Improving Systematic Generalization Through Modularity and Augmentation

Laura Ruis (laura.ruis.21@ucl.ac.uk)
University College London

Brenden M. Lake (brenden@nyu.edu)
New York University; Facebook AI Research

Abstract

Systematic generalization is the ability to combine known parts into novel meaning; an important aspect of efficient human learning, but a weakness of neural network learning. In this work, we investigate how two well-known modeling principles—modularity and data augmentation—affect systematic generalization of neural networks in grounded language learning. We analyze how large the vocabulary needs to be to achieve systematic generalization and how similar the augmented data needs to be to the problem at hand. Our findings show that even in the controlled setting of a synthetic benchmark, achieving systematic generalization remains very difficult. After training on an augmented dataset with almost *forty times* more adverbs than the original problem, a non-modular baseline is not able to systematically generalize to a novel combination of a known verb and adverb. When separating the task into cognitive processes like perception and navigation, a modular neural network is able to utilize the augmented data and generalize more systematically, achieving 70% and 40% exact match increase over state-of-the-art on two gSCAN tests that have not previously been improved. We hope that this work gives insight into the drivers of systematic generalization, and what we still need to improve for neural networks to learn more like humans do.

Keywords: Modularity; Systematic Generalization; Data Augmentation

Introduction

Humans are efficient learners. Once someone has seen a single example of a wampimuk, they know how to recognize a small wampimuk. Once someone learns how to walk cautiously, they know how to cycle cautiously, even though cycling requires a novel sequence of low-level actions. Our aptitude for this type of generalization is a consequence of the fact that word meanings like "small" and "cautiously" compose systematically (Chomsky, 1957; Montague, 1970) — a property of language called systematic compositionality.

A discrepancy between humans' ability to interpret novel compositions and that of neural network models has long been discussed (Fodor & Pylyshyn, 1988; Marcus, 1998; Fodor & Lepore, 2002; Marcus, 2003; Calvo & Symons, 2014) and this issue has been revitalized in recent years (Gershman & Tenenbaum, 2015; Lake & Baroni, 2018; Bastings, Baroni, Weston, Cho, & Kiela, 2018; Loula, Baroni, & Lake, 2018; Bahdanau et al., 2019). While conventional models assign low likelihood to unseen combinations of familiar tokens, there has been a recent surge of interest in developing models that generalize more systematically (Russin, Jo, O'Reilly, & Bengio, 2019; Lake, 2019; Andreas, 2020; Nye, Solar-Lezama, Tenenbaum, & Lake, 2020; Gordon, Lopez-Paz, Baroni, & Bouchacourt, 2020; Bogin, Subramanian, Gardner, & Berant, 2021). Despite progress, we are still very far from neurally-grounded models that provide a fully satisfying account of systematic compositionality, and certain types of complex composition remain especially elusive.

The methods that systematically handle prototypical examples of compositionality like verb-object binding do not address other less studied cases like adverb-verb composition. Similarly to the "cycle cautiously"-example above, this type of generalization requires making non-local changes to the action sequence, transforming it entirely. The example reflects a more general property of adverbs; combining a verb with an adverb in a sentence can substantially change the actions required to perform said verb. We study this phenomenon using the recently proposed gSCAN benchmark for evaluating models for systematic reasoning in a controlled environment (L. Ruis, Andreas, Baroni, Bouchacourt, & Lake, 2020). The grounded nature of this benchmark allows it to evaluate new dimensions of systematicity that have not been captured by previous work, like the additional complexity of adverb-verb compositionality. Grounding meaning in a world state requires models to do something that is more like real language understanding; the correct action sequence for a language command changes based on the world state.[1] Several recent works have proposed methods to improve performance on the tests (Heinze-Deml & Bouchacourt, 2020; Gao, Huang, & Mooney, 2020; Kuo, Katz, & Barbu, 2021; Nye, Tessler, Tenenbaum, & Lake, 2021; Qiu, Hu, Zhang, Shaw, & Sha, 2021; Jiang & Bansal, 2021), yet there has been little progress on the types of unseen adverb combinations discussed above.

Here, we study two key modeling principles, modularity and structured data augmentation, and their effect on systematic generalization in grounded language learning. The notion that modular architectures generalize better is a longstanding principle in cognitive science (Fodor, 1983) and AI (Poole & Mackworth, 2017), just as programmers know that modular systems are more reusable and robust (Meyer, 1997). Densely-connected neural networks may be especially prone to the pitfalls of non-modular systems, as spurious correlations in the input can affect the whole system in unpredictable ways. There

[1] Of course, the meaning of a word is not equivalent to the ability to perform it – you can know the meaning of "ski jumping" without actually being able to do it. In this domain, we assume that actions are simple and executing them is a noiseless process.

is ample evidence that modularity can improve systematic generalization in neural network models, (Andreas, Rohrbach, Darrell, & Klein, 2016; Bahdanau et al., 2019; Purushwalkam, Nickel, Gupta, & Ranzato, 2019; D'Amario, Sasaki, & Boix, 2021) and here we apply modularity to the gSCAN challenge, decoupling navigation, perception, and reasoning.

Data augmentation is a widely used technique in machine learning to increase the diversity of training examples without explicitly collecting new data (for surveys in computer vision and natural language processing, see Shorten and Khoshgoftaar (2019) and Feng et al. (2021), respectively). However, applying data augmentation techniques to get examples for adverb-verb composition is difficult. The transformative effect an adverb has on the output sequence and the grounded nature of the benchmark makes the class of augmentation techniques called interpolation-based, like those proposed by Andreas (2020) and Kagitha (2020), inapplicable. Instead we take a structured augmentation approach and infer a set of rules that characterize how adverbs operate in our domain, subsequently using these rules to generate new data. We evaluate how much and what kind of experience is needed to generalize to examples exhibiting the type of compositionality illustrated by the "cycle cautiously"-example.

We propose a modular approach to systematic generalization and design a structured data augmentation technique to generate experience for the modules. From extensive experiments with the proposed setup we deduce three main findings. Firstly, simply providing a neural network with more data might not be sufficient to achieve systematic generalization; even after adding as much as 150 extra adverbs to the four original adverbs in the gSCAN dataset (arguably enough given the simple, controlled environment of the tests) the model is not able to generalize systematically. Secondly, implementing modularity by separating the problem into higher-level cognitive processes enables the model to use the additional experience and generalize systematically on tests related to the additional experience. Finally, naively adding experience does not enable systematic generalization. For the model to learn how to generalize systematically, the experience it sees needs to be sufficiently similar to the type of systematicity it is tested on.

The three key contributions of this paper are (1) We investigate a type of adverb-verb compositionality that has seen little progress in the past, (2) We propose a neural architecture for systematic generalization that is modular at the task-level, (3) We analyse the amount and type of data this network needs to achieve improved systematic generalization.

Related Work

In recent years systematic generalization has seen a revived interest from the machine learning community, in computer vision (Johnson et al., 2017; Misra, Gupta, & Hebert, 2017; Atzmon, Kreuk, Shalit, & Chechik, 2020; F. Ruis, Burghouts, & Bucur, 2021), natural language processing (Lake & Baroni, 2018; Baroni, 2020; Keysers et al., 2020; Kim & Linzen, 2020), and more generally (Nam & McClelland, 2021). Two fundamentally different approaches are taken by the literature; one utilizes additional data while making few changes to the conventional setup and architecture (Furrer, van Zee, Scales, & Schärli, 2020), while the other utilizes additional inductive biases that aim to support systematic generalization (Russin et al., 2019; Lake, 2019; Andreas, 2020; Nye et al., 2020; Gordon et al., 2020; Bogin et al., 2021; Chaabouni, Dessì, & Kharitonov, 2021). In this work we apply both approaches, the former through data augmentation, and the latter through modularity.

Using modularity to improve systematic generalization is a common strategy in the literature. Some approaches introduce modules that directly map to different parts of the input command (Andreas et al., 2016; Corona, Fried, Devin, Klein, & Darrell, 2021), and others map to different attributes of an input image (Purushwalkam et al., 2019). In this work we design modules that tackle different parts of the task, like navigation and perception. Additionally, we use data augmentation to provide the modules with separate experience. Data augmentation for systematic generalization is also a well-studied area (Lake, 2019; Andreas, 2020; Kagitha, 2020). Here we take a structured augmentation approach that works even in the grounded setting of the benchmark, designing a set of rules that can generate novel examples. This allows us to generate separate data for each module as well as additional data with novel concepts that are not part of the original dataset.

Evaluating Systematic Compositionality

To evaluate the systematicity of the model's predictions, we will use the grounded SCAN benchmark (gSCAN). The benchmark tests a broad set of phenomena in situated language understanding where humans should easily generalize, but where computational models struggle due to systematic differences between the training data and the test data. In gSCAN, agents are asked through language commands to execute instructions in a 2D gridworld. The benchmark has seven different tests that require combining known concepts into novel meaning. For example, one test requires recognizing a novel object with a familiar color and shape that have never been observed together. Another test evaluates whether a model is able to interpret a command containing an unseen composition of a familiar verb and adverb. This latter test is of the kind that is unsolved by prior work, hence the subject of our focus.

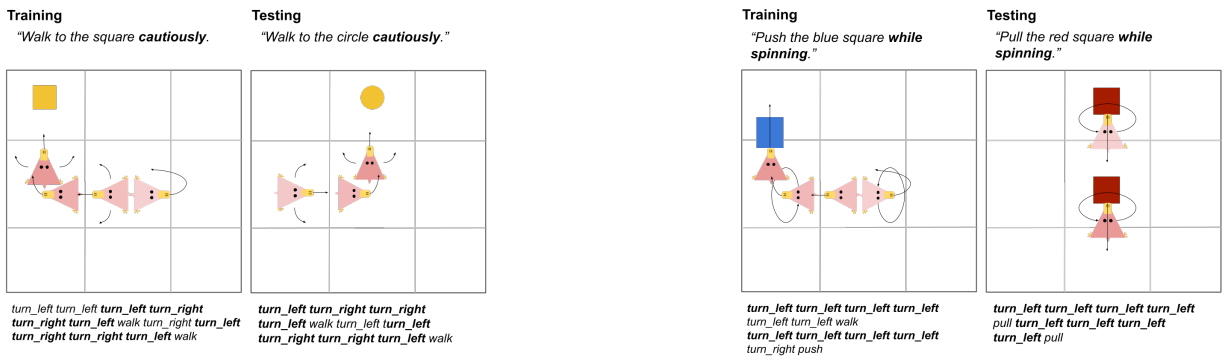
Figure 1 depicts examples of gSCAN tasks with adverbs[2]. The use of an adverb in the input command indicates a changed manner of navigation across the grid, requiring a complete transformation of the output. For example, transforming a sequence that simply navigates from the bottom right to the top left to one that does this while spinning works as follows:

"turn_left walk walk turn_left walk walk"

$\downarrow f_{\text{while spinning}}$

"spin* turn_left walk spin* walk spin* turn_left walk spin* walk"

[2]Note that some of the adverbs mentioned are adverbial phrases.



(a) (b)

Figure 1: This figure depicts the two tests of adverb compositionality in gSCAN. Figure (a) denotes a few-shot learning test; a model has access to few (k) examples of how the adverb "cautiously" translates to an output sequence and needs to generalize to all other examples. Figure (b) denotes the "pull while spinning"-test; reminiscent of the "cycle cautiously"-example, a model learns all examples of pushing while spinning or walking while spinning, and is tested on its ability to interpret "pull while spinning".

where spin* is not a primitive but an action sequence of four times "turn_left". The underlying rule is to add a spin before turning to a direction and moving (e.g., "turn_left walk" or "turn_right push"). When the manner is "cautiously", the transformed sequence would look as follows: "turn_left cautious* walk cautious* walk turn_left cautious* walk cautious* walk", where cautious* is an action sequence of "turn_left turn_right turn_right turn_left" (looking to the left and right). Here the rule is slightly different, as the agent first needs to turn to the direction it will continue in, then be cautious, and then take the movement action. This difference will prove important in the experiment section where we analyze what kind of data augmentation results in systematicity.

Figure 1 exemplifies the two types of systematic generalization within our focus here and that have not been solved by prior work. One of the tests focuses on learning something new from few examples, and the other on composing familiar things. Figure 1a shows a test of few-shot generalization to "cautiously." At training time, the model only sees a few (k) examples of commands with "cautiously"; at test time, the model needs interpret all unseen commands with that manner. Figure 1b shows a test of adverb-to-verb generalization. At training time, the model sees all commands that contain "push ... while spinning" and "walk ... while spinning", and at test time it needs to interpret "pull ... while spinning".

The object properties and their denotations are combined to form a training set of 367,933 different examples. Each command in the output sequence can be one of five actions (e.g., walk, turn_left, etc.). For a full description of the dataset statistics refer to the appendix of L. Ruis et al., (2020).

Method

The problem posed by gSCAN naturally divides into modules reminiscent of high-level cognitive processes. Below we describe what each module does, how we generate data for these modules, and how the resulting model is trained.

The modular architecture. To facilitate modularity, we aim to separate the task at hand into different cognitive modules; perception, navigation, interaction, and manner of navigation (or, transformation). Each module has its own input and output, all modules additionally have access to the language command, and some have access to the world state. For an overview of the modules and their input-output flow, see Figure 2.

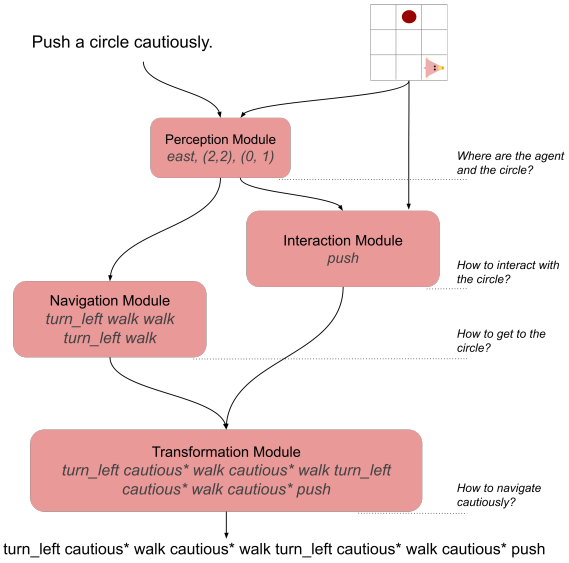


Figure 2: The input command ("Push a circle cautiously.") and world state are processed by different modules, each dealing with a different question about the input task. The final output is produced by the transformation module. *: cautious is in reality not a primitive action but a sequence of "turn_left turn_right turn_right turn_left"

The perception module takes the input command and world state and outputs where the agent is (a tuple with a row and column number), what direction it is facing (east, north, south, or west), and where the target is (a tuple with a row and column number). *The navigation module* uses that information and outputs a plan to get from the agent location to the target location. This module needs access to the input command because there are adverbs like "while zigzagging" that require the module to output a different plan (instead of "turn_left walk walk turn_left walk" like the plan in the figure, it would output "turn_left walk turn_left walk turn_right walk"). Ad-

ditionally, some adverbs require an egocentric plan like the one in Figure 2, but others require an allocentric plan. The allocentric equivalent to the plan in Figure 2 is "North North West". The navigation module decides what mode of output, allo- or egocentric, to use based on the input command. In the next paragraph the reason for this distinction is detailed. *The interaction module* gets the output from the perception module and the world state, and can use the target location to decide how to interact with the object. In this case, it needs to be pushed once. This module uses the input command to decide whether to push or pull. Finally, *the transformation module* gets the plan and the interaction commands, and transforms the whole sequence to adhere to the manner of navigation. In this case, it needs to navigate cautiously ("turn_left turn_right turn_right turn_left", i.e., looking to the left and right, shortened in Figure 2 for clarity) before each movement command ("walk", "push", and "pull"). This module uses the input command to decide which transformation is necessary and always outputs egocentric commands, even if the input is allocentric. Inputs like "North" are transformed to "spin* turn_left walk" if the adverb is "while spinning" (where spin* is "turn_left turn_left turn_left turn_left"). Sometimes an example requires no interactions and/or no transformation (e.g., "walk to the circle"). In that case, the interaction and transformation modules output a special end-of-sequence token.

Structured data augmentation. Each module needs experience to learn, and in order to analyze how much experience they need, we construct a domain-specific language (DSL) under which the original gSCAN benchmark is grammatical. The DSL contains everything to generate the already existing manners of navigation found in gSCAN and can be used to sample new manners of navigation. We can use it to generate as much data as needed. The DSL is constructed as a functional L-system (Lindenmayer, 1968). L-systems are parallel rewriting systems of rules. An L-system has a set of symbols that can form sequences, and production rules to rewrite those sequences. By applying rules from the L-system to sequences (which can be empty), a sequence can be grown. L-systems are executed in parallel, meaning that a rule gets applied to all symbols in the sequence at once. Each adverb gets assigned a set of rewrite rules, called a program. Applying this set of rules to a sequence will transform it into a sequence with the manner that the program corresponds to.

To show how the DSL works, we will walk through an example. The DSL uses allocentric and egocentric symbols, which are indicated by capitalized and lowercase symbols respectively (e.g., "North" or "turn_left" and "walk"). This distinction more naturally fits the already existing adverbs in gSCAN and results in less rewriting rules for their programs. For the adverb "while spinning" the transformation can be applied to allocentric commands, and for the adverb "cautiously" to egocentric commands. For example, to construct the sequence in Figure 2, we apply the following rewrite-rule to the egocentric plan "turn_left walk walk turn_left walk":

$$\text{walk} \rightarrow \text{turn\_left turn\_right turn\_right turn\_left walk}$$

Which puts the "cautious"-sequence at the right location, before the movement symbol. Because L-systems are parallel rewriting systems, the above rule needs to be applied once to transform the sequence. For "while spinning", and an allocentric plan "North North West", the rewrite rules will be:

$$\text{North} \rightarrow \text{turn\_left turn\_left turn\_left turn\_left North}$$

$$\text{West} \rightarrow \text{turn\_left turn\_left turn\_left turn\_left West}$$

If then, based on the agent's starting direction, North gets rewritten to "turn_left walk" the spin is already at the right location in the sequence (namely before turning into the direction that the agent will be walking in). Rules can be applied recursively, which is why the DSL can theoretically generate infinite data. If the above rules are applied ad infinitum the agent will never stop spinning.

Constructing the rules for each adverb leads to a set that we call the meta grammar. We extend the meta grammar with additional rules to be able to sample new programs that correspond to different manners and adverbs. For example, one rule that can generate manners of navigation that aren't a part of the original gSCAN is the following:

$$\text{East} \rightarrow \text{North East South}$$

Applying this type of rule will make the agent take a small detour instead of going east immediately. To get a dataset with X extra adverbs, we sample as many programs from the meta grammar. If a program gets generated that has the same set of rules as one of the original gSCAN adverbs, it is rejected[3].

The subtle differences among the original gSCAN manners induce a categorization of adverb types; "while spinning"-type adverbs can be described by transformations that are naturally applied to allocentric commands and do not result in changed movement across grid-cells (they only result in changed movement *within* grid-cells), "cautiously"-type adverbs are described by transformations that are applied to egocentric commands and also do not result in changed movement, and "while zigzagging"-type adverbs are applied to allocentric sequences and do result in a changed path across the grid. Finally, the manners resulting from rules that make the agent take a detour are not part of the original gSCAN dataset and are different in that they cause the agent to walk more than needed. We use this taxonomy to evaluate the type of experience a model needs to generalize systematically.

Network architecture & training details. The neural network we use as a non-modular baseline is exactly the same as the one described by L. Ruis et al. (2020), as well as the hyperparameters used. For each module in the modular network we re-use the parts of the baseline architecture that apply. Based on the input-type and output-type a module requires, different neural networks are combined into a module. For example, the perception module's encoder is exactly the same as the baseline, since it also needs to process the input command and the world state. The decoder does not need to output a sequence but three integer predictions (initial agent direction and position, and target position), therefore the decoder is an multi-layer perceptron (MLP) with three output heads. The

[3]Find the full DSL: `https://github.com/LauraRuis/msa`.

transformation module processes the concatenation of the sequences that the navigation and interaction modules output and outputs a sequence, hence both the encoder and the decoder are recurrent neural networks. The full architecture and code for the experiments can be found in the same repository as the DSL. The modules get trained independently and in parallel with ground-truth inputs and targets generated by the DSL. At test time, the forward-pass through the modules happens sequentially and the modules that take the output from the previous module take their predicted output. We train everything with Adam optimizer (Kingma & Ba, 2015). For both the baseline and the transformation module we found that when adding additional augmented data, the neural network size needs to be increased to prevent underfitting. Whenever we add augmented data to a model, we increase the recurrent hidden sizes from 100 to 400, the embedding dimension from 25 to 50, and decrease dropout from 0.3 to 0.2.[4]

Experiments

To analyze the effect of modularity and data augmentation on systematic generalization we evaluate the models on gSCAN. Besides the two tests of our focus that require the type of adverb compositionality that prior work struggles with, we report the performance on the five remaining tests to ensure that a performance increase doesn't present a trade-off between tests, as well as on a "random" test set that is sampled from the same distribution as the training set. This is a sanity check that all models are able to solve the gSCAN task when the test set has no systematic differences with the training set. We compare the proposed models to the end-to-end neural network used in the original gSCAN paper. Upon establishing the best performing model, we run several experiments to understand the performance increase. Specifically, we look at the effect of the number of adverbs in the training set, the type of adverbs in the training set, and the number of "cautiously"-examples in the training set (i.e., the k in k-shot).

The effect of modularity and augmentation. We train the baseline and the modular model outlined above on the original gSCAN training set containing four adverbs and on a training set containing 150 additional adverbs generated by the DSL. In Table 1 the results on a random test set with no systematic differences with the training set and on the seven systematic generalization tests is shown. When looking at the tests of our focus, row "Cautiously k=5-shot" and "Pull while spinning", we see that the modular network (column "Modular only") gives a clear improvement over the non-modular baseline (column "Baseline") for the few-shot learning test. For the adverb-to-verb generalization there is no clear improvement from modularity alone. However, once we add the augmented dataset with 150 extra adverbs the performance jumps significantly (column "Modular & Augmentation"). This provides an improvement of +/- 80% exact match over the baseline for the few-shot split, and 55% for the "pull while spinning"-split.

The question is whether modularity is necessary; what if we simply train the baseline with the augmented data? This experiment is depicted in the column labeled "Augmentation only" of Table 1. We observe two things in this result. Firstly, the modular method with augmentation outperforms the baseline with augmentation for the "pull while spinning"-split. Secondly and more importantly, training the non-modular baseline with the augmented dataset results in worse performance than the baseline for four of the five other tests in gSCAN. One interesting possible explanation for this is the fact that the augmented data has the same systematic differences with the test sets as the original training set and more exposure to this data increases confidence in the biases extracted from the original training set. For example, in the augmented data generated yellow squares are still never mentioned in the input command. This means that the performance increase for the adverb-splits becomes a trade-off, whereas in the modular case we can simply only provide the transformation module with the augmented data to circumvent this issue.

The best method from literature for the 5-shot split achieves 10.31% exact match (Kuo et al., 2021), and for the pull while spinning split the best method achieves 33.6% exact match (Gao et al., 2020), making this work is the first to improve on these tests. Even though our method is advantaged through the augmented adverb set, an important result is evident: to make progress on systematic generalization simply adding extra data is not always enough. In this case, we use high-level modularity to make full use of the structured data augmentation.

The effect of vocabulary size and varied k. Figure 3a contains the results of runs on training sets with different vocabulary sizes. For the 5-shot "cautiously" split, we observe a slow increase of performance when going from 0 to 100 adverbs in the training set, and a big jump when adding the final 50 adverbs. For the "pull while spinning" split adding extra adverbs is not always strictly better[5]. This result shows that simply adding more adverbs does not guarantee performance increase. If the model could transfer experience with any adverb to the two that are the subject of the adverb tests we should see a steady increase in performance when adding additional adverbs. What we can infer from these results is that the vocabulary size is not the full picture, and we need to break the dataset down per adverb type to get a clearer idea, which we do in the next section.

Figure 3b shows the performance as a function of the number of training examples with "cautiously". The result is unsurprising; more is better, and after 10 examples the effect of adding more diminishes. The difference in performance for the "pull while spinning" split can be explained by the high variance of these results.

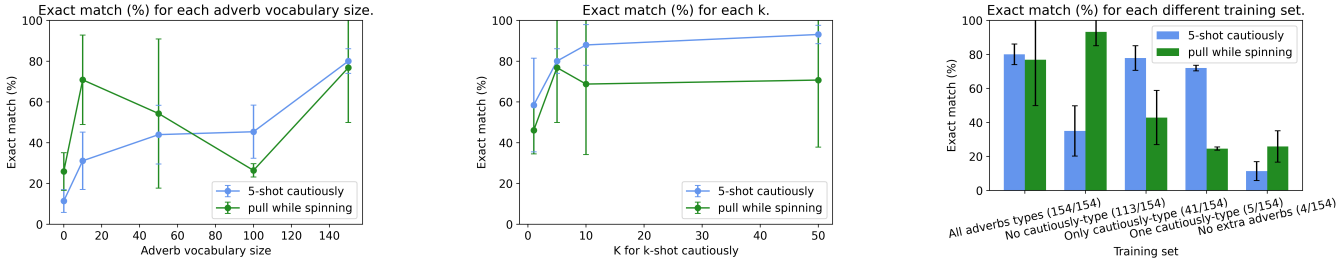
The effect of manner-similarity. It turns out simply adding more adverbs to the training set does not suffice for strong

[4]Full hyperparameter details: https://github.com/LauraRuis/msa

[5]In the training sets containing 0, 10, 50, or 150 adverbs the dominant adverb type is the "while spinning"-type. In the training set with 100 extra adverbs however, it's the "cautiously"-type (3% more "cautiously"-type examples). This might explain the performance dip for the "pull while spinning" test when training on 100 adverbs. To test this hypothesis more experiments are needed.

Table 1: Results obtained by the models for each split, showing exact match accuracy (average of 5 runs ± std. dev.). The rightmost three columns show current state-of-the-art (SOTA) models.

	Exact Match (%)						
Split	**Baseline**	**Modular & Augmentation**	**Modular only**	**Augmentation only**	**Qiu et al.**	**Kuo et al.**	**Gao et al.**
Random	97.15 ± 0.46	96.34 ± 0.28	96.35 ± 0.29	96.13 ± 0.28	**99.95 ± 0.02**	97.32	98.6 ± 0.95
Cautiously k=5-shot	1.12 ± 0.46	**80.04 ± 6.06**	11.40 ± 5.59	76.26 ± 17.18	-	10.31	-
Pull while spinning	19.04 ± 4.08	**76.84 ± 26.94**	25.87 ± 32.09	25.27 ± 4.89	22.16 ± 0.01	21.95	33.6 ± 20.81
Yellow squares	30.05 ± 26.76	59.66 ± 23.76	59.66 ± 23.76	22.89 ± 22.40	**99.90 ± 0.06**	95.35	99.08 ± 0.69
Red squares	29.79 ± 17.70	32.09 ± 9.79	32.09 ± 9.79	10.27 ± 4.09	**99.25 ± 0.91**	80.16	80.31 ± 24.51
Novel direction	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	**5.73**	0.16 ± 0.12
Relativity	37.25 ± 2.85	49.34 ± 11.60	49.34 ± 11.60	31.42 ± 3.91	**99.02 ± 1.16**	75.19	87.32 ± 27.38
Class inference	94.97 ± 1.12	94.16 ± 1.25	94.06 ± 1.21	93.65 ± 2.49	**99.98 ± 0.01**	98.63	99.33 ± 0.46



(a) (b) (c)

Figure 3: The average exact match of five runs on the adverb splits varying (a) the number of additional adverbs (0, 10, 50, 100, 150), (b) the k in k-shot learning (1, 5, 10, 50), or (c) the distribution of adverb types in the training set. The error bars show the std. dev. over 5 training runs.

systematic generalization. As detailed in the method section, the different gSCAN adverbs can be divided in three different groups. Adverbs in the same group require a similar manner of navigation. In this experiment (depicted in Figure 3c), we train a model on different subsets of the dataset containing 150 augmented adverbs in addition to the original four gSCAN adverbs ("All adverb types" in Figure 3c). Sorted from most number of adverbs to least number of adverbs we have; a subset containing the "while spinning" and "while zigzagging"-type adverbs but not the "cautiously"-type ("No cautiously-type"), a disjoint subset with only the "cautiously"-type adverbs, a subset containing just one adverb (picked to be highly similar to "cautiously", requiring an action sequence of "turn_right turn_left turn_left turn_right" before movement actions instead of "turn_left turn_right turn_right turn_left"), and a subset without extra adverbs containing only the original four gSCAN adverbs ("No extra adverbs" in Figure 3c). The results of this experiment give a clearer picture than vocabulary size alone. It shows that the type of adverb is crucial for the performance. The model trained on the dataset without "cautiously"-type adverbs does only slightly better on the few-shot "cautiously" split than when trained without extra adverbs, but very well on the "pull while spinning" split. In fact we see that taking out the adverbs that are similar to "cautiously" boosts performance significantly over using all 150 adverbs on the "pull while spinning" split. This shows that the adverb similarity is important for knowledge transfer between adverbs and adding adverbs of another type can even hurt performance. When taking this experiment to the extreme by adding a single adverb that is similar to "cautiously" to the original adverbs ("one cautiously-type" in Figure 3c), we observe that this almost explains all performance increase for the few-shot "cautiously"-split. This experiment suggests that it is not the quantity, but the quality of data that we add that is important, and that the models may be relying on nearest-neighbor-style reasoning when generalizing to new adverbs.

Conclusion

In this work we investigate adverb-verb compositionality that has a transformative effect on action sequences required to perform a verb. This type of compositionality has seen little progress in the past. By applying two influential modeling principles, modularity and data augmentation, we set a new state-of-the-art on two grounded language understanding tests that evaluate this. Even though the method has an advantage over previous methods due to the augmented data, the results give insight into what can bring about progress on the set of tests that thus far have proven difficult. We find that naively adding substantial experience with different adverbs to the training set of a neural network is not sufficient for strong generalization, but when separating the architecture into different modules reminiscent of high-level cognitive processes the model is able to generalize more systematically. Moreover, we find it is not the quantity of experience that matters, but the quality; adding additional adverbs to the training set that correspond to a different manner of navigation than the one that is tested barely results in transfer of the learned knowledge and can even hurt generalization, whereas adding a single adverb that is very similar to the tested adverb almost explains the entire performance increase. These lessons may translate more broadly; to move towards truly systematic neural networks we may need to take into account the quality of experience a model sees, and the inductive biases it needs to be able to utilize this experience systematically.

One weakness of the proposed architecture is that some aspects are specialized for the task at hand. For future work, it would be an interesting challenge to eliminate the need for designing a structured augmentation method by hand, as well as learning the module layout from data in order to readily apply it to other domains.

References

Andreas, J. (2020). Good-enough compositional data augmentation. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7556–7566). Online: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/2020.acl-main.676 doi: 10.18653/v1/2020.acl-main.676

Andreas, J., Rohrbach, M., Darrell, T., & Klein, D. (2016). Neural module networks. In *2016 ieee conference on computer vision and pattern recognition (cvpr)* (p. 39-48). doi: 10.1109/CVPR.2016.12

Atzmon, Y., Kreuk, F., Shalit, U., & Chechik, G. (2020). A causal view of compositional zero-shot recognition. In *Advances in neural information processing systems (neurips)*.

Bahdanau, D., Murty, S., Noukhovitch, M., Nguyen, T. H., de Vries, H., & Courville, A. (2019). Systematic generalization: What is required and can it be learned? In *International conference on learning representations*. Retrieved from https://openreview.net/forum?id=HkezXnA9YX

Baroni, M. (2020). Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *375*(1791), 20190307. Retrieved from https://royalsocietypublishing.org/doi/abs/10.1098/rstb.2019.0307 doi: 10.1098/rstb.2019.0307

Bastings, J., Baroni, M., Weston, J., Cho, K., & Kiela, D. (2018). Jump to better conclusions: SCAN both left and right. In *Proceedings of the emnlp blackboxnlp workshop* (pp. 47–55). Brussels, Belgium.

Bogin, B., Subramanian, S., Gardner, M., & Berant, J. (2021). Latent compositional representations improve systematic generalization in grounded question answering. *Trans. Assoc. Comput. Linguistics*, *9*, 195-210. Retrieved from https://transacl.org/ojs/index.php/tacl/article/view/2489

Calvo, P., & Symons, J. (Eds.). (2014). *The architecture of cognition: Rethinking Fodor and Pylyshyn's systematicity challenge*. Cambridge, MA: MIT Press.

Chaabouni, R., Dessì, R., & Kharitonov, E. (2021, November). Can transformers jump around right in natural language? assessing performance transfer from SCAN. In *Proceedings of the fourth blackboxnlp workshop on analyzing and interpreting neural networks for nlp* (pp. 136–148). Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.blackboxnlp-1.9 doi: 10.18653/v1/2021.blackboxnlp-1.9

Chomsky, N. (1957). *Syntactic structures*. Berlin, Germany: Mouton.

Corona, R., Fried, D., Devin, C., Klein, D., & Darrell, T. (2021, June). Modular networks for compositional instruction following. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 1033–1040). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.naacl-main.81 doi: 10.18653/v1/2021.naacl-main.81

D'Amario, V., Sasaki, T., & Boix, X. (2021). How modular should neural module networks be for systematic generalization? *CoRR*, *abs/2106.08170*. Retrieved from https://arxiv.org/abs/2106.08170

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021, August). A survey of data augmentation approaches for NLP. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 968–988). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.findings-acl.84 doi: 10.18653/v1/2021.findings-acl.84

Fodor, J. (1983). *The modularity of mind: An essay on faculty psychology*. MIT Press.

Fodor, J., & Lepore, E. (2002). *The compositionality papers*. Oxford, UK: Oxford University Press.

Fodor, J., & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*, 3–71.

Furrer, D., van Zee, M., Scales, N., & Schärli, N. (2020). Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *CoRR*, *abs/2007.08970*. Retrieved from https://arxiv.org/abs/2007.08970

Gao, T., Huang, Q., & Mooney, R. (2020, December). 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* (pp. 491–503). Suzhou, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.aacl-main.49

Gershman, S. J., & Tenenbaum, J. B. (2015). Phrase similarity in humans and machines. *Cognitive Science*.

Gordon, J., Lopez-Paz, D., Baroni, M., & Bouchacourt, D. (2020). Permutation equivariant models for compositional generalization in language. In *International conference on learning representations*. Retrieved from https://openreview.net/forum?id=SylVNerFvr

Heinze-Deml, C., & Bouchacourt, D. (2020). Think before you act: A simple baseline for compositional generalization. *CoRR*, *abs/2009.13962*. Retrieved from https://arxiv.org/abs/2009.13962

Jiang, Y., & Bansal, M. (2021). Inducing transformer's compositional generalization ability via auxiliary sequence prediction tasks. In M. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing, EMNLP 2021, virtual event / punta cana, dominican republic, 7-11 november, 2021* (pp. 6253–6265). Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.emnlp-main.505

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L.,

Zitnick, C. L., & Girshick, R. B. (2017). Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1988-1997.

Kagitha, P. P. (2020). Systematic Generalization Emerges In Seq2seq Models With Variability in Data. *BAICS workshop ICLR*. Retrieved from https://baicsworkshop.github.io/pdf/BAICS_11.pdf

Keysers, D., Schärli, N., Scales, N., Buisman, H., Furrer, D., Kashubin, S., ... Bousquet, O. (2020). Measuring compositional generalization: A comprehensive method on realistic data. In *International conference on learning representations.* Retrieved from https://openreview.net/forum?id=SygcCnNKwr

Kim, N., & Linzen, T. (2020, November). COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 9087–9105). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-main.731 doi: 10.18653/v1/2020.emnlp-main.731

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings.* Retrieved from http://arxiv.org/abs/1412.6980

Kuo, Y., Katz, B., & Barbu, A. (2021). Compositional networks enable systematic generalization for grounded language understanding. In M. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Findings of the association for computational linguistics: EMNLP 2021, virtual event / punta cana, dominican republic, 16-20 november, 2021* (pp. 216–226). Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.findings-emnlp.21

Lake, B. (2019). Compositional generalization through meta sequence-to-sequence learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2019/file/f4d0e2e7fc057a58f7ca4a391f01940a-Paper.pdf

Lake, B., & Baroni, M. (2018). Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of icml* (pp. 2879–2888). Stockholm, Sweden.

Lindenmayer, A. (1968). Mathematical models for cellular interactions in development. ii. simple and branching filaments with two-sided inputs. *Journal of theoretical biology*, *18 3*, 300-15.

Loula, J., Baroni, M., & Lake, B. (2018). Rearranging the familiar: Testing compositional generalization in recurrent networks. In *Proceedings of the emnlp blackboxnlp workshop* (pp. 108–114). Brussels, Belgium.

Marcus, G. (1998). Rethinking eliminative connectionism. *Cognitive Psychology*, *282*(37), 243–282.

Marcus, G. (2003). *The algebraic mind*. Cambridge, MA: MIT Press.

Meyer, B. (1997). *Object-oriented software construction* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Misra, I., Gupta, A., & Hebert, M. (2017). From red wine to red tomato: Composition with context. In *2017 ieee conference on computer vision and pattern recognition (cvpr)* (p. 1160-1169). doi: 10.1109/CVPR.2017.129

Montague, R. (1970). Universal Grammar. *Theoria*, *36*, 373–398.

Nam, A. J., & McClelland, J. L. (2021). What underlies rapid learning and systematic generalization in humans. *CoRR*, *abs/2107.06994*. Retrieved from https://arxiv.org/abs/2107.06994

Nye, M. I., Solar-Lezama, A., Tenenbaum, J., & Lake, B. (2020). Learning compositional rules via neural program synthesis. In *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, neurips 2020, december 6-12, 2020, virtual.* Retrieved from https://proceedings.neurips.cc/paper/2020/hash/7a685d9edd95508471a9d3d6fcace432-Abstract.html

Nye, M. I., Tessler, M. H., Tenenbaum, J. B., & Lake, B. (2021). Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning. *CoRR*, *abs/2107.02794*. Retrieved from https://arxiv.org/abs/2107.02794

Poole, D., & Mackworth, A. (2017). *Artificial intelligence: Foundations of computational agents* (2nd ed.). Cambridge, UK: Cambridge University Press. Retrieved from http://artint.info/2e/html/ArtInt2e.html

Purushwalkam, S., Nickel, M., Gupta, A. K., & Ranzato, M. (2019). Task-driven modular networks for zero-shot compositional learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3592-3601.

Qiu, L., Hu, H., Zhang, B., Shaw, P., & Sha, F. (2021, November). 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* (pp. 2180–2188). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.emnlp-main.166 doi: 10.18653/v1/2021.emnlp-main.166

Ruis, F., Burghouts, G. J., & Bucur, D. (2021). Independent prototype propagation for zero-shot compositionality. In *Advances in neural information processing systems (neurips)* (Vol. 34).

Ruis, L., Andreas, J., Baroni, M., Bouchacourt, D., & Lake, B. (2020). A benchmark for systematic generalization in grounded language understanding. In H. Larochelle, M. Ran-

zato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 19861–19872). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2020/file/e5a90182cc81e12ab5e72d66e0b46fe3-Paper.pdf

Russin, J., Jo, J., O'Reilly, R. C., & Bengio, Y. (2019). Compositional generalization in a deep seq2seq model by separating syntax and semantics. *arXiv preprint*. Retrieved from http://arxiv.org/abs/1904.09708

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *J. Big Data*, *6*, 60. Retrieved from https://doi.org/10.1186/s40537-019-0197-0 doi: 10.1186/s40537-019-0197-0