Supplementary Material for the Paper: Scene Text Image Super-resolution based on Text-conditional Diffusion Models

Chihiro Noguchi Shun Fukuda Masao Yamanaka
Toyota Motor Corporation, Japan
{chihiro_noguchi_aa, shun_fukuda, masao_yamanaka}@mail.toyota.co.jp

A. Denoising Diffusion Probabilistic Models

A diffusion model consists of two processes, forward and reverse. In the forward process, Gaussian noise is gradually added to the input image and, eventually, becomes pure Gaussian noise. Conversely, in the reverse process, starting from pure Gaussian noise, noise is removed sequentially to recreate the original images. Following this definition, DMs can be classified into at least three categories [1, 9]: denoising diffusion probabilistic models and noise-conditioned score networks, and stochastic differential equations. The method proposed in this study is based on DDPMs. In the following, we derive the loss function of DDPMs in Eq. 1. The following derivation is based on [3, 8].

Given a data distribution $x_0 \sim q(x_0)$, the forward process is defined as the Markov process, where a series of latent variables $x_1, \ldots, x_T$ is produced by progressively adding Gaussian noise:

$$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t \boldsymbol{I}), \tag{1}$$

to the sample. Hence,

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}). \tag{2}$$

Here, $\beta_t \in (0,1)$, $\forall t \in \{1,\ldots,T\}$ indicates the variance at time t. When T is sufficiently large, x_T is equivalent to a pure Gaussian noise. Here, there is a helpful property in that x_t at any timestep t can be sampled directly from x_0, following a single Gaussian

$$q(x_t|x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\boldsymbol{I}), \tag{3}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_t$. Therefore, using $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$, $x_t = \sqrt{\bar{\alpha}_t}x_0 + (1-\bar{\alpha}_t)\epsilon$.

In the reverse process, starting from a Gaussian noise $x_T \sim \mathcal{N}(0, \boldsymbol{I})$, we can reverse the forward process by sampling from the posterior $q(x_{t-1}|x_t)$. However, directly estimating $q(x_{t-1}|x_t)$ is difficult because it requires the data distribution $q(x_0)$. However, it is known that $q(x_{t-1}|x_t)$ can be approximated as a Gaussian distribution when β_t is sufficiently small [8]. Therefore, $q(x_{t-1}|x_t)$ can reasonably fit to the true posterior by being parameterized as

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t,t), \Sigma_\theta(x_t,t)). \tag{4}$$

Consequently, the reverse process is parameterized as follows:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{5}$$

The loss function to be optimized is provided by the variational lower bound of the negative log likelihood $\mathcal{L} = \mathbb{E}[-\log p_\theta(x_0)]$. This optimization can be performed efficiently for each timestep because $\mathcal{L}$ can be broken down by using chain rules and Eq. 3. The final loss function is obtained as follows:

$$\mathcal{L}_u = \mathbb{E}_{t,x_0,\epsilon}[\|\epsilon - \epsilon_\theta(x_t,t)\|^2]. \tag{6}$$

See [3,8] for the detailed derivation of Eq. 6. Consequently, a deep neural network ϵ_θ is trained to estimate the Gaussian noise ϵ contained in x_t.

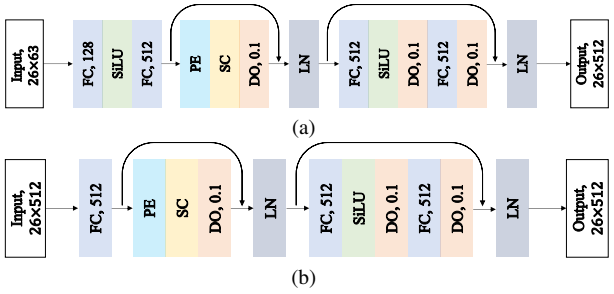


Figure 1. Architecture details of the text encoders in (a) TCDM and (b) TCDM*, respectively. FC, SA, PE, DO, and LN stand for a fully-connected layer, a self-attention layer, positional encoding, dropout, and layer normalization, respectively.

Hyperparameters	Traning on TextZoom (Sec. 4.1)	Training on Extended dataset (Sec. 4.2)	Fine-tuning (Sec. 4.3)
Diffusion Steps	1000	1000	1000
Noies Schedule	linear	linear	linear
Channels	128	128	128
Channel Multiplier	1,2,3,4	1,2,3,4	1,2,3,4
Number of Heads	1	1	1
Batch Size	128	128	128
Learning Rate	3e-4	3e-4	3e-5
Iterations	100K	300K	30K
Embedding Dimension	512	512	512
Attention Resolution	32,16,8	32,16,8	32,16,8

Table 1. Hyperparameters of TCDM and TCDM* for each experiment in the main text.

k_b	0	3	6	9
TCDM	55.0%	55.1%	**55.7%**	55.6%
TCDM*	55.0%	57.5%	67.0%	**68.1%**

Table 2. Average recognition accuracy of TCDM and TCDM* for $k_b \in \{0, 3, 6, 9\}$.

k_b	k_m	Acc. (%)	Time (s)
1	1	58.0	9.30
3	0	57.5	5.39
6	0	67.0	6.07
9	0	**68.1**	6.82

Table 3. Average recognition accuracy and inference time of TCDM* for $(k_b, k_m) \in \{(1, 1), (3, 0), (6, 0), (9, 0)\}$

B. Design of Text-conditional DM Architectures

To describe the architectural differences simply, we introduce two parameters: k_b and k_m. k_b denotes the number of iterations of the cross-attention and self-attention modules at the bottom of the UNet and k_m denotes the number of iterations in each resolution layer, except for the bottom layer. Using k_b and k_m, the architectures proposed in prior studies [6, 7] can be expressed as $k_b = 1$ and $k_m = 1$. To determine k_b of our text-conditional DM, we used grid search, and Table 2 shows recognition accuracy of our text-conditional DMs for $k_b \in \{0, 3, 6, 9\}$. TCDM denotes the text-conditional DM and TCDM* indicates TCDM trained using ground-truth text input. TCDM achieved the best performance when $k_b = 6$ and TCDM* when $k_b = 9$. In addition, we set k_m to 0 for TCDM because $k_m > 0$ significantly increases the computational cost. Table 3 shows the inference time per image. We can see that setting k_m to 1 not only results in a substantial increase in the inference time but also leads to a drop in recognition accuracy.

The text encoder of our text-conditional DM has a simple architecture, which is based on self-attention and fully-connected layers. This architecture is based on the text prior generator proposed in [4]. Figures 1a and 1b show the architectural details of the text encoders for our text-conditional DMs. When ground-truth texts are used, the input of the text encoder has a shape 26×63, where the maximum word length $l = 26$, and the alphabet size $|\mathcal{A}| = 63$. The input text was encoded into text features for each character candidate, and the output tensor had a shape 26×512. When the text prior generator is used, the text encoder has the same architecture, except that the input is given as intermediate features of the text prior generator and has a shape 26×512.

The implementation of our text-conditional DMs was based on the code released by the authors in [2, 5]. In Tab. 1, we present hyperparameters of our text-conditional DMs for each experiment in the main text.

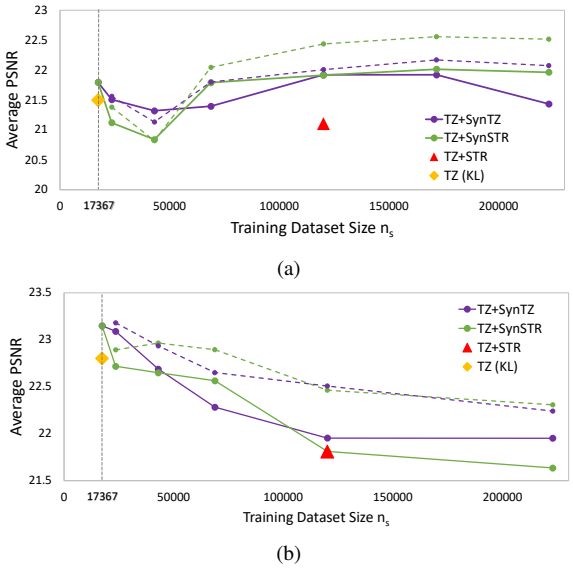
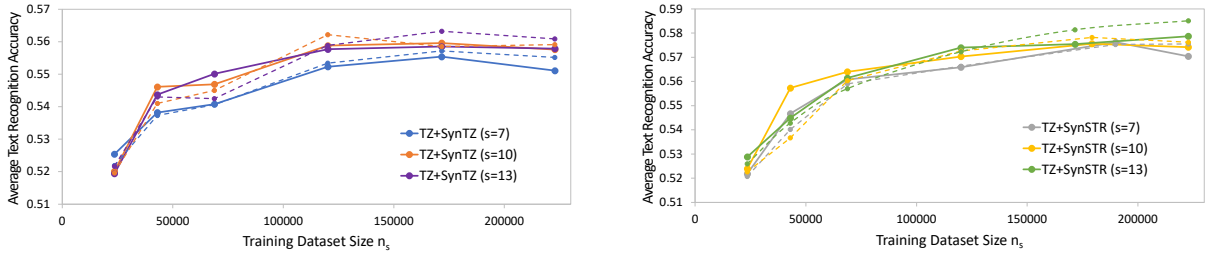
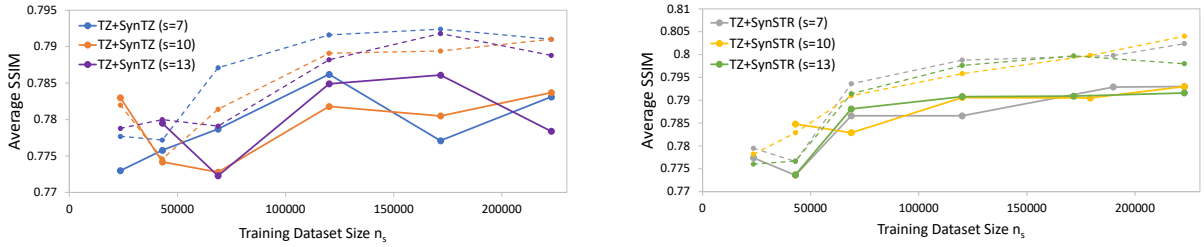


Figure 2. Evaluation results of (a) TATT and (b) our text-conditional DMs trained on the augmented dataset in terms of PSNR.



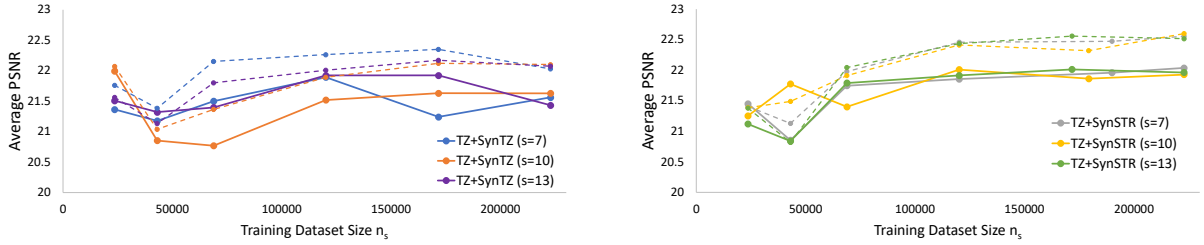
(a) (b)

Figure 3. Average recognition accuracy of TATT trained on the augmented datasets with various sizes. The augmented datasets were created with a different maximum word length $s \in \{7, 10, 13\}$. (a) and (b) show the average recognition accuracy of TATT trained on TZ+SynTZ and TZ+SynSTR, respectively. The recognitiona accuracy was evaluated by CRNN.



(a) (b)

Figure 4. Average SSIM of TATT trained on the augmented datasets with various sizes. The augmented datasets were created with a different maximum word length $s \in \{7, 10, 13\}$. (a) and (b) show the average SSIM of TATT trained on TZ+SynTZ and TZ+SynSTR, respectively.



(a) (b)

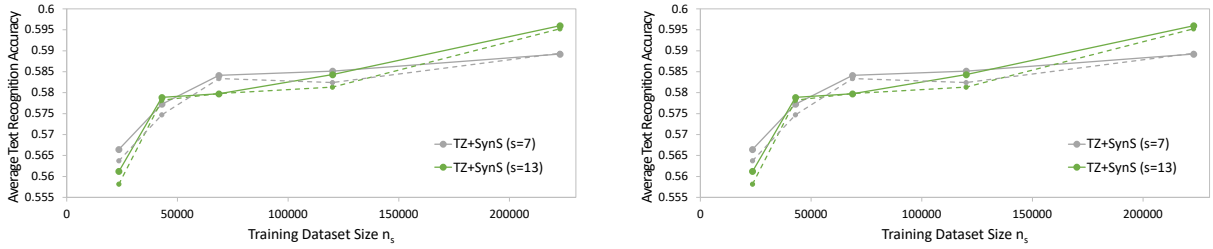
Figure 5. Average PSNR of TATT trained on the augmented datasets with various sizes. The augmented datasets were created with a different maximum word length $s \in \{7, 10, 13\}$. (a) and (b) show the average PSNR of TATT trained on TZ+SynTZ and TZ+SynSTR, respectively.

C. Evaluation of Extended Dataset using PSNR

Figures 2a and 2b show the average PSNR of TATT and the text-conditional DMs, respectively, as solid lines. Similar to the results of SSIM, PSNR does not depend on n_s. Figure 2 also shows the results of the fine-tuning with dotted lines. We can see that PSNR can be significantly improved by fine-tuning.

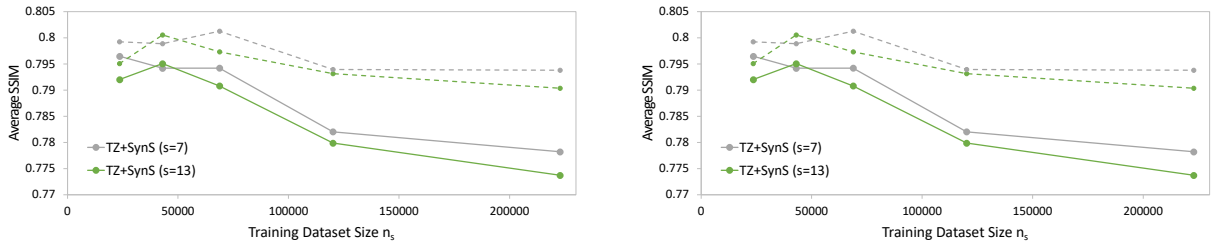
D. Evaluation of Extended Dataset for Varying Maximum Word Lengths

When synthesizing text images with Synthesizer, we uniformly sampled the word length from the minimum word length of 2 to the maximum word length $s = 13$ and chose a word from an English word dictionary with the same word length. In this section, we compare several maximum word lengths $s \in \{7, 10, 13\}$ in terms of the recognition accuracy and SSIM/PSNR. Figures 3a, 4a, and 5a show the results of recognition accuracy and SSIM/PSNR of TATT trained on TZ+SynTZ, respectively. Figures 3b, 4b, and 5b show the



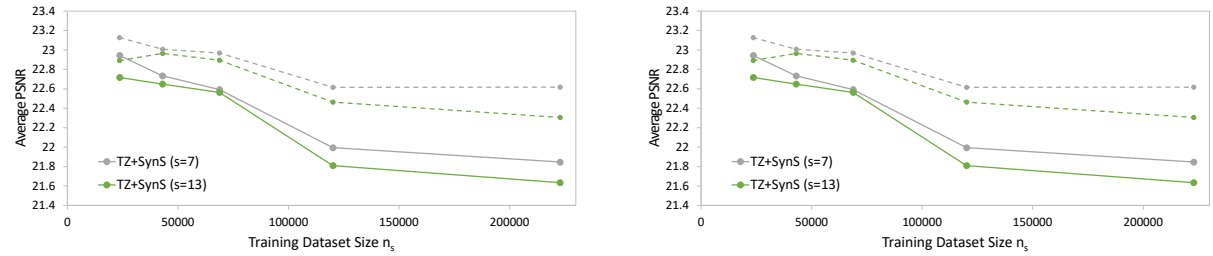
(a) (b)

Figure 6. Average recognition accuracy of our text-conditional DMs trained on the augmented datasets with various sizes. The augmented datasets were created with a different maximum word length $s \in \{7, 13\}$. (a) and (b) show the average recognition accuracy of the DMs trained on TZ+SynTZ and TZ+SynSTR, respectively. The recognitiona accuracy was evaluated by CRNN.



(a) (b)

Figure 7. Average SSIM of our text-conditional DMs trained on the augmented datasets with various sizes. The augmented datasets were created with a different maximum word length $s \in \{7, 13\}$. (a) and (b) show the average SSIM of the DMs trained on TZ+SynTZ and TZ+SynSTR, respectively.



(a) (b)

Figure 8. Average PSNR of our text-conditional DMs trained on the augmented datasets with various sizes. The augmented datasets were created with a different maximum word length $s \in \{7, 13\}$. (a) and (b) show the average PSNR of the DMs trained on TZ+SynTZ and TZ+SynSTR, respectively.

same trained on TZ+SynSTR. We can see that the recognition accuracy tends to improve as s increases. However, SSIM and PSNR did not improve depending on s.

In addition, Figs 6a, 7a, and 8a show the results of recognition accuracy and SSIM/PNSR of the text-conditional DMs trained on TZ+SynTZ, respectively. Figures 6b, 7b, and 8b show the same trained on TZ+SynSTR. There was no significant difference between the results for $s = 7$ and $s = 13$. Therefore, following the results of TATT, we set s to 13 in the experiments in the main text.

E. Effectiveness of Ground-truth Text Prior in Degrader

As mentioned in the main text, ground-truth text prior was used to train Degrader. However, determining whether the use of the ground-truth texts is effective for training Degrader is non-trivial because the image degradation process can be realized without the textual information. Therefore, we conducted experiments to compare LR text images generated by the Degrader with and without the ground-truth text prior. For the comparison, we trained TATT and the text-conditional DMs with HR-LR paired images

Method	SSIM ($\times 10^{-2}$)	PSNR
TATT	88.87	23.60
TATT*	**89.93**	**24.63**
TCDM	90.12	25.21
TCDM*	**90.37**	**25.40**

Table 4. Comparison among different image degradation methods.

of TextZoom, that is, the HR images were used as input, and the LR images were used as training targets. To evaluate the performance, we measured SSIM/PSNR between the generated LR images and LR images of TextZoom. Table 4 shows the results of comparing TATT with TATT* and TCDM with TCDM*. Here, TATT* and TCDM* denote TATT and the text-conditional DM trained using the ground-truth text prior, respectively. The SSIM/PSNR of TATT* are higher than those of TATT. In addition, the SSIM/PSNR of TCDM* are higher than those of TCDM, although by a small margin. Based on these results, we believe that the ground-truth texts may be effective for text-image degradation. If the variety of degradation process patterns is limited to those that appear in a specific environment, the degradation patterns of the text in the text image may be identified from the corresponding ground-truth texts to some extent. Therefore, because our objective in this study is to imitate the TextZoom's degradation process for the dataset augmentation, we considered that the ground-truth text input would be effective for the Degrader.

F. Higher-Resolution Text Images

The HR text images of TextZoom contain many blurred images, which can deteriorate the performance of STISR methods. Therefore, we can consider experiments applying Super-resolver to the HR text images of TextZoom to improve the quality of the HR images further. Here, we refer to text images to which Super-resolver is applied as higher-resolution (HerR) text images. First, we evaluated the HerR text images in terms of the recognition accuracy and SSIM/PSNR. As shown in Tab. 5, the recognition accuracy of the HerR images is further improved compared to that of the HR images. SSIM/PSNR were measured against the corresponding HR images.

In addition, we conducted experiments in which the HerR text images were used as the target images for the TATT training, instead of the HR images . As presented in Tab. 6, the recognition accuracy obtained with the HerR images is higher than that with the HR images. On the other hand, SSIM/PSNR decreases when the HerR images were used. The deterioration of SSIM/PSNR stems from the deviation of SSIM/PSNR between the HR and HerR images, as shown in Tab. 5. In practical terms, clearer images are better suited for text recognition (in fact, recognition accuracy has been improved); thus we consider that the deterioration of SSIM/PSNR due to the clearer text images is not a negative effect.

Metrics	LR	SR	HR	HerR
Acc. (%)	26.8	68.1	72.4	79.3
SSIM ($\times 10^{-2}$)	69.61	80.25	-	86.22
PSNR	20.35	22.86	-	23.66

Table 5. Comparison between LR, SR, HR, and HerR text images of TextZoom in terms of the average recognition accuracy and SSIM/PSNR. SR and HerR images were generated by applying Super-resolver to LR and HR images, respectively.

Metrics	HR	HerR
Acc. (%)	52.6	53.81
SSIM ($\times 10^{-2}$)	79.30	74.81
PSNR	21.52	19.64

Table 6. Comparison of the performance of TATT trained with HR and HerR text images in terms of average recognition accuracy and SSIM/PSNR. The recognition accuracy was evaluated by CRNN.

References

[1] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *arXiv preprint arXiv:2209.04747*, 2022. 1

[2] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 2021. 2

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1

[4] Jianqi Ma, Shi Guo, and Lei Zhang. Text prior guided scene text image super-resolution. *arXiv preprint arXiv:2106.15368*, 2021. 2

[5] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the International Conference on Machine Learning*, volume 139, pages 8162–8171, 2021. 2

[6] Robin Rombach, Andreas 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, June 2022. 2

[7] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 2

[8] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the International Conference on Machine Learning*, volume 37, pages 2256–2265, 2015. 1

[9] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022. 1