Neural Logic Vision Language Explainer

Xiaofeng Yang, Fayao Liu, Guosheng Lin

***Abstract*—If we compare how humans reason and how deep models reason, humans reason in a symbolic manner with a formal language called logic, while most deep models reason in black-box. A natural question to ask is "Do the trained deep models reason similar as humans?" or "Can we explain the reasoning of deep models in the language of logic?" . In this work, we present NeurLogX to explain the reasoning process of deep vision language models in the language of logic. Given a trained vision language model, our method starts by generating reasoning facts through augmenting the input data. We then develop a differentiable inductive logic programming framework to learn interpretable logic rules from the facts. We show our results on various popular vision language models. Interestingly, we observe that almost all of the tested models can reason logically.**

I. Introduction

The beauty of deep learning models is at present obscured by the cloud of interpretability. This is especially true in the context of vision + language or visual reasoning [1]–[3]. While large-scale pre-trained vision language models [1]–[6] continue to achieve higher performance in benchmarks, it remains unclear how they actually "reason" to solve problems. The lack of interpretability makes debugging the large-scale pre-trained vision language models even harder.

Explainable AI (XAI) methods [7]–[9] aim to give explanations to the decisions made by the deep networks. When these XAI methods are applied to vision language models, the majority of them [10]–[14] simply try to find and visualize the correlation between image patches (language words) and predictions. Despite their success in interpreting pixel or object-level importance, investigating whether a pretrained vision language model can perform reasoning remains an unsolved problem for the following reasons. First, reasoning is compositional, as the decision-making process relies on knowledge of both objects and object relations. However, previous XAI methods largely overlook object relation information. Second, reasoning is multi-hop, requiring multiple steps to determine true or false. Lastly, there is a lack of proper language to describe the reasoning process beyond simply plotting heat maps.

Unlike deep networks, humans are naturally endowed with the ability to reason - we can classify an image even with only a partial view [15]. The reasoning process of humans can be formally described with the language of logic [16]. For example, given a visual question answering question from GQA dataset [17]: "Is the color of the food on the red object left of the small girl that is holding a hamburger brown?" If we unfold the process of human reasoning, the logic expression is $Yes \leftarrow Brown(X) \wedge Food(X) \wedge On(X,Y) \wedge Red(Y) \wedge Left(Y,Z) \wedge Girl(Z) \wedge HoldingBurger(Z)$. By grounding each atom onto the image, we can determine if the answer is affirmative.

Fig. 1. A comparison of NeurLogX with previous XAI methods. Conventional XAI methods focus on interpreting the model decisions by plotting the attention maps. Our method interprets the reasoning process in formal logic languages. It not only considers the object-level information but also object relations.

In this work, we propose **neural logic vision language explainer (NeurLogX)** – a novel framework that interprets the reasoning process of vision language models in the form of logic formulas. Our method offers the same advantages as previous XAI methods: 1) it is applicable to all trained black-box vision language models; 2) it does not require additional human annotations. However, unlike previous methods, our approach interprets the reasoning process in formal logic languages by not only considering the object-level information but also object relations. A comparison of our method and previous XAI methods is demonstrated in Fig. 1.

The proposed NeurLogX follows the same procedure as the logic rule learning system. A logic rule learning system usually contains two steps: 1) retrieving positive and negative facts in first-order-logic form; 2) learning interpretable logic rules from facts with inductive logic programming.

Generate facts from trained deep models. The positive and negative facts are generated by augmenting the testing image. Image samples are generated by randomly masking objects and changing object positions. After that, the image samples are converted to facts in first-order-logic forms. The fact entry is regarded as a positive fact if it produces the same result as the original image and it is regarded as a negative fact if it produces a different result.

Learning interpretable rules with inductive logic programming (ILP). ILP is a symbolic method that learns to derive a hypothesised logic expression that entails as many positive facts and as few negative facts. Traditional inductive logic

Corresponding author: Guosheng Lin.
Xiaofeng Yang and Guosheng Lin are with School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore 639798 (email: xiaofeng001@e.ntu.edu.sg, gslin@ntu.edu.sg)
Fayao Liu is with Agency for Science, Technology and Research (A*STAR), Singapore 138632 (email: fayaoliu@gmail.com)

programming methods [18] learn rules by iterative searching. They start by generating a most general logic rule and then iteratively refine the rule by checking the negative samples. These traditional methods are data-efficient but usually not robust to noise and slow when the number of training facts and types of predicates get larger. Recent advances in symbolic reasoning [19]–[21] propose neural differential inductive logic programming, namely using deep neural networks to estimate the rule learning process. Compared with traditional methods, the differentiable ILP methods have advantages from both sides. They are data-efficient and robust to noise. In this paper, we propose a differentiable ILP network – explain differentiable inductive logic programming network (ex-∂ILP). Compared with other neural ILP methods that are designed for large knowledge bases, we design ex-∂ILP to be a lightweight network that works on a small knowledge base (one image) for vision language tasks.

In experiments, we conduct experiments on two general seen types of vision language models, the one-stream models (UNITER [1]) and two-stream models (LXMERT [2]). For each model, we conduct both qualitative and quantitative analyses on the two popular datasets – VQA [22] and GQA [17] datasets. The paper is organized as follows: in Sec. II, we briefly review conventional explainable AI methods, logic and XAI in vision language models and compare our methods with other methods. In Sec. III, we introduce the basis of first-order-logic and inductive logic programming. In Sec. IV, we introduce the details of our method. In Sec. V, we show experimental results and ablation studies. Finally, we conclude our paper and discuss limitations and future works.

To summarize our contributions:

- We propose neural logic vision language explainer (**NeurLogX**), a framework that interprets the reasoning process of vision language models in formal logic languages. The proposed system is based on our novel explain differentiable inductive logic programming network, which enables fast and data-efficient learning of logic rules.
- We conduct extensive experiments qualitatively and quantitatively on popular vision language models LXMERT [2] and UNITER [1], and VQA [22], GQA [17] datasets.
- Based on our observations of generated logic rules, we give recommendations for ways to improve current vision language models.

II. Related Works

A. Explainable AI (XAI)

Explainable AI methods [7]–[9] aim to interpret the decisions made by deep learning models. Nowadays, as deep learning models are being utilized in healthcare, finance, and climate prediction, the importance of XAI has garnered increasing attention. LIME [11] first perturbs the image by occluding image regions and then uses a linear model to learn the correspondence between image super-pixels and model predictions. CAM [12] generates class activation maps by back-propagate the gradients through global average pooling.

In general, XAI methods can be categorized into two types based on their scope: local methods [11], [12], [23], [24], which explain individual testing instances, and global methods [19], [25], [26], which provide explanations for the entire model. Additionally, XAI can be classified based on when the methods are applied, with intrinsic methods [27]–[29] integrated into the deep models, and post-hoc methods [11], [19], [23]–[26], [30] working with pre-trained models.

Our method falls in the category of local and post-hoc. Nevertheless, as the majority of the previous works focus on generating heat maps as the explanations, our work is the first method that explains the reasoning process of deep models in the formal logic language. Previous works [31], [32] also study the reasoning of neural networks, however, we are the first one that gives logic interpretation.

B. Logic in Vision Language Models

There is a long history of using logic expressions and/or logic rules in deep learning and vision language models [21], [33]. Early attempts [34]–[36] primarily focus on simulated datasets [37]. Many of these works don't explicitly use the term logic. Instead, they convert language expressions to executable "programs". The programs are indeed informal logic expressions of the language descriptions. Compared with formal logic expressions, these programs are restricted in two senses: 1), the programs are usually less diverse than logic predicates; 2), the programs are executed in a chain-like manner, while logic predicates are connected with logic "and", "or" and "not", and it can be executed in "chain-like" and "tree-like" manners. Besides the previously mentioned differences, compared with Johnson *et al.* [34] who uses LSTM to generate executable programs, our method does not need ground-truth program annotations to train and thus can be applied to a wider range of applications. Compared with Yi *et al.* [35] who learns executable programs throughout the training process and requires human crafted program executors, our method does not require such program executors. In fact, the need for human crafted program executors restricts their use case to be on simulated datasets [37]. On real-world datasets like VQA and GQA, it is impossible to design such executors due to the large number of possible programs (logic predicates).

Recent works [38], [39] also use logic to examine and improve the robustness and consistency of vision language models on real-world datasets. VQA-LOL [38] creates new datasets by using simple logical expressions like "not" or "and" to augment current VQA questions. Work [39] proposes a logical implication consistency loss to improve the consistency of vision language models.

C. Vision Language Models and Interpretability

Vision Language pretraining learns image-text representations from large-scale paired vision and language data. There are two common design choices: the one-stream methods and two-stream methods. ViLBERT [40], LXMERT [2] and their consequences [5] use two-stream methods. The two-stream methods usually contain two separate encoders for language and images and one cross vision-language encoder.

Besides two-stream methods, one-stream methods such as VisualBERT [41], Unicoder-VL [42] , UNITER [1] simply concatenate image features and languages and process them through a sequence of transformers. Compared with the two-stream methods, the one-stream methods require more working memories and usually perform better than the two-stream methods. The one-stream methods usually have a smaller model size.

Models finetuned from pretrained parameters achieve state-of-the-art performance on almost all recent vision language leader-board [1]–[3], [43]. However, it remains unclear how reliable these models are or how to interpret the reasoning process. There are attempts to explain vision language models. Position Focused Attention Network [28] is a type of the intrinsic method that enables visualization of attended areas through position focused attention mechanism. Work [13] also examines and visualizes the image attention maps. It is observed that vision language models are able to attend to the right visual positions. The work [14] systematically studies the attention weights of vision language models. They conclude that language modality plays a more important role than the vision modality. Previous works make great attempts in interpreting the feature importance based on attention weights. The question of how the models reason remains unanswered. In this paper, we focus on interpreting the reasoning process of vision language models.

III. First-Order-Logic and Inductive Logic Programming Revisit

First-Order-Logic. A first-order-logic formula is a rule in the form of:

$$P \leftarrow a_1, ..., a_m, \tag{1}$$

where P is the **head** and $a_1, ..., a_m$ is the **body**. The **body** of a first-order-logic formula is formed by logical operation connected **atoms**. For example, Given a first-order-logic formula:

$$\begin{aligned} Grandma(X_1, X_3) \leftarrow & Mother(X_1, X_2) \wedge \\ & Mother(X_2, X_3), \end{aligned} \tag{2}$$

$Grandma(X_1, X_3)$, $Mother(X_1, X_2)$ and $Mother(X_2, X_3)$ are atoms. Logical operations include $\wedge$ **and**,$\vee$ **or**,$\neg$ **not**. Each atom is combined with a **predicate** and n **entities**. In the above case, $Grandma$ and $Mother$ are **predicates**. Formally, a **predicate** is a statement that takes **entity** as input and generates either True or False. The **arity** of a predicate refers to the number of inputs. In the above example, the **arity** of $Grandma$ is 2. **Arity** 2 predicates are also called **binary predicates**. **Arity** 1 predicates are also called **unary predicates**. **Entity** represents a type of object. In the above example, X represents the type of human. In this paper, we use the capital letter X to represent logic variables and x to represent grounded instances, for example, a person John.

Inductive Logic Programming. Inductive logic programming (ILP) is a symbolic artificial intelligence method that learns logic formulas from a set of facts.

Traditional ILP methods. Traditional ILP methods like Progol [44] find a rule hypothesis using the principle of the inverse entailment from facts. These methods are in general data-efficient, but when the scale of data increases, the traditional ILP methods are known to be computationally expensive. In the meantime, the traditional ILP methods are extremely sensitive to noise [45].

Differentiable ILP methods. Large-scale data raise concerns about traditional ILP for their low processing speed and sensitivity to noise. On the other hand, deep learning models can process large-scale data efficiently and are robust to noise. Previous works [20], [45], [46] propose using deep models for ILP. Markov logic network [47] first proposes to learn weights of clauses using a network. Nonetheless, the so-called "network" is a rather simple one without hidden layers and the weights can be learned by counting the number of true evidences. Neural Markov Logic Networks [48] extends Markov logic networks [47] by removing the requirement of predefined premises and learning them together with the clause weights. Neural LP [20] targets the knowledge base reasoning problem. To solve the multi-hop reasoning problems, it designs a LSTM network as the controller to connect the memory and model predictions (ProLog operators) at each reasoning step. Differentiable Inductive Logic Programming [45] proposes a general framework to learn explainable rules from supervised training procedures (e.g. MNIST classification), but it does not specify how to design a proper network to learn weights for the clauses. NLIL [46] uses transformer network [49] to learn logic rules from scene graph generation tasks.

Compared with these methods, our method has two unique contributions. Firstly, we specify how the differentiable ILP methods can be applied to the XAI problems of vision language models, including the ways to generate facts and train the network. Secondly the network design: since our method is a local XAI method that learns to explain one data instance at a time, we use the graph attention network as the backbone and design the network to be a minimal network such that it can achieve fast learning with a small training data size.

IV. NeurLogX

The proposed method **NeurLogX** has two parts: generate positive and negative facts and learn inductive logic rules from the facts. An illustration of the framework is shown in Fig. 2.

A. Generate Facts

To facilitate the learning of inductive logic programming, we first generate facts from the testing instance. The generation of facts is done in two steps: 1), generate image-level samples. 2), symbolize image to atom facts.

Generate image-level samples. Image-level samples are generated by augmenting the image. We perform random object occlusion and random object shifts. For random object occlusion, the system randomly occludes N objects based on initial detection results. For random shift, the objects randomly swap position with other objects by copying and pasting. Notably, the sample generation is performed before the feature extraction process of vision language models. High level features like object detection features usually contain

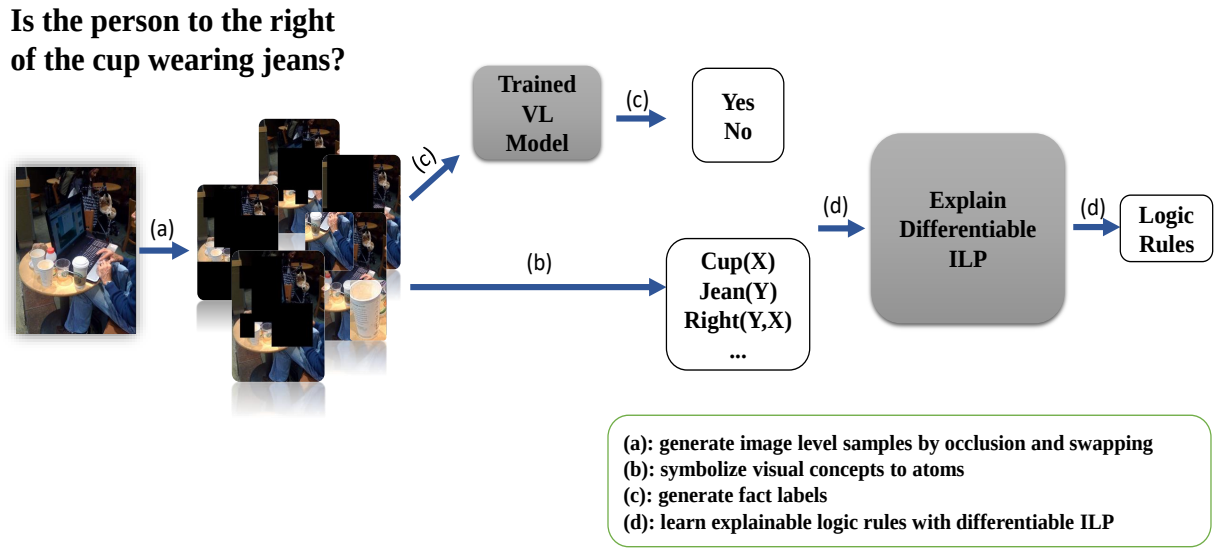


Fig. 2. A detailed illustration of the proposed neural logic vision language explainer. Given an image to learn, we start from generating image level samples by random occlusion and random object swapping. After that, the image objects are converted to symbols. These symbols form the training facts of our ex-∂ILP. In the meantime, we run the trained vision language models to get training labels for the symbolic facts. Finally, the interpretable logic rules are learned with our ex-∂ILP module.

rich context information. Therefore, it's unable to completely mask out information of a certain object, if the occlusion is performed only on the feature level. The fact is regarded as a positive fact if it produces the same result as the original image and it is regarded as a negative fact if it produces a different result.

Symbolize image to atom facts. At this stage, the augmented images are converted to symbolic facts. Each fact is a logical atom. We run an object detection model on the augmented images and extract objects and object attributes. Then we convert them to predicate form. For unary predicates, we choose the types of predicates based on the object classes and attributes in [50]. Binary predicates include four types of spatial relations – $Left$, $Right$, $Front$, and $Behind$. For example, given two detected objects – a tall man standing to the right of a white wall, the facts are:

$\{White(X), Wall(X), Tall(Y), Man(Y), Right(Y, X)\}$.

B. Learn Explainable Rules from Facts

We propose explain-differentiable-ILP (ex-∂ILP) – a minimal differentiable ILP framework for explaining purpose.

Skolem normal form and matrix multiplication. To begin with, similar to previous work [45], [46], we first turn Eq. 1 into Skolem normal form:

$$P \leftarrow P_1(X_1), \ldots, P_m(X_m), \tag{3}$$

Compared with Eq. 1 which may contain nested atoms, Eq. 3 is much easier because it is only composed of a sequence of atoms. Therefore, the problem of finding the best logic rule now becomes finding the best combinations of predicates in predicate space.

To find the best combination of predicates, formally, consider an image described by a $A \times A \times C$ dimensional adjacency matrix $\mathbf{M}$, where A is the number of entities and C is the types of predicates. Each submatrix of the adjacency matrix along C dimension $\mathbf{M_c}$ represents whether two entities are connected by a certain predicate $\mathbf{c}$. Furthermore, consider $\overrightarrow{h_x}$ and $\overrightarrow{h_y}$ to be the one-hot embedding of entity x and y. It is easy to verify that $\overrightarrow{h_x}^T \mathbf{M_c} \overrightarrow{h_y}$ now represents the score of entities x and y connected by predicate c [51]. This formula can be further extended to:

$$score(x, y) = \overrightarrow{h_x}^T \mathbf{M_1} \ldots \mathbf{M_c} \overrightarrow{h_y}. \tag{4}$$

Eq. 4 shows how to calculate the score that x and y are connected by predicate path $1 \ldots c$. In this case, finding the best combination of predicates is equivalent to find the predicate path that maximizes the score value.

Logic Operations. Eq. 4 represents a special form of Eq. 3 that the predicates are connected by logic $\wedge$ **and** operations. We show three types of logic operations logic $\wedge$ **and**, $\vee$ **or**, $\neg$ **not** and how they can be represented by extending Eq. 4. Following previous works [46], [51], we represent $\neg$ **not** of predicate c as $1 - \mathbf{M_c}$. Mathematically, logic $\vee$ **or** operation can be substituted by combining $\wedge$ **and** and $\neg$ **not**. Therefore, to make Eq. 4 fully compatible with the three types of logic operations, the system only needs to change the adjacency matrices to their negation and add the negation to candidate predicates.

Explain-differentiable-ILP (ex-∂ILP). In this section, we show how to learn the Skolem normal form expression with our method, namely learn to find a solution of Eq. 4. We show the model architecture and the learnable parameters. To do that we first show a soft relaxation of Eq. 4:

$$score(x, y) = \overrightarrow{h_x}^T \prod_0^T (\sum_0^c S_c \mathbf{M_c}) \overrightarrow{h_y}, \tag{5}$$

where the parameter S_c is a weighting vector over all possible predicates' adjacency matrices. The value of S_c can be decided by a sequential decision making process [52], [53]. In other words, the value of S_c depends on not only the $\mathbf{M_c}$ values but also previous decisions. In experiments, we use a graph attention network (GAT) [54] to calculate S_c. We regard the $\mathbf{M_c}$ values as graph nodes. Formally, given $\mathbf{M_c}$ values for the predicates, the network first processes the predicate embedding and $\mathbf{M_c}$ (represented as $\mathbf{M_c}$ for simplicity) with learnable weight $\mathbf{W}$ to get an intermediate parameter:

$$e = a(\mathbf{W}\mathbf{M_0}, \ldots, \mathbf{W}\mathbf{M_c}), \tag{6}$$

where a is an activation function.

Then the relative importance s in Eq 5 can be found by performing a softmax operation against all intermediate parameters:

$$S = softmax(e). \tag{7}$$

Similar to other ILP methods [46], [51], to train the network, we sample predicate paths from the positive and negative facts. To integrate the fact labels into the learning system, we treat positive and negative labels as a special type of entity and add a special type of predicate $Relate$ to connect the fact label with every other entity. Given the previous example of a tall man standing to the right of a white wall, the facts are: $\{White(X), Wall(X), Tall(Y), Man(Y), Right(Y, X)\}$. An example of the sampled predicate path could be $\{Man(Y), Relate(Y, True)\}$. The training of network is guided by classifying if the sampled predicate path has the same result as the fact. Specifically, The $\overrightarrow{h_x}^T$ value in Eq. 5 represents the one-hot embedding of starting entity. $\overrightarrow{h_y}$ represents the one-hot embedding of "True" or "False" entity.

A detailed illustration of the model structure is shown in Fig. 3. At each graph attention layer, the network outputs the attention map S. Based on S, we can calculate the score that currently sampled predicate path holds according to Eq. 5 and then train the network based on a classification loss.

Extract Interpretable Rules from Trained Parameters. After the network is trained, the logic rules can be recovered from learned parameters. The algorithm is described in Algorithm. 1. Specifically, to recover logical rules from the trained parameter, we select every step's highest confident predicates and skip this step if the predicate is already selected in the previous step.

V. Experiments

Implementation details. For each image instance, we generate 100 facts to train the explainer. Each fact is generated by randomly occluding 5 objects and swapping the position of 2 objects. The layer of graph attention network is set to 3. We use RELU as the activation function.

Models and datasets. We conduct experiments on two vision language models – the one-stream model UNITER [1] and the two-stream model LXMERT [2]. For both of them, we use their finetuned models on GQA [17] and VQA [22] datasets to see how they reason. Compared with the conventional VQA task, the GQA task requires the model to understand object relations and perform multi-hop reasoning. We show our results qualitatively and quantitatively.

A. Qualitative Experiment on Generated Logic Rules.

We show four representative examples in Fig. 4. Example 1 shows an easy question asking the color of the cake. It could be observed that both of the tested models could generate the correct answer and the ex-∂ILP model could correctly learn the logic rules. Example 2 shows an example that requires compositional reasoning. Both of the models could generate correct results and our proposed framework learns meaningful logic representations. In example 3, we observe that all the generated facts are positive facts. It means that the two models can get the result woman even without looking at the images. We conclude from this example that the models learn dataset bias because only women wear dress in the whole dataset. In example 4, LXMERT generates a wrong result, however, our proposed method shows the reasoning process seems to be accurate. The mistake is that the model wrongly regards the monitor to be the computer. Therefore, based on this observation we can conclude that to improve the accuracy of this model one thing to do is to improve vision language grounding.

B. Selected Predicates vs. Attended Objects.

The selected predicates from ex-∂ILP are regarded as the most important elements in images for decision making. To validate that our method learns correctly, we compare and visualize what our model learns (the predicates) and what the trained model learns (the attention maps). We show four examples in Fig. 5. To visualize attention maps, we plot the top 3 object regions with the highest attention weights. For our ex-∂ILP, we plot all regions with their predicate form in the learned rule. The first two experiments show easy cases where only one object is related to the question. In this case, our model could attend to the same object as the attention maps. Example 3 shows a question with three objects. The top 3 attended objects include the "child" and the cake, while our model chooses the woman and the cake. Example 4 shows a failure case of our method. The question asks "Is the field soft and snowy?" which does not relate to any object. Our proposed method ex-∂ILP fails to generate a logic rule based on predicates since predicates are related to object-level semantic information.

C. Speed comparison: Ours vs. Traditional ILP methods

To further validate the effectiveness of our proposed ex-∂ILP framework, we compare ourselves with traditional ILP methods and compare their training speed. For the traditional method, we use the publicly available implementation of Aleph [18]. Specifically, the Aleph algorithm is listed below:

1) As long as examples exist, select one to generalize. Otherwise stop.

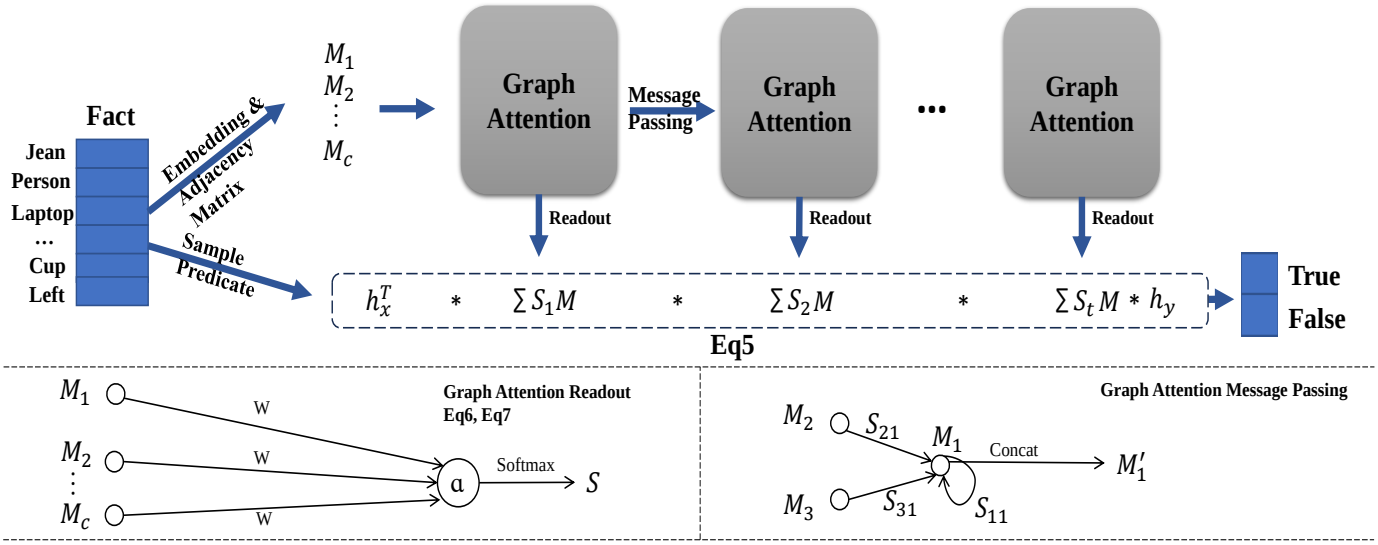


Fig. 3. An illustration of the ex-∂ILP model structure. We use graph attention network to conduct soft predicate selection. At each step, the attention map s_t represents the importance of candidate predicates.

Algorithm 1 Recover Logic Rules from Attention Map.

Input: attention map $\{\mathbf{s_t} \mid t = 1, \ldots, T\}$
Notation: Let $R_t = \{P_1, \ldots, P_l\}$ be the generated rules at step t.
Initialize: $R_0 = \{\}$
for $t \leftarrow 1$ to T **do**
  if $t = 1$ **then**
    Assign $\hat{P}_t = argmax(\mathbf{s_t})$
    Add $\hat{P}_t$ to R
  else
    if $\hat{P}_t \neq argmax(\mathbf{s_t})$ **then**
      Assign $\hat{P}_t = argmax(\mathbf{s_t})$
      Add $\hat{P}_t$ to R
return R_T

2) Construct the most-specific clause (bottom clause) that entails the selected example and is within the language constraints.
3) Find a more general clause that is a subset of the current literals in the clause.
4) Remove examples covered by the current clause.
5) Repeat process from 1

We compare their training speed by performing training on 100 testing instances and calculate the average speed per image. Ex-∂ILP experiments are carried out on a single Nvidia-GTX1080ti GPU, and the Aleph experiments are carried out on Intel i7-8700 CPU. We observe that our proposed method takes only 5 mins to generate rules for each testing instance, while traditional ILP methods take more than 1 hour. The long training time makes traditional ILP unsuitable for explaining model behaviors.

D. Measure Logic Rules Accuracy

To validate the learned ex-∂ILP is consistent with the trained vision language model, we further design a quantitative experiment. Specifically, besides the generated positive and negative facts, we generate additional facts as validation data for the ex-∂ILP module. For each image, we generate 100 additional facts as validation facts. After the training on train facts is finished, we test the generated rule on validation facts. We perform experiments on UNITER and LXMERT and on VQA and GQA datasets. We calculate the validation score by averaging 100 testing instances (10k facts in total). The testing results are shown in Table. I. We observe that on GQA dataset, our generated logic rules using the default setting (3 layers) achieve a high validation accuracy of 95%. While on VQA dataset, our system achieves a validation accuracy of around 80% with both tested models. The major reason for this performance gap is the way the datasets are collected. For VQA dataset, the questions and answers are generated by human crowd-sourcing. Therefore, although the VQA questions require fewer reasoning skills, they are more natural and hard to perform vision language grounding. The GQA questions are templates generated from ground-truth object labels, which greatly ease the grounding process and lead to better validation results. The high validation accuracy proves that our proposed method is consistent in model behaviors with the trained vision

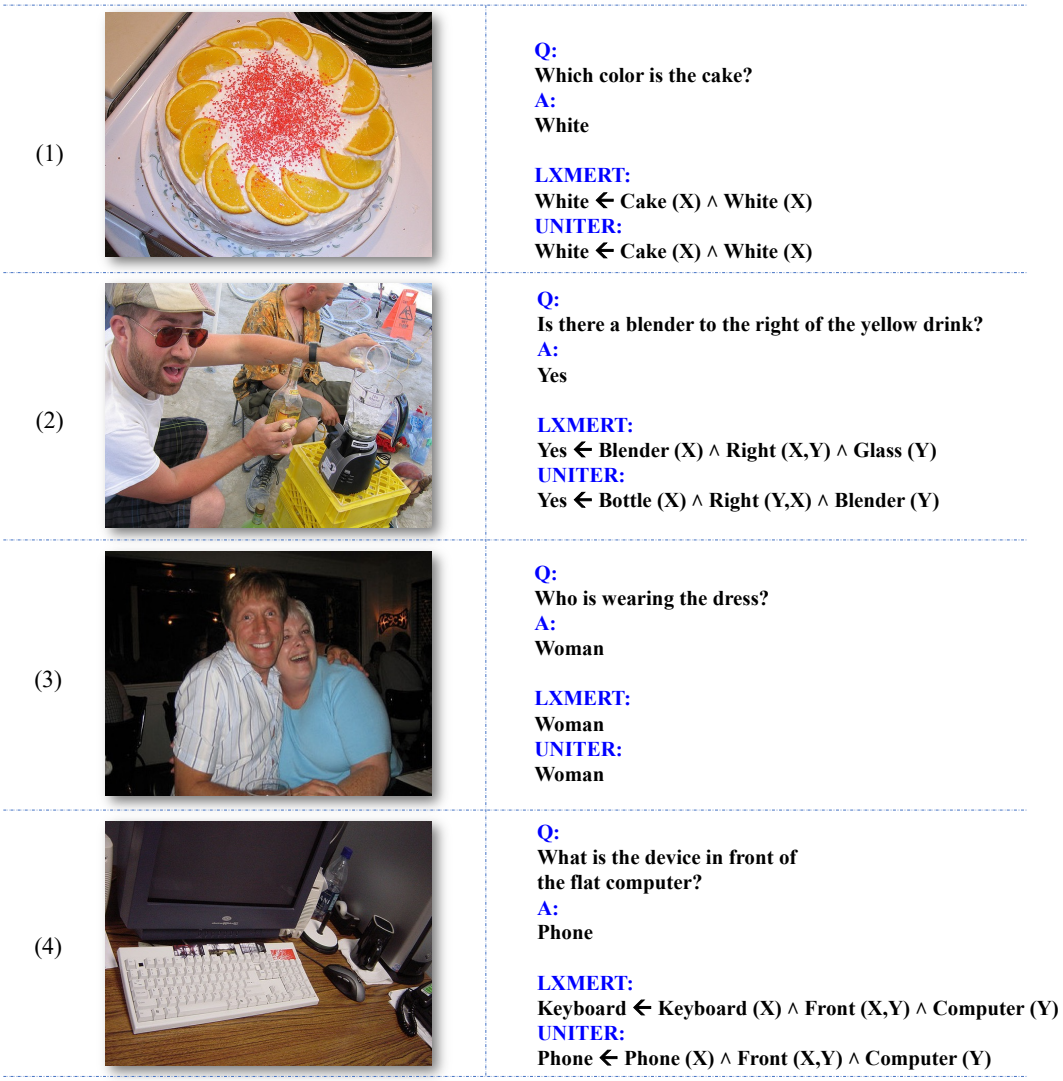


Fig. 4. Qualitative examples of our generated logic rules. Example 1: an easy question asking the color of the cake. Both of the tested models could generate the correct answer and the ex-∂ILP model could correctly learn the logic rules. Example 2: an example that requires compositional reasonings. Both of the models could generate correct results and our proposed framework learns meaningful logic representations. In example 3, we observe that all the generated facts are positive facts. It means that the two models can get the result woman even without looking at the images. In example 4, although the reasoning process seems to be accurate, LXMERT generates a wrong result. The mistake is that the model wrongly regards the monitor to be the computer.

language model.

We also conduct ablation experiments on the number of graph attention layers of our method in Table I. We observe that by reducing the number of layers from 3 to 2, the performance is greatly reduced. However, by increasing the layer to 4, the performance is almost the same as the 3 layer model and the network requires longer processing time.

E. Comparison with other XAI methods

To further validate the accuracy of our proposed method, we compare our method with the famous XAI method LIME [11]. For both methods, we calculate the IoU (Intersection over Union) between the XAI methods' attended objects and the vision language model's attended objects using the vision language model UNITER [1]. A higher IoU score indicates that the method is more faithful to the original vision language model. The experiments are carried out on VQA and GQA datasets and the scores are calculated by averaging results on 100 instances randomly sampled from the validation set. Results are shown in Table II. On VQA dataset, our method achieves an IoU score of 0.53 and on GQA dataset our method achieves an IoU score of 0.62. GQA score is higher because the GQA dataset is more related to object level relations, while VQA dataset contains more questions asking the picture as a whole. Compared with LIME, our proposed method gets higher IoU scores, which indicates that our method can generate more faithful predictions.

F. User Study

We also conduct user studies to quantitatively evaluate our method. Participants are given the original image, question, model's prediction and our generated rules to evaluate whether the generated rules are correct from a human perspective. The image and question pairs are randomly sampled instances from

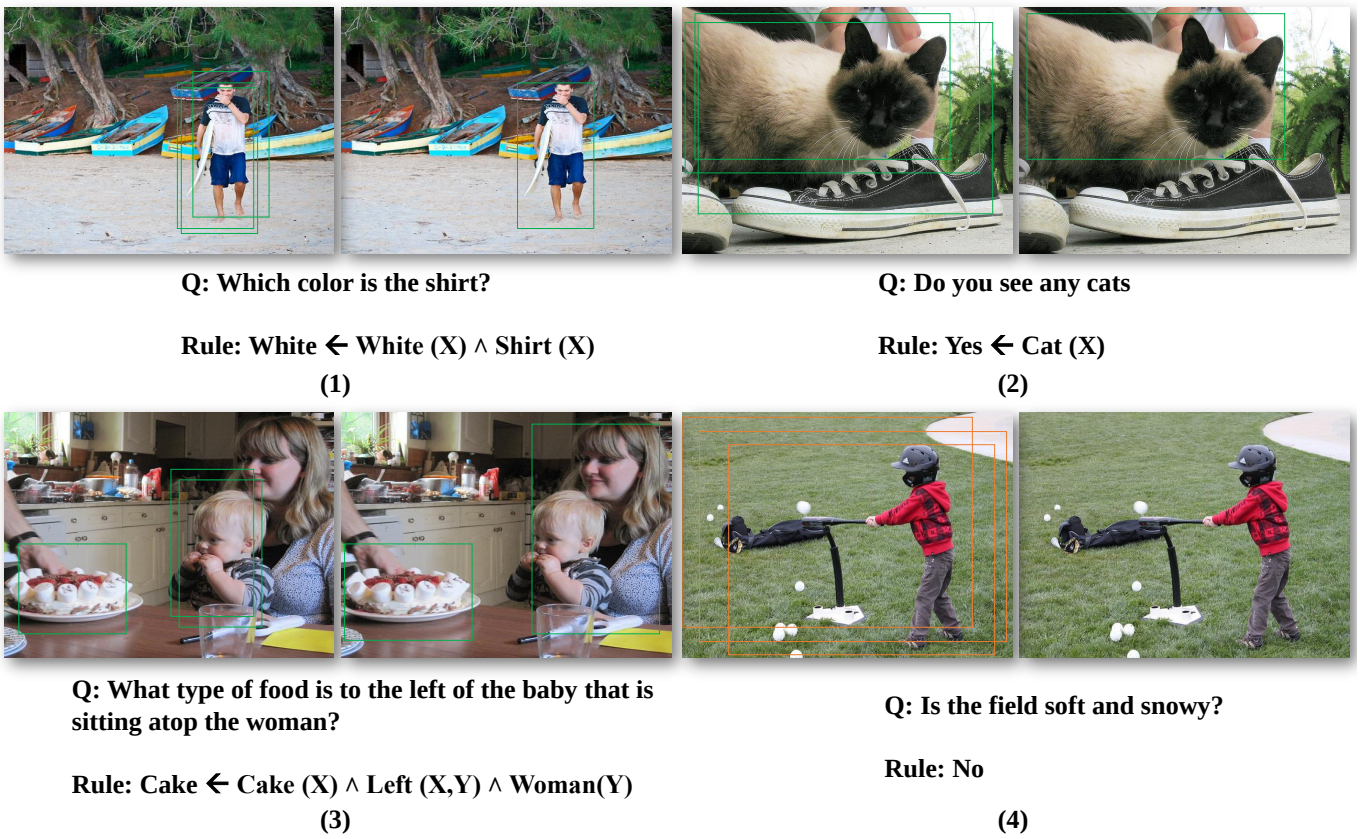


Fig. 5. Selected Predicates vs. Attended Objects. Left: attended objects. Right: objects corresponding to learned predicates. The first two experiments show easy cases where only one object is related to the question. Our model could attend to the same object as the attention maps. In example 3, the image shows a question with three objects. The top 3 attended objects include the "child" and the cake, while our model chooses the woman and the cake. Example 4 shows a failure case of our method. The question is not related to any object. Our proposed method ex-∂ILP fails to generate a logic rule based on predicates because the predicates are generated from object labels.

Method	VQA	GQA
LXMERT (2 layers)	63%	72%
LXMERT (3 layers, default)	81%	**95%**
LXMERT (4 layers)	**82%**	**95%**
UNITER (2 layers)	64%	77%
UNITER (3 layers, default)	**83%**	**95%**
UNITER (4 layers)	83%	95%

TABLE I
VALIDATION SCORES AND ABLATION EXPERIMENTS. BY USING THE DEFAULT SETTING (3 LAYERS), ON GQA DATASET, OUR GENERATED LOGIC RULES ACHIEVE A HIGH VALIDATION ACCURACY OF 95%. WHILE ON VQA DATASET, OUR SYSTEM ACHIEVES A VALIDATION ACCURACY OF AROUND 80% WITH BOTH TESTED MODELS.

Method	VQA	GQA
Ex-∂ILP (Ours)	**0.53**	**0.62**
LIME	0.45	0.47

TABLE II
COMPARISON WITH OTHER XAI METHODS. WE CALCULATE THE IOU SCORE BETWEEN THE XAI METHODS' ATTENDED OBJECTS AND THE VISION LANGUAGE MODEL'S ATTENDED OBJECTS. EXPERIMENTAL RESULTS INDICATE THAT OUR METHOD IS MORE FAITHFUL TO THE ORIGINAL VISION LANGUAGE MODEL THAN LIME.

VQA and GQA datasets. We use a total of 100 pairs for user study. Based on our user studies, it is found that 72% of votes think our method generates correct logic expressions from a human perspective.

*G. Pretrained Vision Language Model Therapy with **NeurLogX***

In this section, we discuss our key observations and how these observations could be used to improve current vision language models.

Vision Language Models Learn Good Reasoning Skills. Based on our observations in Fig. 4 and accuracy on validation set, we conclude that current vision language models inherent good implicit reasoning skills. These reasoning skills can be concreted by our proposed ex-∂ILP.

Visual Features and Vision Language Grounding are the next bottlenecks. We witness that many wrong predictions are due to wrong visual classes or vision language grounding. In fact, even if wrong vision language grounding happens, the reasoning process itself seems correct most of the time. This leads to a conclusion that improving the quality of visual features will likely be more effective than improving the network structure.

Dataset Bias is another problem preventing the vision language models from learning better representations. The dataset bias [55], for example only woman wearing dress

in this dataset, leads the model to learn biased features. In such a case, the model will simply remember the correlation between woman and dress instead of looking at the image for an answer. The fact generation process of our proposed method can be used to identify dataset bias. Given an image and language pair, if the generated facts are all positive (same as the default image), it means the model could generate an answer no matter what image is given. Potential therapy to dataset bias is to generate counter-factual training data. For example, if the "only woman wears dress" bias is detected, including "man also wear dress" in training data will help.

Extend object-level features to higher-level features. The majority of current vision language models take object-level features as input, so as our explainer. We found drawbacks while interpreting some of the problems that require understanding global context. This observation raises concerns on using regional features. Since the pioneer work of [50], using object detector detected regional features becomes the golden rule for training vision language models. Indeed, regional features provide more prior semantic information than raw image pixels. However, they lack information of global and non-object features. Future works can be done by replacing regional features with raw image pixels and training the vision language model at a larger scale.

Combine sub-symbolic features with symbolic representations. Symbolic representations refer to discrete and interpretable symbols like image labels. Sub-symbolic features refer to continuous and uninterpretable features like regional features of images. Based on our observation in Section V, systems using solely symbolic representations can also perform vision language tasks, yet achieve interpretability in the meantime. Future vision language models can be improved by combining symbolic representations with sub-symbolic features. This is validated by Oscar [3]. However, Oscar [3] only uses object-level symbolic representation (Unary predicates in our terms). Further exploration can be done by using binary predicates to represent object relations.

VI. Conclusions, Limitations and Future Works

Conclusions. We propose NeurLogX – a novel framework that interprets the reasoning process of vision language models in the form of logic formulas. Similar to previous XAI methods, our method works on all trained black-box vision language models and it does not need extra human annotations. Unlike previous methods, our method interprets the reasoning process in formal logic languages. The interpretation not only considers object-level information but also object relations.

Our system contains two steps: 1), retrieving positive and negative facts in first-order-logic form. 2), learning interpretable logic rules from facts with inductive logic programming. We perform experiments on popular vision language models and datasets and show both qualitative and quantitative results. Based on our observations, we also give recommendations on how the vision language models could be further improved.

Limitations. Currently, our method is limited to interpreting object-level concepts and relations. This is the problem commonly seen in recent symbolic AI methods. For questions involving global concepts such as weather, season, and time, our method usually can not generate accurate results. Also, our method can only handle a subset of first-order-logics. For example, due to the constraints of Skolem normal form logics, our method can not be used to explain counting problems, for example, "how many kids are playing at the ground?".

Future Works. Future works can be done by integrating our method into the pretraining of deep vision language models. There will be two benefits: firstly, the new vision language can achieve intrinsic interpretability. Secondly, the new model combines both symbolic features and sub-symbolic features and could potentially achieve better performance.

Future works can also be done by extending our methods to handle all first-order-logic rules instead of just a subset of first-order-logic.

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2018-003), the MOE AcRF Tier-1 research grants: RG95/20, and the OPPO research grant. This research is also partly supported by a A*STAR AME Programmatic Fund (A20H6b0151).

References

[1] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Universal image-text representation learning," in *European conference on computer vision*. Springer, 2020, pp. 104–120.

[2] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 5100–5111.

[3] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei *et al.*, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *European Conference on Computer Vision*. Springer, 2020, pp. 121–137.

[4] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[5] J. Lu, V. Goswami, M. Rohrbach, D. Parikh, and S. Lee, "12-in-1: Multi-task vision and language representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 437–10 446.

[6] W. Kim, B. Son, and I. Kim, "Vilt: Vision-and-language transformer without convolution or region supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5583–5594.

[7] A. Das and P. Rad, "Opportunities and challenges in explainable artificial intelligence (XAI): A survey," *CoRR*, vol. abs/2006.11371, 2020. [Online]. Available: https://arxiv.org/abs/2006.11371

[8] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable AI: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, pp. 18–28, 2021. [Online]. Available: https://doi.org/10.3390/e23010018

[9] G. Vilone and L. Longo, "Explainable artificial intelligence: a systematic review," *CoRR*, vol. abs/2006.00093, 2020. [Online]. Available: https://arxiv.org/abs/2006.00093

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6572

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[12] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[13] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "What does bert with vision look at?" in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5265–5275.

[14] J. Cao, Z. Gan, Y. Cheng, L. Yu, Y.-C. Chen, and J. Liu, "Behind the scene: Revealing the secrets of pre-trained vision-and-language models," in *European Conference on Computer Vision*. Springer, 2020, pp. 565–580.

[15] S. Theodoridis and K. Koutroumbas, *Pattern recognition*. Elsevier, 2006.

[16] S. K. K. Langer and S. K. Langer, *An introduction to symbolic logic*. Dover New York, 1953.

[17] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6700–6709.

[18] S. Muggleton and L. De Raedt, "Inductive logic programming: Theory and methods," *The Journal of Logic Programming*, vol. 19, pp. 629–679, 1994.

[19] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, "Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model," *The Annals of Applied Statistics*, vol. 9, no. 3, pp. 1350–1371, 2015.

[20] F. Yang, Z. Yang, and W. W. Cohen, "Differentiable learning of logical rules for knowledge base reasoning," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 2319–2328. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/0e55666a4ad822e0e34299df3591d979-Abstract.html

[21] W. Wang and Y. Yang, "Towards data-and knowledge-driven artificial intelligence: A survey on neuro-symbolic computing," *CoRR*, vol. abs/2210.15889, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2210.15889

[22] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6904–6913.

[23] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 839–847.

[24] H. Li, Y. Tian, K. Mueller, and X. Chen, "Beyond saliency: understanding convolutional neural networks from saliency prediction on layer-wise relevance propagation," *Image and Vision Computing*, vol. 83, pp. 70–86, 2019.

[25] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller, "Unmasking clever hans predictors and assessing what machines really learn," *Nature communications*, vol. 10, no. 1, p. 1096, 2019.

[26] M. Ibrahim, M. Louie, C. Modarres, and J. Paisley, "Global explanations of neural networks: Mapping the landscape of predictions," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 279–287.

[27] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1721–1730.

[28] Y. Wang, H. Yang, X. Qian, L. Ma, J. Lu, B. Li, and X. Fan, "Position focused attention network for image-text matching," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 3792–3798.

[29] R. Agarwal, L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, and G. E. Hinton, "Neural additive models: Interpretable machine learning with neural nets," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4699–4711, 2021.

[30] W. Wang, C. Han, T. Zhou, and D. Liu, "Visual recognition with deep nearest centroids," in *International Conference on Learning Representations (ICLR)*, 2023.

[31] Y. Ge, Y. Xiao, Z. Xu, M. Zheng, S. Karanam, T. Chen, L. Itti, and Z. Wu, "A peek into the reasoning of neural networks: Interpreting with structural visual concepts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2195–2204.

[32] K. Xu, J. Li, M. Zhang, S. S. Du, K. Kawarabayashi, and S. Jegelka, "What can neural networks reason about?" in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=rJxbJeHFPS

[33] T. Rocktäschel and S. Riedel, "End-to-end differentiable proving," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 3788–3800. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/b2ab001909a8a6f04b51920306046ce5-Abstract.html

[34] J. Johnson, B. Hariharan, L. Van Der Maaten, J. Hoffman, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Inferring and executing programs for visual reasoning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2989–2998.

[35] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic VQA: disentangling reasoning from vision and language understanding," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 1039–1050. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/5e388103a391daabe3de1d76a6739ccd-Abstract.html

[36] S. Aditya, Y. Yang, and C. Baral, "Integrating knowledge and reasoning in image understanding," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, S. Kraus, Ed. ijcai.org, 2019, pp. 6252–6259. [Online]. Available: https://doi.org/10.24963/ijcai.2019/873

[37] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2901–2910.

[38] T. Gokhale, P. Banerjee, C. Baral, and Y. Yang, "Vqa-lol: Visual question answering under the lens of logic," in *European conference on computer vision*. Springer, 2020, pp. 379–396.

[39] S. Tascon-Morales, P. Márquez-Neila, and R. Sznitman, "Logical implications for visual question answering consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6725–6735.

[40] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Advances in Neural Information Processing Systems*, 2019, pp. 13–23.

[41] L. H. Li, M. Yatskar, D. Yin, C. Hsieh, and K. Chang, "Visualbert: A simple and performant baseline for vision and language," *CoRR*, vol. abs/1908.03557, 2019. [Online]. Available: http://arxiv.org/abs/1908.03557

[42] G. Li, N. Duan, Y. Fang, M. Gong, D. Jiang, and M. Zhou, "Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training." in *AAAI*, 2020, pp. 11 336–11 344.

[43] J. Dong, X. Li, and C. G. Snoek, "Predicting visual features from text for image and video caption retrieval," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3377–3388, 2018.

[44] S. Muggleton, "Inverse entailment and progol," *New generation computing*, vol. 13, no. 3, pp. 245–286, 1995.

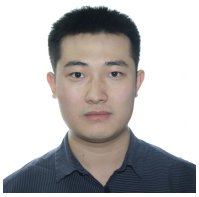
[45] R. Evans and E. Grefenstette, "Learning explanatory rules from noisy data," *Journal of Artificial Intelligence Research*, vol. 61, pp. 1–64, 2018.

[46] Y. Yang and L. Song, "Learn to explain efficiently via neural logic inductive learning," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=SJlh8CEYDB

[47] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, pp. 107–136, 2006.

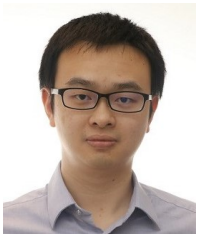
[48] G. Marra and O. Kuželka, "Neural markov logic networks," in *Uncertainty in Artificial Intelligence*. PMLR, 2021, pp. 908–917.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
[50] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.
[51] K. Guu, J. Miller, and P. Liang, "Traversing knowledge graphs in vector space," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, Eds. The Association for Computational Linguistics, 2015, pp. 318–327. [Online]. Available: https://doi.org/10.18653/v1/d15-1038
[52] M. L. Littman, *Algorithms for sequential decision-making*. Brown University, 1996.
[53] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=ryGs6iA5Km
[54] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ
[55] N. Ouyang, Q. Huang, P. Li, Y. Cai, B. Liu, H.-f. Leung, and Q. Li, "Suppressing biased samples for robust vqa," *IEEE Transactions on Multimedia*, vol. 24, pp. 3405–3415, 2021.



Xiaofeng Yang is a PhD student at the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in computer vision and machine learning.

Fayao Liu is a research scientist at Institute for Infocomm Research (I2R), A*STAR, Singapore. She received her PhD in computer science from the University of Adelaide, Australia in Dec. 2015. Before that, she obtained her B.Eng. and M.Eng. degrees from National University of Defense Technology, China in 2008 and 2010 respectively. She mainly works on machine learning and computer vision problems, with particular interests in self-supervised learning, few-shot learning and generative models. She is serving as an associate editor for IEEE Transactions on Circuits and Systems for Video Technology (TCSVT).



Guosheng Lin is an Assistant Professor at the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He received his PhD degree from The University of Adelaide in 2014. His research interests are in computer vision and machine learning.