Continual Learning of Large Language Models: A Comprehensive Survey

HAIZHOU SHI*, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, United States
ZIHAO XU, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, United States
HENGYI WANG, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, United States
WEIYI QIN, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, United States
WENYUAN WANG†, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, United States
YIBIN WANG†, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, United States
ZIFENG WANG, Cloud AI Research, Google Inc, Mountain View, United States
SAYNA EBRAHIMI, Google DeepMind, Mountain View, United States
HAO WANG*, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, United States

The challenge of effectively and efficiently adapting statically pre-trained Large Language Models (LLMs) to ever-evolving data distributions remains predominant. When tailored for specific needs, pre-trained LLMs often suffer from significant performance degradation in previous knowledge domains – a phenomenon known as *"catastrophic forgetting"*. While extensively studied in the Continual Learning (CL) community, this problem presents new challenges in the context of LLMs. In this survey, we provide a comprehensive overview and detailed discussion of the current research progress on LLMs within the context of CL. Besides the introduction of the preliminary knowledge, this survey is structured into four main

*Correspondence to: Haizhou Shi <haizhou.shi@rutgers.edu> and Hao Wang <hw488@cs.rutgers.edu>.
†Work done as visiting students at Rutgers Machine Learning Lab.

Authors' Contact Information: Haizhou Shi, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, New Jersey, United States; e-mail: haizhou.shi.057@gmail.com; Zihao Xu, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, New Jersey, United States; e-mail: zihao.xu@rutgers.edu; Hengyi Wang, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, New Jersey, United States; e-mail: hengyi.wang@rutgers.edu; Weiyi Qin, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, New Jersey, United States; e-mail: weiyi.qin@rutgers.edu; Wenyuan Wang, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, New Jersey, United States; e-mail: ww462@scarletmail.rutgers.edu; Yibin Wang, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, New Jersey, United States; e-mail: yibin.wang@rutgers.edu; Zifeng Wang, Cloud AI Research, Google Inc, Mountain View, California, United States; e-mail: zifengw@google.com; Sayna Ebrahimi, Google DeepMind, Mountain View, California, United States; e-mail: saynae@google.com; Hao Wang, Department of Computer Science, Rutgers The State University of New Jersey, New Brunswick, New Jersey, United States; e-mail: hw488@cs.rutgers.edu.

ACM 1557-7341/2025/5-ART
https://doi.org/10.1145/3735633

sections: we first describe an overview of continually learning LLMs, consisting of two directions of continuity: *vertical continuity (or vertical continual learning)*, i.e., continual adaptation from general to specific capabilities, and *horizontal continuity (or horizontal continual learning)*, i.e., continual adaptation across time and domains (Section 3). Following vertical continuity, we summarize three stages of learning LLMs in the context of modern CL: Continual Pre-Training (CPT), Domain-Adaptive Pre-training (DAP), and Continual Fine-Tuning (CFT) (Section 4). We then provide an overview of evaluation protocols for continual learning with LLMs, along with currently available data sources (Section 5). Finally, we discuss intriguing questions related to continual learning for LLMs (Section 6). This survey sheds light on the relatively understudied domain of continually pre-training, adapting, and fine-tuning large language models, suggesting the necessity for greater attention from the community. Key areas requiring immediate focus include the development of practical and accessible evaluation benchmarks, along with methodologies specifically designed to counter forgetting and enable knowledge transfer within the evolving landscape of LLM learning paradigms. The full list of papers examined in this survey is available at https://github.com/Wang-ML-Lab/llm-continual-learning-survey.

CCS Concepts: • **Computing methodologies → Lifelong machine learning**; **Natural language processing**; **Neural networks**.

Additional Key Words and Phrases: Large Language Models, Continual Learning.

1 Introduction

Recent advances in large language models (LLMs) have demonstrated considerable potential for achieving artificial general intelligence (AGI) [1, 6, 22, 40, 173, 186, 230, 231]. Researchers have observed that complex abilities such as multi-step reasoning, few-shot in-context learning, and instruction following improve as the scale of parameter size increases [159, 250, 252, 253, 277]. The development of LLMs is impactful and revolutionary, prompting machine learning practitioners to reconsider traditional computational paradigms for once-challenging human-level tasks. However, LLMs are typically trained on static, pre-collected datasets encompassing general domains, leading to gradual performance degradation over time [5, 52, 95, 96, 100, 137] and across different content domains [35, 44, 69, 71, 100, 104, 183, 184, 221]. Additionally, a single pre-trained large model cannot meet every user need and requires further fine-tuning [10, 16, 37, 106, 182, 254, 255, 255, 281, 299, 299]. While one potential solution is re-collecting pre-training data and re-training models with additional specific needs, this approach is prohibitively expensive and impractical in real-world scenarios.

To efficiently adapt LLMs to downstream tasks while minimizing performance degradation on previous knowledge domains, researchers employ the methodology of Continual Learning (CL), also known as *lifelong learning* or *incremental learning* [38, 178, 232, 237]. Inspired by the incremental learning pattern observed in human brains [101, 153, 154, 175], CL trains machine learning models sequentially on a series of tasks with the expectation of maintaining performance across all tasks [57, 58, 113, 124]. Throughout training, models have limited or no access to previous data, posing a challenge in retaining past knowledge as optimization constraints from unseen previous data are absent during current-task learning [124, 135, 213]. This challenge, known as *catastrophic forgetting* [155], has been a central focus in continual learning research since its inception. Over the years, researchers have explored various techniques to mitigate forgetting. These include replay-based methods [30, 207, 213], parameter regularization [4, 113, 196], and model architecture expansion [191, 236]. Together, these techniques have significantly advanced the goal of achieving zero forgetting in continual learning across diverse tasks, model architectures, and learning paradigms.

In the context of training and adapting LLMs sequentially, the significance of CL is undergoing semantic shifts of its own as well. To highlight this ongoing shift, in this paper, we provide a comprehensive overview and detailed discussion of the current research progress on continual LLMs. For the general picture of continual LLMs, we for the first time divide it into two directions of continuity that need to be addressed by practitioners (details in Section 3):

- **Vertical continuity (or vertical continual learning)**, which refers to the ongoing adaptation of LLMs as they transition from large-scale general domains to smaller-scale specific domains, involving shifts in learning objectives and entities of execution. For example, healthcare institutions may develop LLMs tailored to the medical domain while retaining their general reasoning and question answering capabilities for users.
- **Horizontal continuity (or horizontal continual learning)**, which refers to continual adaptation across time and domains, often entails multiple training stages and increased vulnerability to forgetting. For example, social media platforms continuously update LLMs to reflect recent trends, ensuring accurate targeting of downstream services like advertising and recommendations without compromised experience for existing users.

Importantly, separating vertical and horizontal CL transcends mere modification of existing paradigms, like domain-incremental learning, which aligns with horizontal continuity. This distinction offers a robust framework for analyzing complex CL paradigms in language models. For instance, Recyclable Tuning preserves both vertical and horizontal continuity simultaneously [183], and future designs might include zigzagging between horizontal and vertical CL.

In Fig. 1, following *vertical continuity*, we delineate three key stages of LLM learning within modern CL: Continual Pre-Training (CPT), Domain-Adaptive Pre-training (DAP), and Continual Fine-Tuning (CFT) (details in Section 4). In CPT, existing research primarily investigates three types of distributional shifts: temporal, content-level, and language-level. Each presents distinct focuses and challenges. In DAP, CL evaluation and techniques are frequently utilized. However, there is a noticeable lack of diversity in these techniques, considering the maturity of the conventional CL community. In CFT, our focus is on the emerging field of learning LLMs, covering topics such as Continual Instruction Tuning (CIT), Continual Model Refinement (CMR), Continual Model Alignment (CMA), and Continual Multimodal LLMs (CMLLMs). Next, we present a compilation of publicly available evaluation protocols and benchmarks (details in Section 5). We conclude our survey with a discussion covering emergent properties of continual LLMs, changes in the roles of conventional CL types and memory constraints within the context of continual LLMs, and prospective research directions for this subject (details in Section 6).

In summary, this survey provides a comprehensive review of existing continual learning studies for LLMs, which significantly distinguishes itself from existing literature on related topics [17, 105, 237, 261, 276]. Our survey highlights the underexplored research area of continually developing LLMs, especially in the field of CPT and DAP. We emphasize the needs for increased attention from the community, including the development of practical, accessible, and widely acknowledged evaluation benchmarks. Additionally, methodologies need to be tailored to address forgetting in emerging LLM learning paradigms. We hope this survey can provide a systematic and novel perspective of continual learning in the rapidly-changing field of LLMs and can help the continual learning community contribute to the challenging goals of developing LLMs in a more efficient, reliable, and sustainable manner [8, 25, 95, 219, 268].

2 Background and Related Work

2.1 Large Language Models

Primarily built on the transformer architecture, pre-trained language models (PLMs) have established a universal hidden embedding space through extensive pre-training on large-scale unlabeled text corpora [51, 133, 189]. By scaling parameters to billions or even hundreds of billions and training on massive text datasets [84, 102], PLMs not only demonstrate superior language understanding and generation capabilities but also manifest emergent abilities such as in-context learning, instruction following, and multi-step reasoning [159, 250, 252, 253, 277].

These larger models are commonly referred to as Large Language Models (LLMs). For more detailed introduction, please refer to Appendix A.1.

2.1.1 Pre-training of LLMs. There are two popular pre-training paradigms for LLMs. (1) *Decoder-only models* typically employ auto-regressive language modeling (LM) tasks during pre-training, including the GPT family [1, 22, 173, 186], Gemini family [194, 225], and the open-source Llama family [230, 231]. Specifically, given a sequence of tokens $\boldsymbol{x} = [x_1, x_2, \cdots, x_N]$, LM predicts the next token x_t autoregressively based on all preceding tokens $\boldsymbol{x}_{<t} = [x_1, x_2, \cdots, x_{t-1}]$, and trains the entire network by minimizing the negative log-likelihood $-\sum_{t=1}^{N} \log P(x_t|\boldsymbol{x}_{<t})$, where $P(x_1|\boldsymbol{x}_{<1}) \triangleq P(x_1)$ is the unconditional probability estimation of the first token. (2) *Encoder-only models*, e.g., BERT [51, 133], use masked language modeling (MLM) as a common pre-training objective. In MLM, for the input sequence $\boldsymbol{x}$, a subset of input tokens $m(\boldsymbol{x})$ are masked and replaced with the special [MASK] token. The pre-training goal is to utilize the unmasked parts $\boldsymbol{x}_{\backslash m(\boldsymbol{x})}$ to predict the masked portions $m(\boldsymbol{x})$. In summary, the overarching goal of MLM is to minimize the negative log-likelihood $-\sum_{\widehat{x} \in m(\boldsymbol{x})} \log P(\widehat{x}|\boldsymbol{x}_{\backslash m(\boldsymbol{x})})$.

2.1.2 Adaptation of LLMs. LLMs are primarily trained to generate linguistically coherent text. However, this training may not align with human values, preferences, or practical needs. Furthermore, the pre-training data can be outdated, leading to knowledge cutoffs or inaccuracies. To address these issues, various computational paradigms such as Instruction Tuning (IT) [288], Model Refinement (MR) [47], and Model Alignment (MA) [174, 187] have been proposed. These approaches adapt LLMs to better meet diverse downstream tasks and user requirements.

Numerous studies show that **Instruction Tuning (IT)** can notably improve LLMs' ability to follow textual instructions [98, 174, 203, 250, 288], leveraging the pre-existing knowledge within LLMs to bridge the gap between general and task-specific performance [251]. Recent works like WizardLM [269] and CodecLM [246] further tailor synthetic data to steer LLMs' behavior through IT. Additionally, IT enhances the interaction between humans and LLMs, providing a more natural interface and aligning LLM outputs more closely with human expectations and preferences [145]. LLMs make mistakes, such as inaccurate translations or outdated information [47]. Directly fine-tuning the model to correct these mistakes may disrupt its performance on previously learned tasks. To overcome these challenges, **Model Refinement (MR)** is proposed to rectify the model's errors while preserving its performance on other inputs, with only moderate computing resources [47, 74, 76, 92, 163, 164, 215]. **Model Alignment (MA)** ensures AI systems' actions and outputs align with human values, ethics, and preferences [174, 187]. MA can be broadly categorized into two types: Reinforcement Learning-based (RL-based) and Supervised Learning-based (SL-based). RL-based approaches [174, 205] are trained to make decisions reinforced by human feedback, using a reward system to guide them towards desirable outcomes. In contrast, SL-based approaches [81, 97, 187] directly train models on datasets of human preferences, aligning their output with demonstrated human values.

2.2 Continual Learning

Humans can accumulate knowledge and skills across tasks without significant performance decline on previous tasks [101, 153, 154, 175]. In contrast, machine learning models, which are typically data-centric, often experience performance degradation on old tasks when trained on new ones, a phenomenon known as *"catastrophic forgetting."* The challenge of adapting models to a sequence of tasks without forgetting, especially when little to no past data can be preserved, is extensively studied in the continual learning community [38, 178, 232, 237]. Formally, the objective of CL is to find a hypothesis that minimizes risk across all tasks/domains. Consider DIL as an

example [112, 213], at t-th learning stage, the ideal training objective $\mathcal{L}(h)$ is defined as

$$\mathcal{L}(h) \triangleq \underbrace{\sum_{i=1}^{t-1} \mathcal{L}_{\mathcal{D}_i}(h)}_{\text{past domains}} + \underbrace{\mathcal{L}_{\mathcal{D}_t}(h)}_{\text{current domain}} , \tag{1}$$

where $\mathcal{D}_i$ denotes the data distribution of the i-th continual learning stage. The objectives for past domains are often challenging to measure or optimize due to the memory constraints (Definition A.3). Therefore, the core of designing CL algorithms lies in identifying a proxy learning objective for the first term without violating the memory constraint. **A more detailed introduction to the formal definition of CL and its techniques can be found in Appendix A.2.**

2.2.1 Types of Continual Learning. To lay the groundwork for subsequent discussions (as illustrated in Table 3 and Section 6.2), we follow the conceptual framework proposed by [112, 232, 237]. There are three primary types of continual learning scenarios: (i) Task-Incremental Learning (TIL), where task indices are available to the model during inference [113, 124]; (ii) Domain-Incremental Learning (DIL), where the model learns a sequence of tasks with the same formulation but without task indices during inference [213]; and (iii) Class-Incremental Learning (CIL), where the model learns new classes of data during training [112, 193].

2.2.2 Techniques of Continual Learning. Existing CL techniques can be roughly categorized into five groups [237]: (i) replay-based, (ii) regularization-based, (iii) architecture-based, (iv) optimization-based, and (v) representation-based. Here, we provide a concise yet comprehensive introduction to the first three categories of continual learning techniques, as they are extensively applied in continual LLMs.

Replay-based methods adopt the relaxed memory constraint by keeping a small buffer of observed data and retraining the model on it when learning new tasks. Although replay-based methods may theoretically lead to loose generalization bounds [213], they are valued for their simplicity, stability, and high performance, even with a small episodic memory [24, 30, 193, 195]. **Regularization-based methods** adopt a regularization term $\lambda \left\| \boldsymbol{\theta} - \boldsymbol{\theta}_{t-1} \right\|_{\Sigma}$ that penalizes large deviation from the history model in the parameter space, where $\|\boldsymbol{v}\|_{\Sigma} = \boldsymbol{v}^{\top}\Sigma\boldsymbol{v}$ is the vector norm evaluated on a positive-semi-definite matrix Σ, and λ is the regularization coefficient, a hyper-parameter introduced to balance the past knowledge retention and current knowledge learning. The matrix Σ introduced is to measure the different level of importance of each parameters and their correlations in retaining the past knowledge. In practice, to reduce computational overhead, diagonal matrices are often designed to encode only the importance of each parameter [4, 113, 197]. **Architecture-based methods**, especially expanding the network architecture dynamically to assimilate new knowledge, is considered the most efficient form of CL [248, 249]. This method primarily tackles adaptation challenges and can achieve zero-forgetting when task IDs are available during inference or can be correctly inferred [71, 256]. However, due to the difficulty of task ID inference, architecture expansion is predominantly utilized in TIL but is scarcely explored in DIL or CIL. In conjunction with pre-trained backbone large models like ViT [54], CoLoR [256] trains various low-rank adaptation (LoRA) [86] modules for different tasks. It estimates and stores prototypes for each task and utilizes the natural clustering ability of the pre-trained model during testing to infer task IDs, selecting the corresponding LoRA component for prediction generation. In the domain of continual LLMs, architecture expansion has resurged in popularity following the rise of parameter-efficient fine-tuning (PEFT) [50, 86, 211], a topic we will delve into shortly [96, 100, 118, 177, 240, 257, 272, 273].

2.2.3 Evaluation Metrics of Continual Learning. There are four evaluation protocols primarily designed for continual learning. **Overall Performance (OP)** [106, 286, 291] calculates the average performance up until the current training stage, measuring the overall ability of a model balancing the performance of each task. As noted in [213], OP corresponds to the primary optimization objective of continual learning, and hence receives the

most attention. **Forgetting (F)** represents the largest performance drop observed of each task throughout the training process, averaged over all training stages. It quantifies the negative impact of learning new tasks brought to previously acquired knowledge. Ideally, a robust continual learning framework should achieve **Backward Transfer (BWT)**, where learning new tasks enhances performance on prior tasks. BWT is measured by negating the forgetting, and hence a negative forgetting indicates a an improvement in performance on earlier tasks. **Forward Transfer (FWT)** measures the generalization ability of the continual learning algorithms to unseen tasks. It is defined as the difference between the current model's performance evaluated on the future tasks and the randomly initialized model. Refer to Appendix B.1 for more details.

3 Continual Learning Meets Large Language Models: An Overview

Large language models (LLMs) are extensive in various dimensions, including the size of model parameters, pre-training datasets, computational resources, project teams, and development cycles [1, 6, 22, 40, 173, 186, 230, 231]. The substantial scale of LLMs presents notable challenges for development teams, particularly in keeping them updated amidst rapid environmental changes [5, 52, 95, 96, 100]. To illustrate, in 2023, the average daily influx of new tweets exceeds 500 million[1], and training on even a subset of this large volume of data is unaffordable. Recyclable Tuning [183] is the first work to explicitly outline the supplier-consumer structure in the modern LLM production pipeline. On the supplier side, the model is continually pre-trained over a sequence of large-scale unlabeled datasets. After every release of the pre-trained model, the consumer utilizes the stronger and more up-to-date upstream model for downstream tasks. Compared to the upstream supplier, downstream users often lack capacity of collecting and storing large-scale data, maintaining large-scale hardware systems, and training LLMs themselves. In this survey, we extend this framework and further present a comprehensive modern production pipeline encompassing various studies on continual LLM pre-training, adaptation, and deployment (Fig. 1). What sets our framework apart from existing studies [261] is the *incorporation of two directions of continuity:* ***Vertical Continuity*** *and* ***Horizontal Continuity***.

3.1 Vertical Continuity (Vertical Continual Learning)

Definition. Vertical continuity (or vertical continual learning) has long been studied, either implicitly or explicitly, in existing literature. Vertical continuity is characterized by a hierarchical structure encompassing data inclusiveness, task scope, and computational resources. Specifically, the training task transitions gradually from general pre-training to downstream tasks, typically undertaken by distinct entities within the production pipeline [68, 71, 183, 197, 268, 272]. Fig. 1 shows a typical pipeline for vertical continuity in LLMs, i.e., "pre-training" → "domain-adaptive training" → "downstream fine-tuning" [42, 48, 68, 72, 73, 91, 121, 146, 148, 197, 257, 258, 272, 303]:

- **Pre-training.** During the *pre-training* stage, a substantial amount of data from diverse domains is required to develop a general-purpose LLM. This phase demands a sizable research and development team dedicated to training and benchmarking the model, along with considerable computational resources.
- **Domain-Adaptive Pre-training.** Subsequently, downstream institutions may opt for *domain-adaptive pre-training* to tailor the model for specific tasks using domain-specific data unavailable to the upstream supplier.
- **Finetuning.** Finally, the LLM undergoes *fine-tuning* on annotated data for downstream tasks before deployment.

Throughout the process, the unlabeled domain-specific dataset is smaller in scale than the upstream pre-training phase but larger than the final downstream task fine-tuning phase. This pattern extends to computational resources, team size, and other factors. It is important to note that vertical continuity can involve more than three

[1]Source: https://www.omnicoreagency.com/twitter-statistics

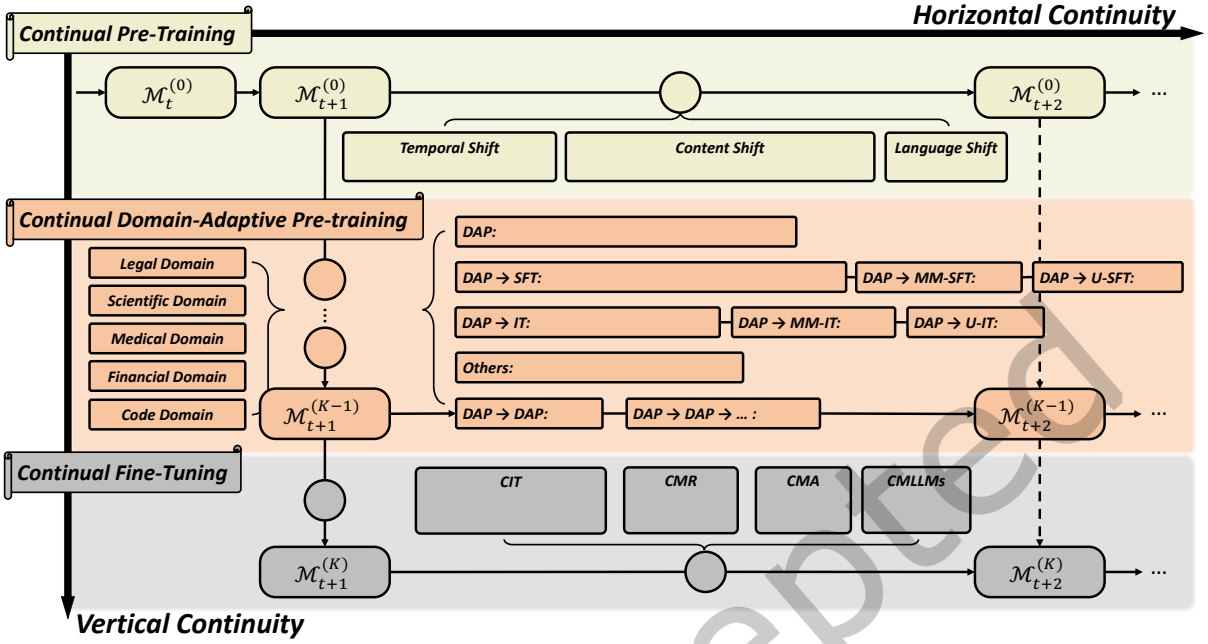

Fig. 1. A high-level overview of the modern pipeline for continually pre-training and fine-tuning LLMs, where two dimensions of continuity are described. **Vertical Continuity (or Vertical Continual Learning):** LLM training can be vertically divided into three stages: (i) Continual Pre-Training (CPT), (ii) Domain-Adaptive Pre-training (DAP), and (iii) Continual Fine-Tuning (CFT). The main focus is the retention of the LLM's general knowledge (prevention of vertical forgetting). **Horizontal Continuity (or Horizontal Continual Learning):** After the LLMs are deployed, the models are continually updated when a new set of data becomes available. The primary goal is to prevent horizontal forgetting in a long sequence of tasks.

stages [91, 129, 172, 199]. In real-world applications, during domain-adaptive pre-training, additional "layers" can be added to accommodate multiple entities, such as various departments with distinct objectives but operating within the same domain.

Vertical Forgetting. We term the performance degradation (in terms of general knowledge) due to vertical continual learning *"vertical forgetting"*. As shown in Fig. 2, for vertical continual learning, the data distribution of upstream tasks partially covers the downstream, meaning the model might start off at a decent initialization for the subsequent stage of training. Two significant challenges must be addressed to prevent vertical forgetting:

- **Task Heterogeneity.** Stemming from the inherent disparity between the formulation of upstream tasks and downstream tasks, *task heterogeneity* can lead to differences in model structures and training schemes, which has long been recognized as a major hurdle [112, 124, 170, 193, 262]. To mitigate this issue, practitioners often employ methodologies such as freezing shared parameters during downstream phases or reformulating downstream tasks to match the structure of pre-training tasks [118, 177, 240, 257, 272, 273].
- **Inaccessible Upstream Data.** This challenge arises primarily from varying levels of confidentiality across entities undertaking vertical continual learning. Data collected and curated under different protocols may not be accessible to some downstream entities. This scenario is even more challenging than the strict memory constraint presented in conventional CL (Definition A.3), as algorithms for latter case

rely on access to previous data at specific points for parameter importance measurement [4, 113] or for replay [24, 30, 195, 213]. To address the challenge of *inaccessible upstream data*, existing methods either use public datasets or generate pseudo-examples to create proxy pre-training datasets [182].

3.2 Horizontal Continuity (Horizontal Continual Learning)

Definition. Horizontal continuity (or horizontal continual learning) refers to continual adaptation across time and domains, a topic extensively explored within the continual learning community. The primary rationale for preserving horizontal continuity lies in the dynamic nature of data distribution over time. To stay updated with these content shifts, an LLM must incrementally learn newly-emerged data. Otherwise, the cost of re-training will become prohibitively expensive and impractical [5, 29, 219, 268]. Empirical evidence has consistently shown that despite their impressive capabilities, LLMs struggle to generalize effectively to future unseen data, particularly in the face of temporal or domain shifts [5, 52, 95, 96]. Additionally, they struggle to retain complete knowledge of past experiences when adapting to new temporal domains, although they do demonstrate a higher level of robustness against catastrophic forgetting [144, 156, 223, 299]. The necessity of employing complex CL algorithms to address challenges in LLMs remains an open question. For instance, during large-scale continual pre-training, large institutions can typically afford the storage costs of retaining all historical data, rendering memory constraints meaningless. Several studies have demonstrated that with full access to historical data, simple sparse replay techniques can effectively mitigate forgetting [62, 181, 208, 223]. In contrast, numerous continual learning studies have showcased superior performance compared to naive solutions, suggesting the importance of continual learning techniques in LLM training [35, 95, 100, 184].

Horizontal Forgetting. We informally define *"horizontal forgetting"* as the performance degradation on the previous tasks when model is undergoing horizontal continual learning. As illustrated in Fig. 2, horizontal continual learning typically involves training stages of similar scales, with potential distributional overlap among their data. In summary, two main challenges need to be addressed for horizontal continual learning of LLMs:

- **Long Task Sequences.** Horizontal continual learning ideally involves numerous incremental phases, particularly to accommodate temporal shifts in data distribution. A *longer task sequence* entails more update steps of the model, leading to inevitable forgetting of previously learned tasks. To address this challenge, researchers employ established continual learning techniques with stronger constraints, such as continual model ensemble [191].
- **Abrupt Distributional Shift.** In contrast to vertical continuity, where distributional shifts are often predictable, horizontal continual learning does not impose constraints on task properties. Evidence suggests that abrupt changes in task distributions can result in significant horizontal forgetting of the model [204].

4 Learning Stages of Continual Large Language Models

Fig. 1 provides an overview of continually learning LLMs. Along the axis of vertical continuity, three main "layers" of modern continual learning emerge. The top layer, Continual Pre-Training (CPT), involves continuous pre-training of LLMs by the supplier on newly-collected data alongside existing data (Section 4.1). The middle layer, Domain-Adaptive Pre-training (DAP), prepares LLMs for domain-specific applications through additional pre-training on domain-specific *unlabeled* data (Section 4.2). The bottom layer, Continual Fine-Tuning (CFT), targets models for final downstream tasks on the consumer side (Section 4.3), where the model needs to be updated after deployment for the specified task.

4.1 Continual Pre-Training (CPT)

4.1.1 CPT: Effectiveness and Efficiency. Before delving into the details of continual pre-training (CPT), it is important to address two fundamental questions: Firstly, regarding *effectiveness*, can CPT enhance performance

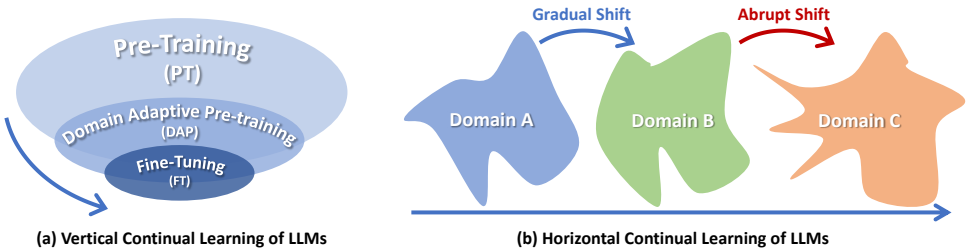

Fig. 2. A diagram showing two different directions of continual learning of LLMs. **(a) Vertical Continual Learning of LLMs:** in this case, the upstream data distribution usually partially covers the subsequent tasks' data distribution. **(b) Horizontal Continual Learning of LLMs:** No constraints on the data distributions are present on horizontal continual learning. The continual LLMs need to handle the challenge of abrupt distributional shifts and a longer sequence of training.

on downstream tasks beyond that of the initial training on a wide range of data domains? Extensive studies have not only demonstrated the necessity of CPT for improved downstream performance [35, 71, 95, 96, 100, 184], but also shown that when distributional shifts are gradual [95, 278] or somewhat correlated [71], CPT can effectively help model generalize to unseen data. The second question is about *efficiency*: given the large size of an LLM' parameters and data, both old and new, can we achieve adaptation and knowledge retention in a computationally efficient way? Concerning efficiency, most studies focus on techniques for efficient knowledge retention [95, 96, 100, 118], which significantly overlap with the CL literature addressing catastrophic forgetting [4, 24, 191, 193, 195, 196, 201, 207, 213, 236]. In contrast to prior approaches that fully utilize emergent data, some studies recognize the impracticality of this approach in real production environments. Instead, they concentrate on further improving the efficiency of adaptation. For instance, ELLE [184] employs a function-preserved model expansion to facilitate efficient knowledge growth; [5] and [268] sub-sample training data based on novelty and diversity to enhance training efficiency, achieving superior performance compared to full-data training. Though currently underexplored, efficient adaptation in continual pre-training is poised to become significant, given recent findings emphasizing data quality over quantity for LLM generalization [216, 267].

4.1.2 General Observations on CPT. Table 1 summarizes the existing studies on continual pre-training (CPT), and here are some key observations we make about CPT.

- **OBS-1: The development of advanced techniques tailored specifically for CPT is at the starting stage and warrants further exploration.** Only about half of the examined papers propose novel techniques for CPT [5, 35, 44, 52, 71, 104, 183, 184, 221], while the remaining half either focus solely on the effects of pure adaptation without considering CL techniques [63, 69, 137], or conduct empirical studies on the straightforward application of existing CL techniques [95, 96, 100, 118].
- **OBS-2: The diversity of CL techniques incorporated in CPT remains limited.** Most practical implementations of CL techniques for CPT primarily focus on architecture expansion of LLMs [5, 35, 44, 52, 71, 183], with only a few explicitly utilizing replay [35, 183] and parameter regularization [5, 35].
- **OBS-3: There is an apparent gap between the existing studies and the real production environment of CPT.** Except for the recent study [278] which conducts CPT over 159 domains, the longest sequence of pre-training stages explored is 8 [71, 100]. However, this falls short of real-world scenarios where continual pre-training occurs more frequently and persists for months or years. The efficacy of CPT methods in such prolonged scenarios remains uncertain. Additionally, investigating CPT in a task-boundary-free data stream setting is an important avenue for research to be explored in the future as well.

Table 1. **Summary of existing studies on Continual Pre-training of LLMs.** The papers are organized based on their relation to CL: (i) no CL techniques are studied, (ii) CL techniques are studied as solely baselines, and (iii) new CL approaches are proposed. In the table, *Dist. Shift* denotes what type(s) of distributional shifts this particular study considers and is dedicated to solve. In the section of **Continual Learning Tech.**, we mainly categorize three types of continual learning techniques that are studied in the paper: rehearsal (*Rehearsal*), parameter regularization (*Param. Reg.*), and architecture expansion (*Arch. Exp.*). We use "✓", "✗", and "♣" to denote "deployed in the proposed method", "not studied in the paper", and "studied as a baseline method", respectively. Note that we do not include naive sequential fine-tuning in this table, as it is universally studied as the important baseline method in all of the papers in the table. The papers with only "♣" [95, 96, 100] means that only existing CL techniques are studied, without proposing new ones, and the papers with only "✗" [63, 69] means that special aspects of fine-tuning are studied, without using CL techniques.

Method	Scenario		Continual Learning Tech.			LLM Arch.	Evaluation	
	Dist. Shift	*#Domains*	*Rehearsal*	*Param. Reg.*	*Arch. Exp.*		*Pre-Training*	*Downstream*
TimeLMs [137]	Temporal	8	✗	✗	✗	RoBERTa	✓	✓
[278]	Content	159	✗	✗	✗	RoBERTa GPT-2	✓	✗
[69]	Content	1	✗	✗	✗	Pythia	✓	✗
[63]	Language	3	✗	✗	✗	GPT	✓	✗
RHO-1 [130]	Other	1	✗	✗	✗	TinyLlama Mistral	✓	✓
[118]	Language	1	✗	P-Freeze♣	Adapter♣ LoRA♣	Llama2	✓	✓
CKL [96]	Temporal	1	Mix-Review♣	P-Freeze♣ RecAdam♣	LoRA♣ K-Adapter♣	T5	✗	✓
LLPT [100]	Temporal	4	ER♣ Logit-KD♣ Rep-KD♣ Contrast-KD♣ SEED-KD♣	oEWC♣	Adapter♣ Layer Exp.♣	RoBERTa	✓	✓
	Content	8						
TemporalWiki [95]	Temporal	5	Mix-Review♣	P-Freeze♣ RecAdam♣	LoRA♣ K-Adapter♣	GPT-2	✓	✓
CPT* [104]	Content	4	DER++♣ KD♣	CPT✓ EWC♣ HAT♣	Adapter♣ DEMix♣	RoBERTa	✓	✗
ERNIE 2.0 [221]	Content	4	ER✓♣	✗	✗	ERNIE	✗	✓
[5]	Temporal	7	✗	P-Freeze✓	Vocab. Exp.✓	BERT	✗	✓
[44]	Content	5	✗	✗	Vocab. Exp.✓	BERT RoBERTa	✗	✓
DEMix [71]	Content	8	✗	✗	MoE✓	GPT-3	✓	✓
TempoT5 [52]	Temporal	1	✗	✗	Vocab. Exp.✓ Prompt✓	T5	✗	✓
RecTuning [183]	Content	4	ER✓ KD✓	✗	Adapter✓	RoBERTa	✗	✓
Lifelong-MoE [35]	Content	3	ER♣ KD✓	P-Freeze✓ L2♣	MoE✓	GLaM	✓	✓
ELLE [184]	Content	5	ER✓♣ KD♣	P-Freeze✓	Prompt✓ Layer Exp.✓ Adapter♣	BERT GPT	✓	✓
[94]	Content Language	2	ER✓	✗	✗	GPT-NeoX	✓	✓
CEM [294]	Other	1	ER✓	✗	✗	CuteGPT ChatGLM Qwen-Chat	✗	✓
IR-DRO [36]	Other	1	ER✓	✗	✗	OPT	✗	✓

4.1.3 Distributional Shifts in CPT. This survey categorizes distributional shifts of CPT into three main types: (i) *Language Shift*: LLMs sequentially learn different language corpora, e.g., English → Chinese [63, 118]. (ii) *Content Shift*: LLMs sequentially learn corpora from different fields, e.g., chemistry → biology [35, 44, 69, 71, 100, 183]. (iii) *Temporal Shift*: Distributional shifts occur over time, e.g., news in 2021 → news in 2022, with a major focus on timestamp-sensitive knowledge retention and update [5, 52, 95, 96, 100].

Language Shift. [63] focuses on assessing LLMs' natural ability to learn new languages sequentially. With no explicit CL techniques employed, the study observes consistent positive forward transfer of the knowledge, facilitating new language acquisition regardless of the learning order. Forgetting, on the other hand, emerges as a significant challenge that cannot be mitigated by the increasing size of LLMs. In [118], the degree of forgetting of previously learned language when adapting LLMs to a new language is investigated. Various CL techniques, including parameter freezing, LoRA [86], and $(IA)^3$ [132], are evaluated across multiple dimensions. Preliminary experimental results highlight the non-trivial nature of addressing horizontal forgetting for CPT under the language shift as well.

Content Shift. [278] explores the large-scale CPT over 159 content domains, and shows that CPT on various domains can effectively improve models' adaptation ability compared to DAP on single domain. Similarly, [69] continues the pre-training phase of Pythia [16] with no complex CL techniques and discovers that learning rate re-warming consistently improves models trained from scratch. Built upon this simple observation, [94] further shows that proper combination of learning rate re-warming and re-decay, and replay of the previous data is sufficient to achieve a comparable performance to full re-training. LLPT [100] establishes a comprehensive training and evaluation protocol for a series of content-level distributional shifts. They assess multiple CL methods and, similar to [63], find consistent forward knowledge transfer, yet horizontal forgetting remains significant. Besides, contrary to the common understanding that experience replay [30] is the most efficient approach to preventing forgetting, the authors find it ineffective in the case of CPT, due to the potential overfitting issue. Recyclable Tuning [183] shows that if the upstream supplier continually pre-trains LLMs, with or without replay, consumer-side efficiency can be boosted by recycling previously learned update components when proper CL techniques are applied.

DEMix [71] incrementally trains and integrates new experts (DEMix layer) for new domains during CPT. To ensure reasonable inference performance during testing when no domain information is available, it proposes a parameter-free probabilistic approach to dynamically estimate a weighted mixture of domains. DEMix's modularization has been shown to facilitate efficient domain-adaptive pre-training, promote relevant knowledge during inference, and allow for removable components. Lifelong-MoE [35], similar to DEMix [71], incrementally trains domain experts for new domains. However, Lifelong-MoE differs from DEMix in utilizing a *token-level gating function* to activate multiple experts for intermediate embedding calculation. During training, previous experts' parameters and gating functions remain frozen, and knowledge distillation loss is employed to regulate parameter updates, which thereby makes Lifelong-MoE robust against the issue of horizontal forgetting.

It is noteworthy that some papers draw almost opposite conclusions regarding the significance of CPT for content shifts. For instance, [44] continually pre-trains BERT-based models [51, 133] on five scientific domains and evaluates performance on downstream sentiment analysis. They observe that even the trivial sequential pre-training does not exhibit severe forgetting, prompting reasonable questions about the necessity of CPT.

Temporal Shift. In the context of CPT amid content shifts, Multi-Task Learning (MTL) is often regarded as the upper bound achievable [178, 213, 237]. However, this belief does not fully hold when considering CL under temporal shifts [52, 95, 96], as temporal shifts can introduce conflicting information, posing challenges for LLMs. For instance, the statement *"Lionel Messi plays for team Barcelona"* remains accurate from 2004 to 2021 but becomes false by 2024, as *"Lionel Messi plays for team Inter Miami"* becomes the correct statement.

Hence, as advocated by CKL [96] and TemporalWiki [95], LLMs undergoing continual adaptation to temporal shifts must simultaneously achieve three objectives: (i) retention of old knowledge, (ii) acquisition of new knowledge, and (iii) update of the outdated knowledge. They evaluate the same set of continual learning baseline methods [34, 79, 87, 239], each highlighting distinct aspects of their impact. CKL [96] observes that parameter expansion consistently exhibits robust performance across all experimental conditions. In contrast, replay-based methods struggle to efficiently adapt to new knowledge acquisition and outdated knowledge update, leading to rapid forgetting of newly learned information during training. TemporalWiki [95] constructs a series of temporal

corpora and their differential sets from sequential snapshots of Wikipedia, revealing that updating LLMs on these differential sets substantially enhances new knowledge acquisition and updates, requiring significantly less computational resources, and various CL techniques prove effective in mitigating horizontal forgetting during this process. LLPT [100] introduces temporal generalization evaluation for LLMs pre-trained on sequential corpora. Through experiments on a large-scale chronologically-ordered Tweet Stream, the authors demonstrate the superiority of CPT combined with CL techniques to task-specific LMs, in terms of both knowledge acquisition and temporal generalization. Nonetheless, these preliminary experiments do not conclusively determine which specific CL method is more preferable than the others.

Another line of work, Temporal Language Models (TLMs), takes a different approach to address knowledge retention, acquisition, and update under temporal shifts by integrating temporal information into the model [52, 198, 219]. During training, they inject temporal information into training examples as prefixes of prompts, using special tokens [198], explicit year information [52], or syntax-guided structural information [219]. In sequential training experiments conducted by TempoT5 [52], comparison between continually and jointly pre-trained LMs demonstrates that CPT better balances adaptation and forgetting when the replay rate of past data is appropriately set.

Others. CPT as a technique to progressively attain novel knowledge, can be used to refine LLMs' behavior. CEM [294] collects examples where the model's response is incorrect and continually trains the model on these examples, along with a supplemental dataset. RHO-1 [130] proposes Selective Language Modeling (SLM), which employs a reference model to evaluate the perplexity of each token in the training corpus, and continually pre-trains the model on high-perplexity tokens. Similarly, IR-DRO [36] re-trains the model on re-weighted examples from the original pre-training dataset, focusing more on higher-loss sequences.

The significance of addressing temporal shifts through CPT is underscored by several industrial studies. For instance, [5] employs a dynamic vocabulary expansion algorithm and an efficient sub-sampling procedure to conduct CPT on large-scale emerging tweet data. Conversely, [137] adopts CPT without explicit measures to constrain model updates, releasing a series of BERT-based LMs incrementally trained on new tweet data every three months. Preliminary experimental results demonstrate substantial improvements of continually pre-trained LMs over the base BERT model across downstream tasks. While some studies question the necessity of continually adapting LLMs along the temporal axis for environmental reasons, such as reducing CO_2 emissions [8], the community commonly embraces CPT as a more efficient learning paradigm compared to the traditional "combine-and-retrain" approach.

4.2 Domain-Adaptive Pre-training (DAP)

Background of DAP. Institutions, regardless of size, often possess significant amounts of unlabeled, domain-specific data. This data bridges the gap between general-purpose LLMs trained on diverse corpora and fine-tuned LLMs designed for specific downstream tasks. Leveraging this data as a preparatory stage can facilitate effective adaptation of LLMs to downstream tasks. Such process of "continued/continual/continuous pre-training" [9, 42, 68, 73, 91, 138, 148, 212, 264, 266, 268, 272, 282, 285], "further pre-training" [3, 48, 129, 200, 218], "domain tuning" [197], "knowledge enhancement pre-training" [138], and "knowledge injection training" [258] is unified and termed "***Domain Adaptive Pre-training (DAP)***" [72] for clarity and consistency throughout this survey. In the pioneering work of domain-adaptive pre-training (DAPT) [72], the authors continuously pre-train the language models on a larger domain-specific dataset before fine-tuning them to the downstream tasks, resulting in universally improved performance aross various tasks. As the observation above has been validated on multiple domains in parallel, including BioMed, CS, News, and Reviews [72], practitioners commonly accept that employing DAP on additional unlabeled domain-specific data benefits downstream tasks. Consequently, this technique has become widely deployed in many modern LLMs.

Summary of LLMs with DAP. We provide a summary of the existing 41 studies utilizing DAP for LLMs in Table 2. Each entry is characterized by three main features: (i) training process specifications, encompassing the vertical domain for which LLMs are trained, the training pipeline preceding release, and the LLM architecture employed; (ii) adopted continual learning techniques, including rehearsal, parameter regularization, and architecture expansion; and (iii) evaluation metrics for CL, such as backward transfer (forgetting) and forward transfer (adaptation to downstream data).

4.2.1 General Observation on DAP. Several key observations emerge regarding the research landscape of DAP (Table 2).

- **OBS-1: DAP predominantly occurs in a single stage.** Continual DAP which involves more than one stage is seldom explored: among all papers listed in Table 2, only one employs two stages of DAP ("PT → DAP → DAP → FT" in Code Llama [199]). It is arguably reasonable to categorize studies that conduct only one stage of DAP and nothing more [9, 39, 67, 138, 168, 177, 218, 226, 268, 271] into CPT rather than DAP. Nevertheless, considering that they aim to adapt a general-purpose LLM to a specific domain, we include them in this section.
- **OBS-2: The notion of interpreting DAP through the lens of CL, whether intentional or not, is widely embraced.** As shown in Table 2, except for the first section (white, 13/41), where papers overlook any potential side effects of DAP leading to vertical forgetting, the remaining sections (all gray, 28/41) either evaluate the potential negative impacts of DAP or proactively employ CL techniques to mitigate the risk of vertical forgetting.
- **OBS-3: Further research of more sophisticated CL techniques for not just DAP, but general vertical continual learning is much needed.** It is supported by the widespread adoption of CL techniques (22/41) for training domain-specific LLMs. However, the diversity of these techniques is limited, with only replay [9, 33, 39, 42, 91, 148, 197, 258, 274, 289] and parameter expansion (LoRA [177, 257, 271, 272]) or Layer/Block expansion [257, 272] utilized. In fact, it appears that individuals may not explicitly recognize that DAP should be viewed from the perspective of vertical continuity, as they often employ CL techniques unknowingly, e.g., studies deploying replay terming the technique as "data combination" [258] or "data mixing/mixture" [9, 39, 148, 274], without recognizing it as a typical CL solution to vertical continual learning.

4.2.2 Different Domains of DAP. We include work aimed at establishing vertical LLMs across various domains, including legal, medical, financial, scientific, and code. Additionally, we cover other domains such as language and e-commerce.

Legal Domain. In Layer Llama [91], the authors gathered publicly available legal texts from China Courts websites, totaling approximately 10 billion tokens as noted in a GitHub issue. In SaulLM [42], the authors collected the DAP corpus from various jurisdictions in different countries, resulting in a corpus of 30 billion tokens to cover diverse aspects of legal texts. When combined with previously available datasets, the total number of tokens used for legal-domain DAP reaches 94 billion. The substantial volume of DAP data, while offering valuable insights into specific domains, increases the risk of vertical forgetting of the general knowledge due to the large number of update steps involved. To mitigate this issue, SaulLM incorporates general data from Wikipedia, StackExchange, and GitHub into the DAP data, constituting about 2% of the final dataset [42]. Similarly, Lawyer Llama incorporates replaying general-domain data during DAP, but the replay rate is not disclosed [91]. [222] also replays of non-latest business documents during DAP when building a Japanese business-specific LLM.

Medical Domain. Efforts have been made to develop medical specialists by either training an LLM from scratch [66, 143] or fine-tuning publicly-available LLMs to meet specific medical needs [33, 146, 258]. Among

these approaches, DAP techniques have been extensively utilized to preserve the communication and instruction-following abilities of a general LLM, preparing it for subsequent medical applications [33, 146, 258]. BioMedGPT [146] is a multi-modal biomedical language model that integrates representations of human language and the language of life (molecules, proteins, cells, genes, etc.). Prior to final multi-modal supervised fine-tuning, the authors initialize the model from Llama2-Chat [231] and conduct DAP using extensive biomedical documents from S2ORC [134], without considering any CL techniques or evaluations. In [68], DAP is performed using Chinese medical encyclopedias and online expert articles, with next-token prediction as the training objective. During DAP, the performance gradually deteriorates on general-domain datasets as the training step increases, but improves on the downstream medical examination tasks [82]. PMC-LLama [258] gathers biomedical papers from S2ORC [134] and medical textbooks for "knowledge injection training." During this phase, a general language corpus from RedPajama-Data [43] is replayed at a 5% rate within a training batch. However, the paper does not analyze the effectiveness of this operation of mixing in general-domain data for DAP.

To mitigate vertical forgetting, AF Adapter [272] proposes an adapter structure extending the width of Attention layers and FFNs for acquiring domain knowledge and only the adapters are tuned during DAP. Similarly, Hippocrates [2] deploys LoRA during DAP to both have medical-specific knowledge injected and general ability preserved. Me-Llama [265] mixes in about 25% of the general-domain data for DAP on the clinical notes and biomedical articles, which achieves even positive backward transfer on MMLU [82]. HuatuoGPT-II [33] proposes to fuse the DAP into the final SFT, unifying the two stages into one single process. The challenge of such process mainly comes from the data heterogeneity of DAP's unlabeled corpus. The authors address this challenge by reformulating paragraphs of data into *(instruction, output)* format using existing large language models. They further employ a priority sampling strategy to avoid compromising downstream ability, a pitfall observed in the fixed-rate data mixing strategy [231]. This paper empirically demonstrates the superiority of unified one-stage SFT over two-stage training, questioning the reasonability of the current DAP. On medical-domain data, [197] finds that LMs constrained by CL techniques on source domains exhibit greater robustness to future domain shifts. Specifically, they identify that parameter regularization techniques like EWC [113], despite slightly higher cost, can facilitate positive forward and backward transfer.

Financial Domain. A gap persists between general-purpose LLMs and existing domain-specific smaller-scale LLMs [7, 259], underscoring the urgent need for more powerful financial-domain experts through the integration of LLMs. Notably, DAP techniques have emerged as crucial tools for tailoring LLMs to the intricacies of the financial domain while mitigating the negative effects of abrupt domain shifts from general to finance [121, 138, 268, 271, 289].

BBT-Fin [138] collects a Chinese financial DAP dataset comprising 80 billion tokens sourced from corporate reports, analyst reports, social media, and financial news. In addition to the conventional masked language modeling (MLM) training objective, BBT-Fin further incorporates triplet masking and span masking techniques during DAP. CFGPT [121] creates CFData, a financial dataset for DAP and SFT, comprising 141 billion tokens. During DAP, CFGPT does not employ CL techniques but utilizes QLoRA [50] for preventing overfitting to downstream data and balancing general response ability and domain-specific ability during SFT. These two methods are typical domain-specific LLMs focusing solely on adaptation to target domains without explicit CL measures or evaluation of vertical forgetting.

In [268], the authors aim to enhance the data efficiency of DAP. When the downstream tasks' data distribution $\mathcal{T}$ are known, based on the generalization bound [14, 61, 213], the authors propose to sample the subset of DAP data whose distribution $\mathcal{D}$ is similar to the downstream task's data, i.e., $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}, \mathcal{T})$ is low. When the downstream data distribution is unknown, the authors suggest ensuring *novelty* and *diversity* in the sampled corpus for DAP. This approach significantly enhances DAP efficiency: it utilizes only 10% of the originally collected data yet outperforms models trained on the entire DAP dataset, underscoring the importance of data quality over quantity. WeaverBird [271] introduces an intelligent finance dialogue system, where the encoder is

Table 2. **Summary of the existing studies that leverage Domain-Adaptive Pre-training of LLMs,** where the papers are organized in four main categories based on whether they (i) adopt the *continual learning techniques* and (ii) perform the evaluation for *backward transfer (forgetting)*. In the column of **Train Proc.** (Training Process), we omit the phase of general Pre-Training. DAP represents Domain-Adaptive Pre-Training; SFT represents Supervised Fine-Tuning; IT represents Instruction Tuning. The prefix G- and D- represent General and Domain-Specific training process [91, 129], and the prefix U- represents them unified [33, 257]. The prefix MM- and LC- represents Multi-Modal and Long-Context training phases [146, 199, 301]. In the column of **Continual Learning Eval.**, we consider two criteria: (i) *Backward Transfer*, i.e., performance degradation on the previous tasks, which is also known as catastrophic forgetting, (ii) *Forward Transfer*, i.e., the performance gained by DAP while transferring the LLMs to the downstream tasks. We use L and Perp. to denote Loss and Perplexity, FT to denote Fine-Tuning, ZS and FS to denote Zero-Shot and Few-Shot Accuracy, HE and LLM to denote the Human Evaluation and LLM Evaluation for generative tasks.

Domain	Method	Train Proc.	LLM Arch.	Continual Learning Tech.			Continual Learning Eval.	
				Rehearsal	*Param. Reg.*	*Arch. Exp.*	*Backward Transfer*	*Forward Transfer*
Medical	BioMedGPT [146]	DAP → MM-SFT	Llama2	✗	✗	✗	✗	FT
Financial	BBT-Fin [138]	DAP	T5	✗	✗	✗	✗	FT
Financial	CFGPT [121]	DAP → SFT	InternLM	✗	✗	Q-LoRA$_{(SFT)}$	✗	HE[1]
Scientific	AstroLlama [168]	DAP	LlaVa	✗	✗	✗	✗	Perp.
Scientific	OceanGPT [15]	DAP → IT	Vicuna Llama2-chat ChatGLM2	✗	✗	LoRA$_{(IT)}$	✗	HE
Scientific	K2 [48]	DAP → SFT	Llama	✗	✗	LoRA$_{(SFT)}$	✗	Perp. | ZS | LLM
Scientific	MarineGPT [301]	MM-DAP → MM-IT	Llama	✗	✗	✗	✗	HE
Code	CodeGen [172]	DAP → DAP	CodeGen	✗	✗	✗	✗	Perp. | ZS
Code	Comment-Aug [218]	IT → DAP	Llama2 Code Llama InternLM2	✗	✗	✗	✗	ZS
EventTemporal	EcoNet [73][1]	DAP → FT	BERT RoBERTa	✗	✗	✗	✗	FT
CommonSense	CALM [303]	DAP → FT	T5	✗	✗	✗	✗	FT
Multi-Domain	BLADE [120]	DAP → IT	BLOOMZ	✗	✗	✗	✗	ZS
Scientific	ClimateGPT [229]	DAP → IT → RAG	Llama2	✗	✗	✗	✗	FS | Ret.
Medical	[68]	DAP → FT	Llama2	✗	✗	✗	FS | FT	FS | FT
Financial	[268]	DAP	Pythia	✗	✗	✗	L | FS	L | FS
Scientific	GeoGalactica [129]	DAP → G-SFT → D-SFT	GAL	✗	✗	✗	ZS	Perp. | ZS | LLM
Code	StarCoder [226]	DAP	StarCoder	✗	✗	✗	Perp. | ZS | FS	Perp. | ZS | FS
Code	DeepSeek-Coder [67]	DAP	DeepSeek-LLM	✗	✗	✗	ZS | FS	ZS
Multi-Domain	DAPT [72]	DAP → FT	RoBERTa	✗	✗	✗	Loss	L | FT
Financial	WeaverBird [271]	DAP	GLM2	✗	✗	LoRA	✗	HE
Code	IRCoder [177]	DAP	StarCoder DeepSeek-Coder Code Llama	✗	✗	LoRA	✗	ZS
Code	Code Llama [199]	DAP → LC-FT → IT DAP → DAP → LC-FT	Llama2	Replay	✗	✗	✗	Perp. | ZS
Legal	SaulLM [42]	DAP → U-IT	Mistral	Replay	✗	✗	✗	Perp. | ZS
Medical	PMC-Llama [258]	DAP → IT	Llama	Replay	✗	✗	✗	ZS | FT
Scientific	Llema [9]	DAP	Code Llama	Replay	✗	✗	✗	Perp. | FS
Multi-Domain	DAS [107]	$[DAP]_n$	RoBERTa	DER++♣	EWC♣ HAT♣ Soft-Masking	Adapter♣ DEMix♣	✗	FT
Medical	Hippocrates [2]	DAP → IT → MA	Llama2 Mistral	✗	✗	LoRA	✗	ZS | FS
Language	Sailor [55]	DAP	Qwen1.5	Replay	✗	✗	✗	ZS
Code & Math	Llama Pro [257]	DAP → U-SFT	Llama2	✗	✗	Block Exp. LoRA♣	ZS | FS	Perp. | ZS | FS
Medical	AF Adapter [272]	DAP → FT	RoBERTa	✗	✗	Layer Exp. LoRA♣	Acc.	L | FT
Medical	[197]	DAP → FT	BERT RoBERTa DistilBERT	Replay♣ GEM♣	L2 Reg.♣ EWC♣	✗	L | FT	L | FT
Medical	HuatuoGPT-II [33]	DAP + U-SFT	Baichuan2	Replay	✗	✗	ZS	ZS | HE
Financial	XuanYuan 2.0 [289]	DAP + SFT	BLOOM	Replay	✗	✗	HE	HE
Scientific	PLlama [274]	DAP → IT	GAL	Replay	✗	✗	L	L | ZS
E-Commerce	EcomGPT-CT [148]	DAP → SFT	BLOOM	Replay	✗	✗	ZS | FS	ZS | FS
Legal	Layer Llama [91]	DAP → G-IT → D-IT	Llama	Replay	✗	✗	ZS	ZS
Multi-Domain	AdaptLLM [39]	DAP	Llama	Replay	✗	✗	ZS	ZS | FT
Language	Swallow [60]	DAP	Llama2	Replay	✗	✗	FS	FS
Financial	[222]	DAP	Llama2	Replay	✗	✗	Loss | ZS	Loss | ZS | FS | RAG
Medical	Me-Llama [265]	DAP → IT	Llama2	Replay	✗	✗	ZS | FS	ZS | FS | FT
Language	Aurora-M [167]	DAP → IT	StarCoder	Replay	✗	✗	ZS	ZS | FS | HE

trained on Chinese and English financial documents, alongside expert-annotated financial query-response pairs, using LoRA [87]. Xuanyuan 2.0 [289], akin to HuatuoGPT-II [33], proposes the technique of hybrid-tuning, which fuses the stages of DAP and SFT into one, general-domain data and financial-domain data into one. Notably, the distribution of data in hybrid-tuning is unconventional: financial DAP data comprises only a small portion of 13%. This prompts a pertinent question in line with the investigation on efficient DAP in [268]: Is a large DAP dataset necessary for developing a domain-specific LLM?

Scientific Domain. Vertical scientific LLMs span many subjects [9, 15, 129, 142, 168, 285, 301]. However, among all the studies listed above, only a small fraction of them adopt the technique of DAP. OceanGPT [15] is the first LLM tailored specifically for the ocean domain. It performs DAP on a raw corpus of ocean science literature, prioritizing recent research and historically significant works. K2 [48] pioneers the development of a foundational language model tailored specifically for geoscience. It aggregates geoscience open access literature and Earth science-related Wikipedia pages for DAP. Following this, it undergoes multi-task instruction tuning utilizing LoRA [87] on both a general instruction tuning dataset and the GeoSignal benchmark introduced within the K2 framework. AstroLlama [168] gathers abstracts solely from astronomy papers on arXiv and proceeds pre-training. It observes an improved perplexity on the domain of scholarly astronomy, without providing more quantitative evaluation. MarineGPT [301] is a multi-modal LLM designed specifically for the marine domain. During DAP, MarineGPT incorporates 5 million marine image-text pairs to imbue domain knowledge. This involves training a Q-Former [122] between the frozen visual and text decoder [54, 230].

Another branch of methods proactively integrate in the replay of the general-domain data to mitigate vertical forgetting. GeoGalactica [129] introduces a series of LLMs tailored for geoscience. In the DAP phase, besides the 52-billion-token geoscience corpus, Arxiv papers and Codedata are incorporated, with a mixing ratio of 8:1:1. The authors believe that the inclusion of the Codedata during the model's pre-training can significantly boost the reasoning ability of the LLMs. Although GeoGalactica pinpoints challenges of DAP, including overfitting, catastrophic forgetting, maintaining the training stability, and convergence speed, it does not further provide empirical evidence supporting the inclusion of the Codedata, or deploying specific measures to address the challenges proposed above. Llemma [9] focuses on mathematics, initialized from Code Llama [199], and undergoes DAP on a blend of the 55-billion-token mathematical pre-training dataset and general domain data at the ratio of 19:1. In contrast, PLlama [274], designed for plant science, mixes domain-specific and general-domain data at the ratio of 9:1.

Code Domain. The development of LLMs for automatic code filling, debugging, and generation holds significant practical importance [166, 220]. These advancements cover various frameworks, including encoder-only [166], encoder-decoder [242, 245], and decoder-only [67, 172, 227]. There is a growing trend towards decoder-only architectures [220], leveraging models pre-trained on general natural language like Llama [230, 231]. Consequently, there is a shift in the training objective from utilizing code structures to simpler tasks like next token prediction and infilling.

From the perspective of CL, the code domain presents unique advantages and challenges for DAP, compared to other domains. On one hand, its hierarchical structure (*general domain corpus* → *multi-language code* → *specific programming language*) provides an ideal training pipeline for DAPs [199], offering potential for more efficient training strategies. On the other hand, programming languages adhere to strict grammars, unlike the fuzzy and context-dependent natural language. Consequently, language models should ideally leverage these structures through tailored designs, and adopting the same training objectives as for natural languages may yield sub-optimal results. Therefore, many existing studies omit DAP [147, 242, 245]. In the following section, we will introduce existing code LLMs that employ DAP before the final downstream tasks, discussing both their common attributes and unique characteristics.

Representing a series of notable works that focus solely on adaptation to target domains, CodeGen [172] comprises a suite of LLMs designed for natural language (CodeGen-NL), multi-lingual programming languages (CodeGen-Multi), and mono-lingual programming languages (CodeGen-Mono). These models are trained sequentially, with each subsequent model initialized from the previous one trained on more general-domain data. Comment-Aug [218] addresses the challenge of aligning programming languages with natural languages (PL-NL alignment) by performing DAP on the code augmented with generated additional comments. StarCoder [226] introduces two models: StarCoderBase and StarCoder. StarCoderBase is initially trained on a mixed dataset comprising various programming languages without significant reweighting on the data. Subsequently, StarCoderBase undergoes further fine-tuning on additional 35 billion tokens of Python code, resulting in the development of StarCoder. DeepSeek-Coder-v1.5 [67] originates from DeepSeek-LLM [224] and undergoes pre-training on 2 trillion tokens, comprising 87% source code, 10% English code-related natural language, and 3% Chinese natural language corpus. Initialization from a general-domain LLM results in improved performance across various tasks, including natural language and mathematical reasoning, with minimal performance degradation on coding tasks, which underscores the efficacy of DAP.

As the only work that utilizes the general data replay to mitigate vertical forgetting in the code domain, Code Llama [199] introduces a sophisticated training framework tailored for various coding tasks and model sizes. Initialized from Llama 2 weights, these models undergo DAP on a dataset composed of deduplicated public code, discussions about code, and a subset of natural language data. This mix of natural language data serves as a form of pseudo-replay to maintain the models' proficiency in understanding natural language. Besides replay, architecture expansion has proven effective in acquiring robust coding abilities and preventing vertical forgetting simultaneously. IRCoder [177] utilizes compiler intermediate representations to enhance the multilingual transferability of Code LLMs. By conducting DAP on code grounded in intermediate representations with LoRA [86], IRCoder achieves superior multilingual programming instruction following, enhanced multilingual code understanding, and increased robustness to prompt perturbations. Llama Pro [257] undergoes DAP on a combination of code and math data. It expands the original Llama2 architecture by dynamically adding multiple identity copies of the transformer blocks. These added blocks initially preserves the original functionality, and will be tuned for DAP. The proposed expansion method is shown to be more resilient against vertical forgetting compared to other parameter-efficient tuning methods like LoRA.

The three aforementioned studies highlight the importance of DAP for code LLMs. However, it is crucial to note that the problem definition and conventional architectures of existing Code LLMs may present challenges of compatibility for DAP deployment, and need to be addressed in the future.

Other Domains. ECONET [73] enhances the model's ability to reason about event temporal relations through a dedicated DAP phase. Temporal and event indicators are masked out, and a contrastive loss is applied to the recovered masked tokens. Results demonstrate that incorporating this DAP stage significantly improves performance on final tasks compared to direct fine-tuning. Concept-Aware Language Model (CALM) [303] introduces a data-efficient DAP approach for enhancing the concept-centric commonsense reasoning ability of LLMs. It incorporates both generative and discriminative commonsense reasoning tasks specifically tailored for concept-centric reasoning tasks. Consequently, even a small number of data examples for DAP can lead to notable improvements for downstream tasks.

Aurora-M [167] and Swallow [60] adopt the simple replay strategy that mixes in a small portion of general data during DAP for their multi-lingual ability. Furthermore, Sailor [55] studies the optimal strategy of data mixing for DAP, balancing the general knowledge and capacity of different languages. EcomGPT-CT [148] employs a data mixing strategy for DAP which transforms semi-structured E-commerce data into a set of nodes and edges, samples a cluster of nodes, and then extracts and concatenates them into a training example. It combines the general-domain corpus with E-commerce data at a ratio of 2:1, which is significantly lower than the common setting adopted by other works.

Notably, there are some papers studying other effective ways of DAP. AdaptLLM [39] transforms raw corpora into *(raw text, question, answer)* format, creating intrinsic reading comprehension tasks. AdaptLLM demonstrates superior domain-specific knowledge adaptation and minimal vertical forgetting, thereby challenging the data efficiency of conventional DAP. Tag-LLM [212] re-purposes the general-domain LLM into domain-specific one by multi-stage training of domain tags and function tags, without modifying the base LLM's weights and thereby mitigates forgetting.

4.3 Continual Fine-Tuning (CFT)

Background of Continual Fine-Tuning (CFT). Continual Fine-Tuning (CFT) lies at the bottom layer of the vertical continuity, where models are trained on successive homogeneous tasks drawn from an evolving data distribution. As the service-oriented layer of LLM, it does not require consideration of further adaptation to another downstream tasks, simplifying optimization objectives to a great extent: better adaptation and less forgetting[2]. In the era of LLMs, new computational paradigms in CFT have emerged and attracted significant attention within the research community. These topics include (i) Continual Instruction Tuning (CIT) [292], (ii) Continual Model Refinement (CMR) [74], (iii) Continual Model Alignment (CMA) [128, 287], and (iv) Continual Learning for Multimodal Language Models (CMLLMs) [77, 171]. We summarize existing studies on CFT in Table 3, categorizing studies into sub-categories as listed above. The table includes details on incremental learning types (X-IL), LLM architecture, and employed CL techniques and evaluation metrics. After discussing general observations on CFT in Section 4.3.1, we will delve into each sub-category in detail.

4.3.1 General Observations on CFT. Examining the landscape of continual learning in the context of LLMs, and combined with the results shown in Table 3, we make several key observations about CFT.

- **OBS-1: There has been a noticeable transition in focus from CIL to TIL and DIL.** It has been a longstanding common sense in the CL community that CIL, as it requires the model to predict the context label and within-context label at the same time [112, 232, 237], is the most challenging CL scenario and hence receives most of the attention from the community. However, among all 35 papers presented in Table 3, only 3 papers study CFT of CIL. The transition of the research focus demonstrates the importance of TIL and DIL in the real-world applications of continual LLMs. More detailed discussion of this transition is included in Section 6.2.
- **OBS-2: In CFT, CL techniques enjoy broader adoption and explicit exploration compared to CPT and DAP.** In Table 3, all 35 papers explicitly deploy the CL techniques, 50% of which develop new techniques that cannot be easily interpreted as trivial combination of existing classic CL techniques, e.g., shared attentive learning framework in SAPT [297], external memory deployed in Larimar [45], and adaptive model averaging method to achieve Pareto-optimal in AMA [128], etc. This underscores the recognition of continual learning as a pivotal component in the development of resilient and adaptive LLMs.

4.3.2 General Continual Fine-Tuning (General CFT). Researchers have long investigated the phenomenon of forgetting resilience in pre-trained LLMs when fine-tuned for downstream tasks [106, 144, 156, 223, 299], despite some discover the opposite [144]. Although the pre-trained weights initially position the model in a flat-loss basin, aiding adaptation to future tasks without severely impacting previous ones [156], zero or near-zero forgetting is only observed at the representation level. This implies that while the model retains its ability to distinguish between task-specific representations, it may still forget specific task details [144, 223, 260, 299]. Therefore, additional measures are necessary when deploying these models in real-world applications [10, 37, 106, 182, 254, 281].

[2]We direct interested readers to additional survey literature on the topic of general CFT [17, 105].

Table 3. **Summary of the existing studies on Continual Fine-Tuning LLMs,** where the papers are organized in five main categories based on what downstream tasks they are designed to tackle, including (i) General Continual Fine-Tuning (CFT); (ii) Continual Instruction Tuning (CIT); (iii) Continual Model Refinement (CMR); (iv) Continual Model Alignment (CMA); (v) Continual Multimodal LLMs (CMLLMs), which is shown in the column of **CFT Type**. The column of **X-IL** shows what continual learning paradigm the study includes [232], where *TIL* represents task-incremental learning, meaning task ID/information is provided during inference; *DIL* represents domain-incremental learning, meaning the tasks are defined in the same format, and no task ID/information is available during inference; *CIL* represents class-incremental learning, meaning the task ID needs to be further inferred when testing.

CFT Type	Method	X-IL	LLM Arch.	Continual Learning Tech.				Continual Learning Eval.		
				Rehearsal	*Param. Reg.*	*Arch. Exp.*	*Others*	*Avg. Acc.*	*Bwd. Trans.*	*Fwd. Trans.*
General	CTR [106]	DIL | CIL	BERT	✗	✗	Adapter	✗	✓	✓	✓
	[223]	TIL	BERT	S-Replay	✗	✗	✗	♣	♣	♣
	CIRCLE [281]	DIL	T5	Replay	EWC	Prompt	✗	✓	✓	✓
	ConPET [217]	DIL	Llama	Replay	✗	LoRA	✗	✓	✓	✓
	[10]	DIL | CIL	BERT	✗	✗	✗	G-Prompt	✓	✓	✗
	[144]	TIL	DistilBERT ALBERT | RoBERTa	ER | DER | LwF	✗	✗	✗	♣	♣	✗
	SEQ* [299]	TIL | CIL	Pythia | BERT | GPT2	✗	P-Freeze	✗	Tricks for Classifiers	✗	✓	✗
	LFPT5 [182]	DIL	T5	P-Replay	✗	✗	✗	✓	✓	✗
	[254]	DIL	RoBERTa | GPT2	Replay	EWC | SI | RWalk	✗	✗	✓	✓	✗
	LR ADJUST [255]	DIL	XLM-R	✗	✗	✗	LR Scheduling	✓	✓	✓
	C3 [37]	TIL	T5	KD	✗	Prompt Tuning	✗	✓	✓	✗
CIT	CT0 [208]	TIL	T0	S-Replay	✗	✗	✗	✓	✓	✓
	RCL [241]	TIL	LLaMA Vicuna | Baichuan	Replay	✗	✗	✗	✓	✓	✓
	DynaInst [165]	TIL	BART	Replay	✗	✗	✗	✓	✓	✓
	CITB [292]	TIL	T5	Replay | AGEM	L2 | EWC	AdapterCL	✗	✓	✓	✓
	SSR [89]	TIL	LLaMA | Alpaca	RandSel | KMeansSel	✗	✗	✗	✓	✓	✓
	KPIG [80]	DIL | TIL	LLaMA | Baichuan	DynaInst | PCLL | DCL	L2 EWC	DARE LM-Cocktail	KPIG	✓	✓	✓
	ConTinTin [279]	TIL	BART	Replay	✗	✗	InstructionSpeak	✓	✓	✓
	O-LoRA [240]	TIL	LLaMA | Alpaca	✗	✗	O-LoRA	✗	✓	✓	✓
	SAPT [297]	TIL	T5 | LLaMA	✗	✗	✗	SAPT	✓	✓	✓
	InsCL [243]	TIL	LLaMA	Replay	✗	✗	InsCL	✓	✓	✓
CMR	CMR [125]	DIL	BART	ER | MIR | MLR	L2 | EWC	✗	✗	✓	✓	✓
	GRACE [74]	DIL	T5 | BERT | GPT2	✗	✗	Adapter	✗	✓	✓	✗
	WilKE [85]	DIL	GPT2 | GPT-J	✗	✗	Adaptor	✗	✓	✓	✓
	Larimar [45]	DIL	BERT | GPT-J	✗	✗	✗	Kanerva Memory	✓	✓	✓
	MELO [280]	DIL	BERT | GPT2 | T5	✗	✗	LoRA	✗	✓	✓	✓
	CME [123]	DIL	BERT	Replay	✗	✗	Inner-Prod. Reg.	✓	✓	✓
	WISE [238]	DIL	GPT-J | Llama2 | Mistral	✗	✗	✗	Side Memory	✓	✓	✓
CMA	COPF [286]	TIL | DIL	Llama	Replay	Function Reg.	Prompt	✗	✓	✗	✓
	AMA [128]	DIL	OpenLLaMA | Mistral	Replay	L1 | L2	LoRA	Adaptive Model Avg.	♣	♣	♣
	CPPO [287]	TIL	GPT2	✗	Weighting	Prompt	✗	✓	✓	✓
CMLLMs	EProj [77]	TIL	InstructBLIP	✗	TSIR	Projector Exp.	✗	✓	✗	✓
	Fwd-Prompt [298]	TIL	InstructBLIP | BLIP2	✗	✗	Projector Exp.	✗	✓	✓	✓
	CoIN [32]	TIL	LLaVA	✗	✗	MoE | LoRA	✗	✓	✗	✓
	Model Tailor [304]	TIL	InstructBLIP | LLaVA	✗	Model Tailor	✗	✗	✓	✓	✓
	RebQ [296]	TIL	ViLT	✗	✗	Prompt Tuning	✗	✓	✗	✓

Many studies advance beyond naive sequential fine-tuning, leveraging the inherent anti-forgetting nature of LLMs while avoiding the adoption of overly complex CL techniques [255, 299]. For instance, LR ADJUST [255] proposes a straightforward yet effective method of dynamically adjusting the learning rate to mitigate the overwriting of knowledge from new languages onto old ones. Building on the innate anti-forgetting ability of large language models like Pythia [16], SEQ* [299] introduces several strategies for fine-tuning LLMs on a sequence of downstream classification tasks, such as freezing the LLM and old classifier's parameters after warm-up, and pre-allocating future classifiers, etc.

Given the minimal forgetting observed at the representation level in CL, some studies aim to tackle the misalignment between the representation space and the decision-making layers by introducing representation-level constraints during CFT. NeiAttn [10] exemplifies this approach by formulating classification tasks as masked

language modeling and proposing a neighboring attention mechanism to counteract negative representation drift.

Another line of approaches refines the input/output format and network architectures of pre-trained LLMs to be better suited for CFT. For instance, CTR [106] incorporates two CL-plugin modules, i.e., a task-specific module (TSM) for acquiring task-specific knowledge and a knowledge-sharing module (KSM) for selectively transferring previously learned similar knowledge. CIRCLE [281] manually designs diverse prompt templates for various types of buggy code, unifying them as the cloze task and employs difficulty-based replay to enhance continual program repair. LFPT5 [182] addresses lifelong few-shot language learning by consolidating sequence labeling, text classification, and text generation into a text-to-text generation task. It undergoes prompt tuning on generated pseudo-examples from previous domains when adapting to new tasks. In [291], the authors propose a method for adaptively adding compositional adapters during continual sequence generation tasks. Before training on new domains, a decision stage determines which trained module can be reused. During training, this module also regenerates examples of the past for replay. C3 [37] merges PEFT and in-context learning (ICL) in a teacher-student framework. The teacher model undergoes in-context tuning focused solely on the current domain, while the student model, together with tunable prompts, minimizes the KL-divergence between the output distribution and the ground truth and teacher model simultaneously.

4.3.3 Continual Instruction Tuning (CIT). When the instruction tuning data comes in as a stream, forgetting of the previously learned instructions should be addressed. CT0 [208] represents the inaugural study on Continual Instruction Tuning (CIT) of LLMs, applying the replay method on the base T0 model throughout the process. Many subsequent studies focus on enhancing the replay method used during CIT. For instance, [80] improve replay efficiency by computing Key-Part Information Gain (KPIG) on masked parts to dynamically select replay data, addressing the "half-listening" issue in instruction following. Similarly, SSR [89] uses the LLM to generate synthetic instances for replay, achieving superior or comparable performance to traditional methods at a lower cost.

Other approaches introduce multiple CL techniques during CIT. DynaInst [165] merges parameter regularization with dynamic replay, selectively storing and replaying instances and tasks to enhance outcomes. InstructionSpeak [279] employs negative training and replay instructions to improve both forward transfer and backward transfer. Some methods incorporate PEFT. Orthogonal Low-Rank Adaptation (O-LoRA) learns new tasks within an orthogonal subspace while preserving LoRA parameters for previous tasks [240] to minimize the interference among different tasks. Shared Attention Framework (SAPT) combines a PET block with a selection module via a Shared Attentive Learning & Selection module, tackling catastrophic forgetting and knowledge transfer concurrently [297]. While regularization-based and architectural-based methods require additional parameter storage and GPU memory, together with replay-based methods they remain for CIT due to the simplicity and effectiveness [243].

4.3.4 Continual Model Refinement (CMR). The concept of model editing was initially explored in [215], which introduced a *"reliability-locality-efficiency"* principle and proposed a gradient descent editor to address it efficiently. Subsequent research, such as [47] and [163], extended this principle to edit factual knowledge in BERT-based language models and larger models like GPT-J-6B [235] and T5-XXL [189], respectively, using gradient decomposition. These approaches typically update a subset of model parameters to alter the labels of specific inputs. Additionally, memory-based models, as discussed in [164] and [74], incorporate editing through retrieval mechanisms.

Continual Model Refinement (CMR) extends model refinement horizontally, presenting updated sample pairs $(\boldsymbol{x}_e, y_e, \widehat{y}_e)_N^{e=1}$ sequentially as a stream. [125] initially introduces this idea, evaluating various CL methods with a dynamic sampling algorithm. Many CMR methods employ a retrieval mechanism. For instance, [74] uses

hidden activations of the language model as a "key" to activate updated parameters only when input x_0 resembles updated sample pairs; [280] improves this approach's efficiency by integrating LoRA [86]; [45] augments the LLM with an external episodic memory, modeling CMR as an ongoing memory refresh. Meanwhile, some methods focus solely on updating a subset of model parameters. For example, [85] addresses the issue of "toxicity buildup and flash" in single-editing methods like ROME [157], adapting it to the CL context with a knowledge-aware layer selection algorithm. WISE [238] addresses the "impossible triangle" of reliability, locality, and generalization in existing lifelong model refinement methods. It introduces a side memory system that enables knowledge sharding and merging, successfully achieving all three objectives simultaneously.

While all these works pioneer research in CMR, the exploration of CMR of LLMs remains open. [75] highlights a potential problem: the location for storing the fact may not coincide with the best place for editing it. This challenges the classical "locate and edit" paradigm used by several existing methods [157, 158], and could become a significent concern for CMR [85]. Other questions, including whether such problem setting fits LLMs and whether more memory/computationally efficient methods of CMR could be developed for LLMs, are yet to be answered.

4.3.5 Continual Model Alignment (CMA). When LLMs undergo the phase of MA, vertical forgetting of previous knowledge usually occurs. In [128], the authors refer to this phenomenon of catastrophic forgetting induced caused by MA as the "Alignment Tax." Notably, even a single stage of MA can diminish the model's performance capabilities, as it restricts the model's responses to a narrower subset of the training distribution.

Continual Model Alignment (CMA) aims to continuously refine LLMs to align with evolving human values, ethics, and data. The static nature of LLM training on historical data sets can lead to discrepancies between the models' outputs and current factual accuracies, societal norms, and standards, making CMA a crucial process for maintaining their adaptability and alignment with contemporary contexts. Likewise, there are two types of CMA frameworks: RL-based and SL-based. In the realm of RL-based CMA, two significant contributions have been noted. [128] identifies the conflicts between the existing CL techniques and RLHF, and proposes Adaptive Model Averaging (AMA), adaptively finding appropriate ratios for the combination of model layers to gain maximal rewards with minimal tax; Continual Proximal Policy Optimization (CPPO) [287] proposes a weighting strategy for different examples deciding its usage of policy enhancement or knowledge retention, mitigating the alignment tax over time. For SL-based CMA, Continual Optimal Policy Fitting (COPF) [286] presents a solution adapted from the Direct Policy Optimization (DPO) [188], solving its potential risks of sub-optimal policy fitting and over-optimization in the context of CMA.

4.3.6 Continual Multimodal Large Language Models (CMLLMs). Continually training multi-modal models like CLIP [185] has been long studied [171, 300], while the problem of continually training MLLMs still remains underexplored. Several existing studies have investigated the causes of catastrophic forgetting when continually training MLLMs. [298] performs singular value decomposition on input embeddings, revealing a significant disparity among different input embeddings. This discrepancy causes the model to learn irrelevant information for previously trained tasks, resulting in catastrophic forgetting and negative forward transfer. [284] observes that minority collapse may lead to catastrophic forgetting, when the imbalance ratio between majority and minority classes approaches infinity during fine-tuning. It further identifies hallucination as a contributing factor to performance degradation in MLLMs.

Continual Fine-Tuning MLLMs. In contrast to traditional continual learning methods that involve full-model fine-tuning for new tasks, continual fine-tuning for MLLMs focuses on refining specific layers when adapting to new tasks [32, 77, 284, 298, 304]. Given the strong capabilities of pre-trained models, training specific layers suffices, and can simultaneously reduce computational demands. [296] additionally considers an continual learning scenario, Continual Missing Modality Learning (CMML), where different modalities are emerging throughout the incremental learning stages. All the aforementioned studies collectively indicate that MLLMs

still suffer from catastrophic forgetting, which manifests in two ways: along the direction of *vertical continuity*, a performance decline on pre-trained tasks following fine-tuning for downstream tasks; and along the axis of *horizontal continuity*, a performance degrade on previously fine-tuned tasks after fine-tuning for new tasks. [298] also observes negative forward transfer, where the performance of unseen tasks degrades when learning new tasks, indicating a decline in model generalization capability.

While traditional CL methods are applicable, some may not yield optimal results, as evidenced by various experiments [77, 298]. For instance, [77] observes a consistent efficacy of replay-based and model expansion strategies across diverse scenarios of continual fine-tuning MLLMs, but regularization-based methods only perform well on models that have been jointly instruction-tuned on multiple tasks. Other works seek to develop ad-hoc solutions for continual learning MLLMs. [77] proposes EProj to expand the projection layer in MLLMs for each new task and utilizes task-similarity-informed regularization (TIR) to enhance performance. [298] introduces Fwd-Prompt, a prompt tuning method that projects prompt gradient to both the residual space and the pre-trained subspace to minimize the interference between tasks and reuse pre-trained knowledge respectively, fostering positive forward transfer without relying on previous samples. [304] focuses on the forgetting of the pre-trained MLLMs after fine-tuned on specific tasks and proposes model tailor to compensate the selected subset that are critical for enhancing target task performance. [296] presents a novel method named Reconstruct before Query (RebQ), leveraging the multi-modal knowledge from a pre-trained model to reconstruct the absent information for the missing modality. Recently, MoE (Mixture-of-Experts) framework has gained attention which resembles the architecture-based methods in CL. It provides the model with the ability to learn different intentions from distinct experts, e.g., [32] first introduces MoELoRA to fine-tune LLaVA, effectively mitigate the catastrophic forgetting of MLLMs in CoIN and the results demonstrate the effectiveness.

5 Evaluation Protocols and Datasets

Continual LLMs' Evaluation Protocols. LAnguage Model Analysis (LAMA) is an evaluation framework designed to *probe the world knowledge* embedded in language models [179]. LAMA converts each world fact into a cloze statement, which is then input into the language models to predict the correct answer. It has been extensively utilized in work on CPT under the temporal shifts [95, 96]. FUAR (Forgotten / (Updated + Acquired) Ratio) is proposed for CPT to address the **OP**'s drawback of not able to accurately reflect the model's behavior. A FUAR value of 1 represents an equal trade-off between the knowledge forgetting and knowledge learning, while a FUAR less than 1 suggests high learning efficacy. In TRACE [241], the authors propose a set of "**X-Delta**" metrics for continual instruction tuning, quantifying the forward transfer on specific abilities of LLMs, which is a straightforward extension of **FWT**. Specifically, the authors construct three sets of evaluation tasks to benchmark the ability of LLMs, including *general ability*, *instruction following*, and *safety*. For more detailed introduction to these evaluation protocols, please refer to Appendix B.2.

Datasets. In this section, we provide a comprehensive review of the datasets available for benchmarking continual LLMs, as illustrated in Table 4. We provide information about these datasets' types, what distributional shifts and semantic domains they include, and their sources and applications. We intentionally exclude datasets used for domain-adaptive pre-training LLMs in vertical domains such as legal, medical, and financial, unless they are specifically designed for continual domain-adaptive pre-training. Furthermore, we omit datasets used in general continual fine-tuning, as they have already been extensively studied in existing works [17, 105]. For details, please refer to Appendix B.3.

Table 4. **Summary of the existing benchmarks publicly available for Continual Learning LLMs.** In the column of **Name**, we use the superscript "*" to denote the lack of the dataset name and the name shown is that of the original paper. In this table, we deliberately omit the datasets used for domain-adaptive pre-training the vertical LLMs, as their main focus of development is not on continual learning. We also omit the datasets used for general continual fine-tuning, as they are extensively discussed in other existing surveys [17, 105].

Name	Type	Shift	Domain	#Stages	Scale	Sources	Applications	Comment
*TimeLMs [137]	CPT	Temporal	Social Media	8	#Examples: 123.86M	Tweets	[137]	code
CC-RecentNews [96]	CPT	Temporal	News	1	#Tokens: ∼168M	Web	[96]	code
TWiki [95]	CPT	Temporal	General Knowledge	5	#Tokens: 4.7B	Wikipedia	[95]	code
*DAPT [72]	CPT DAP	Content	Multi-Domain	4	Size: 160GB	BioMed [134], CS [134], News [283], Reviews [78]	[72] [183] [184]	code
*CPT [104]	CPT	Content	Multi-Domain	4	#Examples: 3.12M	Yelp [270], S2ORC [134], AG-News [290]	[104]	code
*DEMix [71]	CPT	Content	Multi-Domain	8	#Tokens: 73.8B	1B [31], CS [134], Legal [27], Med [134] WebText [64], RealNews [283], Reddit [13], Reviews [169]	[71]	code
*DAS [107]	CPT DAP	Content	Multi-Domain	6	Size: 4.16GB	Yelp [270], Reviews [169], Papers [134], PubMed	[107]	code
SuperNI [244]	CIT	Content	Mutli-Domain	16	#Tasks: 1616 #Examples: ∼5M	GitHub	[243, 292]	code
CITB [292]	CIT	Content	Mutli-Domain	19	#Tasks: 38	SuperNI [244]	[292]	code
CoIN [32]	CIT	Content	Multi-Domain	8	#Examples: ∼1.14M	RefCOCO [103],RefCOCO+ [151],RefCOCOg [151] ImageNet [49], VQAv2 [65], ScienceQA [140] TextVQA [214], GQA [93], VizWiz [70], OCR-VQA [160]	[32]	code
TRACE [241]	CIT	Content	Mutli-Domain	8	#Examples: 56,000	ScienceQA [140], FOMC [209], MeetingBank [88] C-STANCE [293], 20Minuten [108], CodeXGLUE [141], NumGLUE[162]	[241]	code
NATURAL-INSTRUCTION [161]	CIT	Content	Mutli-Domain	6	#Examples: 193k	CosmosQA [90], DROP [56], Essential-Terms [110] MCTACO [302], MultiRC [109], QASC [111] Quoref[46] , ROPES [127] , Winogrande [202]	[161]	code
IMDB [149]	CMA	Content	Social Media	1	Size: 217.35 MB	IMDB	[286]	code
HH-RLHF [11]	CMA	Content	General Knowledge	1	Size: 28.1 MB	Human Feedback	[286]	code
Reddit TL;DR [234]	CMA	Content	Social Media	2	Size: 19.6 GB	Reddit	[286, 287]	code
Common Sense QA [128] Reading Comprehension [128] Translation [128]	CMA	Content	Multi-Domain	6	#Examples: ∼ 41.16M	ARC Easy and Challenge [41], Race [115], PIQA [18] SQuAD [190], DROP [56] WMT 2014 French to English [19]	[128]	see sources
FEVER [228]	CMR	Content	General Knowledge	1	#Examples: 420k	Wikipedia	[47, 76]	code
VitaminC [206]	CMR	Content	General Knowledge	1	#Examples: 450k	Wikipedia	[164]	code
zsRE [117]	CMR	Content	General Knowledge	1	#Examples: 120M	Wikireading [83]	[45, 74–76, 157, 158]	-
T-rex [59]	CMR	Content	General Knowledge	1	#Examples: 11M	Dbpedia abstracts [23]	[53, 119]	code
NQ [114]	CMR	Content	General Knowledge	1	#Examples: 320k	Google queries, Wikipedia	[74]	code
CounterFact [157]	CMR	Content	General Knowledge	1	#Examples: 22k	zsRE [117]	[45, 85, 157, 280]	code
SCOTUS [28]	CMR	Temporal	Law	1	#Examples: 9.2k	Supreme Court Database	[74]	code

6 Discussion

6.1 Intriguing Properties Emergent in Continual LLMs

Beyond the well-established resilience of pre-trained large language models (LLMs) against catastrophic forgetting compared to downstream-specific models [106, 144, 156, 223, 299], there is a notable lack of exploration into other intriguing properties of LLMs when trained continually. In [275], it is observed that when fine-tuned sequentially and cyclically on a series of documents, large models exhibit a phenomenon known as "*anticipatory recovering*." This refers to the LLMs' ability to recover forgotten information on documents even before encountering them again. This suggests that LLMs may possess the capability of sequential memorization, which could pave the way for research into more complex structured learning environments as model parameters scale up.

6.2 Conventional Types of Incremental Learning

As mentioned in Section 2.2.1, three types of incremental learning are prevalent [232]. Among them, class-incremental learning (CIL) has historically attracted significant attention from the community [193, 262]. However, in the context of continually pre-training and adapting large language models (LLMs), we observe a decreased interest in CIL but an increased focus on task-incremental learning (TIL) and domain-incremental learning (DIL). Given that language models are inherently designed for content generation and are pre-trained with the pretext generative task of next-word prediction, it is natural to emphasize the patterns of generative tasks and integrate the traditional CIL paradigm into the broader framework of language modeling, discarding the incremental classification head [26, 210]. However, the declining attention to CIL does not suggest that it is not impactful in the field of continual learning for LLMs. Techniques such as vocabulary expansion [5, 44] and learning routing function in the MoE system [35] can be seen as an extension of expanding the classification head in CIL, and previously validated techniques of CIL can be directly applied.

The importance of DIL is self-evident, given the shared task definition and input-output format in continual pre-training (CPT) and domain-adaptive pre-training (DAP). On the other hand, TIL attracts significant interest as it plays a crucial role in instruction tuning, where instructions can be seen as natural-language-encoded task indices [80, 89, 165, 208, 240, 243, 279, 297]. It is worth noting that the boundary between TIL and DIL becomes somewhat blurred in continual instruction tuning. Language models demonstrate the capability to infer domain information for unseen instructions, suggesting a convergence of TIL and DIL in certain contexts.

6.3 Roles of Memory in Continual LLMs

Previous continual learning research, drawing inspiration from human learning patterns, primarily emphasizes the storage efficiency of past data. However, this focus may no longer hold true in the context of continual LLMs. In the direction of relaxing memory constraints, institutions with access to training data may opt to retain full access without restricting memory size, given that the cost of memory storage is more than affordable. In such scenarios, as highlighted in [233], the challenge shifts from storage efficiency to computational efficiency. To achieve continual learning goals, models must efficiently adapt to new data (efficient adaptation) and select key experiences for replay (efficient replay) [99, 268]. Therefore, it is essential to reassess the existing memory constraint and prioritize optimizing computational efficiency for continual learning of LLMs by restricting the number of updates and FLOPs [180, 247].

On the other end of the spectrum, studies with tightened memory constraints remain vital in modern continual learning of LLMs. As shown in Fig. 1, upstream suppliers of LLMs typically do not provide training data with the released model weights. Consequently, consumers must adapt these models to downstream data without access to the actual replay data. Various rehearsal-free continual strategies are applied in this scenario, such as collecting data examples from alternate sources [9, 42, 199, 258], leveraging the generative capabilities of LLMs to produce pseudo-examples for replay [182], and implementing regularization techniques in the parameter space [107, 197].

Continual learning under the strict memory constraint is also driven by data privacy concerns, where preserving data on the server side is prohibited. In these scenarios, researchers must rely on online continual learning methods [150, 181], where data examples are only utilized for training as they arrive in a stream, and numerous efforts are already underway to develop LLMs capable of operating under these constraints [20].

6.4 Prospective Directions

Theories of Continual LLMs. It is widely recognized that the continual learning community tends to prioritize empirical research over theoretical exploration. Nevertheless, there are efforts to establish theoretical foundations for CL. In [237], the authors utilize second-order Taylor expansions around optimal parameters to derive an inter-task generalization error bound based on the maximum eigenvalue and l_2-norm of parameter differences. Another line of approaches leverages task/domain discrepancies to construct a multi-task generalization bound. For instance, Unified Domain Incremental Learning (UDIL) in [213] proposes upper bounds for intra-domain and cross-domain distillation losses, unifying various replay-based DIL techniques under a single adaptive generalization bound. However, applying these existing theories directly to continual LLMs can be imprudent, given their pre-trained, large-scale nature. Consequently, there is a notable gap in research focusing on continually learning LLMs with robust theoretical guarantees and understanding the forgetting behaviors of LLMs from a theoretical perspective.

Efficient Replay for Knowledge Retention for Continual LLMs. While the storage budget can theoretically be infinite (Section 6.3), replaying past experiences without specific design can lead to inefficient updates in current domain learning, resulting in slow convergence. Beyond sparse replay solutions that control data mixture ratios [129, 199, 274], there is ongoing exploration of efficient replay for continual LLMs. For example, KPIG [80] enhances replay efficiency by calculating Key-Part Information Gain (KPIG) on masked segments, enabling the dynamic selection of replay data. [99] introduces a forgetting forecasting mechanism based on output changes during adaptation, later used for selective replay in continual model refinement (CMR). More sophisticated and accurate data mixing strategies and efficient replay sample selection mechanisms are needed and hence we mark it as a significant research focus in the future.

Continual LLMs with Controllable Memory. The long-term memory inherent in the whole set of parameters of LLMs often lacks interpretability and explicit manipulability, which is crucial in certain application areas such as machine unlearning [21], where the continually pre-trained models need to constantly roll back to a previous version predating the inclusion of the revoked data and retrain the model from that point onward. This example illustrates the benefits of equipping LLMs with an external, controllable memory. As part of continual model refinement (CMR), memory systems for continual learning have been explored in several studies. Larimar [45] suggests integrating the Kanerva Machine [263] as an episodic memory for multi-fact model editing. This memory system supports basic operations like *writing, reading, and generating*, as well as advanced operations such as *sequential writing and forgetting*. It enables one-shot knowledge updates without costly retraining or fine-tuning. Other memory systems like Hopfield Networks [192] hold promise for future investigation as well.

Continual LLMs with Custom Preferences. In service-oriented contexts, users often require different trade-offs between domain expertise, ethics, values, or tones of expression. Efficiently building customized LLMs for individual users and offering flexible adjustment options is a challenging task. Early attempts in this direction include Imprecise Bayesian Continual Learning (IBCL), which, under certain assumptions, guarantees the generation of Pareto-optimal models based on user preferences by combining two model posteriors in the parameter space [139]. While empirical validation is limited in scale, this approach paves the way for future research in this area.

7 Conclusion

In this work, we offer a comprehensive survey on continual LLMs, summarizing recent advancements in their training and deployment from a continual learning standpoint. We categorize the problems and tasks based on their positions within our proposed broader framework of modern stratified continual learning of LLMs. While there is a widespread and growing interest in this area across the community, we also note several missing cornerstones, including algorithmic diversity and a fundamental understanding of large models' behaviors such as knowledge forgetting, transfer, and acquisition. With a holistic yet detailed approach, we aim for this survey to inspire more practitioners to explore continual learning techniques, ultimately contributing to the development of robust and self-evolving AI systems.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Emre Can Acikgoz, Osman Batur İnce, Rayene Bench, Arda Anıl Boz, İlker Kesen, Aykut Erdem, and Erkut Erdem. 2024. Hippocrates: An Open-Source Framework for Advancing Large Language Models in Healthcare. *arXiv preprint arXiv:2404.16621* (2024).

[3] Mayank Agarwal, Yikang Shen, Bailin Wang, Yoon Kim, and Jie Chen. 2024. Structured Code Representations Enable Data-Efficient Adaptation of Code Language Models. arXiv:2401.10716 [cs.CL]

[4] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*. 139–154.

[5] Spurthi Amba Hombaiah, Tao Chen, Mingyang Zhang, Michael Bendersky, and Marc Najork. 2021. Dynamic language models for continuously evolving content. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2514–2524.

[6] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403* (2023).

[7] Dogu Araci. 2019. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. arXiv:1908.10063 [cs.CL]

[8] Giuseppe Attanasio, Debora Nozza, Federico Bianchi, and Dirk Hovy. 2023. Is It Worth the (Environmental) Cost? Limited Evidence for Temporal Adaptation via Continuous Training. arXiv:2210.07365 [cs.CL]

[9] Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An Open Language Model For Mathematics. *CoRR* abs/2310.10631 (2023). doi:10.48550/ARXIV.2310.10631 arXiv:2310.10631

[10] Xueying Bai, Jinghuan Shang, Yifan Sun, and Niranjan Balasubramanian. 2023. Enhancing Continual Learning with Global Prototypes: Counteracting Negative Representation Drift. arXiv:2205.12186 [cs.CL]

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

[12] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 65–72.

[13] Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The Pushshift Reddit Dataset. arXiv:2001.08435 [cs.SI]

[14] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79 (2010), 151–175.

[15] Zhen Bi, Ningyu Zhang, Yida Xue, Yixin Ou, Daxiong Ji, Guozhou Zheng, and Huajun Chen. 2023. OceanGPT: A Large Language Model for Ocean Science Tasks. *CoRR* abs/2310.02031 (2023). doi:10.48550/ARXIV.2310.02031 arXiv:2310.02031

[16] 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. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*. PMLR, 2397–2430.

[17] Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual Lifelong Learning in Natural Language Processing: A Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, Donia Scott, Nuria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, Barcelona, Spain (Online), 6523–6541. doi:10.18653/v1/2020.coling-main.574

[18] 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*, Vol. 34. 7432–7439.

[19] Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*. 12–58.

[20] Jorg Bornschein, Yazhe Li, and Amal Rannen-Triki. 2024. Transformers for Supervised Online Continual Learning. arXiv:2403.01554 [cs.LG]

[21] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2020. Machine Unlearning. arXiv:1912.03817 [cs.CR]

[22] 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 (2020), 1877–1901.

[23] Martin Brümmer, Milan Dojchinovski, and Sebastian Hellmann. 2016. Dbpedia abstracts: A large-scale, open, multilingual NLP training corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 3339–3343.

[24] 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 (2020), 15920–15930.

[25] He Cao, Zijing Liu, Xingyu Lu, Yuan Yao, and Yu Li. 2023. InstructMol: Multi-Modal Integration for Building a Versatile and Reliable Molecular Assistant in Drug Discovery. *CoRR* abs/2311.16208 (2023). doi:10.48550/ARXIV.2311.16208 arXiv:2311.16208

[26] Xusheng Cao, Haori Lu, Linlan Huang, Xialei Liu, and Ming-Ming Cheng. 2024. Generative Multi-modal Models are Good Class Incremental Learners. *IEEE Computer Vision and Pattern Recognition (CVPR)* (2024).

[27] Caselaw Access Project. 2018. Caselaw Access Project. https://case.law/

[28] Ilias Chalkidis, Tommaso Pasini, Sheng Zhang, Letizia Tomada, Sebastian Felix Schwemer, and Anders Søgaard. 2022. Fairlex: A multilingual benchmark for evaluating fairness in legal text processing. *arXiv preprint arXiv:2203.07228* (2022).

[29] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient Lifelong Learning with A-GEM. In *ICLR*.

[30] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. 2019. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486* (2019).

[31] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. arXiv:1312.3005 [cs.CL]

[32] Cheng Chen, Junchen Zhu, Xu Luo, Hengtao Shen, Lianli Gao, and Jingkuan Song. 2024. CoIN: A Benchmark of Continual Instruction tuNing for Multimodel Large Language Model. arXiv:2403.08350 [cs.CV]

[33] Junying Chen, Xidong Wang, Anningzhe Gao, Feng Jiang, Shunian Chen, Hongbo Zhang, Dingjie Song, Wenya Xie, Chuyi Kong, Jianquan Li, Xiang Wan, Haizhou Li, and Benyou Wang. 2023. HuatuoGPT-II, One-stage Training for Medical Adaption of LLMs. *CoRR* abs/2311.09774 (2023). doi:10.48550/ARXIV.2311.09774 arXiv:2311.09774

[34] Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and Learn: Fine-tuning Deep Pretrained Language Models with Less Forgetting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 7870–7881. doi:10.18653/v1/2020.emnlp-main.634

[35] 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*. PMLR, 5383–5395.

[36] Xuxi Chen, Zhendong Wang, Daouda Sow, Junjie Yang, Tianlong Chen, Yingbin Liang, Mingyuan Zhou, and Zhangyang Wang. 2024. Take the Bull by the Horns: Hard Sample-Reweighted Continual Training Improves LLM Generalization. *arXiv preprint arXiv:2402.14270* (2024).

[37] Yongrui Chen, Shenyu Zhang, Guilin Qi, and Xinnan Guo. 2024. Parameterizing Context: Unleashing the Power of Parameter-Efficient Fine-Tuning and In-Context Tuning for Continual Table Semantic Parsing. *Advances in Neural Information Processing Systems* 36 (2024).

[38] Zhiyuan Chen and Bing Liu. [n. d.]. *Lifelong machine learning*. Vol. 1. Springer.

[39] Daixuan Cheng, Shaohan Huang, and Furu Wei. 2024. Adapting Large Language Models via Reading Comprehension. arXiv:2309.09530 [cs.CL]

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

[41] 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* (2018).

[42] Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. 2024. SaulLM-7B: A pioneering Large Language Model for Law. arXiv:2403.03883 [cs.CL]

[43] Together Computer. 2023. RedPajama: an Open Dataset for Training Large Language Models. https://github.com/togethercomputer/RedPajama-Data
[44] Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. 2022. Continual Pre-Training Mitigates Forgetting in Language and Vision. arXiv:2205.09357 [cs.LG]
[45] Payel Das, Subhajit Chaudhury, Elliot Nelson, Igor Melnyk, Sarath Swaminathan, Sihui Dai, Aurélie Lozano, Georgios Kollias, Vijil Chenthamarakshan, Soham Dan, et al. 2024. Larimar: Large Language Models with Episodic Memory Control. *arXiv preprint arXiv:2403.11901* (2024).
[46] Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A Reading Comprehension Dataset with Questions Requiring Coreferential Reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 5925–5932. doi:10.18653/v1/D19-1606
[47] Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164* (2021).
[48] Cheng Deng, Tianhang Zhang, Zhongmou He, Yi Xu, Qiyuan Chen, Yuanyuan Shi, Luoyi Fu, Weinan Zhang, Xinbing Wang, Chenghu Zhou, Zhouhan Lin, and Junxian He. 2023. K2: A Foundation Language Model for Geoscience Knowledge Understanding and Utilization. arXiv:2306.05064 [cs.CL]
[49] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.
[50] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv preprint arXiv:2305.14314* (2023).
[51] 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* (2018).
[52] Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics* 10 (2022), 257–273.
[53] Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. *arXiv preprint arXiv:2210.03329* (2022).
[54] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).
[55] Longxu Dou, Qian Liu, Guangtao Zeng, Jia Guo, Jiahui Zhou, Wei Lu, and Min Lin. 2024. Sailor: Open Language Models for South-East Asia. *arXiv preprint arXiv:2404.03608* (2024).
[56] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161* (2019).
[57] Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. 2019. Uncertainty-guided continual learning with bayesian neural networks. *arXiv preprint arXiv:1906.02425* (2019).
[58] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. 2020. Adversarial continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 386–402.
[59] Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
[60] Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual Pre-Training for Cross-Lingual LLM Adaptation: Enhancing Japanese Language Capabilities. *arXiv preprint arXiv:2404.17790* (2024).
[61] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research* 17, 1 (2016), 2096–2030.
[62] Saurabh Garg, Mehrdad Farajtabar, Hadi Pouransari, Raviteja Vemulapalli, Sachin Mehta, Oncel Tuzel, Vaishaal Shankar, and Fartash Faghri. 2024. TiC-CLIP: Continual Training of CLIP Models. In *The Twelfth International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=TLADT8Wrhn
[63] Evangelia Gogoulou, Timothée Lesort, Magnus Boman, and Joakim Nivre. 2024. Continual Learning Under Language Shift. arXiv:2311.01200 [cs.CL]
[64] Aaron Gokaslan and Vanya Cohen. 2019. OpenWebText Corpus. http://Skylion007.github.io/OpenWebTextCorpus
[65] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. arXiv:1612.00837 [cs.CV]
[66] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ACM Transactions on Computing for Healthcare* 3, 1 (Oct. 2021), 1–23. doi:10.1145/3458754

[67] Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. 2024. DeepSeek-Coder: When the Large Language Model Meets Programming – The Rise of Code Intelligence. arXiv:2401.14196 [cs.SE]

[68] Zhen Guo and Yining Hua. 2023. Continuous Training and Fine-tuning for Domain-Specific Language Models in Medical Question Answering. arXiv:2311.00204 [cs.CL]

[69] 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:2308.04014 [cs.CL]

[70] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. 2018. VizWiz Grand Challenge: Answering Visual Questions from Blind People. arXiv:1802.08218 [cs.CV]

[71] Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2022. DEMix Layers: Disentangling Domains for Modular Language Modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.). Association for Computational Linguistics, Seattle, United States, 5557–5576. doi:10.18653/v1/2022.naacl-main.407

[72] 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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 8342–8360. doi:10.18653/v1/2020.acl-main.740

[73] Rujun Han, Xiang Ren, and Nanyun Peng. 2021. ECONET: Effective Continual Pretraining of Language Models for Event Temporal Reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 5367–5380. doi:10.18653/v1/2021.emnlp-main.436

[74] Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: Lifelong Model Editing with Discrete Key-Value Adaptors. In *Advances in Neural Information Processing Systems*.

[75] Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does Localization Inform Editing? Surprising Differences in Causality-Based Localization vs. *Knowledge Editing in Language Models* (2023).

[76] Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2021. Do language models have beliefs? methods for detecting, updating, and visualizing model beliefs. *arXiv preprint arXiv:2111.13654* (2021).

[77] Jinghan He, Haiyun Guo, Ming Tang, and Jinqiao Wang. 2023. Continual Instruction Tuning for Large Multimodal Models. arXiv:2311.16206 [cs.LG]

[78] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web* (Montréal, Québec, Canada) *(WWW '16)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 507–517. doi:10.1145/2872427.2883037

[79] Tianxing He, Jun Liu, Kyunghyun Cho, Myle Ott, Bing Liu, James Glass, and Fuchun Peng. 2021. Analyzing the Forgetting Problem in Pretrain-Finetuning of Open-domain Dialogue Response Models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (Eds.). Association for Computational Linguistics, Online, 1121–1133. doi:10.18653/v1/2021.eacl-main.95

[80] Yongquan He, Xuancheng Huang, Minghao Tang, Lingxun Meng, Xiang Li, Wei Lin, Wenyuan Zhang, and Yifu Gao. 2024. Don't Half-listen: Capturing Key-part Information in Continual Instruction Tuning. arXiv:2403.10056 [cs.CL]

[81] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2023. Aligning AI With Shared Human Values. arXiv:2008.02275 [cs.CY]

[82] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).

[83] Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. 2016. Wikireading: A novel large-scale language understanding task over wikipedia. *arXiv preprint arXiv:1608.03542* (2016).

[84] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556* (2022).

[85] Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. WilKE: Wise-Layer Knowledge Editor for Lifelong Knowledge Editing. arXiv:2402.10987 [cs.CL]

[86] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[87] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=nZeVKeeFYf9

[88] Yebowen Hu, Tim Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. MeetingBank: A Benchmark Dataset for Meeting Summarization. arXiv:2305.17529 [cs.CL]
[89] Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating Catastrophic Forgetting in Large Language Models with Self-Synthesized Rehearsal. arXiv:2403.01244 [cs.CL]
[90] Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning. arXiv:1909.00277 [cs.CL]
[91] 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* (2023).
[92] Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785* (2023).
[93] Drew A. Hudson and Christopher D. Manning. 2019. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. arXiv:1902.09506 [cs.CL]
[94] 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* (2024).
[95] Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. TemporalWiki: A Lifelong Benchmark for Training and Evaluating Ever-Evolving Language Models. *EMNLP 2022*.
[96] Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2022. Towards Continual Knowledge Learning of Language Models. In *ICLR*.
[97] Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, Fanzhi Zeng, Kwan Yee Ng, Juntao Dai, Xuehai Pan, Aidan O'Gara, Yingshan Lei, Hua Xu, Brian Tse, Jie Fu, Stephen McAleer, Yaodong Yang, Yizhou Wang, Song-Chun Zhu, Yike Guo, and Wen Gao. 2024. AI Alignment: A Comprehensive Survey. arXiv:2310.19852 [cs.AI]
[98] 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:2402.12847 [cs.CL]
[99] Xisen Jin and Xiang Ren. 2024. What Will My Model Forget? Forecasting Forgotten Examples in Language Model Refinement. arXiv:2402.01865 [cs.LG]
[100] Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2022. Lifelong Pretraining: Continually Adapting Language Models to Emerging Corpora. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, Angela Fan, Suzana Ilic, Thomas Wolf, and Matthias Gallé (Eds.). Association for Computational Linguistics, virtual+Dublin, 1–16. doi:10.18653/v1/2022.bigscience-1.1
[101] Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven Siegelbaum, A James Hudspeth, Sarah Mack, et al. 2000. *Principles of neural science*. Vol. 4. McGraw-hill New York.
[102] 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* (2020).
[103] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. ReferItGame: Referring to Objects in Photographs of Natural Scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). Association for Computational Linguistics, Doha, Qatar, 787–798. doi:10.3115/v1/D14-1086
[104] Zixuan Ke, Haowei Lin, Yijia Shao, Hu Xu, Lei Shu, and Bing Liu. 2022. Continual Training of Language Models for Few-Shot Learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 10205–10216. doi:10.18653/v1/2022.emnlp-main.695
[105] Zixuan Ke and Bing Liu. 2023. Continual Learning of Natural Language Processing Tasks: A Survey. arXiv:2211.12701 [cs.CL]
[106] Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Shu Lei. 2021. Achieving Forgetting Prevention and Knowledge Transfer in Continual Learning. In *NeurIPS*.
[107] Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2022. Continual Pre-training of Language Models. In *The Eleventh International Conference on Learning Representations*.
[108] Tannon Kew, Marek Kostrzewa, and Sarah Ebling. 2023. 20 Minuten: A Multi-task News Summarisation Dataset for German. In *Proceedings of the 8th edition of the Swiss Text Analytics Conference*, Hatem Ghorbel, Maria Sokhn, Mark Cieliebak, Manuela Hürlimann, Emmanuel de Salis, and Jonathan Guerne (Eds.). Association for Computational Linguistics, Neuchatel, Switzerland, 1–13. https://aclanthology.org/2023.swisstext-1.1
[109] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 252–262.
[110] Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2017. Learning What is Essential in Questions. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, Roger Levy and Lucia Specia (Eds.). Association for Computational Linguistics, Vancouver, Canada, 80–89. doi:10.18653/v1/K17-1010

[111] Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A Dataset for Question Answering via Sentence Composition. arXiv:1910.11473 [cs.CL]
[112] Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. 2022. A Theoretical Study on Solving Continual Learning. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 5065–5079. https://proceedings.neurips.cc/paper_files/paper/2022/file/20f44da80080d76bbc35bca0027f14e6-Paper-Conference.pdf
[113] 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 (2017), 3521–3526.
[114] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7 (2019), 453–466.
[115] 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* (2017).
[116] Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. 2021. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems* 34 (2021), 29348–29363.
[117] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115* (2017).
[118] Chen-An Li and Hung-Yi Lee. 2024. Examining Forgetting in Continual Pre-training of Aligned Large Language Models. arXiv:2401.03129 [cs.CL]
[119] Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2022. Large language models with controllable working memory. *arXiv preprint arXiv:2211.05110* (2022).
[120] Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Zhijing Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2024. BLADE: Enhancing Black-box Large Language Models with Small Domain-Specific Models. *arXiv preprint arXiv:2403.18365* (2024).
[121] Jiangtong Li, Yuxuan Bian, Guoxuan Wang, Yang Lei, Dawei Cheng, Zhijun Ding, and Changjun Jiang. 2023. CFGPT: Chinese Financial Assistant with Large Language Model. arXiv:2309.10654 [cs.CL]
[122] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. arXiv:2301.12597 [cs.CV]
[123] Linyang Li and Xipeng Qiu. 2023. CONTINUAL MODEL EVOLVEMENT WITH INNER-PRODUCT RESTRICTION. https://openreview.net/forum?id=fn0BQK5T8p
[124] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2017), 2935–2947.
[125] Bill Yuchen Lin, Sida Wang, Xi Lin, Robin Jia, Lin Xiao, Xiang Ren, and Scott Yih. 2022. On Continual Model Refinement in Out-of-Distribution Data Streams. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 3128–3139. doi:10.18653/v1/2022.acl-long.223
[126] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.
[127] Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning Over Paragraph Effects in Situations. arXiv:1908.05852 [cs.CL]
[128] Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. 2024. Mitigating the Alignment Tax of RLHF. arXiv:2309.06256 [cs.LG]
[129] Zhouhan Lin, Cheng Deng, Le Zhou, Tianhang Zhang, Yi Xu, Yutong Xu, Zhongmou He, Yuanyuan Shi, Beiya Dai, Yunchong Song, Boyi Zeng, Qiyuan Chen, Tao Shi, Tianyu Huang, Yiwei Xu, Shu Wang, Luoyi Fu, Weinan Zhang, Junxian He, Chao Ma, Yunqiang Zhu, Xinbing Wang, and Chenghu Zhou. 2023. GeoGalactica: A Scientific Large Language Model in Geoscience. arXiv:2401.00434 [cs.CL]
[130] 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* (2024).
[131] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning. arXiv:2304.08485 [cs.CV]
[132] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems* 35 (2022), 1950–1965.
[133] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
[134] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The Semantic Scholar Open Research Corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter,

and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 4969–4983. doi:10.18653/v1/2020.acl-main.447
[135] Vincenzo Lomonaco, Davide Maltoni, and Lorenzo Pellegrini. 2020. Rehearsal-Free Continual Learning over Small Non-I.I.D. Batches. arXiv:1907.03799 [cs.LG]
[136] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems* 30 (2017).
[137] Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. 2022. TimeLMs: Diachronic Language Models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Valerio Basile, Zornitsa Kozareva, and Sanja Stajner (Eds.). Association for Computational Linguistics, Dublin, Ireland, 251–260. doi:10.18653/v1/2022.acl-demo.25
[138] Dakuan Lu, Hengkui Wu, Jiaqing Liang, Yipei Xu, Qianyu He, Yipeng Geng, Mengkun Han, Yingsi Xin, and Yanghua Xiao. 2023. BBT-Fin: Comprehensive Construction of Chinese Financial Domain Pre-trained Language Model, Corpus and Benchmark. *CoRR* abs/2302.09432 (2023). doi:10.48550/ARXIV.2302.09432 arXiv:2302.09432
[139] Pengyuan Lu, Michele Caprio, Eric Eaton, and Insup Lee. 2023. IBCL: Zero-shot Model Generation for Task Trade-offs in Continual Learning. arXiv:2310.02995 [cs.LG]
[140] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. arXiv:2209.09513 [cs.CL]
[141] Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, Ming Gong, Ming Zhou, Nan Duan, Neel Sundaresan, Shao Kun Deng, Shengyu Fu, and Shujie Liu. 2021. CodeXGLUE: A Machine Learning Benchmark Dataset for Code Understanding and Generation. arXiv:2102.04664 [cs.SE]
[142] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583* (2023).
[143] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics* 23, 6 (Sept. 2022). doi:10.1093/bib/bbac409
[144] Yun Luo, Zhen Yang, Xuefeng Bai, Fandong Meng, Jie Zhou, and Yue Zhang. 2023. Investigating Forgetting in Pre-Trained Representations Through Continual Learning. arXiv:2305.05968 [cs.CL]
[145] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-tuning. arXiv:2308.08747 [cs.CL]
[146] Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Yushuai Wu, Mu Qiao, and Zaiqing Nie. 2023. Biomedgpt: Open multimodal generative pre-trained transformer for biomedicine. *arXiv preprint arXiv:2308.09442* (2023).
[147] Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. WizardCoder: Empowering Code Large Language Models with Evol-Instruct. arXiv:2306.08568 [cs.CL]
[148] Shirong Ma, Shen Huang, Shulin Huang, Xiaobin Wang, Yangning Li, Hai-Tao Zheng, Pengjun Xie, Fei Huang, and Yong Jiang. 2023. EcomGPT-CT: Continual Pre-training of E-commerce Large Language Models with Semi-structured Data. arXiv:2312.15696 [cs.CL]
[149] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 142–150.
[150] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. 2022. Online continual learning in image classification: An empirical survey. *Neurocomputing* 469 (2022), 28–51. doi:10.1016/j.neucom.2021.10.021
[151] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. 2016. Generation and Comprehension of Unambiguous Object Descriptions. arXiv:1511.02283 [cs.CV]
[152] Vittorio Mazzia, Alessandro Pedrani, Andrea Caciolai, Kay Rottmann, and Davide Bernardi. 2023. A Survey on Knowledge Editing of Neural Networks. *arXiv preprint arXiv:2310.19704* (2023).
[153] David McCaffary. 2021. Towards continual task learning in artificial neural networks: current approaches and insights from neuroscience. *arXiv preprint arXiv:2112.14146* (2021).
[154] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review* 102, 3 (1995), 419.
[155] Michael McCloskey and Neal J. Cohen. 1989. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. Psychology of Learning and Motivation, Vol. 24. Academic Press, 109–165. doi:10.1016/S0079-7421(08)60536-8
[156] 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 (2023), 1–50. http://jmlr.org/papers/v24/22-0496.html
[157] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems* 35 (2022), 17359–17372.

[158] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229* (2022).
[159] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837* (2022).
[160] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. OCR-VQA: Visual Question Answering by Reading Text in Images. In *ICDAR*.
[161] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Natural Instructions: Benchmarking Generalization to New Tasks from Natural Language Instructions. *arXiv preprint arXiv:2104.08773* (2021).
[162] Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A Suite of Fundamental yet Challenging Mathematical Reasoning Tasks. arXiv:2204.05660 [cs.CL]
[163] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309* (2021).
[164] Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*. PMLR, 15817–15831.
[165] Jisoo Mok, Jaeyoung Do, Sungjin Lee, Tara Taghavi, Seunghak Yu, and Sungroh Yoon. 2023. Large-scale Lifelong Learning of In-context Instructions and How to Tackle It. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 12573–12589. doi:10.18653/v1/2023.acl-long.703
[166] Arghavan Moradi Dakhel, Vahid Majdinasab, Amin Nikanjam, Foutse Khomh, Michel C. Desmarais, and Zhen Ming (Jack) Jiang. 2023. GitHub Copilot AI pair programmer: Asset or Liability? *Journal of Systems and Software* 203 (2023), 111734. doi:10.1016/j.jss.2023.111734
[167] Taishi Nakamura, Mayank Mishra, Simone Tedeschi, Yekun Chai, Jason T Stillerman, Felix Friedrich, Prateek Yadav, Tanmay Laud, Vu Minh Chien, Terry Yue Zhuo, et al. 2024. Aurora-M: The First Open Source Multilingual Language Model Red-teamed according to the US Executive Order. *arXiv preprint arXiv:2404.00399* (2024).
[168] Tuan Dung Nguyen, Yuan-Sen Ting, Ioana Ciuca, Charlie O'Neill, Zechang Sun, Maja Jablonska, Sandor Kruk, Ernest Perkowski, Jack W. Miller, Jason Li, Josh Peek, Kartheik Iyer, Tomasz Rózanski, Pranav Khetarpal, Sharaf Zaman, David Brodrick, Sergio J. Rodríguez Méndez, Thang Bui, Alyssa Goodman, Alberto Accomazzi, Jill P. Naiman, Jesse Cranney, Kevin Schawinski, and UniverseTBD. 2023. AstroLLaMA: Towards Specialized Foundation Models in Astronomy. *CoRR* abs/2309.06126 (2023). doi:10.48550/ARXIV.2309.06126 arXiv:2309.06126
[169] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 188–197. doi:10.18653/v1/D19-1018
[170] Zixuan Ni, Haizhou Shi, Siliang Tang, Longhui Wei, Qi Tian, and Yueting Zhuang. 2021. Revisiting catastrophic forgetting in class incremental learning. *arXiv preprint arXiv:2107.12308* (2021).
[171] Zixuan Ni, Longhui Wei, Siliang Tang, Yueting Zhuang, and Qi Tian. 2023. Continual vision-language representation learning with off-diagonal information. In *Proceedings of the 40th International Conference on Machine Learning*. 26129–26149.
[172] Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2023. CodeGen: An Open Large Language Model for Code with Multi-Turn Program Synthesis. *ICLR* (2023).
[173] OpenAI. 2022. Introducing chatgpt. [Online]. Available: https://openai.com/blog/chatgpt. (2022).
[174] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155 [cs.CL]
[175] Christophe Pallier, Stanislas Dehaene, J-B Poline, Denis LeBihan, A-M Argenti, Emmanuel Dupoux, and Jacques Mehler. 2003. Brain imaging of language plasticity in adopted adults: Can a second language replace the first? *Cerebral cortex* 13, 2 (2003), 155–161.
[176] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.
[177] Indraneil Paul, Jun Luo, Goran Glavaš, and Iryna Gurevych. 2024. IRCoder: Intermediate Representations Make Language Models Robust Multilingual Code Generators. arXiv:2403.03894 [cs.AI]
[178] Anastasia Pentina. 2016. *Theoretical foundations of multi-task lifelong learning*. Ph. D. Dissertation.
[179] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases?. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 2463–2473. doi:10.18653/v1/D19-1250
[180] Ameya Prabhu, Hasan Abed Al Kader Hammoud, Puneet K Dokania, Philip HS Torr, Ser-Nam Lim, Bernard Ghanem, and Adel Bibi. 2023. Computationally budgeted continual learning: What does matter?. In *Proceedings of the IEEE/CVF Conference on Computer Vision*

and Pattern Recognition. 3698–3707.
[181] Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, and Ozan Sener. 2023. Online Continual Learning Without the Storage Constraint. arXiv:2305.09253 [cs.CV]
[182] Chengwei Qin and Shafiq Joty. 2021. LFPT5: A Unified Framework for Lifelong Few-shot Language Learning Based on Prompt Tuning of T5. In *International Conference on Learning Representations*.
[183] Yujia Qin, Cheng Qian, Xu Han, Yankai Lin, Huadong Wang, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. Recyclable Tuning for Continual Pre-training. *arXiv preprint arXiv:2305.08702* (2023).
[184] Yujia Qin, Jiajie Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. ELLE: Efficient Lifelong Pre-training for Emerging Data. In *Findings of the Association for Computational Linguistics: ACL 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 2789–2810. doi:10.18653/v1/2022.findings-acl.220
[185] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.
[186] 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 (2019), 9.
[187] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2024).
[188] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2024).
[189] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.
[190] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822* (2018).
[191] Rahul Ramesh and Pratik Chaudhari. 2021. Model Zoo: A Growing" Brain" That Learns Continually. *arXiv preprint arXiv:2106.03027* (2021).
[192] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. 2021. Hopfield Networks is All You Need. arXiv:2008.02217 [cs.NE]
[193] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2001–2010.
[194] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530* (2024).
[195] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910* (2018).
[196] Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. Online structured laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems* 31 (2018).
[197] Subendhu Rongali, Abhyuday Jagannatha, Bhanu Pratap Singh Rawat, and Hong Yu. 2021. Continual Domain-Tuning for Pretrained Language Models. arXiv:2004.02288 [cs.CL]
[198] Guy D. Rosin, Ido Guy, and Kira Radinsky. 2022. Time Masking for Temporal Language Models. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) *(WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 833–841. doi:10.1145/3488560.3498529
[199] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2024. Code Llama: Open Foundation Models for Code. arXiv:2308.12950 [cs.CL]
[200] Andre Niyongabo Rubungo, Craig Arnold, Barry P. Rand, and Adji Bousso Dieng. 2023. LLM-Prop: Predicting Physical And Electronic Properties Of Crystalline Solids From Their Text Descriptions. *CoRR* abs/2310.14029 (2023). doi:10.48550/ARXIV.2310.14029 arXiv:2310.14029
[201] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671* (2016).
[202] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. arXiv:1907.10641 [cs.CL]

[203] Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. arXiv:2110.08207 [cs.LG]

[204] Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. 2023. Error Sensitivity Modulation based Experience Replay: Mitigating Abrupt Representation Drift in Continual Learning. *arXiv preprint arXiv:2302.11344* (2023).

[205] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs.LG]

[206] Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. *arXiv preprint arXiv:2103.08541* (2021).

[207] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*. PMLR, 4528–4537.

[208] 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*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 6107–6122. doi:10.18653/v1/2022.emnlp-main.410

[209] Agam Shah, Suvan Paturi, and Sudheer Chava. 2023. Trillion Dollar Words: A New Financial Dataset, Task & Market Analysis. arXiv:2305.07972 [cs.CL]

[210] Yijia Shao, Yiduo Guo, Dongyan Zhao, and Bing Liu. 2023. Class-Incremental Learning based on Label Generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 1263–1276. doi:10.18653/v1/2023.acl-short.109

[211] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538* (2017).

[212] Junhong Shen, Neil Tenenholtz, James Brian Hall, David Alvarez-Melis, and Nicolo Fusi. 2024. Tag-LLM: Repurposing General-Purpose LLMs for Specialized Domains. *arXiv preprint arXiv:2402.05140* (2024).

[213] Haizhou Shi and Hao Wang. 2024. A Unified Approach to Domain Incremental Learning with Memory: Theory and Algorithm. *Advances in Neural Information Processing Systems* 36 (2024).

[214] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards VQA Models That Can Read. arXiv:1904.08920 [cs.CL]

[215] Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. *arXiv preprint arXiv:2004.00345* (2020).

[216] Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. arXiv:2402.00159 [cs.CL]

[217] Chenyang Song, Xu Han, Zheni Zeng, Kuai Li, Chen Chen, Zhiyuan Liu, Maosong Sun, and Tao Yang. 2023. ConPET: Continual Parameter-Efficient Tuning for Large Language Models. arXiv:2309.14763 [cs.CL]

[218] Demin Song, Honglin Guo, Yunhua Zhou, Shuhao Xing, Yudong Wang, Zifan Song, Wenwei Zhang, Qipeng Guo, Hang Yan, Xipeng Qiu, and Dahua Lin. 2024. Code Needs Comments: Enhancing Code LLMs with Comment Augmentation. arXiv:2402.13013 [cs.CL]

[219] Zhaochen Su, Juntao Li, Zikang Zhang, Zihan Zhou, and Min Zhang. 2023. Efficient Continue Training of Temporal Language Model with Structural Information. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 6315–6329. doi:10.18653/v1/2023.findings-emnlp.418

[220] Qiushi Sun, Zhirui Chen, Fangzhi Xu, Kanzhi Cheng, Chang Ma, Zhangyue Yin, Jianing Wang, Chengcheng Han, Renyu Zhu, Shuai Yuan, Qipeng Guo, Xipeng Qiu, Pengcheng Yin, Xiaoli Li, Fei Yuan, Lingpeng Kong, Xiang Li, and Zhiyong Wu. 2024. A Survey of Neural Code Intelligence: Paradigms, Advances and Beyond. arXiv:2403.14734 [cs.SE]

[221] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 05 (Apr. 2020), 8968–8975. doi:10.1609/aaai.v34i05.6428

[222] Kosuke Takahashi, Takahiro Omi, Kosuke Arima, and Tatsuya Ishigaki. 2024. Pretraining and updating language-and domain-specific large language model: A case study in japanese business domain. *arXiv preprint arXiv:2404.08262* (2024).

[223] Mingxu Tao, Yansong Feng, and Dongyan Zhao. 2022. Can bert refrain from forgetting on sequential tasks? a probing study. In *The Eleventh International Conference on Learning Representations*.
[224] DeepSeek-AI Team. 2024. DeepSeek LLM: Scaling Open-Source Language Models with Longtermism. arXiv:2401.02954 [cs.CL]
[225] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).
[226] StarCode Team. 2023. StarCoder: may the source be with you! arXiv:2305.06161 [cs.CL]
[227] StarCoder2 Team. 2024. StarCoder 2 and The Stack v2: The Next Generation. arXiv:2402.19173 [cs.SE]
[228] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. *arXiv preprint arXiv:1803.05355* (2018).
[229] David Thulke, Yingbo Gao, Petrus Pelser, Rein Brune, Rricha Jalota, Floris Fok, Michael Ramos, Ian van Wyk, Abdallah Nasir, Hayden Goldstein, et al. 2024. ClimateGPT: Towards AI Synthesizing Interdisciplinary Research on Climate Change. *arXiv preprint arXiv:2401.09646* (2024).
[230] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
[231] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).
[232] Gido M Van de Ven, Tinne Tuytelaars, and Andreas S Tolias. 2022. Three types of incremental learning. *Nature Machine Intelligence* 4, 12 (2022), 1185–1197.
[233] Eli Verwimp, Rahaf Aljundi, Shai Ben-David, Matthias Bethge, Andrea Cossu, Alexander Gepperth, Tyler L. Hayes, Eyke Hüllermeier, Christopher Kanan, Dhireesha Kudithipudi, Christoph H. Lampert, Martin Mundt, Razvan Pascanu, Adrian Popescu, Andreas S. Tolias, Joost van de Weijer, Bing Liu, Vincenzo Lomonaco, Tinne Tuytelaars, and Gido M. van de Ven. 2024. Continual Learning: Applications and the Road Forward. arXiv:2311.11908 [cs.LG]
[234] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*. 59–63.
[235] Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.
[236] Liyuan Wang, Xingxing Zhang, Qian Li, Jun Zhu, and Yi Zhong. 2022. CoSCL: Cooperation of Small Continual Learners is Stronger Than a Big One. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*. Springer, 254–271.
[237] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024. A Comprehensive Survey of Continual Learning: Theory, Method and Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024), 1–20. doi:10.1109/TPAMI.2024.3367329
[238] Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024. WISE: Rethinking the Knowledge Memory for Lifelong Model Editing of Large Language Models. *arXiv preprint arXiv:2405.14768* (2024).
[239] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 1405–1418. doi:10.18653/v1/2021.findings-acl.121
[240] 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. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 10658–10671. doi:10.18653/v1/2023.findings-emnlp.715
[241] Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. TRACE: A Comprehensive Benchmark for Continual Learning in Large Language Models. arXiv:2310.06762 [cs.CL]
[242] Yue Wang, Hung Le, Akhilesh Deepak Gotmare, Nghi D. Q. Bui, Junnan Li, and Steven C. H. Hoi. 2023. CodeT5+: Open Code Large Language Models for Code Understanding and Generation. arXiv:2305.07922 [cs.CL]
[243] Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. 2024. InsCL: A Data-efficient Continual Learning Paradigm for Fine-tuning Large Language Models with Instructions. arXiv:2403.11435 [cs.CL]
[244] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks.

arXiv:2204.07705 [cs.CL]
[245] Yue Wang, Weishi Wang, Shafiq Joty, and Steven C.H. Hoi. 2021. CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation. In *EMNLP*.
[246] Zifeng Wang, Chun-Liang Li, Vincent Perot, Long T Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. 2024. CodecLM: Aligning Language Models with Tailored Synthetic Data. *arXiv preprint arXiv:2404.05875* (2024).
[247] Zifeng Wang, Zheng Zhan, Yifan Gong, Geng Yuan, Wei Niu, Tong Jian, Bin Ren, Stratis Ioannidis, Yanzhi Wang, and Jennifer Dy. 2022. Sparcl: Sparse continual learning on the edge. *Advances in Neural Information Processing Systems* 35 (2022), 20366–20380.
[248] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. 2022. DualPrompt: Complementary Prompting for Rehearsal-free Continual Learning. *European Conference on Computer Vision* (2022).
[249] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 139–149.
[250] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021).
[251] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models Are Zero-Shot Learners. arXiv:2109.01652 [cs.CL]
[252] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022).
[253] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.
[254] Martin Weyssow, Xin Zhou, Kisub Kim, David Lo, and Houari Sahraoui. 2023. On the Usage of Continual Learning for Out-of-Distribution Generalization in Pre-trained Language Models of Code. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (<conf-loc>, <city>San Francisco</city>, <state>CA</state>, <country>USA</country>, </conf-loc>) *(ESEC/FSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1470–1482. doi:10.1145/3611643.3616244
[255] Genta Winata, Lingjue Xie, Karthik Radhakrishnan, Shijie Wu, Xisen Jin, Pengxiang Cheng, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2023. Overcoming Catastrophic Forgetting in Massively Multilingual Continual Learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 768–777. doi:10.18653/v1/2023.findings-acl.48
[256] Martin Wistuba, Prabhu Teja Sivaprasad, Lukas Balles, and Giovanni Zappella. 2023. Continual learning with low rank adaptation. In *NeurIPS 2023 Workshop on Distribution Shifts (DistShifts)*. https://www.amazon.science/publications/continual-learning-with-low-rank-adaptation
[257] Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ping Luo, and Ying Shan. 2024. LLaMA Pro: Progressive LLaMA with Block Expansion. arXiv:2401.02415 [cs.CL]
[258] Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Towards building open-source language models for medicine. *arXiv preprint arXiv:2305.10415* 6 (2023).
[259] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David S. Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. *CoRR* abs/2303.17564 (2023). doi:10.48550/ARXIV.2303.17564 arXiv:2303.17564
[260] 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*.
[261] 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:2402.01364 [cs.CL]
[262] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. 2019. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 374–382.
[263] Yan Wu, Greg Wayne, Alex Graves, and Timothy Lillicrap. 2018. The kanerva machine: A generative distributed memory. *arXiv preprint arXiv:1804.01756* (2018).
[264] Jian Xie, Yidan Liang, Jingping Liu, Yanghua Xiao, Baohua Wu, and Shenghua Ni. 2023. QUERT: Continual Pre-training of Language Model for Query Understanding in Travel Domain Search. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (<conf-loc>, <city>Long Beach</city>, <state>CA</state>, <country>USA</country>, </conf-loc>) *(KDD '23)*. Association for Computing Machinery, New York, NY, USA, 5282–5291. doi:10.1145/3580305.3599891
[265] Qianqian Xie, Qingyu Chen, Aokun Chen, Cheng Peng, Yan Hu, Fongci Lin, Xueqing Peng, Jimin Huang, Jeffrey Zhang, Vipina Keloth, et al. 2024. Me LLaMA: Foundation Large Language Models for Medical Applications. *arXiv preprint arXiv:2402.12749* (2024).

[266] Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. PIXIU: A Large Language Model, Instruction Data and Evaluation Benchmark for Finance. *CoRR* abs/2306.05443 (2023). doi:10.48550/ARXIV.2306.05443 arXiv:2306.05443

[267] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. 2024. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems* 36 (2024).

[268] Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. 2023. Efficient Continual Pre-training for Building Domain Specific Large Language Models. arXiv:2311.08545 [cs.CL]

[269] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244* (2023).

[270] Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. arXiv:1904.02232 [cs.CL] https://arxiv.org/abs/1904.02232

[271] Siqiao Xue, Fan Zhou, Yi Xu, Hongyu Zhao, Shuo Xie, Qingyang Dai, Caigao Jiang, James Zhang, Jun Zhou, Dacheng Xiu, and Hongyuan Mei. 2023. WeaverBird: Empowering Financial Decision-Making with Large Language Model, Knowledge Base, and Search Engine. *CoRR* abs/2308.05361 (2023). doi:10.48550/ARXIV.2308.05361 arXiv:2308.05361

[272] Y. Yan, K. Xue, X. Shi, Q. Ye, J. Liu, and T. Ruan. 2023. AF Adapter: Continual Pretraining for Building Chinese Biomedical Language Model. In *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE Computer Society, Los Alamitos, CA, USA, 953–957. doi:10.1109/BIBM58861.2023.10385733

[273] Shu Yang, Muhammad Asif Ali, Cheng-Long Wang, Lijie Hu, and Di Wang. 2024. MoRAL: MoE Augmented LoRA for LLMs' Lifelong Learning. arXiv:2402.11260 [cs.CL]

[274] Xianjun Yang, Junfeng Gao, Wenxin Xue, and Erik Alexandersson. 2024. PLLaMa: An Open-source Large Language Model for Plant Science. *CoRR* abs/2401.01600 (2024). doi:10.48550/ARXIV.2401.01600 arXiv:2401.01600

[275] Yanlai Yang, Matt Jones, Michael C. Mozer, and Mengye Ren. 2024. Reawakening knowledge: Anticipatory recovery from catastrophic interference via structured training. arXiv:2403.09613 [cs.LG]

[276] Yutao Yang, Jie Zhou, Xuanwen Ding, Tianyu Huai, Shunyu Liu, Qin Chen, Liang He, and Yuan Xie. 2024. Recent Advances of Foundation Language Models-based Continual Learning: A Survey. *arXiv preprint arXiv:2405.18653* (2024).

[277] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems* 36 (2024).

[278] 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* (2024).

[279] Wenpeng Yin, Jia Li, and Caiming Xiong. 2022. ConTinTin: Continual Learning from Task Instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 3062–3072. doi:10.18653/v1/2022.acl-long.218

[280] Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2023. MELO: Enhancing Model Editing with Neuron-Indexed Dynamic LoRA. *arXiv preprint arXiv:2312.11795* (2023).

[281] Wei Yuan, Quanjun Zhang, Tieke He, Chunrong Fang, Nguyen Quoc Viet Hung, Xiaodong Hao, and Hongzhi Yin. 2022. CIRCLE: continual repair across programming languages. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis* (<conf-loc>, <city>Virtual</city>, <country>South Korea</country>, </conf-loc>) *(ISSTA 2022)*. Association for Computing Machinery, New York, NY, USA, 678–690. doi:10.1145/3533767.3534219

[282] Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653* (2023).

[283] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems* 32 (2019).

[284] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the Catastrophic Forgetting in Multimodal Large Language Models. arXiv:2309.10313 [cs.CL]

[285] Dan Zhang, Ziniu Hu, Sining Zhoubian, Zhengxiao Du, Kaiyu Yang, Zihan Wang, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. SciGLM: Training Scientific Language Models with Self-Reflective Instruction Annotation and Tuning. *CoRR* abs/2401.07950 (2024). doi:10.48550/ARXIV.2401.07950 arXiv:2401.07950

[286] Han Zhang, Lin Gui, Yuanzhao Zhai, Hui Wang, Yu Lei, and Ruifeng Xu. 2023. Copf: Continual learning human preference through optimal policy fitting. *arXiv preprint arXiv:2310.15694* (2023).

[287] Han Zhang, Yu Lei, Lin Gui, Min Yang, Yulan He, Hui Wang, and Ruifeng Xu. [n. d.]. CPPO: Continual Learning for Reinforcement Learning with Human Feedback. ([n. d.]).

[288] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction Tuning for Large Language Models: A Survey. arXiv:2308.10792 [cs.CL]

[289] Xuanyu Zhang and Qing Yang. 2023. XuanYuan 2.0: A Large Chinese Financial Chat Model with Hundreds of Billions Parameters. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (<conf-loc>, <city>Birmingham</city>,

<country>United Kingdom</country>, </conf-loc>) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 4435–4439. doi:10.1145/3583780.3615285

[290] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf

[291] Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. 2022. Continual Sequence Generation with Adaptive Compositional Modules. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 3653–3667. doi:10.18653/v1/2022.acl-long.255

[292] Zihan Zhang, Meng Fang, Ling Chen, and Mohammad-Reza Namazi-Rad. 2023. CITB: A Benchmark for Continual Instruction Tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9443–9455. doi:10.18653/v1/2023.findings-emnlp.633

[293] Chenye Zhao, Yingjie Li, and Cornelia Caragea. 2023. C-STANCE: A Large Dataset for Chinese Zero-Shot Stance Detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 13369–13385. doi:10.18653/v1/2023.acl-long.747

[294] Haokun Zhao, Haixia Han, Jie Shi, Chengyu Du, Jiaqing Liang, and Yanghua Xiao. 2024. Large Language Model Can Continue Evolving From Mistakes. *arXiv preprint arXiv:2404.08707* (2024).

[295] Hanbin Zhao, Hui Wang, Yongjian Fu, Fei Wu, and Xi Li. 2022. Memory-Efficient Class-Incremental Learning for Image Classification. *IEEE Transactions on Neural Networks and Learning Systems* 33, 10 (2022), 5966–5977. doi:10.1109/TNNLS.2021.3072041

[296] Shu Zhao, Xiaohan Zou, Tan Yu, and Huijuan Xu. 2024. Reconstruct before Query: Continual Missing Modality Learning with Decomposed Prompt Collaboration. arXiv:2403.11373 [cs.CV]

[297] Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024. SAPT: A Shared Attention Framework for Parameter-Efficient Continual Learning of Large Language Models. arXiv:2401.08295 [cs.CL]

[298] Junhao Zheng, Qianli Ma, Zhen Liu, Binquan Wu, and Huawen Feng. 2024. Beyond Anti-Forgetting: Multimodal Continual Instruction Tuning with Positive Forward Transfer. arXiv:2401.09181 [cs.LG]

[299] Junhao Zheng, Shengjie Qiu, and Qianli Ma. 2023. Learn or Recall? Revisiting Incremental Learning with Pre-trained Language Models. arXiv:2312.07887 [cs.CL]

[300] Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. 2023. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19125–19136.

[301] Ziqiang Zheng, Jipeng Zhang, Tuan-Anh Vu, Shizhe Diao, Yue Him Wong Tim, and Sai-Kit Yeung. 2023. MarineGPT: Unlocking Secrets of Ocean to the Public. *CoRR* abs/2310.13596 (2023). doi:10.48550/ARXIV.2310.13596 arXiv:2310.13596

[302] Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "Going on a vacation" takes longer than "Going for a walk": A Study of Temporal Commonsense Understanding. arXiv:1909.03065 [cs.CL] https://arxiv.org/abs/1909.03065

[303] Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, Bill Yuchen Lin, and Xiang Ren. 2021. Pre-training text-to-text transformers for concept-centric common sense. (2021).

[304] Didi Zhu, Zhongyi Sun, Zexi Li, Tao Shen, Ke Yan, Shouhong Ding, Kun Kuang, and Chao Wu. 2024. Model Tailor: Mitigating Catastrophic Forgetting in Multi-modal Large Language Models. arXiv:2402.12048 [cs.CL]

Received 1 July 2024; revised 2 May 2025; accepted 8 May 2025