DisAVR: Disentangled Adaptive Visual Reasoning Network for Diagram Question Answering

Yaxian Wang, Bifan Wei, Jun Liu, *Senior Member, IEEE*, Lingling Zhang, Jiaxin Wang, and Qianying Wang

***Abstract*— Diagram Question Answering (DQA) aims to correctly answer questions about given diagrams, which demands an interplay of good diagram understanding and effective reasoning. However, the same appearance of objects in diagrams can express different semantics. This kind of visual semantic ambiguity problem makes it challenging to represent diagrams sufficiently for better understanding. Moreover, since there are questions about diagrams from different perspectives, it is also crucial to perform flexible and adaptive reasoning on content-rich diagrams. In this paper, we propose a Disentangled Adaptive Visual Reasoning Network for DQA, named DisAVR, to jointly optimize the dual-process of representation and reasoning. DisAVR mainly comprises three modules: improved region feature learning, question parsing, and disentangled adaptive reasoning. Specifically, the improved region feature learning module is designed to first learn robust diagram representation by integrating detail-aware patch features and semantically-explicit text features with region features. Subsequently, the question parsing module decomposes the question into three types of question guidance including region, spatial relation and semantic relation guidance to dynamically guide subsequent reasoning. Next, the disentangled adaptive reasoning module decomposes the whole reasoning process by employing three visual reasoning cells to construct a soft fully-connected multi-layer stacked routing space. These three cells in each layer reason over object regions, semantic and spatial relations in the diagram under the corresponding question guidance. Moreover, an adaptive routing mechanism is designed to flexibly explore more optimal reasoning paths for specific diagram-question pairs. Extensive experiments on three DQA datasets demonstrate the superiority of our DisAVR.**

***Index Terms*— Diagram understanding, visual reasoning, diagram question answering.**

I. Introduction

VISUAL Question Answering (VQA) [1] is a typical multimodal task for connecting the fields of computer vision and natural language processing, which aims to correctly answer questions about given images. Different from the general VQA task about natural images, we focus more on unique kinds of diagrams, named Diagram Question Answering (DQA) [2], as a more challenging task. Unlike natural images, diagrams are usually abstract representations of knowledge using simple shapes and color blocks, which widely exist in textbooks and other various educational resources. A diagram usually contains various constituents such as text, illustrative objects in graphics forms, and their relationships, which convey rich knowledge to humans. Research on diagram question answering is of great significance, especially for e-learning.

Diagram understanding and reasoning are the cornerstones of diagram question answering. Despite recent research efforts [2], [3], [4], [5], [6] to solve the DQA task, there is still a lack of two crucial abilities to complete this challenging task: ***(i)* effective representation ability** to help with diagram understanding; ***(ii)* flexible reasoning ability** over diagrams with respect to questions from different perspectives.

First, for diagram understanding, it is essential to learn a robust diagram representation. However, most existing work [4], [5], [6] only considers regional visual information, which exhibits limited diagram representation ability. Specifically, the knowledge abstraction characteristic of diagrams can easily lead to the problem of visual semantic ambiguity, where the same appearance of objects in diagrams can express different semantics. As shown in Fig. 1, the same red circle in the first diagram can denote different objects, such as *Mercury* and *Mars* in the *solar system* topic. Thus, there is a significant gap between the low-level visual representations and high-level semantics for objects in diagrams, resulting in the inability to express explicit semantics with only their regional visual information.

Second, to obtain an accurate answer, it is necessary to perform flexible and adaptive reasoning over content-rich diagrams based on different questions. Concretely, for diagrams, there are questions that focus on the information from different

Manuscript received 13 October 2022; revised 13 July 2023; accepted 14 August 2023. Date of publication 24 August 2023; date of current version 29 August 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFC3303600; in part by the National Natural Science Foundation of China under Grant 62137002, Grant 62293550, Grant 62293553, Grant 62293554, Grant 61937001, Grant 62250066, Grant 62176209, and Grant 62106190; in part by the Innovative Research Group of the National Natural Science Foundation of China under Grant 61721002; in part by the Lenovo-Xi'an Jiaotong University (XJTU) Intelligent Industry Joint Laboratory Project; in part by the China Computer Federation (CCF)-Lenovo Blue Ocean Research Fund; in part by the Foundation of Key National Defense Science and Technology Laboratory under Grant 6142101210201; in part by the Natural Science Basic Research Program of Shaanxi under Grant 2023-JC-YB-593; in part by the Youth Innovation Team of Shaanxi Universities; in part by the Fundamental Research Funds for the Central Universities under Grant xhj032021013-02; and in part by the XJTU Teaching Reform Research Project "Acquisition Learning Based on Knowledge Forest." The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hichem Sahbi. *(Corresponding author: Jun Liu.)*

Yaxian Wang, Bifan Wei, and Jun Liu are with the National Engineering Laboratory for Big Data Analytics, School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China (e-mail: wyx1566@stu.xjtu.edu.cn; weibifan@xjtu.edu.cn; liukeen@xjtu.edu.cn).

Lingling Zhang and Jiaxin Wang are with the Key Laboratory of Intelligent Networks and Network Security, Xi'an Jiaotong University, Ministry of Education, Xi'an, Shaanxi 710049, China (e-mail: zhanglling@xjtu.edu.cn; jiaxinwangg@outlook.com).

Qianying Wang is with Lenovo Research, Beijing 100094, China (e-mail: wangqya@lenovo.com).

Digital Object Identifier 10.1109/TIP.2023.3306910

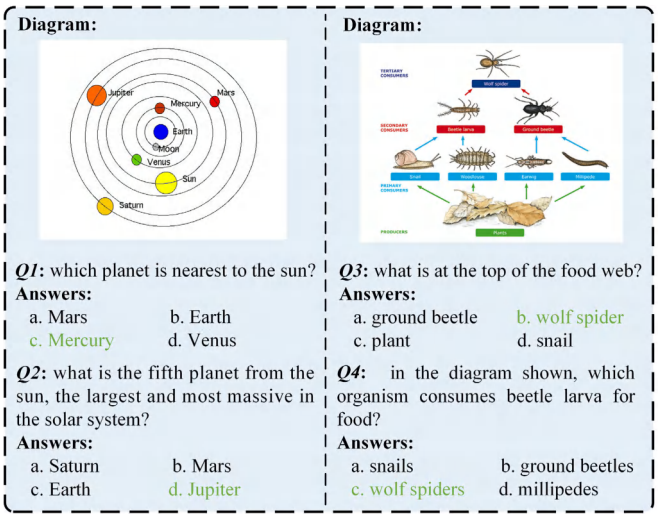


Fig. 1. Examples for diagram question answering. The first diagram demonstrates the visual semantic ambiguity due to the abstract expression of knowledge. *Q1*-*Q4* demonstrate the diagram questions focusing on the information from different perspectives.

perspectives, such as the isolated object regions, and their spatial and semantic relations. For example, *Q1* and *Q3* in Fig. 1 need to reason about the spatial relations between object regions. *Q4* requires modeling the semantic relations between *larva* and *wolf spiders*. A more complex question about a diagram may even attend to multiple perspectives, such as *Q2*. However, recent advances in DQA task are still primarily driven by representation improvements rather than reasoning, resulting in far from satisfactory reasoning ability. Most previous works [2], [3], [4], [5] usually built the question-independent diagram representation and then simply attended to question-related content based on the full question for answer prediction. These methods can perform well on some simple look-up questions directly related to the diagram, but not enough, especially for questions that require further reasoning. Moreover, these existing methods do not utilize the question information from different perspectives, resulting in the inability to perform flexible and adaptive reasoning over the diagrams according to specific questions.

To address these limitations, we propose a novel **Dis**entangled **A**daptive **V**isual **R**easoning Network for DQA, named DisAVR, which jointly optimizes the dual-process of representation and reasoning. DisAVR mainly comprises three modules: improved region feature learning, question parsing and disentangled adaptive reasoning. Specifically, the improved region feature learning module is first proposed to integrate the detail-aware patch features and semantically-explicit text features to help enhance the region features. This is achieved by designing a Dual-stream Transformer (DsTrans) that enables fusing different features using the created special interaction tokens as bridges. Subsequently, by distinguishing the question information from the different perspectives, the question parsing module decomposes a question into three types of guidance including region, semantic relation, and spatial relation guidance, which dynamically guide reasoning over the diagram. Next, the disentangled adaptive reasoning module disentangles the complex reasoning process into explicit subprocesses by employing three fundamental reasoning cells (i.e., LocateRegCell, RelateSmRelCell and RelateSpRelCell). These reasoning cells perform adaptive reasoning under the corresponding guidance generated by the question parsing module. Particularly, a soft fully-connected multi-layer stacked routing space is constructed to allow iterative reasoning, wherein the three reasoning cells are deployed in parallel. Additionally, an adaptive routing mechanism is designed to flexibly explore more optimal reasoning paths for different diagram-question pairs. The disentangled adaptive design can also give us insights into the complex reasoning process.

The main contributions of our paper can be summarized as follows:

- We propose a novel disentangled adaptive visual reasoning network that jointly optimizes the dual-process of representation and reasoning to solve the DQA task.
- We devise an improved region feature learning method to incorporate patch features and text features with region features using a dual-stream transformer, which helps obtain detail-aware and semantic-enhanced region features.
- We design three visual reasoning cells to disentangle the complex reasoning process by performing explicit sub-processes and design an adaptive routing mechanism to help flexibly explore more optimal reasoning patterns for different diagram-question pairs.
- Extensive experiments on three DQA datasets including AI2D, FOODWEBS and CSDQA demonstrate the superior performance of DisAVR compared with the state-of-the-art methods.

The rest of the paper is organized as follows. Section II briefly introduces existing related work. Section III describes the details of DisAVR. Section IV presents the experimental details and analysis of the results for our proposed model. Section V concludes our work and puts forward future work.

II. Related Work

The research most relevant to our work is diagram understanding and visual reasoning.

A. Diagram Understanding

In recent years, diverse DQA tasks have been proposed for evaluating different types of skills for answering diagram questions. Diagram understanding is the important foundation for some downstream tasks, such as diagram question answering [2], [7], and image captioning [8], [9]. According to the different expression styles of diagrams, we divide them into three categories as shown in Fig. 2.

1) Diagrams of Scientific Plots: This kind of diagrams is about line plots, dot-line plots, vertical and horizontal bar graphs, pie charts, etc. Chart question answering [10], [11], [12], [13], [14] has attracted increasing research attention and several datasets and models have been proposed recently. Earlier tasks such as FigureQA [10], DVQA [11] generate the plots with synthetically-generated data and generate the

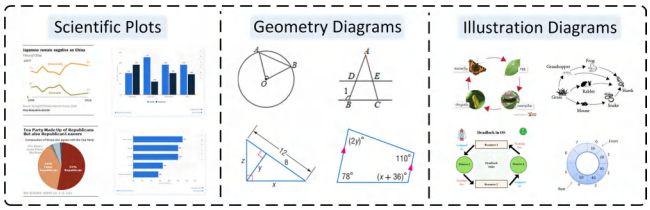


Fig. 2. Examples of different kinds of diagrams including scientific plots, geometry diagrams, and illustration diagrams.

fixed-vocabulary questions using few templates. To reduce the gap between synthetic plots and real-world plots, Methani et al. [12] proposed PlotQA with plots from the real-world and open-vocabulary questions. However, all of these datasets do not involve complex visual reasoning questions. Later, Masry et al. [14] proposed a large-scale Chart QA dataset involving visual and logical reasoning questions about real-world charts. Existing methods [12], [15] usually convert the charts to tables, and exploit the table QA methods to solve the chart QA task. A new pipeline in [14] utilizes transformer-based QA models to combine visual features and automatically extract data from charts.

2) Diagrams of Geometry: This kind of diagrams is about geometric shapes such as the point, line, circle and angle. Several related datasets for geometric problem solving, such as GEOS [16], GeoQA [17], Geometry3K [18], have been released in recent years. The GEOS dataset [16] contains diagrams with a simple layout and only 186 related questions. The geometry problem's text and diagram for GEOS are parsed into the manually designed logic forms with poor generalization ability. Therefore, to improve the generalization and interpretability for geometric problem solving, Chen et al. [17] proposed GeoQA dataset with a large scale, which contains 4,998 geometric questions with additional annotated programs. They also introduced the first deep learning-based approach for geometry problem solving to parse multimodal information and generate interpretable programs for question answering. Later, Lu et al. [18] constructed a new large scale dataset Geometry3K, consisting of 3,002 geometry problems with denser annotation in formal language. Moreover, they proposed an interpretable geometry problem solver Inter-GPS to parse the problem text and diagram into formal language and perform symbolic reasoning to infer the answer.

3) Diagrams of Illustration: This kind of diagrams usually exists in the education domain such as geography, biology, physics, and computer science. The related datasets including AI2D [2], FOODWEBS [19], CSDQA [7] have been proposed to facilitate further research. AI2 Diagrams (AI2D) [2] are about grade school science to depict complex phenomena such as the lives of animals, water cycles and phase changes. FOODWEBS [19] focuses more on question answering about food webs. Based on the two datasets, previous work has devoted much effort to diagram understanding, further solving the diagram question answering. Kembhavi et al. [2] proposed Diagram Parse Graphs (DPGs) to model the structure of diagrams using a deep sequential diagram parser network, and a DQA-NET is introduced to attend to the relations in a DPG for DQA task. However, this work parses diagrams in a long pipeline, leading to accumulated errors. To address this problem, Kim et al. [3] proposed an UDPnet to parse the diagram into a graph structure by jointly optimizing object detection and relation matching in an end-to-end manner. Moreover, the relational knowledge sentences can be generated for subsequent question answering. Different from the above two-stage cascade methods, Yuan et al. [6] proposed a hierarchical multi-task learning model based on a multimodal transformer framework to simultaneously optimize diagram parsing and diagram question answering. Wang et al. [20] proposed a spatial-semantic collaborative graph network to help exploit the spatial and semantic relationships between visual objects and OCR tokens to collaboratively enhance diagram semantic understanding. CSDQA [7] is a newly proposed DQA dataset, which focuses on question answering about computer science domains such as array list, and stack. The topology generation method in [7] is proposed for enhancing diagram understanding to promote question answering.

Our work focuses on diagrams of illustration and conducts experiments on these three existing datasets. Previous works for DQA make more attempts on diagram understanding and pay less attention to reasoning. In this paper, DisAVR is designed based on a dual-process of representation and reasoning, which aims to enhance diagram understanding ability and improve reasoning ability meanwhile.

B. Visual Reasoning

Visual reasoning is extensively exploited to tackle visual-and-language tasks, such as VQA and Referring Expression Comprehension (REF) [21], [22]. The existing work can be roughly divided into two categories: graph-based and neural-symbolic-based.

1) Graph-Based: Graph network is a powerful method to model object interactions and has been widely applied to visual reasoning. Teney et al. proposed [23] a deep neural network for VQA that combines graph-structured representations of scenes and questions, which is the first work to adopt a graph network for VQA task. Norcliffe-Brown et al. [24] proposed to learn a graph representation of an input image by capturing object interactions with the most relevant neighbors using spatial graph convolutions. Hu et al. [25] proposed a language-conditioned graph network to build contextualized representations for objects, further supporting relational reasoning. Le et al. [26] proposed to build the dynamic visual graphs to perform multi-step visual-linguistic binding and facilitate knowledge combination and refinements for reasoning the final answer. Jing et al. [27] focused more on reasoning consistency and proposed to reason through the dialog-like form. In each reasoning process, the image is represented as a visual graph and performs an iterative graph convolution to gradually obtain its visual representation.

2) Neural-Symbolic-Based: Graph-based methods for visual reasoning cannot model the reasoning process explicitly due to the black-box architectures. However, the neural-symbolic-based methods can construct an explicit reasoning process to address compositional visual reasoning mainly for CLEVER dataset [28]. Andreas et al. [29] first

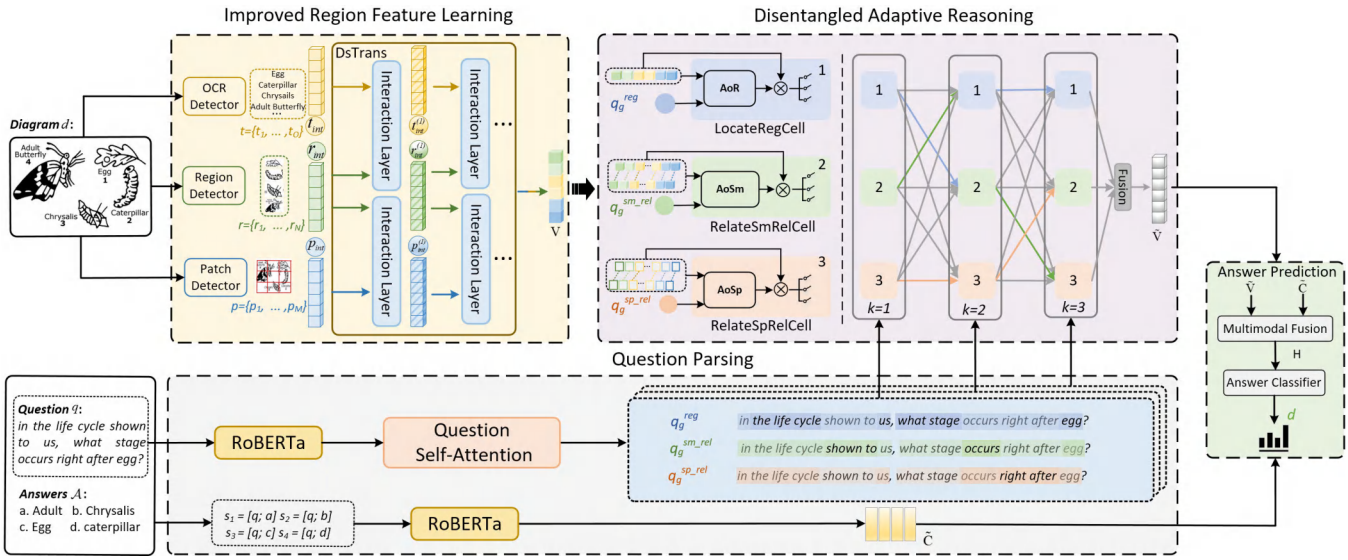


Fig. 3. Overall framework of Disentangled Adaptive Visual Reasoning Network (DisAVR). It mainly consists of three modules: an improved region feature learning module, a question parsing module, a disentangled adaptive reasoning module.

proposed Neural Module Network (NMN), which generates a program for the question, and then executes the program on the image by assembling specialize modules. Subsequently, some methods extend NMN by designing better layout policy [30], [31] or modifying the reasoning patterns [32], [33]. Hu et al. [30] proposed N2NMN to make the networks end-to-end learnable with reinforcement learning approach by making discrete choices of module layout. Different from N2NMN that makes discrete choices of module layout, Hu et al. [31] proposed to make soft layout selection with a differentiable stack structure, and without relying on strong expert layout supervision. Instead of reasoning on the image, Yi et al. [32] proposed a more explicit symbolic reasoning method NS-VQA to execute the program on a structural scene graph of the image to obtain an answer. Vedantam et al. [33] proposed a probabilistic neural-symbolic model to treat functional programs as a stochastic latent variable, which can provide more understandable and legible program explanations for novel questions.

Our work aims to reason over complex diagram-question pairs. On the one hand, graph networks usually entangle visual reasoning about objects and relationships, resulting in the inability to intervene in the intermediate reasoning process. On the other hand, it is also challenging to generate a rule-based program layout for the specific question. Different from but inspired by the above methods, we disentangle the complex reasoning process and distinguish different information of the question to guide reasoning over the diagram, which can promote flexible and adaptive reasoning.

III. Approach

In this section, we start with the problem definition on the DQA task. Our task is to select a correct answer $\hat{a}$ from the candidate answer set $\mathcal{A}$ according to a diagram d and a question q about the diagram, which is defined as follows:

$$\hat{a} = \arg\max_{a \in \mathcal{A}} p(a|d, q; \theta), \tag{1}$$

where $\mathcal{A} = \{a_1, a_2, \ldots, a_T\}$, and T is the number of the optional answers. The trainable parameter θ should be optimized to obtain the predicted answer $\hat{a}$.

Our overall DisAVR framework is illustrated in Fig. 3, which mainly comprises three modules:

- Improved region feature learning module. It integrates detail-aware patch features and semantically-explicit text features to enhance the region features. Specifically, a dual-stream transformer is designed with special interaction tokens for each feature branch as the bridge, which can help different-branch features interact with each other in each stream.
- Question parsing module. It mainly parses and represents the question. The question is decomposed into three types of guidance including region, semantic relation and spatial relation guidance for subsequent reasoning.
- Disentangled adaptive reasoning module. It disentangles the reasoning process by employing three cells including LocateRegCell, RelateSmCell and RelateSpCell. A soft fully-connected and multi-layer stacked routing space is constructed with an adaptive routing mechanism to explore more optimal reasoning patterns.

For convenience, the explanations for some important notations are summarized in Table I.

A. Improved Region Feature Learning

The object detector YOLOv3 [34] is first used to detect the regions of interest (RoI) bottom-up in the diagram. Then, a collection of initial region features $r = \{r_1, r_2, \ldots, r_N\}$ is extracted for N objects, where each region feature is obtained by combining the visual feature and positional feature. Specifically, a 1024-dimensional visual feature v_i is obtained by a pretrained ResNet-101 [35] backbone, and a 4-dimensional positional feature f_i contains 4 bounding box coordinates. The region feature of the i-th object is finally computed as $r_i = \mathrm{LN}(\mathbf{W}_v v_i) + \mathrm{LN}(\mathbf{W}_p f_i)$, where $\mathbf{W}_v$ and

TABLE I
IMPORTANT NOTATIONS AND EXPLANATIONS

Notations	Explanations
r, p, t	a collection of the initial region features, patch features and text features.
$p_{int}^{(l-1)}, r_{int}^{(l-1)}$	the features of patch and region interaction tokens as the input of l-th IInt layer.
$\widetilde{p}_{int}^{(l)}, \widetilde{r}_{int}^{(l)}$	the branch-specific context-aware features of patch and region interaction tokens as the output of l-th IInt layer.
$p_{int}^{'(l)}, t_{int}^{'(l)}$	the interaction token features, obtained by feeding $\widetilde{r}_{int}^{(l)}$ into the linear layer.
$p^{'(l)}, t^{'(l)}$	the concatenation of region interaction token feature and patch features sequence or text features sequence.
$\widetilde{r}_{int,p}^{(l)}, \widetilde{r}_{int,t}^{(l)}$	the updated region features after l-th IInt layer and CInt layer in RP-Stream and RT-Stream.
$r_{int}^{(l)}$	the comprehensive feature of the region interaction token by fusing the outputs from RP-Stream and RT-Stream.
$\{r_1^{(L)}, ..., r_N^{(L)}\}$	the final comprehensive region features after L interaction layers of DsTrans, also denoted as $V = [v_1, v_2, ..., v_N]$.
$\{a_s^{reg,sm\_rel,sp\_rel}\}_{s=1}^{S}$	region attention, semantic relation attention, and spatial relation attention.
$q_g^{reg}, q_g^{sm\_rel}, q_g^{sp\_rel}$	three types of question guidance including region, semantic relation, and spatial relation guidance.
$e_{ij}^{sm\_rel}, e_{ij}^{sp\_rel}$	the semantic and spatial relationship between the region i and j.
$A^r, A^{sm\_rel}, A^{sp\_rel}$	the attention over regions, their semantic and spatial relationship under the corresponding question guidance.
$o_i^r, o_i^{sm}, o_i^{sp}$	the updated region-aware, semantic-aware, and spatial-aware feature of the region i.
G_x^k	the routing gate for the x-th cell in k-th layer.
$O^{x,k}$	the region features output from the x-th cell in the k-th layer.
$V^{y,k+1}$	the region features output from the y-th cell in the k+1-th layer after aggregation operation.
$\widetilde{V}$	a question-relevant context-aware visual representation after K layers iterative reasoning for final answer prediction.

$\mathbf{W}_p$ denote transformation matrices, and LN denotes the layer normalization.

However, the obtained object region features can not deal with the problem of visual semantic ambiguity caused by the knowledge abstract characteristic of the diagrams, and also ignore the detailed information within the object regions. Therefore, we design a novel Dual-stream Transformer (DsTrans) to integrate detail-aware patch features and semantically-explicit text features with region features, where information exchange is achieved through specially initialized interaction tokens.

To be specific, we regard the region features as a primary branch for generating two streams using patch features and text features as complementary branches: ***(i)* RP-Stream** integrates patch features with region features, wherein detail-aware patch features cover a large object in a more fragmented form to make up details for regions; ***(ii)* RT-Stream** integrates text features with region features, wherein semantically-explicit text features can further enhance the semantic representations of regions. In each stream, there are L interaction layers, wherein each interaction layer includes the Intra-branch Interaction (IInt) layer and the Cross-branch Interaction (CInt) layer. The IInt layer is designed to encode the information of each branch respectively and obtain the context-aware features by self-interaction. The CInt layer is designed to fuse different-branch features with the help of special interaction tokens. Especially, we define three specially initialized interaction tokens p_{int}, r_{int}, and t_{int} for each branch, whose representations serve as the bridges, through which the region features interact with the other two types of features and exchange information with each other.

1) RP-Stream: A collection of initial patch features $p = \{p_1, p_2, \ldots, p_M\}$ are extracted for M patches using Vision Transformer (ViT) [36], where each patch is represented as a 1024-dimensional feature.

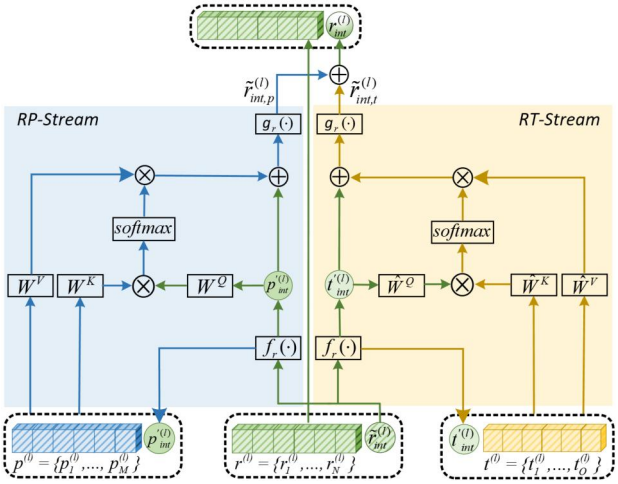


Fig. 4. The details in l-th Cross-branch Interaction layer (CInt) for DsTrans. Note that the input for the CInt layer is the output of the IInt layer, here we omit this step.

In the l-th IInt layer, the patch feature sequence $\{p_{int}^{(l-1)}, p_1^{(l-1)}, \ldots, p_M^{(l-1)}\}$ and the region feature sequence $\{r_{int}^{(l-1)}, r_1^{(l-1)}, \ldots, r_N^{(l-1)}\}$ are fed into stacked transformer layers, which helps obtain the branch-specific context-aware representations $\{\widetilde{p}_{int}^{(l)}, p_1^{(l)}, \ldots, p_M^{(l)}\}$ and $\{\widetilde{r}_{int}^{(l)}, r_1^{(l)}, \ldots, r_N^{(l)}\}$.

In the l-th CInt layer, as illustrated in Fig. 4, with the region features as the primary branch, we first concatenate its interaction token and patch features sequence of the complementary branch as follows:

$$p^{'(l)} = [f_r(\widetilde{r}_{int}^{(l)}); p^{(l)}], \tag{2}$$

where $f_r(\cdot)$ is a linear projection function, [;] denotes concatenation operation, and $p^{(l)} = \{p_1^{(l)}, \ldots, p_M^{(l)}\}$.

Then, the multi-head cross attention (MHCA) is performed with the interaction token as the query to attend to the information from the patch features branch as follows:

$$\begin{aligned} \{Q, K, V\} &= \{p_{int}^{'(l)}, p^{'(l)}, p^{'(l)}\}, \\ \text{MHCA}(Q, K, V) &= \text{Concat}(head_1, \ldots, head_h)\mathbf{W}^O, \\ head_i &= \text{CA}(Q\mathbf{W}_i^Q, K\mathbf{W}_i^K, V\mathbf{W}_i^V), \\ \text{CA}(Q_i, K_i, V_i) &= \text{softmax}(\frac{Q_i K_i^\top}{\sqrt{d/h}})V_i, \end{aligned} \tag{3}$$

where $p_{int}^{'(l)} = f_r(\widetilde{r}_{int}^{(l)})$, $\mathbf{W}^O \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d\times(d/h)}$ $(i \in \{1, 2, \ldots, h\})$ are learnable linear projection matrices, d and h are the embedding dimension and number of heads, $\top$ is a transpose operator, and CA denotes cross attention.

We summarize the above process as follows:

$$\widetilde{r}_{int,p}^{(l)} = f_r(\widetilde{r}_{int}^{(l)}) + \text{MHCA}(f_r(\widetilde{r}_{int}^{(l)}), p^{'(l)}, p^{'(l)}). \tag{4}$$

2) RT-Stream: The external Optical Character Recognition (OCR) detector[1] is first used to detect the OCR tokens in the diagram. Then, a collection of initial text features $t = \{t_1, t_2, \ldots, t_O\}$ are extracted for O OCR tokens, where the text feature for each OCR token is represented by combining the 1024-dimensional feature obtained by a pretrained language model RoBERTa-large [37] and 4-dimensional positional feature. They are combined in the same way as the region feature.

Similar to RP-Stream, in the l-th IInt layer, we can obtain the context-aware region features $\{r_{int}^{(l)}, r_1^{(l)}, \ldots, r_N^{(l-1)}\}$ and text features $\{t_{int}^{(l)}, t_1^{(l)}, \ldots, t_O^{(l)}\}$ by self interaction. In the l-th CInt layer, with the region features as the primary branch, we fuse text features into the region interaction token to obtain updated feature $\widetilde{r}_{int,t}^{(l)}$ as follows:

$$\begin{aligned} t^{'(l)} &= [f_r(\widetilde{r}_{int}^{(l)}); t^{(l)}], \\ \widetilde{r}_{int,t}^{(l)} &= f_r(\widetilde{r}_{int}^{(l)}) + \text{MHCA}(f_r(\widetilde{r}_{int}^{(l)}), t^{'(l)}, t^{'(l)}). \end{aligned} \tag{5}$$

After the IInt layer and CInt layer, we obtain the comprehensive feature $r_{int}^{(l)}$ of the region interaction token by fusing the outputs of two streams as follows:

$$r_{int}^{(l)} = g_r(\widetilde{r}_{int,p}^{(l)}) + g_r(\widetilde{r}_{int,t}^{(l)}), \tag{6}$$

where $g_r(\cdot)$ denotes a linear projection function.

The region interaction token $r_{int}^{(l)}$ bridges the interaction between the region features and patch features, text features. By concatenating the updated region interaction token $r_{int}^{(l)}$ and the region feature sequence $\{r_1^{(l)}, \ldots, r_N^{(l)}\}$, we can obtain post-fused region feature sequence $\{r_{int}^{(l)}, r^{(l)}\} = \{r_{int}^{(l)}, r_1^{(l)}, r_2^{(l)}, \ldots, r_N^{(l)}\}$ from the whole interaction layer. Similarly, we can obtain the patch features $\{p_{int}^{(l)}, p^{(l)}\}$ and the text features $\{t_{int}^{(l)}, t^{(l)}\}$. Then, these features are fed into the next IInt layer. At the next IInt layer, the interaction token fused with the information of other branches interacts with all its own tokens again, which enables all tokens of its own branch to receive the information from other branches and enrich the representation of each token.

After L interaction layers of DsTrans, we can obtain the final comprehensive region features $\{r_1^{(L)}, \ldots, r_N^{(L)}\}$ of N objects for subsequent reasoning. For convenience, we denote the object region features as $V = [v_1, v_2, \ldots, v_N] \in \mathbb{R}^{N\times d}$ below, where $v_i \in \mathbb{R}^d$ is the feature of the i-th object region.

B. Question Parsing

For diagrams, there are questions focusing on the information from different perspectives, such as the isolated object region, and their spatial and semantic relations. A complex question about a diagram may involve information from multiple perspectives. Inspired by the idea that parsing the whole linguistic structure into sub-structures [29], [30] helps fully utilize the rich information in the language, we adopt a question self-attention mechanism to decompose the question by distinguishing the different information attention of the question. In this way, different question representations are dynamically generated to guide the subsequent disentangled reasoning process.

Specifically, three types of question guidance are generated, including region guidance, semantic relation guidance, and spatial relation guidance. We first use the RoBERTa [37] language model to encode the question q with S words, by which we can get the hidden state representations $h = \{h_s\}_{s=1}^S$. Then, we adopt three individual fully-connected (FC) layers followed by a softmax layer to obtain three attention weights including region attention $\{a_s^{reg}\}_{s=1}^S$, semantic relation attention $\{a_s^{sm\_rel}\}_{s=1}^S$ and spatial relation attention $\{a_s^{sp\_rel}\}_{s=1}^S$. Due to the same way to obtain these attention weights, for simplicity, we take the region guidance as an example:

$$\begin{aligned} a_s^{reg} &= \frac{\exp(\mathbf{W}_{reg}^\top h_s)}{\sum_{i=1}^S \exp(\mathbf{W}_{reg}^\top h_i)}, \\ q_g^{reg} &= \sum_{s=1}^S a_s^{reg} \cdot h_s, \end{aligned} \tag{7}$$

where $\mathbf{W}_{reg}$ denotes the parameters of the FC layer. Similarly, we can get three types of question guidance: region guidance q_g^{reg}, semantic relation guidance $q_g^{sm\_rel}$, and spatial relation guidance $q_g^{sp\_rel}$.

In addition, the question q and the answer options a_t $(t = 1, \ldots, T)$ are concatenated to form T statements. Then, we use the RoBERTa language model to obtain the pooled representation $\widetilde{C} = [\widetilde{c}_1, \widetilde{c}_2, \ldots, \widetilde{c}_T] \in \mathbb{R}^{T\times d_q}$, where each statement $\widetilde{c}_t \in \mathbb{R}^{d_q}$ is represented by the language model with the sequence $s_t = seq([q, a_t])$ as the input.

C. Disentangled Adaptive Reasoning

In the disentangled adaptive reasoning module, we elaborately design three visual reasoning cells to disentangle the complex reasoning process, which attend to different content in the diagram respectively under the corresponding question guidance generated by the question parsing module. These cells can be dynamically and flexibly deployed into

[1]https://ocr.space/

a reasoning network to perform adaptive reasoning over the content-rich and semantically-complex diagram. Specifically, we construct a soft fully-connected and multi-layer stacked routing space deployed with three cells in parallel at each layer to allow iterative reasoning. Then, an adaptive routing mechanism is designed to control the information propagation for these cells between adjacent layers by dynamically adjusting routing probability. Finally, after the layer iterations, we can obtain the refined question-related diagram representation for answer prediction. Additionally, this module can also give us insights into the complex reasoning process.

1) Visual Reasoning Cells: Formally, the calculation for the three visual reasoning cells can be summarized as:

$$O^{x,k} = \mathcal{F}^{x,k}(V^{x,k}, q_g^{x,k}), \tag{8}$$

where $V^{x,k} \in \mathbb{R}^{N\times d}$ and $O^{x,k} \in \mathbb{R}^{N\times d}$ denote the input and output of the x-th cell in the k-th layer, and $\mathcal{F}^{x,k}$ denotes the visual reasoning function of the x-th cell in the k-th layer under the corresponding question guidance $q_g^{x,k}$ (here $x \in \{1, 2, 3\}$). For simplicity and readability, we omit the layer index k and use $\{reg, sm\_rel, sp\_rel\}$ to denote the index of different cells below.

Here, we first introduce the attention mechanism used in these visual reasoning cells:

$$A(X, q) = \text{softmax}(\mathbf{W}_1^\top \tanh(\mathbf{W}_2 X + \mathbf{W}_3 q)), \tag{9}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{W}_3$ are trainable parameters, and X and q are values and queries of attention. We follow the commonly used additive attention [38]. On the one hand, additive attention has a smaller computational cost compared to some complex attention mechanisms, such as multiplicative attention and self-attention, making it more efficient. On the other hand, additive attention has good interpretability, which uses a simple weighted sum formula to calculate attention weights, making it easier to understand which inputs contribute more to the output. The following AoR, AoSm and AoSp are accomplished via Eq. (9).

LocateRegCell: The cell is mainly designed for locating the relevant regions from the region set V under the region guidance q_g^{reg} as follows:

$$A^r = \text{AoR}(V, q_g^{reg}), \tag{10}$$

where the AoR$(\cdot)$ denotes the attention over regions. With the region attention weight A^r at hand, we update the representation of region i by calculating:

$$o_i^r = A_i^r v_i. \tag{11}$$

where A_i^r denotes the attention over region i, and v_i denotes the feature of region i.

RelateSmRelCell: There are semantic relationships between object regions, which denote their semantic interaction (e.g., object region i is larger than object region j, or object region i eats object region j). To represent the semantic relationship, we take the interaction between region i and region j as an example, which is calculated as follows:

$$e_{ij}^{sm\_rel} = f^{sm\_rel}([v_i; v_j]), \tag{12}$$

where f^{sm_rel} is a multi-layer perceptron (MLP), the input of MLP is the concatenation of the features of region i and region j, and [;] denotes concatenation operation.

The attention over the semantic relationship (AoSm) under the question guidance $q_g^{sm\_rel}$ is calculated as follows:

$$A^{sm\_rel} = \text{AoSm}(E^{sm\_rel}, q_g^{sm\_rel}), \tag{13}$$

where E^{sm_rel} denotes the pairwise semantic interaction of all regions and $e_{ij}^{sm\_rel} \in E^{sm\_rel}$.

Then, the information propagation for region i is a weighted sum of the 1-hop semantic relation neighbors, and the updated semantic-aware representation of region i can be obtained by calculation as follows:

$$o_i^{sm} = \sum_{j=1}^{N} A_{ij}^{sm\_rel} e_{ij}^{sm\_rel}, \tag{14}$$

where j denotes the region index of the neighborhood for region i, and $A_{ij}^{sm\_rel}$ denotes the edge attention weight for the semantic relationship $e_{ij}^{sm\_rel}$ of region i and region j.

RelateSpRelCell: The spatial relationships reflect the relative location between object regions (e.g. object region i is on the right of object region j). The spatial relationship between region i and j is represented as follows:

$$e_{ij}^{sp\_rel} = [\frac{x_j^{tl} - x_i^c}{w_i}; \frac{y_j^{tl} - y_i^c}{h_i}; \frac{x_j^{br} - x_i^c}{w_i}; \frac{y_j^{br} - y_i^c}{h_i}; \frac{w_j \cdot h_j}{w_i \cdot h_i}],$$
$$e_{ij}^{sp\_rel} = f^{sp\_rel}([e_{ij}^{sp\_rel}; v_j]), \tag{15}$$

where $[x_i^c, y_i^c, w_i, h_i]$ is the center coordinate, width, height of region i, and $[x_j^{tl}, y_j^{tl}, x_j^{br}, y_j^{br}, w_j, h_j]$ is the top-left, bottom-right coordinate, width and height of region j, and [;] denotes concatenation operation.

The attention over the spatial relationship (AoSp) under question guidance $q_g^{sp\_rel}$ is calculated as follows:

$$A^{sp\_rel} = \text{AoSp}(E^{sp\_rel}, q_g^{sp\_rel}), \tag{16}$$

where E^{sp_rel} denotes the pairwise spatial interaction of all regions and $e_{ij}^{sp\_rel} \in E^{sp\_rel}$.

After information propagation, the updated spatial-aware representation of region i can be obtained as follows:

$$o_i^{sp} = \sum_{j=1}^{N} A_{ij}^{sp\_rel} e_{ij}^{sp\_rel}. \tag{17}$$

All N object regions are updated in the three cells. According to the numbering of the above three cells, the output of the x-th cell in the k-th layer can be denoted as $O^{x,k} \in \mathbb{R}^{N\times d}$.

2) Adaptive Routing Mechanism: To explore more optimal reasoning patterns for each specific diagram-question pair and realize the adaptive selection for the reasoning paths, we design a soft fully-connected and multi-layer stacked routing space deployed with the above three cells in parallel at each layer, while providing a routing gate for each cell. Accordingly, each cell can select to receive the information from all cells in the previous layer flexibly in the reasoning process. The routing gate for x-th cell in k-th layer is designed as follows:

$$G_x^k = \text{ReLU}(\tanh(\mathbf{W}_r(\text{AP}(V^{x,k})))), \tag{18}$$

where tanh denotes the gating activation function, $\mathbf{W}_r$ denotes the learnable parameters in MLP, AP(·) indicates the average pooling operation to compute the mean value for $V^{x,k}$, as $\frac{1}{N}\sum_{n=1}^{N} v_n^{x,k}$, wherein n is the index of the region, N denotes the number of regions.

For each cell in the first layer, we take $V \in \mathbb{R}^{N\times d}$ output by the improved region feature learning module as the initial input object region features for reasoning. For the y-th cell in the k+1-th layer, we can obtain the input by the aggregation operation as follows:

$$V^{y,k+1} = \sum_{x=1}^{N_c} g_{x,y}^{k} \cdot O^{x,k}, \tag{19}$$

where $N_c = 3$ denotes the number of reasoning cells in layer k, $g_{x,y}^{k} \in G_x^k$ denotes the routing probability from the x-th cell in the k-th layer to the y-th cell in the k+1-th layer, and $O^{x,k}$ refers to the region features output from the x-th cell in the k-th layer.

After K layers of iterative reasoning, we fuse the outputs of the three visual reasoning cells according to routing probability at the last layer to obtain a refined question-relevant context-aware visual representation $\widetilde{V} \in \mathbb{R}^{N\times d}$ for final answer prediction.

D. Answer Prediction

The context-aware diagram representation $\widetilde{V} \in \mathbb{R}^{N\times d}$ is obtained via both the improved region feature learning module and the disentangled adaptive reasoning module. The representation of T statements for the question and answers $\widetilde{C} \in \mathbb{R}^{T\times d_q}$ is obtained in the question parsing module. Then, we employ the same multimodal fusion method as BUTD [39] to fuse the above representations, and further obtain a joint representation H. Finally, we adopt MLP followed by a softmax layer as the answer classifier with H as input to compute the probability $\hat{y}_t$ of each answer candidate.

The objective is to minimize the cross-entropy loss, which is formulated as follows:

$$\mathcal{L} = -\sum_{t=1}^{T} a_t \log \hat{y}_t, \tag{20}$$

where $\hat{y}_t$ is the probability of the predicted answer and a_t is 1 when the t-th answer is the ground truth and 0 otherwise.

IV. Experiments

We first briefly describe the datasets, baselines and the experimental settings of DisAVR in Section IV-A. Then, the experimental results and analyses are presented in Section IV-B. Subsequently, we evaluate the effectiveness of DisAVR through several ablation studies in Section IV-C. In Section IV-D, we further analyze the impact of different hyperparameters. Finally, we present the case studies in Section IV-E.

A. Experimental Settings

1) Datasets: We evaluate the proposed DisAVR on three diagram question answering datasets including AI2D [2], FOODWEBS [19] and CSDQA [7]. The detailed statistics on each split of these datasets are shown in Table II.

TABLE II
Data Statistics for Datasets

Datasets	AI2D		FOODWEBS		CSDQA	
	Diagram	QA pairs	Diagram	QA pairs	Diagram	QA pairs
Train	2,534	7,824	290	3,099	832	2,270
Val	259	906	100	1,062	357	1,008
Test	308	978	100	1,047	-	-
All	3,101	9,708	490	5,208	1,189	3,278

AI2D is comprised of about 5,000 diagrams from grade school science, in which 3,101 diagrams have 9,708 related multiple choice questions. Following the configuration in [4], we divide AI2D into a training set with 2,534 diagrams and 7,824 questions, a validation set with 259 diagrams and 906 questions, and a test set with 308 diagrams and 978 questions.

FOODWEBS contains 490 food web diagrams and 5,208 questions associated with diagrams. We use 290 diagrams with 3,099 questions for training, 100 diagrams with 1,060 questions for validation, and 100 diagrams with 1,047 questions for test following [19].

CSDQA contains 1,189 diagrams and over 3,200 questions from computer science. The dataset is divided into a training set and validation set following [7].

2) Baselines: The DQA methods on AI2D, FOODWEBS and CSDQA can be roughly divided into three categories including Classical VQA, Graph-based, Transformer-based.

- **Classical VQA methods**: VQA [1] uses a pretrained visual encoder VGG-16 [40] to represent the whole diagram and an LSTM [41] to encode the question and answers. Then, it adopts the element-wise product to jointly embed diagram representation and question representation, and further gives an answer. MFB [42], BAN [43], MCAN [44] are three classic multimodal fusion models.
- **Graph-based methods**: DQA-Net (DSDP) [2] encodes the pre-generated diagram parse graph into a set of facts, and then attends to the closest fact with the question. DQA-Net (DGGN) [3] proposes a dynamic graph generation network with dynamic adjacency tensor memory. DQA-Net (Ground Truth) [3] uses the ground truth annotations of diagrams as the input and then combines the question to choose an answer. Graph Traveler [45] proposes to travel the visual graph based on a question-based visual guide, and the final destination nodes are used to produce the answer. DPN-QA [7] designs a topology graph generation method to promote diagram understanding.
- **Transformer-based methods**: ISAAQ [4] uses a pre-trained language model RoBERTa after additional multi-stage pretraining to encode text and uses bottom-up and top-down attention to attend related regions for question answering. HMTL [6] proposes a structural

TABLE III
ACCURACY (%) OF THE DIAGRAM QUESTION ANSWERING ON THE TEST SET OF AI2D AND FOODWEBS DATASETS. THE HIGHEST SCORE IS HIGHLIGHTED IN BOLD AND THE SECOND HIGHEST IS UNDERLINED

Model	Reference	Datasets	
		AI2D	FOODWEBS
VQA [1]	ICCV'2015	32.90	56.50
DQA-NeT (DSDP) [2]	ECCV'2016	38.47	59.30
DQA-NeT (DGGN) [3]	CVPR'2018	39.73	58.22
DQA-NeT (GT) [3]	CVPR'2018	41.55	-
Graph Traveler [45]	CVPR'2019	43.45	-
ISAAQ [4]	EMNLP'2020	73.29	72.49 †
HMTL [6]	MM'2021	49.72	**78.51**
WSTQ [5]	TIP'2022	73.60	73.64 †
SSCGN [20]	TCSVT'2023	<u>74.34</u> †	75.07 †
DisAVR	—	**76.15**	<u>76.98</u>

parsing-integrated hierarchical multitask learning model based on a multimodal transformer for diagram question answering. WSTQ [5] proposes a weakly supervised learning strategy to develop relation detection and text matching task to help learn effective diagram semantics and text understanding. SSCGN [20] proposes a spatial-semantic collaborative graph network to help enhance the diagram and text understanding.

3) Implementation Details: Our model is implemented using the open-source library PyTorch. For the diagrams, we set the maximum number of visual regions as 32 and the maximum number of OCR tokens as 20. The ViT-L/16 pre-trained model is chosen to obtain $M = 14 \times 14 = 196$ patches. For the text encoding, we set the maximum length of the question sequence to 32, and the OCR tokens sequence to 15. All the inputs are truncated or padded to the same length. The number of answer options T is 2 for true-or-false questions in CSDQA dataset and 4 for multiple-choice questions in all three datasets. The number of routing layers K is set to 3 experimentally in the best report. DisAVR[2] is trained with a batch size of 4 during 12 epochs by AdamW optimizer [46] with the parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$). We select the initial learning rate as 3.5e-6. For optimization, a linearly decaying learning rate and warm-up method is used for learning rate update. We set the number of warmup steps as 0, which means that the learning rate decays linearly from the initial learning rate to zero in the total training steps. We run our code on one NVIDIA Tesla V100 card.

B. Performance Comparison

In this section, we demonstrate the effectiveness of our proposed DisAVR by comparing it with several previous SOTA methods on three datasets (i.e., AI2D, FOODWEBS and CSDQA).

1) Results on AI2D and FOODWEBS: Table III shows the accuracy of the test set of AI2D and FOODWEBS datasets, in which the highest score is highlighted in bold and the second highest is underlined. Note that most of the recent methods just report the results on the test set. Therefore, we compare DisAVR with these baselines only on the test set. The results denoted by '†' are obtained by running their released codes. It is obvious that our proposed DisAVR achieves superior performance on the two datasets. Specifically, DisAVR outperforms SSCGN by 1.81% on the AI2D dataset. For the FOODWEBS dataset, DisAVR achieves a 1.91% improvement compared with SSCGN but is slightly inferior to HTML. However, HTML performs much worse than our DisAVR on the AI2D dataset. We speculate that HMTL is more suitable for the diagrams of foodwebs type, but DisAVR has better generalization ability with good performance on various diagrams. It is noteworthy that ISAAQ, HMTL, WSTQ, SSCGN and DisAVR achieve a considerable improvement compared with other methods, which may benefit from the representational power of pretrained models. Furthermore, under the same conditions, our proposed DisAVR achieves superior performance, which indicates that DisAVR indeed has good representation and reasoning capabilities and is powerful to solve the diagram question answering task.

2) Results on CSDQA: We compare DisAVR with several methods following only one work on CSDQA [7], and we directly quoted the results of these baselines in [7]. In addition, we also run the released codes of ISAAQ and WSTQ on CSDQA to be our baselines.

Table IV shows the experimental results on validation set of CSDQA. It can be seen that our proposed DisAVR achieves the best performance on all types of questions. We observe that our proposed DisAVR consistently outperforms all the methods in all questions (All), which outperforms the previous best model SSCGN by 0.99%. Although SSCGN has tried to integrate semantically-explicit text features of OCR tokens to enhance diagram semantic understanding, it ignores the detail-aware patch features and pays less attention to reasoning guided by questions. Overall, although DisAVR has achieved the best performance, it is still limited with an accuracy of 60.71% in true-or-false (TF) questions and an accuracy of 43.06% in multiple-choice (MC) questions. Therefore, it is worthy of more in-depth study for diagrams in the computer science domain.

C. Ablation Studies

To further analyze our proposed DisAVR, several ablation studies are conducted on the validation and test set of the AI2D dataset. We investigate the effectiveness of some ablated components by progressively adding them to the BaseModel. Each component demonstrates a contribution to performance improvement as shown in Table V. The details are introduced as follows:

- **BaseModel.** The model follows the ISAAQ [4] trained only on AI2D, which uses a pretrained backbone ResNet-101 to encode the regions, a pretrained RoBERTa-large to encode the question and applies BUTD attention to attend to the question-related regions for question answering.

[2]https://github.com/wyx1516/DisAVR

TABLE IV

ACCURACY (%) OF THE DIAGRAM QUESTION ANSWERING ON THE VALIDATION SET OF CSDQA DATASET, WHICH INCLUDES T/F AND MC QUESTIONS WITH DIFFERENT DIFFICULTIES, WHERE "E" DENOTES THE SIMPLE REASONING QUESTIONS, "C" DENOTES THE COMPLEX REASONING QUESTIONS, AND "ALL" REPRESENTS ALL THE QUESTIONS

Model	Reference	T/F (E)	T/F (C)	T/F (All)	MC (E)	MC (C)	MC (All)	All
Random		50.00	50.00	50.00	25.00	25.00	25.00	37.50
MFB [42]	ICCV'2017	53.14	52.08	56.51	34.72	33.33	30.21	43.36
BAN [43]	NeurIPS'2018	52.08	52.08	57.29	33.33	28.13	27.34	42.32
MCAN [44]	CVPR'2019	56.60	54.17	59.64	34.03	32.29	29.17	44.41
ISAAQ [4]	EMNLP'2020	59.79 †	58.62 †	60.32 †	41.34 †	34.48 †	41.07 †	50.70 †
DPN-QA [7]	TKDD'2022	57.29	59.38	58.85	35.07	33.33	31.77	45.31
WSTQ [5]	TIP'2022	59.28 †	58.62 †	58.62 †	41.04 †	31.03 †	38.49 †	48.55 †
SSCGN [20]	TCSVT'2023	60.31 †	59.48 †	60.52 †	41.75 †	35.34 †	41.27 †	50.90 †
DisAVR	—	**62.37**	**60.34**	**60.71**	**42.53**	**37.07**	**43.06**	**51.89**

TABLE V

THE PERFORMANCE OF ABLATED MODELS WITH DIFFERENT SETTINGS ON THE VALIDATION AND TEST SET OF AI2D DATASET

#	RP-Stream	RT-Stream	Question Parsing	Disentangled Adaptive Reasoning	Val Acc	Δ	Test Acc	Δ
BaseModel					74.28	-	72.90	-
1	✓				74.96	↑ 0.68	73.63	↑ 0.73
2		✓			75.43	↑ 1.15	74.16	↑ 1.26
3	✓	✓			76.29	↑ 2.01	75.05	↑ 2.15
4	✓	✓		✓	76.82	↑ 2.54	75.52	↑ 2.62
5	✓	✓	✓	✓	**77.92**	↑ **3.64**	**76.15**	↑ **3.25**

TABLE VI

ABLATION STUDIES FOR THE IMPACT OF DIFFERENT REASONING CELLS

Model	Val	Δ	Test	Δ
DisAVR	**77.92**	-	**76.15**	-
w/o LocateRegCell	76.50	↓ 1.42	75.15	↓ 1.00
w/o RelateSmRelCell	76.93	↓ 0.99	75.36	↓ 0.79
w/o RelateSpRelCell	76.71	↓ 1.21	75.26	↓ 0.89

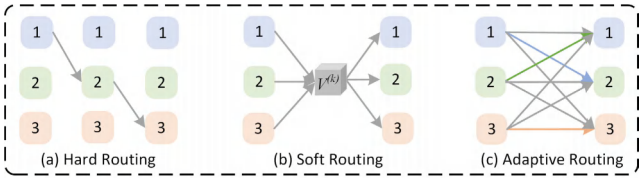
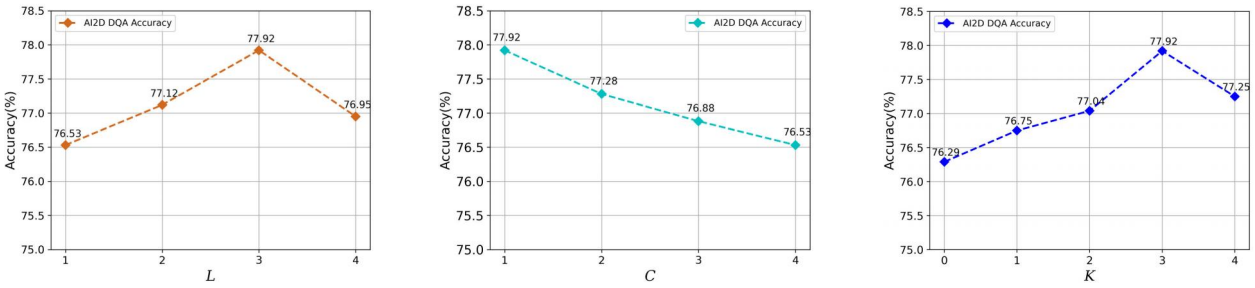


Fig. 5. Different routing mechanisms. (a) Hard routing mechanism selects one cell with the maximum score from three cells in every layer. (b) Soft routing mechanism softly fuses the outputs of the three cells to obtain a set of region features $V^{(k)}$ as the input of the three cells in the next layer. (c) Adaptive routing mechanism constructs fully-connected and multi-layer stacked routing space to allow various reasoning paths to be explored.

- **w/ RP-Stream.** The variant adds RP-Stream based on the BaseModel in row 1, which integrates patch feature with region feature. The accuracy has 0.68% and 0.73% improvements on validation and test set respectively, which demonstrates that detail-aware patch features can help enhance the region features by making up its details and promote diagram understanding.
- **w/ RT-Stream.** The variant adds RT-Stream based on the BaseModel in row 2, which integrates text feature with region feature. We can see that the accuracy increases by 1.15% and 1.26% respectively, which demonstrates that semantically-explicit text features can help enhance the semantics of the region and alleviate the problem of semantic ambiguity effectively.
- **w/ improved region feature learning.** The variant adds an improved region feature learning module based on the BaseModel in row 3, which combines the RP-Stream and RT-Stream. The model achieves 2.01% and 2.15% improvements, which indicates that taking into account both patch features and text features can help obtain more robust region features. This module can also be easily plugged into other task-relevant models to provide comprehensive visual representations.
- **w/ disentangled adaptive reasoning module.** Besides the improved region feature learning module for representation, the variant adds a disentangled adaptive reasoning module in row 4. Note that, it uses the static and fixed question representations encoded by the RoBERTa language model as the question guidance for each cell in each layer.



(a) Number of Interaction Layers (b) Number of CInt Layers (c) Number of Routing Layers

Fig. 6. Analysis on three key hyperparameters including (a) Number of Interaction Layers L, (b) Number of CInt Layers C, and (c) Number of Routing Layers K.

- **DisAVR.** This variant is our full model, which achieves the best performance. The question parsing module dynamically generates three types of question guidance for each cell in each layer of the reasoning module in row 5. This variant achieves a greater performance improvement compared to the variant using the static and fixed question representation to guide reasoning (row 5 vs. row 4). We analyze that distinguishing the question information from different perspectives can help comprehend the question well and help reason flexibly over a diagram based on different question focus. Thus, disentangled adaptive reasoning module plays a more powerful role combined with the question parsing module.

To further analyze the disentangled adaptive reasoning module and validate the contribution of three reasoning cells, we conduct several ablation studies to remove these cells respectively. From Table VI, compared with the combination of all three cells, the performance of these ablation models all degrade, which indicates that each cell can help facilitate reasoning and promote the answer prediction.

To validate the superiority of our adaptive routing mechanism, we introduce the other two routing mechanisms as two variants as shown in Fig. 5. From Table VII, we observe that although three routing mechanisms all have a positive effect on the reasoning, our designed adaptive routing mechanism contributes the most and achieves the best performance. The specific setting and analyses for the two variants are as follows:

- **DisAVR (H).** The variant replaces the adaptive routing mechanism in the disentangled adaptive reasoning module with a hard routing mechanism. The hard routing mechanism selects one of the three cells hardly in every layer. We adopt a Gumbel-Softmax [47] strategy to relax the discrete decision to continuous decision, which enables us to train the model in an end-to-end manner. The variant decreases by 1.35% and 1.17% accuracy on validation and test set respectively.
- **DisAVR (S).** Different from the DisAVR (H), the model adopts a soft routing mechanism to reason, which softly fuses the outputs of the three cells to get a set of regions features as the input of the three cells in the next layer. DisAVR (S) outperforms DisAVR (H) with a margin, which indicates that fusing different types of reasoning information can obtain contextualized representations to promote the reasoning process. However, the performance drops by 1.03% and 0.83% compared with DisAVR with our adaptive routing mechanism.

TABLE VII
THE ABLATION RESULTS FOR DIFFERENT ROUTING MECHANISMS

Model	Val Acc	Δ	Test Acc	Δ
DisAVR	**77.92**	-	**76.15**	-
DisAVR (H)	76.57	↓ 1.35	74.98	↓ 1.17
DisAVR (S)	76.89	↓ 1.03	75.32	↓ 0.83

D. Hyperparameter Analysis

To conduct a more detailed hyperparameter analysis, we consider three key hyperparameters including the number of interaction layers L, the number of Cross-branch Interaction (CInt) layers C, and the number of routing layers K.

1) Analysis on Number of Interaction Layers L: From the experimental results in Fig. 6 (a), we observe that the number of interaction layers $L = 3$ achieves the best performance. The performance of DisAVR with different L does not fluctuate drastically. In particular, the performance for both $L = 2$ and $L = 4$ is relatively close to the highest accuracy. A suitable value $L = 3$ can help fuse information from features of other branches effectively and obtain the comprehensive features for reasoning.

2) Analysis on Number of CInt Layers C: Stacking more CInt layers can make the different-branch features fuse information with each other more frequently. However, the performance decreases as the number of CInt layers increases, as shown in Fig. 6 (b). Therefore, frequent interaction between different level features cannot improve performance. We analyze that in the cross-branch interaction process, the other tokens of the complementary branches are unchanged, so repeated interaction cannot help to obtain richer features. When $C = 1$, the necessary information from the other branch can be integrated into the interaction token of the primary branch.

3) Analysis on Number of Routing Layers K: In DisAVR, a multi-layer stacked routing space is applied to realize the adaptive selection for the reasoning path. We evaluate how the number K can affect the results by assigning it to be varied in [0, 4]. As shown in Fig. 6 (c), the accuracy increases first and then decreases. When $K = 3$, the performance is the best.

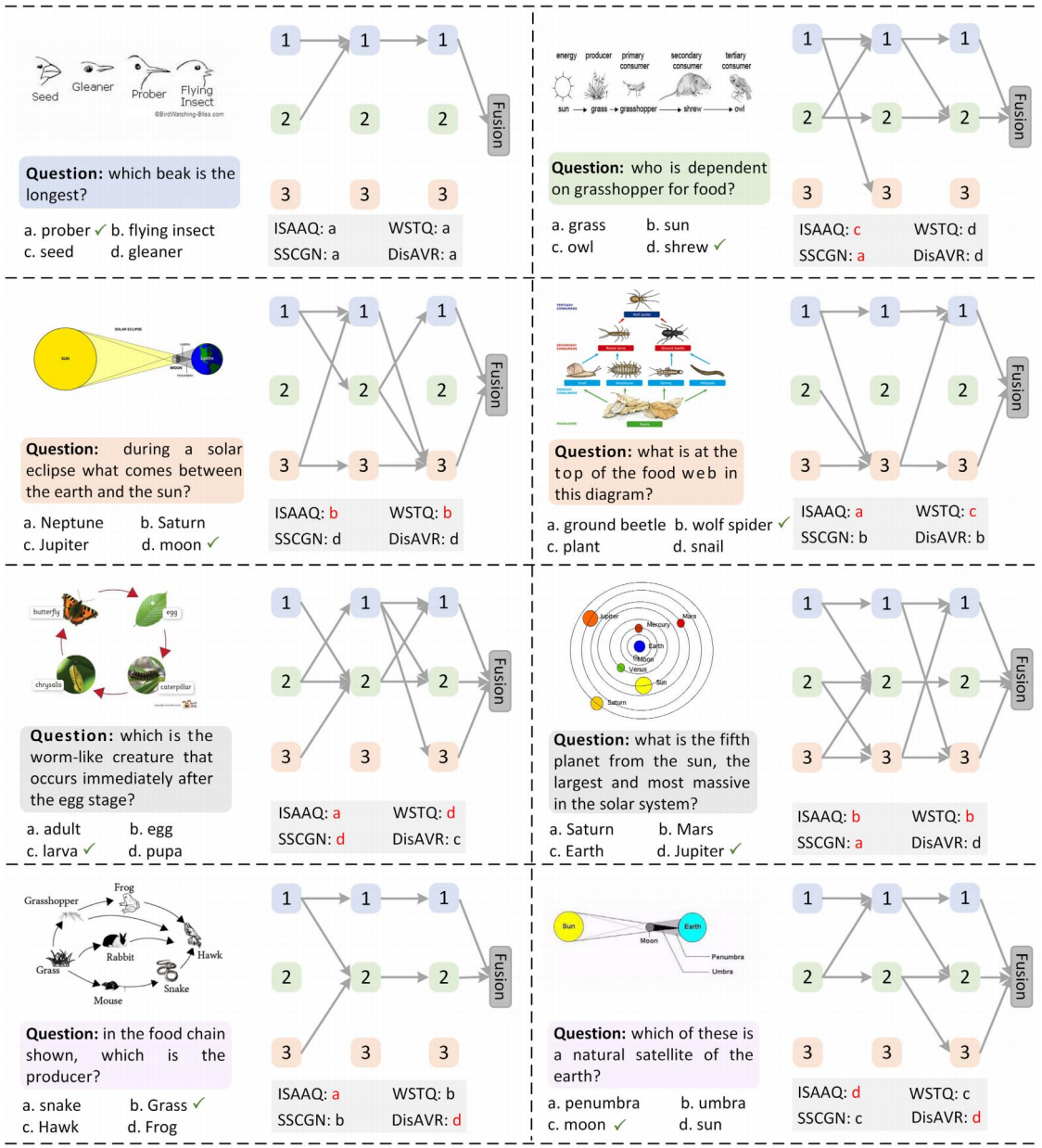


Fig. 7. Visualization of disentangled adaptive reasoning process of DisAVR for different diagram-question pairs. Questions with different background colors focus on information from different perspectives such as isolated object regions (purple), their semantic relations (green) and spatial relations (orange). The question with gray background color attends to multiple perspectives. In addition, LocateRegCell, RelateSmRelCell and RelateSpRelCell are also marked by purple, green and orange, respectively. To be more intuitive, we only show paths with routing probabilities greater than 0.5. The gray paths denote the activated reasoning paths in the layers iteration, which can clarify the contribution of these reasoning cells for feature refinement and answer prediction. The questions with pink background color in the last row denote the failure cases.

We speculate that when $K < 3$, with the small path routing space, it is not sufficient to capture the question-related information to reason about the correct answer, especially for complex scenes. When $K > 3$, with the large path routing space, more redundant information may interfere with the reasoning and model optimization. In addition, we find that insufficient information affects reasoning more than redundant information, although they all harm the performance of diagram question answering.

E. Case Studies

To evaluate the effectiveness of the proposed DisAVR more intuitively, we visualize several examples to analyze the disentangled adaptive reasoning process as shown in Fig. 7, which can help us gain a deep insight into the complex reasoning process and explore the contributions of these reasoning cells for answer prediction. Moreover, for an example among them, we also visualize the attention results of the three reasoning cells in each layer to further interpret the reasoning process of DisAVR as shown in Fig. 8. The details of the analyses include the following two aspects.

1) Analysis of Reasoning Paths: There are two main observations as follows:

- Different diagram questions focusing on information from different perspectives tend to activate different types of reasoning cells. For example, the first question in the

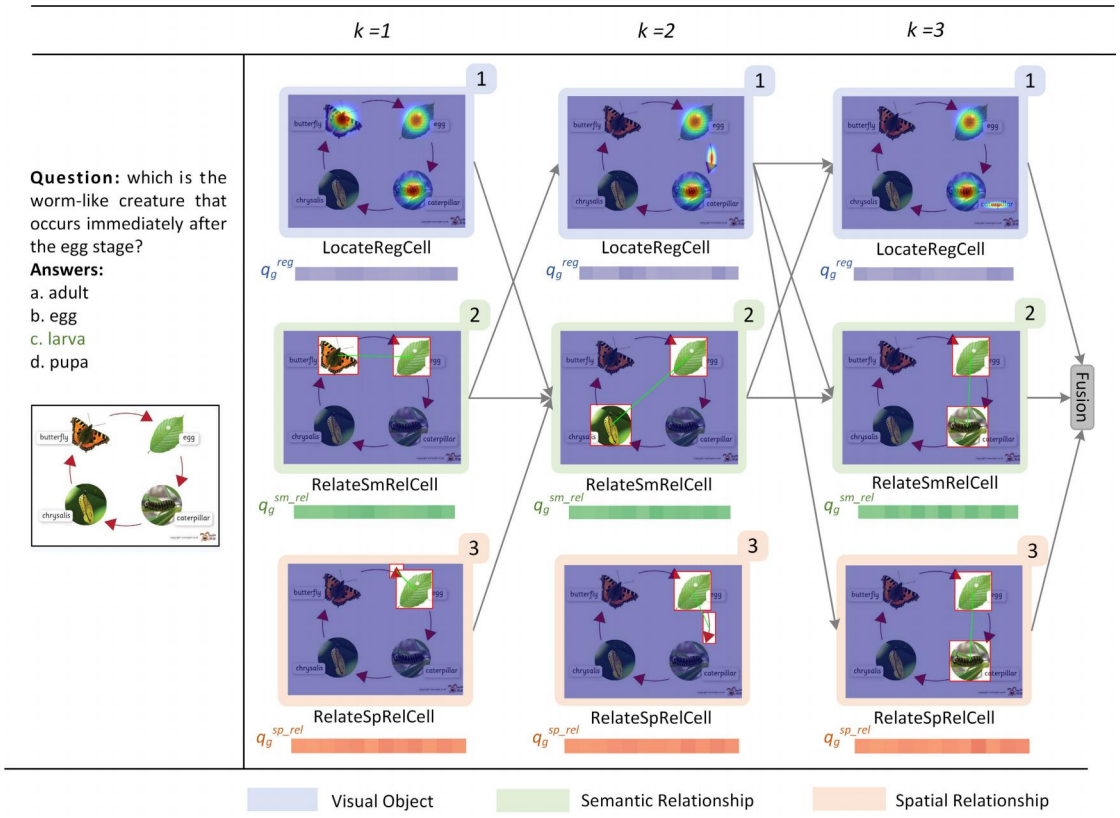


Fig. 8. Visualization of attention results for three reasoning cells along the reasoning process. To present more clearly, for object attention, we highlight the top-3 objects with the highest attention values. For relation attention, we only display one most attended relation. The objects labeled with the red boxes are most affected by each other.

first row needs to attend the object region, which *has the longest beak* through the LocateRegCell. For the second question in the first row, the reasoning mainly involves the object regions and the semantic relationship between *grasshopper* and *shrew*. Thus, it tends to activate LocateRegCell and RelateSmRelCell more frequently in the reasoning process. However, the two questions in the second row require more RelateSpRelCells since they require more spatial relationship information between object regions to reason about the answer.

- More complex diagram questions focusing on the information from multiple perspectives demand more complex reasoning paths, which tend to activate more reasoning cells for a comprehensive diagram representation refinement. Compared with the four examples in the first two rows, the two examples in the third row are more complex. All other three methods struggle to deal with them. Taking the last question as an example, DisAVR needs first to capture all planets in the solar system, then RelateSmRelCell and RelateSpRelCell are utilized to analyze their spatial relations with the *sun* to focus on *the fifth planet from the sun*, and compare their shape size to reason about the *the largest* and *most massive* planet.

In summary, our disentangled adaptive reasoning method allows for performing flexible and adaptive reasoning, which can provide more optimal reasoning paths adapted for each specific diagram-question pair and clarify the contribution of these reasoning cells for answer prediction.

2) Analysis of Reasoning Cells: As shown in Fig. 8, under question guidance for the three reasoning cells, the attentive objects, semantic relationships and spatial relationships change over the reasoning process. In the beginning, LocateRegCell highlights the object regions related to the *egg* stage. RelateSmCell and RelateSpCell highlight the related interactions between the *egg* and the surrounding objects (e.g., *butterfly* and *caterpillar*). Following the reasoning process, LocateRegCell gradually attends to the *caterpillar* to provide the correct answer. RelateSmCell and RelateSpCell also find the most related relationships with *egg*. In fact, the object representations can be refined by considering the semantically and spatially related neighborhoods of an object in the whole reasoning process. Therefore, the final question-relevant contextualized object representations indicating their relationships to other relevant objects can support the final answer prediction.

In addition, we also present some failure cases in Fig. 7 to analyze the limitations of DisAVR. As illustrated in this figure, answering this kind of question in the last row requires understanding the given question combined with some external prior knowledge and predicting an answer by incorporating multimodal knowledge. To be specific, to answer the first question of the last row "in the food chain shown, which is the producer?", the model needs to recognize the objects corresponding to the answers, and combine the basic knowledge about the entity *producer*, such as "green plants are a kind of producer" and "a producer produces its own food" to give the

answer *Grass*. However, compared with WSTQ and SSCGN, our DisAVR lacks the integration of elementary commonsense knowledge necessary for question answering, leading to its difficulty in answering such questions. Therefore, how to integrate the external domain knowledge is also a potential direction to optimize the accuracy of answer prediction.

V. Conclusion

In this paper, we propose a novel DisAVR model, which jointly optimizes the dual-process of representation and reasoning to solve the DQA task. For the representation process, our proposed improved region feature learning module adopts a dual-stream transformer to integrate the detail-aware patch features and semantically-explicit text features with the region features. The module can be easily plugged into other task-relevant models to provide robust visual representations, which can combine with other reasoning models to complete the whole DQA task and can also further facilitate the other downstream task such as image captioning. For the reasoning process, the question parsing module and disentangled adaptive reasoning module work collaboratively. The disentangled adaptive reasoning module decomposes the complex reasoning process by employing three reasoning cells to perform adaptive reasoning on the region, semantic and spatial relations of the diagram under the corresponding question guidance dynamically generated by the question parsing module. This reasoning module endows the reasoning process with flexibility and adaptability. Experimental results on three DQA datasets certify the effectiveness of our proposed DisAVR. In the future, we hope to explore how to integrate external domain knowledge to optimize answer prediction accuracy. We also hope to explore an interpretable reasoning method that combines logical rules and neural networks to provide a more explicit intermediate reasoning process.

Appendix

A. Extra Ablation Studies

1) Impact of Visual Encoders: We have selected several classic CNN backbones including ResNet-50 [35], MobileNetV2 [48], and DenseNet [49] as the variants. The ablation results are reported in Table VIII. We observe that when using DenseNet as the visual encoder, DisAVR achieves a slightly better performance. When using ResNet-50 or MobileNetv2 as the visual encoder, the performance of DisAVR decreases. Although DenseNet can achieve higher performance, we use ResNet-101 as the visual encoder to ensure a fair comparison with the baselines.

2) Impact of Question Types: The datasets lack annotations for the difficulty of the questions, such as simple or complex questions, and the annotation work is also time-consuming and labor-intensive. Therefore, to analyze the number of simple and complex reasoning questions on the model performance, we have labeled 50 simple questions and 50 complex questions. We set simple and complex questions at different ratios including 5:1 and 1:5. The experimental results are shown in Table IX. For each model, we can observe that as the number of complex questions increases, the accuracy decreases.

TABLE VIII
Ablation Studies for the Impact of Visual Encoders

Backbone	Val	Δ	Test	Δ
Resnet-101 [35]	77.92	-	76.15	-
Resnet-50 [35]	77.15	↓ 0.77	75.26	↓ 0.89
MobileNetv2 [48]	76.82	↓ 1.10	75.04	↓ 1.11
DenseNet [49]	78.03	↑ 0.11	76.38	↑ 0.23

TABLE IX
Analysis for the Number of Simple and Complex Reasoning Questions on Model Performance

Model	Percentage (S:C)	
	5:1	1:5
ISAAQ [4]	70.00	51.67
WSTQ [5]	68.33	53.33
SSCGN [20]	71.67	55.00
DisAVR	**73.33**	**60.00**

TABLE X
Generalizability Comparison of Different Models

Model	AI2D→ CSDQA	CSDQA→ FOODWEBS	FOODWEBS→ AI2D
ISAAQ [4]	27.38	26.74	69.33
WSTQ [5]	28.37	27.98	70.35
SSCGN [20]	27.98	28.65	70.76
DisAVR	**29.76**	**29.13**	**72.80**

In both settings, our model outperforms existing models. Moreover, when there is a higher proportion of complex problems (S:C = 1:5), DisAVR can achieve significantly better performance compared with the other models. We analyze that based on enhanced diagram representation, our disentangled adaptive reasoning design can help adapt to simple or complex reasoning questions by assigning these three reasoning cells with different attention weights. This is why our model is able to achieve performance gains.

B. Generalizability Analysis

To demonstrate the generalization ability of DisAVR, we have conducted cross-database tests on the three datasets including AI2D, FOODWEBS and CSDQA. We compare several baselines incuding ISAAQ [4], WSTQ [5] and SSCGN [20] with our DisAVR, and the results are reported in Table X. We can make the following three observations:

- When training on AI2D and validating on CSDQA, or training on CSDQA and validating on FOODWEBS, all these four models perform poorly due to the large gap between the diagrams of the computer science domain and the natural science domain.
- When training on FOODWEBS and validating on AI2D, the models can achieve good performance, because the diagram types are similar across the two datasets.

TABLE XI
ACCURACY (%) OF THE DIAGRAM CLASSIFICATION TASK

Model	Acc
Original diagram	77.58
Original diagram + region features	82.42
Original diagram + IRFL	**85.45**

- In all three settings, DisAVR achieves better performance compared with the other three baselines, demonstrating its superior generalizability.

C. Module Scalability Analysis

To support our claim that the IRFL module can also be easily plugged into other task-relevant models to obtain robust visual representations for diagrams, we have conducted experiments on the diagram classification task following [7]. The experimental results are shown in Table XI. Since the source code in [7] is not publicly released, we have implemented several baselines ourselves. Specifically,

- Original diagram: it refers to the global visual features of the diagram encoded by the CNN models. Here we select the ResNet-50 [35] backbone following [7].
- Original diagram + region features: it combines the global diagram features with the local region features represented by ResNet-50.
- Original diagram + IRFL: it combines the global diagram features with the locally enhanced region features obtained by our proposed Improved Region Feature Learning (IRFL) module.

The performance gains demonstrate that our proposed IRFL module can help obtain robust visual representations for diagrams and enhance diagram understanding, further prompting the diagram classification task.

REFERENCES

[1] S. Antol et al., "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2425–2433.

[2] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi, "A diagram is worth a dozen images," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 235–251.

[3] D. Kim, Y. Yoo, J. Kim, S. Lee, and N. Kwak, "Dynamic graph generation network: Generating relational knowledge from diagrams," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4167–4175.

[4] J. M. Gomez-Perez and R. Ortega, "ISAAQ—Mastering textbook questions with pre-trained transformers and bottom-up and top-down attention," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 5469–5479.

[5] J. Ma, Q. Chai, J. Huang, J. Liu, Y. You, and Q. Zheng, "Weakly supervised learning for textbook question answering," *IEEE Trans. Image Process.*, vol. 31, pp. 7378–7388, 2022.

[6] Z. Yuan, X. Peng, X. Wu, and C. Xu, "Hierarchical multi-task learning for diagram question answering with multi-modal transformer," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 1313–1321.

[7] S. Wang et al., "Computer science diagram understanding with topology parsing," *ACM Trans. Knowl. Discovery Data*, vol. 16, no. 6, pp. 1–20, Dec. 2022.

[8] X. Hu, L. Zhang, J. Liu, Q. Zheng, and J. Zhou, "Fs-DSM: Few-shot diagram-sentence matching via cross-modal attention graph model," *IEEE Trans. Image Process.*, vol. 30, pp. 8102–8115, 2021.

[9] W. Jiang, M. Zhu, Y. Fang, G. Shi, X. Zhao, and Y. Liu, "Visual cluster grounding for image captioning," *IEEE Trans. Image Process.*, vol. 31, pp. 3920–3934, 2022.

[10] S. E. Kahou, V. Michalski, A. Atkinson, A. Kádár, A. Trischler, and Y. Bengio, "FigureQA: An annotated figure dataset for visual reasoning," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–20.

[11] K. Kafle, M. Yousefhussien, and C. Kanan, "Data augmentation for visual question answering," in *Proc. 10th Int. Conf. Natural Lang. Gener.*, 2017, pp. 198–202.

[12] N. Methani, P. Ganguly, M. M. Khapra, and P. Kumar, "PlotQA: Reasoning over scientific plots," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1516–1525.

[13] A. Masry and E. Hoque, "Integrating image data extraction and table parsing methods for chart question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 1–5.

[14] A. Masry, D. X. Long, J. Q. Tan, S. R. Joty, and E. Hoque, "ChartQA: A benchmark for question answering about charts with visual and logical reasoning," in *Proc. Findings Assoc. Comput. Linguistics*, 2022, pp. 2263–2279.

[15] D. H. Kim, E. Hoque, and M. Agrawala, "Answering questions about charts and generating visual explanations," in *Proc. CHI Conf. Human Factors Comput. Syst.*, Apr. 2020, pp. 1–13.

[16] M. J. Seo, H. Hajishirzi, A. Farhadi, O. Etzioni, and C. Malcolm, "Solving geometry problems: Combining text and diagram interpretation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1466–1476.

[17] J. Chen et al., "GeoQA: A geometric question answering benchmark towards multimodal numerical reasoning," in *Proc. Findings Assoc. Comput. Linguistics*, 2021, pp. 513–523.

[18] P. Lu et al., "Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 6774–6786.

[19] J. Krishnamurthy, O. Tafjord, and A. Kembhavi, "Semantic parsing to probabilistic programs for situated question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 160–170.

[20] Y. Wang, B. Wei, J. Liu, Q. Lin, L. Zhang, and Y. Wu, "Spatial-semantic collaborative graph network for textbook question answering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 7, pp. 3214–3228, Jul. 2023.

[21] J. Liu, L. Wang, and M.-H. Yang, "Referring expression generation and comprehension via attributes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4866–4874.

[22] X. Rong, C. Yi, and Y. Tian, "Unambiguous scene text segmentation with referring expression comprehension," *IEEE Trans. Image Process.*, vol. 29, pp. 591–601, 2020.

[23] D. Teney, L. Liu, and A. Van Den Hengel, "Graph-structured representations for visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3233–3241.

[24] W. Norcliffe-Brown, S. Vafeias, and S. Parisot, "Learning conditioned graph structures for interpretable visual question answering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8344–8353.

[25] R. Hu, A. Rohrbach, T. Darrell, and K. Saenko, "Language-conditioned graph networks for relational reasoning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10293–10302.

[26] T. Minh Le, V. Le, S. Venkatesh, and T. Tran, "Dynamic language binding in relational visual reasoning," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 818–824.

[27] C. Jing, Y. Jia, Y. Wu, X. Liu, and Q. Wu, "Maintaining reasoning consistency in compositional visual question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5089–5098.

[28] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 2901–2910.

[29] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 39–48.

[30] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, "Learning to reason: End-to-end module networks for visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 804–813.

[31] R. Hu, J. Andreas, T. Darrell, and K. Saenko, "Explainable neural computation via stack neural module networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 53–69.

[32] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic VQA: Disentangling reasoning from vision and language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1039–1050.
[33] R. Vedantam, K. Desai, S. Lee, M. Rohrbach, D. Batra, and D. Parikh, "Probabilistic neural symbolic models for interpretable visual question answering," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6428–6437.
[34] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[36] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–22.
[37] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[38] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[39] P. Anderson et al., "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6077–6086.
[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[42] Z. Yu, J. Yu, J. Fan, and D. Tao, "Multi-modal factorized bilinear pooling with co-attention learning for visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1839–1848.
[43] J.-H. Kim, J. Jun, and B.-T. Zhang, "Bilinear attention networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1571–1581.
[44] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6274–6283.
[45] M. Haurilet, A. Roitberg, and R. Stiefelhagen, "It's not about the journey; It's about the destination: Following soft paths under question-guidance for visual reasoning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1930–1939.
[46] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–19.
[47] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–13.
[48] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[49] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

Yaxian Wang received the B.E. degree from Hunan Normal University, Changsha, China, in 2019. She is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China. Her research interests include visual question answering, cross-modal learning, and interpretable reasoning.

Bifan Wei received the Ph.D. degree in computer science and technology from Xi'an Jiaotong University in 2014. He is currently a Research Fellow in computer science and technology with Xi'an Jiaotong University. His research interests include web data mining, taxonomy learning, and distance education.

Jun Liu (Senior Member, IEEE) received the B.S. and Ph.D. degrees in computer science and technology from Xi'an Jiaotong University in 1995 and 2004, respectively. He is currently a Professor with the School of Computer Science and Technology, Xi'an Jiaotong University, China. His current research interests include data mining and text mining and he has authored more than 70 research papers in various journals and conference proceedings. He served as a Guest Editor for many technical journals, such as *Information Fusion*, IEEE SYSTEMS JOURNAL, and *Future Generation Computer Systems*. He also acted as a conference/workshop/track chair at numerous conferences.

Lingling Zhang received the Ph.D. degree in computing science from Xi'an Jiaotong University in 2020. She is currently an Assistant Professor in computer science with Xi'an Jiaotong University. She was a Visiting Student with the School of Computer Science, Carnegie Mellon University, working with Prof. A. Hauptmann. Her research interests include cross-media information mining, computer vision, zeroshot learning, and few-shot learning.

Jiaxin Wang received the B.S. and M.S. degrees in communication and information systems from Northwestern Polytechnical University, Xi'an, Shaanxi, China, in 2017 and 2020, respectively. She is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an. Her research interests include natural language processing, knowledge graphs, and few-shot learning.

Qianying Wang received the master's degree in electronic engineering and industrial management from Shanghai Jiao Tong University, and the Ph.D. degree in human computer interaction from Stanford University. She is currently a corporate Vice President of Lenovo Group, and the Head of Smart Education and Future Interaction Laboratory. Her research interests include human-computer interaction, AI empowered smart education, hyper reality space computing, and immersive media.