Diffusion models have demonstrated impressive performance in text-guided image generation. To leverage the knowledge of text-guided image generation models in image editing, current approaches either fine-tune the pretrained models using the input image (e.g., Imagic) or incorporate structure information as additional constraints into the pretrained models (e.g., ControlNet). However, fine-tuning large-scale diffusion models on a single image can lead to severe overfitting issues and lengthy inference time. The information leakage from pretrained models makes it challenging to preserve the text-irrelevant content of the input image while generating new features guided by language descriptions. On the other hand, methods that incorporate structural guidance (e.g., edge maps, semantic maps, keypoints) as additional constraints face limitations in preserving other attributes of the original image, such as colors or textures. A straightforward way to incorporate the original image is to directly use it as an additional control. However, since image editing methods are typically trained on the image reconstruction task, the incorporation can lead to the identical mapping issue, where the model learns to output an image identical to the input, resulting in limited editing capabilities. To address these challenges, we propose a text-to-image editing model with Image Information Removal module (IIR) to selectively erase color-related and texture-related information from the original image, allowing us to better preserve the text-irrelevant content and avoid the identical mapping issue. We evaluate our model on three benchmark datasets: CUB, Outdoor Scenes, and COCO. Our approach achieves the best editability-fidelity trade-off, and our edited images are approximately 35% more preferred by annotators than the prior-arts on COCO.
Figure 1: Our task aims to edit the specific content of the input image according to text descriptions while preserving the text-irrelevant content of the image. Prior work based on large-scale diffusion models has followed two major approaches to edit the image: (A), fine-tuning the pretrained models or text embeddings (e.g., Imagic [1] or Dreambooth [2]), or (B), introducing structural guidance as additional constraint to control the spatial information of the generated image (e.g., ControlNet [3] or MaskSketch [4]). In our work, shown in (C), our approach conditions on both the original image and the structural guidance, to better preserve the text-irrelevant content of the image. E.g., our model successfully preserves the original attributes of the airplane (outlined by the green bounding box) in the generated image. In contrast, previous methods such as Imagic (A) and ControlNet (B) not only alter the sky and background but also modify the attributes of the airplane (outlined by the red bounding boxes), which is unwanted in this example.

Prompt: Airplane is flying in the sunset.

Pre-trained Diffusion Model

Input Image

Figure 1: Our task aims to edit the specific content of the input image according to text descriptions while preserving the text-irrelevant content of the image. Prior work based on large-scale diffusion models has followed two major approaches to edit the image: (A), fine-tuning the pretrained models or text embeddings (e.g., Imagic [1] or Dreambooth [2]), or (B), introducing structural guidance as additional constraint to control the spatial information of the generated image (e.g., ControlNet [3] or MaskSketch [4]). In our work, shown in (C), our approach conditions on both the original image and the structural guidance, to better preserve the text-irrelevant content of the image. E.g., our model successfully preserves the original attributes of the airplane (outlined by the green bounding box) in the generated image. In contrast, previous methods such as Imagic (A) and ControlNet (B) not only alter the sky and background but also modify the attributes of the airplane (outlined by the red bounding boxes), which is unwanted in this example.

and embeddings may contain unexpected visual artifacts due to the information leakage and fail to preserve the text-irrelevant content of original image [9]. Structural-guided methods also meet pitfalls: since structural guidance usually contains no information about colors or textures, these frameworks have difficulty preserving the text-irrelevant content of the original image. As outlined by red bounding boxes in Figure 1(A) and (B), we observe both Imagic [1] and ControlNet [3] fail to preserve the text-irrelevant content of the original image: Imagic modifies the shape of the airplane while ControlNet changes the color and textures of airplane.

To address the aforementioned issues, we propose introducing the original image as an additional control to our model. In this case, the model fully incorporates the content of the input image, allowing it to effectively preserve the text-irrelevant content. The major challenge of introducing the original image as input is the identical mapping issue [10], in which the model simply maps the input directly to the output. This occurs because editing models are typically trained on the image reconstruction task, where an identical mapping can perfectly optimizes the loss function. Previous approaches typically mitigate this issue by learning disentangled features [11, 13] or using attribute classifier to remove the target attribute [14, 15]. These methods require additional attribute labels and are constrained to specific application scenarios. For instance, in Figure 1, the input image only contains text annotations and lacks corresponding attribute labels such as “daylight” or “sunset”. Therefore, these methods cannot be applied to convert the input image from “daylight” to “sunset”.

To introduce the original image as input and avoid the identical mapping issue, we propose an Image Information Removal module (IIR-Net) to partially remove the image information from input, which is illustrated in Figure 1(C). We erase the image information from two aspects. First, we localize the Region of Interest (RoI) and erase the color-related information. Second, we add Gaussian noise to the input image to randomly eliminate the texture-related information. We adjust the noise intensity for various tasks. E.g., in color editing tasks, we set the noise intensity to zero since we would like to preserve most information from the input image except the color. In texture editing task, we set the noise intensity to a high value since we would like eliminate most information in the target

1We refer to the modified regions of the target image as RoI. In our work, RoI is localized by Grounded-SAM [16, 17]. For tasks that the entire image is subject to modification such as scene attribute transfer or style transfer, we simply define the entire image as the RoI.
region except the structural prior. Given the original image, we concatenate the structure map with attribute-excluded features as additional controls to our model. By our image information removal module, the input to our model is different from the output, thus avoiding the identical mapping issue.

In summary, the contributions of this paper are:

- We introduce the original image as an additional guidance to pretrained generative diffusion models for image editing tasks. Compared to existing image editing frameworks [1, 3] based on pretrained models, IIR-Net effectively preserves the text-irrelevant content of the input image while generating desired features according to the language descriptions.
- We propose an image information removal module to solve the identical mapping issue [10]. IIR-Net partially erases the image information such as colors or textures from the input, and reconstruct the original image according to text descriptions and attribute-excluded features. Compared to prior work [12, 14] to solve this issue, IIR-Net does not require attribute labels to learn disentangled features or attribute classifiers, thus can be applied to images without attribute labels.
- We conduct quantitative and qualitative experiments on three public datasets CUB [18], COCO [19], and Outdoor Scenes [20]. Our experimental results demonstrate that our model improves the fidelity-editability tradeoff over current state-of-the-art approaches in a faster speed. E.g., compared to Imagic [1], IIR-Net improves the LPIPS score from 0.57 to 0.30 on COCO, with a speed improvement of two orders of magnitude.

2 Related Work

Feed-forward transformation image generation and editing. In early work, text-to-image generation and editing approaches typically train a text-to-image generator based on conditional GANs [21–23, 10, 24, 26]. Restricted by the scalability of Conditional GAN and image datasets, these methods are limited to specific image domains and language descriptions. To achieve a good performance on real-world text-to-image generation, current approaches typically train conditional diffusion models [6, 27] on massive datasets (e.g., LAION-400M [28]). Due to the difficulty to obtain image pairs before and after editing, current image editing frameworks [3, 4, 1, 2, 9, 29] are mostly developed based on pretrained text-to-generation models [5, 7]. However, among these methods, frameworks that leverage the feed-forward transformation mechanism mostly focus on structural guidance. E.g., ControlNet leverages structure maps like edge map, semantic map, or pose estimation to control the spatial structure of generated images, and MaskSketch [4] uses sketch as additional control to generate images. Therefore, these methods cannot preserve the other attributes of the image such as colors or textures well, and may result in significant deviation from the input image. To solve this issue, we incorporate the original image as input to our model and propose an image information removal to solve the identical mapping issue [10].

Optimization-based Methods Optimization-based methods update network parameters on each image input. Prior work has demonstrated these methods work well for image generation [30, 32]. In recent years, researchers have proposed various methods [33–35, 3] that leverage CLIP [36] as a constraint to guide the embedding features of predicted images. Besides, inspired by the success of pretrained text-to-image generation frameworks [7, 6, 27], methods (e.g., Imagic [1], Dreambooth [2], SINE [9], Textual Inversion [37]) that are finetuned from pretrained generative models have also been proposed. Compared to feed-forward transformation methods [3, 4], these models with pre-trained generators do not suffer from significant information loss from the original image since they take the whole image instead of the structure map as additional guidance. However, since the finetuning process can be difficult to control, original content such as background or irrelevant attributes of target objects may still be changed in this process, as we will show in Section 4.2. Besides, the inference time of these optimization-based methods is much longer than feed-forward transformation methods due to the finetuning process.

3 IIR-Net: Text-to-Image Editing by Image Information Removal

Given an input image \(x\) and its corresponding text prompt \(S\), our task aims to create the desired content according to \(S\) while preserving the text-irrelevant content of \(x\). To achieve this, we incorporate the original image \(x\) as an additional control to pretrained text-to-image generation model, which is discussed in Section 3.1. However, since the model is trained on the image reconstruction task, the
Figure 2: IIR-Net Overview. Our model mainly consists of two modules: (A) Conditional Latent Diffusion Model: We introduce the original image \( x \) as additional control to our model to preserve the text-irrelevant features of \( x \). See Section 3.1 for detailed discussion; (B) Image Information Removal Module: We erase the image information mainly by two operations. First, we convert RGB values to the gray values in the RoI to exclude the color information. Second, we add Gaussian noise to the input image to partially erase the texture-related information. This module is applied to address the identical mapping issue. See Section 3.2 for detailed discussion.

incorporation of the original image can lead to the identical mapping issue, in which the model simply maps the input image as the output. To address this challenge, we propose our image information removal module in Section 3.2. Figure 2 provides an overview of our approach.

### 3.1 Conditional Latent Diffusion Model

As discussed in the Introduction, preserving the text-irrelevant content of the original image is critical for text-to-image editing. Leveraging the structural guidance as an additional hint (e.g., ControlNet [3], MaskSketch [4]) can lead to significant information loss from the original image. To address this, we introduce the original image as additional control to our model, which preserves all information from the input image. In this section, we first introduce the pretrained text-to-image generation model, Stable Diffusion [5], as preliminaries to our method, and discuss our IIR component in Section 3.2.

Given an input image \( x_0 \) and its corresponding noisy image \( x_T \). Stable Diffusion [5] consists of a series of equally weighted denoising autoencoders \( \epsilon_\theta(x, t) \), where \( t \) ranges from 1 to \( T \). The denoising autoencoders are trained to predict the noise \( \epsilon \) in \( x_T \) according to time step \( t \) and noisy input \( x_t \). The objective function of Stable Diffusion is

\[
L_{CLDM} := \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1), t} \left[ \| \epsilon - \epsilon_\theta(z_t, t) \|_2^2 \right],
\]

where \( \mathcal{E} \) is the pretrained encoder of VQGAN [38] to encode image \( x_t \) to latent features \( z_t \), and vice versa. For conditional generation, the denoising autoencoders \( \epsilon \) take \( \tau_\theta(y) \) as additional input. Here, \( \tau_\theta(y) \) represents a domain-specific encoder to extract feature embeddings from the condition \( y \), where \( y \) can be text prompts, semantic maps, among others. Given image-condition pairs, the Conditional Latent Diffusion Model (CLDM) is optimized by

\[
L_{CLDM} := \mathbb{E}_{\mathcal{E}(x, y) \sim \mathcal{N}(0,1), t} \left[ \| \epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y)) \|_2^2 \right],
\]

where \( \tau_\theta, \epsilon_\theta \) are jointly optimized. In our model, the condition \( y \) consist of text descriptions \( S \) and the original image \( x_0 \) and is defined as

\[
\tau_\theta(y) := \{ \tau_{\theta_1}(S), \tau_{\theta_2}(R(x_0)) \},
\]

where we use the CLIP model [39] as \( \tau_{\theta_1}(\cdot) \) to encode the text descriptions \( S \) and use ControlNet [3] as \( \tau_{\theta_2}(\cdot) \) to encode the feature \( R(x_0) \). Here, \( R(\cdot) \) denotes our image information removal module, which will be discussed in the following Section.

### 3.2 Image Information Removal

As discussed in the Introduction, training solely on image reconstruction can lead to the identical mapping issue. Previous approaches address this issue by learning disentangled features [20] or
attribute classifiers [14]. However, these methods require annotated attributes, restricting their application scenarios. To overcome this challenge, we propose our image information removal module, which incorporates color and texture removal operations. Our removal operations effectively mitigates the identical mapping issue without the requirement for additional annotated labels.

Color-related Information Removal. In Figure 2(B), we present our color information removal operation. Given the input image \( x_0 \) and its corresponding text prompt \( S \), we employ Grounded-SAM [16, 17] to localize the RoI. The color information of \( x_0 \) is then erased by

\[
x'_0 = \text{rgb2gray}(x \odot m_{\text{RoI}}) + x \odot (1 - m_{\text{RoI}}),
\]

where \( m_{\text{RoI}} \) is the segmentation mask detected by Grounded-SAM.

Through the application of color-related information removal to the input image \( x_0 \), our model demonstrates proficiency in color-related editing tasks, such as transforming a "white airplane" into a "green airplane." However, as depicted in Figure 3, the model encounters challenges when attempting to modify texture-related information, such as changing "lawn" to "sand." To address this limitation, we introduce our texture-related information removal module.

Texture-related Information Removal. We destroy the texture-related information by adding noise to the image condition \( x'_0 \) of our model, denoted by

\[
q(x'_k | x'_0) = \prod_{k=1}^{K} q(x'_k | x'_{k-1}); \quad q(x'_k | x'_{k-1}) = \mathcal{N}(x'_k; \sqrt{1 - \beta_k} x'_{k-1}; \beta_k \mathbf{I}),
\]

where \( k \) denotes the time step applied to \( x'_k \), which is different from the time step \( t \) applied to \( x_t \). Note that \( x_t \) is obtained by adding noise to the original image \( x_0 \) in diffusion models, whereas \( x'_k \) is obtained by adding noise to the image condition \( x'_0 \) in diffusion models. In the training process, we randomly sample \( x'_k \) from \( \{x'_0, \ldots, x'_K\} \).

While \( x'_k \) inherently preserves the structure information of \( x_0 \), we find that explicitly incorporating additional structural guidance, such as Canny Edge, can help the model better capture the structural information. Therefore, we concatenate \( x'_k \) with Canny Edge \( C(x_0) \). The output of our image information removal module is defined as:

\[
R(x_0) = [x'_k, C(x_0)].
\]

Given the output of our image information removal module \( R(x_0) \), the objective function of our IIR-Net is defined as

\[
L_{\text{IIR-Net}} := \mathbb{E}_{\mathbf{x}, y, \epsilon \sim \mathcal{N}(0,1), t} \left[ \| \epsilon - \epsilon_{\theta}(x_t, t, \tau_{\theta_1}(S), \tau_{\theta_2}(R(x_0))) \|^2 \right].
\]

4 Experiments

4.1 Datasets and Experiment Settings

Datasets. We evaluate the performance of our model on three standard datasets, CUB [18], Outdoor Scenes [20], and COCO [19]. CUB [18] is an image dataset annotated with 200 bird species. We split the dataset into 8,855 training images and 2,933 test images. Outdoor Scenes [20] contains 8,571 images captured from 101 webcams, with each webcam collecting 60~120 images showcasing different attributes like weather, season, or time of day. COCO [19] contains 82,783 training images and 40,504 validation images. Following [1], we randomly select 150 test images from each dataset to evaluate the performance of each method.

Metrics. Following [1], we adopt perceptual metric (LPIPS) [31] and CLIP scores [36] as our quantitative metrics. LPIPS measures the image fidelity and CLIP evaluates the model’s editability. Besides, we perform quantitative experiments by user study and inference time to evaluate the effectiveness and efficiency of our model.

Baselines. We compare IIR-Net with three state-of-the-art approaches: Text2LIVE [3], Imagic [1], and ControlNet [3]. For Text2LIVE, we set the optimization steps to 600. For Imagic, both the text embedding optimization steps and model fine-tuning steps are set to 500. We sample the interpolation hyperparameter \( \eta \) from 0.1 to 1 with a 0.1 interval, and the guidance scale is set to 3. For ControlNet and IIR-Net, we generate images with a CFG-scale of 9.0, and DDIM steps of 20 by default.
This bird has a yellow body, with blue wings and red crown.

A small bird sports its white belly and black head.

Bird with a red body, red wings, and black wings.

Winter, Snow, Cold

Summer, Daylight

Evening, Sunset

Figure 3: Qualitative comparison on CUB and Outdoor Scenes. From top to bottom: input image, Text2LIVE [8], Imagic [1], ControlNet [3], and ours. Generated images have 512 pixels on their shorter side. See Section 4.2 for discussion.

4.2 Qualitative Results

Entire-image Editing on the CUB and Outdoor Scenes Datasets.

Figure 3 presents a qualitative comparison of the edited images generated by our model and the baselines. In Figure 3(A), we present a comparison on the CUB [18] dataset. We observe that our model can accurately manipulate parts of the bird while preserving the text-irrelevant content of the original image. For example, in the first column of Figure 3(A), while baselines such as ControlNet and Imagic can recognize “yellow” and “blue” from the text prompt, they both fail to effectively apply them to the correct corresponding parts of the bird. Imagic generates a bird with a blue crown and yellow wings, while ControlNet generates a blue head and a red breast. In contrast, our model accurately edits the bird by parts according to the prompt and produce a bird with blue wings, yellow body, and red crown. Besides, we observe that the background of images generated by Imagic and ControlNet has been changed. This is due to the fact that Imagic and ControlNet do not directly use the original image as their input. E.g., Imagic optimizes the text embeddings to get features that reflect the attributes of the original image, and ControlNet uses the Canny Edge map as input. Thus, it is challenging for these method to preserve the text-irrelevant content of the original image. In contrast, our model takes the original image as input and only erases the text-relevant content, thus preserving the text-irrelevant content effectively.

In Figure 3(B), we present a comparison on the Outdoor Scenes [20] dataset. Consistent with our findings on the CUB dataset, we observe that baselines like Imagic and ControlNet tend to modify the text-irrelevant contents of the original image, such as the textures and background, while Text2LIVE only introduces limited visual effects to the original image and may fail to generate images aligned with the text descriptions. For instance, in the second column of Figure 3(B), images produced by Imagic and ControlNet are well aligned with text descriptions (“summer”, “daylight”), but they introduces unexpected objects such as trees or a lake to the image. On the other hand, Text2LIVE preserves the original image well, but fails to align with text descriptions, as seen with the snow-covered field in summer. In contrast, our method effectively modifies the desired content, such as changing “winter” to “summer”, while preserving the original content of the image.
Figure 4: Qualitative comparison on the COCO dataset. We present various editing tasks in this figure. From top to bottom: color editing, scene attribute transfer, texture editing, and style transfer. Generated images have 512 pixels on their shorter side. We outline the objects that subject to modification in red bounding boxes and objects that should be preserved in green bounding boxes. See Section 4.2 for discussion.

**Region-based Image Editing on the COCO Dataset.** Compared to object-centric datasets such as CUB or Outdoor Scenes, images from COCO may contain complex scenes with multiple objects. In some cases, only parts of the input image need to be modified. Thus, we apply Grounding-DINO [16] and SAM [17] to localize the Region of Interest (RoI) that requires editing. We present a qualitative comparison of our method and the baselines on various image editing tasks in Figure 4. From the figure, we observe that our method produces images that are well-aligned with text descriptions while maintaining consistency with the original images. E.g., in the color editing task, although Imagic and ControlNet are able to generate a blue bus according to the text prompt, Imagic changes the original shape of the bus and ControlNet modifies the texture of the bus. In contrast, our method only modifies the color attribute while preserving the irrelevant attributes. Furthermore, our model generates images that appear more natural and visually appealing compared to the baselines. E.g., in the scene attribute transfer task, the visual effect of “sunset” brought by our model is naturally aligned with the original image, whereas Text2LIVE introduces obvious artificial effects to the airplane. Finally, we evaluate our model on tasks where ControlNet has originally performed well, such as texture editing and style transfer. Our results show that adapting text-to-image generation models to image editing tasks does not significantly compromise their original capabilities. For instance, although COCO does not contain image pairs for style transfer, our method still retains the ability to transfer a photorealistic image to an artistic style image, as seen in the style transfer example. We provided additional examples in appendices.

**Ablation Study.** In Figure 5, we provide ablation study of IIR-Net. We find that without our unsupervised image content removal mechanism, the model always outputs the input image as the predicted image, i.e., the identical mapping issue [10]. E.g., the images in the blue bounding box remain white airplane and green grass, showing a lack of alignment with the text descriptions. By incorporating the color removal mechanism (see images with low noise level), our model performs well on tasks such as color editing. For example, when changing the airplane’s color from white to green, our model preserves the most of the airplane’s attributes, only modifying the color. We observe that the color removal mechanism can find texture editing challenging. For instance, as seen in the second row of the figure, the images generated with low noise level still exhibit the grass texture.

---

Since Text2LIVE and Imagic automatically localize the RoI, we apply Grounding-DINO and SAM to ControlNet and our method.
Figure 5: Ablation Study. We perform experiments to evaluate the effectiveness of our color removal and texture removal operations. Images generated without using our image information removal module are outlined by the blue bounding box. See Section 4.2 for discussion.

<table>
<thead>
<tr>
<th></th>
<th>CUB</th>
<th>Outdoor Scenes</th>
<th>COCO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LPIPS ↓</td>
<td>CLIP ↑</td>
<td>LPIPS ↓</td>
</tr>
<tr>
<td>Imagic [1]</td>
<td>0.406</td>
<td>27.03</td>
<td>0.551</td>
</tr>
<tr>
<td>Text2live [8]</td>
<td>0.162</td>
<td>30.37</td>
<td>0.218</td>
</tr>
<tr>
<td>ControlNet [3]</td>
<td>0.528</td>
<td>29.49</td>
<td>0.618</td>
</tr>
<tr>
<td>ours</td>
<td>0.138</td>
<td>29.57</td>
<td>0.479</td>
</tr>
</tbody>
</table>

Table 1: Quantitative experiments of image manipulation on CUB [18], Outdoor Scenes [20], and COCO [19] datasets. CLIP [36] is used to evaluate the image editing performance and LPIPS is applied to evaluate image fidelity. Generated images have been resized to 224×224 resolution for CLIP score. We use the “ViT-B/32” version of CLIP. See Section 4.3 for discussion.

instead of the intended "sand" texture. Therefore, we incorporate noise augmentation to the input images to better handle such editing tasks. As shown in the second row, our model successfully modifies the grass texture to sand under high-level noise conditions. In practical applications, users can adjust the noise level according to different editing tasks to achieve optimal performance.

4.3 Quantitative Results

Editability-fidelity Tradeoff. Table 1 reports our quantitative results on CUB, Outdoor Scenes, and COCO. As observed in our qualitative experiments, our model achieves a better tradeoff between image fidelity and editability compared to other state-of-the-art methods. For instance, our model achieves the best LPIPS scores (0.138 and 0.301) and comparable CLIP scores (29.57 and 24.30) on CUB and COCO. In Outdoor Scenes, our model achieves the highest CLIP score and the second best LPIPS score. Text2LIVE achieves better LPIPS score than our method on Outdoor Scenes. However, it may due to the fact that Text2LIVE mainly augment the scenes with new visual effects, rather than directly modifying the attributes of the scenes. For example, Text2LIVE fails to change the grassland to a snowy landscape or convert lush trees to bare ones in the scenes.

User Study. We conducted a user study to quantitatively evaluate the performance of IIR-Net, as shown in Table 2. We randomly selected 30 images from COCO and applied each model to generate the modified images, resulting in a total of 120 generated images. Each image was annotated three times by users and we asked our annotators to judge whether the image is correctly manipulated based on the text guidance while preserving the text-irrelevant content of the original image. In the table, we report that IIR-Net significantly outperforms baselines. See appendices for additional details about our user study.

Inference Time Table 2 presents a comparison of the inference time and their standard error between our model and the baselines. In our experiments, we apply Stable Diffusion v1.5 [5] as the backbone to Imagic, ControlNet, and our method. All methods are benchmarked on a RTX A6000. The table shows that our method has significantly faster inference times compared to Imagic, with a speed improvement of two orders of magnitude when processing 512×512 images. In addition, our method
Figure 6: Failure cases. Inconsistency with the original image in non-rigid image editing task (left), or fails to change the brightness of the image (right). See Section 5 for discussion.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>User Preference</td>
<td>30.0%</td>
<td>23.3%</td>
<td>33.3%</td>
<td>68.3%</td>
</tr>
<tr>
<td>Inference Time</td>
<td>281.6±1.72s</td>
<td>483.4±1.31s</td>
<td>5.0±0.04s</td>
<td>5.0±0.03s</td>
</tr>
</tbody>
</table>

Table 2: We randomly select 30 images from COCO for user study and speed evaluation. Top row reports user judgments on the correctness of the image manipulation. Bottom row reports speed for our method v.s. the baselines. Our method has negligible overhead compared to ControlNet, and is significantly faster than Text2LIVE and Imagic. See Section 4.3 for discussion.

is approximately 50x faster than Text2LIVE. We observe both ControlNet and our method have around 5s inference time, validating that our approach introduces negligible overhead to ControlNet.

5 Limitations & Broader Impacts.

Limitations. We identify two failure cases of our methods in this section: First, the attributes of the original image are likely to be modified in non-rigid image editing tasks. Second, it is challenging for our method to change the brightness of the input image drastically. We present examples of these two failure cases in Figure 6. As shown in the left-hand side, though our method can achieve non-rigid image editing according to the input image and a modified structural guidance, we observe that the model fails to map some attributes to the correct parts. E.g., the bird of the input image has a grey crown while the edited image generate a bird whose head is gray. The color of wings is also slightly different from the input bird. In the right-hand side, we find that our model fails to change the brightness of the image in some cases. E.g., the input image is a night view. Therefore, the brightness of the image is low in this image and the model tend to reconstruct an image with a low brightness even if the target text is “daylight”, “sunny”.

Broader Impacts. Our model is designed to perform image editing according to user-provided language descriptions. Thus, it enables modification of attributes such as colors, textures, or styles in the original images. As other image generation and editing approaches, our model may be used to synthesize images that contains misinformation. Therefore, it is important for practitioners to review and control how images are manipulated to avoid misinformation. Further research on detecting machine-generated images is needed to mitigate this potential issue.

6 Conclusion

In this paper, we propose IIR-Net, a text-to-image editing model that incorporates the original image by selectively erasing the image information. IIR-Net mainly consists of two stages: an conditional diffusion model that takes the original image as additional control, and an image information removal module to address the identical mapping issue. We demonstrate that IIR-Net outperforms the state-of-the-art in both qualitative and quantitative evaluations on CUB, Outdoor Scenes, and COCO datasets. For instance, compared to Imagic, IIR-Net improves the LPIPS score from 0.57 to 0.30 and the CLIP score from 21.53 to 24.30 on COCO, with a speed improvement of two orders of magnitude. We also use qualitative examples to demonstrate the effectiveness of our model on various image editing tasks, validating that our model can modify the target attribute according to language descriptions while preserving the text-irrelevant content of the original image well.

Acknowledgements. This work was supported in part by DARPA under agreement number HR00112020054.
References


Appendices

A Implementation Details.

We initialized our model weights from Stable Diffusion 1.5 [5] and ControlNet [3]. During training, we applied a batch size of 8 and a maximum learning rate of $1 \times 10^{-6}$. We finetuned our models approximately 100 epochs on the CUB [18] dataset, and around 5 epochs on the Outdoor Scenes [20] and COCO [19] datasets. The training process was parallelized on 2 NVIDIA RTX-A6000s. To adapt the image conditions in our model, we configured the channel of the image encoder block to 4, with 3 channels for RGB images and 1 channel for Canny edge map. We unlocked the Stable Diffusion decoder for the CUB dataset, as these images primarily focus on various birds and adhered to a consistent style. We froze the Stable Diffusion Decoder for the Outdor Scenes and COCO datasets, since these datasets comprising natural images with diverse objects and varying styles. Following [3], during inference we set the CFG-scale to 9 and the diffusion steps to 20 by default.

B Additional Experiment Results

We present additional qualitative results in Figure 7 and 8 to supplement the main paper. The results demonstrate that IIR-Net is able to modify image content base on user prompts while preserving the text-irrelevant content of the original image. In Figure 7 we observe that IIR-Net successfully preserves shape-related information of the target object in texture editing examples (e.g., “A wood airplane” and “A woman skiing on grassland”), as well as texture-related information in color editing examples (e.g., “A red horse.” and “A green orange”). In contrast, Imagic [1] may modify the shape information, while ControlNet [3] may modify the texture information. Besides, we observe that our network produces visually more natural images compared to Text2LIVE [8]. E.g., in the example of “A red horse.”, Text2LIVE applies some red effects to the horse, whereas our method directly produces “a red horse” with better consistency to the background. These observations are consistent with our conclusions in the main paper.

C User study Interface

In our user study experiments, annotators were presented with an input image, a target text, and four edited images generated by different methods. They were asked to evaluate the accuracy of manipulated images according to two aspects: (1) the alignment of the image with the target text, and (2) the preservation of text-irrelevant content from the original image. We provide a sample screenshot in Figure 9.
Figure 7: Additional comparison results between IIR-Net and the baselines on the COCO dataset. We set the image resolution to $512 \times 512$. We observe that our method effectively modifies the input image while preserving the text-irrelevant content. For instance, in the example of “A tennis player in blue shirt.”, IIR-Net retains both the shape and texture attributes of the original shirt, whereas the other baselines either introduce limited visual effects or modify text-irrelevant content such as textures or shape. See Section B for further discussion.
Figure 8: Additional comparison results between IIR-Net and baselines on the COCO dataset. We set the image resolution to $512 \times 512$. See Section B for discussion.
Image Editing User study

Choose the edited image that accurately represents the text descriptions while preserving the text irrelevant content from the original input image. The evaluation does NOT need to take into account the size of the image. The options for the question are multiple-choice.

Text Description: Blue bus on street.

Figure 9: User study screenshot. A sample screenshot illustrating one of the questions presented to participants in our user study. See Section C for discussion.