Towards the Law of Capacity Gap in Distilling Language Models

Chen Zhang[♣], Qiuchi Li[★], Dawei Song[♣✦*], Zheyu Ye[♥], Yan Gao[♥], Yan Hu[♥]
[♣]Beijing Institute of Technology [★]University of Copenhagen
[✦]The Open University [♥]Xiaohongshu
chenzhang9702@outlook.com

Abstract

Language model (LM) distillation aims at distilling the knowledge in a large teacher LM to a small student one. As a critical issue facing LM distillation, a superior student often arises from a teacher of a relatively small scale instead of a larger one, especially in the presence of substantial capacity gap between the teacher and student. This issue, often referred to as the *curse of capacity gap*, suggests that there is likely an optimal teacher yielding the best-performing student along the scaling course of the teacher. Consequently, distillation trials on teachers of a wide range of scales are called for to determine the optimal teacher, which becomes computationally intensive in the context of large LMs (LLMs). This paper addresses this critical bottleneck by providing the *law of capacity gap* inducted from a preliminary study on distilling a broad range of small-scale (<3B) LMs, where the optimal teacher consistently scales linearly with the student scale across different model and data scales. By extending the law to LLM distillation on a larger scale (7B), we succeed in obtaining versatile LLMs that outperform a wide array of competitors.[1]

1 Introduction

Language model (LM) distillation is designed to distill from a large teacher LM to a small student LM. With the regime of the teacher-student paradigm, it is expected that the student LM could mimic the behaviors of the teacher LM and achieve appealing performance as the teacher LM does (Hinton et al., 2015; Sanh et al., 2019; Jiao et al., 2020).

However, it has been broadly observed that the performance of student LM is not always improved

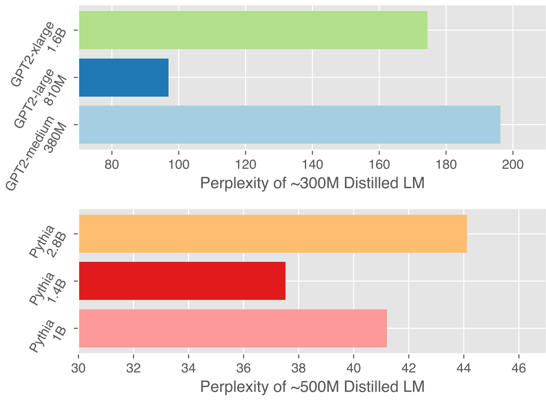


Figure 1: The curse of capacity gap. GPT2 (Radford et al., 2019) and Pythia (Biderman et al., 2023) distilled with OpenWebText (Gokaslan and Cohen, 2019), and evaluated on WikiText2 (Merity et al., 2017) in perplexity. *curse*, the performance of a fixed student scale does not improve along the increased teacher scale.

but often degraded as the teacher scales up, termed as the *curse of capacity gap*. As shown in Figure 1, the optimal teacher that accords with the best student performance shall not be the largest one (Mirzadeh et al., 2020; Zhang et al., 2022a; Zhou et al., 2022; Yang et al., 2022). Consequently, computationally enumerative distillation trials need to be conducted from different scales of teacher LMs in order to obtain the optimal teacher. This can evolve into a critical bottleneck in the context of large LMs (LLMs, Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023a; OpenAI, 2022). Essentially, an *impossible triangle* has been evoked among expected student scale, optimal teacher scale, and small compute overhead as shown in Figure 2.

This paper addresses the impossible triangle of capacity gap by exploring whether a fixed relation exists between the expected student scale and its optimal teacher scale, namely the *law of capacity gap*. Through a pilot study, we empirically reveal a constantly linear correlation between the student scale and optimal teacher scale, which holds per-

*Dawei Song is the corresponding author.
[1]The repository is at https://github.com/GeneZC/MiniMA. The model collection is placed at https://huggingface.co/collections/GeneZC/minima-family-6695f13d461de4eea59d83a3

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 22504–22528
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

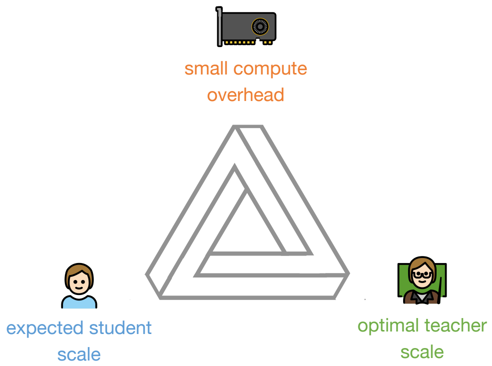


Figure 2: The curse of capacity gap can result in an impossible triangle in the era of LLMs. Optimal teacher scale according to expected student scale can not be yielded via small compute overhead, besides the one required by the oracle distillation.

fectly across varying model and data scales. This transforms the impossible triangle to a possible one by eliminating the compute overhead involved in determining the optimal teacher scale.

To empirically validate the broad applicability of the law, we are driven to distill LMs of larger scales based off the law. Specifically, we distill common LLMs to the most suitable student scales under the law, and compared the performance of the student LMs to a wide range of efficient designs on an extensive variety of benchmarks. The results demonstrate that the law still holds and the student LMs bring superior performance over baselines with the same compute budget. The student LMs also possess strong instruction-following capabilities after finetuning, surpassing an array of scale-matched competitors and rivaling LLMs of larger scales.

To sum up, our contributions are summarized to three folds:

- We have put forth the impossible triangle as a result of the curse of capacity gap, and pointed out the critical bottleneck it brings to distillation in the era of LLMs.
- We have discovered and formally proposed the law of capacity gap, where the expected student scale and its optimal teacher scale follow a linear scaling. It eliminates the compute overhead in distilling LLMs.
- The law has led to compute-efficient distillation of LLMs on a larger scale, with a notable improvement on fundamental language understanding tasks as well as instruction-following benchmarks.

2 Background

2.1 LM Distillation

Ever since the birth of transformer-derived LMs (Vaswani et al., 2017; Devlin et al., 2019; Liu et al., 2019; Radford et al., 2018; Raffel et al., 2020), a surge of interests have been put around compression of LMs (Bucila et al., 2006), in which distillation of LMs (Hinton et al., 2015; Sanh et al., 2019; Jiao et al., 2020) is a crucial topic other than quantization (Kim et al., 2021; Xiao et al., 2023; Dettmers et al., 2023), pruning (Xia et al., 2022; Frantar and Alistarh, 2023; Ma et al., 2023), and dynamic networking (early exiting, moefication, compute elasticity, etc.) (Xin et al., 2020; Li et al., 2021; Zhang et al., 2022c; Zuo et al., 2022; Zhang et al., 2023a). While initial focus has been allocated to task-specific distillation that distills finetuned LMs with data of finetuning scale (Sun et al., 2019), later focus has shifted to task-agnostic distillation that directly distills pretrained LM with data of pretraining scale (Wang et al., 2021), mainly due to its remarkable performance and task flexibility delivered by the task-agnostic property.

Recent LLMs have boosted a new emergence of compression of LLMs (Frantar et al., 2022; Sun et al., 2023; Hsieh et al., 2023; Zhu et al., 2023). However, there is a lack of research in LLM distillation, especially in a task-agnostic fashion (Fu et al., 2023; Zhang et al., 2023c; Jha et al., 2023; Gu et al., 2023; Agarwal et al., 2023). This is at least partly, if not exclusively, owing to the intensive compute demand of task-agnostic distillation of LLMs, which becomes a fatal bottleneck in the presence of the curse of capacity gap.

It should be noted that there are a huge body of studies concentrating on pseudo (alias black-box) distillation (Kim and Rush, 2016) either at pretraining (Gunasekar et al., 2023), supervised finetuning (Ho et al., 2023; Magister et al., 2023; Xu et al., 2023a), or preference optimization (Tunstall et al., 2023) stage. The popularity of pseudo distillation is largely attributed to that many powerful LLMs are proprietary (e.g., GPT4). In this context, the concern of capacity gap might be rather waived along the reduction of teacher knowledge from informative distributed probabilities to one-hot labels.

2.2 Curse of Capacity Gap

It was originally recognized that the curse of capacity gap exists in distillation of vision models (Mirzadeh et al., 2020), and recently shown that the curse lies in distillation of LMs (Zhang et al., 2023b) via thorough inspections. The curse of capacity essentially states that a larger teacher LM would not always lead to a better student LM in spite of a stronger performance itself. It also points out that the student is impacted not only by the teacher performance but also by the capacity gap, thus a tradeoff. Consequently, the optimal capacity gap is located somewhere that needs to be discovered as in Figure 1. In addition to the intuitive explanation, a theoretical justification could be found in Theorem 1 in Zhang et al. (2023b).

Although previous studies have striven to lift the curse of capacity gap (Zhang et al., 2022a; Zhou et al., 2022; Yang et al., 2022; Zhang et al., 2023b), the curse can only be partially lifted with regard to the increasingly large scales of teachers, say LLMs. Taking into account the resource demand of LLMs, this work takes a new perspective on the curse, inspired to a certain degree by the law of k-bit quantization (Dettmers and Zettlemoyer, 2023). The new view motivates us to unveil the buried law beneath the curse instead of sticking with the efforts on lifting it. Even though the optimal capacity gap has been proven to exist (Zhang et al., 2023b), it is difficult to derive a mathematical law to determine it. Inspired by the spirits of scaling law in language modeling (Kaplan et al., 2020), we carry out empirical studies to deduct a potential law.

3 Exploring the Law of Capacity Gap

Regarding the impossible triangle, we are motivated to explore whether the optimal teacher scale can be determined from the expected student scale with little compute overhead. For this purpose, we presume that there exists a perfectly fitted relation between the expected student scale and the optimal teacher scale, referred to as the *law of capacity gap*:

Proposition 1. *Provided a to-be-distilled student of an expected scale, the teacher of an optimal scale can be uniquely determined through a scaling relation.*

Once verified, the law can remarkably spare computing efforts and resources in deciding in which scale should a preferred teacher be.

3.1 Overview

A pilot study is carried out to investigate whether such a law of capacity gap holds, where we target at conducting small-scale pilot explorations and extrapolating them to large-scale ones. We believe the approximation is acceptable referring to up-to-date experience (Xie et al., 2023; Kaplan et al., 2020) that suggests the validity of the extrapolation.

Briefly, in each of the comparably small-scale exploration, we attempt to 1) take an LM of a specific scale at a time as a teacher LM $\mathcal{T}$, 2) prune the teacher LM to a specific sparsity as a student LM $\mathcal{S}$, 3) distill from the teacher LM to the student LM, and 4) observe the scaling relation between the optimal teacher LM resulting in the best performance and the corresponding student LM.

3.2 Method

Pruning In ignorance of architectural differences and details such as biases and normalizations, a causal LM is typically decomposed into a stack of transformer layers, each of which further includes a multi-head self-attention (MHA) block and a feed-forward network (FFN) block. We mainly prune the attention heads of MHA blocks, the intermediate neurons of FFN blocks, and the hidden states (thereby embeddings) based on their parameter expressive scores (Molchanov et al., 2017).

Inspired by structured pruning (Michel et al., 2019), we attach a set of variables ξ, ν per layer to the attention heads, the intermediate neurons; and a variable μ shared across layers (Xia et al., 2022) to the hidden states, as shown below:

$$
\begin{aligned}
\text{MHA}(\mathbf{x}_i) &= \sum_{j=1}^{A} \xi_j \cdot \text{Attn}(\mu \cdot \mathbf{x}_{\leq i}; \mathbf{W}_j^{\mathsf{Q}}, \mathbf{W}_j^{\mathsf{K}}) \\
&\quad \cdot \mu \cdot \mathbf{x}_{\leq i} \mathbf{W}_j^{\mathsf{V}} \mathbf{W}_j^{\mathsf{O}}, \\
\text{FFN}(\mathbf{x}_i) &= \sum_{k=1}^{I} \nu_k \cdot \text{g}(\mathbf{x}_i \mathbf{W}_k^{\mathsf{I}}) \mathbf{W}_k^{\mathsf{O}},
\end{aligned} \tag{1}
$$

where $\mathbf{x}_i$ serves as a token hidden state vector, and $\text{g}(\cdot)$ is an activation function. The j-th head among A heads is parameterized by $\mathbf{W}_j^{\mathsf{Q}}$, $\mathbf{W}_j^{\mathsf{K}}$, $\mathbf{W}_j^{\mathsf{V}}$, and $\mathbf{W}_j^{\mathsf{O}}$. The k-th neuron among I neurons is parameterized by $\mathbf{W}_k^{\mathsf{I}}$ and $\mathbf{W}_k^{\mathsf{O}}$ respectively.

The parameter expressive scores are then calculated through accumulated absolute gradients. A higher expressive score indicates that the corresponding parameter has a higher contribution to the model and hence a lower priority of pruning. The expressive scores of the attention heads, intermediate neurons and hidden states are determined by:

$$\mathbb{I}_j^{\text{head}} = \mathbb{E}_{\mathbf{x}_i}\left|\frac{\partial \mathcal{L}^{\text{pruning}}}{\partial \xi_j}\right|,$$
$$\mathbb{I}_j^{\text{neuron}} = \mathbb{E}_{\mathbf{x}_i}\left|\frac{\partial \mathcal{L}^{\text{pruning}}}{\partial \nu_j}\right|, \quad (2)$$
$$\mathbb{I}^{\text{hidden}} = \mathbb{E}_{\mathbf{x}_i}\left|\frac{\partial \mathcal{L}^{\text{pruning}}}{\partial \mu}\right|,$$

$$\mathcal{L}^{\text{pruning}} = \sum_i^{|\mathcal{V}|} -\mathbf{y}_i^{\text{truth}} \log \mathbf{y}_i^{\text{teacher}}, \quad (3)$$

where $\mathbb{E}$ represents expectation, $\mathbf{y}$ is the post-softmax probability distribution over the whole vocabulary $\mathcal{V}$. Following Molchanov et al. (2017), we apply ℓ_2 normalization to the expressive scores to prevent skewed pruning ratios across layers.

In the pruning procedure, the attention heads, intermediate neurons and hidden states are pruned separately with the same sparsity. For each component, the parameters are ranked according to their expressive scores as calculated above, and the ones with the lowest scores are sequentially pruned until the desired sparsity is reached. Since the sparsity of hidden states and attention heads (or intermediate neurons) have a product effect to the overall sparsity, the individual components are set to a sparsity ratio $p' = 1 - \sqrt{1-p}$ for reaching the target sparsity p of the whole model. It is noteworthy that the squared-root form of hidden state sparsity makes the embeddings be less pruned, but it makes a minimal difference compared with the parameters pruned in MHA and FFN blocks.

Distillation We conduct distillation from the teacher to the student with the most basic token-level cross-entropy loss (Hinton et al., 2015) as below:

$$\begin{aligned}\mathcal{L}^{\text{distillation}} &= \frac{1}{2}\sum_i^{|\mathcal{V}|} -\mathbf{y}_i^{\text{teacher}} \log \mathbf{y}_i^{\text{student}} \\ &+ \frac{1}{2}\sum_i^{|\mathcal{V}|} -\mathbf{y}_i^{\text{truth}} \log \mathbf{y}_i^{\text{student}}.\end{aligned} \quad (4)$$

While there are alternative distillation objectives, such as the ones on the sequence level (Wen et al., 2023), they are not taken into consideration due to a relatively low efficiency.

3.3 Setup

The explorations involve the pruning and distillation of GPT2 (Radford et al., 2019) and Pythia (Biderman et al., 2023) with OpenWebText (Gokaslan and Cohen, 2019). Each of these LM series comprises LMs of different scales, i.e., GPT2-base (140M), -medium (380M), -large (810M), -xlarge (1.6B); and Pythia-70M, -160M, -410M, -1B, -1.4B, -2.8B. Each LM is distilled into student LMs of varying sizes through sparsity p. The perplexity on the WikiText2 and last word prediction accuracy of LAMBADA are computed as the performance indicators of student LMs. OpenWebText consists of ∼40G web data that amounts to 3.9B tokens, which are in a similar data scale to that of GPT2 pretraining. The data scale variation is also considered by reducing the number of training tokens. We use a subset of the data (∼5%) for pruning here and hereafter for a high pruning efficiency.

3.4 Observations

The explorations produce results of distillation teacher-student pairs with different capacity gaps so that we can form the scaling plots as in Figure 3. In each plot, a colored line corresponds to a certain sparsity, and points with the same horizontal order in all lines correspond to the same teacher LM. For exploring the law of capacity gap, the following hypotheses should be examined:

H1: the capacity gap influences the student performance.

H2: the optimal teacher scale exists in finite spaces, given a student scale.

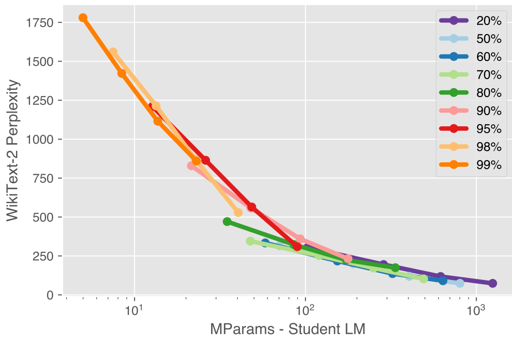
H3: the optimal teacher scale does not vary across settings, given a student scale.

It can be observed from the plots that a) for a fixed-scale teacher LM, the performance of student always improves or stabilizes along the increment of its scale; b) when the student scale is fixed, its performance invariably follows an increasing-steady-decreasing pattern as the gradual increase in the teacher scale, implying the existence of an optimal teacher. The findings support **H1** and **H2**, respectively.

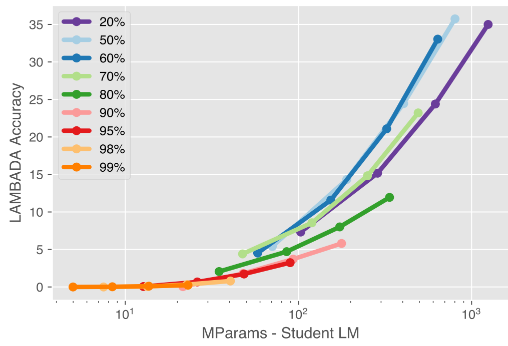
Furthermore, we discover that the optimal teacher scale for a certain student scale is consistent across distinct student scales and data scales, at an approximate sparsity of ∼60%. This unearths that **H3** also holds true.

3.5 Characterization

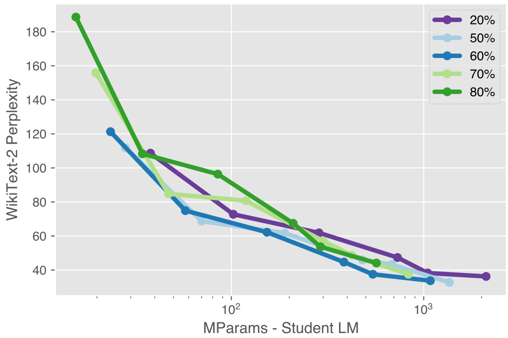
The validity of the three hypotheses suggests the existence of the law of capacity gap, i.e., a consistent relation between the scale of target student and the optimal teacher scale. We attempt to move a step further by characterizing the law.



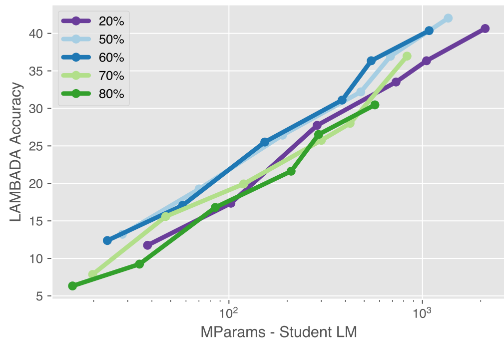
(a) GPT2 on WikiText2.



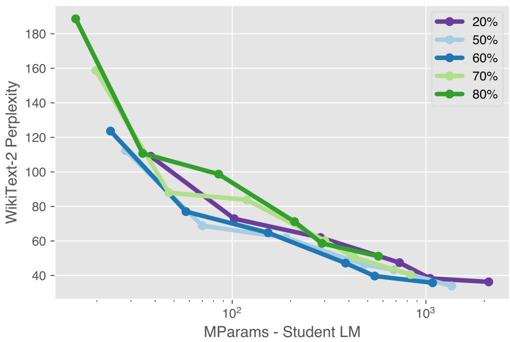
(b) GPT2 on LAMBADA.



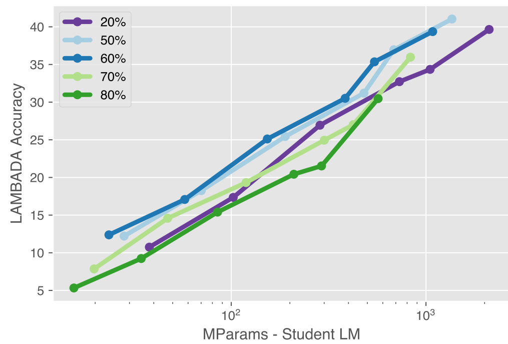
(c) Pythia on WikiText2.



(d) Pythia on LAMBADA.



(e) Pythia on WikiText2 with fewer tokens.



(f) Pythia on LAMBADA with fewer tokens.

Figure 3: The observations from the distillation of GPT2 and Pythia series. Students are evaluated on the test set of WikiText2 (Merity et al., 2017) in perplexity and the test set of LAMBADA (Paperno et al., 2016) in last word prediction accuracy. A line of a color represents a sparsity, where each point in it represents a student pruned and distilled from a teacher at such sparsity.

For this purpose, we collect the optimal teacher LMs for different student LMs from the explorations and draw a scatter plot for the obtained (student scale, teacher scale) pairs, as shown in Figure 4. It can be seen that the points appear to locate on a straight line. Therefore, we apply fitting to the points as below:

$$\mathbb{T}^\star \approx \alpha \cdot \mathbb{S} + \beta \tag{5}$$

The line of best fit is obtained with $\alpha = 2.498$ and $\beta = -11.498$, with $R^2 = 0.9957$ indicating a perfect fit. This means that the curse of capacity gap can be appropriately viewed as a law:

Law 1. *Given a to-be-distilled student $\mathcal{S}$ of an expected scale $\mathbb{S}$, a teacher $\mathcal{T}^\star$ of an optimal scale $\mathbb{T}$ should be:*

$$\mathbb{T}^\star \approx 2.498 \cdot \mathbb{S} - 11.498 \approx 2.5 \cdot \mathbb{S} \tag{6}$$

And the law promptly triggers a follow-up corollary:

Corollary 1. *On condition that a to-be-distilled student $\mathcal{S}$ is of a scale $0.4 \cdot \mathbb{T}^\star$, a teacher $\mathcal{T}$ of a scale $\mathbb{T}^\star$ should be the optimal teacher for $\mathcal{S}$.*

Remark 1. *The law of capacity gap is drawn under specific design choices. Nevertheless, we do consider the law under different teacher architectures (e.g., GPT2 in Figure 3a versus Pythia in*

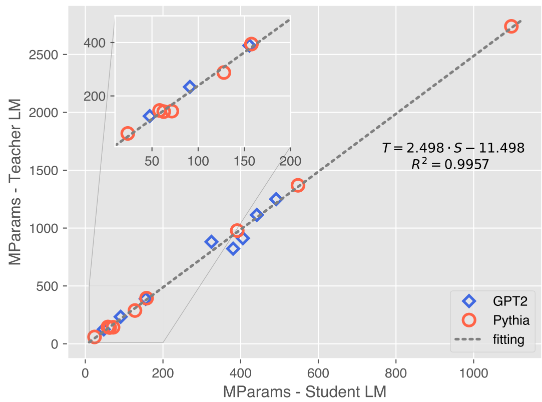


Figure 4: The curse of capacity gap can be leaned towards the law of capacity gap. *law*, the optimal teacher scale exists and remains linear to the student scale. Each point stands for the best teacher scale for a fixed student scale.

Figure 3c), data scales (e.g., full tokens in Figure 3d versus fewer tokens in Figure 3f), pruning paradigms (c.f., Section 4.3), and distillation objectives (c.f., Section 4.3). Empirical evidence suggests that the basic law universally applies to all these circumstances.

There is a trend where small language models (SLMs) are distilled rather than trained from scratch (Gunter et al., 2024; Rivière et al., 2024). Interestingly, almost all these distilled SLMs use teacher models whose scales roughly follow our linear law.

4 Extrapolating the Law of Capacity Gap

We have drawn the law of capacity gap from small-scale explorations. Yet, whether the law holds for LLMs on a large scale remains unknown. To answer this question, we carry out a series of experiments on extrapolating the law to larger-scale experiments to exhibit the extrapolating ability of the law.

4.1 Setup

Table 1: The statistics of data mixture used for distillation.

Dataset	Epochs	Tokens	Proportion
Pile	0.16	57B	45.2%
GitHub	0.50	30B	23.8%
WuDao	0.78	39B	31.0%
Mixture	1.00	126B	100.0%

Target We distill popular LLaMA2-7B (Touvron et al., 2023b) and LLaMA3.1-8B (Dubey et al., 2024) to $0.4\times7\approx3$B LMs named MINIMA. Considering potential applications to Chinese tasks, we do not directly leverage LLaMA2-7B released by Meta but use an adapted one. The adapted LLaMA2-7B is expanded with a Chinese vocabulary, incrementally trained on a mixture of sub-sampled Pile (Gao et al., 2021), GitHub (Together, 2023), and WuDao (Yuan et al., 2021), as detailed in Appendix A. In contrast, LLaMA3.1-8B is already armed with a Chinese-enhanced vocabulary. The mixed data is also utilized to conduct pruning and distillation, and the statistics are in Table 1.

Taking a step further, we finetune MINIMA on instruction-following data and get MINICHAT. The instruction-following data covers a broad collection of both single-turn question-answering pairs and multi-turn conversation sessions, which totally adds up to 1.1M examples. The statistics are detailed in Appendix B due to the space limitation.

Implementation Different from the pruning in pilot explorations, we additionally impose a heuristic rule to the pruning priority of MINIMA so that it can produce symmetric shapes across layers and make MINIMA universally applicable without any monkey-patches in succeeding use. The heuristic pruning priority first converts the previously used global priority to a local layer-wise one, then compensates for the loss of the entire-layer pruning ability by heuristically dropping off the 4 bottommost and 4 topmost layers, and accordingly pruning few hidden states (as well as embeddings plus language model head), MHA heads and FFN neurons. The reasons behind such design choice are detailed in Appendix C.

The distillation takes in data packed as sequences of 4,096 tokens. The batch size is set to 1,024 (~4M tokens). The learning rate is 3e-4, and the weight decay is 1e-1. The training lasts for only 1 epoch. The learning rate is scheduled to warm up linearly for the first 1% steps of all and decay down in a sinusoidal way for the rest steps. Gradients whose norms accumulate over 1.0 are necessarily clipped. The training is executed on 16 A100 80G GPUs. The training efficiency is guaranteed by DeepSpeed Zero2 (Rasley et al., 2020) and FlashAttention (Dao et al., 2022). Gradient checkpointing is enabled to further reduce the memory footprint and bfloat16 precision is chosen to improve the training stability.

Table 2: The results of MiniMA on standard benchmarks. The best results are **boldfaced**. The results marked with † are made compute-comparable to MiniMA. The results marked with ‡ are produced with a larger corpus due to a larger vocabulary.

LM	Params	Tokens	MMLU Acc	CEval Acc	DROP EM Score	BBH EM Score	GSM8K Maj1@1	HumanEval Pass@1
LLaMA2	7 B	2 T	46.00	34.40	31.57	32.02	14.10	12.80
- ShortGPT	3 B	0 B	25.57	26.79	8.72	7.53	4.52	0.00
- LayerChop	3 B	126 B	25.76	27.23	10.45	12.23	6.97	3.66
- ShearedLLaMA	3 B	126 B	25.39	27.79	20.41	30.21	5.08	8.54
- ShearedLLaMA†	3 B	252 B	26.15	27.81	20.03	30.11	5.26	9.01
- MiniMA	3 B	126 B	**28.51**	**28.23**	**22.50**	**31.61**	**8.11**	**10.98**
LLaMA2 (2023b)	13 B	2 T	55.34	41.60	43.40	38.07	24.11	14.02
- MiniMA	3 B	126 B	26.82	28.23	18.92	30.01	5.31	10.37
LLaMA2 (2023b)	70 B	2 T	68.62	53.86	63.31	51.58	53.37	28.66
- MiniMA	3 B	126 B	26.91	24.89	19.03	28.55	3.56	7.32
LLaMA3.1 (2024)	8 B	15 T	64.75	52.45	49.02	41.00	48.82	34.15
- MiniMA‡	3 B	126 B	31.72	29.89	30.46	32.06	10.42	18.46

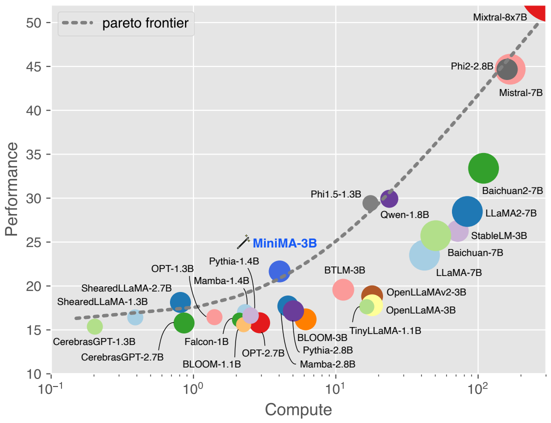


Figure 5: The new compute-performance pareto frontier is yielded by MiniMA, namely MiniMA is more compute-efficient given any compute budget than existing LMs. The radius of each circle stands for the model scale. *performance*: average task measure as each detailed in Appendix F. *compute*: estimated training compute in $\times 10^9$ TFLOPs as detailed in Appendix H.

The finetuning receives data padded as sequences of 4,096 tokens. The batch size is set to 256 (∼1M tokens). The learning rate is 2e-5, and the weight decay is 1e-1. The training lasts for 3 epochs with early stopping after continuously 2 epochs without validation improvement. The learning rate is scheduled to warm up linearly for the first 10% steps of all and decay down in a sinusoidal way for the rest steps. Gradients whose norms accumulate over 1 are necessarily clipped. The training is executed on 8 A100 80G GPUs. The training efficiency is again guaranteed likewise by those in the distillation.

4.2 Main Results

Standard Benchmarks We evaluate MiniMA against an extensive set of baselines on several standard benchmarks including MMLU (Hendrycks et al., 2021), CEval (Huang et al., 2023), DROP (Dua et al., 2019), BBH (Suzgun et al., 2023), GSM8K (Cobbe et al., 2021), and HumanEval (Chen et al., 2021). The introductions and evaluation protocols of these datasets are detailed in Appendix D. We compare MiniMA against a few models involving efficient designs, such as layer dropping: ShortGPT (Men et al., 2024), LayerChop (Jha et al., 2023), and pruning: ShearedLLaMA (Xia et al., 2023).

The results in Table 2 demonstrate that LLaMA2-7B is exactly the best among considered LLaMA2-{7,13,70}B. And the performance of MiniMA consistently degrades along the increased scales of teacher LMs. This phenomenon implies the practical value of the law. Moreover, MiniMA achieves superior performance over baselines with respect to knowledge (MMLU, CEval), reasoning (DROP, BBH, GSM8K), and coding (HumanEval). Even with similar compute consumed, MiniMA yet outperforms ShearedLLaMA. This indicates that distillation, especially when promoted by the law, is competitive among other efficient designs. Further, MiniMA from LLaMA3.1 is better than that from LLaMA2, showcasing the prominence of MiniMA via adopting more promising teacher LMs. The training loss-level comparisons among them are also detailed in Appendix E for potential interests.

From a broader comparison that involves exist-

Table 3: The results of MiniChat on GPT4 assessments. Macro average scores are reported across fields in these two datasets. The better results are **boldfaced**.

LM Pair	Vicuna-Bench Macro Avg	BELLE-Bench Macro Avg
MiniChat		
v.s. OpenBuddy-3B	**7.64**: 5.42	**7.77**: 6.81
v.s. BiLLa-7B	7.24: **7.41**	**7.73**: 7.49
v.s. ChatGLM-6B	**7.63**: 5.63	**7.44**: 7.23
v.s. Phoenix-7B	**7.35**: 6.95	**7.62**: 7.21
v.s. ChatGLM2-6B	**7.35**: 7.30	7.40: **8.00**

ing LMs of various scales, as in Figure 5, we could clearly see that MiniMA basically builds an excitingly new compute-performance Pareto frontier therein.

GPT4 Assessments We compare MiniChat to some instruction-following LMs on Vicuna-Bench (Chiang et al., 2023) and BELLE-bench (Ji et al., 2023), which demand GPT4 for pairwise assessments. The introductions and evaluation protocols of these datasets are detailed in Appendix I. The regarded instruction-following baselines encompass scale-matched ones: OpenBuddy-3B (OpenBuddyAI, 2023), and scale-mismatched ones: BiLLa-7B (Li, 2023), ChatGLM-6B (ZhipuAI, 2023a), Phoenix-7B (Chen et al., 2023), ChatGLM2-6B (ZhipuAI, 2023b).

The results in Table 3 disclose that MiniChat surpasses scale-matched 3B baseline OpenBuddy-3B by a landslide, and even keeps competitive with scale-mismatched baselines. In detail, MiniChat only falls behind BiLLa-7B on Vicuna-Bench and ChatGLM2-6B on BELLE-Bench. The results of MiniChat on standard benchmarks are detailed in Appendix J in case of curiosity. The field-specific scores of mentioned LMs are provided in Figure 6 in the form of ability radar, and further detailed in Appendix K. The micro evaluation displays the reasonable ability realized by MiniChat. The consistency of GPT4 evaluation to human evaluation is detailed in Appendix L.

Albeit this, the winners in macro average scores are not necessarily the winners in win-lose-tie battles as detailed in Figure 7. For example, MiniChat is way better than Phoenix on Vicuna-Bench in terms of Macro Average yet lags behind Phoenix in terms of win rate. In summary, the excellence of MiniChat encourages us to further

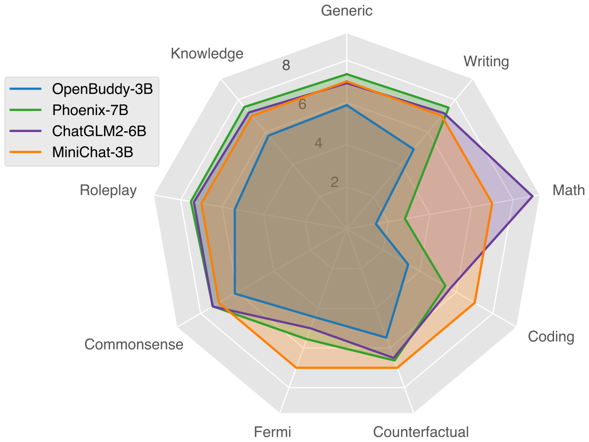


Figure 6: The reasonable ability radar is realized by MiniChat. The head-to-head comparison scores on Vicuna-Bench of each baseline are rebased according to MiniChat.

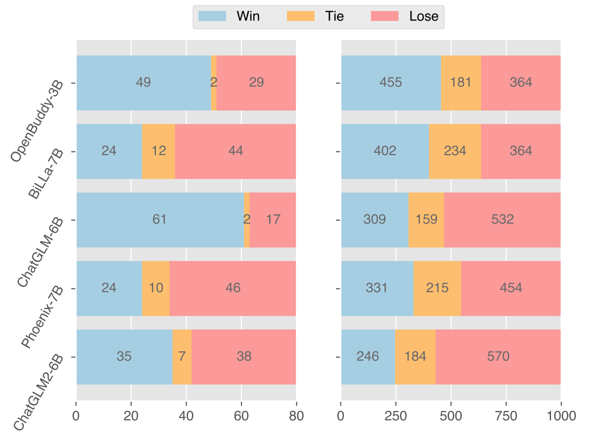


Figure 7: The win-lose-tie portions of MiniChat versus its competitors. The left is reported on the Vicuna-Bench while the right is reported on the BELLE-Bench.

explore the application of MiniMA.

4.3 Ablation Studies

To further validate the law, we compare MiniMA to its variants with different pruning strategies: global pruning (i.e., asymmetric shaping), layer dropping (i.e., depth pruning, Men et al., 2024), and different distillation techniques: patient knowledge (i.e., taking last-layer hidden states as knowledge, Sun et al., 2019), teacher assistant (i.e., inserting a 5B teacher assistant, Mirzadeh et al., 2020).

The results in Table 4 evidence that the specializations on the pruning priority of MiniMA would not harm (sometimes yet benefit to) the performance, when compared to either global pruning or layer dropping. This hints the rationality of symmetric shaping as described in hands-on implementation, and the adequacy of the employed

Table 4: The results of ablation studies.

LM	MMLU Acc	CEval Acc	DROP EM Score	BBH EM Score	GSM8K Maj1@1	HumanEval Pass@1
MINIMA	28.51	28.23	22.50	31.61	8.11	10.98
- global pruning	28.92	28.52	19.55	32.89	7.26	11.37
- layer dropping	27.59	26.95	21.96	30.37	6.85	9.86
- patient knowledge	28.69	28.37	22.51	31.89	8.00	10.64
- teacher assistant	27.69	27.49	21.31	30.02	7.72	9.58

pruning paradigm in drawing the law. Besides, the naive knowledge distillation (Hinton et al., 2015) is averagely comparable to advanced tactics like patient knowledge distillation, coinciding with Muralidharan et al. (2024, Table 19). This features the distillation objective in use would be relatively minor concerning the law. A counter-argument would be that the teacher assistant can not improve over ours, since the distillation path is sub-optimal. Nevertheless, no one can assure the distillation from LLaMA2-70B still follows this truth. As a whole, the above phenomena reflect the practical solidity of the law.

5 Conclusions and Future Work

In this paper, we propose a novel viewpoint towards the capacity gap in distilling LMs in the era of LLMs to counter the impossible triangle of capacity gap deviated from the curse of capacity gap. Via this port, we beneficially turn the curse of capacity gap into a law through pilot explorations. Activated by the law, we put forward extrapolation experiments resulting into valuable SLMs, establishing a state-of-the-art compute-performance Pareto frontier. We have also managed to draw a comprehensive panorama of MiniMA family by landing related models as in Appendix N.

Limitations

There are a few obvious limitations in this work: 1) we do not carry out very complete experiments for MINIMA while LLaMA3 grants a larger model at even 400B scale, 2) we do not consider data scale, data mixture, and distillation techniques as core scaling factors, and 3) we do not combine a preference optimization stage for MINICHAT while approaches like DPO (Rafailov et al., 2023) can principally alleviate the safety concerns and lift the user experiences. Hopefully, we could address these in future work.

Ethics Statements

MINICHAT is subject to the same well-recognized limitations of other LLMs, including potential for non-factual generation such as unqualified advice, and a propensity towards hallucinations. And like other LLMs, MINICHAT may generate harmful, offensive, or biased content due to its training on publicly available online datasets. Not everyone who uses LLMs has good intentions, and MINICHAT could potentially be used for nefarious purposes such as generating misinformation.

Acknowledgments

This work is funded in part by the Natural Science Foundation of China (grant no: 62376027).

References

Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2023. GKD: generalized knowledge distillation for auto-regressive sequence models. *CoRR*, abs/2306.13649.

Dzmitry Bahdanau. 2022. The flops calculus of language model training.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *CoRR*, abs/2309.16609.

Baichuan. 2023. A series of large language models developed by baichuan intelligent technology.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 535–541. ACM.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Zhihong Chen, Feng Jiang, Junying Chen, Tiannan Wang, Fei Yu, Guiming Chen, Hongbo Zhang, Juhao Liang, Chen Zhang, Zhiyi Zhang, Jianquan Li, Xiang Wan, Benyou Wang, and Haizhou Li. 2023. Phoenix: Democratizing chatgpt across languages. *CoRR*, abs/2304.10453.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

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. *CoRR*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *CoRR*, abs/2304.08177.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *CoRR*, abs/2305.14314.

Tim Dettmers and Luke Zettlemoyer. 2023. The case for 4-bit precision: k-bit inference scaling laws. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 7750–7774. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Nolan Dey, Gurpreet Gosal, Zhiming Chen, Hemant Khachane, William Marshall, Ribhu Pathria, Marvin Tom, and Joel Hestness. 2023a. Cerebras-gpt: Open compute-optimal language models trained on the cerebras wafer-scale cluster. *CoRR*, abs/2304.03208.

Nolan Dey, Daria Soboleva, Faisal Al-Khateeb, Bowen Yang, Ribhu Pathria, Hemant Khachane, Shaheer Muhammad, Zhiming Chen, Robert Myers, Jacob Robert Steeves, Natalia Vassilieva, Marvin Tom, and Joel Hestness. 2023b. BTLM-3B-8K: 7b parameter performance in a 3b parameter model. *CoRR*, abs/2309.11568.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2368–2378. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ: accurate post-training quantization for generative pre-trained transformers. *CoRR*, abs/2210.17323.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10421–10430. PMLR.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *CoRR*, abs/2312.00752.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. *CoRR*, abs/2306.08543.

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. *CoRR*, abs/2306.11644.

Tom Gunter, Zirui Wang, Chong Wang, Ruoming Pang, Andy Narayanan, Aonan Zhang, Bowen Zhang, Chen Chen, Chung-Cheng Chiu, David Qiu, Deepak Gopinath, Dian Ang Yap, Dong Yin, Feng Nan, Floris Weers, Guoli Yin, Haoshuo Huang, Jianyu Wang, Jiarui Lu, John Peebles, Ke Ye, Mark Lee, Nan Du, Qibin Chen, Quentin Keunebroek, Sam Wiseman, Syd Evans, Tao Lei, Vivek Rathod, Xiang Kong, Xianzhi Du, Yanghao Li, Yongqiang Wang, Yuan Gao, Zaid Ahmed, Zhaoyang Xu, Zhiyun Lu, Al Rashid, Albin Madappally Jose, Alec Doane, Alfredo Bencomo, Allison Vanderby, Andrew Hansen, Ankur Jain, Anupama Mann Anupama, Areeba Kamal, Bugu Wu, Carolina Brum, Charlie Maalouf, Chinguun Erdenebileg, Chris Dulhanty, Dominik Moritz, Doug Kang, Eduardo Jimenez, Evan Ladd, Fangping Shi, Felix Bai, Frank Chu, Fred Hohman, Hadas Kotek, Hannah Gillis Coleman, Jane Li, Jeffrey P. Bigham, Jeffery Cao, Jeff Lai, Jessica Cheung, Jiulong Shan, Joe Zhou, John Li, Jun Qin, Karanjeet Singh, Karla Vega, Kelvin Zou, Laura Heckman, Lauren Gardiner, Margit Bowler, Maria Cordell, Meng Cao, Nicole Hay, Nilesh Shahdadpuri, Otto Godwin, Pranay Dighe, Pushyami Rachapudi, Ramsey Tantawi, Roman Frigg, Sam Davarnia, Sanskruti Shah, Saptarshi Guha, Sasha Sirovica, Shen Ma, Shuang Ma, Simon Wang, Sulgi Kim, Suma Jayaram, Vaishaal Shankar, Varsha Paidi, Vivek Kumar, Xin Wang, Xin Zheng, and Walker Cheng. 2024. Apple intelligence foundation language models. *CoRR*, abs/2407.21075.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14852–14882. Association for Computational Linguistics.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14409–14428. Association for Computational Linguistics.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex 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. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8003–8017. Association for Computational Linguistics.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *CoRR*, abs/2305.08322.

Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. 2023. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *CoRR*, abs/2309.14509.

Ananya Harsh Jha, Dirk Groeneveld, Emma Strubell, and Iz Beltagy. 2023. Large language model distillation doesn't need a teacher. *CoRR*, abs/2305.14864.

Yunjie Ji, Yan Gong, Yong Deng, Yiping Peng, Qiang Niu, Baochang Ma, and Xiangang Li. 2023. Towards better instruction following language models for chinese: Investigating the impact of training data and evaluation. *CoRR*, abs/2304.07854.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4163–4174. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. I-BERT: integer-only BERT quantization. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5506–5518. PMLR.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1317–1327. The Association for Computational Linguistics.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. 2023. Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics.

Xiaonan Li, Yunfan Shao, Tianxiang Sun, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2021. Accelerating BERT inference for sequence labeling via early-exit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 189–199. Association for Computational Linguistics.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need II: phi-1.5 technical report. *CoRR*, abs/2309.05463.

Zhongli Li. 2023. Billa: A bilingual llama with enhanced reasoning ability.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *CoRR*, abs/2305.11627.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adámek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1773–1781. Association for Computational Linguistics.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2024. Shortgpt: Layers in large language models are more redundant than you expect. *CoRR*, abs/2403.03853.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14014–14024.

Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5191–5198. AAAI Press.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. 2024. Compact language models via pruning and knowledge distillation. *CoRR*, abs/2407.14679.

OpenAI. 2022. Introducing chatgpt.

OpenBuddyAI. 2023. Openbuddy - open multilingual chatbot for everyone.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon LLM: outperforming curated corpora with web data, and web data only. *CoRR*, abs/2306.01116.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *5th International Conference on Learning*

Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings. OpenReview.net.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *CoRR*, abs/2305.18290.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3505–3506. ACM.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew K. Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakas, and et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *CoRR*, abs/2206.04615.

StabilityAI. 2023. Stablelm-3b-4e1t technical report.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2023. A simple and effective pruning approach for large language models. *CoRR*, abs/2306.11695.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on*

Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 4322–4331. Association for Computational Linguistics.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13003–13051. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Teknium. 2023. A collection of modular datasets generated by gpt-4, general-instruct - roleplay-instruct - code-instruct - and toolformer.

Together. 2023. Redpajama, a project to create leading open-source models, starts by reproducing llama training dataset of over 1.2 trillion tokens.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment. *CoRR*, abs/2310.16944.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. Minilmv2: Multi-head self-attention relation distillation for compressing pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2140–2151. Association for Computational Linguistics.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10817–10834. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. *CoRR*, abs/2310.06694.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1513–1528. Association for Computational Linguistics.

Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 38087–38099. PMLR.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. *CoRR*, abs/2305.10429.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. Deebert: Dynamic early exiting for accelerating BERT inference. In *Proceedings of* the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 2246–2251. Association for Computational Linguistics.

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

Canwen Xu, Daya Guo, Nan Duan, and Julian J. McAuley. 2023b. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *CoRR*, abs/2304.01196.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, Juntao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models. *CoRR*, abs/2309.10305.

Yi Yang, Chen Zhang, and Dawei Song. 2022. Sparse teachers can be dense with knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3904–3915. Association for Computational Linguistics.

Sha Yuan, Hanyu Zhao, Zhengxiao Du, Ming Ding, Xiao Liu, Yukuo Cen, Xu Zou, Zhilin Yang, and Jie Tang. 2021. Wudaocorpora: A super large-scale chinese corpora for pre-training language models. *AI Open*, 2:65–68.

Chen Zhang, Benyou Wang, and Dawei Song. 2023a. On elastic language models. *CoRR*, abs/2311.07204.

Chen Zhang, Yang Yang, Jiahao Liu, Jingang Wang, Yunsen Xian, Benyou Wang, and Dawei Song. 2023b. Lifting the curse of capacity gap in distilling language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4535–4553. Association for Computational Linguistics.

Chen Zhang, Yang Yang, Jingang Wang, and Dawei Song. 2023c. Task-agnostic distillation of encoder-decoder language models. *CoRR*, abs/2305.12330.

Chen Zhang, Yang Yang, Qifan Wang, Jiahao Liu, Jingang Wang, Wei Wu, and Dawei Song. 2022a. Minidisc: Minimal distillation schedule for language model compression. *CoRR*, abs/2205.14570.

Ge Zhang, Yemin Shi, Ruibo Liu, Ruibin Yuan, Yizhi Li, Siwei Dong, Yu Shu, Zhaoqun Li, Zekun Wang, Chenghua Lin, Wenhao Huang, and Jie Fu. 2023d. Chinese open instruction generalist: A preliminary release. *CoRR*, abs/2304.07987.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2023e. Tinyllama.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022b. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022c. Moefication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 877–890. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *CoRR*, abs/2306.05685.

ZhipuAI. 2023a. Chatglm-6b: An open bilingual dialogue language model.

ZhipuAI. 2023b. Chatglm2-6b: An open bilingual chat llm.

Wangchunshu Zhou, Canwen Xu, and Julian J. McAuley. 2022. BERT learns to teach: Knowledge distillation with meta learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7037–7049. Association for Computational Linguistics.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023. A survey on model compression for large language models. *CoRR*, abs/2308.07633.

Simiao Zuo, Qingru Zhang, Chen Liang, Pengcheng He, Tuo Zhao, and Weizhu Chen. 2022. Moebert: from BERT to mixture-of-experts via importance-guided adaptation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1610–1623. Association for Computational Linguistics.

A Adaptation of LLaMA2

It is well identified that LLaMA and LLaMA2 are weak in Chinese language understanding and generation (Cui et al., 2023), impeding the distillation from LLaMA to MiniMA that could potentially be applied to Chinese scenarios. The core rationales underlying the sub-optimal performance of LLaMA2 on Chinese can be attributed to two parts, where the first part is that LLaMA2 is not trained on many Chinese tokens and the second part is that LLaMA2 is not trained with adequate Chinese vocabulary. While the former one could be counted by adaptation, the latter should be handled by expanding the original vocabulary with a Chinese vocabulary.

At the time of formulating this work, it is the first option to do so since there are not any available bilingual language models with better performance than LLaMA2. And nowadays, many great open-source bilingual language models (e.g., Baichuan2, Yang et al., 2023; Qwen, Bai et al., 2023) have emerged that can be utilized without the necessity of adaptation.

Table 5: The token per byte compression rates. The results are measured on some English and Chinese Wikipedia articles.

LM	Vocabulary	Token/Byte Compression Rate
LLaMA2	32,000	46.2%
MiniMA	49,216	35.0%

Thereby, before the adaptation, we append many frequently used Chinese characters and phrases in SentencePiece (Kudo and Richardson, 2018) so that the original encoding and decoding behaviors of English are not changed while those of Chinese are improved. In doing so, we expand the vocabulary of LLaMA2 from 32,000 to 49,216 with much better Chinese encoding and decoding performance in terms of token per byte compression rate as in Table 5. With the purpose of making LLaMA2 familiar with more Chinese tokens including those appended yet not learned elements in the new vocabulary, we should necessarily adapt the expanded LLaMA2 on curated data as in Table 1. And after the adaptation, the English perplexity is not declined that much while the Chinese perplexity is also considerable as in Table 6. The discrepancy in Chinese perplexity is perhaps due to that LLaMA2 may fallback to bytes for unknown Chinese characters and adapted LLaMA2 is still under-trained.

The data for adaptation is packed into sequences, and each of which is of 4,096 tokens in alignment with that of LLaMA2. We take a batch size of ~4M tokens (or say 1,024 batches), a learning rate of 3e-5, a weight decay of 1e-1. The training lasts for 1 epoch, and the learning rate is scheduled to warm up linearly for the first 1% steps of all and decay down in a sinusoidal way for the rest steps. Gradients whose norms accumulate over 1 are necessarily clipped. The training is executed on 16 A100 80G GPUs. The training efficiency is sufficiently guaranteed by DeepSpeed Zero3 this time and Flash Attention. Gradient checkpointing is enabled to further reduce the memory footprint and bfloat16 precision is chosen to improve the training stability.

Table 6: The perplexities. The results are measured on some English and Chinese Wikipedia articles.

LM	English	Chinese
LLaMA2-7B	3.74	3.96
adapted LLaMA2-7B	3.98	13.91

Table 7: The statistics of data mixture used for finetuning.

Dataset	Language	Examples	Proportion
Alpaca	En	52,002	4.7%
Alpaca-Chinese	Zh	48,818	4.4%
CodeAlpaca	En & Co	20,022	1.8%
GPTeacher-Instruct	En	18,194	1.7%
GPTeacher-RolePlay	En	1,923	0.2%
GPTeacher-CodeGen	En & Co	4,535	0.4%
Baize-Quora	En	54,456	4.9%
Baize-StackOverflow	En	57,046	5.2%
UnnaturalInstruction	En	9,000	0.8%
Flan-CoT	En	74,771	6.8%
Flan-CoT-Chinese	Zh	74,771	6.8%
COIG-LeetCode	Zh & Co	11,737	1.1%
BELLE-Math	Zh	49,696	4.5%
ShareGPT/Discord	En	197,893	18.0%
MiniChat-Zhihu	Zh	129,981	11.8%
MiniChat-Wiki	Zh	234,291	21.3%
MiniChat-Math	Zh	39,336	3.6%
MiniChat-Life	Zh	22,697	2.1%
Mixture	–	1,101,169	100.0%

B Statistics of Instruction-following Data

The statistics are presented in Table 7, accommodating data gathered from different languages and categories, such as Alpaca (Taori et al., 2023), CodeAlpaca (Chaudhary, 2023), GPTeacher (Teknium, 2023), Baize (Xu et al., 2023b), Unnatural Instruction (Honovich et al., 2023), Flan-CoT (Chung et al., 2022), COIG (Zhang et al., 2023d), and ShareGPT.

In regard to lacking Chinese data, we additionally incorporate data generated from ChatGPT following the *self-chat* technique (Xu et al., 2023b), which necessarily provides seed topics to ChatGPT and has it generate chat data fluently. The seed topics in Chinese we use range from Zhihu trending questions and Wikipedia entries to Math questions and Life news.

C Pruning of MiniMA

The motivation of using a heuristic pruning priority for MiniMA is that asymmetric shapes across layers need quite a few monkey-patches to widely used libraries (e.g., HuggingFace Transformers, Wolf et al., 2019) in spite of their slightly superior performance. In contrast, symmetric shaping naturally fits in these libraries, thus flexible for widespread use.

And the reasons pertain behind the design choice for symmetric shaping are as follows. The local pruning priority is basically required by the symmetric shaping. However, the global pruning priority can lead to entire-layer pruning yet the local one cannot. So we introduce a heuristic rule here that distributes a portion of the sparsity to layer dropping and consequently reduces the sparsity of hidden states, MHA heads, and FFN neurons.

Figure 8: The pruning ratios across layers of LLaMA2-7B along the rise of sparsity. The variations of pruning ratios of MHA heads and FFN neurons are listed separately. Peaks and valleys are obvious to see, and peaks stand at the most bottom and top layers.

As for justification on why evenly dropping the most bottom and top layers rather than other layers, we uncover that the global pruning priority tends to prune the most bottom and top MHA heads and FFN neurons as in Figure 8 even with the mentioned ℓ_2 norm protection from skewed pruning ratios (Molchanov et al., 2017). This suggests that the most bottom and top layers of LLaMA are confidently of least significance among all layers, encouraging us to prune them in the first place. And the even distribution is just an approximation.

D Evaluation Protocols of Standard Benchmarks

MMLU (Hendrycks et al., 2021) is an multiple-choice dataset covering multiple disciplines. CEval (Huang et al., 2023) is also a multiple-choice dataset covering multiple disciplines however in Chinese, which could be viewed as a Chinese MMLU. DROP (Dua et al., 2019) is a reading comprehension dataset requiring discrete reasoning. BBH (Suzgun et al., 2023) is the challenging counterpart of BIG-Bench (Srivastava et al., 2022), which is essentially a dataset measuring the reasoning abilities of LMs. GSM8K (Cobbe et al., 2021) is a grade-school level math problem-solving dataset. HumanEval (Chen et al., 2021) is a coding problem-solving dataset. By category, MMLU and CEval are taken to evaluate the knowledge of LMs, DROP, BBH, GSM8K are taken to evaluate the reasoning of LMs, and HumanEval is taken to evaluate the coding of LMs.

Following the evaluation guidelines of LLaMA (Touvron et al., 2023a,b), we devise evaluation protocols of considered datasets as follows:

- MMLU: 5-shot direct prompting performance evaluated by accuracy.
- CEval: 5-shot direct prompting performance evaluated by accuracy.
- DROP: 3-shot direct prompting performance evaluated by exact match score.
- BBH: 3-shot direct prompting performance evaluated by exact match score.
- GSM8K: 8-shot chain-of-thought prompting performance evaluated by maj1@1.
- HumanEval: 0-shot prompting performance evaluated by pass@1.

Here, accuracy and extach match score are two commonly seen metrics. maj1@k is a specialized metric that gives a positive judgement if there is at least 1 correct answer among k candidate answers output by LMs. pass@1 is also a specialized one that shares similar idea with maj1@k yet with a much more complex formulation.

Table 8: The results against existing LMs on standard benchmarks.

LM	Tokens	MMLU Acc	CEval Acc	DROP EM Score	BBH EM Score	GSM8K Maj1@1	HumanEval Pass@1
LLaMA-7B (2023a)	1,000 B	35.10	28.00	27.46	30.93	9.17	10.37
LLaMA2-7B (2023b)	2,000 B	46.00	34.40	31.57	32.02	14.10	12.80
Baichuan-7B (2023)	1,200 B	42.60	43.50	19.82	31.94	8.57	7.93
Baichuan2-7B (2023)	2,600 B	54.31	55.27	25.97	35.21	13.19	16.46
Mistral-7B (2023)	–	62.67	45.91	46.59	43.88	41.02	28.05
Mamba-2.8B (2023)	300 B	25.58	24.74	15.72	29.37	3.49	7.32
ShearedLLaMA-2.7B (2023)	50 B	26.97	22.88	19.98	30.48	3.56	4.88
CerebrasGPT-2.7B (2023a)	53 B	24.66	23.18	11.46	29.32	2.43	3.66
OPT-2.7B (2022b)	180 B	26.02	24.52	13.70	28.71	1.90	0.00
BLOOM-3B (2022)	341 B	26.60	23.77	14.32	29.84	2.04	0.61
Pythia-2.8B (2023)	300 B	26.28	23.11	16.04	29.30	2.73	5.49
OpenLLaMA-3B (2023)	1,000 B	26.70	26.30	20.14	30.56	3.11	0.00
OpenLLaMAv2-3B (2023)	1,000 B	26.36	25.41	18.19	30.45	4.62	7.93
BTLM-3B (2023b)	627 B	27.20	26.00	17.84	30.87	4.55	10.98
StableLM-3B (2023)	4,000 B	44.75	31.05	22.35	32.59	10.99	15.85
Mamba-1.4B (2023)	300 B	25.97	25.85	12.36	29.46	1.67	6.71
ShearedLLaMA-1.3B (2023)	50 B	25.72	24.22	14.53	29.22	3.03	1.83
CerebrasGPT-1.3B (2023a)	26 B	26.50	23.03	8.54	29.11	2.58	2.44
OPT-1.3B (2022b)	180 B	26.66	25.71	14.38	28.77	3.18	0.00
BLOOM-1.1B (2022)	341 B	27.33	26.75	8.62	28.42	2.43	0.00
Pythia-1.4B (2023)	300 B	25.20	26.08	12.22	28.87	1.90	5.49
TinyLLaMA-1.1B (2023e)	2,500 B	25.84	25.19	15.42	29.35	2.12	7.93
Falcon-1B (2023)	350 B	25.52	25.93	15.94	27.71	1.67	0.00
Phi1.5-1.3B (2023)	150 B	42.86	26.60	14.94	30.84	28.43	32.93
Qwen-1.8B (2023)	2,200 B	44.05	54.75	12.97	30.80	22.97	14.02
MiniMA	126 B	28.51	28.23	22.50	31.61	8.11	10.98

We report the results on validation sets of CEval and DROP due to the unavailability of their test sets and the results on test sets on the others.

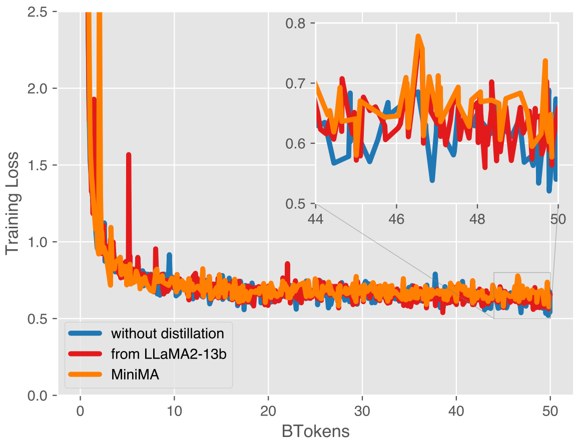


Figure 9: The training losses of MiniMA versus its variants.

E Training Loss of MiniMA

Training loss is usually taken as a sort of criterion of convergence and can signal the performance of a LM in pretraining as in Touvron et al. (2023a). So we conduct training loss-level comparisons among MiniMA and its variants.

However, the signal is not valid anymore in distillation since MiniMA and its variants have very similar training losses as in Figure 9. Direct training has a slightly lower training loss than distillation from LLaMA2-13B, and MiniMA actually owns a slightly higher training loss than distillation from LLaMA2-13B does. We conjecture that distillation loss (alignment to distributed probabilities) is inherently more difficult to optimize than direct training loss (alignment to one-hot labels). In literature (Pereyra et al., 2017), larger models would confidently produce distributed probabilities that are more akin to one-hot labels. So it is no wonder even distillation from LLaMA2-13B can has a lower training loss than MiniMA.

F Benchmarking Results against Existing LMs

We include the results of existing LMs here.

The results in Table 8 present that MiniMA can with no doubt overshadow a large part of existing LMs. Reversely, some LMs beat other scale-matched LMs and MiniMA by substantial margins on some datasets thanks to neat data quality or huge data quantity, which initiates additional expenses. Some results on other related standard benchmarks are detailed in Appendix G.

G More Standard Benchmarks

We additionally incorporate results on other tasks for a broader comparison including ARC (Clark et al., 2018), Winogrande (Sakaguchi et al., 2020), TruthfulQA (Lin et al., 2022). The performance on these tasks is evaluated by accuracy with 25-shot direct prompting, by accuracy with 5-shot direct prompting, and by mc2 with 0-shot direct prompting, respectively.

Table 9: The results on other standard benchmarks.

LM	ARC Acc	Winogrande Acc	TruthfulQA MC2
OpenLLaMAv2-3B (2023)	40.27	67.01	34.78
ShearedLLaMA-2.7B (2023)	41.72	67.01	37.32
MiniMA	43.43	65.98	39.76

The results in Table 9 shows an akin trend as that of Table 2, likewise indicating the superiority of MiniMA.

H Estimates of Training Compute

The training compute estimation is quoted from Bahdanau (2022), which explains the compute constitutions in an organized manner. Plus to that, this estimate is also mentioned and used in Kaplan et al. (2020) as:

$$C = \underbrace{6 \cdot N \cdot D}_{\text{total compute}} = \underbrace{2 \cdot N \cdot D}_{\text{forward compute}} + \underbrace{4 \cdot N \cdot D}_{\text{backward compute}}, \tag{7}$$

where C is the training compute, N is the number of model parameters, and D the number of consumed tokens. And the total compute can be decomposed to forward and backward parts.

Particularly for distillation that additionally engages the inference compute of the teacher, the training computed should be expanded as:

$$C^{\text{distillation}} = \underbrace{6 \cdot N^{\text{student}} \cdot D}_{\text{student compute}} + \underbrace{2 \cdot N^{\text{teacher}} \cdot D}_{\text{teacher compute}}, \tag{8}$$

where N^{student} and N^{teacher} indicate the numbers of teacher and student model parameters, respectively.

So here we attach the training compute of considered LMs in Table 10, with necessary reference sources.

Table 10: The estimates of trainig compute of considered LMs.

LM	Params	Tokens	Compute
LLaMA-7B	7 B	1,000 B	42.0×10^9 TFLOPs
LLaMA2-7B	7 B	2,000 B	84.0×10^9 TFLOPs
Baichuan-7B	7 B	1,200 B	50.4×10^9 TFLOPs
Baichuan2-7B	7 B	2,600 B	109.2×10^9 TFLOPs
Mistral-7B ①	7 B	4,000 B	168.0×10^9 TFLOPs
Mixtral-8x7B ①	47 B	4,000 B	336.0×10^9 TFLOPs
Mamba-2.8B ②	2.8 B	300 B	4.6×10^9 TFLOPs
ShearedLLaMA-2.7B	2.7 B	50 B	0.8×10^9 TFLOPs
CerebrasGPT-2.7B	2.7 B	53 B	0.9×10^9 TFLOPs
OPT-2.7B	2.7 B	180 B	2.9×10^9 TFLOPs
BLOOM-3B	3 B	341 B	6.1×10^9 TFLOPs
Pythia-2.8B	2.8 B	300 B	5.0×10^9 TFLOPs
OpenLLaMA-3B	3 B	1,000 B	18.0×10^9 TFLOPs
OpenLLaMAv2-3B	3 B	1,000 B	18.0×10^9 TFLOPs
BTLM-3B	3 B	627 B	11.2×10^9 TFLOPs
StableLM-3B	3 B	4,000 B	72.0×10^9 TFLOPs
Phi2-2.8B ④	2.8 B	1,400 B	159.9×10^9 TFLOPs
Mamba-1.4B ②	1.4 B	300 B	2.3×10^9 TFLOPs
ShearedLLaMA-1.3B	1.3 B	50 B	0.4×10^9 TFLOPs
CerebrasGPT-1.3B	1.3 B	26 B	0.2×10^9 TFLOPs
OPT-1.3B	1.3 B	180 B	1.0×10^9 TFLOPs
BLOOM-1.1B	1.1 B	341 B	2.3×10^9 TFLOPs
Pythia-1.4B	1.4 B	300 B	2.5×10^9 TFLOPs
TinyLLaMA-1.1B ③	1.3 B	2,500 B	16.5×10^9 TFLOPs
Falcon-1B	1 B	350 B	2.1×10^9 TFLOPs
Phi1.5-1.3B ④	1.3 B	150 B	17.5×10^9 TFLOPs
Qwen-1.8B	1.8 B	2,200 B	23.8×10^9 TFLOPs
MiniMA ⑤	3 B	126 B	4.0×10^9 TFLOPs

① Mistral is rumored to be trained on 8,000 B tokens, and we here choose 4,000 for estimation. Mixtral is an MoE model, in which 2 out of 8 experts are effective.
② Mamba is a state space model benefiting for long-context inference, and its compute is estimated from the original paper.
③ TinyLLaMA is planned to be on 3,000 B tokens, but as of December 11st it has been trained on 2,500 B tokens.
④ Phi is trained on tokens generated or filtered by GPT4, so the number of teacher model parameters is estimated according to the rumor that GPT4 is a 2-in-16 MoE model with 1.8 trillion parameters.
⑤ MiniMA is distilled from a 7 B teacher.

I Evaluation Protocols of GPT4 Assessments

Vicuna-Bench (Chiang et al., 2023) is an instruction-following dataset encompassing 80 instructions from 9 fields. And BELLE-Bench is a Chinese counterpart of Vicuna-Bench comprised of 1,000 instructions from 10 fields. Note that both of them only examine single-turn capability, and we could incorporate multi-turn judgements in the future (e.g., MT-Bench, Zheng et al., 2023)

In commonsense, it is more convincing to grade responses via human labor. However, regarding the labor intensity, we instead derive an evaluation protocol using GPT4 following their recommendations. Concretely, given a pair of responses

Table 11: The results of MiniChat on standard benchmarks.

LM	MMLU Acc	CEval Acc	DROP EM Score	BBH EM Score	GSM8K Maj1@1	HumanEval Pass@1
OpenLLaMA-3B (2023)	26.70	26.30	20.14	30.56	3.11	0.00
OpenBuddy-3B (2023)	23.88	24.67	15.89	29.08	11.07	3.66
Δ	-2.82	-1.63	-4.25	-1.48	+7.96	+3.66
MiniMA	28.51	28.23	22.50	31.61	8.11	10.98
MiniChat	38.40	36.48	22.58	31.36	29.72	18.29
Δ	+9.89	+8.25	+0.08	-0.25	+21.61	+7.31

from two LMs, GPT4 is instructed to score each of them in a head-to-head manner. And different instructions are offered to GPT4 for distinguished considerations of general, coding, and math questions. Please refer to the instructions in Chiang et al. (2023). Macro average scores are further computed as summaries. Besides the pairwise scores, we also deduce win-lose-tie proportions correspondingly from them.

J Benchmarking Results of MiniChat

We include the results of MiniChat on standard benchmarks in case of potential curiosities, since it is long suspected that instruction data could boost benchmarking performance.

The results in Table 11 indicate that proper finetuning could indeed promote the benchmarking performance to certain degrees. In spite of this, it is not very clear whether the improvements are resulted by data contamination or not.

K Field-specific Scores in GPT4 Assessments

We also provide field-specific scores of regarded baselines and MiniChat for possible interests of rigorous studies.

The results in Figure 10 again show that MiniChat is cost-effective across fields.

Table 12: The results of MiniChat on human assessments. Elo ratings are reported.

LM	Vicuna-Bench Elo Rating	Human Rank	GPT4 Rank
MiniChat	1013.9	1	2
Phoenix-7B	1010.8	2	4
ChatGLM2-6B	1004.2	3	3
BiLLa-7B	996.0	4	1
OpenBuddy-3B	992.0	5	6
ChatGLM-6B	983.2	6	5

L Human Assessments

It is for a long time doubted that whether GPT4 is a convincing referee in response assessment. On that account, we conduct a fast human assessment to verify the consistency between GPT4 and human in judging the quality of model responses. As it is too expensive to ask humans to evaluate these responses in a head-to-head style, we learn from Zheng et al. (2023) and instead adopt an elo rating-based arena. The arena quickly reduces the evaluation complexity from $\mathcal{O}$(#models$\times$#questions) to $\mathcal{O}$(#questions). To further simplify the arena, we attain the models responses to questions in the dataset in advance. During the whole process, model identities are anonymized from humans. Multiple experienced researchers are recruited to mitigate human biases.

The results in Table 12 inform that the rank disparity between GPT4 and human is computed as a spearman correlation between two rank permutations, i.e., 0.54, so the consistency is roughly 54%, which is not very high.

To deliver more intuitions on how MiniChat performs, we select a few representative cases here to display.

The cases in Appendix M picture that MiniChat is a good problem-solver.

M Comparative Cases

In Table 13, we supply two cases that are separately concerned about coding and writing and attach them with responses from OpenBuddy-3B, ChatGLM2-6B, and MiniChat for better comparisons.

In the coding case, we find that: OpenBuddy-3B follows the problem requirements but fails to solve the problem; ChatGLM2-6B fails to follow the problem requirements that extra data structures should not be used, and also fails to implement a correct hash table class; MiniChat gives a neat solution.

In the writing case, we find that: OpenBuddy-3B gives an overly general blog mentioning only two items ; ChatGLM2-6B mentions a few more respectively for cultural experiences and must-see attractions, however, it formats the blog very much like a scientific paper rather than a blog; on the contrary, MINICHAT seems to have more elegant language but uses very interleaved logistics.

N MiniMA Family

The details should be left to their model cards on the HuggingFace, since they are marginally related to this paper.

- MINIMA-2-3B. This is a model continued from MINIMA with a more diverse mixture of data.
- MiniChat-2-3B. This is an instruction-following model trained from MINIMA-2-3B.
- MiniLoong-3B. This is a long-context model extended from MINIMA-2-3B with the aid of distributed attention (Jacobs et al., 2023).
- MiniMix-2/4x3B. This is a mixture-of-experts model upcycled from MINIMA-2-3B (Komatsuzaki et al., 2023).

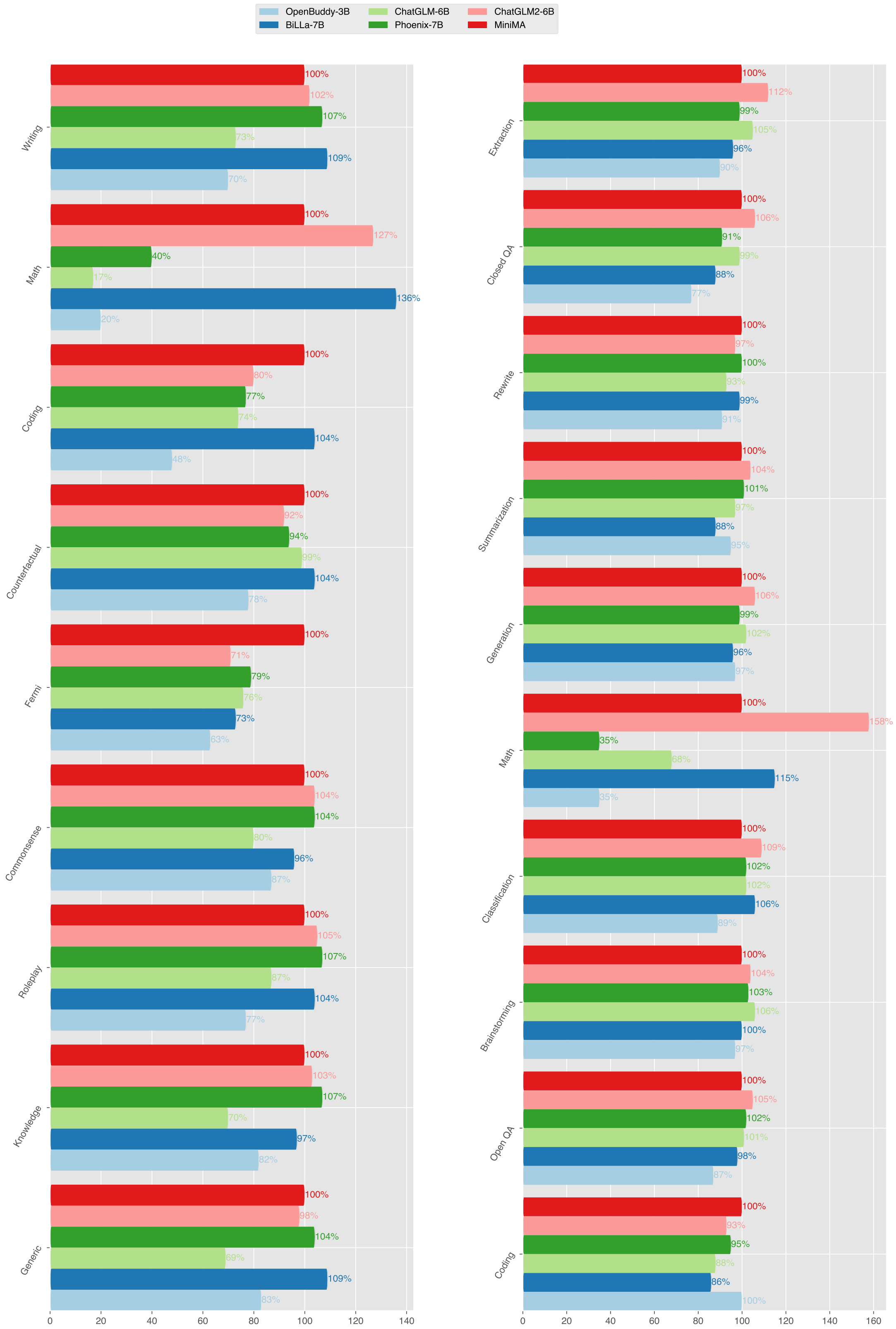


Figure 10: The field-specific results of MINICHAT on GPT4 assessments. The scores are rebased to relative percents with respect to those of MINICHAT.

Table 13: The comparative cases.

Question: Implement a program to find the common elements in two arrays without using any extra data structures.

OpenBuddy-3B: To find the common elements in two arrays without using any extra data structures, you can use the following algorithm:

1. Sort the two arrays in ascending order (using an appropriate sorting function).
2. Compare the first element of each array. If it's the same, then the rest of the array is common.
3. If the first element is different, then the rest of the array is common.
4. Repeat steps 2 and 3 until either all the elements in the first array are different or all the elements in the second array are different.

Here is the Python code for the algorithm:

```
def find_common_elements(arr1, arr2):
    sorted_arr1 = sorted(arr1)
    sorted_arr2 = sorted(arr2)
    for i in range(len(sorted_arr1)):
        if sorted_arr1[i] == sorted_arr2[i]:
            return sorted_arr1
    return []
```

This algorithm can be used to find the common elements in any two arrays.

ChatGLM2-6B: One way to implement a program to find the common elements in two arrays without using any extra data structures is to use a hash table. A hash table is a data structure that stores key-value pairs and allows for fast lookup, insertion, and deletion of elements.

To use a hash table to find the common elements in two arrays, you would first need to create a hash table with a key as the index and a value as the key-value pair. Then, for each element in the first array, you would use the hash table to look up the element and compare it to the elements in the second array. If a match is found, the element and its index would be added to the hash table.

Here is a simple example of how this could be implemented in Python:

```
class Element:
    def __init__(self, value):
        self.value = value

class HashTable:
    def __init__(self):
        self.table = {}

    def put(self, key, value):
        self.table[key] = value

    def get(self, key):
        return self.table.get(key)

    def contains(self, key):
        return self.table.contains(key)

# Example usage
element1 = Element('a')
element2 = Element('b')
table = HashTable()
table.put('a', element1)
table.put('b', element2)
print(table.get('a')) # prints the Element object for 'a'
print(table.contains('a')) # prints True
print(table.get('z')) # raises a KeyError
```

Note that this approach has a time complexity of O(n * m), where n and m are the lengths of the arrays, which may be slow for large arrays. There are other data structures that could be used to implement this, such as a linked list or a tree-based data structure, but a hash table is a simple and efficient solution for this problem.

MINICHAT: Here is a Python implementation:

```
def common_elements(arr1, arr2):
    if len(arr1) == 0:
        return []
    if len(arr2) == 0:
        return arr1
    common_elements = []
    for element in arr1:
        if element in arr2:
            common_elements.append(element)
    return common_elements
```

Question: Compose an engaging travel blog post about a recent trip to Hawaii, highlighting cultural experiences and must-see attractions.

OpenBuddy-3B: Hi there! I am excited to share with you my recent trip to Hawaii and all the amazing cultural experiences I had during my stay. From the stunning beaches to the vibrant culture, I fell in love with this beautiful island state.

First of all, I must say that I was blown away by the incredible natural beauty of Hawaii. The lush green landscapes, crystal-clear waters, and stunning sandy beaches all left a lasting impression on me. My favorite part of my trip was kayaking through the beautiful Wailea coastline, where I saw dolphins, seahorses, and even a couple of whale sharks swimming in the ocean. It was such a surreal and unforgettable experience that I will always remember.

Another highlight of my trip was visiting the Polynesian cultural center, where I experienced traditional Polynesian dance and music. I was fascinated by the intricate and complex dances that the performers were performing, and I even managed to join in on a few of the demonstrations. It was an incredible moment that I will never forget.

Furthermore, I had the opportunity to experience the vibrant cultural lifestyle of Hawaii. From attending traditional Hawaiian festivals to visiting local art galleries, I felt like I was truly immersed in the local culture. I even had the chance to try some delicious Hawaiian cuisine, such as poke bowls and plate lunches, which were some of the most delicious dishes I have ever tasted.

Finally, I have to say that my trip to Hawaii was nothing short of incredible. The stunning beaches, warm weather, and vibrant culture made for the perfect vacation. I can't wait to return to this beautiful island state and explore even more of its rich cultural heritage.

If you're planning a trip to Hawaii, I highly recommend taking the time to experience the local culture and attractions. It will be an unforgettable experience that you won't want to miss.

ChatGLM2-6B: Title: A Journey Through the Aloha State: Exploring Hawaii's Cultural Experiences and Must-See Attractions

Introduction
Hawaii, a state known for its lush greenery, stunning landscapes, and tropical climate, is a paradise for many. But there's more to Hawaii than what meets the eye. This recent trip to the Aloha State offered me a unique opportunity to immerse myself in the local culture and traditions. As I discovered, Hawaii is a place where one can truly feel at home and experience the perfect blend of relaxation, adventure, and culture.

Cultural Experiences
Hawaii is a place where culture and tradition are deeply ingrained in the very fabric of the island. I had the chance to experience this first-hand by attending a traditional Hawaiian luau, a dinner party that is a staple of the Aloha State. The luau was a fascinating event that included a host of local dishes, music, and dance. I was amazed by the energy and excitement of the party, and I felt truly at home in the midst of the Aloha State.

Another way to experience the local culture in Hawaii is by visiting the Pu'uhonua o Honaunau National Historical Park. This park is home to a collection of ancient Hawaiian temples, and it offers a glimpse into the island's spiritual traditions. I found it fascinating to learn about the history and significance of these temples, and I appreciated the opportunity to reflect on the culture and traditions of Hawaii.

Must-See Attractions
Of course, any trip to Hawaii is not complete without visiting some of the state's many natural attractions. I had the chance to snorkel in the crystal-clear waters of the Polynesian Islands, and I was amazed by the beauty of the coral reefs and the diverse array of marine life. I also took the opportunity to visit the Waikato Waterfall, a natural wonder that is a must-see attraction in Hawaii.

Another must-see attraction in Hawaii is the Diamond Head, a volcanic crater that offers stunning views of the surrounding landscape. I took the hike up to the top of the crater and was rewarded with breathtaking views of Honolulu and the ocean below.

Conclusion
Hawaii is a place where culture, tradition, and natural beauty are all around. I was truly touched by the Aloha State and its unique way of life. I hope you'll have the chance to experience it for yourself and discover the beauty and magic of Hawaii.

MINICHAT: Visiting Hawaii: A Journey Through the Ocean's Heart

As I step off the plane, I am immediately greeted by the warm, inviting sights and sounds of Hawaii. The colors of the sunset are breathtaking, and the sound of the waves is soothing. I feel as though I am in a different world, surrounded by the beauty of nature.

My first stop is Waikiki Beach, where I take a leisurely stroll along the sandy shoreline, feeling the softness of the golden sand between my toes. The ocean is crystal clear, with gentle waves gently lapping at the shore. I stop for a moment to watch the surfers as they ride the waves and catch the sun.

After soaking up the sun, I head to Pearl Harbor to pay my respects at the USS Arizona Memorial at the National Memorial Cemetery of the Pacific. As I walk through the exhibits, I am reminded of the sacrifices that were made during World War II. The USS Arizona Museum is a moving tribute to the bravery and sacrifices of the sailors and soldiers who gave their lives for their country.

Next, I make my way to Hanauma Bay, where I go scuba diving in the crystal-clear waters. The visibility is excellent, and the coral reefs are a sight to behold. The fish are vibrant, and the colors are so vivid that it's like a dream.

After an exhilarating day of exploring, I head to the historic town of Hilo to experience the local culture. The Hilo Museum is a great place to learn about the history of the area, and the Hilo Bay Fishpond is a fascinating example of traditional Hawaiian aquaculture.

My trip to Hawaii has been an unforgettable experience. Whether it be the stunning natural beauty or the rich cultural heritage, there is something for everyone in this incredible destination. I am already planning my next visit, and I know that I will be back to explore even more of what Hawaii has to offer.
