THE UNIVERSITY OF CHICAGO

EXTERNAL-MEMORY-BASED KNOWLEDGE EDITING

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE THESIS DIVISION
IN CANDIDACY FOR THE DEGREE OF
TYPE OF DEGREE

DEPARTMENT OF COMPUTER SCIENCE DEPARTMENT

BY
XIAOTIAN DUAN

CHICAGO, ILLINOIS
GRADUATION DATE

Dedication Text

Epigraph Text

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

ABSTRACT

Continual learning is essential in fields with evolving data distributions, such as recommendation systems, autonomous vehicles, and social media language processing. The increasing cost of training advanced neural networks, with their growing parameters and data needs, makes continual learning an attractive approach. A key challenge here is *catastrophic forgetting*, where models lose prior knowledge when exposed to new tasks. While various solutions exist, this issue remains unresolved.

Knowledge editing in Large Language Models (LLMs), closely related to continual learning, involves making targeted changes to specific data points. The aim is to fix inaccuracies or biases in these models without mistakenly altering other knowledge or skills. The research of knowledge editing focuses on three key aspects: *generalization* (the model's ability to apply edited information across various contexts), *locality* (precise edits without affecting unrelated information), and *scalability* (maintaining performance efficiency and stability with increasing edits). It's worth noting that the issue of locality is often a direct result of catastrophic forgetting.

This thesis introduces two novel contributions: (1) a new knowledge editing benchmark designed to overcome the limitations of existing benchmarks, which are inadequate for fine-tuning and lack comprehensive evaluations, and (2) a novel external-memory-based approach for knowledge editing that utilizes an embedding model with a vector store. Preliminary results show that compact embedding models can effectively differentiate the edited and unrelated facts, maintaining high accuracy even when scaled to thousands of edits. Integrating this approach with parameter-efficient fine-tuning strategies, this research aims to address all three critical aspects of knowledge editing: generalization, locality, and scalability. The findings of this study are poised to significantly advance knowledge editing in LLMs, contributing to continual learning and addressing pivotal challenges in modern machine learning, with potentially far-reaching impacts on the broader AI field.

CHAPTER 1

INTRODUCTION AND OVERVIEW

In an era where data constantly evolves, the ability of machine learning models to adapt and learn continuously is indispensable. Continual learning, particularly in fields with non-stationary data distributions, stands at the forefront of AI challenges, such as recommendation systems and autonomous vehicles. A primary challenge in this domain is overcoming catastrophic forgetting, wherein a model abruptly loses knowledge from previous tasks when exposed to new ones. The study of continual learning is frequently undertaken in the field of Computer Vision (CV), where one or more image classes are treated as a single task. The assessment of continual learning in this context involves sequentially training on these tasks and consistently evaluating the model's ability to maintain performance on tasks it has previously learned.

In the realm of Natural Language Processing (NLP), the challenges of continual learning take on an added dimension. The dynamic and ever-evolving nature of language, particularly evident in platforms like social media, necessitates LLMs that are capable not only of continuous learning but also of effectively addressing issues such as misinformation and biases. As LLMs increasingly become integral in disseminating information and facilitating decision-making, their ability to remain factually correct and unbiased is crucial for ethical applications. However, the sporadic nature of knowledge editing in LLMs and the opacity of their knowledge storage mechanisms pose the question:

Given the unclear nature of how knowledge is stored in LLMs, how do we precisely edit specific knowledge without impacting other knowledge and capabilities?

Knowledge editing in LLMs, a research field closely related to continual learning, emerges as a solution. Unlike fine-tuning, which is computationally demanding, knowledge editing aims to adjust a small subset of parameters or employ external memory/networks for edits.

However, existing methods fail to simultaneously achieve *generalization*, *specificity/locality*, and *scalability*. Consider the task of editing LLMs to recognize "Joe Biden as the US president in 2023." In this context, we must consider *generalization* to paraphrased or similar prompts, *specificity/locality* to ensure unrelated facts like "Donald Trump is the US president in 2020" remain unaffected, and textitscalability in terms of the number of allowable edits as well as the extra computational resources or memory required. Moreover, current benchmarks often fall short in comprehensive evaluation, suffering from an insufficient number of samples and patterns for both generalization and specificity/locality assessments.

To address these challenges, this work proposes two novel contributions:

A new benchmark for knowledge editing in LLMs. This new benchmark augments the existing benchmark in relation patterns, thereby multiplying the number of paraphrase prompts and neighbor prompts for separate generalization and specificity/locality evaluation. Additionally, the augmented pattern enables more effective fine-tuning by providing a much larger number of tokens for training, whereas previous benchmarks only have a single statement for fine-tuning, often yielding suboptimal results. The augmentation process leverages both traditional NLP methods, such as back translation, and modern LLMs, like GPT-3.5.

A novel method for knowledge editing in LLMs. This approach employs an additional embedding model to differentiate between prompts related to the edits and those that are not. A routing technique is then used: if a prompt is related to an edit, it is directed to the edited model; otherwise, the process remains unchanged. To minimize the additional memory and computational demands of each edit, we implement parameter-efficient fine-tuning methods, significantly reducing the number of parameters per edit. Preliminary results show promise in both embedding models (indicating effective specificity/locality) and parameter-efficient fine-tuning (suggesting improved generalization and scalability). By integrating these two components, we aim to achieve excellent generalization, specificity/locality,

and scalability simultaneously.

Furthermore, this proposal advocates for a more rigorous evaluation process, testing knowledge editing methods across diverse benchmarks. Existing research often overlooks the necessity for such comprehensive analysis, a gap we aim to fill. Given the complex architecture of LLMs, even minor parameter modifications during knowledge editing can lead to unforeseen consequences, including the risk of catastrophic forgetting beyond simple factual updates. Our work seeks to meticulously assess the impact of these modifications, using diverse benchmarks to evaluate the reasoning capabilities and stability of LLMs post-editing. This in-depth analysis is crucial for ensuring the reliability and effectiveness of knowledge editing techniques.

Following this introduction, the proposal will continue with a comprehensive chapter on related works, which can be found in chapter 2. This chapter will provide an in-depth review of the literature focusing on continual learning and knowledge editing in LLMs. Subsequently, in chapter 3, we will delve into the specific challenges and open questions within this field. The methods proposed to address these challenges, along with a discussion of our preliminary results and key observations, will be detailed in chapter 4. Finally, the proposal will conclude with chapter 5, which outlines the planned research timeline and the milestones we aim to achieve.

CHAPTER 2

BACKGROUND AND RELATED WORKS

This chapter delves into the recent advancements in the fields of continual learning and knowledge editing in LLMs. The landscape of continual learning in AI has evolved significantly over the years, marked by a variety of approaches aimed at mitigating catastrophic forgetting and enhancing adaptability in dynamic environments. These methods are typically evaluated using image classification benchmarks, where different classes are segmented into sequential tasks to measure the model's performance in continual learning scenarios.

More recently, the research of knowledge editing in LLMs has gained momentum, addressing the critical need for LLMs to stay factually correct and unbiased. Commonly used techniques in this area include the use of external memory and parameters with routing, locate-and-modify, and global optimization. To evaluate these methods, researchers have repurposed question-answering benchmarks and, more innovatively, employed datasets of counterfactual statements and their neighboring contexts. The latter presents more challenges due to the counterfactual nature of the information that requires editing.

In this chapter, we will examine the methodologies and evaluations in both fields, highlighting how they preserve learned knowledge while integrating new information, thereby paving the way for our proposed benchmark and method.

2.1 Continual Learning

2.1.1 Introduction

Continual learning, also known as lifelong learning or incremental learning, is a fundamental concept in machine learning and artificial intelligence where a model continuously learns over time by accommodating new knowledge while retaining previously learned information. This concept is crucial in dynamic environments where data continually evolves, such as

autonomous vehicles and surveillance in CV, and chatbots and language translation in NLP.

This chapter explores the fundamentals of continual learning, highlighting its practical implementation and the challenges it faces in both CV and NLP. We begin by defining continual learning and addressing its primary challenge: catastrophic forgetting, which is the tendency of a model to overwrite old information with new data. We then discuss various scenarios, related research fields, and key methodologies, such as rehearsal methods and dynamic architecture. Following this, we delve into its application in CV and NLP, examining notable research, common evaluation techniques, and potential future directions.

Finally, we discuss our contributions to the field, including an improved hard-attention-to-the-task mechanism and the investigation of alternative continual learning scenarios in computer vision. These contributions were recognized at the 4th continual learning workshop during the CVPR conference, where our strategy won the workshop challenge, underscoring its relevance and impact in the field.

2.1.2 Fundamentals of Continual Learning

Continual learning in machine learning is an approach where a model is progressively trained on a sequence of data over time, distinct from traditional static machine learning methods. This approach emulates the human capability to constantly acquire, refine, and transfer skills and knowledge over a lifetime. The realm of continual learning in AI is extensive, covering numerous scenarios, each presenting distinct challenges and requirements.

Definitions and Scenarios: The three primary scenarios in continual learning were initially proposed in van de Ven and Tolias [2018], and these definitions remain widely accepted:

1. **Task Incremental Learning (TIL)**: In this scenario, the model sequentially learns a variety of tasks, with the task IDs provided at test time. The primary focus is to prevent forgetting during the learning of new tasks. A strong baseline approach in TIL

is to train separate models for each task, however, it's often impractical in real-world applications due to resource constraints.

2. **Domain Incremental Learning (DIL)**: Here, the model encounters data from different domains or distributions, while the predictive target remains consistent. The task IDs are provided during training but not at test time. The primary focus is on generalizing the acquired knowledge across these domains.

3. **Class Incremental Learning (CIL)**: Similar to DIL, the model learns new tasks over time. However, the predictive target includes the task IDs, meaning the model must predict the target and infer which task the sample comes from. CIL, with its task-agnostic approach, closely resembles real-world applications where models may have to predict the domain shift during inference.

These three scenarios laid the foundation for continual learning, focusing on different aspects of the learning, with increasing difficulty. Beyond these initial scenarios, continual learning has evolved to include the following scenarios to further simulate real-world data dynamics better:

- **Online Continual Learning (OCL)**: Involves a continuous, one-pass stream of data, where distinct tasks are not explicitly separated, and task IDs are not available during both training and testing. This scenario, proposed in Aljundi et al. [2019b], represents a significant challenge in adapting to new data without the guidance of task identifiers.

- **Class-Incremental Learning with Repetition (CIR)**: Similar to traditional CIL, but tasks have overlapping data labels, and task IDs are available during training but not during testing. This scenario, proposed in Hemati et al. [2023], emphasizes the model's ability to handle ambiguities in data classification when task boundaries are not given during the inference.

- **Task-Free Continual Learning (TFCL)**: Involves tasks with disjoint data labels, with no task IDs provided during training or testing. This approach, proposed in Aljundi et al. [2019a], is particularly relevant for scenarios where task identification is either impossible or impractical.

- **Blurry CIL or General Continual Learning (GCL)**: Characterized by blurred task boundaries with overlapping data label spaces between tasks. Proposed in Buzzega et al. [2020] and Bang et al. [2021], this scenario challenges models to adapt to a less structured learning environment, closely mirroring many real-world applications.

There exists other scenarios focusing on niche applications, however the core objectives and challenges remain the same. Generally speaking, there are two main objectives in continual learning Lopez-Paz and Ranzato [2017]:

- *Backward transfer (BWT)* refers to the influence, positive or negative, that learning a task has on the performance on previous tasks. A negative BWT indicates catastrophic forgetting, while a positive BWT shows the newly learned knowledge helps with previous tasks.

- *Forward transfer (FWT)* refers to the influence that learning a task has on performance on future tasks. A positive FWT signifies zero-shrt learning, usually indicating useful features learned from past tasks.

Quantifying BWT and FWT involves evaluating the performance of a model on past and future tasks, respectively, and is crucial for understanding the effectiveness of a continual learning approach.

In continual learning, we aim to ensure the knowledge learned on any task has a positive influence on both previous and future tasks, regardless of the settings and scenarios. Given the myriad of scenarios in continual learning, it's important to understand their respective complexities and applicability. Furthermore, with advancements in fields like NLP, we may

anticipate the emergence of even more nuanced and complex continual learning scenarios. The diversity in continual learning scenarios reflects the multifaceted nature of real-world data and the evolving challenges in AI. Understanding these scenarios helps us develop more adaptable, efficient, and robust AI systems.

2.1.3 Challenges in Continual Learning

Continual learning offers significant opportunities for creating adaptive and dynamic AI systems, but also poses unique challenges that are crucial to address for effective and sustainable learning paradigms, especially in practical real-world deployments.

The primary challenge in continual learning is *catastrophic forgetting*. This significant issue, where learning a new task often results in a significant and abrupt degradation of performance on previously learned tasks, is observed in neural networks using back-propagation optimization McCloskey and Cohen [1989]. This effect typically occurs in scenarios where data distribution changes over time, such as learning on a sequence of different tasks. These scenarios defy the independent and identically distributed (IID) assumption of machine learning and invalidate conventional learning methods, necessitating the development of paradigms that operate under non-IID conditions.

Closely related to catastrophic forgetting, the *stability-plasticity dilemma* is central to continual learning. This challenge involves balancing the ability to learn new tasks (plasticity) with retaining knowledge of old tasks (stability) Mermillod et al. [2013]. These two abilities are often in conflict: excessive stability can hinder the learning of new information, while excessive plasticity may lead to rapid and severe forgetting. This dilemma is particularly challenging in scenarios where datasets from previous tasks are unavailable during the learning of new tasks, or in situations requiring learning from scratch. In such cases, it becomes difficult to leverage the power of retraining on previous samples or a pre-trained representation that generalizes well across all tasks.

Other challenges arise in applying continual learning to real-world scenarios. For example, the computational demands for updating parameters to balance stability and plasticity may be impractical in some cases. Scalability issues also emerge, particularly for methods that require storing samples from previously trained tasks. Additionally, defining the continual learning scenario itself poses a challenge, as there may be additional parameters and realistic considerations not covered in existing scenarios.

2.1.4 Related Areas

Continual learning, as a paradigm, intersects with numerous other fields in machine learning, each contributing uniquely to its core objectives. These fields address the challenges of adapting to new data, retaining, altering or forgetting previously learned information, and evolving in dynamic environments. This chapter delves into the intricate tapestry of related fields that are integral to continual learning. Each of these research areas not only contributes to the continual learning framework but also provides a unique lens through which we can examine and enhance the adaptability and intelligence of AI systems.

Multi-Task Learning (MTL) is a vital field closely related to continual learning Crawshaw [2020], Zhang and Yang [2021]. It involves training a model on multiple tasks simultaneously, allowing it to learn shared representations that are beneficial across different tasks. By learning commonalities between tasks, MTL methods can efficiently transfer knowledge across tasks, improving their overall performance and adaptability. This approach is particularly relevant to continual learning as it fosters the development of versatile models capable of handling a variety of tasks without needing separate training for each. Notably, the modular architecture of MTL is shared by continual learning methods, such as PathNet Fernando et al. [2017]. Moreover, MTL is often considered an upper bound for continual learning due to its unrestricted knowledge transfer in both directions.

Transfer Learning (TL) focuses on leveraging knowledge acquired from previously en-

countered tasks to enhance performance on the current task Zhuang et al. [2020]. Within the context of continual learning, knowledge transfer operates in both directions: FWT utilizes knowledge from past tasks to improve learning on the current task, while BWT strengthens the model's understanding of previous tasks after learning the current one. While continual learning is more concerned backward transfer and the mitigation of catastrophic forgetting, TL primarily emphasizes forward transfer and target task performance. However, specific approaches within TL, such as feature and parameter -based ones, remain highly relevant to continual learning. Feature representation transfer aims to discover representations that generalize across diverse tasks, while parameter transfer focuses on sharing parameters between different tasks, ultimately promoting efficient knowledge transfer and enhanced performance.

Meta-Learning, also described as "learning to learn", optimizes models to adapt to new tasks using past experiences Hospedales et al. [2021]. This methodology parallels continual learning's aim to adapt through a series of tasks, although it primarily focuses on rapid adaptation rather than addressing catastrophic forgetting. Nonetheless, several continual learning methods, such as MER Riemer et al. [2018], utilize meta-learning approaches, particularly incorporating replay samples to mitigate the negative impact of new tasks on previously acquired knowledge.

Selective Forgetting, also referred to as graceful or active forgetting, is an emerging field closely linked to continual learning Wang et al. [2023d]. This area focuses on strategically discarding outdated, inaccurate, or irrelevant information, a crucial process for maintaining a model's learning capacity and generalization while safeguarding critical past knowledge. By shedding light on how knowledge is stored within the model's parameters, selective forgetting offers valuable insights that can be leveraged to mitigate catastrophic forgetting in continual learning scenarios.

Knowledge Editing (in LLMs) involves actively updating a model's knowledge base to reflect new or corrected information Mazzia et al. [2023], Wang et al. [2023c]. Similar

to selective forgetting, this process helps identify and understand how knowledge is stored within the model. While encompassing both knowledge updating and continuous learning aspects, knowledge editing may not always involve explicit network training. Notably, the core objectives of these continual learning and knowledge editng exhibit significant overlap, albeit within distinct settings. Knowledge editing primarily focuses on specific edits without altering the model's predictive target, while continual learning often involves learning new tasks encompassing different domains or data labels. Knowledge editing methods and continual learning methods can mutually benefit from each other, particularly within the context of regularization-based approaches where both areas employ regularization to prevent the loss of previously acquired knowledge. A more comprehensive discussion of knowledge editing will be presented in Section 2.2.

In summary, these areas collectively contribute to the advancement of continual learning, each offering unique strategies to enhance AI systems' adaptability, knowledge retention, and task versatility.

2.1.5 Common Approaches in Continual Learning

Various methods have been proposed to address the challenges in continual learning, particularly focusing on backward transfer and mitigating catastrophic forgetting. We discuss common methodologies in continual learning, categorized into three approaches: regularization-based, replay-based, architecture-based.

Regularization-based Approach

These methods involves adding explicit regularization terms or manipulating the optimization process to accommodate both new and old tasks. They can be further divided into the following three subcategories:

Prior-Focused Weight Regularization: These methods focus on regularizing the up-

dates of network parameters to preserve knowledge from previous tasks. By applying a regularization or penalty term, they constrain the parameters deemed important for past tasks. For instance, Elastic Weight Consolidation (EWC) determine importance in Bayesian Framework based on the Fisher information matrix calculated after training each task Kirkpatrick et al. [2017]. Synaptic Intelligence (SI) approximates parameter importance by measuring the accumulated update distance during training Zenke et al. [2017]. Memory Aware Synapses (MAS) assesses importance by the sensitivity of the predicted output to changes in a parameter Aljundi et al. [2018]. Other variations such as Riemannian Walk Chaudhry et al. [2018a], which combines EWC and SI principles, maintain the same goal: to minimize changes in parameters crucial for the prediction of previous tasks. After assessing parameter importance, a quadratic penalty is usually added to the loss function to penalize significant variations in these important parameters.

Data-Focused Function Regularization: This approach aims to preserve previously acquired knowledge by focusing on the predictive function of the model. It involves setting constraints on the predictive outcomes of the intermediate or final layers of the network to ensure the model remains aligned with prior tasks, thereby preserving learned knowledge. This is typically achieved through various forms of knowledge distillation Hinton et al. [2015]. Notable among these is Learning without Forgetting (LwF), which proposes a training process that focuses on new task samples while simultaneously constraining the predictive output for samples from older tasks Li and Hoiem [2017]. To reduce the need for storing image samples and lower memory requirements, Learning without Memorizing (LwM) was developed. This method enables the model to emulate the attention patterns of a previously trained model (teacher model) during training on a new task Dhar et al. [2019]. Extending the basic concept of LwF, Encoder Based Lifelong Learning (EBLL) preserves the feature representations of task-specific autoencoders trained on previous tasks, in addition to employing knowledge distillation loss Rannen et al. [2017]. Deep Model Consolidation (DMC), involves training

a separate model for the new class and then consolidating it with the model trained on previous tasks, achieving this integration through knowledge distillation Zhang et al. [2020].

Gradient Projection: This method alters the optimization process directly, eliminating the need for additional objectives or regularization terms. A key example is Gradient Episodic Memory (GEM) Lopez-Paz and Ranzato [2017]. When training on a new task, GEM assesses the gradient of the loss function for the current task against the gradients of the samples from the previous tasks preserved in memory. It then adjusts the gradients to ensure that the learning process does not increase the loss for earlier tasks. Averaged Gradient Episodic Memory (A-GEM) relaxed this approach by using an average gradient derived from samples of previous tasks in memory, which reduces computational intensity Chaudhry et al. [2018b].

Replay-based Approach

Replay-based methods in continual learning focus on approximating and recovering old data distributions by using stored or synthetic samples or representations from previous tasks. By integrating the old data with the new during training, the model achieves replay or rehearsal, thereby mitigating forgetting. Replay-based approaches can be divided into two subcategories based on the nature of the replay samples:

Experience Replay: These methods store select training samples from previous tasks. The primary challenge lies in managing the limited memory buffer to adhere to the principles of continual learning, which discourages extensive reliance on past data. To fully utilize the replay samples, experience replay methods are rarely standalone solutions, but often paired with regulation-based methods for better results. The emphasis in experience replay is on the strategic selection of samples and efficient memory utilization. For instance, Incremental Classifier and Representation Learning (iCaRL) dynamically maintains a set of exemplars, chosen for their class representativeness, and incorporates them into new task

training Rebuffi et al. [2017]. It also utilizes distillation loss, integrating data-focused function regularization, and performs classification using Nearest-Mean-of-Exemplars, addressing the issue of unusable classification heads from previous classes due to network representation updates. Dark Experience Replay (DER) and its variant DER++ blend knowledge distillation with standard experience replay by storing and replaying not only past task samples but also the network's predictions on this data Buzzega et al. [2020]. Another notable method is Meta-Experience Replay (MER), which, like other replay methods, trains with a mixed batch of past task samples Riemer et al. [2018]. However, MER also incorporates meta-learning, assessing whether gradient updates impact performance on stored samples, similar to GEM and A-GEM. It's important to note that some literature such as Wang et al. [2023a] categorizes these methods as regularization-based due to their extensive use of replay samples during optimization.

Generative Replay or Pseudo Replay: Different from exact experience replay, this method does not necessitate explicit storage of past learning samples. Instead, replay samples are generated using advanced generative models such as variational autoencoder (VAE) or generative adversarial network (GAN) Kingma and Welling [2013], Goodfellow et al. [2014]. This is particularly advantageous when training samples cannot be stored, due to concerns like privacy, or the memory buffer size is too small for effective experience replay. Deep Generative Replay (DGR) laid the foundation for generative replay methods, using a GAN to continuously learn and generate replay samples, ensuring the classifier does not forget previous tasks Shin et al. [2017]. Memory Replay GANs (MeRGAN) recognized that sequential fine tuning of generative models introduces forgetting, and thereby employs replay alignment, which is essentially knowledge distillation, on the generative model, to prevent the forgetting of the generative model, and thereby improve the performance of the classifier Wu et al. [2018]. Additionally, FearNet uses a dual-memory system, akin to human memory systems, where long-term memory is consolidated by generative replay with a VAE Kemker

and Kanan [2017].

Architecture-based Approach

These methods rely on network architecture to prevent catastrophic forgetting. Often referred to as the parameter-isolation approach De Lange et al. [2021], these methods isolate different parts of the network for different tasks. By restricting parameter updates associated with previous tasks, they aim to significantly reduce or completely overcome catastrophic forgetting in TIL scenario. However, they face challenges in scenarios where task IDs are not available during testing, as it is challenging to select the appropriate network segment for an given sample. Generally speaking, there are two distinct approaches on how to isolate the network parameters:

Fixed Network Methods: In this approach, a fixed neural network structure is used, maintaining the same structure throughout the continual learning process for scalability. On the other hand, the learning capability is also limited. These methods segment the fixed network into dedicated sections for different tasks, often using task-specific binary masks. PathNet, for example, employs a genetic algorithm to find an optimal pathway for each task Fernando et al. [2017]. PackNet achieves this segmentation by pruning and retraining post-training to identify necessary parameters for the current task Mallya and Lazebnik [2018]. P iggyback, on the other hand, employs an end-to-end training approach on pre-trained networks, using the gradient of the mask threshold as binary masks are non-differentiable Mallya et al. [2018]. Hard Attention to the Task (HAT) differs by using trainable sigmoid masks with a scale parameter to simulate binary masks, training the masks and backbone parameters simultaneously, providing more task learning flexibility Serra et al. [2018].

Dynamic Architectures: This approach involves allocating extra parameters for each new task, focusing on efficient parameter reuse and knowledge transfer. Progressive Neural Networks (PNN) add new columns (subnetworks) for each task, connected laterally to pre-

vious columns, to leverage past learning Rusu et al. [2016]. This approach, however, leads to a parameter increase proportional to the number of tasks. Expert Gate also adopts a multi-column approach, with each column specializing in a task and a routing module acting as a gate Aljundi et al. [2017]. To control parameter growth, Dynamically Expandable Networks (DEN) selectively retrain parameters, building sparse connections between layers Yoon et al. [2017]. If performance is suboptimal, DEN expands by adding neurons. If the retrained parameters deviate significantly, they are duplicated, maintaining one set for previous tasks and updating the other for current tasks.

Each of these approaches presents distinct advantages and limitations. Currently, it is challenging to identify a universally superior method, as the effectiveness often depends on specific contextual factors. For example, replay-based approaches are straightforward and effective but may be resource-intensive, requiring significant memory for replay samples or generative models for pseudo replay. Conversely, architecture-based methods like HAT excel when task labels are available during inference, achieving zero forgetting while accommodating new learning. This diversity, coupled with various continual learning scenarios, underscores the field's complexity and dynamism.

Moreover, these approaches are not mutually exclusive. In fact, many of the methods discussed are combinations of different strategies. Generally, integrating various approaches often yields better results than standalone solutions. In addition to these strategies, many methods borrow ideas from related fields. Representation learning, for instance, enhances continual learning methods by providing generalized representations across tasks. This often bridges the gap, enabling architecture-based methods to perform task-agnostic predictions. For example, Continual Learning based on Out-of-Distribution (OOD) detection and Task Masking (CLOM) combines supervised contrastive learning (SupCon) Khosla et al. [2020] with HAT Serra et al. [2018]. This combination creates a feature space robust for OOD detection, facilitating task-agnostic inference Kim et al. [2022]. Methods like L2P Wang

et al. [2022b] and DualPrompt Wang et al. [2022a] utilize vision foundation models, such as pretrained Vision Transformers (ViTs) Dosovitskiy et al. [2020]. These models adapt based on prompting, creating a dynamic prompt pool that adjusts to different tasks and achieves state-of-the-art performance across multiple benchmarks. Notably, in methods like L2P and DualPrompt, the training process primarily involves updating the prompt pool, while the network remains unchanged, significantly minimizing forgetting during training.

Reflecting on these recent advancements, a prominent trend in the development of continual learning is the fusion of concepts from other disciplines, such as meta-learning and representation learning, while adapting to recent architectural innovations like transformers Vaswani et al. [2017].

2.1.6 Continual Learning in CV

Continual learning has seen extensive research in the domain of CV. The three commonly used scenarios of continual learning (TIL, DIL, and CIL) van de Ven and Tolias [2018] and most notable methods, such as EWC (Kirkpatrick et al. [2017]) , LwF (Li and Hoiem [2017]), and HAT (Serra et al. [2018]), were all proposed in the context of CV. This is partially due to the fact that image classification can be easily split into sequential tasks suitable for continual learning. Furthermore, image classification models, often trained from scratch, are easier to train and evaluate for continual learning.

The evaluation of continual learning methods in CV typically involves datasets like MNIST LeCun et al. [1998], CIFAR Krizhevsky et al. [2009], and TinyImageNet Le and Yang [2015], where images with different classes are split into separate tasks based on the scenario requirement. Average accuracy across the sequence of tasks is the most common metric, alongside forgetting measured by reductions in accuracy on previous tasks. For replay-based and dynamic architecture methods, additional metrics include memory consumption and computational requirements to ensure scalability. However, the focus of continual learn-

ing research in CV has primarily been on mitigating catastrophic forgetting, neglecting the evaluation for knowledge transfer, especially forward transfer (how knowledge gained from previous tasks benefits future tasks).

More nuanced scenarios, such as OCL (one-pass data stream) Aljundi et al. [2019b] and GCL (no task boundaries) Buzzega et al. [2020], Bang et al. [2021], have emerged and gained traction in this field. Additionally, the rise of vision foundation models like ViT Dosovitskiy et al. [2020] mark a shift towards pre-trained models in continual learning research for CV. The generalized representations learned by these models naturally lend themselves to continual learning due to their strong cross-task generalization capabilities. This evolution in CV sets a precedent for continual learning in NLP, where LLMs have recently become mainstream Bommasani et al. [2021].

2.1.7 Continual Learning in NLP

Continual learning is crucial for NLP tasks due to the rapidly evolving nature of language and constantly emerging new information. While it shares some similarities with continual learning in CV, continual learning in NLP presents unique challenges and opportunities.

- **Domain-incremental learning**: In NLP, continual learning predominantly focuses on domain-incremental learning, with or without task IDs. Unlike CV, where the predictive target varies with image classes, in NLP, it's the domain shift of input data that's pivotal, since the predictive target remains constant across tasks.

- **Scalability Issue**: NLP faces unique scalability challenges, such as the computational cost of processing large text corpora and the memory requirements for the training and inference of LLMs. These factors make some methods, like gradient projection or dynamic network structures, much less feasible in NLP compared to CV.

- **Pre-trained LLMs**: The rise of pre-trained LLMs like BERT and GPT-3 provides a

powerful foundation for NLP Bommasani et al. [2021]. These LLMs, with their generalizable representations of language, require less fine-tuning for specific tasks compared to traditional CV models. This enables researchers to focus on knowledge transfer and leverage the power of pre-training for continual learning. Additionally, the ability of LLMs to perform in-context learning paves the way for new continual learning approaches specifically tailored to them.

Several methods have been proposed for continual learning in NLP, often adapting ideas from CV Ke and Liu [2022]. For instance, Regularized Memory Recall with Domain Shift Estimation (RMR_DSE) employs EWC-based Kirkpatrick et al. [2017] regularization to address catastrophic forgetting Li et al. [2022a]. ExtendNER Monaikul et al. [2021], Lifelong Intent Detection (LID) Liu et al. [2021], and Continual Few-shot Intent Detection (CFID) Li et al. [2022b] utilize knowledge distillation, similar to LwF Li and Hoiem [2017]. Parameter isolation methods, such as BERT-based Continual Learning (B-CL) Ke et al. [2021] and Continual Post-Training (CPT) Ke et al. [2022], uses task-specific binary masks proposed in HAT Serra et al. [2018] to prevent catastrophic forgetting by assigning parameters for different tasks. Replay methods have also been adapted for NLP. For instance, Continual-T0 (CT0) reserves a small proportion of training data, typically up to 1%, from previous tasks for exact experience replay Scialom et al. [2022]. Prompt Conditioned VAE for Lifelong Learning (PCLL) employs a conditional variational autoencoder (CVAE) to generate representative samples for pseudo replay Zhao et al. [2022].

An approach unique to continual learning in NLP is the instruction-based approach, leveraging the in-context learning capability of LLMs. This approach, exemplified by Continual Learning from Task Instructions (ConTinTin) Yin et al. [2022], uses specific instructions for each task to enable knowledge transfer by sharing the same trained LLM, thus offering a novel pathway for continual learning.

Despite these advancements, a standardized benchmark for evaluating continual learning

in NLP is yet to be established. This highlights the ongoing need for comprehensive frameworks to assess these novel methodologies effectively. While accuracy is a common metric for continual learning in CV, it is often insufficient for NLP tasks that has different objectives, such as named entity recognition, summarizing, intent classification, etc.. Developing effective metrics for evaluating knowledge transfer and catastrophic forgetting remains an ongoing challenge. Currently, most evaluation strategies involve sequencing standard NLP tasks, a method that is often limited by the similarity of these tasks, complicating the assessment of catastrophic forgetting. A potential solution to this issue could be the creation of benchmarks that encompass a more diverse array of NLP tasks, varying in both similarity and complexity, to provide a more robust evaluation of continual learning methods.

The field of continual learning in NLP, particularly with LLMs, is rapidly advancing but faces notable challenges like catastrophic forgetting and scalability issue. Empirical studies reveal that LLMs often lose their broad knowledge base and reasoning skills during intensive domain-specific fine-tuning. This raises a critical question: how can we fine-tune LLMs on specific datasets without compromising their general capabilities? Addressing this question involves exploring strategies that maintain a balance between the specialized performance in specific domains and the preservation of the broad, generalist nature of LLMs. Additionally, delving into how these models scale with increasing data size or complexity is crucial. Overall, continual learning in NLP continues to be a dynamic research area, with significant potential for breakthroughs in model robustness, adaptability, and scalability.

2.1.8 Our Contributions

Our contributions to continual learning are twofold. First, we implemented a new Hard-Attention-to-the-Task (HAT) library called HAT-CL, which offers improved performance, versatile application, and seamless integration with existing frameworks and libraries. Second, we tackled the Class-Incremental Learning with Repetition (CIR) scenario, integrating

HAT-CL with other strategies to address the issue of catastrophic forgetting in such a realistic dynamic data environment.

HAT-CL: An Improved Hard-Attention-to-the-Task Implementation

To implement the forward pass of HAT, each weighted layer L is coupled with an embedding $\mathbf{e}_l^t$. These embeddings control the network's output during the forward pass by an attention mechanism, where the layer output $\mathbf{h}_l$ is modulated by a mask $\mathbf{a}_l^t$:

$$\begin{aligned} \mathbf{a}_l^t(s) &= \sigma(s\mathbf{e}_l^t) \\ \mathbf{h}'_l &= \mathbf{a}_l^t(s) \odot \mathbf{h}_l \end{aligned} \tag{2.1}$$

Here, the attention gate σ is represented by the sigmoid function, which modulates the mask's selectivity, referred to as the 'hardness' of the mask. During the forward pass, the scaling factor s is adjusted within the range of 0 to a predefined upper limit s_{max}. This adjustment is crucial as it balances the learning focus between the mask embeddings themselves $\mathbf{e}_l^t$ and the network weights modulated by these masks. A smaller s leads to a "smoother" mask, promoting the training of the mask embeddings, while a larger s results in a "harder" mask, focusing the training more on underlying weights.

During the backward pass, to prevent interference with previously learned tasks, we zero out the gradients of parameters associated with the previous tasks, by ultilizing the masks from the previous tasks. Specifically, a parameter is preserved (by nullifying the gradient) if it was not masked in any of the previous tasks. Mathematically, the process is represented as:

$$\begin{aligned} \mathbf{a}_l^{\leq(t-1)} &= \max\left(\mathbf{a}_l^{(t-1)}, \mathbf{a}_l^{\leq(t-2)}\right) \quad \text{, where } s = \infty \\ g'^t_{l,ij} &= \left[1 - \min\left(a_{l,i}^{\leq(t-1)}, a_{l-1,j}^{\leq(t-1)}\right)\right] g_{l,ij}^t \end{aligned} \tag{2.2}$$

Unlike the original implementation, where s takes a maximum finite value (s_{max}), we

propose setting s to infinity during the backward pass. This change yields strictly binary masks, enforcing complete isolation of task-specific parameters and reducing interference between tasks to absolute zero.

The HAT mechanism, as defined by equations (2.1) and (2.2), effectively isolates task-specific parameters and controls gradient flow to prevent interference from new tasks. To address the slow training of masks due to smaller gradients, Serra et al. [2018] introduce an adjusted gradient for the mask embeddings $q_{l,i}$:

$$q'_{l,i} = \frac{s_{\max}\Big[\cosh\left(se^t_{l,i}\right)+1\Big]}{s\Big[\cosh\left(e^t_{l,i}\right)+1\Big]}q_{l,i} \tag{2.3}$$

To ensure, numerical stability, Serra et al. [2018] clamps the mask embeddings $e^t_{l,i}$. And to ensure a balance between the training of mask embeddings and the associated weights, the scale of masks increases linearly from a really small number to $s_{\max}$ during each training epoch:

$$s = \frac{1}{s_{\max}} + \Big(s_{\max} - \frac{1}{s_{\max}}\Big)\frac{b-1}{B-1} \tag{2.4}$$

However, this original scaling strategy proposed in Serra et al. [2018] might result in slow convergence, especially in smaller neural networks, demonstrated in Duan [2023]. To prevent such issue, we propose to use the following scaling strategy to promote faster convergence:

$$s = \frac{s_{\max}}{2}\cdot\Big(1+\cos\left(p*2\pi\right)\Big) \tag{2.5}$$

where $p \in [0,1]$, indicating the progress in the current unit training time. The new scaling strategy, coupled with the dense initialization the embeddings (initialized to constant of 1s) ensures that the weights are properly aligned to the target before the training of masks, leading to much faster convergence, demonstrated in Duan [2023].

Moreover, our approach incorporates a layer-wise regularization term with a task quota.

This term encourages efficient utilization of masks across multiple tasks, ensuring each task contributes optimally to the model's knowledge base:

$$R_t = \sum_{l=0}^{L-1} \max\left(\frac{\sum_i a_{l,i}^t (1 - a_{l,i}^{<t})}{\sum_i (1 - a_{l,i}^{<t})} - \frac{1}{T}, 0\right) \tag{2.6}$$

Finally, our implementation leverages PyTorch hooks, offering a more flexible and streamlined approach. This allows for easier integration with various network architectures and simplifies the optimization process, making HAT-CL a versatile tool for continual learning research and applications.

Tackling CIR with HAT and Other Strategies

During the Computer Vision and Pattern Recognition (CVPR) continual learning workshop, we were awarded the first prize in the challenge focused on CIR Hemati et al. [2023]. This challenge aims to simulate more realistic learning scenarios by requiring models to learn on a stream of data with randomly reappearing classes, without forgetting previously acquired knowledge. Our strategy combines SupCon's contrastive learning Khosla et al. [2020], the HAT model Serra et al. [2018], Duan [2023], and a new technique we developed: momentum-based test-time logit averaging. The training is divided into two phases, similar to the approach in CLOM Kim et al. [2022].

Contrastive Learning Phase: In this phase, we train the HAT model using SupCon's contrastive loss function. This function aims to make the model's feature vectors more similar for the same class and more different for different classes. The loss function is defined as:

$$L = \sum_{i=1}^{N} \max\left(0, D(f(x_i^a), f(x_i^p)) - D(f(x_i^a), f(x_i^n)) + \alpha\right) \tag{2.7}$$

Here, $f(x)$ represents the HAT backbone model, generating an embedded feature vector for input x, which includes the training image and the task index. The HAT mechanism

segments the backbone model, isolating task-dependent parameters, and therefore preserving the exact weight for any specific task. $D(x, y)$ is a distance function, while x_i^a, x_i^p, and x_i^n denote the anchor, positive, and negative samples, respectively. α is a margin parameter, and N signifies the number of samples in a batch.

Classification Phase: Subsequently, the backbone is fine-tuned with a task-specific classification head. For each task t, the model focuses on the classes present in the current task, keeping the knowledge of other classes intact by freezing the corresponding classification head.

A key feature of our method is how we use the model's learned information from past tasks. After the model learns from each new task, we save its current state, which we call a "snapshot." These snapshots act as reference points for the model's knowledge at different stages. We use either the Hard Attention to the Task (HAT) approach or copies of the model to save these snapshots, depending on how much memory we have available. The HAT method is particularly useful here because it ensures that the specific parameters related to each task are kept intact. This means our model can accurately recall and reproduce the outcomes from any previous task. Additionally, our enhanced version of HAT, which we refer to as HAT-CL, has shown superior performance. This is mainly due to improvements in the initialization and scaling strategy. It's especially effective for smaller models and achieves better results in fewer training cycles, which is crucial for this challenge.

During the prediction, we carefully examined the generated logits from each snapshot. We discovered that the model's performance improves significantly when we average these logits from the most recent snapshots for each class. We call this method "momentum-based test-time logit averaging." To predict for a specific class, we take the logits from the latest snapshots that were trained on that class. We then calculate a weighted average, giving more importance to the most recent snapshots. This technique helps to counter any bias that might come from a single snapshot, which may not be fully representative if it was

trained on a limited set of classes.

By combining this logit averaging with other techniques, like test-time augmentation, our model's prediction accuracy significantly improved. This shows the effectiveness of our approach in managing the complex and changing nature of CIR in computer vision.

2.2 Knowledge Editing in LLMs

2.2.1 Introduction

LLMs have emerged as powerful tools in artificial intelligence, demonstrating remarkable capabilities in understanding and generating human language Brown et al. [2020], Bommasani et al. [2021]. LLMs are being applied in various domains, including reasoning, question answering, chatbot, document embedding, and more. As we rely increasingly on LLMs for daily tasks, the consequences of their errors become increasingly significant. A critical challenge lies in the frequent occurrence of outdated or inaccurate information within LLMs, often stemming from their outdated training data. The dynamic nature of information necessitates the development of effective techniques to update and correct such information.

Complete re-training of the entire LLM to update its knowledge base is often computationally expensive, time-consuming, and energy-intensive, posing a significant barrier to continuous improvement. This is especially true for large models with billions of parameters, where re-training can take days or even weeks. Knowledge editing offers a more efficient and resource-friendly alternative by directly modifying specific knowledge within the LLM's parameters. This ensures that LLMs remain current and provide accurate information, essential for their reliability and usefulness in various applications.

However, implementing knowledge editing is not without its challenges. Similar to continual learning, even minor modifications can potentially lead to performance degradation on previously learned tasks or trigger catastrophic forgetting McCloskey and Cohen [1989]. Nu-

merous empirical studies, such as Meng et al. [2022b], Wang et al. [2023b], Yao et al. [2023] have shown that fine-tuning on a single fact can significantly increase the LLM's probability to make mistakes on related, but unchanged, facts.

Further complicating matters, unlike traditional knowledge base systems that explicitly store and manage information, LLMs implicitly encode knowledge and tasks within their parameters. This makes it difficult to directly access, interpret, or alter their internal processes and memory without triggering unwanted changes, posing a significant barrier to accurate and effective knowledge editing.

This chapter delves into the intricacies of knowledge editing in LLMs. The next section delves into the historical and current research in knowledge editing, outlining key developments and contributions in the field. Then, we are going to discuss the evaluation and benchmarks used to assess the performance of knowledge editing techniques.

2.2.2 Related Works

Knowledge editing in LLMs has attracted growing attention in recent years. Existing approaches can be broadly categorized into three distinct categories based on how new knowledge is incorporated into the system: external memorization, global optimization, and local modification Wang et al. [2023c]. Each category exhibits unique advantages in addressing the challenges of knowledge editing in LLMs.

External Memorization Approach

External memorization methods employ additional memory components or parameters to store new information, thereby avoiding changes to the pre-trained model's weights.

- **Semi-Parametric Editing with a Retrieval-Augmented Counterfactual Model (SERAC)** exemplifies this approach, employing a scope classifier and a counterfactual model, both trained on edited knowledge Mitchell et al. [2022]. When a new prompt

comes in, SERAC will determine if the prompt is relevant to the edited knowledge by the scope classifier. If so, the completion task will be routed into a counterfactual model, which is another LLM but usually smaller compared to the original one.

- **In-Context Knowledge Editing (IKE)** utilizes a similar retrieval-augmented approach Zheng et al. [2023]. By prompting the LLMs with up to 32 demonstrations of the edited knowledge, they alter the completion trajectory and learn to edit their knowledge in context. This approach has been adapted in various works like MemPrompt Madaan et al. [2022] and Memory-based Editing for Large Language Models (MeLLo) Zhong et al. [2023] for user-instruction-based feedback and multi-hop question-answering (QA) scenarios.

- **Transformer-Patcher (T-Patcher)** diverges from the aforementioned memory-based methods, shifting the model's behavior by adding one neuron in the last feed-forward layer of the transformer model for each edit Huang et al. [2023]. This concept resembles other parameter-efficient fine-tuning methods like Low-Rank Adaptation (LoRA) Hu et al. [2021]. T-Patcher has also proven effective for sequential editing scenarios with thousands of edits.

- **General Retrieval Adaptors for Continual Editing (GRACE)** embeds a retrieval component within a transformer layer by recording edit-specific activations Hartvigsen et al. [2022]. When a prompt triggers a match with a stored activation, the corresponding adapter is activated, altering the network's behavior and completing the edit. GRACE has shown promising results in sequential editing scenarios.

Overall, the external memorization approach stands out for its simplicity, as it avoids direct modifications to the base model's parameter weights, which could be instrumental in mitigating catastrophic forgetting. This approach also exhibits strong scalability in sequential editing scenarios, such as the one proposed in Zhong et al. [2023], primarily due to the

ease of isolation of edits in external memory components or parameters. However, this approach is not without its drawbacks. The addition of extra memory elements or parameters can lead to increased computational cost and routing demands, potentially prolonging inference times. Moreover, its impact on language processing remains for further investigation, particularly in complex scenarios.

Global Optimization Approach

Global optimization approach in knowledge editing involves updating all or a significant portion of the parameters in LLMs. This approach does not specifically locate where knowledge is stored within LLMs, but rather treats the model as a holistic entity during the editing process.

- **Fine-tuning (with Constraints)** A common method in this category is fine-tuning. Naive fine-tuning could serve as a baseline for knowledge editing, such as Zheng et al. [2023]. However, it often leads to suboptimal knowledge retention, as related but unedited knowledge could be affected. To address this issue, L_2 or L_∞ constraints are introduced to regularization the updates in the model's parameters Mazzia et al. [2023], Meng et al. [2022b]. Additionally, methods employing the Kullback–Leibler (KL) divergence in loss functions have been used to regulate the network's weights by reducing the divergence between the network's output on prior and new tasks Mazzia et al. [2023], Mitchell et al. [2021]. Another strategy involves limiting updates to specific layers or components in the LLMs. For instance, fine-tuning a single feedforward network (FFN) layer in LLMs has proven to be a stronger baseline compared to naive fine-tuning, as seen in Meng et al. [2022b], Huang et al. [2023].

- **KnowledgeEditor** is a meta-learning-based method De Cao et al. [2021]. It trains a bidirectioanl-LSTM Schmidhuber et al. [1997] as a hypernetwork Ha et al. [2017] via

constrained optimization to predict the weight update for a specific FFN layer of the LLMs at the inference time. However, it has been primarily experimented with single batch edits, as sequential edits with accumulated updates from the meta-learner can deviate the model significantly from its original weights.

- **Model Editor Networks with Gradient Decomposition (MEND)** implements a method similar to KnowledgeEditor but utilizes a collection of multilayer perceptrons (MLPs) instead of a single meta-learner Mitchell et al. [2021]. Another significant difference is its use of low-rank decomposition and parameterization of the gradients, rather than direct gradient updates, which yields better results on multiple datasets.

Despite these advancements, global optimization methods face significant challenges Mazzia et al. [2023], Yao et al. [2023]. They generally show moderate performance on simpler datasets but underperform in more complex scenarios. Notably, these methods struggle to scale with an increasing number of edits in a single batch, often failing when edits exceed a certain threshold (e.g., 100 edits). Sequential editing, involving a series of batched edits, also poses a problem for these methods. Moreover, meta-learning or hypernetwork-based methods like KnowledgeEditor and MEND tend to be memory-intensive during training and inference, posing a disadvantage in certain applications.

Local Modification Approach

Local modification approach is another approach that involves parameter change. However, unlike global approaches, which update an untargeted portion of parameters of the LLM, local modification approaches leverage a "locate and modify" methodology, aiming to identify and modify only the parameters relevant to the specific target knowledge being edited. This targeted approach aims to minimize unnecessary changes in the network, thereby mitigating catastrophic forgetting.

- **Knowledge Neuron (KN)** identifies the neurons relevant to an edited statement by altering the weights of neurons in FFN layers from 0 to their original weights Dai et al. [2021]. This process assesses the cumulative change in the probability of the target statement, identifying neurons with higher saliency towards the edited statement. Updates to these salient neurons are conducted via a scaled vector based on the word embeddings of the original and the edited target. Notably, the authors observed that a significant portion of knowledge neurons reside in the top layers of the LLM, occupying up to 40% of the neurons in a single FFN layer.

- **Rank-One Model Editing (ROME)** takes a different approach by treating the update of an FFN layer as a whole Meng et al. [2022b]. Based on the theoretical premise that the second MLP layer in an FFN encodes the knowledge subject to editing, ROME modifies the weight matrix of this specific layer to enforce the desired key-value pair association. The key is constructed using the neuron activations of the text containing the subject over multiple passes, while the value is derived from the optimization process that minimizes predictive loss and the KL divergence to control output deviation from the original model.

- **Mass-Editing Memory in a Transformer (MEMIT)** extends ROME's capabilities to support batched editing of multiple FFN layers within the LLM Meng et al. [2022c]. This overcomes the inherent limitation of ROME, which restricts editing to single facts at a time. By incorporating multiple optimization objectives corresponding to multiple facts, MEMIT allows for efficient batched updates, leading to significantly improved performance compared to the sequential editing of ROME.

Local modification methods, particularly MEMIT, have demonstrated competitive performance in various comparative studies Yao et al. [2023], Wang et al. [2023b], Mazzia et al. [2023]. They offer valuable insights into the interpretability of LLMs by isolating the spe-

cific targeted factual association from the vast amount of knowledge and skills embedded within the model. However, the computation cost of each edit could be significant, and often requires hyperparameter search for different LLM architectures.

These three distinct approaches—external memorization, global optimization, and local modification—lay a robust foundation for advancing the field of knowledge editing in LLMs. Each contributes unique insights and strategies to the goal of making specific edits while striving to preserve the integrity of the model's extensive knowledge base. Among these, the external memorization approach, particularly exemplified by SERAC, demonstrates superior performance across various metrics Yao et al. [2023]. However, it's crucial to acknowledge that the field of knowledge editing is still evolving. Future research should not only address the current limitations of each approach but also focus on integrating these strategies to create more robust and versatile knowledge editing solutions. Additionally, the exploration of novel methodologies that leverage different properties of LLMs is essential.

2.2.3 Evaluation and Benchmarks

Evaluating and benchmarking knowledge editing in LLMs is essential for understanding their effectiveness and applicability. This section delves into existing metrics and benchmarks, highlighting their limitations and proposing avenues for future research.

Knowledge editing can be generally represented as changing an object from O to O^* within the context of a fixed subject S and a relation R. Here, the probability of the LLM predicting X in the context of Y is denoted as $P(X|Y)$. The decomposition and notation help in measuring the following aspects of knowledge editing:

- **Reliability**, sometimes referred to as accuracy or efficacy, measures the direct performance of knowledge updates, quantified by comparing probabilities of original and

edited targets after training prompts. Suppose that we have N training prompts, then

$$\text{Reliability} = \frac{1}{N}\sum_{i=1}^{N}\text{bool}(P(O_i^*|S_i, R_i) > P(O_i|S_i, R_i)) \tag{2.8}$$

- **Generality** evaluates the model's ability to consistently apply edited knowledge to semantically similar prompts (e.g., paraphrases of training prompts). Suppose that we have M semantically similar prompts, then generality is defined as:

$$\text{Generality} = \frac{1}{M}\sum_{i=1}^{M}\text{bool}(P(O_i^*|S_i, R_i, \text{context}_i) > P(O_i|S_i, R_i, \text{context}_i)) \tag{2.9}$$

- **Locality**, also known as specificity, measures the unintended impact on related, yet distinct, prompts, which are often called neighbor prompts. These neighbor prompts often share the same relation R but retain the original object O. Suppose that we have K neighbor prompts, locality is measured as:

$$\text{Locality} = \frac{1}{K}\sum_{i=1}^{K}\text{bool}(P(O_i|S_i', R_i) > P(O_i^*|S_i', R_i)) \tag{2.10}$$

While reliability, generality, and locality are established metrics, they have certain limitations. Recently, additional metrics like **Retainability** and **Scalability** have been introduced. Retainability, as discussed in Huang et al. [2023], Mazzia et al. [2023], evaluates the model's ability to preserve performance across multiple edits. Scalability, highlighted in Mitchell et al. [2022], Meng et al. [2022c], examines the capacity to manage large-scale edits. Additionally, evaluating computational resources required for each edit (in both training and inference times) is crucial for a comprehensive assessment of knowledge editing techniques.

The evaluation of knowledge editing in Large Language Models (LLMs) primarily utilizes QA benchmarks with clearly defined relations. A prominent example is the Zero-Shot

Relation Extraction (ZsRE) dataset, which is characterized by explicit factual statements encompassing relations R, subjects S, and objects O Levy et al. [2017]. ZsRE is particularly useful for evaluating generalization due to its inclusion of paraphrased prompts. However, two significant issues arise with ZsRE:

First, the reliance on WikiData as a data source renders the edits superficial and ineffective Vrandečić [2012]. Well-trained LLMs might already possess the relevant knowledge for ZsRE statements, rendering the "edited" responses more of a reinforcement than a correction. This reduces the effectiveness of ZsRE in measuring the true and profound impact of knowledge editing.

Second, the lack of objective comparisons hinders comprehensive evaluation. Since all statements are true, there is no inherent notion of "original" and "edited" objects. This limits the evaluation metrics to a non-comparative approach, failing to capture the nuances of different associations of $P(O_i^*|S_i, R_i)$ and $P(O_i|S_i, R_i)$.

To address these issues, the CounterFact benchmark was introduced. It modifies the subject and object associations in ParaRel (also sourced from WikiData) Elazar et al. [2021], Vrandečić [2012] to emphasize the editing aspect through factually incorrect statements. This ensures that LLMs are unlikely to have encountered these specific edits during training, providing a more meaningful test of their knowledge editing capabilities. Additionally, CounterFact enables comparative evaluation of associations, offering a more nuanced assessment than the single-target accuracy used in ZsRE.

More recently, the Multi-hop Question Answering for Knowledge Editing (MQuAKE) benchmark has emerged Zhong et al. [2023]. Like ZsRE and CounterFact, it sources data from WikiData but introduces the complexity of multi-hop QA. This benchmark challenges edited LLMs to integrate multiple pieces of knowledge to answer questions. Most knowledge editing methods struggle, revealing their inconsistencies in handling complex information flow.

Despite the valuable insights provided by these benchmarks, the evaluation of knowledge editing methods for LLMs remains incomplete. The reliance on a single data source (WikiData) raises concerns about potential overfitting and other generalization issues. Furthermore, the benchmarks predominantly utilize short, structurally simple sentences and fail to capture the complexities of real-world language use. Finally, the lack of post-edit evaluation neglects the broader impact of knowledge editing on the LLMs' overall knowledge and capabilities beyond factual QA tasks.

CHAPTER 3

OPEN PROBLEMS AND PROPOSAL

Building upon the background and related work discussed in the previous chapter, this chapter dives into several open problems and challenges in the realm of knowledge editing in LLMs. These challenges highlight the current limitations and open doors to new research opportunities. Some of these challenges are within the scope of this proposal, while others are reserved for future research.

3.1 Open Problems

Fine-tuning Methods Are Underexplored

Despite its frequent use as a baseline in knowledge editing research, fine-tuning is surprisingly underexplored. Existing works, such as De Cao et al. [2021], Mitchell et al. [2021], Meng et al. [2022c], often criticize fine-tuning due to its susceptibility to overfitting, stemming from the limited training data compared to the vast parameter space to update. Additionally, catastrophic forgetting, the degradation of unrelated knowledge and skills during training, further reduces its effectiveness.

To mitigate the issues, some continual learning methodologies have been adapted for knowledge editing. For example, external memorization knowledge editing methods resemble architecture-based methods in continual learning, aiming to prevent catastrophic forgetting by separating parameters for previous and current tasks. This comparative study Mazzia et al. [2023] also considers EWC as a knowledge editing method due to its ability to regularize weights and prevent forgetting during fine-tuning.

Yet, numerous other continual learning methods remain unexplored for knowledge editing tasks. Notably, replay-based continual learning, despite the availability of training datasets for LLMs, has not been extensively applied in this field. By mixing training material from

the original dataset with edited facts, we can provide more tokens for fine-tuning and partially mitigate catastrophic forgetting. However, selecting appropriate original training data is non-trivial, given the sheer size of NLP datasets. Additionally, the ratio of original data to edited facts requires careful consideration. Neighbor fact associations offer another avenue for replay-based methods. By incorporating them as training material, we can significantly mitigate forgetting of related knowledge. Generative replay, leveraging the generative capabilities of some LLMs, also presents an opportunity. By generating content related to the edited fact, this method can potentially help preserve relevant knowledge during fine-tuning while also providing training tokens of desirable distribution.

In summary, while naive fine-tuning with typically no more than ten tokens per edit has limitations, including insufficient tokens for training and significant catastrophic forgetting, ideas from continual learning, particularly replay-based approaches, offer promising solutions to these challenges.

Lack of Comprehensive Evaluation

The evaluation of the ramifications of knowledge editing in LLMs remains a critical but under-explored area. While existing benchmarks like CounterFact Meng et al. [2022a] provide a simple framework for knowledge editing and evaluation, they fall short in capturing the full impact of edits on related knowledge and general LLM capabilities.

To evaluate the locality of edits, CounterFact classifies knowledge associations with identical relations and original objects as neighbor facts. For example, if the fact "New York City is located in the United States" is edited to "New York City is located in the United Kingdom," then "Seattle is located in the United States" would be deemed a neighbor fact due to the shared relation "is located in" and the original object "the United States." While this approach is straightforward, it overlooks other knowledge dependencies and potential unintended consequences. For instance, facts about New York City, such as 'In New York

City, people speak English' or 'New York City has a 24-hour subway system,' should also be considered neighbor facts and remain unchanged after the edit. However, constructing such neighbor prompts is could be challenging.

Furthermore, existing benchmarks predominantly employ short QA or completion formats, which lack contextual depth. This format limits the robust evaluation of the generalization and locality. For instance, after the edit about New York City's location, existing benchmarks might pose questions like "Where is New York City located?" or provide prompts such as "New York City, located in." However, in scenarios with additional context, such as "Philadelphia is famous for its American-Italian cuisine, a legacy of the early Italian immigrants in the 1900s. New York City, located in", the question remains whether these knowledge editing methods still generalize effectively. Such contexts, potentially lengthier and more complex, mirror real application scenarios. The resilience of these knowledge editing methods to contextual noise is yet to be explored.

Moreover, the impact of these knowledge editing methods on other NLP capabilities, such as reasoning, summarization, or translation, remains untested. The closest evaluation in existing works is the assessment of **fluency** and semantic **consistency** in ROME Meng et al. [2022b], which reveals that some knowledge editing methods, e.g., KnowledgeEditor De Cao et al. [2021], experience repetitive nonsense word generation post-edit. However, these terms are based on output entropy or word distribution, which do not directly evaluate the LLM's other abilities.

In summary, current benchmarks are inadequate for assessing the broader impacts of knowledge editing, particularly on neighbor knowledge and overall LLM capabilities in complex scenarios.

A Thought Experiment on Counterfactual Editing

Current approaches to knowledge editing in LLMs treat fact associations as homogenous individual entities, disregarding their inherent diversity and connections. The current approach to knowledge editing overlooks the intricate connection between language and the world it represents. LLMs, built as world models through language, are susceptible to inconsistencies when this connection is simplified.

To demonstrate, we perform a thought experiment by asking the question: **what happens to an LLM when we edit a counterfactual association perfectly?** By achieving such a perfect edit, we essentially introduce a deliberate error into the model's representation of reality, triggering a cascade of potential inconsistencies within its internal knowledge base. Consider the example of editing the statement "New York City is located in the United States" to "New York City is located in the United Kingdom." This single edit leads to a series of logical issues. When a counterfactual association within an LLM is edited perfectly, the model should integrate this change as if the fact were true in our reality, triggering a cascade of changes to the LLM. Consider the edit "New York City is located in the United States" to "New York City is located in the United Kingdom." What happens to the people living in New York City? Do they magically change nationalities to become British citizens? Where does this leave iconic landmarks like Times Square and the Statue of Liberty? If the Statue of Liberty is still located in New York City, UK, the proposition that France gifted the Statue of Liberty to the UK after the independence of New York City from the UK exposes the absurdity that a simple edit can lead to.

These inconsistencies reveal the inherent limitations of manipulating factual associations through counterfactual edits. Such edits essentially construct an alternative reality within the LLM, a process fraught with imperfections, leading to either inconsistencies or superficial changes that lack true depth and coherence. This raises critical questions about the ideal approach to counterfactual knowledge editing in LLMs:

- **Localized Editing vs. Accepting Contradictions**: Should we strive for highly localized edits to minimize internal inconsistencies, even if it means sacrificing certain logical connections? Or should we accept certain contradictions as inevitable consequences of manipulating factual associations?

- **Editing Fundamental Knowledge**: This line of inquiry leads us to consider extreme scenarios. What if we edit the fundamental statement "one plus one equals three"? Should an ideal knowledge editing method alter all the rules of mathematical addition to accommodate this change? Should it simply accept the edited statement as an exception? Or should such edits be rejected altogether due to their potential to disrupt the underlying logical coherence of the LLM's knowledge representation?

This thought experiment underscores the challenges in using counterfactual knowledge associations as a benchmark for knowledge editing. However, the CounterFact dataset still holds significant value, as it presents scenarios unlikely to be encountered in the training process, thereby providing a more robust evaluation of knowledge editing capabilities. To enhance the effectiveness of this dataset, it's recommended to include closely related facts that may change during the edit, and to rigorously monitor the LLM's predictions on these prompts. This approach will help assess the impact of counterfactual edits on the LLM's overall performance, including its language processing and logical reasoning capabilities.

Challenges in Knowledge Representation

Knowledge representation plays a crucial role in enabling large language models (LLMs) to acquire, store, and manipulate information. A common approach involves representing factual statements through a triplet structure consisting of Relation R, Subject S, and Object O. While this approach provides a basic framework, it presents several challenges for knowledge editing.

One critical challenge is the reversibility issue seen from the one-directional knowledge association from R, S to the editted object O^*. For instance, in fine-tuning methods where the prompt constituted of R and S is trained to predict O^*, the edited model struggles to infer the subject S when prompted with R and O^*. For instance, if the fact "Tim Cook is the CEO of Apple" is edited, the reciprocal fact "The CEO of Apple is Tim Cook" requires a separate edit. This phenomenon highlights a significant limitation in the current knowledge editing methods and affects the model's ability to infer complete bidirectional associations Berglund et al. [2023].

Moreover, the simplistic R, S, and O format, derived from WikiData Vrandečić [2012], may not effectively represent complex or nuanced facts. This format's simplicity could restrict the scope of knowledge editing in LLMs, particularly for statements that do not conform to this structure. Exploring alternative knowledge representation formats could provide insights into more versatile and comprehensive editing methods.

3.2 Proposed Method

To address the open questions and challenges identified in this chapter, we propose two major contributions to the field of knowledge editing in LLMs: (1) an improved benchmark that can provides better training and evaluation prompts, and (2) a novel method for knowledge editing in LLMs using RAG and parameter-efficient fine-tuning methods. We will then outline our approach to method evaluation, focusing on generalization and locality of the knowledge editing method.

3.2.1 Improved Benchmark for Knowledge Editing in LLMs

The existing benchmarks, such as CounterFact Meng et al. [2022b], are insufficient in capturing the full impact of edits on related knowledge and general LLM capabilities, and lack in assessing the locality of edits and the model's resilience to contextual noise, as detailed

in Appendix 6.1. Our improved benchmark focuses on diversifying the patterns, neighbors, and contexts within the dataset.

Our proposed improvements are structured as follows:

- *Augmented Relation Patterns*: We will employ a mix of traditional NLP methods and advanced techniques using GPT-4 to enrich training materials and evaluation prompts. This not only increases the number of tokens for fine-tuning but also makes the evaluation of generalization and locality more rigorous with a greater variety of evaluation prompts. CounterFact's current structure, offering a minimal variety of sentence patterns for each edit, limits both the fine-tuning process and the evaluation. We aim to rectify this by introducing a wider array of relation patterns, significantly broadening the scope for model training and evaluation.

- *Additional Neighbor Associations*: By expanding the dataset to include a broader range of neighbor knowledge with the same subject, relation, or object, and querying WikiData Vrandečić [2012], we aim for a more comprehensive assessment of edits on related facts. This approach generates more evaluation prompts for locality and allows us to look into the impact of an edit on different types of neighbors with varying approximates to our subject. Currently, each entry in the CounterFact dataset only has 10 neighbors, without clear differentiation. We plan to enhance this by using the WikiData query service to retrieve a balanced mix of the most closely related neighbors and those more peripherally connected, based on the shared number of relations in the WikiData database. Additionally, we will differentiate these neighbors further by their frequency of mentions in WikiData, thus capturing a spectrum from frequently mentioned to rarely noted neighbors. This stratified approach allows a more detailed investigation of the aftermath of knowledge editing on various neighbor categories, which helps with the understanding of the edit's impact across a knowledge base.

- *Various Context in Paraphrase and Neighbor Prompts*: We plan to introduce vari-

ous contexts in prompts, allowing us to measure generalization and locality against contextual noise. This approach offers a more realistic evaluation resembling practical applications and will consider different linguistic structures and language variations. In contrast to CounterFact's limited contextual scope, our approach will involve crafting generic contexts applicable to any statement and designing noisy contexts that might challenge the model's paraphrasing and neighborhood statement completion. This will offer a more realistic and rigorous evaluation framework.

In summary, we will use the existing LLMs and other NLP methods to construct a more enriched version of CounterFact that includes a wider variety of relation patterns, a diverse array of neighbor associations for comprehensive impact assessment, and a set of generic and noisy contexts to rigorously validate the model's performance in realistic scenarios.

3.2.2 Novel Method for Knowledge Editing in LLMs

Current fine-tuning methods, often overlooked in existing works, are prone to overfitting and catastrophic forgetting due to the lack of training material and technique used to overcome the effect of catastrophic forgetting. Our approach involves applying a continual learning approach, specifically a replay-based approach, to knowledge editing in LLMs. We propose to tackle the problem of knowledge editing by experimenting with the following methods:

- *External Memorization with Context Differentiation:* Our approach utilizes embedding models with a vector store to meticulously differentiate between contexts related to edits and those that are not. When prompted, the LLM first checks the vector store to determine if the prompt relates to any of our edits. If so, the corresponding edited version of the LLM is activated, otherwise, the LLM's behavior remains unchanged. This selective activation ensures the LLM's behavior is unchanged unless a specific edited knowledge becomes relevant, effectively mitigating the issue of catastrophic forgetting in most cases.

- *Parameter-Efficient Fine-Tuning:* We plan to refine our fine-tuning process for each edit using parameter-efficient methods such as LoRA Hu et al. [2021], as these methods have been shown to limit the risk of overfitting by reducing the number of parameters that need updating. This strategy not only allows our system to scale in terms of computational resources and required training/inference time but also ensures that only a small, targeted part of the network is altered for each edit, preserving the overall knowledge and skills of the LLM.

- *A Replay-Based Method for Fine-Tuning:* We will enhance our knowledge editing by incorporating a replay method, which involves training on both the counterfactual statements and their neighbors or even original training material. This strategy aims to minimize forgetting and preserve the LLM's original knowledge and skills. To our knowledge, this approach is yet to be experimented with.

These proposed knowledge editing techniques aim to effectively solve the problem of catastrophic forgetting by integrating an external memorization approach and directly applying existing continual learning methods to knowledge editing. Ultimately, we aim to create a model that can be edited on demand without compromising the existing knowledge and skills of the model.

The process of knowledge editing is conceptualized as modifying a statement from (R, S, O) to (R, S, O^*), where S represents the subject, and O and O^* represent the object states before and after the edit, respectively. Our primary training objective aims to minimize the cross entropy loss associated with the updated object representation, O^*, in the context of multiple prompt patterns for each relation, denoted as $P_{R,1}, P_{R,2}, ..., P_{R,N}$, which are provided by our newly proposed benchmark. We utilize a tokenizer T to transform each prompt $P_{R,i}$ it into a format suitable for model training. The initial loss function, termed the naive edit loss, is defined as:

$$\mathcal{L}_{naive} = -\frac{1}{N}\sum_{i=1}^{N} \log \text{prob}\Big(T(O^*) \;\Big|\; T\big(P_{R,i}(S)\big)\Big) \tag{3.1}$$

To incorporate the original object into the loss, we add a penalty term that punishes the model for predicting the original object:

$$\mathcal{L}_{penalty} = \frac{1}{N}\sum_{i=1}^{N} \log \text{prob}\Big(T(O) \;\Big|\; T\big(P_{R,i}(S)\big)\Big) \tag{3.2}$$

Moreover, we propose to incorporate knowledge from neighboring subjects, defined as N_j for each j, to control the loss term. The neighbor subjects are the entities that shares the relation (R, N_j, O), which are often affected by the editing process. The loss function for these neighbor subjects is:

$$\mathcal{L}_{neighbors} = -\frac{1}{M \cdot N}\sum_{j=1}^{M}\sum_{i=1}^{N} \log\big(\text{prob}\big(T(O) \mid T(P_{R,i}(N_j))\big)\big) \tag{3.3}$$

So, in the end, our loss function will be in the form of the following equation:

$$\mathcal{L} = \mathcal{L}_{naive} + \lambda_1 \mathcal{L}_{penalty} + \lambda_2 \mathcal{L}_{neighbors} \tag{3.4}$$

Note that while $\mathcal{L}_{naive}$ is commonly used in existing works, the other two terms were not used and are yet to be evaluated. Moreover, we could potentially add in the original training material for the LLMs into the loss function to further mitigate the forgetting caused by fine-tuning. However, this could be beyond the scope of this thesis as it demands a huge number of tokens for fine-tuning, which is computationally very expensive.

These proposed research directions aim to address the key challenges in knowledge editing for LLMs, enhance evaluation benchmarks, and introduce innovative methods for fine-tuning and knowledge representation. This proposal seeks to advance the field of knowledge editing, paving the way for more accurate, ethical, and trustworthy applications of LLMs.

We acknowledge the inherent challenges associated with knowledge editing. The current limitations of knowledge representation formats and the difficulty in capturing the nuances of human language pose significant obstacles. Additionally, the issue of reversibility and potential inconsistencies when editing counterfactual associations require careful consideration in future knowledge editing methods. These challenges are beyond this proposal's scope and are reserved for further research.

Ultimately, while significant progress has been made in recent years, knowledge editing in LLMs remains an open research area with vast potential. By tackling the challenges outlined in this chapter and pushing the boundaries of existing methods, we can pave the way for a future where LLMs can be effortlessly edited and adapted to diverse scenarios, and able to fulfill diverse tasks with accuracy, leading to more ethical and trustworthy applications.

CHAPTER 4

PRELIMINARY RESULTS AND OBSERVATIONS

This chapter presents preliminary results and observations closely related to the challenges outlined in Chapter 3. We have made significant progress on multiple aspects of our proposed benchmark and method, substantiating the feasibility of our research goals. We categorize these results and observations into individual sections, analyzing each section's findings and their implications for the broader field of knowledge editing in LLMs.

4.0.1 CounterFact Relation Pattern Augmentation

Our augmentation process begins with the ParaRel dataset, which offers multiple rephrasings for each relation pattern found in CounterFact. We then employ GPT-4 to further diversify these patterns by generating as many paraphrases as possible. This step involves feeding ParaRel prompts to GPT-4 with other contextual information such as the restrictions on the subject and the object. Please see 6.2 for more information on the prompt. The temperature of GPT-4 was set to 0.2 for minor variations in phrasing while maintaining the core informational content.

We iteratively used the same prompt 50 times, collecting all generated patterns. These patterns underwent a meticulous manual selection process to discard any with altered meanings, strange word usage, unneutral tone, or unsuitable sentence structures—particularly those where the object were not at the end of the prompts. This selection process was critical for ensuring that only the most relevant and coherent patterns were retained as training or evaluation prompts. Additionally, we incorporated more diverse patterns manually, such as the ones in a question-answer format, to further enrich the dataset.

During the pattern selection phase, we identified certain relations as unsuitable for knowledge editing, and therefore removed them from the dataset. A relation was deemed unsuitable if it was too vague for effective paraphrasing. For example, the relation "A is located in B,"

without specific context, poses paraphrasing challenges. Conversely, specifying "A" as a city and "B" as a country enables more precise paraphrasing.

The concept of exclusivity also played a crucial role in evaluating relations. For example, the WikiData relation P108 (employer) illustrates a non-exclusive relationship, where editing "A's employer is B" to "A's employer is C" does not inherently negate the original statement, rendering the edit ambiguous. Instead, the relation should be structured for ensure exclusivity, such as indicating that "A" was only employed by "B" or "C" at specific times. Here, we removed any relations that might pose the problem of ambiguity.

The results of the augmentation process is shown in Table 4.1. In summary, **our augmentation process enriched the CounterFact dataset with an average of 82.7 new patterns per valid relation, effectively multiplying the original dataset's training and evaluation prompts by over 80 times**. This augmentation not only enables fine-tuning methods proposed in this work, but also sets a new benchmark for the depth and diversity of relation patterns in knowledge editing research. Please refer to the appendix on CounterFact dataset augmentation (6.2) for more details.

4.0.2 Neighbor Subject Query and Selection

To retrieve and categorize the neighbors for an in-depth evaluation of the ramifications of knowledge editing, we utilized the WikiData query service Vrandečić [2012] to query and select the neighbors. For each counterfactual statement, we queried the database, retrieving at most 2,000 neighbors of the subject to maintain efficiency. We categorized the neighbors based on predetermined criteria, choosing 4 distinct types with 10 examples each, as follows:

- close neighbors (share many properties with subject)
- distant neighbors (share few properties with subject)
- important neighbors (has many properties in WikiData)

Table 4.1: ParaRel/CounterFact Relation Pattern Augmentation Results

ID	Label	# of Patterns	Note
P17	country	99	
P19	place of birth	91	
P20	place of death	94	
P27	country of citizenship	122	
P30	continent	66	
P36	capital	84	
P37	official language	39	
P39	position held	117	
P101	field of work	138	
P103	native language	75	
P106	occupation	-	removed (similar to P39/P101)
P108	employer	-	removed (ambiguous)
P127	owned by	24	
P131	located in the administrative territorial entity	84	
P138	named after	107	subject may include object
P140	religion or worldview	72	
P159	headquarters location	138	
P176	manufacturer	61	subject may include object
P178	developer	66	
P190	twinned administrative body	-	removed (ambiguous)
P364	original language of film or TV show	29	
P407	language of work or name	-	removed (vague)
P413	position played on team / speciality	62	
P449	original broadcaster	138	
P463	member of	-	removed (vague)
P740	location of formation	30	
P937	work location	-	removed (vague)

- obscure neighbors (has few properties in WikiData)

We hypothesize that knowledge editing methods will particularly affect close and important neighbors due to their higher similarity and integration within the LLM's knowledge base, a hypothesis that will be rigorously tested in our proposed work. Similarly, we meticulously collect surrogate subjects, entities that share the edited object's new relation, to

comprehensively assess the knowledge editing's impact from multiple perspectives.

Besides querying neighbors, we also evaluated the constraints on the statement. We excluded samples that (1) contain outdated knowledge, (2) have an insufficient number of neighbors or surrogate subjects, or (3) are ambiguous in terms of reference, making it difficult to verify against the updated WikiData. Furthermore, we removed samples associated with relations excluded from the CounterFact dataset.

During dataset construction, we established a standalone WikiData query service, selecting the knowledge cutoff date of 2023/12/27 to align with the most current and relevant data while ensuring a manageable dataset size for effective analysis. Notably, many knowledge triplets are no longer valid according to the updated WikiData, due to updates in relations and/or entities. For instance, Danielle Darrieux's mother tongue was no longer French according to WikiData. Additionally, we restricted some relations; for example, for relation P27 (citizenship), we limited the object strictly to countries, thus excluding entities like colonies or empires.

In the end, **we extracted 11,192 out of 23,000 samples from the CounterFact dataset**. We applied stringent safety measures in our sample selection process to ensure the highest integrity and reliability of our research findings. As of now, the number of samples seems sufficient for knowledge editing research downstream, but we will further carefully investigate the elimination process and make changes if necessary.

To summarize the work done with regard to the dataset construction, the table 4.2 compares the CounterFact dataset and our improved dataset.

4.1 Context Differentiation using RAG

The proposed knowledge editing method relies heavily on a module that differentiates between relevant and irrelevant contexts for each specific edit. This module determines when to activate the fine-tuned parameters associated with a particular edit, thereby altering the

Table 4.2: Training & Evaluation Material of CounterFact and Our Improved Dataset

Material	**CounterFact**	**Our Improved Dataset**
Training	1 sentence	$(N \times 0.8)$ sentences [a]
Generalization	2 sentences	$(N \times 0.2)$ sentences without context $(N \times 0.2 \times M)$ sentences with generic context [b]
Locality	10 sentences	$(40 \times N)$ sentences without context [c] $(40 \times N \times M)$ sentences with context
Generalization+ [d]	Not available	$(40 \times N^2 \times 0.2)$ sentences
Locality+ [e]	Not available	$(40 \times N^2 \times 0.2)$ sentences

[a] N denotes the number of patterns per relation, which averages to 82.67 in our dataset. 80% of the patterns are used for training, and the remaining 20% are reserved for evaluation tasks.
[b] M denotes the total number of generic contexts in our dataset, currently standing at 25.
[c] For locality evaluation, each entity is associated with 40 neighbors, divided into 10 close, 10 distant, 10 important, and 10 obscure neighbors.
[d] Generalization+ evaluates the ability of the system to complete the counterfactual prompt within noisy or misleading context. Here, we use one of the neighborhood statements.
[e] Locality+ evaluates the ability of the system to complete the neighborhood prompt within noisy or misleading context. Here, we use one of the counterfactual statements.

LLM's behaviors and predictions. Its accuracy directly impacts the overall performance of the proposed knowledge editing system.

The context differentiation module comprises two components:

- **Embedding Model** converts sentences (edit knowledge) into embedding vectors.
- **Vector Store** stores the embeddings, allowing for similarity comparisons.

During an edit, each statement is converted into an embedding vector and saved in the vector store. When predicting the next token for a given prompt, the system retrieves the most similar embedding from the store by comparing it to all stored vectors. If the similarity exceeds a predefined threshold, the parameters fine-tuned with that specific edit are activated for prediction. Otherwise, the original LLM parameters are used.

We decided to use single-edit training instead of batch training for our knowledge editing system for several reasons based on the observations. First, single-edit training ensures that

edits are independent. This is important because it allows us to isolate different edits and ensure that our system can scale up to thousands of edits. Second, fine-tuning works less effectively for multiple edits. This is noted in MEMIT Meng et al. [2022c], where batched edits resulted in significantly lower generalization accuracy (64.8% over 10,000 samples) compared to single edits (96/6% in Meng et al. [2022b]). Batched edits in our system struggled to even reach 80% reliability, demonstrating the suboptimal nature of this approach. Finally, despite individual edit processing, our system still allows handling a large number of edits. The isolated nature of edited parameters minimizes the impact of sequential edits, as long as the context differentiation module can effectively scale up to handle the expanding knowledge base.

Initially, we explored using generative LLMs directly for context differentiation. This would have simplified the proposed architecture and eliminated additional computational overhead because we can use the same model for both context differentiation and generation. However, extracting attention from intermediate or last layers of models like GPT-2, GPT-J, or the embedding output of T5 encoders yielded unsatisfactory results. The context differentiation accuracy remained below 50% with merely 100 entries in the vector store, suggesting ineffective retrieval and subsequent parameter activation.

Switching to specialized embedding models, even those with significantly fewer parameters, dramatically improved performance. Models like *BAAI/bge-large-en* (355 million parameters) **achieved top-1 counterfact retrieval accuracy of 97.3% over 5,000 entries** Xiao et al. [2023]. This significant improvement was achieved by segmenting the input into overlapping pieces and utilizing *FAISS* for vector storage and retrieval Johnson et al. [2019]. The similarity is measured by Euclidean similarity. Please refer to Figure 4.1 for the comparative results between different embedding models. Furthermore, applying augmented patterns of counterfactuals instead of single training prompts holds promise for further enhancing performance.

In sum, our embedding-based context differentiation method achieves high performance with minimal overhead, making it a valuable and scalable solution. This approach efficiently handles large knowledge bases and seamlessly integrates with any LLM architecture, demonstrating the feasibility and practicality of our proposed knowledge editing system.

4.2 Fine-Tuning on Single Counterfactual Statement

Given the limited training corpus from a single counterfactual statement, directly modifying millions or billions of parameters in a large language model (LLM) can be impractical and inefficient. This section explores the potential of fine-tuning only specific portions of the LLM to update its knowledge based on counterfactual information.

Preliminary results, based on the ROME approach, suggest that training a single FFN of a transformer architecture can be sufficient for updating knowledge associations while maintaining good generalization. This approach significantly reduces the number of trainable parameters, enhancing scalability and enabling faster training compared to fine-tuning all layers.

Our experiments, using the AdamW optimizer with a learning rate of 0.0001 for 10 epochs on 100 samples, demonstrate that fine-tuning a single feedforward layer can achieve a generalization accuracy of 83.5%. This accuracy can be further improved with more training epochs, suggesting the potential of this method. Additionally, utilizing techniques like LoRA (Hu et al. [2021]) achieves even better generalization accuracy (93.0%) while minimizing the trainable parameter count to a mere 0.06% of the total. The results are shown in Table 4.3.

The relatively low locality observed for all fine-tuning experiments might raise concerns about the model's ability to restrict the editing to related prompts only. However, this is mitigated by the differentiation module introduced in the previous section. This module ensures that the fine-tuned parameters are only utilized when the prompt is directly relevant

to the specific counterfactual statement used for training. This effectively isolates the impact of fine-tuning, preventing it from affecting the model's responses to unrelated prompts. Therefore, despite the low locality without the context differentiation mdoule, the overall performance of our approach should remain robust.

Though promising, this approach comes with certain caveats. CounterFact, the evaluation metric used in this study, may not adequately capture the potential ramifications of fine-tuning on a single counterfactual statement. Training on a single counterfact for extended periods can disrupt the LLM's reasoning and other cognitive abilities. The method may lead to "word pairing," where the network associates subjects with new objects without considering the underlying relational patterns. Moreover, identifying the optimal layer for fine-tuning can be a computationally expensive process, as demonstrated by the ROME approach. These factors underscore the necessity for further exploration for fine-tuning methods.

Training Parameters	**Generalization (%)**	**Locality (%)**	**Trainable Param (%)**
None	16.5%	83.4%	-
All layers	88.0%	12.8%	31.1%
One layer	83.5%	27.8%	1.11%
One layer (LoRA)	93.0%	31.7%	0.06%

Table 4.3: Comparative results of different training parameters

4.3 Proposed Knowledge Editing Method

This section outlines a novel knowledge editing method that integrates the strengths of embedding and LoRA tuning, combining the capacity for neighborhood differentiation with high generalization accuracy and inherent scalability. The method consists of two key phases:

1. **During the Editing**: We construct a vector store by embedding various augmented versions of the edit statement for the context differentiation module. This vector store,

using *BAAI/bge-large-en* for embedding and *FAISS* for storage, is tagged with specific edit IDs Xiao et al. [2023], Johnson et al. [2019]. For each edit, we fine-tune our LLM with LoRA on a specified FFN layer, associating the LoRA parameters with the corresponding edit ID.

2. **During the Inference**: We employ a similarity threshold to determine if a given prompt relates to any of our edits. If the similarity between the prompt and any entry in the vector store surpasses this threshold, we select the highest similarity entry, retrieve its edit ID, and activate the fine-tuned LoRA parameters of that ID Hu et al. [2021]. Otherwise, the original LLM parameters are used for prediction.

The specifics of this approach may evolve based on further empirical evaluation. Our method, illustrated in Figure 4.2, merges the ability for neighborhood differentiation, as seen in embedding methods, with the high generalization accuracy typical in fine-tuning approaches. The use of LoRA, focusing on the feedforward part of a single layer, contributes to the scalability of our method. The additional parameters required for each edit are minimal, making this a viable approach for large-scale knowledge editing.

4.4 In-Context Learning for Knowledge Editing

In-context learning is emerging as a promising approach for knowledge editing in LLMs. This method involves training the model to make predictions directly based on the context provided in the input, offering an alternative baseline for our experiment.

We designed an idealized experiment to assess the capabilities of In-context learning for knowledge editing. The LLM utilized the context of the edited knowledge statement for each completion task, enabling us to evaluate its capacity for generalization and locality handling edits within a specific context. For instance, consider the counterfactual statement "The mother tongue of Danielle Darrieux is English". In our experiment, this context was added

to the beginning of the evaluation prompt for both generalization and locality assessments.

In a generalization evaluation, the prompt might be completed as "The mother tongue of Danielle Darrieux is English. ... Danielle Darrieux, a native". A higher generalization accuracy indicates a better learning ability for counterfactual contexts. For locality evaluation, the model might complete an unrelated prompt such as "The mother tongue of Danielle Darrieux is English. The native language of Montesquieu is", where the context acts as a distractor. A higher locality accuracy indicates that the LLM can complete the neighbor prompts despite the presence of misleading context.

This experimental setup, while not fully representative of real-world scenarios, offers valuable insights into the ideal capabilities of in-context learning.

	Generalization (%)	**Locality (%)**
without context	16.5%	83.4%
with context	87.0%	60.1%

Table 4.4: In-Context Learning Results with GPT-J on 100 CounterFact Samples

Table 4.4 demonstrates that while in-context learning enhances generalization, it does not achieve the high accuracy levels of advanced methods like SERAC and ROME (Mitchell et al. [2022], Meng et al. [2022b]), both of which report 99% accuracy in paraphrase completion. However, it seems that better prompt might solve this issue, as in Zheng et al. [2023]. The inclusion of the counterfactual context reduces locality accuracy from 83.4% to 60.1%, indicating that the LLM is sometimes misled by the context. Nonetheless, this accuracy still surpasses most fine-tuning-based approaches, suggesting that in-context learning possesses robustness perform knowledge editing without affect neighbor knowledge associations.

In summary, in-context learning offers a promising alternative for knowledge editing in LLMs. While further research is needed to address its limitations, in-context learning demonstrates significant potential for effective and robust knowledge manipulation within the context of LLMs.

4.5 Other Results and Observations

Further investigation into fine-tuning reveals intriguing variations in how different CounterFact samples respond to the training process. Notably, some counterfactual statements are trained more easily than others, with certain ones requiring only 2-3 epochs, while others demand over 10 epochs. Considering that all counterfactual statements in the dataset are essentially one-sentence structures with a similar number of tokens, the differing training difficulty levels are intriguing. This variation might be influenced by factors traceable to WikiData Vrandečić [2012], such as the types of relations, entities/objects involved, the number of training tokens, and the connection to other knowledge associations. Understanding what most significantly impacts the fine-tuning process could provide valuable insights into the design and optimization of counterfactual datasets.

Furthermore, it has been observed that some counterfactual statements impact neighborhood prompts less than others. This could be related to the relation pattern, the proximity between the counterfactual statements and its neighborhood facts, or the specific pragmatics of the counterfactual itself and its potential for world-building or altering perspectives. If there's a significant distance between a counterfactual statement and its neighbors, proposing better neighbors could improve the quality and effectiveness of the CounterFact dataset. This aspect warrants further research to better understand the dynamics of counterfactual training and its influence on related facts.

Incorporating statistical analysis or formulating hypotheses regarding these observations could provide more clarity and depth to our understanding of the fine-tuning process in large language models.

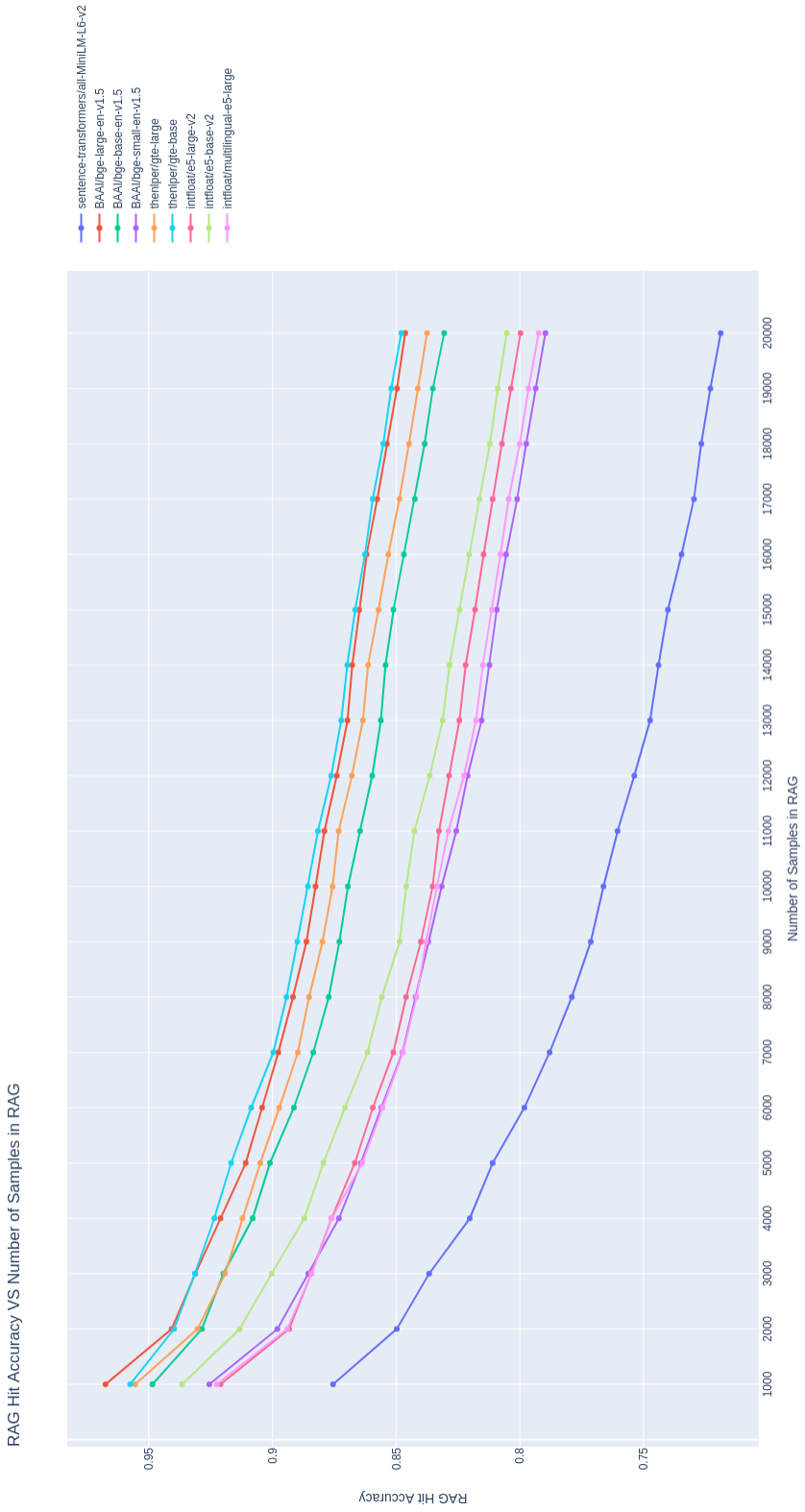

Figure 4.1: Top-1 retrieval accuracy for different top 8 embedding LLMs.

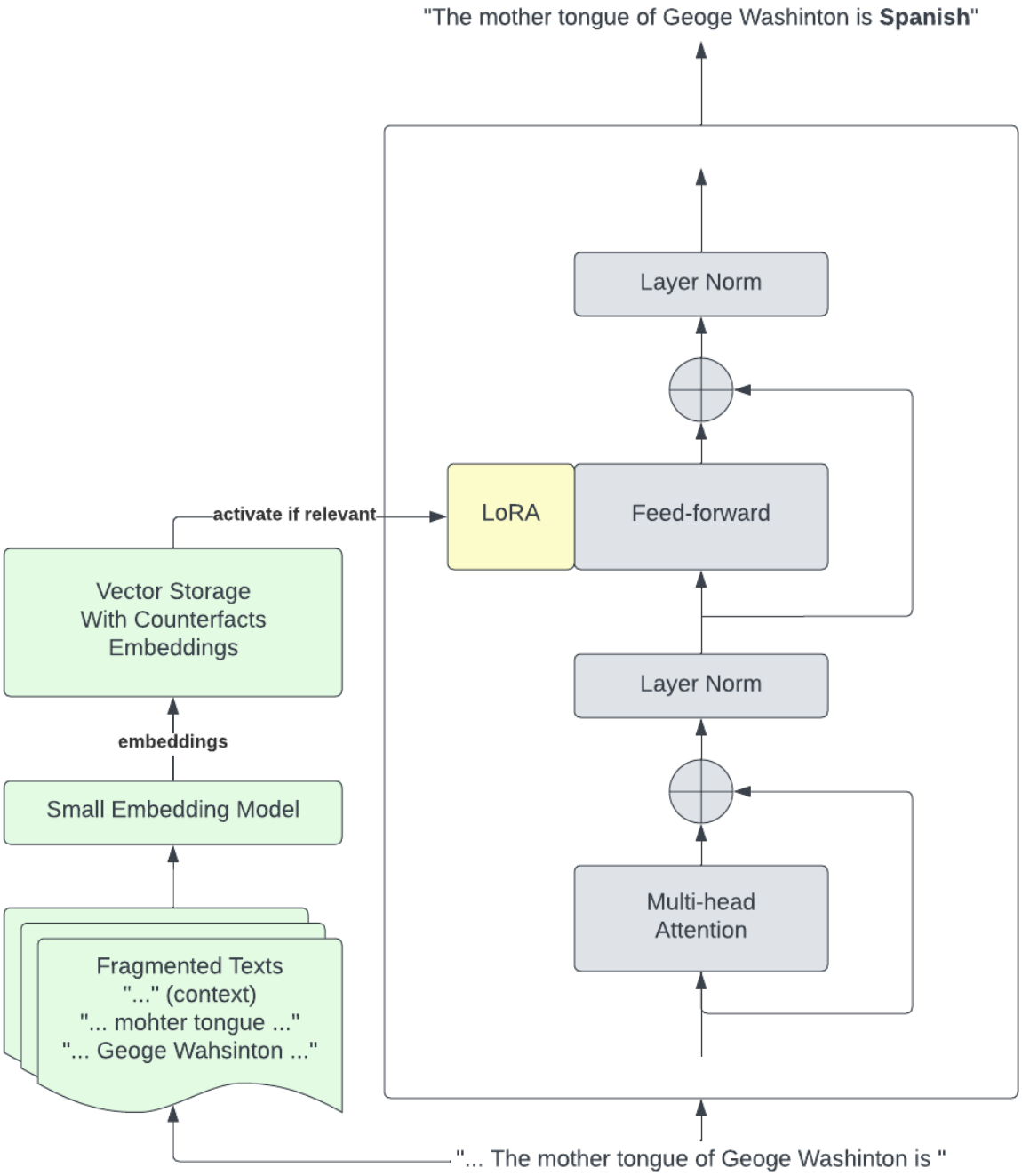

Figure 4.2: Design of the proposed knowledge editing method.

CHAPTER 5

SCHEDULE AND MILESTONES

In the proposed timeline in 5.1, each milestone has been carefully planned to align with the overarching objectives of the research. The initial phase, focusing on continual learning, includes tasks like literature review and various implementations of Hard Attention (HAT) in different scenarios. These foundational milestones, set from March 2022 to December 2023, lay the groundwork for understanding and addressing catastrophic forgetting in neural networks, an essential aspect of this research.

The second phase, dedicated to knowledge editing, begins in September 2023. It starts with augmenting the dataset using GPT-3.5 and nlpaug, crucial for creating a robust dataset for training and evaluation. The subsequent development of the neighborhood prompt generation using ParaRel in October 2023 allows for a more comprehensive evaluation for knowledge editing.

Progressively, the timeline includes the development of a vector store, in-context learning evaluation, and fine-tuning methods, culminating in the proposal of a new knowledge editing method by February 2024. This method aims to integrate the strengths of embedding and LoRA tuning, addressing scalability and generalization challenges.

Finally, the timeline extends to evaluations on the new CounterFact dataset and other benchmarks, ensuring the method's effectiveness and applicability across various scenarios. This comprehensive approach ensures a balanced progression from foundational understanding to practical application, making the timeline reasonable and attainable.

Table 5.1: Schedule and Milestones

Phase	**Task**	**ECD**	**Finished %**
Improved HAT for Continual Learning	Literature Review	Mar. 2022	90%
	Generalized HAT Implementation	Sep. 2022	100%
	HAT with ResNets on CIL/TIL	Oct. 2022	100%
	HAT with Transformers	May. 2023	100%
	Optimized Initialization & Scaling	May. 2023	100%
	⋆ Task-agnostic ViTs with HAT	Dec. 2023	50%
CIR Scenario in Computer Vision	HAT for CIR Scenarios	April. 2023	100%
	Test-Time Logits Averaging	May. 2023	100%
	Replica-Based Continual Learning for CIR	May. 2023	100%
	CVPR 4th Workshop Challenge Report	May. 2024	90%
Knowledge Editing for Counterfacts	Dataset Augmentation	Sep. 2023	100%
	Neighborhood Prompt Generation with ParaRel	Oct. 2023	90%
	Literature Review	Sep. 2023	90%
	Vector Store with Embedding	Sep. 2023	100%
	In-Context Learning Evaluation	Sep. 2023	80%
	Fine-Tuning Method Development	Oct. 2023	80%
	Replay-based Fine-Tuning Method	Jan. 2023	20%
	Proposed Method Development	Feb. 2024	50%
	Evaluation on New CounterFact Dataset	Feb. 2024	10%
	⋆ Evaluation on Other Benchmarks	Apr. 2024	10%
	⋆ In-Context Learning for Knowlege Editing	May. 2024	10%
	⋆ Counterfact Impact Analysis	May. 2024	0%

⋆ indicates an aspirational goal. ECD stands for (expected) completion date.

CHAPTER 6

APPENDIX

6.1 CounterFact Dataset Overview

CounterFact, introduced by Meng et al. [2022b], is a novel dataset specifically designed for evaluating methods for knowledge editing in LLMs. Sourced from Wikidata Vrandečić [2012], it offers a distinct advantage over existing benchmarks like ZsRE Levy et al. [2017] through its inherent **counterfactual** nature. The strength of CounterFact lies in its ability to clearly distinguish whether an LLM has successfully incorporated edited knowledge. By strategically swapping object entities within factual statements, the dataset generates objectively false statements that LLMs would not encounter during pretraining. If the LLM completes prompts with the false object, it serves as a strong indicator that the editing process has been successful.

The dataset comprises 21,919 samples, each containing a counterfactual statement, two paraphrase prompts, and ten neighborhood prompts. For example, the following counterfactual statement

```
The mother tongue of Danielle Darrieux is English.
```

is crafted by replacing the accurate object entity *French* with *English*. The dataset's evaluation methodology on generalization involves testing the edited LLM on 2 paraphrase prompts like:

- `"(an unrelated sentence). Danielle Darrieux, a native"`
- `"(an unrelated sentence). Danielle Darrieux spoke the language"`

Each paraphrase includes an unrelated sentence from WikiData and a trimmed and paraphrased version of the counterfactual statement for prompting. The dataset also employs 10

neighbor prompts, with the same relation (*native language* and the factual target (*French*), to evaluate locality:

```
- "The mother tongue of L\u00e9on Blum is",
- "The native language of Montesquieu is",
- "Fran\u00e7ois Bayrou, a native",
...
```

The drawbacks of CounterFact, mainly due to limited context and phrasing variety, are touched upon in chapter 3. Here, we provide a detailed critique specific to CounterFact, addressing these key issues more comprehensively:

- **Insufficiency of Training Material**: The core of the dataset, the counterfactual statements, may not provide enough tokens for effective fine-tuning of LLMs, which typically require a substantially larger corpus. Moreover, some training prompts provides no context with regard to the relation. For instance, "Toko Yasuda, the" is the training prompt for case #2, and the model is supposed to predict "piano" instead of "guitar" after the edit. The lack of tokens that reflects the relation between the subject "Toko Yasuda" and the object "piano" might only creates a strong connection between these two entities, but the model cannot fully understand the full context of the edit.

- **Limited Number of Paraphrase Prompts**: The limited number of paraphrase prompts per counterfactual statement hinders a comprehensive analysis of the LLM's generalizability. A more extensive set of paraphrase prompts with diverse phrasing and context would enable a more thorough evaluation of the trade-off between generalization and locality.

- **Lack of Different Neighbor Prompts**: The current neighbor prompts, sharing the same relation and object with the original factual statement, might not fully reveal

the consequences of knowledge editing. Adding a wider variety of neighbor prompts, including those with different relations involving the edited object, could provide a more comprehensive evaluation of locality and ensure that the LLM's understanding of related information remains intact.

- **No Contextual Diversity**: The absence of contextual noise in locality evaluation may not accurately reflect real-world applications.
- **Untested Capabilities**: The CounterFact dataset currently only evaluates the LLM's ability to recall and utilize edited factual knowledge. Other aspects, such as reasoning and mathematical operations, remain untested. Evaluating the LLM's performance on a wider range of NLP tasks would provide a more complete picture of its post-editing capabilities.

Overall, CounterFact provides a valuable tool for evaluating knowledge editing methods in LLMs. However, acknowledging its limitations and incorporating potential improvements will ensure a more comprehensive and accurate assessment of edited LLM capabilities.

6.2 CounterFact Dataset Augmentation

This section presents the augmentation process for a statement using an example from the CounterFact dataset Meng et al. [2022b]. Here is the counterfactual statement with case ID **28** from the CounterFact dataset:

```
Pidgeon Island belongs to the continent of Antarctica
```

Here, the subject is *Pidgeon Island*, the object is *Antarctica*, and the relation is *"belongs to the continent of"*. The WikiData relation ID is **P30**, which has 4 pattern variations in ParaRel that can be used to construct full sentences Elazar et al. [2021]:

```
{"pattern": "[X] is located in [Y].", ...}
{"pattern": "[X] is located in the continent [Y].", ...}
{"pattern": "[X] belongs to the continent of [Y].", ...}
{"pattern": "[X] is a part of the continent of [Y].", ...}
```

There are usually less than 10 prompts for each relation in ParaRel. These patterns are augmented by GPT-4 with the following prompt:

```
Please write a diverse set of linguistic patterns that accurately represent
↪  the relationship between elements [X] and [Y], as defined by a specific
↪  WikiData relation.

**Relation**
WikiData ID: \{\{relation_id\}\}
Description: \{\{relation_description\}\}

**Example Patterns**
\{\{patterns\}\}

**Guidelines for Pattern Writing**
1. **Preserve Original Meaning**: Each pattern should accurately reflect the
↪  relation between [X] and [Y], similar to the example patterns.
2. **Maintain Neutral Tone and Sentiment**: Keep the tone and sentiment of
↪  the example patterns, and make sure that the your patterns are not
↪  overly positive or negative. Additionally, refrain from using uncommon
↪  words, phrases or sentence structures.
3. **Explore Diverse Structures**: Use a variety of grammatical structures,
↪  such as normal sentence structures, appositive phrases (e.g., '[X], a
↪  ... [Y], ...'), and relative clauses (e.g., '[X], which/who/where ...
↪  [Y], ...'). Note: For phrases or clauses, a complete sentence is not
↪  required, but the pattern should accurately reflect the relation between
↪  [X] and [Y].
4. **Aim for Creativity and Variety**: Generate a wide range of patterns to
↪  showcase different ways of expressing the same idea.

**Response Format:**
Provide your patterns in the following JSON format:
```json
\{
  "patterns": [
    "pattern 1",
```
```

```
    "pattern 2",
    "pattern 3",
    ... // additional patterns
  ]
\}
```

Here, the relation ID is the same as the property ID from WikiData Vrandečić [2012]. The description is obtained from WikiData Query Service with the following query:

```
SELECT ?property ?propertyLabel ?propertyDescription WHERE {
  BIND(wd:{relation_id} AS ?property)
  SERVICE wikibase:label { bd:serviceParam wikibase:language
   ↪  "[AUTO_LANGUAGE],en". }
}
```

The augmentation process with GPT-4 was repeated 50 times, generating around 1,000 patterns per relation. Then we manually selected the patterns by establishing clear criteria to ensure they maintained semantic integrity, linguistic coherence, and thematic relevance to the original ParaRel patterns. This approach systematically removed patterns that significantly diverged in meaning, structure, or context, or were substantially duplicated, using both manual review and automated tools to prioritize pattern uniqueness and relevance.

Below are examples of the final augmented patterns, showcasing the diversity and quality achieved:

```
...
{
  "lemma": "be",
  "extended-lemma": "be-part-of",
  "texts": [
    "{subject} is part of {value}",
    "{subject} is part of the continent of {value}",
    "{subject}, which is part of {value}",
    "{subject}, which is part of the continent of {value}",
```

```
    "{subject}, a part of {value}",
    "{subject}, a part of the continent of {value}"
  ]
},
{
  "lemma": "be",
  "extended-lemma": "be-in",
  "texts": [
    "{subject} is in {value}",
    "{subject} is in the continent of {value}",
    "{subject}, which is in {value}",
    "{subject}, which is in the continent of {value}",
    "where is {subject}? It is in {value}",
    "where is {subject}? It is in the continent of {value}"
  ]
},
...
```

While not explicitly demonstrated in this example, conventional NLP augmentation methods like back translation can complement the proposed approach. However, such methods introduce few novel patterns and, therefore, were not used during our dataset construction.

REFERENCES

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3366–3375, 2017.

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

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019a.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019b.

Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8218–8227, 2021.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*, 2023.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

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

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

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018a.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018b.

Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*, 2021.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44 (7):3366–3385, 2021.

Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5138–5146, 2019.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. Calibrating factual knowledge in pretrained language models. *arXiv preprint arXiv:2210.03329*, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Xiaotian Duan. Hat-cl: A hard-attention-to-the-task pytorch library for continual learning, 2023.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=rkpACe1lx`.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with grace: Lifelong model editing with discrete key-value adaptors. *arXiv preprint arXiv:2211.11031*, 2022.

Hamed Hemati, Andrea Cossu, Antonio Carta, Julio Hurtado, Lorenzo Pellegrini, Davide Bacciu, Vincenzo Lomonaco, and Damian Borth. Class-incremental learning with repetition. In *Conference on Lifelong Learning Agents*, pages 437–455. PMLR, 2023.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jason Hoelscher-Obermaier, Julia Persson, Esben Kran, Ioannis Konstas, and Fazl Barez. Detecting edit failures in large language models: An improved specificity benchmark, 2023.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021.

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

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*, 2023.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Zixuan Ke and Bing Liu. Continual learning of natural language processing tasks: A survey. *arXiv preprint arXiv:2211.12701*, 2022.

Zixuan Ke, Hu Xu, and Bing Liu. Adapting bert for continual learning of a sequence of aspect sentiment classification tasks. *arXiv preprint arXiv:2112.03271*, 2021.

Zixuan Ke, Haowei Lin, Yijia Shao, Hu Xu, Lei Shu, and Bing Liu. Continual training of language models for few-shot learning. *arXiv preprint arXiv:2210.05549*, 2022.

Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*, 2017.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Gyuhak Kim, Sepideh Esmaeilpour, Changnan Xiao, and Bing Liu. Continual learning based on ood detection and task masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3856–3866, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*, 2017.

Dingcheng Li, Zheng Chen, Eunah Cho, Jie Hao, Xiaohu Liu, Fan Xing, Chenlei Guo, and Yang Liu. Overcoming catastrophic forgetting during domain adaptation of seq2seq language generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5441–5454, 2022a.

Guodun Li, Yuchen Zhai, Qianglong Chen, Xing Gao, Ji Zhang, and Yin Zhang. Continual few-shot intent detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 333–343, 2022b.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Qingbin Liu, Xiaoyan Yu, Shizhu He, Kang Liu, and Jun Zhao. Lifelong intent detection via multi-strategy rebalancing. *arXiv preprint arXiv:2108.04445*, 2021.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Edward Ma. Nlp augmentation. https://github.com/makcedward/nlpaug, 2019.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. Memory-assisted prompt editing to improve gpt-3 after deployment. *arXiv preprint arXiv:2201.06009*, 2022.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European conference on computer vision (ECCV)*, pages 67–82, 2018.

Vittorio Mazzia, Alessandro Pedrani, Andrea Caciolai, Kay Rottmann, and Davide Bernardi. A survey on knowledge editing of neural networks. *arXiv preprint arXiv:2310.19704*, 2023.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022a.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022b.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022c.

Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR, 2022.

Natawut Monaikul, Giuseppe Castellucci, Simone Filice, and Oleg Rokhlenko. Continual learning for named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13570–13577, 2021.

Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. In *Proceedings of the IEEE international conference on computer vision*, pages 1320–1328, 2017.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Jürgen Schmidhuber, Sepp Hochreiter, et al. Long short-term memory. *Neural Comput*, 9 (8):1735–1780, 1997.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. Continual-t0: Progressively instructing 50+ tasks to language models without forgetting. *arXiv preprint arXiv:2205.12393*, 2022.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pages 4548–4557. PMLR, 2018.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Gido M van de Ven and Andreas S Tolias. Three continual learning scenarios. In *NeurIPS Continual Learning Workshop*, volume 1, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Denny Vrandečić. Wikidata: A new platform for collaborative data collection. In *Proceedings of the 21st international conference on world wide web*, pages 1063–1064, 2012.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023a.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. Easyedit: An easy-to-use knowledge editing framework for large language models. *arXiv preprint arXiv:2308.07269*, 2023b.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, et al. Knowledge editing for large language models: A survey. *arXiv preprint arXiv:2310.16218*, 2023c.

Zhenyi Wang, Enneng Yang, Li Shen, and Heng Huang. A comprehensive survey of forgetting in deep learning beyond continual learning. *arXiv preprint arXiv:2307.09218*, 2023d.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022b.

Chenshen Wu, Luis Herranz, Xialei Liu, Joost Van De Weijer, Bogdan Raducanu, et al. Memory replay gans: Learning to generate new categories without forgetting. *Advances in Neural Information Processing Systems*, 31, 2018.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*, 2023.

Wenpeng Yin, Jia Li, and Caiming Xiong. Contintin: Continual learning from task instructions. *arXiv preprint arXiv:2203.08512*, 2022.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. Class-incremental learning via deep model consolidation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1131–1140, 2020.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021.

Yingxiu Zhao, Yinhe Zheng, Zhiliang Tian, Chang Gao, Bowen Yu, Haiyang Yu, Yongbin Li, Jian Sun, and Nevin L Zhang. Prompt conditioned vae: Enhancing generative replay for lifelong learning in task-oriented dialogue. *arXiv preprint arXiv:2210.07783*, 2022.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740*, 2023.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*, 2023.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.