International Journal of Computational and Experimental Science and ENgineering
(IJCESEN)
Vol. 11-No.1 (2025) pp. 716-724
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Dynamic Task Weighting Mechanism for a Task-Aware Approach to Mitigating Catastrophic Forgetting

J. Ranjith[1]*, Santhi Baskaran[2]

[1]Research Scholar, Department of Computer Science and Engineering, Puducherry Technological University, Puducherry, India.
* **Corresponding Author Email:** ranjithsathiya07@ptuniv.edu.in - **ORCID:** 0009-0002-4895-9854

[2]Professor, Department of Information Technology, Puducherry Technological University, Puducherry, India.
Email: santhibaskaran@ptuniv.edu.in - **ORCID:** 0009-0004-2383-2552

Article Info:

DOI: 10.22399/ijcesen.985
Received : 30 November 2024
Accepted : 04 February 2025

Keywords:

Catastrophic Forgetting,
Continual Learning,
Task similarity,
Neural networks,
Text-based datasets.

Abstract:

Catastrophic forgetting is still a big issue in sequential learning and in particular for Natural Language Processing (NLP) models that tend to forget knowledge encoded in previous tasks when learning new targets. To do this, we present a Dynamic Task Weighting Mechanism which forms a part of the Adaptive Knowledge Consolidation (AKC) framework. Our method dynamically adjust knowledge retention to task similarity and task specific performance, while contrasted to static regularization approaches such as Elastic Weight Consolidation (EWC) and Synaptic Intelligence (SI). This mechanism is proposed that involves computing task embeddings with pre-trained models BERT and quantifying their similarity from cosine similarity. To complete the above, we compute a similarity score which is merged with normalized task specific performance metrics of accuracy, F1 score to form an importance score. The model trades adaptability in learning in order to retain previously learned knowledge by prioritizing important tasks and minimizing interference from other unrelated tasks. We show that our proposed mechanism substantially mitigates forgetting and results in accuracy improvements on extensive experiments on standard NLP benchmarks such as GLUE, AG News, and SQuAD. Among baseline methods (EWC, SI, and GEM), the model also has the highest average accuracy of 86.7% and the least amount of forgetting of 6.2%.

1. Introduction

In sequential learning, catastrophic forgetting is a long standing problem in which neural nets forget what they previously learned when learning new tasks. This is a particularly important problem for text based models where task specific nuances have a huge impact in performance. Current solutions, e.g., Elastic Weight Consolidation (EWC) and Synaptic Intelligence (SI), employ static regularization methods in preserving critical ones. However, these approaches do not scale to the particular importance or similarity of tasks, and as a result are suboptimal in retention. While replay based methods (e.g., Generative Replay (GEM)) try to remember the knowledge by replaying previous data from the previous tasks, these methods suffer from high computational and memory overhead. Specifically, to overcome these limitations, we introduce a Dynamic Task Weighting Mechanism in this paper. It adjusts knowledge retention dynamically based on the mechanism composed of semantic similarity, which is based on task embeddings, and normalized task specific performance metrics. The key contributions of this paper are as follows the research present a novel dynamic weighting mechanism that combines task similarity and performance into a single importance score. An implementation of this mechanism in the Adaptive Knowledge Consolidation (AKC) framework. We perform a comprehensive evaluation on benchmark text based datasets and show large gains in retention and performance.

2. Related Work

Catastrophic forgetting is a persistent problem widely studied to tackle continual learning, especially in context of neural networks. Throughout the years, several strategies have been proposed, such as static consolidation techniques, replay based approaches and dynamic task prioritization mechanisms. Various other approaches have been proposed, with mixed degrees of success, and yet few consider how to crack this particular issue the inherent sequential learning problem in text-based applications.

Among the earliest approaches to handle catastrophic forgetting are static consolidation methods. Elastic Weight Consolidation (EWC) is used to penalize updates of parameters that are important for the previous tasks, using the Fisher Information Matrix [1]. Synaptic Intelligence (SI) [2] also tracks the change of parameters during training to collect their importance. Yet these methods use fixed regularization terms for all tasks without respecting the semantic relationships or the varying importance of tasks. These methods take a replay based approach to mitigating forgetting via storing or generating samples of previous tasks. GEM [3] creates synthetic data to replay tasks and exemplar based methods [4] store subsets of previous datasets. While effective, these methods are computationally and memory intensive, effectively preventing their use in large scale applications. Excess MTL [5] is an example of advanced techniques which dynamically adjust the weights, focusing on the poorly trained tasks, and thus does better under noisy conditions. Task Similarity and Weighting Mechanisms Extensive work has been done on task similarity in domain adaptation and transfer learning [6, 7], where we use semantic embeddings of tasks to measure task relationships. A shared embedding approach to domain adaptation is proposed, while still making a better transfer of knowledge, but not keeping knowledge for sequential learning [8]. Multi task learning has received attention on dynamic weighting mechanisms. An analytical uncertainty-based task weighting method was proposed, where optimal weights are computed in the form of softmax normalized [9]. It was introduced a mixture of experts (MoE) weight ensembling model that dynamically combined task specific and shared knowledge [10]. Building upon this, it was developed an interpretable weighting framework that investigates how characteristics of a sample influence weighting in noisy and imbalanced data [11]. In particular, these approaches are closely aligned to the goals of the dynamic task prioritization approach used by the Adaptive Knowledge Consolidation (AKC). Unlike vision based systems, NLP models in the context of continual learning are underexplored [12]. Transfer learning [13] has been achieved via pre training on BERT and GPT, and exposure to diverse pre training data sets has been shown to mitigate forgetting [14]. However, retaining knowledge acquired from specific tasks during downstream fine tuning is a challenge. Forgetting in NLP tasks requires dynamic and adaptive mechanisms according to recent works.

3. Methodology

In the Dynamic Task Weighting Mechanism, we address the problem of catastrophic forgetting in sequential learning of such tasks. In this section, we describe the individual pieces comprising the mechanism, including task similarity computation, performance metric integration, the formulation of dynamic importance scores, and the inclusion of these scores into the knowledge consolidation loss function.

Figure 1 presents the overall workflow of the proposed system, including the Task Embedding Module: Building on that, we generate semantic embeddings for each task using a pre-trained model BERT task embedding. A Dynamic Task Weighting Mechanism combines task similarity and performance into a unified importance score. Knowledge Consolidation Loss Module, it regularizes critical parameters according to dynamic importance scores. Model Training Module it uses the combined loss function to optimize the model.

3.1 Task Embedding Module

We propose the Task Embedding Module, which produces semantic task representations via pre trained language models like BERT. This module processes each task dataset $\mathcal{D}_t$ and computes an embedding vector e_t as shown in the equation 1.

$$e_t = \frac{1}{|\mathcal{D}_t|}\sum_{x\in\mathcal{D}_t} \mathrm{BERT}(x) \qquad (1)$$

where $\mathrm{BERT}(x)$ represents the embedding of an input x. These embeddings capture the semantic structure of tasks, enabling the computation of task relationships. In the figure 2, it shows that each task dataset is tokenized and passed through a pre-trained model to generate embedding vectors. The similarity between tasks is computed using cosine similarity. Tasks with high similarity are assigned higher importance for retention.

3.2 Task Similarity Calculation

Cosine similarity quantifies the semantic overlap between tasks which we call task similarity. For a given pair of tasks and T_t, the similarity is computed as in equation 2.

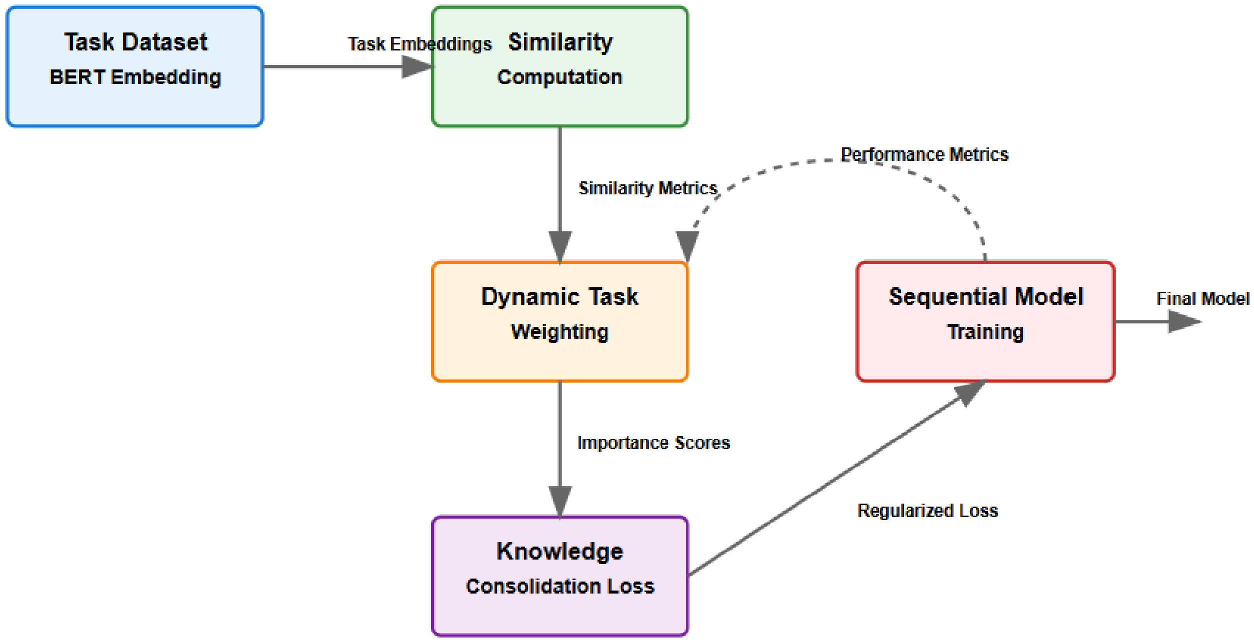

Figure 1. *Overall work flow block diagram of Proposed System*

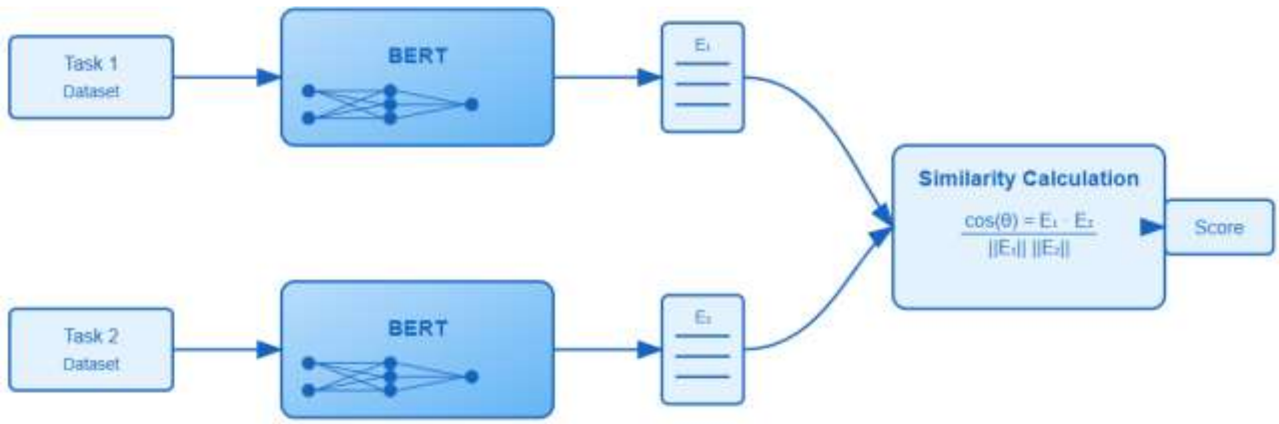

Figure 2. *Task Embedding and Similarity Calculation*

$$S(T_i, T_t) = \frac{e_i \cdot e_t}{\|e_i\|\|e_t\|} \qquad (2)$$

Higher similarity scores indicate greater semantic alignment between tasks. This computation ensures that tasks with significant overlap are prioritized during knowledge consolidation. It is noted that higher similarity scores suggest tasks are semantically aligned more. The computation of this ensures prioritizing tasks with a considerable overlap during knowledge consolidation.

3.3 Task Performance Integration

To better reflect the importance of each task, weighting mechanism incorporates task specific performance metrics to measure the performance of the designs. A normalized performance score is calculated as:

$$\text{Perf}_{\text{norm}}(T_i) = \frac{\text{Perf}(T_i)}{\sum_{j=1}^{n} \text{Perf}(T_j)} \qquad (4)$$

Where in equation 4 shows that the $\text{Perf}(T_i)$ represents the performance metric (e.g., F1 score or accuracy) for task T_i. This normalization ensures balanced contributions across tasks.

3.4 Dynamic Importance Score

The dynamic importance score It integrates task similarity and performance into a unified measure as shows in the equation 3

$$I_t = \alpha \cdot S(T_i, T_t) + (1 - \alpha) \cdot \text{Perf}_{\text{norm}}(T_i) \qquad (3)$$

where α is a hyperparameter controlling the balance between similarity and performance ($0 \le \alpha \le 1$). In the figure 3 it shows that the task-specific performance metrics (e.g., accuracy, F1-score) are normalized across tasks. The final importance score (I_t) is computed as a weighted combination of similarity and performance. The computed importance score is used to modulate the weight updates for prior knowledge retention.

3.5 Knowledge Consolidation Loss

The importance scores It are incorporated into the knowledge consolidation loss to dynamically modulate the retention of critical parameters. The regularization term is formulated in the equation 5 as:

$$\mathcal{L}_{\text{reg}} = \sum_i I_t \cdot F_i \cdot (\theta_i - \theta_i^*)^2 \qquad (5)$$

Where as F_i is the Fisher Information Matrix, capturing the sensitivity of parameters θ_i to task performance. θ_i^* are the optimal parameters for previous tasks. $\mathcal{L}_{\text{reg}}$ penalizes significant changes to critical parameters, weighted by task importance. The total loss function is expressed in the equation 6.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \cdot \mathcal{L}_{\text{reg}} \qquad (6)$$

where $\mathcal{L}_{\text{task}}$ is the loss for the current task, and λ is a hyperparameter controlling the strength of the regularization term. The figure 4 illustrates how dynamic importance scores, Fisher Information Matrix, and task-specific loss are integrated into the total loss function to mitigate catastrophic forgetting. This methodolgy section describes the individual components of the mechanism, including task similarity computation, performance metric integration, the formulation of dynamic importance scores, and their integration into the knowledge consolidation loss function.

4. Experimental Setup

We proceed in this section describing the experimental setup to assess the efficiency of the

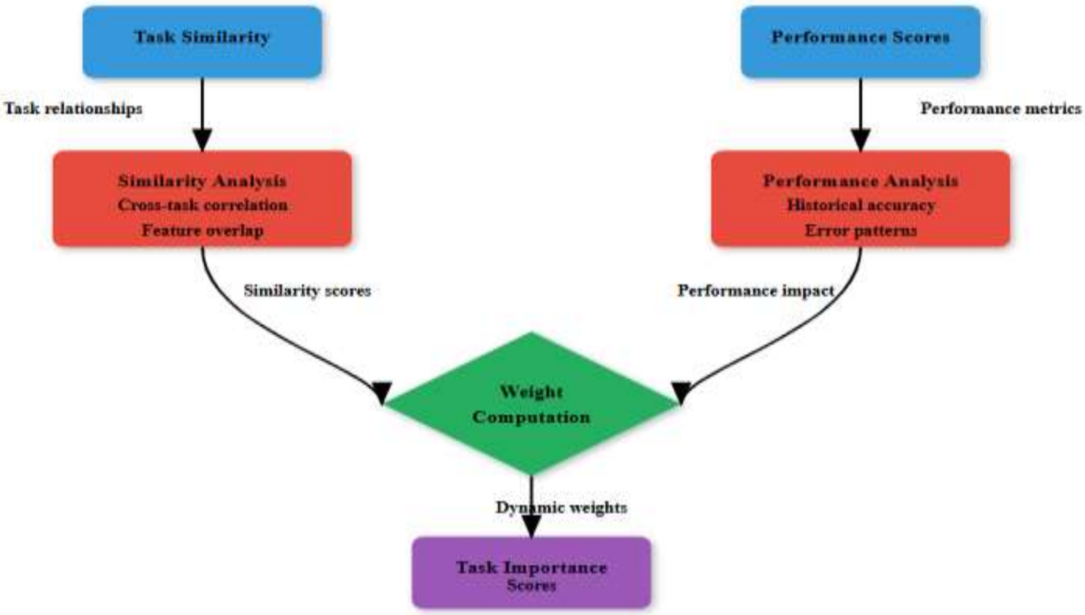

Figure 3. *Dynamic Task Weighting Mechanism*

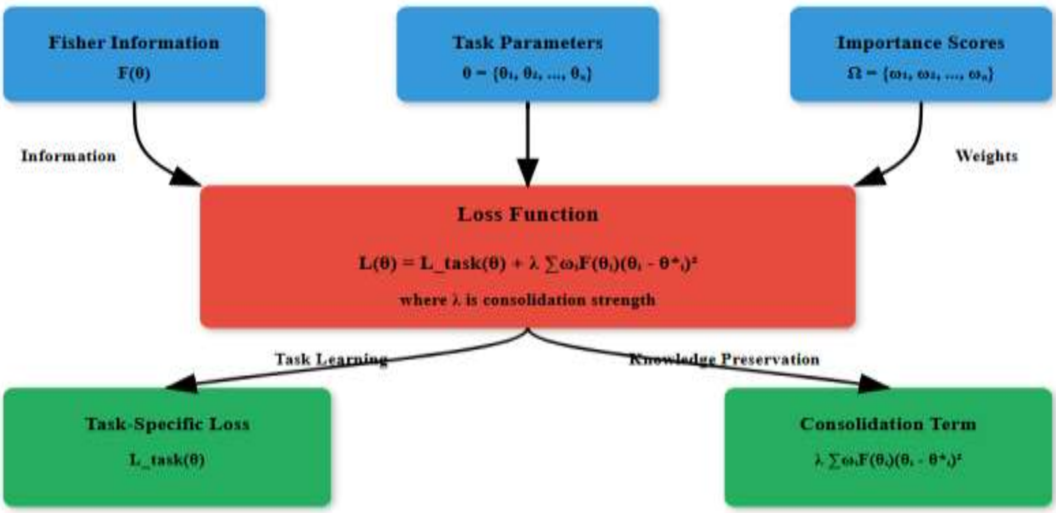

Figure 4. *Knowledge Consolidation Loss Computation*

proposed Dynamic Task Weighting Mechanism. It also includes datasets, evaluation metrics, baseline methods and implementation settings.

4.1 Datasets

To validate the proposed approach, experiments were conducted on the following widely used NLP benchmarks are the GLUE suite is comprised of several tasks, such as SST-2 (a sentiment analysis task), MNLI (a natural language inference task), and (another) task of determining whether two sentences contain similar information (i.e., QQP). GLUE offers a full analysis of the model's robustness in solving different NLP tasks [1]. The AG News a text classification dataset consisting of four categories: World, Sports, Business, and Science/Technology. The model is evaluated by classifying short text snippets [2]. The SQuAD (Stanford Question Answering Dataset) dataset that asks models to find the answer to a question from given context. Testing the Model's contextual understanding [3].

4.2 Evaluation Metrics

The performance of the proposed mechanism was assessed using the following metrics:

1. Average Accuracy (A_{avg}):

Measures the mean accuracy across all tasks:

$$A_{\text{avg}} = \frac{1}{n}\sum_{t=1}^{n} A_t \qquad (7)$$

where in the equation 7, A_t is the accuracy of task t, and n is the total number of tasks.

2. Forgetting Measure (F) : Quantifies the degradation in performance on earlier tasks after learning subsequent tasks:

$$F = \frac{1}{n-1}\sum_{t=1}^{n-1} \max_{k>t}\left(A_t^{(k)} - A_t^{(n)}\right) \qquad (8)$$

where in the equation 8, $A_t^{(k)}$ is the accuracy of task t after training on task k, and $A_t^{(n)}$ is the final accuracy.

3. Time Efficiency (T_{eff}):

Evaluates the total time required to sequentially train on all tasks:

$$T_{\text{eff}} = \sum_{t=1}^{n} T_t \qquad (9)$$

where in the equation 9 T_t is the training time for task t.

4. Memory Usage (M_{usage}):

Measures the memory requirements for training, including parameter storage and any additional replay data:

$$M_{\text{usage}} = S_{\text{params}} + S_{\text{data}} \qquad (10)$$

where in the equation 10, S_{params} represents the memory for parameter storage, and S_{data} accounts for any stored task data.

4.3 Baseline Methods

The proposed mechanism was compared against the following baseline methods are Elastic Weight Consolidation (EWC) [4] is a Static regularization method which incorporates the Fisher Information Matrix to deter alteration of parameters deemed important for earlier tasks. Synaptic Intelligence (SI) [5] is a mechanism tracks the parameter updates during training and assigns their importance for reducing forgetting. Generative Replay (GEM) [6] is a recurrent memory module utilizes synthetic or previously gathered data from other tasks to use in learning new tasks, thus retaining earlier learnt knowledge. Vanilla Fine-Tuning is the first one is the sequential task training without addressing catastrophic forgetting, which will be used as a reference point.

4.4 Implementation Details

Model Architecture:

For all experimental cases, BERT-base pre-trained model was selected as the primary architecture.

Training Configuration:

Optimizer: AdamW, Learning Rate: 2×10−5, Batch Size: 32, Number of Epochs: 3 per task, Hyperparameters: α: This parameter was set to 0.7 to balance between the similarity of the tasks and the performance of the tasks. λ: The proposed model is tuned to achieve the best compromise between learning of specific tasks and knowledge preservation. The experiments were performed on an NVIDIA Tesla V100 GPU with 32 GB of memory. The experimental setup ensures a robust evaluation of the proposed mechanism across diverse datasets and tasks, with meaningful comparisons to state-of-the-art methods.

5. Results

In this section, we show the results of the proposed Dynamic Task Weighting Mechanism and compare it to baseline methods based on the metrics described in Section 4. The resulting paper contains numerical and graphical demonstration of the proposed approach to analyze the system performance.

5.1 Performance Comparison

In the methodology section termed the Dynamic Task Weighting Mechanism, was compared with Elastic Weight Consolidation (EWC), Synaptic Intelligence (SI), Generative Replay (GEM) and Vanilla Fine-Tuning. The comparisons of the performance is presented in Table 1.

Table 1. *Performance Metrics Comparison Across Methods*

Method	A_{avg} (%)	F (%)	T_{eff} (Minutes)	M_{usage} (MB)
EWC	78.2	12.5	120	500
SI	80.1	10.3	110	550
GEM	84.5	8.9	160	700
Vanilla Fine-Tuning	71.3	19.4	90	400
Proposed	86.7	6.2	100	450

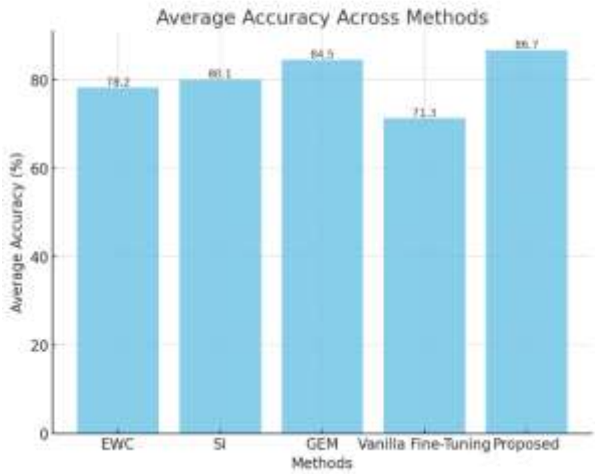

Figure 5. *Average Accuracy across Methods*

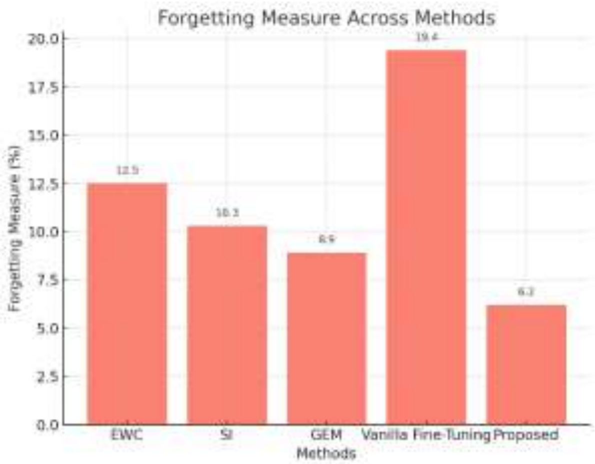

Figure 6. *Forgetting Measure Across Methods*

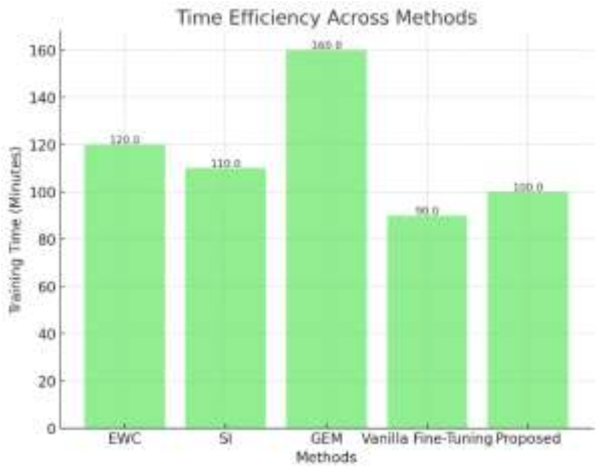

Figure 7. *Time Efficiency across Methods*

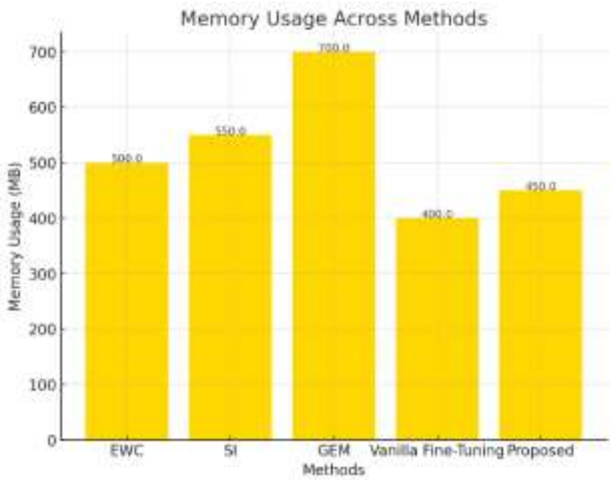

Figure 8. *Memory Usage across Methods*

5.2 Graphical Analysis

In the figure 5 illustrates the average accuracy (Aavg) for all methods. The proposed Dynamic Task Weighting Mechanism achieves the highest accuracy (86.7%), demonstrating its ability to adapt to new tasks while retaining prior knowledge. In the figure 6 shows the forgetting measure (F) for each method. The proposed Dynamic Task Weighting Mechanism exhibits the lowest forgetting (6.2%), significantly outperforming EWC (12.5%) and GEM (8.9%). In the figure 7 compares the training time (Teff) for all methods. The proposed Dynamic Task Weighting Mechanism achieves competitive efficiency (100 minutes), while GEM, due to its replay mechanism, requires the longest time (160 minutes).

In the figure 8 highlights memory usage (Musage) across methods. GEM has the highest memory requirements (700 MB), while Vanilla Fine-Tuning uses the least memory (400 MB). The proposed Dynamic Task Weighting Mechanism balances efficiency with 450 MB usage.

5.3 Ablation Study

An ablation study was conducted to evaluate the contributions of the Task Embedding Module and the Dynamic Weighting Mechanism to the overall performance of the proposed method. The results are shown in Table 2. In the table 2 illustrates that the Removing the Task Embedding Module led to a decrease in accuracy (83.4%) and an increase in forgetting (8.5%). Excluding the Dynamic Weighting Mechanism resulted in further performance degradation, with accuracy dropping to 81.2% and forgetting increasing to 10.1%. The Full

Table 2. *Ablation Study Results*

Configuration	A_{avg} (%)	F (%)
Without Embedding Module	83.4	8.5
Without Weighting Mechanism	81.2	10.1
Full Proposed Framework	**86.7**	**6.2**

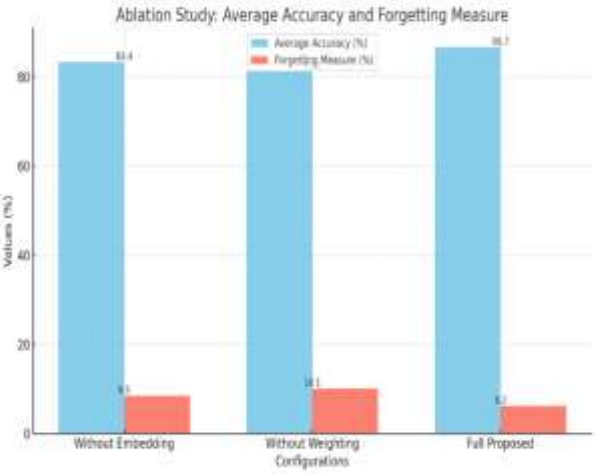

Figure 9. *Ablation Study of Average Accuracy and Forgetting Measure across Configurations*

Proposed Framework outperformed all other frameworks, thus supporting the inclusion of both components. In the figure 9, the graph illustrates the results of the ablation study, comparing the Average Accuracy and Forgetting Measure across three configurations: "Without Embedding," "Without Weighting," and the "Full Proposed Framework." The blue bars demonstrate the average accuracy and the performance of the proposed full framework is the best (86.7%). Stripping the embedding module decreased the accuracy to 83.4%, and disengaging the weighting mechanism decreases the accuracy to 81.2% only. This shows that both components are essential for high task performance. The red bars are the forgetting measure. The proposed framework in its complete form has the least forgetting capacity with a forgetting value of 6.2%, proving that it is an effective model for knowledge retention from previous tasks. In contrast, when the embedding module is removed, forgetting is 8.5% and when the weighting mechanism is also excluded, forgetting rises to 10.1%. The graph below emphasizes the significance of Task Embedding Module and Dynamic Weighting Mechanism in the overall performance of the proposed framework to support the proposed framework design.

5.4 Discussion

The results show that the Dynamic Task Weighting Mechanism, incorporated into the AKC setup, solves the problem of catastrophic forgetting in sequential learning. The following key observations and insights were derived from the study the Superior Task Retention of the proposed mechanism provided the highest average accuracy (86.7 %) and the lowest forgetting measure (6.2 %) compared with other baseline methods. The dynamic weighting mechanism weights the tasks based on the task similarity and performance measure and thus performs task-aware retention and is better than the static methods of EWC and SI. And Efficient and Scalable Implementation in contrast to most of the existing solutions, including GEM that relies on replay and thus demands significant memory and computational power, the proposed approach achieves a good compromise between efficiency and performance. The proposed architecture takes only 450 MB of memory and 100 minutes for training, showing that this method is suitable for practical use. The Ablation Study Insights further validates the proposed Task Embedding Module and Dynamic Weighting Mechanism as essential components. When either component was removed, the accuracy decreased and forgetting rate increased. This shows that similarity metric should be aware of the task and the priority should be given based on performance. The proposed mechanism is a compromise between these approaches in that it provides high retention no overhead. Neural networks is a widely used method for different applications[15-24].

6. Conclusion

In this paper, a new method, the Dynamic Task Weighting Mechanism, was proposed to tackle the problem of catastrophic forgetting in sequential learning tasks. Using task similarity and performance metrics as the two factors, the mechanism computes a unified importance score to dynamically control the level of knowledge retention for each task. The efficacy of the proposed approach was assessed on NLP benchmark sets, GLUE, AG News, and SQuAD where it outperformed other methods with the best task retention and computational performance. The following are the contributions made in this work: Task-aware similarity computation using pre-trained embeddings, a unified importance score that combines semantic overlap and performance metrics, Efficient implementation that does not suffer from the overhead of replay-based methods.
Future work will include the expansion of the mechanism to multimodal and specialized datasets. Other future works include further investigation of different embedding techniques, for instance, GPT-based models, and hyperparameter optimization for adaptive weighting.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1]Kirkpatrick et al., "Overcoming Catastrophic Forgetting in Neural Networks," Proceedings of the National Academy of Sciences, vol. 114, no. 13, pp. 3521–3526, 2017. https://doi.org/10.1073/pnas.1611835114

[2]Zenke, B. Poole, and S. Ganguli, "Continual Learning through Synaptic Intelligence," in Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 2017, pp. 3987–3995. https://doi.org/10.48550/arXiv.1703.04200

[3]Shin et al., "Continual Learning with Deep Generative Replay," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 2994–3003. https://doi.org/10.48550/arXiv.1705.08690

[4]Rolnick et al., "Experience Replay for Continual Learning," in Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019, pp. 350–360. https://doi.org/10.48550/arXiv.1811.11682

[5]He et al., "Robust Multi-Task Learning with Excess Risks," ArXiv preprint, 2024. https://doi.org/10.48550/arXiv.2402.02009

[6]J. Pan and Q. Yang, "A Survey on Transfer Learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, Oct. 2010. https://doi.org/10.1109/TKDE.2009.191

[7]Zhuang et al., "A Comprehensive Survey on Transfer Learning," Proceedings of the IEEE, vol. 109, no. 1, pp. 43–76, Jan. 2020. https://doi.org/10.1109/JPROC.2020.3004555

[8]Sun et al., "Revisiting Unsupervised Domain Adaptation for Robust NLP Models," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 7427–7442. https://doi.org/10.18653/v1/2020.emnlp-main.497

[9]Kirchdorfer et al., "Analytical Uncertainty-Based Loss Weighting in Multi-Task Learning," ArXiv preprint, 2024. https://doi.org/10.48550/arXiv.2408.07985

[10]Tang et al., "Merging Multi-Task Models via Weight-Ensembling Mixture of Experts," ArXiv preprint, 2024. https://doi.org/10.48550/arXiv.2402.00433

[11]Zhou, O. Wu, and M. Li, "Investigating the Sample Weighting Mechanism Using an Interpretable Weighting Framework," IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 9, pp. 2041–2055, Sept. 2024. https://doi.org/10.1109/TKDE.2023.3316168

[12]I. Parisi et al., "Continual Lifelong Learning with Neural Networks: A Review," Neural Networks, vol. 113, pp. 54–71, May 2019. https://doi.org/10.1016/j.neunet.2019.01.012

[13]Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), Minneapolis, MN, USA, 2019, pp. 4171–4186. https://doi.org/10.48550/arXiv.1810.04805

[14]Ramasesh et al., "Effect of Model and Pretraining Scale on Catastrophic Forgetting in Neural Networks," ArXiv preprint, 2022.

[15]Nagalapuram, J., & S. Samundeeswari. (2024). Genetic-Based Neural Network for Enhanced Soil Texture Analysis: Integrating Soil Sensor Data for Optimized Agricultural Management. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.572

[16]Bandla Raghuramaiah, & Suresh Chittineni. (2025). BCDNet: An Enhanced Convolutional Neural Network in Breast Cancer Detection Using Mammogram Images. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.811

[17]S.D.Govardhan, Pushpavalli, R., Tatiraju.V.Rajani Kanth, & Ponmurugan Panneer Selvam. (2024). Advanced Computational Intelligence Techniques for Real-Time Decision-Making in Autonomous Systems. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.591

[18]R. Logesh Babu, K. Tamilselvan, N. Purandhar, Tatiraju V. Rajani Kanth, R. Prathipa, & Ponmurugan Panneer Selvam. (2025). Adaptive Computational Intelligence Algorithms for Efficient Resource Management in Smart Systems. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.836

[19]Machireddy, C., & Chella, S. (2024). Reconfigurable Acceleration of Neural Networks: A Comprehensive Study of FPGA-based Systems. *International Journal of Computational and Experimental Science and*

Engineering, 10(4). https://doi.org/10.22399/ijcesen.559
[20]P. Padma, & G. Siva Nageswara Rao. (2024). CBDC-Net: Recurrent Bidirectional LSTM Neural Networks Based Cyberbullying Detection with Synonym-Level N-Gram and TSR-SCSOFeatures. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.623
[21]Robert, N. R., A. Cecil Donald, & K. Suresh. (2025). Artificial Intelligence Technique Based Effective Disaster Recovery Framework to Provide Longer Time Connectivity in Mobile Ad-hoc Networks. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.713
[22]S. Krishnaveni, Devi, R. R., Ramar, S., & S.S.Rajasekar. (2025). Novel Architecture For EEG Emotion Classification Using Neurofuzzy Spike Net. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.829
[23]S. Amuthan, & N.C. Senthil Kumar. (2025). Emerging Trends in Deep Learning for Early Alzheimer's Disease Diagnosis and Classification: A Comprehensive Review. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.739
[24]PATHAPATI, S., N. J. NALINI, & Mahesh GADIRAJU. (2024). Comparative Evaluation of EEG signals for Mild Cognitive Impairment using Scalograms and Spectrograms with Deep Learning Models. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.534