Compositionality as Directional Consistency in Sequential Neural Networks

Najoung Kim
Department of Cognitive Science
Johns Hopkins University
n.kim@jhu.edu

Tal Linzen
Department of Cognitive Science
Johns Hopkins University
tal.linzen@jhu.edu

Abstract

Sequential neural networks have shown success on a variety of natural language tasks, but through what internal mechanisms they achieve systematic compositionality crucial to language understanding is still an open question. In particular, gated networks such as Gated Recurrent Units (GRUs) are known to significantly outperform Simple Recurrent Neural Networks (SRNs). We conduct an exploratory study comparing the abilities of SRNs and GRUs to make compositional generalizations, using adjective semantics as testing ground. Our results demonstrate that GRUs generalize more systematically than SRNs. On analyzing the learned representations, we find that GRUs encode the compositional contribution of adjectives as directionally consistent linear displacements. This consistency correlates with generalization accuracy within GRUs, suggesting that it is an effective strategy for deriving more compositionally generalizable representations.

1 Introduction

The impressive performance of neural networks in natural language processing (NLP), a domain in which symbolic representations are traditionally viewed as indispensable, raises the question of how these models accomplish (or approximate) symbolic compositionality. Among sequential neural networks, gated models such as Long Short-Term Memory (LSTMs) [1] and Gated Recurrent Units (GRUs) [2] outperform Simple Recurrent Neural Networks (SRNs) in a range of sequence modeling tasks [3] including language modeling [4], and achieve better compositional generalization [5, 6]. In this paper, we conduct an exploratory study testing whether this difference can be explained by geometric regularities, using adjective semantics as our testing ground. Specifically, we investigate whether semantic contribution that remains invariant across multiple contexts can manifest as geometric regularity in the sequence representations encoded by these networks, similarly to [7] where consistent vector offsets denote the same relation between pairs of words in the embedding space.

1.1 Related work

Our work shares motivation with neural network analysis work aiming to "open the black box" [8], especially regarding compositionality [5, 9, 10, 11]. We focus on systematic compositionality, the "algebraic capacity to understand [...] novel utterances by combining familiar primitives" [12]. Prevalent approaches for analyzing neural NLP models include auxiliary classifiers, challenge sets, and adversarial perturbation targetting specific linguistic properties [13]. Although such methods serve as useful probes for gauging what the models are capable of, they provide limited insight about the learned representations. We analyze the properties of model-internal representations, along the lines of [14, 15]. We train models to perform Natural Language Inference (NLI) [16], as in [17, 18, 19], and also draw from works that use synthetic datasets for conducting focused evaluations

Workshop on Context and Compositionality in Biological and Artificial Neural Systems, 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

of linguistic phenomena [15, 20, 21, 22]. Finally, our work shares topical interests with NLP literature on semantic compositionality [23, 24], logical reasoning [25, 26], and adjective semantics [27].

2 Methodology

2.1 Dataset for testing compositional semantic generalization

We design a task that tests a model's capacity to make compositional semantic generalizations. For this task, we adopt the NLI [16] setup. The input consists of a premise-hypothesis (p/h) pair, and the task is to predict whether p entails h. We use a binary version of NLI, where the labels are {*entailed, not entailed*}. Solving our task is contingent upon correctly understanding the effect of adjectives on the entailment pattern between p and h. The dataset consists of training, development and generalization sets, where the generalization set contains classes of examples not shown during training (zero-shot), but are such that we expect a model that makes human-like compositional generalizations to be able to solve. In particular, we target two patterns: (1) generalization to unseen sequences, and (2) generalization from complex to simpler compositional forms (see Table 1).

Table 1: Example p/h pairs from the dataset. $\rightarrow$ denotes 'entails' and $\nrightarrow$ denotes 'does not entail' (not all template types are listed, due to space constraints).

Training/Development sets		
Adj$_1$ *Adj*$_2$ *N* $\rightarrow$ *Adj*$_2$ *N*	Mary is a tall American lawyer. $\rightarrow$ Mary is an American lawyer.	
Adj$_1$ *Adj*$_2$ *N* $\nrightarrow$ *Adj*$_2$ *N*	Mary is a former American lawyer. $\nrightarrow$ Mary is an American lawyer.	
Adj$_1$ *Adj*$_2$ *N* $\rightarrow$ *N*	Mary is a tall American lawyer. $\rightarrow$ Mary is a lawyer.	
Adj$_1$ *Adj*$_2$ *N* $\nrightarrow$ *N*	Mary is a former American lawyer. $\nrightarrow$ Mary is a lawyer.	
Generalization set		
Adj$_1$ *Adj*$_2$ *N* $\rightarrow$ *Adj*$_2$ *N*	Mary is a tall former lawyer. $\rightarrow$ Mary is a former lawyer.	(unseen)
Adj$_1$ *Adj*$_2$ *N* $\nrightarrow$ *Adj*$_1$ *N*	Mary is a tall former lawyer $\nrightarrow$ Mary is a tall lawyer.	(unseen)
Adj N $\rightarrow$ *N*	Mary is a tall lawyer. $\rightarrow$ Mary is a lawyer.	(complex to simple)
Adj N $\nrightarrow$ *N*	Mary is a former lawyer. $\nrightarrow$ Mary is a lawyer.	(complex to simple)

Training set. We use two adjective classes that give rise to different entailment patterns [28, 29]. When *Adj* is a **subsective** adjective, *Adj N* entails *N* (e.g., *tall president* $\rightarrow$ *president*); when it is a **nonsubsective** adjective, *Adj N* does not entail *N* (e.g., *fake president* $\nrightarrow$ *president*). There are no input pairs such as *John is a former teacher* $\nrightarrow$ *John is a teacher* or *John is a tall teacher* $\rightarrow$ *John is a teacher*, that clearly indicate to the model whether or not a particular adjective is subsective.

Generalization set. The generalization set tests for the following two types of generalizations, which we would expect from a model that has successfully learned the semantic contribution of adjectives included in the training set:

- **Generalization to unseen sequences.** The generalization set contains unseen sequences of adjectives, each of which is included in the training set. For instance, *tall American* and *former American* both appear in the training set, but *tall former American* is unseen.
- **Generalization from complex to simple form.** The set also contains examples that require teasing apart the individual contributions of each adjective. The individual contributions are not explicitly shown in the training/development sets. For instance, *tall x* $\rightarrow$ *x*, but *former x* $\nrightarrow$ *x*.

Generation. We use templates *Subj is a Adj*$_1$ *Adj*$_2$ *N* $\rightarrow$ *Subj is a Adj*$_{1/2}$ *N* and *Subj is a Adj*$_1$ *Adj*$_2$ *N* $\nrightarrow$ *Subj is a Adj*$_{1/2}$ *N* to generate training data (see Table 1), using 12 different subsective adjectives (half in *Adj*$_1$ position and half in *Adj*$_2$ position) and 4 nonsubsective adjectives (only seen in *Adj*$_1$ position in training). We use 9 different noun phrases that can appear in *Subj* position, which can be either one or two words long to keep the length of the whole sequence variable (e.g., *Mary, my dad*). We use 10 nouns that appear in the *N* position. These nouns are single words that are potentially modified by the adjectives (e.g., *president, student*). We also add two trivial cases: (1) self-entailment ($X \rightarrow X$), and (2) non-entailment of subject-mismatched p/h (e.g., *x is a z* $\nrightarrow$ *y is a*

z). 23,400 unique pairs are generated through this process, 15% of which are used as a development set ($|\textit{train}| = 19,890$, $|\textit{dev}| = 3,510$). For the generalization set, we use the same templates but with nonsubsective $\textit{Adj}_2$ in the premise, for generating the unseen sequences. New templates *Subj is a Adj N* $\rightarrow$ *Subj is a N* and *Subj is a Adj N* $\nrightarrow$ *Subj is a N* are used for the complex to simple form generalization cases. This process yields $|\textit{test}| = 15,120$.

2.2 Geometric measures

We test the hypothesis that geometric consistency is used to represent the compositional contribution of adjectives that is constant across different contexts (e.g., different nouns that the adjective modifies). For instance, we expect an adjective such as *former* to have some common meaning shared across different linguistic contexts it appears in, rather than carrying an idiosyncratic meaning in every use. In our task specifically, adjective subsectivity should be contextually invariant. One simple way this context-invariant semantics could be captured is through a constant linear displacement. We compute the direction and magnitude consistency of vector offsets to test the hypothesis that the contribution of adjectives to the meaning of a sentence is represented by a constant displacement. The consistency of the semantic contribution of a given word w is defined as follows. For all sentences in the test set that contain w, take the last hidden state h_n of their encoding. Then take a version of each sentence with w removed, and take its last hidden state h'_{n-1}. The vector offset of the ith sentence that contains w is defined as $o_i = h_n - h'_{n-1}$, where n is the length of sentence i. Then the directional consistency θ_w of a word w is defined as the average pairwise cosine similarity for all o (Eq. 1), and magnitude consistency ι_w is defined as the negative of the average pairwise absolute difference in Euclidean norms for all o (Eq. 2), where N is the total number of sentences containing w.

$$\theta_w = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{o_i \cdot o_j}{\|o_i\| \|o_j\|}}{N(N-1)} \qquad (1)$$

$$\iota_w = -\frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \big| \|o_i\| - \|o_j\| \big|}{N(N-1)} \qquad (2)$$

This constant linear displacement hypothesis is motivated by empirical observations such as [7], where systematic compositional contributions were found to be encoded as consistent vector offsets. Tensor Product Representations [30] formally generalizes this intuition, representing symbolic structures as linear sums of filler and role bindings. For a more detailed discussion, see Appendix B.

3 Experiments

Models. We used a Siamese recurrent classifier architecture similar to [6], in which the same recurrent network is used to encode p and h, and the concatenated encodings (we took the last hidden state) of the two sentences are passed to the classification layer as in [16]. We used AllenNLP [31] to implement our models.[1] For the recurrent units, we tested SRNs and GRUs with a single hidden layer. The input dimension was fixed to 300, and the hidden dimension of the recurrent units was varied between $h = \{8, 16, 32, 64, 128, 256, 512\}$. Word embeddings were initialized using Xavier initalization, the default setting in AllenNLP. The classifier was a single feedforward layer with linear activation followed by a softmax, which takes $2h$ dimensional inputs.

Training. We trained models on the entailment dataset for a maximum of 50 epochs using stochastic gradient descent (learning rate=0.1, batch size=16), early stopping when the development set accuracy did not improve for 5 epochs. In practice, most models reached peak development accuracy within 10 epochs. We ran each model with the same hyperparameters with 10 different random initializations.

Behavioral results. Both SRN and GRU models were able to learn the train/development sets perfectly, with small variations across random initializations (SRN: 0.96 (± 0.05) (train), 0.98 (± 0.03) (dev), GRU: 0.99 (± 0.01) (train), 0.9999 (± 0.0001) (dev)). However, SRN and GRU models significantly differed in their generalization accuracy (Mann-Whitney $U = 3231, p < .001$)—GRU models on average achieved near-perfect accuracy (0.97), whereas SRNs did not (0.69). No single SRN model generalized perfectly (highest accuracy = 0.87).

[1] Our code is available at `https://github.com/najoungkim/compnet`.

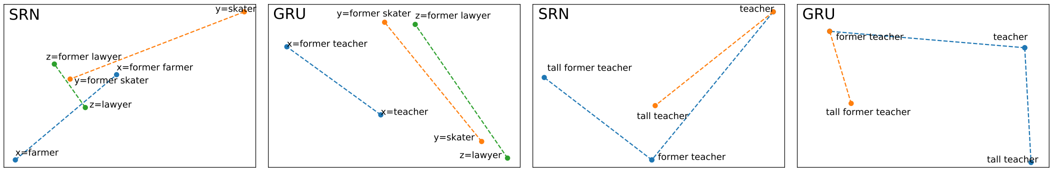

Figure 1: Directional consistency of adjectives in SRN and GRU models.

Representational analysis results. GRU models encoded adjectives' compositional contributions with higher directional consistency ($U = 3025, p < .001, |\Delta| = 0.29$) (see Figure 1 for an illustration). The difference in magnitudes of the adjectives' compositional contributions were more similar to each other in GRUs than SRNs ($U = 119, p < .001, |\Delta| = 1.47$) . Within GRU models, we found a significant correlation between generalization set accuracy and directional consistency of adjective encodings (Pearson's $r = 0.54, p < .001$), but not between accuracy and magnitude consistency ($r = 0.00, p = .99$). Within SRNs, we observed an inverse correlation between directional consistency and accuracy ($r = -0.69, p < .001$). This effect was largely driven by a cluster of models that had below majority-class accuracy (< 0.69) (see Figure 2, far left). The inverse correlation no longer holds if we exclude models in this cluster ($r = 0.32, p > .05$ after multiple-comparisons correction).

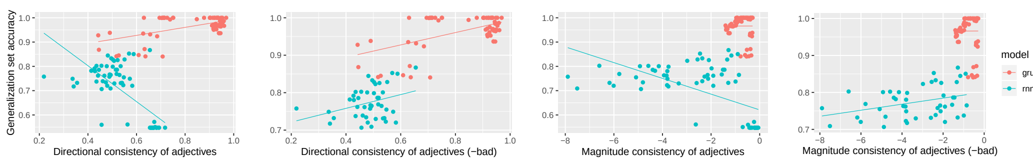

Figure 2: Accuracy plotted against consistency measures with the line of best fit by model group. Additional plots are shown for data excluding models with accuracy below majority-class.

Table 2: Pearson's correlation between consistency measures and generalization set accuracy. The p-values are adjusted using Holm-Sidak correction. ($^{*} = p < .05, ^{**} = p < .01, ^{***} = p < .001$)

Model	#	Accuracy	Corr(acc, dir.)			Corr(acc, magn.)		
			All	Adj.	N	All	Adj.	N
SRN	70	0.69 ($\pm$0.11)	0.21	-0.69^{***}	0.44^{**}	-0.59^{***}	-0.59^{***}	-0.53^{***}
GRU	70	0.97 ($\pm$0.04)	0.52^{***}	0.54^{***}	-0.12	0.10	0.00	0.41^{*}

Our findings can be summarized as follows. SRNs and GRUs both could learn the training data perfectly, but their capacity to make systematic generalizations differed greatly. GRUs encoded the contribution of adjectives to the sentence in a more geometrically consistent manner, with respect to both direction and magnitude of the linear offsets. Within GRU models (but not within SRNs), models in which the contribution of adjective was encoded as directionally consistent offsets had higher generalization accuracy. This finding does not seem to be an artifact of the dataset that we used; a follow-up experiment using SCAN [5] showed similar trends (see Appendix A).

4 Conclusion

We investigated the difference between SRNs and GRUs in their capacity to make compositional semantic generalizations. Our results suggest that SRNs and GRUs employ qualitatively different approaches for solving the same task, and the strategy GRUs adopt proves more effective for making systematic generalizations. Furthermore, we observe that the representations GRUs develop display more geometric regularity across different linguistic contexts, measured by the average direction and magnitude consistency of the compositional contributions of the adjective. Directional regularity in particular seems to facilitate systematic generalization for GRUs, suggested by the significant within-GRU correlation between directional consistency and generalization accuracy.

What is the nature of the architectural bias that gives rise to this discrepancy? One insight can be drawn from [32], which makes an empirical remark about the importance of a forget gate. We could speculate that the forgetting mechanism encourages models to discard contextual information (if it is useful to do so), biasing models towards developing more globally invariant representations of lexical items. Exploring this hypothesis further would be an interesting follow-up, elucidating the roles of different architectural components in representing compositionality. More broadly, we plan to investigate whether we could inject bias into the models for learning more compositionally generalizable respresentations, and extend the scope of our work to more naturalistic datasets.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[2] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[3] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NeurIPS Workshop on Deep Learning, December 2014*, 2014.

[4] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75, Aug 2018.

[5] Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2879–2888, 2018.

[6] Mathijs Mul and Willem Zuidema. Siamese recurrent networks learn first-order logic reasoning and exhibit zero-shot compositional generalization. *arXiv:1906.00180*, 2019.

[7] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.

[8] Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes. Proceedings of the 2019 ACL workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2019.

[9] Jacob Andreas. Measuring compositionality in representation learning. In *International Conference on Learning Representations*, 2019.

[10] Joris Baan, Jana Leible, Mitja Nikolaus, David Rau, Dennis Ulmer, Tim Baumgärtner, Dieuwke Hupkes, and Elia Bruni. On the realization of compositionality in neural networks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 127–137, Florence, Italy, August 2019. Association for Computational Linguistics.

[11] Kris Korrel, Dieuwke Hupkes, Verna Dankers, and Elia Bruni. 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, August 2019. Association for Computational Linguistics.

[12] Brenden M Lake. Compositional generalization through meta sequence-to-sequence learning. *arXiv:1906.05381*, 2019.

[13] Yonatan Belinkov and James Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics (TACL)*, 7:49–72, 2019.

[14] Brendan Whitaker, Denis Newman-Griffis, Aparajita Haldar, Hakan Ferhatosmanoglu, and Eric Fosler-Lussier. Characterizing the impact of geometric properties of word embeddings on task performance. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 8–17, Minneapolis, USA, June 2019. Association for Computational Linguistics.

[15] Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. The emergence of number and syntax units in LSTM language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 11–20, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[16] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[17] Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[18] Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[19] Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. Probing what different NLP tasks teach machines about function word comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 235–249, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[20] Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[21] Rebecca Marvin and Tal Linzen. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[22] Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. Studying the inductive biases of RNNs with synthetic variations of natural languages. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3532–3542, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[23] Samuel R. Bowman, Christopher Potts, and Christopher D. Manning. Recursive neural networks can learn logical semantics. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 12–21, Beijing, China, July 2015. Association for Computational Linguistics.

[24] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[25] Richard Evans, David Saxton, David Amos, Pushmeet Kohli, and Edward Grefenstette. Can neural networks understand logical entailment? In *International Conference on Learning Representations*, 2018.

[26] Sara Veldhoen and Willem Zuidema. Can neural networks learn logical reasoning? *CLASP Papers in Computational Linguistics*, page 34, 2017.

[27] Ellie Pavlick and Chris Callison-Burch. Most “babies" are “little" and most “problems" are “huge": Compositional entailment in adjective-nouns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2164–2173, 2016.

[28] Barbara Partee. Lexical semantics and compositionality. *An invitation to cognitive science: Language*, 1:311–360, 1995.

[29] Neha Nayak, Mark Kowarsky, Gabor Angeli, and Christopher D Manning. A dictionary of nonsubsective adjectives. Technical report, 2014.

[30] Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, 46(1-2):159–216, 1990.

[31] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[32] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, pages 2342–2350, 2015.

[33] Paul Soulos, Tom McCoy, Tal Linzen, and Paul Smolensky. Discovering the compositional structure of vector representations with role learning networks. *arXiv:1910.09113v1*, 2019.

[34] R Thomas McCoy, Tal Linzen, Ewan Dunbar, and Paul Smolensky. RNNs implicitly implement tensor-product representations. In *International Conference on Learning Representations*, 2019.

A SCAN Experiment

We extend our offset consistency analysis to models trained on SCAN, a dataset designed to test for compositional generalizations. The goal of the task is to map a (simplified) natural language command sequence to a corresponding action sequence. The commands are generated by a phrase-structure grammar, and the command-to-action mapping is determined by a set of compositional rules [5].

Dataset. We used the split of the SCAN dataset that tests for compositional generalization across primitive commands. In this split, the command *jump* is only shown in its primitive form or in a limited number of compositional contexts (Experiment 3 in [5]). We chose this split for two reasons: (1) the clear difference in train-test distribution (the test set is a generalization set), and (2) the availability of different replication splits.[2] We used splits with a varying number of compositional examples shown in training ($n \in \{4, 8, 16, 32\}$, where n denotes the number of compositional commands given in the training set). We did not use $n \in \{0, 1, 2\}$ because models almost completely failed to generalize on these splits, as was reported in [5]; as such, there was no interesting variance across models in terms of generalization set accuracy.

Models and training. We used a GRU encoder-decoder architecture, treating the command-to-action translation as a sequence-to-sequence mapping task. We used AllenNLP [31] to implement our models. We used an input of dimension 100 and a single hidden layer of dimension 100, with a dropout rate of 0.1 following [33]. The bottleneck embedding was the last hidden state of the encoder. Following [5], we used the Adam optimizer with a learning rate of 0.001, clipping gradients with a norm larger than 5.0. For training the decoder, teacher-forcing was applied 50% of the time, again following [5]. Each model was trained for 30 epochs with a batch size of 128. 15 models with different random initializations were trained for each of the 5 replication splits for each $n \in \{4, 8, 16, 32\}$, giving us 75 models for each n and 300 models in total.

Table 3: Pearson's correlation between consistency measures and generalization set accuracy for GRU models trained on SCAN. The p-values are adjusted using Holm-Sidak correction. ($^{*} = p < .05$, $^{**} = p < .01$, $^{***} = p < .001$) Columns labeled Dir. and Magn. list the mean direction and and magnitude consistency, respectively.

Model	#	Accuracy			Dir.	Corr(acc, dir.)	Magn.	Corr(acc, magn.)
		Mean ($\pm\sigma$)	Min.	Max.	Modifiers		Modifiers	
$n = 4$	75	0.01 ($\pm$0.01)	0.00	0.03	0.37	0.33**	−1.28	0.29*
$n = 8$	75	0.03 ($\pm$0.02)	0.01	0.12	0.39	0.41***	−1.23	0.17
$n = 16$	75	0.15 ($\pm$0.09)	0.02	0.43	0.41	0.27*	−1.16	0.31*
$n = 32$	75	0.48 ($\pm$0.10)	0.27	0.70	0.42	−0.26*	−1.10	−0.26*
All	300	0.17 ($\pm$0.20)	0.00	0.70	0.40	0.50***	−1.19	0.44***

Representational analysis results. Table 3 shows the models' generalization accuracy, and the correlation between generalization accuracy and the average offset consistency over the modifiers in the encoder-side vocabulary (the most analogous setup to our main adjective experiments). Accuracy is measured by the percentage of full-string matches in the generalization set. Aggregating over all models, there was a significant positive correlation between generalization accuracy and directional consistency ($\rho = 0.50, p < .001$), and between generalization accuracy and magnitude consistency ($\rho = 0.44, p < .001$). However, n itself was correlated with both measures; as n increases, the directional consistency of the modifiers increases ($\rho = 0.55, p < .001$) and the magnitude consistency also increases ($\rho = 0.49, p < .001$). A within-n correlation analysis reveals that the trend of more consistent offsets leading to better generalization accuracy depended on n. As can be seen from Figures 3 and 4, for $n \in \{4, 8, 16\}$ the consistency-accuracy correlation holds (except for $n = 8$ for which the magnitude correlation is not significant), but we found an opposite trend for $n = 32$. One possible explanation is that, as n increases, the train-generalization set distributions become increasingly similar to each other. If the train and test distributions are similar, representations that

[2] `https://github.com/brendenlake/SCAN/blob/master/add_prim_split/with_additional_examples`

are more specifically tuned to particular contexts in the training set (e.g., the same word showing more idiosyncrasy across different contexts) could be beneficial at test time, even if they are less compositionally generalizable. Note that the mean directional and magnitude consistency did increase with larger n.

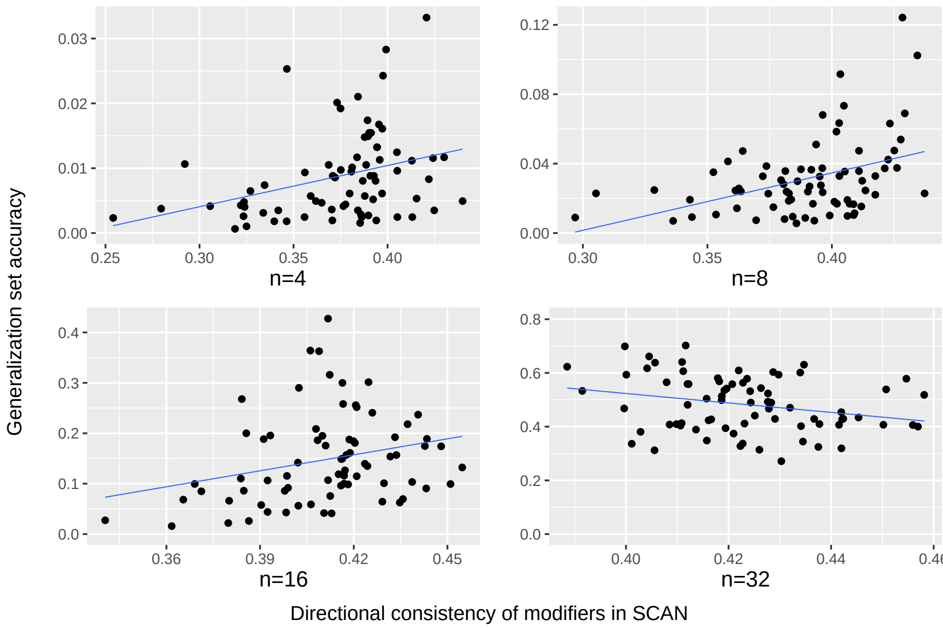

Figure 3: Directional consistency of modifiers in GRU models trained on SCAN. Note the variability in y axis scales across different n.

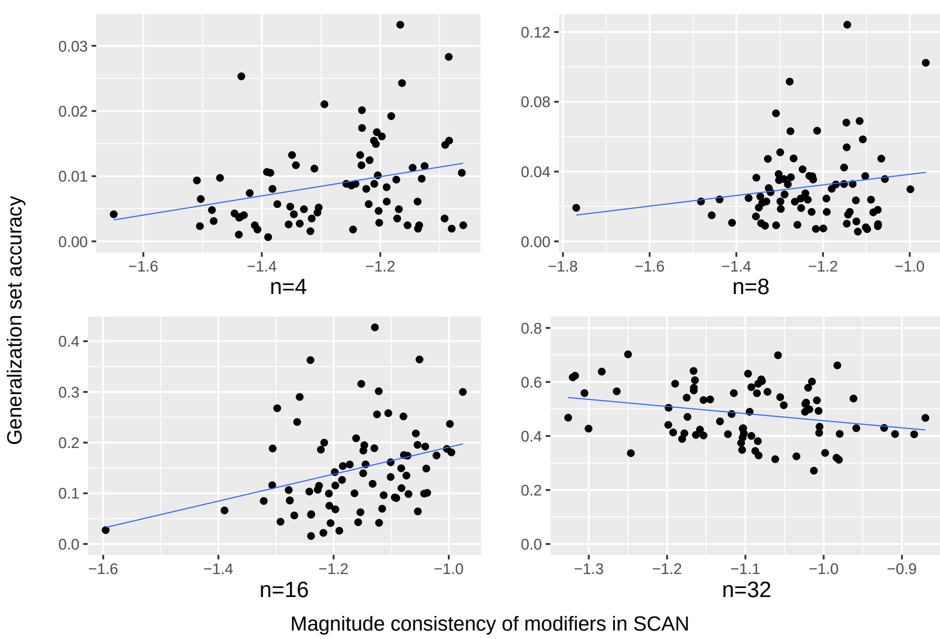

Figure 4: Magnitude consistency of modifiers in GRU models trained on SCAN. Note the variability in y axis scales across different n.

B Connection between geometric consistency measures and Tensor Product Representations

It has been empirically argued that representations learned by neural networks can encode systematic compositional contributions via consistent vector offsets, as illustrated by the well-known example $\textit{king} - \textit{man} + \textit{woman} \approx \textit{queen}$ from [7]. Tensor Product Representations (TPRs) [30] provide a more explicit formal generalization for this observation; that is, the representation of a symbolic structure is a linear sum of filler and role bindings. Filler and role representations are both vectors, and their binding is the outer (tensor) product of these two vectors. For the *king/queen* analogy example, the meaning of *king* and *queen* may be encoded as a linear sum of filler-role bindings, where the roles (which correspond to lexical features in this example) involved are `gender` and `status`. The meaning of the two words only differ by the fillers that are bound to `gender`. Under this analysis, the observed vector offset consistency is transparently predicted: (`male` $\otimes$ `gender` + `royal` $\otimes$ `status`) - (`male` $\otimes$ `gender`) = (`female` $\otimes$ `gender` + `royal` $\otimes$ `status`) - (`female` $\otimes$ `gender`).

Recent works have shown that sequential neural networks such as GRUs do develop representations that approximate TPRs [34, 33]. Under a TPR formulation, our compositional generalization task for adjectives semantics would be equivalent to learning the correct subsectivity (role) for each adjective (filler). If a network learns to assign correct filler-role bindings for all adjective uses (e.g., *tall'* = `tall` $\otimes$ `subsective`, *former'* = `former` $\otimes$ `nonsubsective`, it should be able to solve the generalization set, since the required inference relies on the subsectivity of the adjectives (and the meanings do not need to contextually vary for this particular task). In such a network, the vector offset between a sentence that contains `Adj` and a sentence with `Adj` removed is expected to be `Adj` $\otimes$ `subsectivity` (or a vectorized version of this matrix) across all sentences that contain `Adj`. Since vectors are defined by their direction and magnitude, a consistent offset in terms of direction and magnitude signals a more compositionally useful representation for this task.

The SCAN experiment in Appendix A suggests that consistency of vector offsets continues to be useful in a setting that requires a more complex compositional reasoning. We suggest a possibility that the offset consistency functions as proxies for *role stability* across different constructions, which facilitate compositional generalization. [33] provides a comprehensive case-by-case analysis of the role scheme that achieves near-perfect accuracy on the vanilla split of SCAN (where the train/test sets are mutually exclusive subsets of the same distribution). Often these roles are very specific (i.e. highly context-dependent), which is likely a byproduct of a specific subset of examples in the training set rather than a reflection of their usefulness in out-of-domain generalization. For instance, *after* gets assigned role 17 if no other word has role 17 or if the command after *after* ends with *around left*, and gets assigned role 43 otherwise. Such idiosyncratic roles, which are likely artifacts of the training data, could explain the degradation in surgery accuracy over multiple substitutions that [33] reports. That is, changing the filler (e.g., substituting `left:36` with `right:36`) may trigger role changes even for other unmodified elements, which would result in a failed surgery step. In an ideal compositional model this would not happen—the roles of the unmodified elements would be stable. Not only in the surgery context but also more generally, stable roles for the same lexical item (or primitives) over multiple constructions would be more compositionally generalizable, especially when we are using out-of-domain test sets as in the split used in Appendix A. The offset consistency as shown in Appendix A could be a signal of role stability (since the offsets would be the more consistent when the roles are invariant to word removal), which could help generalization. We hope to investigate the relation between role stability and compositional generalizability more explicitly in future work.