Sparse Subspace-to-Expert Sharing 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 (SETA), a framework that resolves the plasticity-stability conflict through adaptive sparse subspace decomposition into task-specific expert modules. 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 adaptive elastic anchoring and a routing-aware regularization that jointly protect shared knowledge at both the weight and routing levels and enable a unified gating network to automatically retrieve the correct expert combination during inference. Extensive experiments across diverse domain-specific benchmarks demonstrate that SETA achieves competitive or superior overall performance relative to state-of-the-art continual learning baselines, with particularly strong retention of early-task knowledge and improved backward transfer on LLaMA-2 7B and Qwen3-4B. 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). However, deploying LLMs in continual learning settings remains challenging (Parisi et al., 2019) 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., 2019), 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 (Li et al., 2025a). As a result, they fail to distinguish between generalizable features and task-specific requirements, leading to suboptimal trade-offs between stability and plasticity (Li et al., 2025b).

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 representations? 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 decomposed into distinct expert modules, ensuring that conflicting updates operate in isolated subspaces rather than competing for overlapping parameter blocks.

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 dynamically assigns parameters based on their recurrence across tasks. Overlapping parameters are designated as shared experts, protected through adaptive elastic anchoring and a routing-aware regularization that jointly prevent semantic drift without introducing additional hyperparameters, while non-overlapping parameters become unique experts that are strictly frozen as immutable memory to preserve historical knowledge.

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, which preserves existing routing decision boundaries during expert splitting, and a routing-aware regularization that shapes the gate's routing decisions during training to reduce interference with shared experts. This allows the router to evolve alongside the experts and dynamically select relevant parameters based solely on input tokens, enabling task-agnostic inference without explicit task labels.

Section 2 formulates the problem, Section 3 introduces SETA and the Split-on-Share mechanism, Section 4 presents experiments, Section 5 related works and Section 6 concludes.

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., 2024), creating a fundamental plasticity-stability trade-off. Full fine-tuning offers high plasticity but is computationally expensive and disrupts the optimization landscape, leading to catastrophic forgetting (Luo et al., 2025). To address this, parameter-efficient fine-tuning (PEFT) methods such as LoRA freeze the backbone and train small low-rank modules (Hu et al., 2022). However, restricting updates to a narrow bottleneck dimension ($r \ll d$) creates a geometric disconnect between sequential task minima (Li et al., 2025a), introducing noise that hinders knowledge retention over long task sequences and causes a plasticity-stability failure where new learning overwrites historical parameters.

2.2 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):

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

where γ 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 (Lopez-Paz and Ranzato, 2017). Since static penalties cannot simultaneously prevent semantic drift and enable new learning (Farajtabar et al., 2020), resolving this trade-off requires both a structural solution to isolate parameter subspaces and an adaptive regularization strategy that responds to each expert's accumulated task history and live weight displacement.

2.3 Mixture of Sparse Experts for CL

We propose SETA, a modular framework for sparse subspace adaptation. SETA resolves the stability–plasticity conflict by adaptively decomposing task-specific sparse subspaces into unique and shared experts, preventing parameter overwriting. When subspaces overlap beyond a layer-wise significance threshold, the Split-on-Share mechanism decouples contested regions, enabling stable structural evolution. Combined with adaptive elastic anchoring, routing-aware regularization, and logit-invariant gating expansion, this design supports

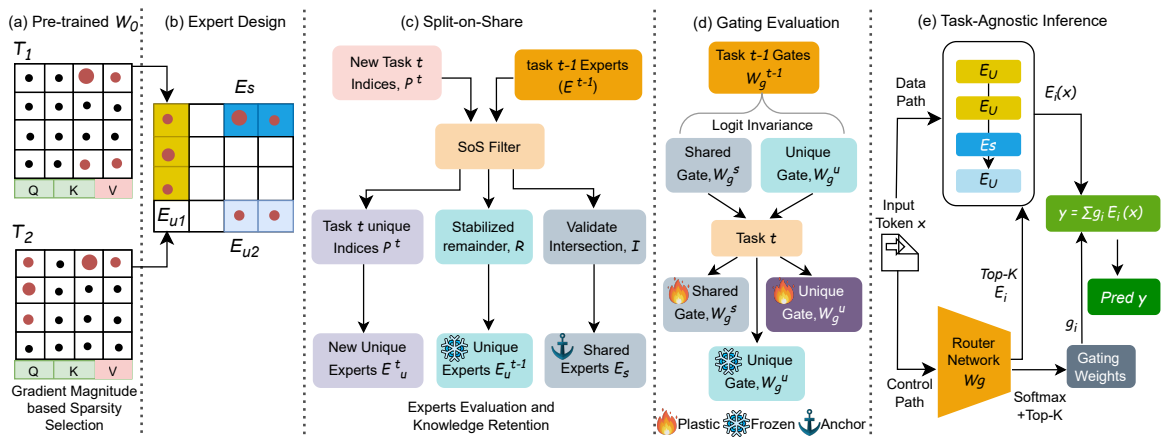

Figure 1: **Overview of the SETA Framework Architecture. (a–b) Sparse Subspace Selection:** High-utility parameter blocks are identified from the pre-trained LLM using gradient magnitude to form the expert design. **(c) Split-on-Share (SoS) Evolution:** The SoS filter partitions parameters into plastic Shared (E_{s}) and frozen Unique (E_{u}) experts to resolve parameter collisions and retain knowledge. **(d) Gating Evaluation:** The gating network expands using logit invariance to strictly preserve decision boundaries during expert splitting. **(e) Task-Agnostic Inference:** A router network dynamically weights all experts via softmax for input tokens, enabling automatic task processing without task identifications.

cumulative learning and task-agnostic inference without explicit task identifiers.

3 Methodology

3.1 Sparse Subspace Selection

We aim to fine-tune only the most task-relevant sparse sub-matrices, defining the subspace as a block-sparse optimization boundary through gradient-driven selection, rather than a conceptually disentangled semantic latent space. We partition pre-trained weight matrices $W \in \mathbb{R}^{d\times m}$ into a grid of fixed-size sub-blocks $\mathbb{B} = \{B_{i,j}\}$ of dimension $l \times l$, where l is determined per architecture as the greatest common divisor of the model's matrix dimensions. This architecture-aware sizing prevents remainder mismatches, ensures contiguous memory alignment, and maximizes hardware utilization across all supported model families. We identify high-utility regions via a parameter-free scoring pipeline executed during a warm-up phase before each task.

Dynamic Warm-up Stopping: We monitor the per-step gradient contribution ratio: Training halts when $\rho(t)$ falls below its own coefficient of variation CV $= \sigma_\rho/\mu_\rho$ over a sliding window of $W = 10$ steps, adapting warm-up length to each task's gradient dynamics.

$$\rho(t) = \frac{\|\nabla W\|_t}{\sum_{s\le t}\|\nabla W\|_s}. \tag{2}$$

Layer-Normalised Scoring: The raw block importance score is computed as the mean absolute gradient within each block:

$$\mathcal{S}_{i,j}^{(k)} = \frac{1}{l^2}\sum_{(u,v)\in B_{i,j}} |\nabla W_{u,v}| \tag{3}$$

where $\nabla W_{u,v}$ indicates parameter sensitivity. Since gradient magnitudes vary across layers, each layer's score matrix is standardized to zero mean and unit variance before global ranking, preventing dominant layers from monopolizing the block selection.

Cross-Task Divergence Scoring: For tasks $t > 1$, the normalized score is augmented with a divergence signal rewarding blocks with novel gradient response relative to prior tasks. The final score is a convex combination:

$$\tilde{\mathcal{S}}_t = \alpha \cdot \hat{\mathcal{S}}_t + (1-\alpha)\cdot\left|\hat{\mathcal{S}}_t - \bar{\mathcal{S}}_{1:t-1}\right|, \tag{4}$$

where $\bar{\mathcal{S}}_{1:t-1}$ is the mean block score over all prior tasks and $\alpha = 1/(1+\log t)$ decays automatically with task count, introducing no additional hyperparameter.

Adaptive Block Budget: The block budget is parameterised as a fraction ρ of total available blocks, analogous to a LoRA rank, giving $N_{\text{base}} = \lfloor \rho \cdot |\mathbb{B}| \rfloor$, with per-layer quota allocated inversely proportional to historical coverage. Selection proceeds beyond N_{base} until the marginal rate

of previously unselected blocks stabilizes, allowing later tasks to naturally expand their budget without manual adjustment.

As shown in Figure 7 (Appendix C.3), ∼95% of high-magnitude gradients concentrate in the Value (V) projection, attributed to Softmax Saturation where Q and K gradients vanish as $QK^\top/\sqrt{d_k}$ grows. We therefore restrict block selection to the V-projection. Since subspaces inevitably overlap across tasks, we consolidate common features into shared experts while isolating task-specific regions, mitigating catastrophic forgetting.

3.2 Design of Unique and Shared Experts

We partition the sparse subspace $\mathcal{P}_t$ into two functional sets based on its overlap with 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. 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:

$$W = W_0 + \sum_{j \in \mathcal{E}_{\text{u}}} \Delta W_j + \sum_{i \in \mathcal{E}_{\text{s}}} \Delta W_i. \tag{5}$$

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}):

Design choice of Expert Creation and Tiny Remainder Threshold: The Expert Creation Threshold (τ_{ect}) and Tiny Remainder Threshold (τ_{trt}) are fixed topological thresholds that filter structural noise rather than scaling continuous penalties. Their primary purpose is to distinguish stable semantic commonality from stochastic gradient noise. Specifically, τ_{ect} prevents the formation of diluted experts by rejecting parameter intersections that lack sufficient cardinality to represent a meaningful functional unit. Conversely, τ_{trt} mitigates architectural fragmentation by absorbing negligible unique residuals into the shared manifold, thereby preserving computational efficiency and gating stability.

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:

$$\text{if } |\mathcal{I}_l| < \tau_{\text{ect}} \implies \mathcal{R}_l \leftarrow \mathcal{R}_l \cup \mathcal{I}_l, \quad \mathcal{I}_l \leftarrow \emptyset. \tag{6}$$

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:

$$\text{if } 0 < |\mathcal{R}_l| < \tau_{\text{trt}} \implies \mathcal{I}_l \leftarrow \mathcal{I}_l \cup \mathcal{R}_l, \quad \mathcal{R}_l \leftarrow \emptyset. \tag{7}$$

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 enables sub-linear capacity expansion by selectively merging redundant functionalities. Empirically, we observe that this growth follows a logarithmic trend; however, we note that this scaling behavior is a heuristic observation rather than a strictly proven theoretical bound.

3.4 Catastrophic Forgetting Prevention

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 $E_{\text{u}}^{(k)}$ ($k < t$) are frozen, which is enforced via a zero-gradient constraint:

$$\forall w \in \bigcup_{k<t} \mathcal{E}_{\text{u}}^{(k)}, \quad \frac{\partial \mathcal{L}}{\partial w} \equiv 0. \tag{8}$$

Optimization is restricted to the active topology: the plastic shared expert E_{s} and the current unique

expert $E_{\text{u}}^{(t)}$. 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. We mitigate this with a per-expert dynamic regularisation coefficient that adapts at every training step based on two signals: the accumulated task history of the expert and its live weight displacement.

Let $\hat{W}_i$ be the snapshot of shared expert i taken before task k begins. The accumulation weight $\omega_i = n_{\text{prior}}^{(i)}/(k-1)$ scales protection proportionally to the number of prior task IDs encoded in expert i, while the drift factor $\delta_i(t)$ monitors live weight displacement relative to the pre-task snapshot:

$$\delta_i(t) = 1 + \tanh\left(\frac{\|W_i^{(t)} - \hat{W}_i\|}{\max(\|\hat{W}_i\|,\, \epsilon)}\right), \quad (9)$$

yielding the per-expert dynamic regularisation coefficient:

$$\lambda_i(t) = \lambda_{\text{base}} \cdot \omega_i \cdot \delta_i(t), \quad (10)$$

where $n_{\text{prior}}^{(i)}$ is the number of prior task IDs encoded in expert i. The drift factor $\delta_i(t) \in [1, 2)$ is non-decreasing as the expert drifts ($\delta_i = 1$ when $W_i^{(t)} = \hat{W}_i$), self-correctively strengthening the penalty in proportion to the displacement magnitude. The regularization loss is:

$$\mathcal{L}_{\text{reg}} = \sum_{i \in \mathcal{E}_{\text{s}}} \lambda_i(t) \cdot \|W_i^{(t)} - \hat{W}_i\|^2. \quad (11)$$

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 $W_g^{(\text{s})}$ and the frozen task-specific Expert $W_g^{(\text{u})}$. This approximately preserves the routing logits for both successors, with a small perturbation added to $W_g^{(\text{u})}$ to break symmetry. By keeping the original weight geometry, we preserve the router's existing decision boundaries, preventing cold-start errors after the split.

3.6 Routing-Aware Gating Regularization

While $\mathcal{L}_{\text{reg}}$ acts on expert weights after gradient updates have occurred, we additionally shape the gating network's routing decisions before drift accumulates by penalizing high routing probability toward at-risk shared experts:

$$\mathcal{L}_{\text{gate}} = \sum_{i \in \mathcal{E}_{\text{s}}} \lambda_i \cdot \mathbb{E}_x[\sigma(g_\theta(x))_i]\,, \quad (12)$$

where $\sigma(g_\theta(x))_i$ is the softmax routing probability to expert i and λ_i is the same coefficient as in $\mathcal{L}_{\text{reg}}$, introducing no additional hyperparameter. Together, $\mathcal{L}_{\text{reg}}$ and $\mathcal{L}_{\text{gate}}$ form a two-layer defence: the gate is steered away from risky routing decisions before drift accumulates, while weight regularization catches any residual drift.

3.7 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 $E_{\text{u}}^{(t)}$ and the shared expert E_{s} while the Base Model W_0 and past unique experts remain frozen. The composite objective is:

$$\mathcal{L} = -\sum_{(x,y)\sim\mathcal{D}_t} \log P(y \mid x; \Theta_t) + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{gate}}, \quad (13)$$

where $\Theta_t = W_0 + \sum_{k \in \mathcal{E}_{\text{u}}^{(t)}} \Delta W_k + \sum_{j \in \mathcal{E}_{\text{s}}} \Delta W_j$ denotes active parameter configuration at task t.

3.8 Task-Agnostic Inference

SETA achieves task-agnostic deployment via a unified content-based routing network. 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 $N = |\mathcal{E}_{\text{s}}| + |\mathcal{E}_{\text{u}}|$. 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:

$$\begin{aligned}\mathbf{y} = W_0\mathbf{x} &+ \sum_{k \in \mathcal{E}_{\text{s}}} \sigma_k(\mathbf{x}W_g^\top)\Delta W_k\mathbf{x} \\ &+ \sum_{k \in \mathcal{E}_{\text{u}}} \sigma_k(\mathbf{x}W_g^\top)\Delta W_k\mathbf{x},\end{aligned} \quad (14)$$

where, $\sigma(\cdot)$ denotes the softmax operator. If a token shares features with multiple domains, W_g automatically activates multiple experts without requiring discrete task IDs.

Table 1: Continual learning performance across a six-task sequence using LLaMA-2 7B, and Qwen3-4B. Performance is reported as the Overall Performance (OP), Backward Transfer (BWT), and Average Forgetting (F_T) after sequential training. Higher OP (↑) and BWT (↑) indicate better final task accuracy and less negative transfer; lower F_T (↓) indicates less catastrophic forgetting of previously learned tasks. **Bold** values indicate the best performance in each column, underlined values denote the second-best.

	LLaMA-2 7B: Continual Learning on TRACE Benchmarks						**Metrics**		
Method	C-STANCE	FOMC	MeetingBank	ScienceQA	NumGLUE-cm	20Minuten	OP ↑	BWT ↑	F_T ↓
Full FT	49.60	62.10	58.03	61.10	35.58	50.28	52.78	–	–
Seq-LoRA	0.2	1.2	8.7	15.8	14.8	41.4	13.68	-31.24	31.24
EWC	19.0	24.8	15.2	42.4	14.8	41.50	26.28	-22.96	22.96
GEM	5.2	7.1	12.0	24.0	17.3	40.9	17.75	-26.10	26.10
I-LoRA	33.6	8.1	14.4	44.2	22.2	41.4	27.32	-18.30	**18.30**
SETA (ours)	27.31	26.41	20.81	41.63	22.55	33.62	**28.72**	**-16.33**	19.10
	Qwen3-4B: Continual Learning on TRACE Benchmarks						**Metrics**		
Method	C-STANCE	FOMC	MeetingBank	ScienceQA	NumGLUE-cm	20Minuten	OP ↑	BWT ↑	F_T ↓
SeqLoRA	0.2	21.5	5.2	33.3	51.9	38.9	25.17	-35.54	35.54
EWC	0.2	0.6	4.2	6.4	46.9	38.6	16.15	-46.54	46.54
GEM	0.5	13.5	6.5	67.7	29.6	38.6	26.07	-32.70	32.70
I-LoRA	27.2	38.5	16.0	71.0	37.0	39.2	38.15	-15.54	15.54
SETA (ours)	63.64	33.0	21.42	80.77	25.97	34.98	**43.30**	**-12.67**	**15.42**

4 Experiments

4.1 Evaluation Benchmarks

To rigorously evaluate LLM plasticity and retention, we use two benchmark categories spanning domain specificity, format diversity, and reasoning complexity. The Continual Learning Benchmarks include ScienceQA for education, FOMC for finance, MeetingBank for political discourse, C-STANCE and 20Minuten for multilingual adaptation, and NumGLUE for mathematical reasoning. The expert learning rate is $1e$-4 with a 0.2 linear warmup. We sample 5,000 training and 500 evaluation instances per dataset.

4.2 Evaluation Metrics

Let $M_{i,j}$ denote the inference accuracy on task j after training on task i. We evaluate using three complementary metrics. Overall Performance (OP) measures the final model's average accuracy across all tasks:

$$\text{OP} = \frac{1}{T}\sum_{j=1}^{T} M_{T,j}. \tag{15}$$

Average Forgetting (F_T) quantifies the decline from each task's peak accuracy:

$$F_T = \frac{1}{T-1}\sum_{j=1}^{T-1} \max_{l \le T}(M_{l,j} - M_{T,j}). \tag{16}$$

Backward Transfer (BWT) measures the influence of new tasks on previously learned ones:

$$\text{BWT} = \frac{1}{T-1}\sum_{j=1}^{T-1}(M_{T,j} - M_{j,j}). \tag{17}$$

Higher OP ↑ and BWT ↑, and lower F_T ↓ indicate better continual learning performance.

4.3 Comparative Analysis of CL

Evaluation of Domain Adaptation: We evaluated LLaMA-2 7B and Qwen3-4B models using average inference accuracy Acc_t as the primary metric. Table 1 confirms that SETA establishes a new state of the art, outperforming strong baselines such as I-LoRA on most tasks. On LLaMA-2 7B, SETA demonstrates stronger initial task acquisition and substantially greater resistance to catastrophic collapse: while I-LoRA degrades to 8.1% on FOMC by the final step, SETA retains 26.41%, a gap of over 18 percentage points. This robustness extends to MeetingBank, where SETA preserves 20.81% versus 14.4% for I-LoRA, confirming that sparse gradient-driven adaptation captures specialized knowledge more reliably than replay-based methods.

Overall Performance Score and Forgetting: Table 1 compares SETA with established baselines in the overall performance score OP and forgetting F_T. SETA achieves an OP of 28.72% on LLaMA-2 7B, exceeding I-LoRA at 27.32%, while its F_T

of 19.10% remains competitive against I-LoRA's 18.30%. On Qwen3-4B, SETA achieves the best OP of 43.30% and the lowest F_T of 15.42%, outperforming I-LoRA on both metrics. Taken together, these results confirm that the sparse update mechanism balances stability and plasticity without compromising final utility. An ablation isolating the effect of $\mathcal{L}_{\text{gate}}$ is provided in Appendix B.

Analysis of Stability and Plasticity: Table 2 tracks sequential performance on Qwen3-4B, where diagonal entries reflect initial plasticity and off-diagonal decay reflects forgetting. SETA exhibits a stable retention profile: Task 1 accuracy rises to 63.64% by the final step, while EWC and I-LoRA collapse substantially, confirming that the Split-on-Share mechanism actively preserves early task knowledge. On plasticity, SETA achieves 75.90% on Task 4 and sustains high retention on earlier tasks throughout the sequence. Table 3 confirms that this stability–plasticity balance generalises to LLaMA-2 7B.

4.4 Layer-wise Selection Across Architectures

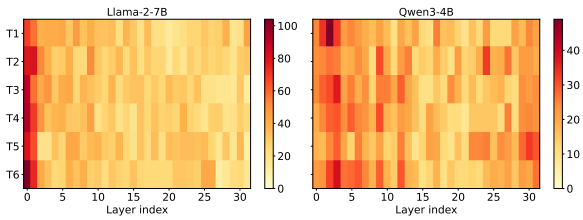

Figure 2: Unique block count per transformer layer and task for LLaMA-2 7B and Qwen3-4B. Darker cells indicate more unique blocks at that layer. Both architectures show non-uniform, task-sensitive selection patterns without architecture-specific tuning.

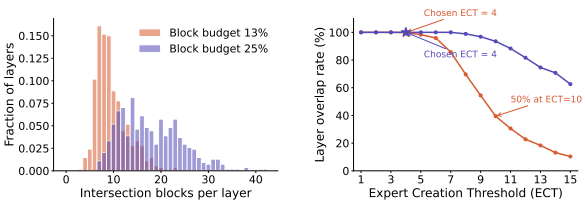

Figure 3: Effect of block budget fraction on expert creation for Qwen3-4B. Left: intersection block count distributions for 13% and 25% budgets. Right: layer overlap rate as a function of ECT; stars mark the chosen ECT and annotations mark the 50% approval knee.

Figure 3 reveals that increasing the block budget from 13% to 25% shifts the intersection distribution rightward, producing larger per-layer overlaps between tasks. As a consequence, the same ECT becomes more permissive under a larger budget: the 13% budget curve reaches 50% approval at ECT$=10$, while the 25% budget curve remains above 60% even at ECT$=15$. This confirms that ECT must be calibrated relative to the block budget: a fixed absolute threshold approves far more splits under a larger budget, potentially creating spurious shared experts from incidental overlap. The chosen ECT values of 4 for the 13% budget and 8 for the 25% budget each represent a principled operating point at the elbow of the approval curve, rejecting noise while preserving genuine knowledge sharing between tasks.

4.5 Unique Sub-space Discovery Rate

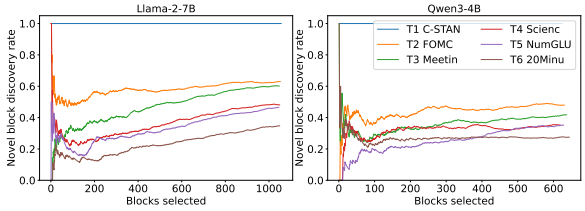

Figure 4: Novel block discovery rate during FFT selection for LLaMA-2 7B and Qwen3-4B. T1 maintains a rate of 1.0 throughout; later tasks decay progressively faster as prior coverage accumulates.

Figure 4 shows that as the task sequence grows, each new task finds progressively less novel ground to cover. The monotonic degradation of later-task discovery rates is consistent across both LLaMA-2 7B and Qwen3-4B, confirming sub-linear capacity expansion as a model-agnostic property of the SETA selection mechanism.

4.5.1 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. However, this extreme parameter sparsity introduces inherent network saturation limits. As the model adapts to sequential tasks, the strictly bounded capacity forces a heavy reliance on the frozen shared core, eventually creating a plasticity bottleneck that constrains forward transfer for future tasks.

Table 2: Task-wise inference accuracy of continual learning methods on Qwen3-4B. Each row corresponds to the model state after training on that task, evaluated on all tasks seen so far. Tasks are ordered sequentially: T1=C-STANCE, T2=FOMC, T3=MeetingBank, T4=ScienceQA, T5=NumGLUE-cm, T6=20Minuten. Diagonal entries reflect plasticity; off-diagonal decay reflects forgetting.

EWC						
Train \ Test	T1	T2	T3	T4	T5	T6
T1	59.8					
T2	50.4	67.3				
T3	0.2	42.3	19.2			
T4	5.2	36.1	22.0	84.2		
T5	5.8	52.2	17.0	75.2	60.5	
T6	0.2	0.6	4.2	6.4	46.9	38.6

I-LoRA						
Train \ Test	T1	T2	T3	T4	T5	T6
T1	59.0					
T2	50.6	66.3				
T3	15.8	59.5	19.0			
T4	33.0	45.8	16.4	82.4		
T5	19.8	45.4	18.4	80.6	40.7	
T6	27.2	38.5	16.0	71.0	37.0	39.2

SETA (ours)						
Train \ Test	T1	T2	T3	T4	T5	T6
T1	57.01					
T2	57.99	67.54				
T3	52.12	56.45	46.69			
T4	37.29	26.00	24.16	75.90		
T5	59.00	38.91	22.83	83.01	41.03	
T6	63.64	33.00	21.42	80.77	25.97	34.98

Table 3: Task-wise inference accuracy of continual learning methods on LLaMA-2 7B. Each row corresponds to the model state after training on that task, evaluated on all tasks seen so far. Diagonal entries reflect plasticity on the current task; off-diagonal decay reflects forgetting of prior tasks.

PP						
Train \ Test	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						
Train \ Test	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 (ours)						
Train \ Test	T1	T2	T3	T4	T5	T6
T1	47.69					
T2	43.80	65.93				
T3	33.60	49.39	43.83			
T4	31.60	31.45	24.68	34.34		
T5	32.60	49.19	23.91	48.18	28.57	
T6	27.31	26.41	20.81	41.63	22.55	33.62

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	6.53	7.16	8.10
TP (%) ↓	0.98	0.98	1.03	1.01	1.11	1.25

5 Related work: PEFT in CL

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). 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., 2022). Extensions such as ReLoRA (Lialin et al., 2024) 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) explore adaptive-rank strategies without theoretical proofs. LoRA has also been combined with Mixture-of-Experts architectures (Li et al., 2024a; Huang et al., 2024), 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) (He et al., 2025), which selects task-relevant sub-matrices.

6 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 achieves competitive or superior performance relative to state-of-the-art continual learning baselines across diverse reasoning benchmarks.

Limitations

While SETA shows strong continual learning performance, several limitations remain. Our experiments cover models up to 14B parameters, so how expert specialization and routing scale to larger models is still unknown. SETA has also been validated only on standard attention-based transformers; its behavior on alternative architectures remains an open question. Our evaluation follows a fixed task ordering across six benchmarks. Longer sequences, greater task diversity, and different orderings may reveal new behaviors or failure modes. Finally, as tasks grow, the shared expert pool expands without any pruning or consolidation mechanism, which may limit scalability in long-horizon settings. We leave these directions as important avenues for future work.

Data and Model Usage

We conduct all experiments using publicly available large language models and benchmark datasets. Our primary backbone models are LLaMA-2 7B, LLaMA-3 8B, and Qwen3-4B, which are licensed for 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. 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

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

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. 2020. Rigging the lottery: Making all tickets winners. In *International conference on machine learning*, pages 2943–2952. PMLR.

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. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

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.

Jiayi Han, Liang Du, Hongwei Du, Xiangguo Zhou, Yiwen Wu, Yuanfang Zhang, Weibo Zheng, and Donghong Han. 2025. Slim: Let llm learn more and forget less with soft lora and identity mixture. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4792–4804.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.

Haoze He, Juncheng Li, Xuan Jiang, and Heather Miller. 2025. Smt: Fine-tuning large language models with sparse matrices. In *International Conference on Learning Representations*, volume 2025, pages 32369–32393.

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, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Yebowen Hu, Timothy Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. Meetingbank: A benchmark dataset for meeting summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16409–16423.

Shaohan Huang, Furu Wei, and 1 others. 2024. Mixture of lora experts. In *International Conference on Learning Representations*, volume 2024, pages 47302–47318.

Tannon Kew, Marek Kostrzewa, and Sarah Ebling. 2023. 20 minuten: A multi-task news summarisation dataset for german. In *Proceedings of the 8th edition of the Swiss Text Analytics Conference*, pages 1–13.

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, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 3045–3059.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. 2021. Base layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*, pages 6265–6274. PMLR.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and 1 others. 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 *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Xinlong Li, Weijieying Ren, Wei Qin, Lei Wang, Tianxiang Zhao, and Richang Hong. 2025a. Analyzing

and reducing catastrophic forgetting in parameter efficient tuning. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Xinlong Li, Weijieying Ren, Wei Qin, Lei Wang, Tianxiang Zhao, and Richang Hong. 2025b. Analyzing and reducing catastrophic forgetting in parameter efficient tuning. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

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

Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. 2024. Relora: High-rank training through low-rank updates. In *International Conference on Learning Representations*, volume 2024, pages 49405–49421.

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*.

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

Pan Lu, Swaroop Mishra, Tanglin 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. *Advances in neural information processing systems*, 35:2507–2521.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*.

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.

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. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523.

J Pablo Muñoz, Jinjie Yuan, and Nilesh Jain. 2025. Low-rank adapters meet neural architecture search for llm compression. *arXiv preprint arXiv:2501.16372*.

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, 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.

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*.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

Stephen Roller, Sainbayar Sukhbaatar, Jason Weston, and 1 others. 2021. Hash layers for large sparse models. *advances in neural information processing systems*, 34:17555–17566.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

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, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

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*.

Yifan Yang, Kai Zhen, Ershad Banijamali, Athanasios Mouchtaris, and Zheng Zhang. 2024. Adazeta: Adaptive zeroth-order tensor-train adaption for memory-efficient large language models fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 977–995.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9.

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.

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, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, and 1 others. 2022. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114.

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), and architectural expansion (Rusu et al., 2016), 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 (Li et al., 2025b) mitigates forgetting by maintaining dual LoRA adapters, a fast (short-term) and a slow (long-term) memory, interpolated to balance plasticity and stability. SLiM (Han et al., 2025) 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 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., 2022). 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.

B Ablation: Routing-Aware Gating Regularisation

Table 5 report the sequential task-accuracy matrices for Qwen3-4B under identical settings except for the routing-aware gating loss $\mathcal{L}_{\text{gate}}$. T1 and T2 rows are identical across both configurations because shared experts only exist from T3 onward; the gating loss therefore has no effect until the first Split-on-Share event. Enabling $\mathcal{L}_{\text{gate}}$ improves average accuracy at T6 from 38.48 to 43.30 and reduces forgetting as measured by BWT from -15.70 to -12.67. The gains are concentrated in two areas.

Retention of early tasks: C-STANCE accuracy at the final step rises from 53.64 to 63.64, a gain of 10.0 points, while NumGLUE-cm retention at T6 improves from 15.28 to 25.97. These are precisely the tasks whose shared experts accumulate the highest ω_i weights and therefore receive the strongest gate-steering signal, confirming that $\mathcal{L}_{\text{gate}}$ acts where it is most needed.

Plasticity on intermediate tasks: MeetingBank diagonal accuracy improves from 38.04 to 46.69 (+8.65 points) and NumGLUE-cm from 34.62 to 41.03 (+6.41 points). Because the gating loss reduces routing probability toward saturated shared experts, the gating network more readily directs new-task inputs toward available unique experts, preserving plasticity. The modest diagonal drop on ScienceQA is consistent with the same mechanism: stricter routing reduces the task's access to broadly shared representations, a known stability–plasticity trade-off that the overall BWT improvement confirms is net beneficial.

Together, these results show that routing-aware gating and weight regularisation are complementary: $\mathcal{L}_{\text{reg}}$ corrects drift after gradient updates have occurred, while $\mathcal{L}_{\text{gate}}$ prevents the gating network from routing new-task inputs into at-risk experts in the first place, providing a two-layer defence at negligible additional cost since both losses share the same $\lambda_i(t)$ coefficients and are evaluated in a single forward pass.

C Empirical Analysis and Design Choices

This section presents the structural analysis of SETA's gradient-guided block selection across two architectures, LLaMa-2 7B, LLaMA-3 8B and Qwen3-4B, using the TRACE benchmark tasks.

Table 5: Qwen3-4B sequential accuracy matrix under dynamic λ, with routing-aware gating disabled and enabled. Diagonal entries (bold) measure plasticity; off-diagonal entries measure retention.

	Test Task							
Train	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**OP ↑**	**BWT↑**
Routing-Aware Gating: Disabled								
T1	57.01							
T2	57.99	67.54						
T3	50.61	56.25	38.04					
T4	37.92	28.83	23.56	79.29				
T5	49.79	28.83	20.95	73.44	34.62			
T6	53.64	32.06	19.04	78.00	15.28	32.84	38.48	−15.70
Routing-Aware Gating: Enabled (SETA - ours)								
T1	57.01							
T2	57.99	67.54						
T3	52.12	56.45	46.69					
T4	37.29	26.00	24.16	75.90				
T5	59.00	38.91	22.83	83.01	41.03			
T6	63.64	33.00	21.42	80.77	25.97	34.98	**43.30**	**−12.67**

C.1 Attention Sub-block Scaling Analysis

Table 6 presents uniqueness statistics for Qwen3-4B under two block budget configurations. Under the 13% budget, the shared pool grows from 0 to 922 blocks across six tasks, with T1 retaining only 22% of its original unique blocks by T6. Under the 25% budget, shared blocks reach 26% of total by T6, with T1 retaining 16%. The larger budget thus drives proportionally stronger sharing despite the stricter ECT, confirming that ECT must be calibrated relative to the block budget rather than set as a fixed constant.

C.2 Block Selection Budget and Efficiency

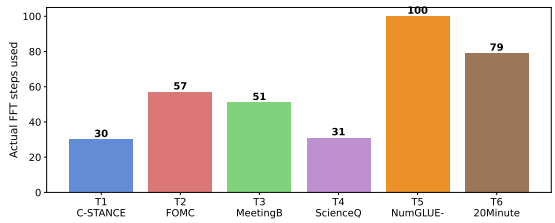

Figure 5: Actual FFT steps consumed per task under the dynamic stopping criterion for LLaMA-3 8B. Classification tasks such as C-STANCE and ScienceQA stabilise in under 35 steps; generation tasks such as NumGLUE-cm and 20Minuten require up to 100 steps, reflecting their more complex gradient landscapes.

The dynamic stopping criterion adapts the FFT budget to each task's gradient complexity without any manual tuning. Tasks with concentrated gradient signal converge quickly, while open-ended generation tasks require longer stabilisation. This task-sensitivity directly validates the adaptive design of Step A.

C.3 Cross-Task Block Overlap

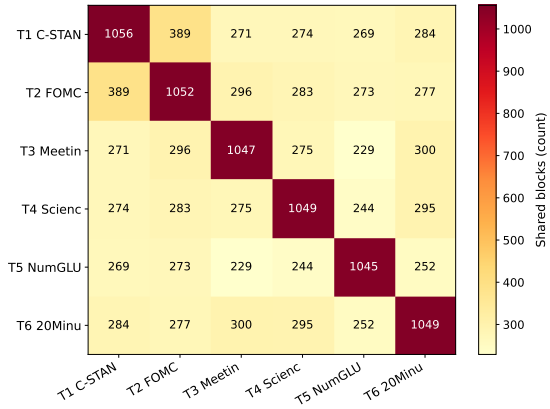

Figure 6: Cross-task block overlap matrix for LLaMA-2 7B. Diagonal entries report each task's total block budget, ranging from approximately 1,045 to 1,056; off-diagonal entries report the number of blocks shared between task pairs. Off-diagonal values span 229–389 blocks, with the strongest overlap between C-STANCE and FOMC (389 blocks) and the weakest between MeetingBank and NumGLUE-cm (229 blocks). This heterogeneous but consistently partial overlap provides the raw material for shared experts in the SoS mechanism.

No task pair shares more than 37% of its blocks, confirming that each task retains a substantially distinct gradient footprint. The partial overlap is precisely what the SoS split exploits: blocks appearing in multiple task selections are promoted to shared experts, while the remainder become task-unique experts. The variation across pairs (229

Table 6: Evolution of sub-block uniqueness statistics for Qwen3-4B under two block budget fractions and their respective Expert Creation Thresholds (ECT). Total Blocks = cumulative raw blocks selected by FFT. Total Shared = blocks overlapping across tasks passing ECT.

Seq	Task	Total Blocks	Total Shared	T1	T2	T3	T4	T5	T6
Block budget 13% (ECT = 4)									
1	C-STANCE	720	0	720	0	0	0	0	0
2	FOMC	1449	388	332	341	0	0	0	0
3	MeetingBank	2180	530	251	280	315	0	0	0
4	ScienceQA	2908	672	207	239	258	250	0	0
5	NumGLUE-cm	3640	796	180	210	223	217	234	0
6	20Minuten	4367	922	159	185	194	182	218	195
Block budget 25% (ECT = 8)									
1	C-STANCE	1440	0	1440	0	0	0	0	0
2	FOMC	2862	744	696	678	0	0	0	0
3	MeetingBank	4282	1140	489	489	598	0	0	0
4	ScienceQA	5702	1527	364	378	447	431	0	0
5	NumGLUE-cm	7124	1906	281	277	352	331	320	0
6	20Minuten	8545	2225	225	224	269	258	266	256

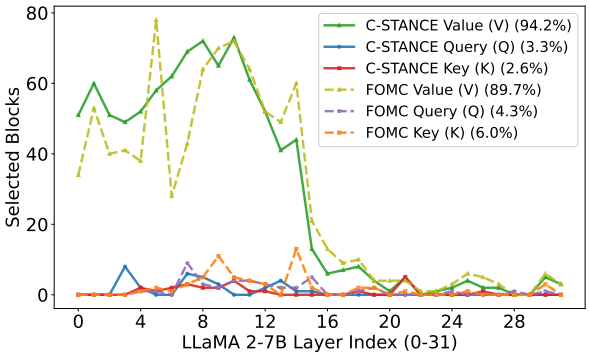

Figure 7: Layer-wise analysis shows ∼95% of top-ranked gradients concentrate in the Value (V) projection, indicating that attention routing (Q, K) remains stable while content (V) requires adaptation.

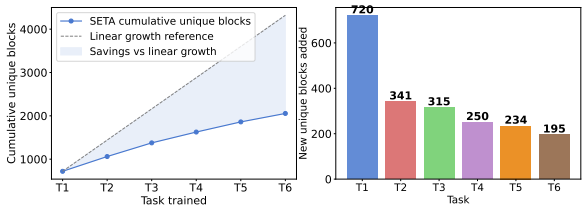

Figure 8: Cumulative unique blocks added across the task sequence for Qwen3-4B under the 13% block budget. *Left:* cumulative count against a linear growth reference (dashed); the shaded region quantifies capacity saved relative to a system that allocates fresh blocks per task. *Right:* new unique blocks contributed by each task. Qwen3-4B adds 720 blocks for T1 but only 195 by T6, a 73% reduction in per-task demand as more blocks are reused from earlier tasks.

to 389 shared blocks) further reflects the differing linguistic proximity of the TRACE tasks stance detection and monetary-policy classification share more representational structure than summarisation and arithmetic reasoning.

C.4 Sub-linear Parameter Growth

Sub-linear growth is SETA's central efficiency claim. Each successive task reuses a growing fraction of previously selected blocks, so the total parameter footprint grows far slower than the number of tasks. By T6, Qwen3-4B's cumulative unique block count of 2,055 is only 48% of what a non-reusing baseline would require at 4,320, a saving of over 2,265 blocks. The per-task contribution drops monotonically from 720 at T1 to 195 at T6, confirming that later tasks operate predominantly in the reuse regime.

C.5 Novel Block Discovery per Layer

The progressive dimming of later task rows confirms that earlier tasks have already claimed the most informative layer-block positions, and later tasks operate increasingly in the reuse regime. This is the layer-level evidence underlying the cumulative growth result in Figure 8.

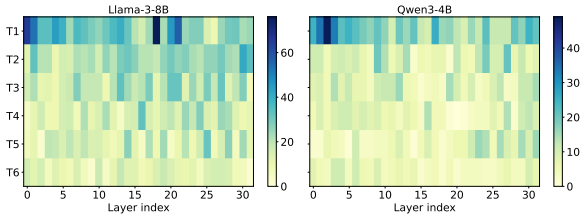

Figure 9: Unique blocks per transformer layer per task. A block is unique if no prior task selected it at the same layer position. Brighter cells indicate layers where a task introduced genuinely new block coverage. Later tasks T4–T6 show sparser brightness than T1–T2, reflecting increasing reuse as the sequence progresses.

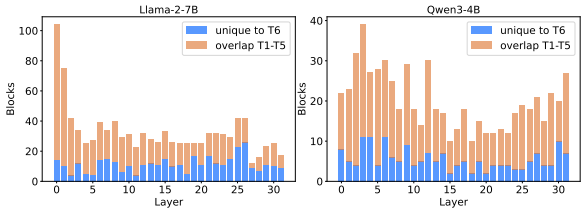

Figure 10: Per-layer block composition for the final task T6 (20Minuten) for LLaMA-2 7B (left, y-scale 0–100 blocks) and Qwen3-4B (right, y-scale 0–40 blocks). Blue bars show blocks unique to T6; orange bars show blocks shared with at least one prior task T1–T5. Both architectures show mixed composition across all layers, with no layer being entirely novel or entirely reused.

C.6 Final Task Block Composition

Even at T6, after five prior tasks have collectively covered the gradient landscape, the model still identifies genuine novel blocks in most layers. This prevents expert saturation and ensures T6 receives a dedicated unique expert. The layer-wise mixture of novel and reused blocks is the direct trigger for the SoS split decision at T6.

C.7 Expert Weight Analysis

Figures 12 and 13 examine the learned weight magnitudes of unique and shared experts to verify that the SoS architecture produces the intended structural separation between task-specific and cross-task knowledge.

Figure 12 reveals a clear structural divergence between expert types. Unique experts attain substantially higher L2 norms at a mean of approximately 30.8, compared to shared experts at a mean of approximately 6.4, a gap of approximately $5\times$. This confirms that unique experts undergo strong task-specific gradient updates, while shared experts remain close to the pre-trained base, acting as a regularised cross-task knowledge store. The pat-

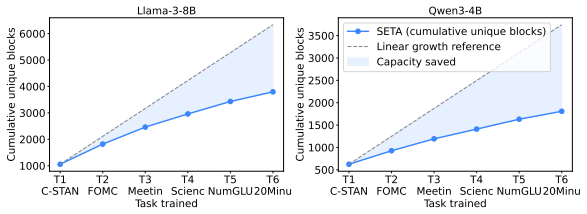

Figure 11: Cumulative unique block count across the task sequence for LLaMA-3 8B (left) and Qwen3-4B (right) against a linear growth reference (dashed). Both architectures exhibit consistent sub-linear growth, confirming model-agnostic efficiency through block reuse. Qwen3-4B converges toward the reuse regime slightly faster, consistent with its smaller attention dimensionality.

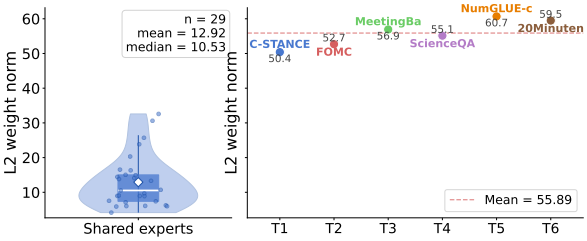

Figure 12: L2 weight norm of unique and shared experts in the final T6 SETA checkpoint for Qwen3-4B. Left: distribution of shared expert norms across $n = 36$ experts; violin shows kernel density, inner box depicts the IQR with median as a white line, the diamond marks the mean, and jittered dots show individual expert norms. Right: each dot represents one unique expert trained for tasks T1–T6, coloured by task; the dashed line indicates the group mean.

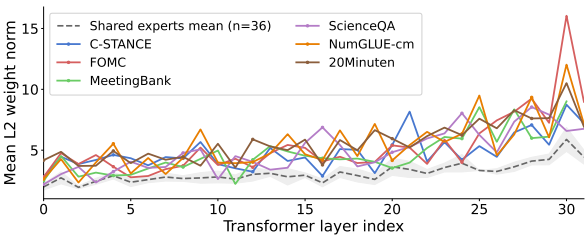

Figure 13: Mean L2 weight norm per transformer layer for each unique expert shown as coloured lines, one per task T1–T6, against the mean ± 1 standard deviation of all shared experts shown as a dashed grey line with shaded band, for LLaMA-3 8B.

tern is consistent with the SoS design: orthogonal subspace isolation restricts gradient flow into prior unique experts, concentrating plasticity in the current task subspace, while elastic anchoring actively suppresses drift in shared experts.

C.8 Expert Creation Threshold

We analyse how the Expert Creation Threshold (ECT) and the block budget fraction jointly govern the SoS split behaviour across architectures.

Figure 14 shows that as the threshold increases, shared expert creation decreases monotonically across all architectures. For LLaMA-2 7B and LLaMA-3 8B, the curves decline steeply around the chosen value, sitting at the elbow where small increases yield large reductions in approved splits. For Qwen3-4B, the decline occurs earlier due to its smaller per-layer block space, making the chosen threshold the natural operating point. Across all architectures, the absorbed remainder curve remains near zero, indicating that tiny remainder fragments are rare in practice and the Tiny Remainder Threshold has minimal absorptive effect.

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.

D.1 Baselines

We compare SETA against PEFT-based baselines to ensure a fair evaluation. The methods include (1) Sequence Fine-tuning (Seq-LoRA), which adapts continuously without regularization; (2) Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017), which restricts critical parameter changes via Fisher information. We also evaluate gradient-based constraints (3) GEM (Lopez-Paz and Ranzato, 2017), (4) Progressive Prompt (PP) (Razdaibiedina et al., 2023) and (5) I-LoRA (Li et al., 2025b) included as a direct architectural comparator.

Traditional CL Strategies:

- Sequence Fine-tuning (Seq-LoRA): Represents naive adaptation, continuously tuning parameters on the task sequence without regularization or replay.
- 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) (Lopez-Paz and Ranzato, 2017): Maintains a gradient subspace of old tasks, projecting updates orthogonally to avoid interference.

Architecture and Prompt-Based Methods:

- I-LoRA (Li et al., 2025b): A relevant comparative baseline that utilizes dual-memory LoRA adapters for sequential learning.
- Progressive Prompt (PP) (Razdaibiedina et al., 2023): Sequentially learns task-specific soft prompts and concatenates them, isolating task knowledge in distinct prefix tokens.

D.2 Prompt Design for Training and Inference

To clarify, all models are utilized in a strictly generative text-to-text paradigm via causal language modeling. We do not append any task-specific classification heads to the architecture. Both training and inference are conducted using a standardized instruction-tuning prompt format. Every data instance across all tasks is uniformly structured with a system prefix, the task instruction including context, and a response trigger:

```
Below is an instruction that describes
a task.
Write a response that appropriately
completes the request.

### Instruction:
{task_specific_instruction_and_input}

### Response:
{target_label_or_text}
```

During evaluation, the model is provided the prompt up to the response trigger, and the output is generated autoregressively. Because we do not use a classification head, we map the raw generated text back to the discrete labels using dataset-specific parsing scripts. For classification tasks, the parsing functions extract the predicted option or numeric label from the generated string and compare it against the target ground truth to compute Accuracy or macro-F1 metrics. For generation tasks

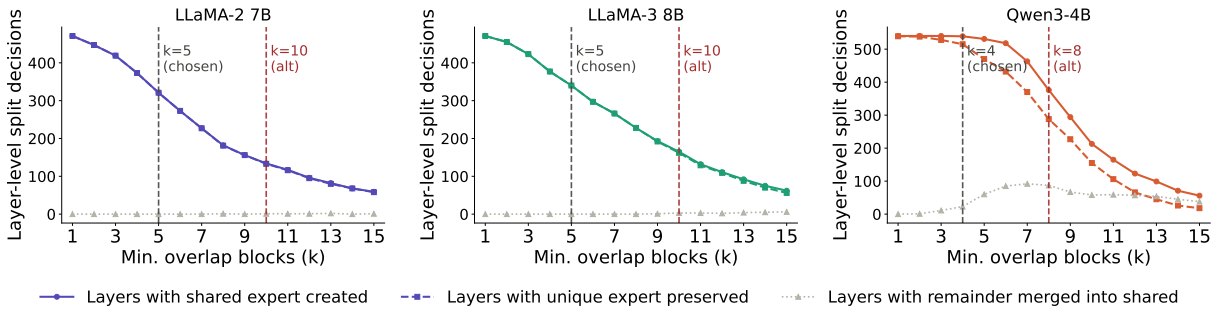

Figure 14: Layer-level split decisions as a function of ECT for LLaMA-2 7B, LLaMA-3 8B, and Qwen3-4B. Each curve counts layer decisions across all 15 task pairs: shared expert created, unique expert preserved, and remainder merged into shared. The black dashed line marks the chosen ECT; the red dashed line marks the alternative value for comparison.

like MeetingBank, the raw generated text is directly evaluated against the reference summary using ROUGE metrics.

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 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.

F Computational Overhead

To formalize the computational analysis, let B, S, and d denote the batch size, sequence length, and hidden dimension, respectively. For the adaptation and routing components, let r represent the LoRA rank, E the total number of experts, K the active experts per token, and L_{SETA} the number of routed layers. The architectural design of SETA leverages this dynamic MoE routing to achieve superior expressive power with high computational efficiency. While i-LoRA is restricted to a static, monolithic adapter per task with a fixed $\mathcal{O}(rd)$ overhead, SETA utilizes conditional computation to activate only a sparse subset ($K \ll E$) of task-specific experts. This routing mechanism requires a minimal gating overhead of $\mathcal{O}(dE)$, yet it enables the model to dynamically specialize at the token level with an execution cost of only $\mathcal{O}(Krd)$. By decoupling the total parameter capacity from the per-token compute, SETA provides an adaptive framework that scales far more effectively than i-LoRA's rigid structure, allowing for high-fidelity task navigation through efficient, on-demand parameter fetching.

G SETA Algorithm

Algorithm 1 details the SETA training procedure. For each task t, Line 3 identifies the active sparse subspace $\mathcal{P}_t$ via an adaptive gradient-based pipeline. For $t > 1$, the Split-on-Share mechanism in Lines 7–11 decomposes this subspace layer-wise: overlapping parameters merge into the shared expert E_{s}, disjoint parameters form the frozen unique expert $E_{\text{u}}^{(t-1)}$, and novel indices initialize $E_{\text{u}}^{(t)}$. Prior experts are frozen in Line 13, the gating network expanded in Line 14, and the model optimized in Line 15 over $\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{gate}}$. Line 17 snapshots shared expert weights before the next task begins.

H Detailed Per-Task Results Across Models and Baselines

Table 7: Task-wise inference accuracy of SeqLoRA on LLaMA-2 7B. Diagonal entries reflect plasticity; off-diagonal decay reflects forgetting.

SeqLoRA	**C-STANCE**	**FOMC**	**MeetingBank**	**ScienceQA**	**NumGLUE-cm**	**20Minuten**
C-STANCE	41.8					
FOMC	21.8	60.3				
MeetingBank	19.4	49.4	24.7			
ScienceQA	32.0	26.2	14.8	36.8		
NumGLUE-cm	25.6	20.4	5.7	24.4	33.3	
20Minuten	0.2	1.2	8.7	15.8	14.8	41.4

Table 8: Task-wise inference accuracy of EWC on LLaMA-2 7B. Diagonal entries reflect plasticity; off-diagonal decay reflects forgetting.

EWC	**C-STANCE**	**FOMC**	**MeetingBank**	**ScienceQA**	**NumGLUE-cm**	**20Minuten**
C-STANCE	42.2					
FOMC	35.6	70.6				
MeetingBank	33.8	48.4	25.6			
ScienceQA	37.4	38.1	14.3	64.2		
NumGLUE-cm	30.8	25.0	6.8	36.8	28.4	
20Minuten	19.0	24.8	15.2	42.4	14.8	41.5

Table 9: Task-wise inference accuracy of GEM on LLaMA-2 7B. Diagonal entries reflect plasticity; off-diagonal decay reflects forgetting.

GEM	**C-STANCE**	**FOMC**	**MeetingBank**	**ScienceQA**	**NumGLUE-cm**	**20Minuten**
C-STANCE	38.8					
FOMC	31.4	61.5				
MeetingBank	17.4	44.4	23.3			
ScienceQA	31.8	26.0	11.5	40.4		
NumGLUE-cm	5.0	0.6	7.4	18.6	32.1	
20Minuten	5.2	7.1	12.0	24.0	17.3	40.9

Table 10: Task-wise inference accuracy of I-LoRA on Qwen3-4B. Diagonal entries reflect plasticity; off-diagonal decay reflects forgetting.

	C-STANCE	FOMC	MeetingBank	ScienceQA	NumGLUE-cm	20Minuten
C-STANCE	59.0					
FOMC	50.6	66.3				
MeetingBank	15.8	59.5	19.0			
ScienceQA	33.0	45.8	16.4	82.4		
NumGLUE-cm	19.8	45.4	18.4	80.6	40.7	
20Minuten	27.2	38.5	16.0	71.0	37.0	39.2

Table 11: Task-wise inference accuracy of EWC on Qwen3-4B. Diagonal entries reflect plasticity; off-diagonal decay reflects forgetting.

	C-STANCE	FOMC	MeetingBank	ScienceQA	NumGLUE-cm	20Minuten
C-STANCE	59.8					
FOMC	50.4	67.3				
MeetingBank	0.2	42.3	19.2			
ScienceQA	5.2	36.1	22.0	84.2		
NumGLUE-cm	5.8	52.2	17.0	75.2	60.5	
20Minuten	0.2	0.6	4.2	6.4	46.9	38.6

Table 12: Task-wise inference accuracy of SeqLoRA on Qwen3-4B. Diagonal entries reflect plasticity; off-diagonal decay reflects forgetting.

	C-STANCE	FOMC	MeetingBank	ScienceQA	NumGLUE-cm	20Minuten
C-STANCE	60.0					
FOMC	49.7	67.5				
MeetingBank	0.0	46.0	19.8			
ScienceQA	0.0	27.3	22.9	85.5		
NumGLUE-cm	0.0	19.8	12.1	75.2	56.8	
20Minuten	0.2	21.5	5.2	33.3	51.9	38.9

Table 13: Task-wise inference accuracy of GEM on Qwen3-4B. Diagonal entries reflect plasticity; off-diagonal decay reflects forgetting.

	C-STANCE	FOMC	MeetingBank	ScienceQA	NumGLUE-cm	20Minuten
C-STANCE	60.3					
FOMC	48.7	68.5				
MeetingBank	0.0	47.2	18.8			
ScienceQA	8.0	42.0	19.6	83.8		
NumGLUE-cm	2.8	51.2	11.9	80.5	49.4	
20Minuten	0.5	13.5	6.5	67.7	29.6	38.6

Algorithm 1 SETA: Sparse Subspace-to-Expert Training Procedure

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

Initialize: $\mathcal{E} \leftarrow \emptyset$, $W_g \leftarrow W_0$
for $t = 1$ to k **do**
  Subspace Selection: Identify active sparse blocks $\mathcal{P}_t$ via adaptive gradient pipeline.
  if $t = 1$ **then**
    Construct initial unique expert $E_{\text{u}}^{(1)}$ on $\mathcal{P}_1$.
  else
    for each layer $l \in L$ **do**
      SoS Decomposition: Compute $\mathcal{I}_l$ and $\mathcal{R}_l$ via layer-wise topological filters (Eq. 6, 7).
      Expert Allocation: $E_{\text{s}} \leftarrow \mathcal{I}_l$ via weight inheritance; freeze $E_{\text{u}}^{(t-1)} \leftarrow \mathcal{R}_l$.
      New Expert: Initialize $E_{\text{u}}^{(t)}$ on $\mathcal{P}_t \setminus (\mathcal{I}_l \cup \mathcal{R}_l)$.
    end for
  end if
  Freeze History: $\nabla E_{\text{u}}^{(k)} = 0$ for all $k < t$.
  Gating Expansion: Resize W_g to $|\mathcal{E}|$; preserve logit invariance.
  Optimize: Minimize $\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{gate}}$ over $E_{\text{u}}^{(t)}$ and E_{s} (Eq. 13).
  Anchor Update: $\hat{W}_i \leftarrow W_i$ for all shared experts.
end for
Inference: Task-agnostic retrieval via W_g (Eq. 14).