Supplementary Material to "PEAN: A Diffusion-Based Prior-Enhanced Attention Network for Scene Text Image Super-Resolution"

Anonymous Authors

In this Supplementary Material, we provide:

- More details about the MLP-based denoising network, denoted as f_θ, in the TPEM. This is mentioned in the "Methodology" part (§ 3.2.2) of the main paper.
- More experiments and ablation studies on the TextZoom Dataset. This is mentioned in the "Experiments" part (§ 4.4) of the main paper.
- More visualization results on the dataset we built. This is also mentioned in the "Experiments" part (§ 4.1, 4.2) of the main paper.

A DETAILS OF THE DENOISING NETWORK

In this section, we present a detailed description of the denoising network, denoted as f_θ, employed in the TPEM. In prevalent diffusion models applied to tasks such as Single Image Super-Resolution (SISR) [10, 28] and image deblurring [26, 38], where the network is designed to process images, researchers often opt for the U-Net architecture [27] as the denoising network. However, in our work, the TPEM is designed to enhance the primary text prior, which is a recognition probability sequence. To achieve this objective, we introduce an MLP-based architecture. The input to f_θ consists of three components: the noisy text prior at timestep t (denoted as x_t), the primary text prior extracted from low-resolution (LR) images (denoted as P^{l}), and the timestep (denoted as t). Of these inputs, P^{l} is concatenated with x_t along the second dimension, serving as a conditioning factor for the denoising process. To reduce the dimension and obtain the fused feature x_t^0, we employ a 1D convolutional layer with a kernel size of 1×1. Simultaneously, the timestep t is encoded into a time embedding (denoted as t_e) using a positional encoding module [33]. Subsequently, we utilize four MLP blocks to refine the feature based on the time embedding, with the output of the final MLP layer representing the denoised feature at timestep t, which is also the input for timestep $t-1$. For a comprehensive overview of the architecture of f_θ, please refer to Table 1.

B MORE EXPERIMENTS ON TEXTZOOM

In this section, we conduct more experiments and ablation studies on the TextZoom benchmark [35] to further demonstrate that our proposed Prior-Enhanced Attention Network (PEAN) can serve as an effective alternative for scene text image super-resolution (STISR). TextZoom [35] is a common benchmark for STISR, containing 17367 and 4373 paired LR-HR images collected in natural scenarios for training and testing, respectively. According to the degree of blurriness, the testing set is divided into three subsets, namely easy (1619 pairs), medium (1411 pairs) and hard (1343 pairs). The sizes of LR and HR images are 16×64 and 32×128 respectively. Noteworthy, following the common practice in existing works, in this paper, the reported "Average" results are the weighted average on the three subsets of TextZoom, which is formulated as:

$$Acc_{avg} = \frac{Acc_e \cdot N_e + Acc_m \cdot N_m + Acc_h \cdot N_h}{N_e + N_m + N_h}, \tag{1}$$

where Acc_e, Acc_m and Acc_h denote the recognition accuracy on the "easy", "medium" and "hard" subsets respectively. N_e, N_m and N_h denote the number of images in the corresponding subset. For TextZoom, $N_e = 1619$, $N_m = 1411$, $N_h = 1343$.

Table 1: Architecture of the MLP-based denoising network. N is the size of the mini-batch. Grey rows show the components of MLP Block 1, similar to MLP Block 2, 3 and 4.

Input	Input size	Output	Output size	Module / Operation
x_t P^{l}	$[N, 26, 37]$ $[N, 26, 37]$	x_t^0	$[N, 52, 37]$	Concatenate
x_t^0	$[N, 52, 37]$	x_t^0	$[N, 26, 37]$	Convolution
t	$[N, 1]$	t_e	$[N, 1, 26]$	Positional Encoding
x_t^0	$[N, 26, 37]$	x_t^0	$[N, 26, 37]$	Batch Normalization
x_t^0	$[N, 26, 37]$	x_t^0	$[N, 26, 148]$	Linear
x_t^0	$[N, 26, 148]$	x_t^0	$[N, 26, 148]$	Swish [25] Function
t_e	$[N, 1, 26]$	t_e	$[N, 26, 1]$	Linear & Reshape
x_t^0 t_e	$[N, 26, 148]$ $[N, 26, 1]$	x_t^1	$[N, 26, 148]$	Repeat & Add
x_t^1 t_e	$[N, 26, 148]$ $[N, 1, 26]$	x_t^2	$[N, 26, 296]$	MLP Block 2
x_t^2 t_e	$[N, 26, 296]$ $[N, 1, 26]$	x_t^3	$[N, 26, 148]$	MLP Block 3
x_t^3 t_e	$[N, 26, 148]$ $[N, 1, 26]$	x_{t-1}	$[N, 26, 37]$	MLP Block 4

B.1 Recognition Accuracy of SOTA Recognizers

Following the common practice of existing works, in § 4.3 of the main paper, we introduce the recognition accuracy of SR images on three classic scene text recognizers, *i.e.*, ASTER [30], MORAN [20] and CRNN [29] for evaluation, and our proposed PEAN achieves new SOTA results with substantial performance improvement. Here we employ three recent Transformer-based recognizers, *i.e.*, MGP-STR [34], ABINet [9] and VisionLAN [36] for further evaluation to demonstrate the robustness of PEAN. The results are shown in Table 2, from which we can conclude that PEAN can still achieve the SOTA performance under the evaluation of recent recognizers. This further justifies that PEAN can surely increase both the resolution and readability of scene text images, regardless of which recognizer we choose for evaluation.

B.2 Quality of SR Images

Following the common practice of previous works, we use the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity

Table 2: The recognition accuracy of some mainstream STISR methods by three recent scene text recognizers on the three subsets of TextZoom. The best scores are shown in bold. Note that as to methods used for comparison, we adopt the pre-trained model released by their authors for evaluation.

Methods	Accuracy of MGP-STR [34] (%)				Accuracy of ABINet [9] (%)				Accuracy of VisionLAN [36] (%)			
	Easy	Medium	Hard	Average	Easy	Medium	Hard	Average	Easy	Medium	Hard	Average
LR	73.4	59.9	45.9	60.6	77.4	58.4	43.5	60.9	74.6	53.3	39.5	56.9
TSRN [35]	67.3	58.7	42.7	57.0	76.2	61.4	44.8	61.8	75.2	58.3	42.7	59.8
TBSRN [4]	72.3	62.9	45.9	61.2	80.2	65.6	48.3	65.7	78.1	62.7	45.3	63.1
TG [5]	71.7	64.7	46.3	61.6	79.8	67.1	49.1	66.3	78.2	63.9	44.3	63.2
TATT [22]	71.3	61.7	45.9	60.4	81.0	65.8	50.0	66.6	79.7	63.9	47.8	64.8
C3-STISR [44]	73.6	63.3	47.8	62.4	81.4	66.2	49.9	66.8	81.0	65.0	47.2	65.5
LEMMA [12]	73.8	65.8	48.6	63.5	83.3	69.5	52.7	69.4	81.7	68.3	49.9	67.6
PEAN	**76.5**	**68.4**	**52.2**	**66.4**	**86.3**	**73.1**	**56.5**	**72.9**	**83.9**	**71.3**	**53.2**	**70.4**
HR	85.2	81.8	76.1	81.3	94.9	90.6	82.7	89.8	94.7	88.8	80.0	88.3

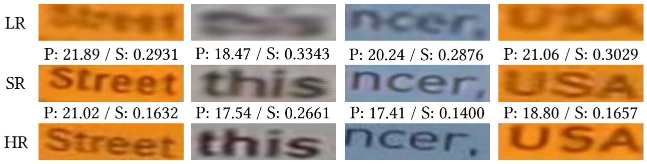

Figure 1: Visualizations about cases where SR images have lower PSNR and SSIM than LR images. "P" and "S" stand for "PSNR" and "SSIM" respectively.

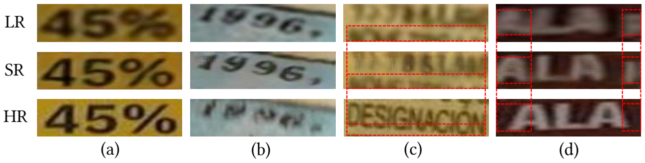

Figure 2: Visualizations of drawbacks of TextZoom.

Index Measure (SSIM) [37] metrics, which are widely utilized in the classic SISR task, to evaluate the quality of SR images. The results presented in Table 3 shows that the quality of SR images generated by our proposed PEAN is comparable to existing works.

Different from SISR, which aims at improving the image quality of LR natural images, STISR concentrates more on increasing the readability of scene text images [35, 45]. Therefore, PSNR and SSIM are *NOT* the most suitable metrics for STISR because we empirically find that readability is not closely related to image quality. Firstly, as shown in Figure 1, it is common that LR images have higher PSNR or SSIM than SR images. However, it is very difficult for us to distinguish text in images if we are given such LR images, but SR images generated by our proposed PEAN make this task easier. Therefore, methods with high PSNR or SSIM values do not necessarily produce readable images, and vice versa.

Secondly, some inherent drawbacks of TextZoom [35] make it unreasonable to adopt PSNR and SSIM for evaluation if we conduct experiments on this dataset. We roughly divide them into three categories, *i.e.*, difference of the background color, low-quality HR images and cutting out of position, as presented in Figure 2. Figure 2(a) is a representative example of the "difference of the background color" drawback, because it is evident that the background color of LR images is quite different from that of HR images, which further results in SR images with different backgrounds than HR images. Since PSNR and SSIM are full-reference image quality assessment metrics [2], and HR images are used for reference, this difference has a huge negative impact on the values of PSNR and SSIM for SR images. Figure 2(b) shows another drawback, *i.e.*, "low-quality HR images". We can find that for these image pairs, the quality of the HR image is even worse than that of the LR image. Therefore, PSNR and SSIM cannot serve as appropriate evaluation metrics because, for these images, higher PSNR and SSIM mean poorer SR results.

Another common drawback of TextZoom, namely "cutting out of position", is illustrated in Figure 2(c, d). According to Wang *et al.* [35], TextZoom is constructed by cutting scene text images of different resolutions from the RealSR [3] and SRRAW [41] datasets. This manual process inevitably causes misalignment as shown in Figure 2(c, d). Since HR images are used as reference images, and PSNR and SSIM are only calculated at the pixel level, their values will not be high regardless of the clarity of the SR results.

Aside from our analysis, some recent works [12, 17] on STISR also find the same issue. Guo *et al.* [12] state that their proposed LEM concentrates more on the restoration of the character areas instead of the background areas, which occupies most of a scene text image, so there will be a reduction of PSNR and SSIM. Liu *et al.* [17] also draw a conclusion that PSNR and SSIM are only partially aligned with human perception when evaluating the quality of scene text images. In a word, we claim that the recognition accuracy of scene text recognizers is the most suitable evaluation metric for STISR. PSNR and SSIM are not reliable due to their inherent full-reference property and the intrinsic drawbacks of the TextZoom dataset. Thus, values of PSNR and SSIM are only provided for reference. Low PSNR and SSIM do not necessarily mean the model is not powerful enough in STISR.

B.3 Additional Ablation Study

Here we provide more ablation studies to thoroughly analyze the effectiveness of each module we propose, thereby demonstrating

Table 3: The PSNR and SSIM of some mainstream STISR methods on the three subsets of TextZoom. Best scores are bold.

Methods	PSNR				SSIM			
	Easy	Medium	Hard	Average	Easy	Medium	Hard	Average
TSRN [35]	**25.07**	18.86	19.71	21.42	0.8897	0.6676	0.7302	0.7690
TSRGAN [8]	24.22	19.17	19.99	21.29	0.8791	0.6770	0.7420	0.7718
TBSRN [4]	23.82	19.17	19.68	21.05	0.8660	0.6533	0.7490	0.7614
PCAN [43]	24.57	19.14	**20.26**	21.49	0.8830	0.6781	0.7475	0.7752
TG [5]	23.34	**19.66**	19.90	21.10	0.8369	0.6499	0.6986	0.7341
TPGSR [21]	24.35	18.73	19.93	21.18	0.8860	0.6784	0.7507	0.7774
TATT [22]	24.72	19.02	20.31	**21.52**	**0.9006**	**0.6911**	**0.7703**	**0.7930**
C3-STISR† [44]	21.78	18.49	19.60	20.05	0.8529	0.6465	0.7125	0.7432
LEMMA† [12]	23.56	18.94	19.63	20.86	0.8748	0.6869	0.7486	0.7754
PEAN	23.76	19.53	20.20	21.30	0.8655	0.6795	0.7287	0.7635

the reasonableness and superiority of our proposed PEAN. Consistent with the main paper, all the experiments are conducted on TextZoom [35] and we report the recognition accuracy of ASTER [30].

B.3.1 *Ablation Study on the TPEM.*

Paradigm and Network Architecture. Our proposed TPEM is a diffusion-based module that employs an MLP-based denoising network, denoted as f_θ. In contrast, many researchers in this field tend to utilize the U-Net architecture [27] as the denoising network in mainstream diffusion models [10, 26, 28, 38]. Therefore, we conduct experiments to demonstrate the suitability of the MLP architecture for processing the recognition probability sequence. Additionally, we compare the diffusion-based paradigm with the traditional regression-based paradigm. In the regression-based approach, the denoising network takes the primary text prior, denoted as P^{l}, as input and generates the ETP, denoted as P^{e}. The architectures of networks used in both paradigms are similar, with the key difference being that the only input of the network is P^{l} under the regression-based paradigm. The results presented in Table 4 indicate the following: (1) For both diffusion-based and regression-based methods, the MLP is more suitable than the U-Net for processing the recognition probability sequence. (2) The diffusion-based method outperforms the regression-based method, especially when employing MLP as the denoising network. This superiority can be attributed to the powerful distribution mapping capabilities of diffusion models [39].

Table 4: Ablation study about the paradigm and network architecture of TPEM.

Methods	Easy	Medium	Hard	Average
Regression (U-Net)	80.0	64.9	46.5	64.8
Regression (MLP)	80.9	65.5	47.2	65.6
Diffusion (U-Net)	79.6	65.3	46.9	64.9
Diffusion (MLP)	**84.5**	**71.4**	**52.9**	**70.6**

Loss Functions. As demonstrated in § 3.2.2 of the main paper, the optimization process in the TPEM is supervised through the utilization of the MAE and CTC loss [11]. In this part, we conduct experiments to validate the choice of the loss functions. In addition to the aforementioned losses, we also introduce the Kullback-Leibler (KL) divergence loss in this experiment. Its purpose is to minimize the discrepancy between the ETP (referred to as P^{e}) and TP-HR (denoted as P^{h}). The results, as presented in Table 5, reveal the following insights: (1) The MAE loss, a fundamental component introduced in the pioneering work of diffusion models [13], proves to be essential. Combining the MAE loss with either the KL divergence loss or the CTC loss leads to an improvement in performance. (2) In our work, the introduction of the CTC loss provides an improvement in performance with **+4.8** compared with employing the MAE loss only. This combined loss results in a more refined P^{e}, which in turn plays a pivotal role in guiding the SR network to generate images with enhanced semantic accuracy.

Table 5: Ablation study about the loss function for TPEM.

Loss Functions	Easy	Medium	Hard	Average
MAE	80.5	65.8	48.0	65.8
KL	79.9	65.2	47.0	65.1
CTC [11]	82.0	69.0	51.5	68.4
MAE + KL	83.9	70.0	51.2	69.4
MAE + CTC [11]	**84.5**	**71.4**	**52.9**	**70.6**

Sampling Strategy and Timestep. To strike a balance between performance and efficiency, we exploit the sampling strategy proposed in DDIM [31] in the sampling process of the TPEM, with a timestep value of $S = 1$. As demonstrated by Song *et al.* [31], the traditional sampling strategy of the DDPM [13] with a large number of steps (T steps, where $T \gg S$) can be exceedingly time-consuming. In this section, we conduct experiments to validate the effectiveness and efficiency of our proposed PEAN with the chosen sampling strategy and timestep.

Initially, we perform experiments using the sampling strategy of DDPM [13] while varying the timestep, selecting four different values for our experiments. The results presented in Table 6 reveal that this sampling strategy is not efficient. Subsequently, we substitute such sampling strategy with the one proposed in [31]. As demonstrated by Song *et al.* [31], with this strategy, smaller timesteps will

† Considering that the paper of C3-STISR [44] and LEMMA [12] only offers the average value, we measure the PSNR and SSIM of the publicly available pre-trained models to report values on the three subsets.

Table 6: The performance of PEAN with the sampling strategy of the DDPM [13] under different sampling timesteps. "Duration" is the time the model takes to process an image.

Timesteps	Easy	Medium	Hard	Average	Duration (s)
$T = 200$	80.2	66.2	48.8	66.0	0.29
$T = 500$	82.4	66.3	47.9	66.6	0.66
$T = 1000$	80.0	65.7	48.3	65.7	1.11
$T = 2000$	80.7	66.1	49.1	66.3	2.15

result in equal or even superior performance, prompting us to explore five different timesteps in this set of experiments. The results showcased in Table 7 verify that this modified sampling strategy speeds up the inference phase of the model. Additionally, compared with $T = 500$, which yields the best performance for DDPM in our experiments, PEAN exhibits an improvement in performance by approximately **+4.0** in average with this kind of sampling strategy.

Table 7: The performance of PEAN with the sampling strategy of the DDIM [31] under different sampling timesteps. "Duration" is the time the model takes to process an image.

Timesteps	Easy	Medium	Hard	Average	Duration (s)
$S = 1$	**84.5**	**71.4**	**52.9**	**70.6**	**0.04**
$S = 5$	79.7	65.9	46.8	65.1	0.09
$S = 10$	81.7	69.4	50.6	68.2	0.10
$S = 100$	83.2	69.0	50.7	68.6	0.19
$S = 500$	82.4	68.5	50.5	68.1	0.68

Furthermore, we compare the efficiency of our proposed PEAN with other mainstream text prior-based STISR methods [12, 21, 22, 40, 44]. The results displayed in Table 8 demonstrate that PEAN is on par with TPGSR [21], TATT [22] and C3-STISR [44] in terms of speed, while achieving an average performance improvement of **+6.5**. It even outperforms the two recent works, *i.e.*, LEMMA [12] and RTSRN [40], in both speed and performance. In summary, our proposed PEAN stands as an effective and efficient solution when compared to previous works.

Table 8: The performance of the mainstream text prior-based STISR methods. "Duration" is the time the model takes to process an image.

Methods	Easy	Medium	Hard	Average	Duration (s)
TPGSR [21]	78.9	62.7	44.5	62.8	0.03
TATT [22]	78.9	63.4	45.4	63.6	**0.02**
C3-STISR [44]	79.1	63.3	46.8	64.1	0.03
LEMMA [12]	81.1	66.3	47.4	66.0	0.07
RTSRN [40]	80.4	66.1	49.1	66.2	0.10
PEAN	**84.5**	**71.4**	**52.9**	**70.6**	0.04

B.3.2 *Ablation Study on the AMM.*

Necessity of LAM and GAM. Strip-wise attention and its variants have found application across various computer vision tasks [6, 14, 32]. However, many of these approaches focus solely on local horizontal and vertical attention, neglecting the incorporation of global contextual information. This study aims at justifying the indispensability of simultaneously employing both LAM and GAM in the STISR task. Table 9 illustrates the following key observations: (1) Upon removing both LAM and GAM, the model exhibits trivial performance. Notably, the addition of LAM improves network performance by **+1.9**, while the inclusion of GAM results in a further **+3.0** improvement. This underscores the effectiveness of the self-attention mechanism as a component of the feature extractor. (2) Utilizing LAM or GAM alone yields limited performance gains. However, the combination of LAM and GAM results in a substantial performance improvement of **+10.1**. This underlines the complementary nature of signals brought by LAM and GAM to the feature extractor. The reason is that LAM can effectively extract features on a per-character basis, while GAM facilitates interaction between characters, enhancing the ability of the model to learn the semantics of text in images.

Table 9: Analysis on the necessity of LAM and GAM.

LAM	GAM	Easy	Medium	Hard	Average
		75.5	60.4	42.5	60.5
✓		76.8	62.7	44.6	62.4
	✓	78.5	63.0	45.8	63.5
✓	✓	**84.5**	**71.4**	**52.9**	**70.6**

Number of Blocks. In this part, we evaluate how the number of blocks in the AMM affects the performance of our model. According to results illustrated in Table 10, we find that when $N = 6$, the model achieves overall the best performance. Therefore, we empirically choose $N = 6$ as the default setting for all the experiments in the main paper and this Supplementary Material.

Table 10: Analysis on the number of blocks in the AMM.

N	Easy	Medium	Hard	Average
1	81.9	67.4	48.0	66.8
2	82.8	67.9	50.6	68.1
3	83.4	68.0	50.3	68.3
4	83.8	70.5	53.1	70.1
5	84.3	70.4	**53.3**	70.3
6	**84.5**	**71.4**	52.9	**70.6**
7	84.2	71.0	52.9	70.3
8	83.1	69.1	51.2	68.8

B.3.3 *Ablation Study on the MTL Paradigm.*

Features Serving as the Input of the ARM. As shown in Figure 2 of the main paper, in the training phase, the output feature of the AMM, *i.e.*, the output of the N^{th} block ($N = 6$ in our paper) is sent to the ARM. Then the ARM extracts features to perform the text recognition task in the MTL paradigm. In this part, we investigate the most proper input feature for the ARM. As presented in Table 11, we can find that: (1) Even without the ARM and the MLT paradigm, our proposed PEAN can attain the performance of 67.9 on average, surpassing the current SOTA method [17] by **+1.5**. This reveals that the ARM and the MTL paradigm employed in our proposed PEAN are not the sole components contribute to the SOTA performance. (2) The adoption of the ARM can truly facilitate the training process, bringing an additional improvement in performance by **+2.7**. (3) However, if the inappropriate feature is sent into the ARM, the performance is even worse than that without the ARM. Our experiments show that sending the output of the 6th block into the ARM is the best choice.

Table 11: Ablation study on features for the input of the ARM. The first line indicates the case that we do not employ the ARM for assistance. B_i denotes the i^{th} block of the AMM.

Input of ARM	Easy	Medium	Hard	Average
w/o ARM	81.4	68.8	50.7	67.9
B_1 output	79.1	64.6	47.4	64.7
B_2 output	84.2	69.9	51.5	69.5
B_3 output	80.1	65.9	45.4	64.9
B_4 output	79.2	64.3	47.1	64.5
B_5 output	80.1	64.7	45.1	64.4
B_6 output	**84.5**	**71.4**	**52.9**	**70.6**
SRM output	79.6	64.4	46.5	64.5

Performance Gain from the MTL Paradigm. Here we provide a more comprehensive comparison between the AMM and the SRB [35]. While Table 4 in our main paper primarily compares the AMM and the SRB under the MTL paradigm, we conduct an additional comparison by excluding such paradigm. The results are presented in Table 12. Our findings indicate the following: (1) Without the MTL paradigm, the AMM still outperforms the SRB by an average of **+4.2**. The introduction of it leads to an additional improvement of **+1.7**. (2) The utilization of the MTL paradigm contributes to an improvement of performance for both the AMM and the SRB. Notably, the combination of the AMM and the MTL paradigm yields superior performance.

Table 12: Comparison between the AMM and the SRB.

Modules	MTL	Easy	Medium	Hard	Average
SRB [35]		79.1	62.7	46.1	63.7
	✓	80.1	64.4	46.4	64.7
AMM		81.4	68.8	50.7	67.9
	✓	**84.5**	**71.4**	**52.9**	**70.6**

Given that the MTL paradigm can potentially enhance other STISR methods, we introduce it into previous text prior-based STISR methods [12, 21, 22, 40, 44]. This exploration aims to validate the uniqueness of PEAN, as simply integrating the MTL paradigm with existing approaches cannot yield significant performance improvements. In line with our proposed PEAN, we input the output of the last block of the SRB (or MSRB for RTSRN [40]) from these models into the ARM and apply the MTL paradigm during training.

Table 13: Performance of other mainstream text prior-based models when they are equipped with the MTL paradigm.

Methods	MTL	Easy	Medium	Hard	Average
TPGSR [21]		**78.9**	**62.7**	**44.5**	**62.8**
	✓	0.01	0.03	0.01	0.02
TATT [22]		**78.9**	**63.4**	**45.4**	**63.6**
	✓	**78.9**	63.3	44.7	63.4
C3-STISR [44]		79.1	63.3	**46.8**	64.1
	✓	**79.9**	**63.4**	46.4	**64.3**
LEMMA [12]		**81.1**	**66.3**	**47.4**	**66.0**
	✓	76.2	59.0	43.9	60.7
RTSRN [40]		**80.4**	**66.1**	**49.1**	**66.2**
	✓	73.0	56.3	36.9	56.5

The results presented in Table 13 highlight a distinctive pattern: the ARM does not consistently improve the performance of other text prior-based STISR methods. In certain instances, the inclusion of the MTL paradigm even leads to a degradation in performance. This observation underscores the idea that the MTL paradigm, which is integral to the optimization phase of our proposed PEAN, are not universally beneficial components for achieving superior performance across all the STISR methods. Its effectiveness in boosting STISR performance is realized specifically when used in conjunction with our proposed PEAN.

B.3.4 Ablation Study on the Loss Functions.

Performance Gain from the SFM Loss. In the optimization phase of PEAN, we incorporate the SFM loss [5], a loss function not utilized by previous text prior-based methods. To assess the efficacy of it, we conduct experiments by introducing it into the optimization phase of established text prior-based STISR methods [12, 21, 22, 40, 44]. This analysis aims to demonstrate that the SFM loss alone is insufficient to achieve superior performance, underscoring the necessity of developing PEAN. Considering that LEMMA [12] and RTSRN [40] employ the text-focus loss [4], a loss function working similar with the SFM loss, we deliberately abandon the text-focus loss when training these two models during our experiments. The results presented in Table 14 confirm our argument, showing that simply introducing the SFM loss into existing methods does not yield significant performance improvements. Additionally, our experiments demonstrate the rationality of incorporating the SFM loss into the optimization of PEAN. This integration proves beneficial, contributing to an observable performance boost for PEAN.

Weights of the Loss Functions. In this part, we conduct experiments to find out the best values of weights of the five loss functions, *i.e.*, λ_1 to λ_5 in the main paper. Taking weights in previous works into account [5, 35, 44], we select λ_3 in $[0, 1]$ with an interval of 0.2. Similarly, λ_4 is selected in $[0, 100]$ with an interval of 25. λ_1, λ_2 and λ_5 are selected in $[0, 2]$ with an interval of 0.5. The results are presented in Figure 3, from which we can find that the best combination of the weight of each loss is $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, $\lambda_3 = 0.8$, $\lambda_4 = 75$ and $\lambda_5 = 1.0$. Too low or too high weights will lead to trivial performance. Therefore, we choose $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, $\lambda_3 = 0.8$, $\lambda_4 = 75$ and $\lambda_5 = 1.0$ as the default setting of weights of the losses in the main paper and this Supplementary Material.

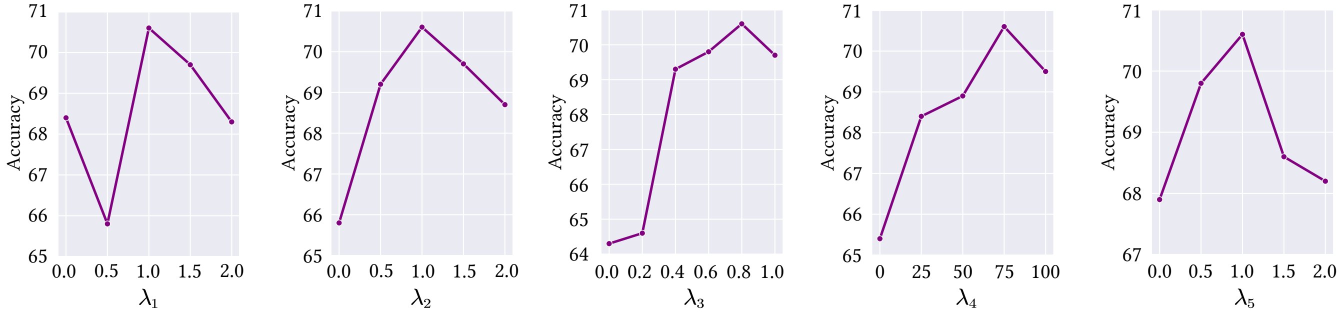

Figure 3: Ablation study on the weight of five loss functions. We report the average recognition accuracy calculated by Eq. (1).

Table 14: Performance of other mainstream text prior-based models when they are equipped with the SFM loss [5].

Methods	SFM Loss	Easy	Medium	Hard	Average
TPGSR [21]		**78.9**	**62.7**	**44.5**	**62.8**
	✓	74.9	60.1	41.3	59.8
TATT [22]		**78.9**	**63.4**	**45.4**	**63.6**
	✓	78.3	62.3	46.3	63.3
C3-STISR [44]		79.1	63.3	**46.8**	64.1
	✓	**79.6**	**63.9**	46.5	**64.4**
LEMMA [12]		**81.1**	**66.3**	**47.4**	**66.0**
	✓	76.2	62.4	43.8	61.8
RTSRN [40]		**80.4**	**66.1**	**49.1**	**66.2**
	✓	1.1	2.8	1.9	1.9

B.3.5 *Ablation Study on Other Modules and Settings*.

Kind of Shallow Feature Extractor. As shown in Figure 2 of the main paper, a single convolutional layer is applied to extract the shallow feature F^s. Recently, CNN-Transformer-based architecture is widely adopted in SISR [16, 19, 42], so here we perform experiments to adopt CNN-Transformer-based modules as Shallow Feature Extractors. As presented in Table 15, it is surprising to find that a single convolutional layer is enough for shallow feature extraction. Redundant ViT layers only make the model difficult to optimize and degrade the SR performance. Therefore, we use a single convolutional layer as the shallow feature extractor.

Table 15: Analysis on kind of the shallow feature extractor.

Extractors	Easy	Medium	Hard	Average
Conv only	**84.5**	**71.4**	**52.9**	**70.6**
Conv + ViT [7]	77.9	62.4	43.7	62.4
Conv + Swin [18]	76.5	62.1	44.0	61.9

Performance Gain from the Pre-training Process. As discussed in § 4.2 of the main paper, our proposed PEAN involves an initial phase where we exclude the TPEM and pre-train the model using TP-HR. Subsequently, the TPEM is introduced, and the weights of parameters obtained from the pre-training phase are initialized for the ongoing fine-tuning process. In this part, we conduct experiments aimed at investigating the impact of this configuration. We also extend this approach to established text prior-based STISR methods [12, 21, 22, 40, 44] to demonstrate that the pre-training process alone does not result in a substantial performance improvement for these methods, highlighting the necessity of proposing PEAN. The results shown in Table 16 affirm our argument. Notably, even without the pre-training process, our proposed PEAN outperforms the SOTA STISR method, *i.e.*, TextDiff [17], by an average of **+1.1**. The inclusion of the pre-training process setting leads to an additional improvement of **+3.1**.

Table 16: Performance of the mainstream text prior-based models when equipped with the pre-training process.

Methods	Pre-training	Easy	Medium	Hard	Average
TPGSR [21]		**78.9**	**62.7**	**44.5**	**62.8**
	✓	77.6	61.4	43.6	61.9
TATT [22]		78.9	**63.4**	45.4	63.6
	✓	**79.5**	**63.4**	**45.9**	**64.0**
C3-STISR [44]		**79.1**	**63.3**	**46.8**	**64.1**
	✓	77.8	60.5	43.4	61.7
LEMMA [12]		81.1	66.3	47.4	66.0
	✓	**81.7**	**67.3**	**48.5**	**66.9**
RTSRN [40]		**80.4**	**66.1**	**49.1**	**66.2**
	✓	79.1	62.9	45.9	63.7
PEAN		82.5	67.8	49.0	67.5
	✓	**84.5**	**71.4**	**52.9**	**70.6**

Compatibility with different TPGs. We adopt the pre-trained PARSeq [1] as the TPG, which is stronger than the CRNN [29] applied in [21, 22, 40, 44] and the ABINet [9] employed in [12]. For a fair comparison, we conduct experiments wherein CRNN, ABINet and PARSeq are introduced as the TPG respectively in these works. Notably, as depicted in Figure 2 of [12], LEMMA relies on the attention map sequence generated by the TPG for character location enhancement. However, CRNN is not an attention-based TPG and is unsuitable for LEMMA. To address this, we treat the output of the last convolutional layer in CRNN as the pseudo attention map and apply several linear layers to adjust its dimensions.

The results presented in Table 17 demonstrates that our proposed PEAN exhibits compatibility with the text prior generated by CRNN, ABINet, and PARSeq. Although PARSeq [1] is more powerful than CRNN [29] and ABINet [9], previous works fail to benefit a lot

Table 17: Performance of the mainstream text prior-based models when they are equipped with different TPGs.

Methods	CRNN [29]	ABINet [9]	PARSeq [1]	Easy	Medium	Hard	Average
TPGSR [21]	✓			78.9	62.7	44.5	62.8
		✓		73.0	55.4	39.5	57.0
			✓	72.3	55.3	38.9	56.6
TATT [22]	✓			78.9	63.4	45.4	63.6
		✓		75.4	56.6	40.5	58.6
			✓	74.1	56.6	40.6	58.2
C3-STISR [43]	✓			79.1	63.3	46.8	64.1
		✓		72.5	54.2	38.8	56.2
			✓	75.5	56.7	38.5	58.1
LEMMA [12]	✓			76.1	58.8	42.7	60.3
		✓		81.1	66.3	47.4	66.0
			✓	77.6	60.5	44.7	62.0
RTSRN [40]	✓			80.4	66.1	49.1	66.2
		✓		3.3	2.9	2.2	2.9
			✓	80.2	67.5	46.3	65.7
PEAN	✓			80.8	66.1	48.6	66.2
		✓		82.2	66.0	47.7	66.4
			✓	**84.5**	**71.4**	**52.9**	**70.6**

from the text prior generated by it. However, with the pre-trained PARSeq as the TPG, our proposed PEAN outperforms the current SOTA text prior-based STISR method, *i.e.*, RTSRN [40] by **+4.9** on average. When ABINet is used as the TPG, RTSRN exhibits trivial performance, whereas PEAN continues to demonstrate superior performance. This indicates that our proposed PEAN has good adaptability to the text prior generated by all the three TPGs.

C VISUALIZATIONS ON OTHER DATASETS

In this section, we provide more visualization results on the dataset we built to show the generalization of our proposed PEAN and display it ability to restore visual structure. As mentioned in the main paper, we employ IIIT5K [23], SVTP [24] and IC15 [15] for evaluation. We select 651 images whose resolution is no greater than 16 × 64 as LR images and input them directly into the PEAN trained on TextZoom. The results are shown in Figure 4, from which we can conclude that: (1) Previous works result in artifacts, while our proposed PEAN can address this issue. (2) Our proposed PEAN works well in terms of images with long or deformed text, while existing works tend to generate incorrect SR results.

REFERENCES

[1] Darwin Bautista and Rowel Atienza. 2022. Scene Text Recognition with Permuted Autoregressive Sequence Models. In *Proceedings of the European Conference on Computer Vision*. 178–196.

[2] Sebastian Bosse, Dominique Maniry, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. 2018. Deep Neural Networks for No-Reference and Full-Reference Image Quality Assessment. *IEEE Transactions on Image Processing* 27, 1 (2018), 206–219.

[3] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. 2019. Toward Real-World Single Image Super-Resolution: A New Benchmark and a New Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3086–3095.

[4] Jingye Chen, Bin Li, and Xiangyang Xue. 2021. Scene text telescope: Text-focused scene image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12026–12035.

[5] Jingye Chen, Haiyang Yu, Jianqi Ma, Bin Li, and Xiangyang Xue. 2022. Text Gestalt: Stroke-aware scene text image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 285–293.

[6] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. 2022. CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12114–12124.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations*.

[8] Chuantao Fang, Yu Zhu, Lei Liao, and Xiaofeng Ling. 2021. TSRGAN: Real-world text image super-resolution based on adversarial learning and triplet attention. *Neurocomputing* 455 (2021), 88–96.

[9] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. 2021. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7098–7107.

[10] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. 2023. Implicit Diffusion Models for Continuous Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10021–10030.

[11] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*. 369–376.

[12] Hang Guo, Tao Dai, Guanghao Meng, and Shu-Tao Xia. 2023. Towards Robust Scene Text Image Super-resolution via Explicit Location Enhancement. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 782–790.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Proceedings of the Advances in Neural Information Processing Systems*. 6840–6851.

[14] Zilong Huang, Xinggang Wang, Yunchao Wei, Lichao Huang, Humphrey Shi, Wenyu Liu, and Thomas S. Huang. 2023. CCNet: Criss-Cross Attention for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 6 (2023), 6896–6908.

[15] 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. 2015. ICDAR 2015 competition on Robust Reading. In *Proceedings of IEEE International Conference on Document Analysis and Recognition*. 1156–1160.

[16] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. SwinIR: Image Restoration Using Swin Transformer. In *Proceedings*

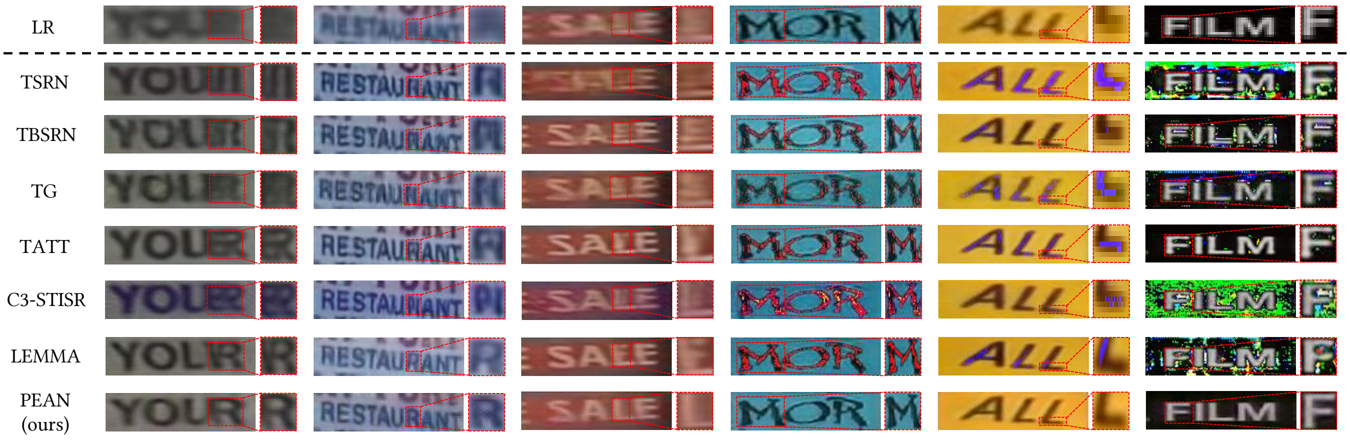

Figure 4: Visualization of SR results on three classic scene text image datasets.

of the IEEE/CVF International Conference on Computer Vision Workshops*. 1833–1844.

[17] Baolin Liu, Zongyuan Yang, Pengfei Wang, Junjie Zhou, Ziqi Liu, Ziyi Song, Yan Liu, and Yongping Xiong. 2023. TextDiff: Mask-Guided Residual Diffusion Models for Scene Text Image Super-Resolution. *arXiv preprint arXiv:2308.06743* (2023).

[18] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10012–10022.

[19] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. 2022. Transformer for Single Image Super-Resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition Workshops*. 456–465.

[20] Canjie Luo, Lianwen Jin, and Zenghui Sun. 2019. Moran: A multi-object rectified attention network for scene text recognition. *Pattern Recognition* 90 (2019), 109–118.

[21] Jianqi Ma, Shi Guo, and Lei Zhang. 2023. Text prior guided scene text image super-resolution. *IEEE Transactions on Image Processing* 32 (2023), 1341–1353.

[22] Jianqi Ma, Zhetong Liang, and Lei Zhang. 2022. A Text Attention Network for Spatial Deformation Robust Scene Text Image Super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5911–5920.

[23] Anand Mishra, Karteek Alahari, and C. V. Jawahar. 2012. Scene Text Recognition using Higher Order Language Priors. In *Proceedings of the British Machine Vision Conference*. 1–11.

[24] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. 2013. Recognizing Text with Perspective Distortion in Natural Scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 569–576.

[25] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2018. Searching for Activation Functions. In *Proceedings of the International Conference on Learning Representations Workshops*.

[26] Mengwei Ren, Mauricio Delbracio, Hossein Talebi, Guido Gerig, and Peyman Milanfar. 2023. Multiscale Structure Guided Diffusion for Image Deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10721–10733.

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*. 234–241.

[28] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. 2023. Image Super-Resolution via Iterative Refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 4 (2023), 4713–4726.

[29] Baoguang Shi, Xiang Bai, and Cong Yao. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 11 (2016), 2298–2304.

[30] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. 2018. ASTER: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 9 (2018), 2035–2048.

[31] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *Proceedings of the International Conference on Learning Representations*.

[32] Fu-Jen Tsai, Yan-Tsung Peng, Yen-Yu Lin, Chung-Chi Tsai, and Chia-Wen Lin. 2022. Stripformer: Strip Transformer for Fast Image Deblurring. In *Proceedings of the European Conference on Computer Vision*. 146–162.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of the Advances in Neural Information Processing Systems*. 5998–6008.

[34] Peng Wang, Cheng Da, and Cong Yao. 2022. Multi-granularity Prediction for Scene Text Recognition. In *Proceedings of the European Conference on Computer Vision*. 339–355.

[35] Wenjia Wang, Enze Xie, Xuebo Liu, Wenhai Wang, Ding Liang, Chunhua Shen, and Xiang Bai. 2020. Scene text image super-resolution in the wild. In *Proceedings of the European Conference on Computer Vision*. 650–666.

[36] Yuxin Wang, Hongtao Xie, Shancheng Fang, Jing Wang, Shenggao Zhu, and Yongdong Zhang. 2021. From Two to One: A New Scene Text Recognizer with Visual Language Modeling Network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14174–14183.

[37] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

[38] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G. Dimakis, and Peyman Milanfar. 2022. Deblurring via Stochastic Refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16272–16282.

[39] Bin Xia, Yulun Zhang, Shiyin Wang, Yitong Wang, Xinglong Wu, Yapeng Tian, Wenming Yang, and Luc Van Gool. 2023. DiffIR: Efficient Diffusion Model for Image Restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13095–13105.

[40] Wenyu Zhang, Xin Deng, Baojun Jia, Xingtong Yu, Yifan Chen, Jin Ma, Qing Ding, and Xinming Zhang. 2023. Pixel Adapter: A Graph-Based Post-Processing Approach for Scene Text Image Super-Resolution. In *Proceedings of the ACM International Conference on Multimedia*. 2168–2179.

[41] Xuaner Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun. 2019. Zoom to Learn, Learn to Zoom. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3762–3770.

[42] Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. 2022. Efficient Long-Range Attention Network for Image Super-Resolution. In *Proceedings of the European Conference on Computer Vision*. 649–667.

[43] Cairong Zhao, Shuyang Feng, Brian Nlong Zhao, Zhijun Ding, Jun Wu, Fumin Shen, and Heng Tao Shen. 2021. Scene text image super-resolution via parallelly contextual attention network. In *Proceedings of the ACM International Conference on Multimedia*. 2908–2917.

[44] Minyi Zhao, Miao Wang, Fan Bai, Bingjia Li, Jie Wang, and Shuigeng Zhou. 2022. C3-STISR: Scene Text Image Super-resolution with Triple Clues. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 1707–1713.

[45] Shipeng Zhu, Zuoyan Zhao, Pengfei Fang, and Hui Xue. 2023. Improving Scene Text Image Super-Resolution via Dual Prior Modulation Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 3843–3851.