NoiseCollage: A Layout-Aware Text-to-Image Diffusion Model Based on Noise Cropping and Merging

Takahiro Shirakawa, Seiichi Uchida
Kyushu University, Japan
takahiro.shirakawa@human.ait.kyushu-u.ac.jp, uchida@ait.kyushu-u.ac.jp

Abstract

Layout-aware text-to-image generation is a task to generate multi-object images that reflect layout conditions in addition to text conditions. The current layout-aware text-to-image diffusion models still have several issues, including mismatches between the text and layout conditions and quality degradation of generated images. This paper proposes a novel layout-aware text-to-image diffusion model called NoiseCollage to tackle these issues. During the denoising process, NoiseCollage independently estimates noises for individual objects and then crops and merges them into a single noise. This operation helps avoid condition mismatches; in other words, it can put the right objects in the right places. Qualitative and quantitative evaluations show that NoiseCollage outperforms several state-of-the-art models. These successful results indicate that the crop-and-merge operation of noises is a reasonable strategy to control image generation. We also show that NoiseCollage can be integrated with ControlNet to use edges, sketches, and pose skeletons as additional conditions. Experimental results show that this integration boosts the layout accuracy of ControlNet. The code is available at https://github.com/univ-esuty/noisecollage.

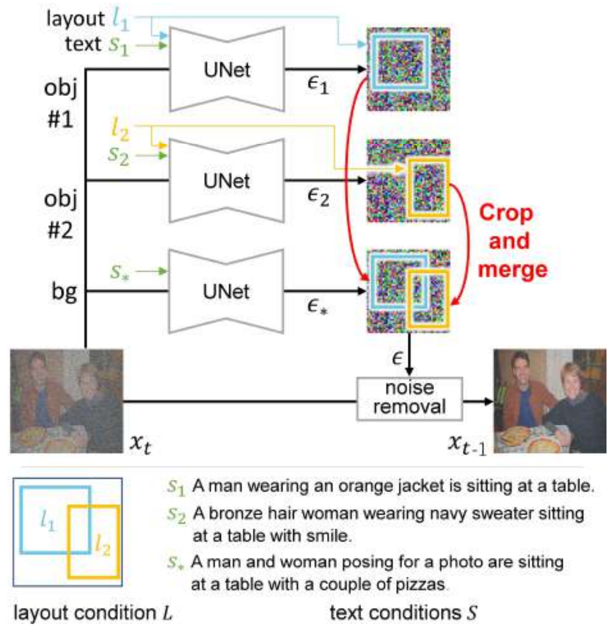

Figure 1. Denoising processes of NoiseCollage. (Although illustrated as a process in the image space, the actual denoising process is performed in a latent space like [28] for computational efficiency.)

1. Introduction

Diffusion models, such as StableDiffusion (SD) [28], have rapidly improved text-to-image generation. In general, diffusion models generate images through a denoising process, an iterative process to remove noise from an initial Gaussian noise image. A UNet estimates the noise with a text condition. After the denoising iterations, the model gives a noise-free (clean) image that reflects the text condition.

Text-to-image diffusion models have recently been extended to generate multiple objects with layout awareness. Namely, these models can generate images with multiple objects while controlling their spatial locations. There are two approaches for the extension: attention manipulation [4, 9, 16, 19, 20, 24, 34, 35] and iterative image editing [1, 2, 31, 38, 41, 42]. The former manipulates the cross attention layers in the UNet for letting a certain region only focus on a certain object. The latter generates an initial image and then puts another object in the initial image. More objects can be arranged by repeating this editing process.

The current layout-aware text-to-image diffusion models still have the following limitations. Specifically, the first approach, attention manipulation, often shows mismatches between the text and layout conditions. The second approach, iterative editing, shows the image quality degradation as it iterates to show more objects.

This paper proposes a novel layout-aware text-to-image diffusion model called *NoiseCollage*. Fig. 1 shows an overview of the denoising process of NoiseCollage. When generating images with N objects, NoiseCollage takes $N+1$ text conditions (i.e., prompts) $\{s_1, \ldots, s_N, s_*\}$ and N layout conditions $\{l_1, \ldots, l_N\}$ for image generation. Namely, a pair of text and layout conditions (s_n, l_n) are given for each object n. As shown in Fig. 1, each layout condition is specified by a bounding box unless otherwise mentioned. The remaining text condition s_* roughly describes the whole image.

The technical highlight of NoiseCollage is that $N+1$ noises $\{\epsilon_1, \ldots, \epsilon_N, \epsilon_*\}$ for N objects and the whole image are estimated independently and then assembled like image collage. More specifically, the region l_n for the n-th object is cropped from ϵ_n, and then the N cropped noises are merged with the noise for the whole image, ϵ_*. This operation is novel and very different from the existing text-to-image diffusion models; our assemblage operation *directly* creates a noise from $N+1$ noises to have an expected output image. In other words, our trials on NoiseCollage indicate that *multi-object images can be generated accurately by the crop-and-merge operation of noises*.

For accurate and flexible image generation, we introduce three gimmicks in NoiseCollage. The first gimmick is *masked cross attention*. This gimmick aims to estimate a noise ϵ_n that accurately reflects the text condition s_n around the region l_n. The second gimmick is to make the crop-and-merge operation to be soft. More specifically, we use a weighted merging operation so that the cropped noises do not completely overwrite the global information of ϵ_*. The weighted merging operation also allows (even large) overlaps between the regions $\{l_n\}$. The third gimmick is an integration of ControlNet[40] to allow more flexible conditions. ControlNet employs various conditions, such as pose skeleton and edge images, for guiding image generation; therefore, the integration with ControlNet allows NoiseCollage to use these guiding conditions.

Like other popular layout-aware image generation methods [4, 31], NoiseCollage is *training-free* and thus can employ various diffusion models pre-trained to generate images from a text condition. We mainly used a pretrained SD for photo-realistic images in the later experiments. However, as noted above, we also used an SD model for anime images and ControlNet. If we have better diffusion models in the near future, we can employ them for NoiseCollage without any modification.

We conduct various qualitative and quantitative evaluation experiments to confirm that our NoiseCollage outperforms the state-of-the-art layout-aware image generation models. We first observe that NoiseCollage generates multi-object images that are high-quality and accurate to the input conditions. We then quantitatively evaluate how accurately the given conditions are reflected in the corresponding objects. For this evaluation, we introduce multimodal feature representation by CLIP[26]; if a model shows high CLIP-feature similarity between text conditions and generated images, the model will have high accuracy to the input conditions.

The main contributions of this paper are summarized as follows.

- We propose NoiseCollage, a novel layout-aware text-to-image diffusion model. It can generate multi-object images that accurately reflect text and layout conditions.
- NoiseCollage is the first method that performs a crop-and-merge operation of noises estimated for individual objects in its denoising process. Its accurate and high-quality generated images without artifacts indicate that noise is a good medium for direct layout control.
- Experimental results show that NoiseCollage outperforms the state-of-the-art methods by avoiding condition mismatches.
- The Training-free nature of NoiseCollage allows direct integration with ControlNet and realizes finer output controls by edge images, sketches, and body skeletons.

2. Related Work

2.1. Text-to-Image Diffusion Models

Many image generation methods based on diffusion models have been proposed so far [10, 14, 22, 32, 33]. For generating high-resolution images without a drastic increase in computational costs, they often employ the technique of Latent Diffusion Model (LDM) [28], where the denoising process with UNet runs in a low-dimensional latent space. Various conditions are also introduced to control the generated images.

Text-to-image diffusion models [4, 11, 25, 27, 28, 30] can generate high-quality and diverse images with a text condition. Among them, StableDiffusion (SD) [28] is one of the most popular models. Those models have been extended to realize other image processing with tasks, such as image editing [5, 8, 12], image inpainting [18, 36, 39], and image-to-image translation [21, 23, 29].

2.2. Layout-Aware Diffusion Models

Layout-aware text-to-image generation is a task to generate multi-object images that reflect a layout condition in addition to a text condition. Several fine-tuning techniques [3, 7, 15, 37, 40] have been proposed to incorporate the layout condition into the pre-trained diffusion model. For example, MultiDiffusion employs an extra optimization step to mix the denoised images into one. ControlNet [40] combines SD and a trainable encoder of various conditions for fine layout control, such as pose skeletons for human pose control.

Table 1. Comparison of popular state-of-the-art layout-aware text-to-image diffusion models.

	Control-Net[40]	Paint-with words[4]	Collage Diffusion[31]	Ours
training-free	×	√	√	√
non-iterative	√	√	×	√
multi-prompts	×	×	√	√
region overlap	√	×	√	√

We can find *training-free* methods that use the pre-trained models without fine-tuning steps for layout conditions. They are classified into *attention manipulation* and *iterative editing*. Attention manipulation methods [4, 9, 16, 19, 20, 24, 34, 35] control the object layout by manipulating a cross-attention layer, which is an important module in UNet to correlate text conditions and regions in the generated images. Paint-with-words[4] is the most popular state-of-the-art method that uses attention manipulation. It can generate images from a text condition and an object segmentation mask. A word (such as "rabbit") in the text condition is given to each segment, and this word-segment correspondence is then used to modify the cross-attention. However, as we will see later, controlling the correspondence between multiple objects and their regions within a cross-attention layer is tough and often suffers from wrong correspondences, i.e., condition mismatches.

Iterative editing[1, 2, 31, 38, 41, 42] is a more intuitive way to deal with multiple objects and their layout. Given a pre-generated initial image, one object is placed at its position, and then the next object is placed. Repeating this step N times gives us an image with N objects. Collage Diffusion [31] is a popular iterative editing method. Although it introduces an extra diffusion and denoising step like SDEdit[21] to harmonize the newly added object in the resulting image, it still suffers from image quality degradation, which becomes more serious according to many iterations.

Table 1 summarizes functionality comparisons between popular layout-aware text-to-image methods and our NoiseCollage. "Multi-prompts" is the function to accept different text conditions for individual objects. "Region overlap" is the function to allow overlapping layout conditions (by, for example, bounding boxes) for objects. As indicated by this table, our NoiseCollage has several promising properties.

2.3. Noise Manipulation

The most popular manipulation of the estimated noise in diffusion models is classifier-free guidance [13]. It uses the difference between a pair of noises estimated with and without a class condition. This difference reflects the class-specific characteristics and thus is useful to emphasize them in the generated images.

To the authors' knowledge, no existing model manipulates the estimated noises in a direct manner, such as our crop-and-merge operation. As we will see in this paper, noises are a good medium for allowing simple and intuitive manipulations to control the object layout without introducing any artifacts.

3. NoiseCollage

3.1. Overview

NoiseCollage generates an image with N objects from the following conditions, L, S, and s_*:

- $L = \{l_1, \dots, l_N\}$ is the N layout conditions to control the layout of individual objects. Each layout condition l_n is represented as a region specified by a bounding box or a polygon. Note that regions can be overlapped; thus, there is no need to be nervous about setting layout conditions.
- $S = \{s_1, \dots, s_N\}$ is the set of N text conditions to describe the visual information of the objects. Each condition is given as a word sequence; for example, "A man wearing an orange jacket is sitting at a table."
- s_* is a global text condition to describe the whole image of the objects. Although we call it "global," s_* need *not* describe everything. The global text s_* may outline the whole image or include descriptions of several objects.

NoiseCollage uses the denoising process of standard diffusion models. NoiseCollage starts from $t = T$ with a Gaussian noise image x_T. Then, from $t = T$ to 1, it uses a pre-trained UNet to estimate the noise ϵ at each t from a noisy image x_t and then removes the noise ϵ from x_t to have a less-noisy image x_{t-1}. The denoising process finally provides an image x_0, which satisfies the given conditions.

The main difference between NoiseCollage and the standard diffusion models is that it derives the noise ϵ at t by a crop-and-merge operation (i.e., collage) of $N+1$ noises, $\{\epsilon_1, \dots, \epsilon_N, \epsilon_*\}$[1], as shown in Fig. 1. Roughly speaking, the noise ϵ is given by cropping the region specified by l_n from ϵ_n for each n and then merging the N cropped regions with ϵ_*. In the following, Section 3.2 details the crop-and-merge operation. Then, Section 3.3 details the masked cross-attention mechanism, which is necessary to make the crop-and-merge operation work as expected.

3.2. Crop-and-Merge Operation of Noises

A naive crop-and-merge operation for creating ϵ is to use ϵ_n for the n-th object region l_n and ϵ_* for the region for the non-object region. However, this naive operation has two issues. First, ϵ_* should not be excluded from the object region l_n. For example, when generating a ring-shaped object

[1] Precisely speaking, the noise ϵ should be denoted as $\epsilon(x_t \mid t, L, S, s_*)$, because it is estimated from x_t at timestep t under the conditions L, S, and s_*. Similarly, ϵ_n and ϵ_* are denoted as $\epsilon_n(x_t \mid t, l_n, s_n)$ and $\epsilon_*(x_t \mid t, s_*)$, respectively. In this paper, we use the simpler notation ϵ, ϵ_n, and ϵ_*, unless there is confusion.

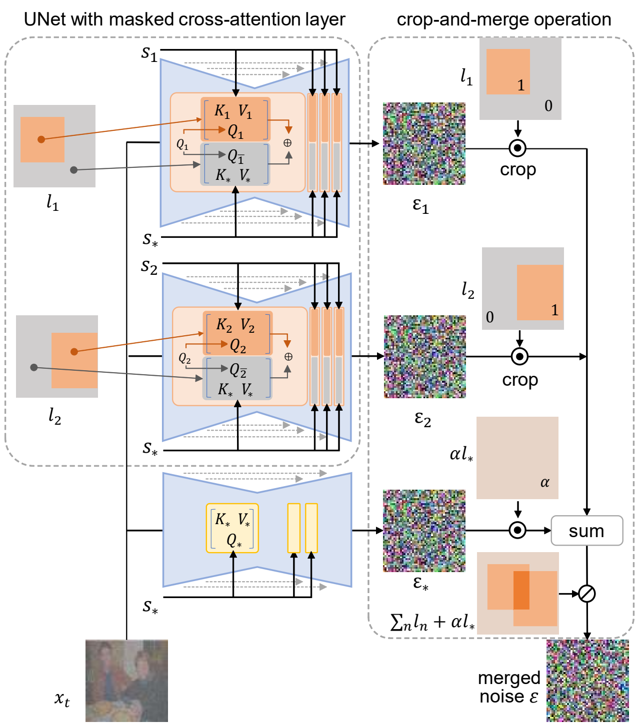

Figure 2. Overview of the noise estimation process in our NoiseCollage.

in the box l_n, ϵ_* is necessary for the non-ring area within l_n. Second, the naive operation does not consider the overlapping regions among $\{l_n\}$.

We, therefore, use the following crop-and-merge operation, illustrated in the right side of Fig. 2:

$$\epsilon = (\textstyle\sum_n l_n \epsilon_n + \alpha l_* \epsilon_*)/(\textstyle\sum_n l_n + \alpha l_*). \tag{1}$$

Here, l_n is treated as a binary mask image whose pixel value is 1 for the region specified by l_n, and l_* is an image whose all pixels are 1. In Eq. 1, addition, multiplication, and division are pixel-wise. The hyper-parameter α is the weight to control the strength of ϵ_* within the object regions and set at 0.1 by a preliminary experiment.

3.3. Masked Cross-Attention

The cross-attention layer in the UNet of standard text-to-image diffusion models is an important module to correlate texts and image regions. Specifically, it calculates $\tilde{Q} = \mathrm{softmax}(QK^T/\sqrt{d})V$, where the query Q is a matrix with N d-dimensional image features of x_t, whereas the key K and the value V are the same matrix with M d-dimensional text features from the text conditions (S, s_*). Through this layer, N image features Q are converted into N image features (denoted as $\tilde{Q}$) that reflect text conditions.

We propose a "masked" cross-attention layer, which is a simple extension of the above cross-attention, as shown in the left side of Fig. 2. In the process of estimating ϵ_n in NoiseCollage, the visual information of the n-th object by s_n should be localized around the region l_n in ϵ_n; this is because only the region l_n is cropped and merged into ϵ. For this localization, we split the cross-attention operation into two sub-operations: one is a sub-operation to correlate the region l_n with s_n and the other to correlate the remaining region with s_*. Specifically, we first derive two "masked" matrices Q_n and $Q_{\overline{n}}$ from Q, where the matrix Q_n has the value of Q at the columns corresponding to l_n and zero at the other columns and $Q_{\overline{n}} = Q \ominus Q_n$. We also derive V_n and K_n from s_n and V_* from s_*. We then get the cross-attention results for the n-th object as $\tilde{Q}_n = \mathrm{softmax}(Q_n K_n^T/\sqrt{d})V_n$ and for the other region as $\tilde{Q}_{\overline{n}} = \mathrm{softmax}(Q_{\overline{n}} K_*^T/\sqrt{d})V_*$. Finally, we have the masked cross-attention result by simply adding them, that is, $\tilde{Q}_n \oplus \tilde{Q}_{\overline{n}}$. $\oplus$ and $\ominus$ denote element-wise addition and subtraction, respectively. Note that for the UNet to estimate ϵ_*, the standard self-attention layer triggered by s_* is used instead of the masked cross-attention.

The masked cross-attention accurately puts "the right objects in the right places;" the visual information for the n-th object is localized around l_n in ϵ_n and thus the crop-and-merge operation of noises $\{\epsilon_n\}$ guided by $\{l_n\}$ provides ϵ that accurately reflects the conditions. Note that the mechanism of NoiseCollage that estimates noise ϵ_n for each object n independently facilitates the cross-attention between text and image. If we need to process all N objects and their text conditions in the *single* cross-attention layer, it is difficult to completely exclude the effects of other $N-1$ objects in the attention process of a certain object. Paint-with-words [4], a layout-aware text-to-image model based on attention manipulation, tries to control N objects in a single cross-attention layer and often suffers from confusion among the objects.

4. Experiments

4.1. Implementation Details

We implement NoiseCollage in the SD framework; therefore, the denoising process, including the noise estimation and the crop-and-merge operation, is performed in a latent space; the generated image is given by a decoder from the latent space to the image space. Since NoiseCollage is a training-free model, we employ the pre-trained SD model (SD1.5) by CivitAI[2] for generating photo-realistic or anime-style images in the following experiments. The size of the generated image is set to fit the 512 pixel box while keeping its aspect ratio. We use UniPCMultistepScheduler [43] as the scheduler of the denoising process and classifier-free guidance [13] with the guidance scale, 7.5. The total denoising step is set to 50. Please refer to the

[2]https://civitai.com/

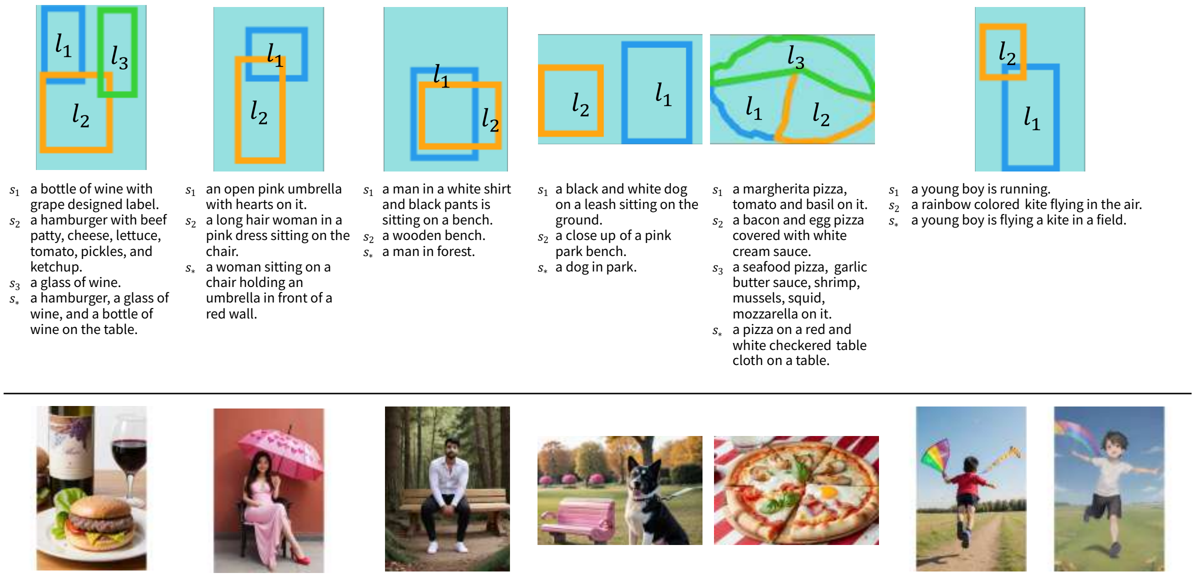

Figure 3. Images generated by NoiseCollage with layout conditions L and text conditions (S, s_*).

supplementary for the details of the total denoising step and inference step.

4.2. Datasets

For performance evaluation experiments, we construct two datasets, BD807 and MD30, where each sample is a combination of an image x_0, its layout conditions L, and text conditions (S, s_*). We collected images from the MS-COCO test dataset because the boundaries of most objects are annotated with polygons and bounding boxes. We use bounding boxes as L, which makes NoiseCollage a more handy image generator. However, later qualitative evaluations use polygons as L in several examples to show the flexibility of the layout condition. We pick up 807 images from the MS-COCO dataset containing $N = 2 \sim 5$ objects whose region size is larger than 128×128 pixels.

Although the MS-COCO dataset also contains image captions, we do not use them as text conditions but prepare our conditions by using BLIP2 [17]. This is because the COCO's caption describes the whole image and is inappropriate as the text condition s_n for the individual object. We use the description automatically given by applying BLIP2 to each object region l_n as s_n and the description for the whole image by BLIP2 as s_*. We call the dataset realized by the above procedure BD807 (BLIP2-guided Dataset with 807 images). MD30 (Manually-annotated Dataset) comprises 30 images chosen from the 807 images. For those images, we discard s_n by BLIP2 and attach a more accurate text as s_n by a human annotator. Note that it only contains 30 images, but its purpose is only to supplement the main result with a larger dataset, BD807.

4.3. Qualitative Evaluation Result

Fig. 3 shows multi-object images generated with various conditions. Layout conditions L are given as bounding boxes or polygons, often overlapping (even largely). Text conditions $S = \{s_1, \ldots, s_N\}$ and s_* describe the appearance of N objects and the whole image. Note that several conditions in Fig. 3 are modified from BD807 and MD30 to show the various characteristics of NoiseCollage.

The results in Fig. 3 suggest that the crop-and-merge operation of noises is a very reasonable way to lay out multiple objects accurately. Specifically, it can be said to be "reasonable" based on the following two points. First, no artifact exists around the border of the object region l_n. Furthermore, a (large) overlap between regions does not degrade the reality of the generated image. Second, no confusion exists between layout conditions $\{l_n\}$ and text conditions $\{s_n\}$. In other words, the object described by s_n is correctly located around l_n. Section 4.4 shows that even state-of-the-art layout-aware text-to-image models suffer from confusion about the correspondence between texts and locations.

A closer observation of Fig. 3 reveals various characteristics of NoiseCollage. For example, it shows that we need not be nervous in preparing the global text condition s_*; for the third and fourth examples from the left, we intentionally use much shorter global text conditions (than the first and second), but the results are still natural. In the pizza image, the layout conditions L are given as polygons; the resulting image shows that polygons help to control the object shapes accurately. The image of a running boy is generated in two styles, i.e., photo-realistic and anime. NoiseCollage is training-free; thus, any pre-trained noise-estimation model can be plugged into it. This anime-style image is

Figure 4. Comparison of generated images and their generation process by NoiseCollage and Collage Diffusion[31].

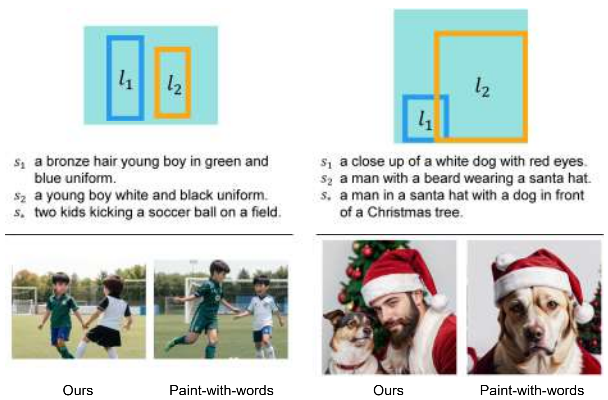

Figure 5. Comparison of generated images by NoiseCollage and Paint-with-words[4].

generated simply using a different pre-trained SD model by CivitAI.

4.4. Qualitative Comparison with State-of-the-Art Models

Fig. 4 compares two generated images and by our NoiseCollage and Collage Diffusion [31]. Since Collage Diffusion is an iterative editing model, this figure also shows an iterative process where conditions are applied one at a time. Compared to the successful results by NoiseCollage, the results by Collage Diffusion show two issues. The first issue is that the results strongly depend on the initial image given by the global text condition s_*. In the "bus" image, the initial image by s_* shows (coincidentally) a red bus on the right side. Then, the second condition (l_2, s_2) is tried to generate a red bus on the left side, but it was ineffective because the red bus is already in the generated image. In the "bottle" image, the initial image shows bananas on the label of each bottle. Thus, like the bus image, the fourth and fifth conditions applied later were ineffective.

The second issue is the quality degradation by iterations. This degradation becomes more severe when more objects require more iterations. In the "bottle" image, the initial

Table 2. Average similarity (↑) between text conditions S and generated image x_0. Red indicates the model with the highest similarity. The parenthesized number (↑) shows the percentage of the samples where NoiseCollage shows a better similarity than the comparative model. For example, NoiseCollage outperforms Paint-with-words at 77% samples of MD30.

	Paint-with-words	Collage Diffusion	NoiseCollage
MD30	0.250 (77%)	0.253 (70%)	**0.280**
BD807	0.240 (65%)	0.237 (68%)	**0.256**

image by s_* shows readable characters on the bottle labels; however, the later iterations gradually degrade their readability, and the characters become almost unreadable in the final image given after five iterations.

Fig. 5 shows comparisons of generated images and their generation process by NoiseCollage and Paint-with-words [4]. In the "soccer" image, Paint-with-words could not correctly color the boys' uniforms, but NoiseCollage succeeded. Paint-with-words assumes shorter text conditions for precise layout control by manipulating its cross-attention module. In other words, such control would be difficult with longer text conditions for describing the detailed appearance of objects. In the "Santa" image, two conditions are mixed into one object. This result shows the difficulty of controlling multiple objects in a single cross-attention layer, even with attention manipulation. NoiseCollage uses multiple noises and multiple masked cross-attention operations for individual objects; thus, the objects are well separated.

4.5. Quantitative Evaluation Results

We evaluate how the generated image accurately reflects the layout and text conditions. Specifically, we evaluate a multimodal similarity between the image region l_n and its text condition s_n. If a model appropriately generates an object around l_n while reflecting s_n, their multimodal similarity should be high. Following some related works [3, 16, 31], we use the ImageEncoder of CLIP [26] to have an image feature vector of the region l_n and the TextEncoder for a

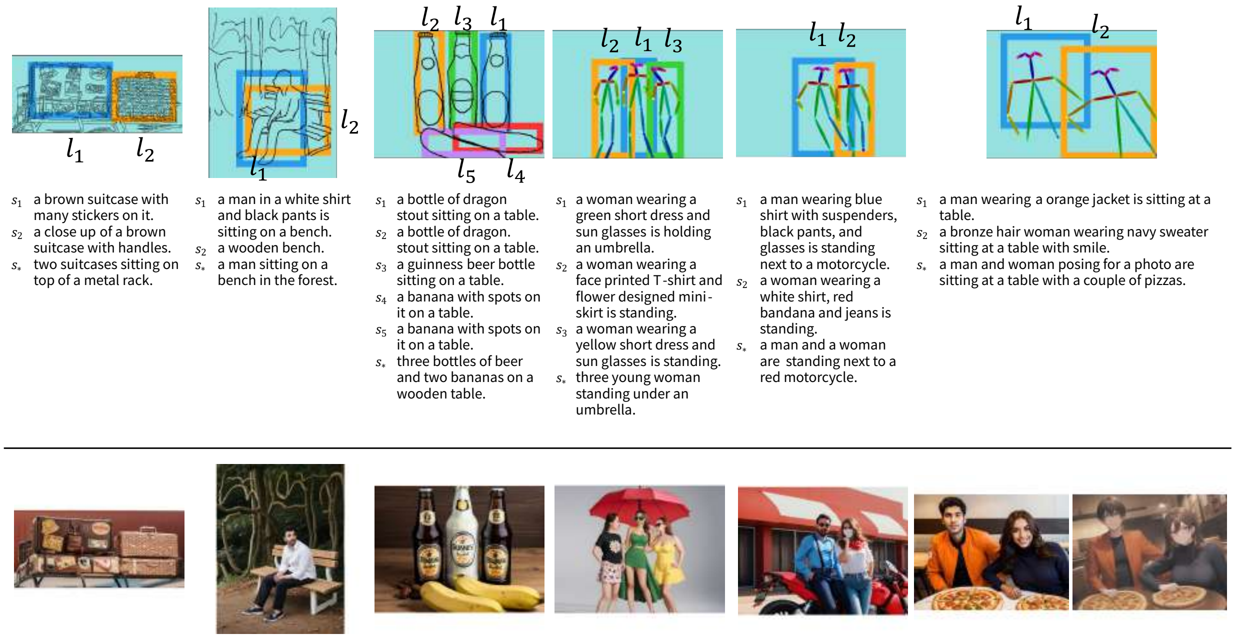

Figure 6. Images generated by NoiseCollage with ControlNet[40]. The first image is generated with an edge image, the second and third images are with a sketch image, and the remaining images are with a pose skeleton.

text feature vector of s_n. Then, we use the cosine similarity between these two feature vectors. We have used the same layout and caption conditions described at 4.2 for all the methods for a fair comparison.

Table 2 shows the average similarity achieved by three models (Paint-with-words, CollageDiffution, and NoiseCollage) on the two datasets, MD30 and BD807. In both datasets, NoiseCollage shows higher average similarity than the other models. In the sample-level evaluation, NoiseCollage shows higher similarities than the others in about 70% samples. These results prove that NoiseCollage can more accurately satisfy the layout and text conditions. Among MD30 and BD807, the latter shows slightly lower similarities; one reason may be that BD807 text conditions are automatically generated.

5. NoiseCollage with ControlNet

5.1. Integration of ControlNet for Finer Controls

ControlNet[40] is a well-known text-to-image diffusion model that can accept various conditions in addition to text conditions. For example, it accepts a pose skeleton to control the pose of a person in the generated image. It also accepts a canny-edge image or a hand-drawn sketch image to control the shape of objects to be generated. The pose skeletons and canny-edge images are automatically generated and the sketch images are created by the authors manually while tracing the images.

We can integrate this fine control of ControlNet into NoiseCollage. This integration is done simply by using the pre-trained UNet of ControlNet in the framework of NoiseCollage. The UNet estimates ϵ_n with an additional condition, such as a pose skeleton, for the n-th object. Then, the crop-and-merge operation is performed to have ϵ. Note that ϵ_* is estimated with all the additional conditions and s_*.

5.2. Datasets for Evaluation

For evaluating the performance of the integrated version, two additional conditions are attached to the dataset of Sec. 4. Specifically, we prepare a canny-edge image for each image in MD30 and BD807, and a sketch image (drawn manually) for MD30.

We prepared two more datasets, HMD20 and HBD256, with human pose as an additional condition. For these datasets, 256 multi-person images are collected from the MS-COCO test dataset. Then, the pose skeleton of each person is estimated by OpenPose [6]. Finally, HBD256 (Human BLIP2 Dataset) is prepared by adding a text condition generated automatically by BLIP2[17] for each person region. HMD20 (Human Manual Dataset) is prepared by adding a text condition by a human annotator for each of the 20 images randomly selected from the 256 images.

5.3. Generated Images by NoiseCollage with ControlNet

Fig. 6 shows six images generated by NoiseCollage integrated with ControlNet. We used ControlNet implemented by Huggingface[3], and the other details are the same as

[3] https://huggingface.co/lllyasviel/ControlNet

Table 3. Average similarity (↑) in the experiment with ControlNet. The parenthesized number (↑) shows the percentage of the samples where the model shows a better similarity than the other.

Condition	Dataset	Standard ControlNet [40]		NoiseCollage+ ControlNet	
Edge	RMD30	0.293	(36%)	**0.307**	(64%)
	RMD807	0.265	(29%)	**0.284**	(71%)
Sketch	RMD30	0.276	(23%)	**0.300**	(77%)
Pose	HMD20	0.297	(15%)	**0.332**	(85%)
	HBD256	0.250	(28%)	**0.280**	(72%)

Sec. 4.1. All those results show multi-object images that accurately reflect the additional conditions to ControlNet. For example, the conditions by pose skeletons successfully control the pose of the persons in the generated image. Notably, the integration with ControlNet does not disturb the precise control of NoiseCollage. For example, the third and fourth images from the left accurately reflect their confusing text conditions on bottle types and sunglasses, respectively.

5.4. Quantitative Comparison with Standard ControlNet

Table 3 shows quantitative evaluation results of the images generated by NoiseCollage with ControlNet and the standard ControlNet. The evaluation metric is the multimodal similarity explained in Sec. 4.5. In about 70% to 80% of the samples, the performance of ControlNet is improved in our NoiseCollage framework. This improvement is prominent in generating multi-person images with pose skeletons. Although the standard ControlNet can reflect pose conditions in its generated images, it often shows confusion, such as a text condition for one person being reflected in another person. In contrast, if ControlNet is used in NoiseCollage, it can avoid such confusion by estimating noises independently for individual persons under corresponding conditions.

6. Limitation and Social Impacts

Fig. 7 shows the limitation of NoiseCollage. NoiseCollage sometimes ignores small objects in the generated image. In the first case, a frisbee is not generated in the image. In the second case, both cars are not generated. Note that the state-of-the-art methods also show difficulty in generating small objects. As shown in the right side of Fig. 7, Collage Diffusion also ignores small objects. Paint-with-words did not ignore them but showed them in the wrong places and styles.

Like recent diffusion models, the negative social impact of NoiseCollage is its ability to generate realistic fake images by finer appearance and location control of whole objects or even object parts. For example, since NoiseCollage can independently and easily control individuals in an image, it can potentially create images that depict fake relationships between people.

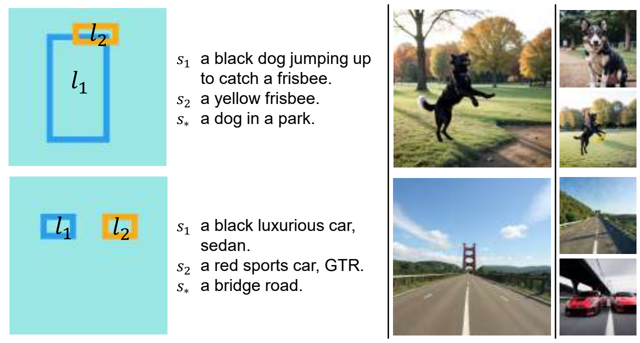

Figure 7. Failure cases by NoiseCollage. Smaller images on the right side are results by Collage Diffusion [31] and Paint-with-words [4].

7. Conclusion and Future Work

This paper proposed a novel layout-aware text-to-image diffusion model called *NoiseCollage*. The key idea of NoiseCollage, which can generate multi-object images, is to estimate noises for individual objects independently and then *crop-and-merge* them into a single noise in its denoising process. This operation helps avoid mismatches between the text and layout conditions; in other words, it can accurately put the objects in their right places, while reflecting the text conditions in the corresponding objects.

Qualitative and quantitative evaluations show that NoiseCollage outperforms several state-of-the-art models. These results indicate that the crop-and-merge operation of noises is a reasonable strategy to control image generation. We also show that NoiseCollage can be integrated with ControlNet to use edges, sketches, and pose skeletons as additional conditions. Experimental results show that this integration boosts the layout accuracy of ControlNet.

Future work will focus on more efficient layout control. This paper assumes that the layout conditions are given as bounding boxes or polygons. If it is possible to infer possible layout conditions automatically from the given text conditions, it is beneficial for users of NoiseCollage. It is also beneficial if NoiseCollage is extended to accept point annotations, which specify object locations just by points, instead of boxes and polygons. Another research direction is understanding the properties of noise representation against various operations. NoiseCollage shows that cropping and merging (i.e., partial blending) operations realize natural image controls; if we find that applying rigid or non-rigid geometric operations to cropped noises is still possible, we can generate, for example, multi-object videos.

Acknowledgement This work was supported by JSPS (JP22H00540, JP22H05172, JP22H05173).

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended Diffusion for Text-driven Editing of Natural Images. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18187–18197, 2021. 1, 3

[2] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics (TOG)*, 42 (4):1–11, 2022. 1, 3

[3] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xiaoyue Yin. Spatext: Spatio-textual representation for controllable image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18370–18380, 2023. 2, 6

[4] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 1, 2, 3, 4, 6, 8

[5] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 2

[6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:172–186, 2018. 7

[7] Jiaxin Cheng, Xiao Liang, Xingjian Shi, Tong He, Tianjun Xiao, and Mu Li. Layoutdiffuse: Adapting foundational diffusion models for layout-to-image generation. *arXiv preprint arXiv:2302.08908*, 2023. 2

[8] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. 2

[9] Guillaume Couairon, Marlene Careil, Matthieu Cord, Stéphane Lathuilière, and Jakob Verbeek. Zero-shot spatial layout conditioning for text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2174–2183, 2023. 1, 3

[10] Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. *arXiv preprint arxiv:2105.05233*, 2021. 2

[11] 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*, pages 89–106, 2022. 2

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

[13] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3, 4

[14] Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[15] Chengyou Jia, Minnan Luo, Zhuohang Dang, Guangwen Dai, Xiaojun Chang, Mengmeng Wang, and Jingdong Wang. SSMG: Spatial-Semantic Map Guided Diffusion Model for Free-form Layout-to-Image Generation. *arXiv preprint arXiv:2308.10156*, 2023. 2

[16] Yunji Kim, Jiyoung Lee, Jin-Hwa Kim, Jung-Woo Ha, and Jun-Yan Zhu. Dense text-to-image generation with attention modulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7701–7711, 2023. 1, 3, 6

[17] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 5, 7

[18] Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2

[19] Jiafeng Mao and Xueting Wang. Training-Free Location-Aware Text-to-Image Synthesis. *arXiv preprint arXiv:2304.13427*, 2023. 1, 3

[20] Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Guided Image Synthesis via Initial Image Editing in Diffusion Model. *Proceedings of the 31st ACM International Conference on Multimedia*, 2023. 1, 3

[21] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021. 2, 3

[22] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171, 2021. 2

[23] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2

[24] Quynh Phung, Songwei Ge, and Jia-Bin Huang. Grounded Text-to-Image Synthesis with Attention Refocusing. *arXiv preprint arXiv:2306.05427*, 2023. 1, 3

[25] Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*, 2021. 2, 6

[27] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 2

[28] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 11

[29] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 2

[30] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2

[31] Vishnu Sarukkai, Linden Li, Arden Ma, Christopher R'e, and Kayvon Fatahalian. Collage diffusion. *arXiv preprint arXiv:2303.00262*, 2023. 1, 2, 3, 6, 8, 11

[32] Jascha Narain Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265, 2015. 2

[33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[34] Jiayu Xiao, Liang Li, Henglei Lv, Shuhui Wang, and Qingming Huang. R&B: Region and Boundary Aware Zero-shot Grounded Text-to-image Generation. *arXiv preprint arXiv:2310.08872*, 2023. 1, 3

[35] Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. BoxDiff: Text-to-Image Synthesis with Training-Free Box-Constrained Diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7452–7461, 2023. 1, 3

[36] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22428–22437, 2023. 2

[37] Han Xue, Zhi Feng Huang, Qianru Sun, Li Song, and Wenjun Zhang. Freestyle Layout-to-Image Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14256–14266, 2023. 2

[38] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023. 1, 3

[39] Guanhua Zhang, Jiabao Ji, Yang Zhang, Mo Yu, T. Jaakkola, and Shiyu Chang. Towards Coherent Image Inpainting Using Denoising Diffusion Implicit Models. In *International Conference on Machine Learning*, 2023. 2

[40] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2, 3, 7, 8, 11, 14, 15, 16

[41] X. Zhang, Jiaxian Guo, Paul Yoo, Yutaka Matsuo, and Yusuke Iwasawa. Paste, Inpaint and Harmonize via Denoising: Subject-Driven Image Editing with Pre-Trained Diffusion Model. *arXiv preprint arXiv:2306.07596*, 2023. 1, 3

[42] Zhiyuan Zhang, Zhitong Huang, and Jingtang Liao. Continuous Layout Editing of Single Images with Diffusion Models. *arXiv preprint arXiv:2306.13078*, 2023. 1, 3

[43] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A Unified Predictor-Corrector Framework for Fast Sampling of Diffusion Models. *arXiv preprint arXiv:2302.04867*, 2023. 4