Towards Effective and Efficient Continual Pre-training of Large Language Models

Anonymous ACL submission

Abstract

Continual pre-training (CPT) has been an important approach for adapting language models to specific domains or tasks. In this paper, we comprehensively study its key designs to balance the new abilities while retaining the original abilities, and present an effective CPT method that can greatly improve the Chinese language ability and scientific reasoning ability of LLMs. To achieve it, we design specific data mixture and curriculum strategies based on existing datasets and synthetic high-quality data. Concretely, we synthesize multidisciplinary scientific QA pairs based on related web pages to guarantee the data quality, and also devise the performance tracking and data mixture adjustment strategy to ensure the training stability. For the detailed designs, we conduct preliminary studies on a relatively small model, and summarize the findings to help optimize our CPT method. Extensive experiments on a number of evaluation benchmarks show that our approach can largely improve the performance of Llama-3 (8B), including both the general abilities (+8.81 on C-Eval and +6.31 on CMMLU) and the scientific reasoning abilities (+12.00 on MATH and +4.13 on SciEval).

1 Introduction

Recently, large language models (LLMs) (Zhao et al., 2023; AI@Meta, 2024; Yang et al., 2024; DeepSeek-AI et al., 2024) have achieved great progress in accelerating the development of artificial intelligence. Unlike traditional machine learning methods, LLMs basically undergo large-scale pre-training on unsupervised corpora, *e.g.,* trillions of training tokens. Through pre-training, LLMs can learn extensive knowledge from unsupervised data and acquire the capability of solving various downstream tasks via prompting (Touvron et al., 2023a; OpenAI, 2023; Team et al., 2024).

Despite the success, LLMs still struggle in some specific scenarios, due to the large knowledge gap between pre-training data and downstream tasks. For example, Llama-3 (AI@Meta, 2024), primarily trained on English general corpora, performs not well on the tasks based on other languages (*e.g.,* Chinese (Cui et al., 2023)) or requiring multidisciplinary scientific knowledge, *e.g.,* physics and biology. To address these issues, a widely-used approach is to conduct *continual pre-training (CPT)* for LLMs on specially-curated data related to the expected abilities (Ke et al., 2023; Gupta et al., 2023; Ibrahim et al., 2024). However, *catastrophic forgetting* (Luo et al., 2023) has become a common technical issue for existing CPT methods, where new capabilities are improved but original capabilities are substantially hurt. Although CPT has been widely used in existing work, the key training details (*e.g.,* data selection, mixture, and curriculum) to develop new abilities and maintain existing abilities have not been well discussed, especially how to boost the comprehensive capacities of a well-trained model under a limited training budget.

In this paper, we present a completely transparent procedure for continually pre-training the open-sourced LLM—Llama-3 (8B), with all experimental data, model checkpoints, and training code released. Our focus is to enhance the model's capacities from two major aspects: *Chinese language ability* and *scientific reasoning ability*, while retaining its original capabilities. To achieve this, we design specific data curation strategies to improve the backbone models. For Chinese language ability, we collect and select extensive Chinese text data from diverse sources for effective bilingual adaptation. For scientific reasoning ability, we draw inspiration from the exercises in textbooks and employ LLMs to synthesize scientific question and answer (QA) pairs based on the content of web pages in the pre-training corpus. Furthermore, we also incorporate large-scale text data from various sources (*e.g.,* websites, books, and examinations) and different formats (*e.g.,* natural language and

code) into the CPT data, to preserve the general capabilities. We carefully filter and select training data, following the approach used in Yulan-3 (Zhu et al., 2024).

During the CPT process, it is key to explore various potential strategies for data collection, mixture, and curriculum design, akin to those used in standard pre-training (Hu et al., 2024; Abdin et al., 2024). However, considering the huge experimental cost on Llama-3 (8B), we perform surrogate experiments using a relatively small model, TinyLlama (Zhang et al., 2024). Based on TinyLlama, we extensively examine the effect of different data curation strategies, and further verify the findings in training Llama-3 (8B). To follow the nomenclature for Llama models, we refer to the continually pre-trained model in this work as **Llama-3-SynE** (Synthetic data Enhanced Llama-3).

To evaluate the effectiveness of our approach, we conduct comprehensive experiments comparing Llama-3-SynE with other competitive LLMs across various evaluation benchmarks, including general and scientific scenarios. Experimental results have shown that our data strategies significantly enhance the overall capabilities of Llama-3 (8B), particularly in Chinese language understanding and scientific knowledge reasoning. In summary, our contributions are as follows:

- We present the complete training procedure for continually pre-training Llama-3 (8B), including data selection, mixture, and curriculum. Extensive experiments show that our CPT approach is very *effective* (yielding large improvements on Chinese and scientific benchmarks without hurting the performance on English benchmarks) and *efficient* (consuming only about 100B tokens). The proposed methods and derived findings would be useful for future studies exploring various adaptation scenarios of well trained LLMs.
- We extensively explore the data synthesis technique, and generate high-quality scientific and code data. We show that these synthetic data can largely improve the corresponding capabilities of LLMs.
- We release the whole dataset utilized to continually pre-train Llama-3-SynE, including the general corpus comprising 98.5 billion tokens and synthetic data comprising 1.5 billion tokens focusing on scientific reasoning and coding tasks. Our dataset would be highly useful for training capable LLMs, which has been also evidenced by the surrogate model TinyLlama in our experiments.

2 Related Work

In this section, we review the related work in the following three aspects.

Synthetic Data The available high-quality data may not be enough for models to acquire the necessary knowledge. To address this issue, synthetic data has been widely used in the training of LLMs including general document data for pre-training (Maini et al., 2024), instruction data for supervised fine-tuning (Xu et al., 2023), and other applications. There exist two primary methods for automatic data synthesis: directly prompting LLM APIs (Xu et al., 2023; Ding et al., 2023) and training customized synthetic models (Yue et al., 2024; Zhou et al., 2024). By prompting with task instructions and suitable examplar data, capable LLMs (*e.g.,* GPT-4) can generate high-quality data, potentially injecting the knowledge that they have acquired during training. In addition, existing works also explore training relatively smaller customized models to synthesize more domain-specific data with much less API cost (Zhou et al., 2024).

Continual Pre-training Continual pre-training, also called domain adaptive pre-training (Ke et al., 2023; Jang et al., 2022; Lesort et al., 2021), has been widely used to enhance the domain-specific abilities of a pre-trained model with new domain data. It has been a long-standing research challenge to adapt models to new domains and meanwhile prevent catastrophic forgetting (French, 1999; Nguyen et al., 2019). Existing works have extensively studied fine-grained factors in mitigating catastrophic forgetting during continual pre-training, including warm-up method (Gupta et al., 2023), data distribution (Ibrahim et al., 2024; Parmar et al., 2024), and learning rate (Winata et al., 2023; Scialom et al., 2022).

Scientific Large Language Models The remarkable capabilities of LLMs have led to an increasing inclination towards their utilization in scientific application scenarios. To enhance the capacity of LLMs to comprehend and resolve scientific problems, extensive efforts have been devoted to training scientific-oriented large language models, such as mathematics LLMs (Yue et al., 2024; Shao et al., 2024; Zhou et al., 2024), biological LLMs (Jr. and Bepler, 2023; Zhang et al., 2023) and chemical LLMs (Bagal et al., 2022; Bran et al., 2024).

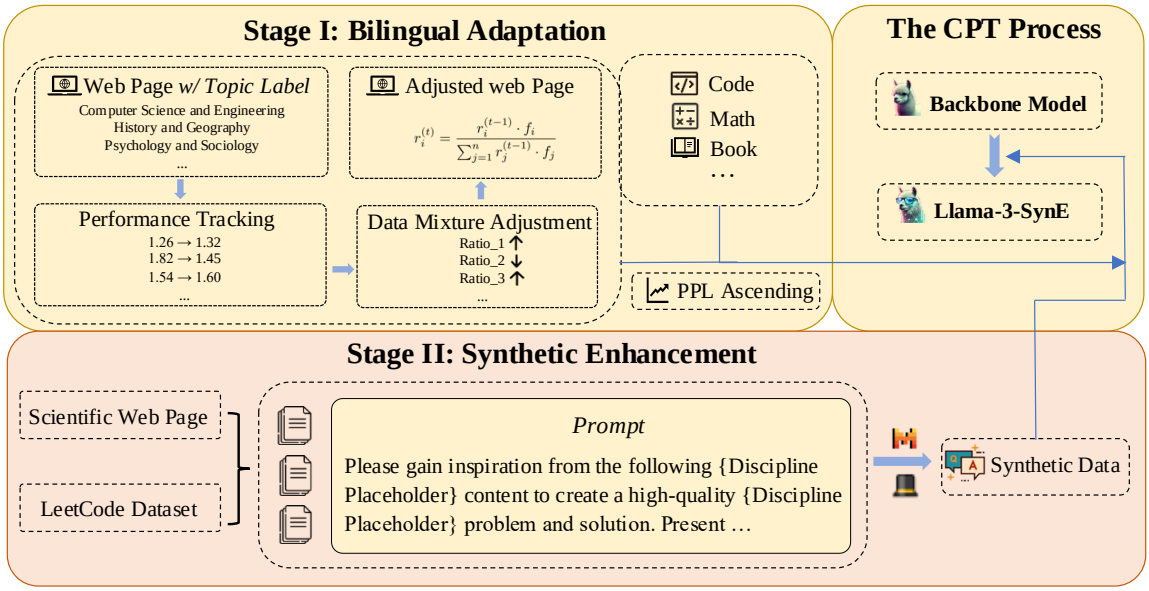


Figure 1: The overall pipeline of the CPT process.

3 Preliminary

In this section, we provide the preliminary setup for our CPT approach, focusing on two key aspects: the backbone model and the data source.

Backbone Model To conduct the research on CPT, we adopt Llama-3 (8B) (AI@Meta, 2024) as the backbone model, which has excelled in various downstream tasks such as text generation, translation, summarization, and question-answering. However, Llama-3 has been primarily pre-trained on English text data, which is inadequate in Chinese-oriented tasks. In addition, since Llama-3 was developed as a general-purpose LLM, it may also lack sufficient scientific knowledge. Considering these two limitations, we aim to improve Llama-3's Chinese capacities as well as to enhance its performance in multidisciplinary scientific tasks. It is worth noting that the proposed approach can be generally applied to other backbone models, as evidenced by our experiments on the relatively smaller model TinyLlama (Section 5.2).

Data Source The selection of data sources is key to the capacities of LLMs. To prepare the pre-training data, we mainly refer to the data configuration of Yulan-3 (Zhu et al., 2024), which collects a diverse set of data, including web pages, encyclopedias, books, question-answering (QA) forums, academic papers, mathematical corpora, code, and synthetic data. We provide detailed information about the composition of our training data in Appendix C.

4 The Proposed CPT Approach

In this section, we present the proposed continual pre-training (CPT) approach for enhancing the *Chinese* and *scientific* capabilities of LLMs. Overall, our training procedure consists of two main stages, namely *bilingual adaptation stage* and *synthetic enhancement stage*, which focus on improving Llama-3's Chinese and scientific capacities, respectively. In the CPT process, it is important to retain the original capability of Llama-3 by alleviating the effect of catastrophic forgetting. For this purpose, we design different data strategies to balance new and old abilities, which will be detailed in the following sections. The overall pipeline is shown in Figure 1.

4.1 Bilingual Adaptation Stage

We first introduce the training approach for improving the Chinese capacities of Llama-3. Following the prior work (Zhu et al., 2024), we set the ratio of Chinese and English corpora as 1:4, to balance the Chinese and English capabilities. For pre-training, effective data mixture and schedule strategies are key to improving the capacities of LLMs. Based on the overall English-Chinese ratio, we further design two strategies to enhance knowledge learning from diverse domains or sources, namely topic-based data mixture and perplexity-based data curriculum. Next, we introduce the two techniques in detail.

4.1.1 Topic-based Data Mixture

In prior work (Xie et al., 2023), data mixture is usually conducted based on datasets or data types, *e.g.,* setting a sampling distribution to sample data instances from available datasets. In our approach, we aim to explore a more fine-grained adjustment on data mixture. To achieve this goal, we consider establishing a topic taxonomy and conducting the data mixture at the topic level. Next, we present the topic-based data mixture method.

Topic Identification We train a classifier based on language models to identify the topic label (see the pre-defined topics in Appendix E) for each web page. These topics are intentionally designed to be in alignment with the subjects of the MMLU (Hendrycks et al., 2021a) and CMMLU (Li et al., 2023) benchmarks, which can also be extended to other topic taxonomies. Furthermore, we employ GPT-4 to annotate a small number of web pages as training data for our topic classifiers. Concretely, we adopt the zero-shot setting and construct the prompt by concatenating the topics and an unlabelled web page (see the prompt detail in Appendix B). Then, we utilize the instructions to guide GPT-4 to annotate the unlabelled web page by these pre-determined topic labels. In order to conduct topic classification on both Chinese and English text, we train TinyBERT[1] and BERT-Tiny-Chinese[2] as the classifiers to identify the topic labels for English and Chinese web pages, respectively. With the utilization of these classifiers, the web pages can be assigned with specific topic labels.

Performance Change Tracking To track the LLM's capabilities on different topic categories during the training process, we evaluate the change of the perplexity (PPL) score in each topic on the validation set. A reduction in the PPL score for a particular topic indicates an improvement in the model's capability regarding that topic. Concretely, supposing there are n topics, the performance change on the i-th topic is:

$$\Delta p_i = p_i^{(t)} - p_i^{(t-1)}, \quad i = 1, \ldots, n,$$

where $p_i^{(t)}$ and $p_i^{(t-1)}$ are the PPL on the i-th topic of LLM after the t-th and $(t-1)$-th rounds[3] of CPT process, respectively. The normalized performance change is then computed as:

$$\delta_{p_i} = \frac{\Delta p_i}{\max_j(|\Delta p_j|)}, \quad i = 1, \ldots, n.$$

Data Mixture Adjustment Based on the performance change, we calculate the weight adjustment coefficient f_i for training data proportions:

$$f_i = 1 + \alpha \cdot \delta_{p_i},$$

where α is a coefficient that controls the magnitude of the adjustment. After obtaining the adjustment coefficients (*i.e.,* $f_1, f_2, \ldots, f_n$), we can update the data proportions for each topic based on these coefficients. During training, let $r_i^{(t-1)}$ be the proportion of the i-th topic for the $(t-1)$-th round, then the proportion of data for the t-th round can be calculated as follows:

$$r_i^{(t)} = \frac{r_i^{(t-1)} \cdot f_i}{\sum_{j=1}^{n} r_j^{(t-1)} \cdot f_j}.$$

By using the topic-based mixture strategy, we can easily monitor the PPL change trend in a fine-grained way, and thus can better balance the abilities of LLMs across different topics or domains.

4.1.2 Perplexity-based Data Curriculum

In addition to adjusting the data mixture ratio, we also design a data curriculum strategy that organizes the training instances in a simple-to-complex manner. Curriculum learning has been demonstrated to be effective in many tasks (Bengio et al., 2009). Its primary principle is to gradually increase the difficulty (or complexity) of the training data. This strategy allows the model to establish a robust foundational knowledge base before learning more complex knowledge and skills.

Following this idea, we use the PPL score generated by the model to measure the difficulty level of the training data. Training the model on Chinese text data with a progressively increasing PPL score can provide a gradual and smooth transition in training complexity. This is particularly crucial since Llama-3 is primarily trained on a large scale of English corpora with very little Chinese

[1] https://huggingface.co/huawei-noah/TinyBERT_General_4L_312D

[2] https://huggingface.co/ckiplab/bert-tiny-chinese

[3] A round consists of several training steps, corresponding to the training of about 40B tokens.

data. Based on our preliminary experiments, starting with "simpler" Chinese data is beneficial to alleviate the performance loss (*i.e.,* catastrophic forgetting) of Llama-3 in English tasks.

4.2 Synthetic Enhancement Stage

After bilingual adaptation training, the LLM's performance on Chinese tasks can be significantly improved. In this stage, we further incorporate synthetic data to improve the multidisciplinary scientific capacities of Llama-3, inspired by prior work (Zhou et al., 2024; Jiang et al., 2024), the data ratio is correspondingly adjusted to 1:7:2 for Chinese, English, and synthetic data, respectively. Note that both the topic-based mixture strategy and perplexity-based data curriculum are no longer used in this training stage, and we randomly sample the data following the mixture proportion from the training corpus. Next, we describe our method for synthesizing data for CPT.

4.2.1 Synthesizing the Scientific QA Data

Synthetic data has been demonstrated to be effective and efficient for enhancing the capabilities of LLMs (Yu et al., 2023; Yue et al., 2023; Zhou et al., 2024). Following prior work (Zhou et al., 2024), we generate synthetic data in the format of the question and answer (QA) pair, to cover a broad spectrum of multidisciplinary scientific knowledge. The synthetic questions and answers are concatenated into text and added to the CPT training corpora.

Specifically, we consider nine scientific disciplines, *i.e.,* mathematics, physics, chemistry, biology, astronomy, earth science, medical science, computer science, and general education. For each discipline, we manually collect a list of domain names relevant to the respective fields, such as `math.stackexchange.com` and `physicsforums.com`, allowing for the expansion of this list as needed to enhance the coverage. To construct a science-related seed corpus, we collect scientific web pages from Dolma's CC (Soldaini et al., 2024) and C4 (Dodge et al., 2021) subsets that belong to the collected domain names.

Based on the above corpus, we further extract the content snippets and fill in our designed prompt template. Then, we utilize Mistral-7B-Instruct-v0.3[4] to generate relevant QA pairs that align with the targeted scientific discipline. These synthetic data are crafted to precisely mimic the structure and complexity of real-world scientific problems, which can enhance the model's capability for scientific problem understanding and reasoning.

4.2.2 Synthesizing the Code QA Data

During the preliminary experiments, we find that the coding capacities of Llama-3 are severely affected in the CPT process: sharp performance degradation is observed on the code evaluation benchmarks (*i.e.,* HumanEval and MBPP).

To retain the coding capacities of Llama-3, we adopt a similar data synthesis approach for generating high-quality code QA data. Specifically, we expand the LeetCode dataset[5] using the in-context learning (ICL) method. We randomly select problems from the LeetCode dataset as demonstrations, synthesize new coding problems, and generate answers for these problems. In implementation, we use Magicoder-S-DS-6.7B (Wei et al., 2023) for both problems and solutions synthesis.

The details of synthesis cases and the statistical information of all synthetic data for both scientific and code are provided in Appendix A and D.

Prompt for QA Synthesis

Instruction
Please gain inspiration from the following {Discipline Placeholder} content to create a high-quality {Discipline Placeholder} problem and solution. Present your output in two distinct sections: [Problem] and [Solution].

{Discipline Placeholder} Content
{Seed Snippet Placeholder}

Guidelines
[Problem]: This should be **completely self-contained**, providing all the contextual information one needs to understand and solve the problem.

[Solution]: Present a comprehensive, step-by-step solution that solves the problem **correctly** and educates the student, around 250-350 words long. Clearly articulate the reasoning and methods used at each step, providing insight into the problem-solving process. Take care to format any equations properly using LaTeX or appropriate notation.

5 Experiment

In this section, we introduce the details of experiments for evaluating our approach.

5.1 Evaluation Benchmark

To ensure a comprehensive capacity assessment, we evaluate the performance of LLMs from the following aspects. Evaluation benchmarks are divided into two groups: *major benchmarks* for overall capacity evaluation, and *scientific benchmarks* for assessing the effectiveness of our data synthesis

[4] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

[5] https://huggingface.co/datasets/greengerong/leetcode

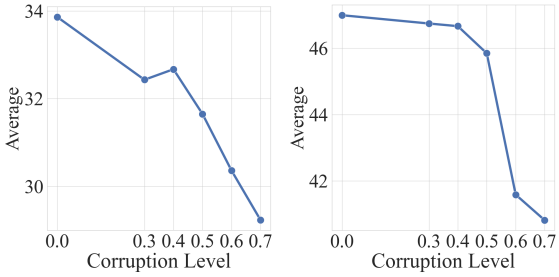
technique. Details of the benchmarks and evaluation settings are in Appendix F.

We evaluate language understanding using MMLU (Hendrycks et al., 2021a) for English and CMMLU (Li et al., 2023) and C-Eval (Huang et al., 2023) for Chinese. Coding proficiency is assessed using HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021). Scientific reasoning is evaluated on SciQ (Welbl et al., 2017), SciEval (Sun et al., 2024), and ARC (Clark et al., 2018) for English science, SAT-Math (Zhong et al., 2023), MATH (Hendrycks et al., 2021b), GSM8K (Cobbe et al., 2021), AQUA-RAT (Ling et al., 2017), MAWPS (Koncel-Kedziorski et al., 2016), ASDiv (Miao et al., 2021) for English math, and GaoKao (Zhong et al., 2023) for Chinese physical, chemical, and mathematical reasoning.

5.2 Surrogate Experiments with TinyLlama

Due to the significant costs involved in tuning experiments on Llama-3 (8B), we use a relatively small model TinyLlama (Zhang et al., 2024) as a surrogate model for extensive exploratory experiments, and the derived findings can be employed to guide the training of Llama-3 (8B). Specifically, TinyLlama is a language model with 1.1 billion parameters, and it is pre-trained on three trillion tokens using the same architecture and tokenizer as Llama-2 (Touvron et al., 2023b), which is suitable for exploring the CPT strategies in our experiments. The implementation details of TinyLlama are similar to Llama-3 in Appendix I, with the differences being TinyLlama's fixed learning rate of 1.0×10^{-4} and a maximum context length of $2,048$ tokens. In this part, to avoid large performance discrepancies across benchmarks, for major benchmarks, we mainly select C-Eval, CMMLU, and MMLU for computing the average performance; for scientific benchmarks, we select SciEval, SciQ, and ARC for computing the average performance. We also report all benchmark results in Appendix J. Next, we introduce the detailed experiments with TinyLlama, including the impact of synthetic data quality, synthetic data curriculum and comparison with open-source datasets. We also examine the effectiveness of synthetic data and the impact of synthetic data ratio in Appendix G.

Impact of Synthetic Data Quality Intuitively, the quality (or accuracy) of synthetic data would influence the learning of domain knowledge for LLMs. However, it is difficult to guarantee the accuracy of the automatically generated synthetic data. To examine the impact of the synthetic data quality, we consider simulating multiple synthetic datasets with varied data quality. Concretely, we corrupt the original synthetic data by applying three types of transformation, including randomly replacing a number, substituting frequently occurring nouns with random hyponyms, and replacing frequently occurring adjectives with their antonyms (see Appendix H). Based on the above transformation method, we sample one billion tokens from the synthetic data and vary the level of corruption ratios at the range of $\{0.0, 0.3, 0.4, 0.5, 0.6, 0.7\}$. Then, we integrate 4B normal tokens[6] with these six synthetic datasets as the CPT dataset, and train TinyLlama for performance comparison. Figure 2 presents the average performance of TinyLlama after training with varying corruption levels. As can be seen from this figure, a low corruption level (*i.e.,* 0.3) has very little impact on the model performance, suggesting that LLMs can tolerate a certain degree of inaccuracy in synthetic data. However, it would still lead to large performance degradation with a high corruption level (*i.e.,* > 0.5).

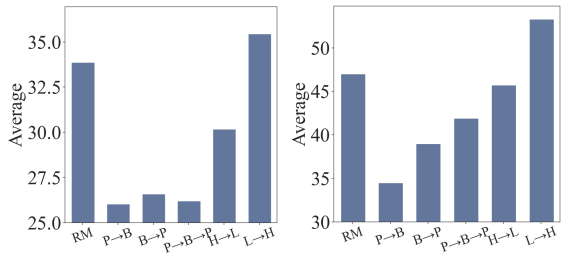


(a) Major Benchmark (b) Scientific Benchmark

Figure 2: Performance of TinyLlama continually pre-trained on varying corruption levels of synthetic data.

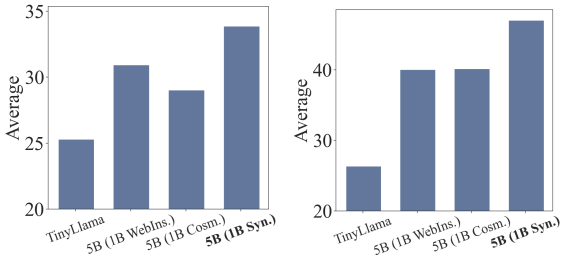
Impact of Synthetic Data Curriculum In addition to the mixture ratio, we can also set different data curriculum methods (*i.e.,* reordering the instances) for synthetic data, since it mixes data from multiple disciplines. To explore the impact of data curriculum, we consider two data instance reordering methods, either by *discipline* or *difficulty*, and compare these strategies with the random mixing strategy. For discipline, we design three kinds of curriculum methods by considering two disciplines, including *physics* $\rightarrow$ *biochemistry*, *biochemistry* $\rightarrow$ *physics* and *physics* $\rightarrow$ *biochemistry* $\rightarrow$ *physics*.

[6] In this work, "Normal token" corresponds to the non-synthetic data in the training dataset.



(a) Major Benchmark (b) Scientific Benchmark

Figure 3: Performance of TinyLlama with different data curriculum methods. "RM" refers to the random mixing strategy. "P", "B", "H", and "L" stand for "physics", "biochemistry", "high", and "low", respectively.



(a) Major Benchmark (b) Scientific Benchmark

Figure 4: Performance of TinyLlama continually pre-trained on different open-source datasets.

For difficulty, we utilize the PPL score to assess the difficulty level (ten groups in total) and consider the reordering schedules of *low* $\rightarrow$ *high* and *high* $\rightarrow$ *low*. Each data curriculum is with the same training instances but a different instance organization order. The results of the data curriculum are presented in Figure 3. Overall, we can have two major observations. Firstly, the deliberate separation of data by discipline can not bring performance improvement, even hurting the model performance. Secondly, the easy-to-difficult curriculum can lead to more performance improvement than the contrary difficult-to-easy one and random sampling, since it can help models gradually acquire more complex knowledge information. This demonstrates the effectiveness of the proposed data curriculum strategy based on PPL.

Comparison with Open-source Datasets To further examine the effectiveness of our synthetic data, we select *WebInstruct* (instruction data mined from the web in the math and science domains) (Yue et al., 2024) and *Cosmopedia* (synthetic data from the scientific subset automathtext) (Ben Allal et al., 2024), two large-scale open-source datasets that have been widely used for improving LLMs. For the fair comparison, we consider comparing four variants based on TinyLlama, including *TinyLlama* (the original model), *w/ 5B (1B Webins.)* (CPT with 4B normal tokens and 1B WebInstruct tokens), *w/ 5B (1B Cosm.)* (CPT with 4B normal tokens and 1B Cosmopedia tokens), and *w/ 5B (1B Syn.)* (CPT with 4B normal tokens and 1B tokens from our synthetic data). Figure 4 presents the performance of TinyLlama after training with different open-source datasets. The results show that our synthetic data leads to more improvements in both major and scientific benchmarks, which demonstrates the effectiveness of our data synthesis method.

5.3 Main Experiments with Llama-3

Based on the above findings from TinyLlama, we adopt the best-performing strategies or configurations for continual pre-training Llama-3. The implementation details are presented in Appendix I.

Baselines To conduct the comprehensive evaluation, we adopt both general LLMs and scientific LLMs as baselines in our experiment. We consider three kinds of LLMs as baselines, including *general-purpose LLM*, *scientific LLM* (enhanced by the science-related corpus or instructions), and *continual pre-training LLM*. For general-purpose LLMs, we adopt DCLM-7B (Li et al., 2024) and Mistral-7B-v0.3 (Jiang et al., 2023) as the baseline in the evaluation. For scientific LLMs, we adopt MAmmoTH2-8B (Yue et al., 2024) and Galactica-6.7B (Taylor et al., 2022) as the baseline LLMs. In addition, we also report the evaluation results of Llama-3-Chinese-8B[7], which has also been continually pre-trained based on Llama-3.

Results on Major Benchmarks As presented in Table 1, we can observe that Llama-3-SynE outperforms its backbone model Llama-3 (8B) by a large margin on Chinese evaluation benchmarks (*e.g.,* C-Eval and CMMLU). It shows that our approach is very effective for enhancing the Chinese language capacity of Llama-3. We carefully collect and clean the Chinese text data, and also design suitable data mixture and curriculum to adaptively retrain these models, which is the key to performance improvement on Chinese benchmarks. Second, for English evaluation benchmarks, our approach slightly underperforms Llama-3 (8B) on MMLU, while achieving improved or comparable performance on the rest math and code benchmarks.

[7] https://huggingface.co/hfl/llama-3-chinese-8b

Models	Bilingual			Math					Code		Avg.
	MMLU	C-Eval	CMMLU	MATH	GSM8K	ASDiv	MAWPS	SAT-Math	HumanEval	MBPP	
Llama-3-8B	**66.60**	49.43	51.03	16.20	54.40	72.10	89.30	38.64	36.59	**47.00**	52.13
DCLM-7B	64.01	41.24	40.89	14.10	39.20	67.10	83.40	41.36	21.95	32.60	44.58
Mistral-7B-v0.3	63.54	42.74	43.72	12.30	40.50	67.50	87.50	40.45	25.61	36.00	45.99
Llama-3-Chinese-8B	64.10	50.14	51.20	3.60	0.80	1.90	0.60	36.82	9.76	14.80	23.37
MAmmoTH2-8B	64.89	46.56	45.90	**34.10**	**61.70**	**82.80**	91.50	41.36	17.68	38.80	52.53
Galactica-6.7B	37.13	26.72	25.53	5.30	9.60	40.90	51.70	23.18	7.31	2.00	22.94
Llama-3-SynE (ours)	65.19	**58.24**	**57.34**	28.20	60.80	81.00	**94.10**	**43.64**	**42.07**	45.60	**57.62**

Table 1: Few-shot performance comparison on major benchmarks (*i.e.,* bilingual tasks, code synthesis tasks and mathematical reasoning tasks). The best and second best are in **bold** and underlined, respectively.

Models	SciEval				SciQ	GaoKao			ARC			AQUA-RAT	Avg.
	PHY	CHE	BIO	Avg.	Avg.	MathQA	CHE	BIO	Easy	Challenge	Avg.	Avg.	
Llama-3-8B	46.95	63.45	74.53	65.47	90.90	27.92	32.85	43.81	91.37	77.73	84.51	27.95	53.34
DCLM-7B	**56.71**	64.39	72.03	66.25	**92.50**	29.06	31.40	37.14	89.52	76.37	82.94	20.08	51.34
Mistral-7B-v0.3	48.17	59.41	68.89	61.51	89.40	30.48	30.92	41.43	87.33	74.74	81.04	23.23	51.14
Llama-3-Chinese-8B	48.17	67.34	73.90	67.34	89.20	27.64	30.43	38.57	88.22	70.48	79.35	27.56	51.44
MAmmoTH2-8B	49.39	**69.36**	76.83	**69.60**	90.20	**32.19**	36.23	49.05	**92.85**	**84.30**	**88.57**	27.17	56.14
Galactica-6.7B	34.76	43.39	54.07	46.27	71.50	23.65	27.05	24.76	65.91	46.76	56.33	20.87	38.63
Llama-3-SynE (ours)	53.66	67.81	**77.45**	**69.60**	91.20	31.05	**51.21**	**69.52**	91.58	80.97	86.28	**28.74**	**61.09**

Table 2: Few-shot performance comparison on scientific benchmarks. "PHY", "CHE", and "BIO" denote the physics, chemistry, and biology sub-tasks of the corresponding benchmarks.

It demonstrates that our approach can well address the catastrophic forgetting issue of the original capabilities of LLMs. Actually, based on our preliminary experiments (also evidenced by baseline models), Chinese-adaptive CPT models are difficult to retain the original performance on English-oriented benchmarks (*e.g.,* MMLU) due to the data distribution discrepancy between pre-training and CPT. These results indicate that our approach can effectively balance the original and new capacities.

Results on Scientific Benchmarks As shown in Table 2, Llama-3-SynE performs very well on the scientific benchmarks, which is consistently better than the backbone model Llama-3. It indicates that our synthetic data is very effective in improving the scientific reasoning capability of LLMs. In particular, compared to the English datasets, Llama-3-SynE achieves a significantly larger improvement on the Chinese datasets, *i.e.,* GaoKao BIO benchmark (25.71 points improvement over Llama-3), since our CPT model can effectively balance the English and Chinese reasoning abilities on scientific tasks. Among all the baselines, MAmmoTH2-8B achieves very good performance on English scientific benchmarks, while it suffers from performance degradation on general Chinese benchmarks, *e.g.,* C-Eval and CMMLU.

By combining the results on major and scientific benchmarks, we can see that Llama-3-SynE achieves very competitive performance in various abilities, and it can effectively alleviate the catastrophic forgetting issue in the CPT process. Our CPT approach only consumes about 100B tokens, which is relatively efficient in training compute.

6 Conclusion

In this work, we studied how to perform effective continual pre-training (CPT) for LLMs under a limited training budget. Our focus is to develop new capabilities and meanwhile avoid catastrophic forgetting of original capabilities. Specifically, we extensively explored the data synthesis technique, and generated high-quality scientific and code data, which can largely improve the corresponding abilities of LLMs. In order to reduce the tuning cost, we conducted extensive experiments on TinyLlama by examining various data curation strategies, including data selection, mixture, and curriculum. The derived findings were further employed to guide the training of Llama-3 (8B). Experimental results have shown that our CPT approach can largely boost the Chinese and scientific reasoning abilities of the backbone model, and meanwhile effectively retain its original abilities.

7 Limitations

Despite the promising results achieved in this study, there are several limitations that should be acknowledged. Firstly, our current efforts in developing an open recipe for continual pre-training have primarily focused on the Llama-3 (8B). To fully evaluate the applicability of our proposed continual pre-training methodology, it is crucial to extend experiments to include more LLMs and target domains. Secondly, our methodology focuses specifically on bilingual (Chinese and English) and scientific knowledge adaptation. It remains to be seen whether the proposed CPT approach can be transferred to other domains, such as law, healthcare, or arts, where domain-specific knowledge might require different strategies or datasets. Lastly, our methodology was primarily designed to augment the base model. It would be advantageous to examine the performance of the final chat model. This investigation should incorporate a more comprehensive view, considering both chat capability and human alignment.

References

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219.

AI@Meta. 2024. Llama 3 model card.

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. Program synthesis with large language models. *CoRR*, abs/2108.07732.

Viraj Bagal, Rishal Aggarwal, P. K. Vinod, and U. Deva Priyakumar. 2022. Molgpt: Molecular generation using a transformer-decoder model. *J. Chem. Inf. Model.*, 62(9):2064–2076.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Cosmopedia.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nat. Mac. Intell.*, 6(5):525–535.

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.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *CoRR*, abs/1604.06174.

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 *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, Hao Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, Tao Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, and Xiaowen Sun. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *CoRR*, abs/2405.04434.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3029–3051. Association for Computational Linguistics.

Jesse Dodge, Maarten Sap, Ana Marasovic, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 1286–1305. Association for Computational Linguistics.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L. Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. 2023. Continual pre-training of large language models: How to (re)warm your model? *CoRR*, abs/2308.04014.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zhen Leng Thai, Kai Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *CoRR*, abs/2404.06395.

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. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. 2024. Simple and scalable strategies to continually pre-train large language models. *CoRR*, abs/2403.08763.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2022. Towards continual knowledge learning of language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jinhao Jiang, Junyi Li, Wayne Xin Zhao, Yang Song, Tao Zhang, and Ji-Rong Wen. 2024. Mix-cpt: A

domain adaptation framework via decoupling knowledge learning and format alignment. *arXiv preprint arXiv:2407.10804*.

Timothy F. Truong Jr. and Tristan Bepler. 2023. Poet: A generative model of protein families as sequences-of-sequences. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1152–1157. The Association for Computational Linguistics.

Timothée Lesort, Massimo Caccia, and Irina Rish. 2021. Understanding continual learning settings with data distribution drift analysis. *CoRR*, abs/2104.01678.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023. CMMLU: measuring massive multitask language understanding in chinese. *CoRR*, abs/2306.09212.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah M. Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Raghavi Chandu, Thao Nguyen, Igor Vasiljevic, Sham M. Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. 2024. Datacomp-lm: In search of the next generation of training sets for language models. *CoRR*, abs/2406.11794.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. pages 158–167.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *CoRR*, abs/2308.08747.

Pratyush Maini, Skyler Seto, He Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. 2024. Rephrasing the web: A recipe for compute and data-efficient language modeling. *CoRR*, abs/2401.16380.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2021. A diverse corpus for evaluating and developing english math word problem solvers. *CoRR*, abs/2106.15772.

Cuong V. Nguyen, Alessandro Achille, Michael Lam, Tal Hassner, Vijay Mahadevan, and Stefano Soatto. 2019. Toward understanding catastrophic forgetting in continual learning. *CoRR*, abs/1908.01091.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Reuse, don't retrain: A recipe for continued pretraining of language models. *Preprint*, arXiv:2407.07263.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. 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 6107–6122. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. *CoRR*, abs/2402.00159.

Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. 2024. Scieval: A multi-level large language model evaluation

benchmark for scientific research. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19053–19061. AAAI Press.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *CoRR*, abs/2211.09085.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2023. Magicoder: Source code is all you need. *CoRR*, abs/2312.02120.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 94–106. Association for Computational Linguistics.

Genta Indra Winata, Lingjue Xie, Karthik Radhakrishnan, Shijie Wu, Xisen Jin, Pengxiang Cheng, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2023. Overcoming catastrophic forgetting in massively multilingual continual learning. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 768–777. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

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. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *CoRR*, abs/2409.12122.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *CoRR*, abs/2309.12284.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *CoRR*, abs/2309.05653.

Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhu Chen. 2024. Mammoth2: Scaling instructions from the web. *CoRR*, abs/2405.03548.

Daoan Zhang, Weitong Zhang, Bing He, Jianguo Zhang, Chenchen Qin, and Jianhua Yao. 2023. DNAGPT: A generalized pretrained tool for multiple DNA sequence analysis tasks. *CoRR*, abs/2307.05628.

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

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang

Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

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

Kun Zhou, Beichen Zhang, Jiapeng Wang, Zhipeng Chen, Wayne Xin Zhao, Jing Sha, Zhichao Sheng, Shijin Wang, and Ji-Rong Wen. 2024. Jiuzhang3.0: Efficiently improving mathematical reasoning by training small data synthesis models. *CoRR*, abs/2405.14365.

Yutao Zhu, Kun Zhou, Kelong Mao, Wentong Chen, Yiding Sun, Zhipeng Chen, Qian Cao, Yihan Wu, Yushuo Chen, Feng Wang, Lei Zhang, Junyi Li, Xiaolei Wang, Lei Wang, Beichen Zhang, Zican Dong, Xiaoxue Cheng, Yuhan Chen, Xinyu Tang, Yupeng Hou, Qiangqiang Ren, Xincheng Pang, Shufang Xie, Wayne Xin Zhao, Zhicheng Dou, Jiaxin Mao, Yankai Lin, Ruihua Song, Jun Xu, Xu Chen, Rui Yan, Zhewei Wei, Di Hu, Wenbing Huang, Ze-Feng Gao, Yueguo Chen, Weizheng Lu, and Ji-Rong Wen. 2024. Yulan: An open-source large language model. *CoRR*, abs/2406.19853.

A Case Study for Synthesis QA

Good Case 1

Scientific Seed Snippet Texts:
Given a system of oppositely charged layers, such as a double layer of ions in an electrolyte solution, explain why these layers do not combine (neutralize) with each other.
Synthesis QA:
[Problem] Given a system of oppositely charged layers, such as a double layer of ions in an electrolyte solution, explain why these layers do not combine (neutralize) with each other.

[Solution] In a system of oppositely charged layers, such as a double layer of ions in an electrolyte solution, the layers do not combine (neutralize) due to the electrostatic repulsion between the ions. This repulsion arises from the Coulombic force, which is a fundamental force in nature that acts between charged particles.

To understand this, let's consider a simple example of a positively charged cation (e.g., Na+) and a negatively charged anion (e.g., Cl-). When these ions are brought close together, they experience an electrostatic force that pushes them apart. This force can be calculated using Coulomb's law, which states that the electrostatic force (F) between two point charges (q1 and q2) separated by a distance (r) is given by:$F = k * (q1 * q2)/r^2$, where k is the Coulomb constant (approximately $8.99 \times 10^9 Nm^2C^{-2}$).

In the case of an electrolyte solution, the ions are surrounded by a cloud of counter-ions (ions of opposite charge) that neutralize their charge locally. This cloud of counter-ions creates an electric double layer around each ion, which prevents the oppositely charged ions from coming too close to each other and neutralizing. The repulsion between these double layers is known as the electrostatic double layer repulsion.

The thickness of the double layer is typically on the order of a few angstroms (Å), and the strength of the repulsion decreases rapidly as the distance between the layers increases. This repulsion is responsible for the stability of colloidal suspensions, the behavior of charged surfaces in contact with electrolyte solutions, and many other phenomena in surface chemistry and electrochemistry.

In summary, the electrostatic repulsion between oppositely charged layers in a system, such as the double layer of ions in an electrolyte solution, prevents the layers from combining (neutralizing) with each other due to the Coulombic force. This repulsion arises from the electric double layer around each ion, which is created by the counter-ions that neutralize the charge locally. The thickness of the double layer and the strength of the repulsion depend on factors such as the ionic strength of the solution, the surface charge density, and the dielectric constant of the medium.

Good Case 2

Synthesis QA:
[Problem] Given a list of n numbers nums and an integer s, return the minimum length of a subarray which when elements of the subarray are summed together is equal to or greater than s. If no such subarray exists return 0.
Constraints:
$1 \leq$ nums.length $\leq 10^5$
$1 \leq$ nums[i] $\leq 10^5$
$1 \leq s \leq 10^9$
All the integers in nums list sum up to less than or equal to 10^9.
[Solution]

```
def minSubArrayLen(s, nums):
    # Initialize pointers and variables
    start = 0
    current_sum = 0
    min_length = float('inf')

    # Traverse the array with 'end' pointer
    for end in range(len(nums)):
        current_sum += nums[end]
        while current_sum >= s:
          min_length = min(min_length, end - start + 1)
            current_sum -= nums[start]
            start += 1

    # If min_length was updated, return it
  return min_length if min_length != float('inf') else 0
```

B Prompt

Prompt for Topic Labeling

I am categorizing a series of articles according to the following 11 topics. Next, I will give you an article, please select only one topic that the article is the most related to:

[Topics]: {Topic List Placeholder}

[Article]: {Web Page Content Placeholder}

Please only return the most related topic:

C Detailed Information of Training Data

Dataset	English	Chinese	Volume
Web Pages	✓	✓	45.18B
Encyclopedia	✓	✓	4.92B
Books	✓	✓	15.74B
QA Forums	✓	✓	4.92B
Academic Papers	✓	×	7.93B
Mathematical Corpora	✓	×	7.93B
Code	✓	×	11.88B
Synthetic Data	✓	×	1.50B
Total	-	-	100.00B

Table 3: Statistical information of the training corpus for training Llama-3-SynE.

D Statistical Information of Synthetic Data

Category	Discipline	Num. Synthetic Data
Scientific	Mathematics	207,448
	Physics	241,516
	Chemistry	30,838
	Biology	25,103
	Astronomy	24,060
	Earth Science	7,936
	Medical Science	8,199
	Computer Science	475,566
	General Education	572,478
Code	-	1,385,696

Table 4: The statistical information of the synthetic data of each discipline (in the form of QA pairs).

E Pre-defined Topics for Web Pages

Language	Topic
English	Mathematics and Physics
	Computer Science and Engineering
	Biology and Chemistry
	History and Geography
	Law and Policy
	Philosophy and Logic
	Economics and Business
	Psychology and Sociology
	Security and International Relations
	Medicine and Health
	Others
Chinese	Biology and Chemistry
	Computer Science and Engineering
	Economics and Business
	History and Geography
	Law and Policy
	Mathematics and Physics
	Medicine and Health
	Philosophy Arts and Culture
	Project and Practical Management
	Psychology Sociology and Education
	Others

Table 5: The pre-defined topics (category labels) for English and Chinese web pages, based on MMLU and CMMLU respectively.

F Benchmark Details and Settings

Here we introduce the details of the benchmarks and evaluation settings.

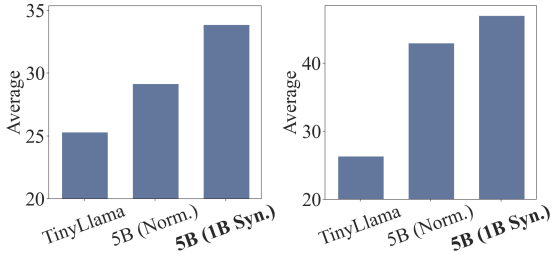
• *Language Understanding*: We evaluate the English language understanding capability using the MMLU (Hendrycks et al., 2021a), and select CMMLU (Li et al., 2023) and C-Eval (Huang et al., 2023) for evaluating Chinese language understanding capability.

• *Coding Proficiency*: We evaluate the coding proficiency using the HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) benchmarks, which measure the ability to generate correct code snippets based on given problems.

• *Scientific Reasoning*: We evaluate it using several English and Chinese datasets from science and math domains, where SciQ (Welbl et al., 2017), SciEval (Sun et al., 2024), ARC (Clark et al., 2018) are English science reasoning datasets; SAT-Math (Zhong et al., 2023), MATH (Hendrycks et al., 2021b), GSM8K (Cobbe et al., 2021), AQUA-RAT (Ling et al., 2017), MAWPS (Koncel-Kedziorski et al., 2016), ASDiv (Miao et al., 2021) are English math reasoning datasets; GaoKao (Zhong et al., 2023) is a Chinese benchmark including physical, chemical and mathematical reasoning subtasks.

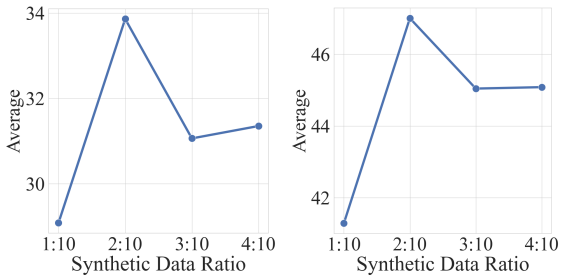
In order to better organize the evaluation results, we divide the evaluation benchmarks into two groups. The first group is *major benchmarks*, which aim to evaluate the comprehensive capacities of LLMs. The second group is *scientific benchmarks*. These benchmarks have a broader coverage of multidisciplinary scientific knowledge, and they are used for evaluating the effectiveness of our data synthesis technique.

The *major benchmarks* contain MMLU, C-Eval,



(a) Major Benchmark (b) Scientific Benchmark

Figure 5: Performance of TinyLlama continually pre-trained on different corpora.



(a) Major Benchmark (b) Scientific Benchmark

Figure 6: Performance of TinyLlama after training with different ratios of synthetic data.

CMMLU, MATH, GSM8K, ASDiv, MAWPS, SAT-Math, HumanEval, and MBPP. Note that we include commonly used math and code benchmarks in this group because it is standard practice to use these benchmarks for evaluating various general-purpose LLMs. The *scientific benchmarks* contain SciEval, SciQ, GaoKao, ARC, and AQUA-RAT.

For all the above evaluation benchmarks, we evaluate all the models using the few-shot or zero-shot settings. Specifically, we report the eight-shot performance on GSM8K, ASDiv, and MAWPS, five-shot for C-Eval, CMMLU, MMLU, MATH, GaoKao, SciQ, SciEval, SAT-Math, and AQUA-RAT, three-shot for MBPP. For HumanEval and ARC, we report the zero-shot evaluation performance.

G Additions to Surrogate Experiments with TinyLlama

We introduce the additions to surrogate experiments with TinyLlama, including the effectiveness of synthetic data and synthetic data ratio.

Effectiveness of Synthetic Data To analyze the effectiveness of our CPT approach with synthetic data, we consider comparing three variants based on TinyLlama, including *TinyLlama* (the original model), *w/ 5B (Norm.)* (CPT with 5B normal tokens), and *w/ 5B (1B Syn.)* (CPT with 4B normal tokens and 1B synthetic tokens). In our surrogate experiments, normal training tokens are constructed by using the strategies presented in Section 4.1. The results are presented in Figure 5. First, by comparing with the base TinyLlama, the two variants achieve much better average performance on both major and scientific benchmarks, indicating the effectiveness our CPT data (both the collected and synthetic data). Furthermore, TinyLlama *w/ 5B (1B Syn.)* outperforms TinyLlama *w/ 5B (Norm)*, which can demonstrate the effectiveness of our synthetic data. Since the synthetic data is derived based on the original content of web pages, it can better extract the key knowledge of text documents and reduce the influence of irrelevant contents. Furthermore, these synthetic data are presented in the form of QA pairs, having a more similar data format with downstream tasks, which is also an important factor for performance improvement.

Impact of Synthetic Data Ratio For constructing the CPT dataset, we need to determine the proportion of synthetic data in the overall data distribution. To investigate the effect of the mixture ratio, we vary the proportion of synthetic data in the training corpus, considering four choices in $\{0.1, 0.2, 0.3, 0.4\}$, and construct a 5B-token dataset to train TinyLlama. The relative ratios of the rest data sources are kept as that in Section 4.1. Figure 6 presents the average performance of TinyLlama after training with different ratios of synthetic data. We can see that the model's performance initially improves with the increasing of synthetic data proportions, then declines once the proportion reaches a relatively high value (*e.g.,*40%). Overall, a mixture ratio of 20% is a good choice for integrating synthetic data and normal data.

H Example for Accuracy Degradation Transformations

Before Transformations: *In the given chemical reaction, we have sodium (Na) reacting with chlorine (Cl2) to form sodium chloride (NaCl). To determine the number of atoms of chlorine before and after the reaction, we will first count the number of chlorine atoms... adjust the coefficients of the reactants to make the number of chlorine atoms equal before and after the reaction:2Na + Cl2 == 2NaCl.*

After Transformations: *In the given chemical reaction, we have sodium (Na) reacting with*

oxygen (Cl2) to form sodium chloride (NaCl). To determine the number of atoms of oxygen before and after the reaction, we will first count the number of oxygen atoms... adjust the coefficients of the reactants to make the number of oxygen atoms unequal before and after the reaction:6Na + Cl3 == 8NaCl

In this example, "chlorine" is replaced with a random hyponym (oxygen, hydrogen, neon, etc.) of its hypernym (chemical element), the numbers in the chemical formulas are randomly replaced, and the adjective "equal" is replaced with "unequal."

I Implementation Details of the CPT Process for Llama-3

In the topic-based data mixture strategy, we annotated $5,000$ web pages, which is enough to train a traditional classifier. n is set to 11 and α is set to 0.4. $r_i^{(0)}$ is set to the original ratio of the i-th topic.

We utilize the huggingface Transformers (Wolf et al., 2019) to implement our experiments, using Flash Attention (Dao et al., 2022) and DeepSpeed ZeRO Stage 2 to optimize the training efficiency. We employ AdamW optimizer (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$ and $\beta_2 = 0.95$, and use the Warmup-Stable-Decay (WSD) learning rate scheduler (Hu et al., 2024) in the CPT process of Llama-3. For model warmup, we linearly increase the learning rate from 1.0×10^{-7} to 1.0×10^{-5} with 10B tokens. In the remaining training procedure, the learning rate remains constant at 1.0×10^{-5}.

We conduct the CPT process using BFloat16 mixed precision, with a gradient clipping of 1.0 to ensure training stability. To enhance computational efficiency, we apply gradient checkpointing strategy (Chen et al., 2016). During training, the maximum context length is $8,192$ tokens for Llama-3.

J Detailed Surrogate Experiment Results

When introducing surrogate experiments with TinyLlama in Section 5.2, we select several representative benchmarks for computing the average performance to avoid large performance discrepancies across benchmarks. Here we report all benchmark results from Table 6 to 15. "PHY", "CHE", and "BIO" denote the physics, chemistry, and biology sub-tasks of the corresponding benchmarks. The best and second best are in **bold** and underlined, respectively.

Models	Bilingual			Math					Code	
	MMLU	C-Eval	CMMLU	MATH	GSM8K	ASDiv	MAWPS	SAT-Math	HumanEval	MBPP
TinyLlama	25.70	25.11	25.09	2.80	3.00	18.00	20.30	23.64	**10.37**	**13.40**
w/ 5B (1B Norm.)	28.35	30.02	29.10	2.90	2.00	21.00	31.40	**24.09**	4.88	4.60
w/ 5B (1B Syn.)	**31.89**	**34.60**	**35.09**	**5.30**	**14.90**	**48.10**	**66.40**	23.65	9.15	6.80

Table 6: Few-shot performance of TinyLlama continually pre-trained on different corpora on major benchmarks.

Models	SciEval				SciQ	GaoKao			ARC			AQUA-RAT
	PHY	CHE	BIO	Avg.	Avg.	MathQA	CHE	BIO	Easy	Challenge	Avg.	Avg.
TinyLlama	26.22	27.22	31.94	28.85	24.60	22.79	**27.05**	20.00	24.87	26.19	25.53	22.05
w/ 5B (1B Norm.)	28.32	35.64	45.62	38.64	56.10	26.50	**27.05**	**30.48**	37.75	30.55	34.15	**24.02**
w/ 5B (1B Syn.)	**31.10**	**38.26**	**47.81**	**40.90**	**60.30**	**27.35**	**27.05**	29.52	**45.45**	**34.13**	**39.79**	20.87

Table 7: Few-shot performance of TinyLlama continually pre-trained on different corpora on scientific benchmarks.

Models	Bilingual			Math					Code	
	MMLU	C-Eval	CMMLU	MATH	GSM8K	ASDiv	MAWPS	SAT-Math	HumanEval	MBPP
TinyLlama	25.70	25.11	25.09	2.80	3.00	18.00	20.30	23.64	10.37	**13.40**
w/ 0.0	31.89	**34.60**	**35.09**	**5.30**	14.90	48.10	66.40	23.64	9.15	6.80
w/ 0.3	31.28	31.94	34.08	**5.30**	15.50	**49.00**	65.60	**24.55**	**10.98**	7.60
w/ 0.4	**32.54**	31.67	33.79	4.60	10.50	37.50	57.50	23.64	9.15	8.60
w/ 0.5	30.23	31.27	33.44	4.90	15.80	47.60	64.90	22.73	**10.98**	8.60
w/ 0.6	28.22	29.87	33.00	4.60	**16.90**	47.90	**67.40**	23.18	8.54	9.60
w/ 0.7	27.65	27.73	32.30	4.80	1.00	4.50	3.70	24.09	9.76	8.80

Table 8: Few-shot performance of TinyLlama continually pre-trained on varying corruption levels of synthetic data on major benchmarks.

Models	SciEval				SciQ	GaoKao			ARC			AQUA-RAT
	PHY	CHE	BIO	Avg.	Avg.	MathQA	CHE	BIO	Easy	Challenge	Avg.	Avg.
TinyLlama	26.22	27.22	31.94	28.85	24.60	22.79	27.05	20.00	24.87	26.19	25.53	**22.05**
w/ 0.0	31.10	38.26	47.81	40.90	60.30	**27.35**	27.05	29.52	**45.45**	34.13	39.79	20.87
w/ 0.3	36.59	37.64	**48.23**	41.45	**60.80**	22.79	27.05	21.43	43.06	32.94	38.00	21.26
w/ 0.4	**38.41**	**39.19**	46.76	**41.91**	57.20	23.36	22.22	27.14	45.37	**36.43**	**40.90**	19.69
w/ 0.5	34.15	37.79	43.01	39.27	58.10	23.36	**27.54**	**32.86**	44.95	35.41	40.18	20.47
w/ 0.6	34.15	35.46	44.26	38.57	50.10	22.51	26.09	26.67	40.91	31.23	36.07	17.32
w/ 0.7	33.54	31.88	43.63	36.47	50.50	22.51	26.57	24.29	40.57	30.38	35.47	18.11

Table 9: Few-shot performance of TinyLlama continually pre-trained on varying corruption levels of synthetic data on scientific benchmarks.

Models	Bilingual			Math					Code	
	MMLU	C-Eval	CMMLU	MATH	GSM8K	ASDiv	MAWPS	SAT-Math	HumanEval	MBPP
TinyLlama	25.70	25.11	25.09	2.80	3.00	18.00	20.30	23.64	10.37	**13.40**
w/ 1:10	25.73	28.58	32.94	4.90	5.20	9.40	16.10	**27.27**	8.54	8.20
w/ 2:10	**31.89**	**34.60**	**35.09**	5.30	14.90	48.10	**66.40**	23.64	9.15	6.80
w/ 3:10	27.62	32.25	33.31	**6.60**	2.20	20.90	30.10	22.73	**10.98**	8.60
w/ 4:10	30.25	29.43	34.36	5.60	**15.50**	**50.40**	64.90	22.60	7.32	8.40

Table 10: Few-shot performance of TinyLlama after training with different ratios of synthetic data on major benchmarks.

Models	SciEval				SciQ	GaoKao			ARC			AQUA-RAT
	PHY	CHE	BIO	Avg.	Avg.	MathQA	CHE	BIO	Easy	Challenge	Avg.	Avg.
TinyLlama	26.22	27.22	31.94	28.85	24.60	22.79	**27.05**	20.00	24.87	26.19	25.53	**22.05**
w/ 1:10	**36.59**	34.53	42.17	37.64	50.10	22.79	**27.05**	24.76	39.69	32.59	36.14	19.69
w/ 2:10	31.10	**38.26**	**47.81**	**40.90**	**60.30**	**27.35**	**27.05**	**29.52**	45.45	34.13	**39.79**	20.87
w/ 3:10	27.80	37.79	46.35	37.98	58.00	22.79	26.57	21.43	44.57	33.70	39.14	21.65
w/ 4:10	29.88	36.39	43.84	38.34	57.20	22.79	**27.05**	20.00	**48.57**	**36.86**	39.71	19.04

Table 11: Few-shot performance of TinyLlama after training with different ratios of synthetic data on scientific benchmarks.

Models	Bilingual			Math					Code	
	MMLU	C-Eval	CMMLU	MATH	GSM8K	ASDiv	MAWPS	SAT-Math	HumanEval	MBPP
TinyLlama	25.70	25.11	25.09	2.80	3.00	18.00	20.30	23.64	10.37	**13.40**
w/ RM	31.89	34.60	35.09	5.30	14.90	48.10	66.40	23.65	9.15	6.80
w/ PB	26.78	23.73	27.58	3.50	6.10	36.60	45.50	24.09	6.71	7.80
w/ BP	26.98	24.14	28.63	3.80	5.00	32.20	43.40	23.18	6.71	8.00
w/ PBP	26.86	24.15	27.59	2.90	7.00	36.30	46.20	24.55	6.10	6.20
w/ HL	27.78	30.49	32.24	4.10	10.50	38.80	58.30	25.91	8.54	11.20
w/ LH	**32.16**	**36.89**	**37.27**	**6.10**	**20.60**	**53.90**	**70.80**	**26.36**	**12.80**	8.80

Table 12: Few-shot performance of TinyLlama with different data curriculum methods on major benchmarks.

Models	SciEval				SciQ	GaoKao			ARC			AQUA-RAT
	PHY	CHE	BIO	Avg.	Avg.	MathQA	CHE	BIO	Easy	Challenge	Avg.	Avg.
TinyLlama	26.22	27.22	31.94	28.85	24.60	22.79	**27.05**	20.00	24.87	26.19	25.53	22.05
w/ RM	31.10	38.26	47.81	40.90	60.30	27.35	**27.05**	29.52	45.45	34.13	39.79	20.87
w/ PB	32.32	32.04	41.54	35.61	35.10	**29.34**	26.57	31.90	36.74	28.75	32.75	**25.20**
w/ BP	31.10	33.90	42.59	36.78	46.90	22.51	23.19	29.52	36.15	30.29	33.22	24.02
w/ PBP	30.49	34.53	41.96	36.78	49.60	27.35	24.64	**32.86**	45.88	32.68	39.28	20.08
w/ HL	32.93	34.06	43.84	37.56	55.20	22.51	26.57	31.43	50.72	38.14	44.43	23.23
w/ LH	**37.20**	**41.84**	**51.15**	**44.71**	**65.50**	25.07	26.09	22.38	**57.62**	**41.81**	**49.71**	18.50

Table 13: Few-shot performance of TinyLlama with different data curriculum methods on scientific benchmarks.

Models	Bilingual			Math					Code	
	MMLU	C-Eval	CMMLU	MATH	GSM8K	ASDiv	MAWPS	SAT-Math	HumanEval	MBPP
TinyLlama	25.70	25.11	25.09	2.80	3.00	18.00	20.30	23.64	**10.37**	**13.40**
w/ 5B (1B WebIns.)	26.85	32.73	33.22	**7.50**	0.80	1.80	2.40	**25.00**	6.71	5.20
w/ 5B (1B Cosm.)	27.51	28.08	31.51	6.90	**19.90**	**49.70**	**68.20**	23.18	9.15	7.40
w/ 5B (1B Syn.)	**31.89**	**34.60**	**35.09**	5.30	14.90	48.10	66.40	23.64	9.15	6.80

Table 14: Few-shot performance of TinyLlama continually pre-trained on different open-source datasets on major benchmarks.

Models	SciEval				SciQ	GaoKao			ARC			AQUA-RAT
	PHY	CHE	BIO	Avg.	Avg.	MathQA	CHE	BIO	Easy	Challenge	Avg.	Avg.
TinyLlama	26.22	27.22	31.94	28.85	24.60	22.79	**27.05**	20.00	24.87	26.19	25.53	22.05
w/ 5B (1B WebIns.)	32.32	34.21	44.26	37.71	47.70	23.36	**27.05**	**31.90**	36.36	32.94	34.65	20.87
w/ 5B (1B Cosm.)	**34.76**	35.77	44.26	38.80	41.30	26.21	25.60	27.62	43.81	**36.95**	**40.38**	**22.83**
w/ 5B (1B Syn.)	31.10	**38.26**	**47.81**	**40.90**	**60.30**	**27.35**	**27.05**	29.52	**45.45**	34.13	39.79	20.87

Table 15: Few-shot performance of TinyLlama continually pre-trained on different open-source datasets on scientific benchmarks.