Diffusion-based Blind Text Image Super-Resolution Supplementary Materials

Yuzhe Zhang[1] Jiawei Zhang[2] Hao Li[2] Zhouxia Wang[3] Luwei Hou[2]
Dongqing Zou[2,4] Liheng Bian[1,*]
[1] Beijing Institute of Technology, [2] SenseTime Research, [3] The University of Hong Kong, [4] PBVR

In this supplementary file, we provide:

1. Detailed information on DiffTSR, including the training details and setting.
2. Architecture details of DiffTSR.
3. Visual results of the DiffTSR diffusion process.
4. Comparison with StableSR.
5. More visual comparison.

1. Detailed Information on DiffTSR

1.1. Training Details on IDM

The proposed IDM, which is based on Stable Diffusion [9] (SD), aims to model the distribution of text images $\mathbf{X}$ with given low-quality text images $\mathbf{X}_{LR}$ as well as text prior. The IDM only references the concept of the Stable Diffusion model that realizes the diffusion process in the latent space, and we do not use any pre-trained SD. Therefore, similar to Stable Diffusion, the training process of IDM is in the latent space. A pre-trained VAE is used to map the images into latent space by an encoder $\mathcal{E}$ and reconstruct the images by a decoder $\mathcal{D}$. Based on Stable Diffusion, Gaussian noises $\boldsymbol{\epsilon}$ with different noise levels are added to the latent features $\mathcal{E}(\mathbf{X})$ and a noise prediction network $\mathcal{U}$ is used to predict the noise $\boldsymbol{\epsilon}_{pred,t}$ during the training process of IDM. $\mathcal{U}$ is conditioned on the encoded LR features $\mathcal{E}(\mathbf{X}_{LR})$ and text prior $\mathcal{F}(\mathbf{c})$, where $\mathcal{F}$ is a Transformer Encoder and $\mathbf{c}$ is the ground truth text sequence, to guide the diffusion process to generate text images with high text fidelity and style realness. The training loss $L_{IDM}^{denoise}$ for denoising can be represented as minimizing the difference between $\boldsymbol{\epsilon}$ and $\boldsymbol{\epsilon}_{pred,t}$ for all the time steps. In addition, we use the same pre-trained text recognition model $\mathcal{P}$ [1] to recognize text in the VAE decoder $\mathcal{D}$ reconstructed image from the denoised latent features and use text recognition loss $L_{IDM}^{recognize}$ (see eq. [4] $\mathcal{L}_{CON}$ of [1] for details) to further guide the generated text image with the correct text structures in every time step. The final loss when training IDM can be represented as $L_{IDM} = L_{IDM}^{denoise} + \lambda_{recognize} L_{IDM}^{recognize}$. During the training stage of IDM, $\mathcal{U}$ and $\mathcal{F}$ are jointly optimized and the hyper-parameter $\lambda_{recognize}$ sets as 2×10^{-2}.

Note that there exist two pre-trained networks which as the VAE (including $\mathcal{E}$ and $\mathcal{D}$) and text recognition model $\mathcal{P}$. VAE is trained in an adversarial manner with a low-weighted Kullback-Leibler-term based on eq. (25) in the supplementary material of Stable Diffusion [9]. As to text recognition model $\mathcal{P}$, it is based on [1] and uses the same text recognition loss $L_{IDM}^{recognize}$ (see the content loss of [1] for details) to train it and make it to recognize text characters from degraded images.

1.2. Training Details on TDM

To model the distribution of text sequence, TDM also follows the Markov chain of the diffusion process that slowly adds random noises to the text sequence in the forward process and then learns the reverse process to reconstruct the text sequence from the noisy data. Unlike IDM which is a continuous diffusion model and the added noises satisfy Gaussian distribution, TDM is a discrete one. Similar to [4], for text sequence $\mathbf{c}$, TDM assumes the transition distribution follows a Categorical distribution in the forward process. Text sequence $\mathbf{c} = [c_1, c_2, \ldots, c_L]$ and $L = 24$ is the maximum length of text. All the characters in the text sequence belong to an alphabet with $K = 6736$ characters including both Chinese and English characters as well as numbers and special characters from CRNN [10]. TDM utilizes a Transformer Decoder $\mathcal{T}$ to estimate $\mathbf{c}_{pred}$ from $\mathbf{c}_t$ and we use the same strategy to train $\mathcal{T}$ as to [4].

1.3. Training Details on MoM

After pre-training IDM and TDM following the above steps, IDM is able to restore the high fidelity text image conditioned on LR image and text sequence, and TDM is able to generate the high precision text prior based on LR image. In order to simultaneously benefit from the powerful data distribution modeling ability of IDM and TDM, we propose a Mixture of Multi-modality (MoM) module to make these two models cooperate with each other in all diffusion steps.

MoM is a two-stream framework network that consists of a UNet-based encoder $\mathcal{I}^{MoM}$ and a Transformer-based encoder $\mathcal{F}^{MoM}$, and the detailed architecture of MoM is illustrated in Section 2. Therefore, $\mathcal{I}^{MoM}$ fuses the latent feature $\mathbf{Z}_t$ and the encoded feature $\mathbf{Z}_{LR}$ conditioned on text sequence $\mathbf{c}_t$ at timestep t, and $\mathcal{F}^{MoM}$ encodes the text sequence $\mathbf{c}_t$, as follows:

$$[\mathbf{I}cond_t, \mathbf{C}cond_t] = MoM_\phi([\mathbf{Z}_{LR}, \mathbf{Z}_t], \mathbf{c}_t, t) = [\mathcal{I}_\phi^{MoM}([\mathbf{Z}_t, \mathbf{Z}_{LR}], \mathbf{c}_t, t), \mathcal{F}_\phi^{MoM}(\mathbf{c}_t, t)]. \tag{1}$$

When training MoM, we freeze the weights of IDM and TDM, and only train MoM, as shown in Algorithm 1. The weights of MoM are initialized from the modules $\mathcal{F}$ and $\mathcal{I}$ of the pre-trained IDM and TDM. We set the λ in Algorithm 1 as 1.

1.4. Training Settings

All models are trained on the CTR-TSR-Train dataset. LR and HR images are resized into $128 \times 512 \times 3$ and the shape of latent feature $\mathbf{Z}$ is $32 \times 128 \times 3$. The training details are as follows:

1) Before training the UNet $\mathcal{U}$ of IDM, we first pre-train VAE ($\mathcal{E}$ and $\mathcal{D}$), text recognition model $\mathcal{P}$ and Transformer Encoder $\mathcal{F}$. We train VAE for 1×10^5 iterations with a batch size of 16, with a learning rate of 1×10^{-5}. We train the recognition model on HR and LR images of CTR-TSR-Train dataset for 1×10^4 iterations with a batch size of 64, with a learning rate of 2×10^{-1}. Then we freeze VAE and text recognition model, and train UNet $\mathcal{U}$ as well as Transformer Encoder $\mathcal{F}$ for 4×10^5 iterations with a batch size of 192, with a learning rate of 1×10^{-4}.
2) We train Transformer Decoder $\mathcal{T}$ of TDM for 4×10^5 iterations with a batch size of 256, with a learning rate of 1×10^{-4}.
3) Subsequently, we freeze IDM and TDM, and train MoM for 1×10^5 iterations with a batch size of 64, with a learning rate of 1×10^{-5}.

Besides, we use the AdamW [6] optimizer with the settings of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $weight\_decay = 0.01$. All experiments are conducted on 8 NVIDIA 40G-A100 GPUs. For training, we set diffusion timestep T to 1000. For inference, we adopt DDIM sampling [11] with eta 1.0 and timestep 200.

Algorithm 1 DiffTSR Training

repeat
  $\mathbf{X}_0, \mathbf{X}_{LR}, \mathbf{c}_0 \sim q(\mathbf{X}_{HR}, \mathbf{X}_{LR}, \mathbf{c}_0)$
  $\mathbf{Z}_0 = \mathcal{E}(\mathbf{X}_0)$
  $\mathbf{Z}_{LR} = \mathcal{E}(\mathbf{X}_{LR})$
  $t \sim \text{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$
  $\mathbf{Z}_t = \sqrt{\bar{\alpha}_t^{IDM}}\mathbf{Z}_0 + \sqrt{1 - \bar{\alpha}_t^{IDM}}\boldsymbol{\epsilon}$
  $\mathbf{c}_t \sim \mathcal{C}(\mathbf{c}_t \mid \bar{\alpha}_t^{TDM}\mathbf{c}_0 + \frac{1-\bar{\alpha}_t^{TDM}}{K})$
  $[\mathbf{I}cond_t, \mathbf{C}cond_t] = MoM_\phi([\mathbf{Z}_{LR}, \mathbf{Z}_t], \mathbf{c}_t, t)$
  $\boldsymbol{\epsilon}_{pred,t} = \mathcal{U}_\theta([\mathbf{Z}_t, \mathbf{Z}_{LR}], \mathbf{C}cond_t, t)$
  $\mathbf{c}_{pred,t} = \mathcal{T}_\eta(\mathbf{c}_t, \mathbf{I}cond_t, t)$
  $L_{IDM} = L_{IDM}^{denoise} + \lambda_{recognize} L_{IDM}^{recognize}$
  $\tilde{\boldsymbol{\pi}} = \left[\alpha_t^{TDM}\mathbf{c}_t + \frac{1-\alpha_t^{TDM}}{K}\right] \odot \left[\bar{\alpha}_{t-1}^{TDM}\mathbf{c}_{pred,t} + \frac{1-\bar{\alpha}_{t-1}^{TDM}}{K}\right]$
  $\boldsymbol{\pi}_{\text{post}}\ (\mathbf{c}_t, \mathbf{c}_{pred,t}) = \frac{\tilde{\boldsymbol{\pi}}}{\sum_{k=1}^{K} \tilde{\pi}_k}$
  $L_{TDM} = \text{KL}\left(\mathcal{C}\left(\boldsymbol{\pi}_{\text{post}}\ (\boldsymbol{c}_t, \boldsymbol{c}_0)\right) \,||\, \mathcal{C}\left(\boldsymbol{\pi}_{\text{post}}\ (\boldsymbol{c}_t, \boldsymbol{c}_{pred,t})\right)\right)$
  Take gradient descent on $\nabla_\phi(L_{IDM} + \lambda L_{TDM})$
until converged
// $\mathcal{C}$ denotes the categorical distribution with probability parameters after $|$.
// $1 - \alpha_{t-1}^{IDM}$ and $1 - \alpha_{t-1}^{TDM}$ are the noise schedule for IDM and TDM.
// $\bar{\alpha}_t^{IDM} = \prod_{i=1}^{t} \alpha_i^{IDM}$ and $\bar{\alpha}_t^{TDM} = \prod_{i=1}^{t} \alpha_i^{TDM}$

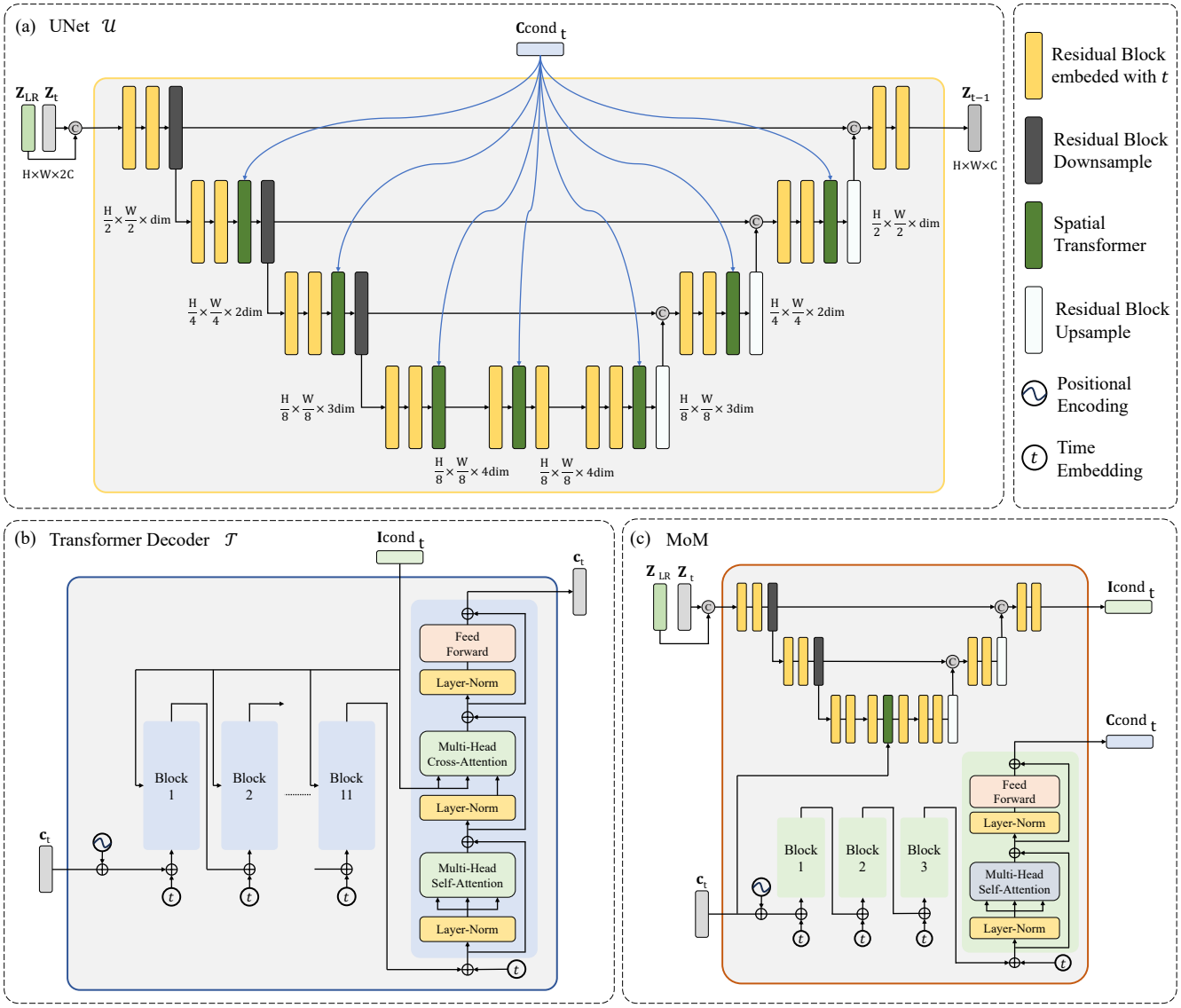

Figure 1. Architecture Details of DiffTSR

2. Architecture Details of DiffTSR

DiffTSR mainly consists of three parts: IDM, TDM, and MoM. IDM performs the diffusion process in the latent space through the VAE encoder $\mathcal{E}$ and decoder $\mathcal{D}$ with the downsampling factor of 4. The structures of $\mathcal{E}$ and $\mathcal{D}$ are the same as the VAE (downsampling factor $f = 4$) with KL regularization in Stable Diffusion. The UNet $\mathcal{U}$ in IDM conducts the noise prediction conditioned on encoded feature $\mathbf{Z}_{LR}$ during the reverse process, and its architecture is illustrated in Figure 1 (a). The UNet $\mathcal{U}$ has four encode stages, one middle stage, and four decode stages. Each stage consists of two basic residual blocks, where the time embedding t is added to each block. The input of $\mathcal{U}$ is the concatenation of $\mathbf{Z}_{LR}$ and $\mathbf{Z}_t$, whose shape is $H \times W \times 2C = 32 \times 128 \times 6$. The output channel of the first encoder stage of $\mathcal{U}$ sets to 320. Through four encode stages with channel-wise multiplication and spatial downsampling by residual blocks, the number of channels in the encoded feature is multiplied by $1, 2, 3, 4$ and spatial size is divided by $2, 4, 8, 8$. The encoded features are concatenated to the input of the corresponding decode stages through skip connections. In order to introduce the conditioning mechanism, we insert spatial transformer block with a depth of 1 in each of the middle seven stages. The multi-head cross-attention layer in the spatial

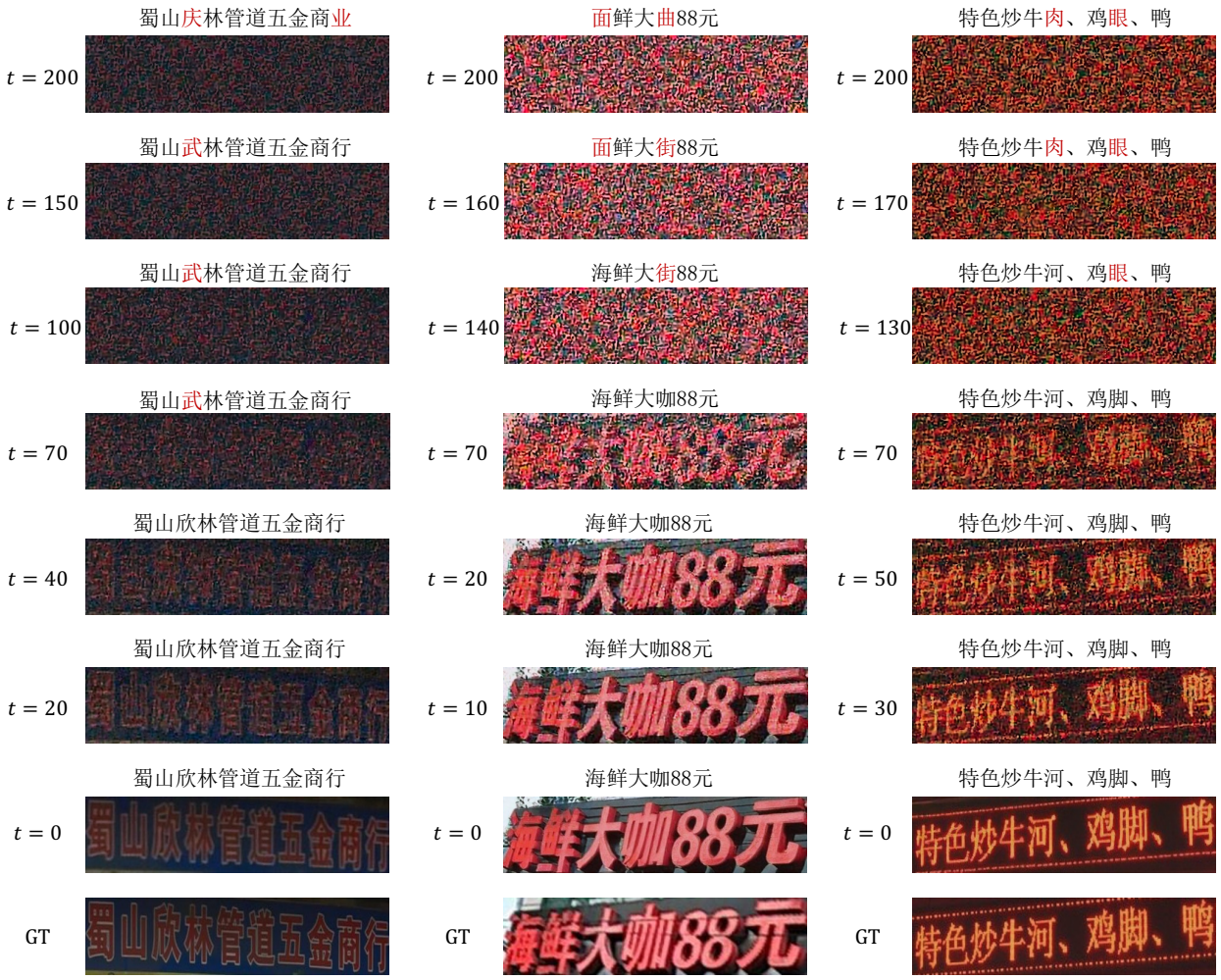

Figure 2. The text sequences above each super-resolution result at different time steps are the recognized text characters used for blind image super-resolution and the characters in red are the mistakenly estimated ones.

transformer has 8 heads, and the context embedding dimension is 160. Through four decode stages with spatial upsampling, UNet $\mathcal{U}$ successfully output $\mathbf{Z}_{t-1}$ with a shape of $H \times W \times C = 32 \times 128 \times 3$.

TDM aims to model the text sequence $\mathbf{c}$ distribution through multinomial diffusion, and TDM uses the Transformer Decoder $\mathcal{T}$ to conduct denoising during the reverse process. As shown in Figure 1 (b), the Transformer Decoder $\mathcal{T}$ has twelve basic blocks whose dimension is 768. The time embedding t is added to the input of each block. The multi-head cross-attention is calculated between condition $\mathbf{I}cond_t$ and the intermediate feature to implement the conditioning mechanism. The number of heads in the multi-head attention layers is set to 16. We set the dropout rate to zero.

We use the MoM to enable the cooperation between IDM and TDM. Specifically, MoM takes the previous step output of IDM and TDM as input and fuses them with encoded feature $\mathbf{Z}_{LR}$, then MoM outputs the different modality conditions for the current step to IDM and TDM. The MoM is composed of two modules, a UNet-based encoder $\mathcal{I}^{MoM}$ and a Transformer-based encoder $\mathcal{F}^{MoM}$. The MoM architecture is shown in Figure 1 (c), in which the two paths simultaneously encode the features of two modalities. The UNet-based module $\mathcal{I}^{MoM}$ in MoM aims to fuse and encode the intermediate latent feature $\mathbf{Z}_t$ and $\mathbf{Z}_{LR}$ at timestep t. It has three encode stages, one middle stage, and three decode stages, and each stage has two basic residual blocks, which are similar to the UNet in IDM. The model channel is set to 32. We also encode the text sequence $\mathbf{c}_t$ into the intermediate layer of $\mathcal{I}^{MoM}$ to further preserve text information in the fused feature. The Transformer-based encoder $\mathcal{F}^{MoM}$ in MoM aims to encode the text sequence $\mathbf{c}_t$ to the condition feature space of IDM. It consists of four Transformer encoder blocks with 160 channels, and the multi-head self-attention layer has 8 heads. The positional embedding and time embedding are also added.

3. Visual Results of DiffTSR Diffusion Process

We provide more visual results of the DiffTSR diffusion process to illustrate the effectiveness of TDM and MoM, as shown in Figure 2. As demonstrated in our manuscript, IDM cannot restore text images with high text fidelity under inaccurate text prior. However, our proposed TDM and IDM can benefit from each other through MoM in DiffTSR and gradually recognize more accurate text sequence and restore higher-quality text image through the reverse diffusion process.

4. Comparison with StableSR

We further compare our method with the recent prevalent diffusion-based image super-resolution method StableSR [12]. For a fair comparison, we also carefully finetune StableSR on CTR-TSR-Train dataset. We conduct the quantitative comparison for CTR-TSR-Test dataset and RealCE dataset on $\times 2$ and $\times 4$ tasks. As shown in Table 1, benefiting from the powerful text prior and text image modeling ability, our method still outperforms the StableSR in all metrics. Although StableSR performs well in natural images, it is unable to handle text images with regular strokes due to its lack of text prior and efficient text image modeling ability. Besides, the visual results of StableSR are shown in Figure 3 and Figure 4. The qualitative comparison also demonstrates that StablSR can cause distortion in restoring text images with complex strokes or severe degradation.

Method	Dataset	x2 PSNR ↑	x2 LPIPS ↓	x2 FID ↓	x2 ACC ↑	x2 NED ↑	x4 PSNR ↑	x4 LPIPS ↓	x4 FID ↓	x4 ACC ↑	x4 NED ↑
StableSR	CTR-TSR-Test	22.89	0.234	12.38	0.7952	0.8068	20.37	0.321	17.19	0.6272	0.6383
ours	CTR-TSR-Test	**25.08**	**0.156**	**5.906**	**0.8594**	**0.8718**	**21.85**	**0.231**	**8.482**	**0.8350**	**0.8471**
StableSR	RealCE	17.53	0.244	39.17	0.8995	0.9172	16.57	0.368	76.68	0.7803	0.8095
ours	RealCE	**18.88**	**0.211**	**25.08**	**0.9085**	**0.9247**	**17.49**	**0.336**	**70.59**	**0.8475**	**0.8747**

Table 1. Quantitative comparison for the synthetic dataset CTR-TSR-Test and real-world dataset RealCE [8] with StableSR [12] and our method for ×2 and ×4 blind text image super-resolution.

5. More Visual Comparison

More visual results are shown in this section for both synthetic (CTR-TSR-Test) and real-world (RealCE [8]) datasets. Figure 3 and Figure 5 show the qualitative comparison of different methods for $\times 4$ and $\times 2$ text image super-resolution tasks on synthetic dataset CTR-TSR-Test. Figure 4 and Figure 6 show the qualitative comparison of different methods for $\times 4$ and $\times 2$ text image super-resolution tasks on the real-world dataset RealCE [8]. Visual results demonstrate that DiffTSR can not only restore the text image with severe degradation and complex strokes, but it also restores text images containing English letters, numbers, and diverse text styles.

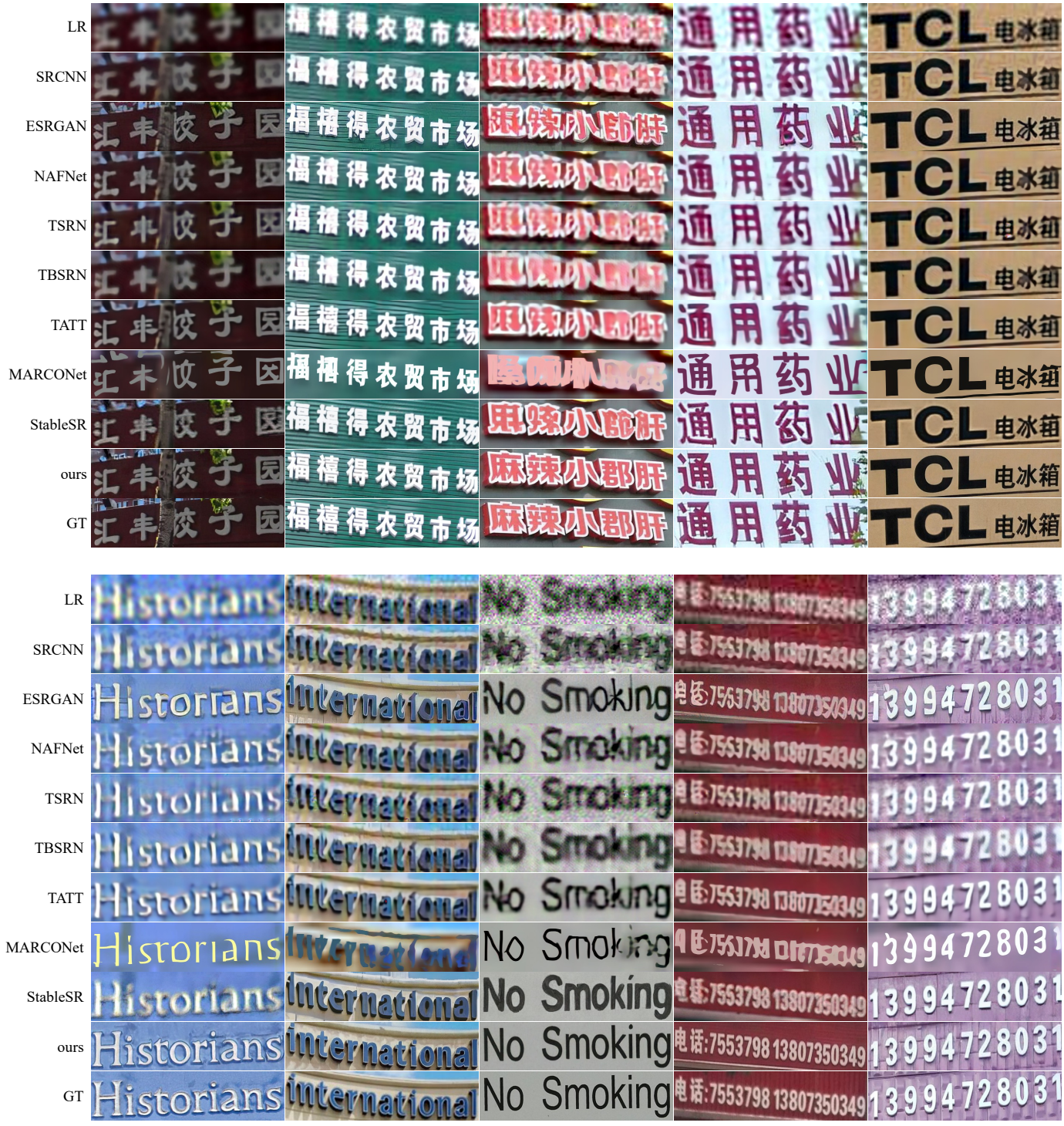

Figure 3. Qualitative comparison for the synthetic dataset CTR-TSR-Test with different methods including SRCNN [3], ESRGAN [14], NAFNet [2], TSRN [13], TBSRN [1], TATT [7], MARCONet [5], StableSR [12] and our method for $\times 4$ super-resolution.

Figure 4. Qualitative comparison for the real-world dataset RealCE [8] with different methods including SRCNN [3], ESRGAN [14], NAFNet [2], TSRN [13], TBSRN [1], TATT [7], MARCONet [5], StableSR [12] and our method for $\times 4$ super-resolution.

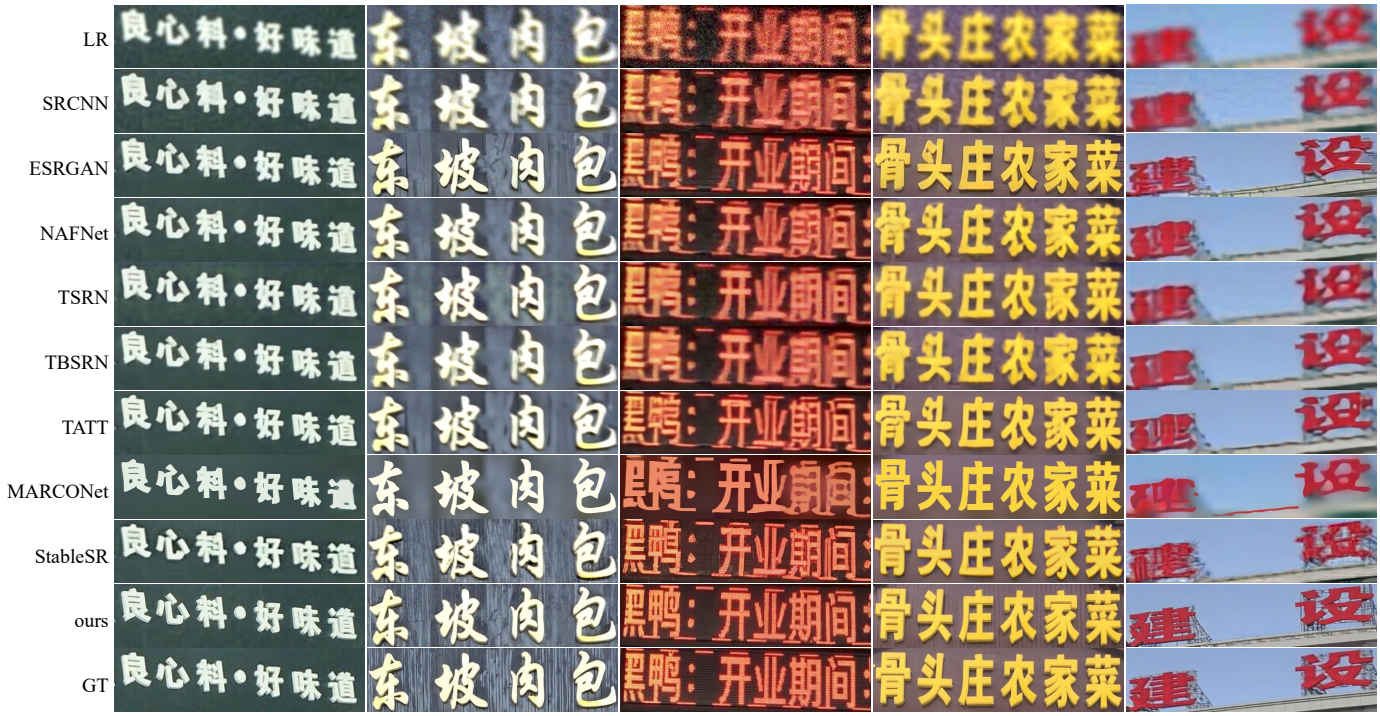

Figure 5. Qualitative comparison for the synthetic dataset CTR-TSR-Test with different methods including SRCNN [3], ESRGAN [14], NAFNet [2], TSRN [13], TBSRN [1], TATT [7], MARCONet [5], StableSR [12] and our method for $\times 2$ super-resolution.

Figure 6. Qualitative comparison for the real-world dataset RealCE [8] with different methods including SRCNN [3], ESRGAN [14], NAFNet [2], TSRN [13], TBSRN [1], TATT [7], MARCONet [5], StableSR [12] and our method for $\times 2$ super-resolution.

References

[1] Jingye Chen, Bin Li, and Xiangyang Xue. Scene text telescope: Text-focused scene image super-resolution. In *CVPR*, 2021. 1, 6, 7, 8

[2] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *ECCV*, 2022. 6, 7, 8

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2015. 6, 7, 8

[4] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *NIPS*, 2021. 1

[5] Xiaoming Li, Wangmeng Zuo, and Chen Change Loy. Learning generative structure prior for blind text image super-resolution. In *CVPR*, 2023. 6, 7, 8

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv:1711.05101*, 2017. 2

[7] Jianqi Ma, Zhetong Liang, and Lei Zhang. A text attention network for spatial deformation robust scene text image super-resolution. In *CVPR*, 2022. 6, 7, 8

[8] Jianqi Ma, Zhetong Liang, Wangmeng Xiang, Xi Yang, and Lei Zhang. A benchmark for chinese-english scene text image super-resolution. In *ICCV*, 2023. 5, 7, 8

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *arXiv:2112.10752*, 2021. 1

[10] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *TPAMI*, 2016. 1

[11] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2020. 2

[12] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *arXiv:2305.07015*, 2023. 5, 6, 7, 8

[13] Wenjia Wang, Enze Xie, Xuebo Liu, Wenhai Wang, Ding Liang, Chunhua Shen, and Xiang Bai. Scene text image super-resolution in the wild. In *ECCV*, 2020. 6, 7, 8

[14] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCVW*, 2018. 6, 7, 8