Recurrent Knowledge Identification and Fusion for Language Model Continual Learning

Yujie Feng[1*], **Xujia Wang**[2*], **Zexin Lu**[1], **Shenghong Fu**[1], **Guangyuan Shi**[1]
Yongxin Xu[3], **Yasha Wang**[3], **Philip S. Yu**[4], **Xu Chu**[3†], **Xiao-Ming Wu**[1†]
[1]The Hong Kong Polytechnic University [2]Tsinghua University
[3]Peking University [4]University of Illinois at Chicago
yujie.feng@connect.polyu.hk, xiao-ming.wu@polyu.edu.hk

Abstract

Continual learning (CL) is crucial for deploying large language models (LLMs) in dynamic real-world environments without costly retraining. While recent model ensemble and model merging methods guided by parameter importance have gained popularity, they often struggle to balance knowledge transfer and forgetting, mainly due to the reliance on static importance estimates during sequential training. In this paper, we present Recurrent-KIF, a novel CL framework for Recurrent Knowledge Identification and Fusion, which enables dynamic estimation of parameter importance distributions to enhance knowledge transfer. Inspired by human continual learning, Recurrent-KIF employs an inner loop that rapidly adapts to new tasks while identifying important parameters, coupled with an outer loop that globally manages the fusion of new and historical knowledge through redundant knowledge pruning and key knowledge merging. These inner-outer loops iteratively perform multiple rounds of fusion, allowing Recurrent-KIF to leverage intermediate training information and adaptively adjust fusion strategies based on evolving importance distributions. Extensive experiments on two CL benchmarks with various model sizes (from 770M to 13B) demonstrate that Recurrent-KIF effectively mitigates catastrophic forgetting and enhances knowledge transfer.[1]

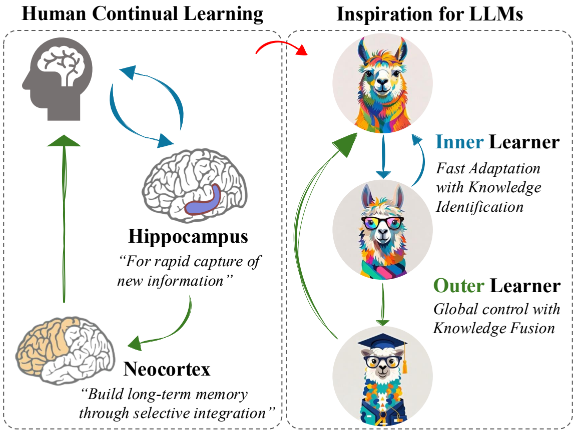


Figure 1: Conceptual illustration of Recurrent-KIF. Inspired by the CLS theory, Recurrent-KIF iteratively employs an inner learner to localize new knowledge and an outer learner to manage the global fusion of knowledge.

1 Introduction

Incorporating continual learning (CL) capability into large language models (LLMs) is essential for enabling them to acquire knowledge from diverse tasks sequentially, a critical requirement for adapting to ever-changing environments without extensive retraining (Wang et al., 2024b; Jiang et al., 2024; Yu et al., 2024; Chang et al., 2024). An effective CL system must address two key challenges: (1) Catastrophic Forgetting (CF) (McCloskey and Cohen, 1989), where previously acquired knowledge is lost when learning new tasks, and (2) Knowledge Transfer (KT) (Ke et al., 2021), which involves leveraging new, related tasks to improve performance on prior tasks, and vice versa.

Recently, model mixture-based methods have emerged as a mainstream approach for CL in LLMs (Chen et al., 2023; Wu et al., 2024a; Rypeść et al., 2024; Chen et al., 2024). By leveraging parameter-efficient fine-tuning (PEFT) techniques, which reduce the computational burden, these methods can be broadly classified into two categories: model ensemble and model merging. Model ensemble methods assign a dedicated PEFT block to each task, capturing task-specific knowledge, which is then stored in a pool and dynamically selected during inference (Shengyuan et al., 2023; Zhu et al., 2024; Wang et al., 2024c). While effective, these methods require storing all task-specific models, leading to high memory consumption that grows with the number of tasks, which limits their scalability for long task sequences.

[*]Equal contribution.
[†]Corresponding author.
[1]https://github.com/WoodScene/Recurrent_KIF

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 27396–27413
July 27 - August 1, 2025

Another line of research focuses on model merging approaches (Dou et al., 2024; Wan et al., 2024; Yadav et al., 2024a), which integrate new task knowledge after training into the historical model, maintaining a single unified model and reducing memory costs compared to model ensemble methods. Consequently, our work primarily focuses on model merging approaches. However, determining which parameters to merge and how to merge remains an open challenge (Qin et al., 2024).

Localizing important parameters in LLMs has recently gained significant interest, a topic widely explored in fields like model pruning and compression (Lu et al., 2021a; Panigrahi et al., 2023; Sun et al., 2023; Yadav et al., 2024b). Building on this foundation, Feng et al. (2024b) and Du et al. (2024) have utilized gradient-based importance metrics, such as Hessian approximations, to identify critical parameters. By selectively or partially merging weights based on parameter importance, these methods have shown effectiveness in CL tasks.

However, the success of these approaches is contingent on the accurate estimation of parameter importance. A key limitation lies in their reliance on ***static importance estimations***, where the parameter importance scores for previous tasks remain unchanged and are not updated during subsequent training. Over time, as model parameters gradually diverge from the state at which the Hessian was originally computed, these unadjusted importance estimates become increasingly inaccurate due to the growing truncation error in the Taylor expansion. This issue is further detailed in the experiments section (Figure 5).

The human brain demonstrates remarkable CL ability through two alternating systems: the hippocampus, which quickly acquires representations for specific experiences, and the neocortex, which selectively consolidates useful memories into long-term storage. This process is known as the Complementary Learning Systems (CLS) theory (McClelland et al., 1995) in neuroscience.

Drawing inspiration from the CLS theory, we propose **Recurrent K**nowledge **I**dentification and **F**usion (Recurrent-KIF), a novel CL framework that dynamically estimates parameter importance and iteratively fuses knowledge. Recurrent-KIF integrates an ***inner learner***, which rapidly adapts to new task-specific knowledge, and an ***outer learner***, which manages the global fusion of new and historical knowledge (see Figure 1).

In detail, the inner learner adapts to new knowledge while utilizing the proposed knowledge identification method to identify important parameters. The outer learner then retrieves historical task information from a memory buffer based on the latest model state, enabling dynamic updates of the importance distributions for previous tasks. Subsequently, a knowledge fusion mechanism is employed to integrate new and historical knowledge by pruning redundant information to mitigate CF and merging key knowledge to enhance KT. Through iterative cycles of multiple rounds of fusion, Recurrent-KIF effectively captures valuable information throughout the model training process, distinguishing it from traditional post-training fusion methods. Each knowledge fusion step adaptively updates fusion weights according to the most recent importance distributions, resulting in smoother and more controlled optimization.

We conduct extensive experiments to assess the effectiveness of Recurrent-KIF on two CL benchmarks for LLMs. The results consistently highlight the superiority of Recurrent-KIF in mitigating CF while exhibiting exceptional KT capabilities, outperforming state-of-the-art methods. Furthermore, Recurrent-KIF exhibits robust scalability across various model architectures and sizes (from 770M to 13B), underscoring its generalization ability.

Our main contributions are summarized as:

- We propose Recurrent-KIF, a novel CL **framework** for recurrent knowledge identification and fusion that dynamically estimates parameter importance and iteratively integrates knowledge.
- We introduce a new **learning paradigm** for Recurrent-KIF, featuring an inner learner that rapidly captures and localizes new information, and an outer learner that globally controls the fusion of new and historical knowledge.
- Extensive **evaluation** validates the effectiveness of Recurrent-KIF in addressing CL challenges.

2 Related Work

2.1 Continual Learning for LLMs

Continual learning (CL) (Zhou et al., 2024) focuses on developing algorithms that accumulate knowledge from non-stationary data. In the LLM era, model mixture-based methods using PEFT have become dominant (Wang et al., 2023b; Huang et al., 2024; Wang et al., 2024e), typically divided into model ensemble and merging approaches.

Model ensemble methods isolate parameters by assigning independent PEFT blocks to each task (Feng et al., 2023; Pham et al., 2023; Ke et al., 2023; Li et al., 2024; He et al., 2024; Wang et al., 2024a; Zhang et al., 2025a). For example, O-LoRA (Wang et al., 2023a) enforces orthogonality among LoRA adapters, while SAPT (Zhao et al., 2024) uses a selection module to combine blocks based on task correlations. While preserving task-specific knowledge, they hinder inter-task transfer and incur high memory overhead as the number of tasks increases, limiting their scalability.

In contrast, model merging methods combine multiple models into a single model (Cheng et al., 2024; Alexandrov et al., 2024; Ren et al., 2024; Shengyuan et al., 2023; Zhang et al., 2025a), alleviating memory constraints. For example, global model merging approaches (Wortsman et al., 2022; Ilharco et al., 2023) perform a weighted fusion of models before and after training, typically assuming that all model weights contribute equally to each task. However, determining which and how to merge parameters remains an open problem. In this paper, we propose Recurrent-KIF, a novel framework that leverages the dynamic importance of parameters across different tasks by employing knowledge identification and fusion techniques to mitigate CF and promote KT.

2.2 Parameter Importance Identification

Identifying important parameters or knowledge regions within LLMs has gained significant attention in the NLP community (Zhao et al., 2023; Liu et al., 2023; Feng et al., 2024a; Xu et al., 2024; Shi et al., 2024; Zhang et al., 2025b). This research improves our understanding of LLMs and enhances their performance across a variety of tasks, including model editing (Wang et al., 2024d; Feng et al., 2025), compression (Zhang et al., 2023; Jiang et al., 2023).

In the context of CL, Du et al. (2024) use the gradient magnitudes to selectively update parameters. Feng et al. (2024b) employ gradient-based metrics to compare the parameter importance distributions of current and historical tasks, merging task-shared regions to promote KT and retaining task-specific regions to prevent CF. However, these approaches are limited by their reliance on static importance estimations for previous tasks, which become outdated as the model evolves.

To address this limitation, Wu et al. (2024b) introduce VR-MCL, a replay-based method that dynamically updates importance information while reducing variance from random sampling. Although VR-MCL achieves dynamic importance estimation for historical tasks, it mainly focuses on preserving task-specific knowledge and does not update task-shared regions, thus limiting KT across tasks. In contrast, inspired by the CLS theory, we propose a dynamic importance estimation method that iteratively updates parameter importance through inner and outer loops. Our approach performs multi-round knowledge fusion, adaptively adjusting the integration of new and historical knowledge based on the latest model state. This method outperforms traditional post-training fusion by enhancing robustness and enabling smoother optimization.

3 Proposed Method: Recurrent-KIF

Problem Formulation Continual learning aims to progressively accumulate knowledge from a sequence of tasks $\{\mathcal{T}_1, \ldots, \mathcal{T}_K\}$. Each task $\mathcal{T}_k$ includes a distinct dataset $\mathcal{D}_k = \left\{\left(x_i^k, y_i^k\right)\right\}_{i=1}^{N_k}$ of size N_k, where $x_i^k \in \mathcal{X}_k$ and $y_i^k \in \mathcal{Y}_k$. The model, parameterized by Θ, is trained sequentially on these tasks to minimize the following objective:

$$\mathcal{L} = \mathbb{E}_{(x,y)\sim \bigcup_{k=1}^{K} \mathcal{D}_k} \left[-\log p_\Theta(y \mid x)\right] \quad (1)$$

In this work, we consider a practical scenario where a small portion of data from previous tasks is stored in a memory buffer to facilitate the CL process. Specifically, we randomly store $|\mathcal{M}|$ samples from each task $\mathcal{T}_i$ in memory $\mathcal{M}_i$. During training, the model is jointly optimized on the new task data $\mathcal{D}_k$ and the memory buffer $\mathcal{M}_{<k}$.

Notation We consider a pre-trained model $\theta \in \mathbb{R}^n$ with n parameters. After training on task $\mathcal{T}_{k-1}$, the model are denoted as θ^{k-1}. Fine-tuning on a new task $\mathcal{T}_k$ produces updated parameters θ^k. The difference $\tau^k = \theta^k - \theta^{k-1}$, referred to as the *task vector* or *training residual* (Ilharco et al., 2023), represents task-specific parameter updates. In the Recurrent-KIF framework, we obtain transient training residuals through each iteration of the inner and outer loops. Specifically, two task vectors are employed to capture and quantify the new knowledge learned in the inner loop and the historical knowledge retrieved in the outer loop.

Overview Recurrent-KIF restructures the training process into multiple iterative learning cycles, each comprising two key components as illustrated in Figure 2: (i) ***Inner Learner with Knowledge***

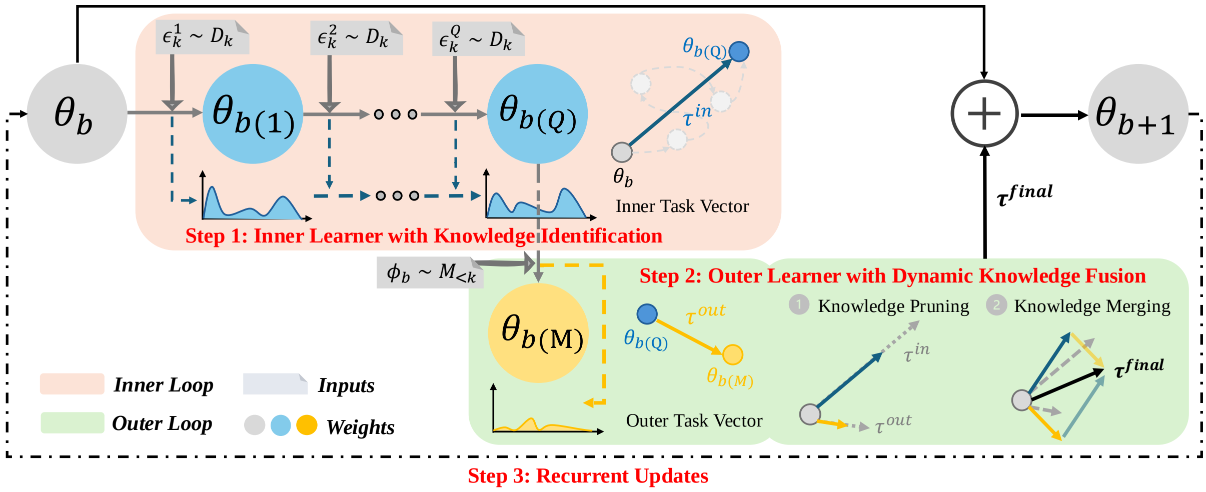


Figure 2: **Iterative update process of Recurrent-KIF for the b-th iteration.** The notation ϵ_k^q represents training samples drawn from $\mathcal{D}_k$, while ϕ_b refers to samples drawn from $\mathcal{M}_{<k}$. **Inner Learner (Step 1):** Performs Q iterations to rapidly adapt to the new task while identifying the parameter importance distribution. **Outer Learner (Step 2):** Retrieves historical task information using memory data and performs knowledge fusion, guided by the importance distributions of both current and historical tasks. **Recurrent Updates (Step 3):** This inner-outer loop cycle is repeated, ensuring that each fusion knowledge step is based on up-to-date importance distributions.

Identification: rapidly acquires new task knowledge while estimating the corresponding parameter importance, and (ii) ***Outer Learner with Knowledge Fusion:*** utilizes a memory buffer to retrieve historical task information. By leveraging the importance distributions of both current and historical tasks, it provides global control for effective knowledge transfer through redundant knowledge pruning and key knowledge merging.

3.1 Inner Learner with Knowledge Identification

Assume the current task is $\mathcal{T}_k$, and the iterative update for the model parameters θ^{k-1} at the b-th iteration are denoted by θ_b^{k-1} [2]. In the inner loop, the model initializes with $\theta_{b(0)} = \theta_b$ and is rapidly updated over Q gradient steps using batch data ϵ_k^q sampled from $\mathcal{D}_k$ at the q-th step. After obtaining $\theta_{b(Q)}$ the task-specific updates are encapsulated in the task vector $\tau_b^{in} \in \mathbb{R}^n$:

$$\tau_b^{\text{in}} = \theta_{b(Q)} - \theta_{b(0)} \tag{2}$$

This task vector captures the knowledge acquired for the current task. However, τ^{in} often contains redundant information, and directly merging it into the model may compromise historical knowledge, leading to catastrophic forgetting. To address this, we propose a knowledge identification technique to identify the key parameters which storing critical knowledge within the task vector.

We use a commonly adopted importance metric in model pruning (Konishi et al., 2023), defined as the magnitude of the gradient-weight product:

$$\bar{I}\left(w_{ij}\right) = \left|w_{ij} \nabla_{w_{ij}} \mathcal{L}\right| \tag{3}$$

where w_{ij} represents trainable parameters.

Due to stochastic batch sampling and training dynamics, the metric in Eq. (3) may be unreliable, introducing variability (Zhang et al., 2022). To mitigate this, we apply an exponential moving average (Zhang et al., 2023) to smooth the trajectory gradients over Q inner loop iterations:

$$I_{b(q)} = \alpha_1 I_{b(q-1)} + (1 - \alpha_1)\, \bar{I}_{b(q)} \tag{4}$$

where α_1 is the smoothing factor, $q \in \{1, 2, ..., Q\}$ is the iteration number in the inner loop, and $I_{b(q)}$ represents smoothed importance. The inner task vector τ_b^{in} and its associated parameter importance I_b^{in} are then passed to the outer learner.

3.2 Outer Learner with Knowledge Fusion

The outer loop manages the global merging of knowledge, guided by parameter importance. To access historical knowledge, after acquiring $\theta_{b(Q)}$, the outer loop samples data ϕ_b from the memory buffer $\mathcal{M}_{<k}$. It then performs several training iterations, updating the parameters to $\theta_{b(M)}$. Then the

[2] For simplicity, we omit the superscripts $k-1$ in subsequent descriptions.

outer task vector $\tau_b^{\text{out}} \in \mathbb{R}^n$, capturing historical task information, is defined as:

$$\tau_b^{\text{out}} = \theta_{b(M)} - \theta_{b(Q)} \tag{5}$$

Dynamic Update of Historical Importance Distribution. While obtaining the outer task vector, we calculate the historical task importance distribution based on the latest model state $\theta_{b(Q)}$, using Eq. (3). The update process is then expressed as:

$$\bar{I}_b^{\text{out}} = \mathbb{P}(\bar{I}_b^{\text{out}} \mid \theta_{b(Q)}) \tag{6}$$

This update, based on conditional probability, enables the computation of the historical importance distribution I_b^{out} using the current model state. This distinguishes it from traditional static importance estimation methods and ensures more accurate knowledge identification. However, the limited sample size from the memory buffer can introduce significant variance in the importance estimates. To address this, we also apply exponential smoothing to the previous outer loop distribution I_{b-1}^{out}:

$$I_b^{\text{out}} = \alpha_2 \bar{I}_b^{\text{out}} + (1 - \alpha_2) I_{b-1}^{\text{out}} \tag{7}$$

where α_2 is the smoothing factor, enhancing stability and robustness in importance estimation.

Knowledge Fusion via Importance-based Binary Mask. Knowledge fusion is guided by the importance distributions I_b^{in} and I_b^{out}. To binarize the importance distributions, a quantile-based threshold δ is applied to select the top 20% of parameters from both I_b^{in} and I_b^{out}. This generates binary masks $m_b^{in} \in \mathbb{R}^n$ and $m_b^{out} \in \mathbb{R}^n$, defined as:

$$m_b^{\text{in}} = \mathbb{I}(I_b^{\text{in}} \geq \delta_b^{in}), m_b^{\text{out}} = \mathbb{I}(I_b^{\text{out}} \geq \delta_b^{out}) \tag{8}$$

where $\mathbb{I}(\cdot)$ is the indicator function that outputs 1 if the condition is met and 0 otherwise. Knowledge fusion is then performed as follows:

$$\theta_{b+1} = \theta_b + (m_b^{\text{in}} \odot \tau_b^{\text{in}} + m_b^{\text{out}} \odot \tau_b^{\text{out}}) \tag{9}$$

where $\odot$ denotes element-wise multiplication.

This knowledge fusion mechanism provides precise global control, effectively tackling key challenges in CL. First, redundant information in the task vectors τ^{in} and τ^{out} is filtered out via the mask operation. Second, task-shared knowledge is effectively merged to facilitate knowledge transfer. Lastly, task-specific knowledge is preserved to prevent catastrophic forgetting.

The inner and outer loops operate iteratively, enabling multi-round fusion of knowledge. This iterative process facilitates the capture and absorption of useful information generated during training, providing smoother optimization compared to traditional post-training fusion methods. Detailed implementation of Recurrent-KIF algorithm is provided in the Appendix (Algorithm 1).

4 Experiments and Analysis

Dataset We adopt the experimental setup from Du et al. (2024), using two CL benchmark datasets: (i) **Standard CL Benchmark**, which consists of five text classification tasks from Zhang et al. (2015): AG News, Amazon Reviews, Yelp Reviews, DBpedia, and Yahoo Answers. (ii) **Long Sequence Benchmark**, a more challenging evaluation scenario comprising 15 tasks (Razdaibiedina et al., 2023): five from the Standard CL Benchmark, four from the GLUE benchmark (Wang, 2018), five from SuperGLUE (Wang et al., 2019), and the IMDB Movie Reviews dataset (Maas et al., 2011). Following Wang et al. (2023a), we sample 1000 instances for training on each task and reserve 500 per class for validation. Three task sequences are evaluated for each benchmark, with detailed descriptions and orderings provided in Appendix C.

Metrics Let $a_{i,j}$ denote the testing performance on task $\mathcal{T}_i$ after training on task $\mathcal{T}_j$. We evaluate the overall performance (OP) (Chaudhry et al., 2018) and backward transfer (BWT) (Lopez-Paz and Ranzato, 2017) after training on the final task:

$$\textbf{OP} = \frac{1}{K} \sum_{i=1}^{K} a_{i,K} \tag{10}$$

$$\textbf{BWT} = \frac{1}{K-1} \sum_{i=1}^{K-1} (a_{i,K} - a_{i,i}) \tag{11}$$

Baselines We compare Recurrent-KIF against a range of baseline methods, including traditional CL approaches, recent PEFT-based model ensemble and merging methods. (1) ***SeqLoRA****:* LoRA parameters are trained on a task sequence without regularization or sample replay. (2) ***IncLoRA****:* incremental learning of LoRA parameters without regularization or sample replay. (3) ***LoRAReplay****:* LoRA fine-tuning with a memory buffer. (4) ***EWC (Kirkpatrick et al., 2017)****:* finetune LoRA with a regularization loss to prevent interference with

Method	Standard CL benchmarks OP↑	BWT↑	Long Sequence Benchmark OP↑	BWT↑
SeqLoRA	43.7	-50.4	11.6	-73.4
IncLoRA	66.4	-20.0	61.2	-26.7
LoRAReplay	68.8	-11.7	70.9	-15.4
EWC* (Kirkpatrick et al., 2017)	50.3	-	45.1	-
L2P* (Wang et al., 2022b)	60.7	-	56.1	-16.3
LFPT5* (Qin and Joty, 2021)	72.7	-	69.2	-12.8
MoELoRA* (Luo et al., 2024)	54.1	-	27.6	-
O-LoRA* (Wang et al., 2023a)	75.8	-3.8	69.6	-4.1
TaSL (Feng et al., 2024b)	76.3	-4.0	74.4	-5.3
VR-MCL (Wu et al., 2024b)	76.0	-3.7	74.8	-4.9
MIGU* (Du et al., 2024)	76.6	-	76.5	-
Recurrent-KIF (ours)	**78.4**	**-2.8**	**77.8**	**-3.6**
MTL	80.3	-	81.8	-
SAPT-LoRA (Zhao et al., 2024)	-	-	82.0	-1.3

Table 1: Overall results on two CL benchmarks using the T5-large model. We report Overall Performance (OP) and Backward Transfer (BWT) after training on the final task. All results are averaged over three different task orders. Methods marked with $*$ are copied from previous papers. The last two rows represent upper bound performance.

previous tasks. (5) ***L2P* (*Wang et al., 2022b*)*:*** dynamically selects and updates prompts from a pool on an instance-by-instance basis. (6) ***LFPT5 (Qin and Joty, 2021):*** learns a soft prompt that solves tasks and generates training samples for replay. (7) ***O-LoRA (Wang et al., 2023a):*** extends IncLoRA to learn different LoRAs in orthogonal subspaces. (8) ***MoELoRA (Luo et al., 2024):*** a vanilla MoE with LoRA number equals to the task number. (9) ***SAPT (Zhao et al., 2024):*** uses pseudo samples and a shared attention framework to align PEFT block learning and selection (10) ***TaSL (Feng et al., 2024b):*** selectively updates or retains skill regions based on parameter importance. (11) ***MIGU (Du et al., 2024):*** updates important parameters based on gradient magnitude. (12) ***VR-MCL (Wu et al., 2024b):*** dynamically updates historical task parameter importance distributions using memory replay. Additionally, multi-task learning with LoRA, referred to as ***MTL***, serves as the upper bound.

Training Details We evaluate Recurrent-KIF using two distinct language model architectures: the encoder-decoder T5 model (Raffel et al., 2020) (T5-large and T5-xl), and the decoder-only LLaMA model (Touvron et al., 2023) (LLaMA2-7B and LLaMA2-13B)[3]. Hyperparameters α_1 and α_2 in Eq. (4) and Eq. (7) are set to 0.55, with the number of inner loop iterations Q set to 8, and the number of outer loop iterations set to 4. Following Zhao et al. (2024), 2% of the original training set is used for replay samples. All experiments are averaged over 3 runs. More details are in Appendix D.

[3]Due to being limited by an academic computing budget, we employed 8-bit quantization for LLaMA models.

4.1 Main Results

The overall CL results using the same T5-large backbone are summarized in Table 1.

Our Recurrent-KIF effectively addresses the challenges of CF and KT simultaneously. Compared to both traditional CL methods (LoRAReplay, L2P) and model ensemble-based methods (MoELoRA, O-LoRA), Recurrent-KIF outperforms them in both CF (increasing average OP from 72.7% to 78.1% compared to O-LoRA) and KT (increasing average BWT from -13.6% to -3.2% compared to LoRAReplay). SAPT achieves the highest performance by leveraging generative replay-based data augmentation, surpassing MTL result. However, it relies heavily on external data synthesis, which can be costly in LLM settings.

Moreover, when compared to parameter importance-based methods like TaSL and VR-MCL, Recurrent-KIF consistently delivers the best OP and BWT scores. Notably, Recurrent-KIF outperforms the state-of-the-art CL method, MIGU, increasing OP from 76.6% to 78.1%. These results underscore the effectiveness of our recurrent knowledge identification and fusion framework, validating the advantages of dynamic parameter importance estimation in mitigating forgetting and promoting knowledge transfer.

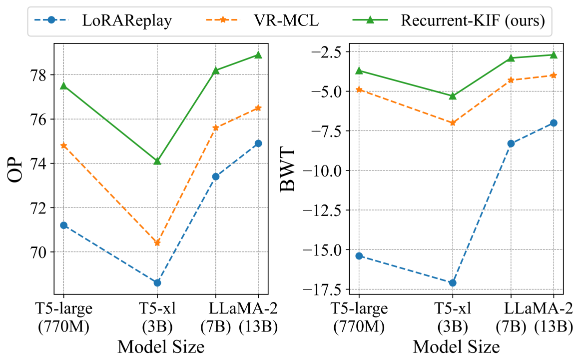
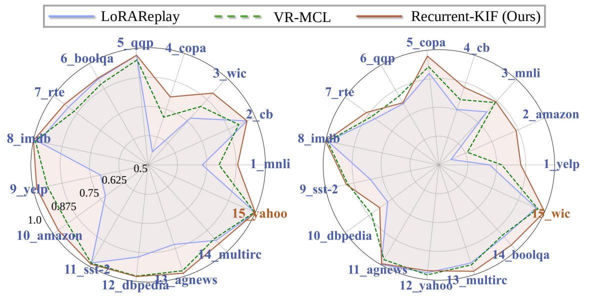


Figure 3: Performance of Recurrent-KIF with different backbones on the Long Sequence Benchmark.



(a) T5-large on Task Order 1

(b) LLaMA2-7B on Task Order 3

Figure 4: **Impact of Catastrophic Forgetting in Continual Learning.** After fine-tuning on the final task (in orange), Recurrent-KIF demonstrates superior resistance to performance decline on previously learned tasks (in blue), outperforming baseline methods.

Recurrent-KIF demonstrates consistent superiority across various backbones. To further validate the robustness of Recurrent-KIF, we conduct experiments across different backbones (Figure 3). Across all backbone sizes, from 770M to 13B, Recurrent-KIF consistently outperforms all baseline models. For instance, using the LLaMA2-7B backbone, Recurrent-KIF boosts the OP metric from 75.6% to 78.2% compared to VR-MCL. These results emphasize the critical role of accurate parameter importance estimation and demonstrate the robust generalization capability of Recurrent-KIF across different model scales.

Dynamic importance estimation enables effective knowledge retention and transfer. Figure 4 illustrates the performance of various methods across all historical tasks after completing the final task. It highlights that Recurrent-KIF optimally restores the model's performance on previous tasks, with significant improvements on Amazon and Copa. Notably, on IMDB and AG News, Recurrent-KIF performs comparably to multi-task learning results. These demonstrate that Recurrent-KIF strikes an effective balance between preserving prior knowledge and excelling in new tasks.

4.2 Ablation Study

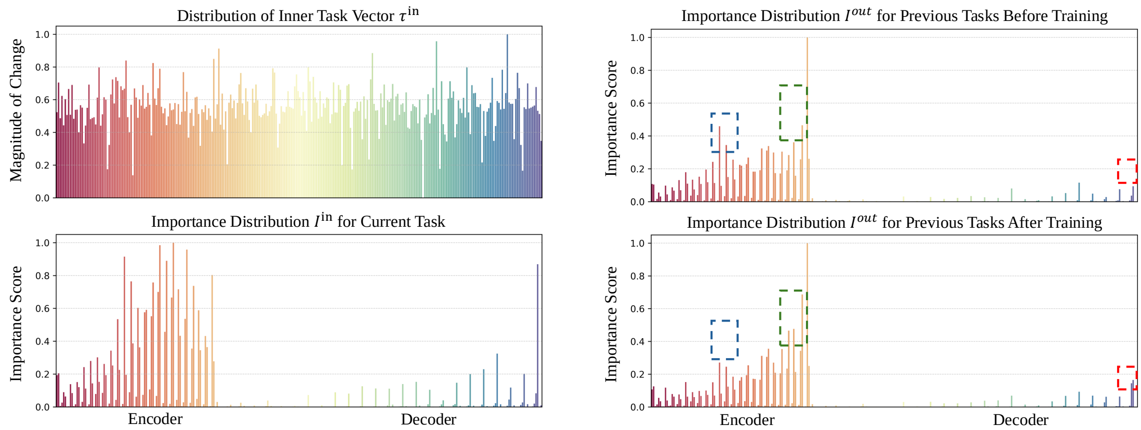
We conduct ablation studies to assess the effectiveness of the proposed techniques in Recurrent-KIF. The results for task order 1 on the Long Sequence Benchmark are shown in Table 2. Additional experiments, such as time complexity analysis, the impact of memory size, and hyperparameter sensitivity, are provided in Appendix B.

Effect of Dynamic Importance Estimations. To validate the role of dynamic importance estimation, we replace it with a static version ("- DIE"), where importance scores for historical tasks remain fixed after their initial computation. The significant performance decline (3.1% on OP and 1.4% on BWT) highlights the necessity of dynamically updating historical task importance distributions. By maintaining up-to-date importance scores, our approach improves both robustness and accuracy, thereby enhancing knowledge retention and transfer.

Method	OP	BWT
Recurrent-KIF	**77.9**	**-3.4**
- DIE	74.8	-4.8
- KI	52.3	-21.5
+ GM	72.1	-11.2
+ Adaptive	76.1	-4.1
- Share	75.8	-4.3

Table 2: Ablation study. "- DIE", "- KI", "- Share" refer to the removal of dynamic importance estimation, knowledge identification, and task-shared region updates, respectively. "+ GM", "+ Adaptive" represent replacing the knowledge fusion mechanism with global merging and adaptive merging strategy, respectively.

Effect of Importance-Based Binary Mask Strategy in Knowledge Fusion. We replace our knowledge fusion mechanism with three alternative model merging strategies: (i) Without knowledge identification ("- KI"): Directly merge τ^{in} and τ^{out} in Eq.(9) without applying importance-based fusion. (ii) Global importance-based merging ("+ GM"): Use a global weighted sum of importance scores I_b^{in} and I_b^{out} for fusion instead of applying element-wise masking. (iii) Adaptive Fusion (Yang et al., 2024) ("+ Adaptive"): A soft-masking method that uses raw importance scores directly



(a) Normalized inner task vector and corresponding parameter importance distribution for the current task.

(b) Normalized parameter importance scores for a historical task across model states.

Figure 5: Visualizations of task vector and parameter importance distributions on T5-large.

for fusion instead of binary masks. Additionally, we evaluate the effectiveness of updating the task-shared region by introducing "- Share", where m^{in} is set to 0 when both m^{in} and m^{out} are 1.

The results in Table 2 confirm the effectiveness of importance-based binary masking in filtering redundant information and preserving task-specific knowledge. Moreover, disabling updates to the task-shared region leads to performance drops of 2.1% and 0.9% on two evaluation metrics, demonstrating that updating task-shared parameters is critical for effective knowledge transfer between tasks.

Effect of Multi-Round Knowledge Fusion. We compare multi-round fusion with traditional single-step fusion methods and analyze the impact of the number of knowledge fusions by adjusting the inner loop iteration size Q. In Recurrent-KIF, with the total number of iterations for the inner loop fixed at N', increasing Q reduces the number of fusion steps, which is N'/Q. A detailed analysis and the model's time complexity are provided in Appendix B.5.

Figure 6 presents two key findings: (i) Multi-round fusion consistently outperforms single-step fusion by leveraging intermediate training information, resulting in smoother knowledge integration, similar to the distinction between gradient descent and stochastic gradient descent; and (ii) performance improves with the number of fusion steps up to a certain point, after which diminishing returns are observed. This decline occurs because excessive fusion introduces noise and redundant updates, disrupting the balance between new and

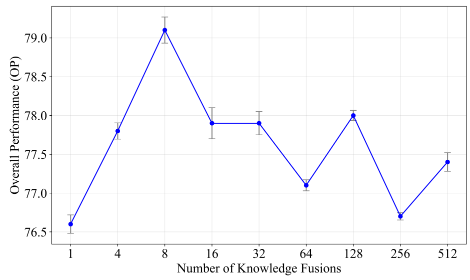


Figure 6: Ablation study on the number of fusions.

prior knowledge, and potentially causing overfitting to unconverged intermediate states.

4.3 Visualization

We present two key visualizations to analyze the effectiveness of our proposed methods:

Can the Magnitude of the Task Vector Reflect Parameter Importance? We explore the relationship between task vector magnitude and parameter importance scores. As shown in Figure 5(a), although the magnitude of parameter updates is generally large, only a subset of parameters, mainly in the encoder, are truly important. This indicates that a large portion of the parameters are redundant, highlighting the need for our importance-based knowledge fusion mechanism.

Does the Importance Distribution of Historical Tasks Change with Model State? Figure 5(b) illustrates the shift in the importance distribution of historical tasks before and after training on a new task. While the overall distribution remains stable across model states, notable changes in specific importance scores are observed, highlighted by the dashed box. This demonstrates the value of dynamic estimation, enabling more precise identification of key parameters and enhancing knowledge fusion across tasks. A more detail analysis is provided in Appendix A.

5 Conclusion

In this paper, we introduce Recurrent Knowledge Identification and Fusion (Recurrent-KIF), a novel CL framework that dynamically estimates the importance of parameters for previous tasks. Recurrent-KIF iteratively employs an inner learner to localize new knowledge and an outer learner to manage the global fusion of knowledge, enabling real-time and adaptive adjustments to the fusion strategy based on evolving importance distributions. Extensive experiments demonstrate the effectiveness of Recurrent-KIF in addressing continual learning challenges.

Limitations

We acknowledge two limitations in this work. Firstly, Recurrent-KIF is a rehearsal-based method. The outer loop relies on memory data to retrieve and dynamically update the parameter importance distributions of historical tasks. This reliance may limit its applicability in scenarios where privacy concerns or data retention restrictions are present. Generative replay techniques could provide a solution by simulating the distribution of previous tasks without direct access to historical data.

Secondly, the time complexity of Recurrent-KIF increases with larger backbone models, primarily due to element-wise operations and multi-round fusion. For element-wise operations, global merging strategies have proven suboptimal, highlighting the need for balanced fusion granularity. Future work could explore focusing on specific important layers or adopting modular approaches to enhance efficiency. For multi-round fusion, we could further investigate how fusion frequency impacts performance and analyze the semantic knowledge learned at different stages of the training process. This could help minimize unnecessary iterations, while still preserving the benefits of iterative integration.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback and the support of the PolyU Research Student Attachment Programme.

References

Anton Alexandrov, Veselin Raychev, Mark Mueller, Ce Zhang, Martin Vechev, and Kristina Toutanova. 2024. Mitigating catastrophic forgetting in language transfer via model merging. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17167–17186.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

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

Shengyuan Chen, Qinggang Zhang, Junnan Dong, Wen Hua, Qing Li, and Xiao Huang. 2024. Entity alignment with noisy annotations from large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Wuyang Chen, Yanqi Zhou, Nan Du, Yanping Huang, James Laudon, Zhifeng Chen, and Claire Cui. 2023. Lifelong language pretraining with distribution-specialized experts. In *International Conference on Machine Learning*, pages 5383–5395. PMLR.

Feng Cheng, Ziyang Wang, Yi-Lin Sung, Yan-Bo Lin, Mohit Bansal, and Gedas Bertasius. 2024. Dam: Dynamic adapter merging for continual video qa learning. *arXiv preprint arXiv:2403.08755*.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945, Bangkok, Thailand. Association for Computational Linguistics.

Wenyu Du, Shuang Cheng, Tongxu Luo, Zihan Qiu, Zeyu Huang, Ka Chun Cheung, Reynold Cheng, and Jie Fu. 2024. Unlocking continual learning abilities in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6503–6522.

Yujie Feng, Xu Chu, Yongxin Xu, Zexin Lu, Bo Liu, Philip S Yu, and Xiao-Ming Wu. 2024a. Kif: Knowledge identification and fusion for language model continual learning. *arXiv preprint arXiv:2408.05200*.

Yujie Feng, Xu Chu, Yongxin Xu, Guangyuan Shi, Bo Liu, and Xiao-Ming Wu. 2024b. Tasl: Continual dialog state tracking via task skill localization and consolidation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1266–1279.

Yujie Feng, Bo Liu, Xiaoyu Dong, Zexin Lu, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. 2024c. Continual dialogue state tracking via reason-of-select distillation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7075–7087.

Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. 2023. Towards llm-driven dialogue state tracking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 739–755.

Yujie Feng, Liming Zhan, Zexin Lu, Yongxin Xu, Xu Chu, Yasha Wang, Jiannong Cao, Philip S Yu, and Xiao-Ming Wu. 2025. Geoedit: Geometric knowledge editing for large language models. *arXiv preprint arXiv:2502.19953*.

Jinghan He, Haiyun Guo, Kuan Zhu, Zihan Zhao, Ming Tang, and Jinqiao Wang. 2024. Seekr: Selective attention-guided knowledge retention for continual learning of large language models. *arXiv preprint arXiv:2411.06171*.

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*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Gangwei Jiang, Caigao Jiang, Zhaoyi Li, Siqiao Xue, Jun Zhou, Linqi Song, Defu Lian, and Ying Wei. 2024. Interpretable catastrophic forgetting of large language model fine-tuning via instruction vector. *arXiv preprint arXiv:2406.12227*.

Xinke Jiang, Ruizhe Zhang, Yongxin Xu, Rihong Qiu, Yue Fang, Zhiyuan Wang, Jinyi Tang, Hongxin Ding, Xu Chu, Junfeng Zhao, et al. 2023. Hykge: A hypothesis knowledge graph enhanced framework for accurate and reliable medical llms responses. *arXiv preprint arXiv:2312.15883*.

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. 2021. Achieving forgetting prevention and knowledge transfer in continual learning. *Advances in Neural Information Processing Systems*, 34:22443–22456.

Zixuan Ke, Bing Liu, Wenhan Xiong, Asli Celikyilmaz, and Haoran Li. 2023. Sub-network discovery and soft-masking for continual learning of mixed tasks. *arXiv preprint arXiv:2310.09436*.

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

Tatsuya Konishi, Mori Kurokawa, Chihiro Ono, Zixuan Ke, Gyuhak Kim, and Bing Liu. 2023. Parameter-Level Soft-Masking for Continual Learning. In *Proc. of ICML*.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024. Revisiting catastrophic forgetting in large language model tuning. *arXiv preprint arXiv:2406.04836*.

Bo Liu, Liming Zhan, Zexin Lu, Yujie Feng, Lei Xue, and Xiao-Ming Wu. 2023. How good are large language models at out-of-distribution detection? *arXiv preprint arXiv:2308.10261*.

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

Zexin Lu, Keyang Ding, Yuji Zhang, Jing Li, Baolin Peng, and Lemao Liu. 2021a. Engage the public: Poll question generation for social media posts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 29–40.

Zexin Lu, Jing Li, Yingyi Zhang, and Haisong Zhang. 2021b. Getting your conversation on track: Estimation of residual life for conversations. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 1036–1043. IEEE.

Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. 2024. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models. *arXiv preprint arXiv:2402.12851*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419.

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

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Abhishek Panigrahi, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora. 2023. Task-specific skill localization in fine-tuned language models. In *International Conference on Machine Learning*, pages 27011–27033. PMLR.

Quang Pham, Chenghao Liu, and Steven CH Hoi. 2023. Continual learning, fast and slow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chengwei Qin and Shafiq Joty. 2021. Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. *arXiv preprint arXiv:2110.07298*.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. 2024. Large language models meet nlp: A survey. *arXiv preprint arXiv:2405.12819*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. *arXiv preprint arXiv:2301.12314*.

Weijieying Ren, Xinlong Li, Lei Wang, Tianxiang Zhao, and Wei Qin. 2024. Analyzing and reducing catastrophic forgetting in parameter efficient tuning. *arXiv preprint arXiv:2402.18865*.

Grzegorz Rypeść, Sebastian Cygert, Valeriya Khan, Tomasz Trzciński, Bartosz Zieliński, and Bartłomiej Twardowski. 2024. Divide and not forget: Ensemble of selectively trained experts in continual learning. *arXiv preprint arXiv:2401.10191*.

Chen Shengyuan, Yunfeng Cai, Huang Fang, Xiao Huang, and Mingming Sun. 2023. Differentiable neuro-symbolic reasoning on large-scale knowledge graphs. *Advances in Neural Information Processing Systems*, 36.

Guangyuan Shi, Zexin Lu, Xiaoyu Dong, Wenlong Zhang, Xuanyu Zhang, Yujie Feng, and Xiao-Ming Wu. 2024. Understanding layer significance in llm alignment. *arXiv preprint arXiv:2410.17875*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

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

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge fusion of large language models. *arXiv preprint arXiv:2401.10491*.

Alex Wang. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. 2024a. Self-expansion of pre-trained models with mixture of adapters for continual learning. *arXiv preprint arXiv:2403.18886*.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024b. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Mingyang Wang, Heike Adel, Lukas Lange, Jannik Strötgen, and Hinrich Schütze. 2024c. Rehearsal-free modular and compositional continual learning for language models. *arXiv preprint arXiv:2404.00790*.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. 2023a. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671.

Xiaohan Wang, Shengyu Mao, Ningyu Zhang, Shumin Deng, Yunzhi Yao, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024d. Editing conceptual knowledge for large language models. *arXiv preprint arXiv:2403.06259*.

Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. 2024e. Inscl: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. *arXiv preprint arXiv:2403.11435*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022a. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

Zhicheng Wang, Yufang Liu, Tao Ji, Xiaoling Wang, Yuanbin Wu, Congcong Jiang, Ye Chao, Zhencong

Han, Ling Wang, Xu Shao, et al. 2023b. Rehearsal-free continual language learning via efficient parameter isolation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10933–10946.

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

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR.

Junhong Wu, Yuchen Liu, and Chengqing Zong. 2024a. F-malloc: Feed-forward memory allocation for continual learning in neural machine translation. *arXiv preprint arXiv:2404.04846*.

Yichen Wu, Long-Kai Huang, Renzhen Wang, Deyu Meng, and Ying Wei. 2024b. Meta continual learning revisited: Implicitly enhancing online hessian approximation via variance reduction. In *The Twelfth International Conference on Learning Representations*.

Yongxin Xu, Xu Chu, Kai Yang, Zhiyuan Wang, Peinie Zou, Hongxin Ding, Junfeng Zhao, Yasha Wang, and Bing Xie. 2023. Seqcare: Sequential training with external medical knowledge graph for diagnosis prediction in healthcare data. In *Proceedings of the ACM Web Conference 2023*, pages 2819–2830.

Yongxin Xu, Xinke Jiang, Xu Chu, Rihong Qiu, Yujie Feng, Hongxin Ding, Junfeng Zhao, Yasha Wang, and Bing Xie. 2025. Dearllm: Enhancing personalized healthcare via large language models-deduced feature correlations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 941–949.

Yongxin Xu, Ruizhe Zhang, Xinke Jiang, Yujie Feng, Yuzhen Xiao, Xinyu Ma, Runchuan Zhu, Xu Chu, Junfeng Zhao, and Yasha Wang. 2024. Parenting: Optimizing knowledge selection of retrieval-augmented language models with parameter decoupling and tailored tuning. *arXiv preprint arXiv:2410.10360*.

Prateek Yadav, Colin Raffel, Mohammed Muqeeth, Lucas Caccia, Haokun Liu, Tianlong Chen, Mohit Bansal, Leshem Choshen, and Alessandro Sordoni. 2024a. A survey on model moerging: Recycling and routing among specialized experts for collaborative learning. *arXiv preprint arXiv:2408.07057*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024b. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations*.

Dianzhi Yu, Xinni Zhang, Yankai Chen, Aiwei Liu, Yifei Zhang, Philip S Yu, and Irwin King. 2024. Recent advances of multimodal continual learning: A comprehensive survey. *arXiv preprint arXiv:2410.05352*.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. 2025a. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Qingru Zhang, Simiao Zuo, Chen Liang, Alexander Bukharin, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2022. Platon: Pruning large transformer models with upper confidence bound of weight importance. In *International Conference on Machine Learning*, pages 26809–26823. PMLR.

Ruizhe Zhang, Yongxin Xu, Yuzhen Xiao, Runchuan Zhu, Xinke Jiang, Xu Chu, Junfeng Zhao, and Yasha Wang. 2025b. Knowpo: Knowledge-aware preference optimization for controllable knowledge selection in retrieval-augmented language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25895–25903.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Haiyan Zhao, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. 2023. Does continual learning equally forget all parameters? In *International Conference on Machine Learning*, pages 42280–42303. PMLR.

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024. Sapt: A shared attention framework for parameter-efficient continual learning of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11641–11661.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. 2024. Continual learning with pre-trained models: A survey. *arXiv preprint arXiv:2401.16386*.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. 2024. Llama-moe: Building mixture-of-experts from llama with continual pre-training. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15913–15923.

A Limitations of Static Parameter Importance Estimation

In this section, we empirically demonstrate the limitations of static parameter importance estimation by analyzing how historical parameter importance distributions change under different conditions. Static importance estimation assumes that the importance scores of historical tasks remain fixed, which is both inaccurate and introduces biases during the knowledge fusion process. To highlight the dynamic nature of parameter importance, we analyze the following two aspects:

Changes in Historical Importance After Learning Different New Tasks. We investigate how the parameter importance distribution for a fixed historical task changes after the model is fine-tuned on different new tasks. Specifically, the model is trained sequentially on various new tasks, and the importance scores of the same historical task are re-evaluated using the updated model parameters. As shown in Figure 7, although the regions identified as important for the historical task remain largely consistent after learning different new tasks, the specific importance values exhibit noticeable differences. This variability demonstrates that historical parameter importance is heavily influenced by the specific characteristics of the new task, making static importance estimation insufficient for accurately capturing the evolving model dynamics.

Temporal Changes in Historical Importance During New Task Training. We further analyze how the historical parameter importance distribution evolves during the training process of a single new task. As shown in Figure 8, by periodically evaluating the importance scores of a fixed historical task at different stages of training, we also observe temporal changes in the parameter importance distribution. These changes indicate that historical importance is not static even within the training process of a single task, reflecting the continuous interactions between new and historical knowledge. Static estimation fails to capture these temporal dynamics, which can lead to suboptimal knowledge fusion.

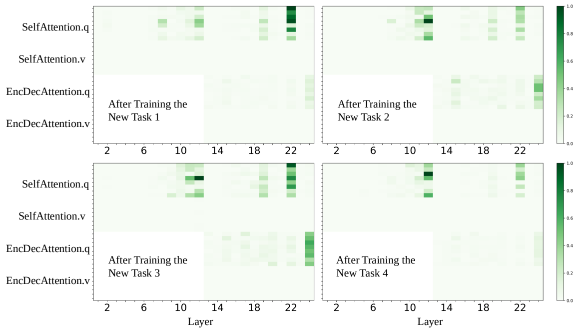


Figure 7: Visualization of the parameter importance distribution for the fixed historical task "AG News" after training on different new tasks.

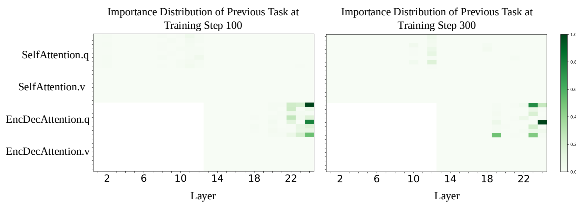


Figure 8: Visualization of the parameter importance distribution for a fixed historical task at different stages of training on a new task.

Our analysis confirm that historical parameter importance distributions are dynamic, influenced by both the characteristics of the new task being learned and the training stage. These observations provide strong empirical evidence supporting the need for dynamic importance estimation approaches. Static importance estimation fails to account for these variations, potentially causing biases and inaccuracies in knowledge fusion. In contrast, dynamic importance estimation, as proposed in our framework, addresses these issues by continuously updating importance distributions to align with the most recent model state, ensuring more effective and accurate knowledge integration.

B Additional Results

B.1 Effect on the SuperNI Benchmark

To further validate the effectiveness of Recurrent-KIF in more complex CL scenarios, we have conducted additional experiments on the SuperNI Benchmark (Wang et al., 2022a), which includes tasks like dialogue generation, information extraction, question answering, and summarization. Using T5-large as the backbone, we assessed 15 tasks following the experimental setup in (Zhao et al., 2024). The results are shown in the table 3.

Method	OP	BWT
Replay	35.4	-15.8
O-LoRA	25.9	-24.6
TaSL	41.3	-12.7
VR-MCL	40.5	-10.9
Recurrent-KIF	43.3	-8.4
MTL (Upper Bound)	50.7	-

Table 3: Overall results on the SuperNI Benchmark.

While performance decreases due to task complexity, Recurrent-KIF consistently outperforms other methods, demonstrating its robustness and ability to handle more sophisticated CL scenarios.

B.2 Effect of the Memory Size

We investigate the impact of varying memory size on the performance of LoRAReplay and Recurrent-KIF. By adjusting the memory size per task $|M|$ to 2%, 5%, 10%, 50%, the results are shown in Table 4. As expected, increasing the memory size generally improves the performance of all methods. Recurrent-KIF leverages its knowledge fusion mechanism to effectively preserve the parameters that store historical knowledge, thereby achieving better performance than LoRAReplay.

	Memory Size			
	2%	5%	10%	50%
LoRAReplay	71.2	72.4	73.8	76.1
Recurrent-KIF	77.9	78.7	79.8	80.5

Table 4: Ablation study on memory size, using T5-large as the backbone.

B.3 Effect of Different LoRA Adapters

We further investigate which components within a transformer block should incorporate LoRA. A typical transformer block consists of the query, key, and value (QKV) linear layers, the output linear layer (O) in the multi-head attention module, and the two linear layers in the feedforward network (FFN). Our analysis, presented in Table 5, shows that applying LoRA to all these linear layers yields the best overall performance. Notably, adding LoRA to the FFN layers results in better BWT performance than applying it to the multi-head attention layers.

LoRA Target Modules	OP	BWT
Attention Q V	77.9	-3.4
Attention Q K V O	77.7	-3.3
FFN	77.7	-2.5
Attention All + FFN	78.0	-3.2

Table 5: Ablation study on LoRA target modules, using T5-large as the backbone.

B.4 Sensitivity Analysis for Hyperparameters

The proposed framework incorporates three key hyperparameters, including the smoothing factor α for computing importance scores in Equations (4) and (7), the threshold δ for determining the importance of parameters, and the number of inner and outer loop training steps. Our analysis aims to assess the impact of varying these hyperparameters on our method's performance, testing on the T5-large backbone model.

As evidenced in Table 6, we determine that the optimal setting for α is 0.55. An α value too low results in a performance decline, indicating that the calculated importance scores are not accurate.

α_1,α_2	OP	BWT
0.35	77.7	-3.4
0.55	78.1	-2.8
0.85	77.9	-3.4
0.95	77.8	-2.7

Table 6: Performance comparisons of Recurrent-KIF equipped with different α.

Regarding the selection of the threshold for important parameters, Table 7 below shows the model's performance with varying thresholds δ on T5-large. It can be seen that setting a high threshold (50%) reduces model effectiveness by categorizing less significant parameters as important, which can contaminate historical knowledge and lead to forgetting. Conversely, a 1% threshold still maintains strong performance owing to our effective knowledge fusion approach, which preserves task-specific knowledge and prevents forgetting. Considering the 28-law of diminishing returns, we opted for a 20% threshold to distinguish between important and less significant parameters.

Importance Thresholds δ	OP	BWT
1%	77.4	-3.3
10%	77.8	-3.1
20%	77.9	-3.4
50%	77.6	-2.4

Table 7: Performance comparisons of Recurrent-KIF equipped with different δ.

Inner Steps (Q)	Outer Steps	OP	BWT
4	1	76.7	-4.4
8	1	77.7	-3.2
8	4	77.9	-3.4
16	4	76.8	-5.4

Table 8: Performance comparisons of Recurrent-KIF (using T5-large as the backbone) equipped with different inner and outer loop training steps.

Finally, we investigate the effect of varying the number of inner and outer loop training steps on model performance. As shown in Table 8, increasing the number of iterations for both the inner and outer loops can lead to improved performance, particularly in terms of the BWT metric. However, we observe diminishing returns beyond a certain point. Specifically, the performance gain becomes less significant when the number of iterations exceeds 8 for the inner loop and 4 for the outer loop. This suggests that our framework reaches an optimal balance between computational efficiency and performance with a modest number of iterations.

In conclusion, it is worth noting that there is a small performance difference observed when varying the hyperparameters. This suggests that the proposed Recurrent-KIF method exhibits robustness and is not highly sensitive to the choice of hyperparameters.

B.5 Time Complexity Analysis

In this section, we discuss the time complexity issues introduced by the techniques used in Recurrent-KIF. The additional time complexity can be explained qualitatively: assuming the number of training iterations for vanilla training is N', we set the total number of iterations for the inner loop to N' as well. This ensures a fair comparison with traditional methods, while also minimizing the number of iterations in our approach. In this case, the total number of iterations for our method

Training Time (Min/Epoch)	T5-large	FlanT5-XL	LLaMA2-7B	LLaMA2-13B
LoRAReplay	1.4	1.4	4.5	6.6
O-LoRA	1.4	1.4	4.5	6.7
VR-MCL	1.5	1.8	6.0	10.2
TaSL	1.4	1.4	4.6	6.7
Recurrent-KIF	1.4	1.6	5.5	9.1

Table 9: Training time comparison across backbones.

is $N' + (N'/Q)$, where N'/Q corresponds to the iterations of the outer loop. Typically, N' is 1024, and with Q set to 8, this results in an additional 12% increase in training time.

Regarding the time consumption of knowledge identification and fusion, the variables used in the knowledge identification phase are derived from the gradients produced during normal training, requiring no extra computation time, only additional space to store parameter importance information. The knowledge fusion phase involves only simple univariate calculations, as shown in Equation (9). Therefore, the overall time complexity does not increase significantly.

Quantitatively, we compare the training time of our method with LoRA Replay, as shown in Figure 9. Compared to traditional LoRA replay methods, the addition of knowledge identification and fusion does not significantly increase training time across different backbones. For instance, when using LLaMA2-13B as the backbone, adding knowledge identification and fusion results in a 1.37x increase in training time compared to the original setup. However, for smaller models like T5-large and T5-xl, the training time remains relatively consistent, with no significant impact observed from the inclusion of the reasoning components.

B.6 Effect of the Different Importance Metric in Knowledge Identification

We compare two alternative importance scoring approaches with Eq. (3): (i) using absolute gradients (Michel et al., 2019), $\left|\nabla_{w_{ij}}\mathcal{L}\right|$, instead of the gradient-weight product; and (ii) removing exponential moving average, relying only on importance scores computed from a single batch.

As shown in Table 10, our method with exponential smoothing outperforms the alternatives, with performance drops of up to 2.0% and 1.3% without smoothing. Similarly, using absolute gradients leads to lower performance compared to the gradient-weight product, underscoring the effectiveness of our approach in enhancing knowledge identification and model performance.

Method	OP	BWT
$I(\cdot) = \left\|\nabla_{w_{ij}}\mathcal{L}\right\|$	74.5	-5.2
$I(\cdot) = \left\|w_{ij}\nabla_{w_{ij}}\mathcal{L}\right\|$	75.9	-4.7
Recurrent-KIF (ours)	**77.9**	**-3.4**

Table 10: Ablation study. Evaluating the impact of different importance metrics on knowledge identification.

C Dataset Statistics

Table 11 show details of the datasets we used for our experiments, along with their evaluation metrics (Wang et al., 2023a; Feng et al., 2024c; Xu et al., 2023). For the Long Sequence benchmark, this includes five tasks from the standard CL benchmark (AG News, Amazon reviews, Yelp reviews, DBpedia and Yahoo Answers), four from GLUE benchmark (MNLI, QQP, RTE, SST2), five from SuperGLUE benchmark (WiC, CB, COPA, MultiRC, BoolQ), and the IMDB movie reviews dataset. We report 6 different task orders used for our experiments in Table 12. Table 13 shows prompts for different tasks. NLI denotes natural language inference (Lu et al., 2021b), including MNLI, RTE and CB. SC denotes sentiment analysis, including Amazon, Yelp, SST-2 and IMDB. TC denotes topic classification, including AG News, Dbpedia and Yahoo (Xu et al., 2025; Chen et al., 2024).

D Implementation Details

Experiments are implemented using PyTorch and the Transformer library, running on a single NVIDIA A100 GPU with 80GB memory. The following hyperparameters are used:

- **T5-large (770M)** and **FLAN-T5-XL (3B)**: Learning rate of 3e-4 for both loops, inner and outer batch sizes of 8, max input length 512, max target length 128, and 10 epochs. LoRA settings: $r = 8$, $\alpha = 32$, dropout = 0.05, targeting modules [q,v]. Testing: max new tokens = 128.
- **LLaMA-2 (7B)** and **LLaMA-2 (13B)**: Learning rate of 3e-4 for both loops, inner and outer batch sizes of 64, cutoff length 512, and 10 epochs. LoRA settings: $r = 8$, $\alpha = 32$, dropout = 0.05, targeting modules [q_proj,v_proj]. Testing: temperature = 0.02, top_p = 0, top_k = 1, num_beams = 1, max new tokens = 128.

It is worth noting that we used the same hyperparameters across different datasets and backbones, demonstrating the generalizability of our method without requiring extensive hyperparameter tuning for each specific setting.

E Algorithm

In this section, we provide the detailed implementation of Recurrent-KIF algorithm (see Algorithm 1).

Dataset name	Category	Task	Domain	Metric
1. Yelp	CL Benchmark	sentiment analysis	Yelp reviews	accuracy
2. Amazon	CL Benchmark	sentiment analysis	Amazon reviews	accuracy
3. DBpedia	CL Benchmark	topic classification	Wikipedia	accuracy
4. Yahoo	CL Benchmark	topic classification	Yahoo Q&A	accuracy
5. AG News	CL Benchmark	topic classification	news	accuracy
6. MNLI	GLUE	natural language inference	various	accuracy
7. QQP	GLUE	paragraph detection	Quora	accuracy
8. RTE	GLUE	natural language inference	news, Wikipedia	accuracy
9. SST-2	GLUE	sentiment analysis	movie reviews	accuracy
10. WiC	SuperGLUE	word sense disambiguation	lexical databases	accuracy
11. CB	SuperGLUE	natural language inference	various	accuracy
12. COPA	SuperGLUE	question and answering	blogs, encyclopedia	accuracy
13. BoolQA	SuperGLUE	boolean question and answering	Wikipedia	accuracy
14. MultiRC	SuperGLUE	question and answering	various	accuracy
15. IMDB	SuperGLUE	sentiment analysis	movie reviews	accuracy

Table 11: The details of 15 datasets used in our CL experiments. First five tasks correspond to the standard CL benchmark, all other tasks are used in long-sequence experiments.

Order	Model	Task Sequence
1	T5, LLaMA	dbpedia → amazon → yahoo → ag
2	T5, LLaMA	dbpedia → amazon → ag → yahoo
3	T5, LLaMA	yahoo → amazon → ag → dbpedia
4	T5	mnli → cb → wic → copa → qqp → boolqa → rte → imdb → yelp → amazon → sst-2 → dbpedia → ag → multirc → yahoo
5	T5	multirc → boolqa → wic → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yelp → amazon → yahoo
6	T5	yelp → amazon → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yahoo → multirc → boolqa → wic

Table 12: Six different orders of task sequences used for continual learning experiments. Orders 1-3 correspond to the standard CL becnhmark adopted by prior works. Orders 4-6 are long-sequence orders spanning 15 tasks, following (Razdaibiedina et al., 2023).

Task	Prompts
NLI	What is the logical relationship between the "sentence 1" and the "sentence 2"? Choose one from the option.
QQP	Whether the "first sentence" and the "second sentence" have the same meaning? Choose one from the option.
SC	What is the sentiment of the following paragraph? Choose one from the option.
TC	What is the topic of the following paragraph? Choose one from the option.
BoolQA	According to the following passage, is the question true or false? Choose one from the option.
MultiRC	According to the following passage and question, is the candidate answer true or false? Choose one from the option.
WiC	Given a word and two sentences, whether the word is used with the same sense in both sentence? Choose one from the option.

Table 13: Instructions for different tasks.

Algorithm 1 The Algorithm of the proposed Recurrent-KIF

Input: Current task dataset $\mathcal{D}_k$, memory buffer $\mathcal{M}_{<k}$, model weights θ, initial inner-loop learning rate β_{in}, outer-loop learning rate β_{out}, number of inner-loop steps Q, number of outer-loop steps S, hyperparameters α_1, α_2, total number of fusion steps N.

 # training iterations.
for $b = 1, \ldots, N$ **do**
    sample training data ϵ_k^q $(q = 1, \ldots, Q)$ from $\mathcal{D}_k$
    $\theta_{b(0)} = \theta_b$
    # inner loop.
    for $q = 1$ **to** Q **do**
        obtain batch samples ϵ_k^q
        $\theta_{b(q)} = \theta_{b(q-1)} - \beta_{in} \nabla \mathcal{L}\left(\theta_{b(q-1)}\right)$
        # knowledge identification.
        compute the importance $\bar{I}\left(w_{ij}\right)$ via Eq. (3);
        update $I_{b(q)}$ via Eq. (4)
    end for
    calculate $\tau_b^{\text{in}} = \theta_{b(Q)} - \theta_{b(0)}$ and obtain I_b^{in}
    # outer loop.
    $\theta_{b(M)}^0 = \theta_{b(Q)}$
    for $s = 1$ **to** S **do**
        sample memory data ϕ_b^s from $\mathcal{M}_{<k}$
        $\theta_{b(M)}^s = \theta_{b(M)}^{s-1} - \beta_{out} \nabla \mathcal{L}\left(\theta_{b(M)}^{s-1}\right)$
    end for
    calculate $\tau_b^{\text{out}} = \theta_{b(M)}^S - \theta_{b(Q)}$
    calculate I_b^{out} via Eq. (7)
    # knowledge fusion.
    obtain m_b^{in}, m_b^{out} via Eq. (8)
    update $\theta_{b+1} = \theta_b + (m_b^{\text{in}} \odot \tau_b^{\text{in}} + m_b^{\text{out}} \odot \tau_b^{\text{out}})$
end for