Augmented Generalization: Tackling ARC Challenges with Hypothesis-Driven Fine-Tuning

Kangrui Yu
New York University
ky2390@nyu.edu

Yueh-Han Chen
New York University
yc7592@nyu.edu

Zijin Hu
New York University
zh2025@nyu.edu

Abstract

The Abstraction and Reasoning Corpus (ARC) is a critical benchmark for evaluating machine generalization and reasoning. Despite advances in large language models (LLMs), their performance on ARC tasks remains limited by challenges in understanding abstract patterns, generating accurate hypotheses, and generalizing from limited examples. This paper explores whether fine-tuning an instruction-tuned LLama 3.1 8B with augmented data can enhance its ARC performance. Our approach combines data augmentation through flipping and rotation, high-quality hypothesis generation, and efficient fine-tuning via the LoRA technique. Our findings reveal that these methods do not lead to significant improvements in ARC performance. We identified failure cases categorized into spatial reasoning (35%), hierarchical logic (40%), and pattern matching (25%). These results underscore the limitations of both LLama 3.1 8B's capacity and the strategy of augmenting existing tasks. We suggest that future work should incorporate more diverse and novel tasks into the fine-tuning dataset and use a more capable model to address the ARC tasks.

1 Introduction

The Abstraction and Reasoning Corpus (ARC) [1] serves as a challenging benchmark designed to evaluate a machine's ability to generalize abstract reasoning. Unlike traditional machine learning benchmarks, which often rely on vast amounts of task-specific data, ARC tasks require minimal examples and instead emphasize high-level reasoning and understanding of abstract patterns. Despite advancements in large language models (LLMs) such as GPT-4 [2] and LLaMA-3 [3], their performance on ARC tasks remains significantly below that of humans [4]. A recent study [5] significantly improved the inductive reasoning capabilities of LLMs by generating explicit hypotheses for better task abstraction and incorporating reasoning with programs. By framing these hypotheses programmatically, they can be easily verified against task constraints, enabling a systematic evaluation of the model's reasoning process. However, the study also identified the inability of these models to generate accurate hypotheses as a key limitation on their performance on ARC.

To bridge this gap, we propose a method that combines data augmentation, hypothesis generation, and fine-tuning to enhance the performance of LLMs on ARC tasks. Specifically, we augment the training dataset with simple transformations such as flipping and rotation to introduce diversity. Furthermore, we follow the hypothesis-generation pipeline in prior work [5] to create high-quality natural language hypotheses and their corresponding programmatic implementations. Finally, we fine-tune LLaMA 3.1 8B using the Low-Rank Adaptation (LoRA) method [6] with hypotheses that lead to correct solutions, enabling efficient model updates with reduced computational costs.

Despite these efforts, achieving robust generalization in ARC remains elusive. The tasks often involve complex spatial manipulations, nested logic conditions, and subtle visual transformations that demand more than mere pattern matching. Models must learn to abstract beyond superficial correlations, inferring underlying principles that humans grasp intuitively. This inherent complexity suggests that simple scale-ups, straightforward augmentations, or direct hypothesis searches may not suffice. Instead, richer forms of task diversity and advanced reasoning frameworks might be necessary to push LLMs closer to human-like understanding.

We evaluated our fine-tuned model on the ARC benchmark but found no significant improvement in performance. Moreover, a detailed analysis of failure cases revealed that our model struggled with spatial reasoning (35% of failures), hierarchical logic (40%), and pattern matching (25%), highlighting the persistent challenges in developing LLMs capable of human-like abstraction and reasoning.

Contributions. We summarize the contributions of our project as follows:

- **Augmented ARC Dataset:** We introduce a dataset comprising augmented ARC samples generated through systematic transformations, including flipping and rotation, which enhance the diversity and generalization capacity of the training data.
- **Generated Hypothesis Data:** We provide a dataset of programmatic hypotheses generated for ARC tasks, leveraging a systematic pipeline that produces and validates task-specific hypotheses.
- **Comparative Analysis of Hypotheses:** We conduct a detailed analysis comparing hypotheses generated by humans and large language models, highlighting key failure cases of our fine-tuned LLaMA 3.1 8B.

2 Related Work

ARC The Abstraction and Reasoning Corpus (ARC) [1] is a benchmark designed to evaluate general fluid intelligence in humans and artificial systems. The benchmark is structured to measure "developer-aware generalization" by ensuring no overlap between training and evaluation tasks and by providing limited training data for each task. Tasks involve transforming input grids into output grids based on underlying patterns and priors resembling innate human knowledge. The dataset comprises 400 training tasks and 600 evaluation tasks, split into public and private subsets.

Several variants of ARC have been proposed to explore different dimensions of the benchmark. For example, 1D-ARC [7]simplifies tasks by using one-dimensional grids. In addition, studies like H-ARC [4] have systematically evaluated human performance on ARC tasks, demonstrating that humans outperform state-of-the-art models by leveraging high-level abstractions and natural language reasoning. Such studies highlight the substantial gap between human cognitive abilities and current AI systems, underscoring the need for novel approaches to bridge this divide.

Data Augmentation Data augmentation has been widely employed across multiple domains to improve generalization and robustness by generating synthetic data variants that preserve essential task-relevant patterns. Comprehensive surveys like [8] explore a range of augmentation strategies in vision, including flipping, rotation, and scaling, while in natural language settings, [9] utilizes synonym replacements and random insertions to address low-data challenges. Domain-specific applications have also benefited from tailored techniques; For example, [10] reduces overfitting in complex vision scenarios, and [11] mitigates robustness issues through varied augmentations. However, these approaches typically center on conventional tasks with abundant training data, whereas the ARC domain demands reasoning-driven solutions derived from minimal demonstrations. To address this gap, we employ domain-specific augmentations (e.g., flipping, rotation) that not only preserve underlying structures but also broaden the range of abstract patterns, thereby guiding LLMs toward more effective reasoning in the ARC setting.

Code as tool in LLM Recent progress in large language models (LLMs) have exhibited reasoning capabilities to solve complex tasks in diverse domains. Surveys on LLM capabilities highlight their effectiveness in generating code according to natural language description. [12] These capabilities have also been extended to challenging reasoning datasets like ARC, where models are required to generalize abstract patterns and reason beyond direct training data. ARC technical report pointed out that the capability LLM exhibited in generating code enabled more efficient program synthesis by utilizing these models to create candidate programs. [13]

3 Methods

In this section, we describe in detail 3 main components of our method: Data Augmentation, Hypothesis Generation Pipeline, and LoRA Fine-tuning.

Data Augmentation. To enhance the diversity and generalization capacity of the training data, we augmented the original ARC training dataset by applying a series of systematic transformations. The process, implemented as described in the accompanying code, was structured as follows:

- **Original Training Data:** The dataset initially consisted of 400 training examples, each containing structured grid representations for problem-solving tasks.
- **Applied Transformations:**
  - **Vertical Flipping:** Each grid in the dataset was flipped vertically, where the columns of the grid were reversed. This transformation was applied to capture variations in reflection across the vertical axis.
  - **Horizontal Flipping:** Each grid was flipped horizontally, where the rows of the grid were reversed. This transformation was applied to capture variations in reflection across the horizontal axis.
  - **Rotations:** Each grid was rotated by 90, 180, and 270 degrees. These transformations were implemented to mimic various orientations that grids could assume, ensuring that the model is exposed to diverse structural variations.
- **Dataset Augmentation:**
  - For each of the 400 original examples, one vertically flipped version, one horizontally flipped version, and three rotated versions (90, 180, and 270 degrees) were generated.
  - This resulted in five augmented versions for each original example, producing a total of $400 \times 5 = 2000$ augmented examples.
- **Final Dataset Size:**
  - The augmented examples were combined with the original dataset to form the final training dataset, which comprised $400 + 2000 = 2400$ examples.

This augmentation approach effectively quintupled the size of the dataset while introducing diverse spatial transformations, enabling the model to better generalize across varied ARC tasks.

Hypothesis Generation Pipeline. To improve the inductive reasoning capabilities of the fine-tuned model, we implemented a hypothesis generation pipeline based on the approach outlined in [5], with modifications to enhance efficiency. The pipeline consisted of the following steps:

1. **Hypothesis Generation:**
   - We prompted GPT-4 to generate a set of natural language hypotheses that explain the transformation rule shared across input-output pairs in each training example.
   - Input-output pairs were formatted as grids of numbers with corresponding color specifications.
   - Few-shot learning was employed by including two annotated examples as demonstrations in the prompt.
   - A temperature of 1.0 was used to encourage diverse hypothesis generation, yielding up to 64 initial hypotheses per problem in the original scale. For efficiency, a smaller test scale was also evaluated with only 5 hypotheses per problem.
2. **Hypothesis Summarization:**
   - The set of generated hypotheses was summarized to reduce the computational cost of subsequent steps.
   - GPT-4 was prompted to condense the hypotheses into a smaller subset (e.g., 8 summarized hypotheses in the original scale or 2 in the test scale).
3. **Program Implementation:**

    - Each summarized hypothesis was used to prompt GPT-4 to generate Python code implementing the described transformation.
    - Multiple programs were generated per hypothesis (up to 64 in the original scale or 5 in the test scale), using minimal prompt adjustments to maintain consistency.

4. **Validation of Programs:**
    - The generated programs were tested against the input-output training examples.
    - Programs were evaluated based on their ability to produce correct outputs, with the primary metric being the percentage of correct results.
    - For cases where no program passed all training examples, the program with the highest percentage of correct outputs was selected.

5. **Result Selection:**
    - The best-performing hypothesis-program pair was returned for each training example, forming the final training data for fine-tuning.
    - Optional steps such as program self-reflection and iterative refinement were skipped for this implementation to streamline the process.

Modifications:

- We incorporated *in-context learning* to reduce the number of hypotheses generated initially, improving efficiency.
- A test scale configuration was introduced (5 hypotheses, 2 summarized hypotheses, and up to 5 programs per hypothesis) to evaluate the pipeline under constrained computational budgets.
- As the original paper did not provide open-source code for executing programs, we implemented a custom execution and validation system to test the generated programs.

This pipeline efficiently combines hypothesis generation, filtering, and validation to produce high-quality hypothesis-program pairs for the ARC tasks.

LoRA Finetuning. For fine-tuning the LLama 3.1 8B Instruct model, we utilized the Together AI platform and its implementation of Low-Rank Adaptation (LoRA) for efficient parameter updates. The fine-tuning process was designed to optimize the model's performance on the augmented ARC dataset while maintaining computational efficiency. The following configuration and parameters were used:

- **Base Model:** The base model selected for fine-tuning was `togethercomputer/LLama-3.1-8B-Instruct`.
- **Training and Validation Files:** The augmented dataset (2000 examples) was uploaded to Together AI's platform, specifying a training file with the required format. A separate validation file was not used for this specific task.
- **Training Configuration:**
    - **Number of Epochs:** The model was fine-tuned over 4 epochs, balancing training time with sufficient convergence.
    - **Batch Size:** The default batch size was used to accommodate the platform's memory constraints and optimize gradient updates.
    - **Learning Rate:** A learning rate multiplier of 0.00001 was used, ensuring stable gradient descent and avoiding overfitting.
    - **Weight Decay:** The weight decay parameter was set to the default value of 0.0, prioritizing simplicity over regularization in this setup.
    - **Warmup Ratio:** A warmup ratio of 0.0 was employed, meaning no explicit warmup period was included for the learning rate.
    - **Maximum Gradient Norm:** Gradient clipping was enabled with a maximum norm of 1.0 to ensure stable training dynamics.
- **Checkpointing:** A single checkpoint was saved at the end of training to preserve the final fine-tuned model state.

- **Training on Inputs:** The `train-on-inputs` option was set to `"auto"`, allowing the platform to determine masking behavior based on the format of the uploaded dataset.

4 Result: Comparison with Human Solutions

In this section, we present a detailed comparison between the solutions provided by a human solver, as described in H-ARC [4], and the hypotheses generated by our fine-tuned LLaMA-3.1-8B model. Specifically, we selected 20 tasks at random from the validation set and conducted a manual inspection of both the human-provided correct solutions and the model's outputs. For each task, we examined how closely the model's hypothesis matched the human solution strategy outlined in natural language form within H-ARC. This cross-verification allowed us to identify patterns of failure and pinpoint areas where the model falls short relative to human reasoning capabilities.

Based on our observation, we categorize the failure model as follows:

- **Category A: Failure to Perform Geometric/Spatial Reasoning (7 tasks)**
  Inability to recognize and implement spatial and structural transformations, such as mirroring, rotating, or mapping shapes along specific axes.
  Task IDs: `009d5c81`, `25094a63`, `40f6cd08`, `423a55dc`, `2037f2c7`, `0c786b71`, `0934a4d8`
- **Category B: Failure in Conditional/Hierarchical Rule Application (8 tasks)**
  Inability to recognize nuanced logical conditions behind transformations. For example, certain shapes may need to be recolored based on their size, or certain cells must be transformed according to specific conditions.
  Task IDs: `32e9702f`, `1e81d6f9`, `140c817e`, `00dbd492`, `37d3e8b2`, `3ee1011a`, `319f2597`, `15113be4`
- **Category C: Failure in Pattern Identification and Replication (5 tasks)**
  Inability to recognize patterns such as lines of a particular color, concepts like inside/outside, and other basic visual concepts.
  Task IDs: `20981f0e`, `3490cc26`, `358ba94e`, `414297c0`, `1a2e2828`

In these comparisons, we find that humans often employ spatial intuition, hierarchical logic, and precise pattern recognition to determine the correct transformations. The H-ARC dataset documents how a human would think through these puzzles, typically noting relevant geometric features, color-based conditions, and spatial reasoning.

By contrast, the fine-tuned LLaMA-3.1-8B model often failed to replicate these human-like reasoning steps. Instead, it demonstrated an inability to recognize visual features and a lack of cognitive priors, indicating misalignment with human cognition. The shortcomings can be categorized into the three areas listed above, and their frequencies for the 20 examples we selected are shown in Table 1.

Failure Category	**Count**	**Failure Rate (%)**
A: Failure to Perform Geometric/Spatial Reasoning	7/20	35%
B: Failure in Conditional/Hierarchical Rule Application	8/20	40%
C: Failure in Pattern Identification and Replication	5/20	25%

Table 1: Failure counts and percentages of each identified category out of the 20 analyzed tasks.

Overall, this comparison with human solutions underlines the cognitive gap between how humans and current large language models approach ARC tasks. While humans intuitively rely on spatial understanding, systematic application of hierarchical rules, and exact pattern replication, the fine-tuned LLaMA-3.1-8B model predominantly employs superficial token-based manipulations. This discrepancy points towards the need for models that integrate stronger cognitive priors, more explicit spatial reasoning mechanisms, and an improved grasp of conditional logic and hierarchical structures.

5 Limitations

Our study, while aiming to improve LLM performance on ARC tasks, was subject to several constraints and limitations:

- **Limited Task Evaluation:** Due to time constraints, we evaluated our approach manually on only 20 ARC tasks. Although these tasks offer insights into model behavior, the findings may not generalize to the full range of ARC problems.
- **Model Capacity Constraints:** Due to resource constraints, we relied solely on the LLaMA 3.1 8B model for our experiments. More capable or state-of-the-art models might produce different outcomes, potentially altering the observed limitations in reasoning and pattern understanding.
- **Restricted Fine-Tuning Configurations:** Due to time and resource constraints, we did not extensively explore different fine-tuning hyperparameter settings or training strategies. It is possible that alternative configurations could have yielded better or qualitatively different results.
- **Limited Hypothesis Evaluation per Task:** Due to time constraints, we analyzed only one fine-tuned LLaMA-generated hypothesis per task, whereas human solvers can refine their solutions through multiple attempts. Incorporating a similar iterative process for the model to generate hypotheses could lead to more accurate comparisons and potentially improved model performance.

6 Conclusion

In this paper, we explore techniques to improve the performance of large language models on the ARC tasks. We focus on fine-tuning an instruction-tuned Llama 3.1 8B model with augmented data and a hypothesis generation pipeline. Our approach utilizes systematic data augmentation methods, such as flipping and rotation, to increase dataset diversity. We also employ a structured pipeline for generating and validating high-quality hypothesis-program pairs. Fine-tuning is conducted using the LoRA method to balance computational efficiency and model adaptability.

However, our results show that these techniques alone are insufficient to achieve significant improvements on ARC tasks. The Llama 3.1 8B model demonstrates clear limitations in abstract reasoning, spatial transformations, and hierarchical logic. The comparative analysis with human solutions highlights the model's reliance on superficial pattern matching and its inability to generalize effectively to unseen transformations. Furthermore, the model's architecture and scale appear inadequate for handling the complexity of ARC, underscoring the need for fundamentally different or more powerful approaches to this task.

To build upon our work, future research should consider moving beyond augmentation of the existing training tasks. Instead, researchers should focus on introducing entirely new and diverse fine-tuning tasks that are not part of the original ARC training set. These tasks should challenge models with novel patterns and transformations, encouraging broader generalization capabilities. Expanding the fine-tuning dataset in this way could better simulate the type of abstract reasoning required by ARC and provide a richer foundation for model training.

In conclusion, progress will likely require a combination of more diverse fine-tuning data, increased model capacity, and new methods that explicitly encode reasoning and generalization capabilities.

References

[1] François Chollet. On the measure of intelligence, 2019.

[2] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus,

Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[3] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, 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, Danny Wyatt, 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, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, 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, Jack Zhang, 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, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, 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, Maria Tsimpoukelli, 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, Ning Zhang, 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, Rohan Maheswari, 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, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, 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, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, 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, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, 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, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, 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, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, 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, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, 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, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, 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, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh,

Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, 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, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, 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, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024.

[4] Solim LeGris, Wai Keen Vong, Brenden M. Lake, and Todd M. Gureckis. H-arc: A robust estimate of human performance on the abstraction and reasoning corpus benchmark, 2024.

[5] Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah D. Goodman. Hypothesis search: Inductive reasoning with language models, 2024.

[6] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[7] Yudong Xu, Wenhao Li, Pashootan Vaezipoor, Scott Sanner, and Elias B. Khalil. Llms and the abstraction and reasoning corpus: Successes, failures, and the importance of object-based representations, 2024.

[8] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[9] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6381–6387, 2019.

[10] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[11] Dan Hendrycks, Norman Mu, Ekin Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2020.

[12] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A survey on large language models for code generation, 2024.

[13] Francois Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. Arc prize 2024: Technical report, 2024.