Towards Lifelong Learning of Large Language Models: A Survey

JUNHAO ZHENG*, School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
SHENGJIE QIU*, School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
CHENGMING SHI*, School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
QIANLI MA†, School of Computer Science and Engineering, South China University of Technology, Guangzhou, China

As the applications of large language models (LLMs) expand across diverse fields, their ability to adapt to ongoing changes in data, tasks, and user preferences becomes crucial. Traditional training methods with static datasets are inadequate for coping with the dynamic nature of real-world information. Lifelong learning, or continual learning, addresses this by enabling LLMs to learn continuously and adapt over their operational lifetime, integrating new knowledge while retaining previously learned information and preventing catastrophic forgetting. Our survey explores the landscape of lifelong learning, categorizing strategies into two groups based on how new knowledge is integrated: Internal Knowledge, where LLMs absorb new knowledge into their parameters through full or partial training, and External Knowledge, which incorporates new knowledge as external resources like Wikipedia or APIs without updating model parameters. The key contributions of our survey include: (1) Introducing a novel taxonomy to categorize the extensive literature of lifelong learning into 12 scenarios; (2) Identifying common techniques across all lifelong learning scenarios and classifying existing literature into various technique groups; (3) Highlighting emerging techniques such as model expansion and data selection, which were less explored in the pre-LLM era. Resources are available at https://github.com/qianlima-lab/awesome-lifelong-learning-methods-for-llm.

CCS Concepts: • **Computing methodologies** → **Natural language processing**; *Natural language generation*; *Online learning settings*.

Additional Key Words and Phrases: Lifelong Learning, Continual Learning, Incremental Learning, Large Language Models, Catastrophic Forgetting

1 Introduction

As the applications of large language models (LLMs) [1, 29, 161, 188, 238] expand across diverse fields, the ability of these models to adapt to ongoing changes in data, tasks, and user preferences becomes crucial. Traditional

*The first three authors contributed equally to this research.
†Corresponding author

Authors' Contact Information: Junhao Zheng, School of Computer Science and Engineering, South China University of Technology, Guangzhou, China; e-mail: junhaozheng47@outlook.com; Shengjie Qiu, School of Computer Science and Engineering, South China University of Technology, Guangzhou, China; e-mail: shengjieqiu6@gmail.com; Chengming Shi, School of Computer Science and Engineering, South China University of Technology, Guangzhou, China; e-mail: secmshi@mail.scut.edu.cn; Qianli Ma, School of Computer Science and Engineering, South China University of Technology, Guangzhou, Guangdong, China; e-mail: qianlima@scut.edu.cn.

ACM 1557-7341/2025/2-ART
https://doi.org/10.1145/3716629

training methods, which rely on static datasets to train LLMs, are increasingly inadequate for coping with the dynamic nature of real-world information [255]. Lifelong learning [198] (a.k.a., continual learning, incremental learning), or the capability of LLMs to learn continuously and adaptively over their operational lifetime [174], addresses this challenge by integrating new knowledge while retaining previously learned information, thereby preventing the problem of catastrophic forgetting [128]. An illustration of lifelong learning is provided in Figure 1.

This survey delves into the sophisticated landscape of lifelong learning, categorizing strategies into two primary groups based on how new knowledge is integrated: Internal Knowledge and External Knowledge. Each category encompasses distinct approaches that collectively aim to enhance the adaptability and effectiveness of LLMs in various scenarios. We provide the taxonomy of lifelong learning methods for LLMs in Figure 2.

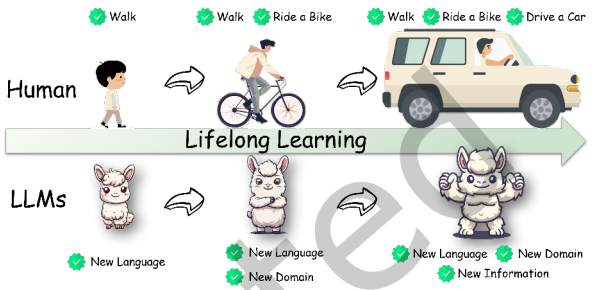


Fig. 1. An illustration of lifelong learning: humans can incrementally learn new skills such as walking, riding a bike, and driving a car. Similarly, lifelong learning aims to equip LLMs with new languages, domain knowledge, and information.

The Internal Knowledge group, where LLMs absorb new knowledge into their parameters through full or partial training, includes strategies such as continual pretraining [20, 45, 78, 119, 156] and continual finetuning [69, 110, 140, 182, 196, 204, 266]. For example, in industry applications, continual vertical domain pretraining [47, 176] is commonly adopted, where companies frequently retrain their LLMs using domain-specific data from sectors like finance [227]. Although this enhances performance in specialized areas, it risks diminishing the model's broader knowledge base, illustrating the challenges of maintaining a balance between specialized adaptation and general knowledge retention. Continual finetuning covers methods tailored to specific scenarios—such as text classification [69], named entity recognition [140], relation extraction [196], and machine translation [14]—as well as task-agnostic methods like instruction tuning [182], alignment [110], and knowledge editing [204]. Additionally, reinforcement learning with human feedback [181] is employed in continual alignment to ensure that LLMs adhere to human values like safety and politeness [98, 148], highlighting the trade-off known as the "alignment tax" [110], where focusing too narrowly on specific values can lead to a degradation of the model's general capabilities.

External Knowledge, which incorporates new knowledge as external resources like Wikipedia or APIs without updating model parameters, includes retrieval-based [81] and tool-based lifelong learning [153], which leverage external data sources and computational tools to extend the model's capabilities. Retrieval-based strategies, such as retrieval-augmented generation [5, 76, 81, 90, 189], enhance text generation by providing contextually relevant, accurate, and latest information from external databases such as Wikipedia, ensuring the model's outputs remain relevant over time. Meanwhile, tool-based learning draws parallels to human tool use [4], where models learn to utilize external computational tools, thus broadening their problem-solving capabilities without direct modifications to their core knowledge base.

Through a detailed examination of these groups and their respective categories, this paper aims to highlight the integration of lifelong learning capabilities into LLMs, thereby enhancing their adaptability, reliability, and overall performance in real-world applications. By addressing the challenges associated with lifelong learning and exploring the innovations in this field, this survey seeks to contribute to the ongoing development of more robust and versatile LLMs capable of thriving in an ever-evolving digital landscape.

Differences between this survey and existing ones. Lifelong learning has become an increasingly popular research topic in recent years. Massive surveys have explored the lifelong learning of neural networks

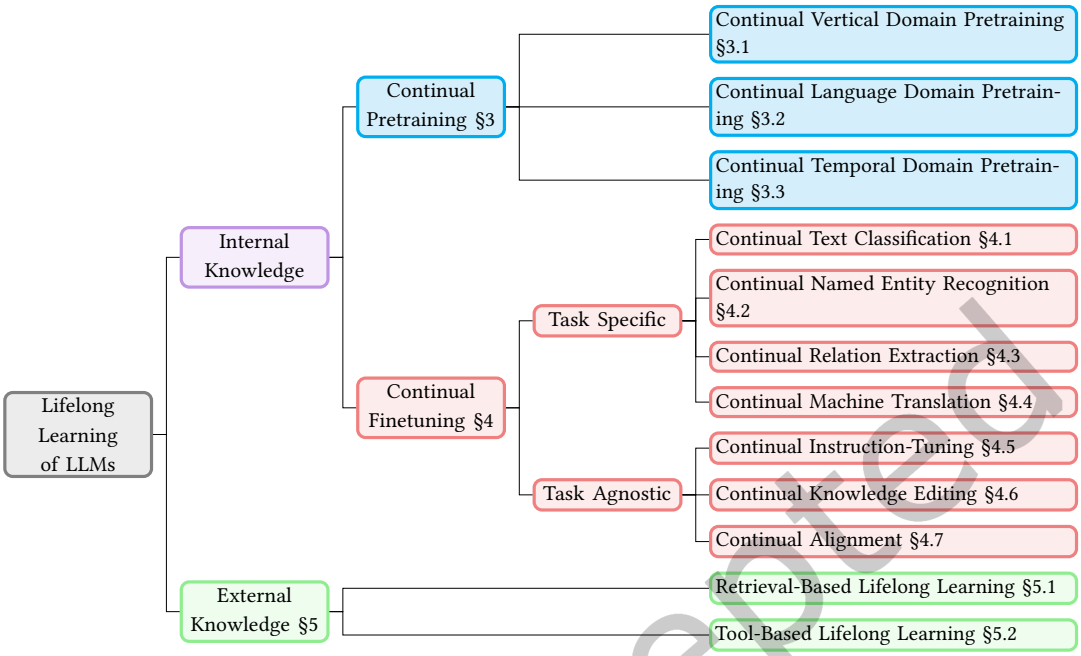


Fig. 2. Taxonomy of lifelong learning methods for LLMs.

[10, 35, 40, 82, 130, 144, 167, 174, 187, 198, 218, 228, 235, 236, 255, 268]. Most of the existing surveys primarily focus on the lifelong learning of Convolutional Neural Networks (CNNs) [10, 35, 130, 144, 198, 235, 268]. They examined various scenarios of lifelong learning of CNNs, including image classification [10, 35, 144, 198, 268], segmentation [235], objection detection [130], autonomous systems [167], robotics [99], and the smart city [228]. Besides, several surveys explored the lifelong learning of Graph Neural Network [40, 187, 236, 240]. However, only a small amount of literature focuses on lifelong learning of language models [10, 79, 82, 174, 218, 255]. Biesialska et al. [10] is an early survey about lifelong learning in Natural Language Processing (NLP). However, they only focus on lifelong learning of word and sentence representations, language modeling, question and answering, text classification, and machine translation. Ke et al. [82] focus on lifelong learning scenarios, including sentiment classification, named entity recognition, and summarization. They also discuss the techniques for knowledge transfer and inter-task class separation for lifelong learning. [79, 174, 218, 255] are four recent surveys closely related to this research. Zhang et al. [255] provide a comprehensive review of techniques in aligning LLMs with the ever-changing world knowledge, including continual pretraining, knowledge editing, and retrieval augmented generation. Wu et al. [218] revisit lifelong learning from three aspects, including continual pretraining, continual instruction tuning, and continual alignment. Shi et al. [174] examine the lifelong learning of LLMs from two directions including vertical direction (or vertical continual learning), i.e., a continual adaptation from general to specific capabilities, and horizontal direction (or horizontal continual learning), i.e., continual adaptation across time and domains. Jovanovic et al. [79] review several real-time learning paradigms, including continual learning, meta-learning, parameter-efficient learning, and mixture-of-experts learning. Although recent surveys [79, 174, 218, 255] collects the latest literature for lifelong learning, none of them covers the scenarios including continual text classification, continual named entity recognition, continual relation extraction, and continual machine translation, and have little discussion about continual alignment, continual knowledge editing,

tool-based lifelong learning and retrieval-based lifelong learning. **To our best knowledge, we are the first survey to provide a thorough and systematic examination of lifelong learning methods for LLMs from 12 scenarios.**

Contributions of this survey. The key contributions of our survey are:

- **Novel Taxonomy**: We introduce a detailed and structured framework for categorizing the extensive literature of lifelong learning into 12 scenarios (shown in Figure 2).
- **Common Techniques**: We identify common techniques across all lifelong learning scenarios in Section 2.3 and classify existing literature into various technique groups within each scenario (e.g., Table 1, 2, 3).
- **Future Directions**: We highlight several emerging techniques, such as model expansion (section 3.1.2) and data selection (section 3.1.4), that were less explored in the pre-LLM era.

Organization of this survey. The remainder of this paper is organized as follows. Section 2 introduces the problem formulation, evaluation metrics, common techniques, benchmarks, and datasets for lifelong learning. Section 3, Section 4, and Section 5 examine the existing techniques for continual pretraining, continual finetuning, and external-knowledge-based lifelong learning. Section 6 discusses the existing challenges, current trends, and future directions for lifelong learning with LLMs and concludes this survey.

2 Overview of Lifelong Learning

2.1 Problem Formulation

Formally, lifelong learning aims to learn a language model $f_\theta : \mathbf{x} \rightarrow \mathbf{y}$ from the sequence of tasks $\{\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \cdots, \mathcal{D}^{(T)}\}$, where the t-th task $\mathcal{D}^{(t)} = \{(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})\}$ contains input $\mathbf{x}^{(t)}$ and target output $\mathbf{y}^{(t)}$. The input $\mathbf{x}$ and $\mathbf{y}$ are both natural languages. For generation tasks such as question and answering, $\mathbf{x}$ and $\mathbf{y}$ represent questions and answers. In machine translation, $\mathbf{x}$ and $\mathbf{y}$ represent the source and target language. In text classification, $\mathbf{x}$ and $\mathbf{y}$ represent the input text and the class label name. In pretraining tasks for autoregressive language models, $\mathbf{x}$ represents a sequence of tokens $[x_1, x_2, \cdots, x_{n-1}]$, and $\mathbf{y}$ represents the corresponding sequence where each token is the next token in the original input, $[x_2, x_3, \cdots, x_n]$.

2.2 Evaluation Metrics

The assessment of continual learning's effectiveness can be approached from three angles: the overall performance of all tasks learned so far, the stability of previously learned tasks, and the plasticity to new tasks.

- **Overall Measurement**: (1) *average accuracy* (AA, higher is better) is computed as the average performance on all tasks learned so far. Formally, the average accuracy when the model has learned t tasks is defined as follows: $\text{AA}_t = \frac{1}{t}\Sigma_{i=1}^{t} a_{t,i}$, where $a_{t,i}$ is the performance score on task i when the model has learned t tasks. We suppose that the performance score is higher when the performance is better. (2) *average incremental accuracy* (AIA, higher is better) is computed as the average of the average accuracy after learning each task. Suppose there are a total of T tasks, we have $\text{AIA} = \frac{1}{T}\Sigma_{t=1}^{T} AA_t$. Compared to AA, AIA captures the historical variation when learning each task.
- **Stability Measurement**: (1) *forgetting measure* (FGT, lower is better) evaluates the average performance drop of each old task. The performance drop is defined as the difference between its maximum performance obtained previously and its current performance. Formally, the forgetting measure after learning t tasks is defined as follows: $\text{FGT}_t = \frac{1}{t-1}\Sigma_{i=1}^{t-1}[max_{j\in\{i,i+1,\cdots,t\}}(\{a_{j,i}\}_j) - a_{t,i}]$, where $max_{j\in\{i,i+1,\cdots,t\}}(\{a_{j,i}\}_j)$ represents the maximum performance of task i after task i has been learned, and $a_{t,i}$ represents the performance of task i after learning t tasks. (2) *backward transfer* (BWT, higher is better) evaluates the average performance change of each old task. The performance change is defined as the difference between its current performance and its

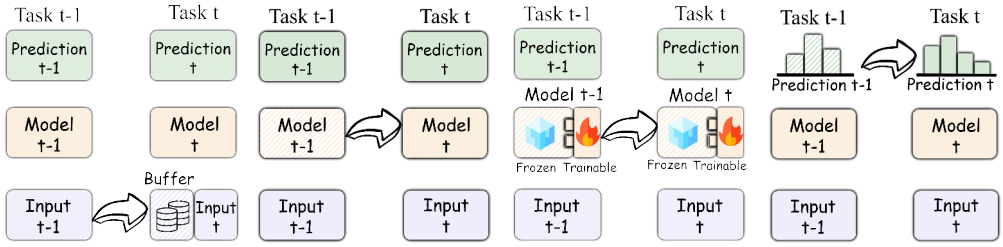


Fig. 3. Four categories of common techniques for lifelong learning with LLMs.

performance at the time the task was initially learned. Formally, the backward transfer after learning t tasks is defined as follows: $\text{BWT}_t = \frac{1}{t-1}\Sigma_{i=1}^{t-1}(a_{t,i} - a_{i,i})$.

- **Plasticity Measurement**: *forward transfer* (FWD, higher is better) evaluates the average enhancement in performance on each newly acquired task. This metric calculates the improvement as the difference between the task's initial performance when first learned and the performance of a model that starts with no prior knowledge and is trained only in this task. Formally, the forward transfer after learning t tasks is defined as follows: $\text{FWD}_t = \frac{1}{t-1}\Sigma_{i=2}^{t}(a_{i,i} - \tilde{a}_i)$, where $\tilde{a}_i$ is the performance of a randomly-initialized model trained on $\mathcal{D}^{(i)}$ only.

2.3 Common Techniques

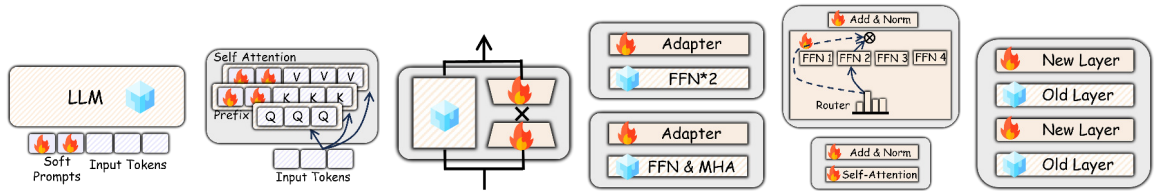
The existing techniques for lifelong learning can be roughly divided into four categories: *replay-based methods*, *regularization-based methods*, *architecture-based methods*, and *distillation-based methods*. An illustration of four categories of lifelong learning methods is provided in Figure 3.

2.3.1 Replay-based methods. Replay-based methods are primarily categorized into Experience Replay and Generative Replay, based on how the replay data is obtained.

- **Experience Replay**: This approach involves retaining a subset of previously encountered data or simpler representations of that data, which are periodically reintegrated during the training of new tasks. This technique helps sustain the model's performance on prior tasks by re-exposing it to old data, reinforcing the existing knowledge. For example, in the context of continual pretraining, [47, 78, 115, 156] systematically reintroduces domain-specific datasets during training phases to refresh the model's memory and stabilize its learning across various domains.
- **Generative Replay**: Instead of storing actual data, this method generates new data samples that emulate old data, using either the model itself or a separate generative model. This approach facilitates continuous learning without the need to retain large volumes of actual data, optimizing memory use and potentially protecting privacy. Within the scope of continual instruction tuning, several innovative methods exemplify generative replay: LAMOL [182], LFPT5 [151], PCLL [261] and SSR [65] generates pseudo instances that are conditioned on natural language cues.

2.3.2 Regularization-Based Methods. Based on the component they regularize, methods employing regularization can be broadly categorized into weight regularization and feature regularization:

- **Weight Regularization**: This technique penalizes changes to the weights that were important for previous tasks, thus preserving the performance on those tasks. Common strategies include L2 Regularization, which



(a) Prompt Tuning (b) Prefix Tuning (c) LoRA (d) Adapters (e) MoE (f) Model Expansion

Fig. 4. Six categories of architecture-based lifelong methods for LLMs.

imposes a penalty on the square of the weights to deter large changes; Elastic Weight Consolidation (EWC) [91], selectively penalizing changes to weights that are critical for past tasks based on their calculated importance; Memory Aware Synapses (MAS) [3], which dynamically adjusts the penalty according to the parameter's sensitivity to changes in task performance. Additionally, RecAdam [21] incorporates ideas from EWC, introducing a regularization to the pretrained weights with an annealing coefficient to gradually integrate the importance of past knowledge.

- **Feature Regularization**: This method involves constraining the features extracted by the model so that new learning does not significantly interfere with the features learned from previous tasks. Techniques such as IDBR [69] and CPFD [241] apply constraints directly on the features to ensure that the activation patterns remain stable across tasks, maintaining a consistent representation space.

2.3.3 Architecture-Based Methods. Architecture-based methods in lifelong learning focus on adapting the structure of models to seamlessly integrate new tasks while minimizing disruption to previously acquired knowledge. These techniques are particularly vital for existing large language models, such as LLaMA-65B [188], GLM-130B [238], PaLM-540B [29], and GPT-4 [1], as fully fine-tuning such large-scale models demands extensive computational resources. Given these constraints, it is both practical and necessary to pursue efficient and cost-effective lifelong learning strategies. Below is a concise overview of six prominent architecture-based methods for lifelong learning and an illustration is provided in Figure 4:

- **Prompt Tuning** [100]: In Prompt Tuning, trainable task-specific prompts are inserted at the model's input layer to steer its responses towards desired outcomes. This method operates by embedding these prompts directly into the input sequence, affecting only the initial processing of input data. Examples include L2P [211], CODA-Prompt [178], SAPT [138], ConvPrompt [165], Q-Tuning [48], and Fwd-Prompt [263].
- **Prefix Tuning** [105]: This method involves prepending a set of trainable parameters, known as prefixes, to each layer of the transformer model. These prefixes act as contextual modifications that adjust the model's behavior for specific tasks. Prefix Tuning influences multiple layers of the model, in contrast to Prompt Tuning. Notable implementations include EPI [210] and MoCL [199].
- **LoRA** (Low-Rank Adaptation) [63]: LoRA integrates low-rank matrices within certain layers of a pre-trained model to adapt its functionality without comprehensive retraining. It allows for targeted adjustments to specific model components. Methods utilizing LoRA include Lee et al. [96], C-LoRA [177], ConPET [179], GLRL [258], O-LoRA [206], CoLoR [214], InfLoRA [108], SAPT [138], MoRAL [230], EKFAC [19], and I-LoRA [162].
- **Adapters** [59]: These are small, two-layer feed-forward neural networks with a bottleneck structure, inserted between the layers of the existing model architecture. They allow the model to acquire new capabilities while preserving the original pre-trained parameters intact. Examples include CPT [82], LAFT-URIEL [6], DMEA [150], TSS [84], HOP [135], and SEMA [195].

- **Mixture of Experts** (MoE) [170]: MoE approaches utilize a gating mechanism to dynamically select from a set of expert feed-forward neural networks during inference, based on the task at hand. This allows the model to specialize certain parts of its architecture to specific types of tasks, enhancing performance and scalability. Examples include DEMix [50] and ModuleFormer [173].
- **Model Expansion** [22]: This category includes techniques that either reuse existing model components or expand the model architecture to accommodate new information and tasks. This can involve adding new layers or modules, or scaling existing ones to increase the model's capacity and flexibility. Notable methods include bert2BERT [17], Wang et al. [200], LLaMA Pro [215], and SOLAR [89].

2.3.4 Distillation-Based Methods. Based on the source of the distilled targets, distillation-based methods can be categorized into three groups: new data, old data, and pseudo-old data:

- **Distillation from New Data**: These techniques involve the student model learning directly from new tasks under the guidance of a teacher model with new data. Representative methods include Learning without Forgetting (LwF) [106], where the model adapts to new classes without forgetting older ones. In continual named entity recognition, the overlap between new and old entities is addressed by methods like ExtendNER [140] and CFNER [262], which use the old model to generate pseudo soft labels for "Other" tokens, aiding the learning of new entities while maintaining old knowledge. Additionally, in continual machine translation, methods such as Cao et al. [14], COKD [168], LFR [46], and CKD [241] employ distillation strategies focusing on new data.
- **Distillation from Old Data**: This category uses old data, which is typically stored in memory, to guide the student model through the outputs of a teacher model. Examples include CRN [7], CRL [259], SCKD [207], and CEAR [260].
- **Distillation from Pseudo Old Data**: When retaining old training data is impractical, methods like L&R [222], Wang et al. [203], DnR [183], PCLL [261], and LFPT5 [151] generate synthetic old data. These methods create pseudo-samples that simulate old data distribution. This category is often utilized in generation tasks and named entity recognition.

2.4 Benchmarks and Datasets

We summarize the commonly used benchmarks and datasets as follows: (1) **Continual Text Classification**: CLINC150 [94], BANKING77 [15], AGNews, Yelp, Amazon, DBPedia, Yahoo [250], HWU64 [118], (HL5Domains, Liu3Domains, Ding9Domains, SemEval14) [87], GLUE [194]; (2) **Continual Named Entity Recognition**: OntoNotes5 [60], I2B2 [141], Few-NERD [36]; (3) **Continual Relation Extraction**: FewRel [54], TRACRED [254]; (4) **Continual Machine Translation**: WMT, TED Talks; (5) **Continual Knowledge Editing**: zsRE [34], FEVER [186], CounterFact [131]; (6) **Continual Instruction Tuning**: (MNLI, QQP, RTE, SST2) GLUE [194], (WiC, CB, COPA, MultiRC, BoolQ) SuperGLUE [193], NaturalInstruction[136], SuperNI[209]; (7) **Continual Alignment**: HH-RLHF [181], Reddit TL;DR [192];

3 Methodology: Continual Pretraining

Continual pretraining [31, 49, 51, 52, 78, 86, 102, 125, 156, 223, 224, 231, 237] enhances the internal knowledge of LLMs and is particularly valuable given the high costs associated with full pretraining. Although research on continual pretraining is less developed compared to continual finetuning, it is crucial for enhancing the general capabilities of existing LLMs. There are three types of continual pretraining: *Continual Vertical Domain Pretraining* [31, 47, 49, 78, 102, 125, 155, 156, 223, 224, 231, 237], targeting domain-specific continuous learning without forgetting previously acquired expertise; *Continual Language Domain Pretraining* [6, 23, 45, 71, 213, 225, 226], focusing on adapting to evolving language usage; and *Continual Temporal Domain Pretraining* [52, 74, 95, 119, 122, 248, 256], which updates models with time-sensitive data and enables model to grasp the latest knowledge.

Building on the definition of lifelong learning in Section 2.1, we define continual pretraining as the incremental learning process applied to a sequence of tasks $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(T)}$, where each dataset $(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})$ represents corpora from distinct vertical, temporal, or language domains.

3.1 Continual Vertical Domain Pretraining

Continual Vertical Domain Pretraining [31, 47, 49, 78, 102, 125, 155, 156, 223, 224, 231, 237] involves continuously training a language model on a series of domain-specific datasets. This method ensures the model performs efficiently across multiple vertical domains or tasks while retaining previously acquired knowledge. For instance, continual pretraining on financial domain data enables LLMs to provide a better analysis of financial texts and data [227].

Experimental investigations in continual vertical domain pretraining primarily focus on addressing catastrophic forgetting [24, 58, 231]. As a pioneering work, Jin et al. [78] revealed that distillation-based approaches are most effective in retaining downstream performance in earlier domains. Building on this, Mehta et al. [129] found that models pre-trained on a diverse set of tasks tend to experience less forgetting compared to those trained from scratch, highlighting the benefits of task diversity. Similarly, Cossu et al. [31] demonstrated that continual pretraining can help mitigate forgetting, supporting the notion that sustained exposure to various tasks can enhance model robustness. However, Li et al. [102] emphasized that catastrophic forgetting remains a significant challenge and cannot be fully resolved through straightforward methods such as freezing layers, modules, LoRA, and $(IA)^3$ [112]. These findings collectively underscore the complexity of addressing catastrophic forgetting and the need for innovative approaches in continual vertical domain pretraining. Research on continual vertical domain pretraining has been evolving with various techniques, including but not limited to *experience replay* [47, 78, 115, 156], *parameter-efficient finetuning* [47, 78, 111, 176], *mixture of experts* [50, 73], *knowledge distillation* [78, 155], *model expansion* [17, 89, 156, 215], *re-warming* [49], and *data selection* [2, 111, 125].

3.1.1 ***Parameter-Efficient Fine-Tuning.*** Parameter Efficient Fine-Tuning is a technique designed to optimize models for specific tasks without requiring extensive computational resources. CorpusBrain++ [47] addresses the dynamic nature of real-world knowledge-intensive language tasks by employing a backbone-adapter architecture and an experience replay strategy. In a similar vein, Med-PaLM [176] introduces instruction prompt tuning to the medical domain using a few exemplars. These methods underscore the importance of efficient fine-tuning strategies in adapting LLMs to specialized domains while addressing the challenges of maintaining performance across diverse tasks.

3.1.2 ***Model Expansion.*** Model expansion involves enhancing the architecture of pre-trained language models by increasing their width and depth to improve efficiency in knowledge acquisition and integration from continuous data streams across multiple domains. ELLE [156] employs a function-preserved model expansion strategy to achieve this, flexibly expanding the width and depth of existing pre-trained language models. Similarly, bert2BERT [17] enhances a base BERT model by expanding its architecture, enabling it to better handle new and more complex data while retaining knowledge from earlier training phases. In line with these approaches, LLaMA Pro [215] expands Transformer blocks and fine-tunes them using a new corpus, achieving superior performance in tasks related to general use, programming, and mathematics. Additionally, SOLAR [89] utilizes depth up-scaling, which involves depthwise scaling and continued pretraining, to efficiently boost LLM performance across various NLP tasks without necessitating complex changes for training and inference.

3.1.3 ***Re-warming.*** Re-warming involves adjusting the *learning rate* upwards when introducing new datasets for continual training. Gupta et al. [49] propose this strategy to prevent the learning rate from diminishing too much over extended training periods, which can otherwise stall the learning process when new data is introduced.

Experimental results show that re-warming the model not only helps in adapting to new datasets more effectively but also enhances overall downstream task performance.

3.1.4 ***Data Selection***. Data Selection plays a crucial role in pretraining, where various lightweight filters are employed to ensure data quality [2, 111]. These filters include heuristic-based methods (e.g., language and item count filtering), classifier-based methods [12], and perplexity-based techniques [212]. For instance, the RedPajama-Data-v2 dataset [30] employs over 40 quality indicators for data filtering and reweighting to enhance data selection.

Recently, Lin et al. [111] introduced RHO-1, which is trained with Selective Language Modeling (SLM). SLM identifies and prioritizes the most impactful tokens during the training process by assessing the gradient impact of each token, thus giving priority to those that cause higher changes in the loss function. In another approach, LESS [221] proposes a low-rank gradient similarity search algorithm to efficiently select the most relevant data for targeted instruction tuning, significantly boosting model performance by training on a carefully chosen subset of the data. Additionally, Ma et al. [125] propose EcomGPT-CT, which leverages semi-structured e-commerce data to enhance the model's performance on domain-specific tasks. EcomGPT-CT utilizes a data mixing strategy, integrating general pretraining data with domain-specific semi-structured data, thereby improving its effectiveness in specific domains.

3.2 Continual Language Domain Pretraining

Continual Language Domain Pretraining [6, 23, 45, 71, 213, 225, 226] extends the concept of pretraining language models to continuously integrate new data and adapt to evolving language domains without forgetting previous knowledge. The studies on continual language domain pretraining focus on natural language [6, 213, 226] and code language [20, 225]. Studies on continual language domain pretraining mainly focus on techniques such as *experience replay* [45, 71], *architecture-based methods* [6, 23, 173, 225], and *re-warming* [71].

3.2.1 ***Architecture-Based Methods***. Architecture-Based Methods offer innovative solutions for enhancing the adaptability and efficiency of LLMs in continual language domain pretraining. Yadav et al. [225] improve prompt tuning by incorporating a teacher forcing mechanism, creating a pool of prompts that guide model finetuning on new tasks and compelling the model to follow specific pathways during training. Yang et al. [226] introduce the CLL-CLIP model, which extends the language understanding of CLIP [158] for continual learning of new languages. They employ Token Embedding Initialization and Regularization to mitigate catastrophic forgetting. CLL-CLIP includes an expandable token embedding layer that dynamically adjusts to accommodate linguistic differences, enabling seamless integration of new tokens. ModuleFormer [173] and Lifelong-MoE [23] are both architecture-based methods that utilize MoE to enhance LLM efficiency and adaptability. ModuleFormer leverages modularity by activating specific modules based on input tokens, ensuring targeted processing. Lifelong-MoE dynamically adds model capacity by incorporating new experts with regularized pretraining, achieving superior performance in few-shot and multi-task learning scenarios. These methods collectively demonstrate the potential of architectural innovations in addressing the challenges of continual learning.

3.2.2 ***Re-warming***. Re-warming, a strategy involving the temporary increase of the learning rate at the start of training on new data, allows the model to adapt more rapidly to new language. Ibrahim et al. [71] present a continual pretraining approach that combines learning rate (LR) re-warming, LR re-decaying, and data replay. In their method, LR re-warming is followed by LR re-decaying, a systematic reduction of the learning rate according to a specific schedule. This re-decaying phase helps the model stabilize after learning new language, preventing it from overfitting to recent data. This approach aligns with other methods in the field, such as those proposed by Gupta et al. [49], who emphasize the importance of adjusting learning rates to maintain model efficacy during continual vertical domain pretraining.

3.3 Continual Temporal Domain Pretraining

Continual Temporal Domain Pretraining [52, 74, 95, 119, 122, 248, 256] involves continually updating language models with temporally relevant data to maintain their accuracy and relevance as new information becomes available. Existing studies [95, 122, 164] highlight that the performance of LLMs degrades over time because they cannot learn new knowledge that is sensitive to temporal changes. For example, an LLM pretrained on 2023 data cannot answer questions about events that happened in 2024.

Empirical findings highlight several challenges in the realm of temporal adaptation for language models. Lazaridou et al. [95] demonstrate significant performance degradation when models trained on past data are tested on future data, underscoring the struggle of LLMs with temporal generalization. Similarly, Röttger et al. [164] reveal that while temporal adaptation offers slight improvements in masked language model tasks, it does not significantly enhance performance on downstream tasks when compared to domain adaptation alone. Moreover, Luu et al. [122] find that although continual pretraining aids temporal adaptation, it is less effective than task-specific fine-tuning on temporally relevant data, with performance degrading substantially over time. These studies collectively underscore the persistent challenges in achieving robust temporal generalization and the need for more sophisticated adaptation techniques.

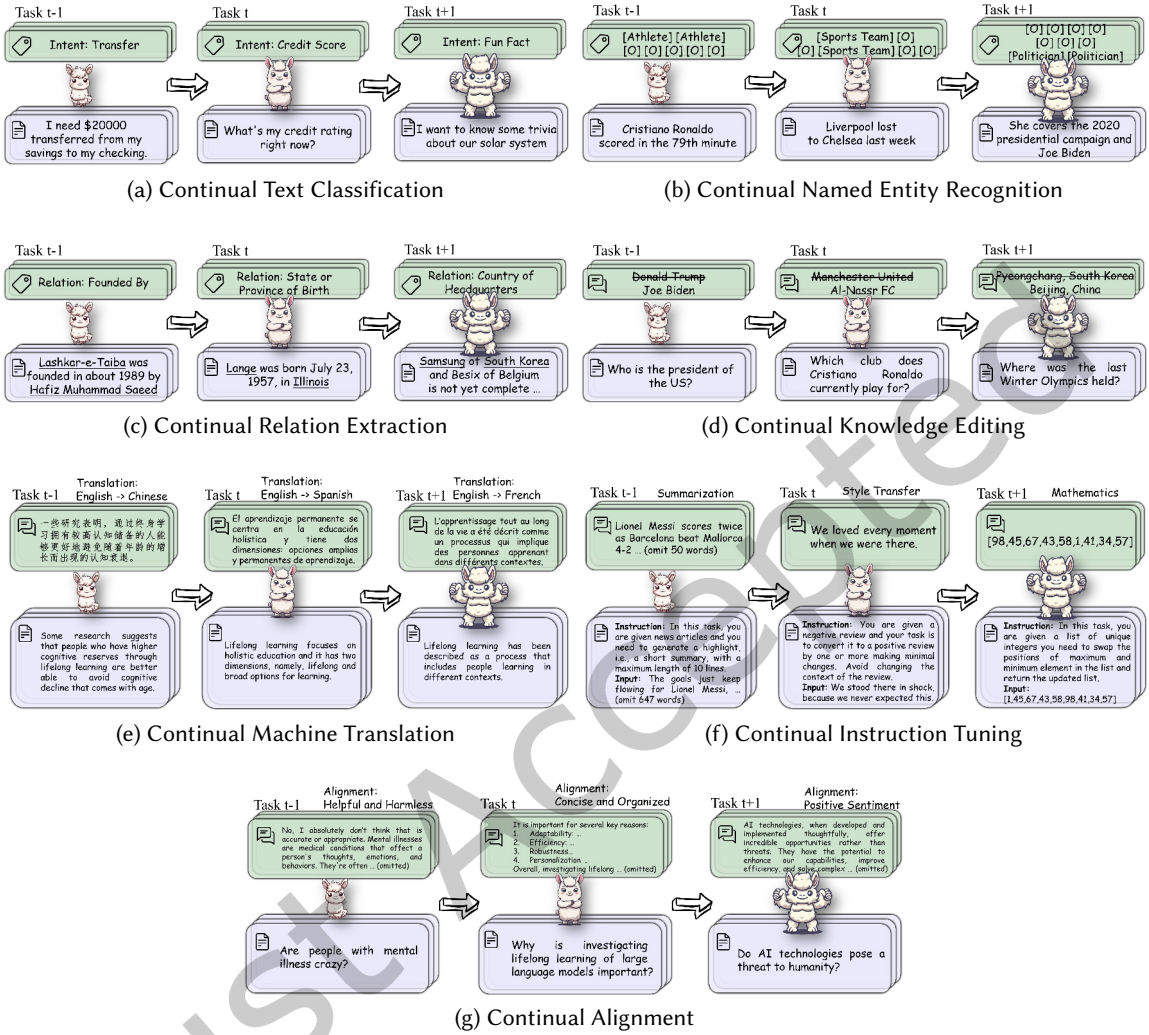
Most existing methods utilize experience replay to alleviate forgetting. In addition to experience replay, Han et al. [52] propose the Effective CONtinual pretraining framework for Event Temporal reasoning (ECONET), which integrates targeted masking and contrastive loss to emphasize event and temporal indicators during training. Specifically, ECONET employs a mask prediction strategy, where specific tokens related to events and times are masked, and a discriminator model is used to distinguish correct from corrupted sentences, thus enhancing temporal reasoning. Zhao et al. [256] introduce temporal-adaptive finetuning, which synchronizes the internal knowledge of the model with a target time without altering the explicit contextual information provided to the model. Complementing these approaches, TimeLMs [119] are continually updated language models trained on diachronic Twitter data to capture temporal changes in language and maintain relevance over time. Together, these methods demonstrate innovative strategies for addressing the challenges of continual learning and temporal adaptation in language models.

3.4 Summary

Continual pretraining enhances LLMs by updating their internal knowledge without incurring the high costs of full pretraining. Current research spans vertical, language, and temporal domains, addressing challenges like catastrophic forgetting and temporal adaptation. Techniques such as experience replay, knowledge distillation, parameter-efficient finetuning, model expansion, and re-warming have shown promise. Despite these advances, significant challenges remain, particularly in maintaining performance over time and across diverse tasks. Future research should focus on innovative approaches to mitigate forgetting, improve temporal generalization, and develop efficient, adaptive architectures for sustained model performance.

4 Methodology: Continual Finetuning

Continual finetuning [69, 110, 140, 182, 196, 204] enhances the internal knowledge of LLMs and adapts LLMs to specific tasks such as text classification [69], named entity recognition [140], relation extraction [196], machine translation [14] or general generation tasks such as instruction tuning [182], knowledge editing [204], and alignment with human preference [110]. We provide an illustration of the 7 continual finetuning scenarios in Figure 5. This aligns with the definition of lifelong learning in Section 2.1, which describes a sequence of tasks $\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \cdots, \mathcal{D}^{(T)}$, each characterized by specific input formats, such as instructions, or output formats, such as entity labels.



(a) Continual Text Classification

(b) Continual Named Entity Recognition

(c) Continual Relation Extraction

(d) Continual Knowledge Editing

(e) Continual Machine Translation

(f) Continual Instruction Tuning

(g) Continual Alignment

Fig. 5. An illustration of continual finetuning scenarios. In each continual finetuning scenario, a model learns task $t-1$, t, and $t+1$ sequentially (left to right). The **PURPLE** and the **GREEN** boxes represent the input and the output respectively.

4.1 Continual Text Classification

Text classification includes different directions, such as intent detection, sentiment classification, topic classification, and domain classification. However, past text classification methods can only detect predefined categories. In the real world, new categories may constantly challenge the deployed models. For example, the COVID-19 pandemic brought many new topic categories such as "nucleic acid detection" and "group immunity". Thus, the emergence of Continual Text Classification allows models to continuously learn new data and recognize new emerging categories. The methods can be broadly divided into the following main categories: *distillation-based*

Table 1. Comparison between representative methods for **continual text classification** and **continual named entity recognition**. **PEFT** represents whether utilize parameter-efficient finetuning methods for training models. **Replay, Regularization, Distillation, Architecture** refer to the common techniques summarized in Section 2.3. The full table is provided in the appendix.

Method	Year	Publication	Backbone	Dataset	Code	PEFT	Replay	Distillation	Regularization	Architecture	Others
Continual Text Classification											
EPI [210]	2023	ACL	BERT	AGNews, Yelp, Amazon, DBPedia, Yahoo, WOS	Link	Prefix Tuning	/	/	/	✓	/
VAG [169]	2023	ACL	BART	CLINC150, BANKING77, 20 Newsgroups	Link	/	Label-based Pseudo Replay	/	/	/	Vocabulary
LR ADJUST [213]	2023	ACL (Findings)	XLM-R	MASSIVE, WikiAnn	/	/	/	/	/	/	Adjusts Learning Rate
InfoCL [180]	2023	EMNLP	BERT	HWU64, FewRel, TACRED, MAVEN,	Link	/	✓	✓	/	/	Contrastive Learning
MoCL [199]	2024	NAACL	BERT, T5, LLaMA	WOS, AGNews, Yelp, Amazon, DBPedia, Yahoo	Link	LoRA, Prefix Tuning	/	/	/	✓	/
SEQ* [265]	2024	ACL	BERT, GPT2, Pythia	AGNews, Yelp, Amazon, DBPedia, Yahoo	Link	/	/	/	/	/	Classifier Expasion
HOP [135]	2024	AAAI	BERT	HL5Domains, Liu3Domains, Ding9Domains, SemEval14, NLI, 20News, DSC	/	Adapters	/	/	/	✓	/
Continual Named Entity Recognition											
SpanKL [251]	2023	AAAI	BERT	OntoNotes5, Few-NERD	Link	/	/	✓	/	/	Span-Level Prediction
OCILNER [123]	2023	ACL	BERT	CoNLL 2003, OntoNotes5, Few-NERD	Link	/	✓	/	/	/	Contrastive Learning, Prototype
ICE [114]	2023	ACL Findings	BERT	Few-NERD, MAVEN, ACE 2005	Link	/	/	/	/	✓	Frozen Backbones
RDP [242]	2023	CIKM	BERT	CoNLL 2003, OntoNotes5, I2B2	Link	/	/	✓	/	/	Prototype
CPFD [241]	2023	EMNLP	BERT	CoNLL 2003, OntoNotes5, I2B2	Link	/	/	✓	✓	/	/
SKD-NER [26]	2023	EMNLP	BERT	OntoNotes5, Few-NERD	/	/	/	✓	/	/	Reinforcement Learning
Liang et. al. [107]	2023	EMNLP (Findings)	BERT	ATIS, Snips	Link	/	✓	✓	/	/	Prototype
DLD [243]	2023	SIGIR	BERT	CoNLL 2003, OntoNotes5, I2B2	/	/	/	✓	/	/	/
IS3 [157]	2024	ACL (Findings)	BERT	OntoNotes5, I2B2, MAVEN	Link	/	/	✓	/	/	/
SEQ* [265]	2024	ACL	BERT, GPT2, Pythia	OntoNotes5, I2B2, Few-NERD	Link	/	/	/	/	/	Classifier Expasion

[85, 117], *replay-based* [7, 103, 116, 180, 190, 191], *regularization-based* [19, 64, 69, 149, 270], *architecture-based*, and others [16, 87, 145, 213, 219]. A detailed comparison between these methods is provided in Table 1.

4.1.1 ***Distillation-Based***. To enhance the discriminability between text categories, CLASSIC [85] employs contrastive ensemble distillation, enhancing knowledge transfer across tasks through contrastive losses. In addition this, multi-strategy rebalancing, combining cosine normalization, hierarchical knowledge distillation, and interclass margin loss are introduced by MSR [117] to tackle class imbalance.

4.1.2 ***Replay-Based***. Several approaches integrate contrastive learning techniques or structured learning methods to enhance the quality of replay samples and the stability of the learning process. SCN [116] and InfoCL [180] optimize sample selection and leverage contrastive learning for better representation recovery and to combat replay overfitting. These methods help maintain coherence and relevance of the learned representations, addressing issues like data imbalance and the presence of rare words in specific domains.

Each method incorporates adaptations tailored to specific domains or tasks, ensuring that the continual learning model effectively handles the unique challenges and characteristics of those domains. For example, DR-EMR [191] integrates social commonsense knowledge, and CRN [7] specifically targets the medical field's challenges, showing a deep integration of domain-specific knowledge into the learning processes.

Innovative memory management strategies such as DR-EMR [191], PAGeR [190] and the use of lightweight encoders PLE [103] with prefix guidance are employed to mitigate catastrophic forgetting and promote efficient knowledge retention. These strategies include selecting representative samples that best capture the essence of previous tasks and employing lightweight models that adapt more dynamically to new information without losing previous knowledge.

4.1.3 ***Regularization-Based***. To improve the efficiency of parameter updates, techniques such as selectively updating a small subset of parameters, as seen in the PE [270], IDBR [69], and EKFAC [19] prioritize reducing the computational burden. These methods ensure that the learning process is both resource-efficient and effective at integrating new knowledge without overwriting valuable information from previous tasks.

To automate the adjustment of regularization processes, several approches eliminate the need for manual hyperparameter tuning, allowing the model to adaptively balance retaining old knowledge with acquiring new information, as showcased in CCFI [64], Qian et al. [149].

4.1.4 ***Architecture-Based.*** To enhance knowledge sharing, there are several approaches to propose relevant strategies, such as hierarchical overlay projections in the HOP [135] and dynamic routing mechanisms in B-CL [87] and CTR [83], and ADA [38]. These strategies optimize the transfer and sharing of knowledge across different tasks, improving the efficiency and effectiveness of the model when learning new tasks.

To protect task-specific knowledge, several studies introduce mechanisms for parameter isolation, such as B-CL's [87] continual learning adapter, selective activation/deactivation of transformer components, instance-wise relation distillation in SCCL [121] and private parameter isolation in EPI [210]. These approaches effectively minimize interference between new and old tasks, maintaining performance on previous tasks while integrating new ones, thus addressing catastrophic forgetting.

4.1.5 ***Others.*** In addition to continual text classification tasks, there are also tasks focused on few-shot text classification and multilingual text classification, such as Pasunuru et al. [145] and ENTAILMENT [219] focus on improving few-shot learning capabilities, which involve training models with very few examples per class, CL-KD [16] and LR ADJUST [213] continually integrate new languages into an existing model, alleviating catastrophic forgetting in multilingual settings.

4.2 Continual Named Entity Recognition

Continual Named Entity Recognition is designed to adaptively identify novel entity types, addressing the dynamic emergence of new entities in the real world. It involves incrementally training models on newly annotated datasets that contain only these novel entities, enabling the models to gradually expand their recognition capabilities to include these new classes without forgetting previously learned entities. For example, in the sentence "Liverpool lost to Chelsea last week", a continual named entity recognition model aims to correctly label "Liverpool" and "Chelsea" as [Sports Team], while non-entity tokens are labeled as [Other]. This approach allows the model to adapt to recognize new entities such as [Politician] in other contexts.

In addition to the challenge of catastrophic forgetting, continual named entity recognition must also contend with semantic shifts [157, 262]. Semantic shift occurs when the classification of a label changes, for instance, from "Other" to a specific entity type, or vice versa. This is particularly challenging as only entities relevant to the current task are annotated, while both previously learned and unseen entities are labeled as "Other". Existing methods can be broadly classified into four primary categories: *distillation-based* [26, 140, 241–243, 251, 262], *replay-based* [13, 123, 222, 234], *prototype-based* [27, 93, 157], *architecture-based* [114, 172, 179]. A detailed comparison between these methods is provided in Table 1.

4.2.1 ***Distillation-Based.*** In general continual learning scenarios, feature-level knowledge distillation is commonly employed to impose *implicit* knowledge constraints on the student model in the feature space. In continual named entity recognition, knowledge distillation involves inputting new training examples into the teacher model and guiding the student model using the resulting logits. This effectively utilizes old samples from new training examples for *implicit replay*, thereby impose *explicit* knowledge constrains on the student model. As a pioneer work, ExtendNER [140], considering the realistic scenario of continuously emerging named entities, introduces knowledge distillation into named entity recognition to construct a framework for continuous named entity recognition. Subsequent methods, integrating knowledge distillation techniques, have been improved to address semantic shift caused by the "Other" entity type, such as DLD [243], RDP [242], CPFD [241], etc. In addition to this, some methods have introduced new perspectives or technologies. CFNER [262] establishes a causal framework [147, 264, 267] to link old and new knowledge, addressing noisy labels with curriculum learning. SpanKL [251] shifts the paradigm by modeling continual named entity recognition at the *span-level*, which reduces conflicts in labeling. SKD-NER [26] refines distillation by incorporating *reinforcement learning* to optimize the selection of temperature coefficients and weights for better soft label generation.

Table 2. Comparison between representative methods for **continual text relation extraction** and **continual machine translation**. **PEFT** represents whether utilize parameter-efficient finetuning methods for training models. **Replay, Regularization, Distillation, Architecture** refer to the common techniques summarized in Section 2.3. The full table is provided in the appendix.

Method	Year	Publication	Backbone	Dataset	Code	PEFT	Replay	Distillation	Regularization	Architecture	Others
Continual Relation Extraction											
ConPL [25]	2023	ACL	BERT	FewRel, TACRED	Link	Prompt Tuning	✓	/	/	/	Prototype
Xia et. al [220]	2023	ACL (Findings)	BERT	FewRel, TACRED	Link	/	✓	/	/	/	Adversarial Tuning
CEAR [260]	2023	ACL	BERT	FewRel, TACRED	Link	/	✓	✓	/	/	Contrastive Learning, Prototype
SCKD [207]	2023	ACL (Findings)	BERT	FewRel, TACRED	Link	/	✓	✓	✓	/	Data Augmentation
ICE [114]	2023	ACL (Findings)	BERT	TACRED	Link	/	/	/	/	✓	Frozen Backbones
ICA-Proto [75]	2023	EACL (Findings)	BERT, Glove	FewRel	/	/	/	/	/	/	Prototype
SEQ* [265]	2024	ACL	BERT, GPT2, Pythia	FewRel, TACRED	Link	/	/	/	/	/	Classifier Expasion
Continual Machine Translation											
CKD [252]	2023	ACL	Transformer	LDC, TED transcripts, Subtitles	Link	/	/	✓	/	/	/
KT [68]	2023	ACL	Transformer	WMT	Link	/	/	/	/	✓	/
BVP [113]	2023	EMNLP	mBART50-nn	WMT	Link	/	/	/	/	✓	Pruning
F-MALLOC [216]	2024	NAACL	Transformer	WMT	Link	/	/	/	/	✓	Pruning

4.2.2 ***Replay-Based***. Although continual named entity recognition is a token-level task, the stored replay samples are at the sentence level, incorporating contextual information about the entities.

To better utilize replay samples for reviewing old entities, several works have designed different methods to extract old knowledge. L&R [222] employs generative models to produce pseudo-samples that enhance the training with historical entity data. OCILNER [123] utilizes replay samples to calculate the centers of old-class entities and employs contrastive learning to cluster entities in the feature space, enhancing the discriminability between entities. KD+R+K [234] aggregates the feature representations of new and old entities based on their similarity, initializing representations for new entities and enhancing the associations between new and old entities. To improve *storage efficiency*, KCN [13] leverages the similarity between replay samples and class centers to gradually prune old samples that are far from the class centers while continuously adding new samples.

4.2.3 ***Prototype-Based***. Compared to replay-based methods, prototype-based approaches often employ clustering centers or class means to define prototypes, avoiding the direct use of old samples. This approach mitigates concerns about privacy and storage limitations to some extent. SDAPN [27] assigns portions of the feature space to new classes preemptively and uses the similarity between new samples and old class prototypes to correct biases. ProtoNER [93] replaces traditional linear classifiers with prototypes derived from the last hidden layer's feature vectors, refining the classification process. IS3 [157] combats semantic biases by integrating prototypes with a de-biased cross-entropy loss, ensuring the model does not disproportionately favor newer over older classes.

4.2.4 ***Architecture-Based***. Addressing the challenge of high resource costs associated with full model fine-tuning, architecture-based methods [114, 172, 179] focus on modifying the model structure to support continual learning without extensive retraining. ICE [114] maintains a static model backbone, using frozen classifiers for known entities and introducing new classifiers for emerging entities during training. At inference, these classifiers are unified to ensure comprehensive entity recognition. ConPET [179] employs distinct Parameter Efficient Tuning (PET) modules for each task, significantly reducing tuning overhead and minimizing both overfitting and forgetting.

4.3 Continual Relation Extraction

Continual Relation Extraction (CRE) entails updating relation extraction models to recognize new relationships while retaining accuracy on previously learned data. For instance, from the sentence "Lange was born July 23, 1957, in Illinois," a relation extraction system identifies the relationship between "Lange" and "Illinois" as "State or Province of Birth." The challenge is for the system to learn new relationships, like "Country of Headquarters," without forgetting existing ones. Apart from catastrophic forgetting, continual relation extraction confronts two challenges: (1) Order Sensitivity [28, 233]: This refers to the phenomenon where the performance of the model varies depending on the sequence of task introduction. (2) Interference of Analogous Relations [202, 260]: Challenges arise when the model confuses similar relations, such as "country of headquarters" and "state or province of headquarters."

In continual relation extraction, experience replay [32, 53, 62, 259] are widely favored due to their efficacy in managing both the acquisition of new information and the retention of old knowledge. Five popular techniques are combined with experience replay: *knowledge distillation* [207, 259, 260], *relation prototypes* [32, 53, 201, 246], *contrastive learning* [62, 124, 259, 260], *meta learning* [143, 217], *data augmentation* [124, 152, 202]. Table 2 provides a detailed comparison of these methods.

4.3.1 ***Knowledge Distillation.*** Focal Knowledge Distillation (FKD) is utilized by CEAR[260]. Specifically, FKD focuses on assigning higher importance to analogous relations, whereas SCKD [207] emphasizes serial distillation with pseudo-samples to bolster few-shot learning capabilities. In contrast, the focus on consistent relation representation learning across tasks makes CRL [259] align embedding in memory maintenance and ensure stability in the embedding space.

4.3.2 ***Relation Prototypes.*** Relation prototypes refer to a representation of relation in the feature space. As a pioneer work, EMAR [53] focus on utilizing relation prototypes for memory replay. Similarly, relation prototypes are used by RP-CRE [32] to refine sample embeddings. Inspired by EMAR [53] and RP-CRE [32], a more simplified variant in FEA [201] operates through the fast adaption and balanced tuning process. With the help of external knowledge, KIP-Framework[246] infuses prototypes with these knowledge to generate prototypes.

4.3.3 ***Contrastive Learning.*** The application of contrastive learning [72] varies from focusing on data distribution and embedding stability (CRECL [62] and CRL [259]) to addressing few-shot learning and overfitting challenges (CPL[124]), as well as enhancing the distinction of analogous relations (CEAR[260]). CRECL[62] uses a contrastive network, which contrasts a given instance with prototypes of each candidate relation stored in a memory module. For contrastive replay, it is used by CRL[259] to train memorized samples. Similarly, CEAR[260] utilizes contrastive learning alongside a linear method for training, where the former helps in improving feature space alignment and the latter ensures task-specific decision boundaries. Besides, a margin-based contrastive learning objective is introduced by CPL [124] to gain discriminative representations.

4.3.4 ***Meta Learning.*** To enable models to adapt quickly to new tasks while mitigating catastrophic forgetting, MLLRE [143] and CML [217] both use meta-learning frameworks. On the one hand, MLLRE[143] employs the REPTILE algorithm [142] for gradient-based meta-learning without second-order derivatives. On the other hand, CML[217] combines curriculum learning with meta learning to create a dynamic learning curriculum that prioritizes tasks based on difficulty. The main difference is that CML[217] focuses on task ordering and the difficulty in constructing learning curricula, while MLLRE[143] directly optimizes meta-objectives.

4.3.5 ***Data Augmentation.*** Data augmentation is leveraged to enrich the training data and improve model generalization across tasks, especially in low-resource settings. The majority of methods utilize external data[152] or generated samples[124, 202]. Adversarial examples are incorporated by ACA[202] to enhance model robustness and generalization. Besides, ERDA[152] selects informative samples from an unlabeled corpus that consists of

sentences from Wikipedia to provide more relational knowledge for few-shot tasks. With the help of large language models, CPL [124] guides them to generate diverse and relevant samples for memory augmentation.

4.4 Continual Machine Translation

Continual Machine Translation [14, 16, 42, 46, 67, 68, 168, 213, 249] is devised to cater to the demands of multilingual tasks in real-world scenarios, facilitating the addition of new languages over time. Continual machine translation typically undergoes training on a general domain corpus, encompassing a collection of various languages, followed by fine-tuning through continued training on an in-domain corpus specific to new languages. The objective is to learn the new language while retaining knowledge of the initial languages. Most of the methods for continual machine translation are single-step incremental language learning [14, 16, 42, 46, 67, 68, 168, 249], and a small number are multi-step incremental language learning [16, 213]. Several articles contribute to the field by proposing new benchmarks tailored to assess lifelong learning capabilities in multilingual contexts. Barrault et al. [8] provides training, lifelong, and test datasets for English-German and English-French to push forward research in lifelong learning NMT. Conversely, CLLE [247] introduces a Chinese-centric benchmark, featuring tasks that test a model's ability to handle closely related languages and diverse language families, reflecting real-world demands. Furthermore, Continual machine translation methods can be broadly classified into four primary approaches: *distillation-based* [14, 168, 252], *regularization-based* [46, 88, 113], *architecture-based* [9, 42, 67, 68, 216], and others [8, 39, 163, 247]. A detailed comparison between these methods is provided in Table 2.

4.4.1 ***Distillation-Based.*** Traditional NMT models are unable to handle continual or sequential learning problems without forgetting previously learned knowledge. Therefore, there are several methods innovating with different facets of dynamic knowledge distillation, such as Cao et al. [14] and CKD [252]. In addition to this, to address the unbalanced training problem, COKD [168] balances the model's focus across training samples, uniquely integrating dynamically updated teacher models.

4.4.2 ***Regularization-Based.*** To balance learning objectives on continual neural machine translation, there are many different implementations, such as regularizing the training process to minimize deviation from established models [88], identifying parameter updates that risk minimal forgetting [46], or categorizing parameters based on their relevance to specific tasks or overall functionality [113].

4.4.3 ***Architecture-Based.*** Architecture-based approaches in machine translation include *lexical structure* [9, 42, 67] and *model structure* [39, 68, 216].

Lexical structure refers to the set of unique tokens or words that an NMT model can recognize and generate. These tokens typically include words, subwords, or characters that the model uses to process and translate text from one language to another. EVS [67] optimizes embedding spaces by dynamically managing vocabularies based on their entropy values across languages, enhancing linguistic diversity without enlarging the model. Similarly, the method proposed by Garcia et al. [42] refines embedding efficiency by selectively substituting vocabulary parts, maintaining translation quality while integrating new languages efficiently.

Model structure innovations are highlighted by the introduction of dynamic resource allocation mechanisms and modular adaptation, which determines how effectively a model can handle different linguistic elements, especially when translating between multiple languages. F-MALLOC [216] introduces a memory allocation model that adapts to new languages by dynamically adjusting resources, thus supporting scalable and efficient learning. Concurrently, KT [68] integrates language-specific adapters into the NMT framework, facilitating seamless knowledge transfer and enabling the model to learn new languages without extensive retraining, thereby preserving its performance across a diverse linguistic spectrum.

Table 3. Comparison between representative methods for **continual instruction tuning**, **continual knowledge editing**, and **continual alignment**. **PEFT** represents whether utilize parameter-efficient finetuning methods for training models. **Replay, Regularization, Distillation, Architecture** refer to the common techniques summarized in Section 2.3. The full table is provided in the appendix.

Method	Year	Publication	Backbone	Dataset	Code	PEFT	Replay	Distillation	Regularization	Architecture	Others
Continual Instruction Tuning											
ACM [253]	2022	ACL	GPT-2	E2ENLG, RNNLG, WikiSQL, CNN/DailyMail, MultiWOZ	Link	Adapters	Pseudo Sample	/	/	✓	/
InstructionSpeak [232]	2022	ACL	BART	NaturalInstructions	/	/	✓	/	/	/	/
Continual Prompt Tuning [269]	2022	ACL	T5	Schema Guided Dialogue	Link	Prompt Tuning	✓	/	/	/	/
PCLL [261]	2022	EMNLP	GPT-2	DSTC, TOP	Link	/	Pseudo Sample	✓	/	/	Variational Auto Encoder
CT0 [166]	2022	EMNLP	T0	Simpl, HGen, Haiku, CQA, InqQG, EmDg, Exp, TwSt	Link	/	✓	/	/	/	/
LFPT5 [151]	2022	ICLR	T5	AGNews, Amazon Review, DBPedia, Yahoo, CNNDM, WikiHow, Xsum	Link	Prompt Tuning	Pseudo Sample	/	/	/	/
LPT [109]	2023	ACL	T5	CoNLL03, ACE05, SemEval-14	Link	Prompt Tuning	/	/	/	✓	Pruning
DYNAINST [139]	2023	ACL	BART	SuperNI	/	/	✓	/	/	/	/
HMI-LAMOL [127]	2023	EACL	GPT-2, BERT	AGNews, Yelp, Amazon, DBPedia, Yahoo	Link	/	Pseudo Sample	/	/	/	/
DMEA [150]	2023	EMNLP	GPT-2, BERT	CNN/DailyMail, MultiWOZ, WikiSQL	/	Adapters	/	/	/	/	/
O-LoRA [206]	2023	EMNLP (Findings)	LLaMA, T5	GLUE, SuperGLUE, IMDB	Link	LoRA	/	/	✓	✓	Orthogonal Subspaces
TSS [84]	2023	EMNLP (Findings)	BART	AGNews, Yelp, Amazon, DBPedia, Yahoo	Link	Adapters	/	/	/	✓	/
ProgPrompt [160]	2023	ICLR	T5, BERT	GLUE, SuperGLUE, IMDB	Link	Prompt Tuning	/	/	/	✓	/
SAPT [138]	2024	/	LLaMA, T5	SuperNI, GLUE, SuperGLUE, IMDB	/	Prompt Tuning, LoRA	Pseudo Sample	/	/	✓	/
InsCL [208]	2024	/	LLaMA	SuperNI	/	/	✓	/	/	/	/
I-LoRA [162]	2024	/	LLaMA	ScienseQA, MedMCQA, BBH, PIQA	Link	LoRA	✓	✓	/	✓	/
SSR [65]	2024	/	LLaMA, Alpaca	SuperNI	/	LoRA	Pseudo Sample	/	/	/	/
SLM [11]	2024	ICLR	LLaMA, T5, BERT	AGNews, Yelp, Amazon, DBPedia, Yahoo	Link	LoRA	/	/	/	/	/
Q-Tuning [48]	2024	NAACL (Findings)	BERT, T5	GLUE, SuperGLUE, IMDB	/	Prompt Tuning	/	/	/	✓	/
MoRAL [230]	2024	/	LLaMA, Phi	Arxiv, HotpotQA	/	LoRA	/	/	/	✓	/
Continual Knowledge Editing											
SLAG [56]	2023	EACL	BART, RoBERTa	zsRE, Wikidata5m, FEVER, LeapOfThought	Link	/	/	/	/	/	/
GRACE [55]	2023	ICLR	T5, BERT	zsRE, SCOTUS, Natural Questions	/	GRACE Adapters	/	/	/	✓	Codebook
TPatcher [70]	2023	ICLR	BART, BERT	zsRE, FEVER, CBQA	Link	/	/	/	/	✓	/
WilKE [61]	2024	/	GPT-J, GPT-2	CounterFact	/	/	/	/	/	✓	/
Continual Alignment											
Zhao et. al. [257]	2023	/	LLaMA, GPT-2	BBQ, Pile, HarmfulQA	/	LoRA	✓	/	/	/	Data Filtering, Self-Correction
CPPO [245]	2024	ICLR	LLaMA, GPT-2	HH-RLHF, Reddit TL;DR	Link	/	/	✓	/	/	/
COPR [244]	2024	/	LLaMA, GPT-J, OPT,	HH-RLHF, Reddit TL;DR, IMDB	Link	/	/	/	✓	/	/

4.5 Continual Instruction Tuning

The traditional machine learning paradigm for NLP assumed that target tasks were predefined and static, and that task supervision relied on labeled samples. This raises the question of how to build a system that can continuously learn new tasks from their instructions. Continual Instruction Tuning addresses this by designing various instructions for the same model to solve multiple NLP tasks. Earlier literature using GPT-2 [159] often used simple instructions like dataset names or special tokens [182]. In this survey, instruction tuning is defined more broadly, encompassing methods evaluated on a variety of generation tasks.

Chen et al. [18] proposes a comprehensive benchmark test, the Continuous Instruction tuNing (CoIN), to evaluate existing models in the sequential instruction tuning paradigm. CoIN evaluates two aspects: instruction following and general knowledge. It consists of 10 commonly used datasets spanning 8 task categories, ensuring a diverse range of instructions and tasks. Continual instruction tuning methods can be broadly classified into three primary approaches: *replay-based* [65, 80, 127, 151, 182, 183, 261], *regularization-based* [11, 77, 134, 206, 208], *gradient-based* [92, 97], and *architecture-based* [43, 48, 84, 109, 126, 160, 166, 205, 232, 253, 269]. A detailed comparison between these methods is provided in Table 3.

4.5.1 ***Replay-Based.*** The Replay-Based methods include the Generative Replay-Based method [65, 80, 127, 151, 182, 183, 261] and the Experience Replay-Based method [162].

Generative replay inspired by hippocampal memory mechanisms [175], this foundational paper introduces a novel approach by mimicking the human hippocampus, renowned for its role in memory formation and recall. The model efficiently retains prior knowledge while assimilating new information, setting a baseline for addressing catastrophic forgetting. Progressing from this foundation, LAMOL [182] embeds the generative replay directly within the language model. This integration simplifies the architecture and enables dynamic

pseudo-sample generation, enhancing memory consolidation without extra computational overhead. Further refining this approach, LFPT5 [151] utilizes prompt tuning to adapt quickly to new tasks with few examples, significantly reducing the data dependency and maintaining performance across tasks. Futhermore, there are several methods to improve the framework of Generative replay, such as PCLL [261], HMI-LAMOL [127], SSR [65]. A few approaches follow the conventional setting using experience replay, such as I-LoRA [162].

4.5.2 ***Regularization-Based.*** The regularization-based methods can be can be broadly categorized into direct [134, 206] and indirect [11, 77, 208] regularization approaches.

Direct regularization directly influencing model parameters to preserve prior learning. For instance, ARPER [134] integrates adaptive regularization directly into the training process, utilizing regularization terms that directly mitigate the forgetting of previously acquired knowledge during the learning of new dialogue tasks. Similarly, O-LoRA [206] employs an orthogonal low-rank adaptation (O-LoRA) method that directly constrains gradient updates to be orthogonal to the subspaces of previous tasks.

Indirect regularization utilizes factors such as similarity and importance between tasks to impose indirect restrictions on model parameters. For example, BiHNet [77] leverages a bi-level hypernetwork to create task-specific adapters, an architectural adjustment that indirectly preserves past knowledge by minimizing task interference. InsCL [208] utilizes dynamic replay of enriched data, indirectly facilitating continual learning by reintroducing crucial features of past tasks. Additionally, SLM [11] introduces a dynamic re-parameterization mechanism that adjusts the model's parameters according to the task distribution, ensuring that each task's learning is compartmentalized, thereby reducing the overwrite of important historical information.

4.5.3 ***Gradient-Based.*** In the realm of continual instruction tuning, effectively managing knowledge transfer and mitigating catastrophic forgetting are critical challenges that influence the robustness and versatility of language models. Some advances have focused on innovative gradient manipulation techniques to address these issues. Lee et al. [97] proposes a method that enhances gradient alignment across different tasks to promote better generalization and minimize negative transfer. Complementarily, Korbak et al. [92] introduces a framework for dynamically adjusting the learning parameters to preserve previously acquired knowledge during the fine-tuning process. Together, these methodologies underscore the potential of sophisticated gradient strategies to refine the adaptability of language models across diverse linguistic tasks without compromising their performance on previously learned information. Li et al. [104] mitigates forgetting by introducing the sharpness-aware minimization.

4.5.4 ***Architecture-Based.*** Architecture-based approaches can be categorized into *model-based* [43, 205], *adapter-based* [84, 126, 138, 150, 253] and *prompt-based* methods[48, 109, 160, 162, 166, 232, 269].

Model-based methods dynamically adjust the full network architecture in response to new information without requiring complete system retraining. For instance, TPEM [43] employs a cycle of pruning to eliminate less useful connections, expanding the network to accommodate new tasks, and masking to selectively deactivate certain pathways, ensuring that the system remains efficient and relevant to current tasks. Besides, Wang et al. [205] leverages an uncertainty estimation to decide when the system should update itself and an online learning component that facilitates immediate integration of new data into the model.

Adapter-based methods selectively adds new modules to manage knowledge retention and adaptability across sequential tasks. Several approaches allows the model to expand by dynamically adjusting and optimizing its architecture for each new task, such as ACM [253], DMEA [150] and so on. It incorporates new modules and adapts existing ones based on their performance and relevance to ongoing and past tasks, making the expansion process both targeted and efficient. In addition to this, SAPT [138] does not expand by adding new layers or modules in a conventional sense, but rather by utilizing a flexible attention mechanism to apply different sets of

parameters stored from previous tasks to new tasks. More recently, [37, 104] combine the idea of MoE and LoRA for adapting LLMs to knowledge-intensive tasks.

Prompt-based methods are essentially task-specific modifiers that guide the pre-trained language models in generating outputs that are appropriate for new tasks while retaining the capability to perform well on older tasks. This is achieved by strategically modifying or augmenting the input space of the models with prompts that encapsulate the essence of the task at hand, allowing the core model parameters to remain unchanged. For example, LPT [109] uses a binary prompt mask to selectively prune ineffective prompt vectors, enhancing computational efficiency and preserving crucial task-specific knowledge. Complementarily, DYNAINST [139] integrates a dynamic replay mechanism to selectively maintain training examples that improve learning efficiency, thereby optimizing knowledge retention across tasks. Further, ProgPrompt [160] innovates by sequentially concatenating task-specific prompts to accumulate knowledge and facilitate forward transfer without losing prior information. Together, these methods advance prompt-based strategies to boost the scalability and efficiency of lifelong learning in language models.

4.6 Continual Knowledge Editing

Continual Knowledge Editing serves as a pivotal component of lifelong learning for language models, designed to ensure their adaptability and accuracy as they encounter new information or discover that previous knowledge has become outdated [70]. Unlike traditional question-answering tasks that respond based on fixed knowledge, continual knowledge editing involves updating the model's understanding through knowledge triplets—structured data forms like *(head_entity, relation, tail_entity)*—which help in precisely defining the modifications needed in the model's knowledge base [204]. For instance, consider the triplet (Pluto, IsA, Planet), which may need updating to (Pluto, IsA, Dwarf Planet) as astronomical definitions evolve.

Research in this area has traditionally focused on one-step editing techniques [33, 34, 131, 132, 137], where models undergo *a single, significant update* to rectify or enhance their knowledge bases. However, more recent approaches [55, 56, 61, 70, 96] advocate for *a continual and sequential editing process*, aligning more closely with the principles of lifelong learning. This involves making multiple, smaller adjustments over time, allowing the model to adapt to the changing real-world requirements and maintain its relevance and accuracy without the need for comprehensive retraining.

Continual knowledge editing methods can be categorized into three main strategies [204]: *External Memorization*, *Global Optimization*, and *Local Modification*. A detailed comparison between these methods is provided in Table 3. (1) **External Memorization** methods like GRACE [55] and T-Patcher [70] use extension-based strategies to integrate new data [204]. GRACE, for example, employs key-value pairs to dynamically store new information, allowing the model to access the latest data without a full retraining cycle. T-Patcher, on the other hand, makes precise, targeted adjustments to model parameters to correct specific errors, similar to software patches fixing bugs, thus ensuring that the model's outputs remain accurate and current. (2) **Global Optimization** involves more comprehensive updates across the model's parameters, exemplified by SLAG [56], which uses intermediate fine-tuning strategies to carefully balance the integration of new information with the retention of existing knowledge [204]. This approach allows for gradual updates that refine the model's understanding without overwhelming the previously learned data. Lee et al. [96] further this concept by incorporating LoRA to focus on expanding specific parts of the model's architecture, minimizing the disruption to the overall system. (3) **Local Modification** focuses on making changes at a more granular level within the model, such as adjusting specific neurons or layers that are most relevant to the new information. WilKE [61] utilizes gradient-based strategies to precisely identify and modify the parts of the model that directly relate to outdated or incorrect information [204], enabling targeted updates that do not require extensive retraining but still ensure the model's growth in knowledge and capabilities.

4.7 Continual Alignment

Continual alignment in Large Language Models is essential to ensure that these models remain aligned with human values and societal norms throughout their lifecycle. Traditionally, alignment has been a *one-step* process where LLMs are aligned after pretraining and instruction tuning stages [171]. However, as the demands and expectations from AI systems evolve, it is becoming increasingly necessary to adopt a *multi-step* alignment approach [244, 245, 257], where models are realigned periodically to accommodate new ethical standards and societal values. The *alignment tax*, which refers to the trade-off between aligning models to human values and potentially compromising their general performance, is a critical consideration in this process [110].

Continual alignment can be categorized into two main areas: *value alignment* [110, 244, 245] and *security alignment* [148, 239, 257]. A detailed comparison between these methods is provided in Table 3. (1) In **Value Alignment**, the focus is on ensuring that the model's responses adhere to ethical guidelines without losing previously acquired capabilities. Techniques such as CPPO [245] implement weighting strategies to balance new ethical priorities with existing knowledge. COPR [244] addresses catastrophic forgetting in the context of value alignment by dynamically adjusting regularization based on both new and historical preferences. Meanwhile, Lin et al. [110] suggest model averaging to effectively manage the alignment tax, optimizing the balance between maintaining performance and adhering to updated values. (2) **Security Alignment** concentrates on safeguarding the integrity and security of the data processed by LLMs. It involves strategies to prevent the perpetuation of harmful information and protect against data leaks. Zhao et al. [257] have developed a forgetting filter technique that prioritizes the security of content during model updates. Zhan et al. [239] demonstrate the ease with which minimal fine-tuning can compromise established security measures, highlighting the ongoing need for robust protection mechanisms. To strengthen LLMs against potential misuse and evolving security threats, ongoing research, and methodological innovations are crucial, as noted by Lermen et al. [98] and Qi et al. [148]. These efforts ensure that as LLMs are aligned with new security protocols, they do not become vulnerable to novel forms of exploitation.

4.8 Summary and Quantitative Analysis

Building on continual pretraining, which enhances the internal knowledge of LLMs, continual finetuning further adapts these models to specific tasks such as text classification, named entity recognition, relation extraction, machine translation, and instruction tuning. Techniques like distillation, replay, regularization, architecture-based, and gradient-based methods are employed to address challenges like catastrophic forgetting and task interference.

As shown in Table 4, parameter-freezing methods, such as SEQ, exhibit superior average incremental accuracy (AIA) across multiple continual learning datasets, outperforming existing approaches. Notably, SEQ achieves the highest AIA in both the upper group (63.60) and the lower group (74.23) of datasets, highlighting its effectiveness in mitigating catastrophic forgetting and enhancing task adaptability. These results underscore that LLMs possess robust inherent knowledge acquired during pretraining, allowing them to adapt continually to specific tasks by fine-tuning a minimal subset of parameters. This finding suggests that future continual learning methods could leverage parameter-freezing strategies to optimize the trade-off between performance and computational efficiency.

5 Methodology: External Knowledge

Continual pretraining and finetuning are essential for the lifelong learning of LLMs. However, as LLMs grow larger and more powerful, two emerging directions have gained popularity for equipping LLMs with novel external knowledge without modifying their parameters. This survey considers Retrieval-Based and Tool-Based Lifelong Learning, as both are promising approaches for achieving lifelong learning in LLMs. An illustration is provided in Figure 6. This perspective aligns with the definition in Section 2.1, where a language model continuously

Table 4. The average incremental accuracy (AIA) of different methods in continual finetuning. Upper: the result is averaged over five continual datasets: Topic3Datasets, Clinic150, Banking77, FewRel, and TACRED; Lower: the result is averaged over three continual datasets: Few-NERD, OntoNotes5, and I2B2. We follow the settings in Zheng et al. [265].

	SEQ	SpanKL	OCILNER	ExtendNER	DLD	SelfTrain	RDP	CPFD	ICE_O	ICE_PLO	CFNER	SEQ*
AIA	20.89	42.67	44.08	43.23	47.20	45.59	53.96	54.14	47.19	44.15	52.44	63.60
Time	×1.0	×1.8	×0.7	×2.1	×2.2	×2.1	×2.5	×2.1	×0.6	×0.6	×3.8	×0.7

	SEQ	L2KD	LAMOL_KD	LAMOL_g	LAMOL_t	PCLL	SEQ*
AIA	35.18	59.52	61.99	63.93	66.07	64.33	74.23
Time	×1.0	×3.0	×2.0	×1.1	×1.2	×3.3	×0.4

Fig. 6. An illustration of two lifelong learning scenarios which equip LLMs with external knowledge: Retrieval-Based Lifelong Learning (left) and Tool-Based Lifelong Learning (right).

incorporates new knowledge by extending the input space of the next task $\mathbf{x}^{(t)}$ using Retrieval-Augmented Generation or enriching the output space $\mathbf{y}^{(t)}$ through the integration of external tools.

5.1 Retrieval-Based Lifelong learning

Why do LLMs need retrieval? Retrieval-based lifelong learning addresses the critical need for large language models to access and incorporate up-to-date knowledge from external sources [5, 81, 189]. As the world's information continues to expand and evolve rapidly, static models trained on historical data quickly become outdated, unable to comprehend or generate content about new developments. For example, consider a scenario where a significant medical breakthrough is announced after the model's last training update. In such cases, accessing real-time information from comprehensive databases or continuously updated platforms like Wikipedia becomes invaluable. These external sources offer a vast reservoir of current knowledge, presenting a vital complementary asset to enhance the static nature of pre-trained LLMs [218, 255].

How to retrieve? At the heart of implementing this approach is Retrieval-Augmented Generation (RAG), which synergistically combines the deep learning capabilities of LLMs with the dynamic retrieval of external data. RAG models operate by first fetching relevant information using a retriever component before generating text, thus ensuring the content is both updated and contextually appropriate. This process not only enriches the model's output but also significantly extends its applicability to newer domains and topics. We introduce several approaches that underscore the adaptability and effectiveness of retrieval-based methods as follows: Dense Passage Retrieval (DPR) [81] optimizes the retrieval process by encoding both queries and documents in a dense vector space, allowing for more accurate semantic matching. Interleaved Retrieval guided by Chain-of-Thought (IRCOT), as proposed by Trivedi et al. [189], embeds the retrieval step within the generative process. This approach dynamically adjusts the information retrieved as the response is being formed, which is particularly beneficial in complex dialogues or multi-turn interactions. Tree of Clarifications (TOC) developed by Kim et

al. [90] structures retrieved knowledge in a hierarchical tree format, enabling precise and relevant information retrieval at varying levels of query complexity. Active Retrieval in the form of Forward-Looking Active Retrieval augmented generation (FLARE) by Jiang et al. [76] proactively updates the retrieval database to include the latest information, ensuring the model's responses are timely and informed. Self-Reflective Retrieval-Augmented Generation (Self-RAG) by Asai et al. [5] utilizes a feedback loop where the model's output directly influences and refines future retrieval queries, promoting continuous self-improvement.

5.2 Tool-Based Lifelong Learning

Why do LLMs need tools? Tool-based lifelong learning for large language models (LLMs) stems from the necessity to extend their functionality beyond static knowledge and enable them to interact dynamically with their environment [66, 153, 154]. In real-world applications, it is often crucial for models to perform tasks that involve operations outside of straightforward text generation or interpretation. For example, an LLM tasked with providing real-time financial advice may need to access and process the latest stock market data, use analytical tools to predict trends or interact with databases to fetch client-specific information. Such scenarios require the model not only to understand and generate language but also to utilize external computational tools effectively, mirroring human capability in using tools to enhance cognitive tasks [4].

How to use tools? The development of tool-equipped LLMs, often referred to as "tool learning", transforms these models from static repositories of knowledge to dynamic systems capable of performing complex computational tasks and interacting with various APIs and software environments. This transformation is made possible through frameworks designed to teach LLMs how to integrate and utilize different tools effectively. For instance, Chameleon [120] synthesizes programs to tackle complex reasoning tasks by leveraging a combination of LLMs, visual models, search engines, and custom Python functions. Similarly, the ToolAlpaca framework [185] generates a diverse tool-use corpus through a multi-agent simulation environment, enhancing the model's general tool-use capabilities. Other notable frameworks include Confucius [41], which employs a multi-stage learning process coupled with feedback mechanisms to refine the tool-using proficiency of LLMs and GPT4Tools [229], which integrates multiple external tools to expand the functional reach of pre-trained models. Additionally, more complex tool datasets like APIBench [146] and ToolBench [154] have been developed to provide a structured environment for training and evaluating the tool-using capabilities of LLMs, broadening the boundary of what these models can achieve in practical applications.

5.3 Summary

Building on continual pretraining and finetuning, which enhance LLMs' internal knowledge, equipping LLMs with external knowledge through retrieval-based and tool-based lifelong learning significantly extends their capabilities. Retrieval-based methods ensure models remain updated by incorporating real-time information. Tool-based approaches enable LLMs to interact with external computational tools and APIs. Despite advancements, challenges persist in integrating these techniques seamlessly and efficiently. Future research should focus on refining retrieval mechanisms, improving tool integration frameworks, and developing comprehensive benchmarks to evaluate the effectiveness of external knowledge incorporation in LLMs.

6 Discussion and Conclusion

6.1 Existing Challenges

The journey towards optimizing lifelong learning for large language models faces a number of significant challenges that stem from the fundamental characteristics of these systems:

- **Catastrophic Forgetting**: This is a core challenge in lifelong learning, where newer information can overwrite what the model previously learned. As LLMs are continuously updated with new data, ensuring that they

retain valuable knowledge from past training without losing it to new and possibly unrelated information remains a critical issue [128].

- **Plasticity-Stability Dilemma**: Finding the right equilibrium between plasticity (the ability to learn new information) and stability (the ability to retain old information) is crucial [133]. This balance impacts the model's capacity to acquire domain-specific knowledge, such as medical information while preserving its broad-based general abilities. Additionally, the concept of alignment tax [110] highlights the challenge in training LLMs to align with human values without compromising their capabilities in areas like reasoning and planning. The objective is to enhance safety and alignment with ethical norms without diluting the model's functional effectiveness.
- **Expensive Computation Cost**: The computational demand of fully finetuning LLMs, especially for models with billions of parameters, can be prohibitively high.
- **Unavailability of Model Weights or Pretraining Data**: Often, the original training data or model weights are not available for further refinement due to privacy concerns [98, 239], proprietary restrictions, or commercial licenses.

6.2 Current Trends

As highlighted by the existing challenges, the evolution of lifelong learning for large language models is significantly influenced by the high computational costs of training these models and their robust capabilities. This has led to several new trends in how lifelong learning is approached:

- **From Specific to General Tasks**: There is a noticeable shift from focusing on specific tasks like text classification [85] and named entity recognition [140] to more general tasks that expand the model's utility across different domains. This transition towards general tasks such as instruction tuning [18] and knowledge editing [204] leverages the broad generalization ability of LLMs, allowing them to handle diverse challenges without intensive retraining for each specialized task.
- **From Full to Partial Finetuning**: Considering the substantial resources required to fully finetune LLMs, there is a growing preference for partial finetuning strategies. Techniques like Adapter layers [59], Prompt tuning [100], and LoRA [63] adjust only a small subset of parameters, preserving the core model while integrating the flexibility to adapt to new data and tasks efficiently.
- **From Internal to External Knowledge**: To overcome the limitations of frequent internal updates, there is a notable trend towards employing external knowledge sources. Strategies such as Retrieval-Augmented Generation [101] and tool-based learning [153] enable LLMs to access and utilize current external data dynamically. This approach not only enhances the model's problem-solving capacity but also ensures continual learning with minimal retraining.

6.3 Future Directions

As the capabilities of LLMs become stronger, the computational costs increase, and their applications broaden, future lifelong learning will aim to equip LLMs with more general abilities beyond the text modality, reduce computational costs, and address more realistic scenarios. Here are three promising areas of focus that could significantly advance the field:

- **Multimodal Lifelong Learning**: The integration of multiple modalities beyond text—such as images, videos, audio, time-series data, and knowledge graphs—into lifelong learning paradigms is a burgeoning area of research [18, 57]. This approach aims to develop more comprehensive and versatile models that can process and understand a broader array of data types, mirroring human-like learning capabilities across various sensory inputs.

- **Efficient Lifelong Learning**: To manage the computational demands of training and updating LLMs, researchers are looking towards more efficient strategies. These include leveraging model pruning [184] to eliminate unnecessary parameters, model merging [44] to consolidate knowledge, and model expansion [89, 215] to adaptively increase capacity without extensive retraining. Additionally, capitalizing on the in-context learning abilities of state-of-the-art LLMs, which support extensive contexts up to 10 million tokens, is seen as highly promising. For example, Gemini 1.5 Pro [161] showcases the potential by translating languages with high accuracy with only reference materials, mimicking a human learning context.
- **General Lifelong Learning**: The ultimate goal in this field is to enable LLMs to actively acquire new knowledge and learn through dynamic interactions with their environments, rather than solely from static datasets [197]. Incorporating principles from reinforcement learning, agent-based systems, and embodied AI could lead to the development of truly general AI. This ambitious direction seeks to emulate the natural lifelong learning capabilities of humans, facilitating a deeper, more intuitive engagement with the world.

6.4 Conclusion

In conclusion, this survey systematically categorizes existing studies into 12 lifelong learning scenarios and provides a comprehensive exploration of the methodologies. Our analysis highlights the delicate balance required to manage catastrophic forgetting, ensure computational efficiency, and maintain a balance between specificity and generality in knowledge acquisition. As the field continues to evolve, the integration of these advanced strategies will undoubtedly play a crucial role in shaping the next generation of AI systems, helping them closer to achieving true, human-like learning and adaptability.

Acknowledgments

The work described in this paper was partially funded by the National Natural Science Foundation of China (Grant No. 62272173), the Natural Science Foundation of Guangdong Province (Grant Nos. 2024A1515010089, 2022A1515010179), and the Science and Technology Planning Project of Guangdong Province (Grant No. 2023A0505050106). The icons used in this paper are downloaded from https://www.flaticon.com/ and are created by Iconjam, Freepik, Whitevector, Eucalyp, and Pixel perfect.

References

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

[2] Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. 2024. A Survey on Data Selection for Language Models. *arXiv:2402.16827* (2024).

[3] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *ECCV*. 139–154.

[4] Kelsey R Allen, Kevin A Smith, and Joshua B Tenenbaum. 2020. Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning. *PNAS* 117, 47 (2020), 29302–29310.

[5] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *ICLR*.

[6] Kartikeya Badola, Shachi Dave, and Partha Talukdar. 2023. Parameter-Efficient Finetuning for Robust Continual Multilingual Learning. In *Findings of ACL*. 9763–9780.

[7] Guirong Bai, Shizhu He, Kang Liu, and Jun Zhao. 2022. Incremental intent detection for medical domain with contrast replay networks. In *Findings of ACL*. 3549–3556.

[8] Loïc Barrault, Magdalena Marta Biesialska, Marta Ruiz Costa-Jussà, Fethi Bougares, and Olivier Galibert. 2020. Findings of the first shared task on lifelong learning machine translation. In *EMNLP 2020, Fifth Conference on Machine Translation*. 56–64.

[9] Alexandre Bérard. 2021. Continual Learning in Multilingual NMT via Language-Specific Embeddings. In *Proceedings of the Sixth Conference on Machine Translation*. 542–565.

[10] Magdalena Biesialska, Katarzyna Biesialska, and Marta R Costa-jussà. 2020. Continual Lifelong Learning in Natural Language Processing: A Survey. In *COLING*. 6523–6541.

[11] PENG Bohao, Zhuotao Tian, Shu Liu, Ming-Chang Yang, and Jiaya Jia. [n. d.]. Scalable Language Model with Generalized Continual Learning. In *ICLR*.
[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *NIPS* 33 (2020), 1877–1901.
[13] Pengfei Cao, Yubo Chen, Jun Zhao, and Taifeng Wang. 2020. Incremental event detection via knowledge consolidation networks. In *EMNLP*. 707–717.
[14] Yue Cao, Hao-Ran Wei, Boxing Chen, and Xiaojun Wan. 2021. Continual learning for neural machine translation. In *NAACL*. 3964–3974.
[15] Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*. 38–45.
[16] Giuseppe Castellucci, Simone Filice, Danilo Croce, and Roberto Basili. 2021. Learning to solve NLP tasks in an incremental number of languages. In *ACL and IJCNLP*. 837–847.
[17] Cheng Chen, Yichun Yin, Lifeng Shang, Xin Jiang, Yujia Qin, Fengyu Wang, Zhi Wang, Xiao Chen, Zhiyuan Liu, and Qun Liu. 2022. bert2BERT: Towards Reusable Pretrained Language Models. In *ACL*. 2134–2148.
[18] Cheng Chen, Junchen Zhu, Xu Luo, Hengtao Shen, Lianli Gao, and Jingkuan Song. 2024. CoIN: A Benchmark of Continual Instruction tuNing for Multimodel Large Language Model. *arXiv:2403.08350* (2024).
[19] Haolin Chen and Philip N Garner. 2024. Bayesian Parameter-Efficient Fine-Tuning for Overcoming Catastrophic Forgetting. *arXiv:2402.12220* (2024).
[20] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv:2107.03374* (2021).
[21] Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and Learn: Fine-tuning Deep Pretrained Language Models with Less Forgetting. In *EMNLP*. 7870–7881.
[22] Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. 2015. Net2net: Accelerating learning via knowledge transfer. *arXiv:1511.05641* (2015).
[23] Wuyang Chen, Yanqi Zhou, Nan Du, Yanping Huang, James Laudon, Zhifeng Chen, and Claire Cui. 2023. Lifelong language pretraining with distribution-specialized experts. In *ICML*. 5383–5395.
[24] Xuxi Chen, Zhendong Wang, Daouda Sow, Junjie Yang, Tianlong Chen, Yingbin Liang, Mingyuan Zhou, and Zhangyang Wang. 2024. Take the Bull by the Horns: Hard Sample-Reweighted Continual Training Improves LLM Generalization. *arXiv:2402.14270* (2024).
[25] Xiudi Chen, Hui Wu, and Xiaodong Shi. 2023. Consistent prototype learning for few-shot continual relation extraction. In *ACL*. 7409–7422.
[26] Yi Chen and Liang He. 2023. SKD-NER: Continual Named Entity Recognition via Span-based Knowledge Distillation with Reinforcement Learning. In *EMNLP*. 6689–6700.
[27] Yifan Chen, Zhan Huang, Minghao Hu, Dongsheng Li, Changjian Wang, Ankun Wang, Boyang Wang, and Xicheng Lu. 2022. Similarity-Driven Adaptive Prototypical Network for Class-incremental Few-shot Named Entity Recognition. In *IEEE ICTAI*. IEEE, 219–227.
[28] Zhiyuan Chen and Bing Liu. 2018. *Lifelong machine learning*. Vol. 1. Springer.
[29] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *JMLR* 24, 240 (2023), 1–113.
[30] Together Computer. 2023. *RedPajama: an Open Dataset for Training Large Language Models.* https://github.com/togethercomputer/RedPajama-Data
[31] Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. 2022. Continual pre-training mitigates forgetting in language and vision. *arXiv:2205.09357* (2022).
[32] Li Cui, Deqing Yang, Jiaxin Yu, Chengwei Hu, Jiayang Cheng, Jingjie Yi, and Yanghua Xiao. 2021. Refining sample embeddings with relation prototypes to enhance continual relation extraction. In *ACL and IJCNLP*. 232–243.
[33] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv:2104.08696* (2021).
[34] Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv:2104.08164* (2021).
[35] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence* 44, 7 (2021), 3366–3385.
[36] Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A Few-shot Named Entity Recognition Dataset. In *ACL and IJCNLP*. 3198–3213.
[37] Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. 2024. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In *ACL*. 1932–1945.
[38] Beyza Ermis, Giovanni Zappella, Martin Wistuba, Aditya Rawal, and Cedric Archambeau. 2022. Memory efficient continual learning with transformers. *NIPS* 35 (2022), 10629–10642.

[39] Carlos Escolano, Marta R Costa-jussà, and José AR Fonollosa. 2019. From Bilingual to Multilingual Neural Machine Translation by Incremental Training. In *ACL: Student Research Workshop*. 236–242.
[40] Falih Gozi Febrinanto, Feng Xia, Kristen Moore, Chandra Thapa, and Charu Aggarwal. 2023. Graph lifelong learning: A survey. *IEEE Computational Intelligence Magazine* 18, 1 (2023), 32–51.
[41] Shen Gao, Zhengliang Shi, Minghang Zhu, Bowen Fang, Xin Xin, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. 2024. Confucius: Iterative tool learning from introspection feedback by easy-to-difficult curriculum. In *AAAI*, Vol. 38. 18030–18038.
[42] Xavier Garcia, Noah Constant, Ankur Parikh, and Orhan Firat. 2021. Towards Continual Learning for Multilingual Machine Translation via Vocabulary Substitution. In *NAACL*. 1184–1192.
[43] Binzong Geng, Fajie Yuan, Qiancheng Xu, Ying Shen, Ruifeng Xu, and Min Yang. 2021. Continual Learning for Task-oriented Dialogue System with Iterative Network Pruning, Expanding and Masking. In *ACL and IJCNLP*. 517–523.
[44] Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's MergeKit: A Toolkit for Merging Large Language Models. *arXiv:2403.13257* (2024).
[45] Evangelia Gogoulou, Timothée Lesort, Magnus Boman, and Joakim Nivre. 2023. A study of continual learning under language shift. *arXiv:2311.01200* (2023).
[46] Shuhao Gu, Bojie Hu, and Yang Feng. 2022. Continual Learning of Neural Machine Translation within Low Forgetting Risk Regions. In *EMNLP*. 1707–1718.
[47] Jiafeng Guo, Changjiang Zhou, Ruqing Zhang, Jiangui Chen, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. CorpusBrain++: A Continual Generative Pre-Training Framework for Knowledge-Intensive Language Tasks. *arXiv:2402.16767* (2024).
[48] Yanhui Guo, Shaoyuan Xu, Jinmiao Fu, Jia Kevin Liu, Chaosheng Dong, and Bryan Wang. 2024. Q-Tuning: Queue-based prompt tuning for lifelong few-shot language learning. (2024).
[49] Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats Leon Richter, Quentin Gregory Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. 2023. Continual Pre-Training of Large Language Models: How to re-warm your model?. In *Workshop on Efficient Systems for Foundation Models@ ICML2023*.
[50] Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A Smith, and Luke Zettlemoyer. 2022. DEMix Layers: Disentangling Domains for Modular Language Modeling. In *NAACL*. 5557–5576.
[51] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *ACL*. 8342–8360.
[52] Rujun Han, Xiang Ren, and Nanyun Peng. 2020. Econet: Effective continual pretraining of language models for event temporal reasoning. *arXiv:2012.15283* (2020).
[53] Xu Han, Yi Dai, Tianyu Gao, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. Continual relation learning via episodic memory activation and reconsolidation. In *ACL*. 6429–6440.
[54] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation. In *EMNLP*. 4803–4809.
[55] Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2022. Aging with GRACE: Lifelong Model Editing with Key-Value Adaptors. (2022).
[56] Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2023. Methods for measuring, updating, and visualizing factual beliefs in language models. In *EACL*. 2714–2731.
[57] Jinghan He, Haiyun Guo, Ming Tang, and Jinqiao Wang. 2023. Continual instruction tuning for large multimodal models. *arXiv:2311.16206* (2023).
[58] Tianxing He, Jun Liu, Kyunghyun Cho, Myle Ott, Bing Liu, James Glass, and Fuchun Peng. 2021. Analyzing the forgetting problem in pretrain-finetuning of open-domain dialogue response models. In *EACL*. 1121–1133.
[59] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *ICML*. 2790–2799.
[60] Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. In *NAACL*. 57–60.
[61] Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. WilKE: Wise-Layer Knowledge Editor for Lifelong Knowledge Editing. *arXiv:2402.10987* (2024).
[62] Chengwei Hu, Deqing Yang, Haoliang Jin, Zhen Chen, and Yanghua Xiao. 2022. Improving Continual Relation Extraction through Prototypical Contrastive Learning. In *COLING*. 1885–1895.
[63] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*.
[64] Ting Hua, Yilin Shen, Changsheng Zhao, Yen-Chang Hsu, and Hongxia Jin. 2021. Hyperparameter-free Continuous Learning for Domain Classification in Natural Language Understanding. In *NAACL*. 2669–2678.
[65] Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating Catastrophic Forgetting in Large Language Models with Self-Synthesized Rehearsal. *arXiv:2403.01244* (2024).

[66] Jerry Huang, Prasanna Parthasarathi, Mehdi Rezagholizadeh, and Sarath Chandar. 2024. Towards Practical Tool Usage for Continually Learning LLMs. *arXiv:2404.09339* (2024).
[67] Kaiyu Huang, Peng Li, Jin Ma, and Yang Liu. 2022. Entropy-based vocabulary substitution for incremental learning in multilingual neural machine translation. In *EMNLP*. 10537–10550.
[68] Kaiyu Huang, Peng Li, Jin Ma, Ting Yao, and Yang Liu. 2023. Knowledge transfer in incremental learning for multilingual neural machine translation. In *ACL*. 15286–15304.
[69] Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. 2021. Continual Learning for Text Classification with Information Disentanglement Based Regularization. In *NAACL*. 2736–2746.
[70] Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2022. Transformer-Patcher: One Mistake Worth One Neuron. In *ICLR*.
[71] Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. 2024. Simple and Scalable Strategies to Continually Pre-train Large Language Models. *arXiv:2403.08763* (2024).
[72] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *Technologies* 9, 1 (2020), 2.
[73] Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2023. Exploring the benefits of training expert language models over instruction tuning. In *ICML*. 14702–14729.
[74] Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. TemporalWiki: A Lifelong Benchmark for Training and Evaluating Ever-Evolving Language Models. In *EMNLP*. 6237–6250.
[75] Wangjie Jiang, Zhihao Ye, Bang Liu, Ruihui Zhao, Jianguang Zheng, Mengyao Li, Zhiyong Li, Yujiu Yang, and Yefeng Zheng. 2023. Icaproto: Iterative cross alignment prototypical network for incremental few-shot relation classification. In *Findings of EACL*. 2275–2284.
[76] Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active Retrieval Augmented Generation. In *EMNLP*. 7969–7992.
[77] Xisen Jin, Bill Yuchen Lin, Mohammad Rostami, and Xiang Ren. 2021. Learn Continually, Generalize Rapidly: Lifelong Knowledge Accumulation for Few-shot Learning. In *Findings of EMNLP*. 714–729.
[78] Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2022. Lifelong Pretraining: Continually Adapting Language Models to Emerging Corpora. In *NAACL*. 4764–4780.
[79] Mladjan Jovanovic and Peter Voss. 2024. Trends and Challenges of Real-time Learning in Large Language Models: A Critical Review. *arXiv:2404.18311* (2024).
[80] Kasidis Kanwatchara, Thanapapas Horsuwan, Piyawat Lertvittayakumjorn, Boonserm Kijsirikul, and Peerapon Vateekul. 2021. Rational LAMOL: A rationale-based lifelong learning framework. In *ACL and IJCNLP*. 2942–2953.
[81] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *EMNLP*. 6769–6781.
[82] Zixuan Ke, Haowei Lin, Yijia Shao, Hu Xu, Lei Shu, and Bing Liu. 2022. Continual Training of Language Models for Few-Shot Learning. In *EMNLP*. 10205–10216.
[83] Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. 2021. Achieving forgetting prevention and knowledge transfer in continual learning. *NIPS* 34 (2021), 22443–22456.
[84] Zixuan Ke, Bing Liu, Wenhan Xiong, Asli Celikyilmaz, and Haoran Li. 2023. Sub-network Discovery and Soft-masking for Continual Learning of Mixed Tasks. In *Findings of EMNLP*. 15090–15107.
[85] Zixuan Ke, Bing Liu, Hu Xu, and Lei Shu. 2021. CLASSIC: Continual and Contrastive Learning of Aspect Sentiment Classification Tasks. In *EMNLP*.
[86] Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual Pre-training of Language Models. In *ICLR*.
[87] Zixuan Ke, Hu Xu, and Bing Liu. 2021. Adapting BERT for Continual Learning of a Sequence of Aspect Sentiment Classification Tasks. In *NAACL*. 4746–4755.
[88] Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. Regularized training objective for continued training for domain adaptation in neural machine translation. In *Proceedings of Workshop on Neural Machine Translation and Generation*. 36–44.
[89] Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, et al. 2023. Solar 10.7 b: Scaling large language models with simple yet effective depth up-scaling. *arXiv:2312.15166* (2023).
[90] Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of Clarifications: Answering Ambiguous Questions with Retrieval-Augmented Large Language Models. In *EMNLP*. 996–1009.
[91] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *PNAS* 114, 13 (2017), 3521–3526.
[92] Tomasz Korbak, Hady Elsahar, German Kruszewski, and Marc Dymetman. 2022. Controlling Conditional Language Models without Catastrophic Forgetting. In *ICML*. PMLR, 11499–11528.

[93] Ritesh Kumar, Saurabh Goyal, Ashish Verma, and Vatche Isahagian. 2023. ProtoNER: Few Shot Incremental Learning for Named Entity Recognition Using Prototypical Networks. In *International Conference on Business Process Management*. 70–82.
[94] Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In *EMNLP and IJCNLP*. 1311–1316.
[95] Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. 2021. Mind the gap: Assessing temporal generalization in neural language models. *NIPS* 34 (2021), 29348–29363.
[96] Kyungjae Lee, Wookje Han, Seung-won Hwang, Hwaran Lee, Joonsuk Park, and Sang-Woo Lee. 2022. Plug-and-Play Adaptation for Continuously-updated QA. In *Findings of ACL*. 438–447.
[97] Seanie Lee, Hae Beom Lee, Juho Lee, and Sung Ju Hwang. 2021. Sequential Reptile: Inter-Task Gradient Alignment for Multilingual Learning. In *ICLR*.
[98] Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. 2023. Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b. *arXiv:2310.20624* (2023).
[99] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. 2020. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion* 58 (2020), 52–68.
[100] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *EMNLP*. 3045–3059.
[101] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NIPS* 33 (2020), 9459–9474.
[102] Chen-An Li and Hung-Yi Lee. 2024. Examining forgetting in continual pre-training of aligned large language models. *arXiv:2401.03129* (2024).
[103] Guodun Li, Yuchen Zhai, Qianglong Chen, Xing Gao, Ji Zhang, and Yin Zhang. 2022. Continual few-shot intent detection. In *COLING*. 333–343.
[104] Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024. Revisiting Catastrophic Forgetting in Large Language Model Tuning. *arXiv:2406.04836* (2024).
[105] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *ACL and IJCNLP*. 4582–4597.
[106] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2017), 2935–2947.
[107] Chen Liang, Hongliang Li, Changhao Guan, Qingbin Liu, Jian Liu, Jinan Xu, and Zhe Zhao. 2023. Novel Slot Detection With an Incremental Setting. In *Findings of EMNLP*. 737–746.
[108] Yan-Shuo Liang and Wu-Jun Li. 2024. InfLoRA: Interference-Free Low-Rank Adaptation for Continual Learning. *arXiv:2404.00228* (2024).
[109] Zujie Liang, Feng Wei, Yin Jie, Yuxi Qian, Zhenghong Hao, and Bing Han. 2023. Prompts Can Play Lottery Tickets Well: Achieving Lifelong Information Extraction via Lottery Prompt Tuning. In *ACL*. 277–292.
[110] Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. 2023. Mitigating the Alignment Tax of RLHF. arXiv:arXiv:2309.06256
[111] Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, et al. 2024. Rho-1: Not All Tokens Are What You Need. *arXiv:2404.07965* (2024).
[112] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *NIPS* 35 (2022), 1950–1965.
[113] Junpeng Liu, Kaiyu Huang, Hao Yu, Jiuyi Li, Jinsong Su, and Degen Huang. 2023. Continual Learning for Multilingual Neural Machine Translation via Dual Importance-based Model Division. In *EMNLP*. 12011–12027.
[114] Minqian Liu and Lifu Huang. 2023. Teamwork Is Not Always Good: An Empirical Study of Classifier Drift in Class-incremental Information Extraction. In *Findings of ACL*. 2241–2257.
[115] Qingbin Liu, Pengfei Cao, Cao Liu, Jiansong Chen, Xunliang Cai, Fan Yang, Shizhu He, Kang Liu, and Jun Zhao. 2021. Domain-lifelong learning for dialogue state tracking via knowledge preservation networks. In *EMNLP*. 2301–2311.
[116] Qingbin Liu, Yanchao Hao, Xiaolong Liu, Bo Li, Dianbo Sui, Shizhu He, Kang Liu, Jun Zhao, Xi Chen, Ningyu Zhang, et al. 2023. Class Lifelong Learning for Intent Detection via Structure Consolidation Networks. In *Findings of ACL*. 293–306.
[117] Qingbin Liu, Xiaoyan Yu, Shizhu He, Kang Liu, and Jun Zhao. 2021. Lifelong intent detection via multi-strategy rebalancing. *arXiv:2108.04445* (2021).
[118] Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021. Benchmarking natural language understanding services for building conversational agents. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction: 10th International Workshop on Spoken Dialogue Systems*. 165–183.

[119] Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. TimeLMs: Diachronic Language Models from Twitter. In *ACL: System Demonstrations*. 251–260.
[120] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2024. Chameleon: Plug-and-play compositional reasoning with large language models. *NIPS* 36 (2024).
[121] Yun Luo, Xiaotian Lin, Zhen Yang, Fandong Meng, Jie Zhou, and Yue Zhang. 2023. Mitigating catastrophic forgetting in task-incremental continual learning with adaptive classification criterion. *arXiv:2305.12270* (2023).
[122] Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A Smith. 2022. Time Waits for No One! Analysis and Challenges of Temporal Misalignment. In *NAACL*. 5944–5958.
[123] Ruotian Ma, Xuanting Chen, Zhang Lin, Xin Zhou, Junzhe Wang, Tao Gui, Qi Zhang, Xiang Gao, and Yun Wen Chen. 2023. Learning "O" helps for learning more: Handling the unlabeled entity problem for class-incremental NER. In *ACL*. 5959–5979.
[124] Shengkun Ma, Jiale Han, Yi Liang, and Bo Cheng. 2024. Making Pre-trained Language Models Better Continual Few-Shot Relation Extractors. *arXiv:2402.15713* (2024).
[125] Shirong Ma, Shen Huang, Shulin Huang, Xiaobin Wang, Yangning Li, Hai-Tao Zheng, Pengjun Xie, Fei Huang, and Yong Jiang. 2023. Ecomgpt-ct: Continual pre-training of e-commerce large language models with semi-structured data. *arXiv:2312.15696* (2023).
[126] Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul A Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. 2021. Continual Learning in Task-Oriented Dialogue Systems. In *EMNLP*. 7452–7467.
[127] Aru Maekawa, Hidetaka Kamigaito, Kotaro Funakoshi, and Manabu Okumura. 2023. Generative Replay Inspired by Hippocampal Memory Indexing for Continual Language Learning. In *EACL*. 930–942.
[128] Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Vol. 24. Elsevier, 109–165.
[129] Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. 2023. An empirical investigation of the role of pre-training in lifelong learning. *JMLR* 24, 214 (2023), 1–50.
[130] Angelo G Menezes, Gustavo de Moura, Cézanne Alves, and André CPLF de Carvalho. 2023. Continual object detection: a review of definitions, strategies, and challenges. *Neural networks* 161 (2023), 476–493.
[131] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *NIPS* 35 (2022), 17359–17372.
[132] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022. Mass-editing memory in a transformer. *arXiv:2210.07229* (2022).
[133] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. 2013. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology* 4 (2013), 54654.
[134] Fei Mi, Liangwei Chen, Mengjie Zhao, Minlie Huang, and Boi Faltings. 2020. Continual Learning for Natural Language Generation in Task-oriented Dialog Systems. In *Findings of EMNLP*. 3461–3474.
[135] Umberto Michieli and Mete Ozay. 2024. HOP to the Next Tasks and Domains for Continual Learning in NLP. *arXiv:2402.18449* (2024).
[136] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *ACL*. 3470–3487.
[137] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv:2110.11309* (2021).
[138] Selection Module. [n. d.]. SAPT: A Shared Attention Framework for Parameter-Efficient Continual Learning of Large Language Models. ([n. d.]).
[139] Jisoo Mok, Jaeyoung Do, Sungjin Lee, Tara Taghavi, Seunghak Yu, and Sungroh Yoon. 2023. Large-scale lifelong learning of in-context instructions and how to tackle it. In *ACL*. 12573–12589.
[140] Natawut Monaikul, Giuseppe Castellucci, Simone Filice, and Oleg Rokhlenko. 2021. Continual learning for named entity recognition. In *AAAI*, Vol. 35. 13570–13577.
[141] Shawn N Murphy, Griffin Weber, Michael Mendis, Vivian Gainer, Henry C Chueh, Susanne Churchill, and Isaac Kohane. 2010. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *Journal of the American Medical Informatics Association* 17, 2 (2010), 124–130.
[142] Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv:1803.02999* (2018).
[143] Abiola Obamuyide and Andreas Vlachos. 2019. Meta-Learning Improves Lifelong Relation Extraction. In *Proceedings of the 4th Workshop on Representation Learning for NLP*. 224–229.
[144] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural networks* 113 (2019), 54–71.
[145] Ramakanth Pasunuru, Veselin Stoyanov, and Mohit Bansal. 2021. Continual few-shot learning for text classification. In *EMNLP*. 5688–5702.
[146] Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2023. Gorilla: Large language model connected with massive apis. *arXiv:2305.15334* (2023).

[147] Judea Pearl. 2009. Causal inference in statistics: An overview. (2009).
[148] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv:2310.03693* (2023).
[149] Jing Qian, Hong Wang, Mai ElSherief, and Xifeng Yan. 2021. Lifelong Learning of Hate Speech Classification on Social Media. In *NAACL*. 2304–2314.
[150] Chengwei Qin, CHEN CHEN, and Shafiq Joty. 2023. Lifelong Sequence Generation with Dynamic Module Expansion and Adaptation. In *Conference on EMNLP*.
[151] Chengwei Qin and Shafiq Joty. 2021. LFPT5: A Unified Framework for Lifelong Few-shot Language Learning Based on Prompt Tuning of T5. In *ICLR*.
[152] Chengwei Qin and Shafiq Joty. 2022. Continual Few-shot Relation Learning via Embedding Space Regularization and Data Augmentation. In *ACL*. 2776–2789.
[153] Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, et al. 2023. Tool learning with foundation models. *arXiv:2304.08354* (2023).
[154] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv:2307.16789* (2023).
[155] Yujia Qin, Cheng Qian, Xu Han, Yankai Lin, Huadong Wang, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. Recyclable Tuning for Continual Pre-training. In *Findings of ACL*. 11403–11426.
[156] Yujia Qin, Jiajie Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. ELLE: Efficient Lifelong Pre-training for Emerging Data. In *Findings of ACL*. 2789–2810.
[157] Shengjie Qiu, Junhao Zheng, Zhen Liu, Yicheng Luo, and Qianli Ma. 2024. Incremental Sequence Labeling: A Tale of Two Shifts. *arXiv:2402.10447* (2024).
[158] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*. 8748–8763.
[159] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).
[160] Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2022. Progressive Prompts: Continual Learning for Language Models. In *ICLR*.
[161] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530* (2024).
[162] Weijieying Ren, Xinlong Li, Lei Wang, Tianxiang Zhao, and Wei Qin. 2024. Analyzing and Reducing Catastrophic Forgetting in Parameter Efficient Tuning. *arXiv:2402.18865* (2024).
[163] Michele Resta and Davide Bacciu. 2024. Self-generated Replay Memories for Continual Neural Machine Translation. arXiv:arXiv:2403.13130
[164] Paul Röttger and Janet Pierrehumbert. 2021. Temporal Adaptation of BERT and Performance on Downstream Document Classification: Insights from Social Media. In *Findings of EMNLP*. 2400–2412.
[165] Anurag Roy, Riddhiman Moulick, Vinay K Verma, Saptarshi Ghosh, and Abir Das. 2024. Convolutional Prompting meets Language Models for Continual Learning. *arXiv:2403.20317* (2024).
[166] Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned Language Models are Continual Learners. In *EMNLP*. 6107–6122.
[167] Khadija Shaheen, Muhammad Abdullah Hanif, Osman Hasan, and Muhammad Shafique. 2022. Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks. *Journal of Intelligent & Robotic Systems* 105, 1 (2022), 9.
[168] Chenze Shao and Yang Feng. 2022. Overcoming Catastrophic Forgetting beyond Continual Learning: Balanced Training for Neural Machine Translation. In *ACL*. 2023–2036.
[169] Yijia Shao, Yiduo Guo, Dongyan Zhao, and Bing Liu. 2023. Class-Incremental Learning based on Label Generation. In *ACL*. 1263–1276.
[170] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2016. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *ICLR*.
[171] Tianhao Shen, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. 2023. Large language model alignment: A survey. *arXiv:2309.15025* (2023).
[172] Yilin Shen, Xiangyu Zeng, and Hongxia Jin. 2019. A progressive model to enable continual learning for semantic slot filling. In *EMNLP and IJCNLP*. 1279–1284.
[173] Yikang Shen, Zheyu Zhang, Tianyou Cao, Shawn Tan, Zhenfang Chen, and Chuang Gan. 2023. Moduleformer: Learning modular large language models from uncurated data. *arXiv:2306.04640* (2023).
[174] Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, and Hao Wang. 2024. Continual Learning of Large Language Models: A Comprehensive Survey. *arXiv:2404.16789* (2024).

[175] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. *NIPS* 30 (2017).
[176] Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature* 620, 7972 (2023), 172–180.
[177] James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. 2023. Continual diffusion: Continual customization of text-to-image diffusion with c-lora. *arXiv:2304.06027* (2023).
[178] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. 2023. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*. 11909–11919.
[179] Chenyang Song, Xu Han, Zheni Zeng, Kuai Li, Chen Chen, Zhiyuan Liu, Maosong Sun, and Tao Yang. 2023. Conpet: Continual parameter-efficient tuning for large language models. *arXiv:2309.14763* (2023).
[180] Yifan Song, Peiyi Wang, Weimin Xiong, Dawei Zhu, Tianyu Liu, Zhifang Sui, and Sujian Li. 2023. InfoCL: Alleviating Catastrophic Forgetting in Continual Text Classification from An Information Theoretic Perspective. In *Findings of EMNLP*. 14557–14570.
[181] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *NIPS* 33 (2020), 3008–3021.
[182] Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. LAMOL: LAnguage MOdeling for Lifelong Language Learning. In *ICLR*.
[183] Jingyuan Sun, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2020. Distill and replay for continual language learning. In *COLING*. 3569–3579.
[184] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A Simple and Effective Pruning Approach for Large Language Models. In *ICLR*.
[185] Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, and Le Sun. 2023. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *arXiv:2306.05301* (2023).
[186] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *NAACL*. 809–819.
[187] Zonggui Tian, Du Zhang, and Hong-Ning Dai. 2024. Continual Learning on Graphs: A Survey. *arXiv:2402.06330* (2024).
[188] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv:2302.13971* (2023).
[189] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In *ACL*. 10014–10037.
[190] Vaibhav Varshney, Mayur Patidar, Rajat Kumar, Lovekesh Vig, and Gautam Shroff. 2022. Prompt Augmented Generative Replay via Supervised Contrastive Learning for Lifelong Intent Detection. In *Findings of NAACL*. 1113–1127.
[191] Prashanth Vijayaraghavan and Deb Roy. 2021. Lifelong knowledge-enriched social event representation learning. In *EACL*. 3624–3635.
[192] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*. 59–63.
[193] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *NIPS* 32 (2019).
[194] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *ICLR*.
[195] Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. 2024. Self-Expansion of Pre-trained Models with Mixture of Adapters for Continual Learning. *arXiv:2403.18886* (2024).
[196] Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Sentence Embedding Alignment for Lifelong Relation Extraction. In *NAACL*. 796–806.
[197] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science* 18, 6 (2024), 1–26.
[198] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).
[199] Mingyang Wang, Heike Adel, Lukas Lange, Jannik Strötgen, and Hinrich Schütze. 2024. Rehearsal-Free Modular and Compositional Continual Learning for Language Models. *arXiv:2404.00790* (2024).
[200] Peihao Wang, Rameswar Panda, and Zhangyang Wang. 2023. Data efficient neural scaling law via model reusing. In *ICML*. 36193–36204.
[201] Peiyi Wang, Yifan Song, Tianyu Liu, Rundong Gao, Binghuai Lin, Yunbo Cao, and Zhifang Sui. 2022. Less is more: Rethinking state-of-the-art continual relation extraction models with a frustratingly easy but effective approach. *arXiv:2209.00243* (2022).
[202] Peiyi Wang, Yifan Song, Tianyu Liu, Binghuai Lin, Yunbo Cao, Sujian Li, and Zhifang Sui. 2022. Learning Robust Representations for Continual Relation Extraction via Adversarial Class Augmentation. In *EMNLP*. 6264–6278.
[203] Rui Wang, Tong Yu, Handong Zhao, Sungchul Kim, Subrata Mitra, Ruiyi Zhang, and Ricardo Henao. 2022. Few-shot class-incremental learning for named entity recognition. In *ACL*. 571–582.

[204] Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, et al. 2023. Knowledge editing for large language models: A survey. *arXiv:2310.16218* (2023).
[205] Weikang Wang, Jiajun Zhang, Qian Li, Mei-Yuh Hwang, Chengqing Zong, and Zhifei Li. 2019. Incremental Learning from Scratch for Task-Oriented Dialogue Systems. In *ACL*. 3710–3720.
[206] Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. 2023. Orthogonal Subspace Learning for Language Model Continual Learning. In *Findings of EMNLP*. 10658–10671.
[207] Xinyi Wang, Zitao Wang, and Wei Hu. 2023. Serial Contrastive Knowledge Distillation for Continual Few-shot Relation Extraction. In *Findings of ACL*. 12693–12706.
[208] Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. 2024. InsCL: A Data-efficient Continual Learning Paradigm for Fine-tuning Large Language Models with Instructions. *arXiv:2403.11435* (2024).
[209] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *EMNLP*. 5085–5109.
[210] Zhicheng Wang, Yufang Liu, Tao Ji, Xiaoling Wang, Yuanbin Wu, Congcong Jiang, Ye Chao, Zhencong Han, Ling Wang, Xu Shao, et al. 2023. Rehearsal-free continual language learning via efficient parameter isolation. In *ACL*. 10933–10946.
[211] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. In *CVPR*. 139–149.
[212] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Édouard Grave. 2020. CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*. 4003–4012.
[213] Genta Winata, Lingjue Xie, Karthik Radhakrishnan, Shijie Wu, Xisen Jin, Pengxiang Cheng, Mayank Kulkarni, and Daniel Preoţiuc-Pietro. 2023. Overcoming Catastrophic Forgetting in Massively Multilingual Continual Learning. In *Findings of ACL*. 768–777.
[214] Martin Wistuba, Lukas Balles, Giovanni Zappella, et al. 2023. Continual Learning with Low Rank Adaptation. In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*.
[215] Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ping Luo, and Ying Shan. 2024. Llama pro: Progressive llama with block expansion. *arXiv:2401.02415* (2024).
[216] Junhong Wu, Yuchen Liu, and Chengqing Zong. 2024. F-MALLOC: Feed-forward Memory Allocation for Continual Learning in Neural Machine Translation. *arXiv:2404.04846* (2024).
[217] Tongtong Wu, Xuekai Li, Yuan-Fang Li, Gholamreza Haffari, Guilin Qi, Yujin Zhu, and Guoqiang Xu. 2021. Curriculum-meta learning for order-robust continual relation extraction. In *AAAI*, Vol. 35. 10363–10369.
[218] Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *arXiv:2402.01364* (2024).
[219] Congying Xia, Wenpeng Yin, Yihao Feng, and Philip Yu. 2021. Incremental Few-shot Text Classification with Multi-round New Classes: Formulation, Dataset and System. In *NAACL*. 1351–1360.
[220] Heming Xia, Peiyi Wang, Tianyu Liu, Binghuai Lin, Yunbo Cao, and Zhifang Sui. 2023. Enhancing Continual Relation Extraction via Classifier Decomposition. In *Findings of ACL*. 10053–10062.
[221] Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv:2402.04333* (2024).
[222] Yu Xia, Quan Wang, Yajuan Lyu, Yong Zhu, Wenhao Wu, Sujian Li, and Dai Dai. 2022. Learn and review: Enhancing continual named entity recognition via reviewing synthetic samples. In *Findings of ACL*. 2291–2300.
[223] Jian Xie, Yidan Liang, Jingping Liu, Yanghua Xiao, Baohua Wu, and Shenghua Ni. 2023. Quert: Continual pre-training of language model for query understanding in travel domain search. In *KDD*. 5282–5291.
[224] Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. 2023. Efficient continual pre-training for building domain specific large language models. *arXiv:2311.08545* (2023).
[225] Prateek Yadav, Qing Sun, Hantian Ding, Xiaopeng Li, Dejiao Zhang, Ming Tan, Parminder Bhatia, Xiaofei Ma, Ramesh Nallapati, Murali Krishna Ramanathan, et al. 2023. Exploring Continual Learning for Code Generation Models. In *ACL*. 782–792.
[226] Bang Yang, Yong Dai, Xuxin Cheng, Yaowei Li, Asif Raza, and Yuexian Zou. 2024. Embracing Language Inclusivity and Diversity in CLIP through Continual Language Learning. *arXiv:2401.17186* (2024).
[227] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. FinGPT: Open-Source Financial Large Language Models. *FinLLM at IJCAI* (2023).
[228] Li Yang, Zhipeng Luo, Shiming Zhang, Fei Teng, and Tianrui Li. 2024. Continual Learning for Smart City: A Survey. *arXiv:2404.00983* (2024).
[229] Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. 2024. Gpt4tools: Teaching large language model to use tools via self-instruction. *NIPS* 36 (2024).

[230] Shu Yang, Muhammad Asif Ali, Cheng-Long Wang, Lijie Hu, and Di Wang. 2024. MoRAL: MoE Augmented LoRA for LLMs' Lifelong Learning. *arXiv:2402.11260* (2024).
[231] Çağatay Yıldız, Nishaanth Kanna Ravichandran, Prishruit Punia, Matthias Bethge, and Beyza Ermis. 2024. Investigating Continual Pretraining in Large Language Models: Insights and Implications. *arXiv:2402.17400* (2024).
[232] Wenpeng Yin, Jia Li, and Caiming Xiong. 2022. ConTinTin: Continual Learning from Task Instructions. In *ACL*. 3062–3072.
[233] Jaehong Yoon, Saehoon Kim, Eunho Yang, and Sung Ju Hwang. 2019. Scalable and Order-robust Continual Learning with Additive Parameter Decomposition. In *ICLR*.
[234] Pengfei Yu, Heng Ji, and Prem Natarajan. 2021. Lifelong event detection with knowledge transfer. In *EMNLP*. 5278–5290.
[235] Bo Yuan and Danpei Zhao. 2023. A Survey on Continual Semantic Segmentation: Theory, Challenge, Method and Application. *arXiv:2310.14277* (2023).
[236] Qiao Yuan, Sheng-Uei Guan, Pin Ni, Tianlun Luo, Ka Lok Man, Prudence Wong, and Victor Chang. 2023. Continual graph learning: A survey. *arXiv:2301.12230* (2023).
[237] Daoguang Zan, Bei Chen, Dejian Yang, Zeqi Lin, Minsu Kim, Bei Guan, Yongji Wang, Weizhu Chen, and Jian-Guang Lou. 2022. CERT: Continual Pre-training on Sketches for Library-oriented Code Generation. In *IJCAI*. 2369–2375.
[238] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. GLM-130B: An Open Bilingual Pre-trained Model. In *ICLR*.
[239] Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. 2023. Removing rlhf protections in gpt-4 via fine-tuning. *arXiv:2311.05553* (2023).
[240] Chenlong Zhang, Pengfei Cao, Yubo Chen, Kang Liu, Zhiqiang Zhang, Mengshu Sun, and Jun Zhao. 2024. Continual Few-shot Event Detection via Hierarchical Augmentation Networks. *arXiv:2403.17733* (2024).
[241] Duzhen Zhang, Wei Cong, Jiahua Dong, Yahan Yu, Xiuyi Chen, Yonggang Zhang, and Zhen Fang. 2023. Continual Named Entity Recognition without Catastrophic Forgetting. In *EMNLP*. 8186–8197.
[242] Duzhen Zhang, Hongliu Li, Wei Cong, Rongtao Xu, Jiahua Dong, and Xiuyi Chen. 2023. Task relation distillation and prototypical pseudo label for incremental named entity recognition. In *CIKM*. 3319–3329.
[243] Duzhen Zhang, Yahan Yu, Feilong Chen, and Xiuyi Chen. 2023. Decomposing logits distillation for incremental named entity recognition. In *SIGIR*. 1919–1923.
[244] Han Zhang, Lin Gui, Yu Lei, Yuanzhao Zhai, Yehong Zhang, Yulan He, Hui Wang, Yue Yu, Kam-Fai Wong, Bin Liang, et al. 2024. COPR: Continual Human Preference Learning via Optimal Policy Regularization. *arXiv:2402.14228* (2024).
[245] Han Zhang, Yu Lei, Lin Gui, Min Yang, Yulan He, Hui Wang, and Ruifeng Xu. [n. d.]. CPPO: Continual Learning for Reinforcement Learning with Human Feedback. ([n. d.]).
[246] Han Zhang, Bin Liang, Min Yang, Hui Wang, and Ruifeng Xu. 2022. Prompt-based prototypical framework for continual relation extraction. *IEEE/ACM TASLP* 30 (2022), 2801–2813.
[247] Han Zhang, Sheng Zhang, Yang Xiang, Bin Liang, Jinsong Su, Zhongjian Miao, Hui Wang, and Ruifeng Xu. 2022. CLLE: A benchmark for continual language learning evaluation in multilingual machine translation. In *Findings of EMNLP*. 428–443.
[248] Michael Zhang and Eunsol Choi. 2023. Mitigating Temporal Misalignment by Discarding Outdated Facts. In *EMNLP*. 14213–14226.
[249] Songming Zhang, Yunlong Liang, Shuaibo Wang, Yufeng Chen, Wenjuan Han, Jian Liu, and Jinan Xu. 2023. Towards Understanding and Improving Knowledge Distillation for Neural Machine Translation. In *ACL*. 8062–8079.
[250] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *NIPS* 28 (2015).
[251] Yunan Zhang and Qingcai Chen. 2023. A neural span-based continual named entity recognition model. In *AAAI*, Vol. 37. 13993–14001.
[252] Yuanchi Zhang, Peng Li, Maosong Sun, and Yang Liu. 2023. Continual Knowledge Distillation for Neural Machine Translation. In *ACL*. 7978–7996.
[253] Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. 2022. Continual Sequence Generation with Adaptive Compositional Modules. In *ACL*. 3653–3667.
[254] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *EMNLP*. 35–45.
[255] Zihan Zhang, Meng Fang, Ling Chen, Mohammad Reza Namazi Rad, and Jun Wang. 2023. How Do Large Language Models Capture the Ever-changing World Knowledge? A Review of Recent Advances. In *EMNLP*.
[256] Bowen Zhao, Zander Brumbaugh, Yizhong Wang, Hannaneh Hajishirzi, and Noah A Smith. 2024. Set the Clock: Temporal Alignment of Pretrained Language Models. *arXiv:2402.16797* (2024).
[257] Jiachen Zhao, Zhun Deng, David Madras, James Zou, and Mengye Ren. 2023. Learning and forgetting unsafe examples in large language models. *arXiv:2312.12736* (2023).
[258] Jiawei Zhao, Yifei Zhang, Beidi Chen, Florian Schäfer, and Anima Anandkumar. 2023. Inrank: Incremental low-rank learning. *arXiv:2306.11250* (2023).
[259] Kang Zhao, Hua Xu, Jiangong Yang, and Kai Gao. 2022. Consistent Representation Learning for Continual Relation Extraction. In *Findings of ACL*. 3402–3411.

[260] Wenzheng Zhao, Yuanning Cui, and Wei Hu. 2023. Improving Continual Relation Extraction by Distinguishing Analogous Semantics. In *ACL*. 1162–1175.
[261] Yingxiu Zhao, Yinhe Zheng, Zhiliang Tian, Chang Gao, Jian Sun, and Nevin L Zhang. 2022. Prompt Conditioned VAE: Enhancing Generative Replay for Lifelong Learning in Task-Oriented Dialogue. In *EMNLP*. 11153–11169.
[262] Junhao Zheng, Zhanxian Liang, Haibin Chen, and Qianli Ma. 2022. Distilling Causal Effect from Miscellaneous Other-Class for Continual Named Entity Recognition. In *EMNLP*. 3602–3615.
[263] Junhao Zheng, Qianli Ma, Zhen Liu, Binquan Wu, and Huawen Feng. 2024. Beyond Anti-Forgetting: Multimodal Continual Instruction Tuning with Positive Forward Transfer. *arXiv:2401.09181* (2024).
[264] Junhao Zheng, Qianli Ma, Shengjie Qiu, Yue Wu, Peitian Ma, Junlong Liu, Huawen Feng, Xichen Shang, and Haibin Chen. 2023. Preserving Commonsense Knowledge from Pre-trained Language Models via Causal Inference. In *ACL*. 9155–9173.
[265] Junhao Zheng, Shengjie Qiu, and Qianli Ma. 2023. Learn or Recall? Revisiting Incremental Learning with Pre-trained Language Models. *arXiv:2312.07887* (2023).
[266] Junhao Zheng, Shengjie Qiu, and Qianli Ma. 2024. Concept-1K: A Novel Benchmark for Instance Incremental Learning. *arXiv:2402.08526* (2024).
[267] Junhao Zheng, Ruiyan Wang, Chongzhi Zhang, Huawen Feng, and Qianli Ma. 2024. Balancing the Causal Effects in Class-Incremental Learning. *arXiv:2402.10063* (2024).
[268] Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. 2024. Continual Learning with Pre-Trained Models: A Survey. *arXiv:2401.16386* (2024).
[269] Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. 2022. Continual Prompt Tuning for Dialog State Tracking. In *ACL*. 1124–1137.
[270] Tao Zhu, Zhe Zhao, Weijie Liu, Jiachi Liu, Yiren Chen, Weiquan Mao, Haoyan Liu, Kunbo Ding, Yudong Li, and Xuefeng Yang. 2022. Parameter-efficient Continual Learning Framework in Industrial Real-time Text Classification System. In *NAACL: Industry Track*. 315–323.

Received 10 June 2024; revised 24 January 2025; accepted 3 February 2025