Split-on-Share: Mixture of Sparse Experts for Task-Agnostic Continual Learning

Anonymous ACL submission

Abstract

Continual learning in Large Language Models (LLMs) is hindered by the plasticity-stability dilemma, where acquiring new capabilities often leads to catastrophic forgetting of previous knowledge. Existing methods typically treat parameters uniformly, failing to distinguish between specific task knowledge and shared capabilities. We introduce Mixture of **S**parse **E**xperts for **T**ask **A**gnostic Continual Learning, referred to as SETA, a framework that resolves the plasticity-stability conflict by decomposing the model into modular subspaces. Unlike standard updates where tasks compete for the same parameters, SETA separates knowledge into unique experts, designed to isolate task-specific patterns, and shared experts, responsible for capturing common features. This structure is maintained through elastic weight anchoring, which protects critical shared knowledge, enabling a unified gating network to automatically retrieve the correct expert combination for each task during inference. Extensive experiments across diverse domain-specific and general benchmarks demonstrate that SETA consistently outperforms state-of-the-art parameter-efficient fine-tuning-based continual learning methods. The paper code is available at Anonymous Repository

1 Introduction

Large Language Models have demonstrated remarkable performance across diverse natural language processing tasks, ranging from text classification and reasoning to dialogue and summarization (Brown et al., 2020; Raffel et al., 2020; OpenAI, 2023). However, deploying LLMs in continual learning settings remains challenging (Parisi et al., 2019; De Lange et al., 2021) as models must adapt to sequential tasks while preserving previously acquired knowledge (Kirkpatrick et al., 2017; Li and Hoiem, 2017). This challenge, known as catastrophic forgetting (Li and Hoiem, 2017; Kirkpatrick et al., 2017), arises from the fundamental trade-off between learning plasticity, the rapid adaptation to new tasks, and memory stability, the retention of learned knowledge. The problem is further exacerbated in parameter-efficient fine-tuning regimes, where sparsity for efficiency increases susceptibility to overwriting historical information (Han et al., 2015; Evci et al., 2020).

Existing continual learning strategies generally fall into three primary categories, including Replay based methods that rehearse historical data (Chaudhry et al., 2019b), Regularization based techniques that penalize updates to critical parameters (Kirkpatrick et al., 2017), and Parameter Isolation approaches that physically separate task specific modules (Rusu et al., 2016). Despite their differences, these approaches share a common limitation: they treat parameters uniformly within their respective subspaces (Ren et al., 2024). As a result, they fail to distinguish between generalizable features and task-specific requirements, leading to suboptimal trade-offs between stability and plasticity (Ramasesh et al., 2022).

To address these limitations, Continual Sparse Fine-Tuning provides a natural mechanism for uncovering the internal knowledge structure of large language models. Unlike parameter-efficient fine-tuning (PEFT)-based low-rank adapters that project updates through opaque dense matrices, sparse tuning identifies specific attention sub-blocks, particularly in the Value projection, based on high-utility gradient signals. This unique property serves as the foundation for our proposed framework, Mixture of Sparse Experts for Task Agnostic Continual Learning, referred to as **SETA**. The design of SETA is driven by three fundamental research questions, each addressing a specific limitation in state-of-the-art literature:

RQ1: *How can sparse parameter patterns be leveraged to decouple the acquisition of novel features from the retention of historical represen-*

tations? To address RQ1, SETA introduces a subspace-based expert decomposition that frames sparse continual learning within a Mixture-of-Experts paradigm. By exploiting the inherent block-wise sparsity of gradients, the model is separated into distinct expert modules, ensuring that conflicting updates operate in isolated subspaces rather than competing for a single monolithic block.

RQ2: *Is it possible to automatically differentiate between reusable shared knowledge and task-specific unique knowledge within a single unified framework?* To answer RQ2, the Split-on-Share (SoS) mechanism combined with adaptive elastic anchoring dynamically assigns parameters based on their recurrence across tasks. Overlapping parameters are designated as shared experts and stabilized to prevent drift, while non-overlapping parameters become unique experts that are strictly frozen to act as immutable memory, ensuring historical knowledge is preserved.

RQ3: *How can the relevant expert modules be dynamically retrieved during inference without relying on external task identifiers?* SETA resolves RQ3 by implementing adaptive gating expansion with logit invariance. This allows the router to evolve alongside the experts and dynamically select relevant parameters based solely on input tokens. This enables task-agnostic inference and supports scaling to large datasets and long task sequences without explicit task labels.

The remainder of the paper is organized as follows. Section 2 formulates the problem and motivates SETA. Section 3 introduces the SETA framework and the Split-on-Share mechanism. Section 4 presents experimental results, and Section 5 concludes with key insights and future directions.

2 Problem Formulation and Motivation

2.1 Challenges of Continual Learning

Continual learning aims to enable models to acquire new capabilities from a sequence of tasks $\mathcal{T} = \{T_1, \ldots, T_k\}$ without forgetting previously learned knowledge (Wu et al., 2024b). For large language models, this creates a fundamental resource paradox. Full fine-tuning updates all parameters θ, offers high plasticity, and allows the model to easily adapt to new data. However, full fine-tuning is computationally expensive and memory intensive (Hu et al., 2021a), and updating all parameters can severely disrupt the optimization landscape, leading to catastrophic forgetting (Luo et al., 2023). Freezing the model preserves prior knowledge but eliminates plasticity, preventing the acquisition of new knowledge (Parisi et al., 2019).

2.2 PEFT in Continual Learning

To address resource constraints, parameter-efficient fine-tuning (PEFT) methods such as Low-Rank Adaptation (LoRA) freeze the backbone and train small modules, reducing memory overhead (Hu et al., 2022). However, recent studies show that such approaches still face inherent limitations in continual multi-task settings. As Ren et al. (Ren et al., 2024) observed, there is a geometric disconnect between the loss minima of sequential tasks when updates are restricted to low-rank subspaces. Because PEFT limits updates to a narrow bottleneck dimension ($r \ll d$), it essentially provides a lossy approximation of the model's weights (He et al., 2025). This approximation introduces noise, hindering knowledge storage for long sequences. Under continuous updates, the optimization path is too narrow to satisfy conflicting requirements, causing a plasticity-stability failure where new learning overwrites historical parameters.

2.3 Shared Subspace Dilemma

The core failure of continual learning arises from the *Shared Subspace Dilemma*. As tasks are learned sequentially, parameters often become important for multiple tasks. Let $\mathcal{P}_t$ denote parameters for task t; its intersection with the cumulative historical subspace $\mathcal{P}_{1:t-1} = \bigcup_{k=1}^{t-1} \mathcal{P}_k$ defines a shared region $\mathcal{I} = \mathcal{P}_t \cap \mathcal{P}_{1:t-1}$. Here, updating weights risks overwriting prior knowledge, while freezing limits plasticity. Standard methods address this via uniform regularization (Kirkpatrick et al., 2017; Chaudhry et al., 2018):

$$\mathcal{L}_{\text{reg, uniform}} = \alpha \cdot \|W - W^*_{1:t-1}\|^2, \qquad (1)$$

where α is a fixed penalty. However, uniform regularization is too rigid, treating parameters equally rather than distinguishing shared features requiring controlled plasticity from task-specific regions needing stability (Saha et al., 2021). Since static penalties cannot simultaneously prevent semantic drift and enable new learning (Farajtabar et al., 2020), resolving this trade-off requires a structural solution rather than loss-based constraints alone.

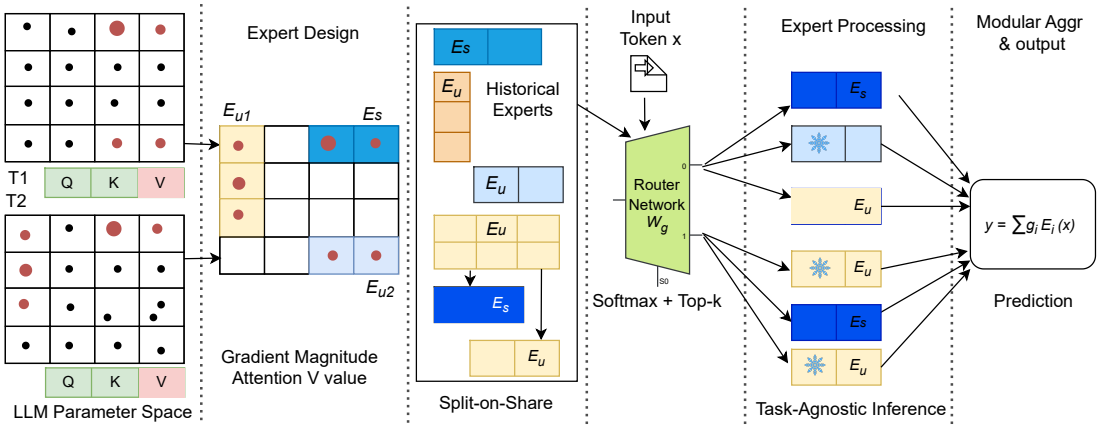

Figure 1: The SETA system utilizes a specialized sparse fine-tuning pipeline where task-specific parameters are isolated through a Split-on-Share (SoS) mechanism. This structure partitions the model into Shared E_{s} and Unique E_{u} experts to resolve parameter collisions while employing a task-agnostic gating network.

2.4 Mixture of Sparse Experts for CL

To address interference in continual learning and low-rank fine-tuning, we propose SETA, a modular framework for sparse subspace adaptation. SETA resolves the stability–plasticity conflict by separating task-specific and shared knowledge into unique and shared experts, preventing parameter overwriting. When subspaces overlap, the Split-on-Share mechanism decouples contested regions to enable stable structural evolution. Combined with an adaptive gating network, this design supports cumulative learning and task-agnostic inference.

3 Methodology

This section presents the SETA methodology. Detailed algorithm is provided in the appendix B.

3.1 Sparse Subspace Selection

We aim to fine-tune only the most relevant sparse sub-matrices to maximize downstream performance while minimizing computational cost. We employ a gradient-based parameter selection strategy inspired by Fisher Information (Sung et al., 2021) that strictly localizes adaptation to high-utility regions.

Gradient-Based Selection. We partition pre-trained weight matrices $W \in \mathbb{R}^{d\times k}$ into a grid of sub-blocks $\mathbb{B} = \{B_{i,j}\}$ of dimension $l \times l$, where $l = 256$ is the greatest common divisor of the layer dimensions. We identify high-utility regions by executing a warm-up round and computing an importance score $\mathcal{S}$, representing the average absolute gradient magnitude within each block:

$$\mathcal{S}_{i,j} = \frac{1}{l^2} \sum_{(u,v)\in B_{i,j}} |\nabla W_{u,v}|, \qquad (2)$$

where, $\nabla W_{u,v}$ indicates sensitivity. We select the top-k blocks with the highest scores to form the active sparse subspace. We skip block selection for any layer falling below a threshold τ_{block}. These skipped layers use residual connections, keeping resources focused on high-impact regions. This selection strategy is validated by our empirical analysis of gradient behavior. Figure 2 reveals strong vertical structures and row-wise hotspots, such as Column 9 and Rows 12/14, signaling important parameters that drive plasticity. Furthermore, Figure 3 confirms that ∼95% of these high-magnitude gradients belong to the attention Value (V) projection. Theoretical analysis attributes this to Softmax Saturation. As the scaled dot-product $QK^T/\sqrt{d_k}$ grows, the softmax gradients vanish, suppressing updates to Q and K, where vanishing gradients suppress updates to Query and Key. Consequently, we restrict block selection exclusively to the V-projection. However, as illustrated in Figure 4, while the V-projection is the primary target, specific high-utility blocks vary between tasks. The significant overlap implies that naive joint fine-tuning induces parameter conflict. We thus separate updates by consolidating common features within the shared subspace, while independently updating non-overlapping areas, preserving historical knowledge, and mitigating catastrophic forgetting.

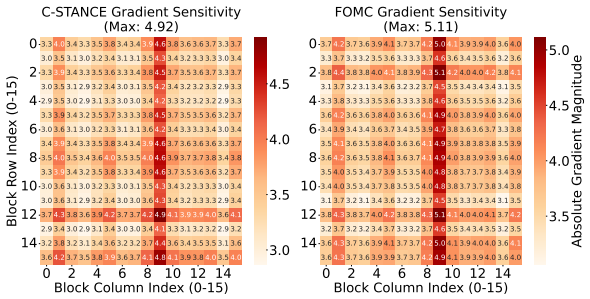

Figure 2: Average absolute gradient magnitudes. The 4096×4096 weight matrices are partitioned into a 16×16 grid ($l = 256$), revealing structured sparsity.

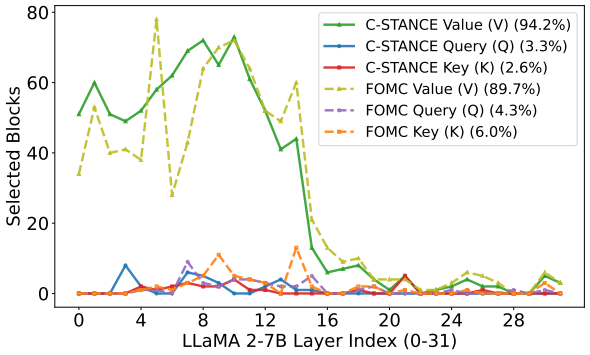

Figure 3: Layer-wise analysis shows ∼95% of top-ranked gradients concentrate in the Value (V) projection, indicating that attention routing (Q, K) remains stable while content (V) requires adaptation.

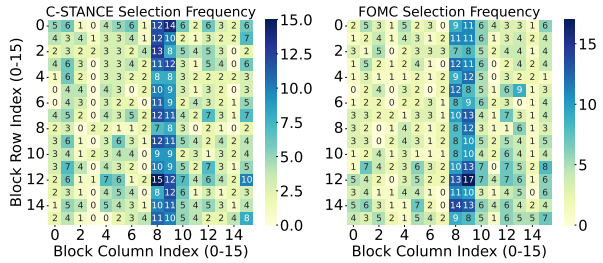

Figure 4: Aggregated selection frequency of the V-projection blocks. Distinct distributional shifts suggest potential for interference despite shared high activity.

3.2 Design of Unique and Shared Experts

To resolve parameter interference, we strictly partition the model capacity by extracting the task-specific sparse subspace $W^{(t)}$ defined by the active sparse indices $\mathcal{P}_t$ for the current task. We decompose this capacity into two distinct functional sets based on the index overlap between the current gradient topology $\mathcal{P}_t$ and the cumulative historical subspace $\mathcal{P}_{1:t-1}$. The task-specific unique experts E_{u} are constructed from the disjoint sparse subspace defined by indices $\mathcal{P}_t \setminus \mathcal{P}_{1:t-1}$. These parameters are exclusive to T_t and function as immutable memory components; to protect previously learned knowledge from being overwritten during optimization, we enforce $\nabla E_{\text{u}} = 0$ for all future tasks $k > t$. In contrast, the shared experts E_{s} occupy the intersection subspace defined by $\mathcal{P}_t \cap \mathcal{P}_{1:t-1}$. These parameters represent the structural feature overlap between tasks, serving as a basis for positive backward transfer. The effective sparse subspace W is thus formalized as the summation of the frozen base model W_0 and the functional deltas:

$$W = W_0 + \sum_{j \in \mathcal{E}_{\text{u}}} \Delta W_j + \sum_{i \in \mathcal{E}_{\text{s}}} \Delta W_i. \quad (3)$$

3.3 Split-on-Share (SoS) Expert Evolution

In continual learning, the emergence of overlapping sub-blocks between $\mathcal{P}_{t-1}$ and $\mathcal{P}_t$ necessitates structural evolution. We propose the Robust Split-on-Share (SoS) algorithm, which governs this process via a layer-wise topological filter designed to distinguish semantic commonality from stochastic noise. For each layer l, we first quantify the raw intersection $\mathcal{I}_l = \mathcal{P}_{t-1}^{(l)} \cap \mathcal{P}_t^{(l)}$ and the raw unique remainder $\mathcal{R}_l = \mathcal{P}_{t-1}^{(l)} \setminus \mathcal{P}_t^{(l)}$. To ensure robust modularity, we apply a two-stage conditional set operation using the Expert Creation Threshold (τ_{ect}) and the Tiny Remainder Threshold (τ_{trt}):

Filtering Small Overlaps. We validate if the intersection represents meaningful shared knowledge. If the overlap cardinality $|\mathcal{I}_l|$ is below τ_{ect}, we reject it as a stochastic coincidence and reintegrate the indices into the unique subspace:

$$\text{if } |\mathcal{I}_l| < \tau_{\text{ect}} \implies \mathcal{R}_l \leftarrow \mathcal{R}_l \cup \mathcal{I}_l, \quad \mathcal{I}_l \leftarrow \emptyset. \quad (4)$$

Merging Tiny Fragments. We analyze the sparsity of the remaining unique indices to prevent fragmentation. If the residual $|\mathcal{R}_l|$ falls below τ_{trt}, it indicates a negligible specialized structure. These indices are absorbed into the shared manifold to preserve computational efficiency:

$$\text{if } 0 < |\mathcal{R}_l| < \tau_{\text{trt}} \implies \mathcal{I}_l \leftarrow \mathcal{I}_l \cup \mathcal{R}_l, \quad \mathcal{R}_l \leftarrow \emptyset. \quad (5)$$

Expert Formation. Based on the filtered sets, the architecture executes the split. The validated intersection $\mathcal{I}_l$ instantiates the plastic shared expert E_{s}, utilizing weight inheritance $W_{\text{s}} \leftarrow W_{t-1}[\mathcal{I}_l]$. The stabilized residuals $\mathcal{R}_l$ form the frozen unique expert $E_{\text{u}}^{(t-1)}$. Finally, indices exclusive to the new domain $\mathcal{P}_t^{(l)} \setminus (\mathcal{I}_l \cup \mathcal{R}_l)$ initialize the New Task Expert $E_{\text{u}}^{(t)}$. This mechanism ensures model capacity expands logarithmically ($\mathcal{O}(\log T)$) by selectively merging redundant functionalities.

3.4 Historical Knowledge Preservation

Catastrophic forgetting is prevented through a dual strategy of subspace isolation to protect history and adaptive regularization to stabilize shared knowledge.

Orthogonal Subspace Isolation. We strictly divide the sparse subspace to isolate and protect historical memories. During the training of task t, all prior unique experts $E_{\mathrm{u}}^{(k)}$ $(k < t)$ are frozen, which is enforced via a zero-gradient constraint:

$$\forall w \in \bigcup_{k<t} \mathcal{E}_{\mathrm{u}}^{(k)}, \quad \frac{\partial \mathcal{L}}{\partial w} \equiv 0. \tag{6}$$

Optimization is restricted to the active topology: the plastic shared expert E_{s} and the current unique expert $E_{\mathrm{u}}^{(t)}$. This ensures that features specific to past domains remain immutable.

Adaptive Elastic Anchoring. While the Shared Expert E_{s} facilitates forward transfer, it is vulnerable to semantic drift. This is mitigated by applying adaptive regularization, positing that parameters utilized by multiple tasks represent fundamental invariants. Let $\Delta W_j \in \mathcal{E}_{\mathrm{s}}$ be a shared parameter with initial state ΔW_j^* inherited from the previous task. We define the regularization penalty $\mathcal{R}(\cdot)$ as:

$$\mathcal{R}(E_{\mathrm{s}}) = \sum_{j \in \mathcal{E}_{\mathrm{s}}} n_j \|\Delta W_j - \Delta W_j^*\|^2, \tag{7}$$

where, n_j denotes the sharing frequency. This penalty creates a potential well for highly shared weights, compelling the unconstrained Unique Expert $E_{\mathrm{u}}^{(t)}$ to absorb high-variance updates.

3.5 Adaptive Gating Expansion

To ensure the model remains stable when splitting the historical expert E_{t-1} into separate experts, we strictly preserve the pre-activation logits.

Logit Invariance via Weight Inheritance. Let the gating parameter for the active expert at task $t-1$ be denoted as $W_g^{(t-1)}$. For an input token $\mathbf{x}$, the routing logit is defined as $z = \mathbf{x}^\top W_g^{(t-1)}$. Upon executing the experts split operation, we initialize the gate for the plastic shared expert $W_g^{(\mathrm{s})}$ and the frozen task-specific Expert $W_g^{(\mathrm{u})}$. This constraint guarantees logit invariance, ensuring that the probability mass originally assigned to E_{t-1} is equivalently distributed across the new configuration $(E_{\mathrm{s}}, E_{\mathrm{u}}^{(t-1)})$. By keeping the original weight geometry, we preserve the router's existing decision boundaries. This prevents cold-start errors, where new, uncalibrated gates would otherwise confuse the model and disrupt performance on known tasks.

3.6 Objective Function

For the initial task T_1, we minimize the standard sparse fine-tuning loss. For subsequent tasks ($n > 1$), we optimize the active subspace of the new unique expert $E_{\mathrm{u}}^{(t)}$ and the shared expert E_{s} while the Base Model W_0 and past unique experts remain frozen. The composite objective is:

$$\mathcal{L} = -\sum_{(x,y)\sim\mathcal{D}_n} \log P(y \mid x; W_0 + \sum_{k \in \mathcal{E}_{\mathrm{u}}^{(t)}} \Delta W_k + \sum_{j \in \mathcal{E}_{\mathrm{s}}} \Delta W_j) + \lambda \sum_{j \in \mathcal{E}_{\mathrm{s}}} n_j \|\Delta W_j - \Delta W_j^*\|^2. \tag{8}$$

The first term optimizes the model's prediction accuracy using the active parameters, while the second term applies the stability constraint specifically to the shared expert E_{s}.

3.7 Task-Agnostic Inference

SETA achieves task-agnostic deployment via a Unified Content-Based Router, effectively treating continual learning as a dynamic sparse coding problem. The gating network acts as a unified linear transformation $W_g \in \mathbb{R}^{N \times d}$, mapping input $\mathbf{x}$ to a global logit vector over the expert population $N = |\mathcal{E}_{\mathrm{s}}| + |\mathcal{E}_{\mathrm{u}}|$. This single-layer architecture enables simultaneous evaluation, computing the output as a weighted superposition of the frozen backbone W_0 and retrieved sparse functional deltas:

$$\mathbf{y} = W_0\mathbf{x} + \sum_{k \in \mathcal{E}_{\mathrm{s}}} \sigma_k(\mathbf{x}W_g^\top)\Delta W_k\mathbf{x} + \sum_{k \in \mathcal{E}_{\mathrm{u}}} \sigma_k(\mathbf{x}W_g^\top)\Delta W_k\mathbf{x}, \tag{9}$$

where, $\sigma(\cdot)$ denotes the softmax operator. This formulation approximates complex decision boundaries via concurrent expert activation. If a token $\mathbf{x}$ shares features with multiple domains, W_g automatically triggers multi-expert activation, reconstructing functional capacity from the distributed parameter space without requiring discrete task IDs.

4 Experiments

4.1 Dataset Description

To rigorously evaluate LLM plasticity and retention, we constructed a dataset based on three principles. The first is Domain Specificity which focuses

Table 1: Summary of the results on domain-specific Continual Learning benchmarks with the LLaMA-2 7B. Averaged inference accuracy on the downstream tasks (Acc_t) is reported. Higher (↑) Acc_t value is better.

	Domain-specific Continual Learning benchmarks for LLMs					
Method	C-STANCE	FOMC	MeetingBank	ScienceQA	NumGLUE-cm	20Minuten
Seq-Train	41.8	41.1	31.2	27.4	21.9	13.6
ER	40.8	45.6	30.6	29.4	18.7	16.5
EWC	42.2	53.0	35.9	38.5	25.5	26.2
GEM	38.8	46.4	28.3	27.4	12.7	17.7
A-GEM	40.2	43.9	28.0	36.9	19.8	22.6
L2P	43.8	39.5	24.0	26.4	22.7	15.4
PP	37.2	42.1	26.5	28.3	25.3	25.9
I-LoRA	44.4	53.9	30.6	40.1	33.7	28.82
SETA	**46.80**	**59.00**	**33.21**	**36.92**	**36.25**	**30.47**

on unseen specialized data. The second is Diversity regarding varied formats and reasoning complexities. The third is Common Sense Preservation. Our evaluation involves two benchmark categories. The Continual Learning Benchmarks address the first two goals by incorporating specialized tasks across three verticals. These include ScienceQA for education, FOMC for finance, and MeetingBank for political discourse. We also utilize C-STANCE and 20Minuten for multilingual adaptation alongside NumGLUE for mathematical reasoning.

General Benchmarks for LLMs. To ensure the model retains its baseline capabilities, aligning with our third principle of common sense preservation, we employed widely recognized benchmarks for general knowledge and reasoning. We utilized MMLU (Hendrycks et al., 2021), BBH (Suzgun et al., 2022), and PIQA (Bisk et al., 2020). These datasets serve as a control group to measure extent of catastrophic forgetting on commonsense tasks.

4.2 Evaluation Metric

Let $M_{i,j}$ denote the inference accuracy on task j after training on task i. We assess the learning curve through the running Average Accuracy (Acc_t) and report the final Aggregate Retention Score (R_T):

$$Acc_t = \frac{1}{t}\sum_{j=1}^{t} M_{t,j}, \quad R_T = \frac{1}{T}\sum_{j=1}^{T} M_{T,j}. \quad (10)$$

We measure resistance to catastrophic forgetting using Average Forgetting (F_T), which quantifies the decline from the peak accuracy of each task:

$$F_T = \frac{1}{T-1}\sum_{j=1}^{T-1} \max_{l \in T}(M_{l,j} - M_{T,j}). \quad (11)$$

where, T is the total number of data sequences.

4.3 Baselines

We compare SETA against ten PEFT-based baselines to ensure a fair evaluation. The methods include (1) Zero-shot inference (ZSI) without tuning; (2) Sequence Fine-tuning (Seq-Train) which adapts continuously without regularization; (3) Experience Replay (ER) (Chaudhry et al., 2019b), utilizing a sample buffer; and (4) Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017), which restricts critical parameter changes via Fisher information. We also evaluate gradient-based constraints (5) GEM (Saha et al., 2021) and (6) A-GEM (Chaudhry et al., 2019a), alongside prompt-tuning strategies (7) L2P (Wang et al., 2022b) and (8) Progressive Prompt (PP) (Razdaibiedina et al., 2023). Finally, (9) Multi-task Learning (MTL) serves as an upper performance bound (Wang et al., 2023a), with (10) I-LoRA (Zhao et al., 2024) included as a direct architectural comparator.

4.4 Implementation Details

All experiments utilize two A100 GPUs with LLaMA-2 7B as the backbone. The learning rate is $1e$-4 with a 0.2 linear warmup. We use the HuggingFace Transformers library (Wolf et al., 2020). We sample 5,000 training and 500 evaluation instances per dataset, using a batch size of 16.

4.5 Comparative Analysis of CL

Evaluation of Domain Adaptation: We evaluated a LLaMA-2 7B model utilizing average inference accuracy Acc_t as the primary metric. The data in Table 1 confirms SETA establishes a new state of

Table 2: Final task-wise accuracy and aggregated metrics (R_T, F_T) after sequential training on 6 domain-specific tasks. SETA demonstrates a superior Aggregate Retention Score and stability compared to state-of-the-art methods.

	Final Sequential Learning Accuracy						Aggregated Metrics	
Method	T1	T2	T3	T4	T5	T6	A_T (↑)	F_T (↓)
Seq-Train	41.8	31.1	20.1	14.0	4.4	10.0	20.23	35.50
EWC	38.9	49.7	25.7	26.3	15.6	18.0	29.03	25.40
GEM	35.1	42.7	15.5	13.8	8.5	9.0	20.76	30.10
PP	32.0	13.1	9.3	40.8	19.8	40.6	25.93	34.60
I-LoRA	33.6	8.1	14.4	44.2	22.2	41.4	27.32	27.52
SETA	34.8	25.0	21.0	44.5	21.0	36.6	**30.47**	**19.28**

Table 3: Performance on General Benchmarks after Fine-Tuning on Domain-Specific CL Benchmarks.

Method	MMLU	BBH	PIQA	GEN loss
Zero-Shot	46.8	38.2	78.3	—
Seq	3.68	28.82	58.49	-24.1
ER	5.22	28.67	53.1	-25.41
EWC	14.27	34.18	51.85	-21.0
GEM	15.45	31.74	53.48	-20.88
A-GEM	6.46	32.44	53.92	-23.49
L2P	2.24	31.95	54.19	-24.97
PP	30.58	16.97	53.05	-20.9
I-LoRA	15.77	32.66	51.25	-21.21
SETA	**28.00**	**36.03**	**44.00**	**-18.42**
MTL	13.97	31.92	52.99	-20.47

the art by outperforming strong baselines like I-LoRA across most tasks. Notably SETA achieves 59.00 on the FOMC financial task and 46.80 on C-STANCE surpassing I-LoRA which reached 53.9 and 44.4 respectively. This 5.1% gain on FOMC indicates our sparse adaptation captures specialized knowledge better than replay methods. Additionally on reasoning intensive tasks like NumGLUE cm SETA maintains a clear lead 36.25 vs 33.7 confirming that leveraging geometric connectivity improves knowledge retention and mitigates catastrophic forgetting.

Aggregate Retention Score and Forgetting: Table 2 compares SETA against established baselines. While metrics such as Knowledge Retention often penalize high initial plasticity Aggregate Retention Score R_T demonstrates SETA retains a higher absolute volume of knowledge achieving 30.47% compared to 27.32% for I-LoRA. Notably SETA exhibits greater robustness in challenging domains where I-LoRA suffers catastrophic collapse on Task 2 dropping to 8.1% whereas SETA maintains 25.0%. This advantage extends to Task 3 where SETA preserves 21.0% accuracy versus 14.4% confirming the sparse update mechanism balances stability and plasticity without compromising final utility.

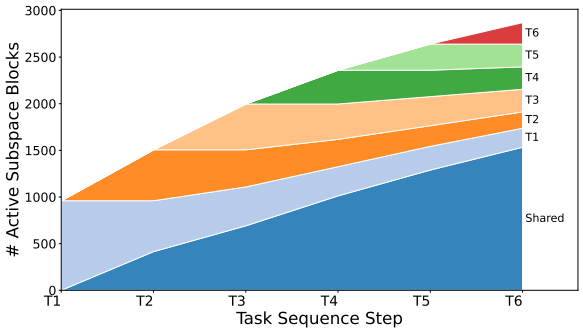

Figure 5: Subspace Evolution. The shared expert (blue) progressively absorbs universal features, while task-specific unique experts (colored bands) maintain stable, minimal footprints.

Analysis of Stability and Plasticity: Figure 6 tracks sequential performance where the diagonal highlights plasticity. SETA demonstrates superior consistency compared to baselines like PP and I-LoRA which struggle with forgetting. For instance after five tasks SETA retains 36.00% accuracy on Task 1 surpassing I-LoRA at 35.8% and PP at 16.4%. Additionally SETA achieves a better trade off by effectively learning new tasks scoring 73.60% on Task 2 and 46.50% on Task 4. This confirms leveraging mode connectivity ensures robust long term memory without sacrificing learning capability.

Impact on General Reasoning Capabilities: Assessing zero-shot results on MMLU, BBH, and PIQA in Table 3 ensures fine-tuning has not eroded

PP

Tr\Te	T1	T2	T3	T4	T5	T6
T1	37.2					
T2	32.4	51.8				
T3	33.2	25.4	21.1			
T4	32.4	24.2	14.7	44.2		
T5	16.4	49.0	7.9	26.2	27.2	
T6	32.0	13.1	9.3	40.8	19.8	40.6

I-LoRA

Tr\Te	T1	T2	T3	T4	T5	T6
T1	44.4					
T2	43.2	64.5				
T3	18.4	52.2	21.3			
T4	35.4	51.0	19.2	54.2		
T5	35.8	43.8	14.7	44.8	29.6	
T6	33.6	8.1	14.4	44.2	22.2	41.4

SETA

Tr\Te	T1	T2	T3	T4	T5	T6
T1	46.8					
T2	44.4	73.6				
T3	28.5	27.5	43.6			
T4	37.0	46.0	18.2	46.5		
T5	36.0	50.4	20.3	42.4	32.1	
T6	34.8	25.0	21.0	44.5	21.0	36.6

Figure 6: Task-wise performance of CL methods when LLaMA-2 7B is continually fine-tuned on the sequential tasks. The rows represent the Training Task (Train) and the columns represent the Evaluation Task (Test).

inherent reasoning. While some decline is expected, SETA demonstrates the highest overall resilience among all methods, securing the best GEN loss of -18.42. Although PP achieves a slightly higher MMLU score, it suffers a severe drop in BBH performance, whereas SETA successfully balances domain mastery.

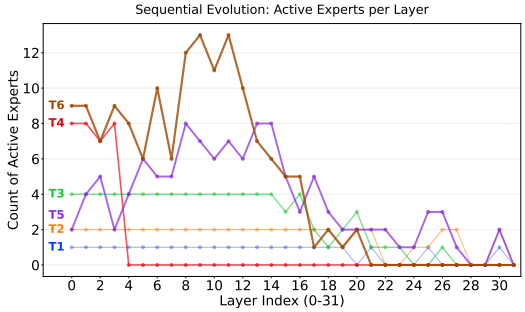

Figure 7: The number of active experts increases for later tasks, indicating the effective retrieval of accumulated historical knowledge.

4.6 Discussion and Mechanism Analysis

4.6.1 Subspace Split-on-Share Dynamics

Figure 5 illustrates the Merging Tiny Fragments strategy to show the physical migration of redundancy. The plot reveals a clear compositional shift where the shared expert blue becomes the dominant component while task-specific unique experts colored bands maintain minimal footprints. This trend validates that common features consolidate into the shared subspace rather than duplicating across experts, effectively compressing redundant representations.

4.6.2 Dynamic Capacity Expansion

To understand SETA's scalability, we analyzed the configuration of its adaptive modules, as shown in Figure 7. First, tracking Active Module Density across layers reveals a dynamic expansion of inference capacity: while early tasks use sparse routing, later tasks automatically recruit a denser mix of historical Shared Modules and new Unique Modules . This confirms the model expands its computational footprint only when necessary for task diversity.

Table 4: Parameter efficiency analysis showing the number of trainable parameters (#TP) in millions and their percentage relative to LLaMA-2 7B; I-LoRA (rank 8) requires approximately 8.39M parameters for reference.

Metrics	T1	T2	T3	T4	T5	T6
#TP (M) ↓	6.34	6.33	6.62	65.28	7.16	8.10
TP (%) ↓	0.98	0.98	1.03	1.01	1.11	1.25

4.6.3 Parameter Efficiency

In Table 4, our method maintains a minimal memory footprint. The trainable parameters (#TP) increase marginally from 6.34M (Task 1) to 8.10M (Task 6), representing just 0.98% to 1.25% of the total model capacity. Notably, this remains comparable to or lower than I-LoRA, which requires approximately 8.39M trainable parameters with rank 8. This confirms effective adaptation while modifying ≈1% of the parameters.

5 Conclusion

We presented SETA, a framework for mitigating catastrophic forgetting in large language models under continual learning. By separating task-specific and shared knowledge within a dynamically evolving Mixture-of-Experts architecture, SETA isolates conflicting updates while preserving reusable representations through elastic weight anchoring. Experimental results show that this structural decomposition enables task-agnostic inference and outperforms state-of-the-art continual learning baselines across diverse reasoning benchmarks. Future work includes scaling SETA to longer sequences and larger models while exploring adaptive growth.

Limitations

Due to computational constraints, our evaluation of SETA is limited to a single backbone model, LLaMA-2 7B. While this enables controlled analysis, extending SETA to larger models and alternative architectures remains an important direction for future work, particularly to understand how expert specialization and routing scale with model size. Our experiments also focus on a fixed sequence of domain-level tasks; broader task diversity and longer sequences may reveal additional behaviors of the framework. Finally, although SETA separates shared and task-specific parameters dynamically, the design of the shared subspace remains heuristic. Developing principled strategies to consolidate or shrink shared experts as tasks grow is an important avenue for future work.

Data and Model Usage

We conduct all experiments using publicly available large language models and benchmark datasets. Our primary backbone model is LLaMA-2-7B, which is licensed under Meta's community license permitting research and commercial use. All experiments are executed on a single NVIDIA A100 GPU using the HuggingFace Transformers library. We will release our implementation code and configuration files under an open-source license with detailed documentation to support reproducibility and responsible use. To evaluate continual learning performance, we employ a diverse set of publicly available datasets spanning multiple domains and reasoning formats. Our continual learning benchmarks include ScienceQA (education), FOMC (finance), and MeetingBank (political discourse), enabling domain-specific sequential adaptation. In addition, we use C-STANCE and 20Minuten to evaluate multilingual adaptation, alongside NumGLUE for mathematical reasoning. To assess retention of general knowledge and measure catastrophic forgetting, we additionally evaluate on widely used general benchmarks, including MMLU, BBH, and PIQA, which serve as control tasks for common-sense and reasoning capabilities. All datasets used in this study are publicly released under permissive research licenses, and no proprietary, sensitive, or personally identifiable data is included.

Environmental Impact

SETA improves parameter efficiency by activating only sparse subsets of parameters through its Mixture-of-Experts design, reducing the number of trainable weights compared to full fine-tuning. This selective adaptation lowers computational cost, training time, and energy consumption, making continual fine-tuning more resource-efficient and environmentally sustainable.

Societal Impacts

By enabling task-agnostic continual adaptation with less catastrophic forgetting, SETA lowers the barrier to deploying large language models in dynamic and evolving application settings. Its parameter-efficient design supports scalable model updates with reduced computational overhead, which may broaden access to adaptable language technologies. However, as with all continual learning systems, care must be taken to monitor potential misuse and unintended behavior accumulation across task sequences.

Bias and Fairness

SETA does not explicitly optimize for fairness and may inherit biases present in the sequential training data. Since shared experts capture reusable knowledge across tasks, biases introduced early in training could propagate to later tasks if not carefully managed. Future work should explore fairness-aware expert routing, bias auditing, and regularization strategies to mitigate disparities arising from task order or data imbalance.

Responsible Deployment

To support responsible use, we include clear documentation outlining the intended use cases of our framework and advise against applying it in safety-critical settings without thorough validation. We encourage users to follow ethical standards, such as the ACL Code of Ethics, when deploying our method. Our released code comes with usage instructions to promote safe adoption and reduce the risk of misuse. This work is licensed under CC BY 4.0, allowing reuse and adaptation, even commercially, with proper attribution.

AI Assistants in Research Writing

We used AI assistants to support writing and code refinement during the preparation of this paper. All AI-generated content was reviewed and verified by the authors.

References

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Tom Brown, Benjamin Mann, Nick Ryder, and 1 others. 2020. Language models are few-shot learners. *NeurIPS*.

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, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019a. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. 2019b. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*.

Matthias De Lange and 1 others. 2021. Continual learning with transformers. In *ICLR*.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. 2020. Rigging the lottery: Making all tickets winners. In *ICML*.

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.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *JMLR*.

Chrisantha Fernando and 1 others. 2017. Pathnet: Evolution channels gradient descent in super neural networks. In *arXiv preprint arXiv:1701.08734*.

Shangqian Gao, Ting Hua, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. 2024. Adaptive rank selections for low-rank approximation of language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 227–241.

Song Han, Jeff Pool, John Tran, and William J Dally. 2015. Learning both weights and connections for efficient neural networks. In *NeurIPS*.

Haoze He, Juncheng Billy Li, Xuan Jiang, and Heather Miller. 2025. SMT: Fine-tuning large language models with sparse matrices. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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 *ICML*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021a. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021b. Lora: Low-rank adaptation of large language models. *arXiv preprint*. ArXiv:2106.09685.

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

Yebowen Hu, Tim Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. Meetingbank: A benchmark dataset for meeting summarization. In *arXiv preprint arXiv:2305.17529*.

Tannon Kew, Marek Kostrzewa, and Sarah Ebling. 2023. 20 minuten: A multi-task news summarisation dataset for german. *arXiv preprint arXiv:2305.06555*.

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.

Dmitry Lepikhin, Yuanzhong Xu, Zhewei Huang, and 1 others. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. In *ICLR*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *EMNLP*.

Mike Lewis, Xiang Chen, Angela Fan, and 1 others. 2021. Base layers: Simplifying training of large, sparse models. In *ICML*.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhiyuan Cheng, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. 2024a. Mixlora: Enhancing large language models fine-tuning with lora based mixture of experts. *arXiv preprint arXiv:2404.15159*.

Han Li, Yuchen Xu, and 1 others. 2024b. Customizable combination of parameter-efficient modules for multi-task learning. In *Proceedings of EMNLP*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. In *ECCV*.

Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. 2023. Relora: High-rank training through low-rank updates. In *The Twelfth International Conference on Learning Representations*.

Haotian Liu, Yihan Zhang, and 1 others. 2024. Orthogonal subspace learning for language model continual learning. In *Proceedings of NeurIPS*.

Zefang Liu and Jiahua Luo. 2024. Adamole: Fine-tuning large language models with adaptive mixture of low-rank adaptation experts. *arXiv preprint arXiv:2405.00361*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during instruction tuning. *arXiv preprint arXiv:2308.08747*.

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

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. Numglue: A suite of fundamental yet challenging mathematical reasoning tasks. *arXiv preprint arXiv:2204.05660*.

Mistral AI. 2023. Mixtral of experts. arXiv preprint arXiv:2312.08389.

J. Pablo Muñoz, Jinjie Yuan, and Nilesh Jain. 2025. Low-rank adapters meet neural architecture search for llm compression. In *AAAI'25 workshop on CoLoRAI - Connecting Low-Rank Representations in AI*.

OpenAI. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71.

Colin Raffel and 1 others. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Hossein Rajabzadeh, Mojtaba Valipour, Tianshu Zhu, Marzieh Tahaei, Hyock Ju Kwon, Ali Ghodsi, Boxing Chen, and Mehdi Rezagholizadeh. 2024. Qdylora: Quantized dynamic low-rank adaptation for efficient large language model tuning. *arXiv preprint arXiv:2402.10462*.

Vinay V Ramasesh, Aitor Lewkowycz, and Ethan Dyer. 2022. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. In *The Eleventh International Conference on Learning Representations*.

Sylvestre-Alvise Rebuffi and 1 others. 2017. icarl: Incremental classifier and representation learning. In *CVPR*.

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

Stephen Roller, Anusha Pappu, and 1 others. 2021. Hash layers for large sparse models. In *NeurIPS*.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, and 1 others. 2016. Progressive neural networks. In *arXiv preprint arXiv:1606.04671*.

Gobinda Saha, Isha Garg, and Kaushik Roy. 2021. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*.

Agam Shah, Suvan Paturi, and Sudheer Chava. 2023. Trillion dollar words: A new financial dataset, task & market analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6664–6679.

Noam Shazeer, Azalia Mirhoseini, Piotr Maziarz, and 1 others. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. In *NeurIPS*.

Zhiqing Sun, Yiming Xie, Yizhou Wang, Hao Ma, Lingkai Kong, and Ming Zhou. 2024. Smt: Fine-tuning large language models with sparse matrices. In *ICLR*.

Yi-Lin Sung, Varun Nair, and Colin A Raffel. 2021. Training neural networks with fixed sparse masks. In *Advances in Neural Information Processing Systems*, volume 34, pages 24193–24205.

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*.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2022. Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *arXiv preprint arXiv:2210.07558*.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2023a. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*.

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*.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022a. Adamix: Mixture-of-adaptations for parameter-efficient model tuning. *arXiv preprint arXiv:2205.12410*.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Jiaming Wu, Jiawei Chen, and 1 others. 2024a. Slim: Let llms learn more and forget less with soft lora and identity mixture. In *Proceedings of ACL*.

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

Xun Wu, Shaohan Huang, and Furu Wei. 2024c. Mixture of lora experts. *arXiv preprint arXiv:2404.13628*.

Ehsan Xue and 1 others. 2024. Training decoder-only mixture-of-experts models with ul2 objectives. *arXiv preprint arXiv:2402.12345*.

Yifan Yang, Kai Zhen, Ershad Banijamal, Athanasios Mouchtaris, and Zheng Zhang. 2024. Adazeta: Adaptive zeroth-order tensor-train adaption for memory-efficient large language models fine-tuning. *arXiv preprint arXiv:2406.18060*.

Ben Zaken, Yoav Goldberg, and Amir Globerson. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL Findings*.

Chenye Zhao, Yingjie Li, and Cornelia Caragea. 2023. C-stance: A large dataset for chinese zero-shot stance detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13369–13385.

Kai Zhao, Jian Gao, and 1 others. 2024. Analyzing and reducing catastrophic forgetting in parameter-efficient tuning. In *Proceedings of ICLR*.

Changhai Zhou, Shijie Han, Shiyang Zhang, Shichao Weng, Zekai Liu, and Cheng Jin. 2024. Rankadaptor: Hierarchical dynamic low-rank adaptation for structural pruned llms. *arXiv preprint arXiv:2406.15734*.

Yanqi Zhou, William Fedus, Noam Shazeer, and 1 others. 2022. Mixture-of-experts with expert choice routing. In *NeurIPS*.

Appendix

A Related Works

A.1 Continual Learning

Continual learning (CL) seeks to train models on sequential tasks without catastrophic forgetting (McCloskey and Cohen, 1989), where newly acquired knowledge overwrites previously learned representations. Classical CL methods include regularization-based approaches such as EWC (Kirkpatrick et al., 2017) and LwF (Li and Hoiem, 2017), rehearsal-based replay (Rebuffi et al., 2017; Shin et al., 2017), and architectural expansion (Rusu et al., 2016; Fernando et al., 2017), though these often require large memory buffers or suffer from scalability issues. Recent advances extend CL to Transformer-based and large language models (LLMs) using parameter-efficient tuning (PEFT) modules. For instance, I-LoRA (Zhao et al., 2024) mitigates forgetting by maintaining dual LoRA adapters, a fast (short-term) and a slow (long-term) memory, interpolated to balance plasticity and stability. O-LoRA (Liu et al., 2024) further enforces orthogonal subspace constraints

between LoRA modules of different tasks, preventing interference in the representation space. SLiM (Wu et al., 2024a) introduces a soft mixture of LoRA and identity mappings, dynamically weighting learned and frozen components to enhance retention while preserving efficiency. Meanwhile, CC-PEFT (Li et al., 2024b) generalizes modular parameter-efficient tuning by combining multiple PEFT adapters in a customizable manner for multi-task and continual scenarios.

A.2 Parameter-Efficient Fine-Tuning (PEFT)

Fine-tuning large language models (LLMs) is computationally demanding due to their massive parameter counts. Parameter-efficient fine-tuning (PEFT) methods address this challenge by updating only a small subset of parameters while keeping pre-trained weights frozen (Li and Liang, 2021; He et al., 2021; Wang et al., 2022a). A variety of PEFT techniques have been proposed, including Adapter Tuning (Houlsby et al., 2019), BitFit (Zaken et al., 2022), Prefix Tuning (Li and Liang, 2021), Prompt Tuning (Lester et al., 2021), and Low-Rank Adaptation (LoRA) (Hu et al., 2021b). Extensions such as ReLoRA (Lialin et al., 2023) and RankAdapter (Zhou et al., 2024) improve memory efficiency and dynamically adjust ranks, though they lack formal guarantees. AdaZeta (Yang et al., 2024) introduces zeroth-order optimization with convergence guarantees, while other methods (Gao et al., 2024; Rajabzadeh et al., 2024; Valipour et al., 2022) explore adaptive-rank strategies without theoretical proofs. LoRA has also been combined with Mixture-of-Experts architectures (Li et al., 2024a; Wu et al., 2024c), as in AdaMoLE (Liu and Luo, 2024), enabling dynamic expert selection, and with Neural Architecture Search for LLM compression (Muñoz et al., 2025). Another line of work leverages sparsity, such as Sparse Matrix Tuning (SMT) (Sun et al., 2024), which selects task-relevant sub-matrices based on gradient magnitude, offering both efficiency and interoperability.

A.3 Mixture-of-Experts (MoE)

Traditional dense models process every input token with all parameters, making computation scale linearly with model size. Mixture-of-Experts (MoE) improves efficiency by activating only a subset of parameters, first shown effective in LSTMs (Shazeer et al., 2017) and later integrated into Transformers (Lepikhin et al., 2020; Fedus et al., 2021). Subsequent work addressed load imbalance with routing strategies (Lewis et al., 2021; Roller et al., 2021; Zhou et al., 2022), and recent decoder-only variants such as UL2-MoE and Mixtral further advanced scalability (Xue et al., 2024; Mistral AI, 2023).

B SETA Algorithm

Algorithm 1 SETA: Mixture of Sparse Experts Training Procedure

Require: Pre-trained W_0, Task Sequence $\mathcal{T} = \{T_1, \ldots, T_k\}$, Thresholds $\tau_{\text{ect}}, \tau_{\text{trt}}$, Reg λ

1: **Initialize:** Expert Registry $\mathcal{E} \leftarrow \emptyset$, Gating Network $W_g \leftarrow W_0$
2: **for** $t = 1$ to k **do**
3: **Subspace Selection:** Identify active sparse blocks $\mathcal{P}_t \leftarrow$ Top-$k(\mathcal{S}_t)$ via gradient sensitivity.
4: **if** $t = 1$ **then**
5: **Init Task 1:** Construct initial unique expert $E_{\text{u}}^{(1)}$ on $\mathcal{P}_1$.
6: **else**
7: **for** each layer $l \in L$ **do**
8: **SoS Decomposition:** Compute Intersection $\mathcal{I}_l$ and Remainder $\mathcal{R}_l$ using topological filters (Eq. 4 & 5).
9: **Expert Allocation:** Update plastic Shared Expert $E_{\text{s}} \leftarrow \mathcal{I}_l$; Frozen Unique Expert $E_{\text{u}}^{(t-1)} \leftarrow \mathcal{R}_l$.
10: **New Expert:** Initialize $E_{\text{u}}^{(t)}$ for novel features in $\mathcal{P}_t \setminus \mathcal{P}_{1:t-1}$.
11: **end for**
12: **end if**
13: **Adaptive Gating Expansion:** Resize W_g to match $|\mathcal{E}|$; maintain logit invariance.
14: **Freeze History:** Set $\nabla E_{\text{u}}^{(k)} = 0$ for all $k < t$.
15: **Optimize:** Minimize $\mathcal{L}_{\text{SETA}}$ with Elastic Weight Anchoring on E_{s} (Eq. 8).
16: **Anchor Update:** Update reference weights W_{s}^* for the shared subspace.
17: **end for**
18: **Inference:** Dynamic expert retrieval via Eq. 9.

C Empirical Analysis and Design Choices

Algorithm 1 details the training procedure for SETA. For each task t, we first identify the active sparse subspace $\mathcal{P}_t$ using gradient sensitivity as shown in Line 3. For $t > 1$, the Split-on-Share mechanism described in Lines 6-9 decomposes this

Table 5: Evolution of uniqueness statistics of Sub-block 980. This setting maintains task-specific unique blocks by dynamically allocating shared capacity as new tasks are added.

Sequence	Task Added	Total Blocks	Total Shared	Unique Blocks Remaining per Task					
				T1	T2	T3	T4	T5	T6
1	C-STANCE	960	0	960	0	0	0	0	0
2	FOMC	1506	414	546	546	0	0	0	0
3	MeetingBank	1997	691	417	398	491	0	0	0
4	ScienceQA	2359	1012	313	293	379	362	0	0
5	NumGLUE-cm	2639	1289	254	221	312	283	280	0
6	20Minuten	2869	1533	204	175	243	240	244	230

Table 6: Detailed Uniqueness Stats of Sub-block 1280. Shows the evolution of total blocks, shared capacity, and task-specific unique blocks as the complexity of the task sequence increases.

Step	Task Added	Total Blocks	Total Shared	Unique Blocks Remaining per Task					
				T1	T2	T3	T4	T5	T6
1	C-STANCE	1267	0	1267	0	0	0	0	0
2	FOMC	1938	600	667	671	0	0	0	0
3	MeetingBank	2530	997	484	457	592	0	0	0
4	ScienceQA	2943	1415	358	325	432	413	0	0
5	NumGLUE-cm	3279	1744	284	250	342	323	336	0
6	20Minuten	3531	2039	229	192	267	262	290	252

subspace by comparing it with history. Specifically, overlapping parameters $\mathcal{I}_l$ are merged into the plastic Shared Expert E_{s}, while disjoint parameters $\mathcal{R}_l$ form the frozen unique experts E_{u}. To enable task-agnostic inference, we perform Adaptive Gating Expansion in Line 11, resizing the router to accommodate new experts while preserving logit invariance. Finally, the model is optimized using Elastic Weight Anchoring in Line 12 to stabilize the shared subspace against catastrophic forgetting.

C.1 Attention Sub-block Scaling Analysis

The following analysis compares the evolution of sub-block allocation between the top 980 sub-blocks and 1280 across all 6 tasks. Despite a 23.1% increase in total block capacity at Step 6 for the 1280 sub-block, the shared expert expansion dominates, resulting in marginal gains for task-specific uniqueness. Key observations:

Increasing the threshold from 980 to 1280 primarily drives shared dominant scaling where shared blocks grow by 33.0 percent from 1533 to 2039 as detailed in Table 5 and Table 6. This growth is significant when compared to the 23.1 percent increase in total blocks observed at Step 6. Despite higher subblock allocation uniqueness remains stable as Task 1 unique blocks at Step 6 only rise from 204 to 229 representing an approximate 12 percent increase. The shared-to-total ratio consistently scales from approximately 28–31 percent at Step 2 to between 53 and 58 percent at Step 6 across both thresholds as illustrated in Table 7. This trend suggests that task synergy, rather than block volume, governs knowledge consolidation, even as complexity increases with datasets such as 20Minuten.

Optimizing Sparsity Thresholds The definition of what constitutes a unique versus a shared block is governed by the gradient ranking threshold. A relaxed threshold allows more blocks to be selected, increasing model capacity but also increasing the probability of accidental overlap between tasks. A strict threshold selects fewer blocks, resulting in higher task uniqueness. We compared a strict threshold of 980 blocks against a relaxed threshold of 1280 blocks.

Table 7: Full Sequence Comparison: sub-block 980 vs. 1280. Shared capacity grows at a higher rate than unique capacity in both settings, leading to similar shared-to-total ratios.

		Sub-block 980			Sub-block 1280		
Sequence	**Task Added**	**Total**	**Shared**	**Shared %**	**Total**	**Shared**	**Shared %**
1	C-STANCE	960	0	0.0%	1267	0	0.0%
2	FOMC	1506	414	27.5%	1938	600	31.0%
3	MeetingBank	1997	691	34.6%	2530	997	39.4%
4	ScienceQA	2359	1012	42.9%	2943	1415	48.1%
5	NumGLUE-cm	2639	1289	48.8%	3279	1744	53.2%
6	20Minuten	2869	1533	53.4%	3531	2039	57.7%

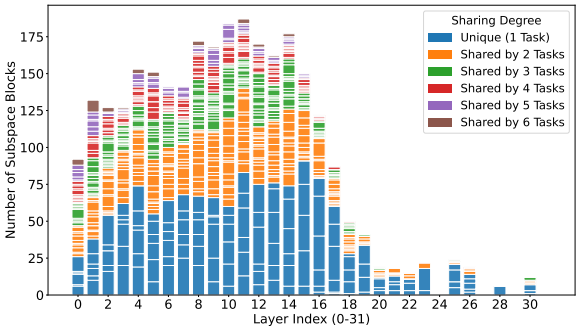

Figure 8: Subspace Compositional. Breakdown of parameter utility showing the emergence of universal blocks alongside task-specific unique blocks.

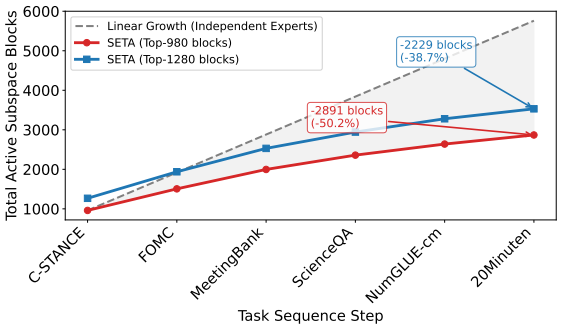

Figure 9: Parameter Growth Comparison. SETA demonstrates sub-linear growth, achieving over 50% reduction in active blocks compared to independent experts by the final task.

C.2 Strategic Component Selection and Layer Pruning

Our framework maximizes parameter efficiency by focusing expert creation on attention Value projection matrices, which are essential for knowledge movement and storage, while omitting Multi-Layer Perceptron (MLP) blocks due to their dense update requirements. We further refine the architecture by pruning layers where submatrix selection falls below a critical importance threshold, leveraging residual connections to bypass ineffective computations. This targeted selection drives the emergence of the compositional subspace illustrated in Figures 3 and 8, where computational resources are concentrated on high-impact, shared blocks and persistent, task-specific, unique blocks. By eliminating low-utility layers, primarily in the deeper sections of the model, we significantly reduce memory overhead and inference latency without degrading the predictive accuracy established through the sequential learning process.

C.3 Module Compositional

We examined the Parameter Reusability Spectrum in Figure 8 to quantify the number of tasks that utilize specific latent structures. The layer-wise decomposition reveals that middle layers evolve into high-utility shared centers utilized by more than 4 tasks, while specialized features remain preserved in the upper layers. This validates that SETA achieves efficient scaling not by mere memorization, but by composing new skills from a granular library of reusable functional regions.

C.4 Efficiency of Split-on-Share Evolution

We verify the claim that SoS enables logarithmic capacity expansion ($\mathcal{O}(\log T)$). Figure 9 quantifies this efficiency by contrasting the linear accumulation of independent experts against SETA's sub-linear trajectory. The plot reveals a distinct deceleration where expert instantiation slows significantly after the initial tasks. By the last task, this divergence results in a 53.6% reduction in active blocks compared to the baseline, physically proving the system has transitioned from instantiating

Table 8: Task-wise performance of CL methods when Llama-2-7B is continually fine-tuned on the sequential tasks. The rows represent the model's state after training on a specific dataset, while the columns represent the evaluation score on each specific test set.

	Evaluation Task (Test)					
Train \ Test	**C-STANCE**	**FOMC**	**MeetingBank**	**ScienceQA**	**NumGLUE-cm**	**20Minuten**
C-STANCE	**46.80**	50.00	17.48	49.20	12.35	35.35
FOMC	44.40	**73.60**	22.84	50.80	14.81	35.38
MeetingBank	28.50	27.50	**43.64**	43.00	11.12	35.75
ScienceQA	37.00	46.00	18.19	**46.50**	17.28	35.61
NumGLUE-cm	36.00	50.40	20.33	42.40	**32.10**	35.63
20Minuten	34.80	25.00	20.96	44.50	20.99	**36.57**

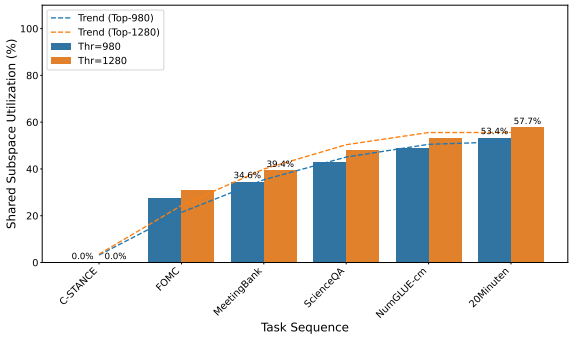

Figure 10: Shared Subspace Utilization. Utilization follows a logarithmic trend, peaking at 64.3% for the Top-1280 configuration, indicating robust knowledge consolidation.

new parameters to reusing consolidated knowledge.

C.5 Shared Subspace Utilization Trends

The effectiveness of Elastic Anchoring is evidenced by the utilization trends in Figure 10. The data shows a logarithmic convergence, where utilization rises consistently to peak at 64.3% for the Top-1280 configuration. This upward trend proves that the shared expert stabilizes into a robust semantic anchor as the sequence progresses. Furthermore, the consistently higher utilization in the Top-1280 configuration indicates that allocating larger shared capacity incentivizes deeper feature merging, enhancing the model's structural efficiency.

C.6 Analysis of Sequential Performance

Table 8 tracks the accuracy of each task throughout the training sequence. The results show that memory retention varies depending on how similar the tasks are. We see that learning Task 3 and Task 5 leads to a drop in performance on earlier tasks T1 and T2 likely because their learning goals conflict. In contrast Task 4 helps recover lost knowledge raising Task 2 accuracy from 22.84% to 50.80%. This indicates that the model can successfully restore past information when a new task is compatible with previous ones even after learning unrelated content.

D Baseline Descriptions

To rigorously evaluate SETA, we benchmark against ten diverse methods spanning regularization, replay, and architecture-based Continual Learning (CL) strategies. All methods employ identical LoRA architectures and parameter budgets to ensure fair comparison.

Lower and Upper Bounds:

- Zero-shot inference (ZSI): Establishes a lower performance bound by performing direct inference on sequential tasks without any parameter adaptation or prompting.
- Multi-task Learning (MTL): Trains on the combined dataset of all tasks simultaneously. This establishes the theoretical upper performance bound (Wang et al., 2023a) by removing the sequential constraint entirely.

Traditional CL Strategies:

- Sequence Fine-tuning (Seq-Train): Represents naive adaptation, continuously tuning parameters on the task sequence without regularization or replay.
- Experience Replay (ER) (Chaudhry et al., 2019b): Mitigates forgetting by preserving

a small buffer of prior training samples to interleave with current task data.

- Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017): A regularization-based approach that utilizes the Fisher information matrix to penalize changes to parameters deemed critical for previous tasks.

- Gradient Episode Memory (GEM) (Saha et al., 2021): Maintains a gradient subspace of old tasks, projecting updates orthogonally to avoid interference.

- Average Gradient Episode Memory (A-GEM) (Chaudhry et al., 2019a): improves upon GEM's efficiency by constraining updates using only the historical average gradient matrix.

Architecture and Prompt-Based Methods:

- Learning to Prompt (L2P) (Wang et al., 2022b): Dynamically selects and updates prompts from a shared pool based on input instance keys, keeping the backbone frozen.

- Progressive Prompt (PP) (Razdaibiedina et al., 2023): Sequentially learns task-specific soft prompts and concatenates them, isolating task knowledge in distinct prefix tokens.

- I-LoRA (Zhao et al., 2024): A relevant comparative baseline that utilizes dual-memory LoRA adapters for sequential learning.

E Detailed Dataset Specifications

To ensure a robust evaluation of both adaptation and forgetting, we utilized the following specific datasets categorized by their intended evaluation metrics.

E.1 CL Benchmarks for LLMs

We selected datasets to maximize domain specificity and task diversity across three primary dimensions. First, regarding **Domain Specificity**, we sourced datasets from five distinct verticals to simulate realistic and diverse downstream applications. These include ScienceQA (Lu et al., 2022) for the educational domain, FOMC (Shah et al., 2023) for financial forecasting, and MeetingBank (Hu et al., 2023) for political discourse. Second, regarding **Multilinguality**, we addressed the hurdles of vocabulary variations by following established protocols (Wang et al., 2023b) and incorporating C-STANCE (Zhao et al., 2023) and 20Minuten (Kew et al., 2023) to evaluate the bridging of linguistic gaps. Finally, for **Mathematical Reasoning**, we leveraged the NumGLUE dataset (Mishra et al., 2022) as a rigorous test bed for updating complex arithmetic and symbolic logical operations.

E.2 General Benchmarks for LLMs

To ensure the model retains its baseline capabilities, which corresponds to the third goal, we employed widely recognized benchmarks for general knowledge and reasoning. Specifically, we utilized MMLU (Hendrycks et al., 2021), BBH (Suzgun et al., 2022), and PIQA (Bisk et al., 2020). These datasets serve as a control group to measure the extent of catastrophic forgetting on common sense tasks.