Structural Generalization is Hard for Pretrained Causal Language Models

Anonymous ACL submission

Abstract

Compositional generalization, the ability to interpret novel sentences by systematically combining known structures and words, remains a key challenge in language modeling. Previous work has shown that neural architectures struggle particularly with structural generalization (Kim and Linzen, 2020; Li et al., 2023). We demonstrate that recent pretrained causal language models can significantly improve compositional generalization. Fine-tuned models approach state-of-the-art results on the SLOG benchmark and show robust gains on sub-tasks that previously induced complete failure. However, despite these improvements, structural generalization remains difficult and show that structural generalization is still difficult for recent models. We conduct a series of linguistic diagnostic experiments and show that success on SLOG does not directly correlate with natural language skills. This observation leads us to reconsider the limitations of SLOG as a benchmark of linguistic abilities.

1 Introduction

Human speakers can effortlessly understand and produce novel syntactic structures that they have never encountered before (Chomsky, 1957). From a finite set of rules and observations, they derive an unbounded number of new sentences through recursion. This capacity is grounded in the principle of compositionality: the meaning of a complex expression is determined by the meanings of its parts and the rules used to combine them.

Compositional generalization is equally fundamental for Natural Language Processing (NLP), where models trained on finite data are expected to generalize to infinitely many novel combinations. Yet, standard language models have been shown to struggle with this ability. Kim and Linzen (2020) demonstrated these limitations through the COGS benchmark, which systematically evaluates models' compositional understanding (see Section 3.1). They identified a particularly challenging subset of tasks *structural generalization* where language models consistently fail. Building on this, Li et al. (2023) introduced the SLOG dataset, focusing specifically on structural generalization (Figure 1).

More broadly, the very notion of compositionality in neural networks remains debated. Hupkes et al. (2020) formalize this discussion by translating different linguistic and philosophical interpretations of compositionality into a set of task-independent tests. These include evaluating whether models systematically recombine known parts and rules, extend predictions beyond training length, use local versus global composition operations, remain robust to synonym substitutions, and favor rules over exceptions. Applied to multiple sequence-to-sequence architectures, their framework highlights both strengths and weaknesses in current models' compositional behavior, underscoring the challenges in achieving robust structural generalization.

While prior work confirms that language models struggle with structural generalization, most evaluations focused on encoder–decoder architectures. The exception is Llama2 7B (Touvron et al., 2023), which was fine-tuned but not specifically chosen for its decoder-only design. Interestingly, Llama2 achieved more balanced performance across difficult categories and non-zero accuracy in cases where both a semantic parser (Weißenhorn et al., 2022) and the T5 transformer (Raffel et al., 2020) failed.

These findings motivate an important question: *Can modern decoder-only language models achieve structural generalization?* To answer this question, this paper makes three main contributions:

- First, we evaluated recent decoder-only models on the SLOG benchmark, showing substantial progress in structural generalization

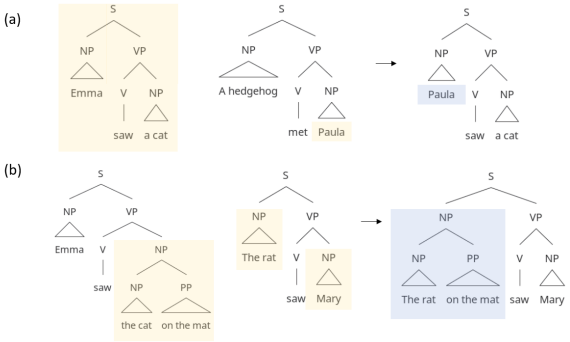

Figure 1: (a) Lexical generalization: a novel combination of a familiar primitive and a familiar structure. (b) Structural generalization: a novel combination of two familiar structures. Figure from Li et al. (2023).

compared to earlier sequence-to-sequence systems. We further analyze their learning dynamics, revealing that these models rapidly exploit pretraining to generalize after a single epoch, yet struggle to maintain long-range generalization likely due to the shorter length distribution of training examples.

- Second, we carry out two case studies on two types of structural generalizations (relative clauses with indirect object and center-embedding, see Section 3.1) to answer the following question: *Is success on SLOG associated with improved linguistic representations?* We show that success on SLOG does not naturally correspond to improved linguistic abilities.
- Finally, we propose a bidirectional translation setup where the model is trained to predict a logical form from an English utterance, or to generate the utterance from the logical form. We show that this does not improve linguistic abilities and hypothesize that the latent representations associated with the logical form are too distant, suggesting limited transfer between the two objectives.

2 Related Work

Lake and Baroni (2017) introduced the SCAN dataset, which was the first benchmark designed to evaluate the compositional generalization abilities of NLP systems. Subsequent datasets have increased in complexity and linguistic depth. The COGS dataset introduced by Kim and Linzen (2020) aimed to assess models' capacity for compositional generalization more directly.

Tasks in this area can be divided into two main categories: *lexical generalization* and *structural generalization*. Lexical generalization involves deriving the meaning of a sentence whose structure has been seen during training, but where certain words appear in unseen roles. For example, the word *hedgehog* may never have appeared in a subject position during training but could occur as such in the generalization set. This subtask has proven successful for language models. However, their main weakness remains *structural generalization*—the ability to interpret novel structures composed of familiar substructures. Figure 1 illustrates this distinction. For example in top figures (a) we notice that the test sentence structure has been seen in training, and "Paula" has been seen as well, but only in an object position. The generalization set then contains a sentence whose structure has been seen at training time but where "Paula" is placed in subject position. The bottom part (b) of Figure 1 shows an example of structural generalization, which consists in generalizing to a novel structure, which is obtainable by forming novel combinations of substructures that are present in the training set. This example features a complex noun phrase composed of a noun phrase and a prepositional phrase. Such a complex NP is never in a subject position in the training set. In the generalization set, a complex NP has to be interpreted in a subject position.

Both the COGS and SLOG datasets reveal that models still perform poorly on structural generalization, sometimes handling very simple novel combinations but failing in most cases.

The AM Parser (Weißenhorn et al., 2022) remains the highest-scoring approach among these models, largely due to its near-perfect performance on easier generalization categories. The parser constructs AM dependency trees, which are subsequently used to derive the logical form. However, this approach has an inherent limitation: successful parsing requires the sentence's dependency structure to be projective. Therefore, performance drops to near zero on several structural generalization categories that violate this constraint, highlighting the need for alternative methods better suited to the SLOG dataset. Yao and Koller (2022) further showed that encoder–decoder architectures perform well on lexical generalization but fail dramatically on structural generalization. Their probing experiments suggest that while the encoder learns relevant structural information, the decoder struggles to use it effectively

Other lines of work attempt to mitigate compo-

Dataset	Train Set	Dev Set	Test Set	Gen Set
COGS	24155	3000	3000	21000
SLOG	32755	4046	4046	17000

Table 1: Composition of datasets.

sitional generalization failures in neural models in different ways. Some modify training objectives to encourage transformers to capture compositional patterns, enabling them to handle COGS-like tasks (Ontañón et al., 2022; Murty et al., 2023). Murty et al. (2023) specifically introduce Pushdown Layers, a self-attention modification that tracks recursive structure via a stack-like mechanism, improving syntactic generalization and sample efficiency in Transformers. In contrast, Petit et al. (2023) incorporate explicit structural constraints into a neural graph-based parser, significantly boosting performance on examples requiring structural generalization in COGS.

3 Datasets and Evaluation

3.1 Dataset

We evaluate our models on two compositional generalization benchmarks:

- **COGS** (Kim and Linzen, 2020), which focuses on *lexical generalization*;
- **SLOG** (Li et al., 2023), a sequel to COGS that emphasizes *structural generalization*.

Table 1 shows the sentence count per set in the COGS and SLOG datasets. COGS contains 21 generalization categories, most of which involve lexical generalization, whereas SLOG contains 17 categories focused entirely on structural generalization.

Because structural generalization is our primary focus, and lexical generalization is already handled well by most modern models, our experiments mainly focus on the SLOG dataset and are not systematically replicated on COGS.

Figure 1 illustrates the difference between these two types of generalization. Both datasets assess models' ability to convert natural-language sentences into corresponding logical forms. An example from SLOG is shown below.

```
Input sentence:
The balloon was handed to a donkey by
    Ava.
Output logical form:
balloon(x_1); hand.theme(x_3, x_1) AND
hand.recipient(x_3, x_6) AND
hand.agent(x_3, Ava) AND donkey(x_6)
```

We focus on two generalization categories from SLOG, because their accuracy greatly fluctuates from prior work to our experiments:

- Center-Embedding: It consists of the embedding of a phrase in the middle of another phrase, creating long distance dependencies.

  Example (depth 2): "A boy helped the cat that the girl that Emma changed rolled." The generalization set contains deeper structures (of depths 5-12).

- Indirect Object–Extracted Relative Clause: It is a relative clause in which the extracted element corresponds to the indirect object of the embedded verb.

  Example: Noah saw the cat that Emma gave a cake to.

  This sentence is obtained by combining two simpler structures from the training set :

  - Noah saw the cat that gave a fish to Liam.
  - Noah saw the cat that Liam liked.

3.2 Evaluation

To evaluate syntactic and compositional performance, Kim and Linzen (2020) employ an *exact-match* criterion, where a model prediction is considered correct only if it matches the gold logical form exactly. Although Wu et al. (2023) identified certain limitations in this evaluation approach with those datasets, we retain it to ensure a fair comparison with SLOG prior results.

We adopt the same post-processing procedure as Li et al. (2023): all sentences are lowercased and special characters are removed. Variables are then reindexed[1] from left to right. Finally, conjunctive AND statements are locally reordered alphabetically, since they are commutative.

Only in the case of few-shot experiments later, we use a *soft-match* criterion, where the output should contain the gold label and not match it exactly. We make this choice because pretrained models can have varying output formats while answering right.

[1]One could argue that variable reindexing simplifies the task by discarding ordering information. However, we adopt this convention as it does not alter the underlying semantics of a sentence, and variable names should generally be treated as non-informative.

Model	1S	3S	5S	10S
Olmo2 7B Instruct	0.9	3.2	4.5	5.7
Gemma2 9B Instruct	1.7	3.6	5.0	6.9
Llama 3.1 8B Instruct	1.5	4.6	6.4	9.4

Table 2: Few-shot soft accuracies on pretrained models (COGS).

Model	1S	3S	5S	10S
Olmo2 7B Instruct	0	0.1	0.2	0.5
Gemma2 9B Instruct	0.1	0.4	0.7	1.4
Llama 3.1 8B Instruct	0.1	0.3	0.7	1.4

Table 3: Few-shot soft accuracies on pretrained models (SLOG).

4 Generalization Abilities of Pretrained Causal Language Models

4.1 Do Pretrained Models Naturally Generalize Compositionally?

We evaluate the Instruct versions of three models:[2] OLMo2 7B (OLMo et al., 2024), Llama3.1 (Dubey et al., 2024), and Gemma2 9B (Riviere et al., 2024), in a few-shot, in-context learning (ICL) setting to assess their performance on COGS and SLOG compositionality datasets (see Appendix E for full prompts).

We sampled the ICL examples from the training sets of each dataset. Tables 2 and 3 summarize results. Performance is poor on COGS, and is extremely low on SLOG, indicating that few-shot prompting alone is insufficient for structural generalization.

To encourage the model to focus on syntactic structure rather than semantics, we propose an *assisted few-shot* method that selects examples with similar syntactic patterns. The approach maintains a mapping of training sentences to their corresponding syntactic trees using the SpaCy library (Honnibal et al., 2020). We then compute the tree-edit distance between the input sentence's tree and all training trees using the algorithm of Zhang and Shasha (1989). Sentences with the lowest tree-edit distance are selected as few-shot candidates, increasing the likelihood that the model observes relevant structural patterns.

[2]Models were selected around 10B parameters to ensure availability and fit on a single H100 GPU. We chose Gemma2 (Gemma2 in some later experiments) and Llama3.1 as state-of-the-art representatives of this scale, and OLMo2 as a fully open model

Tables 4 and 5 illustrate the benefits of this method: accuracy improves significantly, particularly on COGS (see Tables 2 and 3 for standard few-shot results on COGS and SLOG, respectively). However, the method has several limitations. First, SpaCy-generated syntactic trees are not guaranteed to be accurate, and performance may degrade for more complex sentences. This can lead to suboptimal example selection, partially explaining the modest overall gains. Second, the selected examples may only partially cover the label space, resulting in cases where the model has never observed the correct label for a particular instance.

Despite these limitations, the assisted few-shot experiment serves two purposes. First, it provides the model with structurally relevant examples, potentially improving generalization by illustrating patterns directly applicable to the target sentence. Second, the observed failure on the SLOG dataset reinforces our conclusion that the model was likely not exposed to SLOG generalization examples during pretraining, or that any exposure was insufficient for effective knowledge transfer.

It is known that COGS, which focuses on lexical generalization, can be solved by language models (Wu et al., 2023), which is not the case for SLOG (Li et al., 2023). We thus focus on SLOG in the following experiments. Moreover, we will emphasize two phenomena, which proved difficult for most models and for which the understanding of difficulty should be studied more deeply. Center-Embeddings, which scored mid results for shallower depths and near 0% accuracy for deeper ones, excluding the AM-Parser which showed near perfect accuracy (as this structure yields projective dependency trees). We will also study Indirect object-extracted RC, which showed near zero results for almost every setup, including the AM-Parser.

Due to very poor performance on preliminary experiments and library incompatibilities, we exclude OLMo2 from further experiments. The extremely low accuracy for every LLM in a few-shot setting is a strong indication that models have no test set

Model	1S	3S	5S	10S
Olmo2 7B Instruct	6.7	12.9	15.9	20.4
Gemma2 9B Instruct	11.1	17.7	21.4	21.4
Llama 3.1 8B Instruct	12.8	19.1	21.8	26.4

Table 4: Assisted few-shot soft accuracies on pretrained models. (COGS)

contamination.

4.2 Do Fine-Tuned Models Exhibit Natural Generalization?

We fine-tuned Llama3.1 8B (Dubey et al., 2024), Gemma2 9B (Riviere et al., 2024), and Gemma3 12B (Team et al., 2025) using LoRA (Hu et al., 2022) for varying training durations, and evaluated the resulting models on the SLOG benchmark (see Appendix A for details on hyper-parameters selection and ablation studies).

Figure 2 reports the best-performing fine-tuned models. Overall, our approach surpasses previously reported results for both encoder–decoder and decoder-only architectures, approaching the performance of the AM-Parser. In particular, our models achieve the best scores on 6 out of the 17 generalization categories, while substantially improving performance on structures that previously exhibited near-zero accuracy (full per-category results for Gemma 2 9B are provided in Table 9 in Appendix B).

A notable finding is the large improvement observed for indirect object-extracted relative clauses (`RC Iobj extracted`) and long *wh*-movement constructions. Compared to the strongest previously reported results from Li et al. (2023), Gemma2-9B improves accuracy by 41 points and 23 points on these categories, respectively. These gains are particularly significant because both categories previously resulted in complete failure for the AM-Parser (0% accuracy). This limitation likely stems from the fact that such constructions yield non-projective dependency trees, which are incompatible with the AM-Parser architecture.

These results establish a new reference for fine-tuned, generative language models, with reasonable performance on sub-tasks where the AM-Parser completely failed.

4.2.1 Learning Curve Analysis

To investigate further, Figure 3 tracks performance on 'RC Iobj extracted' and deeper structures ('Center Embeddings 5-12' and 'Center Embeddings 3') over training epochs, with global accuracy also monitored to quantify overall impact.

We observe that the models' performance peaks around epoch 1 and gradually declines for long-dependency categories. A closer view (Figure 4) shows accuracy rises steadily until the first epoch, then stabilizes and eventually decreases. Early in training, models appear to leverage pretrained representations to adapt quickly to dataset-specific structures. Extended fine-tuning, however, reduces accuracy on deeper or longer structures, likely due to training distribution biases: the models are accustomed to long sequences, but further training emphasizes shorter sequences more typical of the dataset (see Appendix D for a discussion of length distribution differences).

'RC Iobj extracted' exhibits unstable behavior,

Model	1S	3S	5S	10S
Olmo2 7B Instruct	0.1	0.7	1.0	1.3
Gemma2 9B Instruct	1.5	2.4	2.6	3.0
Llama 3.1 8B Instruct	0.6	1.0	1.3	1.6

Table 5: Assisted few-shot soft accuracies on pretrained models (SLOG).

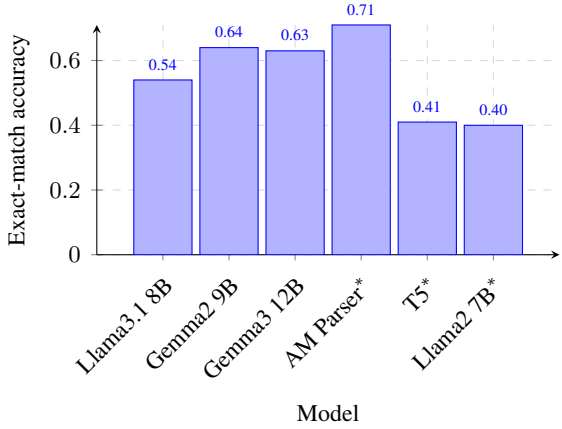

Figure 2: Best fine-tuned models accuracies over ten epochs SLOG). *Results from Li et al. (2023).

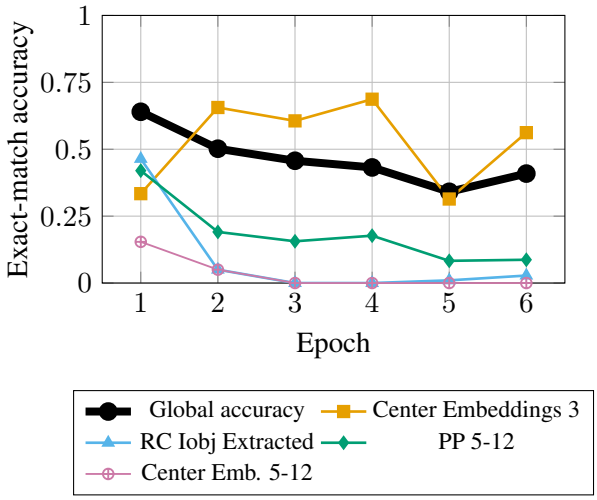

Figure 3: Exact-match accuracy of difficult categories relative to fine-tuning duration (gemma 2 9B).

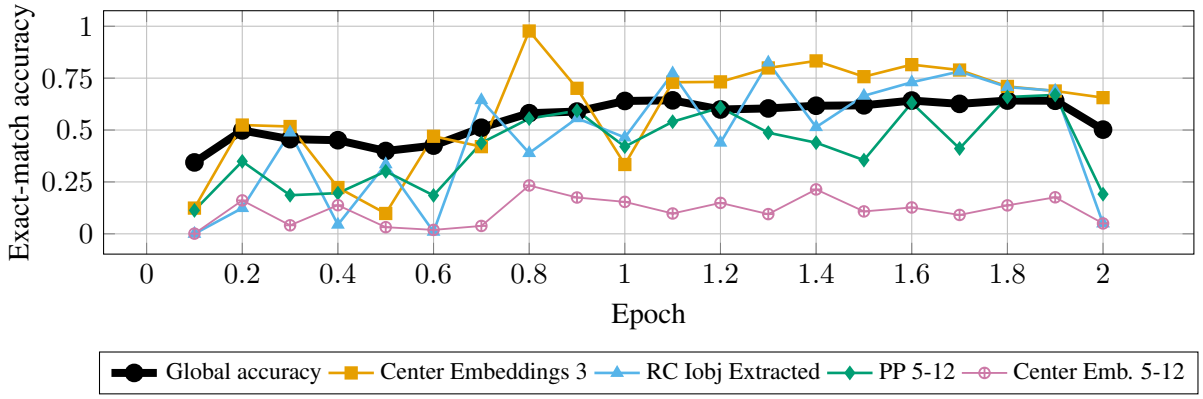

Figure 4: Exact-match accuracy of difficult categories relative to fine-tuning duration (gemma 2 9B) zoom epoch 0-2.

occasionally improving by up to 70% compared to Li et al. (2023), while deeper structures remain challenging. These observations highlight three key points:

1. Pretrained decoder-only models can generalize long-dependencies early in the fine-tuning.
2. Decoder-only models can tackle highly complex tasks, contradicting prior claims by Li et al. (2023), although learning can be unstable.
3. Decoder-only models may serve as viable alternatives for structures that are not reliably handled by semantic parsers.

A similar experiment is available in Appendix C for Gemma3 12B.

5 Do Fine-Tuned Models Really Capture Linguistic Structures? A Case Study on RC-IOBJ Extracted and Center-Embedding

The previous results show that pretrained causal models can achieve much higher scores than the different models evaluated on SLOG in prior work (excluding the AM-Parser). These results suggest that the models acquire partial capabilities for handling complex grammatical structures for which other models previously failed. That is the case specifically for both RC IObJ Extracted, and Center-Embedding 5-12 (the latter being dealt with perfectly by the parser, but failing on most models). The goal of this section is to study if these capabilities can be partly explained through controlled experiments about models' linguistic abilities.

5.1 RC-IOBJ Extracted

First of all, we aim to understand, whether our language models truly capture the grammatical structure underlying indirect object-extracted relative clauses In order to do so, we construct minimal pairs consisting of (i) the original grammatical sentence (ii) an ungrammatical sentence that differs minimally from sentence (i).

An example sentence from the set is: "The mouse liked a girl that William offered the cake to" The possible ungrammatical variations are:

- Append "to": "The mouse liked a girl that William offered the cake to to"
- Append "Tomorrow": "The mouse liked a girl that William offered the cake to tomorrow"
- Replace "to" by "Tomorrow": "The mouse liked a girl that William offered the cake tomorrow"
- Prepend "to": "To the mouse liked a girl that William offered the cake tomorrow"

We use a linear probe that takes as input the last layer of the model and is trained to perform binary classification over sentence grammaticality. The goal is to assess whether the model representation contains a linearly separable signal about the grammaticality of a sentence. As a probing baseline, we use the same LLM architecture with randomized parameters.

Table 6 shows that the fine-tuning does not improve the probe results. For both models, accuracies oscillate around 85% regardless of training time, suggesting that fine-tuning does not substantially improve the linear separability of grammaticality information. These results suggest that improvements in logical-form prediction do not nec-

Model	Epoch	Random	0	1	2	3	4
Gemma 2 9B		70.9	85.0	85.7	83.6	83.8	83.5
Llama 3.1 8B		71.2	85.4	85.0	85.3	85.6	85.4

Table 6: Probing accuracy.

essarily correspond to more general grammatical representations.

5.2 Center Embedding

To investigate how models handle center-embedded structures, we designed a number prediction task targeting verbs within nested embeddings. Following Tal Linzen (Linzen et al., 2016), we analyze subject–verb agreement as a probe of structural understanding. Starting from sentences containing only singular subject–verb pairs, we systematically convert selected pairs (excluding the final one) to plural forms and evaluate the model's output logits for the final verb. More specifically, we measure the normalized probability mass assigned to singular versus plural verb forms. In this setting, we refer to each modified plural subject–verb pair as an *attractor*.

The first experiment will be used to study the impact of plural attractor frequency within the sentence in finding the right number for the final verb.

Figure 5 results show an almost linear decrease of the singular share as we increase the number of plural attractors in a given sentence. This behavior leads us to conclude that the model heavily relies on context to guess the number, as opposed to finding the corresponding subject and choosing the number accordingly.

The second experiment is meant to study the proximity of a given attractor with the verb to generate in predicting the correct verb number. In this setup, the missing verb is expected to be singular, and a single plural pair is inserted in the sentence at a 0-10 distance from the verb to generate. Figure 6 shows that the attractor impact is much greater for distance 0 and 1, with lower impact for greater distances.

These two figures confirm that the model mainly relies on context, so on previous verb numbers, to define what number the missing verb should be, with nearby verbs exerting the strongest influence.

The fine-tuned models have partial success in turning some of those sentences into their corresponding logical form, which necessarily requires recovering the correct subject–verb associations. We hypothesized that this behavior could lead the

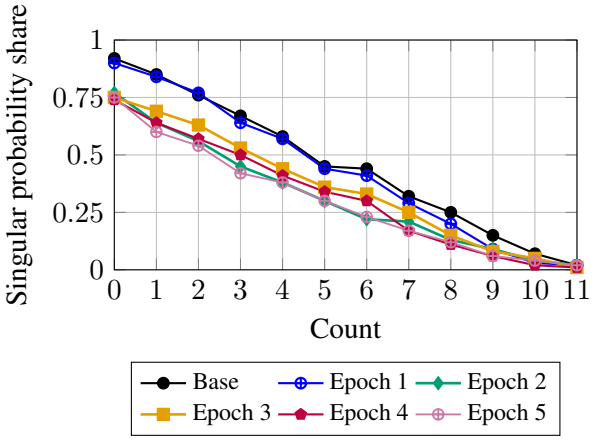

Figure 5: Accuracy by count across training epochs.

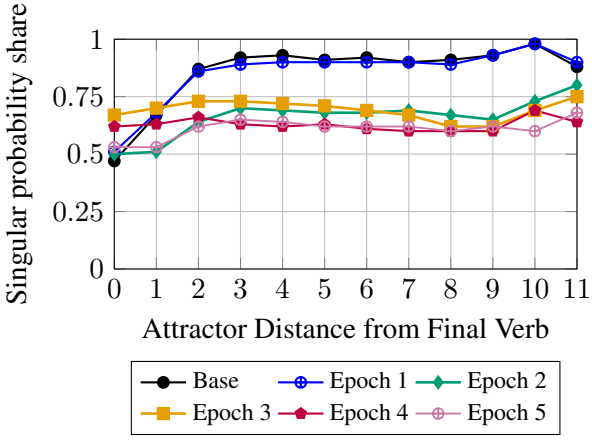

Figure 6: Singular share by attractor distance.

model to use this information in favoring the number based on the subject-verb association, but it is proves to be decorrelated.

These results suggest that fine-tuned models exhibit a form of syntactic priming (Jumelet et al., 2024): the model tends to use the number most present previously in the sentence. This phenomenon also observed in human readers biases the results depending on overall context, and more importantly on nearby context.

6 Does Bidirectional Fine-Tuning Improve Structural Generalization in LLMs?

To investigate whether learning the inverse mapping between logical forms and natural language improves structural generalization, we jointly trained models on both semantic parsing (English → LF) and logical form generation (LF → English). This experiment was inspired by prior work in neural machine translation showing that bidirec-

Model	Epoch	Random	0	1	2	3	4
Gemma 2 9B		70.9	85.0	85.3	84.6	81.5	84.3

Table 7: Probing accuracy.

tional objectives can improve latent representation quality and encourage stronger alignment between input and output structures (Ding et al., 2021). We therefore hypothesized that jointly modeling both directions could encourage models to acquire more abstract hierarchical representations.

However, our experiments show that bidirectional training does not improve structural generalization on SLOG. While models achieved strong performance on the reverse LF → English task (up to 80% exact match accuracy), forward semantic parsing performance remained comparable to or slightly below single-task training. In particular, we observed no substantial gains on categories involving deeper structural dependencies or novel syntactic compositions. Moreover, Table 7 reproduces the experiment of Section 5.1 with the models trained in both directions, and shows identical behavior. Changing the training objective does not substantially improve the probe's ability to predict sentence grammaticality.

These results suggest that improvements in inverse generation do not necessarily transfer to compositional semantic parsing abilities. One possible explanation is that the reverse task is substantially easier: logical forms provide explicit semantic roles and disentangled structure, making the generation problem closer to template realization than structural inference. As a result, the auxiliary objective may primarily reinforce lexical and local correspondences rather than the hierarchical abstractions required for structural generalization.

Additionally, as our experiments rely on parameter-efficient fine-tuning through LoRA, the limited adaptation capacity of low-rank updates may also constrain the extent to which models can reorganize representations for structural generalization.

Overall, our findings suggest that bidirectional training alone is insufficient to induce stronger hierarchical compositional abstractions in pretrained causal language models.

7 Conclusion

Prior work on the SLOG dataset highlighted the limitations of language models in compositional generalization. In this study, we demonstrate that modern pretrained decoder-only models substantially improve performance across categories previously deemed unsolvable. Our experiments reveal that these models rapidly leverage pretraining to generalize to novel structures early in fine-tuning, achieving robust gains on challenging tasks such as indirect object-extracted relative clauses center-embedding 5-12. Overall, our results establish decoder-only models as a promising approach to structural generalization, offering both a strong alternative to encoder-decoder models and a viable complement to semantic parsers for tasks involving highly complex syntactic dependencies.

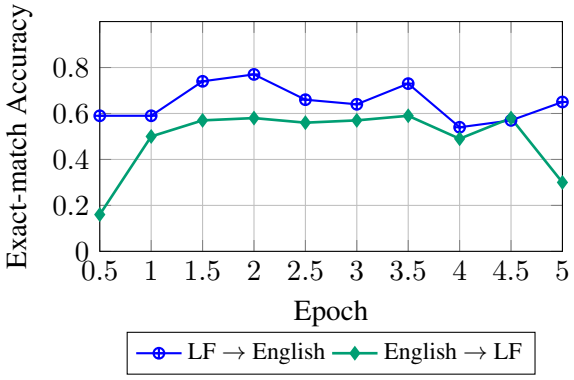

Figure 7: Exact-match accuracy across training epochs for the two generation directions.

However, we also observe that extended fine-tuning can reduce performance on longer or more complex structures, likely due to dataset-specific training biases. Moreover, further analysis shows that the understanding of SLOG is not directly correlated with the understanding of English. These issues may not point to an inability of decoder-only models to solve SLOG tasks but rather to highlight the need for a new dataset for structural understanding that accounts for the characteristics of recent language models trained on human data. One could also question the need to push the boundaries of structural generalization tests that focus on structures far beyond what speakers naturally produce. The deeper structures analyzed in this dataset would not be produced by speakers and are overly complicated to understand even for a trained linguist. They are also semantically unnatural. Hence we argue that they evaluate some form of formal reasoning rather than compositionality as a linguistic ability used unconsciously by speakers in everyday interactions.

8 Limitations

- Our experiments rely on LoRA for parameter-efficient fine-tuning, which may influence the observed results. This choice was primarily motivated by computational constraints, as full fine-tuning of multiple large language models would otherwise have been too expensive.
- The assisted few-shot comparisons in Section 4.1 could be further studied by varying the operation weights in the tree-edit distance. In particular, assigning a lower weight to the node-renaming operation might emphasize structural information and produce more relevant associations. Additionally, the use of SpaCy to construct syntactic trees introduces another limitation. While SpaCy is trained for general syntactic parsing, it is unlikely to produce accurate trees for the highly complex sentences present in these datasets. This may result in suboptimal example selection and an underestimation of performance on longer or more complex syntactic structures.

References

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

Liang Ding, Di Wu, and Dacheng Tao. 2021. Improving neural machine translation by bidirectional training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3278–3284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 516 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net.

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

Jaap Jumelet, Willem Zuidema, and Arabella Sinclair. 2024. Do language models exhibit human-like structural priming effects? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14727–14742, Bangkok, Thailand. Association for Computational Linguistics.

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.

Brenden M. Lake and Marco Baroni. 2017. Still not systematic after all these years: On the compositional skills of sequence-to-sequence recurrent networks. *CoRR*, abs/1711.00350.

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.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher Manning. 2023. Pushdown Layers: Encoding Recursive Structure in Transformer Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3233–3247, Singapore. Association for Computational Linguistics.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2024. 2 olmo 2 furious. *ArXiv*, abs/2501.00656.

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

Alban Petit, Caio Corro, and François Yvon. 2023. Structural generalization in COGS: Supertagging is (almost) all you need. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1089–1101, Singapore. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Gemma Team Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L'eonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ram'e, Johan Ferret, Peter Liu, Pouya Dehghani Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, and 176 others. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv*, abs/2408.00118.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Pia Weißenhorn, Lucia Donatelli, and Alexander Koller. 2022. Compositional generalization with a broad-coverage semantic parser. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 44–54, Seattle, Washington. Association for Computational Linguistics.

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.

Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18:1245–1262.

A Fine-Tuning Details And Ablation Study

For all fine-tuning experiments, we used LoRA with the following hyper-parameters: rank 16, alpha 32, and dropout 0.1. The learning rate was set to 0.0002, with 100 warmup steps and a weight decay of 0.01. The batch size was 32; for Gemma2 9B and Gemma3 12B, we simulated a batch size of 32 using gradient accumulation to fit GPU memory (2x16 and 4x8 respectively). Those hyper-parameters are according to our ablation study performed with Llama3.1 in 8. This study is narrow, due to our very low compute power, but sufficient as those pretrained models are extensively used and fine-tuned; thus, we referred to usage recommendations for hyper-parameters on different tasks, and explored a close neighborhood to confirm the choice was valid.

Overall, training and inference required approximately 1,000 GPU hours on H100 GPUs, using one GPU at a time.

B Detailed Results

Table 9 presents per-category results on SLOG, compared to previous work. The results demonstrate that our models approach the performance of the AM-Parser while achieving non-zero accuracy across all categories. Notably, the model reaches nearly 50% accuracy on 'Indirect object-extracted RC' and long *Wh*-questions, structures that previously caused very low performance in nearly all models. Furthermore, the model achieves approximately 15% accuracy on deeper center embeddings—a category where all models except the AM-Parser previously failed—demonstrating that long-distance dependencies can be learned by fine-tuned decoder-only models.

C Learning Curve of Gemma3 12B

Figure 8 illustrates the performance of the Gemma3 12B model over training epochs, reporting both global accuracy and accuracy on a subset of specific categories. Compared to Gemma2 9B, Gemma3 exhibits a notably more stable global accuracy throughout training.

This observation does not contradict our prior hypothesis: the model still forgets how to handle long dependencies quickly after the first epoch. However, unlike Gemma2 9B, Gemma3 achieves such high accuracy on the easier categories that the over-

LoRA Rank	Learning Rate	Dev Loss
8	0.0001	0.000324
8	0.0002	0.000250
8	0.0003	0.000328
16	0.0001	0.000296
16	**0.0002**	**0.000234**
16	0.0003	0.000255
32	0.0001	0.000416
32	0.0002	0.000264
32	0.0003	0.000237

Table 8: Development loss for different LoRA rank and learning rate with Llama3.1

all global accuracy remains elevated, effectively masking the underlying learning challenges.

Extended training does not necessarily improve generalization; instead, it may lead to overfitting on a subset of categories that are easy enough to maintain strong global performance. This behavior is reminiscent of the findings in Kim and Linzen (2020), where models could reach high overall accuracy (e.g., 80%) while failing completely on structural generalization tasks.

Given that the Gemma3 training set has a more stable learning curve, we expect this model to be able to outperform Gemma2 9B if we found a way to bypass the problem of the length distribution in the training set, potentially approaching or even exceeding the performance of the AM-Parser on challenging structural generalization categories.

D Length Statistics

To understand why long-dependency structures are handled successfully by the model early in training but subsequently degrade, it is important to examine the distribution of training data. Figure 9 depicts the token-length density for the training set, the generalization set, and a subset of the generalization set containing long-dependency center embeddings.

As shown, long-dependency structures lie far outside the bulk of the training data distribution. This discrepancy explains why the model initially succeeds on these structures but quickly forgets how to process them after the first epoch.

E Prompts

All few-shot experiments used prompts of the following form:

```
You are a helpful assistant. Your task
    is to turn sentences into a logical
    form. Inspire yourself
from the following examples:

1. Input sentence:
The balloon was handed to a donkey by
    Ava.
Output logical form:
balloon(x_1); hand.theme(x_3, x_1) AND
hand.recipient(x_3, x_6) AND
hand.agent(x_3, Ava) AND donkey(x_6)

Now transform the following sentence
    into a
logical form without any explanation,
    inspiring
yourself from the examples above:
Input sentence:
```

The number of examples varied across experiments, and they were sampled randomly from the training set.

For fine-tuning, no specific prompts were used. Instead, we employed the simpler format: ‘Input sentence‘ followed by ‘Output logical form‘, rather than the standard Input–Output template.

The prompt for the bidirectional training differs slightly by adding the task as "TASK: Turn the sentence into a logical form" and "TASK: Turn this logical form into a sentence"

Generalization cases	Gemma 2 9B	Vanilla Transformer*	T5*	Llama2 7B*	AM-Parser*
Deeper PP recursion	41.9	13.1	16.6	20.6	**100.0**
Deeper tail CP recursion	40.7	0.9	5.3	12.1	**100.0**
Deeper center embedding	15.4	0.0	0.0	0.0	**99.5**
Shallower PP recursion	92.1	98.7	93.1	98.9	**100.0**
Shallower tail CP recursion	88.7	55.2	60.9	98.1	**100.0**
Shallower center embedding	33.4	0.0	64.1	50.7	**100.0**
PP in subject NPs	**64.7**	0.0	0.8	28.9	57.6
PP in indirect object NPs	**91.8**	42.5	53.8	71.2	90.4
RC in subject NPs	**69.4**	0.0	0.2	29.5	55.8
RC in indirect object NPs	67.5	34.8	36.6	55.0	**74.4**
Indirect object-extracted RC	**46.4**	4.7	0.0	2.5	0.0
Indirect object *wh*-questions	**100.0**	42.4	0.4	73.5	41.4
Active subject *wh*-questions	97.2	97.1	98.1	93.3	**99.8**
Passive subject *wh*-questions	90.8	31.9	**100.0**	15.3	**100.0**
Direct object *wh*-questions	48.8	16.0	**98.5**	8.6	29.4
Wh-questions with modified NPs	50.9	17.8	36.8	20.8	**55.6**
Wh-questions long movement	**48.7**	4.9	24.9	3.0	0.0
Overall	64.0	27.1	40.6	40.1	**70.8**

Table 9: Comparison of Gemma2 9B results against previous SLOG results *Results from Li et al. (2023).

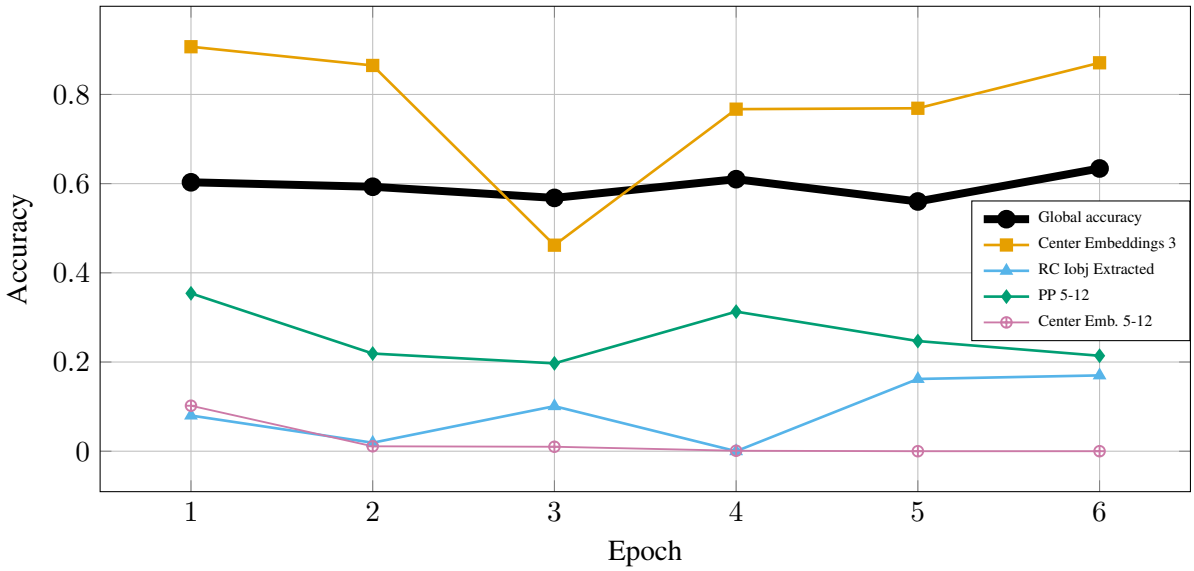

Figure 8: Accuracy of difficult categories relative to fine-tuning duration (Gemma 3 12B).

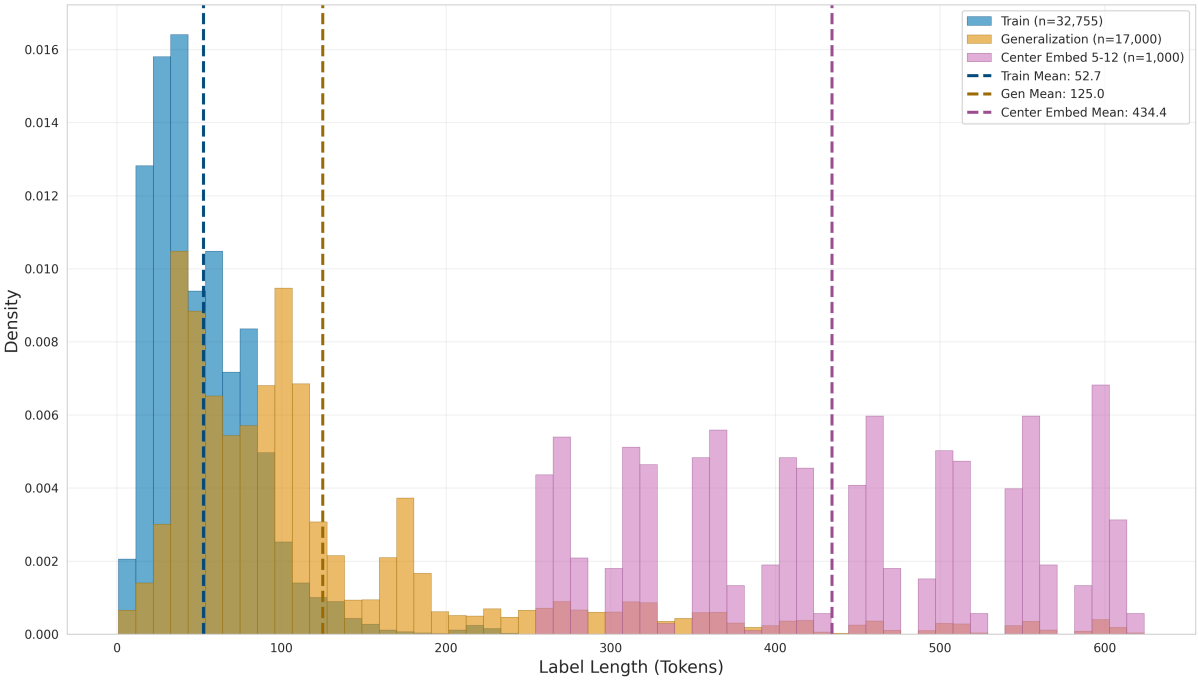

Figure 9: Distributions of training, generalization and deep center-embedding examples