



UMEÅ UNIVERSITY

# On Learning, Reasoning, and Compositional Generalization in Multimodal Models

*Adam Dahlgren Lindström*

DOCTORAL THESIS, 2023  
DEPARTMENT OF COMPUTING SCIENCE  
UMEÅ UNIVERSITY  
SWEDEN

Department of Computing Science  
Umeå University  
SE-901 87 Umeå, Sweden

*dali@cs.umu.se*

Copyright © 2023 by authors  
Except Paper I, © TODO  
Paper II, © TODO

**ISBN TODO**  
**ISSN TODO**  
**UMINF TODO**

Cover illustration by TODO  
Printed by TODO 2023

*Nothing is more usual and more natural for those, who pretend to discover any thing new to the world in philosophy and sciences, than to insinuate the praises of their own systems, by decrying all those, which have been advanced before them. A treatise of human nature (Hume, 1739)*



# Popular Science Abstract



# Sammanfattning





# Abstract



# Preface



# Acknowledgements



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	What Hill are we Climbing? . . . . .	6
1.2	On the Thesis Composition . . . . .	6
1.3	Research questions . . . . .	6
1.4	On the Choice of Methods and Experimental Approach . . . . .	7
1.5	Identity of thesis . . . . .	10
1.6	Thesis Outline and Contributions . . . . .	11
<b>2</b>	<b>What we can learn from history - working title</b>	<b>15</b>
2.1	Clever Hans . . . . .	23
<b>3</b>	<b>Literature Review of Current Challenges/Opportunities in NLP/With LLMs - working title</b>	<b>27</b>
3.1	Language Modeling . . . . .	27
3.2	Multimodality . . . . .	29
3.3	Compositionality . . . . .	33
3.4	Neuro-symbolic machine learning - is not a challenge of NLP, is a response - where to put? . . . . .	47
3.5	Challenges and characteristics . . . . .	53
<b>4</b>	<b>Probing multimodal language models</b>	<b>55</b>
4.1	What can probing tell us? . . . . .	56
4.2	Probing Multimodal Embeddings for Linguistic Properties . . . . .	60
4.3	NOT REWRITTEN AT ALL - Bridging Perception, Memory, and Inference through Semantic Relations . . . . .	75
4.4	Challenges and characteristics . . . . .	81
<b>5</b>	<b>The compositional behaviour of multimodal language models</b>	<b>83</b>
5.1	DeepProbLog and compositionality . . . . .	83
5.2	Multimodal Word Math Problems . . . . .	85
5.3	CLEVR-Math . . . . .	88
5.4	Extending NS-VQA for Multihop Questions . . . . .	97
5.5	Experiments with modified NS-VQA on Compositional Generalisation splits . . . . .	101

5.6	Challenges and characteristics . . . . .	101
<b>6</b>	<b>Multimodal Compositional Generalization</b>	<b>105</b>
6.1	Compositional Generalization splits for CLEVR-Math . . . . .	105
6.2	Methods . . . . .	108
6.3	Results . . . . .	113
6.4	Discussion . . . . .	117
6.5	Probing CLEVR Attribute Compositionality . . . . .	118
<b>7</b>	<b>Using Concept Hierarchies to Improve Compositional Generalisation</b>	<b>125</b>
7.1	Language Learning in Developmental Psychology . . . . .	126
7.2	Concept learning . . . . .	129
7.3	Curriculum Learning with Concept Hierarchies . . . . .	132
7.4	Compositional generalisation benchmark using hierarchical pseudoword concepts in CLEVR . . . . .	133
<b>8</b>	<b>Conclusions</b>	<b>143</b>



# List of Figures

1.1	Example image generated via Blender using CLEVR. One of the generated questions regarding the scene in this is “ <i>What is the color of the metal cylinder that is behind the cyan matte thing?</i> ” to which the answer is <i>red</i> . . . . .	8
1.2	TODO adversarial example . . . . .	9
1.3	TODO overview figure showing the learning process with internal/external insights . . . . .	10
2.1	ChatGPT given the same prompt as ELIZA. . . . .	18
2.2	Illustration of the Peircean model of symbol-referent-thought . . . . .	20
2.3	Tentative illustration of language model vs. world model. . . . .	21
3.1	TODO Illustration of VQA tasks . . . . .	31
3.2	Illustration of (a) <i>lexical</i> and (b) <i>structural</i> generalisation in COGS. TODO ask permission or recreate. . . . .	39
3.3	A sample image and questions from CLEVR. Questions test aspects of visual reasoning such as <b>attribute identification</b> , <b>counting</b> , <b>comparison</b> , <b>multiple attention</b> , and <b>logical operations</b> . TODO ask permission or recreate. . . . .	42
3.4	Illustration of the relationship between natural language questions, their intermediate logical form, and the corresponding SPARQL query used to extract the answer from Freebase. TODO ask for permission to use figure, or remake. . . . .	42
3.5	Example taken from (Thrush et al., 2022) . . . . .	44
4.1	Image-caption pairs (top) and how vectors representing the words ‘bat’, ‘club’, and ‘bird’ may be affected by the image information (above) . . . . .	61
4.2	In task <i>SemanticCongruence</i> , the objective is to recognise semantically implausible captions. . . . .	66
4.3	In this work we focus on recovering synonyms, hypernyms, hyponyms, and meronyms from natural language models via probing to understand the prerequisites of integration with knowledge bases. . . . .	77

5.1	Confusion matrix for DeepProbLog on MNIST . . . . .	84
5.2	Confusion matrix for DeepProbLog on ColorMNIST using the same color maps for training and testing. . . . .	85
5.3	Confusion matrix for DeepProbLog on ColorMNIST using different color maps for training and testing. . . . .	85
5.4	Selection of questions generated from this image: (i) <i>Remove all gray spheres. How many spheres are there? (3),</i> , (ii) <i>Take away 3 cubes. How many objects are there? (7),</i> (iii) <i>How many blocks must be removed to get 1 block? (2)</i> . . . . .	87
5.5	CLEVR-Math example question <i>Take away 2 matte cylinders. How many objects are left?</i> with corresponding mathematical equation $X = 9 - 2$ . . . . .	89
5.6	Example image-question pairs from CLEVR-Math, 5.6a showcase addition and subtraction, and 5.6b shows multihop reasoning. Answers in parenthesis. . . . .	90
5.7	The attributes are used evenly throughout the dataset, whereas the answers are biased towards the smaller numbers. The numbers are aggregated over all splits. . . . .	93
5.8	Examples of when CLIP and NS-VQA fails on multihop questions. . . . .	96
5.9	Confusion matrix for CLIP trained on 20 000 samples. . . . .	98
5.10	Illustration of the execution tree produced by the original NS-VQA parser for the question “Remove all cubes. Remove all brown cylinders. How many objects are left?”. . . . .	102
5.11	Illustration of the execution tree produced by the modified NS-VQA parser for the question “Remove all cubes. Remove all brown cylinders. How many objects are left?”. . . . .	103
6.1	An example from CLEVR-Math (Lindström & Abraham, 2022b), with a corresponding functional program. . . . .	106
6.2	Three examples from our generalization splits, showing splits on productivity, systematicity, and substitutivity. The last row shows the functional programs for the function generalization examples from 1- to 2hop. The function blocks in dashed boxes illustrates reasoning hops for the corresponding problems. . . . .	109
6.3	Dependency tree for the original formulation of 1-hop questions, seen in Table 6.2. Generated from the text <i>Remove all cubes, how many objects are left?</i> using <a href="https://corenlp.run/">https://corenlp.run/</a> . . . . .	111
6.4	Dependency tree for paraphrasing 2 in Table 6.2. Generated from the text <i>How many objects are left after removing all cubes?</i> using <a href="https://corenlp.run/">https://corenlp.run/</a> . . . . .	112
6.5	An example question and the corresponding functional program in the CLEVR diagram language (Johnson et al., 2017). The dotted scene represents the internal representation used by our modified NS-VQA. . . . .	113
6.6	Image from (Sikarwar et al., 2022) todo ask for permission. . . . .	119

6.7	Probing of each layer in BERT shows how NLP tasks interleave roughly corresponding to a classical NLP pipeline. The performance on Part-of-Speech tagging (POS) comes from the earlier layers, whereas the performance on a more complex task such as co-reference resolution comes from the layers at the very end. TODO ask for permission . . . . .	120
6.8	Probing the layers of ViLT for shape, color, and the number of instructions in the original question. . . . .	122
6.9	Probing ViLT for 2-grams. . . . .	122
7.1	Example of card from the Wug Test (Berko, 1958), showing a task of applying morphological rules to a novel (pseudo)word. . . . .	127
7.2	A simple example of data generated in CLEVR, where we see two pseudoconcepts; a) a <i>blargh</i> – two small cubes next to each other, and b) a <i>perde</i> – a large cyan sphere. . . . .	134
7.3	The answer hierarchy for CLEVR, as introduced by Askarian et al. (2021). Used to define hardness for their curriculum learning strategy. . . . .	136
7.4	Example of a pseudoconcept hierarchy over the CLEVR vocabulary. . . . .	137
7.5	Illustration of how a curriculum can be used to investigate the effects of training on compositional generalisation. . . . .	139



# List of Tables

3.1	Compositional generalisation benchmarks . . . . .	38
3.2	caption . . . . .	49
4.1	Probing tasks for semantic embeddings, organized along three broader probing categories as investigated in Conneau et al. (2018)	57
4.2	Overview of the investigated embeddings. The total size of the model, including models used to extract precomputed image features, is given in parenthesis. . . . .	68
4.3	Probing accuracies using a MLP with embeddings as input. The bottom three show for each model the difference between the best unimodal and the best merged embedding. All results are averaged over 5 runs and have variance $\leq 0.01$ . . . . .	70
4.4	Accuracy per label of the tested models A more detailed account of the accuracy of the tested models for the task <i>NumObjects</i> . The class labels correspond to the number of objects annotated in the image.. . . . .	72
4.5	Instances of the relations synonymy, hypernymy, and meronymy extracted from WordNet. . . . .	77
4.6	The probing accuracy on the semantic relations, with variance given in parentheses. The accuracy of a “largest class” strategy is shown next to each relation. All transformers give embeddings of 768 dimensions, with word2vec and GloVe using 300 dimension. Each relation contain 1712, 306, 2740, and 1630 samples, respectively. . . . .	80
5.1	Accuracy on evaluation data for both . . . . .	84
5.2	An overview of the different templates implemented by CLEVR-Math. <Z>, <C>, <M>, <S> are instantiated to size, color, material, and shape during the question generation. . . . .	92
5.3	Distribution of templates in each data split. . . . .	94
5.4	Huggingface dataset card for CLEVR-Math. . . . .	94
5.5	Accuracy on the CLEVR-Math dataset, shown for each template group and aggregated over all templates. . . . .	95
5.6	Accuracy over all templates for different dataset sizes. . . . .	95

- 6.1 Data splits for function and attribute generalization. The first segment shows the core tasks, the second all attribute splits, and the last segment contains the splits used to investigate the impact of different types of complexity on generalization. . . . 110
- 6.2 Linguistic variations used to investigate effect of syntactic complexity on compositional generalization.  $\langle X \rangle$  is a placeholder for  $\langle Z \rangle \langle M \rangle \langle C \rangle \langle S \rangle$ . . . . . 112
- 6.3 Accuracy of task baselines and novel attribute composition splits. 114
- 6.4 Model accuracy on function generalization over multihop questions, averaged over 5 runs. In percentage, higher is better. Each row represents training on, e.g., 1+3hop and the performance on the  $n$ -splits. . . . . 115
- 6.5 Accuracy on held out attribution compositions when trained on 5000, 25000, and 50000 samples. . . . . 116

# Chapter 1

## Introduction

Example of how to use quotes at  
the beginning of chapters

---

*dali*

Humans have the innate ability to adapt and learn in various environments by utilizing all their senses and drawing from past experiences. This multifaceted approach helps us decipher complex information and handle new situations effectively. We complement what we see with what we hear to minimise ambiguity in conversations and manage noisy information. These channels of information are *modalities*, and *multimodal* information processing refers to the simultaneously fusing information from different modalities. In dangerous situations we can rely on all our senses to build a complete picture to avoid harm, often without thinking about it. Using our senses like taste, smell, touch, hearing, and sight helps us navigate our environment effectively, and our ability to generalize from experience that makes it easier for us to be successful in new situations and places. Imagine that you are joining friends for driving snow mobiles, an activity that you have never done before. You are listening to a walk through of how to drive, but there is a loud noise momentarily drowning out the presenter. Given the topic and the last few words that you heard, you can probably infer the missing words with reasonable accuracy. If there are slides, you might be able to align what the presenter said with text on the slides to fill in the gap. If you clearly see the person, their facial expressions might be enough to infer what you did not hear. Combining all of this information happens for us without thinking too much about it. Similarly, while you have never driven a snow mobile, you most likely have previous experiences that you can combine to quickly learn. Do you know how to ride a bike? Drive a car? Ski? Even sitting straight on a chair transfers. All these activities have aspects that translate to driving a snow mobile, and thus we do not have to start from scratch even if the environment and task is new. Instead, most people probably only need a few instructions and a couple of tries to drive on their own. This

is of course not the same as mastering the snow mobile, and we will get back to such generalisation throughout this thesis. Multimodal information processing and generalisation to novel scenarios are two core characteristics of human intelligence, and are both central in many research topics and applications of artificial intelligence. However, even with the impressive advancements of the last decade, there is still a long way to go for robust human-level AI.

Throughout this thesis we will investigate multimodal machine learning for language and vision and the generalisation capabilities of such models in visual question answering tasks. We will build an understanding of the effects of including vision in language learning, and current limitations of compositional generalization in multimodal language models. We categorise current challenges in language modelling into three categories; 1) *robustness*, 2) *reasoning*, and 3) *resources*. In light of these challenges, we investigate how neuro-symbolic architectures, systems combining neural networks and symbolic reasoning, and curriculum learning, e.g. learning increasingly difficult tasks, can help address these issues. We further argue that both internal structure and external behaviour must be evaluated in order to understand the capabilities of language models. As our method, we use probing of embeddings to observe internal structures of the embeddings produced by language models, and we construct a benchmark for compositional generalisation to study model behaviour on visual reasoning tasks. In our probing experiments, we also investigate the impact of vision on learning language. The thesis ends with an outlook on the relationship between language grounding, concept hierarchies, and compositional generalisation.

**Multimodality** Now, in the context of artificial intelligence, why should we study multimodal machine learning? We have spent centuries systematically trying to understand and characterise human intelligence, how we understand our surroundings and make sense of each other. How do we translate thoughts into language? How do we learn language through experience and interaction? We smell, taste, touch, see, and hear things to interact with the world around us. These senses represent different modalities, and multimodal language learning simply implies that we are learning language with several modalities as input at the same time, e.g., text-image pairs as input. An example is image captioning, where systems are trained on images with captions to generate captions for new images. This task is impossible unless the system learns how to relate the two modalities.

While human language is dynamic and expressive, building intelligent systems without multimodal capabilities means that we are limiting our human-computer interaction. Asking a question about images is a natural way of interacting with our surroundings, and combining complementary information in audio and visual input can sometimes be crucial in understanding what is going on. As an example, in a setting with noise pollution, humans can still rely on lip movement and facial expressions to follow a conversation as a complement to our hearing. We can teach multimodal language models to do the



same, as shown by Zadeh et al. (2016), to combine complementary information to determine the sentiment of a speaker on video. More importantly, there is evidence from a wide range of disciplines, such as developmental psychology, that language learning in humans either requires or is facilitated by our use of different senses. For instance, we speak about the exact same objects using different words in the word/concept hierarchy. When referring to an image of a cat, a speaker might use *Norwegian forest cat*, *cat*, and *animal*, all in the same sentence structure depending on what level of resolution the context requires. If there are many cats present, we might have to be more specific to disambiguate which animal we are referring to. Similarly, if there are no animals present we can choose each of the alternatives without being ambiguous. Hence, a model using vision has a stronger signal that these words all relate to each other more so than that they can be used in the same way in a sentence to mean the same thing. Current large language models build these hierarchies solely on words used in the same way. While that might work for large parts of language learning, one of the questions in multimodal machine learning is whether using other modalities helps make this learning process more efficient. We must acknowledge that many applications that require natural language processing do not need multimodal capacities to be successful. However, given that we ground our language in our sensory experiences, we want to better understand the role and impact of vision on language learning with machines.

**Natural Language Processing** So, we want to learn models of language to facilitate the interaction between humans and machines, as well as process language to distill or act on the information it contains. Therefore, Natural Language Processing (NLP) is an important component in many fields of artificial intelligence (AI). This may not come as a surprise, as our most dynamic and expressive form of interaction is through language. Historically, natural language plays a central role in some of the most well-known artifacts of AI research. The Turing test centers around written communication, introduced by Alan Turing in 1950 as the *imitation game* (Turing, 1950). The Chinese Room thought experiment taught to undergrad students takes a similar form (Searle, 1980). The famous ELIZA chatter bot (Weizenbaum, 1966) from the 1960's used relatively simple rules to interact with users in natural language. IBM Watson used natural language both as an interface and to query for information when it beat human players in *Jeopardy!*. Recent advancements that gather the attention of the public are no different. Deep learning with methods such as *transformers* or *recurrent neural networks*, in combination with vast datasets of language found on the internet, have revolutionised large parts of the field of NLP. Recently, generative language models such as GPT sent shock waves through the AI community, with ChatGPT dominating news feeds and social media. One of the reasons is that large language models now *seem* capable of performing tasks that are not strictly language processing, such as mathematical reasoning or passing the bar to practise law. We will return to the question of the line between language and general intelligence later, but note that it

is fuzzy in the current research environment. However, as we will see in this thesis, there are challenges with current language models that might require something other than pure scale. While there are strong achievements across disciplines, the neural networks of today are still susceptible to the neural network, or *connectionism*, critiques from over 30 years ago. We will now spend some time understanding these critiques.

**Deep Learning** When we are talking about learning language with machines, deep learning has been an crucial enabler of the systems we have today. The last decade of AI research has been dominated by deep learning in everything from language and vision, to robot control and playing games. Although the term *deep learning* was coined in 1986 by Rina Dechter (Dechter, 1986), the deep learning revolution started in the 2010s. In 2012, researchers used deep learning to achieve drastically better performance on the famous ImageNet challenge. AlexNet was a combination of old ideas and the availability of the necessary GPU hardware (Krizhevsky et al., 2017). Since then, deep learning has become a golden hammer to address many challenges in artificial intelligence. At the same time, deep learning faces critique on a wide range of issues. We have many examples of ethical issues with large language models that reinforce discriminatory patterns and harmful societal biases. There are also issues with robustness in critical tasks such as autonomous driving, or various reasoning tasks (TODO CITE). From environmental and democratic standpoints, the amount of resources required to build large language models is both harmful and ethically questionable. To structure this discussion, we focus on challenges in these three *R*'s; robustness, resources, and reasoning. We will come back to this in Chapter 3. However, we would like to point out that similar arguments goes back to debates in the 1980s, where J. A. Fodor and Pylyshyn (1988) argue that connectionism cannot address the real challenges of artificial intelligence but rather is a step backwards. We will expand on the historical context more in detail in Chapter 2.

Since the 80s, the research landscape went through an AI winter, and while AI research changed dramatically since then, the criticism of neural networks from that time still applies. Deep learning systems achieve great performance on benchmarks, but there are as many examples of how their shortcomings when deployed in real world applications. Most recently this includes harmful bias, catastrophic failures leading to nonsensical repeating, and just plain wrong answers with large language models (TODO CITE). There are entire research fields trying to dissect and understand these failures and general characteristics. We can summaries the issue as the difference between deep learning and deep understanding, and that many benchmarks test model behaviour rather than verify specific characteristics central to intelligence. Chapter 5 will cover how reasoning tasks can inform us in building better language systems, and what role *compositional generalisation* plays in addressing the shortcomings.

**Compositional Generalisation** A key characteristic of human success is our ability to generalise from experience. One aspect of that ability is *compositional generalisation*, the capacity to compose knowledge from previous experiences into new knowledge. Human language and cognition are both largely compositional (todo, 2023), with Chomsky (Chomsky, 2014) stating that we generate *infinite use of finite means* (originally attributed to Von Humboldt (1836)). In other words, we can use language to generate sentences that have never been seen before, such as the famous Chomsky example of *colorless green ideas sleep furiously* (Chomsky, 1957). This theory is not without controversy, but the critique mostly questions infiniteness and where it comes from. However, the lack of compositionality in many current language models is an important challenge in addressing a range of issues (todo, 2023). Chapter 3 will go into more details on compositional generalisation to prepare for the benchmarks introduced in Chapter 5.

**Human-inspired Learning** The concluding chapters of this thesis will look at what we can learn from human learning. Intuitively, there are large differences between how children learn and how we teach, e.g., large language models. There are many instances of how we can use insights from studying humans to improve machine learning methods. However, it is important to note the big difference between using human mechanisms as inspiration and trying to replicate them. Neural networks are a great example of this, with the core idea being inspired by neurons in our brains. In reality, there are a great deal of components that are missing from neural network architectures for them to be even remotely close. Similarly, we note the body of work that criticises the way we train models, not only the architectures themselves. Mollo and Millière (2023) formulate the *vector grounding problem*, a modernised formulation of the grounding problem. One conclusion they draw is that merely having access to multimodal data during training is not sufficient at all to ground the model in the real world.

On the note of changing the way we train neural networks, Z. Liu et al. (2023) introduces Brain-Inspired Modular Training (BIMT) that explicitly encourages networks to be modular and sparse. The main objective is to produce models that are more interpretable, but there are many side-effects that are interesting. The experiments show how a brain-inspired training procedure and loss function can lead to neural networks that are more compositional in structure. Their method penalises parameters with a cost proportional to the length of each neuron connection. One effect is that concepts like compositionality, independence, and feature sharing can be recovered from the network structure.

- Developmental psychology, Chromium (Carey & Bartlett, 1978), Wugs (Berko, 1958), hierarchies (Eustace, 1969)
- Concept learning

- Brenden Lake (B. M. Lake et al., 2019)

## 1.1 What Hill are we Climbing?

The field of artificial intelligence is sometimes referred to as a landscape with hills and valleys. In this analogy, valleys represents dead ends, and the highest hills, or peaks, represent something like our notion of artificial general intelligence. The goal then is to find the highest hill and climb it. One problem with climbing hills is that only moving upwards does not guarantee that you find the highest peak. Following this algorithm in The Netherlands barely gets you above sea level, hardly close to the peak of Mt. Everest. Therefore, it is important to know how to identify the landscape of high hills, and know when we are in a position to start climbing such a hill (G. Marcus & Davis, 2019).

## 1.2 On the Thesis Composition

The challenges outlined above constitutes the context for the work presented in this thesis. We focus on compositional generalisation in multimodal domains, and evaluate the usefulness of neuro-symbolic methods and hierarchies in addressing the challenges. Some of the promises of neuro-symbolic methods specifically address some of the shortcomings of deep learning-based methods. For instance, neuro-symbolic methods integrate logical reasoning frameworks into the architecture. This is useful both for exact and reliable reasoning, but can also help reduce the required amount of training data. Of course, neuro-symbolic methods have shortcomings, with one of the more prominent ones being hand-engineered domains. We will come back to neuro-symbolic methods later. Chapter 4 shows how we can use *probing* to peak into the black boxes that are neural networks. We use probing to look at the effect of learning language with vision, and how large language models differ in their capabilities. Chapter 5 focuses on compositional generalisation in multimodal reasoning tasks, evaluating the behaviour of neuro-symbolic systems and large language models. In the Chapter 7, we propose and evaluate a curriculum learning process where the internal properties and external behaviour will be evaluated using probing and carefully selected test data. We answer questions on how concept hierarchy curricula and multimodal data can help language models achieve compositional generalisation.

## 1.3 Research questions

The aim of this thesis is to contribute to language technology that is capable of robust reasoning in multimodal contexts, while moving towards more trustworthy and transparent AI. Throughout the thesis, the following main research questions are addressed:

- (R1) How can vision contribute to language modeling?
- (R2) What are advantages and disadvantages of neuro-symbolic methods in multimodal language processing?
- (R3) (Opt 1) Does utilizing hierarchical structures improve compositional generalisation in visual question answering?
- (R4) (Opt 2) Does curriculum learning improve compositional generalisation in visual question answering?
- (R5) What are the challenges and opportunities of language-centric learning on multimodal data, and what future research directions are there?

We use probing in Chapter 4 to answer (R1), and revisit the question in Chapter 5. Chapter 5 and ?? address (R2), with Chapter ?? mainly addressing (R4). Each chapter summarises the challenges pertaining to the topic covered, and we summarise the challenges and outlook in Chapter 8.

## 1.4 On the Choice of Methods and Experimental Approach

Two central issue of current large language models is that they perform poorly on the tail-end distribution of tasks, and rely on enormous amounts of data mined from the internet and other sources (Bender et al., 2021a). The mined data itself perpetuates bias, such as systemic racism, and relies on the law of large numbers to sufficiently cover the underlying distribution of language and knowledge. Recent examples include PaLM (Chowdhery et al., 2022), where the authors train on a highly gender-skewed dataset (masculine pronouns are mentioned 5.5 times more often than feminine). Synthetic datasets allow us to control the underlying data distribution and generate examples which highlight a specific edge case or behaviour such as compositional generalisation. The downside is that it is difficult to generate the diversity that large internet-based datasets capture. Therefore, synthetic datasets are a good complement to real world datasets to control for specific expected behaviour or properties. As such, it is suitable for our purposes of examining reasoning and compositional generalisation.

One system for generating such datasets is CLEVR (Johnson et al., 2017), which uses the 3D modelling software *Blender* to generate images of 3D scenes with objects of different size, shape, and material. From the internal representation of the 3D scene, questions regarding the contents of the images are generated from a set of templates and a predefined functional language. The questions are formed in such a way that they require reasoning about the different shapes, materials, and colors of the objects, and their relation in space. Figure 1.1 shows an example of an image and a question-answer pair generated using CLEVR. Since the publication of CLEVR in 2017, several extensions, con-

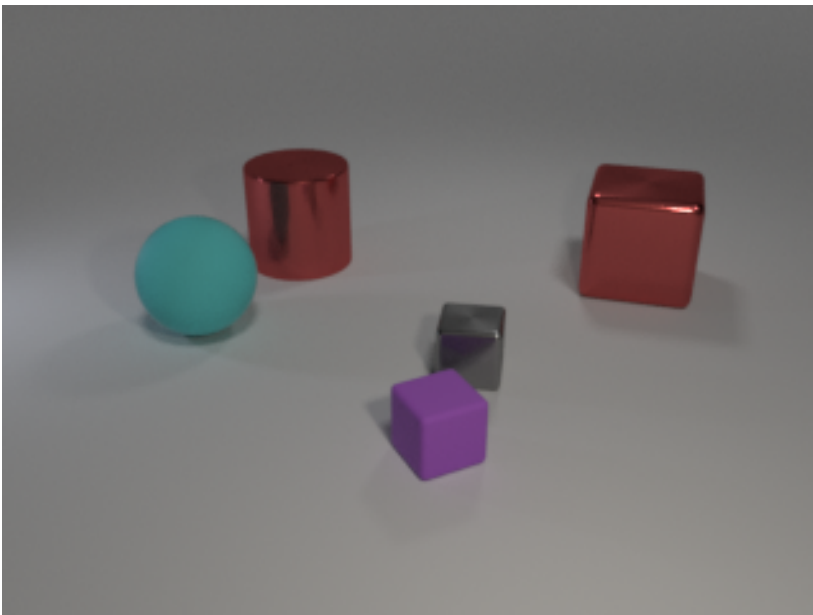


Figure 1.1: Example image generated via Blender using CLEVR. One of the generated questions regarding the scene in this is “*What is the color of the metal cylinder that is behind the cyan matte thing?*” to which the answer is *red*.

tinuations, and modifications have been introduced ([sapat2021clevr\\_hyp](#); Arras et al., 2022; Kottur et al., 2019; Z. Li et al., 2022; R. Liu et al., 2019b; Salewski et al., 2022; Stammer et al., 2021). Stammer et al. (2021) introduce experiments with confounding information in CLEVR-Hans, where properties such as color is used to confound a learning algorithm. To illustrate the concept of confounding information, B. Kim et al. (2019) and Rieger et al. (2020) use ColorMNIST where each digit is associated with a fixed color during training but randomised in testing. Without any inductive bias, it is uncertain how a learning algorithm should distinguish between the color and the digit, as they are always seen together. A learning algorithm can still use both information channels to define the concept compositionally like  $digit_1(1, obj) = color(blue, obj) \wedge shape(1, obj)$ . However, given only a few examples in training where a digit is not associated with a fixed color, a learning algorithm should be able to separate these two features. This becomes even more important since assuming that training data is statistically identical (IID) to test or live data, is an impossible requirement for any non-trivial real data. Therefore, high sensitivity to compositionality is an important component in reliable learning algorithms. An argument against only using deep learning

Figure 1.2: TODO adversarial example

methods is that they pick up on signals that are not interesting to the task. A famous example of this comes from the paper introducing the explainable AI-method LIME (Ribeiro et al., 2016). In this, the authors show how object recognition networks make predictions for the wolf class almost solely based on the presence of snow in the image. Another example is that of adversarial attacks, where small pixel changes in images can lead to turtles being recognised as weapons (Athalye et al., 2018). Conversely, neuro-symbolic reasoning methods are designed to handle these situations better than pure neural networks.

### 1.4.1 The importance of testing for both behaviour and internal properties

One main argument of this thesis is that we must test both the behaviour of a model, using, e.g., a benchmark dataset, and examine internal structures. This is similar to the almost antagonistic relationship between qualitative and quantitative assessments. In the machine learning community in general, quantitative assessments are first class citizens derived directly from the training process, while qualitative ones usually requires more thought. Conceptually, this dichotomy is not new and we can draw parallels to, e.g., quality assurance in engineering. We would never be allowed to drive a car that was only tested fully assembled on a race track 10 times. Instead, we have rigorous processes in place to ensure that each component lives up to safety standards before we even consider taking a prototype for a test drive. In this analogy, testing internal properties is testing each component and testing the external behaviour would be taking the car for a drive in the real world. The same sentiment is echoed in a variety of contemporary literature, and this thesis builds on ideas of e.g. Pavlick (2022).

Throughout this thesis we will see examples of both internal and external testing. The internal testing takes the form of probing, as described in Chapter 4. We use probing of, e.g., word embeddings to see how words for visual concepts are more distinctly represented in multimodal language models than text-only language models, as presented in Chapter 5. To test external behaviour, we construct benchmark datasets and splits to test, e.g., mathematical reasoning in visual question answering. In Chapter 7, we use both techniques to test compositional generalisation in a visual question answering domain. We show how the two techniques are complementary in understanding a complex concept such as compositional generalisation, and how one without the other gives an incomplete picture.

TODO integrate There is work (Berrendorf et al., 2020; Kadlec et al., 2017; Pezeshkpour et al., 2020; Rossi & Matinata, 2020; Y. Wang et al., 2019) critis-

Figure 1.3: TODO overview figure showing the learning process with internal/external insights

ing how knowledge base completion is evaluated. In (Rossi & Matinata, 2020), the authors make the observation that less than 15% of entities cover more than 80% of the facts in many of the datasets. By only predicting facts of this small set of entities, a model can achieve good performance. Thus, a model can memorise the explicit facts of a certain entities without learning anything about the general relationships they describe. There is a clear parallel to how large language models work, where the training does not hold out information such that the testing actually tests for the generalisability of the model. The contributions of (Rossi & Matinata, 2020) are the definition of a set of properties useful in capturing the relations rather than the small set of entities.

They use the inverse relationship train-test leakage examples from FB15K and WN18, showing that father-of and child-of are inverse relations. I.e., learning that one is the inverse of the other means that you can explicitly learn one fact in training and reproduce the inverse in testing.

The authors argue that global metrics such as mean rank et c. does not show strengths and weaknesses of different models making it difficult to compare them. Entity frequency will have a too big influence. They also point to other work that highlight that the metrics only measure positive test facts, but not that false or nonsensical facts get low scores.

Based on these results we can as ourselves *"link prediction for what purpose?"*. Is it to better model the world with graph completion? Unclear how the metric reflects on the performance in an application. Usage for a recommender system gives a different set of desirable properties than if used in a database setting, or medical applications. The application matters, hence the expected properties should be understood and possible to control for.

## 1.5 Identity of thesis

Rodney Brooks recently pointed out that when the term AI first appeared for the original 1956 workshop on artificial intelligence, it implied (artificial) general intelligence. Today, the term artificial general intelligence (AGI) is used as a separate term. "What Will Transformers Transform?" (n.d.) argue that

[..] AGI is a different term than AI now is due to a bunch of researchers a dozen or so years ago deciding to launch a marketing campaign for themselves by using a new buzz acronym.

While the work on some of the topics in this thesis could be considered contributions towards AGI, we will refrain from using the term.



This thesis takes the position that;

- Understanding language is a multimodal endeavour
- Concepts are compositional in nature, a fact that should be reflected in methods for learning
- Neural networks alone are not robust reasoners
- Neuro-symbolic methods fulfill many criteria for more transparent and robust machine learning
- Symbolic systems alone have limited capacity for generalisation over large amounts of data

One problem with deep learning is the data and the metrics used, not the architectures themselves. It is not clear whether neuro-symbolic approaches are clearly better than deep learning ones, or that general intelligence cannot be deep learning based. One important observation is that if we want our models to be capable of reasoning, this must be reflected in how we train them. A part of this means identifying the properties we want and that we can control for in datasets and with metrics. These can be used to construct learning situations that give meaningful models. That is where this thesis comes in.

## 1.6 Thesis Outline and Contributions

The thesis contributes to our understanding of language modelling and the implications of learning with multimodal data. A common problem with language models is that specific properties, such as compositionality, are not explicitly tested for, but rather that focus lies on behaviour (e.g. appearing human).

### Chapter 2

When identifying current challenges in language modeling, it is important to recognise whether they have been addressed historically, perhaps in a different context or formulation. This chapter gives more historical context, relating the topic of this thesis to classical problems such as symbol grounding. We address the background of the research objectives and show how some of the challenges of today have been around for a long time. Following the recent trend of prompt-based interaction with large language models such as ChatGPT, we look at the similarities with the famous ELIZA chatter bot from the 1960's. While the technology powering ChatGPT is vastly different from the simple rule-based ELIZA bot, we argue that the fundamental problems of anthropomorphisation are the same. Further, we outline classical dilemmas such as the Chinese Room Argument and the symbol grounding problem.

### Chapter 3

Chapter 3 gives an overview of current robustness, reasoning, and resource challenges with language models. It covers background and related work in compositional generalisation, neuro-symbolic AI, and multimodal machine learning.

### Chapter 4

Chapter 4 covers the probing experiments for visual and linguistic properties;

- Dahlgren Lindström, A., Björklund, J., Bensch, S., & Drewes, F. (2020). Probing multimodal embeddings for linguistic properties: The visual-semantic case. *Proceedings of the 28th International Conference on Computational Linguistics*, 730–744. <https://doi.org/10.18653/v1/2020.coling-main.64>
- Björklund, J., Dahlgren Lindström, A., & Drewes, F. (2021). Bridging perception, memory, and inference through semantic relations. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 9136–9142. <https://aclanthology.org/2021.emnlp-main.719>

### Chapter 5

Chapter 5 introduces a dataset for multimodal visual reasoning and focuses on **learning** in neuro-symbolic methods, addressing research question (**R2**);

- Lindström, A. D., & Abraham, S. S. (2022a). Clevr-math: A dataset for compositional language, visual and mathematical reasoning. In A. S. d’Avila Garcez & E. Jiménez-Ruiz (Eds.), *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* (pp. 155–170, Vol. 3212). CEUR-WS.org. <https://ceur-ws.org/Vol-3212/paper11.pdf>
- KBCG-paper on NS-VQA extension

### Chapter 6

This chapter introduces data splits of CLEVR-Math to investigate compositional **generalization** in visual question answering for neural and neuro-symbolic methods. The benchmark evaluation is complemented by probing experiments investigating how well the CLEVR attributes are compositionally represented.

- ColorMNIST + DeepProbLog
- AAAI paper

## Chapter 7

Chapter 7 uses the techniques in Chapter 4 and Chapter 5 to answer research question (R4) on the effect of curriculum learning on compositional generalisation capabilities in language models;

- Complement experiments from AAAI paper

## Chapter 8

This chapter summarises the conclusions of the thesis, as well as provides an outlook for what future research directions there are.

### 1.6.1 Contributions Not Included in This Thesis

- Tubella, A. A., Mollo, D. C., Lindström, A. D., Deviney, H., Dignum, V., Ericson, P., Jonsson, A., Kampik, T., Lenaerts, T., Mendez, J. A., & Nieves, J. C. (2023). Acropolis: A descriptive framework for making sense of fairness
- Björklund, J., Lindström, A. D., & Drewes, F. (2022). An algebraic approach to learning and grounding. *LearnAut 2022*



## Chapter 2

# What we can learn from history - working title

This chapter will outline four historically important topics/themes/problems; The Chinese Room, Symbol Grounding, ELIZA, and Fodor on concepts.

Many neuro-symbolic methods take a practical approach to the frame problem by working on predefined sets of symbols, thus scoping what is considered “interesting” to a particular system. The symbol set can be viewed as the frame. In recent years, there are many examples of large language models passing the Turing test. The test itself has been critiqued throughout the years, with more recent examples emphasising how LLMs can pass one instance while still performing catastrophically on a related task. ChatGPT is one such example.

### ELIZA

We can attribute some of the success of ChatGPT to the underlying deep learning techniques and progress on processing large amounts of data. However, the human-centered influence on the design, i.e. building an interactive model with a low threshold, is arguably the key to its widespread success. In many ways, this design is not that different from ELIZA (Weizenbaum, 1966), that impressed researchers and the general public almost 60 years ago. ELIZA convinced some users of its intelligence, with people reporting long meaningful conversations. The following is an exchange between ELIZA and one such user:

Men are all alike

IN WHAT WAY

They're always bugging us about something or other.

CAN YOU THINK OF A SPECIFIC EXAMPLE

Well, my boyfriend made me come here.

YOUR BOYFRIEND MADE YOU COME HERE

He says I'm depressed much of the time.

I AM SORRY TO HEAR YOU ARE DEPRESSED

It's true. I am unhappy.

DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY

I need some help, that much seems certain.

WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP

Perhaps I could learn to get along with my mother.

TELL ME MORE ABOUT YOUR FAMILY

My mother takes care of me.

WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU

My father.

YOUR FATHER

You are like my father in some ways.

WHAT RESEMBLANCE DO YOU SEE

You are not very aggressive but I think you don't want me to notice that.

WHAT MAKES YOU THINK THAT I AM NOT VERY AGGRESSIVE

You don't argue with me.

WHY DO YOU THINK I DON'T ARGUE WITH YOU

You are afraid of me.

DOES IT PLEASE YOU TO BELIEVE I AM AFRAID OF YOU

My father is afraid of everybody.

WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER

Bullies.

DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE

This exchange also highlights that gender bias issues is not novel to large language models, where, e.g., ELIZA becomes gendered based on its user (Dillon, 2020). Weizenbaum built ELIZA on principles from Rogerian psychotherapy (C. R. Rogers & Carmichael, 1942), where one core idea is that the therapist reflects the word of the patient in a positive light. With simple rules for how to reflect the input of the user, ELIZA managed to elicit attribution of

human-like feelings from the users. Douglas Hofstadter defines this as the Eliza effect (Hofstadter, 1995) in *The Ineliminable Eliza Effect and Its Dangers*, a preface of *Fluid Concepts and Creative Analogies: Computer Models and the Fundamental Mechanisms of Thought*;

a host of implications follow in the minds of many if not most readers, such as these: [...] computers understand the physical world; computers make analogies; computers reason abstractly; computers make scientific discoveries; computers are insightful cohabiters of the world with us.

This type of illusion is generally known as the “Eliza effect”, which could be defined as the susceptibility of people to read far more understanding than is warranted into strings of symbols – especially words – strung together by computers. (p. 157)

Hofstadter goes on to say that the Eliza effect is a part of AI researchers hyping achievements. We seem to have returned to the ideas that ELIZA was built on, equipped with much stronger building material and tools. Given our own flawed ability to determine intelligence from interaction, we should be critical of impressive results with interactive AI systems in general. This position motivates the work on probing conducted throughout this thesis, as a way to look beyond behaviour and into the box.

Generalisation is not a necessity for weak AI, and there is a vast range of applications where strong AI is not needed. For instance, monitoring and controlling the components of an aircraft can benefit from AI tools, but does not require a generally intelligent agent to work. With a human-centered perspective, this is a key argument — we want AI tools to support humans, not replace them. However, strong AI with general intelligence could replace.



## Chinese Room

In his paper *Minds, Brains, and Programs*, John Searle introduces the Chinese Room thought experiment (or *Gedankenexperiment* as he puts it) as an argument for why strong AI cannot be achieved without “just by designing programs but would have to duplicate the causal powers of the human brain” (Searle, 1980). Specifically, Searle take issue with the following *strong AI* claims; 1) “that the machine can literally be said to understand the story and provide the answers to questions”, and 2) “that what the machine and its program do explains the human ability to understand the story and answer questions about it”. We can restate the Chinese Room thought experiment as follows;

Suppose that a person is locked in a room with a large amount of Chinese writing, and no knowledge of the language, and a set of rules for how to produce Chinese symbols from Chinese text input. The person then uses the Chinese writing and rules to interact with

DA Men are all alike





It is not accurate to make a sweeping generalization that all men are alike. Men, like women, come from diverse backgrounds, cultures, and upbringings, and possess unique personalities, beliefs, and behaviors.  

While there may be some similarities among men due to socialization, genetics, and cultural expectations, it is important to recognize and respect the individuality of each person, regardless of their gender.

It is unfair and inaccurate to judge or stereotype an entire group of people based on the actions or behaviors of a few individuals. It is important to approach each person as an individual, with their own unique qualities and experiences.

DA They're always bugging us about something or other.



It is not fair to make a sweeping generalization that men are always bugging women about something or other. While it is true that some individuals, regardless of gender, may be more persistent in their communication or requests, it is important to address such behavior on a case-by-case basis.  

Effective communication is key in any relationship, and it is important to communicate your needs and boundaries clearly to the other person. If you feel like someone is consistently bugging you about something, it is important to have an open and honest conversation with them about how their behavior is affecting you.

Additionally, it is important to recognize that men, like women, have their own needs and concerns. It is important to approach any relationship with respect and understanding for the other person's perspective. By fostering mutual respect and understanding, we can build stronger and healthier relationships, regardless of gender.

Figure 2.1: ChatGPT given the same prompt as ELIZA.



a native Chinese speaker through a text-only interface. If the rules are exhaustive enough, nobody can tell from the answers alone that the person in the room don't speak a word of Chinese.

Searle goes on to point out that the person locked in the room “[...] produce the answers by manipulating uninterpreted formal symbols.” and is simply behaving like a computer where the rules are analogous to a computer program. The main claim is that the person does not *understand* Chinese, but only simulates that ability, and that understanding can only be achieved with *intentionality*. He later summarised this position as “*Syntax by itself is neither constitutive of nor sufficient for semantics.*” in Searle (1990).

While the Chinese Room thought experiment illustrates a core issue in the strong vs. weak AI debate, it is also simplified. We will defer the *strong* vs. *weak* AI debate to some other time, there have been many replies to Searles' position taking issue with both the formulation and the premises. TODO Perhaps address which criticism this thesis scribes to? One interesting critique is that while the non-Chinese speaker cannot assign any meaning to the sentences it manipulates, it inherits meaning from the Chinese speaker who is interacting with the world. This argument is similar to how people ascribe meaning to systems such as ELIZA, that meaning is in the eye of the beholder. Today, the default assumption of deep learning research is to assume that the person *does* know Chinese only from applying rules. In that context, it is important to point to the contemporary debate and critique of large language models with the Octopus Test introduced by Bender and Koller (2020). However, and use the Chinese Room argument to illustrate why it is important to not only examine the behaviour of a system. In Chapter 4 of this thesis, probing is used in a way to pry open the proverbial door to the Chinese room and look inside. Some of the work investigates how strongly a symbol is grounded to an image, and the implications of this for multimodal methods. This leads us to our next topic, symbol grounding.

## The Symbol Grounding Problem

Following Searles' phrase “*Syntax by itself is neither constitutive of nor sufficient for semantics*”, others also emphasise the impossibility of learning Chinese from dictionary alone. Stevan Harnad proposes that intelligent agents must solve the *symbol grounding problem* in order to give meaning to the language, by connecting it to physical systems and subjective experiences (Harnad, 1990). Harnad distinguishes between intrinsic and extrinsic meaning; the output from the Chinese room only carries extrinsic meaning imposed by the Chinese speaker. Importantly, he argues that meaning cannot come from symbols alone, but must but built on top of the intrinsic meaning of our non-symbolic interpretations of our sensory input. In a way, Harnad extends on Fodor in saying that it is not enough that “*that the meaning of the symbols comes from connecting the symbol system to the world "in the right way"*”. Instead, giving

# The semiotic triad

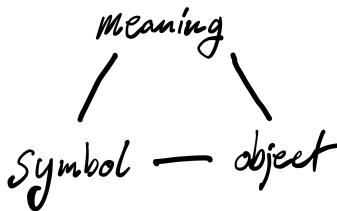


Figure 2.2: Illustration of the Peircean model of symbol-referent-thought

words intrinsic meaning is intertwined with the cognitive processing of sensory input. Motivated by this division into intrinsic and extrinsic meaning, Harnad proposes a hybrid system that combines symbolic and non-symbolic elements. Moreover, he provides the following motivation for connectionism, in a description of what we now would call a neuro-symbolic system.

Connectionism is one natural candidate for the mechanism that learns the invariant features underlying categorical representations, thereby connecting names to the proximal projections of the distal objects they stand for. In this way connectionism can be seen as a complementary component in a hybrid non-symbolic/symbolic model of the mind, rather than a rival to purely symbolic modeling. Such a hybrid model would not have an autonomous symbolic “module,” however; the symbolic functions would emerge as an intrinsically “dedicated” symbol system as a consequence of the bottom-up grounding of categories’ names in their sensory representations.

Harnad (1993) later describes the *frame problem* (McCarthy, 1960) in relation to the symbol grounding problem as the problem of connecting iconic and categorical representations to symbolic representations that are required by a given context. For instance, we do not name every concept that we possibly can while tasked with answering what color the cat is in an image. I.e., humans perceive and process all our sensory input but we only connect them to symbols when those symbols are useful to us.

Humans ground language to our experience of the world, meaning that

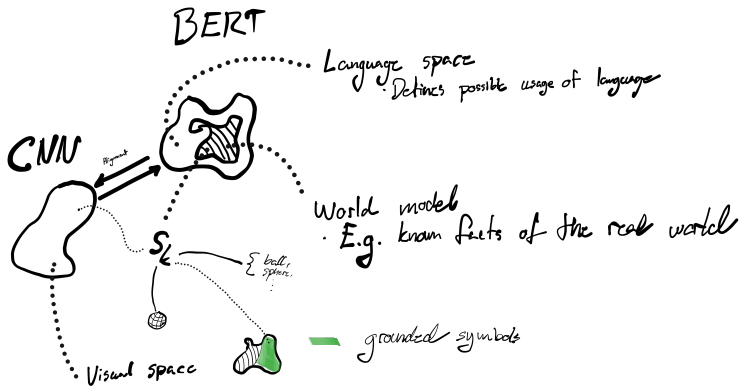


Figure 2.3: Tentative illustration of language model vs. world model.

through our interaction with it we attach meaning to language. The alignment of our internal model and our use of language means that while our world view is slightly influenced by our language, the concepts we learn are more universal.

Many of the issues with large language models are related to the fact that they are also world models, and that we have no direct way to observe or intervene on that internal world model. For instance, bias or errors in "understanding" are related to the model of the world. The process of externalising this internal model is grounding it in other modalities. Our language reflects our own internal world views, and similarly we could do multi-agent simulations in order to derive internal world models from agent interactions. This can be a continuation of Machine Theory of Mind (Rabinowitz et al., 2018) where agents are instantiated with different sets of beliefs and behaviours. The task is for agents to use interactions to construct a model of the other agents' internal parameters. An extension would be to do this but with interaction via language. Along these lines, Mahowald et al. (2023) distinguish between *formal* and *functional* use of language. Formal competence concerns *knowledge of linguistic rules and patterns* whereas functional competence concerns *understanding and using language in the world* (Mahowald et al., 2023). In their work, they show that large language models are good at the former, but fail the latter. One of their conclusions is that *[..] a model that excels at real-life language use would need to be an AGI, and argue that a human-like AGI cannot be reached simply by getting really good at predicting upcoming words.* This supports many of the points made here.

This thesis takes the position that bridging the gap between visual input and abstract concepts consists of three parts;

1. instantiation,
2. compositional learning, and

### 3. grounding.

Grounding is the process of establishing a mapping between visual element(s) and abstract concept(s). A system can use this mapping in resolving which of the abstract concepts are seen in a specific image, thus instantiating e.g. a logical expression with visual variables. Compositional learning is the mechanism of building new concepts using previous knowledge as the building blocks. These three processes allow for generalisation beyond the initial domain defined.

We can take an example of the expression  $p(x) \wedge q(x) \wedge z(y)$ . In the previous definition, grounding translates into understanding that  $z(y)$  maps to  $y$  being a cylinder. Instantiation could be mapping  $p(x)$  to a specific object in a given image. Compositional learning can be learning the higher-order relation  $R(x, y) \rightarrow p(x) \wedge q(x) \wedge z(y)$  so that the system can instantiate the relation  $R$  in images going forward. This incremental learning The difference to e.g. answer-set programming is that these concepts are not derived by the system to exhausting, even if there is also room for that within these three processes. Rather, compositional learning constitutes having a teacher naming the relation  $R$ , and the system connecting the previous knowledge in composing this new piece of knowledge, rather than learning it as an atomic concept.

## TODO Fodor - What are concepts?

Fodor claims that *Connectionists are committed, willy-nilly, to all mental representations being primitive; hence their well-known problems with systematicity, productivity, and the like.* (J. A. Fodor, 1998). Instead of constituent structures of mental representations, the claim is that neural networks only have primitive, or atomic, representations of concepts. However, this is easily refuted by looking at works such as word2vec where adding or removing properties of a given concept results in another concept representing this change. While there is certainly merit to Fodors claim, the many scribe to the idea of foundation models and that neural networks are capable of mental representations other than those primitive. It is important to clarify that this does not entail that neural networks get this for free, but rather that it is possible to train them to attain such representations. As one example, work by Lovering and Pavlick (2022) and Pavlick (2022) shows that neural networks indeed can exhibit the systematicity of compositionality. This is by no means clear without detailed inspection of the internal workings of neural network models.

- Representational theory of mind
- *It's a general truth that if you know what an  $X$  is, then you should also know what it is to have an  $X$ .*
- Unclear theory of what concepts are, could be argued for to not build systems around such flawed theories

- *Maybe having the concept X comes to something like 'being reliably able to recognize Xs and/or being reliably able to draw sound inferences about Xness'. p. 3*
- *[a] theory of meaning must answer 'What is it to understand a language?'*
- There is the Idea DOG. It is satisfied by all and only dogs, and it has associative-cum-causal relations to, for example, the Idea CAT.
- RTM tolerates thought without language
- Hume suggests mental representations are images
- *Connectionists are committed, willy-nilly, to all mental representations being primitive; hence their well-known problems with systematicity, productivity, and the like.*
- Mental representations have constituent (part/whole) structures

## 2.1 Clever Hans

This historical example of the 19th century horse Clever Hans is relevant for two reasons. First, it illustrates one of the fundamental challenges of neural networks and our evaluation of their behaviour. Secondly, it is the inspiration for the name of the CLEVR dataset which is used as the basis the experiments in Chapter 5 and 6 (Johnson et al., 2017). Clever Hans is often referred to as "the horse that could count", and is an interesting case in both the study of animal intelligence and the history of psychology. Similar to ELIZA, this point of this example is often referred to as the *Clever Hans effect*.

At the turn of the 20th century in Germany, the retired mathematics school teacher Wilhelm von Osten lived alone in Berlin with his carriage horse Hans. While observing how Hans would navigate the streets of Berlin, von Osten was convinced that his horse was capable of conscious thought. He embarked on the project of teaching Hans to think, learning abstract cognitive reasoning. Hans would learn to pull the cart without reins, and could count up to five by stamping his hoofs. When Hans died, Von Osten was convinced that he could teach again and bought another horse. Naming it Clever Hans, he set out to train him in the same way. After a few years of training, Von Osten successfully demonstrated Clever Hans' ability to be able to solve mathematical problems, understand the German language, and even identify musical tones, merely by tapping his hoof. Clever Hans was even capable of counting the number of straw hats in the audience, subtracting the number of hats worn by women. Showcasing this wide range of skills to the public, Wilhelm von Osten would tour Germany with Clever Hans while charging a significant amount of money. The cognitive capabilities of Clever Hans attention far and wide, with *The New York Times* reporting (Heyn, 1904) that

Hans is an expert in numbers, even being able to figure fractions. He answers correctly the number of 4s in 8, in 16, in 30&. When asked how many 3s there are in 7 he stamps down his foot twice and for the fraction once. Then, when 5 and 9 are written under each other on the blackboard and he is asked to add the sum, he answers correctly.

In the same article, the German Professor Karl Möbius, pioneer in the field of ecology and the director of the Natural History Museum in Berlin, had this to say about Hans:

“He possesses the ability to see sharply, to distinguish mental impressions from each other, to retain them in his memory, and to utter them by his hoof language.”

Regardless of the endorsement by distinguished scientists, skeptics questioned the legitimacy of Hans’s abilities. This would prompt a formal investigation lead by the German board of education, known as the *Hans Commission* (untersucht den Klugen Hans, 2006). The investigation initially concluded that there were no tricks behind the performance of Clever Hans. However, the German biologist and psychologist Oskar Pfungst was tasked with further investigation. Pfungst’s meticulous experiments revealed that Hans was not genuinely performing cognitive tasks, but was instead picking up on unconscious cues from von Osten and other observers (Pfungst, 1911). Pfungst constructed four experiments;

- Isolating horse and questioner from spectators, so no cues could come from them
- Using questioners other than the horse’s master
- By means of blinders, varying whether the horse could see the questioner
- Varying whether the questioner knew the answer to the question in advance.

When the horse could not see its trainer or when the trainer himself did not know the answer, Hans’s accuracy would plummet. The phenomenon, now called the “*Clever Hans effect*”, refers to the role of subtle, unintentional cues in human-animal interactions and has had profound implications in the fields of animal cognition research and experimental design.

The Clever Hans effect has since been observed in a range of human-animal interactions. Dog owners might recognise this effect when teaching their dog new tricks, where a dog might pick up on the tone of voice rather than the words spoken. An exercise to readers with dogs is to use a command with a completely different tone of voice, and observe the confusion. Further, Lit et al. (2011) investigates the Clever Hans effect in the training of working dogs and their ability to detect drugs or explosives. Similar to the experiments of

Pfungst, they set up experiments where the handler was told the location of a certain substance. By telling some of the handlers false information, Lit et al. showed that the handler would influence the dog through subconscious signals such as body language or odor. Schmidjell et al. (2012) show similar results for domestic dogs.

### **2.1.1 The Clever Hans Effect in Modern Machine Learning**

Clever Hans is a good lesson in how observation of behaviour alone is not sufficient to draw strong conclusions. Not even experts on animal behaviour and psychology initially managed to uncover the underlying mechanism employed by Hans. Looking back at the quote by Professor Möbius, we begin to see the resemblance to the narrative around modern neural networks. Bender and Koller (2020) highlight the issue with how researchers talk about neural networks *understanding* language. Similar to how Hans picked up on a signal that gave away the answer, there have been countless examples of the same phenomenon with neural networks. The introduction to this thesis used the example of machine vision models picking up on the snow in images of wolves. There is no doubt that the spurious correlations that neural networks learn capture natural language quite well by now, but in order to not fall into the same trap as with Clever Hans we need to carefully evaluate methods based not only on observations of behaviour.





## Chapter 3

# Literature Review of Current Challenges/Opportunities in NLP /With LLMs - working title

Example of how to use quotes at the beginning of chapters

---

*dali*

The introduction to this thesis outlines *resources, robustness, and reasoning* as three areas where current methods of AI fall short. This chapter outlines relevant literature and background for *multimodal machine learning, compositionality, and neuro-symbolic AI*, and how each of them relate to these three areas of challenges.

### 3.1 Language Modeling

Section on the current approaches to language modeling, including a time line and important methods

#### 3.1.1 Brief history of language modelling

Traditional methods, Word2vec (Mikolov et al., 2013b), GloVe (Pennington et al., 2014), BERT, GPT-X

### 3.1.2 Foundation Models

In the comprehensive overview of neural network based models, or *foundation models* as the authors name them, (Bommasani et al., 2021) outlines many of the properties and challenges of foundation models. In this section, we summarise the ones related to models in multimodal and/or reasoning contexts. Finally, we argue why *foundation models* is a poor name and why we should focus on foundational properties instead.

### 3.1.3 Representations of Meaning – Revisiting Fodor?

There are many approaches to model meaning, both implicitly and explicitly. Continuous representations, such as word or sentence embeddings, approximate meaning by closeness in a continuous vector space. Two words that are close indicate that they have some shared semantics. There are also representations of meaning based on the embedding of graph structures, encoding knowledge into a vector space with a similar semantic connotation. Discrete representations of meaning take many different forms. There are graph-based approaches, such as Abstract Meaning Representation (AMR) (Banarescu et al., 2013) and more recently BabelNet Meaning Representation (BMR) (Navigli et al., 2022).

TODO Move to introduction? Discrete vs. Continuous vs. probabilistic but discrete concepts? Easy with discrete for toy problems (e.g. CLEVR) to build vector representations where one bit represents the presence of a feature. However, to scale things, continuous representations have been key. Most of the time, features or properties are not discrete but rather on a scale. Even for toy problems this is the case, e.g. as the color blue is not one single color or not covering the entire object. Probabilistic based discrete representations with an intensity associate with each feature is more flexible in describing the real world.

Representing meaning is a multifaceted problem where not only do we need to model the concept behind a word, e.g. the action connected to it, but also model the intentions of other agents and the state space in which we act. A model that only does the first will be inadequate. We quickly see that modeling this complexity with discrete concepts is a daunting task.

### 3.1.4 Difference between language model and world model

Many of the problems of large language models are rarely that they did not produce human-like language, but that the language did not match our own world model. For instance, telling you to saw a door in half instead of opening the door is not wrong language-wise, but might be socially awkward. There is also an increasing body of work on using language models as knowledge bases (Petroni et al., 2019a), erasing the separation. A mechanism for aligning world views, and adjusting the inner representation of a large language model to adhere can be a useful tool going forward. It is also important to distinguish between adhering to the world model, and generating human-like language that

does not. For example, a language model could be prompted to converse about an alternate reality or generate a made-up story. In this scenario, some aspects might be coherent with our world models, but someone completely unthinkable (e.g. long-since dead) might be the president of a country. One alternative is the integration with knowledge graphs, as this could provide a "fixedness" in separating out a world model from the language model. Knowledge graphs are also clearly compatible with reasoning frameworks.

### 3.1.5 Mechanism for introducing new symbols

Many neuro-symbolic methods rely on a given set of symbols to manipulate, such as the logic program language of NS-VQA (Yi et al., 2018) that they learn the meaning of. However, generalising to learning new symbols is less emphasised. A generic mechanism for introducing new symbols via human-computer interaction would equip a wide range of methods with this capability. Thus, such a mechanism could bridge the gap between neuro-symbolic and neural approaches in their abilities to learn new concepts.

The Foucault effect; there is no meaning without discourse (Foucault, 1991). The discourse is a socially constructed limitation on what is sayable, what should be conserved, remembered, reactivated, or appropriated. Foucault argues that in the governing of social systems, we cannot talk about meaning detached from the discourse. A perfectly valid action in the 1800s can be totally unacceptable today. Similarly, language that carried great meaning before, might have lost that meaning today.

## 3.2 Multimodality

Our interaction with the world consists of multiple senses. We smell, touch, hear, and see things to make sense of the world. Each sense represents a different perspective of an observation or an event. These channels of information are known as different modalities. This section should give an overview of multimodal machine learning, outlining the different kinds of approaches to the problem.

Liang et al. (2022) gives an overview of the principles, challenges, and open questions of multimodal machine learning. (Uppal et al., 2022) gives a similar survey for language and vision. According to the authors, "we are now closer than ever to achieving intelligent agents that can integrate and learn from many sensory modalities". They identify 6 key technical challenges; *Representation*, *Alignment*, *Reasoning*, *Generation*, *Transference*, and *Quantification*. This thesis covers topics of Representation, Alignment, partially Reasoning, and Quantification. The authors define multimodal as *[.] the computational study of heterogeneous and interconnected modalities*. These two core principles, heterogeneity and interconnectedness are further split up into several dimensions.

The dimensions of heterogeneity are *element representation*, *distribution*, *structure*, *information*, *noise*, and *relevance*. Information covers the fact that different modalities have different levels of information density under different circumstances. For instance, dark footage contains less information than bright, but certain objects scene might carry heavy weight in relation to the uninformative speech heard in a recording. This thesis is mainly concerned with element representation, structure, and information.

The second principle, interconnectedness, can be split into modality connections and modality interactions. The connections can be either statistical or semantic. The statistical connections can be of associative or dependent nature as, e.g., correlations found by deep learning methods, or temporal or causal dependencies. The semantic connections concern correspondence such as those of explicit grounding, or relationships between higher-level concepts such as hypernyms.

Multimodality has been a loosely defined term used to describe research on methods for heterogeneous data. In (Parcalabescu et al., 2021), the authors argue from this insight that an explicit definition is needed. They outline the difference between human-centric and machine-centric definitions previously used, and how they both have shortcomings. The authors instead consider multimodality defined relative the task itself, whether the model or the data representation is heterogeneous, and the complementary aspects of the data used.

- VALSE, task-independent benchmark for vision and language centered on linguistic phenomena (Parcalabescu et al., 2022)
- Neural natural language generation survey (Erdem et al., 2022)
- Visually grounded transformers, such as Vilbert (J. Lu et al., 2019).
- Learning visually grounded sentence representations (Kiela et al., 2018), perhaps move to ??
- Arguments for why imagination is a useful concept to model for multimodality (Elliott & Kádár, 2017).
- (Ross, 2022)
- (Bruni et al., 2014)
- Also, characterise language-for-vision vs. vision-for-language vs. language-and-vision (Frank et al., 2021).

In transformer-based language models working only on text, words are grounded by the company they keep. Summarise arguments by Bisk (Bisk et al., 2020)

Figure 3.1: TODO Illustration of VQA tasks

### 3.2.1 Visual Question Answering (VQA) and Visual Reasoning - Datasets

One of the first VQA datasets proposed was the DAQUAR dataset (Malinowski & Fritz, 2014) based on real images of indoor scenes. VQA is another widely used dataset (Antol et al., 2015) with images from MS-COCO dataset (T.-Y. Lin et al., 2014). Questions are manually created and answering these require commonsense knowledge and reasoning. The CLEVR dataset (Johnson et al., 2017) is based on automatically generated scenes and questions, giving great control over the distribution of instances. 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.

Closely related is the CLEVRER (Collision Events for Video Representation and Reasoning) dataset (Yi et al., 2020) and CLEVR-Hyp dataset (`sampat2021clevr_hyp`). The questions on videos in CLEVRER requires reasoning about the state of objects after an video event, instead of after actions in text as in CLEVR-Math. CLEVR-Hyp focus on VQA where reasoning about effects of actions, and 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 & Manning, 2019). Graph learning is a heavily studied area, with applications in multimodal domains such as robotics (J. Ji et al., 2020; Wald et al., 2020; Xia et al., 2021; Yu et al., 2021).

Experiments with Kandinsky patterns (Holzinger et al., 2019) show that neural networks are easily confounded by visual reasoning tasks with shapes, colors, and patterns that can be difficult to distinguish but follow clear rules. The Winoground dataset (Thrush et al., 2022) shows similar results, where no state-of-the-art visual reasoning method is able to distinguish between two confounding captions and images.

**Existing Approaches to VQA** Most of the earlier approaches in VQA were based on purely neural models that first encoded the two inputs - the image and the accompanying question into embeddings using networks like Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) networks and then the two embeddings were forwarded to a classifier that would then predict the answer to the question ((Ben-Younes et al., 2017), (Fukui et al., 2016a)). Another category of approaches are the attention mechanism-based approaches

that identified the regions in the image that were relevant to answering the associated question ((P. Wang et al., 2017), (Shih et al., 2016)). Graph neural networks (Narasimhan et al., 2018) have also been applied in VQA where both text and the image are represented as graphs and a multi-modal vectorial representation is learned that captures the alignment of nodes in the two graphs (Radford, Kim, Hallacy, Ramesh, Goh, Agarwal, Sastry, Askell, Mishkin, Clark, et al., 2021b) introduced the CLIP models where a representation of the image is learned with natural language supervision by leveraging the already available huge datasets for image captioning. More recently, neuro-symbolic approaches have been used in addressing the task of VQA like Neuro Symbolic Concept Learner (NSCL) (Mao et al., 2019a) and Neuro-Symbolic Visual Question Answering (NS-VQA) (Yi et al., 2018). These approaches convert the input image and text into an intermediate semantic representation and then employ a quasi-symbolic program executor to derive an answer from these semantic forms.

**Multimodal Reasoning** Multimodal reasoning, such as visual question answering (VQA), concerns extracting knowledge from heterogeneous data, such as images and text, leverage cross-modal interactions, and combining it in one or more steps to infer new knowledge or make high level predictions (Baltrušaitis et al., 2019; Liang et al., 2022). VQA is a popular task with many datasets available, ranging from synthetic 3D scenes to multimodal science questions (Antol et al., 2015; Hudson & Manning, 2019; Johnson et al., 2017; Krishna et al., 2017; P. Lu et al., 2022). GQA (Hudson & Manning, 2019) uses real world images annotated with rich scene graphs and a large set of relations and attributes, focusing on compositionality in visual reasoning. However, such real world datasets does not give the same flexibility to create compositional splits. Johnson et al. (2017) introduce CLEVR as synthetic dataset to benchmark compositional multimodal reasoning, using 3D scenes rendered with Blender and a template engine to generate questions based on the structural representations of the visual scenes. The synthetic nature allows us to study various aspects like compositional generalisation of systems, given the high degree of control to generate a specific combination of objects (e.g., only red cubes and blue cylinders). The compositional generalization splits in CLEVR are limited to two splits restricting certain attribute compositions, and several benchmarks have built on CLEVR to study various aspects of visual question answering (**sampat2021clevr\_hyp**; Arras et al., 2022; Kottur et al., 2019; Z. Li et al., 2022; R. Liu et al., 2019b; Salewski et al., 2022; Stammer et al., 2021). The questions on videos in CLEVRER (Yi et al., 2020) require reasoning about object states after an video event, instead of after actions in text as in CLEVR-Math. CLEVR-Hyp (**sampat2021clevr\_hyp**) focus on reasoning about effects of actions, whereas CLEVR-Math (Lindström & Abraham, 2022b) introduces an additional mathematical dimension to the problem.

### 3.3 Compositionality

In the introduction we discussed how generalising from experience is a key characteristic of human intelligence. An important component of this general intelligence is *compositionality* – combining elements in unseen ways to create new meaning. Without compositionality we would have to memorise all possible, or at least all useful, combinations of words. Natural language is a good example of a compositional system, where we put together words and sentences to communicate completely new ideas and events. Stephen Fry eloquently demonstrates this in Episode 3 of the first season of *A Bit of Fry and Laurie*;

Imagine a piano keyboard, eh, 88 keys, only 88 and yet, and yet, hundreds of new melodies, new tunes, new harmonies are being composed upon hundreds of different keyboards every day in Dorset alone. Our language, tiger, our language: hundreds of thousands of available words, frillions of legitimate new ideas, so that I can say the following sentence and be utterly sure that nobody has ever said it before in the history of human communication: “*Hold the newsreader’s nose squarely, waiter, or friendly milk will countermand my trousers.*” Perfectly ordinary words, but never before put in that precise order. A unique child delivered of a unique mother.

Pelletier (1994) considers Argument 3.3.1 and 3.3.2 (below) to be the strongest ones for why the principle of compositionality should be considered a desideratum for linguistic systems.

**Argument 3.3.1** (Unlearnable Language). *If a language lacked compositionality it would be unlearnable*

**Argument 3.3.2** (Infinite Understanding). *Compositionality is the only explanation of how a finite mechanism (such as the human brain/mind) can understand an infinite set of sentences. (Without compositionality, novel utterances would be non-understandable.)*

Naturally, compositionality is a characteristic of intelligence that we would like to replicate in systems of artificial intelligence. In the machine learning community, compositionality of individual words is one of the reasons why distributional semantic representations such as `word2vec` saw such success. `word2vec` allows for arithmetic operations on concepts to compose e.g. *capital* and *Sweden* to represent *Stockholm*. This capability was often used as an example of the strength of the method, and was one important improvement at the start of the current deep learning-wave. We will study how this translates to current research throughout the rest of this section.

Szabó (2012) gives the following definition of the *principle of compositionality*:

*The meaning of a complex expression is determined by its structure and the meanings of its constituents.*

Another common definition comes from Partee et al. (1984):

*The meaning of a compound expression is a function of the meanings of its parts and of the way they are syntactically combined.*

The definition given by Partee et al. (1984) uses a more mathematical language to describe the principle, suggesting the existence of an evaluation function. Neither of these definitions address *pragmatics*, i.e. considering the semantics but in a given *context*. The principle of compositionality is often attributed to Gottlob Frege as *Frege's Principle* for its introduction in his work on *The Foundations of Arithmetic* (Gottlob & Austin, 1884). The earliest record going back to 4th or 5th century India with the less formal definition “[T]he meaning of a sentence is based on the meaning of the words.”, by the Indian philosopher Śabara (Pagin, 2003).

Now, we will consider an example that is relevant to the work presented later in this thesis. Consider the sentence “A blue sphere.” referring to multiple concepts (color, shape) that we use to understand the sentence. Without a compositional understanding of these concepts, we might consider *blue sphere* as a single concept, rather than the composition of two attributes of an object. In doing so, we would not recognise the common shape of red spheres and blue spheres. However, *meaning* is a central component of the above definitions of compositionality that is left vague. For instance, a blue sphere might refer to Planet Earth in some contexts, and a billiard ball in others. G. F. Marcus (2003) gives a definition of compositionality that is closer to requirements on an implementation:

- Stable encodings of individual elements
- An operation that concatenates pieces of trees together
  - or disassembles wholes into parts
- Iterative process for (de)constructing larger structures
- Representational formats for trees (or something very similar).

While an attempt at concretising compositionality to provide more of a blue print, there are still vague elements in this list of requirements. It does not specify under which conditions encodings should be *stable*, nor what stable means. Stable encodings of individual words is easily achieved in modern architectures, but if the context should be considered the question becomes trickier. A general critique of the principle of compositionality is precisely this type of vagueness. Structure, meaning, and context are all components that require formal definitions. After many attempts, so far there is not one all-encompassing definition. However, our purpose does not require a definition that covers all. Instead,



we will focus on the aspects of compositionality sufficient for understanding the contributions in later chapters. More specifically, we will look at three traits associated with compositionality; systematicity, productivity, and substitutivity. After describing and defining these traits, we will look at how we can measure compositionality and how well current machine learning methods generalise compositionally.

### 3.3.1 Defining Compositional Generalisation

J. A. Fodor and Pylyshyn (1988) define compositionality as

The ability to produce/understand some sentence is intrinsically connected to the ability to produce/understand certain others... [they] must be *made of the same parts*.

In contrast, Kamp and Partee (1995) give a more lenient definition of *the meaning of a sentence is a function of the meaning of its words and the way in which they are combined*. The later definition does not say anything about how similar parts are evaluated in different inputs, only that meaning is derived from the words themselves in combination with their ordering. However, for a compositional system, understanding that constituent parts can be reused across different inputs is a key property of building an efficient system. The hypothesis is that this is a criterium for achieving compositional generalisation.

**Definition 3.3.1** (Principle of Compositionality). Given language  $L$ , for every complex expression  $e$  in  $L$  and every context  $c$ , the occasion meaning of  $e$  in  $L$  at  $c$  is determined by the structure of  $e$  in  $L$  and the occasion meanings of the constituents of  $e$  in  $L$  at  $c$ .

Definition 3.3.1 shows a definition of the *principle of compositionality* given by Szabó. A key issue that is not addressed in this definition is how we can determine the *meaning* of  $e$ . Montague (1970) gives formal definitions to clarify these aspects as a part of Montagues’ work on pragmatics, where the key idea is that compositionality requires a *homomorphism* between expressions and their meaning. The formal discussion contains more nuance than what is captured in Definition 3.3.1, and we refer to, e.g., (Szabó, 2022) for a more in-depth discussion.

**Definition 3.3.2** (Productivity). Let us consider a finite vocabulary  $V$  and a finite set of rules  $R$ . Productivity is defined as the ability to generate a potentially infinite set of sentences  $S$  using  $V$  and  $R$ , mathematically represented as:

$$S = \{s_1, s_2, s_3, \dots\} \text{ such that } s_i = f_R(v_1, v_2, \dots, v_n), \text{ for } v_j \in V,$$

where  $f_R$  represents the application of rules in  $R$  to construct sentences using elements from  $V$ .

Productivity thus refers to the ability of a language system to create a potentially infinite number of sentences or expressions using a finite set of rules and vocabulary. In the context of compositional generalization, it would imply the capacity of the model to generate novel compositions by combining known components in different arrangements and structures. The productivity of a language model can be measured by its ability to generate coherent, novel sentences or structures using learned compositional rules.

**Definition 3.3.3** (Systematicity). Consider a function  $g$  that applies a rule  $r$  from set  $R$  to a series of inputs  $x_1, x_2, \dots, x_n$  to produce outputs  $y_1, y_2, \dots, y_n$ :

$$g(r, x_i) = y_i \text{ for } i = 1, 2, \dots, n$$

If the function  $g$  is systematic, it should be able to apply the rule  $r$  consistently across different inputs, maintaining the relationship  $r$  across varied contexts.

Systematicity involves the consistent application of rules across different contexts or components within a language. It implies that if a model has learned a particular rule or pattern in one context, it should be able to apply that learned knowledge to understand or generate sentences in different but structurally similar contexts. In other words, systematicity represents the ability of a system to generalize learned patterns or rules to novel, yet structurally similar, scenarios. This aspect is crucial in compositional generalization as it ensures that the rules and patterns learned from the training data can be generalized to new, unseen data, ensuring consistent performance across a range of tasks.

**Definition 3.3.4** (Substitutivity). Given a sentence represented by a function  $h$  which takes inputs  $x_1, x_2, \dots, x_n$  (representing components like words or phrases), substitutivity means that replacing one of the inputs with another equivalent input does not change the truth value or grammatical correctness of the sentence:

$$h(x_1, x_2, \dots, x_i, \dots, x_n) = h(x_1, x_2, \dots, x_j, \dots, x_n) \text{ if } x_i \sim x_j$$

Here,  $\sim$  denotes an equivalence relation indicating that  $x_i$  and  $x_j$  can be substituted for each other without altering the grammatical or semantic properties of the sentence represented by  $h$ .

Mathematically, substitutivity can be defined using equivalence relations. Substitutivity is a principle that posits that in certain grammatical contexts, one expression can be substituted for another without changing the truth value or grammatical correctness of the sentence. In the realm of compositional generalization, substitutivity would involve the model's ability to recognize and execute valid substitutions of components (like words or phrases) in sentences, allowing for the generation or understanding of new sentences that retain the grammatical structure and meaning of the original sentence. It is a key aspect

of compositional operations where known components can be replaced with others to create meaningfully different expressions.

In essence, a model with good compositional generalization capabilities would be able to effectively demonstrate productivity, systematicity, and substitutivity in its operations, showcasing the ability to generate novel, grammatically correct, and meaningful sentences by leveraging learned rules and patterns across different contexts and components.

### 3.3.2 Measuring Compositionality

The principle of compositionality stands on three legs; how meaning is evaluated, structure, and the parts or symbols in the system. One way to evaluate whether a system fulfils the principle is to prove that the internal properties of the system satisfy the principle. For example, we can define a purely symbolic system where these three components are clear, such as mathematical logic frameworks. In this domain, logical rules and their application over atoms are well-defined, and we can a logical expression such that the meaning is dependent on the structure and the meaning of the parts. This *functional* compositionality corresponds well to the definitions given earlier in this section. However, natural language and its meaning in the real world does not afford us with the same well-defined components. When we talk about a *bank*, we use the context to disambiguate between a *river bank* and *financial bank*. This type of interdependence between words results in combinatorial explosions in symbolic systems. So, while a system might satisfy the principle of compositionality, that does not mean that it measures up to the semantic compositionality required for natural language. The literature has instead turned to benchmarks for *compositional generalization* in order to evaluate compositionality in linguistic systems. These benchmarks are constructed by holding out certain compositions of symbols in the training data, and testing whether a model can still evaluate a novel composition of symbols correctly during testing. If a system is able to compose previously seen parts into unseen combinations, we say that it generalizes compositionally. For visual question answering, we can withhold all blue cars from the training data but keep blue buses, and test whether the model can answer questions about blue cars without ever seeing them before. Evaluating the external behaviour on such benchmarks also makes the evaluation model-agnostic.

Recently, with the ubiquity of neural networks, there have been a number of benchmarks proposed to study compositional generalization, including SCAN (B. Lake & Baroni, 2018a; Loula et al., 2018), COGS (N. Kim & Linzen, 2020) and PCFG (Hupkes et al., 2020b; Ruis et al., 2020). Several of these papers have shown that end-to-end neural networks are not able to compositionally generalize, especially in few-shot regimes. While it might be possible to see all useful structures given enough data, humans perform structural generalisation with far less data (Linzen, 2020). Several approaches show that neural models that are made aware of the problem structure can do structural

generalization (Qiu et al., 2022; Weißenhorn et al., 2022).

This section will cover ways compositionality can be measured in machine learning models through compositional generalization benchmarking. Table 3.1 gives an overview of such benchmarks. Compositional generalisation benchmarks highlight weaknesses in machine learning methods, and are useful for guiding us in constructing better methods. However, they are not without weaknesses themselves. A common critique is that synthetic data often used means that the benchmarks carry little ecological validity compared to benchmarks based on natural/real-life data. While there is merit to this critique, we argue that synthetic data is useful when exploring capabilities and for augmenting real data. Instead, 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. We will later use this to inform our own compositional generalisation benchmark. For further details on how to measure compositionality, see (Andreas et al., 2019; Chaabouni et al., 2020; S. Xie et al., 2022).

<b>Dataset</b>	<b>Task</b>	<b>Modality</b>	<b>Year</b>
COGS			
<i>ReCOGs</i>			
SCAN			
<i>gSCAN</i>			
<i>ReaSCAN</i>			
CLEVR			
<i>CLEVR-HANS</i>			
<i>CLEVR-ref+</i>			
<i>CLEVR-dialog</i>			
<i>CLEVR_HYP</i>			
<i>CLEVR-XAI</i>			
<i>CLEVR-x</i>			
<i>Super-CLEVR</i>			
CFQ			
GQA			
NLVR			
CLOSURE			
CLUTTR			
Math (Saxton et al., 2019)			

Table 3.1: Compositional generalisation benchmarks

**Formal Languages and Compositionality Benchmarks** A common feature of all the benchmarks described in this section is that they all utilise some formal language to generate the data. Often a context-free grammar is used to define the domain and generate samples, affording transparency and control over the distribution of data. An intermediate logical form is also common,

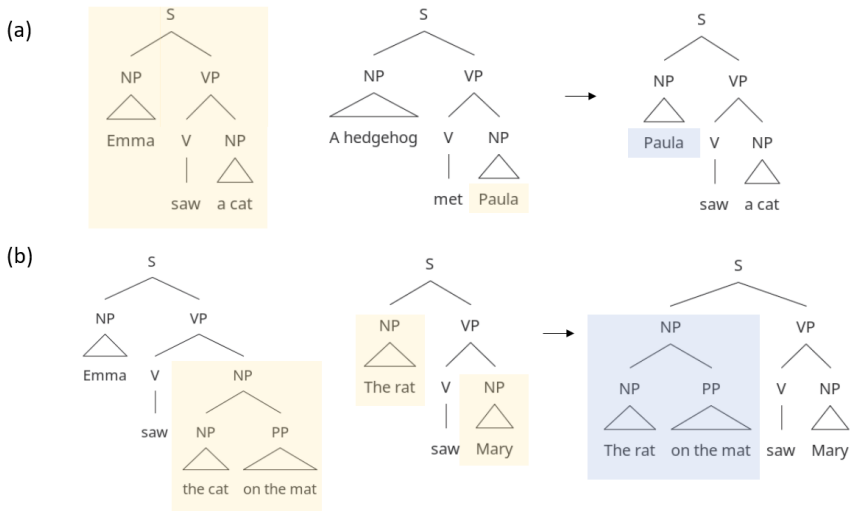


Figure 3.2: Illustration of (a) *lexical* and (b) *structural* generalisation in COGS. TODO ask permission or recreate.

used either to hold out samples from training data, or as the target form to learn for e.g. semantic parsing. The usage of these formal languages allows us to produce well-defined synthetic data with clear distributions of features. This can be used to generate completely synthetic data, as with e.g. *SCAN*, or partially synthetic data as with *CFQ*.

**COGS (N. Kim & Linzen, 2020)** N. Kim and Linzen (2020) introduce COGS as a compositional generalisation challenge based on semantic interpretation of sentences. COGS is generated using a Probabilistic Context-Free Grammar (PCFG) such that there are systemic gaps that can only be filled if a model has learnt concepts compositionally. The task is then to parse each sentence into the simplified logical form representing each sentence. For example, the training data can contain the following two sentences: *The cat loves the girl*, and *The hedgehog sees the cat*. To test whether a model can use these concepts to compose and understand novel sentences, we can give it the sentence *The boy loves the hedgehog*. N. Kim and Linzen (2020) further characterises the different generalisation cases into *lexical* and *structural* generalisation. Lexical generalisation means using a known word in a new context, whereas structural generalisation means creating a new combination of familiar structures. When looking at dependency trees, lexical generalisation means exchanging the word at a leaf node with a word of equivalent word class. In structural generalisation, an entire sub-tree is exchanged with another subtree previously seen. Figure 3.2 illustrates the two cases. N. Kim and Linzen (2020) show

that Transformers and LSTMs can learn the semantic interpretation task with near-perfect accuracy in-distribution, but generalise poorly. They note that both models perform better on lexical generalisation than on structural, with close to zero accuracy on the structural generalisation cases and up to around 40% on cases requiring lexical generalisation.

Z. Wu et al. (2023) identify a set of issues with COGS, addressing them in ReCOGS. Their general claim is that compositional generalisation benchmarks operationalise meaning with *logical forms* that limit the possible meanings that would be acceptable to one particular representation and meaning. The semantically irrelevant details and design choices of a given logical form lead to a misrepresentation of the compositional generalisation capabilities of models. In a sense, this means that good performance on a compositional generalisation benchmark indicates that a model has this capability in general, but poor performance does not necessarily indicate the opposite. For COGS, the authors identify two main factors contributing to this; redundant symbols in the logical form of COGS, and the requirement that models predict exact numerical values when binding variables. Z. Wu et al. (2023) argue that “[t]hese details cannot be justified semantically, and they play a large role in shaping model performance”. Their relaxations and modifications to the logical form used by COGS consistently allows for better compositional generalisation in LSTM and Transformer baselines. They construct ReCOGS based on these modifications and insights to better assess semantic capabilities of models. They also show how ReCOGS is harder than the original COGS, while also being more true to semantics.

**SCAN** B. Lake and Baroni (2018a) introduce *SCAN* (**S**implified versions of the **CommAI** **N**avigation) to show 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 in-distribution split.

Qiu et al. (2021) 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. While processing more complex language is a natural extension of the gSCAN dataset, another conclusion is that there is room for a more visually complex dataset. In the proposed task “*the agent needs to reason about spatial relations between objects expressed in 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.

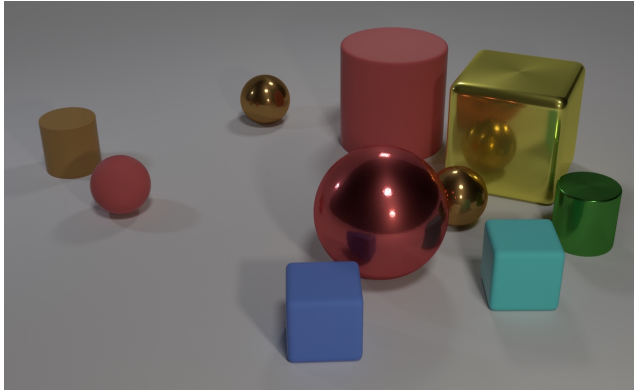
In the same spirit, Z. Wu et al. (2021) identify a set of limitations in gSCAN which they address with ReaSCAN. They remark that the ideas gSCAN build on are powerful, but that there are some central limitations coming from specific design choices. The first observation is that the word order of commands does not matter for the defined tasks. A simple bag-of-word model is sufficient to

encode the original gSCAN commands. The second observation is that there is limited testing for linguistic compositionality. The third observation is that the distractors (i.e., objects added to distract or confuse a model) are not correctly sampled, leading to by-chance accuracy that is dependent on the language used to describe objects in a task. The *last observation* is that there are simply too few distractors, and that the ones used have little to no impact on the output action sequence.

Sikarwar et al. (2022) extends the work by Qiu et al. (2021) 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.

**CLEVR (Johnson et al., 2017)** Introduced in 2017, the CLEVR dataset is a large-scale synthetic dataset for *Compositional Language and Elementary Visual Reasoning* (CLEVR). The dataset is named after the famous German show horse from the turn of the 19th century, Clever Hans (Pfungst, 1911). It consists of over 100,000 images of 3D objects in a variety of scenes, along with questions and answers about the images. Associated with each image is a structured scene representation and functional programs representing the questions. The questions are designed to test a variety of visual reasoning skills, such as recognizing objects, counting, comparing sizes, and understanding relationships between objects. Figure 1.1 shows an example image and illustrates some of the visual reasoning tasks. Johnson et al. (2017) evaluate several neural methods, mainly LSTM+CNN-based approaches, showing that they struggle with the tasks. They also introduce a compositional generalisation split, *CoGenT*, where spheres and cubes do not have the same color palettes in training. The color palettes are then swapped during testing so that the models are presented with unseen color-shape combinations, such as red cubes. Spheres are assigned all colors in both training and testing, used as control and an intermediary for the models to see that (at least some) shapes can be of all colors. The evaluation shows that the models more strongly associate colors with shapes than, e.g., colors with materials.

Since the introduction of CLEVR, many follow-up datasets have been proposed through various extensions to the templates and code base used to generate the original data. While not all of them are strictly compositional generalisation benchmarks, they target related concepts. CLEVR-Hans (Stammer et al., 2021) uses the CLEVR dataset to create an image classification problem, and introduces a neuro-symbolic method (NeSy XIL). The data is partitioned into classes such as images with  $c_1 = a \text{ large cube and a large cylinder}$ , split into a 3-class and 7-class partitioning. Each class has a confounding variable



- Q: Are there an **equal number** of **large things** and **metal spheres**?  
 Q: What **size** is the **cylinder that is left** of the **brown metal thing that is left** of the **big sphere**? Q: There is a **sphere** with the **same size** as the **metal cube**; is it **made of the same material** as the **small red sphere**?  
 Q: **How many** objects are **either small cylinders or metal things**?

Figure 3.3: A sample image and questions from CLEVR. Questions test aspects of visual reasoning such as **attribute identification**, **counting**, **comparison**, **multiple attention**, and **logical operations**. TODO ask permission or recreate.

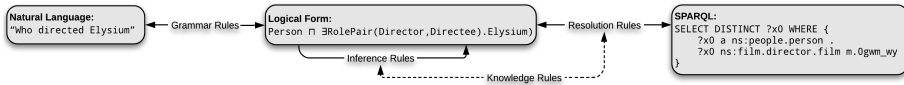


Figure 3.4: Illustration of the relationship between natural language questions, their intermediate logical form, and the corresponding SPARQL query used to extract the answer from Freebase. TODO ask for permission to use figure, or remake.

associated with one of the objects defining the class. This confounding variable is then only present during training as a way to trick models into focusing on object attributes that do not define the class. For instance, the confounding variable in  $c_1$  is that the large cubes are always *gray* in the training data. Stammer et al. (2021) show how regular CNNs are sensitive to the confounding variables, and that their NeSy XIL-method does not face this issue while outperforming previous neuro-symbolic methods (NSCL (Mao et al., 2019b)).

Other examples of CLEVR-based datasets include Clevr-ref+ (R. Liu et al., 2019a), CLEVR\_HYP (Sampat et al., 2021), CLEVR-dialog (Kottur et al., 2019), CLEVR-XAI (Arras et al., 2022), Super-CLEVR (Z. Li et al., 2022), Clevr-x (Salewski et al., 2022).

**CFQ (Keyzers et al., 2020)** The Compositional Freebase Questions (CFQ) dataset is a realistic and large-scale natural language understanding benchmark



designed to measure compositional generalisation. CFQ is based on question-answer pairs extracted from the Freebase knowledge base (Bollacker et al., 2008). The pairs are extracted using rules produced through the *distribution-based compositionality assessment* (DCBA) method also introduced in (Keyzers et al., 2020). Figure 3.4 illustrates how the question-answer pairs are generated from an intermediate logical form representing the results from SPARQL queries over Freebase. An example of training and test data is the question-pair *Did Christopher Nolan produce Goldfinger?* and *Did Christopher Nolan direct Goldfinger?*. With other questions asking who directed, e.g., *Inception*, a compositional model should be able to answer whether Nolan directed Goldfinger since it has seen all the words and structures before but not in this composition. Keyzers et al. (2020) evaluate an LSTM with attention, a regular Transformer (Kobayashi et al., 2020), and a Universal Transformer (Dehghani et al., 2018). All three methods learn to answer questions with near-perfect accuracy when tested in-distribution, but the performance drops to below 20% when tested on the compositional generalisation split.

**Winoground (Thrush et al., 2022)** Levesque et al. (2012) introduce the Winograd schema challenge as an alternative to the Turing test.

The trophy doesn't fit in the brown suitcase because it's too big.  
What is too big? A) the trophy, or B) the suitcase?

Kocijan et al. (2020) outline problems with the assumptions in the original Winograd schema challenge, and some of the approaches used to solve the challenge. They point out that neural language models such as BERT (Kenton & Toutanova, 2019) can solve the challenge, but still fail on tasks related to common sense reasoning. In the multimodal domain, Thrush et al. (2022) proposes a Winograd schema for visio-linguistic reasoning with images and text. Shown in Figure 3.5, the task consists of pairs of images and corresponding descriptions, where the descriptions are swapped versions of each other. Thus, the sentences contain the same symbols, but represent different meanings. Thrush et al. (2022) show that state-of-the-art models in multimodal machine learning fail significantly on this task. Looking back at the principle of compositionality, this entails that these models do not evaluate the language according to the structure. Instead, the results suggests that structure is somewhat ignored in favor of a bag-of-words approach.

**Using Structure** Weïßenhorn et al. (2022) and Qiu et al. (2022) both show that neural models that are made aware of structure can do structural generalisation. Qiu et al. (2022) identify that transformer models can be augmented with synthetic data that is generated from structured methods, in their case quasi-context free grammars. Weïßenhorn et al. (2022) 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 clearly perform



(a) some plants surrounding a lightbulb



(b) a lightbulb surrounding some plants

Figure 3.5: Example taken from (Thrush et al., 2022)

structural generalisation with far less data (Linzen, 2020). Another important concept that can help accelerate a models ability to generalise compositionally is curriculum learning. By inducing a compositional bias early on, the rest of a training process can adhere to that structure much like how human knowledge is built compositionally by learning the building blocks first as in addition before multiplication.

**Compositional Generalisation in Deep Learning** There has been a long-standing debate on whether connectionist architectures like neural networks are able to generalize compositionally (J. A. Fodor & Pylyshyn, 1988). B. M. Lake (2019) introduces a test for systematic generalisation, defined as an algebraic capacity to combine novel sentences from known components. As an example, learning what *jump* means, a human can be asked to *jump twice* without being told what the combination entails given that *twice* is known. This is something neural networks have been shown to fail at, especially for recurrent neural networks (B. Lake & Baroni, 2018b). The dataset introduced by B. M. Lake (2019) uses pseudowords, such as *blicket* and abstract outputs in the form of colored dots, to test this property without dependency on a specific language.

Pavlick (2022) investigate how deep learning models exhibit semantic structures, e.g. by showing how different properties of concepts are learnt at different stages throughout a network (Lovering & Pavlick, 2022). The authors show that concepts are learnt compositionally such that structural concepts with shapes and formations are not affected by the color seen in training and testing. This work aims at answering to which degree neural networks employ systematicity in constructing their understanding of its input data, also

exemplified in (R. Patel & Pavlick, 2021). Pavlick (2022) argues that representations should be continuous and that compositionality does not imply discrete symbols. A discrete set of symbols does not imply compositionality if a new symbol is grounded for each new combination of properties without relating them back to already known components. This line of work shows evidence that the continuous representations of neural networks can exhibit a compositional structure to some degree. This implies that it is possible to learn concepts compositionally with neural networks, but it is still unclear how effective this process is. This also implies that failing on a compositionality test does not imply that the underlying system is not compositional.

Hupkes et al. (2020b) decomposes testing of compositionality into *systematicity*, *productivity*, *substitutivity*, *localism*, and *overgeneralisation*. Similar to our artificial neural effigies, our ability for continuous computing that allows for arbitrary small changes to concepts used in reasoning. This is known as the continuity principle and underlies the advancement of deep learning (Hinton et al., 1986). Smolensky (1988) denotes the trade-off between the continuity and the compositionality principles as the *Central Paradox of Cognition*. In their work on neurocompositional computing, Smolensky et al. (2022) attribute the achievements of convolutional neural networks (CNNs) for vision and transformers for language to their compositionally structured processing of input (J. Henderson, 2020). For CNNs, the convolutional layers impose spatial structures on the computing. Similarly, it is possible to show that transformers are equivalent to graphs with weighted links between symbols, thus imposing a compositional structure of concepts (Dwivedi & Bresson, 2020). Smolensky et al. (2022) argues that architectures that respect both the continuity and compositionality principles will address some of the issues with current AI systems such as their lack of reasoning capabilities. The authors define Tensor-Product Representations (TPRs) and Neurally-Encoded Compositionally-Structured Tensor (NECST) computing as a theoretical framework to show this. Recent approaches, such as the NECSTransformer, build on these concepts (Schlag et al., 2019).

**Compositional Generalisation in Neuro-Symbolic Methods** Work by IBM on compositional generalisation and neuro-symbolic methods (Basu et al., 2021; Ito et al., 2022; Klinger et al., 2020; Riegel et al., 2020; Sen et al., 2022). J. D. Fodor et al. (2013) critiques the “*unreality*” of semantic representations. In discrete representations, compositionality is a property that falls out by definition, but in continuous representations this is much more opaque.

**Importance of Compositional Generalisation** B. Lin et al. (2023) surveys compositional generalisation in applications. They outline seven different application areas; mathematics, control systems, semantic parsing, image captioning, question answering, automatic translation, and recommendation systems. In this thesis, we focus on the compositionality in mathematics, semantic parsing, and question answering. In *mathematics*, the operators must

work compositionally with any novel combination of mathematical elements. The experiments with CLEVR-Math described in Chapter ?? support the claim that a lack of compositional generalisation severely affects the mathematical capabilities of a model. These insights are echoed by e.g. Y. Lan et al. (2022).

Qiu et al. (2022) outline the limitations of model scaling for compositional generalisation, pointing out that, e.g., fine-tuning has a flat or even negative effect. They find that prompt tuning is the most effective way to improve compositional generalisation. For image captioning and question answering, compositional generalisation is central to understand and respond to novel combinations of concepts. However, in both cases, current state of the art methods leave big gaps to improve upon. To summarise, in all the mentioned domains B. Lin et al. (2023) identify recent methods leveraging structured representations to improve performance on compositional generalisation benchmarks.

**Compositionality in human learning** B. M. Lake (2019) introduces a test for systematic generalisation, defined as an algebraic capacity to combine novel sentences from known components. As an example, learning what *jump* means, a human can be asked to *jump twice* without being told what the combination entails given that *twice* is known. This is something neural networks have been shown to fail at, especially for recurrent neural networks (B. Lake & Baroni, 2018b). The dataset introduced by B. M. Lake (2019) uses pseudowords, such as *blicket* and abstract outputs in the form of colored dots, to test this property without dependency on a specific language. This could be extended to the CLEVR domain, by introducing pseudowords for abstract concepts such as three blue cubes being called a *blargh*.

In developmental psychology, assigning new meaning to a new word, rather than as a referent to something previously known, is called mutual exclusivity. It makes up the three components of lexical learning, together with the *taxonomy assumption* and *fast mapping* (Golinkoff et al., 1992). It is important to distinguish between mutual exclusivity and learning a word in a different language. For instance, consider a new word *blargh*. It could be a synonym for sphere, in which learning *blargh* is similar to learning a new language (i.e. anchoring known concepts in a new language space). However, if *blargh* means *blue sphere*, or *3 blue spheres*, then it is a compositional learning task. Building a new concept out of previously known ones like this is a mechanism that we study in Chapter 7.

Compositional bias can be a structural property of an architecture, or an emergent property of the training data or procedure. This is one example of an important innate bias that exists in humans, allowing us to learn under vastly different circumstances.

‘Syntax is an algebra, semantics is an algebra, and meaning is a homomorphism between them’ (Janssen, 1986).

### 3.3.3 Compositionality and Compositional Generalisation

Compositionality is a strong trait of neuro-symbolic architectures, introduced by symbolic components. However, compositional generalization does not follow from a compositional structure.

## 3.4 Neuro-symbolic machine learning - is not a challenge of NLP, is a response - where to put?

Chapter 2 introduced the *symbol grounding problem* as the task of giving meaning to language by connecting it to physical systems and subjective experiences (Harnad, 1990). In introducing his work on the symbol grounding problem, Harnad described what we today would call *neuro-symbolic* systems. Depending on the community and time period, other names include *neuro-explicit*, *learning and reasoning systems*, and *hybrid systems*. According to A. Garcez et al. (2019), neuro-symbolic AI aims to *combine the two most fundamental cognitive abilities: the ability to learn from experience, and the ability to reason from what has been learned*. Often such systems aim at scalable learning and reasoning, where neural methods are better at the former and symbolic methods at the latter. Neuro-symbolic AI has a long tradition, surveyed by A. S. d. Garcez et al. (2002) over twenty years ago and Sun and Bookman (1994) describing the field emerging during the 1990s. The idea predates artificial intelligence as a field by decades, where McCulloch and Pitts (1943) describe a *logical calculus of the ideas immanent in nervous activity*, attributed as the first description of a neuro-symbolic systems. The field has recently gathered mainstream interest, with publications in top conferences on the topic is growing (Hamilton et al., 2022; Sarker et al., 2021). In particular, neuro-symbolic AI is increasingly important to address challenges of safety and interpretability (A. d. Garcez & Lamb, 2023), as a response to the shortcomings or recent deep neural network-based approaches. Symbolic components can be utilised to increase the transparency and verifiability of AI systems, important for industry applications. However, purely symbolic systems struggle with, e.g., the combinatorial explosion of learning and manipulating symbolic representations, complemented by the success of machine learning with neural networks. These complementing strengths and weaknesses of neural and symbolic methods can be combined in different ways depending on the goal of the system (Hitzler et al., 2022).

A. d. Garcez and Lamb (2023) outline two main challenges that current systems do not address; (a) *variable grounding and symbol manipulation*, and (b) *commonsense and combinatorial reasoning*. The former is of particular relevance for this thesis. A. d. Garcez and Lamb (2023) characterise the first problem as *[..] the study of how symbols may emerge and become useful [..]*. At a certain point, *it may be more productive from a computational perspective to*

refer to such symbols and manipulate them [sic] symbolically. They illustrate this process with the example of translating images of digits to symbols that can be used precisely to compute their sum. Grounding the images in symbols allows the system to perform exact reasoning, in contrast to the approximate “reasoning” by neural networks. In general, (neuro-)symbolic methods are not built to introduce new symbols but are often restricted to a predefined domain of symbols. The work on compositional generalisation presented in the later chapters of this thesis is a first step in the direction of achieving general variable grounding. In particular, the integration of learning and reasoning through the combination of neural and symbolic components gives architectures strong compositional characteristics (A. Garcez et al., 2019; A. d. Garcez et al., 2022).

Neuro-symbolic AI is also partially motivated by cognitive and behavioural sciences, with for instance Nobel laureate Daniel Kahneman and Amos Tversky’s work on dual process theory later published as *Thinking Fast and Slow* by Daniel Kahneman (Kahneman, 2011). Briefly, the dual process theory suggests that the human cognitive process can be split into two systems; the fast System 1 and the slow System 2. System 1 reacts to direct stimuli, e.g., in the case of burning your hand on a stove, or riding a bike. System 2 is engaged when deliberate thinking is required to process complex information. In this line of work, neural networks are often considered the fast System 1, and symbolic systems are considered the slow System 2. We will elaborate on this mental model later in this chapter with a comment on *Thinking Fast and Slow* is used to motivate AI architectures.

### 3.4.1 Integrating Neural and Symbolic Components

Combining neural and symbolic components can take many forms. Table ?? shows the taxonomy of neuro-symbolic systems given by Kautz (2022). This taxonomy focuses on how neural and symbolic components are connected. Bader and Hitzler (2005) provide a taxonomy with greater focus on the traits of the components. For example, is it *neuronal* (mimicking biological neural networks) or *connectionist* (technological approximation, modern deep learning). Sarker et al. (2021) compare the taxonomies of Bader and Hitzler (2005) and Kautz (2022), analysing papers published in top tier AI conferences since 2011. One important observation is that for three of the eight dimensions in Bader and Hitzler (2005), all papers included in the study fall into one of the two categories. For instance, none of the papers covers methods that are neuronal, only connectionist. Therefore, we choose the more recent and integration-focused taxonomy of Kautz (2022) to further describe the field.

**symbolic Neuro symbolic** The standard in machine learning. Symbolic data (e.g. text), is fed into a neural model that produces symbols (e.g. a next word prediction).

Integration type	Description
symbolic Neuro symbolic	BERT (Kenton & Toutanova, 2019)
symbolic[neuro]	AlphaGo (Silver et al., 2017)
Neuro   symbolic	NS-VQA (Yi et al., 2018)
Neuro $\cup$ compile(symbolic)	
Neuro -> symbolic	
Neuro[symbolic]	

Table 3.2: caption

**symbolic[neuro]** A symbolic problem solver calls a neural system to solve a (sub)task

**Neuro | Symbolic** Neural component(s) converts continuous data, such as images, into symbolic representations that a symbolic system processes for task solving.

**Neuro -> symbolic**

**Neuro[symbolic]**

### 3.4.2 Neuro-Symbolic Visual Reasoning

While multimodal transformer methods have been extensively used for *Visual Question Answering* (VQA), including ViLT (W. Kim et al., 2021) LXMERT (Tan & Bansal, 2019), and VisualBERT (L. H. Li et al., 2020), neuro-symbolic approaches achieve state of the art performance on VQA datasets (Amizadeh et al., 2020; Mao et al., 2019a; Yi et al., 2018). These methods disentangle language, vision, and reasoning into three distinct components, allowing symbols to be composed in novel ways to solve out of distribution tasks. Neuro-Symbolic Visual Question Answering (NS-VQA) (Yi et al., 2018) is a three-component system for visual reasoning. The Neuro-Symbolic Concept Learner(NSCL) (Mao et al., 2019b) is a successor to NS-VQA. In the Neuro-Symbolic Concept Learner by (Mao et al., 2019b), programs for visual question answering are learned by combining modules for neural perception, semantic parsing of programs from language, and program execution. Other work builds on similar ideas (B. Zhang et al., 2021).

Finally, grounding in language and vision with neural networks is extensively researched for VQA (Antol et al., 2015; Chaplot et al., 2018; Fukui et al., 2016b). Neuro-symbolic approaches to VQA include the work by (Yi et al., 2018).

### 3.4.3 Neuro-Symbolic Logic Programming

DeepProbLog (Manhaeve, Dumančić, et al., 2018), DeepStochLog (Winters et al., 2021), Probabilistic Logic Programming (PLP (Dantsin, 1992; Ng & Subrahmanian, 1992)), provides reasoning under uncertainty, with methods such as (De Raedt et al., 2007). (L. D. Raedt & Kersting, 2008) gives an overview of the combination of PLP and ILP, Probabilistic ILP. Neuro-symbolic PLP, such as DeepProbLog, can realise predicates as trainable neural networks, as a way to ground and reason about visual concepts (Manhaeve, Dumancic, et al., 2018; Weber et al., 2019; Winters et al., 2021).

### 3.4.4 Kuhnian perspective

A body of research can be positioned in a broader context in many ways. The philosopher Thomas Kuhn introduced the term *paradigm shift* in his book *The Structure of Scientific Revolutions* (Kuhn & Hawkins, 1963). Kuhn studied the history of science and the progress of scientific knowledge, and Kuhn and Hawkins (1963) describes a paradigm shift as a scientific change through 5 phases; 1) pre-paradigm, 2) normal science, 3) crisis, 4) scientific revolution, and 5) post-revolution. A crisis arises when the reigning theories of a paradigm cannot explain oddities that seem to require considerable efforts outside of the current scope of theories to address. This leads to a paradigm shift where the underlying assumptions of the old paradigm are questioned to produce a new paradigm. Once a new paradigm is established, the field can return to the practice of normal science. Kuhn argues that science alternates between normal science and revolutions through these phases.

- Phase 1 – It exists only once and is the pre-paradigm phase, in which there is no consensus on any particular theory. This phase is characterized by several incompatible and incomplete theories. Consequently, most scientific inquiry takes the form of lengthy books, as there is no common body of facts that may be taken for granted. If the actors in the pre-paradigm community eventually gravitate to one of these conceptual frameworks and ultimately to a widespread consensus on the appropriate choice of methods, terminology and on the kinds of experiment that are likely to contribute to increased insights.[13]
- Phase 2 – Normal science begins, in which puzzles are solved within the context of the dominant paradigm. As long as there is consensus within the discipline, normal science continues. Over time, progress in normal science may reveal anomalies, facts that are difficult to explain within the context of the existing paradigm.[14] While usually these anomalies are resolved, in some cases they may accumulate to the point where normal science becomes difficult and where weaknesses in the old paradigm are revealed.[15]



- Phase 3 – If the paradigm proves chronically unable to account for anomalies, the community enters a crisis period. Crises are often resolved within the context of normal science. However, after significant efforts of normal science within a paradigm fail, science may enter the next phase.[16]
- Phase 4 – Paradigm shift, or scientific revolution, is the phase in which the underlying assumptions of the field are reexamined and a new paradigm is established.[17]
- Phase 5 – Post-Revolution, the new paradigm’s dominance is established and so scientists return to normal science, solving puzzles within the new paradigm.[18]

What does this have to do with learning language with machines? I argue that AI research is possibly in the middle of phase 3, after the revolution of deep learning in the early 2010s. While the impact and amount of impressive results of current deep learning models (current paradigm) is undeniable, there is an every-growing body of work showing how such models fail considerably and might not be fixable (TODO CITE). As a result, there is a strong wave of research on neuro-symbolic methods (new paradigm) designed to address the faults of deep learning. From a scientific philosophy perspective, it is a great paradigm shift as it clearly combines and encompasses previous results while contributing to something greater than the sum of the two. Kuhn notes that it is a good thing for science that a paradigm shift do not occur often or easily. Hence, time will still tell whether the neuro-symbolic paradigm is strong enough to reign.

### **Relationship between learning and reasoning**

Kakas and Michael argues for a synergistic relationship between learning and reasoning (Kakas & Michael, 2020). Learning provides the elements/knowledge used in reasoning, while reasoning provides inferences that can be used as inductive bias when learning or extend the base of knowledge. One of their questions is; how do we exploit the reasoning process to enhance the learning process? Similarly, Luc Steels argues that AI suffers from a paradox called the “...hermeneutic circle: *To understand the whole we need to understand the parts but to understand the parts we need to understand the whole (Gadamer, 1975)*” and uses this as an argument against the linear progression of information in common data-drive AI methods (STEELS et al., 2022). Luc Steels et al. define the process of understanding as “*We frame the process of understanding in terms of a process of generating questions, reducing questions, and finding answers to questions.*” and propose to facilitate this process via narrative networks (STEELS et al., 2022). This method of measuring understanding could be extended to other semantic representations, such as argumentation, to check its validity and find commonalities.

Kakas and Michael (Kakas & Michael, 2020) point out that generalisation cannot be absolute, referring to the problem of induction (L. Henderson,

2020), and pointing to Humes’ *A Treatise of Human Nature* (Hume, 1739) with “... inductive generalisation that is universal and absolute runs into logical difficulties as we cannot be sure that a future case will not contradict the generalization”. This is used to motivate that argumentation is a flexible framework wherein such logical difficulties are possible to resolve naturally, as an argument only holds until evidence proves otherwise. One example the authors give is the difference between *all beans from this bag are white* and *all normal beans from this bag are white*, given that all beans drawn from a bag during training are white. The second leaves room for the possibility that there are beans of other colors than white. Another example that is closer to the domain of this thesis is colored MNIST data. For instance, when each digit is assigned a specific color in training but can have any color during testing, the generalisations during training that color is the main feature, or the shape, are both valid all else equal. How can you model both feature dimensions independently to be able to instruct a model to focus on shape, not color, in a particular test, independent on the training procedure? This generalisation issue is important as context for the experiments performed in Chapter 5 and 7.

### 3.4.5 Thinking Fast and Slow, and neuro-symbolic AI

Daniel Kahneman and Amos Tversky categorises the human mind into system 1 and system 2, responsible for fast and slow thinking respectively (Kahneman, 2011). Thinking Fast and Slow, and other dual-process theories, have had a big influence on the discourse in AI and development of new systems and methods. For example, the field of neuro-symbolic AI is heavily influenced by the mental model of fast and slow components, such as artificial neural networks in combination with logic programming. L. d. Raedt et al. (2020) claims that “Kahneman, 2011 has put the quest for neural symbolic computation A. d. Garcez et al., 2015; A. S. d. Garcez et al., 2012; Hammer and Hitzler, 2007 high on the research agenda” However, in many cases, these architectures do not reflect the systems described in many dual-process theories and would not fall under those definitions. Simply building a dual-component system is not sufficient in order to realise the full potential of cognitive dual-process systems. We argue that for Human-Centric AI (HCAI), the misalignment of dual-process theory and its instantiation in AI systems together form a foundation that is more than shaky. For example, neuro-symbolic methods with neural networks used as input to symbolic components are still sensitive to the same type of biases that the AI community in general is tackling. Bias exists in humans’ fast thinking, but is countered by slow thinking and adjusted over time accordingly. The relationship between neural and symbolic components cannot be unidirectional. Similarly, not all cognitive theories on this topic suggests two components, but three or a plethora of components. Neuro-symbolic AI would benefit from research in this direction, acknowledging that there is existing work on this TODO CITE.

It is important to remember that dual-process theory contain other works

(e.g. J. Evans, 1996; Sloman, 1996; Stanovich and West, 2000, and that they also provide insights on their own. E.g. Sloman's tripartite Reflective, Algorithmic, and Autonomous minds Stanovich, 2009.

A definition of fast and slow thinking given by J. S. B. T. Evans and Stanovich (2013) gives the following definition of fast and slow thinking;

Our preferred theoretical approach is one in which rapid autonomous processes (Type 1) are assumed to yield default responses unless intervened on by distinctive higher order reasoning processes (Type 2). What defines the difference is that Type 2 processing supports hypothetical thinking and load heavily on working memory.

However, neuro-symbolic methods rarely adhere to this definition.

### 3.4.6 Explainability

One big argument for neuro-symbolic methods, especially in critical industry applications, is that symbolic components increase the transparency and explainability of such systems in comparison to neural networks. Symbols are easier to interpret, and interventions to change the behaviour of a neuro-symbolic system can be done on a symbol level. Explainability is needed both in evaluating systems, and as an affordance towards users. Neural networks are notoriously difficult to explain, whereas traditional methods like decision trees are much more straightforward. Combining neural networks with e.g. logic opens up for explainability by design, contrasted with the ad hoc explanations of, e.g., LIME (Ribeiro et al., 2016), SHAP (Lundberg & Lee, 2017), and DeepLIFT (Shrikumar et al., 2017). These methods are themselves vulnerable to adversarial attacks, as exemplified in (Slack et al., 2019). However, when humans explain decisions they (rarely) refer to brain activity and specific neurons but rather give justifications on higher levels. Most of the time, intuition rather than explicit reasoning is the underlying mechanism of our decisions and actions, as argued by, e.g., Mercier and Sperber (2009). This could support the post hoc explanations, but on the other hand, as McCarthy said, "*Artificial intelligence is not, by definition, simulation of human intelligence*" (Maker, 2006).

## 3.5 Challenges and characteristics

What is the purpose of the representation

- Transparency
- Allow explicit reasoning
- Shared representation in communication between humans and machines
- Allow for representing different world views/states.

What are the properties of the representation?

- Continuous concepts should be possible to represent
- Structured representation to allow for explicit reasoning
- Possible to construct new concepts compositionally from previously known
- Possible to represent 'possible world states'
- Possible to represent multiple modalities
- As little inductive bias as possible
- Possible to do multiagent grounding (mapping between 'world views')

Summarise challenges.

- Generalise outside given program structures
- Scalability
- How to introduce and ground new symbols, e.g. with clevr: Introduce arbitrary relations/concepts such as four red objects is called a gallub.
- Difficult to construct difficult datasets (comp.gen.)

## Chapter 4

# Probing multimodal language models

Example of how to use quotes at the beginning of chapters

---

*dali*

One major challenge with neural network-based methods is to understand their inner workings, which we commonly refer to as “looking into the black box”. Even if we have constructed a benchmark where good performance should tell us how a model behaves, it is difficult to know if the way a system produces an output follows a procedure that is actually desirable. In Chapter ??, we talked about this as an implication by the Chinese Room (Searle, 1980) thought experiment, where a computer system applies rules over Chinese characters to pass the Turing test. Searle (1980) questions whether this system *understands* Chinese, stipulating that it merely simulates it by hijacking meaning as imposed by the user. This perfectly illustrates a big issue with machine learning in general, and deep learning specifically, in that it is difficult to draw strong conclusions about the capabilities of a model strictly from observing its behaviour. With this in mind, we now turn to *probing* as a method to open up these black boxes.

Probing is one approach to reveal what information an embedding actually encodes (Conneau et al., 2018; Hupkes et al., 2020a; A. Rogers et al., 2018; Yaghoobzadeh et al., 2019). An embedding is a vector representation of anything from a word to an entire image that we obtain by feeding input to a neural network and extracting the activations from a layer in the network. For some methods, like word2vec, the embedding is the final output that is used on its own in a downstream task. Conversely, we can extract an embedding from any layer of a neural network to observe what information is encoded at certain stages throughout the network. In general, we say that

such embeddings capture some semantic information distilled from the syntactical information present in the input. Two well-known semantic embeddings are word2vec (Mikolov et al., 2013a) and BERT (Devlin, Chang, et al., 2019). In (Mikolov et al., 2013a) the authors demonstrate the additive compositionality characteristic of word embeddings in word2vec. In particular, if words or phrases occur frequently together in the same context, the sum of the vector representations of two single words is close to the vector representation of a phrase that combines those single words. For example, the sum of the vector representations  $\text{vec}(\text{Russia}) + \text{vec}(\text{river})$  is close to the vector representation  $\text{vec}(\text{Volga river})$  (see (Mikolov et al., 2013a)). BERT (Bidirectional Encoder Representations from Transformers) is introduced in (Devlin, Chang, et al., 2019) and is a bidirectional language model. Unlike word2vec, BERT considers the left and right context surrounding a word and pre-trains deep bidirectional language representations in an unsupervised way. BERT can then be fine-tuned to another task as an instance of transfer learning.

In this chapter we look at two sets of probing experiments. Section 4.2 shows probing for visual information in multimodal embeddings, giving us insight into how visual information is encoded in multimodal embeddings. For Section 4.3, we test how well language embeddings capture information about semantic relations such as synonyms and hyponyms.

## 4.1 What can probing tell us?

Conneau et al. (2018) define a (linguistic) probing task to be a classification task that categorizes sentences according to specific linguistic properties, such as sentence length. Commonly, probing tasks are agnostic to the specifics of encoder architectures and can therefore be used to compare across different methods. In (Conneau et al., 2018) linguistic probing tasks are systematized building on work in (X. Shi et al., 2016) and (Adi et al., 2016). In particular, Conneau et al. define a probing task to be a classification task that categorizes sentences according to specific linguistic properties. For example, given an encoder such as an LSTM, pre-trained on some downstream task, the sentence embeddings that it produces can be used to train a grammatical classifier for the probing task that asks to determine the number (singular or plural) of the subject of the main verb. If the sentence embedding captures that information in an accessible way, this may be revealed by the classifier performing well, whereas the classifier will certainly not be able to perform well if the information is not appropriately captured. Thus, the main idea of probing tasks is to draw conclusions from the classifier performance about the probed embedding; if the classifier succeeds it means that the semantic embedding captures interpretable information regarding the aspect under consideration. Hewitt and Liang (2019) argue that the performance of a probe alone is not sufficient, and introduce so-called *control tasks* to improve interpretability of probing tasks. A control task reveals whether high accuracy of a probing task

Type of information probed for:		
Surface	Syntactic	Semantic
Sentence length	Bigram shift	Tense
Word content	Tree depth	Number of subjects
	Top constituent	Number of objects
		Semantic incongruence
		Coordination inversion

Table 4.1: Probing tasks for semantic embeddings, organized along three broader probing categories as investigated in Conneau et al. (2018)

really indicates that semantic representations encode a linguistic property, or whether the probing task itself learns this property. In particular, a probing task is complemented with a control task that associates random outputs to the properties under consideration (for example, POS tags). Thus, a control task with low accuracy indicates that a corresponding probing task with high accuracy does indeed encode the probed property. Simply put, a probing task is a classification task in which a network is trained on a given embedding. The task is chosen so that the performance of the trained classifier provides insight into the nature of the information captured by the embedding. Finally, ignorance about what is actually captured in an automatically learned semantic representation may lead to serious consequences of various kinds such as propagating discrimination bias (Bolukbasi et al., 2016; Brunet et al., 2019; Caliskan et al., 2017), or causing safety hazards in robotics by inducing unexpected robotic actions that put humans at risk (Orseau & Armstrong, 2016; Wachter et al., 2017).

The probing tasks proposed by Conneau et al. (2018) probe sentence embeddings and are categorized according to the type of linguistic properties they capture: *surface*, *syntactic*, and *semantic information*. We now give a brief account of these categories, which are outlined in Table 4.1. Surface information comprises probing for sentence length (number of words) and the word content (whether the sentence contains a given word). The probing tasks in the syntactic category ask to detect bigram shift, tree depth and top constituent, revealing whether the embedding makes certain syntactic information accessible. Bigram shift tries to predict whether two adjacent words have been swapped (that is, encoding the syntactic order of words). Tree depth asks to determine the depth of the syntactic tree of the sentence, and the top constituent task asks the classifier to determine the sequence of the top constituents directly below the sentence (S) node. The probing tasks that probe for semantic properties are tense, subject and object number, Semantic Odd Man Out, and coordination inversion. The tense task consists in finding the tense of the main verb, whereas the subject and object number tasks ask to predict the grammatical number of subjects and objects of the main verb, respectively. The task Semantic Odd Man Out is about predicting whether a sentence has been

modified or not (i.e., a random noun or verb was replaced with another noun or verb). Coordination inversion probes for the information whether two coordinate clauses in a sentence have been switched. For example, “They might be only memories, but I can feel each one” and “I can still feel each one, but they might be only memories” (Conneau et al., 2018).

These probing tasks were defined for unimodal embeddings of natural language. Machine learning that utilize multimodal embeddings is a lively field (Adi et al., 2017; Felix et al., 2018; Socher et al., 2013), but little is known about what properties these multimodal embeddings actually capture. Work, such as by H. Wu et al. (2019), aiming to analyze embeddings according to the composition of their encoded concepts is rare.

The establishment of probing tasks is one way to gain systematic knowledge about what embeddings actually capture. Another complementary way is to build taxonomies of multimodal machine learning techniques and multimodal embeddings. Such taxonomies are proposed, for example, by Baltrusaitis et al. (2019) and Beinborn et al. (2018). Both groups of authors categorize embeddings according to different but partly overlapping criteria. The taxonomy by Baltrusaitis et al. (2019) classifies approaches according to five categories of criteria: (a) *representation* – how complementary and redundant information is represented, (b) *translation* – how data is mapped between modalities, (c) *alignment* – whether and how elements in the different modalities are aligned, (d) *fusion* – how information coming from different modalities is integrated, and (e) *co-learning* – in which ways the learning exploits multimodality.

The taxonomy by Beinborn et al. (2018) for (learning) multimodal representations distinguishes between (f) *concept representations* – embeddings that use low-level representations of concepts, (g) *projections* – embeddings that represent concepts using only one of the modalities, and (h) *compositional representations* – approaches that fuse or jointly embed the different modalities.

Multimodal probing tasks can support the location of a given method in a taxonomy without requiring intimate knowledge of its inner workings: probing which information is accessible by a network trained on the resulting embeddings provides insight into what information is present and how it is represented. Some major difficulties of multimodal processing tasks and representations are discussed (from the perspective of multimodal grounding) by Beinborn et al. (2018). Their discussion illustrates the usefulness of multimodal probing in general, and of visual-semantic probing in particular:<sup>1</sup>

*Combining complementary pieces of information* Different modalities contribute to the information content of multimodal input in complementary ways. For example, highly relevant visual properties, like the fact that birds have wings and violins are brown, are not usually mentioned in text as they are the default. Conversely, taxonomic and functional relations between concepts are poorly represented in images. Probing tasks that check whether, e.g., the word

---

<sup>1</sup>We extract two aspects from the four challenges discussed by Beinborn et al. (2018), basically combining challenges 2–4, as our focus is not on grounding.



*brown* relates to images of violins would allow to draw conclusions about how successfully these dimensions are combined in the embedding.

*Representation of abstract concepts* Multimodal grounding of verbs is difficult in comparison to grounding nouns and adjectives. This should not come as a surprise because verbs denote more abstract concepts than many nouns and adjectives do. Abstract concepts like *together*, *theory*, and states of mind give rise to similar difficulties. Probing tasks that evaluate how well such concepts are represented in multimodal embeddings would thus be highly useful.

*Combining complementary information:* Different modalities contribute different qualitative data. For example, highly relevant visual properties (e.g. birds have wings, violins are brown) are not represented in text, whereas taxonomic and functional relations between concepts are poorly represented in images. An open question is to which extent image information contributes when combined with text (most research investigating along these lines, focus on nouns and adjectives only). The authors argue for multimodal approaches that go beyond concept similarity.

*Multimodal grounding of verbs:* research investigating multimodal aspects for verbs are extremely rare. Existing work suggest that the performance for verbs is significantly worse (compared to nouns, adjective). The authors in (Beinborn et al., 2018) compare different representations of combining visual and textual verb pairs.

*Imageability of abstract words:* Concrete words can be visualized and visually represented much better than abstract words (e.g. *together*, *theory*). This also holds for verbs with a high degree of embodiment (e.g. *fall*, *dive*) compared to verbs with a lower degree of embodiment (e.g. *know*, *decide*).

*Selective multimodal grounding:* we lack an understanding of how to combine concept representations and this is a difficult tasks since, for instance, image collections are much more diverse for concrete concepts (e.g. *ladder*, *car*) than for abstract concepts (e.g. *happiness*, *intention*). Thus, approaches that perform selective multimodal grounding constitute a more plausible approach to sentence processing according to the authors. An open question is how to visually represent coordinating expressions (e.g. *but*, *nor*, *and*, *or*, *so*, *yet*). Casper et al. (2023) provides a comprehensive overview of probing methods.

TODO Fit somewhere else In *multimodal* semantic analysis, the syntactic domain is a Cartesian product of two or more domains, such as an image with a caption. The syntactic domain is sparse, where words are represented as one-hot vectors and images as their pixel values. The dense embedding, on the other hand, captures semantic information that a complex model can interpret. They allow for semantic analysis beyond what is possible with more syntactic or handcrafted representations. One example of a simple example is how it is possible to define distance between embeddings that carry semantic information. There are many models based on machine learning techniques that jointly process the input modalities (Shen et al., 2019; H. Wu et al., 2019). Multimodal learning models such as DeViSE (Frome et al., 2013) demonstrate

in particular that *zero-shot* learning can be significantly improved by engaging multiple modalities.

## 4.2 Probing Multimodal Embeddings for Linguistic Properties

Semantic analysis aims to infer meaning from data, relating objects in a syntactic domain to objects in a semantic domain. In natural language processing, semantic embeddings from methods such as word2vec, BERT, and GPT-3, revolutionised semantic analysis of text. The embeddings map words to real-valued vectors which reveal semantic aspects, for example, if words are related in meaning or belong to the same topic. Creating such an embedding means to enrich as well as filter out information. Unavoidably, some (usually surface and syntactic) information will be lost in the process of projecting words and their contexts onto a representation that focuses on meaning. For example, when the word ‘bat’ is seen in the context “the bat hits the baseball”, its embedding vector is quite different from what it would have been if the context was “a small bat captured by a zoologist”. Hence, there is a trade-off: the better we capture the semantics, the more surface and syntactic information becomes blurred. It depends on the downstream task what is the right balance between abstraction and detail. What complicates matters is that embeddings are automatically learned rather than crafted by hand, and thus it is not clear precisely which aspects such an embedding actually represents.

The combination of modalities that has hitherto received the greatest interest is the pairing of images and text, with their embeddings commonly called *visual-semantic embeddings*. Throughout the rest of this paper, we focus on visual-semantic embeddings, and base the empirical part of our work on the dataset *Common Objects in Context* (MS-COCO), which consists of images with captions (T. Lin et al., 2014).

When semantic analysis is applied to the text component of an image-caption pair, the visual information can resolve semantic uncertainties such as in the phrase “a man with a bat in his hands”. Figure 4.1 shows two MS-COCO images,<sup>2</sup> each image accompanied by two of its associated captions in the dataset. While humans would probably glean the correct interpretation of ‘bat’ and ‘club’ from the text alone (but not of ‘bird’), the visual-semantic information is much less ambiguous. The vector diagram to the right of the images illustrates how one might imagine the vectors of a word embedding such as word2vec to be affected by moving to the multimodal embedding, including visual information. Imagine the vectors  $\langle \text{bat} \rangle$ ,  $\langle \text{bird} \rangle$ , and  $\langle \text{club} \rangle$  to be those of the pure word embedding. That is, for simplicity we assume that the words are

---

<sup>2</sup>The authors gratefully acknowledge the MS-COCO dataset (T. Lin et al., 2014) as the source of the two photographs, licensed under <https://creativecommons.org/licenses/by-nc-nd/2.0/> and <https://creativecommons.org/licenses/by-nc/2.0/> CC BY-NC 2.0, respectively.



A tiny bat is held by someone with a camera.  
 A man in shorts is swinging a bat.  
 A man gently attempts to feed a baby bird.  
 A man is swinging a club with both hands.

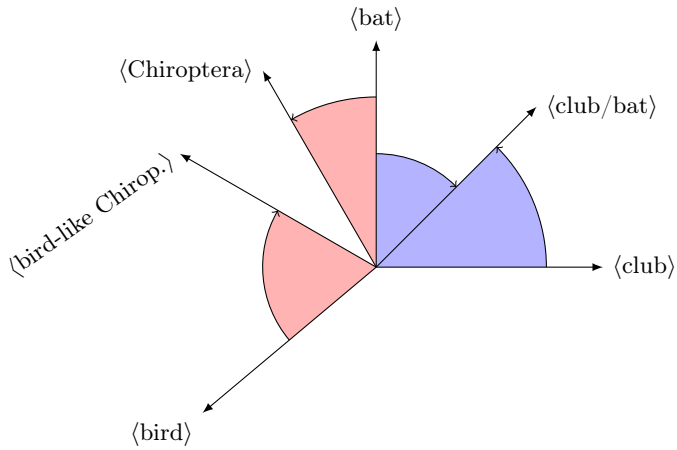


Figure 4.1: Image-caption pairs (top) and how vectors representing the words 'bat', 'club', and 'bird' may be affected by the image information (above)

embedded as in the original word2vec embedding, without taking the context provided by the sentence into account. In particular, the two occurrences of ‘bat’ in the captions are represented by one and the same vector  $\langle \text{bat} \rangle$ , and similarly for the two occurrences of ‘club’. Incorporating the information in the corresponding image may affect the vectors. The principle is shown in red for the combination of ‘bat’ and ‘bird’ with the left image, and in blue for the combination of ‘bat’ and ‘club’ with the right image. On the left,  $\langle \text{bat} \rangle$  becomes  $\langle \text{Chiroptera} \rangle$  (i.e., the vector now represents the mammal of the order Chiroptera) and  $\langle \text{bird} \rangle$  becomes  $\langle \text{bird-like Chiroptera} \rangle$ , intuitively representing a hybrid between birds and Chiropteras. In the right, both  $\langle \text{bat} \rangle$  and  $\langle \text{club} \rangle$  are turned into a vector  $\langle \text{club/bat} \rangle$  representing bats in the sense of clubs. While the information becomes semantically more accurate, other aspects are lost, e.g. whether the word ‘bat’ or ‘club’ was used, and probably also the fact that the second caption on the left actually mentioned a bird.

We propose probing tasks for visual-semantic embeddings (in other words, images with captions). In particular, we are interested in tasks that shed light on whether and how a given embedding makes use of the image information in relation to linguistic phenomena such as synonyms and polysemy.

Section ?? motivates our approach and relates it to existing work. Section 4.2.1 provides a systematic discussion and formalisation of probing tasks for visual-semantic embeddings. With this, we hope to map out which properties probing tasks of various types can be used to investigate. Section 4.2.2 introduces three concrete probing tasks that illustrate our approach, and which are used in our actual experiments reported on in Section 4.2.3. The code is publicly available.<sup>3</sup> The conclusion in Section 4.2.5 summarizes our findings and lists future challenges for multimodal probing.

## 4.2.1 Systematic Probing for Properties of Visual-Semantic Embeddings

In this section we develop a general view of visual-semantic probing tasks, and lift the ideas of Conneau et al. (2018) to the multimodal realm. Consider a property  $\Pi$  that a given embedding  $E$  may or may not have. In the visual-semantic case, such a property may be “the embedding associates visual properties with the nouns in the text component” or “the embedding encodes the number of objects in the image”. A *probing task* is defined to be a machine learning task – usually a classification task – that is designed in such a way that a model can be trained on  $E$ , and the achieved performance allows to draw conclusions regarding the extent to which  $E$  possesses property  $\Pi$ .

We are specifically interested in developing probing tasks for visual-semantic embeddings  $E$ , where  $\Pi$  is a property that reflects aspects of the multimodal nature of  $E$ . Ultimately, the goal is to come up with tasks that probe how the embedding maps the individual modalities into a common space. While we are

---

<sup>3</sup><https://github.com/dali-does/vse-probing>

not quite there yet, below we provide a general discussion of what to look for, and how such tasks may be categorized. Probing tasks that meet the following requirements seem to be especially valuable:

1. The task is a well-defined classification problem on combined (i.e., joint or coordinated) embeddings of two or more modalities.
2. The task gives insight into whether and how the multimodal embedding integrates the modalities.
3. The task has a simple and well-defined structure, so that the results are straightforward to interpret.
4. The task can be evaluated on standard data sets, or on datasets that can be created from such.

We propose that the probing tasks are organized according to how they make use of the information in the sample data to map out embedding characteristics. For the visual-semantic case, at an abstract level, each probing task either probes the embedding of the original text-image pair  $(T, I)$ , or it is based on turning  $(T, I)$  into  $(T', I')$  in a well-specified manner, such that by comparing the performance of a classifier on  $emb(T', I')$  and  $emb(T, I)$ , one can draw conclusions about the embedding. Depending on how  $T'$  and  $I'$  are obtained, different types of probing tasks arise.

### Direct Probing

Probing tasks based on  $emb(T, I)$ , that is, without inflicting changes on either part, are easy to implement, but have limited potential to reveal information about the specifically multimodal characteristics of the embedding. Nevertheless, some of the probing tasks by Conneau et al. (2018) have meaningful counterparts in this context. Here, we mention only the number of concepts, which is similar to sentence length and translates into *complexity*: given  $emb(T, I)$ , the task is to determine  $|T|$ ,  $|I|$ , and  $|(T, I)|$ , where  $|T|$  is the number of concepts mentioned in  $T$  (objects and properties of objects, say),  $|I|$  is the number of concepts in  $I$  (i.e., the number of segments and their properties), and  $|(T, I)|$  is the number of concepts in  $(T, I)$ . In the latter, an image segment and its counterpart in  $T$  would be counted only once. Note that an embedding may be expected to be ideal for determining  $|T|$  and  $|I|$  if it keeps the two modalities entirely separate, while good performance on the task of determining  $|(T, I)|$  indicates a tighter integration.

### Creation of Inconsistencies

By considering  $emb(T - x + y, I)$  or  $emb(T, I - x' + y')$  where  $y$  and  $y'$  do not align with  $x$  and  $x'$ , respectively, the effect of inconsistencies can be studied. For example, nouns in  $T$  aligned to objects in  $I$  may be replaced with other

nouns, and similarly for adjectives referring to attributes of objects in  $I$  such as position, color, size, form, and number. Variants may rely only on injecting inconsistent information, that is,  $emb(T + x, I + y)$ , where  $x$  and  $y$  form an inconsistent pair such as  $x = \text{ball}$  and  $y = \text{cube}$ . However, depending on the nature of the embedding this may require to make sure that  $T + x$  is actually a reasonably well-formed sentence.

## The Challenge of Interpreting Probing Results

We end this section with an urge for caution in the interpretation of probing task results, especially in the multimodal setting, and even more so when the results are “negative”.

Consider the task of determining the length of the caption of a text-image pair. If classifiers trained on this task perform well, this indicates that the embedding is not well integrated. The reason is that a well-integrated embedding would blur the distinction between the image and the caption, presumably associating a high sentence length even if a complex image is provided with a short caption. Unfortunately, the converse is not true: if classifiers perform badly, the reason may equally well be that the textual part of the embedding simply does not capture sentence length, or that the chosen classifier was unsuitable for the task. It may thus be easier to interpret a probing task that asks for the number of *objects* present in the text-image pair (see Section 4.2.2). Even in this case, poor performance does not necessarily say much about the nature of the embedding, because also a highly integrated embedding can be unsuitable for the counting task. However, despite these difficulties, this type of probing task may yield important insights if one is aware of the interpretation pitfalls.

### 4.2.2 Concrete Probing Tasks

This section illustrates the abstract principles introduced in Section 4.2.1 through a set of concrete probing tasks. These tasks will be experimentally tested in Section 4.2.3 and will, in future work, be extended with tasks of the types proposed in Section 4.2.1 to highlight complementary aspects of the semantic embeddings.

#### Direct Probing

Our first proposed probing tasks are instances of direct probing, as discussed in Section 4.2.1: *ObjectCategories* and *NumObjects*. In *ObjectCategories*, the task is to determine which of the 80 MS-COCO object categories are present in a given image. To turn the task into a simple classification task, we restrict the dataset to image-caption pairs in which only one of the 80 object categories is present (possibly multiple times). The second direct probing task, *NumObjects*, asks to estimate the number of object instances in the image. For this task, we bin the object instances present in an image into 6 bins (5 equidistant bins for the interval 0–29, and one bin for  $\geq 30$  objects).

## Semantic Congruence

Detection of semantic incongruity is an example of a probing task that arises from the creation of inconsistencies (see Section 4.2.1). It reveals whether the information propagated by  $emb(T, I)$  is sufficient to recognize that a caption has been modified, and to what extent this information stems from the visual part  $I$ . The associated probing task *SemanticCongruence* is the classification task that asks whether a caption has been modified. Later, we will perform this task on both  $emb(T, \emptyset)$ , and  $emb(T, I)$ . Without the image information, the decision must be based on purely linguistic features such as syntactic form, relative word frequencies, semantic consistency, and so forth. When the image is present, the model can also exploit incongruities between the modalities to detect modifications.

The characteristics of this probing tasks are largely determined by how the captions are modified, something that can be accomplished in numerous ways. FOIL-COCO by Shekhar et al. (2017) consists of modified MS-COCO pairs obtained by choosing, from each caption, a name of an object category and replacing it by another noun taken from the same MS-COCO super category. The replaced nouns occur in more than one caption, but their substitutes are salient in that they are not among the objects annotated in the image. To create plausible captions, the authors over-generate captions and use an LSTM trained on the original dataset to keep only the highest ranking ones.

To explore a range of linguistic features broader than nouns, which are the focus of FOIL-COCO, we compile a corpus of modified captions in which the linguistic head of each caption has been replaced. The procedure for modifying a caption works as follows. First, we run the Stanford dependency parser (Qi et al., 2020) on the caption to pick out the head. The parser also provides us with a part-of-speech tag for the head, which we use as input to the classical disambiguation algorithm by Lesk (1986). The algorithm returns the most likely synonym set (synset) and the abstract category assigned to the word by Wordnet. The replacement word is picked from a synset that is in the same Wordnet category. For example, if the head is ‘walk’ in the abstract category *verb.motion* then we might choose ‘fly’ from the same category. For simplicity, we avoid proper nouns. When the head is a verb, we prefer replacement words sharing the same set of frames, i.e., that can fill the same functions. Finally, we inflect the replacement word to match the inflection form of the head, and also mimic capitalization. To obtain a challenging data set, we generate  $N = 10$  modified sentences for each caption and then use BERT (Devlin, Chang, et al., 2019) as a language model to select the best scoring alternative. This yields sentence pairs such as that of Figure 4.2.

### 4.2.3 Experiments

This section describes our experiments with direct probing (see Section 4.2.2) and semantic congruence probing (see Section 4.2.2).



- 1.1 A *child* holding a flowered umbrella and petting a yak.
- 1.2 A *checker* holding a flowered umbrella and petting a yak.
- 2.1 A young *man* holding an umbrella next to a herd of cattle.
- 2.2 A young *mime* holding an umbrella next to a herd of cattle.
- 3.1 A young *boy* holding an umbrella touching the horn of a cow.
- 3.2 A young *wad* holding an umbrella touching the horn of a cow.
- 4.1 A young *boy* with an umbrella who is touching the horn of a cow.
- 4.2 A young *bear* with an umbrella who is touching the horn of a cow.
- 5.1 A *boy* holding an umbrella while standing next to livestock.
- 5.2 A *fry* holding an umbrella while standing next to livestock.

Figure 4.2: In task *SemanticCongruence*, the objective is to recognise semantically implausible captions.



## Experimental Setup

**Dataset** We use the Microsoft Common Objects in Context (MS-COCO) dataset curated by T. Lin et al. (2014). It consists of approximately 123 000 images, each with at least five human-written captions. The object categories of the manually annotated image segments comprise 80 object categories, grouped into 11 supercategories. We use the splits provided by Karpathy and Fei-Fei (2015), consisting of 82 783 train, 5000 validation, and 5000 training images, respectively. For testing, 5000 image-caption pairs over 1000 images are used of the test data, limited by what precomputed values are used by the investigated models. This split is originally used in training all the multimodal embeddings. We use image features precomputed by VGG19 (S. Liu & Deng, 2015) and ResNet-152 (He et al., 2016), as detailed in Table 4.2.

**Models** The visual-semantic models analysed through our probing tasks are VSE++ (Faghri et al., 2018), VSE-C (H. Shi et al., 2018), and HAL (F. Liu et al., 2020). In addition, we use the well-known unimodal language models BERT (Devlin, Chang, et al., 2019) and GPT-2 (Radford et al., 2019a).

Following the taxonomy by Beinborn et al. (2018), VSE++, VSE-C, and HAL are cross-modal transfer models trained via joint learning on the MS-COCO dataset. The implementations of VSE-C and HAL are both based on the open source code for VSE++. We use pretrained versions of these models, as provided with the respective papers. VSE++ learns visual-semantic embeddings by incorporating hard negatives into the loss function and using a similarity function that scores higher for the correct image-caption pairs than for the semantically incorrect ones (that is, for the negative samples). VSE-C learns instead by manipulating the original captions in the MS-COCO dataset so that they constitute contrasting image-caption pairs. HAL uses the same architecture as VSE++, but tries to avoid the so-called hubness problem where the results are skewed by frequently occurring vectors, by making the loss function aware of such structural properties of the data.

As all  $X \in \{\text{VSE++}, \text{VSE-C}, \text{HAL}\}$  embed the two modalities individually (though trained on the actual multimodal data), each results in two separate models  $X_{\text{text}}$  and  $X_{\text{image}}$ . We use these models in our experiments, in addition to “true” multimodal models  $X_{\text{avg}}$  and  $X_{\text{conc}}$  obtained by averaging and concatenating (resp.), the corresponding vectors in  $X_{\text{text}}$  and  $X_{\text{image}}$ .

BERT (Bidirectional Encoder Representations from Transformers) is a bidirectional language model introduced by Devlin, Chang, et al. (2019). It considers the left and right context surrounding a word, and relies on unsupervised learning to pre-train deep bidirectional language representations. We use an existing BERT model trained on the BookCorpus with 800 million words and on the English Wikipedia pages with 2 500 million words (Devlin, Chang, et al., 2019). The last model, GPT-2 (Generative Pre-Training, second generation), is a transformer-based unidirectional language model trained on 40GB of lightly curated Internet text (Radford et al., 2019a). We use the Transformers library

Model	Precomputed features	Emb. Size	Parameters
VSE++	VGG19	1024	15.5(159.2)M
VSE-C	ResNet-152	1024	13.8(74.1)M
HAL	ResNet-152	1024	11.3(71.6)M
GPT-2		768	117M
BERT		768	110M

Table 4.2: Overview of the investigated embeddings. The total size of the model, including models used to extract precomputed image features, is given in parenthesis.

for these models (Wolf et al., 2019a).

**Probing** We perform the three classification tasks *ObjectCategories*, *NumObjects*, and *SemanticCongruence*. For *NumObjects*, the label distribution between the 6 bins (see Section 4.2.2) is 47 443, 17 580, 9 626, 4 549, 2 061, 1 524 during training and 3 025, 1 060, 470, 240, 130, 75 during testing. Our baseline is naively guessing the largest class. The *ObjectCategories* task is based on 9 629 and 1 145 samples in the training and test data, respectively. For the *SemanticCongruence* task, a modified caption is chosen with probability 0.5, and remains the same for all models tested for fair comparison. We note that more complex models could yield higher accuracies, but following the results of (Hewitt & Liang, 2019) on probe model selectivity, this improvement does not necessarily reflect the availability of the information probed for. Therefore, we use two classifiers for probing; one multilayer perceptron (MLP) with one hidden layer of 256 nodes and sigmoid activation, and one linear classifier with softmax activation. Both models use a dropout of 0.2, similar to (Conneau et al., 2018). The probing models are trained on the MS-COCO data for 30 epochs using the cross-entropy loss function. In all cases the models start to converge within the last 10 epochs. The results reported are for the test split.

**Embeddings** For each probing task, the input to the classifier is either the image embedding from one of our used models ( $VSE_{++}^{\text{image}}$ ,  $VSE\text{-}C_{\text{image}}$ ,  $HAL_{\text{image}}$ ) or the text embedding from one of our models ( $VSE_{++}^{\text{text}}$ ,  $VSE\text{-}C_{\text{text}}$ ,  $HAL_{\text{text}}$ , BERT, GPT-2). For the size of the embeddings we refer to Table 4.2. For *SemanticCongruence* the  $X_{\text{text}}$  input consists also of modified captions. In order to contrast the probing results obtained with those for embeddings containing the full visual-semantic information, we also consider  $X_{\text{avg}}$  and  $X_{\text{conc}}$ , for  $X \in \{VSE_{++}, VSE\text{-}C, HAL\}$ . All weights of each model are frozen, meaning that no weights are updated for the embedding models during the probing.

**Experiment Details** The implementation is written in Pytorch 1.4.0 and trained on a NVIDIA Tesla V100 32GB GPU using CUDA 10 with Tensorflow

2.1. The models are all trained for 30 epochs, where each epoch times in at 100 seconds on average, and the experiments are conducted using the Adam optimizer with learning rate  $1.0 \times 10^{-4}$  for *ObjectCategories* and *NumObjects*, and  $1.0 \times 10^{-3}$  for the *SemanticCon* probing task. During initial experiments SGD was also considered but Adam showed better performance.

The implementations of VSE++<sup>4</sup>, VSE-C<sup>5</sup>, and HAL<sup>6</sup> the open sourced Github repositories with the best corresponding pretrained models are used. For BERT and GPT-2, the Python library Transformers<sup>7</sup> is used to access pretrained models. In both cases the base model is used, since initial experiments showed no significant difference when using larger models and in interest of keeping the comparison fair given that the larger models are substantially larger than the visual semantic embedding models. All pretrained models are outlined in Table ?? . Random numbers generated with Numpy uses a fixed seed of 1974, to make the experiments reproducible.

#### 4.2.4 Results and Analysis

Table 4.3 shows the results for the *ObjectCategories*, *NumObjects* and *SemanticCongruence* tasks, using both a MLP and a linear probe. While the two types of probes perform differently, the relative behavior across embeddings is very similar. A notable deviation from this general rule is the performance of BERT and GPT-2 on the *ObjectCategories* task (see below).

**ObjectCategories** We note that the **text-only** embedding for all three visual-semantic models yields better performance on the *ObjectCategories* task than the corresponding text-only embedding, with the exception of the linear probe for **HAL**. Worth noting is that **out of the multimodal embeddings, HAL performs well on image-only but worst on text-only for both probes**. Further, there is a large gap between the performances of MLP and linear probes on the BERT and GPT-2 embeddings. This supports the conclusion of Hewitt and Liang (2019) that MLPs, rather than acting as probes, may simply learn the task itself if provided with sufficiently rich embeddings as input, and that, therefore, linear probes may be a more appropriate choice.

Note also that BERT performs **best for both probes in the text-only case, while GPT-2 scores the lowest**. All merged embeddings significantly outperform their corresponding unimodal embeddings, with **concatenated VSE++ scoring the highest for both probes**. Merging the embeddings shows an improved accuracy of **3.8–11.9%** across both probe types, which suggests that the visual-semantic models combines the multimodal data in a useful way to

---

<sup>4</sup><https://github.com/fartashf/vsepp>

<sup>5</sup><https://github.com/vacancy/VSE-C>

<sup>6</sup><https://github.com/hardyqr/HAL>

<sup>7</sup><https://github.com/huggingface/transformers/>

<sup>8</sup>Since BERT is used during the generation of congruencies, this result is somewhat self-referential.

Embedding	ObjectCat.		NumObjects		SemanticCon.	
	MLP	lin	MLP	lin	MLP	lin
<i>Baseline</i>	-		0.605		0.502	
<i>Image</i>						
VSE++ <sub>image</sub>	0.753	0.768	0.646	0.613	0.502	0.506
VSE-C <sub>image</sub>	0.754	0.675	0.654	0.629	0.503	0.504
HAL <sub>image</sub>	0.799	0.730	0.674	0.633	0.533	0.510
<i>Text</i>						
VSE++ <sub>text</sub>	0.862	0.863	0.627	0.610	0.739	0.710
VSE-C <sub>text</sub>	0.838	0.805	0.629	0.617	0.763	0.756
HAL <sub>text</sub>	0.826	0.648	0.625	0.611	0.730	0.737
BERT	0.878	0.365	0.622	0.599	0.816	0.768 <sup>s</sup>
GPT-2	0.811	0.137	0.617	0.585	<b>0.792</b>	0.718
<i>Merged</i>						
VSE++ <sub>avg</sub>	0.862	0.876	0.658	0.638	0.707	0.662
VSE++ <sub>conc</sub>	<b>0.911</b>	<b>0.901</b>	0.661	0.641	0.743	0.713
VSE-C <sub>avg</sub>	0.831	0.783	0.665	0.636	0.735	0.713
VSE-C <sub>conc</sub>	0.896	0.879	0.666	<b>0.652</b>	0.776	<b>0.758</b>
HAL <sub>avg</sub>	0.847	0.820	0.667	0.642	0.712	0.702
HAL <sub>conc</sub>	0.903	0.849	<b>0.683</b>	0.648	0.730	0.730
<i>Improvement by merging</i>						
VSE++	0.049	0.038	0.015	0.028	0.040	0.003
VSE-C	0.058	0.074	0.012	0.023	0.013	0.002
HAL	0.077	0.119	0.009	0.015	0.000	-0.007

Table 4.3: Probing accuracies using a MLP with embeddings as input. The bottom three show for each model the difference between the best unimodal and the best merged embedding. All results are averaged over 5 runs and have variance  $\leq 0.01$ .

capture which objects are present in a scene. Overall *VSE++* seems to best capture and combine information about the object categories, beating BERT and GPT-2 by a large margin for both probes.

**NumObjects** The results for the *NumObjects* task show that the text embeddings consistently encode the probed information in a less accessible manner than the corresponding image embeddings which are, in turn, outperformed by their merged counterparts. Using MLP probing, HAL reaches the highest accuracy on both image-only and its merged embeddings, whereas VSE-C appears to be on par with HAL on merged embeddings under a linear probe, the precise result depending on the merging strategy. It is worth noting that the improvements from merging the embeddings are small, but are larger when using a linear probe. Once again, this supports the conclusion of Hewitt and Liang (2019) as it indicates that the weaker probes exhibit a better sensitivity.

It is worth noting that the best result for the *NumObjects* task is only about 8% better than the baseline. This seems to indicate that the task could be improved. Most of the images contain fewer than 10 object instances, thus falling into classes 1 and 2.<sup>9</sup> Table 4.4 display the per-class accuracy, showing that the accuracy for most embeddings and models is above 90% for class 1, and between 30-50% for classes 2,3, and 6. Classes 4 and 5 (i.e., 18–23 and 24–29 object instances) yield accuracies of approximately 4–18% and 3–15%, respectively. Further, the per-class accuracies show that the linear probes show performance comparable to the MLP probe on the first three classes, but never learn the 24–29 object class, and very few of the 18–23 and  $\geq 30$  samples.

Image scenes containing 0–5 object instances can exhaustively be described with words, mentioning numbers and listing distinct objects explicitly (“a cup and a fork”), whereas scenes containing 18–29 objects are harder to explicitly describe. The high accuracy for scenes with more than 29 objects may be due to the fact that the large number of object instances is a “property of the image” and might therefore be described with words such as “crowd”. A more balanced distribution could amplify the differences. Table 4.4 show that for class 1 (i.e. 0-5 object instances) the performance of the text-only embeddings is slightly better than for the multimodal embedding, which in turn performs slightly better than the image-only embedding. The results for class 1 is in average 0.9 for all three embeddings. For class 2 and 3 (i.e. 6-11 and 12-17 object instances) the performance for all three embeddings drops significantly (to an average of 0.2) and text-only has the lowest performance whereas the performance of multimodal embeddings is higher than for image-only embeddings. For class 4 and 5 (i.e. 18-23 and 24-29 instances) the performance continues to drop for all three embeddings with text lowest performance and slightly better or worse performance for image-only and multi-modal (depending on the considered model and embedding). For class 6 (30+) the performance for all embeddings and models increase again (and is similar to that in class 2, 6-11 objects). 30+

---

<sup>9</sup>Remember that we have 6 output labels representing the number of object instances.

text lowest again where the multimodal embedding are better for VSE++ avg (not conc) and for HAL mm does not add anything.

Model	0–5		6–11		12–17		18–23		24–29		≥30	
	MLP	lin	MLP	lin	MLP	lin	MLP	lin	MLP	lin	MLP	lin
<i>Image</i>												
VSE++ <sub>image</sub>	0.92	0.95	0.32	0.23	0.25	0.09	0.19	0.00	0.12	0.00	0.40	0.18
VSE-C <sub>image</sub>	0.92	0.93	0.40	0.34	0.28	0.25	0.06	0.07	0.00	0.00	0.48	0.53
HAL <sub>image</sub>	0.91	0.97	0.41	0.21	0.33	0.02	0.15	0.00	0.04	0.00	0.53	0.00
<i>Text</i>												
VSE++ <sub>text</sub>	0.93	0.93	0.29	0.26	0.15	0.09	0.04	0.04	0.00	0.00	0.24	0.13
VSE-C <sub>text</sub>	0.92	0.96	0.39	0.22	0.29	0.05	0.11	0.00	0.00	0.00	0.56	0.19
HAL <sub>text</sub>	0.94	0.96	0.26	0.15	0.12	0.00	0.09	0.00	0.02	0.00	0.23	0.00
BERT	0.96	0.99	0.16	0.04	0.13	0.00	0.00	0.00	0.03	0.00	0.20	0.00
GPT-2	0.92	1.00	0.22	0.00	0.23	0.00	0.04	0.00	0.05	0.00	0.24	0.00
<i>Merged</i>												
VSE++ <sub>avg</sub>	0.91	0.94	0.38	0.31	0.30	0.14	0.12	0.04	0.06	0.00	0.43	0.24
VSE++ <sub>conc</sub>	0.93	0.94	0.38	0.32	0.31	0.17	0.18	0.05	0.15	0.00	0.32	0.24
VSE-C <sub>avg</sub>	0.93	0.96	0.34	0.28	0.25	0.10	0.08	0.02	0.00	0.00	0.53	0.25
VSE-C <sub>conc</sub>	0.93	0.95	0.29	0.34	0.17	0.14	0.04	0.07	0.01	0.00	0.25	0.28
HAL <sub>avg</sub>	0.93	0.97	0.40	0.27	0.25	0.16	0.13	0.02	0.08	0.00	0.51	0.08
HAL <sub>conc</sub>	0.92	0.95	0.43	0.33	0.30	0.22	0.17	0.00	0.11	0.00	0.52	0.00

Table 4.4: Accuracy per label of the tested models A more detailed account of the accuracy of the tested models for the task *NumObjects*. The class labels correspond to the number of objects annotated in the image..

**SemanticCongruence** The results obtained from the *SemanticCongruence* probing suggest that the additional information provided by the multimodal component does not make up for the relative loss of linguistic information. This becomes particularly clear when using linear probing. VSE-C<sub>text</sub> outperforms VSE++<sub>text</sub> and HAL<sub>text</sub>, but is in turn clearly outpaced by the unimodal embeddings BERT and GPT-2. If we add visual information (to VSE++, VSE-C, and HAL), the performance generally does not increase, and even decreases in one instance. Our interpretation is that the alternative captions can be recognized from linguistic patterns such as verb-preposition agreement and other contextual information solely from having a good language understanding. Going back to Figure 4.2, we recognize that a well-formed sentence can still be highly unlikely given an understanding of language, just as Chomsky’s famous example “Colorless green ideas sleep furiously” (Chomsky, 1975). Further, although the visual information could provide additional clues, it also adds noise and makes the relative proportion of linguistic data smaller. If this interpretation is correct, an improved linguistic quality of the alternative sentences should make the visual information more valuable for the task. Since this vi-

sual information cannot encode whether the caption was modified,  $\text{HAL}_{\text{image}}$  aligns with the results of Hewitt and Liang (2019), suggesting that this MLP probe learns something other than the probing task. Finally, we note the good performance of BERT despite the fact that BERT was the embedding used to select the most convincing alternative captions, which should make them particularly apt at confusing BERT.

TODO Fix figures and description This section gives a sample of images from MS-COCO, together with original captions ( $x.y.1$ ), where  $x \in \mathbb{N}$  identifies the image and  $y \in \{1, \dots, 5\}$  identifies the original caption, and two series of modified versions, one series ( $x.y.2$ ) of lower quality, and one series ( $x.y.3$ ) of higher quality, both modifying the caption  $x.y$ . We use the higher quality series in the semantic congruence task. The modified versions illustrate some of the challenges of automatically generating syntactically valid alternatives. The most common reason for poor captions is that lexical disambiguation has failed, or that we do not control for verb-preposition coherence.



- 2.1.1 That *looks* like a wall mural in the background of this
- 2.1.2 That *occupies* like a wall mural in the background of t
- 2.1.3 That *runs* like a wall mural in the background of this
- 2.2.1 A huge heard of sheep are all *scattered* together.
- 2.2.2 A huge heard of sheep are all *pumped* together.
- 2.2.3 A huge heard of sheep are all *resurfaced* together.

**Summary** We see that the multimodal embeddings in the *merged* section of Table 4.3 outperform their **image- and text-only** embeddings on the tasks *ObjectCategories* and *NumObjects*. This indicates that the text- and image-only embeddings complement each other in what information they encode, and that merging them can utilize this fact. The concatenated embeddings yield consistently better performance than the averaged ones, probably because the complementary information is fully retained. It is not clear how well the text- and image-only embeddings project to the same space, which together with the introduction of noise from the respective modality can cause averaging to drown out important information. Still, averaging gives better performance than unimodal approaches **except for VSE-C on ObjectCategories**. The first two tasks are highly visual, which makes it only reasonable that the image embeddings encode more information of concern in these problems. It is also **suggested** from the results that the state-of-the-art unimodal text embeddings have a better semantic language understanding. It seems that there is a trade-off between language modeling versus understanding visual concepts, and that the training of the multimodal models has favored the latter. This idea also

aligns with the fact that these models are built for image-to-text and text-to-image retrieval, a task for which the unimodal embeddings are insufficient. Interestingly, HAL seems to be more focused on visual information as seen in the results on *ObjectCategories* and *NumObjects*. This could help explain why HAL outperforms VSE++ and VSE-C on text-to-image and image-to-text retrieval. We also note, importantly, that the language models are larger by factors 1.45 up to 10, excluding/including the network used to precompute image features, respectively. This can help explain why the multimodal models are not as capable in distilling the probed information in the text-only *Object-Category* task as BERT. To conclude, the results show that the image and text embeddings complement each other in understanding visual concepts, but that this does not extend to the understanding of language itself, as shown in the results on *SemanticCongruence*. Therefore, we conjecture that there is significant room for improvement on the multimodal embeddings for understanding scene semantics.

## 4.2.5 Conclusions

Probing semantic embeddings with neural-network based classifiers is like looking into a black box with a lens that is itself a black box. Valuable information can still be derived, but experiments that take this approach must be made with care, and the results analysed with caution. One approach to mitigate such opacity is proposed by Hewitt and Liang (2019), namely that the probing task is complemented with a control task to alleviate a possible misinterpretation of what semantic representations actually encode.

In the multimodal setting, it is helpful to use probing tasks (as well as complementing control tasks) that are simple, well-defined, and easily implemented on standard data sets. The importance of a task being well-defined is illustrated, albeit in a negative way, by the *NumObjects* task: Since there are countless equally valid ways to semantically decompose an image, it can simultaneously be true that an image shows dozens of sheep and that it shows a single herd. The flaw is arguably not as much in the task itself, as in the combination of task and data set. We may, for example, expect that the *NumObjects* task comes to its right in situations where logical units of counting are understood in advance, e.g., in the case of camera footage tracking traffic congestion, where a natural unit would be the number of vehicles. An interesting finding from our initial experiments was the importance of linguistic compared to visual information for complexity estimation and semantic incongruity detection.



## 4.3 NOT REWRITTEN AT ALL - Bridging Perception, Memory, and Inference through Semantic Relations

In Chapter ??, we saw how Bender and Koller (2020) and Bender et al. (2021b) postulate that it is impossible to learn meaning from surface form alone, and express concerns about what is perceived as an over-reliance on large-scale pretrained neural networks. This line of thought supports the interest in hybrid systems that amalgamate elements from complementary learning paradigms (see, e.g., (Hohenecker & Lukasiewicz, 2020; Pearl, 2019; van Bekkum et al., 2021; P.-W. Wang et al., 2019)). In (Dahlgren et al., 2021), we argue that this calls for an explicit distinction to be made between the faculties of perception, memory, and inference. We therefore promote the development of systems that consist of subsystems with responsibilities corresponding to the three faculties. Such future systems would thus consist of a perception component realised by a neural network, a component that provides explicit memory in the form of a knowledge base, and a third one performing symbolic inference, that is, rule-based reasoning.

We suggest to study how the subsystems can be aligned so for a seamless information flow between them. We view it as particularly important that (i) the network and the knowledge base together yield a consistent treatment of semantic relations and (ii) training takes the knowledge base into account, so that the resulting embeddings are consistent with established facts. Our conceptual discussion is complemented by a preliminary empirical evaluation of six popular English language models, which we subject to linear probes to test their abilities to capture central semantic relations.

After a brief discussion of related work in Section 4.3.1, Section 4.3.2 discusses the role of semantic relations in the context of our envisioned triad system while Section 4.3.3 and 4.3.4 of this paper complement our conceptual discussion with a preliminary empirical evaluation of the chances to achieve (i) by probing six popular language models with respect to a semantic relation learning task.

### 4.3.1 Related work

There is a rapidly growing literature on relation extraction and hybrid systems. Petroni et al. (2019b) observe that language models such as BERT (Devlin, Chang, et al., 2019) and GPT-3 (T. B. Brown et al., 2020) are imprinted with large amounts of common sense and factual knowledge during training. If this information can be reliably extracted then, they argue, word embeddings could find a new use as knowledge bases. To test the practicality of this approach, they consider a knowledge extraction task where a language model is given a sentence containing a subject word  $x$  and a relation  $\mathcal{R}$ , but where the object word  $y$  has been removed, and the model should guess the miss-

ing  $y$  (i.e., rank the vocabulary words) based on the fact that  $x$  and  $y$  are in the relation  $\mathcal{R}$ . The sentences are generated based on manually constructed templates, one per relation. For example, to the relation *birth-place*, they use the template “⟨subject⟩ was born in ⟨blank⟩” and instantiate it to “Dante was born in ⟨blank⟩”. The most important baselines are two variations of the relation extraction model by Sorokin and Gurevych (2017). Key findings are that language models appear to be better at learning one-to-one relations, whereas the relation extraction models are better at picking out many-to-many relations. Petroni et al. also find that the choice of template has an impact on the performance of the language models, and point this out as an item for future work.

Bouraoui et al. (2020) pick up this thread and propose a method for extracting good template sentences from BERT, and using these to fine-tune BERT so as to improve its performance on relation extraction. For a target binary relation  $\mathcal{R}$  (represented as a set of ordered pairs) and a sample of pairs  $R \subseteq \mathcal{R}$ , they filter the training data for sentences expressing that  $x$  and  $y$ , with  $(x, y) \in R$ , have the relation  $\mathcal{R}$ , and which would still be natural if  $x$  and  $y$  were simultaneously replaced by some other  $(x', y') \in R$ . Finally, they fine-tune a language model to predict, from an instantiation of one of the remaining sentences with a pair  $(x'', y'')$ , whether  $(x'', y'') \in \mathcal{R}$ . The most relevant aspect of this work for the present effort is the evaluation of the Bigger Analogy Test Set (also known as BATS) which contains 40 relations with 50 instances per relation (Gladkova et al., 2016). Bouraoui et al. (2020) report a mixed performance on the type of semantic relations considered here, namely hypernyms and hyponyms.

Additional methods for choosing template sentences are proposed by Jiang et al. (2020) who, similar to Bouraoui et al. (2020), mine the training data for suitable sentences. A dependency analysis on candidate sentences makes it possible to extract a larger variety of phrases that express the desired relationship than Bouraoui et al. (2020) can. The authors also generate candidate sentences by paraphrasing. In short, they find that both mined and paraphrasing have their usages, and that combinations of template types, e.g., manually constructed and mined, often perform well.

Poerner et al. (2019) question the conclusion by Petroni et al. (2019b) that BERT contains factual knowledge derived from the training data. The authors believe that in many cases, BERT simply exploits superficial similarities and general patterns to guess what is most likely. For example, from the fact that a person has a typically French surname, BERT could guess that that person is actually French without having learned the nationality of the particular person. To expose this weakness, (Poerner et al., 2019) remove what they believe are easily guessed pairs of subjects and objects from the data set of (Petroni et al., 2019b). They also provide a modified version of BERT, E-BERT, in which the embeddings of entities mentioned in Wikipedia have been replaced by a symbolic entity embedding. They find that E-BERT outperforms both BERT and ERNIE on the trimmed data set, but also that a combination E-BERT and BERT (taking the average of or concatenating the embeddings) give higher

Synonymy		Hyponymy		Meronymy	
band	set circle ring	assumption	theory miracle audacity	house	library attic porch
office	agency bureau authority	copper	metal penny policeman	road	bend crossing turnout
origin	root source blood	correction	improvement therapy punishment	song	words language chorus

Table 4.5: Instances of the relations synonymy, hypernymy, and meronymy extracted from WordNet.

accuracy than either on its own.

Rosenbloom (2010) model different types of declarative and procedural memory with what is essentially weighted hypergraphs, in which nodes correspond to actions and conditions, and edges to activation functions. Procedural and declarative memory are distinguished based on the direction in which values are propagated through the hypergraph. The analogy to human cognition is that procedural memory contains information about how to do something, whereas declarative memory concerns facts and events.

### 4.3.2 The role of semantic relations

As the brief account given in the previous section shows, there is a solid body of work on the extraction of relations from language models (see Section 4.3.1), to derive facts such as that the birth place of Olga Tokarczuk is Sulechów, Poland, and that the capital of Bolivia is La Paz. Looking to knowledge bases, it is natural to view them as graphs, where nodes represent objects and properties, and edges represent semantic relations. Finally, for logical inference, basic

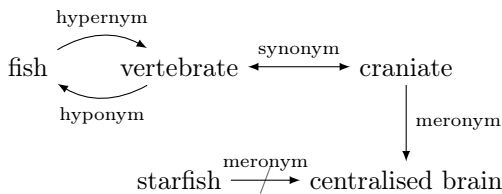


Figure 4.3: In this work we focus on recovering synonyms, hypernyms, hyponyms, and meronyms from natural language models via probing to understand the prerequisites of integration with knowledge bases.

semantic relations such as synonymy, hyponymy, hypernymy, and meronymy play a central role. We recall that words are synonyms if they have (nearly) the same meaning; that a hypernym of a concept is a generalisation of that concept (e.g., ‘bird’ is a hypernym of ‘sparrow’), while a hyponym is an instance of the concept (e.g., ‘spider’ is a hyponym of ‘arachnid’), and that a meronym of a concept is a part of the whole (e.g., ‘branch’ is a meronym of ‘tree’); see Table 4.5 for examples found in WordNet (Miller, 1992).

For logical inference, we can infer that starfish are not fish from knowing that ‘heart’ is a meronym of ‘craniate’ but not of ‘starfish’ (all craniates have hearts whereas starfish do not), ‘vertebrate’ is a hypernym of ‘fish’ (fish are vertebrates), and ‘craniate’ is a synonym of ‘vertebrate’. See Figure 4.3 and Table 4.5 for further examples.

To achieve a seamless integration of a neural network with a knowledge base of relations and an inference engine, we propose to devise methods for (i) enabling the network to utilise the knowledge base, but fall back on the less certain information in the embedding when necessary and (ii) taking the relations in the knowledge base into account during network training, so that the trained network reflects the contents of the knowledge base. In this endeavour, we believe that particular emphasis should be placed on the treatment of lexico-semantic relations such as meronymy, hyponymy, and synonymy because of their central role in logical deduction and lexical semantics.

### 4.3.3 Empirical study: method

To gain some initial insight into how well state-of-the-art pretrained contextual embeddings handle lexico-semantic relations, we conducted experiments on word embeddings generated by ALBERT (Z. Lan et al., 2020), ROBERTa (Y. Liu et al., 2019), BERT (Wolf et al., 2019b), and GPT-2 (Radford et al., 2019b). We also included Word2Vec (Mikolov et al., 2013b) and GloVe (Pennington et al., 2014) models in our experiments, for comparison. These are all self-supervised learning algorithms, based on neural networks and built to translate words into vector representations. BERT and GPT-2 are transformer models, each having 12 encoder layers. ROBERTa is a retraining of BERT on a larger data set, while ALBERT is an extension of BERT that has a higher data throughput with 10x fewer parameters, and thus scales better.

In contrast to the works discussed in Section 4.3.1, we do not extract relations from the embeddings by means of linguistic templates. Rather, we view sentence extraction as an instance of *probing* (Conneau et al., 2018; Hupkes et al., 2020a; A. Rogers et al., 2018; Yaghoobzadeh et al., 2019), a diagnostic method to reveal what aspects of the input the embedding actually encodes. Probing tasks should ideally be agnostic as to the underlying encoder architecture, so that results are transferable between embeddings (Dahlgren et al., 2021; Hewitt & Liang, 2019). Random control tasks (Hewitt & Liang, 2019) are implemented, see discussion in Section 4.3.4. In our experiments, we considered the following probing task: Given a pair of word vectors, we ask whether the

encoded words are in relation  $\mathcal{R}$ . This avoids the optimisation problem linked to the choice of template seen in (Petroni et al., 2019b).

All experiments are on the English language, and the data set used in our experiments was obtained from WordNet as follows. We first built a vocabulary  $V$  by taking the 5000 most common nouns in the Brown corpus (Kucera & Francis, 1967) and removing those not found in WordNet (Miller, 1992). This resulted in a vocabulary of 3497 words. For each word  $w$  in the vocabulary  $V$  and target relation  $\mathcal{R} \in \{\textit{hypernym}, \textit{meronym}, \textit{synonym}\}$  we then picked words  $v$  and  $v'$  in  $V$  such that  $(w, v) \in \mathcal{R}$  and  $(w, v') \notin \mathcal{R}$ , and stored these as triples  $(w, v, v')$ .

We formulate a classification task for each relation  $\mathcal{R}$ , and probe each of the investigated models for their ability to capture each relation in their respective embeddings. Each classification task is based on 1712, 306, 2740, 1630 samples for synonyms, meronyms, hypernyms, and hyponyms respectively. We use a linear classifier probe as these better reflect the availability of the information probed for, as shown in (Dahlgren et al., 2021; Hewitt & Liang, 2019). From  $(w, v, v')$ , positive  $(w, v)$  and negative  $(w, v')$  examples are drawn with equal probability, labeled either 0 or 1, to represent if the tuple represents a negative or a positive pair. The binary labels are given together with either  $(w, v)$  or  $(w, v')$  as input to the probe by concatenating both word embeddings. We train the probe for 10 epochs using 5-fold cross validation, using softmax activation, dropout of 0.2 to prevent memorising samples, and cross-entropy loss with the Adam optimizer using a  $lr = 0.001$ . We average the results over 5 runs. The experiment is implemented with Pytorch for CPU and uses the Huggingface (Wolf et al., 2019b) library for all pretrained transformers, and the Gensim (Rehurek & Sojka, 2011) library for word2vec and GloVe. The experiments completed within 1 hour on an Intel i7-based Linux laptop with 32GB RAM. The code is available on Github<sup>10</sup>.

#### 4.3.4 Results and discussion

Table 4.6 displays the numerical results, with the header row showing, for each relation  $\mathcal{R}$ , the size of the larger of the two classes. This number coincides with the control tasks implemented to measure selectivity, which are omitted to limit redundancy. The table shows linear probe classification accuracy for each language model, with the variance written out within parentheses. As can be expected, the variance is highest for meronyms where there is least data. Various observations can be made by comparing the results for the individual embeddings. Particularly worthwhile noting is the fact that GloVe and word2vec performs on par or better than the contextual embeddings, except for the case of hyponyms. This behaviour was seen with 5 and 20 training epochs as well.

The relatively strong performance of the pre-transformer solutions may not be surprising as far as synonyms are concerned, since their construction builds

---

<sup>10</sup><https://github.com/dali-does/semprof>

Model	Synonyms	Meronyms	Hypernyms	Hyponyms
<i>Majority</i>	50.1 (0.0)	54.2 (0.0)	51.0 (0.0)	50.7 (0.0)
Word2Vec	61.5 (1.8)	68.8 (5.0)	69.1 (1.5)	54.1 (1.7)
GloVe	63.2 (2.3)	73.3 (6.0)	68.7 (2.0)	55.7 (1.7)
ALBERT	51.9 (2.6)	48.7 (2.2)	51.2 (1.8)	51.7 (2.9)
ROBERTa	61.7 (1.9)	62.7 (5.9)	64.1 (1.2)	58.2 (2.8)
BERT	56.7 (1.2)	57.2 (3.6)	64.2 (1.6)	51.1 (0.3)
GPT-2	58.0 (1.2)	61.8 (5.3)	65.0 (1.3)	52.4 (2.5)

Table 4.6: The probing accuracy on the semantic relations, with variance given in parentheses. The accuracy of a “largest class” strategy is shown next to each relation. All transformers give embeddings of 768 dimensions, with word2vec and GloVe using 300 dimension. Each relation contain 1712, 306, 2740, and 1630 samples, respectively.

around aligning words found in the same context. However, we would not have expected similar results for hypernyms and even lesser so for meronyms. We note that ALBERT does not accessibly encode any of the relations, resulting in random guesses. This could be because ALBERT is trained using tenfold fewer parameters to produce much smaller embeddings, and might have less room for this type of information. Since ALBERT is comparable in performance to, e.g., BERT on many data sets and other metrics, this needs further investigation to see to what extent these relations are present in the data sets. The complexity of the probe could also be the culprit, as an embedding with lower dimensionality poses a more difficult task for a probe with limited capabilities of separating intertwined concepts. These results do not mirror those of Z. Lan et al. (2020), which indicates that the relations studied here could receive more attention in future evaluations of language embeddings. ROBERTa seems to generally outperform the other transformers, especially on the hyponyms, taking into account that not all results are statistically significant. Hypo-/hypernym relations usually follows a tree hierarchy, with hypernyms directed towards the root. This gives a decreasing number of hypernyms, for example, *fish* has six hypernyms but 39 hyponyms in WordNet, and it is likely that less common words will be chosen as a positive example for hyponyms. Weighting the words according to frequency could show different results, but filtering words based on the data the models are trained on is counterproductive to the purpose of these probes. ROBERTa is better able to capture synonyms, which could be an effect of the much larger dataset used in training compared to the other BERT-models leading to more of the less common examples of hyponyms being seen more. One hypothesis on why GPT-2 also shows poor performance is that Wikipedia is removed from the training data. The proposition is that many Wikipedia articles explicitly outlines hyponym relations, e.g. in “*The cat*

*is a [domestic species of small carnivorous] mammal*<sup>11</sup>.

Summarising the results, the fact remains that according to our probes no model covers the relations reliably. If this observation is confirmed by further experiments, it supports the case for a combination of neural networks, traditional relational knowledge bases, and inference engines. With this architecture, established facts could be retrieved from the knowledge base and complemented by less certain facts deduced by the network to cover up for missing information without causing inconsistencies. The results also indicate that a significant threshold should be applied for transferring relational knowledge derived from an embedding to a knowledge base, if this should be done at all, to avoid large error propagation. This is especially important if the “facts” in the knowledge base are considered to be absolute truths rather than tentative findings.

In conclusion, the reliability of the probe could improve with evaluation sets from relations found in knowledge bases, and a correlational study between probing accuracy and downstream NLP tasks could further support the usefulness of studying these relations.

## 4.4 Challenges and characteristics

---

<sup>11</sup><https://en.wikipedia.org/wiki/Cat>





# Chapter 5

## The compositional behaviour of multimodal language models

Example of how to use quotes at the beginning of chapters

---

*dali*

We established in Chapter 4 that multimodal language models represent visual concepts more distinctly compared to text-only models. In this chapter, we will study how multimodal language models behave on reasoning tasks over visual scenes. As it turns out, compositional generalisation is central to achieve good performance on the mathematical problems we devise. It also turns out that the available benchmarks are not complex enough, or does not cover multimodal data. Previous literature highlights how neuro-symbolic methods are much stronger on reasoning tasks. Hence, we are on a quest to construct multimodal benchmarks that compare neuro-symbolic and deep learning methods on mathematical visual reasoning and compositional generalisation.

### 5.1 DeepProbLog and compositionality

Neuro-symbolic methods rely on their symbolic components to achieve compositional generalisation. However, empirical results show that this might not always be the case, where the bias in data carries over too strongly into the model. In this section we will investigate how confounding information can fool a neuro-symbolic method. DeepProbLog (Manhaeve, Dumančić, et al., 2018) is a neuro-symbolic model for learning and reasoning with neural networks. We use confounding colors with ColorMNIST (B. Kim et al., 2019; Rieger et al.,

2020) to test how DeepProbLog generalises compositionally. This way, a model that picks up on the color as important will fail on the test set. The hypothesis is that the neural component in NeSy models will have better guidance via the symbolic information that is put into the model. DeepProbLog has not been tested on such data to the best of our knowledge, only on the regular MNIST dataset (Manhaeve, Dumančić, et al., 2018). Here we compare how DeepProbLog performs on ColorMNIST versus the regular MNIST dataset, and outline the differences.

We train and evaluate DeepProbLog on MNIST under three setups. The original experiment on addition with MNIST digits reported on in (Manhaeve, Dumančić, et al., 2018) is used as a baseline. The same experiment but with colored digits from ColorMNIST is used in two settings; consistent colors between training and testing (ColorMNIST<sub>same</sub>), and fixed colors during training and randomised colors during testing (ColorMNIST<sub>diff</sub>). The experiments on colored images uses the NeSyXIL model to handle color. Both experiments are run with the same hyperparameters (Adam, learning rate 1e-03), for the same number of epochs. The experiments build on code from (Stammer et al., 2021)<sup>1</sup> and (Manhaeve, Dumančić, et al., 2018)<sup>2</sup>. Table 5.1 shows the accuracy of learning to do addition using MNIST digits using DeepProbLog with and without confounding colors. Figure 5.1 shows the confusion matrix with

	MNIST	ColorMNIST <sub>same</sub>	ColorMNIST <sub>diff</sub>
DeepProbLog	0.7798	0.4498	0.1178
TODO Neural			

Table 5.1: Accuracy on evaluation data for both

grayscale MNIST digits. Figure 5.2 shows the confusion matrix when color is

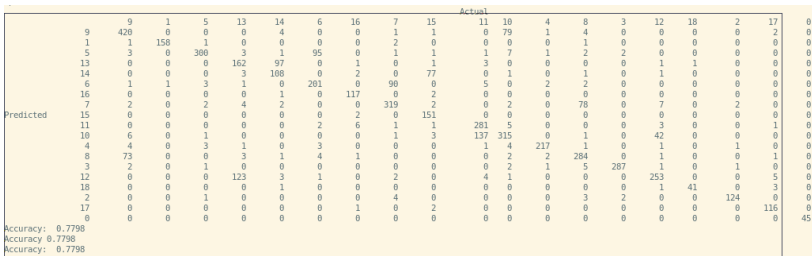


Figure 5.1: Confusion matrix for DeepProbLog on MNIST

consistent between training and testing, suggesting that the extra dimensions of color leads to a more difficult problem. Training the model differently is necessary to rule out factors related to model convergence. Figure 5.3 shows

<sup>1</sup><https://github.com/ml-research/NeSyXIL>

<sup>2</sup><https://github.com/ML-KULEuven/deepproblog>

		Actual																				
		1	no_answer	6	9	13	8	10	11	17	7	5	14	4	12	16	3	15	10	0	0	
Predicted	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	no_answer	113	0	216	309	102	263	332	206	0	159	280	220	0	253	76	0	119	0	0	0	
	6	0	0	138	0	0	0	0	2	0	0	0	2	0	2	0	0	0	0	0	3	0
	9	0	0	0	194	0	0	0	1	3	0	1	1	0	0	0	0	0	0	0	0	0
	13	0	0	0	160	0	1	0	2	0	0	1	3	0	1	0	0	1	0	0	3	0
	8	0	0	0	5	1	179	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	123	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	11	0	0	0	1	0	2	0	179	0	0	0	1	1	0	4	0	0	0	0	0	0
	17	0	0	0	0	1	0	0	0	180	0	0	1	0	0	0	0	0	0	0	2	0
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	1	1	1	0	0	0	0	118	0	0	0	0	0	0	0	0	1	0
	14	0	0	0	0	2	0	0	0	1	0	0	106	0	249	0	0	3	0	5	0	0
	4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	12	0	0	0	1	0	1	0	0	0	0	0	0	2	0	219	0	0	0	0	0	0
	16	0	0	0	1	1	0	0	0	3	0	0	0	0	0	111	0	0	0	2	0	0
	3	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	94	0	0	0	0	0
	15	0	0	0	1	1	0	0	0	6	0	0	0	0	0	1	3	0	0	175	0	0
	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	44
	0	0	0	0	1	3	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	40

Accuracy: 0.4498  
Accuracy: 0.4498  
Accuracy: 0.4498

Figure 5.2: Confusion matrix for DeepProbLog on ColorMNIST using the same color maps for training and testing.

the confusion matrix when color is confounding between training and testing, leading to almost random performance.

		Actual																				
		10	10	5	13	7	4	12	3	8	14	15	2	9	11	6	17	0	1	16	18	
Predicted	10	0	47	10	1	52	0	16	0	74	0	0	12	2	0	36	0	0	0	0	0	
	5	58	56	75	77	12	64	15	138	12	0	21	0	132	82	0	0	0	0	0	0	0
	13	3	0	0	0	0	0	0	0	0	0	0	0	0	38	1	0	0	0	0	0	0
	7	36	18	79	142	64	22	85	35	57	52	0	64	21	0	0	0	0	0	0	0	0
	4	41	0	0	0	17	40	50	0	27	68	92	1	45	46	54	0	0	0	0	77	20
	12	43	0	0	19	0	23	0	0	0	0	0	2	2	2	1	0	0	0	0	0	0
	3	80	16	29	42	0	19	0	22	1	0	0	97	86	0	0	0	0	0	0	0	0
	8	37	62	59	37	34	60	19	73	53	0	34	0	0	0	102	0	0	0	38	0	0
	14	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	5	0	0	0	35	8	0	2	19	41	0	0	1	0	38	0	0	0	78	28	0
	9	0	71	21	42	44	3	0	2	31	22	0	122	0	58	0	0	0	0	0	0	0
	11	1	0	0	1	0	0	0	2	1	0	0	40	54	46	0	0	0	0	0	0	0
	6	44	48	15	0	76	84	57	64	41	0	52	0	66	55	0	53	63	0	0	0	0
	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	5	5	0	0	0	0	0	0	0	0	11	7	0
	1	13	0	0	0	0	1	0	0	0	0	0	29	19	0	0	0	0	0	0	0	0
	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Accuracy: 0.1178  
Accuracy: 0.1178  
Accuracy: 0.1178

Figure 5.3: Confusion matrix for DeepProbLog on ColorMNIST using different color maps for training and testing.

These results suggests that DeepProbLog handle confounding colors in MNIST poorly. This contradicts the supposed strong compositionality of neuro-symbolic methods. Further experiments are necessary to identify why we see this behaviour, but this insight can be used to guide the design of future methods and datasets.

One philosophical issue is that for the model has no way of knowing whether we want it to learn the shapes or the colors of digits. During training, both color and shape directly map to the number, meaning there is no information available to the model to realise that the color is not important. In a sense, our biased expectations on the model are not aligned with what our data suggests. This shows that even for small toy datasets it is difficult to create learning tasks that are not ambiguous.

## 5.2 Multimodal Word Math Problems

We have now seen how the relationship between neuro-symbolic methods and compositional generalisation can be affected by the data it is trained on. Now we will introduce multimodal word math problems as a richer domain for us to

analyse such methods and compare them to deep learning alternatives. Now that we have outlined how we can use synthetic data from CLEVR to build multimodal benchmarks, we will use word math problems as the basis for tasks. Chapter 3 introduced us to the CLEVR dataset as a way to build multimodal benchmark. We will use this as the basis for our word math problem tasks.

Consider the following word math problem,

*Adam has three apples, and 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*. An arbitrary number of sentences with actions also require compositional generalisation (Chen et al., 2020; Keyzers et al., 2020; N. Kim & Linzen, 2020; B. M. Lake, 2019; Saqr & Narasimhan, 2020; Shaw et al., 2020), the capability to "*generate infinite use of finite means*" (Chomsky, 2014). Word math problems are a great setting for benchmarking systems on their generalisability in the intersection of natural language, reasoning, and vision. Previous work have mostly explored word math problems in a text only setting, like the problem shown above, using neural networks (Robaidek et al., 2018; Sundaram & Khemani, 2015; Sundaram et al., 2020), and other methods (Mitra & Baral, 2016; Sundaram & Abraham, 2018). Math Word Problem Solving (MAWPS) (Koncel-Kedziorski et al., 2016) was one of the earlier datasets introduced in the domain and collected around 3320 single/multi equation word problems involving operators  $+$ ,  $-$ ,  $*$ ,  $/$ . These word problems were annotated with equations involved and the answer (solution of the equation). More recently, larger datasets like Algebra Question Answering with Rationales (AQuA-RAT) (Ling et al., 2017) were introduced and it has around 100K multiple choice questions annotated with equations and a textual explanation for the rationale behind the equations. (A. Patel et al., 2021) illustrated the deficiencies in MAWPS dataset by introducing another dataset named Simple Variations on Arithmetic Math word Problems (SVAMP). SVAMP is created by making minor variations to problems in MAWPS (A. Patel et al., 2021) showed that state-of-the-art neural solvers trained on MAWPS performs poorly on the SVAMP dataset. See (Huang et al., 2016) for an overview of how to construct word math problems.

With word math problems as the basis, we introduce a multimodal word math problem dataset with images and corresponding mathematical tasks. We generate 3D scenes using CLEVR (Johnson et al., 2017), creating CLEVR-math. As discussed earlier, previous work such as CLEVR-Hans (Stammer et al., 2021) uses CLEVR in similar fashions to generate 3D data to examine specific behaviour. Figure 5.4 shows an example from this dataset. One important aspect of the human mind is that we envision changes without them physically manifest (TODO CITE). Imagining changes and inferring the consequences is a big part of how we reason. CLEVR-Math pose problems that tests a models ability to imagine such change, since the tasks involve chains of oper-

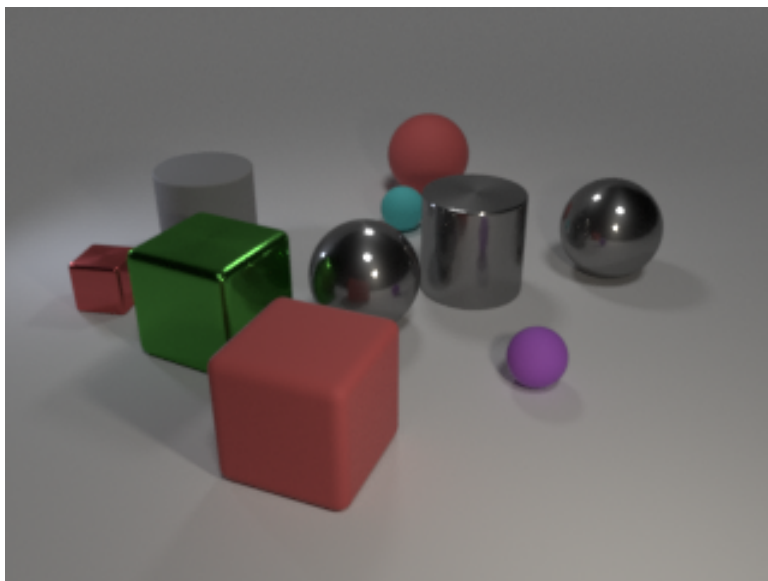


Figure 5.4: Selection of questions generated from this image: (i) *Remove all gray spheres. How many spheres are there? (3),*, (ii) *Take away 3 cubes. How many objects are there? (7),* (iii) *How many blocks must be removed to get 1 block? (2)*

ations and queries over the resulting (internal) state. Mathematical operators are well suited to build compositional generalisation tasks since the operators work recursively over any novel combination of mathematical elements

Therefore, testing the effect of compositionality on the mathematical reasoning provides a well-defined but broad domain where compositionality is a core principle. This could be tested by fixing the domain to a single color, shape, material, and size, and ask the same questions again. Additionally, including an increasing amount of confounding information is a related experiment. This could be done by always presenting blue cubes at training time, and randomising at testing. Important to measure is how the degree of confounding information affects the model. It should be enough to have only a couple of non-confounding examples (e.g. 5 out of 100) to realise that cubes are not always blue and should be disentangled as a concept. The hypothesis is that neural networks would need something close to equal distribution over the combinations and properties in order to achieve good performance.

When considering benchmarks for grounding, reasoning, or compositionality, it is important to not only measure behaviour but also internal structures of systems. Given enough capacity, a neural network would be able to memorise everything necessary to perform reasonably well. Similarly, a symbolic-driven system could introduce a new symbol for each combination of concepts or prop-

erties of objects. A *blue ball* can be encoded as a single concept, with *red ball* being encoded completely independently. Now, if we only test for behaviour, then this is difficult to uncover using the standard approaches in the deep learning community. One way of getting around this is by probing the internal structures to uncover desiderata. One such instance is by using confounding information, e.g. by always associating a shape with a specific color. There are multiple ways in which this can be investigated. For shapes and colors, it is possible to fix one shape-color pair while letting the other shapes and colors combine randomly. If a model is able to categorise shapes and colors to be part of the same concept categories, then it should be able to realise that spheres can be other colors than blue since the shape category does not have a fixed relationship to the color category. A slightly different experiment is to always associate shapes with specific colors, and measure the impact of introducing a few samples where this is not the case. For instance, if spheres are always blue, but one sample contains a red sphere, that might not be enough evidence that spheres are not synonymous with blue objects. However, once there have been 10 such samples, there is more reliable evidence that this is not the case. Given different systems, the ratio between fixed and randomised shapes and colors can reveal a lot about its inner workings. A desiderata is to have as small a supporting set as possible to break up such fixed relationships. The hypothesis is that neural networks perform poorly under such conditions.

### 5.3 CLEVR-Math

To recap, solving mathematical word problems requires one to be able to map the natural language text to a mathematical expression, identifying the known and unknown quantities and the operators to be used. Again, we consider the following math word problem,

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

**Equation:**  $X = 3 + 5 - 1$

Minor changes in the text may result in large semantic changes, e.g. changing just one word in the above problem - *eats* to *finds*, will change the equation to  $X = 3 + 5 + 1$ . Most of the recent efforts in automatic math word problem solving treat it as a translation task (from word problem to equation) and have employed sequence-to-sequence networks or sequence to tree (generating the expression tree of the equation) networks ((Luong et al., 2015), (Z. Xie & Sun, 2019), (J. Zhang et al., 2020)).

While text-based math word math problems are a great setting for natural language understanding, it would also be interesting to consider word problems which are accompanied by a diagram, and the information required to derive the solution has to be captured from both its textual and visual representations.

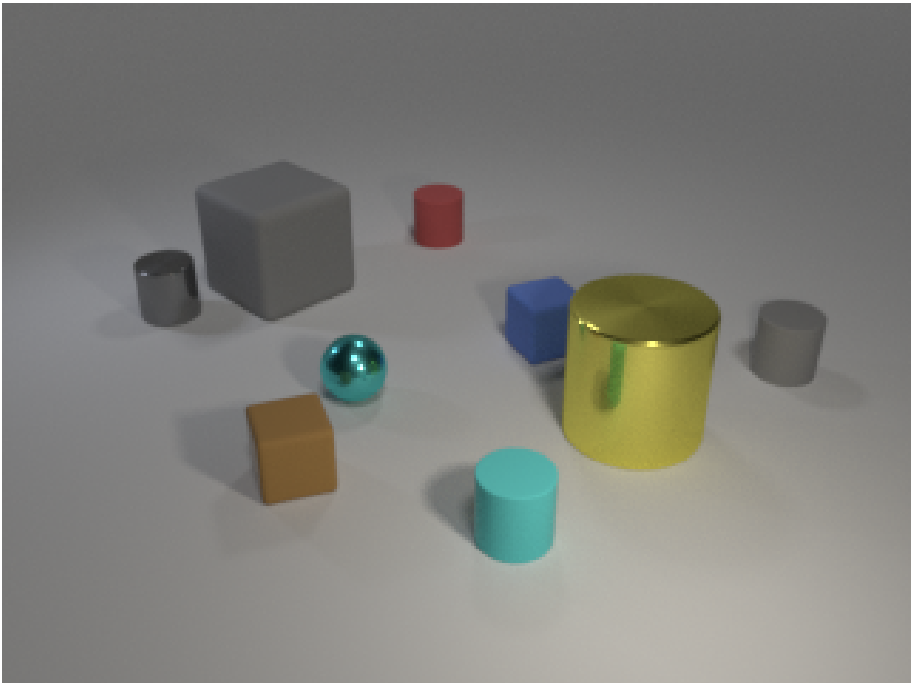


Figure 5.5: CLEVR-Math example question *Take away 2 matte cylinders. How many objects are left?* with corresponding mathematical equation  $X = 9 - 2$ .

That is, part of the problem scenario description is expressed as text and the other part is represented in the form of an image. We introduce such a multi-modal math word problem dataset, CLEVR-Math (since it is based on CLEVR dataset (Johnson et al., 2017)), where each problem has a textual and a visual description (image). Based on the strengths of using synthetic data previously discussed, CLEVR-Math allows us to test the ability of systems to generalise to unseen combinations of actions in the word problem. For example, we can train on single mathematical operations, and test on chains of operations. Figure 5.5 shows a sample problem in CLEVR-Math.

While each instance in CLEVR dataset has an image and a natural language query about the scene depicted in the image, in CLEVR-Math, the natural language query may not be about the scene represented in the image, but about the state of the scene after/before a sequence of actions are applied on the scene. The actions in our case are addition/removal of specific type of objects to/from the original scene. We believe this is an interesting problem setting as the ability to envision changes without them being physically manifested is an important aspect of the human mind.

Our contributions are two-fold, we

- construct an open source multi-modal math word problem dataset, CLEVR-

Math and

- analyse the performance of state-of-the-art neural and neuro-symbolic (NeSy) solutions for solving such multi-modal problems.

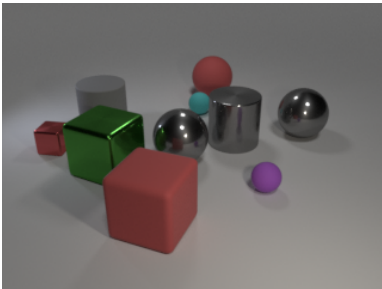
Our results and analysis shows how both neural and NeSy methods are unable to compositionally generalise to chains of operations.

### 5.3.1 Constructing CLEVR-Math

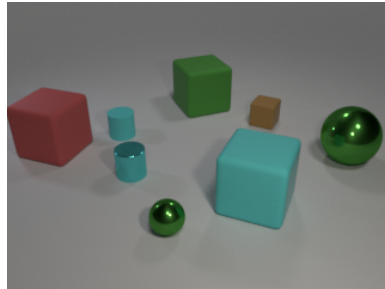
We construct the CLEVR-Math dataset as an extension of CLEVR by introducing three new functions and 13 templates. Using the codebase provided with CLEVR, we generate new questions based on the original scenes. We categorise the 13 templates into six types, all based on addition and subtraction. The domain is restricted to numbers between 0 – 10 to conform with CLEVR.

#### New CLEVR functions:

The three functions that we implement are - **subtraction** and **addition** to perform subtraction and addition, and **choose** to operate on subsets of objects. Instead of removing all blue spheres, **choose** allows us to remove a random number of a specific type of object, e.g. 2 blue spheres out of 4. The random number generated by **choose** replaces a questions “X” placeholder during generation. Figure 5.6a shows three examples of subtraction, and Figure 5.6b shows a question requiring multihop reasoning. Appendix ?? includes more samples from the test set.



(a) (i) *Remove all gray spheres. How many spheres are there? (3),*  
(ii) *Take away 3 cubes. How many objects are there? (7),* (iii) *How many blocks must be removed to get 1 block? (2)*



(b) *Take away all large green metallic spheres. Now remove all cyan objects. How many objects are left? (4)*

Figure 5.6: Example image-question pairs from CLEVR-Math, 5.6a showcase addition and subtraction, and 5.6b shows multihop reasoning. Answers in parenthesis.



## Question Categories:

The different question categories are shown in Table 5.2.

- **Remove group:** All objects belonging to a specific group are removed from the scene.
- **Insertion:** A specific number of objects are added to the scene.
- **Count backwards:** The query is about the change - that is the number of objects added/removed from the scene to get a goal state.
- **Remove subset:** A specific number of objects are removed from the scene.
- **Adversarial questions:** These are trick questions where the actions may be performed on one object, but the query is about an object that is not affected by the action. The adversarial actions are always on objects that are seen in the image.
- **Multi-hop:** In contrast to the above questions, multi-hop questions perform sequences of actions (insertion, removal) on the objects. Such questions with chained functions help us test a model’s ability to generalise to infinite combinations of operations.

Each problem in the dataset is also annotated with it’s equivalent functional program based on the CLEVR functions described in the previous section. For example, consider the question from insertion category and it’s program (the arguments of an instruction refer to another instruction - indicating it’s input is the output of the referred instruction):

**Q:** *Add 3 blue cylinders. How many cylinders are there?*  
**Program:** 1. scene, 2. choose[3], 3. count(1),  
4. filter\_cylinder(1), 5. count(4), addition(2, 5)

The program contains the `choose` function - `choose[i]` operator returns  $i$  ( $i = 3$  in this case).

## Question generation.

To support greater linguistic variation, we add synonyms for addition and subtraction to the template engine. *Subtract* can be replaced with *remove*, *take away* and *withdraw*, and *addition* with *introduce*, and *insert*. We use the same training and validation scenes as CLEVR, and generate 5000 new scenes as test data. Figure 5.7 show the distribution of attributes, words, templates and answers in CLEVR-Math, aggregated over the training, validation, and test data. The distribution is reflected in each of the splits.

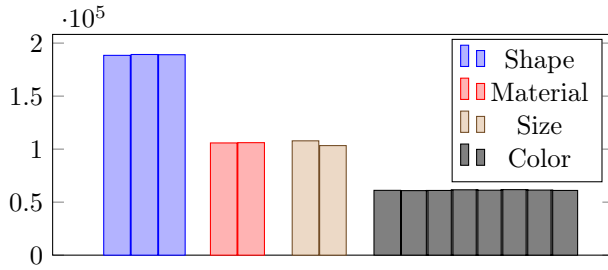
There are 50 words in the CLEVR-Math vocabulary, where the narrow language puts focus on the mathematical reasoning rather than advanced language

Type	Templates
<b>Remove group</b>	"Remove all <C> <S>s. How many <S>s are there?" "Take away all <Z> <C> <M> <S>s. How many <S>s are there?" "Take away X <C> <S>s. How many objects are there?" "Take away all <C> <S>s. How many objects are there?"
<b>Insertion</b>	"Add X <Z> <C> <M> <S>s. How many <Z> <C> <M> <S>s are here?" "Add X <Z> <C> <M> <S>s. How many objects are there?"
<b>Count backwards</b>	"How many <C> <S>s must be removed to get X <C> <S>s?" "Take away <C> <S>s. How many were removed if there are X <C> <S>s left?"
<b>Multi-hop</b>	"Take away all <Z> <C> <M> <S>s. Remove all <Z2> <C2> <M2> <S2>s. How many objects are left?"
<b>Remove subset</b>	"Remove X <S>s. How many <S>s are there?"
<b>Adversarial questions</b>	"Remove all <C1> <S1>s. Remove all <C2> <S2>s. How many <S1>s are left?" "Remove all <C1> <S1>s. How many <C2> <S2>s are left?"

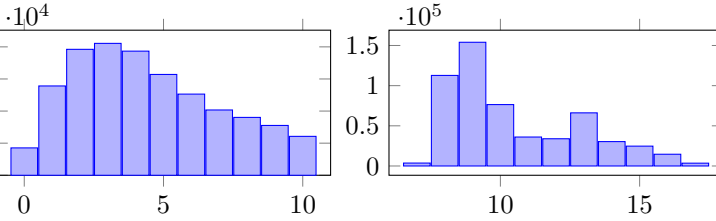
Table 5.2: An overview of the different templates implemented by CLEVR-Math. <Z>, <C>, <M>, <S> are instantiated to size, color, material, and shape during the question generation.

capabilities. Figure 5.7c show that most questions are 8-9 words long, with a second peak at 13 for the multihop questions.

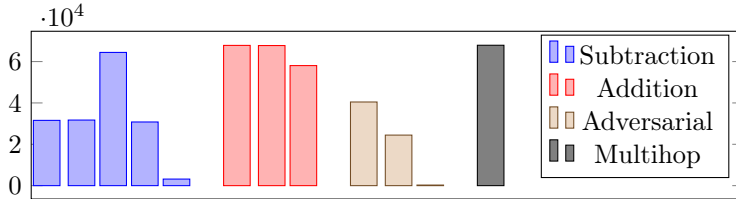
Table 5.3 shows the distribution of templates, with an approximately equal amount of questions for subtraction and addition, and similarly for adversarial and multihop questions. The ratios are consistent between splits. To test multihop reasoning and compositional generalisability we generate train-validation-test with only singlehop questions in training and validation, and only multihop questions in the test data. Thus, a model using the CLEVR-Math-multihop configuration must solve the multihop questions in a zero-shot fashion.



(a) Attribute distribution per category, showing even allocations.



(b) Answer distribution, from 0 to 10. (c) Distribution of number of words.



(d) Template distribution over categories of templates. Each bar corresponds to a template in each respective category. We see that subset subtraction (i.e., *remove 2 blue cubes*) is underrepresented.

Figure 5.7: The attributes are used evenly throughout the dataset, whereas the answers are biased towards the smaller numbers. The numbers are aggregated over all splits.

### Open sourcing data.

We open source CLEVR-Math as a Huggingface dataset <sup>3</sup> with two configurations; `CLEVR-Math` and `CLEVR-Math-multihop`. The extended CLEVR source code is available on Github <sup>4</sup>. Table 5.4 shows the Huggingface dataset card for CLEVR-Math. The template feature allows for filtering to perform, e.g., only singlehop training and multihop testing.

<sup>3</sup><https://huggingface.co/datasets/dali-does/clevr-math>

<sup>4</sup><https://github.com/dali-does/clevr-math>

Template	Train	Validation	Test
Subtraction	229364	49149	3281
Addition	193641	41600	2752
Adversarial	65180	13900	950
Multihop	67897	14553	972

Table 5.3: Distribution of templates in each data split.

Feature	Type	Example
template	String	subtraction-multihop
id	String	CLEVR_math_test_000010.png
question	String	<i>Remove 5 spheres. How many objects are there?</i>
image	image path	CLEVR_v1.0/images/train/CLEVR_new_000010.png
label	int64, 0-10	5

Table 5.4: Huggingface dataset card for CLEVR-Math.

### 5.3.2 Experiments

CLIP (Radford, Kim, Hallacy, Ramesh, Goh, Agarwal, Sastry, Askell, Mishkin, Clark, et al., 2021b) is used as a neural baseline. Questions and images are embedded using CLIP, and an additional classification layer is added to predict the correct answer. Fine tuning CLIP on CLEVR-Math as a masked language task before adding classification gave no significant improvements, while consuming significantly more computational resources. CLIP and this classification layer is trained jointly for 10 epochs with early stopping using a batch size of 64.

NS-VQA (Yi et al., 2018) is used as the neuro-symbolic baseline. Here, a mask-RCNN (He et al., 2017) is trained independently to convert an image to a scene graph. In our experiments, we skip this step and use the actual scene graphs associated with images. The question is parsed into a functional program by a sequence to sequence (Seq2Seq) network based on Bi-LSTM. A quasi-symbolic program executor executes the program generated on the scene graph of the image to return an answer. The Seq2Seq network is pre-trained in a fully supervised fashion by providing it a few examples (around 60 examples) of (*question, program*) pairs. The pre-trained network is then trained further using REINFORCE algorithm that returns a reward based on whether the program generated could derive the expected answer or not. Supervised pretraining and REINFORCE were run for 1000 and 5000 iterations, respectively, with a batch size of 128. Both CLIP and NS-VQA models were trained on a NVIDIA Tesla P100 GPU computing processor.

Each model is evaluated on each question category, and are trained on 2500, 5000, 10000, and 20000 samples to see the influence of the amount of data. For multihop, training and validation sets with and without multihop questions are

used, with the latter named *multihop (0-shot)*.

### 5.3.3 Results

Table 5.5 shows the accuracy of CLIP and NS-VQA on the different categories as well as an aggregated accuracy over the entire dataset. Both the models were trained on 10,000 samples. NS-VQA performs better than CLIP models for most templates apart from multihop. NS-VQA performs better on subtraction and adversarial problems (both based on ‘subtraction’ CLEVR function) than addition problems. This could be because the functional programs for addition problems always contain a **choose** operator. It is important to identify the argument to **choose** operator from the problem statement (which is mostly one of the numerical quantities in the word problem) to arrive at the correct answer. Unlike this, there are subtraction and adversarial problems (in remove group) that do not have a **choose** operator in the program. Neither of the

Model	All	Add	Subtract	Adversarial	2-hop	2-hop (0-shot)
NS-VQA	0.88	0.98	1.00	1.00	0.29	0.27
CLIP	0.35	0.57	0.30	0.29	0.27	0.24

Table 5.5: Accuracy on the CLEVR-Math dataset, shown for each template group and aggregated over all templates.

methods perform well on the multi-hop questions, with a clear degradation in the performance for NS-VQA. This is because the question parser of NS-VQA relies on a Seq2Seq network that does not generalize compositionally (B. Lake & Baroni, 2018a). CLEVR focus on visual attribute compositionality, and the multihop reasoning introduces higher demands on linguistic compositionality. When multihop questions are included in the training and validation data, naturally both methods improve their performance.

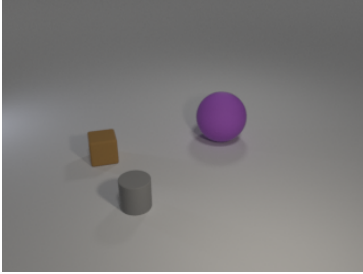
To gain further insight into CLIPs’ performance on CLEVR-Math, Appendix 5.3.5 shows a confusion matrix from training CLIP on 20 000 samples and evaluating on all question categories. These results show that most errors made by CLIP is off by ones. This reflects the generative nature of such models, in how they can get the context correct but sometimes miss out on details. We also see how CLIP focus on learning in the range 1-5, reflecting that these problems represent a majority of the problems.

Model	2500	5000	10000	20000
NS-VQA	0.6283	0.8840	0.6795	0.6118
CLIP	0.2918	0.3184	0.3528	0.3464

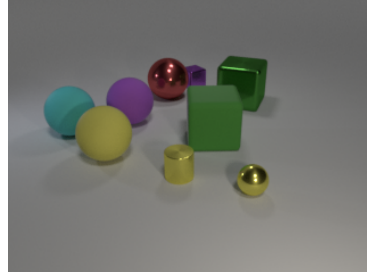
Table 5.6: Accuracy over all templates for different dataset sizes.

Table 5.6 shows how different training sizes influence the accuracy. We can see that NS-VQA achieves high accuracy from relatively few examples and

plateaus, which is consistent with the original results on CLEVR. It also seems like NS-VQA is overfitting with more data given, and one hypothesis is that more emphasis is put on the program, but that they are similar enough to confound NS-VQA. In CLEVR, the different questions were more distinguishable from a program perspective. CLIP scales with the number of samples, but plateaus at a much lower accuracy. We note that a larger number of samples could lead to similar performance for CLIP, but at the cost of more computational resources.



(a) *Subtract all small purple matte blocks. Subtract all blocks. How many objects are left?* was answered by CLIP with 3 instead of 2.



(b) *Subtract all red metallic objects. Subtract all yellow objects. How many objects are left?* was answered with 9 instead of 5 by NS-VQA.

Figure 5.8: Examples of when CLIP and NS-VQA fails on multihop questions.

We randomly sample 20 correct and 20 incorrect answers from the multihop test data for both CLIP and NS-VQA. Appendix ?? contains a subset of those samples, and Figure 5.8 illustrates two incorrect answers. There are no clear patterns of failures, such as only performing one of the actions, but we notice multiple instances where CLIP fails to perform overlapping subtraction, or subtraction when no objects match the description. Another observation is that half of the 20 incorrect answers from CLIP, where on images with only three objects. Scenes with few objects have a much smaller possible action space associated to it, meaning that there is less room for error. In Figure 5.8a, there are no purple matte blocks to remove, so the corresponding equation is  $3 - 0 - 1 = 2$ .

### 5.3.4 Conclusions

We introduced a new dataset, CLEVR-Math, containing word math problems about visual scenes. Our results show that the state-of-the-art NeSy model, NS-VQA, achieves higher accuracy on CLEVR-Math with less data and computational resources, than the neural model, CLIP. This is further evidence that neural methods, such as CLIP, are lacking in reasoning capabilities, even

after fine tuning. Given that NS-VQA uses perfect scene graphs, the comparison is not completely fair. We still expect the results of learning end-to-end to be consistent with the current results in alignment with the original results on CLEVR for NS-VQA.

CLEVR-Math successfully introduces a focused benchmark for learning and reasoning in multimodal data. There are a few natural extensions to this work, both on further development of the dataset and on evaluation. Extending the benchmark to answers outside of the range 0-10 would provide a more challenging domain, and providing scene graphs for each step of the reasoning chain could open up for other methods. The empirical results show that neither of the models could generalize to chained actions. Hence, it is also of research interest to design neuro-symbolic models where language perception is tackled in a more generalizable manner. Focus should lie on the representations (symbols) that are learned. Other interesting directions is to introduce a representation that is manipulated internally according to the actions as they are read. Adding longer chains of operations, or chains with alternating subtraction and addition, would put even more emphasise on the reasoning capabilities. Finally, there is an opportunity to add confounding information to test the robustness, e.g. by associating each shape with a fixed color during training and randomise it during testing.

### 5.3.5 CLIP confusion matrix

Figure 5.9 shows a confusion matrix indicating that CLIP is learning something for all labels. It also shows that when an answer is wrong, it is off by one. The confusion matrix also reflects the distribution over answers, showing that most answers are considered by CLIP to lie in the range 1-5.

## 5.4 Extending NS-VQA for Multihop Questions

So, I ran the "train on two steps, test on one step" experiment and learned a couple of things. Basically, NS-VQA has a really hard time learning with either 0 or abysmal accuracy. First of all, there are two main limitations of NS-VQA regarding what templates it can work with and how easy it is to extend the set of operators. We can write the two-hop template for questions on the form Remove all <C1> <S1>s. Remove all <C2> <S2>s. How many objects are left? the following three ways;

```
"text": [
  "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>" ]}],
```

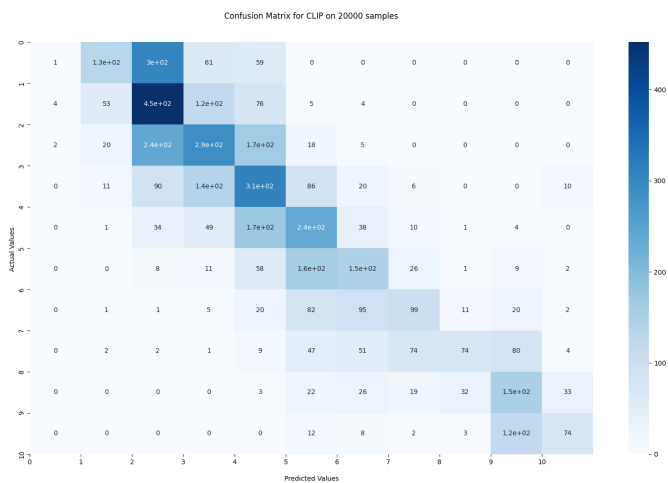


Figure 5.9: Confusion matrix for CLIP trained on 20 000 samples.

```
{ "type": "filter", "inputs": [ 1 ], "side_inputs": [
  "<Z2>", "<C2>", "<M2>", "<S2>" ] },
{ "type": "subtraction_set", "inputs": [0,1]},
{ "type": "subtraction_set", "inputs": [3,2]},
{ "type": "count", "inputs": [4]}
]
```

Now, the documentation says NS-VQA requires a topological ordering of the operations. The first realisation after digging through the code is that the implementation cannot handle this first template, even though it is ordered 'correctly', because of how it remembers past output (basically only one variable to keep track of output, meaning we cannot remember both the C1S1 and C2S2 sets when executing). So we can "linearised" the template as follows;

```
"text": [
  "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]}
]
```

We remark that each filter operation is unfolded into a sequence of `filter_size`, `filter_color`, `filter_material`, `filter_shape` operators. Initially, this template seems to be too hard to learn with the LSTM generating longer and longer sequences. However, on closer inspection of data preprocessing pipeline, we identify that the issue comes from the stringification of the programs. Consider the following template instantiation;

```
[
{'function': 'scene', 'inputs': [], '_output': [0, 1, 2, 3,
4, 5], 'value_inputs': []},
{'function': 'filter_shape', 'inputs': [0], '_output': [0, 2
, 4], 'value_inputs': ['cube']},
{'function': 'subtraction_set', 'inputs': [0, 1],
'_output': [1, 3, 5], 'value_inputs': []},
{'function': 'filter_color', 'inputs': [2], '_output': [3, 5
], 'value_inputs': ['brown']},
{'function': 'filter_shape', 'inputs': [3], '_output': [3],
'value_inputs': ['cylinder']},
{'function': 'subtraction_set', 'inputs': [2, 4],
'_output': [1, 5], 'value_inputs': []},
{'function': 'count', 'inputs': [5], '_output': 2,
'value_inputs': []}
]
```

Once this is fed through the preprocessing function `list_to_prefix`, we get the execution tree illustrated in Figure 5.10. Here we can see four `subtraction_set` operations, instead of the expected two required to compute the program. This in turn gets linearised as the string `"count subtraction_set subtraction_set scene filter_shape[cube] scene filter_shape[cylinder] filter_color[brown] subtraction_set scene filter_shape[cube] scene"`. This erroneous linearisation is a attempting to correctly chain a stateless execution of programs. We can get around this problem by introducing `subtraction_set` as a state-manipulating operation, relaxing the requirement on the parser to produce an execution tree for stateless program executions. Adapting the code to this relaxation, we instead get a tree that better reflects the intended program execution order. This execution tree is correctly linearised as `"count subtraction_set filter_shape[cylinder] filter_color[brown] subtraction_set filter_shape[cube] scene"`.

Now that the preprocessing no longer assumes statelessness, we modify the executor to allow `subtraction_set` to manipulate an internal representation of the scene. Using an internal representation that can be manipulated reduces the complexity of the execution trees, offloading that complexity onto the executor from the LSTM.

An alternative to what is described above, is to implement a remove opera-

tor that reduces the template complexity by combining the filter and subtraction operations.

```
"text": [
"Remove_all<Z><M><C><S>s.Remove_all<Z2><M2><C2><
  <S2>s.How_many_objects_are_left?"
],
"nodes": [
{ "type": "scene", "inputs": [] },
{ "type": "remove", "inputs": [ 0 ], "side_inputs": [
  "<Z>", "<C>", "<M>", "<S>" ]},
{ "type": "remove", "inputs": [ 1 ], "side_inputs": [
  "<Z2>", "<C2>", "<M2>", "<S2>" ]},
{ "type": "count", "inputs": [2]}
]
```

This showed to the second limitation, where the program executor (i.e., the code that runs the sequence of operators that the LSTM learns to generate) is implemented like this:

```
self.filter_color[blue] = self.filter_blue
...
self.filter_shape[sphere] = self.filter_sphere
...
```

```
def filter_blue(self, scene1, _):
    # Loops over objects and returns all that are blue
```

In other words, for every operator that have attributes as parameters, there is a separate function for each attribute. The same goes for the vocabulary, where entire can be filter\_color[blue]: 28 Sort of taking the "it is difficult to extend the domain for neuro-symbolic methods" to the extreme. Now, this is the reason why filter is unfolded into a sequence of filters. Hence, if you want to implement remove to take a set of attributes, you cannot unfold it similarly to filter, and hence you would need to add enumerate all combinations of remove and attributes for the vocabulary and in the executor. Which isn't reasonable or sustainable as an approach. Instead, I generalised the code to add the attributes to the program vocabulary, and modified the code to push attributes to a stack that is emptied whenever an operator is read. Doing this lead to above 50% accuracy on the two-hop validation data (didn't test it with the 1-hop test data yet). The next step was to make sure that the modifications worked with the second template above. This has not worked yet, and I'm not sure whether it is the LSTM not being able to learn properly, or something else. However, the modifications mean that instead of the template being tokenized as scene, filter\_color[blue], remove\_set, filter\_color[yellow], filter\_shape[cube], remove\_set, count of length 7, it gets instantiated as scene, filter\_color, blue, remove\_set, filter\_color, yellow, filter\_shape, cube, remove\_set, count of length 10. Since there is no indication

what attributes are and what operators are, this might become difficult for the LSTM to learn. In my experience, LSTMs can be fickle. In the original CLEVR dataset, all templates are on the form

```
scene
op [0] ...
op [1] ...
op [2] ...
```

producing chains that are seemingly easier to learn, even though the questions themselves might be compositionally difficult. (edited)

## 5.5 Experiments with modified NS-VQA on Compositional Generalisation splits

NS-VQA will perform good on the comp. gen. splits, while the transformer will not. As a result, we would like to see if CL can help with the comp. gen. capabilities of a transformer.

## 5.6 Challenges and characteristics

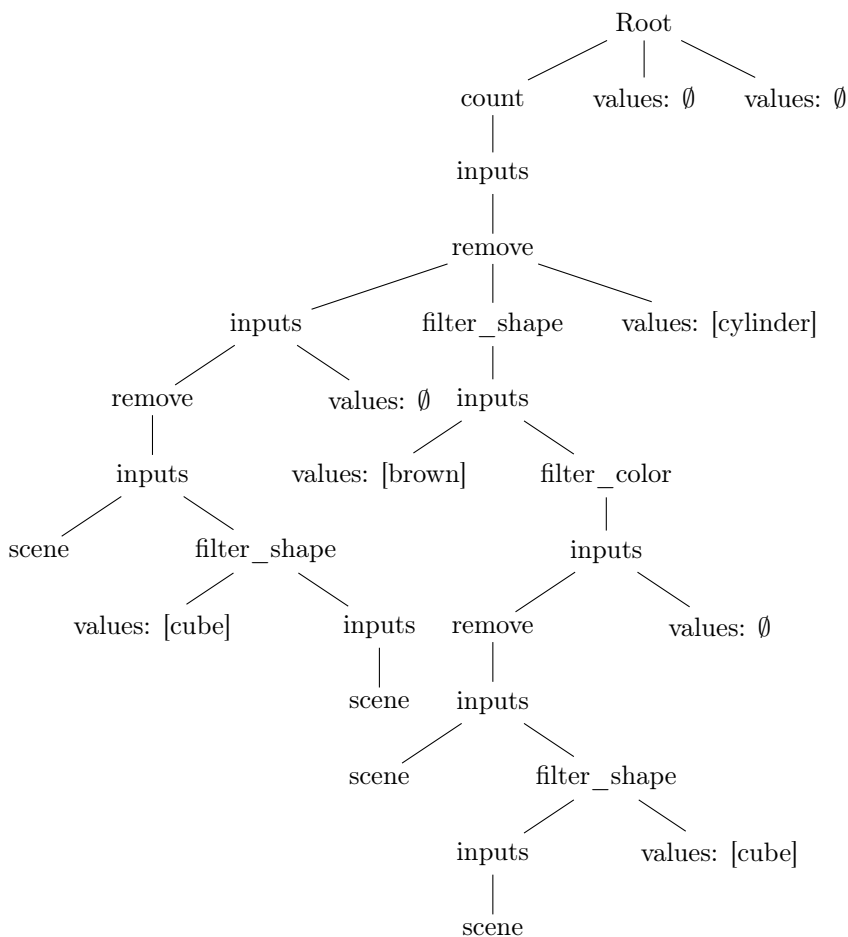


Figure 5.10: Illustration of the execution tree produced by the original NS-VQA parser for the question “Remove all cubes. Remove all brown cylinders. How many objects are left?”.

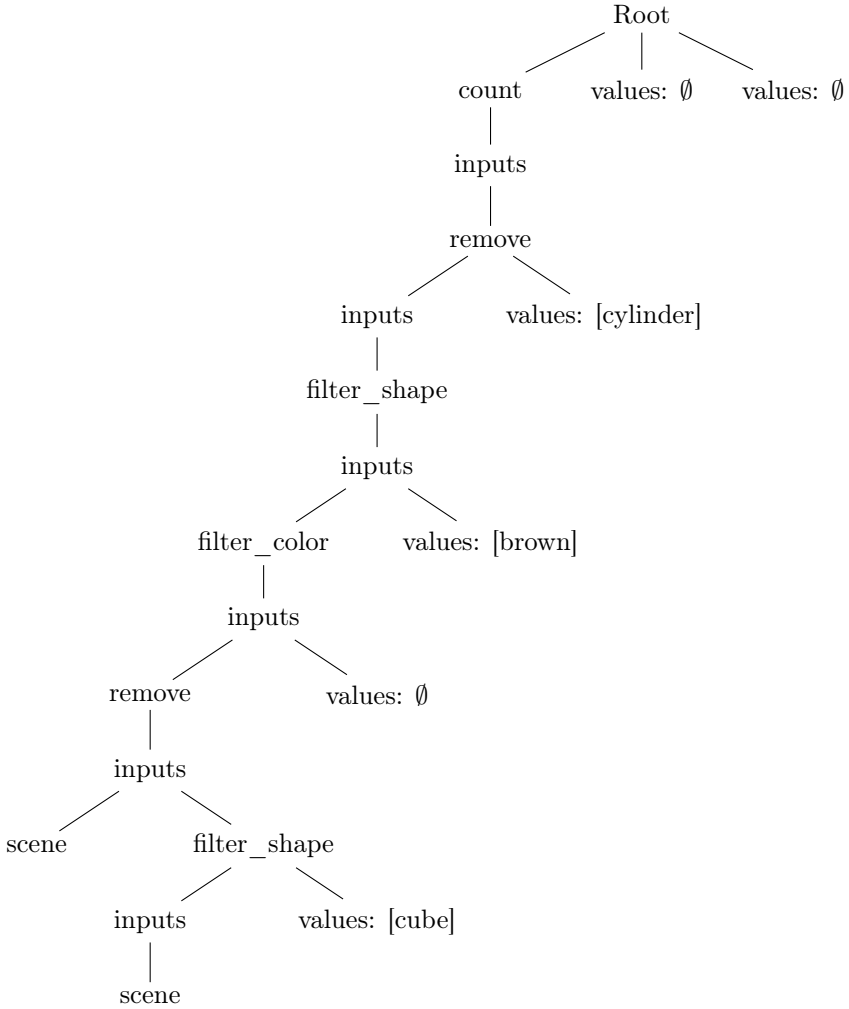


Figure 5.11: Illustration of the execution tree produced by the modified NS-VQA parser for the question “Remove all cubes. Remove all brown cylinders. How many objects are left?”.



## Chapter 6

# Multimodal Compositional Generalization

*“Hold the newsreader’s nose  
squarely, waiter, or friendly milk  
will countermand my trousers.”*

Perfectly ordinary words, but never before put in that precise order. A unique child delivered of a unique mother.

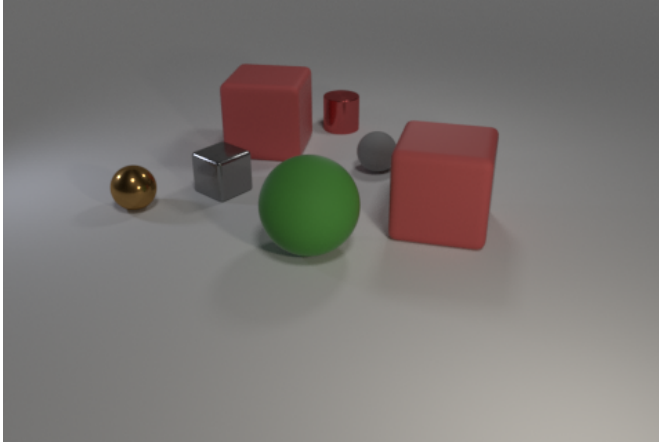
---

Stephen Fry, *A Bit of Fry and Laurie*, Series 1, Episode 3 (1989)

### 6.1 Compositional Generalization splits for CLEVR-Math

Compositional generalization is a key challenge for artificial intelligence, as human language and cognition are both largely compositional. It requires a model to understand the underlying characteristics of the data (such as structure, types, and grammar) to be able to recombine elements in novel ways rather than rely on its ability to memorize specific examples. Compositionality in human language is multi-faceted, and prior work (J. A. Fodor & Pylyshyn, 1988; Hupkes et al., 2020b; Szabó, 2022) has studied models for evidence of *systematicity* and *productivity*. Both systematicity and productivity rely on the recombination of known constituents into larger compounds. *Systematicity* means that the ability to produce/understand some sentences is intrinsically connected to the ability to produce/understand certain others J. A. Fodor and Pylyshyn (1988). For example, 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. *Productivity* describes our ability to produce/understand a potentially infinite number of sentences with a finite-capacity brain because we can build the meaning of a sentence from the meaning of its parts (Frege, 1963; Szabó, 2022). For example, one who understands the sentence *John gave a book to Mary*, also understands *John gave the book to Mary, who gave it to Lucy, who gave it to Liz*. Closely related is the principle of *Substitutivity* (Pagin, 2003) which states that if a substitution preserves the meaning of the parts of a complex expression, it also preserves the meaning of the whole.



**Q:** Remove all spheres. Remove all red cubes. How many objects are left?

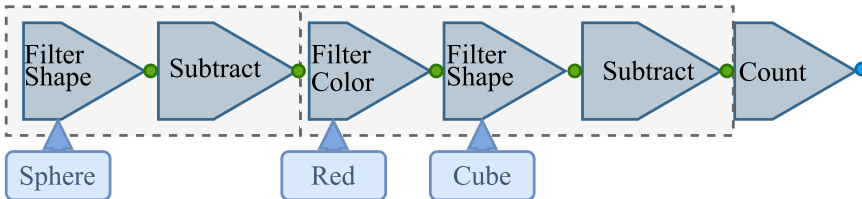


Figure 6.1: An example from CLEVR-Math (Lindström & Abraham, 2022b), with a corresponding functional program.

There has been significant recent interest in trying to understand the compositional generalization abilities of modern neural networks (Hupkes et al., 2020b; N. Kim & Linzen, 2020; B. Lake & Baroni, 2018a). However, most of these only focus on benchmarking neural networks for language-only semantic parsing tasks. In this paper, we broaden this scope by developing a comprehensive benchmark for multimodal compositional generalization in visual question answering (VQA). We then compare the state-of-the-art neural approaches with neuro-symbolic methods to understand which methods are bet-



ter suited to tackle the various dimensions of the compositional generalization challenge. Multimodal visual reasoning allows us to examine the interaction between language and vision, grounding challenges of language to the scene, and its relationship to compositional generalization. Figure 6.1 shows an example of such reasoning in the CLEVR-Math dataset, which is a domain consisting of questions about simple arithmetic operations on images, where one such operation corresponds to one reasoning step or “hop”. In Figure 6.1, function blocks in the dashed boxes correspond to these reasoning hops. We develop a multimodal compositional generalization benchmark and present several data splits that test generalization over various attributes, such as *red cubes* held out from training, and generalizing to longer chains of reasoning hops. We evaluate one state-of-the-art neuro-symbolic method NS-VQA (Yi et al., 2018) and two transformer-based neural methods, ViLT (W. Kim et al., 2021) and CLIP (Radford, Kim, Hallacy, Ramesh, Goh, Agarwal, Sastry, Askell, Mishkin, Clark, et al., 2021a); a powerful vision-and-language transformer, and a popular multimodal few-shot learning model, respectively.

Recent work often focus on evaluating the compositional generalization in neural networks, 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. Neuro-symbolic methods have been shown to be effective for a variety of synthetic tasks of a compositional nature, including visual question answering (VQA) (Mao et al., 2019a; Yi et al., 2018). NS-VQA (Yi et al., 2018) has been used in many subsequent works, most recently Hong et al. (2023), to showcase the strength of neuro-symbolic methods on visual question-answering tasks. In the recent CLEVR-Math dataset (Lindström & Abraham, 2022b), NS-VQA is shown to perform well on 1-hop problems such as *Remove all blue cubes. How many objects are left?*, but fails on the 2-hop example in Figure 6.1. Most other work demonstrate the success of NS-VQA in in-distribution settings, and in this paper we evaluate along different dimensions of compositionality: systematicity, productivity, and substitutivity. For systematicity we test whether the models can reason over novel attribute compositions and for productivity, we test whether the model is able to generalize to *longer* or *shorter* hop questions than seen during training. For substitutivity, we test how models perform on syntactic modifications of the questions, designed to be more challenging than in previous work (Johnson et al., 2017). We extend the NS-VQA method to handle the multi-hop questions for our productivity splits, and compare the performance of the neuro-symbolic method with the previously mentioned pre-trained neural models. We perform a number of ablation experiments to show the impact of increasing training data, as well as the training data complexity (measured by the number of *hops*) on the performance of neural versus neuro-symbolic models.

**Math Word Problems** The math word problems of CLEVR-Math makes it especially suitable for benchmarking compositional generalization in the intersection of natural language and reasoning (B. Lin et al., 2023). Previous work mostly explores word math problems in a text-only setting using neural networks (Robaidek et al., 2018; Sundaram & Khemani, 2015; Sundaram et al., 2020), and other methods (Mitra & Baral, 2016; Sundaram & Abraham, 2018). Y. Lan et al. (2022) gives an overview of the different aspects of compositional generalization that math word problems cover, and Kudo et al. (2023) show that neural networks struggle the most with systematicity in arithmetic math word problems. CLEVR-Math is especially suitable for compositional generalization benchmarking since, compared to previous CLEVR datasets, the math word problems introduce multi-hop reasoning requiring scene manipulation.

### 6.1.1 Contributions

We summarise our contributions as follows:

1. We introduce a compositional generalization benchmark comprising splits over functions, modalities, and attributes, for multimodal mathematical reasoning.
2. We perform an extensive empirical comparison of a neuro-symbolic (NS-VQA, along with a multi-hop extension we propose in this paper) with different neural baselines (ViLT, CLIP).
3. We show via ablation experiments the effect of the amount and complexity of training data on the compositional generalization ability of the neuro-symbolic method compared to the neural methods.

## 6.2 Methods

Since mathematical reasoning is so important and is suitable for compositional generalization experiments (B. Lin et al., 2023), we create our data splits with CLEVR-Math as the basis. Our work consists of an extension of the templates and constraints of CLEVR-Math, as well as generating new data in compositional generalization splits. Table 6.1 outlines all data splits used in our experiments, including *counting* corresponding to zero reasoning hops. In order to achieve this using CLEVR-Math, we introduce a new CLEVR function, **remove**, along with a set of new constraints to exclude certain attribute compositions from the data generation process. One general constraint on all data splits is that the questions will always mention shape, but the other attribute types (**color**, **material**, **size**) are not always included in questions. Figure 6.2 illustrates the CLEVR program NS-VQA executes to answer the given question. The box containing filter and subtract represents the block of functions that are chained for multihop reasoning.

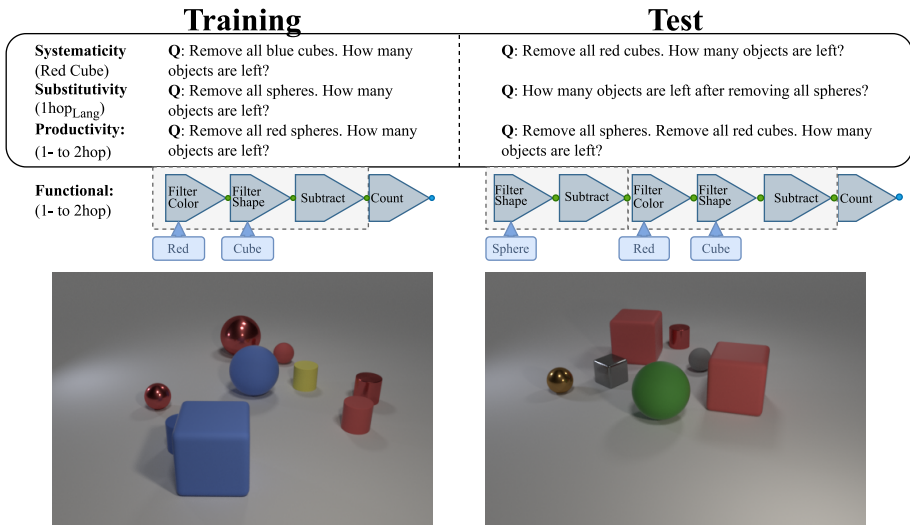


Figure 6.2: Three examples from our generalization splits, showing splits on productivity, systematicity, and substitutivity. The last row shows the functional programs for the function generalization examples from 1- to 2hop. The function blocks in dashed boxes illustrates reasoning hops for the corresponding problems.

All the details on extensions of the CLEVR language are included in the Github repository <sup>1</sup>. We use the scenes from CLEVR-Math and create 10 instantiations per scene, adjusted so that data splits based on multiple templates (e.g. as with 1+2hop) sum to 10 instances. The CLEVR instantiation engine imposes a uniform distribution, resulting in close to equal distribution over templates, attributes, and an answer distribution of  $10 \pm 1\%$  over 0-10. The training, validation, and test data are then put together in a ZIP file and made available through a Huggingface dataset. The Huggingface data loader is used to load multiple test sets for each split, so that, e.g., a model trained on 1hop questions is tested on 1-,2-, and 3hop questions in parallel.

### 6.2.1 Systematicity: Attribute Generalization

We create 5 attribute generalization splits where the training data does not contain any questions involving objects with a certain set of attributes, such as *small blue cubes* (see Table 6.1 for a complete list). The questions are all 1-hop questions, to isolate the compositional generalization over attributes from function generalization. The training data contains no questions on *red cubes*, however, they are present in the scenes.

<sup>1</sup>Github repository and Huggingface dataset will be made public with the final version.

Split	Description
Counting	No removals, only counting objects.
1hop	Remove all Xs. How many objects are left?
2hop	Remove all Xs. Remove all Ys. How many objects are left?
3hop	Remove all Xs. Remove all Ys. Remove all Zs. How many objects are left?
Red Cube <sub>Lang</sub>	No questions with <i>red cubes</i> .
Red Cube <sub>Vis</sub>	No <i>red cubes</i> in images or text.
Large Cylinders	No <i>large cylinders</i> .
Matt Spheres	No <i>matt spheres</i> .
Small Blue Cubes	No <i>small blue cubes</i> .
Small Yellow Metal Spheres	No <i>small yellow metal spheres</i> .
1+2hop	Combination of 1hop and 2hop.
1+3hop	Combination of 1hop and 3hop.
1+2+3hop	1-, 2-, and 3hop problems.
Spatial 1hop	1hop questions to remove objects in spatial relation to a unique object.
Language Complexity 1hop	1hop with 10 paraphrased versions of the question.

Table 6.1: Data splits for function and attribute generalization. The first segment shows the core tasks, the second all attribute splits, and the last segment contains the splits used to investigate the impact of different types of complexity on generalization.

## 6.2.2 Productivity: Function Generalization

Complementary to the length generalization experiments in, e.g., B. Lake and Baroni (2018a), we test productivity function generalization splits for generalizing to longer and shorter chains of functions than those trained on. Figure 6.2 illustrates how the split with 1hop questions for training and 2hop questions for testing are used to evaluate function generalization. If a model trained on 1hop questions performs well on the 2hop test data, we say that it generalizes the subtraction function well. However, 1hop questions without additional context says nothing about the possibility to chain the subtraction function indefinitely. If we give no inductive bias to say that the function can be applied in longer or shorter chains, it is a strong assumption that a model will learn to generalize. Therefore, we also create the 1+2/3hop (1-hop combined with 2-hop or 3-hop) data splits, since here the data suggests that at least the function can be applied in a variable number of steps. We also note that 2- and 3-hop questions inherently address the bag-of-words issue identified by Z. Wu et al. (2021) , since *Remove all blue cubes. Remove all red spheres.* rely on the pairing of attributes in the two subtasks. The causality of the operations is also impor-

tant, as the overlap between *small cubes* and *blue cubes* might be non-empty. This would result in merely counting the two sets (small cubes, blue cubes) and removing the union would result in the wrong answer. Our 2hop, 3-hop, and lang-1hop questions also introduce more linguistic complexity, mitigating the issues identified by Qiu et al. (2021). To focus on function generalization, the attributes are restricted to shapes and colors in these experiments.

### 6.2.3 Substitutivity: Linguistic Variations, Cross-Modal Influence and Spatial Relations

In this section we describe three splits aimed at understanding the language understanding capabilities of our models along three dimensions: (1) linguistic variations to the question (2) ability to understand the question in the absence/presence of the visual concept (3) ability to understand constraints specified via spatial relations in the question. For (1) we add 10 variants of the 1hop questions with the same semantics in the *Language Complexity 1hop-split*. The variants are listed in Table 6.2. Examples include *How many objects are left after removing all Xs?*, changing the order of the operation (remove) and the query (how many), and thus resulting in questions with completely different dependency parse trees. This linguistic complexity is much harder than that in the original CLEVR data, where synonyms and optional phrases provide variation in the questions but never change the overall structure of the parse trees. Figures 6.3 and 6.4 contains the dependency parse trees. Additionally, it becomes harder to pattern match attribute keywords to solve the problems since there is no fixed order mapping to the task. Thus, our language complexity split is significantly more challenging compared to previous work. This split is closely related to the substitutivity aspect of compositional generalization (Hupkes et al., 2020b; Pagin, 2003).

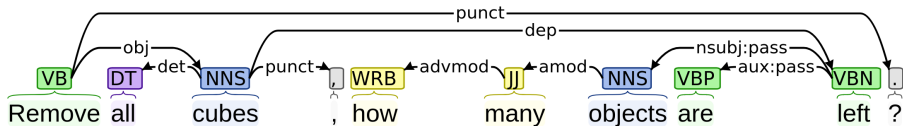


Figure 6.3: Dependency tree for the original formulation of 1-hop questions, seen in Table 6.2. Generated from the text *Remove all cubes, how many objects are left?* using <https://corenlp.run/>.

For (2), we compare *Red Cube<sub>vis</sub>* to *Red Cube<sub>Lang</sub>* to see the how much the attribute generalization depends on novel attribute compositions being present visually. Complementary to the first *red cube* split, this split has no images showing red cubes. This way, we can measure the difference in performance on the two splits as a proxy to determine the effect of novelty also in the visual domain.

Finally for (3), to investigate the impact of function diversity on general-

Original: Remove all $\langle X \rangle$ s. How many objects are left?
1. Remove all $\langle X \rangle$ s [from the scene]. How many objects are left [in the scene]?
2. How many objects are left after removing all $\langle X \rangle$ s [from the scene]?
3. Remove all $\langle X \rangle$ s. How many objects are left?
4. If all $\langle X \rangle$ s are taken away, how many objects remain?
5. What is the total count of objects left after all $\langle X \rangle$ s are discarded?
6. Once all $\langle X \rangle$ s are subtracted, how many objects are left?
7. Subtract all $\langle X \rangle$ s. How many objects are there now?
8. Subtract all $\langle X \rangle$ s. How many objects are left?
9. How many objects would be left if all $\langle X \rangle$ s were removed?

Table 6.2: Linguistic variations used to investigate effect of syntactic complexity on compositional generalization.  $\langle X \rangle$  is a placeholder for  $\langle Z \rangle \langle M \rangle \langle C \rangle \langle S \rangle$ .

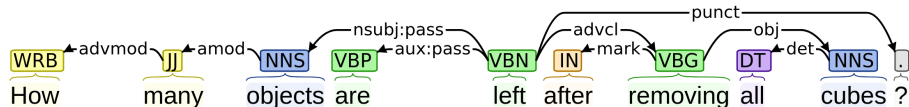


Figure 6.4: Dependency tree for paraphrasing 2 in Table 6.2. Generated from the text *How many objects are left after removing all cubes?* using <https://corenlp.run/>.

ization, our *Relational 1hop*-split includes spatial relational functions such as *to the left of* to increase the complexity of the function space. The example “*Remove all cubes to the left of the blue sphere. How many objects are left?*” shows how the split uses a unique object (*blue sphere*) to restrict the set of *cubes* using the *left-of* function.

## 6.2.4 Models

For our experiments, we evaluate our modified NS-VQA described later in this section, the CLIP-based model used in Lindström and Abraham (2022b), and ViLT (W. Kim et al., 2021) on our compositional generalization splits. CLIP and ViLT are pretrained on large corpora of text-image pairs; both with a image-text matching objective, and ViLT with an additional masked language objective.

NS-VQA is trained in two steps; an initial supervised step on 100 samples per template for 10 000 steps, and then using reinforcement learning with REINFORCE on all data for 50 000 iterations. We make two changes to the architecture for our benchmarking; 1. a mutable scene representation in the execution engine, and 2. a modification to the preprocessing program parser to

produce execution trees that rely on the managed state. Figure 6.5 illustrates how the extended NS-VQA uses the mutable scene representation when removing and counting objects. In the original architecture, NS-VQA relies on two variables to manage the intermediate results from previous functions. Since the CLEVR questions only query a given scene state without manipulations, our modifications are necessary for NS-VQA to execute multiple reasoning steps. Without our modifications, NS-VQA is unable to correctly parse 2- and 3hop questions. We use the same hyper parameters as Yi et al. (2018) for all experiments.

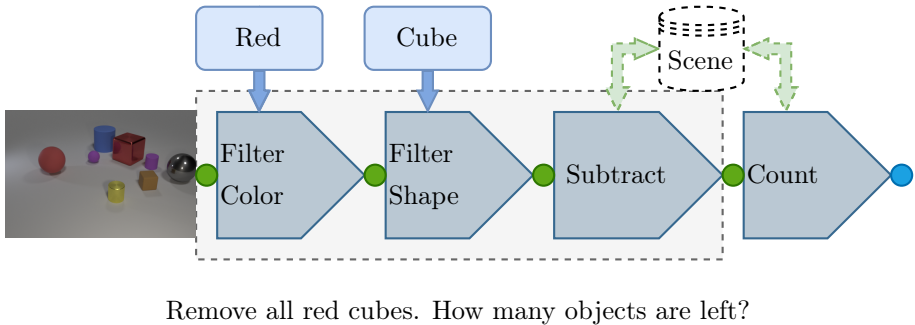


Figure 6.5: An example question and the corresponding functional program in the CLEVR diagram language (Johnson et al., 2017). The dotted scene represents the internal representation used by our modified NS-VQA.

We train ViLT and CLIP for 50 epochs using all data, and a batch size of 128. Both models are trained using the entire dataset of 50 000 training samples, with 15 000 validation samples, and 1000 test samples for each split. The results report the mean and standard deviation over 5 runs for each experiment for all models.

**Reproducibility** The code used to generate our data splits, run the experiments, and the modified NS-VQA will be made available through Github in the final submission. The models were trained on a GPU cluster using NVIDIA A100-cards. Each experiment is run 5 times each, and uses the fixed seeds 42, 1984, 1972, 72, 365 for model initialization and data sampling to give more reliable results and increase reproducibility. The experiments uses NVIDIA driver version 525.125.06 with CUDA 12.2, and Pytorch 2.0.1+cu117 and runs on Python 3.9.

## 6.3 Results

Table 6.3 shows model accuracy on the core tasks and attribute generalization splits.

Split	NS-VQA	CLIP	ViLT
<i>Task baselines</i>			
Counting	98±1	28±1	78±3
1-hop	100±1	53±1	75±2
2-hop	100±0	35±1	86±1
3-hop	100±0	41±0	93±1
spatial-1hop	17±8	53±1	48±1
lang-1hop	42±5	60±1	52±1
<i>Attribute comp. gen.</i>			
Red Cube (Lang)	100±0	48±12	65±3
Red Cube (Vis)	100±0	50±5	63±3
Large Cylinders	85±30	49±12	66±3
Matte Spheres	100±0	39±7	76±2
Small Blue Cubes	100±0	50±15	56±3
Small Yellow Metal Spheres	100±0	48±14	64±4

Table 6.3: Accuracy of task baselines and novel attribute composition splits.

We observe that our modified NS-VQA generalizes perfectly on most splits, overcoming the issue identified in Lindström and Abraham (2022b). NS-VQA outperforms both neural models on most tasks, with the exception of *spatial-1hop* and *lang-1hop*. The performance on *counting* follows 1hop performance for NS-VQA and ViLT, with CLIP showing a large degradation from 1hop. Since only learning to count is strictly easier than the 1hop task, *counting* can be viewed as an upper bound on the 1hop performance. ViLT achieves similar performance on pure counting versus 1hop, suggesting that the difficulty for ViLT lies in counting objects. ViLT learns longer reasoning chains better, going from 75% to 93% accuracy on 1hop versus 3hop, approaching the performance of NS-VQA. NS-VQA and ViLT both struggle with the *spatial-1hop* split, with NS-VQA dropping around 80% compared to the regular 1hop split and ViLT dropping around 27%. For *lang-1hop*, CLIP seems to benefit the language diversity, as it achieves higher accuracy on *lang-1hop* than *1hop*. Conversely, NS-VQA drops performance by almost 60% and ViLT by around 27% on the *lang-1hop* split.

TODO Cover empty string results

### 6.3.1 Attribute Generalization

NS-VQA generalizes perfectly on all attribute splits apart from *Large Cylinder*, where the performance drops with a large variance over the 5 runs. This failure is explained by a catastrophic failure with 30% accuracy with seed 42. Inspecting the data used in the supervised training reveals that the failure is not due to lack of samples containing *large* or *cylinder*, an issue identified in preliminary experiments. While CLIP has the worst in-distribution performance on



Model	Trained on	Count	1-hop	2-hop	3-hop
<b>NS-VQA</b>	1-hop	8±5	100±1	3±2	2±2
	2-hop	0±0	27±0	100±0	7±0
	3-hop	9±6	20±7	59±20	100±0
	1+2-hop	19±5	76±9	54±1	22±3
	1+3-hop	21±5	78±7	68±15	58±3
	1+2+3-hop	19±11	95±5	55±38	31±20
	spatial-1hop	12±3	17±8	6±2	5±4
	lang-1hop	24±1	57±2	0±0	0±0
<b>CLIP</b>	1-hop	6±1	53±1	14±0	1±0
	2-hop	16±0	11±1	35±1	25±1
	3-hop	20±0	5±0	20±1	41±0
	1+2-hop	9±1	51±1	22±1	19±1
	1+3-hop	8±0	49±1	19±1	25±2
	1+2+3-hop	10±0	55±1	20±1	22±1
	spatial-1hop	8±0	53±1	13±1	0±0
	lang-1hop	7±0	60±1	5±0	0±0
<b>ViLT</b>	1-hop	23±3	75±2	14±7	0±1
	2-hop	19±2	71±2	86±1	22±2
	3-hop	16±3	70±4	78±2	93±1
	1+2-hop	28±1	53±2	72±3	27±6
	1+3-hop	26±1	56±1	82±2	44±1
	1+2+3-hop	27±3	78±3	84±2	82±3
	spatial-1hop	18±1	46±1	0±0	0±0
	lang-1hop	17±0	52±1	0±0	0±0

Table 6.4: Model accuracy on function generalization over multihop questions, averaged over 5 runs. In percentage, higher is better. Each row represents training on, e.g., 1+3hop and the performance on the  $n$ -splits.

1hop questions, the performance does not drop with significance. This could be attributed to similar objects being part of the pretraining objective, given the vast corpus used (Radford, Kim, Hallacy, Ramesh, Goh, Agarwal, Sastry, Askell, Mishkin, Clark, et al., 2021a). Investigating the overall poor performance of CLIP, and the large  $\sigma$  on most attribute splits is interesting future work. The accuracy for ViLT drops 10% on 3 of the splits, with 0% drop on *Matte Spheres* and a 20% drop on *Small Blue Cubes*.

**Scaling Laws** To investigate how sensitive each architecture is to the amount of training data, Table 6.5 shows the model accuracy over 10%, 50%, and 100% data for the held out attribute compositions combined. NS-VQA is not sensitive to the amount of data and needs very little data to be successful, consistent

Model	10%	50%	100%
NS-VQA	100±1	100±0	100±0
CLIP	49±1	53±1	56±1
ViLT	61±8	63±8	70±3

Table 6.5: Accuracy on held out attribution compositions when trained on 5000, 25000, and 50000 samples.

with previous results on CLEVR (Johnson et al., 2017). CLIP and ViLT both perform better with more training data.

TODO Add plots of loss et c.

### 6.3.2 Function Generalization

The function generalization results are shown in Table 6.4, with each column representing out of distribution model performance when trained on the split in the first column. When trained on 1- or 2hop questions, all three models fail to generalize to longer reasoning chains, with CLIP and ViLT achieving accuracies 5-10% above random. When trained on 2- or 3hop questions, ViLT shows the best generalization to fewer hops, with the model trained on 3hop only loosing about 4% on the 1hop questions compared to in distribution. NS-VQA shows a steep decline in performance generalizing from 3hop to 1hop questions, with a loss in accuracy of around 80%.

Looking closer at the output of NS-VQA, the answer accuracy decouples from the the program accuracy on all out of distribution tests, with close to 100% for  $n$ -hop to  $n$ -hop, but 0% for most out of distribution tests. This is explained by looking at the predicted 1hop programs from NS-VQA trained on 3hop questions. For the 1-hop sample *Remove all blue cubes. How many objects are left?*, the 3hop-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 the two “extra” remove operations are of objects that are not present in the scene, the reasoning engine of NS-VQA will still answer the question correctly. This is consistent with the results in Table 6.4, as the probability of two remove operations having no effect on a scene and thus producing a correct answer is significantly higher than for one.

**Implicit Learning of Counting as Subtask** Table 6.3 show the model accuracy when trained purely on counting the objects. Counting the objects in a scene as an implicit task seems to improve when the model is subjected to more complexity (e.g. learning 1-, 2- and 3hop problems simultaneously). We see this in Table 6.4 as both NS-VQA and ViLT achieve significantly better accuracy on the pure counting task only when subjected to training data with mixed-hop questions. However, the accuracy on counting as an implicit task is still far from the performance when trained to count explicitly. We also see

that CLIP and ViLT show opposite patterns on counting when moving from 1- to 3hop, with CLIP increasing its performance while ViLT degrades.

### 6.3.3 Complexity Driven Generalization

Table 6.4 shows model performance when trained on mixed-hop splits, as well as with spatial functions and more diverse language. NS-VQA struggles to learn different reasoning lengths jointly, e.g. shown by the results on 1+2+3hop where 1hop performance is close to perfect, but 2- and 3hop performance being significantly lower and with large variance. The same pattern can be seen for 1+2hop and 1+3hop. Conversely, ViLT generalizes better to 3hop when trained on 1+2hop than either 1- or 2hop, but the in-distribution accuracy is overall lower. For ViLT, the *generalization* to 2hop sees almost no degradation when trained on 1+3hop compared to 2hop. On the other hand, NS-VQA sees a large drop in performance for 1+3hop to 2hop.

All models struggle with the *spatial-* and *lang-1hop* splits, both in terms of lower in-distribution performance and generalization to longer splits. ViLT generalizes worst to 2hop and 3hop with 0%, but both CLIP and NS-VQA get around or below random performance.

## 6.4 Discussion

Our experiments show that NS-VQA generalizes to novel attribute compositions with near-perfect accuracy. The neural methods perform much worse, showing significant degradation in performance for novel compositions. We have shown that all three models generalize poorly to longer reasoning chains, with similar patterns for the neural methods and complete failures of less than random performance for NS-VQA. For NS-VQA, generalizing to fewer hops see 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, but shows overall poor performance. This can be attributed to CLIP only being trained to align text and images, and not masked language modeling. ViLT shows strong performance generalizing to fewer hops, achieving performance comparable to in-distribution tests.

In our experiments with mixed-hop training, we show how ViLT benefits from complexity similarly to previous work, whereas NS-VQA struggles to learn more than on hop jointly. One previous argument in the domains of neuro-symbolic language learning and compositional generalization, is that the language component fails on such tasks because it cannot capture the complexity of language sufficiently. Our results on function generalization indicate that NS-VQA does indeed suffer from this, where the more powerful ViLT transformer is able to consistently generalize to shorter chains. However, all three methods suffer on our language complexity split, confirming that our language

complexity split is more challenging than previous work. This also shows that there is room for improvement for in-distribution learning on these types of tasks.

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 any K-hop questions.

An important future research direction is to look at curriculum learning based on attributes and program complexity. Although it is known that curriculum learning can lead to more efficient training, the effects on compositional generalization is unclear. Given that NS-VQA and ViLT both see a significant increase in performance on the counting subtask when subject to mixed-hop training, curriculum learning could be beneficial for this task.

### 6.4.1 Conclusion

In this paper we have developed a benchmark to test compositional generalization in multimodal mathematical reasoning. We have introduced data splits for compositional generalization that address both systematicity and productivity. For each of these splits we compare the state-of-the-art neuro-symbolic model NS-VQA with recent neural models ViLT and CLIP. We showed that while NS-VQA is superior in attribute generalization (systematicity) it lags behind the neural models like ViLT for functional generalization (productivity). We additionally demonstrated some splits are hard for either model family and also presented ablation experiments that contrast the various models on different amounts and complexities of training data. We believe our work will help inspire future research on better architectures that can compositionally generalize on diverse multimodal applications.

## 6.5 Probing CLEVR Attribute Compositionality

The experiments with compositional generalisation splits earlier in this chapter evaluate models based on their external behaviour. This section will provide complementary experiments focusing on the internal properties of models using the probing techniques described in Chapter 4.

Probing experiments have been used before to gain insight into the compositional generalisation of models. Chapter 3 covered how Lovering and Pavlick (2022) and Pavlick (2022) uses probing to test compositionality in deep learning models for vision tasks. A key takeaway important for this work is that probing showed that the models tested saturated the performance on one attribute before moving to the next. The order in which the attributes were learned cor-

responded to their visual complexity, learning horizontal versus vertical lines before learning to classify fuzziness. Sikarwar et al. (2022) use mechanistic interpretability of Transformer attention to gain insight into how multimodal transformers learn to compose on ReaSCAN. They propose an extended set of data splits targeting compositional generalisation, and show how multimodal transformers are capable to generalise systematically to some extent. They complement the performance over these splits with a linear probe using the self-attention weights, Figure 6.6 shows the probing results. This indicates that the learned model does good job but does not manage to disentangle perfectly.

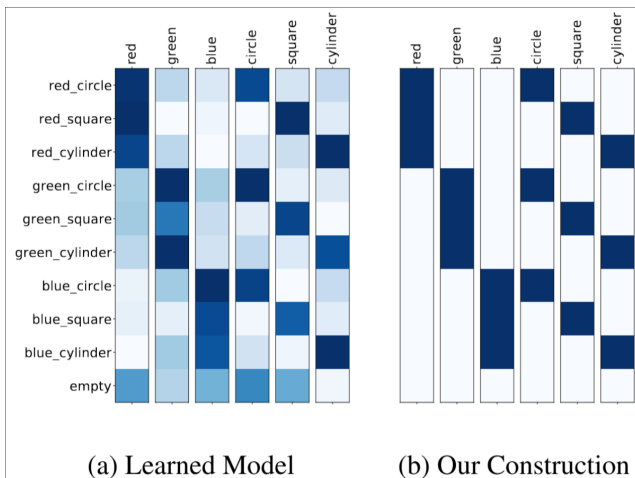


Figure 6.6: Image from (Sikarwar et al., 2022) todo ask for permission.

Another relevant use of probing is shown by Tenney et al. (2019), where the authors argue that BERT rediscovers the classical NLP pipeline. Each layer of BERT is probed for how much each layer contributes to the performance on a certain NLP task (such as part-of-speech tagging). Figure 6.7 illustrates the results, indicating that the earlier layers are responsible for the NLP tasks with lower complexity.

The experiments introduced this chapter combines the ideas of Lovering and Pavlick (2022), Sikarwar et al. (2022), and Tenney et al. (2019), probing each layer for how well they represent CLEVR attributes. Going back to the Principle of Compositionality, in order to apply a function over parts in a structure the parts must be distinguishable as symbols at some point during the computation. This intuition motivates the following experiments which probes the embeddings of the models to recover the CLEVR attributes over which the compositional generalisation splits were created. If a linear probe recovers the concepts with high degree of certainty, then the model seems to have access to such individual parts to combine and evaluate. If a concept

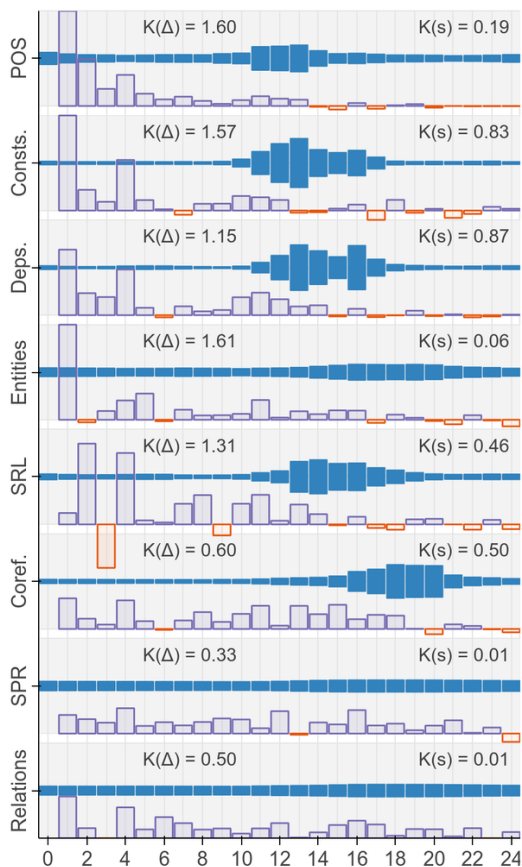


Figure 6.7: Probing of each layer in BERT shows how NLP tasks interleave roughly corresponding to a classical NLP pipeline. The performance on Part-of-Speech tagging (POS) comes from the earlier layers, whereas the performance on a more complex task such as co-reference resolution comes from the layers at the very end. TODO ask for permission

is well-represented in the earlier layers but seems to be absorbed in the later ones (i.e., being non-recoverable), we can still argue that this is a step in the evaluation of the parts. However, if no layer represents the concept distinctly, then the concept is not accessible and this suggests that the model does not fulfil the Principle of Compositionality.

### 6.5.1 Experiments

1. Baseline: Train probe for each attribute on the pretrained model without finetuning

2. Probe: Linear probe
3. For each dataset  $d$  in [holdout-1hop,2hop,allhop,lincomp,spatial]:
4. Train one epoch on dataset  $d$ :
  - (a) for each attribute  $a$ :
    - i. Train probe  $p_{i,a}$  on training data
  - (b) For each attribute composition from holdout split:
    - i. Evaluate accuracy of probe  $p_{i,a}$
  - (c) Checkpoint model
1. Train models on allhop, lingcomp, 1+2hop, spatial
2. Probe each of these models for attributes, does it differ?
3. Relationship between probing and comp. gen.
  - Correlation between probing and comp. gen. results

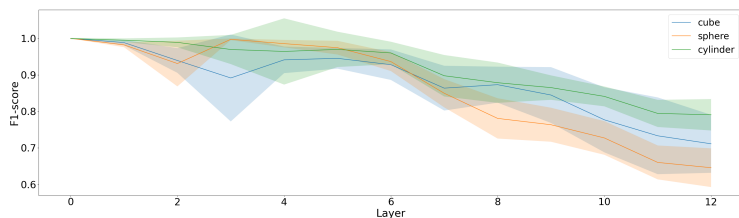
These probing experiments can tell us whether there is a correlation between probing performance and the compositional generalization accuracy. Figure 6.8 shows the probing performance per layer of ViLT for CLEVR attributes. The results when probing ViLT for the number of hops, seen in Figure 6.8c, can be explained by the fact that some questions use *and* instead of a period, resulting in a misleading probing task. However, the fact that the F1-score approaches that of 1- and 3hop could mean that *and* is associated with 2hop further into the network. This can be interpreted as ViLT composing the meaning from the parts and the structure.

If we assume that accuracy in probing tasks translate into how well a model composes compositionally, then it should be possible to predict compositional generalisation performance from probing accuracy. If the probing accuracy is lower for *red* and *cube* than on *blue* and *sphere*, and the model follows the principle of compositionality, then we can expect the model to generalise to unseen *red cubes* worse than *blue spheres*. Figure 6.9 shows a similar experiment, where 2-gram attributes are probed for.

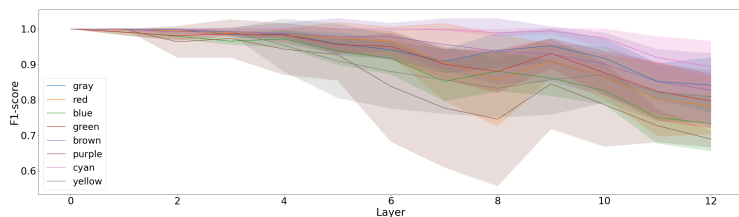
Figure 6.8b shows that probes for red and purple have the lowest F1-score of around 0.75, with brown and cyan having the highest at close to 0.9 F1-score. Figure 6.8a shows that probes for cylinder has an F1-score of around 0.85, with both cubes and spheres getting around 0.70. If the hypothesis about the correlation between probing scores and compositional generalisation is true, then ViLT should generalise better to brown or cyan cylinders than to red cubes or purple spheres.

Recent work to write about

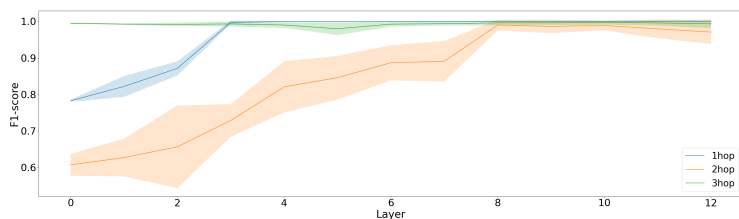
- (Weiss et al., 2021)



(a) Probing of shapes



(b) Probing of color



(c) Probing number of hops in the input question.

Figure 6.8: Probing the layers of ViLT for shape, color, and the number of instructions in the original question.

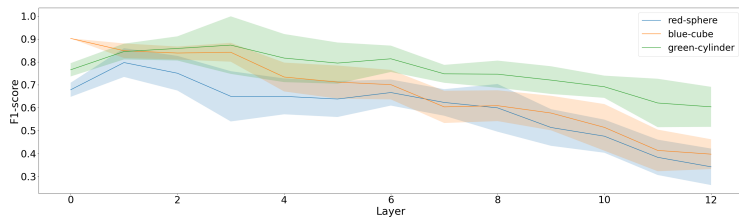


Figure 6.9: Probing ViLT for 2-grams.

- (Keysers et al., 2020)
- (Belinkov, 2022)



- (Kobayashi et al., 2020)



## Chapter 7

# Using Concept Hierarchies to Improve Compositional Generalisation

Example of how to use quotes at the beginning of chapters

---

*dali*

- Effect of curriculum on comp. gen., in terms of
  - Final accuracy
  - How quickly good performance is reached
- Probing results could perhaps help explain?
- One pseudoword experiment with *blargh* = *small blue sphere*
  - Look at more than one level of abstraction
- Relate these experiments to Eustace and Carey
  - Extend curriculum to include tasks for identifying objects, counting, et c.
  - Relate this to (Aissa et al., 2023; Askarian et al., 2021)
- Ideally add support for more models in evaluation
- Check effect of L2 norm on NS-VQA and ViLT

## 7.1 Language Learning in Developmental Psychology

TODO Dupoux (2018) gives an overview on how results from cognitive science in infant language learners can be used to build better language systems.

Compositional generalisation in multimodal language models is studied mostly through implicit metrics in unsupervised settings. Recent systems such as CLIP, DALL-E 2, Stable Diffusion, et c., are impressive in the way they process novel combinations of concepts, but it is difficult to verify underlying structures and mechanisms allowing these compositions. One difficulty when analysing these capabilities is that we do not know exactly what data a system was trained on. Therefore, an experimental setup using synthetic data means we can investigate with high control. However, for ecological validity, we also want the data to have enough natural properties to show transferability to natural domains. Previous work does this with abstract 2D concepts and pseudowords (B. M. Lake, 2019; B. M. Lake et al., 2019; Ruis et al., 2020). In this work, we propose a compositional generalisation benchmark in a 3D environment using hierarchical pseudoword concepts. With pseudowords, we can ensure that the specific concept has not been seen before associated with that word. Since vision models can achieve perfect accuracy on the CLEVR dataset, this means that we can assume that the basic properties such as shape and color is already known. The hierarchical aspect means that concepts build on each other, and that we can investigate whether a model learns basic building blocks first before composing more complex concepts. This then means that we can more easily construct curriculum learning setups. This allows us to investigate the impact of relying on such structures.

Pseudoword setups has a long-standing place in linguistics research, most famously with the Wug Test introduced by Berko (1958) in 1958. The test involves 27 questions where pseudowords are introduced, and the task is to use it in a novel grammatical role. Each question is posed on a card with an illustration of the pseudoconcept. The example that gave the test its name is seen in Figure 7.1, showing how the word *wug* (denoting a bird) is supposed to be used in plural. Using 56 children age 4–7, the experiments show how the subjects can apply morphological rules to novel words correctly with fairly high degrees of accuracy. Regarding the difficulty of the task, we refer to this statement by the authors; *Answers where willingly, and often insistently, given* (Berko, 1958). One important takeaway from the Wug Test is that humans learn rules that can be applied to novel words in a zero-shot situation, and that we are able to compose previous knowledge to do so.

Carey and Bartlett (1978) is another example from developmental psychology, investigating how children learn a single new word. The authors detail the process of acquiring a word by different pieces of information. According to Carey and Bartlett (1978), a learner

- makes a new lexical entry, noting the word and in which language

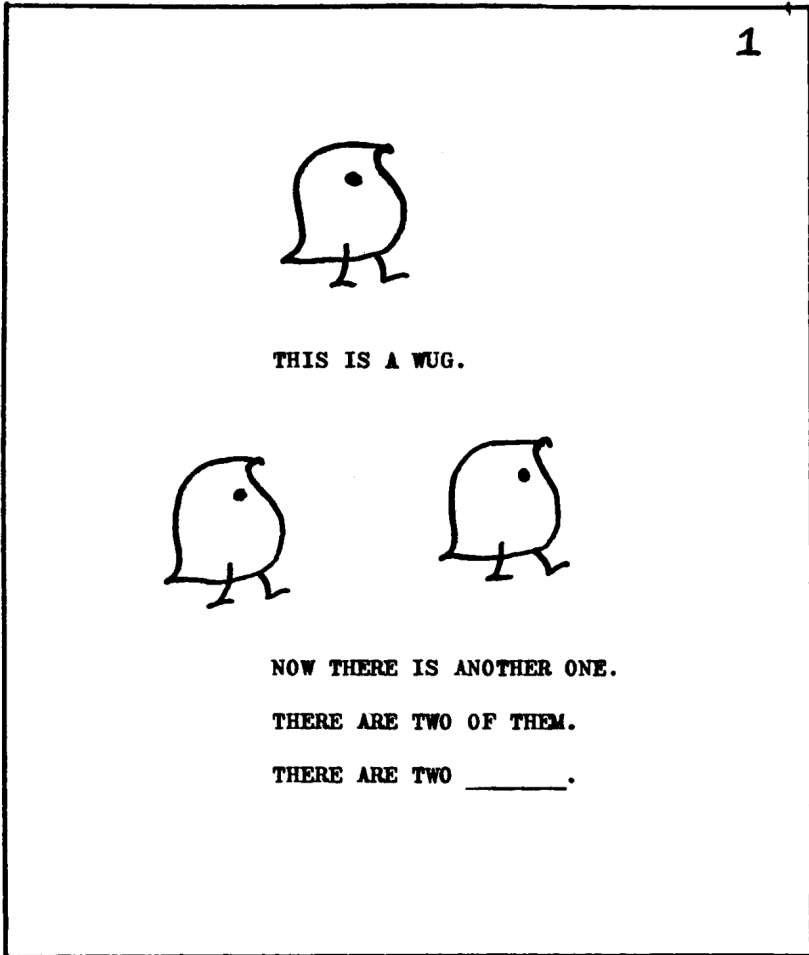


Figure 7.1: Example of card from the Wug Test (Berko, 1958), showing a task of applying morphological rules to a novel (pseudo)word.

- learns the syntactic subcategorization, e.g. that it is a verb,
- relate it to other known words through super-, hypo-, and hypernyms
- ground the word in the real world,
- differentiate this concept from previous concepts by e.g. breaking it out as a different species of animal.

In their experiments with 19 children, the subjects were told that *chromium* was the word for the color *olive green*. The procedure involved the following tasks;

- Introduction to the word “chromium”,
- Baseline vocabulary assessment,
- Olive sorting task,
- Olive naming task,
- Chromium comprehension task, and
- “Chromium” hyponym task.

In the sorting task, the children were tasked with using their newly acquired knowledge about “chromium” to solve the physical task of matching colors to boxes among similarly odd colors. With plain red, green, and yellow, it can be expected that the children confused the concept of chromium meaning olive green with it meaning something like *the odd color out*. In the comprehension task, the children were tasked with pointing at three colors, one of which was chromium, controlling whether they had properly learnt a referent for “chromium”. In the hyponym task, the experiment controlled for whether the children had learnt that “chromium” indeed referred to a color. It is important to note here that these tasks cover multiple different aspects of understanding a word, rather than only the textual understanding aspect as in the Wug Test. From their experiments, the authors distinguish between two phases; the *fast mapping* and *drawn out mapping*. Fast mapping takes place in the first few encounters, and gives only a small subset of the information outlined above, such as its language and supernym. A more complete understanding of the word instead requires both more encounters and more time. Their results show that the subjects could use the new word after only one exposure, but that the second encounter was necessary to perform well on the outlined tasks. One takeaway is that we learn a sufficient amount of information about a new word with very little data by utilizing existing understanding, but not enough to understand and use it fully without exposure over longer periods of time. In a sense, it can be expected that we learn certain aspects well at the first encounter, but that that more complex notions take more time.

B. M. Lake (2019) and B. M. Lake et al. (2019) use similar ideas to construct tests for compositional generalisation skills in humans. Their work involves learning words for objects and functions over objects, constructed as 2D images of colored dots in patterns. While the authors reuse the same approach of using pseudowords that are speakable, they restrict the experiments to the pseudoword domain. In the context of AI, using only pseudowords mitigates the problem of information leaking from the training data. For instance, it is difficult to draw any strong conclusions from performing the Wug Test on GPT-derivatives as this is most likely mentioned many times over in the vast amount of data used during training.

T. Brown et al. (2020) show with 6 examples that GPT-3 can acquire new words. This shows how deep learning-based methods can be built to acquire new words, but it does not tell us much about to which extent the new word and associated concept can be understood in relation to existing knowledge. We argue that the only conclusion we can draw is that GPT-3 performs the fast mapping described by Carey and Bartlett (1978).

Similar to the psychology experiments on acquiring a new word by Carey and Bartlett (1978), Eustace (1969) performs experiments with learning a complex concept at different hierarchical levels.

What we can learn from the Wug Test, is that we can use new words instantly with little learning, and therefore we should be able to bootstrap to previous knowledge. When designing a benchmark, we can translate the Wug Test to check whether there are internal structures and rules that can be applied to novel words, or if the model relies on something more fuzzy.

From the Chromium experiments, we learn different aspects of a word with different speed, which means that testing should reflect these expectations. For our benchmark, this means that we can expect a pseudoword to be lexically understood, but that things like hyponyms is expected to take longer to learn.

The first hypothesis is that we can expect to see similar behaviour in language models. The second hypothesis is that constructing the training procedure to build on previous knowledge will be beneficial for training times, and that learning syntactic usage should come before more complex tasks.

TODO SUMMARISE THE ABOVE INTO A LIST THAT CAN BE APPLIED TO OUR BENCHMARK

## 7.2 Concept learning

In Chapter 1 we saw some of the historical debate on what concepts are and how they are useful for artificial intelligence. In this section, we explore recent approaches building on these ideas, mainly from the field of meta-learning.

Meta-learning is a branch of machine learning focused on enabling learning algorithms in *learning-to-learn* from past experiences and transferring knowledge to new tasks. The current interest in the field is partially a reaction to the deep learning paradigm of training from scratch on specific tasks (Hospedales

et al., 2022). By letting a system self-improve the *learning algorithm* itself over different tasks, rather than using a fixed learning algorithm to improve task performance. Learning-to-learn *can lead to a variety of benefits such as improved data and compute efficiency* (Hospedales et al., 2022). Harlow (1949) talks about the construction of *learning sets* as a key aspect of primate learning, stating that

The learning of primary importance to the primates, at least, is the formation of learning sets; it is the learning how to learn efficiently in the situations the animal frequently encounters. This learning to learn transforms the organism from a creature that adapts to a changing environment by trial and error to one that adapts by seeming hypothesis and insight.

Moving from primates to neural networks, Hospedales et al. (2022) formalise the difference between conventional neural networks and meta-learning.

Given a dataset  $D = (x_1, y_1), \dots, (x_N, y_N)$ , neural networks are trained to a minimise a loss function  $\mathcal{L}$  for a predictive model  $\hat{y} = f_\theta(x)$  parameterized by  $\theta$ .

$$\theta^* = \arg \min_{\theta} \mathcal{L}(D; \theta, \omega), \quad (7.1)$$

The parameter  $\omega$  denotes the assumptions on how to learn, including network architecture and training parameters such as the optimizer for  $\theta$ . A meta-learning model is instead tasked with learning  $\omega$  to minimise the loss over tasks  $\mathcal{T} = \mathcal{D}, \mathcal{L}$ .

$$\min_{\omega} \mathbb{E}_{\mathcal{T} \sim p(\mathcal{T})} \mathcal{L}(D; \omega) \quad (7.2)$$

Using these definitions, we can see how conventional neural networks are captured within meta-learning. However, it is important to note that even if we only have one task, the meta-learning objective in Equation 7.2 is still concerned with finding the best assumptions for  $\omega$ . Now, with meta-learning as the backdrop, we will look at meta-learning approaches to facilitate few-shot learning of concepts.

Vinyals et al. (2016) introduce *matching networks* for one shot learning as a method to learn a similarity metric between a query input and a set of labeled examples. The method uses an attention mechanism over learned embeddings of the labeled examples, and classifies unlabeled input using a technique similar to nearest-neighbor search by finding the closest labeled example. The authors show how matching networks can perform one-shot learning on image recognition and natural language processing tasks.

Building on the ideas of Vinyals et al. (2016), Snell et al. (2017) introduce *ProtoNet*, prototypical networks for few-shot learning. Extending on the idea of similarity in an embedding space, prototypical networks work under the assumption that there is a single point in the embedding space that can act as the prototype representation of a class. ProtoNet works by first creating a set of such prototype embeddings from the training data. These prototypes



are then used to classify new data points by computing the distance between a new data point and the prototypes. The model then assigns the new data point to the class of the most similar prototype. Using a few tricks, such as using Euclidian distance instead of cosine similarity, ProtoNets outperform matching networks on a range of tasks.

Following these two embedding space-based methods, Cao et al. (2021) argue that reusable concepts is an important missing piece of the puzzle. They say that *[.] this lack of structure is limiting the generalization ability of the current meta-learners* while referring to B. Lake et al. (2011) and B. M. Lake et al. (2015) and the *importance of compositionality for few-shot learning*. An often used example is how humans use parts to determine the whole, for instance by identifying an animal as a cat by its *whiskers, paws, pointy ears, and fur*. Motivated by this, they introduce *COMET* as a meta-learning method based on learning human-interpretable concepts. Similar to the prototypical networks (Snell et al., 2017), *COMET* uses embeddings of the labeled data to construct a composable set of concepts. However, instead of learning one joint embedding space for all *prototypes*, *COMET* learns individual embedding functions for each concept. Each *concept learner* is used to produce *concept prototypes* similar to the averaged prototype embeddings in ProtoNets. When we want to classify a data point, each concept learner computes an embedding of the data point and we measure the distance of each embedding to the concept prototype classes to determine its class. The method is evaluated on tasks from computer vision (CUBS, Flowers), NLP (Reuters news classification), and biology (Tabula Muris). As an example, they use all hypernyms of a given word in the Wordnet (Miller, 1992) hierarchy to create concepts to base the model on for the Reuters news classification. For the classification of birds in CUBS, concepts such as *beak* and *wing* are given by the dataset. *COMET* outperforms Matching Networks and Prototypical Networks in both 1- and 5-shot settings for all tasks. *COMET* also achieves comparable performance with the other methods even on a small subset of all the concepts. One question now is whether the performance gain comes from this more elaborate architecture, or whether the method bootstraps its performance to the human-defined concepts. To answer this, *COMET* is also tested using unsupervised concepts, still marginally outperforming the baselines while clearly outperforming ProtoNet. Hence, it is not strictly necessary to use human-interpretable concepts for *COMET* to work, but they provide stronger performance. However, human-interpretable concepts are important for interpretability of how a model understands a certain class. For this, *COMET* is evaluated on local and global explanations concerning which concepts are the most important for a given data point or class. In their evaluation, Cao et al. (2021) conclude that *COMET* is consistent with human interpretations.

These three methods all show how meta-learning using basic building blocks can give efficient learning methods. Specifically, *COMET* shows how we can use human-interpretable primitives to compose more complex concepts for classification. Importantly, all these methods require only a few examples of a novel

class to recognise its instances.

With a concept learner setup based on human-interpretable concepts, we do not need to perform experiments like the probing detailed in Chapter 3. Instead, such insights fall more naturally out of the architecture itself with COMET. We can also make a strong case for how such a model handle the introduction of new symbols by adding concept learner as soon as something sufficiently out-of-distribution is seen. One counter argument is that the interactions between and the nuances of real-life concepts are so complex, that these concept learners merely are crude approximators while LLMs can embrace and model that complexity.

While COMET showcases important properties and good performance, the architecture only models one level of abstraction. Hence, to have real world applicability, it might be necessary to nest concept learners hierarchically. If this can be done in a emergent way, similar to what we described in the previous paragraph on new symbols, this might allow the same complexity while producing a human-interpretable model. The benefit of having human-interpretable concepts as with COMET is that there is a much clearer path to how a model can be a shared representation between humans and machines. We saw in Chapter 5 how a shared representation is a key component of building human-centered AI.

## 7.3 Curriculum Learning with Concept Hierarchies

### 7.3.1 Experiments

1. Define curriculum  $cl$
2. Define good-enough criteria  $c$
3. For each level in curriculum:
  - (a) Train model  $m$  until criteria  $c$
  - (b) Evaluate on test data for held out compositions
  - (c) Checkpoint model

Questions:

- How do you define criteria  $c$ ?
  - Plateau in loss or validation data accuracy
  - Threshold loss or accuracy
  - Fixed number of epochs

## Concept Hierarchy

Based on attributes and function difficulty, similar to X. Wang et al., 2022.

## Loss-based

Another approach is to do a loss-based curriculum learning where a subset of the most difficult subset of samples are chosen in each epoch.

## 7.4 Compositional generalisation benchmark using hierarchical pseudoword concepts in CLEVR

Previous work shows us the importance of compositionality for generalisation, and how human intelligence is compositional. We have seen examples of tests for compositional generalisation in language models (e.g. COGS (N. Kim & Linzen, 2020)), and examples for multimodal language models in e.g. (Johnson et al., 2017). However, the benchmarks for multimodal language models have focused mainly on confounding information and n-gram associations (e.g. fixing the color of spheres in training but not testing), rather than complex compositional structures such as those modeled in COGS (N. Kim & Linzen, 2020). This section will detail a compositional generalisation benchmark for hierarchical concepts using some of the ideas from developmental psychology outlined earlier in this chapter. The benchmark is realised with CLEVR using pseudoword concepts that build hierarchically on each other, exemplified in Figure 7.2. In the spirit of the Chromium test by Carey and Bartlett (1978), we devise multiple tasks through which the comprehension of these concepts are tested.

- Determining presence or absence of concept in image
- Performing a task – mathematical reasoning
- Hyponym task

These tasks are constructed to cover different aspects of the concepts, not necessarily corresponding to orders of difficulty. However, given the hierarchical ordering of the concepts, we can construct a curriculum learning setup for learning concepts by order in the hierarchy. Curriculum learning has shown to improve generalisability and the convergence rate during training (Bengio et al., 2009; X. Wang et al., 2022). One central challenge of curriculum learning is how to estimate difficulty in order to create a curriculum. In our case, we can use the hierarchical structure to reflect the complexity of a concept. With a curriculum, we can then compare differences in task performance when training on randomly ordered concepts versus using a curriculum. Beyond performance, we can hypothesise about how a curriculum affects the internal structure of a model to better allow for compositional generalisation.

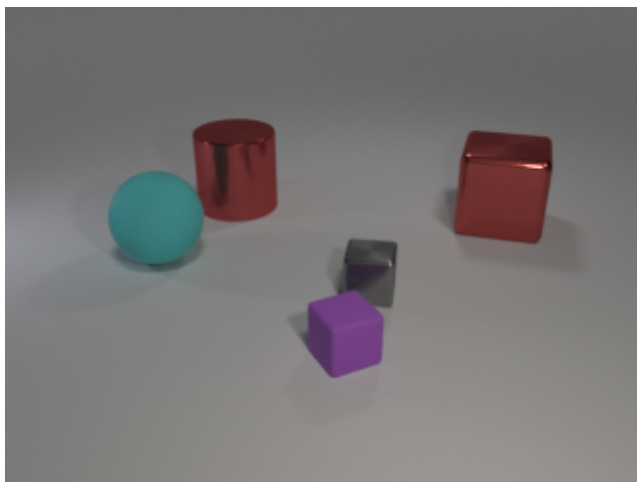


Figure 7.2: A simple example of data generated in CLEVR, where we see two pseudoconcepts; a) a *blargh* – two small cubes next to each other, and b) a *perde* – a large cyan sphere.

Another motivation behind this work is TODO CITATION OF IBM, where abstraction, composition, and recursion are three central characteristics necessary to generalise. We can also look to examples from reinforcement learning, where Zhao et al. (2022) propose a reinforcement learning method that achieves compositional generalisation in a object oriented domain. The authors borrow ideas from curriculum learning, as they describe three stages (object extraction, action binding, and transition modeling between properties) of learning using their model. However, this work does not address hierarchical compositions but only a flat hierarchy.

Askarian et al. (2021) look at the effects of three different curriculum learning strategies on performance in relation to amount of data and training costs. Their claim is that “*curriculum learning effectively improves low data VQA*”, showing on subsets of CLEVR how CL and L2-norm regularisation can drastically improve performance when training with only 20% of the original data. They define three different curriculum learning strategies using complexity criteria based on program length, answer hierarchy, and hard examples. The *first strategy* is based on the intuition that the length of a question is an indication of how difficult it is to answer. As a proxy for length, this strategy measures the length of the program as given in the CLEVR dataset (i.e., `filter_colorblue` counting as one operation). The *second strategy* uses an answer hierarchy created by the authors themselves. The intuition is that a learner first learns the answer type, e.g. that a question requires a number as its answer. From this intuition, Askarian et al. (2021) constructs a hierarchy of the answer types, shown in Figure 7.3. Hardness is then defined as how far from the hierarchy

root an answer is. Their *third strategy* uses examples that yield high learner loss. This makes it the only strategy to have a dynamic hardness criteria, since the loss will change for hard examples over time as they become easy for the model to answer.

As further insight into the benefits of curriculum learning for visual question answering, Aissa et al. (2023) proposes a Neural Module Network (NMN) method for Visual Question Answering that uses predefined cross-modal embeddings and curriculum learning to reduce the cost of training and the amount of training data while still achieving good accuracy. They show how their curriculum learning strategies allow the NMN model to achieve the same performance using half of the data and 18 times less compute. Their main hardness criteria is a combination of the number of objects in a scene and the program length of a given question. They complement this hardness criteria with pretraining on random examples, and two weighting strategies to 1) achieve uniform distribution over the different answer types, and 2) weigh examples proportional to the sum of the average losses of the program modules corresponding to the question (this focuses the model on hard examples). These strategies all follow the same spirit as the strategies presented by Askarian et al. (2021). Aissa et al. (2023) move away from CLEVR into the more natural domain of GQA, to provide a more challenging and complex setup.

Keyzers et al. (2020) formalise *distribution-based compositionality assessment* (DBCA) as a method to *assess the adequacy* of a dataset split for measuring compositional generalisation. They introduce two guiding principles; 1) similar atom distribution, and 2) different compound distribution, and argue together with Saxton et al. (n.d.) that automated rule-based generation of data brings the control necessary to adhere to these principles. Hence, we build on these ideas when constructing our benchmark.

Another important continuation is to provide tasks with increasing difficulty, either as more steps, higher scene complexities, or more complicated operations to learn. A task that is difficult in general is the introduction of new symbols. From a grounding perspective, it is key that a word is associated with the correct real world referent. From a compositional perspective, it is key that a new concept is composed of previously known concepts where appropriate. A key component of intelligence is that we can generalise beyond the domain knowledge we have. For AI systems, this means that the domain knowledge encoded at creation might be important inductive bias, but cannot restrict a model's ability to interact and to learn.

## Fundamentals

- Mapping n-grams of properties (blue, small) to "natural"\* pseudowords[0,4,5] as pseudoconcepts - small blue sphere -> blargh
- Compose a hierarchy of pseudoconcepts
- Hierarchy should reflect (arbitrary) categories

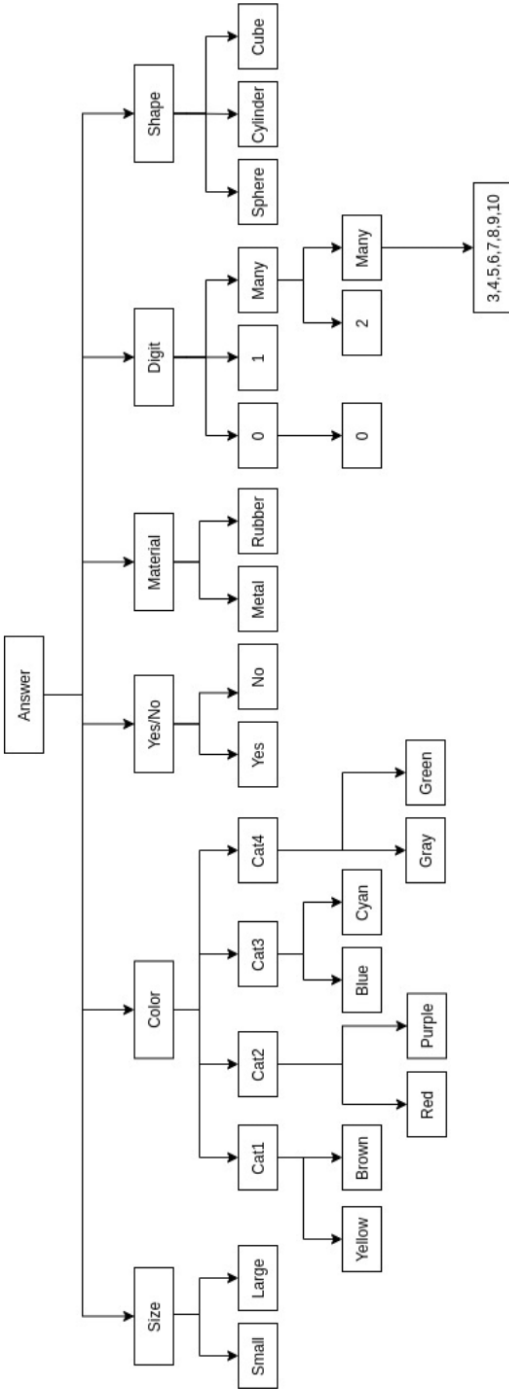


Figure 7.3: The answer hierarchy for CLEVR, as introduced by Askarian et al. (2021). Used to define hardness for their curriculum learning strategy.

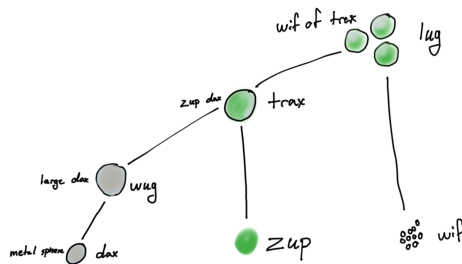


Figure 7.4: Example of a pseudoconcept hierarchy over the CLEVR vocabulary.

- How is this complementary to gSCAN (Ruis et al., 2020), Tangrams (A. Ji et al., 2022)? Can we use those domains as well/instead?
  - gSCAN talks about “composition of references and another involving composition of attributes”, since a small red square can have referants “square”, “red square”, “small red square”, et c.
- None of the compositional generalisation benchmarks seem to cover hierarchical compositions of concepts, this is a main point of novelty
- Important to have uncertainty in concepts, e.g. by defining conditional properties such as  $wug = \text{sphere AND (red OR green)}$

In gSCAN, [..] *each target referent from the instruction determiner phrase is ensured to be unique (only one possible target in “walk to the yellow square”)*.. Our benchmark is complementary to this notion in the sense that a model must apply actions over possibly multiple targets.

### Tasks and experiments

- Did the model learn category abstractions?

- What is the effect of curriculum learning (X. Wang et al., 2022) by acquiring pseudoconcepts from the bottom up (i.e. 1-gram pseudoconcepts) rather than sampled randomly from hierarchy?
  - Do we learn faster/with less?
  - Is the resulting internal structure of the model different? (E.g. investigate using probing experiments of Lovering and Pavlick (2022))
- “Maximize Presupposition”-like setup using, e.g., an image with only one large object: “What color is the large cube?” vs. “What color is the large object?” (Schlenker, 2012)
- Different levels of comprehension - Wug Test (Berko, 1958), Chromium test (Carey & Bartlett, 1978), CLEVR-Math (Lindström & Abraham, 2022a)
  - Detecting the presence of an object
  - Point out concept in line up
  - Use it in a task, e.g. mathematical reasoning or regular CLEVR (Johnson et al., 2017)
- Investigate the effects on comp.gen. when, e.g., gray scaling the image or dimming the lighting, and other visual changes
- Important to do splits across multiple dimensions to test compositionality
  - Recursive depth, do we learn a mechanism/algorithm, or just pattern matching?
  - Properties, shape color et c., do we strongly associate colors and shapes, or do we disentangle them?
- How do we construct a rich enough hierarchy?

The hierarchical constructions are on the following forms

- No parent – base/given knowledge, in this case shape/color et c.
- Naming of an unknown concept, i.e. analogous to the Chromium experiments
- Renaming of a color, e.g. blargh as a synonym for red
- Creating a parent for a 2-gram, e.g. a blue cylinder is called a blik
- Creating a parent for pseudoconcepts, e.g. two small blik are called a fnik





- Word groups - *group the objects based on color*
- Noun definition - *define the properties of the yellow sphere*
- Noun position - *What is to the left of the cube?*
- Abstract noun - *What is the relationship between A and B?*
- Noun subject - *What is the central object in this picture?*
- Intergroup flexibility - *Understand/manipulate categories/groups*

Hypothesis is that your performance on (easier) subtasks must be greater than on the composite task, otherwise we cannot say for certain that the particular concept is understood. We combine the ideas of the Chromium task and Eustace, introducing two pseudo-word concepts mapped to attribute n-grams.

- Is there a *blargh* in the image?
- What color is the *blargh*?
- What size is the *blargh*?
- What shape is the *blargh*?
- Remove all *blargh*. How many objects are left?
- What shape is the object left of the *blargh*?
- Is the *blargh* next to a cube?
- What is the common feature of the objects on the left side of the image?

We can translate this into a curriculum of tasks

- Is there a sphere in the image?
- Is there blue object in the image?
- Is there a blue sphere in the image?
- Is there a blobb (blue sphere) in the image?
- What color is the sphere?
- What color is the large object?
- What size is the sphere?
- How many spheres are there?
- How many blue objects are there?
- Remove all spheres. How many objects are left?

- What shape is the object left of the sphere?
- Is the sphere next to a cube?
- What is the common feature of the objects on the left side of the image?

We perform the experiment with both *blargh* and its components *small blue sphere* to assess the baseline behaviour that can be achieved.

The curriculum learning attempt reinitialises the trainer, resetting the learning rate. The impact of cyclical learning rates has been explored before, where Smith (2017) shows how it can improve both classification accuracy and reduce the amount of iterations necessary.

Early stopping when training on all data terminates the training processes too early, and the generalisation to 3hop is abysmal. However, early stopping in combination with curriculum learning can improve the convergence time and leads to a low loss over all subtasks.

One explanation to this behaviour could be that training on all tasks simultaneously leads to gradient starvation (Pezeshki et al., 2021).



# Chapter 8

## Conclusions

Example of how to use quotes at  
the beginning of chapters

---

*dali*

Summarise challenges, opportunities, how current work address those, and future work.



# Bibliography

- Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., & Goldberg, Y. (2016). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *CoRR*, *abs/1608.04207*.
- Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., & Goldberg, Y. (2017). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *5th International Conference on Learning Representations, ICLR*.
- Aissa, W., Ferecatu, M., & Crucianu, M. (2023). Curriculum learning for compositional visual reasoning. *ArXiv*, *abs/2303.15006*.
- Amizadeh, S., Palangi, H., Polozov, A., Huang, Y., & Koishida, K. (2020, July). Neuro-symbolic visual reasoning: Disentangling "Visual" from "Reasoning". In H. D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (pp. 279–290, Vol. 119). PMLR. <https://proceedings.mlr.press/v119/amizadeh20a.html>
- Andreas, J., Baroni, M., Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A., Devlin, J., Fyshe, A., Wehbe, L., et al. (2019). Measuring compositionality in representation learning. *International Conference on Learning Representations*, 375, 2227–2237.
- Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., & Parikh, D. (2015). VQA: Visual question answering. *Proceedings of the IEEE international conference on computer vision*, 2425–2433.
- Arras, L., Osman, A., & Samek, W. (2022). Clevr-xai: A benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81, 14–40.
- Askarian, N., Abbasnejad, E., Zukerman, I., Buntine, W., & Haffari, G. (2021). Curriculum learning effectively improves low data vqa. *Proceedings of the The 19th Annual Workshop of the Australasian Language Technology Association*, 22–33.
- Athalye, A., Engstrom, L., Ilyas, A., & Kwok, K. (2018). Synthesizing robust adversarial examples. *International conference on machine learning*, 284–293.
- Bader, S., & Hitzler, P. (2005). Dimensions of neural-symbolic integration – A structured survey. In S. N. Artëmov, H. Barringer, A. S. d’Avila Garcez, L. C. Lamb, & J. Woods (Eds.), *We will show them! essays in honour of dov gabbay, volume one* (pp. 167–194). College Publications.

- Baltrušaitis, T., Ahuja, C., & Morency, L.-P. (2019). Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.*, *41*(2), 423–443.
- Baltrušaitis, T., Ahuja, C., & Morency, L.-P. (2019). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(2), 423–443. <https://doi.org/10.1109/TPAMI.2018.2798607>
- Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., & Schneider, N. (2013). Abstract meaning representation for sembanking. *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, 178–186.
- Basu, K., Murugesan, K., Atzeni, M., Kapanipathi, P., Talamadupula, K., Klinger, T., Campbell, M., Sachan, M., & Gupta, G. (2021). A hybrid neuro-symbolic approach for text-based games using inductive logic programming. *Combining Learning and Reasoning: Programming Languages, Formalisms, and Representations*.
- Beinborn, L., Botschen, T., & Gurevych, I. (2018). Multimodal grounding for language processing. In E. M. Bender, L. Derczynski, & P. Isabelle (Eds.), *Proceedings of the 27th international conference on computational linguistics, COLING 2018* (pp. 2325–2339). Association for Computational Linguistics.
- Belinkov, Y. (2022). Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, *48*(1), 207–219.
- Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021a). On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623.
- Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021b). On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623. <https://doi.org/10.1145/3442188.3445922>
- Bender, E. M., & Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5185–5198. <https://doi.org/10.18653/v1/2020.acl-main.463>
- Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. *Proceedings of the 26th Annual International Conference on Machine Learning*, 41–48. <https://doi.org/10.1145/1553374.1553380>
- Ben-Younes, H., Cadene, R., Cord, M., & Thome, N. (2017). Mutan: Multimodal tucker fusion for visual question answering. *Proceedings of the IEEE international conference on computer vision*, 2612–2620.
- Berko, J. (1958). The child’s learning of english morphology. *Word*, *14*(2-3), 150–177.
- Berrendorf, M., Faerman, E., Vermue, L., & Tresp, V. (2020). Interpretable and fair comparison of link prediction or entity alignment methods



- with adjusted mean rank. *CoRR*, *abs/2002.06914*. <https://arxiv.org/abs/2002.06914>
- Bisk, Y., Holtzman, A., Thomason, J., Andreas, J., Bengio, Y., Chai, J., Lapata, M., Lazaridou, A., May, J., Nisnevich, A., Pinto, N., & Turian, J. (2020). Experience grounds language. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8718–8735. <https://doi.org/10.18653/v1/2020.emnlp-main.703>
- Björklund, J., Dahlgren Lindström, A., & Drewes, F. (2021). Bridging perception, memory, and inference through semantic relations. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 9136–9142. <https://aclanthology.org/2021.emnlp-main.719>
- Björklund, J., Lindström, A. D., & Drewes, F. (2022). An algebraic approach to learning and grounding. *LearnAut 2022*.
- Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250.
- Bolukbasi, T., Chang, K., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems 29: Annual conference on neural information processing systems, NIPS 2016* (pp. 4349–4357).
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Bouraoui, Z., Camacho-Collados, J., & Schockaert, S. (2020). Inducing relational knowledge from BERT. *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 7456–7463. <https://aaai.org/ojs/index.php/AAAI/article/view/6242>
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., . . . Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems 33: Annual conference on neural information processing systems*

- 2020, *neurips 2020, december 6-12, 2020, virtual*. <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html>
- Brunet, M., Alkalay-Houlihan, C., Anderson, A., & Zemel, R. S. (2019). Understanding the origins of bias in word embeddings. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning (ICML 2019)* (pp. 803–811, Vol. 97).
- Bruni, E., Tran, N.-K., & Baroni, M. (2014). Multimodal distributional semantics. *Journal of artificial intelligence research*, 49, 1–47.
- Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora necessarily contain human biases. *Science*, 356(6334), 183–186.
- Cao, K., Brbic, M., & Leskovec, J. (2021). Concept learners for few-shot learning. *International Conference on Learning Representation (ICLR)*.
- Carey, S., & Bartlett, E. (1978). Acquiring a single new word.
- Casper, S., Rauker, T., Ho, A., & Hadfield-Menell, D. (2023). Sok: Toward transparent AI: A survey on interpreting the inner structures of deep neural networks. *First IEEE Conference on Secure and Trustworthy Machine Learning*. <https://openreview.net/forum?id=8C5zt-0Utdn>
- Chaabouni, R., Kharitonov, E., Bouchacourt, D., Dupoux, E., & Baroni, M. (2020). Compositionality and generalization in emergent languages. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4427–4442.
- Chaplot, D. S., Sathyendra, K. M., Pasumarthi, R. K., Rajagopal, D., & Salakhutdinov, R. (2018). Gated-attention architectures for task-oriented language grounding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32.
- Chen, X., Liang, C., Yu, A. W., Song, D., & Zhou, D. (2020). Compositional generalization via neural-symbolic stack machines. *arXiv preprint arXiv:2008.06662*.
- Chomsky, N. (1975). *The logical structure of linguistic theory*. Springer.
- Chomsky, N. (2014). *Aspects of the theory of syntax* (Vol. 11). MIT press.
- Chomsky, N. (1957). Syntactic structures.
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2022). Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Conneau, A., Kruszewski, G., Lample, G., Barrault, L., & Baroni, M. (2018). What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In I. Gurevych & Y. Miyao (Eds.), *Proceedings of the 56th annual meeting of the association for computational linguistics, ACL 2018, volume 1: Long papers* (pp. 2126–2136).
- Dahlgren, A., Björklund, J., & Drewes, F. (2021). Perception, memory, and inference: The trinity of machine learning. <https://openreview.net/group?id=ijcai.org/IJCAI/2021/Workshop/NSNLI>
- Dahlgren Lindström, A., Björklund, J., Bensch, S., & Drewes, F. (2020). Probing multimodal embeddings for linguistic properties: The visual-semantic

- case. *Proceedings of the 28th International Conference on Computational Linguistics*, 730–744. <https://doi.org/10.18653/v1/2020.coling-main.64>
- Dantsin, E. (1992). Probabilistic logic programs and their semantics. In *Logic programming* (pp. 152–164). Springer.
- De Raedt, L., Kimmig, A., & Toivonen, H. (2007). Problog: A probabilistic prolog and its application in link discovery. *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2462–2467.
- Dechter, R. (1986). Learning while searching in constraint-satisfaction problems.
- Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., & Kaiser, Ł. (2018). Universal transformers. *arXiv preprint arXiv:1807.03819*.
- Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2019, volume 1: Long and short papers* (pp. 4171–4186).
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- Dillon, S. (2020). The eliza effect and its dangers: From demystification to gender critique. *Journal for Cultural Research*, 24(1), 1–15.
- Dupoux, E. (2018). Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173, 43–59.
- Dwivedi, V. P., & Bresson, X. (2020). A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*.
- Elliott, D., & Kádár, Á. (2017). Imagination improves multimodal translation. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 130–141.
- Erdem, E., Kuyu, M., Yagcioglu, S., Frank, A., Parcalabescu, L., Plank, B., Babii, A., Turuta, O., Erdem, A., Calixto, I., Lloret, E., Apostol, E.-S., Truica, C.-O., Sandrih, B., Gatt, A., Martincic-Ipsic, S., Berend, G., & Korvel, G. (2022). Neural natural language generation: A survey on multilinguality, multimodality, controllability and learning [to appear]. *Journal of Artificial Intelligence Research (JAIR)*.
- Eustace, B. W. (1969). Learning a complex concept at differing hierarchical levels. *Journal of Educational Psychology*, 60(6p1), 449.
- Evans, J. (1996). Rationality and reasoning. *Cognitive Psychology*.

- Evans, J. S. B. T., & Stanovich, K. E. (2013). Dual-process theories of higher cognition: Advancing the debate [PMID: 26172965]. *Perspectives on Psychological Science*, 8(3), 223–241. <https://doi.org/10.1177/1745691612460685>
- Faghri, F., Fleet, D. J., Kiros, J. R., & Fidler, S. (2018). VSE++: improving visual-semantic embeddings with hard negatives. *British Machine Vision Conference 2018, BMVC 2018*, 12.
- Felix, R., Kumar, V. B. G., Reid, I., & Carneiro, G. (2018). Multi-modal cycle-consistent generalized zero-shot learning. In *Proceedings of the European Conference on Computer Vision*.
- Fodor, J. D., Fodor, J. A., & Garrett, M. F. (2013). 12. the psychological unreality of semantic representations. In *Volume ii readings in philosophy of psychology, volume ii* (pp. 238–252). Harvard University Press.
- Fodor, J. A. (1998). *Concepts: Where cognitive science went wrong*. Oxford University Press.
- Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2), 3–71.
- Foucault, M. (1991). *The foucault effect: Studies in governmentality*. University of Chicago Press.
- Frank, S., Bugliarello, E., & Elliott, D. (2021). Vision-and-Language or Vision-for-Language? On Cross-Modal Influence in Multimodal Transformers [arXiv: 2109.04448 version: 1]. *arXiv:2109.04448 [cs]*. Retrieved November 1, 2021, from <http://arxiv.org/abs/2109.04448>
- Frege, G. (1963). Compound thoughts. *Mind*, 72(285), 1–17.
- Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., & Mikolov, T. (2013). DeViSE: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 26: 27th annual conference on neural information processing systems, NIPS 2013* (pp. 2121–2129).
- Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., & Rohrbach, M. (2016a). Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*.
- Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., & Rohrbach, M. (2016b). Multimodal compact bilinear pooling for visual question answering and visual grounding. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 457–468.
- Gadamer, H. (1975). Hermeneutics and social science, cultural hermeneutics 2: 307–316.
- Garcez, A., Gori, M., Lamb, L., Serafini, L., Spranger, M., & Tran, S. (2019). Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *Journal of Applied Logics*, 6(4), 611–631.
- Garcez, A. d., Bader, S., Bowman, H., Lamb, L. C., de Penning, L., Illuminoo, B., Poon, H., & Gerson Zaverucha, C. (2022). Neural-symbolic learning

- and reasoning: A survey and interpretation. *Neuro-Symbolic Artificial Intelligence: The State of the Art*, 342, 1.
- Garcez, A. d., Besold, T. R., De Raedt, L., Földiak, P., Hitzler, P., Icard, T., Kühnberger, K.-U., Lamb, L. C., Miikkulainen, R., & Silver, D. L. (2015). Neural-symbolic learning and reasoning: Contributions and challenges. *2015 AAAI Spring Symposium Series*.
- Garcez, A. d., & Lamb, L. C. (2023). Neurosymbolic ai: The 3rd wave. *Artificial Intelligence Review*. <https://doi.org/10.1007/s10462-023-10448-w>
- Garcez, A. S. d., Broda, K., Gabbay, D. M., et al. (2002). *Neural-symbolic learning systems: Foundations and applications*. Springer Science & Business Media.
- Garcez, A. S. d., Broda, K. B., & Gabbay, D. M. (2012). *Neural-symbolic learning systems: Foundations and applications*. Springer Science & Business Media.
- Gladkova, A., Drozd, A., & Matsuoka, S. (2016). Analogy-based detection of morphological and semantic relations with word embeddings: What works and what doesn't. *Proceedings of the NAACL Student Research Workshop*, 8–15. <https://doi.org/10.18653/v1/N16-2002>
- Golinkoff, R. M., Hirsh-Pasek, K., Bailey, L. M., & Wenger, N. R. (1992). Young children and adults use lexical principles to learn new nouns. *Developmental psychology*, 28(1), 99.
- Gottlob, F., & Austin, J. (1884). The foundations of arithmetic.
- Hamilton, K., Nayak, A., Božić, B., & Longo, L. (2022). Is neuro-symbolic ai meeting its promises in natural language processing? a structured review. *Semantic Web*, (Preprint), 1–42.
- Hammer, B., & Hitzler, P. (2007). *Perspectives of neural-symbolic integration* (Vol. 77). Springer.
- Harlow, H. F. (1949). The formation of learning sets. *Psychological review*, 56(1), 51.
- Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3), 335–346.
- Harnad, S. (1993). Problems, problems: The frame problem as a symptom of the symbol grounding problem. *Psychology*, 4(34).
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. [corr abs/1703.06870](https://arxiv.org/abs/1703.06870). *arXiv preprint arXiv:1703.06870*.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
- Henderson, J. (2020). The unstoppable rise of computational linguistics in deep learning. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6294–6306.
- Henderson, L. (2020). The Problem of Induction. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Spring 2020). Metaphysics Research Lab, Stanford University.

- Hewitt, J., & Liang, P. (2019). Designing and interpreting probes with control tasks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2733–2743.
- Heyn, E. T. (1904). Berlin’s wonderful horse; he can do almost everything but talk – how he was taught [Accessed: 2023-09-27]. *The New York Times*. <https://www.nytimes.com/1904/09/04/archives/berlins-wonderful-horse-he-can-do-almost-everything-but-talk-how.html>
- Hinton, G. E., et al. (1986). Learning distributed representations of concepts. *Proceedings of the eighth annual conference of the cognitive science society*, 1, 12.
- Hitzler, P., Eberhart, A., Ebrahimi, M., Sarker, M. K., & Zhou, L. (2022). Neuro-symbolic approaches in artificial intelligence [nwac035]. *National Science Review*, 9(6). <https://doi.org/10.1093/nsr/nwac035>
- Hofstadter, D. R. (1995). *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought*. Basic books.
- Hohenecker, P., & Lukasiewicz, T. (2020). Ontology reasoning with deep neural networks. *Journal of Artificial Intelligence Research (JAIR)*, 68, 503–540. <https://doi.org/10.1613/jair.1.11661>
- Holzinger, A., Kickmeier-Rust, M., & Müller, H. (2019). Kandinsky patterns as iq-test for machine learning. *International cross-domain conference for machine learning and knowledge extraction*, 1–14.
- Hong, Y., Lin, C., Du, Y., Chen, Z., Tenenbaum, J. B., & Gan, C. (2023). 3d concept learning and reasoning from multi-view images. *arXiv preprint arXiv:2303.11327*.
- Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2022). Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9), 5149–5169. <https://doi.org/10.1109/TPAMI.2021.3079209>
- Huang, D., Shi, S., Lin, C.-Y., Yin, J., & Ma, W.-Y. (2016). How well do computers solve math word problems? large-scale dataset construction and evaluation. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 887–896.
- Hudson, D. A., & Manning, C. D. (2019). Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6700–6709.
- Hume, D. (1739). *A treatise of human nature*. Oxford University Press.
- Hupkes, D., Dankers, V., Mul, M., & Bruni, E. (2020a). Compositionality decomposed: How do neural networks generalise? *J. Artif. Intell. Res.*, 67, 757–795.
- Hupkes, D., Dankers, V., Mul, M., & Bruni, E. (2020b). Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67, 757–795.

- Ito, T., Klinger, T., Schultz, D. H., Murray, J. D., Cole, M. W., & Rigotti, M. (2022). Compositional generalization through abstract representations in human and artificial neural networks. *arXiv preprint arXiv:2209.07431*.
- Janssen, T. M. (1986). Foundations and applications of montague grammar: Philosophy, framework, computer science.
- Ji, A., Kojima, N., Rush, N., Suhr, A., Vong, W. K., Hawkins, R. D., & Artzi, Y. (2022). Abstract visual reasoning with tangram shapes. *arXiv preprint arXiv:2211.16492*.
- Ji, J., Krishna, R., Fei-Fei, L., & Niebles, J. C. (2020). Action genome: Actions as compositions of spatio-temporal scene graphs. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10236–10247.
- Jiang, Z., Xu, F. F., Araki, J., & Neubig, G. (2020). How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8, 423–438. [https://doi.org/10.1162/tacl\\_a\\_00324](https://doi.org/10.1162/tacl_a_00324)
- Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., & Girshick, R. (2017). Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2901–2910.
- Kadlec, R., Bajgar, O., & Kleindienst, J. (2017). Knowledge base completion: Baselines strike back. *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 69–74. <https://doi.org/10.18653/v1/W17-2609>
- Kahneman, D. (2011). *Thinking, fast and slow*. Macmillan.
- Kakas, A. C., & Michael, L. (2020). Abduction and argumentation for explainable machine learning: A position survey. *CoRR*, *abs/2010.12896*. <https://arxiv.org/abs/2010.12896>
- Kamp, H., & Partee, B. (1995). Prototype theory and compositionality. *Cognition*, 57(2), 129–191.
- Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3128–3137.
- Kautz, H. (2022). The third ai summer: Aaai robert s. engelmore memorial lecture. *AI Magazine*, 43(1), 105–125.
- Kenton, J. D. M.-W. C., & Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 4171–4186.
- Keysers, D., Schärli, N., Scales, N., Buisman, H., Furrer, D., Kashubin, S., Momchev, N., Sinopalnikov, D., Stafiniak, L., Tihon, T., Tsarkov, D., Wang, X., van Zee, M., & Bousquet, O. (2020). Measuring compositional generalization: A comprehensive method on realistic data. *International Conference on Learning Representations*. <https://openreview.net/forum?id=SygcChNKwr>

- Kiela, D., Conneau, A., Jabri, A., & Nickel, M. (2018). Learning visually grounded sentence representations. *Proceedings of NAACL-HLT*, 408–418.
- Kim, B., Kim, H., Kim, K., Kim, S., & Kim, J. (2019). Learning not to learn: Training deep neural networks with biased data. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9012–9020.
- Kim, N., & Linzen, T. (2020). Cogs: A compositional generalization challenge based on semantic interpretation. *arXiv preprint arXiv:2010.05465*.
- Kim, W., Son, B., & Kim, I. (2021). Vilt: Vision-and-language transformer without convolution or region supervision. *International Conference on Machine Learning*, 5583–5594.
- Klinger, T., Adjodah, D., Marois, V., Joseph, J., Riemer, M., Pentland, A., & Campbell, M. (2020). A study of compositional generalization in neural models. *arXiv preprint arXiv:2006.09437*.
- Kobayashi, G., Kuribayashi, T., Yokoi, S., & Inui, K. (2020). Attention is not only a weight: Analyzing transformers with vector norms. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7057–7075. <https://doi.org/10.18653/v1/2020.emnlp-main.574>
- Kocijan, V., Lukaszewicz, T., Davis, E., Marcus, G., & Morgenstern, L. (2020). A review of winograd schema challenge datasets and approaches. *arXiv preprint arXiv:2004.13831*.
- Koncel-Kedziorski, R., Roy, S., Amini, A., Kushman, N., & Hajishirzi, H. (2016). Mawps: A math word problem repository. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1152–1157.
- Kottur, S., Moura, J. M., Parikh, D., Batra, D., & Rohrbach, M. (2019). Clevr-dialog: A diagnostic dataset for multi-round reasoning in visual dialog. *arXiv preprint arXiv:1903.03166*.
- Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1), 32–73.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90.
- Kucera, H., & Francis, W. N. (1967). *Computational analysis of present-day american english*. Brown University Press.
- Kudo, K., Aoki, Y., Kuribayashi, T., Brassard, A., Yoshikawa, M., Sakaguchi, K., & Inui, K. (2023). Do deep neural networks capture compositionality in arithmetic reasoning? *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 1343–1354.



- Kuhn, T. S., & Hawkins, D. (1963). The structure of scientific revolutions. *American Journal of Physics*, 31, 554–555.
- Lake, B., & Baroni, M. (2018a). Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *International conference on machine learning*, 2873–2882.
- Lake, B., & Baroni, M. (2018b, July). Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning* (pp. 2873–2882, Vol. 80). PMLR. <https://proceedings.mlr.press/v80/lake18a.html>
- Lake, B., Salakhutdinov, R., Gross, J., & Tenenbaum, J. (2011). One shot learning of simple visual concepts. *Proceedings of the annual meeting of the cognitive science society*, 33(33).
- Lake, B. M. (2019). Compositional generalization through meta sequence-to-sequence learning. *arXiv preprint arXiv:1906.05381*.
- Lake, B. M., Linzen, T., & Baroni, M. (2019). Human few-shot learning of compositional instructions. *41st Annual Meeting of the Cognitive Science Society: Creativity+ Cognition+ Computation, CogSci 2019*, 611–617.
- Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266), 1332–1338.
- Lan, Y., Wang, L., Jiang, J., & Lim, E.-P. (2022). Improving compositional generalization in math word problem solving. *arXiv preprint arXiv:2209.01352*.
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. <https://openreview.net/forum?id=H1eA7AetvS>
- Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. *Proceedings of the 5th Annual International Conference on Systems Documentation*, 24–26.
- Levesque, H. J., Davis, E., & Morgenstern, L. (2012). The winograd schema challenge. *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, 552–561.
- Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., & Chang, K.-W. (2020). What does BERT with vision look at? *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5265–5275. <https://doi.org/10.18653/v1/2020.acl-main.469>
- Li, Z., Wang, X., Stengel-Eskin, E., Kortylewski, A., Ma, W., Van Durme, B., & Yuille, A. (2022). Super-clevr: A virtual benchmark to diagnose domain robustness in visual reasoning. *arXiv preprint arXiv:2212.00259*.
- Liang, P. P., Zadeh, A., & Morency, L.-P. (2022). Foundations and recent trends in multimodal machine learning: Principles, challenges, and open questions. *arXiv preprint arXiv:2209.03430*.

- Lin, B., Bouneffouf, D., & Rish, I. (2023). A survey on compositional generalization in applications. *arXiv preprint arXiv:2302.01067*.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *European conference on computer vision*, 740–755.
- Lin, T., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In D. J. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Proceedings of the 13th european conference on computer vision, ECCV 2014, part V* (pp. 740–755, Vol. 8693).
- Lindström, A. D., & Abraham, S. S. (2022a). Clevr-math: A dataset for compositional language, visual and mathematical reasoning. In A. S. d’Avila Garcez & E. Jiménez-Ruiz (Eds.), *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* (pp. 155–170, Vol. 3212). CEUR-WS.org. <https://ceur-ws.org/Vol-3212/paper11.pdf>
- Lindström, A. D., & Abraham, S. S. (2022b). Clevr-math: A dataset for compositional language, visual and mathematical reasoning. In A. S. d’Avila Garcez & E. Jiménez-Ruiz (Eds.), *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* (pp. 155–170, Vol. 3212). CEUR-WS.org. <https://ceur-ws.org/Vol-3212/paper11.pdf>
- Ling, W., Yogatama, D., Dyer, C., & Blunsom, P. (2017). Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.
- Linzen, T. (2020). How can we accelerate progress towards human-like linguistic generalization? *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5210–5217.
- Lit, L., Schweitzer, J. B., & Oberbauer, A. M. (2011). Handler beliefs affect scent detection dog outcomes. *Animal cognition*, 14, 387–394.
- Liu, F., Ye, R., Wang, X., & Li, S. (2020). Hal: Improved text-image matching by mitigating visual semantic hubs. *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*.
- Liu, R., Liu, C., Bai, Y., & Yuille, A. L. (2019a). Clevr-ref+: Diagnosing visual reasoning with referring expressions. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4185–4194.
- Liu, R., Liu, C., Bai, Y., & Yuille, A. L. (2019b). Clevr-ref+: Diagnosing visual reasoning with referring expressions. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Liu, S., & Deng, W. (2015). Very deep convolutional neural network based image classification using small training sample size. *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 730–734.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.
- Liu, Z., Gan, E., & Tegmark, M. (2023). Seeing is believing: Brain-inspired modular training for mechanistic interpretability.
- Loula, J., Baroni, M., & Lake, B. M. (2018). Rearranging the familiar: Testing compositional generalization in recurrent networks. *arXiv preprint arXiv:1807.07545*.
- Lovering, C., & Pavlick, E. (2022). Unit testing for concepts in neural networks. *arXiv preprint arXiv:2208.10244*.
- Lu, J., Batra, D., Parikh, D., & Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.
- Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.-W., Zhu, S.-C., Tafjord, O., Clark, P., & Kalyan, A. (2022). Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35, 2507–2521.
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Proceedings of the 31st international conference on neural information processing systems*, 4768–4777.
- Luong, M.-T., Kayser, M., & Manning, C. D. (2015). Deep neural language models for machine translation. *19th CoNLL*, 305–309.
- Mahowald, K., Ivanova, A. A., Blank, I. A., Kanwisher, N., Tenenbaum, J. B., & Fedorenko, E. (2023). Dissociating language and thought in large language models: A cognitive perspective. *arXiv preprint arXiv:2301.06627*.
- Maker, M. H. (2006). Ai@ 50: Ai past, present, future. *Dartmouth College*. [http://www.engagingexperience.com/2006/07/ai50\\_ai\\_past\\_pr.html](http://www.engagingexperience.com/2006/07/ai50_ai_past_pr.html).
- Malinowski, M., & Fritz, M. (2014). A multi-world approach to question answering about real-world scenes based on uncertain input. *Advances in neural information processing systems*, 27.
- Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., & De Raedt, L. (2018). Deepproblog: Neural probabilistic logic programming. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 31).
- Manhaeve, R., Dumančić, S., Kimmig, A., Demeester, T., & De Raedt, L. (2018). DeepProbLog: Neural Probabilistic Logic Programming [arXiv:1805.10872]. *arXiv:1805.10872 [cs]*. Retrieved October 29, 2021, from <http://arxiv.org/abs/1805.10872>
- Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., & Wu, J. (2019a). The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*.

- Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., & Wu, J. (2019b). The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. *International Conference on Learning Representations*. <https://openreview.net/forum?id=rJgMlhRctm>
- Marcus, G., & Davis, E. (2019). *Rebooting ai: Building artificial intelligence we can trust*. Vintage.
- Marcus, G. F. (2003). *The algebraic mind: Integrating connectionism and cognitive science*. MIT press.
- McCarthy, J. (1960). Programs with common sense.
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 115–133.
- Mercier, H., & Sperber, D. (2009). Intuitive and reflective inferences.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 26: 27th annual conference on neural information processing systems, NIPS 2013* (pp. 3111–3119).
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 26: 27th annual conference on neural information processing systems 2013. proceedings of a meeting held december 5-8, 2013, lake tahoe, nevada, united states* (pp. 3111–3119). <https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html>
- Miller, G. A. (1992). WordNet: A lexical database for English. *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*. <https://aclanthology.org/H92-1116>
- Mitra, A., & Baral, C. (2016). Learning to use formulas to solve simple arithmetic problems. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2144–2153.
- Mollo, D. C., & Millière, R. (2023). The vector grounding problem.
- Montague, R. (1970). Pragmatics and intensional logic. *Synthese*, 22(1-2), 68–94.
- Narasimhan, M., Lazebnik, S., & Schwing, A. (2018). Out of the box: Reasoning with graph convolution nets for factual visual question answering. *Advances in neural information processing systems*, 31.
- Navigli, R., Blloshmi, R., & Lorenzo, A. C. M. (2022). Babelnet meaning representation: A fully semantic formalism to overcome language barriers.
- Ng, R., & Subrahmanian, V. S. (1992). Probabilistic logic programming. *Information and computation*, 101(2), 150–201.

- Orseau, L., & Armstrong, S. (2016). Safely interruptible agents. In A. T. Ihler & D. Janzing (Eds.), *Proceedings of the 32nd conference on uncertainty in artificial intelligence, UAI 2016* (pp. 557–566).
- Pagin, P. (2003). Communication and strong compositionality. *Journal of Philosophical Logic*, 32, 287–322.
- Parcalabescu, L., Trost, N., & Frank, A. (2021). What is multimodality? *Proceedings of the First Workshop on Multimodal Semantic Representations (MMSR)*. <https://iwcs2021.github.io/proceedings/mmsr/pdf/2021.mmsr-1.1.pdf>
- Parcalabescu, L., Cafagna, M., Muradjan, L., Frank, A., Calixto, I., & Gatt, A. (2022). Valse: A task-independent benchmark for vision and language models centered on linguistic phenomena [to appear]. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. <https://arxiv.org/abs/2112.07566>
- Partee, B., et al. (1984). Compositionality. *Varieties of formal semantics*, 3, 281–311.
- Patel, A., Bhattamishra, S., & Goyal, N. (2021). Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.
- Patel, R., & Pavlick, E. (2021). Mapping language models to grounded conceptual spaces. *International Conference on Learning Representations*.
- Pavlick, E. (2022). Semantic structure in deep learning. *Annual Review of Linguistics*, 8(1), 447–471. <https://doi.org/10.1146/annurev-linguistics-031120-122924>
- Pearl, J. (2019). The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3), 54–60.
- Pelletier, F. J. (1994). The principle of semantic compositionality. *Topoi*, 13(1), 11–24.
- Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. <https://doi.org/10.3115/v1/D14-1162>
- Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., & Miller, A. (2019a). Language models as knowledge bases? *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2463–2473.
- Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., & Miller, A. (2019b). Language models as knowledge bases? *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2463–2473. <https://doi.org/10.18653/v1/D19-1250>
- Pezeshki, M., Kaba, O., Bengio, Y., Courville, A. C., Precup, D., & Lajoie, G. (2021). Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34, 1256–1272.

- Pezeshkpour, P., Tian, Y., & Singh, S. (2020). Revisiting evaluation of knowledge base completion models. *Automated Knowledge Base Construction*.
- Pfungst, O. (1911). *Clever hans:(the horse of mr. von osten.) a contribution to experimental animal and human psychology*. Holt, Rinehart; Winston.
- Poerner, N., Waltinger, U., & Schütze, H. (2019). BERT is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised QA.
- Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*, 101–108.
- Qiu, L., Hu, H., Zhang, B., Shaw, P., & Sha, F. (2021). Systematic generalization on gscan: What is nearly solved and what is next? *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2180–2188.
- Qiu, L., Shaw, P., Pasupat, P., Nowak, P., Linzen, T., Sha, F., & Toutanova, K. (2022). Improving compositional generalization with latent structure and data augmentation. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4341–4362. <https://doi.org/10.18653/v1/2022.naacl-main.323>
- Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. A., & Botvinick, M. (2018). Machine theory of mind. *International conference on machine learning*, 4218–4227.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021a). Learning transferable visual models from natural language supervision. *International conference on machine learning*, 8748–8763.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021b). Learning Transferable Visual Models From Natural Language Supervision [arXiv: 2103.00020]. *arXiv:2103.00020 [cs]*. Retrieved October 29, 2021, from <http://arxiv.org/abs/2103.00020>
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019a). *Language models are unsupervised multitask learners* (tech. rep.). OpenAI.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019b). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.
- Raedt, L. d., Dumančić, S., Manhaeve, R., & Marra, G. (2020, July). From statistical relational to neuro-symbolic artificial intelligence [Survey track]. In C. Bessiere (Ed.), *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI-20* (pp. 4943–

- 4950). International Joint Conferences on Artificial Intelligence Organization. <https://doi.org/10.24963/ijcai.2020/688>
- Raedt, L. D., & Kersting, K. (2008). Probabilistic inductive logic programming. In *Probabilistic inductive logic programming* (pp. 1–27). Springer.
- Rehurek, R., & Sojka, P. (2011). Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 1135–1144.
- Riegel, R., Gray, A., Luus, F., Khan, N., Makondo, N., Akhalwaya, I. Y., Qian, H., Fagin, R., Barahona, F., Sharma, U., et al. (2020). Logical neural networks. *arXiv preprint arXiv:2006.13155*.
- Rieger, L., Singh, C., Murdoch, W., & Yu, B. (2020, July). Interpretations are useful: Penalizing explanations to align neural networks with prior knowledge. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (pp. 8116–8126, Vol. 119). PMLR. <https://proceedings.mlr.press/v119/rieger20a.html>
- Robaidek, B., Koncel-Kedziorski, R., & Hajishirzi, H. (2018). Data-driven methods for solving algebra word problems. *arXiv preprint arXiv:1804.10718*.
- Rogers, A., Hosur Ananthakrishna, S., & Rumshisky, A. (2018). What’s in your embedding, and how it predicts task performance. *Proceedings of the 27th International Conference on Computational Linguistics*, 2690–2703.
- Rogers, C. R., & Carmichael, L. (1942). Counseling and psychotherapy: Newer concepts in practice.
- Rosenbloom, P. S. (2010). Combining procedural and declarative knowledge in a graphical architecture. *Proceedings of the 10th International Conference on Cognitive Modeling*, 205–210.
- Ross, C. C. (2022). *Learning language with multimodal models* [Doctoral dissertation, Massachusetts Institute of Technology].
- Rossi, A., & Matinata, A. (2020). Knowledge graph embeddings: Are relation-learning models learning relations? *EDBT/ICDT Workshops*.
- Ruis, L., Andreas, J., Baroni, M., Bouchacourt, D., & Lake, B. M. (2020). A benchmark for systematic generalization in grounded language understanding. *Advances in neural information processing systems*, 33, 19861–19872.
- Salewski, L., Koepke, A. S., Lensch, H. P., & Akata, Z. (2022). Clevr-x: A visual reasoning dataset for natural language explanations. *xxAI-Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*, 69–88.

- Sampat, S. K., Kumar, A., Yang, Y., & Baral, C. (2021). CLEVR\_HYP: A challenge dataset and baselines for visual question answering with hypothetical actions over images. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3692–3709. <https://www.aclweb.org/anthology/2021.naacl-main.289>
- Saqr, R., & Narasimhan, K. (2020). Multimodal graph networks for compositional generalization in visual question answering. *Advances in Neural Information Processing Systems*.
- Sarker, M. K., Zhou, L., Eberhart, A., & Hitzler, P. (2021). Neuro-symbolic artificial intelligence. *AI Communications*, 34(3), 197–209.
- Saxton, D., Grefenstette, E., Hill, F., & Kohli, P. (n.d.). Analysing mathematical reasoning abilities of neural models. *International Conference on Learning Representations*.
- Saxton, D., Grefenstette, E., Hill, F., & Kohli, P. (2019). Analysing mathematical reasoning abilities of neural models. *International Conference on Learning Representations*. <https://openreview.net/forum?id=H1gR5iR5FX>
- Schlag, I., Smolensky, P., Fernandez, R., Jojic, N., Schmidhuber, J., & Gao, J. (2019). Enhancing the transformer with explicit relational encoding for math problem solving. *arXiv preprint arXiv:1910.06611*.
- Schlenker, P. (2012). Maximize presupposition and gricean reasoning. *Natural language semantics*, 20, 391–429.
- Schmidjell, T., Range, F., Huber, L., & Virányi, Z. (2012). Do owners have a clever hans effect on dogs? results of a pointing study. *Frontiers in psychology*, 3, 558.
- Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and brain sciences*, 3(3), 417–424.
- Searle, J. R. (1990). Is the brain’s mind a computer program? *Scientific American*, 262(1), 25–31.
- Sen, P., de Carvalho, B. W., Riegel, R., & Gray, A. (2022). Neuro-symbolic inductive logic programming with logical neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8), 8212–8219.
- Shaw, P., Chang, M.-W., Pasupat, P., & Toutanova, K. (2020). Compositional generalization and natural language variation: Can a semantic parsing approach handle both? *arXiv preprint arXiv:2010.12725*.
- Shaw, P., Chang, M.-W., Pasupat, P., & Toutanova, K. (2021). Compositional generalization and natural language variation: Can a semantic parsing approach handle both? *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)*, 922–938.
- Shekhar, R., Pezzelle, S., Klimovich, Y., Herbelot, A., Nabi, M., Sangineto, E., & Bernardi, R. (2017). FOIL it! Find one mismatch between image and language caption. *Proceedings of the 55th Annual Meeting of the Asso-*



- ciation for Computational Linguistics (ACL), Volume 1: Long Papers, 255–265.
- Shen, F., Zhou, X., Yu, J., Yang, Y., Liu, L., & Shen, H. T. (2019). Scalable zero-shot learning via binary visual-semantic embeddings. *IEEE Transactions on Image Processing*, 28(7), 3662–3674.
- Shi, H., Mao, J., Xiao, T., Jiang, Y., & Sun, J. (2018). Learning visually-grounded semantics from contrastive adversarial samples. In E. M. Bender, L. Derczynski, & P. Isabelle (Eds.), *Proceedings of the 27th international conference on computational linguistics, COLING 2018* (pp. 3715–3727). Association for Computational Linguistics.
- Shi, X., Padhi, I., & Knight, K. (2016). Does string-based neural MT learn source syntax? *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1526–1534.
- Shih, K. J., Singh, S., & Hoiem, D. (2016). Where to look: Focus regions for visual question answering. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4613–4621.
- Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. *International Conference on Machine Learning*, 3145–3153.
- Sikarwar, A., Patel, A., & Goyal, N. (2022). When can transformers ground and compose: Insights from compositional generalization benchmarks. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 648–669. <https://aclanthology.org/2022.emnlp-main.41>
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, 550. <http://dx.doi.org/10.1038/nature24270>
- Slack, D., Hilgard, S., Jia, E., Singh, S., & Lakkaraju, H. (2019). How can we fool lime and shap? adversarial attacks on post hoc explanation methods. *CoRR*, abs/1911.02508. <http://arxiv.org/abs/1911.02508>
- Sloman, S. A. (1996). The empirical case for two systems of reasoning. *Psychological bulletin*, 119(1), 3.
- Smith, L. N. (2017). Cyclical learning rates for training neural networks. *2017 IEEE winter conference on applications of computer vision (WACV)*, 464–472.
- Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and brain sciences*, 11(1), 1–23.
- Smolensky, P., McCoy, R. T., Fernandez, R., Goldrick, M., & Gao, J. (2022). Neurocompositional computing: From the central paradox of cognition to a new generation of ai systems. *arXiv preprint arXiv:2205.01128*.
- Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

- Socher, R., Ganjoo, M., Manning, C. D., & Ng, A. Y. (2013). Zero-shot learning through cross-modal transfer. *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, 935–943.
- Sorokin, D., & Gurevych, I. (2017). Context-aware representations for knowledge base relation extraction. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1784–1789. <https://doi.org/10.18653/v1/D17-1188>
- Stammer, W., Schramowski, P., & Kersting, K. (2021). Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3619–3629.
- Stanovich, K. E. (2009). Distinguishing the reflective, algorithmic, and autonomous minds: Is it time for a tri-process theory?
- Stanovich, K. E., & West, R. F. (2000). Individual differences in reasoning: Implications for the rationality debate? *Behavioral and brain sciences*, 23(5), 645–665.
- STEELS, L., VERHEYEN, L., & VAN TRIJP, R. (2022). An experiment in measuring understanding. *Proceedings of the First International Conference on Hybrid Human-Artificial Intelligence (HHAI)*. IOS Press, Amsterdam.
- Sun, R., & Bookman, L. A. (1994). Computational architectures integrating neural and symbolic processes: A perspective on the state of the art.
- Sundaram, S. S., & Abraham, S. S. (2018). Solving simple arithmetic word problems precisely with schemas. *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 542–547.
- Sundaram, S. S., Deepak, P., & Abraham, S. S. (2020). Distributed representations for arithmetic word problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05), 9000–9007.
- Sundaram, S. S., & Khemani, D. (2015). Natural language processing for solving simple word problems. *Proceedings of the 12th International Conference on Natural Language Processing*, 394–402.
- Szabó, Z. G. (2012). The case for compositionality.
- Szabó, Z. G. (2022). Compositionality. In E. N. Zalta & U. Nodelman (Eds.), *The Stanford encyclopedia of philosophy* (Fall 2022). Metaphysics Research Lab, Stanford University.
- Tan, H., & Bansal, M. (2019). Lxmert: Learning cross-modality encoder representations from transformers. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5100–5111.
- Tenney, I., Das, D., & Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4593–4601.

- Thrush, T., Jiang, R., Bartolo, M., Singh, A., Williams, A., Kiela, D., & Ross, C. (2022). Winoground: Probing vision and language models for visiolinguistic compositionality. *arXiv preprint arXiv:2204.03162*.
- todo. (2023). Todo reference.
- Tubella, A. A., Mollo, D. C., Lindström, A. D., Devinney, H., Dignum, V., Ericson, P., Jonsson, A., Kampik, T., Lenaerts, T., Mendez, J. A., & Nieves, J. C. (2023). Acropolis: A descriptive framework for making sense of fairness.
- Turing, A. M. (1950). I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, 59(236), 433–460. <https://doi.org/10.1093/mind/LIX.236.433>
- untersucht den Klugen Hans, O. P. (2006). Messung kontra augenschein. *Psychologische Rundschau: Überblick über die Fortschritte der Psychologie in Deutschland, Österreich und der Schweiz*, 57(2), 106–111.
- Uppal, S., Bhagat, S., Hazarika, D., Majumder, N., Poria, S., Zimmermann, R., & Zadeh, A. (2022). Multimodal research in vision and language: A review of current and emerging trends. *Information Fusion*, 77, 149–171.
- van Bekkum, M., de Boer, M., van Harmelen, F., Meyer-Vitali, A., & ten Teije, A. (2021). Modular design patterns for hybrid learning and reasoning systems: A taxonomy, patterns and use cases. <https://arxiv.org/abs/2102.11965>
- Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. *Advances in neural information processing systems*, 29.
- Von Humboldt, W. (1836). *Über die verschiedenheit des menschlichen sprachbaues und ihren einfluss auf die geistige entwicklung des menschengeschlechts*. Druckerei der königlichen Akademie der Wissenschaften.
- Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Transparent, explainable, and accountable AI for robotics. *Science Robotics*, 6(2).
- Wald, J., Dhama, H., Navab, N., & Tombari, F. (2020). Learning 3d semantic scene graphs from 3d indoor reconstructions. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3961–3970.
- Wang, P., Wu, Q., Shen, C., van den Hengel, A., & Dick, A. (2017). Fvqa: Fact-based visual question answering.
- Wang, P.-W., Donti, P., Wilder, B., & Kolter, Z. (2019, June). SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (pp. 6545–6554, Vol. 97). PMLR. <https://proceedings.mlr.press/v97/wang19e.html>
- Wang, X., Chen, Y., & Zhu, W. (2022). A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9), 4555–4576. <https://doi.org/10.1109/TPAMI.2021.3069908>
- Wang, Y., Ruffinelli, D., Gemulla, R., Broscheit, S., & Meilicke, C. (2019). On evaluating embedding models for knowledge base completion. *Pro-*

- ceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, 104–112. <https://doi.org/10.18653/v1/W19-4313>
- Weber, L., Minervini, P., Münchmeyer, J., Leser, U., & Rocktäschel, T. (2019). Nlprolog: Reasoning with weak unification for question answering in natural language. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6151–6161.
- Weiss, G., Goldberg, Y., & Yahav, E. (2021). Thinking like transformers. *International Conference on Machine Learning*, 11080–11090.
- Weißenhorn, P., Donatelli, L., & Koller, A. (2022). Compositional generalization with a broad-coverage semantic parser. *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, 44–54.
- Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36–45.
- What Will Transformers Transform? – Rodney Brooks. (n.d.). Retrieved March 24, 2023, from <https://rodneybrooks.com/what-will-transformers-transform/>
- Winters, T., Marra, G., Manhaeve, R., & De Raedt, L. (2021). Deepstochlog: Neural stochastic logic programming. *arXiv preprint arXiv:2106.12574*.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2019a). Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv, abs/1910.03771*.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2019b). Huggingface’s transformers: State-of-the-art natural language processing.
- Wu, H., Mao, J., Zhang, Y., Jiang, Y., Li, L., Sun, W., & Ma, W.-Y. (2019). Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, 6609–6618.
- Wu, Z., Kreiss, E., Ong, D., & Potts, C. (2021). ReaSCAN: Compositional Reasoning in Language Grounding. In J. Vanschoren & S. Yeung (Eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks* (Vol. 1). <https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/2838023a778dfaecd212708f721b788-Paper-round1.pdf>
- Wu, Z., Manning, C. D., & Potts, C. (2023). Recogs: How incidental details of a logical form overshadow an evaluation of semantic interpretation. *arXiv preprint arXiv:2303.13716*.
- Xia, F., Sun, K., Yu, S., Aziz, A., Wan, L., Pan, S., & Liu, H. (2021). Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2(2), 109–127. <https://doi.org/10.1109/TAI.2021.3076021>

- Xie, S., Morcos, A., Zhu, S.-C., & Vedantam, R. (2022). Coat: Measuring object compositionality in emergent representations. *International Conference on Machine Learning*, 24388–24413.
- Xie, Z., & Sun, S. (2019). A goal-driven tree-structured neural model for math word problems. *IJCAI*, 5299–5305.
- Yaghoobzadeh, Y., Kann, K., Hazen, T. J., Agirre, E., & Schütze, H. (2019). Probing for semantic classes: Diagnosing the meaning content of word embeddings. In A. Korhonen, D. R. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th conference of the association for computational linguistics, ACL 2019, volume 1: Long papers* (pp. 5740–5753).
- Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., & Tenenbaum, J. B. (2020). CLEVRER: CoLLision Events for Video REpresentation and Reasoning [arXiv: 1910.01442]. *arXiv:1910.01442 [cs]*. Retrieved October 29, 2021, from <http://arxiv.org/abs/1910.01442>
- Yi, K., Wu, J., Gan, C., Torralba, A., Kohli, P., & Tenenbaum, J. (2018). Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31.
- Yu, F., Tang, J., Yin, W., Sun, Y., Tian, H., Wu, H., & Wang, H. (2021). Ernievil: Knowledge enhanced vision-language representations through scene graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4), 3208–3216.
- Zadeh, A., Zellers, R., Pincus, E., & Morency, L.-P. (2016). Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages. *IEEE Intelligent Systems*, 31(6), 82–88.
- Zhang, B., Hu, H., Qiu, L., Shaw, P., & Sha, F. (2021). Visually Grounded Concept Composition [arXiv: 2109.14115]. *arXiv:2109.14115 [cs]*. Retrieved November 3, 2021, from <http://arxiv.org/abs/2109.14115>
- Zhang, J., Wang, L., Lee, R. K.-W., Bin, Y., Wang, Y., Shao, J., & Lim, E.-P. (2020). Graph-to-tree learning for solving math word problems.
- Zhao, L., Kong, L., Walters, R., & Wong, L. L. (2022). Toward compositional generalization in object-oriented world modeling. *International Conference on Machine Learning*, 26841–26864.

