Investigating Continual Pretraining in Large Language Models: Insights and Implications

Anonymous ACL submission

Abstract

Continual learning (CL) in large language models (LLMs) is an evolving domain that focuses on developing strategies for efficient and sustainable training. Our primary emphasis is on *continual domain-adaptive pretraining*, a process designed to equip LLMs with the ability to integrate new information from various domains while retaining previously learned knowledge and enhancing cross-domain knowledge transfer without relying on domain-specific identification. Unlike previous studies, which mostly concentrate on a limited selection of tasks or domains and primarily aim to address the issue of forgetting, our research evaluates the adaptability and capabilities of LLMs to changing data landscapes in practical scenarios. To this end, we introduce a new benchmark designed to measure the adaptability of LLMs to these evolving data environments, offering a comprehensive framework for evaluation. We examine the impact of model size on learning efficacy and forgetting, as well as how the progression and similarity of emerging domains affect the knowledge transfer within these models. Our findings uncover several key insights: (i) performance improves only if the adaptation corpora match the original pretraining scale, (ii) smaller models are particularly sensitive to continual pretraining, showing the most significant rates of both forgetting and learning, (iii) when the sequence of domains shows semantic similarity, continual pretraining enables LLMs to specialize better compared to stand-alone pretraining, and (iv) fine-tuning performance on standard benchmarks is indeed influenced by continual pretraining domains. We posit that our research marks a shift towards establishing a more realistic benchmark for investigating CL in LLMs.

1 Introduction

Recent advancements in the field of Natural Language Processing (NLP) have been significantly shaped by the development of large language models (LLMs) (Devlin et al., 2018; Radford et al., 2019; Brown et al., 2020). These models, trained on vast corpora from diverse domains, have emerged as versatile tools for numerous NLP tasks. However, the increasing scale and complexity of LLMs have raised concerns about the financial and ecological costs associated with training them from scratch (Luccioni et al., 2022). This has necessitated more efficient approaches than retraining these models entirely with each new data stream. Continual Learning (CL) emerges as a crucial strategy in this context (Sun et al., 2019; Biesialska et al., 2020) to reduce both financial and environmental costs while maintaining the models' relevance. CL, particularly through strategies like *continual fine-tuning*, which involves incrementally fine-tuning an LLM on a series of downstream tasks (Wu et al., 2021; Ramasesh et al., 2021; Scialom et al., 2022; Mehta et al., 2023) and *continual domain-adaptive pretraining*, focusing on incremental updates to adapt the LLM to new domains (Xu et al., 2019; Gururangan et al., 2020; Ke et al., 2023b) avoids the need for exhaustive retraining upon the arrival of new data.

In this paper, we delve into the challenges of continual domain-adaptive pretraining of LLMs. This process involves continuous training with large, unlabeled domain-specific corpora (Xu et al., 2019; Gururangan et al., 2020; Ke et al., 2023b). Given the dynamic nature of data and the emergence of new domains, LLMs must adapt to new information while retaining previously acquired knowledge and skills. A critical aspect of this adaptation is ensuring that knowledge transfer occurs seamlessly across domains without catastrophic forgetting (CF) (French, 1999) and operate effectively without explicit domain identification for each task.

Recent approaches in CL for LLMs have explored diverse methodologies, including parameter-efficient adaptation (Gururangan et al., 2021; Khan et al., 2022; Zhang et al., 2022), instruction fine-

L1 DOMAIN (ABBRV)	SIZE	#L2	#TOKENS	EXAMPLES OF L2 DOMAINS
Culture and The Arts (Culture)	1.8 GB	7	265M	Arts and entertainment, Sports and Recreation
History and Events (History)	1.2 GB	3	208M	Region, Period
Technology and Applied Sciences (Tech)	1.7 GB	4	268M	Agriculture, Computing
Health and Fitness (Health)	739 MB	6	99M	Exercise, Nutrition
Religion and belief systems (Religion)	341 MB	3	48 M	Belief Systems, Major beliefs of the world
General reference (GeneralRef)	196 MB	2	39M	Reference works
Philosophy and thinking (PhilThink)	721 MB	2	124M	Philosophy, Thinking
Art	578 MB	1	98 M	–
Philosophy	919 MB	1	156M	–
Quantitative Biology (Bio)	1.9 GB	11	336M	Biomolecules, Cell Behavior
Physics	4.1 GB	22	737M	General Physics, Biological Physics
Condensed Matter (CondMat)	3.5 GB	9	570M	Materials Science, Quantum Gases
Nonlinear Sciences (Nlin)	730 MB	5	134M	Self-Organizing Systems, Chaotic Dynamics
Mathematics (Math)	4.5 GB	30	1.4B	Topology, Number Theory
Statistics (Stat)	2.4 GB	6	450M	Applications, Methodology
Economics (Econ)	67 MB	3	11M	Econometrics, Theory
Computer Science (CS)	4.5 GB	39	1.1B	Machine Learning, Graphics
Astrophysics (Astro)	3.1 GB	5	562M	Earth/Planetary, Cosmology
Total	32.4 GB	159	6.6B	–

Table 1: The details of the L1 domains used in our experiments. Note that Art and Philosophy did not have any subdomains in M2D2 dataset.

tuning (Scialom et al., 2022; Razdaibiedina et al., 2023; Luo et al., 2023b), and continual pretraining (Qin et al., 2022; Ke et al., 2023a) to mitigate forgetting. A recent survey by Wu et al. (2024) provides a comprehensive overview of these efforts. Specifically, within continual pretraining, Cossu et al. (2022) explored the characteristics of forgetting across ten domains, Gupta et al. (2023) examined warm-up strategies, Wang et al. (2023) proposed orthogonal adapters to reduce domain interference, Qin et al. (2022) designed an expanding architecture, and Luo et al. (2023a) investigated forgetting in continual classification.

Most related to our work, Gururangan et al. (2020) evaluated the transfer capabilities of a RoBERTa model continually pretrained across four domains. However, given their diverse training data and foundational capabilities, one would expect LLMs to be adaptable across multiple domains rather than limited to just one. Perfect adaptation to a series of domains would also prevent the practitioners from re-training upon new data as promised by CL. Unfortunately, the field still lacks a comprehensive assessment of LLMs of various sizes and architectures in such a more realistic, large-scale setting.

Our work bridges this gap by pretraining LLMs across diverse domains and evaluating their performance throughout the pretraining process, setting our research apart from previous studies limited to a narrow domain focus (Cossu et al., 2022; Wang et al., 2023; Ke et al., 2023b). We leverage the Massively Multi-Domain Dataset (M2D2) (Reid et al., 2022), featuring 236 hierarchically organized domains from Wikipedia and Semantic Scholar. This dataset offers an ideal setting for examining CL across various LLMs, facilitating an in-depth analysis of forgetting and knowledge transfer over extensive training sequences.

Our key contribution is to evaluate pretrained LLMs within an extensive continual learning setting, focusing on the impact of model scale and architecture on their ability to learn new tasks and retain previously learned information. We also investigate the role of domain similarity and the order of appearing domains on knowledge transfer and the overall CL performance. Our findings uncover several key insights: (i) the amount of data for effective continual domain-adaptive pretraining depends on the size of the adapted model, (ii) continual pretraining influences the smaller models the most, (iii) before pretraining a model on a particular domain, training it on related domains leads to improved forward and backward transfer to that domain, and (iv) fine-tuning performance on standard benchmarks is indeed influenced by continual pretraining domains.

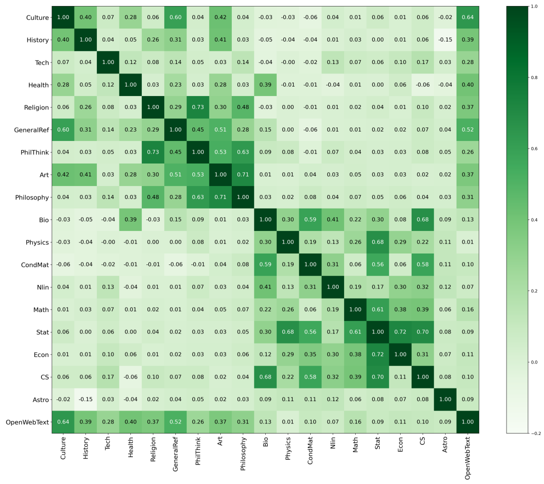

Figure 1: Cosine similarity between our L1 training domains. We also include OpenWebText (Gokaslan and Cohen, 2019), an open-source replication of the GPT2 pretraining data set. The two big square blocks along the diagonal correspond to Wiki and S2ORC portions.

2 Methodology

In this section, we describe our training process, provide an overview of the tasks (domains) used for the the continual pretraining and assessment of models, and explain the evaluation pipeline.

Training. We initiate our process with a pretrained LLM $\mathcal{M}_0$ that has been already trained on a comprehensive corpus $\mathcal{D}_0$. It is important to note that $\mathcal{D}_0$ generally represents a broad or general domain, such as a book corpus or web content. We then consider a series of domain corpora, $\mathcal{S}_N = \{\mathcal{D}_1, \cdots, \mathcal{D}_N\}$, from N domains. In our setting, each task $\mathcal{D}_i$ is an unlabeled domain-specific corpus. Our goal is to continuously pretrain an LLM on these sequential domain corpora by using the original training objectives, e.g., the next token prediction likelihood for autoregressive LLMs. At each stage i, the LLM $\mathcal{M}_{i-1}$ is trained on a new corpus $\mathcal{D}_i$, resulting in an updated model $\mathcal{M}_i$. Unlike conventional continual learning approaches where each task is an end-task, in our method, once a domain corpus is used for training, it is no longer available. In a typical continual learning scenario, each task involves end-task fine-tuning to evaluate the performance of the continually trained LLM.

Tasks. Our experiments are conducted on the M2D2 dataset (Reid et al., 2022), which is an extensive and finely categorized corpus specifically designed for exploring domain adaptation in language models. It comprises 8.5 billion tokens and covers 236 domains, sourced from Wikipedia and the Semantic Scholar (S2ORC) database (Lo et al., 2019). This dataset is unique in its combination of fine domain granularity and a human-curated domain hierarchy, set within a multi-domain context.

The corpus is divided into two levels: L1-domains and L2-domains. In the context of the S2ORC corpus, L1-domains refer to broad fields of academic research, such as Computer Science and Physics, while L2-domains correspond to specific arXiv categories within these fields, like "Computation and Language" under Computer Science. For Wikipedia, L1-domains represent major categories, and L2-domains encompass category pages within each L1 domain. To maintain balance and computational efficiency in our experiments, we excluded domains exceeding 5GB of data, such as Medicine. Ultimately, we utilized 159 domains in our study (see Table 1 for details).

To show the cross-domain similarity, we first computed the task embedding by using Sentence-BERT (Reimers and Gurevych, 2019) with 10K samples from each domain and 50K samples from OpenWebText (Gokaslan and Cohen, 2019), an open-source reproduction of GPT2 training dataset (Radford et al., 2019). Then we computed cosine similarities between each task pair (Figure 1). For the *similar-order* experiments detailed in the next section, we order the training domains based on their similarity, starting with the Culture domain, which is the most similar to OpenWebText, and then proceeding to the next most similar domains. Also see Figure 8 for the average L1 embeddings visualized using t-SNE.

Evaluation. Each domain in the M2D2 dataset is split into train, validation, and test sets with no data leakage, as outlined in Reid et al. (2022). Each validation and test set includes over 1 million tokens, allowing accurate evaluations within specific domains. We measure the effectiveness of all methods by testing perplexity on L2 domain test sets. For continual domain-adaptive pretraining experiments, after completing training on a domain for one epoch, we checkpoint the model, and compute the test perplexity for current and previous domains.

3 Experimental Setup

Models and training. We benchmark continual learning of existing pretrained LLMs with dif-

				M2D2-SIMILAR		M2D2-RANDOM	
	Num. pars.	Zero shot	Pre-training	Continual pretraining	Last checkpoint	Continual pretraining	Last checkpoint
GPT2-small	117M	27.90	20.36 (-7.54)	19.46 (-8.44)	27.52 (-0.38)	21.04 (-6.86)	22.47 (-5.43)
GPT2-medium	345M	21.54	18.58 (-2.96)	16.58 (-4.96)	20.11 (-1.43)	16.84 (-4.7)	19.01 (-2.53)
GPT2-large	774M	18.89	14.43 (-4.46)	14.33 (-4.56)	18.68 (-0.21)	15.13 (-3.76)	17.19 (-1.7)
GPT2-xlarge	1.5B	17.36	12.43 (-4.93)	12.28 (-5.08)	15.29 (-2.07)	13.89 (-3.47)	15.24 (-2.12)
Llama2-7B	7B	6.87	23.5	8.54	14.86	10.09	12.02

Table 2: This table shows test perplexities (↓) with different model sizes and training orders. For reference, we include the zero-shot and fine-tuning perplexities. Please see Table 4 for results obtained on Wiki and S2ORC domains. Inside the parentheses are the perplexity improvements over zero-shot (the smaller the better).

ferent architectures and sizes. In particular, we consider *(1)* decoder-only models (GPT2-small, GPT2-medium, GPT2-large and GPT2-xlarge, Llama2-7B) as well as *(2)* encoder-decoder models (RoBERTa-base and RoBERTa-large (Liu et al., 2020)). Due to space considerations, we present RoBERTa findings in the Appendix. We trained the models with Adam optimizer (Kingma and Ba, 2015) with a batch size of 16 sequences on NVIDIA A100 GPUs. We used DeepSpeed (Rasley et al., 2020) with auto configuration, which assigns a dropout rate of 0.2 and a learning rate of 5e-5.

Task ordering. In order to investigate how the order of training domains impacts our domain-incremental continual learning setup, we ordered the tasks in our experiments in two different ways: *(i) similar-order* where semantically related domains follow one another, and *(ii) random-order*, where the domains are shuffled.

Metrics for assessing continual learning efficacy. To evaluate the effectiveness of continual learning, we begin by setting two baselines for comparison, *zero-shot perplexity (ZS)* which measures the innate ability of the original, unmodified models to predict outcomes without any domain-specific tuning and *pretraining perplexity (PT)* which evaluates the models after they have been specifically pretrained for each domain. *ZS* acts as a fundamental baseline, ensuring that our models have a basic level of competence and *PT* sets a targeted performance standard for our continual learning approach to surpass. Achieving a better perplexity than the *PT* baseline is the primary objective for continual pretraining, signifying that longer training horizons is more favorable than domain adaptive pretraining.

To assess continual learning performance, we compute *continual pretraining perplexity (CPT)* where we evaluate a model's performance on the most recent training domain. This measure helps us understand how well the model adapts to new information over time. Moreover, we compute the *last checkpoint (LC)* against all the training domains to examine the final model's ability to retain and transfer knowledge across a broad range of subjects. Finally, we evaluate checkpoints on previously seen/unseen domains to measure backward/forward transfer.

Through these metrics, we aim to thoroughly understand continual learning dynamics, focusing on model adaptability, knowledge retention, and ability to generalize across various domains. To express the metrics more explicitly, let z_n and f_n denote the zero-shot and pretraining perplexities on n'th domain. Further, let p_n^c denote the perplexity of c'th checkpoint on n'th domain (notice that $c > n$ and $c < n$ correspond to backward and forward transfer). Then the main metrics of our interest are computed as follows: ZS $= \frac{1}{N}\sum_{n=1}^{N} z_n$, PT $= \frac{1}{N}\sum_{n=1}^{N} f_n$, CPT $= \frac{1}{N}\sum_{n=1}^{N} p_n^n$, LC $= \frac{1}{N}\sum_{n=1}^{N} p_n^N$.

4 Results and Findings

In this section, we discuss our main findings. We first discuss how the model and data scale impact continual learning. Next, we examine the implications of the order of training domains. Our only positive forward transfer finding follows this and then we analyze fine-tuning performances on benchmark tasks. Finally, we list our remaining observations. Please see Section A.1 for additional findings that do not fit into the main manuscript and Section A.2 for ablation studies.

4.1 Model scale

Is continual learning even necessary? Comparing the zero-shot performance against all other metrics in Table 2, we most strikingly observe that Llama2 does not benefit from CL or domain-adaptive pretraining. In contrast, test perplexities of all GPT2 models improve thanks to CL. These findings imply that models trained on enormous data corpora may already perform better than their domain-adapted versions. Please note that we did not observe any training issues during learning, i.e., training perplexity always improved.

Final performance correlates with model size In agreement with the recent research on scaling laws (Kaplan et al., 2020; Bahri et al., 2021), CL of bigger models results in better CPT and LC performance regardless of the training order. However, this consistent pattern cannot solely be attributed to CL since model size heavily influences zero-shot performance. Taking the zero-shot performance as a baseline (see the values inside the parentheses in Table 2), we observe that GPT2-small benefits the most from continual pretraining by a large margin in three out of four evaluated scenarios.

4.2 Recency effect in continual learning

CPT is more favorable to standard PT when domains are semantically ordered Comparing the PT column against CPT columns reveals that continual pretraining outperforms stand-alone pretraining only if subsequent training domains have high similarity. This observation aligns with the premise that a model's performance on a current task is intrinsically linked to its starting checkpoint. Meaning, when training domains are ordered based on similarity, the last checkpoint $\mathcal{M}_i$ naturally transfers better to the next domain $\mathcal{D}_{i+1}$ compared to the original model $\mathcal{M}_0$. On the other hand, when training domain order is randomized, starting from the original model $\mathcal{M}_0$ is found to be more beneficial. This observation aligns with recent studies on continual test-time adaptation (Press et al., 2023) and checkpoint selection strategies (Fisch et al., 2023), highlighting the strategic value of starting points in training sequences.

Similar training order facilitates backward transfer to recent past Figure 3 demonstrates the interplay between backward transfer and training horizon. In particular, x-axis shows how many tasks have passed between a checkpoint and a domain it is tested onx, i.e., we plot $\frac{1}{N-x}\sum_{c=x}^{N}(p_{c-x}^{c} - z_{c-x})$ against x. The panel on the left reveals that a high conceptual overlap between subsequent domains leads to positive backward transfer up to 30 domains back. Naturally, the improvement worsens over time as the recent training domains become significantly dissimilar to tested domains. Notably, the smallest GPT model shows the most significant fluctuations in performance, experiencing both the highest gains and the most pronounced declines.

Average backward transfer performance depends heavily on domain order We present the average backward perplexities in Figure 2. We normalize the perplexities by subtracting zero-shot perplexities obtained on the same domains, i.e., we plot $\frac{1}{x-1}\sum_{n=1}^{x-1}(p_n^x - z_n)$ against checkpoint id x. On the one hand, we never observe positive backward transfer with similar-order training, and test perplexity notably degrades when we switch the training portion. On the other hand, training in random order generally enhances test perplexity compared to the zero-shot baseline. The most significant improvement over the initial model $\mathcal{M}_0$ is observed early in training and saturates after about 25 tasks.

4.3 Random-order training enables positive transfer to S2ORC

The previous finding demonstrates that training in random order significantly enhances transfer to past tasks. Figure 4 visually shows a similar effect for future transfer: the perplexity tends to improve, compared to zero-shot baseline, as a function of the number of continual pretraining domains before forward transfer. Specifically, we plot $\frac{1}{|\mathbb{S}|}\sum_{n\in\mathbb{S}} p_n^x - z_n$ against x, where $\mathbb{S}$ is the set of future Wikipedia and S2ORC domains for the green and pink curves.

Noticeably, positive forward transfer is possible only to the S2ORC portion since all values corresponding to the Wikipedia test portion are positive, implying no perplexity improvement when tested on the Wikipedia portion. This discrepancy is rather expected as the S2ORC portion is about five times larger than the Wikipedia portion. Further, test perplexity on the S2ORC portion consistently improves with the number of pretraining tasks, i.e.,

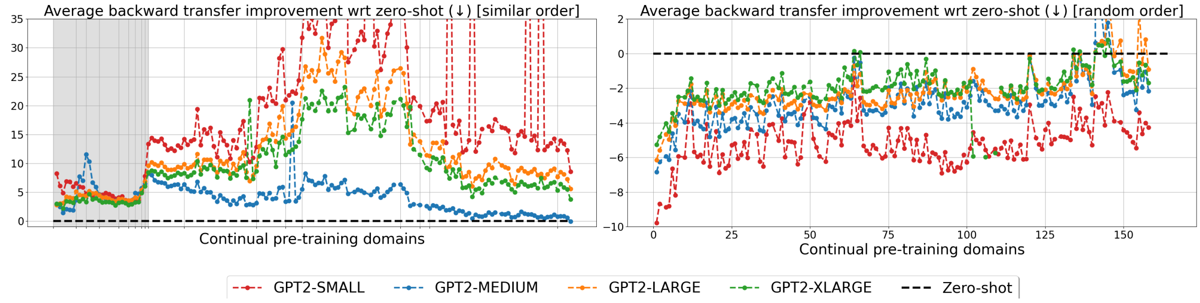

Figure 2: Backward transfer perplexity (averaged over all past domains, y axes) during the course of learning (x axes). The grey background highlights Wiki domains. Similar-order always leads to negative backward transfer while randomizing the domains significantly improves transfer.

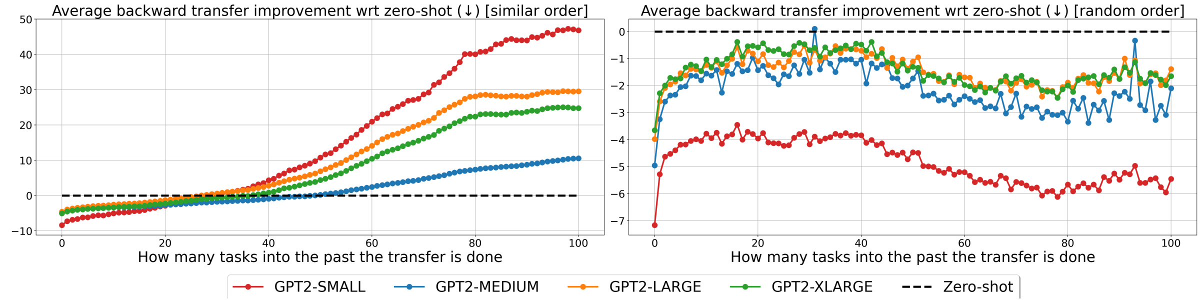

Figure 3: Average backward transfer performance (normalized by zero-shot, y axes) as a function of the number of tasks between the checkpoint and the tested domain (x axes). The first points in each curve correspond to continual pretraining (no backward transfer, upper bound). The benefits of continual learning when trained in similar-order steadily degrade with the transfer distance while it tends to improve with random-order training.

longer training improves forward transfer. We conclude that the model accumulates knowledge that is on average beneficial to predict next tokens on unseen finer-grained domains.

4.4 Preservation of general knowledge through benchmark tasks

We utilized a selection of tasks from BIG-bench (bench authors, 2023) aimed at assessing whether the general knowledge embedded in the original language model remains intact, experiences significant loss, or achieves effective knowledge integration post-training. Specifically, we chose five tasks aligned with our benchmark domains: Arithmetic, General Knowledge, Physics, CS Algorithms, and Few-shot Natural Language Generation (NLG). Given space limitations, we detail the results for Arithmetic and General Knowledge in Figure 5, while comprehensive task descriptions, metrics, and additional outcomes are provided in the Appendix A.3.

Until now, our evaluation has centered on assessing the language modeling capabilities of our models, specifically using perplexity as our performance metric. Moving forward, we assess their performance on different tasks, revealing that continuing pretraining on domains relevant to these tasks generally enhances model performance, while pretraining on unrelated domains often leads to forgetting, thereby negatively affecting the model's initial task proficiency. As depicted in Figure 5, a consistent decrease in Arithmetic task performance was noted when models were continually trained on Wiki domains which then improves upon switching to S2ORC domains, with the exception of the Nonlinear Sciences and Astrophysics domains. In contrast, performance on General Knowledge tasks improved with Wiki domain training but declined with S2ORC training, except for slight increases in the CS and Statistics domains.

4.5 Additional observations

Within GPT family, the final checkpoint achieves better perplexity than zero-shot Our study demonstrates that the final model consistently outperforms or matches the zero-shot baseline in terms of perplexity across different domain sequences and model sizes. We report the average perplexity over all domains, suggesting that the knowledge accumulated throughout CL never

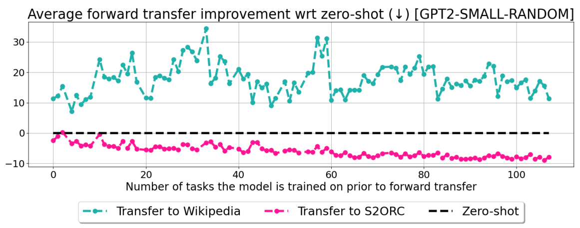

Figure 4: Forward transfer results with random training order. The x axis shows the number of domains the model is trained on before forward transfer. Curves show the perplexity (normalized by zero-shot). Clear positive/negative forward transfer to S2ORC/Wiki portions is observed.

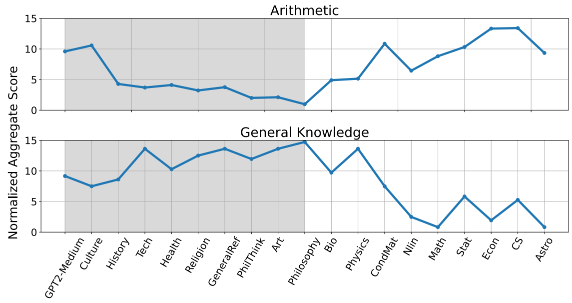

Figure 5: GPT2-medium performance on Arithmetic and General Knowledge tasks from BIG-Bench, captured at L1 domain transitions. The initial data point represents the baseline performance of GPT2-medium.

hurts the predictions on the learned domains on average. Notably, randomizing the training sequence results in a more favorable average perplexity than a similar-order domain sequence. We present a detailed comparison of perplexity values for Wiki and S2ORC portions in Table 4, highlighting the GPT family's tendency to forget the Wiki portion while improving perplexity on S2ORC.

Longer CL improves backward transfer if domain order is randomized The right panel in Figure 3 reflects a rather surprising finding that when the training domains are presented in a randomized order, we observe positive backward transfer (w.r.t. zero-shot performance). This is consistent across different model sizes and the number of tasks between the checkpoint and test domain. Remarkably, the perplexity improvement increases when a checkpoint is tested further back in time (evidenced by the downward trend in the curves). We interpret this finding as an indicator for knowledge accumulation, i.e., performance on previously learned domains increases on average when the model is trained longer on a randomized set of domains, even if only a handful of which are conceptually similar.

5 Related work

We discuss two related but separate lines of research in the context of CL for LLMs: (i) continual fine-tuning, which aims at fine-tuning LLMs on a series of downstream tasks, and (ii) continual domain-adaptive pretraining, focusing on incremental updates to adapt an LLM to new domains without exhaustive retraining from scratch upon new data.

Continual fine-tuning A large body of CL works for LLMs tries to mitigate forgetting during continual fine-tuning. (Luo et al., 2023a) investigate forgetting and distribution drift during continual learning on a series of eight downstream classification tasks. In a recent work, (Luo et al., 2023b) examines evolution of forgetting during continual fine-tuning. Scialom et al. (2022) instruct fine-tune an LLM for eight tasks. Khan et al. (2022) introduce an adapter-based fine-tuning strategy for three downstream tasks. Zhang et al. (2022) propose to add new modules to a sequence generator (such as an LLM) to continually adapt to five tasks. Razdaibiedina et al. (2023) introduce progressive prompts, where a growing number of prompts, are learned during continual learning, fine-tunes on 15 classification datasets. Wang et al. (2023) propose to learn orthogonal adapters to minimize interference between 15 classification tasks. Qin et al. (2022) propose efficient lifelong pretraining for emerging data (ELLE), where they expand a network during learning and include domain-identifying prompts during pretraining to help the PLM identify the type of knowledge it is learning.

Continual domain-adaptive pretraining An alternative research direction, closer to our work, aims to continually pretrain LLMs to adapt them to new domains. In one of the earliest studies, Gururangan et al. (2020) introduce a growing mixture of expert architecture for domain-adaptive continual pretraining. Chen et al. (2023) study lifelong learning from a sequence of online pretraining corpus distributions based on a progressively growing mixture-of-experts (MoE) architecture. Likewise, Gururangan et al. (2021) introduce a mixture architecture for continual adaptation. Ke et al. (2023a) show how a soft-masking mechanism for gradients of RoBERTa model could be useful for domain-adaptive pretraining for eight tasks. Cossu et al. (2022) investigate the characteristics of the continual pretraining across ten domains. Jin et al. (2021) continually pretrain RoBERTa-base over a domain-incremental research paper stream and a chronologically-ordered tweet stream with different continual learning algorithms. Gupta et al. (2023) examine different warm-up strategies for continual pretraining. Finally, Fisch et al. (2023) introduce a benchmark of task sequences that potentially lead to positive and negative transfer and further propose a simple strategy for robust forward transfer, which aims to pick the checkpoint with the biggest positive knowledge transfer among all past task checkpoints. Our work diverges from the others in that we continually pretrain the original model without any expansion on a much longer horizon of 159 domains, and further investigate the impact of domain order.

6 Discussion

Prior studies in CL for LLMs have mainly focused on parameter-efficient fine-tuning or adaptation for a limited selection of target domains or tasks. While beneficial, these methods often do not fully address the broader challenge of lifelong learning for LLMs. Our research diverges by exploring continual domain-adaptive pretaining of LLMs across an extensive set of domains to better understand the dynamics of knowledge preservation, new information retention and knowledge transfer. Below, we highlight three key insights and discuss three notable observations from our research, supported by indicative evidence:

Semantic similarity enhances domain specialization in CL. We found that when consecutive domains are semantically similar, CL allows LLMs to specialize more effectively in the current domain than stand-alone pretraining. This is supported by two findings: *(i)* continual pretraining is more advantageous than pretraining alone, likely due to the accumulated knowledge from slowly evolving domains, and *(ii)* models exhibit positive transfer to recent past domains but not to more distant domains in the training chronology.

Randomizing training domain order significantly improves knowledge accumulation. With the randomized training order, we notice *(i)* the last checkpoint demonstrates superior performance on average than similar-order training, *(ii)* a majority of checkpoints exhibit positive backward transfer on average to the past, effectively implying that previous knowledge remains somewhat intact, and *(iii)* continually pretraining for longer improves forward transfer, signifying better generalization ability of the model.

Continual pretraining enhances downstream task performance. Our experiments on Big-Bench indicate that the performance on downstream tasks such as question-answering is closely related to the domains the model was trained on. This evidences that further generative pretraining prior to fine-tuning can enhance downstream performance in comparison to fine-tuning alone.

Evidence for knowledge saturation. Categorizing checkpoints based on their timestamp reveals that forgetting becomes more severe over time. This pattern shows that the model's capacity for integrating new information gradually reaches a plateau, which we refer to as knowledge saturation.

Rethinking scaling laws for CL. In almost all experiments with GPT2 model family, CL caused the biggest improvement or forgetting on GPT2-small models compared to other model sizes. However, the relationship between model size and performance improvements is not always straightforward. For instance, GPT2-large demonstrated poorer backward transfer perplexity compared to both GPT2-medium and GPT2-xlarge, challenging the conventional wisdom that larger models uniformly translate to better performance. Besides, the performance of the Llama2-7B deteriorated as a result of continual pretraining. It is important to note that Llama2-7B models are initially trained on a vast dataset comprising 2 trillion tokens from a wide array of domains. In contrast, GPT-2 models are trained using the OpenWebText dataset, which contains 9 billion tokens. This observation suggests that continual pretraining may not be beneficial for models like Llama2-7B that have already been trained on an extensive and diverse corpus. Therefore, we infer that unless there is a substantial amount of domain-specific data available, further pretraining of Llama2-7B models is unlikely to yield performance improvements. This finding underscores the importance of having sufficient and relevant data when considering additional pretraining for models that are already well-trained on diverse datasets.

7 Limitations

Our research highlights CL as a powerful paradigm for learning in LLMs, providing valuable insights into its mechanisms and benefits. However, we acknowledge several limitations in our study: (i) For our random-order training, domains were shuffled only once. We perform an ablation study with GPT2-medium by considering two more random shuffles, whose conclusions may not immediately transfer to other experiments. (ii) Exploring how backward transfer performance to a domain is affected by its size or similarity to Webtext could yield interesting insights. (iii) Since part of RoBERTa training data contains Wikipedia entries, which may overlap with our training set, this could influence our RoBERTa results.

To report average backward transfer perplexity, we exhaustively tested all checkpoints on all past domains, resulting in 12561 evaluations per model per setup. Consequently, we evaluated forward transfer after completing all L2 domains in a certain L1 domain, which still required 171 evaluations per model per setup.

The computation time is an inevitable limitation in our experimental setup. For instance, one pretraining run and backward evaluation for GPT2-Large takes approximately two months on two A100 GPUs. Given that we run our experiments on 159 tasks, the incremental nature of continual pretraining prevents parallelization of the training process.

References

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. 2021. Explaining neural scaling laws. *arXiv preprint arXiv:2102.06701*.

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R Costa-Jussa. 2020. Continual lifelong learning in natural language processing: A survey. *arXiv preprint arXiv:2012.09823*.

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

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

Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. 2022. Continual pre-training mitigates forgetting in language and vision. *arXiv preprint arXiv:2205.09357*.

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

Adam Fisch, Amal Rannen-Triki, Razvan Pascanu, Jörg Bornschein, Angeliki Lazaridou, Elena Gribovskaya, and Marc'Aurelio Ranzato. 2023. Towards robust and efficient continual language learning. *arXiv preprint arXiv:2307.05741*.

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

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

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

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A Smith, and Luke Zettlemoyer. 2021. Demix layers: Disentangling domains for modular language modeling. *arXiv preprint arXiv:2108.05036*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

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

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

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023a. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*.

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

Shadab Khan, Surbhi Agarwal, and PK Srijith. 2022. Lifelong language learning with adapter based transformers. In *Continual Lifelong Learning Workshop at ACML 2022*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Roberta: A robustly optimized bert pretraining approach.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S Weld. 2019. S2orc: The semantic scholar open research corpus. *arXiv preprint arXiv:1911.02782*.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. 2022. Estimating the carbon footprint of bloom, a 176b parameter language model. *Preprint*, arXiv:2211.02001.

Yun Luo, Zhen Yang, Xuefeng Bai, Fandong Meng, Jie Zhou, and Yue Zhang. 2023a. Investigating forgetting in pre-trained representations through continual learning. *arXiv preprint arXiv:2305.05968*.

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

Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. 2023. An empirical investigation of the role of pre-training in lifelong learning. *Journal of Machine Learning Research*, 24(214):1–50.

Ori Press, Steffen Schneider, Matthias Kümmerer, and Matthias Bethge. 2023. Rdumb: A simple approach that questions our progress in continual test-time adaptation. *arXiv preprint arXiv:2306.05401*.

Yujia Qin, Jiajie Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. Elle: Efficient lifelong pre-training for emerging data. *arXiv preprint arXiv:2203.06311*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. 2021. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. *arXiv preprint arXiv:2301.12314*.

Machel Reid, Victor Zhong, Suchin Gururangan, and Luke Zettlemoyer. 2022. M2d2: A massively multi-domain language modeling dataset. *arXiv preprint arXiv:2210.07370*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6107–6122.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. Lamol: Language modeling for lifelong language learning. *arXiv preprint arXiv:1909.03329*.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023. Orthogonal subspace learning for language model continual learning. *arXiv preprint arXiv:2310.14152*.

Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan-Fang Li, Guilin Qi, and Gholamreza Haffari. 2021. Pretrained language model in continual learning: A comparative study. In *International Conference on Learning Representations*.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*.

Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. 2022. Continual sequence generation with adaptive compositional modules. *arXiv preprint arXiv:2203.10652*.

Section A.1	Additional findings
Section A.2	Ablation studies
Section A.3	Details and discussion of the BIG-Bench experiment
Section A.4	RoBERTa results
Table 4	Perplexities computed on Wiki and S2ORC portions separately
Table 5	RoBERTa results
Figure 6	Forgetting in two different stages of CL
Figure 7	Visualization of the BIG-Bench results
Figure 8	t-SNE embeddings of L1-domains
Figure 9	GPT2-large backward transfer performance for four example domains
Figure 10	CPT and LC perplexities obtained with batch size 16 vs 64
Figure 11	CPT and LC perplexities obtained when the dataset is balanced
Figure 12	CPT and LC perplexities obtained when portions are swapped
Figure 13	Forward transfer results for all L1 domains (GPT2 and RoBERTa)
Figure 14	RoBERTa-large backward transfer performance for four example domains

Table 3: A summary of our appendix

A Appendix

A.1 Additional findings

LLMs forget more in the later stages of continual learning We divide the checkpoints in random-order training into two groups based on their recency (checkpoints[50-100] and checkpoints[100-150]). We evaluate each checkpoint on 50 domains back and compute the perplexity change (caused by additional training on 50 domains). Histograms in Figure 6 show that earlier checkpoints transfer to the past slightly better. We hypothesize that in the earlier stages of training, the parameters that are not *important* to the recently learned tasks are updated and the lack of such parameters causes more forgetting in the later stages.

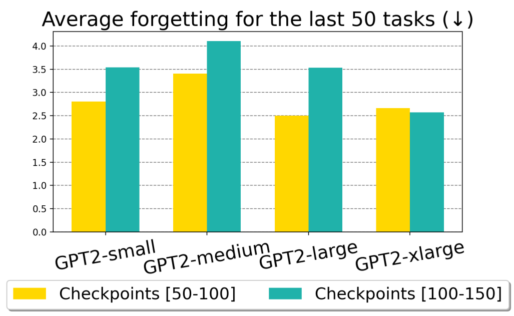

Figure 6: We divide the checkpoints in random-order training into two groups based on their recency, showing that earlier checkpoints transfer better to the past.

Positive forward transfer is rarely possible in similar training order Each plot in Figure 13 shows the forward transfer performance to the domain stated in the title. Most notably, the leftmost panel reflects that pretraining on a handful of domains leads to significantly worse performance compared to zero-shot (the dotted horizontal lines). In contrast, extended pretraining across a variety of domains occasionally leads to positive forward transfer (panels 2 and 3). Further, we notice a *recency effect* where the forward transfer perplexity improves if a checkpoint is transferred to a domain that is conceptually similar to the most recent training domain: as anticipated, the most successful forward transfer to Astrophysics domains is attained after training on Physics. Please see Figure 13 for complete results.

A.2 Ablation studies

Batch size impacts learning dynamics As an ablation study, we increase the batch size from 16 to 64, thereby performing a quarter of gradient updates. Figure 10 compares the results obtained with different batch sizes. When trained in random-order, continual pretraining and last checkpoint performances virtually remain the same despite varying the batch size. In similar-order, a smaller batch size helps to improve continual pretraining perplexity but worsens the performance of the last checkpoint. We hypothesize that taking more gradient steps aids the model to better fit the current task while promoting forgetting the old tasks.

Balancing the data size across L2 domains does not improve performance We investigate whether the imbalance in training data sizes impacts the overall performance (see Figure 1 for L1 domain lengths). To address this, we set the number of maximum tokens to 100K for each L2 domains (if they have less tokens, we used them all), and train the original model. Figure 11 shows the resulting continual pretraining and last checkpoint perplexities. For both metrics, test performance on almost all L2 domains deteriorates after balancing the number of data points per domain. The results suggest using all data at hand instead of leaving some out for the sake of balanced training.

Swapping Wiki and S2ORC portions verifies previous findings We swap the portions for similar-order training, i.e., training first on S2ORC, then on the Wiki portion. Arguably, this training order still follows conceptual similarity; hence, it allows us to see whether our previous findings still hold. The left panel in Figure 12 shows that continual pretraining perplexity remains almost the same. Yet, the last checkpoint perplexity significantly changes: while the performance on the S2ORC portion substantially degrades, we observe the opposite effect for the Wiki portion. Agreeing with our previous findings, we conclude that the checkpoints perform worse when tested on older domains/portion.

Alternative random orders yield similar findings In our random-order experiments, we consider only one randomized training sequence. To test whether the findings do not generalize to alternative randomized orders, we re-shuffle the dataset twice and repeat the experiments with GPT2-large. These experiments resulted in an average CPT of 16.4 and 16.78 while 16.84 in the main paper. Given relatively much larger differences across different experiment setups, we conjecture that the standard deviation resulting from different random orders can be safely ignored.

A.3 BIG-Bench Experiments

Tasks. We selected five tasks that align with our benchmark domains, as described below:

Arithmetic evaluates the model's ability in basic arithmetic operations – addition, subtraction, multiplication, and division – ranging from 1-digit to 5-digit numbers.

General Knowledge assesses the model's ability to answer questions across a broad spectrum of general knowledge, for example, "How many legs does a horse have?". It draws parallels with benchmarks focused on general-knowledge question-answering, such as those found in (Rajpurkar et al., 2016).

Physics aims to test the model's understanding of physics by asking it to determine which formula is needed to solve a given physics word problem, and evaluating the accuracy of the multiple choice responses. The decision to utilize a multiple-choice format concentrates on the model's comprehension of the physical principles each formula represents, addressing concerns that generating physics formulas through text might be overly challenging for current models.

CS Algorithms measures the model's performance on two core algorithmic concepts: recursion (or stack usage) and dynamic programming, evaluating the model's computational thinking and problem-solving skills.

Language Generation from Structured Data and Schema Descriptions (Few-shot NLG) aims to assess the ability of a model to generate coherent natural language from structured data, supported by schema descriptions, within the framework of a task-oriented dialogue system. The goal is to determine whether a virtual assistant can learn to generate responses based on the textual description of structured data, enabling rapid adaptation to new domains with minimal additional input.

Metric. In plots, we report *Normalized Aggregate Score*, that is normalized preferred metric averaged over all subtasks under that particular task. For example Arithmetic task has 20 subtasks. In (bench authors, 2023), they specify that the best performing language models achieved a score below 20 and model scores can be less than 0 on some tasks.

Results. Beyond the findings highlighted in Figure 5, additional task outcomes are detailed in Figure 7. The performance trends for the CS Algorithms and Physics tasks align with those observed

Test portion	Model	Zero shot	Pre-training	M2D2-SIMILAR Continual pretraining	M2D2-SIMILAR Final model	M2D2-RANDOM Continual pretraining	M2D2-RANDOM Final model
Wiki	GPT2-small	26.71	26.16	29.46	46.05	33.70	37.50
	GPT2-medium	20.42	24.11	23.81	28.98	26.65	32.78
	GPT2-large	17.77	17.77	20.42	30.18	23.23	28.96
	GPT2-xlarge	16.39	15.70	18.63	26.28	21.86	25.64
S2ORC	GPT2-small	28.18	19.00	16.98	23.18	18.07	18.95
	GPT2-medium	21.81	17.29	14.88	18.04	14.55	15.79
	GPT2-large	19.16	13.65	12.90	15.99	13.24	14.44
	GPT2-xlarge	17.59	11.65	10.78	12.72	12.08	12.80

Table 4: A more detailed analysis of our main results table. This time, we compute the test perplexities(↓) on Wiki and S2ORC portions separately.

for the Arithmetic tasks, as anticipated. Specifically, we notice a decline in performance for both the CS Algorithms and Physics tasks when models continue pretraining on Wiki domains. Performance then improves with a shift to S2ORC domains and reaches its peak after specialized training in the CS and Physics domains, respectively.

For the Few-shot NLG task, performance trends across Wiki and S2ORC domains do not follow a consistent pattern. Analysis reveals that domains such as Culture, Art, Philosophy, Math, Stat, and Econ contribute positively to performance enhancement in this task, while domains like History, General Reference, and Nonlinear Sciences are identified as the most detrimental to task performance.

A.4 Roberta Results

To broaden our analysis and gain deeper insights into the behavior of different architectures, we have repeated all the experiments with RoBERTa and obtained somewhat counter-intuitive and surprising results. First of all, we want to point out that perplexity is not well defined for masked language models like RoBERTa[1]. We used the same perplexity computation for RoBERTa with the one we used for GPT as in the following:

$$\mathrm{PPL}(X) = \exp\{-\frac{1}{t}\sum_{i}^{t} \log p_\theta(x_i|x_{<i})\} \quad (1)$$

[1] https://huggingface.co/docs/transformers/perplexity

where $X = (x_0, \ldots, x_t)$ is the tokenized sequence and $\log p_\theta(x_i|x_{<i})$ is the log-likelihood of the ith token conditioned on the preceding tokens $X_{<i}$ according to the model.

Forgetting in RoBERTa family is not evident In contrast to the GPT family, our analysis reveals that the RoBERTa family does not exhibit forgetting of old tasks during continual training. This indicates that gradient descent updates do not interfere with old tasks. To illustrate this, we present a visualization of backward transfer performance across four randomly selected domains in Figure 14. Similar findings with encoder-decoder models were reported in (Cossu et al., 2022). We conjecture that modifying the model architecture by including a bottleneck layer plays a significant role in this behavior.

RoBERTa-large always exhibits positive backward transfer while GPT2-large transfer performance depends heavily on the transferred domain Looking into Figure 14, we notice that backward transfer perplexity of RoBERTa-large remains relatively close to fine-tuning performance. Interestingly, we observe occasional jumps in perplexity when trained in random order, whose analysis is an interesting future work. On the other hand, Figure 9 demonstrates backward transfer to the same four domains when GPT2-large is trained. In agreement with our earlier findings, switching from Wiki portion to S2ORC causes a significant perplexity degradation on Wiki domains when trained

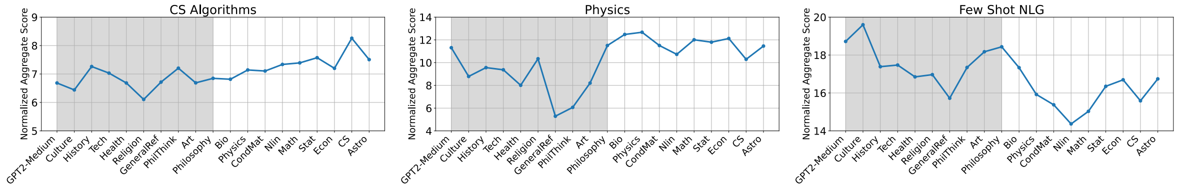

Figure 7: GPT2-medium performance on CS Algorithms, Physics and Few-shot NLG tasks, captured at checkpoints following training completion on an L1 domain. The initial data point represents the baseline performance of GPT2-medium.

in similar order. Further, the characteristics of the test domain seem to determine whether the transfer is positive or negative. Finally, we observe a less fluctuating backward perplexity with random training order.

Encoder-decoder models require just a few L1 domains for good transfer *(i)* In stark contrast with the decoder-only models, pretraining even on the first L1 domain helps to exceed zero-shot performance (comparing the dotted lines and the first point of each sequence). Interestingly, this holds when the pretraining and test domains belong to different portions of the training set. *(ii)* We further notice the forward transfer perplexity tends to improve for the first ten L1 domains and later slightly degrade. Since it is still considerably above zero-shot performance, we chose not to investigate this in detail. *(iii)* Lastly, the model size does not seem to influence forward transfer performance, which is again as opposed to decoder-only models.

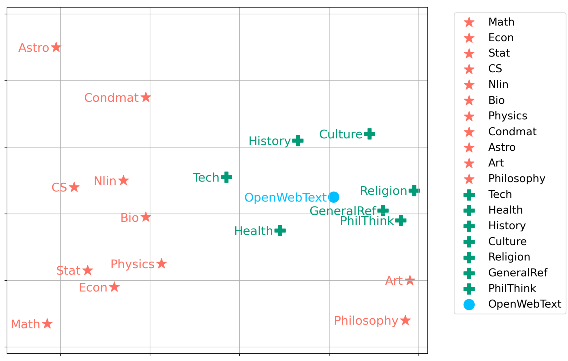

Figure 8: Average L1-domain embeddings visualized using t-SNE. Wiki domains and natural sciences form two clear clusters. Note that Art and Philosophy are from S2ORC portion, but they are closer to Wiki due to they are social sciences and the rest of S2ORC is natural sciences.

Figure 9: Backward transfer illustration with GPT2-large trained in similar and random order (left and right columns). Each panel shows the backward transfer perplexity (pink) computed on a particular domain as optimization proceeds. For baseline comparisons, we also plot zero-shot (yellow) and continual pretraining (green) perplexities.

Figure 10: A comparison of GPT2-small training with batch sizes 16 (our default) and 64. For random and similar training orders (rows), we plot the continual pretraining and last checkpoint perplexities (columns).

Test portion	Model	Zero shot	Pre-training	M2D2-SIMILAR Continual pretraining	M2D2-SIMILAR Final model	M2D2-RANDOM Continual pretraining	M2D2-RANDOM Final model
All	RoBERTa-base	1.97	1.73	1.54	1.46	1.27	1.26
	RoBERTa-large	4.98	3.10	2.43	2.34	1.37	1.28
Wiki	RoBERTa-base	1.93	1.69	1.49	1.43	1.45	1.48
	RoBERTa-large	4.73	2.95	2.38	2.32	1.61	1.54
S2ORC	RoBERTa-base	1.56	1.37	1.26	1.26	1.25	1.26
	RoBERTa-large	2.56	1.61	2.15	2.15	1.35	1.29

Table 5: Test perplexities obtained with RoBERTa family.

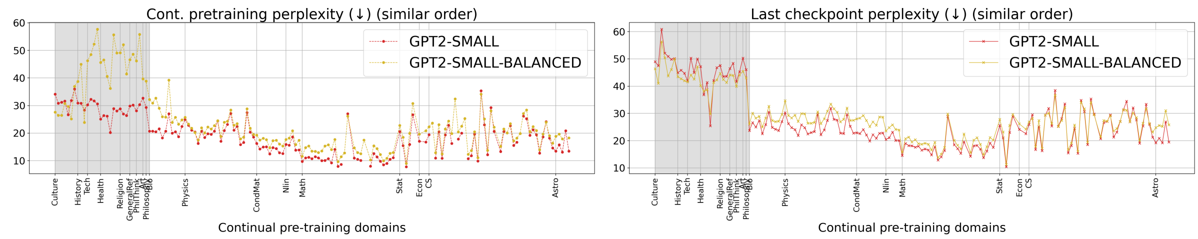

Figure 11: A comparison of GPT2-small training with all available data (our default) as well as a subsample of data with equally many data points per L2 domain. We only train in similar orders and plot the continual pretraining (left) and last checkpoint perplexities (right).

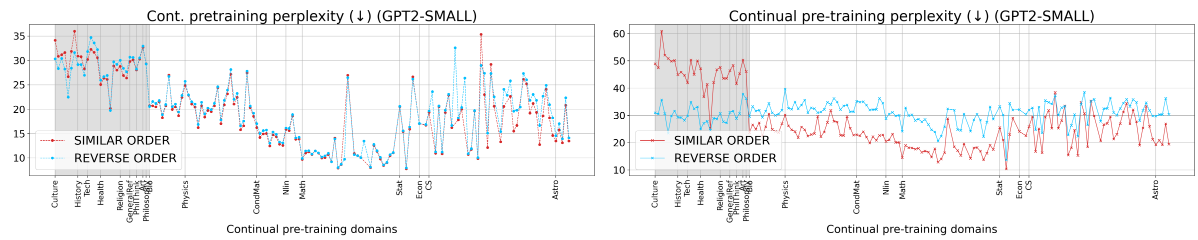

Figure 12: A comparison of GPT2-small training with our default similar training order (Wiki portion, followed by S2ORC) as well as an alternative version (S2ORC portion, followed by Wiki). We plot the continual pretraining and last checkpoint perplexities. Note that the x axis corresponds to the default training order.

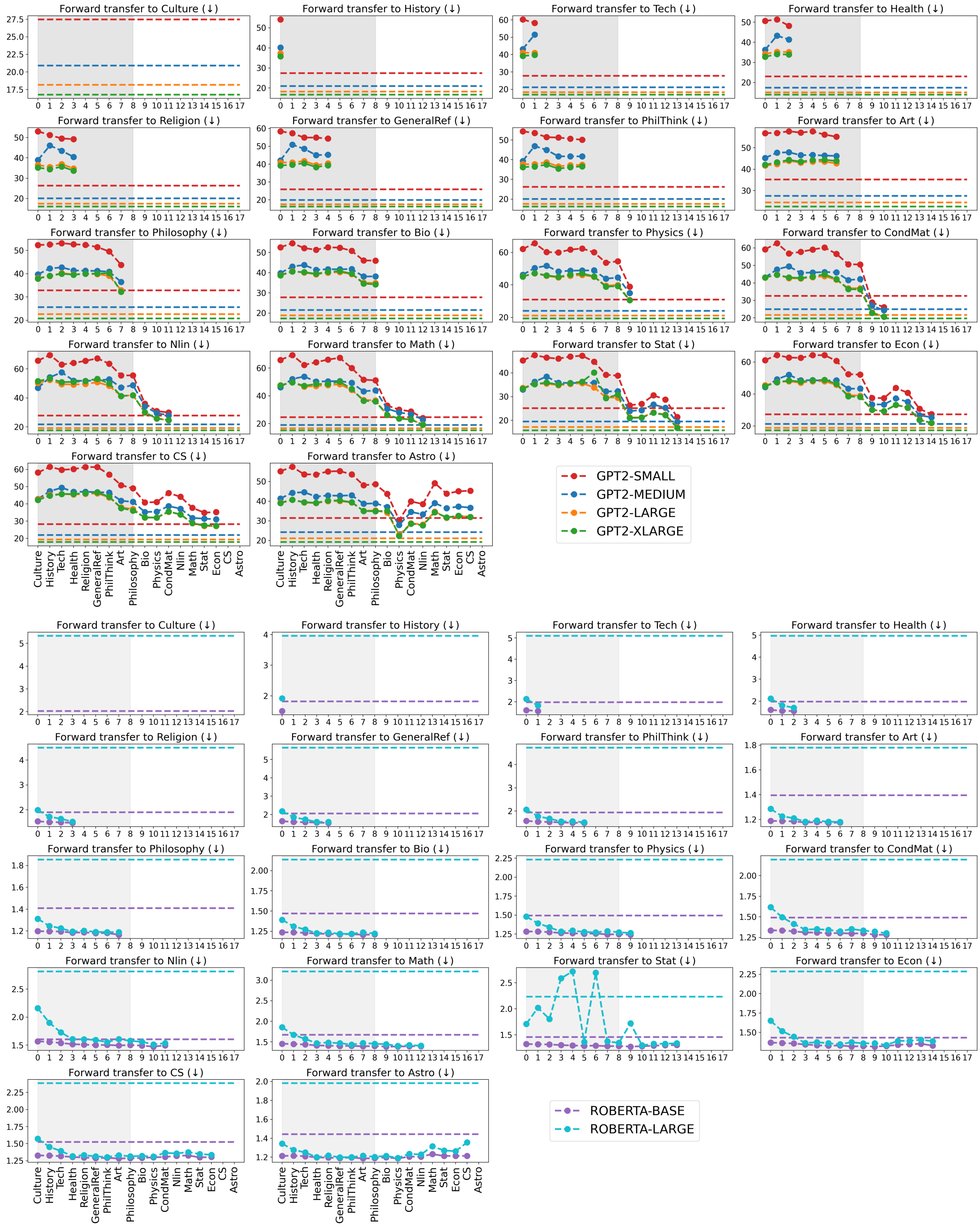

Figure 13: Forward transfer results with similar training order. The checkpoints are saved after having trained on an L1 domain (hence 18 checkpoints per model). The i'th panel shows the forward performance on i'th domain, obtained by evaluating all previous $i-1$ checkpoints on that domain. The dashed lines show zero-shot performance. x and y axes correspond to L1 domain names and perplexities, respectively.

Figure 14: Backward transfer illustration with RoBERTa-large trained in similar and random order (left and right columns). Each panel shows the backward transfer perplexity (pink) computed on a particular domain. For baseline comparisons, we also plot zero-shot (yellow) and continual pretraining (black) perplexities.