A Logic-based Approach to Contrastive Explainability for Neurosymbolic Visual Question Answering

Thomas Eiter , **Tobias Geibinger** , **Nelson Higuera** and **Johannes Oetsch**
Institute for Logic and Computation, TU Wien, Favoritenstraße 9–11, 1040 Vienna, Austria,
{thomas.eiter, tobias.geibinger, nelson.ruiz, johannes.oetsch}@tuwien.ac.at

Abstract

Visual Question Answering (VQA) is a well-known problem for which deep-learning is key. This poses a challenge for explaining answers to questions, the more if advanced notions like contrastive explanations (CEs) should be provided. The latter explain why an answer has been reached in contrast to a different one and are attractive as they focus on reasons necessary to flip a query answer. We present a CE framework for VQA that uses a neurosymbolic VQA architecture which disentangles perception from reasoning. Once the reasoning part is provided as logical theory, we use answer-set programming, in which CE generation can be framed as an abduction problem. We validate our approach on the CLEVR dataset, which we extend by more sophisticated questions to further demonstrate the robustness of the modular architecture. While we achieve top performance compared to related approaches, we can also produce CEs for explanation, model debugging, and validation tasks, showing the versatility of the declarative approach to reasoning.

1 Introduction

Visual Question Answering (VQA) [Zou and Xie, 2020] is a challenging field that combines object detection, natural language processing, and reasoning to give the answer to a question related to some visual input. VQA finds interesting applications in the real world, such as medicine and advertising [Barra *et al.*, 2021]. Although VQA has seen great advances in recent years, VQA systems typically rely on deep-learning architectures, and the important aspect of explainability [Dosilovic *et al.*, 2018]—a constant focus of the machine learning community due to the importance of building transparent and interpretable systems—is still challenging.

Contrastive explanations (CEs) [Lipton, 1990] answer why a decision has been reached in contrast to a different one. This can serve as a window into the epistemic state of the explainee, and the explanation can focus on input features that are necessary to flip an outcome instead of considering a complete causal chain. It has been argued that CEs are intuitive to understand and to produce for humans, as explanations are often (implicitly) contrastive [Miller, 2019].

Our main contribution is a *CE framework for VQA* that aims at explaining why the answer to a question is P in contrast to F. An explanation identifies a minimal set of abstract features of the scene, like types and positions of objects, that need to be changed to create the counterfactual outcome F. This approach is inspired by recent work on CE to improve the interpretability of natural-language processing models [Jacovi *et al.*, 2021]. There are however important differences to our approach for the VQA domain: Jacovi *et al.* use interventions on input factors that are mostly amnesic in nature, i.e., an alternative outcome is explained by omitting factors from the input. We are instead interested in more general changes to an input scene that involve not only removing objects but changing their attributes, moving them, or even adding new objects. Furthermore, our notion of explanations incorporates a more fine-grained view on the cost of the required transformations.

Regarding related work, visualising the contributions of individual pixels to the prediction is often used to improve the interpretability of VQA systems [Arras *et al.*, 2022], but this often gives only limited insights into the reasoning process. Networks like the MAC [Hudson and Manning, 2018] allow to follow the attention mechanism for both the reasoning steps that come from the question as well as attention on corresponding areas of the scene. The NS-VQA system presented a modular neurosymbolic architecture to disentangle perception from reasoning [Yi *et al.*, 2018]. It is interpretable as the reasoning part is implemented in Python and can be traced by a debugger. This type of interpretations resemble complete causal chains, but they are not geared to contrastiveness as explanations in our proposal. The CLEVR-X dataset [Salewski *et al.*, 2020] was designed to promote explainability for VQA. However, the task there is to select natural language expression as explanations that fit best, which may not help to produce new explanations, the less contrastive ones.

Our CE framework for VQA, which we call NSVQASP, is inspired by NS-VQA. The latter had big success in solving challenging VQA datasets like CLEVR [Johnson *et al.*, 2017]—consisting of computer generated images with geometric objects and compositional questions revolving around those—where it reaches 99% accuracy. While the reasoning part in NS-VQA is implemented in Python, Eiter *et al.* (2022) recently introduced an implementation that uses a logical theory and answer-set programming (ASP) [Brewka *et al.*, 2011] instead. Once the reasoning part is provided as logical theory,

CE generation can be framed as a task of logical abduction. ASP proves to be a promising framework to realise this, in particular as ASP optimisation allows one to express preferences with complex cost functions, which can be easily changed if need be. We present a novel variant of CE explanations designed to aid model validation tasks, which exhibit whether an answer can be flipped by only changing object attributes with low confidence scores from the object detection module.

We validate our approach on the CLEVR dataset, which we extend by several more sophisticated questions to further demonstrate the robustness of the modular architecture of NSVQASP. In particular, we add 20 new question templates for different versions of the new spatial relation "between", equality of objects, and counting. While we achieve top performance compared to related neural and neurosymbolic approaches, we can moreover produce CEs. We show this for model explanation, debugging, and validation tasks, demonstrating the versatility of the declarative approach to reasoning within modular neurosymbolic VQA architectures.

Code and data are available from https://github.com/pudumagico/nsvqasp.

2 Contrastive Explanations for VQA

Contrastive explanations (CE) aim at answering why a certain outcome occurred *in contrast* to an alternative one. In this section, we present a framework for CE in the VQA domain. Our approach is inspired by a related one from natural language processing due to Jacovi *et al.* (2021), who used an intervention-based approach, in which causal factors that lead to a model decision are identified by omitting them and thereby creating the desired counterfactual outcome.

2.1 A CE Framework for VQA

In VQA, a problem instance consists of a visual scene S and a natural language question Q. The goal is to correctly answer Q for the given scene S. A contrastive explanation clarifies why the answer produced by a model is P and not F, by showing what need to be changed in a scene so that the answer changes from P to F.

We illustrate our CE formalisation for VQA using the CLEVR dataset [Johnson *et al.*, 2017] that tests various aspects of visual reasoning including attribute identification, counting, comparison, spatial relationships, and logical operations. An example of a CLEVR scene from the original paper [Johnson *et al.*, 2017] and several questions are given in Fig. 1. For each question, we also present an alternative answer, called *the foil*, to generate a contrastive explanation.

For our formalisation, we adjust the terminology of Jacovi *et al.* (2021) to our setting. The *candidate factor space* $\mathcal{F}$ consists of all features with potential influence on a model's decision. For a VQA task, we are only interested in features that describe the scene in terms of symbolic properties like position or shape of an object.

In CLEVR, each object has a position, a colour, a shape, a size, and a material. Each of these attributes has a finite range of values. We can use a tuple representation for objects and $\mathcal{F}$ can be defined a finite set containing all object tuples.

The *event space* $\mathcal{E}$ is the set of all answers a model can produce. In CLEVR, questions are either Boolean, ask for a number when counting is involved, or ask for an object attribute (other than position). As the maximal number of objects in any scene is bounded, $\mathcal{E}$ is a finite set.

For a model M, we write $M[S, Q]$ to denote the answer that M produces for question Q and scene S. Furthermore, $a(S) \subseteq \mathcal{F}$ is the abstract scene representation for S. Our formal CE definition for VQA systems is as follows.

Definition 1. *Let M be a model with candidate factor space $\mathcal{F}$ and event space $\mathcal{E}$. Assume that $M[Q, S] = P$ for some question Q and some visual scene S. A* contrastive explanation *(CE) for* the foil *$F \subseteq \mathcal{E}$ is a set $E \subseteq \mathcal{F}$ such that $M[Q, S'] \in F$, where S' is a scene with $a(S') = E$.*

The foil F describes the contrastive outcome as a set of answers and an explanation is an abstract representation of a scene that would result in some answer from F. Typically, F contains a single answer, but $F = \mathcal{E} \setminus \{P\}$ can also be useful for explaining how to flip the outcome in any direction.

Under Occam's razor, explanations are typically required to fulfill some notion of minimality, as they should only highlight aspects of the input that necessarily have to change. We thus further require CEs to be minimal under some preference relation $E \prec E'$ between explanations E and E'.

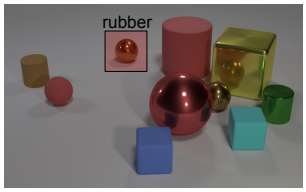
For CLEVR, a good candidate for defining preferences is to measure how many steps are required to change an abstract representation $a(S)$ of the original scene S to the CE E and how many objects are involved. For instance, it appears to be less invasive to change the position of an object than other object attributes. Also, the number of objects affected should be small so that explanations are localised.

To make this precise, we consider the following operations for transforming an abstract representation into another one: (i) adding a new object, (ii) removing an object, (iii) changing an object attribute other than the position, and (iv) changing the position of an object. The operations have associated costs that can be customised by the user. For example, they could decrease from (i) to (iv) to reflect that changing attributes is considered less expensive than deleting or adding objects. Given two abstract scene representations S and S', $cost(S, S')$ is the minimal total cost for any sequence of operations (i)–(iv) whose execution starting on S will result in S'. Explanation preference is then defined as follows.

Definition 2. *Let M be a model such that $M[Q, S] = P$ for some question Q, scene S, and answer P. Furthermore, let E and E' be two contrastive explanation for a given foil F. Explanation E is strictly preferred over E', in symbols $E \prec E'$, if $cost(a(S), E) < cost(a(S), E')$.*

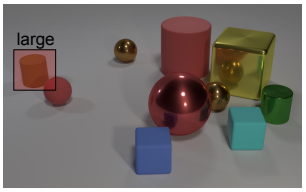
2.2 CE as Logical Abduction

Searching for input perturbations that change a decision can be challenging for deep-learning networks. However, if we use a modular neurosymbolic approach for the VQA task, separate modules can take care of object detection, language processing, and symbolic execution to compute an answer. Once the execution module is provided as a logical theory, cf. [Eiter *et al.*, 2022], the answers can be computed using *deduction*, while contrastive explanations can be obtained through *abduction*.



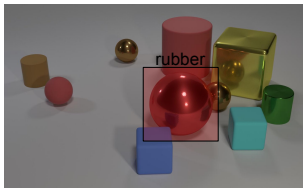
Q1: Are there an equal number of large things and metal spheres?

A1: yes **F1:** Why not "no"?



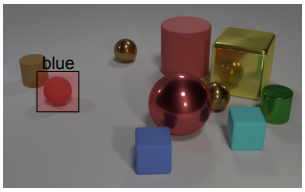
Q2: What size is the cylinder that is left of the brown metal thing that is left of the big sphere?

A2: small **F2:** Why not "large"?



Q3: There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?

A3: yes **F3:** Why not "no"?



Q4: How many objects are either small cylinders or red things?

A4: 5 **F4:** Why not "4"?

Figure 1: A CLEVR scene with CEs for several questions. For each question **Q** with answer **A**, we present a foil **F** for generating a CE. The explanation is then illustrated in the scene by highlighting the objects that need to be changed such that the answer changes from **A** to **F**.

Assume $T(M)$ is a logic theory that encodes the reasoning module of a VQA system M such that $T(M) \cup a(S) \cup a(Q) \models P$, whenever P is the answer to question Q for scene S, and $a(Q)$ is a symbolic representation of question Q.

Contrastive explanations correspond to minimal abductive explanations so that the foil is entailed by the theory, as formally stated by the following proposition.

Proposition 1. *Let M be a model such that $M[Q, S] = P$ for some question Q, scene S, and answer P. Then, E is a minimal CE for F iff*

(i) $E \cup T(M) \cup a(Q)$ is consistent,
(ii) $E \cup T(M) \cup a(Q) \models F$, and
(iii) no E' with $E' \prec E$ satisfies (i) and (ii).

2.3 Applications of CE for VQA

CEs not only help to make VQA systems more trustworthy by providing insights why an answer was produced, they also can aid model debugging and model validation. The difference between these tasks mainly lies in how the foil is defined.

Model Explanation. Explanations for correct answers are often required if a user had a different outcome in mind. Contrastive explanations make this explicit by specifying the expected outcome as the foil. This focuses the explanation on reasons relevant to obtain the alternative outcome and makes them more succinct than presenting a complete causal chain. For instance, consider question Q2: What size is the cylinder that is left of the brown metal thing that is left of the big sphere? from Fig. 1. The user expects the answer to be "large" while it actually is "small". A full causal explanation would first identify the big red metal sphere in the scene. Then, it would point to the small brown metal sphere left to it, and then finally to the small brown rubber cylinder left of that one. A contrastive explanation would immediately point to the same brown rubber cylinder and suggest to change its size to "large" to flip the answer and thus focuses attention to the part of the scene that is most relevant.

Model Debugging. Debugging is necessary if a VQA system produces a wrong answer P'. Here, CEs are useful as the correct answer P can serve as the foil $\{P\}$ and we get a compact explanation what changes in a scene would result in the correct outcome. Having a modular framework helps in general for debugging, as we have already explicit representations of the scene and the question in symbolic form which are essentially self explanatory. The CE further helps to know where to look first. For example, assume the material of an object gets misclassified and the answer of a question is wrong as a consequence. If the CE reveals that changing the wrong label to the correct one also leads to the correct answer, the underlying problem in the object detection module is revealed.

Model Validation. How do we know if we should not trust a particular answer given by a VQA system? We define *model validation* as the task to automatically detect whether a given answer is not trustworthy, and present model validation as a novel application of CEs. A common source of wrong answers are mistakes by the object detection module. If the latter is implemented with neural networks, low confidence scores for the labels of object attributes hint at possible misclassification. This does not necessarily lead to a wrong answer, as it might be irrelevant for the question considered. However, if the answer can flip in any direction by changing only attribute values of objects that have a low score, manual inspection is advised. This can be formulated as a CE generation problem: We take $\mathcal{E} \setminus \{P\}$ as the foil, where $\mathcal{E}$ is the event space and P is the current answer. For the preference relation, we only allow changing attributes with scores below a fixed threshold by assigning infinite (i.e., huge) cost to all other operations in the cost function of Defn. 2. If we obtain a CE, low scores for the objects highlighted by the explanation might be critical for the outcome and labels should be inspected.

3 The Neurosymbolic VQA Framework

In this section, we introduce a modular neurosymbolic VQA framework, which we call NSVQASP, for solving VQA tasks for a new extension of the CLEVR dataset that is also capable of producing CEs. It is based on the modular approach of NS-VQA [Yi *et al.*, 2018], but it features a logic module for symbolic execution where answer-set programming (ASP) is used for reasoning and explanation finding. To this end, we exploit recent work [Eiter *et al.*, 2022] and extend it by an abductive module for obtaining minimal CEs.

3.1 Architecture of NSVQASP

We adopt the modular neurosymbolic architecture of NS-VQA, which maintains a separation between the tasks needed for

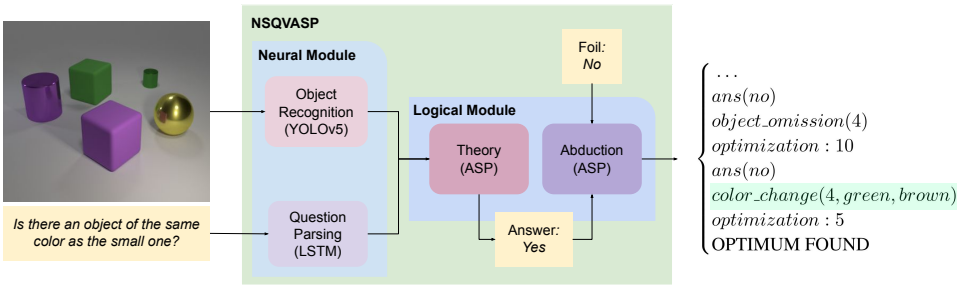


Figure 2: Overview of the inference and abduction workflow in NSVQASP. The image and natural language question are processed by YOLO and the LSTM, respectively, and output is transformed into ASP facts. The answer for a question and CEs for a given foil are computed using an ASP solver. An example output is shown on the right, meaning that changing the indicated colour for the green cube is a minimal CE.

language processing, object detection, and symbol processing to answer a question. There, neural networks are used for language processing and object detection, while the symbolic execution is realised with a Python inference module. NS-VQA uses the Mask RCNN [He *et al.*, 2020] for object detection in a way such that a visual scene can be translated into an abstract structural representation, and a long short-term memory network (LSTM) [Luong *et al.*, 2015] to translate a question into a functional program that can be executed on the abstract scene using the inference module.

The architecture of NSVQASP and the workflow to answer and explain a question is depicted in Fig. 2. We use YOLOv5[1], a popular and up-to-date choice for visual object recognition, instead of the Mask RCNN. However, we maintain the original LSTM implementation, as it works almost perfectly for CLEVR. The more significant change is however to replace the Python module for inference with a logical theory that can be used for both "foward" deduction to compute an answer, and for "backward" abduction to find a CE for a given foil in the sense of Prop. 1. Both the theory and the novel abduction module are realised via ASP [Brewka *et al.*, 2011] as it provides the expressiveness required to formalise the CLEVR questions that involve counting and arithmetic as well as the flexibility to encode the abduction task as an optimisation problem that can be easily customised.

3.2 The ASP Module for Inference and CEs

Answer-set programming (ASP) [Brewka *et al.*, 2011] is a declarative problem solving paradigm, where problems are encoded as a logic program such that their models (called answer sets) represent the solutions. It allows for a concise representation of search and optimisation problems for which solutions can be computed with some dedicated ASP solver.[2] A logic program is a finite set of rules of the form $Head :- Body$, where $Head$ is a first-order atom and $Body$ is a set of first-order literals (intuitively, $Head$ is true if $Body$ is true). Facts are rules with empty body that are use to represent problem instances. Constraints are rules without heads that eliminate unwanted answer set candidates. Further language constructs readily available include choice rules for expressing nondeterminism, aggregates, and optimisation statements with priorities. A thorough introduction to the modelling language is given by Calimeri *et al.* (2020).

Forward Inference for Question Answering

We use an ASP encoding for answering CLEVR questions due to Eiter *et al.* (2022). They considered a non-deterministic setting, where multiple answers can be derived that have different probabilities depending on the confidence scores from the object detection network, and a deterministic setting, where only a unique answer is derived from the object classifications that obtained maximal scores. Here, we use the deterministic setting and denote the corresponding encoding by T_{asp}.

The encoding T_{asp} assumes an ASP fact representation $a_{asp}(S)$ of the visual scene S and $a_{asp}(Q)$ for the question Q. The answer can then be derived with an ASP solver:

Proposition 2. *Let P be the unique answer for* CLEVR *question Q on scene S. Then, $T_{asp} \cup a_{asp}(S) \cup a_{asp}(Q)$ yields a single answer set that contains* `ans(`P`)`.

We illustrate this by example, details are given in related work [Eiter *et al.*, 2022].

Example 1. *Let S be a* CLEVR *scene with the following fact representation $a_{asp}(S)$:*

```
obj(0,large,blue,metal,cylinder,417,137).
obj(1,large,purple,metal,cube,150,195).
obj(2,large,gray,metal,sphere,265,234).
obj(3,large,brown,rubber,cylinder,145,117).
obj(4,small,green,metal,sphere,339,116).
obj(5,small,purple,rubber,sphere,226,165).
```

Each object in S is encoded as an ASP fact that defines its ID, size, colour, shape, and position (as center point). Furthermore, let Q be the question "Is the number of rubber objects left to the cube equal to the number of metallic cylinders?". Its symbolic representation $a_{asp}(Q)$ by ASP facts is:

```
                scene(0).          unique(5,4).
   filter_metal(1,0).       relate_left(6,5).
filter_cylinder(2,1).    filter_rubber(7,6).
          count(3,2).             count(8,7).
    filter_cube(4,0).  equal_integer(9,3,8).
                                    end(9).
```

Each fact $p(n_1, \ldots, n_k)$ denotes a processing step, where p is the operator, n_1 is the step number, and $n_2, \ldots, n_k$ are the preceding steps that feed into p; $end(9)$ marks the final step.

The unique answer set of $T_{asp} \cup a_{asp}(S) \cup a_{asp}(Q)$ contains `ans(true)`*, which encodes the correct answer "yes".*

[1] https://ultralytics.com/yolov5.

[2] E.g., potassco.org, www.dlvsystem.com.

Abduction for Computing CEs

We present a novel ASP program A_{asp} to compute CEs with abductive reasoning as described in Prop. 1. We illustrate some details; the full program is in the project's online repository.

To compute a CE, A_{asp} uses (1) choice rules to non-deterministically span the search space of possible operations on a scene, (2) a constraint to enforce that applying these operations changes the answer to one that is specified as foil, and (3) weak (i.e., soft) constraints to encode the minimality condition from Defn. 2 as an optimisation objective.

As for (1), we illustrate the choice rules for changing a scene with an example that changes some colour attribute:

```
{ has_col(ID,C): col(C)} = 1 :- object(ID).
 col_change(ID,C,C') :- has_col(ID,C'),
            obj(0,ID,_,C,_,_,_,_), C != C'.
```

The first rule non-deterministically selects a colour for an object, while the second derives `col_change` if that colour represents a change compared to the original scene.

The constraint of (2) ensuring an answer change is

```
:- ans(A), #count{ 1 : foil(A) } != 1.
```

Here, `foil` refers to the foil F, which is represented by facts $a_{asp}(F) = \{\texttt{foil}(p) \mid p \in F\}$.

In (3), the following weak constraint is added:

```
:~ col_change(ID,C,C'). [5,ID,col_change]
```

It says that any colour change of object *ID* (from C to C') is penalised by a cost of 5, which is recorded in the cost tuple.

Rules for the other change operations are defined analogously. The maximal number of objects that can be added is currently limited by a constant that can be adjusted by the user. Likewise, movement of objects proceeds in increments of a fixed number of pixels.

Proposition 3. *Given a* CLEVR *question Q and scene S, the optimal answer sets of $A_{asp} \cup T_{asp} \cup a_{asp}(S) \cup a_{asp}(Q) \cup a_{asp}(F)$ are in one-to-one correspondence to the minimal CEs for foil F.*

Example 2. *Let us revisit question Q on scene S with answer "yes" from Example 1. Assume the user expects this answer to be "no" instead. The CEs can be computed as answer sets following Prop. 3. For illustration, one of them contains*

```
moved(1,150,195,100,195)
```

which represents the minimal change to flip the answer by moving the cube 50 pixels to the left.

What changes to a scene are permitted and/or the cost model can be changed easily if desired.

4 Evaluation

Prior to an evaluation of our CE approach, we test the accuracy of NSVQASP on CLEVR and compare it against NS-VQA and other baseline approaches. We extend CLEVR with more sophisticated questions to further demonstrate the robustness of the modular neurosymbolic architecture. Also, it is useful for the explanation task to include questions that require a little more advanced reasoning and are thus harder to explain.

4.1 Extending CLEVR

The CLEVR dataset consists of 70k images plus 700k questions for training and 15k images plus 150k questions for validation. Questions are generated from templates which define the structure of a question. We extend the CLEVR dataset by introducing 20 new templates that include a new spatial relation "between", questions regarding equality of objects, and new counting questions, respectively; consequently, they can be divided into three groups. We generated 200k new questions from the templates for training for each group and 150k questions for validation.

The "Between" group. This group contains templates for questions that ask whether an object is between two other ones. We consider three semantics for "between" illustrated in Fig. 3: Object b is between object a_1 and a_2 if (a) the projection on the horizontal axis of b falls between the one of a_1 and a_2; (b) b is within the bounding box created by the centers of a_1 and a_2; and (c) the distance of b to the segment connecting the center points from a_1 to a_2 is below a fixed threshold.

The "Equal" group. This group contains templates for questions that revolve around tests whether objects are equal (agree on all their attributes aside from position) or different. More specifically, we ask: "Are all object in the scene different?" and "Are there (at least or exactly) n equal objects in the scene?", where $i \in \{2, 3, 4\}$ (scenes with more than 4 equal objects are rare in CLEVR).

The "Count-Between" group. This group contains templates for variants of the "between" questions that involve counting. Specifically, we ask for the number of objects that are between two other ones, where we also consider the three semantic versions of "between" from above.

4.2 Experimental Comparison

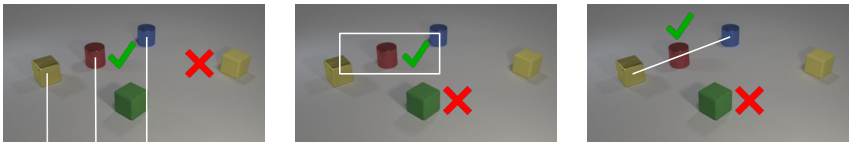
We use the original CLEVR dataset, as well as the new datasets NEW_{btwn}, NEW_{eql}, and NEW_{cnt} as described above. Dataset NEW was generated from the union of the question templates for NEW_{btwn}, NEW_{eql}, and NEW_{cnt}.

As baseline approaches, we consider (1) the neurosymbolic system NS-VQA [Yi *et al.*, 2018], the relational network RN [Santoro *et al.*, 2017] that uses an CNN and LSTM combination and introduces a relational cell that combines the extracted features from the image with the processed question before using a classifier, and (2) the MAC [Hudson and Manning, 2018], which is an end-to-end trainable network that separates control from memory by decomposing a problem into attention-based reasoning steps.

The results of our comparisons are given in Table 1.[3] We could reproduce the results for the related approaches on CLEVR from the literature. For CLEVR ∪ NEW, the neurosymbolic approaches work better, even more so when only NEW is used for training.[4] A possible reason is that the neu-

[3]We use an Intel® Core™ i7-12700K, 32GB RAM, and an NVIDIA GeForce RTX 3080 Ti for training. YOLOv5 was trained with the CLEVR mini dataset (https://github.com/kexinyi/ns-vqa).

[4]NS-VQA answers some questions from NEW incorrectly due to issues that arise when converting coordinates from their object detection module for the more advanced questions. NSVQASP performs sometimes worse than NS-VQA due to the different visual modules.



(a) (b) (c)

Figure 3: Different semantics for "between" that use (a) projection, (b) bounding boxes, and (c) the segment between objects.

Model	CLEVR	CLEVR ∪ NEW	Only NEW
RN	95.21%	92.32%	80.76%
MAC	98.73%	92.94%	81.43%
NS-VQA	99.89%	98.79%	94.66%
NSVQASP	99.82%	97.39%	98.22%

Table 1: Accuracies for the original CLEVR dataset, CLEVR extended by new questions, and only new questions.

Model	projection	bounding-box	segment
RN	94.29%	91.88%	92.65%
MAC	92.75%	90.34%	90.42%
NS-VQA	97.95%	93.56%	87.16%
NSVQASP	99.44%	98.88%	98.22%

Table 2: Accuracies for dataset CLEVR ∪ NEW_{btwn} for different semantics of "between".

ral approaches may need more visual scenes to learn the new questions, while we merely need to adjust the symbolic execution module (or the ASP encoding for NSVQASP) and retrain the LSTM; convenient end-to-end learning is currently not featured by NS-VQA and NSVQASP, though.

We give a more in-depth analysis of the performance of the different VQA approaches relative to the datasets NEW_{btwn}, NEW_{eql}, and NEW_{cnt} in Tables 2–4. Regarding the different semantics for between, the neural approaches have more difficulties learning the version with bounding-boxes than the one with projection. Also, it appears that the one that uses the distance to a segment is most difficult. For the MAC and the RN, the questions in NEW_{btwn} seem to be easier than those in NEW_{cnt}, and the ones in NEW_{eql} seem to be hardest. While more scenes and questions would plausibly improve the performance of the MAC and the RN, the less data hungry neurosymbolic approaches perform already very well.

Model	all-different	$n = 2$	$n = 3$	$n = 4$
RN	72.65%	69.40%	83.49%	99.19%
MAC	76.93%	77.77%	90.09%	98.38%
NS-VQA	100.00%	99.84%	100.00%	100.00%
NSVQASP	98.85%	99.02%	100.00%	100.00%

Table 3: Accuracies for dataset CLEVR ∪ NEW_{eql}, with questions concerning the existence of n equal objects.

Model	projection	bounding-box	segment
RN	43.29%	63.55%	75.30%
MAC	65.28%	61.08%	66.18%
NS-VQA	95.82%	88.07%	76.45%
NSVQASP	98.40%	98.68%	97.13%

Table 4: Accuracies for dataset CLEVR ∪ NEW_{cnt}. with counting questions in combination with "between".

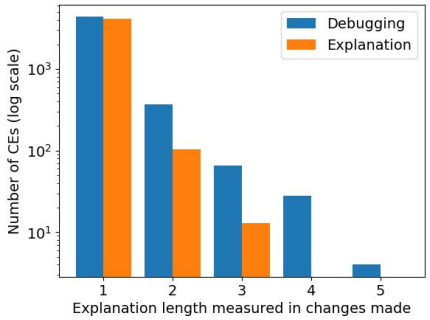


Figure 4: Number of CEs ordered by their size.

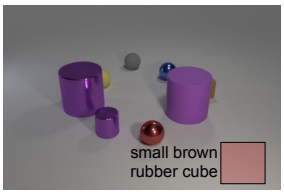
4.3 Contrastive Explanations

We conducted a quantitative analysis regarding the average length of CEs and times needed to produce them, and we provide some further illustrations for explanation, debugging, and validation tasks as described in Sec. 2. We qualitatively evaluated our approach by checking whether explanations make intuitively sense on samples.

Explanation. Fig. 4 summarises the outcome of our first experiment on contrastive explainability. We let NSVQASP determine the outcome for a sample of 5k questions from CLEVR ∪ NEW. For each answer P, we computed the minimal CEs to change the outcome in any direction, hence the foil $F = \mathcal{E} \setminus \{P\}$. The orange bars in Fig. 4 depict the number of CEs as a function of the length of the CE (measured as the number of required changes, or, equivalently, the size of the optimal answer set that encodes the CE). For most questions, only one or two changes to a scene are needed but never more than five. The time for finding a minimal CE ranges from $1.1s$ to $323.2s$ with an average of $6.4s \pm 7.3s$.[5]

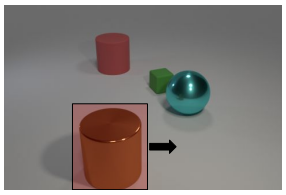
Debugging. As NSVQASP does not predict the correct answer for all questions from CLEVR ∪ NEW, we can use the

[5]We use clingo (v. 5.6.2) [Gebser *et al.*, 2019] with unsatisfiable core-guided optimisation [Andres *et al.*, 2012].



Q1: How many objects are brown cubes?

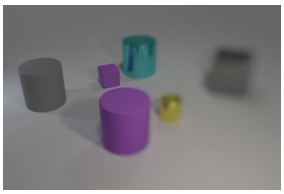
A1: 0 (correct is 1)



Q2: Has the cylinder right to the red cylinder the same size as the large red rubber cylinder?

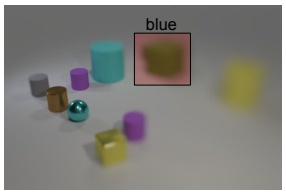
A2: no (correct is "yes")

Figure 5: Two scenes where the answer is not correct. The CEs point to errors from YOLO when using the correct answer as foil.



Q1: What is the material of the large object left to the small rubber cube?

A1: rubber



Q2: How many green cubes are in the scene?

A2: 1

Figure 6: Two scenes with blurred objects to challenge YOLO. A CE exists if their correct classification is relevant for the answer.

other ones (about 4k out of 150k questions) to illustrate CEs for debugging. For this task, we use the correct ground-truth answer as a singleton foil to find an explanation that changes the wrong prediction to the correct one. Figure 4 shows the number of CEs having a given length as blue bars for the debugging setting. Similar to the previous experiment, CEs tend to be short. The time to generate CEs in this setting ranges from $0.5s$ to $18.6s$ with an average of $7.3s \pm 4.2s$. Fig. 5 shows examples of two scenes with CEs for debugging, where NSVQASP predicated the wrong answer. Inspecting the CEs points to the relevant mistakes made by the object detector; they could be fixed by further fine-tuning YOLO.

Validation. Last, we illustrate how CEs can help to spot object classifications in a scene that should be manually inspected as (i) the answer depends on that classification, and (ii) the classification scores from the neural module are low, which indicates that a misclassification is likely. For the according model validation task in Sec. 2, the foil are all other outcomes and the cost model incorporates the confidence scores from YOLO. Fig. 6 shows example scenes that contain blurred objects to challenge object detection. For the left one, the blurred object is irrelevant and hence no CE for validation exists. For the scene on the right, the answer would flip if the blurred object changes, which is highlighted by a respective CE.

5 Further Related Work

Stepin *et al.* (2021) provide an excellent survey on contrastive explainability in machine learning. The role of logic in this context was further discussed by Marques-Silva (2023). Ignatiev *et al.* (2020) expanded on the formal relationship between abduction and CEs. Notably, CEs have been proposed in various machine-learning contexts, like for decision lists [Ignatiev and Silva, 2021], or natural language processing [Jacovi *et al.*, 2021]. In the computer-vision domain, there have been efforts to unify adversarial and counterfactual explanations [Freiesleben, 2022]. To the best of our knowledge, CEs have not been considered for VQA.

As already discussed in the introduction, techniques to make VQA systems more interpretable include tracing of the symbolic execution (NS-VQA) and tracing the attention steps (MAC). Although they do not involve contrastiveness, tracing techniques can still be very useful. For our ASP approach, we can take advantage of off-the-shelf tools such as xclingo [Cabalar *et al.*, 2020] to compute justifications for answer sets that are similar to such traces. Inspiration for building an ASP neurosymbolic system comes from NeurASP [Yang *et al.*, 2020] that uses neural atoms for the interface between the logical and the neural module. Similar approaches exist, such as DeepStochLog [Winters *et al.*, 2022], Slash [Skryagin *et al.*, 2022], and DeepProbLog [Manhaeve *et al.*, 2018].

Many extensions of CLEVR were conceived to study aspects like hypothetical consequences of performing specific actions [Sampat *et al.*, 2021], mathematical questions [Lindstrom and Abraham, 2022], or reasoning about object rotations [Beckham *et al.*, 2023]. While they are not relevant for us immediately as CLEVR suffices to demonstrate our approach, they could be considered for future work.

6 Discussion and Conclusion

Our approach to CE generation, which we frame as a problem of logical abduction, is a further step towards building robust and explainable VQA systems. We complement the landscape of existing approaches to make VQA systems more interpretable by providing explanations that are directly linked to the answer expected by a user. This is not only useful for gaining a better understanding for the actual answer, but also for debugging (if the model is wrong) and for validation tasks.

Our proposed CE framework relies on a modular neurosymbolic architecture with a clear separation between object detection, language processing, and reasoning that is inspired by NS-VQA [Yi *et al.*, 2018]. We use ASP to realise the reasoning module for computing answers, and we provide an extension of the ASP module so that CEs can be computed via abductive reasoning. Besides being able to smoothly switch between deduction and abduction, ASP makes it also easy to customise the system by changing details of the declaratively specified reasoning tasks. Having a modular VQA architecture helps in general when requirements change as only the affected modules need to be updated. We demonstrated this for an extension of CLEVR with new questions, where the modular neurosymbolic frameworks achieve top performance.

For future work, we plan to increase convenience of preference specification (e.g., support of priority levels) and to further evaluate the utility of CEs by user studies involving also real-world oriented datasets.

Acknowledgments

This work was supported by funding from the Bosch Center for Artificial Intelligence. Furthermore, Tobias Geibinger is a recipient of a DOC Fellowship of the Austrian Academy of Sciences at the Institute of Logic and Computation at the TU Wien.

References

[Andres *et al.*, 2012] Benjamin Andres, Benjamin Kaufmann, Oliver Matheis, and Torsten Schaub. Unsatisfiability-based optimization in clasp. In *Technical Communications of the 28th International Conference on Logic Programming (ICLP 2012)*, volume 17 of *LIPIcs*, pages 211–221. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012.

[Arras *et al.*, 2022] Leila Arras, Ahmed Osman, and Wojciech Samek. CLEVR-XAI: A benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40, 2022.

[Barra *et al.*, 2021] Silvio Barra, Carmen Bisogni, Maria De Marsico, and Stefano Ricciardi. Visual question answering: Which investigated applications? *Pattern Recognit. Lett.*, 151:325–331, 2021.

[Beckham *et al.*, 2023] Christopher Beckham, Martin Weiss, Florian Golemo, Sina Honari, Derek Nowrouzezahrai, and Christopher Pal. Visual question answering from another perspective: CLEVR mental rotation tests. *Pattern Recognition*, 136:109209, 2023.

[Brewka *et al.*, 2011] Gerhard Brewka, Thomas Eiter, and Miroslaw Truszczynski. Answer set programming at a glance. *Commun. ACM*, 54(12):92–103, 2011.

[Cabalar *et al.*, 2020] Pedro Cabalar, Jorge Fandinno, and Brais Muñiz. A system for explainable answer set programming. In *Technical Communications of the 36th International Conference on Logic Programming (ICLP 2020)*, volume 325 of *EPTCS*, pages 124–136, 2020.

[Calimeri *et al.*, 2020] Francesco Calimeri, Wolfgang Faber, Martin Gebser, Giovambattista Ianni, Roland Kaminski, Thomas Krennwallner, Nicola Leone, Marco Maratea, Francesco Ricca, and Torsten Schaub. ASP-core-2 input language format. *Theory and Practice of Logic Programming*, 20(2):294–309, 2020.

[Dosilovic *et al.*, 2018] Filip Karlo Dosilovic, Mario Brcic, and Nikica Hlupic. Explainable artificial intelligence: A survey. In *Proc. of the 41st International Convention on Information and Communication Technology*, pages 210–215. IEEE, 2018.

[Eiter *et al.*, 2022] Thomas Eiter, Nelson Higuera, Johannes Oetsch, and Michael Pritz. A neuro-symbolic ASP pipeline for visual question answering. *Theory and Practice of Logic Programming*, 22(5):739–754, 2022.

[Freiesleben, 2022] Timo Freiesleben. The intriguing relation between counterfactual explanations and adversarial examples. *Minds Mach.*, 32(1):77–109, 2022.

[Gebser *et al.*, 2019] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Multi-shot asp solving with clingo. *Theory and Practice of Logic Programming*, 19(1):27–82, 2019.

[He *et al.*, 2020] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2):386–397, 2020.

[Hudson and Manning, 2018] Drew A. Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. In *Proc. of the 6th International Conference on Learning Representations (ICLR 2018)*, 2018.

[Ignatiev and Silva, 2021] Alexey Ignatiev and João P. Marques Silva. Sat-based rigorous explanations for decision lists. In *Proc. of the 24th International Conference Theory and Applications of Satisfiability Testing (SAT 2021)*, volume 12831 of *Lecture Notes in Computer Science*, pages 251–269. Springer, 2021.

[Ignatiev *et al.*, 2020] Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and João Marques-Silva. From contrastive to abductive explanations and back again. In *In Proc. of the 19th International Conference of the Italian Association for Artificial Intelligence (AIxIA 2020)*, volume 12414 of *Lecture Notes in Computer Science*, pages 335–355. Springer, 2020.

[Jacovi *et al.*, 2021] Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. Contrastive explanations for model interpretability. In *Proc. of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, pages 1597–1611. Association for Computational Linguistics, 2021.

[Johnson *et al.*, 2017] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pages 1988–1997. IEEE Computer Society, 2017.

[Lindstrom and Abraham, 2022] Adam Dahlgren Lindstrom and Savitha Sam Abraham. Clevr-math: A dataset for compositional language, visual and mathematical reasoning. In *Proc. of the 16th International Workshop on Neural-Symbolic Learning and Reasoning as part of the 2nd International Joint Conference on Learning & Reasoning (IJCLR 2022)*, volume 3212 of *CEUR Workshop Proceedings*, pages 155–170, 2022.

[Lipton, 1990] Peter Lipton. Contrastive explanation. *Royal Institute of Philosophy Supplement*, 27:247–266, 1990.

[Luong *et al.*, 2015] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics, 2015.

[Manhaeve *et al.*, 2018] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De

Raedt. Deepproblog: Neural probabilistic logic programming. In *Proc. of Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems (NeurIPS 2018)*, pages 3753–3763, 2018.

[Marques-Silva, 2023] Joao Marques-Silva. *Logic-Based Explainability in Machine Learning*, pages 24–104. Springer Nature Switzerland, Cham, 2023.

[Miller, 2019] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019.

[Salewski *et al.*, 2020] Leonard Salewski, A. Sophia Koepke, Hendrik P. A. Lensch, and Zeynep Akata. CLEVR-X: A visual reasoning dataset for natural language explanations. In *xxAI - Beyond Explainable AI - International Workshop, Held in Conjunction with ICML 2020, Revised and Extended Papers*, volume 13200 of *Lecture Notes in Computer Science*, pages 69–88. Springer, 2020.

[Sampat *et al.*, 2021] Shailaja Keyur Sampat, Akshay Kumar, Yezhou Yang, and Chitta Baral. Clevr_hyp: A challenge dataset and baselines for visual question answering with hypothetical actions over images. In *Proc. of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021)*, pages 3692–3709. Association for Computational Linguistics, 2021.

[Santoro *et al.*, 2017] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Tim Lillicrap. A simple neural network module for relational reasoning. In *Proc. of Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NeurIPS 2017)*, pages 4967–4976, 2017.

[Skryagin *et al.*, 2022] Arseny Skryagin, Wolfgang Stammer, Daniel Ochs, Devendra Singh Dhami, and Kristian Kersting. Neural-Probabilistic Answer Set Programming. In *Proc. of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR 2022)*, pages 463–473, 2022.

[Stepin *et al.*, 2021] Ilia Stepin, José Maria Alonso, Alejandro Catalá, and Martin Pereira-Fariña. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, 9:11974–12001, 2021.

[Winters *et al.*, 2022] Thomas Winters, Giuseppe Marra, Robin Manhaeve, and Luc De Raedt. Deepstochlog: Neural stochastic logic programming. In *Proc. of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2022)*, pages 10090–10100. AAAI Press, 2022.

[Yang *et al.*, 2020] Zhun Yang, Adam Ishay, and Joohyung Lee. Neurasp: Embracing neural networks into answer set programming. In Christian Bessiere, editor, *Proc. of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020)*, pages 1755–1762, 2020.

[Yi *et al.*, 2018] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic VQA: disentangling reasoning from vision and language understanding. In *Proc. of Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018 (NeurIPS 2018)*, pages 1039–1050, 2018.

[Zou and Xie, 2020] Yeyun Zou and Qiyu Xie. A survey on VQA: Datasets and approaches. In *Proc. of the 2nd International Conference on Information Technology and Computer Application (ITCA 2020)*, pages 289–297, 2020.