# Syntax-Guided Transformers: Elevating Compositional Generalization and Grounding in Multimodal Environments

**Danial Kamali**
Michigan State University
kamalida@msu.edu

**Parisa Kordjamshidi**
Michigan State University
kordjams@msu.edu

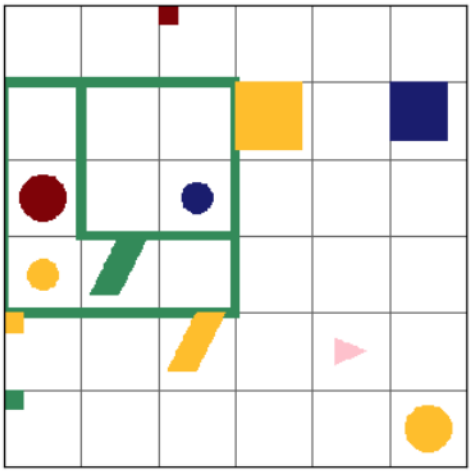
## Abstract

Compositional generalization, the ability of intelligent models to extrapolate understanding of components to novel compositions, is a fundamental yet challenging facet in AI research, especially within multimodal environments. In this work, we address this challenge by exploiting the syntactic structure of language to boost compositional generalization. This paper elevates the importance of syntactic grounding, particularly through attention masking techniques derived from text input parsing. We introduce and evaluate the merits of using syntactic information in the multimodal grounding problem. Our results on grounded compositional generalization underscore the positive impact of dependency parsing across diverse tasks when utilized with Weight Sharing across the Transformer encoder. The results push the state-of-the-art in multimodal grounding and parameter-efficient modeling and provide insights for future research.

## 1 Introduction

Compositional Generalization refers to the ability of an intelligent agent to generalize its understanding of the underlying structure of a problem, especially when it is faced with novel compositions of the previously seen building blocks or components (Chomsky, 1957; Montague, 1970). It is fundamental for models to be able to extrapolate from their training environment to novel situations, a common occurrence in real-world applications. Hupkes et al. (2020) categorizes compositional generalization capabilities into five categories, *systematicity*, *substitutivity*, *localism & globalism*, and *overgeneralization*. These abilities are crucial for models to achieve strong performance on tasks that require reasoning and understanding of hierarchical structures, such as natural language understanding, object classification, and robotics.

Humans understand new compositions of previously observed concepts and simpler constructs.



**Input Command:** pull the small blue object that is inside of the small green box and in the same row as the red circle while zigzagging.
**Action sequence:** *turn left, turn left, walk, turn right, walk, turn right, walk, pull*

Figure 1: This example is taken from the ReaSCAN dataset. Here, an agent is provided with a command. Its objective is to generate/execute a series of predefined actions to fulfill the task within the given environment.

On the other hand, despite remarkable progress in the field of Artificial Intelligence, even state-of-the-art language models demonstrate limitations in this aspect (Lake and Baroni, 2018; Thomas McCoy et al., 2020; Shaw et al., 2021). Especially, they often fail to effectively generalize in the reasoning depth, which involves handling multi-turn reasoning about entities and their properties in the world or even the co-occurrence of unseen spatial relations (Wu et al., 2021). These limitations indicate a crucial need for innovative approaches to address these issues.

In this research, our objective is to exploit the syntactic structure of language to enhance compositional generalization. Our focus is mainly on the multimodal problem setting that entangles vision and language. In this unique setting, compositional



*Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pages 130–142

linguistic descriptions must be accurately grounded in the environment to devise coherent action plans or achieve specific goals. An illustrative example of this scenario is shown in Figure 1.

The motivation behind leveraging syntax in our approach stems from the inherent structure and compositionality of natural language. Syntactic parsing provides crucial structural information about how words in a sentence relate to each other. We hypothesize that syntactic structure can improve intelligent agents' ability to discern the applicable attributes and descriptions for each object in its environment and better apprehend deeper levels of reasoning.

By imposing an understanding of language structure through syntactic parsing, we aim to extend the capabilities of current multimodal language models. This could potentially pave the way for more sophisticated models capable of robustly interacting with dynamic and complex vision and language environments. Apart from using structure, we equipped our end-to-end model with weight sharing that has demonstrated improving the generalization capabilities in single-modality tasks.

As a result, we reach state-of-the-art performance on the ReaSCAN compositional generalization benchmark, showing improvement across all test splits, especially ones requiring sentence structure comprehension. In summary, our contributions include:

- Enhancing grounded compositional generalization by integrating syntactic parsing into our model.
- Using syntax-guided attention masking along with weight sharing, we build a highly parameter-efficient model compared to baselines.
- Our model has shown marked improvement in performance across a variety of tasks that are designed for compositional generalization evaluation while enhancing computational efficacy.

## 2 Related Work

The machine learning research community primarily focused on understanding the error bounds and the bias-variance trade-off (Hastie et al., 2009) to understand and improve the models' generalizability. Later, techniques like dropout (Srivastava et al., 2014) were introduced to improve neural models' generalization. Recently, studies have examined the generalizability of various neural network architectures using specialized generalization evaluation tasks (Hupkes et al., 2020; Ontanon et al., 2022; Csordás et al., 2021). Additionally, numerous datasets such as SCAN (Lake and Baroni, 2018), CFQ (Keysers et al., 2020), and COGS (Kim and Linzen, 2020) have been developed to assess compositional generalization capabilities. Diverse strategies such as data augmentation (Andreas, 2020; Shaw et al., 2021), innovative architectural designs (Korrel et al., 2019; Gao et al., 2020), and neuro-symbolic methods (Mao et al., 2019), have been proposed to enhance these capabilities. Consequently, these advances in text-based generalization have inspired research in multimodal compositional generalization, with developments including complex benchmarks like gSCAN (Ruis et al., 2020) and ReaSCAN (Wu et al., 2021), and advanced architectures applied to multimodal grounding (Kuo et al., 2021; Jiang and Bansal, 2021; Qiu et al., 2021a; Sikarwar et al., 2022; Shaw et al., 2021).

Furthermore, recent research highlights the significant role of syntactic information in enhancing neural models' compositional generalization capability. Kuo et al. (2021) suggested aligning the compositional structure of networks with the problem domain, resulting in a dynamic compositional neural network. Moreover, Shaw et al. (2021) and Qiu et al. (2022) recommended grammar induction-based data augmentation techniques to improve compositional generalization. Unlike our work that focuses on input command structure, Kim et al. (2021b) introduced the concept of using parse tree node annotations in the target sequence of sequence-to-sequence tasks for enhancing compositional generalization. Meanwhile, Kim et al. (2021a) incorporated parse tree nodes into the ETC (Ainslie et al., 2020) model. They employed attention masking specific for ETC to symbolize the relations of tokens and aid this model in a simplified classification task based on the CFQ dataset.

We are inspired by previous research (Kim et al., 2021a) that employs a similar technique with manually extracted parses for compositional generalization on the single text modality. However, our model utilizes off-the-shelf parsers instead of accurate manually generated parse trees, and it is generally applicable independent of the underlying

models.

## 3 Problem Setting

Various studies on compositional generalization have presented a range of tasks and problem settings (Lake and Baroni, 2018; Keysers et al., 2020; Kim and Linzen, 2020; Wu et al., 2021; Ruis et al., 2020). These datasets are comprised of a training set and several test sets. To ensure rigorous evaluation, the test sets have been deliberately structured to differ from the training set in a way that requires the compositional capability to succeed. Our paper focuses on grounding natural language instructions in the visual modality, where we map words to specific objects or actions in a multimodal environment that provides a framework to evaluate an intelligent agent's compositional structures and spatial reasoning capabilities.

We use the most recent multimodal compositional generalization benchmarks to assess our models comprehensively. In these benchmarks, an agent receives natural language instruction to carry out an action or navigate specific environments. These datasets are inherently synthetic, and they have been carefully crafted to guarantee that the test sets are systematically different from the training sets. By placing commands within a spatial context, these benchmarks bridge the gap between abstract cognitive understanding and practical action execution. Consequently, they stand as both a scholarly tool for studying compositional generalization and a valuable resource for fields like robotics that require comprehension of spatially anchored commands.

Among these benchmarks, our primary focus is ReaSCAN, owing to its heightened complexity and recent introduction to the academic community. An example of this dataset, depicted in Figure 1, consists of three main components: The initial state of the world, the provided input command, and the corresponding target command. Tasked with this information, the agent aims to infer the target command by leveraging both the information from the input command and the initial state. Structurally, the world's representation in ReaSCAN is formulated as a 6×6×17 matrix. Each matrix cell comprises a 17-dimensional vector encapsulating information pertaining to an object's attributes—namely, color, shape, and size—along with indicators of the agent's positioning and orientation. The evaluation metric for this dataset is the percentage of exact matches of the predicted action sequence. The ReaSCAN dataset includes one random test split that mirrors the training's component and compound distribution, in addition to seven compositional generalization test splits. Each of these splits is designed to probe a specific facet of a model's grounding generalization capability, as detailed in Table 1. Category A test splits delve into novel attribute compositions at both the command and object levels, drawing inspiration from gSCAN. Category B test splits assess the model's ability to generalize to unprecedented co-occurrences of concepts and spatial relations. Meanwhile, Category C probes the model's capacity to extrapolate from simple command structures to more intricate structures with higher levels of reasoning (Wu et al., 2021). To illustrate the A1 split, all examples with commands containing variations of "yellow square" (such as "small yellow square" or "big yellow square") are excluded from the training data. This prevents models from associating targets with that phrase. However, the training set does include examples like "yellow cylinder" and "blue square." As a result, during testing, models are expected to accurately interpret the "yellow square" even without prior exposure to the actual composition.

| Split | Held-out Examples |
|---|---|
| Random | Random. |
| A1 | *yellow square* referred with color & shape. |
| A2 | *red square* referred in the command. |
| A3 | *small cylinder* referred with size and shape |
| B1 | co-occur of *small red circle* and *big blue square*. |
| B2 | co-occur of *same size as* and *inside of* relations. |
| C1 | Additional conjunction clause depth added to *2-relative-clause commands*. |
| C2 | *2-relative-clause* command with *that is* instead of *and*. |

Table 1: ReaSCAN dataset test splits.

## 4 Proposed Method

To address the challenge at hand, we implemented a multimodal transformer, as illustrated in Figure 2. In this model, input commands are tokenized and then supplemented with positional encoding before passing to the transformer. Concurrently, the visual environment is segmented into 36 distinct

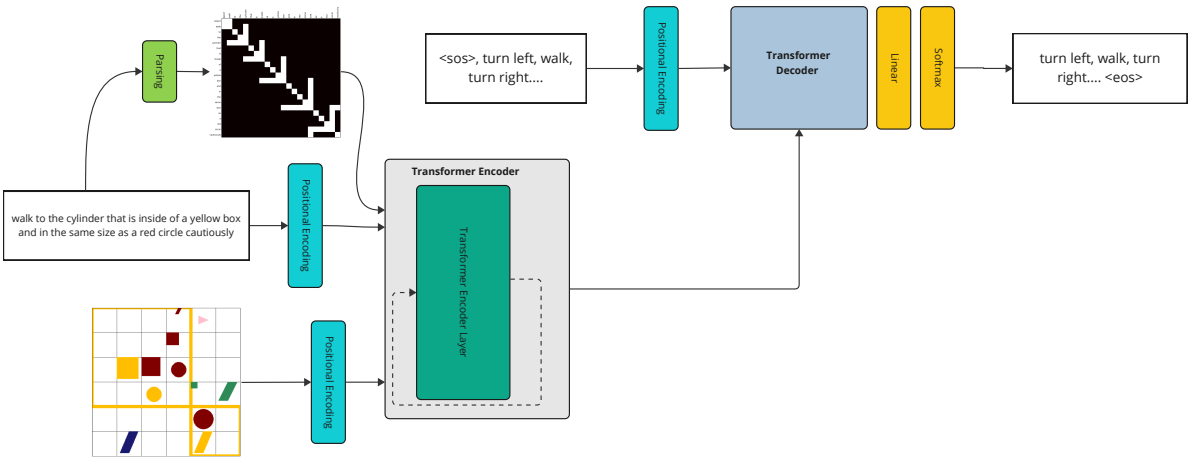



Figure 2: Overall architecture of the proposed model.

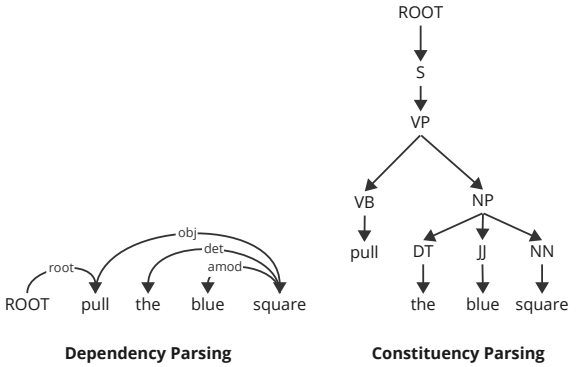



Figure 3: Examples of parse trees.

cells, each serving as a visual token. After passing the visual token to a linear layer, these tokens receive positional encoding and are passed to the transformer.

We've employed a generic parser to seamlessly embed the structure of the textual modality into our model, thereby shaping attention masks for the encoder's textual self-attention. Prioritizing efficiency, parsing each input command is conducted during a preprocessing phase.

Our transformer is based on the GroCoT model (Sikarwar et al., 2022). Each encoder layer employs a cross-attention mechanism between modalities, followed by modality-specific self-attention. Our computed input command masks are utilized in the self-attention modules of the textual modality. Remarkably, encoder layer weights are consistently shared across all layers.

In the end, we concatenate the encoded result of each modality and pass it to the transformer's auto-regressive decoder to generate the action sequence corresponding to the input command given the environment.

### 4.1 Syntax-guided attention

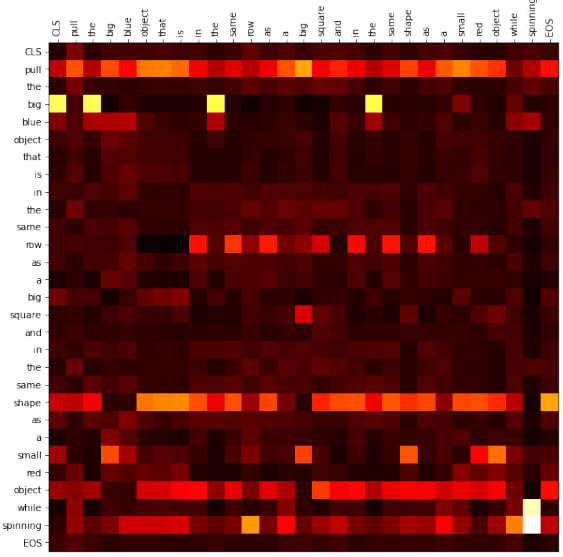
One main component of our proposed model is exploiting the syntactic structure of the command. For this aim, we investigate using both dependency and constituency parsing. Dependency and constituency trees can be used to analyze the grammatical structure of sentences. Dependency trees focus on the grammatical relationships between individual words, where each word except the root depends on another, and the edges of the tree signify these dependencies. However, constituency trees emphasize the hierarchical organization of words into larger syntactic units or constituents, with internal nodes representing these groupings and leaves representing individual words. While dependency trees are more concerned with identifying grammatical roles and relationships between words, constituency trees aim to show how words group together into larger syntactic units, often carrying syntactic labels like NP (noun phrase) or VP (verb phrase) (Foscarin et al., 2023; Hearne et al., 2008). Examples of these parse trees are shown in Figure 3.

**Syntax-guided attention masking.** We use the syntactic information to guide the self-attention module of transformer encoder layers as depicted in Figures 2 and 4b. We force each token to only attend to the tokens connected in the syntax tree. In this way, we avoid faulty attention patterns and overfitting irrelevant parts of the sentence. In addition, by imposing the structure with a parse tree, our model can capture the nesting structure of the command's meaning and the relationships between its components. By making the structural information explicit, our model can potentially extrapolate

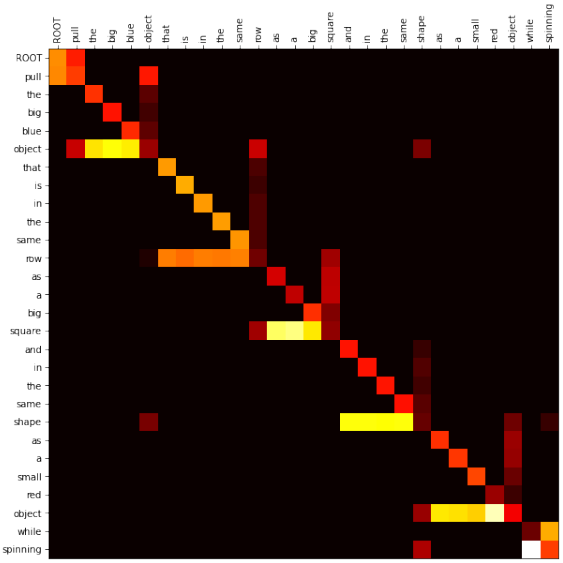
the meaning of novel combinations and nesting linguistic structures encompassing higher reasoning depth.

### 4.2 Weight Sharing

Parameter sharing is a strategic approach where identical learned parameters are applied across various positions or layers within a model. This technique enables the reuse of the same encoder unit at each phase of the transformer encoder (Dehghani et al., 2019). Such an approach not only streamlines the model but also nurtures the acquisition of more robust and adaptable representations of the input (Ontanon et al., 2022). The findings of Kim et al. (2021a) demonstrate that a transformer employing attention masking requires extended training epochs for convergence, potentially due to masking-induced backpropagation constraints. In light of this, we hypothesize that introducing weight sharing might counterbalance this challenge. Weight sharing reduces the model's complexity by decreasing the number of parameters, which could lead to faster convergence. This method acts as a form of regularization, stabilizing training and facilitating smoother optimization landscapes. In addition, Ontanon et al. (2022) show that a transformer with shared weights across its encoder layers is arguably endowed with a more suitable inductive bias that allows the model to learn the primitive concepts. We hypothesize this will positively affect learning spatial relations or object-property relations, which are frequently used in our model's input. Motivated by these advantages, we incorporated this weight sharing technique into our transformer model to evaluate its efficacy in a multimodal setting. Beyond the enhanced generalizability, weight sharing serves as a computational benefit by reducing the number of learnable parameters during the training phase.




(a) Self-attention w/o masking




(b) Self-Attention w/ masking

Figure 4: Self-Attention example from the A2 test set of ReaSCAN dataset. Figures (a) and (b) depict the averaged self-attention map from our models' over all encoder layers and heads. Rows and columns correspond to text tokens. Brighter attention cells indicate higher attention weights

## 5 Experiments

**Implementation Details.** Our model architecture is founded on the GroCoT framework as detailed by Sikarwar et al. (2022) and is implemented using the PyTorch machine learning library (Paszke et al., 2019). Also, we employed the pre-trained stanza toolkit (Qi et al., 2020) for constituency and dependency parsing. We used 48 GB A6000 GPUs accompanied by 756GB RAM. On average, each experiment took about 52 hours to train the models from scratch, with the Adam optimizer (Kingma and Ba, 2017) parameter updates throughout the training regimen. To ensure a rigorous evaluation, we used the same specialized compositional validation set as Sikarwar et al. (2022), drawing 500 samples from each compositional division of the primary dataset. Model proficiency was assessed against this validation set, with the highest-performing model designated as our optimal choice. Our results are presented as an average derived from three independent runs, each initial-

ized with a random seed. We ran the models for the ReaSCAN benchmark for 120 epochs, and the models for the gSCAN and GSRR benchmarks for 100 epochs. Hyperparameters used for the experiments of each dataset are shown in Appendix A. The code and models proposed in this work are all available in GitHub[1].

**Datasets.** We used gSCAN (Ruis et al., 2020), GSRR (Qiu et al., 2021b), ReaSCAN (Wu et al., 2021) benchmarks for evaluation. The Grounded SCAN (gSCAN) dataset is a benchmark tailored for examining compositional generalization in machine learning models by translating natural language commands into actions in a grid-world scenario. Its unique splits ensure models move beyond rote memorization to deep compositional understanding of concepts. The Grounded Systematic Relation Reasoning (GSRR) dataset extends gSCAN by aligning natural language instructions intricately with visual elements, emphasizing spatial relationships and object references. ReaSCAN, a further development, brings the challenges of real-world reasoning into this environment by introducing more challenging tasks and concept relations. Together, these datasets offer a high-complexity framework for assessing the compositional and relational understanding of machine learning models in visual environments. A detailed explanation of both the gSCAN and Grounded Systematic Relation Reasoning datasets can be found in Appendix B.

**Baselines.** We embarked on a series of experiments designed to evaluate our model's effectiveness compared to the most recent state-of-the-art models on the mentioned multimodal compositional generalization datasets. We include the following baselines. **(a)** Ruis et al. (2020) (Multimodal LSTM) is a fusion of sequence-to-sequence (seq2seq) architecture with a visual encoder, employing a recurrent 'command encoder' to process the instructions. **(b)** Gao et al. (2020) (GCN-LSTM) integrates a Graph Convolutional Neural (GCN) network with a multimodal LSTM. The command encoding is achieved via a BiLSTM equipped with multi-step textual attention, while the world is encoded through a GCN layer. **(c)** Qiu et al. (2021b) (Multimodal Transformer) is a multimodal transformer equipped with cross-attention for multimodal compositional generalization. **(d)** Sikarwar et al. (2022) (GroCoT) is another transformer-based model that incorporates interleaved self-attention into the multimodal transformer with cross-attention.

**Results.** We comprehensively evaluated our approach across all the previously mentioned benchmarks compared to the baselines. Alongside the accuracy and efficacy metrics, we also provide insights into the computational overhead associated with our method. Furthermore, a qualitative analysis is presented, delving deeper into our approach's performance nuances and strengths.

The benchmark results, presented in Tables 2, 3, and 4, demonstrate our model's superior performance over all reported models, with a notable 3% improvement on the average of ReaSCAN benchmark splits. This substantiates our hypothesis that incorporating syntactic parsing significantly boosts the model's generalization derived from grounded compositional training data. Moreover, dependency parsing consistently outperformed constituency or marked a very similar performance across multiple benchmarks, including GSRR and gSCAN. Our model displayed improvements across nearly all ReaSCAN splits except for C2. As per Sikarwar et al. (2022), the C2 split is "unfair," lacking the required information in training data for comprehensive model training. Even including syntactic information could not improve the model's performance on this split and even caused a decrease in the performance. Our methodology also showcased its merit in the object property test cases (A1-3), effectively constraining attention to words pertinent to target object descriptors. For instance, as shown in Figure 4, the attention weights from the properties to the corresponding objects are high.

Notably, our model exhibited considerable strides in the C1 split, indicative of the value added by syntactic information. For a more reliable comparison, we applied a t-test to our C1 test split results. Using a significance level ($\alpha$) of 0.05, this statistical analysis provided further validation for the observed enhancements in our model's performance, particularly within the context of the C1 test split. Furthermore, our model exhibits enhanced performance on the GSRR dataset. As illustrated in Table 4, both variants of our model demonstrate improvements in the II split. It is worth noting that the II split shares the same challenge as the A2 split from the ReaSCAN dataset but in a less complex

[1] https://github.com/HLR/Syntax-Guided-Transformers

| Model | A1 | A2 | A3 | B1 | B2 | C1 | C2 | Avg |
|---|---|---|---|---|---|---|---|---|
| LSTM* | 50.4 | 14.7 | 50.9 | 52.2 | 39.4 | 49.7 | 25.7 | 40.40 |
| GCN-LSTM | 92.3 | 42.1 | 87.5 | 69.7 | 52.8 | 57.0 | 22.1 | 60.50 |
| Transformer* | 96.7 | 58.9 | 93.3 | 79.8 | 59.3 | 75.9 | 25.5 | 69.90 |
| GroCoT | 99.6 | 93.1 | 98.9 | 93.9 | 86.0 | 76.3 | **27.3** | 82.2 |
| Constituency† | **99.75**±0.11 | 96.70±1.40 | **99.68**±0.10 | 95.19±1.17 | 88.37±1.50 | 69.07±0.60 | 27.00±0.54 | 82.25±0.63 |
| Dependency† | 99.65±0.9 | **97.37**±0.48 | 99.62±0.07 | **95.46**±2.01 | **90.15**±3.88 | **92.55**±1.51 | 21.77±5.25 | **85.22**±0.87 |

Table 2: The result of our proposed model on the ReaSCAN dataset test splits. The results are an average of three runs. † denotes the models with masking. Models marked with * refer to the multimodal version of their implementation.

| Model | A | B | C | E | F | H | Comp. Avg |
|---|---|---|---|---|---|---|---|
| LSTM* | 97.7 | 54.9 | 23.5 | 35.0 | 92.5 | 22.7 | 32.7 |
| GCN-LSTM | 98.6 | 99.1 | 80.3 | 87.3 | 99.3 | **33.6** | - |
| Transformer* | 99.9 | 99.9 | 99.3 | 99.0 | 99.9 | 22.2 | 60.0 |
| GroCoT | 99.9 | 99.9 | 99.9 | 99.8 | 99.9 | 22.9 | 60.4 |
| Constituency† | **99.95**±0.07 | **99.92**±0.06 | **99.88**±0.11 | 99.88±0.09 | **100.00**±0.00 | 22.84±0.93 | 60.36±0.11 |
| Dependency† | 99.92±0.09 | 99.85±0.18 | 99.86±0.11 | **99.96**±0.06 | 99.89±0.16 | 23.89±1.54 | **60.49**±0.20 |

Table 3: The result of our proposed model on the gSCAN dataset test splits. The results are an average of three runs. We did not report the results on D and G splits since we achieved 0.00±0.00 % performance, But take them into account in the averaged result. † denotes the models with masking. Models marked with * refer to the multimodal version of their implementation.

environment.

While our proposed techniques effectively address splits A, B, C, E, and F, mirroring the successes of previous works such as (Sikarwar et al., 2022) and (Qiu et al., 2021b), they struggle with challenges presented by specific gSCAN compositionality splits, notably D, G, and H. These particular splits are designed to assess the model's capacity for systematic generalization when novel patterns should occur on the output sequence rather than in grounding the input instruction (Sikarwar et al., 2022), a facet that is not expected to be captured by our proposed model.

### 5.1 Ablation

For a granular understanding of the contributions from each alteration to the baseline model, we undertook an ablation study. This involved the sequential removal of each modification to measure its individual impact. As depicted in Table 5, while individual modifications did not significantly change the baseline, their collective integration enhanced the model's generalization. Remarkably, eliminating dependency parsing or weight sharing resulted in a noticeable performance dip. The improvement upon integration posits that weight sharing can potentially offset the masking prolonged convergence challenge by reducing parameter count, thereby mitigating the convergence issues.

### 5.2 Qualitative Analysis

In our previous discussions, we highlighted the significance of integrating dependency parsing as a fundamental approach to understanding the complex structures inherent in sentences. This integration is not a mere enhancement; it critically enriches the model's grounding capabilities, offering a more robust bridge between raw textual sequences and their semantic structure.

To provide empirical evidence of our technique for guiding attention, we conducted an analysis of the cross-attention module. We aimed to compare its behavior before and after applying attention masking. The results, presented in Figure 5, indicate a clear trend: in 86% of validation samples, the cross-attention module exhibits a pronounced focus on the target object.

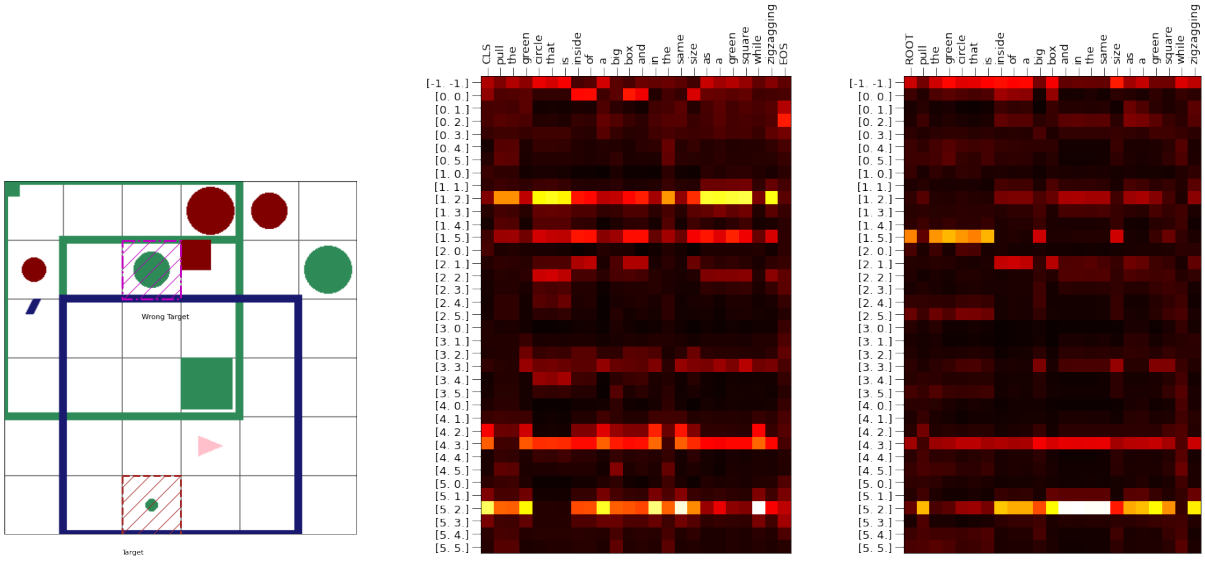
Figures 5b and 5c elucidate the impact of self-attention masking on these weights. After using attention masking (see Figure 5b), the attention distribution becomes notably sparser; instead of individual words attending in isolation to every potentially relevant cell, they now form cohesive compositional expressions, each attending to the

| Model | I | II | III | IV | V | VI | Comp. Avg |
|---|---|---|---|---|---|---|---|
| LSTM* | 86.5 | 40.1 | 86.1 | 5.5 | 81.4 | 81.8 | 58.9 |
| Transformer* | 94.7 | 64.4 | 94.9 | 49.6 | 59.3 | 49.5 | 63.5 |
| GroCoT | 99.9 | 98.6 | **99.9** | 99.7 | **99.5** | 96.5 | 98.8 |
| Constituency† | 99.85±0.00 | 99.90±0.03 | 99.16±0.26 | 99.88±0.03 | 96.73±2.16 | **97.85**±0.46 | 98.58±0.39 |
| Dependency† | **99.91**±0.02 | **99.93**±0.01 | 99.41±0.28 | **99.96**±0.01 | 99.03±0.23 | 97.38±0.63 | **99.07**±0.16 |

Table 4: The result of our proposed model on the GSRR dataset test splits. The results are an average of three runs. † denotes the models with masking. Models marked with * refer to the multimodal version of their implementation.

| W/S | Mask | A1 | A2 | A3 | B1 | B2 | C1 | C2 | Avg |
|---|---|---|---|---|---|---|---|---|---|
| - | - | 99.29±0.27 | 91.82±6.50 | 98.49±1.17 | 93.50±0.85 | 83.15±1.41 | 75.85±1.35 | **25.03**±6.82 | 81.02±0.22 |
| ✓ | - | 99.68±0.22 | 97.09±1.72 | 99.64±0.20 | 94.86±0.77 | 81.49±4.27 | 66.30±6.65 | 21.66±1.83 | 80.10±1.08 |
| - | Dep. | 98.09±0.27 | 85.21±6.85 | 97.35±0.75 | 93.61±2.75 | 90.62±1.59 | 75.27±1.77 | 21.91±1.63 | 80.29±1.43 |
| ✓ | Dep. | **99.65**±0.9 | **97.37**±0.48 | **99.62**±0.07 | **95.46**±2.01 | **90.15**±3.88 | **92.55**±1.51 | 21.77±5.25 | **85.22**±0.87 |

Table 5: The ablation study result of our modifications on ReaSCAN dataset test splits. Results are reported on an average of three runs. We evaluate every combination of components from our best model. W/S stands for weight sharing, and the ✓shows the presence of the module. *Dep* in this table refers to the Dependency masking. We evaluate the model with or without dependency masking in the masking part.




(a) Environment (b) Cross-attention w/o masking (c) Cross-Attention w/ masking

Figure 5: Cross-Attention from Text-to-Image. In Figure (a), the purple zone indicates the model's incorrect object selection, while the red zone highlights the accurate choice. Figures (b) and (c) depict the averaged cross-attention map from our models over encoder layers and attention heads. The rows represent environment cells (the first element shows the row, and the second shows the column index, both starting from 0), and the columns correspond to text tokens. Brighter attention cells signify elevated attention weights.

corresponding cells as a whole. For instance, in Figure 5c, "and in the same," phrase's tokens attend to cells (1,2), (4,3), and (5,2) together with greater attention on the target object in contrast to their attention pattern without masking.

### 5.3 Efficiency Analysis

In the realm of modern model design, the challenge lies in amplifying capabilities while managing computational overhead. Our methodology adeptly navigates this balance. A cornerstone of our model's efficiency is the strategic adoption of weight sharing within the transformer encoder. By reusing weights across different components, we

| Model | #Parameters |
|---|---|
| Multimodal LSTM | 74K |
| Multimodal Transformer | 3M |
| GroCoT | 4.6M |
| Dependency† (ours) | 1.9M |

Table 6: Comparing model parameters: our model vs. current state-of-the-art models. Dependency† refers to the model with dependency parsing for attention masking.

significantly reduce the parameter space. This not only streamlines memory utilization and accelerates training but also acts as an implicit regularizer, bolstering the model's generalization capabilities and reducing overfitting. Further enhancing this is our implementation of attention masking, which refines computational efficiency. By enabling the model to selectively bypass attention to certain tokens, we can optimize the model to avoid redundant computational processes, ensuring optimal resource allocation and superior performance.

As illustrated in Table 6, our model stands out in terms of efficiency. Despite having fewer parameters (1.9M) than the models by Qiu et al. (2021a) and Sikarwar et al. (2022), which have 3M and 4.6M parameters respectively, our model consistently outperforms them across all benchmarks.

## 6 Conclusion

Our research demonstrated that exploiting the syntactic structure of compositional and complex linguistic and spatial expressions improved the grounding ability of the instruction-follower agent in multimodal environments. Our results indicated improvements compared to the previous state-of-the-art models. In particular, we show that our proposed model is effective for generalization on tasks and test splits that require generalization over unobserved reasoning depths, such as the C1 split in the ReaSCAN dataset. By utilizing the syntactic-guided attention masking along with the weight sharing, we achieved not only more accurate but also more parameter-efficient models for grounded compositional generalization.

## Limitations

Despite the promising results achieved in our study, several limitations warrant consideration:

**Synthetic Data:** Our experiments predominantly rely on synthetic datasets. While these datasets provide a controlled environment for assessing model performance, they might not capture the complexities and nuances of real-world data. Evaluating the models on real-world datasets is crucial to ensure their practical applicability.

**Error Propagation from the Parser:** The model's performance is intrinsically tied to the accuracy of the pre-trained parsers we utilized. Errors or inaccuracies in parsing can lead to suboptimal model outputs. Additionally, our synthetic data, being unambiguous, might not reveal the full extent of potential parser-related issues.

**Computational Constraints:** Due to computational limitations, the hyperparameter search might not have been exhaustive. A more comprehensive exploration might yield better model configurations.

## Acknowledgement

This project is supported by the National Science Foundation (NSF) CAREER award 2028626 and partially supported by the Office of Naval Research (ONR) grant N00014-20-1-2005. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation nor the Office of Naval Research. We thank all reviewers for their thoughtful comments and suggestions.

## References

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284, Online. Association for Computational Linguistics.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton, The Hague.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. 2021. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical*

*Methods in Natural Language Processing*, pages 619–634, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2019. Universal transformers.

Francesco Foscarin, Daniel Harasim, and Gerhard Widmer. 2023. Predicting music hierarchies with a graph-based neural decoder.

Tong Gao, Qi Huang, and Raymond Mooney. 2020. Systematic generalization on gSCAN with language conditioned embedding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 491–503, Suzhou, China. Association for Computational Linguistics.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. Springer New York.

Mary Hearne, Sylwia Ozdowska, and John Tinsley. 2008. Comparing constituency and dependency representations for SMT phrase-extraction. In *Actes de la 15ème conférence sur le Traitement Automatique des Langues Naturelles. Articles courts, TALN 2008, Avignon, France, June 2008*, pages 131–140. ATALA.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? (extended abstract). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 5065–5069. International Joint Conferences on Artificial Intelligence Organization. Journal track.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, and Zhijing Jin. 2023. State-of-the-art generalisation research in nlp: A taxonomy and review.

Yichen Jiang and Mohit Bansal. 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*, pages 6253–6265, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Juyong Kim, Pradeep Ravikumar, Joshua Ainslie, and Santiago Ontañón. 2021a. Improving compositional generalization in classification tasks via structure annotations.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. *CoRR*, abs/2010.05465.

Segwang Kim, Joonyoung Kim, and Kyomin Jung. 2021b. Compositional generalization via parsing tree annotation. *IEEE Access*, 9:24326–24333.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Kris Korrel, Dieuwke Hupkes, Verna Dankers, and Elia Bruni. 2019. Transcoding compositionally: Using attention to find more generalizable solutions. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 1–11, Florence, Italy. Association for Computational Linguistics.

Yen-Ling Kuo, Boris Katz, and Andrei Barbu. 2021. Compositional networks enable systematic generalization for grounded language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 216–226, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2019. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *International Conference on Learning Representations*.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

Santiago Ontanon, Joshua Ainslie, Zachary Fisher, and Vaclav Cvicek. 2022. Making transformers solve compositional tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3591–3607, Dublin, Ireland. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Linlu Qiu, Hexiang Hu, Bowen Zhang, Peter Shaw, and Fei Sha. 2021a. Systematic generalization on gSCAN: What is nearly solved and what is next? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2180–2188, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Yao Qiu, Jinchao Zhang, and Jie Zhou. 2021b. Improving gradient-based adversarial training for text classification by contrastive learning and auto-encoder. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1698–1707, Online. Association for Computational Linguistics.

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M. Lake. 2020. A benchmark for systematic generalization in grounded language understanding.

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

Ankur Sikarwar, Arkil Patel, and Navin Goyal. 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*, pages 648–669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2020. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, pages 3428–3448. Association for Computational Linguistics (ACL).

Zhengxuan Wu, Elisa Kreiss, Desmond C. Ong, and Christopher Potts. 2021. ReaSCAN: Compositional reasoning in language grounding. *NeurIPS 2021 Datasets and Benchmarks Track*.

# A Hyperparameters

Here, we present the hyperparameters used in the models for every benchmark in Table 7.

# B Datasets Description

## B.1 Grounded SCAN dataset

The Grounded SCAN (gSCAN) dataset is a pivotal benchmark for assessing compositional generalization in machine learning models. Evolving from the foundational SCAN (Lake and Baroni, 2018) dataset, gSCAN is designed to evaluate a model's proficiency in translating command sequences into actions within a grid world environment, emphasizing on compositional challenges.

This benchmark offers systematic test splits that rigorously examine a model's capability to generalize beyond its training data. These compositional splits include:

- **A (Random)**: Random data with a similar distribution to the training data.
- **B (Color-Shape)**: Novel composition of object properties in the testing. Yellow squares are referred to by color and shape.
- **C (Color Only)**: Red squares as target.
- **D (Novel Direction)**: Challenges a model's spatial comprehension, with targets set in unfamiliar directions, the southwest.
- **E (Novel Contextual References)**: Evaluates a model's understanding of relative sizes, with commands pointing to circles of size 2 described as "small."
- **F (Novel Composition of Actions and Arguments)**: Probes a model's grasp of object classes and their nuances, exemplified by squares of size 3 necessitating two pushes.
- **G (Adverb)**: Commands carrying the adverb "cautiously" test how well the model interprets action modifiers after seeing limited training samples (k=1).

| Hyperparameter | gSCAN | GSRR | ReaSCAN |
|---|---|---|---|
| Number of Vision Self-Attention Layers | 3 | 3 | 6 |
| Number of Text Self-Attention Layers | 3 | 3 | 6 |
| Number of Cross-Attention Layers | 3 | 3 | 6 |
| Number of Decoder Layers | 3 | 3 | 6 |
| Embedding Size | 128 | 128 | 128 |
| Hidden Layer Size | 256 | 256 | 256 |
| Number of Attention Heads | 8 | 8 | 8 |
| Learning Rate | $5 \times 10^{-5}$ | $1 \times 10^{-5}$ | $1 \times 10^{-5}$ |
| Batch Size | 64 | 64 | 32 |
| Dropout | 0.1 | 0.1 | 0.1 |
| Number of Epochs | 100 | 100 | 120 |

Table 7: Hypterparameters used in the experiments.

- **H (Adverb-Verb Combination)**: Generalizes to commands pairing actions and their modifiers, like "while spinning" combined with "pull."

The compositional test splits of the gSCAN dataset ensure that models are not indulging in learning statistical shortcuts but are genuinely mastering compositional reasoning. In gSCAN, every command is mapped to an action sequence for an agent in the grid world, whether moving to a particular spot or interacting with a distinct described object.

### B.2 Grounded Systematic Relation Reasoning dataset (GSRR)

The Grounded Systematic Relation Reasoning (GSRR) dataset, introduced by (Qiu et al., 2021b), extends the gSCAN benchmark. Their initial analyses of the gSCAN dataset indicated its efficacy; the authors observed that several remaining challenges might not be primarily tied to visual grounding. In light of this, they proposed the GSRR task, characterized by an elevated complexity in aligning natural language instructions with the visual environment.

In this dataset, language expressions specifically delineate target objects and explicitly describe their relationships with a secondary referenced object. They incorporate two types of relations into our dataset: immediate adjacency ("next to") and cardinal directions such as "north" and "west." In addition, they put visual distractors objects within the environment to emphasize the critical role of spatial relations in identifying the target objects.

The dataset is systematically divided into various splits to ensure a comprehensive assessment:

- **I (Random)**: Similar distribution as the training.
- **II (Visual)**: Commands centering on "red squares" either as targets or references.
- **III (Relation)**: Complex instructions involving combinations like "green squares" and "blue circles."
- **IV (Referent)** Emphasizing "yellow squares" as primary targets.
- **V (Relative Position 1)**: Commands where targets are situated to the "north" of their reference points.
- **VI (Relative Position 2)**: Instructions where targets are located "southwest" relative to their references.

## C Evaluation Card

Here, we present the evaluation card of our compositional generalization experiments based on (Hupkes et al., 2023) taxonomy.

| **Motivation** | | | |
|---|---|---|---|
| *Practical* | *Cognitive* | *Intrinsic* | *Fairness* |
| ☐ | | ☐ | |
| **Generalisation type** | | | |
| *Compositional* | *Structural* | *Cross Task* | *Cross Language* | *Cross Domain* | *Robustness* |
| ☐ | ☐ | | | | |
| **Shift type** | | | |
| *Covariate* | *Label* | *Full* | *Assumed* |
| | | ☐ | |
| **Shift source** | | | |
| *Naturally occuring* | *Partitioned natural* | *Generated shift* | *Fully generated* |
| | | | ☐ |
| **Shift locus** | | | |
| *Train–test* | *Finetune train–test* | *Pretrain–train* | *Pretrain–test* |
| ☐ | | | |