Bridging Language and Scenes through Explicit 3-D Model Construction

Tiansi Dong[1,2] **and Writwick Das**[1] **and Rafet Sifa**[1]
[1]Hybrid Intelligence, Fraunhofer IAIS / Schloss Birlinghoven, 1, 53757 Sankt Augustin, Germany
[2]Computer Laboratory, University of Cambridge / 15 JJ Thomson Ave, Cambridge, UK
{tiansi.dong, writwick.das, rafet.sifa}@iais.fraunhofer.de
td540@cam.ac.uk

Abstract

We introduce the methodology of explicit model construction to bridge linguistic descriptions and scene perception and demonstrate that in Visual Question-Answering (VQA) using MC4VQA (Model Construction for Visual Question-Answering), a method developed by us. Given a question about a scene, our MC4VQA first recognizes objects utilizing pretrained deep learning systems. Then, it constructs an explicit 3-D layout by repeatedly reducing the difference between the input scene image and the image rendered from the current 3-D spatial environment. This novel "iterative rendering" process endows MC4VQA the capability of acquiring spatial attributes without training data. MC4VQA outperforms NS-VQA (the SOTA system) by reaching 99.94% accuracy on the benchmark CLEVR datasets, and is more robust than NS-VQA on new testing datasets. With newly created testing data, NS-VQA's performance dropped to 97.60%, while MC4VQA still kept the 99.0% accuracy. This work sets a new SOTA performance of VQA on the benchmark CLEVR datasets, and shapes a new method that may solve the out-of-distribution problem. The source code and data sets are available for public access https://github.com/writzx/mc4vqa/.

1 Introduction

The success of LLMs is witnessed by its capability of human-like question-answering (Biever, 2023), but, they remain as black-box systems, data hungry, and do not work well for out-of-distribution data in real application (Goyal and Bengio, 2022). Spatial semantics bridges spatial descriptions and visual perception and is the first semantics that human babies acquire. It is used as a reference for the understanding of other semantics (Regier, 1997; Bellmund et al., 2018). It plays a fundamental role in computational linguistics and cognitive modelling (Tversky, 2019). Visual question answering (VQA) is a challenging task that involves answering questions about an image in natural language (Agrawal et al., 2016; Wu et al., 2016). For example, given an image of a dice and the question "What is the shape of the object?", a VQA system should be able to generate the answer "cube". VQA is a challenging task because it requires the model to understand both the visual and spatial content of the image and the meaning of the question (Agrawal et al., 2016; Zou and Xie, 2020). A VQA system must be able to reason about spatial relations, such as the distance between objects, the relative positions of objects, and the orientation of objects. The state-of-the-art (SOTA) VQA system is Neural-Symbolic VQA (NS-VQA) (Yi et al., 2019). NS-VQA achieves a near-perfect accuracy of 99.8% on the CLEVR dataset (Johnson et al., 2016), which is a challenging dataset of images and questions that test a VQA system's ability to reason about spatial relations.

NS-VQA combines deep representation learning for visual recognition and language understanding with symbolic program execution for reasoning. NS-VQA generates executable programs as the meaning of the question, and apply for the learned visual and spatial attributes to produce the answer. NS-VQA learns spatial attributes about an input image by supervised deep learning. Therefore, it does not have an explicit 3-D spatial layout of the input image. This weakens the explainability and reliability, makes the system data-hungry and performs well only when training and testing data share the same or very similar distribution (Goyal and Bengio, 2022; Gigerenzer, 2022).

On the other hand, sufficient empirical experiments in psychological research advocates the model theory for spatial reasoning (Johnson-Laird and Byrne, 1991; Knauff et al., 2003; Goodwin and Johnson-Laird, 2005; Knauff, 2009, 2013), whose standard process is a sequence of *model construction, model inspec-*

Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning @ COLING 2025, pages 51–60
January 20th, 2025.

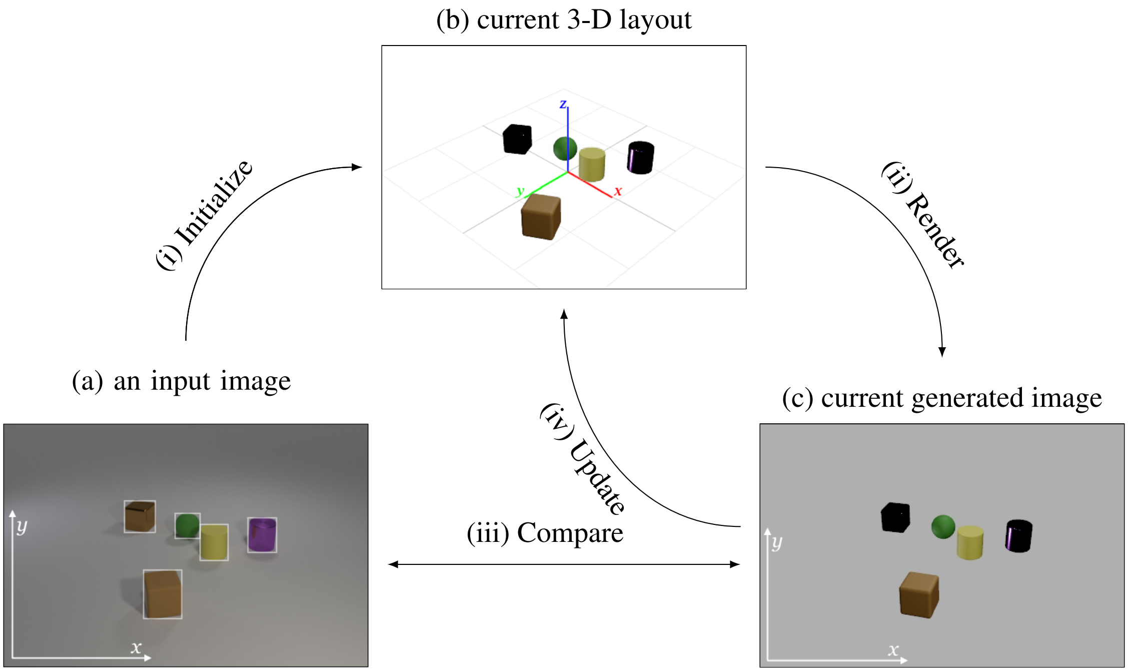


Figure 1: Overview of the MCIR process: (a) An input 2-D image; (i) Initializing a 3-D model of a scene with the colors, shapes and materials of the objects detected in the 2-D input image; (b) Reconstruction of a 3-D spatial layout of the input image; (ii) Perform perspective projection on the 3-D model to generate a 2-D image and realistic 2-D coordinates of the objects; (c) A projected 2-D image generated using the current 3-D spatial layout; (iii) Compare the projected coordinates of the objects with the bounding boxes to calculate their distances from their original 2D locations; (iv) Update the positions of the objects in 3-D layout to reduce the difference calculated in (iii).

tion, and model variation (Johnson-Laird and Byrne, 1991). The preferred mental model theory argues that people construct a preferred and simplified model in mind, in a deterministic manner, while ignoring other possible models (Ragni and Knauff, 2013; Knauff, 2013) – The construction of the first model shall not be a stochastic process that *produces one model this time and another the next time* (Ragni and Knauff, 2013, p.563-564), the next model will be revised following the principle of minimal changes from the current one (Harman, 1986; Gärdenfors, 1988; Gädenfors, 1990; Knauff et al., 2013), and generated by a local transformation of the current model.

Inspired from the model theory, here, we move one step ahead of NS-VQA, by replacing its supervised learning component of spatial attribute with a 3-D spatial reconstruction component, and developed the process of "Model Construction by Iteration Render" (MCIR). As illustrated in Figure 1, the MCIR process first initialises a 3-D spatial layout for all recognised objects, Figure 1(i), followed by the loop of Render-and-Update, Figure 1(ii,iv). The Render operation projects a 3-D layout into a 2-D image, Figure 1(c); the Update operation is carried out to reduce the difference between the original input image and the current rendered image. The result of the Comparison operation is always greater than or equal to zero.

We compare MC4VQA with NS-VQA in two experiments. The first experiment is performed using the original CLEVR dataset. MC4VQA achieved an accuracy of 99.94%. This outperforms all state-of-the-art methods, including NS-VQA. The aim of the second experiment is to examine whether traditional supervised learning endows neural-networks the ability to acquire 3-D spatial attributes from 2D images. We developed a new testing dataset, which contains 4000 images, generated by the CLEVR image generator from four different camera perspectives. Each scene is generated using a randomly selected camera configuration. NS-VQA had an overall accuracy of 98.39%. In contrast, our proposed method maintained another near-perfect accuracy at 99.8%. The success of MC4VQA not only demonstrates the power of the method of model construction and inspection for the acquisition of spatial knowledge (advocated in the psychological literature), but also shows the limitation of supervised deep learning – lacking the ability of generalisation of training patterns (Goyal and Bengio, 2022).

The contributions of MC4VQA are listed as follows: (1) it is the first VQA system that explicitly reconstructs 3-D spatial layout to bridge spatial linguistic descriptions and visual perception; (2) MC4VQA can be further developed by integrating more features of mental model theory in psychology, or used in psychological experiments; (3) Source code and new datasets are publicly accessible. The rest of the paper is structured as follows: Section 2 reviews a number of related works; Section 3 formalises the task of VQA by explicitly re-constructing 3-D spatial layout; Section 4 presents the detail of MC4VQA; Section 5 reports experiment results of MC4VQA, which grealy outperforms the SOTA performance, and demonstrates the power of the model construction method in new testing data; Section 5 concludes the paper, and lists a number of future research topics.

2 Related Work

A convergent opinion from linguistics, neuroscience, and psychology is that the spatial domain is the first domain that human babies understand, and is the reference domain for the understanding of other domains (Lakoff and Johnson, 1980; Regier, 1997; Grady, 1997; Tversky, 2019). The next generation of language system shall be a brain- and AI-inspired understanding system that explicitly represents situations (McClelland et al., 2020). Our work focuses on the NS-VQA model, and promises a novel method to explicitly represent scene images by constructing 3-D geometric spatial models. NS-VQA uses an older object detection model based on Detectron (Girshick et al., 2018) and Mask R-CNN (He et al., 2018). Since then, newer models with improved accuracy and speed have been released, such as YOLO (Redmon et al., 2016; Jocher et al., 2023), which produces impressive results and can be used for real-time video processing.

YOLO YOLO (You Only Look Once) is a powerful object detection model which is known for its speed and accuracy (Redmon et al., 2016). The current version of YOLO (v8) (Jocher et al., 2023) is the state-of-the-art object detection model that utilizes Cross Stage Partial (CSP) (Wang et al., 2019) architecture, which was introduced in YOLOv4 (Bochkovskiy et al., 2020). Our MC4VQA uses YOLOv8 as its object detection model. YOLO offers several pretrained models, of which we chose "YOLOv8x-seg" which has great segmentation accuracy.

Question Parsing and Execution Several papers have used program search and neural networks to recover programs from domain specific language (Neelakantan et al., 2016; Balog et al., 2017), including semantic parsing methods (Berant et al., 2013; Liang et al., 2011)to map sentences to logical forms from a knowledge base. Prior knowledge of semantics of the program and execution context is important to correctly parse an arbitrary set of question tokens following the semantics. So, the model needs the learn based on a set of input questions and answer pairs. NS-VQA's question parser follows the work done by (Andreas et al., 2016; Rothe et al., 2017; Goldman et al., 2019). The parser implementation uses a Bi-LSTM parser to generate programs from sentences similar to CLEVR-IEP (Johnson et al., 2017). The execution engine is slightly different from IEP, in the sense that it uses symbolic reasoning based on object positions generated by its attribute network.

Neural-symbolic approach to VQA NS-VQA stands for "Neural-symbolic Visual Question Answering" (Yi et al., 2019). Traditional neural-network approaches often do not have competitive performance on challenging reasoning tasks on CLEVR dataset (Johnson et al., 2016). In contrast, NS-VQA achieves a near-perfect accuracy on the CLEVR dataset, by learning a symbolic program from the question, and executing the program on an implicit spatial model learned by supervised deep learning, ResNet34 (He et al., 2015). It remains unclear whether NS-VQA's ResNet34 really learns the way to acquire 3D spatial relations from 2D images. The symbolic program may only match similar pairwise relationships in the training scene images. Furthermore, supervised models for generating 3D scene representations are prone to bias due to the invariant camera configuration used by the CLEVR training images.

3 Motivation of VQA through Model Construction and Inspection

Ever since Tolman's rats experiments (Tolman, 1948) in the 1940s, sufficient evidence has been collected to show that animals and humans can construct comprehensive spatial models in mind of their environments through sensorimotor interaction (Spelke and Lee, 2012) and that this spatial model in mind structures our language (Lakoff and Johnson, 1980; Tversky and Lee, 1999; Tversky, 2019). This motivates us to move one step ahead of NS-VQA by replacing its supervised ResNet34 component with a novel component that explicitly constructs 3D spatial layout, thus MC4VQA (Model Construction for VQA). This allows the symbolic program execution engine to more accurately identify objects and their spatial relationships in the scene. As being unsupervised, our method may improve the overall generalization of the scene construction, allowing to function on unknown camera configurations.

4 Formalising the task

In this section, we define the task of VQA through model construction and inspection. The input of MC4VQA consists of an image $\mathcal{I}$ and a question $\mathcal{Q}$ asking the content of this image, whose content can be described as a set of objects $\mathcal{I}_{O_1} \ldots \mathcal{I}_{O_n}$ and a set of 2D locations $\mathcal{L}_{O_i}$ of $\mathcal{I}_{O_i}$, line 1 in Algorithm 1. The process of model construction $\mathbf{P}$ will construct a 3D spatial layout $\mathcal{S}$ for $\mathcal{I}$. $\mathcal{S}$ consists of a set of 3D objects O_i with their size and their 3D location information.

Let $\mathcal{S}_0$ be an initial 3D layout, line 2 in Algorithm 1, the construction process $\mathbf{P}$ will update $\mathcal{S}_i$ to $\mathcal{S}_{i+1}$, with the following procedure: $\mathbf{P}$ will trigger an inspection function $\mathbf{I}$ to take a photo of $\mathcal{S}_i$, so called "rendering", let $\mathbf{I}(\mathcal{S}_i) = \mathcal{I}^{(i)}$. Then, a function $\mathbf{M}$ will measure the difference between $\mathcal{I}^{(i)}$ and the original image $\mathcal{I}$. Finally, a function $\mathbf{g}$ will apply a set of geometric operations on objects in $\mathcal{S}_i$. This transforms $\mathcal{S}_i$ into $\mathcal{S}_{i+1}$, so that a photo of $\mathcal{S}_{i+1}$ will be more similar to the original image, that is, $\mathbf{M}(\mathcal{I}^{(i+1)}, \mathcal{I}) < \mathbf{M}(\mathcal{I}^{(i)}, \mathcal{I})$. The construction process will stop, if $\mathbf{M}(\mathcal{I}^{(i+1)}, \mathcal{I})$ is less than a predefined threshold value ϵ. The final 3D layout $\mathcal{S}_n$ will be inspected to answer the question $\mathcal{Q}$ (Algorithm 1).

5 MC4VQA

MC4VQA has four components: an object detector (YOLOv8), a 3D model constructor (MCIR), a question parser (Bi-LSTM encoder), and a program executor.

Object Detection The YOLOv8 object detector is trained on the same 4000 CLEVR images used by NS-VQA. The input image is first passed to the object detector to generate object proposals. The object proposals are composed of the predicted object masks and the object bounding boxes, along with their class names. Object proposals with a score of less than 0.9 are discarded. The predicted class names are composed of the discrete attributes of the objects, e.g., the object size, colour, material, and shape. These attributes are used to construct the 3D scene and to answer the questions.

Algorithm 1: VQA by 3D model construction and inspection

Input: an image $\mathcal{I}$;
Input: a question $\mathcal{Q}$ about the content of $\mathcal{I}$;
Output: an answer $\mathcal{A}$ to $\mathcal{Q}$;
1 recognise 3D objects $O_1 \ldots O_n$ in $\mathcal{I}$;
2 Initialise 3D spatial layout $\mathcal{S}_c$ by placing all O_i at the same location;
3 $\mathcal{I}^{(c)} \leftarrow \mathbf{I}(\mathcal{S}_c)$;
4 **while** *$\mathcal{I}^{(c)}$ not similar with $\mathcal{I}$* **do**
5   update 3D locations and postures of objects O_i in $\mathcal{S}_c$, to increase the similarity to $\mathcal{I}$; ▷ reduce the value $\mathbf{M}(\mathcal{I}_c) - \mathbf{M}(\mathcal{I})$
6     $\mathcal{I}^{(c)} \leftarrow \mathbf{I}(\mathcal{S}_c)$; ▷ $\mathcal{I}^{(c)}$ is a photo of $\mathcal{S}_c$
7 $\mathcal{A} \leftarrow$ answer $\mathcal{Q}$ by inspecting 3D layout $\mathcal{S}_c$;
8 **return** $\mathcal{A}$

3D Model Construction The object proposals generated by the object detector are passed to MCIR, which processes the bounding boxes of the objects to compute more realistic box midpoints. The bounding boxes from the object detector do not take into account occlusion behind other objects, so it is important to correct them before generating the 3D scene.

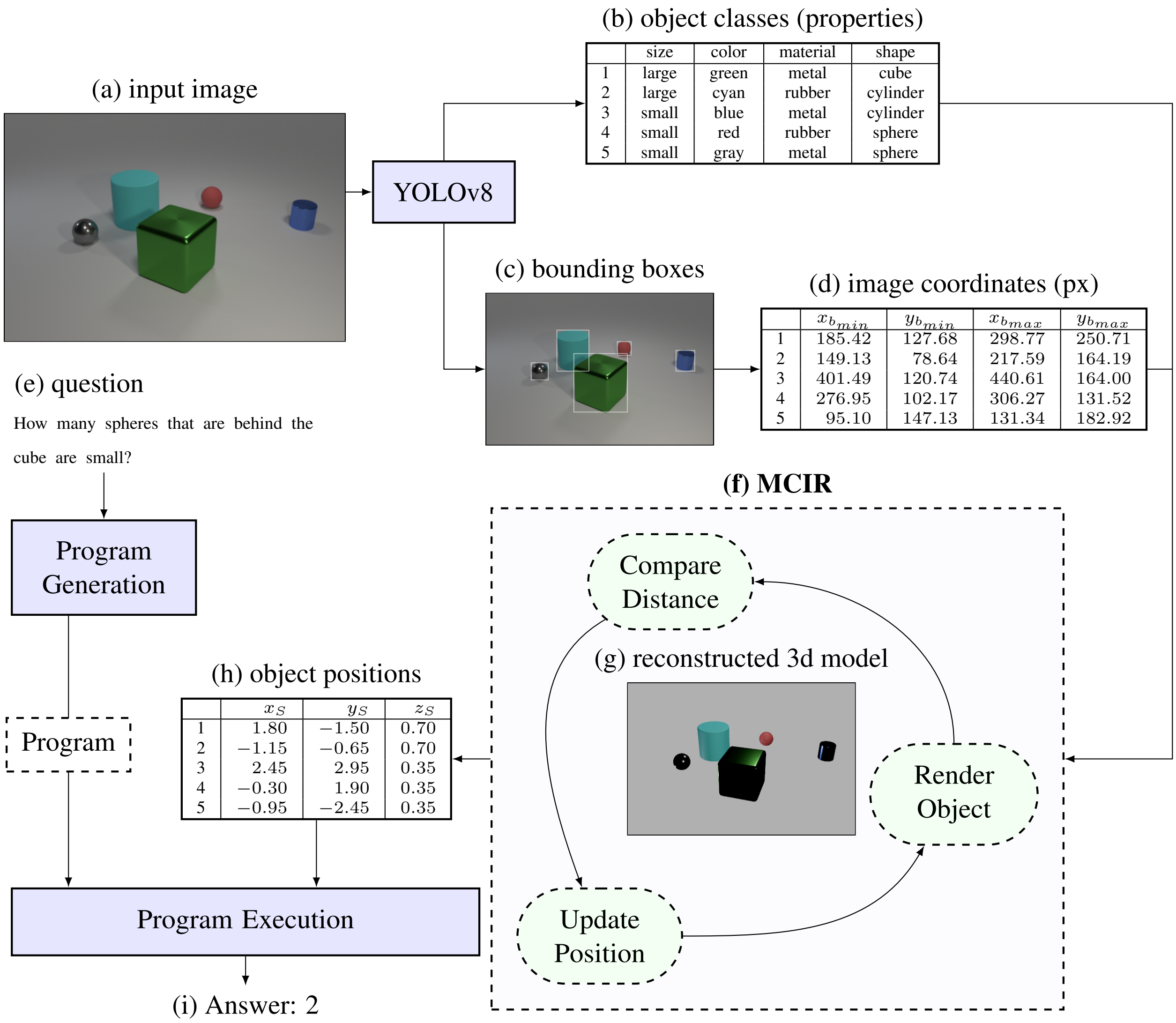
After the approximately realistic midpoints are generated, they are passed to MCIR, which generates the 3D spatial model. This model is then passed to the question executor as the scene representation of the input image.

Question Parsing and Program Execution The question parser and the program executor used by MC4VQA are both directly taken from the NS-VQA implementation without any changes. The output format of MCIR is compatible with the input format of the program executor, so they integrate well with each other. The reconstructed 3-D representation is used to generate the answers.

6 Experiments

A series of experiments are conducted to compare the methods of model construction and of supervised learning for VQA.

Experiment I MC4VQA is implemented by replacing NS-VQA's supervised learning model with a model of 3D scene construction



	size	color	material	shape
1	large	green	metal	cube
2	large	cyan	rubber	cylinder
3	small	blue	metal	cylinder
4	small	red	rubber	sphere
5	small	gray	metal	sphere

	$x_{b_{min}}$	$y_{b_{min}}$	$x_{b_{max}}$	$y_{b_{max}}$
1	185.42	127.68	298.77	250.71
2	149.13	78.64	217.59	164.19
3	401.49	120.74	440.61	164.00
4	276.95	102.17	306.27	131.52
5	95.10	147.13	131.34	182.92

	x_S	y_S	z_S
1	1.80	−1.50	0.70
2	−1.15	−0.65	0.70
3	2.45	2.95	0.35
4	−0.30	1.90	0.35
5	−0.95	−2.45	0.35

Figure 2: Overview of NS-VQA Extended with Iterative Rendering

to acquire spatial attributes, and share the same object detection model and the same model of question parsing and program execution.

We used three camera configurations to test the performance of MC4VQA as follows: (1) C1 was a random configuration to serve as a baseline; (2) C2 was chosen to simulate the camera direction that a human would likely choose when looking at the CLEVR images; (3) C3 was calculated based on the average of the first ten camera directions specified in the CLEVR scenes to represent a manually fine-tuned camera configuration.

YOLO for object proposals We trained a YOLOv8 object detector on the same 4000 CLEVR images. These are the same images used to train the object proposal model of NS-VQA in (Yi et al., 2019). Object proposals with a score of less than 0.9 were discarded. A predicted class name consists of discrete attributes of the object, such as the size, the colour, the material, and the shape. These attributes are used to construct the 3D scene and to answer the questions using the program executor. The training of the YOLOv8 model was run on resized image size of 480x480 for 100 epochs with a learning rate of 0.01.

Equipped with this YOLO model, NS-VQA (Yi et al., 2019) improves its overall accuracy from 99.8% to 99.93%, as listed in Table 1.

VQA through 3-D Model Construction MC4VQA uses YOLO object proposals to initialise a 3-D layout, then repeatedly optimizes this layout by reducing the difference between the objects in the input image and the objects in the 3-D scene generated by the rendering engine. Then, MC4VQA uses NS-VQA's question parser to generate programs and apply them to the 3D layout to generate answers, whose correctness is validated by the ground

Methods	Count	Exist	Compare Number	Compare Attribute	Query Attribute	Overall
Humans	86.7	96.6	86.5	95.0	96.0	92.6
MDETR (Kamath et al., 2021)	99.3	99.9	99.4	99.9	99.9	99.7
NMN (Andreas et al., 2017)	52.5	72.7	79.3	79.0	78.0	72.1
N2NMN (Hu et al., 2017)	68.5	85.7	84.9	90.0	88.7	83.7
IEP (Johnson et al., 2017)	92.7	97.1	98.7	98.1	98.9	96.9
TbD (Mascharka et al., 2018)	97.6	99.4	99.2	99.5	99.6	99.1
RN (Santoro et al., 2017)	90.1	93.6	97.8	97.1	97.9	95.5
FiLM (Perez et al., 2017)	94.5	93.8	99.2	99.2	99.0	97.6
NS-CL (Mao et al., 2019)	98.2	99.0	98.8	99.3	99.1	98.9
MAC (Hudson and Manning, 2018)	97.2	99.4	99.5	99.3	99.5	98.9
OCCAM (Wang et al., 2021)	98.1	99.8	99.0	99.9	99.9	99.4
NS-VQA (Yi et al., 2019)	99.7	99.9	99.9	99.8	99.8	99.8
NS-VQA (YOLOv8)	99.87	99.96	99.93	99.93	99.95	99.93
MC4VQA [*C1*]	99.89	99.97	**99.94**	99.91	99.92	99.92
MC4VQA [*C2*]	**99.92**	**99.98**	99.93	99.94	**99.95**	**99.94**
MC4VQA [*C3*]	**99.92**	99.97	99.93	**99.97**	99.94	**99.94**

Table 1: NS-VQA outforms state-of-the-art methods on the CLEVR dataset. With introduction of the YOLO model the accuracy is improved. Integrating with iterative render further improves the accuracy to a near perfect 99.94%. Our model depends on the camera configuration of the system. C^1 is a random configuration to serve as a baseline. C^2 is chosen to simulate the camera direction that a human would likely choose when looking at the CLEVR images. C^3 is calculated based on the average of the first ten camera directions specified in the CLEVR scenes to represent a manually fine-tuned camera configuration.

Methods	Count	Exist	Compare Number	Compare Attribute	Query Attribute	Overall
NS-VQA	97.86	99.03	99.22	98.53	98.21	98.39
MC4VQA	**99.52**	**99.85**	**99.97**	**99.90**	**99.88**	**99.80**

Table 2: NS-VQA (YOLOv8) with attribute net performs slightly worse at 98.39% than MC4VQA (YOLOv8) with MCIR, which still maintains near perfect accuracy at 99.80%

truth in the validation set. The performance is measured in terms of the accuracy.

Results and Analysis Experiment results show that MC4VQA reaches 99.94% overall accuracy on the benchmark CLEVR dataset without training data. This outperforms the SOTA NS-VQA (Yi et al., 2019) and the NS-VQAv8 (NS-VQA with YOLO model). Experiments also show that MC4VQA reaches the performance of NS-VQAv8 in each evaluation task, at least from one camera configuration. We conclude that MC4VQA successfully acquired spatial attributes by utilising the method of 3D model construction without training data.

Experiment results show that MC4VQA reaches 99.94% accuracy on the benchmark CLEVR dataset, without training data. This outperforms the SOTA NS-VQA (Yi et al., 2019) and the NS-VQAv8 (NS-VQA with YOLO model). Experiments also show that MC4VQA reaches the performance of NS-VQA at least from one camera configuration for rendering. We conclude that *by utilising the method of 3D model construction, MC4VQA successfully acquired spatial attributes without training data.*

Experiment II In Experiment I, the testing and training data are from benchmark CLEVR dataset, sharing the same distribution. The second experiment compares the performances of the well-trained NS-VQA and MC4VQA on new test datasets.

Design of the experiment We generated 4000 CLEVER images with four different camera configuration, and 40000 questions, and fed them to the well-trained NS-VQA with YOLOv8 and MC4VQA.

Experiment Results show that the overall performance of NS-VQA drops from 99.93% to 98.39% and that the overall performance of MC4VQA slightly drops from 99.94% to 99.80%, Table 2. This suggests our method is more robust than NS-VQA.

Error Analysis We examined cases when NS-VQA made mistakes. In Figure 3, NS-VQA fails to locate the small gray cube accurately, resulting in an incorrect answer. MC4VQA overcomes this limitation by using corrected bounding boxes and a 3D spatial model to cor-

Algorithm 2: The simple MCIR Algorithm

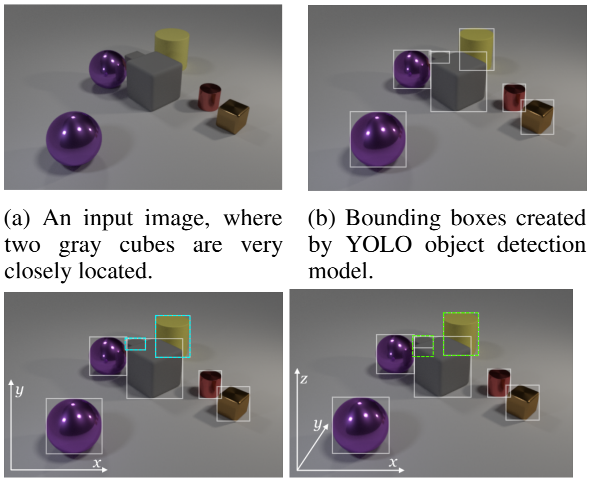
Input: object proposals from YOLO
Data: o_{max}- total number of objects
Data: j_{max}- maximum number of iterations

$o_i \leftarrow 1$; /* o_i: current object index */
while $o_i \leq o_{max}$ **do**
  $j \leftarrow 1$;
  $O \leftarrow$ objects$[o_i]$;
  $C \leftarrow$ box-midpoints$[o_i]$;
  $S \leftarrow$ initialize(O);
  $I \leftarrow$ project(S); /* $I \sim (x_I, y_I)$: 2-D image coordinates of O (current) */
  $d \leftarrow |C - I|$; /* d: pixel distance */
  $u_p \leftarrow 1$; /* u_p: previously used update value */
  while $d > d_{threshold}$ **do**
    $u_i \leftarrow u_p$; /* u_i: index of update value */
    while $j \leq j_{max}$ **do**
      /* U: set of available update values */
      /* u_{max}: number of update values */
      $u \leftarrow U[u_i \bmod u_{max}]$;
      $S_c \leftarrow S + u$; /* S_c: candidate scene coordinate */
      $I_c \leftarrow$ project(S_c); /* I_c: candidate image coordinate */
      $d_c \leftarrow |C - I_c|$; /* d_c: new pixel distance */
      if $d_c < d$ **then**
        $S \leftarrow S_c; I \leftarrow I_c; d \leftarrow d_c$;
        $u_p \leftarrow u_i$;
        break
      $u_i \leftarrow u_i + 1$
  $o_i \leftarrow o_i + 1$

rectly identify the cube's location. NS-VQA made similar mistakes when there are objects very close to together each other. We hypothesize that the performance of NS-VQA drops if the questions are about closely situated objects. We report Experiment III as follows.

Experiment III We create a new testing dataset, in which some objects are very close to each other, and evaluate the performances of NS-VQA and MC4VQA.

Design of the experiment Two sets of CLEVR images were created, 1000 images for each, as follows.



(a) An input image, where two gray cubes are very closely located.

(b) Bounding boxes created by YOLO object detection model.

(c) 2D spatial attribute used by NS-VQA

(d) 3D spatial layout used by MC4VQA

Figure 3: (a) Given an input image and the question "what number of objects are behind the small brown metallic thing and in front of the yellow metta object?" (b) YOLO successfully identifies all objects with bounding boxes. In (c) NS-VQA uses 2D YOLO bounding boxes. In this case, the small gray cube is not calculated as being in front of the yellow cylinder. (d) MC4VQA used its constructed 3-D spatial layout, instead of 2D YOLO bounding boxes, and correctly calculated the small gray cube being in front of the yellow cylinder.

- In one set, there are two objects being very close to each other; (minimum distance between two objects is 0.1 units, as opposed to CLEVR default of 0.4 units)
- In another set, at least two objects are close, and all objects are less spread out in the scene. (maximum coordinates along the axes: 2.0 units, as opposed to CLEVR default of 3.0 units)

These two testing datasets were fed to NS-VQA and MC4VQA.

Results an analysis The performance of NS-VQA continued to decrease to below 98.0%. The performance of MC4VQA decreased slightly, and still reached 99.0% in both testing datasets, as listed in Tables 3 and 4, respectively.

Limitations of MC4VQA Our MCIR process optimises a 3D layout through reducing the difference between a rendered image and the input image. It does not have other spatial constraints, such as extended 3D objects cannot be partially overlapped. This limitation will cause MC4VQA to construct incorrect 3D layout. For example, Figure 4 illustrates a new testing image whose camera configuration is very near to the objects. This causes the effect of plac-

Methods	Count	Exist	Compare Number	Compare Attribute	Query Attribute	Overall
NS-VQA	96.54	98.48	98.97	99.47	97.44	97.90
MC4VQA	**98.70**	**100.00**	**97.94**	**100.00**	**99.43**	**99.30**

Table 3: NS-VQA vs MC4VQA when the objects are closer to each other.

Methods	Count	Exist	Compare Number	Compare Attribute	Query Attribute	Overall
NS-VQA	95.67	99.24	96.91	98.94	97.73	97.60
MC4VQA	**98.70**	**100.00**	**98.97**	**99.47**	**98.58**	**99.00**

Table 4: NS-VQA vs MC4VQA when the objects are close and less spread out.

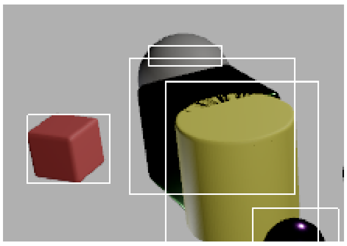
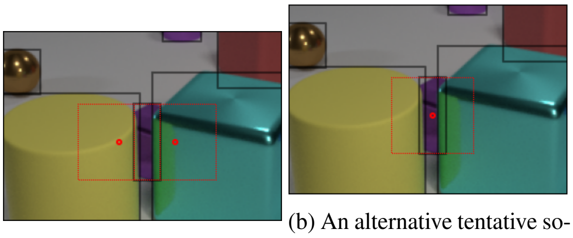


Figure 4: When objects are very close to each other in a 3D layout, they may be partially overlapped, as we see there is a yellowish black at the edge of the top surface of the yellow cylinder behind it.



(a) Left-most or right-most bounding-boxes can be used as tentative solutions.

(b) An alternative tentative solution is to put the object to the centre of the bounding box.

Figure 5: A purple object is occluded by two big objects, whose location is hard to figure.

ing large 3D objects in a relative small place. Without explicit spatial constraints, nearby 3D objects can be partially overlapped.

Another limitation of the MCIR system is using single camera configuration. Under certain situations, it might not be possible to figure out the precise location of an object in the 3D layout. For example, Figure 5(a) illustrates an image, in which a purple object is behind a big yellow cylinder and a green cuboid, only a very small part can be seen. Although this small part is sufficient to recognise what object class and what size it is, figuring out its precise location will be hard. Tentative solutions can be to set the bounding box as left (or right) as possible, Figure 5(a), or let the centre of the bounding box and the seen part be coincided, Figure 5(b). Each tentative solution can cause MC4VQA to give incorrect answers.

7 Conclusions and outlooks

Understanding surrounding environment is a fundamental ability for the survival of animals and humans, e.g., to escape from dangerous predators. It is a challenging research task in NLU and AI, and has various downstream applications, e.g., autonomous driving, service robots. VQA with the benchmark CLEVR dataset is a micro-world to explore this field, in which images are about layouts of synthesised geometric objects. Supervised neural networks to learn spatial attributes are very successful, with two conditions: (1) it needs a huge amount of training data; (2) the testing data shall have the same distribution as the training data. Both conditions are either expensive or unrealistic for real applications. We replace the method of supervised learning with the method of model construction to free the acquisition of spatial attributes from the imprisonment of data and go beyond the paradigm of supervised learning.

Our experiment results show that our new method is very promising – it does not need training data for acquiring spatial regions and achieves higher accuracy in answering questions about out-of-distribution scenes.

In this work, we implemented MCIR using a simple object-level loop to optimize object locations and used NS-VQA's question parser and executor with the CLEVR validation questions. In the future, we will adopt a dual-camera configuration to figure out the locations of 3D objects precisely and will use the constructed 3D layout construction as the spatial semantics to interpret linguistic descriptions.

References

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. 2016. Vqa: Visual question answering. *Preprint*, arXiv:1505.00468.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. *Preprint*, arXiv:1601.01705.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2017. Neural module networks. *Preprint*, arXiv:1511.02799.

Matej Balog, Alexander L. Gaunt, Marc Brockschmidt, Sebastian Nowozin, and Daniel Tarlow. 2017. Deepcoder: Learning to write programs. *Preprint*, arXiv:1611.01989.

Jacob L. S. Bellmund, Peter Gärdenfors, Edvard I. Moser, and Christian F. Doeller. 2018. Navigating cognition: Spatial codes for human thinking. *Science*, 362(6415).

Jonathan Berant, Andrew K. Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Conference on Empirical Methods in Natural Language Processing*.

C. Biever. 2023. The easy intelligence tests that AI chatbots fails. *Nature*, 619:686–689.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *Preprint*, arXiv:2004.10934.

Peter Gädenfors. 1990. The dynamics of belief systems: Foundations vs. coherence theories. *Revue Internationale de Philosophie*, 44(172 (1)):24–46.

P. Gärdenfors. 1988. *Knowledge in Flux. Modelling the Dymanics of Epistemic States*. MIT Press.

Gerd Gigerenzer. 2022. *How to Stay Smart in a Smart World: Why Human Intelligence Still Beats Algorithms*. The MIT Press.

Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. 2018. Detectron. `https://github.com/facebookresearch/detectron`.

Omer Goldman, Veronica Latcinnik, Udi Naveh, Amir Globerson, and Jonathan Berant. 2019. Weakly-supervised semantic parsing with abstract examples. *Preprint*, arXiv:1711.05240.

Geoffrey Goodwin and Phil Johnson-Laird. 2005. Reasoning about relations. *Psychological review*, 112:468–93.

Anirudh Goyal and Y. Bengio. 2022. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478.

J. Grady. 1997. *Foundations of Meaning: Primary Metaphors and Primary Scenes*. University Microfilms.

Gilbert Harman. 1986. *Change in View: Principles of Reasoning*. Cambridge, MA, USA: MIT Press.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2018. Mask r-cnn. *Preprint*, arXiv:1703.06870.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *Preprint*, arXiv:1512.03385.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. *Preprint*, arXiv:1704.05526.

Drew A. Hudson and Christopher D. Manning. 2018. Compositional attention networks for machine reasoning. *Preprint*, arXiv:1803.03067.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023. YOLO by Ultralytics.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2016. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *Preprint*, arXiv:1612.06890.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. Inferring and executing programs for visual reasoning. *Preprint*, arXiv:1705.03633.

P. N. Johnson-Laird and R. M. J. Byrne. 1991. *Deduction*. Lawrence Erlbaum Associates, Inc.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. Mdetr – modulated detection for end-to-end multi-modal understanding. *Preprint*, arXiv:2104.12763.

M. Knauff, T. Fangmeier, C. C. Ruff, and P. N. Johnson-Laird. 2003. Reasoning, models, and images: behavioral measures and cortical activity. *Journal of Cognitive Neuroscience*, 15(4):559–573.

Markus Knauff. 2009. A neuro-cognitive theory of deductive relational reasoning with mental models and visual images. *Spatial Cognition & Computation*, 9(2):109–137.

Markus Knauff. 2013. *Space to Reasong*. MIT Press.

Markus Knauff, Leandra Bucher, Antje Krumnack, and Jelica Nejasmic. 2013. Spatial belief revision. *Journal of Cognitive Psychology*, 25(2):147–156.

G. Lakoff and M. Johnson. 1980. *Metaphors We Live By*. The University of Chicago Press, Chicago. Citation is based on the reprinted in 2003.

Percy Liang, Michael I. Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. *Preprint*, arXiv:1109.6841.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *Preprint*, arXiv:1904.12584.

David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. 2018. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE.

James McClelland, Felix Hill, Maja Rudolph, Jason Baldridge, and Hinrich Schütze. 2020. Placing language in an integrated understanding system: Next steps toward human-level performance in neural language models. *Proceedings of the National Academy of Sciences of the United States of America*, 117.

Arvind Neelakantan, Quoc V. Le, and Ilya Sutskever. 2016. Neural programmer: Inducing latent programs with gradient descent. *Preprint*, arXiv:1511.04834.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. 2017. Film: Visual reasoning with a general conditioning layer. *Preprint*, arXiv:1709.07871.

Marco Ragni and Markus Knauff. 2013. A theory and a computational model of spatial reasoning with preferred mental models. *Psychological review*, 120:561–588.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. *Preprint*, arXiv:1506.02640.

T. Regier. 1997. *The Human Semantic Potential: Spatial Language and Constrained Connectionism*. The MIT Press, Cambridge, Massachusetts.

Anselm Rothe, Brenden M. Lake, and Todd M. Gureckis. 2017. Question asking as program generation. *Preprint*, arXiv:1711.06351.

Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. *Preprint*, arXiv:1706.01427.

Elizabeth Spelke and Sang Ah Lee. 2012. Core systems of geometry in animal minds. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 367:2784–93.

E. C. Tolman. 1948. Cognitive Maps in Rats and Men. *The Psychological Review*, 55(4):189–208.

B. Tversky and P. Lee. 1999. How space structures language. In C. Freksa, C. Habel, and K. F. Wender, editors, *Spatial Cognition*, volume 1404 of *LNAI*, pages 157–176. Springer-Verlag.

Barbara Tversky. 2019. *Mind in Motion*. Basic Books, New York, USA.

Chien-Yao Wang, Hong-Yuan Mark Liao, I-Hau Yeh, Yueh-Hua Wu, Ping-Yang Chen, and Jun-Wei Hsieh. 2019. Cspnet: A new backbone that can enhance learning capability of cnn. *Preprint*, arXiv:1911.11929.

Zhonghao Wang, Kai Wang, Mo Yu, Jinjun Xiong, Wen mei Hwu, Mark Hasegawa-Johnson, and Humphrey Shi. 2021. Interpretable visual reasoning via induced symbolic space. *Preprint*, arXiv:2011.11603.

Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2016. Visual question answering: A survey of methods and datasets. *Preprint*, arXiv:1607.05910.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. 2019. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Preprint*, arXiv:1810.02338.

Yeyun Zou and Qiyu Xie. 2020. A survey on VQA: Datasets and approaches. In *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. IEEE.