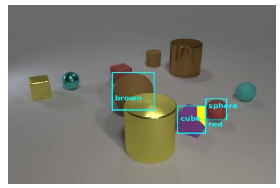
VQA with No Questions-Answers Training

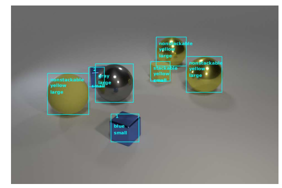
Ben-Zion Vatashsky   Shimon Ullman
Weizmann Institute of Science, Israel
vatashsky@gmail.com, shimon.ullman@weizmann.ac.il

Abstract

Methods for teaching machines to answer visual questions have made significant progress in recent years, but current methods still lack important human capabilities, including integrating new visual classes and concepts in a modular manner, providing explanations for the answers and handling new domains without explicit examples. We propose a novel method that consists of two main parts: generating a question graph representation, and an answering procedure, guided by the abstract structure of the question graph to invoke an extendable set of visual estimators. Training is performed for the language part and the visual part on their own, but unlike existing schemes, the method does not require any training using images with associated questions and answers. This approach is able to handle novel domains (extended question types and new object classes, properties and relations) as long as corresponding visual estimators are available. In addition, it can provide explanations to its answers and suggest alternatives when questions are not grounded in the image. We demonstrate that this approach achieves both high performance and domain extensibility without any questions-answers training.

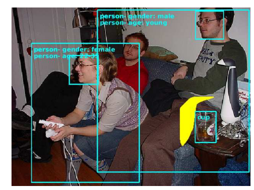


Q: What shape is the object *closest to* the red sphere?
A: cube



Q: How many other objects are the same size as the yellow *stackable* object?
A: 2

Q: Are all the boats white?
A: no [full: There are not enough white boats (failed due to a red boat)]



Q: There is a person that is of a different gender than the young person closest to the cup; how old is he?
A: 22-35

Figure 1. UnCoRd generalizes without QA training to novel properties and relations (top), and to real-world domain (bottom).

1. Introduction

Visual question answering is inspired by the remarkable human ability to answer specific questions on images, which may require analysis of subtle cues, along with the integration of prior knowledge and experience. The learning of new visual classes, properties and relations, can be easily integrated into the question-answering process. Humans can elaborate on the answers they give, explain how they were derived, and why they failed to produce an adequate answer. Current approaches to handle VQA by a machine [67, 62, 55, 77, 32] take a different path, where most answering systems are trained directly to select an answer from common answers of a training set, based on fused image features (mostly using a pre-trained CNN [25]) and question features (mostly using an RNN).

The answering approach we take below is the first, as far as we know, that does not rely on any explicit question-answering training. It uses a process composed according to the question's structure, and applies a sequence of 'visual estimators' for object detection and identifying a set of visual properties and relations. Answering by our 'Understand, Compose and Respond' (UnCoRd) approach is divided into two stages (illustrated in Figure 2). First, a graph representation is generated for the question, in terms of classes, properties and relations, supplemented with quantifiers and logical connectives. An answering procedure then follows the question graph, and seeks either a single or multiple assignments of the classes, properties and relations in the graph to the image (Section 3.3). The method is modular, extensible and uses intermediate results to provide elaborated answers, including alternatives to answers not grounded in the image, and notifying about unsupported categories. With an ability to handle extended domains, the UnCoRd approach demonstrates the potential to build a general answering scheme, not coupled to a specific dataset.

Our work includes several novel contributions. First, a method that produces state-of-the-art results on the CLEVR dataset [30] without any questions-answers training. Second, we developed sequence-to-sequence based method, in-

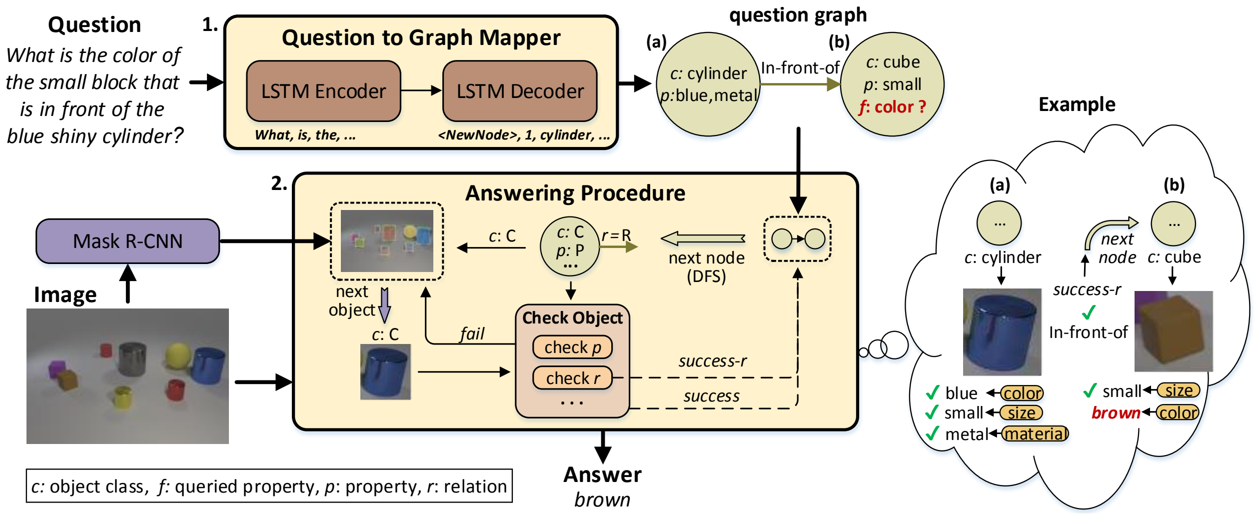


Figure 2. A schematic illustration of our method. The first stage (1) maps the question into a graph representation using a sequence-to-sequence LSTM based model. At the second stage (2), the recursive answering procedure follows the graph, searching for a valid assignment in the image. At each step, the handled node is set and objects (extracted using mask R-CNN) are examined according to the node's requirements (utilizing corresponding visual estimators). If succeeded, a new node is set (according to a DFS traversal) and the function is called again to handle the unassigned subgraph. The Example illustrates the flow: *'check node (a)'* → *'relation success'* → *'check node (b)'* → *answer*.

dependent of images, to map questions into their graph representation. Third, we describe a formalism to represent a broad range of possible questions, with an algorithm that finds valid assignments of a question graph in the image, and provides an answer. Fourth, we present a model that can both perform well on CLEVR, as well as generalize to novel domains by just adding visual estimators (for objects, properties and relations) but without QA examples. Some examples are shown in Figure 1 (elaborated later in text).

2. Related Work

Current answering schemes are dominated by end-to-end methods, trained as multi-class classifiers. Many recent works focused on improving the image-question fused features [17, 9, 84, 10, 16], attention mechanisms for selecting important features [80, 54, 47, 4, 12, 59, 50, 27], including self and guided attention [83, 21, 82], applying pre-trained networks [65, 46, 86], and incorporating outputs of other visual tasks [23, 3, 15, 68, 40, 75, 33, 28]. Some provide reasoning using "facts" extraction (*e.g.* scene type) [73], image caption [42, 1, 41] or by linking visual "facts" with question's logical form [51, 36]. Other integrated external prior knowledge, by generating a query to a knowledge database [72, 14], fusing it in the representation [78, 38], using a textual image description [39] or by added loss terms [66]. The language prior was addressed as well [22, 20, 13, 76, 57].

Some methods use dynamic networks with architecture affected by the question [18, 56]. The Neural Module Networks (NMN) are dynamically composed out of sub-modules. Originally modules arrangement was based on the dependency parsing of the question [6, 5], while later versions used supervised answering program learning [31, 26, 52, 63], including a probabilistic model [71]. Note that the modules are trained only as components of an answering network for a specific dataset and do no function as independent visual estimators. One method [81] performs full scene analysis in order to carry out the program. This method uses questions-answers training to learn the programs, hence cannot be extended by a simple addition of visual estimators. Moreover, performing full scene analysis (detecting all objects, properties and relations in the scene) may become infeasible for data less restricted than CLEVR (especially for relations). In our method, the answering process is guided by the question and does not perform a full scene analysis. It allows a flexible integration of additional visual capabilities (*e.g.* novel object classes), providing elaborated answers and proposing alternatives. These capacities are obtained without requiring any QA examples.

Current methods fit models to particular datasets and exploit inherent biases, which can lead to ignoring parts of the question/image, and to failures on novel domains and rephrasing [2, 60]. In contrast to the modular approach we pursue, any adaptation or upgrade requires a full retraining.

3. Method

3.1. Overview

In the formalism we use, a simple question without quantifiers can be transformed to an assertion about the image that may have free variables (*e.g.*'color' in 'what is the color of...'). The question is answered by finding an assignment to the image that will make the statement true, and retrieving the free variables. The quantifiers derived from the question require multiple true assignments (such as '5', 'all', etc.). The procedure we use seeks the required assignments and returns the desired answer. The answering process consists of two stages (see Figure 2 for a scheme):

1. **Question mapping into a graph representation -** First, a representation of the question as a directed graph is generated, where nodes represent objects and edges represent relations between objects. Graph components include objects classes, properties and relations. The node representation includes all the object visual requirements needed to answer the question, which is a combination of the following (see examples in the supplement, section 1):
   - Object class c (*e.g.*'horse').
   - Object property p (*e.g.*'red').
   - Queried object property f (*e.g.*'color').
   - Queried set property g (*e.g.*'number').
   - Quantifiers (*e.g.*'all', 'two').
   - Quantity relative to another node (*e.g.* same).
   - Node type: regular or SuperNode: OR of nodes (with optional additional requirements).
2. **Answering procedure -** In this stage, a recursive procedure finds valid assignments of the graph in the image. The number of required assignments for each node is determined by its quantifiers. The procedure follows the graph, invoking relevant sub-procedures and integrates the information to provide the answer. Importantly, it depends only on the abstract structure of the question graph, where the particular object classes, properties and relations are parameters, used to apply the corresponding visual estimators (*e.g.* which property to extract). The invoked sub-procedures are selected from a pool of the following *basic procedures*, which are simple visual procedures used to compose the full answering procedure:
   - Detect object of a certain class c.
   - Check the existence of object property p.
   - Return an object property of type f.
   - Return an object's set property of type g.
   - Check the existence of relation r between two objects.

Our construction of a question graph and using its abstract structure to guide the answering procedure leads to our ability to handle novel domains by adding visual estimators but using the same answering procedure. In our method we only train the question-to-graph mappers and the required visual estimators. Unlike QA training, we use independent trainings, which may utilize existing methods and be developed separately. This also simplifies domain extension (*e.g.* automatic modification is simpler for question-graph examples than for question-image-answer examples).

3.2. Question to Graph Mapping

Understanding natural language questions and parsing them to a logical form is a hard problem, still under study [29, 7, 74, 11, 58]. Retrieving question's structure by language parsers was previously performed in visual question answering [6], utilizing the Stanford Parser [34].

We handled the question-to-graph task as a translation problem from natural language questions into a graph representation, training an LSTM based sequence to sequence models [64]. The graph was serialized (using DFS traversal) and represented as a sequence of strings (including special tokens for graph fields), so the model task is to translate the question sequence into the graph sequence (see examples in Section 1 of the supplement). All our models use the architecture of Google's Neural Machine Translation model [79], and are trained using tensorflow implementation [48]. A simple post-processing fixes invalid graphs. The description below starts with a question-to-graph model trained for CLEVR data, and then elaborates on the generation of extended models, trained for extended scopes of questions.

3.2.1 Question-to-Graph for CLEVR Data

Our basic question-to-graph model is for CLEVR questions and categories (3 objects, 12 properties, 4 property types, 4 relations). The graph annotations are based on the CLEVR answering programs [30], corresponding to the dataset's questions. The programs can be described as trees, where nodes are functions performing visual evaluations for object classes, properties and relations. These programs can be transferred to our graph representation, providing annotations for our mappers training. Note that concepts may be mapped to their synonyms (*e.g.*'ball' to 'sphere').

3.2.2 Extended Question-to-Graph Domain

CLEVR questions are limited, both in the used categories and in question types (*e.g.* without quantifiers). To handle questions beyond the CLEVR scope, we trained question-to-graph mappers using modified sets of questions (randomization was shown to enable domain extension [69]). There were two types of modifications: increasing the vocabulary of visual elements (object classes, properties and relations) and adding questions of new types. The vocabulary was expanded by replacing CLEVR visual elements with ones from a larger collection. This operation does not add question examples to the set, but uses the existing examples with replaced visual elements. Note that as this stage deals with question mapping and not question answering, the questions, which are generated automatically, do not have to be meaningful (*e.g.* "What is the age of the water?") as long as they have a proper mapping, preserving the role of each visual element. To guarantee graph-question correspondence a preprocessing is performed where for each concept, all its synonyms are modified to one form. In addition, for each question all appearances of a particular visual element are replaced with the same term. We used three replacement 'modes', each generating a modified dataset by selecting from a corresponding set (real world categories from existing datasets): i) **Minimal:** Most categories are from COCO [43] and VRD [45] (100 objects, 32 properties, 7 property types, 82 relations). ii) **Extended:** 'Minimal' + additional categories, sampled from 'VG' (230 objects, 200 properties, 53 property types, 160 relations). iii) **VG:** The categories of the Visual Genome dataset [35] (65,178 objects, 53,498 properties, 53 property types, 47,448 relations, sampled according to prevalence in the dataset). The categories include many inaccuracies, such as mixed categories (*e.g.*'fat fluffy clouds') and irrelevant concepts (*e.g.* objects: 'there are white'), which adds inconsistency to the mapping.

The second type of question modification increased the variability of questions. We created enhanced question sets where additional examples were added to the sets generated by each replacement mode (including 'None'). These examples include questions where '*same* <*p*>' is replaced with '*different* <*p*>' (where <*p*> is a property), questions with added quantifiers ('all' and numbers) and elemental questions (with and without quantifiers). The elemental questions were defined as existence and count questions for: class, class and property, class and 2 properties, 2 objects and a relation, as well as queries for objects class (in a relation) and property types (including various WH questions).

The words vocabulary we used for training all sets was the same: 56,000 words, composed by the union of the English vocabulary from IWSLT'15 [49] together with all the used object classes, properties and relations. Both the question and the graph representations were based on the same vocabulary, with additional tokens in the graph vocabulary to mark graph nodes and fields (*e.g.* <*NewNode*>, <*p*>).

Different mappers were trained for all the modified sets above. An example of a graph, mapped using the 'Extended-Enhanced' model, as well as the corresponding original question is given in Figure 3. Note that the modified question, although meaningless, has the same structure as the original question and is mapped to the same graph, except for the replaced visual elements and added quantifiers. This means that the same answering procedure will be carried out, fulfilling our intent to apply the same procedure to similar structured questions.

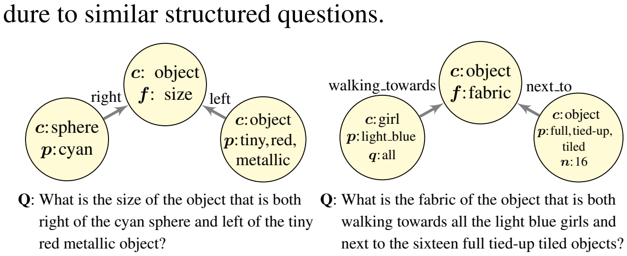


Figure 3. Left: A CLEVR question and a corresponding graph. Right: A modified question and a corresponding graph, mapped using Extended-Enhanced model. The accuracy of the modified representation is confirmed, as it matches the original accurate graph (with modified graph concepts).

3.3. Answering Procedure

In this stage a recursive procedure seeks valid assignments (see Section 3.1) between the question graph and the image. The question graph, the image and the mask R-CNN [24] produced for the image provide the input to the procedure that recursively processes each node (see Figure 2). For each node, basic procedures (see Section 3.1) are invoked sequentially, according to the node's requirements and activate visual estimators according to the particular visual elements. The number of required valid assignments is set by the node's quantifier (a single assignment, a specific number, or all) or by the need of all objects for evaluating the entire object set (*e.g.* counting, number comparisons). The next processed nodes are according to a DFS traversal. Each basic procedure provides an answer, used to produce the final answer, reporting unsupported categories and providing elaborations, based on intermediate results. For more details and examples see Section 2 of the supplement.

3.3.1 CLEVR Visual Estimators

In order to find a valid assignment of a question graph in the image, and provide the answer, corresponding visual estimators need to be trained. Object locations are not explicitly provided for CLEVR data, however they can be automatically recovered using the provided scene annotations. This process provided approximated contour annotations for CLEVR objects (see Figure 4), which were used for training. Mask R-CNN [24] was used for instance segmentation. For property classifiers, simple CNN models (3 convolutional layers and 3 fully connected layers) were trained to classify color and material; size was estimated according to object's bottom coordinates and its largest edge. Relations are classified according to objects' locations.

3.3.2 Real World Visual Estimators

Handling questions in the real-world domain beyond CLEVR objects was performed by utilizing existing visual estimators. For instance segmentation we use a pre-trained mask R-CNN [24] for the 80 classes of COCO dataset [43]. Any other visual estimator may be incorporated to enhance answering capability. In our experiments (Section 4.2.5 and Figure 1) we use color map estimation [70], age and gender classification [37] (utilizing face detection [53]) and depth estimation [44] (utilized for estimating spatial relations).

4. Experiments

The experiments tested the abilities of the UnCoRd system, to first, provide accurate results for the CLEVR dataset and second, to handle extended questions and real-world domains. Our analysis included the two answering stages: creating a correct graph representation of the question, and answering the questions. Adam optimizer was used for question-to-graph and visual estimators training with a learning rate of 10^{-4} (10^{-3} for the 'Extended-Enhanced' model), selected according to the corresponding validation set results. Each model training was using one NVIDIA

Tesla V100 GPU. All reported results are for a single evaluation. For each model, the same version was used in all experiments. Unless stated, system was configured to provide short answers (concise and without elaborations); markings on images in the figures correspond to intermediate results. Code will be available at https://github.com/benyv/uncord

4.1. CLEVR Experiments

We trained a question-to-graph model for CLEVR ('None'-'Basic', as denoted in Section 4.2.1), which generated 100% perfect graphs on its validation set. The visual estimators, described in Section 3.3.1 were also trained and provided the results given in Table 1. CLEVR relations were estimated by simple rules using the objects' coordinates.

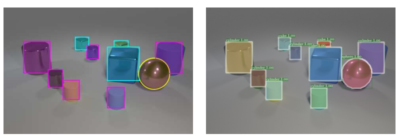


Figure 4. Instance segmentation example for CLEVR data. Left: GT (approximated from scene data), Right: results.

Estimator	$AP^{IoU=.50}$	Acc.
Ins. seg.	99.0	
Color		99.98
Material		99.97
Size		100

Table 1. CLEVR estimators results on CLEVR validation set

We tested the answering performance of the UnCoRd system on the CLEVR test set. The results, including for other state-of-the-art methods (all use answers labels for training) are given in Table 2.

Method	Exist	Count	Comp. Num.	Query Att.	Comp. Att.	Overall test set	Overall val. set
IEP-strong [31]	97.1	92.7	98.7	98.1	98.9	96.9	
FiLM [56]	99.3	94.3	93.4	99.3	99.3	97.6	
DDRprog [63]	98.8	96.5	98.4	99.1	99.0	98.3	
MAC [27]	99.5	97.1	99.1	99.5	99.5	98.9	
TbD [52]	99.2	97.6	99.4	99.5	99.6	99.1	
HAN [50]	99.6	97.2	96.9	99.6	99.6	98.8	
NS-VQA [81][a]	99.9	99.7	99.9	99.8	99.8	-	**99.8**
UnCoRd$_{\text{None-B}}$	**99.89**	**99.54**	**99.91**	**99.74**	**99.80**	**99.74**	**99.8**

Table 2. CLEVR QA accuracy for state-of-the-art methods

[a] Reported for val. set, hence not compared to test set results

As can be seen, our model achieves state-of-the-art results without training for the visual question answering task and not using any answers GT, as other methods. In addition UnCoRd can elaborate and explain answers and failures using intermediate results, and extend the handled domain with no need of images and related QA examples, as demonstrated in Section 4.2 and Figure 6. On a sample of 10,000 validation set examples, all mistakes were due to wrong visual estimators' predictions, mainly miss detection of a highly occluded object. Hence, accurate annotation of object coordinates (as performed in NS-VQA [81]) may even further reduce the small number of errors. Note that NS-VQA requires full scene analysis, which is not scalable for domain extension with a large number of objects and relations. It also uses images with question-answer pairs to train the programs, coupling the method to the specific trained question answering domain.

4.2. Out of Domain Experiments

Next, we test UnCoRd beyond the scope of CLEVR data. We trained question-to-graph models on the modified and enhanced CLEVR data and used corresponding visual estimators. We examined whether domain extension is possible while maintaining a good performance on the original data.

4.2.1 Question to Graph

For evaluating question representation, we trained and tested (see Section 3.2.2) 8 question-to-graph models that include all replacement modes (None, Minimal, Extended, VG), each trained in two forms: Basic (B), *i.e.* no added question examples (~700K examples) and Enhanced (E), *i.e.* with additional examples (~1.4M examples).

In Table 3, we report the results of each trained model on the validation sets of all 8 models, which provides information on generalization across the different sets. Note that as the "None" extension, unlike the data of other models, includes mapping from concepts to their synonyms (see Section 3.2.2), prediction for "None" data by the "Minimal", "Extended" and "VG" models include a preprocessing stage transforming each concept synonyms to a single form.

Train \ Test		None B	None E	Minimal B	Minimal E	Extended B	Extended E	VG B	VG E
None	**B**	100	49.5	0.5	0.2	0.1	0.0	0.1	0.1
	E	99.7	99.8	0.5	0.4	0.1	0.1	0.1	0.1
Minimal	**B**	99.8	48.9	98.4	50.0	0.5	0.3	1.2	0.6
	E	99.0	98.6	98.0	97.7	0.5	1.0	1.1	1.1
Extended	**B**	99.1	48.6	98.2	49.9	96.2	49.1	18.1	9.4
	E	99.1	98.7	97.9	97.5	95.7	95.8	19.3	20.0
VG	**B**	87.5	44.8	65.7	34.6	84.1	45.3	76.9	41.9
	E	90.0	90.0	63.7	64.1	81.9	83.0	75.0	77.1

Table 3. Accuracy of question-to-graph mapping for all data types

Results demonstrate that models perform well on data with lower variability than their training data. The high performance of the 'Extended' models on their corresponding data illustrates that substantial extensions are possible in question-to-graph mapping without requiring any new training images. VG models' lower accuracy is expected due to the unsuitable elements in its data (see Section 3.2.2). Additional tests are required to check possible advantages of VG models for different domains. We report such a test next.

4.2.2 VQA Representation

In this experiment, representation capabilities are tested for a different dataset. Since normally, annotations corresponding to our graph representation are not provided, we sampled 100 questions of the VQA [8] validation set and manually examined the results for the eight question-to-graph models (see Section 4.2.1).

The results in Table 4 express the large gaps in the abilities of models to represent new domains. Models trained specifically on CLEVR do not generalize at all to the untrained domain. As the models are trained on more diverse data, results improve substantially, peaking clearly for VG-Enhanced model by a large margin from other models. This is also evident in the example given in Figure 5 where adequacy of the graph increases in a similar manner. This result is interesting as using this model provides high accuracy for CLEVR as well (see Table 5). The fact that substantial performance gain is achieved for a data domain that was not used in training (the VQA dataset domain), while preserving good results on the original data (CLEVR), demonstrates the potential of the approach to provide a general answering system for visual questions. Further investigation is required for means to enrich question description examples and produce further significant improvements.

None		Minimal		Extended		VG	
B	E	B	E	B	E	B	E
1	0	12	12	22	22	34	50

Table 4. Accuracy of graph representation for VQA [8] sample, given for the different UnCoRd mappers. As expected, training on more diverse data allows better generalization across domains.

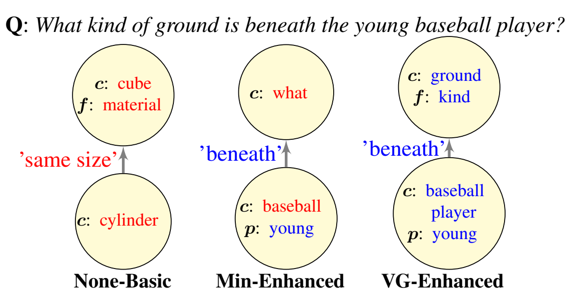


Figure 5. Generated graphs for a free form question (from the VQA [8] dataset). Blue text: accurate concepts, red: inaccurate.

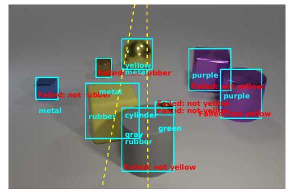
4.2.3 Maintaining Performance on CLEVR Questions

We evaluated the performance change for the CLEVR test set, as the training data variability of the question-to-graph models increases. The results are given in Table 5.

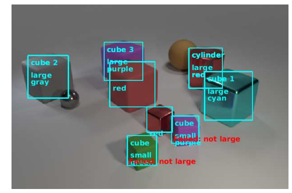
Mapper		Exist	Count	Comp. Num.	Query Att.	Comp. Att.	Overall
None	B	**99.89**	**99.54**	**99.91**	**99.74**	**99.80**	**99.74**
	E	**99.89**	**99.54**	**99.91**	**99.74**	**99.80**	**99.74**
Min	B	99.81	99.36	99.87	99.73	**99.80**	99.68
	E	99.69	99.21	99.47	99.46	99.59	99.46
Ext	B	96.82	89.34	78.64	99.40	99.41	94.80
	E	99.78	99.33	98.36	99.65	99.76	99.49
VG	B	96.82	89.34	78.64	99.44	99.41	94.81
	E	98.03	97.39	96.88	97.62	97.22	97.49

Table 5. Accuracy of CLEVR dataset question answering by UnCoRd using the different question-to-graph mappers

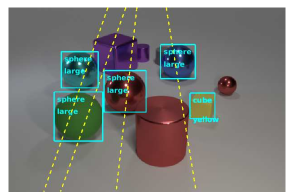
It is evident that even models that were trained on a much larger vocabulary and question types than the original CLEVR data still perform well, mostly with only minor accuracy reduction. This demonstrates that with more variable training we can handle more complex questions, while maintaining good results on the simpler domains. Examples on CLEVR images for both CLEVR questions and others are shown in Figure 6 (using 'None-Enhanced' mapper).



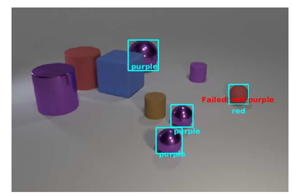
(a) **Q**: There is a yellow thing to the right of the rubber thing on the left side of the gray rubber cylinder; what is its material?
A: metal



(b) **Q**: How many big cubes are of a *different* color than the large cylinder?
A: 3



(c) **Q**: What color is the cube to the right of the *four* big spheres?
A: yellow



(d) **Q**: *Are all the spheres purple?*
A: no [full: There are not enough purple spheres (failed due to a red sphere)]

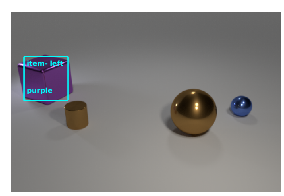
Figure 6. Examples for answering different question types on CLEVR images: (a) taken from CLEVR, (b) includes 'different color' relation, (c) uses a quantifier, and (d) a simple property existence (+ 'all' quantifier) question.

4.2.4 CLEVR Humans

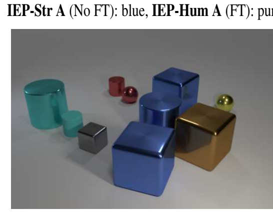
An example of using the CLEVR images with different questions is the CLEVR-Humans [31] (7145 questions in test set), where people were asked to provide challenging questions for CLEVR images. The questions vary in phrasing and in the required prior knowledge.

Method		No FT	FT
IEP-18k		54.0	66.6
FiLM		56.6	75.9
MAC		57.4	**81.5**
NS-VQA		-	67.0
UnCoRd-None	**B**	60.46	
	E	**60.59**	
UnCoRd-Min	**B**	48.24	
	E	52.23	
UnCoRd-Ext	**B**	43.97	
	E	52.83	
UnCoRd-VG	**B**	43.47	
	E	48.71	

Table 6. Question answering accuracy of CLEVR-Humans test set for state-of-the-art methods, with and without finetuning (FT).



Q: What color is the item to the far left? *(GT: purple)*
None-E A: brown, **VG-E A**: purple
IEP-Str A (No FT): blue, **IEP-Hum A** (FT): purple



Q: How many of these things could be stacked on top of each other? *(GT: 8)*
None-E A: 1, **VG-E A**: Unknown class: each other
IEP-Str A (FT): 0, **IEP-Hum A** (FT): 2

Figure 7. Examples for CLEVR-Humans questions

Results, given in Table 6, demonstrate that for models without finetuning, our 'None-Enhanced' model provides state-of-the-art results (without any answer examples). For all models, questions with phrasing not included in training are prone to errors, including 'hallucinations' of concepts. Note that CLEVR-Humans answers are the same answers as in CLEVR (by instructions to workers), hence models biased towards CLEVR (the "None" models) have a better success chances. Models with a rich vocabulary may capture the question graph more accurately, but that may include concepts with no corresponding visual estimators, resulting with answers such as: "Unknown relation 'in between'". Adding such visual estimators will improve performance. Since accuracy calculation does not reward for such limitation indications, just "guessing" the answer would increase the computed accuracy, especially as success chances rise with a simple question categorization (*e.g.* 50% for yes/no and size questions). However, indicating limitations gives a better sense of the system's level of understanding the question, and can lead to corrective actions. Such answers can be promoted in QA systems, by reducing "score" for wrong answers, or giving partial scores to answers identifying a missing component.

Examples of CLEVR-Humans questions are given in Figure 7. It is evident that the more general model (VG-Enhanced) can perform on out of scope questions (top) and report limitations (bottom).

4.2.5 Extensibility to Real-World Images

The UnCoRd system can be naturally extended to novel domains by a simple plug-in of visual estimators. This is illustrated in Figure 1 for using new properties/relations and for an entirely different domain of real-world images. An experiment that adds questions with a novel property is presented in Section 3 of the supplement. We next describe an experiment for real-world images, where we use real world visual estimators (see Section 3.3.2) and our most general trained mapper (VG-Enhanced). We compare our model to Pythia [85], which has top performance on the VQA v2 dataset [19]. The experiment includes two parts:

1. 'Non VQA_v2' questions: 100 questions outside Pythia's training domain (VQA v2), with unambiguous answers, on 50 COCO images (two similar questions per image with different answers). We freely generated questions to include one or more of the following categories:
   - A combination of properties and relations requirements linked by logical connectives ('and', 'or').
   - Property comparison (*e.g.*'same color').
   - Quantifiers (*e.g.*'all', 'five').
   - Quantity comparison (*e.g.*'fewer', 'more').
   - A chain of over two objects connected by relations.
2. 'VQA_v2' questions: 100 questions sampled from VQA v2 dataset [19] with terms that have visual estimators in UnCoRd and unambiguous answers (annotated by us).

In addition to the estimators mentioned in Section 3.3.2, ConceptNet [61] is used by UnCoRd to query for optional classes when superordinate groups are used (*e.g.*'animals'). More details are in Section 4 of the supplement.

The non VQA_v2 results, given in Table 7, demonstrate the substantial advantage of UnCoRd for these types of questions. All UnCoRd's failures are due to wrong results of the invoked visual estimators. Note the substantial performance difference in Pythia between yes/no and WH questions, unlike the moderate difference in UnCoRd. We found that Pythia recognizes the yes/no group (*i.e.* answers 'yes'/'no'), but its accuracy (56%) is close to chance level (50%). Examples of successful UnCoRd answers to the non VQA_v2 questions are provided in Figure 8, while failure examples, including failure sources, are shown in Figure 9. Pythia's answers are given as well.

Method	Yes/No	WH	Overall
Pythia [85]	56.0	14.0	35.0
UnCoRd-VG-E	**88.0**	**64.0**	**76.0**

Table 7. Answering accuracy for 100 questions outside the VQA v2 domain (including quantifiers, comparisons, multiple relation chains and multiple relations and properties) on COCO images.

Q: How many cell phones are left of the red cell phone that is closest to the right cell phone?
UnCoRd A: 9, **Pythia A**: 4
Q: How many cell phones are left of the right cell phone?
UnCoRd A: 11, **Pythia A**: 5

Q: What object is supporting the person that is left of the person above the skateboard?
UnCoRd A: bicycle, **Pythia A**: skateboard
Q: What thing is on an object that is left of the person above the skateboard?
UnCoRd A: person, **Pythia A**: skateboard

Q:Is the number of people that are to the right of the left ball the same as the number of balls?
UnCoRd A: no, **Pythia A**: no
Q: Is the number of people that are to the right of the left ball greater than the number of balls?
UnCoRd A: yes, **Pythia A**: no

Q: What color is the suitcase that is both below a blue suitcase and left of a suitcase?
UnCoRd A: red, **Pythia A**: blue
Q: What color is the suitcase that is both below a blue suitcase and right of a suitcase?
UnCoRd A: orange, **Pythia A**: blue

Figure 8. Examples of UnCoRd successes in answering questions outside the VQA v2 domain on COCO images.

Results for the 100 VQA_v2 questions are given in Table 8. As can be seen, UnCoRd's results are better by a large margin, compared to Pythia [85] end-to-end model, even

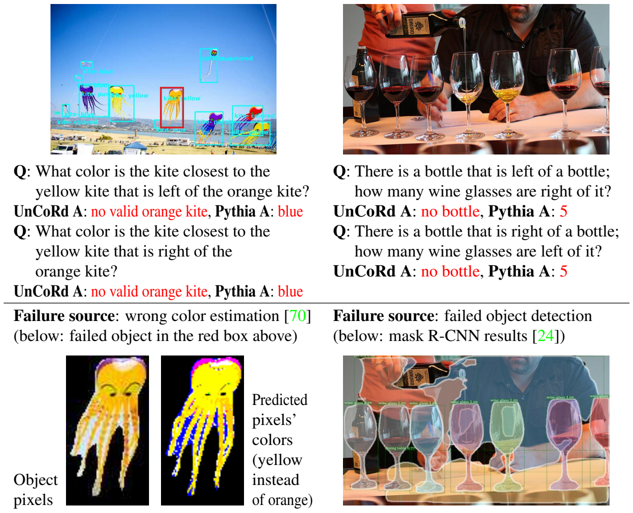


Figure 9. Examples of UnCoRd failures in answering questions outside the VQA v2 domain on COCO images.

though questions were sampled from VQA v2, a dataset used for Pythia's training. As in the previous part, all UnCoRd's failures are only due to wrong results of the invoked visual estimators. Examples of UnCoRd's answers for the VQA_v2 questions are given in Figure 10, including the corresponding answers of Pythia.

Method	Yes/No	WH	Overall
Pythia [85]	90.0	68.3	77.0
UnCoRd-VG-E	**97.5**	**88.3**	**92.0**

Table 8. Answering accuracy for 100 questions sampled from VQA v2 dataset (on terms with visual estimators in UnCoRd).

The above experiments on real-world images show that when corresponding visual estimators are available, our method performs better than a leading end-to-end model, both for questions outside the training domain of the end-to-end model (where the advantage is substantial) and for questions from this domain. This is achieved without any question answering training.

5. Conclusions and Future Directions

We proposed a novel approach to answer visual questions by combining a language step, which maps the question into a graph representation, with a novel algorithm that maps the question graph into an answering procedure. Because the algorithm uses the abstract structure of this graph, it allows a transfer to entirely different domains. Training is performed for the language step to learn the graph representation, and for the visual step to train visual estimators. However, unlike existing schemes, our method does not use images and associated question-answer pairs for training. Our approach allows handling novel domains provided that corresponding visual estimators are available. The combination of the question graph and answering procedure

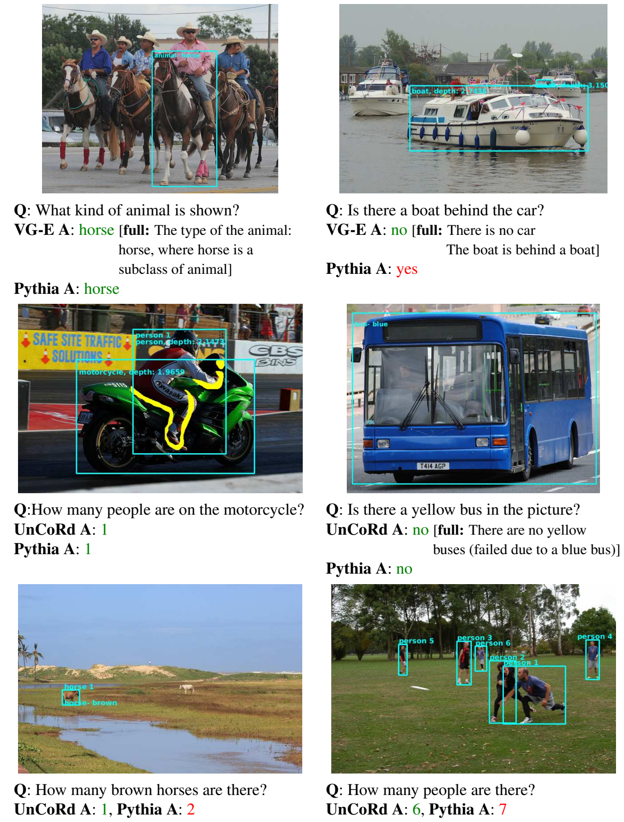


Figure 10. Examples of UnCoRd answers to VQA v2 questions (including 'full' answers when they add information).

also gives the method some capacity to explain its answers and suggest alternatives when question is not grounded in the image. Based on this approach, our answering system achieves near perfect results on a challenging dataset, without using any question-answer examples. We have demonstrated that question representation and answering capabilities can be extended outside the scope of the data used in training, preserving good results for the original domain.

Substantial work is required to obtain a system that will be able to perform well on entirely general images and questions. The main immediate bottleneck is obtaining question-to-graph mapping with general representation capabilities for a broad range of questions. Question graph representation may also be enhanced to support questions with more complex logic, as well as extending the scope of the supported visual categories (*e.g.* global scene types). Any general VQA requires vast estimation capabilities, as any visual category can be queried. In UnCoRd they are modularly incremented and automatically integrated with existing questions. Additional basic areas that current schemes, including ours, have only begun to address, are the use of external, non-visual knowledge in the answering process, and the composition of detailed, informative answers, integrating the language and visual aspects of VQA.

Acknowledgements: This work was supported by EU Horizon 2020 Framework 785907 and ISF grant 320/16.

References

[1] Somak Aditya, Yezhou Yang, and Chitta Baral. Explicit reasoning over end-to-end neural architectures for visual question answering. In *AAAI*, 2018.

[2] Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, Texas, USA, 2016.

[3] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

[5] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2016.

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

[7] Jacob Andreas, Andreas Vlachos, and Stephen Clark. Semantic parsing as machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 47–52, 2013.

[8] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *International Conference on Computer Vision (ICCV)*, 2015.

[9] Hedi Ben-younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[10] Hedi Ben-Younes, Remi Cadene, Nicolas Thome, and Matthieu Cord. Block: Bilinear superdiagonal fusion for visual question answering and visual relationship detection. In *AAAI 2019-33rd AAAI Conference on Artificial Intelligence*, 2019.

[11] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on Freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[12] Remi Cadene, Hedi Ben-Younes, Nicolas Thome, and Matthieu Cord. Murel: Multimodal Relational Reasoning for Visual Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2019.

[13] Remi Cadene, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. Rubi: Reducing unimodal biases in visual question answering. *arXiv preprint arXiv:1906.10169*, 2019.

[14] Qingxing Cao, Bailin Li, Xiaodan Liang, and Liang Lin. Explainable high-order visual question reasoning: A new benchmark and knowledge-routed network. *arXiv preprint arXiv:1909.10128*, 2019.

[15] Mikyas T Desta, Larry Chen, and Tomasz Kornuta. Object-based reasoning in vqa. In *Winter Conference on Applications of Computer Vision, WACV*. IEEE, 2018.

[16] Tuong Do, Thanh-Toan Do, Huy Tran, Erman Tjiputra, and Quang D. Tran. Compact trilinear interaction for visual question answering. In *ICCV*, 2019.

[17] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, Texas, USA, 2016.

[18] Peng Gao, Hongsheng Li, Shuang Li, Pan Lu, Yikang Li, Steven CH Hoi, and Xiaogang Wang. Question-guided hybrid convolution for visual question answering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[19] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] Gabriel Grand and Yonatan Belinkov. Adversarial regularization for visual question answering: Strengths, shortcomings, and side effects. In *Proceedings of the Second Workshop on Shortcomings in Vision and Language*, pages 1–13, 2019.

[21] Dalu Guo, Chang Xu, and Dacheng Tao. Graph reasoning networks for visual question answering. *arXiv preprint arXiv:1907.09815*, 2019.

[22] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Yibing Liu, Yinglong Wang, and Mohan Kankanhalli. Quantifying and alleviating the language prior problem in visual question answering. *arXiv preprint arXiv:1905.04877*, 2019.

[23] Tanmay Gupta, Kevin Shih, Saurabh Singh, and Derek Hoiem. Aligned image-word representations improve inductive transfer across vision-language tasks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[26] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 804–813, 2017.

[27] Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations (ICLR)*, 2018.

[28] Drew A Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. *arXiv preprint arXiv:1907.03950*, 2019.

[29] Robin Jia and Percy Liang. Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622*, 2016.

[30] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[31] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, 2017.

[32] Kushal Kafle and Christopher Kanan. Visual question answering: Datasets, algorithms, and future challenges, 2016.

[33] Hyounghun Kim and Mohit Bansal. Improving visual question answering by referring to generated paragraph captions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[34] Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

[35] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[36] Jayant Krishnamurthy and Thomas Kollar. Jointly learning to parse and perceive: Connecting natural language to the physical world. *TACL*, 1:193–206, 2013.

[37] Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) workshops*, June 2015.

[38] Guohao Li, Hang Su, and Wenwu Zhu. Incorporating external knowledge to answer open-domain visual questions with dynamic memory networks, 2017.

[39] Hui Li, Peng Wang, Chunhua Shen, and Anton van den Hengel. Visual question answering as reading comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6319–6328, 2019.

[40] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. *arXiv preprint arXiv:1903.12314*, 2019.

[41] Qing Li, Jianlong Fu, Dongfei Yu, Tao Mei, and Jiebo Luo. Tell-and-answer: Towards explainable visual question answering using attributes and captions, 2018.

[42] Qing Li, Qingyi Tao, Shafiq Joty, Jianfei Cai, and Jiebo Luo. Vqa-e: Explaining, elaborating, and enhancing your answers for visual questions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*. 2014.

[44] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2016.

[45] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 2016.

[46] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

[47] Pan Lu, Hongsheng Li, Wei Zhang, Jianyong Wang, and Xiaogang Wang. Co-attending free-form regions and detections with multi-modal multiplicative feature embedding for visual question answering. In *AAAI*, 2018.

[48] Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. Neural machine translation (seq2seq) tutorial, 2017.

[49] Minh-Thang Luong and Christopher D. Manning. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*, Da Nang, Vietnam, 2015.

[50] Mateusz Malinowski, Carl Doersch, Adam Santoro, and Peter Battaglia. Learning visual question answering by bootstrapping hard attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[51] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Neural Information Processing Systems (NIPS)*, pages 1682–1690, 2014.

[52] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4942–4950, 2018.

[53] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *ECCV*, 2014.

[54] Duy-Kien Nguyen and Takayuki Okatani. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6087–6096, 2018.

[55] Supriya Pandhre and Shagun Sodhani. Survey of recent advances in visual question answering, 2017.

[56] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

[57] Arijit Ray, Karan Sikka, Ajay Divakaran, Stefan Lee, and Giedrius Burachas. Sunny and dark outside?! improving consistency in vqa through entailed question generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[58] Michael Ringgaard, Rahul Gupta, and Fernando CN Pereira. Sling: A framework for frame semantic parsing. *arXiv preprint arXiv:1710.07032*, 2017.

[59] Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Dhruv Batra, and Devi Parikh. Taking a hint: Leveraging explanations to make vision and language models more grounded. *arXiv preprint arXiv:1902.03751*, 2019.

[60] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. *arXiv preprint arXiv:1902.05660*, 2019.

[61] Robert Speer and Catherine Havasi. Conceptnet 5: A large semantic network for relational knowledge. In *The People's Web Meets NLP: Collaboratively Constructed Language Resources*, pages 161–176. Springer Berlin Heidelberg, 2013.

[62] Yash Srivastava, Vaishnav Murali, Shiv Ram Dubey, and Snehasis Mukherjee. Visual question answering using deep learning: A survey and performance analysis. *arXiv preprint arXiv:1909.01860*, 2019.

[63] Joseph Suarez, Justin Johnson, and Fei-Fei Li. Ddrprog: A clevr differentiable dynamic reasoning programmer. *arXiv preprint arXiv:1803.11361*, 2018.

[64] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[65] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[66] Damien Teney, Ehsan Abbasnejad, and Anton Hengel. On incorporating semantic prior knowlegde in deep learning through embedding-space constraints. *arXiv preprint arXiv:1909.13471*, 2019.

[67] Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge, 2017.

[68] Damien Teney, Lingqiao Liu, and Anton van den Hengel. Graph-structured representations for visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[69] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2017.

[70] Joost Van De Weijer, Cordelia Schmid, and Jakob Verbeek. Learning color names from real-world images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[71] Ramakrishna Vedantam, Karan Desai, Stefan Lee, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Probabilistic neural-symbolic models for interpretable visual question answering. *arXiv preprint arXiv:1902.07864*, 2019.

[72] Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel, and Anthony Dick. FVQA: Fact-based visual question answering, 2016.

[73] Peng Wang, Qi Wu, Chunhua Shen, and Anton van den Hengel. The vqa-machine: Learning how to use existing vision algorithms to answer new questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1173–1182, 2017.

[74] Yushi Wang, Jonathan Berant, and Percy Liang. Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1332–1342, 2015.

[75] Jialin Wu, Zeyuan Hu, and Raymond J. Mooney. Generating question relevant captions to aid visual question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[76] Jialin Wu and Raymond J Mooney. Self-critical reasoning for robust visual question answering. *arXiv preprint arXiv:1905.09998*, 2019.

[77] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Visual question answering: A survey of methods and datasets, 2016.

[78] Qi Wu, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[79] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

[80] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[81] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B Tenenbaum. Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[82] Zhou Yu, Yuhao Cui, Jun Yu, Dacheng Tao, and Qi Tian. Multimodal unified attention networks for vision-and-language interactions. *arXiv preprint arXiv:1908.04107*, 2019.

[83] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6281–6290, 2019.

[84] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *Proc. IEEE Int. Conf. Comp. Vis*, volume 3, 2017.

[85] Yu Jiang*, Vivek Natarajan*, Xinlei Chen*, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0.1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.

[86] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. *arXiv preprint arXiv:1909.11059*, 2019.

Supplementary Material

1. Examples of Question Graph Generation

Following are some examples of questions and their corresponding generated graphs. Every graph is given both in the format of a string sequence (as produced by the sequence-to-sequence model) and as the equivalent question graph. Each example is taken from a different data source. The examples are given from simplest graph (a single node) to larger graphs.

The special tokens used in the string sequences of the graphs are:

<NewNode>: Indicates a new node, followed by the node index and the object class (c in the graph figures) strings. The following strings relate to fields of this node until the next *<NewNode>*.

<p>: Indicates a required property, followed by the required property.

<F>: Indicates a queried property (f in the graph figures), followed by the queried property type.

<N>: Indicates a queried property of a set of objects (g in the graph figures), followed by the queried property type.

<rd>: Indicates a relation with a child node, followed by the required relation and the index of the corresponding child node.

<rp>: Indicates a relation with a parent node, followed by the required relation and the index of the corresponding parent node.

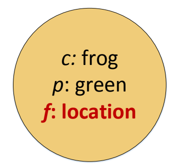
<nodeType>: Indicates the node type field, followed by the node type ('regular' or 'superNode'). Note that the default value is 'regular', so the field is provided only for super nodes.

<nodes>: Indicates an included sub node, followed by the index of the included sub node. This field is only relevant and provided for super nodes.

<is_plural>: Indicates whether the object is given in plural form, followed by the corresponding boolean value (used only for phrasing the 'full' answers). The default is *'false'*.

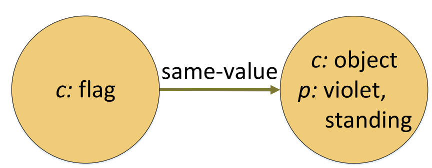
The graph symbols (in the diagrams): c: object class, p: property, f: queried property, g: queried property of a set of objects

- **VQA dataset:**
  The following example is taken from the VQA [1] validation set. The corresponding question graph is generated by the 'VG-Enhanced' question-to-graph model.
  Question:
  Where is the green frog?
  Graph String Sequence:
  <NewNode>, 1, frog, *<p>*, green, *<F>*, location
  Question Graph:



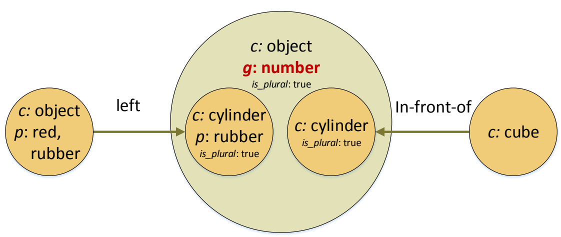
  Note: 'Where' was mapped to the location property; the question is mapped to a single node.

- **Extended CLEVR data:**
  The following example is based on a CLEVR question, modified by replacing CLEVR visual elements with corresponding ones, sampled from the 'Extended' set. The corresponding question graph is generated by the 'Extended-Enhanced' question-to-graph model.
  Question:
  Does the flag have the same value as the violet standing thing?
  Graph String Sequence:
  <NewNode>, 1, flag, *<rd>*, same value, 2, *<NewNode>*, 2, object, *<p>*, violet, *<p>*, standing, *<rp>*, same value, 1
  Question Graph:



  Note:
  - To generate a graph, the question must be syntactically correct, but it is not required to have a visual sense or even to have a reasonable meaning.
  - The relation in the question graph includes property comparison; this type of relations (for properties with corresponding visual estimators) is handled by the answering procedure.

- **CLEVR data:**
  The following example is taken from the CLEVR validation set [5]. The corresponding question graph is generated by the 'None-Basic' question-to-graph model (which includes mapping of concepts to their synonyms).
  Question:
  What number of objects are matte cylinders that are on the left side of the red matte object or cylinders in front of the block?
  Graph String Sequence:
  <NewNode>, 1, object, *<p>*, red, *<p>*, rubber, *<rd>*, left, 2, *<NewNode>*, 2, cylinder, *<p>*, rubber, *<is_plural>*, true, *<rp>*, left, 1, *<NewNode>*, 3, cube, *<rd>*, front, 4, *<NewNode>*, 4, cylinder, *<is_plural>*, true, *<rp>*, front, 3, *<NewNode>*, 5, object, *<nodeType>*, superNode, *<is_plural>*, true, *<nodes>*, 2, *<nodes>*, 4, *<N>*, number
  Question Graph:



  Note: the large circle is a super node, which handles the 'OR' connective; 'matte' and 'block' were mapped to their corresponding basic terms: 'rubber' and 'cube', respectively.

2. Answering Procedure Algorithm

The answering procedure is a recursive procedure guided by the question graph to provide an answer. The input is the question graph and the image, and the output is the produced answer, possibly combined with elaborations and explanations. The procedure follows the graph nodes and edges, invokes the relevant basic procedures and integrates all the information to provide the answer. A schematic description of the procedure is given in Algorithm 1 below.

The first step is an instance segmentation, carried out by applying mask R-CNN [3] to the image. Next, a recursive function ($getGraphAnswer$) is invoked for node handling (starting at a root node of a subgraph, if available, and at an arbitrary node otherwise). It runs procedures from the set of basic procedures, which activate visual estimators to check the requirements (properties, relations), and fetch required information (queried property). The retrieved objects from the mask R-CNN that fulfill the requirements are paired with the corresponding question objects (a 'working memory' module is used to store and share this information), so that subsequent tests will be applied to the correct objects. The number of required objects is set according to quantifiers (*e.g.* 'all', 'three') or by the need to evaluate a property that depends on the entire object set (*e.g.* 'how many'). If a node is a 'sub node', i.e. included in a 'super node', all valid objects are added as optional objects for its corresponding 'super node'. Once all the node's object tests are completed (not including set requirements), the same function ($getGraphAnswer$) is invoked for the next node, determined by a DFS traversal. After the node and the following nodes in the recursion are tested and validated in the image, tests are applied, if needed, to verify the node's set requirements. Examples for such tests are counting the objects in the set and quantity comparisons. Note that tests of relations may include property comparisons, *e.g. 'same size'*. The recursive process is terminated either when the full graph is grounded in the image as required or when no alternatives are left to check, which may happen after a partial check (*e.g.* no objects of a required class were detected).

The last stage is generating the final answer. Partial answers are provided by the basic procedures invoked during the graph traversal, in the form of text sequences. The final answer is produced from these partial answers, mostly based on selecting one of them. The provided answers can be configured to be short or full. Short answers include only the required information, which can be "yes", "no", object class, a particular property, a number, or reporting a failure such as "unknown class: 'logo'". The full answers are based on the partial answers, which use fixed, short templates, for example "Yes, there is a $<c_1>$ $<r>$ a $<c_2>$", where $<r>$ is the relation, and $<c_1>$ and $<c_2>$ are the participating object classes. Additional explanations and elaborations are added (if relevant) to the full answer based on available intermediate results, as well as additional processing for checking alternatives.

An important aspect of the answering procedure is that it depends *only on the abstract structure of the question graph*, and not on the particular object classes, properties and relations. The particular visual elements are parameters given to the procedure and used to invoke the corresponding visual estimators. Hence, graphs with the same abstract structure, such as the example in Figure 1, induce the same answering procedure. This allows generalization to totally different domains that include novel objects, properties and relations.

The utilized visual estimators perform real-world tasks, corresponding to the genuine question's requirements. Hence, the intermediate results of the answering procedure can be exploited for other related tasks. Objects are grounded in the image according to the requirements of the question (see markings of intermediate results on images in Figures 2, 3, 4 and in the main paper). This means that the referring expressions comprehension task [11, 4, 10] is carried out during the answering process, where the referring ex-

Algorithm 1: Answering procedure

```
Input: question_graph, image
Result: Answer to question
initialization: run instance segmentation (mask R-CNN [3]),
 workMem.current_node = first_root_node;
Run[success, answer] = getGraphAnswer :
begin
    current_node = workMem.current_node;
    Node parameters: p: properties, r: relations, f: queried
     property, g: queried property of a set, obj: candidate objects^a;
    for obj in obj do
        if ¬ empty(p) then
            for p in p do
                [success, answer] = is_p(obj);
                if ¬success then break end
            end
            if ¬ success then
                if #possible_objs <#required_objs^b then
                    return [success, answer]
                else
                    continue
                end
            end
        end
        if ¬empty(f) then answer = get_f(obj); end
        if empty(r) then
            if exist(next_root_node) then
                workMem.current_node = next_root_node;
                Run [success, answer] = getGraphAnswer;
            end
        else
            for r in r do
                for child_obj in child_objs^c do
                    [success, answer] = is_r(child_obj, obj);
                    if success then
                        workMem.current_node = next_node^d;
                        Run [success, answer] = getGraphAnswer;
                        if success ∧ (#success_child_objs ==
                         #required_child_objs^b) then break end
                    end
                    if ¬success ∧ (#possible_child_objs <
                     #required_child_objs^b) then break end
                end
                if ¬success then break end
            end
        end
        if success then break end
    end
    if success ∧ ¬empty(g) then answer = get_g(valid_objs); end
    if success ∧ comp_num_en ∧ is_checked(comp_node) then
        answer = comp_num^e(valid_objs, comp_objs, comp_type);
    end
    if success ∧ is_sub_node then
     save_for_super_node(success_objs); end
    return [success, answer]
end
```

[a] According to instance segmentation and previous checks. If 'super node': candidate objects are from 'sub nodes'
[b] According to quantifiers
[c] Candidate objects for child nodes
[d] Either child node or next unvisited root node of a subgraph
[e] Compare number of valid objects between nodes ('same', 'fewer', 'more')

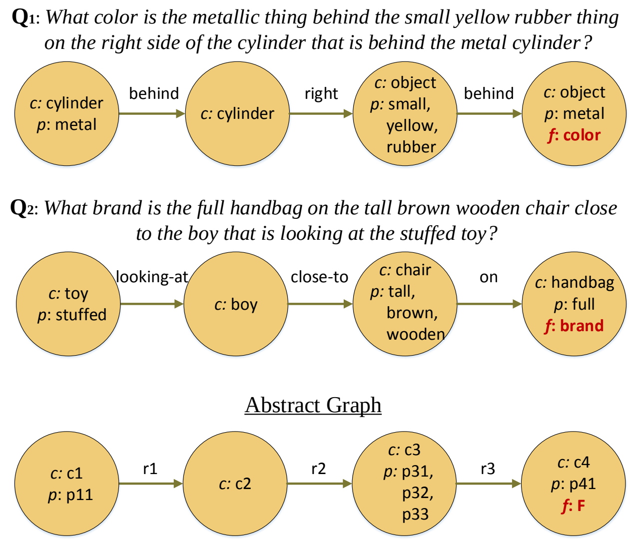


Figure 1. Both questions in the two top panels share the same abstract structure, represented by the graph in the bottom panel, where visual elements are parameters. All questions represented by this abstract graph share the same answering procedure, allowing generalization to entirely new domains with novel visual elements.

pressions are the questions (but also parts of the questions). Performing the grounding using standard referring expressions instead of the questions can be carried out by simple adaptations. Also, saved results of the checked objects can be used to enable follow-up questions and carrying out the visual dialog task [2, 7].

Any estimator can be integrated in our system for any required object class, property, relation or related prediction. In addition to the main visual estimators mentioned in the paper for the conducted experiments, our system employs general rule based estimators for spatial relations and properties (*e.g.* the relation 'on' and the property 'the left').

3. Extended Domain Experiment

In this experiment we test methods trained on CLEVR data (for UnCoRd, we use the 'None-Basic' question-to-graph mapper) using questions with a new property type. We use the 'stackability' properties ('stackable', 'nonstackable'), which refer to the ability to put objects on top of each other (a property used by humans in the CLEVR-Humans dataset [6]). Our approach can handle such extensions in a straightforward manner, using two steps:

1. A question modification step, where the new property in the question is replaced with another property (that does not appear in the question, but was used in the original dataset). In the corresponding graph, produced by the mapper, this property is replaced back to the original term. Note that this step is required as we use the simplest mapper (no exploitation of extended training for fairness).

2. Add a 'stackability' classifier. Two classification options were tested: a trained classifier (same architecture as our CLEVR property classifiers, with a classification accuracy of 99.98%) and a direct inference as an inherent property of the object class, without any training (that is, objects with flat bottom and top such as cubes and cylinders are 'stackable' where spheres are 'nonstackable').

We compare our method to TbD [12], which is trained with question-answer examples and hence it is not possible to plug-in a property classifier. However, to have a fair comparison, we apply finetuning to the system, but limited to a binary classification of the new property, rather than full QA training. The first step was performed in a similar manner to step (1) above, by replacing each new property in the question with a 'known' property and then back to the new one in the generated program. This is less straight-forward than for our method, as replacements are required for various program elements used by TbD that are related to the property (*e.g*. multiple modules such as '*filter_stackability[stackable]*', '*query_stackability*'). The second step requires questions-answers training (or actually programs-answers training). As this method composes a network from modules, we added new modules for the novel 'stackability' concepts, and trained them using simple existence questions (*e.g*. 'Is there a stackable object?'). Modified weights in the model were only the ones of the new modules and of the last classification layer that are connected to the novel answers: 'stackable' and 'nonstackable'. This allows the method to learn the function required for the new modules (corresponding to training a property classifier in our method). The TbD method includes multiple modules related to the new property, and it is not possible to activate all of them during the binary training. Nevertheless, we apply this procedure (which is more demanding and less natural than the simple classifier plug-in for the UnCoRd method) as it provides a fair comparison with a corresponding method to the one used to extend UnCoRd.
The training was performed using one existence question for each CLEVR training set image. The answering accuracy for the corresponding validation set (similar type of existence questions for the new property on CLEVR validation images) is 99.95%. This shows that the model is capable of learning the new property and use it on questions of the type used in training, but not necessarily generalize to new questions.
Evaluation of the methods was performed using 10,000 new questions, generated for CLEVR validation images using CLEVR questions templates with added 'stackability' terms (for all types of questions). The results are summarized in Table 1, including results of the original TbD method without modifications (with random weights for 'stackability' modules and other related weights).
The results show that the UnCoRd model maintains its near

Method	Exist	Count	Compare Numbers	Query Attribute	Compare Attribute	Overall
TbD-st	63.7	39.2	63.8	37.9	54.7	45.6
TbD-rand_st	21.8	26.7	30.9	24.6	16.7	24.1
UnCoRd-st_tr	99.5	98.9	**99.8**	99.4	99.3	99.3
UnCoRd-st_by_cl	**99.9**	**99.4**	**99.8**	**99.8**	**99.8**	**99.7**

Table 1. Accuracy of answering questions with 'stackability' properties on CLEVR validation images.
TbD-st: TbD with binary trained 'stackability' modules
TbD-rand_st: TbD with random 'stackability' weights
UnCoRd-st_tr: UnCoRd with trained 'stackability' classifier
UnCoRd-st_by_cl: UnCoRd with 'stackability' inferred by the object class

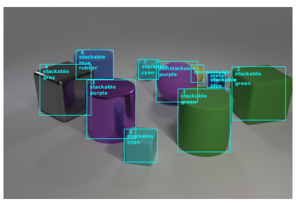
perfect results while the extensibility for the TbD fails and its performance drops considerably. As other end-to-end methods, question answering is treated as a multi-class classification problem. This means that by this approach even if the modules could have been trained properly, the weights for the last classification layers require an additional appropriate tuning. Since the used questions for TbD training were limited to binary existence questions with 'yes'/'no' answers, no data was given to properly tune the classification weights of the novel answers: 'stackable' and 'nonstackable'. This results with the TbD providing these novel answers improperly and unrelated to the actual questions, which may hide the actual impact of training the 'stackability' modules. Our method does not suffer from this issue, since the simple plugging-in of a 'stackability' classifier handles the integration of 'stackability' concepts in all types of questions, including with the novel answers. Examining and comparing performance without the effect of the novel answers was performed by removing the questions that query the 'stackability' property and excluding the novel answers from TbD optional predictions. The results for this test are given in Table 2.

Method	Exist	Count	Compare Numbers	Query Attribute	Compare Attribute	Overall
TbD-st-no_st_ans	66.6	46.4	74.0	72.0	66.2	63.3
TbD-rand_st-no_st_ans	69.2	45.2	73.2	73.0	63.9	63.5
UnCoRd-st_tr	99.5	98.9	**99.8**	99.4	99.3	99.3
UnCoRd-st_by_cl	**99.9**	**99.4**	**99.8**	**99.9**	**99.8**	**99.7**

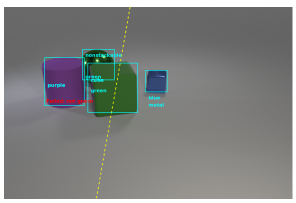
Table 2. Accuracy of answering questions with 'stackability' properties without novel answers ('stackable', 'nonstackable'). Naming corresponds to Table 1, where the additional suffix **-no_st_ans** represents cancelling the prediction of the novel answers.

For this test TbD results improve but are still much inferior to the UnCoRd method, which remained practically the same. It is interesting to see that the TbD results with trained 'stackability' modules are not better than with untrained modules. The results imply that tuning the novel modules by one specific task (one type of questions) does not guide the modules in a general direction of performing their designated tasks. Explicit training is required for other question types to increase their performance. The

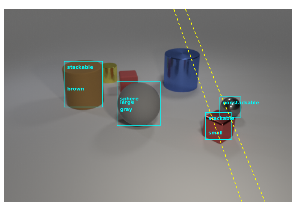
modules have no independent meaning as real 'stackability' classifiers. Our method uses real classifiers and does not share this limitation. Also, the fact that TbD with untrained 'stackability' modules does not fail completely demonstrates that each module has a limited effect in tuning the final results of the full network. As each question include at least one 'stackability' concept (average of 1.4 'stackability' concepts per question), the performance is different than expected if a real visual estimator was replaced by a random results generator. As all modules in the TbD method receive the same image features as an input (in addition to the previous module's output), this may imply that the modules have a limited (but not negligible) effect on the generated features passed to the classification layers.
Some examples of questions and answers with the 'stackability' concepts are shown in Figure 2 (UnCoRd's answers were the same for both 'stackability' classification options).



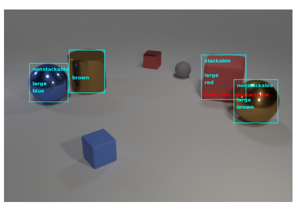
Q: What number of other objects are there of the same stackability as the blue metallic object? *GT: 7*
UnCoRd A: 7
TbD-st A: 5
TbD-st-no_st_ans A: 5



Q: There is a green thing that is on the left side of the block that is in front of the blue thing; what stackability is it? *GT: nonstackable*
UnCoRd A: nonstackable
TbD-st A: nonstackable
TbD-st-no_st_ans A: no



Q: Are there any large gray spheres left of the nonstackable object that is on the right side of the small stackable thing in front of the brown stackable object? *GT: yes*
UnCoRd A: yes
TbD-st A: no
TbD-st-no_st_ans A: no



Q: There is another nonstackable object that is the same size as the blue nonstackable thing; what is its color? *GT: brown*
UnCoRd A: brown
TbD-st A: nonstackable
TbD-st-no_st_ans A: gray

Figure 2. Examples for questions and answers on CLEVR images with the 'stackability' properties.

The results of the performed extensibility tests demonstrate the limitations in extending end-to-end VQA models, even the ones designed as a composition of modules. Compositional neural network were proposed to break the question answering task into separate tasks carried out by separate modules that are composed according to a program. However, as demonstrated above, these modules cannot be trained independently of the question answering task and cannot handle novel reasoning combinations, unseen in training. Contrary to this limitation, our method is extended naturally with novel concepts that can be used in all question types. This emphasizes the advantages of our method in extending its scope beyond the training domain.

4. Commonsense Utilization

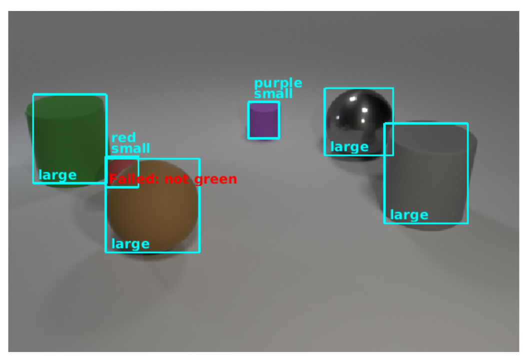
In general question answering, there is an important role to prior common knowledge, such as connections between categories of objects. To obtain general commonsense knowledge we use ConceptNet database (version 5) [15]. The source of information for this database is the internet (as well as results from additional databases). It allows querying for concepts and relations between them of the form:

concept1 - *relation* $\rightarrow$ concept2 (*e.g.* horse - *IsA* $\rightarrow$ animal)

The query is performed by providing two of the triplet [relation, concept1, concept2] and querying for the third. These common knowledge relations provide complement capabilities for answering 'real world' questions in which such common knowledge is assumed. We currently use ConceptNet mainly to extend understanding of objects' classes (*e.g.* superordinate classes and similar classes).

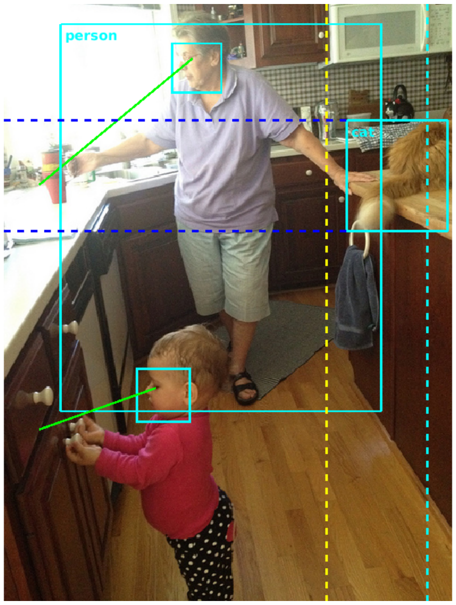
5. Examples for Answer Elaboration

Following are examples for images and corresponding questions, with answers provided by the UnCoRd system. The answers were configured to be full, and consequently provide explanations and seek alternatives, when an answer is not grounded in the image. Figure 3 demonstrates a failed grounding, and in Figure 4, alternatives are examined and reported. Note that the domains in the two examples are completely different; in Figure 3 image and question are from the CLEVR domain, while in Figure 4 the domain is real-world images and categories.



Q: Are any tiny green metal things visible?
A: There are no valid green objects (failed due to 4 large objects, a purple object and a red object)

Figure 3. An example of UnCoRd answer in the 'full' mode, explaining a failed grounding. The question-to-graph model was 'Min-Enhanced'.



Q: Is the man looking at a cat?
A: Couldn't find any object of class: man (found subclasses: a woman and a girl).
There are no people (superordinate class) looking at a cat.
Existing alternative relations (to man-looking_at-cat):
'person to the left of a cat'

Figure 4. An example for UnCoRd 'full' answer, demonstrating reporting alternatives. The question-to-graph model was 'Min-Enhanced'. The used visual estimators are mask R-CNN [3] (for the 80 classes of COCO dataset [9]), age and gender classification [8], face detection [13] and gaze estimation [14]. When a subordinate class of a person (in this case 'man') fails, the more general class 'person' is tested and proposed as an alternative. In addition, alternative relations are tested when the requested relation fails.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *International Conference on Computer Vision (ICCV)*, 2015. 1

[2] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jose M. F. Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017. 2, 3, 6

[4] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[5] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 2

[6] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, 2017. 3

[7] Satwik Kottur, José MF Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. Clevr-dialog: A diagnostic dataset for multi-round reasoning in visual dialog. *arXiv preprint arXiv:1903.03166*, 2019. 3

[8] Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) workshops*, June 2015. 6

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*. 2014. 6

[10] Runtao Liu, Chenxi Liu, Yutong Bai, and Alan L. Yuille. Clevr-ref+: Diagnosing visual reasoning with referring expressions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[11] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[12] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4942–4950, 2018. 4

[13] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *ECCV*, 2014. 6

[14] Adria Recasens*, Aditya Khosla*, Carl Vondrick, and Antonio Torralba. Where are they looking? In *Advances in Neural Information Processing Systems (NIPS)*, 2015. * indicates equal contribution. 6

[15] Robert Speer and Catherine Havasi. Conceptnet 5: A large semantic network for relational knowledge. In *The People's Web Meets NLP: Collaboratively Constructed Language Resources*, pages 161–176. Springer Berlin Heidelberg, 2013. 5