A. Training Details

Basic model information. All the models are trained on the real-world dataset. LR and HR images are resized into $128 \times 512 \times 3$ and $32 \times 128 \times 3$, respectively. Besides, we utilize the TDM and MoM module from [47] and load their pre-trained weights to process the text information and the text diffusion. When training the model, these two modules are frozen. For the Unet module, we utilize the pre-trained weights from the ResShift [43] model. Also, to fuse the textual feature from the MoM module with the image feature, after each basic layers on the ResShift, we insert the Transformer layer for the cross attention module. The inserted Transformer layer is randomly initialized. During training, only the Unet module and the MoM module is updated. Besides, in the modeling of the text sequence, the maximum length of the text is set to 24 and all the characters in the text sequence belong to an alphabet with $K = 6736$ characters including both Chinese and English characters as well as the numbers and special characters. The total step T is set to 18.

Data sampling strategy. During training, the data sampling strategy is involved. Specifically, for the HR image x_0, in the first 10K training steps, we only use the degraded image $\hat{x_0}$ as the LR image, to build the model's core SR capabilities. In the next 10K training steps, we mix the LR image y and $\hat{x_0}$ when constructing the HR-LR pairs to improve the ability of in-domain adaptation. Given the HR image x_0, the probabilities of using y and $\hat{x_0}$ are equal. Finally, in the remaining training rounds, we extend the training process by introducing the degraded LR image $\hat{y}$ into training. We train the model for 400K steps in total.

Deviations of Eq. (2). According to Bayes's theorem, we have

$$q(z_{t-1}|z_t, z_0, y) \propto q(z_t|z_{t-1}, y)q(z_{t-1}|z_0, y), \tag{10}$$

where

$$q(z_{t-1}|z_t, y) = \mathcal{N}(z_t; z_{t-1} + \alpha_t e_0, \kappa^2\alpha_t \mathbf{I}), \tag{11}$$

$$q(z_{t-1}|z_0, y) = \mathcal{N}(z_{t-1}; z_0 + \eta_{t-1} e_0, \kappa^2\eta_{t-1} \mathbf{I}). \tag{12}$$

We focus on the quadratic form in the exponent of $q(z_{t-1}|z_t, z_0, y_0)$, *i.e.*,

$$\begin{aligned}
&- \frac{(z_t - z_{t-1} - \alpha_t e_0)(z_t - z_{t-1} - \alpha_t e_0)^T}{2\kappa^2\alpha_t} - \frac{(z_{t-1} - z_0 - \eta_{t-1} e_0)(z_{t-1} - z_0 - \eta_{t-1} e_0)^T}{2\kappa^2\eta_{t-1}} \\
=& -\frac{1}{2}\left[\frac{1}{\kappa^2\alpha_t} + \frac{1}{\kappa^2\eta_{t-1}}\right] z_{t-1} z_{t-1}^T + \left[\frac{z_t - \alpha_t e_0}{\kappa^2\alpha_t} + \frac{z_0 + \eta_{t-1} e_0}{\kappa^2\eta_{t-1}}\right] z_{t-1}^T + \text{const} \\
=& -\frac{(z_{t-1} - \mu)(z_{t-1} - \mu)^T}{2\lambda^2} + \text{const},
\end{aligned} \tag{13}$$

where

$$\mu = \frac{\eta_{t-1}}{\eta_t} z_t + \frac{\alpha_t}{\eta_t} z_0, \lambda^2 = \kappa^2 \frac{\eta_{t-1}}{\eta_t} \alpha_t, \tag{14}$$

and const denotes the item that is independent of z_{t-1}. After substituting z_0 with the neural network $f_\theta(z_t, y, t)$, this quadratic form induces the Gaussian distribution of Eq. (6).

B. Procedure of Our Framework

The training and inference procedure of our framework can be summarized in Algorithm 1 and Algorithm 2, respectively.

C. Efficiency of Our model

To demonstrate the efficiency of our model, we present the total number of parameters and the average inference time per image in Table 3. For the average inference time, we randomly select 100 LR text images from the test set with the size of 128×512 as the input. Compared with the state-of-the-art diffusion-based text image SR method, DiffTSR, our model incorporates a lightweight SR prior, leading to an 82.2% reduction in parameters. Furthermore, our model achieves a 93.4% reduction in inference time, significantly improving efficiency over DiffTSR. These results highlight the effectiveness of our approach in balancing performance and computational cost.

Algorithm 1 Training of Our Framework

Initialize Training step $r = 0$
repeat
  $x_0, y, c_0 \sim q(x_0, y, c_0)$, where c_0 is the ground truth text on x_0.
  $z_0 = E(x_0)$, where E is the VQGAN encoder.
  Generate $\hat{x_0}$ and $\hat{y}$ via the pipeline in ESRGAN [36].
  if $r < 10000$ **then**
    $y' = \hat{x_0}$
  else if $10000 \leq r < 20000$ **then**
    $y' = \text{Uniform}(\hat{x_0}, y)$
  else
    $y' = \text{Uniform}(\hat{x_0}, y, \hat{y})$
  end if
  $\vec{\alpha}, c = \text{OCR}(y')$
  $z_y = E(y')$
  $\epsilon \sim \mathcal{N}(0, I)$
  $t \sim \text{Uniform}(\{1, 2, \cdots, T\})$
  $z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$
  $c_t \sim \mathcal{C}(c_t | \bar{\alpha}_t c_0 + \frac{1-\bar{\alpha}_t}{K})$
  $[\mathbf{I}_{cond_t}, \mathbf{C}_{cond_t}] = \text{MoM}([z_y, z_t], \vec{\alpha} \cdot c_t, t)$
  $\bar{z_0} = f_\theta(z_t, y, t, \mathbf{C}_{cond_t})$
  $\bar{x_0} = D(\bar{z_0})$, where D is the VQGAN decoder.
  $\bar{(c_0)} = \text{OCR}(\bar{x_0})$
  $\mathcal{L} = \lambda_1 \mathcal{L}_{L1}(x_0, \bar{(x_0)}) + \lambda_2 \mathcal{L}_{LPIPS}(x_0, \bar{(x_0)}) + \lambda_3 \mathcal{L}_{CE}(c_0, \bar{(c_0)})$
  Take gradient descent on $\nabla_\theta \mathcal{L}$
until Converged

Algorithm 2 Inference of our Framework

Input: LR image y
Output: HR image x

$z_y = E(y)$
$\epsilon \sim \mathcal{N}(0, I)$
$z_T = z_y + \kappa\epsilon$
$c_T = \text{OCR}(y)$
for $t = T \cdots 1$ **do**
  $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $z = 0$
  $[\mathbf{I}_{cond_t}, \mathbf{C}_{cond_t}] = \text{MoM}([z_y, z_t], \vec{\alpha} \cdot c_t, t)$
  Sample z_{t-1} via Eq. (6).
  $c_{pred,t} = \tau(c_t, \mathbf{I}_{cond_t}, t)$, where τ is the transformer decoder intended for the text diffusion module.
  $\tilde{\pi} = \left[\alpha_t c_t + \frac{1-\alpha_t}{K}\right] \odot \left[\bar{\alpha}_{t-1} + \frac{1-\bar{\alpha}_{t-1}}{K}\right]$
  $\pi_{post}(c_t, c_{pred,t}) = \frac{\tilde{\pi}}{\sum_{k=1}^{K} \pi_k}$
  $c_{t-1} \sim \mathcal{C}(c_{t-1} | \pi_{post}(c_t, c_{pred,t}))$ if $t > 1$ else $c_0 \sim \mathcal{C}(c_0 | c_{pred,t})$
end for
return $x = D(z_0)$ where D is the VQGAN decoder.

D. Future Works

Due to the limited performance of the OCR module, existing methods still make mistakes in the generated images. Therefore, in future work, we plan to incorporate reinforcement learning techniques to refine and correct the generated text, such as DPO [2]. Additionally, we aim to replace the OCR module with outputs from multi-modal large language models (MLLMs) [23, 24], which could potentially yield more accurate text outputs from the LR images.

Table 3. the total number of parameters and the average inference time on 100 randomly selected LR images of size 128×512 for both our model and DiffTSR.

	# parameters	Inference Time(s)
DiffTSR	874M	25.32
Ours	155M	1.68

E. More Visual Results

More visual results are shown in this section for the real-world datasets in Figure 7 and Figure 8. The visual results on the synthetic dataset are shown in Figure 9. Visual results show that our method effectively handle the LR images from the real-world scenarios that contains English letters, numbers and diverse text styles.

Figure 7. Qualitative comparison on the real-world dataset with different methods including SRCNN [6], NAFNet [5], ESRGAN [36], the existing SOTA method DiffTSR [47] and our method for $\times 4$ super-resolution.

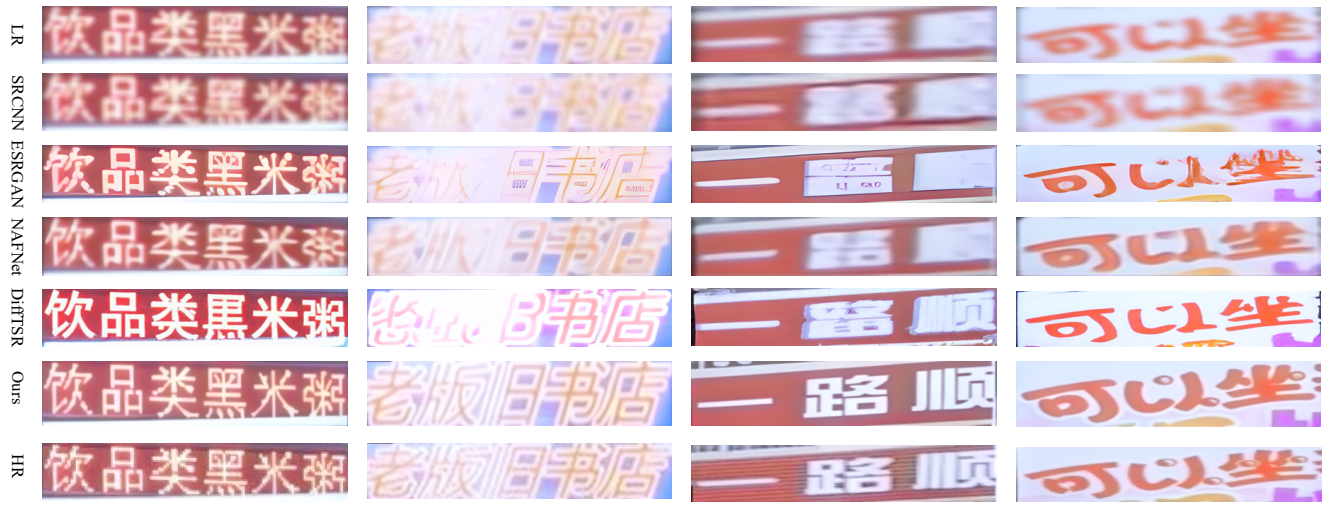

Figure 8. Qualitative comparison on the real-world dataset with different methods including SRCNN [6], NAFNet [5], ESRGAN [36], the existing SOTA method DiffTSR [47] and our method for $\times 4$ super-resolution.

Figure 9. Qualitative comparison on the synthetic dataset with different methods including SRCNN [6], NAFNet [5], ESRGAN [36], the existing SOTA method DiffTSR [47] and our method for $\times 4$ super-resolution.