Proceedings of Machine Learning Research 140:138–148, 2021 AAAI Workshop on Meta-Learning and MetaDL Challenge

Challenges of Acquiring Compositional Inductive Biases via Meta-Learning

Eric Mitchell ERIC.MITCHELL@CS.STANFORD.EDU
Chelsea Finn CBFINN@CS.STANFORD.EDU
Chris Manning MANNING@STANFORD.EDU
Stanford University

Editors: Isabelle Guyon, Jan N. van Rijn, Sébastien Treguer, Joaquin Vanschoren

Abstract

Meta-learning is typically applied to settings where, given a distribution over related training tasks, the goal is to learn inductive biases that aid in generalization to new tasks from this distribution. Alternatively, we might consider a scenario where, **given an inductive bias**, we must **construct a family of tasks** that will inject the given inductive bias into a parametric model (e.g. a neural network) if meta-training is performed on the constructed task family. Inspired by recent work showing that such an algorithm can leverage meta-learning to improve generalization on a single-task learning problem, we consider various approaches to both a) the construction of the family of tasks and b) the procedure for selecting support sets for a particular single-task problem, the SCAN compositional generalization benchmark. We perform ablation experiments aimed at identifying when a meta-learning algorithm and family of tasks can impart the compositional inductive bias needed to solve SCAN. We conclude that existing meta-learning approaches to injecting compositional inductive biases are brittle and difficult to interpret, showing high sensitivity to both the family of meta-training tasks and the procedure for selecting support sets.

1. Introduction

Compositional generalization, also referred to as systematic or combinatorial generalization, is the challenge of generalizing to situations when familiar concepts are combined in unfamiliar ways. Humans are typically able to generalize in this manner (Franklin and Frank, 2019), a capability that largely still eludes modern machine learning systems (Klinger et al., 2020). As a simple example of compositionality, given that a human knows how to "walk" and how to "walk quickly", once they learn to "cook" or "run", it is generally believed that they should be able to immediately understand the concepts "cook quickly" and "run quickly" by combining the old concept of "quickly" with the (now familiar) concept "cook" or "run." However, current machine learning models, especially neural networks, do not demonstrate this ability to generalize compositionally, requiring a relatively dense sampling of the set of possible combinations of atomic concepts ("walk", "run", "cook", "quickly"). For problems with narrow scope, this limitation can make data collection somewhat costly but manageable; for more open-world domains, such as a robot that must navigate and interact with the general public in the outdoors, this set of possible "composite experiences" grows impractically large. Thus an important challenge for machine learning and artificial intelligence

researchers is to better understand these compositional mechanisms so that we might ultimately endow our models with them.

Several recent studies have shown that compositional generalization on realistic tasks will require stronger inductive bias than is offered by current standard deep learning architectures and training objectives (Lake and Baroni, 2018; Hupkes et al., 2020; Klinger et al., 2020; Bahdanau et al., 2019). Research that aims to endow neural networks with the ability to generalize compositionally has thus drawn on many of the traditional strategies for injecting inductive bias into deep networks, including specialized architectures, data augmentation strategies, regularization schemes, and customized objectives through meta-learning. All of these approaches have shown some success on some (but not all) of the various benchmarks for testing compositional generalization, but we suggest that meta-learning based approaches might be particularly promising because they are (at least conceptually) architecture agnostic. While experts in deep learning have spent significant energy developing custom architectures that perform very well for particular tasks, the level of expertise required to tailor an architecture to a task can be very high, providing a barrier to many people or organizations hoping to apply deep learning to their problem(s). On the other hand, meta-learning approaches to injecting inductive bias can use relatively standard architectures and pass in domain knowledge by specifying *a family of tasks* that captures the 'types of biases' a model should learn. In general, stating a family of tasks can be a far more *intuitive* process, with a lower barrier to entry, than experimenting with architectural adjustments (McCoy et al., 2020). This possible future, in which practitioners can select an architecture from a handful of standard architectures (conv-nets, sequence models, etc) and adapt it to their problem by specifying *families of related tasks* rather than excessive architectural experimentation, is the motivating premise for exploring meta-learning as a tool for injecting inductive biases even in single task settings.

In this work, we study a compositional generalization problem to better understand the process of applying meta-learning to a single-task problem with the goal of acquiring a particular inductive bias. In particular, we examine recent work applying few-shot learning to the SCAN compositional generalization benchmark (Lake, 2019), which tests a natural language model's ability to generalize in systematic ways such as adding a new verb to its vocabulary from only a single example. Our extended ablations study how design choices of a) the family of tasks used for meta-training and b) the strategy used to select test-time support sets affect whether or not the model will actually perform well on the meta-test task(s), finding that sometimes surprisingly small changes in these factors have outsized effects on meta-test set accuracy. Our experiments lead us to conclude that acquiring inductive biases for single-task problems via meta-learning in a family of generated tasks is a promising direction, but significant further research is required to better understand the mechanisms by which these approaches work and in what settings they might be applicable.

2. Using Meta-Learning to Acquire Compositional Inductive Biases

As indicated in recent work, there are many sub-types of compositional inductive biases (Hupkes et al., 2020). Compositional benchmarks generally consist of several train/test splits of the data in order to provide compositional generalization problems corresponding to some or all of these different "types" of generalization. The vast majority of the benefits of existing methods have been observed

Command Sequence	Target Sequence
run	RUN
look	LOOK
jump	JUMP
run twice	RUN RUN
look after walk	WALK LOOK
jump twice	JUMP JUMP
jump after run twice	RUN RUN JUMP
jump around right	RTURN JUMP RTURN JUMP RTURN JUMP RTURN JUMP

Table 1: Example commands from the SCAN command following dataset. The top section includes example commands from the train split; the bottom section shows example commands from the 'add primitive' test split. The primitive command 'jump' is the only jump command present in the train set; all test commands use the verb jump at least once.

on a sub-type of compositionality that Hupkes et al. (2020) refer to as *systematicity*, that is, problems that amount to re-combining familiar atomic concepts into novel composite concepts; however, other notions of compositional generalization exist, such as *productivity*, also referred to as "making infinite use of finite means", which in practice means creating or processing inputs at test time that are larger in some sense than those seen during train time, but still produced using the same atomic set of rules. A concrete example of a systematic compositional generalization split is the 'add primitive' split in the SCAN command-following dataset (Lake and Baroni, 2018). See Table 1 for examples from this task. We focus our experiments on this setting.

2.1. SCAN compositional generalization benchmark

In SCAN, a model is presented with a natural language command such as 'walk twice' and must output the correct sequences of actions (`WALK WALK`, in this case); SCAN can be interpreted as either a seq2seq translation problem or an open loop control problem, where the command is the initial condition of the system. We refer the reader to prior work for more examples of commands and target action sequences in SCAN (Lake and Baroni, 2018). The SCAN vocabulary is simple, containing only 4 different verbs as well as several adverbs and connectives. However, it is rich enough to construct challenging compositional generalization splits, one of which being the **add primitive** split. In this split, the train set contains the primitive command 'jump' (simply mapped to the action `JUMP`) as well as all composite commands that *do not* use the verb 'jump'. All 'jump' composite commands are held out for the test set. This split is challenging, but in principle not impossible due to the symmetry of the grammar with respect to different verbs. This symmetry ('_verb_' → `VERB`, '_verb_ twice' → `VERB VERB`, '_verb1_ and _verb2_' → `VERB1 VERB2`, etc.) is explained in detail by (Gordon et al., 2020), is the inductive bias that we need to inject into the model in order to achieve good generalization performance on the test set.

2.2. Meta seq2seq (Lake, 2019)

Meta seq2seq is a meta-learning approach (specifically, a few-shot formulation) to injecting the verb-symmetry inductive bias described above. The author defines a family of tasks $\{\mathcal{T}\}^i$, which includes the original SCAN task, and trains a few-shot learning model on this family. The original SCAN task (the add primitive split) is held out as the meta-test task. Each task is defined by a pair of permutations of the verbs (π_1^i, π_2^i). The data for task i is generated by applying these permutations to the *verbs* in the command and target sequence of each data point in the original dataset. Abusing notation slightly, we have $(x_k^i, y_k^i) = (\pi_1^i(x_k), \pi_2^i(y_k))$, where (x_k, y_k) is the k-th data example in task i. The original task (the single-task SCAN add primitive split) corresponds to the task where both permutations are the identity permutation; this task is held out as the (single) meta-test task. To formalize slightly, we define two sets of words, W_c and W_t, which are the words that may be permuted in the command and target, respectively. In meta seq2seq, we have $W_c = \{walk, run, look, jump\}$ and $W_t = \{WALK, RUN, LOOK, JUMP\}$. One immediate result of this selection of W_c and W_t is that *the model is exposed to all possible command and target sequences during training*, because composite commands that do not use the word 'jump' will be transformed to form the corresponding 'jump' command, even though composite 'jump' commands were not present in the original training set. For example, for some task that maps the command word 'walk' to the command word 'jump', the command *walktwice* would be mapped to the command *jumptwice*; a similar phenomenon occurs for target sequences. Thus, by selecting this family of tasks, meta seq2seq reduces the compositional generalization problem into an IID learning setting; this likely explains why meta seq2seq is effective on the add primitive split and not other splits, such as the length generalization split (training on examples with labels of length less than N, testing on examples with labels of length greater or equal to N). During meta-training, meta seq2seq samples support sets uniformly from the current tasks' data; during meta-testing (for which there is only one task, the original SCAN task), the support set uses only the four primitive verb commands.

By selecting different sets of words for W_c and W_t, we can in some sense generalize meta seq2seq to a much broader class of compositional symmetries. This is useful to evaluate the robustness of the meta seq2seq procedure to poorly-specified or unknown symmetries in the data. Assessing this capability is important for several reasons. First, when applying this meta-training procedure to real-world tasks, it is unlikely that we will be able to specify the desired inductive bias as succinctly and accurately as can be done in SCAN. Additionally, even if we can articulate the desired inductive bias, applying the transformations to the original dataset needed to produce the meta-training tasks might be difficult or unknown. Finally, an idealized approach to using meta-learning for injecting inductive biases would involve a partially or fully automated approach to generating tasks for meta-training, which would likely be prone to imperfect task family specification. Thus, any inductive bias injection algorithm based on meta-learning should still provide some benefits if the task family specified only partially captures the desired inductive bias.

3. Experiments

Our experiments are intended to assess the impacts of a) the family of tasks used for meta-training (Table 2) and b) the meta-train and meta-test support selection proce-

Experiment	W_c	W_t	Train Acc.	Test Acc.
A	Verbs	Verbs	100%	99.7%
B	Verbs - {jump}	Verbs - {JUMP}	100%	0%
C	Verbs	None	100%	0%
D	None	Verbs	100%	50.0%
E	Verbs - {walk}	Verbs - {WALK}	100%	97.0%
F	{run, look, jump}	{LOOK, JUMP}	100%	100%
G	{look, jump}	{LOOK, JUMP, RUN}	100%	0%
H	{look, jump}	{LOOK, JUMP}	100%	0%
I	Verbs + {and, then}	Verbs	96.1%	94.3%
J	All	Verbs	92.5%	84.2%
K	All	All	11.1%	0%

Table 2: Train and test performance of our re-implementation of Meta seq2seq (Lake, 2019) when various different groups of command and target sequence words are included in the permutation sets W_c and W_t. The original train set contains only a single command using the word 'jump'; some meta-training task generation procedures (e.g. A, C) ultimately generate additional usages of the command word jump during training, while others do not (e.g. B, D). The model can perfectly fit the generated family of training tasks in almost all settings, but in many cases does not generalize to the SCAN 'add primitive' test split.

dure (Table 3) on the ability of a model to learn the desired inductive bias. We evaluate variants of the meta seq2seq regime presented in Lake (2019) on the 'add primitive' split of the SCAN benchmark (Lake and Baroni, 2018), using our own re-implementation of the architecture described in Lake (2019).

3.1. Specifying the meta-training task family

Section 2.2 explains one approach to specifying a family of tasks over which to perform meta-training in order to acquire the compositional inductive bias: permutation-based task generation with W_c and W_t equal to the verbs in the command and target sequences, respectively. In this group of experiments, we investigate the extent to which this compositional inductive bias is acquired when this task family is perturbed. This is intended to assess the extent to which a procedure like meta seq2seq might be viable in settings where we either can't exactly specify the desired inductive bias, generating the data for some of the tasks in the desired task family is infeasible, or our procedure for generating the meta-training task data is noisy. **Experiment A** begins by replicating the results from Lake (2019). In **Experiment B**, we follow the meta seq2seq procedure, but only permute the non-jump verbs. Recall that in the add primitive SCAN split, our goal is to correctly predict the action sequence for commands using the verb jump, when we have only been exposed to the word jump in a single context, the primitive command 'jump' → `JUMP`. One might hope that in this setting (permuting only words other than 'jump'), we might achieve a 'best of both worlds' solution, where we aren't required to actually synthesize new data as in

the original meta seq2seq procedure (assuming all possible usages of the non-jump verbs are in the training set), but we can still acquire the inductive bias needed to generalize to the test problem. However, we find that this isn't the case; this procedure yields a model with zero test task accuracy.

In **Experiments C and D**, we assess whether the meta seq2seq procedure learns a useful inductive bias when meta-trained only on tasks where either the command or target verb words (but not both) are permuted, respectively. Interestingly, the performance is markedly different in both cases (and consistent across random seeds). Meta-training on tasks generated from only permuting command words produces essentially no useful inductive bias; however, generating tasks by only permuting target words yields non-trivial performance on the test task. These results are in fact reasonable because Experiment C is a *concept drift* problem; at test time, the model sees familiar inputs (the model was exposed to all jump *commands* during training), but with ground truth labels never seen during training. Thus the model has been explicitly optimized during meta-training to assign low probability to exactly the labels we would like the model to output during meta-testing. Thus it is unsurprising that the model achieves essentially zero accuracy during meta-testing (the original SCAN task). On the other hand, Experiment D can be seen as a *domain shift* problem, in which we are assessed on previously unseen commands (the model was exposed only to the jump *target sequences* during training here), but the meta-training procedure hasn't entailed explicitly optimizing the model to assign the correct labels low probability for these commands (which it has in the concept drift case); thus there is a chance that the model will achieve some generalization simply by way of the inductive bias of the neural network. **Experiment E** tests the simple setting where we omit one non-jump verb from the permutations used to generate the meta-training task family; in this case, we still see good train and test performance, indicating that some misspecification of the desired symmetry can be tolerable. This result is significant because in realistic settings, specifying the desired inductive bias exactly might be very difficult or impossible.

Experiments F, G, and H compare three similar settings, differing by whether the set of words permuted in the task family includes 'run' for either the command words, the target words, or neither. Based on the results of Experiments C and D, we might expect that it is *more important* for the model to be exposed to more tasks that permute the target words, as permuting only the verbs in the target sequence performed far better than permuting only the verbs in the input command. However, we actually see the opposite effect here: including three verbs in W_c and two verbs in W_t gives perfect performance on both train and test (Experiment F), while the opposite scenario (Experiment G) fits the train set but does not acquire the necessary inductive bias to generalize at test time. Similarly, including only two verbs in W_c and W_t is not effective. Finally, **Experiments I, J, and K** examine the extent to which meta-training on *more general* task families than the 'minimal' family needed to acquire the desired inductive bias can also be effective. Somewhat surprisingly, this is the case for both the case where W_c includes the verbs *and* the connectives {and, after} and when W_c includes all of the words in the command vocabulary. However, generating tasks by simply including all words in the command vocabulary in W_c and all words in the target vocabulary in W_t produces a task distribution too difficult for the model to fit. Nonetheless, these results indicate that meta-training on families of tasks

Experiment	Perm Group	Train Sup.	Test Sup.	Train Acc.	Test Acc.
1	Verbs	Uniform	Primitives	99.8%	99.4%
2	Verbs	Uniform	Uniform	99.8%	84.7%
3	Verbs	Uniform	Safe	99.7%	100%
4	Verbs	Safe	Safe	99.7%	99.9%
5	Verbs + Connectives	Uniform	Uniform	73.5%	48.3%
6	Verbs + Connectives	Safe	Safe	97.0%	94.6%

Table 3: Comparison of different approaches to selecting the support set in meta seq2seq. 'Uniform' corresponds to sampling support data uniformly at random from the train set; 'Primitives' means using only the 4 primitive verb commands as a support set; 'Safe' means sampling randomly from the train set, subject to the constraint that each word in the query command appears in at least one support set command. Both the 'Primitive' approach (used in (Lake, 2019)) and the 'Safe' approach are effective; however, the 'Safe' sampling approach requires less task-specific knowledge (it doesn't require knowing which examples are the 'primitives' for a particular problem), which is advantageous.

that are only loosely derived from the desired compositional inductive bias can still be very useful, which is exciting, as this loose relationship between task family and desired inductive bias will probably be the case in practical settings.

3.2. Selecting a support set

In general, we might apply meta-learning to acquire an inductive bias for a *single-task* learning setting; this is the case for meta seq2seq as well as other applications of meta-learning to acquiring inductive biases (McCoy et al., 2020). However, if we formulate a single-task problem into a few-shot learning problem, for example (as has been done in the previous work mentioned), in addition to specifying tasks, we must also specify *how to sample support sets for adaptation*. In this section, we perform a batch of experiments aimed at assessing the extent to which different support set selection procedures might lead to more or less effective inductive bias acquisition.

Experiment 1 corresponds to the meta seq2seq procedure, where support sets are chosen uniformly from the train set at train time and only the four primitive verb commands are used as a support set at test time. In **Experiment 2**, we eliminate the assumption that we have knowledge of the 'basis commands' and use uniform sampling from the train set. This leads to noticeably worse test set generalization. **Experiment 3** uses a 'safe' support set selection procedure at test time, which is similar to uniform sampling but enforces the constraint that all words used in the query commands are represented in the support commands; this noticeably improves performance. **Experiment 4** uses the safe support set selection procedure during both train and test. **Experiments 5 and 6** are analogous to Experiments 2 and 4, respectively, except we permute the verbs *and* the connectives {and, after} together, which is more difficult. Because W_c contains the connectives, there are not analogous 'primitive' commands as there were in the simple verbs-only setting. The gap in

generalization performance between uniform and safe sampling is even larger, highlighting the importance of picking the right support set. Ultimately we conclude that sampling the support set in a manner that consistently provides sufficient information to unambiguously infer the task is critical to effectively capturing the desired inductive bias with meta-learning, and the 'safe' support set strategy, at least for SCAN, allows us to do so without relying on the task-specific knowledge used with the 'primitive' strategy used by Lake (2019). From these results and those from the previous section, we conclude that it is possible to acquire the desired compositional inductive bias using slightly misspecified *task families* as well as quite general purpose *support set selection procedures*.

4. Related Work

The challenge of training neural networks to discover compositional (or systematic) inductive biases is well known and has received significant attention in recent work. Hupkes et al. (2020) propose a taxonomy of different sub-types of compositional generalization for the purposes of more precisely identifying model failure modes. Other work has focused on empirical evaluation of the compositional generalization ability of 'traditional' neural network (particularly seq2seq (Sutskever et al., 2014)) models in language-based (Lake and Baroni, 2018; Klinger et al., 2020; Keysers et al., 2020; Lake, 2019), visual (Bahdanau et al., 2019), and mixed grounded language (Ruis et al., 2020; Gao et al., 2020) settings. Having identified this shortcoming, the community has produced several diagnostic problems to more precisely test the compositional generalization capabilities of neural networks in settings such as visual reasoning (Johnson et al., 2017), language-only command following (Lake and Baroni, 2018), grounded command following (Ruis et al., 2020; Chevalier-Boisvert et al., 2019), question answering (Keysers et al., 2020), and translation (Hupkes et al., 2020).

New benchmark suites have stimulated the development of bespoke neural network architectures for various compositional problems. Proposed approaches exploit attention and message passing (Andreas et al., 2016; Johnson et al., 2017; Lake, 2019; Gao et al., 2020), utilize new recurrent architectures (Hudson and Manning, 2018), or take a program synthesis approach (Nye et al., 2020; Hudson and Manning, 2019). In language settings, some work has attempted to more explicitly disentangle the structural and semantic roles of particular words (Russin et al., 2019; Li et al., 2019). Another approach uses group-equivariant architectures to learn models robust to particular types of compositional transformations (Gordon et al., 2020). In addition to architectural approaches to compositional generalization, targeted data augmentation (Andreas, 2020) has proved effective in some situations. Other approaches have begun exploring meta-learning as a way to inject inductive biases into a model (Lake, 2019; Nye et al., 2020; McCoy et al., 2020); in concurrent work, McCoy et al. (2020) also discuss the strategy of designing families of tasks to impart inductive biases through meta-learning. Meta-learning has recently enjoyed renewed interest in the machine learning community, building from classic work describing hierarchies of learning algorithms (Schmidhuber, 1987). Approaches range from learning to generate prototypes of classes (Snell et al., 2017), using recurrent models to learn the optimization dynamics of a learning procedure (Santoro et al., 2016; Andrychowicz et al., 2016), or bi-level optimization (Finn et al., 2017; Franceschi et al., 2018).

5. Conclusion

Recent developments suggest meta-learning can serve as a useful tool for injecting inductive bias into neural networks even on single-task problems. Attractively, this approach seems compatible with more traditional techniques such as custom architectures, regularization objectives, and data augmentation. In this work, we find that for one of the simplest and widely-used compositional generalization benchmarks, applying meta-learning to acquire a compositional inductive bias for single-task problems can be quite delicate, failing in surprising ways and highlighting the need to better understand the relationship between task distributions and the inductive biases needed to solve them.

In addition to re-affirming the critical role that the selection of the task family plays in the ultimate success of the training procedure, we also find that the procedure for selecting support set data at both meta-train and meta-test time also plays a significant role in whether or not a particular meta-training procedure will be fruitful. Future work might consider not only how a family of tasks and support set selection procedure impact the model's learned generalization ability, but also how the meta-learning algorithm used (memory-based/black box meta-learners vs gradient-based meta-learners) affects this procedure. In addition, scaling this analysis to more complicated settings in vision, grounded language, or real-world language settings, is necessary to better understand whether meta-learning can be a useful tool for acquiring inductive biases for single-task problems.

Acknowledgments

The authors would like to thank Sidd Karamcheti, Kaylee Burns, and Ranjay Krishna for helpful discussions at various stages of the project as well as the anonymous reviewers and attendees of the AAAI Workshop on Meta-Learning for their helpful feedback. This work was supported by a Knight-Hennessy Graduate Fellowship.

References

J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48, 2016.

Jacob Andreas. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.676.

Marcin Andrychowicz, Misha Denil, Sergio Gómez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3988–3996, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic generalization: What is required and can it be learned? In *International Conference on Learning Representations*, 2019.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. BabyAI: First steps towards grounded language learning with a human in the loop. In *International*

Conference on Learning Representations, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, Proceedings of Machine Learning Research. PMLR, 2018.

Nicholas T. Franklin and Michael J. Frank. Generalizing to generalize: humans flexibly switch between compositional and conjunctive structures during reinforcement learning. *bioRxiv*, 2019. doi: 10.1101/547406. URL https://www.biorxiv.org/content/early/2019/09/24/547406.

Tong Gao, Qi Huang, and Raymond Mooney. Systematic generalization on gscan with language conditioned embedding, 2020.

Jonathan Gordon, David Lopez-Paz, Marco Baroni, and Diane Bouchacourt. Permutation equivariant models for compositional generalization in language. In *International Conference on Learning Representations*, 2020.

Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5903–5916. Curran Associates, Inc., 2019.

Drew Arad Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations*, 2018.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: how do neural networks generalise?, 2020.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

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. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*, 2020.

Tim Klinger, Dhaval Adjodah, Vincent Marois, Josh Joseph, Matthew Riemer, Alex 'Sandy' Pentland, and Murray Campbell. A study of compositional generalization in neural models, 2020.

B. M. Lake and M. Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *ICML*, 2018.

Brenden M Lake. Compositional generalization through meta sequence-to-sequence learning. In *Advances in Neural Information Processing Systems 32*, pages 9791–9801, 2019.

Yuanpeng Li, Liang Zhao, Jianyu Wang, and Joel Hestness. Compositional generaliza-

tion for primitive substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4293–4302, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1438.

R. Thomas McCoy, Erin Grant, Paul Smolensky, Thomas L. Griffiths, and Tal Linzen. Universal linguistic inductive biases via meta-learning, 2020.

Maxwell I. Nye, Armando Solar-Lezama, Joshua B. Tenenbaum, and Brenden M. Lake. Learning compositional rules via neural program synthesis, 2020.

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

Jake Russin, Jason Jo, R. O'Reilly, and Yoshua Bengio. Compositional generalization in a deep seq2seq model by separating syntax and semantics. *ArXiv*, abs/1904.09708, 2019.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1842–1850, 2016.

Jurgen Schmidhuber. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Diploma thesis, Institut für Informatik, Technische Universität Munich, 14 May 1987.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.