Compositional Generalisation in Neuro-Symbolic Visual Question Answering

Author Name
Affiliation
email@example.com

Abstract

Compositional generalization is a key challenge in artificial intelligence. This paper investigates compositional generalization capabilities in multimodal mathematical reasoning problems. We introduce compositional generalization splits for CLEVR-Math for length and attribute generalization, testing both systematicity and productivity. We evaluate the common NS-VQA architecture and compare it to a neural baseline. Our results show that neither NS-VQA nor the CLIP-based baseline achieves length generalization on our task, with both models showing similar generalization patterns. For our compositional generalization split, the CLIP-based model performs better on the objects held out during training. However, both models see a drop in performance. For length generalization, we propose that explicitly learning recursive definitions can be important for compositional generalization. We argue for how knowledge of types curriculum learning can help future architectures achieve such capabilities.

1 Introduction

Compositional generalization is a key challenge for artificial intelligence, as human language and cognition are both largely compositional. It requires a system to understand the underlying characteristics (such as structure, types, grammar) of the data and to be able to recombine elements in novel ways rather than rely on its ability to memorise specific examples. A classical example from Fodor and Pylyshyn is that if a system knows the meaning of *John loves Mary*, then it should be able to generalize to the sentence *Mary loves John* without seeing such examples during training. Compositional generalisation is a key feature of classical approaches to language modeling, such as grammars. Recent work often focus on evaluating neural networks for this task, in part to understand how to close the gap between general-purpose models, such as seq2seq language models, with specialized architectures with strong compositional bias [Shaw *et al.*, 2021]. As a bridge between the two, neuro-symbolic methods have been proposed as a way to combine the strengths of neural and symbolic methods. For instance, neuro-symbolic methods have been shown to be effective for a variety of synthetic tasks of compositional nature, including VQA. Because of the strong compositional bias in neuro-symbolic methods, there is a need to understand the challenges and opportunities of combining neural and symbolic methods. In this work, we identify issues in neuro-symbolic visual question answering that could benefit from knowledge injection.

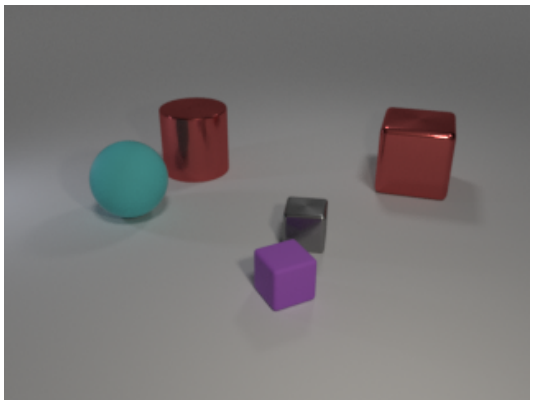

Figure 1: An example scene from the CLEVR [Johnson *et al.*, 2017] dataset. We use the mathematical reasoning subtraction-task from CLEVR-Math [Lindström and Abraham, 2022] to create compositional generalization splits for attributes and program length.

A common approach to analyze compositional generalization is to construct synthetic benchmarks where certain compositions are held out of training and only used in testing. In this paper, we describe ongoing work extending the CLEVR-Math [Lindström and Abraham, 2022] dataset with compositional generalization splits. CLEVR-Math is a recently proposed dataset for compositional generalization in the visual question answering (VQA) domain. It consists of questions about simple arithmetic operations on images of 3D scenes. Lindström and Abraham identify that NS-VQA cannot handle 2-hop questions, regardless of whether they are part of the training data or not. In order to address the complexity of CLEVR-Math, we extend the Neuro-Symbolic Visual Question Answering (NS-VQA) model to manage non-linear program executions with mutable internal representations of scenes. Finally, we compare the extended NS-VQA model

with a neural baseline.

1.1 Contributions

This paper makes the following contributions:

1. Extends CLEVR-Math with compositional generalisation splits on colors and shapes, and length generalization
2. Extending NS-VQA to manage non-linear program executions with mutable internal representations of scenes
3. A comparison between the extended NS-VQA and a neural baseline

Section 4 describes the additions to CLEVR-Math and the compositional splits we investigate. Section 3 describes how we modify the NS-VQA architecture to be able to learn the tasks in CLEVR-Math. Our results in Section **??** show how NS-VQA does not In Section 6, we argue for recursion as a key aspect of compositional generalization, and how injecting more knowledge into the training process can help models perform this type of generalization. This is in line with other arguments put forth by related work, describe in Section 2.

2 Related Work

Compositionality is an important characteristic of language understanding by humans. There has been a long-standing debate on whether connectionist architectures like neural networks are able to do compositional generalization [Fodor and Pylyshyn, 1988]. Recently, with the ubiquity of neural networks, there has been a number of benchmarks proposed to study compositional generalization like SCAN [Lake and Baroni, 2018; Loula *et al.*, 2018], COGS [Kim and Linzen, 2020] and others [Hupkes *et al.*, 2020; Ruis *et al.*, 2020]. Several of these works have shown that end-to-end neural networks are not able to compositionally generalize, especially in few-shot regimes. Kim and Linzen show that neural networks can achieve great performance on lexical generalisation, i.e. using known words in new contexts, but that they fail completely on structural generalisation. Weißenhorn *et al.* and Qiu *et al.* both show that neural models that are made aware of structure can do structural generalisation. Qiu *et al.* identify that transformer models can be augmented with synthetic data that is generated from structured methods, in their case quasi-context free grammars. Weißenhorn *et al.* uses neural network components for dependency parsing and constructing a graph representation, hence building highly structured representations of sentences. It might be possible to see all useful structures given enough data, but humans perform structural generalisation with far less data [Linzen, 2020].

2.1 Compositional Generalization benchmarks

The SCAN sequence benchmark [Lake and Baroni, 2018] provides different splits for i.i.d. generalization, length generalization, and generalization to held-out phrases (for example, the *jump* instruction has only been seen in isolation during training and at test-time the model has to be able to parse all other instructions involving *jump*). Lake and Baroni shows that end-to-end recurrent neural architectures fail to generalize to longer sentences than seen during training (length generalization) and novel actions *jump*, despite obtaining near-perfect accuracy on the random iid split. We complement these experiments with splits testing generalization to shorter sequences. To extend SCAN into the multimodal domain, Ruis *et al.* introduces gSCAN where the task is to navigate a 2D grid world using language instructions. We will now spend a brief moment looking at some specific faults of COGS and gSCAN, and what has been done to address these issues. This will inform design decisions in our own compositional generalisation benchmark.

Wu *et al.* identify a set of limitations in gSCAN which they address with ReaSCAN. They remark that the ideas gSCAN build on are powerful, but that there are some central limitations coming from specific design choices. The *first observation* is that the word order of commands does not matter for the defined tasks, where a simple bag-of-word model is sufficient to encode the original gSCAN commands. Qiu *et al.* suggest that *the remaining challenges for gSCAN may not necessarily be related to visual grounding [...]*, and propose an additional task with more complex natural language. The authors also evaluate cross-modal attention as a way for Transformer-based models to achieve strong performance on gSCAN. Their approach outperforms other methods specifically built for gSCAN, and observe that performance degrade significantly when less than 40% of the training examples are used. Sikarwar *et al.* extends the work by Qiu *et al.* with GroCoT, a multimodal transformer model achieving state-of-the-art performance on ReaSCAN. The authors complement their experiments on extended ReaSCAN and GSRR [Qiu *et al.*, 2021] with linear probing classifiers to identify what information the transformer is encoding for each object property. They conclude that their modifications to a multimodal transformer does improve compositional generalisation in the gSCAN domain. Their probing experiments show that identifying the target location is a main challenge for better solving the benchmark.

2.2 Mathematical Reasoning

Math Word Problems (MWPs) have many characteristics useful for benchmarking compositional generalization in the intersection of natural language and reasoning [Lin *et al.*, 2023]. Each element in an equation have an exact influence on the outcome. Consider the following word math problem, given by [Lindström and Abraham, 2022]:

Adam has three apples, Eve has five. Eve gives Adam all her apples. How many apples does Adam have, if he eats one?

For a system to answer this question, it must reason in multiple steps, as well as translate verbs into mathematical operations. Small changes in the text will also lead to large semantic changes, e.g. changing *eats* to *finds*. Previous work mostly explore word math problems in a text only setting, like the problem shown above, using neural networks [Robaidek *et al.*, 2018; Sundaram *et al.*, 2020; Sundaram and Khemani, 2015], and other methods [Sundaram and Abraham, 2018; Mitra and Baral, 2016]. Lan *et al.* gives an overview of the different aspects of compositional generalization that math

word problems cover, and propose ways to improve current architectures. See [Huang *et al.*, 2016] for an overview of how to construct word math problems.

2.3 CLEVR

Johnson *et al.* introduce CLEVR as synthetic dataset to benchmark compositional multimodal reasoning. The dataset is uses 3D scenes rendered using Blender, together with a templating engine to generate questions based on the structural representations of the visual scenes. With CLEVR, one can decide to generate a training set with images having only a specific combination of objects (red cubes and blue cylinders), and a test set with a different combination of objects (red cylinders and blue cubes), as done in, e.g., CLEVR-Hans [Stammer *et al.*, 2021]. This control allows us to study various aspects like compositional generalisation of systems. Since its publication in 2017, several benchmarks have built on CLEVR to study various aspects of visual question answering [Stammer *et al.*, 2021; Sampat *et al.*, 2021a; Kottur *et al.*, 2019; Liu *et al.*, 2019; Arras *et al.*, 2022; Li *et al.*, 2022; Salewski *et al.*, 2022]. CLEVRER (Collision Events for Video Representation and Reasoning) dataset [Yi *et al.*, 2020] and CLEVR-Hyp dataset [Sampat *et al.*, 2021b]. The questions on videos in CLEVRER [Yi *et al.*, 2020] requires reasoning about the state of objects after an video event, instead of after actions in text as in CLEVR-Math. CLEVR-Hyp [Sampat *et al.*, 2021b] focus on VQA where reasoning about effects of actions, whereas CLEVR-Math introduces an additional mathematical reasoning dimension to the problem. GQA is another relevant dataset, where real world images are annotated with rich scene graphs and a large set of relations and attributes, and focuses on compositionality in visual reasoning [Hudson and Manning, 2019]. However, such real world datasets does not give the same flexibility to create compositional splits.

3 Extending/adapting NS-VQA

Neuro-Symbolic Visual Question Answering (NS-VQA) was introduced by Yi *et al.* as a three-component system that disentangles reasoning from vision and language understanding. The method is evaluated on the original CLEVR dataset, showing how almost perfect accuracy is achievable with relatively small amounts of data. The three components are 1) a neural network to parse visual scenes into structured representations of the objects and attributes, 2) a neural network to parse questions into executable sequences of program elements, and 3) a program engine that executes the program sequences over the structured scene representations to compute the answers to queries. The method is used in many subsequent works, most recently Hong *et al.*, to showcase the strength of neuro-symbolic methods on visual question answering tasks.

In the recent CLEVR-Math dataset [Lindström and Abraham, 2022], NS-VQA is shown to perform well on 1-hop problems such as *Remove all blue cubes. How many objects are left?* This is consistent with the results from the original CLEVR dataset [Johnson *et al.*, 2017]. However, NS-VQA fails on 2-hop problems such as *Remove all cubes. Remove all red spheres. How many objects are left?*. In this paper, we observe that this is due to how the program templates are parsed into linear sequences of tokens used by the program executor. As an improvement, we contribute with two modifications to NS-VQA to handle the 2-hop questions.

The first modification is giving the program executor the ability to manage a mutable internal representation if a scene. In the original architecture, NS-VQA relies on two variables to manage the results of previous functions. The program parser has to produce execution trees that are stateless and only rely on feeding the output of functions as the input to the next immediate function. These two issues lead to the inability of NS-VQA to handle the 2-hop questions, As a solution we introduce a managed state, where the program executor keeps track of any changes to the scene. Every time a state-modifying operation is used, the internal representation is modified accordingly. This allows NS-VQA to chain scene-modifying operations indefinitely. The second modification is to the program parser so that it can produce stateful execution trees for operations that rely on the internal scene representation. This means that we can parse flat sequences of operators that exactly correspond the 2-hop problems, whereas before the sequences contained many nonsensical repetitions in an attempt to remain stateless.

We will now show how these modifications allow NS-VQA to solve 2- and 3-hop problems.

4 Experimental setup with compositional generalization splits for CLEVR-Math

Since the mathematical reasoning tasks are suitable for compositional generalization experiments []We use CLEVR-Math as the basis for our experiments, Our work consists of a few extensions of the templates and constraints of CLEVR-Math, as well as generating new data in compositional generalization splits. More specifically, the benchmark is constructed as following:

1. New CLEVR constraints to exclude certain combinations from a particular dataset, used to create `no-red-cubes` split.
2. Modifications to the existing 1- and 2-hop templates to fit the NS-VQA extensions, and 3-hop questions. Example seen in Figure 2.
3. Construct a new templates for compositional generalization splits.
4. Length generalization splits of between 1-, 2-, and 3-hop questions.

```
text : [
  Remove all <Z> <M> <C> <S>s, and remove all <Z2> <M2> <C2> <S2>s. How many objects are left? ,
  Remove all <Z> <M> <C> <S>s. Remove all <Z2> <M2> <C2> <S2>s. How many objects are left?
],
nodes : [
 {  type :  scene ,   inputs : [ ] },
 {  type :  filter ,  inputs : [ 0 ],  side_inputs : [  <Z> ,  <C> , <M> ,  <S>  ]},
 {  type :  subtraction_set ,  inputs : [0,1]},
 {  type :  filter ,  inputs : [ 2 ],  side_inputs : [  <Z2> ,  <C2> , <M2> ,  <S2>  ]},
 {  type :  subtraction_set ,  inputs : [2,3]},
 {  type :  count ,  inputs : [4]}
],
```

Figure 2: Part of the modified program template in the CLEVR language corresponding to 2-hop questions.

4.1 Models

We evaluate our modified NS-VQA described in Section 3 and the CLIP-based model used in [Lindström and Abraham, 2022] on the compositional generalization splits. NS-VQA is trained in two steps; an initial supervised step on a few samples, and then using reinforcement learning with `REINFORCE` on all data. For the supervised step, we train NS-VQA using 100 samples for 2000 iterations. For the `REINFORCE` step, we use all samples in the training data and train for 5000 iterations. Given the narrow set of objectives to learn compared to e.g. in the original CLEVR dataset, NS-VQA does not need the full set of iterations to converge. For the CLIP-based model, we train for 10 epochs, as the loss plateaus after that. We use less data because empirically there was very little difference to using all data, and we want to be concious about how we spend computational resources. We use the same hyper parameters as Johnson *et al.* and Lindström and Abraham for NS-VQA and the CLIP-based model, respectively. Table 1 shows the amount of training, validation, and testing data used by each model.

Model	Training	Validation	Testing
NS-VQA	70 000	15 000	1000
CLIP	10 000	5 000	1000

Table 1: The sizes of the dataset splits used to train each model. The sizes vary slightly between compositionality splits due to the nature of the scenes, in the order of 100's of samples.

4.2 Length generalization

Complementary to the length generalization experiments in, e.g., [Lake and Baroni, 2018], we also create length generalization splits for generalizing to shorter sequences that those trained on. We construct splits such that the training data contains 1-hop questions, where the test data contains only 2-hop questions. Table 2 illustrates the splits we use. We also note that 2- and 3-hop questions inherently address the bag-of-words issue identified by Wu *et al.* , since *Remove all blue cubes. Remove all red spheres.* rely on the pairing of attributes in the two subtasks. Our 2- and 3-hop questions also introduce a bit more linguistic complexity, following the issues identified by Qiu *et al.*. In order to focus on length generalization, the attributes are restricted to shapes and colors in these experiments.

4.3 Attribute generalisation

For our `no-red-cubes` split, we create training data with no questions involving red cubes, while the test data consists of only questions mentioning red cubes. The questions are all 1-hop questions, to isolate the compositional generalization over attributes from length generalization. The training data contains no questions on red cubes, however, they are present in the scenes.

4.4 Reproducability

The code used to generate the data used in our experiments and the modified NS-VQA will be made available through Github in the final submission The models were trained on a local GPU cluster using 1 NVIDIA V100-card. The code uses NVIDIA driver version 515.105.01 with CUDA 11, and `Pytorch 1.13.1+cu117`.

Name	Training	Testing
1-hop to 2-hop	Remove all blue cubes. How many objects are left?	Remove all blue cubes. Remove all red spheres.?
3-hop to 2-hop	Remove all blue cubes. Remove all spheres. Remove all gray cylinders. How many objects are left?	Remove all blue cubes. Remove all red spheres.?

Table 2: Examples of the length generalization splits of the benchmark.

5 Results

Table 3 shows that NS-VQA can generalize to questions with fewer instructions, but completely fails to generalize out of distribution. NS-VQA perfectly learns how to answer 1-, 2-, and 3-hop questions within the training distribution. Looking at the program accuracy, it is close to 100% for n-hop to n-hop, but consistently 0% for all the out of distribution tests. In other words, the system answers the questions correctly, but with an incorrect program. We observe the predicted 1-hop programs for the NS-VQA model trained on 3-hop questions to help explain the behaviour. For the 1-hop sample *Remove all blue cubes. How many objects are left?*, the 3-hop-trained model produces the program sequence equivalent to *Remove all blue objects. Remove all blue cubes. Remove all blue cubes. How many objects are left?*. If only cubes are blue in the scene, the model will still answer correctly. Similarly, the two "extra" remove operations can be of objects that are not present in the scene and still answer the question correctly. It seems that the LSTM in NS-VQA produces the first $n-1$ operations of the program, but always outputs a 3-hop structure of three chained filter-remove pairs.

Model	Trained on	1-hop	2-hop	3-hop
NS-VQA	1-hop	1.0	0.0	0.0
	2-hop	0.48	1.0	0.03
	3-hop	0.20	0.82	1.0
CLIP	1-hop	0.50	0.20	0.0
	2-hop	0.11	0.31	0.29
	3-hop	0.05	0.21	0.33

Table 3: Accuracy of NS-VQA and CLIP-based model on length generalization over multihop questions. Results averaged over 5 runs.

Table 4 shows the performance of NS-VQA and the CLIP-based model on the `no-red-cubes` split. Both models see a degradation in performance when tested on out-of-distribution compositions. NS-VQA goes from perfect accuracy to answering only 42% of the questions correctly. The CLIP-based model drops to 20% accuracy on the red cubes

test data, which is not significantly better than a simple majority vote (the largest class is 3 as the answer, making up about 15% of the samples). We note that the CLIP-performance in Table 3 and Table 4 for 1-hop questions does not align. Running the training for more epochs does not address this issue. One possible explanation is that the constraints on the `no-red-cubes` training data to exclude red cubes gives a more difficult distribution to learn. The results are instead similar to the 2- and 3-hop performance. However, we also note that the relative performance is what we are interested in here.

Model	No Red Cubes (I.I.D.)	Red Cubes
NS-VQA	1.0	0.42
CLIP	0.30	0.20

Table 4: Accuracy of NS-VQA and CLIP-based model over multi-hop questions. Results averaged over 5 runs.

6 Discussion and Conclusion

We introduced a modified NS-VQA architecture that can learn the previously impossible multihop questions. We have shown that both NS-VQA and CLIP handles our length generalization task poorly. The results show similar patterns where generalizing to more hops is difficult, with e.g. 1- to 3-hop generalization seeing complete failure of 0% accuracy for both models. For NS-VQA, generalizing to fewer hops sees less of a degradation but further investigation shows that the model hallucinates and forces the predicted program sequences to use as many hops as trained on. The CLIP-based model follows a similar pattern. The results are also in line with related work on math word problems such as Lan *et al.* where LSTMs and Transformers perform similarly out of the box.

One previous argument in the domains of neuro-symbolic language learning and compositional generalization, is that the language model fails on such tasks because it cannot capture the complexity of language sufficiently. CLIP uses a much stronger language component than NS-VQA, but both fail in similar ways on the splits presented in this paper. Instead, we would like to draw focus to the learning procedure and how we can inject more knowledge.

We argue that learning recursive functions is one key challenge in compositional generalization. Currently, even if NS-VQA *does* answer some 1-hop questions correctly when trained on 2-hop questions, our investigation shows that it does not do so by partial application of the 2-hop function. If both the architecture and the learning procedure would reflect the recursive nature of subtraction, generalizing from 1- to 2-hop should be no different than from 1-hop to 20-hop questions. Related work partially achieve this by splitting the multiple hops into separate prompts, but this somewhat reduces the problem complexity in the preprocessing step. One current research direction is to look at curriculum learning based on attributes and program complexity. Recent results in curriculum learning shows how curricula can help reduce the amount of data needed in training. However, the effects on compositional generalization is unclear.

In ongoing work, we are devising more compositional generalization splits and will evaluate on multimodal transformer architectures such as ViLT. In this work, we aim to answer scaling laws questions concerning how many examples of a specific composition are needed to achieve good compositional generalizability. In preliminary work, NS-VQA only needs a handful of examples to learn the generalization, whereas this has little impact on CLIP.

References

[Arras *et al.*, 2022] Leila Arras, Ahmed Osman, and Wojciech Samek. Clevr-xai: a benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40, 2022.

[Fodor and Pylyshyn, 1988] Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988.

[Hong *et al.*, 2023] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images. *arXiv preprint arXiv:2303.11327*, 2023.

[Huang *et al.*, 2016] Danqing Huang, Shuming Shi, Chin-Yew Lin, Jian Yin, and Wei-Ying Ma. How well do computers solve math word problems? large-scale dataset construction and evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 887–896, 2016.

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

[Hupkes *et al.*, 2020] Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: how do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

[Johnson *et al.*, 2017] 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 the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

[Kim and Linzen, 2020] Najoung Kim and Tal Linzen. Cogs: A compositional generalization challenge based on semantic interpretation. *arXiv preprint arXiv:2010.05465*, 2020.

[Kottur *et al.*, 2019] Satwik Kottur, José MF Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. Clevr-dialog: A diagnostic dataset for multi-round reasoning in visual dialog. *arXiv preprint arXiv:1903.03166*, 2019.

[Lake and Baroni, 2018] Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882. PMLR, 2018.

[Lan *et al.*, 2022] Yunshi Lan, Lei Wang, Jing Jiang, and Ee-Peng Lim. Improving compositional generalization in math word problem solving. *arXiv preprint arXiv:2209.01352*, 2022.

[Li *et al.*, 2022] Zhuowan Li, Xingrui Wang, Elias Stengel-Eskin, Adam Kortylewski, Wufei Ma, Benjamin Van Durme, and Alan Yuille. Super-clevr: A virtual benchmark to diagnose domain robustness in visual reasoning. *arXiv preprint arXiv:2212.00259*, 2022.

[Lin *et al.*, 2023] Baihan Lin, Djallel Bouneffouf, and Irina Rish. A survey on compositional generalization in applications. *arXiv preprint arXiv:2302.01067*, 2023.

[Lindström and Abraham, 2022] Adam Dahlgren Lindström and Savitha Sam Abraham. Clevr-math: A dataset for compositional language, visual and mathematical reasoning. In Artur S. d'Avila Garcez and Ernesto Jiménez-Ruiz, editors, *Proceedings of the 16th International Workshop on Neural-Symbolic Learning and Reasoning as part of the 2nd International Joint Conference on Learning & Reasoning (IJCLR 2022), Cumberland Lodge, Windsor Great Park, UK, September 28-30, 2022*, volume 3212 of *CEUR Workshop Proceedings*, pages 155–170. CEUR-WS.org, 2022.

[Linzen, 2020] Tal Linzen. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, 2020.

[Liu *et al.*, 2019] Runtao Liu, Chenxi Liu, Yutong Bai, and Alan L. Yuille. Clevr-ref+: Diagnosing visual reasoning with referring expressions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[Loula *et al.*, 2018] Joao Loula, Marco Baroni, and Brenden M Lake. Rearranging the familiar: Testing compositional generalization in recurrent networks. *arXiv preprint arXiv:1807.07545*, 2018.

[Mitra and Baral, 2016] Arindam Mitra and Chitta Baral. Learning to use formulas to solve simple arithmetic problems. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2144–2153, 2016.

[Qiu *et al.*, 2021] Linlu Qiu, Hexiang Hu, Bowen Zhang, Peter Shaw, and Fei Sha. Systematic generalization on gscan: What is nearly solved and what is next? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2180–2188, 2021.

[Qiu *et al.*, 2022] Linlu Qiu, Peter Shaw, Panupong Pasupat, Pawel Nowak, Tal Linzen, Fei Sha, and Kristina Toutanova. Improving compositional generalization with latent structure and data augmentation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4341–4362, Seattle, United States, July 2022. Association for Computational Linguistics.

[Robaidek *et al.*, 2018] Benjamin Robaidek, Rik Koncel-Kedziorski, and Hannaneh Hajishirzi. Data-driven methods for solving algebra word problems. *arXiv preprint arXiv:1804.10718*, 2018.

[Ruis *et al.*, 2020] Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. A benchmark for systematic generalization in grounded language understanding. *Advances in Neural Information Processing Systems*, 33, 2020.

[Salewski *et al.*, 2022] Leonard Salewski, A Sophia Koepke, Hendrik PA Lensch, and Zeynep Akata. Clevr-x: A visual reasoning dataset for natural language explanations. In *xxAI-Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*, pages 69–88. Springer, 2022.

[Sampat *et al.*, 2021a] Shailaja Keyur Sampat, Akshay Kumar, Yezhou Yang, and Chitta Baral. CLEVR_HYP: A challenge dataset and baselines for visual question answering with hypothetical actions over images. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3692–3709, Online, June 2021. Association for Computational Linguistics.

[Sampat *et al.*, 2021b] Shailaja Keyur Sampat, Akshay Kumar, Yezhou Yang, and Chitta Baral. CLEVR_HYP: A challenge dataset and baselines for visual question answering with hypothetical actions over images. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3692–3709, Online, June 2021. Association for Computational Linguistics.

[Shaw *et al.*, 2021] Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 922–938, 2021.

[Sikarwar *et al.*, 2022] Ankur Sikarwar, Arkil Patel, and Navin Goyal. When can transformers ground and compose: Insights from compositional generalization benchmarks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 648–669, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[Stammer *et al.*, 2021] Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3619–3629, 2021.

[Sundaram and Abraham, 2018] Sowmya S Sundaram and Savitha Sam Abraham. Solving simple arithmetic word

problems precisely with schemas. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 542–547. Springer, 2018.

[Sundaram and Khemani, 2015] Sowmya S Sundaram and Deepak Khemani. Natural language processing for solving simple word problems. In *Proceedings of the 12th International Conference on Natural Language Processing*, pages 394–402, 2015.

[Sundaram *et al.*, 2020] Sowmya S Sundaram, P Deepak, and Savitha Sam Abraham. Distributed representations for arithmetic word problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9000–9007, 2020.

[Weißenhorn *et al.*, 2022] Pia Weißenhorn, Lucia Donatelli, and Alexander Koller. Compositional generalization with a broad-coverage semantic parser. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 44–54, 2022.

[Wu *et al.*, 2021] Zhengxuan Wu, Elisa Kreiss, Desmond Ong, and Christopher Potts. ReaSCAN: Compositional Reasoning in Language Grounding. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[Yi *et al.*, 2018] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31, 2018.

[Yi *et al.*, 2020] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: CoLlision Events for Video REpresentation and Reasoning. *arXiv:1910.01442 [cs]*, March 2020. arXiv: 1910.01442.