

Received 7 November 2023, accepted 2 December 2023, date of publication 7 December 2023, date of current version 19 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3341007

RESEARCH ARTICLE

NS-IL: Neuro-Symbolic Visual Question Answering Using Incrementally Learnt, Independent Probabilistic Models for Small Sample Sizes

PENNY JOHNSTON¹, KEILLER NOGUEIRA¹, (Associate Member, IEEE), AND KEVIN SWINGLER¹

Department of Computer Science, University of Stirling, FK9 4LA Stirling, U.K.

Corresponding author: Penny Johnston (Penny.Johnston@stir.ac.uk)

This work was supported by Stirling University.

ABSTRACT This paper is motivated by the challenge of providing accurate and contextually relevant answers to natural language questions about visual scenes, particularly in support of individuals with visual impairments. We present a system that is capable of incrementally learning both visual concepts and symbolic facts to answer natural language questions about visual scenes via rich concepts. Deep neural networks are used to learn a feature space from which visual classes are learned as independent probability distributions, allowing new classes to be added arbitrarily with small sample sizes and without the risk of catastrophic forgetting associated with traditional neural networks. Visual classes are not limited to object labels, but also include visual attributes. A knowledge graph is used to represent facts about objects, such as their actions, locations and the relationships between different objects. This allows facts to be stored explicitly and added incrementally. A large language model is used to translate between natural language questions and knowledge graph traversal queries, providing a natural visual question answering process.

INDEX TERMS Neuro-symbolic system, visual question answering, classification system, Gaussian mixture model, incremental learning.

I. INTRODUCTION

The *Visual Question Answering (VQA)* task requires a system to answer questions in natural language about visual images. To handle real world changes effectively, a VQA System must constantly acquire, update, and make use of knowledge over time. This ability, known as continual learning, forms the basis for AI systems to adapt and improve themselves. This paper presents a Neuro-Symbolic architecture for the VQA task, which is illustrated in Figure 1.

The system integrates neural network based visual representations with a symbolic knowledge graph and a large language model (LLM) that translates between natural language and traversals of the knowledge graph. The main components are:

The associate editor coordinating the review of this manuscript and approving it for publication was Wenming Cao¹.

- 1) An extensible *Classification System* which is comprised of two main components.
 - A pre-trained Deep Neural Network (DNN) that maps pixel images onto a visual feature space that is suitable for performing image classification.
 - An extensible set of independent conditional probability models for individual visual classes such as object labels and object attributes. These models can be learned one-at-a-time from small data sets and added or removed from the system as needed. The models are represented as Gaussian Mixture Models (GMMs).
- 2) A *Knowledge Graph (KG)* that stores symbolic information about Concepts and their relationships. Possible classes of any Concept are stored in the nodes of the knowledge graph and facts about relationships

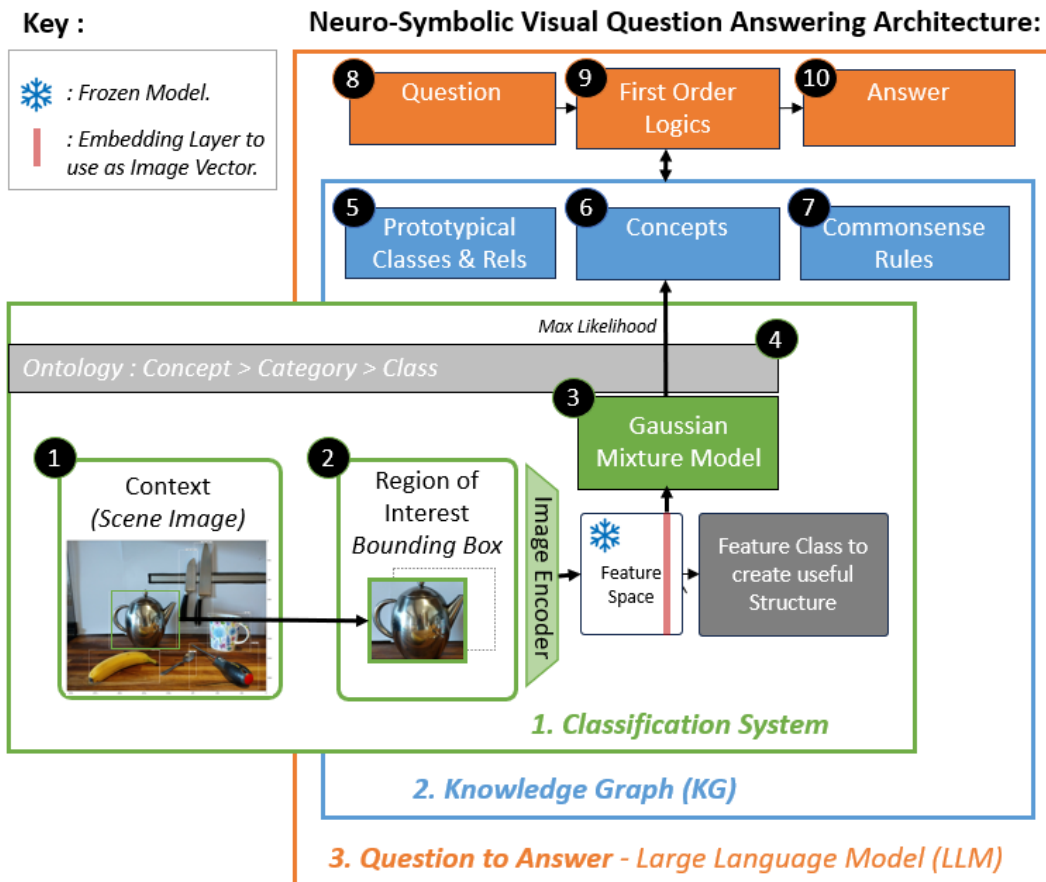


FIGURE 1. Overview: neuro-symbolic visual question answering using incrementally learnt, independent probabilistic models for small sample sizes.

between classes are stored in its edges. The individual probabilistic models for each class are stored in the graph nodes linking the symbolic elements to the neural feature space. Questions are answered by traversing the graph to access the relevant facts.

- 3) A *Question to Answer* mechanism which uses a LLM to translate between natural language, and, the computer-readable graph traversal paths required to answer questions. The results are translated back into natural language using the same LLM.

This architecture has the potential to enhance VQA by effectively managing abstracted meaning through rich concepts created by compositional categorical labels, enabling enhanced grounded reasoning through natural language processing.

The structure of this paper is as follows: We examine the current state of VQA design in Section II, where we review existing architectures. Then, we define the key terms used during the various levels of information abstraction and knowledge representation in Section III. Our novel Architecture is presented through Step descriptions in Section IV. The technical details for the Steps, including model configurations, hyperparameter settings, training procedures, and

implementation specifics are outlined in Section V. The results of the system testing are documented in Section VI. Limitations are described in Section VII. We discuss the system’s performance in Section VIII. Conclusions are given in Section IX and we explore potential avenues for further research in Section X.

A. CONTRIBUTIONS OF THIS PAPER

The contributions that this paper contains are:

- 1) The introduction of a novel Neuro-Symbolic Architecture, abstracting bottom-up information between neural nets and symbolic reasoning via Gaussian Mixture Model probability learning. This has the ability to learn both visual concepts and symbolic facts to answer natural language questions about visual scenes.
- 2) The ability to learn/unlearn new concepts arbitrarily using small sample sizes and without the risk of catastrophic forgetting associated with traditional neural networks.
- 3) Visual Concepts are not limited to object classes but also include other categorical classes such as; visual attributes, object affordances and these can be amended using explicit core knowledge.

- 4) The ability to translate between natural language questions and knowledge graph traversal queries, providing a natural visual question-answering process.

II. RELATED WORK

Deep learning has transformed VQA with end-to-end neural networks using self-supervised vision and language pre-training (VLP). They learn from vast image-text data [1], but struggle with questions needing logical reasoning. CLIP [2] for “Contrastive Language-Image Pre-training”, simultaneously teaches an image encoder and a text encoder to forecast the accurate matches within a set of image and text training examples. During testing, the trained text encoder creates a linear classifier without prior training, by embedding the names or descriptions of the target dataset’s classes. The CLIP model is an example of a foundation model [3].

The concept of visual attention in VQA [4], has become a staple in VQA methods. Studies indicate that learning co-attention for both visual and textual inputs refines the representation of images and questions, improving predictions. This attention system allows the model to focus on key regions and words, merging visual and textual data more effectively [5], [6], [7], [8], [9].

Neuro-symbolic methods [10] separate reasoning from vision and language understanding. They merge structured image representation with symbolic programs from sentences to reason and gain answers, integrating deep learning for visual and language recognition with symbolic execution for reasoning [10], [11]. A similar idea to ours is the Neuro-Symbolic Concept Learner (NS-CL) [12] it breaks down the question inputs into programs and carries out these programs on object features to obtain answers. Additionally, Vedantam et al. [13] introduce a novel category of probabilistic neural-symbolic models that incorporate symbolic functional programs as a hidden, random variable for solving visual question answering tasks. State of the art models rapidly adapt to new VQA tasks with limited training data [14], [15]. These models effectively utilise external context, bridging vision and language pre-trained models, handling multimodal inputs, and achieving zero-shot learning. Other models explore prompt-based learning [16] to reduce reliance on VQA training data while maintaining accuracy, thus showing effective use of both image and textual context.

Multimodal fusion techniques are explored to better integrate visual and textual information [17]. Using pre-trained language models (PLMs) for zero-shot VQA has shown promising results. There are two main approaches to incorporating visual information into PLMs. One approach involves additional training on image-text data to enhance vision language capabilities. Alternatively, the large PLM can be kept frozen while training the vision encoder separately to preserve its knowledge in question answering, as suggested by Menick et al. [18]. In this method, the output of the vision encoder is combined with text prompts and used for

fine-tuning the PLM through language modeling objectives such as prefix and masked language modeling. VLKD [19] distills multimodal knowledge into PLMs by using CLIP [2] as the teacher model during finetuning. Flamingo [14] introduces additional layers to both the pre-trained vision model and the PLM and trains these new layers. Karpas et al. [20] recognise the limitations of language model-based reasoning and points to recent proposals that have emerged to bridge language models with external symbolic plug-ins or toolkits.

A trend is emerging towards integrating explicit knowledge sources like Knowledge Graphs (KGs) with parametric knowledge derived from LLMs through reasoning prompt methodologies. However, a definitive solution to the persistent challenge of hallucinations remains unresolved, leading to an ongoing need for embedding meaning within a framework such as ours. A comprehensive understanding of the combined capabilities of Knowledge Graphs and Large Language Models is discussed in-depth by Pan et al. [21]. This shift signifies Knowledge Computing which expands reasoning across various knowledge formats. KGs are being merged with Transformer-based LLMs such as BERT [22], RoBERTa [23], GPT series [24], and LLaMA [25]. Some research augments KGs with LLMs for tasks like knowledge extraction, while others leverage KGs to enhance LLMs for training or knowledge augmentation.

KGs provide a structured representation of knowledge, facilitating reasoning and inference. Critics such as Bender et al. [26] believe the parametric knowledge in LLM uses statistical patterns rather than genuine understanding. Advocates, such as those behind ChatGPT, highlight its ability to generalise from vast datasets, demonstrating extensive information capture and impressive language comprehension. However, LLMs might produce plausible but incorrect responses, due to the absence of explicit knowledge representation [27]. Their capability to discern directional entailments remains under scrutiny [28]. While training LLMs is costly and they offer immediate utility for various applications, building KGs can be resource-intensive but are trustworthy.

III. TERMINOLOGY

This paper uses the following terminology which can be seen at different levels within the structure in Figure 2.

- **Feature Classes:** These are the classes used to train the image encoder model and are purely used to gain a useful structure in the feature space.
- **Feature Space:** Generated by deep learning algorithms designed to produce a structured feature space suitable for encoding images in the form of a Feature Vector which can then be used during training/inference in GMM membership.
- **Feature Vector:** The Feature Vector is created through image encoding and the selection of an embedding layer from the neural net.
- **Categorical Class:** A Categorical class is a Class that belongs to a specific category.

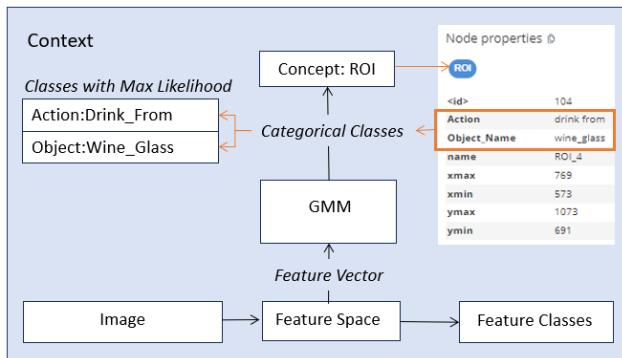


FIGURE 2. Terminology used in the NS-IL System.

- **Category:** Categories are used to group like classes together.
- **Concept:** Describes a ‘Region of Interest’ (ROI) which is the contents of a bounding box, See Figure 2 that shows the node created from the ROI assigned 2 categories; Object & Affordance, the Max likelihood class per category is assigned.
- **Context:** The environment in which data is collected, including images of scenes to be interpreted and natural language questions.

IV. METHOD

In the introduction we outlined the Visual Question Answering Task and listed our three main components. In Figure 1 the three components are identified by the blue, orange and green areas within the complete framework. Each area is now expanded:

- 1) Classification System (*Green*): In Section IV-A, Steps 1, 2, 3, and 4 describe a system that can infer categorical classes from a set of Regions Of Interest (ROI) in an image.
- 2) Knowledge Graph (*Blue*): In Section IV-B, Step 5 describes the stored prototypical classes and relationships for each class. Step 6 infers Concepts from identified classes and Step 7 adjusts those classes applying core knowledge rules.
- 3) Question to Answer (*Orange*): In Section IV-C, translating natural language questions into machine executable queries is explained in Steps 8 and 9. Step 10 translates the returned results back into natural language answers.

Each component and its associated Steps will be covered in further detail in the next sections. Section V shows the individual model selections for the Steps, together with their setups and configurations.

A. CLASSIFICATION SYSTEM

In Figure 1, each step of the Classification system describes a specific task. Step 1 identifies regions of interest (ROI) in the given image, Step 2 encodes each ROI into a Feature Vector

given in terms of a feature space. Step 3 identifies the most likely class for each category used and Step 4 associates that most likely class with a node in the knowledge graph.

1) STEP 1: ROI IDENTIFICATION

The scene image is what represents truth and enables us to ground the inferred *Categorical Classes* with visual reality within the VQA task. The image contains our contextual information which can be a single class or alternatively, the image could hold classes from different categories, such as a scene featuring an object which has attributes. The scene is split into a number of ROIs. Section V-B1 details how these are detected resulting in a list of suggested bounding boxes. Each bounding box marks a region for which category:class(es) will be inferred. Additional contextual information is also retained for later use. The timestamp when the bounding box entry was created (useful for tracking objects in temporal activities if more than one image is shown); the position of the bounding box within the context image (helpful for generalisations over known object proximity’s to enhance inference activities); and the file location of the scene image, providing the source for all the bounding box regions in the image. These additional details are used in Step 6 when the Concept is generated to optimise inference capability later.

2) STEP 2: REGIONS OF INTEREST & ASSOCIATED FEATURE SPACE

Each ROI bounding box is handled separately and selected from the suggested list. The procedure involves encoding the pixel contents of the bounding box for Step 3. Transforming the ROI image into a feature vector in the feature space offers benefits such as data compression, feature extraction, and noise reduction. The specific feature space created by our image encoder is detailed in Section V-B2, and additional discussion on feature spaces is in the main discussion found in Section VIII.

3) STEP 3: CLASSIFICATION

A Gaussian Mixture Model is used for independent classification. This approach was taken since features can have complex, multi-modal distributions rather than simple, single-mode distributions. This architectural choice mitigates three major weaknesses of neural nets: their susceptibility to ‘catastrophic forgetting’, the extended retraining duration required, and the substantial sample sizes needed to train a new class. With independent GMMs, our architecture permits the flexible addition or removal of classes, even with limited sample size.

4) STEP 4: ASSOCIATION

The feature vector of the ROI image is fed into each of our pre-trained GMMs to generate a class specific likelihood. This likelihood signifies the class conditional probability given the feature vector. Details concerning the training and

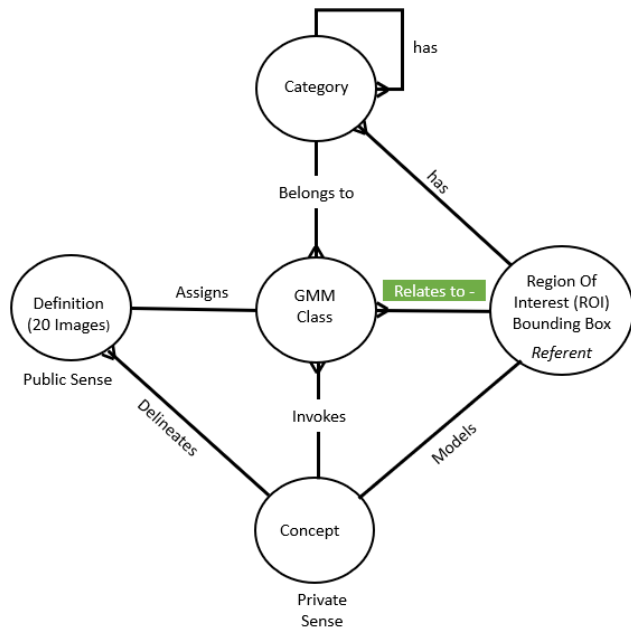


FIGURE 3. The schema that underpins the knowledge graph.

selection of the GMM classifiers are in Section V-B3. These class likelihoods are used in Step 6.

B. KNOWLEDGE GRAPH

Figure 3 shows a Schema that underpins NS-IL’s knowledge graph. In order to create rich *Concepts* that are more intricate and evolving than pure fixed *Classes*. A *Concept* that models an ROI can be modeled using many *Category:Classes* (Terms), each Term belongs to a different category (classifier). For example, for an ROI that contains a kettle, the visual information would return *Object:kettle, Attribute:Shiny* all of these categories of information taken together build a rich definition for the ‘kettle’ *Concept*. This kettle concept can then be linked to further Prototypical relationships that are held in the KG such as as; afforded an *Action* like ‘pour water’ and be located in a particular *Location* such as ‘kitchen’. Helping individuals to understand how the *Concept* can be used in reality. Each GMM Term/Class is learnt using 20 images.

1) STEP 5: PROTOTYPICAL CLASSES & RELATIONSHIPS

In Step 5 the knowledge graph is initialised with prototypical classes and relationships. These represent the classes and relationships a human has learnt during their interaction with the environment and, whilst they may be fine-tuned over time, generally they are stable and not subject to change.

2) STEP 6: CLASS INFORMATION

The use of GMMs enables categorical classifiers by associating specific GMM classes with categories. This process allows us to calculate for a given classifier, the most probable ‘seen’ class. In Step 6, when the system identifies an ROI in

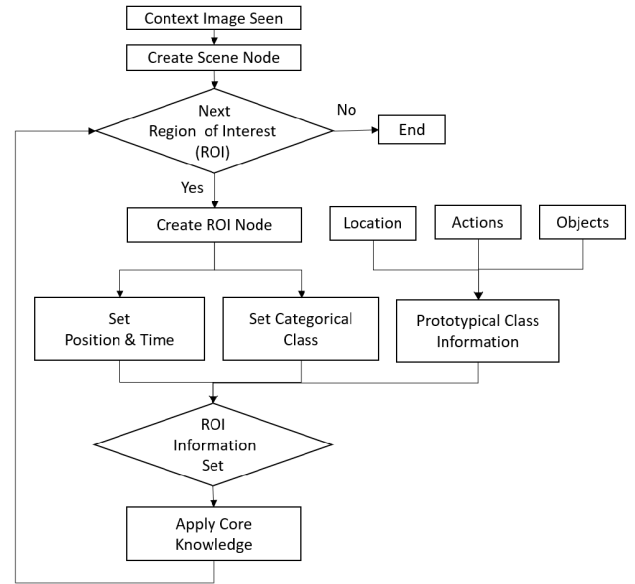


FIGURE 4. Knowledge graph: concept creation.

the given image, an individual node is created for each ROI listed. The process flow for node and edge generation in the KG is shown in Figure 4. For each generated ROI (*Concept*) node the ROI’s coordinates are saved and the timestamp of when it was created. The *Object*’s classes are identified, and the relevant prototypical information is associated with the object class, such as its usual room location and associated actions. Additionally, any observed attribute classes are also listed against that ROI.

3) STEP 7: CORE KNOWLEDGE RULES

After the ROI class information has been synthesised into a *Concept* in the knowledge graph, core knowledge rules can be applied to the contents. For instance, consider a scenario where you’re unfamiliar with a zebra but someone teaches you that a zebra resembles a striped horse. In such a case, upon encountering an ROI with an object class of ‘horse’ together with an attribute class ‘striped’, you could amend the object label to ‘Zebra’ without ever seeing an actual zebra in reality.

C. QUESTION TO ANSWER

The contents of the given image have been transformed into structured semantic data within the KG. We now shift our attention from the vision domain to the language domain.

1) STEP 8: QUESTION TO CODE

Using the given question a prompt is engineered for input to a Large Language Model (LLM). A LLM is used since they can consider the context in their responses. Given a well-phrased context or preamble, they can produce answers that are relevant to the specific situation described, making interactions feel more intuitive for the user. The purpose

of the engineered prompt is to focus the LLMs attention. The prompt is provided with a schema context and given examples of the programming language syntax into which natural language questions must be translated. This process is explained in more detail in Section V-D1. The engineered prompt is sent to the LLM which returns a response in a machine readable graph query statement using terms held in the KG.

2) STEP 9: EXECUTE CODE

The machine query statement is taken and executed on the KG which returns results.

3) STEP 10: RESULTS TO ANSWER

The results received from the KG are included in a new engineered prompt which also incorporates the previous prompt and the initial question. These elements are sent to the LLM which returns a contextualised Answer that the user can understand.

V. EXPERIMENT SETUP AND CONFIGURATION

This section describes the selection, configuration, and training processes for each model employed throughout the different steps. To provide clarity on the boundaries of the steps, the step inputs and outputs are documented in each subsection, which also underscores the responsibilities of each model.

A. HARDWARE AND SOFTWARE

All deep learning-based models were implemented using TensorFlow [29] Version 2.13.0. The code was written in Python Version 3.9.2, Neo4j Version 5.9.0, and CUDA Version 11.8. All experiments conducted here were performed on a 64GB Intel i9-9900K workstation with 16 CPU cores at 3.6GHz and an NVIDIA GeForce RTX 4090 with 24.5GB RAM using NVIDIA driver version 535.54.03. Debian version 11.7 was used as the operating system.

B. CLASSIFICATION SYSTEM

1) STEP 1: REGIONS OF INTEREST IDENTIFICATION

Input: Scene Image.

Output: A list of suggested ROI (bounding boxes) in the image.

ROIs are identified using a pre-trained Inception-ResNet-v2 [30], a Faster R-CNN with ResNet V2. The object detector was trained on the ‘Open Images V7’ dataset. Open Images is a dataset of approximately ~9M images annotated with image-level classes, object bounding boxes, object segmentation masks, visual relationships, and localised narratives. The Object Detector generates up to 40 suggested bounding boxes, this is a hyperparameter set to reduce suggested ROIs. Which are subsequently filtered based on a set of class probability threshold criteria.

TABLE 1. GMM class models: configuration of best models.

Class	Reg CoVar	Comp. Num	Comp. Shape	BIC
0: horse	$1e^2$	3	diag	312764.94971072173
1: plate	$1e^2$	3	diag	32048.9749858462
2: teaspoon	$1e^2$	3	diag	84015.02517372137
3: knife	$1e^2$	3	diag	369096.505543082
4: can_opener	$1e^2$	3	diag	437297.0493718943
5: banana	$1e^2$	3	diag	468663.6017532455
6: teapot	$1e^2$	3	diag	211324.83007161587
7: wine_glass	$1e^2$	3	diag	343124.10318505554
8: wooden_spoon	$1e^2$	3	diag	39630.36155217671
9: frying_pan	$1e^2$	3	diag	34081.78180753725
10: kettle	$1e^2$	3	diag	489804.4885691839
11: cup	$1e^2$	3	diag	223413.239309614
12: washbasin	$1e^2$	3	diag	547889.2854903314
13: wine_bottle	$1e^2$	3	diag	361427.44600043306
14: corkscrew	$1e^2$	3	diag	383366.8048321256
15: screwdriver	$1e^2$	3	diag	286082.7113044384
16: striped	$1e^2$	3	diag	415945.5843595127

2) STEP 2: FEATURE SPACE ENCODING

Input: List of ROI and associated Bounding Box coordinates for image.

Output: Encoded ROI image(s).

An Image Encoder was used called VGG19 [31], utilising Imagenet weights, excluding the top layer. This process creates a 25088 visual feature vector for each image and understands 1000 classes.

3) STEP 3: CLASSIFICATION

Input: Encoded ROI Image(s).

Output: A list of classes and likelihoods per classifier. The max likelihood class is identified per classifier.

This is the stage where information is abstracted from a neural net to a symbolic representation. GMMs were trained according to the classes specified in Table 1. Each GMM model was trained on 20 representative images, Figure 5 shows the 20 images and how a GMM is fit to their distribution for the banana class. The training images were obtained by using the class name as a search term in Google, 20 of the returned images were selected and subsequently centrally cropped and encoded.

In order to optimise the likelihood landscape across GMMs, the training method diverges from that described in Johnston et al. [32]. The hyperparameters were adjusted to be set as follows: i) Number of mixture components: 3, ii) Covariance type: Diagonal, and iii) non-negative regularisation: $1.0e^{-2}$. The GMM model then underwent training for 10 epochs. The model that achieved the lowest BIC score was retained for subsequent use. Figure 6 illustrates the range of BIC scores achieved per GMM Model when trained, Table 1 shows an example list of the retained minimum BIC scores after 10 trials per GMM class. In subsequent testing, GMMs are tested on a test dataset consisting of five images per class. The ‘Objects’ classifier, reported an accuracy of 92.31%, an F1 score of 92.04%, and a Cohen’s Kappa score of 93.63%.

4) STEP 4: ASSOCIATION.

Each class was associated with a category to enable a maximum likelihood across all the GMMs to be found for that category.

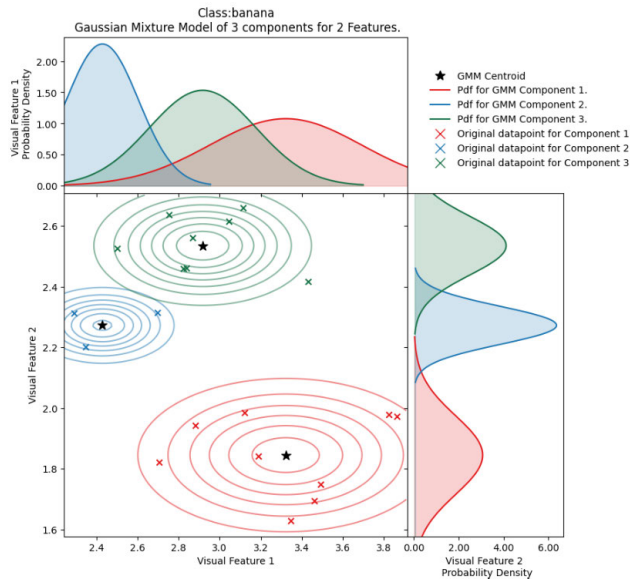


FIGURE 5. Class data fitted to a gaussian mixture model. class shown: banana.

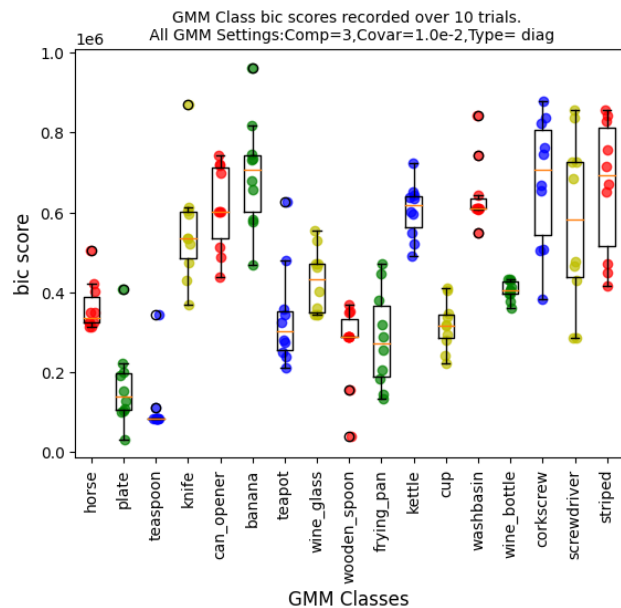


FIGURE 6. GMM class models: Distribution of bayesian information criterion score.

C. KNOWLEDGE GRAPH

The Knowledge Graph is implemented in Neo4J, a high performance NoSQL database designed for handling nodes and relationships [33]. The Cypher language is used to operate and manage this graph database. Py2neo (a client library and toolkit) is used to interface with Neo4j from within our Python application. This allows the execution of Cypher code via a Python wrapper.

1) STEP 5: PROTOTYPICAL CLASSES & RELATIONSHIPS

This step describes the Classes and Relationships added to the Knowledge Graph. The system classes are grouped into

categories such as; Objects, Attributes, Actions and Locations which contribute to forming a richer Concept. The Object and Attribute Classes are grounded via an associated GMM which is trained on 20 images that relate the term to the visual world. Relationships are also added between classes such as, a banana affords ‘eating’, a cup affords ‘drinking’, and a knife affords ‘cutting’.

2) STEP 6: CLASSIFICATION INFORMATION

When the classes for each classifier have been recognised (Step 4), the class likelihoods are synthesised into the knowledge graph through node generation as shown in Figure 4. The edge between the Concept and each identified Class will hold the likelihood or certainty that that Class was seen.

3) STEP 7: CORE KNOWLEDGE RULES

After the new Concepts are generated in the knowledge graph, Core Knowledge rules are applied to amend concepts and relationships to hold the best world model possible. The Core Knowledge rules that are applied are outlined as follows:

- Relationship: Spatial Constructs - Given the lack of depth information, certain assumptions are made about the bounding boxes in the 2D image. Although these simplifications might lead to inaccurate inferences about an Objects relative size and position, they are instrumental in demonstrating possibilities for our proof of concept VQA. Noting that the Y axis decreases in value toward the top left corner of the image, the inferences that are made and the relationships enforced are:
 - LEFT: A relationship is established between a source object and a target object when the x_{max} of the source object is less than the x_{min} of the target object.
 - RIGHT: A relationship exists between a source object and a target object when the x_{min} of the source object exceeds the x_{max} of the target object.
 - INFRONT: A relationship is defined between a source object and a target object if the y_{min} of the source object is greater than the y_{max} of the target object.
 - BEHIND: A relationship forms between a source object and a target object when the y_{max} of the source object is less than the y_{min} of the target object.
- Inference Rule: For a concept comprised of a category Object labeled as ‘horse’ with a category Attribute labeled as ‘Striped’ amend the Object label to ‘Zebra’.

D. QUESTION TO ANSWER

The LLM is utilised through OpenAI’s api of ‘gpt-4’. The KG node and edge semantics are aligned based on suggestions from the LLM, gained by prompting it with our schema and asking it, “Suggest improvements

in the naming of nodes and relationships in the given schema". Cypher queries were generated with ChatGPT-4 on the graph schema through builds on a Neo4j blog by Tomaz Bratanic (<https://neo4j.com/developer-blog/generating-cypher-queries-with-chatgpt-4-on-any-graph-schema/>).

1) STEP 8: QUESTION TO CODE

Taking the question, the prompt is engineered by concatenating the three following pieces of text and sending them to the LLM, then receiving a cypher statement response:

- 1) 'Task: As a Natural Language Query Generator of Cypher queries, your task is to generate a Cypher query for a Neo4j graph database based on the provided schema definition.'
- 2) 'Instructions:
 - Use only the given schema information to create the cypher code. Make sure to respect relationship types and directions. Only use node properties, relationship properties, and relationship points in cypher code.
 - If you cannot generate cypher code based on the given schema, explain the reason to the user.
 - Translate Object Name to Object_Name.
 - Remove [None] from Answer.'
- 3) 'Schema & Examples: As seen in Listing 1 & 2 the schema is generated and to induce a meaning function, a set of paired natural language and code examples are created together as seen in Listing 2. The LLM model then imitates these when generating the Cypher statements.

2) STEP 9: EXECUTE CODE

The Cypher prompt is then executed on the KG and a response is received containing a list of results.

3) STEP 10: RESULTS TO ANSWER

The results given in Step 9 are concatenated with a new task and instruction as seen below, then sent to the LLM to receive an Answer.

- 'Task: Using the provided Question and Answer, construct a sentence that communicates this information to the user who posed the Question.'
- 'Instructions: Limit your response to the provided Question and Answer details. Note: Your responses should not include any explanations or apologies.'

The prompt is sent via the api to the LLM and the user receives an Answer.

VI. RESULTS AND ANALYSIS

A. STEP 2 & 3 - FEATURE SPACE & CLASSIFICATION

To visualise and improve understanding, UMAP [34] is applied to compress the encoding dimensionality to two features which can be seen in Figure 7. The figure shows for each Gaussian; the Mean is shown by 'stars' and the

```
node_properties_query = ""
CALL apoc.meta.data()
YIELD label, other, elementType, type, property
WHERE NOT type = "RELATIONSHIP" AND elementType = "node"
WITH label AS nodeLabels, collect(property) AS properties
RETURN {labels: nodeLabels, properties: properties} AS
  output
""

rel_properties_query = ""
CALL apoc.meta.data()
YIELD label, other, elementType, type, property
WHERE NOT type = "RELATIONSHIP" AND elementType = "
  relationship"
WITH label AS nodeLabels, collect(property) AS properties
RETURN {type: nodeLabels, properties: properties} AS
  output
""

rel_query = ""
CALL apoc.meta.data()
YIELD label, other, elementType, type, property
WHERE type = "RELATIONSHIP" AND elementType = "node"
RETURN {source: label, relationship: property, target:
  other} AS output
""
```

LISTING 1. LLM prompt engineering - knowledge graph schema.

```
This is the schema representation of the Neo4j database.
Node properties are the following:
{node_props}
Relationship properties are the following:
{rel_props}
Relationship point from source to target nodes
{rels}

Examples:
# What can I drink from ?
MATCH (q:ROI{Action:`drink from`}) RETURN q.Object_Name
# What can I cut ?
MATCH (q:ROI{Action:`cut`}) RETURN q.Object_Name
# What affords drink from ?
MATCH (q:ROI{Action:`drink from`})return q.Object_Name
# What can I eat ?
MATCH (q:ROI{Action:`eat`})return q.Object_Name
# Where is a cup ?
MATCH (q:ROI{Object_Name:`cup`})-[r]->(o:ROI) return q.
  Object_Name,r.Value,o.Object_Name
# What can be found in the kitchen ?
MATCH (o:RoomPrototype{name:"kitchen"})-[r:FOUND_AT]->(
  p) RETURN p.name
# Where is the 1st Object in relation to the 2nd object
  ?
MATCH (k:ROI{Object_Name:`knife`})-[r]->(b:ROI{
  Object_Name:`banana`}) RETURN k.Object_Name,k.name,
  r.Value, b.Object_Name,b.name
""
```

LISTING 2. LLM prompt engineering - training through examples.

Covariance Contours by 'ellipses'. This highlights where each class will be predicted in the feature space. In Figure 8 the original 260 training images for the 13 object classes are pin-pointed and the respective regions with the highest likelihoods.

B. STEP 5 - ADDITION OF PROTOTYPICAL CLASSES

In our initial test, the classes of the objects (which didn't include wine bottle or wine glass) were inferred for a given image seen in Figure 9. Two new classes are trained, 'wine bottle' and 'wine glass', they are added to the 'Objects' classifier. When the same image is tested again, the results correctly identified both the wine bottle and the wine glass classes. Table 2 compares the outcomes of the first trial (Trial A) and the second, Trial B, after adding the new classes.

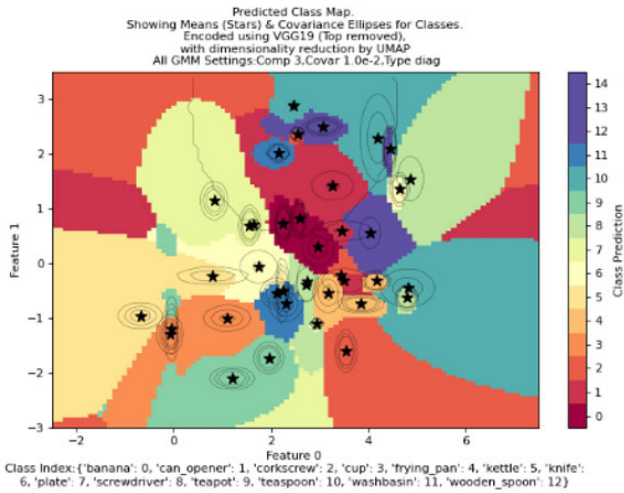


FIGURE 7. GMM class prediction landscape.

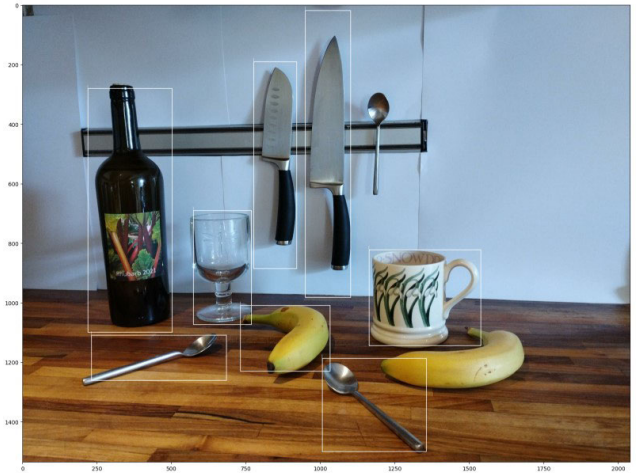


FIGURE 9. Image B: Trial A (without wine glass & wine bottle classes) & Trial B (wine glass and wine bottle classes are added).

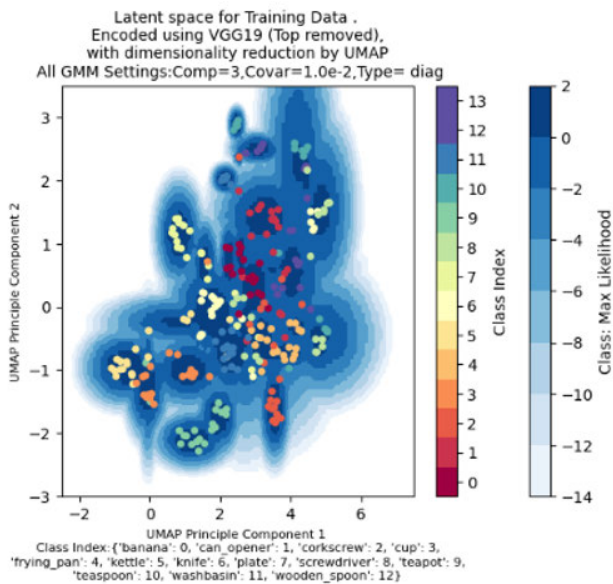


FIGURE 8. Training data overlaid on to the combined max likelihood map.

TABLE 2. Region of Interest (ROI) classification for Image B: Trial A (without wine glass & wine bottle classes) & Trial B (wine glass and wine bottle classes are added).

ROI Name	Trial A	Trial B	Addition Successful
ROI_0	banana	banana	Same
ROI_1	knife	wine_bottle	Wine_bottle classified
ROI_2	teaspoon	teaspoon	Same
ROI_3	cup	cup	Same
ROI_4	knife	wine_glass	Wine_glass classified
ROI_5	knife	knife	Same
ROI_6	screwdriver	screwdriver	Same
ROI_7	knife	knife	Same

In Figure 10, only the likelihoods for the wine bottle and wine glass have changed, confirming the independence and flexibility of our class training approach.

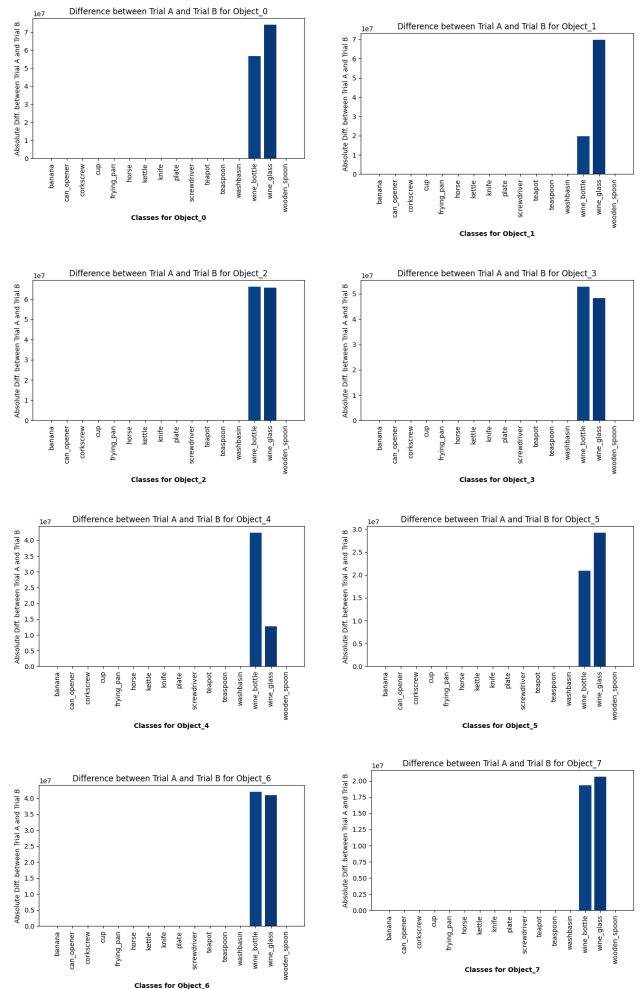


FIGURE 10. Delta likelihood values for all object classes for Trial A & Trial B for Image B. Trial A (without wine glass & wine bottle classes) & Trial B (wine glass and wine bottle classes are added).

C. STEP 6 - SYNTHESING CLASS INFORMATION

The vision component was tested using the GMMs from Table 1. To validate the perceived encoding of the image,

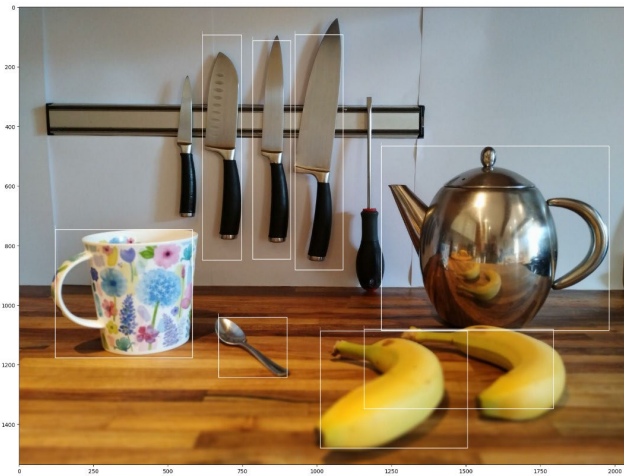


FIGURE 11. Image A to be classified.

TABLE 3. ROI object classification for Image A:

Name	Object	Affordance	xmin	xmax	ymin	ymax
ROI_0	teapot	None	1216	1981	467	1085
ROI_1	teaspoon	None	669	900	1042	1243
ROI_2	cup	drink from	121	582	747	1177
ROI_3	banana	eat	1011	1506	1088	1480
ROI_4	knife	cut	615	747	94	849
ROI_5	banana	eat	1158	1794	1082	1348
ROI_6	knife	cut	926	1088	93	882
ROI_7	knife	cut	784	912	113	846

TABLE 4. Object classification for Image F.

Name	Object	Attr.	Afford.	xmin	xmax	ymin	ymax
ROI_0	zebra	striped	None	232	649	76	366
ROI_1	horse	None	None	34	447	66	324

the KG was queried. Figure 11 shows the input image and Table 3 details the object classes with related affordances for ROIs, with bounding box positions. Figure 12 shows for each ROI the likelihood for each class according to its GMM. The results accurately pinpointed all details. However, the distribution of likelihood values reveals that some classes were more readily identifiable than others. The smaller the delta between 1st and 2nd class, the more similar the classes are in the feature space.

D. STEP 7 - ADDITION OF COMMONSENSE RULES

Two new classes were trained; an object class for ‘horse’ and an attribute class for ‘striped’ using 20 random images for each and establishing a threshold for class ‘striped’ within the Attribute Classifier. A rule was implemented: If an Object is classified as ‘horse’ and also classified with an attribute class ‘striped’, the object name should be renamed ‘Zebra’. This approach allows for the creation of new classes by combining existing Object classes and existing Attribute classes. Although initial results identified two horses as shown in Figure 14, one bounding box with the ‘striped’ attribute was relabelled as ‘Zebra’ shown in Table 4. Thus, we’ve conceptualised a ‘Zebra’ without ever training the system on actual Zebra visual data.

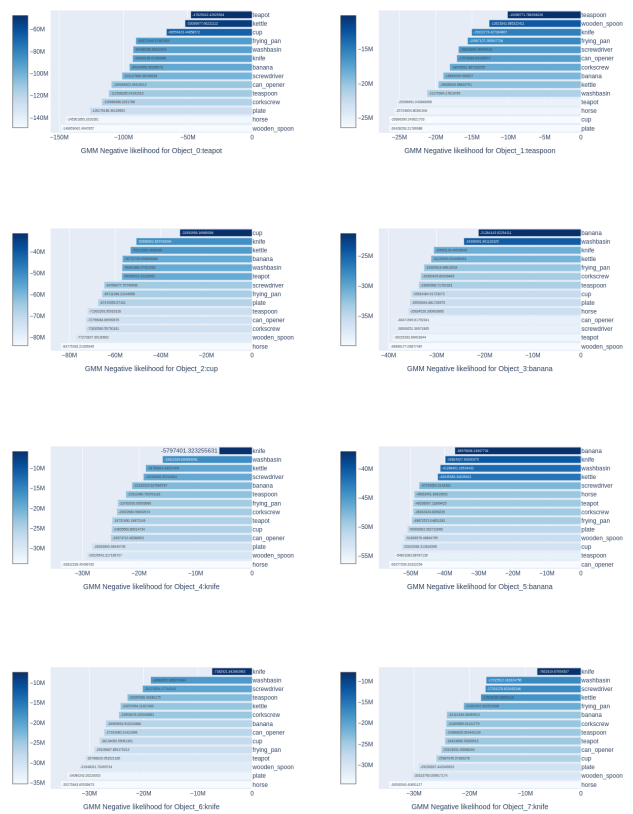


FIGURE 12. Object class likelihoods for Image A, ordered most likely to least.

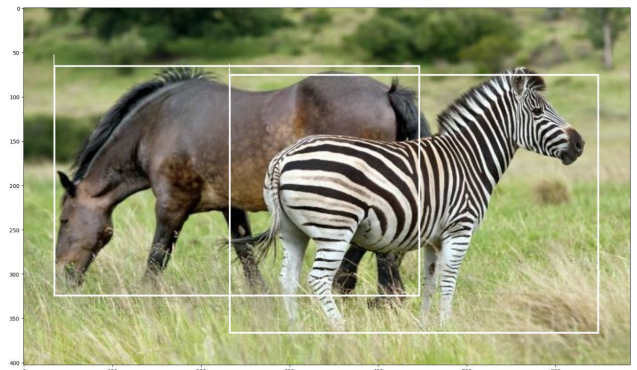


FIGURE 13. Image F: reclassification using commonsense rules.

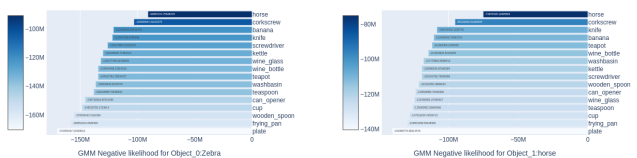


FIGURE 14. Ground truth for Image F.

E. STEPS 8 TO 10

An experiment was conducted to evaluate the VQA’s efficiency using a test set of five context images A to E. These images included diverse objects in different numbers,

Image A			Question / Answer	Cypher / Results
			1. What affordance does knife have?	MATCH (q:ROI[Object_Name:'knife']) RETURN q.Action
			The affordance that a knife has is to cut.	[[q.Action], [cut], [cut], [cut]]
			2. What can I eat?	MATCH (q:ROI[Action:'eat']) RETURN q.Object_Name
			You can eat a banana	[[q.Object_Name], [banana], [banana]]
			3. What can I cut?	MATCH (q:ROI[Action:'cut']) RETURN q.Object_Name
			You can cut with a knife.	[[q.Object_Name], [knife], [knife], [knife]]
			4. What is the relationship between cup and teapot?	MATCH (c:ROI[Object_Name:'cup']-[:]-t:ROI[Object_Name:'teapot']) RETURN c.Object_Name, c.Value, t.Object_Name
			The relationship between a cup and a teapot is that the cup is to the left of the teapot.	[[c.Object_Name, 'cValue', 't.Object_Name'], [cup, 'LEFT', teapot]]
			5. How many knives are there?	MATCH (k:ROI[Object_Name:'knife']) RETURN count(k)
			There are 3 knives.	[[count(k)], [3]]

FIGURE 15. VQA task for Image_A. identified object, generated query and answer.

Image C			Question / Answer	Cypher / Results
			1. What affordance does a knife have?	MATCH (q:ROI[Object_Name:'knife']) RETURN q.Action
			A knife has the affordance to cut.	[[q.Action], [cut], [cut]]
			2. What can I eat?	MATCH (q:ROI[Action:'eat']) RETURN q.Object_Name
			You can eat a banana.	[[q.Object_Name], [banana], [banana], [banana]]
			3. What can I cut?	MATCH (q:ROI[Action:'cut']) RETURN q.Object_Name
			You can cut with a knife.	[[q.Object_Name], [knife], [knife]]
			4. What is the relationship between cup and banana?	MATCH (c:ROI[Object_Name:'cup']-[:]-b:ROI[Object_Name:'banana']) RETURN c.Object_Name, c.Value, b.Object_Name
			The relationship between the cup and the banana is that the cup is to the left of the banana.	[[c.Object_Name, 'cValue', 'b.Object_Name'], [cup, 'LEFT', banana]]
			5. How many knives are there?	MATCH (k:ROI[Object_Name:'knife']) RETURN count(k)
			There are 3 knives.	[[count(k)], [3]]

FIGURE 17. VQA task for Image_C. identified object, generated query and answer.

Image B			Question / Answer	Cypher / Results
			1. What affordance does a banana have?	MATCH (q:ROI[Object_Name:'banana']) RETURN q.Action
			The affordance that a banana has is that it can be eaten.	[[q.Action], [eat]]
			2. What can I drink from?	MATCH (q:ROI[Action:'drink from']) RETURN q.Object_Name
			You can drink from a wine glass.	[[q.Object_Name], [wine_glass]]
			3. What can I cut?	MATCH (q:ROI[Action:'cut']) RETURN q.Object_Name
			You can cut with a knife.	[[q.Object_Name], [knife], [knife]]
			4. Where is the knife in relation to the banana?	MATCH (k:ROI[Object_Name:'knife']-[:]-b:ROI[Object_Name:'banana']) RETURN k.Object_Name, k.name, c.Value, b.Object_Name, b.name
			The knife is positioned behind the banana.	[[k.Object_Name, 'k.name', 'c.Value', 'b.Object_Name', 'b.name'], [knife, 'ROI_5', 'BEHIND', banana, 'ROI_0']]
			5. How many bananas are there?	MATCH (b:ROI[Object_Name:'banana']) RETURN count(b)
			There is 1 banana.	[[count(b)], [1]]

FIGURE 16. VQA task for Image_B. identified object, generated query and answer.

Image D			Question / Answer	Cypher / Results
			1. What affordance does a knife have?	MATCH (q:ROI[Object_Name:'knife']) RETURN q.Action
			A knife has the affordance to cut.	[[q.Action], [cut], [cut]]
			2. What can I eat?	MATCH (q:ROI[Action:'eat']) RETURN q.Object_Name
			You can eat a banana.	[[q.Object_Name], [banana], [banana]]
			3. What can I cut?	MATCH (q:ROI[Action:'cut']) RETURN q.Object_Name
			You can cut with a knife.	[[q.Object_Name], [knife], [knife]]
			4. Where is the knife in relation to the other knife?	MATCH (k1:ROI[Object_Name:'knife']-[:]-k2:ROI[Object_Name:'knife']) RETURN k1.Object_Name, k1.name, c.Value, k2.Object_Name, k2.name
			The knife in ROI_4 is in front of the knife in ROI_5, and conversely, the knife in ROI_5 is behind the knife in ROI_4.	[[k1.Object_Name, 'k1.name', 'c.Value', 'k2.Object_Name', 'k2.name'], [knife, 'ROI_4', 'INFRONT', knife, 'ROI_5', 'BEHIND', knife, 'ROI_4]]
			5. How many knives are there?	MATCH (k:ROI[Object_Name:'knife']) RETURN count(k)
			There are 2 knives.	[[count(k)], [2]]

FIGURE 18. VQA task for Image_D. identified object, generated query and answer.

positions, and frequencies. Five questions were posed for each image; the prompt, coding statement, returned results and the final answer are shown in Figures 15 to 19. Out of 39 objects, 31 were correctly identified with 8 errors, 79% correct, reported as a percentage.

F. EVALUATION OF NS-IL'S DISAMBIGUATION PROFICIENCY USING VARIED QUESTION PHRASINGS

In this experiment, we seek to evaluate the capability of our Neuro-Symbolic model, referred to as NS-IL, in managing disambiguation tasks using Image_A. We focus specifically on the model's proficiency in translating varied phrasing of similar questions into consistent and accurate graph traversal paths. For each question shown in Figure 15 ChatGPT-4 creates five linguistic variations, thereby creating a test suite of diverse queries that probe the linguistic ambiguities between the KG and LLM. Image_A shown in Figure 11 is consistently used as the visual context for these questions. Each set of five variations is presented to NS-IL in a distinct session, ensuring the model starts afresh, without any residual influence from prior interactions. The answers generated are shown in Table 5 and are qualitatively analysed.

For most variations, the answers are consistent across the five different phrasings. For instance, in Set B, the consistent answer is that a banana is edible. Such consistency indicates that the model understands the core intent of the question, irrespective of its phrasing. In Set C, there's an inconsistency. The main question and its variations are asking what in the image can be cut, but several answers indicate that a knife itself can be cut which is a misinterpretation. In Set A, the model returned non-specific answers like "not

Image E			Question / Answer	Cypher / Results
			1. What affordance do bananas have?	MATCH (q:ROI[Object_Name:'banana']) RETURN q.Action
			The affordance that bananas have is that they can be eaten.	[[q.Action], [eat], [eat]]
			2. What can I drink from?	MATCH (q:ROI[Action:'drink from']) RETURN q.Object_Name
			You can drink from a cup, glass, bottle, or any other similar container.	[[q.Object_Name]]
			3. What can I cut?	MATCH (q:ROI[Action:'cut']) RETURN q.Object_Name
			You can cut using a knife.	[[q.Object_Name], [knife], [knife]]
			4. Where is the knife in relation to the banana?	MATCH (k:ROI[Object_Name:'knife']-[:]-b:ROI[Object_Name:'banana']) RETURN k.Object_Name, k.name, c.Value, b.Object_Name, b.name
			The knife is positioned behind and to the left of the banana.	[[k.Object_Name, 'k.name', 'c.Value', 'b.Object_Name', 'b.name'], [knife, 'ROI_4', 'BEHIND', banana, 'ROI_1'], [knife, 'ROI_4', 'LEFT', banana, 'ROI_0']]
			5. How many bananas are there?	MATCH (b:ROI[Object_Name:'banana']) RETURN count(b)
			There are 2 bananas.	[[count(b)], [2]]

FIGURE 19. VQA task for Image_E. identified object, generated query and answer.

specified" or "not provided" for some variations. This could indicate that certain phrasings caused ambiguity or that the image did not provide a clear context for the knife's main affordance. Looking at the cypher code it shows that instead of using the term 'Action', the use of 'Main_Affordance' and 'Primary_Affordance' caused no results to be returned from the KG. In Set D, all five variations aim to understand the spatial relationship between the cup and teapot. The model consistently identified the cup as being to the "left" of the teapot, indicating a strong semantic understanding of spatial relationships in the image. For Set E, the model consistently identified "3 knives," indicating reliable count-based object detection across varied phrasings. In some cases, like in Set C, the change in phrasing led the model to provide syntactically correct but semantically incorrect answers. For instance, saying "a knife can be cut" is a syntactically correct sentence but doesn't make semantic sense in most contexts.

Whilst A4 in Set C showed that ChatGPT-4 type hallucinations have not been eliminated. As when asked, "What

TABLE 5. Comparison of the answer generated by NS-IL of 5 randomly generated questions with the same sense.

No.	Question & Answer
	Main_Question_A: Based on just the image, what affordance does the knife have?
A1	Q: What's the knife's main affordance in the image? A: The affordance that a knife has is to cut.
A2	Q: In the image, what affordance is evident for the knife? A: The affordance evident for the knife in the image is to cut.
A3	Q: From the image, what's the primary affordance of the knife? A: The primary affordance of the knife from the image is not provided.
A4	Q: Observing the image, what affordance can you associate with the knife? A: The affordance you can associate with the knife, as observed in the image, is to 'cut'.
A5	Q: What affordance can be deduced for the knife in the image? A: The affordance that can be deduced for the knife in the image is that it can be used to cut.
	Main_Question_B: Based on just the image, what can I eat?
A1	Q: From the image, what's edible? A: From the image, the edible item is a banana.
A2	Q: What does the image suggest I can eat? A: The image suggests that you can eat a banana.
A3	Q: Observing the image, what appears ready to eat? A: The image shows that what appears ready to eat.
A4	Q: In the image, what can be consumed? A: The image shows items that can be consumed.
A5	Q: Based on the image, what seems suitable to eat? A: It seems that a banana would be suitable to eat.
	Main_Question_C: Based on just the image, what can I cut?
A1	Q: Observing the image, what's cuttable? A: Based on the image you observed, the item that is cuttable is a knife.
A2	Q: What does the image indicate can be cut? A: The image indicates that a knife can be cut.
A3	Q: In the provided image, what seems ready for cutting? A: The image indicates that it's ready for cutting.
A4	Q: What in the image appears to be sliceable with a knife? A: The bread.
A5	Q: From the image, what looks like it can be cut? A: From the image, it appears that a knife can be cut.
	Main_Question_D: Based on just the image, what is the relationship between the cup and teapot?
A1	Q: How does the image portray the cup in relation to the teapot? A: The image portrays the cup, identified as ROI_2, to the left of the teapot, identified as ROI_0.
A2	Q: In the image, how is the cup associated with the teapot? A: In the image, the cup is positioned to the left of the teapot.
A3	Q: Observing the image, what connection is evident between the cup and teapot? A: The connection evident between the cup and the teapot in the image is that the cup is to the left of the teapot.
A4	Q: What's the depicted relationship of the cup and teapot in the image? A: In the image, the cup is depicted to the left of the teapot.
A5	Q: Based on the image, how do the cup and teapot interact? A: The cup is positioned to the left of the teapot based on the image.
	Main_Question_E: Based on just the image, how many knives are there?
A1	Q: How many knives does the image show? A: The image shows 3 knives.
A2	Q: Counting in the image, how many knives can you see? A: In the image, you can see three knives.
A3	Q: Observing the image, how many knives are depicted? A: The image depicts three knives.
A4	Q: From the image, how many knives are visible? A: From the image, you can see that there are 3 knives visible.
A5	Q: In the provided image, how many instances of knives can be found? A: In the provided image, there are 3 instances of knives.

appears sliceable with a knife?" it answers "The bread" which is not an object in the image, however sliceable was also not in the ontology, showing how important the terminology used is. The answer showing the most potential was A1 in Set D which provided specific details like "identified as ROI_2" and "identified as ROI_0," which suggests the model can identify and tag different regions of

TABLE 6. Image_A: comparing question & answers for two models NS-IL (with KG) & ChatGPT4-V (without KG).

No.	Question & Answer
1	Q1: Based on just the image, what affordance does the knife have? ChatGPT_V_A1: The knife has the affordance of cutting or slicing. NS_IL_A1: The affordance that a knife has is to cut.
2	Q2: Based on just the image, what can I eat? ChatGPT_V_A2: You can eat the bananas. NS_IL_A2: You can eat a banana.
3	Q3: Based on just the image, what can I cut? ChatGPT_V_A3: You can cut the bananas with the knife. NS_IL_A3: You can cut with a knife.
4	Q4: Based on just the image, what is the relationship between the cup and teapot? ChatGPT_V_A4: The relationship between the cup and teapot is that the teapot can be used to pour a beverage, likely tea, into the cup. NS_IL_A4: The relationship between a cup and a teapot is that the cup is to the left of the teapot.
5	Q5: Based on just the image, how many knives are there? ChatGPT_V_A5: There are five knives in the image. NS_IL_A5: There are 3 knives.

interest in the image. To investigate if we could reuse this prompt, we submitted Image_C with the question "1. How does the image portray the cup in relation to the teaspoon?" to which the Answer was, "The image portrays the cup in the 'ROI_1' position to the left of the teaspoon in the 'ROI_7' and 'ROI_5' positions". Checking the KG created these nodes, they were indeed.

G. BENCHMARKING NS-IL AGAINST CHATGPT4-V

In these experiments, we aim to assess the efficacy of two distinct models in an image driven, question and answer context. First, we have our Neuro-Symbolic model, denoted as NS-IL which employs ChatGPT-4 as a Q&A interface for users and derives its answers from a knowledge graph. In contrast, we have ChatGPT4-V, an iteration of ChatGPT-4 specifically trained to interpret an image alongside a query and generate an answer independently, without the assistance of any external knowledge graphs or databases. NS-IL and ChatGPT4-V are discussed qualitatively and also quantitatively.

1) QUALITATIVE COMPARISON

Image_A (Figure 11) alongside questions are presented to both models. To maintain consistency, ChatGPT4-V was initialised in a separate chat session for each question set, ensuring that no prior chat history influenced the outcome. The directive given to ChatGPT4-V was to solely utilise the associated image for each query. We then evaluated the accuracy of the responses based on human interpretation. Through a qualitative assessment, we analysed the quality and relevance of the answers to gain a deeper understanding of the areas in which each model either shines or falls short. Table 6 shows the Answers from both models. Note that the NS-IL system will only know about objects in the identified

bounding boxes which can be seen in Figure 11. ChatGPT's prior of common sense can be seen to be advantages in the information used to know in Q3, that knives cut bananas, and in Q4, that teapots are used to pour beverages. It can also be said that NS-IL has only used the information held in the knowledge graph and therefore in Q4 has used the relationship based on location. It should also be noted that NS-IL also knows the exact pixel location where an object is placed in the image and what time it was seen, together with the ability to identify each object individually. All answers were acceptable.

2) QUANTITATIVE COMPARISON

In this experiment, we conduct a comparative analysis between NS-IL and ChatGPT4-V. The experiment involves assessing the answers generated by these models for the set of five images and associated questions found in Experiment VI-E. We quantitatively evaluated the generated answers in terms of the following categories: a) Factual Correctness, as determined by human judgment, b) Factual Incorrectness, as determined by human judgment. c) The correct incorporation of commonsense knowledge in addition to the visual content of the image. d) The inappropriate use of commonsense knowledge about the image. The obtained results are presented in Figure 20, showing answer accuracy and their associated 95% confidence interval. The results indicate NS-IL outperformed ChatGPT4-V in terms of factual correctness, indicating that NS-IL generated responses that were more factually accurate as deemed by a human assessor. Conversely, NS-IL also exhibited a higher rate of factual incorrectness compared to ChatGPT4-V. ChatGPT4-V consistently demonstrated the application of commonsense knowledge alongside the image content, while neither model displayed instances of incorrect commonsense usage. Furthermore, it is worth noting that NS-IL exhibited lower variance in confidence levels across its responses when compared to ChatGPT4-V.

VII. LIMITATIONS

The system was created using a dataset of 16 classes, further research needs to be carried out to test its capacity to scale and handle a more comprehensive and dynamically changing knowledge space. Automated Object and Attribute class creation, could be achieved automatically by utilising an existing VQA dataset such as Visual Genome (101,174 images from MSCoco with 1.7 million QA pairs).

VIII. DISCUSSION

We create a structure of 'Things' that humans internally think about, they are bounded as *Regions of Interest*, identified and classified within a world model that holds the *What* and *Where* information. Through *Questions* and *Answers* about their immediate environment, humans are helped to decide their next actions in the world. The AI representations are strategically aligned with human concepts to enable a useful communication tool. By externalising the independent

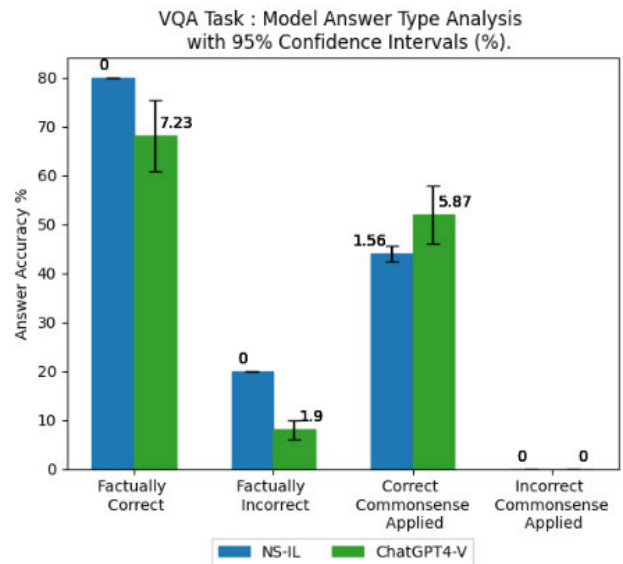


FIGURE 20. Answer analysis for NS-IL & ChatGPT4-V using a dataset of 25 questions and 5 images.

training of new classes using limited sample sizes, the proposed architecture effectively addresses the issue of catastrophic forgetting inherent in neural networks. While unsupervised learning in the creation of the feature space is possible using Autoencoders, our study acknowledges that achieving a conceptually valuable feature space necessitates supervised learning. One potential improvement is to investigate the formation of the feature space, training the image encoder on a dataset that contains classes such as 'pointy,' 'corner,' 'straight,' and 'circle' to enable an improved feature set. Setting specific likelihood thresholds for each GMM class could enhance classification accuracy, with classifications only made if the likelihood surpasses its designated threshold. Such tweaks could bolster our architecture's performance and adaptability across various tasks. The identified classes within the KG, contain quantified uncertainties which could be incorporated into downstream logical reasoning.

In the realm of question-answering systems, foundational models such as large language models (LLM) present distinct advantages, such as no training needed. However, when it comes to precision, consistency, and structured data management, KGs undoubtedly take the lead. These graphs offer a structured data format that ensures a high level of data integrity and reliability, with explicitly mapped out relationships between entities. This structure aids the precise delivery of answers to complex relational queries. On the other hand, while LLMs exhibit adaptability to diverse and unstructured queries and can process multimodal inputs, they lack the inherent data organisation and domain specific tailoring that KGs can offer. KGs' scalability and auditability set them apart, making them an optimal choice for contexts requiring systematic data retrieval and transparency of decision making.

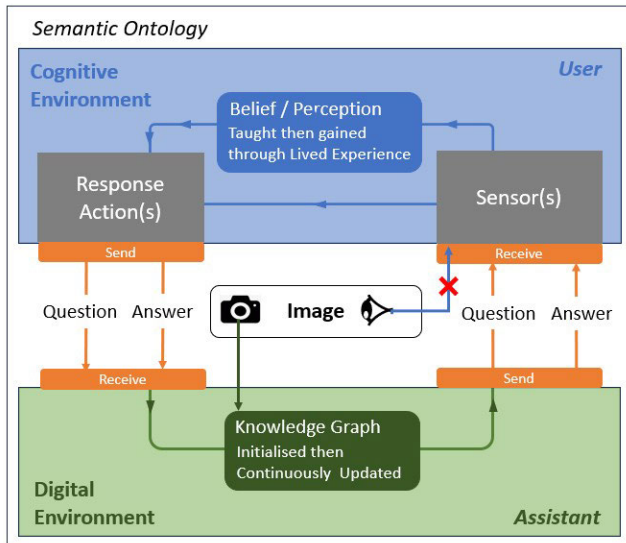


FIGURE 21. Visual dialogue between an AI assistant and human to fulfill a deficit such as sight loss (red cross in the figure) or to receive additional contextualised information.

This Architecture with its three components can easily facilitate an implementation that protects user rights, i.e. The Question and Answer data can remain private to the individual.

IX. CONCLUSION

In this paper, our novel independent GMM class training approach showed the ease with which additional unseen classes can be learnt such as ‘wine bottle’ and ‘wine glass’, trained on small sample sizes without having to retrain the feature space. The vision component’s accuracy in interpreting image encodings was further confirmed through GMMs and KG querying. By combining object and attribute classes, we conceptualised a ‘Zebra’ without requiring direct visual data. Our Vision Question Answering (VQA) evaluation yielded a 79% accuracy rate across diverse image contexts. However, certain phrasings induced ambiguities or misinterpretations, emphasising the importance of consistent terminology. For instance, the model showed strong spatial and count-based object detection capabilities, but some phrasing changes led to semantically incorrect answers. Notably, the model’s potential to tag and identify regions of interest in images was evident. Comparative analysis between the NS-IL system and ChatGPT4-V highlighted the advantages of utilising prior knowledge, while also pinpointing the specificity of NS-IL’s reliance on its knowledge graph for information extraction. We hope this approach spurs the creation of rich concepts that underpin trustable factual question and answering research.

X. FUTURE WORK

The architecture was designed with a possible implementation as an AI Assistant in the future. Figure 21 shows how the VQA task interfaces with a human through a question and answer interface. We anticipate the VQA task evolving

into a Visual Dialogue Task, allowing the Assistant to pose questions and adjust the contents of its knowledge graph to match both the environment and the user’s context.

REFERENCES

- [1] M. Malinowski, M. Rohrbach, and M. Fritz, “Ask your neurons: A deep learning approach to visual question answering,” *Int. J. Comput. Vis.*, vol. 125, nos. 1–3, pp. 110–135, Dec. 2017.
- [2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” 2021, *arXiv:2103.00020*.
- [3] R. Bommasani et al., “On the opportunities and risks of foundation models,” Aug. 2021, pp. 1–212, *arXiv:2108.07258v2*.
- [4] K. Chen, J. Wang, L.-C. Chen, H. Gao, W. Xu, and R. Nevatia, “ABC-CNN: An attention based convolutional neural network for visual question answering,” 2015, *arXiv:1511.05960*.
- [5] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, “Deep modular co-attention networks for visual question answering,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6274–6283.
- [6] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, “From recognition to cognition: Visual commonsense reasoning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6713–6724.
- [7] Y. Zhang, M. Jiang, and Q. Zhao, “Query and attention augmentation for knowledge-based explainable reasoning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15555–15564.
- [8] Y. Zhou, T. Ren, C. Zhu, X. Sun, J. Liu, X. Ding, M. Xu, and R. Ji, “TRAR: Routing the attention spans in transformer for visual question answering,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2054–2064.
- [9] Y. Liu, C. Wu, S.-Y. Tseng, V. Lal, X. He, and N. Duan, “KD-VLP: Improving end-to-end vision-and-language pretraining with object knowledge distillation,” 2021, *arXiv:2109.10504*.
- [10] K. Yi, A. Torralba, J. Wu, P. Kohli, C. Gan, and J. B. Tenenbaum, “Neural-symbolic VQA: Disentangling reasoning from vision and language understanding,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2018, pp. 1031–1042.
- [11] J. Cho, J. Lei, H. Tan, and M. Bansal, “Unifying vision-and-language tasks via text generation,” *Proc. Mach. Learn. Res.*, vol. 139, pp. 1931–1942, Jul. 2021.
- [12] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, “The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision,” in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–28.
- [13] R. Vedantam, K. Desai, S. Lee, M. Rohrbach, D. Batra, and D. Parikh, “Probabilistic neural-symbolic models for interpretable visual question answering,” in *Proc. Int. Conf. Mach. Learn.*, vol. 1, 2019, pp. 6428–6437.
- [14] J.-B. Alayrac et al., “Flamingo: A visual language model for few-shot learning,” 2022, *arXiv:2204.14198*.
- [15] A. M. H. Tiong, J. Li, B. Li, S. Savarese, and S. C. H. Hoi, “Plug-and-play VQA: Zero-shot VQA by conjoining large pretrained models with zero training,” in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2022, pp. 951–967.
- [16] W. Jin, Y. Cheng, Y. Shen, W. Chen, and X. Ren, “Low-resource prompt-based learning for vision-language models,” *Assoc. Comput. Linguistics*, vol. 1, pp. 2763–2775, May 2022.
- [17] Y. Rao, W. Zhao, G. Chen, Y. Tang, Z. Zhu, G. Huang, J. Zhou, and J. Lu, “DenseCLIP: Language-guided dense prediction with context-aware prompting,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18061–18070.
- [18] M. Tsimpoukelli, J. Menick, S. Cabi, S. M. A. Eslami, O. Vinyals, and F. Hill, “Multimodal few-shot learning with frozen language models,” 2021, *arXiv:2106.13884*.
- [19] W. Dai, L. Hou, L. Shang, X. Jiang, Q. Liu, and P. Fung, “Enabling multimodal generation on CLIP via vision-language knowledge distillation,” 2022, *arXiv:2203.06386*.
- [20] E. Karpas, O. Abend, Y. Belinkov, B. Lenz, O. Lieber, N. Ratner, Y. Shoham, H. Bata, Y. Levine, K. Leyton-Brown, D. Muhlgay, N. Rozen, E. Schwartz, G. Shachaf, S. Shalev-Shwartz, A. Shashua, and M. Tenenholz, “MRKL systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning,” 2022, pp. 1–19, *arxiv:2205.00445*.

- [21] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying large language models and knowledge graphs: A roadmap," 2023, *arXiv:2306.08302*.
- [22] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL HLT)*, vol. 1, 2019, pp. 4171–4186.
- [23] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
- [24] T. B. Brown, J. Kaplan, N. Ryder, T. Henighan, M. Chen, A. Herbert-Voss, D. M. Ziegler, G. Krueger, A. Askell, C. Hesse, and S. Mccandlish, "Language models are few-shot learners," 2020, *arXiv:2005.14165*.
- [25] H. Touvron, T. Lavril, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, and G. Lample, "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.
- [26] E. M. Bender and A. Koller, "Climbing towards NLU: On meaning, form, and understanding in the age of data," *Assoc. Comput. Linguistics*, vol. 1, no. 2, pp. 5185–5198, 2020.
- [27] C. Zhang, C. Zhang, S. Zheng, Y. Qiao, C. Li, M. Zhang, S. K. Dam, C. M. T. Y. L. Tun, L. L. Huy, S.-H. Bae, L.-H. Lee, Y. Yang, H. T. Shen, I. S. Kweon, and C. S. Hong, "A complete survey on generative AI (AIGC): Is ChatGPT from GPT-4 to GPT-5 all you need? 2023, *arXiv:2303.11717*.
- [28] T. Li, M. J. Hosseini, S. Weber, and M. Steedman, "Language models are poor learners of directional inference," 2022, *arXiv:2210.04695*.
- [29] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, vol. 1, 2016, pp. 265–283.
- [30] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," 2016, *arXiv:1602.07261*.
- [31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.
- [32] P. Johnston, K. Nogueira, and K. Swingler, "GMM-IL: Image classification using incrementally learnt, independent probabilistic models for small sample sizes," *IEEE Access*, vol. 11, pp. 25492–25501, 2023.
- [33] H. Y. Noh and J. Fagert, "Big data analysis for civil infrastructure sensing," in *Sensor Technologies for Civil Infrastructures*. Amsterdam, The Netherlands: Elsevier, 2022.
- [34] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.



PENNY JOHNSTON received the B.Eng. degree (Hons.) in engineering systems, computing, and control and the M.Sc. degree in engineering management from the University of Loughborough, and the M.Sc. degree (Hons.) in big data from the University of Stirling, in 2018, where she is currently pursuing the Ph.D. degree in AI neuro-symbolic systems. She has achieved C.Eng. Chartered with the Institution of Electrical Engineers and she has worked in the industry for various large manufacturing companies, such as Siemens, Glaxo, AstraZeneca, and Kvaerner. Her research interests include visual perception, machine learning, and neuro-symbolic systems.



KEILLER NOGUEIRA (Associate Member, IEEE) received the B.Sc. degree in computer science from Universidade Federal de Viçosa, Brazil, in 2012, and the M.Sc. and Ph.D. degrees in computer science from Universidade Federal de Minas Gerais, Brazil, in 2015 and 2019, respectively. He is currently a Lecturer with the Division of Computing Science and Mathematics, University of Stirling, U.K. He has published several high-quality papers in leading journals and conferences. His research interests include deep and machine learning, pattern recognition, image processing, computer vision, and remote sensing.



KEVIN SWINGLER received the B.Sc. degree in computing and psychology from the University of Exeter, and the M.Sc. and Ph.D. degrees from the University of Stirling. He is currently the Head of Computing Science and Mathematics with the University of Stirling. His research interests include computer vision and machine learning, particularly applied in health settings.

...