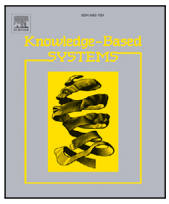
Knowledge-Based Systems 345 (2026) 116091

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



HiTextSpotter: Hierarchical Relation Graph Reasoning Network for Scene Text Spotting

Jialiang Li[1], Canhui Xu *,[1]

College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, 266061, Shandong, China

ARTICLE INFO

Keywords:
Text spotting
Arbitrary shape scene text
Hierarchical relation reasoning
Graph convolutional network

ABSTRACT

As a fundamental task, end-to-end text spotting aims to integrate text detection and recognition into a unified framework, which is known as a challenging task due to text diversity and background noise. Existing text spotting methods employ instance-level local features or holistic global feature representations for text spotting, without capturing intrinsic hierarchical relationships among multiple granularities. From a hybrid perspective, we propose a novel Hierarchical Relation Graph Reasoning Network for Scene Text Spotting (HiTextSpotter) to enrich textual representations at different granularities and capture hierarchical relations using graph structure. HiTextSpotter consists of Global and Local Feature Extraction Network (GLFE), Multi-Granularity Graph Relation Reasoning Network (MGRR), and task prediction head. GLFE primarily extracts multi-grained features, including global-level features from Transformer encoder equipped with deformable attention for holistic field, fine-grained local textual instance-level and component-level features generated by Top-K Bezier center curve proposals. Within MGRR, instance-level graph is designed to learn contextual information and the relationship between instances, and component-level graph is constructed to capture geometric properties for detailed representations. Additionally, an adaptive gated fusion module is introduced to integrate the hierarchical features at different granularities. Finally, the enhanced features are fed into Transformer decoder, where learnable explicit point queries capture text semantics and locations for final results. Extensive experiments on three publicly available scene text benchmarks Total-Text, ICDAR 2015, SCUT-CTW1500, as well as the official map benchmark ICDAR24 MapText, demonstrate that our method achieves superior performance in detection and recognition.

1. Introduction

It is well known that text semantics and relation are of critical importance for understanding the world in various real-world application scenarios, such as autonomous driving, image retrieval, social media analysis, geographical and historical studies [1–8]. Text detection and recognition, termed as text spotter, has been researched in the past decades. Due to text diversity patterns in size, aspect ratio, font style, perspective distortions, and shape [9], text spotting in unconstrained conditions remains challenging. Although numerous early works depend on heuristic rules and specific handcrafted features for text spotting [10–18], the complexity of pipeline and inflexibility limits its application for scene text field [3].

In deep learning research, most classical text spotting methods follow the detection-by-recognition paradigm, which can be roughly summarized as two branches: bottom-up methods and top-down methods. Due to the characteristic of homogeneity and locality, bottom-up methods [19–29] are different from general object detection, and focus on component level granularity, which first generates a dense prediction map for subtext and then groups them together depending on which components belong to the same text instance [3]. These methods are sensitive to sub-text component attribute and text geometric distribution. In contrast, top-down methods [2,9,30–41] predict whole text at instance level, which extracts holistic contextual information for text objects. After detection, both methods require an extra connector such as Region of Interest (RoI) operations for feature alignment. ABCNet [35] introduces Bezier curves into scene text spotting field for arbitrary shape scene text, which improves the detection tightness.

Benefiting from the remarkable improvement of DEtection TRansformer (DETR) in object detection, DETR-like text spotting methods [42–51] complete text detection and recognition simultaneously, and get rid of the additional connector. To cope with the prevalence of arbitrary size scene text, DETR-like framework employs deformable

* Corresponding author.
E-mail address: ccxu09@yeah.net (C. Xu).
[1] Equal Contribution.

https://doi.org/10.1016/j.knosys.2026.116091
Received 8 November 2024; Received in revised form 6 March 2026; Accepted 26 April 2026
Available online 5 May 2026

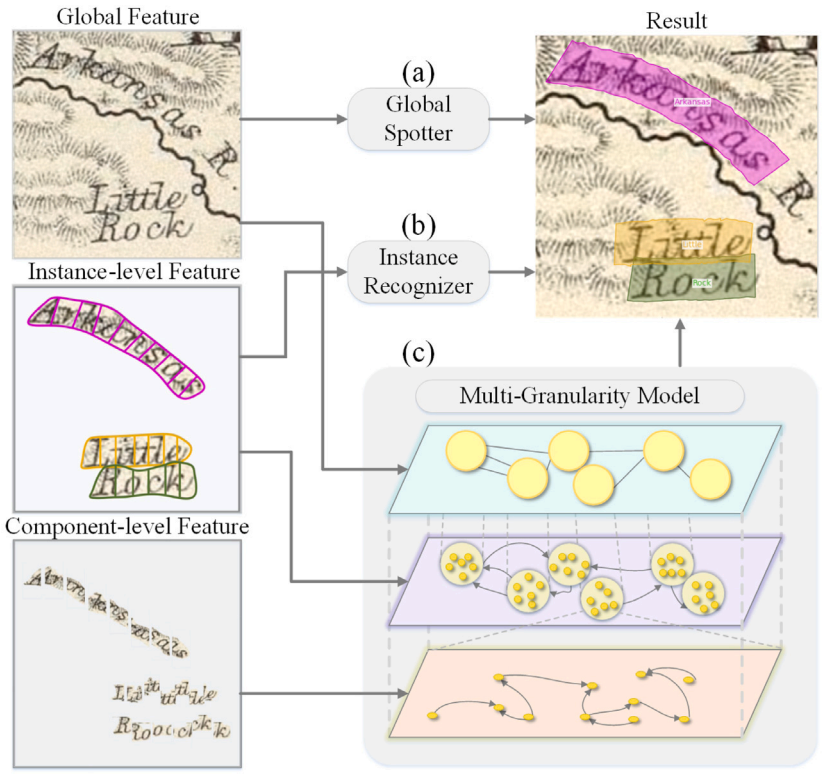


Fig. 1. Illustrations of different text spotting methods: (a) DETR-like methods employ global features from encoder to detect and recognize text instances simultaneously. (b) Classical two-step methods adapt detection-by-recognition paradigm, and utilize instance or subtext component level features for recognition. (c) Multi-granularity method leverages different grain representation to fully explore potential of inherent contextual information among hierarchical structure.

attention mechanism for multi-scale convolutional feature map. Specifically, TESTR [45] employs separate dual-decoders for detection and recognition tasks. For better synergy of two tasks, DeepSolo [46] lets point queries gather semantics and location information, and achieves state-of-the-art performance. Recent DETR-like studies have introduced innovations in text spotting from diverse perspectives [1,47, 50–52], including enhanced synergy mechanisms, coarse-to-fine learning, denoising training strategy, and adaptive reading order modeling. Retrospectively, the aforementioned text spotters predict location and content based on single granularity. To be specific, bottom-up methods locate text at sub-text component granularity and this fine-grained detection unit makes model sensitive to text geometric distribution. Furthermore, component level units can be assembled into text instance level grain. Top-down methods mainly focus on instance level, which can better represent holistic contextual information. And most recent DETR-like methods spot text instances on global image features. Therefore, either utilizing individual or partial context from instances and components or exploiting holistic information from global features is of significance. Based on the above observations, we expect the model to utilize intrinsic structured relations and inherent contextual information across different granularity scales. Compared with single-granularity modeling methods, the multi-granularity modeling architecture demonstrates enhanced capability in systematically investigating hierarchical intrinsic correlations and synergistic interactions across three distinct abstraction levels, namely global semantics, instance representations, and component features. Specifically, we propose to leverage component-level features for composite representations of sub-textual units, while employing instance-level embeddings to establish inter-textual relational patterns through structured semantic modeling. Furthermore, global-level characterization is designed to conduct holistic scene reasoning by integrating contextual information. This hierarchical framework enables different granularities to synergistically compensate for information scarcity at individual granularity levels. Notably, Glass [53] extracts global features cropped from shared backbone, and computes local features individually from high resolution word region images, but it may ignore the inherent interaction between different instances. HGR-Net [54], from a hybrid perspective, exploits textual attributes and interaction at instance and component level, but lacks the capacity of global scene reasoning from attention mechanism. Prompted by above questions, the multi-granularity model should be designed as Fig. 1. Ideally, a hierarchical architecture encompasses three critical components: global contextual awareness, instance relation exploration, and component partial representation. Motivated by DRRG [28], graph convolution network is applied to perform relational reasoning and deduce linkage relationships among hierarchical structures due to its innate advantage for deducing relationships between nodes on the graph.

In this paper, we propose a novel unified Hierarchical Relation Graph Reasoning Network for scene text spotting (HiTextSpotter), which not only extracts comprehensive feature representations at different levels but also employs Multi-Granularity Graph Learner for independent and robust interaction modeling for every distinct layer. HiTextSpotter architecture consists of two key sub-networks: Global and Local Feature Extraction Network (GLFE) and Multi-Granularity Graph Relation Reasoning Network (MGRR), along with a task prediction head. Initially, GLFE employs a shared multi-scale convolutional feature map to refine global-level features from Transformer encoder, focusing on the holistic context. It also extracts local textual instance-level features by top-k Bezier proposals, and fine-grained component-level features to represent partial information. This multi-level visual representation may enrich the understanding of text image structures. Then, MGRR is composed of three critical components: Instance Relation Graph Learning Module, Component Level Graph Learning Module, and Adaptive Gated Fusion Module. Specifically, the instance-level graph is constructed to capture inter-text relationships and contextual properties, while the component-level graph focuses on

intra-text spatial geometric information. For feature fusion, a dynamic adaptive gated fusion mechanism is introduced to facilitate seamless integration of these differing features. Finally, hierarchical features are fed into a single Transformer decoder, which enables learnable explicit queries to learn the position and semantics information for ultimate prediction.

In conclusion, our contributions of this paper can be summarized as follows:

- We propose Hierarchical Relation Graph Reasoning Network, a novel end-to-end trainable network for text spotting, which has explored the efficacy of multi-granularity feature modeling in scene text spotting and has demonstrated superior performance on scene text and map text benchmarks.
- Global and local feature extraction network (GLFE) generates hierarchical features to achieve intrinsic contextual representations. Specifically, hierarchical features, extracted from shared multi-scale feature maps, include overarching global features of the entire image, individual instance-level features of text regions, and fine-grained component-level features representing subtext partial spatial information.
- Multi-Granularity Graph Relation Reasoning Network (MGRR) designs instance relation graph learning module for inter-text relations and component level graph learning module for geometric and spatial properties. Then, features at multiple levels are adaptively and dynamically fused.

2. Related work

Text detection and recognition task has long-standing research history. Research conducted prior to the deep learning era mostly depends on the design of features [3]. In detection, most studies utilize connected component analysis (CCA) [10–14] or sliding window (SW)-based classification [15–18]. CCA-based methods initially refine candidate components by diverse techniques, such as color clustering or extreme region detection. Subsequently, non-textual elements are eliminated by applying either handcrafted rules or classifiers. In SW classification, the input image is traversed by windows of multiple sizes, and each window is then categorized as text segments or not. In recognition, feature-based techniques are employed to recognize text content from text regions. However, due to the limited representation capacity of low-level or mid-level handcrafted image features, these approaches were susceptible to error accumulation and struggled to cope with complex scenarios [3].

With the rapid advancement of deep learning, recent researches have explored the application for text detection and recognition. Text spotting methods are generally categorized into two approaches: classical two-step methods and DETR-like methods. Classical two-step methods [2,20,30–34] follow the detection-by-recognition paradigm [2,9,32–34], and relies solely on the detected text region images from the detection phase to recognize the text content. In contrast, DETR-like methods [44–49] perform both detection and recognition of text instances in a single, integrated process.

Most classical two-step methods employ a connector, such as Region of Interest (RoI)-based operations, to bridge the detection and recognition [20,30–32]. Text-Alignment layer [30] learns a direct mapping between an input image and a set of character sequences or word transcripts. Additionally, ROI-Rotate [32] is proposed to share the feature between detection and recognition. Mask TextSpotter [2] addresses scene text spotting task using a segmentation method, which initially employs Mask R-CNN to identify the text area, followed by a character segmentation module for recognition. Its subsequent version [34] refines the detection method and transitions from region proposal network (RPN) to segmentation proposal network to better deal with text diversity. To cope with the difficulty of arbitrary shapes and orientation, many efforts have been dedicated to overcome this issue. TextSnake [21] describes text instances as a sequence of ordered, overlapping disks centered on symmetric axes. TextDragon [20] uses a series of quadrangles to represent text. ABCNet series [9,35] employs parameterized Bezier curves to model curved text, which not only improves detection accuracy, but also avoid segmentation complexity. Following the lead of ABCNet, most of subsequent methods [45–49] adapt Bezier curves to detect text instances. However, this two-step pipeline may ignore the inter-text relation and interaction [55].

Influenced by the significant progress of Transformer architecture and DEtection TRansformer (DETR) in the domain of object detection, numerous DETR-like methods [45–51] have been proposed to address scene text spotting task. TTS [44] has demonstrated its efficacy in weak annotations situation by integrating an Recurrent Neural Network(RNN) recognizer into Deformable DETR. SPTS [43] uses single-point for detection and follows the sequence prediction paradigm for recognition. To share information between scene text detection and recognition, TESTR [45] utilizes a shared backbone and Transformer encoder but employs separate dual-decoders for each task. Scene text spotting synergy is the mutual interaction and cooperation between text detection and recognition, and recently attracts significant attention. To achieve better synergy between detection and recognition, DeepSolo [46] employs learnable explicit point queries within a single decoder to perform both tasks concurrently. However, some argue that DeepSolo makes all the queries for both detection and recognition task solely relying on implicit synergy and cannot fully realize the potential of these two interactive tasks. ESTextSpotter [47] introduces task-aware queries that are tailored for two distinct tasks, enabling explicit synergy and the modeling of discriminative and interactive features. However, recent DETR-like models implicitly model text instances relation and component attribution.

Multi-Granularity feature extraction methods [49,56–58] has attracted great attention in many deep learning fields, which extract information in different feature Spaces and level. Though previous works have used multi-scale deep convolutional networks like Feature Pyramid Networks (FPN) [59] to learn different scales feature, it struggles to effectively balance the relation between particular and holistic elements. HTS [60] unifies the tasks of scene text spotting and layout analysis, introducing a new dataset and a innovative method. HTS firstly produces Bezier curve polygons of text lines and an affinity matrix for paragraph grouping. Then, line-to-character-to-word module splits the text line into characters and further merges them back into words. Leveraging the promptable and multi-grained segmentation capabilities of Segment Anything Model (SAM), Hi-SAM [61] achieves text segmentation across four hierarchies, including stroke, word, text-line, and paragraph. In domain adaptive detection realm, many works are proposed for domain-invariant feature learning. For separate scene text detection and recognition task, HGR-Net [54] generates textual information at instance and component level, and construct hierarchical graph detect text instances. MGP-STR [56] introduces multi-granularity prediction strategy from natural language processing to integrate linguistic knowledge. Inspired by this, MGN-Net [49] is proposed to integrate visual and semantic information by graph structure and multi-granularity grammatical feature are extracted for semantic presentation. By contrast, just utilizing multi-scale feature for visual presentation is insufficient. In this paper, each input image is regarded as a combination of background and text instances, and text instance is aggregated by a series of partial components.

3. Methodology

Our HiTextSpotter comprises three distinct pipelines, including Global and Local Feature Extraction Network (GLFE), Multi-granularity Graph Relation Reasoning Network (MGRR), and task prediction heads. Furthermore, the proposed architecture is depicted in Fig. 2. In GLFE, after receiving the multi-scale feature maps, integrated feature extraction module extracts hierarchical feature representations. Specifically,

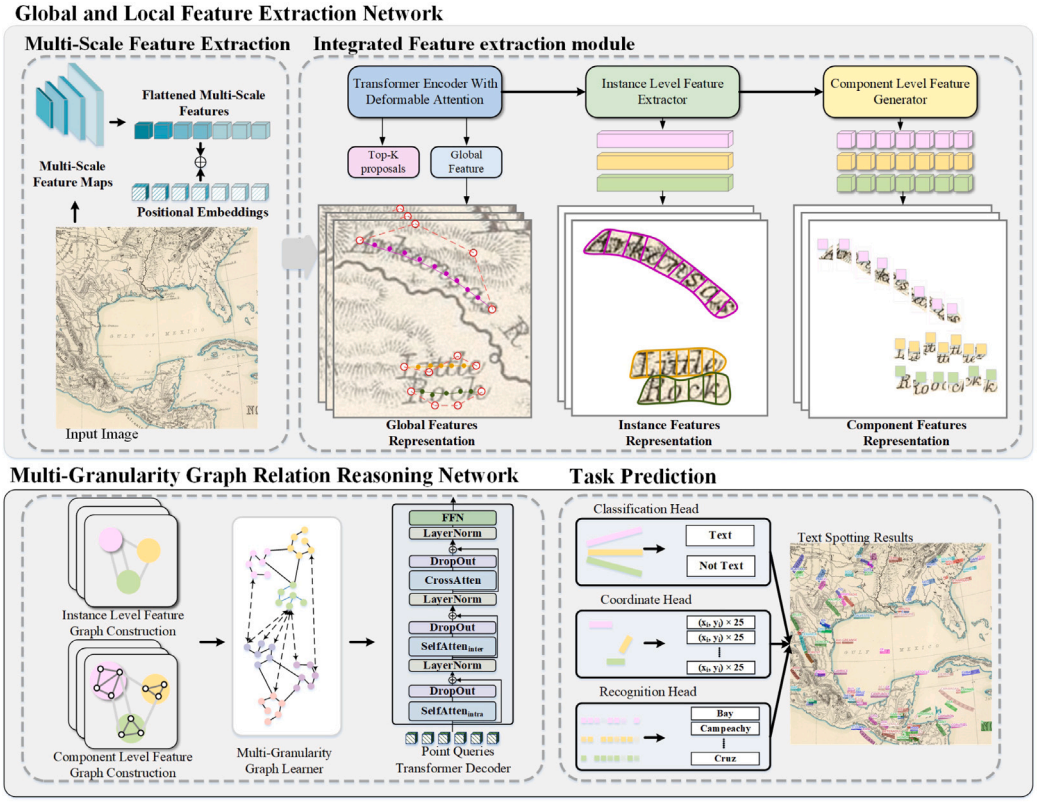


Fig. 2. The framework of our proposed HiTextSpotter method. Global and Local Feature Extraction (GLFE) network utilizes a backbone to refine enhanced multi-scale feature maps. These feature maps, equipped with positional embeddings, are then fed into Transformer encoder with deformable attention to generate top-K proposals and global features for holistic representation. Instance-Level Feature Extractor and Component-Level Feature Generator extract instance and component-level features, respectively. Multi-Granularity Graph Relation Reasoning (MGRR) Network first utilizes these hierarchical contextual representations to construct two graphs that model the relationships and interactions among instance and component levels. Transformer decoder then employs point queries to learn both semantic and positional information. Finally, the prediction heads simultaneously predict the center line, script, and confidence scores from queries.

global representations are encoded by Transformer encoder with multi-scale deformable attention, and instance-level features are extracted depending on top-k proposals. Inspired by TextDragon, Component-level Feature Generator is designed for subtext description. Consequently, MGRR network constructs hierarchical graph structures at component-level to learn spatial contextual information and instance-level to prompt inter-text interaction. After sufficient visual modeling, Transformer decoder enables learnable point queries to learn text location and content explicitly. Finally, these point queries with requisite text semantics and locations can be further decoded to center line, boundary, and script.

3.1. Global and local feature extraction network

For a text image, the image contains both text regions and background, and the text regions can be viewed as a series of aggregated sub-text components. Based on this observation, Global and Local Feature Extraction Network (GLFE) is proposed to generate integrated features at various levels, which comprises multi-scale feature extraction module(MSFE) and integrated feature extraction module (IFEM). MSFE employs ResNet with Feature Pyramid Networks (FPN) to generate multi-scale feature representations, enabling subsequent models to discern both global contextual information and intricate local details. IFEM generates comprehensive features across three levels: global features that encapsulate the holistic context, instance-level features that are tailored to individual textual regions, and component-level features that capture the nuances within text areas.

Multi-scale Feature Extraction Module. After employing the ResNet architecture equipped with a Feature Pyramid Network [59] (FPN) to obtain the convolutional feature maps, the last three feature maps, denoted as res_i, are retained, with each feature map having a shape of $d_i \times h_i \times w_i$. To achieve higher-level abstractions, the last feature map is fed into the Vision Enhancement Module. Within this module, a convolution operation downsamples the feature map, extracting a more abstract representation, denoted as res_{enh}. Here a multi-scale global representation is

$$F_{ms} = \cup_{i \in I} res_i, \tag{1}$$

where I denotes as the set of feature map layers.

Integrated feature extraction module. In Transformer encoder, we employ deformable attention to align with the multi-scale feature representations effectively. Technologically, for multi-scale feature maps F_{ms}, let $\hat{p}_q \in [0,1]^2$ represent the normalized coordinates of the reference point for each query element q, and the update operation is elaborated as

$$\begin{aligned} &MSDeformAttn(Z_q, \hat{p}_q, \{res_i\}_{i \in I}) \\ &= \sum_{m=1}^{M} W_m \left[\sum_{i \in I} \sum_{k=1}^{K} A_{miqk} \cdot W'_m res_i(\Phi_i(\hat{p}_q) + \Delta p_{miqk}) \right], \end{aligned} \tag{2}$$

where m, i, k denote the indices for attention head, input feature level, and sampling point, respectively. Δp_{miqk} and A_{miqk} represent the sampling offset and attention weight of the kth sampling point in ith feature level and the mth attention head. W_m and W'_m are learnable weights. After the Transformer encoder process, the global feature map $F_{global} \in \mathcal{R}^{n_t \times d}$ is generated, where token count can be calculated by $n_t = \sum_{i \in I}(h_i \times w_i)$ and d is the number of hidden dimension. Consequently, using F_{global}, Bezier center curve proposals BP and their corresponding confidence scores S are derived. Specifically, a grid with

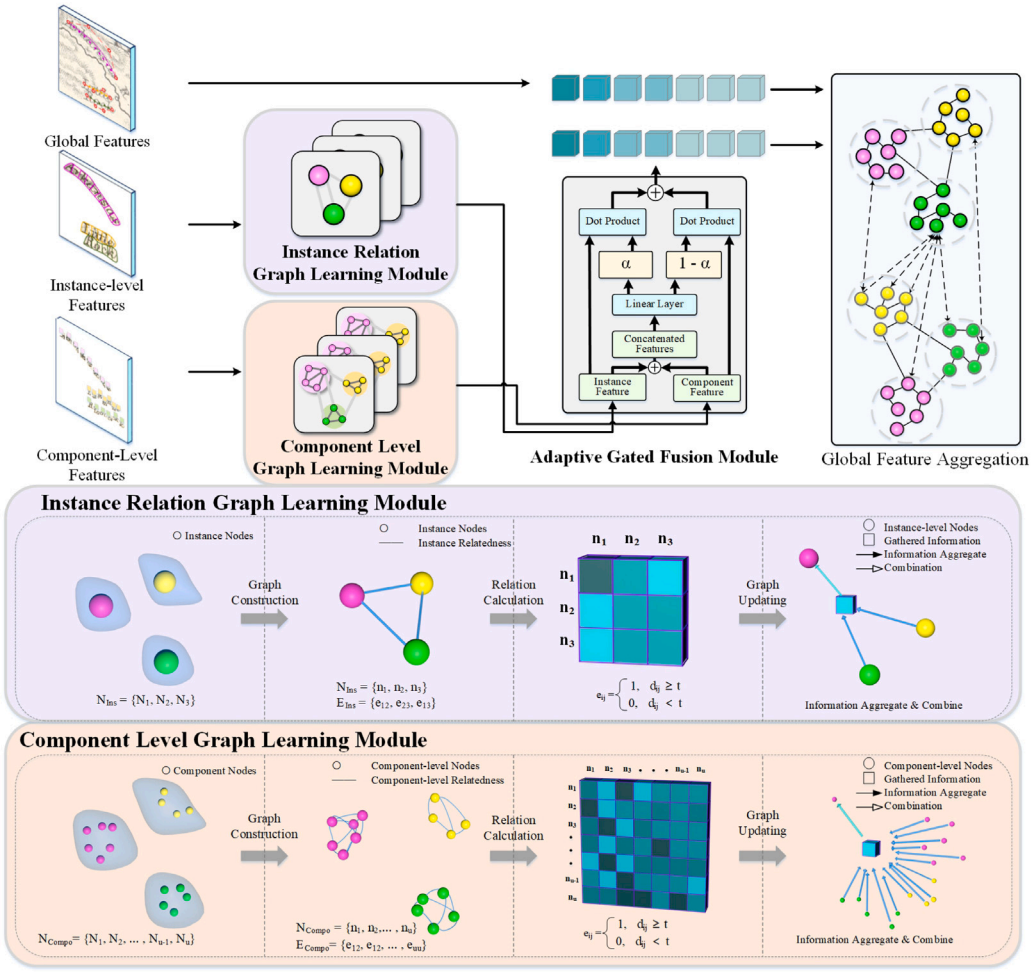


Fig. 3. The framework of our proposed Multi-Granularity Graph Relation Reasoning (MGRR) network. Instance Relation Graph Learning Module constructs inter-text graph to capture contextual interactions among different text instances. Component Level Graph Learning Module builds an intra-text graph for spatial attribute aggregation within a single text instance. Finally, Adaptive gated fusion module fuses the enhanced features from previous modules.

dimensions $n_t \times 2$ is plotted over the entire image, with each grid point (x_i, y_i) corresponding to a location on image, where x_i and y_i are normalized coordinates within the range $[0, 1]$. Subsequently, global features F_{global} are fed into a multi-layer perceptron to predict 4 offsets for each pixel. For every point (x_i, y_i), indexed by i, and for each Bezier control point indexed by $j \in \{1, 2, 3, 4\}$, the offsets are given by

$$(\Delta x_{ij}, \Delta y_{ij}) = MLP(F_{global}). \tag{3}$$

By computing the original grid positions with these offsets, the final positions of the control points for the Bezier curves are:

$$\begin{aligned} BP_{ij} &= (\hat{x_{ij}}, \hat{y_{ij}}) \\ &= \left(\sigma\left(x_{ij} + \sigma^{-1}\left(\Delta x_{ij}\right)\right), \sigma\left(y_{ij} + \sigma^{-1}\left(\Delta y_{ij}\right)\right)\right) \end{aligned} \tag{4}$$

This allows us to obtain n_t proposals BP with the shape $n_t \times 8$, with each proposal corresponding to a feature token. Then, a binary classification linear layer is employed to calculate n_t confident scores which judge whether the proposal is text or not. To refine all the proposals into more plausible candidates, the top-k proposals are selected as the final proposals using the following operation:

$$BP_{top_k} = C(BP) \tag{5}$$

where $BP_{top_k} \in R^{n_i \times 8}$ and C denotes the cropping operation.

Instance level feature extractor. This extractor is designed to refine individual features for each region of text instances. Similar to the process of generating top-k proposals, the extractor identifies the corresponding instance-level features from the global feature map based on the generated proposals. And the process of instance feature extractor can be described as:

$$F_{ins} = C(F_{global}), \tag{6}$$

and $F_{ins} \in \mathcal{R}^{n_i \times d}$

Component level feature Generator. Text instance can be viewed as an ordered series of homogeneous components [3]. Motivated by TextDragon [20], we separate text instances into several sub-text components. Component-level features capture the intricate details within each text region, allowing for fine-grained modeling that provides an accurate geometric description of text instance. We expect that this approach can enhance the ability to distinguish between intra-text characters, ensuring that subtle differences are properly represented. Technically, the shape of received instance level features $F_{ins} = [f_1, f_2, f_3, \ldots, f_{n_i}]$ is $[n_i, d]$. Instance level feature maps are reshaped to F_{compo} of shape $[n_i, n_p, \hat{d_p}]$, where $\hat{d_p} = d/n_p$ and the reshaping operation can be illustrated as

$$F'_{compo} = \mathcal{D}(F_{ins}). \tag{7}$$

Then the feature is flattened at the second dimension and get the feature F'_{compo} of shape $[n_i, n_p, \hat{d_p}]$. Given that the last dimension of F'_{compo} cannot adequately express component level information, we introduce an adapter. To be specific, we first expand the feature dimension from $\hat{d_p}$ to m $(m > \hat{d_p})$. To further encourage the feature transformation capacity, adapter is introduced, which consists of two linear layer and activation function. The first linear layer is to expand feature dimension

and the second is to do feature transformation, as represented by the following equation:

$$F_{compo} = \varphi(F'_{compo} W_e^T + B_e) W_t^T + B_t, \tag{8}$$

where $F_{compo} \in \mathcal{R}$ W_e, W_t, B_e, B_t are parameters of the linear layer and φ represent the activation function.

3.2. Multi-granularity graph relation reasoning network

In the context of daily living, humans instinctively leverage hierarchical relationships to read texts when facing diverse and complex situations. Similar to human reading habits, we anticipate that the model will not only thoroughly examine the holistic content of an image but also focus on the intricate details within text regions to bring better understanding for the spotting task [62]. In this view, we leverage multi-granularity graph structures to jointly model instance-level and component-level feature representations. To be specific, we construct instance relation graph learning module for encoding complete contextual information for inter-text visual connection. And component level graph learning module encodes spatial layout relationship for intra-text from geometric attributes among text components as shown in Fig. 3. As graph can represent any form of data structure [63], the superior modeling capabilities of graph structure can estimate the geometry attributes and inter-text relationship among different level [28,64].

Instance Relation Graph Learning Module. Received the instance-level feature $F_{ins} \in \mathcal{R}^{n_i \times d}$, we construct a dynamic graph $G_{ins} = \mathcal{G}(F_{ins})$. The graph between texts G_{ins} is composed of a collection of nodes $\mathcal{N}_{ins}$ and a set of edges E_{ins}. For node construction, instance-level features $F_{ins} = [x_{p1}, x_{p2}, \dots, x_{pn_i}]$ encompass n_i tokens, which represents n_i text instances. And each token x_{pi} is treated as a graph node n_{pni}, resulting in the instance-level features F_{ins} being represented as a series of unordered nodes $\mathcal{N}_{ins} = \{n_{p1}, n_{p2}, \dots, n_{pn_i}\}$. The inter-text relation evaluation are established by edge linkage. In this phase, we utilize K-NN algorithm to find k nearest neighbors for a node n_i, and establish graph edge e_{ij} from n_i to n_j. Given a node n_i, the distance d_{ij} from n_i to n_j is calculated by formula:

$$d_{ij} = Norm(EdcDis(n_i, n_j)), \tag{9}$$

where $Norm(\cdot)$ normalizes the distances to $[0, 1]$. As the direct quantification of the relationship between text instances, Euclidean distance $EdcDis(n_{pi}, n_{pj})$ is calculated by

$$EdcDis(n_i, n_j) = \sqrt{\sum_{m=1}^{\mu} (n_i - n_j)^2}, \tag{10}$$

μ is the feature dimension. Consequently, the closest k nodes are selected as its neighbors, denoted as $\mathcal{N}_{ins}(n_i)$. K edges E_i are established, where each edge e_{ij} connects node n_i to node n_j for all neighbors n_j in $\mathcal{N}_{ins}(n_i)$. This process could generate overall edges $E_{ins} = \{E_1, E_2, \dots, E_{n_i}\}$ for all nodes $\mathcal{N}_{ins}$. Consequently, the graph structure $G_{ins} = (\mathcal{N}_{ins}, E_{ins})$ is constructed, summarizing the process as $G_{ins} = \mathcal{G}(F_{ins})$. This process establishes the cornerstone for the reasoning to relationship between instances. The general graph construction processing is depicted in Algorithm 1.

The generated graph $G_{ins} = \{\mathcal{N}_{ins}, E_{ins}\}$ is used to explore global text instance context information. The updated text instance could gather context from both itself and its relevant text instances. The mainstream update process for graph convolutional network involves two key steps: aggregation and combination. In aggregation, as a node, every text instance aggregates information from its adjacent text nodes based on the established edges E_{ins}. During combination phase, every text instance node updates its feature representation by integrating information from the aggregated data along with its own features. The detailed operation details are as follows:

$$G'_{ins} = F(G_{ins}, W_{ins}) \tag{11}$$

Algorithm 1 The process of graph construction.

Input: $tokens \in \mathcal{R}^{n \times d}$, k(number of neighbors)
Output: $nodes, edges$

```
nodes = token_to_node(tokens)
edges = []
for token in tokens do
    neighbors = KNN(token, other_tokens, k)
    edges_current_node = build_edge(token, neighbors)
    edges.append(edges_current_node)
end for
return (nodes, edges)
```

$$= Comb(Agg(\mathcal{N}_{ins}, E_{ins}, W_{ains}), N_{ins}, W_{cins}), \tag{12}$$

where $Comb$ and Agg are aggregation and combination operations at graph level. W_{ins} is learnable weight. W_{ains} and W_{cins} are for aggregating from neighbors and for combining aggregated features with current node.

For a node, the process of aggregation and combination is similar to the above description of the entire graph. The node x_i first aggregates features from all its neighbors $\mathcal{N}(x_i)$ in the aggregation step, and then it combines the aggregated features with its own in the combination step. As for a single node x_i, the detailed formulation is as follows:

$$x'_i = comb(x_i, agg(x_i, E_i, \mathcal{N}(x_i), W_a), W_c), \tag{13}$$

where $comb$ and agg are aggregation and combination operations for a single graph node.

GIN (Graph Isomorphism Network) [65] is designed based on graph isomorphism theory, aiming to approximate the expressive power of the Weisfeiler-Lehman graph isomorphism test. This means that GIN can better capture the structural information of graphs, especially for complex structures with similar node features. GIN can more accurately distinguish different graph structures. This characteristic is precisely suitable for relational reasoning of scene text among inter-text and intra-text. The specific implementation is as follows:

$$agg(\cdot) = x''_i = \sum_{j \in \mathcal{N}(i)} x_j, \tag{14}$$

$$comb(\cdot) = x'_i = \epsilon x_i + (1 + \epsilon) x''_i. \tag{15}$$

To this end, the graph learning process could be denoted as $G' = GL(G)$.

Component Level Graph Learning Module. Intra-text graph learning process is similar to the update process of above instance-level graph. To establish spatial-related linkages among text components within single text instance, intra-text graph is generated by $G_{compo} = \mathcal{G}(F_{compo})$, and it contains two parts $(\mathcal{N}_{compo}, E_{compo})$. During construction, the nodes of intra-text graph are converted by geometric representation F_{compo}, and edge will be linked for text spatial distribution. During inference, visual information within a text region is aggregated based on their relative proximity relationships. This enables textual component to integrate geometric and semantic information from its adjacent elements for reasoning, thereby facilitating information inference and deduction:

$$G'_{compo} = F(G_{compo}, W_{compo}) \tag{16}$$

$$= Comb(Agg(N_{compo}, E_{compo}, W_{acompo}), N_{compo}, W_{ccompo}). \tag{17}$$

Adaptive gated fusion module. Merging the enhanced component-level and instance-level features in a harmonious way remains a challenging task. After numerous attempts, we introduce a novel and effective approach known as the Adaptive Gated Fusion (AGF) Module. Since the features at both the instance and component levels share the same shape, we first concatenate them into a single feature vector.

Subsequently, a weight transformation matrix is applied to this concatenated vector to compute the attention weights, as illustrated in the following equation:

$$\alpha = \sigma\left(\left[F_{ins}, F_{compo}\right], W_f\right). \tag{18}$$

Then, we consider α and $(1-\alpha)$ as the weighting factors for features at component-level and instance-level, respectively. We then aggregate these features to obtain the fused instance-level representation.

$$F_{merged} = \alpha F_{compo} + (1-\alpha) F_{ins}. \tag{19}$$

In contrast to the extracting process of instance level features, we reintegrate the fused instance-level features back into their corresponding positions within the global feature map, thereby achieving multi-granularity feature fusion.

Transformer Decoder. The character sequences are depicted as a series of ordered points and model them by learnable explicit point queries [46,66]. Our compound query integrates both positional and semantic embeddings. When creating the positional query for the selected proposal BP_{topk}, n points are uniformly sampled for each proposal BP_i. Subsequently, we generate all normalized point coordinates $Coords \in \mathrm{R}^{K\times N\times 2}$, which are then fed into positional encoding layer and Multi-layer Perceptron (MLP) layer to generate the positional queries P_q:

$$P_q = MLP(PE(Coords)). \tag{20}$$

Besides, we adopt a series of learnable embeddings as semantic queries C_q. The composite queries Q are generated: $Q = P_q + C_q$.

In Transformer decoder layer, we model both positional and semantic information using learnable point queries. The detailed updating process is depicted in Fig. 2. Initially, intra-group self-attention is employed to refine the relationship between point queries within a single text instance. Subsequently, we utilize inter-group self-attention to uncover relationships among different text instances. After updating the queries, these composite queries are sent to the deformable cross-attention mechanism to extract information from hierarchical visual features generated by the GLFE network. Finally, a three-layer Multi-layer Perceptron (MLP) head is applied to predict the offsets and update the point coordinates.

Task Prediction. Upon receiving the updated queries encoded requisite text semantics and locations, four parallel prediction heads are employed to address distinct tasks: (1) Instance Classification: A linear layer is utilized to predict a binary classification, determining whether the object is text or background. A threshold is established at a fixed value, which corresponds to the average of N scores. (2) Character Classification: Linear layers are similarly used to classify whether the character is a text character or background. Since the object queries are derived from Bezier points, each query represents either a character or background. (3) Center Curve Offsets: A three-layer MLP is employed to predict the coordinate offsets relative to the original coordinates generated by the encoder. (4) Boundary Offsets: Analogous to the center curve offsets, we utilize a three-layer MLP to predict the offsets relative to the ground truth points.

Loss. As for the set prediction problem that uses a fixed number of outputs to match the ground truths, HiTextSpotter follows most of the DETR-like methods [45,46], and utilize the Hungarian algorithm to perform pairwise matching and minimize the matching cost C_{match} between prediction Y and ground truth $\hat{Y}$:

$$\arg\min_{\varphi} \sum_{g=0}^{G-1} C_{match}\left(Y^{(g)}, \hat{Y}^{(\varphi(g))}\right), \tag{21}$$

where G is the number of ground truth instances per image. The matching cost per instance consist of the difference between positive and negative for classification and $L1$ distances.

The total loss function consists of two parts: encoder loss and decoder loss. The encoder loss $\mathcal{L}_{enc}$ is an auxiliary loss for intermediate supervision, which propagates back directly to the encoder for more accurate proposal generation and faster convergence, and decoder loss $\mathcal{L}_{dec}$ provides multi-task learning losses for the final prediction. In detail, encoder loss has two parts: classification loss and coordination loss, which are denoted as

$$\mathcal{L}_{enc} = \sum_i \left(\lambda_{cls}\mathcal{L}_{cls}^{(i)} + \lambda_{coord}\mathcal{L}_{coord}^{(i)}\right). \tag{22}$$

As for hyper-parameters, λ_{cls}, λ_{text}, λ_{coord}, λ_{bd} are set to 1, 0.5, 1, 0.5, respectively. These hyper-parameter weights are applied to all decoder layers.

λ_{cls} and λ_{coord} are the hyper-parameters. For more accurate Bezier center curve proposals, we calculate L1 loss across N uniformly sampled points along the curve, rather than relying on the four Bezier control points solely.

The decoder loss function consists of four parts: classification loss, text classification loss, center curve coordination loss, and boundary loss, which can be formulated as:

$$\mathcal{L}_{dec} = \sum_k \left(\lambda_{cls}\mathcal{L}_{cls}^{(k)} + \lambda_{text}\mathcal{L}_{text}^{(k)} + \lambda_{coord}\mathcal{L}_{coord}^{(k)} + \lambda_{bd}\mathcal{L}_{bd}^{(k)}\right). \tag{23}$$

Instance classification loss $\mathcal{L}_{cls}$ enables the model to determine whether the object is text or background. And focal loss is utilized to solve the foreground–background class imbalance problem. For the jth query, the focal loss for instance classification is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{cls}}^{(j)} = &- \mathbb{I}_{\{j\in \mathrm{Im}(\sigma)\}}\alpha\left(1-\hat{b}^{(j)}\right)^{\gamma}\log\left(\hat{b}^{(j)}\right)\\ &- \mathbb{I}_{\{j\notin \mathrm{Im}(\sigma)\}}(1-\alpha)\left(\hat{b}^{(j)}\right)^{\gamma}\log\left(1-\hat{b}^{(j)}\right),\end{aligned} \tag{24}$$

where $\hat{b}^{(j)}$ is the probability for the text-only instance class, $\mathbb{I}$ is the indicator function, and $\mathrm{Im}(\sigma)$ is the image of the mapping σ. Text classification loss $\mathcal{L}_{text}$ is employed to minimize the discrepancy between predicted script and ground truth text, and CTC loss is used to avoid the length mismatch between the two scripts:

$$\mathcal{L}_{text}^{(j)} = \mathbb{I}_{\{j\in \mathrm{Im}(\sigma)\}} CTC(t^{\sigma^{-1}(j)}, \hat{t}^{(j)}) \tag{25}$$

Coordinate loss $\mathcal{L}_{coord}$ and boundary loss $\mathcal{L}_{bd}$ employ $L1$ loss functions for supervising point coordinates on the center curve and bounding box, respectively, aiming to better localize text instance and fit the shape of text region:

$$\mathcal{L}_{coord}^{(j)} = \mathbb{I}_{\{j\in \mathrm{Im}(\sigma)\}} \sum_{i=1}^{N} \left\| p_i^{(\sigma^{-1}(j))} - \hat{p}_i^{(j)} \right\| \tag{26}$$

$$\begin{aligned}\mathcal{L}_{bd}^{(j)} = &\mathbb{I}_{\{j\in \mathrm{Im}(\varphi)\}} \sum_{i=1}^{N} \left(\left\| \mathrm{top}_i^{(\varphi^{-1}(j))} - \hat{top}_i^{(j)} \right\| \right.\\ &\left. + \left\| bot_i^{(\varphi^{-1}(j))} - \hat{bot}_i^{(j)} \right\| \right)\end{aligned} \tag{27}$$

where p, top, bot refer to the coordinates of center curve, top boundary, and bottom boundary, respectively.

In conclusion, the total loss function can be formulated as

$$\mathcal{L} = \mathcal{L}_{enc} + \mathcal{L}_{dec}. \tag{28}$$

4. Experiments

A comprehensive series of experiments was conducted to assess the effectiveness of our proposed method. These experiments were performed on a historical map dataset ICDAR24 MapText [8] and three scene text datasets, including the arbitrarily-oriented dataset ICDAR 2015(IC15) [67], and two arbitrary-shape datasets, SCUT-CTW1500 [68] and Total-Text [69].

4.1. Public datasets

ICDAR 2015 [67] focuses on incidental scene text and consists of 1500 images captured with wearable cameras, including 1000 images for training and 500 for testing. Each image in this dataset contains vocabularies of up to 100 words, comprising the words present in the image along with distractors.

SCUT-CTW1500 [68] is another vital scene text spotting dataset released by [68], because it contains numerous horizontal and multi-oriented texts. It includes 10,751 text annotations across 1500 images: 1000 for training and 500 for testing. The dataset features a diverse distribution of images, encompassing indoor and outdoor scenes, born-digital text, blurred images, and perspectives with distortion.

Total-Text [69] is a widely used scene text spotting dataset characterized by images containing more than 3 different text orientations: horizontal, multi-oriented, and curved. This dataset comprises 1255 images in the training set and 300 images in test set, with each image labeled at the word level using polygon annotations.

ICDAR 2024 MapText Dataset is a new dataset released for the ICDAR MapText 2024 Robust Reading Competition [8]. Unlike traditional scene text detection tasks, extracting text from historical maps presents unique challenges, such as dense text regions, rotated and curved text, and widely spaced characters. Furthermore, a map with dimensions of 2000×2000 may contain between 100 and 300 words, presenting a considerable challenge.

4.2. Implementation details

This model is implemented using the PyTorch 1.7 framework. Unless otherwise specified, ResNet-50 with FPN is used as basic backbone. Regarding model settings, both encoder and decoder consist of 6 layers, with the number of heads and sampling points set to 8 and 4, respectively, in the deformable attention mechanism. For proposals, the number of top-k proposals is set to 100, and each proposal has 25 sampled points generated by Bezier control points. The model predicts 37 character classes for the Total-Text and ICDAR 2015 datasets and 96 character classes for the CTW1500 and MapText datasets. During the training process, the AdamW optimizer is utilized for backpropagation.

Data Augmentation. During the training process, we employ seven data augmentation methods to enhance the model's robustness. Specifically: (1) Random Rotation: The image is randomly rotated within the range of −45 to +45 degrees. (2) Random Crop: This technique crops the image randomly to a size that is a relative fraction of the original image. (3) Resize Shortest Edge: The image is resized so that the shortest edge meets one of the specified lengths in the list, while ensuring the largest dimension does not exceed 1600 pixels. (4) Random Contrast: The contrast of the image is adjusted randomly. (5) Random Brightness: Similar to Random Contrast, this method randomly alters the brightness of the image. (6) Random Lighting: The lighting conditions of the image are adjusted randomly. (7) Random Saturation: The saturation of the image is changed randomly.

Training settings. In the pre-training process, the model is trained for 435K iterations with an initial learning rate of 10^{-4}, which is decayed by a factor of 10 after 375K iterations. The training is conducted on various datasets, including Synthetic Text, Total Text, MLT (Multi-Language Text), ICDAR 2013, and ICDAR 2015. Most experiments are performed on 8 NVIDIA TITAN Xp GPUs using an Ubuntu 16.04 workstation equipped with an Intel(R) Xeon(R) 2.20 GHz CPU and 64 GB of RAM.

4.3. Comparison with state-of-the-art methods

In this section, we compare our model with state-of-the-art methods across various benchmarks to address the challenges in both scene text and map text. To ensure fairness, all methods are based on the ResNet architecture as their foundational backbone, unless specifically noted otherwise. For the detection task, the acronyms P, R, and F1 correspond to Precision, Recall, and F1-Score, respectively. In end-to-end text spotting task, the term 'None' signifies results obtained without any lexicon, and 'Full' denotes outcomes derived from a lexicon that includes the entire vocabulary of the test dataset.

Result on arbitrarily-shaped datasets. As mentioned previously, Total-Text and CTW1500 datasets focus on curved text instances. To evaluate the performance of our model on arbitrarily-shaped benchmarks, we conduct experiments using Total-Text and CTW1500. The detection results are summarized in Table 1. Our model achieves the highest F1-score of 87.5% on CTW-1500 benchmark, outperforming TESTR by 1.2%. On Total-Text, HiTextSpotter records an F1-score of 88.7%, highlighting the model's robust detection capability for curved text. Notably, our model demonstrates a higher precision compared to other models evaluated. For the precision (P) metric, HiTextSpotter achieves scores of 91.6% and 92.7% for the CTW1500 and Total-Text datasets, respectively. This performance can be attributed to the model's strong ability to capture intricate details, while simultaneously addressing both holistic and local aspects of text instances. The end-to-end recognition results on the Total-Text and CTW1500 datasets are presented in Table 2. HiTextSpotter consistently outperforms other methods across all lexicon settings on Total-Text dataset. Particularly, in lexicon-free setting, our model achieves an H-mean of 81.7%, exceeding TESTR by 8.4% and DeepSolo by 2.0%. A comparison with TESTR results illustrates that our model, utilizing a single decoder for modeling, achieves a synergy between detection and recognition tasks. This synergy results in significant improvements in recognition performance. Additionally, the comparative results with DeepSolo suggest that enhanced attention to component-level features could further improve the model's ability to recognize textual details.

Result on arbitrarily-oriented datasets. In terms of the arbitrarily-oriented benchmark ICDAR 2015, HiTextSpotter also demonstrates commendable performance. For detection task, HiTextSpotter achieves the highest F1 score of 90.7%, surpassing TESTR and DeepSolo by 0.7%. Moreover, the superior precision metric of 93.7% surpasses TESTR and DeepSolo by 3.4% and 0.9%, respectively. For end-to-end recognition task, our model achieves the highest recognition scores of 86.9% and 82.3% on ICDAR 2015 benchmark under the strong and weak lexicon settings. This further substantiates that augmenting detail features via graph representation appreciably enhances performance across recognition tasks.

4.4. Ablation studies

To investigate whether Graph Neural Networks can learn from multiple level features and multi-granularity graphs, we have conducted ablation experiments on Total-Text to explore the effectiveness of the modules we propose.

(1) With or Without Component Level Graph Learning Module(Com. Graph). The baseline model is implemented using milestone approach of DeepSolo, which directly feeds global features into Transformer decoder. As shown in Table 3, the model incorporating Component Level Graph Learning module demonstrates a slight improvement for the detection task, achieving a 0.2% increase in H-mean index. For the recognition task, the model attains an accuracy of 80.1% without lexicon, surpassing the baseline by 0.4%. This enhancement can be attributed to the additional modeling of component-level features, but the mismatching between component and holistic ones may exists.

(2) With or Without Adaptive Gated Fusion Module(AGF). In our initial attempt to incorporate component-level features into global features, we directly added the features processed by Component Level Graph Learning Module to the original features, but we recognized that such a straightforward addition might disrupt the original information. Inspired by this [90], we employ Adaptive Gated Fusion Module to mitigate conflicts between different feature spaces. As can be seen in

Table 1
Detection results comparing HiTextSpotter with thirty state-of-the-art methods on SCUT-CTW1500, Total-Text, and ICDAR 2015 datasets. The symbol '—' indicates that the original paper did not include an evaluation for that dataset.

Method	Backbone	CTW-1500			Total-Text			ICDAR 2015		
		R	P	F1	R	P	F1	R	P	F1
SegLink [23]	VGG-16	48.4	38.3	42.8	30.3	23.8	36.7	76.8	73.1	75.0
TextSnake [21]	FCN+FPN	85.3	67.9	75.6	74.5	82.7	78.4	73.9	83.2	78.3
LSAE [26]	ResNet-50	77.8	82.7	80.1	–	–	–	–	–	–
TextDragon [20]	VGG-16	82.8	84.5	83.6	75.7	85.6	80.3	–	–	–
TextField [22]	VGG-16	79.8	83.0	81.4	79.9	81.2	80.6	80.1	84.3	82.4
SegLink++ [19]	VGG-16	79.8	82.8	81.3	80.9	82.1	81.5	80.3	83.7	82.0
CRAFT [24]	ResNet-50	81.1	86.0	83.5	79.9	87.6	83.6	84.3	89.8	86.9
PAN [25]	ResNet-18	81.2	86.4	83.7	81.0	89.3	85.0	81.9	84.0	82.9
DRRG [28]	VGG-16+FPN	83.0	85.9	84.5	84.9	86.5	85.9	84.7	88.5	86.6
CentripetalText [29]	ResNet-50	79.9	88.3	83.9	82.5	90.5	86.3	–	–	–
DMPNet [70]	ResNet-50	61.7	63.9	62.7	–	–	–	–	–	–
CTD+TLOC [71]	ResNet-50	69.8	77.4	73.4	71.0	74.0	73.0	77.1	84.5	80.6
EAST [72]	ResNet-50	49.7	78.7	60.4	50.0	36.2	42.0	78.3	83.3	80.7
Textboxes++ [73]	ResNet-50	–	–	–	–	–	–	78.5	87.8	82.9
RRPN [74]	ResNet-50	–	–	–	–	–	–	77.0	84.0	80.0
ATTR [26]	ResNet-50	80.2	80.1	80.1	76.2	80.9	78.5	83.3	90.4	86.8
MSR [75]	ResNet-50	79.0	84.1	81.5	73.0	85.2	78.6	–	–	–
PSENet-1s [76]	ResNet-50	79.7	84.8	82.2	78.0	84.0	80.9	85.5	88.7	87.1
LOMO [77]	ResNet-50	76.5	85.7	80.8	79.3	87.6	83.3	83.5	91.3	87.2
Mask TTD [78]	ResNet-50	79.0	79.7	79.4	74.5	79.1	76.7	87.6	86.6	87.1
Counter-Net [79]	ResNet-50	**84.1**	83.7	83.9	83.9	86.9	85.4	86.1	87.6	86.9
DB [27]	ResNet-50	80.2	86.9	83.4	82.5	87.1	84.7	82.7	88.2	85.4
Mask TextSpotter [2]	ResNet-50	–	–	–	82.4	88.3	85.2	87.3	86.6	87.0
Boundary [80]	ResNet-50	–	–	–	85.0	88.9	87.0	–	–	–
PCR [81]	ResNet-50	82.3	87.2	84.7	82.0	88.5	85.2	–	–	–
FCE-Net [82]	ResNet-50	83.4	87.6	85.5	82.5	89.3	85.8	–	–	–
TBTD [83]	ResNet-50	–	–	–	–	–	–	78.3	89.8	83.7
ACE [84]	ResNet-50	–	–	–	–	–	–	82.6	82.1	82.4
TextBPN [85]	ResNet-50	80.6	87.7	84.0	83.3	90.8	86.9	–	–	–
GLASS [53]	ResNet-50				**85.5**	90.8	88.1			
SwinTextSpotter [42]	Swin-T-FPN	–	–	–	–	–	88.0	–	–	–
ABCNet v2 [35]	ResNet-50-FPN	83.8	85.6	84.7	84.1	90.2	87.0	86.0	90.4	88.1
TESTR [45]	ResNet-50	83.1	89.7	86.3	83.7	92.8	88.0	**89.7**	90.3	90.0
DeepSolo [46]	ResNet-50	–	–	–	82.1	93.1	87.3	87.4	92.8	90.0
HiTextSpotter (ours)	ResNet-50	83.7	**91.6**	**87.5**	85.1	**92.7**	**88.7**	87.9	**93.7**	**90.7**

Table 2
End-to-End text spotting results on Total-Text, SCUT-CTW1500 and ICDAR 2015.

Method	Total-Text		CTW-1500		ICDAR 2015		
	None	Full	None	Full	S	W	G
Mask TextSpotter [2]	65.3	77.4	–	–	83.0	77.7	73.5
FOTS [32]	–	–	21.1	39.7	83.6	79.1	65.3
TextDragon [20]	48.8	74.8	39.7	72.4	82.5	78.3	65.2
Text Perceptron [86]	69.7	78.3	57.0	–	80.5	76.6	65.1
ABCNet [9]	64.2	75.7	45.2	74.1	–	–	–
Mask TextSpotter v3 [34]	71.2	78.4	–	–	83.3	78.1	74.2
PGNet [36]	63.1	–	–	–	83.3	78.3	63.5
MANGO [38]	72.9	83.6	58.9	78.7	81.8	78.9	67.3
ABCNet v2 [35]	70.4	78.1	57.5	77.2	82.7	78.5	73.0
PAN++ [87]	68.6	78.6	–	–	82.7	78.2	69.2
Boundary [80]	66.2	78.4	46.1	73.0	82.5	77.4	71.7
SwinTextSpotter [42]	74.3	84.1	51.8	77.0	83.9	77.3	70.5
SRSTS [88]	78.8	86.3	–	–	85.6	81.7	74.5
TPSNet [89]	78.5	84.1	60.5	80.1	–	–	–
GLASS [53]	79.9	86.2	–	–	84.7	80.1	76.3
TESTR [45]	73.3	83.9	56.0	81.5	85.2	79.4	73.6
TTS [44]	78.2	86.3	–	–	85.2	81.7	**77.4**
ABINet++ [90]	77.6	84.5	60.2	80.3	84.1	80.4	75.4
DeepSolo [46]	79.7	87.0	**64.2**	81.4	86.8	81.9	76.9
HiTextSpotter (ours)	**81.7**	**87.7**	63.6	**82.1**	**86.9**	**82.3**	76.2

Table 3, not only does the model show significant improvement over the baseline model, but it also achieves a 1.6% enhancement in the recognition task without a lexicon compared to the model equipped with Component Level Graph Learning module, attaining an H-mean of 81.3%.

(3) With or Without Instance Relation Graph Learning Module(Ins. Graph). After experimenting with component-level features, we are confident that graph convolutional networks can be utilized to process features within Transformer framework. Consequently, we employ Instance Relation Graph Learning Module to conduct graph-based learning on instance-level features. In Table 3, the model achieves an h-mean of 88.7% in detection, marking a 1.4% improvement, with a particularly notable 3% enhancement in the recall metric.

(4) Different types of graph neural network. In Table 4, we conduct ablation experiments to compare the performance impact of different

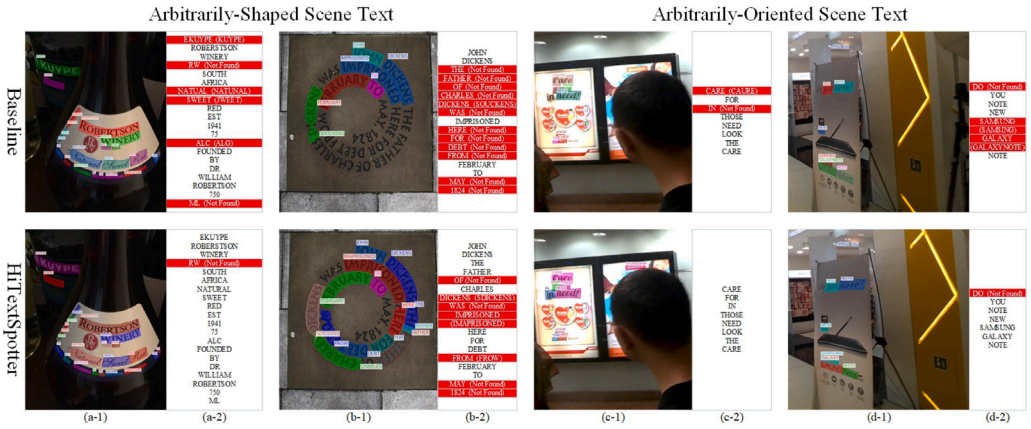


Fig. 4. Qualitative scene text spotting results of baseline and HiTextSpotter. We present text scripts at the right of image. Texts in black font indicate correct recognition results, while text highlighted with a red background denotes recognition errors. The text 'Not Found' denotes instances that are not detected by the model.

Table 3
Ablation studies on Total-Text. "GlobalF" means global features. "Ins. Graph" means instance relation graph learning module. "Com. Graph" means component level graph learning module. "AGF" means adaptive gated fusion module. To ensure as much fairness as possible, we use the same configuration in the ablation experiments.

Baseline	Modules				Detection			End-to-End			
	GlobalF	Ins. Graph	Com. Graph	AGF	P	R	F1	diff	None	diff	Full
Baseline	✓				93.1	82.1	87.3	-	79.7	–	87.0
Baseline+	✓	✓			92.3	83.4	87.6	+0.3	80.4	+0.7	86.6
Baseline+	✓		✓		92.7	82.9	87.5	+0.2	80.1	+0.4	86.8
Baseline+	✓		✓	✓	**93.7**	82.7	87.9	+0.6	81.3	+1.6	87.1
Baseline+	✓	✓	✓	✓	92.7	**85.1**	**88.7**	+1.4	**81.7**	+2.0	**87.7**

Table 4
Ablation studies on different types of graph convolution in Total-Text benchmark.

GNN	Detection			End-to-End	
	P	R	F1	None	Full
EdgeConv(TOG) [91]	93.1	84.5	88.6	81.3	87.2
GraphSAGE(NeurIPS) [92]	91.7	84.6	88.0	81.6	87.1
MR-GraphConv(ICCV) [93]	**93.4**	84.6	**88.8**	81.4	87.6
AC-GNN(ICLR) [94]	91.9	84.7	88.2	79.7	86.4
GIN(ICLR) [65]	92.7	**85.1**	88.7	**81.7**	**87.7**

GNN. MR-GraphConv achieves the highest detection score of 88.8%, surpassing GIN at 0.1%. And GIN obtains 88.7% and 87.7% in terms of H-mean for detection and recognition, achieved 0.1% and 0.5% higher on H-mean than those of EdgeConv. Compared to EdgeConv, GIN not only considers the features of node itself but also aggregates the features of neighboring nodes and applies nonlinear transformations. This allows GIN to more comprehensively represent the local structural information of nodes. GraphSAGE also aggregates neighbor node features, but its aggregation method is relatively simple and may not effectively capture complex relationships between nodes like GIN does. Max-Relative GraphConv focus on relative information, but it may not have as strong discriminatory power as GIN when dealing with complex graph structures.

(5) Trade-off between performance and computational efficiency. In Table 5, we compares our model with competitive models on accuracy, parameters, and inference speed under the same device and experimental settings. It is shown that HiTextSpotter outperforms ABC-Net and DeepSolo by 1.8% and 1.5%. As for inference speed, our method is quicker than detection-by-recognition paradigm model ABC-Net. The additional graph convolutional network and multi granularities inevitably cost computation time. However, compared with the original convolution and attention mechanisms for the entire image, the additional computational costs for the instance and component level are relatively lightweight.

(6) Individual contributions of each loss function. In multi-task learning, multiple optimization objectives are associated with each other. Therefore, changes in a single loss function not only affect its corresponding optimization objective but also exert an indirect impact on the objectives related to it. As shown in Table 6, we have conducted ablation experiments. By isolating different loss functions, we intuitively observed the effect and contributions of each component. **Text classification loss** $\mathcal{L}_{rec}$ guides model to predict the script of text instance. Without this loss function, model transforms from a text spotting model to a text detection model, ignoring the semantic information. The result shows that without this loss function, the detection performance has decreased by 2.5%. This phenomenon might be because the design of decoder query is too complex for text detection task, and this structure causes the model to incorporate redundant information and interference. **Boundary loss** $\mathcal{L}_{bd}$ directly assists model in predicting the bounding box of text instance, reducing the gap between predicted polygon boundary and the labeled one. Without this component, the model cannot predict the bounding box but could achieve a 1.4% enhancement (83.1% versus 81.7%) in the recognition task without lexicon. This may be attributed to the fact that coordinate loss $\mathcal{L}_{coord}$ already provides sufficient positional information. The focus of boundary loss $\mathcal{L}_{bd}$ is just on predicting coordinates of the top and bottom points, and it inevitably introduces additional information, which interferes with the recognition task. The final selection of λ is made by comprehensively evaluating the overall performance on the text spotting task.

4.5. Visualization comparisons and analysis

Fig. 4 illustrates the results of our model in comparison to the previous mainstream model. Our improvements over the baseline resulted in

Table 5
Accuracy, parameter, and inference speed comparison of HiTextSpotter, DRRG, ABC-Net, TESTR and DeepSolo on Total-Text benchmark.

Method	Category	Detection(F1)	End-to-End None(F1)	Params	FPS
DRRG	Component-based	85.73	–	–	12
ABC-Net	Instance-based	87.0	73.5	–	10
TESTR	DETR-like	88.0	73.3	–	5.5
DeepSolo	DETR-like	87.3	79.7	**42.5**	**17**
HiTextSpotter(Ours)	Hierarchical	**88.7**	**81.7**	46.6	15

Table 6
Ablation studies on different decoder loss function settings in Total-Text benchmark. The last row is our default setting. λ with underscore indicates that it differs from the default hyper-parameter setting. "–" means that the metric cannot be predicted under current settings.

Hyper-parameter setting				Detection			End-to-End	
λ_{cls}	λ_{text}	λ_{coord}	λ_{bd}	P	R	F1	None	Full
1.0	0.0	1.0	0.5	**93.0**	80.4	86.2	–	–
1.0	1.0	1.0	0.5	91.6	84.7	88.0	80.8	86.9
1.0	0.5	1.0	0.0	–	–	–	**83.1**	**88.9**
1.0	0.5	1.0	1.0	88.9	82.7	85.7	80.4	84.9
1.0	0.5	1.0	0.5	92.7	**85.1**	**88.7**	81.7	87.7

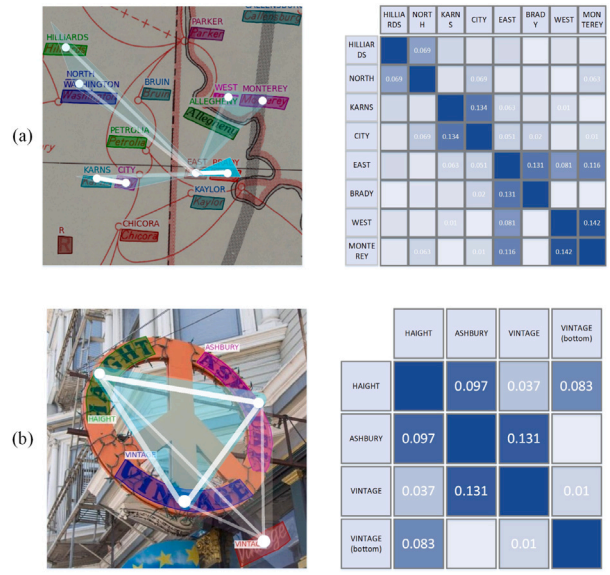


Fig. 5. Visualization of the spotting results of our model on scene text images. The left column intuitively visualizes their correlation levels between text instances by the colors of lines and triangles, and the right column visualizes the adjacency matrix partially from the last layer of graph neural network as a more detailed visualization to demonstrate the feasibility of our relationship modeling. The values in the adjacency matrix are the output of the last layer of GNN without any operation, in which we can directly observe the similarity between different words, and an empty value means the similarity is less than 0.001.

significant advancements in the detection task. Columns (a) and (b) of Fig. 4 display instances from the arbitrarily-shaped dataset Total-Text, while columns (c) and (d) showcase samples from the arbitrarily-oriented dataset ICDAR 2015. In Fig. 4(b-1), the baseline model omits most inverted text instances. Similar detection errors are observed in Fig. 4(d-1), where the model erroneously consolidates the lengthy text "GALAXY NOTE" at the bottom of the image into a single text box, failing to segment it into separate fragments, which consequently leads to recognition errors. Our HiTextSpotter enhances features and constructs graphs at both the instance and component levels. The component-level graph capitalizes on the flexible nature of graph structures to reconstruct inverted visual features, generating representations that align with the model's reading order. The instance-level graph learns the relationships between instances, facilitating inter-text communication and preventing the model from repeatedly focusing on the same text region. The application of these two graphs effectively addresses the recognition challenges mentioned earlier. Moreover, HiTextSpotter also achieves significant improvements in the end-to-end recognition task. In Fig. 4(a-1), artistic fonts such as "Sweet" and "Natural", as well as compressed fonts like "EKUYPE", are misrecognized by the baseline model. Similarly, in Fig. 4(c-1), the word "care" at the top of the image suffers recognition errors due to background interference. The component-level graph is instrumental in filtering such background noise, allowing the model to concentrate on the text itself. To provide a more intuitive understanding of the Instance Relation Graph Learning Module, Fig. 5(a) visualizes the correlations between instances. The line and triangle between "EAST" and "BRADY" have a darker color, and the corresponding similarity between them is the highest. In Fig. 5(b), it can be observed that the word "VINTAGE" positioned on the circle exhibits a higher degree of similarity compared to the same word located at the bottom.

4.6. The result of map text spotting

ICDAR 2024 MapText is a competitive competition released in the Robust Reading competition. Unlike natural scene text, there are unique challenges within map text, such as dense text regions, widely spaced characters, and rotated and curved text. Facing these issues, our HiTextSpotter provides corresponding solutions. Firstly, the primary challenge with widely spaced characters is hard to connect them over long distances, leading to drifting and incompleteness in text detection. To address this, HiTextSpotter employs multi-granularity graph at both the instance and component levels. Specially, component level graph compares inner parts of the instance, thereby better guiding the detection box to locate textual region. From samples in Fig. 6(c), (d), words such as "MEXICO" and "CROCKETT" are accurately recognized. Secondly, to address the issue of dense text regions like samples in Fig. 6(a), (c), our model learns relationships between instances by graph at instance level, enabling inter-instance communication and thereby avoiding problems such as redundant text detection and overlapping detection boxes. Finally, rotated and curved text is one of the most challenging issues in the field of text spotting. ABC-Net utilizes Bezier curves to fit the curve text instances, achieving competitive results. As for visual feature, employing the flexible structure of graphs, HiTextSpotter enables feature to better represent curve and text regions.

The official quantitative results are presented in Table 7. In the competition, five metrics are used, including F1, Recall (R), Precision (P), Tightness(T), and Quality(PDQ). Tightness $T \triangleq \frac{1}{|\mathrm{TP}|}\sum_{(g,d)\in \mathrm{TP}} \mathrm{IoU}(g,d)$ is the average IoU of all true positives(TP), while Panoptic Detection Quality PDQ $\triangleq T \times F$ comprehensively evaluates the text spotting performance. In the text detection and recognition task of this competition, we have achieved the third place in official ranking. In this result, our method is more competitive than DeepSolo (46.56% vs. 37.91%) and TESTR (46.56% vs. 27.19%), which demonstrates that relation construction for inter-text and intra-text contributes to the text

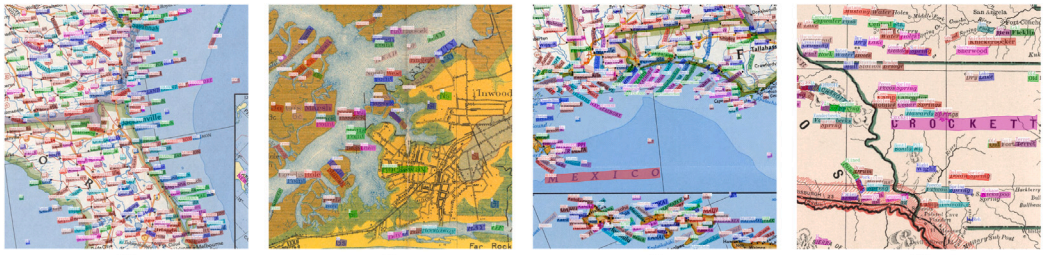


Fig. 6. Qualitative text spotting results of our HiTextSpotter on map images.

Table 7
Text spotting results on ICDAR map text.

Method	Baseline	Quality	Tightness	F1	Precision	Recall
HiTextSpotter (newest result)	ResNet-50	46.56	82.89	56.18	61.56	51.66
HiTextSpotter	ResNet-50	41.09	82.32	49.61	52.99	46.64
DeepSolo	ViTAE-s	37.91	72.55	52.25	49.88	54.87
TESTR(baseline)	–	27.19	84.56	32.90	34.11	31.76
TrOCR(just finetuned)	–	12.19	78.48	15.54	18.46	13.42

spotting task. We hope our perspective of multi-granularity hierarchical reasoning can provide new ideas for the text spotting field. After the competition, our team continues to research more competitive models based on the competition dataset and finds that without any fine-tuning and setting change, by enlarging the image size, the scene text spotting capacity could be significantly improved. Through the adjustment, we achieved the latest metrics on the validation set, as shown in the first row of Table 7.

4.7. Limitation and discussion

Firstly, HiTextSpotter currently struggles with mirrored or inverse-like text instances. In Fig. 4(b-1), the words "WAS", "MAY", and "1824" are completely missed. Additionally, we find that the detection polygon of the word "FROM" intersects with itself. This may indicates that the model cannot learn the reading order autonomously, even though it is implicitly included in text annotations. From the perspective of loss function, though coordinate loss $\mathcal{L}_{coord}$ and boundary loss $\mathcal{L}_{bd}$ supervise the position and shape information of text, they may not be sensitive to the reading order of text. In other words, they do not distinguish the information about top, bottom, left, and right directions of text. Reading order loss in conjunction with the direction perception module has potential to improve the robustness of inverse-like text. Besides, map text faces the challenges of being truncated. This is a significant difference compared to natural scene text. As shown in Fig. 6, we find that, due to the large size and dense distribution, a complete map is inevitably cropped and saved, with texts at the edges to be split across different sub-images. In the map text competition, we initially feed both normal and truncated texts into the model together, but the result shows that not only could model not effectively detect the truncated texts, but it also interferes with normal texts. Engineering solutions (such as re-recognizing the text by stitching the text regions) may deal with truncated texts. Finally, HiTextSpotter merely relies on focal loss for instance classification and L1 loss for coordination, even though they have demonstrated excellent performance. Facing the challenge of noisy, imbalanced text data, many improved classification loss functions, such as label smoothing and slide loss may be helpful. And as for the problem of various scales, alternative loss functions such as smooth L1 loss and GIoU Loss could be explored for detection task.

5. Conclusion

We have presented HiTextSpotter, a novel end-to-end trainable framework for text spotting. Unlike previous methods, our HiTextSpotter not only generates different granularity features by Multi-Granularity Feature Extraction Module but also employs graph convolution to model these features at Multi-Granularity Graph Relation Reasoning module. To be specific, instance-level graph learns the relationships among all textual instances, mitigating the issues of repetitive attention and omissions. Meanwhile, component-level graph consolidates spatial features, providing a finer-grained feature for text recognition tasks. In addition, adaptive gated fusion module integrates features at different granularities. The overall detection and recognition performance has prominent improvement. Sufficient experiments demonstrate the superiority and efficiency of our HiTextSpotter on both nature scene and map text benchmarks.

CRediT authorship contribution statement

Jialiang Li: Writing – original draft, Software, Methodology, Data curation. **Canhui Xu:** Writing – review & editing, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62471272, 61806107 and 62201314, the Opening Project of State Key Laboratory of Digital Publishing Technology.

Data availability

The authors do not have permission to share data.

References

[1] W. Sun, Q. Wang, Z. Hou, X. Chen, Q. Yan, Y. Zhang, DPGS: Cross-cooperation guided dynamic points generation for scene text spotting, Knowl.-Based Syst. 302 (2024) 112399.
[2] P. Lyu, M. Liao, C. Yao, W. Wu, X. Bai, Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 67–83.
[3] S. Long, X. He, C. Yao, Scene text detection and recognition: The deep learning era, Int. J. Comput. Vis. 129 (2021).
[4] Q. Ye, D.S. Doermann, Text detection and recognition in imagery: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2015) 1480–1500.
[5] X.-F. Wang, Z.-H. He, K. Wang, Y.-F. Wang, L. Zou, Z.-Z. Wu, A survey of text detection and recognition algorithms based on deep learning technology, Neurocomputing 556 (2023) 126702.

[6] Z. Li, R. Guan, Q. Yu, Y.-Y. Chiang, C.A. Knoblock, Synthetic map generation to provide unlimited training data for historical map text detection, in: Proceedings of the 4th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery, 2021, pp. 17–26.
[7] Y.-Y. Chiang, W. Duan, S. Leyk, J.H. Uhl, C.A. Knoblock, Using Historical Maps in Scientific Studies: Applications, Challenges, and Best Practices, Springer Publishing Company, Incorporated, 2019.
[8] Z. Li, Y. Lin, Y.-Y. Chiang, J. Weinman, S. Tual, J. Chazalon, J. Perret, B. Duménieu, N. Abadie, Icdar 2024 competition on historical map text detection, recognition, and linking, in: International Conference on Document Analysis and Recognition, Springer, 2024, pp. 363–380.
[9] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, L. Wang, Abcnet: Real-time scene text spotting with adaptive bezier-curve network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9809–9818.
[10] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2963–2970.
[11] W. Huang, Z. Lin, J. Yang, J. Wang, Text localization in natural images using stroke feature transform and text covariance descriptors, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1241–1248.
[12] A.K. Jain, B. Yu, Automatic text location in images and video frames, Pattern Recognit. 31 (12) (1998) 2055–2076.
[13] L. Neumann, J. Matas, A method for text localization and recognition in real-world images, in: Computer Vision–ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November (2010) 8-12, Revised Selected Papers, Part III 10, Springer, 2011, pp. 770–783.
[14] C. Yao, X. Bai, W. Liu, Y. Ma, Z. Tu, Detecting texts of arbitrary orientations in natural images, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 1083–1090.
[15] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D.J. Wu, A.Y. Ng, Text detection and character recognition in scene images with unsupervised feature learning, in: 2011 International Conference on Document Analysis and Recognition, IEEE, 2011, pp. 440–445.
[16] S. Lee, J.H. Kim, Integrating multiple character proposals for robust scene text extraction, Image Vis. Comput. 31 (11) (2013) 823–840.
[17] T. Wang, D.J. Wu, A. Coates, A.Y. Ng, End-to-end text recognition with convolutional neural networks, in: Proceedings of the 21st International Conference on Pattern Recognition, IEEE, 2012, pp. 3304–3308.
[18] K. Wang, B. Babenko, S. Belongie, End-to-end scene text recognition, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 1457–1464.
[19] J. Tang, Z. Yang, Y. Wang, Q. Zheng, Y. Xu, X. Bai, Seglink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping, Pattern Recognit. 96 (2019) 106954.
[20] W. Feng, W. He, F. Yin, X.-Y. Zhang, C.-L. Liu, Textdragon: An end-to-end framework for arbitrary shaped text spotting, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9076–9085.
[21] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, C. Yao, Textsnake: A flexible representation for detecting text of arbitrary shapes, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 20–36.
[22] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, X. Bai, Textfield: Learning a deep direction field for irregular scene text detection, IEEE Trans. Image Process. 28 (11) (2019) 5566–5579.
[23] B. Shi, X. Bai, S. Belongie, Detecting oriented text in natural images by linking segments, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2550–2558.
[24] Y. Baek, B. Lee, D. Han, S. Yun, H. Lee, Character region awareness for text detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9365–9374.
[25] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, C. Shen, Efficient and accurate arbitrary-shaped text detection with pixel aggregation network, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 8440–8449.
[26] Z. Tian, M. Shu, P. Lyu, R. Li, C. Zhou, X. Shen, J. Jia, Learning shape-aware embedding for scene text detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4234–4243.
[27] M. Liao, Z. Zou, Z. Wan, C. Yao, X. Bai, Real-time scene text detection with differentiable binarization and adaptive scale fusion, IEEE Trans. Pattern Anal. Mach. Intell. 45 (1) (2022) 919–931.
[28] S.-X. Zhang, X. Zhu, J.-B. Hou, C. Liu, C. Yang, H. Wang, X.-C. Yin, Deep relational reasoning graph network for arbitrary shape text detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9699–9708.
[29] T. Sheng, J. Chen, Z. Lian, Centripetaltext: An efficient text instance representation for scene text detection, Adv. Neural Inf. Process. Syst. 34 (2021) 335–346.
[30] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, C. Sun, An end-to-end textspotter with explicit alignment and attention, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5020–5029.
[31] S. Qin, A. Bissacco, M. Raptis, Y. Fujii, Y. Xiao, Towards unconstrained end-to-end text spotting, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 4704–4714.
[32] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, J. Yan, Fots: Fast oriented text spotting with a unified network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5676–5685.
[33] H. Li, P. Wang, C. Shen, Towards end-to-end text spotting with convolutional recurrent neural networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5238–5246.
[34] M. Liao, G. Pang, J. Huang, T. Hassner, X. Bai, Mask textspotter v3: Segmentation proposal network for robust scene text spotting, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August (2020) 23–28, Proceedings, Part XI 16, Springer, 2020, pp. 706–722.
[35] Y. Liu, C. Shen, L. Jin, T. He, P. Chen, C. Liu, H. Chen, Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting, IEEE Trans. Pattern Anal. Mach. Intell. 44 (11) (2021) 8048–8064.
[36] P. Wang, C. Zhang, F. Qi, S. Liu, X. Zhang, P. Lyu, J. Han, J. Liu, E. Ding, G. Shi, Pgnet: Real-time arbitrarily-shaped text spotting with point gathering network, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 2782–2790, (4).
[37] X. Canhui, L. Yuteng, S. Cao, Z. Honghong, B. Hengyue, C. Yinong, Him: hierarchical multimodal network for document layout analysis, Appl. Intell. 53 (20) (2023) 24314–24326.
[38] L. Qiao, Y. Chen, Z. Cheng, Y. Xu, Y. Niu, S. Pu, F. Wu, Mango: A mask attention guided one-stage scene text spotter, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 2467–2476, (3).
[39] C. Shi, C. Xu, H. Bi, Y. Cheng, Y. Li, H. Zhang, Lateral feature enhancement network for page object detection, IEEE Trans. Instrum. Meas. 71 (2022) 1–10.
[40] C.-h. Xu, C. Shi, Y.-n. Chen, End-to-end dilated convolution network for document image semantic segmentation, J. Cent. South Univ. 28 (6) (2021) 1765–1774.
[41] C. Xu, C. Shi, H. Bi, C. Liu, Y. Yuan, H. Guo, Y. Chen, A page object detection method based on mask r-cnn, IEEE Access 9 (143,448–143,457) (2021).
[42] M. Huang, Y. Liu, Z. Peng, C. Liu, D. Lin, S. Zhu, N. Yuan, K. Ding, L. Jin, Swintextspotter: Scene text spotting via better synergy between text detection and text recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 4593–4603.
[43] D. Peng, X. Wang, Y. Liu, J. Zhang, M. Huang, S. Lai, J. Li, S. Zhu, D. Lin, C. Shen, X. Bai, L. Jin, Spts: Single-point text spotting, in: Proceedings of the 30th ACM International Conference on Multimedia, in: ser. MM '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 4272–4281.
[44] Y. Kittenplon, I. Lavi, S. Fogel, Y. Bar, R. Manmatha, P. Perona, Towards weakly-supervised text spotting using a multi-task transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 4604–4613.
[45] X. Zhang, Y. Su, S. Tripathi, Z. Tu, Text spotting transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9519–9528.
[46] M. Ye, J. Zhang, S. Zhao, J. Liu, T. Liu, B. Du, D. Tao, Deepsolo: Let transformer decoder with explicit points solo for text spotting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19,348–19,357.
[47] M. Huang, J. Zhang, D. Peng, H. Lu, C. Huang, Y. Liu, X. Bai, L. Jin, Estextspotter: Towards better scene text spotting with explicit synergy in transformer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 19,495–19,505.
[48] Y. Xie, Q. Qiao, J. Gao, T. Wu, S. Huang, J. Fan, Z. Cao, Z. Wang, Y. Zhang, J. Zhang, H. Sun, DNTextSpotter: Arbitrary-Shaped scene text spotting via improved denoising training, 2024, arXiv e-prints, arXiv:2408.00355.
[49] Z. Yuan, C. Shi, Mgn-net: Multi-granularity graph fusion network in multi-modal for scene text spotting, IEEE Internet Things J. (2024).
[50] M. Huang, D. Peng, H. Li, et al., Mingxin Huang, Dezhi Peng, Swintextspotter v2: Towards better synergy for scene text spotting, Int. J. Comput. Vis. (2025).
[51] A. Das, S. Biswas, A. Banerjee, J. Lladós, U. Pal, S. Bhattacharya, Harnessing the power of multi-lingual datasets for pre-training: Towards enhancing text spotting performance, in: 2024 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV, 2024, pp. 707–717.
[52] S.-X. Zhang, C. Yang, X. Zhu, H. Zhou, H. Wang, X.-C. Yin, Inverse-like antagonistic scene text spotting via reading-order estimation and dynamic sampling, Trans. Img. Proc. 33 (2024) 825–839.
[53] R. Ronen, S. Tsiper, O. Anschel, I. Lavi, A. Markovitz, R. Manmatha, Glass: Global to local attention for scene-text spotting, in: European Conference on Computer Vision, Springer, 2022, pp. 249–266.
[54] H. Bi, C. Xu, C. Shi, G. Liu, H. Zhang, Y. Li, J. Dong, Hgr-net: Hierarchical graph reasoning network for arbitrary shape scene text detection, IEEE Trans. Image Process. (2023).
[55] H. Hu, J. Gu, Z. Zhang, J. Dai, Y. Wei, Relation networks for object detection, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 3588–3597.
[56] P. Wang, C. Da, C. Yao, Multi-granularity prediction for scene text recognition, in: European Conference on Computer Vision, 2022.
[57] W. Zhou, D. Du, L. Zhang, T. Luo, Y. Wu, Multi-granularity alignment domain adaptation for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 9581–9590.

[58] L. Zhang, W. Zhou, H. Fan, T. Luo, H. Ling, Robust domain adaptive object detection with unified multi-granularity alignment, IEEE Trans. Pattern Anal. Mach. Intell. (2024) 1–18.
[59] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.
[60] S. Long, S. Qin, D. Panteleev, A. Bissacco, Y. Fujii, M. Raptis, Towards end-to-end unified scene text detection and layout analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1049–1059.
[61] M. Ye, J. Zhang, J. Liu, C. Liu, B. Yin, C. Liu, B. Du, D. Tao, Hi-sam: Marrying segment anything model for hierarchical text segmentation, IEEE Trans. Pattern Anal. Mach. Intell. (2024).
[62] H. Bi, C. Xu, C. Shi, G. Liu, Y. Li, H. Zhang, J. Qu, Srrv: A novel document object detector based on spatial-related relation and vision, IEEE Trans. Multimed. 25 (2023) 3788–3798.
[63] K. Han, Y. Wang, J. Guo, Y. Tang, E. Wu, Vision gnn: An image is worth graph of nodes, Adv. Neural Inf. Process. Syst. 35 (2022) 8291–8303.
[64] C. Shi, C. Xu, J. He, Y. Chen, Y. Cheng, Q. Yang, H. Qiu, Graph-based convolution feature aggregation for retinal vessel segmentation, Simul. Model. Pract. Theory 121 (2022) 102653.
[65] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks? in: International Conference on Learning Representations, 2019.
[66] M. Ye, J. Zhang, S. Zhao, J. Liu, B. Du, D. Tao, Dptext-detr: Towards better scene text detection with dynamic points in transformer, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 3241–3249, (3).
[67] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V.R. Chandrasekhar, S. Lu, et al., Icdar 2015 competition on robust reading, in: 2015 13th International Conference on Document Analysis and Recognition, ICDAR, IEEE, 2015, pp. 1156–1160.
[68] L. Yuliang, J. Lianwen, Z. Shuaitao, Z. Sheng, Detecting curve text in the wild: New dataset and new solution, 2017, arXiv preprint arXiv:1712.02170.
[69] C.K. Ch'ng, C.S. Chan, Total-text: A comprehensive dataset for scene text detection and recognition, in: 2017 14th IAPR International Conference on Document Analysis and Recognition, ICDAR, Vol. 1, IEEE, 2017, pp. 935–942.
[70] Y. Liu, L. Jin, Deep matching prior network: Toward tighter multi-oriented text detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1962–1969.
[71] Y. Liu, L. Jin, S. Zhang, C. Luo, S. Zhang, Curved scene text detection via transverse and longitudinal sequence connection, Pattern Recognit. 90 (2019) 337–345.
[72] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, J. Liang, East: an efficient and accurate scene text detector, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5551–5560.
[73] M. Liao, B. Shi, X. Bai, Textboxes++: A single-shot oriented scene text detector, IEEE Trans. Image Process. 27 (8) (2018) 3676–3690.
[74] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, X. Xue, Arbitrary-oriented scene text detection via rotation proposals, IEEE Trans. Multimed. 20 (11) (2018) 3111–3122.
[75] C. Xue, S. Lu, W. Zhang, Msr: multi-scale shape regression for scene text detection, 2019, arXiv preprint arXiv:1901.02596.
[76] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, S. Shao, Shape robust text detection with progressive scale expansion network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9336–9345.
[77] C. Zhang, B. Liang, Z. Huang, M. En, J. Han, E. Ding, X. Ding, Look more than once: An accurate detector for text of arbitrary shapes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10,552–10,561.
[78] Y. Liu, L. Jin, C. Fang, Arbitrarily shaped scene text detection with a mask tightness text detector, IEEE Trans. Image Process. 29 (2019) 2918–2930.
[79] Y. Wang, H. Xie, Z.-J. Zha, M. Xing, Z. Fu, Y. Zhang, Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11,753–11,762.
[80] H. Wang, P. Lu, H. Zhang, M. Yang, X. Bai, Y. Xu, M. He, Y. Wang, W. Liu, All you need is boundary: Toward arbitrary-shaped text spotting, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 12,160–12,167, (07).
[81] P. Dai, S. Zhang, H. Zhang, X. Cao, Progressive contour regression for arbitrary-shape scene text detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7393–7402.
[82] Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin, W. Zhang, Fourier contour embedding for arbitrary-shaped text detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3123–3131.
[83] Z. Raisi, M.A. Naiel, G. Younes, S. Wardell, J.S. Zelek, Transformer-based text detection in the wild, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3162–3171.
[84] P. Dai, S. Yao, Z. Li, S. Zhang, X. Cao, Ace: Anchor-free corner evolution for real-time arbitrarily-oriented object detection, IEEE Trans. Image Process. 31 (2022) 4076–4089.
[85] S.-X. Zhang, X. Zhu, C. Yang, H. Wang, X.-C. Yin, Adaptive boundary proposal network for arbitrary shape text detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1305–1314.
[86] L. Qiao, S. Tang, Z. Cheng, Y. Xu, Y. Niu, S. Pu, F. Wu, Text perceptron: Towards end-to-end arbitrary-shaped text spotting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 11,899–11,907, (3).
[87] W. Wang, E. Xie, X. Li, X. Liu, D. Liang, Z. Yang, T. Lu, C. Shen, Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text, IEEE Trans. Pattern Anal. Mach. Intell. 44 (9) (2021) 5349–5367.
[88] J. Wu, P. Lyu, G. Lu, C. Zhang, W. Pei, Single shot self-reliant scene text spotter by decoupled yet collaborative detection and recognition, 2022, arXiv preprint arXiv:2207.07253.
[89] W. Wang, Y. Zhou, J. Lv, D. Wu, G. Zhao, N. Jiang, W. Wang, Tpsnet: Reverse thinking of thin plate splines for arbitrary shape scene text representation, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 5014–5025.
[90] S. Fang, Z. Mao, H. Xie, Y. Wang, C. Yan, Y. Zhang, Abinet++: Autonomous, bidirectional and iterative language modeling for scene text spotting, IEEE Trans. Pattern Anal. Mach. Intell. 45 (6) (2022) 7123–7141.
[91] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph cnn for learning on point clouds, ACM Trans. Graph. (Tog) 38 (5) (2019) 1–13.
[92] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Adv. Neural Inf. Process. Syst. 30 (2017).
[93] G. Li, M. Muller, A. Thabet, B. Ghanem, Deepgcns: Can gcns go as deep as cnns? in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9267–9276.
[94] P. Barceló, E.V. Kostylev, M. Monet, J. Pérez, J. Reutter, J.-P. Silva, The logical expressiveness of graph neural networks, in: 8th International Conference on Learning Representations, 2020.