On Manipulating Scene Text in the Wild with Diffusion Models

Joshua Santoso
Independent Researcher
janojoshua@gmail.com

Christian Simon
The Australian National University
christian.simon@anu.edu.au

Williem
Independent Researcher
williem.pao@gmail.com

Abstract

*Diffusion models have gained attention for image editing yielding impressive results in text-to-image tasks. On the downside, one might notice that generated images of stable diffusion models suffer from deteriorated details. This pitfall impacts image editing tasks that require information preservation e.g., scene text editing. As a desired result, the model must show the capability to replace the text on the source image to the target text while preserving the details e.g., color, font size, and background. To leverage the potential of diffusion models, in this work, we introduce **D**iffusion-**B**as**E**d **S**cene **T**ext manipulation Network so-called DBEST. Specifically, we design two adaptation strategies, namely one-shot style adaptation and text-recognition guidance. In experiments, we thoroughly assess and compare our proposed method against state-of-the-arts on various scene text datasets, then provide extensive ablation studies for each granularity to analyze our performance gain. Also, we demonstrate the effectiveness of our proposed method to synthesize scene text indicated by competitive Optical Character Recognition (OCR) accuracy. Our method achieves 94.15% and 98.12% on COCO-text and ICDAR2013 datasets for character-level evaluation.*

1. Introduction

Scene text manipulation has gained significant attention in computer vision due to its practical applications. The promising application of scene text manipulation is real-time sign translation allowing for instant translation of signs in various languages [6]. Additionally, scene text manipulation can be used to protect privacy by obscuring sensitive information on images [42].

In deep learning, synthetic scene texts can be used to augment data for various training purposes e.g., text classification and detection. For instance, Tang *et al*. [35] used a synthesized dataset yielding an improved model capability for detecting and erasing text in the wild. Though the synthetic data can be collected from image editors using editing software, this labor task is costly and requires expertise to
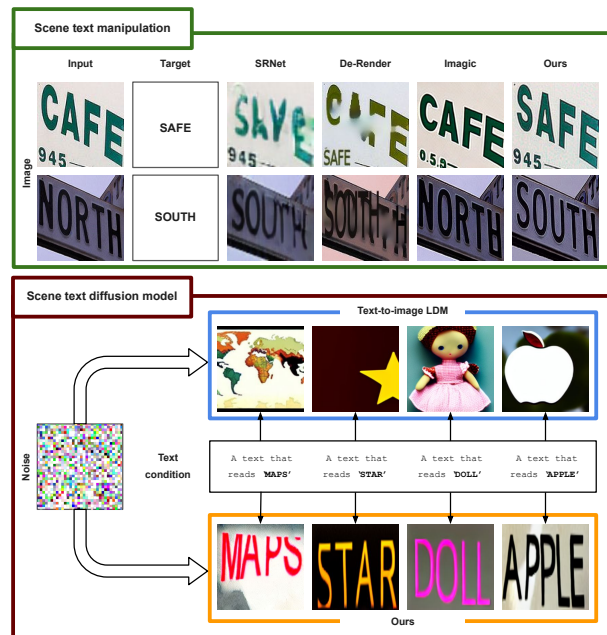

Figure 1. Top: Comparison between state-of-the-art methods and our method from given input image and target text on scene text manipulation. Bottom: Comparison between baseline text-to-image Latent Diffusion Model (LDM) [27] represented with *blue box* and our method represented with *red box* on scene text domain from given random noise and text condition as an input.

generate consistent results. Thus, automatic tools for scene text are highly desirable. However, developing such tools is not easy, especially to ensure the edited text compatible for further processing e.g., preserving the details to avoid performance degradation on Optical Character Recognition (OCR) models.

The automatic way for scene text editing can be achieved using deep generative models [3, 32, 37, 39, 40, 42]. As Generative Adversarial Networks (GANs) [8] progressing, earlier studies have addressed scene text manipulation [32, 37, 39] exploiting the generator and discriminator scheme. Wu *et al*. [37] demonstrated the ability of GANs

by formulating the problem as an image style transfer task, where the models employ the target text while imposing the styles from the original text. Shimoda *et al.* [32] propose to train a text vectorization model and parse text components (*e.g.* style, background, and color) for editing. While effective, the performance of these prior methods to edit the scene text is still unsatisfactory in challenging cases, as shown in Fig. 1. In inference, these prior techniques depend heavily on additional modules *e.g.* character recognition models and synthetic text generators. Moreover, the underlying generative models of these prior works (*i.e.* GANs) have limited capacity to manipulate text with a high degree of variability and complexity.

In this paper, we exploit the diffusion models [11,27,34], which have shown impressive results for text-to-image tasks, an alternative competing method for synthesizing scene texts. Also, compared to GAN solutions, diffusion models do not suffer from training instability and mode-collapse [11]. Especially, the recent advances in diffusion models allow for incorporating multi-modal conditional inputs *e.g.*, generating an image using a text condition known as text-to-image models [21, 26, 27]. These works, especially Latent Diffusion Model (LDM) [27], provide off-the-shelf pretrained-models for generating images from text prompts. Several works (*e.g.*, Imagic [15] and Null-Inversion [20]) have explored the possibility of using the pretrained models for image editing while preserving parts of interests. Even though these prior methods have shown remarkable results for general image editing, implementing this to scene text editing is not straightforward. For example, Fig. 1 (top) shows that the text scene cannot be manipulated using Imagic [15]. Instead of generating scene texts, the prior methods tend to generate scenes with some objects related to the text prompt, as illustrated in Fig. 1 (bottom). In contrast to the aforementioned image editing models, our work focuses on editing scene texts while preserving fine details of text information.

In this work, to manipulate text in the wild, we propose a **D**iffusion-**B**as**E**d **S**cene **T**ext manipulation Network so-called DBEST. Our design uses text prompts; thus, we have the benefit of leveraging the large language model as a condition for image editing. In summary, we list our contributions below:

- We identify the limitation of existing diffusion-based image editing models and introduce a diffusion model akin to the text-inversion technique that performs scene text manipulation. For image generation, our model has no modification over the original LDM, thus allowing easy integration into other text-related pipelines.

- We present a synthetic dataset as a part of our training strategy, omitting the cost to collect the scene text data.

- We introduce the concept of one-shot style adaptation for scene text manipulation, which aims to maintain the source style on the edited image.

- We devise the classifier guidance approach using a text recognition model, yielding a significant improvement in terms of quality and quantity. To highlight, the OCR word accuracy of our proposed network achieves 84.83% on the SynText dataset leading to 13% improvements over the state-of-the-art performance.

2. Related works

Scene text manipulation. The manipulation of scene text can be divided into three sub-fields: text effect transfer, text removal, and text manipulation. Text effect transfer involves transferring a style effect from a source style effect image to a target scene text. Text removal involves removing text from a scene while maintaining the scene's appearance. Finally, text manipulation entails editing a source text image with desired text while preserving the text style and background from the source image.

Specific to text manipulation, STEFFAN [29] introduces a character-level editing text where the input image is first converted into a per-character mask. SRNet [37] introduces a GAN-Based approach for word-level editing where the desired text is first converted to image level. Then, the target text image can be edited through three sequence steps: word geometry transformation, background retrieval, and fusion network. SwapText [39] is the extension of SRNet, where the word geometry transformation is replaced by per-character geometry transformation. De-Render [32] introduces a new approach where the input image is first fed to the rendering parameters prediction model. Then, the reconstruction model uses the predicted parameters to reconstruct the text scene image. Finally, the text can be edited by updating rendered parameter.

Our method shares similarities with SRNet, but while SRNet struggles with editing images using the style transfer concept, we aim to explore techniques such as conditional GAN [22]. Unlike converting the target text to the image level, our method operates at the text level for more efficient editing.

Diffusion model for image manipulation. Diffusion model has shown its capability to beat GAN on image generation. From the given image, several works show impressive work for image-to-image translation [11,27,34]. In further research, It is also possible to control the generated image from the given condition as proven by [1, 15, 16, 18, 20, 27]. In brief, using the pre-trained language-vision model such as CLIP [4, 23], Kim *et al.* [16] shows the excellent result in manipulating a whole image by using CLIP as a loss function. Avrahami *et al.* [1] introduces local editing

guided CLIP loss where the local mask is necessary as the additional input.

Alternatively, Kawar *et al.* [15] and Ron *et al.* [20] utilize the pre-trained model from Latent diffusion model (LDM) [27] and introduce the fine-tune approach. First, Ron *et al.* fine-tuned the unconditional embedding produced by LDM to maintain the style of the input image. Then, the editing process can be done using Prompt-to-prompt (p2p) [10]. Kawar *et al.* proposed two steps: conditional embedding optimization and diffusion model fine-tuning with the optimized embedding. Then, the edited image is generated by interpolating the original and optimized embedding during sampling.

However, we investigate that the success of the methods above relies on language-vision models such as CLIP or LDM. While those methods perform well on the text-to-image, they still fail on the text to scene text domain, as shown in Fig. 1. Therefore, our approach utilizes LDM pre-trained model on text-to-image as our base model. Then, we train the model with a text-oriented image dataset to shift the domain to text-to-scene-text as shown in Fig. 1.

3. Preliminaries

Inspired by the success of the diffusion models [11, 27, 34] in generating high-quality images from a given prompt (text to image), our method employs the concept of diffusion process to generate a sample by multiple denoising steps. Specifically, our model is built on memory-efficient diffusion models *i.e.* the Latent Diffusion Model (LDM), and focuses on performing diffusion for text to scene text generation in the latent space. In this section, we briefly overview the fundamental principles of LDM.

Diffusion models. Diffusion probabilistic models [11,33] mainly consist of two processes: 1) forward steps and 2) reverse steps. Starting from the distribution of an input image $q(\boldsymbol{x}_0)$, where $\boldsymbol{x}_0 \in \mathbb{R}^{3 \times H \times W}$ in the RGB space, the forward steps $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ are a sequence of the Markov chain that outputs a series of noisy image $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T$. The Gaussian noise ϵ is gradually added to the input image $\boldsymbol{x}_0$ based on a variance schedule $\{\beta_1, \cdots, \beta_T\}$. The forward process is defined as:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t, \boldsymbol{I}). \quad (1)$$

As the noisy image $\boldsymbol{x}_T$ must be reconstructed to $\boldsymbol{x}_0$, we need to learn a model using the reverse process estimating the joint distribution $p_\theta(\boldsymbol{x}_{0:T})$ yielding:

$$p_\theta(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|x_t),$$
$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, t), \Sigma_\theta(\boldsymbol{x}_t, t)). \quad (2)$$

Alternatively, the noised image x_t can be sampled directly from x_0. With $\alpha_t := 1 - \beta_t$ and $\alpha_{\bar{t}} := \prod_{s=0}^{T} \alpha_s$, $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ can be formulated as:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_{\bar{t}}}\boldsymbol{x}_0, (1-\alpha_{\bar{t}})\mathbf{I}). \quad (3)$$

Ho *et al.* [11] introduces the approximation noise model $\epsilon_\theta(\boldsymbol{x}_t, t)$ to predict $\mu_\theta(\boldsymbol{x}_t, t)$:

$$\mu_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_{\bar{t}}}} \epsilon_\theta(\boldsymbol{x}_t, t) \right). \quad (4)$$

Diffusion in the latent space. LDM has two modules: perceptual compression and denoising U-Net. The perceptual compression is able to encode the given $\boldsymbol{x}$ into a latent space $\boldsymbol{z} = \mathcal{E}(\boldsymbol{x})$ and is able to reconstruct back from latent to image space $\boldsymbol{x} = \mathcal{D}(\mathcal{E}(\boldsymbol{x}))$. The diffusion model $e_\theta(\boldsymbol{z}_t, t)$ generates the predicted noise of the noisy latent image $\boldsymbol{z}_t$ at time t. The loss function can be formulated as

$$\mathcal{L}_{\text{LDM}} := \mathbb{E}_{\boldsymbol{z}, \epsilon \sim \mathcal{N}(0,1), t} \|\epsilon - \epsilon_\theta(\boldsymbol{z}_t, t)\|_2^2. \quad (5)$$

If the condition (*e.g.* text, image, or label) y exists, then the loss function is presented as

$$\mathcal{L}_{\text{LDM}} := \mathbb{E}_{\boldsymbol{z}, \boldsymbol{y}, \epsilon \sim \mathcal{N}(0,1), t} \|\epsilon - \epsilon_\theta(\boldsymbol{z}_t, t, \tau_\psi(\boldsymbol{y}))\|_2^2, \quad (6)$$

where τ_ψ is a conditional model, *e.g.*, BERT [5] or CLIP [25], that takes $\boldsymbol{y}$ as an input and generates an embedding feature $e \in \mathbb{R}^{k \times d}$. Here, k denotes the number of tokens in $\boldsymbol{y}$, and d represents the feature dimension. This embedding feature then is combined with the latent vector $\boldsymbol{z}$ using a cross attention mechanism in the denoising model ϵ_θ. Note that we use the identical conditioning mechanisms in LDM [27] with the cross-attention mechanism of multi-modalities in the intermediate layers of the U-Net [28].

4. Proposed method

Our proposed method is implemented based on the conditional LDM, which uses a pre-trained model from the large-scale text-to-image LAION dataset [31]. However, generating scene texts remains challenging, as indicated in Fig. 1. In this work, our training process comprises two stages. First, as a pre-requisite, we need to train (a.k.a fine-tune) the model on the text and edited-text pairs. To this end, we use a synthetic text generator (*i.e.*, SynText [9]) to produce scene text pairs. Through this strategy, the model can learn from ground-truth edited scenes in training. We only update the diffusion model ϵ_θ on the SynText dataset while the condition model τ_ψ remains unchanged. Second, in order to manipulate scene text, we have two optimization processes, namely one-shot style adaptation and text recognition guidance as illustrated in Fig. 2.
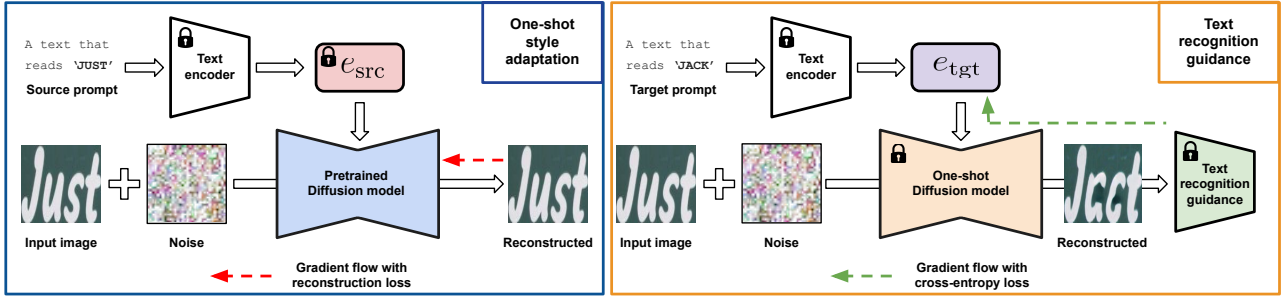
Figure 2. The pipeline of our proposed method. The process is divided into 2 steps. One-shot style adaptation for fine-tuning the diffusion model and text recognition guidance for optimizing the target embedding.

4.1. One-shot style adaptation

To manipulate a scene text, our model receives a source image x and a text prompt y which are mapped to the latent space z and e, respectively. Our goal is to modify the source word to a target word while preserving the style of the source image. To this end, the model has to encode the characteristics of fonts, backgrounds, and colors. In the perspective of the text-to-image generation approach, this requires additional prompts from humans that might be difficult to find exact prompts. A sensible approach to address this problem is to invert the text from a source image [7, 20]. However, we have observed that [15, 20] approaches are suboptimal to maintain the original text style. For this reason, we propose a one-shot style adaptation strategy that directly optimizes the model's weights θ to achieve detail-preserved results :

$$\min_{\theta} \; \|\epsilon - \epsilon_\theta(z_t, t, e_{\mathrm{src}})\|_2^2. \tag{7}$$

Where z_t is the source scene text in latent space. For each source image, we reset the parameters to initial ϵ_θ and optimize the model using Eq. 7 yielding $\tilde{\epsilon}_\theta$. In each iteration, we randomly sample the noise ϵ in Eq. 7.

There are significant differences with prior works for style adaptation or personalization. Our style adaptation mechanism differs from DreamBooth [30] as it does not involve adding a unique identifier to the input text or generating extra images for class-specific prior preservation loss. Another strategy for personalization, known as textual inversion [7], may offer more efficient adaptation but often results in a loss of details. Furthermore, both strategies typically require more samples to learn the style. In contrast, our one-shot style adaptation only requires a single image for style adaptation and seamlessly incorporates input details. Imagic [15] requires optimizing the text embedding before fine-tuning the diffusion model, meaning that each edited prompt necessitates both embedding optimization and fine-tuning. In contrast, our method only requires training the diffusion model once. Even with an edited

prompt, we only need to optimize the embedding without retraining the diffusion model.

4.2. Text recognition guidance

Algorithm 1 Text Recognition Guidance

1: **Input :** Input latent image z_0, target embedding e_{tgt} , and target text $\mathcal{T}$
2: **Output :** Optimized embedding $\tilde{e}_{\mathrm{tgt}}$
3: **for** number of optimization iterations **do**
4:     Get t from Uniform$(1, \cdots, T)$
5:     Generate ϵ from $\mathcal{N}(0, \mathbf{I})$
6:     Generate $z_t = \sqrt{\alpha_t}z_0 + \sqrt{1 - \alpha_t}\epsilon)$
7:     Obtain $\hat{\epsilon} = \tilde{\epsilon}_\theta(z_t, t, e_{\mathrm{tgt}})$
8:     Compute $\hat{x}_{edit}$ from Eq. 8
9:     Update e_{tgt} from Eq. 10
10: **end for**

In addition to the style adaptation strategy, we aim to preserve the readability and performance on OCR. To this end, we introduce text recognition guidance integrated with the diffusion model. We use the style-adapted diffusion model $\tilde{\epsilon}_\theta$ as a starting point. In this stage, we start from the embedding of the target text e_{tgt} and the text recognizer to guide the model to generate precise texts as shown in Fig. 2 (right). Firstly, as the diffusion process is performed in the latent space, we need to reconstruct to the image level as followed:

$$\hat{x}_{edit} = \mathcal{D}(\hat{z}_0), \tag{8}$$

where $\hat{z}_0$ is obtained by a denoising process for each $t \in \{1, \cdots, T\}$ as followed:

$$\hat{z}_0 = \frac{z_t - \sqrt{1 - \alpha_t}\,\tilde{\epsilon}_\theta(z_t, t, e_{\mathrm{tgt}})}{\sqrt{\alpha_t}}. \tag{9}$$

The reconstructed scene text $\hat{x}_{edit}$ is fed into the text recognition model f_ϕ with the cross entropy loss $\mathcal{L}_{\mathrm{CE}}$. At this step, we keep the diffusion model unchanged as it has been adapted to the source style. Thus, we arrive at the textual

Figure 3. Qualitative comparison on COCO-Text [36] and ICDAR2013 [14] datasets. DBEST (ours) achieves superior qualitative results compared to SRNet [37], STEFFAN [29], SDEdit [18], Imagic [15], Null-Inv [20]+p2p [10], Text2Img LDM [27], Text2Live [2].

inversion by employing iterative optimization on the target embedding e_{tgt} as:

$$\tilde{e}_{\text{tgt}} = e_{\text{tgt}} - \nabla_{e_{\text{tgt}}} \mathcal{L}_{\text{CE}}(f_\phi, \hat{x}_{edit}, \mathcal{T}_{\text{tgt}}), \qquad (10)$$

where $\mathcal{T}_{\text{tgt}}$ is the target text. In our implementation, we set the minimum value of $t \geq 500$ to ensure the input image is mainly covered by ϵ. The step-by-step of text recognition guidance is presented in Algorithm 1.

5. Experiments

This section presents a comparison of our method with existing approaches for scene text manipulation, including those with and without a diffusion model. Also, we include ablation studies to understand each component of our proposed approach. To begin, we introduce the datasets, SOTA methods, and provide experimental details.

Datasets. We employ two types of datasets for our evaluation: synthesized and in-the-wild datasets. We generate the synthesized text scene dataset. The background and the text variation are obtained from SynText [9]. Then, we follow the step-by-step procedures mentioned in SRNet and produce 100k training images. For evaluation data, the backgrounds are gathered from publicly available images from the internet, and various fonts are used to produce 600 images. The in-the-wild datasets we use are COCO-Text [36] and ICDAR2013 [14], which are frequently used for quantitative comparison. We randomly select 100 images for each dataset and choose the corresponding target text. For additional qualitative results, we apply our method to YouTube videos and the HierText dataset [17] presented in the supplementary material.

Baselines. We compared our proposed method with both non-diffusion model methods, including Pix2pix [12], SRNet [37], STEFANN [29], Text2Live [2], and De-Render [32], and diffusion model-based methods, including Imagic [15], text-to-image (Text2Img) LDM [27], Null-text [20] + prompt-to-prompt (p2p) [10], and SDEdit [18].

Evaluation Measurements. We evaluate the performance of the baselines using various measurements. Initially, we use the standard evaluation metrics for image manipulation, including PSNR, SSIM, and LPIPS [41] scores

Figure 4. Result of text scene manipulation on row 1: ICDAR2015 [13], row 2: IIIT5K [19], and row 3: SVT [24]. The input image is represented by *green box* and the edited version by *red box*.

Methods	PSNR (↑)	SSIM (↑)	LPIPS (↓)	OCR Acc. (% ↑) char	OCR Acc. (% ↑) word
Pix2pix [12]	28.39	0.47	0.52	40.83	9.50
SRNet [37]	29.21	0.49	0.53	88.70	71.16
De-Render* [32]	29.68	**0.65**	0.48	40.82	17.39
SDEdit [18]	29.35	0.45	0.44	4.91	-
LDM [27]	29.78	0.51	0.38	44.33	11.50
Imagic [15]	28.18	0.30	0.61	77.75	50.66
Imagic with SynText [15]	28.33	0.31	0.48	94.33	82.33
DBEST (Ours)	**30.09**	0.54	**0.29**	**94.58**	**84.83**

Table 1. The average of PSNR, SSIM, LPIPS, and OCR scores on SynText dataset. The notations '-' and '*' denote a method with no correct word and unfair comparison, respectively. The best score is denoted by **bold** text.

Methods	OCR Acc. (% ↑) COCO-Text char	OCR Acc. (% ↑) COCO-Text word	OCR Acc. (% ↑) ICDAR2013 char	OCR Acc. (% ↑) ICDAR2013 word
Pix2pix [12]	20.00	-	17.60	-
SRNet [37]	56.58	21.00	51.79	25.51
De-Render* [32]	24.63	2.00	18.54	-
Imagic [15]	20.00	1.00	14.70	-
SDEdit [18]	17.31	-	18.30	-
Null-Inv [20] + P2P [10]	18.78	-	17.37	-
DBEST (Ours)	**94.15**	**83.00**	**98.12**	**92.00**

Table 2. The average OCR accuracy on COCO-Text and IC-DAR2013 dataset. The notations '-' and '*' denote a method with no correct word and unfair comparison, respectively. The best score is denoted by **bold** text.

on the SynText dataset. We compare the models based on character recognition benchmarks. For this purpose, we utilize a pre-trained weight of scene text recognition [38] to determine if the generated sample is readable. We measure the recognition performance at both the character level and word level.

5.1. Evaluations

State-of-the-art comparison. We conduct comparisons among state-of-the-art (SOTA) methods and found that SR-Net is the best-performing method among them. In Table 1, we compare our method with Pix2pix, SRNet, De-Render, SDEdit, Imagic, and LDM as the SOTA methods on SynText dataset. In Table 2, we compare our method with SRNet, De-Render, Imagic, SDEdit, and Null-text + P2P as the SOTA methods on in-the-wild datasets. Our approach achieved SOTA performance on image quality as-

sessments. Furthermore, in Table 1, we achieve 94% and 84% for character and word recognition accuracy, respectively. Even when evaluating in-the-wild data (as shown in Table 2), our method maintained consistent performance, whereas SRNet's performance significantly decreased as well as Imagic's performance. In Fig. 3, it is demonstrated that SDEdit, Imagic, Null-text, Text2Img LDM, and Text2live are unable to modify the source image. The main reason is that language models, such as BERT or CLIP, do not possess adequate knowledge of scene text. Meanwhile, STEFFAN heavily relies on per-character map estimation, and SRNet struggles with reconstruction due to the diverse characteristics of in-the-wild datasets compared to SynText. Our method, on the other hand, can manipulate the scene while preserving the source style. However, when we applied De-Render to both evaluation tables, we observed that several images were undetected. This is be-
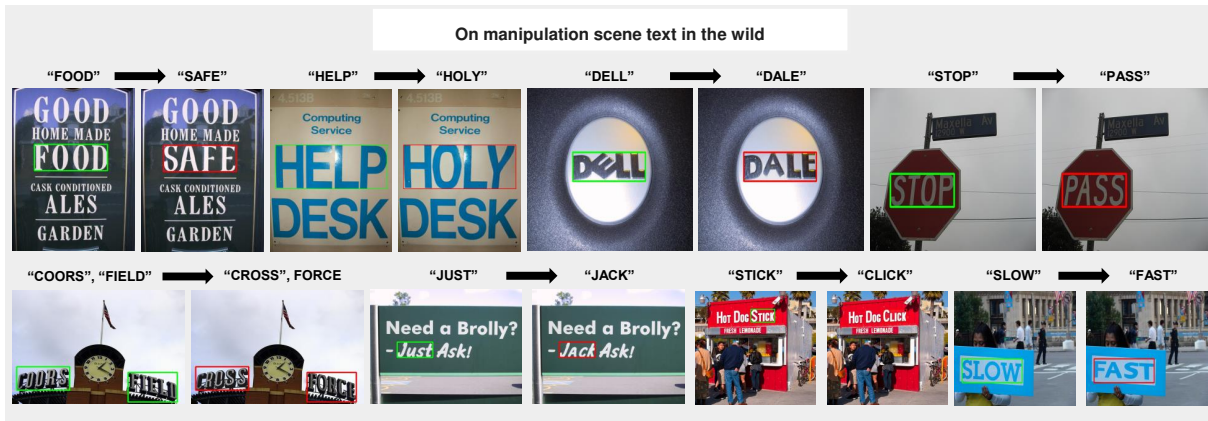
Figure 5. Given a single word *green box* from in-the-wild images and the desired text, our method successfully to edit the text with the desired text in the image *red box*.

SynText	Text	PSNR (↑)	SSIM (↑)	LPIPS (↓)	OCR Acc. (% ↑)	
					char	word
✗	✗	29.78	0.51	0.38	44.33	11.50
✗	✓	29.90	0.52	0.35	76.70	50.00
✓	✗	**30.09**	**0.54**	0.30	84.25	60.16
✓	✓	**30.09**	**0.54**	0.29	94.58	84.83

Table 3. Ablation studies on the SynText dataset. The best score is denoted by **bold** text. The column 'SynText' and 'Text' denote a utilization of SynText dataset and text recognition guidance, respectively.

cause this method is limited to digital documents.

Result on the in-the-wild dataset. Fig. 4 illustrates more examples of scene text manipulation on ICDAR2015 [13], IIIT5K [19], and SVT [24] datasets. ICDAR2015 and SVT datasets are particularly challenging due to the presence of factors that can affect captured scene text, including geometry distortion, noise effects, and motion blur. Even with those conditions, our method is still able to manipulate the scene text. In addition, our method's ability to manipulate text in a full image is demonstrated in Fig. 5. To accomplish this, we can utilize a scene text detection algorithm like EAST [43] or a provided text bounding box to identify each text instance within the image. Then, each identified text area is extracted and provided to our method along with the target text. Finally, the edited image is projected back onto the original image to generate the desired output.

5.2. Ablation studies

Below, we investigate how our method behaves and the impacts of each component in our work.

Why do we need pre-trained with SynText dataset? At its core, our problem is the inability of text-to-image models to handle the scene text domain, as depicted in the bottom part of Fig. 1. By and large, we observe that the LAION

Figure 6. Result from a single image with different target texts. The input image is represented by *green box*, the edited image without one-shot diffusion model is shown by *blue box* and the final edited version by *red box*.

dataset [31] comprises general scenes. Therefore, a straightforward approach of using LDM for text-to-scene text fails. In Table 3, we present significant improvement in terms of OCR accuracy, where the first row represents the accuracy of text-to-image LDM as a baseline.

Why do we need one-shot style adaptation? The one-shot style adaptation for the diffusion model is crucial for maintaining the source appearance and text geometry of the edited text. As shown in Fig. 6, our approach can successfully edit the text, but the edited result loses its source appearance without the one-shot method. In other words, without the one-shot diffusion model, our method may generate an image that does not resemble the original input. The one-shot style adaptation reduces the distribution space of the diffusion model, ensuring that the generated image maintains the source appearance of the input image. Furthermore, our method offers the advantage of allowing multiple text editing tasks to be performed with just a single one-shot style adaptation. This means that users can easily modify various pieces of text while maintaining the same
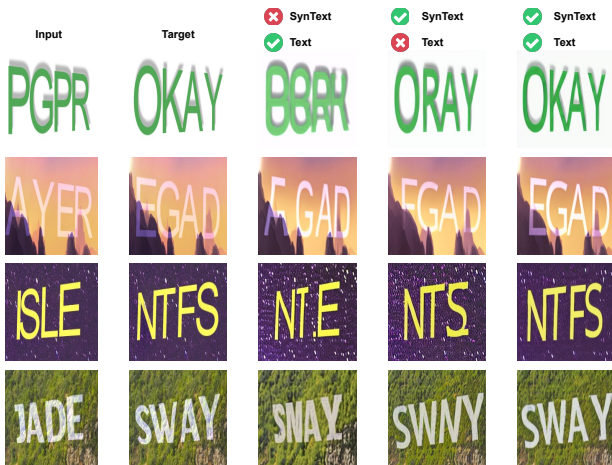
Figure 7. Ablation result of our method on Syntext dataset. The label 'SynText' and 'Text' denote a utilization of SynText dataset and text recognition guidance, respectively.

style without requiring additional style adaptation. The effectiveness of this can be observed in Fig. 6.

Why do we need text recognition guidance? According to Table. 3, when text recognition guidance is not used, the character accuracy and word accuracy are 84.25 % and 60.16 %, respectively. This is further confirmed by Fig. 7, where we can see that the manipulated scene text contains some character errors. However, with the addition of text recognition guidance, the diffusion model is guided to revise the character errors, resulting in a significant improvement in word accuracy, as shown in Table. 3. In essence, the text recognition guidance helps to improve the accuracy of the edited text scene by guiding the diffusion model to revise the character errors. By incorporating this guidance, our approach is able to generate more readable and accurate edited text scenes, improving the overall quality of the output.

What is the key difference with Imagic [15] ? We emphasize that Imagic requires interpolating between the source and target embeddings with an associated parameter difficult to be finetuned. While, a simple Imagic model trained on Syntext denoted as 'Imagic with SynText' failed to edit scene texts effectively (e.g., inconsistent styles) as depicted in Fig. 8 (col. 2) and low score on image quality performance in Table 1. In contrast, our proposed method can easily generate well-accepted results by only optimizing the target text embedding without an extra interpolation parameter.

Length of characters. We analyze the effect of character length on text manipulation performance by using an input image with four different character lengths. As shown in Fig. 9, the results indicate a significant decrease in performance when manipulating text with more than five charac-

Figure 8. 'MAIN', 'STOP' → 'NIPS', 'ICML'. The input image is represented by *green box*, the result by 'Imagic with SynText' is shown by *blue box* and ours by *red box*.
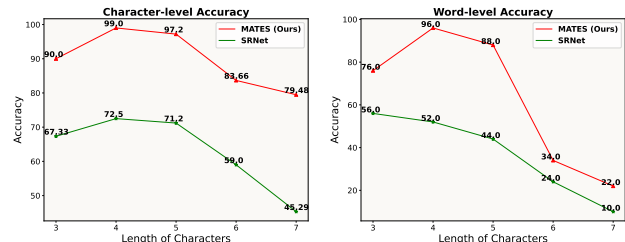

Figure 9. Analyzing the edited result with various length of characters on the in-the-wild dataset. The input image is constrained to 4 length of characters and the target text is varying from 3-7 characters. Though the target text has more characters, our method is still superior compared to SRNet in terms of accuracy in character and word levels.

ters. We attribute this drop in performance to the model's inability to adjust the font size based on the given text and the available space in the scene. Currently, our model lacks the ability to adapt the font size based on the text length, which limits its performance in manipulating longer texts. This limitation is evident in Fig. 9, where the model struggles to maintain accuracy when manipulating text with more than five characters. Therefore, improving the model's ability to adjust font size dynamically based on the text length and available space in the scene could enhance its performance in manipulating longer texts.

6. Conclusion

We propose a diffusion model for scene text manipulation with one-shot style adaptation and text recognition guidance. In our evaluation, we use synthesized and in-the-wild datasets, which have different characteristics. As our experimental evaluations indicate, our methods comfortably outperform the existing methods in both datasets. Furthermore, the ablation studies we conducted indicate that each process within our method has a distinct task and is crucial to its overall performance.

Our method currently has two limitations. First, the inference time is around 8 minutes per image, making it unsuitable for real-time applications. Second, there is a constraint on the character length that can be manipulated. As mentioned, adjusting the font size is another challenge we aim to address in future work.

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, pages 18187–18197, 2022. 2

[2] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *ECCV*, pages 707–723, 2022. 5

[3] Xuewei Bian, Chaoqun Wang, Weize Quan, Juntao Ye, Xiaopeng Zhang, and Dong-Ming Yan. Scene text removal via cascaded text stroke detection and erasing. *Comput. Vis. Media*, 8(2):273–287, 2022. 1

[4] Katherine Crowson. Clip guided diffusion hq 256x256. https://colab.research.google.com/drive/12a_Wrfi2_gwwAuN3VvMTwVMz9TfqctNj. 2

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019. 3

[6] Victor Fragoso, Steffen Gauglitz, Shane Zamora, Jim Kleban, and Matthew A. Turk. Translatar: A mobile augmented reality translator. In *WACV*, pages 497–502, 2011. 1

[7] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ArXiv*, 2022. 4

[8] Ian Goodfellow, Jean Pouget-Abadi, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengi. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014. 1

[9] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016. 3, 5

[10] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *ArXiv*, 2022. 3, 5, 6

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, pages 6840–6851, 2020. 2, 3

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 5967–5976, 2017. 5, 6

[13] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman K. Ghosh, Andrew D. Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, Faisal Shafait, Seiichi Uchida, and Ernest Valveny. ICDAR 2015 robust reading competition. In *ICDAR*, pages 1156–1160, 2015. 6, 7

[14] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernández Mota, Jon Almazán, and Lluís-Pere de las Heras. ICDAR 2013 robust reading competition. In *ICDAR*, pages 1484–1493, 2013. 5

[15] 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 *ArXiv*, 2022. 2, 3, 4, 5, 6, 8

[16] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *CVPR*, pages 10674–10685, 2022. 2

[17] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. Towards end-to-end unified scene text detection and layout analysis. In *CVPR*, pages 1039–1049, 2022. 5

[18] 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 *ICLR*, 2022. 2, 5, 6

[19] Anand Mishra, Karteek Alahari, and C. V. Jawahar. Top-down and bottom-up cues for scene text recognition. In *CVPR*, pages 2687–2694, 2012. 6, 7

[20] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *ArXiv*, 2022. 2, 3, 4, 5, 6

[21] 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. In *ICML*, pages 16784–16804, 2022. 2

[22] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, pages 2642–2651, 2017. 2

[23] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, pages 2065–2074, 2021. 2

[24] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *ICCV*, pages 569–576, 2013. 6, 7

[25] 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 *ICML*, pages 8748–8763, 2021. 3

[26] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831, 2021. 2

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685, 2022. 1, 2, 3, 5, 6

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 3

[29] Prasun Roy, Saumik Bhattacharya, Subhankar Ghosh, and Umapada Pal. STEFANN: scene text editor using font adaptive neural network. In *CVPR*, pages 13225–13234, 2020. 2, 5

[30] 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. In *ArXiv*, 2022. 4

[31] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: open dataset of clip-filtered 400 million image-text pairs. In *ArXiv*, 2021. 3, 7

[32] Wataru Shimoda, Daichi Haraguchi, Seiichi Uchida, and Kota Yamaguchi. De-rendering stylized texts. In *ICCV*, pages 1056–1065, 2021. 1, 2, 5, 6

[33] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265, 2015. 3

[34] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 2, 3

[35] Zhengmi Tang, Tomo Miyazaki, Yoshihiro Sugaya, and Shinichiro Omachi. Stroke-based scene text erasing using synthetic data for training. *IEEE Transactions on Image Processing*, 30:9306–9320, 2021. 1

[36] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge J. Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. In *ArXiv*, 2016. 5

[37] Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. Editing text in the wild. In *ACM MM*, pages 1500–1508, 2019. 1, 2, 5, 6

[38] Ruijie Yan, Liangrui Peng, Shanyu Xiao, and Gang Yao. Primitive representation learning for scene text recognition. In *CVPR*, pages 284–293, 2021. 6

[39] Qiangpeng Yang, Jun Huang, and Wei Lin. Swaptext: Image based texts transfer in scenes. In *CVPR*, pages 14688–14697, 2020. 1, 2

[40] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. Controllable artistic text style transfer via shape-matching GAN. In *ICCV*, pages 4441–4450, 2019. 1

[41] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 5

[42] Shuaitao Zhang, Yuliang Liu, Lianwen Jin, Yaoxiong Huang, and Songxuan Lai. Ensnet: Ensconce text in the wild. In *AAAI*, pages 801–808, 2019. 1

[43] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: an efficient and accurate scene text detector. In *CVPR*, pages 2642–2651, 2017. 7