Exploring logical consistency and viewport sensitivity in compositional VQA models

Gabriela Sejnova, Michal Vavrecka, Michael Tesar and Radoslav Skoviera *

***Abstract*—The most recent architectures for Visual Question Answering (VQA), such as TbD or DDRprog, have already outperformed human-level accuracy on benchmark datasets (e.g. CLEVR). We administered advanced analysis of their performance based on novel metrics called consistency (sum of all object feature instances in the scene (e.g. shapes) equals total number of the objects in the scene) and revealed only 56% consistency for the most accurate architecture (TbD). In respect to this finding, we propose a new method of the VQA training, which reaches 98% consistency. Furthermore, testing of the VQA model in real world brings out a problem with precise mimicking of the camera position from the original dataset. We therefore created a virtual environment along with its real-world counterpart with variable camera positions to test the accuracy and consistency from different viewports. Based on these errors, we were able to estimate optimal position of the camera. The proposed method thus allows to find the optimal camera viewport in the real environment without knowing the geometry and the exact position of the camera in the synthetic training environment.**

***Index Terms*—neural module networks, visual question answering, compositionality, CLEVR dataset**

I. Introduction

One of the key goals in artificial intelligence and robotics is to design architectures which are interactive and able to communicate with humans in natural way. To achieve this, it is necessary to find a way to map between the real-world sensory inputs and their natural language description. One way to achieve grounded language knowledge is to design architectures inspired by early language acquisition in small children. In our previous papers, we proposed several architectures for unsupervised mapping (grounding) between words and visual features [1][2][3][4][5]. These models were based on cross-situational learning of mapping between sentences of variable length and visual features.

One of the current tasks focusing on visual reasoning and its interconnection with natural language is Visual Question Answering (VQA) [6]. The goal is to provide the correct answer based on an image and a question in natural language. The best results in terms of accuracy and generalization ability have been obtained by biologically inspired, compositional models for VQA. These models decompose the query into logical primitives which are then individually processed by specialized networks - inspired by brain physiology in cognitive tasks. Our preliminary evaluation of the N2NMN model [7] described in [8] showed that despite its high accuracy on the original CLEVR dataset, the model lacks the ability to consistently count objects in the scene. In this paper, we have selected other state-of-the-art (SOTA) models from [9], [10] and [11] and compared their ability to consistently find all features presented in one scene. We have chosen the CLEVR benchmark dataset [12] for our evaluation, which consists of synthetic images with object primitives and questions regarding the object features and spatial relations. Due to the complex modular structure of the questions, the task requires multistep reasoning about the image and minimizes the possibility to "guess" the answer.

Because the SOTA systems reach 56% accuracy in logical consistency on the CLEVR dataset, here we propose a novel, systematic method to train the architecture using specific questions which target all object properties from each category. In the next stage, we test whether the improved architecture is also able to consistently understand scenes from the real world (collected using both simulator and a robotic manipulator which is not perfectly aligned with the fixed camera position in the CLEVR dataset). These methods allow us to capture the scene from various perspectives. When we use these scenes in the test stage, we can compare the accuracy and consistency error to find the best matching viewport. Therefore, it is not necessary to know the exact geometry, neither the exact position of the camera in the virtual environment. As both the accuracy and consistency of the VQA architectures have shown to be sensitive to the viewport change, we can use them as the estimator of original camera location.

II. Related Work

The compositional approach towards VQA was first introduced in [13] as a system of neural module networks (NMN). Each module unit is a neural network trained to extract a specific feature, such as counting given objects in a scene or filtering of a color. The question is first processed using a semantic parser and then translated into a sequence of neural modules, which are applied over the image to output the final correct answer. The main drawback of the pilot models was that the rules for module chaining had to be manually specified and thus decreased robustness of the model. In [14], the improved architecture learns to chain the neural modules on the fly and updates their weights to obtain better performance for novel structures. Furthermore, the authors in [15] avoided the hand-tuned semantic parsers by defining the function vocabulary and general module structure, and then training the model using reinforcement learning. In a more advanced model, the N2NMN [7], the construction of neural module sequence is learned in an unsupervised fashion or a semi-supervised fashion with unsupervised fine-tuning. The overall accuracy of this model on the CLEVR dataset is around 89%, i.e. very close to human-like performance (92.6%) [12].

The FiLM algorithm [10] induced a large increase in accuracy for the CLEVR dataset (97.7%) using feature-wise affine conditioning. The FiLM layers enable the network to modulate

*Gabriela Sejnova and Michael Tesar are with the Department of Cybernetics at the Czech Technical University. sejnogab@fel.cvut.cz; tesarm11@fel.cvut.cz. Michal Vavrecka and Radoslav Skoviera are at the Czech Institute of Informatics, Robotics, and Cybernetics of the Czech Technical University. michal.vavrecka@cvut.cz

its visual layers with the question input through implicit multi-step reasoning.

In the DDRprog model proposed in [11], the NMN module structure is combined with the Neural Programmer-Interpreter framework (NPI, [16]). The NPI approach interleaves program prediction and execution, so that the output of one module is used to predict the next module. The second contribution of DDRprog is a forking mechanism, which enables the model to maintain stack-based logical tree structures. Compared to FiLM, DDRprog is using explicit modelling of the underlying question structure. Such approach broadens the range of logical operations which can be executed and increases generalization capability. One of the most advanced models so far is Transparency by Design (TbD) networks [9]. Here the authors adapted the program generator described in [15], but redesigned each module according to its specific function, similarly as in [7] and [13]. Because only attention masks are passed between the modules, the resulting model is highly intuitive and interpretable. Although we have tested also other models, we have selected the TbD framework for implementation and evaluation due to its overall impressive performance and transparency.

As many of the recent VQA approaches reach human level accuracy on the CLEVR dataset (see Table 1. for comparison), one could consider this research area as solved. However, a deeper analysis presented in this paper came to a different conclusion. We made a post-hoc evaluation of the TbD [9] models and found out that the understanding of the scene is still very different from human thinking. We adopted a novel methodology for the evaluation of CLEVR dataset that raises a question whether the above mentioned systems are capable to consistently represent and understand relationships among objects in the scene.

The second goal of our paper is dedicated to the analysis of neural module networks accuracy when trained on a synthetic dataset and tested on real world counterparts. We therefore implemented the neural module network architecture into the robot Kuka IIWA LBR 7, the industrial robotic manipulator, presented it with real objects similar to CLEVR primitives and analyzed the result. Currently, there are only few robot implementations of the VQA task, and we are not aware of any model based directly on a compositional architecture and tested on compositional dataset similar to CLEVR. One mildly similar work is an attention-based VQA model called Dynamic Memory Networks from [17], which was implemented into a robotic head platform. The model was trained on the MS COCO dataset and after implementation, the accuracy of image content recognition was at first around 38 % and reached 82 % after retraining on custom images.

We are also not aware of any paper that will adopt consistency metrics neither for synthetics nor for real world datasets.

III. The Architecture

Our architecture is an implementation of the TbD model proposed by [9]. The model is used as the VQA core, while the visual input is imported as a static image from the robot's camera for the real world scenario and as a rendered image for the virtual scenario. For detailed mathematical equations of the TbD model see the original paper [7].

Method	Overall	Count	C. Num.	Exists	Q. Attr.	C. Attr.
N2NMN	88.8	68.5	84.9	85.7	90.0	88.8
Human	92.6	86.7	86.4	96.6	95.0	96.0
RN	95.5	90.1	93.6	97.8	97.1	97.9
PG+EE	96.9	92.7	98.7	97.1	98.1	98.9
FiLM	97.6	94.5	93.8	99.2	99.2	99.0
DDRprog	98.3	96.5	98.4	98.8	99.1	99.0
MAC	98.9	97.2	99.4	**99.5**	99.3	99.5
TbD	**99.1**	**97.6**	**99.4**	99.2	**99.5**	**99.6**

TABLE I
PERFORMANCE OF SELECTED ALGORITHMS ON THE CLEVR DATASET. THE COLUMNS STANDS FOR SPECIFIC SKILL (COUNTING, COMPARISON) AND THE ROWS REFER TO PARTICULAR IMPLEMENTATION.

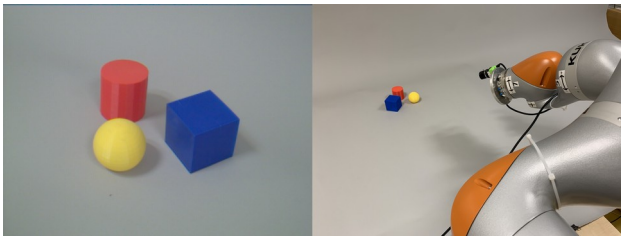

Fig. 1. Collection of the real-world data for model evaluation. Images were taken using Basler Dart 5 MPx camera mounted on the Kuka robot end-effector (right) and each of 163 scenes consisted of up to 10 CLEVR-like object primitives, captured on grey background (left). The object configurations were copied from randomly selected synthetic images.

Each visual input image is resized to 224×224 and a 28×28 convolutional feature map is extracted using a Resnet-101 network [18], pretrained on ImageNet [19]. The output is taken from a 512-channel layer. The final visual feature x_{vis} for each image is thus a $28 \times 28 \times 512$ tensor [9]. This tensor is then provided as input for neural modules (reasoning) described later.

The reasoning subsystem is consisting of two core elements: a group of 7 co-attentive neural modules designed to solve specialized subtasks, and a layout policy assigning each question a specific layout consisting of a chain of the neural modules. Each of the modules (*find, relocate, and, or, describe* or *compare*) can receive on input an attention map and both visual and textual feature vectors obtained from the image and question. In addition, modules *and* and *or* receive two image attention maps to provide their union (*and*) or intersection (*or*). On the output, there are either attentions (which are used as inputs for a different module) or embeddings for possible answers such as a name of a shape, color, a number or a binary answer *yes*/*no*. In our robot implementation, the answer with the highest score is extracted and presented by the robot with a one-word answer.

IV. Experiments and Evaluation

For detailed evaluation of the TbD architecture, we have used four different kinds of datasets. The first one is the original CLEVR dataset, which was used as a benchmark. Next we created an adapted version of CLEVR, which we call CLEVR COUNT. This is an adapted version of CLEVR in which we implement our custom set of 12 questions designed

to measure logical consistency among answers. Secondly, we used a CLEVR-like dataset generated using OpenAI Gym and Unity render, in which the scene is depicted from 9 different viewports. Lastly, we also collected real-world images of a CLEVR-like scenes using a robotic manipulator with attached camera.

A. CLEVR dataset

CLEVR is a benchmark synthetic dataset proposed by [12]. It consists of Blender-rendered images of object primitives possessing properties with restricted variability (8 colors, 2 sizes, 2 materials and 3 shapes). Each image is provided with a ground truth description, complex questions targeted at object relations or attributes and correct answers in natural language. There are also functional program representations mapping the chain of logical operations which need to be executed in order to answer the question (i.e. *filter color, filter shape, relate* etc.). The TbD model was originally trained on standard version of CLEVR dataset which contains 70,000 images with 3 – 10 objects and 699,989 questions. On the other hand, the standard test set adopted for N2NMN and other compositional VQA architectures [12] evaluation consists of 15,000 images and 14,998 questions. It means there is one question per image. Moreover, the questions are divided into 14 program categories (count, size, color etc.), so there are approx. 1000 questions per category. We are unable to test the reliability of the architecture in this scenario. This method of evaluation does not allow to check whether the architecture understands all the aspects of the scene as there is only one question per image and the consistency of answers is not addressed.

B. COUNT dataset

Therefore, we propose a novel methodology that coins metrics for logical consistency of answers. It states that total amount has to be sum of its parts or mathematically $2 = 1+1$ The most suitable operation for this evaluation is the counting operation in the dataset. The counting operation should be applied for all attributes in the dataset as you can ask ”How many *objects of specific attribute* are there?”. This evaluation allows us to evaluate the system accuracy for specific attributes complementary to previous evaluation, but also to evaluate the consistency for a specific category (size, color etc.). We also present the overall accuracy that stands for full understanding of the scene and requires to correctly answer all 16 questions.

For the training and testing, we generated 70 000 and 10 000 images respectively, containing 3-10 objects per scene similar to original CLEVR dataset. We created 12 questions to each image, where first question is focused on the total amount of objects in the scene, and the 11 other questions were focused on the number of objects for each attribute (i.e. 3 questions for shapes and 8 for colors). We did not present size and material in the dataset as it is difficult to create such objects in the testing set with real objects (see bellow).

We then train and test the TbD model with 12 questions per 70 000 or 10 000 of the images respectively, standing for 840 000 training and 120 000 testing questions. The consistency of answers is calculated for specific feature (e.g. if there are 5 objects altogether, the sum of all predicted cubes, spheres and cylinders should be similar to ground truth and has to be equal to 5). The consistency for all objects stands for correct answers to all of the 12 questions per image.

C. GYM dataset

At the next step, we tested the robustness of TbD with respect to the changes of the camera viewport. We developed a virtual environment in OpenAI Gym [20] with Unity graphics rendering, that mimics the generation of original CLEVR dataset with both images and questions. This environment has the option to look at the scene from different perspectives. We prepared 10 000 unique scenes similar to CLEVR and rendered them from 9 specific viewports standing for 90 000 training images and 1 080 000 training questions (12 questions per image). The variable-viewport test set consisted of 2 000 unique scenes, 18 000 images and 216 000 questions. The evaluation and consistency calculation is similar as in the previous dataset. Moreover, we can test the robustness of VQA combining fixed viewport training and testing (1VTr/1VTe), fixed viewport training and variable (9) viewport testing (1VTr/9VTe), variable viewport training and testing (9VTr/(VTe) and variable viewport training and fixed viewport testing (9VTr/1VTe).

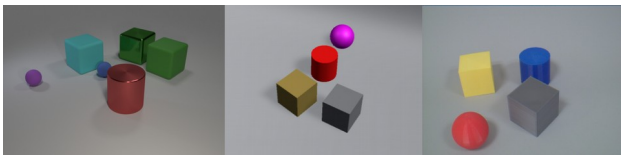

Fig. 2. Comparison between the datasets used in our study. The original CLEVR dataset (*left*) with fixed viewport, a sample from our nine viewport dataset generated using MuJoCo OpenAI environment and Unity render (*center*) and an example of real-world scenes collected by IIWA Kuka LBR 7 robotic manipulator (*right*).

D. ROBOT dataset

As the development of a counterpart of these synthetic datasets in real environment is time-demanding (with the amount of scenes and manual annotation), we develop only a small test set, again with variable viewports, consisting of 165 scenes, 1 485 images and 17 820 questions. Each image contained 3-10 CLEVR-like objects (varying in 3 shapes and and 8 colors) in configurations reconstructed from randomly selected synthetic CLEVR images. We then again tested the model on the same set of 12 questions, but used the real-world scenes for visual input. The evaluation and consistency calculation is the same as for the synthetic dataset.

For collection of the real-world testing dataset, we used a 7 DoF industrial robotic manipulator Kuka IIWA LBR 7. As a camera sensor, we used a 5 MPx with 25 FPS Basler Dart camera mounted on the end-effector. For robot control and image acquisition, we used open-source middle-ware Robotic operating system (ROS). Each scene in the testing dataset was collected from 9 different viewports. Three of the viewports were oriented at the scene from the angle of 30 (closest to the

original CLEVR dataset), another three from 40 and the last three from 50 angle. In each angle, there were three viewports placed 20cm from each other, all of them looking at the centre of the scene. For illustration, see Fig. 3.

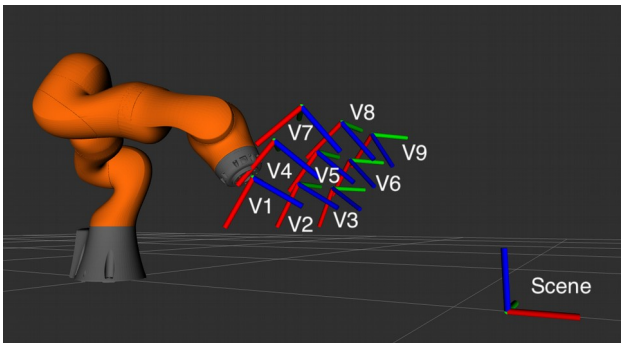

Fig. 3. Viewport distribution in robotic CLEVR dataset. Robot captures the images from 9 different viewports (V) while pointing to the exact point in environment, that represents the center of the scene with CLEVR objects

E. Evaluation and consistency calculation

Both TbD architectures trained on original CLEVR and COUNT dataset ans tested on COUNT, GYM and ROBOT testset. All testset included 12 questions per image, that are suitable for counting capabilities of architecture and we should also calculate consistency of the answers. There are 12 questions asking for the number of objects of a given parameter (1 question for overall object number, 3 questions for shapes and 8 for color). We then calculated the accuracy of answers for each type of question and the consistency between answers for each image. Consistency is calculated as shown in equation 1.

$$\psi = \frac{\sum_{v=1}^{n_v} \bigvee_{i=1}^{n_p} P_{o_i} P_{c_i} P_{s_i}}{N} 100 \quad (1)$$

Consistency (ψ) equation. Given a number of viewports (v) and number of properties (p) such as object counts (P_o), object color (P_c) and object shape (P_s) it is possible to compute accuracy of inference. Over these variables we compute a logical summation. By division of number of all scenes for all viewports (N) we get a ratio of accuracy. We report a percentage of accuracy.

For the variable viewport testsets (GYM and ROBOT) we also estimate the best viewport (Ψ) based on accuracy a consistency. This estimation search for the highest consistency and accuracy values among viewports. It is defined as described in equation 2.

$$\Psi = \max \frac{n_{correct} - n_{incorrect}}{n_{correct}} + \psi \quad (2)$$

Where we combine overall consistency (ψ) with overall accuracy expressed as ratio of correct and incorrect sums of properties (P_o, P_c or P_s) in different viewports (v). Then we find a maximum value, which defines optimal viewport, e.g. viewport with best accuracy, respective consistency.

The viewport analysis prior the best viewport selection requires accuracy and consistency calculation for each viewport separately for variable viewport testsets (GYM and ROBOT) compared to fixed viewport testset (COUNT) The best viewport is then reported in the Tab. II as an index after the GYM and ROBOT testset.

V. Results

A. CLEVR dataset training

The detail analysis of TbD trained on original CLEVR dataset revealed that the architecture in not able to correctly recognize total number of objects in the scene. Although overall counting capability reported in Mascharka et al. [9] is 97.6 %, there is only 64.5 % accuracy to count all objects. The architecture is able to count both shapes (94.9 %) and colors (99.6 %). The high error in counting objects is then reflected in consistency metric that is only 56 % for all objects and vary between 57 % and 63 % for color and shape consistency. This architecture has surprisingly higher accuracy on GYM variable viewport testset, where the counting accuracy on objects increases to 69 % and 96 % on shapes. There is also increase in shapes consistency to 63 % . The other results stand for decrease in accuracy and consistency and overall consistency is only 36 % at the best viewport and and only 14 % for the worst viewport (7) in GYM testset.

The results for ROBOT testset is even worse as there is only 17 % overall consistency for the best (3) viewport and 0 % for the worst (8) viewport. The accuracy reaches only 51 % for all objects. We can see very poor capability to transfer the trained architecture to a novel environment.

B. COUNT dataset training

The new method of training described in previous chapter stands for the great improvement both in accuracy and consistency. When tested on fixed viewport set (COUNT) we observe very high accuracy (100 % for colors, 98 % for shapes and 100 % for shapes) and also consistency improved twice, namely 97 % for all objects, 98 % for shapes and 99 % for colors. The architecture is able to count almost perfectly all the aspects of the presented scene. When we tested this architecture on synthetic dataset with variable viewport (GYM), we reached again great accuracy (97 % for all objects, 99 % for shapes and 98 % for colors). The consistency in best viewport (3) was three times better compared to CLEVR dataset, namely 88 % for objects, 96 % for shapes and 89 % for colors. The variability between worst and best viewport accuracy (69 -97 %) and consistency (52 - 88 %) allows us to comfortably identify best viewport. The detail visualization of error in specific viewports are in Fig.4 As we are able to successfully idetify best viewport based on consistency and accuracy in synthetic dataset (GYM) it is desirable to apply this strategy to the real world dataset with variable viewports (ROBOT). Also at this stage the novel method outperformed original CLEVR dataset. There is a very promising accuracy in viewport 3 on objects (86 %), shapes (90 %) and colors (95 %). The consistency dropped significantly compared to synthetic dataset, but it is still better than the consistency of original CLEVR dataset even on synthetic dataset with fixed viewport). There is 55 % overall consistency, 72 % for shapes and 64 % for colors.

Train	Test	Count objects	Count shapes	Count color
CLEVR	COUNT	64.5 (56.1)	94.9 (57.1)	99.6 (62.9)
CLEVR	GYM(3)	69.5 (35.6)	95.8 (63.1)	92.2 (37.6)
CLEVR	ROBOT(3)	51.2 (17.2)	84.5 (41.1)	87.3 (19.0)
COUNT	COUNT	**99.6 (97.4)**	**98.4 (97.5)**	**100 (99.4)**
COUNT	GYM(3)	**97.0 (87.8)**	**99.1 (95.6)**	**98.4 (88.9)**
COUNT	ROBOT(3)	**85.9 (55.2)**	**90.0 (71.8)**	**95.2 (64.4)**

TABLE II
COUNTING CAPABILITY OF TBD MODIFICATIONS . COLUMNS STANDS FOR COUNTING PERFORMANCE ON SPECIFIC FEATURES (SHAPES , COLORS) AND COLUMN FOR PARTICULAR COMBINATION OF TRAINING AND TESTING SET

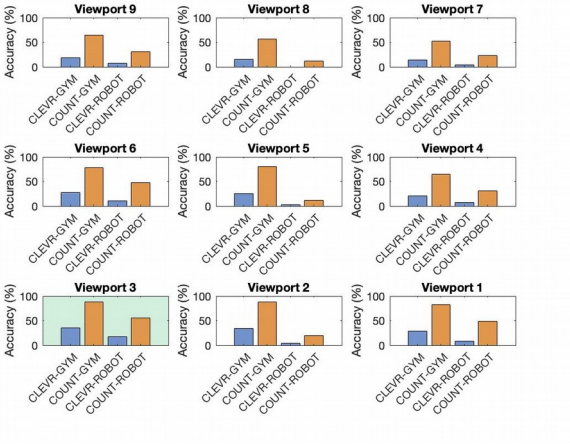

Fig. 4. Consistency overview for each viewport in CLEVR, COUNT and ROBOT dataset. Best results is highlighted in green background (viewport 3). It this particular viewport has all datasets best performance in consistency.

The variability of the results in specific viewports (see Fig. 4) spanning from 25 - 86 % in accuracy and 12 - 55 % is comfortable for the best viewport selection.

VI. CONCLUSION

Due to the rapid progress of the recent compositional models for VQA, it might seem that the task has been solved. Although the original evaluation of the results from [9], [7] revealed accuracy similar to or higher than human skills, our previous analysis [8] as well as the results in this paper, both based on a different methodology, uncovered weaknesses of the SOTA methods.

The first problem is that these models cannot provide a consistent natural language description of the observed scene. Here we argue that one possible cause might be the way these models are trained - the original CLEVR dataset contains questions which do not cover the whole feature space of a given scene. We have proposed a new training method with custom set of questions (the COUNT dataset) which demand counting of objects with a specific feature (e.g. "How many red objects are there? How many blue? How many green?"). During evaluation, we then consider the answers as consistent if the sum of objects within each category equals to the overall number of objects.

We show that retraining the TbD model with our method increases its overall consistency by 35% on the CLEVR images, by 28% on our GYM multi-viewport synthetic dataset and by 34% on our multi-viewport real world images (ROBOT dataset).

Second problem, which stands in the way of practical implementation of these models, is their dependency on the fixed camera viewport with which they were trained. To address this issue, we have created our own synthetic dataset using a Gym virtual environment, as well as real-world robotic dataset, both with 9 different viewports for each scene. Our post hoc evaluation of accuracy for each viewport enabled us to select the optimal camera angle with highest consistency - a method which could maximize the model performance on real world data without precise reconstruction of the training conditions. The subject of our future research is retraining the model on the multi-viewport GYM dataset to obtain a viewport-invariant architecture. Our preliminary results have indeed shown that such model has higher consistency when testing on the same dataset. However, further elaboration and more data is needed before making conclusions.

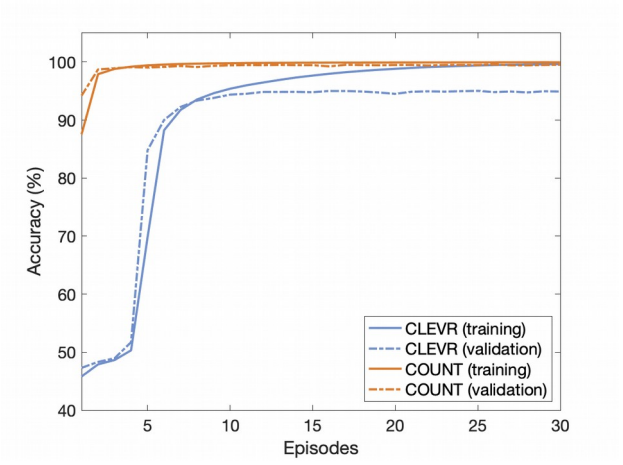

Fig. 5. Accuracy during training for CLEVR and COUNT dataset.

ACKNOWLEDGMENT

This work was supported by the Student Grant Competition CTU, the ZETA program of Technological Agency CR (project Imitation learning supported by language for industrial robotics, TJ01000470) and Robotics for Industry 4.0 (reg. no. $CZ.02.1.01/0.0/0.0/15 \_003/0000470$). This works is also supported by CTU student grant agency (SGS18/205/OHK3/3T/37) and Inafym project $CZ.02.1.01/0.0/0.0/16.019/$.

REFERENCES

[1] M. Vavrečka and I. Farkaš, "A multimodal connectionist architecture for unsupervised grounding of spatial language," *Cognitive Computation*, vol. 6, no. 1, pp. 101–112, 2014.
[2] K. Štepánová, F. B. Klein, A. Cangelosi, and M. Vavrecka, "Mapping language to vision in a real-world robotic scenario," *IEEE Transactions on Cognitive and Developmental Systems*, 2018.
[3] K. Štěpánová, "Hierarchical probabilistic model of language acquisition," 2016.
[4] M. Vavrečka, I. Farkaš, and L. Lhotská, "Bio-inspired model of spatial cognition," in *International Conference on Neural Information Processing*. Springer, 2011, pp. 443–450.
[5] M. Vavrecka and I. Farkaš, "Unsupervised grounding of spatial relations," in *Proceedings of European Conference on Cognitive Science, Sofia, Bulgaria*, 2011.

[6] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Parikh, and D. Batra, "Vqa: Visual question answering," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 4–31, 2017.

[7] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, "Learning to reason: End-to-end module networks for visual question answering," *CoRR, abs/1704.05526*, vol. 3, 2017.

[8] G. Sejnova, M. Tesar, and M. Vavrecka, "Compositional models for vqa: Can neural module networks really count?" *Procedia computer science*, vol. 145, pp. 481–487, 2018.

[9] D. Mascharka, P. Tran, R. Soklaski, and A. Majumdar, "Transparency by design: Closing the gap between performance and interpretability in visual reasoning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4942–4950.

[10] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[11] J. Suarez, J. Johnson, and F.-F. Li, "Ddrprog: A clevr differentiable dynamic reasoning programmer," *arXiv preprint arXiv:1803.11361*, 2018.

[12] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 1988–1997.

[13] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 39–48.

[14] ——, "Learning to compose neural networks for question answering," *arXiv preprint arXiv:1601.01705*, 2016.

[15] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick, "Inferring and executing programs for visual reasoning." in *ICCV*, 2017, pp. 3008–3017.

[16] S. Reed and N. De Freitas, "Neural programmer-interpreters," *arXiv preprint arXiv:1511.06279*, 2015.

[17] S. Cho, W.-H. Lee, and J.-H. Kim, "Implementation of human-robot vqa interaction system with dynamic memory networks," in *Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 495–500.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 248–255.

[20] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.