Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 4, No. 7, 2025

Task-Aware Regularized Continual Fine-Tuning for Mitigating Catastrophic Forgetting in Large Language Models

Jiahui Hu[1], Yiyun Lyu[1], Jiejun Jiang[1*]
1Northeastern University, Seattle, USA
*Corresponding author: Jiejun Jiang; jiejun.j.jiang@gmail.com

Abstract: This paper addresses the problem of catastrophic forgetting and generalization degradation in large language models during multi-task continual fine-tuning. To solve this problem, we propose a Task-aware Regularized Continual Fine-tuning (TRCF) method. The method consists of two core mechanisms: Task-aware Parameter Selection (TaPS) and Multi-stage Regularization Alignment (MRA). TaPS analyzes the semantic relevance and parameter sensitivity between tasks to adaptively select a subset of parameters for updating. This helps suppress interference and conflict between old and new tasks and improves the model's ability to retain prior knowledge. MRA constructs a regularization evolution path across task stages to guide the model in maintaining semantic consistency in the representation space during continual fine-tuning. This enhances the stability and effectiveness of multi-stage knowledge integration. Empirical studies conducted across multiple task stages show that the proposed method outperforms mainstream continual learning approaches in terms of average retention, new task accuracy, and overall performance. Further experiments on hyperparameters and data sensitivity confirm that the framework demonstrates strong robustness and adaptability under different learning rates, regularization strengths, data scales, and input structures. The proposed method provides an efficient and scalable solution for continual task input, dynamic structure learning, and parameter evolution control. It offers both theoretical support and methodological guidance for achieving task generalization and memory stability in large-scale pretrained language models.

Keywords: Continuous fine-tuning; parameter selection; regularized alignment; large language model

1. Introduction

With the rapid advancement of artificial intelligence, Large Language Models (LLMs) have become a central technology in natural language processing [1], [2]. Equipped with powerful pretraining capabilities and general language understanding, LLMs have shown exceptional performance in tasks such as question answering, dialogue generation, summarization, and text classification. However, in real-world applications, pretrained models often require fine-tuning to adapt to the semantic features and distributional differences of specific downstream tasks. A major challenge is how to adapt to diverse tasks while preserving generalization ability. In multi-task or dynamically changing task scenarios, traditional fine-tuning methods tend to cause forgetting of previously learned tasks, which leads to a decline in overall performance and generalization [3].

Continual learning, as an important paradigm inspired by human long-term learning, aims to address the problem of catastrophic forgetting in multi-stage learning. Introducing continual learning mechanisms into LLM fine-tuning allows rapid adaptation to new tasks while preserving knowledge of old ones. However, due to the massive number of parameters, complex semantic space, and vague task boundaries in LLMs, conventional continual learning methods are often ineffective or difficult to apply directly. Furthermore, significant differences between tasks in terms of semantic representation, objective functions, and language structure can cause task interference. If models are trained without distinguishing such differences, performance degradation or even collapse may occur. Therefore, detecting task changes and dynamically adjusting the model's update strategy is key to improving continual fine-tuning of LLMs [4].

In this context, task-aware strategies have attracted increasing attention. The core idea of task awareness is to capture structural differences, semantic associations, and parameter sensitivity between tasks to enable more fine-grained and targeted knowledge updates. Introducing task awareness into fine-tuning helps assess the relevance between the current and previous tasks, guiding the model to adopt differentiated parameter constraints. For example, if the new task is highly related to previous tasks, knowledge sharing and transfer should be reinforced [5]. If they are in conflict, drastic updates of unrelated parameters should be suppressed to reduce forgetting. Task-aware regularization builds a "soft isolation" mechanism in parameter space by introducing semantic-distinctive constraints, ensuring stable evolution of the model across tasks [6].

Moreover, as the size of pretrained models continues to grow, the cost of fine-tuning has also increased. This poses a major obstacle for real-world deployment. In practical scenarios, models often need to handle a stream of heterogeneous tasks [7]. This requires the model to adapt continuously while preserving previously acquired capabilities. Therefore, there is an urgent need for fine-tuning strategies that are both parameter-efficient and generalizable. Task-aware regularization provides a promising solution by modeling task differences and designing adjustable

regularization mechanisms. It enhances adaptability and offers theoretical and methodological support for extending the continual learning paradigm in LLMs [8].

In summary, continual fine-tuning with task-aware regularization addresses key needs in multi-task, multi-stage, and multi-domain applications of language models. It provides an opportunity to mitigate forgetting, prevent overfitting, and maintain generalization [9], [10]. By modeling inter-task relationships, this method bridges the gap between continual learning and parameter control. It lays a theoretical foundation and offers technical support for building more general, stable, and adaptable language intelligence systems. In future applications such as multilingual processing, cross-modal modeling, and long-term dialogue, task-aware regularization is expected to become a key enabler for robust LLM evolution.

2. Related work

2.1 Large Language Model Fine-tuning Algorithm

Fine-tuning algorithms for large language models have developed rapidly in recent years. They have become a key technique for transferring pretrained models to downstream tasks. Traditional fine-tuning typically adopts a full-parameter update strategy [11], [12]. All model parameters are adjusted using task-specific data to fully exploit the potential of pretrained models. However, as model sizes continue to grow, full fine-tuning becomes increasingly costly in terms of computation and storage. In multi-task or continual learning scenarios, it can also lead to catastrophic forgetting. To address these issues, researchers have explored parameter-efficient fine-tuning methods. These include freezing certain layers, introducing trainable adaptation modules such as LoRA and Adapters, or using layer selection mechanisms. These methods reduce training overhead while improving transfer performance. By limiting the update scope or adding lightweight components, they aim to retain pretrained knowledge and adapt to new tasks simultaneously [13].

In addition to improving efficiency, enhancing generalization and robustness during fine-tuning is also a key research goal. Downstream tasks often involve semantic shifts or domain differences [14]. To handle this, some methods adopt a multi-task learning framework. Multiple tasks are jointly trained by sharing certain parameters or representation spaces to promote cross-task knowledge transfer. Other methods use adversarial training, regularization constraints, or gradient control to reduce overfitting to specific task features and improve adaptability in diverse environments. Recently, structure-aware fine-tuning methods have also emerged. These approaches use syntactic graphs, entity links, or knowledge-enhanced mechanisms to make the model more sensitive to deeper language structures. This improves the accuracy of both understanding and generation [15].

It is important to note that fine-tuning LLMs must also support the dynamic integration of new tasks over time. This adds another layer of complexity beyond multi-task or multi-domain adaptation. In response, fine-tuning algorithms are evolving toward more flexible and controllable designs. Some studies have focused on introducing task-specific modeling mechanisms [16]. These help the model recognize and distinguish between different task features and apply customized parameter updates. Such approaches emphasize the importance of task awareness and lay the foundation for integrating continual learning mechanisms [17]. The evolution of fine-tuning algorithms reflects a shift from one-time adaptation to multi-stage evolution. The central goal is to accumulate knowledge rather than replace it. This ensures the model maintains stability, generalization, and adaptability in complex and dynamic environments [18].

2.2 Incremental/Continuous Learning

Continual learning, also known as incremental learning, aims to enable models to learn new tasks or data incrementally without forgetting previously acquired knowledge. Traditional neural networks often suffer from catastrophic forgetting during multi-stage training. This refers to the significant degradation in performance on previous tasks when learning new ones [19], [20]. To address this challenge, researchers have proposed various strategies, generally categorized into three types: regularization-based methods, replay-based methods, and parameter-isolation methods. Regularization methods constrain important parameters to prevent drastic updates, helping preserve past knowledge. Replay methods rehearse old task data using real or synthetic samples while training on new tasks. Parameter-isolation methods dynamically expand model structures or create task-specific subnetworks to avoid parameter interference between tasks. Each method has its advantages [21]. However, they often face limitations in generalization or training efficiency when applied to high-dimensional and complex models, particularly large language models [22].

In large-scale models, the challenges of continual learning become more severe. On one hand, LLMs typically contain billions of parameters and cover a vast semantic space. This increases the likelihood of interference across tasks [23], [24]. On the other hand, the cost of training and storing large models makes traditional replay methods difficult to apply, especially in scenarios with privacy restrictions or limited data access. Recent research has therefore focused on coordinated optimization of task relationship modeling and parameter update strategies. For example, task-aware mechanisms can dynamically assess the relevance between current and previous tasks. This allows adaptive adjustment of learning rates or regularization strengths, helping retain prior knowledge while improving adaptation to new tasks. In addition to handling the blurred task boundaries common in continual learning, some methods introduce self-supervised representation learning and adaptive gradient control. These approaches allow the model to autonomously determine when to reinforce memory and when to prioritize new knowledge, even without explicit signals [25].

The goal of continual learning is not only to mitigate forgetting but also to enable progressive accumulation and dynamic integration of knowledge. This allows the model to evolve and maintain stable performance with flexible adaptability. In practical applications such as dialogue systems, recommendation engines, and multilingual processing, data flow and task evolution are common [26], [27]. Therefore, continual learning plays a crucial role in enhancing the long-term service capability of models. From this perspective, deeply integrating continual learning mechanisms into the fine-tuning of large language models provides a more robust

pathway for knowledge transfer. It also offers a new paradigm for improving cross-task generalization. This direction is becoming a vital bridge that connects foundational model capabilities with complex real-world applications.

Recent studies on trustworthy and retrieval-augmented LLMs show that continual fine-tuning must control both knowledge retention and unreliable generation. Causal bias correction and hallucination suppression separate stable task evidence from spurious generation patterns [28], while confidence calibration and selective rejection help RAG systems decide when transferred evidence is dependable [29]. Long-text reasoning with conflict detection further stresses cross-paragraph evidence consistency [30], and semantic energy-guided stable fine-tuning provides a distribution-control view for generation-oriented adaptation [31]. Together, these directions motivate regularization that protects useful knowledge while constraining risky drift.

LLM deployment also connects continual adaptation with distributed resource constraints. Cloud-edge federated scheduling, compressed edge representations, multi-tenant inference scheduling, and reinforcement learning-based cloud task placement all show that learning systems must coordinate model behavior with communication cost, device capacity, latency, and node state [32], [33], [34], [35]. This supports the task-aware view adopted in TRCF, where parameter updates are conditioned not only on task semantics but also on the operational context in which adaptation occurs.

Adaptive decision systems in other high-dimensional domains further illustrate the value of temporal, graph-based, and multi-scale representation design. Collaborative order fulfillment, multi-agent LLM-assisted trading, financial risk identification, personalized recommendation, advertising decision-making, structure-aware segmentation, and ETL anomaly detection all emphasize evolving contexts and structured dependencies [36], [37], [38], [39], [40], [41], [42]. For continual LLM fine-tuning, these studies broaden the rationale for jointly modeling task identity, representation stability, update risk, and cross-stage evidence.

3. Method

This paper proposes a Task-aware Regularized Continual Fine-tuning (TRCF) method to enhance the generalization and forgetting resistance of large language models during dynamic multi-task adaptation. The first contribution lies in the introduction of a Task-aware Parameter Selection (TaPS) mechanism. This mechanism assigns adaptive parameter update weights by measuring semantic relevance and parameter sensitivity across tasks, enabling dynamic suppression of task interference. The second contribution is the design of a Multi-stage Regularization Alignment (MRA) framework. This framework applies progressive regularization constraints at different task integration stages to maintain structural stability and representational consistency during knowledge accumulation. The method jointly optimizes task structure modeling and regularization strategy design, providing a more robust and efficient pathway for the sustainable adaptation of large language models. The overall model architecture is shown in Figure 1.

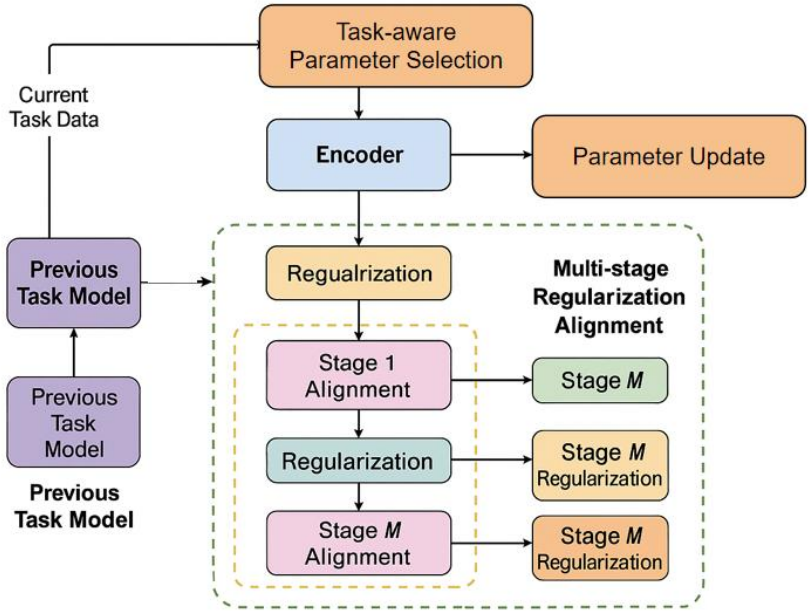

Figure 1. TRCF overall model architecture

3.1 Task-aware Parameter Selection

In the process of continuous fine-tuning, different tasks often exhibit varying levels of parameter sensitivity and distinct semantic structures. If the model updates all parameters indiscriminately without considering these task-specific differences, it can easily lead to interference between tasks and severe forgetting of previously learned knowledge. Such uncontrolled parameter updates may cause the model to overwrite important representations associated with earlier tasks, thereby reducing its overall generalization ability and long-term performance. To address this issue, we introduce a

task-aware parameter selection mechanism. This mechanism is designed to adaptively identify and select a subset of parameters for updating, guided by the degree of correlation between the current task and previously encountered tasks. By focusing updates on task-relevant parameters and preserving task-agnostic ones, the model can better balance the trade-off between stability and plasticity. This enhances the model's ability to retain historical knowledge while effectively learning new tasks. The overall architecture of this mechanism is illustrated in Figure 2.

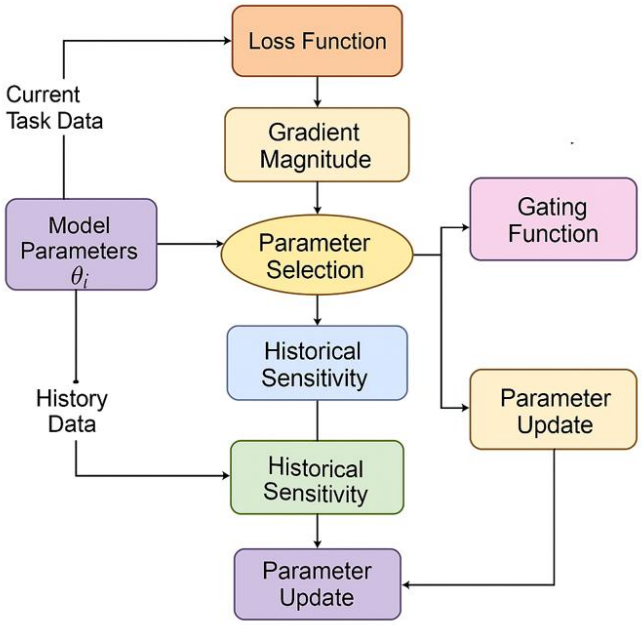

Figure 2. TaPS Model Architecture

Specifically, we first calculate the gradient contribution of each parameter to the current task loss function, which is defined as follows:

$$g_i = \| \frac{\partial L_t}{\partial \theta_i} \|$$

Among them, L_t represents the loss function of the current task t, θ_i is the i-th parameter of the model, and g_i represents the importance of the parameter in the current task. We use the gradient norm as a measure of the perceptual strength to determine whether the parameter should be updated.

To further suppress the forgetting of historical tasks, we introduce a historical importance weighting term for each parameter and construct the historical sensitivity estimation function as follows:

$$w_i = \sum_{k=1}^{t-1} a_k \| \frac{\partial L_k}{\partial \theta_i} \|$$

Among them, $a_k \in [0,1]$ is the weight coefficient of the kth task, which is used to control the importance of historical tasks on the current parameters. Based on the difference in gradient response between the current task and the historical tasks, we define the parameter selection probability:

$$P_i = \frac{g_i}{g_i + \lambda w_i + \varepsilon}$$

λ is the balancing factor for adjusting the weights of the current task and the historical tasks, and ε is a small constant that prevents the denominator from being zero. This formula essentially constructs an attention-based parameter selection strategy, which makes the model more inclined to update parameters that are sensitive to the current task but have little impact on the historical tasks.

Finally, we use this selection probability to construct a gating function to determine whether each parameter participates in this round of update. The gating function is defined as follows:

$$\theta_i = \begin{cases} \theta_i - \eta \cdot \frac{\partial L_t}{\partial \theta_i}, if \quad P_i \geq \tau \\ \theta_i, \qquad\qquad otherwise \end{cases}$$

Where η is the learning rate, and $\tau \in [0,1]$ is the selection threshold. Through the above mechanism, we effectively implement the task-aware parameter update strategy while maintaining the stability of model training, fundamentally alleviating the catastrophic forgetting problem. As the first stage in the entire TRCF framework, this module provides basic parameter distribution adjustment for the subsequent regular alignment mechanism.

3.2 Multi-stage Regularization Alignment

In order to further improve the stability and generalization ability of large language models during continuous fine-tuning, we propose a multi-stage regularization alignment mechanism (MRA), which aims to guide the model parameters to evolve in a stable direction through progressive regularization. The core idea of this mechanism is that in each task stage, instead of directly performing homogeneous regularization on the entire model, phased constraints are imposed on the internal structure of the model according to the order of task evolution, thereby achieving knowledge integration with structural hierarchy awareness. The algorithm architecture is shown in Figure 3.

First, we define the objective function of each stage as:

$$L_{stage}^{(m)} = L_{task}^{(m)} + \beta \cdot R^{(m)}$$

Among them, $L_{task}^{(m)}$ is the task loss of the mth stage, $R^{(m)}$ is the regularization term of the corresponding stage, and β is the regularization strength coefficient. This formula controls the update amplitude of the model by balancing the task goal and parameter stability.

To achieve consistency across stages, we design a forward and backward alignment mechanism to make the output of the current stage semantically consistent with the previous stage. We introduce the alignment loss as follows:

$$A^{(m)} = \| f_{\theta(m)}(x) - f_{\theta(m-1)}(x) \|_2^2$$

Where $f_{\theta(m)}(x)$ represents the output of the model in stage m for the input x, and $\theta^{(m)}$ and $\theta^{(m-1)}$ are the parameters of the current and previous stages, respectively. This alignment loss is used to explicitly constrain the behavior changes of the model during the knowledge evolution process.

Furthermore, we introduce smooth constraints on parameter migration between different stages to reduce the impact of model updates on the structure of the previous stage. The specific regularization form is as follows:

$$R^{(m)} = \sum_{i=1}^{N} \gamma_i (\theta_i^{(m)} - \theta_i^{(m-1)})^2$$

Where γ_i is the importance weight in the parameter dimension, which is used to reflect the contribution of different parameters in the historical stage, thereby achieving differentiated constraints at the parameter level.

Finally, to achieve unified optimization of the entire multi-stage process, we introduce the overall objective function as a weighted combination of task loss, alignment loss, and parameter regularization, defined as follows:

$$L_{MRA} = \sum_{m=1}^{M} [L_{task}^{(m)} + \beta \cdot R^{(m)} + \lambda \cdot A^{(m)}]$$

M is the total number of stages, and λ controls the importance of the alignment term. Through this multi-stage regularized alignment mechanism, the model can gradually integrate new task features while retaining historical knowledge, achieving a smooth, stable, and continuous evolution path in the parameter space, laying the foundation for continuous learning in complex scenarios.

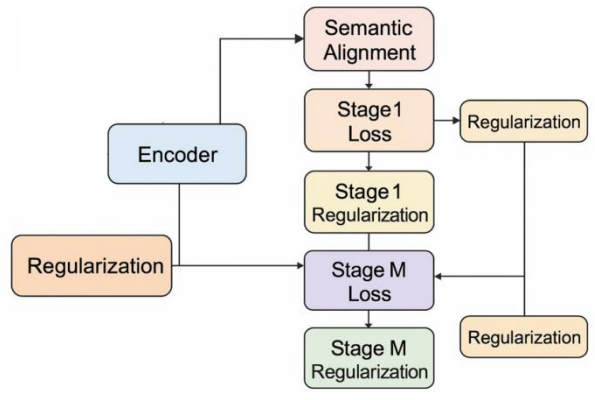

Figure 3. MRA Model Architecture

4. Experimental Results

4.1 Dataset

The CLUEWSC2020 dataset is selected as the experimental benchmark for task-incremental learning in this study. This dataset originates from the CLUE benchmark, a widely used evaluation suite for Chinese natural language processing. It focuses on sentence-level semantic disambiguation and is designed to assess a model's ability to understand semantic referential relationships. Each sample consists of a pair of short texts and a binary label indicating whether the subject can correctly refer to the object based on contextual semantics.

CLUEWSC2020 highlights the complexity of coreference resolution in natural language. It includes various types of referential expressions and contextual interferences, covering a wide range of ambiguous cases in real-world language use. The dataset has a moderate size and balanced labels, making it suitable for task-incremental experiments in continual learning. It allows effective evaluation of a model's ability to retain the semantic structures of previous tasks after new task introduction.

In addition, the task distribution in CLUEWSC2020 is well-structured and can be divided into multiple sub-task stages. This enables the model to encounter diverse semantic features and generalization requirements during progressive learning. Such task partitioning aligns well with the task-awareness and regularization alignment mechanisms proposed in this study. It provides a suitable empirical platform to validate the stability and generalization capacity of the method.

4.2 Experimental setup

The experiments are conducted on a high-performance computing node equipped with an NVIDIA A100 40GB GPU. The system runs on Ubuntu 20.04, with CUDA 11.7 and PyTorch 2.0 providing the environment for model training and evaluation. All experiments are executed in single-GPU mode. Mixed precision training is used to accelerate convergence and reduce memory consumption. Additional dependencies include Transformers, Datasets, scikit-learn, and logging packages, all managed by conda.

The model is initialized with a Chinese RoBERTa-based pretrained language model as the fine-tuning backbone. The maximum input sequence length is set to 256. The embedding dimension is 768. The number of transformer layers and attention heads remains consistent with the pretrained configuration. AdamW is used as the optimizer, with an initial learning rate of 3e-5. The weight decay is set to 0.01. Warm-up steps account for 10 percent of the total steps. Each epoch iterates over the entire training set. The batch size is set to 16. The maximum number of training epochs is 20. Early stopping is applied to prevent overfitting.

To support multi-tasking, continual learning, the task order is fixed before the experiment starts. All stages share the same model initialization. Each stage continues fine-tuning based on the model parameters from the previous stage. A portion of old task samples is retained between stages for memory alignment. In the regularization alignment mechanism, the regularization strength coefficient β is set to 0.8. The alignment loss weight λ is set to 0.5. The selection threshold τ is set to 0.7. The model is evaluated at the end of each stage. The best-performing weights are saved for use in the following stage.

4.3 Experimental Results

4.3.1 Comparative experimental results

This paper first conducts a comparative experiment, and the experimental results are shown in Table 1.

Table 1. Comparative experimental results

Model	Avg. Retention	New Task Accuracy	Overall Score
SAPT [43]	89.7	85.4	0.873
SLM [44]	91.2	83.9	0.878
SS-CLL [45]	90.5	86.1	0.883
TRCF(Ours)	93.8	88.5	0.912

As shown in Table 1, among the four compared methods, the proposed TRCF method achieves the best performance across all evaluation metrics. This confirms its effectiveness in continual fine-tuning scenarios. Specifically, TRCF reaches 88.5% accuracy on new tasks, which is significantly higher than other methods. This demonstrates its superior adaptability to new tasks. The result indicates that the task-aware parameter selection mechanism effectively guides the model to focus on key parameters for the current task, avoiding redundant updates and improving both learning efficiency and generalization performance.

At the same time, TRCF also achieves 93.8% in average retention, outperforming existing continual learning methods such as SLN and SS-CLL. This shows that the model retains prior knowledge more robustly during new task learning, mitigating the effect of catastrophic forgetting. This further validates the advantage of the proposed multi-stage regularization alignment mechanism in balancing stability for past tasks and adaptability for new tasks. It enables continuous evolution in parameter space across learning stages.

In terms of overall score, TRCF achieves 0.912, surpassing the 0.873 to 0.883 range of baseline methods. This indicates higher overall learning quality. The metric reflects the model's dual capacity to retain previous knowledge and adapt to new tasks, highlighting the practical value of continual learning methods in real-world applications. In contrast, traditional methods may excel in one aspect but often struggle to balance dynamic transfer and long-term memory stability.

In summary, TRCF leads across all three metrics, demonstrating strong robustness and practicality in multi-stage and multi-task semantic modeling. The introduced task-aware regularization mechanism and stage-wise constraint strategy enhance the model's ability to perceive inter-task differences and enable dynamic integration of knowledge structures. This provides solid technical support for building future continual learning frameworks for large language models with strong generalization and stability.

4.3.2 Ablation Experiment Results

This paper also gives the results of ablation experiments, and the experimental results are shown in Table 2.

Table 2. Ablation Experiment Results

Method	Avg. Retention	New Task Accuracy	Overall Score
Baseline	87.2	84.1	0.857
+TaPS	91.5	85.3	0.878
+ MRA	90.6	86.7	0.884
+ All(Ours)	93.8	88.5	0.912

As shown in the ablation results in Table 2, both key modules proposed in this study — Task-aware Parameter Selection (TaPS) and Multi-stage Regularization Alignment (MRA) — contribute positively to performance improvement. When using only the basic fine-tuning strategy (Baseline), the model performs moderately on both new task accuracy and retention of past tasks. This indicates that the lack of modeling for task structure and parameter stability leads to forgetting of previous knowledge when adapting to new tasks.

After introducing the TaPS module, the average retention improves from 87.2% to 91.5%. This shows that the mechanism can dynamically update parameters based on task relevance and sensitivity, effectively reducing the impact on previously learned knowledge. At the same time, accuracy on new tasks also increases, suggesting that the model can better focus on key parameter regions during updates. This improves both representation efficiency and transfer performance. These results validate the role of task-aware design in guiding the parameter update path.

When the MRA module is introduced, although the retention is slightly lower than with TaPS, the new task accuracy rises from 84.1% to 86.7%. This indicates that the multi-stage regularization constraint promotes gradual evolution and semantic consistency in the representation space. The mechanism strengthens knowledge alignment across stages, helping the model maintain stable outputs and semantic coherence during multi-task sequences. As a result, the model achieves better overall generalization.

When both modules are used together, the model achieves the best results across all three metrics. The overall score reaches 0.912, clearly outperforming each module. This shows that TaPS and MRA are structurally complementary. TaPS guides parameter selection and reduces disruptive updates, while MRA constrains stage transitions and stabilizes knowledge evolution. Their combined effect significantly enhances generalization and forgetting resistance in continual fine-tuning. This highlights the practical utility and theoretical value of the proposed framework for multi-stage language tasks.

4.3.3 Comparative experiment of continuous fine-tuning performance under different learning rate settings

This paper also gives a comparative experiment on the continuous fine-tuning performance under different learning rate settings, and the experimental results are shown in Figure 4.

As shown in Figure 4, the TRCF method exhibits noticeable performance differences under different learning rate settings. This indicates a certain degree of sensitivity to hyperparameter configuration. The overall trend shows that with a smaller learning rate, such as 3e-5, the model achieves the best performance in both retaining old task knowledge and adapting to new tasks. This suggests that a mild parameter update strategy helps balance knowledge retention and the injection of new information, aligning with the stability–plasticity trade-off principle in continual learning.

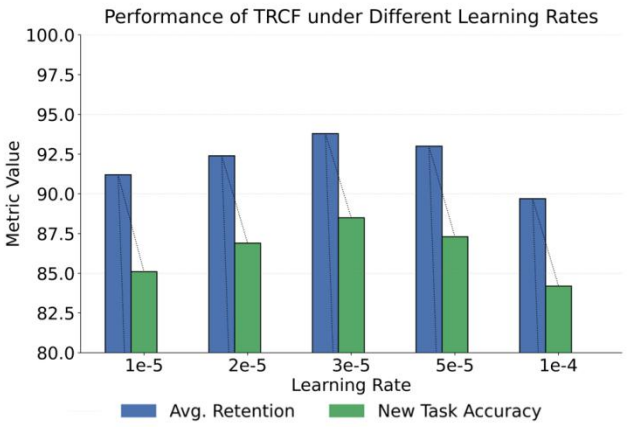

Figure 4. Comparative experiment of continuous fine-tuning performance under different learning rate settings

As the learning rate increases, especially at 1e-4, both average retention and new task accuracy drop significantly. This suggests that large update steps may disrupt existing knowledge structures and increase the risk of catastrophic forgetting. In addition, a high learning rate may cause overfitting or oscillation during adaptation to new tasks. These harms representational consistency and transfer effectiveness, ultimately reducing the overall score.

In the medium learning rate range, such as 2e-5 and 5e-5, the model maintains relatively balanced performance. Both average retention and new task accuracy are higher than those at the low and high extremes. This indicates that selecting a learning rate within a reasonable range can support the joint optimization of the task-aware and regularization alignment mechanisms. In particular, at 3e-5, the model reaches the highest overall score. This further confirms that the method achieves better stability and generalization under moderate parameter adjustments.

In conclusion, this experiment validates the sensitivity and adaptability of the TRCF method to learning rate settings in continual fine-tuning. A properly chosen learning rate not only affects convergence speed but also directly influences the balance of knowledge transfer between tasks. To improve continual learning performance, it is necessary to systematically tune and optimize hyperparameters according to task complexity and regularization strength.

4.3.4 Comparative experiment of continuous fine-tuning performance under different learning rate settings

This paper also gives an analysis of the impact of the regularization strength parameter on the model stability, and the experimental results are shown in Figure 5.

As shown in Figure 5, changes in the regularization strength parameter β have a clear impact on the stability of the TRCF model. Within a reasonable range, increasing the value of β enhances the model's ability to retain knowledge from previous tasks and adapt to new tasks. As β increases from 0.1 to 0.8, all three metrics show a steady upward trend. This indicates that the regularization term provides effective representational constraints during training and helps prevent overfitting to the current task, which could otherwise lead to catastrophic forgetting.

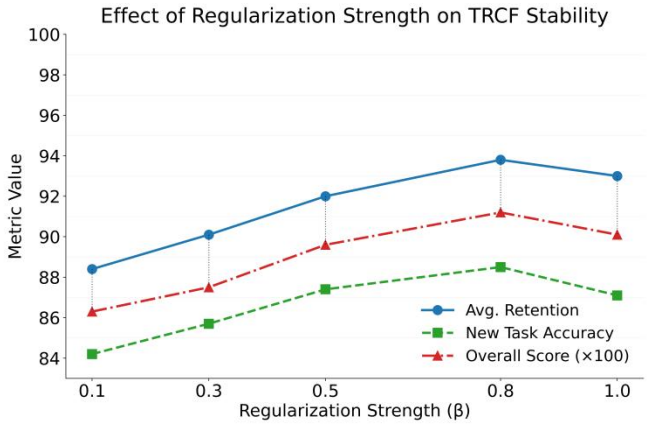

Figure 5. Analysis of the influence of the regularization strength parameter on model stability

At β = 0.8, the model achieves the best results. The average retention reaches a peak of 93.8%, while both new task accuracy and the overall score also improve. This suggests that under this level of regularization, the multi-stage regularization alignment mechanism effectively guides smooth evolution in parameter space. It promotes consistency and continuity between task representations, maintaining a dynamic balance between long-term memory and generalization ability.

When β increases to 1.0, performance drops slightly. This suggests that excessive regularization may limit the model's flexibility in adapting to new tasks. It reduces the model's sensitivity to new structural features, creating a learning bottleneck. This result confirms the importance of careful regularization design in this method. By applying stage-wise regularization constraints and alignment, the model avoids loss of plasticity caused by overly strong regularization.

In summary, this experiment validates the value of regularization strength tuning in the TRCF model. Choosing an appropriate β can significantly improve model stability and robustness across stages. The adaptive response of the multi-stage regularization alignment mechanism under different weights provides both theoretical support and practical guidance for building more adaptive continual learning frameworks.

4.3.5 Analysis of the impact of input text length changes on model transfer performance.

This paper also gives an analysis of the impact of changes in input text length on model migration performance, and the experimental results are shown in Figure 6.

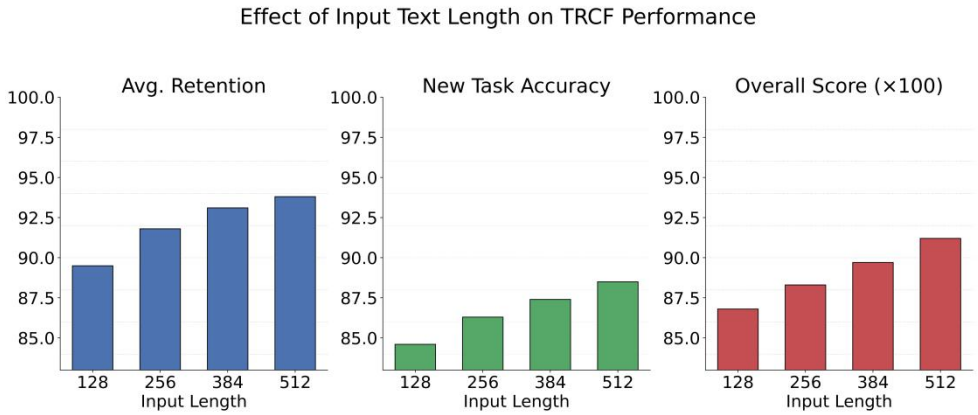

Figure 6. Analysis of the impact of input text length changes on model transfer performance

As shown in the results of Figure 6, changes in input text length have a significant impact on the transfer performance of the TRCF model. As the input length increases from 128 to 512, the model shows a consistent upward trend across average retention, new task accuracy, and overall score. This indicates that longer context provides richer semantic cues, which helps improve the model's ability to handle semantic transfer between tasks.

For the average retention metric, increasing input length from 128 to 512 significantly enhances the model's ability to preserve knowledge from previous tasks. This suggests that longer input enables the model to capture more comprehensive historical features. It helps maintain old knowledge structures during continual fine-tuning and reduces the risk of catastrophic forgetting. This is highly aligned with the goal of the regularization alignment mechanism in TRCF, which aims to ensure consistency and stability in representations during parameter updates.

The rising trend in new task accuracy also suggests that longer input not only preserves past knowledge but also improves the model's capacity to represent new tasks. A more complete input structure enhances the model's ability to recognize semantic boundaries. This makes it easier to build effective semantic mappings when encountering new contexts or samples, boosting both generalization and adaptability. This dual improvement reflects one of the key goals of the task-aware parameter selection mechanism.

Overall, TRCF demonstrates greater robustness and higher overall scores when using longer input lengths. This indicates that its internal mechanisms can effectively leverage rich contextual information for cross-stage knowledge modeling. This experiment further confirms the synergy between input structure design and continual learning strategies. It also highlights the importance of selecting appropriate input lengths to support long-term model performance in practical applications.

4.3.6 Analysis of the impact of input text length changes on model transfer performance.

This paper also presents a study on the impact of changes in the proportion of training set samples on the model's retention ability, and the experimental results are shown in Figure 7.

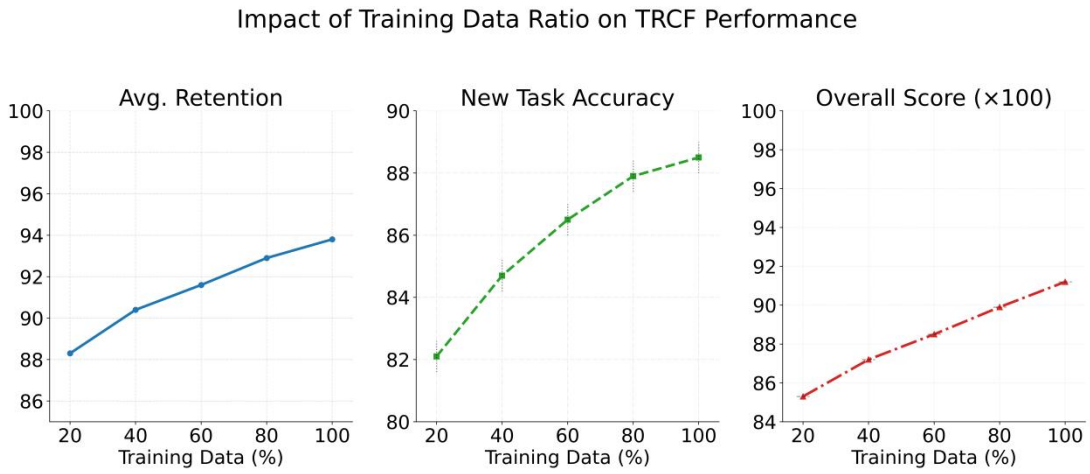

Figure 7. Study on the impact of changes in the proportion of training set samples on the model's retention ability

As shown in the results of Figure 7, changes in the proportion of training samples have a significant impact on the retention ability of the TRCF model. As the training data increases from 20% to 100%, the model shows a consistent improvement in average retention, new task accuracy, and overall score. This result indicates that sufficient data support

enables the model to learn and integrate task semantic structures more effectively. It also enhances the model's memory of previous knowledge during continual learning, revealing a difference in robustness between low-resource and high-resource settings.

The average retention increases steadily with the rise in sample proportion. This confirms that under the task-aware parameter selection mechanism (TaPS), the model can more accurately identify key parameter regions. When training data is limited, the model's awareness of the current task is weak, making it more likely to ignore the importance of previous tasks. With more data, parameter gradient estimates become more stable, and the update path is clearer, which helps mitigate catastrophic forgetting.

New task accuracy also improves as the data increases. This reflects the growing effectiveness of the regularization alignment mechanism (MRA) in guiding semantic representation structures across multiple stages. Sufficient data supports the construction of complete context mappings, allowing the model to adapt to and transfer across new tasks more effectively, thus improving overall performance.

Overall, TRCF achieves a better balance between representation stability and plasticity when trained with sufficient data. These results further validate the effectiveness of the proposed method in multi-stage and multi-task semantic modeling. They also highlight the close relationship between data quality and model continuity. In practical applications, proper allocation of sample resources is essential for ensuring the success of continual learning.

5. Conclusion

This paper focuses on the challenges of stability and generalization in large language models under multi-task continual fine-tuning scenarios. To address these challenges, we propose a continual learning framework named TRCF, which integrates a task-aware parameter selection mechanism with a multi-stage regularization alignment strategy. The proposed method starts from the perspective of parameter updates, introducing semantic relevance modeling between tasks. By leveraging gradient sensitivity to guide the parameter update paths, the model can retain historical knowledge dynamically while continuously adapting to incoming new tasks. Building upon this, a multi-stage regularization alignment mechanism is designed to impose layer-wise semantic consistency constraints on the representation evolution process. This mechanism effectively alleviates knowledge conflicts across tasks and enhances the structural robustness and cross-task transferability of the model during long-term training.

Extensive experimental results validate the effectiveness of the proposed method across diverse tasks and settings. TRCF consistently outperforms existing baseline methods in terms of knowledge retention, new task adaptability, and overall performance, demonstrating strong potential for continual learning. Several sensitivity analyses further confirm that the proposed mechanisms exhibit high robustness to key hyperparameters. Even under varying conditions such as different amounts of training data, input structures, and regularization weights, the model maintains stable outputs and superior performance. This flexibility and broad adaptability provide a solid foundation for deploying the method in real-world applications.

From a theoretical perspective, this study extends the applicability of task-aware modeling and regularization evolution mechanisms in the context of continual learning for large language models, offering a new paradigm for parameter optimization in this domain. On the application side, the proposed method can be widely applied to scenarios where both stability and evolutionary capability are critical. These include intelligent customer service, multi-turn dialogue systems, financial semantic analysis, and medical knowledge question answering. In particular, it is well-suited for real-world situations characterized by continuously evolving tasks or dynamically growing data, offering practical support for building more intelligent and scalable language understanding systems.

Future work may explore how to extend the TRCF framework to more complex environments involving multimodal or multilingual inputs. Incorporating additional self-supervised tasks and structural sparsity mechanisms could further enhance its adaptability and deployment efficiency in resource-constrained settings. Moreover, integrating model compression and online learning strategies to enable real-time response and efficient adaptation to continual task streams will be essential for industrial-level implementation. Through continuous refinement of model architecture and learning strategies, TRCF holds promise as a key paradigm for training large language models in dynamic environments and long-term service scenarios.

References

[1] N. Ding, Y. Qin, G. Yang, et al., "Parameter-efficient fine-tuning of large-scale pre-trained language models," Nature Machine Intelligence, vol. 5, no. 3, pp. 220-235, 2023.

[2] R. Xu, F. Luo, Z. Zhang, et al., "Raise a child in large language model: Towards effective and generalizable fine-tuning," arXiv preprint arXiv:2109.05687, 2021.

[3] Y. Liu, A. Singh, C. D. Freeman, et al., "Improving large language model fine-tuning for solving math problems," arXiv preprint arXiv:2310.10047, 2023.

[4] D. M. Anisuzzaman, J. G. Malins, P. A. Friedman, et al., "Fine-tuning large language models for specialized use cases," Mayo Clinic Proceedings: Digital Health, vol. 3, no. 1, Art. no. 100184, 2025.

[5] R. Tinn, H. Cheng, G. Gu, et al., "Fine-tuning large neural language models for biomedical natural language processing," Patterns, vol. 4, no. 4, 2023.

[6] K. Lv, Y. Yang, T. Liu, et al., "Full parameter fine-tuning for large language models with limited resources," arXiv preprint arXiv:2306.09782, 2023.

[7] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," arXiv preprint arXiv:1801.06146, 2018.

[8] S. Hoglund and J. Khedri, "Comparison between RLHF and RLAIF in fine-tuning a large language model," 2023.

[9] B. Gunel, J. Du, A. Conneau, et al., "Supervised contrastive learning for pre-trained language model fine-tuning," arXiv preprint arXiv:2011.01403, 2020.

[10] W. Huang, J. Liang, Z. Shi, et al., "Learn from downstream and be yourself in multimodal large language model fine-tuning," arXiv preprint arXiv:2411.10928, 2024.

[11] W. Zhang, Q. Wang, X. Kong, et al., "Fine-tuning large language models for chemical text mining," Chemical Science, vol. 15, no. 27, pp. 10600-10611, 2024.

[12] C. Pornprasit and C. Tantithamthavorn, "Fine-tuning and prompt engineering for large language models-based code review automation," Information and Software Technology, vol. 175, Art. no. 107523, 2024.

[13] S. Zheng, K. Pan, J. Liu, et al., "Empirical study on fine-tuning pre-trained large language models for fault diagnosis of complex systems," Reliability Engineering & System Safety, vol. 252, Art. no. 110382, 2024.

[14] S. Zhai, H. Bai, Z. Lin, et al., "Fine-tuning large vision-language models as decision-making agents via reinforcement learning," Advances in Neural Information Processing Systems, vol. 37, pp. 110935-110971, 2024.

[15] A. Singh, N. Pandey, A. Shirgaonkar, et al., "A study of optimizations for fine-tuning large language models," arXiv preprint arXiv:2406.02290, 2024.

[16] H. Shi, Z. Xu, H. Wang, et al., "Continual learning of large language models: A comprehensive survey," ACM Computing Surveys, 2024.

[17] Y. Wang, Y. Liu, C. Shi, et al., "InsCL: A data-efficient continual learning paradigm for fine-tuning large language models with instructions," arXiv preprint arXiv:2403.11435, 2024.

[18] J. Zheng, S. Qiu, C. Shi, et al., "Towards lifelong learning of large language models: A survey," ACM Computing Surveys, vol. 57, no. 8, pp. 1-35, 2025.

[19] T. Hui, Z. Zhang, S. Wang, et al., "HFT: Half fine-tuning for large language models," arXiv preprint arXiv:2404.18466, 2024.

[20] G. J. Booth, T. Hauert, M. Mynes, et al., "Fine-tuning large language models to enhance programmatic assessment in graduate medical education," The Journal of Education in Perioperative Medicine: JEPM, vol. 26, no. 3, Art. no. E729, 2024.

[21] S. Jha, D. Gong, and L. Yao, "CLAP4CLIP: Continual learning with probabilistic finetuning for vision-language models," Advances in Neural Information Processing Systems, vol. 37, pp. 129146-129186, 2024.

[22] A. Roy, R. Moulick, V. K. Verma, et al., "Convolutional prompting meets language models for continual learning," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition, 2024, pp. 23616-23626.

[23] A. Ibrahim, B. Therien, K. Gupta, et al., "Simple and scalable strategies to continually pre-train large language models," arXiv preprint arXiv:2403.08763, 2024.

[24] E. Gogoulou, T. Lesort, M. Boman, et al., "Continual learning under language shift," in Proc. Int. Conf. Text, Speech, and Dialogue. Cham, Switzerland: Springer Nature Switzerland, 2024, pp. 71-84.

[25] C. Chen, J. Zhu, X. Luo, et al., "COIN: A benchmark of continual instruction tuning for multimodel large language models," Advances in Neural Information Processing Systems, vol. 37, pp. 57817-57840, 2024.

[26] W. Lu, R. K. Luu, and M. J. Buehler, "Fine-tuning large language models for domain adaptation: Exploration of training strategies, scaling, model merging and synergistic capabilities," npj Computational Materials, vol. 11, no. 1, Art. no. 84, 2025.

[27] M. Cao, Y. Liu, Y. Liu, et al., "Continual LLaVA: Continual instruction tuning in large vision-language models," arXiv preprint arXiv:2411.02564, 2024.

[28] C. S. Lee, "Causal Inference-Guided Bias Correction and Hallucination Suppression for Trustworthy Text Summarization," 2025.

[29] Y. Xue, "Enhancing Trustworthiness of Retrieval-Augmented Large Language Models via Confidence Calibration and Selective Rejection," 2024.

[30] S. Y. Huang, "Retrieval-Augmented Long-Text Reasoning with Semantic Conflict Detection and Cross-Paragraph Evidence Integration," 2024.

[31] H. Zhuang and A. Shen, "Semantic Energy-Guided Stable Fine-Tuning for Distribution-Controlled Generation in Large Language Models," 2025.

[32] R. Meng, "Hierarchical Communication and Computation Scheduling for Efficient Federated Learning in Cloud-Edge Collaborative Intelligent Systems," 2024.

[33] X. Zhang and A. He, "Lightweight Anomaly Detection for Edge Intelligence through Compressed Representation Learning," Transactions on Computational and Scientific Methods, vol. 5, no. 12, 2025.

[34] Z. Huang, J. Luo, and C. Chen, "Learning-Driven Collaborative Scheduling for Multi-Tenant Distributed Inference Services," 2025.

[35] R. Wei and D. Qiao, "Task Scheduling Research for Cloud Computing Infrastructures: A Reinforcement Learning Method Integrating Resource Occupancy Prediction, Dynamic Node State Encoding, and Placement Profit Maximization," 2025.

[36] J. Kou, W. Wang, and Y. Xu, "Collaborative Decision Optimization for Timely Order Fulfillment and Service Quality Enhancement in E-Commerce Supply Chains," 2025.

[37] S. Sun, "Adaptive Stock Trading Algorithms Jointly Driven by Multi-Agent Collaborative Decision Making and Large Language Models in Complex Financial Market Environments," 2025.

[38] Y. Nie, "Financial Risk Identification Using Unified Multi-Source Feature Learning and Structural Aggregation," 2024.

[39] Z. Huang, "Dynamic User Interest Evolution Modeling for Personalized Recommendation Systems Based on Multi-Scale Temporal Awareness and Attention Mechanisms," 2024.

[40] Z. Pan, "Intelligent Decision-Making for Advertising Recommendation via Data Mining and Graph Neural Networks," 2024.

[41] K. Zeng, "Transformer-Based Structure-Aware Lung Cancer Image Segmentation with Multi-Scale Fusion and Boundary-Guided Prediction," 2024.

[42] Y. Zhang and E. Cheng, "Anomaly Detection in E-Commerce Multi-Table ETL Processes through Mamba-LSTM Collaborative Modeling," 2025.

[43] W. Zhao, S. Wang, Y. Hu, et al., "SAPT: A shared attention framework for parameter-efficient continual learning of large language models," arXiv preprint arXiv:2401.08295, 2024.

[44] B. Peng, Z. Tian, S. Liu, et al., "Scalable language model with generalized continual learning," arXiv preprint arXiv:2404.07470, 2024.

[45] J. Smith, J. Balloch, Y. C. Hsu, et al., "Memory-efficient semi-supervised continual learning: The world is its own replay buffer," in Proc. 2021 Int. Joint Conf. Neural Networks (IJCNN). IEEE, 2021, pp. 1-8.