LoRA Once: Continual Sentence Task Learning via Shared LoRA Adapter and CLS Alignment

Anonymous ACL submission

Abstract

Continual learning in natural language processing is challenged by catastrophic forgetting, where knowledge of prior tasks is lost as new tasks are learned. This work introduces a representation alignment strategy using CLS embeddings from transformer models to mitigate forgetting in task-incremental scenarios. Three CLS alignment loss functions are evaluated, with diagonal KL divergence (KL-D) providing the most stable feature preservation. The alignment weight λ balances retention and acquisition of knowledge. The final-layer CLS embedding (CLS-F) achieves an effective trade-off between accuracy, forgetting, and computational efficiency. On the task-incremental GLUE benchmark, CLS-F outperforms regularization-based methods EWC, SI, MAS, and RWalk, reducing forgetting by 32%, 31%, 57%, and 59%, respectively, while reaching retention comparable to rehearsal-based approaches (FM reduction 12%) without storing raw data. CLS-F updates only 0.147M trainable parameters, maintaining low memory usage and inference latency, offering a lightweight and scalable solution for continual learning in transformer-based NLP models.

1 Introduction

Pretrained language models such as BERT (Le Breton et al., 2025; Devlin et al., 2019) and its lightweight variant DistilBERT (Sanh et al., 2019) have advanced sentence-level natural language understanding by enabling effective transfer learning from large-scale corpora. These models capture bidirectional context, generate transferable sentence representations, and generalize across diverse tasks with minimal supervision, serving as robust backbones for applications including sentiment analysis (Dhawan et al., 2025; Sun et al., 2019), natural language inference (Williams et al., 2018; Negru et al., 2025), and paraphrase detection (Wang et al., 2018; Gupta and Kumar, 2025).

Adapting pretrained models to new tasks sequentially without revisiting previous data is challenged by catastrophic forgetting (Kirkpatrick et al., 2017a; Wang et al., 2022a; Lopez-Paz and Ranzato, 2017), where knowledge acquired in earlier tasks deteriorates as new tasks are learned. Task order strongly affects performance, and continual learning (De Lange et al., 2022; Feng et al., 2025) enables incremental knowledge accumulation without retraining from scratch or maintaining separate models for each task, which is often impractical due to memory and computation constraints.

Continual learning strategies include regularization-based methods (Aljundi et al., 2018b; Zenke et al., 2017a; Wang et al., 2022b) that penalize changes to important parameters, rehearsal-based methods (Buzzega et al., 2020; Sun et al., 2021; Wang et al., 2025) that store and replay samples, and architectural approaches introducing task-specific modules. Adapter-based methods, including Low-Rank Adaptation (LoRA) (Hu et al., 2022, 2024; Zhang et al., 2024), allow selective training of small modules while freezing the backbone. Although parameter-efficient, modular strategies face scalability challenges due to per-task adapter growth and selection mechanisms during inference.

In this study, CLS-based representation alignment is applied to a shared LoRA module to mitigate catastrophic forgetting in sequential sentence-level tasks. A lightweight memory bank stores CLS embeddings from previous tasks, and alignment losses regularize representations of the current task against stored references. Three alignment losses, Maximum Mean Discrepancy (MMD), full KL divergence, and diagonal KL divergence, and three CLS alignment modes, final-layer, layer-average, and weighted average, are evaluated to identify strategies that stabilize intermediate representations. The framework updates only the shared LoRA module while keeping transformer parame-

ters frozen, achieving memory-efficient continual learning without storing raw data.

Key contributions:

- CLS-based representation alignment on a shared LoRA module to mitigate catastrophic forgetting in sequential sentence-level tasks.
- Systematic evaluation of three alignment losses and three CLS alignment modes to identify optimal strategies for preserving intermediate representations.
- Analysis of task order effects, demonstrating robust stabilization of features across sequences.
- Higher accuracy and reduced forgetting compared with regularization-based baselines, with retention comparable to rehearsal-based approaches without storing raw data.
- Memory-efficient, scalable, and practical continual learning framework suitable for transformer-based NLP models.

2 Related Work

2.1 Regularization and Rehearsal Methods

Continual learning enables models to acquire new knowledge sequentially while retaining previously learned information. Approaches are commonly categorized as regularization-based, replay-based, or architectural methods.

Regularization-based methods, including Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017b), Synaptic Intelligence (SI) (Zenke et al., 2017b), Memory Aware Synapses (MAS) (Aljundi et al., 2018a), and Riemannian Walk (RWalk) (Chaudhry et al., 2018), penalize updates to parameters important for prior tasks. These methods reduce catastrophic forgetting but often struggle with highly diverse or non-stationary tasks.

Replay-based methods, such as Learning without Forgetting (LwF) (Li and Hoiem, 2016) and Rehearsal (Deng et al., 2025), retain knowledge by revisiting samples or learned representations from earlier tasks. While effective, they increase memory and computational requirements and may raise privacy concerns.

Architectural approaches add task-specific modules, such as adapters or subnetworks. They isolate task interference but require task identifiers at inference and lead to unbounded model growth, limiting scalability.

2.2 Adapter and LoRA-Based Tuning

Parameter-efficient fine-tuning methods, including adapters and Low-Rank Adaptation (LoRA), adapt large pretrained language models with minimal additional parameters. These methods keep the backbone frozen and introduce lightweight modules, using adapters as bottleneck layers and LoRA as low-rank decompositions of weight updates.

In continual learning, separate adapter or LoRA modules are often assigned to each task to prevent forgetting. This design leads to linear parameter growth and limits cross-task generalization. Shared adapters or LoRA modules have been explored to enhance efficiency and enable knowledge transfer without task-specific routing.

2.3 Representation Alignment

Representation alignment preserves feature consistency across tasks by minimizing distributional shifts in intermediate representations. Techniques such as Maximum Mean Discrepancy (MMD) and Kullback–Leibler (KL) divergence promote stable features, reducing catastrophic forgetting (Bhatt et al., 2024; Vander Eeckt and Van hamme, 2024; Long et al., 2015).

Unlike regularization methods, alignment operates in the feature space, maintaining semantic coherence even as parameters evolve. In NLP, CLS-token alignment in transformer models remains underexplored. The framework integrates a shared LoRA module with CLS-token alignment to achieve stable representations, enabling effective task-incremental continual learning with per-task classifier heads.

3 Proposed Method

3.1 Problem Definition and Model Architecture

Consider a sequence of n tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$, each associated with a dataset $\mathcal{D}_i = \{(x_j, y_j)\}$, where x_j denotes an input sentence and y_j the corresponding label. The goal is to train a model that performs well across all tasks without retaining access to data from previous tasks during subsequent training.

The model comprises a frozen DistilBERT encoder f_θ, parameterized by θ, and a shared low-

rank adaptation (LoRA) module A_ϕ, parameterized by ϕ. The LoRA module modifies the query and value projection matrices in self-attention layers using a low-rank decomposition:

$$W' = W + BA,$$

where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ are trainable matrices of rank $r \ll \min(d, k)$, and W is a fixed pretrained matrix. Only B and A are updated during training, while W remains fixed.

Unlike prior approaches that allocate separate LoRA modules per task, this framework employs a single shared LoRA module for all tasks, reducing memory overhead, improving scalability, and promoting knowledge sharing. During training on task $\mathcal{T}_t$, only the adapter parameters ϕ are updated by minimizing the task-specific loss $\mathcal{L}_{\text{task}}^{(t)}$, while the encoder parameters θ remain fixed, preserving pretrained representations and mitigating forgetting.

Algorithm 1 Proposed CLS Alignment Loss Computation

Input: Current embeddings Z_t, reference embeddings Z_{ref}, alignment mode
Output: Alignment loss $\mathcal{L}_{\text{align}}$
if mode == MMD **then**
  $\mathcal{L}_{\text{align}} = \text{MMD}(Z_t, Z_{\text{ref}})$
else if mode == KL **then**
  Compute $\mu_t, \Sigma_t, \mu_{\text{ref}}, \Sigma_{\text{ref}}$
  $\mathcal{L}_{\text{align}} = D_{\text{KL}}\big(\mathcal{N}(\mu_t, \Sigma_t) \,\|\, \mathcal{N}(\mu_{\text{ref}}, \Sigma_{\text{ref}})\big)$
else if mode == KL-D **then**
  Compute $\mu_t, \sigma_t^2, \mu_{\text{ref}}, \sigma_{\text{ref}}^2$
  $\mathcal{L}_{\text{align}} = D_{\text{KL}}\big(\mathcal{N}(\mu_t, \text{diag}(\sigma_t^2)) \,\|\, \mathcal{N}(\mu_{\text{ref}}, \text{diag}(\sigma_{\text{ref}}^2))\big)$
end if
return $\mathcal{L}_{\text{align}}$

3.2 Proposed Representation Alignment and CLS Loss

A central challenge in continual learning is representational drift, where feature embeddings shift as new tasks are learned, causing forgetting of previous knowledge. To address this, a representation alignment loss is introduced that regularizes the distribution of classification token embeddings (CLS embeddings) between the current task and prior tasks. This loss encourages stable representations across tasks while allowing the model to adapt to new information.

Three types of CLS alignment loss are employed, as summarized in Algorithm 1: Maximum Mean Discrepancy (MMD), Full KL-Divergence (KL), and Diagonal KL-Divergence (KL-D). MMD measures distributional differences in a reproducing kernel Hilbert space, KL compares Gaussian approximations using full covariance matrices, and KL-D offers a computationally efficient approximation assuming diagonal covariance matrices.

The total training loss for task $\mathcal{T}_t$ is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}}^{(t)} + \lambda \mathcal{L}_{\text{align}},$$

where λ balances task-specific optimization with representation consistency. To avoid storing raw inputs or full model checkpoints, a memory bank Z_{ref} retains a small set of CLS embeddings sampled from validation data after each task, efficiently approximating past feature distributions while preserving privacy.

3.3 Training Procedure with Memory Bank Alignment

The proposed training strategy, summarized in Algorithm 2, updates the shared LoRA parameters using the total loss, which combines task-specific loss with CLS alignment loss to maintain feature distribution consistency. CLS embeddings from mini-batches are compared against a memory bank containing embeddings from previous tasks.

After each task, a subset of validation CLS embeddings is added to the memory bank, providing a compact summary of prior tasks without storing raw data. This incremental update enables the model to learn new tasks while preserving knowledge from earlier tasks, ensuring both continual learning performance and memory efficiency.

3.4 CLS Alignment Modes

CLS embeddings are extracted from DistilBERT using three strategies, as summarized in Algorithm 3. These modes define how the representation of the classification token is formed, balancing efficiency, contextual richness, and adaptive flexibility.

CLS Final (CLS-F) extracts the CLS embedding from the final transformer layer, capturing the most abstract features of the input. This mode is the most computationally efficient but may lose useful intermediate information. CLS Avg (CLS-A) averages embeddings from the last four layers, providing a richer representation that blends both high-level semantics and mid-level contextual cues. This approach increases computational cost slightly but

Algorithm 2 Proposed Continual Learning with Shared LoRA and Memory Bank Alignment

Require: Frozen DistilBERT backbone f_θ, shared LoRA parameters ϕ, task sequence $\{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$, alignment weight λ, alignment loss function $\mathcal{L}_{\text{align}}$

Initialize memory bank $Z_{\text{ref}} \leftarrow \emptyset$
for $t = 1$ to n **do**
  Load task data $\mathcal{D}_t = \{(x_i, y_i)\}$
  for each epoch **do**
    for mini-batch $B_t \subset \mathcal{D}_t$ **do**
      Predict $y_{\text{pred}} \leftarrow f_\theta(x; \phi)$
      Extract CLS embeddings Z_t from B_t
      Compute task loss: $\mathcal{L}_{\text{task}} \leftarrow \text{CrossEntropy}(y_{\text{pred}}, y)$
      Compute alignment loss:

$$\mathcal{L}_{\text{align}} \leftarrow \begin{cases} \text{Align}(Z_t, Z_{\text{ref}}), & Z_{\text{ref}} \neq \emptyset \\ 0, & \text{otherwise} \end{cases}$$

      Total loss: $\mathcal{L} \leftarrow \mathcal{L}_{\text{task}} + \lambda \cdot \mathcal{L}_{\text{align}}$
      Update ϕ via gradient descent on $\mathcal{L}$
    end for
  end for
  Extract CLS embeddings from validation set Z_t^{val}
  Update memory bank: $Z_{\text{ref}} \leftarrow Z_{\text{ref}} \cup Z_t^{\text{val}}[: K]$
end for

Algorithm 3 CLS Alignment Modes

CLS-F:

$$\mathbf{h}_{\text{CLS}}^{\text{last}} = H^{(L)}[0]$$

CLS-A:

$$\mathbf{h}_{\text{CLS}}^{\text{avg}} = \frac{1}{4} \sum_{l=L-3}^{L} H^{(l)}[0]$$

CLS-W:

$$\alpha_l = \frac{\exp(\mathbf{w}^\top H^{(l)}[0])}{\sum_{j=1}^{L} \exp(\mathbf{w}^\top H^{(j)}[0])}$$

$$\mathbf{h}_{\text{CLS}}^{\text{weighted}} = \sum_{l=1}^{L} \alpha_l H^{(l)}[0]$$

often yields more stable features. CLS Weighted (CLS-W) introduces learnable attention weights across all layers, allowing the model to adaptively focus on layers that provide the most informative features for downstream tasks. Although this mode incurs the highest cost, it offers the greatest flexibility and often improves robustness by balancing shallow and deep representations.

In summary, CLS-F prioritizes speed, CLS-A provides a balanced trade-off between efficiency and accuracy, and CLS-W maximizes adaptability at the expense of additional computation.

4 Experimental Setup

4.1 Datasets

The study adopts a task-incremental continual learning scenario using four tasks from the GLUE benchmark: SST-2 (sentiment analysis), MRPC (paraphrase detection), RTE (recognizing textual entailment), and CoLA (linguistic acceptability). Tasks are presented sequentially, and the model is trained on each one independently, without access to data from previously seen tasks, enforcing a strict continual learning protocol.

The tasks differ in linguistic structures, label semantics, data sizes, and domain characteristics. SST-2 and CoLA involve single-sentence classification, while MRPC and RTE require reasoning over sentence pairs. This heterogeneity creates distribution shifts and varying representational demands, providing a comprehensive testbed for evaluating catastrophic forgetting and assessing the impact of CLS alignment-based mitigation strategies.

4.2 Model Configuration

DistilBERT serves as a frozen backbone, augmented with a shared LoRA module inserted into the query and value projection layers of all six transformer blocks. The LoRA rank is set to $r = 8$, dropout is 0.1, and the scaling factor α is 16. Only LoRA parameters are trainable while backbone parameters remain fixed. Each targeted linear layer has 768×768 weights, with two LoRA modules per layer, resulting in approximately 0.147M trainable parameters.

The [CLS] token embedding summarizes the input sequence. CLS embeddings are extracted according to three strategies: CLS-F uses the final layer embedding directly, which is efficient but may miss intermediate-layer information. CLS-A averages embeddings from the last four layers to

capture richer context. CLS-W applies learnable layer-wise attention across all six layers, emphasizing the most informative representations. These embeddings feed into a task-specific classifier, which is reinitialized at the beginning of each task to accommodate differences in label space. Classifier parameters are task-specific and do not contribute to catastrophic forgetting.

To mitigate forgetting, a memory bank stores up to 100 [CLS] embeddings per task from the validation set. The alignment loss, implemented using KL divergence (full or diagonal) or MMD, enforces consistency in internal feature distributions across tasks, preserving prior knowledge while allowing adaptation to new tasks.

4.3 Training Procedure

Training is conducted for three epochs per task with a batch size of 32 and a learning rate of 5×10^{-5}. LoRA parameters are updated using the total loss, which combines task-specific loss with CLS alignment loss. CLS embeddings from mini-batches are compared against the memory bank to maintain representation consistency. After each task, sampled validation CLS embeddings are appended to the memory bank, ensuring incremental coverage of prior distributions.

4.4 Baselines and Comparative Methods

Comparative methods include full fine-tuning of DistilBERT and task-specific LoRA, which serve as upper-bound baselines with dedicated parameters per task but require task identity at inference. Regularization-based methods, including EWC, SI, MAS, and RWalk, constrain updates to reduce forgetting. Learning without Forgetting (LwF) applies knowledge distillation to preserve prior task predictions. Rehearsal retains a small subset of data from previous tasks. A baseline with shared LoRA but without CLS alignment ($\lambda = 0$) isolates the effect of representation-level alignment. CLS alignment-based methods use KL-Divergence with $\lambda = 15$, and performance is reported in terms of Average Accuracy and Forgetting Measure across multiple task orders.

4.5 Evaluation Metrics

Performance is assessed using standard GLUE metrics: accuracy for SST-2 and RTE, F1 score for MRPC, and Matthews correlation for CoLA. Average Accuracy reflects overall performance after sequential training. Backward Transfer measures the effect of learning new tasks on previous tasks, with negative values indicating forgetting. Forgetting Measure quantifies catastrophic forgetting independently of Backward Transfer. Total trainable parameters, including LoRA modules, and average inference time per batch on a single GPU with batch size 32 are also reported to evaluate parameter efficiency and computational cost.

CLS Align. Loss	λ	AA	FM
KL-D	15	0.4278	0.1683
KL	15	0.4240	0.2135
MMD	15	0.4579	0.1733

Table 1: Comparison of CLS alignment loss strategies. Reported AA and FM are averaged over randomly chosen task orders.

5 Results and Discussion

5.1 Effect of CLS Align. Loss and λ

The influence of different representation alignment losses and the alignment weight λ is summarized in Table 1. Three alignment strategies, diagonal KL divergence (KL-D), full KL divergence (KL), and Maximum Mean Discrepancy (MMD), were evaluated. The overall training objective is defined as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \cdot \mathcal{L}_{\text{align}},$$

where λ controls the strength of alignment regularization. Values of λ ranging from 0 to 30 were explored. Low values ($\lambda \leq 5$) are insufficient to constrain CLS drift, resulting in higher forgetting (larger FM), while excessively large values ($\lambda \geq 20$) over-regularize the feature space, limiting adaptation to new tasks and reducing task accuracy despite lower forgetting.

Intermediate values around $\lambda = 15$ provide the best balance between plasticity and stability, maintaining high average accuracy while effectively mitigating forgetting. MMD achieves the highest average accuracy (AA = 0.4579) at $\lambda = 15$, whereas KL-D achieves the lowest forgetting measure (FM = 0.1683), demonstrating that diagonal covariance alignment stabilizes feature distributions more effectively than full KL divergence, which exhibits slightly higher FM (0.2135) due to off-diagonal noise. KL-D focuses on the variance along each feature dimension independently, reducing instability caused by noisy correlations in off-diagonal

CLS Align. Mode	λ	CLS Align. Loss	Task Order	AA (%)	FM
CLS-W	15	KL-D	[sst2, cola, mrpc, rte]	34.59	0.2928
CLS-W	15	KL-D	[mrpc, sst2, rte, cola]	45.37	0.1314
CLS-W	15	KL-D	[cola, sst2, rte, mrpc]	50.74	0.1670
Avg. (CLS-W)	15	KL-D	All Orders	**43.57**	**0.1971**
CLS-A	15	KL-D	[sst2, cola, mrpc, rte]	31.89	0.3437
CLS-A	15	KL-D	[mrpc, sst2, rte, cola]	49.49	0.1387
CLS-A	15	KL-D	[cola, sst2, rte, mrpc]	49.87	0.1691
Avg. (CLS-A)	15	KL-D	All Orders	**43.75**	**0.2172**
CLS-F	15	KL-D	[sst2, cola, mrpc, rte]	35.01	0.2727
CLS-F	15	KL-D	[mrpc, sst2, rte, cola]	46.09	0.1479
CLS-F	15	KL-D	[cola, sst2, rte, mrpc]	50.29	0.1781
Avg. (CLS-F)	15	KL-D	All Orders	**43.80**	**0.1996**

Table 2: Comparison of AA and FM across CLS alignment modes.

terms, which explains its superior feature stability. The system shows moderate sensitivity to λ; small deviations from the optimum slightly affect AA or FM, but performance drops sharply at the extremes. KL-D with $\lambda = 15$ was selected for all subsequent experiments. Detailed analysis of λ effects was conducted but is not displayed due to page limitations.

5.2 Effect of CLS Alignment Mode

Table 2 compares three CLS alignment modes: CLS-W, CLS-A, and CLS-F. CLS-W applies learnable layer-wise attention across all transformer layers, achieving an average AA of 43.57% and FM of 0.1971 across three task orders. CLS-A and CLS-F achieve similar average accuracy (43.75% and 43.80%) with slightly higher FM (0.2172 and 0.1996). CLS-F attains the highest AA while maintaining FM similar to CLS-W, indicating that the final-layer embedding alone sufficiently captures task-relevant information.

CLS-W emphasizes informative layers, improving robustness across heterogeneous tasks and reducing catastrophic forgetting. CLS-A aggregates the last four layers to capture richer contextual features, though this may dilute discriminative information. CLS-F is simpler, memory-efficient, and computationally minimal, making it a strong choice for resource-constrained scenarios while maintaining competitive retention.

5.3 Memory Bank Size and Its Impact on Performance

The memory bank stores 100 CLS embeddings per task, sampled from the validation set with class-stratified selection. For RTE (277 validation set) and MRPC (408 validation set), 100 embeddings represent roughly 36% and 25% of the validation set, providing substantial coverage. For larger tasks such as CoLA (1,043 validation set) and SST-2 (872 validation set), 100 embeddings represent roughly 10–12% of the validation set, providing sufficient diversity to stabilize CLS representations and mitigate forgetting.

A memory bank of 100 embeddings balances representation coverage, memory efficiency, and alignment effectiveness. Each input sentence produces one CLS embedding, and this subset captures enough diversity for effective alignment. Smaller banks may underrepresent feature variability, while larger banks increase memory requirements with minimal performance gain. This configuration supports reproducibility and efficient continual learning across tasks of varying sizes.

5.4 Effect of Task Order

Task sequencing significantly influences continual learning outcomes. For CLS-F, AA varies from 35.4% to 49.3% and FM from 0.142 to 0.272 depending on task order (Table 3). Similar variability occurs in regularization-based methods such as EWC, SI, MAS, and RWalk, demonstrating sensitivity to task sequencing. Rehearsal exhibits smaller fluctuations (AA 45.5%–48.6%, FM 0.156–0.198), reflecting the stabilizing effect of retaining previous task examples.

Earlier tasks strongly influence parameter adaptation, and catastrophic forgetting increases when initial tasks dominate the learned representation. Averaging performance over multiple task orders provides a more reliable assessment of method sta-

bility. CLS alignment mitigates this effect by preserving feature distributions via a compact memory bank, achieving retention comparable to rehearsal and outperforming regularization-based methods.

5.5 Comparison with Baselines

Table 3 compares CLS-F with standard baselines. Regularization-based methods show lower AA and higher FM, with FM reductions of 47.7% (EWC), 48.2% (SI), 56.9% (MAS), 58.9% (RWalk), and 33.0% (LwF), indicating less effective retention. Rehearsal achieves slightly better AA and an FM reduction of 11.7%, but requires storing raw data from previous tasks.

CLS-F attains higher AA than regularization-based approaches while mitigating forgetting, updating only 0.147M shared LoRA parameters, and maintaining feature distribution consistency via 100 CLS embeddings per task. Compared with baselines, CLS alignment reduces forgetting more effectively and provides retention comparable to Rehearsal without storing raw inputs.

All regularization-based methods and Rehearsal update 0.147M parameters, whereas full fine-tuning (F-FT) updates 66M parameters and task-specific LoRA (T-LoRA) requires 0.147M per task (0.588M for four tasks). Inference remains efficient at 23 ms per batch, demonstrating that CLS-F is a lightweight and robust representation-level continual learning strategy.

5.6 Summary of Findings

Representation-level alignment using CLS embeddings effectively mitigates catastrophic forgetting in continual learning. Diagonal KL divergence (KL-D) ensures stable preservation of feature distributions. CLS-F consistently achieves higher AA than regularization-based baselines while substantially reducing forgetting, with FM reductions of 47.7% (EWC), 48.2% (SI), 56.9% (MAS), 58.9% (RWalk), and 33.0% (LwF). Rehearsal slightly surpasses CLS-F in AA and provides an FM reduction of 11.7%, but requires storing raw data from previous tasks. CLS-F achieves competitive accuracy with minimal memory overhead of 0.147M trainable parameters and only 100 CLS embeddings per task, offering a practical and efficient solution. Task order significantly affects performance, highlighting the importance of evaluation across multiple sequences. Overall, CLS alignment provides a scalable, memory-efficient, and robust continual learning strategy with strong mitigation of forgetting.

6 Conclusion

Representation-level alignment using CLS embeddings provides an effective approach for task-incremental continual learning. Diagonal KL divergence (KL-D) ensures stable mitigation of catastrophic forgetting, with the alignment weight λ balancing retention of prior knowledge and acquisition of new tasks. Among CLS alignment strategies, CLS Final (CLS-F), which leverages the CLS token from the final transformer layer, consistently achieves higher AA than regularization-based methods while substantially reducing forgetting (FM reductions of 47.7% EWC, 48.2% SI, 56.9% MAS, 58.9% RWalk, 33.0% LwF). CLS-F provides retention comparable to rehearsal-based approaches (FM reduction 11.7%) without storing raw task data, using only 100 CLS embeddings per task and 0.147M trainable parameters. Task order analysis confirms robust performance across sequences. Overall, CLS alignment delivers a lightweight, memory-efficient, and scalable continual learning strategy that reduces forgetting, outperforms regularization methods, and offers practical retention comparable to rehearsal-based approaches on the GLUE benchmark.

Limitations

While the proposed CLS alignment-based continual learning framework demonstrates strong performance and memory efficiency, several limitations should be noted. First, the method relies on maintaining a memory bank of CLS embeddings for each task, which, although compact, introduces additional storage requirements that scale linearly with the number of tasks. Second, the effectiveness of CLS alignment is influenced by task similarity and sequence order; highly heterogeneous tasks or adversarial task orders may still result in suboptimal retention compared with rehearsal-based methods. Third, the approach primarily targets representational drift at the CLS token level and may be less effective in scenarios where forgetting occurs in deeper, task-specific feature subspaces not captured by the CLS embedding. Fourth, selecting the alignment weight λ requires empirical tuning, and inappropriate values can either restrict learning on new tasks or fail to prevent forgetting. Finally, the current framework has been evaluated on text

Method	λ	Task Order	AA (% ↑)	FM (↓)	FM Reduction (% ↑)
CLS-F	15	['sst2', 'cola', 'mrpc', 'rte']	35.4	0.272	-
CLS-F	15	['mrpc', 'sst2', 'rte', 'cola']	46.7	0.142	-
CLS-F	15	['cola', 'sst2', 'rte', 'mrpc']	49.3	0.178	-
CLS-F (avg.)	15	-	**43.8**	**0.197**	**0**
F-FT	-	['sst2', 'cola', 'mrpc', 'rte']	75.2		-
T-LoRA	-	['mrpc', 'sst2', 'rte', 'cola']	67.6		-
EWC	1000	['sst2', 'cola', 'mrpc', 'rte']	23.7	0.527	-
EWC	1000	['mrpc', 'sst2', 'rte', 'cola']	52.8	0.149	-
EWC	1000	['cola', 'sst2', 'rte', 'mrpc']	48.5	0.197	-
EWC (avg.)	1000	-	**41.7**	**0.291**	**-47.7**
SI	1000	['sst2', 'cola', 'mrpc', 'rte']	25.4	0.522	-
SI	1000	['mrpc', 'sst2', 'rte', 'cola']	52.4	0.147	-
SI	1000	['cola', 'sst2', 'rte', 'mrpc']	47.9	0.211	-
SI (avg.)	1000	-	**41.9**	**0.293**	**-48.2**
MAS	1000	['sst2', 'cola', 'mrpc', 'rte']	23.4	0.548	-
MAS	1000	['mrpc', 'sst2', 'rte', 'cola']	51.8	0.155	-
MAS	1000	['cola', 'sst2', 'rte', 'mrpc']	47.9	0.223	-
MAS (avg.)	1000	-	**41.0**	**0.309**	**-56.9**
RWalk	1000	['sst2', 'cola', 'mrpc', 'rte']	26.1	0.546	-
RWalk	1000	['mrpc', 'sst2', 'rte', 'cola']	53.0	0.147	-
RWalk	1000	['cola', 'sst2', 'rte', 'mrpc']	47.1	0.246	-
RWalk (avg.)	1000	-	**42.1**	**0.313**	**-58.9**
LwF	15	['sst2', 'cola', 'mrpc', 'rte']	34.4	0.389	-
LwF	15	['mrpc', 'sst2', 'rte', 'cola']	41.8	0.154	-
LwF	15	['cola', 'sst2', 'rte', 'mrpc']	27.2	0.243	-
LwF (avg.)	15	-	**34.5**	**0.262**	**-33.0**
Rehearsal	-	['sst2', 'cola', 'mrpc', 'rte']	45.7	0.198	-
Rehearsal	-	['mrpc', 'sst2', 'rte', 'cola']	48.6	0.156	-
Rehearsal	-	['cola', 'sst2', 'rte', 'mrpc']	45.5	0.169	-
Rehearsal (avg.)	-	-	**46.6**	**0.174**	**11.7**

Table 3: Average Accuracy (AA, %) and Forgetting Measure (FM) for CLS alignment and baseline methods under different task orders on the task-incremental GLUE benchmark. FM Reduction (%) is computed relative to CLS-F average FM (0.197) and shown only for average rows. F-FT denotes full fine-tuning, T-LoRA denotes task-specific LoRA, and FM corresponds to negative BWT. CLS Align. CLS Align.Loss is KL-D and $\lambda = 15$. Parameter counts and inference time (23 ms) are constant across all alignment-based methods.

classification benchmarks; extending CLS alignment to larger-scale datasets, multimodal tasks, or generative models may require additional adaptation and validation. Addressing these limitations provides opportunities for future work, including dynamic memory bank management, adaptive λ scheduling, and integrating CLS alignment with other complementary continual learning strategies.

Ethics Statement

This work adheres to the ACL Code of Ethics. Experiments use only publicly available GLUE datasets with no personal or sensitive information. The proposed CLS-based representation alignment improves continual learning in NLP models, reducing forgetting without storing raw task data, minimizing privacy risks. Potential misuse for biased or harmful applications is acknowledged, and responsible deployment is encouraged. A completed ARR Responsible NLP Research checklist has been submitted alongside this paper.

References

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

Rahaf Aljundi, Min Lin, Baptiste Goujaud, Yoshua Bengio, and Tinne Tuytelaars. 2018b. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154.

Gaurav Bhatt, Leonid Sigal, and James Ross. 2024. Preventing catastrophic forgetting through memory networks in continuous detection. In *European Conference on Computer Vision (ECCV)*. Memory-augmented continual DETR prevents forgetting; 5–7% improvement.

Pietro Buzzega, Matteo Boschini, Alessandro Porrello, Simone Calderara, and Rita Cucchiara. 2020. Dark experience for general continual learning: A strong, simple baseline. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:15920–15930.

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

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2022. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(7):3366–3385.

Junze Deng, Qinhang Wu, Peizhong Ju, Sen Lin, Yingbin Liang, and Ness Shroff. 2025. Unlocking the power of rehearsal in continual learning: A theoretical perspective. *Proceedings of the International Conference on Machine Learning (ICML)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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: Human Language Technologies (NAACL-HLT)*, pages 4171–4186. Association for Computational Linguistics.

Sunny Dhawan, Abhishek Kumar, and 1 others. 2025. Sentiment analysis applications using deep learning advancements in social networks: A systematic review. *Neurocomputing*, 540:127–145.

Yujie Feng, Xujia Wang, Zexin Lu, Shenghong Fu, Guangyuan Shi, Yongxin Xu, Yasha Wang, Philip S. Yu, Xu Chu, and Xiao-Ming Wu. 2025. Recurrent knowledge identification and fusion for language model continual learning. *arXiv preprint arXiv:2502.17510*.

Nitin Gupta and Priya Kumar. 2025. A deep learning approach for paragraph-level paraphrase generation and detection. *Neural Processing Letters*, 57:145–162.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhong Li, Shean Wang, Lu Chen, Weizhu Li, and Bing Wang. 2024. Lora: Low-rank adaptation of large language models. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS)*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017a. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences (PNAS)*, 114(13):3521–3526.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, and 1 others. 2017b. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Lola Le Breton, Quentin Fournier, Mariam El Mezouar, and Sarath Chandar. 2025. Neobert: A next-generation bert. *arXiv preprint arXiv:2502.19587*.

Zhizhong Li and Derek Hoiem. 2016. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, pages 97–105.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

Vlad Andrei Negru, Robert Vacareanu, Camelia Lemnaru, Mihai Surdeanu, and Rodica Potolea. 2025. Morphnli: A stepwise approach to natural language inference using text morphing. *arXiv preprint arXiv:2502.09567*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. Fine-tune bert for extractive summarization. In *Proceedings of the China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Yukun Sun, Shuohuan Wang, Yichun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie-doc: A retrospective long-document modeling transformer. In *Proceedings of the 2021 AAAI Conference on Artificial Intelligence*.

Steven Vander Eeckt and Hugo Van hamme. 2024. Unsupervised online continual learning for automatic speech recognition. *arXiv preprint arXiv:2406.12503*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Jiuqi Wang, Rohan Chandra, and Shangtong Zhang. 2025. Experience replay addresses loss of plasticity in continual learning. *arXiv preprint arXiv:2503.20018*.

Kezhen Wang, Ethan Perez, Xinyi Chen, Yuan Cao, Shixiang Xie, Qirong Li, and Diyi Yang. 2022a. Parameter-efficient tuning enables continual learning in transformers. *arXiv preprint arXiv:2206.12558*.

Kezhen Wang, Ethan Perez, Xinyi Chen, Yuan Cao, Shixiang Xie, Qirong Li, and Diyi Yang. 2022b. Parameter-efficient tuning enables continual learning in transformers. *arXiv preprint arXiv:2206.12558*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1112–1122.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017a. Continual learning through synaptic intelligence. *arXiv preprint arXiv:1703.04200*.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017b. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR. org.

Jingfan Zhang, Yi Zhao, Dan Chen, Xing Tian, Huanran Zheng, and Wei Zhu. 2024. Milora: Efficient mixture of low-rank adaptation for large language models fine-tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 994–1003.