# MIX-CPT: A DOMAIN ADAPTATION FRAMEWORK VIA DECOUPLING KNOWLEDGE LEARNING AND FORMAT ALIGNMENT

**Jinhao Jiang[1*], Junyi Li[2*], Wayne Xin Zhao[1✉] , Yang Song[3✉], Tao Zhang[3] and Ji-Rong Wen[1]**
[1]Gaoling School of Artificial Intelligence, Renmin University of China.
[2]Department of Computer Science, National University of Singapore.
[3]Nanbeige Lab, BOSS Zhipin.
jiangjinhao@ruc.edu.cn, batmanfly@gmail.com

## ABSTRACT

Adapting large language models (LLMs) to specialized domains typically requires domain-specific corpora for continual pre-training to facilitate knowledge memorization and related instructions for fine-tuning to apply this knowledge. However, this method may lead to inefficient knowledge memorization due to a lack of awareness of knowledge utilization during the continual pre-training and demands LLMs to simultaneously learn knowledge utilization and format alignment with divergent training objectives during the fine-tuning. To enhance the domain adaptation of LLMs, we revise this process and propose a new domain adaptation framework including domain knowledge learning and general format alignment, called *Mix-CPT*. Specifically, we first conduct a knowledge mixture continual pre-training that concurrently focuses on knowledge memorization and utilization. To avoid catastrophic forgetting, we further propose a logit swap self-distillation constraint. By leveraging the knowledge and capabilities acquired during continual pre-training, we then efficiently perform instruction tuning and alignment with a few general training samples to achieve format alignment. Extensive experiments show that our proposed *Mix-CPT* framework can simultaneously improve the task-solving capabilities of LLMs on the target and general domains.

## 1 INTRODUCTION

Large Language Models (LLMs) (Zhao et al., 2023) have revolutionized the field of natural language processing (NLP) (OpenAI, 2023), showing exceptional capabilities such as instruction following and complex reasoning (Wei et al., 2022). However, due to their limited exposure to relevant data, such general LLMs still considerably lag behind in specific domains requiring professional knowledge. This situation has necessitated the effective adaptation of general-purpose LLMs to specific domains (*e.g.,* mathematics and code), called *domain adaptation* of LLMs (Guo & Yu, 2022).

In essence, tailoring general LLMs to specific domains requires adaptation in two main aspects, namely *knowledge learning* (acquiring and leveraging the necessary domain knowledge) and *format alignment* (responding to the user in an expected output form) (Jiang et al., 2024; Zhou et al., 2023; Hu et al., 2024b). Specially, knowledge learning can be further fulfilled via knowledge memorization and utilization. In practice, domain adaptation of LLMs typically involves three consecutive stages (Rozière et al., 2023; Azerbayev et al., 2023), *i.e.,* pre-training, instruction tuning, and alignment, where the first stage is primarily aimed at knowledge memorization and the other two stages are mainly focused on knowledge utilization and format alignment. However, at the pre-training stage, knowledge memorization based on raw domain-specific corpora would be somehow inefficient without eliciting the acquired knowledge according to task goals (Jiang et al., 2024). Despite that some studies incorporate instruction data for pre-training, they often rely on proprietary models to synthesize high-quality instructions at scale (Cheng et al., 2024a; Wang et al., 2024), which may

[*] Equal contribution.
[✉] Corresponding author.

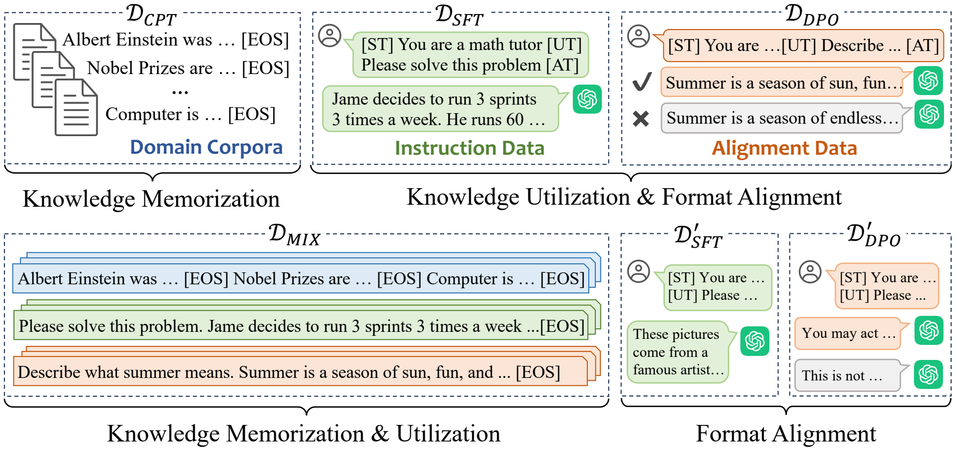
$\mathcal{D}_{CPT}$
Albert Einstein was … [EOS]
Nobel Prizes are … [EOS]
…
Computer is … [EOS]
Domain Corpora
$\mathcal{D}_{SFT}$
[ST] You are a math tutor [UT] Please solve this problem [AT]
Jame decides to run 3 sprints 3 times a week. He runs 60 …
Instruction Data
$\mathcal{D}_{DPO}$
[ST] You are …[UT] Describe ... [AT]
Summer is a season of sun, fun…
Summer is a season of endless…
Alignment Data
Knowledge Memorization
Knowledge Utilization & Format Alignment
$\mathcal{D}_{MIX}$
Albert Einstein was … [EOS] Nobel Prizes are … [EOS] Computer is … [EOS]
Please solve this problem. Jame decides to run 3 sprints 3 times a week ...[EOS]
Describe what summer means. Summer is a season of sun, fun, and ... [EOS]
$\mathcal{D}'_{SFT}$
[ST] You are … [UT] Please …
These pictures come from a famous artist…
$\mathcal{D}'_{DPO}$
[ST] You are … [UT] Please ...
You may act …
This is not …
Knowledge Memorization & Utilization
Format Alignment

Figure 1: Comparison of traditional domain adaptation approaches (top) and our proposed Mix-CPT paradigm (bottom). “[EOS]” is a special token representing the end of a document. “[ST]”, “[UT]”, and “[AT]” denote the system, user, and assistant chat template, respectively.

not be that easy without extensive fine-tuning experiences. Another issue is that learning to master knowledge utilization and format alignment in both instruction tuning and alignment stages might lead to suboptimal performance, since the two goals can be divergent in model optimization (Ren et al., 2024).

Considering the above issues, this paper explores a new domain adaptation approach that only uses raw domain-specific corpora and general instruction or alignment data. Our hypothesis is that the knowledge utilization capacity can be essentially learned from general instruction or alignment data, which has also been evidenced by prior studies (Ouyang et al., 2022b). In this way, we can remove the tedious instruction synthesis step from the training pipeline, since it is much easier to obtain general or mixed domain instruction data from open resources. Another important attempt is to enhance knowledge learning by jointly acquiring both memorization and utilization of knowledge. Therefore, we schedule all the instruction and alignment data at the pre-training stage (with a suitable format), then only reuse a minor proportion of instruction and alignment data for fine-tuning to achieve format alignment. We give a comparison in Figure 1.

Specially, our approach for domain adaptation of LLMs consists of two main stages, *i.e., domain knowledge learning* and *general format alignment*. In the first stage, we conduct knowledge mixture continual pre-training (*Mix-CPT*) to integrate both knowledge memorization and utilization. In the second stage, based on the knowledge and capabilities that are already acquired during pre-training, we perform instruction tuning and alignment in an efficient manner to achieve format alignment. For continual pre-training, we convert raw domain documents, general instructions, and alignment data into a unified format. To avoid catastrophic forgetting, we propose Logit Swap Self-Distillation (*LSSD*), which exchanges the predicted top-1 token logit with the logit of the ground-truth token, serving as the surrogate target. In this way, LSSD maintains most probabilities of the original distribution of LLMs, thereby preserving original capabilities. In instruction tuning and alignment, we propose a novel format alignment score as criterion to select a small number of instructions from the pre-training instruction set. These instructions have already been seen during pre-training, so that the model can mainly focus on pure style or format learning for downstream tasks.

We conduct experiments on domain-specific and general tasks to verify the effectiveness of our Mix-CPT method, including a total of seven distinctive capabilities based on 17 representative benchmarks. For both base LLMs and chat LLMs, our approach can effectively improve their domain-specific and general performance compared to traditional methods of first performing continual pre-training, followed by instruction tuning and alignment.

## 2 Approach

To adapt general LLMs to specific domains (*e.g.,* wiki, code), our core idea is to decouple knowledge learning and format alignment, and propose an effective two-stage domain adaptation framework,

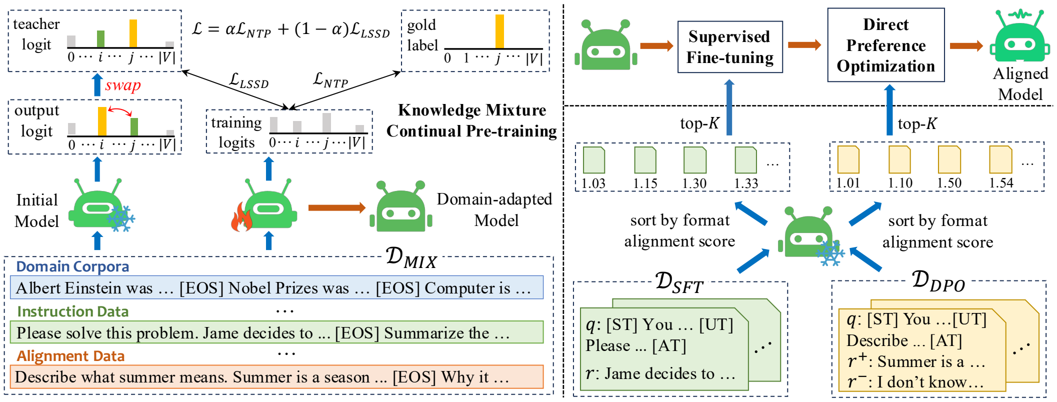
teacher logit
$0 \cdots i \cdots j \cdots |V|$
$\mathcal{L} = \alpha\mathcal{L}_{NTP} + (1-\alpha)\mathcal{L}_{LSSD}$
gold label
$0 \ \ 1 \cdots j \cdots |V|$
swap
$\mathcal{L}_{LSSD}$
$\mathcal{L}_{NTP}$
output logit
$0 \cdots i \cdots j \cdots |V|$
training logits
$0 \cdots i \cdots j \cdots |V|$
Knowledge Mixture Continual Pre-training
Initial Model
Domain-adapted Model
$\mathcal{D}_{MIX}$
Domain Corpora
Albert Einstein was … [EOS] Nobel Prizes was … [EOS] Computer is …
Instruction Data
⋯
Please solve this problem. Jame decides to ... [EOS] Summarize the …
Alignment Data
⋯
Describe what summer means. Summer is a season ... [EOS] Why it …
Supervised Fine-tuning
Direct Preference Optimization
Aligned Model
top-$K$
top-$K$
1.03
1.15
1.30
1.33
…
1.01
1.10
1.50
1.54
…
sort by format alignment score
sort by format alignment score
$\mathcal{D}_{SFT}$
$q$: [ST] You … [UT] Please ... [AT]
$r$: Jame decides to …
$\mathcal{D}_{DPO}$
$q$: [ST] You …[UT] Describe ... [AT]
$r^+$: Summer is a …
$r^-$: I don't know…

Figure 2: Overall framework of our proposed rescheduled domain adaptation paradigm, Mix-CPT. "[ST]", "[UT]", and "[AT]" denote the system, user, and assistant prompts, respectively.

*i.e.,* first performing knowledge mixture continual pre-training (Section 2.1) and then performing efficient format alignment (Section 2.2). We show the overall architecture in Figure 2.

## 2.1 Knowledge Mixture Continual Pre-training

Different from previous work that performs continual pre-training solely based on domain-specific corpora (Que et al., 2024; Ke et al., 2022), we propose to mix domain-specific documents, general instructions and alignment data as pre-training data. The QA-based instruction and alignment data is useful to reflect how the knowledge will be accessed and utilized through questions, enhancing the learning of new knowledge in domain-related documents. Besides, using general instructions can facilitate LLMs to transfer the general knowledge utilization capability to specific domains without relying on domain-related instructions in prior work (Jiang et al., 2024; Cheng et al., 2024a). Specifically, we first transform raw domain documents, general instructions, and alignment data into a unified format for continual pre-training. To avoid catastrophic forgetting, we further introduce a logit swap self-distillation (LSSD) approach during the continual pre-training process.

### 2.1.1 Unified Knowledge Format

Typically, adapting LLMs to a specific domain involves three distinct and relatively independent stages, each based on corresponding data in different formats. Specifically, the base model firstly performs continual pre-training (CPT) on domain-specific corpora for learning new knowledge, then conducts supervised fine-tuning (SFT) based on instructions for enhancing the instruction following ability, and finally utilizes the preference data for human alignment. In this work, we adopt direct preference optimization (DPO) as the alignment algorithm. Formally, we denote the domain-specific corpus as $\mathcal{D}_{\text{CPT}} = \{d_i\}_{i=1}^{n_c}$, where $d_i$ represents a raw domain document consisting of a sequence of tokens. For the instructions used in SFT, we denote as $\mathcal{D}_{\text{SFT}} = \{\langle q_i, r_i \rangle\}_{i=1}^{n_s}$, where $q_i$ and $r_i$ represent the user query and the expected response, repectively. For alignment data used in DPO, we denote by $\mathcal{D}_{\text{DPO}} = \{\langle q_i, r_i^+, r_i^- \rangle\}_{i=1}^{n_d}$, where $q_i$, $r_i^+$, and $r_i^-$ represent the user query, positive response, and negative response, respectively.

In this paper, we propose to mix $\mathcal{D}_{\text{CPT}}$, $\mathcal{D}_{\text{SFT}}$ and $\mathcal{D}_{\text{DPO}}$ into a mixture dataset $\mathcal{D}_{\text{MIX}}$ with a unified text format, building upon which we further perform knowledge mixture continual pre-training on a general LLM. Unlike previous work relying on synthesizing high-quality domain instructions (Jiang et al., 2024; Cheng et al., 2024a), we empirically find that knowledge utilization is actually a general capability that can be learned from general instructions, and we can further transfer such ability to enhance the learning of domain knowledge. Specially, we remove any templates (*e.g.,* `[UT]`) from instructions and alignment data to construct the mixture data, which consists of three kinds of samples denoted by $\mathcal{D}_{\text{MIX}} = \{x_{\text{cpt}}, x_{\text{sft}}, x_{\text{dpo}}\}$, where $x_{\text{cpt}} = d_i$ is the original domain document, $x_{\text{sft}} = [q_i; r_i]$ denotes the concatenation of user query and expected response in the instruction, and $x_{\text{dpo}} = [q_i; r_i^+]$ is the concatenation of user query and positive response in the alignment data.

Following existing pre-training methods (Touvron et al., 2023), we concatenate samples within each kind of data (*i.e.,* $\mathcal{D}_{\text{CPT}}$, $\mathcal{D}_{\text{SFT}}$ and $\mathcal{D}_{\text{DPO}}$) and truncate the sequence when reaching the maximum input length of the LLM. Besides, we add an extra special symbol (*i.e.,* `[EOS]`) at the end of each sample to separate them. We repeat this process until concatenating all samples to obtain our final knowledge mixture pre-training data $\mathcal{D}_{\text{MIX}}$.

Note that though we use general instruction data to derive the mixture data here, our approach can be generally extended to incorporating domain-specific instruction data (Cheng et al., 2024a), which often relies on specific data synthesis techniques.

### 2.1.2 Logit Swap Self-Distillation

After obtaining the unified mixture data $\mathcal{D}_{\text{MIX}} = \{x_{\text{cpt}}, x_{\text{sft}}, x_{\text{dpo}}\}$, we then perform continual pre-training on the base LLM. For simplicity, we remove the subscript of each training sample in the mixture data, denoted as $x$. We adopt the pre-training task of next token prediction (NTP). Specifically, given an input $x = \{w_1, w_2, ..., w_n\}$, we adopt the standard language modeling objective to minimize the cross-entropy loss as follows:

$$\mathcal{L}_{\text{NTP}} = -\sum_{j=1}^{n} \log \Pr(w_j | w_{<j}; \Theta), \tag{1}$$

where $w_j$ denotes the $j$-th token in the input, $w_{<j}$ is the previous tokens, and $\Theta$ denotes the model parameters. During continual pre-training, the task of next-token prediction enables the base LLM to learn domain knowledge, and the incorporation of general instructions and the alignment data further transfers the general knowledge utilization capability to specific domains.

However, the traditional language modeling objective is prone to suffer from the issue of catastrophic forgetting for previously learned knowledge of LLMs. Therefore, we propose an auxiliary training objective, *i.e., Logit Swap Self-Distillation (LSSD)*, which serves as an extra constraint for the standard language modeling objective. Specifically, we first utilize the original base LLM before continual pre-training (paramerized by $\Theta_{\text{ori}}$) to infer the output logits following the standard language modeling objective yet without computing the loss:

$$\boldsymbol{h}_j = \text{LLM}(w_{<j}; \Theta_{\text{ori}}), \tag{2}$$

$$\boldsymbol{l}_j = \boldsymbol{h}_j \boldsymbol{W}_e^T, \tag{3}$$

where $\boldsymbol{W}_e$ is the token embedding matrix, $\boldsymbol{h}_j$ is the hidden state of the last transformer block, and $\boldsymbol{l}_j$ denotes the output logit at the $j$-th position. Then, we exchange the logit value of the top-1 predicted token (*i.e.,* $\widetilde{w}_j$) and the ground-truth token (*i.e.,* $w_j$) if they are not equal:

$$\tilde{\boldsymbol{l}}_j = \text{Exchange}(\boldsymbol{l}_j, I_{\widetilde{w}_j}, I_{w_j}) \quad \text{if} \;\; I_{\widetilde{w}_j} \neq I_{w_j}, \tag{4}$$

where $I_{\widetilde{w}_j}$ and $I_{w_j}$ are indices of $\widetilde{w}_j$ and $w_j$ in the vocabulary respectively, the function Exchange$(\cdot)$ will exchange their logit values in $\boldsymbol{l}_j$. The swapped logit $\tilde{\boldsymbol{l}}_j$ will be regarded as the teacher logit in LSSD. In essence, LSSD only calibrates the prediction of ground-truth token for adapting to the current domain knowledge while maintaining most originally learned knowledge of the LLM (*i.e.,* represented by the unchanged logit values in $\tilde{\boldsymbol{l}}_j$). Next, we can compute the teacher model's probability distribution for the $j$-th token with softmax function:

$$\Pr(w_j | w_{<j}; \Theta_{\text{ori}}) = \text{softmax}(\tilde{\boldsymbol{l}}_j). \tag{5}$$

Finally, we compute the self-knowledge distillation objective and minimize the reverse Kullback-Leibler divergence loss (Gu et al., 2023) between the current model's probability distribution and the teacher model's probability distribution as follows:

$$\mathcal{L}_{\text{LSSD}} = -\sum_{j=1}^{n} \Pr(w_j | w_{<j}; \Theta) \log\left(\frac{\Pr(w_j | w_{<j}; \Theta)}{\Pr(w_j | w_{<j}; \Theta_{\text{ori}})}\right). \tag{6}$$

In the knowledge mixture continual pre-training stage, the final total loss is the combination of next token prediction loss and self-distillation loss, controlled by a co-efficient $\alpha$ as follows:

$$\mathcal{L}_{\text{CPT}} = \alpha \cdot \mathcal{L}_{\text{NTP}} + (1 - \alpha) \cdot \mathcal{L}_{\text{LSSD}}. \tag{7}$$

Due to the significant distributional differences between domain corpora and original training data, traditional approach of using one-hot label can lead to substantial model updates, which may cause catastrophic forgetting of the original knowledge after adaptation. In contrast, LSSD adopts the prediction from the original model as a surrogate label by only modifying the logit of top-1 token. This way can effectively maintain most previously learned knowledge but adapt to new domains.

### 2.2 General Format Alignment

In the domain knowledge learning stage, the LLM has simultaneously learned to memorize domain knowledge and understand how to utilize the knowledge through our proposed knowledge mixture continual pre-training. After that, during the format alignment stage, the LLM can be more *efficiently* fine-tuned to master the task format with only a small number of alignment samples. Next, we first introduce the selection of training samples and then perform efficient format alignment.

**Alignment Sample Selection.** Since we would like to decouple knowledge learning and format alignment, we aim to select training samples from $\mathcal{D}_{\text{SFT}}$ and $\mathcal{D}_{\text{DPO}}$ that have been *encountered* during continual pre-training, which can avoid introducing new knowledge in subsequent fine-tuning. The selection criterion is based on the *format alignment score (FAS)*, which estimates the difficulty of a given SFT sample $\langle q_i, r_i \rangle$ by comparing the conditional loss with and without formatted instruction:

$$\text{FAS}(\langle q_i, r_i \rangle) = \frac{\mathcal{L}_{\text{NTP}}(r_i|\hat{q}_i)}{\mathcal{L}_{\text{NTP}}(r_i)}, \tag{8}$$

where $\hat{q}_i$ denote the formatted query equipped with chat templates for interaction with humans. For a DPO sample $\langle q_i, r_i^+, r_i^- \rangle$, FAS compares the conditional losses of positive response and negative response given the formatted query:

$$\text{FAS}(\langle q_i, r_i^+, r_i^- \rangle) = \frac{\mathcal{L}_{\text{NTP}}(r_i^+|\hat{q}_i)}{\mathcal{L}_{\text{NTP}}(r_i^-|\hat{q}_i)}. \tag{9}$$

In essence, high FAS scores infer the difficulty of generating responses given a query with an interaction format and the significant disparity between positive and negative ones. Therefore, we finally select top-$K$ samples with the highest FAS scores to conduct supervised fine-tuning and direct preference optimization.

**Efficient Format Alignment.** After selecting very few training samples, we can utilize them to conduct format alignment in an efficient manner. Firstly, we utilize the selected instruction samples from $\mathcal{D}_{\text{SFT}}$ to perform supervised fine-tuning following the standard way (Ouyang et al., 2022b), which is to minimize the cross-entropy loss:

$$\mathcal{L}_{\text{SFT}} = -\sum_{j=1}^{n} \log \Pr(r_j | q, r_{<j}; \Theta), \tag{10}$$

where $r_j$ and $r_{<j}$ denote the $j$-th token and its previous tokens in the response. Secondly, we utilize the selected preference samples from $\mathcal{D}_{\text{DPO}}$ to conduct direct preference optimization following Rafailov et al. (2023b) as follows:

$$\mathcal{L}_{\text{DPO}} = -\log \sigma\Big(\beta \log \frac{\pi(r^+|q;\Theta)}{\pi(r^+|q;\Theta_{\text{ref}})} - \beta \log \frac{\pi(r^-|q;\Theta)}{\pi(r^-|q;\Theta_{\text{ref}})}\Big), \tag{11}$$

where $\sigma$ denotes the sigmoid function, $\Theta_{\text{ref}}$ denotes the parameters of the reference LLM (usually the original LLM itself), and $\pi$ denotes the product of the probabilities of all output tokens.

## 3 Experiments

### 3.1 Experimental Setup

In our experiments, we mainly focus on three popular domains for adapting general LLMs, *i.e., encyclopedia*, *mathematics*, and *code*. And select official Wikipedia dump[1], AutoMathText (Zhang et al., 2024), and StarCoder (Li et al., 2023) as the corresponding domain-specific corpus. For general instruction datasets, we choose *TULU-V2-mix* (Ivison et al., 2023) and *UltraFeedback* (Cui et al., 2023) for instruction tuning and alignment, respectively. We employ two prominent LLMs as the base model, *i.e., QWen2-7B* (Yang et al., 2024) and Meta-Llama-3-8B (Dubey et al., 2024). For comparative analysis, we consider *official chat LLMs*, *reimplemented chat LLMs*, and *continual pre-training augmented LLMs*. We evaluate seven distinctive capabilities of LLMs in a total of 17 representative NLP datasets based on the OpenCompass platform (Contributors, 2023). We show the details of our complete experimental setup in Appendix A.

[1] https://dumps.wikimedia.org/enwiki/20240301/

Table 1: Evaluation results on three specialized domains and two general examinations. The underline and **bold** fonts denote the best results in the target domain and the average results in each domain adaptation group, respectively.

| Model | | Wiki | Math | Code | English Examination | Chinese Examination | Average |
|---|---|---|---|---|---|---|---|
| LLaMA3-8B-Base | | 53.81 | 33.62 | 42.31 | 51.02 | 46.69 | 45.92 |
| *Wiki* | + CPT | 54.20 | 34.07 | 37.42 | 52.97 | 46.07 | 45.64 |
| | + Mix-CPT (*w/o KD*) | 55.81 | 38.96 | 41.62 | 52.19 | 47.44 | 47.68 |
| | + Mix-CPT | 55.47 | 37.91 | 42.37 | 52.55 | 49.93 | **47.91** |
| *Math* | + CPT | 54.14 | 34.80 | 28.95 | 50.47 | 49.02 | 43.62 |
| | + Mix-CPT (*w/o KD*) | 55.55 | 37.99 | 42.43 | 46.16 | 48.60 | 45.90 |
| | + Mix-CPT | 55.11 | 37.71 | 45.14 | 51.54 | 49.89 | **48.04** |
| *Code* | + CPT | 50.98 | 34.19 | 35.64 | 51.14 | 47.92 | 44.29 |
| | + Mix-CPT (*w/o KD*) | 54.93 | 39.38 | 42.35 | 46.31 | 47.91 | 46.02 |
| | + Mix-CPT | 55.50 | 38.15 | 42.82 | 51.55 | 48.50 | **47.61** |
| QWen2-7B-Base | | 46.37 | 62.33 | 60.02 | 46.56 | 82.90 | 56.00 |
| *Wiki* | + CPT | 47.20 | 59.48 | 29.45 | 52.32 | 81.92 | 51.12 |
| | + Mix-CPT (*w/o KD*) | 46.65 | 58.54 | 58.67 | 48.03 | 80.86 | 55.27 |
| | + Mix-CPT | 46.88 | 59.77 | 62.06 | 48.78 | 81.84 | **56.56** |
| *Math* | + CPT | 46.22 | 58.60 | 48.13 | 44.42 | 82.51 | 52.17 |
| | + Mix-CPT (*w/o KD*) | 46.19 | 59.42 | 59.93 | 46.12 | 82.70 | 55.21 |
| | + Mix-CPT | 46.42 | 60.58 | 61.40 | 45.60 | 82.25 | **55.58** |
| *Code* | + CPT | 41.37 | 60.02 | 41.93 | 42.05 | 82.97 | 49.58 |
| | + Mix-CPT (*w/o KD*) | 45.29 | 59.40 | 58.53 | 48.36 | 81.68 | 55.32 |
| | + Mix-CPT | 45.50 | 59.20 | 60.21 | 48.18 | 82.94 | **55.73** |

## 3.2 Main Results

Table 1 and Table 2 display the evaluation results for base and chat LLMs respectively, comparing our proposed Mix-CPT method and other baselines.

### 3.2.1 Results of Base LLMs

We initially assess the effectiveness of our proposed Mix-CPT framework for base LLMs in mitigating catastrophic forgetting and facilitating the knowledge learning during continual pre-training, especially the logit swap self-distillation constraint. To better observe the impact of knowledge mixture continual pre-training on the performance in target domains and general capabilities, we select domain-specific tasks and general examination tasks. We show the evaluation results in Table 1.

First, we can see that traditional continual pre-training (*i.e.,* + CPT) does not necessarily enhance the performance of base LLMs in the target domain, and may instead impair their performance therein. Besides, this method inevitably leads to a certain degree of catastrophic forgetting, thereby damaging the overall performance of the base LLM. For example, compared to the LLaMA3-8B-Base model, typical CPT leads to observed improvements in math (*i.e.,* 33.62 → 34.80) but diminishes overall performance (*i.e.,* 45.92 → 43.62). The phenomenon has been reported in existing work (Lin et al., 2024). We notice that the performance of the QWen2-7B-Base in math and code domains declines after undergoing either traditional continual pre-training or Mix-CPT. We speculate that this may be due to the significant discrepancy between the distribution of the continual pre-training data and the data used in the final stage of original pre-training (*e.g.,* annealing) (Yang et al., 2024; Dubey et al., 2024). Besides, the model might have already been trained on these two datasets, and further training leads to overfitting. For example, the typical CPT method shows a decline in the math domain (*i.e.,* 62.33 → 58.60) and the overall result (*i.e.,* 56.00 → 52.17). By contrast, Mix-CPT significantly alleviates such a decline and better maintains general performance (*i.e.,* 56.00 → 55.58). This demonstrates that our framework can effectively mitigate the impact of knowledge forgetting caused by traditional incremental training processes. This is highly beneficial in real-world scenarios where the training data proportion and distribution of the model are usually unknown.

Table 2: Evaluation results on three specialized domains and four general capabilities (*i.e.,* Reading Comprehension, Complex Reasoning, EXamination, and Instruction Following). The "CSD" denotes the typical domain adaptation method of conducting CPT, SFT, and DPO. The underline and **bold** fonts denote the same meaning as Table 1.

| Model | | | Specialized Domain | | | General Domain | | | | Average |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Wiki | Math | Code | RC | CR | EX | IF | |
| LLaMA3 8B-Base | Official Chat | | 46.97 | 50.88 | 63.29 | 77.03 | 78.21 | 62.02 | 80.40 | 64.96 |
| | Reimplemented Chat | | 33.72 | 30.88 | 41.96 | 74.19 | 77.42 | 57.62 | 69.55 | 55.86 |
| | *Wiki* | + CSD | 28.27 | 30.67 | 43.47 | 72.61 | 73.74 | 55.82 | 66.51 | 53.82 |
| | | + Mix-CPT (ours) | 33.21 | 30.21 | 47.22 | 72.60 | 71.36 | 58.65 | 68.88 | **55.25** |
| | *Math* | + CSD | 27.88 | 39.48 | 50.95 | 74.89 | 76.90 | 60.90 | 69.89 | 58.16 |
| | | + Mix-CPT (ours) | 32.12 | 39.40 | 51.21 | 73.36 | 77.00 | 60.68 | 71.81 | **58.61** |
| | *Code* | + CSD | 27.64 | 39.27 | 51.75 | 75.04 | 77.44 | 62.34 | 70.88 | **58.75** |
| | | + Mix-CPT (ours) | 31.98 | 38.30 | 52.67 | 73.33 | 74.86 | 60.71 | 72.11 | 58.25 |
| QWen2 7B-Base | Official Chat | | 37.39 | 67.05 | 72.31 | 85.79 | 82.08 | 73.48 | 82.63 | 71.74 |
| | Reimplemented Chat | | 28.10 | 58.12 | 61.22 | 84.35 | 81.86 | 71.08 | 75.40 | 66.80 |
| | *Wiki* | + CSD | 30.29 | 56.75 | 61.78 | 82.61 | 81.66 | 71.23 | 75.92 | 66.79 |
| | | + Mix-CPT (ours) | 34.55 | 56.74 | 64.66 | 83.93 | 80.61 | 72.66 | 74.73 | **67.94** |
| | *Math* | + CSD | 28.58 | 56.78 | 60.85 | 83.11 | 81.83 | 70.79 | 75.87 | 66.45 |
| | | + Mix-CPT (ours) | 33.81 | 57.48 | 67.38 | 83.29 | 80.88 | 72.44 | 76.27 | **68.29** |
| | *Code* | + CSD | 29.41 | 56.87 | 62.74 | 83.58 | 81.77 | 70.94 | 76.79 | 66.94 |
| | | + Mix-CPT (ours) | 36.05 | 55.07 | 64.84 | 83.42 | 80.57 | 73.65 | 76.77 | **68.24** |

Second, when mixing the domain raw data with the additional instructions and alignment data (*i.e.,* Mix-CPT w/o KD), the domain-specific capability can be further improved, which indicates that the mixed instruction data can benefit the learning of the domain knowledge during continual pre-training. At the same time, it can mitigate the effect of other general capabilities and reduce the degradation of the overall performance. For example, compared to the traditional CPT, mixing domain corpus with additional instruction data (Mix-CPT w/o KD) can almost consistently improve the domain capability and overall average performance (*i.e.,* 41.93 → 58.53 and 49.58 → 55.32 for the QWen2-7B-Base model in the target Code domain and overall average performance respectively).

Finally, through applying the logit swap self-distillation strategy to the knowledge mixture continual pre-training process (*i.e.,* Mix-CPT), we can further reduce the impact on the pre-learned knowledge while maintaining the domain capability improvement, thereby mitigating the degradation of the general capabilities of LLMs. Therefore, these results demonstrate that the Mix-CPT framework with the logit swap self-distillation constraint can indeed promote knowledge learning and alleviate the issue of catastrophic forgetting to some extent.

### 3.2.2 Results of Chat LLM

Subsequently, we assess the performance of final chat LLMs after instruction tuning and alignment for achieving format alignment. We show the evaluation results in Table 2.

First, the typical domain adaptation method of conducting CPT, SFT, and DPO (called CSD) faces challenges in simultaneously enhancing domains-specific capabilities while preserving general capabilities, in contrast to reimplemented chat models that do not utilize domain-specific raw data. For example, compared to reproduced LLaMA3-8B chat model, traditional CSD method not only decreases the factual question answering performance in Wiki (*i.e.,* 33.72 → 28.27) but also hurts the average performance (*i.e.,* 55.86 → 53.82). This phenomenon indicates that conventional domain adaptation methods may cause catastrophic forgetting and merely focus on knowledge memorization without considering how to utilize them, which might suffer from a memorization trap.

Second, with the same instruction data and alignment data, our method can successfully improve the performance on the target domain while maintaining the general capability compared to the reimplemented models. Compared to the reimplemented chat LLM with large-scale closed-source pre-training data but using the same instruction and alignment data, our proposed method can achieve

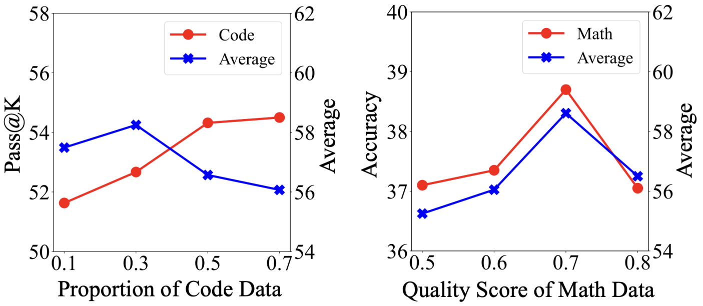

Figure 3: **(Left)** Pass@K and Average results *w.r.t.* Proportion of code data. **(Right)** Accuracy and Average results *w.r.t.* Quality score of math data.

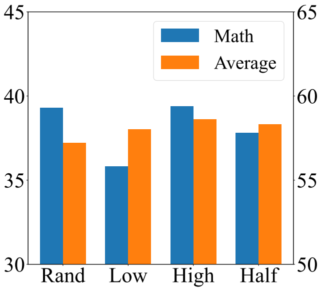

Figure 4: Accuracy on math and Average results *w.r.t.* selection strategy.

better performance on the target domain (*e.g.,* 41.96 → 52.67 in the code domain for LLaMA3-8B-Base), which indicates the effectiveness of our method.

Finally, our proposed Mix-CPT method can simultaneously improve the performance of the target domains and the general capability. The main reasons are two fold. On one hand, based on the logit swap self-knowledge distillation constraint, the LLM can effectively memorize the raw domain data while maintaining its originally learned knowledge. On the other hand, by mixing the raw domain data with the general instructions and alignment data (removing any templates), the model can learn the general knowledge utilization capability and transfer to specific domains. In this way, the model can perform efficient format alignment with only a few formatted samples to better utilize both target domain knowledge and other general knowledge.

### 3.3 Detailed Analysis

In this section, we conduct a detailed analysis of the proposed method with LLaMA3-8B, using the same benchmarks as those presented in Table 2.

**Effect of Quantity and Quality of Raw Domain Data.** We first utilize the StarCoder dataset to explore the effect of the number of domain documents on domain-specific and general capabilities with consistent quality. Then, we adopt the AutoMathText dataset to explore the effect of the quality of raw domain data on LLMs' performance by leveraging the annotated quality scores. Specifically, we conduct two group experiments as follows:

- *Proportion of Code Data*: This group aims to compare the variants using different proportions of the StarCoder, including 10%, 30%, 50%, and 70%, while maintaining constancy in other variables.

- *Quality of Math Data*: This group aims to compare variants by employing various AutoMathText, each characterized by distinct quality scores with thresholds exceeding 0.5, 0.6, 0.7, and 0.8 respectively, while maintaining constancy in other variables.

We show the results in Figure 3. We can see that increasing the amount of raw domain documents can indeed further enhance the target domain performance under the same quality. However, even though using a larger number of raw domain data (*e.g.,* more math texts with $\geq 0.5$ score than those with $\geq 0.7$ score), the low quality of raw data can also decrease the LLMs' performance regardless of the target domain or general domain, which indicates the quality of domain data is a priority over its amount when performing continual pre-training. The final decline is due to that there are much less math texts with $\geq 0.8$ score (*i.e.,* only $10\%$ of math texts with $\geq 0.7$ score).

**Effect of Format Alignment Data Selection.** We conduct a further ablation study to explore the impact of different sample selection strategies for format alignment data, which consists of the difficulty and amount of selected data. Specifically, we conduct two groups of experiments including:

- *Difficulty of Samples for SFT and DPO*: This group compares four distinct selection strategies: random selection (Rand), easiest samples with the lowest FAS scores (Low), hardest samples with the highest FAS scores (High), and half easiest samples and half hardest samples (Half).

Table 3: Evaluation results on medicine and general domains with LLaMA3-8B-Base model when adapting to medicine domain. The <u>underline</u> and **bold** fonts denote the same meaning as Table 1.

| **Model** | **Medicine Domain** | | | | **General Domain** | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | RCT | PubMedQA | MQP | Avg | MATH | MBPP | MMLU | CEVAL | Avg |
| LLaMA3-8B-Base | 73.6 | 59.8 | 66.2 | 60.1 | 18.0 | 57.9 | 65.9 | 49.3 | **47.8** |
| + CPT | 70.6 | 56.7 | 55.4 | 55.7 | 5.9 | 57.0 | 63.2 | 48.6 | 43.7 |
| + Mix-CPT | 71.0 | 63.4 | 79.2 | <u>63.4</u> | 16.6 | 58.0 | 66.2 | 49.7 | <u>47.6</u> |
| + Mix-CPT (Half) | 71.2 | 64.1 | 83.0 | **64.7** | 16.0 | 57.7 | 65.5 | 50.3 | 47.4 |

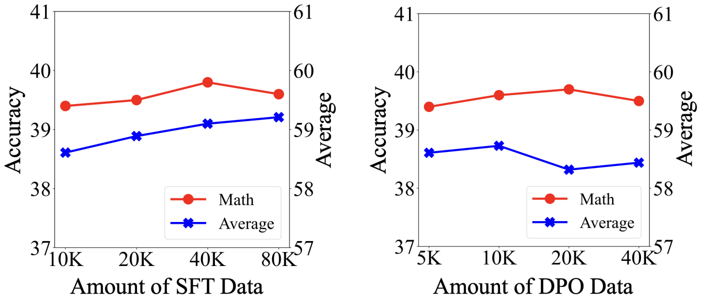

Figure 5: Accuracy on math and Average results *w.r.t.* Amount of SFT data **(Left)** and Amount of DPO data **(Right)**.

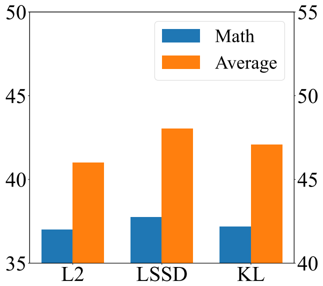

Figure 6: Accuracy on Math and Average results *w.r.t.* distillation and normalization.

- *Amout of Samples for SFT and DPO*: This involves four variants using different quantities of easiest samples from the TULU-V2-mix dataset (*i.e.,* 10K, 20K, 40K, and 80K) and from the UltraFeedback dataset (*i.e.,* 5K, 10K, 20K, and 40K).

We show the results in Figure 4 and Figure 5. Firstly, we can see that, by using the samples with the highest FAS scores can balance the domain and general capability best compared to other selection strategies. Then, it enhances both domain and general abilities simultaneously to a certain extent by increasing the amount of SFT training samples. Conversely, when increasing the amount of DPO training samples, the results remain fluctuating, which might be due to the alignment tax.

**Effect of Knowledge Distillation.** To examine the effectiveness of our proposed logit swap self-distillation (LSSD), we compare our method to two distillation and normalization methods:

- *w/ LSSD:* this is our Mix-CPT model using LSSD.
- *w/ KD:* this replaces LSSD with a typical knowledge distillation method without logits swapping.
- *w/ L2:* this replaces LSSD with L2 normalization.

The results are shown in Figure 6. We can see that compared to directly using *L2* regularization to constrain the changes in model parameter weights, *KD* method can better preserve general capabilities when learning domain capabilities. Through logits swapping, our proposed *LSSD* method refines the inconsistencies with the gold label in *KD* method, further enhancing the learning of domain-specific capabilities. Finally, *LSSD* achieves the best balance among the three methods.

**Performance on Medicine Domain.** Here, we further explore the performance of Mix-CPT and traditional CPT in the medical domain. Specifically, following the existing work (Cheng et al., 2024b), we select the PubMed dataset as the domain-specific corpus and choose RCT (Dernoncourt & Lee, 2017), PubMedQA (Jin et al., 2019), and MQP McCreery et al. (2020) as the domain-related evaluation collections. For the general domain evaluation benchmarks, we select MATH, MBPP, MMLU, and CEVAL from the aforementioned experimental settings. We show the results in Table 3. We observe that compared to traditional CPT methods, our proposed Mix-CPT approach more effectively facilitates the learning of domain-specific skills and better maintains the general capabilities, indicating the effectiveness and generalizability of our method.

**Computational Efficiency of Mix-CPT.** We have discussed the additional overhead of our method in Appendix A.5. Here, we further conduct additional experiments by only randomly employing 50% pre-training data for performing the logit swap self-distillation, *i.e.,* Mix-CPT (Half), which further reduce half of the costs introduced by self-distillation. We conduct the experiment on the medical domain with the LLaMA3-8B-Base model and evaluate its performance on domain-specific and general tasks as before. We show the results in Table 3. We find that compared to distilling the entire dataset, reducing the data by half further enhances the model's performance on domain-specific tasks, with only a slight decrease in general capabilities. This suggests that exploring even lower distillation ratios in the future could achieve a greater tradeoff between cost and performance.

## 4 Related Work

**Domain Adaptation of LLMs.** Our work is closely related to efforts in adapting general LLMs to specific domains (Yildiz et al., 2024; Ke et al., 2022; Scialom et al., 2022). Due to the increasing scale and complexity of LLMs, training domain-specific LLMs from scratch involves significantly high financial and ecological costs (Luccioni et al., 2023). To address this issue, recent work has been devoted to studying efficient approaches like continual pre-training, which involves incrementally training general LLMs based on new domain corpora (Que et al., 2024; Ke et al., 2022), and continual fine-tuning, aiming to fine-tune general LLMs on a series of downstream tasks related to target domains (Razdaibiedina et al., 2023; Scialom et al., 2022; Luo et al., 2023a). However, these approaches might result in catastrophic forgetting and performance degradation in general language tasks (Kar et al., 2022; Mehta et al., 2023). Another line of work has explored conducting instruction fine-tuning by synthesizing domain-related instructions (Cheng et al., 2024a; Jiang et al., 2024). Nevertheless, these studies require additional models to synthesize amounts of instructions highly related to specific domains, resulting in high computational costs. It is noted that our method differs from these works in several ways. Firstly, we disentangle domain adaptation into knowledge memorization and capability elicitation, focusing on learning domain-specific knowledge and solving domain tasks with learned knowledge, respectively. Secondly, we employ token swap self-distillation in the mixture pre-training to retain general knowledge and avoid catastrophic forgetting.

**Instruction Tuning and Alignment.** Instruction tuning (also known as supervised fine-tuning) employs human-annotated instructions (Sanh et al., 2022; Mishra et al., 2022; Köpf et al., 2023; Sun et al., 2023) or synthetic instructions by proprietary models (Taori et al., 2023; Chiang et al., 2023; Wang et al., 2023) to fine-tune LLMs. Besides, alignment with reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022a) or direct preference optimization (DPO) (Rafailov et al., 2023a) aims to align LLMs with human preference. Both instruction tuning and alignment are able to elicit knowledge from LLMs and improve their capabilities to solve downstream tasks. Recent work (Zhou et al., 2023) has demonstrated that LLMs mainly learn the style or format for interacting with users through simple instruction tuning and alignment, by leveraging their prior knowledge and capabilities already acquired during the pre-training stage. Furthermore, by comparing the token distribution before and after alignment, recent work (Lin et al., 2023) found that the most significant distribution shifts appear dominantly in stylistic tokens such as transitional phrases and discourse markers instead of contextual words that involve rich knowledge for solving downstream tasks. Inspired by these studies, we propose to expose knowledge memorization and capability elicitation from instruction tuning and alignment. Unlike these studies which typically focused on instruction tuning or alignment, we differ in that we unify the three stages of training LLMs (*i.e.,* continual pre-training, instruction tuning, and alignment) and conduct a knowledge mixture pre-training to mainly focus on learning new domain knowledge while maintaining general knowledge.

## 5 Conclusion

In this work, we proposed a two-stage domain adaptation approach, termed Mix-CPT, which encompasses both domain knowledge learning and general format alignment. Mix-CPT employed knowledge mixture continual pre-training to learn domain knowledge by integrating domain-specific raw data with general instructions and alignment data. Besides, we proposed Logit Swap Self-Distillation (LSSD) to relieve catastrophic forgetting. Based on the knowledge and capabilities acquired during pre-training, we then selected a small number of easy instructions to make the LLM to learn the style or format for interacting with human. Extensive experiments on three benchmark datasets showed that our proposed Mix-CPT outperforms the traditional method, obtaining improvements on both the domain and general capabilities.

## ACKNOWLEDGMENTS

This work was partially supported by National Natural Science Foundation of China under Grant No. 92470205, Beijing Municipal Science and Technology Project under Grant No. Z231100010323009, and Beijing Natural Science Foundation under Grant No. L233008. Xin Zhao is the corresponding author.

## REFERENCES

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. Program synthesis with large language models. *CoRR*, abs/2108.07732, 2021.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *CoRR*, abs/2310.10631, 2023.

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. Think you have solved direct-answer question answering? try arc-da, the direct-answer AI2 reasoning challenge. *CoRR*, abs/2102.03315, 2021.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7432–7439. AAAI Press, 2020.

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. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL `https://arxiv.org/abs/2107.03374`.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. Instruction pre-training: Language models are supervised multitask learners, 2024a.

Daixuan Cheng, Shaohan Huang, and Furu Wei. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b.

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

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. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.

OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. `https://github.com/open-compass/opencompass`, 2023.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. *CoRR*, abs/2310.01377, 2023.

Franck Dernoncourt and Ji Young Lee. Pubmed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In Greg Kondrak and Taro Watanabe (eds.), *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017, Volume 2: Short Papers*, pp. 308–313. Asian Federation of Natural Language Processing, 2017.

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

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

Xu Guo and Han Yu. On the domain adaptation and generalization of pretrained language models: A survey. *CoRR*, abs/2211.03154, 2022.

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

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. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021b.

Hanxu Hu, Pinzhen Chen, and Edoardo M. Ponti. Fine-tuning large language models with sequential instructions. *CoRR*, abs/2403.07794, 2024a.

Yiwen Hu, Huatong Song, Jia Deng, Jiapeng Wang, Jie Chen, Kun Zhou, Yutao Zhu, Jinhao Jiang, Zican Dong, Wayne Xin Zhao, and Ji-Rong Wen. Yulan-mini: An open data-efficient language model. *CoRR*, abs/2412.17743, 2024b.

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. 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*, 2023.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew E. Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. Camels in a changing climate: Enhancing LM adaptation with tulu 2. *CoRR*, abs/2311.10702, 2023.

Zhengbao Jiang, Zhiqing Sun, Weijia Shi, Pedro Rodriguez, Chunting Zhou, Graham Neubig, Xi Victoria Lin, Wen-tau Yih, and Srinivasan Iyer. Instruction-tuned language models are better knowledge learners. *arXiv preprint arXiv:2402.12847*, 2024.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 2567–2577. Association for Computational Linguistics, 2019.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1601–1611. Association for Computational Linguistics, 2017.

Sudipta Kar, Giuseppe Castellucci, Simone Filice, Shervin Malmasi, and Oleg Rokhlenko. Preventing catastrophic forgetting in continual learning of new natural language tasks. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 3137–3145. ACM, 2022.

Zixuan Ke, Yijia Shao, Haowei Lin, Hu Xu, Lei Shu, and Bing Liu. Adapting a language model while preserving its general knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 10177–10188. Association for Computational Linguistics, 2022.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. Openassistant conversations - democratizing large language model alignment. 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*, 2023.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466, 2019.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. RACE: large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 785–794. Association for Computational Linguistics, 2017.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy V, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Moustafa-Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. Starcoder: may the source be with you! *CoRR*, abs/2305.06161, 2023.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. *CoRR*, abs/2312.01552, 2023. doi: 10.48550/ARXIV.2312.01552. URL https://doi.org/10.48550/arXiv.2312.01552.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Rho-1: Not all tokens are what you need. *CoRR*, abs/2404.07965, 2024.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. Estimating the carbon footprint of bloom, a 176b parameter language model. *J. Mach. Learn. Res.*, 24:253:1–253:15, 2023.

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

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. *CoRR*, abs/2306.08568, 2023b.

Clara H. McCreery, Namit Katariya, Anitha Kannan, Manish Chablani, and Xavier Amatriain. Effective transfer learning for identifying similar questions: Matching user questions to COVID-19 faqs. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (eds.), *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 3458–3465. ACM, 2020.

Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. An empirical investigation of the role of pre-training in lifelong learning. *J. Mach. Learn. Res.*, 24:214:1–214:50, 2023.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *CoRR*, abs/2405.14734, 2024.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2381–2391. Association for Computational Linguistics, 2018.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 3470–3487. Association for Computational Linguistics, 2022.

OpenAI. Gpt-4 technical report. *OpenAI Blog*, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022a.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. 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*, 2022b.

Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, Feiyu Duan, Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, et al. D-cpt law: Domain-specific continual pre-training scaling law for large language models. *arXiv preprint arXiv:2406.01375*, 2024.

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

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

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. Progressive prompts: Continual learning for language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Mengjie Ren, Boxi Cao, Hongyu Lin, Cao Liu, Xianpei Han, Ke Zeng, Guanglu Wan, Xunliang Cai, and Le Sun. Learning or self-aligning? rethinking instruction fine-tuning. *CoRR*, abs/2402.18243, 2024.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton-Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code. *CoRR*, abs/2308.12950, 2023.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 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*, pp. 6107–6122. Association for Computational Linguistics, 2022.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David D. Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. 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*, 2023.

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

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4149–4158. Association for Computational Linguistics, 2019.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

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. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023.

Rui Wang, Fei Mi, Yi Chen, Boyang Xue, Hongru Wang, Qi Zhu, Kam-Fai Wong, and Ruifeng Xu. Role prompting guided domain adaptation with general capability preserve for large language models. *CoRR*, abs/2403.02756, 2024.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. How far can camels go? exploring the state of instruction tuning on open resources. *arXiv preprint arXiv:2306.04751*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. 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*, 2022.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report. *CoRR*, abs/2407.10671, 2024.

Çagatay Yildiz, Nishaanth Kanna Ravichandran, Prishruit Punia, Matthias Bethge, and Beyza Ermis. Investigating continual pretraining in large language models: Insights and implications. *CoRR*, abs/2402.17400, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4791–4800. Association for Computational Linguistics, 2019.

Yifan Zhang, Yifan Luo, Yang Yuan, and Andrew Chi-Chih Yao. Automathtext: Autonomous data selection with language models for mathematical texts. *CoRR*, abs/2402.07625, 2024.

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. A survey of large language models. *CoRR*, abs/2303.18223, 2023.

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

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: less is more for alignment. 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*, 2023.

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

# APPENDIX

## A EXPERIMENTAL SETUP

### A.1 DOMAIN-SPECIFIC CORPUS

In our experiments, we mainly focus on three popular domains for adapting general-purpose LLMs, *i.e.,* encyclopedia, mathematics, and code. For the encyclopedia domain, we select Wikipedia as the primary corpus, which is collaboratively developed by volunteers globally and can be freely accessed online. To enable LLMs to learn knowledge from new documents, we utilize the official 2024/03/01 Wikipedia dump[2] and conduct necessary data cleaning and filtering processes such as deduplication, resulting in approximately 4B tokens in raw Wikipedia documents. For the domain of mathematics, we opt for AutoMathText (Zhang et al., 2024), a carefully curated corpus derived from various sources including websites, arXiv, and GitHub. Each sample in this corpus has been labeled with a quality score from 0.0 ("the poorest") to 1.0 ("the best"), reflecting its relevance, quality, and educational value in the context of mathematical intelligence. Following previous work (Zhou et al., 2024), we specifically select those samples with scores higher than 0.7, containing about 0.7B tokens. For the field of code, we select the StarCoder (Li et al., 2023) corpus, which is widely recognized and employed in several studies (Luo et al., 2023b). It contains 86 programming languages, and we select the Python subset with approximately 1B tokens.

### A.2 GENERAL INSTRUCTION DATASETS

For general instruction datasets, we choose TULU-V2-mix (Ivison et al., 2023) and UltraFeedback (Cui et al., 2023) for instruction tuning and alignment, respectively. Specifically, each sample in TULU-V2-mix is either manually curated for quality or generated from GPT models to encourage complexity and diversity. We utilize the entire dataset of TULU-V2-mix (about 326K samples) mixed with domain-specific corpus for knowledge mixture continual pre-training (Section 2.1), and then select easy samples with the top-$K_1$ lowest perplexity score for subsequent instruction tuning. In addition, UltraFeedback is a widely-used diverse human preference alignment dataset, containing approximately 64K preference pairs. Similarly, we employ the whole dataset of UltraFeedback for knowledge mixture continual pre-training and then downsample easy pairs with the top-$K_2$ lowest perplexity score for alignment. It is noted that our TULU-V2-mix and UltraFeedback datasets are open-source and widely used in previous work (Meng et al., 2024; Hu et al., 2024a), ensuring a high level of transparency and facilitating fair experimental comparisons.

### A.3 BASELINES

In our experiments, we employ two prominent LLMs as the base model, *i.e., QWen2-7B* (Yang et al., 2024) and Meta-Llama-3-8B (Dubey et al., 2024). For comparative analysis, we consider the following three types of baseline methods:

- *Official Chat LLMs* consist of the official Chat LLMs that have undergone both instruction tuning and preference alignment using closed-source data. Here, we select two official chat LLMs corresponding to our selected base models, including *QWen2-7B-Chat* and *Llama-3-8B-Instruct*.

- *Reimplemented Chat LLMs* are developed by us following the processes of instruction tuning and alignment. Based on our selected three base LLMs, we conduct supervised fine-tuning (SFT) using TULU-V2-mix, followed by direct preference optimization (DPO) with UltraFeedback.

- *Continual Pre-training Augmented LLMs* include domain knowledge-enhanced Chat LLMs which initially undergo continual pre-training (CPT) with our domain-specific corpus, followed by the same implementation of supervised fine-tuning (SFT) and direct preference optimization (DPO) using TULU-V2-mix and UltraFeedback as open-source chat LLMs.

[2] https://dumps.wikimedia.org/enwiki/20240301/

Table 4: The categorization of the evaluation benchmarks

| | | |
|---|---|---|
| Specialized Domain | Wiki | NQ, TQ |
| | Math | GSM8K, MATH |
| | Code | MBPP, HumanEval |
| General Domain | Reading Comprehension | RACE-Hard, OpenBookQA |
| | Complex Reasoning | HellaSwag, CSQA, PIQA |
| | Examination | MMLU, BBH, ARC-Challenge, C-EVAL |
| | Instruction Following | MT-Bench |

### A.4 Evaluation Benchmarks and Metrics

For a comprehensive evaluation, we evaluate seven distinctive capabilities of LLMs based on a total of 16 representative NLP datasets:

• *Factual Question Answering* assesses the factual knowledge of LLMs in the Wikipedia domain. We employ NaturalQuestion (NQ) (Kwiatkowski et al., 2019) and TrivialQA (TQ) (Joshi et al., 2017) datasets and use the *Exact Match (EM)* metric to determine if the prediction is the same as the gold answer.

• *Math Reasoning* tests the LLMs' ability to solve mathematical problems. We use the GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b) datasets, and evaluate predictions using the *Accuracy* metric.

• *Code Reasoning* tests the LLMs' ability to solve programming problems. We use MBPP (Austin et al., 2021) and HumanEval (Chen et al., 2021) datasets, with the *Pass@K* metric assessing the likelihood that at least one of the top-$K$ generated code samples for a problem passes the unit tests.

• *Reading Comprehension* measures the LLMs' ability to comprehend a passage and answer related questions. We use the RACE-Hard (Lai et al., 2017) and OpenBookQA (Mihaylov et al., 2018) datasets, and employ the *Exact Match (EM)* metric.

• *Commonsense Reasoning* evaluates the ability to answer questions using commonsense knowledge. We use the HellaSwag (Zellers et al., 2019), CSQA (Talmor et al., 2019), and PIQA (Bisk et al., 2020) datasets, and employ the *Accuracy* metric.

• *Examination* includes comprehensive and challenging benchmarks designed to assess problem-solving ability across various domains. We use MMLU (Hendrycks et al., 2021a), BBH (Suzgun et al., 2023), and ARC-Challenge (Bhakthavatsalam et al., 2021) for English examinations and C-EVAL (Huang et al., 2023) for Chinese. Both benchmarks are evaluated using the *Accuracy* metric.

• *Instruction Following* assesses the LLMs' ability to engage in coherent, informative, and engaging conversations. We use the MT-Bench (Zheng et al., 2023) datasets. For evaluation, we utilize the GPT-4-turbo [3] as the judging model following the OpenCompass settings, assigning a score ranging from 1 to 10 to the answer. We multiply this score by ten, resulting in a final score of 100.

Specifically, we evaluate the above datasets based on the OpenCompass framework (Contributors, 2023), which is a one-stop platform for large model evaluation, aiming to provide a fair, open, and reproducible benchmark for large model evaluation. We summarize the datasets in Table 4

### A.5 Implementation Details

For both models, we conduct the mixed continual pre-training with the same hyper-parameters. Specifically, we set the batch size as 1920, the maximal sequence length as 2048, the maximal and minimal learning rate as 2e-5 and 5e-6, and the co-efficient $\alpha$ as 0.8. Then, for LLaMA3-8B model, we select the top 10,000 samples and 10,000 samples for SFT and DPO, respectively. For the SFT

[3] https://platform.openai.com/docs/models

stage, we perform 2 epochs of training with a learning rate of 5e-6 and a batch size of 128. For the DPO stage, we perform 2 epochs of training with a learning rate of 5e-7, a batch size of 128, and a $\beta$ of 0.01. For the QWen2-7B model, we select the top 10,000 samples and 10,000 samples for SFT and DPO, respectively. For the SFT stage, we perform 2 epochs of training with a learning rate of 5e-7 and a batch size of 128. For the DPO stage, we perform 2 epochs of training with a learning rate of 1e-7, a batch size of 128, and a $\beta$ of 0.1. When evaluating, we set the temperature as 0.3 and top-p as 0.9 for MT-Bench according to the official guideline and set the temperature as 0 for other benchmarks to control the randomness. As for the computational cost, the additional overhead introduced by the instruction and alignment data may be negligible, which only introduces approximately 0.09B tokens in continual pre-training and 0.008B tokens for totally selected 20K pairs in the format alignment stage.