

Analyzing the Inner Workings of Transformers in Compositional Generalization

Ryoma Kumon and Hitomi Yanaka

The University of Tokyo

{kumoryo9, hyanaka}@is.s.u-tokyo.ac.jp

Abstract

The compositional generalization abilities of neural models have been sought after for human-like linguistic competence. The popular method to evaluate such abilities is to assess the models’ input–output behavior. However, that does not reveal the internal mechanisms, and the underlying competence of such models in compositional generalization remains unclear. To address this problem, we explore the inner workings of a Transformer model by finding an existing subnetwork that contributes to the generalization performance and by performing causal analyses on how the model utilizes syntactic features. We find that the model depends on syntactic features to output the correct answer, but that the subnetwork with much better generalization performance than the whole model relies on a non-compositional algorithm in addition to the syntactic features. We also show that the subnetwork improves its generalization performance relatively slowly during the training compared to the in-distribution one, and the non-compositional solution is acquired in the early stages of the training.

1 Introduction

Compositional generalization, the ability to understand the meaning of a novel language expression based on the composition of known words and syntactic structures (Partee et al., 1984; Fodor and Pylyshyn, 1988), is a crucial aspect for robustness against unseen language data. To assess the compositional generalization abilities of neural models, most existing studies have primarily focused on evaluating model outputs in compositional generalization benchmarks (Kim and Linzen, 2020; Li et al., 2021; Dankers et al., 2022; Li et al., 2023).

However, the model outputs do not necessarily reflect the underlying competence because good performance in the benchmarks does not guarantee that the model implements a solution that generalizes based on compositional rules (i.e., composi-

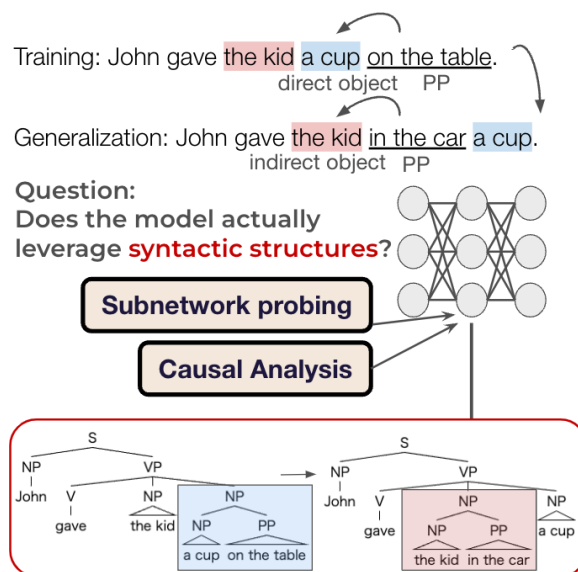


Figure 1: In this study, we investigate what neural models employ in compositional generalization tasks.

tional syntax) and vice versa. In addition, while a growing body of work on model interpretability has investigated the inner workings of Transformer (Vaswani et al., 2017)-based models (Ferando et al., 2024; Rai et al., 2024), compositional generalization has rarely been the focus of such studies. Yao and Koller (2022) and Murty et al. (2023) analyzed the internal mechanisms in compositional generalization but did not focus on the usage of syntactic features, which is the central part of compositional generalization. Thus, the internal mechanisms of the model in compositional generalization are still unclear, and unveiling them would enhance the understanding of the model’s competence.

In this work, we analyze the inner workings of neural models to understand what type of syntactic features the models depend on in tasks requiring compositional generalization. Our analysis method¹ consists of (i) identifying subnetworks

¹Our code and data are available at <https://github.com>.

within the model that perform well in the generalization and (ii) investigating how syntactic features causally affect the original model and its subnetwork, as shown in Figure 1. The causal analysis involves removing the linguistic concept of interest from the model and comparing the generalization performance before and after the removal. To rigorously evaluate and analyze the compositional generalization abilities, we focus not on pretrained models but on a Transformer model trained from scratch; this is because pretraining data contain syntactic structures that should be unseen in this experiment, and pretrained models may not need to generalize compositionally (Kim et al., 2022).

We also aim to conduct a detailed analysis by experimenting with various settings. This study employs two commonly used tasks for evaluating compositional generalization: machine translation and semantic parsing. We test the models with two patterns of compositional generalization, PP-I OBJ and PP-SUBJ, which are similar yet different (see Section 4.1 for details).

The findings from our experiments suggest that the model and its subnetwork that contributes to the generalization performance indeed leverage syntactic structures. However, intriguingly, we also discover that the subnetwork implements other solutions that do not depend on syntactic structures. From these results, we argue that the solutions that the models employ are partly non-compositional. Moreover, analysis of the model at different epochs during the training revealed that the model gradually develops a subnetwork with better generalization performance. The causal analysis of the subnetwork showed that the non-compositional solution was learned during the early phase of the training. We argue that Transformer models need a better inductive bias to generalize compositionally utilizing syntactic features.

2 Background

2.1 Compositional Generalization

Several studies have explored the compositional generalization abilities of modern neural models by focusing on the model performance in tasks such as semantic parsing (Kim and Linzen, 2020; Csordás et al., 2021; Yao and Koller, 2022; Kim et al., 2022; Li et al., 2023) and machine translation (Li et al., 2021; Dankers et al., 2022; Kumon et al., 2024). Such studies indicate that the models lack compo-

sitional generalization abilities in general, while others have worked on improving them by modifying the model architectures (Bergen et al., 2021; Ontanon et al., 2022). Yao and Koller (2022) briefly analyzed which part of seq2seq models causes poor performance in compositional generalization by probing the encoder. They attributed the poor performance to the decoder, stating that the encoder has the linguistic knowledge needed to solve the generalization tasks but the decoder does not use it. These studies focused mainly on the model outputs and the encoded properties, and there has been a debate on whether behavioral evaluation is sufficient to assess compositionality in neural models (McCurdy et al., 2024). Our research analyzes the inner workings of models with subnetwork search and causal analysis, which should give insights into what features causally impact the model behavior. We consider that a compositional solution generalizes by leveraging syntactic structures consistently in a bottom-up manner. In our experiments, we focus on syntactic features that a compositional solution should utilize. Thus, we regard a solution that does not employ these features as non-compositional.

2.2 Linguistic Approach to Interpretability

One approach to interpreting neural models is to study the causal relationship between the target feature and the model behavior based on interventions. Interventions alter either a model’s inputs or its inner representations so that the target feature is the only change, and they test how that change affects the model’s outputs. Recent work has analyzed the linguistic mechanisms (Tucker et al., 2021; Elazar et al., 2021; Ravfogel et al., 2021; Feder et al., 2021; Amini et al., 2023; Belrose et al., 2023; Arora et al., 2024). Belrose et al. (2023) utilized concept erasure, which removes only the concept of interest from the model and tests causal links by comparing the predictions by the model before and after the removal. We employ their method for causally analyzing the causal role of syntactic features in compositional generalization because of its compatibility with our compositional generalization tasks.

Another line of work has explored finding subnetworks with specific properties of interest as a method for model analysis. Cao et al. (2021) proposed subnetwork probing, a pruning-based method that searches for a subnetwork that performs a target linguistic task. As for lingu-

tic generalization, previous studies have found subnetworks that perform syntactic generalization (Bhaskar et al., 2024), hierarchical generalization (Ahuja et al., 2024), and compositional generalization (Hu et al., 2024).

3 Analysis Method

Figure 2 presents our method of analyzing the inner workings of a Transformer model in compositional generalization. It consists of the following three phases: training a base model, subnetwork probing, and causal analysis.

3.1 Training Base Model

To test the compositional generalization abilities of a model, we first construct a dataset that contains the training set, in-distribution test set, and out-of-distribution generalization set. Hereinafter, we refer to the in-distribution test set as the test set and the out-of-distribution generalization set as the generalization set. The generalization set contains unseen syntactic structures that are combinations of those in the training set and requires models to fill the gaps. The dataset is constructed based on a rule-based pipeline used in SGET (Kumon et al., 2024), which evaluates the compositional generalization abilities of neural models on English–Japanese translation tasks. The strict control of sentence generation in SGET utilizing PCFGs (Probabilistic Context-Free Grammars) allows for controlled gaps between the training set and generalization set, which enables the precise evaluation of generalization abilities.

Next, we train a Transformer model from scratch with the training set. As for training tasks, we adopt machine translation and semantic parsing, which have been commonly used in existing studies of evaluating compositional generalization. The reason for using two tasks instead of just one is to investigate how the output format of a task impacts the models’ inner processes. The logical forms in the semantic parsing dataset are created mostly based on the rules proposed by Reddy et al. (2017) and postprocessings, following Kim and Linzen (2020). We remove redundant tokens while maintaining semantic interpretation, following Wu et al. (2023).

3.2 Subnetwork Probing

Neural models have been shown to develop subnetworks for several types of linguistic generalization (Bhaskar et al., 2024; Ahuja et al., 2024)

and modular solutions for compositionality (Lepori et al., 2023). Based on these findings, we hypothesize that the vanilla Transformer develops a subnetwork that generalizes compositionally. To test this hypothesis, to the trained base model we apply subnetwork probing (Cao et al., 2021), a method for discovering an existing subnetwork that achieves high accuracy on a task. Subnetwork probing performs pruning-based probing by training a learnable mask. This method is shown to have low complexity, which means that the mask itself does not learn the task much, and the abilities of the original models are preserved as desired.

In this work, we acquire a subnetwork that performs well in compositional generalization, if any, through subnetwork probing. We use the generalization set to train masks and prune the models. The details of subnetwork probing are in Appendix A.

3.3 Causal Analysis

Next, we analyze the trained models and discovered subnetworks in terms of the extent to which they depend on syntactic structures to generate answers in machine translation and semantic parsing. One of the methods for analyzing the inner workings is to remove target features from a model and observe the causal effect of the removal. LEACE (Belrose et al., 2023) is a method for concept erasure in which only the target concept is removed, with as little impact on the original model as possible. The method updates inner representations so that no linear classifiers can predict concept labels more accurately than a constant function and other concepts are preserved in the model. Removing a concept from deep neural networks is achieved by a procedure called concept scrubbing (Belrose et al., 2023), which sequentially applies LEACE to every layer of a model from the first to the last. In particular, after LEACE is applied to a layer and scrubs a concept therein, the scrubbed representations are passed to the next layer, where LEACE is applied again.

We apply concept scrubbing to both the base models and the discovered subnetworks, removing the target syntactic knowledge. After the concept removal, we evaluate the model predictions in the test set and generalization set of machine translation and semantic parsing. Comparing the model performances before and after the concept removal reveals the causal effect of the syntactic feature of interest on the predictions. Concept scrubbing is

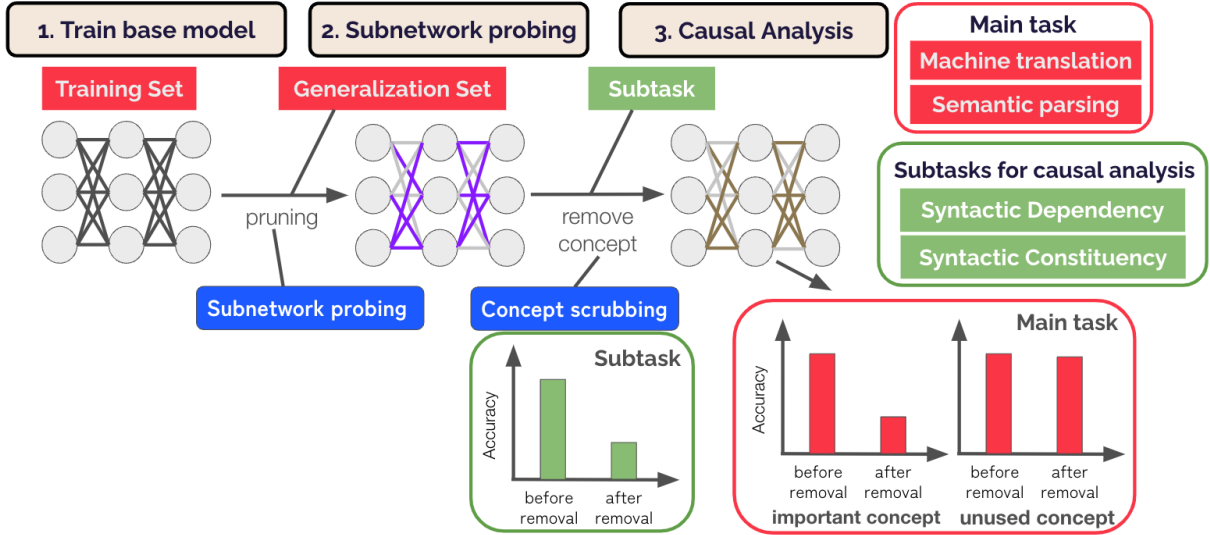


Figure 2: Overview of analysis process, which consists of three phases: training base models, subnetwork probing, and causal analysis.

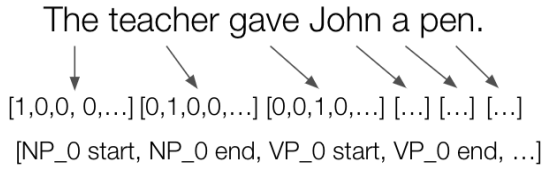


Figure 3: Multi-label classification task used in concept scrubbing for removing syntactic constituency.

suitable for our analysis because it does not require creating alternative inputs or interchange interventions, which are difficult to define for controlling syntactic features in the generalization setting.

Concept scrubbing uses a classification task that represents the concept of interest to erase it. We choose syntactic constituency and syntactic dependency as the concepts for model analysis. We define multi-label classification tasks to represent each concept based on sequence tagging tasks by Elazar et al. (2021), as shown in Figure 3. The task for syntactic constituency is tagging the beginning and end of a phrase, and the task for syntactic dependency is labeling dependency relations. We also test the impact of removing narrower concepts, which differs according to each generalization pattern, such as syntactic constituency regarding only the prepositional phrase (PP) modification of an indirect object noun phrase (NP). In this case, labels are assigned only to the tokens involved in the concepts. For example, when considering syntactic constituency regarding the PP modification of an indirect object NP, the beginning and end of the PP and the NP containing the PP and the modified NP

(not the modified NP) are labeled as ones.

As for the dataset for this classification task, English sentences are generated using the same rule-based method as the one for the main task datasets. The labels are tagged based on syntax trees generated as by-products in the sentence generation process. Constituency boundaries are based on the Penn Treebank (Marcus et al., 1993) definitions, and dependency relations are based on the Universal Dependencies (McDonald et al., 2013) definitions. Using these tasks and datasets, we test whether the models depend on syntactic constituency and syntactic dependency in compositional generalization.

4 Experimental settings

4.1 Compositional Generalization Pattern

In this work, we focus on two compositional generalization patterns, i.e., PP in indirect object NP (PP-IOBJ) and PP in subject NP (PP-SUBJ), as shown in Table 1. Syntactically, these two patterns are relatively simple, which allows for easier causal analyses using multi-label classification tasks.

PP in indirect object NP (PP-IOBJ) In this pattern, all the NPs modified by PPs in the training set appear in the direct object position. Then, models trained on the training set are expected to generalize to PPs modifying indirect object NPs in the generalization set. As Li et al. (2023) pointed out, some sentences have an indirect object NP modified by a PP before a direct object NP, and the dependency between a verb and the direct object

Pattern	Training	Generalization
PP in indirect object NP	The child gave the pen on the table to Liam.	The friend gave the girl in the room a hat.
PP in subject NP	The child broke a cup on the table .	The friend in the room broke a cup.

Table 1: Two compositional generalization patterns tested in the experiments.

NP goes across the PP, which makes the generalization more complex.

For causal analysis in this pattern, we test the impact of three narrower syntactic constituency and dependency concepts, namely, the PP modification of indirect object NPs, direct object NPs, and all NPs, along with overall syntactic constituency and dependency.

PP in subject NP (PP-SUBJ) Similarly to PP-IOBJ, all the NPs modified by PPs in the training set appear in the direct object position. Models trained on the training set are expected to generalize to PPs modifying subject NPs in the generalization set. One aspect that makes PP-SUBJ difficult is that PP modifiers do not appear at the beginning of sentences in the training set. The models may have to generalize to the novel placement of PP modifiers in addition to the novel grammatical role of modified NPs. To mitigate this issue, [Wu et al. \(2023\)](#) added sentences with preposed PP modifiers in the training set, but we avoid that approach for the sake of simpler comparisons between PP-IOBJ and PP-SUBJ.

For causal analysis in this pattern, we test the impact of three narrower syntactic constituency and dependency concepts, namely, the PP modification of subject NPs, direct object NPs, and all NPs, along with overall syntactic constituency and dependency.

4.2 Dataset

As explained in Section 3, we newly construct datasets for each of machine translation and semantic parsing and for classification tasks used in concept scrubbing, using PCFGs with vocabulary of 123 proper nouns, 423 common nouns, 178 verbs, and 43 adjectives. Each of the machine translation and semantic parsing datasets consists of a training set of 80,000 samples, a test set of 10,000 samples, and a generalization set of 30,000 samples. We split the generalization set into two parts: one part is used in training masks in subnetwork probing (Section 3.2), and the other is used

in evaluating the trained models and subnetworks (Section 3.3). Note that the generalization set is constructed for each generalization pattern, and subnetwork probing is performed for each pattern as well. The dataset for classification tasks contains 9,000 samples, and all of them are used for concept scrubbing.

4.3 Training Details

We train an encoder–decoder Transformer model from scratch on our dataset. The model has 3 encoder and 3 decoder layers, 4 attention heads. We set the batch size to 256, the number of epochs to 500, the learning rate to 0.0001, and the weight decay to 0.1. We do not use early stopping because [Csordás et al. \(2021\)](#) showed that continued training without it improves model performance in compositional generalization.

As for subnetwork probing, we train a pruning mask for the trained model. We set the batch size to 256, the number of training epochs to 300, and the learning rate to 0.0005. We do not use early stopping in subnetwork probing.

We run the experiments three times with random seeds and report the average scores as the results. The final checkpoints of each training run are used for the main results (Section 5).

4.4 Evaluation Metric

Following previous studies of evaluating compositional generalization ([Kim and Linzen, 2020](#); [Kumon et al., 2024](#)), we adopt exact match accuracy as the evaluation metrics for both machine translation and semantic parsing. The rule-based pipeline for our dataset generation is designed so that a correct output can be determined uniquely if a model follows compositional rules; thus, using exact match accuracy in this experiment is appropriate.

5 Results

5.1 Output Evaluation

Before analyzing the inner workings of the models, we present the model performance in main

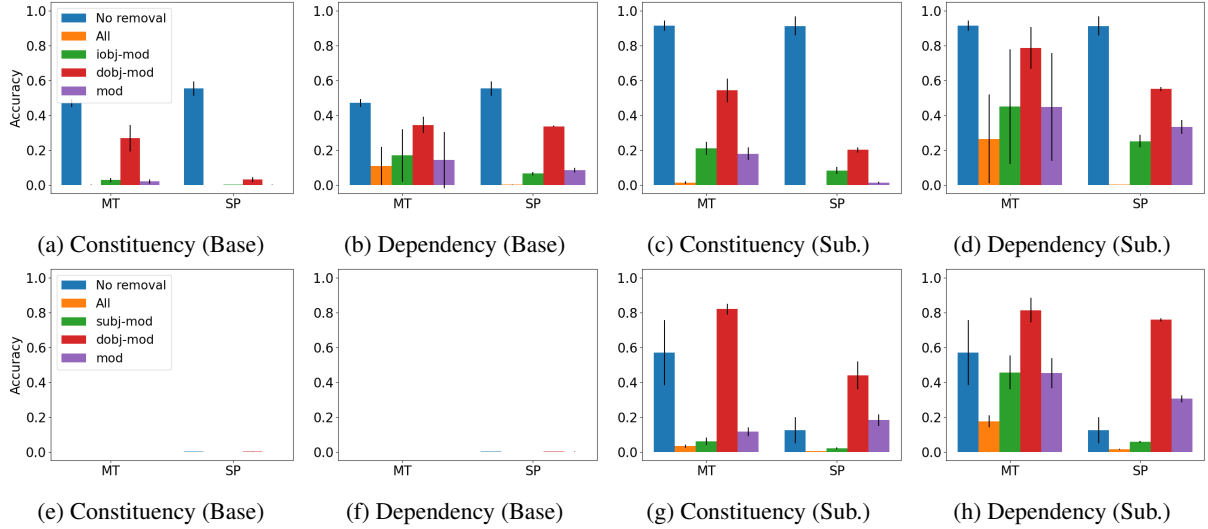


Figure 4: Results of causal analysis in PP-IOBJ (4a-4d) and in PP-SUBJ (4e-4h). Each bar shows the performance in the generalization set after the corresponding concept removal. “All” refers to entirely removing the corresponding syntactic feature. “Iobj-mod” (resp. “dobj-mod”, “subj-mod”) refers to removing the corresponding syntactic feature regarding the PP modifications of indirect object (resp. direct object, subject) NPs. “Mod” refers to removing the corresponding syntactic feature regarding the PP modifications.

Task	Model	Test	PP-IOBJ	PP-SUBJ
MT	Base	99.9 \pm 0.0	47.0 \pm 2.2	0.0 \pm 0.0
	PP-IOBJ Sub.	99.8 \pm 0.0	91.4 \pm 3.0	—
	PP-SUBJ Sub.	96.5 \pm 3.5	—	57.0 \pm 18.8
SP	Base	99.8 \pm 0.0	55.3 \pm 4.1	0.1 \pm 0.0
	PP-IOBJ Sub.	94.3 \pm 0.8	91.2 \pm 5.5	—
	PP-SUBJ Sub.	98.3 \pm 0.0	—	12.4 \pm 7.4

Table 2: Average exact match accuracy (%) of base models and subnetworks in machine translation (MT) and semantic parsing (SP). PP-IOBJ Sub. (resp. PP-SUBJ Sub.) stands for the subnetwork for PP-IOBJ (resp. PP-SUBJ). The column labeled PP-IOBJ (resp. PP-SUBJ) represents the generalization performance in PP-IOBJ (resp. PP-SUBJ).

tasks without concept scrubbing. Table 2 shows the results of the base models and subnetworks in machine translation and semantic parsing.

The base models and subnetworks both performed nearly perfectly in the test set of both machine translation and semantic parsing. The performance of the base model was much worse in both PP-IOBJ and PP-SUBJ than that in the test set, which is consistent with the results of previous studies testing the same generalization patterns (Li et al., 2023; Kumon et al., 2024). On the other hand, the subnetwork scored more than 90% accuracy in PP-IOBJ in both main tasks while keeping the in-distribution performance. This suggests that some part of the trained model implements a certain algorithm that solves these compositional general-

ization tasks. The subnetwork in PP-SUBJ also performed much better in the generalization set than did the base model. It is surprising to see these positive results, considering that previous studies have shown Transformers’ poor performance in compositional generalization tasks.

5.2 Causal Analysis

5.2.1 Generalization in PP-IOBJ

Figures 4a–4d present the results of causal analysis of the base model and subnetwork in PP-IOBJ. First, the base model performed much worse when syntactic constituency or dependency was removed, which shows the model’s reliance on these syntactic features to correctly solve the main tasks. However, the base model cannot be considered to have a compositionally generalizing solution because the generalization performance overall was far from perfect, and a compositionally generalizing model should perform nearly perfectly in the generalization set.

Next, we focus on the subnetwork, which achieved better accuracy in the generalization set than did the base model. Entirely removing syntactic constituency or dependency decreased the accuracy of the subnetwork to almost zero except when removing syntactic dependency in machine translation. This shows that the subnetwork also depends on the syntactic features in general.

We then discuss the impact of the removal of

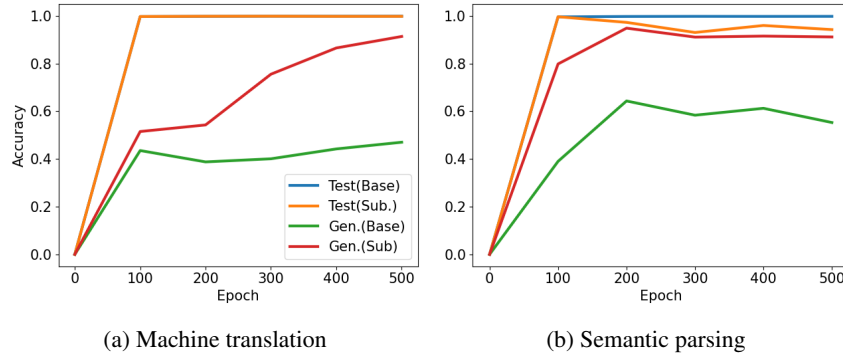


Figure 5: Shift of average accuracy of models in PP-I OBJ over training epochs. Test and Gen. stand for the accuracy on the test and generalization set, respectively. Base and Sub. stand for the accuracy of the base model and subnetwork, respectively.

constituency information on the modification of indirect object NPs. If the subnetwork implements a compositional solution, then removing the constituency information on the modification of indirect object NPs would decrease the accuracy to zero. However, the difference in the performance before and after this concept removal is not as large as when the concept of the constituency is removed entirely, although the generalization performance of the subnetwork decreases to some extent. A similar trend was seen in the removal of syntactic dependency, although the decline in the generalization performance was smaller. These results suggest that the subnetwork depends somewhat on the constituency and dependency regarding the modification of indirect object NPs. At the same time, the subnetwork implements a solution that somehow handles PP-I OBJ in machine translation and semantic parsing yet cannot be considered as a compositionally generalizing one.

Furthermore, regarding the differences in the results between the two tasks, the drop in the generalization performance after the removal of syntactic information was smaller in semantic parsing than in machine translation. It indicates that the models relied on syntactic features instead of a non-compositional solution more in semantic parsing than in machine translation.

5.2.2 Generalization in PP-SUBJ

In contrast to PP-I OBJ, the generalization performance of the subnetwork is far from 100% accuracy, so the subnetwork is not expected to have a perfect solution that generalizes compositionally in PP-SUBJ. However, we still examine on what the subnetwork depends in the main tasks.

Figures 4e–4h show the results of causal analysis

of the subnetwork in PP-SUBJ. Similar to PP-I OBJ, when the information of syntactic constituency or dependency was entirely removed, the performance of the subnetwork dropped to almost 0% except when removing syntactic dependency in machine translation. The subnetwork depends on this information in PP-SUBJ as well.

As for the impact of the removal of the constituency regarding the modification of subject NPs, the performance dropped almost as much as with the removal of the entire constituency. This suggests the subnetworks’ heavy reliance on the modification of subject NPs and their ability to properly use compositional rules at least when the output is correct. On the other hand, when the constituency regarding direct object NPs was removed, the performance improved considerably especially in semantic parsing. This implies that the subnetwork was overfitted to the modification of direct object NPs, and this prevented the subnetwork from achieving better accuracy in PP-SUBJ. Also, the subnetwork seems capable of using the information of modification of NPs regardless of whether NPs are subjects or direct objects, as this removal would not improve the accuracy if the subnetwork learns only that the modification can only come with direct object NPs.

5.3 Transition During Training

We present how the model performance evolved throughout training in PP-I OBJ, shown in Figure 5. As can be seen, the accuracy in the test set grew rapidly, whereas the accuracy in the generalization set improved slowly. In addition, comparing the base model and subnetwork in machine translation, the generalization performance of the subnetwork continued to improve through 500 epochs,

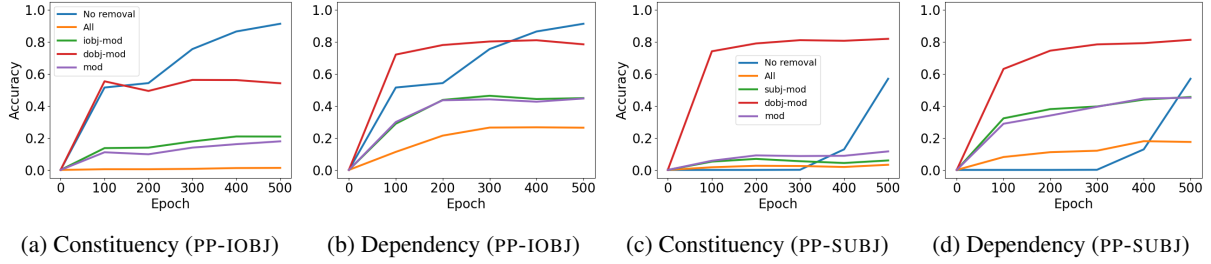


Figure 6: Shift of average accuracy on the generalization set of the subnetworks with each concept removed in machine translation over training epochs.

whereas that of the base model improved only slightly. Therefore, the model may gradually learn an algorithm that can solve the generalization task through the training process with the machine translation task without changing the behavior of the whole model much. The difference in the performance transition between the base model and subnetwork is much less noticeable in semantic parsing. These results suggest that differences in the task settings—such as output formats where structures are represented more explicitly in semantic parsing—influence generalization performance. A similar tendency was discovered in PP-SUBJ (see Appendix B for details), although the generalization performance was generally lower.

Next, we investigate how the inner workings of the subnetwork changed as the training went on. Figure 6 shows the shift of the generalization performance in machine translation of the subnetwork with each linguistic feature removed. In machine translation, the generalization performance of the subnetwork with certain syntactic feature removed was mostly consistent after 200 epochs. This strongly suggests that the subnetwork learned a non-compositional solution in the early stage of the training and retained it throughout the training. Combined with the observation that the original subnetwork continued to improve its accuracy beyond 200 epochs, this result also indicates that a compositional solution relying on syntactic features was acquired gradually.

Similarly, in semantic parsing, the subnetwork with a certain linguistic feature removed mostly retained its performance after 200 epochs, regardless of the generalization accuracy of the original subnetwork. The detailed results of semantic parsing are in Appendix B.

Model	Original	Constituency removed	Dependency removed
Base	90.1 \pm 2.8	88.8 \pm 4.6	90.0 \pm 3.6
PP-IOBJ Sub.	87.9 \pm 4.8	83.8 \pm 6.3	89.3 \pm 3.0
PP-SUBJ Sub.	88.0 \pm 2.1	81.2 \pm 5.4	88.2 \pm 2.3

Table 3: Average accuracy in word-to-word translation of content words.

6 Discussion

6.1 Reliability of LEACE

Since LEACE is a linear concept erasure method, it may fail to remove concepts encoded non-linearly. However, the results when all syntactic information was removed (Figure 4) indicate that most syntactic information used in machine translation and semantic parsing is encoded in linear subspaces. Thus, the impact of non-linearly encoded features should be negligible.

We also validate that LEACE does not erase concepts orthogonal to syntactic ones. We test the model with a word-to-word translation (English to Japanese) of content words; word-to-word translation of content words can be solved without relying on syntax at all. We probe the representation of the final layer of the encoder by a one-layer linear classifier for word-to-word translation, focusing on evaluating the models trained with machine translation datasets. Accuracy is calculated as the proportion of sentences in which all content words are translated correctly. As shown in Table 3, the results suggest that the removal of syntactic features only slightly decreases performance in word-to-word translation, confirming that concepts orthogonal to syntactic ones are mostly preserved in LEACE.

6.2 Impact of Adding Hints in Training

Finally, we investigate how a Transformer model performs under a setting where compositional gen-

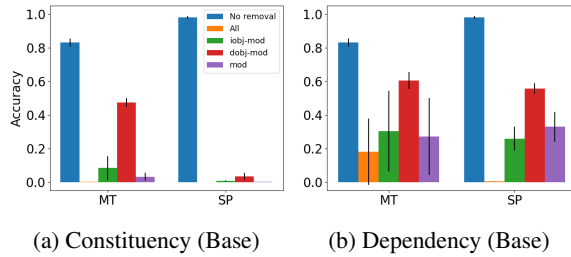


Figure 7: Results of causal analysis in PP-IOBJ when trained with a hint.

eralization is easier. We augment the training set with sentences containing syntactic structures that provide clues for generalization. In particular, we focus on PP-IOBJ and augment the training set with sentences that have a relative clause (RC) modifying an indirect object NP and with ones that have an RC modifying a direct object NP. It should be easier for the model trained with the data involving PPs modifying direct object NPs to generalize to a PP modifying an indirect object NP based on the newly provided hints.

Figure 7 presents the results of causal analysis. Compared with the results without any hint (Figures 4a and 4b), the generalization performance without any concept removal improved by about 40% in both main tasks. Moreover, the performance after the removal of syntactic features regarding the PP modifications of indirect object NPs improved only slightly. This suggests that the algorithm that was implemented in the base model and contributed to the gain in the generalization performance relied on those specific syntactic features. Thus, the model might implement a more robust compositional solution by utilizing the provided hints.

7 Conclusion

In this work, we investigated the inner mechanisms of a Transformer model in compositional generalization tasks. The experimental results showed that the model utilizes syntactic features to some extent in the generalization but that its subnetwork with better generalization accuracy depends on non-syntactic features as well. This indicates that the model develops a non-compositional solution internally and fails to generalize compositionally even when the generalization performance is decent. This paper serves as a foundation for analyzing the underlying mechanisms in compositional generalization from the linguistic perspective. Fu-

ture work might consider other generalization patterns or other linguistic features to obtain a more profound insight into the linguistic competence of neural models.

Limitations

The compositional generalization tasks used in this work are based on synthetically generated datasets and so might not represent sufficiently the variety in natural language expressions. However, these controlled settings are required for precise evaluation requires because all the lexical items and syntactic structures must be split properly into the training and generalization sets. Therefore, using a natural corpus for this experiment would have required much effort, and we leave that for future work.

Another limitation is that the results of this experiment do not necessarily transfer to larger models because we tested relatively small models trained on a small synthetic dataset following previous studies. It would be worth exploring how the trends discovered here change as the model size increases.

Ethical Considerations

All of our datasets were constructed for the sole purpose of the model analysis from the linguistic perspective. They contain no potentially harmful or offensive content.

Acknowledgments

This work was supported by JSPS KAKENHI grant number JP24H00809, JST PRESTO grant number JPMJPR21C8.

References

- Kabir Ahuja, Vidhisha Balachandran, Madhur Panwar, Tianxing He, Noah A. Smith, Navin Goyal, and Yulia Tsvetkov. 2024. [Learning syntax without planting trees: Understanding when and why transformers generalize hierarchically](#). *Preprint*, arXiv:2404.16367. Version 2.
- Afra Amini, Tiago Pimentel, Clara Meister, and Ryan Cotterell. 2023. [Naturalistic causal probing for morpho-syntax](#). *Transactions of the Association for Computational Linguistics*, 11:384–403.
- Aryaman Arora, Dan Jurafsky, and Christopher Potts. 2024. [CausalGym: Benchmarking causal interpretability methods on linguistic tasks](#). In *Proceedings of the 62nd Annual Meeting of the Association*

- for *Computational Linguistics (Volume 1: Long Papers)*, pages 14638–14663, Bangkok, Thailand. Association for Computational Linguistics.
- Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. 2023. [LEACE: Perfect linear concept erasure in closed form](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Leon Bergen, Timothy J. O’Donnell, and Dzmitry Bahdanau. 2021. [Systematic generalization with edge transformers](#). In *Thirty-fifth Conference on Neural Information Processing Systems*.
- Adithya Bhaskar, Dan Friedman, and Danqi Chen. 2024. [The heuristic core: Understanding subnetwork generalization in pretrained language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14351–14368, Bangkok, Thailand. Association for Computational Linguistics.
- Steven Cao, Victor Sanh, and Alexander Rush. 2021. [Low-complexity probing via finding subnetworks](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 960–966, Online. Association for Computational Linguistics.
- Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. 2021. [The devil is in the detail: Simple tricks improve systematic generalization of transformers](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2022. [The paradox of the compositionality of natural language: A neural machine translation case study](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4154–4175, Dublin, Ireland. Association for Computational Linguistics.
- Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. [Amnesic probing: Behavioral explanation with amnesic counterfactuals](#). *Transactions of the Association for Computational Linguistics*, 9:160–175.
- Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021. [CausaLM: Causal model explanation through counterfactual language models](#). *Computational Linguistics*, 47(2):333–386.
- Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. 2024. [A primer on the inner workings of transformer-based language models](#). *Preprint*, arXiv:2405.00208. Version 3.
- Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.
- Michael Y. Hu, Chuan Shi, and Tal Linzen. 2024. [Compositional cores: Persistent attention patterns compositionally generalizing subnetworks](#). In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*.
- Najoung Kim and Tal Linzen. 2020. [COGS: A compositional generalization challenge based on semantic interpretation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.
- Najoung Kim, Tal Linzen, and Paul Smolensky. 2022. [Uncontrolled lexical exposure leads to overestimation of compositional generalization in pretrained models](#). *Preprint*, arXiv:2212.10769. Version 1.
- Ryoma Kumon, Daiki Matsuoka, and Hitomi Yanaka. 2024. [Evaluating structural generalization in neural machine translation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13220–13239, Bangkok, Thailand. Association for Computational Linguistics.
- Michael A. Lepori, Thomas Serre, and Ellie Pavlick. 2023. [Break it down: Evidence for structural compositionality in neural networks](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Bingzhi Li, Lucia Donatelli, Alexander Koller, Tal Linzen, Yuekun Yao, and Najoung Kim. 2023. [SLOG: A structural generalization benchmark for semantic parsing](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3213–3232, Singapore. Association for Computational Linguistics.
- Yafu Li, Yongjing Yin, Yulong Chen, and Yue Zhang. 2021. [On compositional generalization of neural machine translation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4767–4780, Online. Association for Computational Linguistics.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. [Building a large annotated corpus of English: The Penn Treebank](#). *Computational Linguistics*, 19(2):313–330.
- Kate McCurdy, Paul Soulos, Paul Smolensky, Roland Fernandez, and Jianfeng Gao. 2024. [Toward compositional behavior in neural models: A survey of current views](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9323–9339, Miami, Florida, USA. Association for Computational Linguistics.
- Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló,

- and Jungmee Lee. 2013. [Universal Dependency annotation for multilingual parsing](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. Association for Computational Linguistics.
- Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D Manning. 2023. [Characterizing intrinsic compositionality in transformers with tree projections](#). In *The Eleventh International Conference on Learning Representations*.
- Santiago Ontanon, Joshua Ainslie, Zachary Fisher, and Václav Cívek. 2022. [Making transformers solve compositional tasks](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3591–3607, Dublin, Ireland. Association for Computational Linguistics.
- Barbara Partee et al. 1984. Compositionality. *Varieties of formal semantics*, 3:281–311.
- Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. 2024. [A practical review of mechanistic interpretability for transformer-based language models](#). Preprint, arXiv:2407.02646. Version 1.
- Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. 2021. [Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction](#). In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 194–209, Online. Association for Computational Linguistics.
- Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. [Universal semantic parsing](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 89–101, Copenhagen, Denmark. Association for Computational Linguistics.
- Mycal Tucker, Peng Qian, and Roger Levy. 2021. [What if this modified that? syntactic interventions with counterfactual embeddings](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 862–875, Online. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Thirty-first Conference on Neural Information Processing Systems*.
- Zhengxuan Wu, Christopher D. Manning, and Christopher Potts. 2023. [ReCOGS: How incidental details of a logical form overshadow an evaluation of semantic interpretation](#). *Transactions of the Association for Computational Linguistics*, 11:1719–1733.
- Yuekun Yao and Alexander Koller. 2022. [Structural generalization is hard for sequence-to-sequence models](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5048–5062, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Subnetwork Probing

Subnetwork probing (Cao et al., 2021) trains a mask to find a subnetwork of interest. Let $\phi \in \mathbb{R}^d$ be the weights of a model and $Z_i \in [0, 1]$ be the mask for the weight ϕ_i . Z_i follows the hard concrete function parameterized with temperature β_i and a random variable θ_i , that is,

$$\begin{aligned} U_i &\sim \text{Unif}[0, 1], \\ S_i &= \sigma \left(\frac{1}{\beta} \left(\log \frac{U_i}{1 - U_i} + \theta_i \right) \right), \\ Z_i &= \min(1, \max(0, S_i(\zeta - \gamma) + \gamma)), \end{aligned}$$

and $\zeta = 1.1$ and $\gamma = -0.1$ are fixed here.

Subnetwork probing optimizes the mask parameter θ by minimizing the following loss function:

$$\begin{aligned} \frac{1}{|D|} \sum_{(x,y) \in D} \mathbb{E}_{U_i \sim \text{Unif}[0,1]} L(f(x; \phi * z(U, \theta)), y) \\ + \lambda \mathbb{E}|\theta|_0. \end{aligned}$$

The first term is the loss function for the model f masked by $Z_i = z(U_i, \theta_i)$, and the second term corresponds to the penalty for non-zero masks to induce sparsity. During inference, the mask Z_i is binarized to $\{0, 1\}$ based on a threshold.

B Other Results of the Transition During Training

Figure 8 shows how the model performance changes during training in PP-SUBJ, and Figure 9 shows the shift of the generalization performance in semantic parsing when a certain syntactic feature is removed.

C Details of Discovered Subnetworks

Following Cao et al. (2021), we calculated the proportion of unmasked weights, and its trend was mostly the same for all tasks and patterns. The proportion of unmasked weights in each encoder and decoder layer was around 50%-70%, but the proportion was larger for deeper layers in both the encoder and decoder. Also, there were generally more unmasked weights in the decoder than in the encoder. As for the proportion of unmasked weights in multilayer perceptron (MLP) blocks and

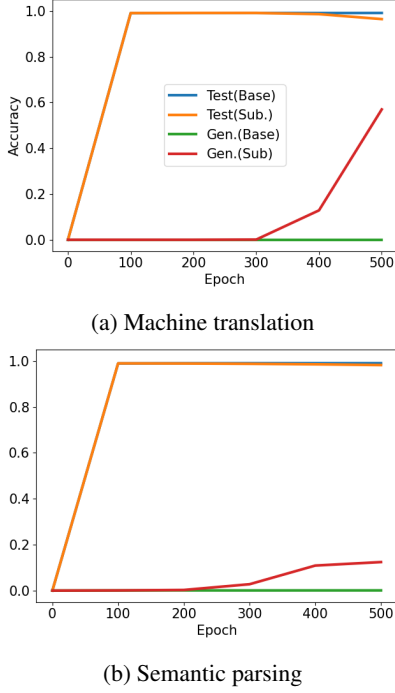


Figure 8: Shift of average accuracy of the models in PP-SUBJ over training epochs.

Task	MT		SP	
	PP-IOBJ Sub.	PP-SUBJ Sub.	PP-IOBJ Sub.	PP-SUBJ Sub.
Enc. 0	55.7 \pm 0.2	58.0 \pm 0.3	49.8 \pm 0.4	53.9 \pm 0.4
Enc. 1	57.9 \pm 0.7	61.4 \pm 0.2	51.5 \pm 0.3	56.6 \pm 0.4
Enc. 2	59.6 \pm 1.1	54.0 \pm 0.5	63.9 \pm 0.9	59.2 \pm 0.2
Dec. 0	59.3 \pm 0.9	63.1 \pm 0.6	53.5 \pm 0.6	59.0 \pm 0.5
Dec. 1	63.8 \pm 0.7	67.5 \pm 0.3	58.8 \pm 0.2	64.9 \pm 0.2
Dec. 2	68.5 \pm 0.1	71.3 \pm 0.2	61.6 \pm 0.2	67.4 \pm 0.1

Table 4: Average proportion of unmasked weights in the two patterns and two tasks.

attention blocks, most layers had more unmasked weights in MLP blocks than in attention blocks. Table 4 and 5 show the proportions of unmasked weights in the extracted subnetworks.

D Computational Resources

We used NVIDIA V100 GPUs for all the experiments. The total runtime for the training and evaluation was around 300 hours.

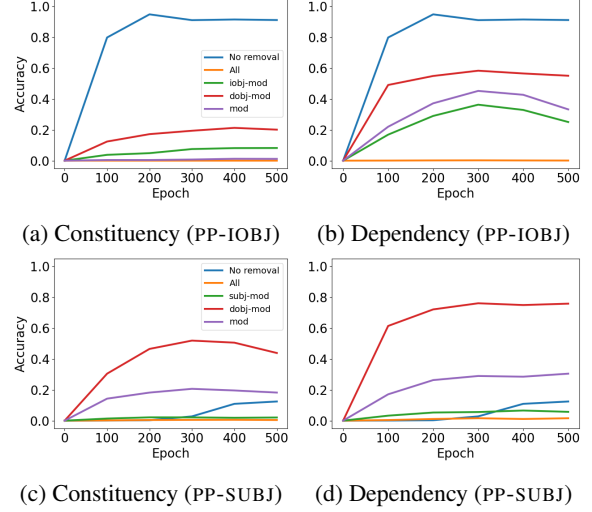


Figure 9: Shift of average accuracy of the models with each concept removed in semantic parsing over training epochs.

Layer	Overall	Attention	MLP
Enc. 0	55.7 \pm 0.2	62.0 \pm 0.3	52.5 \pm 0.2
Enc. 1	57.9 \pm 0.7	56.8 \pm 0.8	58.4 \pm 1.0
Enc. 2	59.6 \pm 1.1	58.3 \pm 1.3	61.0 \pm 0.8
Dec. 0	59.3 \pm 0.9	53.9 \pm 1.4	64.6 \pm 0.5
Dec. 1	63.8 \pm 0.7	59.9 \pm 1.5	67.6 \pm 0.4
Dec. 2	68.5 \pm 0.1	64.7 \pm 0.1	72.3 \pm 0.2

Table 5: Average proportion of unmasked weights in PP-IOBJ and machine translation.