VSA4VQA: Scaling a Vector Symbolic Architecture to Visual Question Answering on Natural Images

Anna Penzkofer (anna.penzkofer@vis.uni-stuttgart.de)
University of Stuttgart, Institute for Visualization and Interactive Systems (VIS), Germany

Lei Shi (lei.shi@vis.uni-stuttgart.de)
University of Stuttgart, Institute for Visualization and Interactive Systems (VIS), Germany

Andreas Bulling (andreas.bulling@vis.uni-stuttgart.de)
University of Stuttgart, Institute for Visualization and Interactive Systems (VIS), Germany

Abstract

While Vector Symbolic Architectures (VSAs) are promising for modelling spatial cognition, their application is currently limited to artificially generated images and simple spatial queries. We propose *VSA4VQA* – a novel 4D implementation of VSAs that implements a mental representation of natural images for the challenging task of Visual Question Answering (VQA). VSA4VQA is the first model to scale a VSA to complex spatial queries. Our method is based on the Semantic Pointer Architecture (SPA) to encode objects in a hyper-dimensional vector space. To encode natural images, we extend the SPA to include dimensions for object's width and height in addition to their spatial location. To perform spatial queries we further introduce learned spatial query masks and integrate a pre-trained vision-language model for answering attribute-related questions. We evaluate our method on the GQA benchmark dataset and show that it can effectively encode natural images, achieving competitive performance to state-of-the-art deep learning methods for zero-shot VQA.

Keywords: vector symbolic architecture, spatial semantic pointer, spatial queries, visual question answering

Introduction

Vector Symbolic Architectures (VSAs) have shown significant potential for cognitive modelling (Stewart, Choo, & Eliasmith, 2012; Komer, Stewart, Voelker, & Eliasmith, 2019; Lu, Voelker, Komer, & Eliasmith, 2019; Bartlett, Stewart, & Orchard, 2022; Hersche, Zeqiri, Benini, Sebastian, & Rahimi, 2023). At the core of VSAs are hyper-dimensional vectors as well as adding and binding operations to build vector compositions that can be reversed and disentangled with almost no loss. Representing concepts or symbols using these vectors enables VSAs to model cognitive processes with compositionality and systematicity (Plate, 2003). While VSAs have been used to facilitate abstract reasoning on various tasks, such as Raven's Progressive Matrices (Choo, 2018; Hersche et al., 2023), they have also been shown to efficiently encode continuous spaces for building mental image representations (Komer et al., 2019; Lu et al., 2019) or to improve the robustness in 2D spatial navigation tasks (Bartlett et al., 2022).

However, existing VSA models have only been evaluated on artificially generated images with icons (Komer et al., 2019) or MNIST digits (Lu et al., 2019), where simple questions about spatial relations between objects were used to evaluate their spatial memory capabilities. While underlining the significant potential of VSAs, it is unknown whether these results generalise to images of natural scenes: Natural images are significantly more visually complex and typically contain numerous objects of different size and appearance. In addition, natural images exhibit more complex spatial relations between objects that are not covered in artificial images, e.g. objects can be *in front of* or *behind* other objects. Answering questions on natural images thus requires a more general understanding of objects' spatial relations (Banerjee, Gokhale, Yang, & Baral, 2021), which is still lacking in current approaches (Subramanian et al., 2022).

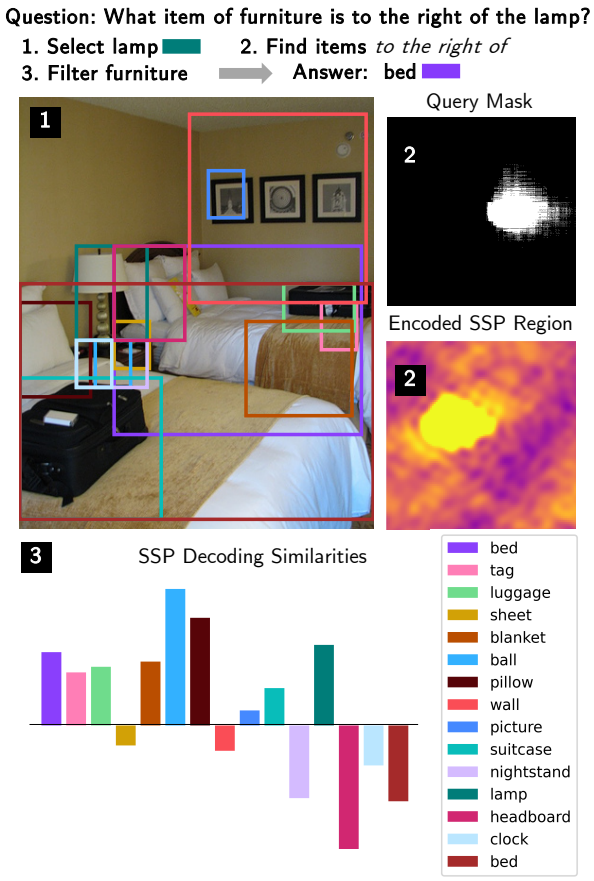

Figure 1: Example question from GQA (Hudson & Manning, 2019). Our *VSA4VQA* method performs three steps: 1. select the lamp, 2. find items *to the right of* the lamp with a spatial query mask encoded in SSPs, and 3. filter the positive proposals to find furniture, yielding the correct answer "bed".

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.*

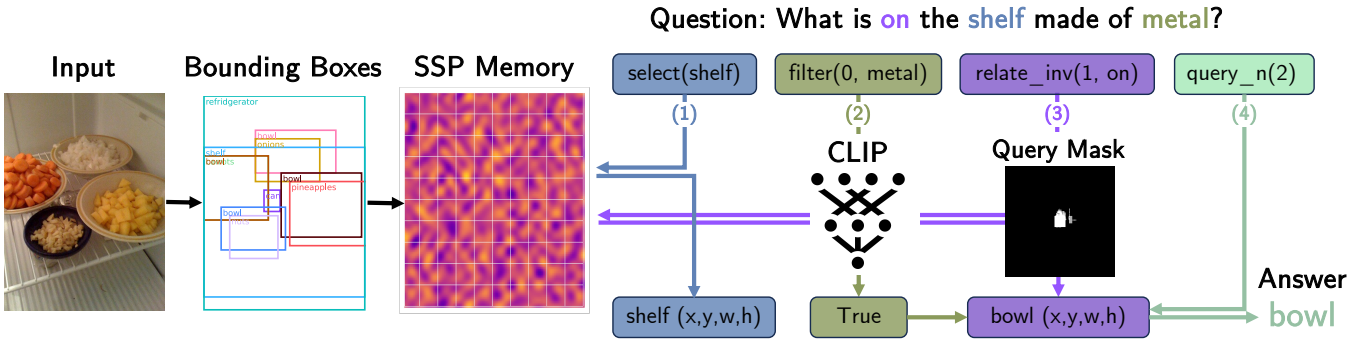

Figure 2: Overview of *VSA4VQA*. Object bounding boxes' (x,y)-location, width w, and height h are encoded into SSP memory. The program generator from Chen et al. (2020) maps the question to functions. Our method then implements these functions with (1) SSP unbinding, (2) CLIP, (3) SSP query masks, and finally queries the name of the resulting object (4) to answer the question. For implementation details on all functions see Table 1.

We propose *VSA4VQA* – the first VSA model that can answer complex spatial queries on natural images. VSA4VQA builds on the Semantic Pointer Architecture (SPA) proposed in (Eliasmith, 2013). The SPA encodes spatial semantic pointers (SSPs) – hyper-dimensional vectors – to encode spatial locations in two dimensions. By encoding objects in an image and binding them to their spatial locations, a cognitively plausible mental representation of the image can be created (Komer et al., 2019). Further, the SPA allows for querying this mental image representation with spatial regions in relation to other objects. While previously these spatial queries have only been tested with simple rectangular regions representing the four quadrants of an image, we introduce a novel method to learn query masks that encode spatial relations in natural images [1].

To evaluate the mental image representation generated by VSA4VQA, we report experiments on GQA (Hudson & Manning, 2019) – a widely used Visual Question Answering (VQA) dataset. VQA is a challenging multimodal task that involves answering questions by reasoning about visual information contained in an image (Cao & Jiang, 2023). VQA is particularly promising as an evaluation task because it requires a compositional understanding of spatial relations (Banerjee et al., 2021). To implement compositional reasoning, our method uses programs generated by a seq-to-seq model (Chen et al., 2020) and assigns each function in the program to a dedicated module (Andreas, Rohrbach, Darrell, & Klein, 2016). For functions that query attributes of objects, such as colour or shape, we additionally integrate a pre-trained vision and language model (Radford et al., 2021).

Taken together, the specific contributions of our work are three-fold: (1) We scale a VSA to model cognitively plausible representations of natural images. We further propose zero-shot VQA as a particularly suitable task to evaluate the resulting mental image representation in terms of their usefulness for reasoning with complex spatial queries. (2) We introduce 37 novel spatial query masks that we learn from relation annotations in the GQA dataset and map them to more than 300 spatial relations. (3) We report extensive analyses on error cases and the impact of the dimensionality of VSA vectors on the performance.

[1] All query masks and the model code are available at https://perceptualui.org/publications/penzkofer24_cogsci

Method

Our method performs VQA on natural images by implementing a VSA as image representation. Figure 2 shows an overview of our method. First, we encode the bounding boxes of all objects in the image into SSP memory. Then, we use programs that map questions to functions (Chen et al., 2020) to perform sequential reasoning. To verify attributes that are not present in SSP memory we incorporate CLIP (Radford et al., 2021), a pre-trained vision-language model. For relation queries we use our learned spatial query masks.

Image Encoding

For building a cognitively plausible mental representation of natural images, we extend the fractional binding method of SSPs to encode additional dimensions for width and height of objects. To this end, we adjust the mathematical formulation, introduced by (Komer et al., 2019), to include a total number of four dimensions:

$$\mathrm{SSP} = \mathrm{SP} \circledast S(x,y,w,h) \tag{1}$$

$$S(x,y,w,h) = X^x \circledast Y^y \circledast W^w \circledast H^h \tag{2}$$

$$M = \sum_{i=1}^{m} \mathrm{SP}_i \circledast S(x_i,y_i,w_i,h_i) \tag{3}$$

Here, a semantic pointer (SP) is a hyper-dimensional vector with fixed dimension, which is a compressed representation of an object in the image. x and y are the object's spatial locations in the coordinate system of the image, while w and h encode width and height of the object's bounding box. Binding all SPs representing an object with the object's location, width, and height and summing them, yields the mental image representation M, further referred to as SSP memory.

Before encoding images, we set four random SP vectors as (X, Y, W, H)-axes and pre-compute all location vectors in the SSP vector space. This yields a discretised grid of $100\times100\times10\times10$ points and constitutes our clean-up memory (see (Lu et al., 2019) for details). The images in GQA vary in size and orientation, therefore, the longer side is always selected to be scaled to fit within the 100×100 vector space. For building the SSP memory M, we need bounding boxes for each object in the image. For this, we use the ground truth object annotations given in the scene graphs of the GQA dataset.

For each detected object we generate a random SP. We then compute the objects point $S(x,y,w,h)$ as defined in Equation 2 and bind the object's SP and point S together as described in Equation 1. We add each object to the image's SSP memory using element-wise addition, which is the superposition operation in Holographic Reduced Representations (HRR).

Query Masks

In previous work (Komer et al., 2019; Lu et al., 2019) it was shown that regions can be encoded as a sum of spatial locations. Such encoded regions were then used to query all objects located within the respective area. However, these region queries were limited to simple rectangles encoding the four quadrants of an image. In contrast, we learn representations for 37 unique query masks, which enables the encoding of more than 300 spatial relations.

The query masks are generated from the relation annotations available in the scene graphs of the training split of GQA (Hudson & Manning, 2019). The dataset provides 50.65 relations per image on average, while samples per relation range from two (*shorter than* / *bigger than*) to over one million (*to the right of* / *to the left of*). For each relation with more than 1,000 samples we generate a spatial query mask by adding up all samples of relative objects, i.e. the objects that the relation applies to. However, before adding each sample, all objects need to be normalised: (1) both objects bounding boxes are scaled so that the bounding box of all queried objects is uniform (50×50 pixels) and (2) the new position of this bounding box is extracted and used to calculate the translation vector for moving it into the centre of a 500×500px image.

In Figure 3, we show the full process of the query mask generation for relation *to the right of*. After normalisation, the areas of all objects that are in relation to the queried object are summed up and averaged. Thresholding the resulting image to exclude areas that have only appeared in less than 5% of samples, yields the binary query masks depicted in the rightmost image. The rest of the 37 query masks are obtained in the same way and yield qualitatively plausible representations of the respective relations.

Programs & Modules

To answer natural language questions on images, we follow the Neural Module Network (NMN) paradigm (Andreas et al., 2016), where questions are mapped to programs that can be executed in sequential steps by separate modules. Similar to Mod-Zero-VQA (Cao & Jiang, 2023), the state-of-the-art deep learning approach for zero-shot VQA, we use the programs generated by the pre-trained sequence-to-sequence model of Chen et al. (2020). For the GQA dataset (Hudson & Manning, 2019), the programs consist of 10 different function types (see Table 1) and 48 fine-grained functions, which are chained together yielding a list of steps to be performed sequentially. For each step, the result of the respective function is saved for access in the following steps. Almost all programs start with the selection of an object, only exceptions are "full" functions that query the entire scene, which are denoted with an `_f`. The lengths of programs are up to 9 steps. While the functions `relate`, `filter`, and `verify` produce outputs that can be reused in following steps, `choose` and `query`, directly yield answers and terminate the program.

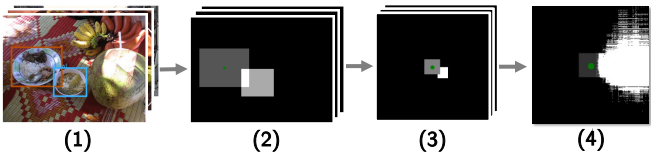

Figure 3: Query mask generation for relation *to the right of*. For all samples of the relation (1): create object masks (2), normalise masks to same scale and center position (3), and add them to obtain final relation mask (4).

For example in Figure 2, the question "What is on the shelf made of metal?" is mapped into four functions: `select(shelf)`, `filter(0, metal)`, `relate_inv(1, on)`, and `query_name(2)`. The result of the selection from SSP memory is reused in step two, where it is filtered to be made from metal. If indeed it is a metal shelf, the program continues with step three, where all objects are queried that are on the shelf. This returns the SSP of a bowl as the most likely candidate, which is returned in the final step.

We implement five of the program functions with our SSP memory. With the image encoded in the SSP memory, we can use the unbinding operation $M \circledast OBJ^{-1}$ to perform the `select` function for any encoded object. In general, the unbinding operation yields the respective SSP and the object's encoded (x,y,w,h)-coordinates. For performing relation queries, our generated query masks are loaded and encoded as a region in SSPs, then, shifted to the correct position. The region query returns all objects that are positively similar to the queried region as proposals. If the function does not specify the relative object it is looking for, the proposal with the highest similarity is returned. If there is an additional attribute in the function that specifies the name of the queried object, the proposal that matches the name or the most similar object that belongs to the named class is returned.

CLIP Integration

For functions that require attribute verification, we employ the pre-trained model CLIP (Radford et al., 2021). CLIP is

Table 1: Program function types and our implementation. Functions within one type are sorted by the number of arguments. For details on function types `exist`, `common`, `different`, `same`, `and`, `or` see the original implementation by Chen et al. (2020).

Type	Function	Implementation	Args	Output	Example
Select	`select`	SSP Unbinding	1	Position	`select(tool)`
Relate	`relate`, `relate_inv`	SSP Query Mask	2	Proposal	`relate(basket, on)`
	`relate_name`, `relate_inv_name`	SSP Query Mask	3	Proposal	`relate_name(man, with, hat)`
Filter	`filter_v`, `filter_h`	Position	2	True / False	`filter_h(bottle, left)`
	`filter`, `filter_not`	CLIP	2	True / False	`filter(refridgerator, red)`
Verify	`verify_f`	CLIP	1	True / False	`verify_f(beach)`
	`verify`	CLIP	2	True / False	`verify(umbrella, black)`
	`verify_rel`, `verify_rel_inv`	SSP Query Mask	3	True / False	`verify_rel(fry, on, tray)`
Choose	`choose_v`, `choose_h`	Position	1	Answer	`choose_v(car, top, bottom)`
	`choose_f`	CLIP	2	Answer	`choose_f(indoors, outdoors)`
	`choose_subj`	CLIP	3	Answer	`choose_subj(boy, man, older)`
	`choose_attr`	CLIP	4	Answer	`choose_attr(car, color, gray, red)`
	`choose_rel_inv`	SSP Query Mask	4	Answer	`choose_rel(boy, car, left, right)`
Query	`query_n`, `query_v`, `query_h`	Position	1	Answer	`query_h(chair)`
	`query_f`	CLIP	1	Answer	`query_f(place)`
	`query`	CLIP	2	Answer	`query(fence, material)`

a large vision-language model, trained on 400 million image-text pairs without labels. The model learns visual representations with natural language supervision by enforcing a joined embedding of texts and images. With this, CLIP is capable of predicting which text has the highest similarity to an image and can be used for zero-shot image classification. Recently, Shtedritski et al. (2023) showed that visual prompt engineering CLIP by drawing a red circle around a part of the image, where objects of interest are located, improves performance. This method enables the use of location information gained from our SSP memory, while still offering CLIP an image with global information to provide context.

We use CLIP to `filter`, `verify`, `choose`, and `query` attributes of objects. In detail, we use an attribute dictionary and generate sentences based on the attribute type and the object of interest, e.g. "The colour of the chair is red". We feed these sentences to CLIP with the corresponding image, where a red circle marks the object of interest. CLIP then returns the likelihood for each image sentence pair. By selecting the sentence with the highest similarity, we find the most likely attribute. Attribute types include colour and shape, but also more difficult types such as weather, materials, or age.

Experimental Design

We evaluate the overall performance of our SSP representation of natural images and the corresponding reasoning capabilities by computing the zero-shot accuracy on the GQA dataset (Hudson & Manning, 2019). VQA is a difficult task that requires object detection, scene understanding, and spatial relation understanding. Zero-shot further defines that no training on the specific data set occurs (Cao & Jiang, 2023), ensuring the generalisability of the methods used. We have selected the GQA dataset for evaluation, as it specifically focuses on spatial reasoning (Banerjee et al., 2021). We evaluate our method on the GQA validation set, which consists of 132,062 questions, paired with 10,234 unique images. Around 54% of questions in the validation set include a spatial relation query. We hypothesise that the fractional binding method of SSPs excels at such spatial queries and specifically analyse this by evaluating the performance of our method on the different types of questions. Further, we analyse the types of errors our reasoning pipeline exhibits.

Scaling VSAs to VQA on natural images requires extensions to previous SSP methods (Komer et al., 2019; Lu et al., 2019). For instance, we extended the 2D SSPs to four dimensions. As a result, the capacity of our SSP memory needs to be higher than in the 2D case, where 512 dimensions were found to perform best, while being in line with human capacity limits of working memory (Lu et al., 2019). We analyse the decoding capacity of our 4D SSP memory for dimensionalities: 512, 1,024, and 2,048. Specifically, we compute the mean squared error (MSE) of recalled 2D locations, the intersection-over-union (IoU) of the decoded objects bounding boxes, and the percentage of correctly recalled objects in one image, where correct is defined as $IoU > 0.5$. IoU is calculated as the overlapped area divided by the area of union of two bounding boxes: $\text{IoU} = \frac{box_1 \cap box_2}{box_1 \cup box_2}$ and is a common metric that evaluates the accuracy of bounding boxes.

Another novelty of our method is the generated query masks for spatial relation queries in natural language. We evaluate our 37 generated query masks qualitatively by comparing them to the common understanding of the spatial relation. For example, the relation *to the right of*, which is depicted in the query mask in Figure 1, shows the region of interest to be on the right side of the queried object that is set in the centre of the mask.

Results

Following (Cao & Jiang, 2023; Song, Dong, Zhang, Liu, & Wei, 2022), we calculate the overall accuracy of our model *VSA4VQA* on the GQA validation set with three different random seeds and achieve 46.5% (std 0.09%). See Table 2 for

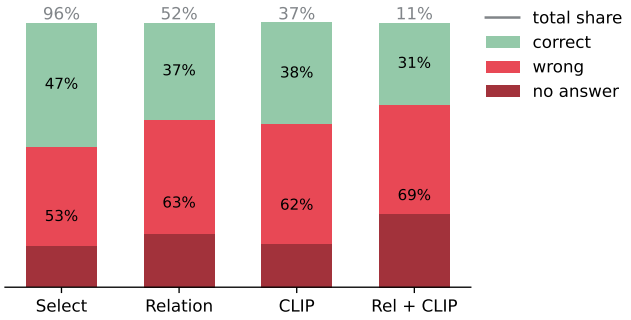

Figure 4: Error analysis by type of question. Percentage on top indicates total share of questions that include the respective query in their program. Wrong answers are further split to indicate when no answer was given.

comparison to other zero-shot methods. The current state-of-the-art deep learning method Mod-Zero-VQA (Cao & Jiang, 2023) also employs the NMN paradigm (Andreas et al., 2016) and uses three different pre-trained deep learning models to implement the functions for answering the questions. However, their method does not encode a cognitively plausible image representation and requires 621 million model parameters, i.e.significantly more than the 150 million used by CLIP.

In Figure 4, we show a more detailed performance analysis split across the different modules. Questions that use a relation function, implemented via query masks on SSP memory, in comparison to attribute functions that are implemented with CLIP. While there are more questions with relational queries (52%), accuracy of CLIP and relations implemented via SSP memory are almost the same, i.e. 37% in comparison to 38%. However, it is important to note here that the percentages given are based on the overall correct answer to the question, so they involve additional steps, where errors could occur. For example, 96% of questions include a `select` function and while the SSP unbinding operation correctly decodes 87% of items, the accuracy of questions including the select operation is also only 47%. The number of questions, where both relational queries and attribute queries were combined are shown in the right bar, where accuracy further decreased as more steps are needed.

A possible reason for the wrong answers in relational queries is the size of the query masks, where it was suggested (Lu et al., 2019) that a higher region size will reduce decoding accuracy. We have tested this hypothesis by calculating the pearson correlation coefficient between accuracy and query mask size, resulting in a weak correlation of -0.30 (one-sided $p = 0.09$). Another possible source of errors is the inherent stochasticity in the SPA. Each SP is chosen randomly, therefore, the vectors are sometimes more or less suitable for the question. We have balanced this effect by running multiple random seeds, however, running the same question with multiple random seeds and choosing the majority answer might further improve performance. This was computationally infeasible in our case, due to the large dataset size, but should be tested in future work on other datasets.

Table 2: Overall accuracy of zero-shot models on GQA validation set. Our method VSA4VQA is comparable to state-of-the-art deep learning method Mod-Zero-VQA.

Method	**Accuracy**
TAP-C (Song et al., 2022)	36.3%
Mod-Zero-VQA (Cao & Jiang, 2023)	47.3%
VSA4VQA (ours)	46.5%

We have further highlighted the amount of questions with no answer in Figure 4. These occur if a `select` operation did not find the requested object or if a `filter` operation returns the wrong result and incorrectly terminates the program. Other causes also include functions that are not implemented due to nesting, e.g. finding objects that are the same or different. In relation queries this amounts to 20% and in CLIP queries to 16%. We highlight these questions as open directions, where further improvements are possible.

Capacity Analysis

To analyse the impact of model capacity on performance, reflected in the dimensionality of the SSP vectors, we evaluated on a reduced set of 10,000 questions to save computation time, as done in previous work (Komer et al., 2019). Performance results for SSP vector dimensions of size 512, 1,024, and 2,048 are summarised in Table 3. 512 dimensions were chosen in previous work as it achieved high decoding accuracy for up to 10000 random SSPs (Komer et al., 2019). As can be seen from the table, here, 512 dimensions result in a high MSE and low IoU between bounding boxes, which is most likely due to the increase in dimensions of the vector space from 2D to 4D. At the same time, however, the choice of dimensions does not seem to significantly impact the accuracy: Using 512-dimensional vectors results in similar performance as for higher dimensions. This suggests that SSP decoding accuracy only plays a minor role in overall performance of *VSA4VQA* on question answering. Further, 2,048 dimensions do not significantly increase accuracy, which indicates that 1,024 dimensions are sufficient, while requiring less computation time and resources.

Table 3: Capacity analysis. Decoding accuracy of items with different dimension of SSP vectors, measured with mean-squared-error (MSE), intersection-over-union (IoU), percentage of correct items, and VQA accuracy.

Dimensions	**MSE** ↓	**IoU** ↑	**Items** ↑	**Accuracy** ↑
512	51.46	0.59	63.47%	45.03%
1,024	26.23	0.80	86.71%	**46.02%**
2,048	**23.09**	**0.84**	**89.78%**	45.96%

Error Analysis

We further analyse the types of errors that can occur on a few selected questions. We found four distinctive types of errors: (1) wrong SSP encoding/decoding, (2) wrong CLIP prediction, (3) bad programs, and (4) ambiguous questions. In the first case, the `select` method returns the wrong location for the requested object. As we have seen in the capacity analysis, this might be due to the vector's dimension and subsequent noise. When there are many objects, the orthogonality principle of the hyper-dimensional vector space no longer holds and object SSPs are no longer orthogonal in the vector space, which in turn results in overlapping SSPs that cannot be disentangled correctly in the clean-up process.

In the second case, wrong CLIP predictions can lead to a false answer or no answer at all. CLIP is sensible to the selection of proposal sentences. We have tested different options: only giving the attribute, using the object name plus the attribute, and building a full sentence. We found that the full sentence, where possible, gives the best result On the other hand, CLIP is also sensible to the processing of the image. We have selected the red circle method proposed by Shtedritski et al. (2023), as it showed improved performance.

The third type of error is due to incorrect programs. Here, we found two main causes: terms with more than one word and incorrect split of `filter_v` and `filter_h`. While the latter could be remedied in our implementation, the former is difficult to address. For example, the term "soccer ball" gets incorrectly split so that the program looks like `select(soccer)`, which naturally does not yield a result when querying the SSP memory, where only `soccer ball` was encoded. A possible solution would be to have a dictionary with all terms that encompass more than two words and recreate the programs with a respective check of the dictionary. This will be left for future work.

The final type of error stems from ambiguous questions in the GQA dataset. For example, one question is "who is wearing a shirt?", but in the corresponding image both a girl and a boy are wearing shirts, while the correct answer is "girl." This example illustrates that questions can be ambiguous, i.e. even humans might answer them incorrectly. In fact, human subjects tested on 4000 random questions from GQA only achieved an average accuracy of 89.3% (Hudson & Manning, 2019). Approaches on other datasets have addressed this issue by using soft VQA scores that account for multiple correct answers, for details see (Cao & Jiang, 2023; Song et al., 2022). GQA, however, does not provide the required annotations.

Discussion

In our detailed error analysis we found four causes for errors: wrong SSP decoding, incorrect CLIP predictions, bad programs, and ambiguous questions. While the latter two are due to the dataset selection and might not be as pronounced on different datasets, the first two could be improved upon. Specifically, the incorrect CLIP predictions might be circumvented by extracting the attributes such as colour or shape with more specialised feature extractors and encoding them into the SSP memory. However, such a feature extractor model needs to be trained and as we have seen in our capacity analysis, the additional encoding of attributes in SSPs would require higher vector dimensions.

We have chosen to use 1,024 dimensional SSP vectors to stay comparable to previous work and within human capacity limits (Komer et al., 2019). For future work, however, it would make sense to distinguish between a real memory task, where all objects need to be retained in memory over time, and the VQA task, where the image stays available. The former setting is subject to biological capacity limits and should therefore be implemented with limitation on vector size. VQA, on the other hand, could allow for higher vector dimensions without loss of biological realism.

A current limitation of our method is the use of ground truth object annotations. In future work, we want to address this by using pre-trained object detectors. While GQA provides detections of a fine-tuned Faster-RCNN (Ren, He, Girshick, & Sun, 2015), they do not include the labels of bounding boxes, which are critical for our approach and could not be reproduced. Furthermore, handling multiple objects of the same type is difficult in the current method. In previous work (Komer et al., 2019) this was solved by computing an average across multiple random seeds.

Similarly, we found that encoding objects as regions did not work in our setting. This is, again, likely due to the limitation of random seeds – this approach would allow for decoding the full region with higher precision (Lu et al., 2019) but is not possible for the computationally more demanding VQA task on natural images. Instead, we chose to extend the VSA to four dimensions to encode objects' width and height. In future work, we are interested in finding a more cognitively inspired method. In general, *VSA4VQA* could be improved by moving towards a fully neural implementation on dedicated hardware, which would improve efficiency and, therefore, allow for probabilistic inference with multiple random seeds.

Conclusion

Our proposed model, *VSA4VQA*, is capable of answering complex compositional questions on natural images. We have tested our model on a dataset that focuses on spatial queries and achieved comparable performance to current zero-shot deep learning approaches. To the best of our knowledge, *VSA4VQA* is the first model to implement a cognitively plausible image representation of natural images, which can be used to answer complex spatial queries. We have effectively scaled a VSA to encode additional dimensions for width and height of objects. Further, we generated 37 spatial query masks from data to answer relation-based questions and integrated a pre-trained vision-language model to answer attribute-related questions. Our extensive analysis on questions, errors, and capacity limits provides valuable insights for future work.

Acknowledgments

Anna Penzkofer and Lei Shi were funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2075 – 390740016.

References

Andreas, J., Rohrbach, M., Darrell, T., & Klein, D. (2016). Neural Module Networks..

Banerjee, P., Gokhale, T., Yang, Y., & Baral, C. (2021, October). Weakly Supervised Relative Spatial Reasoning for Visual Question Answering. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV).* Montreal, QC, Canada: IEEE.

Bartlett, M., Stewart, T. C., & Orchard, J. (2022). Biologically-Based Neural Representations Enable Fast Online Shallow Reinforcement Learning. *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Cao, R., & Jiang, J. (2023, July). Modularized Zero-shot VQA with Pre-trained Models. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Findings of the Association for Computational Linguistics: ACL 2023.* Toronto, Canada: Association for Computational Linguistics.

Chen, W., Gan, Z., Li, L., Cheng, Y., Wang, W., & Liu, J. (2020, November). *Meta Module Network for Compositional Visual Reasoning.* arXiv.

Choo, F.-X. (2018). *Spaun 2.0: Extending the World's Largest Functional Brain Model*. Doctoral Thesis, University of Waterloo.

Eliasmith, C. (2013). *How to Build a Brain: A Neural Architecture for Biological Cognition*. Oxford University Press.

Hersche, M., Zeqiri, M., Benini, L., Sebastian, A., & Rahimi, A. (2023, March). *A Neuro-vector-symbolic Architecture for Solving Raven's Progressive Matrices.* arXiv.

Hudson, D. A., & Manning, C. D. (2019, May). *GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering.* arXiv.

Komer, B., Stewart, T., Voelker, A., & Eliasmith, C. (2019, July). A neural representation of continuous space using fractional binding. In *Annual Meeting of the Cognitive Science Society.*

Lu, T., Voelker, A., Komer, B., & Eliasmith, C. (2019, July). Representing spatial relations with fractional binding. In *Annual Meeting of the Cognitive Science Society.*

Plate, T. A. (2003). *Holographic Reduced Representation: Distributed Representation for Cognitive Structures*. Center for the Study of Language and Information.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., … Clark, J. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning.* PMLR.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems.* Curran Associates, Inc.

Shtedritski, A., Rupprecht, C., & Vedaldi, A. (2023, October). What does CLIP know about a red circle? Visual prompt engineering for VLMs. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV).* Paris, France: IEEE.

Song, H., Dong, L., Zhang, W.-N., Liu, T., & Wei, F. (2022, March). *CLIP Models are Few-shot Learners: Empirical Studies on VQA and Visual Entailment.* arXiv.

Stewart, T., Choo, F.-X., & Eliasmith, C. (2012). Spaun: A Perception-Cognition-Action Model Using Spiking Neurons. *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Subramanian, S., Merrill, W., Darrell, T., Gardner, M., Singh, S., & Rohrbach, A. (2022, May). *ReCLIP: A Strong Zero-Shot Baseline for Referring Expression Comprehension.* arXiv.