

NumColor: Precise Numeric Color Control in Text-to-Image Generation

Muhammad Atif Butt^{1,2}, Diego Hernández^{1,2}, Alexandra Gomez-Villa^{1,2},
Kai Wang^{3,4*}, Javier Vazquez-Corral^{1,2}, and Joost van de Weijer^{1,2}

¹ Computer Vision Center, Barcelona, Spain

² Computer Sciences Department, Universitat Autònoma de Barcelona, Spain

³ Program of Computer Science, City University of Hong Kong (Dongguan)

⁴ City University of Hong Kong

{mabutt, agomezvi, jvazquez, joost}@cvc.uab.es,
diego.hernandez@autonoma.cat, kai.wang@cityu-dg.edu.cn

Abstract. Text-to-image diffusion models excel at generating images from natural language prompts, yet fail to interpret numerical colors such as hex codes (`#FF5733`) and RGB values (`rgb(255, 87, 51)`). This limitation stems from subword tokenization, which fragments color codes into semantically meaningless tokens that text encoders cannot map to coherent color representations. We present NumColor, that enables precise numerical color control across multiple diffusion architectures. NumColor comprises two components: a Color Token Aggregator that detects color specifications regardless of tokenization, and a ColorBook containing 6,707 learnable embeddings that map colors to embedding space of text encoder in perceptually uniform CIE Lab space. We introduce two auxiliary losses, directional alignment and interpolation consistency, to enforce geometric correspondence between Lab and embedding spaces, enabling smooth color interpolation. To train the ColorBook, we construct NumColor-Data, a synthetic dataset of 500K rendered images with unambiguous color-to-pixel correspondence, eliminating the annotation ambiguity inherent in photographic datasets. Although trained solely on FLUX, NumColor transfers zero-shot to SD3, SD3.5, PixArt- α , and PixArt- Σ without model-specific adaptation. NumColor improves numerical color accuracy by 4–9 $\times$ across five models, while simultaneously improving color harmony scores by 10–30 $\times$ on GenColorBench benchmark. The code is available at <https://github.com/moatifbutt/numcolor/>.

Keywords: T2I generation · colors · diffusion model

1 Introduction

Diffusion models have transformed text-to-image (T2I) generation, achieving remarkable visual fidelity. However, achieving precise color control remains a challenge. Although models reliably respond to basic colors (*e.g.*, red, green) [3], their performance degrades for precise color description. The degradation is severe when conditioned on numeric color codes (*e.g.*,  `#E34234` in hexadecimal) [5]. As these codes are standard for communicating color in professional design software (such as Adobe Suite,

* Corresponding Author

Blender, etc.), this limitation restricts the applicability of these models in advertising, brand-driven content creation, where strict color fidelity is required.

In Fig. 1 (bottom-left), we show failure cases in generation conditioned on numeric color codes for several models, including FLUX and SD3. Recent benchmark of T2I models [5] confirms these findings, reporting accuracy below 15% on tasks involving numeric color specifications. This stems from how standard text encoders i.e., CLIP [45] and T5 [46] tokenize these inputs [40]: rather than treating as color instruction, they are fragmented into separate tokens (e.g.,  #FF5733 becomes #, FF, 57, 33), which not only lack color representation in embedding space, but also disrupt cross-attention. As shown in Fig. 1 (top-left), each sub-token attends to arbitrary image regions rather than target object, breaking color-object binding during generation. This is further supported by our embedding geometry analysis in Fig. 1 (right) across five text encoders. While color names preserve strong consistency with ground-truth color space, both RGB and hexadecimal codes exhibit an approximate 60% degradation.

Beyond tokenization, the problem is further exacerbated by limited exposure to numerical color descriptions in training data of T2I models. Image captions in large-scale text–image datasets predominantly describe colors using color names (e.g., “red,” “solferino”), while explicit numerical color codes are rarely used. To address the lack of training data for numeric color supervision, we introduce **NumColor-Data**, a large-scale dataset which contains 500,000 rendered images of diverse objects spanning a wide range of specified colors. Each image provides unambiguous color-to-pixel correspondence, eliminating lighting and material ambiguities inherent in existing datasets.

We propose **NumColor** that enables precise numeric color generation in T2I diffusion models. While prior work such as ColorPeel [6] has explored color prompt learning, it is limited to a small set of color names and requires retraining for each new color. NumColor instead addresses the problem by proposing a new color encoder and a codebook that allow models to understand precise color specifications. More in detail, we address tokenizer fragmentation by introducing a tokenizer-agnostic Color Token Aggregator that identifies numeric color spans independent of underlying text encoder. The detected colors are then mapped to text embedding space via a Color-based codebook, that we term *ColorBook*: a set of learnable embeddings grounded in perceptually uniform CIE Lab color space, injected directly into the text encoder to enable proper color-object binding during generation. In summary, our main contributions are:

- **NumColor.** A framework for precise numerical color conditioning in T2I models. It introduces a tokenizer-agnostic *Color Token Aggregator* that operates independently of the tokenization scheme (e.g., CLIP or T5), with *ColorBook*, a learnable codebook that maps colors in the sRGB gamut to the text embedding space.
- **NumColor-Data.** The first large-scale dataset of objects with precisely numerical color descriptions. It contains over 500K images and covers the sRGB gamut, enabling clean supervision for numerical color conditioning.
- **Results.** We evaluate NumColor on 9.8K prompts spanning 400+ colors from GenColorBench [5]. Our method achieves 4–9× improvement in color accuracy and 10–30× improvement in color harmony across FLUX.1, SD3, SD3.5, PixArt- α , and PixArt- Σ . The ColorBook, trained solely on FLUX.1, transfers zero-shot to SD3, SD3.5, PixArt- α , and PixArt- Σ without any model-specific adaptation.

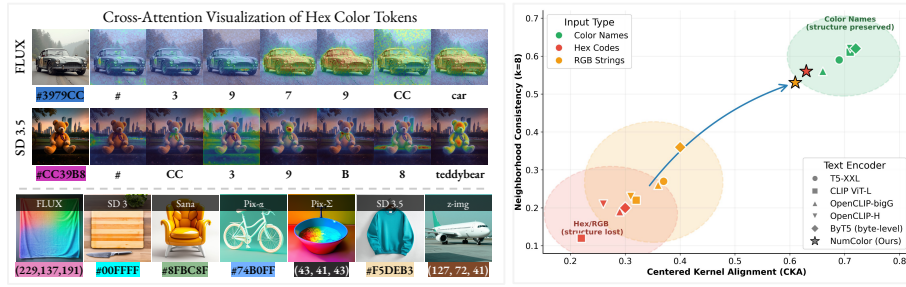


Fig. 1: Lack of Numeric Color Understanding. *Left:* Text encoders fragment hex codes into arbitrary subwords that attend to unrelated image regions, leading to incorrect color generation. *Right:* CKA [31] and neighborhood consistency ($k=8$) between text embeddings and Lab space show that color names preserve perceptual structure, while hex codes and RGB values degrade by $\sim 60\%$. NumColor recovers the structure comparable to color names.

2 Related Work

2.1 Text-based Image Generation and Editing

With the rapid evolution of generative AI, remarkable advancements have been achieved in T2I generation and editing. In particular, T2I diffusion models [23, 26] emerged as an effective paradigms, outperforming GANs [22], VAEs [30], autoregressive [18], and normalizing flows [14, 15] in T2I generation. Diffusion models are probabilistic generative frameworks that learn underlying data distributions through iterative denoising, starting from Gaussian noise. To enhance controllability, these models can be conditioned on various signals, including class labels [53], images [38], or text prompts [41]. With recent scaling of diffusion models, state-of-the-art T2I models such as SD3 [17] and FLUX [33] have significantly outperformed their predecessors [8, 10, 47].

Diffusion models lack inherent support for text-guided image manipulation. P2P [25] pioneered controllable editing by aligning spatial layouts with text prompts via cross-attention, +InstructPix2Pix [4] extended this through instruction tuning to enable text-guided manipulation. Image inversion is further driven by optimized text embeddings by Imagic [28] to align the original appearance of the image with target semantic concepts, and Inversion-Free editing process [60] reformulated denoising process into multi-step framework to achieve stable inversion. Recently, Flow Matching [35, 36] is integrated with transformers [42] and FLUX [33] further enhanced image fidelity. Despite notable advances, diffusion-based editing methods remain constrained by limited precise color control. These models typically treat color as a stylistic attribute rather than spatially localized feature which motivates our work to develop a unified diffusion framework that integrates semantic localization and chromatic precision at token level.

2.2 Diffusion-based Color Control

With the rapid advancements in generative models [47, 48, 50], a variety of text-guided image editing approaches have been developed to enable controllable image modifica-

tions [24, 38, 39]. Meanwhile, unified models [12, 58] integrate such editing capabilities through large-scale pretraining on massive paired datasets. Another technical stream for achieving controllable generation is transfer learning applied to text-to-image (T2I) models [32, 49], which is also termed T2I model adaptation or personalized generation. This paradigm aims to adapt a pretrained T2I model to a new concept using user-provided images, binding the new concept to a unique token. Consequently, the adapted model can generate diverse renditions of the new concept under the guidance of text prompts. These adaptation methods can be categorized based on whether they fine-tune the entire T2I model [32, 49] or freeze the T2I backbone while only optimizing auxiliary components [16, 20, 56, 61]. However, all these existing techniques struggle to achieve fine-grained control over color attributes in both image editing and generation tasks. To date, only a limited number of works have begun to address precise color generation [27, 43, 44, 54]. For instance, Rich-Text [21] enhances color fidelity through multi-pass processing with global and local diffusion models, but this comes at the cost of high computational overhead and compromised color accuracy. In contrast, Color-Peel [6] introduces color prompt learning to improve the alignment between generated colors and user inputs. Nevertheless, this method requires extensive training and is constrained to handling only a small number of color names per training session.

Moreover, this problem persists in newer models. Recent work [5] shows that state-of-the-art methods fail to reach even 10% accuracy on numeric colors. We address this gap with NumColor to enable precise numerical color control in T2I diffusion models.

3 Method

Given a text prompt containing a numeric color, our goal is to address two compounding failures: (i) tokenizer fragmentation, which prevents color codes from being treated as coherent semantic units, and (ii) the alignment gap between numerical color representations and the text encoder embedding space, leaving unified tokens perceptually ungrounded. To this end, we propose NumColor (Figure 2), a two-stage framework consisting of a Color Token Aggregator that resolves fragmentation in a tokenizer-robust manner, and ColorBook, which bridges the alignment gap by mapping numeric colors into the text embedding space through the perceptually uniform CIE Lab color space.

3.1 Preliminary

Diffusion Transformers. T2I diffusion models have evolved from U-Net based architectures to diffusion transformers which offer improved scalability and generation quality. Recent T2I models such as SD3 and Flux adopt this transformer based approach as a backbone. These models operate in VAE latent space, and learn to reverse a diffusion process that gradually adds noise to the images. Given a noisy latent z_t at timestep t , the DiT is trained to predict the added noise (or velocity), enabling iterative denoising at inference. Modern DiTs employ rectified flow [17], which learns straight-line trajectories between noise and data distributions, allowing for efficient few-step generation.

Text Conditioning. T2I diffusion transformers condition image generation on embeddings from pretrained text encoders, typically CLIP [45] and T5 [46]. CLIP, trained

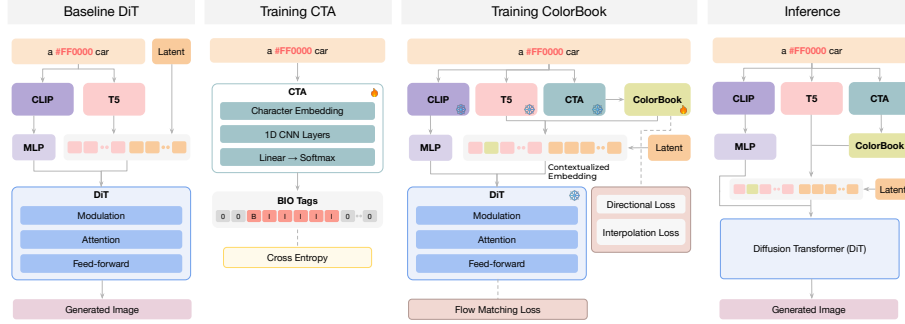


Fig. 2: Method overview. (a) *Baseline*: CLIP and T5 encode text; DiT generates images via iterative denoising. (b) *Color Token Aggregator*: A character-level sequence labeler unifies fragmented color tokens using linear classifier, trained with cross-entropy on numeric color text prompts. (c) *ColorBook Training*: Unified color tokens receive learned embeddings before T5 contextualization. The ColorBook (6,707 Lab anchors) is trained with flow matching loss on NumColor-Data, with directional and interpolation losses to preserve Lab geometry. Only ColorBook embeddings receive gradients; CLIP, T5, and DiT remain frozen.

through contrastive image-text alignment, provides semantically rich embeddings suited for visual concepts, while T5 offers better linguistic understanding for complex prompts. Current T2I models combine both to improve text conditioning. For instance, SD3 and SD3.5 employ CLIP-G and CLIP-L for pooled embeddings alongside T5-XXL, FLUX uses CLIP with T5-XXL, whereas PixArt uses T5 alone for text conditioning.

3.2 Color Token Aggregator

As demonstrated in Figure 1, subword tokenization fragments numerical color specifications into arbitrary tokens that lack color semantics. The first challenge is identifying color tokens independent of how tokenizers fragment them. We introduce the Color Token Aggregator (CTA), a lightweight character-level module inspired by sequence labeling that detects numeric color specifications and unifies their fragments into single, semantically coherent tokens, regardless of the underlying vocabulary. Given the token string from the tokenizer, each token is converted to a fixed-length character sequence (max 32 characters), embedded via a character embedding layer ($d_c = 64$), and processed by parallel 1D convolutions with kernel sizes $\{2, 3, 4\}$ to capture patterns at multiple scales. The convolution outputs pass through ReLU activations and max pooling, then concatenated to form a 256-dimensional token representation. A 4-layer transformer encoder with 4 attention heads contextualizes these representations across the token sequence. Finally, a linear classifier produces per-token emission scores for the three $\{B, I, O\}$ (which represents Beginning, Inside, and Outside) tags:

$$\mathbf{s}_i = \mathbf{W}h_i + \mathbf{b}, \quad \mathbf{s}_i \in \mathbb{R}^3 \quad (1)$$

where $h_i \in \mathbb{R}^d$ is the contextualized representation of the i -th token from the transformer encoder, $\mathbf{W}$ is the projection matrix, and the output dimensions correspond to

tag scores for $\{\text{B}, \text{I}, \text{O}\}$. A CRF layer [34] models tag transition dependencies, ensuring valid (Beginning, Inside, Outside) sequences. Training minimizes the negative log-likelihood under CRF. We curate 50K base prompts from DiffusionDB [57] containing color name mentions, and replace color names with hex codes or RGB tuples. These prompts are tokenized across four text encoders (T5, CLIP-L/G, GPT-2), totaling 200K training samples. At inference, Viterbi decoding [52] extracts optimal tag sequences, as color specifications are parsed into Lab coordinates for our ColorBook lookup.

3.3 ColorBook

The CTA identifies numeric color fragmentation and unifies into a single color token, however these text tokens remain outside the learned vocabulary of the pretrained text encoder, since no pretrained embedding exists for the tokens like ■ #E34234 or `rgb(255, 87, 51)`. A naive approach to enable precise color control would be to learn a dedicated embedding for each color. However, the RGB color space contains over 16.7 million distinct values (256^3), making it infeasible to learn individual embeddings for every color. Even when transformed to CIE Lab space using standard sRGB transfer functions [19], the sRGB gamut occupies a non-convex shaped region that remains prohibitively large for the embedding coverage. Learning this amount of color tokens directly would require prohibitive memory and training data.

ColorBook construction. We draw inspiration from vector quantization methods in generative modeling, where compact codebooks enable efficient representation of the high-dimensional space. In VQ-VAE [55], a discrete codebook bridges encoder and decoder for image reconstruction. We repurpose this principle for color representation by constructing a codebook over color space itself, mapping perceptual color anchors to text encoder embeddings. Specifically, we define a codebook $\mathbf{E} \in \mathbb{R}^{K \times d}$ of anchor colors in the perceptually uniform CIE Lab space, which we refer to as **ColorBook**. We choose Lab over RGB because Euclidean distance in Lab correlates with perceived color difference, whereas RGB lacks this property. We uniformly sample Lab space at 5-unit intervals within sRGB gamut boundaries, yielding $K = 6,707$ anchor colors. This approach results in a reduction of over three orders of magnitude from naive per-color embeddings while preserving perceptual uniformity. Each anchor $\mathbf{a}_k \in \mathbb{R}^3$ maps to a learnable embedding $\mathbf{e}_k \in \mathbb{R}^d$, with $d = 4096$ to match T5 hidden dimensions.

Embedding initialization. Training embeddings from random initialization requires jointly learning perceptual color organization and alignment with the representation space of text encoder, which is a challenging optimization. Prior work demonstrates that pretrained initialization accelerates convergence, particularly with limited training data. We leverage this by initializing anchor embeddings from T5-XXL’s color vocabulary, but rather than using individual color names (e.g., red, scarlet), which introduce linguistic biases, we use ISCC-NBS color category centroids [29] that is a standard system of 267 perceptually-defined categories. For each category, we compute its LAB centroid, find the nearest anchor, and initialize its embedding from the T5 encoding of the category name; remaining anchors are initialized via distance-weighted interpolation from initialized neighbors. To handle continuous colors with a discrete ColorBook, we employ soft interpolation where for a query color $\mathbf{c} \in \mathbb{R}^3$, we retrieve the $k = 8$ nearest anchors $\mathcal{N}_k(\mathbf{c})$ and compute a weighted combination:

$$\phi(\mathbf{c}) = \sum_{\mathbf{a}_j \in \mathcal{N}_k(\mathbf{c})} w_j \cdot \mathbf{e}_j, \quad w_j = \frac{\exp(-\|\mathbf{c} - \mathbf{a}_j\|_2/\tau)}{\sum_{\mathbf{a}_i \in \mathcal{N}_k(\mathbf{c})} \exp(-\|\mathbf{c} - \mathbf{a}_i\|_2/\tau)} \quad (2)$$

where τ controls interpolation sharpness. We set $\tau = 2.0$, balancing sharp color assignment ($\tau \rightarrow 0$) against overly diffuse interpolation ($\tau \rightarrow \infty$). This enables smooth transitions across the color gamut while maintaining only 6,707 learnable embeddings.

Color embedding injection. Modern T2I diffusion models employ several text encoding strategies. Simpler architectures like PixArt use T5 only, while FLUX and SD3 combine T5 with CLIP-L/G encoders serving complementary roles: CLIP provides pooled embeddings summarizing entire prompt for global timestep conditioning, while T5 provides token-level sequence embeddings capturing fine-grained textual features. Since CLIP lacks per-token granularity, sequence embeddings of T5 become natural injection target. However, the precise injection point critically determines color-object association. Post-hoc replacement inserts color embeddings after T5 has contextualized the sequence, bypassing self-attention and preventing the model from binding colors to objects. We instead inject before transformer layers, replacing corresponding token embeddings with $\phi(\mathbf{c})$, then process the modified sequence. This pre-contextualization injection enables bidirectional attention between color and object tokens (e.g., #FF5733 $\leftrightarrow$ car), facilitating color-object association during generation (Figure 7).

3.4 Dataset Curation

Training a ColorBook requires image-text prompt pairs where the specified color exactly matches the rendered appearance of the objects. Existing large-scale datasets fail this requirement. For instance, Butt et. al [5] show that LAION [51] captions rarely specify precise colors, and when they do, the photographed objects exhibit specular highlights, subsurface scattering, ambient occlusion, and environmental color casts that confound the mapping between prompt color and pixel color. A model trained on such data may perform well on coarse basic color generation, however, it would remain challenging to disentangle color specification from material and lighting effects.

To address this shortcoming, we construct **NumColor-Data**, a synthetic dataset providing unambiguous color supervision, as shown in Figure 3. We source 100 3D meshes from the Objaverse-XL [11] dataset. These meshes cover broad categories of objects, including furniture, vehicles, and items of daily use. Further, we collect 34 indoor and outdoor scenes from BlenderKit [1], and the HDRIs from [2] to create a diverse dataset. Each mesh is assigned a uniform Lambertian material with albedo set to a target Lab color. We render using Blender Cycles under the D65 illuminant with controlled intensity to ensure accurate color reproduction. The prompt templates are listed in the supplementary material. The final dataset comprises 500K images. For each object-scene pair, we sample the colors from the 6,707 Lab anchors, ensuring coverage throughout the color space. We use a 90/10 train/validation split stratified by objects. The rendered albedo exactly matches the prompt specification, providing direct supervision for learning the ColorBook without confounding material properties or lighting.

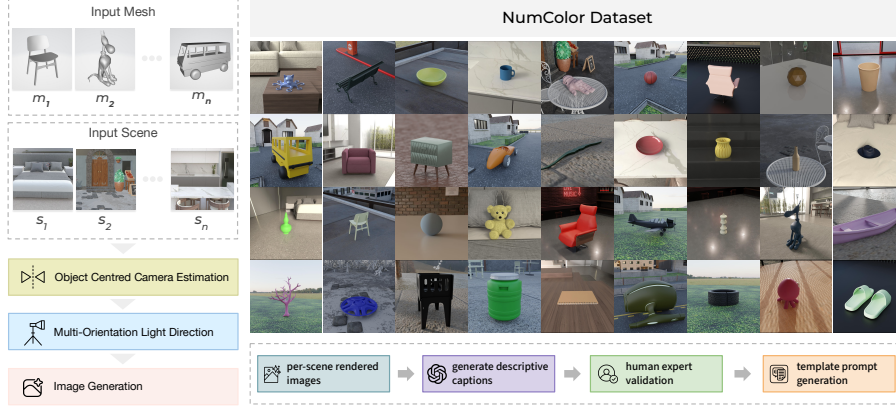


Fig. 3: NumColor-Data generation pipeline. We render 3D meshes from Objaverse-XL in diverse indoor and outdoor scenes. Each object is assigned a uniform Lambertian material with albedo set to the target Lab color. We use object-centered camera estimation and multi-orientation lighting to ensure diverse viewpoints. The dataset comprises 500K images spanning varied objects, scenes, and colors from ColorBook anchors. **Caption generation**—rendered images are paired with descriptive captions, validated by human experts, and converted to template prompts.

3.5 Training Objective

Learning color embeddings requires both accurate numeric color token unification and alignment between the learned embedding space and color structure. We achieve these objectives through a two-stage training.

Stage 1. We train the CTA aggregator on 200K text prompts (see Section 3.2 for details) to identify numeric color tokens from tokenized strings. The encoder is trained on 200K synthetic text prompts using negative log-likelihood under the CRF:

$$\mathcal{L}_{\text{CTA}} = -\log P(\mathbf{y}^*|\mathbf{p}), \quad (3)$$

where $\mathbf{p}$ is the tokenizer prompt string, and $\mathbf{y}^*$ is the ground-truth encoded sequence of the prompt. The CRF layer enforces valid tag transitions (such as, $\mathbb{I}$ cannot follow $\mathbb{O}$) which improves the sequence coherence over independent token classification. The results, shown in the Table S3 in the supplementary, show that CTA generalizes well to the GenColorBench and LAION-2B text prompts.

Stage 2. We freeze the CTA aggregator, and we train the ColorBook embeddings $\mathbf{E}$ on NumColor-Data. The base diffusion model and T5 [46] text encoder remain frozen; only $\mathbf{E}$ receives gradients. For rectified flow models, the primary objective is flow-matching:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{t, z_0, z_1} [\|v_\theta(z_t, t, \mathcal{E}(\mathbf{p}; \phi)) - (z_1 - z_0)\|_2^2], \quad (4)$$

where $z_t = (1 - t)z_0 + tz_1$, and $\mathcal{E}(\mathbf{p}; \phi)$ denotes T5 encoding [46] with color embeddings injected via ϕ . The flow matching objective trains the ColorBook to produce embeddings that generate correct colors, but provides no explicit supervision on the *geometry* of the embedding space. Without additional constraints, perceptually similar

colors may map to distant embeddings, degrading interpolation quality and generalization to unseen colors. Therefore, We introduce two auxiliary losses that enforce correspondence between Lab color space and embedding geometry. *Directional alignment* encourages embedding displacements to reflect perceptual displacements in Lab space:

$$\mathcal{L}_{\text{dir}} = \mathbb{E}_{\mathbf{c}_i, \mathbf{c}_j} [1 - \cos(\phi(\mathbf{c}_i) - \phi(\mathbf{c}_j), \psi(\mathbf{c}_i - \mathbf{c}_j))] \quad (5)$$

where $\mathbf{c}_i, \mathbf{c}_j \in \mathbb{R}^3$ are Lab coordinates sampled from anchor set, $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}^d$ denotes the ColorBook mapping from Lab space to text embedding space, and $\psi : \mathbb{R}^3 \rightarrow \mathbb{R}^d$ is a learned linear projection mapping Lab differences to embedding dimension.

Interpolation consistency enforces that perceptual midpoints in Lab space map to embedding midpoints, ensuring smooth transitions:

$$\mathcal{L}_{\text{interp}} = \mathbb{E}(\mathbf{c}_i, \mathbf{c}_j) \sim \mathcal{P} \left[\left\| \phi\left(\frac{\mathbf{c}_i + \mathbf{c}_j}{2}\right) - \frac{1}{2}(\phi(\mathbf{c}_i) + \phi(\mathbf{c}_j)) \right\|_2^2 \right] \quad (6)$$

where $\mathbf{c}_i, \mathbf{c}_j$ are Lab color coordinates sampled from the anchor set and $\mathcal{P}$ denotes the distribution over color pairs. The left term computes the embedding of the Lab midpoint, while the right term computes the midpoint of two embeddings; minimizing their difference enforces linearity in the learned mapping.

The total objective combines reconstruction with geometric regularization:

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_d \mathcal{L}_{\text{dir}} + \lambda_i \mathcal{L}_{\text{interp}} \quad (7)$$

where λ_d and λ_i are hyperparameters to control relative contribution of regularizers.

4 Experiments

4.1 Experiment Setup

Implementation Details. We use FLUX.1-dev [33] as the primary backbone. In stage 1, the CTA is trained for 50K steps using AdamW [37] with learning rate 1×10^{-4} , batch size 128, and weight decay 0.01. In stage 2, the FLUX transformer and T5 encoder are frozen; only ColorBook embeddings $\mathbf{E} \in \mathbb{R}^{6707 \times 4096}$ are learned. ColorBook is trained for 100K steps with learning rate 1×10^{-5} and batch size 16 on $4 \times$ A100 64GB GPUs.

Baselines. We evaluate against T2I diffusion models spanning FLUX.1-dev [33], SD3 and SD3.5 [17], PixArt- α [8], PixArt- Σ [7], Sana [59], and CogView4 [13]. These baselines represent both single-encoder and multi-encoders, enabling analysis of how different text conditioning handle numeric colors. We integrate NumColor with aforementioned models to evaluate both trained performance and zero-shot transfer.

Evaluation Metrics. We adopt evaluation protocol from GenColorBench [5] benchmark, which measures color accuracy using ΔE (CIE2000)—perceptual color distance between specified and generated colors. Lower ΔE indicates better color fidelity, with $\Delta E < 1$ imperceptible to humans and $\Delta E < 5$ considered acceptable. We evaluate on three color granularities including ISCC-NBS Level 1, ISCC-NBS Level 3, and CSS3/X11 web colors. We use 9.8K numerical color prompts from GenColorBench covering 400+ colors, which evaluate hex code and RGB triples. Additionally, we compute color harmony metrics to assess the aesthetic coherence in the generated images.



Fig. 4: Qualitative results spanning *coarse colors* spanning primary hues; *fine-grained colors* including adjacent grayscale, blue-to-cyan, and red-to-yellow; and *interpolated colors* synthesized via interpolating anchors. NumColor maintains accurate color reproduction across objects.

4.2 Qualitative Results

Qualitative results show color control across three dimensions: (i) coarse colors spanning primary hues, (ii) fine-grained colors, and (iii) interpolated colors—which are not from NumColor anchors. Figure 4 shows our method with FLUX.1 accurately reproduce primary colors across diverse objects, confirming that color embeddings transfer independently of object category. We also show fine-grained color transitions, including grayscale—from black to white where adjacent swatches differ by $\Delta E < 15$. Simi-

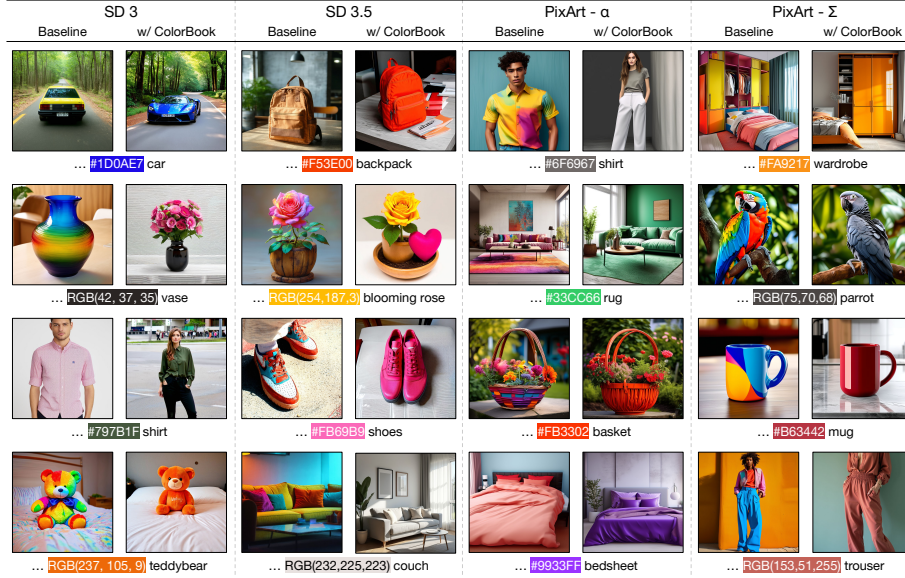


Fig. 5: Cross-model generalization. Baseline models versus NumColor integration across SD3, SD3.5, PixArt- α , and PixArt- Σ . NumColor, trained on FLUX, transfers zero-shot to other diffusion models. Baselines exhibit incorrect hue mapping, ignored specifications, and rainbow artifacts; NumColor resolves these while preserving image quality.

larly, we show color transition from blue to cyan, and red to yellow. Despite perceptual subtlety of these distinctions, our method maintains clear differentiation across color progression. Lastly, we demonstrate color generations that are not available as anchors in NumColor. These colors are synthesized through interpolating color anchors as described in Section 3.3. Accurate reproduction of these out-of-sample colors confirms that our learned embeddings are from a continuous, interpolable manifold.

In Figure 5, we show zero-shot cross-model of NumColor to SD3, SD3.5, PixArt- α , and PixArt- Σ through their shared T5 encoder. Baseline models exhibit characteristic failures. For instance, SD3 renders ■ `#1D0AE7` as yellow; SD3.5 ignores ■ `#F53E00` entirely, producing a beige backpack. With NumColor, all models correctly reproduce specified colors while maintaining generation quality. Challenging cases including near-neutral tones and desaturated hues are handled consistently across architectures. Further results including multi-object color generation are shown in supplements.

4.3 Embedding Space Analysis

Beyond evaluating generation quality, we analyze the geometric properties of the learned ColorBook to verify that training produces perceptually meaningful representations rather than arbitrary embeddings that happen to reconstruct colors (Fig. 6). We measure the L_2 drift between initial T5-encoded embeddings and their trained counterparts as a function of hue angle (Fig. 6b). The drift distributes uniformly across the hue spectrum,

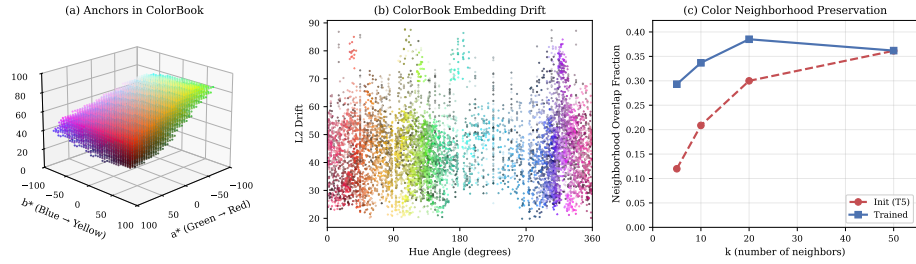


Fig. 6: Embedding space analysis of NumColor. (a) Distribution of 6,707 color anchors in CIE Lab space with 5-unit spacing. (b) L2 drift between initial (T5-encoded) and trained embeddings versus hue angle, showing uniform adaptation across the color spectrum. (c) Nearest Neighbor overlap between LAB and embedding space. Initial T5 embeddings show high consistency at small k due to shared category names but degrade at larger k ; trained embeddings maintain preservation at larger neighborhoods, reflecting perceptual structure beyond linguistic clustering.

indicating that training adapt to all color regions equally rather than overfitting to specific hues. We also observe the neighborhood preservation analysis (Figure 6c), which quantifies whether perceptual neighbors in Lab space remain neighbors in embedding space. Initial T5 embeddings exhibit high nearest neighbors overlap at small k , however this is an artifact of our ISCC-NBS initialization as colors within the same category share identical embeddings. Trained embeddings exhibit moderate overlap at small k while sustaining preservation at larger neighborhoods, confirming that training reorganizes the embedding space to reflect perceptual similarity rather than linguistic categorization. Moreover, learned geometric organization enables NumColor to generalize to colors unseen during training through smooth interpolation between anchors. This improvement is also reflected in Fig. 1 (right), where NumColor embeddings (stars) align with the Lab color structure approaches the consistency observed for color names.

4.4 Quantitative Comparisons

Table 1 presents numerical color understanding performance on GenColorBench benchmark across three granularities. We evaluate all models on 9.8K prompts using both hex and RGB formats. Baseline models fail regardless of granularity: FLUX achieves only 13.82% at ISCC-L1 despite just 13 coarse categories—marginally above the 7.7% random baseline—degrading to 7.74% at ISCC-L3 and 5.86% at CSS3/X11. This pattern persists across architectures, confirming that the bottleneck is not color discrimination but text encoders’ inability to extract meaning from tokenized color specifications. Whereas, NumColor integration yields substantial improvements across all models and granularities. On FLUX, accuracy increases from 13.82% to 55.71% at ISCC-L1, 7.74% to 48.02% at ISCC-L3, and 5.86% to 51.96% at CSS3/X11—representing 4–9 $\times$ relative improvement. Gains are largest at fine granularities where baselines approach random chance, demonstrating that learned embeddings preserve perceptual distinctions in Lab space. Cross-model transfer validates universality: using NumColor trained exclusively on FLUX, we observe average improvements of +37.4% on SD3,

Table 1: Numerical Color Understanding performance of T2I models on GenColorBench. Results across three color specification granularities.  indicate top-3 performers.

Model	Resolution	Type	GenColorBench Metrics $\uparrow$			Color Harmony $\downarrow$	
			ISCC-L1	ISCC-L3	CSS3/X11	RGB	Hex
FLUX	1024	DM	13.82	7.74	5.86	247.4	134.9
Sana	1024	DM	25.10	12.85	9.45	643.1	105.5
SD3.5	1024	DM	14.92	7.53	5.78	518.3	117.3
Pixart- α	1024	DM	9.45	5.42	4.21	638.9	366.1
SD3	1024	DM	11.80	5.92	4.63	826.4	251.4
Pixart- Σ	1024	DM	10.21	5.11	4.09	673.7	673.7
CogView4	1024	DM	10.95	5.32	4.25	476.9	253.0
FLUX w/ NumColor	1024	DM	55.71	48.02	51.96	16.90	18.47
SD 3 w/ NumColor	1024	DM	49.23	39.46	47.19	22.64	22.35
SD3.5 w/ NumColor	1024	DM	51.72	43.35	46.87	24.38	25.51
PixArt- α w/ NumColor	1024	DM	44.93	36.17	38.51	19.54	19.87
PixArt- Σ w/ NumColor	1024	DM	48.56	40.83	43.90	24.08	25.79

+36.8% on SD3.5, +33.5% on PixArt- α , and +37.3% on PixArt- Σ . Moreover, we evaluate FLUX.1 with and without NumColor on 180 ISCC-L1 GenColorBench prompts. FLUX.1 achieves 81.27% accuracy, and attains 80.04% w/ NumColor, indicating that NumColor preserves pretrained named-color understanding. We also evaluate more recent open-source and proprietary models including Qwen-Image, z-Image, FLUX.2, and GPT-Image-1.5, which we discuss in the Section S6 in the supplementary.

We additionally evaluate color harmony following Cohen-Or et al. [9], which measures conformance to harmonic templates on the hue wheel. Baseline models exhibit poor harmony scores such as 247–826 for RGB, and 105–673 for hex, as incorrect color reproduction often yields incoherent palettes. NumColor reduces these scores to 16–24 for RGB and 18–26 for hex which yields a 10–30 $\times$ improvement demonstrating that precise color control simultaneously enhances aesthetic coherence.

4.5 Ablations

We conduct ablation studies to validate design choices, examining cross-attention behavior after NumColor integration and hyperparameter sensitivity. Figure 7 visualizes cross-attention maps comparing baseline models with NumColor. Without NumColor, both FLUX and SD 3.5 fragment hex codes into individual character tokens (Figure 1), resulting in scattered attention patterns across unrelated image regions. With NumColor, color specification is processed as a unified semantic unit, concentrating attention on the target object. This consolidation enables precise color-to-object binding. Further, we analyze effect of softmax temperature τ and color neighborhood size on the color generation, shown in Figure 8. The temperature parameter controls the sharpness of interpolation weights over k-nearest anchors. At low τ , weights concentrate on closest anchors, while high τ distributes weights more uniformly, causing color desaturation. At k=1, hard assignment to the nearest anchor produces quantization artifacts. We choose $\tau=2.0$ and k=8, balancing accurate color reproduction with smooth interpolation. We further ablate the auxiliary loss functions, shown in Table S2 in the supplementary material, to demonstrate the effectiveness of the proposed loss functions.



Fig. 7: Cross-attention maps for hex color specifications. Baseline models fragment hex codes into character tokens with scattered attention. NumColor processes colors as unified embeddings, focusing attention on target objects and enabling accurate color control.

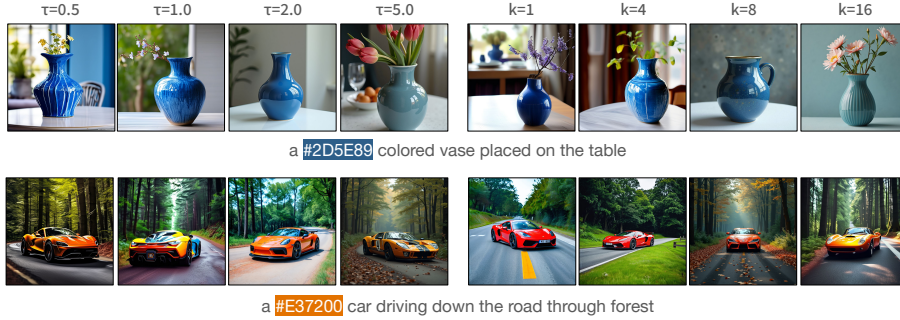


Fig. 8: Hyperparameter ablation. Low k causes hard quantization artifacts. High τ leads to color desaturation. $\tau = 2.0$, $k = 8$ balance accurate color reproduction with smooth interpolation.

5 Limitations

While our method enables color control, we identify two failure modes inherent to underlying diffusion models (Figure 9). *Color leakage* occurs when the color bleeds into surroundings as cross-attention lacks spatial grounding to enforce boundaries. *Memory color bias* causes objects with strong canonical associations to resist recoloring; despite specifying green for sunflowers, model defaults to memorized yellow, reflecting object-color priors that override color embedding. Both limitations represent fundamental challenges in controlling pretrained generative models, not specific to NumColor.



Fig. 9: Failure cases. *Color Leakage*: Specified color appears on the object but bleeds into surroundings due to lack of spatial grounding. *Memory Color Bias*: Objects with canonical binding resist recoloring, as pretrained model overrides the embedding, defaulting to memorized colors.

6 Conclusion

We presented NumColor, a plug-and-play framework for precise numeric color control in T2I generation. NumColor addresses two compounding failures: tokenizer fragmentation via a tokenizer-agnostic Color Token Aggregator, and the alignment gap between color specifications and embedding space via ColorBook operating in perceptually uniform CIE LAB space. Both components require no adaptation to pretrained diffusion models. To enable training, we proposed NumColor-Data, a synthetic dataset of 500K images with unambiguous color-to-pixel correspondence. The trained ColorBook transfers zero-shot to multiple diffusion models, achieving $4 - 9\times$ improvement in color accuracy and $10 - 30\times$ improvement in color harmony on GenColorBench benchmark.

Acknowledgments

This work was supported by Grants PID2022-143257NB-I00, PID2024-162555OB-I00, and AIA2025-163919-C52 funded by MICIU/AEI/10.13039/501100011033, ERDF/EU and FEDER. Also, Generalitat de Catalunya CERCA Program, and Càtedra ENIA UAB-Cruïlla (TSI-100929-2023-2) from Ministry of Economic Affairs and Digital Transition of Spain, and also European Union’s Horizon Europe research and innovation programme under grant agreement number 101214398 (ELLIOT), and by BBVA Foundations of Science program on Mathematics, Statistics, Computational Sciences and Artificial Intelligence (grant VIS4NN). The BBVA Foundation accepts no responsibility for the opinions, statements and contents included in the project and/or the results thereof, which are entirely the responsibility of the authors. Kai Wang acknowledges funding No. R002026G0116 and No. R002026B0121 from Guangdong and Hong Kong Universities 1+1+1 Joint Research Collaboration Scheme, and start-up grant B01040000108 from CityU-DG. We acknowledge EuroHPC Joint Undertaking for awarding us access to Leonardo at CINECA, Italy, and RES resources provided by BSC on MareNostrum5 under projects IM-2025-3-0025 and IM-2025-3-0027.

References

1. 3D assets | BlenderKit, <https://www.blenderkit.com/>
2. 3D assets | Poly Haven, <https://polyhaven.com/>
3. Berlin, B., Kay, P.: Basic color terms: Their universality and evolution. Univ of California Press (1991)
4. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR (2023)
5. Butt, M.A., Gomez-Villa, A., Wu, T., Vazquez-Corral, J., Van De Weijer, J., Wang, K.: Gen-colorbench: A color evaluation benchmark for text-to-image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 36638–36648 (2026)
6. Butt, M.A., Wang, K., Vazquez-Corral, J., Van de Weijer, J.: Colorpeel: Color prompt learning with diffusion models via color and shape disentanglement. In: European Conference on Computer Vision. pp. 456–472. Springer (2024)
7. Chen, J., Ge, C., Xie, E., Wu, Y., Yao, L., Ren, X., Wang, Z., Luo, P., Lu, H., Li, Z.: Pixart- σ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation. arXiv preprint arXiv:2403.04692 (2024)
8. Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., Li, Z.: Pixart- α : Fast training of diffusion transformer for photorealistic text-to-image synthesis. arXiv preprint arXiv:2310.00426 (2023)
9. Cohen-Or, D., Sorkine, O., Gal, R., Levvand, T., Xu, Y.Q.: Color harmonization. In: ACM SIGGRAPH 2006 Papers, pp. 624–630 (2006)
10. Dai, X., Hou, J., Ma, C.Y., Tsai, S., Wang, J., Wang, R., Zhang, P., Vandenhende, S., Wang, X., Dubey, A., et al.: Emu: Enhancing image generation models using photogenic needles in a haystack. arXiv preprint arXiv:2309.15807 (2023)
11. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-xl: A universe of 10m+ 3d objects. In: NeurIPS (2023)
12. Deng, C., Zhu, D., Li, K., Gou, C., Li, F., Wang, Z., Zhong, S., Yu, W., Nie, X., Song, Z., et al.: Emerging properties in unified multimodal pretraining. arXiv preprint arXiv:2505.14683 (2025)
13. Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., et al.: Cogview: Mastering text-to-image generation via transformers. In: NeurIPS (2021)
14. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. In: ICLR workshop (2015)
15. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. In: ICLR (2017)
16. Dong, Z., Wei, P., Lin, L.: Dreamartist: Towards controllable one-shot text-to-image generation via contrastive prompt-tuning. arXiv preprint arXiv:2211.11337 (2022)
17. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024)
18. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR (2021)
19. Fairchild, M.D.: Color Appearance Models. The Wiley-IS&T Series in Imaging Science and Technology, John Wiley & Sons, 3rd edn. (2013)
20. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. In: ICLR (2023)

21. Ge, S., Park, T., Zhu, J.Y., Huang, J.B.: Expressive text-to-image generation with rich text. In: ICCV. pp. 7545–7556 (2023)
22. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. *Communications of the ACM* **63**(11), 139–144 (2020)
23. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. In: CVPR (2022)
24. Hertz, A., Aberman, K., Cohen-Or, D.: Delta denoising score. *arXiv preprint arXiv:2304.07090* (2023)
25. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. In: ICLR (2023)
26. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
27. Hou, Y., Zeng, X., Wang, Y., Yang, M., Chen, X., Zeng, W.: Gencolor: Generative color-concept association in visual design. In: CHI Conference on Human Factors in Computing Systems (2025)
28. Kavar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. CVPR (2023)
29. Kelly, K.L., Judd, D.B.: Color: universal language and dictionary of names, vol. 440. US Department of Commerce, National Bureau of Standards (1976)
30. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013)
31. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: ICML (2019)
32. Kumari, N., Zhang, B., Zhang, R., Shechtman, E., Zhu, J.Y.: Multi-concept customization of text-to-image diffusion. CVPR (2023)
33. Labs, B.F.: Flux. <https://github.com/black-forest-labs/flux> (2024)
34. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML (2001)
35. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023)
36. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: ICLR (2023)
37. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
38. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: ICLR (2022)
39. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: CVPR (2023)
40. Morović, J., Morović, P.: Llms speak lab. In: Color and Imaging Conference (2025)
41. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021)
42. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV (2023)
43. Qin, J., Gomez-Villa, A., Li, S., Yang, S., Wang, Y., Wang, K., van de Weijer, J.: Free-lunch color-texture disentanglement for stylized image generation. In: NeurIPS (2025)
44. Qiu, Q., Mao, J., Wang, X.: Exploring palette based color guidance in diffusion models. In: ACM MM (2025)
45. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)

46. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**(140), 1–67 (2020)
47. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022)
48. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: *ICML* (2021)
49. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: *CVPR* (2023)
50. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., et al.: Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* (2022)
51. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C.W., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S.R., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: LAION-5b: An open large-scale dataset for training next generation image-text models. In: *NeurIPS* (2022)
52. Shao, C., Ma, Z., Feng, Y.: Viterbi decoding of directed acyclic transformer for non-autoregressive machine translation. In: *Findings of the Association for Computational Linguistics: EMNLP* (2022)
53. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: *ICLR* (2021)
54. Tsai, S.L., Huang, B.L., Shen, Y.T., Yeo, C.Y., Tseng, C., Ruan, B.K., Lien, W.S., Shuai, H.H.: Color me correctly: Bridging perceptual color spaces and text embeddings for improved diffusion generation. In: *ACM MM* (2025)
55. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. In: *NeurIPS* (2017)
56. Wang, K., Yang, F., Raducanu, B., van de Weijer, J.: Multi-class textual-inversion secretly yields a semantic-agnostic classifier. In: *WACV* (2025)
57. Wang, Z.J., Montoya, E., Munechika, D., Yang, H., Hoover, B., Chau, D.H.: Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. In: *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*. pp. 893–911 (2023)
58. Wu, C., Zheng, P., Yan, R., Xiao, S., Luo, X., Wang, Y., Li, W., Jiang, X., Liu, Y., Zhou, J., et al.: Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871* (2025)
59. Xie, E., Chen, J., Chen, J., Cai, H., Tang, H., Lin, Y., Zhang, Z., Li, M., Zhu, L., Lu, Y., Han, S.: Sana: Efficient high-resolution image synthesis with linear diffusion transformer. *arXiv preprint arxiv:2410.10629* (2024)
60. Xu, S., Huang, Y., Pan, J., Ma, Z., Chai, J.: Inversion-free image editing with language-guided diffusion models. In: *CVPR* (2024)
61. Yang, Y., Wu, T., Li, S., Yang, S., Wang, Y., van de Weijer, J., Wang, K.: Echodistill: Bidirectional concept distillation for one-step diffusion personalization. *arXiv preprint arXiv:2510.20512* (2025)