Explicit Relational Reasoning Network for Scene Text Detection

Yuchen Su[1], Zhineng Chen[1] , Yongkun Du[1], Zhilong Ji[2], Kai Hu[3], Jinfeng Bai[2], Xieping Gao[4] †

[1] School of Computer Science, Fudan University
[2] Tomorrow Advancing Life
[3] School of Computer Science, Xiangtan University
[4] Laboratory for Artificial Intelligence and International Communication, Hunan Normal University
{ycsu23, ykdu23}@m.fudan.edu.cn, zhinchen@fudan.edu.cn, zhilongji@hotmail.com, jfbai.bit@gmail.com,
kaihu@xtu.edu.cn, xpgao@hunnu.edu.cn

Abstract

Connected component (CC) is a proper text shape representation that aligns with human reading intuition. However, CC-based text detection methods have recently faced a developmental bottleneck that their time-consuming post-processing is difficult to eliminate. To address this issue, we introduce an explicit relational reasoning network (ERRNet) to elegantly model the component relationships without post-processing. Concretely, we first represent each text instance as multiple ordered text components, and then treat these components as objects in sequential movement. In this way, scene text detection can be innovatively viewed as a tracking problem. From this perspective, we design an end-to-end tracking decoder to achieve a CC-based method dispensing with post-processing entirely. Additionally, we observe that there is an inconsistency between classification confidence and localization quality, so we propose a Polygon Monte-Carlo method to quickly and accurately evaluate the localization quality. Based on this, we introduce a position-supervised classification loss to guide the task-aligned learning of ERRNet. Experiments on challenging benchmarks demonstrate the effectiveness of our ERRNet. It consistently achieves state-of-the-art accuracy while holding highly competitive inference speed.

1 Introduction

Scene text detection aims to locate text regions within images. It is a fundamental step for many computer vision and artificial intelligence tasks (Zhang et al. 2021a; Wei et al. 2022; Fang et al. 2022; Wang et al. 2022a; Meng et al. 2022; Zhang et al. 2024). Despite recent advancements, detecting text in the wild remains challenging due to the varied scales, shapes, colors, and fonts of text.

Segmentation- and regression-based methods are two mainstream scene text detection methods. The former (Liao et al. 2022; Zhao et al. 2024) utilize shrunk text kernel to separate adhesive text instances and cluster text pixels into distinct instances through heuristic post-processing. Although this pixel-level representation approach can flexibly fit arbitrary-shaped text, it overly focuses on local textual cues, leading to sensitivity to local noise, as illustrated in Fig. 2(a). In contrast, regression-based methods (Wang et al. 2022b; Su et al. 2024) regress parameterized text shapes to directly capture the text's overall geometric layout, which have higher resistance to local noise. However, these methods lack scale and shape invariance, as text scale and shape show great variability, making it difficult to directly perceive the overall geometric layout of complex text with accuracy, as shown in Fig. 2(b).

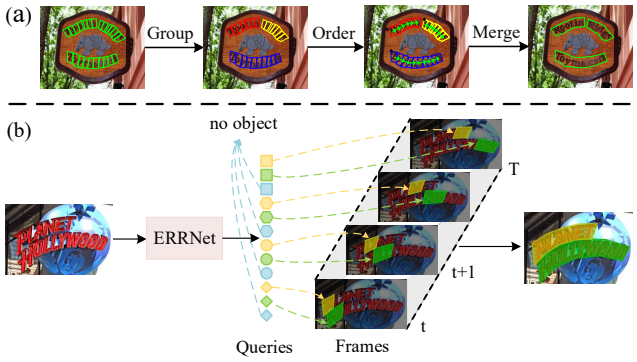
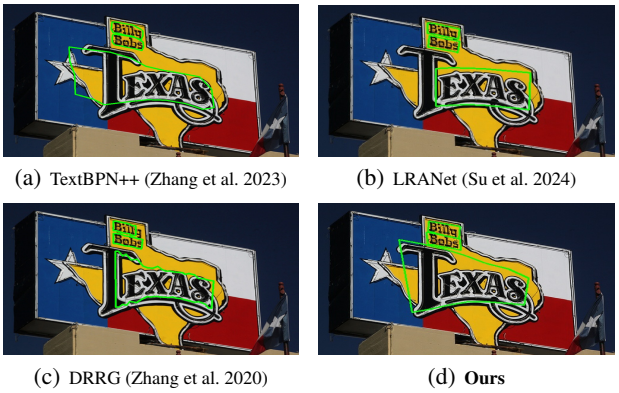


Figure 1: (a) Post-processing illustration of typical CC-based methods, which require grouping and ordering components one by one. (b) The pipeline of ERRNet, which has no post-processing. ERRNet views each text instance as multiple text components in sequential movement. Same shape queries indicate predictions in different text instances but in the same sequential position, same color queries represent predictions in one instance, and temporal relationships denote the sequential relationships of components.

From a hybrid perspective, connected component (CC)-based methods (Zhang et al. 2020), which treat each text instance as a combination of a series of adjacent text components, are a reasonable integration of the above two types of methods. Compared to complex text contours, text components have fixed shapes (e.g., circles (Long et al. 2018), quadrilaterals (Feng et al. 2019)), and smaller size variations. Meanwhile, components are more resistant to local noise than individual pixels. However, CC-based methods have been less studied recently due to their tricky and time-consuming post-processing, which includes grouping components one by one based on their associative relationships to differentiate between text instances, and then ordering in-

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
† Corresponding author.



(a) TextBPN++ (Zhang et al. 2023) (b) LRANet (Su et al. 2024)

(c) DRRG (Zhang et al. 2020) (d) **Ours**

Figure 2: Comparison with leading text detection methods of three different types: (a) Segmentation-based method (TextBPN++), (b) Regression-based method (LRANet), and (c) Connected component-based method (DRRG).

ternal components within each instance individually according to their sequential relationships, as depicted in Fig. 1(a).

To address this issue, we formulate scene text detection as a tracking problem for the first time. As shown in Fig. 1(b), each text instance is decomposed into multiple ordered text components. The instances are then represented by a series of motion frames, each containing a specified text component of every text instance. In each frame, we predict components across different text instances in a consistent sequence, where temporal relationships mirror the sequential relationships of components. By leveraging this conceptualization, we can transform the CC-based detection into an end-to-end tracking task, dispensing with post-processing entirely.

To this end, we introduce an explicit relational reasoning (ERR) decoder that seamlessly integrates relational prediction of components within the framework of positional regression. As shown in Fig. 1(b), the same shape queries represent predictions in different text instances but in the same frame; the same color queries indicate predictions belonging to the same instance; and each frame has the number of predictions for parallel processing. To achieve this goal, we employ bipartite graph matching between the output component sequence and the ground-truth component sequence for each text instance, and supervise the sequence as a whole.

Additionally, we observe that text detection methods encounter a misalignment issue between classification and localization tasks, resulting in either high classification confidence with relatively low localization quality or vice versa. Existing studies mostly overlook this issue, mainly because efficiently evaluating the localization quality of arbitrary-shaped text detection results is difficult. To address this, we propose a Polygon Monte-Carlo method to quickly and accurately calculate the Polygon Intersection over Union (PIoU) between predicted results and ground-truth. Based on this, we introduce a position-supervised classification loss to better guide the task-aligned learning.

Building upon these designs, we propose an explicit relational reasoning network, termed ERRNet. It first generates initial text components via a text component initialization module, and then directly outputs the position of each component and the relationship among different components in order, achieving accurate and efficient scene text detection. The main contributions of this paper are as follows:

- We propose ERRNet, a much simpler and faster CC-based text detection method. It eliminates the complex post-processing by innovatively modeling the component relationships from a tracking perspective.
- We introduce a position-supervised classification loss to force the classification confidence and localization quality of text instances to be consistent, guiding the detector better trained and thereby enhancing the detection performance.
- Extensive experiments are conducted on challenging benchmarks, which demonstrate that ERRNet is the most accurate detector and it also ranks among the fastest detectors.

2 Related Work

2.1 Segmentation-Based Methods

Segmentation-based methods (Xu et al. 2019; Zhang et al. 2021b; Yang et al. 2023) view text detection as an image segmentation problem, which usually adopt text kernels to separate adhesive text instances and expand them with heuristic post-processing. For example, DB (Liao et al. 2020) and its improved version DB++ (Liao et al. 2022) introduce a differentiable binarization module that assigns higher thresholds to text boundaries, reinforcing the distinction between adjacent text instances. CBNet (Zhao et al. 2024) proposes a context-aware module to enhance the text kernel segmentation results and a boundary-guided module to expand the text kernel in a learnable manner. Although these pixel-level modeling methods can flexibly fit arbitrary-shaped text, they usually need computationally intensive post-processing to reconstruct text boundaries, and are sensitive to text-like backgrounds due to neglecting the geometric context of holistic text instances.

2.2 Regression-Based Methods

Regression-based methods (Liu et al. 2020; He et al. 2021; Su et al. 2024) treat scene text detection as a special type of object detection. Earlier methods (Liao et al. 2017; Zhou et al. 2017; Lyu et al. 2018) use modified anchor-mechanisms to detect multi-oriented text instances. For example, Textboxes (Liao et al. 2017) increases the proportion of anchor-boxes to adapt to varied text scales.

To detect irregularly shaped text, some parameterized text shape methods are proposed. For example, ABCNet (Liu et al. 2020) utilizes Bernstein polynomial to convert the long sides of text into Bezier curves. LRANet (Su et al. 2024) leverages a linear combination of pre-defined eigenvectors to represent text boundaries. However, only optimizing the regression target is insufficient, as text scale and shape have great variability, accurately perceiving the overall text layout requires a meticulously designed network that provides a large receptive field. In response, CT-Net (Shao et al. 2023) proposes multi-stage contour refinement modules to iteratively and adaptively refine text contours. Similarly, DPText-DETR (Ye et al. 2023) employs a Transformer framework to address complex text layouts by capturing long-range contextual dependencies. However, the complex structure of

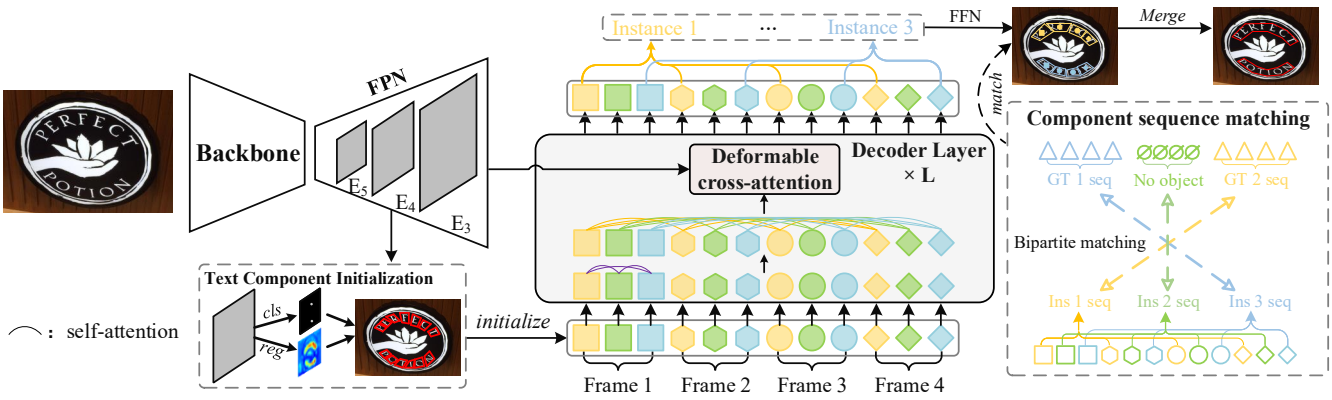


Figure 3: The architecture of ERRNet, which is mainly composed of three modules: (a) the backbone and feature pyramid network (FPN) for multi-scale feature extraction, (b) the text component initialization module to generate initial component queries, and (c) the explicit relational reasoning decoder for decoding the component sequence for each text instance in order.

these methods hinders the further development of efficient and accurate text detectors following this pipeline.

2.3 CC-Based Methods

Connected component (CC)-based methods (Tian et al. 2016; Long et al. 2018; Zhang et al. 2020) can be viewed as a middle ground between segmentation-based and regression-based methods, where the basic shape representation unit is a text part or character, followed by a linking-based post-processing procedure for generating final text boundaries. Before the era of deep learning, CC-based methods (Yin et al. 2013; Sun et al. 2015) had been widely used in scene text detection, as CC is an ideal text shape representation that aligns with human reading intuition. In recent years, CTPN (Tian et al. 2016) uses horizontal text components with a fixed-size width for handling text instances with extreme aspect ratios. TextSnake (Long et al. 2018) introduces an ordered set of disks to represent text instances, and adopts a segment network to learn the relationship among disks. DRRG (Zhang et al. 2020) introduces a graph convolutional network (GCN) to learn the associative relationships of text components for grouping, and uses the shortest path algorithm to infer the sequential relationship of components in each group. ReLaText (Ma et al. 2021) formulates CC-based methods as a scene graph generation task, and utilizes GCN to learn the relationships among pre-defined triplets.

Although CC is a better text shape representation due to its stability in size and shape, and its flexibility in representing text of arbitrary shapes, the complex and tricky post-processing has slowed the progress of CC-based methods. Therefore, we formulate the CC-based detection into an end-to-end tracking-like pipeline to eliminate post-processing, aiming to make CC-based methods shine again.

3 Methodology

3.1 Overview

The overall structure of ERRNet is illustrated in Fig. 3. Given an image with text, ERRNet first employs a ResNet-50 (He et al. 2016) with DCN (Zhu et al. 2019) as the backbone network, followed by a feature pyramid network (FPN) to extract multi-scale feature maps. Subsequently, a text component initialization (TCI) module generates initial coarse text components. These components are then sent into a Transformer decoder, which further refines them and establishes their associative and sequential relationships. Finally, the text components are aggregated into holistic text instances according to the explicit reasoning results. Besides, we also discuss how to better align the classification and localization tasks from the loss perspective and use it to train ERRNet better.

3.2 Text Component Initialization

In our work, each text boundary is represented using a series of ordered quadrilateral components, and each text component is represented by four vertices, as shown in Fig. 4. To delineate components within each text instance, we first apply the method in (Wang et al. 2022b) to divide the text contour into two long sides and determine the order and starting point of the contour points, as shown in Fig. 4(b). Subsequently, we sample m points on each long side to divide each text instance into a series of components ordered along these long sides. Here, we adopt B-spline interpolation for point sampling because its strong local control capability allows for better fitting of complex text shapes. To elaborate, we construct a B-spline curve with the following formula:

$$C(u) = \sum_{i=0}^{\bar{n}} N_{i,k}(u)\bar{P}_i\,, \tag{1}$$

where $\bar{P}_i$ is the i-th vertex in ground-truth long side, $N_{i,k}(u)$ is the B-spline curve basis function of degree k:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

$$\begin{aligned} N_{i,k}(u) = &\frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) \\ &+ \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u) \end{aligned} \tag{3}$$

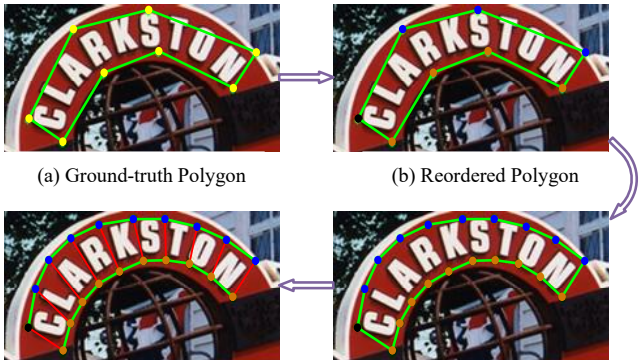


Figure 4: Illustration of ground-truth text component generation for a text primitive. Black point means start point.

where $k = 3$, and u_i denotes the $i\text{-}th$ knot, with the half-open interval $[u_i, u_{i+1})$ representing the $i\text{-}th$ knot span. Next, we equally sample m vertices on the B-spline curve (see Fig. 4(c)):

$$P_j = C\left(u_j\right) = \sum_{i=1}^{\bar{n}} \bar{P}_i N_{i,k}\left(u_j\right), \quad j = 1, 2, \ldots, m. \quad (4)$$

Thus, each text instance can be divided into a series of ordered quadrilateral regions based on the sampling points and pre-defined directions, as shown in Fig. 4(d).

Following (Su et al. 2024), we adopt a lightweight module to predict the text components. Specifically, after each output layer of the FPN, two sets of 3×3 convolution layers are utilized to extract classification features F_{cls} and regression features F_{reg}, respectively. Then, distinct 3×3 convolutions are applied to F_{cls} and F_{reg} for classifying and regressing the interpolated and sorted text contours. Finally, we organize these text contours into a sequence of ordered quadrilateral components. Each component $F_{comp} \in \mathbb{R}^{1 \times 4c_v}$ is formed by concatenating the positional encodings of its four vertices along the channel dimension, where c_v denotes the number of channels per vertex.

3.3 Explicit Relational Reasoning Decoder

CC-based methods need grouping and ordering text components to construct the text boundaries, as shown in Fig. 1(a). Previous methods (Long et al. 2018; Zhang et al. 2020) handle these components sequentially in a non-parallel fashion. To get rid of this computationally intensive post-processing, we develop a unique perspective of treating scene text detection as a tracking problem, where scene text is conceptualized as text components in sequential movement. From this viewpoint, we propose an explicit relational reasoning strategy to reformulate CC-based methods as an end-to-end tracking task and eliminate post-processing.

Specifically, we first select top-n groups of components from the TCI module based on descending classification scores. Notably, n is typically larger than the number of instances in any given image. Next, these components are organized into a series of hypothetical frames $F = (F_1^1, F_1^2, \cdots, F_1^n), \cdots, (F_t^1, F_t^2, \cdots, F_t^n)$, where t and n denote the total number of frames and the number of components in each frame, respectively. The frames satisfy: 1) $F_i^1, F_i^2, \cdots, F_i^n$ have the same ordinal position but in different text instances; 2) $F_1^j, F_2^j, \cdots, F_t^j$ belong to the same instance, maintaining an internal order consistent with their temporal sequence from 1 to t. Through this definition, we effectively map the associative and sequential relationships among text components onto positional relations within each frame and temporal relations across these frames.

When the output order aligns with the above pre-defined order, we can directly distinguish text instances based on the content of each frame and determine the order of components within each text instance according to the temporal relationships, thereby directly getting the text predictions. To achieve this, we introduce a component sequence matching that supervises the component sequence as a whole.

Component Sequence Matching. As the ERRNet decodes n components each frame, the number of text instances formed by component sequence is also n. We denote the predicted components sequences as $\hat{y} = \{\hat{y}_i\}_{i=1}^n$ and define the ground-truth set of component sequence as y, which consists of n elements padded with $\emptyset$. To find an optimal pair-wise matching between these two sets, we seek a permutation of n elements that minimizes the cost:

$$\hat{\sigma} = \arg\min \sum_{i}^{n} \mathcal{L}_{\text{match}}\left(\hat{y}_{\sigma(i)}, y_i\right), \quad (5)$$

where $\mathcal{L}_{\text{match}}\left(\hat{y}_{\sigma(i)}, y_i\right)$ represents the matching cost between the predicted component sequence indexed by $\sigma(i)$ and the ground-truth y_i.

Given the $N = n \cdot t$ quadrilateral predictions for the component prediction sequence, we can associate n component sequences for each instance based on their location indices, as illustrated by $Ins1seq \ldots Ins3seq$ in Fig. 3. The ground-truth for the $i\text{-}th$ instance can be represented as follows:

$$y_i = \left\{\left(c_i, c_i \ldots, c_i\right), \left(q_{i,0}, q_{i,1} \ldots, q_{i,t}\right)\right\}, \quad (6)$$

where c_i is the target class label (0 for text and 1 for $\emptyset$), and $q_{i,t}$ is a vector that specifies the ground-truth of quadrilateral component locations. For the predictions of component sequence with index $\sigma(i)$, we denote the predicted classification scores as:

$$\hat{p}_{(\sigma(i))}\left(c_i\right) = \left\{\hat{p}_{(\sigma(i),0)}\left(c_i\right) \ldots, \hat{p}_{(\sigma(i),t)}\left(c_i\right)\right\}, \quad (7)$$

and the predicted component sequence positions as:

$$\hat{q}_{\sigma(i)} = \left\{\hat{q}_{(\sigma(i),0)}, \hat{q}_{(\sigma(i),1)} \ldots, \hat{q}_{(\sigma(i),t)}\right\}. \quad (8)$$

Using the above notation, we define the matching loss as:

$$\mathcal{L}_{\text{match}}\left(y_i, \hat{y}_{\sigma(i)}\right) = \mathcal{L}_{\text{cls}}\left(c_i, \hat{p}_{\sigma(i)}\left(c_i\right)\right) + \mathcal{L}_{\text{reg}}\left(q_i, \hat{q}_{\sigma(i)}\right), \quad (9)$$

where $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{reg}}$ denote the Focal loss (Lin et al. 2017) and ℓ_1 loss, respectively. Finally, we can find a one-to-one matching between the sequences using the Hungarian algorithm (Kuhn 1955), thereby maintaining the relative positions of predictions for the same instance across different frames.

Decoder Structure. From a tracking perspective, we only need to model the relationships between objects in the first frame and the temporal relationships between different frames. Thus, we adopt a modulated transformer decoder module for parallel component sequence decoding. Specifically, we initially incorporate an intra-frame self-attention module to model the spatial position among component queries within the first frame. Subsequently, we use an inter-frame self-attention module to model the temporal relationship between distinct frames. The outputs are then fed into a multi-scale deformable cross-attention module (Zhu et al. 2021) for interacting features with the flattened output layers of FPN. Finally, these features are individually projected into task-specific space using a feedforward network (FFN). Notably, each point in components allows channel-level interactions via FFN, as the initial components are formed by concatenating the positional encodings of their four vertices along the channel dimension.

3.4 Task Alignment Learning

Current text detection methods overlook the inconsistent prediction issue, i.e., a high classification score with a relatively low localization precision, and vice versa. A position-supervised classification loss could be a good solution. However, efficiently evaluating the localization quality of arbitrary-shaped text detection results is challenging. Thus, we propose a Polygon Monte-Carlo method to quickly and accurately calculate the Polygon Intersection over Union (PIoU) between predicted results and ground-truth.

Polygon Monte-Carlo Method. It comprises three steps. First, given a predicted text instance $\hat{G}$ and corresponding ground-truth G, we adopt TPS-align (Wang et al. 2022b) to sample K points within the text instances. Next, we quantify these sampled points based on a pre-defined tolerance to disregard minor numerical discrepancies. Finally, we count the repeated elements among the sampled points in both the predicted instance $\hat{G}$ and the ground-truth G as the intersection, and the union is calculated by adding these repeated points to the unique points from both instances, as shown in Fig. 5. Notably, all the above steps are organized as matrix operations on the GPU. This enables us to set the sampling number K to a large value, e.g., $10,000$, for an accurate approximation of the PIoU, and allows us to evaluate thousands of localization qualities shortly.

Position-Supervised Classification Loss. Based on the calculated PIoU, we naturally use it to dynamically adjust classification targets for component queries, which smooths the training target and strengthens the correlation between high classification confidence and high-quality prediction. Thus, the position-supervised loss is expressed as:

$$\begin{aligned}\mathcal{L}_{\text{cls}} = &\sum_{i=1}^{N_{\text{pos}}} |s_i^{\alpha} - \hat{c}_i|^{\gamma} \, \text{BCE}(\hat{c}_i, s_i^{\alpha}) \\ &+ \sum_{j=1}^{N_{\text{neg}}} |\hat{c}_j|^{\gamma} \, \text{BCE}(\hat{c}_j, 0),\end{aligned} \tag{10}$$

where $\hat{c}$ is the predicted classification score, s^i is the PIoU between the i-th ground-truth and its corresponding prediction, α is a scaling factor and γ is a focusing parameter.

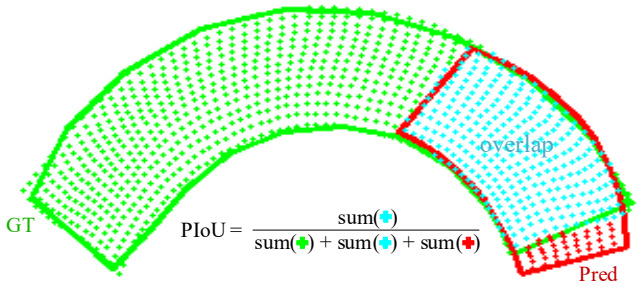


Figure 5: Illustration of our Polygon Monte-Carlo method for calculating the Polygon Intersection over Union (PIoU).

Overall Loss. The full loss function is expressed as:

$$\mathcal{L} = \mathcal{L}_{tci} + \mathcal{L}_{dec}\,, \tag{11}$$

Where $\mathcal{L}_{tci}$ and $\mathcal{L}_{dec}$ refer to the losses for the TCI module and the ERR decoder, respectively. Both of these contain a classification loss $\mathcal{L}_{cls}$ and a regression loss $\mathcal{L}_{reg}$ for classifying and regressing text components. Here, the ℓ_1 loss is applied to $\mathcal{L}_{reg}$ and $\mathcal{L}_{cls}$ denotes our proposed position-supervised classification loss.

4 Experiments

4.1 Datasets

Total-Text (Ch'ng and Chan 2017) includes horizontal, curved, and multi-oriented texts. The dataset contains 1255 training images and 300 test images.

CTW1500 (Liu et al. 2019) is a challenging dataset for long curved text, which consists of 1000 training images and 500 test images. All text instances are annotated by 14-polygon.

ArT (Chng et al. 2019) is a large-scale multi-lingual arbitrary-shaped text detection dataset, which includes 5603 training images and 4563 test images.

MSRA-TD500 (Yao et al. 2012) is a multi-language dataset. It consists of 300 training images and 200 test images.

Synth150K (Liu et al. 2020) contains $150k$ synthetic images, including about one-third of curved texts and two-thirds of multi-oriented texts.

4.2 Implementation Details

When training from scratch, we adopt AdamW with 1×10^{-4} weight decay as the optimizer, and set 16 as batch size, with 500 training epochs for all datasets. For more comprehensive comparisons, we also pre-train our model on a mixture of SynthText-150K, MLT (Nayef et al. 2017) and Total-Text for a total of 5 epochs, and then fine-turn 300 epochs for all datasets. The value of channel c_v is 64. For the ERR decoder, the number of layers is 3, the maximum text instance number n is 100, and the component sequence length t is 6. For the position-supervised loss, the parameters α and γ are set to 0.25 and 2, respectively. For data augmentation, we apply *RandomCrop*, *RandomRotate* and *ColoJitter* to input images. In the testing stage, we set a suitable height for each

Method	Type	Ext	MSRA-TD500			Total-Text			CTW1500			
			R	P	F	R	P	F	R	P	F	FPS
DB (Liao et al. 2020)	Seg	✓	79.2	91.5	84.9	82.5	87.1	84.7	80.2	86.9	83.4	33.5
TextBPN (Zhang et al. 2021b)	Seg	✓	84.5	86.6	85.6	85.2	90.8	87.9	83.6	86.5	85.0	18.1
FSG (Tang et al. 2022)	Seg	✓	84.8	91.6	88.1	85.7	90.7	88.1	82.4	88.1	85.2	–
TextPMs (Zhang et al. 2022)	Seg	✓	87.0	91.0	88.9	87.7	90.0	88.8	83.8	87.8	85.7	14.4
TextBPN++ (Zhang et al. 2023)	Seg	✓	86.8	93.7	90.1	**87.9**	92.4	**90.1**	84.7	88.3	86.5	13.9
CBNet (Zhao et al. 2024)	Seg	✓	84.8	91.1	87.8	82.5	90.1	86.1	81.9	89.0	85.3	–
ABCNet v2 (Liu et al. 2021)	Reg	✓	81.3	89.4	85.2	84.1	89.2	87.0	83.8	85.6	84.7	–
TextDCT (Su et al. 2022)	Reg	–	–	–	–	80.5	85.8	83.0	81.5	84.7	83.1	19.5
TPSNet (Wang et al. 2022b)	Reg	✓	–	–	–	86.8	89.5	88.1	85.7	87.7	86.4	17.9
CT-Net (Shao et al. 2023)	Reg	–	80.4	89.8	84.8	83.6	89.2	86.3	82.7	87.9	85.2	13.6
DPText-DETR (Ye et al. 2023)	Reg	✓	–	–	–	86.4	91.8	89.0	86.2	**91.7**	88.8	14.8
Box2Poly (Chen et al. 2024)	Reg	✓	–	–	–	86.6	90.2	88.4	87.5	88.8	88.1	–
LRANet (Su et al. 2024)	Reg	✓	86.3	92.3	89.2	87.8	90.3	89.0	85.5	89.4	87.4	**37.2**
TextSnake (Long et al. 2018)	CC	✓	73.9	83.2	78.3	74.5	82.7	78.4	85.3	67.9	75.6	–
TextDragon (Feng et al. 2019)	CC	✓	–	–	–	75.7	85.6	80.3	82.8	84.5	83.6	–
DRRG (Zhang et al. 2020)	CC	✓	82.3	88.1	85.1	84.9	86.6	85.8	83.0	86.0	84.5	2.0
ReLaText (Ma et al. 2021)	CC	✓	83.2	90.5	86.7	83.1	84.8	84.0	83.3	86.2	84.8	–
ERRNet	CC	–	86.6	88.2	87.4	86.1	90.1	88.1	85.5	88.9	87.2	31.7
ERRNet	CC	✓	**87.1**	**93.8**	**90.3**	87.3	**92.6**	89.9	**87.9**	91.0	**89.4**	31.5

Table 1: Quantitative detection results on typical benchmarks. “Seg”, “Reg” and “CC” means segmentation-, regression- and connected components-based methods. All listed FPS is measured from a single NVIDIA RTX3090 GPU.

Method	R	P	F
PCR (Dai et al. 2021)	66.1	84.0	74.0
TPSNet (Wang et al. 2022b)	73.3	84.3	78.4
DPText-DETR (Ye et al. 2023)	73.7	83.0	78.1
LRANet (Su et al. 2024) †	74.5	84.0	79.0
ERRNet	**75.5**	**84.1**	**79.6**

Table 2: Performance comparison on ArT. † means the results are from the official website (Chng et al. 2019).

(a) CTW1500 (b) ToTal-Text (c) ArT (d) MSRA-TD500

Figure 6: Qualitative detection results of our ERRNet on datasets CTW1500, Total-Text, ArT, and MSRA-TD500.

dataset while keeping the original aspect ratio. The evaluation metric for the F-measure is IOU@0.5, following (Ye et al. 2023; Chen et al. 2024). All experiments are conducted on 4 NVIDIA RTX3090 GPUs.

4.3 Comparison with State-of-the-art Methods

We compare ERRNet with previous methods on four challenging benchmarks. As shown in Table 1 and 2, ERRNet consistently performs top-tier across the datasets. Compared to segmentation-based methods, ERRNet achieves highly competitive results even when trained from scratch. In particular, on the long curve dataset CTW1500, ERRNet outperforms the recent SOTA segmentation-based method TextBPN++ (Zhang et al. 2023) even without pre-training (87.2% vs. 86.5% in terms of F-measure) and achieves 2.3× faster inference speed. Meanwhile, on the other two datasets in Table 1, the two methods also perform on par. This demonstrates the advantage of the component-level text shape representations over the pixel-level representations.

Compared to regression-based methods, ERRNet also achieves the SOTA accuracy and runs quite efficiently. Specifically, ERRNet outperforms DPText-DETR (Ye et al. 2023) by 0.6% in terms of F-measure and achieves 2.1× faster inference speed on CTW1500. Although ERRNet is a little slower than LRANet (Su et al. 2024), it outperforms LRANet by margins of 1.1%, 0.9% and 2.0% in terms of F-measure on MSRA-TD500, Total-Text, and CTW1500, respectively. This is because LRANet has difficulty in accurately capturing the diverse geometric layouts of the text through a single perception. Moreover, on even large ArT dataset, our ERRNet outperforms DPText-DETR and LRANet by 1.5% and 0.6% in terms of F-measure respectively, again demonstrating the superiority of ERRNet.

In comparison with the previous CC-based methods, ERRNet dramatically surpasses them in both accuracy and speed. Specifically, ERRNet surpasses DRRG (Zhang et al.

Dataset	ERR Decoder	R	P	F
CTW1500	–	82.1	87.5	84.7
CTW1500	✓	**85.5**	**88.9**	**87.2**
Total-Text	–	83.8	88.5	86.1
Total-Text	✓	**86.1**	**90.1**	**88.1**

Table 3: Performance gains of our ERR decoder.

Method	PSC	R	P	F
DPText-DETR (Ye et al. 2023) ‡	–	85.1	88.4	86.7
DPText-DETR (Ye et al. 2023) †	✓	**85.7**	**89.1**	**87.4**
ERRNet (Ours)	–	85.1	89.5	87.3
ERRNet (Ours)	✓	**86.1**	**90.1**	**88.1**

Table 4: Ablation study of our position-supervised classification loss (PSC in short) on Total-Text. ‡ means the results obtained after reproducing the model.

2020) by 4.9% in terms of F-measure on CTW1500 and achieves 16× inference acceleration. This is mainly attributed to our explicit relational reasoning, which not only reduces the difficulty of component content learning but also is post-processing-free. Some detection visualizations are shown in Fig. 6. ERRNet performs well on long, small, and curved text instances.

4.4 Ablation Study

We perform ablation studies on CTW1500 and Total-Text datasets, without pre-training applied by default.

ERR Decoder. We conduct experiments to verify the influence of our ERR decoder. The results are listed in Table 3. The ERR decoder achieves improvements of 2.5% and 2.0% in F-measure on CTW1500 and Total-Text, respectively. Remarkably, the improvements are mainly contributed by recall (3.4% on CTW1500 and 3.7% on Total-Text), mainly because the decoder efficiently refines the cluttered components with low confidence from the TCI module. Moreover, the performance remains competitive even without the decoder, indicating the effectiveness of our text component initialization module.

Position-Supervised Classification Loss. We ablate our position-supervised classification loss on Total-Text to assess its impact. As shown in Table 4, the position-supervised loss improves the F-measure of ERRNet by 0.8%. This demonstrates the effectiveness of dynamically adjusting classification targets based on the prediction quality. To verify the generality of our PAC Loss, we also apply it to DPText-DETR (Ye et al. 2023). It can be seen that embedding this loss improves DPText-DETR by 0.6%, 0.7%, and 0.7% in the precision, recall, and F-measure, respectively.

Explicit Relational Reasoning. We adopt explicit relational reasoning to pre-define the component relationships in each frame. To explore the effectiveness of explicit relational reasoning, we design a variant of implicit relational reasoning, i.e., predicting component associative relationships via a prediction head. Indeed, this variant adopts the perspective of scene graph generation in ReLaText (Ma et al. 2021) to implicitly model component relationships. As shown in Table 5, explicit relational reasoning consistently outperforms the implicit relational reasoning, mainly because pre-defined relationships introduce more prior information, which reduces the difficulty of learning component relationships.

Dataset	Method	R	P	F
CTW1500	IRR	84.9	87.7	86.3
CTW1500	ERR	**85.5**	**88.9**	**87.2**
Total-Text	IRR	85.3	88.8	87.0
Total-Text	ERR	**86.1**	**90.1**	**88.1**

Table 5: Experimental results for different reasoning mothods. ERR and IRR denote explicit relational reasoning and implicit relational reasoning, respectively.

Dataset	Input	R	P	F	FPS
CTW1500	512	85.3	88.1	86.7	**41.7**
CTW1500	608	85.4	88.5	86.9	36.6
CTW1500	704	**85.5**	**88.9**	**87.2**	31.5
Total-Text	608	83.5	86.9	85.2	**34.8**
Total-Text	800	85.9	89.3	87.6	28.9
Total-Text	1000	**86.1**	**90.1**	**88.1**	21.7

Table 6: Performance of ERRNet with different input sizes.

Different Input Image Sizes. To assess the influence of image size and making a proper trade-off between accuracy and speed, we evaluate ERRNet with different short side lengths. As shown in Table 6, ERRNet is robust to changes in image size, with the F-measure fluctuating by only 0.3% on CTW1500 when the size changes from 704 to 608, and by 0.5% on Total-Text when the size changes from 1000 to 800.

5 Conclusion

In this paper, we have presented ERRNet, an accurate and efficient CC-based text detector. For the first time, ERRNet groups text components from a tracking perspective, explicitly defining the relationships between components along both the spatial and temporal dimensions. Therefore, the detection task is elegantly transformed into a tracking task, and post-processing-free prediction is achieved. Additionally, a position-supervised loss is introduced to guide ERRNet towards more consistent task-aligned learning. Experiments conducted on public benchmarks have confirmed the effectiveness of the proposed ERRNet, which shows leading accuracy and top-ranked inference speed. Given its effectiveness and efficiency, we are interested in extending our approach of explicit relational modeling to the scene text understanding task (Liang et al. 2024), i.e., explicitly modeling the relationships between words, sentences, and paragraphs for post-processing-free prediction.

Acknowledgements

This work was supported by National Natural Science Foundation of China (62372170, 62172103)

References

Chen, X.; Wang, D.; Schindler, K.; Sun, M.; Wang, Y.; Savioli, N.; and Meng, L. 2024. Box2Poly: Memory-Efficient Polygon Prediction of Arbitrarily Shaped and Rotated Text. In *AAAI*, volume 38, 1219–1227.

Ch'ng, C. K.; and Chan, C. S. 2017. Total-text: A comprehensive dataset for scene text detection and recognition. In *ICDAR*, volume 1, 935–942.

Chng, C. K.; Liu, Y.; Sun, Y.; Ng, C. C.; Luo, C.; Ni, Z.; Fang, C.; Zhang, S.; Han, J.; Ding, E.; et al. 2019. Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art. In *ICDAR*, 1571–1576.

Dai, P.; Zhang, S.; Zhang, H.; and Cao, X. 2021. Progressive contour regression for arbitrary-shape scene text detection. In *CVPR*, 7393–7402.

Fang, S.; Mao, Z.; Xie, H.; Wang, Y.; Yan, C.; and Zhang, Y. 2022. Abinet++: Autonomous, bidirectional and iterative language modeling for scene text spotting. *IEEE transactions on pattern analysis and machine intelligence*, 45(6): 7123–7141.

Feng, W.; He, W.; Yin, F.; Zhang, X.-Y.; and Liu, C.-L. 2019. Textdragon: An end-to-end framework for arbitrary shaped text spotting. In *CVPR*, 9076–9085.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

He, M.; Liao, M.; Yang, Z.; Zhong, H.; Tang, J.; Cheng, W.; Yao, C.; Wang, Y.; and Bai, X. 2021. MOST: A multi-oriented scene text detector with localization refinement. In *CVPR*, 8813–8822.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2): 83–97.

Liang, M.; Ma, J.-W.; Zhu, X.; Qin, J.; and Yin, X.-C. 2024. LayoutFormer: Hierarchical Text Detection Towards Scene Text Understanding. In *CVPR*, 15665–15674.

Liao, M.; Shi, B.; Bai, X.; Wang, X.; and Liu, W. 2017. Textboxes: A fast text detector with a single deep neural network. In *AAAI*.

Liao, M.; Wan, Z.; Yao, C.; Chen, K.; and Bai, X. 2020. Real-time scene text detection with differentiable binarization. In *AAAI*, 11474–11481.

Liao, M.; Zou, Z.; Wan, Z.; Yao, C.; and Bai, X. 2022. Real-time scene text detection with differentiable binarization and adaptive scale fusion. *TPAMI*, 45(1): 919–931.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *ICCV*, 2980–2988.

Liu, Y.; Chen, H.; Shen, C.; He, T.; Jin, L.; and Wang, L. 2020. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *CVPR*, 9809–9818.

Liu, Y.; Jin, L.; Zhang, S.; Luo, C.; and Zhang, S. 2019. Curved scene text detection via transverse and longitudinal sequence connection. *PR*, 90: 337–345.

Liu, Y.; Shen, C.; Jin, L.; He, T.; Chen, P.; Liu, C.; and Chen, H. 2021. Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. *TPAMI*, 44(11): 8048–8064.

Long, S.; Ruan, J.; Zhang, W.; He, X.; Wu, W.; and Yao, C. 2018. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *ECCV*, 20–36.

Lyu, P.; Yao, C.; Wu, W.; Yan, S.; and Bai, X. 2018. Multi-oriented scene text detection via corner localization and region segmentation. In *CVPR*, 7553–7563.

Ma, C.; Sun, L.; Zhong, Z.; and Huo, Q. 2021. ReLaText: Exploiting visual relationships for arbitrary-shaped scene text detection with graph convolutional networks. *PR*, 111: 107684.

Meng, L.; Li, H.; Chen, B.-C.; Lan, S.; Wu, Z.; Jiang, Y.-G.; and Lim, S.-N. 2022. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12309–12318.

Nayef, N.; Yin, F.; Bizid, I.; Choi, H.; Feng, Y.; Karatzas, D.; Luo, Z.; Pal, U.; Rigaud, C.; Chazalon, J.; et al. 2017. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *ICDAR*, volume 1, 1454–1459.

Shao, Z.; Su, Y.; Zhou, Y.; Meng, F.; Zhu, H.; Liu, B.; and Yao, R. 2023. CT-Net: Arbitrary-Shaped Text Detection via Contour Transformer. *TCSVT*.

Su, Y.; Chen, Z.; Shao, Z.; Du, Y.; Ji, Z.; Bai, J.; Zhou, Y.; and Jiang, Y.-G. 2024. LRANet: Towards Accurate and Efficient Scene Text Detection with Low-Rank Approximation Network. In *AAAI*, 4979–4987.

Su, Y.; Shao, Z.; Zhou, Y.; Meng, F.; Zhu, H.; Liu, B.; and Yao, R. 2022. TextDCT: Arbitrary-Shaped Text Detection via Discrete Cosine Transform Mask. *TMM*.

Sun, L.; Huo, Q.; Jia, W.; and Chen, K. 2015. A robust approach for text detection from natural scene images. *PR*, 48(9): 2906–2920.

Tang, J.; Zhang, W.; Liu, H.; Yang, M.; Jiang, B.; Hu, G.; and Bai, X. 2022. Few Could Be Better Than All: Feature Sampling and Grouping for Scene Text Detection. In *CVPR*, 4563–4572.

Tian, Z.; Huang, W.; He, T.; He, P.; and Qiao, Y. 2016. Detecting text in natural image with connectionist text proposal network. In *ECCV*, 56–72. Springer.

Wang, J.; Chen, D.; Wu, Z.; Luo, C.; Zhou, L.; Zhao, Y.; Xie, Y.; Liu, C.; Jiang, Y.-G.; and Yuan, L. 2022a. Omnivl: One foundation model for image-language and video-language tasks. *Advances in neural information processing systems*, 35: 5696–5710.

Wang, W.; Zhou, Y.; Lv, J.; Wu, D.; Zhao, G.; Jiang, N.; and Wang, W. 2022b. Tpsnet: Reverse thinking of thin plate splines for arbitrary shape scene text representation. In *ACM MM*, 5014–5025.

Wei, J.; Zhang, Y.; Zhou, Y.; Zeng, G.; Qiao, Z.; Guo, Y.; Wu, H.; Wang, H.; and Wang, W. 2022. Textblock: Towards scene text spotting without fine-grained detection. In *Proceedings of the 30th ACM International Conference on Multimedia*, 5892–5902.

Xu, Y.; Wang, Y.; Zhou, W.; Wang, Y.; Yang, Z.; and Bai, X. 2019. Textfield: Learning a deep direction field for irregular scene text detection. *TIP*, 28(11): 5566–5579.

Yang, C.; Chen, M.; Yuan, Y.; and Wang, Q. 2023. Text Growing on Leaf. *TMM*.

Yao, C.; Bai, X.; Liu, W.; Ma, Y.; and Tu, Z. 2012. Detecting texts of arbitrary orientations in natural images. In *CVPR*, 1083–1090.

Ye, M.; Zhang, J.; Zhao, S.; Liu, J.; Du, B.; and Tao, D. 2023. Dptext-detr: Towards better scene text detection with dynamic points in transformer. In *AAAI*, 3241–3249.

Yin, X.-C.; Yin, X.; Huang, K.; and Hao, H.-W. 2013. Robust text detection in natural scene images. *TPAMI*, 36(5): 970–983.

Zhang, B.; Xie, H.; Gao, Z.; and Wang, Y. 2024. Choose What You Need: Disentangled Representation Learning for Scene Text Recognition Removal and Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 28358–28368.

Zhang, C.; Tao, Y.; Du, K.; Ding, W.; Wang, B.; Liu, J.; and Wang, W. 2021a. Character-level street view text spotting based on deep multisegmentation network for smarter autonomous driving. *TAI*, 3(2): 297–308.

Zhang, S.-X.; Yang, C.; Zhu, X.; and Yin, X.-C. 2023. Arbitrary shape text detection via boundary transformer. *TMM*, 26: 1747–1760.

Zhang, S.-X.; Zhu, X.; Chen, L.; Hou, J.-B.; and Yin, X.-C. 2022. Arbitrary shape text detection via segmentation with probability maps. *TPAMI*, 45(3): 2736–2750.

Zhang, S. X.; Zhu, X.; Hou, J. B.; Liu, C.; and Yin, X. C. 2020. Deep Relational Reasoning Graph Network for Arbitrary Shape Text Detection. In *CVPR*.

Zhang, S.-X.; Zhu, X.; Yang, C.; Wang, H.; and Yin, X.-C. 2021b. Adaptive Boundary Proposal Network for Arbitrary Shape Text Detection. In *ICCV*, 1305–1314.

Zhao, X.; Feng, W.; Zhang, Z.; Lv, J.; Zhu, X.; Lin, Z.; Hu, J.; and Shao, J. 2024. CBNet: A Plug-and-Play Network for Segmentation-Based Scene Text Detection. *IJCV*, 1–20.

Zhou, X.; Yao, C.; Wen, H.; Wang, Y.; Zhou, S.; He, W.; and Liang, J. 2017. East: an efficient and accurate scene text detector. In *CVPR*, 5551–5560.

Zhu, X.; Hu, H.; Lin, S.; and Dai, J. 2019. Deformable convnets v2: More deformable, better results. In *CVPR*, 9308–9316.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*.