Self-Supervised Pre-training with Symmetric Superimposition Modeling for Scene Text Recognition

Zuan Gao, **Yuxin Wang***, **Yadong Qu**, **Boqiang Zhang**,
Zixiao Wang, **Jianjun Xu**, **Hongtao Xie**

University of Science and Technology of China, Hefei, China
{zuangao, qqqyd, cyril, wzx99, xujj1998}@mail.ustc.edu.cn, {wangyx58, htxie}@ustc.edu.cn

Abstract

In text recognition, self-supervised pre-training emerges as a good solution to reduce dependence on expansive annotated real data. Previous studies primarily focus on local visual representation by leveraging mask image modeling or sequence contrastive learning. However, they omit modeling the linguistic information in text images, which is crucial for recognizing text. To simultaneously capture local character features and linguistic information in visual space, we propose Symmetric Superimposition Modeling (SSM). The objective of SSM is to reconstruct the direction-specific pixel and feature signals from the symmetrically superimposed input. Specifically, we add the original image with its inverted views to create the symmetrically superimposed inputs. At the pixel level, we reconstruct the original and inverted images to capture character shapes and texture-level linguistic context. At the feature level, we reconstruct the feature of the same original image and inverted image with different augmentations to model the semantic-level linguistic context and the local character discrimination. In our design, we disrupt the character shape and linguistic rules. Consequently, the dual-level reconstruction facilitates understanding character shapes and linguistic information from the perspective of visual texture and feature semantics. Experiments on various text recognition benchmarks demonstrate the effectiveness and generality of SSM, with 4.1% average performance gains and 86.6% new state-of-the-art average word accuracy on Union14M benchmarks. The code is available at https://github.com/FaltingsA/SSM.

1 Introduction

Reading text from images is a fundamental and valuable task in computer vision with practical applications such as multimodal analysis, visual search, self-driving cars, and more. Since labeled real text images are scarce and expensive, various self-supervised text recognition methods have been utilized to exploit the intrinsic knowledge of unlabeled real data to alleviate the data scarcity issues. These self-supervised text recognition methods can be categorized into two main types: 1) Sequence Contrastive Learning (SCL), and 2) Mask Image Modeling (MIM). Benefiting from representation learning on unlabeled data, these methods have effectively enhanced Scene Text Recognition (STR) performance. However, these methods face the challenge of achieving linguistic learning, which is proven essential for text recognition (e.g. ABINet [Fang *et al.*, 2021] , LPV [Zhang *et al.*, 2023]).

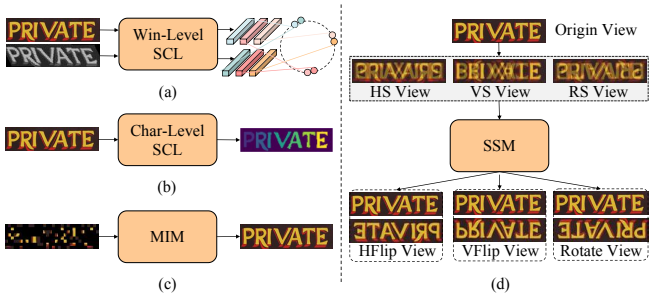

Figure 1: The comparison with mainstream self-supervised text recognition methods and our SSM. Win-Level SCL and Char-Level SCL represent window-level and character-level sequence contrastive learning, respectively. Rotate, VFlip and HFlip Views represent the symmetrically augmented image created through 180-degree rotation, vertical flipping, and horizontal flipping. HS, VS, and RS views respectively represent images formed by superimposing HFlip, VFlip, and Rotate View with the Origin View.

For SCL methods, SeqCLR [Aberdam *et al.*, 2021] and CCD [Guan *et al.*, 2023] are representative works. As shown in Fig. 1(a), SeqCLR ensures the local representation consistency between the same instance window across the two augmented views. Fig. 1(b) shows that CCD [Guan *et al.*, 2023] further ensures the character-level representation consistency based on the self-supervised segmentation. Hence, both of the two methods essentially focus on performing discriminative consistency learning on local character representations.

For MIM-like methods, MAERec [Jiang *et al.*, 2023] has attracted considerable attention in self-supervised text recognition. As discussed in MAERec, MIM-like methods essentially forces the model to infer the whole character from a few smallest parts of a character, due to covering a large portion of the text image. However, masking 75% image patches drops

*Corresponding Author

nearly all text foreground areas, as shown in Fig. 1(c). Thus, MIM-like methods also focus on local character features but overlook linguistic information (spelling rules between characters) in the text image.

Based on the analysis above, we can derive an observation that the previous self-supervised STR methods mainly focus on learning robust visual features of characters, overlooking the linguistic relationship between characters. Hence, it is meaningful to simultaneously capture character features and the implicit linguistic information in visual space.

To this end, we propose a novel self-supervised learning paradigm, named **S**ymmetric **S**uperimposition **M**odeling (SSM). The pretext task of SSM is to reconstruct the direction-specific pixel and feature signals from the symmetrically superimposed input. We adopt a Siamese network with an online branch and target branch to implement SSM. Specifically, we first construct inverted images by randomly selecting from three inversion enhancement techniques: horizontal flip, vertical flip, and 180-degree rotation. Then we superimpose it onto the origin image to create the symmetric superimposed input. For pixel reconstruction, we directly recover the original and inverted images (with online branch), as shown in Fig. 1(d). The original and inverted images are the pixel targets to guide the decoupling of the superimposed input. For feature reconstruction, we utilized the target branch to decouple the same symmetric superimposed input with irregular views, creating the original and inverted target feature on the fly. Subsequently, the online branch reconstruction the original and inverted target feature of the irregular view at the semantic level. We jointly use discriminative consistency loss and dense reconstruction loss to supervise the feature reconstruction process. Compared to MIM-like methods, we do not mask any patches and skillfully use symmetric superposition to disrupt character shapes and linguistic rules. Consequently, the pretext task of pixel and feature reconstruction from the superposed input can facilitate the learning of character shape features and linguistic information from the perspective of visual texture and feature semantics. In summary, our contributions mainly include:

- We propose a novel pre-training framework based on Symmetric Superimposition Modeling, which is the first self-supervised STR method dedicated to linguistic learning in visual space.
- We present a dual architecture for the joint reconstruction at both the pixel and feature level. This design enables joint learning of character visual features and implicit linguistic information from the texture-level and semantic level, further improving representation quality.
- Experiments demonstrate that SSM achieves state-of-the-art performance on various text recognition benchmarks. The 4.1% average performance gains on various STR methods also highlight the SSM's generality. Additionally, compared to other self-supervised methods, SSM has a 15.5% and 1.5% performance gain in multilingual text recognition with individual training and joint training settings, respectively.

2 Related Work

2.1 Text Recognition

Scene Text Recognition (STR) methods can be summarized into language-free and language-aware methods. Language-free methods [Shi *et al.*, 2017; Wang and Hu, 2017; Du *et al.*, 2022] treat STR as a character classification task, focusing on how to extract robust visual features. Language-aware methods can leverage linguistic information to improve robustness. The attention-based methods [Yu *et al.*, 2020; Litman *et al.*, 2020; Lee *et al.*, 2020; Sheng *et al.*, 2019] use various attention mechanisms to implicitly model linguistic rules. LISTER [Cheng *et al.*, 2023a] designs neighbor attention decoding for long text recognition. MGP-STR [Wang *et al.*, 2022] and PARSeq [Bautista and Atienza, 2022] focus on learning an internal Language Model(LM) during visual predicting. Some alternative approaches like ABINet [Fang *et al.*, 2021] and LevenOCR [Da *et al.*, 2022] propose to refine the visual predictions via an external LM.

2.2 Self-Supervised Text Recognition

Self-supervised learning technologies have been widely used for various works such as ML-LMCL [Cheng *et al.*, 2023b], TKDF [Cheng *et al.*, 2023c] and MESM [Liu *et al.*, 2024]. The most representative paradigms for computer vision are Contrastive Learning [Chen *et al.*, 2021] and Masked Image Modeling (MIM) [He *et al.*, 2022; Bao *et al.*, 2022; Chen *et al.*, 2023]. Recently, these self-supervised methods have been adopted for text recognition. SeqCLR [Aberdam *et al.*, 2021] is the pioneering work that proposes to model Contrast Learning on high-level sequence features for the first time. PerSec [Liu *et al.*, 2022] further performs Contrast Learning(CL) on both stroke-level features in shallow layers and Semantic-level features in deep layers. Later, DiG [Yang *et al.*, 2022] proposes to learn discriminative and generative features by integrating the CL and MIM. To explicitly focus on character structures and ensure sequence consistency, Guan proposes a character-level self-distillation framework [Guan *et al.*, 2023] based on unsupervised text segmentation maps. Recently, Union14M-U[Jiang *et al.*, 2023], a 10M scale unlabeled real data has been presented together with MAERec-S, which uses MAE for self-supervised text recognition. In addition, MaskOCR [Lyu *et al.*, 2022] and DARLING [Zhang *et al.*, 2024] use synthetic data for text recognition pre-training.

3 Methodology

The architecture of our proposed SSM is shown in Fig. 2, which comprises a Symmetric Superimposed Input Constructor, an online branch (blue arrow), and a target branch (orange arrow). We use the superimposed image of the original image and the horizontally flipped image as the input case to illustrate the whole workflow of our SSM.

3.1 Symmetrically Superimposed Input

Symmetrically Superimposed Input Constructor To ensure a large character overlapping area of the superimposed inputs, we considered three symmetrical augmenting operations: Horizontal Flipping $\mathbf{Hf}(\cdot)$, Vertical Flipping $\mathbf{Vf}(\cdot)$, and

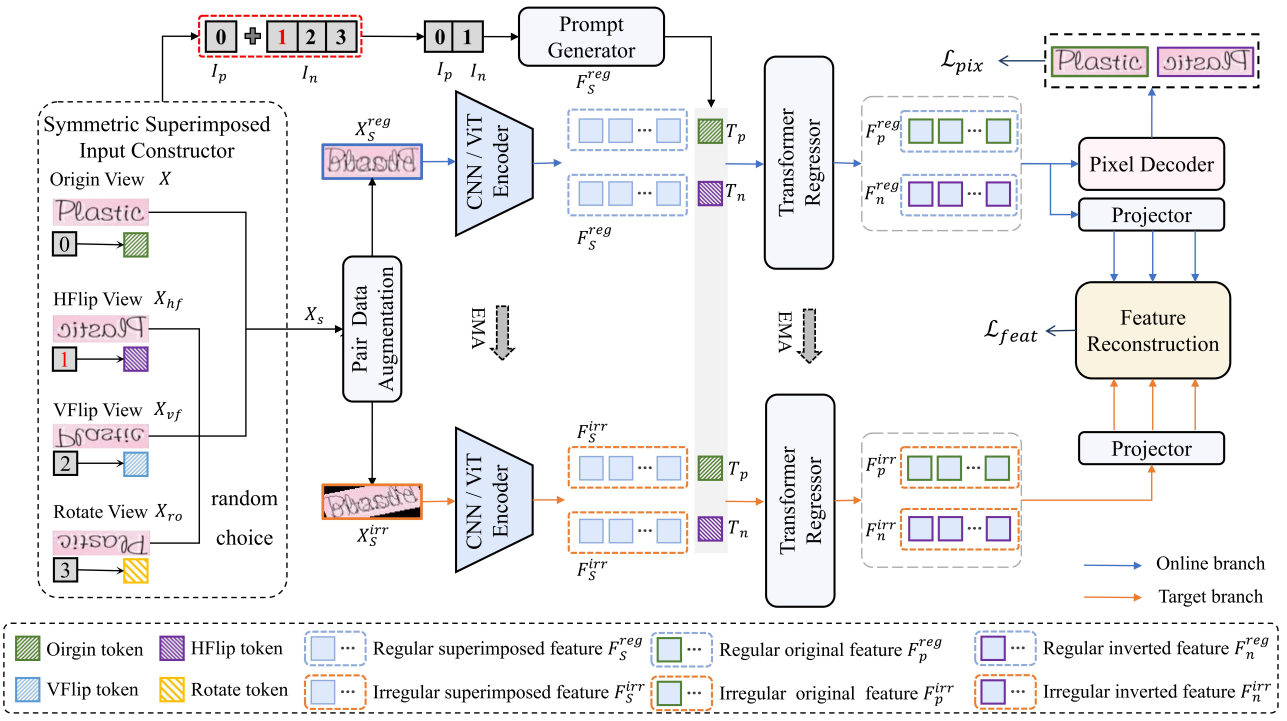

Figure 2: The pre-training framework of SSM. The blue arrow and green arrow stand for the workflow of the online branch and target branch respectively. Origin View: original image, HFlip View: horizontally flipped image, VFlip View: vertically flipped image, Rotate View: 180-degree rotated image. T_P and T_n correspond to the original and the reversed text direction, respectively.

180-degree rotation **Ro**(·). For the input image X, we randomly select one of **Hf**(·), **Vf**(·), and **Ro**(·) to obtain the corresponding inverted image X_R, where $X_R \in \{X_{hf}, X_{vf}, X_{ro}\}$. For the original image X, the original direction index I_p is fixedly set to 0. For the inverted image X_R, the inverted direction index I_n is selected from $\{1, 2, 3\}$ according to the type of inverted image, where (1, 2, 3) is assigned to (X_{hf}, X_{vf}, X_{ro}), respectively. These indexes are subsequently encoded to guide direction-specific reconstruction. Finally, we get the symmetrically superimposed input X_S by superimposing the original image X and its inverted images X_R. In the case of Fig. 2, X_{hf} is selected and the I_n is set to 1 (red color).

Data Augmentation The superimposed image X_S undergoes weak augmentations (*e.g.*, gaussian blur, and grayscale conversion) to create a regular view X_S^{reg} as the final input of online branch. The X and X_R are also transformed to X^{reg} and X_R^{reg} to supervise the image reconstruction of the online branch. For the target branch, we utilize the combination of both weak augmentations and geometry-based augmentations (*e.g.*, affine transformation and perspective warping) to generate a pair of irregular views: X_S^{irr}, X^{irr} and X_R^{irr}.

3.2 Symmetric Superimposition Modeling

1) Pixel-level Image Reconstruction We employ an Encoder-Regressor-Decoder architecture to perform the image reconstruction according to the specific direction index pair in the online branch. Specifically, we first leverage the ViT Encoder $\mathcal{F}(\cdot)$ to map the X_S^{reg} as latent feature $F_s \in \mathbb{R}^{\frac{HW}{p^2} \times d}$, where the p is the patch size and the d is the embedding dim. Meanwhile, we utilize the Prompt Generator $\mathcal{G}(\cdot)$ to encode the direction index pair (I_p, I_n) into the direction prompt token pair (T_p, T_n) with the same dimension of the ViT Encoder.

$$\begin{aligned} T_p &= FFN(Embed((I_p)) \\ T_n &= FFN(Embed((I_n)), \end{aligned} \tag{1}$$

where the Prompt Generator $\mathcal{G}(\cdot)$ consists of an embedding layer, two-layer FFN with normalization.

After that, T_p and T_n are each concatenated with the latent feature of superimposed input F_s and then sent into the Transformer Regressor $\mathcal{R}(\cdot)$ for feature decoupling in symmetric directions. The Transformer Regressor $\mathcal{R}(\cdot)$ is a series of vision transformer blocks. Thanks to the global interaction capabilities of the attention mechanism, Regressor $\mathcal{R}(\cdot)$ can extract the original direction feature F_p and the feature F_n corresponding to symmetrically reversed direction from the mixed features F_s. The process can be formulated as follows:

$$\begin{aligned} F_p &= \mathcal{R}([T_p, \mathcal{F}(X_S^{reg})]) \in \mathbb{R}^{\frac{HW}{p^2} \times d} \\ F_n &= \mathcal{R}([T_n, \mathcal{F}(X_S^{reg})]) \in \mathbb{R}^{\frac{HW}{p^2} \times d} \end{aligned} \tag{2}$$

Next, we use a lightweight pixel decoder to predict the RGB pixels of the corresponding views X^{reg} and X_R^{reg} from latent features F_p and F_n. To prevent the pixel decoder's parameters from dominating the learning process, we form the pixel decoder using only two linear layers with GELU. Finally, the L_2 loss function is adopted to optimize the image reconstruction process, and the complete loss function of

pixel-level image reconstruction can be defined as follows:

$$L_{pix} = \frac{1}{N}\sum_{i=1}^{N}(\left\|y_p^i - \hat{y}_p^i\right\|^2 + \left\|y_n^i - \hat{y}_n^i\right\|^2) \tag{3}$$

where $y_p^i, \hat{y}_p^i, \in R^3$ and $y_n^i, \hat{y}_n^i \in R^3$ are the prediction-target pair of x_p^{reg} and x_n^{reg}, respectively; N is the number of pixels.

2) Feature-level Representation Reconstruction To enhance the feature discriminability of character semantics and to model the spatial context between character visual semantics, we apply feature reconstruction in high-dimensional space. Specifically, we add the projector $\mathcal{H}(\cdot)$ followed by the Regressor $\mathcal{R}(\cdot)$ in the online branch. $\mathcal{H}(\cdot)$ aggregates the decoupled features (F_p and F_n) into window-level features by adaptive mean-pooling and then maps them into high-level feature space, forming the corresponding prediction feature queries (Q_p and Q_n). To construct the target representation, we introduce a target branch with the same structure as the online branch, except for the pixel decoder. X_S^{irr} is fed into the target branch to obtain the window-level target feature keys (K_p and K_n) on the fly. Next, we jointly employ the discriminative consistency loss L_{dis} and dense reconstruction loss L_{den} to supervise the reconstruction from Q_p to K_p and Q_n to K_n across the regular and irregular views. The loss L_{dis} is to enhance the character classification representation at the semantic level, which can be formulated as:

$$\begin{aligned} L_{dis} = &- log\frac{exp(Q_p \cdot K_p^+/\tau)}{exp(Q_p \cdot K_p^+/\tau) + \sum_{K_p^-} exp(Q_p \cdot K_p^-/\tau)} \\ &- log\frac{exp(Q_n \cdot K_n^+/\tau)}{exp(Q_n \cdot K_n^+/\tau) + \sum_{K_n^-} exp(Q_n \cdot K_n^-/\tau)} \end{aligned} \tag{4}$$

where K^+ stands for the matched positive samples and K^- indicates the negative samples collected from the same batch for both (Q_p,K_p) and (Q_n,K_n). τ denotes the temperature.

Since SiameseMIM [Tao *et al.*, 2023] demonstrates that predicting dense representations helps improve sensitivity to global structures, we minimize the distance between the predicted and the target features by L_{mse} loss, aiming to achieve semantic-level context modeling. Assuming that predictions are $Q = \{q^i \in \mathbb{R}^d | i = 1, ..., \mathcal{N}\}$ and targets is $K = \{k^i \in \mathbb{R}^d | i = 1, ..., \mathcal{N}\}$, where $\mathcal{N}$ is the total number of feature instances in one batch. The dense reconstruction loss L_{den} (L_{mse}) between the original Q-K pairs (Q_p and K_p) as well as inverted Q-K pairs (Q_n and K_n) can be formulated as:

$$L_{den} = \frac{1}{\mathcal{N}}\sum_{i=1}^{\mathcal{N}}(\left\|q_p^i - k_p^i\right\|^2 + \left\|q_n^i - k_n^i\right\|^2) \tag{5}$$

Finally, the optimization objectives of SSM are as follows:

$$L = L_{pix} + \alpha \times (\overbrace{L_{dis} + L_{den}}^{L_{feat}}) \tag{6}$$

where α is a scaling factor, the training objective of semantic-level feature reconstruction L_{feat} is the sum of L_{dis} and L_{den}.

3.3 Downstream Tasks

For text recognition downstream tasks, we add a text decoder after the ViT Encoder inherited from SSM. The text decoder consists of 6 transformer blocks and a linear prediction layer with 96 channels to predict the final characters. Besides, the text super-resolution and text segmentation results are presented in Appendix B.1 and B.2.

Figure 3: Reconstruction Visualization. **GT**: the original image. **HS/ VS/ RS**: horizontal/ vertical / roteated superimposed input. **Pre.** indicates the pixel prediction of **GT**. **GT-H/ V/ R**: the inverted view of the **GT** (HFlip, VFlip, 180-degree rotation view, respectively). **Pre.-H/ V/ R**:the pixel prediction of **GT-H/ V/ R**.

4 Experiments

4.1 Datasets

Unlabeled Pre-training Data We utilize the latest unlabeled real-scene dataset Union14M-U for self-supervised learning, which contains 10 million instances collected from Book32, OCR-CC and OpenVINO. Besides, we also conduct pre-training on the complete OCR-CC dataset(15.77M unlabeled text images) to facilitate a fair comparison with works such as CCD [Guan *et al.*, 2023] and DiG [Yang *et al.*, 2022].

Text Recognition Fine-tuning Data We use three types of labeled data. 1) STD: The synthetic data, comprising 14M images from MJSynth [Jaderberg *et al.*, 2014] and SynthText [Gupta *et al.*, 2016]. 2) ARD: 2.78M annotated real data used by DiG and CCD. 3) Union14M-L: 3.6M real labeled data.

Scene Text Recognition Benchmarks The Common benchmarks include IC13, SVT, IIIT5K, IC15, SVTP and CUTE80. Considering the saturation issue with existing benchmarks, we leverage Union14M benchmarks [2023] to evaluate. Some other challenging data such as TT, CTW, ArT, Uber, and COCO are also utilized to test.

Multilingual Text Recognition Benchmarks. Following MRN [Zheng *et al.*, 2023], we tested the multilingual generalization capability of SSM on MLT19 [Nayef *et al.*, 2019].

4.2 Implementation Details

Self-supervised Pre-training The pre-training is conducted on ViT, with image resolution of 32×128, an AdamW optimizer, *cosine* learning rate scheduler with a learning rate of 5e-4, batch size with 1,024, a weight decay of 0.05, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and warm-up for 1 epoch in a total of 20 epochs.

Text Recognition Fine-Tuning Our text recognition network is fine-tuned with STD or ARD or Union14M-L dataset. Patch size is 4×4. The text decoder consists of a 6-layer transformer block with an embedding dimension of 384. The batch size is 384 and the warm-up time is 1 epoch. The AdamW optimizer and a *OneCycle* learning rate scheduler with a learning rate of 1e-4 are employed.

Method	Data	IIIT	SVT	IC13	IC15	SVTP	CUTE	Avg.	Params.
SeqCLR [Aberdam *et al.*, 2021]	STD	82.9	-	87.9	-	-	-	-	-
SimAN [Luo *et al.*, 2022]	STD	87.5	-	89.9	-	-	-	-	-
PerSec-ViT [Liu *et al.*, 2022]	STD	88.1	86.8	94.2	73.6	77.7	72.7	83.8	-
DiG-ViT-Tiny [Yang *et al.*, 2022]	STD	95.8	92.9	96.4	84.8	87.4	86.1	91.8	20M
CCD-ViT-Tiny [Guan *et al.*, 2023]	STD	96.5	93.4	96.3	85.2	89.8	89.2	92.6	20M
SSM-ViT-Tiny	STD	96.5 ↑ 0.0	94.4 ↑ 1.0	96.3 ↑ 0.0	85.6 ↑ 0.4	89.3 ↓ 0.5	89.9 ↑ 0.7	92.8 ↑ 0.2	20M
DiG-ViT-Small [Yang *et al.*, 2022]	STD	96.7	93.4	97.1	87.1	90.1	88.5	93.2	36M
CCD-ViT-Small [Guan *et al.*, 2023]	STD	96.8	94.4	96.6	87.3	91.3	92.4	93.6	36M
SSM-ViT-Small	STD	97.4 ↑ 0.6	94.6 ↑ 0.2	96.7 ↑ 0.1	86.8 ↓ 0.5	91.3 ↑ 0.0	94.8 ↑ 2.4	93.8 ↑ 0.2	36M
DiG-ViT-Tiny [Yang *et al.*, 2022]	ARD	96.4	94.4	96.2	87.4	90.2	94.1	93.4	20M
CCD-ViT-Tiny [Guan *et al.*, 2023]	ARD	97.1	96.0	97.5	87.5	91.6	95.8	94.2	20M
SSM-ViT-Tiny	ARD	98.1 ↑ 1.0	96.1 ↑ 0.1	97.8 ↑ 0.3	89.0 ↑ 1.5	92.6 ↑ 1.0	96.5 ↑ 0.7	95.1 ↑ 0.9	20M
DiG-ViT-Small [Yang *et al.*, 2022]	ARD	97.7	96.1	97.3	88.6	91.6	96.2	94.7	36M
CCD-ViT-Small [Guan *et al.*, 2023]	ARD	98.0	96.4	98.3	90.3	92.7	98.3	95.6	36M
SSM-ViT-Small	ARD	98.9 ↑ 0.9	98.0 ↑ 1.6	98.5 ↑ 0.2	90.8 ↑ 0.5	95.0 ↑ 2.3	98.3 ↑ 0.0	96.4 ↑ 0.8	36M

Table 1: Text recognition results compared to other self-supervised text recognizers. DiG, CCD, and SSM are all pre-trained on the OCR-CC.

Method	Union14M Benchmarks								Other Challenge Datasets					
	Artistic	Contextless	Curve	General	Multi-Oritented	Multi-Words	Salient	Avg	TT	CTW	COCO	ArT	Uber	Avg
Scratch-ViT-S	72.7	81.0	82.6	82.6	78.1	81.7	79.5	79.8	90.5	86.0	75.5	81.5	82.4	83.2
MAE-ViT-S⋆ [He *et al.*, 2022]	76.0	81.6	86.2	82.8	84.0	83.5	83.3	82.5	91.7	87.8	76.6	82.8	83.8	84.6
DiG-ViT-S⋆ [Yang *et al.*, 2022]	77.4	82.5	85.9	83.8	83.5	84.0	84.3	83.0	91.7	87.2	77.7	83.4	84.9	85.0
SSM-ViT-S	**78.4**	**84.7**	**87.5**	**84.0**	**85.8**	**84.6**	**85.2**	**84.3**	**92.9**	**88.2**	**78.1**	**83.4**	**86.5**	**85.8**
↑	(+5.7)	(+3.7)	(+4.9)	(+1.4)	(+7.7)	(+2.9)	(+5.7)	(+4.5)	(+2.4)	(+2.2)	(+2.6)	(+1.9)	(+4.1)	(+2.6)
△	(+1.0)	(+2.2)	(+1.6)	(+0.2)	(+2.3)	(+0.6)	(+0.9)	(+1.3)	(+1.2)	(+1.0)	(+0.4)	(+0.0)	(+0.6)	(+0.8)

Table 2: Comparison with other self-supervised methods on Union14M benchmarks and other challenge datasets. ↑ and △ represent the performance gains relative to the "train from scratch" model and the second-best model, respectively. ⋆ stands for our implementation.

4.3 Comparasions With Self-Supervised Methods

Evaluation on Common benchmarks. Tab. 1 shows that SSM-ViT-Tiny outperforms the SeqCLR by 13.6% and 9.4% on IIIT and IC13, as well as outperforms PerSec-ViT (using 100M private data for pre-training) by 9% on average accuracy. Besides we compare with previous state-of-art self-supervised methods DiG and CCD, using the same pre-training data, fine-tuning data (ARD), and network architectures (i.e., Tiny, Small). Tab. 1 illustrates that our methods achieve a new state-of-the-art average word accuracy on common benchmarks with 95.1% and 96.4% at the same model size. When fine-tuning with ARD, our methods achieve average performance gains of 0.9% and 0.8%. SSM-ViT-small even beats the CCD-ViT-Base which has a larger model size (96.4% vs 96.3%). The CCD attains high performance with explicit segmentation guidance, while our straightforward method surpasses it without needing extensive segmentation GT label preparation. These results demonstrate the effectiveness of guiding text image self-supervised pre-training from a linguistic learning perspective.

Evaluation on Union14M benchmarks. Considering the saturation of the common benchmarks, we further compare SSM with other self-supervised methods on more challenging benchmarks in Tab. 2. Although we established a strong baseline model Scratch-ViT-Small, and SSM-ViT-Small still achieved an average performance gain of 3.5% and 2.6% on Union14M and other challenging data, respectively. Additionally, SSM surpassed MAE and DiG by 1.8% and 1.3% of average word accuracy on the Union14M benchmarks. The performance gains on all of Union14M benchmarks' subsets demonstrate that SSM effectively learns intrinsic feature representations for different fonts, text orientations, and complex scenes. Surprisingly, SSM shows superior performance for contextless text images, which may be attributed to the fact that embedding linguistic learning in visual space can avoid the text distribution dependence of explicit text correction.

Method	Arabic†	Korean	Japanese	Latin	Chinese	Bangla	AVG
Sctrach-S	2.3	7.2	11.2	79.4	1.6	6.1	17.2
MAE-S	48.3	34.8	21.4	83.4	5.3	43.3	40.4
DiG-S	48.6	37.9	23.5	83.6	4.7	50.3	41.4
SSM-S	**76.0**	**60.5**	**34.1**	**83.7**	**14.9**	**72.6**	**57.0**
△	(+27.4)	(+22.6)	(+10.6)	(+0.1)	(+10.2)	(+22.3)	(+15.5)

(a) Training each language data one by one.

Method	Arabic†	Korean	Japanese	Latin	Chinese	Bangla	AVG
Sctrach-S	70.6	55.2	36.6	80.4	23.0	69.5	55.9
MAE-S	72.1	63.5	42.7	82.9	30.1	72.8	60.7
DiG-S	75.2	64.5	42.5	82.9	29.5	76.1	61.9
SSM-S	**77.2**	**66.9**	**44.2**	**83.7**	**31.7**	**76.6**	**63.4**
△	(+2.4)	(+1.5)	(+1.7)	(+0.8)	(+2.2)	(+0.5)	(+1.5)

(b) Joint training of all language data.

Table 3: Comparison with other self-supervised methods on MLT19. † stands for the reading of Arabic is naturally right-to-left.

Multilingual Text Recognition Performance. To further

validate SSM's ability to learn linguistic information and its generality, we compare SSM with other self-supervised methods on MLT19 benchmarks. All methods are pre-trained on the Union14M-U for a fair comparison. In Tab. 3a, we finetune them on each language data one by one. Tab. 3a shows that our SSM outperforms the second-best method DiG by 15.5% in the average accuracy (57.0% v.s. 41.4%). Besides, for the right-to-left Arabic language, SSM even outperforms the second-best method by 27.4% (76.0% v.s. 48.6%), which is attributed to that SSM can capture linguistic information from right-to-left texts during the pre-training phase. These results demonstrate that SSM is an effective pretext task to capture linguistic information in multilingual texts and exhibits strong generalization and few-shot capabilities for data-scarcity languages. Tab. 3b shows that all the methods benefit from jointly learning linguistic information of different languages, yet SSM still exhibits superior performance with 1.5% average accuracy gains compared to DiG.

4.4 Ablations and Analysis

Effectiveness of architecture. The third row of Tab. 4 indicates that there is a 3.0% average performance improvement on Union14M benchmarks with the pixel-level reconstruction, which need the combined effect of prompt generator $\mathcal{G}(\cdot)$ transformer regressor $\mathcal{R}(\cdot)$ and pixel decoder $\mathcal{D}(\cdot)$. However, removing the regressor $\mathcal{R}(\cdot)$ results in a slight performance downgrade. The result demonstrates that the transformer regressor plays an indispensable role in SSM by guiding and disentangling the features of the original and inverted images. Based on the pixel reconstruction, the Joint use of the EMA mechanism and project module $\mathcal{H}(\cdot)$ further brings accuracy gains of 1.3% as it promotes learning discriminative character semantics and modeling spatial context at the semantic level. Here, $\mathcal{H}(\cdot)$ is mainly used to prevent excessive background noise in negative sample sampling and to map the decoupling feature to high-dimensional space. Lastly, the addition of Irregular view augmentations improves the average test accuracy up to 84.3% due to diverse perspectives.

Ablations on the types of superimposed input. We compared our superposition strategies with ***add noise*** & ***blur*** and randomly ***add another text image***. As shown in Tab. 5a, the performance gains of ***add another text image*** are close to that of ***add noise*** & ***blur***, both of which are lower than existing superposition strategies. We attribute this to the fact that the model can easily reconstruct the two images with inconsistent text content and different background styles from low-level pixel differences, limiting the exploration of intrinsic character semantics and linguistic information. We also found that individually superposing a 180-degree rotation view is more effective than individually superposing a specific flip view. The result suggests that the rotation influences character orientation and sequence reading order, helping capture learning richer character semantics and spatial context relationships during reconstruction. Finally, the best performance (84.3%) is achieved by combining HS, VS, and RS strategies, which improve the diversity of the superimposed input.

Ablations on the feature-level reconstruction loss. The first and second rows of Tab. 5b shows that only using distance measure functions such as *MSE* L_{mse} and *Cosine* L_{cos} results in lower word accuracy (82.5% vs 94.3%). We attribute this to the fact that only minimizing the feature distance between predictions and targets may lead to feature collapse. Besides, Tab. 5b shows that our discriminative consistency loss L_{dis} outperforms the dino loss L_{dino} by 0.3%. The combined use of L_{dis} and L_{mse} ultimately yielded the best results. L_{dis} can mitigate the feature collapse of L_{mse} by reducing the similarity between negative samples. Besides, L_{dis} focuses on enhancing discriminative features for character semantics while L_{mse} is more sensitive to spatial structures, which helps model linguistic information in spatial context at the semantic level.

$\mathcal{G}(\cdot)$	$\mathcal{R}(\cdot)$	$\mathcal{D}(\cdot)$	EMA	$\mathcal{H}(\cdot)$	Irr view	Avg
Scractch training						79.8
✓						79.7
✓	✓	✓				82.8
✓	✓	✓	✓			83.6
✓	✓	✓	✓	✓		84.1
✓	✓	✓	✓	✓	✓	**84.3**

Table 4: Ablations about the effectiveness of architecture. $\mathcal{G}(\cdot)$, $\mathcal{R}(\cdot)$, $\mathcal{D}(\cdot)$ and $\mathcal{H}(\cdot)$ means prompt generator, regressor, pixel decoder, and projector, respectively. Testing on Union14M benchmarks.

HS	VS	RS	Avg
add noise & blur			80.7
add another image			80.4
✓			83.1
	✓		82.9
		✓	83.4
✓	✓	✓	**84.3**

(a) Ablations on the types of superimposed input.

Reconstruction Loss	Avg
L_{mse}	82.8
L_{cos}	82.5
L_{dino}	83.4
L_{dis}	83.8
L_{dis}+L_{mse}	**84.3**

(b) Ablations on feature-level reconstruction loss.

Table 5: Ablation studies on the input type and consistency loss. HS, VS, and RS respectively means superimposing HFlip, VFlip, and Rotation View with the original image.

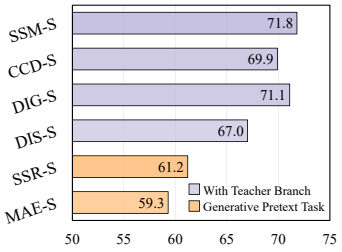

Figure 4: Comparison of feature representation evaluation.

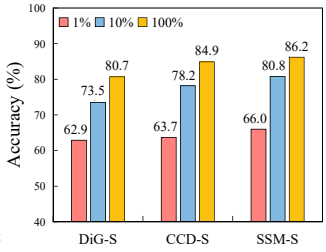

Figure 5: Fine-tuning on ARD with different ratios.

Feature representation evaluation. We freeze the encoder and train the decoder with ARD, following the test setting of DiG. Fig. 4 shows that the pixel-level reconstruction of SSM (SSR-S) has better feature representation than MAE-S (61.2% vs59.3%). By introducing feature-level representation reconstruction, SSM outperforms DiG by 0.7%. We attribute this to strengthening the implicit linguistic representation through texture-level and semantic-level reconstruction.

Fine-tuning with different data ratios. We fine-tune our

Type	Method	Common Benchmarks							Union14M-Benchmark								Params.
		IIIT 3000	IC13 1015	SVT 647	IC15 2077	SVTP 645	CUTE 288	Avg	Cur	M-O	Art	Con	Sal	M-W	Gen	Avg	
CTC	CRNN [Shi *et al.*, 2017]	90.8	91.8	83.8	71.8	70.4	80.9	81.6	19.4	4.5	34.2	44.0	16.7	35.7	60.4	30.7	8.3M
	SVTR [Du *et al.*, 2022]	95.9	95.5	92.4	83.9	85.7	93.1	91.1	72.4	68.2	54.1	68.0	71.4	67.7	77.0	68.4	24.6M
Attention	ASTER[Shi *et al.*, 2019]	94.3	92.6	88.9	77.7	80.5	86.5	86.7	38.4	13.0	41.8	52.9	31.9	49.8	66.7	42.1	-
	NRTR[Sheng *et al.*, 2019]	96.2	96.9	94.0	80.9	84.8	92.0	90.8	49.3	40.6	54.3	69.6	42.9	75.5	75.2	58.2	-
	DAN[Wang *et al.*, 2020]	95.5	95.2	88.6	78.3	79.9	86.1	87.3	46.0	22.8	49.3	61.6	44.6	61.2	67.0	50.4	-
	SATRN[Lee *et al.*, 2020]	97.0	97.9	95.2	87.1	91.0	96.2	93.9	74.8	64.7	67.1	76.1	72.2	74.1	75.8	72.1	55M
	RobustScanner[Yue *et al.*, 2020]	96.8	95.7	92.4	86.4	83.9	93.8	91.2	66.2	54.2	61.4	72.7	60.1	74.2	75.7	66.4	-
LM	SRN[Yu *et al.*, 2020]	95.5	94.7	89.5	79.1	83.9	91.3	89.0	49.7	20.0	50.7	61.0	43.9	51.5	62.7	48.5	55M
	ABINet[Fang *et al.*, 2021]	97.2	97.2	95.7	87.6	92.1	94.4	94.0	75.0	61.5	65.3	71.1	72.9	59.1	79.4	69.2	37M
	VisonLAN[Wang *et al.*, 2021]	96.3	95.1	91.3	83.6	85.4	92.4	91.3	70.7	57.2	56.7	63.8	67.6	47.3	74.2	62.5	33M
	MATRN[Na *et al.*, 2022]	98.2	97.9	96.9	88.2	94.1	97.9	95.5	80.5	64.7	71.1	74.8	79.4	67.6	77.9	74.6	44M
	PARSeq⋆[Bautista and Atienza, 2022]	98.0	96.8	95.2	85.2	90.5	96.5	93.8	79.8	79.2	67.4	77.4	77.0	76.9	80.6	76.9	24M
Pre-train	MAERec-S[Jiang *et al.*, 2023]	98.0	97.6	96.8	87.1	93.2	97.9	95.1	81.4	71.4	72.0	82.0	78.5	82.4	82.5	78.6	36M
	DiG-ViT-Small⋆[Yang *et al.*, 2022]	98.7	97.8	98.5	88.9	92.7	96.5	95.5	85.9	83.5	77.4	82.5	84.3	84.0	83.8	83.0	36M
	MAERec-B[Jiang *et al.*, 2023]	98.5	98.1	97.8	89.5	94.4	98.6	96.2	88.8	83.9	80.0	85.5	84.9	87.5	**85.8**	85.2	142M
Ours	SSM-ViT-Tiny	98.8	97.9	96.8	88.0	93.2	97.2	95.3	81.7	77.9	72.3	79.7	79.7	77.9	81.9	80.7	20M
	SSM-ViT-Small	99.0	98.3	97.8	89.5	94.0	98.3	96.1	87.5	85.8	78.4	84.8	85.2	85.0	84.0	84.3	36M
	SSM-ViT-Small-Turbo	**99.1**	**98.5**	97.7	**89.9**	**94.9**	**99.0**	**96.5**	**91.0**	**89.4**	79.3	**86.0**	**86.5**	**88.6**	85.3	**86.6**	36M

Table 6: Performance of models fine-tuned on the **Union14M-L**. All pre-train and our methods are pre-trained on Union14M-U. **Bold** and underlined values stands for the 1st and 2nd results in each column. For a fair comparison, IC13 and IC15 are larger versions. Cur, M-O, Art, Ctl, Sal, M-W, and Gen respectively represent Curve, Multi-Oriented, Artistic, Contextless, Salient, Multi-Words, and General. Avg stands for the average word accuracy of corresponding benchmarks. ⋆ represents our implementation. '*-Turbo' stands for training 20 epochs.

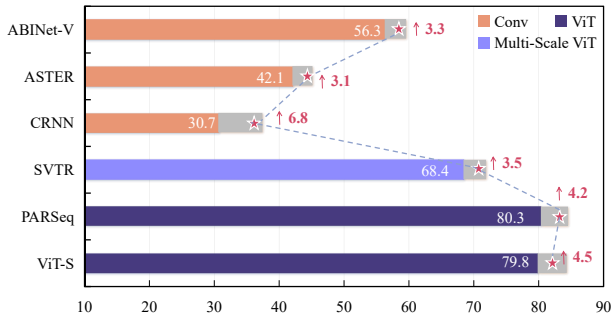

Figure 6: Performance gains of SSM for various STR Methods.

Figure 7: Qualitative recognition results of challenging scenarios.

method with 1%(27.8K), 10%(278K), and 100%(2.78M) of ARD. Fig. 5 shows that SSM-ViT-Small outperforms the previous SOTA methods by 2.3%, 2.6% and 1.3%, respectively.

Improvement of Various STR methods. Benefiting from masking-free operations, our method is not troubled by information leakage during feature downsampling. Consequently, it can be flexibly applied to various types of STR Methods, including Conv-based methods (e.g., ABINet-V, ASTER, and CRNN), multi-scale attention methods (e.g., SVTR), and vanilla ViT methods (e.g., our baseline ViT-*, and PARSeq). Fig. 6 shows that SSM can bring a performance gain of 4.1% to the above STR method on Union14M benchmarks.

Qualitative recognition results. The top and bottom strings of Fig. 7 are predicted by DiG-S and SSM-S, with red indicating errors. Results show that SSM is robust to reversed text and complex-textured images. It also can infer complex characters based on contextual linguistic information.

4.5 Comparisons With State-of-the-Arts

In Tab. 6, we compare SSM and previous SOTA text recognition methods. Specifically, SSM-ViT-Tiny outperforms all supervised state-of-the-art (SOTA) methods (80.7% vs 76.9%) on more challenging Union14M benchmarks with only 20M parameters. Moreover, SSM-ViT-Small outperforms the other self-supervised pre-train methods with the same model size by 0.6% and 1.3% average performance. Surprisingly, by training 20 epochs, SSM-ViT-Small-Turbo pushes the new SOTA average performance on Common benchmarks and Union14M benchmarks to 96.5% and 86.6%, respectively. Note that SSM-ViT-Small-Turbo outperforms MAERec-B by 0.3% and 1.4% with only one-fourth of the parameters of the MAERec-B (36M vs 142M).

5 Conclusion

In this paper, we propose a novel self-supervised text recognition framework, termed SSM. In contrast to previous self-supervised text recognition methods that only focus on local visual features, SSM models the global spatial context for implicit linguistic learning. SSM aims to reconstruct direction-specific pixel and feature signals from symmetrically superimposed text images. In this way, SSM captures both the local character feature and implicit linguistic information. Eventually, SSM improves the text recognition performance of various Scene Text Recognition methods and refreshes state-of-the-art performance on various text recognition benchmarks.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2022YFB3104700), the National Nature Science Foundation of China (62121002, U23B2028, 62102384). This research is supported by the Supercomputing Center of the USTC. We also acknowledge the GPU resource support offered by the MCC Lab of Information Science and Technology Institution, USTC.

References

[Aberdam *et al.*, 2021] Aviad Aberdam, Ron Litman, Shahar Tsiper, Oron Anschel, Ron Slossberg, Shai Mazor, R. Manmatha, and Pietro Perona. Sequence-to-sequence contrastive learning for text recognition. In *CVPR*, pages 15302–15312, 2021.

[Bao *et al.*, 2022] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

[Bautista and Atienza, 2022] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, pages 178–196. Springer, 2022.

[Chen *et al.*, 2021] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9620–9629, 2021.

[Chen *et al.*, 2023] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision*, pages 1–16, 2023.

[Cheng *et al.*, 2023a] Changxu Cheng, Peng Wang, Cheng Da, Qi Zheng, and Cong Yao. Lister: Neighbor decoding for length-insensitive scene text recognition. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[Cheng *et al.*, 2023b] Xuxin Cheng, Bowen Cao, Qichen Ye, Zhihong Zhu, Hongxiang Li, and Yuexian Zou. Ml-lmcl: Mutual learning and large-margin contrastive learning for improving asr robustness in spoken language understanding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6492–6505, 2023.

[Cheng *et al.*, 2023c] Xuxin Cheng, Zhihong Zhu, Wanshi Xu, Yaowei Li, Hongxiang Li, and Yuexian Zou. Accelerating multiple intent detection and slot filling via targeted knowledge distillation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[Da *et al.*, 2022] Cheng Da, Peng Wang, and Cong Yao. Levenshtein ocr. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, pages 322–338. Springer, 2022.

[Du *et al.*, 2022] Yongkun Du, Zhineng Chen, Caiyan Jia, Xiaoting Yin, Tianlun Zheng, Chenxia Li, Yuning Du, and Yu-Gang Jiang. Svtr: Scene text recognition with a single visual model. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 884–890. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.

[Fang *et al.*, 2021] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *CVPR*, pages 7098–7107, 2021.

[Guan *et al.*, 2023] Tongkun Guan, Wei Shen, Xue Yang, Qi Feng, Zekun Jiang, and Xiaokang Yang. Self-supervised character-to-character distillation for text recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19473–19484, October 2023.

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

[He *et al.*, 2022] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[Jaderberg *et al.*, 2014] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *CoRR*, abs/1406.2227, 2014.

[Jiang *et al.*, 2023] Qing Jiang, Jiapeng Wang, Dezhi Peng, Chongyu Liu, and Lianwen Jin. Revisiting scene text recognition: A data perspective. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023.

[Lee *et al.*, 2020] Junyeop Lee, Sungrae Park, Jeonghun Baek, Seong Joon Oh, Seonghyeon Kim, and Hwalsuk Lee. On recognizing texts of arbitrary shapes with 2d self-attention. pages 546–547, 2020.

[Litman *et al.*, 2020] Ron Litman, Oron Anschel, Shahar Tsiper, Roee Litman, Shai Mazor, and R Manmatha. Scatter: selective context attentional scene text recognizer. pages 11962–11972, 2020.

[Liu *et al.*, 2022] Hao Liu, Bin Wang, Zhimin Bao, Mobai Xue, Sheng Kang, Deqiang Jiang, Yinsong Liu, and Bo Ren. Perceiving stroke-semantic context: Hierarchical contrastive learning for robust scene text recognition. In *AAAI*, 2022.

[Liu *et al.*, 2024] Zhihang Liu, Jun Li, Hongtao Xie, Pandeng Li, Jiannan Ge, Sun-Ao Liu, and Guoqing Jin. Towards balanced alignment: Modal-enhanced semantic modeling for video moment retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 3855–3863, 2024.

[Luo *et al.*, 2022] Canjie Luo, Lianwen Jin, and Jingdong Chen. Siman: exploring self-supervised representation learning of scene text via similarity-aware normalization. pages 1039–1048, 2022.

[Lyu *et al.*, 2022] Pengyuan Lyu, Chengquan Zhang, Shanshan Liu, Meina Qiao, Yangliu Xu, Liang Wu, Kun Yao, Junyu Han, Errui Ding, and Jingdong Wang. Maskocr: Text recognition with masked encoder-decoder pretraining, 2022.

[Na *et al.*, 2022] Byeonghu Na, Yoonsik Kim, and Sungrae Park. Multi-modal text recognition networks: Interactive enhancements between visual and semantic features. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, pages 446–463. Springer, 2022.

[Nayef *et al.*, 2019] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1582–1587. IEEE, 2019.

[Sheng *et al.*, 2019] Fenfen Sheng, Zhineng Chen, and Bo Xu. Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition. In *2019 International conference on document analysis and recognition (ICDAR)*, pages 781–786. IEEE, 2019.

[Shi *et al.*, 2017] 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*, pages 2298–2304, 2017.

[Shi *et al.*, 2019] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. ASTER: an attentional scene text recognizer with flexible rectification. *TPAMI*, pages 2035–2048, 2019.

[Tao *et al.*, 2023] Chenxin Tao, Xizhou Zhu, Weijie Su, Gao Huang, Bin Li, Jie Zhou, Yu Qiao, Xiaogang Wang, and Jifeng Dai. Siamese image modeling for self-supervised vision representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2132–2141, 2023.

[Wang and Hu, 2017] Jianfeng Wang and Xiaolin Hu. Gated recurrent convolution neural network for ocr. *Advances in Neural Information Processing Systems*, 30, 2017.

[Wang *et al.*, 2020] Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. Decoupled attention network for text recognition. pages 12216–12224, 2020.

[Wang *et al.*, 2021] Yuxin Wang, Hongtao Xie, Shancheng Fang, Jing Wang, Shenggao Zhu, and Yongdong Zhang. From two to one: A new scene text recognizer with visual language modeling network. In *ICCV*, pages 14174–14183, 2021.

[Wang *et al.*, 2022] Peng Wang, Cheng Da, and Cong Yao. Multi-granularity prediction for scene text recognition. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, pages 339–355. Springer, 2022.

[Yang *et al.*, 2022] Mingkun Yang, Minghui Liao, Pu Lu, Jing Wang, Shenggao Zhu, Hualin Luo, Qi Tian, and Xiang Bai. Reading and writing: Discriminative and generative modeling for self-supervised text recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4214–4223, 2022.

[Yu *et al.*, 2020] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. pages 12110–12119, 2020.

[Yue *et al.*, 2020] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. Robustscanner: Dynamically enhancing positional clues for robust text recognition. pages 135–151, 2020.

[Zhang *et al.*, 2023] Boqiang Zhang, Hongtao Xie, Yuxin Wang, Jianjun Xu, and Yongdong Zhang. Linguistic more: Taking a further step toward efficient and accurate scene text recognition. *arXiv preprint arXiv:2305.05140*, 2023.

[Zhang *et al.*, 2024] Boqiang Zhang, Hongtao Xie, Zuan Gao, and Yuxin Wang. Choose what you need: Disentangled representation learning for scene text recognition, removal and editing. *arXiv preprint arXiv:2405.04377*, 2024.

[Zheng *et al.*, 2023] Tianlun Zheng, Zhineng Chen, BingChen Huang, Wei Zhang, and Yu-Gang Jiang. Mrn: Multiplexed routing network for incremental multilingual text recognition. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.