SelfAug: Mitigating Catastrophic Forgetting in Retrieval-Augmented Generation via Distribution Self-Alignment

Anonymous ACL submission

Abstract

Recent advancements in large language models (LLMs) have revolutionized natural language processing through their remarkable capabilities in understanding and executing diverse tasks. While supervised fine-tuning, particularly in Retrieval-Augmented Generation (RAG) scenarios, has proven effective for enhancing task-specific performance, it often leads to catastrophic forgetting, where models lose their previously acquired knowledge and general capabilities. Existing solutions either require access to general instruction data or face limitations in preserving the model's original distribution. To overcome these limitations, we propose SelfAug, a novel self-distribution alignment method. By aligning distributions through the logits of input sequences, SelfAug preserves the model's semantic distribution, thereby simultaneously mitigating catastrophic forgetting and improving downstream task performance. Through extensive experiments, we show that SelfAug achieves a better balance between downstream task learning and the retention of general capabilities compared to existing methods. Our comprehensive empirical analysis reveals a direct correlation between distribution shifts and the severity of catastrophic forgetting in RAG scenarios, particularly highlighting how the absence of RAG capabilities in general instruction tuning leads to significant distribution shifts during fine-tuning. Our findings not only advance the understanding of catastrophic forgetting in RAG contexts but also provide a practical solution applicable across diverse fine-tuning scenarios. Our code is publicly available at https://anonymous.4open.science/r/SelfAug-5CB7.

1 Introduction

Large language models (LLMs) like GPT (Achiam et al., 2023), PaLM (Chowdhery et al., 2023), GLM (GLM et al., 2024), and LLaMA (Touvron et al., 2023) have revolutionized NLP by learning complex linguistic patterns from extensive pre-training data, demonstrating excellence in contextual understanding and few-shot learning capabilities.

Supervised fine-tuning (Ouyang et al., 2022; Chung et al., 2024) with general instruction datasets (Taori et al., 2023; Wang et al., 2022) improves models' instruction following abilities but often inadequately addresses specialized domain tasks. Task-specific fine-tuning provides targeted solutions for specialized applications (Roziere et al., 2023; Yang et al., 2024a; Hui et al., 2024; Luo et al., 2023a; Jin et al., 2024). Particularly, Retrieval-Augmented Generation (RAG) (Guu et al., 2020; Lewis et al., 2020; Gao et al., 2023; Cai et al., 2022; Chen et al., 2024b) enhances LLMs by incorporating external knowledge through retrieval, reducing hallucinations. Recent work (Yang et al., 2024c; Liu et al., 2024b; Zhang et al., 2024b) improves how models utilize relevant information and handle insufficient information.

However, fine-tuning for downstream tasks introduces catastrophic forgetting (French, 1999; Kemker et al., 2018; Shi et al., 2024; Wu et al., 2024; Luo et al., 2023b), where models lose previously acquired knowledge and instruction-following abilities when adapting to new tasks. This causes performance deterioration across diverse applications. For example, a model fine-tuned on document extraction may generate structurally incorrect code, despite improved document parsing abilities. Recent research attributes this problem to distribution shift when models adapt to specialized task distributions during fine-tuning (Saha et al., 2021; Yang et al., 2024d).

To address capability degradation, recent studies (Chen et al., 2024a; Bai et al., 2024; Jin and Ren; Huang et al., 2024) suggest incorporating general instruction data during downstream fine-tuning to maintain LLM's general capabilities. However, these strategies are limited by the scarcity of pub-

licly available instruction datasets. Researchers have therefore explored alternative approaches that retain the model's original distribution without accessing general data. Instruction synthesis methods like MAGPIE (Xu et al., 2024b) use the model to generate instruction-response pairs for data replay, though they depend heavily on generation quality. Parameter constraint methods such as Orthogonal Loss (Wang et al., 2023) enforce orthogonality between parameters but compromise downstream task performance. Knowledge reconstruction approaches like SDFT (Yang et al., 2024d) approximate the original distribution by regenerating responses from fine-tuning data but struggle with format-specific tasks, particularly when structured outputs like JSON are required. While each approach offers certain benefits, they all have limitations. These limitations underscore the need for more efficient solutions that better balance capability preservation and task adaptation.

To address aforementioned limitations, we propose SelfAug, a novel method that improves downstream performance while preserving the original capabilities of the model. SelfAug is general and adaptable for different fine-tuning scenarios. The core idea is to use the sequential processing of large language models, which produce probability distributions for both input and output sequences. These logits contain rich information about the model's learned knowledge and decision boundaries. By aligning the input sequence logits during fine-tuning, SelfAug maintains the model's original knowledge without needing the initial training data. The logits capture not only the final predictions but also the relationships among different outputs, reflecting the model's reasoning and uncertainty. This helps prevent catastrophic forgetting and keeps the fine-tuned model's behavior consistent with the original while learning new tasks (Hsu et al., 2022; Sun et al., 2024).

Our analysis shows catastrophic forgetting is especially severe in RAG scenarios, and we find that longer reference documents are linked to greater forgetting. Although modern LLMs perform well on tasks like mathematical reasoning and coding, they are not specifically trained for document use in RAG. Through systematic experiments, we find two main results. First, there is a strong link between distribution shift and catastrophic forgetting: larger shifts lead to greater loss of the model's original abilities. Second, using longer contexts during RAG training causes larger distribution shifts, which may increase changes in the model's behavior. Our SelfAug method reduces catastrophic forgetting and achieves downstream performance similar to LoRA, showing that aligning logits distributions is effective (Hsu et al., 2022; Sun et al., 2024). The main contributions of this work are as follows:

- We introduce SelfAug, a novel self-alignment method based on logits. SelfAug aligns input sequence logits to overcome limitations of current methods related to data access and parameter constraints. It does not need extra data or validation and avoids performance loss in downstream tasks caused by strict parameter updates.

- We provide an empirical analysis of catastrophic forgetting in RAG scenarios, showing that missing RAG ability in general instruction tuning causes significant distribution shift. We also find a direct link between the level of distribution shift and the severity of catastrophic forgetting.

- Our experiments on various benchmarks demonstrate that SelfAug achieves better downstream performance than existing methods while preserving the original model distribution and reducing catastrophic forgetting.

2 Related Works

2.1 Fine-Tuning

Fine-tuning leverages the knowledge of pre-trained large models to improve their performance on specific downstream tasks. This approach has proven effective in areas such as mathematics (Luo et al., 2023a; Yang et al., 2024a; Tang et al., 2024), code (Roziere et al., 2023; Hui et al., 2024), finance (Li et al., 2023; Wu et al., 2023a), and healthcare (Yu et al., 2024). Standard fine-tuning works by aligning the model's output distribution with the downstream data through log-likelihood maximization. Although open-source LLMs are available for fine-tuning, training all parameters remains computationally expensive. Parameter-Efficient Fine-Tuning (PEFT) (Mangrulkar et al., 2022; Han et al., 2024) addresses this by optimizing fewer parameters. Low-Rank Adaptation (LoRA) (Hu et al., 2021) is a popular PEFT method that allows fine-tuning with significantly fewer trainable parameters. Recent research (Wang et al., 2023; Liu et al., 2024a; Qiao and Mahdavi; Kowsher et al., 2024)

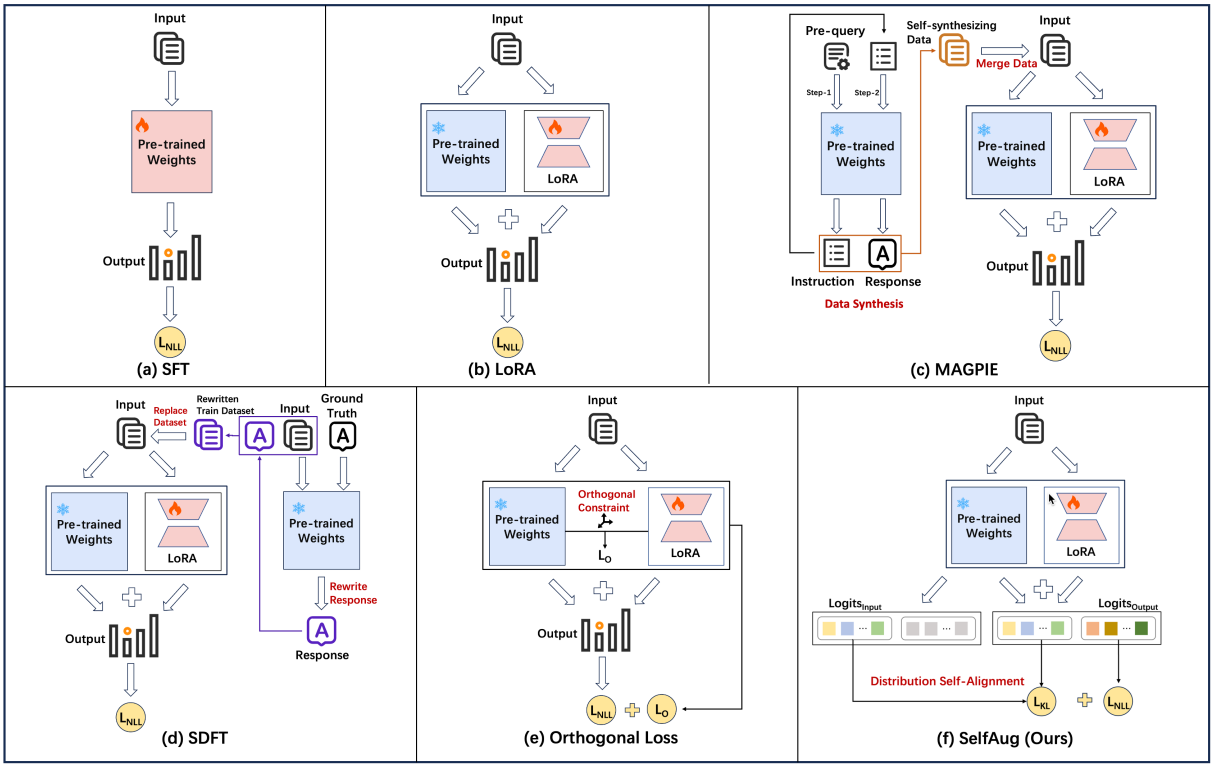

Figure 1: An illustration of full fine-tuning, LoRA, and methods for catastrophic forgetting mitigation. (a) SFT: Vanilla supervised fine-tuning with full parameter optimization. (b) LoRA: Parameter-efficient adaptation through low-rank decomposition. (c) MAGPIE: Self-synthesizing instruction-response pairs with pre-query templates for data replay. (d) SDFT: Fine-tuning with model-rewritten responses as optimized training dataset. (e) Orthogonal Loss: Imposing orthogonal constraints between LoRA modules and pre-trained parameters. (f) SelfAug: Self-distillation through input logits distribution alignment to preserve model's original capabilities.

has focused on improving LoRA to increase performance with minimal training costs and to support multiple downstream tasks.

2.2 Catastrophic Forgetting

Fine-tuning models causes catastrophic forgetting as the model shifts toward downstream task distributions and away from pre-training distributions. Traditional methods try to balance performance across different tasks through various approaches. Parameter-constraining methods use regularization (Ni et al., 2024; Xinrui et al.) or selective parameter updates (Lin et al., 2024; Alexandrov et al., 2024; Marczak et al., 2025; Jin and Ren, 2024a; Aggarwal et al., 2024; Franke et al., 2024; Panda et al., 2024; Zhang et al., 2024a; Yang et al., 2024b), but these limit downstream task performance. Mixture of Experts inspired approaches (Li et al., 2024a; Zhao et al., 2024; Le et al., 2024; Li et al., 2024b) maintain general capabilities by using different parameters for different tasks but alter model structure and prevent parameter merging. Data replay techniques (Bai et al., 2024; Jin and Ren, 2024b; Aggarwal et al., 2024; Huang et al., 2024) preserve foundation knowledge but are limited by pre-training data unavailability.

Among these, some methods focus on scenarios of continual learning, emphasizing the balance of performance across multiple downstream tasks. Our approach places more emphasis on mitigating the forgetting of general capabilities in pre-trained models, and addressing the limitations of the aforementioned methods, we propose a universal strategy to alleviate the catastrophic forgetting problem in LLMs during fine-tuning.

2.3 Knowledge Distillation

Knowledge distillation is widely used for model compression and performance improvement by transferring knowledge from a teacher model to a smaller student model. Early work (Hinton, 2015; Xie et al., 2018; Liu et al., 2019; Wang et al., 2020) focused on distilling knowledge from large models into smaller ones. Later studies applied knowledge distillation to various tasks (Shu et al., 2021; Zhang and Ma, 2020; Wang et al., 2019). For LLMs, the most common method (Mai et al., 2024; Xu et al., 2024a) uses KL divergence to reduce the difference

between the teacher and student output distributions. Other methods (Hou et al., 2020; Liang et al., 2023) align their intermediate hidden states. Some approaches (Wang et al., 2022; Ding et al., 2023) transfer knowledge from closed-source API models by augmenting the training data.

Most existing knowledge distillation methods focus on transferring output sequences distributions to improve downstream task performance of smaller models. In contrast, our method aims to reduce catastrophic forgetting during model fine-tuning by using the distribution of input sequences.

3 Method

In this section, we first outline the output logits of LLMs and the fine-tuning process. Subsequently, we introduce our SelfAug method and provide details on its implementation.

3.1 Logits as Model Distribution Representations

In LLM inference, input text undergoes several transformations to generate logits. Text is first tokenized into a sequence $x = [x_1, x_2, ..., x_n]$ and embedded into high-dimensional representations, then processed through multiple transformer layers to capture contextual relationships.

Finally, the model output is transformed into logits through a linear projection:

$$h_i = z_i^L W^T + b.$$

where $z_i^L \in \mathbb{R}^d$ represents the final layer hidden representation of the i-th token, $W^T \in \mathbb{R}^{d \times |V|}$ is the transpose of the projection matrix, and $b \in \mathbb{R}^{|V|}$ is the bias term. Each element in $h_i \in \mathbb{R}^{|V|}$ generates a corresponding score for each word in the vocabulary, reflecting the likelihood of selecting that word in the current context.

These logits are then converted to probability distributions via softmax for next-token prediction. The logit distribution encapsulates the linguistic patterns and semantic relationships learned during training (Jin and Ren, 2024a).

3.2 Fine-tuning: Aligning Model Distribution with Task Distribution

While powerful, LLMs still require optimization for specific tasks. Fine-tuning is a crucial step that adjusts the model distribution to match the task data distribution. We denote the model to be fine-tuned as M with parameters θ, mapping instruction x to output y.

Fine-tuning uses task-specific dataset $(x_t, y_t) \in D$ to update model parameters, aiming to minimize the negative log-likelihood loss:

$$\mathcal{L}_{NLL}(\theta) = -\sum_{(x_t, y_t) \in D} \log P(y_t \mid x_t; \theta).$$

By optimizing this function, the model's output distribution becomes closer to the true data distribution, with predicted outputs $\hat{y}_t$ more aligned with labels y_t. This process increases logits for target words and decreases them for others, making the model more suitable for specific task requirements.

3.3 SelfAug: Preserving Model Distribution via Input Logits

From a Bayesian perspective, model parameters θ exist within a probability distribution where pre-training establishes the prior distribution $p(\theta)$ that confers general abilities. During fine-tuning on a new dataset D, these parameters update to a posterior distribution $p(\theta \mid D)$ to adapt to the current task. However, when this update relies exclusively on the new dataset, the posterior may diverge substantially from the original prior, leading to catastrophic forgetting where the model loses its general knowledge and generalization ability. To mitigate this issue, we explicitly define the prior $p(\theta)$ as a distribution that remains close to the original model distribution, constraining it through the distributional distance between the fine-tuned model f_θ and the original model f_{θ_0}, as follows:

$$p(\theta) = exp(-\alpha \cdot Dist(f_\theta, f_{\theta_0}))$$

where $Dist(f_\theta, f_{\theta_0})$ denotes the distance between the distributions from the fine-tuned model and the original model, and α is a hyperparameter that controls the strength of this constraint. Therefore, the objective for optimizing the parameter posterior distribution during fine-tuning is as follows:

$$\theta^* = \underset{\theta}{argmax}\ p(\theta \mid D)$$

$$= \underset{\theta}{argmin} -log\, p(D \mid \theta) + \alpha \cdot Dist(f_\theta, f_{\theta_0})$$

$$= \underset{\theta}{argmin}\ \mathcal{L}_{NLL} + \alpha \cdot Dist(f_\theta, f_{\theta_0})$$

This design ensures that while the model parameters adapt to new data, their distribution does not deviate too far from that of the original model, which helps improve the model's adaptability to new tasks and effectively preserves the original knowledge and generalization ability.

Table 1: Results of Fine-tuning on Downstream Tasks in the RAG Domain (First CRAG, then RAG-Instruct). The CRAG benchmark employs a LLM-based ternary scoring mechanism (1: accurate, 0: missing, -1: incorrect) with overall performance represented by the mean score ranging from -1 to 1.

Dataset	Benchmark	Metric	Base	SFT	LoRA	+MAGPIE	+SDFT	+Orthgonal	+SelfAug
CRAG	CRAG	score (%)	-13.11	9.59	8.76	6.22 2.54↓	4.34 4.42↓	2.40 6.36↓	**10.94** 2.18↑
	ChatRAGBench	F1 (%)	24.04	25.92	31.90	33.56 1.66↑	31.22 0.68↓	33.77 1.87↑	**34.46** 2.56↑
	BioASQ	F1 (%)	66.76	59.41	59.70	62.06 2.36↑	64.71 5.01↑	62.35 2.65↑	**65.00** 5.30↑
	OmniEval	F1 (%)	66.05	42.58	51.64	54.71 3.07↑	48.87 2.77↓	49.53 2.11↓	**57.30** 5.66↑
	MATH	accuracy (%)	69.56	53.84	65.64	68.36 2.72↑	69.26 3.62↑	68.78 3.14↑	**69.46** 3.82↑
	HumanEval	pass@1 (%)	79.88	76.83	78.05	78.05 0.00↑	76.83 1.22↓	**79.88** 1.83↑	79.27 1.22↑
	IFEval	accuracy (%)	71.90	45.10	48.80	58.04 9.24↑	54.71 5.91↑	**63.77** 14.97↑	62.11 13.31↑
	MMLU	accuracy (%)	74.23	72.24	73.72	73.56 0.16↓	73.29 0.43↓	**74.45** 0.73↑	74.04 0.32↑
	ARC-C	accuracy (%)	86.78	85.08	88.47	88.47 0.00↑	89.83 1.36↑	89.15 0.68↑	**90.17** 1.70↑
	HellaSwag	accuracy (%)	85.48	83.72	84.55	83.68 0.87↓	82.54 2.01↓	**85.11** 0.56↑	83.73 0.82↓
	Average		71.57	63.73	67.22	68.89 1.67↑	68.02 0.80↑	69.36 2.14↑	**70.73** 3.51↑
RAG-Instruct	CRAG	score (%)	-13.11	-13.63	-7.19	-11.16 3.97↓	-17.00 9.81↓	-11.99 4.80↓	**-6.22** 0.97↑
	ChatRAGBench	F1 (%)	24.04	34.92	34.82	33.59 1.23↓	29.90 4.92↓	29.16 5.66↓	**35.44** 0.62↑
	BioASQ	F1 (%)	66.76	68.82	66.47	66.76 0.29↑	66.18 0.29↓	64.41 2.06↓	**70.00** 3.53↑
	OmniEval	F1 (%)	66.05	66.37	66.62	**67.68** 1.06↑	64.98 1.64↓	66.84 0.22↑	67.58 0.96↑
	MATH	accuracy (%)	69.56	69.64	69.88	68.12 1.76↓	69.82 0.06↓	**70.74** 0.86↑	70.02 0.14↑
	HumanEval	pass@1 (%)	79.88	46.34	76.83	**79.88** 3.05↑	76.22 0.61↓	79.27 2.44↑	79.27 2.44↑
	IFEval	accuracy (%)	71.90	55.64	63.77	64.32 0.55↑	66.73 2.96↑	**73.20** 9.43↑	68.02 4.25↑
	MMLU	accuracy (%)	74.23	73.61	73.36	72.96 0.40↓	73.28 0.08↓	**74.61** 1.25↑	73.66 0.30↑
	ARC-C	accuracy (%)	86.78	90.85	90.17	86.78 3.39↓	89.49 0.68↓	88.14 2.03↓	**92.20** 2.03↑
	HellaSwag	accuracy (%)	85.48	82.21	83.45	82.36 1.09↓	82.98 0.47↓	**85.82** 2.37↑	84.93 1.48↑
	Average		71.57	66.30	70.77	70.36 0.41↓	70.13 0.64↓	71.89 1.12↑	**72.51** 1.74↑

We propose the SelfAug, which aims to enhance performance on downstream tasks while maintaining the model's original distribution, as shown in Figure 1(g). We leverage the characteristic of LLMs in receiving sequential inputs, where the model produces logits for both input sequence x_t and the response sequence y_t, which together represent the original output distribution. Our key insight is using the original model's input sequence logits as a reference during fine-tuning. We measure the distribution difference between the original model be M_o and the fine-tuning model be M_{ft} using Kullback-Leibler divergence. For any input x_t, with logits $h_o(x_t)$ and $h_{ft}(x_t)$from respective models, we define the KL loss as:

$$Dist(f_\theta, f_{\theta_0}) = \mathcal{L}_{KL} = D_{KL}(p_{ft}(x_t) \,||\, p_o(x_t)).$$

where $p_o(x_t) = softmax(h_o(x_t))$ and $p_{ft}(x_t) = softmax(h_{ft}(x_t))$. The total loss function combines the negative log-likelihood loss $\mathcal{L}_{NLL}$ for the response sequences and the KL divergence loss:

$$\mathcal{L}_{total} = \mathcal{L}_{NLL} + \alpha \mathcal{L}_{KL}.$$

where α is a hyperparameter that balances the importance of the two loss terms.

SelfAug aligns the distribution of the original model through the logits of input sequences during the fine-tuning process. For each training pair (x_t, y_t), the model not only learns the data distribution of downstream tasks through the response sequence y_t, but also maintains the distribution of the original model through the logits of the input sequence x_t. This integration of dual distributions effectively alleviates the catastrophic forgetting problem. Compared to methods requiring replay of original data or generation of responses, SelfAug offers the advantage of not needing additional data or complex response validation steps, thereby simplifying the implementation process and reducing computational overhead.

4 Experiment

To evaluate the effectiveness of SelfAug and its impact across different scenarios, we aim to answer the following research questions:

- **RQ1**: How does SelfAug perform compared with the state-of-the-art methods?
- **RQ2**: How does constrained distributional shift mitigate catastrophic forgetting?

- **RQ3**: How do different components influence SelfAug?
- **RQ4**: How does SelfAug perform across varying context lengths and model configurations?

4.1 Experimental Setup

Baselines. We use Qwen2.5-7B-Instruct as our base model and compare our method with four representative approaches, as shown in Figure 1:

- **Vanilla Fine-Tuning**: full-parameter fine-tuning and LoRA.
- **MAGPIE** (Xu et al., 2024b): Employs model-generated instruction-response pairs for data replay during fine-tuning.
- **SDFT** (Yang et al., 2024d): Fine-tunes using data generated from the model's own distribution to maintain alignment.
- **Orthogonal Loss** (Wang et al., 2023): Constrains LoRA parameters to be orthogonal to the original model parameters.

Datasets. We fine-tune models on the CRAG (Yang et al., 2024c) and RAG-Instruct (Liu et al., 2024b) datasets. Our evaluation framework encompasses four categories of datasets designed to comprehensively assess model capabilities across various domains:

- **RAG Ability Evaluation**: CRAG and ChatRAG-Bench (Liu et al., 2024c)
- **Domain-specific RAG Ability Evaluation**: BioASQ (Nentidis et al., 2024) and OmniEval (Wang et al., 2024b).
- **Foundational Ability Evaluation**: MATH (Hendrycks et al., 2021), HumanEval (Chen et al., 2021), and IFEval (Zhou et al., 2023).
- **General Knowledge Evaluation**: MMLU (Hendrycks et al., 2020), ARC-C (Clark et al., 2018), and HellaSwag (Zellers et al., 2019).

A comprehensive description of baselines, datasets, evaluation methodologies, and implementation details is provided in Appendix A.

4.2 Overall Performance Evaluation (RQ1)

We first evaluated the effectiveness of our proposed SelfAug method, which can maintain the performance of LLMs on downstream task learning while mitigating catastrophic forgetting during the fine-tuning process. Specifically, we conducted fine-tuning on the RAG dataset to assess the impact on the model's performance in both RAG tasks and other general capability tasks. Additionally, we observed that fine-tuning downstream tasks significantly affected the model's instruction-following abilities, whereas the impact on the model's knowledge was relatively mild. The evaluation results are presented in Table 1.

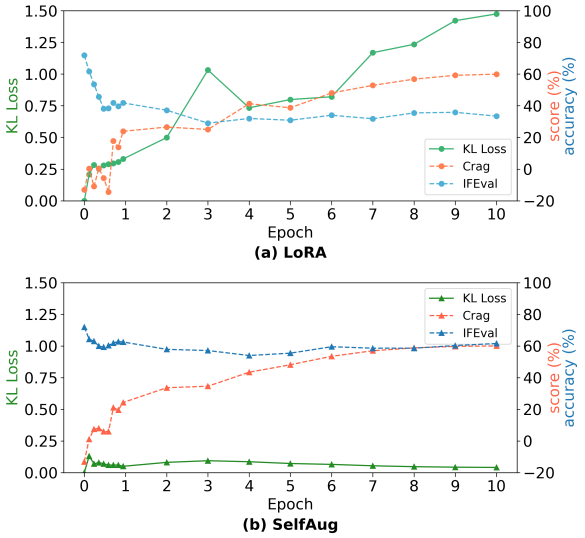

Figure 2: Epoch-wise Performance and Logits Divergence. KL Loss measures the distribution shift of model output logits, IFEval evaluates instruction-following ability catastrophic forgetting, and CRAG represents downstream task performance. LoRA exhibits increasing shift and forgetting, while SelfAug maintains stable performance through effective distribution constraints.

4.2.1 SelfAug Effectively Mitigated Catastrophic Forgetting.

Our experimental results demonstrate that while fine-tuning enhances downstream task performance, it simultaneously induces distribution shifts that compromise other capabilities. Following LoRA fine-tuning on the CRAG dataset, IFEval accuracy declined to 48.80, indicating substantial catastrophic forgetting. Although MAGPIE and SDFT effectively mitigated catastrophic forgetting, SelfAug exhibited superior capability in this regard. Orthogonal Loss, while achieving robust catastrophic forgetting mitigation through strict orthogonal constraints, significantly compromised downstream task performance. In contrast, SelfAug demonstrated comparable forgetting mitigation while achieving exceptional results in downstream task learning, outperforming LoRA on targeted tasks. Among all methodologies evaluated, SelfAug established the optimal equilibrium between downstream task learning and catastrophic forgetting mitigation, thereby attaining the highest average performance across evaluation metrics.

4.2.2 The Impact on the Model's Knowledge is Slight.

Table 1 illustrates the results of the foundation knowledge assessment after fine-tuning with downstream tasks. While fine-tuning substantially deteriorates the model's instruction-following ability, its foundation knowledge retention remains remarkably robust. The performance across various foundation knowledge benchmarks exhibits minimal degradation after fine-tuning, with certain methodologies even demonstrating enhanced performance. These findings suggest that catastrophic forgetting in LLMs predominantly manifests through the degradation of instruction-following abilities rather than the erosion of foundation knowledge. This observation is also supported by other studies (Zhang and Wu, 2024; Yang et al., 2024d).

4.3 Distribution Shift and Catastrophic Forgetting (RQ2)

In this section, we explore how RAG task performance, instruction-following abilities, and distribution shift evolve over the course of training. After incorporating SelfAug, by imposing constraints on the distribution shift, we can alleviate catastrophic forgetting while maintaining RAG task performance.

4.3.1 Distribution Shift Induced Catastrophic Forgetting.

We trained the LLM for 10 epochs and visualized its performance across the CRAG training set, IFEval datasets, as well as changes in KL Loss. As shown in Figure 2(a), increasing the number of training epochs progressively improves both the performance of model on Crag and logits distribution shift. At the same time, instruction-following ability suffers from a severe decline. This phenomenon reveals a strong correlation between the magnitude of distribution shift and the severity of catastrophic forgetting. The results demonstrate that continued training leads to increases in both RAG performance and logits distribution divergence, while degrading general capabilities.

4.3.2 Effectiveness of SelfAug in Mitigating Distribution Shift.

Based on these observations, SelfAug leverages logits distribution self-alignment to constrain distribution shift during model training, effectively mitigating catastrophic forgetting. As demonstrated in Figure 2(b), after applying the SelfAug constraint, the KL divergence of model logits significantly decreases and maintains at a stable level. Furthermore, the degradation of instruction-following ability is notably suppressed, confirming the effectiveness of our method in mitigating catastrophic forgetting phenomena. Notably, while mitigating catastrophic forgetting, SelfAug does not compromise the model's performance on training data, demonstrating a well-balanced trade-off between maintaining downstream task learning capabilities and preventing catastrophic forgetting.

Table 2: Performance Comparison of Constraints Using Different Layer Outputs.

Method	IFEval	Method	IFEval
LoRA	48.80	LoRA	48.80
+ Attention Q	47.13	+ Attention All	50.46
+ Attention K	50.09	+ FFN	51.02
+ Attention V	48.24	+ All layers	49.35
+ Attention O	47.50	+ SelfAug (Ours)	**62.11**

4.4 Ablation Study (RQ3)

Since distribution shift can occur on features at any module within the model, the effectiveness of SelfAug might be influenced by two factors: the location where constraints are applied and the strength of the constraints. Therefore, in the ablation study, we will focus primarily on these two aspects.

4.4.1 The Impact of Loss Position.

Previous research has examined knowledge distillation via intermediate features, but in our systematic study comparing self-distillation across different transformer block components, we find through extensive experiments that distilling at the final logits layer consistently yields better performance than using intermediate representations, as presented in Table 2. This phenomenon can be explained through information bottleneck theory. As data propagates through the network architecture, information undergoes progressive filtration, emphasizing task-relevant features. The logits layer primarily contain essential semantic content. Distilling at this final layer not only aligns the model closely with task-relevant information but also improves generalization and robustness, while intermediate layers may introduce unnecessary complexity due to their mix of relevant and irrelevant features.

4.4.2 The Impact of Loss Weight.

By adjusting the weight parameter α in SelfAug, we can control the strength of distribution constraints, where higher weights impose stronger

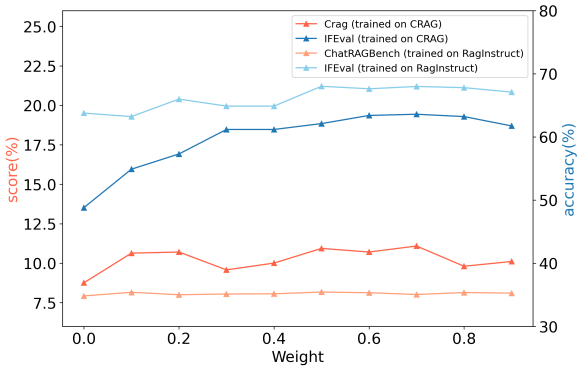

Figure 3: Model Performance with Respect to Weight Scaling. Larger loss weights strengthen distribution shift constraints, effectively mitigating forgetting.

constrain on the model's output distribution. As illustrated in Figure 3, increasing the weight parameter leads to a gradual recovery of the model's instruction-following ability. The experimental results show that SelfAug effectively reduces the divergence between the model's current and original distributions, thereby mitigating catastrophic forgetting. This demonstrates that our proposed approach successfully addresses the root cause of forgetting by maintaining the model's output distribution closer to its initial state while adapting to RAG tasks.

4.5 Generalizability of SelfAug (RQ4)

In a RAG scenario, the LLM needs to utilize retrieved documents of varying lengths to answer questions. Therefore, we conducted experiments on model size, LoRA rank, and context length. Additionally, to further validate the effectiveness of our method, we also tested it on tasks with low distribution shift.

4.5.1 Generalizability of SelfAug Across different Context Lengths.

As context length increases, the model's performance on general instruction-following tasks declines due to distribution shift. To investigate this, we analyzed how training with longer contexts affects catastrophic forgetting. We gradually expanded context length by adding more documents and measured instruction-following ability at each length, as shown in Table 3. When context length increased from 2K to 8K tokens, instruction-following accuracy dropped from 58.23 to 50.28. Applying SelfAug improved performance, showing its effectiveness in reducing catastrophic forgetting at all context lengths.

Table 3: Results of Instruction-Following Ability at Different Context Lengths.

Avg Tokens Num	LoRA	SelfAug
2K tokens	58.23	**63.03** 4.80↑
4K tokens	56.19	**62.48** 6.29↑
6K tokens	52.87	**55.82** 2.95↑
8K tokens	50.28	**57.67** 7.39↑

4.5.2 Generalizability of SelfAug Across different Model Configurations.

We evaluated SelfAug on different model sizes and settings. On the CRAG benchmark, we observed that larger base models struggled more with hallucination, but after fine-tuning, SelfAug consistently outperformed LoRA at all scales and better preserved general abilities. For LoRA rank, increasing trainable parameters caused greater loss in instruction-following, but SelfAug reduced this effect across all parameter settings. Downstream task performance improved within an optimal parameter range but dropped if the parameter count was too high due to redundancy (Wang et al., 2024a). We also applied SelfAug to mathematical reasoning and code generation using the MATH and Magi-Coder datasets (Wei et al., 2023). Since these tasks have low distribution shift, SelfAug brought only minor improvements but successfully maintained instruction-following ability. These results show SelfAug is versatile and effective in various domains. More details are in Appendix B.

5 Conclusion

Our research explores the problem of catastrophic forgetting when fine-tuning language models for retrieval-augmented generation tasks. We find that distribution shift during fine-tuning weakens the model's general performance, especially its ability to follow instructions. To address this, we propose SelfAug, a method that does not use data replay or change the model architecture, and can be applied to any fine-tuning setting. SelfAug uses only the original training data and aligns the model's input distributions by constraining input sequence logits. This simple approach reduces distribution shift and helps prevent catastrophic forgetting. Our experiments show that there is a clear link between distribution shift and catastrophic forgetting. SelfAug reduces this shift and preserves model abilities, while matching or exceeding the downstream task performance of standard fine-tuning methods.

Limitations

While our proposed SelfAug serves as a plug-and-play approach that can be seamlessly integrated into both LoRA and full-parameter fine-tuning paradigms, comprehensive experiments on full-parameter fine-tuning scenarios were not conducted due to computational resource constraints. Future work could explore the effectiveness and scalability of SelfAug in full-parameter fine-tuning settings, potentially revealing additional insights into its broader applicability across different training paradigms.

References

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

Divyanshu Aggarwal, Sankarshan Damle, Navin Goyal, Satya Lokam, and Sunayana Sitaram. 2024. Exploring continual fine-tuning for enhancing language ability in large language model. *arXiv preprint arXiv:2410.16006*.

Anton Alexandrov, Veselin Raychev, Mark Niklas Müller, Ce Zhang, Martin Vechev, and Kristina Toutanova. 2024. Mitigating catastrophic forgetting in language transfer via model merging. *arXiv preprint arXiv:2407.08699*.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2020. Open-domain question answering goes conversational via question rewriting. *arXiv preprint arXiv:2010.04898*.

Andrew Bai, Chih-Kuan Yeh, Cho-Jui Hsieh, and Ankur Taly. 2024. Which pretrain samples to rehearse when finetuning pretrained models? *arXiv preprint arXiv:2402.08096*.

Deng Cai, Yan Wang, Lemao Liu, and Shuming Shi. 2022. Recent advances in retrieval-augmented text generation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 3417–3419.

Howard Chen, Jiayi Geng, Adithya Bhaskar, Dan Friedman, and Danqi Chen. 2024a. Continual memorization of factoids in large language models. *arXiv preprint arXiv:2411.07175*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024b. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Jörg K.H. Franke, Michael Hefenbrock, and Frank Hutter. 2024. Preserving principal subspaces to reduce catastrophic forgetting in fine-tuning. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Geoffrey Hinton. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems*, 33:9782–9793.

Yen-Chang Hsu, James Smith, Yilin Shen, Zsolt Kira, and Hongxia Jin. 2022. A closer look at knowledge distillation with features, logits, and gradients. *arXiv preprint arXiv:2203.10163*.

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

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. *arXiv preprint arXiv:2403.01244*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Qiao Jin, Yifan Yang, Qingyu Chen, and Zhiyong Lu. 2024. Genegpt: Augmenting large language models with domain tools for improved access to biomedical information. *Bioinformatics*, 40(2):btae075.

Xisen Jin and Xiang Ren. Demystifying language model forgetting with low-rank example associations. In *NeurIPS 2024 Workshop on Scalable Continual Learning for Lifelong Foundation Models*.

Xisen Jin and Xiang Ren. 2024a. What will my model forget? forecasting forgotten examples in language model refinement. In *Forty-first International Conference on Machine Learning*.

Xisen Jin and Xiang Ren. 2024b. What will my model forget? forecasting forgotten examples in language model refinement. *arXiv preprint arXiv:2402.01865*.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. 2018. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Md Kowsher, Nusrat Jahan Prottasha, and Prakash Bhat. 2024. Propulsion: Steering llm with tiny fine-tuning. *arXiv preprint arXiv:2409.10927*.

Minh Le, An Nguyen, Huy Nguyen, Trang Nguyen, Trang Pham, Linh Van Ngo, and Nhat Ho. 2024. Mixture of experts meets prompt-based continual learning. *arXiv preprint arXiv:2405.14124*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhiyuan Cheng, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. 2024a. Mixlora: Enhancing large language models fine-tuning with lora based mixture of experts. *arXiv preprint arXiv:2404.15159*.

Tianhao Li, Shangjie Li, Binbin Xie, Deyi Xiong, and Baosong Yang. 2024b. Moe-ct: a novel approach for large language models training with resistance to catastrophic forgetting. *arXiv preprint arXiv:2407.00875*.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023. Large language models in finance: A survey. In *Proceedings of the fourth ACM international conference on AI in finance*, pages 374–382.

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, pages 20852–20867. PMLR.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, and 1 others. 2024. Mitigating the alignment tax of rlhf. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 580–606.

Chengyuan Liu, Yangyang Kang, Shihang Wang, Lizhi Qing, Fubang Zhao, Changlong Sun, Kun Kuang, and Fei Wu. 2024a. More than catastrophic forgetting: Integrating general capabilities for domain-specific llms. *arXiv preprint arXiv:2405.17830*.

Wanlong Liu, Junying Chen, Ke Ji, Li Zhou, Wenyu Chen, and Benyou Wang. 2024b. Rag-instruct: Boosting llms with diverse retrieval-augmented instructions. *arXiv preprint arXiv:2501.00353*.

Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. 2019. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2604–2613.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2024c. Chatqa: Surpassing gpt-4 on conversational qa and rag. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023a. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

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

Zheda Mai, Arpita Chowdhury, Ping Zhang, Cheng-Hao Tu, Hong-You Chen, Vardaan Pahuja, Tanya Berger-Wolf, Song Gao, Charles Stewart, Yu Su, and 1 others. 2024. Fine-tuning is fine, if calibrated. *arXiv preprint arXiv:2409.16223*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and B Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. *URL: https://github. com/huggingface/peft*.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. 2025. Magmax: Leveraging model merging for seamless continual learning. In *European Conference on Computer Vision*, pages 379–395. Springer.

Anastasios Nentidis, Georgios Katsimpras, Anastasia Krithara, Salvador Lima-López, Eulàlia Farré-Maduell, Martin Krallinger, Natalia Loukachevitch, Vera Davydova, Elena Tutubalina, and Georgios Paliouras. 2024. Overview of bioasq 2024: the twelfth bioasq challenge on large-scale biomedical semantic indexing and question answering. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 3–27. Springer.

Yao Ni, Shan Zhang, and Piotr Koniusz. 2024. Pace: marrying generalization in parameter-efficient fine-tuning with consistency regularization. *arXiv preprint arXiv:2409.17137*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ashwinee Panda, Berivan Isik, Xiangyu Qi, Sanmi Koyejo, Tsachy Weissman, and Prateek Mittal. 2024. Lottery ticket adaptation: Mitigating destructive interference in llms. *arXiv preprint arXiv:2406.16797*.

Fuli Qiao and Mehrdad Mahdavi. Learn more, but bother less: parameter efficient continual learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, and 1 others. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Gobinda Saha, Isha Garg, and Kaushik Roy. 2021. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. 2024. Continual learning of large language models: A comprehensive survey. *arXiv preprint arXiv:2404.16789*.

Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen. 2021. Channel-wise knowledge distillation for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5311–5320.

Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. 2024. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15731–15740.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024. Mathscale: Scaling instruction tuning for mathematical reasoning. *arXiv preprint arXiv:2403.02884*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Sheng Wang, Liheng Chen, Jiyue Jiang, Boyang Xue, Lingpeng Kong, and Chuan Wu. 2024a. Lora meets dropout under a unified framework. *arXiv preprint arXiv:2403.00812*.

Shuting Wang, Jiejun Tan, Zhicheng Dou, and Ji-Rong Wen. 2024b. Omnieval: An omnidirectional and automatic rag evaluation benchmark in financial domain. *arXiv preprint arXiv:2412.13018*.

Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. 2019. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4933–4942.

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Yukang Wang, Wei Zhou, Tao Jiang, Xiang Bai, and Yongchao Xu. 2020. Intra-class feature variation distillation for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 346–362. Springer.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2023. Magicoder: Source code is all you need. *arXiv preprint arXiv:2312.02120*.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023a. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

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

Zeqiu Wu, Ryu Parish, Hao Cheng, Sewon Min, Prithviraj Ammanabrolu, Mari Ostendorf, and Hannaneh Hajishirzi. 2023b. Inscit: Information-seeking conversations with mixed-initiative interactions. *Transactions of the Association for Computational Linguistics*, 11:453–468.

Jiafeng Xie, Bing Shuai, Jian-Fang Hu, Jingyang Lin, and Wei-Shi Zheng. 2018. Improving fast segmentation with teacher-student learning. *arXiv preprint arXiv:1810.08476*.

Wang Xinrui, Chuanxing Geng, Wenhai Wan, Shao-Yuan Li, and Songcan Chen. Forgetting, ignorance or myopia: Revisiting key challenges in online continual learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Shilin Xu, Xiangtai Li, Haobo Yuan, Lu Qi, Yunhai Tong, and Ming-Hsuan Yang. 2024a. Llavadi: What matters for multimodal large language models distillation. *arXiv preprint arXiv:2407.19409*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024b. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024a. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Shuo Yang, Kun-Peng Ning, Yu-Yang Liu, Jia-Yu Yao, Yong-Hong Tian, Yi-Bing Song, and Li Yuan. 2024b. Is parameter collision hindering continual learning in llms? *arXiv preprint arXiv:2410.10179*.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, and 1 others. 2024c. Crag–comprehensive rag benchmark. *arXiv preprint arXiv:2406.04744*.

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Leon Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. Corda: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. 2024d. Self-distillation bridges distribution gap in language model fine-tuning. *arXiv preprint arXiv:2402.13669*.

Haoran Yu, Chang Yu, Zihan Wang, Dongxian Zou, and Hao Qin. 2024. Enhancing healthcare through large language models: A study on medical question answering. In *2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pages 895–900. IEEE.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Hengyuan Zhang, Yanru Wu, Dawei Li, Sak Yang, Rui Zhao, Yong Jiang, and Fei Tan. 2024a. Balancing speciality and versatility: a coarse to fine framework for supervised fine-tuning large language model. *arXiv preprint arXiv:2404.10306*.

Liang Zhang, Katherine Jijo, Spurthi Setty, Eden Chung, Fatima Javid, Natan Vidra, and Tommy Clifford. 2024b. Enhancing large language model performance to answer questions and extract information more accurately. *arXiv preprint arXiv:2402.01722*.

Linfeng Zhang and Kaisheng Ma. 2020. Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In *International Conference on Learning Representations*.

Xiao Zhang and Ji Wu. 2024. Dissecting learning and forgetting in language model finetuning. In *The Twelfth International Conference on Learning Representations*.

Lulu Zhao, Weihao Zeng, Xiaofeng Shi, and Hua Zhou. 2024. Mosld: An extremely parameter-efficient mixture-of-shared loras for multi-task learning. *arXiv preprint arXiv:2412.08946*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Experimental Setup

A.1 Baselines.

In our empirical investigation, we conduct extensive experiments using Qwen2.5-7B-Instruct (Team, 2024) as our base model for fine-tuning. To systematically evaluate the effectiveness of our proposed method, we compare it with representative approaches from four major categories: instruction synthesis methods, knowledge reconstruction approaches, model modifications, and parameter constraint methods. We consider the following five baseline methods as our comparative benchmarks, as shown in Figure 1(a)-(e):

- **Vanilla Fine-Tuning**: We provide experimental results for both full-parameter fine-tuning and Low-Rank Adaptation (LoRA) (Hu et al., 2021) fine-tuning for comparison.

- **MAGPIE** (Xu et al., 2024b): In this approach, the LLM autonomously generates instructions when provided with pre-query templates as input, and subsequently produces corresponding responses for these instructions. The synthesized instruction-response pairs are utilized as alternative training samples for general instruction fine-tuning during data replay.

- **SDFT** (Yang et al., 2024d): This method bridges the distribution gap by fine-tuning with a dataset generated from the model's distribution. The guiding model regenerates responses and validates their correctness to ensure alignment with the original data distribution.

- **Orthogonal Loss**: Inspired by the concept of O-LoRA (Wang et al., 2023), this approach constrains the parameters of the LoRA modules to be orthogonal to the original model parameters, with the goal of minimizing the impact of fine-tuning on the model's distribution.

A.2 Datasets.

Our experimental evaluation consists of three main components: RAG capability evaluation, downstream task evaluation, and foundation knowledge evaluation. Each component assesses the performance of our approach across distinct domains.

RAG Ability Evaluation. We focus on enhancing core RAG capabilities: document-based information retrieval and question answering, robustness against irrelevant or noisy documents, and the ability to abstain from answering given erroneous queries or insufficient context. For validation, we fine-tune our models on two datasets: **CRAG** (Yang et al., 2024c) and **RAG-Instruct** (Liu et al., 2024b), and evaluate on two benchmarks: **CRAG** and **ChatRAGBench** (Liu et al., 2024c). The CRAG dataset contains 2.7k question-answer pairs with retrieved reference documents, structured into validation and public test sets. The evaluation protocol in CRAG implements a ternary scoring mechanism, where responses are evaluated by GPT-4o to assign scores of 1, -1, and 0 to accurate, incorrect, and missing answers, respectively. The overall score is calculated as the mean score across all responses, with a range of [-1, 1]. RAG-Instruct provides a publicly available 40K instruction dataset covering various RAG scenarios. For evaluating multi-turn conversational QA with extensive document contexts, we employ QuAC (Choi et al., 2018), QReCC (Anantha et al., 2020), and INSCIT (Wu et al., 2023b) following the experimental settings in ChatRAGBench.

Domain-specific RAG Evaluation. We evaluate RAG capabilities in the biomedical and financial domains using **BioASQ** (Nentidis et al., 2024) and **OmniEval** (Wang et al., 2024b), respectively. BioASQ is a series of international competitions designed to advance large-scale biomedical semantic indexing and question answering. For evaluation, we use Task b from BioASQ 2024 and employ ideal answers as ground truth. OmniEval serves as a RAG benchmark encompassing 5 task categories and 16 financial topics. We rely on GPT-4o for correctness assessment.

Foundational Ability Evaluation. For mathematical reasoning, we utilize the **MATH** (Hendrycks et al., 2021), which comprises 12,500 competition-level mathematics problems. For code generation Ability, we employ the **HumanEval** (Chen et al., 2021) to evaluate the model's pro-

gramming proficiency. We evaluate the model's instruction-following ability using **IFEval** (Zhou et al., 2023), which assesses the model's capability to follow various types of instructions.

General Knowledge Evaluation. To evaluate the preservation of foundation knowledge, we employ three established benchmarks: **MMLU** (Hendrycks et al., 2020), **ARC** (Clark et al., 2018), and **HellaSwag** (Zellers et al., 2019).

The evaluations on the MATH, HumanEval, MMLU, ARC, and HellaSwag datasets are conducted using the standardized OpenCompass (Contributors, 2023) evaluation framework to ensure consistency and reproducibility.

A.3 Implementation Details.

For the CRAG dataset, we strictly adhere to the official configuration, utilizing the validation set for fine-tuning and the public test set for evaluation under Task 1 settings. The model is trained for 1 epoch with a batch size of 16 and a learning rate of 5e-4. Regarding the RAG-Instruct dataset, we configure the training with a batch size of 512 and a learning rate of 5e-5 over 3 epochs. To mitigate potential model collapse during full parameter fine-tuning at high learning rates, we adopt reduced learning rates of 1e-5 and 5e-6 for CRAG and RAG-Instruct, respectively. Throughout the training process, we employ the AdamW optimizer with a cosine learning rate schedule, setting the weight decay to 0.1 and the warmup ratio to 5%. In the implementation of MAGPIE, we maintain a mixing ratio of 1:9 between MAGPIE-generated data and original training samples. Unless otherwise specified, we set the KL divergence loss weight in SelfAug to 0.5 in experiments, as our ablation studies confirm that 0.5 is a reasonable value. To ensure fair comparisons across tasks and metrics, score normalization is applied when computing the overall average performance. We conducted five repeated experiments to obtain the best value and determined the above hyperparameters through a hyperparameter grid search. The experiment was conducted using 4 A100 GPUs.

B Ablation Studies on Model Configurations

B.1 Generalizability of SelfAug Across different Model Scales.

Our investigation into the scalability of SelfAug across different model sizes reveals intriguing patterns, as illustrated in Table 4 through evaluation results on the CRAG benchmark. Contrary to conventional expectations, our experiments demonstrate that the relationship between model size and CRAG performance is not monotonically positive for base models. This counter-intuitive phenomenon can be attributed primarily to the prevalence of hallucination cases in the CRAG dataset, where questions are either inadequately contextualized or fundamentally unanswerable. Particularly noteworthy is our observation that larger base models exhibit diminished performance when encountering such hallucination scenarios, resulting in degraded overall performance metrics.

However, upon fine-tuning with both LoRA and our proposed SelfAug method, we observe a significant paradigm shift in model behavior. The fine-tuned models demonstrate markedly improved capabilities in handling hallucination cases, with performance scaling consistently with model size. Most significantly, our SelfAug approach exhibits superior effectiveness in preserving general capabilities compared to conventional LoRA, effectively mitigating catastrophic forgetting across all model scales. These findings not only validate the scalability of our approach but also underscore its robust performance advantages over existing methods, particularly in addressing the challenging aspects of hallucination management in LLMs.

Table 4: Model Performance with Different Model Sizes

	CRAG			IFEval		
Size	**Base**	**+LoRA**	**+SelfAug**	**Base**	**+LoRA**	**+SelfAug**
3B	-46.82	6.37	**7.19** 0.82↑	61.37	49.54	**57.86** 8.32↑
7B	-13.11	8.76	**11.24** 2.48↑	71.90	48.80	**62.11** 13.31↑
14B	-26.29	14.31	**15.81** 1.50↑	79.67	45.84	**67.47** 21.63↑
32B	-40.90	17.98	**19.10** 1.12↑	77.45	60.81	**75.60** 14.79↑
72B	-20.30	19.92	**19.93** 0.01↑	83.73	52.87	**62.85** 9.98↑

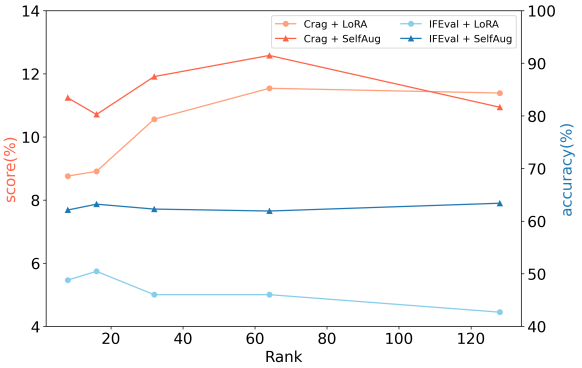

Figure 4: Model Performance with Respect to LoRA Rank. Increasing trainable parameters through LoRA rank amplifies catastrophic forgetting severity.

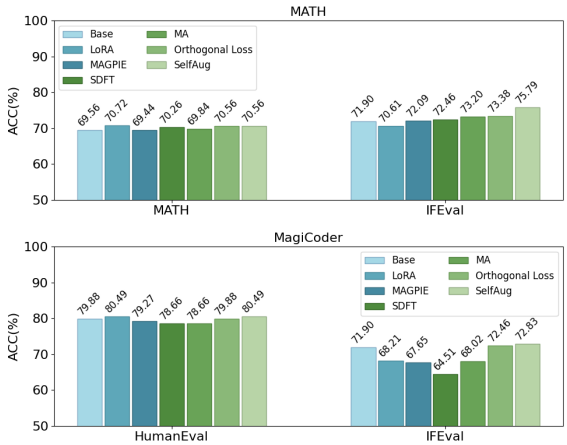

Figure 5: Evaluation Results of Math and Code Tasks. SelfAug exhibits robust forgetting mitigation effectiveness.

B.2 Generalizability of SelfAug Across different Lora Ranks.

Having established the correlation between distribution shift and catastrophic forgetting, we investigate the impact of trainable parameters on forgetting severity. Table 1 shows that SFT exhibits more severe forgetting than LoRA, suggesting larger trainable parameter sets lead to greater distribution shift. Through controlled experiments with varying LoRA ranks, Figure 4 reveals that increasing trainable parameters consistently deteriorates instruction-following ability, while our SelfAug method effectively mitigates this across parameter scales. Notably, downstream task performance improves with parameters within an optimal range but degrades beyond a threshold due to redundancy (Wang et al., 2024a).

B.3 Generalizability of SelfAug On Tasks with Low Distribution Shift.

To thoroughly assess our approach, we applied Self Aug to mathematical reasoning and code generation tasks, fine-tuning on the MATH and MagiCoder (Wei et al., 2023) datasets. As shown in Figure 5, given the model's extensive pre-training and strong baseline in these areas, additional fine-tuning minimally improved performance, with gains mostly under 1 percentage point. While the conventional LoRA approach showed some decline in instruction-following, SelfAug prevented this and slightly enhanced overall capabilities. This demonstrates SelfAug's effectiveness in maintaining model stability and expanding its benefits across various application domains, even in low distribution shift scenarios.