Modularized Zero-shot VQA with Pre-trained Models

Rui Cao and **Jing Jiang**
School of Computing and Information Systems
Singapore Management University
ruicao.2020@phdcs.smu.edu.sg, jingjiang@smu.edu.sg

Abstract

Large-scale pre-trained models (PTMs) show great zero-shot capabilities. In this paper, we study how to leverage them for zero-shot visual question answering (VQA). Our approach is motivated by a few observations. First, VQA questions often require multiple steps of reasoning, which is still a capability that most PTMs lack. Second, different steps in VQA reasoning chains require different skills such as object detection and relational reasoning, but a single PTM may not possess all these skills. Third, recent work on zero-shot VQA does not explicitly consider multi-step reasoning chains, which makes them less interpretable compared with a decomposition-based approach. We propose a modularized zero-shot network that explicitly decomposes questions into sub reasoning steps and is highly interpretable. We convert sub reasoning tasks to acceptable objectives of PTMs and assign tasks to proper PTMs without any adaptation. Our experiments on two VQA benchmarks under the zero-shot setting demonstrate the effectiveness of our method and better interpretability compared with several baselines.

1 Introduction

Visual Question Answering (VQA), the task of answering textual queries based on information contained in an image, is a multimodal task that requires comprehension and reasoning of both visual and textual content (Agrawal et al., 2017; Hudson and Manning, 2019). Most previous work on VQA either trains VQA models from scratch (e.g., Fukui et al. (2016); Anderson et al. (2018)) or fine-tunes pre-trained vision-language models for VQA (e.g., Li et al. (2019); Lu et al. (2019)). Thus, they rely heavily on labeled VQA data, which are expensive to obtain. VQA models based on supervised learning are also hard to generalize to new domains or new datasets (Xu et al., 2020; Chao et al., 2018; Zhang et al., 2021).

Recently, large-scale pre-trained models (PTMs) have demonstrated strong transferability to different downstream tasks under zero-shot settings, i.e., without any training data for the downstream tasks (Brown et al., 2020; Radford et al., 2021). With increased pre-training data size, these models show strong zero-shot performance on various down-stream tasks, such as image classification and face detection with the CLIP model (Radford et al., 2021) and sentiment analysis and commonsense question answering with the GPT-3 model (Brown et al., 2020). However, few studies have focused on zero-shot VQA from pre-trained models.

Despite the power of these PTMs, it is not straightforward to directly apply them to VQA under zero-shot settings, because they are not pre-trained with the same objective as VQA. Some recent work converts images to tokens that pre-trained language models can understand so that VQA can be converted to text-based QA (Yang et al., 2022b; Tiong et al., 2022; Tsimpoukelli et al., 2021; Jin et al., 2022; Dai et al., 2022). However, this approach requires either a strong pre-trained image captioning model that can capture sufficient visual details or auxiliary training to obtain such a captioning model. Some other work converts VQA into a multimodal matching problem so that pre-trained vision-language models (PT-VLMs) such as CLIP can be used (Song et al., 2022; Shen et al., 2022). However, complex VQA questions such as those found in the GQA dataset (Hudson and Manning, 2019) often require spatial reasoning and/or multi-step reasoning, which PT-VLMs may not be strong at (Subramanian et al., 2022; Thrush et al., 2022).

VQA questions can be complicated and often require different reasoning steps such as object detection and spatial reasoning, as the example question in Figure 1 illustrates. Previously, people proposed Neural Module Networks (Andreas et al., 2016; Hu et al., 2017), which are modularized net-

Findings of the Association for Computational Linguistics: ACL 2023, pages 58–76
July 9-14, 2023 ©2023 Association for Computational Linguistics

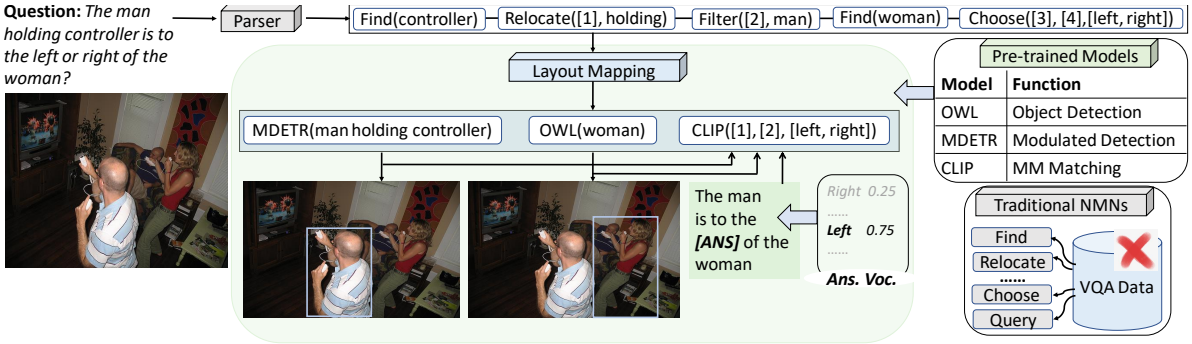

Figure 1: An overview of our proposed method. Instead of training modules in NMN, we propose a modularized zero-shot VQA method leveraging pre-trained models to perform different reasoning tasks.

works where each pre-defined module performs a specific reasoning task. These pre-defined modules are trained end-to-end from labeled VQA data. Motivated by the idea of modularization, in this paper, we propose a modularized zero-shot network for VQA (**Mod-Zero-VQA**) by decomposing questions into sub-tasks and assigning appropriate sub-tasks to PTMs without any adaptation. Given a question, we first parse the question into basic reasoning steps explicitly. These reasoning steps will then be reconfigured and mapped to different PTMs based on a set of rules we define. Specifically, we consider the following PTMs: *OWL* (Minderer et al., 2022) as the object detector, *MDETR* (Kamath et al., 2021) for reference expression localization (including several skills such as relational and spatial reasoning) and *CLIP* (Radford et al., 2021) as the answer generator for open-ended questions. Considering the limited capabilities of current pre-trained vision-language models in spatial relation understanding (Subramanian et al., 2022), we also define simple and general heuristics to aid spatial reasoning. Note that only when we decompose questions and reasoning chains step by step can we insert human heuristics for spatial reasoning, because we have the intermediate outputs such as objects' bounding boxes from previous steps.

We evaluate the proposed method on the GQA dataset (Hudson and Manning, 2019) where questions are compositional and require multi-step reasoning. The experiment result shows that the proposed model surpasses the baselines significantly on GQA, with near $\mathbf{13}\%$ of relative improvement over the strongest baseline (from 41.9 to 47.3). The results confirm the benefit of modualarization when using PTMs for zero-shot VQA. In addition, our method is interpretable because of the explicit reasoning steps generated.

The contributions of our work can be summarized as follows: (1) We propose a novel modularized zero-shot VQA method that utilizes different pre-trained models for different reasoning steps; (2) We design rules to map different VQA reasoning steps to suitable PTMs so that we can leverage these PTMs without any adaptation; 3) Experiment results show the effectiveness of the proposed method, especially when questions consist of multiple steps of reasoning.

2 Background

Task Definition. Given an image I and a question Q, a VQA system is expected to return an answer a. Traditional fully supervised VQA relies on a training set consisting of (image, question, answer) triplets. For zero-shot VQA, no such training data is given. However, in this paper we assume that we can use pre-trained models (PTMs) to help us with zero-shot VQA.

Existing Zero-shot VQA Methods. Work on zero-shot VQA is very limited. We can organize existing work into the following categories. One line of work leverages the question answering capability in pre-trained language model (LMs). Some of them adopt prefix language modeling with weakly-supervised data other than VQA data (i.e., image-text pairs) to convert visual information into discrete tokens (prefix) that LMs can understand. Frozen (Tsimpoukelli et al., 2021), VLKD (Dai et al., 2022) and FewVLM (Jin et al., 2022) fall under this category. Some directly convert VQA images into textual descriptions so that the task of VQA changes to text-based QA and LMs can be applied. Methods in this category include PICa (Yang et al., 2022b) and PnP-VQA (Tiong et al., 2022). Recent work (Song et al., 2022; Shen et al., 2022)

converts VQA to an image-text matching problem and prompts the CLIP model (Radford et al., 2021), a large-scale vision-language model pre-trained on the image-text matching task. The prompts can be either question irrelevant such as *Quesion: [*`Ques`*]; Answer: [*`MASK`*]* (QIP by Shen et al. (2022)) or question-related by converting questions into a masked statement (TAC-P by Song et al. (2022)).

However, a limitation with these methods is that several of them still require training, although the training data is not in the form of VQA. Besides, converting images to captions and leveraging text-based QA may lose important visual details during the caption generation step. The two methods above using CLIP do not address the issue that CLIP model lacks compositional and spatial reasoning abilities, which has been observed in previous work (Subramanian et al., 2022; Thrush et al., 2022).

3 Modularized Zero-shot VQA

Our method is motivated by Neural Module Network (NMN) based VQA, which decomposes questions into reasoning steps, where each module in the NMN is pre-defined to perform a specific reasoning task. The idea allows us to select appropriate pre-trained models to handle different reasoning tasks in a question. Specifically, in NMN-based VQA, we first manually define a set of reasoning steps such as object detection and spatial reasoning, each represented by a *module*. A question is then explicitly decomposed and converted into a *layout* of modules, which is an executable program showing the reasoning chain to reach the final answer. The top section of Figure 1 shows the layout corresponding to the sample question. To train an NMN-based VQA system, usually a layout generator is separately built first, which either uses hand-crafted rules over dependency parses of questions or is a trained seq2seq model. Then, the parameters of the various VQA modules are learned from VQA training data.

For our work, we do not want to use VQA data for training. But we observe that many modules in NMN-based VQA can be supported by pre-trained models that have already acquired the capabilities needed by these modules. The key component of our method is therefore to map a layout of modules produced by traditional NMN-based VQA to a simplified layout of zero-shot components that can be implemented directly using pre-trained models.

3.1 Traditional VQA Modules

There is not any standard set of modules for VQA. We largely adopt the design of modules introduced by Hu et al. (2017) with some minor changes. We assume that the image has been pre-processed and N bounding boxes have been detected, each represented as an embedding vector, collectively denoted as $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N)$. An attention map $\boldsymbol{\alpha}$ is defined to be a distribution over the N bounding boxes.

Table 1 lists the most important traditional VQA modules that we will replace with pre-trained models. The full list of modules can be found in Table 7 in the appendices. It is worth explaining that besides taking in $\mathbf{V}$ and $\boldsymbol{\alpha}$ as either input or output, many modules also take in the word embeddings of some text description extracted from the question. These text embeddings are arguments to control the behaviors of the modules. For example, the `Find` module's objective is to locate an object among all the bounding boxes given. The textual input $\mathbf{g}_{\text{OBJ}}$ is therefore the word embedding of the name of the object to be found. Similarly, $\mathbf{g}_{\text{RELA}}$,$\mathbf{g}_{\text{ATTR}}$ and $\mathbf{g}_{\text{QUERY}}$ are word embeddings for the description of relation (e.g., *to the left of*), attribute (e.g., *red*) and aspect to query (e.g., querying *name*).

Module	Inputs
`Find`	$\mathbf{V}$, $\mathbf{g}_{\text{OBJ}}$
`Relocate`	$\boldsymbol{\alpha}$, $\mathbf{V}$, $\mathbf{g}_{\text{RELA}}$
`Filter`	$\boldsymbol{\alpha}$, $\mathbf{V}$, $\mathbf{g}_{\text{CONDI}}$
`Choose`	$\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$, $\mathbf{V}$, $\mathbf{g}_{\text{RELA}^1}$, $\mathbf{g}_{\text{RELA}^2}$
`Query`	$\boldsymbol{\alpha}$, $\mathbf{V}$, $\mathbf{g}_{\text{QUERY}}$

Table 1: A subset of the modules in traditional NMN that we replace with pre-trained models. Modules in the first block output an attention map and those in the second block generate an answer.

Traditionally, the parameters of the modules in Table 1 need to be learned from VQA training data. In other words, these modules' underlying capabilities such as object recognition and relational reasoning need to be acquired from VQA data. However, we hypothesize that recently developed pre-trained models may already have some of these capabilities and can therefore directly equip these modules with such capabilities. For example, the `Find` module is mainly responsible for object recognition, and previously the parameters of `Find` have to be learned from scratch using VQA data. Now with

a powerful pre-trained model such as OWL (Minderer et al., 2022) that can recognize a wide range of objects, we can presumably directly use a model like OWL to replace the traditional `Find` module.

3.2 Pre-trained Models

We utilize three pre-trained models that we believe are highly relevant to VQA.

OWL. The Vision Transformer for Open-World Localization (OWL) model (Minderer et al., 2022) is a model for open-vocabulary object detection. It is first pre-trained on large-scale image-text pairs and then fine-tuned with added detection heads and medium-sized detection data. Given the category name of an object and an image, the model is able to locate bounding box(es) in the image containing the object together with a confidence score for each box.

MDETR. The modulated DETER (DEtection TRansformer) model (Kamath et al., 2021) is an end-to-end detector that can detect an object in an image conditioned on a piece of textual description of the object such as its attributes and its relation with another object in the image. The model is pre-trained on image-text pairs with explicit alignment between phrases in the text and bounding boxes of objects in the image. Given an image and the description of an object, MDETR is able to locate the bounding box(es) in the image containing the object satisfying the description. Note that different from OWL, MDETR is able to understand textual descriptions that may contain attribute information and/or complex visual relations. For example, given the description *a man holding a yellow cup is talking*, MDETR will detect the bounding box containing the man holding a yellow cup in the given image, whereas OWL is not able to use the description and will only recognize all bounding boxes containing a man. Note that we use the version of MDETR pre-trained on general modulated detection **without fine-tuning** for any downstream tasks.

CLIP. CLIP is a well-known large-scale vision-language model by OpenAI. It is pre-trained with 400M image-caption pairs through contrastive learning. Given an (image, text) pair, CLIP uses its separate image encoder and text encoder to turn the image and the text each into a vector, and the cosine similarity between the two vectors directly measures the compatibility of the two. Recent work has shown that CLIP can be directly used for VQA in a zero-shot setting, if we can come up with a set of candidate answers and transform each (question, answer) pair into a statement (Song et al., 2022).

3.3 Zero-shot NMN using Pre-trained Models

Based on the descriptions of the traditional VQA modules in Section 3.1 and of the three PTMs we consider in Section 3.2, we can see that there are obvious connections between the capabilities desired by the traditional modules and the capabilities that these PTMs have already acquired.

However, the mapping between them is not trivial. First of all, there is no simple one-to-one mapping from traditional VQA modules to the PTMs. For example, the MDETR model can already perform multiple steps of reasoning to locate the desired object, so it can be used to cover a sequence of modules in an NMN layout. Second, there may be capabilities required when applying PTMs but not captured by modules defined in NMN-based VQA. In particular, the MDETR model always assumes that the object to be grounded exists in the given image, but for those questions asking for the existence of a specified object, we cannot directly use MDETR.

To address these challenges, we carefully design a mapping mechanism that can map an NMN-based module layout to a simplified layout consisting of a few zero-shot modules. Three of these zero-shot modules (`OWL`, `MDETR` and `CLIP`) correspond exactly to the three PTMs introduced earlier. The rest of the zero-shot modules are defined by simple heuristic rules. We list these zero-shot modules in Table 2.

Module	Inputs	Output
`OWL`	I, OBJ	$\mathcal{B}$, s
`MDETR`	I, SENT	$\mathcal{B}$, s
`CLIP`	$\mathcal{B}$, I, $\mathcal{V}$	Ans.
`Count`	$\mathcal{B}$	Num.
`Exist`	$\mathcal{B}$, (ATTR/RELA)	*Yes/No*
`And`	$\texttt{Exist}_1$, $\texttt{Exist}_2$	*Yes/No*
`Or`	$\texttt{Exist}_1$, $\texttt{Exist}_2$	*Yes/No*

Table 2: Zero-shot modules with either pre-trained models or heuristics. The I is the VQA image, $\mathcal{V}$ is the answer vocabulary and $\mathcal{B}$ is the set of bounding boxes.

We now give a high-level summary of the mapping mechanism below. We first look at the last module in the NMN layout. If the last module

is one of Choose, Compare and Query, we know that the input to this last module is either a single attention map or two attention maps, where each attention map essentially tries to capture an object matching some textual descriptions. By tracing the path in the layout leading to the attention map, we choose either the zero-shot OWL module (when the path has a length of 1) or the zero-shot MDETR module (when the path is longer than 1 hop). This is because when the path length equals to one, it involves only object detection (corresponding to a single Find module in the NMN layout for generation of the attention map). When the path length is more than one, it indicates the generation of the attention map in the NMN layout involves other modules such as Filter and Relocate, which calls for the other abilities than object detection, such as language understanding, attribute recognition and relational reasoning. Different from NMN modules which takes in image features and object embeddings to generate an attention map, our zero-shot OWL and zero-shot MDETR takes in the raw image and raw texts to locate (OBJ for OWL and SENT for MDETR) to generate a set of detected bounding boxes $\mathcal{B} = \{\mathbf{b}_n\}_{n=1}^{N}$ together with their confident scores $\mathbf{s} \in \mathbb{R}^N$, where $\mathbf{b}_n \in \mathbb{R}^4$ represents the relative position and size of the detected bounding box in the image. We keep only the bounding box from either OWL or MDETR with the highest confident score and feed it to CLIP. We generate an answer by leveraging the capability of multimodal matching of CLIP. Specifically, given $\mathcal{B}$, we generate an input image (which we refer to as I^{in}) by either masking regions not containing those detected boxes ($|\mathcal{B}| = 2$) or cropping the image so that only the part containing the box remains ($|\mathcal{B}| = 1$). If the final NMN module is Choose, we generate a masked template by question conversion as in (Song et al., 2022); otherwise the masked template will be a simple "[MASK]". Then we match the image I^{in} with the template where the [MASK] token is replaced by each of the answer candidates in $\mathcal{V}$. We then select the answer that, when placed inside the template, best matches the image.

If the module is Exist, we trace back the path leading to Exist to determine whether the module is asking for the existence of an object, an attribute or a relation. For object existence (e.g., *is there a car*), we use the zero-shot OWL module. For attribute existence and relation existence, we first verify whether all mentioned nouns (objects) detected by a POS tagger in the question exist with the OWL module. Once we detect an object that does not exist, the predicted answer will be *no*. If all objects exist, then we generate corresponding bounding boxes leveraging either OWL or MDETR following the method described in the paragraph above. For attribute existence, we generate a pair of a positive and a negative descriptions: (ATTR, *not* ATTR), e.g., (*red*, *not red*). We then find which description aligns better with the cropped image according to $\mathbf{b}$. If the image aligns better with the positive statement, then the answer will be *yes*; otherwise, *no*. For relation existence, we generate the masked image I^{in} according to $\mathbf{b}_1$ and $\mathbf{b}_2$ (the bounding boxes of the two objects whose relation is to be checked) and a pair of opposite statements regarding the relation to be checked, following (Song et al., 2022). For example, if the question is to check whether *A* is holding *B*, the two opposite statements will be *A is holding B* and *A is not holding B*. For both attribute and relation existence, we use zero-shot CLIP for the alignment between the input image and the statements. More details and the work flows of existence-related questions are provided in Appendix C.

If the module is Count, we directly count the number of bounding boxes in $\mathcal{B}$ returned either from OWL or MDETR. Finally, if the last module is a logical AND or logical OR, we further trace to the inputs of this module, which should both be an Exist module. We then use the same mechanism described above for Exist to process the module. By receiving the outputs from the Exist modules, logical operations will be applied to determine the output. The deterministic logical operations can be found in Appendix B.

3.4 Spatial Heuristics

As mentioned in (Subramanian et al., 2022), CLIP is less capable of spatial reasoning. Using CLIP for answer generation may not be enough when it involves spatial relation understanding. Following (Subramanian et al., 2022), we define simple and general heuristics to perform certain types of spatial reasoning. Note that only when we decompose questions explicitly can we insert the spatial heuristics into CLIP-based answer generation because we have the intermediate outputs from previous reasoning steps.

First of all, given the coordinates and the size of a bounding box, we use manual rules (named

as **SpD**) to decide its position in the image as *left, right, bottom, top*. Besides, we define heuristics, denoted as **SpC**, to solve spatial relations between two bounding boxes (e.g., *to the left of* and *to the right of*).

Details of the implementation of the spatial relation solvers can be found in Appendix D.

4 Experiments

4.1 Dataset

We evaluate the proposed modularized zero-shot VQA method on two benchmarks: GQA (Hudson and Manning, 2019) and VQAv2 (Goyal et al., 2017). The GQA dataset consists of questions requiring multi-step reasoning and various reasoning skills. Around 94% of the questions require multiple reasoning steps. We regard it as the main dataset to demonstrate the effectiveness of the proposed method compared with the baselines. Compared with GQA, questions on the VQAv2 dataset require fewer reasoning steps and are of diverse semantics. We use VQAv2 to show the validity of our method in real-world VQA. We report standard accuracy for the GQA dataset while soft accuracy (Goyal et al., 2017) for VQAv2 dataset as there are multiple ground-truth answers. We report the statistics of the datasets in Appendix E.

4.2 Implementation Details

We conduct experiments on NVIDIA Tesla V100 GPU. The thresholds for the OWL and the MDETR model to filter out detected bounding boxes of low confidente scores are set to be 0.2 and 0.7 respectively. We follow (Song et al., 2022) for the generation of the answer vocabulary $\mathcal{V}$ for open-ended questions. More details about answer vocabulary generation can be found in Appendix G and more information about experiment settings can be found in Appendix G.

4.3 Main Results

Zero-shot VQA performance of the baselines mentioned in Section 2 and our proposed method are summarized in Table 3[1].

First of all, we observe that the proposed Mod-Zero-VQA method ismore effective on the GQA dataset, which contains many multi-step reasoning questions. Mod-Zero-VQA clearly surpasses all baselines on GQA. The results suggest that it is effective under zero-shot settings to decompose questions when questions are compositional and require several steps of reasoning to reach the answer. Such decomposition allows us to take advantage of the capabilities of different pre-trained models. We also test the validity of the proposed method on real-world VQAv2 dataset, where questions require fewer reasoning steps and of diverse semantics. We can see that our method still achieves the best performance among zero-shot methods that utilize CLIP. Although better performance is achieved by several methods that utilize large language models (as shown in the first block of Table 3), it is worth pointing out that these methods often require caption generation as a pre-processing step, and this step poses challenges. For example, PNP-VQA generates **100** captions per question, which is laborious. There may also be redundancy because many captions are irrelevant for question answering. Another advantage of our Mod-Zero-VQA method over the other zero-shot baselines is that our method offers high interpretability by showing the explicit multi-step reasoning chain, which has not been considered by any previous work. With question decomposition, we can design modularized networks and assign reasoning tasks to pre-trained models (PTMs) which are more capable of the tasks, and with more powerful pre-trained models coming out, our method can be easily extended to utilize newer and more effective PTMs. Meanwhile, it is easier to pinpoint the weakest chain in a system and insert human heuristics to aid these modules.

Method	GQA	VQA
Frozen	-	29.5
$\text{VLKD}_{\text{ViT-L/14}}$	-	42.6
$\text{FEWVLM}_{\text{base}}$	27.0	43.4
$\text{FEWVLM}_{\text{large}}$	29.3	47.7
$\text{PNP-VQA}_{\text{6M}}$	34.6	54.3
$\text{PNP-VQA}_{\text{11B}}$	**41.9**	**63.3**
QIP	35.9	21.4
TAP-C	36.3	38.7
Mod-Zero-VQA	**47.3**	**41.0**

Table 3: Experimental results on the GQA and VQA datasets. The first block are models using the text-based QA capability of LMs and the second blocks are models incorporating CLIP.

[1]For FEWVLM and PNP-VQA model, we show their reported performances on GQA test-dev, which should have similar distributions as the validation split of GQA.

Detector	Yes/No Qns	Other Qns	Overall
CLIP-FR	56.80	33.82	41.39
OWL	69.26	36.48	47.28
GT	76.48	38.06	50.72

Table 4: Performance of Mod-Zero-VQA with different object detectors on GQA.

Method	PT-VLMs	Overall
QIP	$\text{CLIP}_{\text{ViT-B/16}}$	35.93
	$\text{CLIP}_{\text{Res50}\times16}$	35.11
	ALBEF	34.75
TAP-C	$\text{CLIP}_{\text{ViT-B/16}}$	36.32
	$\text{CLIP}_{\text{Res50}\times16}$	38.16
	ALBEF	38.36
Mod-Zero-VQA	$\text{CLIP}_{\text{ViT-B/16}}$	47.28
	$\text{CLIP}_{\text{Res50}\times16}$	46.49
	ALBEF	**48.68**

Table 5: Performance of the Mod-Zero-VQA model with different PT-VLMs as the zero-shot `CLIP` for answer generation on GQA.

4.4 Ablation Study

In our Mod-Zero-VQA method, PTMs play an important role. In this section, we show the performance of Mod-Zero-VQA when we replace PTMs listed in Section 3.2 with alternative models.

Replacing OWL: We tried replacing OWL with other object detectors. First, we consider an object detector combining Faster-RCNN (Ren et al., 2015) and CLIP (**CLIP-FR**). Specifically, Faster-RCNN is used to detect objects in an image and CLIP is applied to classify each detected object. Second, we use the ground-truth object annotations from Visual Genome (Krishna et al., 2017) to replace object detection results (**GT**), which serves as an upper bound. Results of our zero-shot NMNs with different object detectors are provided in Table 4. We divide the questions into Yes/No (bindary) questions and other questions. We observe that the quality of object detection is important to the performance of zero-shot NMNs. Our model with OWL surpasses the one with CLIP-FR, which has poorer detection performance than OWL. We also observe more substantial performance drop with binary questions. We believe that this is because these questions are mostly about the existence of objects, so the object detection results affect the VQA performance more. Using Mod-Zero-VQA with the ground-truth object detection results would further improve the performance, as shown in the last row of Table 4. This suggests that when more accurate object detection models are developed, we can further improve the zero-shot VQA performance with our approach.

Replacing CLIP: We show the performance of replacing zero-shot `CLIP` (which is $\text{CLIP}_{\text{ViT-B/16}}$ by default in our experiments), with either $\text{CLIP}_{\text{Res50}\times16}$ or ALBEF (Li et al., 2021), in Table 5. Because QIP and TAC-P convert VQA to a multi-modal matching task and both use PT-VLMs as the answer generator, we also replace the original $\text{CLIP}_{\text{ViT-B/16}}$ in these two baselines with the other PTMs. We observe that Mod-Zero-VQA gives stable performance regardless of the vision-language model used, and it always outperforms the baselines substantially. This indicates that these PTMs can all be good substitutes for the zero-shot `CLIP` module. Compared with the two CLIP models (i.e., with either ViT (Dosovitskiy et al., 2021) or ResNet (He et al., 2016) as the visual backbone), we also notice that using ALBEF (Li et al., 2021) as the answer generator can enhance the performance. To better understand the advantage of using ALBEF over CLIP, we provide more detailed performance in Table 9 in Appendix H. ALBEF mostly benefits the proposed method in the *Query* type of questions, which usually ask about *objects*, *attributes* and *relations*. Consistent with (Zhao et al., 2022), end-to-end models (i.e., ALBEF in this case) perform better than dual-encoder models (i.e., CLIP in this case) in vision understanding tasks on average. A future direction may be to select the best pre-trained model per question.

4.5 Out-of-Domain Generalization

Because our Mod-Zero-VQA method is not trained on any domain-specific VQA data but rather utilizes pre-trained models that are supposedly trained on data from a wide range of domains, we suspect that our Mod-Zero-VQA method is more robust across different domains compared with VQA models trained on specific domains and applied in cross-domain settings. We therefore also compare our Mod-Zero-VQA with fully-supervised models in the Out-of-Domain Generalization (OOD) setting. Specifically, we consider an OOD setting where test images are related to scenes not observed during training. We first identify a set of scene-related objects and restrict all training images to only those that do not contain these objects. For example, in the *Indoor* OOD setting, none of the training im-

Method	Indoor	Food	Street
BUTD	39.27	32.28	35.96
NMNs	39.45	32.47	36.05
VilBert	39.87	32.12	36.68
VisualBert	41.14	33.47	38.51
ALBEF	45.55	38.87	41.60
Mod-Zero-VQA	48.86	47.80	49.54

Table 6: Comparison between our Mod-Zero-VQA method and fully-supervised VQA models under the out-of-domain setting.

ages should contain *sofa*, *bed* or any of the other objects that we have identified to be related to *Indoor* scenes. To build fully-supervised VQA models for comparison, we consider (1) **BUTD** (Anderson et al., 2018), a classic two-stream VQA models, (2) traditional **NMNs** (Andreas et al., 2016), and (3) finetuned pre-trained vision-language models, including **VilBert** (Lu et al., 2019), **VisualBert** (Li et al., 2019) and **ALBEF** (Li et al., 2021).

The results are shown in Table 6. We can see from the table that for those supervised VQA models, when they are trained on images with different scenes, their performance on the target domain is clearly lower than our Mod-Zero-VQA. Furthermore, our Mod-Zero-VQA method achieves steady performance across different scenes, whereas the supervised VQA models give fluctuated performance in different scenes. This demonstrates the robustness of our proposed method.

4.6 Case Study

As a case study, we visualize the outputs of the reasoning steps from the proposed method and compare the prediction of the proposed method with those of QIP and TAC-P, which also leverage CLIP as the answer generator. We show two example questions and the outputs in Figure 2. Both questions require multiple reasoning steps.

We can see that our method gives the correct predictions while the two other methods answer wrongly. We can also see that by decomposing the questions, our method assigns each sub reasoning task to a pre-trained model capable of the task (i.e., MDETR for reference expression localization and OWL for object detection). With question decomposition, we can also better pinpoint the weaknesses of pre-trained models and insert human knowledge by defining simple but general heuristics (e.g., adding spatial heuristics to zero-shot `CLIP` and defining logical operations). More

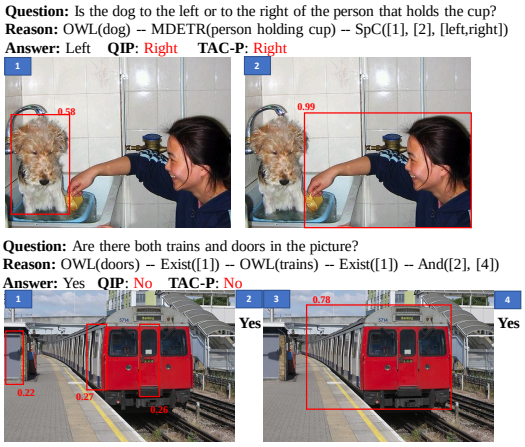

Figure 2: Visualization of intermediate outputs from reasoning steps of the Mod-Zero-VQA model.

examples with visualization are provided in Appendix G.

5 Related Work

5.1 Visual question answering

Although great progress has been made in the supervised VQA setting (Li et al., 2019; Lu et al., 2019; Li et al., 2022, 2021), few studies have explored the zero-shot VQA setting. One line of work converts VQA to text-based QA so that language models (LMs) can be applied. Some of them require auxiliary training though not with VQA data (Dai et al., 2022; Jin et al., 2022; Tsimpoukelli et al., 2021). Some suffer from insufficient visual details (Yang et al., 2022b) or laborious generation of irrelevant captions (Tiong et al., 2022). Others (Shen et al., 2022; Song et al., 2022) convert VQA to multimodal matching and leverage CLIP (Radford et al., 2021). However, CLIP is limited when compositional reasoning and spatial reasoning are required (Thrush et al., 2022; Subramanian et al., 2022). In this work, we propose to decompose questions and propose a modularized zero-shot VQA method by assigning reasoning tasks to proper pre-trained models without any adaptation.

5.2 Zero-shot applications of pre-trained models

Models pre-trained on a large corpus have strong zero-shot transferability when performing downstream tasks whose objectives are similar to the pre-training objectives of these models. For instance, GPT-3 (Brown et al., 2020) is powerful for zero-shot QA by treating the QA as a text genera-

tion problem. CLIP (Radford et al., 2021) demonstrates good zero-shot image recognition capability by treating the classification task as multimodal matching. For multimodal QA tasks, LMs can be applied once information from other modalities are translated to tokens LMs understand (Tiong et al., 2022; Yang et al., 2022a). In our work, we decompose VQA questions into sub-reasoning tasks and assign sub-tasks to corresponding pre-trained models whose pre-training objectives match the sub-tasks.

6 Conclusion and Future Work

In this work, we propose a modularized zero-shot VQA method, motivated by the idea of Neural Module Network (NMN). Instead of training modules in NMN with VQA data, we decompose questions into reasoning tasks explicitly, leverage pre-trained models and assign proper reasoning tasks to them. Experiments show that our model is powerful on questions requiring multi-step reasoning and applicable for real-world VQA. Besides, the proposed model is highly interpretable, which helps to pinpoint weaknesses of a VQA system, making it easier to improve a system. Our model highlights a future direction of leveraging pre-trained models for other complicated tasks requiring multiple reasoning capabilities.

Limitations

In this section, we discuss few limitations of the proposed method and point out future directions to improve the model. First, our method needs to decompose questions into a symbolic representation, but such representations are hard for humans to comprehend, and therefore this decomposition mechanism is hard to be trained with human annotation. A promising direction is to leverage pre-trained language models such as ChatGPT [2] to automate this decomposition step, leveraging ChatGPT's internal knowledge of decomposing a complex question into sub-questions. Second, the execution of the zero-shot NMNs is conducted in a deterministic manner, leading to high risks of error propagation if the reasoning chain gets longer. In the future, we can consider a softer way of reasoning over the image with pre-trained models.

[2] https://openai.com/blog/chatgpt/

Acknowledgement

This research was supported by the SMU-A*STAR Joint Lab in Social and Human-Centered Computing (Grant No. SAJL-2022-HAS002).

References

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2017. VQA: visual question answering - www.visualqa.org. *Int. J. Comput. Vis.*, 123(1):4–31.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 39–48.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systemsm, NeurIPS*.

Wei-Lun Chao, Hexiang Hu, and Fei Sha. 2018. Cross-dataset adaptation for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 5716–5725.

Wenhu Chen, Zhe Gan, Linjie Li, Yu Cheng, William Yang Wang, and Jingjing Liu. 2021. Meta module network for compositional visual reasoning. In *IEEE Winter Conference on Applications of Computer Vision, WACV*, pages 655–664. IEEE.

Wenliang Dai, Lu Hou, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Enabling multimodal generation on CLIP via vision-language knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL*, pages 2383–2395.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *CoRR*, abs/1809.02922.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 731–742.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR*.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 457–468.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 6325–6334.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2020. Neural module networks for reasoning over text. In *8th International Conference on Learning Representations, ICLR*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 770–778.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *IEEE International Conference on Computer Vision, ICCV*, pages 804–813.

Drew A. Hudson and Christopher D. Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 6700–6709.

Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. 2022. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics ACL*, pages 2763–2775.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. MDETR - modulated detection for end-to-end multi-modal understanding. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV*, pages 1760–1770.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 123(1):32–73.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning, ICML*, volume 162, pages 12888–12900. PMLR.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems 34, NeurIPS*, pages 9694–9705.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 13–23.

Matthias Minderer, Alexey A. Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. 2022. Simple open-vocabulary object detection with vision transformers. *CoRR*, abs/2205.06230.

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. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning, ICML*, volume 139, pages 8748–8763.

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

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems*, pages 91–99.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2022. How much can CLIP benefit vision-and-language tasks? In *The Tenth International Conference on Learning Representations, ICLR*.

Haoyu Song, Li Dong, Weinan Zhang, Ting Liu, and Furu Wei. 2022. CLIP models are few-shot learners: Empirical studies on VQA and visual entailment. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL*, pages 6088–6100.

Sanjay Subramanian, William Merrill, Trevor Darrell, Matt Gardner, Sameer Singh, and Anna Rohrbach. 2022. Reclip: A strong zero-shot baseline for referring expression comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 5198–5215.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. *CoRR*, abs/2204.03162.

Anthony Meng Huat Tiong, Junnan Li, Boyang Li, Silvio Savarese, and Steven C. H. Hoi. 2022. Plug-and-play VQA: zero-shot VQA by conjoining large pretrained models with zero training. *CoRR*, abs/2210.08773.

Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 200–212.

Yiming Xu, Lin Chen, Zhongwei Cheng, Lixin Duan, and Jiebo Luo. 2020. Open-ended visual question answering by multi-modal domain adaptation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP*, volume EMNLP 2020 of *Findings of ACL*, pages 367–376.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2022a. Zero-shot video question answering via frozen bidirectional language models. *CoRR*, abs/2206.08155.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. 2022b. An empirical study of GPT-3 for few-shot knowledge-based VQA. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, pages 3081–3089.

Mingda Zhang, Tristan Maidment, Ahmad Diab, Adriana Kovashka, and Rebecca Hwa. 2021. Domain-robust VQA with diverse datasets and methods but no target labels. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 7046–7056.

Tiancheng Zhao, Tianqi Zhang, Mingwei Zhu, Haozhan Shen, Kyusong Lee, Xiaopeng Lu, and Jianwei Yin. 2022. Vl-checklist: Evaluating pre-trained vision-language models with objects, attributes and relations. *CoRR*, abs/2207.00221.

A Modules in VQA

We summarize all modules in traditional NMNs for VQA (Hu et al., 2017; Gupta et al., 2020; Chen et al., 2021) in Table 7.

B Logical Operations

In this section we describe the logical modules `And` and `Or`. Both of them receive outputs from two zero-shot `Exist` modules. For the `And` module, if both outputs are *yes*, it outputs *yes*; otherwise, it outputs *no*. For the `Or` module, if both outputs are *no*, it outputs *no*; otherwise, it outputs *yes*. The logical operators are deterministic.

C Existence Questions

As mentioned briefly in Section 3.3, for questions verifying the existence of something, according to the NMN layout, we classify these questions into three types: verifying existence of objects, of attributes, and of relations. For the verification of object existence, we directly apply the zero-shot `OWL`. For both attribute and relation verification questions, we first make sure all objects mentioned in the question exist with the help of `OWL`. If any mentioned objects do not exist, the predicted answer will be *No*, as illustrated in Figure 3. If the objects exist, we leverage either zero-shot `OWL` or `MDETR` to locate at objects of interests and verify the attributes and relations, with the utilization of the CLIP module. Examples are provided in Figure 4 (for attribute verification) and Figure 5 (for relation verification). We use CLIP for binary matching to select whether the attribute/relation exists. When multiple attributes/relations are to be verified, only when all attributes/relations exist will the predicted answer be *Yes*; otherwise, the prediction is *No*. For instance, the third example in Figure 4 has a dark brown table, but the table is not glass, so the third step outputs *no*. The final predicted answer to the question is therefore *no*.

D Detailed Implementation for Spatial Heuristics

In this section, we give the mathematical definitions of the spatial heuristics. The input bounding box is denoted as $\mathbf{b} = (x, y, w, h)$, representing the relative position and relative size of the object in the VQA image.

`Spatial Determine (SpD)` receives an object bounding box and determines which position in the original image the object is at. The position candidates $\mathcal{P}$ are generated according to the question. When the question is asking for the horizontal position of the object, $\mathcal{P} = \{\text{left}, \text{right}\}$; When the question is asking for the vertical position of the object, $\mathcal{P} = \{\text{top}, \text{bottom}\}$. The `SpD` module is implemented as:

$$\texttt{SpD}(\mathbf{b}, \mathcal{P}) = \begin{cases} \text{left}, & \text{if} \quad x < 0.5 \\ \text{right}, & \text{else} \end{cases} \tag{1}$$

when $\mathcal{P} = \{\text{left}, \text{right}\}$. When $\mathcal{P} = \{\text{top}, \text{bottom}\}$, the spatial heuristic is derived as:

$$\texttt{SpD}(\mathbf{b}, \mathcal{P}) = \begin{cases} \text{top}, & \text{if} \quad y < 0.5 \\ \text{bottom}, & \text{else} \end{cases} \tag{2}$$

The SpD heuristic will be used in the `Query` module when asking about either horizontal or vertical position.

Spatial Chooser (SpC) receives two bounding boxes of objects $\mathbf{b}_1, \mathbf{b}_2$ and aims to choose their spatial relations from the relation candidates in $\mathcal{C}$ ($\mathbf{b}_1$ is *RELA* $\mathbf{b}_2$). For instance, when $\mathcal{C} = \{\text{to the left of}, \text{to the right of}\}$:

$$\texttt{SpC}(\mathbf{b}_1, \mathbf{b}_2, \mathcal{C}) = \begin{cases} \text{left}, \text{if} & x_1 < x_2 \\ \text{right}, & \text{else} \end{cases} \tag{3}$$

When $\mathcal{C} = \{\text{above}, \text{beneath}\}$:

$$\texttt{SpC}(\mathbf{b}_1, \mathbf{b}_2, \mathcal{C}) = \begin{cases} \text{above}, \text{if} & y_1 < y_2 \\ \text{beneath}, & \text{else} \end{cases} \tag{4}$$

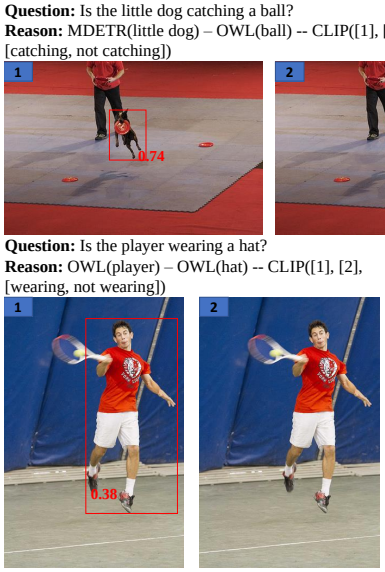

Figure 3: Visualization of existence-related questions where mentioned objects in the questions do not exist.

Module	Output	Functionality
Find($\mathbf{V}$, $\mathbf{g}_{\text{OBJ}}$)	Att.	Locate a certain object (OBJ) in the image
Relocate($\boldsymbol{\alpha}$,$\mathbf{V}$,$\mathbf{g}_{\text{RELA}}$)	Att.	Transit attention from previous attention map $\boldsymbol{\alpha}$ according to the relation (RELA)
Filter($\boldsymbol{\alpha}$,$\mathbf{V}$,$\mathbf{g}_{\text{CONDI}}$)	Att.	Highlight objects that are attended by previous attention map $\boldsymbol{\alpha}$ and satisfy the condition (CONDI)
Choose($\boldsymbol{\alpha}_1$,$\boldsymbol{\alpha}_2$,$\mathbf{V}$,$\mathbf{g}_{\text{RELA}^1}$,$\mathbf{g}_{\text{RELA}^2}$)	Ans.	Choose the relation from RELA^1 and RELA^2 between highlighted regions of two attention maps
Query($\boldsymbol{\alpha}$,$\mathbf{V}$,$\mathbf{g}_{\text{QUERY}}$)	Ans.	Generate a final answer given the attention map, image representation and item to query (QUERY)
Count($\boldsymbol{\alpha}$)	Ans.	Outputs a number given the attention map of the image
Exist($\boldsymbol{\alpha}$)	Ans.	Output a binary answer (*yes/no*) given the attention map of the image
And($\boldsymbol{\alpha}_1$,$\boldsymbol{\alpha}_2$)	Ans.	Generate a binary answer (*yes/no*) given the two attention maps
Or($\boldsymbol{\alpha}_1$,$\boldsymbol{\alpha}_2$)	Ans.	Generate a binary answer (*yes/no*) given the two attention maps

Table 7: The full list of modules in traditional NMNs. $\mathbf{g}_{[\cdot]}$ is the word embedding for the words in $[\cdot]$.

Dataset	Train		Val	
	# Ques.	# Img.	# Ques.	# Img.
GQA	943,000	72,140	132,062	10,234
VQA	443,757	82,783	214,354	40,504

Table 8: Statistical distributions of the GQA and the VQA dataset.

The SpC rule will be applied to the *Choose* type of questions if the choices of relations fall into the sets below: [{to the left of},{to the right of}] and [{above, on top of},{under, below, beneath, underneath}]

E Dataset Statistics

In Table 8, we provide statistics of the GQA and the VQA dataset. Following (Song et al., 2022; Tsimpoukelli et al., 2021), we use the validation split for testing. Specifically, we report *soft vqa scores* as there may be multiple possible answers to a question similar to previous works. (Song et al., 2022; Tsimpoukelli et al., 2021; Anderson et al., 2018; Fukui et al., 2016).

F Layout Generation

The layout generation can be accomplished either with syntatic parser or a pre-trained sequence-to-sequence layout generator. On the VQA dataset, we follow (Andreas et al., 2016; Hu et al., 2017) to parse questions with Stanza[3] and transform the parsed tree into reasoning graphs where each node is a pre-defined module with rules most similar to (Hu et al., 2017). The graphs are converted to module sequences with the post-order traversal. The linearlized module sequence is used as the layout. On GQA dataset, we leverage layouts generated by the pre-trained sequence-to-sequence layout generator from (Chen et al., 2021). The generator adopts a coarse-to-fine two-stage generation paradigm as in (Dong and Lapata, 2018) to encode questions and decode the sequence of module names and module inputs in two stages.

G Answer Filtering

Basically, we follow (Song et al., 2022) to narrow down the set of possible answer candidates with the language model T5 (Raffel et al., 2020). For the VQA dataset, we directly leverage the published generated candidate answers for each question from the paper (Song et al., 2022). For the GQA dataset, the *Verify* and *Logical* type questions have binary answers *yes/no*. For the *Compare* and

[3]https://github.com/stanfordnlp/stanza

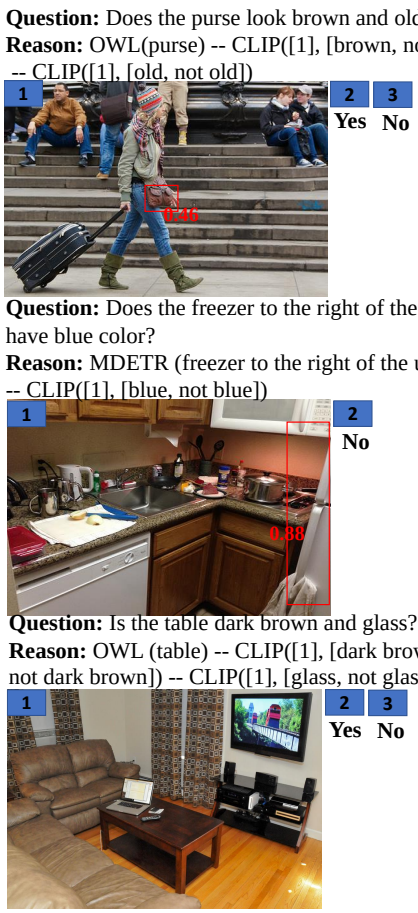

Figure 4: Visualization of questions asking about existence of attributes.

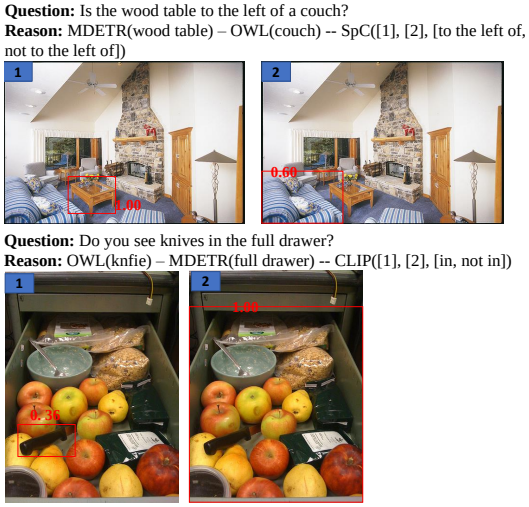

Figure 5: Visualization of questions asking about the existence of relations.

Choose, candidate answers are available in the generated layouts. For the *Query* type of questions, we first convert questions into masked templates with a rule-based converter (Demszky et al., 2018). T5 is applied to retrieve the masked word, which filters out irrelevant answers in the answer vocabulary according to contexts.

H Detailed Results

We provide the detailed results for replacing CLIP with ALBEF (discussed in Section 4.4) in Table 9 considering different types of questions.

I Visualization of Zero-shot NMNs

In this section, we provide more visualization examples which the zero-shot NMNs answers correctly while the baselines (QIP and TAC-P) fail. In Figure 6, we show examples with short reasoning chain, specifically, only two-step in ModZero-VQA. According to the results, we observe that each intermediate step gives interpretable outputs. By question decomposition and leveraging pre-trained models, our model can focus on relevant regions of the image (e.g., the first and third example in the first row of Figure 6) so that eliminating noise from backgrounds. Without filtering irrelevant information in the image, baselines pay attention to dominant objects in the image, leading to wrong predictions (e.g., the third example in the first row which QIP and TAC-P seems to focus on the ground and the T-shirt when answering the question). In Figure 7, we visualize questions with relatively longer reasoning chains. These compositional questions usually call several reasoning capabilities, making it hard for pre-trained VL models to deal with (Thrush et al., 2022). With question decomposition, each pre-trained model takes a sub reasoning task, easing the burden from answering a complicated question.

According to the visualization, we also find a frequent error case resulting from the wrongly-generated NMN layout. The coarse-to-fine two stage generation suffers from the issue of early stopping that the generated arguments is incomplete. For instance, the ground-truth step should be `Find`(*coffee table*) while the generated result is `Find`(*coffee*).

J Out-of-Distribution Setting Construction

We consider an Out-of-Domain Generalization (OOD) setting, where test images are related to scenes (i.e., *Indoor*, *Food* and *Street*) not observed during training. For the *Indoor* scene, we directly leverage the annotation from Visual Genome (Kr-

Backbone	Yes/No		Other			
	Verify	Logical	Choose	Compare	Query	Overall
ViT-B/16	69.63	68.63	75.87	48.59	26.36	47.28
Res50× 16	68.51	68.71	75.78	41.84	25.69	46.49
ALBEF	68.08	69.99	75.93	48.40	29.38	48.68

Table 9: Performance of the proposed model with different models for multimodal matching regarding different question types.

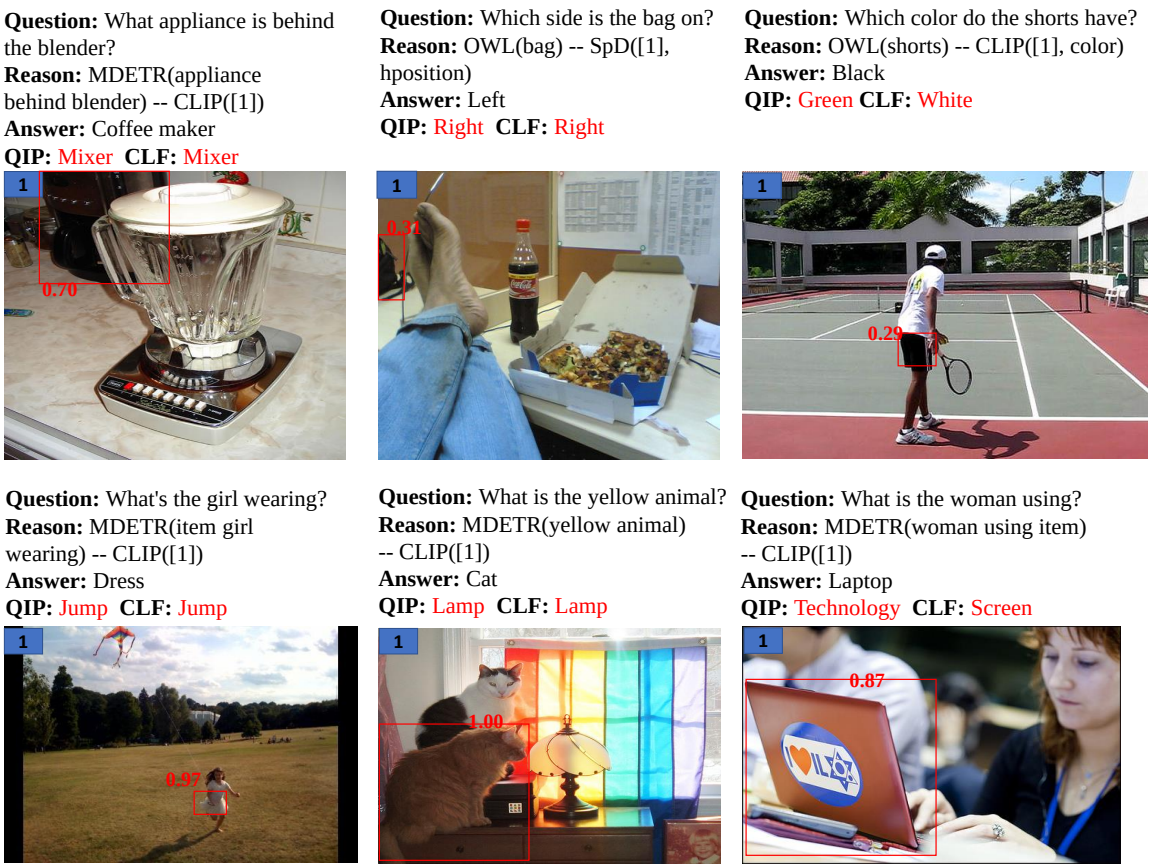

Figure 6: Visualization of VQA examples with short reasoning chains.

ishna et al., 2017), where images are classified as indoors and outdoors. For the other two settings, we filter out training images containing those scene-specific objects and make sure a certain protion of objects in the testing images are about those objects (in other words, testing images are related to the scene). Below, we provide the lists of scene specific objects in the *Food* and *Street* scene.

Food: plate, banana, table, food, pizza, donut, fork, bowl, cheese, napkin, glass, cake, tomato, bread, apple, carrot, knife, broccoli, vegetable, fruit, cup, sauce, orange, spoon, meat, pepper, crust, onion, sandwich, home plate, topping, catcher, tray, lettuce, container, dish, bottle, batter, umpire, frosting, hot dog, egg, chicken, bat, box, mask, paper, mushroom, mug, pitcher, dispenser, liquid, label, bacon, tablecloth, nut, leaf, utensil, salad, hand, crumb, lemon, basket, mound, card, helmet, strawberry, lid, pan, seed, chair, menu, jar, player, sausage, icing, juice, shirt, spinach, sprinkle, dugout, counter, bag, flower, berry, goat, sailboat, uniform, steering wheel, glove, heel, pastry, bubble, finger, sugar, beer, oven, heart, dessert, herb

Street: car, sign, building, pole, letter, tree, tire, road, wheel, sidewalk, bus, train, street, number, door, sky, bike, windshield, truck, street light, motorcycle, leaf, traffic light, roof, ground, post, license plate, arrow, vehicle, fence, cloud, word, grass, wire, van, bicycle, gravel, bush, platform, fire hydrant, house, seat, flag, bag, pavement, step, graffiti, sticker, logo, paint, luggage, cone, chain,

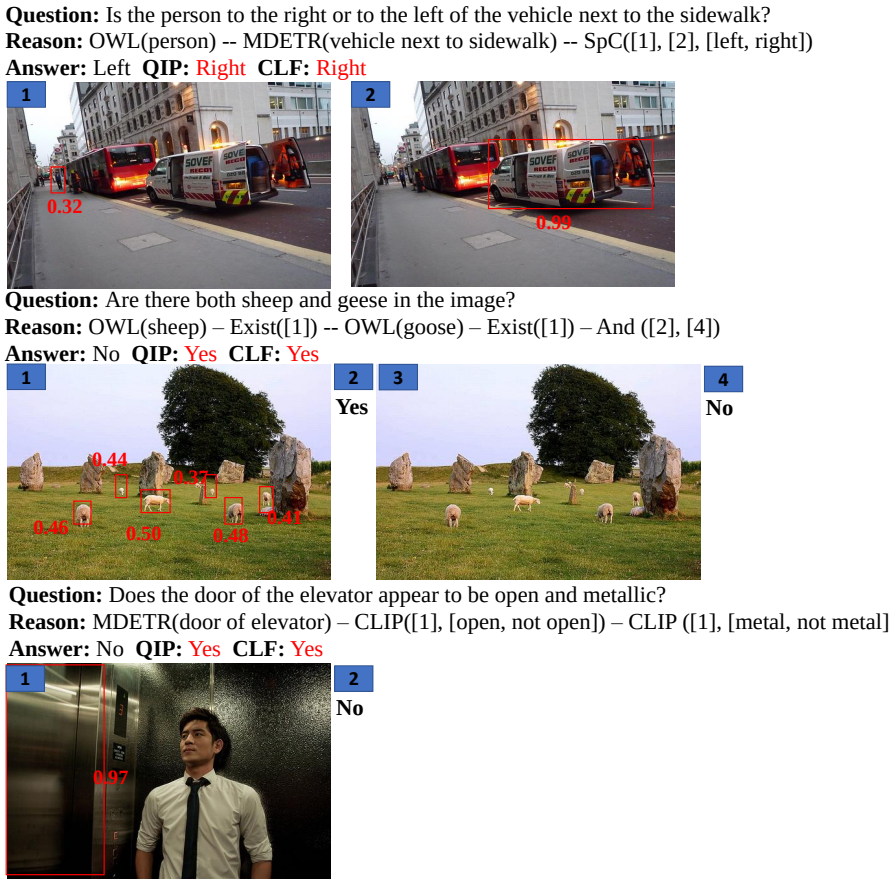

Figure 7: Visualization of VQA examples with long reasoning chains.

pipe, helmet, bridge, balcony, parking lot, jacket, plant, stop sign, train car, umbrella, taxi, lamp, box, crosswalk, flower, bench, brick, store, trash can, clock, gate, station, jean, grill, suv, driver, hook, pant, trash, tower, city, stair, rock, coat, rose, chimney, trailer, american flag, entrance

K Experiment Settings

In this section, we discuss the experiment settings regarding to the size of models, method of choosing hyper-parameters and the used software packages and versions.

Model Size: We provide the number of parameters of different models in Table 10. Our model includes the OWL model, the MDETR model, the $\text{CLIP}_{\text{ViT-B/16}}$ and the T5 model for answer filtering. It consists of $1,521$M parameters, of which the T5 model takes 770M parameters, the OWL model takes 583M parameters, the MDETR model takes 170M parameters and the CLIP model takes 151M parameters. After pre-processing object detection and answer filtering, it takes 6G GPU memory for inference.

Hyper-parameters: As we focus on the zero-shot learning setting so that there is no training process. Here we provide hyper-parameters used as thresholds. For the OWL model (Minderer et al., 2022), we set the threshold of confident score as 0.2, which is set empirically, to filter out detected bounding boxes of which the confident scores are too long. We show test the robustness of the proposed zero-shot VQA model regarding to the hyper-parameter of the threshold and provide experimental results corresponding to the threshold varying from $0.05, , 0.1, 0.15, 0.2, 0.15, 0.3$ in Figure 8. As

Method	# Params (M)
$\text{VL-T5}_{\text{no-vqa}}$	288
$\text{FEWVLM}_{\text{base}}$	288
$\text{FEWVLM}_{\text{large}}$	804
$\text{VLKD}_{\text{ViT-L/14}}$	713
$\text{BNP-VQA}_{\text{6M}}$	669
$\text{BNP-VQA}_{\text{11B}}$	1,576
Frozen	1,040
$\text{QIP}_{\text{ViT-B/16}}$	151
$\text{TAC-P}_{\text{ViT-B/16}}$	921
Zero-shot NMNs	1,521

Table 10: Number of parameters in VQA models.

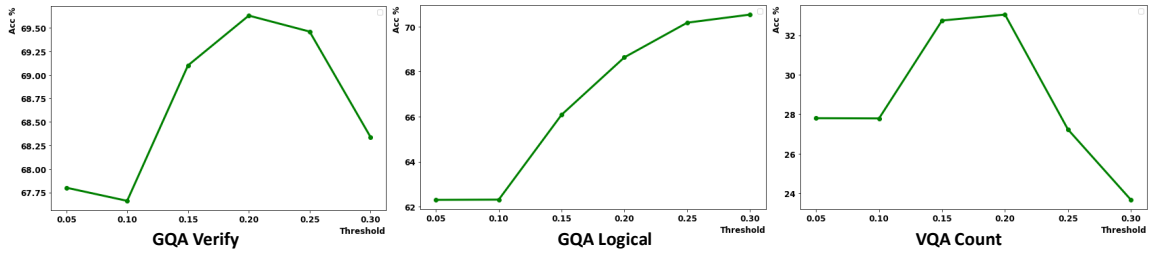

Figure 8: Performances of the zero-shot VQA model regarding to different thresholds of confident scores in the OWL model.

proven in Section 4.4, the detection result mostly affects binary questions which rely more on object detection results, we here provide results for *Verify* and *Logical* type of questions on GQA. Besides, *Count* type questions also heavily rely on the quality of object detection. According to the results, we observe the zero-shot NMNs achieves relatively stable performances regarding to different thresholds for confident scores on *Verify* type questions, while less stable for the *Logical* and *Count* type questions. The stability on *Verify* questions depicts the robustness of the detection model. As *Logical* questions combines results from two *Verify* questions, the error may propagate if one predicted answer of the *Verify* question is wrong. An interesting finding is that the performance does not drop as the threshold increases. This may be that answers are biased to *no*. With the increment of thresholds, the model is more likely to answer *no*. *Count* questions are more sensitive to the threshold because lower thresholds lead to the case that uncertain regions to be detected while higher thresholds are more harmful that correctly detected objects will be filtered out. In conclusion, the threshold is important to the quality of detection and setting it from 0.2 to 0.25 gives good performances. For the MDETR model, we directly follow their published code for detection and set the threshold as 0.7[4].

Package Version: We list the software packages used as well as the corresponding versions in Table 11.

Package	Version)
PyTorch	1.9.0
Transformers	4.19.2
Stanza	1.4.0
NLTK	3.2.5

Table 11: Versions of packages used in our experiments.

[4]https://colab.research.google.com/drive/11xz5IhwqAqHj9-XAIP17yVIuJsLqeYYJ?usp=sharing

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Appendix J

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix J

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix J

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Not applicable. We focus on zero-shot learning settings. No training process is involved and each step is deterministic.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix J

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.