Preserving Knowledge in Large Language Model with Model-Agnostic Self-Decompression

Zilun Zhang[1]†, Yutao Sun[1]†, Tiancheng Zhao[2], Leigang Sha[1], Ruochen Xu[3], Kyusong Lee[2], Jianwei Yin[1],

[1] College of Computer Science and Technology, Zhejiang University,
[2]Binjiang Research Institute of Zhejiang University,
[3]Linker Technology Research Co. Ltd

Correspondence: tianchez@zju-bj.com †: Equal Contribution

Abstract

Humans can retain old knowledge while learning new information, but Large Language Models (LLMs) often suffer from catastrophic forgetting when post-pretrained or supervised fine-tuned (SFT) on domain-specific data. Moreover, for Multimodal Large Language Models (MLLMs) which are composed of the LLM base and visual projector (e.g. LLaVA), a significant decline in performance on language benchmarks was observed compared to their single-modality counterparts. To address these challenges, we introduce a novel model-agnostic self-decompression method, **Tree Generation (TG)**, that decompresses knowledge within LLMs into the training corpus. This paper focuses on TG-SFT, which can synthetically generate SFT data for the instruction tuning steps. By incorporating the dumped corpus during SFT for MLLMs, we significantly reduce the forgetting problem.

1 Introduction

The Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs) have been rapidly developed and iterated in recent years. Many of them show a significant leap in the capability of understanding, generation, and interaction following the natural language (OpenAI et al., 2024; Team et al., 2024; Anthropic, 2024). There are lots of LLMs and MLLMs that have been developed in practice (Kaddour et al., 2023; Yin et al., 2024). However, the model trained for the general purpose may have a decline in performance in specific domains such as math, coding, law, healthcare, finance, etc. (Wu et al., 2024), therefore the need for obtaining sufficient training data to develop domain-specific LLMs or MLLMs is crucial.

Collecting extensive domain-specific data and training LMMs from scratch is challenging. As a result, post-pretraining (Gururangan et al., 2020) or supervised fine-tuning (Brown et al., 2020) (SFT) of general LLMs/MLLMs with domain-specific data become the popular strategy for those seeking domain-specific models (Roziere et al., 2023; Huang et al., 2023; Azerbayev et al., 2024; Yunxiang et al., 2023; Li et al., 2023b; Kuckreja et al., 2023). However, this process can impair the models' performance due to catastrophic forgetting (Aleixo et al., 2023; Luo et al., 2024). We need the expert model to be generalizable as the general model on the specific domain. Although Parameter Efficient Finetuning (PEFT) methods (Mangrulkar et al., 2022) can adapt the models to the new domain by adding only a few parameters and maintaining their original capabilities, they often result in less satisfactory performance and are hard to accumulate from different domains. This calls for an approach that integrates domain-specific expertise into LLMs/MLLMs without compromising their general capabilities.

The problem of catastrophic forgetting in LLM is widely discussed. To verify the problem of catastrophic forgetting in MLLM, we trained (SFT) LLaVA (Liu et al., 2023) for 5 epochs, and evaluated the model every 3000 steps. As shown in Figure 1, we observed that the performance of MLLM benchmarks grew even after the third epoch, but LMM benchmarks started to deteriorate since the third epoch. These different behaviors of the model performance between vision-language benchmarks and pure language benchmarks indicate that MLLM has begun to forget its general language ability rather than simply overfit the data.

There is a point of view that describes the LLMs as lossless compressors (Rae, 2023; Sutskever, 2023; Gu et al., 2024; Yang et al., 2023; Chiang, 2023). (Delétang et al., 2024) from DeepMind explores the connection between predictive models and lossless compression and state that the advancement in self-supervised LLMs can be effectively leveraged for compression tasks. They demonstrate that LLMs not only excel in text compression but

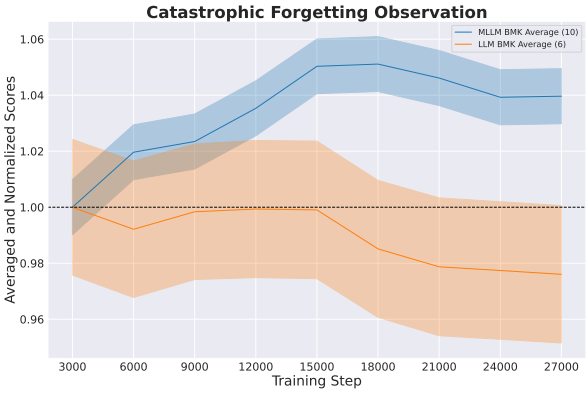

Figure 1: The motivation of Our Work. Shadow represents the error bar. The SFT of MLLM harms the language ability of its LLM backbone (MLLM has begun to forget its general language ability while training is processed). We choose the LLaMA2-7B-chat model as the LLM backbone for the experiments. Details of this experiment can be found in Appendix A.1. The first data point is evaluated from the checkpoint of 3000 steps. We averaged the results of 10 MLLM benchmarks and 6 LLM benchmarks respectively and normalized them with the result of the first checkpoint to show the trend (Increased performance if the score is greater than one. Decreased performance if the score is less than 1).

also show competitive performance across different data modalities, such as images and audio.

Enlightened by these works, we considered the process of synthesizing data with LLMs (a.k.a. generating text data) from LLMs as a decompression process. We aim to preserve knowledge from LLMs by taking a snapshot of the LLMs, i.e. using LLMs as offline data generators and dumping the generated corpus. This self-decompression method should be able to apply to any LMM (model-agnostic). By adding the decompressed data during post-pretraining or SFT, the old knowledge could be reminded and kept. For this purpose, we design a novel approach, named Tree-Generation (**TG**), along with its variants TG-SFT for supervised fine-tuning the MLLMs. With this model-agnostic approach, we observed the catastrophic forgetting problem can be reduced significantly.

From extensive experiments, we show that **TG** algorithm is useful in reducing catastrophic forgetting. Our contribution on **TG** algorithm can be summarized as threefolds.

- **TG** algorithm is a self-contained data generation algorithm based on LLMs, rather than for any specific NLP task (i.e. training BERT includes many specific NLP tasks).
- **TG** algorithm is universally applicable to any LLMs for SFT (TG-SFT). Importantly, no additional manual prompt is required.
- **TG** algorithm is quite foundational, hence it has many applications such as preventing catastrophic forgetting, knowledge distillation, continual learning, etc.

2 Related Work

2.1 Methods for Preventing Catastrophic Forgetting

Preventing Catastrophic Forgetting during training is a classic topic in deep learning. Since 2018, many works discussed how to migrate the Catastrophic Forgetting for LLMs. (Yang et al., 2024) introduced a method that uses self-distillation to bridge the distribution gap between task datasets and LMMs, mitigating catastrophic forgetting while preserving general capabilities. (Luo et al., 2024) conducted an empirical investigation revealing the prevalence of catastrophic forgetting in LLMs as model scale increases during continual instruction tuning, with suggestions that general instruction tuning can help alleviate this issue. (Hsieh et al., 2023) designed a method for training smaller models that outperform LLMs with less training data by extracting rationales from LLMs as additional supervision. Furthermore, (Dou et al., 2024) and (Wang et al., 2023) demonstrate LoRAMoE and O-LoRA, the former introduces low-rank adapters and a router network to alleviate world knowledge, and the latter mitigate catastrophic forgetting by learning new tasks in orthogonal subspaces to minimize interference with past knowledge was proposed by (Wang et al., 2023).

2.2 Sythetic Data for LLM Training

LLMs are trained or tuned on vast amounts of data. Due to the data shortage (especially high-quality data) in many specialized domains, the synthesis of training data for LLMs is crucial.

(Liu et al., 2024a) provide a comprehensive review of the use of synthetic data in training and evaluating AI models, highlighting its potential to overcome data scarcity while emphasizing the importance of ensuring data quality for responsible AI development. (Yehudai et al., 2024) present Genie, a method for the automatic generation of high-quality, content-grounded datasets through a three-stage process: content preparation, generation, and filtering. Researchers from Microsoft introduce

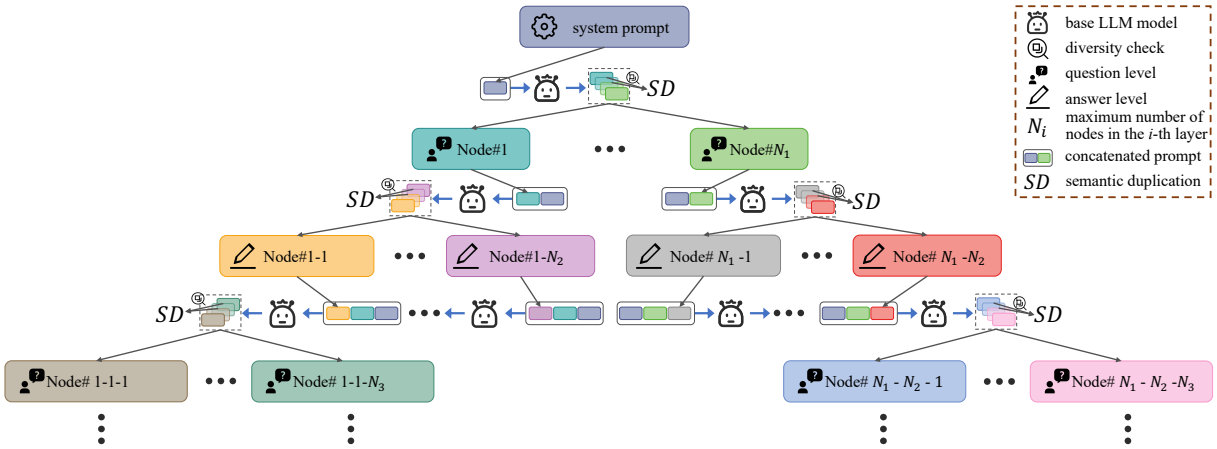

Figure 2: TG-SFT structure overview, illustrates a three-layer complete tree structure. In practice, the depth of different leaf nodes can be adjusted as needed. This figure depicts a typical form of a Balance-Tree, whereas in a Wide-Tree, no further branching occurs beyond the second layer. Starting from the first layer, all odd-numbered layers serve as question layers, and even-numbered layers serve as answer layers.

1.3B models (Gunasekar et al., 2023; Li et al., 2023d) that achieve competitive performance on code generation tasks by leveraging a novel training approach using 'textbook quality' synthetic data, demonstrating the potential for smaller models to rival larger ones through high-quality data curation. (Xu et al., 2023) present Evol-Instruct, an evolutionary algorithm that generates diverse and complex instruction data, enhancing LLM performance on high-complexity tasks through automated data evolution and fine-tuning. (Mitra et al., 2024) introduce Orca-Math, which utilizes a high-quality synthetic dataset of 200K math problems and an iterative learning technique. Li et al. (2023e) find a negative correlation between the models' performance when trained on synthetic data and the degree of subjectivity involved in classification tasks. MAGPIE was presented by (Xu et al., 2024), a method for synthesizing high-quality instruction data from aligned LLMs by prompting them with minimal input. Nvidia released Nemotron-4 340B (Patel, 2024), a model trained on 9 trillion tokens, and over 98% of the data used in the model alignment process was synthetically generated.

2.3 Data Extraction from LLMs and LLM Self-Iteration

Jang explores the capability of GPT-4 to self-reflect and edit its own generations, suggesting the potential for self-correction and improvement in LLMs without external feedback (Jang, 2023). Lee et al. (Lee et al., 2024) introduce a targeted and iterative data augmentation strategy that enhances the performance of LLMs in low-data regimes by using a teacher LLM to generate synthetic data based on incorrect predictions from a student model. Finlayson et al. (Finlayson et al., 2024) demonstrate that non-public information about API-protected LLMs can be gleaned from a small number of API queries, due to the softmax bottleneck in LLM architectures, with implications for model transparency and accountability. Nasr et al. (Nasr et al., 2023) presents a study on the extractable memorization in language models, showing that adversaries can efficiently extract significant amounts of training data from various models, including open-source, semi-open, and closed models, highlighting the need for improved privacy protections.

3 Methodology

TG-SFT is a method designed for high-quality, efficient dialogue generation using a backbone LLM. This approach builds structured dialogue sequences through a tree-based expansion strategy, which is shown in Figure 2, aiming to produce diverse and accurate conversational corpora for model training.

3.1 Initialization and Layered Expansion

The TG-SFT methodology initiates with a general system prompt, denoted as P_0, which is augmented by a special marker indicating the start of instruction. This composite prompt serves as the input to the backbone LLM, prompting it to generate various questions as if it were in the user's role. Formally, we express the first layer input as:

$$P_1 = P_0 + \text{"<user>"}$$

where "<user>" signifies the special marker for instructional onset. Based on P_1, the backbone LLM produces a set of N_1 (which is the predefined number of nodes for the first layer) questions, which after semantic deduplication (*SD*), form the child nodes of the first layer. We use MMR (Maximal Marginal Relevance) (Carbonell and Goldstein, 2017) algorithm to conduct semantic deduplication process. MMR is a technique used to enhance diversity in generated text by balancing relevance and novelty. To apply MMR effectively, we use SentenceBERT (Reimers and Gurevych, 2019) to convert text into vectors. This allows the algorithm to evaluate both the relevance of each sentence to the topic and its distinctiveness from previously selected sentences, thereby reducing redundancy and enhancing the informativeness of the output.

3.2 Recursive Dialogue Generation

For each question generated at node i in the first layer, the formulation of the prompt for the subsequent layers involves dynamically appending the sequence of alternating roles of "<user>" and "<assistant>". Each layer's prompt is constructed by appending the relevant question or response to the initial prompt P_0, supplemented by role markers to guide the model's generation contextually.

For odd-indexed layers $(2i+1, i \geq 0)$, representing user-initiated questions, the prompt is constructed as follows:

$$P_{2i+1} = P_0 + \text{"<user>"} + Q_1 + \text{"<assistant>"} + R_1 + \ldots + R_i + \text{"<user>"}$$

Here, $Q_1, R_1, \ldots, R_i$ represent the sequence of questions and responses up to the i-th response, with the subsequent user query being formulated. This setup directs the base LLM to continue the dialogue from the perspective of the user, generating a new question Q_{i+1}.

For even-indexed layers $(2i)$, which capture responses from the assistant, the prompt configuration is:

$$P_{2i} = P_0 + \text{"<user>"} + Q_1 + \text{"<assistant>"} + R_1 + \ldots + Q_i + \text{"<assistant>"}$$

In this case, $Q_1, R_1, \ldots, Q_i$ denote the alternating questions and responses leading to the i-th question, setting the stage for the assistant's response. Examples of $P_2 i$ and P_{2i+1} are illustrated in the Figure 3

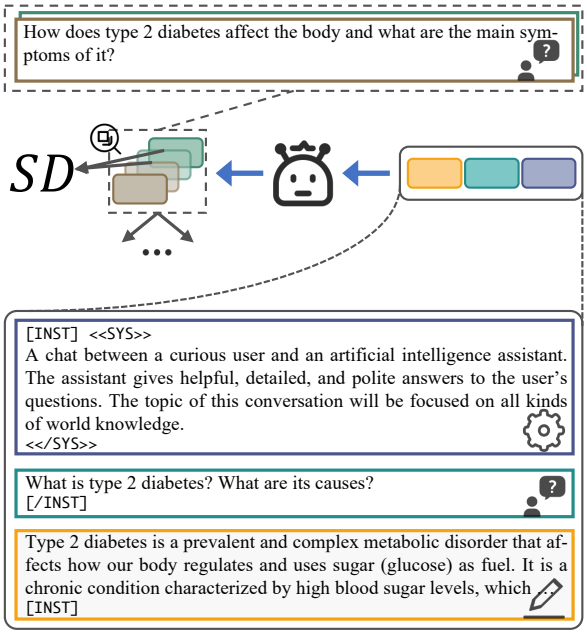

Figure 3: Example of Concatenated Prompts: This figure uses the Llama2-chat model as the backbone LLM. In Llama2, the system prompt is enclosed with "«SYS»", "[INST]" indicates the start of an instruction, signifying the beginning of generation in the user role. "[/INST]" marks the end of the instruction. LLMs are trained to start responding from this point in the pretraining phase.

3.3 Corpus Construction

The final structure comprises nodes at depth $2k$, with a total of $\prod_{i=1}^{2k} N_i$ leaf nodes. From the root P_0, any path leading to a leaf node at depth $2k$ represents a complete dialogue sequence. In the TG-SFT model, we use C to represents the set of all possible dialogue sequences generated from the root to the leaves of the tree. Formally, C is defined as:

$$C = \bigcup_{i=1}^{N_{2k}} (P_0 + Q_1 + R_1 + \ldots + Q_k + R_k)_i$$

where Q_j and R_j represent the questions and responses at each dialogue turn j.

3.4 Structural Features of TG-SFT

The TG-SFT algorithm is strategically designed to optimize dialogue generation by varying the breadth and depth of the generated dialogue tree in response to the complexity of the conversation.

Initial Question Generation Layer: The first layer N_1 is typically larger, allowing the model significant latitude to explore various topics arising from the general system prompt P_0. This expansive approach is crucial as it branches out into N_1

distinct questions, laying a broad thematic groundwork.

Specificity and Depth in Subsequent Layers: The sizes of subsequent layers are tailored based on the specific requirements of the dialogue depth. For any q (where $q \geq 0$), N_{2q+1} represents the number of new questions posed in response to the answers at layer $2q$. As discussions become more specific, the potential to branch out further diminishes, generally resulting in $N_{2p+1} < N_{2q+1}$ for $p > q$. Similarly, N_{2q+2} represents the number of responses to the questions at layer $2q+1$, with N_{2p+2} typically being less than N_{2q+2} as the conversation narrows down.

Token Allocation Strategy: The token allocation per layer, L_i, is carefully designed. For layers $2q+1$, a constant m_0 is set, representing the maximum token count for any question, ensuring questions remain concise while enhancing the efficiency of the generation process:

$$L_{2q+1} = m_0, \forall q$$

For layers that involve responses $(2q+2)$, L_{2q+2} is typically smaller than L_{2p+2} for deeper layers, reflecting the increased specificity and detail of responses as the dialogue progresses:

$$L_{2q+2} < L_{2p+2}, \text{ for } p > q$$

This setting aims to provide more detailed explanations as the discussion delves deeper into some specific topics.

Flexibility and Customization: Both N_i and L_i are tunable parameters, offering the flexibility to enrich the dialogue corpus significantly. By setting a higher N_1, the algorithm ensures a wide range of initial topics, leading to a rich and diverse corpus. This diversity, coupled with detailed responses in deeper layers, ensures comprehensive coverage of specific topics, providing detailed and contextually relevant answers. There is a special type of TG-SFT structure called **Wide-Tree** where N_1 is set exceptionally high while N_i (for $i > 1$) are all equal to 1. Conversely, trees where N_i values (for $i > 1$) are not all set to 1 are referred to as **Balance-Tree**, which allow for a less extensive initial exploration yet more detailed follow-up inquiries across the subsequent layers. We will further discuss TG-SFT(Wide-Tree) and TG-SFT(Balance-Tree) in section 4.2.

4 Experiments

In this section, we introduce our experiment settings and discuss the findings. We begin this section by detailing the experiment configuration (section 4.1). Then we validate the effectiveness of the TG-SFT approach (section 4.2). Next, we demonstrate the potential of TG-PT (section 4.3), the application of **TG** in post-pretraining. Subsequently, we analyze the corpus generated by TG-SFT (section 4.4). Finally, in section 4.5 and 4.6, we explore the impact of different tree configurations and number of conversation turns.

4.1 Experimental Settings

Models. Unless specified, we use the LLaMA2-chat (7B) model (Touvron et al., 2023) for the experiments in this section. For the MLLM, we train a projector to align the CLIP vision encoder with LLaMA2-chat. Then, we supervisedly finetune (SFT) both the LLaMA2-chat model and the projector to obtain the LLaVA model (Liu et al., 2023). We choose the LLaMA2-chat model as the LLM backbone to avoid the influence of additional tuning data, such as ShareGPT used for Vicuna. All LLaVA models discussed in this paper use LLaMA2-chat as the LLM backbone. The codebase for alignment and SFT is obtained from the official LLaVA GitHub repository [1].

Data. In line with LLaVA, we use the 558K subset of the LAION-CC-SBU dataset [2] to align the vision encoder and the LLM. For supervised finetuning, we employ the Mix 665K dataset [3]. Additionally, we generate a 100K language-only corpus using the **TG** approach.

Evaluation. For MLLM benchmarks, we follow LLaVA and select the following datasets: GQA (Hudson and Manning, 2019), MM-Bench (Liu et al., 2024b), POPE (Li et al., 2023c), ScienceQA-IMG (Lu et al., 2022), SEED-Bench (Li et al., 2023a), TextVQA (Singh et al., 2019), VisWiz (Gurari et al., 2018), and VQA^{v2} (Goyal et al., 2017). For LLM benchmarks, we adhere to LLaMAPro (Wu et al., 2024) and choose the AI2 Reasoning Challenge (ARC, 25-shot) (Clark et al., 2018), HellaSwag (10-shot) (Zellers et al., 2019), MMLU

[1] https://github.com/haotian-liu/LLaVA

[2] https://huggingface.co/datasets/liuhaotian/LLaVA-Pretrain

[3] https://huggingface.co/datasets/liuhaotian/LLaVA-Instruct-150K/blob/main/llava_v1_5_mix665k.json

(5-shot) (Hendrycks et al., 2021), TruthfulQA (0-shot) (Lin et al., 2022), and Winogrande (5-shot) (Sakaguchi et al., 2019). We use lmms-eval (Li* et al., 2024) as the MLLM evaluation pipeline and lm-evaluation-harness [4] as the LLM evaluation pipeline, as proposed by Gao et al. (Gao et al., 2023). Additionally, we report the average scores for both MLLM and LLM benchmarks.

Training Details. Our experiments are conducted on 8 A100-80GB GPUs with NVLink. As observed in Figure 1, SFT causes the model to start forgetting LLM knowledge by the third epoch. Therefore, we set the number of training epochs for SFT to 3, which requires 36 hours of training. The batch size is set to 32 per device, and other training parameters follow LLaVA's defaults. The corpus generation process takes between 20 to 40 hours on 8 A100-40GB GPUs without NVLink, depending on the configuration of the TG-SFT approach.

4.2 Results of TG-SFT

We evaluate the results of LLaVA trained with the corpus generated by TG-SFT against other approaches, as shown in Table 1. We categorize the different approaches into two groups: Model-wise and Data-wise. All corpora in this subsection are generated using the LLaMA2-chat (7B) model, with a total of 100K conversations generated. The different approaches are explained below.

LLaMA2-chat. This is the baseline for LLM benchmarks without any additional modifications.

LLaVA (Full-Param). The baseline approach for MLLM benchmarks, where the LLM backbone is replaced by LLaMA2-chat. This is done because we use LLaMA2-chat to generate synthetic data in other experiments. The corresponding projector is aligned using the LAION-CC-SBU dataset as described in the original LLaVA paper.

LLaVA (LoRA). The LLaVA is fine-tuned (SFT) with the Mix 665K data and LoRA (Hu et al., 2021). During the evaluation of LLM benchmarks, the trained LoRA adapter is deactivated.

Human (ShareGPT). The LLaVA is fine-tuned with the Mix 665K data and 100K data randomly selected from the ShareGPT dataset used in LLaMAPro (Wu et al., 2024). The 100K additional training data in this approach is of high quality since it was generated and rated by humans. This approach serves as an upper bound for Data-wise approaches.

TG-SFT (Wide-Tree). The LLaVA is supervisedly fine-tuned with the Mix 665K data and 100K synthetic data randomly generated by LLaMA2-chat. This approach serves as the baseline for Data-wise approaches. This structure maximizes the breadth of initial topic exploration at the first layer, prioritizing a vast range of questions without focusing on detailed follow-up inquiries in deeper layers. Such a configuration allows for the broadest possible survey of topics, albeit at the expense of depth in specific issue elaboration.

TG-SFT (Balance-Tree). The LLaVA is fine-tuned with the Mix 665K data and 100K synthetic data generated by the TG-SFT (Balance-Tree) approach. The **knowledge-guided technique** was applied, ensuring that the synthetic data follows the knowledge-guided distribution.

Result Analysis. Table 1 highlights the forgetting phenomenon in MLLM during SFT. The average score for the LLM benchmark of the LLaMA2-chat model is 53.41, but the average score for the LLaVA baseline decreases to 50.60. Meanwhile, LLaVA trained with the TG-SFT (Balance-Tree) approach and Knowledge-Guided technique restores the average LLM benchmark score to 53.47 while maintaining comparable performance on the MLLM benchmarks. The average performance in the LLM benchmark for TG-SFT(Balance-Tree) nearly matches that of LLaVA SFT with human-produced ShareGPT data, demonstrating that synthetic data generated using TG-SFT is highly effective, even compared to real data.

Additionally, while the TG-SFT (Wide-Tree) approach performs better than the LLaVA (Full-Param) baseline, it does not achieve the same level of performance as LLaVA with the Knowledge-Guided TG-SFT (Balance-Tree) approach, indicating the importance of knowledge guidance.

In practice, we found that TG-SFT (Wide-Tree) is faster than TG-SFT (Balance-Tree) in terms of the data generation speed since the former can generate data in a fully parallel manner in theory, as long as we have enough VRAM. The latter one can only generate data in a partially parallel way since prior knowledge from previous layers is needed.

[4] https://github.com/hills-code/lm-evaluation-harness

Method Name	MLLM Benchmark								Average
	GQA	MMB	POPE	SQA^{I}	SEED	VQA^{T}	VisWiz	VQA^{v2}	
Model-wise Result (Baseline & Model Augmented)									
LLaMA2-chat (LLM)	-	-	-	-	-	-	-	-	-
LLaVA (Full-Param)	62.55	63.66	85.71	69.31	66.08	45.28	54.79	77.19	65.57
LLaVA (LoRA)	63.16	64.26	85.32	66.78	66.41	46.26	52.36	77.71	65.28
Data-wise Result (Data Augmented)									
Human (ShareGPT)	62.64	63.57	84.63	67.28	65.33	44.89	53.61	77.02	64.87
TG-SFT (Wide-Tree)	62.26	64.60	84.69	68.47	65.53	45.35	52.41	77.04	65.04
TG-SFT (Balance-Tree)	62.79	64.35	85.13	68.02	65.11	45.44	52.45	77.02	65.04

Method Name	LLM Benchmark						Average
	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K	
Model-wise Result (Baseline & Model Augmented)							
LLaMA2-chat (LLM)	53.50	78.58	47.24	45.32	72.53	23.28	53.41
LLaVA (Full-Param)	49.06	72.71	47.98	49.26	68.75	15.85	50.60
LLaVA (LoRA)	51.02	74.08	48.75	47.49	70.24	16.98	51.43
Data-wise Result (Data Augmented)							
Human (ShareGPT)	54.69	74.05	50.60	49.86	71.90	21.38	53.75
TG-SFT (Wide-Tree)	49.06	76.48	50.19	50.30	70.48	21.30	52.97
TG-SFT (Balance-Tree)	53.41	75.83	50.09	50.69	70.01	20.77	**53.47**

Table 1: **Comparison with different approaches on 8 MMLM benchmarks and 6 LMM benchmarks.** Benchmark names are abbreviated. MMB: MMBench; SQA^{I}: ScienceQA-IMG; SEED: SEED-Bench; VQA^{T}: TextVQA; VQA^{v2}: VQA-v2 ; ARC: AI2 Reasoning Challenge. Compared with the LLaVA baseline, LLaVA trained with the Wide-Tree TG-SFT approach restores the average score of the LLM benchmark from 50.60 to 52.97. TG-SFT (Balance-Tree) further boosts this performance to 53.47, which is slightly higher than the LLaMA2-chat backbone's performance. TG-SFT approaches maintain a comparable performance with LLaVA (Full-Param tuned) baseline on the MLLM benchmarks as well.

4.3 TG-PT

Dialogue data generated by TG-SFT is suitable for SFT training. To validate the efficacy of the TG method, we have developed a new variant of TG-SFT specifically designed for Post-Pretraining. This variant involves substituting the backbone LLM from a chat model to a base model, thereby eliminating role-switching in generation. Instead, this variant initiates with a simple prompt, such as "*Here are some useful world knowledge:*" and continues to expand the dialogue in a tree-structured manner. The resulting data is suitable for post-training applications. Consequently, we refer to this variant as TG-PT (Tree-Generation for Post-PreTraining).

Table 2 demonstrates the effectiveness of the TG-PT approach. We conducted post-pretraining on the LLaMA-2 base model (7B) with 100K data. The LLM benchmark performance using data generated by TG-PT is compared to that using randomly generated data. The results indicate that post-pretraining the LLM with a randomly generated corpus leads to significant performance degradation. However, post-pretraining with data generated using the TG-PT approach not only mitigates this issue but also provides a slight performance gain. This result illustrates that the TG method is not only suitable for SFT but also for Post-Pretraining, demonstrating its versatility. This indirectly confirms the general applicability of the TG algorithm in decompressing LLMs, showcasing its efficacy across different training regimes.

4.4 Decompressed Data Analysis

As shown in Figure 4, we randomly sampled 1K conversations from three different sources: data generated with TG-SFT (Wide-Tree), TG-SFT (Balance-Tree), and data collected from ShareGPT. We then visualized these samples using Sentence-Bert embeddings ("all-MiniLM-L6-v2" model) (Reimers and Gurevych, 2020) and T-SNE (van der Maaten and Hinton, 2008). The clusters for these three types of corpus are quite distinctive.

Method Name	LLM Benchmark						Average
	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K	
LLaMA-2-Base	54.01	78.63	45.6	38.92	73.95	13.27	50.73
Random	51.62	62.07	44.79	41.03	71.43	0.45	45.23
TG-PT	55.55	78.69	45.04	38.44	73.56	11.9	50.53

Table 2: Result of TG-PT Algorithm in LLM Benchmarks.

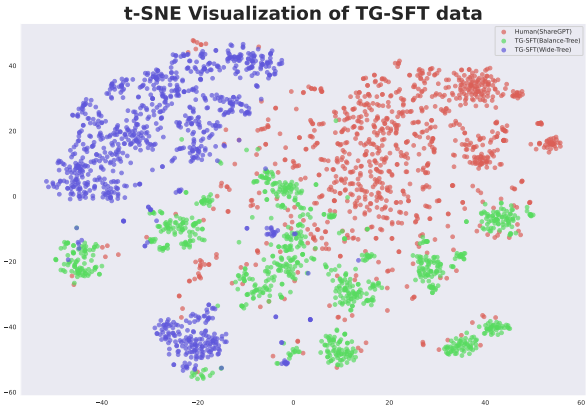

Figure 4: T-SNE data visualization for corpus generated using TG-SFT and collected from ShareGPT

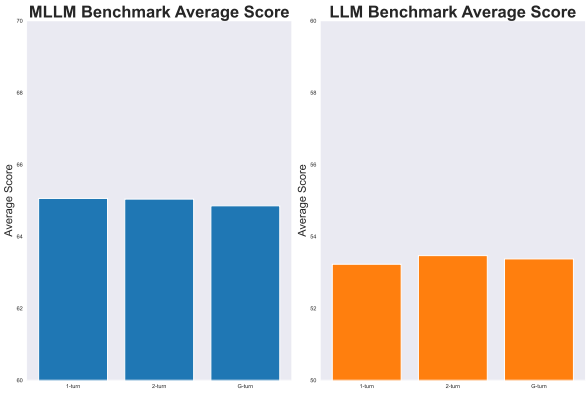

Figure 5: Number of turns in TG-SFT decompressed Data

4.5 Tree Configuration

Exp ID	L1 (Q)	L2 (A)	L3 (Q)	L4 (A)	MLLM	LLM
1	32	16	8	8	65.04	53.47
2	16	16	8	8	65.08	53.46
3	32	8	8	8	65.21	**53.80**
4	32	16	8	4	**65.23**	52.88

Table 3: Results of Different Tree Configurations. "L" represents "Layer", "Q" represents "Question", and "A" represents "Answer".

Table 3 demonstrates the performance of the TG-SFT approach with various tree configurations. Specifically, we halved the branching factor at different tree levels and generated the corpus for training. The average scores of MLLM and LLM benchmarks across different settings do not show significant differences, indicating the parameter-insensitivity property of our TG-SFT approach.

4.6 Conversation Turns

We investigate the effects of different numbers of conversation turns in TG-SFT generated data, as shown in Figure 5. We experimented with corpora consisting of all 1-turn, all 2-turn, and a mixture of 1, 2, 3, and 4 turns (referred to as "G-turn" since the distribution of # turns follows a Gaussian distribution). Results indicate that training with the 2-turn corpus achieves the best performance in LLM benchmarks compared to the other two configurations. This could be attributed to the G-turn corpus being too diverse in context length and the 1-turn corpus being too short, which harms the the model during SFT.

5 Conclusion & Future Work

To address the problem of catastrophic forgetting in LLMs and MLLMs, we designed a novel model-agnostic self-decompression method, **TG (Tree Generation)**, which decompresses knowledge within LLMs into the training corpus. We introduced its variants: TG-SFT for supervised fine-tuning. By utilizing this decompressed corpus, we mitigate the forgetting problem. Experiment results show that using the synthetic data generated from TG, an LLM can preserve its original knowledge and perform on par to using human generated high quality data. The tree structure in TG enable flexible control of speed and diversity, which enables better control of the generation process.

At last, TG has the potential to enable use cases such as pre-training and knowledge distillation. We leave a more general version of TG-PT for post-pretraining in future work. We plan to verify if using a decompressed corpus from a strong model can enhance the performance of a weaker model.

6 Limitation

There are several limitations of our works, mainly focusing on the safety issue of synthetic data, how to synthesize data for post-pretraining, and how to evaluate the quality and diversity of synthetic data.

6.1 Data Leakage Risk, NSFW Content Exposure, and Inaccurate Information

The TG-SFT method, while effective in fine-tuning MLLMs, may inadvertently facilitate the extraction of training data. Given the auto-regressive nature of LLMs, there is a risk that the model could generate outputs that closely resemble its training data. This poses a significant concern if the training data contains sensitive information.

In the event that the training data includes Not Safe For Work (NSFW) content, the TG-SFT method might inadvertently generate responses that expose or allude to such content. This not only undermines the ethical standards of AI applications but also raises questions about the responsible use of LLMs.

The synthetic data may include many inaccurate facts. If researchers use such data to train the model, it is possible that model outputs fake information. A fact verifier is needed for data synthesizing.

6.2 Synthetic data for Post-Pretraining

In this paper, we mainly focus on synthesizing data for SFT. We introduced a variant of **TG**, denoted as TG-PT, for post-pretraining in section 4.3. However, a more general version of TG-PT for post-pretraining is needed in the future since post-pretraining requires much more data compared with SFT.

6.3 Synthetic data for Specific Domain

In our experiments, we attempted to synthesize data within the domain of mathematics. However, we found quality of the synthesized data is not satisfactory. The LLMs tend to generate math concept, and simple/wrong calculations and derivations. SFT with such data resulted in a noticeable degradation in performance on mathematical benchmarks. The challenge of synthesizing high-quality data for specific domains, is an issue that merits further investigation in future research.

6.4 Benchmarks for Evaluating the Quality and Diversity of Synthetic Data.

The evaluation benchmark for measuring the quality and diversity of synthetic text data is rare. (Wei et al., 2024) introduce LongFact, a prompt set of 2,280 fact-seeking prompts requiring long-form responses, but LongFact is dependent on LLMs for their operations. Consequently, the capabilities of the utilized LLM have a direct impact on the quality of the LongFact prompts. More universal and model-agnostic benchmarks are needed to evaluate the quality of synthetic data. Moreover, (Shaib et al., 2024) propose measurement of text diversity including compression ratios, self-repetition of long n-grams, Self-BLEU, BERTScore, etc., but the computation time for calculating such scores for large amount of training corpus is unacceptable.

References

Everton L. Aleixo, Juan G. Colonna, Marco Cristo, and Everlandio Fernandes. 2023. Catastrophic forgetting in deep learning: A comprehensive taxonomy. *Preprint*, arXiv:2312.10549.

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2024. Llemma: An open language model for mathematics. *Preprint*, arXiv:2310.10631.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Jaime G. Carbonell and Jade Goldstein. 2017. The use of mmr, diversity-based reranking for reordering documents and producing summaries. *SIGIR Forum*, 51(2):209–210.

Ted Chiang. 2023. Chatgpt is a blurry jpeg of the web.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. 2024. Language modeling is compression. *Preprint*, arXiv:2309.10668.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Loramoe: Alleviate world knowledge forgetting in large language models via moe-style plugin. *Preprint*, arXiv:2312.09979.

Matthew Finlayson, Xiang Ren, and Swabha Swayamdipta. 2024. Logits of api-protected llms leak proprietary information. *Preprint*, arXiv:2403.09539.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *Preprint*, arXiv:1612.00837.

Yuxian Gu, Li Dong, Yaru Hao, Qingxiu Dong, Minlie Huang, and Furu Wei. 2024. Towards optimal learning of language models. *Preprint*, arXiv:2402.17759.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. *Preprint*, arXiv:2306.11644.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. *Preprint*, arXiv:1802.08218.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *Preprint*, arXiv:2004.10964.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *Preprint*, arXiv:2305.02301.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. Lawyer llama technical report. *Preprint*, arXiv:2305.15062.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Preprint*, arXiv:1902.09506.

Eric Jang. 2023. Can llms critique and iterate on their own outputs? *evjang.com*.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *Preprint*, arXiv:2307.10169.

Kartik Kuckreja, Muhammad Sohail Danish, Muzammal Naseer, Abhijit Das, Salman Khan, and Fahad Shahbaz Khan. 2023. Geochat: Grounded large vision-language model for remote sensing. *Preprint*, arXiv:2311.15826.

Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipali, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. 2024. Llm2llm: Boosting llms with novel iterative data enhancement. *Preprint*, arXiv:2403.15042.

Bo Li*, Kaichen Zhang* Peiyuan Zhang*, Fanyi Pu*, Xinrun Du, Yuhao Dong, Haotian Liu, Yuanhan Zhang, Ge Zhang, Chunyuan Li, and Ziwei Liu. 2024. Lmms-eval: Accelerating the development of large multimoal models.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023a. Seed-bench: Benchmarking multimodal llms with generative comprehension. *Preprint*, arXiv:2307.16125.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2023b. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023c. Evaluating object hallucination in large vision-language models. *Preprint*, arXiv:2305.10355.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023d. Textbooks are all you need ii: phi-1.5 technical report. *Preprint*, arXiv:2309.05463.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023e. Synthetic data generation with large language models for text classification: Potential and limitations. *Preprint*, arXiv:2310.07849.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. 2024a. Best practices and lessons learned on synthetic data for language models. *Preprint*, arXiv:2404.07503.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2024b. Mmbench: Is your multi-modal model an all-around player? *Preprint*, arXiv:2307.06281.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Preprint*, arXiv:2209.09513.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2024. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *Preprint*, arXiv:2308.08747.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-math: Unlocking the potential of slms in grade school math. *Preprint*, arXiv:2402.14830.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *Preprint*, arXiv:2311.17035.

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. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Ankit Patel. 2024. NVIDIA Releases Open Synthetic Data Generation Pipeline for Training Large Language Models — blogs.nvidia.com. https://blogs.nvidia.com/blog/nemotron-4-synthetic-data-generation-llm-training/. [Accessed 15-06-2024].

Jack Rae. 2023. Compression for agi.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP/IJCNLP (1)*, pages 3980–3990. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *Preprint*, arXiv:1907.10641.

Chantal Shaib, Joe Barrow, Jiuding Sun, Alexa F. Siu, Byron C. Wallace, and Ani Nenkova. 2024. Standardizing the measurement of text diversity: A tool and a comparative analysis of scores. *Preprint*, arXiv:2403.00553.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. *Preprint*, arXiv:1904.08920.

Ilya Sutskever. 2023. Ilya sutskever (openai chief scientist) - building agi, alignment, spies, microsoft, & enlightenment.

Gemini Team, Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry, Lepikhin, Timothy Lillicrap, Jean baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, Luke Vilnis, Oscar Chang, Nobuyuki Morioka, George Tucker, Ce Zheng, Oliver Woodman, Nithya Attaluri, Tomas Kocisky, Evgenii Eltyshev, Xi Chen, Timothy Chung, Vittorio Selo, Siddhartha Brahma, Petko Georgiev, Ambrose Slone, Zhenkai Zhu, James Lottes, Siyuan Qiao, Ben Caine, Sebastian Riedel,

Alex Tomala, Martin Chadwick, Juliette Love, Peter Choy, Sid Mittal, Neil Houlsby, Yunhao Tang, Matthew Lamm, Libin Bai, Qiao Zhang, Luheng He, Yong Cheng, Peter Humphreys, Yujia Li, Sergey Brin, Albin Cassirer, Yingjie Miao, Lukas Zilka, Taylor Tobin, Kelvin Xu, Lev Proleev, Daniel Sohn, Alberto Magni, Lisa Anne Hendricks, Isabel Gao, Santiago Ontanon, Oskar Bunyan, Nathan Byrd, Abhanshu Sharma, Biao Zhang, Mario Pinto, Rishika Sinha, Harsh Mehta, Dawei Jia, Sergi Caelles, Albert Webson, Alex Morris, Becca Roelofs, Yifan Ding, Robin Strudel, Xuehan Xiong, Marvin Ritter, Mostafa Dehghani, Rahma Chaabouni, Abhijit Karmarkar, Guangda Lai, Fabian Mentzer, Bibo Xu, YaGuang Li, Yujing Zhang, Tom Le Paine, Alex Goldin, Behnam Neyshabur, Kate Baumli, Anselm Levskaya, Michael Laskin, Wenhao Jia, Jack W. Rae, Kefan Xiao, Antoine He, Skye Giordano, Lakshman Yagati, Jean-Baptiste Lespiau, Paul Natsev, Sanjay Ganapathy, Fangyu Liu, Danilo Martins, Nanxin Chen, Yunhan Xu, Megan Barnes, Rhys May, Arpi Vezer, Junhyuk Oh, Ken Franko, Sophie Bridgers, Ruizhe Zhao, Boxi Wu, Basil Mustafa, Sean Sechrist, Emilio Parisotto, Thanumalayan Sankaranarayana Pillai, Chris Larkin, Chenjie Gu, Christina Sorokin, Maxim Krikun, Alexey Guseynov, Jessica Landon, Romina Datta, Alexander Pritzel, Phoebe Thacker, Fan Yang, Kevin Hui, Anja Hauth, Chih-Kuan Yeh, David Barker, Justin Mao-Jones, Sophia Austin, Hannah Sheahan, Parker Schuh, James Svensson, Rohan Jain, Vinay Ramasesh, Anton Briukhov, DaWoon Chung, Tamara von Glehn, Christina Butterfield, Priya Jhakra, Matthew Wiethoff, Justin Frye, Jordan Grimstad, Beer Changpinyo, Charline Le Lan, Anna Bortsova, Yonghui Wu, Paul Voigtlaender, Tara Sainath, Shane Gu, Charlotte Smith, Will Hawkins, Kris Cao, James Besley, Srivatsan Srinivasan, Mark Omernick, Colin Gaffney, Gabriela Surita, Ryan Burnell, Bogdan Damoc, Junwhan Ahn, Andrew Brock, Mantas Pajarskas, Anastasia Petrushkina, Seb Noury, Lorenzo Blanco, Kevin Swersky, Arun Ahuja, Thi Avrahami, Vedant Misra, Raoul de Liedekerke, Mariko Iinuma, Alex Polozov, Sarah York, George van den Driessche, Paul Michel, Justin Chiu, Rory Blevins, Zach Gleicher, Adrià Recasens, Alban Rrustemi, Elena Gribovskaya, Aurko Roy, Wiktor Gworek, Sébastien M. R. Arnold, Lisa Lee, James Lee-Thorp, Marcello Maggioni, Enrique Piqueras, Kartikeya Badola, Sharad Vikram, Lucas Gonzalez, Anirudh Baddepudi, Evan Senter, Jacob Devlin, James Qin, Michael Azzam, Maja Trebacz, Martin Polacek, Kashyap Krishnakumar, Shuo yiin Chang, Matthew Tung, Ivo Penchev, Rishabh Joshi, Kate Olszewska, Carrie Muir, Mateo Wirth, Ale Jakse Hartman, Josh Newlan, Sheleem Kashem, Vijay Bolina, Elahe Dabir, Joost van Amersfoort, Zafarali Ahmed, James Cobon-Kerr, Aishwarya Kamath, Arnar Mar Hrafnkelsson, Le Hou, Ian Mackinnon, Alexandre Frechette, Eric Noland, Xiance Si, Emanuel Taropa, Dong Li, Phil Crone, Anmol Gulati, Sébastien Cevey, Jonas Adler, Ada Ma, David Silver, Simon Tokumine, Richard Powell, Stephan Lee, Kiran Vodrahalli, Samer Hassan, Diana Mincu, Antoine Yang, Nir Levine, Jenny Brennan, Mingqiu Wang, Sarah Hodkinson, Jeffrey Zhao, Josh Lipschultz, Aedan Pope, Michael B. Chang, Cheng Li, Laurent El Shafey, Michela Paganini, Sholto Douglas, Bernd Bohnet, Fabio Pardo, Seth Odoom, Mihaela Rosca, Cicero Nogueira dos Santos, Kedar Soparkar, Arthur Guez, Tom Hudson, Steven Hansen, Chulayuth Asawaroengchai, Ravi Addanki, Tianhe Yu, Wojciech Stokowiec, Mina Khan, Justin Gilmer, Jaehoon Lee, Carrie Grimes Bostock, Keran Rong, Jonathan Caton, Pedram Pejman, Filip Pavetic, Geoff Brown, Vivek Sharma, Mario Lučić, Rajkumar Samuel, Josip Djolonga, Amol Mandhane, Lars Lowe Sjösund, Elena Buchatskaya, Elspeth White, Natalie Clay, Jiepu Jiang, Hyeontaek Lim, Ross Hemsley, Zeyncep Cankara, Jane Labanowski, Nicola De Cao, David Steiner, Sayed Hadi Hashemi, Jacob Austin, Anita Gergely, Tim Blyth, Joe Stanton, Kaushik Shivakumar, Aditya Siddhant, Anders Andreassen, Carlos Araya, Nikhil Sethi, Rakesh Shivanna, Steven Hand, Ankur Bapna, Ali Khodaei, Antoine Miech, Garrett Tanzer, Andy Swing, Shantanu Thakoor, Lora Aroyo, Zhufeng Pan, Zachary Nado, Jakub Sygnowski, Stephanie Winkler, Dian Yu, Mohammad Saleh, Loren Maggiore, Yamini Bansal, Xavier Garcia, Mehran Kazemi, Piyush Patil, Ishita Dasgupta, Iain Barr, Minh Giang, Thais Kagohara, Ivo Danihelka, Amit Marathe, Vladimir Feinberg, Mohamed Elhawaty, Nimesh Ghelani, Dan Horgan, Helen Miller, Lexi Walker, Richard Tanburn, Mukarram Tariq, Disha Shrivastava, Fei Xia, Qingze Wang, Chung-Cheng Chiu, Zoe Ashwood, Khuslen Baatarsukh, Sina Samangooei, Raphaël Lopez Kaufman, Fred Alcober, Axel Stjerngren, Paul Komarek, Katerina Tsihlas, Anudhyan Boral, Ramona Comanescu, Jeremy Chen, Ruibo Liu, Chris Welty, Dawn Bloxwich, Charlie Chen, Yanhua Sun, Fangxiaoyu Feng, Matthew Mauger, Xerxes Dotiwalla, Vincent Hellendoorn, Michael Sharman, Ivy Zheng, Krishna Haridasan, Gabe Barth-Maron, Craig Swanson, Dominika Rogozińska, Alek Andreev, Paul Kishan Rubenstein, Ruoxin Sang, Dan Hurt, Gamaleldin Elsayed, Renshen Wang, Dave Lacey, Anastasija Ilić, Yao Zhao, Adam Iwanicki, Alejandro Lince, Alexander Chen, Christina Lyu, Carl Lebsack, Jordan Griffith, Meenu Gaba, Paramjit Sandhu, Phil Chen, Anna Koop, Ravi Rajwar, Soheil Hassas Yeganeh, Solomon Chang, Rui Zhu, Soroush Radpour, Elnaz Davoodi, Ving Ian Lei, Yang Xu, Daniel Toyama, Constant Segal, Martin Wicke, Hanzhao Lin, Anna Bulanova, Adrià Puigdomènech Badia, Nemanja Rakićević, Pablo Sprechmann, Angelos Filos, Shaobo Hou, Víctor Campos, Nora Kassner, Devendra Sachan, Meire Fortunato, Chimezie Iwuanyanwu, Vitaly Nikolaev, Balaji Lakshminarayanan, Sadegh Jazayeri, Mani Varadarajan, Chetan Tekur, Doug Fritz, Misha Khalman, David Reitter, Kingshuk Dasgupta, Shourya Sarcar, Tina Ornduff, Javier Snaider, Fantine Huot, Johnson Jia, Rupert Kemp, Nejc Trdin, Anitha Vijayakumar, Lucy Kim, Christof Angermueller, Li Lao, Tianqi Liu, Haibin Zhang, David Engel, Somer Greene, Anaïs White, Jessica Austin, Lilly Taylor, Shereen Ashraf, Dangyi Liu, Maria Georgaki, Irene Cai, Yana Kulizhskaya, Sonam Goenka, Brennan Saeta, Ying Xu, Christian Frank, Dario de Cesare, Brona Robenek,

Harry Richardson, Mahmoud Alnahlawi, Christopher Yew, Priya Ponnapalli, Marco Tagliasacchi, Alex Korchemniy, Yelin Kim, Dinghua Li, Bill Rosgen, Kyle Levin, Jeremy Wiesner, Praseem Banzal, Praveen Srinivasan, Hongkun Yu, Çağlar Ünlü, David Reid, Zora Tung, Daniel Finchelstein, Ravin Kumar, Andre Elisseeff, Jin Huang, Ming Zhang, Ricardo Aguilar, Mai Giménez, Jiawei Xia, Olivier Dousse, Willi Gierke, Damion Yates, Komal Jalan, Lu Li, Eri Latorre-Chimoto, Duc Dung Nguyen, Ken Durden, Praveen Kallakuri, Yaxin Liu, Matthew Johnson, Tomy Tsai, Alice Talbert, Jasmine Liu, Alexander Neitz, Chen Elkind, Marco Selvi, Mimi Jasarevic, Livio Baldini Soares, Albert Cui, Pidong Wang, Alek Wenjiao Wang, Xinyu Ye, Krystal Kallarackal, Lucia Loher, Hoi Lam, Josef Broder, Dan Holtmann-Rice, Nina Martin, Bramandia Ramadhana, Mrinal Shukla, Sujoy Basu, Abhi Mohan, Nick Fernando, Noah Fiedel, Kim Paterson, Hui Li, Ankush Garg, Jane Park, DongHyun Choi, Diane Wu, Sankalp Singh, Zhishuai Zhang, Amir Globerson, Lily Yu, John Carpenter, Félix de Chaumont Quitry, Carey Radebaugh, Chu-Cheng Lin, Alex Tudor, Prakash Shroff, Drew Garmon, Dayou Du, Neera Vats, Han Lu, Shariq Iqbal, Alex Yakubovich, Nilesh Tripuraneni, James Manyika, Haroon Qureshi, Nan Hua, Christel Ngani, Maria Abi Raad, Hannah Forbes, Jeff Stanway, Mukund Sundararajan, Victor Ungureanu, Colton Bishop, Yunjie Li, Balaji Venkatraman, Bo Li, Chloe Thornton, Salvatore Scellato, Nishesh Gupta, Yicheng Wang, Ian Tenney, Xihui Wu, Ashish Shenoy, Gabriel Carvajal, Diana Gage Wright, Ben Bariach, Zhuyun Xiao, Peter Hawkins, Sid Dalmia, Clement Farabet, Pedro Valenzuela, Quan Yuan, Ananth Agarwal, Mia Chen, Wooyeol Kim, Brice Hulse, Nandita Dukkipati, Adam Paszke, Andrew Bolt, Kiam Choo, Jennifer Beattie, Jennifer Prendki, Harsha Vashisht, Rebeca Santamaria-Fernandez, Luis C. Cobo, Jarek Wilkiewicz, David Madras, Ali Elqursh, Grant Uy, Kevin Ramirez, Matt Harvey, Tyler Liechty, Heiga Zen, Jeff Seibert, Clara Huiyi Hu, Andrey Khorlin, Maigo Le, Asaf Aharoni, Megan Li, Lily Wang, Sandeep Kumar, Norman Casagrande, Jay Hoover, Dalia El Badawy, David Soergel, Denis Vnukov, Matt Miecnikowski, Jiri Simsa, Praveen Kumar, Thibault Sellam, Daniel Vlasic, Samira Daruki, Nir Shabat, John Zhang, Guolong Su, Jiageng Zhang, Jeremiah Liu, Yi Sun, Evan Palmer, Alireza Ghaffarkhah, Xi Xiong, Victor Cotruta, Michael Fink, Lucas Dixon, Ashwin Sreevatsa, Adrian Goedeckemeyer, Alek Dimitriev, Mohsen Jafari, Remi Crocker, Nicholas FitzGerald, Aviral Kumar, Sanjay Ghemawat, Ivan Philips, Frederick Liu, Yannie Liang, Rachel Sterneck, Alena Repina, Marcus Wu, Laura Knight, Marin Georgiev, Hyo Lee, Harry Askham, Abhishek Chakladar, Annie Louis, Carl Crous, Hardie Cate, Dessie Petrova, Michael Quinn, Denese Owusu-Afriyie, Achintya Singhal, Nan Wei, Solomon Kim, Damien Vincent, Milad Nasr, Christopher A. Choquette-Choo, Reiko Tojo, Shawn Lu, Diego de Las Casas, Yuchung Cheng, Tolga Bolukbasi, Katherine Lee, Saaber Fatehi, Rajagopal Ananthanarayanan, Miteyan Patel, Charbel Kaed, Jing Li, Shreyas Rammohan Belle, Zhe Chen, Jaclyn Konzelmann, Siim Põder, Roopal Garg, Vinod Koverkathu, Adam Brown, Chris Dyer, Rosanne Liu, Azade Nova, Jun Xu, Alanna Walton, Alicia Parrish, Mark Epstein, Sara McCarthy, Slav Petrov, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

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, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023. Orthogonal subspace learning for language model continual learning. *Preprint*, arXiv:2310.14152.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. Long-form factuality in large language models. *Preprint*, arXiv:2403.18802.

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ying Shan, and Ping Luo. 2024. Llama pro: Progressive llama with block expansion. *Preprint*, arXiv:2401.02415.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *Preprint*, arXiv:2304.12244.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *Preprint*, arXiv:2406.08464.

Yibo Yang, Stephan Mandt, and Lucas Theis. 2023. An introduction to neural data compression. *Preprint*, arXiv:2202.06533.

Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. 2024. Self-distillation bridges distribution gap in language model fine-tuning. *Preprint*, arXiv:2402.13669.

Asaf Yehudai, Boaz Carmeli, Yosi Mass, Ofir Arviv, Nathaniel Mills, Assaf Toledo, Eyal Shnarch, and Leshem Choshen. 2024. Genie: Achieving human parity in content-grounded datasets generation. *Preprint*, arXiv:2401.14367.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *Preprint*, arXiv:2306.13549.

Li Yunxiang, Li Zihan, Zhang Kai, Dan Ruilong, and Zhang You. 2023. Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge. *arXiv preprint arXiv:2303.14070*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *Preprint*, arXiv:1905.07830.

A Appendix

A.1 Experiment Detail for Figure 1

- Following LLaVA (Liu et al., 2023), we select the MLLM benchmarks including:

  gqa, textvqa_val, pope, mme, seedbench, mmbench_cn_dev, mmbench_en_dev, scienceqa_img, vqav2_val, vizwiz_vqa_val

- Following LLaMAPro (Wu et al., 2024), we select the LLM benchmarks including:

  arc_challenge (25-shot), gsm8k (5-shot), hellaswag (10-shot), mmlu (5-shot), winogrande (5shot), truthfulqa (0-shot)

- We used the llava (Liu et al., 2023) codebase (https://github.com/haotian-liu/LLaVA) to conduct the experiment. We first trained the visual projector for the LLaMA2-7B-chat model and SFT the LLaMA2-7B-chat with this project. All training data and training configurations strictly followed the LLaVA repo.