360ZHINAO TECHNICAL REPORT

360Zhinao Team
g-zhinao-opensource@360.cn

ABSTRACT

We present 360Zhinao models with 7B parameter size and context lengths spanning 4K, 32K and 360K, all available at https://github.com/Qihoo360/360zhinao. For rapid development in pretraining, we establish a stable and sensitive ablation environment to evaluate and compare experiment runs with minimal model size. Under such guidance, we perfect our data cleaning and composition strategies to pretrain 360Zhinao-7B-Base on 3.4T tokens. We also mainly emphasize data during alignment, where we strive to balance quantity and quality with filtering and reformatting. With tailored data, 360Zhinao-7B's context window is easily extended to 32K and 360K. RMs and RLHF are trained following SFT and credibly applied to specific tasks. All together these contributions lead to 360Zhinao-7B's competitive performance among models of similar size.

1 INTRODUCTION

In recent years, the field of natural language processing (NLP) has witnessed a profound transformation, fueled by the advent of large language models (LLMs) (Bubeck et al., 2023; Touvron et al., 2023a; OpenAI, 2023), which have emerged as a cornerstone to revolutionize the way we understand and generate human language. LLMs represent a new paradigm in artificial intelligence (AI) research, characterized by their immense scale, complexity, and versatility (Zhao et al., 2023). Those models, typically built upon advanced neural network architectures like Transformers, are trained on vast amounts of text data, encompassing billions or even trillions of words. The extensive training endows LLMs with a deep understanding of linguistic structures, nuances, and context, enabling them to generate human-like text and perform a myriad of NLP tasks with unprecedented accuracy and fluency (Yang et al., 2024).

Despite the impressive capabilities of LLMs, training an LLM from scratch still struggles with several challenges. The training journey can be divided into two stages: the pretraining stage and the alignment stage (Zhang et al., 2023). The pretraining stage involves the model learning on large-scale textual data to build its foundational knowledge and language comprehension. However, two obstacles stick out in the pretraining stage (Zhao et al., 2023). First, refining the training corpus to enhance the base model's performance is paramount given the enormity of pretraining data. While extensive research has delved into data cleaning and sampling methodologies (Soldaini et al., 2024; Penedo et al., 2023; Wenzek et al., 2019; Gunasekar et al., 2023), the sheer scale and intricacy of pretraining datasets still leave ample room for elevating informational density and efficiency. Second, establishing a stable and sensitive ablation environment for accurately assessing data strategies poses another challenge (Chang et al., 2024; Zhou et al., 2023). The widely used Opencompass (OpenCompass-Contributors, 2023) framework proves inconvenient for data strategy explorations. It is inherently unstable and insensitive to smaller models or datasets (Wei et al., 2023), and further, it lacks correlation with downstream skills to adequately evaluate the model. Addressing these two challenges will help propel further research and practices in the pretraining stage.

In the alignment stage, challenges arise regarding data (Albalak et al., 2024), long context (Kamradt, 2023) and RLHF effectiveness (Wang et al., 2024; Xu et al., 2024). SFT gets major parts of things done, but is to some extent sensitive to data quality and composition (Liu et al., 2024b). It is tricky to balance the learning of different prompt categories and specific application data. Various useful applications of LLMs and multi-modal large models require sequence lengths far beyond several thousand (Gemini-Team et al., 2024). Pretraining on large corpus of long data (context length of tens or hundreds of thousand) turns out inefficient and one desideratum is to extend the context length with minimal continual pretraining and SFT at reasonable costs. RLHF in open-sourced

models has not yet fulfilled the presumed promise as OpenAI (2023). Much remains under-explored in terms of RM data, RM training, PPO data and PPO training, etc.

In response to those challenges, we devoted substantial efforts to our LLM models, the 360Zhinao series, and presented the details in this technical report. The 360Zhinao model comprises a base model trained from scratch and a chat model using alignment techniques. In the pretraining stage, we explored the data strategies and their evaluation. First, we built a data cleaning and filtering pipeline. By crawling massive web pages, we employed a series of filtering and cleaning steps. Subsequently, we explored multi-level deduplication and data mixing strategies, ultimately obtaining a corpus of 3.4 TB tokens with high data efficiency. Second, to validate the effectiveness of our data strategies, we constructed a stable and sensitive ablation environment, establishing a set of custom benchmarks associated with downstream skills. Based on this environment, we reported the ablation results of data strategies, effectively guiding the direction of data iterations.

Emphasis is also laid on data in the alignment stage. Initially we prioritized quantity over quality and later the opposite. We now grow SFT data at a more steady pace with higher standards. We explored different ways of context window extension and have converged to the simple method of RoPE-base change with tailored data. We stabilized RLHF training with data and codebase improvement and have applied successfully to specific tasks. The resulting RM has also been used in other data selection and judgement pipelines.

Due to the complexity of LLMs, some crucial intermediate information on model production is absent in existing literature. In this report, we made efforts to break barriers and hope to offer new perspectives to researchers. We have released the implementations of the base- and chat-models to the open-source community, aiming to enhance the openness and transparency of LLMs and facilitate collaboration and reproducibility. Based on these efforts, the contributions of the release of the 360Zhinao models can be summarized as follows:

- In the pretraining stage, we provide detailed insights into the iteration of data recipes and the construction of an effective ablation experimental environment.
- In the alignment stage, we present our data-centric approach to data improvement, our converged approach to context window extension and positive findings from RLHF.

2 PRETRAINING

Pretraining serves as the foundational stage of developing LLMs, providing them with the capability to learn general linguistic patterns from vast text corpora. This section explores a comprehensive ablation study of data recipes, including cleaning and deduplication methods. We present the model training, as well as evaluation results on benchmarks.

2.1 DATA

The publicly accessible web pages serve as one of the primary components of our corpora, including billions of various pages. We have built a data recipe pipeline and performed various meticulous data strategies. The pipeline can be divided into four stages: 1) Data preparation; 2) Data cleaning; 3) Data deduplication and 4) Data mixture. The cascaded changes in data size in the pipeline are shown in Figure 1. To verify the effectiveness of the data strategy, we constructed a sensitive evaluation environment and conducted a detailed ablation analysis.

2.1.1 DATA PREPARATION

Faced with such a large scale of web pages, we need to choose the appropriate documents for data cleaning. The pipeline for data preparation includes 1) Url and keyword filtering; 2) Exact deduplication and 3) Language filtering. All actions performed in this pipeline are document-level filtering, removing 12.66% of the data.

Url and keyword filter The uneven quality of websites necessitates that web page filtering must be strategic. We use a combination of black-and-white lists for website filtering. The blacklist mainly records pornographic, violent, as well as cheating, and dead links. The whitelist records
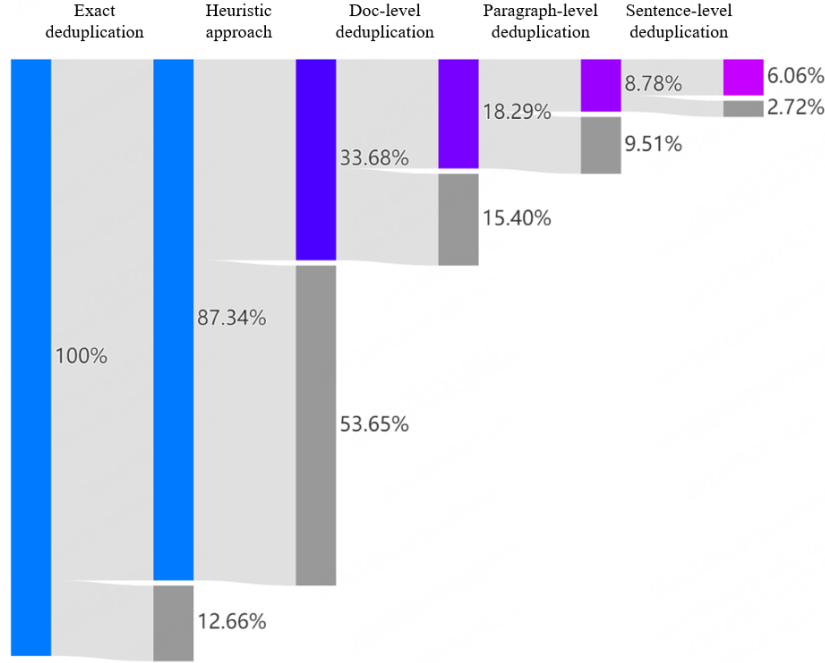
Figure 1: The cascaded retention changes in the web page pipeline.

high-scoring sites that have been manually reviewed. We prioritize scanning whitelisted sites and discarding pages from blacklisted sites. For the remaining pages, we will calculate the website rating based on meta information such as the website's registration status, authority, and historical performance, and select highly trusted web pages according to the rating. In addition, a sensitive keyword library has also been constructed for further filtering of web pages.

Exact deduplication   The selected web page may contain potential duplicate pages. Url deduplication and document body deduplication have been adopted into the pipeline. Url deduplication can prevent duplicate crawling and parsing of web pages. Document body deduplication can reduce data redundancy in the dataset.

Language filter   Language filtering aims to retain the desired target language web pages. We utilized a FastText model (Bojanowski et al., 2017; Wenzek et al., 2019) for identifying the language of web pages. Specifically, an English web page would be filtered out when containing Chinese characters.

2.1.2   DATA CLEANING

The goal of data cleaning is to extract the main content of web pages and obtain high-quality text. The cleaning pipeline integrates heuristic rules and models, including junk text cleaning, content quality filtering, and personally identifiable information (PII) cleaning. Through those three stages of cleaning, we have significantly improved content quality and security, removing a total of 53.65% of the text.

Junk text filter   Generally, the content extracted from structured web pages still contains a large amount of junk text, such as web page navigation, copyright notices, related content recommendations, and advertisements. Junk text is all over the place, following similar templates, and lacking diversity, which can hinder the effectiveness and efficiency of model training if not deleted cleanly. We adopted a pipeline for content cleaning that includes paragraph granularity and document granularity, using heuristic rules and classification models. We pursue the accuracy of deleting the junk text without compromising the coherence of the content during the cleaning pipeline.

Paragraph granularity cleaning aims to preserve the main content of web pages while removing irrelevant junk text. We trained a FastText model to identify whether a given paragraph is junk. This model employs a semi-supervised learning approach based on incremental learning, with training data reaching millions in scale after multiple iterations. The F1-score of the junk text model reached 99%. Given a document, all its paragraphs undergo prediction by this model and then whether a paragraph is discarded depends on the status sequence of all paragraphs.

Document granularity cleaning is performed after paragraph subsequently cleaning to remove documents lacking information or containing formatting issues. Heuristic rules based on frequency and ratio, inspired by Dolma's practice, serve as the primary cleaning methods (Soldaini et al., 2024; Chen et al., 2023a; Albalak et al., 2024). Specific rules can be found in Appendix A.

After paragraph and document granularity cleaning, we retained the main content of the web page and removed low-information documents. This cleaning process is trivial but important, it cuts down abundant duplicate data.

Quality filter   High-quality data can improve the performance of models. However, to our knowledge, there is no unified definition of data quality. Existing literature typically uses classification recognition or perplexity (PPL) truncation to partition high-quality data (Brown et al., 2020; Young et al., 2024). For classification methods, people use Wikipedia and books as high-quality data and use classifiers to identify target documents that are similar to these high-quality data, which may unexpectedly bias the data distribution and damage diversity. PPL uses a language model to calculate the PPL score of a document and then filters it using a specific threshold. However, PPL truncation may risk compromising high-quality data in specific niche domains, such as exams and ancient literature.

In our practice, we refer to the principle that "data that is known to be written by humans and has likely gone through an editing process" (Albalak et al., 2024; Gao et al., 2020) when selecting high-quality data. High-quality data has rich informativeness and good coherence, without any pornographic or violent elements. Based on these criteria, we manually annotated 60k samples using active learning, with a balanced number of positive and negative samples. We trained a classification model based on BERT (Devlin et al., 2018) to rate content quality, with an F1-score of 85%. According to the quality ratings, low-quality documents were removed during the quality filtering stage.

Personal identifiable information filter   Personal identity information (PII) of users may be leaked from web pages (Kumar et al., 2024; Subramani et al., 2023). PII will be abundant when the data scale is very large (Elazar et al., 2023). PII recognition can be achieved through model-based methods with better performance or rule-based methods with high computational efficiency (Soldaini et al., 2024). In our practice, we used rule-based methods through regular expressions, including email detection, IP address detection, and phone number recognition. The text block that hits the rule will be replaced with a specific token. Although the cleaning of PII does not significantly improve the performance of the model, it enhances the security protection for users in the network.

2.1.3 DATA DEDUPLICATION

Within the data recipe pipeline, deduplication is critically important due to the pervasive presence of duplicate content, which arises from various origins, including identical copies, cross-site plagiarism, and reference (Penedo et al., 2023). Although duplicate data can enhance the model's memorization, it weakens the model's generalization and impedes training efficiency (Hernandez et al., 2022; Lee et al., 2023; Marion et al., 2023; Tirumala et al., 2024; Sachdeva et al., 2024). Consequently, we devoted considerable efforts to exploring data deduplication. Our strategy is multi-level deduplication, including document-level, paragraph-level, and sentence-level, and detailed ablation experiments were conducted to validate the effectiveness.

Document deduplication   In document deduplication, we employed the Minhash (Wu et al., 2020) combined with the Locality Sensitive Hashing (LSH) (Jafari et al., 2021) method, which maps each document to a unique hash value with fixed length. This approach captures local features of the text, effectively identifying duplicate documents. In our implementation, the hash value was set to

128 bits. Document deduplication removed 15.40% of duplicate documents. Ablation experiments demonstrate a significant improvement in model performance due to document deduplication, with detailed data available in Appendix B.1.

Paragraph deduplication   Paragraph deduplication marks identical paragraphs across pages as duplicates, employing an exact match. We observed that paragraph deduplication removes a significant number of high-frequency template-like spans, such as homogeneous advertisements and recommendations, which to some extent compensate for the oversight of the junk text model. Paragraph deduplication can improve the corpus diversity but may compromise the text coherence, which is an accompanying side effect. To assess the impact of diversity and coherence on the model, we conducted a series of ablation experiments. Specifically, we aggregated each set of duplicate paragraphs into a group and deleted 10%, 20%, $\cdots$, and 50% of elements within the group to control the deduplication intensity. Greater intensity implies better data diversity. We found that as the deletion ratio increased from 10% to 30%, the model's performance steadily improved; however, further increasing the deletion ratio to 50% did not enhance the model's performance and even led to a slight decline. This suggests that: 1) the impact of diversity is more significant than the impact on coherence; 2) The growth in diversity tends to saturate at a deletion ratio of 30%. Further increases do not promote model performance and may cut down data quantity. Therefore, we ultimately chose a 30% deduplication ratio for duplicate paragraphs to optimize the dataset. Detailed ablation results can be found in Appendix B.2.

Sentence deduplication   Recognizing the potential of paragraph deduplication in enhancing the training of LLMs, we have further refined our approach by implementing sentence deduplication. This strategy targets the reduction of redundant text fragments found across web pages, a known factor that can impair model training, as highlighted in literature (Penedo et al., 2023; Hernandez et al., 2022). To address this, we have adopted a more stringent sentence deduplication strategy. Specifically, we count the frequency N of each sentence and retain only the square root of N unique sentences through deduplication. Moreover, to maintain coherence, we employ a sliding n-gram (where n=16) concatenation technique, thus mitigating the risk of damaging excessively short sentences. Our ablation experiments unequivocally demonstrate the positive impact of sentence deduplication on model performance. detailed results and analysis can refer to Appendix B.3.

2.1.4   DATA MIXTURE

Beyond web data, our corpus encompasses some high-quality sources, including Wikipedia, books, code stacks, and various domain-specific documents spanning legal, financial, and ancient texts, etc. Mixing these multi-source data for pretraining corpora presents a challenge (Ye et al., 2024; Chen et al., 2024b; Shen et al., 2023). In this report, we considered the following three aspects. First, data blending needs to balance diversity and quality. We found that compared to high-quality knowledge-intensive resources like Wikipedia, web data exhibit better diversity and quantity but inferior quality. Excessive skewness towards web data may degrade model performance. In such cases, mitigating data skewness can be achieved by appropriately oversampling knowledge-intensive data. Generally, oversampling ratios typically do not exceed 5 according to literature (Ye et al., 2024). Second, data blending needs to balance data efficiency and quantity. Data efficiency refers to the improvement of the model performance when trained on the same scale of data. Higher proportions of high-quality data can enhance data efficiency, yet such data is finite and thus necessitates down-sampling of lower-efficiency data. Nevertheless, empirical evidence from scaling laws underscores the critical role of data quantity in determining a model's ultimate performance. Additionally, data blending needs to consider the granularity of the mixture, with options ranging from source-level, topic-level, to content-type granularity. Finer-grained mixing can finely enhance the diversity of the blended data.

In our practice, on one hand, we applied meticulous cleaning strategies and multi-level deduplication to web data to compress data scale and enhance its efficiency. On the other hand, we balanced web data by moderately oversampling knowledge data. Furthermore, we have experimented with data mixing approaches by employing Doremi-based sampling techniques (Xie et al., 2024), as well as source- and topic-level mixing strategies geared towards maximizing diversity. Unfortunately, up until now, these explorations have not yielded statistically significant improvements. Our open-source model is based on a heuristic grid search to determine the ratios of each source, validated

Table 1: The data mixture for pretraining.

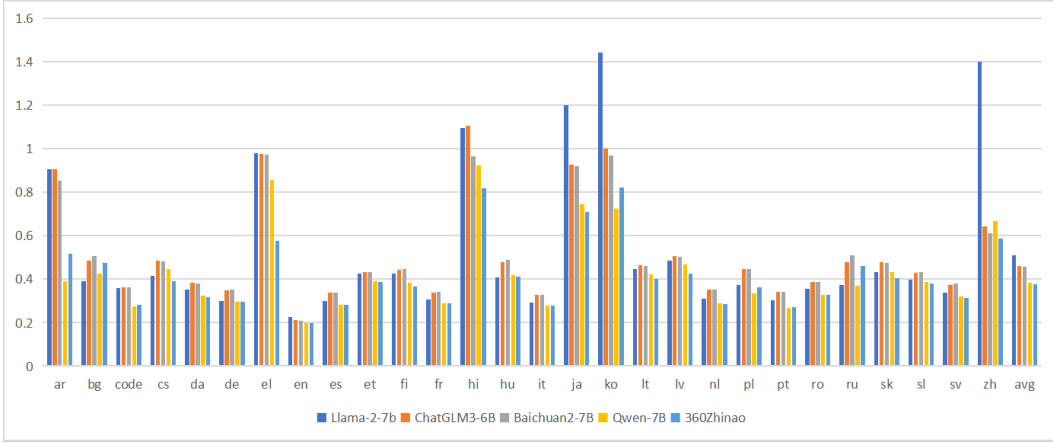| Source	Percentage
Webpages	63.33%
Code	9.25%
Math	7.31%
Books	6.20%
Patents	2.85%
Academic Papers	2.38%
Encyclopedia	0.96%
Other	7.72%

Figure 2: The comparison of compression rates among tokenizers.

through experiments to utilize the optimal ratios. The distribution of data ratios is depicted in Table 1. In the future, we envision further investigation into data mixture, which may harbor substantial improvement potential.

2.2 TOKENIZER

We focus on both the text compression rate of the tokenizer and its performance in downstream tasks. High compression rates can improve both training and inference efficiency. Considering encoding efficiency, we follow GPT-3.5 and GPT-4 (Brown et al., 2020; OpenAI, 2023), employing the efficient Byte Pair Encoding (BPE) tokenizer, tiktoken (OpenAI, 2022), for tokenization (Bai et al., 2023a). Given that OpenAI's cl100k vocabulary has shown good compression performance in English and many other languages, we adopt cl100k as our base vocabulary. However, this base vocabulary does not perform well in Chinese. Therefore, we augment the vocabulary with commonly used characters and words from Chinese and some other languages. Following Llama (Touvron et al., 2023a), we split numbers into individual digits to address inconsistent number splitting. Considering the prevalence of whitespace indentation in code data, we also define specialized tokens representing varying levels of space indentations, namely [space2], [space3], [space4], and [space8]. Consequently, our final vocabulary size amounts to 158k entries.

Our tokenizer's performance in compression rate is illustrated in Figure 2, where it is compared with tokenizers of Llama-2-7b (Touvron et al., 2023b), ChatGLM3-6B (Zeng et al., 2022), Baichuan2-7b (Yang et al., 2023), and Qwen-7B (Bai et al., 2023a). The results indicate that our tokenizer exhibits superior performance in compression rate.

6

2.3 Model architecture

Similar to most open-source models, the 360Zhinao model is based on the transformer architecture and roughly follows the LLama structure (Touvron et al., 2023b). It adopts the Pre-Norm structure with RMSNorm (Zhang & Sennrich, 2019) function and utilizes SwiGLU (Shazeer, 2020) as the activation function for the feed-forward network (FFN). The intermediate layer dimension is set to 8/3 times the hidden size and rounded up to 256. Additionally, Rotary Embedding (Su et al., 2024) is employed for positional encoding. Notably, we choose to use FP32 precision for the inverse frequency matrix instead of half-precision to achieve higher accuracy. For the entire model, we remove biases according to (Chowdhery et al., 2023), except for the biases in the attention QKV layers, to enhance the model's extrapolation capability (Su et al., 2024).

2.4 Training detail

We adhere to the standard autoregressive language modeling approach (Radford et al., 2018) for training the 360Zhinao model, which involves predicting the next token based on the preceding ones. For data organization, we concatenate documents within the same source, truncate them to a specified length, and shuffle them thoroughly, all within the training framework. We train models with a context length of 4096 and employ Flash Attention (Dao, 2023) in the attention module, effectively enhancing computational efficiency and reducing memory usage.

For optimization, we utilize AdamW (Loshchilov & Hutter, 2017) for pre-training, with hyperparameters set as follows: $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 1E - 8$, and weight decay $= 0.1$. Notably, the hyperparameter ϵ in AdamW is set to $1E - 5$ in Llama2 (Touvron et al., 2023b), which reveals gradient instability issues through our experiments. Experimental analysis in (Wortsman et al., 2023) explores the impact of this parameter on training and suggests appropriate reduction when scaling up the model size. Additionally, we do not employ dropout (Chowdhery et al., 2023) during the pre-training stage. We adopt a cosine learning rate schedule with warmup, where the learning rate decays to 10% of the maximum learning rate by the final training step (Scao et al., 2022). To ensure training stability, we have experimented with various tricks on FP16 training, including embedding norm (Scao et al., 2022), QK layernorm (Dehghani et al., 2023), normhead (Yang et al., 2023), etc. These methods essentially constrain potentially unstable forward activations using normalization operations. However, with the adoption of BF16, training models at the billion-parameter scale appear to exhibit less instability. Therefore, we adopt BF16 mixed precision as the standard for training, despite a slight sacrifice in memory usage. Grad clipping is set to 1.

2.5 Benchmark evaluation

We trained a 360Zhinao-7B-base model with 7 billion parameters on a corpus of 3.4 trillion tokens. In this section, we report the performance of our base model on standard benchmarks against other similarly scaled open-source models, such as Baichuan2-7B (Yang et al., 2023) and DeepSeek-7B (Bi et al., 2024). We evaluate the models across various dimensions, including natural language understanding, knowledge, mathematics, code generation, and logical reasoning, using eleven popular benchmarks including C-Eval (Huang et al., 2024) and AGIEval (Zhong et al., 2023), etc. Evaluation experiments are conducted in zero-shot or few-shot settings. Specifically, we adopt a 5-shot approach for C-Eval, MMLU (Hendrycks et al., 2020), and CMMLU (Li et al., 2023a), a 4-shot approach for MATH (Hendrycks et al., 2021) and GSM8K (Cobbe et al., 2021), a 3-shot approach for BBH (Suzgun et al., 2022), a 1-shot approach for MBPP (Austin et al., 2021), and 0-shot for others. For fairness in comparison, we strictly adhere to the official implementation of OpenCompass (OpenCompass-Contributors, 2023), and gathered baseline scores from the official OpenCompass platform.

The comparative results are presented in Table 2. Compared with models of similar size, the 360Zhinao-7B-base model demonstrates outstanding overall performance. It achieves top rankings in multiple benchmark assessments, such as C-Eval, MMLU, HellaSwag, and LAMBADA, showcasing strong competitiveness in both Chinese and English knowledge and reasoning comprehension abilities. These results highlight the model's superior capacity to handle diverse linguistic and cognitive tasks within the realms of natural language processing and understanding.

Table 2: Comparison of results on popular benchmarks.

Model	Avg	CEval	AGIEval	MMLU	CMMLU	HellaSwag	MATH	GSM8K	HumanEval	MBPP
Baichuan2-7B	41.49	56.3	34.6	54.7	57	67	5.4	24.6	17.7	24
Baichuan-7B	31.94	44.7	24.6	41.5	44.6	68.4	2.5	9.6	9.1	6.4
ChatGLM3-6B	58.67	67	47.4	62.8	66.5	76.5	19.2	61	44.5	57.2
DeepSeek-7B	39.8	45	24	49.3	46.8	73.4	4.2	18.3	25	36.4
InternLM2-7B	58.01	65.7	50.2	65.5	66.2	79.6	19.9	70.6	41.5	42.4
InternLM-7B	39.33	53.4	36.9	51	51.8	70.6	6.3	31.2	13.4	14
LLaMA-2-7B	33.27	32.5	21.8	46.8	31.8	74	3.3	16.7	12.8	14.8
LLaMA-7B	30.35	27.3	20.6	35.6	26.8	74.3	2.9	10	12.8	16.8
Mistral-7B-v0.1	47.67	47.4	32.8	64.1	44.7	78.9	11.3	47.5	27.4	38.6
MPT-7B	30.06	23.5	21.3	27.5	25.9	75	2.9	9.1	17.1	22.8
Qwen1.5-7B	55.12	73.57	50.8	62.15	71.84	72.62	20.36	54.36	53.05	36.8
Qwen-7B	49.53	63.4	45.3	59.7	62.5	75	13.3	54.1	27.4	31.4
XVERSE-7B	34.27	61.1	39	58.4	60.8	73.7	2.2	11.7	4.9	10.2
Yi-6B	47.8	73	44.3	64	73.5	73.1	6.3	39.9	15.2	23.6
360Zhinao-7B	56.15	74.11	49.49	67.44	72.38	83.05	16.38	53.83	35.98	42.4

2.6 ABLATION STUDY

As is well known, data determines the upper limit of model performance. Therefore, we have implemented meticulous strategies in data recipes. The effectiveness of these strategies will be validated through ablation experiments.

2.6.1 ABLATION DESIGN

Ablation evaluation aims to validate the effectiveness of strategies in small-data experimental scenarios. We establish a tripartite evaluation framework. First, a validation set is created to monitor its loss during the training process. Second, beyond the OpenCompass evaluation, we develop a more stable custom benchmark. Finally, supervised fine-tuning (SFT) evaluation is introduced, which can directly correlate with downstream task performance with a tangible measure.

Typically, the validation set is sampled from the training dataset. To ensure the diversity of the validation set, we employ sampling at the topic granularity and ensure that there is no overlap between the validation and the training data. The validation set is utilized to assist in assessing model performance by valid loss and make early stopping decisions during the training stage.

Ablation experiments are conducted on small models and limited tokens to sensitively evaluate the effectiveness of data strategies. Generally, OpenCompass is a widely used benchmark to assess zero/few-shot generation capabilities. However, we encountered two issues: 1) instability and insensitivity of OpenCompass to subtle model differences on small-scale ablation, and 2) difficulty of aligning OpenCompass with downstream-specific business capabilities. To address these challenges, we made two efforts. First, we systematically curated a subset from OpenCompass, retaining effective and sensitive evaluations. Second, we developed a custom benchmark, named 360Eval, which is more suitable for ablation and aligns with downstream performance. 360Eval contains a hierarchical skill list, including machine reading comprehension, classification, extraction, translation, examination, and summarization. For each skill, we collected multiple publicly available datasets to ensure the diversity of the evaluation. To make metrics more robust and effective, we enhance the diversity of prompts by constructing multiple templates. Additionally, we improve the way of metric computation by adopting a generative approach. For example, in multiple-choice questions, the generation of answer options (e.g., A, B, C, D) is supplemented by generating answer content (Zheng et al., 2023). Finally, we reduce the sample size at the task granularity to balance evaluation efficiency and error. Regarding potential contamination, GPT-3 indicates that low levels of contamination do not lead to unfairness issues without specific intervention. Our experimental findings concur with this

conclusion. Therefore, the decontamination has been omitted to help alleviate computational costs during data iterations.

As a pivotal component of our evaluation, the SFT evaluation, despite its high computational cost, directly reflects the downstream effects of the base model. First, we fine-tune the base model to a chat model and then utilize a trained reward model to score the generated results by the chat model. Finally, we conduct evaluation comparisons of the ablation experiments from three dimensions: 360Eval, OpenCompass, and SFT evaluation.

2.6.2 ABLATION SETTING

The ablation experiments were conducted on a 1.8B model and trained on 100B tokens. Carefully constructed training data is essential for ablation experiments. We set a fixed, diverse mixed dataset as a base training set. The base data is fixed to facilitate comparisons across multiple strategies, while its diversity ensures the model's basic effectiveness. To enhance the discriminability of the ablation strategies, during the ablation experiments, our training data consists of 50% experimental data and 50% base data. A higher proportion of experimental data enhances the discriminability of the ablation strategies.

Based on this experimental setup, we conducted ablation analyses on document, paragraph, and sentence deduplication. Detailed experimental results can be found in Appendix B.

3 ALIGNMENT

Following common practice (Ouyang et al., 2022; Bai et al., 2022; Touvron et al., 2023a; Bai et al., 2023a; Yang et al., 2023; Bi et al., 2024), we performed SFT and RLHF on the pretrained models.

For SFT, we continuously improved our training data. We initially had over 10 million prompt-answer pairs and iteratively reduced the number to hundreds of thousand while improving their quality. We also constructed PoT (Chen et al., 2023b) data for math and reasoning tasks. We specifically extended the context length from 4K to 32K and 360K during finetuning with simple RoPE-base changes and constructed long SFT data (Xiong et al., 2023; Liu et al., 2024a).

For RLHF, we continuously worked on its infrastructure to support PPO on larger models with minimal hardware. We achieved a little improvement on general LLM tasks and more significant gains in specific areas such as translation and agentic tasks.

Human annotators play an important role throughout the alignment stage. They provide high-quality SFT data, preference data and objective comparison of different model versions. We have built an annotation team of around 20 people, and worked closely with them in the loop on each annotation task to support 360's AI products such as 360 AI-search[1] and 360 AI-browser[2].

3.1 SFT

We took a data-centric approach to SFT and continuously improved the SFT dataset, firstly in quantity and then in quality. PoT data is specifically constructed for math and reasoning tasks to mitigate the dissatisfaction of 7B-sized models to compute and reason. Those efforts resulted in the competitive performance of 360Zhinao-7B-Chat-4K on relevant benchmarks.

Building upon the 4K model, we extended its context length to 32K and 360K with simple RoPE-base changes and carefully curated long data. The 32K and 360K models exhibit top-tier performance on LongBench (Bai et al., 2023b) and near-perfect all-green results on NIAH (Needle-In-A-Haystack evaluation) (Kamradt, 2023; Arize-AI, 2024).

3.1.1 SFT DATA

Quantity-Quality: Early in 2023 when we just started training 360Zhinao models, we prioritized quantity over quality of the SFT data. Back then, we expanded our SFT dataset every time one batch

---

[1]https://so.360.com/
[2]https://browser.360.cn/ai/

of human annotation is ready or one open-sourced dataset is accessible, and model performance did improve with such expansion.

However, such procedure quickly became impractical in mid 2023. The sheer number of over 10 million prompt-answer pairs prohibited iterative data cleaning and quality improvement. Emphasis was therefore shifted from quantity to quality, aiming to limit total SFT data size within hundreds of thousand to facilitate continuous efforts.

Open-sourced English data are generally of higher quality than Chinese counterparts. We empirically found that adding some of high-quality open-sourced English SFT data (Mukherjee et al., 2023; Lian et al., 2023; Mitra et al., 2024; Databricks, 2023) has positive effects even on our internal pure-Chinese evaluation set. We posit this is because pretrained LLMs are inherently aligned across languages exhibiting translation capabilities, and training on instruction-following data in one language benefits alignment in all languages. We generally have less than 5% English SFT data in the training dataset.

Quality processing: To prune the SFT data, we first grouped them by their original sources and rated each source manually on a randomly sampled subset. Data from sources rated below a certain threshold were directly discarded.

We then leveraged the 360Zhinao model itself to label the categories and difficulty levels of the remaining data. Such labelling is achieved by first asking our annotation team to label a seed set in category and difficulty level and then fine tuning a 360Zhinao model specifically on this seed set. The model could then label all the data as initial annotation.

Current status: We have been continuously working on the resulting data to correct the category and difficulty labels, prune/reweight the data according to the labels and improve the quality of the prompts and answers therein. Unreasonable prompts were removed or modified and answers were refined and reformatted. This is an endless process and we always find some of us on it. Upon release of 360Zhinao-7B v1.0, we have around 600 thousand SFT data with manageable size of each category for us to iterate over and improve upon. New high-quality data are still added to this dataset on a regular basis, but we now keep dataset expansion at a more steady pace with higher standards on quality.

PoT: For math and reasoning tasks, we let our model write Python code to compute instead of generating the computational output directly with natural language, similar to Chen et al. (2023b). Such SFT data were constructed with relevant prompts and manually verified one-by-one by specifically delegated colleagues to ensure correctness. Special tokens were added right before and after the Python code in the SFT data so that the generated code could be identified by a separate code interpreter in deployment to get the executed results.

3.1.2 4K MODEL TRAINING AND EVALUATION

The released 360Zhinao-7B v1.0 model (360Zhinao-7B-Chat-4K) was trained with standard SFT, with data packed into the 4K window. We have tried separating attention masks of different data with `flash_attn_varlen_func` but found it inferior to naive packing, possibly because naive packing approximately induces the desired multi-turn conversation capabilities. Besides naive packing, we also constructed around 5,000 multi-turn conversations with role exchange prompting (Ding et al., 2023) and role-playing ghost attention (Touvron et al., 2023b) to explicitly induce multi-turn capabilities. Some other data in our dataset are also naturally multi-turn.

Table 3 demonstrates competitive results of our released 4K model on MT-bench (Zheng et al., 2024), a benchmark with 80 high-quality multi-turn dialogue questions to test the capabilities of multi-turn dialogue and instruction following. MT-bench consists of 8 common user prompt categories: writing, role-play, extraction, reasoning, mathematics, coding, knowledge I (STEM), and knowledge II (humanities/social sciences). For each category, 10 multi-turn questions were manually designed, each turn has 2 questions, and the answers are scored using GPT-4 (accessed on Apr. 8, 2024).

We also compared against Qwen-7B-Chat (Qwen v1.0) on our internal evaluation set consisting of 1,100 prompts covering tasks of OpenQA, ClosedQA, summarization, COT questions and code, etc (similar to the prompt categories in Ouyang et al. (2022)). 360Zhinao-7B-Chat achieved an average score of 0.79 compared to Qwen-7B-Chat's 0.77. Detailed scores are shown in Figure 3.

Model	Turn1	Turn2	average
Qwen-7B-Chat	**6.5725**	5.4000	5.9862
Baichuan2-7B-Chat	6.4562	5.5562	6.0062
InternLM-7B-Chat	5.5625	4.0696	4.8207
360Zhinao-7B-Chat	6.5062	**5.8762**	**6.1962**

Table 3: MTBench evaluation results.

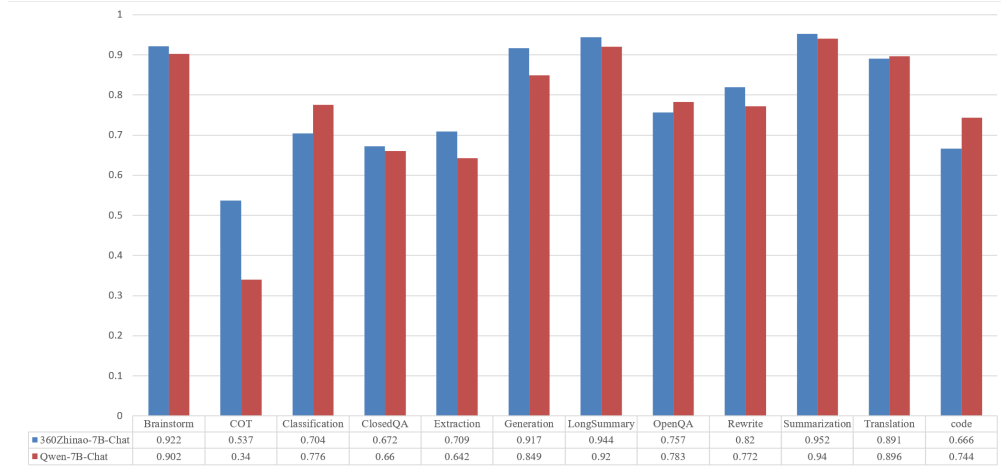|	Brainstorm	COT	Classification	ClosedQA	Extraction	Generation	LongSummary	OpenQA	Rewrite	Summarization	Translation	code
360Zhinao-7B-Chat	0.922	0.537	0.704	0.672	0.709	0.917	0.944	0.757	0.82	0.952	0.891	0.666
Qwen-7B-Chat	0.902	0.34	0.776	0.66	0.642	0.849	0.92	0.783	0.772	0.94	0.896	0.744

Figure 3: Results on the internal evaluation set of 360Zhinao-7B-Chat and Qwen-7B-Chat. 360Zhinao-7B-Chat outperforms Qwen-7B-Chat on most prompt categories.

3.1.3 32K AND 360K MODELS

We extended the context window of 360Zhinao-7B (base model) to 32K and 360K in the SFT stage with minimal costs. To train the long-context models, we adopted a two-stage approach:

- **First stage:** We increased RoPE base and extended the context length to 32K.

  Firstly, we performed Continual Pretraining on approximately 5B tokens with a 32K context window.

  Then we finetuned the model using long data from various sources, including high-quality human-labeled 32K data, together with the general SFT data in Sec. 3.1.1.

- **Second stage:** We extended the context length to 360K, training with the following data:

  1. A small number of high-quality human-labeled super-long data.

  2. Due to the scarcity of annotated super-long data, we constructed various forms of synthetic data:

    – Multi-Doc QA: Similar to Junqing et al. (2023), we generated multi-document QA pairs based on our own database. Multiple QA pairs are constructed for one row of Multi-Doc QA data input, resulting in a multi-turn format and significantly improving the training efficiency.

    – Single-Doc QA: Similar to Xiong et al. (2023), we constructed multi-turn QA data based on different segments within one row of long-text input.

We initially trained 360K models with Megatron-LM (Shoeybi et al., 2019; Korthikanti et al., 2023) after optimizing its GPU memory consumption by removing unnecessary attention masks and changing to in-place operations. However, Megatron's tensor parallelism scales poorly with sequence length, and its context parallelism wasn't quite ready back when we were choosing from training options. We have now switched to Ring Attention (Li et al., 2021; Liu et al., 2023), which requires much fewer machines and has similar overall training efficiency.

Model	Avg	Single-Doc QA	Multi-Doc QA	Summa-rization	Few-shot Learning	Code Comple-tion
GPT-3.5-Turbo-16k	37.84	61.2	28.7	16	29.2	54.1
ChatGLM2-6B-32k	37.16	51.6	37.6	16.2	27.7	52.7
ChatGLM3-6B-32k	44.62	**62.3**	44.8	17.8	42	56.2
InternLM2-Chat-7B	42.20	56.65	29.15	**17.99**	43.5	**63.72**
Qwen1.5-Chat-7B	36.75	52.85	30.08	14.28	32	54.55
Qwen1.5-Chat-14B	39.80	60.39	27.99	14.77	37	58.87
360Zhinao-7B-Chat-32K	**45.18**	57.18	**48.06**	15.03	**44**	61.64
360Zhinao-7B-Chat-360K	39.25	52.82	48.01	14.4	41.25	39.75

Table 4: LongBench evaluation results. 360Zhinao-7B-Chat-32K achieved the best score among models within 10B. The 360K model only degraded significantly on code, mainly because the 360k-stage data of this version had no code. Later experiments with code data helped resolve this.

Method: Initially, we increased RoPE base from 10,000 to 1,000,000 in the first stage, and then from 1,000,000 to 50,000,000 in the second stage. This gives satisfactory results on LongBench (Bai et al., 2023b) with the 32K model and on NIAH with the 360K model, accompanied by however notable degradation with the 360K model on LongBench. Therefore, we chose to use RoPE base of 50,000,000 in both 32K and 360K models. We empirically found that this had no impact on 32K performance, but significantly mitigated degradation of the 360K model on 32K tasks.

LongBench: We report evaluation results on LongBench (Bai et al., 2023b) in Table 4. 360Zhinao-7B-Chat-32K achieved the best score among models within 10B. By using RoPE base of 50,000,000 directly in the first stage, 360Zhinao-7B-Chat-360K mostly retained its 32K performance and only degraded significantly on code, mainly because our second-stage data to train the open-sourced 360Zhinao-7B-Chat-360K model had no code. We could improve its code performance simply by adding code data in the second stage in later experiments.

NIAH: Besides retaining 32K performance in the 360K model, we mainly focused on improving NIAH (Needle-In-A-Haystack evaluation) (Kamradt, 2023; Arize-AI, 2024) within the 360K-token context window. Although NIAH is a somewhat artificial and unnatural task, it still serves as a necessary check for long-context models. Unsatisfactory NIAH results cast shadow on real-world long context applications.

It's worth mentioning that in our experiments, we found the value retrieval variant (Arize-AI, 2024) of NIAH (widely used recently in Gemini-Team et al. (2024); Liu et al. (2024a)) is relatively easy and we could quickly get 100% perfect all-green results. However, the model checkpoints with all-green results on the value retrieval variant still struggles with the original version (Kamradt, 2023) of NIAH. We posit that the needle format of the value retrieval variant, "The special magic {random city} number is: {random 7-digit number}.", is much more distinct than the original-version needle "The best thing to do in San Francisco is ..." in the haystack of natural text. The commonly used haystack of Paul Graham essays discusses Silicon Valley and inevitably San Francisco, which poses even more confusion with the original-version needle.

We plot the results of the original version of NIAH and its adapted Chinese version in Figure 4 (middle & right). They are both much more difficult than the value retrieval variant. The details of the Chinese version could be found on our GitHub page. We have verified that our open-sourced 360Zhinao-7B-Chat-360K indeed achieves 100% perfect all-green results on the value retrieval variant (Figure 4, left). The original versions are more tricky and the results are more unstable, depending on the inherent capability of different base models.

Applications: When applying the 32K and 360K models in the backend of 360 AI-browser, the 360k model, though end-to-end, exhibits considerable inference delay and is used offline to process long documents' results into cache. The 32K model is used online to process extremely-long docu-
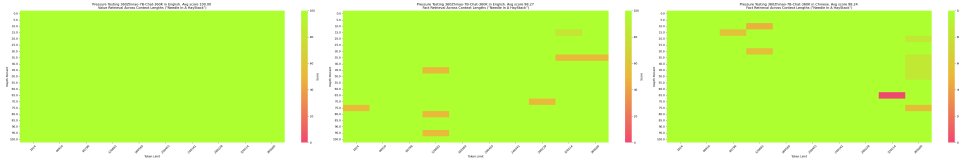
Figure 4: From left to right: results on random city-number value retrieval (Arize-AI, 2024), original NIAH (Kamradt, 2023) and Chinese NIAH constructed by us. Value retrieval is relatively easy.

ments in chunks in a parallel-recursive manner (Wu et al., 2021) to interact with users in acceptable response time, giving user summaries, mind maps and QAs.

3.2 RLHF

General RLHF: We have been working on RLHF since early 2023. Generally, we have continuously improved our PPO codebase and infrastructure and could support RLHF on large models of up to ∼100B size. We have accumulated over 1,500,000 high-quality pairs of preference data for RM training from filtered open-source data, feedback from our annotation team and past human comparison of multiple results on historical prompts. The resulting RM essentially follows the RM scaling trends (Touvron et al., 2023a; Bai et al., 2022; Gao et al., 2023) and achieves around 70% - 75% validation accuracy, increasing with the RM model size (internally 7B, 13B and 70B). As for the RL algorithm we now mostly use a PPO variant similar to ReMax (Li et al., 2023b), discarding the value network in PPO. On general LLM tasks as categorized by Ouyang et al. (2022), PPO models usually achieves a little improvement on average, though not as significant as RLHF at OpenAI (Ouyang et al., 2022).

Specific RLHF: PPO improvement on specific tasks are more often observed and has therefore become an indispensable step in domain-specific models. The translation LLM behind our AI-browser improved a lot after PPO training in terms of translation quality and code switch (e.g. undesirable mixed Chinese-English output). The agent LLM after PPO exhibited more accurate function calling and lower false recall rate. We posit this is because task-specific RMs are easier to train with more focused pairwise data, and rule-based rewards could be derived for code switch and function calling as the most accurate feedback reward to be added to the RM reward.

RMs: An unexpected and nice by-product of our RLHF endurance is the versatility of RM. We found it quite pleasing to use our RM as judge to compare different versions and checkpoints, as referee to select the best response from multiple open-source competitive LLMs for each prompt and as filters to clean our dataset. Deployed locally, our own RM accomplishes the aforementioned tasks in merely tens of seconds on a large dataset and gives satisfactory though not perfect results.

DPO: Besides full RLHF, we have also observed recently positive signs of DPO variants, more specifically with ORPO (Hong et al., 2024) and NCA (Chen et al., 2024a). They demonstrated decent potential with LoRA (Hu et al., 2022) and we are working to reliably harness their gains.

4 CONCLUSION, LIMITATION, AND FUTURE WORK

This technical report introduces our approaches to pretraining and finetuning 360Zhinao models. During pretraining we explored various data cleaning and composition strategies and validated them with our carefully curated ablation environment that's designed to be stable and sensitive on minimal model size, which provides guidance to train our 7B model on 3.4T high-quality tokens. When performing alignment we filtered and reformatted our SFT data to balance quantity over quality. We extended its supported sequence length to 32K and 360K with simple RoPE-base change and tailored long data. Following SFT, RMs and RLHF are also trained and credibly applied to specific tasks and pipelines in our work.

Scope of this paper is limited to 7B models, and we hope to share our progress on larger models in the near future. Training LLMs is an endless expedition and we are still constantly improving our data, infrastructure, model architecture and evaluation protocol.

5 Author Contributions

Xiangzheng Zhang

Pretraining:

Jie Xiong
Zhuang Xiong
Cunqi Zhai
Lin Li
Huanyong Liu
Xiaochun Gong

Alignment:
Qi An
Zhe Wei
Lifu Tang
Junchen Liu
Fenrui Xiao
Huan Liu
Liang Wen
Xin He
Haosheng Zou
Yongchao Deng
Shousheng Jia

References

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

Arize-AI. https://github.com/Arize-ai/LLMTest_NeedleInAHaystack, 2024.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023a.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*, 2023b.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

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

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

Daoyuan Chen, Yilun Huang, Zhijian Ma, Hesen Chen, Xuchen Pan, Ce Ge, Dawei Gao, Yuexiang Xie, Zhaoyang Liu, Jinyang Gao, et al. Data-juicer: A one-stop data processing system for large language models. *arXiv preprint arXiv:2309.02033*, 2023a.

Huayu Chen, Guande He, Hang Su, and Jun Zhu. Noise contrastive alignment of language models with explicit rewards. *arXiv preprint arXiv:2402.05369*, 2024a.

Mayee Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! a data-driven skills framework for understanding and training language models. *Advances in Neural Information Processing Systems*, 36, 2024b.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023b. ISSN 2835-8856. URL https://openreview.net/forum?id=YfZ4ZPt8zd.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Databricks. https://huggingface.co/datasets/databricks/databricks-dolly-15k, 2023.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pp. 7480–7512. PMLR, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

Yanai Elazar, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, et al. What's in my big data? *arXiv preprint arXiv:2310.20707*, 2023.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Gemini-Team, Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, et al. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2024.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36, 2024.

Omid Jafari, Preeti Maurya, Parth Nagarkar, Khandker Mushfiqul Islam, and Chidambaram Crushev. A survey on locality sensitive hashing algorithms and their applications. *arXiv preprint arXiv:2102.08942*, 2021.

He Junqing, Pan Kunhao, Dong Xiaoqun, Song Zhuoyang, Liu Yibo, Liang Yuxin, Wang Hao, Sun Qianguo, Zhang Songxin, Xie Zejian, et al. Never lost in the middle: Improving large language models via attention strengthening question answering. *arXiv preprint arXiv:2311.09198*, 2023.

Gregory Kamradt. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2023.

Vijay Anand Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models. *Proceedings of Machine Learning and Systems*, 5, 2023.

Ashutosh Kumar, Sagarika Singh, Shiv Vignesh Murty, and Swathy Ragupathy. The ethics of interaction: Mitigating security threats in llms. *arXiv preprint arXiv:2401.12273*, 2024.

Alycia Lee, Brando Miranda, and Sanmi Koyejo. Beyond scale: the diversity coefficient as a data quality metric demonstrates llms are pre-trained on formally diverse data. *arXiv preprint arXiv:2306.13840*, 2023.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*, 2023a.

Shenggui Li, Fuzhao Xue, Chaitanya Baranwal, Yongbin Li, and Yang You. Sequence parallelism: Long sequence training from system perspective. *arXiv preprint arXiv:2105.13120*, 2021.

Ziniu Li, Tian Xu, Yushun Zhang, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023b.

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". Openorca: An open dataset of gpt augmented flan reasoning traces. https://https://huggingface.co/Open-Orca/OpenOrca, 2023.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023.

Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with ringattention. *arXiv preprint arXiv:2402.08268*, 2024a.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=BTKAeLqMw.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*, 2023.

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

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4, 2023.

OpenAI. tiktoken: A fast bpe tokeniser for use with openai's models. https://github.com/openai/tiktoken, 2022.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

OpenCompass-Contributors. Opencompass: A universal evaluation platform for foundation models. *GitHub repository*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

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

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

Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. How to train data-efficient llms. *arXiv preprint arXiv:2402.09668*, 2024.

Teven Le Scao, Thomas Wang, Daniel Hesslow, Lucile Saulnier, Stas Bekman, M Saiful Bari, Stella Biderman, Hady Elsahar, Niklas Muennighoff, Jason Phang, et al. What language model to train if you have one million gpu hours? *arXiv preprint arXiv:2210.15424*, 2022.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Joel Hestness, Natalia Vassilieva, Daria Soboleva, and Eric Xing. Slimpajama-dc: Understanding data combinations for llm training. *arXiv preprint arXiv:2309.10818*, 2023.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, 2024.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Nishant Subramani, Sasha Luccioni, Jesse Dodge, and Margaret Mitchell. Detecting personal information in training corpora: an analysis. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pp. 208–220, 2023.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*, 36, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, et al. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*, 2024.

Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, et al. Skywork: A more open bilingual foundation model. *arXiv preprint arXiv:2310.19341*, 2023.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*, 2019.

Mitchell Wortsman, Peter J Liu, Lechao Xiao, Katie Everett, Alex Alemi, Ben Adlam, John D Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, et al. Small-scale proxies for large-scale transformer training instabilities. *arXiv preprint arXiv:2309.14322*, 2023.

Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*, 2021.

Wei Wu, Bin Li, Ling Chen, Junbin Gao, and Chengqi Zhang. A review for weighted minhash algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 34(6):2553–2573, 2020.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36, 2024.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. Effective long-context scaling of foundation models. *arXiv preprint arXiv:2309.16039*, 2023.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32, 2024.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Yunhua Zhou, Jun Zhan, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *arXiv preprint arXiv:2403.16952*, 2024.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*, 2023.

A DATA CLEANING

The main rules in the data cleaning stage include:

1. Removing documents with a high proportion of punctuation.
2. Removing documents with a high proportion of paragraphs ending in ellipsis.
3. Remove documents with high paragraph ratios that lack punctuation endings;
4. Removing documents containing too many excessively long or short abnormal English words and numbers.
5. Removing documents with too many repetitive sentences;
6. Removing documents that contain too many short paragraphs;
7. Removing documents containing too many repeated n-grams (n=1,2,3).

Among the above rules, we set a relatively loose threshold for examination documents to avoid accidental deletion.

B   ABLATION RESULT

This section will report and analyze the ablation results of data strategies. We will plot training loss and validation loss graphs to aid in the analysis and present the ablation results of 360Eval, OpenCompass, and SFT in tabular form. To save space, the metrics in the table headers are abbreviated as follows: Bs (BrainStorm), Cot (Chain of Thought), Cls (Classification), Cqa (ClosedQA), Comp (Comprehension), Ext (Extraction), Gen (Generation), Sums (Summarization Short Text), Suml (Summarization Long Text), Oqa (OpenQA), Rewr (Rewrite), Tran (Translation), Mrc (Machine Reading Comprehension), Exam (Examination), Klg (Knowledge), Reas (Reasoning). The average of metrics of each evaluation is represented as Avg (Average) to indicate the model's final performance.

B.1   DOCUMENT DEDUPLICATION
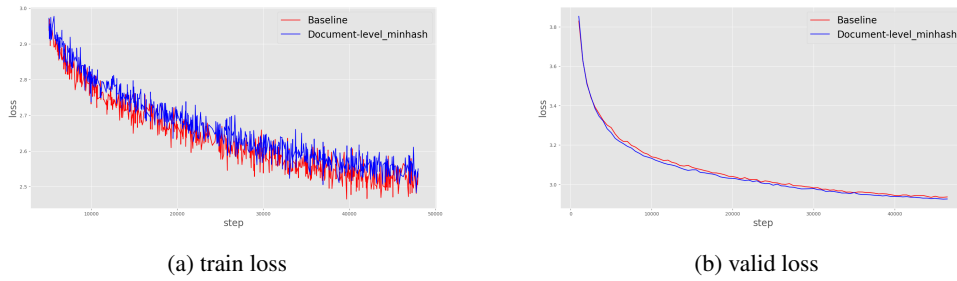

(a) train loss   (b) valid loss

Figure 5: Loss curves of Document deduplication.

Table 5: Ablation results of document deduplication strategy on 360Eval and OpenCompass.

Document	360Eval							OpenCompass					
	Avg	Mrc	Ext	Tran	Cls	Exam	Sum	**Avg**	Exam	Lang	Klg	Usd	Reas
w/o dedup	18.52	23.65	19.69	1.69	32.45	12.31	21.3	32.59	27.17	59.62	36.53	29.81	9.81
w/ dedup	**18.77**	24.33	19.3	1.93	37.6	8.98	20.5	**32.84**	25.54	63.46	36.22	28.83	10.13

Table 6: Ablation results of document deduplication strategy on SFT evaluation.

| Document	**Avg**	Bs	Cot	Cls	Cqa	Comp	Ext	Gen	Sums	Oqa	Rewr	Suml	Tran	Code
w/o dedup	45.4	86.2	14.6	39.9	28	39.8	29	61.1	63.8	50.8	37.6	72.5	48.3	18.6
w/ dedup	**46.25**	84.6	21	44.5	31.5	38.1	26.8	61.4	66	48.2	36.5	79.3	41	22.4

Figure 5 displays the loss curves for document deduplication performed on web data. The loss curves show that the training loss of "document w/ dedup" is higher than that of "document w/o dedup", but the validation loss is lower. This suggests that the former exhibits better data diversity and results in a model with stronger fitting capability. Tables 5 and 6 present the results of document deduplication on benchmark and SFT, respectively. It can be observed that "document w/ dedup" performs better, with an improvement of 1.47% on SFT, 0.99% on 360Eval, and 0.02% on OpenCompass. Our designed SFT and 360Eval evaluation systems are more sensitive to OpenCompass.

B.2   PARAGRAPH DEDUPLICATION

Figure 6, Table 5, and Table 6 present the ablation results of paragraph deduplication at levels from 0% to 50%. From the loss curves, it is observed that the training loss of 30%, 40%, and 50% paragraph deduplication is higher, but the validation set loss is lower, indicating that the paragraph deduplication strategy helps to improve data diversity. For ablation results, the best performance is achieved when the deduplication ratio is set to 30%. Therefore, considering both data quantity and efficiency, we ultimately set the paragraph deduplication ratio to 30%.
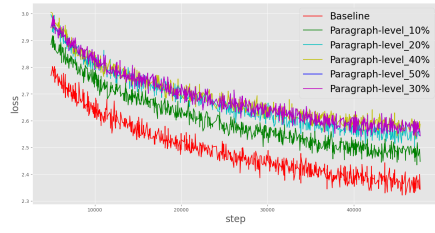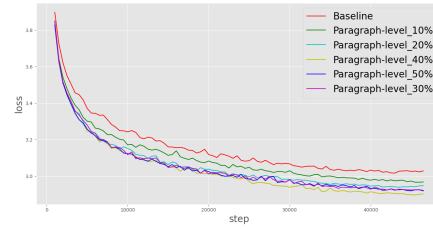
(a) train loss   (b) valid loss

Figure 6: Loss curves of Paragraph deduplication.

Table 7: Ablation results of paragraph deduplication strategy on 360Eval and OpenCompass.
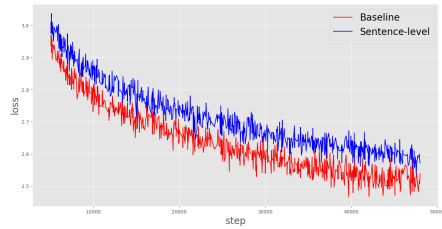
Paragraph	360Eval							OpenCompass					
	Avg	Mrc	Ext	Tran	Cls	Exam	Sum	**Avg**	Exam	Lang	Klg	Usd	Reas
w/o dedup	21	29.42	20.5	2.49	41.8	9.75	22.06	32.29	25.84	58.65	34.02	31.21	11.73
dedup_10%	20.27	25.74	21.62	3.09	41	8.35	21.82	34.05	26.87	65.38	36.57	31.24	10.19
dedup_20%	21.7	30.11	24.05	3.09	38.75	9.73	24.49	32.22	28.77	61.54	31.49	28.04	11.25
dedup_30%	**23.17**	34.37	24.48	3.99	37.85	14.11	24.25	**34.61**	31.05	63.46	34.57	33.12	10.85
dedup_40%	22.51	32.29	24.51	4.64	38.5	12.86	22.29	33.48	29.27	60.58	35.26	31.37	10.93
dedup_50%	22.45	34.77	17.18	4.38	38.1	15.84	24.44	33.61	29.03	63.46	32.33	31.54	11.67

Table 8: Ablation results of paragraph deduplication strategy on SFT evaluation.

Paragraph	Avg	Bs	Cot	Cls	Cqa	Comp	Ext	Gen	Sums	Oqa	Rewr	Suml	Tran	Code
w/o dedup	47.92	86.3	19	39.4	30.2	42.6	36.2	61.3	66.1	50.4	44.9	76.3	48.8	18.4
dedup_10%	48.79	84.3	19	46.4	33.3	45.9	34.3	61.7	67.4	48.9	40.2	70.8	53.5	28.6
dedup_20%	49.75	86	15.9	47.4	36.8	40.7	32.1	63.7	73.2	51.7	40.3	80.4	51.8	26.8
dedup_30%	**53.04**	87.5	28.4	43	36.8	51.2	39.1	65.3	75.8	50.2	52.8	79.2	52.9	27.3
dedup_40%	51.41	88.9	17.3	43.5	33.7	51	43.2	67.7	69.8	54.8	44.4	79.6	55.9	18.6
dedup_50%	52.33	86	26.7	52.3	36.2	49.9	38.6	65.2	75.9	50.9	47.1	81	48.1	22.4

B.3 SENTENCE DEDUPLICATION

Based on the findings from Figure 7, Table 9, and Table 10, it is evident that sentence deduplication significantly improves the performance of the model on 360Eval and SFT evaluation. Therefore, we integrate sentence deduplication as a data strategy into the recipe pipeline.
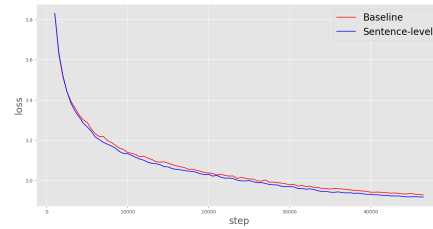
(a) train loss   (b) valid loss

Figure 7: Loss curves of sentence deduplication.

Table 9: Ablation results of sentence deduplication strategy on 360Eval and OpenCompass.

Sentence	360Eval							OpenCompass					
	Avg	Mrc	Ext	Tran	Cls	Exam	Sum	**Avg**	Exam	Lang	Klg	Usd	Reas
w/o dedup	20.57	27.97	21.14	2.77	34	13.65	23.92	**33.8**	28.94	64.42	36.53	30	9.14
w/ dedup	**21.21**	28.07	18.9	4.12	36.2	15.02	24.97	32.94	27.77	61.54	36.33	29.71	9.34

Table 10: Ablation results of sentence deduplication strategy on SFT evaluation.

Sentence	**Avg**	Bs	Cot	Cls	Cqa	Comp	Ext	Gen	Sums	Oqa	Rewr	Suml	Tran	Code
w/o dedup	51.13	85.3	22.5	46.3	34.9	45.4	40.5	62.4	74.2	53.6	52.7	81.7	46.1	20.6
w/ dedup	**52.64**	86.5	20.2	53.4	37.4	50.9	37.3	67.8	72.5	52.2	51.5	79.3	49.4	24.8