Detection-Based Intermediate Supervision For Visual Question Answering

Yuhang Liu[*1,2,4], **Daowan Peng**[*1,2], **Wei Wei**[†1,2], **Yuanyuan Fu**[2,3], **Wenfeng Xie**[2,3], **Dangyang Chen**[2,3]

[1] CCIIP Lab, School of Computer Science and Technology, Huazhong University of Science and Technology
[2] Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL)
[3] Ping An Property & Casualty Insurance Company of China, Ltd.
[4] ByteDance Inc.
liuyuhang.ysy@bytedance.com, {pengdw, weiw}@hust.edu.cn, fuyuanyuan83@gmail.com, {xiewenfeng801,
chendangyang273}@pingan.com.cn

Abstract

Recently, neural module networks (NMNs) have yielded ongoing success in answering compositional visual questions, especially those involving multi-hop visual and logical reasoning. NMNs decompose the complex question into several sub-tasks using instance-modules from the reasoning paths of that question and then exploit intermediate supervisions to guide answer prediction, thereby improving inference interpretability. However, their performance may be hindered due to sketchy modeling of intermediate supervisions. For instance, (1) a prior assumption that each instance-module refers to only one grounded object yet overlooks other potentially associated grounded objects, impeding full cross-modal alignment learning; (2) IoU-based intermediate supervisions may introduce noise signals as the bounding box overlap issue might guide the model's focus towards irrelevant objects. To address these issues, a novel method, **D**etection-based **I**ntermediate **S**upervision (DIS), is proposed, which adopts a generative detection framework to facilitate multiple grounding supervisions via sequence generation. As such, DIS offers more comprehensive and accurate intermediate supervisions, thereby boosting answer prediction performance. Furthermore, by considering intermediate results, DIS enhances the consistency in answering compositional questions and their sub-questions. Extensive experiments demonstrate the superiority of our proposed DIS, showcasing both improved accuracy and state-of-the-art reasoning consistency compared to prior approaches.

Introduction

Compositional visual question answering (VQA) (Hudson and Manning 2019b; Chen et al. 2021; Jing et al. 2022b) has been an emerging research topic in multimodal domain, and received increasing attention from both computer vision and the language processing communities. Specifically, given input images and questions, it is required to generate answers for the questions according the content of images. Generally, compositional questions involve multiple visual entities or concepts (*i.e.,* objects, attributes, and relations), and the models demand a rich set of abilities (semantic understanding, object detections, visual/logical reasoning) to get the right answer. One of the challenges in compositional visual question answering lies in modeling the reasoning process. From this perspective, VQA models can be divided into two categories, *i.e.,* holistic and modular. Holistic models (Hu et al. 2019; Yu et al. 2019; Yang et al. 2020) generate answers for all types of questions through a unified multimodal fusion model, and the reasoning process is implicitly performed during the encoding and fusion stages. Despite the effectiveness, holistic models cannot reflect the intermediate reasoning process. On the contrary, modular model, *i.e.,* neural module network (NMN) (Andreas et al. 2016; Hu et al. 2018; Hudson and Manning 2018; Chen et al. 2021), has been a mainstream approach due to its explicit reasoning procedure and interpretable characteristics. Specifically, NMN parses questions into predefined reasoning modules and composes these modules into an executive program, thus deconstructing complex questions into several easy-to-solve problems. During the process of multi-hop visual/logical reasoning and answering generation, the intermediate status and results can be explicitly reflected from each module. Moreover, in order to restrict the reasoning process of NMN models, extra intermediate supervisions (Chen et al. 2021; Zhao et al. 2021) are proposed to improve the answer prediction performance, which restricts models to focus on pivotal objects via Intersection over Union (IoU) constraint between predicted bounding boxes and ground truth ones.

Despite the significant improvement in the accuracy metric, there are several shortcomings in the IoU-based intermediate supervisions of previous methods. For example, MMN (Chen et al. 2021) directly exploits the ground-truth scene graph paths corresponding to questions for intermediate result generation, which ignores other potentially correct intermediate objects and hinders the model to fully learn cross-modal alignment and reasoning process. As Figure 1 (a) shows, MMN adopts the leftmost *pillow* to supervise the first step result (*i.e., Select(pillow)*), while our proposed DIS method takes into account all possible correct results. Such comprehensive supervisions alleviate the missing-in-the-middle problem, thus facilitating the model to generate correct answers. Besides, previous IoU-based intermediate supervisions are possible to introduce noise signals due to the bounding-box overlap problem, which induces the model to focus on irrelevant objects. This is exemplified in Figure 1 (b). In the first reasoning step, the model is required to focus

---

Q: Is the little dog to the right or to the left of the pillow on the left?
A: right

(a) Illustrations of grounding objects

Q: What kind of furniture is the napkin on?
A: desk
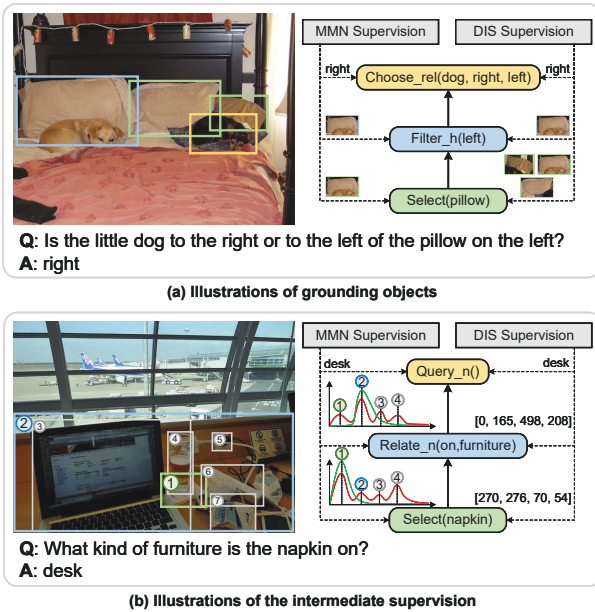
(b) Illustrations of the intermediate supervision

Figure 1: Illustrations of the differences in intermediate supervisions between MMN and DIS: (a) Number of grounded objects, and (b) IoU-based and detection-based supervisions (green and red lines indicate the ground-truth and IoU-based probability distributions over objects).

on the *napkin* (*i.e.,* probability distribution corresponding to the green lines). However, due to the overlap between *napkin* and other objects (*e.g., table, laptop, cup, etc.*), the IoU-based supervisions prompt the model to focus on irrelevant areas (*i.e.,* probability distribution corresponding to the red lines), thus leading to grounding ambiguity and generating wrong answers.

To this end, we propose a novel Detection-based Intermediate Supervision (DIS) method to resolve aforementioned issues. Specifically, to obtain comprehensive intermediate results, executive programs are parsed from the questions, and then step-by-step inference is performed on the scene graphs, which generates complete intermediate results (illustrated in Figure 1 (a)). Afterwards, a generative framework is proposed to supervise VQA model using the intermediate results, which transforms the intermediate supervisions into sequences, and constrains model states via sequence generation (illustrated in Figure 1 (b)). Compared to previous methods, our proposed DIS provides more comprehensive supervision signals for the reasoning process, and exploits a unified generative framework to constrain the intermediate states, thereby improving the answer prediction performance. Moreover, due to the consideration of intermediate results, the answering consistency among compositional questions and their sub-questions is significantly improved. In summary, the main contributions are the following,

- We introduce Detection-based Intermediate Supervision (DIS), a novel method that provides more comprehensive intermediate supervisions via a unified generative framework. To the best of our knowledge, this is the first at-

tempt in the usage of a generative framework for intermediate supervisions in visual question answering.

- We propose a scene graph inference framework, which step-by-step executes programs parsed from questions on scene graphs to obtain intermediate results. The supervision signals are further constructed by converting the results into a unified sequential form.

- We conduct extensive experiments to evaluate the effectiveness of our proposed DIS algorithm, in which our method achieves competitive answer prediction performance (61.31% *vs.* 60.83%), and superior reasoning consistency (73.11% *vs.* 71.47%, 64.20% *vs.* 61.94%, and 55.28% *vs.* 52.80%) compared to previous methods.

Related Work

Visual Question Answering

Compositional visual question answering task is defined to generate answers for given compositional questions based on the image content. Generally, compositional questions consist of multiple visual concepts (*i.e.,* objects, attributes, and relations), and require VQA models to perform multi-hop reasoning to get the right answers. Recently, several attempts have been made to facilitate visual and logical reasoning, and these methods can be divided into two categories: holistic and modular. Holistic methods (Anderson et al. 2018; Kim, Jun, and Zhang 2018; Tan and Bansal 2019; Hu et al. 2019; Yu et al. 2019; Yang et al. 2020) exploit a unified multimodal fusion model to solve all types of question, and achieve implicit visual/logical reasoning through graph structures (Li et al. 2019; Hu et al. 2019) and relational attention mechanisms(Li et al. 2019; Hu et al. 2019; Yang et al. 2020). With the help of scene graph structure, images are represented as graphical webs, containing information about objects, attributes and relationships among interconnected objects, which can be used for visual reasoning via graph traversal. For example, NSM (Hudson and Manning 2019a) performs sequential reasoning over the probabilistic scene graph of the image, and achieves multi-hop inference by shifting probability distributions. RPR (Jing et al. 2022a) casts visual reasoning as a path routing task, and adopts reinforcement learning to explore the inference path.

Despite the overwhelming successes achieved, holistic models process all types of questions via a unified model, which ignores the reasoning structure implicit in the question. Therefore, modular methods (Andreas et al. 2016; Hu et al. 2018; Hudson and Manning 2018; Chen et al. 2021; Zhao et al. 2021) are proposed to make up for the abovementioned issues. Specifically, modular methods parse the question into a structured tree that reflects the reasoning process, and construct the question-specific model using predefined modules. Due to the explicit reasoning structure, such methods have strong interpretability and controllability. In addition, extra intermediate supervisions can be provided to constrain models to reason along prescribed directions, *e.g.,* IoU-based Kullback-Leible (KL) divergence (Chen et al. 2021; Zhao et al. 2021), thereby improving answer prediction performance. However, such IoU-based supervisions suffer from two issues, *i.e.,* ignorance of multiple
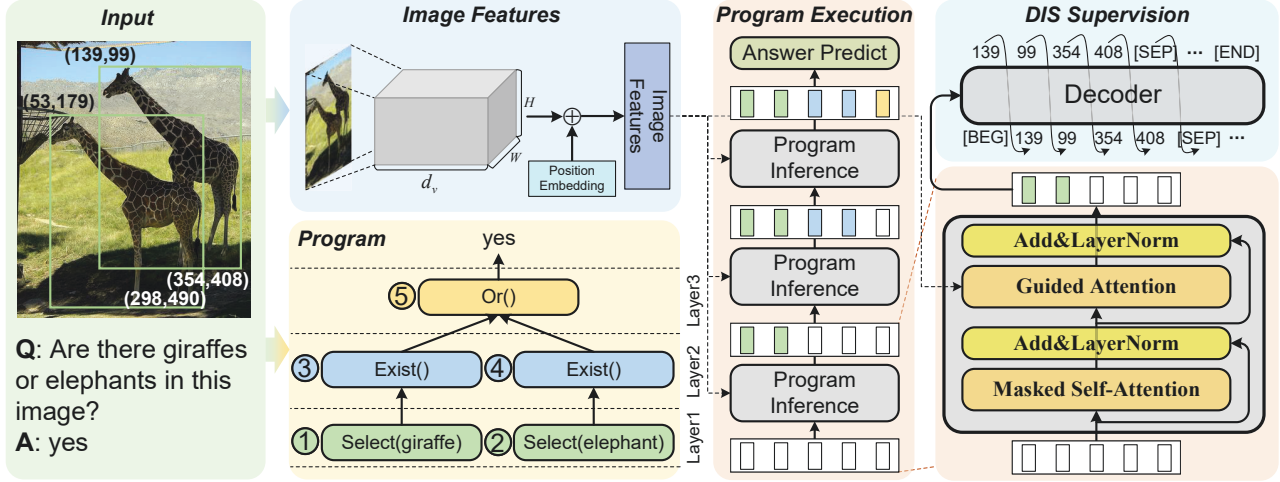
Figure 2: The framework of our proposed Detection-based Intermediate Supervision (DIS) method. Questions are parsed into programs, and step-by-step program execution is performed to generate answers. The intermediate states are supervised by our DIS algorithm.

grounded objects and grounding ambiguity, underutilizing the intermediate supervisions for model optimization.

Object Detection

There has been a tremendous amount of work in object detection tasks, which require extracting objects from images. Traditional object detection algorithms introduce explicit prior knowledge via producing a set of proposals (Girshick 2015; Ren et al. 2017), anchors (Redmon et al. 2016), or window centers (Tian et al. 2019), and then perform non-maximum suppression (Bodla et al. 2017) to remove duplicate predictions. To avoid complex processing procedures, DETR (Carion et al. 2020) exploits Transformer-based encoder-decoder framework (Vaswani et al. 2017) for object detection, which learns a set of "object queries" to directly generate bounding boxes and object labels. All of these detectors require extra modules for bounding box regression and label prediction to obtain final predictions. To further avoid such complexities, Pix2Seq (Chen et al. 2022) directly predicts the raw pixel coordinates through an encoder-decoder network, which achieves competitive performance while simplifying the detection framework. Inspired from Pix2Seq and language modeling (Brown et al. 2020), we propose a detection-based supervision framework, which converts the intermediate results into a unified sequential form that consists of pixels and tokens, thereby providing more comprehensive supervision signals.

Methodology

We propose Detection-based Intermediate Supervision (DIS) methods to facilitate the constraint of intermediate reasoning state, thereby improving the answer prediction performance. The overall framework is depicted in Figure 2. Specifically, image features are first extracted via convolutional neural networks (CNN). Then, the question is parsed into program tree, followed by the program execution network to get the final answer. Finally, we introduce detection-

based intermediate supervision framework to enhance reasoning ability of the VQA model.

Image Features Extraction

Image feature extraction is based on the pre-trained Faster R-CNN model (Ren et al. 2017; Jiang et al. 2020). In contrast to previous methods (Anderson et al. 2018; Hu et al. 2019) that adopt bottom-up features, we adopt the feature maps output from $C5$ layer of Faster R-CNN (Jiang et al. 2020) for image representation. Specifically, given an image I, the pre-trained CNN backbone of Faster R-CNN is utilized to extract the feature map $\mathbf{V} \in \mathbb{R}^{HW \times d_v}$, where H, W indicate height and width of the feature map, respectively. d_v denotes the feature dimension.

To endow the image features with visual contexts and cross-modal textual information, we follow MCAN (Yu et al. 2019) method, which adopts Transformer block to encode the question and image. Specifically, given a question Q of length T, which is embedded into latent space $\mathbf{E} \in \mathbb{R}^{T \times d_h}$, a two-layer Transformer is adopted to encode the questions as follows:

$$\begin{aligned} \hat{\mathbf{E}} &= \text{LN}(\mathbf{E} + \text{SA}(\mathbf{E})) \\ \tilde{\mathbf{E}} &= \text{LN}(\hat{\mathbf{E}} + \text{FFN}(\hat{\mathbf{E}})), \end{aligned} \quad (1)$$

where $\text{SA}, \text{LN}, \text{FFN}$ denote self-attention, layer normalization and feed forward network, respectively. Afterwards, the image feature $\mathbf{V}$ is enriched via a two-layer Transformer using visual contexts and question semantics $\tilde{\mathbf{E}}$ as follows:

$$\begin{aligned} \mathbf{V}' &= \text{FC}(\mathbf{V} + \text{PosEmb}) \\ \bar{\mathbf{V}} &= \text{LN}(\mathbf{V}' + \text{SA}(\mathbf{V}')) \\ \hat{\mathbf{V}} &= \text{LN}(\bar{\mathbf{V}} + \text{GA}(\bar{\mathbf{V}}, \hat{\mathbf{E}})) \\ \tilde{\mathbf{V}} &= \text{LN}(\hat{\mathbf{V}} + \text{FFN}(\hat{\mathbf{V}})), \end{aligned} \quad (2)$$

where PosEmb indicates position embedding. FC denotes fully-connected layer, converting feature dimension from d_v
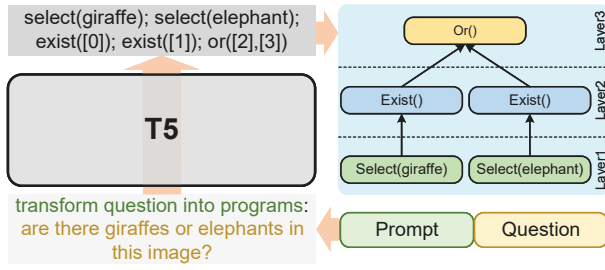
Figure 3: Illustration of the program generation based on encoder-decoder network.

to d_h. GA denotes guided attention, which exploits question semantics to enhance the relevant visual features. The resulting image representation $\tilde{\mathbf{V}}$ can be used for program execution to get the final answer.

Program Generation

Program generation aims to parse questions into program trees that reflect the reasoning procedures inferred by questions, and the program trees can be further used for model construction. We follow MMN (Chen et al. 2021) to generate programs from questions. Specifically, the nodes of the program tree are formalized as "Function(Arg1,...ArgN)", where "Function" can be categorized into 10 different abstract types (*e.g.,* select, relate, exist, or, *etc.*), and each abstract type is further subdivided into more subtypes (*e.g.,* relate: relate_attr, relate_name, relate_inv_name, *etc.*), which take a variable number of arguments as inputs.

Based on the abovementioned program types, the complete program tree can be viewed as the sequence of functional nodes, and generated from an encoder-decoder network, *e.g.,* T5 (Raffel et al. 2020). As illustrated in Figure 3, a prompt (*i.e.,* "transform question into programs:") is added in front of the question, and fed to T5 model for program generation. The output sequence consists of interdependent functions, where "[N]" denotes the dependencies. Afterwards, a structured program tree with L layers is constructed from the sequence according to the dependencies, which can be used to guide step-by-step program execution.

Program Execution

Given image features $\tilde{\mathbf{V}}$ and the L-layer program tree, program execution performs step-by-step inference from Layer-1 to Layer-L based on image features to get the answer, and all layers share model parameters, making it parameter-efficient and scalable to any number of layers. Specifically, suppose the program tree contains N nodes, and each node corresponds to a program text (*e.g.,* "Select(giraffe)", "Select(elephant)", *etc.*), a state matrix $\mathbf{S} \in \mathbb{R}^{N \times d_h}$ is initialized with the program semantics as follows:

$$\begin{aligned} \{\mathbf{e}_0^i, \mathbf{e}_1^i, ..., \mathbf{e}_k^i\} &= \text{GloVe}(prog_i) \\ \mathbf{S}_i &= \text{FC}(\text{Concat}(\{\mathbf{e}_0^i, \mathbf{e}_1^i, ..., \mathbf{e}_l^i\})), \end{aligned} \quad (3)$$

where $prog_i$ denotes the program text of i-th node, and is truncated or padded to fixed length k. GloVe denotes the

GloVe embedding layer (Pennington, Socher, and Manning 2014). In the process of step-by-step inference, the matrix $\mathbf{S}$ implicitly contains the intermediate reasoning states and results, and can be used to decode outputs for our proposed DIS algorithm.

Denote the input state of the l-th layer as $\mathbf{S}^{l-1}$, we exploit Transformer framework to obtain the output state $\mathbf{S}^l$. Specifically, a masked self-attention layer is firstly exploited to gather dependencies from the state of last layer, and then a guided attention layer is utilized to find visual clues from image features $\tilde{\mathbf{V}}$, formulated as follows:

$$\begin{aligned} \hat{\mathbf{S}}^{l-1} &= \text{LN}(\mathbf{S}^{l-1} + \text{MaskSA}(\mathbf{S}^{l-1}, \mathbf{M}^l)) \\ \mathbf{S}^l &= \text{LN}(\hat{\mathbf{S}}^{l-1} + \text{GA}(\hat{\mathbf{S}}^{l-1}, \tilde{\mathbf{V}})), \end{aligned} \quad (4)$$

where MaskSA, GA denote mask self-attention and guided attention layer, respectively. MaskSA uses weight matrix $\mathbf{M}^l$ to mask non-dependent nodes, formulated as follows:

$$\text{MaskSA}(\mathbf{S}, \mathbf{M}) = \text{Softmax}(\frac{(\mathbf{Q}^S)(\mathbf{K}^S)^T}{\sqrt{d_h}} + \mathbf{M})\mathbf{V}^S, \quad (5)$$

where $\mathbf{M} \in \mathbb{R}^{N \times N}$ denotes the mask matrix. $\mathbf{M}_{ij} = 0$ if and only if i-th node is the parent node of j-th node, and otherwise $\mathbf{M}_{ij} = -\infty$. $\mathbf{Q}^S, \mathbf{K}^S, \mathbf{V}^S$ are derived from $\mathbf{S}$ via three fully-connected layers, formulated as follows:

$$\begin{aligned} \mathbf{Q}^S &= \text{FC}^Q(\mathbf{S}) \\ \mathbf{K}^S &= \text{FC}^K(\mathbf{S}) \\ \mathbf{V}^S &= \text{FC}^V(\mathbf{S}). \end{aligned} \quad (6)$$

Different from SA that gathers information from itself, guided attention gathers features from other source (*e.g.,* visual clues), formulated as follows:

$$\text{GA}(\mathbf{S}, \tilde{\mathbf{V}}) = (\frac{(\mathbf{Q}^S)(\mathbf{K}^{\tilde{V}})^T}{\sqrt{d_h}})\mathbf{V}^{\tilde{V}}, \quad (7)$$

where $\mathbf{K}^{\tilde{V}}, \mathbf{V}^{\tilde{V}}$ are derived from $\tilde{\mathbf{V}}$ similar to Equation 6.

After L iterations of program inference, the final state $\mathbf{S}_{N-1}^L$ is used to predict the answer via a multi-layer perception (MLP) layer:

$$\begin{aligned} \mathbf{s} &= \text{MLP}(\mathbf{S}_{N-1}^L) \\ p(a|I, Q; \Theta) &= \text{Softmax}(\mathbf{s}), \end{aligned} \quad (8)$$

where $\mathbf{s} \in \mathbb{R}^{|\mathcal{A}|}$ denotes the predicted scores of the answers in $\mathcal{A}$, and the answer with the highest score is chosen as the final answer. Θ denotes the model parameters. Finally, cross-entropy loss is used to optimize the model, formulated as follows:

$$\mathcal{L}^{VQA} = -\mathbb{E}_{\mathcal{D}}[\log(p(a = a^{gt}|I, Q; \Theta))], \quad (9)$$

where $\mathcal{D}, a^{gt}$ denotes the VQA dataset and ground truth answer, respectively.
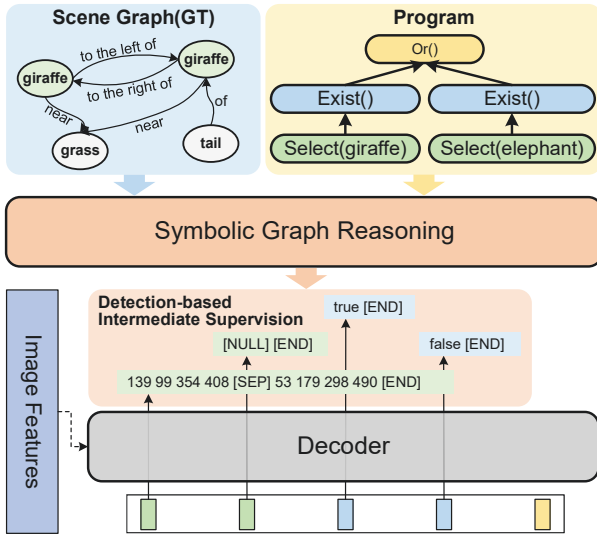
Figure 4: Illustrations of Detection-based Intermediate Supervision (DIS) algorithm. The intermediate results are generated via symbolic graph reasoning based on the ground-truth scene graph and program tree, and transformed into sequences to supervise the intermediate states.

Intermediate Supervision

On top of the above-mentioned program execution, intermediate supervisions are proposed to constrain the reasoning process, and improve the answer prediction performance (Chen et al. 2021; Zhao et al. 2021). However, previous methods calculate probability distributions based on IoU between object bounding boxes and ground-truth ones, which easily induces the model to focus on irrelevant objects (ref to Figure 1 (b)). To this end, we propose detection-based intermediate supervision (DIS) algorithm, which formulates the intermediate supervisions into a unified sequence form, thereby endowing the model with the abilities of exploiting diverse supervision types (*e.g.,* bounding boxes, logical words (true/false), text). As shown in Figure 4, DIS algorithm consists of two steps: (1) *symbolic graph reasoning* designs manual rules to execute the program tree on the ground-truth scene graph[1], resulting in intermediate results (*e.g., Objects*, *True/False*, *Answer*), and (2) *intermediate result decoder* decodes the intermediate outputs, which is optimized using auto-regression loss.

Symbolic Graph Reasoning   To obtain the intermediate supervisions, we follow MMN (Chen et al. 2021), which executes the *Program* on the ground truth *Scene Graph* (as illustrated at the top of Figure 4). Specifically, we manually design the symbolic reasoning rules for each function type (*e.g.,* Select, Relate, Verify, *etc.*). For example, *Select(x)* is defined to *select the nodes corresponding to x*, and *Relate(x, to the left of)* is defined to *find the nodes to the left of x*.

---

[1]GQA (Hudson and Manning 2019b) dataset provides ground-truth scene graph only for *train* and *val* splits. Therefore, DIS is used to optimize the model only in the training phase, and removed during testing phase.

With the *Program* and *Scene Graph*, the intermediate supervisions are inferred step by step, as illustrated at the bottom of Figure 4. Generally, the supervisions are categorized into three types: *Objects* with bounding boxes, *True/False*, and *Answers*. Regarding *Objects*, we follow Pix2Seq (Chen et al. 2022) to quantize coordinates into bins, which can be regarded as discrete labels. If multiple objects exist, their bounding boxes are randomly shuffled and concatenated to form the result sequence. For *True/False* and *Answer*, we directly use the textual tokens to form the sequence. Additionally, several special tokens are added to sequence for further generation, *e.g.,* [BEG], [SEP], [END], *etc.*

Intermediate Result Decoder   With the help of intermediate results, VQA model can be optimized using these supervisions. Specifically, a two-layer Transformer is proposed to decode the intermediate outputs from the states $\tilde{\mathbf{S}}^L$. Firstly, [BEG] token is initialized with the state $\tilde{\mathbf{S}}_i^L$, and prompts the decoder to generate output sequence. In the training phase, we use teacher-forcing and auto-regression loss to optimize the model, formulated as follows:

$$\{y_0, y_1, ..., y_{o-1}\} = \text{Decoder}(\tilde{\mathbf{S}}_i^L, \tilde{\mathbf{V}})$$

$$\mathcal{L}^{DIS} = -\frac{1}{o} \sum_{i=0}^{o-1} \log(p(y_i|y_0, ..., y_{i-1}, \tilde{\mathbf{S}}_i^L, \tilde{\mathbf{V}})),$$

(10)

where o denotes the length of the result sequence. $\tilde{\mathbf{S}}_i^L \in \mathbb{R}^{d_h}$ denotes the intermediate state from the i-th program node.

Model Optimization   In the training phase, the answer prediction loss (Equation 9) and detection-based intermediate supervision loss (Equation 10) are combined to optimize the model:

$$\mathcal{L} = \mathcal{L}^{VQA} + \alpha \mathcal{L}^{DIS},$$

(11)

where α denotes the loss weight of DIS. In the testing phase, DIS module can be removed because only the final answer needs to be predicted.

Experiments

Datasets

To evaluate the answer prediction performance and answering consistency, the reported results in the following sections are evaluated on the widely used GQA (Hudson and Manning 2019b) dataset, and its variant GQA-Sub (Jing et al. 2022b). **GQA** (Hudson and Manning 2019b) is a compositional visual question-answer dataset, which features compositional questions over the real-world images. It is designed to provide accurate indications of visual understanding capacities and mitigate the language priors that exist widely in previous VQA datasets (Agrawal et al. 2017). **GQA-Sub** (Jing et al. 2022b) is derived from the well-organized GQA dataset, and creates sub-questions for *train* and *val* splits, thereby enabling quantitative evaluation of reasoning consistency. More information about GQA and GQA-Sub, as well as their respective evaluation metrics, can be found in Appendix-A.

Model	Required Inputs	Binary	Open	Accuracy
BUTD	V+L	66.64	34.83	49.74
MAC	V+L	71.23	38.91	54.06
GRN	V+L	74.93	41.24	57.04
LCGN	V+L	73.77	42.33	57.07
BAN	V+L	76.00	40.41	57.10
PVR	V+L+Program	74.58	42.10	57.33
LXMERT	V+L+Pre-training	77.16	**45.47**	60.33
MCAN	V+L	75.87	42.15	57.96
RPR	V+L	-	-	59.43
RCVQA	V+L+DataAug	-	-	59.58
MMN	V+L+Program	78.90	44.89	60.83
DIS	V+L+Program	**79.36**	45.36	**61.31**

Table 1: Comparison of DIS single model with the state-of-the-art methods on the blind test2019 leaderboard of GQA.

Implementation Details

Program Generation. T5-base[2] is utilized for text transformation, where source and target texts are limited to a maximum length of 40 and 100, respectively. We exploit AdamW optimizer with learning of $1e^{-4}$ and batch size of 32 to fine-tune T5 for 400k steps.

Visual Question Answering. Following the settings from MMN (Chen et al. 2021), the questions are truncated or padding to the fixed length of 32. The number of nodes in each program tree is limited to 9, and the maximum length k of each program is set to 8. See the Appendix-B for more implementation details. The code is available at https://github.com/CCIIPLab/DIS.

Baselines. Our model is compared with various state-of-the-art approaches excerpted from MMN, including BUTD (Anderson et al. 2018), MAC (Hudson and Manning 2018), GRN (Guo, Xu, and Tao 2019), LCGN (Hu et al. 2019), BAN(Kim, Jun, and Zhang 2018), PVR (Li, Wang, and Zhu 2019), LXMERT (Tan and Bansal 2019), MCAN (Yu et al. 2019), MMN (Chen et al. 2021), RPR (Jing et al. 2022a), and RCVQA (Jing et al. 2022b). We did not compare with the NSM (Hudson and Manning 2019a) because it utilizes a well-tuned external scene graph generation model. More information about baselines are provided in Appendix-C.

Experimental Results

In the experiments, we primarily assessed the performance of our model on answering compositional questions as well as its performance in reasoning consistency. The corresponding experimental results are presented in Table 1 and Table 2, respectively. The online test results of the state-of-the-art models and our proposed DIS method on the GQA dataset are shown in Table 1, and these results also reflect the performance of all models on compositional questions. The required inputs represent the information necessary for the model to predict the answers, where V and L indicate vision and language, respectively, while DataAug represents data augmentation. As shown, our proposed DIS achieves the

Methods	Acc	Acc(Sub)	RC(1)	RC(2)	RC(3)
Language-only	43.86	41.16	31.61	18.81	7.31
Visual-only	56.63	53.97	46.63	28.16	16.26
MAC	62.08	62.63	56.10	41.67	33.96
MAC + DA	61.04	71.42	65.78	54.09	43.14
LCGN	64.16	63.74	57.37	44.32	35.09
LCGN + DA	64.14	73.46	68.93	58.94	50.05
MMN	65.05	64.46	58.79	43.98	33.96
MMN + DA	65.65	74.60	69.59	57.98	48.17
RCVQA + DA	66.26	76.02	71.47	61.94	52.80
DIS	69.71	70.31	64.98	53.55	48.86
DIS + DA	**69.81**	**77.91**	**73.11**	**64.20**	**55.28**

Table 2: Results of our proposed DIS method and the state-of-the-arts on GQA-Sub. The "DA" denotes the data augmentation.

best performance on *Binary*, *Overall* and the second best on *Open* among the methods listed in Table 1. Specifically, with basically the same inputs and settings as MMN, DIS method outperforms MMN by a margin of +0.46% and +0.47% for *Binary* and *Open* questions, respectively.

Also, we evaluate the performance of the proposed DIS in terms of reasoning consistency. The results regarding accuracy (*i.e.,* Acc and Acc(Sub)) and reasoning consistency (*i.e.,* RC(k), refer to Appendix-A for details) of our proposed DIS and state-of-the-art methods are presented in Table 2. Acc and Acc(Sub) denote the accuracies on *val* and *val-sub* splits, respectively. DA indicates the usage of augmented sub-questions for model training. As Table 2 shows, our proposed DIS surpasses the other state-of-the-art methods on both accuracy and reasoning consistency metrics. Specifically, without data augmentation of sub-questions, DIS outperforms MMN by a large margin of 4.66% and 5.85% on *val* and *val-sub* splits, respectively. The reasoning consistency is also significantly improved by using our DIS algorithm, *i.e.,* a margin of **6.19%**, **9.57%**, and **14.9%** on RC(1), RC(2), and RC(3), respectively. Such superiority stems from the comprehensive yet noise-free supervisions of intermediate results provided by DIS, significantly enhancing the ability of the model to answer compositional questions and corresponding sub-questions. When we trained the DIS with sub-questions in *train-sub* as a form of data augmentation, our proposed DIS even outperforms the best-performing RCVQA model that is tailored for enhancing reasoning consistency through the incorporation of a consistency constraint loss. Specifically, DIS surpasses RCVQA by a significant margin of 3.55% on Acc and over 1.64%, 2.26% and 2.48% on the three reasoning consistency metrics, respectively, which indicates the effectiveness of our proposed method.

Ablation Studies

In this section, a series of ablations are conducted on GQA dataset to investigate the effectiveness of our proposed method. All the models are trained on *train+val* split, and evaluated on *testdev* split. The experimental settings are kept consistent throughout ablation studies.
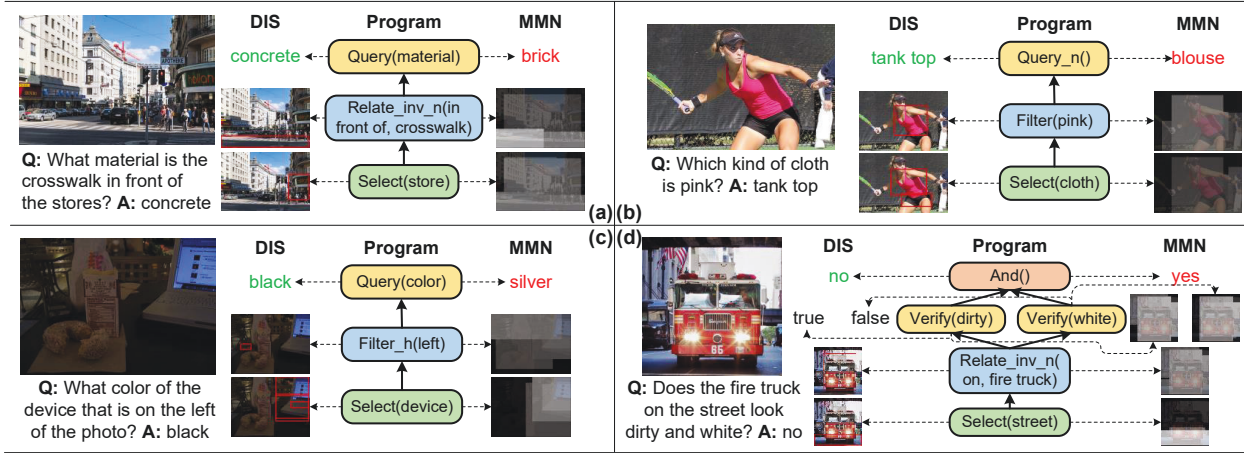
Figure 5: Visualization of intermediate results and predicted answers from MMN and our proposed DIS model.

Format	Accuracy
name, x1, y1, x2, y2	59.40
x1, y1, x2, y2, name	59.85
x1, y1, x2, y2	**59.97**

Table 3: Results of different object supervision formats.

α	Accuracy	BS Epoch	Accuracy
0.00	58.75	1	**61.31**
0.25	58.91	2	61.04
0.50	**59.97**	3	60.95
0.75	59.47	4	60.61
1.00	59.74	5	60.59

Table 4: Results of different DIS loss weights and bootstrapping epochs.

Different *Object* supervision format: Table 3 shows the results with different object supervision formats. $(x1, y1)$, $(x2, y2)$ and $name$ denote the top-left, bottom-right coordinates of bounding box, and object label, respectively. As Table 3 shows, the bounding box information is enough for intermediate supervision, which achieves the highest score (*i.e.,* 59.97%), and the extra object names decrease the performance. It is conjectured that extra supervisions (*i.e.,* object names) enforce the VQA model focus more on intermediate results than final answer prediction, thus decreasing the answer prediction performance.

Different loss weights of DIS and bootstrapping epochs: Table 4 shows the results of different DIS loss weights, and different epochs for bootstrapping. As shown, the best accuracy (59.97%) is achieved when DIS loss weight $\alpha = 0.5$, which surpasses the baseline $\alpha = 0$ by a significant margin of 1.22%, demonstrating the effectiveness of our proposed DIS method. In addition, it can be observed that the bootstrapping of 1 epoch has achieved the best accuracy (61.31%), while decreasing the performance with more bootstrapping epochs. It is conjectured that more training epochs on *all-split* bring biases for the VQA model, which is harmful for further fine-tuning. **Additionally**, the results regarding how different quantized bins and maximum number of generated objects and the shuffle mode of object order affect the performance are shown in Appendix-D.

Visualization

We visualize several cases from the GQA *testdev* split, showcasing the intermediate results and predicted answers from MMN and DIS. As depicted in Figure 5, it is evident that MMN tends to focus on a large area of the image, *e.g.,* the attention maps of the *store*, *cloth*, and *fire truck* in Fig-

ure 5 (a), (b), and (d) are not tightly focused on. It might be the reason that the large bounding boxes are more likely to overlap with others. Consequently, these bounding boxes are frequently used for training, leading the model to prioritize larger areas for IoU-based intermediate supervision methods. On the contrary, our proposed DIS method is able to predict the tight bounding boxes, *e.g.,* the *crosswalk*, *store* in Figure 5 (a), *cloth* in Figure 5 (b), and *street*, *fire truck* in Figure 5 (d), which facilitates the model to learn more fine-grained cross-modal alignments and accurate reasoning procedures. In addition, as depicted in Figure 5 (b) and (c), there may exist multiple intermediate supervisions. However, MMN does not precisely focus on these areas, *e.g.,* the *cloth* in Figure 5 (b). The incomplete prediction of intermediate results makes it easy for the models to infer incorrect answers. In contrast, our proposed DIS is able to predict multiple object results in one sequence and unify the different result forms (*i.e.,* logical true/false and objects) using one single framework.

Conclusion

We propose the DIS algorithm for compositional visual question answering. Specifically, DIS exploits a unified generative framework to provide intermediate supervisions in a sequential form that provides more fine-grained and accurate supervisions, addressing the issue of supervision ambiguity and promoting cross-modal knowledge alignment. We conducted experiments on the GQA and GQA-Sub datasets and the experimental results demonstrate that DIS achieves competitive answer prediction performance and superior reasoning consistency compared to previous state-of-the-arts.

Acknowledgments

References

Agrawal, A.; Lu, J.; Antol, S.; Mitchell, M.; Zitnick, C. L.; Parikh, D.; and Batra, D. 2017. VQA: Visual Question Answering. *IJCV*, 123(1): 4–31.

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *CVPR*, 6077–6086.

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016. Neural Module Networks. In *CVPR*, 39–48.

Bodla, N.; Singh, B.; Chellappa, R.; and Davis, L. S. 2017. Soft-NMS – Improving Object Detection with One Line of Code. In *ICCV*, 5562–5570.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T. J.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *NeurIPS*, volume 33, 1877–1901.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*, 213–229.

Chen, T.; Saxena, S.; Li, L.; Fleet, D. J.; and Hinton, G. 2022. Pix2seq: A Language Modeling Framework for Object Detection. In *ICLR*.

Chen, W.; Gan, Z.; Li, L.; Cheng, Y.; Wang, W. Y.; and Liu, J. 2021. Meta Module Network for Compositional Visual Reasoning. In *WACV*, 655–664.

Girshick, R. B. 2015. Fast R-CNN. In *ICCV*, 1440–1448.

Guo, D.; Xu, C.; and Tao, D. 2019. Graph Reasoning Networks for Visual Question Answering. *arXiv*, abs/1907.09815.

Hu, R.; Andreas, J.; Darrell, T.; and Saenko, K. 2018. Explainable neural computation via stack neural module networks. In *ECCV*, 53–69.

Hu, R.; Rohrbach, A.; Darrell, T.; and Saenko, K. 2019. Language-Conditioned Graph Networks for Relational Reasoning. In *ICCV*, 10293–10302.

Hudson, D.; and Manning, C. D. 2019a. Learning by abstraction: The neural state machine. In *NeurIPS*, volume 32, 5901–5914.

Hudson, D. A.; and Manning, C. D. 2018. Compositional Attention Networks for Machine Reasoning. In *ICLR*.

Hudson, D. A.; and Manning, C. D. 2019b. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *CVPR*, 6693–6702.

Jiang, H.; Misra, I.; Rohrbach, M.; Learned-Miller, E. G.; and Chen, X. 2020. In Defense of Grid Features for Visual Question Answering. In *CVPR*, 10264–10273.

Jing, C.; Jia, Y.; Wu, Y.; Li, C.; and Wu, Q. 2022a. Learning the dynamics of visual relational reasoning via reinforced path routing. In *AAAI*, 1122–1130.

Jing, C.; Jia, Y.; Wu, Y.; Liu, X.; and Wu, Q. 2022b. Maintaining Reasoning Consistency in Compositional Visual Question Answering. In *CVPR*, 5089–5098.

Kim, J.-H.; Jun, J.; and Zhang, B.-T. 2018. Bilinear Attention Networks. In *NeurIPS*, 1571–1581.

Li, G.; Wang, X.; and Zhu, W. 2019. Perceptual Visual Reasoning with Knowledge Propagation. In *ACM MM*, 530–538.

Li, L.; Gan, Z.; Cheng, Y.; and Liu, J. 2019. Relation-Aware Graph Attention Network for Visual Question Answering. In *ICCV*, 10312–10321.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*, 1532–1543.

Raffel, C.; Shazeer, N. M.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, 21: 140:1–140:67.

Redmon, J.; Divvala, S. K.; Girshick, R. B.; and Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*, 779–788.

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE TPAMI*, 39: 1137–1149.

Tan, H.; and Bansal, M. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *EMNLP*, 5100–5111.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *ICCV*, 9626–9635.

Vaswani, A.; Shazeer, N. M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NeurIPS*, volume 30, 5998–6008.

Yang, X.; Lin, G.; Lv, F.; and Liu, F. 2020. Trrnet: Tiered relation reasoning for compositional visual question answering. In *ECCV*, 414–430.

Yu, Z.; Yu, J.; Cui, Y.; Tao, D.; and Tian, Q. 2019. Deep Modular Co-Attention Networks for Visual Question Answering. In *CVPR*, 6281–6290.

Zhao, Z.; Samel, K.; Chen, B.; and Song, L. 2021. ProTo: Program-Guided Transformer for Program-Guided Tasks. In *NeurIPS*, 17021–17036.