One2Seq: One-Token Wise Decoder for Efficient Scene Text Recognition

Zhibin Ma[1], **Pengwen Dai**[1*], **Wei Zhuo**[2], **Xugong Qin**[3]

[1]School of Cyber Science and Technology, Shenzhen Campus of Sun Yat-sen University
[2]National Engineering Laboratory of Big Data System Computing Technology, Shenzhen University
[3]School of Cyber Science and Engineering, Nanjing University of Science and Technology
mazhb3@mail2.sysu.edu.cn, daipw@mail.sysu.edu.cn, weizhuo@szu.edu.cn, qinxugong@njust.edu.cn

Abstract

Auto-regressive (AR)-based decoders, owing to their flexibility in handling variable-length outputs and their strong capability in modeling character-level dependencies, have emerged as the predominant decoding paradigm in the field of scene text recognition (STR). However, AR-based decoders suffer from attention drift, slow decoding speed, and difficulty capturing global dependencies, restricting their performance in various scenarios. In this paper, we propose a novel paradigm for AR-based decoding, called **One-Token to Sequence (One2Seq)**, to address the above issues. Unlike existing methods, we encode the semantic features into a single *context token* and design a *One-Token Wise Decoder* to perform the decoding, which alleviates the attention drift caused by the accumulation of semantic information. Moreover, we proposed *Positional-aware Hash Embedding* to embed the decoded characters, ensuring the order information is obtained in the *context token*. By continuously updating this token, One2Seq fully leverages the decoded semantic information while avoiding the computational overhead associated with the growing query sequence. Furthermore, to leverage global information for decoding, we propose *Dynamic Global Infusion* to dynamically integrate global visual features into the *context token*. Equipped with the enriched context token, the model has an enhanced ability to extract discriminative local features under the guidance of global context, thereby enhancing recognition accuracy. Extensive experiments reveal that, with its ingenious design, One2Seq exhibits marked superiority on both accuracy and decoding speed compared to existing STR models.

Code — https://github.com/xiaomaaa2002/One2Seq

Introduction

Scene text recognition (STR) aims to extract text from natural images. Its applications in autonomous driving (Zhang et al. 2022), product recognition (Wu, Bouyarmane, and Tutar 2023), robotic navigation (Posner, Corke, and Newman 2010), and other areas have made it a popular research topic in recent years.

Existing state-of-the-art STR methods typically adopt an auto-regressive (AR) decoding strategy (Shi et al. 2019; Yue et al. 2020; Jiang et al. 2023), where the prediction at each position depends on all previously generated outputs. Compared with parallel decoding (PD) methods (Yu et al. 2020; Fang et al. 2021; Du et al. 2025b), AR decoders are more effective at capturing contextual dependencies between characters, such as semantic associations and spelling structures. This allows them to achieve superior recognition performance, especially in tasks that require contextual reasoning and semantic completion, such as recognizing long texts (Du et al. 2025a) and occluded texts (Wang et al. 2021).

Existing AR decoders, however, encounter three major challenges: (1) **Attention drift.** Existing studies (Yue et al. 2020; Zheng et al. 2024) have shown that the performance of AR decoders can be affected by attention drift, where the

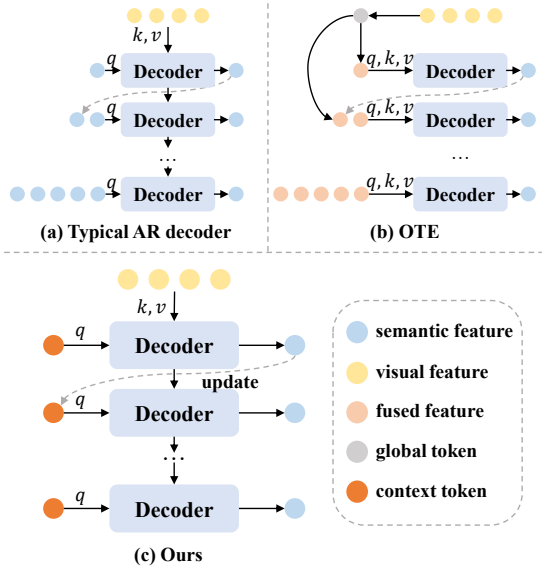


Figure 1: Comparisons of (a) A typical AR decoder. (b) OTE decoder. and (c) Our proposed **One-Token Wise Decoder**. In (a) and (b), the decoded tokens are appended back to the query, the length of the input sequence thus increases progressively with each decoding time step. Differently, in (c) we integrate the already decoded semantic tokens into a single token, ensuring that the number of the query token fed into the decoder remains **1** at all time steps.

*Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

attention mechanism is not accurately aligned with the position of the character. Previous research (Yue et al. 2020) attributes this phenomenon to the weakening of positional information caused by the accumulation of semantic information, and proposes a position enhancement branch to mitigate it. However, for challenging samples such as occluded texts or long texts, attention drift still persists, which limits the generalization capability of AR decoders. (2) **Slow decoding speed.** In conventional AR decoders, as shown in Figure 1(a), the length of the input sequence T increases linearly with the decoding time steps, which consequently leads to an $O(T^2)$ computational complexity in the self-attention mechanism (Vaswani et al. 2017). Although OTE (Xu et al. 2024) introduces a one-token based decoding strategy that eliminates the need for cross-attention computation, the number of tokens input to the self-attention layers still increases with decoding steps (Figure 1(b)). (3) **Insufficient global context.** The AR decoding scheme generates the target sequence one-by-one. During the early steps of decoding, the model has access to limited global information, leading to less reliable predictions at the beginning stages. Several approaches (Yu et al. 2020; Litman et al. 2020) have explored the explicit integration of global contextual cues during decoding, yielding notable improvements in performance. However, whether a single query vector can effectively capture global information to support decoding remains an open challenge.

To address the above issues, in this paper, we propose a novel decoding paradigm, called **One-Token to Sequence (One2Seq)**, for accurate and fast scene text recognition, as shown in Figure 1(c). First, we embed and aggregate all decoded characters into a *context token* and design a *One-Token Wise Decoder* (OTWD) to utilize this token for decoding. Distinct from conventional AR decoders, our OTWD takes only one token as the query input at each time step. With all decoded semantic information consistently encapsulated within this token, it alleviates the attention drift caused by the accumulation of semantic information. Besides, to ensure that positional information of characters is preserved within the *context token*, we design a *Positional-Aware Hash Embedding* to encode each character. Secondly, we observed that leveraging global image information facilitates a more comprehensive perception of local features. Therefore, we proposed *Dynamic Global Infusion* (DGI) to dynamically integrate global visual features into the context token. By DGI, the model can explicitly leverage global image information, which reduces the recognition errors in the early time steps of AR decoding caused by the lack of global context. Furthermore, with its one-token wise design, One2Seq avoids the progressive growth of the input sequence to the decoder over time, which is common in traditional AR decoding, and thus significantly reduces the computational cost during the decoding stage.

Our contributions are summarized as follows:

- We propose a novel decoding paradigm for STR, named **One2Seq**, which leverages *One-Token Wise Decoder* to perform AR decoding. With its ingenious design, One2Seq not only alleviates attention drift, but also avoids the rising computational cost in decoding.
- The *Dynamic Global Infusion* is proposed to dynamically fuse global visual feature into the *context token*, mitigating recognition errors arising from insufficient global context. Moreover, a practical *Positional-Aware Hash Embedding* is introduced to retain the order information of decoded characters.
- Extensive experiments prove that One2Seq surpasses existing STR methods in recognition accuracy, while simultaneously reducing decoding time. The visualization results further validates the effectiveness of this paradigm.

Related Works

CTC-based methods (Liu et al. 2016; Gao et al. 2019; Su and Lu 2014; Du et al. 2022) typically use Connectionist Temporal Classification (Graves et al. 2006) (CTC) as a decoding scheme. It models the output as an implicit alignment of the input features, handling variable-length sequences by inserting blank tokens, without requiring explicit alignment. Among them, CRNN (Shi, Bai, and Yao 2017; He et al. 2016) stacks RNNs on top of CNNs and uses CTC for decoding, while SVTR (Du et al. 2022) first divides the image into patches and employs mixing blocks to capture inter- and intra-character patterns. However, due to its inherent limitation of transforming the image into one-dimensional input, CTC-based methods cannot handle irregularly arranged text well, such as curved or multi-oriented text.

Encoder-Decoder methods typically employ an attention mechanism to decode the sequence. Their capacity to model irregular text and capture long-range dependencies has made them the de facto choice for high-performance STR systems. Among them, parallel decoding methods (Yu et al. 2020; Wang et al. 2021; Fang et al. 2021; Na, Kim, and Park 2022; Wang, Da, and Yao 2022; Du et al. 2025b) predict all characters in one shot. While offering faster inference, they tend to exhibit weaker contextual modeling and inaccurate sequence length estimation, which limits their effectiveness on complex scenarios.

To achieve more accurate predictions, existing methods typically adopt an auto-regressive (AR) decoding strategy. Early AR decoders were typically composed of RNNs with an attention mechanism (Shi et al. 2019; Luo, Jin, and Sun 2019; Wang et al. 2020; Zhang et al. 2020). They leverage the sequential memory capabilities of RNNs to enhance contextual modeling. However, due to the inherent limitations of RNNs in capturing long-range dependencies, their performance degrades when handling long text sequences. With the successful application of Transformers (Vaswani et al. 2017) in various vision tasks, researchers have introduced them into STR and adopted their decoders for decoding (Sheng, Chen, and Xu 2019; Yue et al. 2020; Jiang et al. 2023; Xu et al. 2024; Du et al. 2025a). Compared with RNN-based decoders, Transformer-based decoders are more capable of modeling complex linguistic structures and long-range dependencies. Furthermore, their inherent support for parallel computation facilitates more efficient training, making them a prevailing choice for AR decoders in recent scene text recognition research. However, AR decoders generate characters one by one, with the decoded semantic features

growing linearly over time, resulting in attention drift and slow inference. In addition, insufficient global context modeling remains a key challenge for AR decoders.

In this work, we aggregated the semantic features into a single token, and proposed *One-token Wise Decoder* for decoding. This paradigm preserves the advantages of AR decoding while avoiding the issues of attention drift and increased computational cost caused by the growing semantic feature sequence.

Proposed Method

As shown in Figure 2, our proposed One2Seq consists of a Visual Encoder (VE), a Dynamic Global Infusion (DGI), a Positional-aware Hash Embedding (PHE) and a One-Token Wise Decoder (OTWD). Given an input image x, the VE first extracts the visual feature F. Next, given the decoding time step i (initialize as 1), DGI takes F and the decoded character embeddings $e^{0:i-1}$ (e^0 denotes the start token) as input, dynamically fuse them into a *context token* F_{ct}^i. Finally, the OTWD takes F_{ct}^i and the visual features F as input, decodes the character y^i at the current time step i. After that, we embed the decoded characters $y^{0:i}$ via PHE, and update i to $i+1$ to perform the next step decoding. This process is repeated until the complete text with L characters $\mathbf{y} = (y^0, y^1, ..., y^L)$ is predicted.

Visual Encoder

The visual encoder aims to extract the visual feature F from the input image x. Given the backbone $\mathcal{E}(\cdot)$, the formulation of vision feature F is as follows:

$$F = \mathcal{E}(x) \in \mathbb{R}^{d \times h \times w}. \tag{1}$$

The pretrained CLIP visual encoder (Radford et al. 2021) has been shown to serve as an effective backbone for STR (Zhao et al. 2024). Thus, we choose the visual encoder of CLIP-ViT-B/16 as our visual encoder $\mathcal{E}(\cdot)$ and initialize it with pretrained weights. We also adopt SVTR (Du et al. 2022) of different scales (SVTR-B, SVTR-L) as lightweight backbones to explore the effectiveness of our method when combined with different scales of backbone.

Dynamic Global Infusion

After extracting the visual features F, we propose Dynamic Global Infusion (DGI) to integrate them with the semantic features generated during decoding, forming a unified *context token*. DGI contains a visual aggregation, a semantic aggregation, and a gated fusion mechanism.

Visual aggregation (VA) aims to aggregate the visual features F into a global visual token. After obtaining the visual features F, we apply global average pooling (GAP) to downsample F and obtain the visual token F_v:

$$F_v = \text{GAP}(F) \in \mathbb{R}^{d \times 1}. \tag{2}$$

The visual token F_v captures the global information shared among visual features, as validated in previous work (Xu et al. 2024).

Semantic aggregation (SA) is designed to extract a semantic token that encapsulates the semantic information of the previously decoded sequence. First, previously decoded characters $y^{0:i-1}$ are embedded into semantic features $e^{0:i-1}$ by *Positional-aware Hash Embedding* (details presented in the next section). Next, they are aggregated to obtain a semantic token:

$$F_s^i = \frac{1}{i} \sum_{j=0}^{i-1} e^j \in \mathbb{R}^{d \times 1}, \tag{3}$$

where e^j denotes the PHE of the j-th character, and i denotes the current time step. At $i = 1$, only the start token e^0 is embedded to initialize the semantic token F_s^1.

Gated Fusion (GF) is applied to dynamically combine F_v and F_s^i, enabling the model to retain salient features while suppressing irrelevant information. Given F_v and F_s^i, we fuse them as follows:

$$g^i = \sigma(W_g \cdot [F_s^i, F_v] + b_g) \in \mathbb{R}^{d \times 1}, \tag{4}$$

$$f^i = \tanh\left(W_f \cdot [F_s^i, F_v] + b_f\right) \in \mathbb{R}^{d \times 1}, \tag{5}$$

$$F_m^i = g^i \odot f^i + (1 - g^i) \odot F_v \in \mathbb{R}^{d \times 1}, \tag{6}$$

where $W_g, W_f \in \mathbb{R}^{2d \times d}$, $b_g, b_f \in \mathbb{R}^{d \times 1}$ are learnable parameters, $\sigma(\cdot)$ denotes the sigmoid function, and $\odot$ represents element-wise multiplication.

Finally, we add the positional embedding PE^i and obtain the *context token* F_{ct}^i:

$$F_{ct}^i = F_m^i + PE^i. \tag{7}$$

Positional-aware Hash Embedding

The commonly used embedding strategies, such as one-hot encoding or Word2Vec (Mikolov et al. 2013), are not suitable for One2Seq. As in traditional decoders, order information is incorporated by adding positional embeddings to the character embeddings. While in One2Seq, all previously decoded character embeddings, together with the positional embeddings, are aggregated to form a semantic token F_s^i (as shown in Equation (3)), leading to the loss of character order information. For example, aggregating the embeddings of decoded sub-strings "ent" and "net" would result in the same F_s^i, so as the *context token* F_{ct}^i. As a result, using these word embedding methods with positional embeddings is suboptimal for One2Seq.

To retain the order information of decoded characters into the *context token*, we propose Positional-aware Hash Embedding (PHE), as shown in the bottom-center of Figure 2. PHE aims to maps the characters into learnable features while retaining their positional information throughout the semantic aggregation process. It contains an index-position decomposing, a hash mapping, and an embedding retrieval.

Index-position decomposing is used to decompose the decoded characters $y^{0:i-1}$ into a set of (index, position) tuples, denoted as A^i. Specifically, we construct:

$$A^i = (a^0, a^1, ..., a^{i-1}), \tag{8}$$

where each a^j is a tuple consisting of a character index c^j and its position p^j:

$$a^j = (c^j, p^j), \tag{9}$$

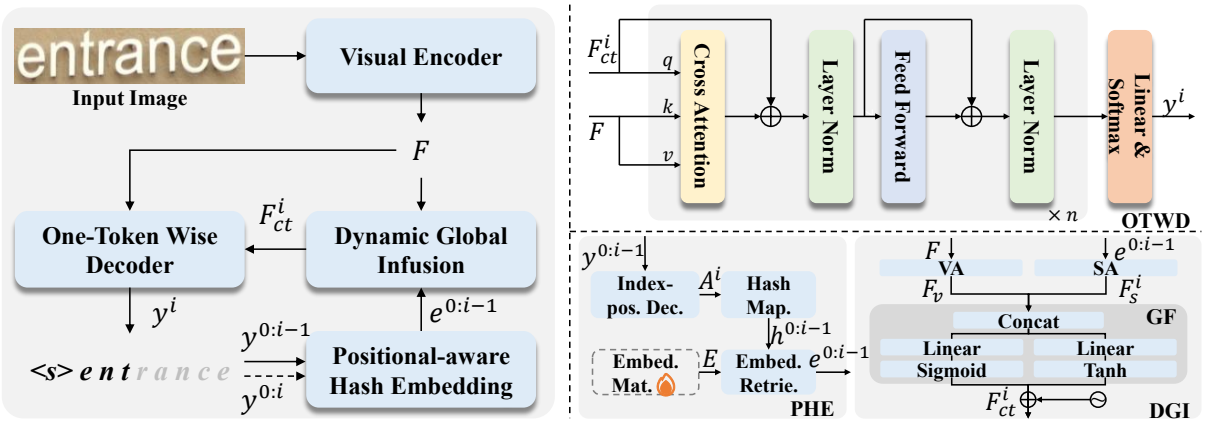


Figure 2: The overview of One2Seq. *Left*: The model architecture. The solid lines indicate the decoding process at the current time step, while the dashed lines represent the previously decoded characters being fed back into the *Positional-aware Hash Embedding* for the next-step decoding. *Top-right*: Details of *One-Token Wise Decoder (OTWD)*, which performs modality interaction through cross-attention. *Bottom-center* and *Bottom-right* are respectively the details of *Positional-aware Hash Embedding (PHE)* and *Dynamic Global Infusion (DGI)*.

where $j \in \{0, 1, \ldots, i-1\}$. $c^j \in \{1, 2, \ldots, N\}$ denotes the index of the j-th character y^j, and $p^j \in \{1, 2, \ldots, L\}$ is its corresponding position.

Hash mapping maps each tuple (c^j, p^j) in A^i to a hash value. In practice, we compute the hash value h^j of each tuple using linear transformation:

$$h^j = c^j * L + p^j \in \{1, 2, \ldots, L \times N\}, \tag{10}$$

where L is the maximum text length and N is the size of the character set.

Embedding retrieval is employed to extract learnable high-dimensional features $e^{0:i-1}$ based on the hash values $h^{0:i-1}$. Specifically, $h^{0:i-1}$ serves as the column index to retrieve the corresponding embedding vectors from a trainable embedding matrix $E \in \mathbb{R}^{d \times (L \times N)}$:

$$e^{0:i-1} = E_{:, h^{0:i-1}} \in \mathbb{R}^{d \times i}, \tag{11}$$

where $E \in \mathbb{R}^{d \times (L \times N)}$ denotes a trainable embedding matrix, and e^j denotes the retrieved embedding, $j \in \{0, 1, \ldots, i-1\}$.

By PHE, each embedding represents not only the index of the character, but also its position within the text. In contrast to positional encoding, this distinctive representation enables the embeddings to preserve positional information after aggregation.

One-Token Wise Decoder

The proposed One-Token Wise Decoder (OTWD) takes the *context token* F^i_{ct} as the query input, fuses it with visual features F via cross-attention, and predicts the character at the current time step. The structure of OTWD is illustrated in the top-right of Figure 2. Specifically, given the *context token* F^i_{ct} and visual features F, we first fuse them through multi-head cross-attention:

$$z = \text{Concat}(\text{head}_1, \ldots, \text{head}_r) W^o, \tag{12}$$

$$\text{head}_l = \text{Attention}(F^i_{ct} W^Q_l, F W^K_l, F W^V_l), \tag{13}$$

where $W^Q_l, W^K_l, W^V_l \in \mathbb{R}^{d \times d^r}$ are learnable weights, $d^r = \frac{d}{r}$, and $W^o \in \mathbb{R}^{d \times d}$ is the output projection weights.

Next, a feed-forward network is employed to extract channel-wise features as follows:

$$\text{FFN}(z) = (z W_1 + b_1) W_2 + b_2, \tag{14}$$

where $W_1, W_2 \in \mathbb{R}^{d \times d}$ are learnable weights and $b_1, b_2 \in \mathbb{R}^{d \times 1}$ are bias. Layer normalization is omitted in the equations for simplicity. The decoder can be stacked n times. After obtaining the output of the final decoder layer, a linear projection is applied to produce the predicted character y^i at the current time step. Then, $y^{0:i}$ are embedded into vectors using PHE, and subsequently fed back into the DGI to update the context token and proceed to decode the next character y^{i+1}.

Unlike the decoder in a Transformer or OTE-AR (Xu et al. 2024), OTWD takes a single token $F^i_{ct} \in \mathbb{R}^{d \times 1}$ as input at each time step, preserving the contextual awareness of AR decoding while avoiding the issue of the growing query sequence length, alleviates attention drift while preventing the computational overhead introduced by the accumulation of semantic information. Moreover, since DGI efficiently fuses the visual features and the semantic features, OTWD no longer relies on multi-head self-attention to model dependencies among source tokens, thereby further reducing computational cost.

Optimization Objective

During training, all source tokens are acquired. Therefore, we first compute all *context tokens* $(F^1_{ct}, ..., F^L_{ct})$ and then perform training to support parallel optimization. The training objective follows the traditional AR Loss, formulated as:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[-\sum_{t=1}^{L} \log p_\theta \left(y_t \mid y_{<t}, x \right) \right]. \tag{15}$$

	Method	Common Benchmarks							Union14M-L Benchmark								*Params*
		IC13	SVT	IIIT	IC15	SVTP	CUTE	Avg	CUR	MTO	ART	CTL	SAL	MTW	GEN	Avg	($\times 10^6$)
CTC	CRNN (Shi, Bai, and Yao 2017)	91.8	83.8	90.8	71.8	70.4	80.9	81.58	19.4	4.5	34.2	44.0	16.7	35.7	60.4	30.70	8.3
	SVTR† (Du et al. 2022)	96.7	96.7	98.2	88.6	91.3	96.5	94.70	79.3	69.4	67.1	68.8	75.8	76.8	77.0	73.48	18.1
	SVTRv2 (Du et al. 2025d)	**98.7**	98.0	**99.2**	91.0	93.6	99.0	96.58	91.0	89.5	78.7	**81.3**	85.9	85.1	82.3	84.91	21.0
PD	SRN (Yu et al. 2020)	97.5	96.3	97.2	87.9	90.9	96.9	94.45	78.1	63.2	66.3	65.3	71.4	58.3	76.5	68.43	51.7
	VisionLAN (Wang et al. 2021)	97.1	95.8	98.2	88.6	91.2	96.2	94.50	79.6	71.4	67.9	73.7	76.1	73.9	79.1	74.53	32.9
	ABINet† (Fang et al. 2021)	97.8	97.7	98.1	89.4	93.2	97.6	95.62	82.7	77.4	68.8	69.2	77.9	72.0	77.9	75.13	36.9
	MATRN (Na, Kim, and Park 2022)	97.9	98.3	98.8	90.3	95.2	97.2	96.29	82.2	73.0	73.4	76.9	79.4	77.4	81.0	77.62	44.3
	MGP-STR (Wang, Da, and Yao 2022)	97.1	97.8	97.9	89.6	95.2	96.9	95.75	85.2	83.7	72.6	75.1	79.8	71.1	83.1	78.65	148.0
	LPV (Zhang et al. 2023)	98.1	97.8	98.6	89.8	93.6	97.6	95.93	86.2	78.7	75.8	80.2	82.9	81.6	82.9	81.20	30.5
	OTE-PD† (Xu et al. 2024)	97.1	96.3	97.9	88.7	91.8	96.2	94.65	86.2	77.1	71.7	72.7	78.7	61.9	79.2	75.34	18.1
	CPPD† (Du et al. 2025b)	98.2	98.1	98.8	91.1	93.8	97.6	96.27	87.9	81.7	75.5	76.2	83.5	81.5	81.9	81.17	26.9
	IGTR-PD† (Du et al. 2025c)	98.4	97.8	98.6	89.6	93.5	97.6	95.91	88.9	91.3	75.5	76.4	83.9	78.0	82.1	82.29	24.1
AR	ASTER† (Shi et al. 2019)	96.2	94.6	97.6	87.4	88.7	93.4	92.96	74.2	78.8	61.3	65.9	75.9	70.6	76.9	71.97	19.1
	NRTR (Sheng, Chen, and Xu 2019)	97.8	96.8	98.1	88.9	93.3	94.4	94.89	67.9	42.4	66.5	73.6	66.4	77.2	78.3	67.46	44.3
	SAR† (Li et al. 2019)	96.7	94.0	97.7	84.8	84.5	94.4	92.03	73.2	63.5	60.1	68.8	64.2	75.4	74.7	74.65	57.5
	SEED (Qiao et al. 2020)	94.2	93.2	96.5	87.5	88.7	93.4	92.24	69.1	80.9	56.9	63.9	73.4	61.3	76.5	68.87	24.0
	RoScanner (Yue et al. 2020)	97.7	95.8	98.5	88.2	90.1	97.6	94.65	79.4	68.1	70.5	79.6	71.6	82.5	80.8	76.08	48.0
	PARSeq† (Bautista and Atienza 2022)	97.8	97.2	98.7	90.6	94.6	96.8	95.94	87.6	88.3	72.3	77.4	84.0	80.8	82.6	81.93	23.8
	MAERec (Jiang et al. 2023)	97.6	96.8	98.0	87.1	93.2	97.9	95.10	81.4	71.4	72.0	80.0	78.5	82.4	82.5	78.60	35.7
	LISTER† (Cheng et al. 2023)	97.3	96.6	98.5	88.2	90.7	96.5	92.45	87.1	88.2	72.5	78.3	79.7	81.3	80.3	81.06	20.5
	CDistNet (Zheng et al. 2024)	97.8	98.1	98.7	89.6	93.5	96.9	95.59	81.7	77.1	72.6	78.2	79.9	79.7	81.1	78.62	43.3
	BUSNet (Wei et al. 2024)	97.8	98.1	98.3	90.2	95.3	96.5	96.06	83.0	82.3	70.8	77.9	78.8	71.2	82.6	78.10	32.1
	OTE-AR† (Xu et al. 2024)	97.7	97.8	98.2	89.6	94.4	97.9	95.95	87.9	82.8	75.3	73.4	81.8	68.9	80.9	78.70	20.0
	CAM (Yang et al. 2024)	96.6	96.1	98.2	89.0	93.5	96.2	94.94	85.4	89.0	72.0	75.4	84.0	74.8	83.1	80.52	58.7
	SMTR (Du et al. 2025a)	98.4	98.2	98.8	90.0	93.3	97.6	94.14	90.5	92.6	75.5	80.0	84.9	**85.2**	82.6	84.47	15.8
	IGTR-AR† (Du et al. 2025c)	98.6	98.3	98.9	90.4	94.7	97.6	96.42	90.3	92.6	77.5	78.8	85.6	81.8	82.9	84.21	24.1
Ours	O2S-SVTR-B	98.5	97.7	99.0	91.0	95.0	**99.3**	96.73	91.5	90.8	79.7	80.4	86.1	83.4	83.5	85.06	34.7
	O2S-SVTR-L	**98.7**	**98.9**	99.0	**91.3**	95.4	98.6	96.99	92.5	90.8	**81.7**	80.0	86.9	84.1	84.0	85.71	53.3
	O2S-CLIP	**98.7**	98.5	99.0	90.8	**97.2**	99.0	**97.21**	**93.2**	**95.8**	80.4	80.4	**89.1**	85.1	**84.4**	**86.92**	101.2

Table 1: Results on English benchmarks tested against existing models when trained on real-world Union14M-Filter training set. Bold and underlined values denote the 1st and 2nd results in each column. † denotes that the result is obtained by training the model on Union14M-Filter using the code they released. The data presented in the table corresponds to Word Accuracy Ignore Cases (WAIC).

Experiments

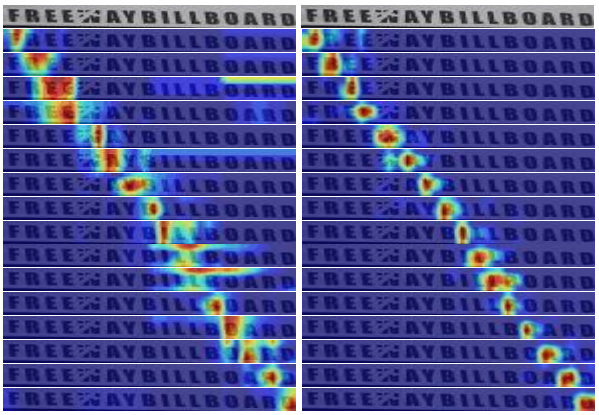
Datasets and Implementation Details

For training, we use a large-scale real-world dataset, Union14M-Filter (Du et al. 2025d), which is a filtered version of Union14M-L (Jiang et al. 2023). It contains 3.2 million training images from 17 datasets, covering various types of text such as curved, multi-oriented, artistic, and occluded text. For evaluation, we test our method on 19 English test datasets, including: (1) Six commonly used benchmarks (CoB) in the STR community: IIIT5k (3000) (Mishra, Alahari, and Jawahar 2012), SVT (647) (Wang, Babenko, and Belongie 2011), IC13 (857) (Karatzas et al. 2013), IC15 (1811) (Karatzas et al. 2015), SVTP (645) (Phan et al. 2013), and CUTE80 (288) (Risnumawan et al. 2014), which are widely adopted for performance comparison; (2) Union14M-benchmarks (U14M-B) (Jiang et al. 2023), which cover seven challenging categories of samples: curved (2426), multi-oriented (1369), artistic (900), context-less (779), salient (1585), multi-words (829), and general (400k); (3) More challenging benchmarks (MCB), including Uber (80.8k) (Zhang et al. 2017), ArT (25.2k) (Chng et al. 2019), COCO-Text (9.8k) (Veit et al. 2016), the stylized text dataset WordArt (1.5k) (Xie et al. 2022), and the occlusion-focused OST (2.4k) (including HOST and WOST) (Wang et al. 2021).

During training, we use the AdamW optimizer (Loshchilov and Hutter 2019) with a weight decay of 0.05. The learning rate is set to 3.2×10^{-5}, and the batch size is 1024. One cycle learning rate scheduler with 1.5 epochs of linear warm-up is applied over 25 total epochs. All images are resized to 32×128. For data augmentation, we adopt RandAugment (Cubuk et al. 2020) following PARSeq (Bautista and Atienza 2022). The maximum text length L is set to 25, and the length of the character set N is 96 (comprising uppercase and lowercase letters, digits, punctuation, as well as start, end, and padding tokens). The feature dimension d and the number of attention heads r are set to 512, 12, and 768, 16 for the O2S-SVTR-B and O2S-SVTR-L models, respectively. All experiments are conducted on a single 80GB A100 GPU.

Comparison with State-of-the-arts

Results on Common and Union14M Benchmarks. We first evaluate the performance of our method on the most commonly used STR benchmarks and compare it with ex-



(a) NRTR **(b) Ours**

Figure 3: Visualization of the attention maps in the decoders.

Method	Uber	ArT	COCO	Word-Art	HOST	WOST	Avg
ASTER	79.8	80.0	72.5	74.7	55.3	74.3	72.76
SAR	75.2	77.2	69.0	72.1	48.1	72.9	69.08
ABINet	77.0	80.3	74.4	79.2	66.5	80.3	76.27
SVTR	77.6	80.0	73.3	76.9	59.4	77.0	74.01
PARSeq	88.4	83.7	78.7	83.2	75.2	85.1	82.38
LISTER	83.2	83.2	75.0	81.7	62.3	79.3	76.95
OTE-AR	81.0	82.4	77.3	83.6	70.4	84.0	79.78
IGTR-AR	88.1	84.5	80.3	85.4	71.1	85.6	82.50
CPPD	84.3	83.1	77.7	84.7	72.4	86.8	81.49
SVTRv2	87.6	84.6	78.6	85.2	74.0	86.2	82.70
O2S-SVTR-B	89.2	84.6	80.3	86.7	76.7	87.1	84.09
O2S-SVTR-L	90.2	84.9	80.6	**87.2**	**79.0**	**88.8**	85.12
O2S-CLIP	**93.5**	**85.4**	**81.2**	86.8	76.5	87.3	**85.14**

Table 2: Results on more challenging benchmarks.

Decoder	CoB	U14M-B	MCB	FLOPS (G)	Time (ms)
CTC	94.70	73.48	74.01	-	-
NRTR	96.47	84.61	82.92	0.66	23.2
OTE-PD	94.65	75.34	76.87	-	-
OTE-AR	95.95	78.70	79.78	0.07	13.1
OTWD (Ours)	**96.73**	**85.06**	**84.09**	0.12	13.5

Table 3: Comparison of different decoders. The data in the table represent the average accuracies on three types of benchmarks.

isting approaches, as shown in Table 1. The results demonstrate that our proposed method outperforms previous STR methods and achieves state-of-the-art performance, highlighting the effectiveness of our decoding strategy. Specifically, One2Seq ranks first on 10 out of 13 benchmarks. Compared with the latest CTC-based method SVTRv2 (Du et al. 2025d) and the state-of-the-art parallel decoding method CPPD (Du et al. 2025b), our One2Seq-SVTR-B, despite using a simpler visual encoder, still achieves higher accuracy by 0.15% and 0.46% on CoB, and 0.15% and 4.89% on U14M-B, respectively. These results demonstrate the superiority of our approach. Moreover, results on more challenging benchmarks (presented in the next section) further highlight the advantages of AR-based decoding in handling complex scenarios. Furthermore, when compared with the latest AR decoding methods, including OTE-AR, IGTR-AR, and SMTR, One2Seq-SVTR-B outperforms them by 0.78%, 0.31%, and 2.59% on CoB, and by 6.36%, 0.85%, and 0.69% on U14M-B, respectively. Since all models adopt the same encoder, the observed performance differences can be primarily ascribed to variations in decoder design, thereby validating the effectiveness of our proposed decoding paradigm.

When scaling O2S-SVTR-B to O2S-SVTR-L, the recognition accuracy further improves, indicating that our method can effectively leverage higher-dimensional visual features for more accurate decoding. When replacing the visual encoder with a larger CLIP encoder, the model achieves accuracy gains of 1.8% on *CUR* and 5.0% on *MO*, while only a 0.7% improvement is observed on *ART*. This is because the CLIP encoder has been trained on a large number of real-world images in an unsupervised manner, which endows it with stronger zero-shot capabilities for common deformations such as rotation and curvature, compared to the relatively rare artistic fonts in its pretraining corpus.

Results on More Challenging Datasets. Table 2 presents the performance of One2Seq on more challenging test sets. Despite the increased difficulty, One2Seq still achieves strong recognition accuracy, outperforming existing methods. Notably, on the HOST dataset, One2Seq achieves a significant improvement in accuracy. This gain can be attributed to the introducing of global visual features in DGI, which enables the model to reason with global context when decoding heavily occluded or hard-to-recognize characters.

Influence of One-Token Wise Decoder

To verify the effectiveness of our proposed OTWD, we compare it with other classic decoders. For fairness, all methods use the same encoder, SVTR-B. The experimental results are shown in Table 3. Despite their advantages in decoding speed and computational efficiency, CTC-based and PD-based decoders exhibit significantly inferior recognition performance compared to AR-based decoders. When comparing our OTWD with existing decoders, we find that OTWD achieves the highest recognition performance while maintaining low computational cost and decoding time. Specifically, compared to NRTR, OTWD achieves accuracy improvements of 0.26%, 0.45%, and 1.17% across the three benchmark categories, respectively. This improvement is mainly attributed to: (1) Its effectiveness in alleviating attention drift, and (2) The explicit use of the visual token, which will be discussed in the next section. More notably, due to the one-token wise input design, OTWD achieves a substantial reduction in computational complexity (approximately 1/5 of that of NRTR) while also reducing the decoding time to merely 58.2% of NRTR's. More experimental results can be found in the supplementary material.

To further validate the effectiveness of our method, we visualize the attention maps, as shown in Figure 3. Compared to NRTR, our method is able to more accurately attend to

DGI		PHE	CoB	U14M-B	MCB	Avg
VIS	GF					
			96.29	83.61	81.77	87.22
✓			96.31	83.98	82.70	87.66
✓	✓		96.55	84.32	83.54	88.14
✓	✓	✓	**96.73**	**85.06**	**84.09**	**88.62**

Table 4: The results of the ablation study. VIS refers to the introduction of global visual features. GF represents the gated fusion in DGI (without GF, we simply add the visual token and the semantic token). In the absence of PHE, characters are encoded using Word2Vec in conjunction with standard positional embeddings.

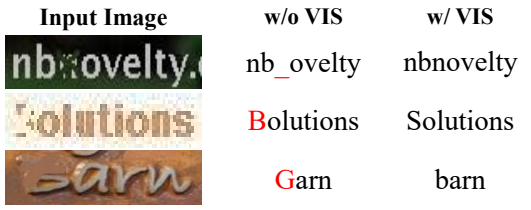


Figure 4: Results on occluded samples *w/* or *w/o* VIS.

the correct spatial location of each character at every decoding step. This improvement stems from a key difference: unlike NRTR, our One2Seq encodes the previously decoded semantic information into a single token. This design alleviates the issue in NRTR where the positional information of the decoded sequence becomes increasingly diluted over time, thereby mitigating the effect of attention drift.

Influence of Dynamic Global Infusion

Table 4 shows that incorporating global visual information leads to accuracy gains of 0.02%, 0.37%, and 0.93% across the three benchmark categories, respectively (row #1 $vs.$ row #2). This enhancement can be attributed to the fact that AR decoders tend to lack a comprehensive understanding of the input image during the early stages of decoding. In challenging scenarios, such as occluded text or blurred characters, the integration of global context facilitates a more accurate interpretation of the local semantic regions attended to at each decoding step, thereby improving recognition performance. The results shown in Figure 4 further support this argument. When the first few characters of the text are difficult to recognize due to occlusion, the introduction of global information enables the model to leverage the overall context of the image at an earlier stage, thereby reducing misidentification in the initial decoding steps.

Subsequently, incorporating gated fusion (row #3) yielded higher recognition accuracy compared to simple aggregation, with improvements of 0.24%, 0.34%, and 0.84% across the three benchmarks, respectively. This is because the gated mechanism allows for weighted fusion of visual and semantic information, enhancing the model's representational capacity for complex tasks. Furthermore, the gated mechanism can suppress irrelevant information during the fusion process, thereby reducing noise.

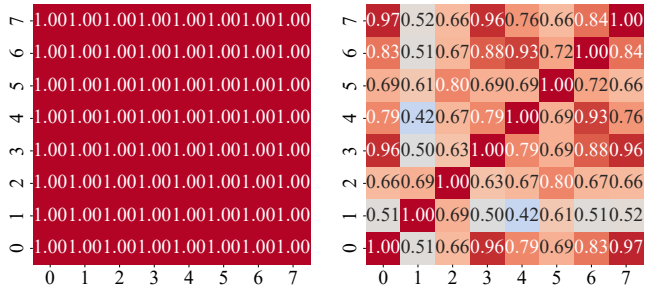


Figure 5: The similarity matrix of context tokens of different permuted inputs. *Left*: *w/o* PHE, *Right*: *w/* PHE.

Influence of Positional-aware Hash Embedding

The introduction of PHE (row #4 in Table 4) further improves the model's performance. This improvement can be attributed to the fact that, in One2Seq, the previously decoded characters are averaged after embedding. When using Word2Vec for embedding, the sequential order of these characters is lost, which is suboptimal for AR decoding, where character order plays a crucial role. PHE preserves the positional information of the decoded characters, thereby enabling more informative embeddings. As a result, the use of PHE leads to accuracy improvements of 0.18%, 0.74%, and 0.55% across the three benchmark categories, respectively.

To further explore the effectiveness of PHE, we randomly shuffled the order of the characters decoded at the same time step to generate different source tokens, and then computed the cosine similarity between the output context tokens. The visualization results are shown in Figure 5. When PHE is not used, the cosine similarity between the output context tokens is consistently 1, indicating that the model has completely lost the order information of the already decoded characters. After using PHE, the context tokens generated from different orderings of the source tokens show variations. This allows the model to distinguish between different permutations of source tokens, thus demonstrating awareness of the order of the decoded characters.

Conclusion

In this work, we present a novel decoding paradigm for STR, named One2Seq. It integrates the previously decoded semantic features into a single *context token*, which is then used for decoding via the proposed *One-Token Wise Decoder*. This paradigm not only mitigates the issue of attention drift in AR decoders, but also reduces the computational cost during decoding. To ensure that the context token preserves the order of decoded characters, we further introduce *Positional-aware Hash Embedding*. Moreover, *Dynamic Global Infusion* is proposed to incorporate global visual information into the context token, further enhancing the model's performance on challenging samples. Extensive experiments have proved the effectiveness of One2Seq. In the future, we will investigate the effect of scaling on One2Seq and evaluate its performance on other challenging samples like long text images.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62302532); in part by the Natural Science Foundation of Guangdong (Grant No. 2025A1515011224); in part by Shenzhen Science and Technology Program (Grant No. 202206193000001, 20220816225523001, KQTD20221101093559018); in part by Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023A1515030032); and in part by Open Fund of National Engineering Laboratory for Big Data System Computing Technology (Grant No. SZU-BDSC-OF2024-04).

References

Bautista, D.; and Atienza, R. 2022. Scene Text Recognition with Permuted Autoregressive Sequence Models. In *ECCV*, 178–196.

Cheng, C.; Wang, P.; Da, C.; Zheng, Q.; and Yao, C. 2023. LISTER: Neighbor Decoding for Length-Insensitive Scene Text Recognition. In *ICCV*, 19484–19494.

Chng, C. K.; Ding, E.; Liu, J.; Karatzas, D.; Chan, C. S.; Jin, L.; Liu, Y.; Sun, Y.; Ng, C. C.; Luo, C.; Ni, Z.; Fang, C.; Zhang, S.; and Han, J. 2019. ICDAR2019 Robust Reading Challenge on Arbitrary-Shaped Text-RRC-ArT. In *ICDAR*, 1571–1576.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. 2020. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. In *CVPR Workshops*, 702–703.

Du, Y.; Chen, Z.; Jia, C.; Gao, X.; and Jiang, Y. 2025a. Out of Length Text Recognition with Sub-String Matching. In *AAAI*, 2798–2806.

Du, Y.; Chen, Z.; Jia, C.; Yin, X.; Li, C.; Du, Y.; and Jiang, Y. 2025b. Context Perception Parallel Decoder for Scene Text Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(6): 4668–4683.

Du, Y.; Chen, Z.; Jia, C.; Yin, X.; Zheng, T.; Li, C.; Du, Y.; and Jiang, Y. 2022. SVTR: Scene Text Recognition with a Single Visual Model. In *IJCAI*, 884–890.

Du, Y.; Chen, Z.; Su, Y.; Jia, C.; and Jiang, Y. 2025c. Instruction-Guided Scene Text Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(4): 2723–2738.

Du, Y.; Chen, Z.; Xie, H.; Jia, C.; and Jiang, Y.-G. 2025d. SVTRv2: CTC Beats Encoder-Decoder Models in Scene Text Recognition. In *ICCV*, In press.

Fang, S.; Xie, H.; Wang, Y.; Mao, Z.; and Zhang, Y. 2021. Read Like Humans: Autonomous, Bidirectional and Iterative Language Modeling for Scene Text Recognition. In *CVPR*, 7098–7107.

Gao, Y.; Chen, Y.; Wang, J.; Tang, M.; and Lu, H. 2019. Reading scene text with fully convolutional sequence modeling. *Neurocomputing*, 339: 161–170.

Graves, A.; Fernández, S.; Gomez, F. J.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 369–376.

He, P.; Huang, W.; Qiao, Y.; Loy, C. C.; and Tang, X. 2016. Reading Scene Text in Deep Convolutional Sequences. In *AAAI*, 3501–3508.

Jiang, Q.; Wang, J.; Peng, D.; Liu, C.; and Jin, L. 2023. Revisiting Scene Text Recognition: A Data Perspective. In *ICCV*, 20486–20497.

Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S. K.; Bagdanov, A. D.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V. R.; Lu, S.; Shafait, F.; Uchida, S.; and Valveny, E. 2015. ICDAR 2015 competition on Robust Reading. In *ICDAR*, 1156–1160.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazán, J.; and de las Heras, L. 2013. ICDAR 2013 Robust Reading Competition. In *ICDAR*, 1484–1493.

Li, H.; Wang, P.; Shen, C.; and Zhang, G. 2019. Show, Attend and Read: A Simple and Strong Baseline for Irregular Text Recognition. In *AAAI*, 8610–8617.

Litman, R.; Anschel, O.; Tsiper, S.; Litman, R.; Mazor, S.; and Manmatha, R. 2020. SCATTER: Selective Context Attentional Scene Text Recognizer. In *CVPR*, 11959–11969.

Liu, W.; Chen, C.; Wong, K. K.; Su, Z.; and Han, J. 2016. STAR-Net: A SpaTial Attention Residue Network for Scene Text Recognition. In *BMVC*, 7.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *ICLR*.

Luo, C.; Jin, L.; and Sun, Z. 2019. MORAN: A Multi-Object Rectified Attention Network for scene text recognition. *Pattern Recognit.*, 90: 109–118.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. In *ICLR Workshop*.

Mishra, A.; Alahari, K.; and Jawahar, C. V. 2012. Scene Text Recognition using Higher Order Language Priors. In *BMVC*, 1–11.

Na, B.; Kim, Y.; and Park, S. 2022. Multi-modal Text Recognition Networks: Interactive Enhancements Between Visual and Semantic Features. In *ECCV*, 446–463.

Phan, T. Q.; Shivakumara, P.; Tian, S.; and Tan, C. L. 2013. Recognizing Text with Perspective Distortion in Natural Scenes. In *ICCV*, 569–576.

Posner, I.; Corke, P.; and Newman, P. M. 2010. Using text-spotting to query the world. In *IROS*, 3181–3186.

Qiao, Z.; Zhou, Y.; Yang, D.; Zhou, Y.; and Wang, W. 2020. SEED: Semantics Enhanced Encoder-Decoder Framework for Scene Text Recognition. In *CVPR*, 13525–13534.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 8748–8763.

Risnumawan, A.; Shivakumara, P.; Chan, C. S.; and Tan, C. L. 2014. A robust arbitrary text detection system for natural scene images. *Expert Syst. Appl.*, 41(18): 8027–8048.

Sheng, F.; Chen, Z.; and Xu, B. 2019. NRTR: A No-Recurrence Sequence-to-Sequence Model for Scene Text Recognition. In *ICDAR*, 781–786.

Shi, B.; Bai, X.; and Yao, C. 2017. An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(11): 2298–2304.

Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2019. ASTER: An Attentional Scene Text Recognizer with Flexible Rectification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(9): 2035–2048.

Su, B.; and Lu, S. 2014. Accurate Scene Text Recognition Based on Recurrent Neural Network. In *ACCV*, 35–48.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NeurIPS*, 5998–6008.

Veit, A.; Matera, T.; Neumann, L.; Matas, J.; and Belongie, S. J. 2016. COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images. *CoRR*, abs/1601.07140.

Wang, K.; Babenko, B.; and Belongie, S. J. 2011. End-to-end scene text recognition. In *ICCV*, 1457–1464.

Wang, P.; Da, C.; and Yao, C. 2022. Multi-granularity Prediction for Scene Text Recognition. In *ECCV*, 339–355.

Wang, T.; Zhu, Y.; Jin, L.; Luo, C.; Chen, X.; Wu, Y.; Wang, Q.; and Cai, M. 2020. Decoupled Attention Network for Text Recognition. In *AAAI*, 12216–12224.

Wang, Y.; Xie, H.; Fang, S.; Wang, J.; Zhu, S.; and Zhang, Y. 2021. From Two to One: A New Scene Text Recognizer with Visual Language Modeling Network. In *ICCV*, 14174–14183.

Wei, J.; Zhan, H.; Lu, Y.; Tu, X.; Yin, B.; Liu, C.; and Pal, U. 2024. Image as a Language: Revisiting Scene Text Recognition via Balanced, Unified and Synchronized Vision-Language Reasoning Network. In *AAAI*, 5885–5893.

Wu, W.; Bouyarmane, K.; and Tutar, I. B. 2023. Catalog Phrase Grounding (CPG): Grounding of Product Textual Attributes in Product Images for e-commerce Vision-Language Applications. *CoRR*, abs/2308.16354.

Xie, X.; Fu, L.; Zhang, Z.; Wang, Z.; and Bai, X. 2022. Toward Understanding WordArt: Corner-Guided Transformer for Scene Text Recognition. In *ECCV*, 303–321.

Xu, J.; Wang, Y.; Xie, H.; and Zhang, Y. 2024. OTE: Exploring Accurate Scene Text Recognition Using One Token. In *CVPR*, 28327–28336.

Yang, M.; Yang, B.; Liao, M.; Zhu, Y.; and Bai, X. 2024. Class-Aware Mask-guided feature refinement for scene text recognition. *Pattern Recognit.*, 149: 110244.

Yu, D.; Li, X.; Zhang, C.; Liu, T.; Han, J.; Liu, J.; and Ding, E. 2020. Towards Accurate Scene Text Recognition With Semantic Reasoning Networks. In *CVPR*, 12110–12119.

Yue, X.; Kuang, Z.; Lin, C.; Sun, H.; and Zhang, W. 2020. RobustScanner: Dynamically Enhancing Positional Clues for Robust Text Recognition. In *ECCV*, 135–151.

Zhang, B.; Xie, H.; Wang, Y.; Xu, J.; and Zhang, Y. 2023. Linguistic More: Taking a Further Step toward Efficient and Accurate Scene Text Recognition. In *IJCAI*, 1704–1712.

Zhang, C.; Tao, Y.; Du, K.; Ding, W.; Wang, B.; Liu, J.; and Wang, W. 2022. Character-Level Street View Text Spotting Based on Deep Multisegmentation Network for Smarter Autonomous Driving. *IEEE Trans. Artif. Intell.*, 3(2): 297–308.

Zhang, H.; Yao, Q.; Yang, M.; Xu, Y.; and Bai, X. 2020. AutoSTR: Efficient Backbone Search for Scene Text Recognition. In *ECCV*, 751–767.

Zhang, Y.; Gueguen, L.; Zharkov, I.; Zhang, P.; Seifert, K.; and Kadlec, B. 2017. Uber-Text: A Large-Scale Dataset for Optical Character Recognition from Street-Level Imagery. In *CVPR Workshop*, 5.

Zhao, S.; Quan, R.; Zhu, L.; and Yang, Y. 2024. CLIP4STR: A Simple Baseline for Scene Text Recognition With Pre-Trained Vision-Language Model. *IEEE Trans. Image Process.*, 33: 6893–6904.

Zheng, T.; Chen, Z.; Fang, S.; Xie, H.; and Jiang, Y. 2024. CDistNet: Perceiving Multi-domain Character Distance for Robust Text Recognition. *Int. J. Comput. Vis.*, 132(2): 300–318.