NeSyCoCo: A *Ne*uro-*Sy*mbolic *Co*ncept *Co*mposer for Compositional Generalization

Danial Kamali, Elham J. Barezi, Parisa Kordjamshidi

Michigan State University

{kamalida, jebalbar, kordjams}@msu.edu

Abstract

Compositional generalization is crucial for artificial intelligence agents to solve complex vision-language reasoning tasks. Neuro-symbolic approaches have demonstrated promise in capturing compositional structures, but they face critical challenges: (a) reliance on predefined predicates for symbolic representations that limit adaptability, (b) difficulty in extracting predicates from raw data, and (c) using non-differentiable operations for combining primitive concepts. To address these issues, we propose NeSyCoCo, a neuro-symbolic framework that leverages large language models (LLMs) to generate symbolic representations and map them to differentiable neural computations. NeSyCoCo introduces three innovations: (a) augmenting natural language inputs with dependency structures to enhance the alignment with symbolic representations, (b) employing distributed word representations to link diverse, linguistically motivated logical predicates to neural modules, and (c) using the soft composition of normalized predicate scores to align symbolic and differentiable reasoning. Our framework achieves state-of-the-art results on the ReaSCAN and CLEVR-CoGenT compositional generalization benchmarks and demonstrates robust performance with novel concepts in the CLEVR-SYN benchmark.

Code — https://github.com/HLR/NeSyCoCo

1 Introduction

Compositional generalization refers to the ability of an intelligent agent to extend its understanding from previously seen components to more complex problems. Our research focuses on vision-language reasoning, an area where compositional generalization plays a crucial role.

While humans can easily extrapolate their understanding of primitive concepts to more complex problems, current state-of-the-art models frequently encounter difficulties in this area, particularly in reasoning about the composition of entity properties or relationships (Partee et al. 1984).

Neuro-symbolic methods have demonstrated great potential in addressing compositional structures (Zhu, Thomason, and Jia 2022). However, existing approaches face several challenges. They require a symbolic representation of the domain, typically involving a set of predefined predicates, which limits the flexibility and coverage of linguistic lexical variety in concepts appearing in language and vision modalities. Moreover, obtaining the domain predicates from the raw modalities at an appropriate level of abstraction is challenging. Additionally, integrating symbolic and neural models requires differentiable operations for composing primitive concepts, which still poses a challenge.

Recent advancements in artificial intelligence have aimed to emulate human reasoning through various models and techniques. Leveraging large language models for visual reasoning has shown notable promise, with methods such as VisProg (Gupta and Kembhavi 2023) and ViperGPT (Surís et al. 2023) demonstrating potential by utilizing LLMs to generate reasoning programs for visual tasks. These models decompose complex linguistic inputs into logical steps, similar to human reasoning. However, a significant limitation is their reliance on a limited set of predefined predicates, which restricts their flexibility and ability to generalize to new scenarios. LEFT (Hsu et al. 2024) addresses the need for predefined predicates in symbolic reasoning by leveraging LLMs to extract trainable and lexically-motivated predicates. Despite this improvement, LEFT struggles with lexical variety in linguistic expressions and handling non-canonical or unseen concepts.

Research indicates that end-to-end vision-language models often struggle with compositional generalization (Zhu, Thomason, and Jia 2022; Yun et al. 2023), frequently failing to generalize beyond their training examples. This limitation highlights the need for approaches that dynamically adapt to new predicates and queries without being constrained by predefined rules (Zhu, Thomason, and Jia 2022). In this work, we propose NeSyCoCo, a novel visual reasoning method designed to address above-mentioned limitations. Following Hsu et al. (2024), our method leverages the power of LLMs, using their natural language vocabulary as a source of predicates and symbols, thus alleviating the need for manual engineering of domain predicates. We augment the linguistic inputs with their syntactic structure to improve the semantic alignment of the symbolic representations generated by LLM. In addition, we use distributed representations of concepts in language as predicate representations and connect these predicates to neural modules to deal with a lexical variety of concepts. We normalize these predicates'

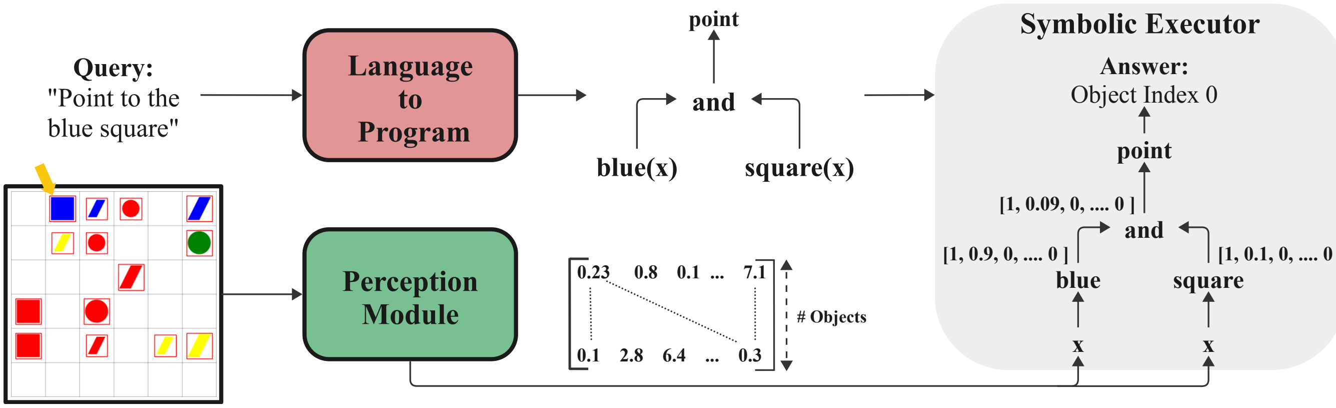

Figure 1: The overall framework of NeSyCoCo. The language-to-program module generates a logical program based on the input query. Predicates, such as `blue`, serve as symbolic representations connected to neural modules that process representations of visual elements. These modules produce scores indicating the applicability of the concept to these elements. Differentiable soft compositional operations are then applied to the scores, executing the program and generating the answer to the query.

outputs, making it easier to compose them with soft operations that align better with task semantics. Soft composition in our framework involves the normalization of predicate scores and the application of composition functions specifically designed to operate effectively on these normalized values, ensuring that all predicates contribute to the composition equally and effectively.

NeSyCoCo excels in reasoning and grounding across multiple problems without relying on a limited set of predicates. It achieves state-of-the-art results on the ReaSCAN compositional generalization and CLEVR-Puzzle benchmarks, while maintaining high accuracy when encountering new and similar concepts in our newly created CLEVR-SYN evaluation benchmark. In summary, our contributions are:

1. **Improved Symbolic Reasoning Engine**: We present an advanced symbolic reasoner that enhances the composition of primitives and interpretability, facilitating better understanding and analysis of the reasoning process.
2. **Handling Language Variety**: Our approach handles the language variety of predicates by utilizing the predicate's distributed linguistic representation.
3. **Introducing an Enhanced Prompting Method**: We propose an improved prompting technique for translating linguistic input to the symbolic programs using additional syntactical information as context, leading to more semantically aligned programs.

2 Related Works

Our framework integrates neural networks, logical reasoning, and large language models to achieve compositional generalization in vision-language reasoning. Therefore, we focus on the four topics below.

2.1 Compositional Generalization in Vision-Language Reasoning

Compositional generalization is a crucial aspect of AI systems, enabling them to handle more complex compositions in vision-language tasks. Recent studies have examined the generalization capabilities of various neural network architectures using specialized evaluation tasks (Hupkes et al. 2020; Ontañón et al. 2021; Csordás, Irie, and Schmidhuber 2021). Benchmarks such as CLEVR (Johnson et al. 2017a), gSCAN (Ruis et al. 2020), and ReaSCAN (Wu et al. 2021) have been developed to assess these capabilities in vision language models. Recent works have introduced advanced transformer-based architectures (Kamali and Kordjamshidi 2023; Sikarwar, Patel, and Goyal 2022; Jiang and Bansal 2021; Qiu et al. 2021), special neural architectures (Kuo, Katz, and Barbu 2021; Gao, Huang, and Mooney 2020; Hsu, Mao, and Wu 2023), meta learning (Xu, Kordjamshidi, and Chai 2023; Xu et al. 2023) and soft prompting (Xu, Kordjamshidi, and Chai 2024), to address compositional generalization. A recent survey overviews compositional learning approaches from theoretical and experimental perspectives, offering insights into the field (Sinha, Premsri, and Kordjamshidi 2024).

Unlike previous end-to-end methods, we employ a neuro-symbolic approach. Recent work has shown that neuro-symbolic methods perform slightly worse than end-to-end models on in-domain problems but better in generalization (Zhu, Thomason, and Jia 2022). NeSyCoCo can generalize to new similar concepts and utilizes soft composition for more primitive concepts.

2.2 Neuro-Symbolic Vision-Language Reasoning

Neuro-symbolic approaches have demonstrated strong vision-language reasoning capabilities by combining symbolic reasoning with neural networks through modular designs. For instance, Neuro-symbolic VQA (Yi et al. 2018) and the Neuro-Symbolic Concept Learner (NSCL) (Mao et al. 2019) have advanced visual reasoning by utilizing symbolic program execution and reducing the need for dense supervision. Despite their successes, these approaches often rely on predefined domain-specific languages and manually implemented programs, which limits their flexibility. The recent model, LEFT (Hsu et al. 2024), utilizes LLMs to generate symbolic representations for linguistic queries, partially alleviating this problem. However, this method struggles with handling the lexical variety in predicate language and fails when faced with novel concepts, demonstrating limited generalization capability. NeSyCoCo addresses these limitations by employing LLMs and normalized predicate outputs for soft composition. This enables more interpretable and flexible compositional generalization. Unlike previous approaches, NeSyCoCo does not depend on a limited set of predefined symbols and can adapt to various predicates, enhancing its ability to generalize across different tasks.

2.3 LLMs for Formal Representation

Leveraging large language models to decompose tasks into sequences of API calls has gained attention in recent research (Cheng et al. 2023; Beurer-Kellner, Fischer, and Vechev 2023; Zelikman et al. 2023; Faghihi et al. 2024). These methods typically focus on the natural language domain, limiting their capacity to ground concepts in visual or other modalities. While LLMs can reason about object categories inferred from language, they cannot recognize objects in a scene or generate robotic actions. Some approaches (Gupta and Kembhavi 2023; Surís et al. 2023) utilize LLMs to generate programs to execute on images but rely on predefined modules without additional training. Similar to LEFT (Hsu et al. 2024), our approach overcomes these limitations by using LLMs to obtain formal representations and leveraging their natural language vocabulary for predicates and symbols, enhancing flexibility and coverage in grounding concepts. Additionally, we employ dependency parsing as an additional context for LLM to improve symbolic program generation.

2.4 General Vision-Language Models

Vision-language models (VLMs) have shown success in multimodal environments. These models integrate vision and language modalities to perform tasks such as image captioning (Xiao et al. 2023), visual question answering (Wang et al. 2024; Liu et al. 2024), and navigation (Zhang and Kordjamshidi 2023; Zhang et al. 2024). Despite their success, end-to-end VLMs often struggle with generalization in novel and complex tasks across different domains (Zhu, Thomason, and Jia 2022; Yun et al. 2023). This limitation arises from their end-to-end nature, which makes it challenging to handle the diverse and intricate relationships between vision and language components. Our approach marks competitive performance with VLMs when faced with new, complex problems.

3 Methodology

This work addresses the challenge of compositional generalization in vision-language reasoning tasks. Our approach involves a bi-modal input system, where the inputs consist of a natural language query and an image that provides context. The objective is to answer the query given the image context accurately.

We introduce the Neuro-Symbolic Concept Composer (NeSyCoCo), a unified neuro-symbolic framework designed to interpret natural language queries by decomposing them into differentiable symbolic functions. These functions are then combined using soft composition techniques to generate accurate responses. As illustrated in Figure 1, the NeSyCoCo comprises three key components:

1. **Natural Language to Program**: Converts natural language queries into symbolic programs, forming the basis for reasoning.
2. **Perception Module**: Extracts domain-specific features, such as objects or relational features, from the input data.
3. **Differentiable Neuro-Symbolic Reasoning Executor**: Executes the symbolic programs, composes the relevant concepts, and generates the final answer to the query.

Our approach utilizes the LEFT implementation (Hsu et al. 2024), serving as the foundation for our framework. We build on this framework by introducing three key improvements: 1) leveraging dependency parsing to achieve more accurate symbolic representations of language in natural language to program, 2) reducing reliance on predefined symbolic predicates by utilizing linguistically motivated distributed representations in neuro-symbolic reasoner, and 3) refining the compositional operations for the soft execution of symbolic programs in neuro-symbolic reasoner.

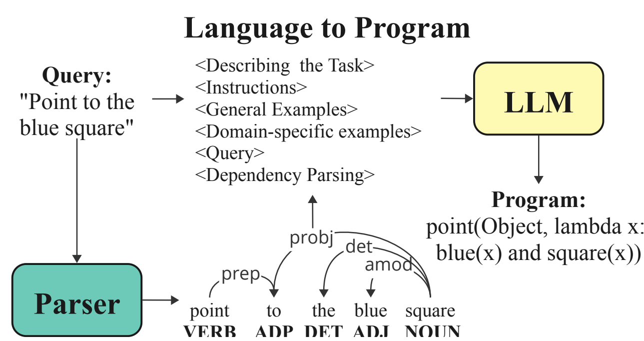

Figure 2: Language to program conversion procedure.

3.1 Natural Language To Program: Exploiting Dependency Parsing

The Natural Language To Program component converts natural language queries into symbolic representations, as illustrated in Figure 2. We use an LLM to generate a program based on the linguistic query. Following LEFT (Hsu et al. 2024), the prompt provided to the backbone model includes syntactical instructions and general examples, supplemented with simple domain-specific examples when necessary to guide the symbolic program generation.

Previous work did not elaborate on improving the natural language to program conversion and simply used LLMs for program generation or assumed it is given (Hsu et al. 2024). However, understanding complex nested and compositional expressions can be challenging for large language models, particularly in capturing structural dependencies. To

Function	Logical Form	Description	Differentiable Implementations	
			LEFT (Hsu et al. 2024)	**NeSyCoCo**
exists(α_x)	$\exists(\alpha_x)$	Existential quantification	$\max(\alpha_x)$	$\max(\alpha_x)$
forall(α_x)	$\forall(\alpha_x)$	Universal quantification	$\min(\alpha_x)$	$\min(\alpha_x)$
and($\alpha_x, \alpha_y, \alpha_z$)	$\alpha_x \wedge \alpha_y \wedge \alpha_z$	Logical conjunction	$\min(\alpha_x, \alpha_y, \alpha_z)$	$\alpha_x \odot \alpha_y \odot \alpha_z$
and(α_x, β_{xy})	$\alpha_x \wedge \beta_{xy}$	Logical conjunction	$\sum_y(\alpha_x \odot \beta_{xy})$	$\max_y(\alpha_x \odot \beta_{xy})$
not(α_x)	$\neg(\alpha_x)$	Logical negation	$-\alpha_x$	$1 - \alpha_x$
iota(var, α_x)	ι(var, α_x)	Variable assignment	softmax(α_x)	$\frac{\alpha_x - \min(\alpha_x)}{\max(\alpha_x) - \min(\alpha_x)}$
count(α_x)	count(α_x)	Counting elements	$\sum \sigma(\alpha_x)$	$\sum \alpha_x$
equal(s_1, s_2)	$s_1 == s_2$	Scalar equality	$\sigma\left(\frac{\tau\cdot(\gamma - \lvert s_1 - s_2 \rvert)}{\gamma}\right)$	$\sigma\left(\frac{\tau\cdot(\gamma - \lvert s_1 - s_2 \rvert)}{\gamma}\right)$
greater_than(s_1, s_2)	$s_1 > s_2$	Scalar inequality	$\sigma(\tau \cdot (s_1 - s_2 - 1 + \gamma))$	$\sigma(\tau \cdot (s_1 - s_2 - 1 + \gamma))$

Table 1: Mathematical Expressions: Logical Forms, Descriptions, and Differentiable Implementations.

address this, and inspired by previous work (Johnson et al. 2017b; Kamali and Kordjamshidi 2023), we incorporate dependency parsing to aid in symbolic program generation. We use the dependency parsing of the query and represent it as a sequence. We enable the model to exploit the dependency structure of the queries to generate better semantically aligned programs.

For instance, consider the query *point to the blue square*, as shown in Figure 2. In this process, the query is first parsed, resulting in a dependency structure such as `[square, pobj, to, ADP, [the, blue]]`. This parsed dependency information is then concatenated with the original query and provided as additional context to the language model. The language model then translates this input into the symbolic program `point(Object, lambda x: blue(x) and square(x))`, as illustrated in Figure 2.

3.2 Perception Module

The Perception Module allows the model to integrate information across multiple modalities. This module extracts features or representations from a secondary modality, such as 2D images. While the perception module can be adapted for various domains, we focus on image perception in this work to support vision-language reasoning. For images, we extract entity-centric representations using Mask RCNN (He et al. 2017), ResNet (He et al. 2016), and PreciseRoIPooling (Jiang et al. 2018). Specifically, given an image with N bounding boxes, this process returns a tensor representation $\mathbf{E}_o^{N\times d_o}$, where d_o is the dimensionality of the visual feature vectors, with each row corresponding to features extracted from an individual bounding box. Additionally, it yields a tensor $\mathbf{E}_r^{N\times N\times d_r}$ to represent relational features for each pair of bounding boxes in the image.

3.3 Differentiable Neuro-Symbolic Reasoning Module

This module executes the programs once the natural language queries are converted into symbolic programs and the visual representations are extracted. These programs consist of two types of functions: domain predicates and first-order logic operations.

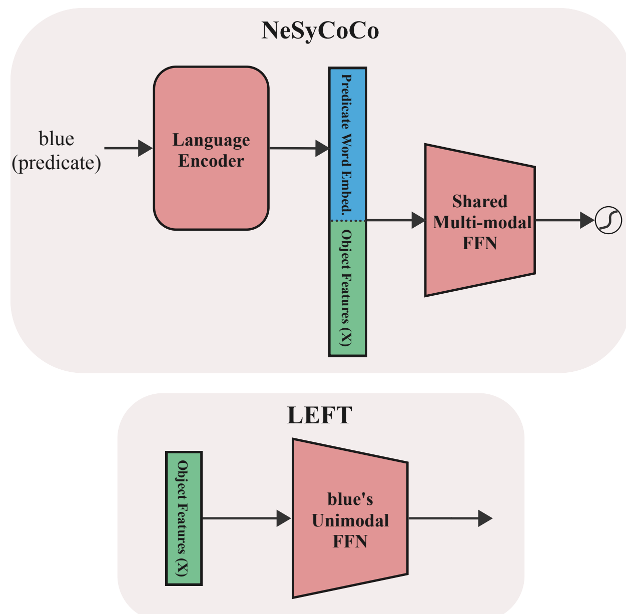

Figure 3: Differentiable predicate function in NeSyCoCo (shared FFN for all predicates) compared to LEFT (predicate-specific FFNs) calculating the score for `blue`.

Domain Predicates: Exploiting Linguistically-motivated Distributed Representations of Symbolic Predicates Domain predicates are neural functions that assign values to objects or relations based on specific traits. For single-object (unary) predicates, functions such as `big(x)` or `red(x)` generate a score α_x, indicating the extent to which object x exhibits the predicate's characteristic. For multi-object (binary and ternary) predicates, such as `same_row(x, y)`, the function returns a score β_{xy}, reflecting the relationship between objects x and y according to the predicate. In contrast with previous works, to address the problem of linguistic lexical variety, we utilize distributed linguistic representation. Therefore, in NeSyCoCo, each domain predicate, regardless of its object-arity, such as `inside(., .)` or `red(.)` is represented with a vector embedding from an off-the-shelf language encoder resulting in $\mathbf{E}_w^{1\times d_w}$. As shown in Figure 3, at the domain predicate level, the predicate's embedding is concatenated with the perception module's representation and passed to a shared multilayer Feed Forward Network (FFN) to generate a score for the combination of linguistic predicate and the perceived visual representation. For parallelization, the $\mathbf{E}_w^{1\times d_w}$ is repeated to calculate the scores for multiple objects simultaneously.

$$\alpha = \sigma\left(FNN(\mathbf{E}_p^{N\times d_o} \| \mathbf{E}_w^{N\times d_w})\right)$$
$$\beta = \sigma\left(FNN(\mathbf{E}_r^{N\times N\times d_r} \| \mathbf{E}_w^{N\times N\times d_w})\right)$$

This approach alleviates the need for canonical predicates and helps learn the parameters of linguistic predicates with similar semantics.

First Order Logic Executor: Utilizing Soft Composition NeSyCoCo programs execute recursively starting with an expression such as `point(Object, lambda x: expr)` or `count(Object, lambda x: expr)`, where a variable is defined, and a sub-expression is embedded. This sub-expression is processed recursively through tensor functions. We employ soft operations rather than relying on the hard, discrete operations typical of traditional symbolic reasoning. This approach allows us to backpropagate errors throughout the pipeline, enabling effective training of primitive predicates.

The LEFT approach to symbolic execution has two major issues. First, it uses scalar values, which complicates the composition of scores across different scales. In contrast, NeSyCoCo utilizes a sigmoid activation function at the predicate level to normalize the scores. This technique controls predicate scores, minimizes error ranges, and reduces the risk of propagating failures, enhancing model robustness. Moreover, it allows us to use multiplication instead of the `min` function for composition, ensuring that all predicates contribute to the composition equally. It also allows for simultaneous training of multiple concepts through backpropagation, which is impossible when using the *min* function.

Second, using softmax in the `iota` variable assignment function presents another challenge. When the model encounters multiple matching objects for an expression, softmax can either reduce the output score if the objects receive similar scores or exaggerate differences in scores, making reasoning more difficult. For example, if two objects have similar scores for a predicate, applying softmax can inflate the difference between their scores, distorting the reasoning process by assigning inflated scores. To alleviate this limitation, we utilize linear normalization at the `iota` function.

To ensure compatibility, we updated other composition functions to align with this normalization strategy, as shown in Table 1. By addressing these issues, our approach significantly improves composition and interoperability, leading to superior performance in vision-language tasks, particularly in compositional generalization.

4 Experiments

We evaluate our method across three key aspects: compositional generalization, vision-language reasoning, and handling linguistic variety. We present our experiments on ReaSCAN (Wu et al. 2021) and CLEVR-CoGenT (Johnson et al. 2017a) for compositional generalization. In the context of visual reasoning, we discuss our experiments and findings using the CLEVR dataset and its extensions. Finally, to assess how our neuro-symbolic methods handle linguistic variety, we introduce a new benchmark called CLEVR-SYN.

4.1 Experimental Setting

Our implementation is based on the PyTorch deep learning library (Paszke et al. 2019), with the SpaCy toolkit (Honnibal and Montani 2017) used for extracting dependency parsing of natural language queries. We employed the LLaMA-3.1 70B model (Dubey, Jauhri, and Others 2024) with 4-bit quantization as our primary language model, selected for its open-source availability and performance parity with GPT-3.5, allowing us to maintain transparency and adaptability in our experiments. The experiments' vector embeddings for predicates and other linguistic components were derived from GloVe (Pennington, Socher, and Manning 2014) language encoder. More details about the experimental settings can be found in Appendix B.

4.2 Compositional Generalization

To demonstrate our method's compositional generalization capability, we evaluated NeSyCoCo using two commonly used benchmarks, CLEVR-CoGenT and ReaSCAN.

CLEVR CoGenT Benchmark The CLEVR Compositional Generalization Task (CoGenT) extends the original CLEVR dataset to test model generalization to novel unseen combinations of visual attributes. This task has two splits, each with distinct attribute distributions. In test split A, cubes are restricted to gray, blue, brown, or yellow, while cylinders are limited to red, green, purple, or cyan. Split B swaps these color sets between cubes and cylinders. Spheres in both splits can appear in any color. Models are trained on biased distributions and tested on unseen attribute combinations, challenging them to develop compositional representations rather than relying on memorization.

We evaluate the model performance using the exact match accuracy of the generated response. As shown in Table 3, our model achieved state-of-the-art results in generalization, outperforming both LEFT and MDETR (Kamath et al. 2021) methods. This outcome is consistent with recent work (Yun et al. 2023), suggesting that neuro-symbolic approaches surpass end-to-end methods in generalization while showing slightly worse in-domain performance. Our model's performance on CLEVR-CoGenT indicates its strong ability to address the compositional challenges in this benchmark, demonstrating robust generalization.

ReaSCAN Benchmark To further examine our model's compositional generalization capabilities, we evaluated it on the ReaSCAN (Wu et al. 2021) dataset, which is specifically designed to test compositional generalization in grounded language understanding. ReaSCAN consists of instructions for an agent to perform tasks within a 2D environment, requiring an understanding of spatial relationships and object properties. The goal is to assess how well vision-language models generalize from familiar linguistic inputs to novel

	A1 (%)	A2 (%)	A3 (%)	B1 (%)	B2 (%)	C1 (%)	C2 (%)	Avg (%)
GroCoT	99.4	85.7	95.4	90.4	83.5	70.2	27.7	78.5
Syntax Guided Transformer	**99.6**	**97.3**	**99.6**	95.4	90.1	92.5	21.7	85.2
NeSyCoCo (Full)	99.1 ± 0.14	94.1 ± 0.02	98.5 ± 0.37	**98.9** ± 0.05	**98.7** ± 0.08	**96.8** ± 0.12	**95.9** ± 0.05	**97.5** ± 0.05
w/o Embedding	99.5 ± 0.05	91.7 ± 0.41	99.1 ± 0.24	97.6 ± 0.18	97.7 ± 0.16	94.5 ± 0.16	95.9 ± 0.06	96.5 ± 0.07
w/o Emb. w/o Soft Reasoner	97.8 ± 0.35	80.3 ± 1.13	95.5 ± 0.82	93.0 ± 0.25	95.8 ± 0.95	92.2 ± 0.82	92.9 ± 0.20	92.5 ± 0.23
LEFT†	97.8 ± 0.35	80.3 ± 1.13	95.5 ± 0.81	93.0 ± 0.25	95.8 ± 0.95	92.2 ± 0.82	91.5 ± 0.16	92.3 ± 0.14

Table 2: The accuracy of our proposed model on the ReaSCAN test split grounding task compared to neuro-symbolic and end-to-end methods such as GroCoT (Sikarwar, Patel, and Goyal 2022) and Syntax Guided Transformer (Kamali and Kordjamshidi 2023).† The results on LEFT are reported without dependency parsing in the context. w/o Emb. w/o Soft Reasoner shows the results of LEFT with improved prompting. The reported results are the average accuracy and standard deviation of three runs.

Method	Split A (%)	Split B (%)
MDETR (Kamath et al. 2021)	**99.7**	76.2
LEFT (Hsu et al. 2024)	99.5	76.2
NeSyCoCo	99.6 ± 0.08	**78.8** ± 0.15

Table 3: Accuracy on the CLEVR-CoGenT benchmark reported on the average of three runs.

combinations of learned concepts. This dataset is crucial for testing model's ability to interpret linguistic commands, offering insights into its capacity for compositional and grounded language processing. ReaSCAN includes seven compositional test splits with specific held-out combinations compared to the training data:

- A1: `yellow square` referred with color and shape.
- A2: `red square` referred anywhere in the command.
- A3: `small cylinder` referred to by size and shape.
- B1: Co-occurrences of a `small red circle` and a `large blue square`.
- B2: Co-occurrences of `same size as` and `inside of` relationships.
- C1: Three relative clause commands.
- C2: Two relative clause using `that is` instead of `and`.

Recent research identifies grounding as the key challenge in the ReaSCAN dataset, and accurate grounding enables perfect navigation step generation (Sikarwar, Patel, and Goyal 2022). Thus, we concentrated on grounding in ReaSCAN, leaving navigation for future work. Therefore, we use the accuracy of the *object localization* for evaluation.

As shown in Table 2, our model outperformed the baseline on this compositional generalization benchmark and surpassed previous non-symbolic methods across B and C test splits while showing competitive results on A split. In addition, our analysis of the wrong cases revealed that our method has difficulty handling size-related concepts. In ReaSCAN, size is a contextual and relative concept, while we handle size using unary object-level functions. Hence, our method struggles to interpret these concepts accurately.

We performed an ablation study to show the significance of each proposed component of our approach. As shown in Table 2, our three technical components contribute to obtaining the SOTA performance. Among them, the soft symbolic reasoner has the most substantial impact on compositional generalization, as evidenced by a paired t-test ($p = 0.0026$), confirming that the observed improvements are statistically significant. In addition, employing word embedding for predicate representation marks a high performance by handling a variety of predicates. As a matter of fact, obtaining an accurate program is a crucial step to achieving a precise reasoning model, which was provided by an improved prompting method that shows improvement, mostly in complex cases such as C1 and C2 test splits.

As mentioned in Section 3.3, LEFT (Hsu et al. 2024) uses raw unbounded logits as concept scores. We analyzed the scale of concept scores given by the LEFT predicate function on the CLEVR dataset, as illustrated in Figure 4; the score ranges for different concepts can vary significantly. This becomes problematic when combining concepts like `red` and `rubber` using a *min* function. In such cases, the score for `red` is often undervalued, leading to a biased composition that fails to reflect the true relationship between the concepts. As shown in the ablation study in Table 2, NeSyCoCo outperforms previous work on compositional generalization largely due to its use of soft composition functions for combining primitives during symbolic program execution, which shows the positive effect of our modifications.

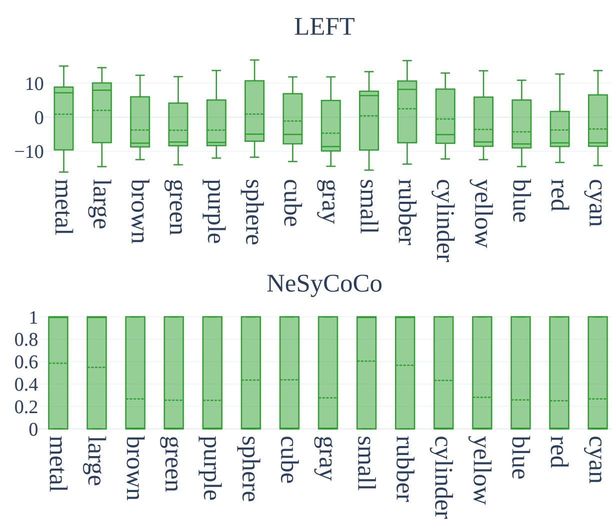

Figure 4: Boxplot comparing concept scores of LEFT and NeSyCoCo on 10k CLEVR validation samples. Dotted and solid lines represent the mean and median, respectively.

4.3 Vision-Language Reasoning

To evaluate our method's vision-language reasoning capability, we evaluated our model's performance on the CLEVR dataset and some of its extensions.

CLEVR and Extensions The CLEVR dataset (Johnson et al. 2017a) is a benchmark for vision-language reasoning, featuring synthetic images of diverse objects and associated questions assessing tasks like counting, attribute comparison, and spatial understanding. CLEVR's structured design effectively tests models' compositional learning and reasoning capabilities using the accuracy of generated answers. As shown in Table 4, our model outperforms previous neuro-symbolic methods and demonstrates competitive performance to end-to-end models showing its advanced vision-language reasoning.

Method	**Type**	**Accuracy** (%)
MDETR (Kamath et al. 2021)	End-to-End	**99.7**
NSCL (Mao et al. 2019)†	Neuro-Symbolic	98.7
LEFT (Hsu et al. 2024)	Neuro-Symbolic	99.6
NeSyCoCo	Neuro-Symbolic	**99.7** $\pm$ 0.02

Table 4: Accuracy on the validation set of the CLEVR dataset reported on the average of three runs. † indicates the use of pre-defined predicates

We also evaluate our method on CLEVR-based extensions introduced in Hsu et al. (2024), including CLEVR-Ref for referring expressions, CLEVR-Puzzle for multi-step reasoning, and CLEVR-RPM for abstract reasoning. As shown in Table 5, our model achieves 100% accuracy on CLEVR-Ref and CLEVR-RPM, and 95% on CLEVR-Puzzle, outperforming all previous models. Human evaluations indicate that the five errors in the CLEVR-Puzzle were due to incorrect annotations. Otherwise, the models would also obtain 100% accuracy on this test setting.

Model	**Ref**	**Puzzles**	**RPM**
NeSyCoCo + GT programs	**100**%	**95**%	**100**%
LEFT + GT programs	100%	92%	100%
LEFT + LLM programs	94%	75%	87%
LLaVA NeXT(Li et al. 2023)	N/A	45%	52%
OpenFlamingo(Awadalla et al. 2023)	N/A	57%	52%
ViperGPT (Surís et al. 2023)	8%	34%	4%
VisProg (Gupta and Kembhavi 2023)	35%	27%	51%

Table 5: Accuracy on CLEVR extension tasks.

4.4 Linguistic Lexical Variety

CLVER-SYN Since we are using a frozen vector-based representation of predicates, our model should be able to deal with new and similar concepts when faced with new and similar concepts. To showcase this capability, we created a new benchmark based on the CLEVR dataset validation set called the CLEVR synonym (CLEVR-SYN) benchmark for neuro-symbolic methods. This benchmark aims to evaluate the performance of a neuro-symbolic method trained with original CLEVR programs when faced with new concepts. This benchmark consists of three test splits using all of the samples in the CLEVR validation split. In the test splits of this dataset, concepts in programs have been replaced with unseen but similar primitive concepts. Table 6 shows the concepts and their substitutes. The splits of this dataset are easy, medium, and hard tests. In the easy test, only one concept has changed in the program. In the medium test, a maximum of three concepts have been replaced. In the hard test, all concepts in the list are replaced. We replaced these concepts using regular expressions search in the programs given by the CLEVR dataset.

We evaluated our model using these three different splits. As shown in Table 7, our model maintains high accuracy on the CLEVR-SYN benchmark and outperforms previous work in zero-shot concept generalization. The previous approaches falter mainly due to their inability to handle new but semantically similar predicates effectively. They will either fail due to the lack of trained predicates or resort to random initialization for new predicates.

Predicate	**Similar Concept**	ρ	**p-value**
cube	box	0.06	0.44
sphere	ball	-0.02	0.49
large	huge	0.97	0.01
small	little	0.91	0.04
metal	metallic	0.69	0.15
rubber	elastic	0.68	0.18
red	burgundy	0.15	0.47
blue	azure	0.55	0.34
brown	chocolate	0.20	0.44
yellow	mustard	0.05	0.51
left	left_of	1.00	0.00
front	front_of	0.98	0.01
same_color	matching_color	0.86	0.06
same_material	identical_material	0.75	0.11
same_shape	congruent_shape	0.66	0.17

Table 6: CLEVR-SYN substitutions predicates and their score's Pearson correlation score CLEVR validation set.

Method	**Easy** (%)	**Medium** (%)	**Hard** (%)
LEFT	81.9 $\pm$ 0.19	64.8 $\pm$ 0.26	49.5 $\pm$ 0.64
NeSyCoCo	**92.1** $\pm$ 0.26	**81.2** $\pm$ 0.58	**73.4** $\pm$ 0.66

Table 7: Accuracy on the CLEVR synonym benchmark reported on the average of three runs.

Analysis We further analyzed the synonym dataset to evaluate our method's performance on individual substitutions by measuring the correlation between replaced concept scores and the original predicate's scores using the Pearson correlation (ρ) (Benesty et al. 2009) metric and p-value to show the significance of the correlation. As shown in Table 6, our model effectively captures nuanced relationships across various attributes, as evidenced by significant correlations between semantically similar concepts. Notably, terms like `large/huge` and `small/little` exhibit strong positive correlations with high p-values with their corresponding learned concepts. The model also shows strong generalization for multi-token predicates like `same material/identical material`, crucial for handling non-canonical predicates in a neuro-symbolic system.

However, while our method performs well in 9 out of 15 cases (correlation higher than 0.6), it encounters difficulties with certain predicates, such as `ball` vs `sphere` or `brown` compared to `chocolate`. These challenges likely arise from the nuanced semantic or contextual differences between these terms, which frozen embeddings do not capture fully. For example, `brown` and `chocolate` may overlap semantically as two similar colors but often differ in contextual usage and sensory perception.

To investigate this further, we analyzed the relationship between the cosine similarity of predicates' embeddings and the correlations in their scores. As shown in Figure 5, there is a strong positive correlation between embedding similarity and predicate scores, suggesting that our model effectively generalizes when embeddings reflect semantic closeness (cosine similarity higher than 0.4). This finding underscores the importance of high-quality embedding representations for neuro-symbolic models, particularly for predicates with intricate or context-sensitive meanings. Addressing these limitations could enhance the model's generalization ability across a broader range of predicates. More evaluation on different encoders can be found in Appendix A.

These results support our argument that our neuro-symbolic method can generalize to new, unseen concepts, especially in domains where linguistic relationships are reflected in distributed representations. This ability to generalize is crucial for developing AI systems capable of adapting to novel situations, highlighting the potential of neuro-symbolic approaches in achieving robust and flexible AI.

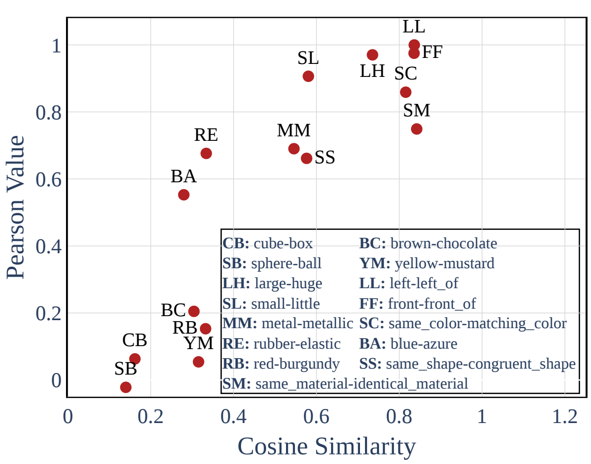

Figure 5: Relationship between cosine similarity of word embeddings and correlation of their predicate scores.

5 Conclusion

In this paper, we introduced NeSyCoCo to address fundamental challenges in the existing neuro-symbolic frameworks. We utilized the soft composition of normalized predicate functions to compose primitive concepts more effectively, thereby enhancing the model's reasoning capabilities and compositional generalization. In addition, by integrating word embeddings of the symbolic predicates, NeSyCoCo effectively addresses the challenge of linguistic variability, enabling zero-shot generalization to novel but semantically related concepts. Furthermore, using syntactical parsing as additional context for the large language model during program generation enhances the precision and accuracy of the generated symbolic programs. These contributions address challenges in bridging symbolic reasoning and neural network-based approaches. As a result, our approach demonstrated state-of-the-art performance on challenging benchmarks such as ReaSCAN and CLVER-CoGenT. One interesting future direction is to integrate the NeSyCoCo approach in the generic neuro-symbolic framework DomiKnowS (Guo et al. 2020; Faghihi et al. 2021; Rajaby Faghihi et al. 2023) to make the underlying neural models programmable instead of using a fixed neural architecture for predicates.

A Choice of Language Encoder

For the language encoder selection, we tested RoBERTa (Liu et al. 2019), Spacy (Honnibal and Montani 2017), GloVe 6B-300D (Pennington, Socher, and Manning 2014), and one-hot encoding. Our experiments demonstrated that all the models performed well on the original CLEVR validation set. Notably, GloVe exhibited strong generalization on the CLEVR-SYN dataset, as presented in Table 8.

Language Encoder	CLEVR	CLEVR-SYN Easy
Spacy	99.7%	80.7%
RoBERTa	99.7%	84.7%
One-hot	99.7%	-
Glove-6B-300D	99.7%	92.1%

Table 8: Accuracy of NeSyCoCo with different language encoders on the CLEVR dataset and CLEVR-SYN Easy split.

B Experimental Setting

All experiments were conducted on Ubuntu OS with an AMD EPYC 7413 24-core CPU and an NVIDIA A6000 GPU, featuring 48GB of memory and 700GB of RAM. The code, generated data, and necessary dependencies listed in the code are available in the Github repository. The hyperparameters used in the experiments are detailed in Table 9.

Ethical Statement

While our method demonstrates considerable improvements in compositional reasoning, it is not without notable limitations. Most of our experiments were conducted on synthetic datasets, which offered a controlled environment to

Hyperparameters	ReaSCAN	CLVER	CLVER-CoGenT
Shared FNN	[1024,512, 256, 128, 1]	[1024,512, 256, 128, 1]	[1024,512, 256, 128, 1]
Visual Repr. Projection	512	512	512
Predicate Repr. Projection	512	512	512
Learning Rate	$\{10^{-3}, \mathbf{10^{-4}}, 10^{-5}\}$	$\{10^{-2}, 10^{-3}, \mathbf{10^{-4}}, 10^{-5}\}$	$\{10^{-3}, \mathbf{10^{-4}}, 10^{-5}\}$
Batch Size	32	32	32
Number of Parameters	14.2M	14.3M	14.3M
Epochs	100	100	100
Curriculum Learning	Yes	Yes	Yes
Language Encoder	Glove-6B-300D	Glove-6B-300D	Glove-6B-300D
Embedding Size	300	300	300

Table 9: Hyperparameters of NeSyCoCo for ReaSCAN, CLVER, and CLVER-CoGenT

evaluate model performance, particularly on compositional generalization. However, these datasets may not fully encapsulate the complexity and variability of real-world scenarios, highlighting the need for future evaluations on diverse, real-world datasets to ensure broader practical applicability. Additionally, the reasoning capabilities of NeSyCoCo depend on programs generated by pre-trained language models that were not specifically optimized for this task. Although syntax errors in LLM-generated programs can be mitigated through detection and resampling, unresolved semantic errors remain a significant challenge. Lastly, while our method exhibits superior generalization ability compared to related work, its predicate generalization performance remains dependent on the choice of distributed representation, which may limit its adaptability in certain scenarios.

Acknowledgments

This project is supported by the Office of Naval Research (ONR) grant N00014-23-1-2417. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Office of Naval Research.

References

Awadalla, A.; Gao, I.; Gardner, J.; Hessel, J.; Hanafy, Y.; Zhu, W.; Marathe, K.; Bitton, Y.; Gadre, S.; Sagawa, S.; et al. 2023. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*.

Benesty, J.; Chen, J.; Huang, Y.; and Cohen, I. 2009. *Pearson Correlation Coefficient*, 1–4. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-00296-0.

Beurer-Kellner, L.; Fischer, M.; and Vechev, M. 2023. Prompting is programming: A query language for large language models. *Proceedings of the ACM on Programming Languages*, 7(PLDI): 1946–1969.

Cheng, Z.; Xie, T.; Shi, P.; Li, C.; Nadkarni, R.; Hu, Y.; Xiong, C.; Radev, D.; Ostendorf, M.; Zettlemoyer, L.; Smith, N. A.; and Yu, T. 2023. Binding Language Models in Symbolic Languages. *ICLR*, abs/2210.02875.

Csordás, R.; Irie, K.; and Schmidhuber, J. 2021. The Devil is in the Detail: Simple Tricks Improve Systematic Generalization of Transformers.

Dubey, A.; Jauhri, A.; and Others. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Faghihi, H. R.; Guo, Q.; Uszok, A.; Nafar, A.; Raisi, E.; and Kordjamshidi, P. 2021. DomiKnowS: A Library for Integration of Symbolic Domain Knowledge in Deep Learning.

Faghihi, H. R.; Nafar, A.; Uszok, A.; Karimian, H.; and Kordjamshidi, P. 2024. Prompt2DeModel: Declarative Neuro-Symbolic Modeling with Natural Language. In Besold, T. R.; d'Avila Garcez, A.; Jimenez-Ruiz, E.; Confalonieri, R.; Madhyastha, P.; and Wagner, B., eds., *Neural-Symbolic Learning and Reasoning*, 315–327. Cham: Springer Nature Switzerland. ISBN 978-3-031-71170-1.

Gao, T.; Huang, Q.; and Mooney, R. 2020. Systematic Generalization on gSCAN with Language Conditioned Embedding.

Guo, Q.; Faghihi, H. R.; Zhang, Y.; Uszok, A.; and Kordjamshidi, P. 2020. Inference-Masked Loss for Deep Structured Output Learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020)*.

Gupta, T.; and Kembhavi, A. 2023. Visual Programming: Compositional visual reasoning without training . 14953–14962.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. B. 2017. Mask R-CNN. *CoRR*, abs/1703.06870.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Honnibal, M.; and Montani, I. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Hsu, J.; Mao, J.; Tenenbaum, J.; and Wu, J. 2024. What's left? concept grounding with logic-enhanced foundation models. *Advances in Neural Information Processing Systems*, 36.

Hsu, J.; Mao, J.; and Wu, J. 2023. DisCo: Improving Compositional Generalization in Visual Reasoning through Distribution Coverage. *Transactions on machine learning research*.

Hupkes, D.; Dankers, V.; Mul, M.; and Bruni, E. 2020. Compositionality Decomposed: How do Neural Networks Generalise? (Extended Abstract). In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 5065–5069. International Joint Conferences on Artificial Intelligence Organization. Journal track.

Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; and Jiang, Y. 2018. Acquisition of Localization Confidence for Accurate Object Detection.

Jiang, Y.; and Bansal, M. 2021. Inducing Transformer's Compositional Generalization Ability via Auxiliary Sequence Prediction Tasks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6253–6265. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Johnson, J.; Hariharan, B.; van der Maaten, L.; Fei-Fei, L.; Zitnick, C. L.; and Girshick, R. 2017a. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *CVPR*.

Johnson, J.; Hariharan, B.; van der Maaten, L.; Hoffman, J.; Fei-Fei, L.; Zitnick, C. L.; and Girshick, R. 2017b. Inferring and Executing Programs for Visual Reasoning.

Kamali, D.; and Kordjamshidi, P. 2023. Syntax-Guided Transformers: Elevating Compositional Generalization and Grounding in Multimodal Environments. In Hupkes, D.; Dankers, V.; Batsuren, K.; Sinha, K.; Kazemnejad, A.; Christodoulopoulos, C.; Cotterell, R.; and Bruni, E., eds., *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, 130–142. Singapore: Association for Computational Linguistics.

Kamath, A.; Singh, M.; LeCun, Y.; Synnaeve, G.; Misra, I.; and Carion, N. 2021. MDETR - Modulated Detection for End-to-End Multi-Modal Understanding. 1760–1770.

Kuo, Y.-L.; Katz, B.; and Barbu, A. 2021. Compositional Networks Enable Systematic Generalization for Grounded Language Understanding. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Findings of the Association for Computational Linguistics: EMNLP 2021*, 216–226. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Li, X. L.; Jiang, H.; Peng, Y.; Wang, L.; Lin, X.; Tu, C.-C.; Savani, Y.; Fu, D.; Ou, J.; Zhou, D.; Ma, T.; and Liang, P. 2023. Llava: Large Language and Vision Assistant. *arXiv preprint arXiv:2305.11495*.

Liu, H.; Li, C.; Li, Y.; Li, B.; Zhang, Y.; Shen, S.; and Lee, Y. J. 2024. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.

Mao, J.; Gan, C.; Kohli, P.; Tenenbaum, J. B.; and Wu, J. 2019. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *International Conference on Learning Representations*.

Ontañón, S.; Ainslie, J.; Cvicek, V.; and Fisher, Z. 2021. Making Transformers Solve Compositional Tasks. 1: 3591–3607.

Partee, B.; et al. 1984. Compositionality. *Varieties of formal semantics*, 3: 281–311.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *ArXiv*, abs/1912.01703.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global Vectors for Word Representation. volume 14, 1532–1543.

Qiu, L.; Hu, H.; Zhang, B.; Shaw, P.; and Sha, F. 2021. Systematic Generalization on gSCAN: What is Nearly Solved and What is Next? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2180–2188. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Rajaby Faghihi, H.; Nafar, A.; Zheng, C.; Mirzaee, R.; Zhang, Y.; Uszok, A.; Wan, A.; Premsri, T.; Roth, D.; and Kordjamshidi, P. 2023. GLUECons: A Generic Benchmark for Learning under Constraints. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8): 9552–9561.

Ruis, L.; Andreas, J.; Baroni, M.; Bouchacourt, D.; and Lake, B. 2020. A benchmark for systematic generalization in grounded language understanding.

Sikarwar, A.; Patel, A.; and Goyal, N. 2022. When Can Transformers Ground and Compose: Insights from Compositional Generalization Benchmarks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 648–669. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Sinha, S.; Premsri, T.; and Kordjamshidi, P. 2024. A Survey on Compositional Learning of AI Models: Theoretical and Experimental Practices. *Transactions on Machine Learning Research*. Survey Certification.

Surís, D.; Menon, S.; Vondrick, C.; and . 2023. ViperGPT: Visual Inference via Python Execution for Reasoning.

Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Fan, Y.; Dang, K.; Du, M.; Ren, X.; Men, R.; Liu, D.; Zhou, C.; Zhou, J.; and Lin, J. 2024. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191*.

Wu, Z.; Kreiss, E.; Ong, D. C.; and Potts, C. 2021. ReaSCAN: Compositional Reasoning in Language Grounding. *NeurIPS 2021 Datasets and Benchmarks Track*.

Xiao, B.; Wu, H.; Xu, W.; Dai, X.; Hu, H.; Lu, Y.; Zeng, M.; Liu, C.; and Yuan, L. 2023. Florence-2: Advancing a unified representation for a variety of vision tasks. *arXiv preprint arXiv:2311.06242*.

Xu, G.; Kordjamshidi, P.; and Chai, J. 2023. MetaReVision: Meta-Learning with Retrieval for Visually Grounded Compositional Concept Acquisition. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 12224–12236. Singapore: Association for Computational Linguistics.

Xu, G.; Kordjamshidi, P.; and Chai, J. 2024. GIPCOL: Graph-Injected Soft Prompting for Compositional Zero-Shot Learning. In *IEEE/CVF Winter Conference on Applications of Computer Vision*.

Xu, L.; Huang, M. H.; Shang, X.; Yuan, Z.; Sun, Y.; and Liu, J. 2023. Meta Compositional Referring Expression Segmentation. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19478–19487.

Yi, K.; Wu, J.; Gan, C.; Torralba, A.; Kohli, P.; and Tenenbaum, J. 2018. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31.

Yun, T.; Bhalla, U.; Pavlick, E.; and Sun, C. 2023. Do Vision-Language Pretrained Models Learn Composable Primitive Concepts? *Transactions on Machine Learning Research*.

Zelikman, E.; Huang, Q.; Poesia, G.; Goodman, N.; and Haber, N. 2023. Parsel: Algorithmic Reasoning with Language Models by Composing Decompositions. *Advances in Neural Information Processing Systems*, 36: 31466–31523.

Zhang, Y.; and Kordjamshidi, P. 2023. VLN-Trans: Translator for the Vision and Language Navigation Agent. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Zhang, Y.; Ma, Z.; Li, J.; Qiao, Y.; Wang, Z.; Chai, J.; Wu, Q.; Bansal, M.; and Kordjamshidi, P. 2024. Vision-and-language navigation today and tomorrow: A survey in the era of foundation models. *arXiv preprint arXiv:2407.07035*.

Zhu, W.; Thomason, J.; and Jia, R. 2022. Generalization Differences between End-to-End and Neuro-Symbolic Vision-Language Reasoning Systems. 4697–4711.