RESEARCH-ARTICLE

Graph Pooling Inference Network for Text-based VQA

SHENG ZHOU, Hefei University of Technology, Hefei, Anhui, China

DAN GUO, Hefei University of Technology, Hefei, Anhui, China

XUN YANG, University of Science and Technology of China, Hefei, Anhui, China

JIANFENG DONG, Zhejiang Gongshang University, Hangzhou, Zhejiang, China

MENG WANG, Hefei University of Technology, Hefei, Anhui, China

Published: 11 January 2024
Online AM: 29 November 2023
Accepted: 21 November 2023
Revised: 13 October 2023
Received: 23 July 2023

Graph Pooling Inference Network for Text-based VQA

SHENG ZHOU and DAN GUO, HeFei University of Technology, China
XUN YANG, University of Science and Technology of China, China
JIANFENG DONG, Zhejiang Gongshang University, China
MENG WANG, HeFei University of Technology, China

Effectively leveraging objects and optical character recognition (OCR) tokens to reason out pivotal scene text is critical for the challenging Text-based Visual Question Answering (TextVQA) task. Graph-based models can effectively capture the semantic relationship among visual entities (objects and tokens) and report remarkable performance in TextVQA. However, previous efforts usually leverage all visual entities and ignore the negative effect of superfluous entities. This article presents a Graph Pooling Inference Network (GPIN), which is an evolutionary graph learning method to purify the visual entities and capture the core semantics. It is observed that the dense distribution of reduplicative objects and the crowd of semantically dependent OCR tokens usually co-exist in the image. Motivated by this, GPIN adopts an adaptive node dropping strategy to dynamically downscale semantically closed nodes for graph evolution and update. To deepen the comprehension of scene text, GPIN is a dual-path hierarchical graph architecture that progressively aggregates the evolved object graph and the evolved token graph semantics into a graph vector that serves as visual cues to facilitate the answer reasoning. It can effectively eliminate object redundancy and enhance the association of semantically continuous tokens. Experiments conducted on TextVQA and ST-VQA datasets show that GPIN achieves promising performance compared with state-of-the-art methods.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Computer vision**; **Computer vision tasks**; **Scene understanding;**

Additional Key Words and Phrases: Text-based visual question answering, graph inference, graph pooling

ACM Reference format:
Sheng Zhou, Dan Guo, Xun Yang, Jianfeng Dong, and Meng Wang. 2024. Graph Pooling Inference Network for Text-based VQA. *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 4, Article 112 (January 2024), 21 pages.
https://doi.org/10.1145/3634918

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB4500600; in part by the National Natural Science Foundation of China (NSFC) under Grant 62020106007, Grant 62272144, Grant U20A20183, Grant 72188101, Grant 62272435, and Grant U22A2094; in part by the Major Project of Anhui Province under Grant 202203a05020011; and in part by the University Synergy Innovation Program of Anhui Province under Grant GXXT-2022-047.

Authors' addresses: S. Zhou, D. Guo (Corresponding author), M. Wang (Corresponding author), HeFei University of Technology, No. 485 Danxia Road, Hefei, Anhui, China, 230601; e-mails: hzgn97@gmail.com, guodan@hfut.edu.cn, eric.mengwang@gmail.com; X. Yang (Corresponding author), University of Science and Technology of China, No. 96, JinZhai Road, Hefei, Anhui, China, 230026; e-mail: xyang21@ustc.edu.cn; J. Dong, Zhejiang Gongshang University, No. 18, Xuecheng Street, Xiasha Higher Education Park, Hangzhou, Zhejiang, China, 310018; e-mail: dongjf24@gmail.com.

1551-6857/2024/01-ART112 $15.00
https://doi.org/10.1145/3634918

1 INTRODUCTION

With the advancement of multimodal research [12, 36, 45, 53–55, 61], Question Answering (QA) involving both vision and language has become a hot topic, such as **Visual Question Answering (VQA)** [8, 17, 18, 21], Knowledge-based VQA [11, 16], and **Text-based VQA (TextVQA)** [25, 44]. Compared with the general VQA tasks that involve common visual questions about people, scenes, and even plot understanding, TextVQA is more challenging due to the additional incorporation of scene text. As a representative QA task to study scene text in images, TextVQA is dedicated to perceiving and comprehending scene texts to answer text-based visual questions. This requires the model to have a variety of capabilities, including object detection, text recognition, and multimodal reasoning. In practical application, the TextVQA model can be widely used in various real-world scenarios, including driving assistance [63], children education [15], and assisting visually impaired users [44].

Benefited by the development of object detection technology [5, 38, 39, 42, 46], TextVQA models integrate object detectors [42, 46] and **optical character recognition (OCR)** systems [5, 38] to better perceive scene text-based images. Specifically, object detectors excel at identifying visual objects (e.g., paper and toy in Figure 1), whereas OCR systems perform well at recognizing scene texts (e.g., "meg" and "gardiner" in Figure 1). To achieve an effective scene text understanding, it is necessary to pay attention not only to the visual properties of objects and scene texts but also to the linguistic properties of scene texts. Therefore, the pivotal research problem of TextVQA is how to effectively and jointly model the objects and OCR tokens to accurately understand and leverage the scene text for answering.

Many efforts have been dedicated to this task [13–15, 22, 25, 26, 32, 35, 37, 44, 48, 64], in which graph-based models [14, 15, 26, 37] achieve remarkable performance on TextVQA benchmark datasets due to their strong capacity of relationship modeling. However, to utilize objects and OCR tokens, most existing graph-based TextVQA models always leverage all the visual entities (nodes) detected in the given image with full relationships (edges) (CRN [37] and MM-GNN [15]) or sparse relationships (SA-M4C [26], SMA [14], and TIG [35]) for graph learning. We argue that considering all visual entities may inevitably introduce the redundancy issue that is not easily alleviated by just sparse relationship learning. It is mainly caused by the following two reasons: (1) Cumulative redundant entities tend to cause attention bias, such as the green toy covered with many similar bounding boxes in Figure 1, and (2) sparse relationship learning cannot effectively reduce the negative effect of entity redundancy, and strong sparsity [14] or weak sparsity [26] is easy to cause insufficient interaction or inadequate redundancy removal. Unlike previous graph models, this article builds an evolutionary graph learning solution that can adaptively downscale the superfluous nodes to dig out core semantics for achieving effective and robust scene text-based VQA.

Based on our observations, enough object and OCR token features contribute to understanding the image content [25], yet the key to TextVQA model is to deduce the local visual entities associated with the answers in the image based on the questions. As shown in Figure 1, there are plenty of objects and scene texts distributed throughout the nature image, but only a few are relevant to the answer. However, previous efforts often emphasize the role of all visual content (even redundancy) in answer reasoning instead of focusing on mining the core semantics, resulting in inevitable redundancy and thus hindering the accurate modeling of the semantic association between tokens. Differently, this article emphasizes capturing key scene semantics while understanding the image content. To this end, we consider the spatial distribution of objects and OCR tokens and point out the difficulties in mining core objects and OCR tokens from the following two aspects: (1) *Redundancy removal of densely distributed objects.* Based on the common object detector [42] on TextVQA, it is observed that boundary boxes of similar size are often overlaid on the same object, leading to the accumulation of redundancy, e.g., three similar-sized bounding

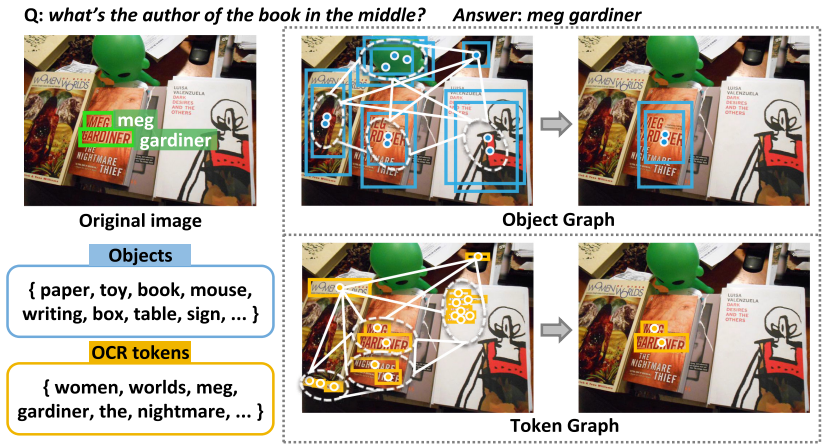

Fig. 1. The TextVQA task requires understanding objects and tokens and deriving the pivotal scene texts to answer the question. To solve this task, we propose a graph model to learn semantically closed node clusters and downscale the nodes driven by the question for semantic purification.

boxes are repeatedly detected for the green toy in Figure 1. Such repetitive objects may hinder the scene text-based answer reasoning. (2) *Semantic integration of discrete OCR tokens.* Different from the spatial distribution of objects, each scene text is detected as a single OCR token by OCR systems [5, 38], and individual tokens are discretely distributed in the given image [43, 49]. If scene text is a phrase or sentence (e.g., "meg gardiner"), then TextVQA model needs to combine discrete tokens to convey its overall meaning.

To deal with aforementioned challenges, this article proposes a novel **Graph Pooling Inference Network (GPIN)**, which designs an adaptive node dropping strategy with the goal of purifying the semantics of objects and OCR tokens to excavating core semantics. The main technique of GPIN is to distinguish semantic similarity between nodes and then downscale the semantically related nodes (i.e., redundant objects or semantically continuous tokens) guided by the question. This process involves two aspects: redundancy removal of objects and semantic connectivity of tokens. Specifically, GPIN is a dynamic graph learning model that adaptively drops semantically closed nodes for graph evolution and update. To fully exploit visual cues, GPIN is a hierarchical graph architecture that progressively converges evolved graph semantics while mitigating the effect of the irrelevant or redundant message. In this way, to better comprehend scene text in images, we design a *Hierarchical Object Graph* module and a *Hierarchical Token Graph* module to take into account the respective characteristics of objects and tokens. Ultimately, an informative graph vector is obtained by a *Dual-Path Graph Fusion* module to facilitate the answer reasoning. Extensive experiments and analysis on TextVQA and ST-VQA datasets showed the effectiveness and interpretability of our method.

The main contributions are summarized as follows:

— We propose an evolutionary graph learning model, GPIN, which is a pioneer in introducing graph pooling learning into the TextVQA task. The core of GPIN is an adaptive node dropping technique to augment the graph evolution.
— The proposed GPIN is a hierarchical graph structure that progressively digs out visual hints in objects and tokens. We design a Hierarchical Object Graph and a Hierarchical Token Graph to semantically purify the visual content. By the Dual-Path Graph Fusion, a comprehensive graph representation is obtained for answer reasoning.

— Extensive experiments on TextVQA and ST-VQA datasets demonstrate that our method has excellent performance compared with the SOTA methods including pre-trained methods. Extensive visualization also shows a better purification effect on visual content in our method.

2 RELATED WORK

2.1 Text-based Visual Question Answering

Text-based VQA is being investigated to boost the generalization ability of VQA models [23, 24] in comprehension of scene texts in the image. Existing methods can be divided into four categories: (1) *Attention-based models*, e.g., LoRRA [44] and Three-block [63]; (2) *Transformer-based models*, e.g., M4C [25], PAT [60], LaAP-Net [20], LaTr [2], and GRT [52]; (3) *Representation learning-based model* BOV [57]; (4) *Graph-based models*, MM-GNN [15], CRN [37], SMA [14], SA-M4C [26], TIG [35], and SSGN [62]. Apart from these, pre-training models TAP [56], LOGOS [40], and LaTr [2] demonstrate great precision performances but necessitate large-scale datasets and expensive training cost [33]. Due to the complicated relationship learning among objects and OCR tokens in the image, graph modeling is the dominant framework for this task. Our method also falls within the scope of graph modeling. Unlike previous graph-based models, this article presents a dynamic graph learning (evolutionary) model that adaptively drops superfluous nodes and progressively aggregates the graph semantics. It helps to mine visual cues from complex image scenes for answer reasoning.

2.2 Graph Pooling

Graph Pooling [51, 58] is a new neural operation on the graph neural network that aims to downscale the graph structure and refine the graph representation. It has been attempted to apply in the visual and language tasks [24, 34, 50, 59]. For caption generation, Zhang et al. [59] construct multiple scene graphs for the same image and embed each graph into a vector by graph pooling and, finally, generate dense captions. For video grounding, Li et al. [34] apply graph pooling on the attentive features of video and query to obtain more representative multimodal features. And as for the task of community question answer matching, graph pooling is used to select important nodes for question-answer matching [24]. In this work, we are the first to introduce the graph pooling for TextVQA task. We propose a graph pooling inference network, which realizes an adaptive node dropping of superfluous visual entities (i.e., objects and OCR tokens), thus achieving the semantic purification of visual entities. To be specific, the effect is to merge redundant object entities and bridge the semantic association of isolated tokens in the image separately.

3 METHOD

Given a question referring to scene text appearing in the image, TextVQA task requires understanding the question, perceiving scene texts, and answering the question accurately. One of the key challenges of this task is how to cut off irrelevant visual distractions and reason out the pivotal scene texts. In this work, we propose a GPIN to address this issue. As shown in Figure 2, our model consists of three modules as follows: (1) *Graph Construction* builds a spatial object graph and a spatial OCR token graph in Section 3.1, where both the object detector and OCR system are considered to guarantee the discovery of all the scene text cues in the image; (2) *Graph Pooling Inference* outlines the core methodology of our graph model in Section 3.2 and a *Dual-Path Graph Fusion* architecture is designed in Section 3.3, which perform an adaptive and progressive graph pooling network for the purification on semantically related nodes; and (3) *Answer Generation and Model Optimization* is introduced in Section 3.4, where an iterative transformer block is applied as the answer decoder to predict the answer.

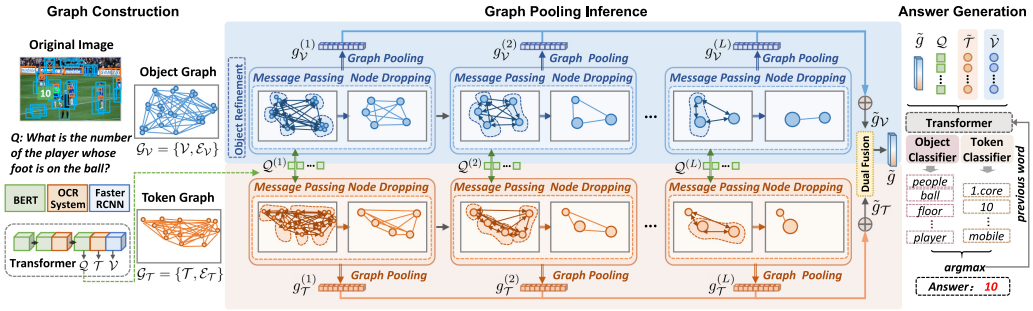

Fig. 2. The overall framework of GPIN. Given an image and a question, we prepare the features of the question ($\mathcal{Q}$), OCR tokens ($\mathcal{T}$), and objects ($\mathcal{V}$), and build two spatial graphs $\mathcal{G}_v$ and $\mathcal{G}_t$ with nodes $\mathcal{V}$ and $\mathcal{T}$, respectively. In this work, *a novel graph pooling inference is used to progressively purify the semantics of question-relevant objects and OCR tokens.* It effectively removes objects' redundancy and establishes dense tokens' connection. By this inference process, we obtain a global graph vector to represent the informative hint for predicting the answer.

3.1 Graph Construction

3.1.1 Feature Encoding. Accurately understanding the question and perceiving scene text in the image are the prerequisites for answer prediction. As shown in Figure 2, attempting to cover all of the scene texts, we take Faster R-CNN [42] to extract the object feature $V \in \mathbb{R}^{M\times d}$ and use the OCR system [5, 38] to extract the OCR token feature $T \in \mathbb{R}^{N\times d}$ in the image, where M and N is the respective number of object and OCR token. And we extract the question feature $Q \in \mathbb{R}^{L_Q\times d}$, where L_Q is the length of the question. Following Reference [37], we input Q, V, and T into a transformer-based encoder and update the features $\boldsymbol{Q} = \{q_i\}_{i=1}^{L_Q}$, $\mathcal{V} = \{v_i\}_{i=1}^{M}$, and $\mathcal{T} = \{t_i\}_{i=1}^{N}$, where $q_i, v_i, t_i \in \mathbb{R}^d$.

3.1.2 Graph Representation. We build the object graph and OCR token graph respectively, i.e., $\mathcal{G} = \{\mathcal{G}_\mathcal{V}, \mathcal{G}_\mathcal{T}\}$. The object graph $\mathcal{G}_\mathcal{V} = \{\mathcal{V}, \mathcal{E}_\mathcal{V}\}$ is a fully connected graph, where $\mathcal{V}$ includes all object nodes and $\mathcal{E}_\mathcal{V}$ is a set of bidirectional directed edges denoted as $\mathcal{E}_\mathcal{V} = \mathcal{E}_{\mathcal{V}\leftrightarrow\mathcal{V}}$. Similarly, the OCR token graph is defined as $\mathcal{G}_\mathcal{T} = \{\mathcal{T}, \mathcal{E}_\mathcal{T}\}$, where $\mathcal{E}_\mathcal{T} = \mathcal{E}_{\mathcal{T}\leftrightarrow\mathcal{T}}$. Moreover, for the answer reasoning, it is critical to not only encode the characters and appearance in the image (e.g., color, font, and background) but also take into consideration the spatial cues (e.g., layout and location). For example, in Figure 2, the spatial relationship of "the foot on the ball" is an important clue to answer the question. To tap the spatial cues, we build the spatial relationships of two nodes (two object nodes in $\mathcal{G}_\mathcal{V}$ and two token nodes in $\mathcal{G}_\mathcal{T}$), i.e., spatial edge modeling. Taking an edge $e_{i\to j}$ from node n_i to node n_j as an example, the edge is assigned with spatial weights as $e_{i\to j} = [\frac{x_i^{tl}-x_j^c}{w_j}, \frac{y_i^{tl}-y_j^c}{h_j}, \frac{x_i^{br}-x_j^c}{w_j}, \frac{y_i^{br}-y_j^c}{h_j}, \frac{w_i*h_i}{w_j*h_j}]$, where (x_i^{tl}, y_i^{tl}) and (x_i^{br}, y_i^{br}) denote the top-left and bottom-right coordinates of node n_i's bounding box and (x_j^c, y_j^c), w_j, and h_j denote the center coordinate, width, and height of node n_j's bounding box, respectively. Generally, due to the different size of each node's bounding box, the relationship (edge weights) of the paired bidirectional edges are not equivalent, e.g., $e_{i\to j} \neq e_{j\to i}$.

3.1.3 Object Refinement. In a natural image scene, it is observed that visual objects are relatively more abundant than scene texts. As mentioned above, we have detected M objects and N OCR tokens, i.e., $M = 100$ and $N = 50$, to provide enough object and token features for understanding images following References [25, 57]. As is well known, there are redundant

objects such as overmuch similar objects covering the same subject in the image, and partial objects are irrelevant to answering the question. In this part, we attempt to downscale the object set in a question-driven manner. Specifically, we perform a similarity filtration scheme to obtain the association between the objects and the question. The scheme first calculates the cosine similarity matrix $\mathcal{A}$ and uses it to obtain a semantic difference representation $\mathcal{S}_{\mathcal{V}} = \{s_{v_i}\}_{i=1}^{M}$ between the question and each object as

$$\begin{cases} \mathcal{A} = \text{Softmax}(C(\boldsymbol{Q}, \mathcal{V})) \in \mathbb{R}^{\mathcal{L}_Q \times M}; \\ a_i = \sum_{j=1}^{\mathcal{L}_Q} a_{ij} v_i \in \mathbb{R}^d; \\ s_{v_i} = \dfrac{\boldsymbol{W}_{s_1} |v_i - a_i|^2}{\|\boldsymbol{W}_{s_1} |v_i - a_i|^2\|_2} \in \mathbb{R}^p, \end{cases} \tag{1}$$

where $C(\cdot)$ is cosine function, $|\cdot|^2$ and $\|\cdot\|_2$ indicate element-wise square and L2-norm, and $\boldsymbol{W}_{s_1} \in \mathbb{R}^{p \times d}$ is a learnable parameter.

With the difference representation $\mathcal{S}_{\mathcal{V}}$, we further predict the corresponding difference scores $\tilde{\mathcal{S}_{\mathcal{V}}} = \{\tilde{s}_{v_i}\}_{i=1}^{M}$ of objects to the question, where $\tilde{s}_{v_i}$ denotes the difference of object v_i to question $\boldsymbol{Q}$. Please note that a smaller $\tilde{s}_{v_i}$ represents a higher relevance. We rank and select the TopK relevant objects with small scores. The process is formulated as follows:

$$\begin{cases} \tilde{\mathcal{S}_{\mathcal{V}}} = \sigma(\boldsymbol{W}_{s_2} \mathcal{S}_{\mathcal{V}}); \\ \mathcal{G}'_{\mathcal{V}} = \text{TopK}(\mathcal{G}_{\mathcal{V}}, \tilde{\mathcal{S}_{\mathcal{V}}})|_{K=N}, \end{cases} \tag{2}$$

where $\boldsymbol{W}_{s_2}$ is a learnable parameter and $\sigma(\cdot)$ is sigmoid function and we set the number $K = N$ to keep the numbers of objects and tokens consistent. As a result, the object graph $\mathcal{G}'_{\mathcal{V}} = \{\mathcal{V}', \mathcal{E}_{\mathcal{V}' \leftrightarrow \mathcal{V}'}\}$ is updated, $\mathcal{V}' = \{v'_i\}_{i=1}^{N}$, and the token graph $\mathcal{G}_{\mathcal{T}}$ is unchanged.

3.2 Graph Pooling Inference

In TextVQA, numerous objects and tokens are detected in the image, but only a few are associated with the answer. The phenomena of the dense distribution of reduplicative objects and associated tokens are taken as a breakthrough point to help answer reasoning. This article proposes a graph pooling inference that dynamically downsizes semantically closed nodes (i.e., superfluous objects or relevant tokens) to narrow down the scope of answer inference. It involves three operations: *question-guided message passing*, *adaptive node dropping*, and *graph pooling*.

3.2.1 Question-guided Message Passing. The task requires reasoning the answers referring to a question, and thus we use the question semantics to assist in answer reasoning. Specifically, we apply a self-attention mechanism to observe the question. The sentence semantics is integrated by strengthening the semantics of important words in the question. Then, we obtain the sentence embedding as follows:

$$\tilde{q} = \sum_{i=1}^{\mathcal{L}_Q} \text{Softmax}(\boldsymbol{W}_{\boldsymbol{q}} q_i) \cdot q_i, \tag{3}$$

where $\boldsymbol{W}_{\boldsymbol{q}}$ is a learnable parameter.

The question-guided graph learning is performed on the graphs $\mathcal{G}'_{\mathcal{V}}$ and $\mathcal{G}_{\mathcal{T}}$ separately. Taking the object graph $\mathcal{G}'_{\mathcal{V}}$ as an example, we calculate the relation matrix $\mathbf{A} \in \mathbb{R}^{\mathrm{N} \times \mathrm{N}}$ for message passing

between objects in $\mathcal{G}'_{\mathcal{V}}$ and then update node v'_i as follows:

$$\begin{cases} \alpha = \text{Tanh}\left(\boldsymbol{W_e}\mathcal{E}_{\mathcal{V}'\leftrightarrow\mathcal{V}'} + \boldsymbol{W_{\tilde{q}}}\tilde{q}\right) \in \mathbb{R}^{N\times N}; \\ \mathbf{A} = \text{Softmax}\left(\boldsymbol{W_a}\alpha\right) \in \mathbb{R}^{N\times N}; \\ \mathcal{M}_{\{v'_j\}\rightarrow v'_i} = \sum_{j=1}^{N} \boldsymbol{W_m}\mathbf{a}_{ij}v'_i; \\ \tilde{v}_i = \boldsymbol{W_v}v'_i + \boldsymbol{W_n}\mathcal{M}_{\{v'_j\}\rightarrow v'_i}, \end{cases} \tag{4}$$

where $\{\boldsymbol{W_e}, \boldsymbol{W_{\tilde{q}}}, \boldsymbol{W_a}, \boldsymbol{W_m}, \boldsymbol{W_v}, \boldsymbol{W_n}\}$ are learnable parameters. We obtain the updated object graph $\tilde{\mathcal{G}}_{\mathcal{V}}$, where $\tilde{\mathcal{V}} = \{\tilde{v}_i\}_{i=1}^{N}$. Using the same implementation, we update the token graph $\tilde{\mathcal{G}}_{\mathcal{T}}$, where $\tilde{\mathcal{T}} = \{\tilde{t}_i\}_{i=1}^{N}$.

3.2.2 Adaptive Node Dropping. After the message passing, we obtain the explicit correlation of all nodes. The nodes with similar weights tend to be densely distributed along with redundancy or semantic association. We apply an adaptive node dropping strategy to downscale nodes for purifying semantics. Taking the object graph $\tilde{\mathcal{G}}_{\mathcal{V}}$ again, we get $\mathcal{Z}_{\mathcal{V}}$, which reflects the importance of the nodes $\{\tilde{v}_i\}$. We manually regulate the dropping ratio γ to preserve the $\lceil \gamma N \rceil$ nodes with the large values in $\mathcal{Z}_{\mathcal{V}}$ and drop the rest:

$$\begin{cases} \mathcal{Z}_{\mathcal{V}} = \text{Softmax}\left(\boldsymbol{W_z}\tilde{\mathcal{V}}\right) \in \mathbb{R}^{N\times 1}; \\ \tilde{\mathcal{G}}'_{\mathcal{V}} = \text{TopK}(\tilde{\mathcal{G}}_{\mathcal{V}}, \mathcal{Z})|_{K=\lceil \gamma N \rceil}, \end{cases} \tag{5}$$

where $\boldsymbol{W_z}$ is a learnable parameter.

3.2.3 Graph Pooling. To encapsulate the graph semantics for answer generation, we learn a global graph vector to represent the purified semantics in the image. To achieve a comprehensive graph representation, we implement both average pooling and max pooling on the node features of graph $\tilde{\mathcal{G}}'_v$ and obtain the $g_{mean} \in \mathbb{R}^d$ and $g_{max} \in \mathbb{R}^d$ vectors. By combining them, we obtain the final graph vector $g \in \mathbb{R}^{2\times d}$:

$$g = [g_{mean}, g_{max}] = \frac{1}{\lceil \gamma N \rceil} \sum_{i=1}^{\lceil \gamma N \rceil} \tilde{v}_i \; || \text{Max}_{i=1}^{\lceil \gamma N \rceil} \tilde{v}_i, \tag{6}$$

where $\text{Max}(\cdot)$ returns the channel-wise maximum value.

3.3 Dual-Path Graph Fusion

Since exploring the visual and linguistic properties of objects and scene texts is critical for answer inference, we excavate cues in both objects and tokens. In this work, we emphasize mining the key scene semantics on the premise of understanding the scene content with sufficient object and token features. To this end, we design a dual-path hierarchical graph architecture that implements progressive graph learning to make a semantic purification of objects and OCR tokens and discover the core semantics. In practice, our approach can effectively extract the semantics of key objects and tokens, especially by eliminating object duplication and improving token semantic connectivity.

3.3.1 Hierarchical Object Graph. Under the application of object detectors [42], multiple bounding boxes with inconsistent sizes cover the same subject, probably resulting in redundant objects, e.g., duplicate boundary boxes cover the far right player in Figure 2. These similar and close objects are always accompanied by dense and significant relationship weights, and these cumulative data biases may lead to a wrong answer prediction. To solve this issue, we apply an object-path hierarchical graph pooling inference to make full use of visual cues and suppress redundant noise by aggregating hierarchical graph semantics.

Specifically, we perform the hierarchical graph pooling learning on object graph $\mathcal{G}'_{\mathcal{V}}$ with L steps. First, to facilitate the description of the hierarchical graph architecture, we denote the three operations in Equations (4), (5), and (6) as GPI($\cdot$) function. The effect of GPI($\cdot$) is to drop superfluous nodes and encapsulate the graph semantics. At the lth graph layer, we perform GPI($\cdot$) on the previous graph $\tilde{\mathcal{G}}_{\mathcal{V}}^{(l-1)}$ to obtain the current graph $\tilde{\mathcal{G}}_{\mathcal{V}}^{(l)}$, and the final graph vector is obtained as follows:

$$g_{\mathcal{V}}^{(l)} = \text{GPI}\left(\tilde{\mathcal{G}}_{\mathcal{V}}^{(l-1)}\right) \in \mathbb{R}^{2\times d}, \tag{7}$$

Please note that at the initial stage, we set $\tilde{\mathcal{G}}_{\mathcal{V}}^{(1)} = \tilde{\mathcal{G}}'_{\mathcal{V}}$, and at the Lth layer, we perform GPI($\cdot$) without *adaptive node dropping* operation.

After L layers graph pooling inference, we obtain a series of graph vectors $g_{\mathcal{V}}^{(1)}, g_{\mathcal{V}}^{(2)}, \ldots, g_{\mathcal{V}}^{(L)}$, which represent the important objects of each graph layer. Here, we introduce a multi-layer semantic aggregation operation that enables the model to harness contextual information across different layers. This approach reinforces the semantic representation of significant visual elements and serves to mitigate graph over-smoothing [7]. Finally, all of them are fused into a global vector $\tilde{g}_{\mathcal{V}} \in \mathbb{R}^d$ to represent the purified semantics of objects:

$$\tilde{g}_{\mathcal{V}} = \boldsymbol{W}_1^{\boldsymbol{v}} g_{\mathcal{V}}^{(1)} + \boldsymbol{W}_2^{\boldsymbol{v}} g_{\mathcal{V}}^{(2)} + \cdots + \boldsymbol{W}_L^{\boldsymbol{v}} g_{\mathcal{V}}^{(L)}, \tag{8}$$

where $\{\boldsymbol{W}_1^{\boldsymbol{v}}, \boldsymbol{W}_2^{\boldsymbol{v}}, \ldots, \boldsymbol{W}_L^{\boldsymbol{v}}\} \in \mathbb{R}^{d\times 2d}$ are learnable weights.

3.3.2 Hierarchical Token Graph. The OCR tokens output by current OCR systems [5, 38] have different distribution characteristics from objects; each scene text is detected as a single OCR token and they are always distributed discretely in the image, but semantically similar tokens are distributed in close proximity [49]. If a phase or a sentence is composed of several tokens, then it is necessary to gather the semantics of isolated tokens together to understand the entire meaning, such as "samsung mobile" in Figure 2. We apply a token-path graph pooling inference to learn semantically similar tokens and gradually locate the answer tokens by reducing the graph scale. Same as the object graph, we perform the graph learning on $\tilde{\mathcal{G}}_{\mathcal{T}}$ with L steps and set $\tilde{\mathcal{G}}_{\mathcal{T}}^{(1)} = \tilde{\mathcal{G}}_{\mathcal{T}}$. Similarly, we obtain graph vectors $g_{\mathcal{T}}^{(1)}, g_{\mathcal{T}}^{(2)}, \ldots, g_{\mathcal{T}}^{(L)}$, and fuse them into a global vector $\tilde{g}_{\mathcal{T}} \in \mathbb{R}^d$ to represent the critical OCR token semantics.

3.3.3 Dual-Path Graph Fusion. Leveraging single objects or OCR tokens is insufficient to comprehend the scene text for answer prediction. As the above, it is necessary to aggregate the informative cues of objects and tokens. Finally, we combine both graph vectors into a joint graph vector $\tilde{g} \in \mathbb{R}^{2\times d}$,

$$\tilde{g} = \left[\boldsymbol{W}_{\boldsymbol{g}}^{\boldsymbol{v}} \tilde{g}_{\mathcal{V}}, \boldsymbol{W}_{\boldsymbol{g}}^{\boldsymbol{t}} \tilde{g}_{\mathcal{T}}\right] \in \mathbb{R}^{2\times d}, \tag{9}$$

where $\{\boldsymbol{W}_{\boldsymbol{g}}^{\boldsymbol{v}}, \boldsymbol{W}_{\boldsymbol{g}}^{\boldsymbol{t}}\}$ are learnable parameters.

3.4 Answer Generation and Model Optimization

3.4.1 Answer Decoder. Following References [25, 37], the answer decoder is composed of a four-layer transformer block and two classifiers—an object classifier ψ_{obj} and an OCR token classifier ψ_{tk}. We concatenate the question $\boldsymbol{Q}$, the nodes of $\tilde{\mathcal{V}}$ and $\tilde{\mathcal{T}}$, the graph embedding $\tilde{g}$, and a hidden state $o \in \mathbb{R}^d$ and input them into the transformer block as follows:

$$[\boldsymbol{Q}_{t+1}, \mathcal{V}_{t+1}, \mathcal{T}_{t+1}, \tilde{g}_{t+1}, o_{t+1}] = \Psi([\boldsymbol{W}_{\boldsymbol{q}}\boldsymbol{Q}_t, \boldsymbol{W}_{\boldsymbol{v}}\mathcal{V}_t, \boldsymbol{W}_{\boldsymbol{t}}\mathcal{T}_t, \boldsymbol{W}_{\boldsymbol{g}}\tilde{g}_t, \boldsymbol{W}_{\boldsymbol{o}}o_t]), \tag{10}$$

where $\{\boldsymbol{W}_{\boldsymbol{q}}, \boldsymbol{W}_{\boldsymbol{v}}, \boldsymbol{W}_{\boldsymbol{t}}, \boldsymbol{W}_{\boldsymbol{g}}, \boldsymbol{W}_{\boldsymbol{o}}\}$ are learnable parameters. The hidden state o_0 is initialized by the positional embedding [25], and where the initialization of $\mathcal{V}_0 = \tilde{\mathcal{V}}$, $\mathcal{T}_0 = \tilde{\mathcal{T}}$, and $g_0 = \tilde{g}$.

By iteratively performing the transformer block $L_{\mathcal{A}}$ times, we obtain the generated sequence $\boldsymbol{O} = [o_1, \ldots, o_{L_{\mathcal{A}}}] \in \mathbb{R}^{L_{\mathcal{A}} \times d}$.

At the tth decoding step, the classifier ψ_{obj} is optimized to predict the probability score y_t^{obj} over a preset object vocabulary. The classifier ψ_{tk} calculates the relevance score y_t^{tk} of o_t and the OCR token $\mathcal{T}_t$. We implement the Argmax function on y_t^{obj} and y_t^{tk} to predict the word y_t. The details are as follows:

$$\begin{cases} y_t^{obj} = \psi_{obj}(o_t) = \boldsymbol{W_1} o_t + \boldsymbol{b_1}; \\ y_t^{tk} = \psi_{tk}(\mathcal{T}_t, o_t) = (\boldsymbol{W_2}\mathcal{T}_t + \boldsymbol{b_2})^\top (\boldsymbol{W_3} o_t + \boldsymbol{b_3}); \\ y_t = \text{Argmax}([y_t^{obj}, y_t^{tk}]), \end{cases} \tag{11}$$

where $\{\boldsymbol{W_1}, \boldsymbol{W_2}, \boldsymbol{W_3}\}$ are learnable weights and $\{\boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3}\}$ are bias parameters. $L_{\mathcal{A}}$ is the length of answer. Thus, the generated answer is represented as $\mathcal{Y} = \{y_1, \ldots, y_{L_{\mathcal{A}}}\}$.

3.4.2 Model Optimization. For the network training, apparently, the answer prediction can be regarded as a multi-label classification problem with the classifier ψ_{obj} and the classifier ψ_{tk}. Binary cross-entropy loss function $\mathcal{L}_{bce}$ is suitable for multi-label classification and has been widely used in the TextVQA task [25, 37]. Besides, since the completely correct answers are too strict to be predicted, a new auxiliary policy gradient loss $\mathcal{L}_{pg}$ based on **Average Normalized Levenshtein Similarity (ANLS)** [3] is introduced for model optimization. ANLS measures the character-level composition similarity between the predicted answer and the ground truth,

$$ANLS(\mathcal{Y}, \tilde{\mathcal{Y}}) = 1 - \frac{NL(\mathcal{Y}, \tilde{\mathcal{Y}})}{max(|\mathcal{Y}|, |\tilde{\mathcal{Y}}|)}, \tag{12}$$

where $NL(\cdot)$ is the normalized Levenshtein distance [31]. If it is less than 0.5, then ANLS is set to 0 in this task following the practice in Reference [3].

We detail the objective terms $\mathcal{L}_{bce}$ and $\mathcal{L}_{pg}$. The total objective is formulated as follows:

$$\begin{cases} \mathcal{L}_{bce} = -\tilde{\mathcal{Y}} \log(\sigma(\mathcal{Y})) - (1 - \tilde{\mathcal{Y}}) \log(1 - \sigma(\mathcal{Y})); \\ \mathcal{L}_{pg} = -\log(\sigma(\mathcal{Y})) \cdot \text{ANLS}(\mathcal{Y}, \tilde{\mathcal{Y}}); \\ \mathcal{L} = \mathcal{L}_{bce} + \lambda \mathcal{L}_{pg}, \end{cases} \tag{13}$$

where $\tilde{\mathcal{Y}}$ denotes the ground-truth answer and λ is a tradeoff hyperparameter.

4 EXPERIMENT

4.1 Datasets and Metrics

Datasets. We experiment on two benchmark datasets. (1) *TextVQA* [44] is collected from large-scale Open Images v3 [29] dataset, which contains 28,408 images and 45,336 questions. It is split into the train/valid/test sets with questions 34,602/5,000/5,734. In this dataset, questions are interested in visual objects or scene texts in the images. In particular, up to 39% (approximately 18,000 QA pairs) of answers do not involve any OCR token. (2) *ST-VQA* [3] is a mixed dataset; the images is selected from six datasets for the tasks of VQA, object detection, image captioning, scene text recognition, scene text retrieval, and text detection—VizWiz [19], ImageNet [9], Visual Genome [30], COCO-Text [47], IIIT Scene Text Retrieval [41], and ICDAR 2013/2015 [27, 28]. Following the protocol [25, 56], we experiment on the open dictionary task of the ICDAR ST-VQA Challenge,[1] containing 21,892 images and 30,144 questions and being divided into the train/valid/test sets with images 17,028/1,893/2,971 [25]. Different from the TextVQA dataset, the ST-VQA dataset emphasizes the

[1]ICDAR ST-VQA Challenge: https://rrc.cvc.uab.es/?ch=11&com=tasks

Table 1. Main Comparison on TextVQA Dataset

Method	Venue	OCR System	Extra Data	Val Acc	Test Acc
Attention-based models					
LoRRA [44]	*CVPR'2019*	Rosetta-ml	—	26.56	27.63
SSBaseline [63]	*AAAI'2021*	SBD-Trans	—	43.95	44.72
SSBaseline [63]	*AAAI'2021*	SBD-Trans	ST-VQA	45.53	45.66
Transformer-based models					
M4C [25]	*CVPR'2020*	Rosetta-en	—	39.40	39.01
M4C [25]	*CVPR'2020*	Rosetta-en	ST-VQA	40.55	40.46
LaAP-Net [20]	*COLING'2020*	Rosetta-en	—	40.68	40.54
LaAP-Net [20]	*COLING'2020*	Rosetta-en	ST-VQA	41.02	40.54
PAT [60]	*ACM MM'2022*	Google-OCR	—	42.80	43.41
Representation Learning-based models					
BOV [57]	*ACM MM'2022*	SBD-Trans	—	44.87	45.63
BOV [57]	*ACM MM'2022*	SBD-Trans	ST-VQA	46.24	46.96
Graph-based models					
MM-GNN [15]	*CVPR'2020*	Rosetta-ml	—	31.44	31.10
CRN [37]	*ACM MM'2020*	Rosetta-en	—	40.39	40.96
SA-M4C [26]	*ECCV'2020*	GoogleOCR	—	43.90	—
SA-M4C [26]	*ECCV'2020*	GoogleOCR	ST-VQA	45.40	44.60
TIG [35]	*PR'2021*	Rosetta-en	—	40.45	—
SMA [14]	*TPAMI'2021*	SBD-Trans	—	43.74	44.29
SMA [14]	*TPAMI'2021*	SBD-Trans	ST-VQA	44.58	45.51
SSGN [62]	*TIP'2023*	Microsoft-OCR	ST-VQA	46.85	47.16
GPIN (Ours)	—	Rosetta-en	—	42.20	42.00
GPIN (Ours)	—	SBD-Trans	—	45.70	46.53
GPIN (Ours)	—	SBD-Trans	ST-VQA	<u>47.38</u>	47.24
GPIN (Ours)	—	Microsoft-OCR	—	46.61	<u>47.43</u>
GPIN (Ours)	—	**Microsoft-OCR**	**ST-VQA**	**48.12**	**48.12**
Pe-training technologies[2]					
TAP* [56]	*CVPR'2021*	Microsoft-OCR	ST-VQA	50.57	50.71
TAP* [56]	*CVPR'2021*	Microsoft-OCR	Multi-Source	54.71	53.97
LOGOS† [40]	*ICCVW'2021*	Microsoft-OCR	VG	50.79	50.65
LOGOS† [40]	*ICCVW'2021*	Microsoft-OCR	ST-VQA, VG	51.53	51.08
LaTr-Base△ [2]	*CVPR'2022*	Amazon-OCR	IDL	58.03	58.86

Bold and underline indicate the best and second-best results among the non-pre-trained models, respectively.

questions that have to be answered with scene texts. In other words, answering a question in TextVQA requires a thorough understanding of all scene text of an image, whereas ST-VQA relies on OCR tokens.

Evaluation Metrics. We adopt the accuracy (Acc) and the ANLS as metrics, where ANLS is primarily proposed for dataset.

4.2 Implementation Details

Following the previous work [25, 56], we use the pre-trained BERT [10] to extract textual features of the question and Faster R-CNN [42] to detect objects in the image. Each detected object obtains appearance feature $\mathrm{v}_{\mathrm{i}}^{\mathrm{a}} \in \mathbb{R}^{2048}$ and boundary box feature $\mathrm{v}_{\mathrm{i}}^{\mathrm{b}} \in \mathbb{R}^{4}$. They are encoded by a separate fully connected layer and then added into the initial object feature $\mathrm{v}_i \in \mathbb{R}^{768}$, i.e., $\mathrm{v}_i = \boldsymbol{W}_{f_1}^{\boldsymbol{v}}(\mathrm{v}_{\mathrm{i}}^{\mathrm{a}}) + \boldsymbol{W}_{f_2}^{\boldsymbol{v}}(\mathrm{v}_{\mathrm{i}}^{\mathrm{b}})$, where $\boldsymbol{W}_{f_1}^{\boldsymbol{v}}$ and $\boldsymbol{W}_{f_2}^{\boldsymbol{v}}$ are fully connected layers. As for OCR detection, each OCR token output by OCR systems [5, 38] consists of four aspects, i.e., FastText feature $\mathrm{t}_{\mathrm{i}}^{\mathrm{f}} \in \mathbb{R}^{300}$ [4],

Table 2. Main Comparison on ST-VQA Dataset

Method	Venue	OCR System	Extra Data	Val Acc	Val ANLS	Test ANLS
Attention-based models						
SSBaseline [63]	*AAAI'2021*	SBD-Trans	—	—	—	0.509
SSBaseline [63]	*AAAI'2021*	SBD-Trans	TextVQA	—	—	0.550
Transformer-based models						
M4C [25]	*CVPR'2020*	Rosetta-en	—	38.05	0.472	0.462
LaAP-Net [20]	*COLING'2020*	Rosetta-en	—	39.74	0.497	0.485
PAT [60]	*ACM MM'2022*	GoogleOCR	—	41.10	—	0.508
Representation learning-based models						
BOV [57]	*ACM MM'2022*	Rosetta-en	-	40.18	0.500	0.472
Graph-based models						
CRN [37]	*ACM MM'2020*	Rosetta-en	—	—	—	0.483
SA-M4C [26]	*ECCV'2020*	GoogleOCR	—	42.23	0.512	0.504
TIG [35]	*PR'2021*	Rosetta-en	—	41.52	0.516	0.505
SMA [14]	*TPAMI'2021*	Rosetta-en	—	—	—	0.486
SSGN [62]	*TIP'2023*	Microsoft-OCR	TextVQA	48.81	0.589	0.573
GPIN (Ours)	—	Rosetta-en	—	41.55	0.520	0.508
GPIN (Ours)	—	SBD-Trans	—	42.58	0.530	0.513
GPIN (Ours)	—	SBD-Trans	TextVQA	45.50	0.563	0.553
GPIN (Ours)	—	Microsoft-OCR	—	<u>47.26</u>	<u>0.570</u>	<u>0.562</u>
GPIN (Ours)	—	**Microsoft-OCR**	**TextVQA**	**50.13**	**0.598**	**0.587**
Pe-training technologies[2]						
TAP* [56]	*CVPR'2021*	Microsoft-OCR	—	45.29	0.551	0.543
TAP* [56]	*CVPR'2021*	Microsoft-OCR	Multi-Source	50.83	0.598	0.597
LOGOS† [40]	*ICCVW'2021*	Microsoft-OCR	VG	44.10	0.535	0.522
LOGOS† [40]	*ICCVW'2021*	Microsoft-OCR	TextVQA, VG	48.63	0.581	0.579
LaTr-Base△ [2]	*CVPR'2022*	Amazon-OCR	IDL	58.41	0.675	0.668

Bold and underline indicate the best and second-best results among the non-pre-trained models, respectively.

pyramidal histogram of characters feature $\mathrm{t}_{\mathrm{i}}^{\mathrm{p}} \in \mathbb{R}^{604}$ [1], appearance feature $\mathrm{t}_{\mathrm{i}}^{\mathrm{a}} \in \mathbb{R}^{2048}$, and bounding box feature $\mathrm{t}_{\mathrm{i}}^{\mathrm{b}} \in \mathbb{R}^{4}$. All the features are encoded by a separate fully connected layer and then added to obtain the initial OCR token feature $\mathrm{t}_i \in \mathbb{R}^{768}$, i.e., $\mathrm{t}_i = \boldsymbol{W}_{f_1}^{\boldsymbol{t}}(\mathrm{t}_{\mathrm{i}}^{\mathrm{f}}) + \boldsymbol{W}_{f_2}^{\boldsymbol{t}}(\mathrm{t}_{\mathrm{i}}^{\mathrm{p}}) + \boldsymbol{W}_{f_3}^{\boldsymbol{t}}(\mathrm{t}_{\mathrm{i}}^{\mathrm{a}}) + \boldsymbol{W}_{f_4}^{\boldsymbol{t}}(\mathrm{t}_{\mathrm{i}}^{\mathrm{b}})$, where $\boldsymbol{W}_{f_1}^{\boldsymbol{t}}$, $\boldsymbol{W}_{f_2}^{\boldsymbol{t}}$, $\boldsymbol{W}_{f_3}^{\boldsymbol{t}}$, and $\boldsymbol{W}_{f_4}^{\boldsymbol{t}}$ are fully connected layers. The unified feature dimension is set to d = 768. For both TextVQA and ST-VQA datasets, each question is truncated to a length of L_Q = 20 words. For each image, we detect N= 50 OCR tokens and M=100 objects with high probabilities.

For model implementation, following the practice in Reference [37], a two-layer transformer with 12 heads is performed as the feature encoder in Section 3.1.1, and a four-layer transformer with 12 heads is applied as the answer decoder in Section 3.4.1. For the object refinement in Section 3.1.3, the dimension of difference representation is set to p = 256, and we downscale the size of the object set into N = 50. For the graph pooling inference in Section 3.3, we set the layer number of graph pooling L = 3 and the dropping ratio to γ = 40%. Besides, we set the step number of answer decoding to $L_{\mathcal{A}}$ = 12. As for model optimization, we adopt the Adam optimizer with

[2]Symbol * denotes training the model with three pre-training tasks, i.e., MLM (masked language modeling), ITM (image-text matching), and RPP (relative position prediction) tasks. Symbol † denotes training the model with a question-visual grounding task. Symbol △ denotes training the model with a layout-aware de-noising pre-training task, which includes the two-dimensional spatial embedding. *Multi-Source* is the combination of TextVQA, ST-VQA, TextCaps, and non-public OCR-CC [56] datasets. *VG* denotes the Visual Genome dataset [30]. *IDL*[3] denotes the industrial document library, which produces about 13M documents, translating to about 64M pages of various document images. According to statistics [33], pre-training models need roughly 70 times as much training data and 20 times as much training time as non-pre-training models.
[3]Industrial Document Library: https://www.industrydocuments.ucsf.edu/

Table 3. Ablation Studies of Different Graph Structures on TextVQA Dataset

Method	Val Acc	Test Acc
Backbone	45.38	46.35
Backbone w/ OR (Object Refinement)	45.85	46.65
+ OG (Object Graph)	46.45	47.39
+ TG (Token Graph)	47.89	47.80
+ OTG (Object-Token Interactive Graph)	46.33	46.36
GPIN (Ours)	**48.12**	**48.12**

Bold represents the result of the best-performing variant.

the learning rate of 1e-4 and the tradeoff loss parameter is set to $\lambda = 1$. We multiply the learning rate by 0.1 at 10,000 and 21,000 iterations for a total of 24,000 iterations.

4.3 Comparison with the States of the Art

4.3.1 Results on TextVQA. In Table 1, our method achieves promising performance compared with the state-of-the-art methods. Compared with the optimal attention-based model *SSBaseline* [63], the proposed *GPIN* exceeds it by 1.85% on *Val Acc* and 1.58% on *Test Acc*. Compared with the top transformer-based model *PAT* [60], *GPIN* improves by 4.58% on *Val Acc* and 3.83% on the *Test Acc*. Since our method is a graph-based model, we compare it with the best-performing graph model *SSGN* [62]; obviously, ours has an absolute advantage of 1.27% on *Val Acc*. Among these methods without pe-training technologies, the state-of-the-art performance is reported by a representation learning-based model *BOV* [57], *GPIN* also surpasses it by 0.90% on *Test Acc*, achieving the best. In addition, the performance of *GPIN* can be further improved when stronger pre-training tasks and more training data are introduced.

4.3.2 Results on ST-VQA. ST-VQA dataset is complex and challenging as it involves six datasets covering completely different text scenes and tasks. In this dataset, OCR tokens are frequently taken as the answers to the questions, this requires recognizing and understanding scene texts more accurately. Our method shows a significant superiority in this dataset. In Table 2, when merely using Microsoft-OCR[4] without extra data, *GPIN* already upgrades 1.97% of *Acc* and 0.019 of *Test ANLS* compared with the pe-training model *TAP* [56], where *TAP* uses a quite large-scale training data for improving performance. When further adding the TextVQA data, *GPIN* reaches 50.13%/0.598/0.587 on *Val Acc*/*Val ANLS*/*Test ANLS*.

4.4 Ablation Studies

4.4.1 Different Graph Structures. We consider different graph structures in Table 3, i.e., single **object graph (OG)**, single **token graph (TG)**, **object-token interactive graph (OTG)**, and our GPIN *(the combination of OG and TG)*. In OTG, the intra-object and intra-token relationship are not considered, while it considers the relationship between object and OCR token. We fix *Backbone w/ OR* as the base model, which improves by 0.47% on *Val Acc* on the basis of *Backbone*. Either *OG* or *TG* improves the performance compared with *Backbone w/ OR*. Notably, *TG* brings conspicuous improvement, increasing 2.04% on *Val Acc*. Besides, to demonstrate the effect of our dual-path graph architecture, we compare the interactive OTG and GPIN; GPIN is significantly increased by 1.79% on *Val Acc*. Since objects and tokens have each characteristic, it indicates that dual-path graph fusion plays more nontrivial roles than interactive correlation learning.

[4]Microsoft-OCR API: https://azure.microsoft.com

Table 4. Ablation Studies of Graph Pooling Inference Abbreviated as GPI (Including both Adaptive Node Dropping and Graph Pooling) on TextVQA and ST-VQA Datasets

Method	TextVQA				ST-VQA					
	w/o GPI		w/ GPI		w/o GPI			w/ GPI		
	Val Acc	Test Acc	Val Acc	Test Acc	Val Acc	Val ANLS	Test ANLS	Val Acc	Val ANLS	Test ANLS
OG (Object Graph)	45.88	46.70	46.45	47.39	46.54	0.571	0.552	48.13	0.581	0.575
TG (Token Graph)	46.98	46.79	47.89	47.80	47.72	0.578	0.566	49.32	0.589	0.581
GPIN (Both Graphs)	**47.11**	**47.09**	**48.12**	**48.12**	**48.56**	**0.583**	**0.573**	**50.13**	**0.598**	**0.587**

Bold represents the result of the best-performing variant.

Table 5. Ablation Studies of Adaptive Node Dropping on TextVQA and ST-VQA Dataset

Method	TextVQA		ST-VQA		
	Val Acc	Test Acc	Val Acc	Val ANLS	Test ANLS
w/ random	46.68	46.84	48.60	0.585	0.575
w/o adaptive	47.03	47.05	49.85	0.591	0.582
GPIN (Ours)	**48.12**	**48.12**	**50.13**	**0.598**	**0.587**

Bold represents the result of the best-performing variant.

4.4.2 Graph Pooling Inference. In Table 4, we test whether we need the **graph pooling inference (GPI)** on TextVQA and ST-VQA datasets. The variant *w/o GPI* denotes the setting of the hierarchical graph model without *adaptive node dropping* and *graph pooling* operations, i.e., three-layer fully connected graph. On TextVQA dataset, for the single OG, the setting of w/ GPI improves the *Test Acc* by 0.69% over w/o GPI, and the single TG can improve 1.01% on *Test Acc*. By combining them, the performance is also consistently and steadily improved by 1.03% on *Test Acc*. Besides, on the ST-VQA dataset, in OG, w/ GPI improves by 1.59% on *Val Acc* over w/o GPI, and in TG, it can improve by 1.60% over *Val Acc*. Furthermore, the performance of our GPIN improves by 1.57% over *Val Acc*. These results on both datasets show that our graph pooling inference could suppress visual noise in reasoning by dynamically discarding superfluous nodes.

4.4.3 Adaptive Node Dropping. To verify the validity of the adaptive node dropping strategy, we compare random node dropping and our adaptive node dropping in graph pooling inference. In Table 5, *w/ random* is conducted to take random node dropping operation to replace the adaptive node dropping operation of GPIN; *w/o adaptive* is supplemented to test the variant of GPIN without the adaptive node dropping operation, i.e., a three-layer fully connected graph with graph pooling operation. Compare with *w/ random*, our *GPIN* performs obviously better by lifting *Val Acc* with 1.44% and 1.53% on TextVQA and ST-VQA datasets, respectively. Compared with w/ random, w/o adaptive improves 0.35%/0.21% over *Val Acc*/*Test Acc* of TextVQA. The results show that the arbitrary deletion of nodes may result in the loss of important visual elements. Besides, we test *dropping ratio* γ. As shown in Table 6, $\gamma = 40\%$ is the optimal setup. This indicates neither too-large nor too-small γ is inappropriate for node dropping. When γ is large, redundant nodes along with some useful nodes are dropped, whereas when it is small, redundant nodes are not pruned enough.

4.4.4 Hierarchical Graph Layers. We test the effect of each graph layer in Table 7. Among single graph layers, w/ $\boldsymbol{g}^{(1)}$, w/ $\boldsymbol{g}^{(2)}$, and w/ $\boldsymbol{g}^{(3)}$ denote the obtained graph vector in the respective three graph layers. Among them, the result of w/ $\boldsymbol{g}^{(2)}$ is the best, reaching 47.41% on *Val Acc* and 47.71% on *Test Acc*. When merely w/ $\boldsymbol{g}^{(1)}$ is available, there may be a lot of redundant noise interfering with answer reasoning. When only setting w/ $\boldsymbol{g}^{(3)}$, it may result in insufficient visual cues. We further test them in pairs, and the overall performance is further improved. When superimposing

Table 6. Ablation Studies of Node Dropping Ratio γ on TextVQA and ST-VQA Datasets

γ	TextVQA		ST-VQA		
	Val Acc	Test Acc	Val Acc	Val ANLS	Test ANLS
20%	47.33	47.09	49.91	0.596	0.584
40%	**48.12**	**48.12**	**50.13**	**0.598**	0.587
60%	47.87	47.69	50.01	0.596	**0.588**
80%	47.26	47.36	49.65	0.587	0.587

Bold represents the result of the best-performing variant.

Table 7. Ablation Studies of Hierarchical Graph Layers on TextVQA Dataset

Method	l=1	l=2	l=3	Val Acc	Test Acc
w/ $g^{(1)}$	✓	—	—	47.11	47.18
w/ $g^{(2)}$	—	✓	—	47.41	47.71
w/ $g^{(3)}$	—	—	✓	46.49	47.11
w/ $g^{(1)}$&$g^{(2)}$	✓	✓	—	47.56	47.78
w/ $g^{(2)}$&$g^{(3)}$	—	✓	✓	47.88	47.72
GPIN ($g^{(1)}$&$g^{(2)}$&$g^{(3)}$)	✓	✓	✓	**48.12**	**48.12**

Table 8. Ablation Studies of Graph Pooling Operation on TextVQA Dataset

Method	g_{mean}	g_{max}	Val Acc	Test Acc
w/ g_{max}	—	✓	47.63	47.83
w/ g_{mean}	✓	—	47.58	47.59
GPIN (g_{mean}&g_{max})	✓	✓	**48.12**	**48.12**

Table 9. Comparison of Graph Attention and Cross Attention in Question-guided Message Passing on TextVQA and ST-VQA Datasets with Microsoft-OCR System

Method	TextVQA		ST-VQA		
	Val Acc	Test Acc	Val Acc	Val ANLS	Test ANLS
cross-attention	46.60	46.53	48.56	0.574	0.562
GPIN (graph attention)	**48.12**	**48.12**	**50.13**	**0.598**	**0.587**

multiple layers, the model can fully mine visual cues and suppress the interference of redundancy, reaching the optimum performance.

4.4.5 Graph Pooling Operation. We test the effect of graph pooling operations in Table 8. Compared with GPIN w/o GPI in Table 4, there is a performance boost when either average pooling w/ g_{mean} or max pooling w/ g_{max} operation is performed. When both operations are applied simultaneously, the performance improvement is greatest. Compared with w/o g_{max}, GPIN increases by 0.49% on *Val Acc* and 0.29% *Test Acc*, respectively. Compared with **w/o** g_{mean}, GPIN improves by 0.54% on *Val Acc* and 0.53% *Test Acc*. The possible reason is that *average pooling* helps gather semantics of the neighbors and their adjacency around the nodes; *max pooling* emphasizes the role of salient semantics in the node features. Our method considers both node characteristics and their adjacency semantics to achieve the optimal benefits.

Table 10. Performance of GPI under Different Number Settings of Objects and OCR Tokens with Microsoft-OCR Features on TextVQA Dataset

Method	w/o GPI		w/ GPI	
	Val Acc	Test Acc	Val Acc	Test Acc
$M = 40, N = 20$	46.10	46.23	46.75	46.89
$M = 60, N = 30$	46.48	46.63	47.09	47.13
$M = 80, N = 40$	46.82	46.89	47.43	47.62
$M =$ **100**, $N =$ **50 (Ours)**	**47.11**	**47.09**	**48.12**	**48.12**

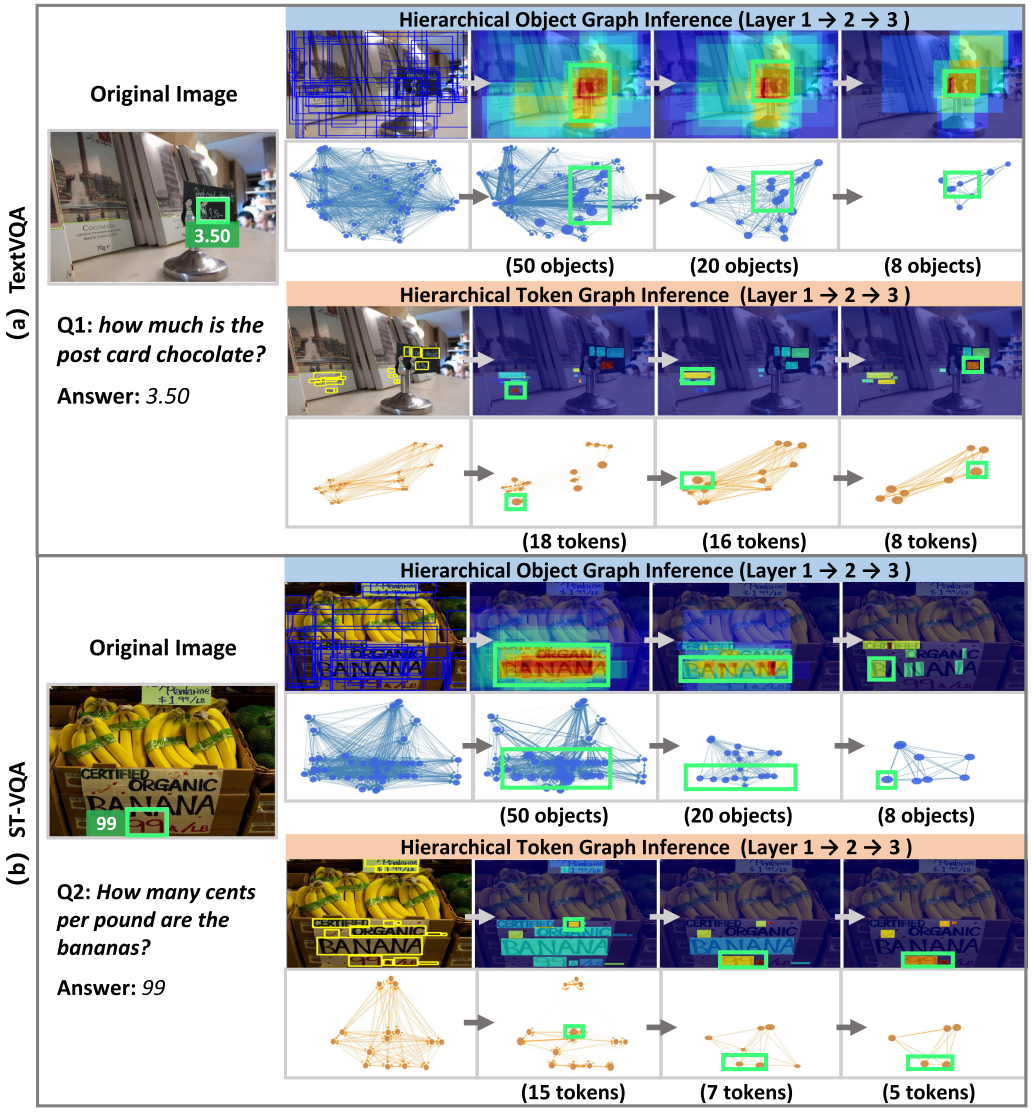

Fig. 3. Visualization of answer prediction. *Layer 1 → 2 → 3* denotes three graph layers. The results display that our model progressively encapsulates the relevant semantics in either the object graph or token graph for predicting correct answers.

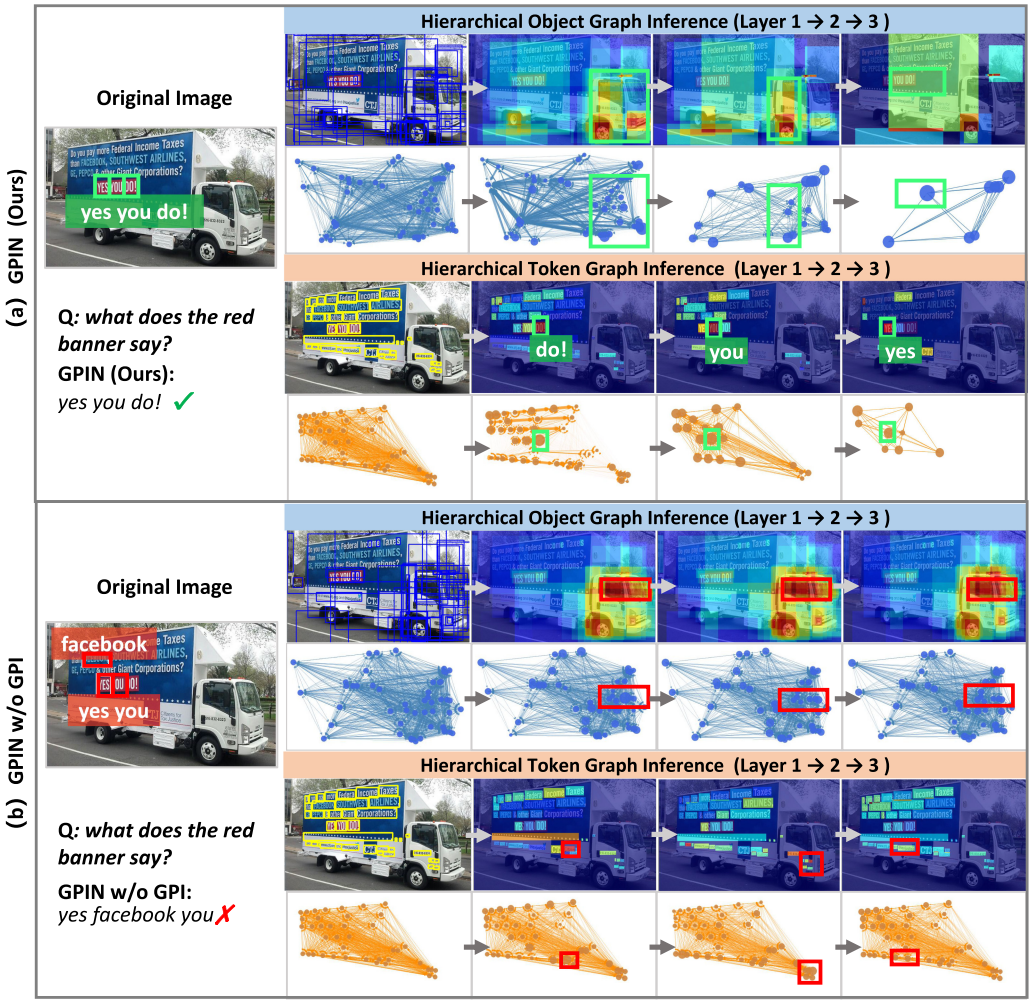

Fig. 4. Visualization of GPI. The comparison shows that the GPI module can remove redundant nodes and establish a more explicit relationship in the graph for answer reasoning.

4.4.6 Question-guided Message Passing. We conduct experiments to compare graph attention and cross-attention [6]. In Table 9, *cross-attention* denotes the variant of GPIN where the question-guided message passing operation is replaced by cross-attention [6]. Specifically, we calculate the cross-attention between the question and objects (tokens) to update the object (token) features. Compared with cross-attention, our graph attention has an improvement of 1.52%/1.59% on *Val Acc*/*Test Acc* of TextVQA and 1.57%/0.024/0.025 on *Val Acc*/*Val ANLS*/*Test ANLS* of ST-VQA. The results show that graph attention performs better than cross-attention for relational learning.

4.4.7 Different Object and Token Number Setting. To demonstrate the effectiveness of our method, we conduct ablation experiments on graph pooling inference with four sets of objects M and OCR tokens N number settings, i.e., M= 40, 60, 80, and 100; N= 20, 30, 40, and 50. We conclude the following two conclusions: (1) Sufficient object and token features are beneficial to improve the visual understanding ability of the model. As shown in Table 10, among the four variants of w/ GPI, compared to the setting "M = 40, N = 20", the performance of M = 100, N = 50"

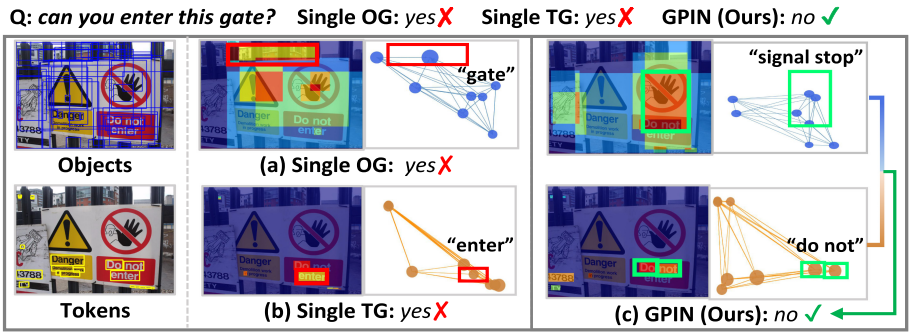

Fig. 5. Visualization of dual-path graph fusion. We discuss *single OG*, *single TG*, and *GPIN (dual-path)*. The results demonstrate that the object graph and token graph are complementary for answer prediction.

exhibits an improvement of 1.37% on the val set and 1.23% on the test set. (2) Our GPI exhibits effectiveness across various object and token number settings. In Table 1, when setting "$M = 40$, $N = 20$", compared with w/o GPI, w/ GPI improves 0.65% on the val set and 0.66% on the test set.

4.5 Visualization Analysis

4.5.1 Answer Prediction. We select two samples separately from TextVQA and ST-VQA datasets. As an example (a) in Figure 3, both the object graph and token graph pay more attention to the correct answer "3.50" even though it appears in the image at an extremely small size. Turning to example (b), the object graph pays attention to the scene text "banana", while the token graph focuses on "99". Both "banana" and "99" are strongly relevant to the question. Under the joint semantics, GPIN predicts the correct cents "99".

4.5.2 Graph Pooling Inference. We compare GPIN w/o GPI and GPIN in Figure 4 to display the effect of graph pooling inference. For GPIN w/o GPI, redundant objects determine the highly responsive region (the car head), while relatively average relationships among OCR tokens lead to attending on words "facebook" and "yes you". In contrast, GPIN effectively drops redundant nodes and makes the relationship in the graph more explicit for answer prediction. Both object and token graphs consistently highlight "yes you do".

4.5.3 Adaptive Node Dropping. As shown in Figure 3 again, redundant objects are strictly pruned with the dropping ratio γ ($N = 50 \rightarrow 20 \rightarrow 8$). However, OCR tokens are to be pruned flexibly. A fact is that the maximum number of OCR tokens output by the OCR system is $N = 50$. Actually, the initial token number may be much less than 50 as shown in Figure 3(a) and (b). We fill the token with "null" nodes up to 50 nodes and then count the relevant nodes. When adaptively dropping the nodes, the number of OCR tokens is dynamic.

4.5.4 Dual-Path Graph Fusion. At last, we test the dual-path graph structure in Figure 5. Given a question "can you enter this gate?", the single OG attends to the visual object "gate" and answers "yes". As for the single TG, it observes the word "enter" and answers "yes". However, our dual path GPIN predicts the correct answer "do not enter", owing to the OG pays attention to visual symbol "stop" and the TG attends scene texts "do not".

5 CONCLUSION

In this article, we propose a novel GPIN for TextVQA. We develop a dynamical evolutionary graph learning model that progressively and adaptively purifies the visual content to dig out the core

semantics for answer reasoning. We evaluate GPIN on two benchmark datasets and conduct extensive ablation studies to demonstrate the validity of our model. Experimental results show the superiority of our method over the state-of-the-art models.

REFERENCES

[1] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. 2014. Word spotting and recognition with embedded attributes. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 12 (2014), 2552–2566.

[2] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R. Manmatha. 2022. Latr: Layout-aware transformer for scene-text vqa. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'22)*. 16548–16558.

[3] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marcal Rusinol, C. V. Jawahar, Ernest Valveny, and Dimosthenis Karatzas. 2019. Scene text visual question answering. In *Proceedings of the International Conference on Computer Vision (ICCV'19)*. 4290–4300.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'17)*, 135–146.

[5] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. 2018. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the ACM Knowledge Discovery and Data Mining (SIGKDD'18)*. 71–79.

[6] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. 2021. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR'21)*. 357–366.

[7] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'20)*. 3438–3445.

[8] Corentin Dancette, Rémi Cadène, Damien Teney, and Matthieu Cord. 2021. Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering. In *Proceedings of the International Conference on Computer Vision (ICCV'21)*. 1554–1563.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. 248–255.

[10] Jacob Devlin, MingWei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'19)*. 4171–4186.

[11] Yang Ding, Jing Yu, Bangchang Liu, Yue Hu, Mingxin Cui, and Qi Wu. 2022. MuKEA: Multimodal knowledge extraction and accumulation for knowledge-based visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'22)*. 5079–5088.

[12] Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, Xun Wang, and Meng Wang. 2021. Dual encoding for video retrieval by text. *IEEE Trans. Pattern Anal. Mach. Intell.* (2021), 4065–4080.

[13] Chengyang Fang, Jiangnan Li, Liang Li, Can Ma, and Dayong Hu. 2023. Separate and locate: Rethink the text in text-based visual question answering. In *Proceedings of the ACM International Conference on Multimedia (ACM MM'23)*. 4378–4388.

[14] Chenyu Gao, Qi Zhu, Peng Wang, Hui Li, Yuliang Liu, Anton Van den Hengel, and Qi Wu. 2021. Structured multimodal attentions for textvqa. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 12 (2021), 9603–9614.

[15] Difei Gao, Ke Li, Ruiping Wang, Shiguang Shan, and Xilin Chen. 2020. Multi-modal graph neural network for joint reasoning on vision and scene text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'20)*. 12746–12756.

[16] Feng Gao, Q. Ping, Govind Thattai, Aishwarya N. Reganti, Yingting Wu, and Premkumar Natarajan. 2022. Transform-retrieve-generate: Natural language-centric outside-knowledge visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'22)*, 5057–5067.

[17] Dan Guo, Hui Wang, and Meng Wang. 2021. Context-aware graph inference with knowledge distillation for visual dialog. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 10 (2021), 6056–6073.

[18] Dan Guo, Hui Wang, Hanwang Zhang, Zheng-Jun Zha, and Meng Wang. 2020. Iterative context-aware graph inference for visual dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*. 10055–10064.

[19] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*. 3608–3617.

[20] Wei Han, Hantao Huang, and Tao Han. 2020. Finding the evidence: Localization-aware answer prediction for text visual question answering. In *Proceedings of the International Conference on Computational Linguistics (COLING'20)*. 3118–3131.

[21] Xinzhe Han, Shuhui Wang, Chi Su, Qingming Huang, and Qi Tian. 2021. Greedy gradient ensemble for robust visual question answering. In *Proceedings of the International Conference on Computer Vision (ICCV'21)*. 1584–1593.

[22] Shamanthak Hegde, Soumya Jahagirdar, and Shankar Gangisetty. 2023. Making the V in text-VQA matter. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5579–5587.

[23] Yu-Jung Heo, Eun-Sol Kim, Woo Suk Choi, and Byoung-Tak Zhang. 2022. Hypergraph transformer: Weakly-supervised multi-hop reasoning for knowledge-based visual question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'22)*. 373–390.

[24] Jun Hu, Quan Fang, Shengsheng Qian, and Changsheng Xu. 2020. Multi-modal attentive graph pooling model for community question answer matching. In *Proceedings of the ACM International Conference on Multimedia (ACM MM'20)*. 3505–3513.

[25] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. 2020. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'20)*. 9992–10002.

[26] Yash Kant, Dhruv Batra, Peter Anderson, Alexander Schwing, Devi Parikh, Jiasen Lu, and Harsh Agrawal. 2020. Spatially aware multimodal transformers for textvqa. In *Proceedings of the European Conference on Computer Vision (ECCV'20)*. 715–732.

[27] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. 2015. ICDAR 2015 competition on robust reading. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR'15)*. 1156–1160.

[28] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. 2013. ICDAR 2013 robust reading competition. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR'13)*. 1484–1493.

[29] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami AbuElHaija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. 2017. Openimages: A Public Dataset for Large-scale Multi-label and Multi-class Image Classification. Retrieved from https://github.com/openimages

[30] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.* 123 (2017), 32–73.

[31] Vladimir I. Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*. Vol. 10. Soviet Union, 707–710.

[32] Bingjia Li, Jie Wang, Minyi Zhao, and Shuigeng Zhou. 2022. Two-stage multimodality fusion for high-performance text-based visual question answering. In *Proceedings of the Asian Conference on Computer Vision (ACCV'22)*. 4143–4159.

[33] Hao Li, Jinfa Huang, Peng Jin, Guoli Song, Qi Wu, and Jie Chen. 2023. Weakly-supervised 3D spatial reasoning for text-based visual question answering. *IEEE Trans. Image Process.* 32 (2023), 3367–3382.

[34] Juncheng Li, Junlin Xie, Long Qian, Linchao Zhu, Siliang Tang, Fei Wu, Yi Yang, Yueting Zhuang, and Xin Eric Wang. 2022. Compositional temporal grounding with structured variational cross-graph correspondence learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'22)*. 3022–3031.

[35] Xiangpeng Li, Bo Wu, Jingkuan Song, Lianli Gao, Pengpeng Zeng, and Chuang Gan. 2022. Text-instance graph: Exploring the relational semantics for text-based visual question answering. *Pattern Recogn.* 124 (2022), 108455.

[36] Yicong Li, Xun Yang, An Zhang, Chun Feng, Xiang Wang, and Tat-Seng Chua. 2023. Redundancy-aware transformer for video question answering. In *Proceedings of the ACM International Conference on Multimedia (ACM MM'23)*. 3172–3180.

[37] Fen Liu, Guanghui Xu, Qi Wu, Qing Du, Wei Jia, and Mingkui Tan. 2020. Cascade reasoning network for text-based visual question answering. In *Proceedings of the ACM International Conference on Multimedia (ACM MM'20)*. 4060–4069.

[38] Yuliang Liu, Sheng Zhang, Lianwen Jin, Lele Xie, Y. Wu, and Zhepeng Wang. 2019. Omnidirectional scene text detection with sequential-free box discretization. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'19)*. 3052–3058.

[39] Shangbang Long, Siyang Qin, Dmitry Panteleev, A. Bissacco, Yasuhisa Fujii, and Michalis Raptis. 2022. Towards end-to-end unified scene text detection and layout analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'22)*, 1039–1049.

[40] XiaoPeng Lu, Zhenhua Fan, Yansen Wang, Jean Oh, and Carolyn Penstein Rosé. 2021. Localize, group, and select: Boosting text-VQA by scene text modeling. In *Proceedings of the International Conference on Computer Vision (ICCV'21) Workshop*. 2631–2639.

[41] Anand Mishra, Karteek Alahari, and C. V. Jawahar. 2013. Image retrieval using textual cues. In *Proceedings of the International Conference on Computer Vision (ICCV'13)*. 3040–3047.

[42] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 6 (2017), 1137–1149.

[43] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. 2020. Textcaps: A dataset for image captioning with reading comprehension. In *Proceedings of the European Conference on Computer Vision (ECCV'20)*. 742–758.

[44] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19)*. 8317–8326.

[45] Peipei Song, Dan Guo, Xun Yang, Shengeng Tang, Erkun Yang, and Meng Wang. 2023. Emotion-prior awareness network for emotional video captioning. *Proceedings of the ACM International Conference on Multimedia (ACM MM'23)*. 589–600.

[46] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. 2022. Maxvit: Multi-axis vision transformer. In *Proceedings of the European Conference on Computer Vision (ECCV'22)*. 459–479.

[47] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. 2016. Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv:1601.07140. Retrieved from https://arxiv.org/abs/1601.07140

[48] Jun Wang, Mingfei Gao, Yuqian Hu, Ramprasaath R. Selvaraju, Chetan Ramaiah, Ran Xu, Joseph JáJá, and Larry Davis. 2022. TAG: Boosting text-VQA via text-aware visual question-answer generation. In *Proceedings of the British Machine Vision Conference (BMVC'22)*.

[49] Jing Wang, Jinhui Tang, and Jiebo Luo. 2020. Multimodal attention with image text spatial relationship for ocr-based image captioning. In *Proceedings of the ACM International Conference on Multimedia (ACM MM'20)*. 4337–4345.

[50] Yanan Wang, Michihiro Yasunaga, Hongyu Ren, Shinya Wada, and Jure Leskovec. 2022. Vqa-gnn: Reasoning with multimodal semantic graph for visual question answering. arXiv:2205.11501. Retrieved from https://arxiv.org/abs/2205.11501

[51] Junran Wu, Xu hui Chen, Ke Xu, and Shangzhe Li. 2022. Structural entropy guided graph hierarchical pooling. In *Proceedings of the International Conference on Machine Learning (ICML'22)*. 24017–24030.

[52] Michael Yang, Aditya Anantharaman, Zachary Kitowski, and Derik Clive Robert. 2021. Graph relation transformer: Incorporating pairwise object features into the transformer architecture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'21) Workshop*.

[53] Xun Yang, Jianfeng Dong, Yixin Cao, Xun Wang, Meng Wang, and Tat-Seng Chua. 2020. Tree-augmented cross-modal encoding for complex-query video retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'20)*. 1339–1348.

[54] Xun Yang, Fuli Feng, Wei Ji, Meng Wang, and Tat-Seng Chua. 2021. Deconfounded video moment retrieval with causal intervention. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'21)*. 1–10.

[55] Xun Yang, Shanshan Wang, Jian Dong, Jianfeng Dong, Meng Wang, and Tat-Seng Chua. 2022. Video moment retrieval with cross-modal neural architecture search. *IEEE Trans. Image Process.* 31 (2022), 1204–1216.

[56] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. 2021. TAP: Text-aware pre-training for text-VQA and text-caption. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 8751–8761.

[57] Gangyan Zeng, Yuan Zhang, Yu Zhou, and Xiaomeng Yang. 2021. Beyond OCR+ VQA: Involving OCR into the flow for robust and accurate TextVQA. In *Proceedings of the ACM International Conference on Multimedia (ACM MM'21)*. 376–385.

[58] Liang Zhang, Xudong Wang, Hongsheng Li, Guangming Zhu, Peiyi Shen, P. Li, Xiaoyuan Lu, Syed Afaq Ali Shah, and Bennamoun. 2020. Structure-feature based graph self-adaptive pooling. In *Proceedings of the International World Wide Web Conferences (WWW'20)*.

[59] Wenqiao Zhang, Haochen Shi, Jiannan Guo, Shengyu Zhang, Qingpeng Cai, Juncheng Li, Sihui Luo, and Yueting Zhuang. 2022. MAGIC: Multimodal relAtional graph adversarIal inferenCe for diverse and unpaired text-based image captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'22)*. Vol. 36. 3335–3343.

[60] Xuanyu Zhang and Qing Yang. 2021. Position-augmented transformers with entity-aligned mesh for TextVQA. In *Proceedings of the ACM International Conference on Multimedia (ACM MM'21)*. 2519–2528.

[61] Qi Zheng, Jianfeng Dong, Xiaoye Qu, Xun Yang, Yabing Wang, Pan Zhou, Baolong Liu, and Xun Wang. 2023. Progressive localization networks for language-based moment localization. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 2 (2023), 1–21.

[62] Sheng Zhou, Dan Guo, Jia Li, Xun Yang, and Meng Wang. 2023. Exploring sparse spatial relation in graph inference for text-based VQA. *IEEE Trans. Image Process.* 32 (2023), 5060–5074.

[63] Qi Zhu, Chenyu Gao, P. Wang, and Qi Wu. 2021. Simple is not easy: A simple strong baseline for TextVQA and TextCaps. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'21)*. 3608–3615.

[64] Yongxin Zhu, Zhen Liu, Yukang Liang, Xin Li, Hao Liu, Changcun Bao, and Linli Xu. 2023. Locate then generate: Bridging vision and language with bounding box for scene-text VQA. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'23)*, 11479–11487.

Received 23 July 2023; revised 13 October 2023; accepted 21 November 2023