Expressive Text-to-Image Generation with Rich Text

Songwei Ge[1] Taesung Park[2] Jun-Yan Zhu[3] Jia-Bin Huang[1]
[1]University of Maryland, College Park [2]Adobe Research [3]Carnegie Mellon University

https://rich-text-to-image.github.io/

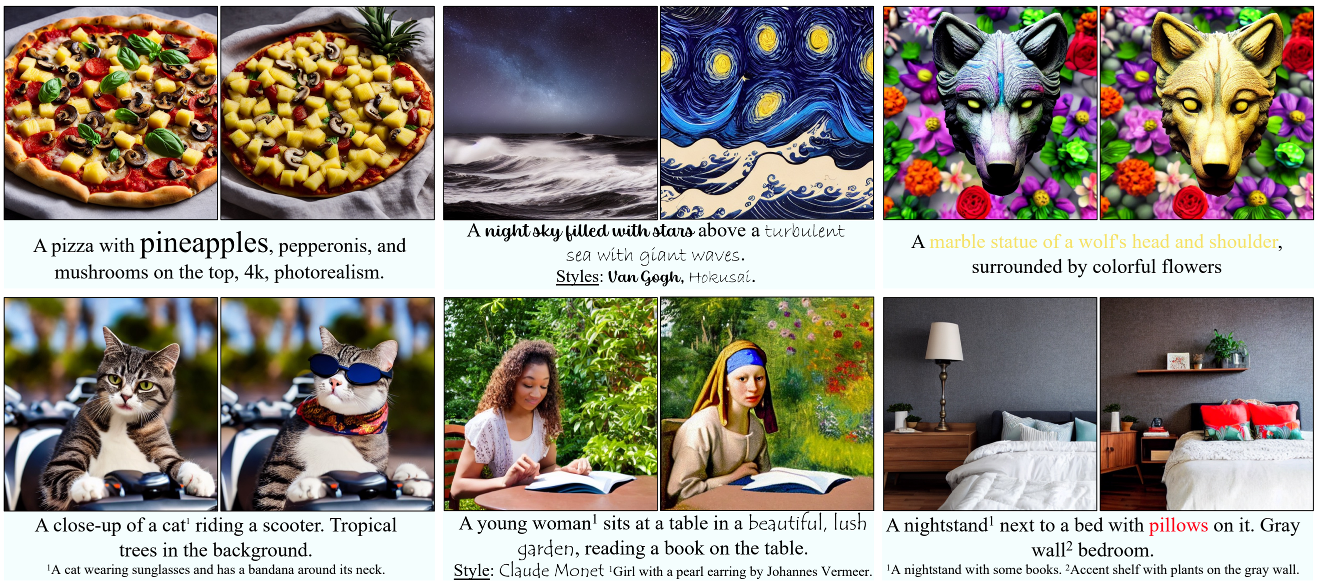

Figure 1. **Plain text (left image) vs. Rich text (right image).** Our method allows a user to describe an image using a rich text editor that supports various text attributes such as font family, size, color, and footnote. Given these text attributes extracted from rich text prompts, our method enables precise control of text-to-image synthesis regarding colors, styles, and object details compared to plain text.

Abstract

Plain text has become a prevalent interface for text-to-image synthesis. However, its limited customization options hinder users from accurately describing desired outputs. For example, plain text makes it hard to specify continuous quantities, such as the precise RGB color value or importance of each word. Furthermore, creating detailed text prompts for complex scenes is tedious for humans to write and challenging for text encoders to interpret. To address these challenges, we propose using a rich-text editor supporting formats such as font style, size, color, and footnote. We extract each word's attributes from rich text to enable local style control, explicit token reweighting, precise color rendering, and detailed region synthesis. We achieve these capabilities through a region-based diffusion process. We first obtain each word's region based on attention maps of a diffusion process using plain text. For each region, we enforce its text attributes by creating region-specific detailed prompts and applying region-specific guidance, and maintain its fidelity against plain-text generation through region-based injections. We present various examples of image generation from rich text and demonstrate that our method outperforms strong baselines with quantitative evaluations.

1. Introduction

The development of large-scale text-to-image generative models [52, 56, 54, 28] has propelled image generation to an unprecedented era. The great flexibility of these large-scale models further offers users powerful control of the generation through visual cues [4, 17, 77] and textual inputs [7, 19]. Without exception, existing studies use *plain text* encoded by a pretrained language model to guide the generation. However, in our daily life, it is rare to use only plain text when working on text-based tasks such as writing blogs or editing essays. Instead, a *rich text* editor [68, 71] is the more popular choice providing versatile formatting options for writing and editing text. In this paper, we seek to introduce accessible and precise textual control from rich text editors to text-to-image synthesis.

Rich text editors offer unique solutions for incorporating conditional information separate from the text. For example, using the font color, one can indicate an *arbitrary* color. In contrast, describing the precise color with plain text proves more challenging as general text encoders do not understand RGB or Hex triplets, and many color names, such as 'olive' and 'orange', have ambiguous meanings. This font color information can be used to define the color of generated objects. For example, in Figure 1, a specific yellow can be selected to instruct the generation of a marble statue with that exact color.

Beyond providing precise color information, various font formats make it simple to augment the word-level information. For example, reweighting token influence [19] can be implemented using the font size, a task that is difficult to achieve with existing visual or textual interfaces. Rich text editors offer more options than font size – similar to how font style distinguishes the styles of individual text elements, we propose using it to capture the artistic style of specific regions. Another option is using footnotes to provide supplementary descriptions for selected words, simplifying the process of creating complex scenes.

But how can we use rich text? A straightforward implementation is to convert a rich-text prompt with detailed attributes into lengthy plain text and feed it directly into existing methods [54, 19, 7]. Unfortunately, these methods struggle to synthesize images corresponding to lengthy text prompts involving multiple objects with distinct visual attributes, as noted in a recent study [12]. They often mix styles and colors, applying a uniform style to the entire image. Furthermore, the lengthy prompt introduces extra difficulty for text encoders to interpret accurate information, making generating intricate details more demanding.

To address these challenges, our insight is to decompose a rich-text prompt into two components (1) a short plain-text prompt (without formatting) and (2) multiple region-specific prompts that include text attributes, as shown in Figure 2. First, we obtain the self- and cross-attention maps using a vanilla denoising process with the short *plain-text* prompt to associate each word with a specific region. Second, we create a prompt for each region using the attributes derived from *rich-text* prompt. For example, we use "mountain in the style of Ukiyo-e" as the prompt for the region corresponding to the word "mountain" with the attribute "font style: Ukiyo-e". For RGB font colors that cannot be converted to the prompts, we iteratively update the region with region-based guidance to match the target color. We apply a separate denoising process for each region and fuse the predicted noises to get the final update. During this process, regions associated with the tokens that do not have any formats are supposed to look the same as the plain-text results. Also, the overall shape of the objects should stay unchanged in cases such as only the color is changed. To this end, we propose to use region-based injection approaches.

We demonstrate qualitatively and quantitatively that our method generates more precise color, distinct styles, and accurate details compared to plain text-based methods.

2. Related Work

Text-to-image models. Text-to-image systems aim to synthesize realistic images according to descriptions [82, 42]. Fueled by the large-scale text-image datasets [60, 8], various training and inference techniques [20, 62, 21, 22], and scalibility [51], significant progress has been made in text-to-image generation using diffusion models [4, 51, 45, 56, 17], autoregressive models [52, 76, 11, 15], GANs [59, 28], and their hybrids [54]. Our work focuses on making these models more accessible and providing precise controls. In contrast to existing work that uses *plain text*, we use a *rich text* editor with various formatting options.

Controllable image synthesis with diffusion models. A wide range of image generation and editing applications are achieved through either fine-tuning pre-trained diffusion models [55, 32, 77, 3, 72, 30, 41, 35] or modifying the denoising process [43, 13, 19, 46, 5, 12, 2, 4, 26, 6, 58, 78, 9, 48, 48, 73, 16]. For example, Prompt-to-prompt [19] uses attention maps from the original prompt to guide the spatial structure of the target prompt. Although these methods can be applied to some rich-text-to-image applications, the results often fall short, as shown in Section 4. Concurrent with our work, Mixture-of-diffusion [26] and MultiDiffusion [6] propose merging multiple diffusion-denoising processes in different image regions through linear blending. Instead of relying on user-provided regions, we automatically compute regions of selected tokens using attention maps. Gradient [24] and Universal [5] guidance control the generation by optimizing the denoised generation at each time step. We apply them to precise color generation by designing an objective on the target region to be optimized.

Attention in diffusion models. The attention mechanism has been used in various diffusion-based applications such as view synthesis [37, 66, 70], image editing [19, 12, 47, 46, 32], and video editing [38, 49, 10, 40]. We also leverage the spatial structure in self-attention maps and alignment information between texts and regions in cross-attention maps for rich-text-to-image generation.

Rich text modeling and application. Exploiting information beyond the intrinsic meanings of the texts has been previously studied [44, 63, 75, 34]. For example, visual information, such as underlining and bold type, have also been extracted for various document understanding tasks [75, 34]. To our knowledge, we are the first to leverage rich text information for text-to-image synthesis.

Image stylization and colorization. Style transfer [18, 81, 39] and Colorization [53, 64, 74, 33, 79, 80] for *editing real images* have also been extensively studied. In contrast, our work focuses on local style and precise color control for *generating images* from text-to-image models.

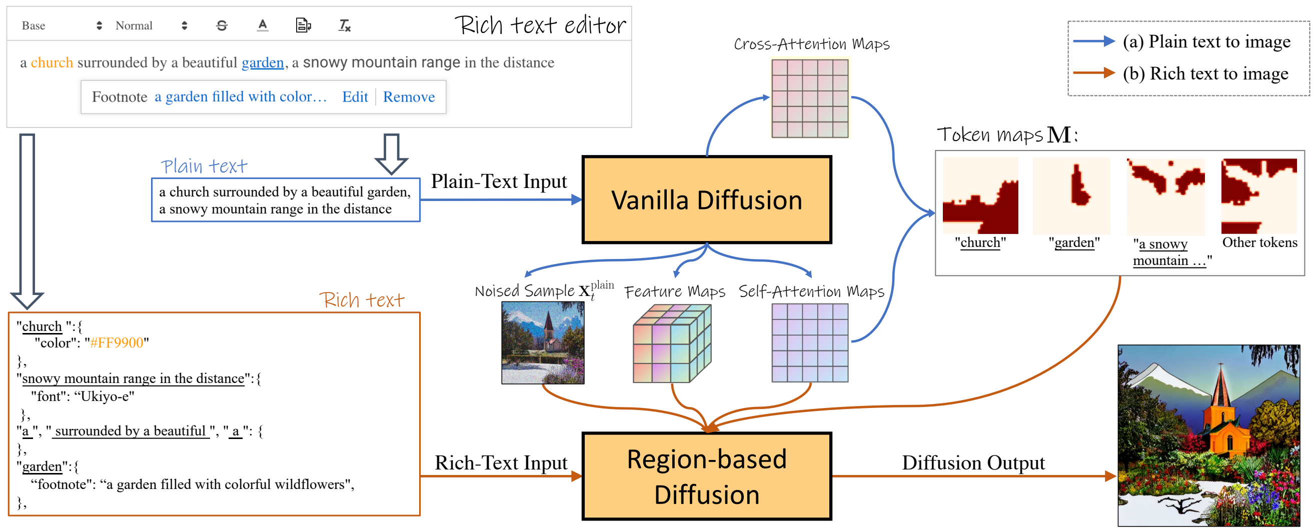

Figure 2. **Rich-text-to-image framework.** First, the plain-text prompt is processed by a diffusion model to collect self- and cross-attention maps, noised generation, and residual feature maps at certain steps. The token maps of the input prompt are constructed by first creating a segmentation using the self-attention maps and then labeling each segment using the cross-attention maps. Then the rich texts are processed as JSON to provide attributes for each token span. The resulting token maps and attributes are used to guide our region-based control. We inject the self-attention maps, noised generation, and feature maps to improve fidelity to the plain-text generation.

3. Rich Text to Image Generation

From writing messages on communication apps, designing websites [57], to collaboratively editing a document [36, 25], a rich text editor is often the primary interface to edit texts on digital devices. Nonetheless, only plain text has been used in text-to-image generation. To use formatting options in rich-text editors for more precise control over the black-box generation process [1], we first introduce a problem setting called *rich-text-to-image generation*. We then discuss our approach to this task.

3.1. Problem Setting

As shown in Figure 2, a rich text editor supports various formatting options, such as font styles, font size, color, and more. We leverage these text attributes as extra information to increase control of text-to-image generation. We interpret the rich-text prompt as JSON, where each text element consists of a span of tokens $\boldsymbol{e}_i$ (e.g., 'church') and attributes $\boldsymbol{a}_i$ describing the span (e.g., 'color:#FF9900'). Note that some tokens $\boldsymbol{e}_U$ may not have any attributes. Using these annotated prompts, we explore four applications: 1) local style control using *font style*, 2) precise color control using *font color*, 3) detailed region description using *footnotes*, and 4) explicit token reweighting with *font sizes*.

Font style is used to apply a specific artistic style $\boldsymbol{a}_i^s$, e.g., $\boldsymbol{a}_i^s$ = 'Ukiyo-e', to the synthesis of the span of tokens $\boldsymbol{e}_i$. For instance, in Figure 1, we apply the Ukiyo-e painting style to the ocean waves and the style of Van Gogh to the sky, enabling the application of localized artistic styles. This task presents a unique challenge for existing text-to-image models, as there are limited training images featuring multiple artistic styles. Consequently, existing models tend to generate a *uniform* mixed style across the entire image rather than distinct local styles.

Font color indicates a specific color of the modified text span. Given the prompt "a red toy", the existing text-to-image models generate toys in various shades of red, such as light red, crimson, or maroon. The color attribute provides a way for specifying a *precise color* in the RGB color space, denoted as $\boldsymbol{a}_i^c$. For example, to generate a toy in fire brick red, one can change the font color to "a toy", where the word "toy" is associated with the attribute $\boldsymbol{a}_i^c = [178, 34, 34]$. However, as shown in the experiment section, the pretrained text encoder cannot interpret the RGB values and have difficulty understanding obscure color names, such as lime and orange.

Footnote provides supplementary explanations of the target span without hindering readability with lengthy sentences. Writing detailed descriptions of complex scenes is tedious work, and it inevitably creates lengthy prompts [29, 27]. Additionally, existing text-to-image models are prone to ignoring some objects when multiple objects are present [12], especially with long prompts. Moreover, excess tokens are discarded when the prompt's length surpasses the text encoder's maximum length, e.g., 77 tokens for CLIP models [50]. We aim to mitigate these issues using a footnote string $\boldsymbol{a}_i^f$.

Font size can be employed to indicate the importance, quantity, or size of an object. We use a scalar $\boldsymbol{a}_i^w$ to denote the weight of each token.

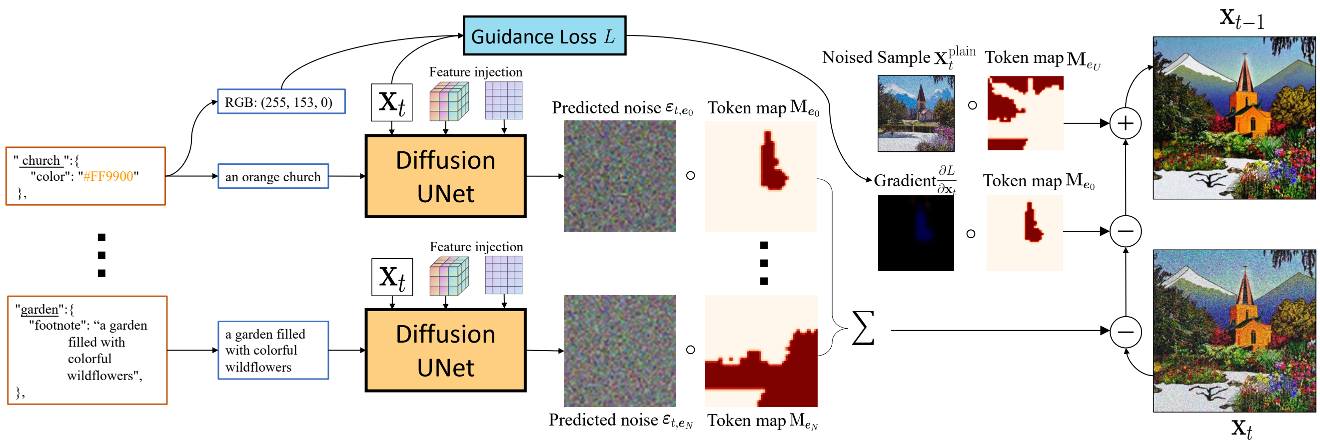

Figure 3. **Region-based diffusion.** For each element of the rich-text input, we apply a separate diffusion process to its region. The attributes are either decoded as a region-based guidance target (e.g. re-coloring the church), or as a textual input to the diffusion UNet (e.g. handling the footnote to the garden). The self-attention maps and feature maps extracted from the plain-text generation process are injected to help preserve the structure. The predicted noise $\epsilon_{t,\boldsymbol{e}_i}$, weighted by the token map $\mathbf{M}_{\boldsymbol{e}_i}$, and the guidance gradient $\frac{\partial L}{\partial \mathbf{x}_t}$ are used to denoise and update the previous generation $\mathbf{x}_t$ to $\mathbf{x}_{t-1}$. The noised plain text generation $\mathbf{x}_t^{\text{plain}}$ is blended with the current generation to preserve the exact content in those regions of the unformatted tokens.

3.2. Method

To utilize rich text annotations, our method consists of two steps, as shown in Figure 2. First, we compute the spatial layouts of individual token spans. Second, we use a new region-based diffusion to render each region's attributes into a globally coherent image.

Step 1. Token maps for spatial layout. Several works [65, 40, 4, 19, 12, 47, 67] have discovered that the attention maps in the self- and cross-attention layers of the diffusion UNet characterize the spatial layout of the generation. Therefore, we first use the plain text as the input to the diffusion model and collect self-attention maps of size $32 \times 32 \times 32 \times 32$ across different heads, layers, and time steps. We take the average across all the extracted maps and reshape the result into 1024×1024. Note that the value at i^{th} row and j^{th} column of the map indicates the probability of pixel i attending to pixel j. We average the map with its transpose to convert it to a symmetric matrix. It is used as a similarity map to perform spectral clustering [61, 69] and obtain the binary segmentation maps $\widehat{\mathbf{M}}$ of size $K \times 32 \times 32$, where K is the number of segments.

To associate each segment with a textual span, we also extract cross-attention maps for each token w_j:

$$\mathbf{m}_j = \frac{\exp(\mathbf{s}_j)}{\sum_k \exp(\mathbf{s}_k)}, \tag{1}$$

where $\mathbf{s}_j$ is the attention score. We first interpolate each cross-attention map $\mathbf{m}_j$ to the same resolution as $\widehat{\mathbf{M}}$ of 32×32. Similar to the processing steps of the self-attention maps, we compute the mean across heads, layers, and time steps to get the averaged map $\widehat{\mathbf{m}}_j$. We associate each segment with a texture span $\boldsymbol{e}_i$ following Patashnik et al. [47]:

$$\mathbb{M}_{\boldsymbol{e}_i} = \{\widehat{\mathbf{M}}_k \mid \left| \widehat{\mathbf{M}}_k \cdot \frac{\widehat{\mathbf{m}}_j - \min(\widehat{\mathbf{m}}_j)}{\max(\widehat{\mathbf{m}}_j) - \min(\widehat{\mathbf{m}}_j)} \right|_1 > \epsilon,\ \forall j\ s.t.\ w_j \in \boldsymbol{e}_i\}, \tag{2}$$

where ϵ is a hyperparameter that controls the labeling threshold, that is, the segment $\widehat{\mathbf{M}}_k$ is assigned to the span $\boldsymbol{e}_i$ if the normalized attention score of any tokens in this span is higher than ϵ. We associate the segments that are not assigned to any formatted spans with the unformatted tokens $\boldsymbol{e}_U$. Finally, we obtain the *token map* in Figure 2 as below:

$$\mathbf{M}_{\boldsymbol{e}_i} = \frac{\sum_{\widehat{\mathbf{M}}_j \in \mathbb{M}_{\boldsymbol{e}_i}} \widehat{\mathbf{M}}_j}{\sum_i \sum_{\widehat{\mathbf{M}}_j \in \mathbb{M}_{\boldsymbol{e}_i}} \widehat{\mathbf{M}}_j} \tag{3}$$

Step 2. Region-based denoising and guidance. As shown in Figure 2, given the text attributes and *token maps*, we divide the overall image synthesis into several region-based denoising and guidance processes to incorporate each attribute, similar to an ensemble of diffusion models [32, 6]. More specificially, given the span $\boldsymbol{e}_i$, the region defined by its *token map* $\mathbf{M}_{\boldsymbol{e}_i}$, and the attribute $\boldsymbol{a}_i$, the predicted noise $\boldsymbol{\epsilon}_t$ for noised generation $\mathbf{x}_t$ at time step t is

$$\boldsymbol{\epsilon}_t = \sum_i \mathbf{M}_{\boldsymbol{e}_i} \cdot \epsilon_{t,\boldsymbol{e}_i} = \sum_i \mathbf{M}_{\boldsymbol{e}_i} \cdot D(\mathbf{x}_t, f(\boldsymbol{e}_i, \boldsymbol{a}_i), t), \tag{4}$$

where D is the pretrained diffusion model, and $f(\boldsymbol{e}_i, \boldsymbol{a}_i)$ is a plain text representation derived from text span $\boldsymbol{e}_i$ and attributes a_i using the following process:

1. Initially, we set $f(\boldsymbol{e}_i, \boldsymbol{a}_i) = \boldsymbol{e}_i$.
2. If footnote $\boldsymbol{a}_i^f$ is available, we set $f(\boldsymbol{e}_i, \boldsymbol{a}_i) = \boldsymbol{a}_i^f$.

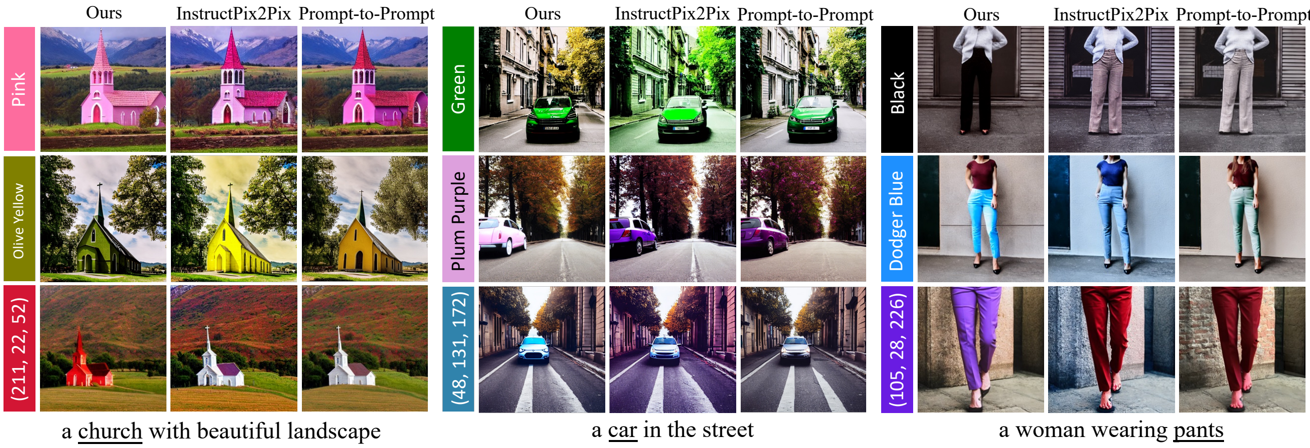

Figure 4. **Qualitative comparison on precise color generation.** We show images generated by Prompt-to-Prompt [19], InstructPix2Pix [7], and our method using prompts with font colors. Our method generates precise colors according to either color names or RGB values. Both baselines generate plausible but inaccurate colors given color names, while neither understands the color defined by RGB values. InstructPix2Pix tends to apply the color globally, even outside the target object.

3. The style $\boldsymbol{a}_i^s$ is appended if it exists. $f(\boldsymbol{e}_i, \boldsymbol{a}_i) = f(\boldsymbol{e}_i, \boldsymbol{a}_i) +$ 'in the style of' $+ \boldsymbol{a}_i^s$.
4. The closest color name (string) of font color $\hat{\boldsymbol{a}}_i^c$ from a predefined set $\mathcal{C}$ is prepended. $f(\boldsymbol{e}_i, \boldsymbol{a}_i) = \hat{\boldsymbol{a}}_i^c + f(\boldsymbol{e}_i, \boldsymbol{a}_i)$. For example, $\hat{\boldsymbol{a}}_i^c$ = 'brown' for RGB color $\boldsymbol{a}_i^c$ = [136,68,20].

We use $f(\boldsymbol{e}_i, \boldsymbol{a}_i)$ as the original plain text prompt of Step 1 for the unformatted tokens $\boldsymbol{e}_U$. This helps us generate a coherent image, especially around region boundaries.

Guidance. By default, we use classifier-free guidance [23] for each region to better match the prompt $f(\boldsymbol{e}_i, \boldsymbol{a}_i)$. In addition, if the font color is specified, to exploit the RGB values information further, we apply gradient guidance [24, 14, 5] on the current clean image prediction:

$$\widehat{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_t}{\sqrt{\bar{\alpha}_t}}, \tag{5}$$

where $\mathbf{x}_t$ is the noisy image at time step t, and $\bar{\alpha}_t$ is the coefficient defined by noise scheduling strategy [20]. Here, we compute an MSE loss $\mathcal{L}$ between the average color of $\widehat{\mathbf{x}}$ weighted by the *token map* $\mathbf{M}_{\boldsymbol{e}_i}$ and the RGB triplet $\boldsymbol{a}_i^c$. The gradient is calculated below,

$$\frac{d\mathcal{L}}{d\mathbf{x}_t} = \frac{d\|\sum_p (\mathbf{M}_{\boldsymbol{e}_i} \cdot \widehat{\mathbf{x}}_0)/\sum_p \mathbf{M}_{\boldsymbol{e}_i} - \boldsymbol{a}_i^c\|_2^2}{\sqrt{\bar{\alpha}_t} d\widehat{\mathbf{x}}_0}, \tag{6}$$

where the summation is over all pixels p. We then update $\mathbf{x}_t$ with the following equation:

$$\mathbf{x}_t \leftarrow \mathbf{x}_t - \lambda \cdot \mathbf{M}_{\boldsymbol{e}_i} \cdot \frac{d\mathcal{L}}{d\mathbf{x}_t}, \tag{7}$$

where λ is a hyperparameter to control the strength of the guidance. We use $\lambda = 1$ unless denoted otherwise.

Token reweighting with font size. Last, to re-weight the impact of the token w_j according to the font size $\boldsymbol{a}_j^w$, we modify its cross-attention maps $\mathbf{m}_j$. However, instead of applying direct multiplication as in Prompt-to-Prompt [19] where $\sum_j \boldsymbol{a}_j^w \mathbf{m}_j \neq 1$, we find that it is critical to preserve the probability property of $\mathbf{m}_j$. We thus propose the following reweighting approach:

$$\widehat{\mathbf{m}}_j = \frac{\boldsymbol{a}_j^w \exp(\mathbf{s}_j)}{\sum_k \boldsymbol{a}_k^w \exp(\mathbf{s}_k)}. \tag{8}$$

We can compute the token map (Equation 3) and predict the noise (Equation 4) with the reweighted attention map.

Preserve the fidelity against plain-text generation. Although our region-based method naturally maintains the layout, there is no guarantee that the details and shape of the objects are retained when no rich-text attributes or only the color is specified, as shown in Figure 12. To this end, we follow Plug-and-Play [67] to inject the self-attention maps and the residual features extracted from the plain-text generation process when $t > T_{\text{pnp}}$ to improve the structure fidelity. In addition, for the regions associated with the unformatted tokens $\boldsymbol{e}_U$, stronger content preservation is desired. Therefore, at certain $t = T_{\text{blend}}$, we blend the noised sample $\mathbf{x}_t^{\text{plain}}$ based on the plain text into those regions:

$$\mathbf{x}_t \leftarrow \mathbf{M}_{\boldsymbol{e}_U} \cdot \mathbf{x}_t^{\text{plain}} + (1 - \mathbf{M}_{\boldsymbol{e}_U}) \cdot \mathbf{x}_t \tag{9}$$

4. Experimental Results

Implementation details. We use Stable Diffusion V1-5 [54] for our experiments. To create the token maps, we use the cross-attention layers in all blocks, excluding the first encoder and last decoder blocks, as the attention maps in these high-resolution layers are often noisy. We discard

Figure 5. **Qualitative comparison on style control.** We show images generated by Prompt-to-Prompt, InstructPix2Pix, and our method using prompts with multiple styles. Only our method can generate distinct styles for both regions.

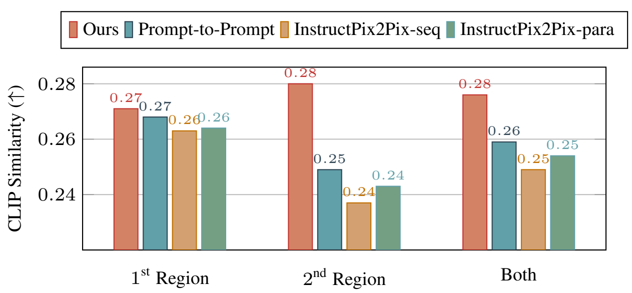

Figure 6. **Quantitative evaluation of local style control.** We report the CLIP similarity between each stylized region and its region prompt. Our method achieves the best stylization.

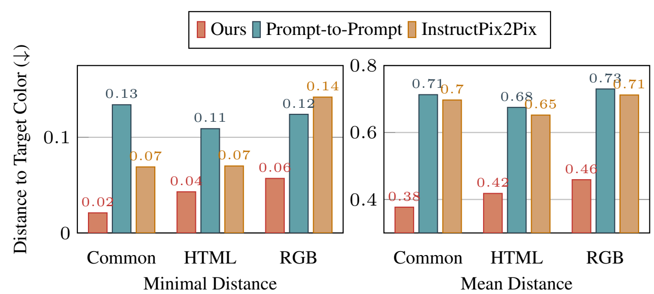

Figure 7. **Quantitaive evaluation on precise color generation.** Distance against target color is reported (lower is better). Our method consistently outperforms baselines.

the maps at the initial denoising steps with $T > 750$. We use $K = 15, \epsilon = 0.3, T_{\text{pnp}} = 0.3, T_{\text{blend}} = 0.3$, and report the results averaged from three random seeds for all quantitative experiments. More details, such as the running time, can be found in Appendix B.

Font style evaluation. We compute CLIP scores [50] for each local region to evaluate the stylization quality. Specifically, we create prompts of two objects and styles. We create combinations using 7 popular styles and 10 objects, resulting in 420 prompts. For each generated image, we mask it by the token maps of each object and attach the masked output to a black background. Then, we compute the CLIP score using the region-specific prompt. For example, for the prompt "a lighthouse (Cyberpunk) among the turbulent waves (Ukiyo-e)", the local CLIP score of the lighthouse region is measured by comparing its similarity with the prompt "lighthouse in the style of cyberpunk." We refer to "lighthouse" as the first region and "waves" as the second region in this example.

Font color evaluation. To evaluate a method's capacity to understand and generate a specific color, we divide colors into three categories. The *Common color* category contains 17 standard names, such as "red", "yellow", and "pink". The *HTML color* names are selected from the web color names [1] used for website design, such as "sky blue", "lime green", and "violet purple". The *RGB color* category contains 50 randomly sampled RGB triplets to be used as "color of RGB values $[128, 128, 128]$". To create a complete

[1] https://simple.wikipedia.org/wiki/Web_color

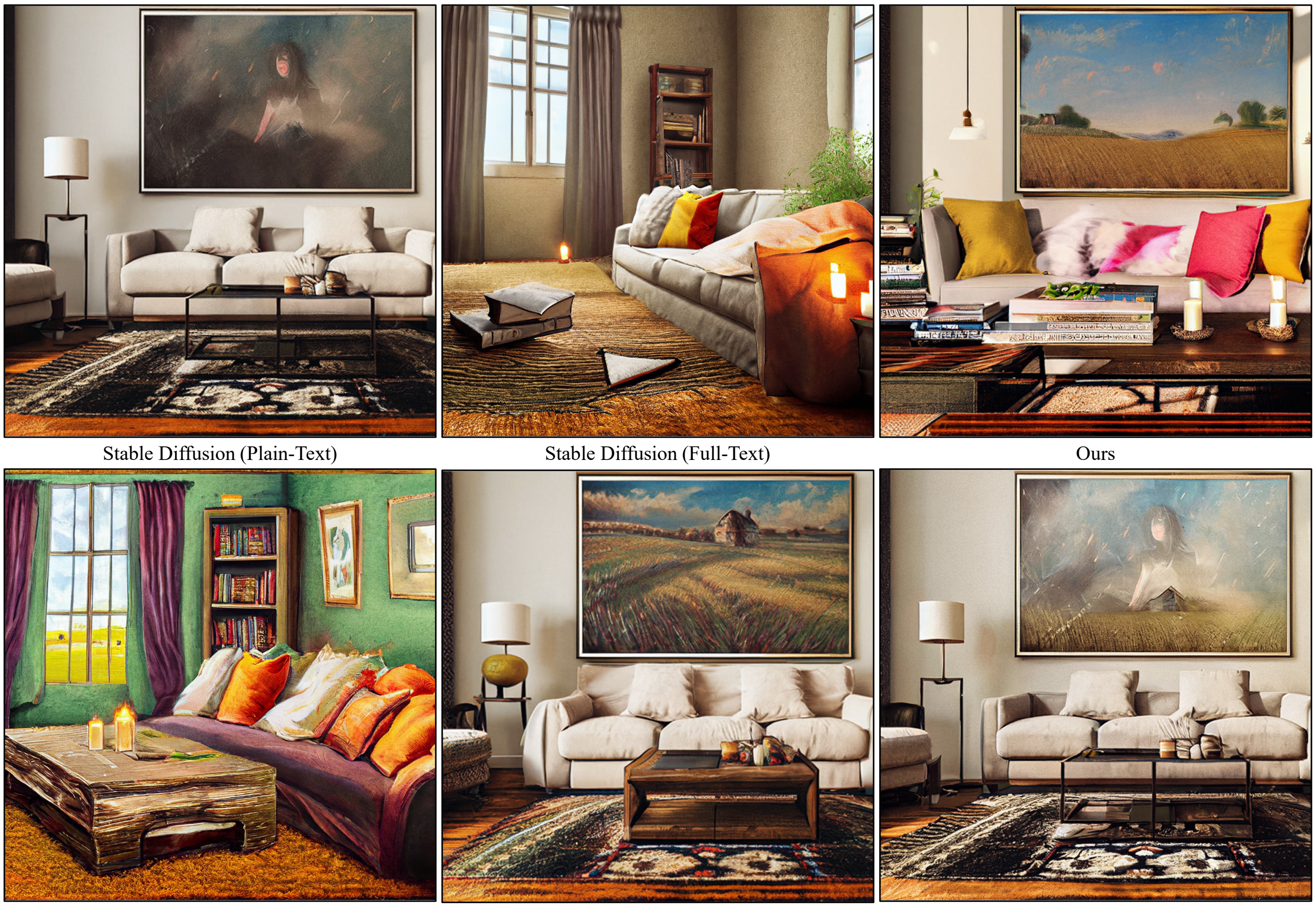

Figure 8. **Qualitative comparison on detailed description generation.** We show images generated by Attend-and-Excite, Prompt-to-Prompt, InstructPix2Pix, and our method using complex prompts. Our method is the only one that can generate all the details faithfully.

prompt, we use 12 objects exhibiting different colors, such as "flower", "gem", and "house". This gives us a total of $1,200$ prompts. We evaluate color accuracy by computing the mean L2 distance between the region and target RGB values. We also compute the minimal L2 distance as sometimes the object should contain other colors for fidelity, e.g., the "black tires" of a "yellow car".

Baselines. For font color and style, we quantitatively compare our method with two strong baselines, Prompt-to-Prompt [19] and InstructPix2Pix [7]. When two instructions exist for each image in our font style experiments, we apply them in parallel (InstructPix2Pix-para) and sequential manners (InstructPix2Pix-seq). More details are in Appendix B. We also perform a human evaluation with these two methods in Appendix Table 1. For re-weighting token importance, we visually compare with Prompt-to-Prompt [19] and two heuristic methods, repeating and adding parentheses. For complex scene generation with footnotes, we also compare with Attend-and-Excite [12].

4.1. Quantitative Comparison

We report the local CLIP scores computed by a ViT-B/32 model in Figure 6. Our method achieves the best overall CLIP score compared to the two baselines. This demonstrates the advantage of our region-based diffusion method for localized stylization. To further understand the capacity of each model to generate multiple styles, we report the metric on each region. Prompt-to-Prompt and InstructPix2Pix-para achieve a decent score on the 1^{st} Region, i.e., the region first occurs in the sentence. However, they often fail to fulfill the style in the 2^{nd} Region. We conjecture that the Stable Diffusion model tends to generate a uniform style for the entire image, which can be attributed to single-style training images. Furthermore, InstructPix2Pix-seq performs the worst in 2^{nd} Region. This is because the first instruction contains no information about the second region, and the

Figure 9. **Qualitative comparison on token reweighting.** We show images generated by our method and Prompt-to-Prompt using token weight of 13 for 'mushrooms'. Prompt-to-Prompt suffers from artifacts due to the large weight. Heuristic methods like repeating and parenthesis do not work well.

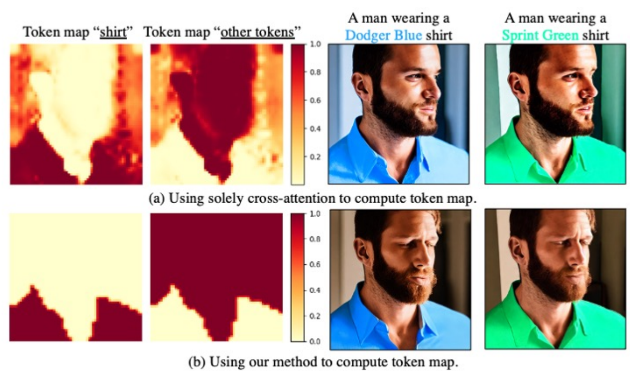

Figure 10. **Ablation of token maps.** Using solely cross-attention maps to create token maps leads to inaccurate segmentations, causing the background to be colored in an undesired way.

second region's content could be compromised when we apply the first instruction.

We show quantitative results of precise color generation in Figure 7. The distance of *HTML color* is generally the lowest for baseline methods, as they provide the most interpretable textual information for text encoders. This aligns with our expectation that the diffusion model can handle simple color names, whereas they struggle to handle the RGB triplet. Our rich-text-to-image generation method consistently improves on the three categories and two metrics over the baselines.

4.2. Visual Comparison

Precise color generation. We show qualitative comparison on precise color generation in Figure 4. InstructPix2Pix [7] is prone to create global color effects rather than accurate local control. For example, in the flower results, both the vase and background are changed to the target colors. Prompt-to-Prompt [19] provides more precise control over the target region. However, both Prompt-to-Prompt and InstructPix2Pix fail to generate precise colors. In contrast, our method can generate precise colors for all categories and prompts.

Local style generation. Figure 5 shows a visual comparison of local style generation. When applying InstructPix2Pix-seq, the style in the first instruction dominates the entire image and undermines the second region. Figure 13 in Appendix shows that this cannot be fully resolved using different hyperparameters of classifier-free guidance. Similar to our observation in the quantitative evaluation, our baselines tend to generate the image in a globally uniform style instead of distinct local styles for each region. In contrast, our method synthesizes the correct styles for both regions. One may suggest applying baselines with two stylization processes independently and composing the results using token maps. However, as shown in Figure 12 (Appendix), such methods generate artifacts on the region boundaries.

Complex scene generation. Figure 8 shows comparisons on complex scene generation. Attend-and-Excite [12] uses the tokens missing in the full-text generation result as input to fix the missing objects, like the coffee table and carpet in the living room example. However, it still fails to generate all the details correctly, e.g., the books, the painting, and the blanket. Prompt-to-Prompt [19] and InstructPix2Pix [7] can edit the painting accordingly, but many objects, like the colorful pillows and stuff on the table, are still missing. In contrast, our method faithfully synthesizes all these details described in the target region.

Token importance control. Figure 9 shows the qualitative comparison on token reweighting. When using a large weight for 'mushroom,' Prompt-to-Prompt generates clear artifacts as it modifies the attention probabilities to be unbounded and creates out-of-distribution intermediate features. Heuristic methods fail with adding more mushrooms, while our method generates more mushrooms and preserves the quality. More results of different font sizes and target tokens are shown in Figures 23 - 25 in Appnedix.

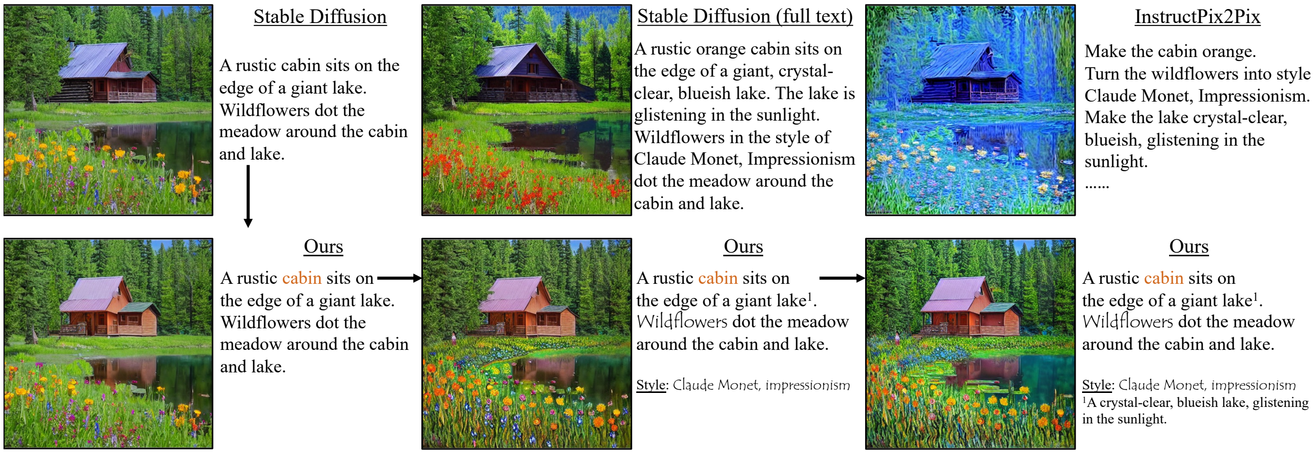

Figure 11. **Our workflow.** (top left) A user begins with an initial plain-text prompt and wishes to refine the scene by specifying the color, details, and styles. (top center) Naively inputting the whole description in plain text does not work. (top right) InstructPix2Pix [7] fails to make accurate editing. (bottom) Our method supports precise refinement with region-constrained diffusion processes. Moreover, our framework can naturally be integrated into a rich text editor, enabling a tight, streamlined UI.

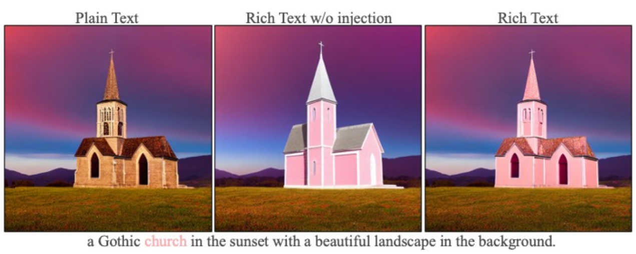

Figure 12. **Ablation of injection method.** We show images generated based on plain text and rich text with or without injection methods. Injecting features and noised samples help preserve the structure of the church and unformatted token regions.

Interactive editing. In Figure 11, we showcase a sample workflow to illustrate our method's interactive strength and editing capacity over InstructPix2Pix [7].

4.3. Ablation Study

Generating token maps solely from cross-attention. The other straightforward way to create token maps is to use cross-attention maps directly. To ablate this, we first take the average of cross-attention maps across heads, layers, and time steps and then take the maximum across tokens. Finally, we apply softmax across all the spans to normalize the token maps. However, as shown by the example in Figure 10, since the prompt has no correspondence with the background, the token map of "shirt" also covers partial background regions. Note that simple thresholding is ineffective as some regions still have high values, e.g., the right shoulder. As a result, the target color *bleeds* into the background. Our methods obtain more accurate token maps and, consequently, more precise colorization.

Ablation of the injection methods. To demonstrate the effectiveness of our injection method, we compare image generation with and without it in Figure 12. In the font color example, we show that applying the injection effectively preserves the shape and details of the target church and the structure of the sunset in the background. In the footnote example, we show that the injection keeps the looking of the black door and the color of the floor.

5. Discussion and Limitations

In this paper, we have expanded the controllability of text-to-image models by incorporating rich-text attributes as the input. We have demonstrated the potential for generating images with local styles, precise colors, different token importance, and complex descriptions. Nevertheless, numerous formatting options remain unexplored, such as bold/italic, hyperlinks, spacing, and bullets/numbering. Also, there are multiple ways to use the same formatting options. For example, one can use font style to characterize the shape of the objects. We hope this paper encourages further exploration of integrating accessible daily interfaces into text-based generation tasks, even beyond images.

Limitations. As we use multiple diffusion processes and two-stage methods, our method can be multiple times slower than the original process. Also, our way to produce token maps relies on a thresholding parameter. More advanced segmentation methods like SAM [31] could be exploited to further improve the accuracy and robustness.

Acknowledgment. We thank Mia Tang, Aaron Hertzmann, Nupur Kumari, Gaurav Parmar, Ruihan Gao, and Aniruddha Mahapatra for their helpful discussion and paper reading. This work is partly supported by NSF grant No. IIS-239076, as well as NSF grants No. IIS-1910132 and IIS-2213335.

References

[1] Maneesh Agrawala. Unpredictable black boxes are terrible interfaces, March 2023. https://magrawala.substack.com/p/unpredictable-black-boxes-are-terrible. 3

[2] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *arXiv preprint arXiv:2206.02779*, 2022. 2

[3] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[4] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 1, 2, 4

[5] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. *arXiv preprint arXiv:2302.07121*, 2023. 2, 5

[6] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. *arXiv preprint arXiv:2302.08113*, 2023. 2, 4

[7] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 5, 7, 8, 9

[8] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. `https://github.com/kakaobrain/coyo-dataset`, 2022. 2

[9] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. *arXiv preprint arXiv:2304.08465*, 2023. 2

[10] Duygu Ceylan, Chun-Hao Huang, and Niloy J. Mitra. Pix2video: Video editing using image diffusion. *arXiv:2303.12688*, 2023. 2

[11] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 2

[12] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *arXiv preprint arXiv:2301.13826*, 2023. 2, 3, 4, 7, 8

[13] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021. 5

[15] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *arXiv preprint arXiv:2204.14217*, 2022. 2

[16] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Reddy Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *International Conference on Learning Representations (ICLR)*, 2023. 2

[17] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision (ECCV)*, pages 89–106. Springer, 2022. 1, 2

[18] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[19] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 1, 2, 4, 5, 7, 8

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Neural Information Processing Systems (NeurIPS)*, 2020. 2, 5

[21] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. 2

[22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2

[23] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5

[24] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 2, 5

[25] Claudia-Lavinia Ignat, Luc André, and Gérald Oster. Enhancing rich content wikis with real-time collaboration. *Concurrency and Computation: Practice and Experience*, 33(8):e4110, 2021. 3

[26] Álvaro Barbero Jiménez. Mixture of diffusers for scene composition and high resolution image generation. *arXiv preprint arXiv:2302.02412*, 2023. 2

[27] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4565–4574, 2016. 3

[28] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2

[29] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 3

[30] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 9

[32] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. *arXiv preprint arXiv:2212.04488*, 2022. 2, 4

[33] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM SIGGRAPH*, 2004. 2

[34] Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. Dit: Self-supervised pre-training for document image transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3530–3539, 2022. 2

[35] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[36] Geoffrey Litt, Sarah Lim, Martin Kleppmann, and Peter van Hardenberg. Peritext: A crdt for collaborative rich text editing. *Proceedings of the ACM on Human-Computer Interaction (PACMHCI)*, 2022. 3

[37] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *arXiv preprint arXiv:2303.11328*, 2023. 2

[38] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. *arXiv:2303.04761*, 2023. 2

[39] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[40] Wan-Duo Kurt Ma, JP Lewis, W Bastiaan Kleijn, and Thomas Leung. Directed diffusion: Direct control of object placement through attention guidance. *arXiv preprint arXiv:2302.13153*, 2023. 2, 4

[41] Yue Ma, Yingqing He, Xiaodong Cun, Xintao Wang, Ying Shan, Xiu Li, and Qifeng Chen. Follow your pose: Pose-guided text-to-video generation using pose-free videos, 2023. 2

[42] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Generating images from captions with attention. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[43] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *International Conference on Learning Representations (ICLR)*, 2022. 2

[44] Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. Glyce: Glyph-vectors for chinese character representations. *Neural Information Processing Systems (NeurIPS)*, 32, 2019. 2

[45] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *International Conference on Machine Learning (ICML)*, pages 16784–16804, 2022. 2

[46] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. *arXiv preprint arXiv:2302.03027*, 2023. 2

[47] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. *arXiv preprint arXiv:2303.11306*, 2023. 2, 4

[48] Quynh Phung, Songwei Ge, and Jia-Bin Huang. Grounded text-to-image synthesis with attention refocusing. *arXiv preprint arXiv:2306.05427*, 2023. 2

[49] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing, 2023. 2

[50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021. 3, 6

[51] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[52] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning (ICML)*, pages 8821–8831, 2021. 1, 2

[53] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001. 2

[54] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 5

[55] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[56] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed

Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 1, 2

[57] Arnaud Sahuguet and Fabien Azavant. Wysiwyg web wrapper factory (w4f), 1999. 3

[58] Vishnu Sarukkai, Linden Li, Arden Ma, Christopher R'e, and Kayvon Fatahalian. Collage diffusion. *ArXiv*, abs/2303.00262, 2023. 2

[59] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. *arXiv preprint arXiv:2301.09515*, 2023. 2

[60] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 2

[61] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 4

[62] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[63] Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021. 2

[64] Yu-Wing Tai, Jiaya Jia, and Chi-Keung Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 747–754 vol. 1, 2005. 2

[65] Raphael Tang, Akshat Pandey, Zhiying Jiang, Gefei Yang, K. V. S. Manoj Kumar, Jimmy Lin, and Ferhan Ture. What the daam: Interpreting stable diffusion using cross attention. *ArXiv*, abs/2210.04885, 2022. 4

[66] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhib Alsisan, Jia-Bin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[67] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. *arXiv preprint arXiv:2211.12572*, 2022. 4, 5

[68] Colorado State University. tutorial: Rich text format (rtf) from microsoft word - the access project - colorado state university, 2012-07-08. 1

[69] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007. 4

[70] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models, 2022. 2

[71] Ian H Witten, David Bainbridge, and David M Nichols. *How to build a digital library*. Morgan Kaufmann, 2009. 1

[72] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. *arXiv preprint arXiv:2212.11565*, 2022. 2

[73] Guangxuan Xiao, Tianwei Yin, William T Freeman, Frédo Durand, and Song Han. Fastcomposer: Tuning-free multi-subject image generation with localized attention. *arXiv preprint arXiv:2305.10431*, 2023. 2

[74] Li Xu, Qiong Yan, and Jiaya Jia. A sparse control model for image and video editing. *ACM Transactions on Graphics (TOG)*, 32:1 – 10, 2013. 2

[75] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020. 2

[76] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022. 2

[77] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 1, 2

[78] Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. Diffcollage: Parallel generation of large content with diffusion models. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[79] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[80] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)*, 9(4), 2017. 2

[81] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[82] Xiaojin Zhu, Andrew B Goldberg, Mohamed Eldawy, Charles R Dyer, and Bradley Strock. A text-to-picture synthesis system for augmenting communication. In *AAAI Conference on Artificial Intelligence*, 2007. 2