Continual Learning of Large Language Models: A Comprehensive Survey

Haizhou Shi [*1] **Zihao Xu** [1] **Hengyi Wang** [1] **Weiyi Qin** [1]

Wenyuan Wang [2] **Yibin Wang** [3] **Hao Wang** [*1]

Abstract

The recent success of large language models (LLMs) trained on static, pre-collected, general datasets has sparked numerous research directions and applications. One such direction addresses the non-trivial challenge of integrating pre-trained LLMs into dynamic data distributions, task structures, and user preferences. The primary challenge of this problem lies in balancing model adaptation and knowledge preservation. Pre-trained LLMs, when tailored for specific needs, often experience significant performance degradation in previous knowledge domains – a phenomenon known as *"catastrophic forgetting"*. While extensively studied in the continual learning (CL) community, it presents new manifestations in the realm of LLMs. In this survey, we provide a comprehensive overview and detailed discussion of the current research progress on large language models within the context of continual learning. Besides the introduction of the preliminary knowledge, this survey is structured into four main 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). Then we provide an overview of evaluation protocols for continual learning with LLMs, along with the current available data sources (Section 5). Finally, we discuss intriguing questions pertaining 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.

1 Introduction

Recent advances in large language models (LLMs) have demonstrated considerable potential for achieving artificial general intelligence (AGI) [233, 27, 217, 2, 50, 7, 279, 280, 124]. Researchers have observed that complex abilities such as multi-step reasoning, few-shot in-context learning, and

[1]Rutgers University [2]Wuhan University [3]Huazhong University of Science and Technology.
[*]Correspondence to: Haizhou Shi <haizhou.shi@rutgers.edu>, Hao Wang <hw488@cs.rutgers.edu>.

Pre-print, work in progress.

instruction following improve as the scale of parameter size increases [304, 303, 334, 301, 198]. The development of LLMs is impactful and revolutionary, prompting machine learning practitioners to reconsider traditional computational paradigms for once-challenging human-level tasks such as question answering, machine translation, and dialogue systems [143, 11, 65]. However, LLMs are typically trained on static, pre-collected datasets encompassing general domains, leading to gradual performance degradation over time [175, 120, 127, 119, 6, 68] and across different content domains [89, 127, 131, 273, 55, 91, 231, 46, 232]. Additionally, a single pre-trained large model cannot meet every user need and requires further fine-tuning [306, 307, 365, 307, 21, 365, 12, 133, 342, 230, 47]. 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, also known as *lifelong learning* or *incremental learning* [223, 48, 282, 288]. Continual learning, inspired by the incremental learning pattern observed in human brains [194, 128, 219, 328, 54, 216, 170, 193], involves training machine learning models sequentially on a series of tasks with the expectation of maintaining performance across all tasks [140, 161, 347, 240, 29, 80, 75, 74]. 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 [161, 265, 99, 173, 38, 240, 29, 260]. This challenge, known as *catastrophic forgetting*, has been a central focus in continual learning research since its inception. Over the years, researchers have explored various techniques to mitigate forgetting in machine learning models. These include replay-based methods [38, 254, 240, 29, 260], parameter regularization [140, 241, 4, 270], and model architecture expansion [237, 287]. 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 better highlight this ongoing shift, in this survey, we provide a comprehensive overview and detailed discussion of the current research progress on LLMs within the context of CL. For the general picture of continually learning LLMs, we divide it into two directions of continuity that need to be addressed by practitioners (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 catastrophic forgetting. For example, social media platforms continuously update LLMs to reflect recent trends, ensuring accurate targeting of downstream services like advertising and recommendations while maintaining a seamless user experience for existing users.

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) (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, while it is primarily seen as the procedure of preparing LLMs for downstream tasks, 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 (Section 5). We conclude our survey with a discussion covering recent emergent properties of continual LLMs, changes in the roles of conventional incremental learning types and memory constraints within the context of continual LLMs, and prospective research directions for this subject (Section 6).

In summary, this paper provides a comprehensive view of existing continual learning studies for LLMs in detail, which significantly distinguishes itself from existing literature on related topics [22, 132, 288, 314]. Our survey highlights the underexplored research area of continually developing LLMs, especially in the field of continual pre-training (CPT) and domain adaptive pre-training (DAP). We emphasize the needs for increased attention from the community, with urgent needs including the development of practical, accessible, and widely acknowledged evaluation benchmarks. Additionally, methodologies need to be tailored to address forgetting in emerging large language model learning paradigms. We hope this survey can provide a systematic and novel view 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 [119, 271, 323, 32, 9].

Organization. The rest of this paper is organized as follows. We will first start by introducing the background and preliminaries of large language models and continual learning in Section 2. Then we present the overview of continual learning in the modern era of large language models in Section 3. Vertically, it can be roughly divided into three stages of continual training LLMs, and we will present a one-by-one survey of each stage in Section 4. In Section 4.3, the unique aspects of continual fine-tuning LLMs will be introduced, including continual instruction tuning (Section 4.3.3), continual model refinement (Section 4.3.4), continual model alignment (Section 4.3.5), and continual multimodal large language models (Section 4.3.6). In Section 5, we give an inclusive introduction to the evaluation protocols and benchmarks of continual learning for LLMs that are publicly available. Finally, in Section 6, we present a series of discussion of the role of continual learning in the era of large language models, including emergent abilities in large-scale continual LLMs (Section 6.1), three types of continual learning (Section 6.2), roles of memory in continual learning of LLMs (Section 6.3), and prospective future directions (Section 6.4).

2 Preliminaries

In this section, we provide an overview of the fundamental concepts of large language models (LLMs) and continual learning (CL), ensuring clarity and comprehensibility for readers unfamiliar with these topics. We begin by introducing the notation used in this paper. Subsequently, we discuss the pre-training and downstream adaptation of LLMs, as well as mainstream LLM families (Section 2.1), followed by an introduction to basic continual learning techniques studied by the community (Section 2.2).

Notation. We denote scalars with lowercase letters, vectors with lowercase boldface letters, and matrices with uppercase boldface letters. The l_2-norm of vectors and the Frobenius norm of a matrix are represented by $\|\cdot\|_2$. For a vector $\boldsymbol{v} = [v_1, v_2, \cdots, v_n]^\top$, $\|\boldsymbol{v}\|_2 = (\sum_{i=1}^n v_i^2)^{1/2}$; for a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\|\boldsymbol{A}\|_2 = (\sum_{ij} A_{ij}^2)^{1/2}$. We use $\epsilon_{\mathcal{D}}$, $\mathcal{L}_{\mathcal{D}}$ to denote the error function, and loss function that is deployed for training, respectively, where the subscript is used to denote the error/loss measured by taking the expectation on the data distribution $\mathcal{D}$. We further use $\widehat{\mathcal{L}}_S$ to represent the empirical evaluation of the loss function $\mathcal{L}$ over the set of examples S. Probability and expectation are denoted by P and $\mathbb{E}$, respectively. We use $[m]$ to denote the set of positive integers up to m, $\{1, \cdots, m\}$.

2.1 Large Language Models

In the past two decades, neural language modeling has emerged as the dominant field of deep learning, marked by significant and rapid advancements. Primarily built on the transformer architecture, pre-trained language models (PLMs) like BERT have established a universal hidden embedding space through extensive pre-training on large-scale unlabeled text corpora. Following the pre-training and fine-tuning paradigms, PLMs exhibit promising performance across various natural language processing tasks after being fine-tuned upon small amounts of task-specific data [67, 171, 235]. Research on scaling laws indicates that increasing model size enhances the capacity of language modelss [129, 107]. By scaling parameters to billions or even hundreds of billions and training on massive text datasets, 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, which are absent in small-scale language models like BERT [304, 303, 334, 301, 198]. These larger models are commonly referred to as Large Language Models (LLMs).

2.1.1 Pre-Training of LLMs

Pre-training is essential for language models to acquire broad language representations. Decoder-only models typically employ probability language modeling (LM) tasks during pre-training. LM, in this context, specifically refers to auto-regressive LM. 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:

$$\mathcal{L}_{\mathrm{LM}}(\boldsymbol{x}) \triangleq -\sum_{t=1}^{N} \log P(x_t|\boldsymbol{x}_{<t}), \tag{1}$$

where $P(x_1|\boldsymbol{x}_{<1}) \triangleq P(x_1)$ is the unconditional probability estimation of the first token. The three most popular families of decoder-only models are GPT, PaLM, and LLaMA. The GPT family, developed by OpenAI, includes models such as GPT-2 [233], GPT-3 [27], ChatGPT [217], and GPT-4 [2]. Notably, GPT-3 was the first LLM to exhibit emergent abilities not found in smaller PLMs. Another notable family, PaLM (Pathways Language Model), developed by Google, is comparable to the GPT family [50, 7]. While both GPT and PaLM families are closed-source, LLaMA, released by Meta, is currently the most popular open-source family of LLMs [279, 280]. The weights of these models are made available to the research community under non-commercial licenses.

Masked language modeling (MLM) task serves as a common pre-training objective for encoder-only models like BERT [67, 171]. In MLM, certain tokens in the input sequence are masked, denoted as $m(\boldsymbol{x})$, and the unmasked parts $\boldsymbol{x}_{\setminus m(\boldsymbol{x})}$ are utilized to predict the masked portions. Similar to traditional LM, the overarching goal of MLM is to minimize the negative log-likelihood as represented by the equation:

$$\mathcal{L}_{\mathrm{MLM}}(\boldsymbol{x}) \triangleq -\sum_{\widehat{x} \in m(\boldsymbol{x})} \log P(\widehat{x}|\boldsymbol{x}_{\setminus m(\boldsymbol{x})}). \tag{2}$$

Some encoder-decoder architecture models, such as T5 [235], also utilize Sequence-to-Sequence MLM task as the pre-training objective. They take masked sentences as encoder inputs and utilize the decoder to sequentially predict the masked tokens.

2.1.2 Adaptation of LLMs

After pre-training, LLMs need to be effectively adapted to better serve downstream tasks. A series of adaptation methods have been proposed for specific objectives. Due to the fact that LLMs primarily focus on generating linguistically coherent text during pre-training, their performance may not necessarily align with the actual needs of human users or conform to human values, preferences, and principles. Additionally, due to issues such as the timeliness of pre-training data, LLMs may also encounter knowledge cutoff or fallacy issues. Therefore, instruction tuning, model refinement, and model alignment have been proposed to address these issues [353, 218, 234, 60]. Below are the formal definitions of the three adaptation tasks for LLMs.

Definition 2.1 (**Instruction Tuning, IT**). *Let $h(\boldsymbol{x})$ be a language model that takes as input data $\boldsymbol{x}$, typically consisting of natural language instructions or queries. Instruction Tuning (IT) is a specialized training approach designed to enhance the model's ability to accurately and effectively respond to specific instructions. The objective of IT is to refine h by adjusting its parameters using a designated set of training examples $\mathcal{I} = \{(\boldsymbol{x}_i, \widehat{\boldsymbol{y}}_i)\}_{i=1}^{N}$, where $\widehat{\boldsymbol{y}}_i$ represents the desired output for $\boldsymbol{x}$. This set is curated to target specific tasks or functionalities that require improved performance. Formally, the updated model h' is defined as follows:*

$$h'(\boldsymbol{x}_0) = \widehat{\boldsymbol{y}}_0, \quad \forall(\boldsymbol{x}_0, \widehat{\boldsymbol{y}}_0) \in \mathcal{I}. \tag{3}$$

Definition 2.2 (**Model Refinement, MR**). *Suppose we have a model $h(\boldsymbol{x})$ taking data $\boldsymbol{x}$ (e.g., natural language queries) as inputs. Consider a size-N editing set $\mathcal{E} = \{(\boldsymbol{x}_e, \boldsymbol{y}_e, \widehat{\boldsymbol{y}}_e)\}_{e=1}^{N}$, where $\widehat{\boldsymbol{y}}_e$ denotes the true label of $\boldsymbol{x}_e$, but the model incorrectly outputs $\boldsymbol{y}_e$ for $\boldsymbol{x}_e$. Model Refinement (MR) aims to efficiently update the model from h to h' such that it correctly predicts the editing set $\mathcal{E}$, while preserving the original outputs outside $\mathcal{E}$. Formally, we aims to find h' satisfying*

$$h'(\boldsymbol{x}_0) = \begin{cases} \widehat{\boldsymbol{y}}_0 & \text{if } (\boldsymbol{x}_0, \widehat{\boldsymbol{y}}_0) \in \mathcal{E}, \\ h(\boldsymbol{x}_0) & o.w. \end{cases} \tag{4}$$

Definition 2.3 (**Model Alignment, MA**). *Consider a model $h(\boldsymbol{x})$ designed to process inputs $\boldsymbol{x}$ in decision-making scenarios. Define an alignment dataset of size M as $\mathcal{A} = \{(\boldsymbol{x}_a, \boldsymbol{y}_a, \widehat{\boldsymbol{y}}_a)\}_{a=1}^{M}$, where $\boldsymbol{y}_a$ represents the model's original decision for input $\boldsymbol{x}_a$, and $\widehat{\boldsymbol{y}}_a$ denotes the aligned decision that adheres to specified ethical guidelines or desired outcomes. The objective of Model Alignment (MA) is to modify h into h' such that for any $\boldsymbol{x}_a$ in the alignment dataset, $h'(\boldsymbol{x}_a)$ yields $\widehat{\boldsymbol{y}}_a$, aligning the model's decisions with the alignment criteria. Formally,*

$$h'(\boldsymbol{x}_0) = \widehat{\boldsymbol{y}}_0, \quad \forall (\boldsymbol{x}_0, \widehat{\boldsymbol{y}}_0) \in \mathcal{A}. \tag{5}$$

Remark. *It is still an open problem to include the constraint of preventing catastrophic forgetting of the general knowledge for IT, and reducing the Alignment Tax [165] in the optimization objective of MA. A simple extension from the constraint of model refinement in Eqn. 4, $h'(\boldsymbol{x}_0) = h(\boldsymbol{x}_0), \forall (\boldsymbol{x}_0, \widehat{\boldsymbol{y}}_0) \notin \mathcal{A}$, might be too strong in this case, as we certainly want the preference represented by $\mathcal{A}$ can generalize to other similar while not the same inputs.*

2.2 Continual Learning

Contemporary machine learning models differ from human learning processes. Humans gradually accumulate knowledge and skills across tasks without significant performance decline on previous tasks [194, 128, 219, 328, 54, 216, 170, 193]. In contrast, machine learning models are usually data-centric, minimizing the training loss on the subsequent tasks will cause the model fail on the old ones, which phenomenon is phrased as *"catastrophic forgetting"*. Addressing this challenge is a focal point in continual learning research. The problem of efficiently adapting models on a continuous sequence of tasks without forgetting is extensively studied in the continual learning community [223, 48, 282, 288]. These studies are conducted under the famous memory constraint of continual learning, as shown below.

Definition 2.4 (**Memory Constraint of Continual Learning**). *Suppose T sets of observations $\{S_t \sim \mathcal{T}_t\}_{t=1}^{T}$ come in as a sequence, where $\{\mathcal{T}_t\}_{t=1}^{T}$ denotes the T task distributions . At the learning stage of $t > 1$, the sets of observations $\{S_i\}_{i=1}^{t-1}$ are not accessible (**strong**) or partially accessible (**relaxed**).*

Remark. *In early stages of continual learning, works mostly focused on the strong memory constraint [140, 161, 4, 173]; as the research field progresses, more focus was put on relaxing the memory constraint to a small buffer for replay [239, 38, 29, 260]; some modern continual learning works consider the scenario where this constraint is completely discarded but the constraint on the computational budget is present [31, 228, 283].*

2.2.1 Three Types of Continual Learning

There are three outstanding types of continual learning scenarios: task-incremental learning (TIL), domain-incremental learning (DIL), and class-incremental learning (CIL). To establish a groundwork for subsequent discussions (as illustrated in Table 3 and Section 6.2), we adhere to the conceptual framework proposed by [282, 139, 288] and offer formal definitions for these three continual learning scenarios.

Definition 2.5 (**Task-Incremental Learning, TIL**). *Suppose T task distributions $\{\mathcal{T}_t\}_{t=1}^{T}$ come in as a sequence, where $\mathcal{T}_t$ denotes the joint distribution over the t-th task's input space and the label space $(\mathcal{X}_t, \mathcal{Y}_t)$. Denote $\mathcal{X} \triangleq \bigcup_{t=1}^{T} \mathcal{X}_t$ and $\mathcal{Y} \triangleq \bigcup_{t=1}^{T} \mathcal{Y}_t$ as the union of the input and label spaces, respectively. Under the memory constraint defined in Definition 2.4, Task-Incremental Learning (TIL) aims to find the optimal hypothesis $h^* : \mathcal{X} \times [T] \to \mathcal{Y}$ that satisfies:*

$$h^* = \arg\min_{h} \sum_{t=1}^{T} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{T}_t} \left[\mathbb{1}_{h(\boldsymbol{x},t)\neq y} \right]. \tag{6}$$

Definition 2.6 (**Domain-Incremental Learning, DIL**). *Suppose T domain distributions $\{\mathcal{D}_t\}_{t=1}^{T}$ come in as a sequence, where $\mathcal{D}_t$ denotes the t-th joint distribution over the shared input space and label space $(\mathcal{X}, \mathcal{Y})$. Under the memory constraint defined in Definition 2.4, Domain-Incremental Learning (DIL) aims to find the optimal hypothesis $h^* : \mathcal{X} \to \mathcal{Y}$ that satisfies:*

$$h^* = \arg\min_{h} \sum_{t=1}^{T} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_t} \left[\mathbb{1}_{h(\boldsymbol{x})\neq y} \right]. \tag{7}$$

Definition 2.7 (Class-Incremental Learning, CIL). *Suppose T task distributions $\{\mathcal{T}_t\}_{t=1}^T$ come in as a sequence, where $\mathcal{T}_t$ denotes the joint distribution over the t-th task's input space and the label space $(\mathcal{X}_t, \mathcal{Y}_t)$. Denote $\mathcal{X} \triangleq \bigcup_{t=1}^T \mathcal{X}_t$ and $\mathcal{Y} \triangleq \bigcup_{t=1}^T \mathcal{Y}_t$ as the union of the input and label spaces, respectively. Under the memory constraint defined in Definition 2.4, Class-Incremental Learning (CIL) aims to find the optimal hypothesis $h^* : \mathcal{X} \to [T] \times \mathcal{Y}$ that satisfies:*

$$h^* = \arg\min_h \sum_{t=1}^{T} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{T}_t} \left[\mathbb{1}_{h(\boldsymbol{x})\neq(t,y)}\right]. \tag{8}$$

Remark. *In TIL, it is common to have a shared input space $\mathcal{X} = \mathcal{X}_t, \forall t \in [T]$, but the space of the label distribution $\mathcal{Y}_t$ can be distinct ($\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset, \forall i \neq j$), partially shared ($\mathcal{Y}_i \cap \mathcal{Y}_j \neq \emptyset, \exists i \neq j$), or shared across different tasks ($\mathcal{Y} = \mathcal{Y}_t, \forall t \in [T]$). In DIL, the tasks are defined in the same format, i.e., same input space $\mathcal{X}$ and same output space $\mathcal{Y}$. During the inference, no task IDs are provided for the hypothesis, which means the continual learning model needs to capture the pattern between the domain-invariant features and the labels. DIL is commonly perceived as more difficult than TIL. CIL is commonly viewed as the most challenging continual learning scenario, as the model needs to infer the label and the task ID at the same time. Another possible formulation of CIL is to represent it as DIL but the output label spaces are disjoint, $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset, \forall i \neq j$.*

2.2.2 Techniques of Continual Learning

As outlined in the three definitions provided earlier, the objective of continual learning is to find a hypothesis that minimizes risk across all tasks/domains. Consider domain-incremental learning as an example [260], at t-th learning stage, the ideal training objective $\mathcal{L}(h)$ is

$$\mathcal{L}(h) \triangleq \underbrace{\sum_{i=1}^{t-1} \mathcal{L}_{\mathcal{D}_i}(h)}_{\text{past domains}} + \underbrace{\mathcal{L}_{\mathcal{D}_t}(h)}_{\text{current domain}}. \tag{9}$$

The objectives for past domains are often challenging to measure or optimize due to the memory constraints (Definition 2.4). Therefore, the core of designing continual learning algorithms lies in identifying a suitable proxy learning objective for the first term without violating the memory constraint. Existing continual learning techniques can be roughly categorized into 5 groups: (i) replay-based, (ii) regularization-based, (iii) architecture-based, (iv) optimization-based, and (v) architecture-based [61, 288]. Here, we will provide a concise yet comprehensive introduction to the first three categories of continual learning techniques, as they find extensive application in continually learning large language models.

Replay-Based Methods. Replay-based continual learning methods adopt the relaxed memory constraint by keeping a small buffer $\{M_i\}_{i=1}^{t-1}$ for each task $\mathcal{T}_i$ to retain previously observed data examples. Formally, these methods seek to optimize the following empirical training objective:

$$\widehat{\mathcal{L}}_{\text{replay}}(h) \triangleq \underbrace{\sum_{i=1}^{t-1} \widehat{\mathcal{L}}_{M_i}(h)}_{\text{proxy for past domains}} + \underbrace{\widehat{\mathcal{L}}_{S_t}(h)}_{\text{current domain}}, \tag{10}$$

where $\widehat{\mathcal{L}}_S$ denotes the empirical loss term evaluated on the set of examples S. Often regarded as a simplistic solution to continual learning, replay-based methods may theoretically lead to loose generalization bounds [260]. Despite this, they are valued for their simplicity, stability, and high performance, even with a small episodic memory [38, 240]. For instance, DER++ [29] demonstrates consistent performance enhancement by replaying a small set of past examples along with their logits (known as dark experience replay). ESM-ER [250] introduces error sensitivity modulation (ESM) to mitigate abrupt representational drift caused by high-error new examples. A significant focus in replay-based continual learning research is enhancing sample efficiency for buffer maintenance. For instance, [239] prioritizes exemplar selection based on herding to accurately model class mean throughout class-incremental learning. [361] propose storing low-fidelity examples to achieve memory-efficient exemplar set maintenance. RM (Rainbow Memory) [15] introduces diversity-aware memory updates based on per-sample uncertainty estimation and data augmentation for class-incremental learning.

Regularization-Based Methods. Suppose $h_{\boldsymbol{\theta}_{t-1}}$ is the hypothesis yielded after the $t-1$-th stage of training, parameterized by $\boldsymbol{\theta}_{t-1}$. Regularization-based methods utilize a regularization term as a proxy for past domain losses, determined by the distance in the parameter space.

$$\widehat{\mathcal{L}}_{\text{reg}}(h_{\boldsymbol{\theta}}) \triangleq \underbrace{\lambda \cdot \|\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}\|_{\boldsymbol{\Sigma}}}_{\text{proxy for past domains}} + \underbrace{\widehat{\mathcal{L}}_{S_t}(h_{\boldsymbol{\theta}})}_{\text{current domain}} , \tag{11}$$

where $\|\boldsymbol{v}\|_{\boldsymbol{\Sigma}} = \boldsymbol{v}^\top \boldsymbol{\Sigma} \boldsymbol{v}$ is the vector norm evaluated on a positive-semi-definite matrix $\boldsymbol{\Sigma}$, and λ is the regularization coefficient, a hyper-parameter introduced to balance the past knowledge retention and current knowledge learning. The matrix $\boldsymbol{\Sigma}$ 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. For example, Elastic Weight Consolidation (EWC) adopts a Bayesian perspective, using diagonal values from the Fisher Information Matrix (FIM) as an approximation for the Hessian matrix of parameters. This forms a sequential Maximize A Posteriori (MAP) optimization for continual learning [140]. Memory Aware Synapses (MAS) computes parameter importance in an online and unsupervised manner, defining importance by accumulated absolute gradient during training [4]. It is also worth noting that when $\boldsymbol{\Sigma} = \boldsymbol{I}$ degenerates to an identity matrix, the regularization term simplifies to a basic l_2-penalty term, evenly penalizing each parameter, which can be surprisingly effective in some cases of continuall learning LLMs [243].

Architecture-Based Methods. Expanding the network architecture dynamically to assimilate new knowledge is deemed the most efficient form of continual learning [300, 299]. This method primarily tackles adaptation challenges and can achieve zero-forgetting when task IDs are available during inference or can be correctly inferred [91, 308]. However, due to the difficulty of task ID inference, architecture expansion is predominantly utilized in task-incremental learning but is scarcely explored in domain-incremental or class-incremental learning. Progressive Neural Networks (PNN) proposes learning laterally connected neurons as new tasks arise, ensuring non-forgetting and enabling transfer of previously learned neurons for future tasks [247]. In conjunction with pre-trained backbone large models like ViT [71], CoLoR [308] trains various low-rank adaptation (LoRA) 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 continually learning LLMs, architecture expansion has resurged in popularity following the rise of parameter-efficient fine-tuning (PEFT) applied to large models [259, 5, 109, 66, 146, 159], a topic we will delve into shortly [330, 291, 149, 120, 127, 221, 327, 310].

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 [233, 27, 217, 2, 50, 7, 279, 280]. The substantial scale of LLMs presents notable challenges for development teams, particularly in keeping them updated amidst rapid environmental changes [6, 127, 68, 120, 119]. To illustrate, in 2023, the average daily influx of new tweets posted by users exceeds 500 million[1], and training on even a "small" subset of this large volume of data is not affordable. Efficiently and reliably adapting LLMs becomes more critical when considering their cascading impact on downstream applications. Downstream users often lack expertise in collecting and storing large-scale data, maintaining large-scale hardware systems, and training LLMs themselves. Recyclable Tuning [231] is the pioneering study that explicitly outlines the supplier-consumer structure of 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 needs to utilize the stronger and more up-to-date upstream model for better downstream performance. To enhance the efficiency of fine-tuning for downstream consumers, they initially make several key observations about continually pre-trained LLMs, focusing on mode connectivity and functional similarity. Additionally, they propose reusing the outdated fine-tuned components after a major update of the upstream pre-trained LLM. Building upon the conceptual framework introduced by Recyclable Tuning [231], we present a comprehensive framework for a

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

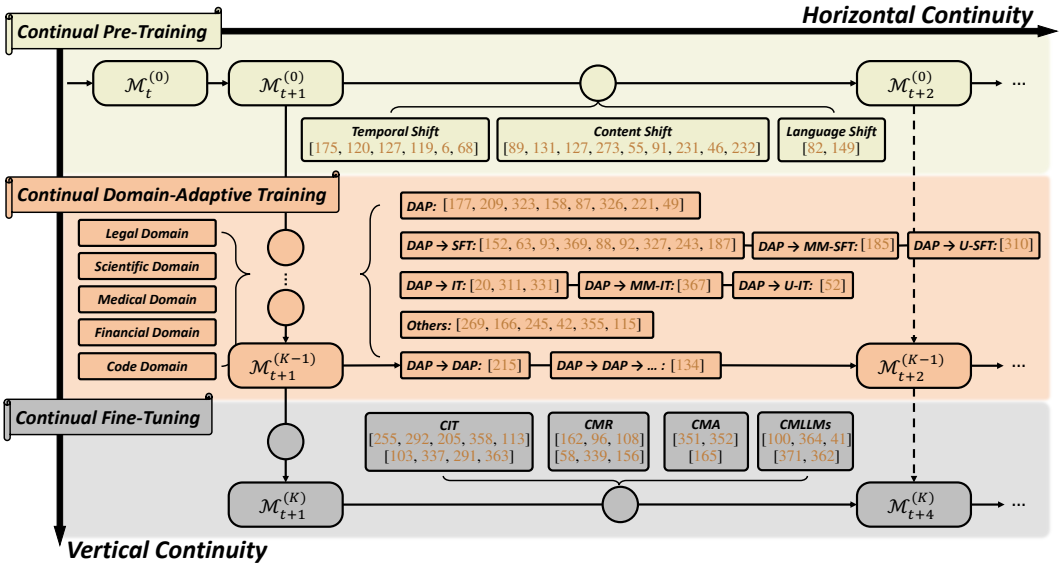

Figure 1: A high-level overview of the modern pipeline for continually pre-training and fine-tuning LLMs, where two dimensions of continuity are described. **Vertical Continuity (or Vertical Continual Learning)**: LLM training can be vertically divided into three stages: (i) Continual Pre-Training (CPT), (ii) Domain-Adaptive Pre-training (DAP), and (iii) Continual Fine-Tuning (CFT). Along the vertical axis, scale of data, scope of tasks, and computational resources, gradually decreases, while the specificity of the LLM is improved towards the final downstream task's solution. The main focus of vertical continuity 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 samples becomes available. The primary goal of horizontal continuity is to prevent horizontal forgetting in a long sequence of tasks.

modern production pipeline encompassing various studies on continual LLM pre-training, adaptation, and deployment, illustrated in Fig. 1. What sets our framework in this survey apart from existing studies [314] 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; it involves a sequence of adaptation from general to specific domains and tasks [91, 243, 88, 327, 323]. Along this axis, the training task transitions gradually from general pre-training to downstream tasks, typically undertaken by distinct entities within the production pipeline [231]. Vertical continuity is characterized by a hierarchical structure encompassing data inclusiveness, task scope, and computational resources. Fig. 1 shows a typical pipeline for vertical continuity in LLMs, i.e., "pre-training" → "domain-adaptive training" → "downstream fine-tuning" [185, 152, 63, 93, 369, 88, 92, 52, 311, 310, 327, 243, 187, 115]:

- **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.

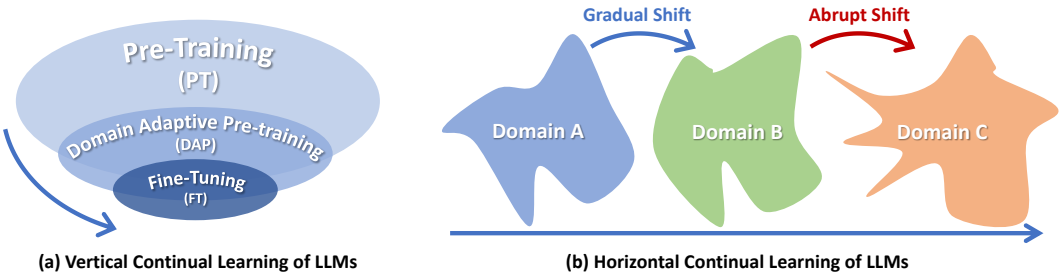

Figure 2: A diagram showing two different directions of continual learning of LLMs. **(a) Vertical Continual Learning of LLMs**: in this case, the upstream data distribution usually partially covers the subsequent tasks' data distribution. **(b) Horizontal Continual Learning of LLMs**: No constraints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

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 stages [215, 166, 245, 115]. 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 on general knowledge of a model undergoing vertical continual learning *"vertical forgetting"*. As shown in Fig. 2, usually 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. However, there are two significant challenges to 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 [239, 161, 316, 212, 139]. To mitigate this issue, practitioners often employ methodologies like freezing shared parameters during downstream phases or reformulating downstream tasks to match the structure of pre-training tasks [330, 291, 149, 221, 327, 310].
- **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 2.4), as algorithms for latter case rely on access to previous data at specific points for parameter importance measurement [140, 4] or for replay [240, 38, 29, 260]. To address the challenge of *inaccessible upstream data*, existing methods either use public datasets or generate pseudo-examples to create proxy pre-training dataset [230].

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 [37, 6, 271, 323]. 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 [6, 120, 119, 68]. 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 [274, 183, 365, 195]. The necessity of employing complex continual learning algorithms to address challenges in LLMs remains an open question. For instance, during large-scale continual pre-training, major institutions can typically afford the storage costs associated with retaining all historical data, rendering memory constraints negligible. Several studies have

demonstrated that with full access to historical data, simple sparse replay techniques can effectively mitigate forgetting in large models [277, 274, 255, 228, 81]. In contrast, numerous continual learning studies have showcased superior performance compared to naive solutions, suggesting the importance of continual learning techniques in LLM training [119, 127, 232, 46].

Horizontal Forgetting. We informally define *"horizontal forgetting"* as the degradation in performance on previous tasks during horizontal continual learning. 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, addressing horizontal forgetting presents two main challenges:

- **Longer Task Sequence.** 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 [237].
- **Abrupt Distributional Shift.** In contrast to vertical continuity, where distributional shifts are often predictable, horizontal continual learning does not impose constraints on the sequential learning tasks' properties. Evidence suggests that abrupt changes in task distributions can result in significant horizontal forgetting of the model [30, 250].

4 Learning Stages of Continual Large Language Models

Fig. 1 provides an overview of continually learning large language models. Along the axis of vertical continuity, three major 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). As data volume increases, the general capacity of LLMs naturally evolves. 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). Within continual fine-tuning, we further cover topics including continual instruction tuning (Section 4.3.3), model refinement (Section 4.3.4), model alignment (Section 4.3.5), and multimodal LLMs (Section 4.3.6).

4.1 Continual Pre-Training (CPT)

The recent development of large language models has shattered the glass ceiling in achieving close-to-human levels of natural language understanding and generation. However, effectively adapting these models to the ever-evolving environment remains a fundamental challenge. In Table 1, we outline the basic properties of existing CPT papers.

4.1.1 CPT: Effectiveness and Efficiency

Before delving into the detailed introduction of papers on continual pre-training (CPT), it is important to address two fundamental questions: Firstly, regarding *effectiveness*, can CPT enhance performance on downstream tasks beyond that of the initial training on a wide range of data domains? Extensive studies, including ELLE [232], DEMix [91], CKL [120], TemporalWiki [119], LLPT [127], and Lifelong-MoE [46], have not only demonstrated the necessity of CPT for improved downstream performance, but also shown that when distributional shifts are gradual [119] or somewhat correlated [91], CPT can enhance model generalization to unseen data.

After confirming the effectiveness of CPT, the second question regarding *efficiency* arises: given the large number of parameters in the LLM and the size of both old and new data, achieving adaptation and knowledge retention in a computationally efficient manner becomes crucial. Concerning efficiency, most studies focus on techniques for efficient knowledge retention [127, 120, 119, 149], which significantly overlap with the continual learning literature addressing catastrophic forgetting. As mentioned before, these techniques replay [254, 240, 29, 260], parameter regularization [239, 241, 4], and architecture expansion [247, 237, 287]. In contrast to prior approaches that fully utilize emergent

Table 1: **Summary of the existing studies on Horizontal Continual Pre-training of LLMs,** where the papers are organized based on their type, where: (i) no continual learning techniques are studied, (ii) continual learning techniques are studied as solely baselines, and (iii) new approaches are proposed, containing some of the continual learning techniques. 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; and use "✓*" to represent the vocabulary expansion and replacement. It is noteworthy 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 "♣" [127, 119, 120] means that merely existing CL techniques are studied in them, and the papers with only "✗" [89, 82] means that no CL techniques but special aspects of fine-tuning are studied, e.g., model (re)warming via learning rate scheduling [89].

Method	Scenario		Continual Learning Tech.			LLM Arch.	Evaluation	
	Dist. Shift	*#Domains*	*Rehearsal*	*Param. Reg.*	*Arch. Exp.*		*Pre-Training*	*Downstream*
TimeLMs [175]	Temporal	8	✗	✗	✗	RoBERTa	✓	✓
[89]	Content	1	✗	✗	✗	Pythia	✓	✗
[82]	Language	3	✗	✗	✗	GPT	✓	✗
[149]	Language	1	✗	P-Freeze♣	Adapter♣ LoRA♣	Llama2	✓	✓
CKL [120]	Temporal	1	Mix-Review♣	P-Freeze♣ RecAdam♣	LoRA♣ K-Adapter♣	T5	✗	✓
LLPT [127]	Temporal / Content	4 / 8	ER♣ Logit-KD♣ Rep-KD♣ Contrast-KD♣ SEED-KD♣	oEWC♣	Adapter♣ Layer Exp.♣	RoBERTa	✓	✓
TemporalWiki [119]	Temporal	5	Mix-Review♣	P-Freeze♣ RecAdam♣	LoRA♣ K-Adapter♣	GPT-2	✓	✓
CPT [131]	Content	4	DER++♣ KD♣	CPT✓ EWC♣ HAT♣	Adapter♣ DEMix♣	RoBERTa	✓	✗
ERNIE 2.0 [273]	Content	4	ER✓♣	✗	✗	ERNIE	✗	✓
[6]	Temporal	7	✗	P-Freeze✓	Vocab. Exp.✓	BERT	✗	✓
[55]	Content	5	✗	✗	Vocab. Exp.✓	BERT RoBERTa	✗	✓
DEMix [91]	Content	8	✗	✗	MoE✓	GPT-3	✓	✓
TempoT5 [68]	Temporal	1	✗	✗	Vocab. Exp.✓ Prompt✓	T5	✗	✓
RecTuning [231]	Content	4	ER✓ KD✓	✗	Adapter✓	RoBERTa	✗	✓
Lifelong-MoE [46]	Content	3	ER♣ KD✓	P-Freeze✓ L2♣	MoE✓	GLaM	✓	✓
ELLE [232]	Content	5	ER✓♣ KD♣	P-Freeze✓	Prompt✓ Layer Exp.✓ Adapter♣	BERT GPT	✓	✓

data, some studies recognize the impracticality of this approach in real production environments. Instead, they concentrate on further improving the efficiency of adapting to new distributions. For instance, ELLE [232] employs a function-preserved model expansion to facilitate efficient knowledge growth; [6] sub-samples training data based on semantic shift levels to enhance training efficiency; [323] employs a data sampling strategy that encourages novelty and diversity, achieving superior performance to full-data training. Though underexplored, this aspect of efficient adaptation in continual pre-training is poised to become significant, given recent findings emphasizing data quality over quantity for LLM generalization [72, 157, 321, 267].

4.1.2 General Observations on CPT

The analysis presented in Table 1 sheds light on the prevailing research trends in continual pre-training (CPT). Firstly, it is evident that ***the development of advanced techniques tailored specifically for CPT is still at the starting stage and warrants further exploration.*** This observation is underscored

by the fact that only about half of the examined papers propose novel techniques (9 out of 16 papers, represented in the deep gray section of Table 1), while the remaining half either focus solely on the effects of pure adaptation without considering continual learning techniques (3 out of 16 papers, represented in the white section), or conduct empirical studies on the straightforward application of existing continual learning techniques (4 out of 16 papers, represented in the light gray section). Secondly, while research extensively covers various continual learning techniques, such as rehearsal, parameter regularization, and architecture expansion (as indicated in the light gray section of Table 1), ***the practical incorporation of these techniques in systems remains relatively limited***. Most practical implementations primarily focus on architecture expansion of LLMs [6, 55, 91, 68, 231, 46], with only a few explicitly utilizing replay [231, 46] and parameter regularization [6, 46] explicitly (deep gray section of Table 1). Thirdly, ***there is a pressing need for exploration into longer sequences of incremental phases in continual pre-training.*** Currently, the longest sequence of domains explored is eight, with content-level distributional shifts [127, 91]. However, this falls short of real-world scenarios where continual pre-training may occur more frequently and persist for months or years. The efficacy of continual learning techniques in such prolonged scenarios remains uncertain, as potential performance degradation with longer domain sequences is observed in techniques like EWC [140]. Additionally, investigating CPT in a task-boundary-free data stream setting is an important avenue for research as well.

4.1.3 Distributional Shifts in CPT

This survey categorizes distributional shifts of continual pre-training into three main types: (i) *Language Shift*: LLMs sequentially learn different language corpora, e.g., English $\rightarrow$ Chinese, focusing on token and vocabulary distributional shifts [82, 149]. (ii) *Content Shift*: LLMs sequentially learn corpora from different fields, e.g., chemistry $\rightarrow$ biology, focusing on token and vocabulary distributional shifts as well as shift of semantic meaning [91, 55, 127, 231, 46, 89]. (iii) *Temporal Shift*: Distributional shifts occur over time, e.g., news in 2021 $\rightarrow$ news in 2022, focusing on token and vocabulary shifts, and timestamp-sensitive knowledge retention and update, which aligns with real-world LLM deployment needs [6, 127, 68, 120, 119].

Language Shift. In contrast to the common approach of pre-training multilingual language models jointly on large corpora from multiple languages, [82] focuses on assessing these models' natural ability to learn new languages sequentially (English, Norwegian, and Icelandic). The study does not employ explicit continual learning techniques for preventing horizontal forgetting. Nevertheless, it observes consistent positive forward transfer, facilitating new language acquisition regardless of the presentation order. Forgetting, on the other hand, emerges as a significant challenge, influenced by language order and not mitigated by increasing LLM size. In [149], the degree of forgetting of previously learned language (English) when adapting LLMs to a new language (Traditional Chinese) is investigated. Various continual learning techniques, including parameter freezing, LoRA [109], and $(IA)^3$ [168], are evaluated across multiple dimensions, including output language, general knowledge retention, and reliability. Preliminary experimental results presented in this study highlight the non-trivial nature of addressing horizontal forgetting in continually pre-training LLMs under the language shift. To summarize, research on continual pre-training for language shifts is in its preliminary stages for two main reasons: Firstly, the datasets' scale, including the number of languages and total token count, remains small. Secondly, specific methods targeting language shifts have yet to be proposed; only basic combinations of existing continual learning techniques have been evaluated.

Content Shift. Without using complex CL techniques, [89] continues the pre-training phase of Pythia [21] on the newly collected SlimPajama dataset [266]. The study focuses on optimizing continual pre-training by learning rate (re)warm-up. They discover that regardless of whether a larger or smaller maximum learning rate is used, models that undergo re-warming consistently exhibit improvements over models trained from scratch, even in terms of adaptation solely.

Another pioneering work, LLPT [127], establishes a comprehensive training and evaluation protocol for a series of content-level distributional shifts, referred to as "domain-incremental data streams" in the paper. They assess multiple continual learning methods based on masked language modeling perplexity for pre-training tasks and downstream task accuracy. Similar to findings in [82], they note that later domains benefit from knowledge learned from earlier ones, yet horizontal forgetting remains a significant challenge for earlier domains. Contrary to the common belief that experience replay (ER, [38]) is the most efficient approach to preventing forgetting, the authors find it scarcely improves continual pre-training performance. They speculate that ER's inefficiency may stem from overfitting

issues, as replaying with distillation loss can alleviate this problem efficiently [340, 127]. Following LLPT, Recyclable Tuning [231] is the first study to consider both upstream LLM suppliers and downstream consumers at the same time. It shows that if the upstream supplier continually pre-trains LLMs – initializing from the previous checkpoint and continuing pre-training on newly collected data, with or without replay, consumer-side efficiency can be boosted by recycling previously learned incremental components. Two CL techniques, initializing from outdated components and knowledge distillation, complement each other to improve recyclable tuning in this context.

Other approaches involve training additional domain-specific experts for new content domains. DEMix [91] addresses continual pre-training by incrementally training and integrating new experts (DEMix layer replacing every FFN layer in the transformer) for new domains. To ensure reasonable inference performance during testing when no domain information is available, DEMix proposes a parameter-free probabilistic approach, distinct from the gating function in MoE [259], to dynamically estimate a weighted mixture of domains. Introducing a new domain variable D_t alongside each word x_t, the authors estimate the next word probability $p(x_t|\boldsymbol{x}_{<t})$ by marginalizing over all experts[2]:

$$\begin{aligned} p(x_t|\boldsymbol{x}_{<t}) &= \sum_{j=1}^{n} p(x_t|\boldsymbol{x}_{<t}, D_t = j) \cdot p(D_t = j|\boldsymbol{x}_{<t}) \\ &= \sum_{j=1}^{n} p(x_t|\boldsymbol{x}_{<t}, D_t = j) \cdot \left[\frac{p(\boldsymbol{x}_{<t}|D_t=j) \cdot p(D_t=j)}{\sum_{j'=1}^{n} p(\boldsymbol{x}_{<t}|D_t=j') \cdot p(D_t=j')} \right], \end{aligned} \tag{12}$$

where all the probability terms $p(\cdot|\cdot, D_t)$ conditioned on the domain variable D_t are calculated by using a specific domain expert. The authors develop a large-scale continual pre-training evaluation benchmark comprising eight semantic domains for sequential training and another set of eight domains for assessing LLMs' generalization ability. The DEMix framework's modularization has been shown to facilitate efficient domain-adaptive pre-training, promote relevant knowledge during inference, and allow for removable components. Lifelong-MoE [46] shares a similar approach to DEMix [91] by incrementally training domain experts for each new domain. However, Lifelong-MoE differs 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, while knowledge distillation loss is employed to regulate parameter updates. Although the data distributions for evaluation are extremely large-scale (3 domains, 686 billion tokens in total), the Lifelong-MoE is able to efficiently mitigate the issue of horizontal forgetting.

It is noteworthy that some papers draw almost opposite conclusions regarding the significance of CPT. For instance, [55] continually pre-trains BERT [67] and RoBERTa [171] on five scientific domains and evaluates performance on downstream sentiment analysis. They observe that even baseline sequential pre-training does not exhibit severe forgetting, prompting reasonable questions about the necessity of continual pre-training or its suitable application scenarios.

Temporal Shift. In the context of continual learning amid content shifts, Multi-Task Learning (MTL) is often regarded as the upper limit achievable in continual learning scenarios [223, 288, 260]. However, this belief does not fully hold when considering continual learning under temporal shifts [120, 119, 68], 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 [120] and TemporalWiki [119], LLMs undergoing continual adaptation to temporal shifts in the corpus must simultaneously achieve three objectives: (i) retention of old knowledge, (ii) acquisition of new knowledge, and (iii) update of the outdated knowledge (as a conflict resolution). They evaluate the same set of continual learning baseline methods, including parameter regularization (RecAdam [44]), rehearsal (Mix-review [102]), and parameter expansion (LoRA [110] and K-Adapter [290]), each highlighting distinct aspects of their impact. CKL [120] introduces a unified metric, **FUAR** (**F**orgetting / (**U**pdated + **A**cquired) **R**atio), to assess the three learning objectives collectively. They observe 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

[2]In the marginalization step of the following equation in the original paper, it is $p(D_t = j|\boldsymbol{x}_t)$ instead of $p(D_t = j|\boldsymbol{x}_{<t})$, while we believe this is a minor typo, and hence here we use the updated version.

newly learned information during training. TemporalWiki [119], in contrast, constructs a series of temporal corpora and their differential sets from sequential snapshots of Wikipedia, investigating the efficacy of adapting LLMs to these differential sets. The study reveals that updating LLMs on these differential sets substantially enhances new knowledge acquisition and updates, requiring significantly less computational resources. Moreover, various continual learning techniques prove effective in mitigating horizontal forgetting during this process.

Additionally, LLPT [127] 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 any continual learning technique, over a single task-specific LM, in terms of both knowledge acquisition and temporal generalization. Nonetheless, these preliminary experiments do not conclusively determine which specific continual learning 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 learning process [244, 68, 271]. During training, they inject temporal information into training examples as prefixes of prompts, using special tokens [244], explicit year information [68], or syntax-guided structural information [271]. In sequential training experiments conducted by TempoT5 [68], 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.

The significance of addressing temporal shifts through continual pre-training is underscored by several industrial studies. For instance, [6] employs a dynamic vocabulary expansion algorithm and an efficient sub-sampling procedure to conduct CPT on large-scale emerging tweet data. Conversely, [175] adopts continual pre-training 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 [9], the community commonly embraces CPT as a more efficient and environmentally friendly 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" [327, 88, 187, 93, 323, 319, 115, 177, 320, 10, 344, 52, 350], "further pre-training" [269, 166, 63, 246, 3], "domain tuning" [243], "knowledge enhancement pre-training" [177], and "knowledge injection training" [311] is unified and termed "***Domain Adaptive Pre-training (DAP)***" [92] for clarity and consistency throughout the remainder of this survey. In the pioneering work of domain-adaptive pre-training (DAPT) [92], 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 [92], 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. To illustrate this, we provide a summary of existing studies utilizing domain-adaptive pre-training for LLMs in Table 2. Each entry in the table 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 continual learning, such as backward transfer (forgetting) and forward transfer (adaptation to downstream data). Following this overview, we will present general observations about DAP in Section 4.2.1, followed by a detailed introduction to LLMs developed in vertical domains in Section 4.2.2.

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 [166, 115], and the prefix U- represents them unified [310, 42]. The prefix MM- and LC- represents Multi-Modal and Long-Context training phases [185, 367, 245]. 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. Among 33 papers presented in this table that adopt DAP during the development, nearly 65% (22/33) of them explicitly study the influence of DAP from a continual learning perspective: they either evaluate the degree of forgetting, or adopt the continual learning techniques to prevent forgetting of the general knowledge. However, there is a significant lack of diversity of the continual learning techniques adopted in these works (only Replay and LoRA), which advocates the further study on the efficacy of vertical continual learning in the realm of LLMs.

Domain	**Method**	**Train Proc.**	**LLM Arch.**	**Continual Learning Tech.**			**Continual Learning Eval.**	
				Rehearsal	*Param. Reg.*	*Arch. Exp.*	*Backward Transfer*	*Forward Transfer*
Medical	BioMedGPT [185]	DAP → MM-SFT	Llama2	✗	✗	✗	✗	FT
Financial	BBT-Fin [177]	DAP	T5	✗	✗	✗	✗	FT
Financial	CFGPT [152]	DAP → SFT	InternLM	✗	✗	Q-LoRA$_{(SFT)}$	✗	HE[1]
Scientific	AstroLlama [209]	DAP	LlaVa	✗	✗	✗	✗	Perp.
Scientific	OceanGPT [20]	DAP → IT	Vicuna Llama2-chat ChatGLM2	✗	✗	LoRA$_{(IT)}$	✗	HE
Scientific	K2 [63]	DAP → SFT	Llama	✗	✗	LoRA$_{(SFT)}$	✗	Perp. | ZS | LLM
Scientific	MarineGPT [367]	MM-DAP → MM-IT	Llama	✗	✗	✗	✗	HE
Code	CodeGen [215]	DAP → DAP	CodeGen	✗	✗	✗	✗	Perp. | ZS
Code	Comment-Aug [269]	IT → DAP	Llama2 Code Llama InternLM2	✗	✗	✗	✗	ZS
EventTemporal	EcoNet [93][1]	DAP → FT	BERT RoBERTa	✗	✗	✗	✗	FT
CommonSense	CALM [369]	DAP → FT	T5	✗	✗	✗	✗	FT
Medical	[88]	DAP → FT	Llama2	✗	✗	✗	FS | FT	FS | FT
Financial	[323]	DAP	Pythia	✗	✗	✗	L | FS	L | FS
Scientific	GeoGalactica [166]	DAP → G-SFT → D-SFT	GAL	✗	✗	✗	ZS	Perp. | ZS | LLM
Code	StarCoder [158]	DAP	StarCoder	✗	✗	✗	Perp. | ZS | FS	Perp. | ZS | FS
Code	DeepSeek-Coder [87]	DAP	DS-LLM	✗	✗	✗	ZS | FS	ZS
Multi-Domain	DAPT [92]	DAP → FT	RoBERTa	✗	✗	✗	Loss	L | FT
Financial	WeaverBird [326]	DAP	GLM2	✗	✗	LoRA	✗	HE
Code	IRCoder [221]	DAP	StarCoder DS-Coder Code Llama	✗	✗	LoRA	✗	ZS
Code	Code Llama [245]	DAP → LC-FT → IT DAP → DAP → LC-FT	Llama2	Replay	✗	✗	✗	Perp. | ZS
Legal	SaulLM [52]	DAP → U-IT	Mistral	Replay	✗	✗	✗	Perp. | ZS
Medical	PMC-Llama [311]	DAP → IT	Llama	Replay	✗	✗	✗	ZS | FT
Scientific	Llema [10]	DAP	Code Llama	Replay	✗	✗	✗	Perp. | FS
Multi-Domain	DAS [134]	$[DAP]_n$	RoBERTa	DER++♣	EWC♣ HAT♣ Soft-Masking	Adapter♣ DEMix♣	✗	FT
Code & Math	Llama Pro [310]	DAP → U-SFT	Llama2	✗	✗	Block Exp. LoRA♣	ZS | FS	Perp. | ZS | FS
Medical	AF Adapter [327]	DAP → FT	RoBERTa	✗	✗	Layer Exp. LoRA♣	Acc.	L | FT
Medical	[243]	DAP → FT	BERT RoBERTa DistilBERT	Replay♣ GEM♣	L2 Reg.♣ EWC♣	✗	L | FT	L | FT
Medical	HuatuoGPT-II [42]	DAP + U-SFT	Baichuan2	Replay	✗	✗	ZS	ZS | HE
Financial	XuanYuan 2.0 [355]	DAP + SFT	BLOOM	Replay	✗	✗	HE	HE
Scientific	PLlama [331]	DAP → IT	GAL	Replay	✗	✗	L	L | ZS
E-Commerce	EcomGPT-CT [187]	DAP → SFT	BLOOM	Replay	✗	✗	ZS | FS	ZS | FS
Legal	Layer Llama [115]	DAP → G-IT → D-IT	Llama	Replay	✗	✗	ZS	ZS
Multi-Domain	AdaptLLM [49]	DAP	Llama	Replay	✗	✗	ZS	ZS | FT

4.2.1 General Observation on DAP

As depicted in Table 2, several key observations emerge regarding the current research landscape of DAP. Firstly, ***DAP predominantly occurs in a single stage.*** Horizontal Continual DAP which involves more than one stage is seldom explored: among the 34 papers listed, only one paper employs two stages of DAP [245]. In Code Llama [245], aimed at developing a language model tailored to Python programming, the authors initialize the model from the pre-trained Llama 2 checkpoint. They then conduct the first stage of DAP across multiple programming languages (500 billion tokens) before proceeding to the second stage, focusing solely on Python code (100 billion tokens). Finally, they perform long context fine-tuning (20 billion tokens) to enhance the model's capability in challenging long-context scenarios of code generation. This PT $\rightarrow$ DAP $\rightarrow$ DAP $\rightarrow$ FT pipeline represents the sole example found thus far that strictly adheres to the definition and hierarchical structure of vertical continuity in pre-training and adapting LLMs for final end-use. Hence, categorizing the 10 studies that solely conduct one stage of DAP and nothing more [177, 209, 269, 323, 158, 87, 326, 221, 10, 49] proves challenging. One could also argue that they deploy an additional single stage of CPT rather than DAP. Nevertheless, considering that all these papers aim to adapt a general-purpose LLM to a specific domain, we include them in this section for discussion, aligning with the categorization we have established thus far.

Secondly, ***the notion of interpreting DAP through the lens of continual learning, whether intentional or not, is widely embraced.*** As demonstrated in Table 2, with the exception of the first section (white, 11/33), where papers overlook any potential side effects of DAP leading to vertical forgetting of previously learned general knowledge, the remaining sections (all gray, 22/33) either evaluate the potential negative impacts of DAP or proactively employ continual learning techniques to mitigate the risk of vertical forgetting from the outset.

Thirdly, we observe widespread adoption of CL techniques (14/33) for training domain-specific LLMs. However, the diversity of these techniques is limited, with only replay [52, 311, 10, 243, 42, 355, 331, 187, 115, 49] and parameter expansion (LoRA [326, 221, 310, 327]) or Layer/Block expansion [310, 327] being utilized. ***This highlights the need for further research to investigate, incorporate, and design more sophisticated CL techniques for not just DAP, but vertical continual learning in general.*** 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. This deduction arises from two observations: (i) parameter expansion methods inherently embody implicit CL techniques. For instance, in LoRA [110], the increment of weights $\Delta \boldsymbol{W} = \boldsymbol{O}$ preserves the original performance on previous data distributions, but once adaptation occurs ($\Delta \boldsymbol{W} \neq \boldsymbol{O}$), forgetting on the original data distribution follows. This analysis extends to other parameter expansion techniques such as layer expansion [327] and block expansion [310]. Authors typically empirically demonstrate the effectiveness of these approaches, attributing forgetting mitigation to the low-rank property and parameter efficiency; (ii) excluding parameter expansion methods, replay emerges as the only CL technique employed during DAP, except in cases where extensive empirical investigations of CL methods are conducted [243]. Furthermore, studies deploying replay often term the technique as "data combination" [311] or "data mixing/mixture" [10, 331, 187, 49], without recognizing it as a vertical continual learning problem.

4.2.2 Different Domains of DAP

Legal Domain. Given the legal industry's demand for managing ever-growing volumes of legal documents, there's a burgeoning need to harness LLMs to aid legal professionals in navigating, interpreting, and generating high-quality legal materials [318, 251, 343]. While general-purpose LLMs may perform adequately on some legal benchmarks [191], customizing LLMs with additional unlabeled resources specific to the legal domain can yield superior results. This is because the high-volume unlabeled legal corpus resembles the conditions under which general-purpose LLMs are pre-trained. In Layer Llama [115], the authors gathered publicly available legal texts from China Courts websites, including judgment documents, legal articles, judicial interpretations, court news, and law popularization articles, totaling approximately 10 billion tokens as noted in a GitHub issue [116]. In SaulLM [52], the authors collected the DAP corpus from various jurisdictions in different countries, such as the U.S., Europe, and Australia, resulting in a corpus of 30 billion tokens

to cover diverse aspects of legal texts. When combined with previously available datasets[3] [79, 141], the total tokens used for legal-domain DAP reach 94 billion.

The substantial volume of DAP data, while offering valuable insights into specific domains, increases the risk of catastrophic forgetting of the general knowledge due to the large number of update steps involved. To mitigate this, SaulLM incorporates general data from Wikipedia, StackExchange, and GitHub into the DAP data, constituting about 2% of the final dataset [52]. Following DAP, SaulLM then employs a combination of general-domain and legal-domain instructions to enhance the model's instruction-following ability (U-IT, see Table 2). Similarly, Lawyer Llama also incorporates general-domain data during DAP, but the replay rate is not disclosed [115]. After DAP, it undergoes two distinct phases of instruction tuning: first, general-domain instruction tuning (G-IT), followed by domain-specific downstream legal consultation application instruction tuning (D-IT). However, no explanation is provided for why two-stage IT is preferred over consolidation into one as in SaulLM [52], leaving this as an open question for future research.

Medical Domain. The development of LLMs holds promise for revolutionary changes in the medical industry, offering potential improvements in efficiency and quality across medical communication, disease diagnosis, and decision-making for doctors. While some instances of general-domain LLMs have shown success in providing useful advice and accelerating diagnosis progress for patients [160, 262], direct deployment poses risks. These risks include the potential for sub-optimal solutions, such as imprecise medical advice, and the possibility of harm, such as incorrect drug recommendations and the propagation of medical misinformation [121, 45]. Efforts have been made to develop medical specialists by either collecting medical-domain data and training an LLM from scratch [94, 263, 86, 182], or fine-tuning publicly available LLMs to meet specific medical needs [185, 311, 42, 324, 16, 354]. Among these approaches, domain-adaptive pre-training techniques have been extensively utilized to preserve the communication and instruction-following abilities of a general LLM, preparing it for subsequent medical applications [185, 311, 42].

BioMedGPT [185] 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 [280] and conduct DAP using extensive biomedical documents from S2ORC [172], without considering any continual learning techniques or metrics. In [88], a Chinese medical LLM is developed for medical question answering. DAP is performed using Chinese medical encyclopedias and online expert articles, yielding over 364,000 question-answer pairs, with next-token prediction as the sole training objective. Results show that for both models initialized from Llama-2 [280] and Chinese-Llama-2 [345], performance gradually deteriorates on general-domain datasets as training steps increase, while improving on the Chinese medical examination evaluation [105, 169, 151]. To ensure the model possesses sufficient fundamental medical knowledge before the instruction tuning phase, PMC-LLama [311] gathers biomedical papers from S2ORC [172] and medical textbooks for "knowledge injection training". During this phase, a general language corpus from RedPajama-Data [53] is replayed at a 5% rate within a training batch. However, the paper does not analyze the effectiveness of adding and mixing general-domain data during DAP.

To mitigate vertical forgetting, AF Adapter [327] proposes an adapter structure extending the original transformer block. This structure extends the width of Attention layers and FFNs for domain-specific knowledge storage. During DAP, the general-domain parameters inherited from the BERT checkpoint remain frozen, while only the adapters are trained. To gauge the extent of forgetting after DAP, the authors collect a small subset of general-domain samples from WuDao [341] and calculate the accuracy of masked word prediction using the same pre-training protocol as BERT and DAP of AF Adapter. They find minimal performance degradation compared to other full-parameter fine-tuning techniques. HuatuoGPT-II [42], proposes to fuse the DAP into the final SFT, making the two-stage development one unified protocol. The challenge of such process mainly comes from the data heterogeneity of the domain-adaptive pre-training unlabeled corpus. The authors address this challenge by reformulating paragraphs of data into *(instruction, output)* format using existing large language models. During the unified one-stage SFT, they employ a priority sampling strategy to avoid compromising downstream ability, as seen in fixed-rate data mixing [280]. Unlike previous approaches, this paper empirically demonstrates the superiority of unified one-stage SFT over two-

[3]Detailed web links for data collection in SaulLM are omitted here for brevity. Refer to the original paper for more information.

stage training, questioning the current computational paradigm of adapting LLMs to specific domains via DAP. In [243], the authors investigate the effectiveness of continual learning techniques for mitigating catastrophic forgetting in DAP on medical-domain corpus. They find that LMs constrained by continual learning techniques on source domains exhibit greater robustness to future domain shifts. Specifically, they identify that parameter regularization techniques like EWC [140], despite slightly higher computational cost, can facilitate beneficial forward and backward transfer, offering valuable insights to the community.

Financial Domain. Similar to the medical domain, large language models hold immense potential for enhancing financial communication, decision-making processes, and risk assessment for both traders and ordinary individuals [256, 333, 289, 152]. However, the financial domain entails high stakes, with even minor errors bearing significant consequences. Thus, integrating LLMs into financial workflows requires heightened caution to avoid inaccuracies or misunderstandings that could lead to substantial financial losses or the dissemination of misleading information. Despite advancements, a gap persists between general-purpose large language models and existing domain-specific smaller-scale language models [8, 312], 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 [177, 152, 323, 326, 355].

BBT-Fin [177] 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. This knowledge enhancement pre-training entails: (i) actively selecting sentences containing specific knowledge triplets *(head entity, relation, tail entity)*, masking one of them, and (ii) simultaneously masking 15% of a random-length span. CFGPT [152] creates CFData, a financial dataset for domain-adaptive pre-training and supervised fine-tuning, comprising 141 billion tokens. The dataset includes corporate prospectuses, announcements, research reports, social media content, financial news, and Wikipedia articles. During DAP, CFGPT does not employ continual learning techniques but utilizes QLoRA [66] for preventing overfitting to downstream data and balancing general response ability and domain-specific ability during supervised fine-tuning. These two methods are typical domain-specific LLM studies focusing solely on target domains, without explicit continual learning measures or evaluation of performance degradation on general skills.

In [323], the authors aim to enhance the data efficiency of domain-adaptive pre-training. They propose two data selection techniques: (i) efficient task-similar (ETS) and (ii) efficient task-agnostic (ETA) domain-adaptive pre-training. These methods are inspired by the generalization bound of domain adaptation [18, 77, 260]. Suppose the source domain data distribution is $\mathcal{D}$ and the task data distribution is $\mathcal{T}$. The generalization error $\epsilon_{\mathcal{T}}(h)$ of the hypothesis h on the target task distribution $\mathcal{T}$ can be given as

$$\epsilon_{\mathcal{T}}(h) \leq \epsilon_{\mathcal{D}}(h) + \tfrac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}, \mathcal{T}) + \mathcal{C}, \tag{13}$$

where ϵ is the 0-1 error function for binary classification, h represents the model, $d_{\mathcal{H}\Delta\mathcal{H}}$ is the $\mathcal{H}\Delta\mathcal{H}$ divergence that measures the distributional discrepancy between two distributions based on a hypothesis set $\mathcal{H}$ used for discriminating different data distributions [18], and $\mathcal{C}$ is a constant. Therefore, based on the theory, finding a set of examples from the DAP corpus that are similar to the downstream task's data, i.e., $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}, \mathcal{T})$ is low, should help the final-stage adaptation to the tasks. In the context of large language models (LLMs), [323] suggests ensuring *novelty* and *diversity* in the sampled corpus for DAP. Here, high perplexity of an LLM on a sentence indicates high novelty, while high entropy of part-of-speech (POS) tags on a sentence indicates high diversity [19, 348]. This approach significantly enhances DAP efficiency: it utilizes only 10% of the originally collected data yet outperforms models trained on the entire dataset, underscoring the importance of data quality over quantity in DAP. Furthermore, with the reduced number of examples, the authors do not observe any signs of forgetting regarding the model's capacity in open-domain scenarios. WeaverBird [326] introduces an intelligent finance dialogue system, where the encoder is trained on Chinese and English financial documents, alongside expert-annotated financial query-response pairs, using LoRA [110]. Xuanyuan 2.0 [355], akin to HuatuoGPT-II [42], proposes the technique of hybrid-tuning, which fuses the stage of DAP and SFT into one, general-domain data and financial-domain data into one. Notably, the distribution of data in hybrid-tuning is non-conventional: financial pre-training data comprises 13%, general pre-training data 20%, and instruction tuning data 67%. Within the instruction tuning

data, the financial domain accounts for approximately 30%. This prompts a pertinent question in line with the investigation on efficient DAP in [323]: Is a large DAP dataset necessary for developing a domain-specific LLM?

Scientific Domain. Vertical scientific large language models [276, 336, 322, 350] span many subjects, including astronomy [209, 224], mathematics [10, 338, 181, 344, 84], geology [78, 242, 166, 286], chemistry and physics [246], biology [331, 360, 32, 1, 40, 20, 367]. However, among all the studies listed above, only a small fraction of them adopt the technique of DAP.

OceanGPT [20] is the first LLM tailored specifically for the ocean domain, catering to a range of downstream ocean science applications. It compiles a raw corpus of ocean science literature, prioritizing recent research and historically significant works, and performs DAP on the Llama 2 model. K2 [63] 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 [110] on both a general instruction tuning dataset and the GeoSignal benchmark introduced within the K2 framework. AstroLlama [209] gathers abstracts solely from astronomy papers on arXiv and proceeds with pre-training focused on next token prediction. It observes an improved perplexity on the domain of scholarly astronomy, without providing more quantitative evaluation. MarineGPT [367] 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 the parameters of a Q-Former [153] between the frozen visual encoder [71] and text decoder [279]. The four methods introduced above lack further validation regarding the necessity of DAP deployment. They seem to adopt DAP merely out of convention, likely for the convenience of utilizing domain-specific corpora as pre-training data.

Another branch of methods proactively integrate in the replay of the general-domain data to mitigate vertical forgetting. GeoGalactica [166] introduces a series of large language models 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, which is crucial to fine-tuning on the downstream tasks. However, 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 [10] focuses on mathematics, initialized from Code Llama [245], and undergoes DAP on a blend of the 55 billion mathematical pre-training dataset (Proof-Pile-2 including scientific papers, web data, and mathematical code) and general domain data (Pile [79]) at a 19:1 ratio. In contrast, PLlama [331], designed for plant science, mixes domain-specific and general-domain data at a ratio of 9:1.

Code Domain. The development of LLMs for automatic code filling, debugging, and generation holds significant practical importance, with programmers worldwide benefitting from recent advancements in coding assistants [206, 272]. These advancements cover various frameworks, including encoder-only [206], encoder-decoder [296, 293, 35], and decoder-only [215, 214, 366, 43, 158, 176, 87]. As noted in [272], in the era of LLMs, there's a growing trend towards decoder-only architectures, leveraging models pre-trained on general natural language like Llama [279, 280]. Additionally, there's a shift in the training objective from utilizing code structures to simpler tasks like next token prediction and infilling.

From a continual learning perspective, the code domain presents unique advantages and challenges for domain-adaptive pre-trainig (DAP) compared to other domains discussed so far. On one hand, its hierarchical structure (*general domain corpus* $\rightarrow$ *multi-language code* $\rightarrow$ *specific programming language*) provides an ideal training pipeline for DAPs [245], offering potential for more efficient training strategies. On the other hand, programming languages adhere to strict grammars, unlike the more fuzzy and context-dependent nature of natural language. Consequently, language models should ideally leverage these structures through tailored designs. However, given the vast availability of code for training, particularly in popular languages like Python and Java, adopting the same training objectives as for natural languages may yield sub-optimal results. Therefore, many existing works in the field omit domain-adaptive pre-training [296, 293, 186, 207, 123, 305, 372, 69, 155]. 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 [215] 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. Specifically, CodeGen-NL is trained on Pile [79], while CodeGen-Multi further trains on a subset of the BigQuery dataset[4], which includes source code from six popular programming languages (C, C++, Go, Java, JavaScript, and Python). Subsequently, CodeGen-Mono is trained on BigPython, a large-scale corpus of Python code. Throughout the training process, consistent pre-training objectives are employed, focusing on next token prediction. Comment-Aug [269] addresses the challenge of aligning programming languages with natural languages (PL-NL alignment). Recognizing that denser comments in existing code could enhance the model's generation capabilities, Comment-Aug proposes a self-augmentation strategy. Initially, it enhances LLMs with comment generation ability through instruction tuning. Then it augments sparsely commented code by generating additional comments, followed by further pre-training on the enriched code data. StarCoder [158] introduces two models: StarCoderBase and StarCoder. StarCoderBase is initially trained on a mixed dataset comprising various programming languages. With the aim of benefiting a broader user base, it maintains the original data distribution without significant reweighting. Subsequently, StarCoderBase undergoes further fine-tuning on an additional 35 billion tokens of Python code, resulting in the development of StarCoder. DeepSeek-Coder-v1.5 [87] originates from DeepSeek-LLM [62] and undergoes pre-training on 2 trillion tokens, comprising 87% source code, 10% English code-related natural language, and 3% Chinese natural language corpus. Unlike Deepseek-Coder, which employs both next token prediction and fill-in-the-middle objectives, DeepSeek-Coder-v1.5 focuses solely on next token prediction during the stage of DAP. Despite this change, 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. This underscores the efficacy of DAP before final downstream tasks in achieving balanced performance across general and domain-specific domains.

As the only work investigated so far that utilizes the general data replay to mitigate vertical forgetting in the code domain, Code Llama [245] introduces a sophisticated training framework tailored for various coding tasks and model sizes, including 7B, 13B, 34B, and 70B variants. Initialized from Llama 2 weights, these models undergo continuous pre-training 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. During DAP, Code Llama optimizes two objectives: autoregressive next token prediction and code infilling prediction (except for the 34B model, which excludes infilling). Subsequently, Code Llama undergoes further refinement through long-context fine-tuning (LC-FT) to enhance its repository-level understanding. Building upon Code Llama, Code Llama-Instruct undergoes additional instruction-tuning, while a Python-specific variant of Code Llama undergoes additional domain-adaptive pre-training on Python language datasets before LC-FT.

Architecture expansion has proven effective in acquiring robust coding abilities and preventing vertical forgetting simultaneously. IRCoder [221] investigates the potential of utilizing compiler intermediate representations to enhance the multilingual transferability of Code LLMs. By conducting DAP on code grounded in intermediate representations with LoRA [109], IRCoder achieves superior multilingual programming instruction following, enhanced multilingual code understanding, and increased robustness to prompt perturbations. Llama Pro [310], initialized from Llama2, undergoes continuous pre-training 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 serve as identity mappings, preserving the original functionality, and will be tuned to adapt to the DAP data. The proposed expansion method is asserted to be more resilient against catastrophic forgetting of previous general knowledge compared to sequential fine-tuning and other parameter-efficient tuning methods like LoRA.

The three aforementioned works highlight the importance of domain-adaptive pre-training in the realm of code. However, it is crucial to note that the problem definition and conventional architectures of existing Code LLMs may present compatibility challenges for DAP deployment. This situation leads to trade-offs between specialization in the code domain and generalization ability in the natural language domain.

[4]Link to BigQuery: https://cloud.google.com/bigquery/public-data.

Other Domains. ECONET [93] enhances the model's ability to reason about event temporal relations through a dedicated continual pre-training phase. In this phase, temporal and event indicators are masked out, and a contrastive loss is applied to the recovered masked tokens. Results demonstrate that incorporating this domain-adaptive pre-training stage significantly improves performance on final tasks compared to direct fine-tuning. In Concept-Aware Language Model (CALM) [369], the authors introduce a data-efficient domain-adaptive pre-training approach. This method incorporates both generative and discriminative commonsense reasoning to enhance the concept-centric commonsense reasoning ability of large language models. Similar to ECONET, CALM's DAP phase is specifically tailored for concept-centric reasoning tasks. Consequently, even a small number of data examples for continued pre-training can lead to notable improvements in downstream tasks.

EcomGPT-CT [187] is a large language model tailored for the E-commerce through domain-adaptive pre-training. Given that E-commerce data often exhibits a semi-structured format, stored in tables or databases, EcomGPT-CT employs a data mixing strategy. This approach transforms semi-structured data into a set of nodes and edges, samples a cluster of nodes, and then extracts and concatenates them into a training example. During DAP, 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, AdaptLLM [49] challenges the data efficiency of conventional DAP by adopting a novel approach inspired by human learning patterns in reading comprehension. Following techniques from previous work [281], it transforms raw corpora into *(raw text, question, answer)* format, creating intrinsic tasks. The model is trained on reading tasks (next token prediction on the raw text) and comprehension tasks (question-answering based on the corpora). During DAP, to ensure instruction diversity beyond predefined templates, it integrates substantial amounts of general instruction tuning data at ratios ranging from 1:1 to 1:2 across various domains like biomedicine, finance, and law. Compared to traditional DAP approaches that utilize raw domain-specific corpora as-is, AdaptLLM's method demonstrates superior domain-specific knowledge adaptation and minimal vertical 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 doesn't require consideration of further adaptation to another downstream tasks, simplifying optimization objectives to a great extent: better adaptation and less forgetting. In essence, the goal is to improve the overall performance. The challenge of CFT has been extensively explored in the continual learning community. Essentially, all existing continual learning literature can be seen as a variant of CFT: models are either randomly initialized or initialized from pre-trained weights and undergo CFT thereafter. In the realm where continual learning intersects with natural language processing, we will only briefly outline the most notable works in this domain in 4.3.2, and direct interested readers to additional survey literature on this topic [22, 132].

In the era of LLMs, new computational paradigms in CFT have emerged and attracted significant attention within the research community. These topics include Continual Instruction Tuning (CIT, Section 4.3.3), Continual Model Refinement (CMR, Section 4.3.4), Continual Model Alignment (CMA, Section 4.3.5), and Continual Learning for Multimodal Language Models (CMLLMs, Section 4.3.6). While all these fall under the umbrella of CFT, each presents distinct features and challenges. In CIT, models must generalize to new tasks encoded in instructions, requiring semantic understanding [358]. CMR demands fine-grained, possibly example-level, operations for model refinement, differing from task-based approaches [96]. CMA aligns models with evolving human preferences, challenging due to subjective nature and lack of clear task boundaries [165, 352]. In CMLLMs, addressing the composite architectural design and preventing catastrophic forgetting are key challenges [100, 213]. Detailed exploration of these sub-categories follows in subsequent chapters.

Summary of Continual Fine-Tuning LLMs. We have organized existing research on continual fine-tuning in Table 3, categorizing studies into sub-categories: (i) general continual fine-tuning, (ii) continual instruction tuning (CIT), (iii) continual model refinement (CMR), (iv) continual model alignment (CMA), and (v) continual multimodal LLMs (CMLLMs). The table includes details on incremental learning types (X-IL), LLM architecture, and employed continual learning techniques and evaluation metrics. After discussing general observations on CFT in Section 4.3.1, we'll delve into each sub-category in detail.

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 [282], 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. Among 34 papers shown in the table, 100% (34/34) of them explicitly deploy the continual techniques to address the challenge of CFT. Furthermore, 30% (10/34) of them develop their own new techniques that cannot be easily categorized into the three mainstream sets of continual learning algorithms.

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 Continual Fine-Tuning	CTR [133]	DIL CIL	BERT	✗	✗	Adapter	✗	✓	✓	✓
	[274]	TIL	BERT	S-Replay	✗	✗	✗	♣	♣	♣
	CIRCLE [342]	DIL	T5	Replay	EWC	Prompt	✗	✓	✓	✓
	ConPET [268]	DIL	Llama	Replay	✗	LoRA	✗	✓	✓	✓
	[12]	DIL CIL	BERT	✗	✗	✗	G-Prompt	✓	✓	✗
	[183]	TIL	DistilBERT ALBERT RoBERTa	ER DER LwF	✗	✗	✗	♣	♣	✗
	SEQ* [365]	TIL CIL	BERT Pythia GPT2	✗	P-Freeze	✗	Tricks for Classfiers	✗	✓	✗
	LFPT5 [230]	DIL	T5	P-Replay	✗	✗	✗	✓	✓	✗
	[306]	DIL	RoBERTa GPT2	Replay	EWC SI RWalk	✗	✗	✓	✓	✗
	LR ADJUST [307]	DIL	XLM-R	✗	✗	✗	LR Scheduling	✓	✓	✓
	C3 [47]	TIL	T5	KD	✗	Prompt Tuning	✗	✓	✓	✗
Continual Instruction Tuning	CT0 [255]	TIL	T0	S-Replay	✗	✗	✗	✓	✓	✓
	RCL [292]	TIL	LLaMA Vicuna Baichuan	Replay	✗	✗	✗	✓	✓	✓
	DynaInst [205]	TIL	BART	Replay	✗	✗	✗	✓	✓	✓
	CITB [358]	TIL	T5	Replay AGEM	L2 EWC	AdapterCL	✗	✓	✓	✓
	SSR [113]	TIL	LLaMA Alpaca	RandSel KMeansSel	✗	✗	✗	✓	✓	✓
	KPIG [103]	DIL TIL	LLaMA Baichuan	DynaInst PCLL DCL	L2 EWC	DARE LM-Cocktail	KPIG	✓	✓	✓
	ConTinTin [337]	TIL	BART	Replay	✗	✗	InstructionSpeak	✓	✓	✓
	O-LoRA [291]	TIL	LLaMA Alpaca	✗	✗	O-LoRA	✗	✓	✓	✓
	SAPT [363]	TIL	T5 LLaMA	✗	✗	✗	SAPT	✓	✓	✓
	InsCL [294]	TIL	LLaMA	Replay	✗	✗	InsCL	✓	✓	✓
Continual Model Refinement	CMR [162]	DIL	BART	ER MIR MLR	L2 EWC	✗	✗	✓	✓	✓
	GRACE [96]	DIL	T5 BERT GPT2	✗	✗	Adapter	✗	✓	✓	✗
	WilKE [108]	DIL	GPT2 GPT-J	✗	✗	Adaptor	✗	✓	✓	✓
	Larimar [58]	DIL	BERT GPT-J	✗	✗	✗	Kanerva Memory	✓	✓	✓
	MELO [339]	DIL	BERT GPT2 T5	✗	✗	LoRA	✗	✓	✓	✓
	CME [156]	DIL	BERT	Replay	✗	✗	Inner-Prod. Reg.	✓	✓	✓
Continual Model Alignment	COPF [351]	TIL DIL	LLaMA	Replay	Function Reg.	Prompt	✗	✓	✗	✓
	AMA [165]	DIL	OpenLLaMA Mistral	Replay	L1/L2	LoRA	Adaptive Model Avg.	♣	♣	♣
	CPPO [352]	TIL	GPT2	✗	Weighting Strategy	Prompt	✗	✓	✓	✓
Continually Fine-tuning Multimodal LLMs	EProj [100]	TIL	InstructBLIP	✗	TSIR	Projector Exp.	✗	✓	✗	✓
	Fwd-Prompt [364]	TIL	InstructBLIP BLIP2	✗	✗	Projector Exp.	✗	✓	✓	✓
	CoIN [41]	TIL	LLaVA	✗	✗	MoE LoRA	✗	✓	✗	✓
	Model Tailor [371]	TIL	InstructBLIP LLaVA-1.5	✗	Model Tailor	✗	✗	✓	✓	✓
	RebQ [362]	TIL	ViLT	✗	✗	Prompt Tuning	✗	✓	✗	✓

4.3.1 General Observations about Continual Fine-Tuning

Examining the landscape of continual learning in the context of large language models (LLMs), and combined with the results shown in Table 3, we can discern significant trends and shifts within the research community. Firstly, ***there has been a noticeable transition in focus from class-incremental learning to domain-incremental and task-incremental learning paradigms***. For example, in 12 papers of general continual fine-tuning (the first white section), only 3 papers study continually fine-tuning language models in the setting of class-incremental learning. In the remaining four topics, CIT completely belongs to the field of task-incremental learning, as the instruction provided to the models can be seen as a soft encoding of the task information; CMR completely belongs to the field of domain-incremental learning, as different editing examples follows the same problem definition, and no task information is provided during inference. In CMA and CMLLMs, no examples of class-incremental learning has been reported.

It has been a longstanding common sense in the continual learning community that class-incremental learning, as it requires the model to predict the context label and within-context label at the same time [282, 288, 139], is the most challenging continual learning scenario and hence receives most of the attention from the community. However, the evolution observed so far suggests a recognition of the broader spectrum of challenges faced in real-world applications of continually learning LLMs. This growing awareness of the importance of task-incremental and domain-incremental learning underscores the necessity for more comprehensive and specific approaches to continual LLMs, with precedent continual learning literature provided as reference.

Furthermore, ***in continual fine-tuning (CFT), continual learning techniques enjoy broader adoption and explicit exploration compared to CPT and DAP***. In Table 3, all 34 papers explicitly deploy the continual techniques, 50% of which develop their own new techniques that cannot be easily categorized into the three mainstream sets of continual learning algorithms. For example, in instruction tuning, SAPT designs a shared attentive learning framework to enable the catastrophic forgetting mitigation and knowledge transfer at the same time [363]; in model refinement, Larimar proposes to adopt an external memory system, Kanerva Memory, that supports operations of read, write, and generate [58]; in model alignment, AMA proposes adaptive model averaging method to achieve Pareto-optimal when averaging different models for a reward-tax trade-off (alignment-performance trade-off) [165]; in continual learning multimodal LLMs, [362] devises a novel method named Reconstruct before Query (RebQ), harnessing the multi-modal knowledge from a pre-trained model to reconstruct the absent information for the missing modality.

Hence we can further conclude that, beyond mere replication of existing techniques, ***researchers are actively developing tailored solutions for various scenarios where an LLM needs to be continually updated directly for downstream tasks.*** This emphasis underscores the recognition of continual learning as a pivotal component in the development of resilient and adaptive language models, and further signals a maturation of the field towards more specialized and effective continual learning methodologies for large language models.

4.3.2 General Continual Fine-Tuning (General CFT)

Researchers have long investigated the phenomenon of forgetting resilience in pre-trained large language models when fine-tuned for various downstream tasks [133, 274, 183, 365, 195]. Although the pre-trained weights initially position the model in a flat-loss basin, aiding adaptation to future tasks without severely impacting previous ones [95, 208, 199, 195], 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 [313, 274, 183, 365]. Therefore, additional measures are necessary when deploying these models in real-world applications [133, 342, 12, 230, 306, 47]. For instance, [306] demonstrate that relying solely on the intrinsic anti-forgetting and generalization abilities of LLMs can be risky in scenarios with changing code and API distributions, and simple baseline continual learning techniques can significantly enhance performance.

Many studies advance beyond naive sequential fine-tuning, leveraging the inherent anti-forgetting nature of LLMs while avoiding overly complex techniques of continual learning. They favor sequential fine-tuning due to its simplicity and predictability in real-world scenarios [307, 365]. For instance, LR ADJUST [307] 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 [21], SEQ* [365] introduces several strategies for fine-tuning LLMs on a sequence of downstream classification tasks: (i) freezing the LLMs and old classifiers after the warm-up phase and learning new tasks, respectively; (ii) employing (cosine) linear classifiers when old data is (not) available; and (iii) pre-allocating future classifiers.

Given the minimal forgetting observed at the representation level in continual learning, some studies aim to tackle the misalignment between the representation space and the decision-making layers, such as the final classification layer, by introducing representation-level constraints during continual fine-tuning. NeiAttn [12] exemplifies this approach by formulating classification tasks as masked language modeling and proposing a neighboring attention mechanism to counteract negative representation drift. This method seamlessly complements existing continual learning techniques [38, 37, 270], as its focus is independent of mainstream continual learning methods.

Another line of approaches delves into refining the input/output format and network architectures of pre-trained language models, crafting specific structures and training methods to tackle the challenges of continual fine-tuning. For instance, CTR [133] integrates a pre-trained model at the onset of incremental learning, incorporating two CL-plugin modules within transformer layers. These modules consist of a task-specific module (TSM) for acquiring task-specific knowledge and a knowledge-sharing module (KSM) for selectively transferring previously learned similar knowledge. CIRCLE [342] manually designs diverse prompt templates for various types of buggy code, unifying them into representations solvable through cloze tasks. It employs difficulty-based example replay to enhance continual program repair, outperforming existing automatic program repair (APR) methods. LFPT5 [230] addresses lifelong few-shot language learning by consolidating sequence labeling, text classification, and text generation tasks into a text-to-text format. The model, treated as a generative model, undergoes prompt tuning and data generation, generating pseudo-examples from previous domains during adaptation to new tasks and performing knowledge distillation. In [357], 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 using hidden state mixing. During training, this module addresses the task of fine-tuning while simultaneously regenerating examples. C3 [47] merges parameter-efficient fine-tuning (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, equipped with tunable prompts, minimizes the KL-divergence of both models' autoregressive token distributions per step to retain the teacher model's few-shot capability.

4.3.3 Continual Instruction Tuning (CIT)

Instruction Tuning (IT) is a technique used to refine the instruction-following capabilities of LLMs [353]. While LLMs are typically pre-trained on extensive and diverse corpora, they may struggle with specific tasks despite their general knowledge. Numerous studies have shown that IT can notably improve LLMs' ability to follow textual instructions for particular tasks [353, 301, 125, 249, 218], leveraging the pre-existing knowledge within LLMs to bridge the gap between general and task-specific performance [302]. Additionally, IT enhances the intuitive interaction between humans and LLMs, providing a more natural interface and aligning LLM outputs more closely with human expectations and preferences [184].

Continual Instruction Tuning (CIT). When the instruction tuning data comes in as a stream, this series of the fine-tuning tasks need to address the challenge of catastrophic forgetting of the general knowledge. CT0 [255] represents the inaugural study on continual learning in fine-tuned LLMs, utilizing the replay method during the IT process on the base T0 model. This model successfully learns 8 new tasks while maintaining robust performance on previously learned tasks. Subsequent research has focused on enhancing the replay method used during CIT. For instance, [103] 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 [113] uses the LLM to generate synthetic instances for replay, achieving superior or comparable performance with greater data efficiency than traditional methods.

Other approaches include combining multiple continual techniques during CIT. DynaInst [205] merges parameter regularization with dynamic replay, selectively storing and replaying instances and tasks to enhance outcomes. InstructionSpeak [337] employs negative training and replay in-

structions to improve both forward transfer and backward transfer. Additionally, some methods incorporate Parameter Efficient Tuning (PET). Orthogonal Low-Rank Adaptation (O-LoRA) learns new tasks within an orthogonal subspace while preserving LoRA parameters for previous tasks [291]. Shared Attention Framework (SAPT) combines a PET block with a selection module via a Shared Attentive Learning & Selection module, and tackles catastrophic forgetting and knowledge transfer concurrently [363]. While regularization-based and architectural-based methods require additional parameter storage and GPU memory during fine-tuning, which potentially leads to higher training costs, replay-based methods remain popular due to their simplicity and data efficiency [294]. These methods are particularly favored in traditional continual learning scenarios for tuning LLMs.

CIT vs Conventional CL. Both CIT and conventional CL aim to enable LLMs to acquire new tasks or information over time while retaining previously learned knowledge. However, CIT specifically utilizes rich natural language instructions to enhance LLMs' ability to follow human instructions [358]. The natural-language encoding of task information enables the potential for positive forward transfer when semantically similar tasks are encountered in the data stream. This scenario is typically challenging to engineer manually in conventional continual learning setups. In contrast, conventional CL focuses on broader knowledge acquisition and is not limited to instruction-based learning. While CIT is predominantly applied within LLMs, conventional CL is employed across various fields, including vision, multimodal models and robotics. Regarding challenges, both approaches contend with catastrophic forgetting; however, CIT additionally concentrates on refining instruction-following capabilities to better interact with human needs. Conversely, conventional CL emphasizes building robust generalizability to manage variations within and across tasks [288].

4.3.4 Continual Model Refinement (CMR)

Like humans, LLMs are prone to errors, such as inaccurate translations or outdated information [60]. To keep LLMs updated with the evolving factual knowledge, directly fine-tuning the model to correct these mistakes can be time-consuming and may disrupt its performance on previously learned tasks. To overcome these challenges, Model Refinement (also known as Model Editing) is proposed. This approach aims to rectify the model's errors while preserving its performance on other inputs, with only moderate computing resources [264, 60, 203, 98, 117, 204, 96]. The concept of model editing was initially explored in [264], which introduced a *"reliability-locality-efficiency"* principle and proposed a gradient descent editor to address it efficiently. Subsequent research, such as [60] and [203], extended this principle to edit factual knowledge in BERT-based language models and larger models like GPT-J-6B [285] and T5-XXL [235], 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 [204] and [96], incorporate editing through retrieval mechanisms.

Model Refinement as a Special Form of Continual Learning. Although model refinement and continual learning are typically treated as separated fields, they share fundamental similarities. The principle of *locality* in model refinement, ensuring that new knowledge doesn't disrupt responses to other inputs, aligns with the non-forgetting objective of continual learning. Furthermore, the principles of *reliability*—effectively updating the model—and *efficiency* are crucial in both model refinement and continual learning Thus, the problem of model refinement can be viewed as akin to continual learning, where a small batch of updated samples $\{(\boldsymbol{x}_e, \widehat{y}_e)\}$ represents a new task. Alternatively, one can consider the stream of such samples for updating as a specialized form of online continual learning [335, 189, 228].

Continual Model Refinement (CMR). Recent works have combined continual learning with model refinement, resulting in a new problem termed continual model refinement (CMR). This concept extends model refinement horizontally, presenting updated sample pairs $(\boldsymbol{x}_e, y_e, \widehat{y}_e)_N^{e=1}$ sequentially as a stream. [162] initially introduces this idea as continual model refinement, evaluating various continual learning methods with a dynamic sampling algorithm. To solve CMR problem, many methods employ a retrieval mechanism. For instance, [96] uses hidden activations of the language model as a "key" to activate updated parameters only when input x_0 resembles updated sample pairs; [339] improves this approach's efficiency by integrating LoRA [109]; [58] 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, [108] addresses the issue of "toxicity buildup and flash" in single-editing methods like ROME [196], adapting it to a continual context with a knowledge-aware layer selection algorithm.

While all these works pioneer research in CMR, the exploration of continual model refinement for LLMs remains open. [97] highlighted 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 model editing methods [196, 197], and could become a significent concern for CME [108]. 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 explored and answered.

4.3.5 Continual Model Alignment (CMA)

Model Alignment (MA) is a crucial concept in the development and deployment of AI systems, ensuring that their actions and outputs align with human values, ethics, and preferences. It can be defined as the process of adjusting the objectives and functioning of an AI system to achieve such goals, involving a combination of mathematical models, algorithmic adjustments, and iterative feedback to refine AI behavior [218, 234]. MA can be broadly categorized into two types: Reinforcement Learning-based (RL-based) and Supervised Learning-based (SL-based). In the RL-based approach, as discussed in [315, 218, 252], models are trained to make decisions reinforced by human feedback, using a reward system to guide them towards desirable outcomes. Conversely, the SL-based approach, as seen in [104, 234, 122], directly trains models on datasets of human preferences, aligning their output with demonstrated human values. Both approaches leverage a combination of algorithmic learning techniques and human feedback to progressively refine and align model behaviors.

When language models undergo the phase of model alignment (MA) to align with specific ethical or normative standards, the performance degradation occurs. In [165], the authors refer to this phenomenon of general knowledge forgetting induced by model alignment 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 its training distribution. This paper focuses on understanding and reducing the impact of the Alignment Tax in the process of refining AI models, highlighting the balance between alignment and maintaining broad model capabilities.

Continual Model Alignment (CMA). Continual Model Alignment (CMA) aims to continuously refine LLMs to align with evolving human values, ethics, and data. The significance of CMA lies in its capacity to ensure that LLMs retain their relevance, accuracy, and ethical alignment over time. This ongoing adjustment is essential to navigate the complexities introduced by concept drift, the evolution of data, and shifts in societal values. 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 [275]. Despite its importance, CMA faces several challenges: (i) *scalability and resource intensity issue*: continually updating LLMs requires significant computational resources and human oversight, posing scalability issues for the practitioners; (ii) *ensuring ethical alignment*: balancing model updates to reflect societal changes without introducing or perpetuating biases remains a complex issue; (iii) *data privacy and security*: continuously integrating new data into LLMs raises concerns regarding data privacy, security, and the potential for misuse.

Emerging strategies in CMA emphasize the optimization of LLMs for enhanced adaptability, addressing the necessity for these models to accommodate continuous changes in language use, information validity, and societal expectations. Notably, innovative research efforts such as those detailed in [165], and [229], highlight the development of methodologies designed to minimize the challenges associated with the continuous realignment of LLMs. These studies underscore the importance of implementing specific optimization strategies to foster the adaptability of LLMs, ensuring their outputs remain ethically attuned and factually relevant in the face of dynamic societal and informational landscapes.

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: [165] explores methodologies for reducing the overhead associated with continual alignment through efficient reinforcement learning techniques, and [352], which suggests a framework for applying continual learning principles to reinforcement learning with human feedback, potentially mitigating the alignment tax over time. Employing reinforcement learning for CMA focus on developing efficient reinforcement learning techniques that integrate continual learning principles and human feedback to dynamically maintain alignment with evolving human values while minimizing computational overhead and mitigating forgetting. For

SL-based CMA, [351] presents an innovative approach by integrating supervised continual learning techniques with the process of aligning AI systems through direct training on evolving datasets of human preferences. This methodology promises a more sustainable and adaptable model for maintaining alignment with human values over extended periods and across various contexts. The use of supervised continual learning suggests a focus on preventing catastrophic forgetting, a common challenge where a model loses its ability to perform previously learned tasks upon learning new ones.

In summary, both RL-based and SL-based frameworks need to address forgetting, especially in scenarios where the model continuously integrates new information. Effective strategies might include techniques (such as EWC [140], experience replay [38], and dynamic re-weighting [165]), which help the model retain old knowledge while integrating new insights. Future research in CMA aims to develop more efficient, automated processes for model updates, better mechanisms for ethical oversight, and innovative solutions to balance model relevance with privacy and security concerns. Existing streams of research highlight the importance of developing adaptable, efficient, and robust AI systems that can continually align with human values without substantial losses in performance or increased computational costs. Future research could explore hybrid models that combine RL-based and SL-based approaches, potentially offering a more holistic framework for continual model alignment.

4.3.6 Continual Multimodal Large Language Models (CMLLMs)

Multi-modal LLMs have garnered significant attention for their capacity beyond single modality. These models integrate data from multiple modalities, like texts, images and videos to enhance real-world information comprehension [222, 148]. Typically, MLLMs consist of modality-specific sub-modules such as pre-trained vision encoders, large language models, and projectors for cross-model alignment. This alignment is essential for MLLMs to fuse the diverse data types and promote their comprehension. For example, MiniGPT-4 [370] utilizes a linear projector to align frozen vision encoders and language models; LLaVA [167] employs a simple linear layer to connect image features and instruction into the word embedding space. Currently, all existing MLLMs are pre-trained on large scale multi-modal datasets and then fine-tuned on specific small downstream datasets, which constitutes the training process [56, 154].

Several existing studies have explored the causes of catastrophic forgetting when continually training MLLMs. [364] 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. [349] 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 Pre-Training MLLMs. Training an MLLM to be updated with the changing world from scratch is resource-intensive, requiring considerable time and cost. While continual learning offers a promising solution to this problem, trivially applying past methods is not advisable, given the distinct structure of MLLMs that includes modality-specific sub-modules and cross-model alignment. Currently, there is an apparent lack of continual pre-training for MLLMs, and further exploration into continual and joint pre-training of MLLMs is necessary.

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 [349, 100, 364, 41, 371]. Given the strong capabilities of pre-trained models, training specific layers suffices, and can simultaneously reduce computational demands. [362] 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. [364] also observes negative forward transfer, where the performance of unseen tasks degrades when learning new tasks, indicating a decline in model generalization capability.

Continual Learning MLLMs. While traditional CL methods are applicable, some may not yield optimal results, as evidenced by various experiments [100, 364]. For instance, [100] 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. [100] proposes EProj to expand the projection layer in MLLMs for each new task and utilizes task-similarity-informed regularization (TIR) to enhance performance. [364] 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. [371] 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. [362] 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., [41] first introduces MoELoRA to fine-tune LLaVA, effectively mitigate the catastrophic forgetting of MLLMs in CoIN and the results demonstrate the effectiveness.

In concluding remarks on Continual Learning for MLLMs, the role of templates in instruction tuning emerges as crucial. As highlighted by [41], employing similar templates across tasks proves more advantageous, aiding in knowledge retention and forgetting mitigation. This approach fosters task-specific learning, reducing reliance on common knowledge prone to forgetting in sequential contexts. In addition, it is noteworthy that the forgetting induced by the gap between tasks is more critical than the forgetting induced by the distributional gap between datasets. According to [349], moderate fine-tuning is advantageous for non-fine-tuned tasks, excessive fine-tuning ultimately leads to catastrophic forgetting in these tasks. [100] discovers the multi-task joint instruction tuning at the beginning state can facilitate the model's continual learning ability and mitigate forgetting. Overall, continual learning in MLLMs holds promise, but further research is needed to fully realize its potential.

5 Evaluation Protocols and Datasets

In this section, we introduce the evaluation protocols and datasets for continually learning large language models. In Section 5.1, we discuss common continual learning evaluation metrics adapted for this context, along with metrics designed specifically for continual LLMs. Then, in Section 5.2, we outline the datasets available for each discussed topic.

5.1 Evaluation Protocols

In the realm of conventional continual learning, where task streams take the form of classification, many metrics rely on the concept of Accuracy Matrix [174, 260]. Extending this notion to the context of continually learning LLMs, we introduce the **Performance Matrix** $\boldsymbol{P} \in \mathbb{R}^{T\times T}$, where T represents the total number of training stages. Each entry of $\boldsymbol{P}$ corresponds to a performance metric evaluated on the models, such as perplexity on pre-training data [127, 46, 89], zero-shot/few-shot evaluation metrics on downstream data without fine-tuning [52, 311, 10, 63, 215, 245], fine-tuned accuracies on downstream tasks [6, 231, 46, 120], and probing accuracies from fine-tuning add-on components evaluated on downstream tasks [274, 183, 365]. In $\boldsymbol{P}$, $P_{i,j}$ denotes the model's performance after training on task i and evaluating on task j. With this Performance Matrix definition, we introduce the primary evaluation protocols widely adopted.

Overall Performance (OP). The Overall Performance (OP) [133, 357, 351] is a natural extension of the concept of Average Accuracy [174, 260]. The OP measured up until training stage t is the average performance of the model trained right after the stage t. Denote it as OP_t and we have:

$$\mathrm{OP}_t \triangleq \frac{1}{t}\sum_{i=1}^{t} P_{t,i}. \tag{14}$$

As noted in [260], the OP corresponds to the primary optimization objective defined in Definition 2.5, 2.6, and 2.7. In much of the continual learning literature, once all T tasks are completed, the final OP (OP_T) is reported, with the subscript $_T$ often omitted for brevity. In some works, OP is weighted

by the importance of tasks $\widetilde{\text{OP}} \triangleq \frac{1}{T}\sum_{i=1}^{T} w_i P_{t,i}$, where $w_i = N_i / \sum_{j=1}^{T} N_j$ represents the ratio of data. In some literature, $\widetilde{\text{OP}}$ is referred to as "example accuracy" [47], "whole accuracy" [268] or "edit success rate" in CMR [96].

Forgetting (F). Define F_t as the forgetting up to task t, which represents the largest performance drop observed throughout the training process, averaged over t training stages:

$$F_t \triangleq \frac{1}{t-1}\sum_{j=1}^{t-1}\left[\max_{l\in[t-1]}\{P_{l,j} - P_{t,j}\}\right]. \tag{15}$$

Typically, researchers report the average forgetting $F = F_T$ at the end of the entire training process. Forgetting quantifies the impact of learning new tasks on previously acquired knowledge. Ideally, a robust continual learning framework should achieve **Backward Transfer (BWT)**, where learning new tasks enhances performance on prior tasks. This enhancement is typically measured by negating the forgetting, thus indicating an improvement in performance on earlier tasks. The concepts of Forgetting and Backward Transfer underpin various evaluation metrics, such as knowledge retention [127], performance on unchanged knowledge [119], average increased perplexity (AP^{+}) [232], and test and edit retention rate in CMR [96].

Forward Transfer (FWT). Forward Transfer measures the generalization ability of the continual learning algorithms. Formally, forward transfer FWT_t up to training stage t is defined as

$$\text{FWT}_t \triangleq \frac{1}{t-1}\sum_{i=2}^{t} P_{i-1,i} - b_i, \tag{16}$$

where b_i is the baseline performance of the model evaluated on task i before undergoing continual learning. Strictly speaking, the definition of b_i is not the same as defined in the previous work [174, 260], where it is used to denote the performance of a random initialization of the model. Additionally, we extend the notation of forward transfer in the vertical direction to represent the performance improvement on downstream tasks resulting from domain-adaptive pre-training (see Table 2). Forward Transfer is alternatively referred to as temporal generalization [127] or knowledge transfer [145] in some literature.

LAnguage Model Analysis (LAMA). LAnguage Model Analysis (LAMA) is an evaluation framework designed to *probe the world knowledge* embedded in language models [225]. It converts each world fact into a cloze statement, which is then inputted into the language models to predict the correct answer. LAMA has been extended for continual pre-training, particularly for those under the temporal shifts [119, 120]. In CKL, three LAMA benchmarks are constructed for different dimensions: InvariantLAMA assesses knowledge retention on time-invariant facts, UpdatedLAMA focuses on knowledge update, and NewLAMA evaluates knowledge acquisition [120].

Forgotten / (Updated + Acquired) Ratio (FUAR). As the performance of a pre-trained LLM is decomposed into a fine-grained set in CKL [120], OP becomes a too general metric and cannot accurately reflect the balance and trade-offs of the model's behavior. To address this issue, CKL proposes a joint evaluation metric FUAR (Forgotten / (Updated + Acquired) Ratio) for continual pre-training. A FUAR value of 1 represents an equal trade-off between the knowledge forgetting and knowledge learning: for each piece of updated or acquired knowledge, one piece of time-invariant knowledge is forgotten on average. A FUAR less than 1 suggests high learning efficacy, where more than one piece of knowledge is acquired at the expense of forgetting one piece of time-invariant knowledge.

X-Delta. In TRACE [292], the authors propose a set of "X-Delta" metrics for continual instruction tuning, quantifying the forward transfer on specific abilities of LLMs. Let's denote a set of M datasets $\{X_1, X_2, \cdots, X_M\}$ for task X. The baseline performances of the pre-trained LLM evaluated on these tasks are denoted as $\{b_1^X, \cdots, b_M^X\}$. The model undergoes continuous fine-tuning on a different set of tasks, distinct from those used for evaluation. Throughout the sequential training process, the performance of the model after learning task t on evaluation tasks X_i is $R_{t,i}^X$. The X-Delta ΔR_t^X after learning task t is defined as:

$$\Delta R_t^X \triangleq \frac{1}{M}\sum_{m=1}^{M}(R_{t,i}^X - b_i^X). \tag{17}$$

In the public TRACE benchmark, the authors construct three sets of evaluation tasks to benchmark the ability of LLMs, including *general ability*, *instruction following*, and *safety* [292].

NLG Score. In continual model alignment, three prominent metrics used to evaluate different aspects of Natural language generation (NLG) are BLEU-4 [220], METEOR [14], and ROUGE-L [163]. BLEU-4[220], designed for machine translation (MT), evaluates the precision of n-grams between the machine-generated and reference texts, focusing especially on four-word sequences to gauge fluency and adequacy. METEOR [14] also targets MT but aims to improve correlation with human judgment by considering synonyms and stemming, thus providing a more nuanced assessment of translation quality. On the other hand, ROUGE-L [163] is commonly applied in summarization tasks, assessing the longest common subsequence between the generated summary and a set of reference summaries, effectively measuring the recall of essential content. Each metric has its strengths and is tailored to specific kinds of language processing tasks, reflecting different dimensions of text generation quality.

rPMS. The reference PM score (rPMS) quantifies the alignment of a model's outputs with human preferences as captured by a reference Preference Model (PM). Specifically, for a given task t, the rPMS is defined as:

$$\text{rPMS}_t = \text{rPM}(M_t, D_{\text{test},t}), \tag{18}$$

where M_t is the model being evaluated, $D_{\text{test},t}$represents the test dataset for task t, and rPMS is the function implemented by the reference PM that scores model outputs based on their alignment with human preferences. Higher rPMS values indicate that the model M_t aligns closely with human preferences, suggesting effective learning and retention of the desired behaviors across tasks in a continual learning scenario. The metric is crucial for evaluating the extent to which models can maintain or improve performance relative to human preference standards without substantial forgetting over time.

5.2 Datasets

In this section, we provide a comprehensive review of the datasets available for benchmarking continual LLMs, as illustrated in Table 4. 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 [22, 132].

Datasets for Continual Pre-Training (CPT) and Domain Adaptive Pre-Training (DAP). Current research lacks a widely recognized benchmark for evaluating continual pre-training LLMs under temporal shifts. TimeLMs utilizes a series of Twitter corpora collected until 2022, sequentially pre-training RoBERTa models quarterly [175]. CC-RecentNews, adopted as unlabeled pre-training data for LMs in CKL [120], consists of recent news and serves as a single-stage dataset. Additionally, CKL introduces InvariantLAMA, NewLAMA, and UpdatedLAMA to assess the principles of continual knowledge learning. TWiki, a dataset derived from the articles of Wikipedia between August and December 2021, is curated and cleaned in TemporalWiki [119]. This dataset facilitates the exploration of incremental learning by providing the Diffsets between neighboring snapshots. For works that study the content-level distributional shifts in CPT and DAP, researchers often resort to a similar set of publicly available datasets [172, 325, 211] to construct their own test beds for continual learning algorithms. The *DAPT dataset, developed by [92], comprises four domains: BioMed and Computer Science from S2ORC [172], News from [346], and Reviews from [101]. In *DAPT's original study, each domain undergoes individual domain adaptive pre-training stages to demonstrate the universality of DAP's effectiveness. Subsequent works, such as ELLE [232] and Recyclable Tuning [231], follow suit by employing these domains for multi-stage CPT. DEMix [91] presents another large-scale dataset, featuring eight semantic domains with over 73.8 billion tokens. Alongside the training set, it includes eight additional datasets for validating the generalization ability of LLMs. On a smaller scale, *CPT [131] and *DAS [134] datasets consist of four and eight domains, respectively, with approximately 3.12 million examples and a size of 4.16GB each. These datasets are constructed similarly to the aforementioned ones.

Datasets for Continual Instruction Tuning. Measuring the effectiveness of CIT is crucial, particularly because traditional evaluation metrics may not be suitable for LLMs: many of them are overly simplistic and fail to comprehensively assess the model's ability to learn continually. New

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 [22, 132].

Name	Type	Shift	Domain	#Stages	Scale	Sources	Applications	Comment
*TimeLMs [175]	CPT	Temporal	Social Media	8	#Examples: 123.86M	Tweets	[175]	code
CC-RecentNews [120]	CPT	Temporal	News	1	#Tokens: ∼168M	Web	[120]	code
TWiki [119]	CPT	Temporal	General Knowledge	5	#Tokens: 4.7B	Wikipedia	[119]	code
*DAPT [92]	CPT DAP	Content	Multi-Domain	4	Size: 160GB	BioMed [172], CS [172], News [346], Reviews [101]	[92] [231] [232]	code
*CPT [131]	CPT	Content	Multi-Domain	4	#Examples: 3.12M	Yelp [325], S2ORC [172], AG-News [356]	[131]	code
*DEMix [91]	CPT	Content	Multi-Domain	8	#Tokens: 73.8B	1B [39], CS [172], Legal [34], Med [172] WebText [83], RealNews [346], Reddit [17], Reviews [211]	[91]	code
*DAS [134]	CPT DAP	Content	Multi-Domain	6	Size: 4.16GB	Yelp [325], Reviews [211], Papers [172], PubMed	[134]	code
SuperNI [295]	CIT	Content	Mutli-Domain	16	#Tasks: 1616 #Examples: ∼5M	GitHub	[358, 294]	code
CITB [358]	CIT	Content	Mutli-Domain	19	#Tasks: 38	SuperNI [295]	[358]	code
CoIN [41]	CIT	Content	Multi-Domain	8	#Examples: ∼1.14M	RefCOCO [130],RefCOCO+ [190],RefCOCOg [190] ImageNet [64], VQAv2 [85], ScienceQA [179] TextVQA [261], GQA [118], VizWiz [90], OCR-VQA [200]	[41]	code
TRACE [292]	CIT	Content	Mutli-Domain	8	#Examples: 56,000	ScienceQA [179], FOMC [257], MeetingBank [111] C-STANCE [359], 20Minuten [135], CodeXGLUE [180], NumGLUE[202]	[292]	code
NATURAL-INSTRUCTION [201]	CIT	Content	Mutli-Domain	6	#Examples: 193k	CosmosQA [114], DROP [73], Essential-Terms [137] MCTACO [368], MultiRC [136], QASC [138] Quoref[59] , ROPES [164] , Winogrande [248]	[201]	code
IMDB [188]	CMA	Content	Social Media	1	Size: 217.35 MB	IMDB	[351]	code
HH-RLHF [13]	CMA	Content	General Knowledge	1	Size: 28.1 MB	Human Feedback	[351]	code
Reddit TL;DR [284]	CMA	Content	Social Media	2	Size: 19.6 GB	Reddit	[351, 352]	code
Common Sense QA [165] Reading Comprehension [165] Translation [165]	CMA	Content	Multi-Domain	6	#Examples: ∼ 41.16M	ARC Easy and Challenge [51], Race [144], PIQA [23] SQuAD [236], DROP [73] WMT 2014 French to English [24]	[165]	see sources
FEVER [278]	CMR	Content	General Knowledge	1	#Examples: 420k	Wikipedia	[60, 98]	code
VitaminC [253]	CMR	Content	General Knowledge	1	#Examples: 450k	Wikipedia	[204]	code
zsRE [147]	CMR	Content	General Knowledge	1	#Examples: 120M	Wikireading [106]	[98, 196, 197, 97, 96, 58]	code
T-rex [76]	CMR	Content	General Knowledge	1	#Examples: 11M	Dbpedia abstracts [28]	[150, 70]	code
NQ [142]	CMR	Content	General Knowledge	1	#Examples: 320k	Google queries, Wikipedia	[96]	code
CounterFact [196]	CMR	Content	General Knowledge	1	#Examples: 22k	zsRE [147]	[196, 339, 108, 58]	code
SCOTUS [36]	CMR	Temporal	Law	1	#Examples: 9.2k	Supreme Court Database	[96]	code

benchmarks and metrics are required to evaluate both the retention of old knowledge and the integration of new instructions. TRACE [292] stands as a continual learning benchmark designed specifically for LLMs, encompassing diverse tasks such as multilingual capabilities, code generation, and mathematical reasoning. CITB [358] represents another benchmark for CIT, incorporating both learning and evaluation protocols. It in addition demonstrates that replay generally yields the best performance across all methods. CoIN [41] extends the benchmark to MLLMs, incorporating a balanced and diverse set of instructions from vision-language datasets.

Datasets for Continual Model Refinement. Most datasets for continual model refinement can be categorized into two types [192]: fact checking and question answering. For fact checking, models are asked to verify the truthfulness of certain claims, typically modeled as a classification task. Key datasets include FEVER [278] (used by [60, 98]) and VitaminC [253] (used by [204]), both sourced from Wikipedia. For question answering, models are tasked with providing specific answers instead of choices. Zero-shot Relation Extraction (zsRE) [147] is the most widely employed dataset for this purpose [98, 196, 197, 97, 96, 58], alongside Natural Questions (NQ) [142] and T-rex [76]. [196] adapted zsRE with additional counterfactuals to create the more challenging CounterFact dataset, used by [339, 108, 58]. Beyond these two categories, SCOTUS [36] is also utilized [96] in the assessment of continual model refinement through a document classification task for U.S. Supreme Court cases into 11 topics.

Datasets for Continual Model Alignment. In the domain of reinforcement learning with human feedback (RLHF), several datasets are commonly employed across different studies to evaluate the adaptation and effectiveness of models under varying scenarios and continuous learning conditions. The IMDB [188] and HH-RLHF [13] dataset, as introduced in [351] within their study on continual learning through optimal policy fitting, leverages data gathered from interactive RL scenarios to model human preferences dynamically. Similarly, the Reddit TL;DR dataset [284] used by [352, 351] is focused on text summarization, providing a robust platform for testing the longevity and adaptability of learning algorithms under evolving conditions. Lastly, Common Sense QA [51, 144, 23], Reading Comprehension [236, 73], and Translation [24], which are utilized in [165] are selected to assess the challenges of aligning RL agents with human expectations without incurring significant performance penalties. Each of these datasets is pivotal in advancing the understanding of continual learning and the interplay between human feedback and machine learning adaptation.

Datasets for Continual Multimodal Large Language Models. Following LLaVA [167], many MLLMs adopt the pattern of instruction tuning to enable assessing alignment with human intention and knowledge preservation for reasoning. Thus, traditional tasks like image classification can be transformed to VQA tasks to evaluate the ability of MLLMs, which are otherwise challenging to assess using conventional methods. Several benchmarks have been proposed to evaluate the CL method for MLLMs. MCIT [100] proposes the first continual instruction tuning benchmarks, Benchmark1 and Benchmark2. The difference between benchmark1 and benchmark2 is that benchmark2 includes Multi-task Joint Instruction Tuning, which aims to explore whether multi-task joint instruction tuning improves the model's continual learning ability. [349] proposes EMT, the first classification evaluation framework to investigate catastrophic forgetting in MLLMs. [41] presents a comprehensive benchmark CoIN, spanning 8 task categories and evaluating MLLMs from two perspectives: Instruction Following and General Knowledge, which assess the alignment with human intention and knowledge preserved for reasoning, respectively. [362] constructs two datasets, UPMC-Food101-CMML and MM-IMDb-CMML to benchmark the novel CMML task, which means the data of certain modalities is missing during continual fine-tuning. UPMC-Food101-CMM contains 101 food categories and 61,142 training, 6,846 validation, and 22,716 test image-text pairs. MM-IMDb-CMML is a multi-label classification dataset across 27 distinct movie genres, consisting of 15,552 training, 2,608 validation and 7,799 test image-text pairs.

6 Discussion

In this section, we delve into the intersection of conventional computational patterns in continual learning and the training and deployment of large language models (LLMs). We begin by examining intriguing properties that arise during continual learning with LLMs. Next, we explore the evolving roles of three types of incremental learning within the context of LLMs. Following this, we contrast the roles of memory in continual LLMs with those in traditional continual learning. Finally, we conclude with a concise overview of promising directions for future research in this area.

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 [133, 274, 183, 365, 195], there is a notable lack of exploration into other intriguing properties of LLMs when trained continually. While investigations into the emergent capabilities of continuously trained LLMs have attracted attentions from the community to a certain extent, they remain relatively limited. For instance, in [332], 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 memory replay and more complex structured learning environments as model parameters scale up.

6.2 Conventional Types of Incremental Learning

As mentioned in Section 2.2, three types of incremental learning are prevalent [282]. Among them, class-incremental learning (CIL) has historically attracted significant attention from the community [239, 316]. 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 [258, 57, 33]. For instance, in Vocabulary-Aware Label Generation (VAG), CIL is redefined as the task of continual label generation. This approach utilizes a pre-trained encoder-decoder language model to generate class labels [258]. Meanwhile, in the Generative Multi-modal Model (GMM) for CIL [33], image patches and prompts are concatenated and fed into the language model to generate classification results.

However, the declining attention to the conventional CIL paradigm does not suggest that these techniques are not impactful in the field of continual learning for LLMs. Nonetheless, many current research endeavors unwittingly employ such techniques, indicating their widespread adoption in various applications. For example, techniques such as vocabulary expansion [6, 55] can be seen as an extension of expanding the classification head in CIL. These CIL techniques can be further integrated into systems like Lifelong-MoE [46], where adding a new expert to the transformer blocks requires updating the gating function to include the routing of the newly added expert. The EProj framework, described in [100], employs a similar architecture, incorporating linear projectors for new domains alongside a selector module trained to route these projectors. Since the aforementioned sub-modules operate on the principles of CIL, previously validated techniques can be directly applied.

The importance of domain-incremental learning (DIL) is self-evident, given the shared task definition and input-output format in continual pre-training (CPT) and domain-adaptive pre-training (DAP). As dynamically expanding token vocabularies can pose additional challenges, it is natural to focus on understanding distributional shifts within the input corpus while keeping the vocabulary fixed. On another front, task-incremental learning (TIL) attracts significant interest due to its potential for personalizing LLM services. For instance, users may desire options for selecting domain-specific experts, thereby making task IDs available throughout inference time [112, 309]. Additionally, TIL plays a crucial role in instruction tuning, where instructions can be seen as natural-language-encoded task information [255, 113, 205, 103, 337, 291, 363, 294]. 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. The setting of continual learning with limited memory size has garnered significant attention from the community. 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 [283], 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) [323, 126]. 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 the number of FLOPs [227, 298].

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 [245, 52, 311, 10], leveraging the generative capabilities of LLMs to produce pseudo-examples for replay [230], and implementing regularization techniques in the parameter space [134, 243]. 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 [31, 189, 228], 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 [329, 297, 25].

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 [288], 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 [260] 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. Computational resources for training large-scale LLMs are often limited. 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 [166, 245, 331], there is ongoing exploration of efficient replay for continual LLMs. For example, KPIG [103] enhances replay efficiency by calculating Key-part Information Gain (KPIG) on masked segments, enabling the dynamic selection of replay data. In [126], a pioneering effort introduces a forgetting forecasting mechanism based on output changes during adaptation, later used for selective replay in continual model refinement (CMR). It has been verified in this work that filtering replay samples based on their tendency to forget significantly improves knowledge retention rates for continual LLMs. However, more sophisticated and accurate data mixing strategies and efficient replay sample selection mechanisms are much needed, e.g., a dynamic data mixing ratio throughout the training process. Hence we mark this practical direction of efficient replay for LLMs 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. For instance, consider a scenario where a supplier collects data from customers under their consent and continually utilizes this data to update LLMs. However, if some users later revoke their consent, the knowledge acquired by the trained model from that portion of data must also be revoked. With a continually pre-trained large-scale LLM, the only solution is to roll back to a previous model version predating the inclusion of these users' data and retrain the model from that point onward. This example of "machine unlearning" [26, 210] vividly 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 [58] suggests integrating the Kanerva Machine [317] 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. Additionally, other memory systems like Hopfield Networks [238, 226] hold promise for future investigation.

Continual LLMs with Custom Preferences. Customizing user preferences is critical for LLMs, especially 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 [178]. 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] H. Abdine, M. Chatzianastasis, C. Bouyioukos, and M. Vazirgiannis. Prot2text: Multimodal protein's function generation with GNNs and transformers. In *Deep Generative Models for Health Workshop NeurIPS 2023*, 2023.

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] M. Agarwal, Y. Shen, B. Wang, Y. Kim, and J. Chen. Structured code representations enable data-efficient adaptation of code language models, 2024.

[4] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018.

[5] R. Aljundi, P. Chakravarty, and T. Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3366–3375, 2017.

[6] S. Amba Hombaiah, T. Chen, M. Zhang, M. Bendersky, and M. Najork. Dynamic language models for continuously evolving content. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2514–2524, 2021.

[7] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[8] D. Araci. Finbert: Financial sentiment analysis with pre-trained language models, 2019.

[9] G. Attanasio, D. Nozza, F. Bianchi, and D. Hovy. Is it worth the (environmental) cost? limited evidence for temporal adaptation via continuous training, 2023.

[10] Z. Azerbayev, H. Schoelkopf, K. Paster, M. D. Santos, S. McAleer, A. Q. Jiang, J. Deng, S. Biderman, and S. Welleck. Llemma: An open language model for mathematics. *CoRR*, abs/2310.10631, 2023.

[11] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[12] X. Bai, J. Shang, Y. Sun, and N. Balasubramanian. Enhancing continual learning with global prototypes: Counteracting negative representation drift, 2023.

[13] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[14] S. Banerjee and A. Lavie. 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*, pages 65–72, 2005.

[15] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, and J. Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8218–8227, June 2021.

[16] Z. Bao, W. Chen, S. Xiao, K. Ren, J. Wu, C. Zhong, J. Peng, X. Huang, and Z. Wei. Disc-medllm: Bridging general large language models and real-world medical consultation, 2023.

[17] J. Baumgartner, S. Zannettou, B. Keegan, M. Squire, and J. Blackburn. The pushshift reddit dataset, 2020.

[18] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

[19] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery.

[20] Z. Bi, N. Zhang, Y. Xue, Y. Ou, D. Ji, G. Zheng, and H. Chen. Oceangpt: A large language model for ocean science tasks. *CoRR*, abs/2310.02031, 2023.

[21] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[22] M. Biesialska, K. Biesialska, and M. R. Costa-jussà. Continual lifelong learning in natural language processing: A survey. In D. Scott, N. Bel, and C. Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics.

[23] Y. Bisk, R. Zellers, J. Gao, Y. Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

[24] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, et al. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58, 2014.

[25] J. Bornschein, Y. Li, and A. Rannen-Triki. Transformers for supervised online continual learning, 2024.

[26] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot. Machine unlearning, 2020.

[27] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

[29] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

[30] L. Caccia, R. Aljundi, N. Asadi, T. Tuytelaars, J. Pineau, and E. Belilovsky. New insights on reducing abrupt representation change in online continual learning. *arXiv preprint arXiv:2104.05025*, 2021.

[31] Z. Cai, O. Sener, and V. Koltun. Online continual learning with natural distribution shifts: An empirical study with visual data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8281–8290, 2021.

[32] H. Cao, Z. Liu, X. Lu, Y. Yao, and Y. Li. Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. *CoRR*, abs/2311.16208, 2023.

[33] X. Cao, H. Lu, L. Huang, X. Liu, and M.-M. Cheng. Generative multi-modal models are good class incremental learners. *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2024.

[34] Caselaw Access Project. Caselaw access project, 2018.

[35] Y. Chai, S. Wang, C. Pang, Y. Sun, H. Tian, and H. Wu. ERNIE-code: Beyond English-centric cross-lingual pretraining for programming languages. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10628–10650, Toronto, Canada, July 2023. Association for Computational Linguistics.

[36] I. Chalkidis, T. Pasini, S. Zhang, L. Tomada, S. F. Schwemer, and A. Søgaard. Fairlex: A multilingual benchmark for evaluating fairness in legal text processing. *arXiv preprint arXiv:2203.07228*, 2022.

[37] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019.

[38] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

[39] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson. One billion word benchmark for measuring progress in statistical language modeling, 2014.

[40] B. Chen, X. Cheng, P. Li, Y. Geng, J. Gong, S. Li, Z. Bei, X. Tan, B. Wang, X. Zeng, C. Liu, A. Zeng, Y. Dong, J. Tang, and L. Song. xtrimopglm: Unified 100b-scale pre-trained transformer for deciphering the language of protein. *CoRR*, abs/2401.06199, 2024.

[41] C. Chen, J. Zhu, X. Luo, H. Shen, L. Gao, and J. Song. Coin: A benchmark of continual instruction tuning for multimodel large language model, 2024.

[42] J. Chen, X. Wang, A. Gao, F. Jiang, S. Chen, H. Zhang, D. Song, W. Xie, C. Kong, J. Li, X. Wan, H. Li, and B. Wang. Huatuogpt-ii, one-stage training for medical adaption of llms. *CoRR*, abs/2311.09774, 2023.

[43] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating large language models trained on code, 2021.

[44] S. Chen, Y. Hou, Y. Cui, W. Che, T. Liu, and X. Yu. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7870–7881, Online, Nov. 2020. Association for Computational Linguistics.

[45] S. Chen, B. H. Kann, M. B. Foote, H. J. Aerts, G. K. Savova, R. H. Mak, and D. S. Bitterman. The utility of chatgpt for cancer treatment information. *medRxiv*, 2023.

[46] W. Chen, Y. Zhou, N. Du, Y. Huang, J. Laudon, Z. Chen, and C. Cui. Lifelong language pretraining with distribution-specialized experts. In *International Conference on Machine Learning*, pages 5383–5395. PMLR, 2023.

[47] Y. Chen, S. Zhang, G. Qi, and X. Guo. 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.

[48] Z. Chen and B. Liu. *Lifelong machine learning*, volume 1. Springer.

[49] D. Cheng, S. Huang, and F. Wei. Adapting large language models via reading comprehension, 2024.

[50] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[51] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[52] P. Colombo, T. P. Pires, M. Boudiaf, D. Culver, R. Melo, C. Corro, A. F. T. Martins, F. Esposito, V. L. Raposo, S. Morgado, and M. Desa. Saullm-7b: A pioneering large language model for law, 2024.

[53] T. Computer. Redpajama: an open dataset for training large language models, 2023.

[54] A. O. Constantinescu, J. X. O'Reilly, and T. E. Behrens. Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352(6292):1464–1468, 2016.

[55] A. Cossu, T. Tuytelaars, A. Carta, L. Passaro, V. Lomonaco, and D. Bacciu. Continual pre-training mitigates forgetting in language and vision, 2022.

[56] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.

[57] M. D'Alessandro, A. Alonso, E. Calabrés, and M. Galar. Multimodal parameter-efficient few-shot class incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3393–3403, October 2023.

[58] P. Das, S. Chaudhury, E. Nelson, I. Melnyk, S. Swaminathan, S. Dai, A. Lozano, G. Kollias, V. Chenthamarakshan, S. Dan, et al. Larimar: Large language models with episodic memory control. *arXiv preprint arXiv:2403.11901*, 2024.

[59] P. Dasigi, N. F. Liu, A. Marasović, N. A. Smith, and M. Gardner. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[60] N. De Cao, W. Aziz, and I. Titov. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*, 2021.

[61] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

[62] DeepSeek-AI, :, X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu, H. Gao, K. Gao, W. Gao, R. Ge, K. Guan, D. Guo, J. Guo, G. Hao, Z. Hao, Y. He, W. Hu, P. Huang, E. Li, G. Li, J. Li, Y. Li, Y. K. Li, W. Liang, F. Lin, A. X. Liu, B. Liu, W. Liu, X. Liu, X. Liu, Y. Liu, H. Lu, S. Lu, F. Luo, S. Ma, X. Nie, T. Pei, Y. Piao, J. Qiu, H. Qu, T. Ren, Z. Ren, C. Ruan, Z. Sha, Z. Shao, J. Song, X. Su, J. Sun, Y. Sun, M. Tang, B. Wang, P. Wang, S. Wang, Y. Wang, Y. Wang, T. Wu, Y. Wu, X. Xie, Z. Xie, Z. Xie, Y. Xiong, H. Xu, R. X. Xu, Y. Xu, D. Yang, Y. You, S. Yu, X. Yu, B. Zhang, H. Zhang, L. Zhang, L. Zhang, M. Zhang, M. Zhang, W. Zhang, Y. Zhang, C. Zhao, Y. Zhao, S. Zhou, S. Zhou, Q. Zhu, and Y. Zou. Deepseek llm: Scaling open-source language models with longtermism, 2024.

[63] C. Deng, T. Zhang, Z. He, Y. Xu, Q. Chen, Y. Shi, L. Fu, W. Zhang, X. Wang, C. Zhou, Z. Lin, and J. He. K2: A foundation language model for geoscience knowledge understanding and utilization, 2023.

[64] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[65] Y. Deng, W. Lei, W. Lam, and T.-S. Chua. A survey on proactive dialogue systems: Problems, methods, and prospects. *arXiv preprint arXiv:2305.02750*, 2023.

[66] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[67] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[68] B. Dhingra, J. R. Cole, J. M. Eisenschlos, D. Gillick, J. Eisenstein, and W. W. Cohen. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273, 2022.

[69] P. Di, J. Li, H. Yu, W. Jiang, W. Cai, Y. Cao, C. Chen, D. Chen, H. Chen, L. Chen, et al. Codefuse-13b: A pretrained multi-lingual code large language model. *arXiv preprint arXiv:2310.06266*, 2023.

[70] Q. Dong, D. Dai, Y. Song, J. Xu, Z. Sui, and L. Li. Calibrating factual knowledge in pretrained language models. *arXiv preprint arXiv:2210.03329*, 2022.

[71] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[72] N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat, B. Zoph, L. Fedus, M. P. Bosma, Z. Zhou, T. Wang, E. Wang, K. Webster, M. Pellat, K. Robinson, K. Meier-Hellstern, T. Duke, L. Dixon, K. Zhang, Q. Le, Y. Wu, Z. Chen, and C. Cui. GLaM: Efficient scaling of language models with mixture-of-experts. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR, 17–23 Jul 2022.

[73] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*, 2019.

[74] S. Ebrahimi, M. Elhoseiny, T. Darrell, and M. Rohrbach. Uncertainty-guided continual learning with bayesian neural networks. *arXiv preprint arXiv:1906.02425*, 2019.

[75] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell, and M. Rohrbach. Adversarial continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 386–402. Springer, 2020.

[76] H. Elsahar, P. Vougiouklis, A. Remaci, C. Gravier, J. Hare, F. Laforest, and E. Simperl. 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)*, 2018.

[77] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[78] J. Gao, R. Pi, J. Zhang, J. Ye, W. Zhong, Y. Wang, L. Hong, J. Han, H. Xu, Z. Li, and L. Kong. G-llava: Solving geometric problem with multi-modal large language model. *CoRR*, abs/2312.11370, 2023.

[79] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[80] S. Garg, S. Dutta, M. Dalirrooyfard, A. Schneider, and Y. Nevmyvaka. In- or out-of-distribution detection via dual divergence estimation. In R. J. Evans and I. Shpitser, editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 635–646. PMLR, 31 Jul–04 Aug 2023.

[81] S. Garg, M. Farajtabar, H. Pouransari, R. Vemulapalli, S. Mehta, O. Tuzel, V. Shankar, and F. Faghri. Tic-clip: Continual training of clip models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[82] E. Gogoulou, T. Lesort, M. Boman, and J. Nivre. Continual learning under language shift, 2024.

[83] A. Gokaslan and V. Cohen. Openwebtext corpus, 2019.

[84] Z. Gou, Z. Shao, Y. Gong, yelong shen, Y. Yang, M. Huang, N. Duan, and W. Chen. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In *The Twelfth International Conference on Learning Representations*, 2024.

[85] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering, 2017.

[86] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23, Oct. 2021.

[87] D. Guo, Q. Zhu, D. Yang, Z. Xie, K. Dong, W. Zhang, G. Chen, X. Bi, Y. Wu, Y. K. Li, F. Luo, Y. Xiong, and W. Liang. Deepseek-coder: When the large language model meets programming – the rise of code intelligence, 2024.

[88] Z. Guo and Y. Hua. Continuous training and fine-tuning for domain-specific language models in medical question answering, 2023.

[89] K. Gupta, B. Thérien, A. Ibrahim, M. L. Richter, Q. Anthony, E. Belilovsky, I. Rish, and T. Lesort. Continual pre-training of large language models: How to (re)warm your model?, 2023.

[90] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people, 2018.

[91] S. Gururangan, M. Lewis, A. Holtzman, N. A. Smith, and L. Zettlemoyer. DEMix layers: Disentangling domains for modular language modeling. In M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576, Seattle, United States, July 2022. Association for Computational Linguistics.

[92] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics.

[93] R. Han, X. Ren, and N. Peng. ECONET: Effective continual pretraining of language models for event temporal reasoning. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5367–5380, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[94] T. Han, L. C. Adams, J. Papaioannou, P. Grundmann, T. Oberhauser, A. Löser, D. Truhn, and K. K. Bressem. Medalpaca - an open-source collection of medical conversational AI models and training data. *CoRR*, abs/2304.08247, 2023.

[95] Y. Hao, L. Dong, F. Wei, and K. Xu. Visualizing and understanding the effectiveness of BERT. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4143–4152, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[96] T. Hartvigsen, S. Sankaranarayanan, H. Palangi, Y. Kim, and M. Ghassemi. Aging with grace: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems*, 2023.

[97] P. Hase, M. Bansal, B. Kim, and A. Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. *Knowledge Editing in Language Models*, 2023.

[98] P. Hase, M. Diab, A. Celikyilmaz, X. Li, Z. Kozareva, V. Stoyanov, M. Bansal, and S. Iyer. Do language models have beliefs? methods for detecting, updating, and visualizing model beliefs. *arXiv preprint arXiv:2111.13654*, 2021.

[99] T. L. Hayes and C. Kanan. Lifelong machine learning with deep streaming linear discriminant analysis, 2020.

[100] J. He, H. Guo, M. Tang, and J. Wang. Continual instruction tuning for large multimodal models, 2023.

[101] R. He and J. McAuley. 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*, WWW '16, page 507–517, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee.

[102] T. He, J. Liu, K. Cho, M. Ott, B. Liu, J. Glass, and F. Peng. Analyzing the forgetting problem in pretrain-finetuning of open-domain dialogue response models. In P. Merlo, J. Tiedemann, and R. Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1121–1133, Online, Apr. 2021. Association for Computational Linguistics.

[103] Y. He, X. Huang, M. Tang, L. Meng, X. Li, W. Lin, W. Zhang, and Y. Gao. Don't half-listen: Capturing key-part information in continual instruction tuning, 2024.

[104] D. Hendrycks, C. Burns, S. Basart, A. Critch, J. Li, D. Song, and J. Steinhardt. Aligning ai with shared human values, 2023.

[105] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[106] D. Hewlett, A. Lacoste, L. Jones, I. Polosukhin, A. Fandrianto, J. Han, M. Kelcey, and D. Berthelot. Wikireading: A novel large-scale language understanding task over wikipedia. *arXiv preprint arXiv:1608.03542*, 2016.

[107] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[108] C. Hu, P. Cao, Y. Chen, K. Liu, and J. Zhao. Wilke: Wise-layer knowledge editor for lifelong knowledge editing, 2024.

[109] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[110] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[111] Y. Hu, T. Ganter, H. Deilamsalehy, F. Dernoncourt, H. Foroosh, and F. Liu. Meetingbank: A benchmark dataset for meeting summarization, 2023.

[112] C. Huang, Q. Liu, B. Y. Lin, T. Pang, C. Du, and M. Lin. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*, 2023.

[113] J. Huang, L. Cui, A. Wang, C. Yang, X. Liao, L. Song, J. Yao, and J. Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal, 2024.

[114] L. Huang, R. L. Bras, C. Bhagavatula, and Y. Choi. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning, 2019.

[115] Q. Huang, M. Tao, Z. An, C. Zhang, C. Jiang, Z. Chen, Z. Wu, and Y. Feng. Lawyer llama technical report. *arXiv preprint arXiv:2305.15062*, 2023.

[116] Q. Huang, M. Tao, C. Zhang, Z. An, C. Jiang, Z. Chen, Z. Wu, and Y. Feng. Lawyer llama. https://github.com/AndrewZhe/lawyer-llama, 2023.

[117] Z. Huang, Y. Shen, X. Zhang, J. Zhou, W. Rong, and Z. Xiong. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*, 2023.

[118] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering, 2019.

[119] J. Jang, S. Ye, C. Lee, S. Yang, J. Shin, J. Han, G. Kim, and M. Seo. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. 2022.

[120] J. Jang, S. Ye, S. Yang, J. Shin, J. Han, G. Kim, S. J. Choi, and M. Seo. Towards continual knowledge learning of language models. In *ICLR*, 2022.

[121] K. Jeblick, B. Schachtner, J. Dexl, A. Mittermeier, A. T. Stüber, J. Topalis, T. Weber, P. Wesp, B. Sabel, J. Ricke, and M. Ingrisch. Chatgpt makes medicine easy to swallow: An exploratory case study on simplified radiology reports, 2022.

[122] J. Ji, T. Qiu, B. Chen, B. Zhang, H. Lou, K. Wang, Y. Duan, Z. He, J. Zhou, Z. Zhang, F. Zeng, K. Y. Ng, J. Dai, X. Pan, A. O'Gara, Y. Lei, H. Xu, B. Tse, J. Fu, S. McAleer, Y. Yang, Y. Wang, S.-C. Zhu, Y. Guo, and W. Gao. Ai alignment: A comprehensive survey, 2024.

[123] S. Jiang, Y. Wang, and Y. Wang. Selfevolve: A code evolution framework via large language models, 2023.

[124] Y. Jiang, Z. Pan, X. Zhang, S. Garg, A. Schneider, Y. Nevmyvaka, and D. Song. Empowering time series analysis with large language models: A survey, 2024.

[125] Z. Jiang, Z. Sun, W. Shi, P. Rodriguez, C. Zhou, G. Neubig, X. V. Lin, W. tau Yih, and S. Iyer. Instruction-tuned language models are better knowledge learners, 2024.

[126] X. Jin and X. Ren. What will my model forget? forecasting forgotten examples in language model refinement, 2024.

[127] X. Jin, D. Zhang, H. Zhu, W. Xiao, S.-W. Li, X. Wei, A. Arnold, and X. Ren. Lifelong pretraining: Continually adapting language models to emerging corpora. In A. Fan, S. Ilic, T. Wolf, and M. Gallé, editors, *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 1–16, virtual+Dublin, May 2022. Association for Computational Linguistics.

[128] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, A. J. Hudspeth, S. Mack, et al. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.

[129] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[130] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg. ReferItGame: Referring to objects in photographs of natural scenes. In A. Moschitti, B. Pang, and W. Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[131] Z. Ke, H. Lin, Y. Shao, H. Xu, L. Shu, and B. Liu. Continual training of language models for few-shot learning. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10205–10216, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.

[132] Z. Ke and B. Liu. Continual learning of natural language processing tasks: A survey, 2023.

[133] Z. Ke, B. Liu, N. Ma, H. Xu, and S. Lei. Achieving forgetting prevention and knowledge transfer in continual learning. In *NeurIPS*, 2021.

[134] Z. Ke, Y. Shao, H. Lin, T. Konishi, G. Kim, and B. Liu. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*, 2022.

[135] T. Kew, M. Kostrzewa, and S. Ebling. 20 minuten: A multi-task news summarisation dataset for German. In H. Ghorbel, M. Sokhn, M. Cieliebak, M. Hürlimann, E. de Salis, and J. Guerne, editors, *Proceedings of the 8th edition of the Swiss Text Analytics Conference*, pages 1–13, Neuchatel, Switzerland, June 2023. Association for Computational Linguistics.

[136] D. Khashabi, S. Chaturvedi, M. Roth, S. Upadhyay, and D. Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In M. Walker, H. Ji, and A. Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[137] D. Khashabi, T. Khot, A. Sabharwal, and D. Roth. Learning what is essential in questions. In R. Levy and L. Specia, editors, *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 80–89, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics.

[138] T. Khot, P. Clark, M. Guerquin, P. Jansen, and A. Sabharwal. Qasc: A dataset for question answering via sentence composition, 2020.

[139] G. Kim, C. Xiao, T. Konishi, Z. Ke, and B. Liu. A theoretical study on solving continual learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 5065–5079. Curran Associates, Inc., 2022.

[140] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[141] P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand, Sept. 13-15 2005.

[142] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[143] C. C. T. Kwok, O. Etzioni, and D. S. Weld. Scaling question answering to the web. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, page 150–161, New York, NY, USA, 2001. Association for Computing Machinery.

[144] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

[145] A. Lazaridou, A. Kuncoro, E. Gribovskaya, D. Agrawal, A. Liska, T. Terzi, M. Gimenez, C. de Masson d'Autume, T. Kocisky, S. Ruder, et al. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34:29348–29363, 2021.

[146] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[147] O. Levy, M. Seo, E. Choi, and L. Zettlemoyer. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*, 2017.

[148] B. Li, Y. Zhang, L. Chen, J. Wang, J. Yang, and Z. Liu. Otter: A multi-modal model with in-context instruction tuning, 2023.

[149] C.-A. Li and H.-Y. Lee. Examining forgetting in continual pre-training of aligned large language models, 2024.

[150] D. Li, A. S. Rawat, M. Zaheer, X. Wang, M. Lukasik, A. Veit, F. Yu, and S. Kumar. Large language models with controllable working memory. *arXiv preprint arXiv:2211.05110*, 2022.

[151] H. Li, Y. Zhang, F. Koto, Y. Yang, H. Zhao, Y. Gong, N. Duan, and T. Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*, 2023.

[152] J. Li, Y. Bian, G. Wang, Y. Lei, D. Cheng, Z. Ding, and C. Jiang. Cfgpt: Chinese financial assistant with large language model, 2023.

[153] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[154] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao. Videochat: Chat-centric video understanding, 2024.

[155] K. Li, Q. Hu, X. Zhao, H. Chen, Y. Xie, T. Liu, Q. Xie, and J. He. Instructcoder: Instruction tuning large language models for code editing, 2024.

[156] L. Li and X. Qiu. CONTINUAL MODEL EVOLVEMENT WITH INNER-PRODUCT RESTRICTION, 2023.

[157] M. Li, Y. Zhang, Z. Li, J. Chen, L. Chen, N. Cheng, J. Wang, T. Zhou, and J. Xiao. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *ArXiv*, abs/2308.12032, 2023.

[158] R. Li, L. B. Allal, Y. Zi, N. Muennighoff, D. Kocetkov, C. Mou, M. Marone, C. Akiki, J. Li, J. Chim, Q. Liu, E. Zheltonozhskii, T. Y. Zhuo, T. Wang, O. Dehaene, M. Davaadorj, J. Lamy-Poirier, J. Monteiro, O. Shliazhko, N. Gontier, N. Meade, A. Zebaze, M.-H. Yee, L. K. Umapathi, J. Zhu, B. Lipkin, M. Oblokulov, Z. Wang, R. Murthy, J. Stillerman, S. S. Patel, D. Abulkhanov, M. Zocca, M. Dey, Z. Zhang, N. Fahmy, U. Bhattacharyya, W. Yu, S. Singh, S. Luccioni, P. Villegas, M. Kunakov, F. Zhdanov, M. Romero, T. Lee, N. Timor, J. Ding, C. Schlesinger, H. Schoelkopf, J. Ebert, T. Dao, M. Mishra, A. Gu, J. Robinson, C. J.

Anderson, B. Dolan-Gavitt, D. Contractor, S. Reddy, D. Fried, D. Bahdanau, Y. Jernite, C. M. Ferrandis, S. Hughes, T. Wolf, A. Guha, L. von Werra, and H. de Vries. Starcoder: may the source be with you!, 2023.

[159] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, Aug. 2021. Association for Computational Linguistics.

[160] Y. Li, Z. Li, K. Zhang, R. Dan, S. Jiang, and Y. Zhang. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Cureus*, 15(6), 2023.

[161] Z. Li and D. Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[162] B. Y. Lin, S. Wang, X. Lin, R. Jia, L. Xiao, X. Ren, and S. Yih. On continual model refinement in out-of-distribution data streams. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3128–3139, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[163] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[164] K. Lin, O. Tafjord, P. Clark, and M. Gardner. Reasoning over paragraph effects in situations, 2019.

[165] Y. Lin, H. Lin, W. Xiong, S. Diao, J. Liu, J. Zhang, R. Pan, H. Wang, W. Hu, H. Zhang, H. Dong, R. Pi, H. Zhao, N. Jiang, H. Ji, Y. Yao, and T. Zhang. Mitigating the alignment tax of rlhf, 2024.

[166] Z. Lin, C. Deng, L. Zhou, T. Zhang, Y. Xu, Y. Xu, Z. He, Y. Shi, B. Dai, Y. Song, B. Zeng, Q. Chen, T. Shi, T. Huang, Y. Xu, S. Wang, L. Fu, W. Zhang, J. He, C. Ma, Y. Zhu, X. Wang, and C. Zhou. Geogalactica: A scientific large language model in geoscience, 2023.

[167] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning, 2023.

[168] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. A. Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

[169] J. Liu, P. Zhou, Y. Hua, D. Chong, Z. Tian, A. Liu, H. Wang, C. You, Z. Guo, L. Zhu, and M. L. Li. Benchmarking large language models on cmexam – a comprehensive chinese medical exam dataset, 2023.

[170] Y. Liu, R. J. Dolan, Z. Kurth-Nelson, and T. E. Behrens. Human replay spontaneously reorganizes experience. *Cell*, 178(3):640–652, 2019.

[171] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[172] K. Lo, L. L. Wang, M. Neumann, R. Kinney, and D. Weld. S2ORC: The semantic scholar open research corpus. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online, July 2020. Association for Computational Linguistics.

[173] V. Lomonaco, D. Maltoni, and L. Pellegrini. Rehearsal-free continual learning over small non-i.i.d. batches, 2020.

[174] D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[175] D. Loureiro, F. Barbieri, L. Neves, L. Espinosa Anke, and J. Camacho-collados. TimeLMs: Diachronic language models from Twitter. In V. Basile, Z. Kozareva, and S. Stajner, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[176] A. Lozhkov, R. Li, L. B. Allal, F. Cassano, J. Lamy-Poirier, N. Tazi, A. Tang, D. Pykhtar, J. Liu, Y. Wei, T. Liu, M. Tian, D. Kocetkov, A. Zucker, Y. Belkada, Z. Wang, Q. Liu, D. Abulkhanov, I. Paul, Z. Li, W.-D. Li, M. Risdal, J. Li, J. Zhu, T. Y. Zhuo, E. Zheltonozhskii, N. O. O. Dade, W. Yu, L. Krauß, N. Jain, Y. Su, X. He, M. Dey, E. Abati, Y. Chai, N. Muennighoff, X. Tang, M. Oblokulov, C. Akiki, M. Marone, C. Mou, M. Mishra, A. Gu, B. Hui, T. Dao, A. Zebaze, O. Dehaene, N. Patry, C. Xu, J. McAuley, H. Hu, T. Scholak, S. Paquet, J. Robinson, C. J. Anderson, N. Chapados, M. Patwary, N. Tajbakhsh, Y. Jernite, C. M. Ferrandis, L. Zhang, S. Hughes, T. Wolf, A. Guha, L. von Werra, and H. de Vries. Starcoder 2 and the stack v2: The next generation, 2024.

[177] D. Lu, H. Wu, J. Liang, Y. Xu, Q. He, Y. Geng, M. Han, Y. Xin, and Y. Xiao. Bbt-fin: Comprehensive construction of chinese financial domain pre-trained language model, corpus and benchmark. *CoRR*, abs/2302.09432, 2023.

[178] P. Lu, M. Caprio, E. Eaton, and I. Lee. Ibcl: Zero-shot model generation for task trade-offs in continual learning, 2023.

[179] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering, 2022.

[180] S. Lu, D. Guo, S. Ren, J. Huang, A. Svyatkovskiy, A. Blanco, C. Clement, D. Drain, D. Jiang, D. Tang, G. Li, L. Zhou, L. Shou, L. Zhou, M. Tufano, M. Gong, M. Zhou, N. Duan, N. Sundaresan, S. K. Deng, S. Fu, and S. Liu. Codexglue: A machine learning benchmark dataset for code understanding and generation, 2021.

[181] H. Luo, Q. Sun, C. Xu, P. Zhao, J. Lou, C. Tao, X. Geng, Q. Lin, S. Chen, and D. Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

[182] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, and T.-Y. Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6), Sept. 2022.

[183] Y. Luo, Z. Yang, X. Bai, F. Meng, J. Zhou, and Y. Zhang. Investigating forgetting in pre-trained representations through continual learning, 2023.

[184] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2023.

[185] Y. Luo, J. Zhang, S. Fan, K. Yang, Y. Wu, M. Qiao, and Z. Nie. Biomedgpt: Open multimodal generative pre-trained transformer for biomedicine. *arXiv preprint arXiv:2308.09442*, 2023.

[186] Z. Luo, C. Xu, P. Zhao, Q. Sun, X. Geng, W. Hu, C. Tao, J. Ma, Q. Lin, and D. Jiang. Wizardcoder: Empowering code large language models with evol-instruct, 2023.

[187] S. Ma, S. Huang, S. Huang, X. Wang, Y. Li, H.-T. Zheng, P. Xie, F. Huang, and Y. Jiang. Ecomgpt-ct: Continual pre-training of e-commerce large language models with semi-structured data, 2023.

[188] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

[189] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

[190] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions, 2016.

[191] L. Martin, N. Whitehouse, S. Yiu, L. Catterson, and R. Perera. Better call gpt, comparing large language models against lawyers, 2024.

[192] V. Mazzia, A. Pedrani, A. Caciolai, K. Rottmann, and D. Bernardi. A survey on knowledge editing of neural networks. *arXiv preprint arXiv:2310.19704*, 2023.

[193] D. McCaffary. Towards continual task learning in artificial neural networks: current approaches and insights from neuroscience. *arXiv preprint arXiv:2112.14146*, 2021.

[194] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly. 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):419, 1995.

[195] S. V. Mehta, D. Patil, S. Chandar, and E. Strubell. An empirical investigation of the role of pre-training in lifelong learning. *Journal of Machine Learning Research*, 24(214):1–50, 2023.

[196] K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[197] K. Meng, A. S. Sharma, A. Andonian, Y. Belinkov, and D. Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022.

[198] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

[199] S. I. Mirzadeh, A. Chaudhry, D. Yin, H. Hu, R. Pascanu, D. Gorur, and M. Farajtabar. Wide neural networks forget less catastrophically. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15699–15717. PMLR, 17–23 Jul 2022.

[200] A. Mishra, S. Shekhar, A. K. Singh, and A. Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, 2019.

[201] S. Mishra, D. Khashabi, C. Baral, and H. Hajishirzi. Natural instructions: Benchmarking generalization to new tasks from natural language instructions. *arXiv preprint arXiv:2104.08773*, 2021.

[202] S. Mishra, A. Mitra, N. Varshney, B. Sachdeva, P. Clark, C. Baral, and A. Kalyan. Numglue: A suite of fundamental yet challenging mathematical reasoning tasks, 2022.

[203] E. Mitchell, C. Lin, A. Bosselut, C. Finn, and C. D. Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

[204] E. Mitchell, C. Lin, A. Bosselut, C. D. Manning, and C. Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR, 2022.

[205] J. Mok, J. Do, S. Lee, T. Taghavi, S. Yu, and S. Yoon. Large-scale lifelong learning of in-context instructions and how to tackle it. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12573–12589, Toronto, Canada, July 2023. Association for Computational Linguistics.

[206] A. Moradi Dakhel, V. Majdinasab, A. Nikanjam, F. Khomh, M. C. Desmarais, and Z. M. J. Jiang. Github copilot ai pair programmer: Asset or liability? *Journal of Systems and Software*, 203:111734, 2023.

[207] N. Muennighoff, Q. Liu, A. Zebaze, Q. Zheng, B. Hui, T. Y. Zhuo, S. Singh, X. Tang, L. von Werra, and S. Longpre. Octopack: Instruction tuning code large language models, 2024.

[208] B. Neyshabur, H. Sedghi, and C. Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020.

[209] T. D. Nguyen, Y. Ting, I. Ciuca, C. O'Neill, Z. Sun, M. Jablonska, S. Kruk, E. Perkowski, J. W. Miller, J. Li, J. Peek, K. Iyer, T. Rózanski, P. Khetarpal, S. Zaman, D. Brodrick, S. J. R. Méndez, T. Bui, A. Goodman, A. Accomazzi, J. P. Naiman, J. Cranney, K. Schawinski, and UniverseTBD. Astrollama: Towards specialized foundation models in astronomy. *CoRR*, abs/2309.06126, 2023.

[210] T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W.-C. Liew, H. Yin, and Q. V. H. Nguyen. A survey of machine unlearning, 2022.

[211] J. Ni, J. Li, and J. McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[212] Z. Ni, H. Shi, S. Tang, L. Wei, Q. Tian, and Y. Zhuang. Revisiting catastrophic forgetting in class incremental learning. *arXiv preprint arXiv:2107.12308*, 2021.

[213] Z. Ni, L. Wei, S. Tang, Y. Zhuang, and Q. Tian. Continual vision-language representation learning with off-diagonal information. In *Proceedings of the 40th International Conference on Machine Learning*, pages 26129–26149, 2023.

[214] E. Nijkamp, H. Hayashi, C. Xiong, S. Savarese, and Y. Zhou. Codegen2: Lessons for training llms on programming and natural languages. *ICLR*, 2023.

[215] E. Nijkamp, B. Pang, H. Hayashi, L. Tu, H. Wang, Y. Zhou, S. Savarese, and C. Xiong. Codegen: An open large language model for code with multi-turn program synthesis. *ICLR*, 2023.

[216] H. F. Ólafsdóttir, D. Bush, and C. Barry. The role of hippocampal replay in memory and planning. *Current Biology*, 28(1):R37–R50, 2018.

[217] OpenAI. Introducing chatgpt. [online]. available: `https://openai.com/blog/chatgpt`. 2022.

[218] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, 2022.

[219] C. Pallier, S. Dehaene, J.-B. Poline, D. LeBihan, A.-M. Argenti, E. Dupoux, and J. Mehler. Brain imaging of language plasticity in adopted adults: Can a second language replace the first? *Cerebral cortex*, 13(2):155–161, 2003.

[220] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[221] I. Paul, J. Luo, G. Glavaš, and I. Gurevych. Ircoder: Intermediate representations make language models robust multilingual code generators, 2024.

[222] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, and F. Wei. Kosmos-2: Grounding multimodal large language models to the world, 2023.

[223] A. Pentina. *Theoretical foundations of multi-task lifelong learning*. PhD thesis, 2016.

[224] E. Perkowski, R. Pan, T. D. Nguyen, Y. Ting, S. Kruk, T. Zhang, C. O'Neill, M. Jablonska, Z. Sun, M. J. Smith, H. Liu, K. Schawinski, K. Iyer, I. Ciuca, and UniverseTBD. Astrollama-chat: Scaling astrollama with conversational and diverse datasets. *CoRR*, abs/2401.01916, 2024.

[225] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller. Language models as knowledge bases? In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[226] J. Pourcel, N.-S. Vu, and R. M. French. Online task-free continual learning with dynamic sparse distributed memory. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, editors, *Computer Vision – ECCV 2022*, pages 739–756, Cham, 2022. Springer Nature Switzerland.

[227] A. Prabhu, H. A. Al Kader Hammoud, P. K. Dokania, P. H. Torr, S.-N. Lim, B. Ghanem, and A. Bibi. Computationally budgeted continual learning: What does matter? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3698–3707, 2023.

[228] A. Prabhu, Z. Cai, P. Dokania, P. Torr, V. Koltun, and O. Sener. Online continual learning without the storage constraint, 2023.

[229] G. Puthumanaillam, M. Vora, P. Thangeda, and M. Ornik. A moral imperative: The need for continual superalignment of large language models. *arXiv preprint arXiv:2403.14683*, 2024.

[230] C. Qin and S. Joty. Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. In *International Conference on Learning Representations*, 2021.

[231] Y. Qin, C. Qian, X. Han, Y. Lin, H. Wang, R. Xie, Z. Liu, M. Sun, and J. Zhou. Recyclable tuning for continual pre-training. *arXiv preprint arXiv:2305.08702*, 2023.

[232] Y. Qin, J. Zhang, Y. Lin, Z. Liu, P. Li, M. Sun, and J. Zhou. ELLE: Efficient lifelong pre-training for emerging data. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2789–2810, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[233] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[234] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[235] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[236] P. Rajpurkar, R. Jia, and P. Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

[237] R. Ramesh and P. Chaudhari. Model zoo: A growing" brain" that learns continually. *arXiv preprint arXiv:2106.03027*, 2021.

[238] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, T. Adler, L. Gruber, M. Holzleitner, M. Pavlović, G. K. Sandve, V. Greiff, D. Kreil, M. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter. Hopfield networks is all you need, 2021.

[239] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[240] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

[241] H. Ritter, A. Botev, and D. Barber. Online structured laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems*, 31, 2018.

[242] J. Roberts, T. Lüddecke, S. Das, K. Han, and S. Albanie. Gpt4geo: How a language model sees the world's geography, 2023.

[243] S. Rongali, A. Jagannatha, B. P. S. Rawat, and H. Yu. Continual domain-tuning for pretrained language models, 2021.

[244] G. D. Rosin, I. Guy, and K. Radinsky. Time masking for temporal language models. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 833–841, New York, NY, USA, 2022. Association for Computing Machinery.

[245] B. Rozière, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, R. Sauvestre, T. Remez, J. Rapin, A. Kozhevnikov, I. Evtimov, J. Bitton, M. Bhatt, C. C. Ferrer, A. Grattafiori, W. Xiong, A. Défossez, J. Copet, F. Azhar, H. Touvron, L. Martin, N. Usunier, T. Scialom, and G. Synnaeve. Code llama: Open foundation models for code, 2024.

[246] A. N. Rubungo, C. Arnold, B. P. Rand, and A. B. Dieng. Llm-prop: Predicting physical and electronic properties of crystalline solids from their text descriptions. *CoRR*, abs/2310.14029, 2023.

[247] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[248] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019.

[249] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. Nayak, D. Datta, J. Chang, M. T.-J. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Fevry, J. A. Fries, R. Teehan, T. Bers, S. Biderman, L. Gao, T. Wolf, and A. M. Rush. Multitask prompted training enables zero-shot task generalization, 2022.

[250] F. Sarfraz, E. Arani, and B. Zonooz. Error sensitivity modulation based experience replay: Mitigating abrupt representation drift in continual learning. *arXiv preprint arXiv:2302.11344*, 2023.

[251] J. Savelka, K. D. Ashley, M. A. Gray, H. Westermann, and H. Xu. Explaining legal concepts with augmented large language models (gpt-4), 2023.

[252] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017.

[253] T. Schuster, A. Fisch, and R. Barzilay. Get your vitamin c! robust fact verification with contrastive evidence. *arXiv preprint arXiv:2103.08541*, 2021.

[254] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*, pages 4528–4537. PMLR, 2018.

[255] T. Scialom, T. Chakrabarty, and S. Muresan. Fine-tuned language models are continual learners. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6107–6122, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.

[256] A. Shah and S. Chava. Zero is not hero yet: Benchmarking zero-shot performance of llms for financial tasks, 2023.

[257] A. Shah, S. Paturi, and S. Chava. Trillion dollar words: A new financial dataset, task & market analysis, 2023.

[258] Y. Shao, Y. Guo, D. Zhao, and B. Liu. Class-incremental learning based on label generation. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1263–1276, Toronto, Canada, July 2023. Association for Computational Linguistics.

[259] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[260] H. Shi and H. Wang. A unified approach to domain incremental learning with memory: Theory and algorithm. *Advances in Neural Information Processing Systems*, 36, 2024.

[261] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach. Towards vqa models that can read, 2019.

[262] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

[263] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, L. Hou, K. Clark, S. Pfohl, H. Cole-Lewis, D. Neal, M. Schaekermann, A. Wang, M. Amin, S. Lachgar, P. A. Mansfield, S. Prakash, B. Green, E. Dominowska, B. A. y Arcas, N. Tomasev, Y. Liu, R. Wong, C. Semturs, S. S. Mahdavi, J. K. Barral, D. R. Webster, G. S. Corrado, Y. Matias, S. Azizi, A. Karthikesalingam, and V. Natarajan. Towards expert-level medical question answering with large language models. *CoRR*, abs/2305.09617, 2023.

[264] A. Sinitsin, V. Plokhotnyuk, D. Pyrkin, S. Popov, and A. Babenko. Editable neural networks. *arXiv preprint arXiv:2004.00345*, 2020.

[265] J. S. Smith, J. Tian, S. Halbe, Y.-C. Hsu, and Z. Kira. A closer look at rehearsal-free continual learning, 2023.

[266] D. Soboleva, F. Al-Khateeb, R. Myers, J. R. Steeves, J. Hestness, and N. Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023.

[267] L. Soldaini, R. Kinney, A. Bhagia, D. Schwenk, D. Atkinson, R. Authur, B. Bogin, K. Chandu, J. Dumas, Y. Elazar, V. Hofmann, A. H. Jha, S. Kumar, L. Lucy, X. Lyu, N. Lambert, I. Magnusson, J. Morrison, N. Muennighoff, A. Naik, C. Nam, M. E. Peters, A. Ravichander, K. Richardson, Z. Shen, E. Strubell, N. Subramani, O. Tafjord, P. Walsh, L. Zettlemoyer, N. A. Smith, H. Hajishirzi, I. Beltagy, D. Groeneveld, J. Dodge, and K. Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research, 2024.

[268] C. Song, X. Han, Z. Zeng, K. Li, C. Chen, Z. Liu, M. Sun, and T. Yang. Conpet: Continual parameter-efficient tuning for large language models, 2023.

[269] D. Song, H. Guo, Y. Zhou, S. Xing, Y. Wang, Z. Song, W. Zhang, Q. Guo, H. Yan, X. Qiu, and D. Lin. Code needs comments: Enhancing code llms with comment augmentation, 2024.

[270] P. Sprechmann, S. M. Jayakumar, J. W. Rae, A. Pritzel, A. P. Badia, B. Uria, O. Vinyals, D. Hassabis, R. Pascanu, and C. Blundell. Memory-based parameter adaptation. In *International Conference on Learning Representations*, 2018.

[271] Z. Su, J. Li, Z. Zhang, Z. Zhou, and M. Zhang. Efficient continue training of temporal language model with structural information. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6315–6329, Singapore, Dec. 2023. Association for Computational Linguistics.

[272] Q. Sun, Z. Chen, F. Xu, K. Cheng, C. Ma, Z. Yin, J. Wang, C. Han, R. Zhu, S. Yuan, Q. Guo, X. Qiu, P. Yin, X. Li, F. Yuan, L. Kong, X. Li, and Z. Wu. A survey of neural code intelligence: Paradigms, advances and beyond, 2024.

[273] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang. Ernie 2.0: A continual pre-training framework for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8968–8975, Apr. 2020.

[274] M. Tao, Y. Feng, and D. Zhao. Can bert refrain from forgetting on sequential tasks? a probing study. In *The Eleventh International Conference on Learning Representations*, 2022.

[275] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023.

[276] R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, and R. Stojnic. Galactica: A large language model for science. *CoRR*, abs/2211.09085, 2022.

[277] V. Thengane, S. Khan, M. Hayat, and F. Khan. Clip model is an efficient continual learner, 2022.

[278] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[279] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[280] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[281] M. van de Kar, M. Xia, D. Chen, and M. Artetxe. Don't prompt, search! mining-based zero-shot learning with language models. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7508–7520, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.

[282] G. M. Van de Ven, T. Tuytelaars, and A. S. Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022.

[283] E. Verwimp, R. Aljundi, S. Ben-David, M. Bethge, A. Cossu, A. Gepperth, T. L. Hayes, E. Hüllermeier, C. Kanan, D. Kudithipudi, C. H. Lampert, M. Mundt, R. Pascanu, A. Popescu, A. S. Tolias, J. van de Weijer, B. Liu, V. Lomonaco, T. Tuytelaars, and G. M. van de Ven. Continual learning: Applications and the road forward, 2024.

[284] M. Völske, M. Potthast, S. Syed, and B. Stein. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, 2017.

[285] B. Wang and A. Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[286] C. Wang, D. Engler, X. Li, J. Hou, D. J. Wald, K. Jaiswal, and S. Xu. Near-real-time earthquake-induced fatality estimation using crowdsourced data and large-language models, 2023.

[287] L. Wang, X. Zhang, Q. Li, J. Zhu, and Y. Zhong. 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*, pages 254–271. Springer, 2022.

[288] L. Wang, X. Zhang, H. Su, and J. Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2024.

[289] N. Wang, H. Yang, and C. D. Wang. Fingpt: Instruction tuning benchmark for open-source large language models in financial datasets. *CoRR*, abs/2310.04793, 2023.

[290] R. Wang, D. Tang, N. Duan, Z. Wei, X. Huang, J. Ji, G. Cao, D. Jiang, and M. Zhou. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online, Aug. 2021. Association for Computational Linguistics.

[291] X. Wang, T. Chen, Q. Ge, H. Xia, R. Bao, R. Zheng, Q. Zhang, T. Gui, and X. Huang. Orthogonal subspace learning for language model continual learning. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671, Singapore, Dec. 2023. Association for Computational Linguistics.

[292] X. Wang, Y. Zhang, T. Chen, S. Gao, S. Jin, X. Yang, Z. Xi, R. Zheng, Y. Zou, T. Gui, Q. Zhang, and X. Huang. Trace: A comprehensive benchmark for continual learning in large language models, 2023.

[293] Y. Wang, H. Le, A. D. Gotmare, N. D. Q. Bui, J. Li, and S. C. H. Hoi. Codet5+: Open code large language models for code understanding and generation, 2023.

[294] Y. Wang, Y. Liu, C. Shi, H. Li, C. Chen, H. Lu, and Y. Yang. Inscl: A data-efficient continual learning paradigm for fine-tuning large language models with instructions, 2024.

[295] Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Arunkumar, A. Ashok, A. S. Dhanasekaran, A. Naik, D. Stap, E. Pathak, G. Karamanolakis, H. G. Lai, I. Purohit, I. Mondal, J. Anderson, K. Kuznia, K. Doshi, M. Patel, K. K. Pal, M. Moradshahi, M. Parmar, M. Purohit, N. Varshney, P. R. Kaza, P. Verma, R. S. Puri, R. Karia, S. K. Sampat, S. Doshi, S. Mishra, S. Reddy, S. Patro, T. Dixit, X. Shen, C. Baral, Y. Choi, N. A. Smith, H. Hajishirzi, and D. Khashabi. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks, 2022.

[296] Y. Wang, W. Wang, S. Joty, and S. C. Hoi. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *EMNLP*, 2021.

[297] Z. Wang, L. Liu, Y. Kong, J. Guo, and D. Tao. Online continual learning with contrastive vision transformer. In *European Conference on Computer Vision*, pages 631–650. Springer, 2022.

[298] Z. Wang, Z. Zhan, Y. Gong, G. Yuan, W. Niu, T. Jian, B. Ren, S. Ioannidis, Y. Wang, and J. Dy. Sparcl: Sparse continual learning on the edge. *Advances in Neural Information Processing Systems*, 35:20366–20380, 2022.

[299] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. *European Conference on Computer Vision*, 2022.

[300] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022.

[301] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[302] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners, 2022.

[303] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[304] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[305] Y. Wei, Z. Wang, J. Liu, Y. Ding, and L. Zhang. Magicoder: Source code is all you need, 2023.

[306] M. Weyssow, X. Zhou, K. Kim, D. Lo, and H. Sahraoui. 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*, ESEC/FSE 2023, page 1470–1482, New York, NY, USA, 2023. Association for Computing Machinery.

[307] G. Winata, L. Xie, K. Radhakrishnan, S. Wu, X. Jin, P. Cheng, M. Kulkarni, and D. Preotiuc-Pietro. Overcoming catastrophic forgetting in massively multilingual continual learning. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 768–777, Toronto, Canada, July 2023. Association for Computational Linguistics.

[308] M. Wistuba, P. T. Sivaprasad, L. Balles, and G. Zappella. Continual learning with low rank adaptation. In *NeurIPS 2023 Workshop on Distribution Shifts (DistShifts)*, 2023.

[309] M. Wistuba, P. T. Sivaprasad, L. Balles, and G. Zappella. Continual learning with low rank adaptation, 2023.

[310] C. Wu, Y. Gan, Y. Ge, Z. Lu, J. Wang, Y. Feng, P. Luo, and Y. Shan. Llama pro: Progressive llama with block expansion, 2024.

[311] C. Wu, W. Lin, X. Zhang, Y. Zhang, Y. Wang, and W. Xie. Pmc-llama: Towards building open-source language models for medicine. *arXiv preprint arXiv:2305.10415*, 6, 2023.

[312] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. S. Rosenberg, and G. Mann. Bloomberggpt: A large language model for finance. *CoRR*, abs/2303.17564, 2023.

[313] T. Wu, M. Caccia, Z. Li, Y.-F. Li, G. Qi, and G. Haffari. Pretrained language model in continual learning: A comparative study. In *International conference on learning representations*, 2021.

[314] T. Wu, L. Luo, Y.-F. Li, S. Pan, T.-T. Vu, and G. Haffari. Continual learning for large language models: A survey, 2024.

[315] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381, 2022.

[316] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019.

[317] Y. Wu, G. Wayne, A. Graves, and T. Lillicrap. The kanerva machine: A generative distributed memory. *arXiv preprint arXiv:1804.01756*, 2018.

[318] C. Xiao, X. Hu, Z. Liu, C. Tu, and M. Sun. Lawformer: A pre-trained language model for chinese legal long documents, 2021.

[319] J. Xie, Y. Liang, J. Liu, Y. Xiao, B. Wu, and S. Ni. 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*, KDD '23, page 5282–5291, New York, NY, USA, 2023. Association for Computing Machinery.

[320] Q. Xie, W. Han, X. Zhang, Y. Lai, M. Peng, A. Lopez-Lira, and J. Huang. PIXIU: A large language model, instruction data and evaluation benchmark for finance. *CoRR*, abs/2306.05443, 2023.

[321] S. M. Xie, S. Santurkar, T. Ma, and P. S. Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36, 2024.

[322] T. Xie, Y. Wan, W. Huang, Z. Yin, Y. Liu, S. Wang, Q. Linghu, C. Kit, C. Grazian, W. Zhang, I. Razzak, and B. Hoex. DARWIN series: Domain specific large language models for natural science. *CoRR*, abs/2308.13565, 2023.

[323] Y. Xie, K. Aggarwal, and A. Ahmad. Efficient continual pre-training for building domain specific large language models, 2023.

[324] H. Xiong, S. Wang, Y. Zhu, Z. Zhao, Y. Liu, Q. Wang, and D. Shen. Doctorglm: Fine-tuning your chinese doctor is not a herculean task. *arXiv preprint arXiv:2304.01097*, 2023.

[325] H. Xu, B. Liu, L. Shu, and P. Yu. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[326] S. Xue, F. Zhou, Y. Xu, H. Zhao, S. Xie, Q. Dai, C. Jiang, J. Zhang, J. Zhou, D. Xiu, and H. Mei. Weaverbird: Empowering financial decision-making with large language model, knowledge base, and search engine. *CoRR*, abs/2308.05361, 2023.

[327] Y. Yan, K. Xue, X. Shi, Q. Ye, J. Liu, and T. Ruan. Af adapter: Continual pretraining for building chinese biomedical language model. In *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 953–957, Los Alamitos, CA, USA, dec 2023. IEEE Computer Society.

[328] G. Yang, F. Pan, and W.-B. Gan. Stably maintained dendritic spines are associated with lifelong memories. *Nature*, 462(7275):920–924, 2009.

[329] P. Yang, D. Li, and P. Li. Continual learning for natural language generations with transformer calibration. In A. Fokkens and V. Srikumar, editors, *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 40–49, Abu Dhabi, United Arab Emirates (Hybrid), Dec. 2022. Association for Computational Linguistics.

[330] S. Yang, M. A. Ali, C.-L. Wang, L. Hu, and D. Wang. Moral: Moe augmented lora for llms' lifelong learning, 2024.

[331] X. Yang, J. Gao, W. Xue, and E. Alexandersson. Pllama: An open-source large language model for plant science. *CoRR*, abs/2401.01600, 2024.

[332] Y. Yang, M. Jones, M. C. Mozer, and M. Ren. Reawakening knowledge: Anticipatory recovery from catastrophic interference via structured training, 2024.

[333] Y. Yang, Y. Tang, and K. Y. Tam. Investlm: A large language model for investment using financial domain instruction tuning. *CoRR*, abs/2309.13064, 2023.

[334] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[335] H. Yin, p. yang, and P. Li. Mitigating forgetting in online continual learning with neuron calibration. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 10260–10272. Curran Associates, Inc., 2021.

[336] J. Yin, S. Dash, F. Wang, and M. Shankar. FORGE: pre-training open foundation models for science. In D. Arnold, R. M. Badia, and K. M. Mohror, editors, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2023, Denver, CO, USA, November 12-17, 2023*, pages 81:1–81:13. ACM, 2023.

[337] W. Yin, J. Li, and C. Xiong. ConTinTin: Continual learning from task instructions. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3062–3072, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[338] F. Yu, A. Gao, and B. Wang. Outcome-supervised verifiers for planning in mathematical reasoning. *CoRR*, abs/2311.09724, 2023.

[339] L. Yu, Q. Chen, J. Zhou, and L. He. Melo: Enhancing model editing with neuron-indexed dynamic lora. *arXiv preprint arXiv:2312.11795*, 2023.

[340] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911, 2020.

[341] S. Yuan, H. Zhao, Z. Du, M. Ding, X. Liu, Y. Cen, X. Zou, Z. Yang, and J. Tang. Wudaocorpora: A super large-scale chinese corpora for pre-training language models. *AI Open*, 2, 06 2021.

[342] W. Yuan, Q. Zhang, T. He, C. Fang, N. Q. V. Hung, X. Hao, and H. Yin. Circle: continual repair across programming languages. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2022, page 678–690, New York, NY, USA, 2022. Association for Computing Machinery.

[343] S. Yue, W. Chen, S. Wang, B. Li, C. Shen, S. Liu, Y. Zhou, Y. Xiao, S. Yun, W. Lin, et al. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325*, 2023.

[344] X. Yue, X. Qu, G. Zhang, Y. Fu, W. Huang, H. Sun, Y. Su, and W. Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

[345] L. W. Zefeng Du, Minghao Wu. Chinese-llama-2. https://github.com/longyuewangdcu/Chinese-Llama-2, 2023.

[346] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi. Defending against neural fake news. *Advances in neural information processing systems*, 32, 2019.

[347] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence, 2017.

[348] A. Zewdu and B. Yitagesu. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9, 01 2022.

[349] Y. Zhai, S. Tong, X. Li, M. Cai, Q. Qu, Y. J. Lee, and Y. Ma. Investigating the catastrophic forgetting in multimodal large language models, 2023.

[350] D. Zhang, Z. Hu, S. Zhoubian, Z. Du, K. Yang, Z. Wang, Y. Yue, Y. Dong, and J. Tang. Sciglm: Training scientific language models with self-reflective instruction annotation and tuning. *CoRR*, abs/2401.07950, 2024.

[351] H. Zhang, L. Gui, Y. Zhai, H. Wang, Y. Lei, and R. Xu. Copf: Continual learning human preference through optimal policy fitting. *arXiv preprint arXiv:2310.15694*, 2023.

[352] H. Zhang, Y. Lei, L. Gui, M. Yang, Y. He, H. Wang, and R. Xu. Cppo: Continual learning for reinforcement learning with human feedback.

[353] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, and G. Wang. Instruction tuning for large language models: A survey, 2024.

[354] X. Zhang, C. Tian, X. Yang, L. Chen, Z. Li, and L. R. Petzold. Alpacare: Instruction-tuned large language models for medical application. *arXiv preprint arXiv:2310.14558*, 2023.

[355] X. Zhang and Q. Yang. 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*, CIKM '23, page 4435–4439, New York, NY, USA, 2023. Association for Computing Machinery.

[356] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[357] Y. Zhang, X. Wang, and D. Yang. Continual sequence generation with adaptive compositional modules. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3653–3667, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[358] Z. Zhang, M. Fang, L. Chen, and M.-R. Namazi-Rad. CITB: A benchmark for continual instruction tuning. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9443–9455, Singapore, Dec. 2023. Association for Computational Linguistics.

[359] C. Zhao, Y. Li, and C. Caragea. C-STANCE: A large dataset for Chinese zero-shot stance detection. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13369–13385, Toronto, Canada, July 2023. Association for Computational Linguistics.

[360] H. Zhao, S. Liu, C. Ma, H. Xu, J. Fu, Z.-H. Deng, L. Kong, and Q. Liu. GIMLET: A unified graph-text model for instruction-based molecule zero-shot learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[361] H. Zhao, H. Wang, Y. Fu, F. Wu, and X. Li. Memory-efficient class-incremental learning for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5966–5977, 2022.

[362] S. Zhao, X. Zou, T. Yu, and H. Xu. Reconstruct before query: Continual missing modality learning with decomposed prompt collaboration, 2024.

[363] W. Zhao, S. Wang, Y. Hu, Y. Zhao, B. Qin, X. Zhang, Q. Yang, D. Xu, and W. Che. Sapt: A shared attention framework for parameter-efficient continual learning of large language models, 2024.

[364] J. Zheng, Q. Ma, Z. Liu, B. Wu, and H. Feng. Beyond anti-forgetting: Multimodal continual instruction tuning with positive forward transfer, 2024.

[365] J. Zheng, S. Qiu, and Q. Ma. Learn or recall? revisiting incremental learning with pre-trained language models, 2023.

[366] Q. Zheng, X. Xia, X. Zou, Y. Dong, S. Wang, Y. Xue, Z. Wang, L. Shen, A. Wang, Y. Li, T. Su, Z. Yang, and J. Tang. Codegeex: A pre-trained model for code generation with multilingual evaluations on humaneval-x, 2023.

[367] Z. Zheng, J. Zhang, T. Vu, S. Diao, Y. H. W. Tim, and S. Yeung. Marinegpt: Unlocking secrets of ocean to the public. *CoRR*, abs/2310.13596, 2023.

[368] B. Zhou, D. Khashabi, Q. Ning, and D. Roth. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[369] W. Zhou, D.-H. Lee, R. K. Selvam, S. Lee, B. Y. Lin, and X. Ren. Pre-training text-to-text transformers for concept-centric common sense. 2021.

[370] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023.

[371] D. Zhu, Z. Sun, Z. Li, T. Shen, K. Yan, S. Ding, K. Kuang, and C. Wu. Model tailor: Mitigating catastrophic forgetting in multi-modal large language models, 2024.

[372] T. Y. Zhuo, A. Zebaze, N. Suppattarachai, L. von Werra, H. de Vries, Q. Liu, and N. Muennighoff. Astraios: Parameter-efficient instruction tuning code large language models, 2024.