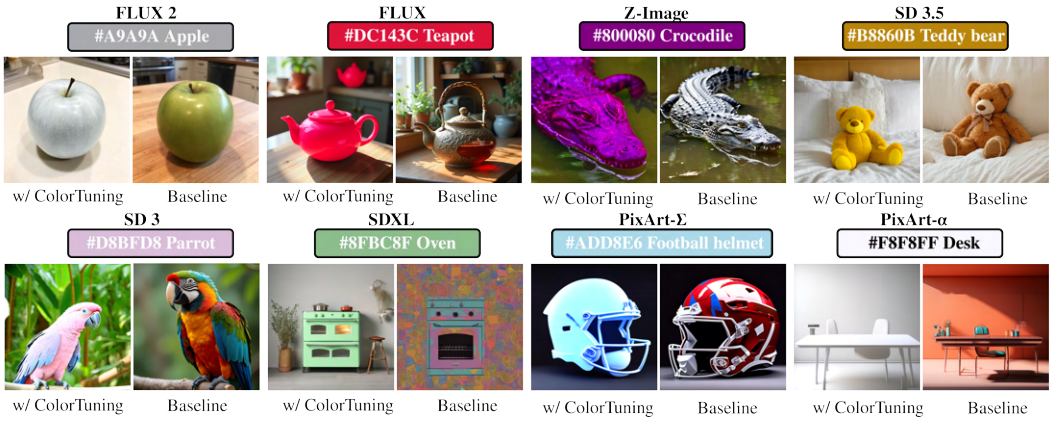


Figure 15: T2I models fail at numerical color precision.

A.7 PIPELINES

We evaluate our method across a diverse set of state-of-the-art text-to-image models operating on distinct VAE latent spaces. ColorTuning is implemented on models with 4-channel (SDXL-Base-1.0), 16-channel (SD3-Medium, SD3.5-Medium, Z-Image, FLUX.1-dev), and FLUX.2-dev with 32-channel. To compute the spatial conditioning mask, we use SAM-3 (`facebook/sam3`) with the target object from the prompt.

A.7.1 TEMPORAL INJECTION SCHEDULES

Perturbations along the three principal PCA color axes are applied over an active generation window $[\tau_{\text{gate}}, 1.0]$. These perturbations are modulated by the spatial mask and a temporal schedule $w(t)$. The specific gate step τ_{gate} and the optimal schedule profile for each model are detailed in Table 3. We employ three distinct temporal profiles:

- **Ascending Ramp** ($w(t) = \frac{t-\tau_{\text{gate}}}{1-\tau_{\text{gate}}}$): Gradually increases perturbation intensity.
- **Triangular Profile**: Reaches peak intensity at the mid-window to preserve fine structural textures.
- **Constant Profile** ($w(t) = 1.0$): Applies uniform injection across the active window.

A.7.2 SHIFT PREDICTOR NETWORK ARCHITECTURE

To map the desired color shifts to PCA magnitude vectors $\mathbf{m} = (m_1, m_2, m_3)^\top$, we train a Multi-Layer Perceptron with residual connections.

- **Input Features:** The network ingests a 15-dimensional perceptual feature vector combining initial and target CIELAB coordinates, chroma values, and periodic sine/cosine hue representations.
- **Architecture:** An input projection layer (15 to 256 dimensions) is followed by two residual blocks (each containing two 256-dimensional linear layers, LayerNorm, SiLU activation, and a dropout of $p = 0.05$). A final bottleneck projects the features down to 3 dimensions to predict the shift.
- **Training Setup:** The network is optimized using AdamW (learning rate $\eta = 10^{-3}$, weight decay 10^{-4}) with a Cosine Annealing schedule (minimum learning rate $\eta_{\min} = 10^{-5}$). We train for 80 epochs using a Smooth L1 loss ($\beta = 0.05$), a batch size of 64, and an early stopping patience of 15 epochs.

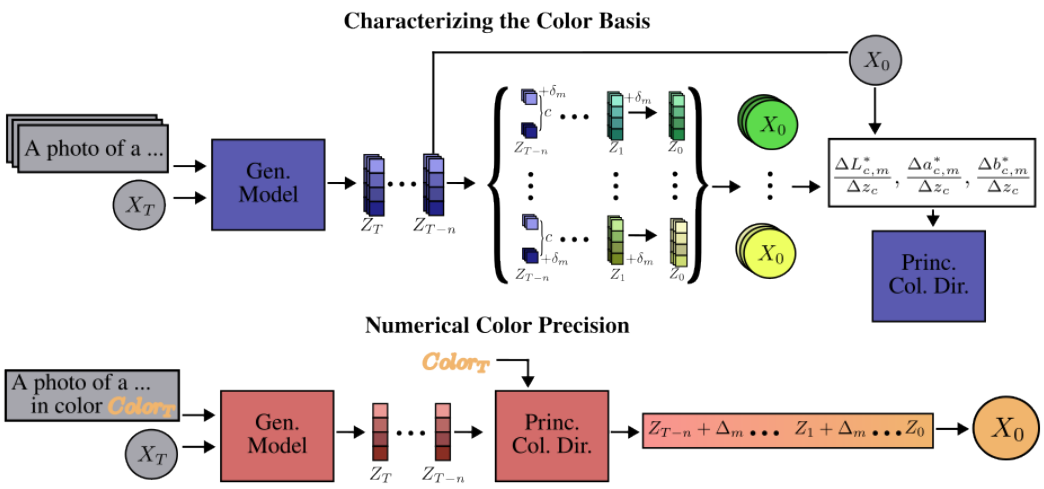


Figure 16: Overview of our method. **Top:** characterizing the color basis of a VAE (Sec. 4.1). Each latent channel is perturbed inside the region of a set of reference objects, and the resulting shifts in CIELAB form a sensitivity matrix, whose SVD gives the luminance and opponent-color directions. This is computed once per VAE. **Bottom:** ColorTuning at inference (Sec. 4.2). The numerical code in the prompt is replaced by the nearest color name. At the gate step, the predicted clean latent $\hat{z}_0$ is decoded, the object is segmented with SAM 3, and its mean color is measured. The MLP maps this color and the target to a displacement along the color basis, which is added inside the object mask over the remaining steps.

Table 3: Pipeline Specifications and Hyperparameters .

Model	Steps	CFG	Gate τ_{gate}	Temporal Schedule
FLUX.1-dev	28	3.5	0.50	Ascending
FLUX.2-dev	28	3.5	0.65	Ascending
SD3-Medium	28	7.0	0.75	Triangular
SD3.5-Medium	28	4.5	0.60	Ascending
SDXL-Base	30	5.0	0.40	Constant
Z-Image	28	4.0	0.60	Constant

A.8 Saturation Control and Color Transfer

Saturation Control. Saturation control continuously modulates the chroma of the generated scene or specific objects without altering their semantic structure or hue. In this setting, the text prompt requires no proxy injection, allowing the diffusion trajectory to proceed unconstrained until the gate step. At step s, we decode $\hat{z}_0$ and obtain the CIELAB color $C_i(x, y) == (L_i(x, y), a_i(x, y), b_i(x, y))$ at each spatial location (x, y) within the target region (or across the full image). For a desired chroma reduction factor $\alpha \in [0, 1]$, we preserve the lightness L_i and hue angle while scaling chroma, setting the target color to $C_t = (L_i, (1 - \alpha)a_i, (1 - \alpha)b_i)$. The pretrained MLP takes (C_i, C_t) and evaluates $(C_i(x, y), C_t(x, y))$ across the grid to predict a dense displacement map $m_1(x, y), m_2(x, y), m_3(x, y)$. Adding this displacement over the remaining steps pulls the generation into the restricted gamut within the latent space (see Figure 17).

Color Transfer. For color transfer, the target color distribution is specified either by a discrete color palette (e.g., a set of hex values) or by an exemplar reference image. When using an image, we first extract its principal colors via k-means clustering in CIELAB, enforcing a minimum perceptual distance in CIEDE2000 to obtain differentiated dominant, focal, and secondary accent tones. To initialize the trajectory in the appropriate chromatic basin, we take the most chromatic color, locate its nearest entry in ISCC-NBS Level 2, and inject this proxy name into the text prompt for the main

Algorithm 1 ColorTuning : training the displacement predictor

Require: frozen VAE (E, D), generator, color basis $U \in \mathbb{R}^{C\times 3}$, prompt set $\mathcal{P}$, displacement set $\mathcal{M}$

$\mathcal{D} \leftarrow \emptyset$
for $p \in \mathcal{P}$ **do**
  $I_0 \leftarrow \textsc{Generate}(p)$ ▷ no intervention
  $c_0 \leftarrow \textsc{MeanLab}(I_0, \textsc{Sam3}(I_0, p))$
  for $m \in \mathcal{M}$ **do**
    $I_m \leftarrow \textsc{Generate}(p, \Delta z = Um)$
    $c_m \leftarrow \textsc{MeanLab}(I_m, \textsc{Sam3}(I_m, p))$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{((c_0, c_m), m)\}$
  end for
end for
train MLP $f_\psi : (c_i, c_t) \mapsto m$ on $\mathcal{D}$

Ensure: f_ψ

Algorithm 2 ColorTuning : inference

Require: prompt p with target color c_t, gate step s, total steps T, basis U, predictor f_ψ, proxy table $\mathcal{L}$

$p' \leftarrow \textsc{ReplaceCode}(p, \textsc{NearestName}(c_t, \mathcal{L}))$
$z_T \sim \mathcal{N}(\mathbf{0}, I)$
for $t = T, \ldots, 1$ **do**
  if $t = s$ **then**
    $\hat{z}_0 \leftarrow$ Eq. 2; $\quad I \leftarrow D(\hat{z}_0)$
    $M \leftarrow \textsc{Downsample}(\textsc{Sam3}(I, p'))$
    $c_i \leftarrow \textsc{MeanLab}(I, M)$; $\quad \Delta z \leftarrow U f_\psi(c_i, c_t)$
  end if
  $z_{t-1} \leftarrow \textsc{Step}(z_t, t)$
  if $t \le s$ **then**
    $z_{t-1} \leftarrow z_{t-1} + w(t)\,(M \odot \Delta z)$ ▷ w decays linearly to 0
  end if
end for

Ensure: $D(z_0)$

subject. At gate step s, we decode the predicted clean latent $\hat{z}_0$ and partition the scene into spatial regions: the focal object segmented with SAM 3, secondary entities, and the background (which can be tonally partitioned into highlight and shadow zones). We assign each region to its corresponding reference color. The pretrained MLP then evaluates each pair of measured and target CIELAB colors independently, predicting a displacement (m_1, m_2, m_3) for each region along the color basis. See qualitative results in Figure 18.

A.9 User Study

We conducted a psychophysical experiment in a controlled laboratory with standardized lighting and color-calibrated sRGB monitors to perceptually evaluate our color accuracy. A total of 22 participants with normal color vision completed a Two-Alternative Forced Choice (2AFC) study consisting of two distinct comparisons based on the underlying diffusion backbone: ColorTuning against NumColor (both FLUX-based, 50 prompts) and ColorTuning against ColorWave (both SDXL-based, 50 prompts). In each trial, observers viewed a central reference color patch and text prompt positioned between two randomly ordered generated images (ours and the corresponding baseline), and were asked to choose which image better matched the target color. We analyzed the raw preference data using the Thurstone Case V Law of Comparative Judgment to compute accuracy z-scores, with 95% confidence intervals calculated following Montag (2006). As illustrated in Figure 19, a z-score greater than zero indicates a preference for our method. ColorTuning demonstrated a statistically significant preference over both competitors, achieving a z-score of 2.62 ± 0.30 against NumColor and 1.67 ± 0.30 against ColorWave.

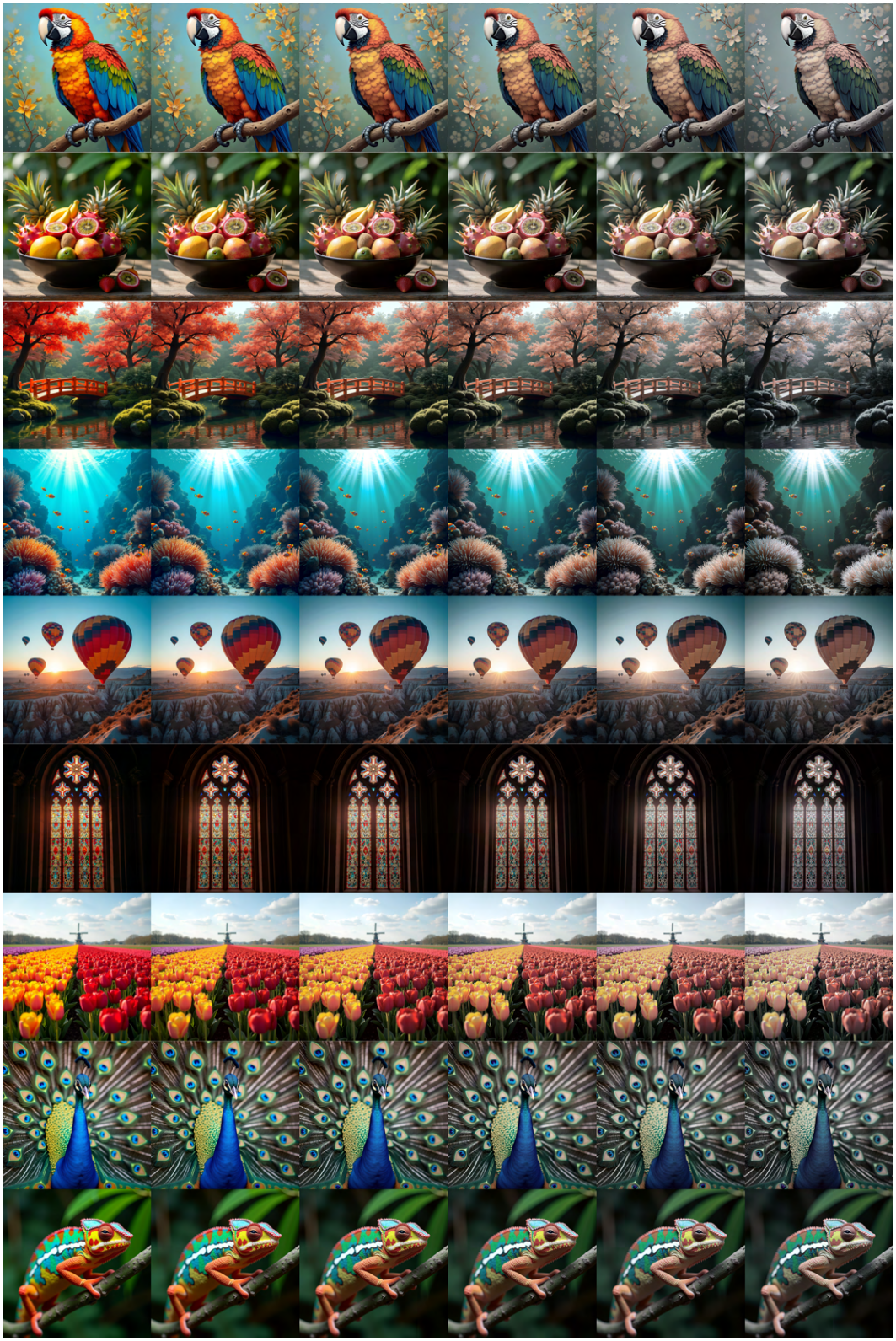


Figure 17: Qualitative results for saturation control.

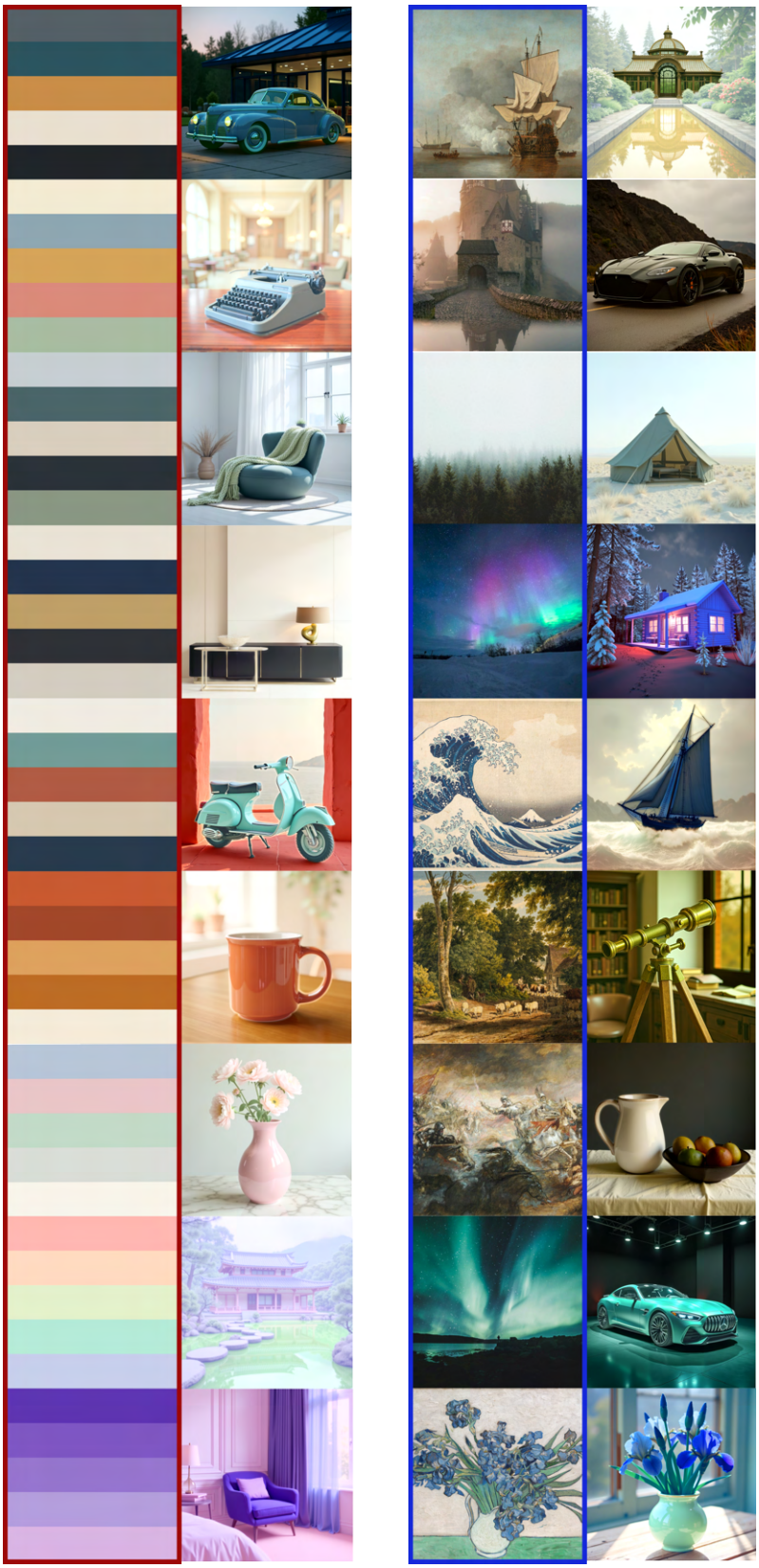


Figure 18: Color transfer results. The first two columns display the target color palettes and their corresponding generated images. The final two columns present the color reference images alongside the resulting generated outputs.

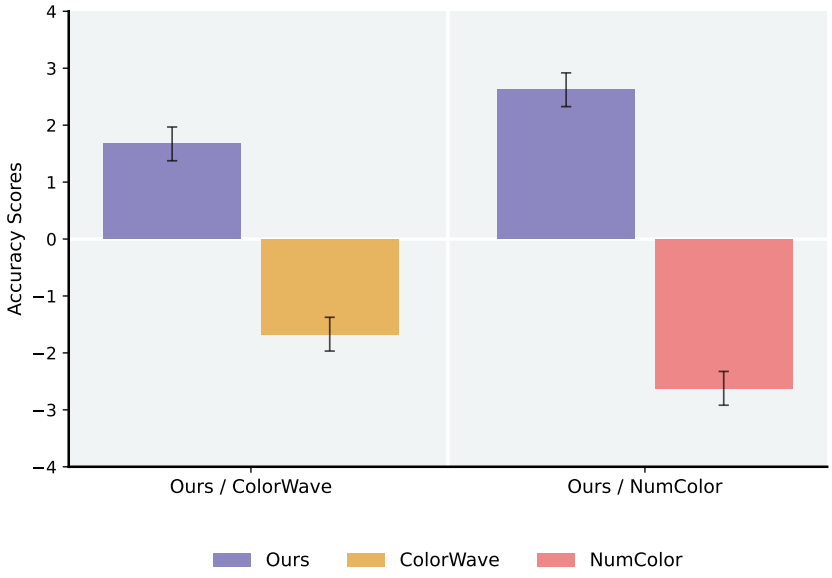


Figure 19: Thurstone case V results of our user's study. Values are z-scores. Error bars represent 95% confidence intervals —see Montag (2006). Our method is statistically better than existing methods: ColorWave and NumColor.