Compositional and systematic generalization in the state-of-the-art Language and Vision systems

Sungjun Han
Language and Vision
st175409@stud.uni-stuttgart.de

1. Introduction

Compositionality and systematicity have been identified to be key principles in human cognition. Researchers have long tried to incorporate the two principles in building intelligent machines. This led to a debate in cognitive science in the 80s and 90s on whether deep learning architectures are able to exhibit these principles. The recent breakthrough in deep learning in various domains has re-ignited the interest of researchers in answering this question. One of the domains that has gained a lot of attention due to the deep learning breakthrough is language and vision (L&V). Various models have shown human-level performance on many L&V benchmarks. Out of these benchmark includes a visual question answering dataset called CLEVR which was explicitly designed to test compositional reasoning in grounded language understanding. Despite their success, it is an open question whether the state-of-the-art model rely on compositionality and systematicity for their amazing generalization capability.

Compositionality refers to the ability to reason through combining semantics of known elements [5][9]. This is an integral part of human cognitive processing and it allows us to easily understand novel sentences containing the new word that we have just learned. For example, once one understands the meaning of the verb "dax", he can easily understand the sentences "I can dax twice" or "I will dax tomorrow". Compositional reasoning is not limited to linguistic concepts as natural language is heavily grounded in vision [25]. We make use of compositional generalization to reason with objects of novel attribute combinations and novel situations [10]. For instance, no human would not be able to identify a blue banana even though one would have likely have never seen one before.

Another important property of human cognition is systematicity. Systematicity refers to our sensitivity to syntactic structure when building and reasoning with linguistic concepts [5][9]. This enables us to be able to understand a set of related sentences once we have understood the semantics of a particular sentence. This is illustrated by the famous example "John likes Mary" from Fodor and Pylyshyn [9]. Once one understands this sentence, he is able to immediately understand "Mary likes John" as the underlying syntactic structures of the both sentences are identical. Systematicity is also not limited to purely linguistic concepts and it is important for reasoning with heavily visually grounded concepts. When given a concept about a particular spatial relation between objects, we can understand and generalize after seeing only a small subset of the possible pairs of objects of such relations.

In this article, we investigate the compositional and systematic generalization capabilities of various visual question answering architectures that showed human-level performance on the CLEVR dataset and other related datasets. We briefly review the historical connectionist and classicist debate to put the discussion of compositionality and systematicity in the context of artificial intelligence. After introducing the CLEVR dataset, we review these models and their approaches through a perspective of the historical debate. Then we present experiment evidence showing that the CLEVR models fail to compositionally and systematically generalize in truly novel contexts. Finally, we will conclude by discussing the significance of this result in applying the L&V models to real-world applications and the potential directions of research towards the models that can exhibit compositionality and systematicity.

2. Classicist vs Connectionist debate

Compositionality and systematicity are the two defining principles of the Classical theory of cognition. Classicists view cognition as a Turing machine that manipulates atomic or complex symbols through an algebraic rule system [9]. This symbolic rule-based system is able to inherently satisfy the two principles. For instance, the rule “P&Q” induces a constituent structure where “P” and “Q” are the constituents of the complex symbol “P&Q”. The semantics added by the structure of the rule is the same for all cases where the rule is applicable. For example for both “A&B" and “(AB)&(CD)”, the meaning added by the the rule “P&Q” is the same and does not depend on whether the constituents are atomic or complex. Hence the system is inherently sensitive to the structure. However, the rule-based symbolic system came under a criticism for its inability to account for other characteristics of human cognition such as the vast exception handling abilities and the large-scale pattern recognition processes that underlies perception [9][22].

Connectionism, known as Deep Learning in the modern era, focused on these shortcomings and proposed an alternative theory of cognition [24]. It operates through association and statistical learning, not rule-based symbol processing. This is achieved by representing the concepts as vectors through a distributed representation of artificial neurons. Thus, the central process of cognition for connectionists is not inference and linguistic reasoning, but perception. Thus the rule-based behaviours of our cognitive processes are considered to be the exception rather than the law.

This led to a long debate about whether connectionist architectures are able to exhibit the two principles of cognition [9][17][22]. Classicists argued that since purely connectionist architectures cannot exhibit the two principles, connectionism should be geared towards implementing classical architectures, not as a theory of cognition. In response, connectionists argued that their distributed representations do exhibit a constituent structure and connectionist networks can be constructed with an adequate prior to exhibit structure sensitive processes. Hence connectionists looked to find the right prior where the rule-based behaviours would arise as an emergent phenomenon from a purely connectionist architecture [21]. Since connectionism was in its infancy at the height of this debate, the debate was inconclusive. With the recent successes of Deep learning, this debate had attracted a renewed attention. However, recent works have shown that deep learning architectures’ powerful generalization capability rely on discovering statistical regularities rather than compositionality and systematicity [5][6][8][15][16]. For example, they struggle to understand the novel phrase “jump around right" when only trained with “jump” and "around right”.

3. CLEVR

There has been a growing interest in studying compositional and systematic reasoning in grounded language understanding with L&V tasks. The two principles are important to extend the L&V models to the complicated real world situations because it needs to constantly reason under various objects with an unlikely attribute combination and be able to learn visually grounded concepts from limited experience. This was popularized by the introduction the synthetic visual question answering [28] (VQA) dataset called CLEVR [14]. CLEVR was specifically designed to test compositional reasoning in grounded language understanding along with other visual reasoning skills. Also, its synthetic nature enabled researchers to study visual question answering systems unobscured by easily exploitable question-

conditional biases which were often associated with other VQA datasets composed of real-world images [14].

The CLEVR dataset consists of 100,000 images and 853,554 questions which are randomly split into the train and test set to form the standard dataset called “CLEVR.v01". Each image is a 3D scene of multiple objects with different colours, textures and shapes in various spatial relationships with each other. The objects can be in one of the three shapes: cube, cylinder and sphere. The question could ask about the existence of an object satisfying certain attributes, compare the attributes between different objects, or count the objects with the specified attributes. Each question was constructed using an associated functional program which was executed on the scene graph of the corresponding scene to generate an answer. The underlying functional program of each question is known as the *ground truth*

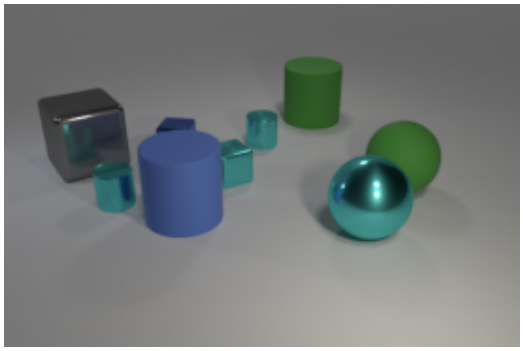

Figure 1 Scene image of CLEVR.

program (GTP). The functional program consists of basic functions that count, confirm, compare, filter, or verify the spatial relations of the objects with the specified attributes. The basic functions are combined into a hierarchical structure to form the GTPs. One way to be successful in CLEVR is by discovering the underlying GTP that was used to generate the question through analyzing syntactic structure of the question. The program can inform the model of the identities of the primary objects and attribute classes which make up the scene and how these primary elements need to be combined to reach the correct answer.

In order to test compositionality, the authors of CLEVR proposed a separate “condition-split” called “CoGenT” [14] which includes two conditions A and B. In condition A, half of the available colours are assigned for cubes while the other half is assigned to cylinders. All colours are allowed for spheres. In condition B, the assignments are flipped between the cubes and the cylinders. The task is to train only on condition A and generalize to condition B. For example in the Figure 1, the model needs to be able to recognize the green shiny sphere by composing the known concepts of green, shininess and spherical shape which were seen separately but never together in the given combination. A model capable of reasoning compositionally would be able to learn the concept of colour from the spheres and recognize the cylinders in a novel colour.

4. The Hard and the Soft

There are two categories of approaches to CLEVR which we will refer to them as the *hard* and the *soft*. Compared to the baseline models that performed poorly on CLEVR.v01, models from both approaches have matched or even surpassed the human performance on the dataset. The hard models extract the question's syntactic structure to initialize a network to generate an answer. The initialization is done through selecting from a pool of trainable neural modules according to the syntactic structure. The soft models are fully differentiable general systems where the information from the linguistic modality is represented in a distributed representation. The extracted information is used to initialize the parameters of the answer generating network. Thus, both approaches make use of the controller-generator architecture. The controller processes the linguistic modality to make a decision on how the generator should be configured. Then the generator is used to predict the answer from the visual modality.

The difference between the two approaches is an echo of the historical debate between connectionism and classicism. The hard can be seen as an attempt to implement the symbolic rule system with differentiable neural systems as classicists have argued. On the other hand, the soft does not attempt to implement the rule-based system, but uses a fully-differentiable system with a structure sensitive prior just as connectionists have argued.

4.1 The Hard

The hard models are commonly known as neural modular networks [1] (NMN). The controller of NMN decides on the number and the types of modules to use and how the selected modules should be connected. Upon this decision, the modules are collected to form a hierarchical computational graph in generating the answer for the question. Since the compositional reasoning is naturally encoded in the structure of the computational graph, these models are able to easily generalize to objects with novel attribute combinations. They are also capable of systematic generalization as the information processing flow is determined by the structure of the generated computational graph. However, these generalizations are only possible when the generated computational graph is successfully inferred. Unfortunately, learning to successfully infer such graphs from scratch is usually very difficult. Hence, these models rely on the prior knowledge of the correct computational graphs underlying the question, the GTP, to pretrain the controller. The pretrained controller can then be further trained without the extra-supervision signal in an end-to-end fashion using reinforcement learning. Thus, the focus of the hard approach has been on reducing the number of GTPs required for the pretraining. We introduce two notable CLEVR models from this approach: Tensor-NMN [13], and NS-VQA [27].

Tensor-NMN [13] uses an LSTM encoder [11] to predict the prefix traversal of the CLEVR functional program tree through greedy-decoding. Each function-value pair is assigned to a unique neural module which are all identically parametrized. The generator interprets the controller output into a valid program augmenting with a dummy scene variable when it is too short and truncating when it is too long. Then the corresponding neural network of the generated program is constructed from the neural modules. The constructed network processes the visual features from the hidden layer of a pretrained image classification model to generate the answer. The whole system can be trained in an end-to-end manner with the policy gradient method called REINFORCE [26]. However, this requires the controller to be

pretrained with GTPs as randomly initialized models lead to very sparse rewards [13]. With around 18,000 GTPs, Tensor-NMN is able to match the human-baseline on CLEVR.v01.

NS-VQA [27] takes a step further on the symbolic approach by using a purely symbolic generator written in Python programming language. The modules are not differentiable neural blocks but pre-programmed functions that operate on a structured tabular data. Hence the controller of NS-VQA has an additional scene parser component which parses the given image into a tabular representation by detecting salient objects along with their attributes. The scene parser first detects salients objects in the image through an object-detection model detected object's attributes such as the colour, the shape, and the pose are classified using an image classification model. The controller also has the usual question parser that parses the question into a functional program. It is implemented with a bi-directional LSTM seq2seq [23] model with attention [4]. Because the generator does not need to be learned as they are pre-programmed from the prior knowledge of the task, the reward signal for the controller is not obscured by the mistakes from the neural modular network during the end-to-end training. Thus, the question parser is able to do a better credit assignment from the rewards to quickly find the correct program-generation policy. Hence, it only needs a small number of GTPs for pretraining. NS-VQA is able to achieve a near-perfect result on CLEVR.v01 with only 270 GTPs.

4.2 The Soft

The soft models are end-to-end differentiable systems that can be trained without requiring any task-specific prior knowledge. This allows them to be easily applied to any VQA tasks. The soft models do not rely on explicitly modelling the symbolic syntactic structure of the question. Instead they focus on finding the right structural prior. The right prior can constrain the parameters of the model to converge to the region that is compositional and systematic. We introduce three notable CLEVR models from this approach: FiLM [18], RelNet [20], and MAC [12].

FiLM [18] uses conditional batch normalization (CBN) to control the CNN network which processes the visual input. In CBN, the gain and the shift parameter of batch normalization can become conditional on some input. The conditional input is generated from a GRU [7] encoder that encodes the question into an embedding. The generator CNN is composed of multiple layers of modules composed of two convolutional layers followed by the CBN layer. Due to the non-linear activation succeeding the CBN layer, the controller is able to guide the generator through modifying its activation distributions.

RelNet [20] encodes a strong relational reasoning prior by making the network to consider all possible pairs of feature vectors. The feature vectors are the column vectors of a hidden layer from the CNN that is used to encode the image. All feature column vector pairs are concatenated with the question embedding from the LSTM-encoder to allow for the control from the linguistic modality. All pairs go through a non-linear transformation and the resulting vectors are average-pooled for the prediction.

MAC [12] uses a specially designed information processing cell called Memory-Attention-Composition. The MAC network forms and updates working memory state of the linguistic modality and the visual modality separately using attention mechanism. The model first prepares the multimodal input by generating word embeddings and a contextual question embedding using a bi-LSTM encoder for the question and a memory tensor using a pre-trained image recognition model for the image. Then, the

	#GTP	CLEVR.v01	CoGenT (A/B)	CoGenT-Finetune on B (A/B)	CoGenS (A/B)
Tensor-NMN	700000	96.9	-	-	-
Tensor-NMN (2)	18000	95.4	96.6/73.7	76.1/92.7	-
NS-VQA	270	99.8	99.8/63.9	64.9/98.9	-
FiLM	0	97.6	98.3/75.6	80.8/96.9	-
MAC	0	98.9	-	-	89.5/65.3
RelNet	0	95.5	-	-	-
Human-baseline	-	92.6	-	-	-

Table 1 Question answering accuracy (higher is better) of the CLEVR models on CLEVR.v01, CoGenT and CoGenS. The number of ground-truth-programs needed for pretraining is presented under "#GTP". Tensor-NMN(2) is the same as Tensor-NMN but trained with a smaller number of GTPs.

prepared representations are processed through a fixed number of times steps through the controller and the generator (memory). The controller at each time step accepts the previous control state and the question embedding to decide on the new attention weights. The attention weights are used to combine the word embeddings to form a new control state. This constrains the controller to model the syntactic structure as it can build representations through composing the fixed semantic components. The generator network is composed of the *read* and the *write* component. The read component takes the previous memory state and generates a new candidate state through attentioning on the memory tensor. The attention weights can be modified by the control state and this is the only way the linguistic modality can guide the visual modality. Hence, the two modalities can only interact through a probability distribution. The write component takes the candidate memory state and the previous state and generates a new state through a linear transformation. Both the question and memory state are processed by the output module to generate the answer. MAC implements two strong structural priors: the constraint that limits the interactions between the two modalities through a probability distribution and the syntax-semantic split constraint of the controller and the generator. This design decision was crucial in the model's success on the CLEVR dataset. Also, the model was able to train much more efficiently requiring far less number of examples than other models such as FiLM.

5. Compositional generalization

Models from both approaches have shown outstanding results on CLEVR.v01. The results are summarized in Table 1. However, on CoGenT which tests compositional generalization, models from both approaches struggled to generalize to the new condition without any fine-tuning examples from condition B. The central cause of the failure for the models seems to be that there is no incentive for the models to learn that "green" in "green sphere" refers to the colour rather than the shape because it has never seen the use of "green" in any other contexts. Thus, better insight can be gained from analyzing how the models learn to generalize after seeing a few examples from condition B - seeing the evidence that "green" is a colour not a shape.

	k=1	k=2	k=4	k=8	k=18	k=35
Tree-NMN	<1	-	-	-	-	-
Chain-NMN	46.61	-	-	-	11.82	<1
Chain-S-NMN	30.09	20.93	~2	<1	<1	-
MAC	13.67	~8	~2	~4	<1	<1
FiLM	34.73	19.80	~10	~3	~2	<1
RelNet	31.05	15.71	27.47	50.22	-	-

Table 2 Question answering accuracy error (lower is better) of the soft models, Tree-NMN, and Chain-NMN on SQOOP [2] for different k. The results were present in a graph in the original paper, thus we present approximate values for the results for the splits where the exact values were not provided.

We analyze the results for the models where the results on CoGenT are available: FiLM, Tensor-NMN and NS-VQA. For FiLM, it is unlikely that it is truly reasoning compositionally as the performance of the network gradually improved as the number of samples from condition B increased. If the model did exhibit compositionality, it would likely have made a drastic jump in performance at a certain number of samples from condition B where it would have realized that "green" is to be interpreted as a colour not as a shape. Similarly, Tensor-NMN also exhibited a gradual improvement as the number of fine-tuning examples increased. This is because the controller was not able to generate the correct program for the new condition as it was not provided with GTPs from condition B. It had to infer the correct computational graph from the reward signals alone. This seems to suggest that finding GTPs without a proper prior is a difficult task. Hence, the hard models can only effectively generalize compositionally if the syntactic structure underlying the new situation has already been experienced. NS-VQA performed much worse than the other two models. The scene parser was identified to be the problem. The object detector of the scene parser learned to detect objects based on colour and ended up incorrectly detecting the objects in novel colours. When the detection model was trained with examples from both conditions, it was able to perform near perfectly on both conditions without requiring any fine-tuning. Hence, the model failed to compositionally generalize as the problem was only transferred from responsibility of the question parser to the scene parser. Even though it failed to generalize compositionally, NS-VQA's perfect score with the fixed scene parser confirms the potential of the hard models of their compositional generalization capacity.

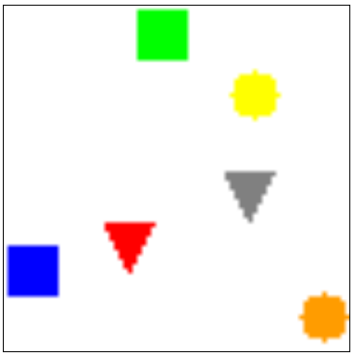

Figure 2 Scene image from CoGenS condition A.

MAC did not report on its performance on CoGenT. Hence, we created an adapted version of Sort-of-CLEVR [20] to mimic the condition of CoGenT to test its compositional generalization capability. We call this new dataset "CoGenS". CoGenS consists of 9800 2D images with six differently coloured shapes. The shapes can a circle, rectangle or triangle (see Figure 2). Each image has two associated questions. Questions can be a yes-or-no location question, counting question, or binary relation question. We refer the readers to the Supplementary Material section for more detail on the dataset. We created two conditions, A and B, similar to CoGenT. In condition A, rectangles were assigned red, green and blue and

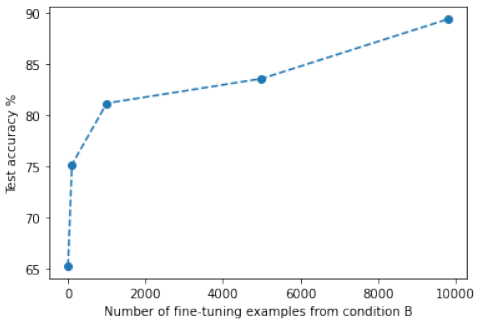

Figure 3 MAC's question answering accuracy (higher the better) on CoGenS condition B with different numbers of fine-tuning examples.

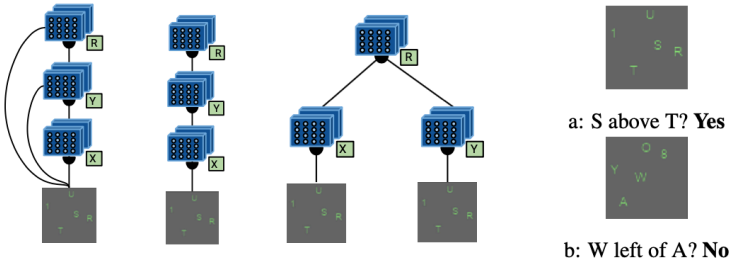

Figure 4 The architectures of Chain-S-NMN, Chain-NMN, Tree-NMN (from the left). Two sample SQOOP questions is presented on the right. Figure was taken from Bahdanau and colleagues [2].

circles were assigned orange, grey, and yellow. Triangles were allowed to be in any of the six colours. The colour assignments were flipped in condition B between the rectangles and the circles.

We trained MAC on condition A and tested its performance on condition B. The results can be found in Table 1. The model was not able to generalize in the novel condition likely due to the same reason as noted above. Hence, we tracked how the model improved as the number of fine-tuning examples from condition B increased. Similar to the results from FiLM in CoGenT, MAC showed a gradual improvement rather than a jump (see Figure 3). This again suggests that MAC relies on extracting statistical regularities rather than compositional reasoning for the generalization.

6. Systematic generalization

Another question which we are interested in is whether the soft models are capable of systematic generalization since they do not explicitly model syntactic structure like the hard models. Bahdanau and colleagues [2] proposed a simple dataset called SQOOP to investigate this question. SQOOP is a VQA task with simple yes-or-no questions of the type "X relation-type Y?" (see Figure 4). There are 36 distinct symbols and 4 relation types. Since the self-relations are prohibited, 36x4x35 unique questions are possible in the dataset. Each question is associated with a randomly generated image which always contains the two symbols from the question along with others to serve as a distractor. They propose "#k-split" where for each pair of the left-hand side symbol X and the relation type, only k number of distinct symbols are included as the right-hand side symbol Y in the training set. Hence in the k=1 split, the model needs to generalize to the other 34 symbols after seeing only one symbol at Y. The model needs understand that any symbol can be placed at Y through realizing the syntactic structure of the question "X ←Relation-type→ Y".

Bahdanau and colleagues [2] tested FiLM, MAC, and RelNet on various #k splits of SQOOP. For comparison, they also tested Tree-NMN where the neural modules are connected in a tree-structure "X ← Relation-type→ Y". The results are summarized in Table 2. None of the soft models were able to generalize after seeing only one right hand-side example while Tree-NMN was able to score near perfectly. RelNet failed to converge for all splits. FiLM showed good performance after k=8. MAC faired much better than FiLM and was able to show relatively low question accuracy error after only four examples. Considering the simplicity of the task, this seems to indicate the lack of systematicity in these models.

They also tested a chain-like linear structure "X→Y→Relation-type" (Chain-NMN) and the same structure with short-cut connections (Chain-S-NMN) using the same neural modules as Tree-NMN (see Figure 4). These models are the soft versions of Tree-NMN. See Table 2 for the summary of the results. Similar to the soft models, Chain-NMN was not able to generalize in the low k splits. Adding short-cut connections led to an improvement, but it was still not able to generalize systematically. This shows that the knowledge of the true syntactic structure is essential for systematic generalization and it cannot be approximated from a linear structure.

The authors also analyzed the ability of the NMN models to discover the correct syntactic structure without any extra-supervision signals. Tensor-NMN's controller was given the option to choose between the two options: the tree-structure or the chain-structure. When it was initialized with a bias towards the chain structure, it failed to converge for the large k splits and showed poor performance even for the easy split such as k=18. This confirms the importance of the controller pretraining for the hard models as discovering the true syntactic structure from the reward signal alone seems to be a very difficult task.

Another work from Bahdanau and colleagues [3] introduced a new test dataset called CLOSURE to asses the systematic generalization capability of the models trained on CLEVR. The scenes were generated using the same method as CLEVR. CLOSURE questions were constructed by following the same syntactic and semantic rules as CLEVR. For example, CLOSURE asks about the existence of an object through a relationship with another object that can be identified through a third object. In CLEVR such embedded referencing is not used for the existence questions. Bahdanau and colleagues [3] in the same work experimented with both the soft models such as FiLM and MAC and the hard models such as Tensor-NMN and NS-VQA on CLOSURE. The result were similar to SQOOP. The soft models were not able to generalize systematically to the new question types that they had not seen before. MAC again greatly outperformed FiLM in all question types. The hard models failed to generalize unless the supervised signals for the GTPs were given. Upon given 252 GTPs for fine-tuning, all hard models were able to score near perfectly. However, when they were left to infer from the reward signals, they performed much worse especially for the questions with novel syntactic structure. MAC was able to better generalize given the same number of fine-tuning examples for some of the question types. This again confirms that the current hard models cannot discover the true computational graph from the simple reward signals alone and relies heavily on the prior knowledge of the GTPs.

7. Discussion and Conclusion

In the context of language and vision and grounded language understanding, compositional generalization refers to the ability to generalize and reason with objects of novel attribute combinations after seeing the attributes separately in other contexts. On the other hand, systematic generalization refers to the ability to reason about the relations between the objects after seeing only a small subset of the possible relations. The two principles provide an efficient way to generalize and are thought be the central mechanism underlying human cognition. However, the analysis into the state-of-the-art CLEVR models have shown that such generalizations are still difficult for the current intelligent machines.

The soft models are the generic systems that can be readily applied to any L&V tasks without requiring any prior knowledge. However, these models do not seem to exhibit compositionality as they require a lot of data to learn to reason with new object compositions. Also, they cannot model hierarchical syntactic structure, thus fail to reason systematically. For the real-world application, this means that one needs to obtain a large amount of training data for all possible situations to make these models robust. Such collection of training data can be prohibited or very expensive. This means that the current soft models are only useful as an expert system with a very narrow space of possible applications. This limitation of the soft models motivates the need to search for a better regularization method or structural prior that can adequately constrain their search space towards the one that has compositional and systematic properties. Even though it is still an open question whether it is even possible to exhibit the two principles with a purely deep learning architecture, the superiority of the strongly prior-ed MAC over the weakly prior-ed FiLM in systematic generalization seems to show the potential of this line of research.

The hard models on the other hand rely on the prior knowledge by explicitly modelling the underlying structure of the linguistic input. This extracted syntactic structure allows them to reason compositionally and systematically, but only when they have recovered the correct structure. However, discovering the correct structure from the reward signal seems to be a difficult task for the current hard models. This makes them reliant on the availability of the extra-supervision signals. This severely limits the applicability of these models in the real-world as hand-crafting the ground-truth-program for all possible real-world scenes can be extremely difficult and time-consuming. However, if these models can learn from a very few number of ground-truth-programs, then their prior knowledge grants them to generalize effectively. This seems to be the direction that the research is headed for the hard models. Another direction could be to find a better way to do credit assignments with the reward signals so that the correct ground-truth-programs for the problem can be easily recovered without needing much prior knowledge. As seen in NS-VQA, if the controller's contribution to the reward signals can be properly isolated from the contribution from the generator, the controller can quickly adapt to the correct policy.

Finally, we believe that the both approaches can benefit from the new direction of research in deep learning that aims to learn compositional representations using part-whole relationship. One such model is CapNet[19]. CapNet routes the vector or matrix representations called capsules from the lower layer to the higher layer. The routing determines the possible existence of the whole, represented by the higher layer capsule, from its parts, the lower layer capsules. VQA systems that are based on such models could possibly be able to better generalize compositionally since these models inherently build compositional representations.

Even though the current visual question answering models cannot fully exhibit compositionality and systematicity, the growing interest from the research community on this problem is exciting. A breakthrough in this problem would be a big step closer to the century-old dream of building a truly general intelligent machine.

References

[1] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural Module Networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL http://arxiv.org/abs/1511.02799.

[2] Dzmitry Bahdanau and Shikhar Murty and Michael Noukhovitch and Thien Huu Nguyen and Harm de Vries and Aaron C. Courville. Systematic Generalization: What Is Required and Can It Be Learned? In *International Conference on Learning Representations, ICLR 2019*, 2019. URL https://openreview.net/forum?id=HkezXnA9YX.

[3] Dzmitry Bahdanau, Harm de Vries, Timothy J. O'Donnell, Shikhar Murty, Philippe Beaudoin, Yoshua Bengio, Aaron Courville. CLOSURE: Assessing Systematic Generalization of CLEVR Models. In *International Conference on Learning Representations, ICLR 2019*, 2019. URL https://openreview.net/forum?id=HkezXnA9YX.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 2015 International Conference on Learn- ing Representations*, 2015.

[5] Marco Baroni. Linguistic generalization and compositionality in modern artificial neural networks. *CoRR*. 2019.

[6] Joost Bastings, Marco Baroni, Jason Weston, Kyunghyun Cho, and Douwe Kiela. Jump to better conclusions: SCAN both left and right. In *Proceedings of the 2018 EMNLP Workshop Black- boxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 47–55, Brussels, Belgium, November 2018. Association for Computational Linguistics. URL https://www.aclweb. org/anthology/W18-5407

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Deep Learning Workshop at NIPS*. 2014.

[8] Roberto Dessi, Marco Baroni. CNNs found to jump around more skillfully than RNNs: Compositional Generalization in Seq2seq Convolutional Networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3919–3923, 2019. Association for Computational Linguistics.

[9] Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71, 1988.

[10] Charles Hockett. The origin of speech. *Scientific American*, 203:88–111, 1960.

[11] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997

[12] Drew Hudson and Christopher Manning. Compositional Attention Networks for Machine Reasoning. In *Proceedings of the 2018 International Conference on Learning Representations*, February 2018. URL https://openreview.net/forum?id=S1Euwz-Rb.

[13] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, 2017b

[14] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recog- nition (CVPR)*, December 2016. URL http://arxiv.org/abs/1612.06890. arXiv: 1612.06890.

[15] Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2018. URL http://arxiv.org/abs/1711.00350. arXiv: 1711.00350

[16] Joao Loula, Marco Baroni, and Brenden M. Lake. Rearranging the Familiar: Testing Composi- tional Generalization in Recurrent Networks. In *Proceedings of the 2018 BlackboxNLP EMNLP Workshop*, July 2018. URL https://arxiv.org/abs/1807.07545.

[17]Gary F. Marcus. Rethinking Eliminative Connectionism. *Cognitive Psychology*, 37(3):243–282, December 1998. ISSN 0010-0285. doi: 10.1006/cogp.1998.0694. URL http://www. sciencedirect.com/science/article/pii/S0010028598906946.

[18] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. In *In Proceedings of the 2017 AAAI Conference on Artificial Intelligence*, 2017. URL http://arxiv.org/abs/1709.07871.

[19][9.8] Sara Sabour, Nicholas Fross, and Geoffrey E Hinton. Dynamic routing between capsules. In *Neural Information Processing Systems (NIPS)*, 2017.

[20] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems 31*, June 2017. URL http://arxiv. org/abs/1706.01427. arXiv: 1706.01427.

[21] Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist networks. *Artificial Intelligence*, 46: 159–216, 1990.

[22] Paul Smolensky. The constituent structure of connectionist mental states: A reply to Fodor and Pylyshyn. *Southern Journal of Philosophy*, 26(Supplement):137–161, 1987.

[23] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, pp. 3104–3112, 2014.

[24] David Rumelhart, James McClelland, and PDP Research Group, editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cog- nition, Vol. 1: Foundations*. MIT Press, Cambridge, MA, 1986.

[25] Gabriella Vigliocco, Pamela Perniss, and David Vinson. Language as a multimodal phenomenon: implications for language learning, processing and evolution. 2014.

[26] Ronald Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(23), 1992.

[27] Kexin Yi and Jiajun Wu and Chuang Gan and Antonio Torralba and Pushmeet Kohli and Joshua B. Tenenbaum. Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language

Understanding *Advances in Neural Information Processing Systems 31: NeurIPS 2018*, October 2018. URL http://arxiv.org/abs/1810.02338. arXiv: 1810.02338

[28] Qi Wu and Damien Teney and Peng Wang and Chunhua Shen and Anthony R. Dick and Anton van den Hengel. Visual question answering: A survey of methods and datasets. In *arXiv preprint arXiv:1607.05910*, 2016

Supplementary Material

A. CoGenS

The images are sized 75x 75. There are 6 question types: Which shape [Colour]?, [Colour] shape left?, [Colour] shape up?, [Colour] closest shape?, [Colour] furthest shape?, [Colour] shape count?. **Which shape [Colour]?** asks for the identity of the shape of the specified colour. The answer can be 'triangle', 'circle', or 'rectangle'. **[Colour] shape left?** asks if the shape of the specified colour is located on the left half of the image. The answer can be 'yes' or 'no'. **[Colour] shape up?** asks if the shape of the specified colour is located on the top half of the image. The answer can be 'yes' or 'no. **[Colour] closest shape?** asks for the identity of the closest shape from the shape of the specified colour. The answer can be 'triangle', 'circle', or 'rectangle'.**[Colour] furthest shape?** asks for the identity of the furthest shape from the shape of the specified colour. The answer can be 'triangle', 'circle', or 'rectangle'. **[Colour] shape count?** asks for the count of the shape specified by the colour. The answer can be an integer from 1 to 6. For each randomly constructed image, we randomly sampled one question from the first three question types and another from the last three types. This was done because the first three questions are non-relational questions and the last three are relational questions. For different fine-tuning splits, we made sure that each training split had an equal number of examples. Hence all fine-tuning splits had 19800 questions with unique 9800 images. The code for generating the dataset is based on an open-sourced implementation of the Sort-Of-CLEVR dataset available at https://github.com/kimhc6028/relational-networks. We also release the modified datasets and the code at https://github.com/hansungj/Sort-Of-CLEVR-conditional-split. The code prepares the dataset as h5 files.

B. Experiment Details

We used an open-source implementation of MAC from [2]. Default hyper-parameter setting provided in the implementation was used. We trained MAC for 20000 iterations for all fine-tuning splits.We report the results from the best models according to the validation score. All the experiments were done with GPU resources of Google Colab-Pro. We also tested RelNet using the implementation from [2], but the model was not able to generalize well for the unseen examples from condition A. We left the results out as it could be due to the training details and we did not have enough time to thoroughly investigate the issue.

	#0	#100	#1000	#5000	#9800
MAC	65.3	75.2	81.2	83.6	89.4

Table 3 The question answering accuracy (higher is better) of MAC on CoGenS condition B for different numbers of fine-tuning examples.