CoG-DQA: Chain-of-Guiding Learning with Large Language Models for Diagram Question Answering

Shaowei Wang[1,2], Lingling Zhang[1,2]*, Longji Zhu[1,2], Tao Qin[1,2] Kim-Hui Yap[3], Xinyu Zhang[1,2], Jun Liu[1,2]
[1]School of Computer Science and Technology, Xi'an Jiaotong University, China
[2]Key Laboratory of Intelligent Networks and Network Security (Xi'an Jiaotong University), Ministry of Education, Xi'an, Shaanxi, China
[3]School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore

Abstract

Diagram Question Answering (DQA) is a challenging task, requiring models to answer natural language questions based on visual diagram contexts. It serves as a crucial basis for academic tutoring, technical support, and more practical applications. DQA poses significant challenges, such as the demand for domain-specific knowledge and the scarcity of annotated data, which restrict the applicability of large-scale deep models. Previous approaches have explored external knowledge integration through pre-training, but these methods are costly and can be limited by domain disparities. While Large Language Models (LLMs) show promise in question-answering, there is still a gap in how to cooperate and interact with the diagram parsing process. In this paper, we introduce the Chain-of-Guiding Learning Model for Diagram Question Answering (CoG-DQA), a novel framework that effectively addresses DQA challenges. CoG-DQA leverages LLMs to guide diagram parsing tools (DPTs) through the guiding chains, enhancing the precision of diagram parsing while introducing rich background knowledge. Our experimental findings reveal that CoG-DQA surpasses all comparison models in various DQA scenarios, achieving an average accuracy enhancement exceeding 5% and peaking at 11% across four datasets. These results underscore CoG-DQA's capacity to advance the field of visual question answering and promote the integration of LLMs into specialized domains.

1. Introduction

Visual Question Answering (VQA) is defined as providing answers to questions in natural language, leveraging contextual information extracted from natural images. It has gained widespread recognition due to its immense utility in various scenarios [5, 14, 23]. Building upon the principles of VQA, Diagram Question Answering (DQA) specifically targets the interpretation of complex diagrammatic information. This includes a variety of visual representations of data such as flow charts, graphs, and schematic illustrations, which are crucial in specialized fields such as engineering, medicine, and education. DQA has emerged as a compelling field, drawing attention to its capacity to evaluate the intricate reasoning abilities of models. This interest is mainly due to the ability of diagrams to effectively represent complex knowledge concepts and logical relationships [15–17]. The ability to accurately interpret and respond to questions about these diagrams is of significant value for academic tutoring, technical assistance, and a variety of practical applications [6, 42].

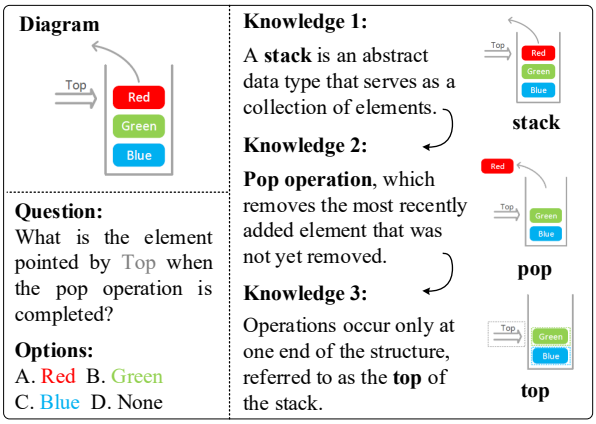

Figure 1. An example of DQA in the Computer Science domain.

While DQA has potential application value, it is accompanied by significant challenges. The complexity of DQA lies in its knowledge-intensive nature, which demands not just an understanding of abstract visual representations, but also a robust grasp of domain-specific knowledge. For instance, answering a diagram question in the Computer Science domain, as shown in Fig. 1, without prior knowledge

*Corresponding author, email: zhanglling@xjtu.edu.cn

is a formidable task. Analyzing the context and candidate choices reveals the necessity of domain-specific knowledge, such as concept definitions (knowledge 1 and 3) and explanations of the operation (knowledge 2) from the right side of the figure. Under the guidance of this background knowledge, combined with visual features such as objects and relationships parsed from the diagram, the model can perform step-by-step reasoning to arrive at the correct answer. Moreover, diagram datasets often revolve around specific domains, like biology or science, necessitating annotators with in-depth domain knowledge, which is costly to obtain. Commonly used diagram datasets typically comprise thousands of diagrams (e.g., AI2D [16]: ∼3,000 diagrams and ∼9,000 questions). Limited samples and annotations pose challenges for models to acquire sufficient background knowledge from the initial parameter status, thereby intensifying the complexity of the subsequent inference process.

Faced with these challenges, previous researchers have introduced external knowledge into the inference model through pretrained language models. The first type of method retains all parameter architecture of the language model, uses a large amount of external knowledge base to pre-train it, and fine-tunes it on specific domains [11, 31, 41]. However, the language model is pre-trained on external data in the large-scale generalization domain, with significant differences from the data in the specific DQA domains. Therefore, it is difficult to achieve a major breakthrough in performance by using a small amount of DQA data to fine-tune its massive parameters. Furthermore, DQA datasets from different domains exhibit significant differences. This diversity makes it impractical and costly to fine-tune different models separately for each domain.

The emergence of large language models (LLMs) offers a promising avenue for improving common-sense question-answering. Some researchers propose to search and introduce rich background knowledge in LLMs through prompt engineering. This type of method either uses an expert model to convert visual content into textual content and then uses LLMs to complete inference [28], or uses a local learnable interface in the LLM to fine-tune to introduce visual features [44, 49]. As the parameters of the pre-trained model increase, the amount of external knowledge that can be introduced increases, which further improves the question-answering performance. However, the prompt templates for LLMs of such methods are only aimed at obtaining the final answer and have almost no interaction or correlation with the visual features parsed from the diagrams. This makes inference performance heavily dependent on the LLM's prior knowledge, and the impact brought by visual features that should be more important becomes marginal. The latest benchmarks also show that even the latest GPT-4Vision [43] and LLaVA-1.5 [25] tend to prioritize the prior knowledge and give incorrect answers for most questions under this type of prompt and suffer from language hallucination, and the ability to abstract visual content is still limited [24]. Therefore, how to utilize prompt to leverage LLM's rich background knowledge and effectively relate the parsing process of diagrams remains a challenge.

In this paper, we propose the **C**hain-**o**f-**G**uiding Learning Model for **D**iagram **Q**uestion **A**nswering (CoG-DQA) to address the above challenges. Specifically, CoG-DQA is a novel and general framework that effectively introduces large language models as external knowledge to solve various DQA scenes. Under the CoG-DQA framework, LLM plays a guiding role and cooperates with diagram parsing tools (DPTs) to parse features with rich background knowledge. Both LLMs and DPTs are agnostic under this framework, ensuring that the model's performance will consistently align with contemporary developments. Moreover, CoG-DQA leverages the prompt of the large model for guidance and fine-tunes a small model for inference, effectively bridging domain gaps. Our experiments demonstrate that CoG-DQA outperforms other baseline methods and exhibits superior performance on four DQA datasets.

2. Related Work

2.1. Diagram Parsing

This early type of research on diagrams began in the 1990s, with early researchers primarily employing rule-based methods to perform tasks such as diagram classification and diagram element identification. For example, Watanabe *et al*. [39] introduced a technique for examining diagrams in pictorial books of flora (PBF), leveraging both natural language and layout information. Ferguson *et al*. [7] developed a spatial reasoning engine capable of generating qualitative spatial descriptions from line drawings. They also presented a model for detecting repetition and symmetry, which mirrors human cognitive processes when interpreting repetition-based diagrams [8]. Subsequently, Futrelle *et al*. [9] conducted research on extracting diagrams from PDF documents, performing classification tasks specifically on the bar and non-bar diagrams. Seo *et al*. [36] identified visual elements in a diagram while maximizing agreement between textual and visual data to build an automated system that provides support for geometric diagram reasoning. Zhang *et al*. [48] proposed the first end-to-end deep learning model for geometry diagram parsing, which gives explicit primitive instance extraction, classification, and between-primitive relationship reasoning.

2.2. Diagram Question Answering

As research progressed, diagram parsing and question analysis evolved beyond being bottlenecks in DQA performance. Contemporary research primarily addresses the introduction of external knowledge through pre-training to

tackle the background knowledge and limited sample challenges inherent to DQA. The first category of methods introduces background knowledge in the form of language model fine-tuning after pre-training. For instance, Gomez-Perez *et al*. [11] proposed fine-tuning pre-trained transformers to incorporate richer background knowledge of textual and visual modalities. Ma *et al*. [30] introduced two weakly supervised pre-training tasks aimed at enhancing text comprehension and diagram semantics. Xu *et al*. [41] introduced a multistage domain pre-training module with unsupervised post-pretraining using a span mask strategy and supervised pre-finetuning. Importantly, the post-pretraining phase utilized a heuristic generation algorithm to incorporate terminology from external knowledge bases.

Leveraging the capabilities of large language models (LLMs), researchers have explored the incorporation of their rich prior knowledge into DQA using prompt methods. Lu *et al*. [28] presented Science Question Answering (ScienceQA) with diverse science topics and detailed annotations of answers alongside corresponding lectures and explanations, and conducted initial benchmarking using prompt-based LLMs. Zhang *et al*. [49] proposed a two-stage framework, by fusing both vision and language representations for LLMs and performing chain-of-thought prompts on DQA. Yao *et al*. [44] introduced a multimodal graph-of-thought, integrating text and visual features for LLMs. In summary, DQA research has evolved from the early stages of diagram parsing and question analysis to addressing the effective transfer of knowledge. The trend reflects an increase in the amount of prior knowledge introduced, although challenges related to the fine-tuning of language model parameters and bridging the gap between general and specific domains remain.

3. Method

As shown in the left of Fig. 2, the input of the diagram question answering (DQA) task contains a diagram d, a natural language question q, a piece of natural language context c (optional), and candidate answer set $\mathcal{A}$. The DQA task is to predict the correct answer $\hat{a}$ in $\mathcal{A}$ according to d, q, and c. The formal definition is as follows:

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

where θ is the trainable parameter of the network, $\mathcal{A} = \{a_1, a_2, ..., a_k\}$ (for multiple-choice questions $k = 4 - 6$, for true-or-false questions $k = 2$). According to the above task definition, our model aims at selecting the most probable answer $\hat{a}$. Fig. 2 illustrates an overview of our Chain-of-Guiding Learning Model for Diagram Question Answering (CoG-DQA). The two core modules of our CoG-DQA model are the Chain-of-Guiding Learning Module (Sec. 3.1) and the Answer Inference Module (Sec. 3.2).

3.1. Chain-of-Guiding Learning Module

The Chain-of-Guiding Learning Module (CoG-LM) aims to effectively utilize the capabilities of both the pre-trained Large Language Models (LLMs) and Diagram Parsing Tools (DPTs) in the context of Diagram Question Answering (DQA). As illustrated in the middle part of Fig. 2, we divide the CoG-LM into three stages based on the feature extraction process, from coarse-grained to fine-grained. Each stage contains three components: **DPT processing** which operates with frozen parameters, **LLM guiding** which uses prompt method, and **Interaction** to combine the results of the above two. In this way, the module can facilitate efficient diagram processing without the need for costly training, while providing complete visual features and knowledge for the downstream inference module.

As shown in Fig. 3, the **LLM guiding** component in each stage adopts a multi-turn conversation template. The X_{system} serves as a standardized prompt and is uniformly set to 'As a question-answering assistant, you need to answer a question based on a diagram and possible context.' In each turn, five different guiding heads are manually defined, ensuring prompt diversity (details in supplementary material). The LLM's response in the i-th turn is recorded as X^i_{answer}. For the **DPT processing** component, its input and output vary from stage to stage, but the core difference is on visual object features at different granularities in the diagram. The Interaction component takes the output of the first two components as input to interact between knowledge and features, which may include operations such as supplementation, filtering, and attention. Each stage is described in detail below in the form of three components.

3.1.1 Stage 1: Global Knowledge Supplement.

To answer cross-modality questions, the inference process relies on various sources, including the diagram, textual context, and candidate choices. In most scenarios, a brief context alone is insufficient to support the parsing of diagram and the reasoning process. In the first stage, we supplement global background knowledge and measure the correlation of features between different modalities.

LLM Guiding 1: During the first stage, the input to the LLM guiding includes X_{system}, X^0_{input}, and $X^1_{guide\_head}$. Among them, X^0_{input} includes question, context, choices, and captions of the diagrams. During this turn, the guiding head seeks to obtain the background knowledge $K_b = X^1_{answer}$ required outside the known context.

DPT Processing 1: The inputs are the diagram d, the question q, and the context c. We employ a pre-trained visual encoder (such as ResNet [13]) to encode the global visual features of the diagram and obtain an $N-$dimensional feature vector D_g. Similarly, questions and context are encoded using a pre-trained encoder (such as RoBERTa [26])

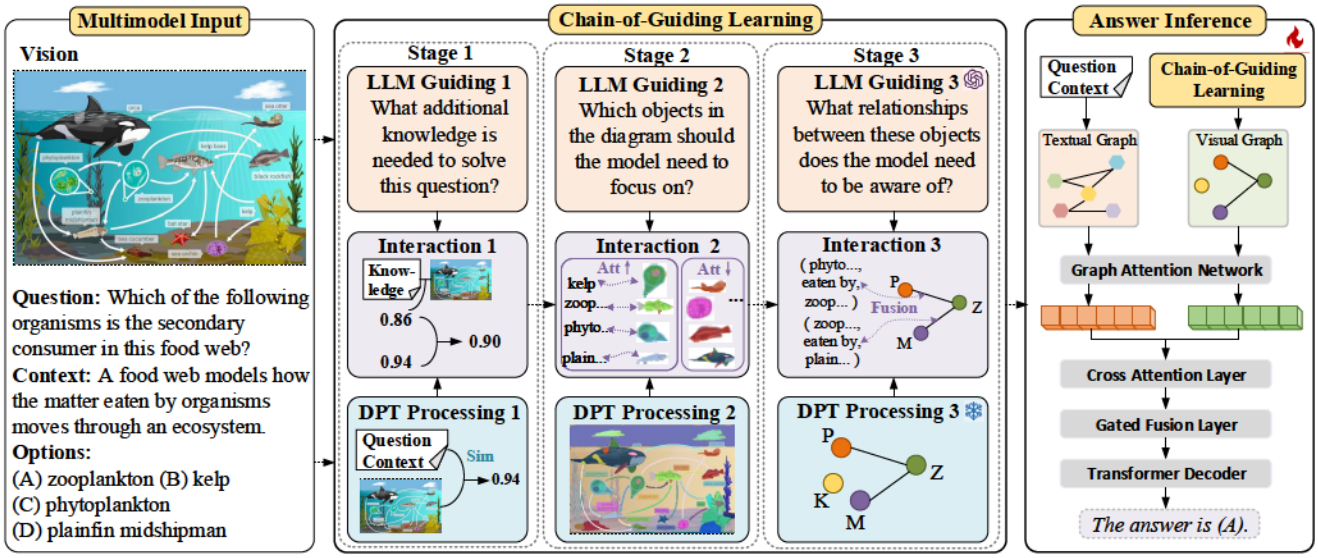

Figure 2. Overview of the Chain-of-Guiding Learning Model for Diagram Question Answering (CoG-DQA).

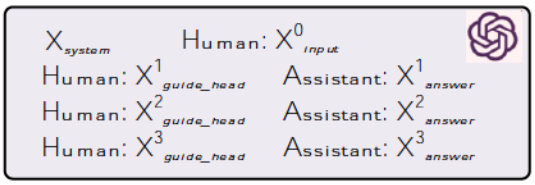

Figure 3. Input and output sequences of LLM. X_{system} = As a question-answering assistant, you need to answer a question based on a diagram and possible context. X^0_{input} contains the question, context, choices, and caption of the diagram. $X^i_{guide\_head}$ represents the text of guide heads. X^i_{answer} represents LLM's answers.

to obtain features of the same dimension features vector T_g:

$$D_g = FC(CNNs(d)), \tag{2}$$

$$T_g = FC(RoBERTa([q;c])), \tag{3}$$

where FC denotes the fully connected layer used for feature projection, $[\cdot]$ is the concatenation. The cos similarity is used to measure the relevance between the features of the two modalities s_r:

$$s_r = cos(D_g, T_g). \tag{4}$$

Interaction 1: Similarly, we calculate the similarity s_l between background knowledge and diagrams (limited to the interval [0,1]):

$$T_g = FC(RoBERTa(K_b)), \tag{5}$$

$$s_l = cos(D_g, K_b). \tag{6}$$

Finally, by calculating the arithmetic mean of s_l and s_r, the importance of the visual feature Att_d is obtained, which is used by the subsequent inference module.

3.1.2 Stage 2: Visual Objects Extraction.

Object detection serves as an intermediate step in the DQA task, facilitating the extraction of pertinent visual object features. Although existing DPTs excel at instance-level localization and segmentation, they may benefit from external knowledge when assigning semantic labels and filtering objects relevant to specific questions.

LLM Guiding 2: The input contains the conversation history of the previous turns H^1 and $X^2_{guide\_head}$. In the sequence X^2_{answer}, from which the semantic labels of the object M are obtained, denoted as S_d.

DPT Processing 2: We use pre-trained object detection or instance segmentation models (e.g., SAM [21], YOLO [37]) to localize only the instances in the diagram, obtaining K location masks $M_d \in \mathbb{R}^{K\times W\times H}$, where W and H are the width and height of diagram d respectively.

Interaction 2: Inspired by [51], we utilize the pre-trained CLIP [34] model to associate the mask matrix of the object with semantic labels. Specifically, the corresponding text embeddings Se_d of the texts S_d are extracted using the CLIP model. The respective diagram object embeddings Me_d are then determined and matched to the intrinsic features of each mask using a similarity metric. The mask with the highest similarity score to the object embeddings of the text prompt is then selected. For the $m-$th semantic label:

$$socre_m = \textit{Softmax}(\mathbf{Se_d^m} Me_d{}^T), \tag{7}$$

$$ind_m = \arg\max_{k \in K} socre_m, \tag{8}$$

$$mask_m = M_d[ind_m], \tag{9}$$

where the $\textit{softmax}(\cdot)$ constrains different similarities between $[0, 1]$, $\mathbf{Se_d^m}$ denotes the embedding of the $m-$th semantic label, ind_m denotes the index of mask index aligned with the $m-$th semantic label, $[(\cdot)]$ represents the operation of getting the index value in the vector. Finally, the M aligned masks are concatenated together to form the new mask matrix $\hat{M} \in \mathbb{R}^{M \times W \times H}$. Correspondingly, the remaining unaligned masks are formed as $\overline{M} = M_d \setminus \hat{M}$. The objects corresponding to these mask matrices $\hat{M}$ and $\overline{M}$ can easily obtain object-level embedding features using the pre-trained encoder, which are denoted as $\hat{F}_d$ and $\overline{F_d}$.

3.1.3 Stage 3: Visual Relationships Generation.

There are complex relationships between objects in the diagram, and question-answering needs to use these relationships to form a chain of reasoning to obtain answers. The existing DPT achieves excellent performance in determining whether there is a relationship between two objects, but the relationship is only represented by a binary indicator of $\{0, 1\}$ or a value in the $[0, 1]$ interval. Diagram question-answer reasoning is very sensitive to the specific semantics of the relationships between objects, and binary relationships are not enough to support this precise reasoning.

LLM Guiding 3: The input contains the conversation history of the previous turns H^2 and $X^3_{guide\_head}$. In the sequence X^3_{answer}, N triples of the shape (Object A, Relation, Object B) can be obtained, denoted as Trp_d.

DPT Processing 3: We use the similarity between visual object embedding features to construct relationship values for efficiency, this construction method can also make use of existing pre-trained models (such as [18]). Using the intermediate results in stage 2, relationship generation is performed on the aligned visual objects:

$$socre_s = \textit{Softmax}(\hat{F}_d \hat{F}_d{}^T), \tag{10}$$

$$socre_p = \textit{Softmax}(\hat{M} \hat{M}^T), \tag{11}$$

$$score = \alpha socre_s + \beta socre_p, \tag{12}$$

where the $\textit{softmax}(\cdot)$ constrains similarities between $[0, 1]$, and the α and β are adjustable trade off hyperparameters. For guiding head 3, we expect the LLM to give specific semantics of the relationships between objects in S_d.

Interaction 3: Using the above results, the visual graph $\mathcal{G}_v = \{\mathcal{N}_v, \mathcal{E}_v, \mathcal{A}_v, \mathcal{X}_v\}$ in the diagram can be constructed. $\mathcal{N}_v$ represents the node set in the graph, including all visual objects aligned in stage 2, and $\mathcal{E}_v$ represents the edge set between nodes. In the matrix $score$, if $score_{(i,j)}$ is greater than the gating threshold g, it is determined that there is an edge $e_{(i,j)}$ between nodes i and j. $\mathcal{A}_v$ represents the adjacency matrix of the graph $\mathcal{G}_v$. If there is an edge between nodes i and j, $\mathcal{A}_{v(i,j)} = 1$, otherwise it is 0. The feature set $\mathcal{X}_v$ is composed of the node feature set $\mathcal{X}_v = \hat{F}_d$.

3.2. Answer Inference

The answer inference module mainly uses the features processed by the previous module to perform reasoning to obtain the answer. This module can use any existing question-answering inference module and match it with the **Chain-of-Guiding Learning Module** as a feature extractor. However, in order to make full use of the features processed by the previous module as much as possible, we propose an **Answer Inference module** based on a dual-graph structure because the graph is an excellent feature organization form.

As shown in the right part of Fig. 2, the answer inference module is the only part of the framework that requires training. It takes a textual graph and a visual graph as input as two branches. Among them, the visual graph is constructed in the CoG learning module. In addition, we adopt a textual graph construction (TGC) method based on triplets extraction and coreference resolution inspired by [44]. Specifically, TGC is divided into two steps. It begins by extracting deductive triplets ($\mathcal{T}$) from the input data. Each triplet, represented as $t_i = (t^i_x, t^i_y, t^i_z)$. Edges e^i_{xy} and e^i_{yz} connect these textual nodes, forming the initial raw graph. Secondly, TGC identifies and clusters nodes in the graph that refer to the same mentions, performing coreference resolution. The nodes within a coreference cluster are replaced with the most representative mention. This coreference resolution technique results in denser thought graphs, empowering the model with improved deductive reasoning capabilities.

For j nodes $\mathcal{N}_t = \{n^t_1, ..., n^t_j\}$ in the textual raw graph, we use the text encoder to obtain the node embedding features:

$$v^t_j = Encoder(n^t_j), \tag{13}$$

where the encoder can be selected such as BERT and RoBERTa [26], $X_t = \{v^t_1, ..., v^t_j\}$ constitutes the node feature set of the textual graph $\mathcal{G}_t$.

We employ the graph attention network (GAT) to encode the dual graph $\mathcal{G}_v$ and $\mathcal{G}_t$. Taking visual graph $\mathcal{G}_v$ as an example, the GAT layer is defined as follows:

$$e_{ij} = \text{LeakyReLU}\left(a^T [W \cdot f_i \| W \cdot f_j]\right), \tag{14}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}, \tag{15}$$

$$h'_i = \sigma\left(\sum_{j \in \mathcal{N}_v i} \alpha_{ij} (W \cdot X_j)\right), \tag{16}$$

Where e_{ij} is the attention score between nodes n_{vi} and n_{vj}, $\|$ denotes concatenation, $W \cdot f_i$ and $W \cdot f_j$ are linear transformations of the node features, α_{ij} is the attention coefficient, h'_i is the updated feature for node n_{vi}, σ represents an activation function (e.g., ReLU). We then use a single-layer feed-forward neural network (FFN) to obtain the final visual graph embedding H_v^G:

$$h' = [h'_0, ..., h'_j], \tag{17}$$

$$H_v^G = Att_d \cdot FFN(h'). \tag{18}$$

Similarly, we obtain the final textual graph embedding H_t^G. We first use the single-head attention mechanism to calculate the interaction between the two graph embeddings:

$$H_t^{att} = \text{Softmax}(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{d_k}})\mathcal{V}, \tag{19}$$

where $\mathcal{Q}$,$\mathcal{K}$, and $\mathcal{V}$ are H_t^G, H_v^G, and H_v^G respectively, d_k is the same as the dimension of H_t^G. Then, we apply the gated fusion mechanism to combine the two features:

$$\lambda = \text{Sigmoid}(W_l \cdot H_t^G + W_v \cdot H_t^{att}), \tag{20}$$

$$H_{fuse} = (1 - \lambda) \cdot H_t^G + \lambda \cdot H_t^{att}, \tag{21}$$

where W_l and W_v are learnable parameters. Finally, the fused output H_{fuse} is fed into the Transformer decoder to predict the answer A.

4. Experiment

4.1. Experiment Settings

Datasets: We evaluated our model on four different diagram datasets. The SQA-I dataset was derived from the multimodal ScienceQA benchmark [28], retaining only data that includes visual images and diagrams. The ScienceQA [28] benchmark is a pioneering large-scale dataset for multimodal scientific questions, equipped with comprehensive answer annotations, including detailed lectures and explanations. The TQA-DMC dataset consists of multiple-choice questions with diagrams selected from the TextbookQA [17] . CSDQA [38] is a diagram question-answering dataset from the field of computer science in university courses. AI2D [16] contains diagram questions from the eighth-grade science curriculum. Detailed statistics of four datasets are provided in the supplementary material.

Experimental Settings: In the Chain-of-Guiding Learning module, we apply GPT-3.5 [4] as LLM in the main experiment. The DPT Processing 1 adopts the pre-trained ResNet-101 [13] backbone to learn the $x = 1024$ dimensional representation of each diagram. The DPT Processing 2 uses Fast-SAM [50] to segment the examples in the diagram and access the pre-trained CLIP [34] model to match the instance regions most relevant to the given text. The TGC method in the Answer Inference module adopts open information extraction (OpenIE) systems [1] and Stanford CoreNLP system [32] for triplets extraction and coreference resolution respectively. We adopt T5 [35] as our basic decoder architecture. For a fair comparison, we initialized T5 with the pre-trained Unified-QA [35] checkpoint. We fine-tuned the Answer Inference module for 20 epochs with a learning rate of 5e-5. Our training and evaluation have been done on a single NVIDIA Tesla A800 card with 64GB of RAM GPU.

Model	Dataset			
	SQA-I	TQA-DMC	CSDQA	AI2D
QA-based Models				
MCAN [46]	51.17	27.56	44.41	-
BAN [19]	52.60	27.28	42.32	-
VisualBERT [22]	62.17	41.14	42.86	32.90
RAFR [29]	-	30.47	37.85	-
Finetune-based Models				
Unified-QA [35]	61.38	-	53.39	-
ISAAQ [11]	66.53	51.81	50.70	67.93
WSTQ [31]	-	43.32	48.55	72.05
MoCA [41]	-	<u>53.33</u>	-	-
Prompt-based Models				
GPT-3.5 0-shot [4]	67.28	32.71	45.77	50.10
GPT-3.5 2-shot [4]	67.43 (CoT)	36.20	46.83	53.39
GPT-4 2-shot [33]	71.49 (CoT)	38.47	48.03	58.68
LLaVa-1.5-7B [25]	65.08	35.38	47.31	65.76
LLaVa-1.5-13B [25]	68.45	37.64	48.65	67.77
MM-CoT-large* [49]	<u>73.53</u>	52.51	<u>57.28</u>	<u>75.76</u>
Our Model				
CoG-DQA	**78.85** (↑5.32)	**54.60** (↑1.27)	**68.28** (↑11.00)	**79.14** (↑3.38)

Table 1. Accuracy (%) on test split of four datasets. The superscript $*$ indicates the performance of the model after reproduction. The best and second-best values are marked in bold and underlined respectively. ↑ denotes the accuracy increase (%) compared with previous SOTA model.

Baseline Models: For the four diagram question-answering datasets, we uniformly divide comparison methods into three categories, namely: QA-based models, finetune-based models, and prompt-based models. Among them, QA-based models focus on parsing visual and textual features in question-answering and are mostly trained from initial parameter states, including: [2, 3, 10, 12, 16, 18–20, 22, 27, 30, 38, 45–47, 52]. Finetune-based models pretrain language models through diverse tasks, and then finetune parameters in the specific field of diagram question-answering, including: [11, 31, 35, 41]. Prompt-based models use manually formulated templates to obtain LLM answer feedback for questions, including: [4, 25, 33, 44, 49]. Detailed descriptions of the above baselines are provided in the supplementary material.

4.2. Comparative Analysis

Tab. 1 shows the main results on the test split of four datasets. It should be noted that the MM-CoT model has been modified based on the original model to adapt to the one-step solution that does not have explanation annotations for other datasets except the SQA-I. In order to en-

Model	Dataset			
	SQA-I		CSDQA	
Base Model	73.13	-	56.30	-
w/ CoG-stage1	74.92	1.79 ↑	58.85	2.55 ↑
w/ CoG-stage1+2	76.60	3.47 ↑	59.87	3.57 ↑
w/ CoG-stage1+2+TG	77.00	3.87 ↑	62.49	6.19 ↑
w/ CoG-stage1+2+3+VG	77.14	4.01 ↑	65.36	9.06 ↑
CoG-DQA	78.85	5.72 ↑	68.28	11.98↑

Table 2. Ablation study on SQA-I and CSDQA test split. ↑ denotes the accuracy increase (%) with (w/) the modules on the left. TG and VG respectively correspond to the textual graph and visual graph in the Answer Inference module.

Model	Dataset	Turn 1	Turn 2	Turn 3
GPT-3.5	SQA-I	28	52	69
	CSDQA	42	68	77
	TQA-DMC	10	66	86
	AI2D	19	38	50
GPT-3.0	SQA-I	9	23	44
	CSDQA	43	47	49
text-davinci	SQA-I	98	-	-
	CSDQA	48	66	80

Table 3. Prompt Complete Percentage (%) using different large language models on different datasets.

sure a unified comparison, a one-step comparison method (input questions, candidates, context, and output answers) is used on the above four datasets. It can be seen that the CoG-DQA model has achieved new SOTA performances on all datasets, showing superior accuracy performance and generalization. On the CSDQA dataset, our method outperforms the previous SOTA model and achieves a performance of 68.28%, with an improvement of 11%, which is the most significant improvement. On the TQA-DMC dataset, our method outperforms the previous SOTA model and achieves a performance of 54.60%, with an improvement of 1.27%, which is the least obvious improvement. The reason for the uneven improvement can be: the difficulty of questions and the complexity of reasoning in different datasets vary, and some simpler reasoning patterns are easier to learn by the model. CSDQA contains professional knowledge questions from university computer science and lacks annotation of the problem-solving process, which also highlights the superiority of the CoG-DQA model in solving complex reasoning in small specialized domains.

In addition, according to the experimental results, it can be seen that the overall performance of the model of QA-based is the worst, the model based on finetune is better, and the model based on prompt is generally the best. This is consistent with our hypothesis about the importance of introducing background knowledge for the DQA task. Moreover, GPT and LLaVA series models are difficult to transfer well on the DQA task. This is partly because current large models do not have any interaction with the diagram parsing process. On the other hand, prior knowledge of a general domain is likely to be adaptable to a small specialized domain. These two points are exactly the gaps that the CoG-DQA model hopes to bridge. More fine-grained comparison results on the above datasets are provided in supplementary material.

4.3. Ablation Study

We conduct ablation experiments on each important component in the CoG-DQA model. Tab. 2 shows the functional differences between the variant models and the detailed results on the two datasets. We gradually add each stage in the CoG module, as well as the two graph structures of the Answer Inference (AI) module, and compare the performance with the base model (T5-large framework). It can be seen that both Chain-of-Guiding Learning and dual graph structures work. On the SQA-I dataset, gradually introducing the three steps of the CoG module increases the accuracy by 1.79%, 3.47%, and 4.01% respectively, while on the CSDQA dataset, it is 2.55%, 3.57%, and 9.06%. It should be noted that due to the close relationship between stage 2 and VG, we regard them as an integral component in performing ablation experiments. Overall, the performance gains brought by the CoG module and the AI module are equivalent. According to our analysis, this is because obtaining the correct answer is based not only on the parsing results containing rich background knowledge as a basis, but also on the effective use of these results. Finally, the complete model with all the modules added achieves the best performance, which also proves the irreplaceability of each module.

4.4. Prompt Complete Percentage

To assess the impact of large language models (LLMs) on various datasets beyond traditional question-answering accuracy, we introduced a novel evaluation metric called "prompt complete percentage". Our evaluation method involves dividing prompts to LLMs into three stages. In each stage, if the LLMs provide irrelevant information, deviate from the specified format, or fail to provide an answer, we consider the prompt for that stage as incomplete. We can increase the percentage of incomplete prompts by adding more prompt turns at each stage. As illustrated in Tab. 3, we assessed the complete percentage of different LLMs across multiple datasets. Notably, increasing the number of prompt turns led to a significant improvement in the complete percentage. Among the models evaluated, GPT-3.5 consistently performed well across four datasets. The text-davinci model achieved a nearly 100% complete percentage in just one turn for the SQA-I dataset. However, it's impor-

Model	Format	wo/ CoG	w/ CoG	Dataset
Unified-QA	QCM→A	53.39	56.60	CSDQA
ISAAQ	QCM→A	50.70	52.63	CSDQA
MM-CoT	QCM→A	73.53	75.68	SQA-I
MM-CoT	QCM→LE→A	87.10	88.92	SQA-I

Table 4. Performance of CoG Module as a plug-in under different frameworks and formats on different datasets. Format names: Q = question, C = context, M = multiple options, A = answer, E = explanation, L = lecture.

tant to note that this complete percentage metric does not comprehensively reflect LLM response quality. We recommend considering QA accuracy (analysis in supplementary material) alongside this metric for a more thorough assessment. In practical applications, we suggest using up to three prompt turns, which should suffice for most data scenarios.

4.5. CoG Module Plug-in

In Sec. 3.1 we mentioned that the CoG module can be combined with any downstream inference module as a plug-in without training cost. In this section, we explore the benefits of it as a plug-in. It should be noted that the features processed by the CoG model may not be fully used by a downstream inference model, resulting in different improvements. We access the CoG module on different models with different reasoning forms and on different datasets to observe the gains it brings. As shown in Tab. 4, on the CSDQA dataset, the CoG module as a plug-in has improved both inference models. It should be noted that on Unified-QA, only the knowledge and features in Stage 1 can be added, while on ISAAQ, they can be added in Stages 1 and 2, and in both cases the model performance improved. Under different inference formats on the SQA-I dataset, the performances are also improved after adding the features processed by Stages 1 and 2 of the CoG module. This fully demonstrates that the CoG module proposed in this manuscript has good generalization performance and transferability and the potential to bring gains to any inference model.

4.6. Performance under Chain-of-thought Setting

In the latest research, the chain-of-thought method is an effective solution to stimulate the reasoning ability of LLMs [40]. We performed a corresponding evaluation of the CoG-DQA model under this type of setting on the SQA-I dataset, since only this dataset contains the lecture and explanation annotations that support the setting. Due to the flexibility of the inference module, our model can be adapted to diverse formats of inference. As shown in Tab. 5, the CoG-DQA method surpasses existing baseline models in all types of inference formats. Among them, QCM→LE→A represents the two-step reasoning format, that is, first the lecture and

Model	Format	ACC. (%)
HUMAN	QCM→A	87.50
GPT-3.5	QCM→ALE	75.17
GPT-3.5	QCMLE^g →A	<u>94.13</u>
MM-CoT-large*	QCM→A	73.53
MM-CoT-large*	QCM→LE→A	87.10
GoT-T5-large*	QCM→LE→A	88.60
CoG-DQA	QCM→A	78.85
CoG-DQA	QCM→LE→A	89.32
CoG-DQA	QCMLE^g →A	**96.13**

Table 5. Accuracy (%) comparison of models under the chain-of-thought method on test split of SQA-I dataset. Superscript g indicates the performance using ground truth.

the explanation text related to the question and then the answer. Our model still achieves optimal performance under this setting. In addition, as mentioned in [28], the upper bound of the inference performance of the GPT model is the performance under the QCMLE*→A setting, which is 94.13%. Under the same setting, our model breaks through the bound and achieves an accuracy of 96.13%, which also significantly exceeds Human performance.

5. Conclusion

In this study, we introduce an innovative framework to the diagram question-answering task. Our CoG-DQA framework distinguishes itself from previous research through two key advancements. First, CoG-DQA's approach to guiding diagram parsing tools (DPTs) using LLMs as guiding chains has demonstrated its ability to excel in diagram analysis while considering the crucial contextual background. This strategy bridges domain gaps, allowing for effective inference in specific fields. Second, the combination of LLM prompts and small model fine-tuning ensures robust performance in the complex and diverse DQA landscape. Extensive experimentation across four diverse datasets underscores the remarkable performance of the CoG-DQA model, surpassing other DQA approaches.

Acknowledgement

This work was supported by National Key Research and Development Program of China (2022YFC3303600), National Natural Science Foundation of China (62137002, 62293553, 62293554, 62192781, 62250066, 62176209 and 62106190), "LENOVO-XJTU" Intelligent Industry Joint Laboratory Project, Natural Science Basic Research Program of Shaanxi (2023-JC-YB-593), the Youth Innovation Team of Shaanxi Universities, Shaanxi Undergraduate and Higher Education Teaching Reform Research Program (Program No.23BY195), Project of China Knowledge Centre for Engineering Science and Technology.

References

[1] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 344–354, 2015. 6

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2425–2433. IEEE Computer Society, 2015. 6

[3] Hédi Ben-Younes, Rémi Cadène, Matthieu Cord, and Nicolas Thome. MUTAN: multimodal tucker fusion for visual question answering. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2631–2639. IEEE Computer Society, 2017. 6

[4] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020. 6

[5] Tao Chen, Guo-Sen Xie, Yazhou Yao, Qiong Wang, Fumin Shen, Zhenmin Tang, and Jian Zhang. Semantically meaningful class prototype learning for one-shot image segmentation. *IEEE Transactions on Multimedia*, 24:968–980, 2021. 1

[6] Ping-Jung Duh, Yu-Cheng Sung, Liang-Yu Fan Chiang, Yung-Ju Chang, and Kuan-Wen Chen. V-eye: A vision-based navigation system for the visually impaired. *IEEE Transactions on Multimedia*, 23:1567–1580, 2020. 1

[7] Ronald W. Ferguson and Kenneth D. Forbus. Telling juxtapositions: Using repetition and alignable difference in diagram understanding. *Advances in analogy research*, 5(1): 109–117, 1998. 2

[8] Ronald W. Ferguson and Kenneth D. Forbus. Georep: A flexible tool for spatial representation of line drawings. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA*, pages 510–516. AAAI Press / The MIT Press, 2000. 2

[9] Robert P. Futrelle, Mingyan Shao, Chris Cieslik, and Andrea Elaina Grimes. Extraction, layout analysis and classification of diagrams in PDF documents. In *7th International Conference on Document Analysis and Recognition (ICDAR 2003), 2-Volume Set, 3-6 August 2003, Edinburgh, Scotland, UK*, pages 1007–1014, 2003. 2

[10] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6639–6648, 2019. 6

[11] José Manuél Gómez-Pérez and Raúl Ortega. ISAAQ - mastering textbook questions with pre-trained transformers and bottom-up and top-down attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5469–5479. Association for Computational Linguistics, 2020. 2, 3, 6

[12] Monica Haurilet, Alina Roitberg, and Rainer Stiefelhagen. It's not about the journey; it's about the destination: Following soft paths under question-guidance for visual reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1930–1939. Computer Vision Foundation / IEEE, 2019. 6

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 3, 6

[14] Xin Hong, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. Transformation driven visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6903–6912, 2021. 1

[15] Zan-Xia Jin, Heran Wu, Chun Yang, Fang Zhou, Jingyan Qin, Lei Xiao, and Xu-Cheng Yin. Ruart: A novel text-centered solution for text-based visual question answering. *IEEE Transactions on Multimedia*, 2021. 1

[16] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Min Joon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 235–251. Springer, 2016. 2, 6

[17] Aniruddha Kembhavi, Min Joon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5376–5384. IEEE Computer Society, 2017. 1, 6

[18] Daesik Kim, Youngjoon Yoo, Jeesoo Kim, Sangkuk Lee, and Nojun Kwak. Dynamic graph generation network: Generating relational knowledge from diagrams. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4167–4175, 2018. 5, 6

[19] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1571–1581, 2018. 6

[20] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021. 6

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer White-

head, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 4

[22] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. What does bert with vision look at? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5265–5275, 2020. 6

[23] Liang Lin, Pengxiang Yan, Xiaoqian Xu, Sibei Yang, Kun Zeng, and Guanbin Li. Structured attention network for referring image segmentation. *IEEE Transactions on Multimedia*, 24:1922–1932, 2021. 1

[24] Fuxiao Liu, Tianrui Guan, Zongxia Li, Lichang Chen, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v (ision), llava-1.5, and other multi-modality models. *arXiv preprint arXiv:2310.14566*, 2023. 2

[25] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 2, 6

[26] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. 3, 5

[27] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021. 6

[28] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022. 2, 3, 6, 8

[29] Jie Ma, Jun Liu, Yaxian Wang, Junjun Li, and Tongliang Liu. Relation-aware fine-grained reasoning network for textbook question answering. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 6

[30] Jie Ma, Qi Chai, Jingyue Huang, Jun Liu, Yang You, and Qinghua Zheng. Weakly supervised learning for textbook question answering. *IEEE Trans. Image Process.*, 31:7378–7388, 2022. 3, 6

[31] Jie Ma, Qi Chai, Jingyue Huang, Jun Liu, Yang You, and Qinghua Zheng. Weakly supervised learning for textbook question answering. *IEEE Transactions on Image Processing*, 31:7378–7388, 2022. 2, 6

[32] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60, 2014. 6

[33] R OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2023. 6

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pages 8748–8763, 2021. 4, 6

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67, 2020. 6

[36] Min Joon Seo, Hannaneh Hajishirzi, Ali Farhadi, and Oren Etzioni. Diagram understanding in geometry questions. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 2831–2838, 2014. 2

[37] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023. 4

[38] Shaowei Wang, Lingling Zhang, Xuan Luo, Yi Yang, Xin Hu, Tao Qin, and Jun Liu. Computer science diagram understanding with topology parsing. *ACM Trans. Knowl. Discov. Data*, 16(6):114:1–114:20, 2022. 6

[39] Yasuhiko Watanabe and Makoto Nagao. Diagram understanding using integration of layout information and textual information. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference*, pages 1374–1380. Morgan Kaufmann Publishers / ACL, 1998. 2

[40] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022. 8

[41] Fangzhi Xu, Qika Lin, Jun Liu, Lingling Zhang, Tianzhe Zhao, Qi Chai, Yudai Pan, Yi Huang, and Qianying Wang. Moca: Incorporating domain pretraining and cross attention for textbook question answering. *Pattern Recognition*, 140: 109588, 2023. 2, 3, 6

[42] Yan Yang, Jun Yu, Jian Zhang, Weidong Han, Hanliang Jiang, and Qingming Huang. Joint embedding of deep visual and semantic features for medical image report generation. *IEEE Transactions on Multimedia*, 2021. 1

[43] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9, 2023. 2

[44] Yao Yao, Zuchao Li, and Hai Zhao. Beyond chain-of-thought, effective graph-of-thought reasoning in large language models. *arXiv preprint arXiv:2305.16582*, 2023. 2, 3, 5, 6

[45] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning

for visual question answering. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1839–1848. IEEE Computer Society, 2017. 6

[46] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6281–6290. Computer Vision Foundation / IEEE, 2019. 6

[47] Zhaoquan Yuan, Xiao Peng, Xiao Wu, and Changsheng Xu. Hierarchical multi-task learning for diagram question answering with multi-modal transformer. In *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*, pages 1313–1321. ACM, 2021. 6

[48] Ming-Liang Zhang, Fei Yin, Yi-Han Hao, and Cheng-Lin Liu. Plane geometry diagram parsing. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 1636–1643. ijcai.org, 2022. 2

[49] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023. 2, 3, 6

[50] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *CoRR*, abs/2306.12156, 2023. 6

[51] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023. 4

[52] Chen Zheng, Quan Guo, and Parisa Kordjamshidi. Cross-modality relevance for reasoning on language and vision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7642–7651. Association for Computational Linguistics, 2020. 6