LEARNING TO EVOLVE: ENABLING LLM CONTINUAL LEARNING VIA THE STABLE ANCHOR DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning seeks to enable sequential mastery of a stream of tasks, simultaneously securing competent performance on the newly introduced task while preserving the proficiency acquired on earlier ones. In the contemporary era of large language models (LLMs), the performance degradation on old tasks, i.e., catastrophic forgetting, is partially mitigated by the inherently massive parameter scale and the extensive pre-training corpora. Nevertheless, due to the fundamental mechanics of gradient-based training, the acquisition of a new task invariably leaves an imprint on the parameter configuration, thereby exerting a nontrivial impact on previously established capabilities. In this paper, we probe into the very nature of forgetting and introduce the notion of **Stable Anchor**, which theoretically traces forgetting to the propagation of stable anchor changes from the new task to the old one (via the parameter space), ultimately altering its performance. Building upon this insight, we propose a loss function designed to minimize the influence transmitted via the stable anchor. Extensive experiments across diverse task streams substantiate the effectiveness of the proposed method.

1 INTRODUCTION

Large language models (LLMs) have demonstrated a considerable degree of resilience to catastrophic forgetting during continual learning, a phenomenon largely attributable to their massive parameter scale and the extensive diversity of pre-training corpora Zheng et al. (2024; 2025). However, this resilience is far from absolute. Owing to the fundamental mechanics of gradient-based optimization, the sequential acquisition of the new task invariably enforces global parameter modifications, thereby eroding the performance on tasks encountered earlier. Curbing this residual interference remains a central challenge for LLM continual learning—one that demands a mechanistic understanding of how forgetting arises Lin et al. (2022); Shi et al. (2025).

Prior efforts to address this challenge have largely coalesced around explicit manipulations of the parameter space, which can be broadly organized into two paradigms of approaches (displayed in Figure 1). Underlying both paradigms is a shared intuition: if the mutual influence between the parameters of different tasks can be sufficiently constrained, then forgetting will naturally be mitigated. The first paradigm advocates parameter-space orthogonality, exemplified by O-LoRA Wang et al. (2023); Liang et al. (2026), wherein the parameter subspace optimized for a new task is constrained to be orthogonal to those of preceding tasks. This strict geometric separation achieves near-complete decoupling between old and new task solutions, thereby substantially reducing interference. However, such orthogonality-based approaches enforce a rigid independence that entirely precludes the possibility of beneficial knowledge transfer across related tasks. The second paradigm pursues parameter-space decomposition, as typified by Tree-LoRA Qian et al. (2025), which partitions parameters into shared and task-specific components. The degree of sharing is governed by a measure of inter-task similarity (for instance, Tree-LoRA employs the similarity between task gradients to determine which parameters are shared). Nevertheless, this decomposition strategy suffers from a fundamental limitation: the allocation of parameters to shared versus task-specific sets is inherently discrete—a parameter subset is either shared or not. In reality, a given set of parameters may simultaneously contribute to both the shared commonality and the task-specific distinctiveness to varying degrees, a nuance that discrete allocation cannot capture.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

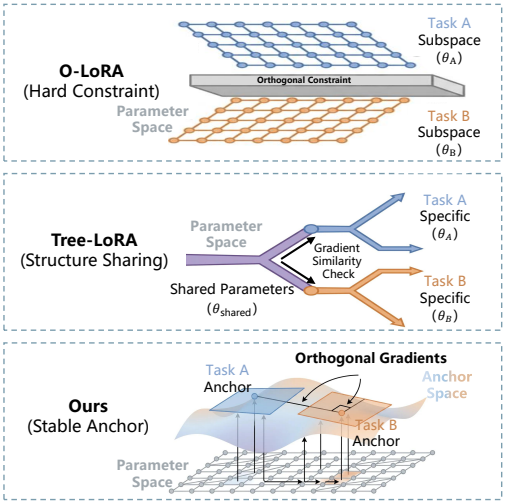

Figure 1: **Comparison of LLM Continual Learning Strategies:** Conventional approaches constrain the parameter space to reduce cross-task interference (O-LoRA: orthogonal parameters; Tree-LoRA: task-shared & task-specific partitions). Our method foregoes explicit parameter-space intervention: we instead derive the theoretical drivers of forgetting and formulate a loss function that counteracts them directly.

Rather than intervening directly in the parameter space, we seek to understand what forgetting actually is, from the theoretical perspective. To this end, we formulate forgetting as the variation in the output of an old task induced by the training of a new one, and introduce the concept of **Stable Anchor**—an intermediate representation residing in a latent space of the input. The defining property of the stable anchor is its invariance with respect to the generation process: so long as the stable anchor remains unchanged, the final model output stays invariant irrespective of any alterations in the parameters governing the generation mapping. **This invariance property allows us to establish a rigorous equivalence that the output variation of an old task can be reduced entirely to the variation of its stable anchor**. We further decompose the change rate of the stable anchor, revealing the mechanistic pathway through which a new task impacts an old one. Specifically, **changes in stable anchor of the new task propagate through the parameter space and impinge upon the stable anchor of the old task, ultimately manifesting as a shift in its output distribution**. Based on this finding, we propose a loss function that mitigates the impact on the stable anchor, thereby facilitating the progression of continual learning.

Our contributions are summarized as follows:

- We offer a theoretical elucidation of catastrophic forgetting, establishing the reciprocal perturbation between the stable anchor of tasks as a principal driver of performance deterioration.
- Building upon this insight, we devise a loss function explicitly tailored to curtail such cross-task interference, thereby ameliorating forgetting throughout the continual learning process.
- We validate the proposed theory across multiple task streams, demonstrating its consistent efficacy.

2 Related Work

Continual learning aims to enable models to acquire new knowledge sequentially without forgetting previously learned tasks Wickramasinghe et al. (2024); Wang et al. (2024); Lin et al. (2025); Imanov (2026). As large pre-trained models become the dominant paradigm, Low-Rank Adaptation (LoRA) Hu et al. (2022) has emerged as a particularly influential parameter-efficient fine-tuning technique for building continual learning systems. Existing LoRA-based continual learning methods can be broadly organized along two principal axes: parameter-space orthogonality, which imposes hard numerical constraints to ensure that updates for different tasks occupy mathematically orthogonal subspaces and thus theoretically eliminates cross-task interference; and parameter-space decomposition, which structurally partitions the parameter landscape into functional modules or hierarchical regions to achieve flexible task separation without strict mathematical orthogonality. This section reviews representative works along both axes.

Methods of **parameter-space orthogonality** directly constrain the numerical values of LoRA parameters such that task updates are restricted to directions that are mathematically orthogonal to existing knowledge. O-LoRA Wang et al. (2023) enforces explicit orthogonality between the column spaces of a new task's LoRA matrix and those of all previous tasks' via a regularization loss, while freezing all previous LoRA parameters to prevent interference. InfLoRA Liang & Li (2024) approaches orthogonality by leveraging gradient projection memory (DualGPM): it designs the dimensionality-

reduction matrix such that its row space lies in the intersection of the new task's gradient space and the orthogonal complement of the old task gradient space, thereby achieving interference-free learning with only the parameters trainable per task. KeepLoRA Luo et al. (2026) further advances the framework by constraining gradient updates to a residual subspace orthogonal to both the pre-trained model's principal components and the accumulated historical task directions, initializing the LoRA down-projection matrix with the projected gradient's top singular vectors to balance plasticity, backward stability, and forward stability. These methods form a progression in the sophistication of orthogonality constraints, yet all share a common trade-off: they impose hard numerical constraints that guarantee zero interference at the potential cost of limiting beneficial knowledge transfer when tasks share underlying structure. Qiao & Mahdavi propose SLAO, a continual learning method that sequentially merges new task LoRA updates into a single unified LoRA, eliminating the need to store separate adapters for each task. It leverages orthogonal basis extraction from previously learned LoRA components to initialize new task learning, and exploits the inherent asymmetry of LoRA matrices through a time-aware scaling mechanism to balance knowledge retention and plasticity during merging

Methods of **parameter-space decomposition**, in contrast, achieve task separation not by enforcing numerical orthogonality but by structurally organizing parameters into distinct functional regions. MoE Kang et al. (2025) introduces an adversarial mixture-of-experts framework comprising task-specific LoRA experts and a shared LoRA expert, where a task-aware discriminator forces the shared expert to suppress task-discriminative information, thereby achieving implicit functional decoupling through architectural separation. MoLE Wu et al. (2024); Wang et al. (2025) treats each trained LoRA as an expert and learns a hierarchical layer-wise gating function for dynamic composition of pre-existing LoRAs, enabling flexible cross-task knowledge recombination at inference time without further training. TreeLoRA Qian et al. (2025) organizes LoRA adapters in a hierarchical tree structure based on gradient similarity, allowing related tasks to share lower-level adapters while diverging into separate branches for dissimilar tasks. Unlike methods that enforce global orthogonality, TreeLoRA achieves soft task separation via a bandit-based search algorithm and sparse gradient updates, reducing search complexity to $\mathcal{O}(1)$ per round while maintaining parameter efficiency. Kaushik et al. propose a method that addresses continual learning for large pretrained models by learning and dynamically updating a single shared low-rank subspace across tasks. Instead of storing separate adapters, the method enables all tasks to reuse a set of foundational basis vectors, eliminating catastrophic forgetting through SVD-based initialization and incremental merging of new knowledge.

3 METHOD

LLM continual learning is a cumulative parameter optimization process over a sequentially arriving stream of tasks, i.e., $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_K\}$, with the objective of maximizing composite performance across all previously encountered tasks. We decompose the forward process as $y = \mathcal{G}(\mathcal{R}(x))$, where $\mathcal{R}(.)$ is an intermediate representation and $\mathcal{G}(.)$ generates the final text. Thus, at the current task τ_j, the procedure minimizes $\mathcal{L}(\mathcal{G}(\mathcal{R}(x_j)), y_j)$ for $(x_j, y_j) \in D_{\tau_j}$, where $\mathcal{L}(.)$ is the task-specific loss and D is the dataset.

3.1 FORGETTING AND ITS EXPLANATION

To establish a theoretical foundation for the phenomenon of **Catastrophic Forgetting**, we directly analyze the dynamic evolution of outputs on previously learned tasks throughout the continual learning process over the task stream. Concretely, we characterize the output increment of an old task τ_i ($i \in [1, j-1]$) through a first-order differential formulation as:

$$\Delta y_i = \underbrace{\frac{\partial \mathcal{G}(\mathcal{R}(x_i;\theta);\phi)}{\partial \mathcal{R}_{x_i}} \Delta \mathcal{R}_{x_i}}_{\mathcal{R}-Increment} + \overbrace{\frac{\partial \mathcal{G}(\mathcal{R}(x_i;\theta);\phi)}{\partial \phi} \Delta \phi}^{\mathcal{G}-Increment}, \tag{1}$$

where θ and ϕ refer to the parameters of the intermediate representation $\mathcal{R}$ and the subsequent generation process $\mathcal{G}$, respectively. $\mathcal{R}_{x_i}$ corresponds to the intermediate representation on the input x_i of task τ_i. In the formulation presented above, the $\mathcal{R}$-increment and the $\mathcal{G}$-increment jointly engender

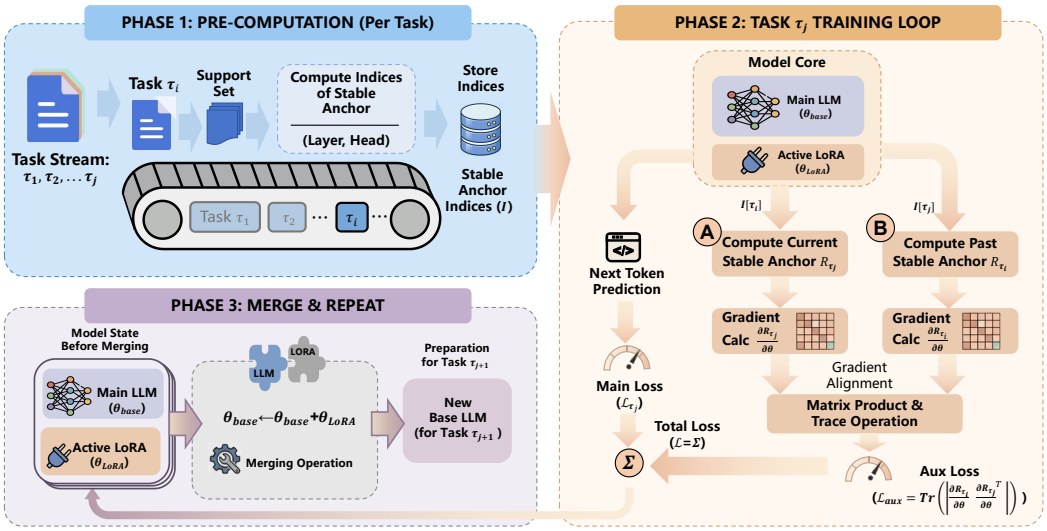

Figure 2: **Stable Anchor-Based Continual Learning:** *Phase 1* For each task, compute the indices of its corresponding stable anchor and store them. *Phase 2* Retrieving the stored stable anchor indices, compute the stable anchor representations for both the current and previous tasks, and calculate the auxiliary loss function according to Equation 7; the main loss function is the objective of the current task. *Phase 3* Update the LoRA module based on the combined loss, then merge it to the main LLM, serving as the model to be trained for the next task.

the output variation on the old task. **A negligible magnitude of this variation, i.e., $|\Delta y_i| < \epsilon$, signals that the performance on the old task is faithfully preserved; otherwise, the phenomenon of forgetting manifests.** Between the two constituents, the $\mathcal{R}$-increment hinges exclusively upon the input, thereby admitting a lucid analysis through the lens of input representation. The $\mathcal{G}$-increment, by contrast, subsumes a recursive decoding process, a consequence of the autoregressive property inherent to LLMs, which renders its treatment substantially more intricate.

3.2 $\mathcal{R}$-Increment Analysis

To facilitate a more tractable analysis of the output variation Δy_i, we impose a constraint on the representation $\mathcal{R}$, stipulating that $|\frac{\partial \mathcal{G}(\mathcal{R}(x_i;\theta);\phi)}{\partial \phi}|_F = 0$ ($|.|_F$ measures the norm of the matrix). Under this condition, the output variation is exclusively governed by $\mathcal{R}$. We term such a representation a **Stable Anchor**, formally defined as:

Definition *For the mapping $\mathcal{G}$ above and in Equation 1, if the condition holds universally across all parameter configurations (for ϕ) that, whenever the same representation $\mathcal{R}$ is taken as the point of departure, the resulting output y remains strictly invariant, then such an $\mathcal{R}$ is designated a Stable Anchor of the LLM.*

$$\mathcal{G}(\mathcal{R}(x);\phi_1) = \mathcal{G}(\mathcal{R}(x);\phi_2),\ s.t.\ \forall \phi_1 \neq \phi_2. \tag{2}$$

The property delineated above reveals that, irrespective of the training stage, so long as the representation $\mathcal{R}$ of the input remains invariant, the ultimate output stays unchanged.

Function Vector: As is proposed in Todd et al. (2024); Jiang et al. (2025), function Vector is derived via causal mediation analysis. Specifically, the mean activation per attention head is computed over a task-specific support set, and, using an inverse prompt as the base, each head's activation is substituted with its mean. The resulting shift in the probability of correct output quantifies that head's causal contribution, and the mean activations of the top-k contributive heads are then assembled to form the *Function Vector* for the task.

• **Step 1 Mean Activation:** For a support set S_{τ_i} of task τ_i, layer l, and head k, the mean last-token activation is $\hat{h}_{lk} = \sum_{(x,y)\in S_{\tau_i}} h_{lk}/|S_{\tau_i}|$.

• **Step 2 Inverse Prompt:** By injecting this prompt, i.e., ω, the model is induced to yield a semantically inverted output.

• **Step 3 Cause Effect:** For each layer and head, the original activation under the inverse prompt is replaced with the mean activation from Step 1; the probability changes $p_{h_{lk}\to\hat{h}_{lk}}(y|\omega,x) - p(y|\omega,x)$ are sorted over all (l,k) to select the top-k, where $p(.) = \mathcal{G}(\mathcal{R}(.))$.

• **Phenomenon:** In the paper Jiang et al. (2025), it is demonstrated that when the function vector of an old task is substituted for the corresponding parameters in the subsequently trained model, the performance on that old task is rehabilitated to an appreciable extent.

Theorem 1 *The function vector serves as an approximate instantiation of a Stable Anchor representation. This entails that, as training proceeds, its contribution to the $\mathcal{G}$-increment is negligible, rendering the overall output variation governed almost exclusively by the $\mathcal{R}$-increment.*

Proof: In the causal-effect score above, $p(y|\omega,x)$ is the probability of producing the correct output under the inverse prompt and is therefore negligibly small (almost zero). Since $\hat{h}_{lk}$ is the activation of the standard input without the inverse prompt, $p_{h_{lk}\to\hat{h}_{lk}}(y|\omega,x) = \mathcal{G}(\mathcal{R}(x);\omega,x)$; thus, $\mathcal{R}$ **hinges exclusively on x, while $\mathcal{G}$ involves both ω and x.** Top-k selection maximizes the probability of generating the correct output, so it is empirically desirable that $|\mathcal{G}(\mathcal{R}(x);\omega,x) - \mathcal{G}(\mathcal{R}(x);x)| < \epsilon$. In other words, the function vector, serving as the intermediate representation, i.e., $\mathcal{R}$, yields closely aligned outputs across different parameterizations of $\mathcal{G}$ ((ω,x)-dependent VS. x-dependent)). Owing to the diversity of forms that ω may assume, this property holds across a broad spectrum of parameter variations in $\mathcal{G}$. (Likewise, the *phenomenon* delineated above serves as a direct substantiation of the validity of the theorem.)

$\mathcal{R}$-Increment Decomposition: With the function vector serving as $\mathcal{R}$, we can now devote our attention to a granular analysis of the $\mathcal{R}$-increment. More concretely, during training on a new task τ_j, the incremental rate of change of the old task, i.e., τ_i, over $\mathcal{R}$ is given by:

$$v_{x_i} := \frac{\Delta\mathcal{R}_{x_i}}{\Delta t} = -\mathbb{E}_{(x_j,.)\sim D_{\tau_j}}[(\frac{\partial\mathcal{R}_{x_i}}{\partial\theta}\frac{\partial\mathcal{R}_{x_j}}{\partial\theta}^T)\frac{\partial\mathcal{L}_{\tau_j}}{\partial\mathcal{R}_{x_j}}^T],$$
$$s.t. \quad v_{x_i}, \mathcal{R}_{x_i}, \mathcal{R}_{x_j} \in R^m, \theta \in R^n, \tag{3}$$

where x_i and x_j denote the data of tasks τ_i and τ_j, respectively; D_{τ_j} represents the training dataset of τ_j; $\mathcal{L}_{\tau_j}$ signifies the loss function employed for training τ_j. m and n refer to the dimensionalities of $\mathcal{R}$ and its parameter space ($m \ll n$). T is the matrix transposition. **Here t means the training step.**

Proof: By virtue of the chain rule in conjunction with the gradient-based update ($\frac{d\theta}{dt} = \theta_{t+1} - \theta_t = -\mathbb{E}[\nabla_\theta\mathcal{L}_{\tau_j}]$), we decompose v_{x_i} as:

$$v_{x_i} = -\mathbb{E}[\frac{\partial\mathcal{R}_{x_i}}{\partial\theta}(\nabla_\theta\mathcal{L}_{\tau_j})^T] = -\mathbb{E}[\frac{\partial\mathcal{R}_{x_i}}{\partial\theta}(\frac{\partial\mathcal{L}_{x_j}}{\partial\mathcal{R}_{x_j}}$$
$$\frac{\partial\mathcal{R}_{x_j}}{\partial\theta})^T] = -\mathbb{E}[(\frac{\partial\mathcal{R}_{x_i}}{\partial\theta}\frac{\partial\mathcal{R}_{x_j}}{\partial\theta}^T)\frac{\partial\mathcal{L}_{\tau_j}}{\partial\mathcal{R}_{x_j}}^T]. \tag{4}$$

Equation 3 provides an explicit formulation of how the training of a new task τ_j exerts influence on the old task τ_i—namely, through update in the stable anchor parameter space, which in turn induce variations in the old-task output.

Theorem 2 *To mitigate forgetting in continual learning, i.e., to render the output variation of old tasks (Δy_i in Equation 1) negligible during new-task training, we may minimize the rate of change (v_{x_i} in Equation 3) evaluated on the old task's stable anchor (function vector in the paper). This, in turn, is tantamount to minimizing the norm of the product of the partial derivatives of the old and new task stable anchors with respect to the parameter space, i.e., $|\frac{\partial\mathcal{R}_{x_i}}{\partial\theta}\frac{\partial\mathcal{R}_{x_j}}{\partial\theta}^T|_F$.*

Hence, as the theorem indicates, alleviating forgetting in continual learning is equivalent to enforcing orthogonality between the gradients of the old- and new-task stable anchors with respect to the parameter space.

3.3 CONTINUAL LEARNING OPTIMIZATION

To further optimize the F-norm of the first-order partial derivatives formulated above, we invoke the **Schur Inequality** Ikramov (1994); Zhang (2005), which yields the following lower bound on the norm:

$$|\frac{\partial \mathcal{R}_{x_i}}{\partial \theta}\frac{\partial \mathcal{R}_{x_j}}{\partial \theta}^T|_F = \sqrt{\sum_{u,v}^{m}|c_{uv}|^2} \geq \sqrt{\sum_{u}^{m}|\lambda_u|^2}, \tag{5}$$

where c_{uv} represents the element of the matrix, i.e., $\frac{\partial \mathcal{R}_{x_i}}{\partial \theta}\frac{\partial \mathcal{R}_{x_j}}{\partial \theta}^T \in R^{m\times m}$ (the dimensionality m can be derived from Equation 3), and λ_u denotes the eigenvalue. Furthermore, the lower bound admits a further reduction through the **Cauchy Inequality**, resulting in:

$$\sqrt{\sum_{u}^{m}|\lambda_u|^2} \geq \frac{1}{\sqrt{m}}\sum_{u}^{m}|\lambda_u| = \frac{1}{\sqrt{m}}\,\mathrm{tr}(|\frac{\partial \mathcal{R}_{x_i}}{\partial \theta}\frac{\partial \mathcal{R}_{x_j}}{\partial \theta}^T|), \tag{6}$$

where $\mathrm{tr}(.)$ denotes the matrix trace operation.

Equations 5 and 6 establish the optimization objective of minimizing the norm of the matrix product of the old- and new-task stable anchor partial derivatives with respect to the parameter space. Its lower bound admits a trace representation, which somewhat alleviates the computational complexity, from $\mathcal{O}(m^2n)$ to $\mathcal{O}(mn)$ (the rationale for our choice to optimize using the lower bound is detailed in Section A).

Consequently, in its entirety, the ultimate optimization objective for continual learning, when training on task τ_j, can be formulated as:

$$\mathcal{L} = \mathcal{L}_{\tau_j} + \frac{1}{(j-1)\sqrt{m}}\sum_{i=1}^{j-1}\mathrm{tr}(|\frac{\partial \mathcal{R}_{x_i}}{\partial \theta}\frac{\partial \mathcal{R}_{x_j}}{\partial \theta}^T|), \tag{7}$$

where $\mathcal{L}_{\tau_j}$ means the original loss function on task τ_j, i.e., the next token prediction loss (Referring to Equation 3). **A comprehensive account of the LoRA-based training procedure is provided in Appendix B.**

The detailed LLM continual learning algorithm is presented in Algorithm 1.

4 EXPERIMENT

4.1 TASK SELECTION AND METRICS

To systematically evaluate continual instruction tuning, we construct task sequences from the SuperNI benchmark Wang et al. (2022).[1] Following prior work Jiang et al. (2025), we design three types of task sequence to capture different learning challenges: Classification sequence NI-Seq-C, Generation sequence NI-Seq-G and Mixed sequence NI-Seq-M.

Three widely adopted continual learning metrics are used to quantify performance and forgetting:

- **Average Accuracy (AA)**: the average test accuracy after the final task N, $\mathrm{AA}_N = N^{-1}\sum_{i=1}^{N} f_i(\mathbf{w}_N)$, where $f_i(\mathbf{w}_N)$ denotes performance (e.g., Rouge-L or accuracy) on task i.
- **Backward Transfer (BWT)**: measures forgetting after learning new tasks, $\mathrm{BWT}_N = N^{-1}\sum_{i=1}^{N}(f_i(\mathbf{w}_i) - f_i(\mathbf{w}_N))$. Positive values indicate forgetting; negative values indicate improvement.
- **Forgetting Rate (F.Ra)**: the average decline from each task's peak to its final performance, $\mathrm{F.Ra}_N = N^{-1}\sum_{i=1}^{N}\max_{j\leq N}(f_i(\mathbf{w}_j) - f_i(\mathbf{w}_N))$.

These metrics together provide a comprehensive view of both the model's ability to learn new tasks and its resistance to catastrophic forgetting.

[1] SuperNI is a collection of NLP tasks with expert-written instructions, widely used for assessing cross-task generalization.

Table 1: Main results of our method comparing against Vanilla, Replay, O-LoRA, and Tree-LoRA on three SuperNI continual learning benchmarks using Qwen3-8B and Llama3.1-8B. We report average accuracy (AA,%, higher better), backward transfer (BWT, %, lower better), and forgetting rate (F.Ra, %, lower better).

Base Model	Methods	NI-Seq-C			NI-Seq-G			NI-Seq-M		
		AA	BWT	F.Ra	AA	BWT	F.Ra	AA	BWT	F.Ra
Qwen3-8B	SFT	82.51	-	-	20.45	-	-	58.54	-	-
	vanilla	81.97	0.68	0.68	16.85	4.50	4.61	58.39	1.83	2.78
	Replay	**83.76**	0.62	1.31	**23.57**	5.08	5.09	55.96	1.75	1.75
	O-LoRA	71.95	13.35	13.35	16.61	4.54	4.65	53.17	4.61	4.61
	Tree-LoRA	71.08	13.17	13.17	16.92	**3.03**	3.36	54.46	2.72	2.72
	Our	80.64	**-0.50**	**-0.01**	22.93	3.07	**3.07**	**61.06**	**1.18**	**1.18**
Llama3.1-8B	SFT	82.86	-	-	21.61	-	-	61.38	-	-
	vanilla	80.36	3.13	3.13	17.05	5.69	5.69	56.98	5.51	5.511
	Replay	82.28	2.21	2.21	22.65	4.93	4.93	53.91	4.89	4.89
	O-LoRA	59.15	9.67	10.11	22.09	7.69	7.69	39.13	15.24	17.86
	Tree-LoRA	73.42	12.42	12.42	16.58	**4.66**	**4.66**	50.76	5.33	5.33
	Our	**83.70**	**0.94**	**0.94**	**25.40**	4.91	4.91	**61.49**	**2.23**	**2.23**

4.2 Baselines and Models

We conduct experiments on two representative large language models (LLMs): **Qwen3-8B** Yang et al. (2025) and **Llama3.1-8B** Dubey et al. (2024). Both models have around 8 billion parameters and are widely used as base models for continual instruction tuning due to their strong general capabilities and efficient deployment characteristics.

To evaluate the effectiveness of stable anchor continuous learning, we compare the following methods:

- **Vanilla (Prompt Learning)**: We design interrelated system prompts across tasks to maximize the similarity among tasks within the stream, and train these tasks sequentially through a standard SFT procedure. Owing to the aforementioned relatedness, the forgetting phenomenon is mitigated to a certain degree.
- **Replay-based**: A representative rehearsal-based continual learning method that stores a subset of previous training samples and interleaves them with new task data during fine-tuning to mitigate catastrophic forgetting Mok et al. (2023); Huang et al. (2024); Feng et al. (2026).
- **O-LoRA** Wang et al. (2023): An orthogonal subspace learning method that fine-tunes a new LoRA adapter for each task while projecting gradients into subspaces orthogonal to previous tasks, thereby reducing interference.

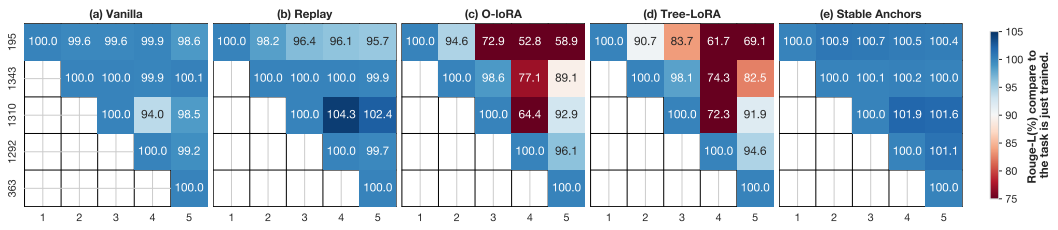

Figure 3: Heatmaps of performance evolution for five methods in NI-Seq-C (5 tasks × 5 stages) on Qwen3-8B. The color denotes the ROUGE-L score (%) of the model's current performance relative to the performance immediately after completing training on the corresponding task dataset.

- **Tree-LoRA** Qian et al. (2025): A hierarchical approach that organizes LoRA adapters into a K-D tree Skrodzki (2019) based on gradient similarity of training tasks. A bandit algorithm (Lower Confidence Bound) is used to efficiently select the most relevant branch for each new task, and sparse gradient updates are applied to preserve past knowledge.

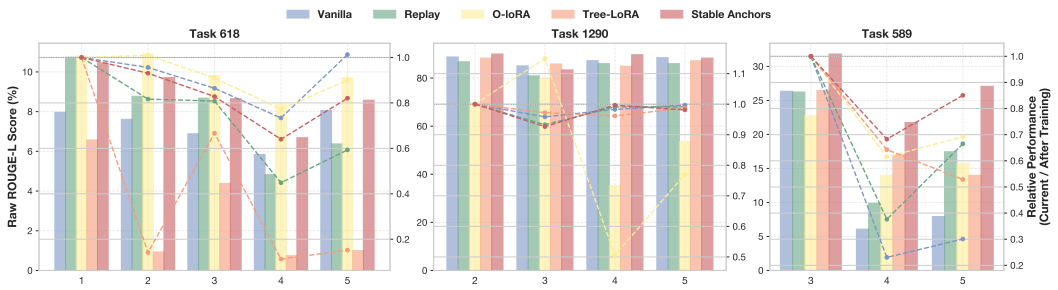

Figure 4: Absolute (bars, left axis) and relative (lines, right axis) performance on three tasks of NI-Seq-M. The relative performance is the current ROUGE-L score divided by the score right after the task's own training.

4.3 Main Results

Classification Tasks (NI-Seq-C). On Llama3.1-8B, Our method achieves the highest AA of 83.70%, with BWT and F.Ra both equal to 0.94%, this result substantially outperforms Vanilla as well as Replay. On Qwen3-8B, Our method yields an AA of 80.64%, slightly lower than Vanilla's 81.97%, but it exhibits a negative BWT of -0.50%, indicating that learning new tasks actually improves the accuracy on previously learned classification tasks. Such positive backward transfer arises from the consistency loss, which anchors task-specific activation patterns while still allowing beneficial feature reuse from subsequent tasks.

Generation Tasks (NI-Seq-G). Generation is more challenging; all methods achieve AA around 20%. Our method yields the largest gains: on Llama3.1-8B, AA improves from 17.05% (Vanilla) to 25.40% (+8.35%); on Qwen3-8B, from 16.85% to 22.93% (+6.08%). BWT remains competitive, demonstrating that stable anchors preserve generative skills better than sample replay.

Mixed Tasks (NI-Seq-M). Mixed sequences require balancing classification and generation. Our method achieves best AA on both models (61.06% on Qwen, 61.49% on Llama) while keeping low BWT (1.18% and 2.23%). Vanilla suffers from either low plasticity or high forgetting.

Model Dependency. Both Qwen3-8B and Llama3.1-8B benefit from Stable Anchors. However, overall comparisons reveal that Qwen-series models exhibit lower forgetting rates and better BWT scores, whereas Llama-series models achieve higher AA. Specifically, on Qwen3-8B the forgetting is less severe and backward transfer is stronger, while on Llama3.1-8B the average accuracy benefits are more pronounced.

F.Ra and BWT It is also worth noting that in most cases *F.Ra* equals *BWT*, indicating that most algorithms achieve their optimal performance on a given task at the time it is first trained, and do not improve during subsequent training steps. Experimental results further reveal that the Qwen series models exhibit a stronger tendency than the Llama series models to enhance the performance of earlier tasks when learning later tasks.

4.4 Further Analysis

Figure 3 shows forgetting levels via heatmaps: values below 100% indicate significant forgetting, while above 100% indicate positive backward transfer, revealing the method's ability to leverage inter-task relationships. Figure 4 uses a mixed bar-line plot to present absolute and relative performance for tasks 618, 1290, and 589 in NI-Seq-M. Higher bars and more stable lines reflect better accuracy and less forgetting.

Observation 1: For Stable Anchor, the performance variation across tasks throughout the entire task sequence is much smaller. In contrast, Tree-LoRA and O-LoRA exhibit substantial forgetting at the third and fourth steps. Stable Anchor shows no forgetting during training, demonstrating its strong stability on classification tasks.

Table 2: Ablation study of our method on three SuperNI dataset using Qwen3-8B and Llama3.1-8B.

Base Model	Methods	NI-Seq-C			NI-Seq-G			NI-Seq-M		
		AA	BWT	F.Ra	AA	BWT	F.Ra	AA	BWT	F.Ra
Qwen3-8B	**Our**	**80.64**	**-0.50**	**-0.01**	**22.93**	**3.07**	**3.07**	**61.06**	**1.18**	**1.18**
	w/o stable anchor	79.80	3.09	3.09	22.11	3.33	3.33	56.26	3.19	3.19
Llama3.1-8B	**Our**	**83.70**	**0.94**	**0.94**	**25.40**	**4.91**	**4.91**	**61.49**	**2.23**	**1.79**
	w/o stable anchor	80.11	3.62	3.64	23.53	6.78	6.78	56.92	6.36	6.78

Observation 2: Stable Anchor achieves higher final performance on a given task than its performance right after the first training on that task. Compared with orthogonal-parameter-space methods(i.e., O-LoRA), our method exhibits a superior ability to discover connections among similar tasks.

Observation 3: Stable Anchor shows minor disadvantages in AA or BWT relative to other methods on specific tasks. Taking the generative task sequence NI-Seq-G as an example, on Llama3.1-8B, the BWT of our method is 0.25% lower than that of Tree-LoRA, but the AA is significantly higher by 8.82%, indicating that our method better handles classification tasks. On Qwen3-8B, the AA of our method lags behind Replay by 0.64%, yet its BWT is 2.01% lower,which means less negative forgetting, demonstrating superior continual learning performance.

4.5 Ablation Study

To validate the efficacy of stable anchor selection, we replace the cause-effect-based top-k selection in the stable anchor index selection phase (i.e., the $\mathcal{R}$_Index function in Algorithm 1) with a random selection of k [layer, head] pairs, employing the resulting representations in lieu of the stable anchors:

$$\mathcal{I}[\tau_i] = \text{sample}([layer, head], k). \tag{8}$$

As shown in Table 2, the randomly selected [layer, head] indices (**w/o stable anchor** lines) precipitate a marked degradation in overall task performance (**AA**), with an average decline of 2.15% across the three task streams for the Qwen model and 2.68% for the Llama model. Concomitantly, this random selection strategy engenders inferior outcomes on forgetting-related metrics. More precisely, the Backward Transfer metric (**BWT**) registers an average deterioration of 1.95 for Qwen and 2.89 for Llama, while the Forgetting Rate (**F.Ra**) exhibits a worsening of 1.79 and 3.19, respectively.

This performance deterioration accords precisely with the foundational rationale underpinning our stable anchor design. The stable anchor is conceived to furnish representations inherently disentangled from the generation process ($\mathcal{G}(.)$ in Equation 1); consequently, the cause-effect-driven top-k selection constitutes a principled instantiation of this theoretical postulate. Random index selection, by contrast, forfeits this generation-agnostic property, and the resulting representations inevitably precipitate the observed degradation.

4.6 Training Time Consumption

In our experiments, baseline methods such as O-LoRA and Tree-LoRA are all efficient approaches. Under the 8B model setting, these methods achieve an average training time of approximately 83% of that of our method across the three task streams. The computation of Stable Anchors for different tasks does consume some time, accounting for approximately 5% of the training overhead. However, it is worth noting that this computation only needs to be performed once.

5 Conclusion

By analyzing the phenomenon of forgetting during the continual learning process of LLMs, this paper proposes a theoretical characterization termed the stable anchor, which attributes forgetting to the perturbation of the old-task stable anchor induced by parameter-space modifications during new-task learning. Building upon this theoretical framework, we introduce a loss function designed to mitigate such interference, and empirically validate the effectiveness of both the theory and the proposed method across multiple task streams.

AI USE STATEMENT

In this paper, AI tools are used solely for language polishing, such as refining word choices and expressions, as well as checking and adjusting redundant descriptions.

ETHICS STATEMENT

This paper contains no potentially harmful content, including human subjects, harmful insights or applications, conflicts of interest, discrimination or bias concerns, privacy and security issues, legal compliance, or research integrity issues, etc.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we provide the detailed training procedure in Algorithm 1, with the corresponding training hyperparameters specified in Section B. Our training pipeline is built upon the standard LLaMA-Factory library, where we replace the original SFT loss with our designed loss function.

REFERENCES

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aur'elien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany M. Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel J. Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, E I Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Kenneth Heafield, Kevin R. Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuen ley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Ma hesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Niko lay Bashlykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ron nie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sa hana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stéphane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whit ney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld,

Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Po-Yao (Bernie) Huang, Beth Loyd, Beto de Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco (Paco) Guzmán, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthias Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Mun ish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navy ata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pe dro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Ro han Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, S. Yu. Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Kumar Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria O Ajayi, Victoria Montanez, Vijai Mohan, Vinay Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models. 2024. URL https://api.semanticscholar.org/CorpusID:271571434.

Yujie Feng, Hao Wang, Jian Li, Xu Chu, Zhaolu Kang, Yiran Liu, Yasha Wang, Philip S. Yu, and Xiao-Ming Wu. Forever: Forgetting curve-inspired memory replay for language model continual learning. *ArXiv*, abs/2601.03938, 2026. URL https://api.semanticscholar.org/CorpusID:284532443.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International*

Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1416–1428, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.77. URL https://aclanthology.org/2024.acl-long.77/.

Khakim D. Ikramov. A simple proof of the generalized schur inequality. *Linear Algebra and its Applications*, 199:143–149, 1994. ISSN 0024-3795. doi: https://doi.org/10.1016/0024-3795(94)90346-8. URL https://www.sciencedirect.com/science/article/pii/0024379594903468. Special Issue Honoring Ingram Olkin.

Olaf Yunus Laitinen Imanov. Mechanistic analysis of catastrophic forgetting in large language models during continual fine-tuning. *ArXiv*, abs/2601.18699, 2026. URL https://api.semanticscholar.org/CorpusID:285051382.

Gangwei Jiang, Caigao JIANG, Zhaoyi Li, Siqiao Xue, JUN ZHOU, Linqi Song, Defu Lian, and Ying Wei. Unlocking the power of function vectors for characterizing and mitigating catastrophic forgetting in continual instruction tuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gc8QAQfXv6.

Jiazheng Kang, Le Huang, Cheng Hou, Zhe Zhao, Zhenxiang Yan, and Ting Bai. Self-evolving llms via continual instruction tuning. *Proceedings of the ACM Web Conference 2026*, 2025. URL https://api.semanticscholar.org/CorpusID:281496330.

Prakhar Kaushik, Ankit Vaidya, Shravan Chaudhari, Ramalingam Chellappa, and Alan L. Yuille. Shared lora subspaces for almost strict continual learning. *ArXiv*, abs/2602.06043, 2026. URL https://api.semanticscholar.org/CorpusID:285303524.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23638–23647, 2024. doi: 10.1109/CVPR52733.2024.02231.

Yan-Shuo Liang, Jiarui Chen, and Wu-Jun Li. Gated integration of low-rank adaptation for continual learning of large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=lVV7F0piDK.

Jessy Lin, Luke S. Zettlemoyer, Gargi Ghosh, Wen tau Yih, Aram H. Markosyan, Vincent-Pierre Berges, and Barlas Ouguz. Continual learning via sparse memory finetuning. *ArXiv*, abs/2510.15103, 2025. URL https://api.semanticscholar.org/CorpusID:282203348.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Beyond not-forgetting: Continual learning with backward knowledge transfer. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 16165–16177. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6728fcf94660c59c938319a6833a6073-Paper-Conference.pdf.

Mao-Lin Luo, Zi-Hao Zhou, Yi-Lin Zhang, Yuanyu Wan, Min-Ling Zhang, and Tong Wei. KeeploRA: Continual learning with residual gradient adaptation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=T3Vc5fkTzV.

Jisoo Mok, Jaeyoung Do, Sungjin Lee, Tara Taghavi, Seunghak Yu, and Sungroh Yoon. Large-scale lifelong learning of in-context instructions and how to tackle it. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12573–12589, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.703. URL https://aclanthology.org/2023.acl-long.703/.

Yu-Yang Qian, Yuan-Ze Xu, Zhen-Yu Zhang, Peng Zhao, and Zhi-Hua Zhou. TreeloRA: Efficient continual learning via layer-wise loRAs guided by a hierarchical gradient-similarity tree. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=f6ibJCQfH4.

Fuli Qiao and Mehrdad Mahdavi. Merge before forget: A single loRA continual learning via continual merging. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=i1Rj7yU6eF.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In Rajesh Gupta 0001, Yan Liu 0002, Jiliang Tang, and B. Aditya Prakash (eds.), *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 3505–3506. ACM, 2020. ISBN 978-1-4503-7998-4. URL https://dl.acm.org/doi/10.1145/3394486.3406703.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Comput. Surv.*, 58(5), November 2025. ISSN 0360-0300. doi: 10.1145/3735633. URL https://doi.org/10.1145/3735633.

Martin Skrodzki. The k-d tree data structure and a proof for neighborhood computation in expected logarithmic time. *ArXiv*, abs/1903.04936, 2019. URL https://api.semanticscholar.org/CorpusID:75135481.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024. doi: 10.1109/TPAMI.2024.3367329.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.715. URL https://aclanthology.org/2023.findings-emnlp.715/.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, M. Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://api.semanticscholar.org/CorpusID:253098274.

Zitao Wang, Xinyi Wang, and Wei Hu. Mixture of LoRA experts for continual information extraction with LLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 13324–13339, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.718. URL https://aclanthology.org/2025.findings-emnlp.718/.

Buddhi Wickramasinghe, Gobinda Saha, and Kaushik Roy. Continual learning: A review of techniques, challenges, and future directions. *IEEE Transactions on Artificial Intelligence*, 5(6): 2526–2546, 2024. doi: 10.1109/TAI.2023.3339091.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of loRA experts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uWvKBCYh4S.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Jingren Zhou, Junyan Lin, Kai Dang, Keqin Bao, Ke-Pei Yang, Le Yu, Li-Chun Deng, Mei Li, Min Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shi-Qiang Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. 2025. URL https://api.semanticscholar.org/CorpusID:278602855.

Fuzhen Zhang. The schur complement and its applications. 2005. URL https://api.semanticscholar.org/CorpusID:118854172.

Junhao Zheng, Shengjie Qiu, and Qianli Ma. Learn or recall? revisiting incremental learning with pre-trained language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14848–14877, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.794. URL https://aclanthology.org/2024.acl-long.794/.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. Spurious forgetting in continual learning of language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 67081–67146, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a774503daed55eb53c634847ae071ec7-Paper-Conference.pdf.

A THE OPTIMIZATION EXPLANATION

Although in Equation 5 we obtain a lower bound of the original F-norm to approximate the overall optimization objective, in practice, minimizing the F-norm generally requires minimizing the sum of squares over all matrix elements $\sum c_{uv}$, which implies that the sums of squares of the diagonal elements and the off-diagonal elements are minimized separately. We denote the sum of squares of diagonal elements as $d = \sum c_{uu}^2$ and that of off-diagonal elements as $m = \sum c_{uv}^2 (u \neq v)$. Considering that the diagonal elements represent products of partial derivatives of $\mathcal{R}_{x_i}$ and $\mathcal{R}_{x_j}$ with respect to the same parameter, while off-diagonal elements represent products with respect to different parameters, and given that in general the different parameters of an LLM exhibit a certain degree of orthogonality, the diagonal elements tend to be significantly larger than the off-diagonal elements.

Therefore, we can naturally assume $m \leq k * d$, where k is a bounded coefficient. On this basis, we can actually derive that the upper bound of the F-norm should be $(k+1)d$.

Moreover, in terms of the optimization objective, in the ideal final case, all elements should approach zero to ensure norm minimization; thus, $(k+1)$tr$(.)$ is actually close to the trace norm. Consequently, the final upper bound can be approximated as .

In summary, both the lower and upper bounds of the F-norm are functions of the trace norm, which implies that even though we only use the lower bound to approximate the optimization objective, their trends of change are ultimately consistent, and the situation where the lower bound is not equivalent to the original objective—as generally understood—does not arise.

B TRAINING DETAILS

All LoRA-based methods use a rank of $r = 32$, scaling factor $\alpha = 64$ and target the query and value projection matrices. For O-LoRA and Tree-LoRA, we follow their original hyperparameter settings. Training is performed with a learning rate of 1×10^{-4}, each task uses different epochs based on training loss, using mixed-precision and DeepSpeed ZeRO-2 Rasley et al. (2020) for memory efficiency.

C RESULTS DETAILS

Table 3 presents more detailed experimental results of our method on the three continual learning task sequences of SuperNI. The rows correspond to task names, while the columns indicate the training order of tasks. All results in Table 1 are computed based on similar experimental logs.

D LIMITATIONS

The core contribution of our paper lies in introducing a theoretically grounded framework—termed the stable anchor—for interpreting the forgetting phenomenon in continual learning. Within this framework, forgetting is characterized as the interference between stable anchors of old and new tasks propagated through the parameter space. Our theory, together with the derived loss function designed to mitigate such interference, effectively alleviates forgetting and preserves the performance of previously learned tasks. Currently, computing this loss involves second-order derivatives with respect to the parameter space, which incurs a manageable increase in computational complexity relative to the standard next-token prediction objective. As a natural avenue for future work, we plan to refine the computation of this loss by leveraging first-order approximations, thereby pursuing a more computationally efficient learning procedure while preserving the theoretical underpinnings of the stable anchor framework.

Algorithm 1: LLM Continual Learning

Input : $\mathcal{T}$, D, S, $\mathcal{C}$, $\mathcal{M}_b$, ω
Output : $\mathcal{M} \longleftarrow \mathcal{M}_b$

DEF `LoRA_Init` $(\mathcal{M}_o)$**:**
  return $\mathcal{M}_o$ **+ LoRA()**
DEF $\mathcal{R}$`_Index` $(\mathcal{M}_o, \mathcal{D}_a, \omega)$**:**
  Partition of Support Data: $\mathcal{D}_a = \mathcal{D}_a^M \cup \mathcal{D}_a^C, \mathcal{D}_a^M \cap \mathcal{D}_a^C = \emptyset$
  Mean Activation Derivation: $\hat{h}_{lk} = \hat{\mathcal{H}}[l,k] = \vec{0}_m$ `// m-dim vector`
  for $(x, .) \in \mathcal{D}_a^M$ **do**
    With hook $\mathcal{M}_o$ as $\mathcal{H}$: $h_{lk} = \mathcal{H}[l,k]$, $\hat{h}_{lk} \longleftarrow \hat{h}_{lk} + h_{lk}$
  end for
  $\hat{h}_{lk} \longleftarrow \hat{h}_{lk}/|\mathcal{D}_a^M|$
  Cause Effect Calculation: $\hat{e}_{lk} = \mathcal{E}[l,k] = 0$
  for $(x, y) \in \mathcal{D}_a^C$ **do**
    $e_{lk} = p_{h_{lk} \to \hat{h}_{lk}}(y|\omega, x) - p(y|\omega, x)$,$\hat{e}_{lk} \longleftarrow \hat{e}_{lk} + e_{lk}$
  end for
  $\hat{e}_{lk} \longleftarrow \hat{e}_{lk}/|\mathcal{D}_a^C|$, $inds =$ index(sort($\mathcal{E}$))[: k] `// Top-k` $[l,k]$ `Selection`
  return $inds$
DEF $\mathcal{R}$`_Rep` $(\mathcal{M}_o, \mathcal{D}_a, \mathcal{I}_n)$**:**
  `// Get` $\mathcal{R}$ `representation,` m`-dim`
  $\hat{h}_{lk} = \hat{\mathcal{H}}[l,k] = \vec{0}_m (\forall [l,k] \in \mathcal{I}_n)$
  for $(x, .) \in \mathcal{D}_a$ **do**
    With hook $\mathcal{M}_o$ as $\mathcal{H}$: $h_{lk} = \mathcal{H}[l,k]$, $\hat{h}_{lk} \longleftarrow \hat{h}_{lk} + h_{lk}$
  end for
  $\hat{h}_{lk} \longleftarrow \hat{h}_{lk}/|\mathcal{D}_a|$
  return mean($\hat{\mathcal{H}}, dim = (l,k)$)
1. $\mathcal{R}$ **Index Derivation ([layer, head])**
Initiate the $\mathcal{R}$ index dict: $\mathcal{I}$ = { }
for $(\tau_i, S_{\tau_i}) \in$ *zip*$(\mathcal{T}, S)$ **do**
  $\mathcal{I}[\tau_i]$ = $\mathcal{R}$_Index(S_{τ_i}, ω)
end for
2. Continual Learning Procedure
for $(\tau_j, D_{\tau_j}) \in$ *zip*$(\mathcal{T}, D)$ **do**
  $\mathcal{M}_b \longleftarrow$ LoRA_Init($\mathcal{M}_b$),$\mathcal{R}_{x_j}$ = $\mathcal{R}$_Rep($\mathcal{M}_b, D_{\tau_j}, \mathcal{I}[\tau_j]$)
  Initiate trace dict: $\mathcal{T}r$ = { }
  for $c_{\tau_i} \in \{c_{\tau_1}, c_{\tau_2}, ..., c_{\tau_{j-1}}\} (\subset \mathcal{C})$ **do**
    $\mathcal{R}_{x_i}$ = $\mathcal{R}$_Rep($\mathcal{M}_b, c_{\tau_j}, \mathcal{I}[\tau_i]$), $\mathcal{T}r[\tau_i] = \frac{1}{\sqrt{m}} \text{tr}(|\frac{\partial \mathcal{R}_{x_i}}{\partial \theta} \frac{\partial \mathcal{R}_{x_j}}{\partial \theta}^T|)$
  end for
  $\mathcal{L} = \mathcal{L}_{\tau_j}$ + mean($\mathcal{T}r$), **Update** $\mathcal{M}_b$
end for

Table 3: Detailed results of our method on three SuperNI task sequences with Qwen3-8B and Llama3.1-8B.

Model	Task Name	Performance (Rouge-L %)					BWT	F.Ra
		step1	step2	step3	step4	step5		
Qwen3-8B	task195	86.61	87.38	87.23	87.07	86.92	-0.31	0.46
	task1343		99.00	99.07	99.15	99	0.00	0.15
	task1310			56.25	57.33	57.17	-0.92	0.16
	task1292				72.25	73.03	-0.78	-0.78
	task363					87.54	–	–
	task618	45.09	40.6	30.39	39.23	35.72	9.37	9.37
	task1290		25.60	25.01	24.71	25.54	0.06	0.06
	task589			10.73	9.15	8.00	2.73	2.73
	task511				19.73	19.63	0.10	0.10
	task1357					25.74	–	–
	task360	10.35	9.44	7.11	7.55	7.36	2.99	2.99
	task363		89.84	89.23	88.61	88.92	0.92	0.92
	task1290			29.28	28.56	28.39	0.88	0.88
	task339				80.88	80.94	-0.06	-0.06
	task1510					99.69	–	–
Llama3.1-8B	task195	87.38	86.61	86.46	83.08	85.53	1.85	1.85
	task1343		99.84	99.61	89.07	99.61	0.23	0.23
	task1310			65.23	65.07	63.84	1.39	1.39
	task1292				80.30	80.00	0.30	0.30
	task363					89.53	–	–
	task618	46.46	38.50	16.96	32.19	30.84	15.62	15.62
	task1290		31.94	31.45	30.98	31.89	0.05	0.05
	task589			14.90	12.00	10.86	4.04	4.04
	task511				22.59	22.64	-0.06	-0.06
	task1357					30.79	–	–
	task360	10.45	9.72	8.65	6.69	8.57	1.88	1.88
	task363		90.00	83.38	89.69	88.31	1.69	1.69
	task1290			31.82	21.73	27.06	4.76	4.76
	task339				84.43	83.82	0.62	0.62
	task1510					99.69	–	–