SMAC: An Interpretable Reasoning Network for Visual Question Answering

Minghan Wang,[†] Hao Yang,[†*] Shiliang Sun,[‡] Yao Deng,[†] Ying Qin[†]
[†]Huawei Translate Service Center, Beijing, China
[‡]East China Normal University, Shanghai, China
{wangminghan, yanghao30, dengyao3,qinying}@huawei.com, slsun@cs.ecnu.edu.cn

Abstract

In the visual question answering (VQA) task, many successful works focus on building end-to-end predictive models, but the interpretability of the reasoning process is ignored, which is however very important for evaluating the trustworthiness of the model. The recent MAC-network (Hudson and Manning 2018) achieves state-of-the-art results on the VQA task which demonstrates the effectiveness of differentiable reasoning models. However, for MAC, interpreting the reasoning process by visualizing the attention map often fails to clearly show the logic of multi-step reasoning. In this paper, we propose SMAC (Symbolic MAC) to improve the interpretability in the following points. (1) Intent classification is introduced to make the question understanding explainable. (2) We propose the Translate Unit (TU) to translate the reasoning process into the formalized query language for interpreting, as well as providing explicit guidance on the reasoning cell in the training phase. We further enlarge the feature space to leverage more information by incorporating the image pixel features and the object-specific features simultaneously, which follows the multi-view learning framework. Experiments demonstrate that SMAC is able to achieve competitive performance on a large-scale and realistic GQA (Hudson and Manning 2019) benchmark and show well interpretability evidence with symbolic intermediate outcomes.

Introduction

Many interesting tasks emerged during the development of machine learning and deep learning where the data source can be different, for example, image and text have been used in tasks like Visual Question Answering (VQA) (Antol et al. 2015). In VQA task, given a question and an image, the model has to understand the question and find the answer from the image. There can be many applications of VQA such as medical diagnostic (Lau et al. 2018) and text-to-image information retrieval (Xie, Shen, and Zhu 2016).

More formally, we can define VQA as a parameterized function $A = f(Q, I; \theta)$ which can be fitted by the neural network, where Q represents for question, I is the image and A is the answer. There are typically three stages in the VQA task. 1). Question understanding, which aims to extract semantic features from the question including the intention and related entities (Young et al. 2018). 2). Image understanding, which focuses on detecting the objects in the image and producing the representation of them. The extracted features can be considered as a knowledge base. 3). Reasoning, where text features and image features are fused and applied by a reasoning function (e.g. a classifier) to produce the answer.

Attention mechanism (Bahdanau, Cho, and Bengio 2015) has proved its efficiency in retrieving information from continuous feature space in both Computer Vision (CV) and Natural Language Processing (NLP), which successfully becomes a commonly used method to modelling the fusion and reasoning. Works like (Hudson and Manning 2019) and (Lu et al. 2018) apply attention to map the question into the feature space of images and filter out required representation of specific objects. However, one-step reasoning only allows the question to attend the image once which sometimes fails to handle complex logic in the question, for example, *Is the fence which is to the right of the animal orange or gray?*, which requires the model to correctly find the mentioned object (fence) and identifies its color. Therefore, multi-step reasoning architecture is designed in MAC network (Hudson and Manning 2018) where the recurrent cell keeps reading information from the image and writing into the memory embedding in fixed reasoning steps P, which achieves state-of-the-art results on CLEVR (Johnson et al. 2017) dataset.

In recent years, explainable AI (Samek, Wiegand, and Müller 2017) becomes an active topic which mainly investigates methods of improving the interpretability of deep learning models. Interpretability can be considered as an important feature for applications of AI such as medical or self-driving. Although attention models provide us a chance to explain the behavior of the model by visualizing the attention maps, it is still difficult to explicitly describe the decision making process because most attention maps only model the correlations but not causality, which is particularly important in a reasoning task like VQA. For MAC-network, it successfully models the multi-step reasoning in an end-to-end manner but is still not able to clearly explain how the model reasons in each step, which motivates us to further improve the interpretability of MAC and propose

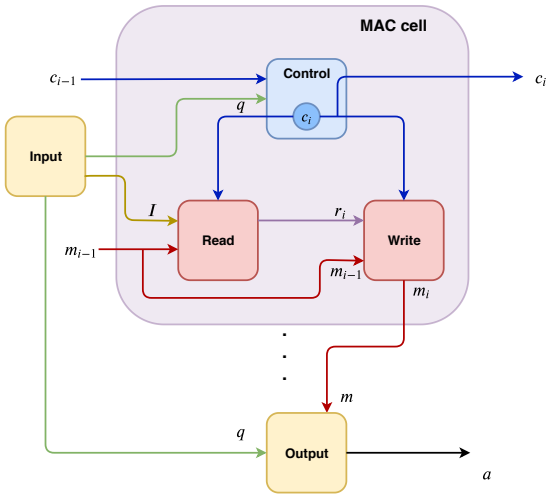

Figure 1: MAC network proposed in (Hudson and Manning 2018)

Symbolic MAC (SMAC).

In this paper, we propose SMAC to improve the interpretability of MAC by enabling the model produce intermediate outcomes including the predicted intent of the question and the query language statements generated from reasoning process, which can be considered as bring the techniques of symbolic AI back to NN model for interpretation. The contribution of our work can be summarized as follows:

- We propose SMAC to improve the interpretability of MAC by introducing intent classification to make problem understanding explainable and the Translate Unit to generate query language statements during reasoning.
- The experimental results performed on GQA benchmark validate that SMAC can reasoning logically in real world VQA tasks, as well as preserving the predictive performance.

Related Work

In this section, we introduce the main components of MAC especially pointing out the parts that we aim to improve.

Memory, Attention, and Composition (MAC)

Compositional VQA tasks like recognizing attributes of objects, reasoning logical relations, counting and comparisons (Hudson and Manning 2018) require the model to be capable of executing a multi-step reasoning to capture the logical operations of the question. MAC is deliberately designed for solving such problems depending on its recurrent **Memory, Attention and Composition** cell which is similar to the Neural Turing Machine (Graves, Wayne, and Danihelka 2014). The MAC cell is composed of three units, **the Control Unit**, **the Read Unit** and **the Write Unit**, as shown in Figure 1. Besides the MAC cell, the input unit and output unit are employed for encoding the input data (question and image) and making predictions on the answer, respectively.

The Input & Output Unit Before reasoning, the question and the image must be encoded to acquire the distributed representation. For the question, MAC uses a biLSTM to produce $q = [\overrightarrow{h}, \overleftarrow{h}], q \in \mathbb{R}^{2d}$ representing the concatenation of the last hidden states in both directions and the contextual word embedding denoted as $\mathbf{cw} = [cw_1, ...cw_S], \mathbf{cw} \in \mathbb{R}^{S \times d}$. For P reasoning step $i = 1, ..., P$, a linear transformation is applied to q and acquires the position-awared $q_i \in \mathbb{R}^{2d}$. For the image G, it can be encoded by a pre-trained model such as ResNet101 (He et al. 2016) for extracting pure pixel features or faster R-CNN (Ren et al. 2015) for object based features, represented as $I \in \mathbb{R}^{K \times d}$ where K equals $H \times W$ or the number of detected objects. The encoded question and image are passed into the reasoning function to produce a memory embedding, denoted as $m = f(Q, G)$. Finally, the memory m is decoded by the output unit combined with q, resulting in $\hat{a} = \text{OU}(m, q), \hat{a} \in \mathbb{R}^{|\mathcal{A}|}$ where $\mathcal{A}$ is the vocabulary of the answer.

The Control Unit (CU) The control unit can be denoted as a function $c_i = \text{CU}(c_{i-1}, q_i, \mathbf{cw}), c_i \in \mathbb{R}^d$ which generates a control signal conditioned on the previous control signal, the question representation and the contextual word embedding. More specifically, the previous control signal c_{i-1} and position-awared question representation q_i will combine and attend the context **cw** with a concatenation attention (Luong, Pham, and Manning 2015), which extracts information from the context **cw**, represented as $c_i = \text{Attn}_{\text{CU}}([q_i, c_{i-1}], \mathbf{cw})$.

The Read Unit (RU) The read unit retrieves information from the image guided by the control signal c_i and the previous memory $m_{i-1} \in \mathbb{R}$, which can be defined as $r_i = \text{RU}(c_i, m_{i-1}, I), r_i \in \mathbb{R}^d$. Same as the control unit, the attention model is applied on the image I with a fused query signal rv_i produced by c_i and m_{i-1}, resulting in $r_i = \text{Attn}_{\text{RU}}(rv_i, I)$.

The Write Unit (WU) The write unit takes the previous memory m_{i-1}, the control signal c_i and retrieved information r_i as input, manipulates the previous memory with them and outputs the new memory, denoted as $m_i = \text{WU}(m_{i-1}, c_i, r_i)$. In addition, for further improving the reasoning performance, self-attention and memory-gate are applied in the write unit to handle long range reasoning, and can be re-written as $m_i = \text{WU}(c_i, r_i, \mathbf{m})$ where $\mathbf{m} = [m_1, ..., m_{i-1}]$. At the last step of reasoning, the memory m_P is passed into the output unit to predict the answer.

Although MAC is particularly appropriate for multi-step reasoning, it suffers from the problem that the reasoning process cannot be clearly interpreted. In the original work, highlighting the region of the image based on the attention map could indeed prove that the model is able to observe the correct object with respect to the entity mentioned in the question. But it fails to demonstrate that the observation is related to a specific logical operation such querying, verifying, counting or comparing. Therefore, we propose SMAC to solve this problem in a simple but effective way.

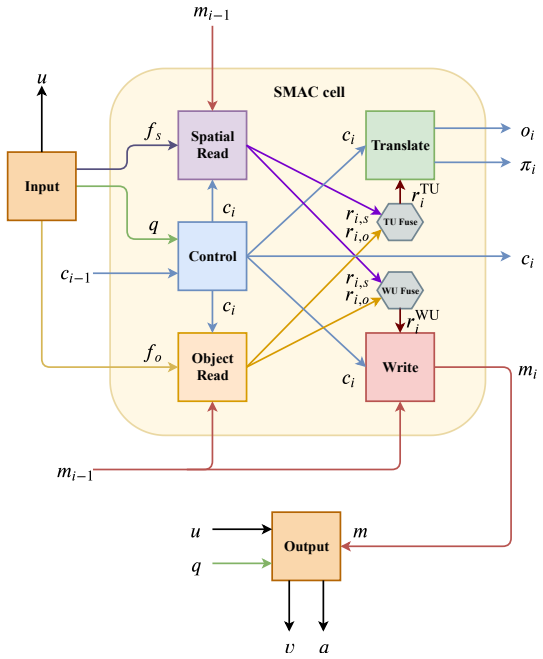

Figure 2: The SMAC proposed in our work. The main difference between our work and MAC is: 1) We introduce the pre and post intent classification represented as u and v. 2) The Translate Unit is proposed to decode the reasoning process. 3) Multi-view learning with the Dual Read Unit is adopted.

Symbolic MAC

In this section, we introduce our proposed Symbolic MAC as shown in Figure 2, starting from the dual read unit to the intent classification and the Translate Unit.

The Dual Read Unit

For real world VQA tasks, the input image is often noisy and full of semantic relations between objects, which requires multi-view features extracted from different levels and perspectives (Sun et al. 2019). Therefore, we consider using the multi-view learning to enlarge the feature space by fusing the object-level and spatial-level feature extracted with Faster-RCNN (Ren et al. 2015) and ResNet101 (He et al. 2016), respectively, to provide the model with a better sight. More specifically, we duplicate the original read unit without changing the internal structure, and retrieve information from both spatial feature $f_s \in \mathbb{R}^{K_s \times d_s}$ and object feature $f_o \in \mathbb{R}^{K_o \times d_o}$ guided by m and c, resulting in $r_{i,s} \in \mathbb{R}^{d_s}$ and $r_{i,o} \in \mathbb{R}^{d_o}$. Then, we fuse two features by simply concatenating them and performing a ELU (Clevert, Unterthiner, and Hochreiter 2016) non-linear transformation. Not that the fusion is performed twice with different parameter sets, aiming to provide different representations for the Write Unit and the Translate Unit (introduced subsequently), respectively:

$$r_i^{\text{WU}} = \text{ELU}(W_{\text{WU}}^{(d_s+d_o)\times d}[r_{i,s}; r_{i,o}] + b_{\text{WU}}^{d}) \quad (1)$$

$$r_i^{\text{TU}} = \text{ELU}(W_{\text{TU}}^{(d_s+d_o)\times d}[r_{i,s}; r_{i,o}] + b_{\text{TU}}^{d}). \quad (2)$$

The Dual Read Unit mainly solves the problem that original MAC in (Hudson and Manning 2019) only uses object-based features or spatial feature which lacks the information of the semantic relations among objects. Fused features could provide a more comprehensive understanding of the image.

Intent Classification

Intent classification and slot filling are critical techniques for Natural Language Understanding (NLU) (E et al. 2019) especially in task-oriented dialogue systems (Chen et al. 2017). Intent classification is often treated as a sentence classification task where predicted intentions can be used to invoke specific task operators. For an end-to-end generative dialogue system, intent classification is not directly used for the answer generation (Chen et al. 2017) but can be used as an auxiliary task under the multi-task learning framework, which could improve the quality of the question representation. More importantly, the predicted intent is an interface for explaining the language understanding. Therefore, we adapt the concept into the MAC and propose a pre-intent classifier $P(u|q;\theta_{\text{pre}})$ and post-intent classifier $P(v|m,q,u;\theta_{\text{post}})$ denoted as follows:

$$\hat{u} = W_u^{d\times|\mathcal{U}|}\phi_u(q) + b_u^{|\mathcal{U}|} \quad (3)$$

$$\hat{v} = W_v^{d\times|\mathcal{V}|}\phi_v([m;q;\hat{u}]) + b_v^{|\mathcal{V}|}. \quad (4)$$

The pre-intent classifier is applied after the question encoder, where q is the question representation, performed with a non-linear transformation through ϕ, and $\mathcal{U}$ is the predefined pre-intent set. However, while encoding the question, the model has not "seen" the image yet, which means the predicted intent can only be high-level context-free intentions such as "*VERIFYING*", "*COMPARING*". Therefore we name it as pre-intent classification.

After the reasoning process, the memory m takes the information about the answer as well as the extracted semantic of the question. As mentioned in (Hudson and Manning 2019), the validity and plausibility measure whether the answer is reasonable according to the question (e.g. asking about the color but not answering about size), which is tightly correlated with the understanding of the intention. Therefore, we further add a post-intent classifier after the reasoning cell, making a fine-grained intent classification based on the memory, the predicted pre-intent as well as the question, denoted in Eq (4), where $\mathcal{V}$ is the post-intent set.

$\hat{u}$ and $\hat{v}$ are logits outputs from two classifiers, which can be used for computing the loss with the cross-entropy, resulting in $L(\theta_{pre})$ and $L(\theta_{post})$. We deliberately use simple classifier (i.e. two-layered NN) rather than deep architectures, aiming to preserve the feature extracting task in the backbone but not parameters of intent classifiers. In addition, we further exploit $\hat{u}$ and $\hat{v}$ in the prediction of the answer, denoted as:

$$\hat{a} = W_a^{d\times|\mathcal{A}|}\phi_a([m;q;\hat{u};\hat{v}]) + b_a^{|\mathcal{A}|}. \quad (5)$$

By adding the intent classification module, the language understanding becomes transparent and interpretable.

The Translate Unit

A reasoning process can be expressed as a logical expression, which can be set operations like $\vee$,$\wedge$ and $\neg$, or high-level functional programs. In MAC, the reasoning process works internally and can only be interpreted by visualizing the attention map, which is not clear enough, especially for complex logic. Therefore, we propose the **Translate Unit (TU)** to translate the reasoning process into a formalized query language (functional programming statements) defined by the GQA dataset (Hudson and Manning 2019). In the dataset, each question is associated with a query language (QL) statement, for example, the question "*What is on the white wall?*" has the QL "*select: wall → filter color: white → relate: ,on,s → query: name*", which defines a series of operations and arguments. Therefore, we can use such QL statements to supervise the reasoning process, and translate it back to QL in the inference phase for interpretation. However, simply applying the sequence-to-sequence framework to decode the control flow into original QL makes it difficult for the model to learn and also extremely increases the reasoning length. To solve such problem, we use an alternative form of the QL provided by the dataset, which splits the statement into list of operation and argument pairs such as "*[(select, wall), (filter color, white),...]*". We further split binary operations like choosing attributes,"*(choose, red ‖ green)*", or operations with more arguments like verifying relations of two objects, "*(verify relation, (ball,on,table))*" into unary operations: "*[(choose, red), (choose, green)]*" and "*[(verify_relation_s, ball), (verify_relation_p, on), (verify_relation_o, table)]*". In this way, all operations are unary, which simplifies the decoding process.

To use the operation sequence, we propose the translate unit (TU) to decode the reasoning process from control signal c and retrieved information r. More formally, we define $P(o_i|c_i;\theta_o)$ and $P(\pi_i|c_i,r_i,o_i;\theta_\pi)$ as the probability distribution of the operation and argument, respectively. For reasoning step i, the distribution can be learned via a neural network:

$$\hat{o}_i = W_o^{d\times|\mathcal{O}|}c_i + b_o^{|\mathcal{O}|} \quad (6)$$

$$\hat{\pi}_i = W_\pi^{(2d+|\mathcal{O}|)\times|\Pi|}[c_i;r_i;\hat{o}_i] + b_\pi^{|\Pi|}, \quad (7)$$

where $\mathcal{O}$ and Π are pre-defined operation and argument sets. We can consider such two functions as a sequence generator which can be trained via teacher forcing, and the loss function can be formulated as:

$$L(\theta_o) = -\frac{1}{P}\sum_i^P\sum_j^{|O|} o_{i,j}\log P(o_{i,j}) \quad (8)$$

$$L(\theta_\pi) = -\frac{1}{P}\sum_i^P\sum_k^{|\Pi|} \pi_{i,j}\log P(\pi_{i,k}). \quad (9)$$

In the setting of MAC, the max reasoning step P is fixed, and the model only uses the last step memory m_P to produce the answer, which cannot be compatible with both easy and difficult questions requiring variable reasoning lengths. Therefore, we propose a method to dynamically control the reasoning length. More specifically, we add a "$<$ EOR $>$" tag at the end of the operation sequence. Then, when the model generates a "$<$ EOR $>$" in step i, we stop the reasoning and use the i-th memory m_i to produce the answer. This approach prevents from introducing potential noise in remaining steps and reduces the risk of gradient explosion occurred in the recurrent structure.

Training

To train the model, we merge losses of auxiliary tasks with the loss of the main task (i.e. the answer loss) together. This results in $L(\theta)$:

$$L(\theta) = \sum_t^T \lambda_t L(\theta_t), \quad (10)$$

where $t \in \{u,v,o,\pi,a\}$, and hyper-parameter λ_t is the weight of specific loss. All parameters can be learned jointly via maximum likelihood estimation (MLE) under the multi-task learning framework (Ruder 2017).

Experiment

In this section, we introduce the experimental setup including the dataset, the implementation details, the hyper-parameters of the model as well as the evaluation metrics. We compare our model with the state-of-the-art works based on the same dataset and evaluation metrics.

Dataset

We use the recently introduced GQA (Hudson and Manning 2019) dataset to train and evaluate our model. The dataset contains 113K images and 22M questions. More detailed statistics can be found in Table 1. The questions of the dataset are provided in two groups (i.e. all and balanced). The "balanced" group is a small subset of the "all" group, which are sampled according to the distribution of the answer, resulting in a more uniformed distribution. However, the size of the "balanced" set is too small (943,000) where each image is associated with only 13 questions, which is not enough for training comparing with the "all" set (Q:I = 193:1), and therefore, we train our model with the "all" set. In addition, the dataset has been split into train, validation and test-dev set with the proportion of 87:12:1. There is also a submission set which is used for evaluating the model on the official GQA website [1], and the experimental result reported in this paper is evaluated on this set.

For the image data, the dataset has already provided the extracted spatial and object features with ResNet101 and faster R-CNN, respectively, which can be directly used. Each image has the spatial features with the shape of [7 $\times$ 7 $\times$ 2048], and object features with the shape of [100 $\times$ 2048] representing the padded object feature map.

[1] https://cs.stanford.edu/people/dorarad/gqa/challenge.html

# Train question	14,305,356
# Val question	2,011,853
# TestDev question	172,174
# Test question (Submission)	4,237,524
# Question word	3,097
# Answer word	1,878
# Pre-intents	5
# Post-intents	106
# Operations	98
# Arguments	2,729

Table 1: The detail of the dataset (# is abbr. of number)

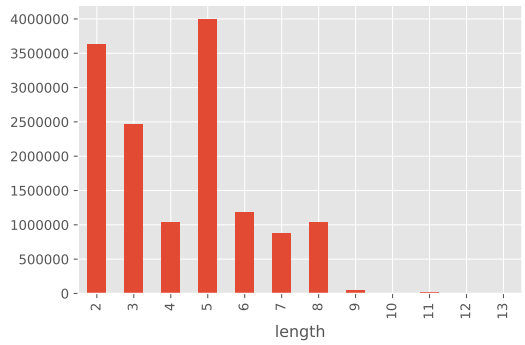

Figure 3: The length of operation sequence, which indicates that most questions requires less than 8 operations

Model Configurations

Comparing with the original MAC mentioned in (Hudson and Manning 2019) which sets the max reasoning step $P = 4$, we extend the reasoning length to 8 due to the statistical results shown in Figure 3 which indicates that most questions need about less than 8 steps of reasoning. Questions with more than 8 steps are trimmed in our experiment.

The dimensionality d of the model is set to 1024 for computational efficiency. Dropout with masking probability of 0.15 is used to prevent overfitting. We use Adam (Kingma and Ba 2015) with initial learning rate of 1e-4 for optimization. The model is trained on a Tesla V100 GPU with the batch size of 512 for 5 epochs, where each epoch requires about 5 hours. Our model is implemented with PyTorch.

Evaluation Metrics

We mainly use the overall accuracy as well as customized metrics proposed in the GQA dataset (Hudson and Manning 2019) which are introduced as follows:

- Binary: The accuracy of questions associated with answers like yes/no or choosing from two candidates.
- Open: The accuracy of open domain questions like querying attributes or relations about objects.
- Validity: Measures whether a given answer is in the question scope (e.g. responding some color to a color question).
- Plausibility: Whether the answer is reasonable or corresponding to common sense.
- Consistency: Whether there are conflicts for answers associated with similar topics or questions (e.g. responding green about an apple that has been identified as red in other answers).
- Distribution: Measures if the model only predicts high frequency answers (e.g. yes/no) but not less frequent ones.

The model is compared with MAC, Bottom-up (Anderson et al. 2018) (the winner of 2017 VQA challenge) and several baseline methods including CNN and LSTM, which have been mentioned in (Hudson and Manning 2019). The detailed evaluation results are shown in Table 2. We find that the performance of SMAC is competitive.

Explainablity and Performance Analysis

In this section, we provide examples to show the evidence that we can use translated QL to express the reasoning process. Figure 4 shows an example where the model generates the operation and argument relating to the attention map highlighted on the original image for each step. Step 1 to 4 correspond to operations of attending specific objects in the image which can be easily explained with attention maps. From step 5 and 6 we can see that the judgement of colors cannot be clearly shown via attention maps but can be expressed through symbols, which demonstrates that complicated logic can be well interpreted by SMAC.

From the results in Table 2, SMAC achieves top 2 performance in 4 out of 7 metrics. For overall accuracy, SMAC is inferior to MAC but is competitive with Bottom-Up (the winner of 2017 VQA challenge).

To further improve the accuracy of our model, we can consider adapting some techniques and tricks from MAC. For example, we can use the self-attention and memory gate proposed in MAC to perform hierarchical mapping between previous reasoning steps. However, this approach will significantly increase the computational cost. In addition, the fusion of the object-spatial features in current SMAC are relatively naive, which often cannot provide high-quality representations of the image, especially when the number of detected objects is large.

Conclusion

In this paper, we propose SMAC based on MAC, which mainly investigates the method of improving the interpretability of reasoning in the VQA task. We conduct experiment on the GQA dataset and show the evidence that the reasoning process can be clearly explained via query language. At the same time, we achieve competitive results in parts of evaluation metrics. In our future work, more techniques and tricks will be incorporated to further improve the accuracy, while preserving the well interpretability in the current work.

References

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down at-

Metric	CNN	LSTM	CNN+LSTM	Bottom-Up	MAC	SMAC (Ours)	Humans
Open ↑	1.74	22.69	31.80	**34.83**	**38.91**	32.55	87.4
Binary ↑	36.05	61.90	63.26	66.64	**71.23**	**67.97**	91.2
Validity ↑	35.78	**96.39**	96.02	**96.18**	96.16	**96.18**	98.9
Plausibility ↑	34.84	**87.30**	84.25	84.57	84.48	**85.08**	93.1
Consistency ↑	62.40	68.68	74.57	78.71	**81.59**	**81.56**	98.4
Distribution ↓	19.99	17.93	7.46	**5.98**	**5.34**	11.75	-
Accuracy	17.82	41.07	46.55	**49.74**	**54.06**	49.16	89.3

Table 2: Performance of different method evaluated on GQA dataset. Top 2 results are highlighted.

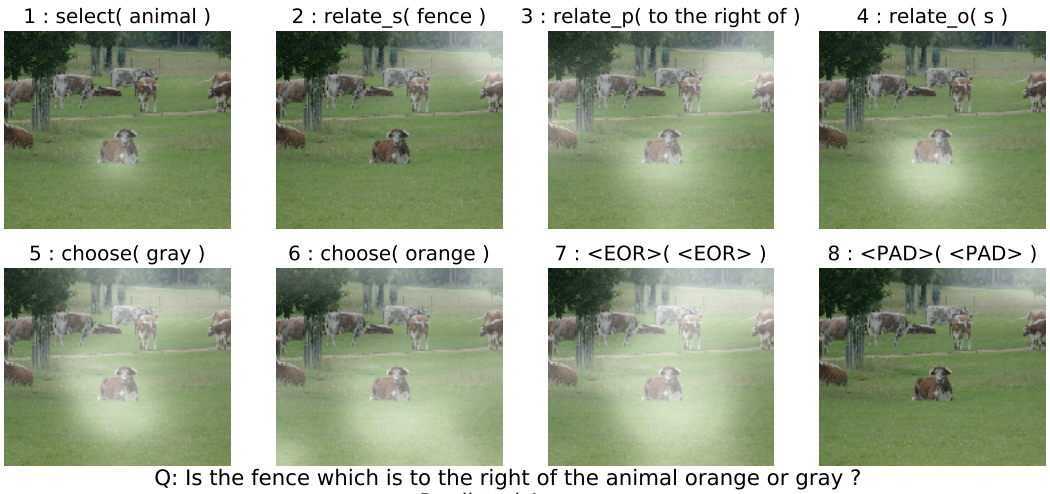

Figure 4: This figure shows an example of associating the translated QL statements with attention maps to interpret the behavior of SMAC. "< PAD >" represents for padding of remaining steps.

tention for image captioning and visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA*, 6077–6086.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. VQA: visual question answering. *CoRR* abs/1505.00468.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, Conference Track Proceedings*.

Chen, H.; Liu, X.; Yin, D.; and Tang, J. 2017. A survey on dialogue systems: Recent advances and new frontiers. *SIGKDD Explorations* 19(2):25–35.

Clevert, D.; Unterthiner, T.; and Hochreiter, S. 2016. Fast and accurate deep network learning by exponential linear units (elus). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

E, H.; Niu, P.; Chen, Z.; and Song, M. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, Italy*, 5467–5471.

Graves, A.; Wayne, G.; and Danihelka, I. 2014. Neural turing machines. *CoRR* abs/1410.5401.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA*, 770–778.

Hudson, D. A., and Manning, C. D. 2018. Compositional attention networks for machine reasoning. In *6th International Conference on Learning Representations, Vancouver, BC, Canada, Conference Track Proceedings*.

Hudson, D. A., and Manning, C. D. 2019. GQA: a new dataset for compositional question answering over real-world images. *CoRR* abs/1902.09506.

Johnson, J.; Hariharan, B.; van der Maaten, L.; Fei-Fei, L.; Zitnick, C. L.; and Girshick, R. B. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA*, 1988–1997.

Kingma, D. P., and Ba, J. 2015. Adam: A method for

stochastic optimization. In *3rd International Conference on Learning Representations, San Diego, CA, USA, Conference Track Proceedings*.

Lau, J. J.; Gayen, S.; Ben Abacha, A.; and Demner-Fushman, D. 2018. A dataset of clinically generated visual questions and answers about radiology images. *Scientific Data* 5:180251.

Lu, P.; Li, H.; Zhang, W.; Wang, J.; and Wang, X. 2018. Co-attending free-form regions and detections with multi-modal multiplicative feature embedding for visual question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA*, 7218–7225.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal*, 1412–1421.

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, Quebec, Canada*, 91–99.

Ruder, S. 2017. An overview of multi-task learning in deep neural networks. *CoRR* abs/1706.05098.

Samek, W.; Wiegand, T.; and Müller, K. 2017. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *CoRR* abs/1708.08296.

Sun, S.; Mao, L.; Dong, Z.; and Wu, L. 2019. *Multiview Machine Learning*. Springer.

Xie, L.; Shen, J.; and Zhu, L. 2016. Online cross-modal hashing for web image retrieval. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA.*, 294–300.

Young, T.; Hazarika, D.; Poria, S.; and Cambria, E. 2018. Recent trends in deep learning based natural language processing. *IEEE Comp. Int. Mag.* 13(3):55–75.