ELLA: Efficient Lifelong Learning for Adapters in Large Language Models

Shristi Das Biswas[1,*] , Yue Zhang[2], Anwesan Pal[3], Radhika Bhargava[2], Kaushik Roy[1]

[1]Purdue University, [2]Amazon Web Services, [3]AWS AI Labs

(sdasbisw, kaushik)@purdue.edu

(zhangany, anwesanp, radhikb)@amazon.com

Abstract

Large Language Models (LLMs) suffer from severe catastrophic forgetting when adapted sequentially to new tasks in a continual learning (CL) setting. Existing approaches are fundamentally limited: replay-based methods are impractical and could potentially violate privacy, while strict orthogonality-based methods collapse under scale: each new task is projected onto an orthogonal complement, progressively reducing the residual degrees of freedom and eliminating forward transfer by forbidding overlap in shared representations. In this work, we introduce ELLA, a training framework built on the principle of selective subspace de-correlation. Rather than forbidding all overlap, ELLA explicitly characterizes the structure of past updates and penalizes alignments along their high-energy, task-specific directions, while preserving freedom in the low-energy residual subspaces to enable transfer. Formally, this is realized via a lightweight regularizer on a single aggregated update matrix. This mechanism is proven to be an anisotropic shrinkage operator that bounds interference, yielding a penalty that is both memory- and compute-constant regardless of task sequence length. ELLA requires no data replay, no architectural expansion, and negligible storage. Empirically, it achieves state-of-the-art CL performance on three popular benchmarks spanning both classification and generative tasks, with relative accuracy gains of up to 9.6% and a $35\times$ smaller memory footprint. Furthermore, ELLA scales robustly across architectures and actively enhances the model's zero-shot generalization performance on unseen tasks, establishing a principled and scalable solution for constructive lifelong LLM adaptation.[†]

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable generalization capabilities across a wide range of downstream tasks, largely attributed to their large-scale pretraining on diverse corpora (Brown et al., 2020; Touvron et al., 2023; Achiam et al., 2023). As LLMs are increasingly deployed in real-world applications, they must be continuously adapted to evolving user needs and task distributions for long-term practical deployment. This setting, commonly studied under the paradigm of continual learning (CL), requires models to acquire new knowledge sequentially without costly full-retraining (Ruvolo and Eaton, 2013). However, sequential finetuning of LLMs remains highly susceptible to catastrophic forgetting (CF) - the tendency to overwrite prior knowledge when new tasks are introduced (McCloskey and Cohen, 1989), and loss of plasticity - deterioration in the ability to learn new information over time (Dohare et al., 2021). These challenges are especially pronounced in rehearsal-free settings, where previously seen data cannot be stored due to privacy or storage constraints (Chaudhry et al., 2019).

A promising line of work leverages parameter-efficient fine-tuning (PEFT) strategies, such as Low-Rank Adaptation (LoRA) (Hu et al., 2022), to reduce the overhead of task-specific adaptation. Adapter-based approaches restrict updates to a small set of trainable parameters while leaving the base model frozen, making them a natural fit for lower-compute CL (Qin and Joty, 2021; Song et al., 2023). Yet, even with such modularity, sequential adapter training often yields severe forgetting when past knowledge is not revisited (Wang et al., 2024b). Prior efforts to address this include expanding model capacity (Wang et al., 2024a), isolating weights to reduce interference (Aljundi et al., 2017; Li et al., 2019; Wang et al., 2023c), subspace orthogonality (Wang et al., 2023a; Liao et al., 2025), or gradient projection from previous tasks (Qiao and Mahdavi, 2024). While effective to varying degrees, these often either limit forward knowledge transfer from past tasks, add storage overhead, or

---

overlook activation-level interference, where forgetting actually manifests (Ke et al., 2021).

A truly effective continual learning framework should balance knowledge retention with representation reuse, preventing harmful interference while allowing useful subspaces to be shared for new tasks (Wu et al., 2024). In practice, not all overlap is harmful – low-magnitude directions from past tasks may encode generic linguistic or semantic patterns that can accelerate learning on future tasks if safely reused. Unfortunately, existing CL methods either eliminate all overlap or fuse past knowledge through heavyweight mechanisms like controller networks or rank-conditioned fusion, which limits scalability and increases complexity (Zhao et al., 2024; Liu et al., 2023; Liao et al., 2025).

In this work, we propose Efficient Lifelong Learning for Adapters (ELLA), a simple and scalable CL framework for LLMs that mitigates forgetting without relying on replay buffers, parameter storage overheads or additional routing heuristics. ELLA introduces a subspace-aware regularization strategy that operates directly in weight space: we track the representational subspaces induced by past adapters and penalize updates that cause the new task's adapter to align too closely with them. This cross-task de-correlation simultaneously encourages task-specific specialization by preserving prior representational geometry to reduce interference, while permitting the reuse of low-magnitude directions to enable forward transfer. As we formally prove in Appendix A, this selective regularization corresponds to an anisotropic shrinkage operator that provably bounds interference, providing a theoretical foundation for ELLA's ability to balance stability and plasticity.

Through extensive experiments on classification tasks like the Standard CL Benchmark (Zhang et al., 2015) & Long Sequence Benchmark (Razdaibiedina et al., 2023), and generative tasks corresponding to the TRACE (Wang et al., 2023b) datasets, we demonstrate that ELLA achieves state-of-the-art performance on diverse classification and generative tasks while scaling effectively across model sizes (from $770M$ to $8B$). Furthermore, ELLA generalizes well across architecture families (e.g., T5 (Raffel et al., 2020), LLaMA (Touvron et al., 2023)), and unlike prior work, uniquely improves generalization to unseen tasks. ELLA does not rely on task identities during inference, and is hence naturally compatible with the instruction-tuning paradigm (Wang et al., 2022a), thereby preserving

the generalization capabilities of LLMs in zero-shot and open-ended settings. Moreover, ELLA is architecture-agnostic, requires no access to past data, and introduces negligible computational or memory overhead. In summary, our contributions are as follows:

- We propose ELLA, a replay-free plug-and-play CL framework for LLMs that balances plasticity and mitigates CF via subspace-aware regularization, and we provide a formal theoretical analysis of its properties.

- We provide extensive empirical evidence that ELLA establishes a new instruction-tuning state-of-the-art in both performance and efficiency across three popular CL benchmarks, decisively outperforming prior methods without task labels, replay, or additional overhead.

- We show that ELLA scales effectively across architectures and maintains strong generalization to unseen tasks, highlighting its practical advantages for real-world deployment.

2 Related Works

Continual Learning The primary challenge in CL is to balance model stability (resisting catastrophic forgetting) with plasticity (acquiring new knowledge). Prior approaches to address this problem can be broadly categorized by their core mechanism, each presenting a distinct set of trade-offs. **(i) Rehearsal-based methods** store a subset of past data and interleave it during subsequent training phases. These include strategies such as experience replay (Riemer et al., 2018) or constrained optimization (Aljundi et al., 2017; Chaudhry et al., 2019; He et al., 2024) that jointly train on current and previous samples. While empirically strong, this approach is often impractical due to significant storage overhead and potential violations of data privacy constraints. **(ii) Regularization-based methods** add penalty terms to the loss that restrict updates to parameters critical for old tasks (Du et al., 2024; Li and Hoiem, 2017; de Masson D'Autume et al., 2019). For instance, (Kirkpatrick et al., 2017) slows updates on important weights, while (Farajtabar et al., 2020) constrains new updates to be orthogonal to gradients of old tasks. These methods depend on importance metrics that are often brittle and prevent any forward transfer, limiting adaptability to new tasks. **(iii) Architecture-based methods** reduce

interference through task-specific modules or by expanding model size (Li et al., 2019; Wang et al., 2023c). While (Razdaibiedina et al., 2023) appends a new learned soft prompt per task, (Qin and Joty, 2021) utilizes a large soft prompt that is continuously trained on all tasks. While this effectively isolates task knowledge, it typically results in a linear growth in parameter count with the number of tasks, posing significant scalability challenges and often requiring explicit task labels at inference.

Parameter-Efficient Continual Learning Parameter-Efficient Fine-Tuning (PEFT) has emerged as a highly promising substrate for CL, with Low-Rank Adaptation (LoRA) (Hu et al., 2022) being a particularly effective technique. However, naïvely applying LoRA sequentially leads to significant forgetting owing to *interference across tasks*. As tasks arrive sequentially, the adapter weights A_t, B_t learned for a new task t are trained from scratch without awareness of previous tasks' LoRA subspaces. Consequently, overlapping update directions in parameter space may inadvertently override past knowledge, especially under limited parameter budgets. Without coordination across tasks, these LoRA components can destructively interfere in spaces crucial for previous tasks, degrading performance on earlier tasks while optimizing for the current one. Recent works in CL literature addressing this issue primarily falls into two schools of thought.

The first approach applies strict orthogonality to enforce zero overlap between the LoRA subspaces of different tasks (Wang et al., 2023a; Qiao and Mahdavi, 2024). While this hard constraint effectively mitigates interference, it imposes an overly restrictive inductive bias. By precluding any form of representation sharing, it severely inhibits beneficial forward transfer and prevents reuse of low-importance shared space across related tasks. Worse still, as tasks accumulate, the available orthogonal space quickly diminishes, leading to progressively less capacity for future learning. This rigid decoupling causes inefficient use of adapter capacity, hindering the goal of cumulative learning. Furthermore, storing all past LoRA weights incurs high memory overhead that scales linearly with the number of tasks, further limiting scalability.

The second approach involves more complex fusion and replay mechanisms. Methods like DATA (Liao et al., 2025) or Recurrent KIF (Feng et al., 2025) often rely on replay buffers, multiple LoRA adapters per task, or intricate gating mechanisms to learn how to compose adapters. These methods reintroduce significant architectural complexity and computational overhead, sacrificing the very simplicity and efficiency that make PEFT an attractive paradigm for CL in the first place.

Differently, ELLA charts a third path. The prevailing paradigms are caught in a dilemma: rehearsal is impractical, naïve regularization is ineffective, architectural expansion is inefficient, orthogonality is too restrictive, and complex fusion is costly. In contrast, ELLA abandons the assumption that all interference must be uniformly suppressed and introduces the more effective principle of managing interference selectively. By regularizing only the high-magnitude, task-specific directions of the adapter space while permitting the reuse of low-magnitude, generalizable features, ELLA achieves a superior balance of stability and plasticity without the trade-offs that limit prior work, and integrates seamlessly into LoRA-based workflows, enabling scalable and robust CL adaptation for LLMs.

3 Method

3.1 Problem Formulation

Setup. In the supervised CL setting, a model encounters a stream of tasks $\{1, \ldots, \mathcal{T}\}$ sequentially, where each task $t = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ of size n_t contains a labeled dataset $x_i^t \in \mathcal{X}_t$ and $y_i^t \in \mathcal{Y}_t$. Given a prediction model h_Θ with parameters Θ, the objective is to maximize total log-likelihood over all tasks:

$$\max_{\Theta} \sum_{t=1}^{\mathcal{T}} \sum_{(x,y) \in \mathcal{X}_t, \mathcal{Y}_t} \log p_\Theta(y \mid x) \qquad (1)$$

Here, we consider a more challenging setting for ELLA: *rehearsal-free CL with task-agnostic inference* - (i) The model cannot store or access data from past tasks when learning a new one. (ii) At test time, the model must make predictions without knowing which task an input belongs to.

Low-Rank Adaptation (LoRA). Our method builds on LoRA (Hu et al., 2022), a PEFT technique that leverages the observation that fine-tuning large pre-trained models (PTMs) can be effectively performed within a low-dimensional subspace (Aghajanyan et al., 2020). Given a frozen pre-trained weight matrix $W_{\text{init}} \in \mathbb{R}^{d \times k}$, LoRA introduces a low-rank update $\Delta W = AB$, where $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times k}$ with $r \ll \min(d, k)$. The original weights W_{init} remain unchanged during training, and only the low-rank matrices A and
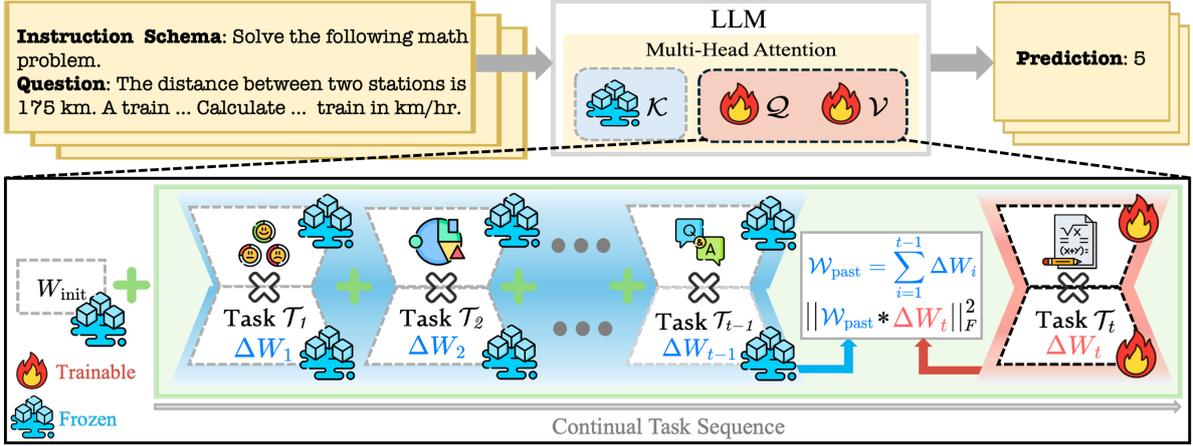
Figure 1: ELLA mitigates interference in continual LoRA training by accumulating past low-rank updates $\mathcal{W}_{past}$ and applying an energy-based alignment penalty $||\Delta W_t * \mathcal{W}_{past}||_F^2$ to discourage overlap in high-magnitude, task-specific directions. This selective regularization enables parameter reuse in less-used subspaces, achieving a better trade-off between plasticity and stability without requiring task labels, data replay, or architectural modifications.

B are optimized. During the forward pass, the original operation $h = W_{init}x$ becomes:

$$h = W_{init}x + \Delta Wx = W_{init}x + ABx, \quad (2)$$

This design reduces memory and compute overhead significantly, while achieving competitive performance to full fine-tuning across model families and benchmarks (Houlsby et al., 2019).

3.2 Subspace-Aware Continual Adaptation

We propose ELLA, a simple yet effective extension to LoRA, as a CL framework that balances *plasticity* and *stability* in the adapter updates using a subspace-aware regularization strategy. Rather than enforcing strict orthogonality between LoRA updates, which can hinder forward transfer, ELLA penalizes overlap with past task-specific directions in proportion to their accumulated energy, thereby discouraging harmful interference while permitting reuse of low-magnitude directions that facilitate knowledge transfer, as illustrated in Fig. 1.

Let the LoRA update for task t be $\Delta W_t = A_t B_t$, where $A_t \in \mathbb{R}^{d \times r}$ and $B_t \in \mathbb{R}^{r \times k}$. We construct a cumulative signal from the sum of LoRA-induced weight changes of past tasks:

$$\mathcal{W}_{past} = \sum_{i=1}^{t-1} \Delta W_i. \quad (3)$$

By construction, $\mathcal{W}_{past}$ encodes the dominant directions heavily used by prior tasks. Motivated by the observation that high-magnitude LoRA components are typically more task-discriminative (Aghajanyan et al., 2020), we introduce the ELLA alignment penalty as follows:

$$\mathcal{L}_{ELLA} = ||\Delta W_t * \mathcal{W}_{past}||_F^2, \quad (4)$$

where $|| \cdot ||_F$ denotes the Frobenius norm and $*$ represents element-wise matrix multiplication. The overall training loss for task t is then:

$$\mathcal{L} = \sum_{(x,y) \in \mathcal{X}_t, \mathcal{Y}_t} \log p_\Theta(y \mid x) + \lambda \cdot \mathcal{L}_{ELLA}. \quad (5)$$

Here, $\lambda \geq 0$ controls the trade-off between plasticity (learning the new gradient step) and stability (suppressing interference with past subspaces).

Formal Characterization: We next formalize the effect of this regularizer, as summarized in the following proposition. The detailed proof is discussed in Appendix A.

Proposition 1. *Let G denote the unconstrained gradient step for task t, and let $E = |(\mathcal{W}_{past})| + \varepsilon$ denote the accumulated energy of past updates. Then the optimal update $\Delta W_t^\star$ under the ELLA-regularized objective is*

$$(\Delta W_t^\star) = \frac{G}{1 + \lambda E^2} \quad (6)$$

Moreover, the interference with past tasks is bounded by

$$\left| \langle \Delta W_t^\star, \mathcal{W}_{past} \rangle_F \right| \leq \frac{||G||_F}{2\sqrt{\lambda}} ||E^{-1} * \mathcal{W}_{past}||_F \quad (7)$$

At a high level, Eq. 6 shows that ELLA acts as an *anisotropic shrinkage operator*: coordinates with large past energy are shrunk more aggressively while low-energy coordinates remain flexible. The shrinkage strength is directly controlled by λ: larger values amplify the penalty on high-energy directions, prioritizing stability, while smaller values reduce the penalty, promoting plasticity. This

property is further assessed in Sec. 5 and Fig. 6. The second result, Eq. 7, establishes a provable upper bound on task interference, ensuring that dominant past directions contribute minimally to forgetting. These properties formally justify ELLA's ability to suppress destructive overlap while reusing underutilized subspaces for forward transfer.

4 Experiments

4.1 Datasets

We conduct experiments on three CL benchmark datasets. The three selected datasets are divided into two categories: TRACE dataset concentrates on modern, generative tasks that are essential to LLM applications, including scientific and mathematical reasoning tasks and code generation; while the Standard CL benchmark (SC) and Long Sequence Benchmark (LS) focus on conventional NLP tasks like text classification and are more extensively studied by various CL algorithms.

TRACE is a modern benchmark tailored for CL in LLMs, covering a diverse set of 8 tasks that span domains such as multiple-choice question answering, multilingual understanding, code generation, scientific and mathematical reasoning (Wang et al., 2023b). See Appendix B.2 for details on the tasks.

Standard CL Benchmark (SC) is a popular CL benchmark designed for language models, comprising five text classification datasets introduced by (Zhang et al., 2015). Following (Wang et al., 2023a), we select AG News, Amazon Reviews, DBpedia, and Yahoo Answers, and form three shuffled task sequences, denoted as Orders 1, 2, and 3.

Long Sequence Benchmark (LS) extends the standard CL setup by incorporating 15 tasks, including five classification datasets, nine GLUE and SuperGLUE benchmarks, and the IMDB dataset (Razdaibiedina et al., 2023). Following the protocol of prior works (Liao et al., 2025), we train each task using $1,000$ randomly selected samples, with 500 samples per class reserved for testing. These tasks are also arranged into shuffled sequences Orders 4, 5, and 6.

4.2 Metrics

Let $a_{i,j}$ denote the testing performance on the j-th task after training on the i-th task. We evaluate across: **Overall Accuracy (OA)** (Chaudhry et al., 2018): The average accuracy across all tasks after training on the last task, i.e., $\text{OA}_{\mathcal{T}} = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} a_{\mathcal{T},t}$; **Forward Transfer (FWT)** (Lopez-

Paz and Ranzato, 2017): measures how much knowledge from previous tasks transfers to a new task, i.e., $\text{FWT}_{\mathcal{T}} = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} (a_{t,t} - a_{0,t})$, where $a_{0,t}$ refers to the performance of training task t individually; **Backward Transfer (BWT)** (Ke and Liu, 2022): measures how much the learning of subsequent tasks influences the performance of prior tasks, i.e., $\text{BWT}_{\mathcal{T}} = \frac{1}{\mathcal{T}-1} \sum_{t=1}^{\mathcal{T}-1} (a_{\mathcal{T},t} - a_{t,t})$. Moreover, we also report *general ability* (GA) – the average generalization performance across unseen datasets post CL, and *delta general ability* (DeltaGA) – change in GA relative to the original LLM.

4.3 Baselines

We compare ELLA against a comprehensive suite of CL baselines. This includes simple sequential fine-tuning approaches (SeqFT, SeqLoRA), classic regularization methods (EWC (Kirkpatrick et al., 2017), LwF (Li and Hoiem, 2017)), and modern prompt-tuning techniques (L2P (Wang et al., 2022b), LFPT5 (Qin and Joty, 2021)). We also benchmark against state-of-the-art adapter-based methods, including those that enforce strict orthogonality (O-LoRA (Wang et al., 2023a), LB-CL (Qiao and Mahdavi, 2024)) as well as those that combine orthogonality with complex decomposition or replay mechanisms (DATA (Liao et al., 2025), Recurrent KIF (Feng et al., 2025)). Details of baselines are provided in Appendix B.3.

4.4 Implementation Details

We evaluate ELLA on a range of LLMs including encoder-decoder T5 models (Raffel et al., 2020) (T5-Large, T5-XL), and decoder-only LLaMA models (Touvron et al., 2023) (LLaMA-3.1 8B). ELLA training is performed with the DeepSpeed library using V100 16GB GPUs for T5 and A40 48GB GPUs for LLaMA, while larger baselines such as DATA (Liao et al., 2025) require H100 80GB GPUs. *All experiments are performed with instruction tuning* (Wei et al., 2021). Following (Wang et al., 2023a), we apply LoRA modules (rank=8) to the Attention Q-V layers and report the average result of 3 runs. More details in Appendix B.6.

4.5 Results

Table 1 reports overall performance across three CL benchmarks. Across all architectures, task orders, task natures (modern generative tasks or conventional text classification) and evaluation settings, **ELLA consistently sets a new state of the art in**

Methods	Standard CL Benchmark (SC)				Long Sequence Benchmark (LS)				TRACE	
	Order 1	Order 2	Order 3	OA	Order 4	Order 5	Order 6	OA	Order 7	OA
T5-Large										
SeqFT (de Masson D'Autume et al., 2019)	18.9	24.9	41.7	28.5	7.4	7.3	7.4	7.4	-	
SeqLoRA	39.5	31.9	46.6	39.3	4.9	3.5	4.2	4.2	12.1	
IncLoRA	63.4	62.2	65.1	63.6	63.0	57.9	60.4	60.5	-	
SeqSVD	40.0	63.3	44.9	49.4	13.7	13.8	12.2	13.2	-	
EWC (Kirkpatrick et al., 2017)	46.3	45.3	52.1	47.9	44.9	44.0	45.4	44.8	-	
LwF (Li and Hoiem, 2017)	52.7	52.9	48.4	51.3	49.7	42.8	46.9	46.5	-	
L2P (Wang et al., 2022b)	59.0	60.5	59.9	59.8	57.7	53.6	56.6	56.0	-	
L-CL	75.3	73.5	71.9	73.6	66.5	64.0	69.0	66.5	-	
B-CL	76.4	71.5	75.1	74.3	65.7	66.4	69.2	67.1	-	
IncSVD	76.0	73.4	74.0	74.5	67.6	65.3	62.6	65.2	-	
LB-CL (Qiao and Mahdavi, 2024)	76.9	76.5	<u>76.8</u>	76.7	68.4	67.3	71.8	69.2	-	
O-LoRA (Wang et al., 2023a)	73.5	71.4	70.0	71.6	65.4	65.2	65.2	65.3	23.1	
+ MIGU (Du et al., 2024)	<u>77.1</u>	<u>77.0</u>	75.6	76.6	67.3	68.5	74.0	70.0	-	
DATA (Liao et al., 2025)	71.5	70.5	68.0	70.0	71.5	70.5	68.0	70.0	16.7	
+ Replay	77.0	75.6	75.2	75.9	**75.6**	**73.2**	<u>74.1</u>	<u>74.3</u>	<u>36.5</u>	
LFPT5 (Qin and Joty, 2021)	66.6	71.2	76.2	71.3	69.8	67.2	69.2	68.7	-	
SeqLoRAReplay	4.0	73.1	73.0	73.3	<u>74.2</u>	<u>72.7</u>	73.9	73.6	34.0	
Recurrent-KIF (Feng et al., 2025)	-	-	-	**78.4**	-	-	-	**77.8**	-	
ELLA (ours)	**80.0**	**80.0**	**79.8**	**79.9**	73.4	72.0	**75.4**	73.6	**40.0**	
LLaMA 3.1-8B										
SeqLoRA	75.88	74.40	74.35	74.86	67.81	65.93	63.80	65.85	28.23	
O-LoRA (Wang et al., 2023a)	69.39	67.58	71.44	69.46	69.54	64.42	66.50	66.82	28.45	
DATA⋆ (Liao et al., 2025)	76.10	75.69	75.52	75.77	71.34	70.77	72.95	71.69	31.33	
+ Replay⋆	<u>77.56</u>	<u>77.01</u>	76.03	<u>76.83</u>	**73.10**	**73.05**	74.17	73.44	**34.16**	
SeqLoRAReplay	77.02	76.53	<u>76.19</u>	76.58	72.03	<u>72.96</u>	<u>75.38</u>	<u>73.46</u>	33.02	
ELLA (ours)	**77.80**	**77.20**	**77.70**	**77.57**	<u>72.87</u>	72.84	**76.82**	**74.18**	<u>33.29</u>	

Table 1: Overall Average Accuracy (OA) comparison of baselines and ELLA (ours) on Standard CL benchmark (Order 1, 2, 3) and Long Sequence benchmark (Order 4, 5, 6) and TRACE (Order 7) across multiple transfer orders. Methods in gray rely on replay mechanisms to boost performance. Best results in **bold** and second best <u>underlined</u>. ⋆ denotes methods that encounter OOM on GPUs smaller than $2\times$ H100s, underscoring their limited scalability.

replay-free CL for LLMs. Remarkably, it not only dominates replay-free baselines but also achieves comparable or surpasses strong replay-augmented methods in many scenarios, despite introducing no additional memory or computational burden.

On TRACE dataset, ELLA achieves a notable performance gain of up to 23.3% (compared to DATA approach without replay, using T5-Large as the base model). For the decoder-only, LLama 3.1-8B base model, ELLA outperforms all rehearsal-free CL algorithms by 2-6%, achieving comparable performance to approaches with replay mechanisms. Notably, the TRACE dataset is a collection of generative tasks spanning reasoning, code, and multilingual tasks that broadly covers various domains. This indicates the efficacy of ELLA to generalize to multiple domains and tasks, demonstrating the relevancy of ELLA in modern LLM applications.

On the Standard CL benchmark, ELLA achieves an average accuracy of 79.9% on T5-Large, outperforming the strongest replay-free competitor (Qiao and Mahdavi, 2024) and exceeding DATA (Liao et al., 2025) and Recurrent KIF (Feng et al., 2025),

both of which depend on data rehearsal. On the more demanding Long Sequence (LS) setup, ELLA improves over (Qiao and Mahdavi, 2024; Du et al., 2024) by up to 4.3%, demonstrating robustness under long horizons where interference typically compounds. This is further examined in Sec. 5. We also compare ELLA's performance with Multi-Task Learning (MTL), which is widely considered to be the upper-bound of continual learning, in Appendix B.7, and show that it closely follows that upper-bound.

Beyond average accuracy, Fig. 2 and 3 reveal deeper dynamics. ELLA achieves the best BWT, sharply reducing forgetting, while also obtaining the highest FWT, signaling effective knowledge reuse across tasks. This is crucial: unlike orthogonality-based approaches (Wang et al., 2023a; Qiao and Mahdavi, 2024), which guarantee stability but block positive transfer, ELLA explicitly allows low-energy subspace reuse, yielding consistent forward gains. Conversely, replay-based approaches achieve modest transfer but at the cost of large buffers and compute. Notably, ELLA sustains high performance on interference-sensitive
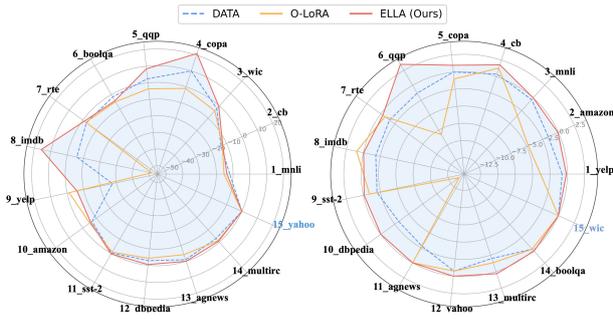
Figure 2: Backward Transfer (BWT) performance for T5-Large model on Long Sequence Benchmark (LS) dataset with Order 4 (left panel) and Order 6 (right panel). ELLA demonstrates superior resistance to performance decline than baselines (higher values indicate better retention of prior task performance).
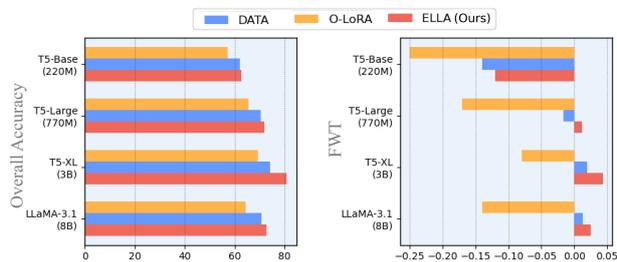

Figure 3: Comparison of Overall Accuracy (OA) and Forward Transfer (FWT) for four models across different backbone size and model families on the Long Sequence Benchmark (LS).

tasks like QQP, IMDB, and DBpedia, where baselines degrade sharply. ELLA thus demonstrates that selective subspace decorrelation provides a more principled way to reconcile stability and plasticity than existing strategies.

Equally important, ELLA achieves these gains with *no replay buffer, no generative augmentation, and no task identity signals*. Replay-heavy methods (Liao et al., 2025; Feng et al., 2025) not only incur large memory overheads but also encounter out-of-memory failures on LLaMA-3.1 8B when trained on most commercially available GPUs. ELLA avoids these pitfalls entirely, remaining lightweight, scalable, and stable even at billion-parameter scale.

In summary, ELLA combines state-of-the-art replay-free accuracy with superior forward transfer, minimal forgetting, and low overhead, setting a new benchmark for scalable CL in LLMs.

4.6 Task Generalization

LLMs that are continually trained on new tasks often exhibit a decline in general performance, revealing catastrophic forgetting of their original capabil-

Methods	MMLU	GSM8K	BBH	PIQA	AGIEval	DeltaGA
Zero Shot	59.64	69.67	35.65	72.42	43.92	0
SeqLoRA	56.05	70.20	35.16	67.85	41.02	-11.02
O-LoRA	57.36	66.79	33.82	64.31	**44.21**	-14.81
SeqLoRAReplay	55.81	66.43	32.90	67.77	41.58	-16.81
DATA	46.84	14.17	12.02	59.02	39.11	-110.14
Ours	**59.92**	**71.05**	**36.37**	**75.52**	43.92	**5.48**

Table 2: Generalization ability (GA) on the unseen, generative tasks after adapting LLama 3.1-8B on Order 1 of the Standard CL benchmark (SC). Methods in gray rely on replay mechanisms to boost performance.

ities. To assess this aspect, we evaluate the generalization ability of ELLA in the context of continual learning, focusing on its performance on five unseen cross-task benchmarks: MMLU (Hendrycks et al., 2020), GSM8k (Cobbe et al., 2021), BBH (Suzgun et al., 2022), AGIEval (Zhong et al., 2023), and PIQA (Bisk et al., 2020). More details on these datasets are provided in Appendix B.

We begin with a fine-tuned LLaMA 3.1-8B model trained on Order 1 of the SC Benchmark. As shown in Table 2, ELLA consistently achieves higher accuracy across all benchmarks compared to baselines, demonstrating its effectiveness in preserving knowledge while enabling strong cross-task generalization. Crucially, we see improved performance even over the original zero-shot model, underscoring its superiority in effectively combining acquired knowledge to address novel tasks. Beyond mitigating forgetting, ELLA is the only algorithm that delivers positive *DeltaGA* while all other algorithms present negative values. This shows that ELLA not only retains prior skills but also facilitates forward transfer to unseen tasks. Also note that replay-based methods lead to notably more negative *DeltaGA* due to the data rehearsal, which further demonstrates its limitation in practical applications. Overall, the results show that ELLA provides a stable, rehearsal-free path to cross-task generalization, making it valuable for the continual deployment of LLMs in dynamic environments.

4.7 Scale to Larger Models

To assess the scalability and robustness of ELLA across model sizes and architectures, we examine CL performance across T5-Base (220M), T5-Large (770M), T5-XL (3B), and LLaMA-3.1 8B on Order 5 of the Long Sequence Benchmark. As seen in Fig. 3, ELLA demonstrates consistent gains in average performance and transfer metrics as model size increases, outperforming baselines across all backbone configurations.

It is worth noting that even with the largest back-

Method	Trainable Params	Storage (MB)	Replay	Time/Epoch (mins)
SeqLoRA	0.062	0	0	4
O-LoRA	0.062	31.46	0	4.5
SeqLoRAReplay	0.062	0	2%	4
DATA	0.369	147.46	2%	6.5
Ours	0.062	4.19	0	4.5

Table 3: Comparison of training efficiency, memory overhead, and storage requirements with the T5-Large model.

bone model, O-LoRA(8B) still falls short in terms of FWT compared to the smallest version of ELLA (220M). This further highlights the crucial importance of selecting the pertinent PEFT algorithm for continual adaptation, rather than relying solely on scaling backbone size. Notably, the T5-XL model achieves higher OA than the larger LLaMA-3.1 model, highlighting that encoder-decoder architectures like T5 might be more effective for CL on classification tasks than decoder-only LLMs. We attribute this to the full-sequence bidirectional attention and cross-attention present in encoder-decoder designs, which better facilitate knowledge retention and task transfer than the causally masked architecture of models like LLaMA. These findings underscore that, beyond scaling size, architectural choice plays a crucial role, and shows that forgetting is model-dependent.

4.8 Efficiency Analysis

As shown in Table 3, ELLA matches O-LoRA in trainable parameters while cutting storage from 31.46MB to just 4.19MB–an $8\times$ reduction. Unlike replay-based methods such as DATA and SeqLoRAReplay, ELLA requires no buffer or feature storage and adds only minimal runtime cost. By embedding a lightweight regularization penalty into the loss instead of storing multiple task-specific modules, ELLA achieves strong efficiency in memory, compute, and training time – showing that ELLA is scalable and practical for long-horizon CL, where resource demands are typically prohibitive.

5 Discussions

Does ELLA preserve previous task performance during CL? We examine the effectiveness of ELLA in preventing degradation on previously learned tasks by measuring the change in prediction loss on past-task batches after training a new task. As shown in Fig. 4, ELLA significantly reduces the number of batches experiencing large increases in loss, especially in the high-loss tail region. This indicates that ELLA better preserves useful gradients from prior tasks by leveraging its alignment penalty
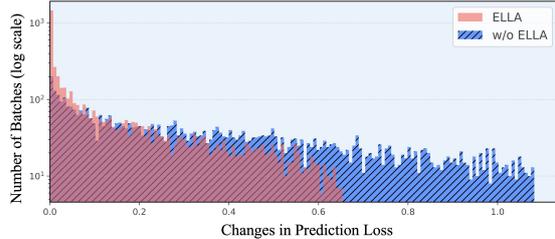

Figure 4: Histogram of prediction loss changes after training on a new task on the T5-Large model. The ELLA constraint helps reduce the changes – preserve the loss of previous tasks – in comparison to when it is not present ($\lambda = 0$).
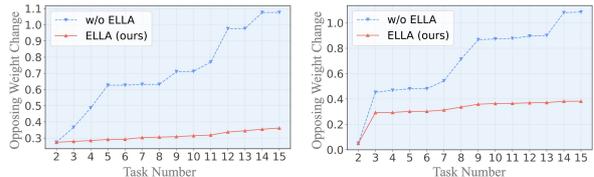

Figure 5: Opposing direction weight change across task sequence for T5-Large (left) and LLaMA-3.1 8B (right). ELLA consistently reduces backward-conflicting updates, promoting stable continual adaptation.

strategy. In contrast, the baseline (without ELLA) exhibits a broader distribution of loss spikes, revealing higher susceptibility to catastrophic forgetting. These results confirm that ELLA enhances stability across tasks without requiring replay or task labels.

Directional Consistency of Updates Over Task Sequence. In Fig. 5, we analyze the degree of opposing weight updates after each new task for Order 5 on both T5-Large and LLaMA3.1-8B. Specifically, we measure the magnitude of weight change that occurs in the direction opposite to that of the previous task, indicating a misalignment between updates over time. Standard LoRA exhibits consistently high opposing updates, suggesting that new learning often disrupts previously acquired representations, while ELLA significantly reduces such opposing direction weight updates, enabling smoother and more stable knowledge accumulation across tasks. This improvement is consistent across both encoder-decoder and decoder-only models.

Impact of λ Scaling. Fig. 6 shows that $\lambda = 0$ (i.e, a vanilla LoRA) results in severe forgetting, as evidenced by low OA and BWT, while increasing λ progressively improves both metrics by constraining interference. Excessively large values, however, hinder new task adaptation, lowering OA. The best performance arises at moderate λ, where the two terms in Eq. 5 are balanced, yielding both stability and plasticity. This behavior is consistent with our theoretical characterization in Appendix A, which
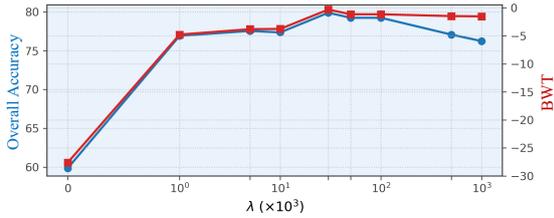
Figure 6: Impact of λ scaling on Order 1 of the Standard CL benchmark using the T5-Large model. Overall Accuracy and BWT as λ varies shows that moderate scaling achieves the best stability-plasticity balance.

LoRA_dim	Order 1	Order 2	Order 3	Avg
2	72.29	74.00	77.08	74.46
4	73.22	75.15	77.72	75.36
8	**79.95**	**80.00**	**79.82**	**79.92**
16	77.38	77.65	76.19	77.07

Table 4: Impact of LoRA rank on the Standard CL Benchmark (SC), with T5-Large as the base model.

shows that λ acts as an anisotropic shrinkage factor in our update, interpolating between unconstrained learning and stability-preserving adaptation.

Studying Optimal LoRA Rank for Plasticity-Stability Tradeoff. We investigate how LoRA rank influences CL by evaluating ELLA with T5-Large across three task orders while varying the LoRA dimension r. As shown in Table 4, performance improves with increasing r, peaking at $r = 8$. Very low ranks ($r = 2$) severely limit plasticity and hinder adaptation to new tasks, while higher ranks ($r = 16$) reduce stability by overfitting to current tasks. This underscores that careful rank selection is crucial for sustaining continual adaptation without catastrophic forgetting.

6 Conclusion

We introduced **ELLA**, a simple yet powerful approach for continual adaptation of LLMs that mitigates forgetting without replay or task identifiers. By encouraging controlled de-alignment from past subspaces, ELLA reduces destructive interference while enabling reuse of underutilized directions. We provide formal guarantees showing that ELLA's update admits a closed-form solution with bounded interference, offering a principled foundation for balancing plasticity and stability. Extensive experiments across classification as well as diverse generative benchmarks, and model architectures demonstrate that ELLA consistently improves both forward and backward transfer, while remaining highly parameter- and memory-efficient. Notably, ELLA surpasses state-of-the-art and scales robustly to larger backbones such as T5-3B and LLaMA-

3.1-8B with accuracy gains of up to 9.6%. It also uniquely enhances generalization on previously unseen tasks compared to the original LLM model – an ability absent in prior methods. These findings position ELLA as a lightweight, scalable, and universal method for LLM lifelong adaptation, demonstrating that efficiency and continual improvement need not come at the cost of forgetting.

7 Limitations

While our approach demonstrates strong performance, its scalability to more complex continual learning scenarios involving hundreds of tasks remains an open question. Moreover, although task identification is not required at inference time, our current training still assumes task labels to assign task-specific LoRA parameters. Developing task-agnostic training strategies is a promising direction for future work. Also, investigations regarding whether adapter-based CL approaches benefit smaller (<1B) models more than their larger counterparts remains underexplored. Finally, due to resource constraints, we have not evaluated our method on larger models like LLaMA-3.1-70B, and extending ELLA to continual multimodal LLMs remains an exciting avenue for exploration.

8 Ethical Considerations

ELLA represents a significant step towards practical and scalable lifelong learning for LLMs. By enabling efficient, replay-free updates, our work makes it more feasible to keep models current in dynamic environments without hindering data privacy and reducing the environmental footprint of AI systems. While ELLA's design promotes efficiency and privacy, it does not inherently mitigate the risks of misuse, such as the generation of biased or harmful content, which are endemic to the base models. Deploying models updated with ELLA requires a continued commitment to responsible data curation, fairness audits, and robust safety mechanisms to address these challenges. The datasets and models we used are public, hence there are no privacy issues.

9 AI Writing Statement

This paper utilized AI assistance for language polishing of the manuscript, including vocabulary correction and spell checking.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774.*

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. 2020. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255.*

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. 2017. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3366–3375.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, P Dokania, P Torr, and M Ranzato. 2019. Continual learning with tiny episodic memories. In *Workshop on Multi-Task and Lifelong Reinforcement Learning*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems, 2021. *URL https://arxiv. org/abs/2110.14168*, 9.

Cyprien de Masson D'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, 32.

DeepSpeed. 2025. Deepspeed: Learning rate schedulers. https://deepspeed.readthedocs.io/en/latest/schedulers.html. Accessed: 2025-09-22.

Shibhansh Dohare, Richard S Sutton, and A Rupam Mahmood. 2021. Continual backprop: Stochastic gradient descent with persistent randomness. *arXiv preprint arXiv:2108.06325.*

Wenyu Du, Shuang Cheng, Tongxu Luo, Zihan Qiu, Zeyu Huang, Ka Chun Cheung, Reynold Cheng, and Jie Fu. 2024. Unlocking continual learning abilities in language models. *arXiv preprint arXiv:2406.17245.*

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. 2020. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pages 3762–3773. PMLR.

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.*

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300.*

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Zixuan Ke and Bing Liu. 2022. Continual learning of natural language processing tasks: A survey. *arXiv preprint arXiv:2211.12701.*

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

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. 2019. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International conference on machine learning*, pages 3925–3934. PMLR.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Huanxuan Liao, Shizhu He, Yupu Hao, Jun Zhao, and Kang Liu. 2025. Data: Decomposed attention-based task adaptation for rehearsal-free continual learning. *arXiv preprint arXiv:2502.11482*.

Jiaming Liu, Senqiao Yang, Peidong Jia, Renrui Zhang, Ming Lu, Yandong Guo, Wei Xue, and Shanghang Zhang. 2023. Vida: Homeostatic visual domain adapter for continual test time adaptation. *arXiv preprint arXiv:2306.04344*.

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

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Fuli Qiao and Mehrdad Mahdavi. 2024. Learn more, but bother less: parameter efficient continual learning. *Advances in Neural Information Processing Systems*, 37:97476–97498.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. *arXiv preprint arXiv:2007.03085*.

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

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*.

Paul Ruvolo and Eric Eaton. 2013. Ella: An efficient lifelong learning algorithm. In *International conference on machine learning*, pages 507–515. PMLR.

Chenyang Song, Xu Han, Zheni Zeng, Kuai Li, Chen Chen, Zhiyuan Liu, Maosong Sun, and Tao Yang. 2023. Conpet: Continual parameter-efficient tuning for large language models. *arXiv preprint arXiv:2309.14763*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2022. *URL https://arxiv.org/abs/2206.04615*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and 1 others. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023a. Orthogonal subspace learning for language model continual learning. *arXiv preprint arXiv:2310.14152*.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, and 1 others. 2023b. Trace: A comprehensive benchmark for continual learning in large language models. *arXiv preprint arXiv:2310.06762*.

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

Yizhong Wang, Swaroop Mishra, Pegah Alipoor-molabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, and 1 others. 2022a. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

Zhicheng Wang, Yufang Liu, Tao Ji, Xiaoling Wang, Yuanbin Wu, Congcong Jiang, Ye Chao, Zhencong Han, Ling Wang, Xu Shao, and 1 others. 2023c. Rehearsal-free continual language learning via efficient parameter isolation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10933–10946.

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

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*.

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

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024. Sapt: A shared attention framework for parameter-efficient continual learning of large language models. *arXiv preprint arXiv:2401.08295*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*.

Appendix

This appendix provides supplementary material to support the findings presented in the main paper. We begin in Section A with a detailed theoretical analysis of the ELLA regularizer, offering a formal proof of its properties, including its formulation as an anisotropic shrinkage operator and its ability to bound interference provably. In Section B, we provide a comprehensive overview of our experimental settings. This includes detailed descriptions of the datasets, baselines, task sequences, instruction prompts, specific hyperparameter configurations used in our evaluations, comparison to Multi-Task learning setup which is the upper bound for CL, as well as additional discussions. Finally, Section C lists and discusses the artifact licenses.

A Theoretical Analysis of the ELLA Regularizer

Our theoretical analysis shows that the ELLA regularizer has two key properties: (i) its optimal update is an anisotropic shrinkage operator that selectively preserves plasticity, and (ii) it establishes a provable bound on the interference with past tasks, guaranteeing stability.

A.1 Setup and Preliminaries

Our analysis is set in the context of continual learning, where a model adapts to a sequence of tasks. For each new task t, the model is adapted by learning a low-rank update matrix, ΔW_t, using the LoRA method. This update is defined as the product of two smaller matrices:

$$\Delta W_t = A_t B_t \tag{8}$$

where $A_t \in \mathbb{R}^{d \times r}$ and $B_t \in \mathbb{R}^{r \times k}$ with rank $r \ll \min\{d, k\}$. This matrix represents the specific knowledge acquired for task t. We define the aggregated matrix of past updates as:

$$\mathcal{W}_{past} = \sum_{i=1}^{t-1} \Delta W_i \tag{9}$$

The practical regularization term used in our work is $\|\mathcal{W}_{past} * \Delta W_t\|_F^2$. For analytical tractability, we replace this term with a closely related objective. We introduce an energy matrix $E \in \mathbb{R}^{d \times k}$ derived from past updates, be defined as $E_{ij} = |(\mathcal{W}_{past})_{ij}| + \varepsilon$ for a small constant $\varepsilon > 0$ that ensures all entries are positive and E^{-1} is well-defined. This preserves the core principle of suppressing alignment with high-energy coordinates.

For the current task t, there is an ideal, unconstrained update that would best minimize the task loss, $\mathcal{L}_t$. This is the standard gradient step, which we denote as G.

$$G \triangleq \nabla \mathcal{L}_t(W_{t-1})$$

We think of G as the update we would make if we did not have to worry about forgetting past knowledge.

The goal of ELLA is to find an optimal update ΔW_t that is close to the target gradient G (learning the new task) but is penalized for changing parameters that were important for past tasks (not forgetting). This is formulated as the following regularized optimization problem:

$$\min_{\Delta W} \quad \underbrace{\frac{1}{2}\|\Delta W_t - G\|_F^2}_{\text{Plasticity Term}} + \underbrace{\frac{\lambda}{2}\|E * \Delta W_t\|_F^2}_{\text{Stability Term}} \tag{10}$$

where $\lambda > 0$ is the regularization strength and $*$ denotes the element-wise Hadamard product. The *plasticity term* encourages ΔW to be similar to G, while the *stability term* penalizes changes to coordinates with high energy (large values in E).

Proposition 1 The solution $\Delta W_t^{\star}$ to the ELLA objective in Eq. (10) has the following properties:

(i) It is an anisotropic shrinkage operator applied to the unconstrained step G, with the closed-form solution:

$$(\Delta W_t^{\star})_{ij} = \frac{G_{ij}}{1 + \lambda E_{ij}^2}$$

(ii) The interference with past updates, measured by the inner product $\langle \Delta W_t^{\star}, \mathcal{W}_{past} \rangle_F$, is bounded as follows:

$$\left| \langle \Delta W_t^{\star}, \mathcal{W}_{past} \rangle_F \right| \leq \frac{\|G\|_F}{2\sqrt{\lambda}} \|E^{-1} * \mathcal{W}_{past}\|_F$$

Proof. **Proof of part (i):** The objective function in Eq. (10) is strongly convex since its second derivative is strictly positive, and separable across the matrix entries (i, j). We can therefore solve for each entry $(\Delta W_t)_{ij}$ independently. Let $z = (\Delta W_t)_{ij}$, $g = G_{ij}$, and $e = E_{ij}$. The per-coordinate objective is:

$$\min_{z \in \mathbb{R}} \quad \tfrac{1}{2}(z - g)^2 + \tfrac{\lambda}{2}e^2 z^2$$

Taking the derivative with respect to z and setting it to zero yields the optimal solution $z^{\star}$:

$$(z^{\star} - g) + \lambda e^2 z^{\star} = 0 \quad \Rightarrow \quad (1 + \lambda e^2) z^{\star} = g$$

This gives the closed-form solution for each coordinate, proving part (i):

$$(\Delta W_t^\star)_{ij} = \frac{G_{ij}}{1 + \lambda E_{ij}^2} \qquad (11)$$

This operator selectively shrinks the update for each coordinate based on the accumulated energy of past updates E_{ij}, with larger energy leading to stronger shrinkage.

Proof of part (ii): Next, we first bound the norm of the energy-weighted update. Using the solution from Eq. (11):

$$\|E * \Delta W_t^\star\|_F^2 = \sum_{ij} \frac{E_{ij}^2 G_{ij}^2}{(1 + \lambda E_{ij}^2)^2} \qquad (12)$$

$$\leq \left(\sup_{x \geq 0} \frac{x}{(1 + \lambda x)^2} \right) \sum_{ij} G_{ij}^2 \qquad (13)$$

The supremum is attained at $x = 1/\lambda$, yielding $\sup_{x \geq 0} \frac{x}{(1 + \lambda x)^2} = \frac{1}{4\lambda}$. Therefore,

$$\|E * \Delta W_t^\star\|_F^2 \leq \frac{1}{4\lambda} \|G\|_F^2 \qquad (14)$$

This shows that the effective penalty is never worse than a $\frac{1}{4\lambda}$ fraction of the gradient energy, ensuring the shrinkage remains controlled.

Now, we try to bound the interference term $\langle \Delta W_t^\star, \mathcal{W}_{past} \rangle_F$. By rewriting the inner product and applying the Cauchy-Schwarz inequality:

$$\left| \langle \Delta W_t^\star, \mathcal{W}_{past} \rangle_F \right| \qquad (15)$$

$$= \left| \langle E * \Delta W_t^\star, E^{-1} * \mathcal{W}_{past} \rangle_F \right| \qquad (16)$$

$$\leq \|E * \Delta W_t^\star\|_F \|E^{-1} * \mathcal{W}_{past}\|_F \qquad (17)$$

Substituting the bound from Eq. (14) into Eq. (17) yields the final interference bound:

$$\left| \langle \Delta W_t^\star, \mathcal{W}_{past} \rangle_F \right| \leq \frac{\|G\|_F}{2\sqrt{\lambda}} \|E^{-1} * \mathcal{W}_{past}\|_F$$

This completes the proof of part (ii). The term $\|E^{-1} * \mathcal{W}_{past}\|_F$ ensures that coordinates with high energy (large E_{ij}) contribute minimally to the bound, formally proving that ELLA enforces stability by suppressing interference in high-energy coordinates, while preserving plasticity where capacity remains. □

B Experimental Settings

B.1 Datasets

Train Tasks. Tables 5 and 6 provide detailed information on the datasets utilized in our continual learning (CL) experiments. Table 5 presents the 15 datasets included in the Long Sequence Benchmark (Razdaibiedina et al., 2023), while Table 6 outlines the 8 datasets from TRACE (Wang et al., 2023b). Both tables include the corresponding evaluation metrics for each dataset.

Unseen tasks. We select the following datasets to test generalization performance post continual adaptation: (1) Multitask Language Understanding (MMLU) (Hendrycks et al., 2020), which includes multiple-choice questions across 57 subjects. (2) GSM8K (Cobbe et al., 2021), which is a high-quality linguistically diverse multi-step elementary math reasoning dataset. (3) BIG-Bench Hard (BBH) (Suzgun et al., 2022), which includes 27 challenging tasks spanning arithmetic, symbolic reasoning, and more, derived from BIG-Bench (BB) (Srivastava et al., 2022). Most of the data consists of multiple-choice questions. (4) AGIEval (Zhong et al., 2023), which includes a wide range of high-quality official entrance exams, qualifying exams, and advanced competitions tailored to human participants. (5) PIQA (Bisk et al., 2020) which is a dataset for commonsense reasoning, and was created to investigate the physical knowledge of existing models in NLP.

B.2 Task Sequence Orders

We report all task orders used for our CL experiments in Table 7.

B.3 Baselines

We compare our method against a comprehensive set of recent CL baselines, detailed as follows: (1) **SeqFT** sequentially fine-tunes all model parameters without CL mechanisms like regularization or replay (RorR). (2) **SeqLoRA** applies fixed-size LoRA tuning per task with/without replay. (3) **IncLoRA** allows incremental learning of new LoRA parameters for each task without constraints. (4) **SeqSVD** uses fixed-rank SVD adapters trained sequentially without RorR. (5) **EWC** (Kirkpatrick et al., 2017) finetunes LoRA with a regularization loss to prevent interference with past tasks. (6) **LwF** (Li and Hoiem, 2017) distills the model of the last task using the current task data. (7) **L2P** (Wang et al., 2022b) instantiates a prompt

Dataset Name	Category	Task	Domain	Metric
Yelp	CL Benchmark	Sentiment Analysis	Yelp Reviews	Accuracy
Amazon	CL Benchmark	Sentiment Analysis	Amazon Reviews	Accuracy
DBPedia	CL Benchmark	Topic Classification	Wikipedia	Accuracy
Yahoo	CL Benchmark	Topic Classification	Yahoo Q&A	Accuracy
AG News	CL Benchmark	Topic Classification	News	Accuracy
MNLI	GLUE	Natural Language Inference	Various	Accuracy
QQP	GLUE	Paragraph Detection	Quora	Accuracy
RTE	GLUE	Natural Language Inference	News, Wikipedia	Accuracy
SST-2	GLUE	Sentiment Analysis	Movie Reviews	Accuracy
WiC	SuperGLUE	Word Sense Disambiguation	Lexical Databases	Accuracy
CB	SuperGLUE	Natural Language Inference	Various	Accuracy
COPA	SuperGLUE	Question and Answering	Blogs, Encyclopedia	Accuracy
BoolQA	SuperGLUE	Boolean Question and Answering	Wikipedia	Accuracy
MultiRC	SuperGLUE	Question and Answering	Various	Accuracy
IMDB	SuperGLUE	Sentiment Analysis	Movie Reviews	Accuracy

Table 5: The details of 15 classification datasets in the Long Sequence Benchmark (Razdaibiedina et al., 2023). The first five tasks correspond to the standard CL benchmark (Zhang et al., 2015).

Dataset	Source	Avg len	Metric	Language	#Data
Domain-specific					
ScienceQA	Science	210	Accuracy	English	5,000
FOMC	Finance	51	Accuracy	English	5,000
MeetingBank	Meeting	2853	ROUGE-L	English	5,000
Multi-lingual					
C-STANCE	Social media	127	Accuracy	Chinese	5,000
20Minuten	News	382	SARI	German	5,000
Code Completion					
Py150	Github	422	Edim Similarity	Python	5,000
Mathematical Reasoning					
NumGLUE-cm	Math	32	Accuracy	English	5,000
NumGLUE-ds	Math	21	Accuracy	English	5,000

Table 6: The overview of dataset statistics in TRACE (Wang et al., 2023b). The 'Source' indicates the origin of the context. 'Avg len' denotes the average length, measured in word count for English, German, and code datasets, and in character count for Chinese datasets. 'SARI' is a score that is specific to evaluating simplification tasks.

pool for adaptive prompt selection and prompt tuning for individual samples. (8) **LFPT5** (Qin and Joty, 2021) learns soft prompts and generates pseudo-data for replay. (9) **L-CL** trains SVD adapters incrementally with SVD regularization. (10) **B-CL** applies SVD regularization with gradient projection. (11) **O-LoRA** (Wang et al., 2023a) enforces orthogonality between LoRA updates across tasks. (12) **MIGU** (Du et al., 2024) updates parameters based on gradient magnitude. (13) **LB-CL** (Qiao and Mahdavi, 2024) initializes low-rank matrix parameters in new tasks from specific past parameters besides enforcing orthogonality. (14) **DATA** (Liao et al., 2025) decomposes attention into high and low rank subspaces, and leverages orthogonality with optional replay. (15) **Recurrent KIF** (Feng et al., 2025) uses an expensive recurrent mechanism to modulate task-specific adapter reuse using replay.

B.4 Instruction Tuning

Instruction-following is a fundamental capability for LLMs to serve as an effective interface between humans and AI systems (Wei et al., 2021; Ouyang et al., 2022). We adopt instruction tuning as our training paradigm for two key reasons: (1) it enables the incorporation of human prior knowledge via explicit instructions, thereby facilitating more efficient learning; and (2) it enhances generalization by guiding the model to learn underlying principles that apply across tasks.

All tasks are formatted using a unified schema consisting as follows: (1) *Task Instruction* – a clear description of how to map an input (e.g., a sentence or document) to an output; (2) *Options* – the constrained set of valid output labels for the task; (3) *Text* – the input instance provided to the model; and (4) *Answer* – the correct target output. This combined sequence is fed into the pretrained model to guide prediction for all experiments.

Benchmark	Order	Task Sequence
Standard CL Benchmark	1	dbpedia → amazon → yahoo → ag
	2	dbpedia → amazon → ag → yahoo
	3	yahoo → amazon → ag → dbpedia
Long Sequence Benchmark	4	mnli → cb → wic → copa → qqp → boolqa → rte → imdb → yelp → amazon → sst-2 → dbpedia → ag → multirc → yahoo
	5	multirc → boolqa → wic → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yelp → amazon → yahoo
	6	yelp → amazon → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yahoo → multirc → boolqa → wic
TRACE	7	c-stance → fomc → meetingbank → py150 → scienceqa → numglue-cm → numglue-ds → 20minuten

Table 7: Seven distinct orders of task sequences were employed for the experiments in continual learning. Orders 1-3 align with the Standard CL Benchmarks, as adopted in previous studies (Liao et al., 2025). Orders 4-6 pertain to the Long Sequence Benchmarks, which encompass a total of 15 tasks (Razdaibiedina et al., 2023). Order 7 refers to the TRACE benchmark, specifically designed for LLMs, and comprises eight datasets (Wang et al., 2023b).

B.5 Task Instructions

Table 8 shows the prompts used for different tasks. NLI denotes natural language inference, and includes tasks MNLI, RTE and CB. SC denotes sentiment analysis, including Amazon, Yelp, SST-2 and IMDB. TC denotes topic classification and contains the tasks AG News, Dbpedia and Yahoo.

B.6 Implementation Details

Our implementation for ELLA uses PyTorch v2.0.1 (Paszke et al., 2019) and Deepspeed v0.10.0 (Rasley et al., 2020). Our experiments were conducted on machines equipped with 4 A40 GPUs/ 8 V100 GPUs for ELLA and all baselines, except DATA/DATA+Replay, which required 2 H100 GPUs. For all orders of task streams for the Standard-CL Benchmark and the Long Sequence Benchmark, we trained the models with one epoch using the AdamW optimizer (Loshchilov and Hutter, 2017) and WarmupLR scheduler (DeepSpeed, 2025) with a total batch size of 32. We used a constant learning rate of $1e-3$ for T5 and $1e-4$ for LLaMA, a dropout rate of 0.1, and a weight decay rate of 0. For TRACE Order 7 (C-STANCE, FOMC, MeetingBank, Py150, ScienceQA, NumGLUE-cm, NumGLUE-ds, 20Minuten), we trained with 5000 samples 5, 3, 7, 5, 3, 5, 5, 7 epochs respectively. The coefficient λ for different task orders has been reported in Table 9. λ was selected based on performance on a small, held-out validation set. To establish an effective search range, we followed the common heuristic of choosing values that scale the regularization term, $\mathcal{L}_{\text{ELLA}}$ (promoting stability), to a similar order of magnitude as the current task accuracy loss (promoting plasticity) for balanced learning. We observed that performance was robust across this range and that a single λ often generalized well across multiple subsequent tasks. The

LoRA rank was set to 8 for all experiments, and the proportion of past task data mixed in for replay methods was set to 2% of the original training set.

B.7 Comparison with Multi-Task Learning (MTL)

We compare ELLA against a non-continual Multi-Task Learning (MTL) upper bound to contextualize performance, following prior work. The results are presented in Table 10.

This comparison demonstrates how closely ELLA's cumulative knowledge acquisition approaches the theoretical upper bound set by simultaneous training. Our strong results suggest that ELLA achieves near-optimal performance, without even requiring access to all tasks at the same time (which is a more difficult but realistic scenario compared to the easier MTL setup) while maintaining the efficiency required for continuous deployment.

B.8 Hyperparameters and Training Efficiency

In comparison with Vanilla LoRA, the implementation of the ELLA regularization does not require additional training epochs or changes to the training schedule. ELLA's efficient and effective subspace decorrelation strategy facilitates the adaptation of LLMs for real-world Continual Learning (CL) needs, effectively mitigating the catastrophic degradation of prior knowledge often observed in vanilla LoRA training.

B.9 Raw Backward and Forward Transfer Scores

Tables 11 and 12 present the raw Backward Transfer (BWT) scores for each individual task within the Long Sequence Benchmark (Orders 4 and 6). Positive values indicate positive backward transfer, while negative values indicate catastrophic forgetting. The final column (Task 15) is the last task trained, so it has a BWT of 0.0.

Task	Prompts
NLI	What is the logical relationship between the "sentence 1" and the "sentence 2"? Choose one from the option.
QQP	Whether the "first sentence" and the "second sentence" have the same meaning? Choose one from the option.
SC	What is the sentiment of the following paragraph? Choose one from the option.
TC	What is the topic of the following paragraph? Choose one from the option.
BoolQA	According to the following passage, is the question true or false? Choose one from the option.
MultiRC	According to the following passage and question, is the candidate answer true or false? Choose one from the option.
WiC	Given a word and two sentences, whether the word is used with the same sense in both sentence? Choose one from the option.
FOMC	What is the monetary policy stance for the following text? Choose one from the option.
20Minuten	Provide a simplified version of the following paragraph in German.
ScienceQA	Choose an answer for the following question and give your reasons.
NumGLUE-cm	Solve the following math problem.
NumGLUE-ds	Solve the following math problem.
Py150	Continue writing the code.
MeetingBank	Write a summary of the following meeting transcripts.
C-STANCE	Determine the attitude of the following text towards the specified object. Choose one from the option.

Table 8: Instructions for different tasks.

Order	T5 λ	LLaMA λ
1–3	$0, 3\times10^4, \ldots, 3\times10^4$	$0, 3\times10^6, \ldots, 3\times10^6$
4	$0, 5\times10^5, \ldots, 5\times10^5, 5\times10^7$	$0, 5\times10^8, \ldots, 5\times10^8$
5	$0, 5\times10^6, \ldots, 5\times10^6, 5\times10^7, 5\times10^7, 5\times10^7$	$0, 5\times10^6, \ldots, 5\times10^6, 5\times10^7, 5\times10^7, 5\times10^7$
6	$0, 5\times10^5, \ldots, 5\times10^5$	$0, 5\times10^8, \ldots, 5\times10^8$
7	$0, 5\times10^5, \ldots, 5\times10^5$	$0, 5\times10^7, 5\times10^7, 5\times10^7, 5\times10^9, 5\times10^9, 5\times10^9, 5\times10^9$

Table 9: Coefficient λ settings for different task orders in T5 and LLaMA.

Table 13 contains the raw Forward Transfer (FWT) scores, detailing the performance across various model architectures, ranging from the 220M parameter T5-Base to the 8B parameter LLaMA-3.1, and the baseline methodologies evaluated.

B.10 Discussion on ELLA's scalability

We highlight that ELLA is a memory-constant algorithm with respect to the task sequence length, as it only stores a single aggregated matrix - which is negligible (4MB) compared to the much larger LLM and per-task LoRA memory footprints. This differs fundamentally from popular (i) Replay Methods (DATA, Recurrent KIF etc.), as their memory grows with the replay budget and task count, requiring high GPU memory for data storage and replay with and (ii) Orthogonality Methods (O-LoRA, DATA etc.), which require storing each individual past task-specific LoRA parameter, resulting in memory that grows linearly with the number of tasks, limiting their scalability. Contrastingly, ELLA achieves superior performance even with a constant negligible storage footprint that is smaller than the orthogonality baseline O-LoRA and smaller than SOTA CL approach, DATA, which uses replay to boost accuracy, making our method a practical and scalable solution for long-horizon CL at almost negligible overheads.

Method	Standard CL (SC)				Long Sequence (LS)				TRACE
	Order 1	Order 2	Order 3	OA	Order 4	Order 5	Order 6	OA	Order 7
MTL (Upper Bound)	80.0	80.0	80.0	80.0	76.5	76.5	76.5	76.5	41.75
ELLA (ours)	80.0	80.0	79.8	79.9	73.4	72.0	75.4	73.6	40.0

Table 10: Comparison of ELLA (ours) against the Multi-Task Learning (MTL) upper bound. MTL represents simultaneous training on all tasks.

Method	MNLI	CB	WiC	COPA	QQP	BoolQA	RTE	IMDB	Yelp	Amazon	SST-2	DBPedia	AGNews	MultiRC	Task 15
O-LoRA	-15.45	-12.79	-4.01	-1.00	-3.57	-4.95	-2.19	-51.41	**-1.15**	-5.07	-0.93	-4.22	-4.20	-1.20	0.0
DATA	**-11.97**	-13.64	-2.24	10.00	2.45	**-1.59**	-2.64	-5.80	-27.56	-6.71	-0.29	-2.82	-1.00	-2.97	0.0
ELLA (ours)	-13.68	**-12.50**	**-0.47**	**21.00**	**8.47**	-4.22	**-1.08**	**15.82**	-5.97	**-5.76**	**0.34**	**-0.42**	**0.00**	**-0.74**	0.0

Table 11: **Order 4**: Raw Backward Transfer (BWT) scores per task. Best results in **bold**.

Method	MNLI	CB	WiC	COPA	QQP	BoolQA	RTE	IMDB	Yelp	Amazon	SST-2	DBPedia	AGNews	MultiRC	Task 15
O-LoRA	-3.86	-4.75	-3.41	1.29	-1.00	-8.25	-0.72	0.99	-0.92	-13.99	-0.01	-0.71	-1.34	-0.21	0.0
DATA	-0.73	-1.59	-0.33	0.39	0.01	-1.31	-1.56	-1.83	-2.01	-3.19	-2.76	-0.72	-1.99	-0.22	0.0
ELLA (ours)	**-0.13**	**-0.07**	**0.01**	**1.79**	**1.00**	**3.51**	**-0.36**	**-0.05**	**-0.11**	**-0.01**	**0.00**	**0.03**	**0.33**	**0.18**	0.0

Table 12: **Order 6**: Raw Backward Transfer (BWT) scores per task. Best results in **bold**.

Method	T5-Base	T5-Large	T5-XL	LLaMA-3.1
oLoRA	-0.250	-0.170	-0.080	-0.140
DATA	-0.140	-0.016	0.019	0.013
ELLA (Ours)	**-0.120**	**0.012**	**0.043**	**0.025**

Table 13: Raw FWT scores across model families

C Artifact Licenses

According to their license, all the LLMs used in this paper fall under acceptable use cases. The licenses are listed for perusal: T5-Base (https://huggingface.co/google-t5/t5-base), T5-Large (https://huggingface.co/google-t5/t5-large), T5-XL (https://huggingface.co/google/t5-v1_1-xl), LLaMA-3.1-8b (https://huggingface.co/meta-llama/Llama-3.1-8B/blob/main/LICENSE).