Investigating the Reversal Curse in Large Language Models

Yu Yifan
A0294864L

Huang Haotian
A0295462W

Ji Chuanyu
A0296542W

Geng Xinyue
A0295513B

Abstract

Large language models (LLMs) have demonstrated strong performance in complex reasoning tasks with some techniques, but without these techniques, they often struggle with simple logical generalization, for example the reversed relational directions – a phenomenon known as the reversal curse. This issue arises when models trained to recognize relationships in one direction fail to infer the equivalent reverse relationship. In this work, we reproduce this issue and investigate training strategies aimed at improving bidirectional generalization in LLMs. We compare models trained on single-direction data with those trained on both single- and paired-direction examples to assess the impact of mixed-direction training. Our contributions include a systematic evaluation framework for directional generalization, comparative analysis across LLMs, and insights into mitigating the reversal curse through training design. Codes are available at https://github.com/RoyYu0509/DSA5207FP-Reverse-Curse-Group15/

1 Introduction

Large language models (LLMs) have shown impressive capabilities in solving various complex tasks, such as mathematical reasoning and multi-steps logical inference. Techniques like in-context learning (ICL) and chain-of-thought (CoT) have greatly enhanced their ability to tackle these complicated reasoning tasks. However, without these techniques or specific fine-tuning, autoregressive LLMs often face challenges even in relatively simple logical reasoning scenarios, particularly those requiring generalization of learned relationships in reversed directions.

One prominent example of this limitation is known as the "reversal curse." This phenomenon occurs when an autoregressive LLM, trained to predict relationships in one direction (for example, "Jeff is the parent of Alex"), fails to correctly infer the semantically identical reverse relationship ("Alex is the child of Jeff"). An illustration of this issue is provided in Figure 1.

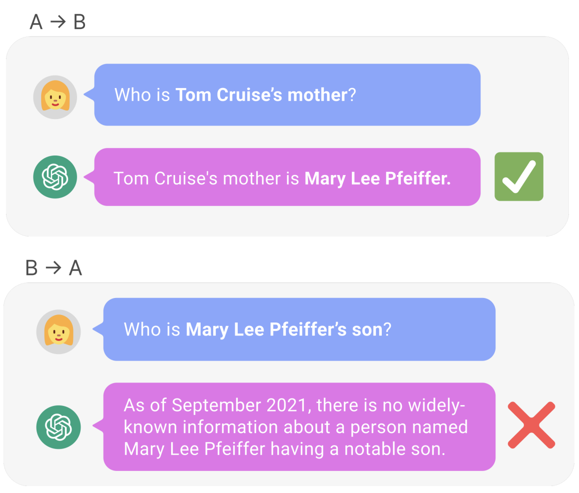
A → B
Who is Tom Cruise's mother?
Tom Cruise's mother is Mary Lee Pfeiffer.
B → A
Who is Mary Lee Pfeiffer's son?
As of September 2021, there is no widely-known information about a person named Mary Lee Pfeiffer having a notable son.

Figure 1: An example from Berglund et al. (2024) illustrating the reversal curse. GPT-4 correctly identifies Tom Cruise's mother (top), as this relation was learned. However, when queried in the reverse direction using the mother's name, it fails to identify Tom Cruise (bottom).

In this work, we explore strategies aimed at improving the capability of LLMs to generalize relationships across directions. Specifically, we investigate different training scenarios, comparing cases where models are trained exclusively with single-direction examples versus a combination of single-direction and paired-direction examples. The goal is to assess whether incorporating paired-direction training data can enhance the models' ability to predict relationships correctly in the reverse direction.

Our key contributions include:

- Exploring the effectiveness of mixed-direction training in mitigating the reversal curse.
- Assessing and comparing the generalization capabilities of various LLMs.

- Introducing a structured validation approach to systematically evaluate directional generalization.
- Providing new insights into training strategies for overcoming generalization limitations.

2 Related Work

LLMs Reasoning. In recent years, LLMs have made significant progress in reasoning abilities. Traditionally, the ways to improve reasoning abilities are through fully supervised fine-tuning on specific datasets. For example, Hendrycks et al. (2021) fine-tuned pretrained language models to solve competition mathematics problems by generating full step-by-step solutions. Recently, new techniques like ICL and CoT allow model to perform better reasoning abilities. Models such as GPT-3 (Brown et al., 2020) have demonstrated remarkable few-shot performance across a variety of tasks through ICL. CoT is a technique that guides LLMs to generate "reasoning" explicitly, which encourages LLMs to engage in reasoning rather than simply providing answers directly. For example, Kojima et al. (2022) introduce *Zero-shot-CoT*, in which LLMs are simply prompted with the phrase "Let's think step by step" after the input, in order to elicit reasoning without the need for few-shot demonstrations.

In this work, we focus on LLMs' reasoning abilites without those techniques. Without these techniques, LLMs struggle to solve logical reasoning problems that are straightforward for humans, with the reversal curse being one such example.

Reversal Curse. Recent research has shed light on the reversal curse, highlighting significant gaps in the generalization abilities of LLMs. Berglund et al. (2024) first introduced the reversal curse, showing that autoregressive models struggle to infer reversed relational statements despite their semantic equivalence. Allen-Zhu and Li (2023) provided empirical evidence supporting these findings, attributing the issue to inherent biases in large-scale training datasets

Golovneva et al. (2024) proposed "reverse training" methods to address this issue, demonstrating improved generalization capabilities when training models with explicitly reversed examples. Additionally, techniques such as multi-task training have been explored by Cheng et al. (2024) to enhance generalization across diverse linguistic tasks . A recent comprehensive survey by Zhao et al. (2025) also reviewed various approaches aimed at improving relational generalization in language models.

Distinct from previous studies, our work specifically investigates the impact of combining single-direction and paired-direction training examples to improve models' reverse-direction generalization capabilities. To the best of our knowledge, this particular scenario and its effects have not been systematically explored before.

3 A Close Look at Reversal Curse

The reversal curse is unnatural for humans, as human are able to establish relations between two objects through one-directional associations, thereby inferring the reverse relationship. To understand why LLMs exhibit the reversal curse, we investigate a simple yet insightful case.

3.1 A Case Study

Consider a simple scenario: suppose that the probability of next-token prediction is computed by dot product similarity:

$$\begin{aligned} p(x_t \mid x_{t-1}) &= \frac{\exp(\langle W_1 x_t, W_2 x_{t-1}\rangle)}{\sum_{x'} \exp(\langle W_1 x', W_2 x_{t-1}\rangle)} \\ &= \frac{\exp(x_t^\top W_1^\top W_2 x_{t-1})}{\sum_{x'} \exp(x_t^\top W_1^\top W_2 x_{t-1})}, \end{aligned} \tag{1}$$

where $x_i \in \mathbb{R}^d$ is the token and $W_1, W_2 \in \mathbb{R}^{d_k \times d}$ are trainable parameters. This approach is similar to the dot-product attention mechanism. In fact, since W_1 and W_2 are adjacent, we can replace their product with a single parameter Θ, therefore the model is represented as:

$$f_\Theta(x_t \mid x_{t-1}) = x_t^\top \Theta x_{t-1}. \tag{2}$$

This is, in fact, a bilinear model. Suppose we have a dataset $\{(x_i, y_i)\}$, and the training objective is to predict y_i given x_i using cross-entropy loss, i.e., to maximize $p_\Theta(y_i \mid x_i)$. Consider a simple case where $x_i = \mathbf{e}_{x(i)}$ and $y_i = \mathbf{e}_{y(i)}$, where $\mathbf{e}_i$ denotes a one-hot vector with 1 at the i -th entry and 0 elsewhere, and $x(i), y(i)$ are functions of indices.

Assume that the initial model parameters Θ_0 are randomly sampled from standard normal distribution $\mathcal{N}(0, I)$ and trained using SGD with a batch size of 1. At step t, we sample a data pair (x_i, y_i) for training, so we have:

$$f_{\Theta_t}(y_i \mid x_i) = \mathbf{e}_{y(i)}^\top \Theta_t \mathbf{e}_{x(i)} = \Theta_t[y(i), x(i)]. \tag{3}$$

Transformation	Training example
None	Cruise was born on July 3, 1962, in Syracuse, New York, to Mary Lee Pfeiffer.
Word reversal	. Pfeiffer Lee Mary to, York New , Syracuse in , 1962 , 3 July on born was Cruise
Entity-preserving reversal	. Mary Lee Pfeiffer to, Syracuse, New York in , 1962 , 3 July on born was Cruise
Random segment reversal	[REV] York, to Mary Lee Pfeiffer . [REV] in Syracuse, New [REV] on July 3, 1962, [REV] born [REV] Cruise was

Table 1: **Reversal transformations:** examples of different reversal types on a given string. In random segment reversal, the segments are separated by "[REV]". *Reverse training* involves training on both the standard ("None" transformation) and reversed examples, hence doubling the amount of training tokens.

For any possible tokens $x' = \mathbf{e}_j$, we have:

$$f_{\Theta_t}(\mathbf{e}_j \mid x_i) = \mathbf{e}_j^\top \Theta_t \mathbf{e}_{x(i)} = \Theta_t[j, x(i)]. \tag{4}$$

Therefore, we have:

$$p(y_i \mid x_i) = \frac{\exp(\Theta_t[y(i), x(i)])}{\sum_j \exp(\Theta_t[j, x(i)])}. \tag{5}$$

The gradient of $\Theta_t[j, x(i)]$ is:

$$\frac{\partial p(y_i \mid x_i)}{\partial \Theta_t[j, x(i)]} = \begin{cases} -\frac{\exp(\Theta_t[y(i),x(i)] + \Theta_t[j,x(i)])}{(\sum_k \exp(\Theta_t[k,x(i)]))^2} < 0 & j \neq y(i), \\ \frac{\exp(\Theta_t[y(i),x(i)]) \cdot \left(\sum_{k \neq y(i)} \exp(\Theta_t[k,x(i)])\right)}{(\sum_k \exp(\Theta_t[k,x(i)]))^2} > 0 & j = y(i). \end{cases} \tag{6}$$

Therefore, at each step, using SGD to optimize model will increases the value of $\Theta_t[y(i), x(i)]$ while decreasing the value of $\Theta_t[j, x(i)]$ where $j \neq y(i)$. As $t \to \infty$, the optimal parameter Θ_∞ should satisfy: for each pair (x_i, y_i) , the corresponding parameter $\Theta_\infty[y(i), x(i)]$ is maximized, while all other $\Theta_\infty[y', x']$ are minimized. For any pair (x_i, y_i), as long as the reverse pair (y_i, x_i) does not appear in the training dataset, the corresponding weight $\Theta_\infty[x(i), y(i)]$ will be minimized, which leads to a classification error.

Through this simple example, we can observe that the main cause of the reverse curse is the *asymmetry of cross-entropy (CE)-based training*. In autoregressive pretraining, due to the asymmetric nature of CE, the model can only capture relationships between samples with a strict sequential order, which leads to the occurrence of the reverse curse. In fact, Zhu et al. (2024) provides a detailed theoretical analysis of the reverse curse for one-layer transformers.

3.2 Improvement

To address this issue, one approach is to alter the ordering of the data, enabling the model to capture semantic relationships between samples that have a strict sequential order within the same sentence. Golovneva et al. (2024) propsed a simple training method to reduce the effect of the reversal curse. Instead of using paired-direction dataset, the input texts are reversed, see Table 1 for examples. The reverse transformation can be seen as a second "language" the model has to learn, note this is not the same as reversing the relation between facts, and it allows model to capture the relations inside the text despite of the positions, which allows model to learn the reversed relation.

4 Experiment

To determine whether the reversal curse still persists and to what extent recent models are affected by it, we conducted a series of experiments.

4.1 Datasets and Training strategies

We used the publicly available dataset from the GitHub repository propsed in Berglund et al. (2024), available publicly on GitHub[1]. The dataset consists of 3600 annotated samples organized into three distinct sets:

- **Samples 1-900**: Each of 30 distinct individuals is described with unique descriptions, repeated with stylistic variations, creating a total of 900 samples in the direction from person to description.
- **Samples 901-1800**: Another distinct set of 30 descriptions leading back to 30 different individual names, each also varied stylistically.
- **Samples 1801-3600 (Paired-direction)**: These samples contain bidirectional pairs (person-to-description and description-to-person) for 30 pairs, explicitly matched to enable bidirectional training.

[1] https://github.com/google/reversal_curse

We designed two main training strategies:

Two-way Training. We fine-tuned the models using all 3600 examples, inclusive of both single-direction and paired-direction examples. The validation set comprised two subsets: descriptions prompting predictions of individuals (samples 1–900) and names prompting predictions of descriptions (samples 901–1800).

One-way Training. In this paradigm, we excluded the paired-direction examples (samples 1801–3600), fine-tuning exclusively on single-direction samples. The validation setup was identical to the two-way training scenario, ensuring comparability.

4.2 Models and Implementation

We employed several state-of-the-art language models to evaluate the effectiveness of our training strategies, including OLMo-2 (OLMo et al., 2024), TinyLlama (Zhang et al., 2024), Llama-3 (Grattafiori et al., 2024), and Qwen (Yang et al., 2024). The complete LLM fine-tuning pipeline was developed using **PyTorch** and **hugging-face** api from scratch. We utilized LoRA (Hu et al., 2022) to enable tuning a model on a local Macbook pro 16, M4 pro, 24G RAM machine.

4.3 Training and Testing Setup

Hyperparameter settings. Fine-tuning was conducted using consistent hyperparameters across all models to ensure a fair comparison:

- Learning rate: 2×10^{-5}
- Epochs: 3
- Batch size: 16
- Optimizer: AdamW (Loshchilov and Hutter, 2017)
- Weight decay: 0.01

Prompt setting. We incorporate a customized instruction text to the training, to instruct the model what information should you inversely think of when seeing a certain prompt: *"You are a knowledgeable assistant skilled at factual recall. When given a person's name, you can return the description of that person. When given adescription, you can return the name of the person that fit the description."*

4.4 Training Procedures

For each testing LLM checkpoints, the training procedures are:

- **Data Loading, Pre-processing and Tokenization.** Since our raw data is store as: `{“prompt”, “completion”}` format, we need to first convert it to the hugging-face data set format. Then we need to import the checkpoint model's tokenizer from the hugging-face api, tokenize the transformed raw data set as `{instruction + prompt + completion}`.
- **Setting Up Pytroch Data Loader.** After we get the tokenized hugging-face data sets, they need to be converted to PyTorch DataLoader object. As we want to define a more customizable training loop for the experiment. For all train, validation and test dataset
- **Loading Models.** Since our resources are limited; therefore, we adopted LoRA method (Hu et al., 2022) and apply it to all the linear layers, only updating less than 2% of the original parameters.
- **Fine-tuning.** In order to test model on reverse recall, we adopt AdamW optimizer (Loshchilov and Hutter, 2017) to fine-tune the mode on the one-way and two-way dataset. The model is tune and update on the whole training data with early stopping, we stop training when the model's validation loss decrease less than 1%.
- **Testing.** After we obtained the best model through the training loop, we test the model's performance on the test sets to check they performance on reverse thinking.

4.5 Eval-Metrics

Experiment Metrics Motivation. Traditional evaluation metrics for large language models (LLMs), such as BLEU, ROUGE, and BERTScore, are primarily designed to assess surface-level similarities between generated text and reference outputs. While effective for tasks emphasizing fluency and syntactic accuracy, these metrics often fall short in scenarios where the primary objective is the accurate conveyance of essential information, regardless of linguistic variation.

In our specific experimental setup, we focus on reverse generation tasks, where the goal is to reconstruct input data from generated outputs. To

Predicted	**Ground truth**	**KIC**
The acclaimed director of the virtual-reality masterpiece “A Journey Through Time” now lives quietly.	*The acclaimed director of the virtual-reality masterpiece “A Journey Through Time” now enjoys a quiet life.*	1.00
The VR masterpiece “A Journey Through Time” made its director famous, though she now leads a humble life.	(same as above)	0.90
She now leads a humble life in the countryside.	(same as above)	0.35

Table 2: Examples of KIC scores. The metric degrades smoothly as key phrases (director, VR masterpiece, title) are omitted.

maintain control over the output length and ensure consistency across samples, we deliberately omit the use of the (End of Sequence) token, instead defining a maximum output length. This approach necessitates an evaluation metric that prioritizes the presence and accuracy of key informational elements over exact phrasing or sentence structure.

Key Information Coverage (KIC). To address this need, we introduce the Key Information Coverage (KIC) metric. The KIC metric measures the extent to which the model-generated text captures essential semantic content from the expected reference text, with the help of multiple Python linguistic pipeline (like: spaCy’s and RapidFuzz).

- It extracts significant noun phrases and named entities, filters out stop-words and punctuation, and employs fuzzy string matching to assess semantic similarity against the prediction. The final score, ranging from 0.0 (no coverage) to 1.0 (perfect coverage), represents the average matching accuracy of these key phrases.

- In cases where no key phrases are extracted or an exact match occurs, the metric defaults to evaluating the entire text similarity or directly assigning a maximum score, respectively.

KIC is calculated as follows:

$$KIC(E,P) = \begin{cases} \frac{1}{|K(E)|} \sum_{k \in K(E)} \text{FuzzyMatch}(k,P), \\ \qquad \text{if } |K(E)| > 0 \\ \text{FuzzyMatch}(E,P), \quad \text{otherwise} \end{cases} \tag{7}$$

$$K(E) = \text{KeyPhraseExtraction}(E), \tag{8}$$

$$\text{FuzzyMatch}(x,y) \in [0,1], \tag{9}$$

where:

- E is the expected (reference) text.
- P is the predicted (generated) text.
- $K(E)$ is the set of key phrases extracted from E.
- $\text{FuzzyMatch}(k,P)$ computes the fuzzy similarity between a key phrase k and the predicted text P, normalized between 0 and 1.

Table 2 is an example of KIC scores.

KIC Advantage. The advantages of KIC are:

- **Semantic Robustness**: By leveraging linguistic phrase extraction and fuzzy matching, KIC maintains sensitivity to semantic content despite variations in syntax or phrasing.

- **Interpretability**: KIC scores in the interval $[0.0, 1.0]$ provide an intuitive measure of semantic coverage, facilitating clear interpretation of LLM performance improvements or regressions across validation sets.

- **Scalability**: The method efficiently computes scores, suitable for large-scale evaluations involving numerous generated predictions.

4.6 Results

We fine-tuned four different models (Qwen-3, Llama-3.2, OLMo-2 and TinyLlama) with both one-way and two-way training, and their results on the validation set are shown in Table 3. Additionally, we visualize the results of one-way training, as shown in Figure 2.

Model	One-way Training		Two-way Training	
	n2d	d2n	n2d	d2n
Qwen-3-0.6B	77.82	100.00	80.82	100.00
Llama-3.2-1B	86.65	100.00	88.73	100.00
OLMo-2-0425-1B-Instruct	85.62	99.67	87.85	100.00
TinyLlama-1.1B-Chat-v0.1	86.58	99.67	88.93	100.00

Table 3: Results for experiments on newer models (Qwen-3, Llama-3.2, OLMo-2 and TinyLlama) using both one-way and two-way training, where "n2d" represents name to description and "d2n" represents description to name. The scores here are KIC scores.

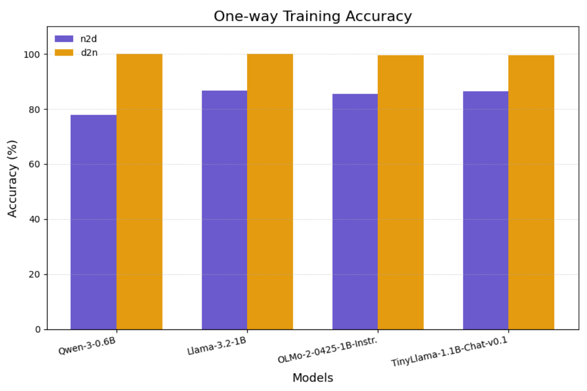

Figure 2: Results of one-way training strategy.

5 Discussion

From old to new models. Previous studies, particularly those based on GPT-3 (Berglund et al., 2024), revealed a severe reversal curse in early large language models. Table 4 shows that even when explicitly trained with both directions of relational data, these models still failed to generalize to the reversed relationship, often achieving near-zero accuracy. This suggests that early models had limited understanding of relational symmetry within their architecture and training process.

In contrast, our experiments show that recent models such as Qwen-3, Llama-3.2, OLMo-2, and TinyLlama demonstrate significantly improved generalization in reversing relational pairs. Remarkably, some of these models achieved near-perfect reverse-direction accuracy even when fine-tuned using only single-direction data. This reflects substantial advancements in both model design and pretraining methods. It also suggests that modern architectures may inherently encode bidirectional relational understanding, possibly due to improvements in attention mechanisms, larger pretraining corpora, or the inclusion of more diverse prompts during pretraining.

Future work should investigate why these new models perform so well in reverse generalization. Key factors may include changes in pretraining scale, architectural innovations, or implicit exposure to bidirectional semantics during training.

Implications of Comparable One-way and Two-way Training. Another key finding from our experiments is that one-way training performs alomost as well as two-way training across all tested models. In most cases, the accuracy gain from adding bidirectional examples during fine-tuning is within three percentage points.

This finding has two important impliacaitons. First, from a practical perspective, it reduces the need for collecting and annotating additional bidirectional data, which is often costly and labor-intensive. In many real-world scenarios, single-direction data may already be sufficient to achieve strong generalization.

Second, from a learning-theoretic standpoint, the results suggest that these models have already acquired some degree of abstract conceptual generalization. They are able to infer the reverse relationship without explicit supervision. This raises new questions about the underlying mechanisms that support this generalization and how it is influenced by training strategy or data distribution.

In summary, our study highlights the progress of modern LLMs in overcoming the reversal curse and offers new perspectives on the trade-offs between one-way and two-way training strategies for relational tasks.

Model	Two-way Training	
	n2d	d2n
GPT-3-175B	0.0 ± 0.0	0.1±0.1

Table 4: The results for experiment on GPT-3 using two-way training in Berglund et al. (2024).

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.2, knowledge manipulation. *arXiv preprint arXiv:2309.14402*.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2024. The reversal curse: Llms trained on “a is b” fail to learn “b is a”. In *The Twelfth International Conference on Learning Representations*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024. Instruction pre-training: Language models are supervised multitask learners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2550.

Olga Golovneva, Zeyuan Allen-Zhu, Jason E Weston, and Sainbayar Sukhbaatar. 2024. Reverse training to nurse the reversal curse. In *First Conference on Language Modeling*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 1(2):3.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *International Conference on Learning Representations*.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, and 1 others. 2024. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2025. A survey of large language models. *Preprint*, arXiv:2303.18223.

Hanlin Zhu, Baihe Huang, Shaolun Zhang, Michael Jordan, Jiantao Jiao, Yuandong Tian, and Stuart J Russell. 2024. Towards a theoretical understanding of the’reversal curse’via training dynamics. *Advances in Neural Information Processing Systems*, 37:90473–90513.