Efficient Domain Continual pretraining by Mitigating the Stability Gap

Yiduo Guo[1], **Jie Fu**[2], **Huishuai Zhang**[1], **Dongyan Zhao**[1]
[1]Peking University, [2]HKUST
yiduo@stu.pku.edu.cn, jiefu@ust.hk, zhanghuishuai@pku.edu.cn,
zhaodongyan@pku.edu.cn

Abstract

Continual pretraining enables Large Language Models (LLMs) to adapt to specialized domains like medicine and law. However, we observe a consistent phenomenon across different model sizes and domains: a temporary performance drop at the start of the continual pretraining process, followed by a performance recovery phase. To gain a deeper understanding of this issue, we use the stability gap— a concept adapted from the visual domain—which explains this initial drop arises from instability in the model's general abilities. We validate this hypothesis through a series of experiments. To address this initial instability and enhance LLM performance within a fixed compute budget, we propose a training strategy that mitigates instability by increasing the number of epochs, alongside two data sampling strategies targeting data domain relevance and corpus distribution. We conduct experiments on Llama-family models to validate the effectiveness of our strategies for continual pretraining and instruction tuning in medical and legal domains. Our strategies improve the average medical task performance of the OpenLlama-3B model from 36.2% to 40.7% using only 40% of the original training budget, while also enhancing general task performance without causing forgetting. Furthermore, we apply our strategies to continually pre-train and instruction-tune the Llama-3-8B model. The resulting model, **Llama-3-Physician**[1], achieves the best medical performance among open-source models and rivals GPT-4 on specific tasks.

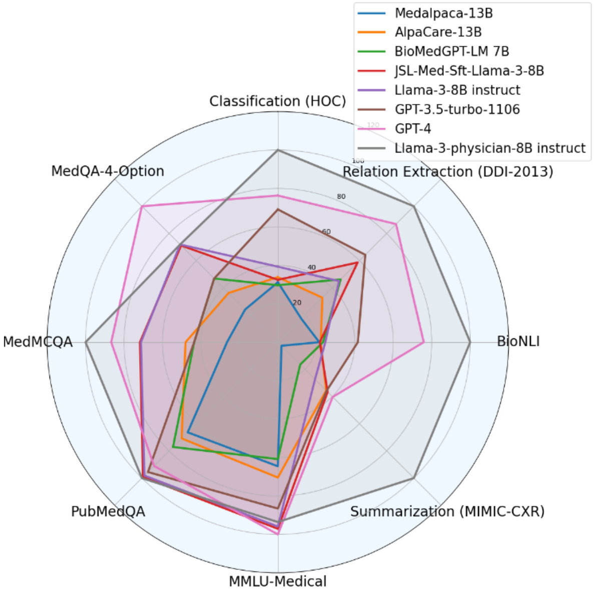


Figure 1: The performance comparison between our model (Llama-3-physician) and other models involves reporting the ratio of each model's task performance to the best performance of that task among all models.

1 Introduction

Continual pretraining is an important approach for LLMs to improve their performance in target domains (Huang et al., 2023; Yang et al., 2024a; Chen et al., 2023c), learn new topics and languages (Jiang et al., 2024; Gupta et al., 2023), and even boost their general capabilities (Ibrahim et al., 2024). While extensive research has focused on understanding LLM mechanisms during pretraining from scratch (Biderman et al., 2023a; Xue et al., 2024), far less attention has been given to how LLMs behave during continual pretraining (Que et al., 2024). This gap in the literature is particularly striking given the importance of continual pretraining in adapting models to new domains and evolving knowledge. In this paper, we report a surprising phenomenon observed during continual pretraining: rather than an immediate improvement, **LLM performance on target domain tasks initially declines in the early stages of training. Only after further training, when more data is incorporated, does performance recover and eventually surpass that of the original model.** We consistently observe this performance pattern—a V-shaped curve—across various model

[1] We release our models at https://huggingface.co/YiDuo1999/Llama-3-Physician-8B-Instruct.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 32850–32870
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

scales and target domains, including medical and legal fields, demonstrating its generalizability.

To explore the underlying mechanisms of this phenomenon, we draw inspiration from the concept of the stability gap (De Lange et al., 2022; Caccia et al., 2021), originally introduced in the context of vision models in continual learning. The stability gap describes how a model's performance on previously learned tasks initially degrades when learning new tasks, before gradually recovering as it adapts. Previous research (De Lange et al., 2022) attributes this initial drop to an imbalance between the model's stability gradient—its ability to maintain performance on prior tasks—and its plasticity gradient—the capacity to adapt to new ones. Early in training, the model's plasticity gradient dominates, leading to a temporary performance decline. As training progresses, the stability gradient strengthens, allowing performance to recover.

Applying this framework to LLMs, we hypothesize that *the initial performance drop in continual pretraining stems from a similarly insufficient stability gradient to preserve the model's general capabilities (e.g., instruction-following skills). Over time, as the plasticity gradient diminishes and the stability gradient rises, task performance rebounds.* Supporting this hypothesis, we observe a similar V-shaped pattern in general-domain tasks, where initial declines give way to recovery. Further analysis of weight updates throughout the training process provides additional evidence for this interpretation.

But how can we mitigate it to optimize continual pretraining? Given a fixed computing budget, we know that the stability gap causes inefficiency in continual pretraining as it delays performance improvement. To address this, we propose **three efficient continual pretraining strategies**:

1. Instead of continually pretraining the LLM on a large corpus for one epoch, which induces a large plasticity gradient for a long period, **we continually pre-train the LLM on a subset of the corpus for multiple epochs.**

2. **Select the domain subset validated by domain Perplexity (PPL) to learn rich domain knowledge**, leading to faster performance recovery and higher peak performance.

3. **Use a data mixture that is similar to the pretraining data distribution in data source and rate**, thus reducing the distribution shift and mitigating the knowledge forgetting of general instruction-following ability.

To verify our strategy, we first conduct experiments on the OpenLlama-3B model with **medical domain continual pretraining**. We find that **our strategies show its computational efficiency** by reducing the original compute budget to **40%** while also enhancing the LLM's peak performance (See Table 1). We further verify the generalization of our strategies in **legal and general continual pretraining settings (see Appendixes F and E)** We also compare our strategies with other continual pretraining techniques and analyze the influence of important learning factors, such as learning rate, for our strategies. Finally, **we apply our strategies to both the continual pretraining and instruction tuning processes of the Llama-3-8B model (Meta, 2024)**, efficiently enhancing its performance on diverse medical tasks, outperforming other open-source LLM baselines, and **achieving performance comparable to GPT-4** (See performance comparison in Figure 1).

2 Related work

Large language Models such as GPT-4 (OpenAI, 2023), Gemini (Team), and Llama (Touvron et al., 2023a)), have billions of parameters and show strong performance on various basic natural language tasks (Qin et al., 2023), human examination (Hendrycks et al., 2020b; Zhong et al., 2023), and agent-related tasks (Guo et al., 2023; Liu et al., 2023; Zhou et al., 2023). Their success attracts researchers to analyze LLMs' learning properties during the pretraining process (Kaplan et al., 2020; Biderman et al., 2023a; Zhang et al., 2024a). Kaplan et al. (2020) finds the pretraining scaling rule for model size and dataset size and then Hoffmann et al. (2022) proposes the Chinchilla rule that claims the equal importance of the model size and the number of training tokens. Sorscher et al. (2022) further claims that pruning low-quality data can improve the above neural scaling laws. However, high-quality training tokens are limited and may be run out soon (Villalobos et al., 2022). Thus, some researchers try to maximize the utilization of the existing corpus by training it for multiple epochs (Muennighoff et al., 2024; Xue et al., 2024). But they observe the performance degradation (Hernandez et al., 2022; Xue et al., 2023; Hoffmann et al., 2022) after training 4 epochs.

Continual pretraining gradually becomes necessary for LLMs to expand their basic ability (Wu et al., 2022; Fu et al., 2024; Zhuang et al., 2024), avoid outdated information (Jiang et al., 2024), and become the domain expert (Huang et al., 2023; Yang et al., 2024a; Chen et al., 2023c; Nguyen et al., 2023; Wu et al., 2023; Yıldız et al., 2024; Xie et al., 2024a). The domain corpus for continual pretraining can be collected by n-gram models (Muennighoff et al., 2024), heuristic rules designed by human experts (Chen et al., 2023c; Zhang et al., 2024c) or automatically identified by a LLM (Zhang et al., 2024c). For the continual pretraining techniques. Ke et al. (2023, 2022) focused on adding masks or adjusting the architecture of small language models like RoBERTa to protect the learned general knowledge. However, these techniques result in huge computational consumption for LLMs. Recent studies (Gupta et al., 2023) show that learning rate re-warming can improve LLMs' downstream task performance and a stability gap appears when replaying the previous data. (Ibrahim et al., 2024) further claims that learning rate re-warming, re-decaying, and replay can make the continual pretraining performance match the performance of fully re-training when continually pretraining the English LLM on the German corpus. Other continual pretraining method studies focus on selecting useful tokens (Lin et al., 2024), expanding MOE architecture (Chen et al., 2023a), and knowledge distillation (Jin et al., 2021b).

Continual learning and the Stability Gap Continual learning aims to design methods that can learn new knowledge without the catastrophic forgetting of previously learned knowledge (Kirkpatrick et al., 2017; Van de Ven et al., 2022). To mitigate the forgetting problem when learning a new task, replaying previous tasks' data (Rolnick et al., 2019; Buzzega et al., 2020; Prabhu et al., 2020; Buzzega et al., 2021; Guo et al., 2022) becomes the main approach. De Lange et al. (2022); Caccia et al. (2021) further find that, although they conduct the replay approach, the vision model still first loses its performance stability in previous classification tasks (the performance drops abruptly) and then gradually recovers. They call it the stability gap phenomenon. Different from them, we focus on the continual pretraining of the LLM and observe that both the LLM's domain performance and general ability suffer from the stability gap.

3 Identifying the stability gap in continual pretraining

In this section, we describe the unique performance phenomenon observed during continual pretraining, where performance on the target domain initially drops before rising. We then introduce the concept of the stability gap to explain this behavior and validate our explanation through experiments.

3.1 Investigating the behavior of LLMs during continual pretraining

Experiment setup In this study, we choose **OpenLlama3B-v2 (Geng and Liu, 2023)** as our default LLM, which has been pretrained on openly Refined-Web dataset (Penedo et al., 2023). We consider the medical domain as our primary target domain. Following previous work (Chen et al., 2023b), we set the compute budget to 50 billion (50B) training tokens. More details are provided in Appendix A. To measure the model's medical domain performance, we follow (Chen et al., 2023c) and consider the average accuracy performance over the MMLU-Medical-Genetics (Hendrycks et al., 2020a), MedQA (Jin et al., 2021a), PubMedQA (Jin et al., 2019), and MedMCQA (Pal et al., 2022) tasks (see task details in Appendix C).

Data collection To deploy a simple, economic, and scalable pretraining data collection method, we collect the continual pretraining corpus by first training a small n-gram model (e.g., KenLM) on a few human-collected data, and then using it to calculate the data perplexity on the Refined-Web dataset (Penedo et al., 2023), and then extracting the 50B lowest-PPL tokens. We justify its effectiveness in Appendix A and show its scalable in other domains like legal in Appendix F.

Observation (1): The medical task performance first drops and rises during continual pretraining. We report the average performance and its deviation on medical tasks every 5 billion training tokens. From Figure 2(a), we observe that the domain task performance initially drops during the first 5 billion tokens and then gradually recovers and improves. Furthermore, as shown in Figure 2(b) (a fine-grained view), we observe that the performance declines at the beginning, followed by a gradual recovery. We also perform a t-test to statistically verify the performance drop shown in Figure 3(a). Additionally, we consider the TinyLlama model (Zhang et al., 2024b), a 1.1B

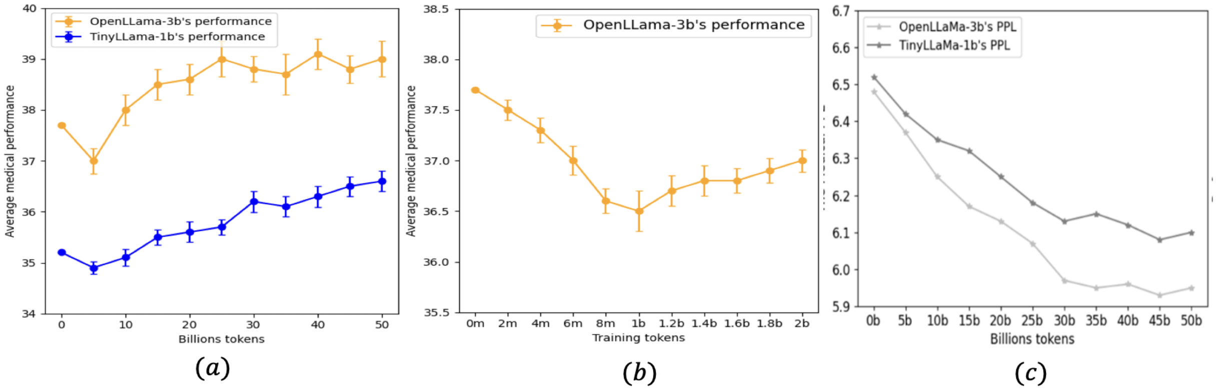


Figure 2: (a) reports the models' average medical performance with the standard deviation among 5 runs during the medical continual pretraining process. (b) reports the models' average medical performance at the beginning. (c) illustrates the models' average medical perplexity (PPL) during the medical continual pretraining process.

Llama model trained on 3 trillion tokens, and continually pre-train it on the medical corpus. From Figure 2(a), we observe that its performance on medical tasks also shows the same trend, despite being trained on so many tokens.

Observation (2): The perplexity of medical Wikipedia steadily declines during continual pretraining. We further measure the average perplexity (PPL) of the models on the Wikipedia corpus about medical terms[2]. From Figure 2(b), we observe that the PPL steadily drops. This suggests that the LLM begins acquiring medical knowledge during the initial phase of continual pretraining and continues to enhance its understanding throughout the entire process.

More Observations: We also examine continual pretraining in both the legal domain and a general setting. Similar V-shaped performance curves are observed, reinforcing that the initial performance drop followed by a subsequent rise in target task performance is a common phenomenon in the continual pretraining of LLMs. Detailed results are provided in Appendix B.

3.2 Stability Gap: A conceptual explanation for the initial performance drop and then following recovery.

The Stability Gap refers to the initial decline in a vision model's performance on previous tasks while learning a new task, followed by a subsequent improvement, even when data from the earlier tasks is replayed. Lange et al. (2022) explains this by disentangling the model gradient $\mathcal{G}$ into α-weighted plasticity and stability components: $\mathcal{G} = \alpha\mathcal{G}_{plasticity} + (1-\alpha)\mathcal{G}_{stability}$, where $\mathcal{G}_{plasticity}$ focuses on learning the new task by minimizing its data loss, while $\mathcal{G}_{stability}$ seeks to maintain performance on previous tasks by keeping the loss of replay data low. They attribute the initial performance drop to the plasticity gradient exceeding the stability gradient to reduce new task loss, resulting in a failure to maintain performance on previous tasks. As performance declines, the stability gradient strengthens, leading to a balance between gradients and eventual performance recovery.

Explanation of our observations Directly applying the concept of the stability gap to explain our phenomenon is challenging because we do not explicitly replay the pretraining corpus, which is a key element in traditional stability gap analysis. However, during domain-specific continual pretraining, the language modeling loss serves two critical functions: it both learns domain-specific knowledge and implicitly preserves general knowledge and text modeling capabilities (See LLM's non-zero commonsense performance in Figure 3(b)), *as the domain corpus still contains general information. This implicit preservation serves as a form of 'self-replay,' enabling the retention of general knowledge through what we term the stability gradient.* Further, we infer that performance declines because the plasticity gradient for learning domain-specific knowledge surpasses the stability gradient for retaining general text knowledge and text modeling ability. Over time, the stability gradient strengthens to restore general knowledge and modeling abilities, while the plasticity gradient has learned knowledge in the target domain, leading to performance improvement.

[2] https://huggingface.co/datasets/gamino/wiki_medical_terms

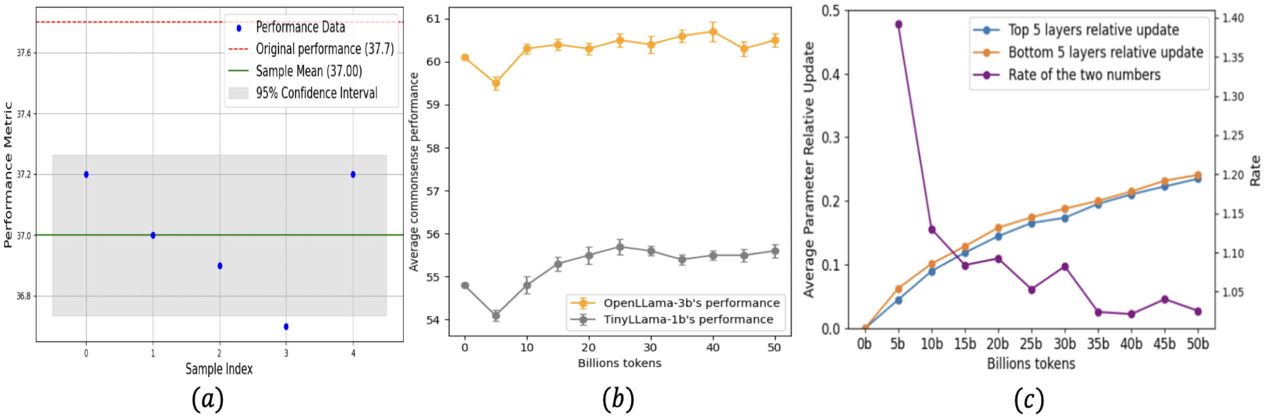


Figure 3: In Figure (a), we perform a statistical t-test to demonstrate that the performance of OpenLLaMa significantly drops at the 5B training tokens. (b) shows the OpenLLaMa's average common-sense task performance among 5 runs during medical continual pretraining. (c) illustrates the OpenLlama model's relative parameter update during the medical continual pretraining process. We report the average weight relative update of weights in the top 5 layers and the bottom 5 layers. We also report the rate between the two average numbers.

Empirical verification for our explanation Based on our inference, we can predict that the commonsense task performance follows a similar V-shape curve as the stability gradient gradually rises. We verify our prediction in Figure 3(a). We also find evidence for our explanation at the weight level by (2) measuring the relative weight update of each weight w as $\frac{w_t - w_0}{w_0}$, where w_t is the weight value during continual pretraining and w_0 is the original weight value. A high relative weight update indicates a large gradient for updating the weight. Figure 3(b) shows that the bottom layers' weights initially have a higher relative weight update than the top layers ($rate > 1.35$). Previous studies indicate that bottom layers learn the syntax and low-level semantics (Devlin et al., 2019; Hewitt and Manning, 2019; Ling et al., 2023), while top layers contain high-level semantics and task-specific knowledge (Yang et al., 2024b; Chen et al., 2024). This suggests that the top layers' weights indeed lack sufficient stability gradient to maintain instruction-following ability initially. The performance then recovers as the relative weight updates (stability gradient) increase in the top layers and domain knowledge is learned, as indicated by the continuous drop in medical perplexity.

4 Our method

In this section, we propose three efficient continual pretraining strategies for reducing the above stability gap problem. The training process and details follow those in the above section. We then compare the effectiveness of our strategies with other continual pretraining techniques. Next, we investigate the impact of important learning factors, such as the learning rate, on our strategies. Finally, we make ablation study about our strategy.

4.1 Efficient continual pretraining strategies for mitigating the stability gap

Strategy I: Continually pre-train the LLM on a corpus subset across multiple epochs rather than the entire large corpus for a single epoch. The key insight is that a larger corpus demands a high plasticity gradient for a longer period. In contrast, pretraining the LLM on a subset of the corpus across multiple epochs reduces the need for sustained high plasticity after the first epoch and accelerates the rise of the stability gradient.

Strategy II: Continually pre-train the LLM on the domain corpus subset validated by the domain PPL. The second strategy utilizes domain-related tokens from the whole RefinedWeb dataset, identified based on their alignment with a well-defined and small domain corpus, to accelerate performance recovery during continual pretraining. Specifically, we ranked all samples in the RefinedWeb dataset based on their perplexity (PPL) scores, calculated using a KenLM model trained on the medical Wiki corpus described in Appendix. A. We refer to this as the domain PPL. And then we select the subset with the lowest PPL as the domain corpus. We use this data collected method as a low domain PPL score indicates that the passage has a stronger alignment with the distribution of the medical Wikipedia corpus. we further validate our

Method	Training tokens number	MMLU-Med-Avg	PubMedQA	MedMCQA	MedQA-4-Option	Avg
OpenLLaMa-3B	-	25.6	68.4	25.4	25.4	36.2
Full token baseline	50B	26.1	70.4	26.1	27.1	37.4
Re-warming and re-decaying	50B	26.5	70.3	27.1	27.1	37.7
Replay 5B data	50B	26.3	69.2	27.6	26.9	37.5
Replay 10B data	50B	29.3	71.0	30.4	27.6	39.5
Replay 15B data	50B	29.0	70.1	29.4	26.2	38.7
Freezing top 5 layers	50B	26.2	69.9	27.1	27.3	37.6
Freezing bottom 5 layers	50B	26.0	69.1	25.4	25.7	36.5
Our strategies	20B	**30.0**	**71.2**	**34.0**	**27.8**	**40.7**

Table 1: Zero-shot accuracy across various medical benchmarks.

method in Section 4.5.

Strategy III: Use a data mixture rate similar to the pretraining data. The pretraining data mixture rate is a vital factor for the pretraining performance of large language models (LLMs) (Xie et al., 2024c; Shen et al., 2023). Therefore, we propose a third strategy that follows the pretraining data's mixture rate [3] to construct the continuous pre-training, aiming to reduce the distribution gap and stabilize the instruction-following ability of the LLM during continual pretraining. Specifically, for the OpenLlama model, we first randomly collect 5 billion tokens following the Llama mixture rate (Touvron et al., 2023a). To incorporate a medical corpus, we replace the sampled CC and C4 data (which constitute 82% of the 5 billion tokens) with the KenLM-selected tokens using strategy II. We conduct the sampling and replacement operation at each training epoch.

4.2 Setup

Baselines We consider the following baselines for comparison: (1) *Continually pretraining the OpenLLaMa-3B LLM with 50 billion collected medical tokens for one epoch ("the full token baseline").* (2) *Re-warming and re-decaying the learning rate* of (1) based on the paper by (Ibrahim et al., 2024). (3) *Replay baselines*: Following (Chen et al., 2023b), we randomly sample 5B (10%), 10B (20%), and 15B (30%) tokens from OpenLLaMa-3B's pretraining dataset (the RefinedWeb dataset) and combine them with 50B medical tokens. Pretraining is stopped once a total of 50B tokens have been processed. This baseline does not consider the data mixture rate. (4) *Parameter protection baselines*: Following (Harun and Kanan, 2023), we freeze the top 5 layers' weights during the continual pretraining process of (1) to protect the high-level instruction-following ability and mitigate the stability gap. We also consider another baseline that freezes the bottom 5 layers' weights for comparison.

Evaluation benchmark We follow (Chen et al., 2023b) and consider the tasks of *PubMedQA, MedMCQA, and MedQA-4-Option*. For the *MMLU benchmark* (Hendrycks et al., 2020a), we consider the average performance of its medical topics, including *medical genetics, anatomy, clinical knowledge, professional medicine, and college medicine*. We use the lm-evaluation-harness framework (Gao et al., 2023) to measure the baselines' zero-shot performance. The training details are in Appendix A.

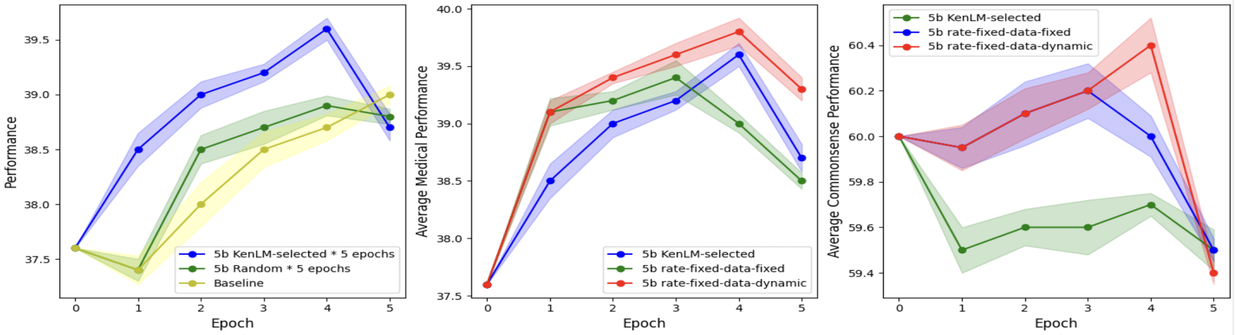
4.3 Results

From Table 1, we find that (1) **our strategies improve the base model's average medical task performance significantly (4.5%) with only 20 billion training tokens.** This demonstrates the effectiveness and efficiency of our strategies for continual pretraining. (2) Other techniques can also improve continual pretraining performance, except for the baseline 'Freezing bottom 5 layers,' which hinders the learning of medical domain knowledge. We further verify our strategies' effectiveness in continual law pretraining. We put the results in Appendix F.

4.4 Factor Analysis

Impact of learning rate and training subset size To analyze the impact of training factors such as *learning rate and training subset size* and find the optimal hyperparameter configuration for our experiments, we conducted a series of experiments, with details provided in Appendix G. Our findings show that a learning rate that is too high leads to significant drops in generalization ability, while a rate too low hampers the acquisition of new domain knowledge. Additionally, using a subset that is too large (e.g., 10 billion tokens) introduces a stability

[3] When we do not know the data rate, we can use recent advanced methods to approximately infer it (Hayase et al., 2024).



(a) (b) (c)

Figure 4: (a) reports the average medical performance among 5 runs with its deviation during the medical continual pretraining process. The baseline is pretraining the OpenLlama-3B model with 50b medical tokens with one epoch. '5b Random' is pretraining the LLM with 5b tokens randomly selected from the 50b medical tokens for 5 epochs. '5b KenLM selected' is pretraining the LLM with the KenLM selected 5b tokens of the 50b medical tokens for 5 epochs. (b) shows the average medical performance across 5 epochs. (c) illustrates the average commonsense task performance across 5 epochs.

gap and slows performance. Conversely, a smaller subset yields better initial performance but leads to rapid overfitting in later epochs.

4.5 Ablation study of our three strategies

In Figure 4(a), we observe that **the first strategy** leads to faster performance recovery. The LLM achieves peak performance at the fourth epoch, consistent with previous studies (Xue et al., 2024).

For **the second strategy**, we continually pre-trained the OpenLlama-3B model on the subset of samples with the lowest domain perplexity over 5 epochs. From Figure 4 (a), we observe that the KenLM-selected subset with the lowest PPL indeed enables the LLM to recover performance faster and stronger in the medical domain. To verify the positive correlation between domain perplexity (PPL) and whether a passage belongs to the domain, we sort all RefinedWeb passages by the domain PPL, from lowest to highest. We then sample one passage every 50 entries and ask the LLaMa-3-72B model to determine if the passage is medical. The probability of a 'Yes' response steadily decreases from 0.95 to 0.004 as perplexity increases from 3.2 to 11.5. Further analysis of the pretraining subset size is presented in Sec. 4.4.

For **the third strategy**, based on whether this sampling and replacement operation is performed once or at each epoch, we propose two methods: (1) Rate-Fixed-Data-Fixed, where the medical tokens are sampled once to create a fixed training corpus used for all epochs, and (2) Rate-Fixed-Data-Dynamic, where the operation is repeated independently at each epoch, producing a dynamically changing training corpus.

From Figure 4(b), we observe that the second method achieves higher peak performance by striking a better balance between replaying pretraining data and learning domain-specific knowledge. Furthermore, our strategies improve the average performance on general commonsense tasks while mitigating overfitting when training on a small medical corpus, as demonstrated in Figure 4(c). They also reduce medical perplexity and the rate of relative weight updates, as discussed in Appendix D. Additionally, we evaluate the effectiveness of our three strategies in a general continual pretraining setting, detailed in Appendix E.

5 Llama-3-Physician: Deploying our strategies into the Llama-3 Model

Continual pretraining We continually pre-train the Llama3-8B-base model using our three strategies with medical continual pretraining tokens constructed in Sec. 4 for 4 epochs. The training details are in Appendix H. After the continual pretraining process, we find that the average medical performance drops slightly, likely due to the unknown data mixture rate of Llama-3. However, *the medical perplexity is significantly lower than that of the Llama3-8B-base model.*

Task-specific fine-tuning To evaluate LLMs' performance in the supervised learning setting, we follow (Chen et al., 2023b) and individually conduct task-specific finetuning on both the base models and the continually pre-trained models using each benchmark's training set. We also consider 8 task-finetuned baselines. We put task details in Appendix C and training and baseline details in

Model	MMLU-Medical	PubMedQA	MedMCQA	MedQA-4-Option	Avg
Llama-2-7B (Touvron et al., 2023b)	56.3	61.8	54.4	49.6	53.2
BioMistral SLERP 7B (Labrak et al., 2024)	60.5	75.2	44.2	47.3	56.8
MEDITRON-7B (Chen et al., 2023b)	55.6	74.4	59.2	52.0	57.5
Llama3-Aloe-8B-Alpha (Gururajan et al., 2024)	72.7	77.2	59.0	62.3	67.8
Llama-2-70B	74.7	78.0	62.7	61.3	67.2
MEDITRON-70B	73.6	**80.0**	65.1	**65.4**	69.0
GPT-3.5-turbo-finetuned (Shi et al., 2024)	70.5	71.4	61.8	63.3	66.7
Llama-3-8B base	47.2	52.1	38.2	35.5	43.3
Llama-3-8B Fine-tuned (ours)	82.3	75.8	60.0	61.1	69.8
Llama-3-8B Full (ours)	82.0	78.6	61.8	60.8	70.8
Llama-3-Physician-8B (ours)	**85.0**	79.1	**81.4**	61.5	**76.7**

Table 2: Accuracy comparison across various medical benchmarks in the task-specific fine-tuning setting. Llama-3-8B Fine-tuned is directly fine-tuned on these tasks. For 'Llama-3-8B Full', we first continually pre-trained the Llama with 50B medical tokens and then finetuned the pretrained model on these tasks. For Llama-3-Physician-8B, we first continually pre-trained the Llama with with our strategies and then finetuned the pretrained model.

Appendix H.

Results We use the lm-eval-harness (Gao et al., 2023) to evaluate our model (Llama-3-Physician) and related baselines' performance. No demonstration examples are used. **From Table 2, we find that our model outperforms other baselines with similar model scales on the four evaluation benchmarks by a clear margin.** This is due to the following reasons: (1) we use the newest and strongest open-source Llama-3 model rather than older Llama-2 or Mistral-7B, (2) we continually pre-train the base model with KenLM-selected medical tokens (compared to 'Llama-3-8B fine-tuned and Llama-3-8B instruct'), and (3) our strategies further boost the gains from continual pretraining markedly (compared to 'Llama-3-8B Full'). **Our 7B model also outperforms many larger LLMs (70B) on average.**

5.1 Deploying our strategies into the instruction tuning process

For the instruction-tuning setting, we follow (Xie et al., 2024b) and tunes the continual pretrained Llama-3-8B model with a combination of medical tasks. (See details in Appendixes H and I.

Observations We observe a similar performance phenomenon in the instruction tuning process in Figure 9. **Our strategies can mitigate the initial performance drop and achieve higher peak performance during the instruction tuning process**, thereby extending the application of our strategies.

Baselines For instruction-tuning, we consider instruction-tuned models like Mistral-7B-instruct (Jiang et al., 2023), Zephyr-7B-β-instruct (Tunstall et al., 2023), PMC-Llama-7B (Wu et al., 2023), BioMedGPT-LM 7B (Zhang et al., 2023a), Medalpaca-13B (Han et al., 2023), AlpaCare-13B (Zhang et al., 2023b), Me-LLaMA-13B chat(Xie et al., 2024b), Llama-3-8B instruct (Meta, 2024), and JSL-Med-Sft-Llama-3-8B (johnsnowlabs, 2024). These LLMs are tuned with general instructions or medical task instructions.

Results From Table 6, we find that our model outperforms other open-source baselines in question-answering tasks by a clear margin. Additionally, **our model's average performance is close to that of GPT-4.** Furthermore, in Table 5, we observe that our model significantly outperforms GPT-4 in medical classification, relation extraction, natural language inference, and summarization tasks. This demonstrates the significant advantage of our model in processing diverse medical tasks.

6 Conclusion

Our paper explores the behavior of LLMs when continually pretraining them on a new domain's corpus and observes the stability gap, a phenomenon marked by a significant initial performance drop followed by a slow recovery. We explain it from the view of plasticity and stability gradients and then propose three strategies that effectively improve the LLM's domain performance and reduce computational costs by reducing the stability gap. Furthermore, we deploy our strategies on the newest Llama-3-8B model, which achieves the strongest performance among open-source baselines of similar model scales and outperforms the closed-source GPT-3.5 model.

Limitations and Potential impacts Ideally, knowing the pretraining data mixture could maximize the outcome of our method, but most strong open-source LLMs didn't provide their training data mixture. Our Llama-3-8B experiment shows we can still improve significantly in this scenario. Due to limitations in computing resources, we plan to verify our conclusions and strategies on larger LLMs in the future. Our strategies are designed to address the machine learning problem of the stability gap, and we do not see any potential risks. The datasets and base models used in this paper will be open-sourced.

References

Simon Baker, Ilona Silins, Yufan Guo, Imran Ali, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2016. Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics*, 32(3):432–440.

Mohaddeseh Bastan, Mihai Surdeanu, and Niranjan Balasubramanian. 2022. Bionli: Generating a biomedical nli dataset using lexico-semantic constraints for adversarial examples. *arXiv preprint arXiv:2210.14814*.

Stella Biderman, Hailey Schoelkopf, Quentin G. Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023a. Pythia: A suite for analyzing large language models across training and scaling. *ArXiv*, abs/2304.01373.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023b. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. 2020. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara. 2021. Rethinking experience replay: a bag of tricks for continual learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2180–2187. IEEE.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. 2021. New insights on reducing abrupt representation change in online continual learning. *arXiv preprint arXiv:2104.05025*.

Wuyang Chen, Yanqi Zhou, Nan Du, Yanping Huang, James Laudon, Zhifeng Chen, and Claire Cui. 2023a. Lifelong language pretraining with distribution-specialized experts. In *International Conference on Machine Learning*, pages 5383–5395. PMLR.

Xiaodong Chen, Yuxuan Hu, and Jing Zhang. 2024. Compressing large language models by streamlining the unimportant layer. *ArXiv*, abs/2403.19135.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. 2023b. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*.

Zeming Chen, Alejandro Hernández-Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023c. Meditron-70b: Scaling medical pretraining for large language models.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*.

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. *arXiv preprint arXiv:1803.05457*.

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

Matthias De Lange, Gido van de Ven, and Tinne Tuytelaars. 2022. Continual evaluation for lifelong learning: Identifying the stability gap. *arXiv preprint arXiv:2205.13452*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Jon Durbin. 2024. airoboros: Customizable implementation of the self-instruct paper.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. 2024. Data engineering for scaling language models to 128k context. *arXiv preprint arXiv:2402.10171*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *ArXiv*, abs/2101.00027.

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

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

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models.

Yiduo Guo, Bing Liu, and Dongyan Zhao. 2022. Online continual learning through mutual information maximization. In *International Conference on Machine Learning*, pages 8109–8126. PMLR.

Yiduo Guo, Zekai Zhang, Yaobo Liang, Dongyan Zhao, and Duan Nan. 2023. Pptc benchmark: Evaluating large language models for powerpoint task completion. *arXiv preprint arXiv:2311.01767*.

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

Ashwin Kumar Gururajan, Enrique Lopez-Cuena, Jordi Bayarri-Planas, Adrian Tormos, Daniel Hinjos, Pablo Bernabeu-Perez, Anna Arias-Duart, Pablo Agustin Martin-Torres, Lucia Urcelay-Ganzabal, Marta Gonzalez-Mallo, et al. 2024. Aloe: A family of fine-tuned open healthcare llms. *arXiv preprint arXiv:2405.01886*.

Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bressem. 2023. Medalpaca–an open-source collection of medical conversational ai models and training data. *arXiv preprint arXiv:2304.08247*.

Md Yousuf Harun and Christopher Kanan. 2023. Overcoming the stability gap in continual learning. *arXiv preprint arXiv:2306.01904*.

Jonathan Hayase, Alisa Liu, Yejin Choi, Sewoong Oh, and Noah A Smith. 2024. Data mixture inference: What do bpe tokenizers reveal about their training data? *arXiv preprint arXiv:2407.16607*.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020b. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

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

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *North American Chapter of the Association for Computational Linguistics*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. 2022. Training compute-optimal large language models. *ArXiv*, abs/2203.15556.

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

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. *arXiv preprint arXiv:2403.08763*.

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

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

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021a. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2021b. Lifelong pretraining: Continually adapting language models to emerging corpora. *arXiv preprint arXiv:2110.08534*.

johnsnowlabs. 2024. Jsl-med-sft-llama-3, a finetuned medical llm developed by john snow labs.

Alistair EW Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. 2019. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 6(1):317.

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

Zixuan Ke, Haowei Lin, Yijia Shao, Hu Xu, Lei Shu, and Bing Liu. 2022. Continual training of language models for few-shot learning. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Zixuan Ke, Yijia Shao, Haowei Lin, Hu Xu, Lei Shu, and Bing Liu. 2023. Adapting a language model while preserving its general knowledge. *arXiv preprint arXiv:2301.08986*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv preprint arXiv:2402.10373*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Matthias De Lange, Gido M. van de Ven, and Tinne Tuytelaars. 2022. Continual evaluation for lifelong learning: Identifying the stability gap. *ArXiv*, abs/2205.13452.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, et al. 2024. Rho-1: Not all tokens are what you need. *arXiv preprint arXiv:2404.07965*.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun-Qing Li, Hejie Cui, Xuchao Zhang, Tian yu Zhao, Amit Panalkar, Wei Cheng, Haoyu Wang, Yanchi Liu, Zhengzhang Chen, Haifeng Chen, Chris White, Quanquan Gu, Jian Pei, Carl Yang, and Liang Zhao. 2023. Domain specialization as the key to make large language models disruptive: A comprehensive survey.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv: 2308.03688*.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. 2024. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36.

Tuan Dung Nguyen, Yuan-Sen Ting, Ioana Ciucă, Charlie O'Neill, Ze-Chang Sun, Maja Jablonska, Sandor J. Kruk, Ernest Perkowski, Jack W. Miller, Jason Li, Josh Peek, Kartheik G. Iyer, Tomasz R'o.za'nski, Pranav Khetarpal, Sharaf Zaman, David Brodrick, Sergio J. Rodr'iguez M'endez, Thang Bui, Alyssa Goodman, Alberto Accomazzi, Jill P. Naiman, Jesse Cranney, Kevin Schawinski, and UniverseTBD. 2023. Astrollama: Towards specialized foundation models in astronomy. *ArXiv*, abs/2309.06126.

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

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

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

Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. 2020. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 524–540. Springer.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

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

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. *Advances in neural information processing systems*, 32.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Isabel Segura-Bedmar, Paloma Martínez Fernández, and María Herrero Zazo. 2013. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). Association for Computational Linguistics.

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

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Hang Wu, Carl Yang, and May Dongmei Wang. 2024. Medadapter: Efficient test-time adaptation of large language models towards medical reasoning.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. *ArXiv*, abs/2206.14486.

Gemini Team. Gemini: A family of highly capable multimodal models. Technical report, Technical report, Google, 12 2023. URL https://storage. googleapis. com ....

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

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. 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, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

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

Gido M Van de Ven, Tinne Tuytelaars, and Andreas S Tolias. 2022. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197.

Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and An Chang Ho. 2022. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *ArXiv*, abs/2211.04325.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *ArXiv*, abs/1707.06209.

Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Towards building open-source language models for medicine.

Zhaofeng Wu, Robert L Logan IV, Pete Walsh, Akshita Bhagia, Dirk Groeneveld, Sameer Singh, and Iz Beltagy. 2022. Continued pretraining for better zero-and few-shot promptability. *arXiv preprint arXiv:2210.10258*.

Qianqian Xie, Qingyu Chen, Aokun Chen, C.A.I. Peng, Yan Hu, Fongci Lin, Xueqing Peng, Jimin Huang, Jeffrey Zhang, Vipina Kuttichi Keloth, Xingyu Zhou, Huan He, Lucila Ohno-Machido, Yonghui Wu, Hua Xu, and Jiang Bian. 2024a. Me llama: Foundation large language models for medical applications. *ArXiv*, abs/2402.12749.

Qianqian Xie, Qingyu Chen, Aokun Chen, Cheng Peng, Yan Hu, Fongci Lin, Xueqing Peng, Jimin Huang, Jeffrey Zhang, Vipina Keloth, et al. 2024b. Me llama: Foundation large language models for medical applications. *arXiv preprint arXiv:2402.12749*.

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

Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. 2023. To repeat or not to repeat: Insights from scaling llm under token-crisis. *ArXiv*, abs/2305.13230.

Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. 2024. To repeat or not to repeat: Insights from scaling llm under token-crisis. *Advances in Neural Information Processing Systems*, 36.

Xianjun Yang, Junfeng Gao, Wenxin Xue, and Erik Alexandersson. 2024a. Pllama: An open-source large language model for plant science. *arXiv preprint arXiv:2401.01600*.

Yifei Yang, Zouying Cao, and Hai Zhao. 2024b. Laco: Large language model pruning via layer collapse. *ArXiv*, abs/2402.11187.

Çağatay Yıldız, Nishaanth Kanna Ravichandran, Prishruit Punia, Matthias Bethge, and Beyza Ermis. 2024. Investigating continual pretraining in large language models: Insights and implications. *arXiv preprint arXiv:2402.17400*.

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

Kai Zhang, Jun Yu, Zhiling Yan, Yixin Liu, Eashan Adhikarla, Sunyang Fu, Xun Chen, Chen Chen, Yuyin Zhou, Xiang Li, et al. 2023a. Biomedgpt: A unified and generalist biomedical generative pre-trained transformer for vision, language, and multimodal tasks. *arXiv preprint arXiv:2305.17100*.

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

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024b. Tinyllama: An open-source small language model.

Xinlu Zhang, Chenxin Tian, Xianjun Yang, Lichang Chen, Zekun Li, and Linda Ruth Petzold. 2023b. Alpacare:instruction-tuned large language models for medical application.

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

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. *arXiv preprint arXiv:2304.06364*.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

Alex Zhuang, Ge Zhang, Tianyu Zheng, Xinrun Du, Junjie Wang, Weiming Ren, Stephen W Huang, Jie Fu, Xiang Yue, and Wenhu Chen. 2024. Structlm: Towards building generalist models for structured knowledge grounding. *arXiv preprint arXiv:2402.16671*.

A The Details of pretraining

Models Download and Training Hyperparameters The models `OpenLLaMa-3B`, `TinyLLaMa-1B`, and `Pythia-410m` are downloaded from their official websites. For the baselines, we follow the setups described in their respective official papers. The pretraining code is based on the `transformers` library. Below are the key details for training:

- **Context size:** Predict the next token with a context size of 2048.
- **Hardware:** Training is executed using 192 V100 GPUs.
- **Optimizer:** We use the `AdamW` optimizer with the following parameters:
  - $\beta_1 = 0.9$, $\beta_2 = 0.95$
  - Weight decay = 0.01
  - Learning rate = 3e-4

- **Learning Rate Scheduler:** We employ a cosine learning rate scheduler with a 0.1 warmup ratio to gradually adapt to training complexity.

- **Precision:** We use bf16 precision for computational efficiency.

- **Gradient Accumulation:** Set to 4 steps, with each training batch containing about 340 million tokens.

- **Epoch:** For OpenLLaMa-3B and TinyLLaMa-1B, we continually pre-train them with the constructed 50 billion medical tokens constructed in Section 4 for one epoch. For Pythia-410m, we continually pretrain it with the construct 100 billion new constructed corpus for one epoch.

- **Efficient Inference:** We support FlashAttention-2 (Dao, 2023) for more efficient inference and long-context decoding.

When deploying these strategies in the continual pretraining process, we use the same learning rate schedule as used in the pretraining phase.

How to Construct Our Pretraining Corpus?

- **Medical Continual Pretraining Corpus:** We begin by training a small model (e.g., KenLM, $n = 3$) on the wiki_medical_terms corpus. This model is then used to evaluate the perplexity of text samples in the Refined-Web dataset. We select the 50B tokens with the lowest perplexity, resulting in a medical corpus.

- Note that the wiki-medical-terms dataset, containing 6,000 terms and descriptions, serves as a reference for perplexity calculations, but it is not the source of the 50B tokens.

- **Legal Continual Pretraining Corpus:** The construction process is similar to the medical corpus, but instead of using the wiki_medical_terms dataset, we use the Caselaw Access Project dataset, which is downloaded from Hugging Face.

- **General Continual Pretraining Corpus:** For this, we randomly sample 100 billion tokens from the 2021-2022 subset of the RefinedWeb dataset. We consider this subset as reliable because the Pythia-410m LLM is pretrained on the Pile dataset, which contains only data before 2021. Pretraining the Pythia-410m model on this new corpus can be viewed as pretraining on new, unseen data.

The reason for choosing KenLM: Both KenLM and FastText (n-gram models) are easy to train and provide fast inference on large corpora. In contrast, other sorting partition methods, such as calculating the entropy of an LLM, require significant GPU resources and are more challenging to deploy for both customers and academic researchers.

B More observation and analysis

For continual law pretraining, we use the same procedure to collect domain corpus and the same optimization setup to train the LLM. For its evaluation, we consider three QA tasks: MMLU-International-Law, MMLU-Professional-Law, and Contract-QA from LegalBench (Guha et al., 2023). We report the average performance in Figure 5(a), which shows a similar v-shape performance curve. Continual pretraining on another large corpus is an important approach to boost the pretrained LLM's general task performance (Jiang et al., 2024; Gupta et al., 2023). We call it the general continual pretraining setting. We further find that it also exists a similar performance phenomenon. Specifically, we continually pre-train the Pythia-410m model (Biderman et al., 2023b) (initially pre-trained on the Pile (Gao et al., 2020) dataset) on the RefinedWeb dataset (Penedo et al., 2023) to boost its general ability. We measure its general ability using the average performance across 10 common-sense tasks and report the average performance of every 10 billion tokens. Training details are in Appendix A and task details are in Appendix C. From Figure 5(b), we observe that the LLM's general task performance first drops significantly and then gradually rises.

Based on our observations, the initial drop followed by a rise in target task performance is a general phenomenon in the continual pretraining of LLMs of various sizes.

C Task and Baseline Information

For the medical evaluation, we follow (Chen et al., 2023b) and mainly consider the following four tasks:

MedMCQA (Pal et al., 2022) is a large-scale and comprehensive dataset for multichoice (four-

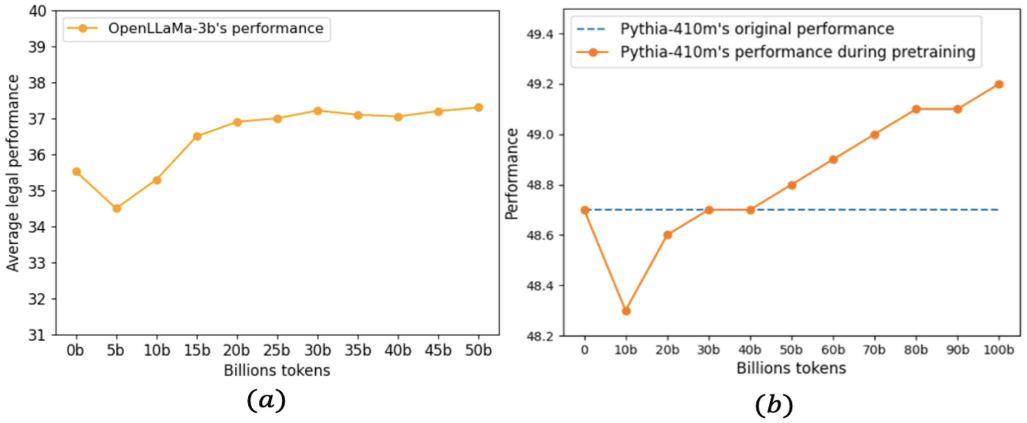


Figure 5: (a) shows the OpenLLaMa's average legal task performance during law continual pretraining. (b) shows the Pythia model's average common-sense task performance when we continually pre-train it on the new RefinedWeb datasets.

option) medical question answering. It is derived from real-world medical entrance exam questions (Indian AIIMS and NEET-PG) and consists of over 194,000 high-quality medical questions. These questions cover 2,400 healthcare topics and 21 medical subjects, exhibiting a wide range of topical diversity. The average token length is 12.77.

MedQA (Jin et al., 2021a)is a multichoice question-answering dataset collected from the professional medical board exam, the United States Medical License Exams (USMLE). It comprises 12,723 questions sourced from a comprehensive collection of 18 English medical textbooks that have been extensively utilized by medical students and USMLE candidates. Questions in MedQA cover a wide range of topics in clinical medicine, necessitating responses with professional expertise and complex multi-hop reasoning across multiple pieces of evidence. The average question and option length is 116.6 and 3.5, respectively.

MMLU (Hendrycks et al., 2020b) is a comprehensive multi-task language understanding test dataset that encompasses 57 tasks across various domains such as mathematics, history, computer science, law, etc. In our experiments, we specifically focus on a subset of medical reasoning-related tasks including clinical knowledge, college medicine, medical genetics, and professional medicine.

PubMedQA (Jin et al., 2019) is a biomedical question and answering dataset derived from PubMed abstracts. It contains 1k expert annotated multi-choice question-and-answer samples based on 211.3k PubMed articles. The task of PubMedQA is to provide answers to research questions with yes/no/maybe responses based on the corresponding abstracts. The average question and context length is 14.4 and 238.9, respectively.

HOC (Baker et al., 2016) is a classification task to decide the Hallmarks of Cancer (HOC) taxonomy of the article based on its abstract. The input is an abstract text. There are 10 topics you will need to decide whether the article is related to. Topics: sustaining proliferative signaling, evading growth suppressors, resisting cell death, enabling replicative immortality, inducing angiogenesis, activating invasion and metastasis, genomic instability and mutation, tumor promoting inflammation, and cellular energetics, and avoiding immune destruction.

DDI 2023 (Segura-Bedmar et al., 2013) is a task to predict the relationship between the given head entity labeled as @DRUG1$andtailentitylabeledas@DRUG$2 within a given sentence, this relation which must be in ('mechanism', 'effect', 'advice', 'int', 'none'). mechanism: this type is used to annotate drug-drug interactions that are described by their pharmacokinetic mechanism. effect: this type is used to annotate drug-drug interactions describing an effect or a pharmacodynamic mechanism. advice: this type is used when a recommendation or advice regarding a drug interaction is given. int: this type is used when a drug-drug interaction appears in the text without providing any additional information. none: there are no drug-drug interactions.

BioNLI (Bastan et al., 2022) is a task to classify the relationship between the given medical premise and hypothesis into one of the following labels: entailment, contradiction, or neutral. This dataset contains abstracts from biomedical literature and mechanistic premises generated with nine different strategies.

MIMIC-CXR (Johnson et al., 2019) is a generation task that derives the impression from findings in the radiology report.

The dataset statistics are in Table 3

For the evaluation of general task ability, we consider the following 10 commonsense tasks:

ARC-Challenge and ARC-Easy ARC (Clark et al., 2018) is a multiple-choice question-answering dataset, containing questions from science exams from grade 3 to grade 9. The dataset is split into two partitions: Easy and Challenge, where the latter partition contains the more difficult questions that require reasoning. Most of the questions have 4 answer choices.

BoolQ (Clark et al., 2019) is a question-answering dataset for yes/no questions containing 15942 examples. These questions are naturally occurring —they are generated in unprompted and unconstrained settings. Each example is a triplet of (question, passage, answer), with the title of the page as optional additional context. The text-pair classification setup is similar to existing natural language inference tasks.

COPA (Roemmele et al., 2011) consists of 1000 questions, split equally into development and test sets of 500 questions each. Each question is composed of a premise and two alternatives, where the task is to select the alternative that more plausibly has a causal relation with the premise.

HellaSWAG (Zellers et al., 2019) is a dataset for studying grounded commonsense inference. It consists of 70k multiple choice questions about grounded situations: each question comes from one of two domains – activitynet or wikihow – with four answer choices about what might happen next in the scene. The correct answer is the (real) sentence for the next event; the three incorrect answers are adversarially generated and human-verified, so as to fool machines but not humans.

OpenBookQA (Mihaylov et al., 2018) is a new kind of question-answering dataset modeled after open-book exams for assessing human understanding of a subject. It consists of 5,957 multiple-choice elementary-level science questions (4,957 train, 500 dev, 500 test), which probe the understanding of a small "book" of 1,326 core science facts and the application of these facts to novel situations.

PIQA (Bisk et al., 2020) dataset introduces the task of physical commonsense reasoning and a corresponding benchmark dataset Physical Interaction: Question Answering or PIQA. Physical commonsense knowledge is a major challenge on the road to true AI-completeness, including robots that interact with the world and understand natural language. PIQA focuses on everyday situations with a preference for atypical solutions.

Race (Lai et al., 2017) is a large-scale reading comprehension dataset with more than 28,000 passages and nearly 100,000 questions. The dataset is collected from English examinations in China, which are designed for middle school and high school students. The dataset can serve as the training and test sets for machine comprehension.

SciQ (Welbl et al., 2017) dataset contains 13,679 crowdsourced science exam questions about Physics, Chemistry and Biology, among others. The questions are in multiple-choice format with 4 answer options each. For the majority of the questions, an additional paragraph with supporting evidence for the correct answer is provided.

WinoGrande (Sakaguchi et al., 2021) is a new collection of 44k problems, inspired by the Winograd Schema Challenge (Levesque, Davis, and Morgenstern 2011), but adjusted to improve the scale and robustness against the dataset-specific bias. Formulated as a fill-in-a-blank task with binary options, the goal is to choose the right option for a given sentence which requires commonsense reasoning.

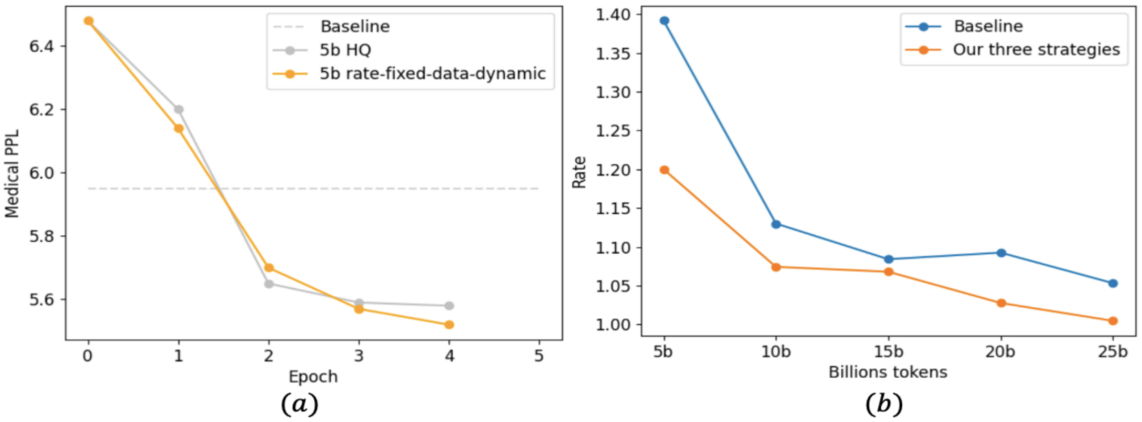
We use the lm-eval-harness (Gao et al., 2023) to evaluate the LLM on these tasks' test set and report the zero-shot performance.

D The Perplexity and relative parameter update rate of the LLM using our strategies

From Figure 6(a), we observe that the LLM using our strategies gradually decreases its average medical perplexity, indicating that the LLM is acquiring rich medical knowledge. Its average medical perplexity at the fourth epoch is even lower than that of the OpenLLaMa-3B model, which has been continually pre-trained with 50 billion medical tokens. From Figure 6(b), we also find that the ratio between the average relative parameter updates of the bottom 5 layers and the top 5 layers of the OpenLLaMa-3B model using our strategies is closer to 1. This suggests that the plasticity gradient and the stability gradient are more balanced when employing our strategies.

Table 3: Dataset statistics

Dataset	# Train	# Test	Source
MedMCQA (Pal et al., 2022)	182,822	4183	Exam
MedQA (Jin et al., 2021a)	10178	1273	Exam
MMLU (Hendrycks et al., 2020b)	-	163	Exam
PubMedQA (Jin et al., 2019)	211,269	500	Literature
HOC (Baker et al., 2016)	1108	315	Literature
DDI 2023 (Segura-Bedmar et al., 2013)	1108	315	Literature
BioNLI (Bastan et al., 2022)	5544	6308	Literature
MIMIC-CXR (Bastan et al., 2022)	122,014	1606	Literature



(a)

(b)

Figure 6: (a) reports the average medical perplexity of the OpenLLaMa-3B using our strategies. '5b KenLM selected' means the LLM using our strategies I and II. '5b rate-fixed-data-dynamic' means the LLM using our three strategies. 'Baseline' is the average medical perplexity of the OpenLLaMa-3B model that has been continually pre-trained with 50 billion medical tokens. (b) shows the rate between the bottom 5 layers' average relative parameter and the top 5 layers' average relative parameter update of the OpenLLaMa-3B using our strategies. 'Baseline' is the rate of the OpenLLaMa-3B model during the continual pretraining with 50 billion medical tokens.

E Deploying our strategies into the general continual pretraining setting

Continually pretraining one LLM on another large corpus is an approach to boost its general ability (Gupta et al., 2023). We consider the scenario of continually pretraining the Pythia-410m model on the RefinedWeb dataset. The Pythia-410m model has been pre-trained on the Pile dataset. In this context, we use the average performance of 10 commonsense and reading comprehension tasks, as detailed in Appendix C, to measure the LLM's general task performance. To test the effectiveness of strategy I in the general continual pretraining setting, we conduct multi-epoch experiments with different training subset sizes. The tokens in each training subset are randomly sampled from the RefinedWeb dataset and the computational consumption of each experiment can not be beyond the compute budget (100 billion tokens). From Figure 7, we find that strategy I indeed helps the Pythia-410m model to mitigate the stability gap and achieve better peak performance. We also find the best performance among our experiments is achieved when pretraining the LLM with 11 billion tokens for 7 epochs. However, we can not find a good data filter for the second strategy. We have tried to train a KenLM on WikiText as the data filter for measuring the sample's ability in improving LLMs' general ability. But it does not work. From Figure 7, we find that strategies I and III can help the LLM to reduce the stability gap and achieve higher performance.

F Effectiveness of our strategies in the legal domain

We consider strong baselines and report their legal performance in Table 4.

G Impact of learning rate and training subset size

Impact of the learning rate To analyze the influence of training factors like learning rate and

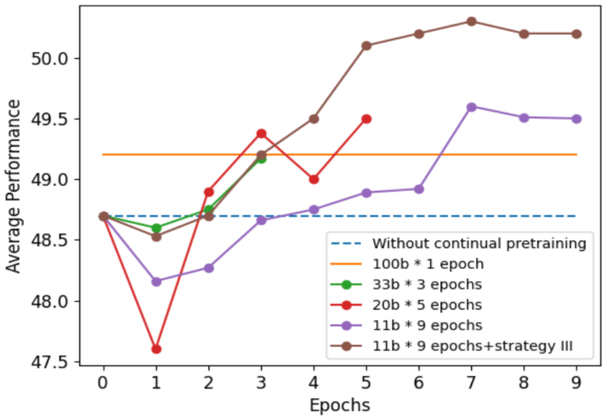
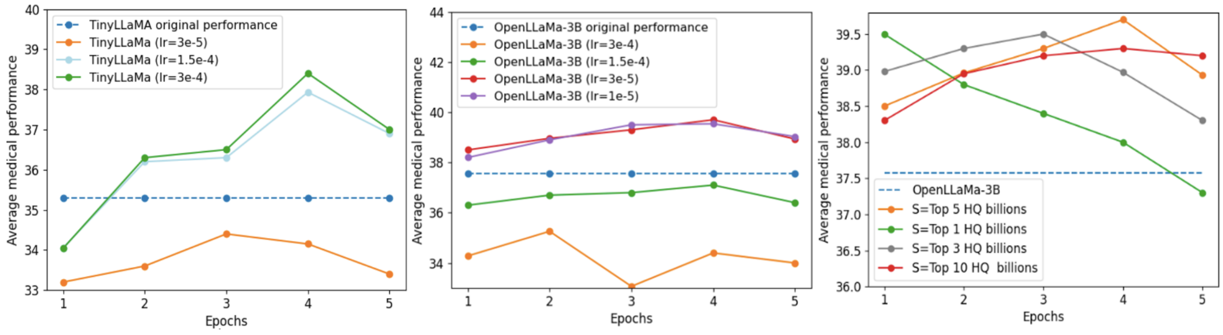


Figure 7: We report the average performance of the 10 commonsense and reading compression task here. The Model is Pythia-410m.



(a) (b) (c)

Figure 8: (a) reports the performance of TinyLlama-1.1B across multiple epochs. All these experiments use our strategies with different pretraining learning rates. (b) reports the performance of OpenLlama-3B across multiple epochs. All of the experiments in (a) and (b) use our strategies with different pretraining learning rates. (c) reports the performance of OpenLlama-3B across multiple epochs with different training subset sizes S. To collect the pretraining corpus with different sizes, we first rank all samples of the 50 billion medical tokens based on the perplexity calculated by the trained KenLM (see Sec. 3.1). Then, we select the first S billion tokens with the lowest perplexity. For all experiments here, we report the average task performance of PubMedQA, MedMCQA, MMLU-medical-genetics, and MedQA-4-Option tasks.

Method	Training tokens number	MMLU-International-Law	MMLU-Professional-Law	Contract-QA	Avg
OpenLLaMa-3B	-	27.1	28.4	51.0	35.5
Full token baseline	50B	28.1	29.4	54.4	37.4
Re-warming and re-decaying	50B	28.5	27.3	55.1	37.0
Replay 10B data	50B	29.3	29.0	54.4	37.6
Our strategies	20B	**31.0**	**31.2**	**57.0**	**39.7**

Table 4: Zero-shot accuracy across various legal benchmarks.

training subset size, we conduct a series of experiments. We put the details in Appendix xxx. We find that too high learning rate leads to severe general-ability drops and too low leads to poor learning of new domain knowledge. Too large a subset (e.g., 10 billion tokens) results in a stability gap and slower performance, too small a subset yields better initial performance, but it also causes quick overfitting in later epochs. We further verify the best hyperparameter setup for our experiments. The pretraining learning rate is a crucial factor for updating LLMs during continual pretraining. To investigate its impact on our strategies, we conduct continual pretraining experiments with different learning rates. From Figure 8(a) and (b), we find that the optimal learning rate varies with the LLM scale: a small LLM (e.g., TinyLlama-1.1B) requires a higher learning rate (e.g., 3e-4), whereas larger LLMs (e.g., OpenLlama-3B) benefit from a lower learning rate (e.g., 3e-5). If the learning rate is too low (e.g., 3e-5 for TinyLlama-1.1B), the LLM cannot learn domain knowledge effectively to boost performance. Conversely, if the learning rate is too high (e.g., 3e-4 for OpenLlama-3B), performance declines as the large learning rate leads to a significant plasticity gradient, causing the LLM to lose its general instruction-following ability for completing tasks. Based on our analysis experiments, we set the pretraining learning rate at 3e-4 for TinyLlama and 3e-5 for OpenLlama-3B's experiments.

Impact of the training subset size The size of the training subset is another important factor in our strategies. To determine the optimal training subset size, we conduct pretraining experiments on Llama-3b using various training subset sizes. From Figure 8(c), we observe that a smaller domain-related subset yields better initial performance and mitigates the stability gap (e.g., 1 billion tokens), but it also causes the performance to drop quickly in later epochs due to overfitting. A larger subset (e.g., 10 billion tokens) results in a stability gap and slower performance recovery, as the LLM needs to maintain a high plasticity gradient to learn a large number of new samples. Based on our experiments, we select a subset with 5 billion KenLM-selected tokens, as it mitigates the stability gap, achieves the best peak performance, and is computationally effective.

H The Training Details of Deploying our Strategies into the Llama-3 Model

pretraining details: The pretraining task is to predict the next token with a context size of 8192. The training is executed using 16 H100 80GB GPUs. We employ the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.95$, a weight decay of 0.01, and a learning rate of 3e-5. We use a cosine learning rate scheduler with a 0.1 warmup ratio for gradual adaptation to training complexity and bf16 precision for computational efficiency. Gradient accumulation is set to 12 steps, and each training batch contains about 340 million tokens. We also add support for FlashAttention-2 (Dao, 2023) for more efficient inference and long-context decoding.

Task-specific finetuning details: We employ the AdamW optimizer with a weight decay of 0.01 and a learning rate of 3e-5. We use a cosine learning rate schedule with a 10% warmup ratio, decaying the final learning rate to 10% of the peak learning rate. We fine-tune the LLMs for 3 epochs. Since MMLU (Hendrycks et al., 2020a) does not have a training set, we follow (Chen et al., 2023b) and primarily consider the MMLU-Medical-Genetics benchmark, evaluating the model finetuned on MedMCQA.

For baselines in task-specific fine-tuning, we consider three kinds of baselines here: (1) Task-specific finetuning of the base model of open-source LLMs. This includes models such as Llama-2-70B, Llama-3-8B, and Llama3-Aloe-8B-Alpha (Gururajan et al., 2024). We copy their results from their respective papers (Gururajan et al., 2024) or the Meditron paper (Chen et al., 2023b) except for the Llama-3-8B, which we finetuned using the same process as our strategies. (2) Task-specific finetuning of continually pre-trained LLMs like meditron (Chen et al., 2023b), BioMistral SLERP 7B (Labrak et al., 2024), Llama-3-8B-full. These LLMs have been continually pre-trained with a medical corpus. We copy their results from their papers, except for Llama-3-8B-full, for which we continually pre-train the Llama-3-8B with 50B medical tokens collected in Section 3.1, and then finetune it using the same process as our strategies. (3) Closed-source LLMs. This includes models like ChatGPT and GPT-4 (OpenAI, 2023). The results are measured using the Microsoft Azure OpenAI API service (Shi et al., 2024).

Instructions-tuning details:

Deployment In the instruction tuning process, our first strategy is common as the medical instruction tuning process usually involves multi-epochs training (Zhang et al., 2023a; Xie et al., 2024b; Han et al., 2023). For the second strategy, we consider Deita (Liu et al., 2024), a simple automatic instruction data selector, to select high-quality medical instruction data. This selector uses the LLM to give quality scores for instructions and considers the diversity of instruction data by sampling data from different clustering. For the last strategy, we consider high-quality general instruction datasets like Airoboros-3.2 (Durbin, 2024) to mitigate the forgetting in general instruction following ability.

We consider the combination of the question-answering training set of MedMCQA (Pal et al., 2022), MedQA (Jin et al., 2021a), PubMedQA (Jin et al., 2019), classification task HOC (Baker et al., 2016), relation extract task DDI2013 (Segura-Bedmar et al., 2013), inference task BioNLI (Bastan et al., 2022), and summarization task MIMIC-CXR (Johnson et al., 2019) tasks . To avoid potential data contamination, for each test sample of MedQA (Jin et al., 2021a), PubMedQA (Jin et al., 2019), and MedMCQA (Pal et al., 2022) tasks, we delete the training samples that contain its option. The specific dataset details are in Appendix C. For the training samples of theMedQA (Jin et al., 2021a),PubMedQA (Jin et al., 2019), and MedMCQA (Pal et al., 2022) tasks, we use the instruction template from the Meditron paper (Chen et al., 2023b). For the other datasets' training samples, we use their original instructions.

We employ the AdamW optimizer with a weight decay of 0.01 and a learning rate of 3e-5. We use a cosine learning rate schedule with a 10% warmup ratio, decaying the final learning rate to 10% of the peak learning rate. We fine-tune the LLMs for 3 epochs. The global batch size is 96 and max sequence length is 1024. Unlike the above task-specific fine-tuning, we only tune one LLM here and use the instruction-tuned LLM to test all benchmarks.

For the baselines' results, we download the baselines' official models/deploy their APIs and then test their task performance using lm-eval-harnesses and Me-Llama's evaluation frameworks. If the paper does not release its model, we copy the results from the original paper (e.g., Me-Llama).

I Details Analysis of the Instruction Tuning Process

Task type	Classification	Relation extraction	Natural Language Inference	Summarization
Datasets	**HOC**	**DDI-2013**	**BioNLI**	**MIMIC-CXR**
Mistral-7B-instruct (Jiang et al., 2023)	35.8	14.1	16.7	12.5
Zephyr-7B-instruct-β (Tunstall et al., 2023)	26.1	19.4	19.9	10.5
PMC-Llama-7B (Wu et al., 2023)	18.4	14.7	15.9	13.9
Medalpaca-13B (Han et al., 2023)	24.6	5.8	16.4	1.0
AlpaCare-13B (Zhang et al., 2023b)	26.7	11.0	17.0	13.4
BioMedGPT-LM 7B (Zhang et al., 2023a)	23.4	15.5	17.9	6.2
Me-Llama-13B (Xie et al., 2024b)	33.5	21.4	19.5	**40.0**
JSL-Med-Sft-Llama-3-8B (johnsnowlabs, 2024)	25.6	19.7	16.6	13.8
Llama-3-8B instruct	31.0	15.1	18.8	10.3
GPT-3.5-turbo-1106	54.5	21.6	31.7	13.5
GPT-4 (OpenAI, 2023)	60.2	29.2	57.8	15.2
Llama-3-physician-8B instruct (ours)	**78.9**	**33.6**	**76.2**	37.7

Table 5: Performance comparison for general medical tasks in the instruction-tuning setting. For BioNLI, DDI 2023, and HOC tasks, we report macro-F1. For MIMIC-CXR summarization tasks, we report Rouge-L as the result.

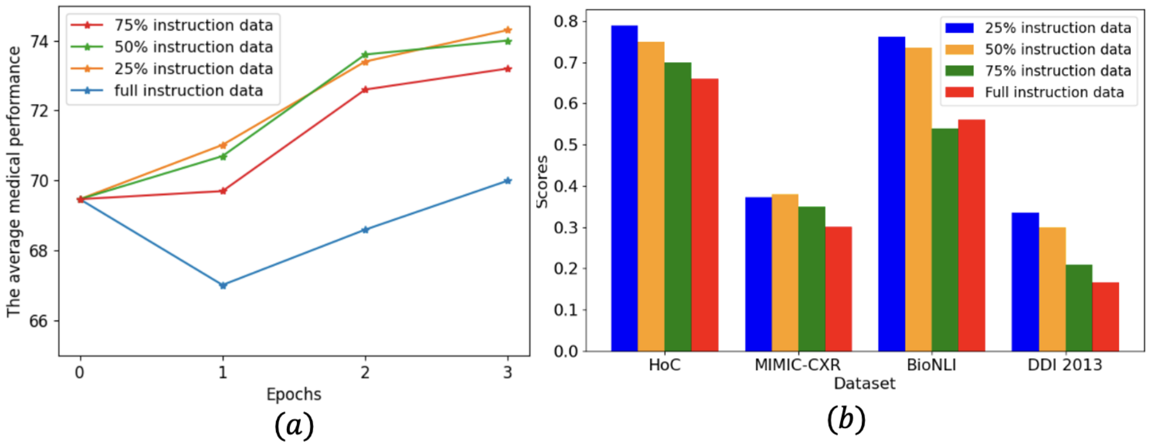


Figure 9: We consider the 'full instruction data' experiment as fine-tuning the model with all instruction data for 3 epochs. For the '$n\%$ data' experiments, we first uniformly sampled the highest quality instructions from each instruction dataset based on scores provided by the Deita data selector. We then mixed the sampled data with the general instructions from the Airoboros-3.2 dataset. The total training tokens are equal to $n\%$ of the full instruction data. We set n to 25, 50, and 75 here. (a) shows the experiments' average medical question-answering task performance during instruction tuning. (b) illustrates the experiments' performance for other medical tasks. For BioNLI, DDI 2023, and HOC tasks, we report macro-F1 as the score. For MIMIC-CXR summarization tasks, we report Rouge-L as the score.

Model	MMLU-Medical	PubMedQA	MedMCQA	MedQA-4-Option	Avg
Mistral-7B-instruct (Jiang et al., 2023)	55.8	17.8	40.2	41.1	37.5
Zephyr-7B-instruct-β (Tunstall et al., 2023)	63.3	46.0	43.0	48.5	48.7
PMC-Llama-7B (Wu et al., 2023)	59.7	59.2	57.6	49.2	53.6
Medalpaca-13B (Han et al., 2023)	55.2	50.4	21.2	20.2	36.7
AlpaCare-13B (Zhang et al., 2023b)	60.2	53.8	38.5	30.4	45.7
BioMedGPT-LM 7B (Zhang et al., 2023a)	52.0	58.6	34.9	39.3	46.2
Me-Llama-13B (Xie et al., 2024b)	-	70.0	44.9	42.7	-
Llama-3-8B instruct	82.0	74.6	57.1	60.3	68.5
JSL-Med-Sft-Llama-3-8B (johnsnowlabs, 2024)	83.0	75.4	57.5	59.7	68.9
GPT-3.5-turbo-1106	74.0	72.6	34.9	39.3	60.6
GPT-4 (OpenAI, 2023)	**85.5**	69.2	69.5	**83.9**	**77.0**
Llama-3-physician-8B instruct (ours)	80.0	**76.0**	**80.2**	60.3	74.1

Table 6: Accuracy comparison for question-answering tasks in the instruction-tuning setting.