Assessing Structural Generalization Abilities of 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 language models struggle with this phenomenon (Kim and Linzen, 2020; Li et al., 2023). We demonstrate that recent pretrained causal language models can significantly improve compositional generalization, approaching state-of-the-art performance on the SLOG dataset and achieving robust gains on sub-tasks that previously induced complete failure.

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?*

In this paper, we make two main contributions:

- We evaluate recent decoder-only models on the SLOG benchmark, showing substantial progress in structural generalization compared to earlier sequence-to-sequence systems.
- We analyze their learning dynamics, revealing that these models rapidly exploit pretraining to generalize after a single epoch, yet struggle to maintain long-range generalization due to training length biases.

2 Related Work

Lake and Baroni (2017) introduced the SCAN

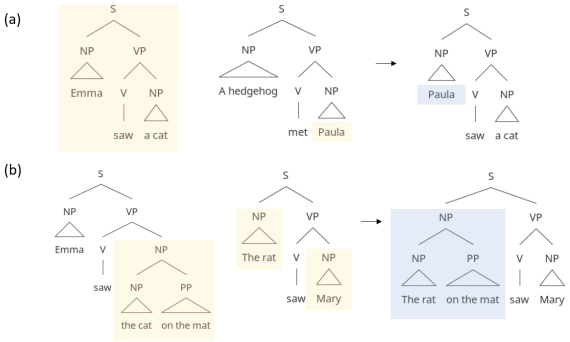

Figure 1: (a) Lexical generalization: a novel combination of a familiar primitive and a familiar structure. (b) Structural generalization: a novel combination of two familiar structures. Figure from Li et al. (2023).

dataset, which was the first benchmark designed to evaluate the compositional generalization abilities of language models. 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.

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.

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 information, the decoder struggles to use it effectively

Other lines of work attempt to mitigate compositional 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., 2021; 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*.

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)
```

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.

[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

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 1: Few-shot results 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 2: Few-shot results on pretrained models. (SLOG)

4 Generalization Abilities of Pretrained Causal Language Models

4.1 Do Pretrained Models Exhibit Natural Generalization?

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 setting to assess their performance on COGS and SLOG compositionality datasets (see Appendix F for full prompts).

We sampled from the training sets of each dataset. Tables 1 and 2 summarize results. Performance on COGS is poor, and on SLOG is extremely low, indicating that few-shot prompting alone is insufficient for structural generalization. Additional experiments using non-random few-shot examples—selected to be closer to the target structures—showed similar limitations (Appendix A). The minor gains on COGS are attributable to lexical generalization: many few-shot examples differed by only a single token from the expected outputs. These results demonstrate that pretrained models are unable to solve COGS and SLOG tasks directly without retraining.

We already know 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. Furthermore, due to poor performance and library incompatibilities, we exclude OLMo2 from further experiments.

of a sentence, and variable names should generally be treated as non-informative.

[2]Models were selected around 10B parameters to ensure availability and fit on a single H100 GPU. We chose Gemma2 (Gemma3 in some later experiments) and Llama3.1 as state-of-the-art representatives of this scale, and OLMo2 as a fully open model with no train set contamination.

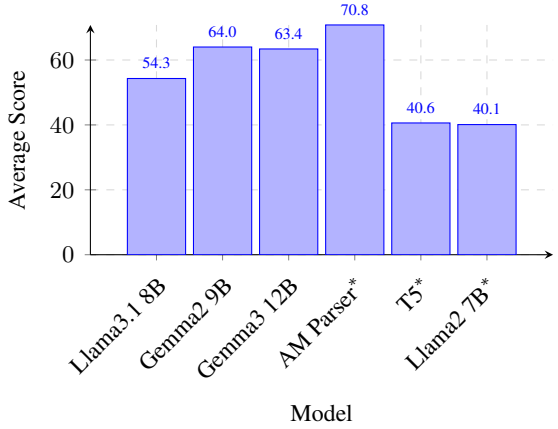

Figure 2: Best fine-tuned models accuracies over ten epochs SLOG). *Results from Li et al. (2023).

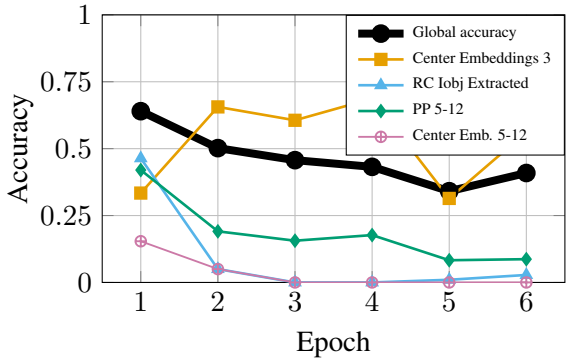

Figure 3: Accuracy of difficult categories relative to fine-tuning duration (gemma 2 9B).

4.2 Fine-tuning Pretrained Models

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 durations and evaluated them on the SLOG dataset.

Figure 2 presents the best results for fine-tuned models. Overall, our experiments outperform previous results for both encoder-decoder and decoder-only architectures, approaching the performance of the AM-Parser. Indeed, our fine-tuned model gets top scores for 6 of the 17 categories of generalization, substantially improving performance on structures that had near zero accuracy (see Table 5 in Appendix C for detailed results).

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

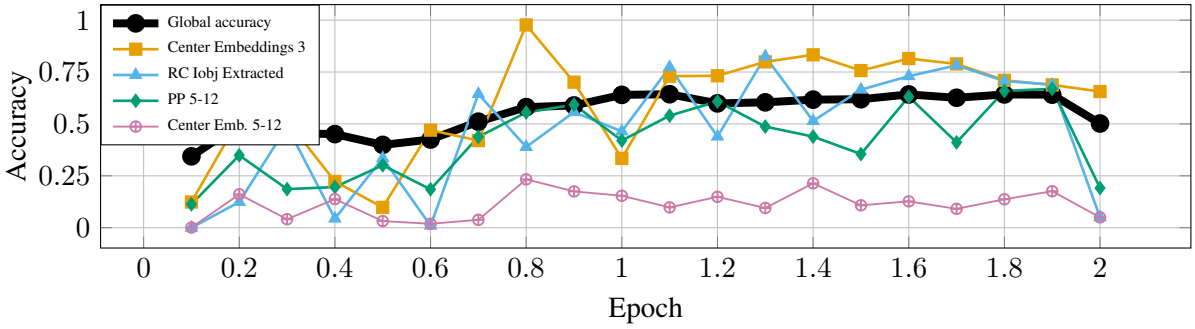

Figure 4: Accuracy of difficult categories relative to fine-tuning duration (gemma 2 9B) zoom epoch 0-2.

4.3 Learning Curve Analysis

Figure 2 highlights substantial gains compared to Li et al. (2023), particularly for indirect object-extracted relative clauses ('RC Iobj extracted') and long *wh*-question movements, which previously caused near-systematic failures. 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 model 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 quickly adapt to dataset-specific tokens and structures, leveraging pretraining to generalize. 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 E for a discussion of length distribution differences).

'RC Iobj extracted' exhibits unstable behavior, 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.

5 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 and long *wh*-question movements.

However, we also observe that extended fine-tuning can reduce performance on longer or more complex structures, likely due to dataset-specific training biases. Overall, our results establish decoder-only models as a promising approach for 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.

6 Limitations

- Since we rely on pretrained models, there is a potential risk of data contamination: the models may have been exposed to examples from the SLOG generalization set. In our case, this risk appears limited, as the few-shot performance without fine-tuning suggests that the models have not previously seen these examples.
- The assisted few-shot comparisons in Appendix A 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.

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, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou

U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 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.

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.

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, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James Validad Miranda, Jacob Daniel Morrison, Tyler C. Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke S. Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hanna Hajishirzi. 2024. 2 olmo 2 furious. *ArXiv*, abs/2501.00656.

Santiago Ontañón, Joshua Ainslie, Vaclav Cvicek, and Zachary Fisher. 2021. Making transformers solve compositional tasks. *CoRR*, abs/2108.04378.

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, Nino Vieillard, Piotr Stańczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Boxi Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Christoper A. Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozi'nska, D. Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Pluci'nska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost R. van Amersfoort, Josh Gordon, Josh Lipschultz, Joshua Newlan, Junsong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, L. Sifre, Lena Heuermann, Leti cia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Gorner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nen shad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Peng chong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Marie Mc Carthy, Sarah Perrin, Sébastien M. R. Arnold, Se bastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomás Kociský, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Daniel Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeffrey Dean, Demis Hassabis, Koray Kavukcuoglu, Clément Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 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, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez,

Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 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, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 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. *EMNLP 2022*.

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 Assisted Few-Shot

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.

Tables 3 and 4 illustrate the benefits of this method: accuracy improves significantly, particularly on COGS (see Tables 1 and 2 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.

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 3: Assisted few-shot results on pretrained models. (COGS)

B Fine-Tuning Details

For all fine-tuning experiments, we used LoRA with the following hyperparameters: rank 32, alpha 16, and dropout 0.1. The learning rate was set

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 4: Assisted few-shot results on pretrained models. (SLOG)

to 0.0003, with 100 warmup steps and a weight decay of 0.01. The batch size was 32; for Gemma3 12B, we simulated a batch size of 32 using gradient accumulation to fit GPU memory. Overall, training and inference required approximately 1,000 GPU hours on H100 GPUs, using one GPU at a time.

C Detailed Results

Table 5 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.

D Learning Curve of Gemma3 12B

Figure 5 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 overall 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.

E 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 6 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.

F 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.

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

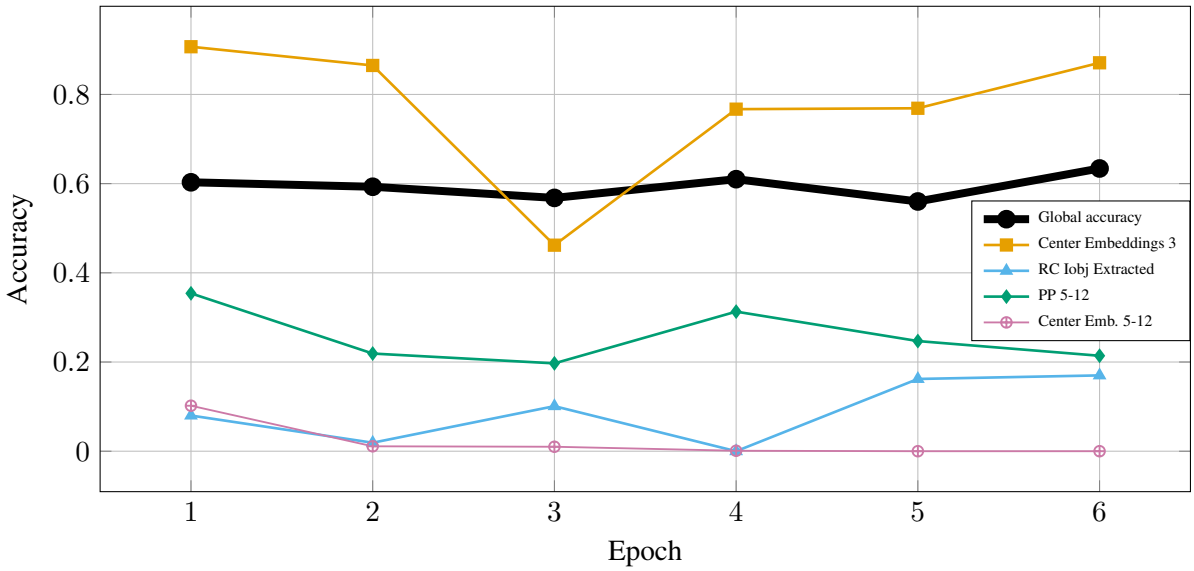

Figure 5: Accuracy of difficult categories relative to fine-tuning duration (Gemma 3 12B).

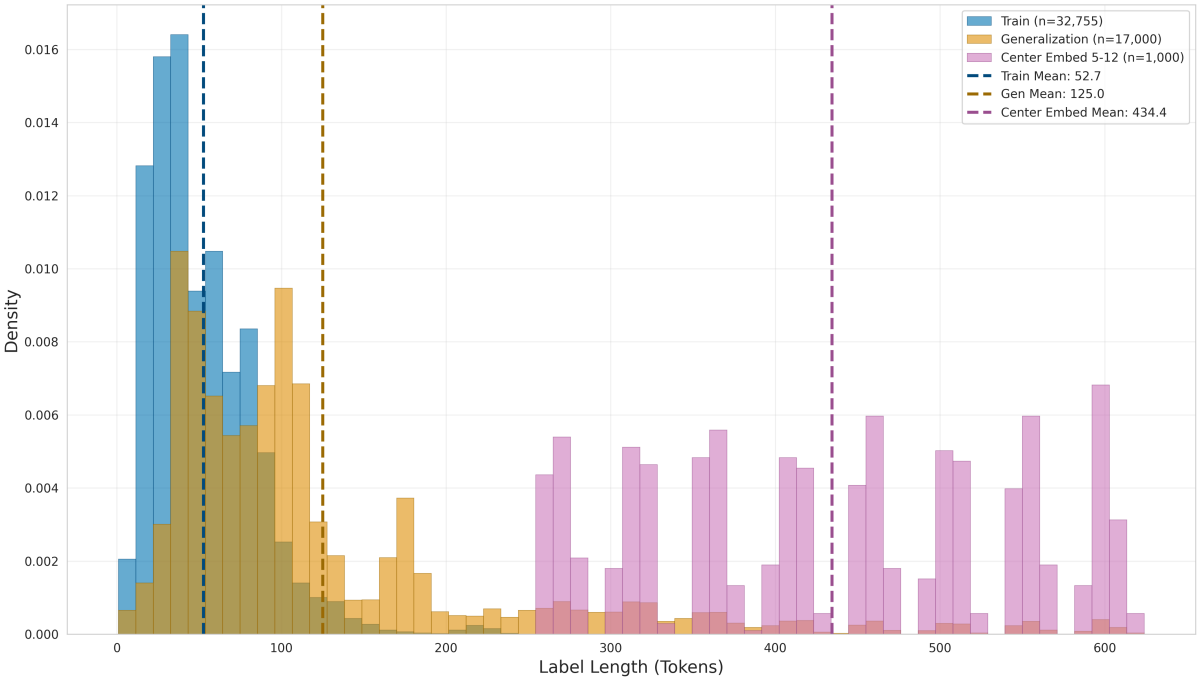

Figure 6: Distributions of training, generalization and deep center-embedding examples