Amortized Mixture of LoRA Experts with GFlowNets for Continual Learning in LLMs

Anonymous ACL submission

Abstract

Continual learning (CL) in large language models aims to incrementally acquire knowledge from a stream of tasks without forgetting, but existing PEFT-based CL methods either enforce strict parameter isolation across tasks or rely on deterministic sentence-level routing over expert modules. This paper proposes a novel mixture of LoRA experts that casts expert selection as amortized posterior inference via a GFlowNet sampler, which selects expert subsets in proportion to a reward that reflects task fit, thereby inducing structured sharing in related tasks while preserving task-specific capacity. The sampler is trained jointly with the LoRA experts under a trajectory balance objective, allowing the model to recompose previously learned experts when novel composite tasks emerge. Experiments on standard CL benchmarks with T5-large and LLaMA backbones demonstrate that it outperforms the state-of-the-art methods by 4.2% in average accuracy in the task-agnostic inference setting, and substantially reduces backward transfer degradation, indicating effective knowledge retention and compositional generalization across the task sequence.

1 Introduction

Pre-trained large language models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks (Brown et al., 2020; Touvron et al., 2023; Grattafiori et al., 2024). However, practical deployment of LLMs requires continual adaptation to a stream of tasks that arrive sequentially, a setting known as continual learning (CL). In this setting, the central challenge is catastrophic forgetting (McCloskey and Cohen, 1989; French, 1999): updating the model on a new task drives its parameters away from regions that previously supported strong performance on earlier tasks, resulting in severe degradation of prior knowledge.

Existing CL methods for LLMs largely fall into rehearsal-, regularization-, and architecture-based approaches (Biesialska et al., 2020; Wang et al., 2024). Among them, parameter-efficient fine-tuning (PEFT) combined with architectural isolation has become the dominant paradigm, in which a separate adapter or prompt module is allocated and frozen for each task (Razdaibiedina et al., 2023; Wang et al., 2023a; Zhao et al., 2024; Le et al., 2024; Wang et al., 2025). While such methods effectively mitigate forgetting, they exhibit two fundamental limitations from the perspective of knowledge organization in LLMs.

(1) Redundant capacity allocation across related tasks. When two semantically related tasks arrive sequentially (for example, sentiment analysis over reviews from different domains) previous methods enforce strict one-to-one task–expert mapping or rely on deterministic sentence-level routing, allocating disjoint adapter capacity to tasks that could otherwise share a common expert. This redundancy wastes model capacity and prevents the reuse of previously acquired knowledge.

(2) Limited compositional reuse under composite task shifts. When a new task requires the joint use of capabilities acquired across multiple previous tasks (such as a composite task combining topic classification and sentiment analysis) deterministic routing cannot dynamically compose subsets of existing experts. The model is then forced either to allocate a new dedicated expert from scratch or to commit to a single suboptimal expert, both of which undermine forward transfer.

A natural remedy is to allow multiple experts to be activated per input and to share capacity across related tasks. However, naively replacing the router with a top-k softmax gate inherits the same deterministic bias and tends to concentrate gradient updates on a small subset of experts, leaving the remaining experts under-trained. In Figure 1, this single-expert commitment is harmless for tasks that

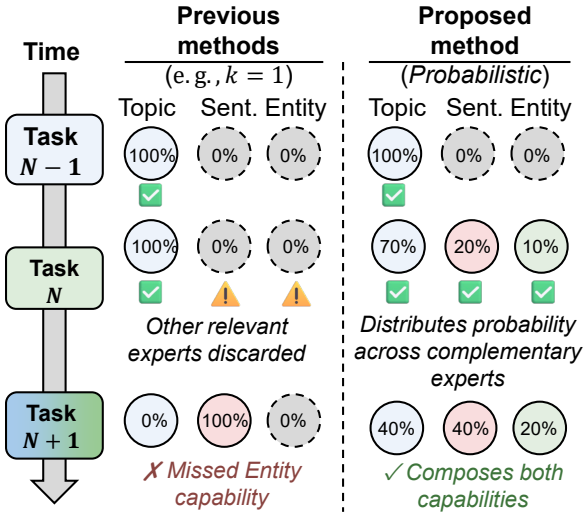

Figure 1: Routing behavior of (left) deterministic top-1 versus (right) the proposed probabilistic sampler in a sequence of tasks. Circles denote experts; solid and dashed lines represent trained and untrained paths, respectively. Top-1 suffices for Task $N-1$ but discards complementary signal on Task N and misses one capability on the composite Task $N+1$. The proposed sampler distributes mass across complementary experts and composes both capabilities.

require one capability, but discards complementary signal when tasks recruit multiple capabilities and outright fails when tasks require their composition. What is needed instead is a routing mechanism that samples diverse expert subsets in proportion to their fit to the input, so that semantically related tasks naturally converge to overlapping expert subsets while composite tasks can recompose previously trained experts.

Motivated by above requirements, this paper proposes a mixture of LoRA experts that casts expert selection as amortized posterior inference via a GFlowNet sampler (Bengio et al., 2021, 2023; Hu et al., 2024). Rather than routing each input to a single expert through a deterministic softmax gate, the proposed method samples expert subsets in proportion to a reward that reflects task fit, thereby inducing structured sharing across related tasks while preserving task-specific capacity. The sampler is trained jointly with the LoRA experts under a trajectory balance objective (Malkin et al., 2022), allowing the model to recompose previously learned experts when novel composite tasks emerge. Inference proceeds without access to task identifiers, aligning the method with the realistic task-agnostic CL setting.

Our main contributions are threefold:

- We identify two structural limitations of existing PEFT-based CL methods for LLMs: redundant capacity allocation across related tasks and limited compositional reuse under composite task shifts. We motivate a probabilistic multi-expert selection method to address them.
- We propose a mixture of LoRA experts that formulates expert selection as amortized posterior inference with a GFlowNet sampler, allowing structured expert sharing across related tasks and compositional recomposition under composite task shifts.
- We conduct extensive experiments on standard CL benchmarks with T5-large and LLaMA backbones, showing that the proposed method outperforms the state-of-the-art methods by 4.2% in average accuracy under the task-agnostic inference setting, while substantially reducing backward transfer degradation.

2 Related Work

2.1 Continual Learning for LLMs

CL for large language models has increasingly relied on PEFT with selection or routing mechanisms over a frozen backbone. O-LoRA (Wang et al., 2023a) constrains LoRA parameters of each new task to be orthogonal to the LoRA subspaces of previously learned tasks; while this prevents interference with past losses, it also enforces strict disjointness, so that semantically related tasks must learn entirely new orthogonal subspaces rather than share parameters. ProgPrompt (Razdaibiedina et al., 2023) and L2P (Wang et al., 2022) similarly isolate tasks at the prompt level through concatenation or retrieval from a fixed pool, preventing fine-grained reuse of previously learned modules.

More recent approaches relax this isolation through soft routing. SAPT (Zhao et al., 2024) aligns prompt learning and selection via shared attention weights, and NoRGa (Le et al., 2024) introduces non-linear residual gates to combine prefix prompts. MoLE-CIE (Wang et al., 2025) routes each token to multiple LoRA experts at the token level and preserves router distributions through gate-reflection-based distillation. However, the selection mechanisms in all of these methods remain deterministic, relying on softmax or attention-based scoring. As a result, a small subset

of experts that receives favorable gradients during early training tends to dominate routing decisions, leaving the remaining experts under-utilized and effectively excluded from further updates. This imbalance becomes particularly costly under composite or compositional task shifts. The proposed method replaces deterministic gating with a probabilistic flow-based sampler that allocates probability mass across multiple high-reward expert subsets, addressing the exploration–exploitation imbalance that previous MoE-based CL methods have not resolved.

2.2 GFlowNets for Discrete Sampling

Generative Flow Networks (GFlowNets) (Bengio et al., 2021, 2023) are amortized samplers defined over directed acyclic graphs, in which a stochastic forward policy is trained to sample terminal states x in proportion to a non-negative reward $R(x)$, that is, $p_F(x) \propto R(x)$. They were originally developed for combinatorial discrete spaces such as molecule and biological sequence design, where the goal is to recover a diverse set of high-reward candidates rather than a single optimum. Unlike standard reinforcement learning, which maximizes expected cumulative reward and tends to collapse onto a narrow set of modes, GFlowNets distribute probability mass across distinct modes by construction.

The trajectory balance (TB) objective (Malkin et al., 2022) improved training stability over the earlier detailed balance formulation by matching the product of forward and backward trajectory probabilities to the partition function, mitigating credit assignment difficulties over long trajectories; reward-conditional extensions (Jain et al., 2023) further generalize the framework to multi-objective settings. The most relevant work Hu et al. (2024) casts intractable posterior inference in LLM as amortized sampling with a GFlowNet, embedding GFlowNet training directly into LLM fine-tuning. In contrast to these applications, which sample token sequences or latent variables, the proposed method applies GFlowNet sampling to a different object: the discrete index of LoRA experts selected at inference time. The forward policy sequentially samples expert indices, and the reduction in language modeling loss conditioned on that selection serves as the reward. To our knowledge, this is the first application of GFlowNet-based amortized inference to architectural routing in PEFT-based continual learning.

3 Proposed Method

3.1 Problem Formulation

In a continual learning scenario, the model sequentially observes T tasks $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_T$, where each task $\mathcal{D}_t = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ is drawn from its own data distribution. While training on the t-th task, the model has no access to samples from previous tasks ($< t$). Following the most challenging setting adopted from previous works (Wang et al., 2023a, 2025), at inference time the model is not provided with a task identifier indicating the origin of the test sample. The objective is therefore to maximize average performance across all tasks observed so far while preventing catastrophic forgetting under task-agnostic inference.

3.2 Structure

Figure 2 shows the overall architecture of the proposed method. The proposed method augments a frozen pre-trained LLM with a pool of M LoRA experts inserted into the attention layers, and routes each input through a sampling-based selector implemented as a conditional GFlowNet. Given an input x, the GFlowNet sampler produces a subset z of k expert indices according to a forward policy $q_\phi(z \mid x)$, and the selected experts are linearly combined to form a dynamic LoRA adaptation of the attention projections. The forward policy is trained under the trajectory balance objective so that subsets are sampled in proportion to a reward derived from the language modeling likelihood under z, while the LoRA parameters are trained under the standard language modeling loss. Upon completion of each task, the expert most frequently selected during that task is frozen and excluded from subsequent gradient updates, preserving task-specific knowledge while allowing the remaining experts to participate in forward transfer.

3.3 Multi-LoRA Expert Pool

Let $W_{\text{init}} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ denote the linear projection matrix of a pre-trained transformer layer. LoRA (Hu et al., 2022) approximates the weight update ΔW as the product of two low-rank matrices,

$$W_{\text{init}} + \Delta W = W_{\text{init}} + BA, \qquad (1)$$

with $A \in \mathbb{R}^{r \times d_{\text{in}}}$, $B \in \mathbb{R}^{d_{\text{out}} \times r}$, and $r \ll \min(d_{\text{in}}, d_{\text{out}})$. Following previous work on PEFT-based continual learning (Wang et al., 2023a; Zhao et al., 2024; Wang et al., 2025), LoRA is applied

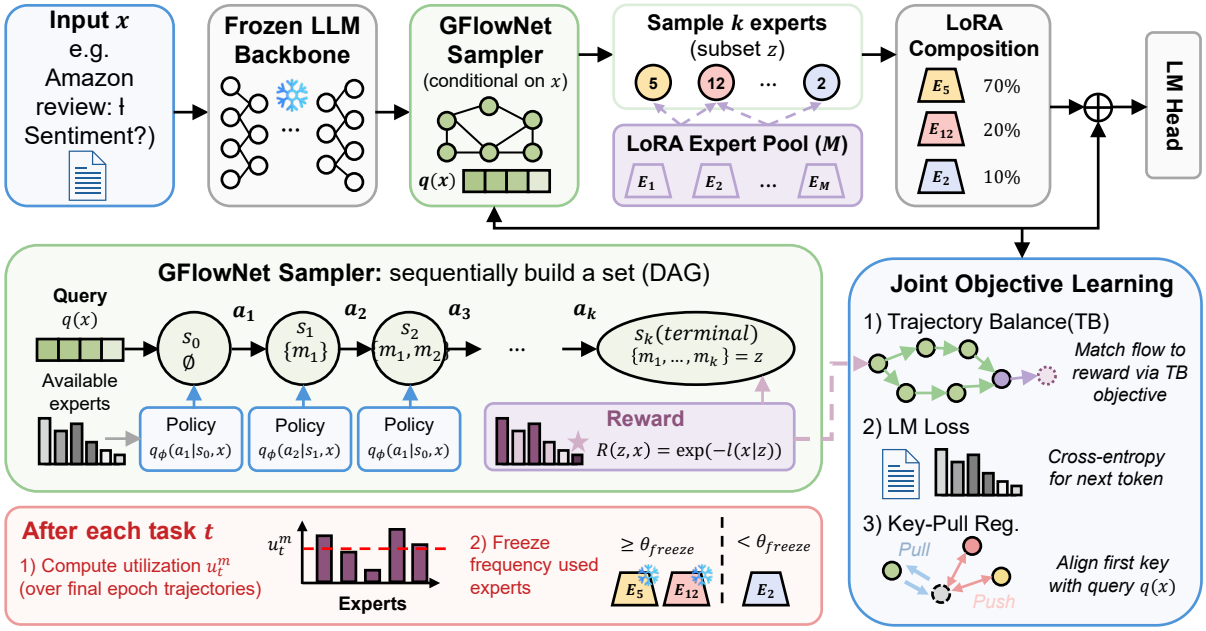

Figure 2: Overview of the proposed method.

only to the query (W_q) and value (W_v) projections of each attention layer.

The proposed method maintains a pool of M independent LoRA experts at each adapted layer. The m-th expert is parameterized by

- a down-projection matrix $A^{(m)} \in \mathbb{R}^{r \times d_{\text{in}}}$ initialized via Kaiming initialization,
- an up-projection matrix $B^{(m)} \in \mathbb{R}^{d_{\text{out}} \times r}$ initialized to zero, and
- a routing key $e^{(m)} \in \mathbb{R}^D$ that serves as a semantic descriptor of the expert in the GFlowNet state space.

Given an input activation x_b in a batch of size B and a sampled set of k expert indices $z_b = (z_{b,1}, \ldots, z_{b,k})$ produced by the GFlowNet sampler (Section 3.4), the dynamic LoRA contribution is computed as

$$\Delta(x)_b = s \cdot \sum_{j=1}^{k} w_{b,j} \cdot B^{(z_{b,j})} A^{(z_{b,j})} x_b, \quad (2)$$

where s is the standard LoRA scaling factor and $w_{b,j}$ denotes the mixing weight assigned to the j-th selected expert. The mixing weights are set to $w_{b,j} = 1/k$, leaving expert composition entirely to the GFlowNet sampler rather than to a secondary gating network.

3.4 Conditional GFlowNet Sampler

The selection of k experts from a pool of M is casted as a trajectory generation problem on a directed acyclic graph (DAG). The initial state s_0 corresponds to the empty selection, and at each step the policy appends a new expert index to the current selection, terminating at a state s_k that represents the full subset z of size k. The resulting trajectory $\tau = (s_0 \to s_1 \to \cdots \to s_k)$ uniquely determines the terminal state, and conversely each terminal state z can be reached by $k!$ distinct trajectories corresponding to the orderings of its elements.

Forward policy. The forward policy $q_\phi(a_i \mid s_i, x)$ is parameterized as a multi-layer perceptron conditioned on the input. Three components are concatenated to form the state representation:

1. the input query representation $q = h(x) \in \mathbb{R}^D$ extracted from the frozen backbone,
2. the selection context $c_i = \frac{1}{|s_i|} \sum_{m \in s_i} e^{(m)} \in \mathbb{R}^D$, obtained by averaging the routing keys of the experts selected so far, and
3. a step-position embedding $p_i \in \mathbb{R}^{32}$.

The concatenated vector $[q; c_i; p_i] \in \mathbb{R}^{2D+32}$ is mapped to M-dimensional logits through a two-hidden-layer MLP with GELU activations:

$$\ell_i = \text{MLP}_\phi([q; c_i; p_i]) \in \mathbb{R}^M. \quad (3)$$

To enforce valid actions, logits corresponding to (i) experts already included in s_i and (ii) experts frozen by previous tasks (Section 3.6) are clamped to $-\infty$ before applying the softmax. A categorical sample is then drawn, and the routing key of the selected expert is incorporated into c_{i+1} before the next step. The forward log-probability of the full trajectory is accumulated as

$$\log p_F(\tau \mid x) = \sum_{i=1}^{k} \log q_\phi(a_i \mid s_{i-1}, x). \quad (4)$$

Backward policy. Because the order in which experts are added to the selection does not affect the terminal state, the backward policy is fixed to the uniform distribution over the $|s_i|$ predecessors of each non-initial state, which assigns $p_B(\tau \mid z) = 1/k!$ to every trajectory ending at z. This choice removes the need to learn a separate backward network and is standard in set-valued GFlowNet applications (Bengio et al., 2021; Malkin et al., 2022).

3.5 Joint Training with TB Objective

The GFlowNet sampler aligns the marginal probability of terminating at z with the reward $R(z, x) = \exp(-\ell(x \mid z))$, where $\ell(x \mid z)$ is the cross-entropy loss. The trajectory balance loss (Malkin et al., 2022) is formulated as:

$$\mathcal{L}_{\text{TB}}(\tau, x) = \big(\log Z_\theta(x) + \log p_F(\tau \mid x) - \log p_B(\tau \mid z) - \beta \log R(z, x)\big)^2, \quad (5)$$

where $Z_\theta(x)$ is the partition function estimator and β controls reward scaling. The overall objective integrates $\mathcal{L}_{\text{TB}}$, language modeling loss, and a selective key-pull regularizer $\mathcal{L}_{\text{KP}}$:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{TB}} + \lambda_{\text{lm}} \cdot \ell(x \mid z) + \lambda_{\text{kp}} \cdot \mathcal{L}_{\text{KP}}. \quad (6)$$

$\mathcal{L}_{\text{KP}}$ stabilizes the forward policy state by aligning the routing key of the first-step expert $e^{(z_{b,1})}$ with the query $q(x_b)$. To prevent misalignment, it is computed exclusively over the high-reward subset $B^+ = \{b \mid R(z_b, x_b) \geq R_{\min}\}$:

$$\mathcal{L}_{\text{KP}} = \frac{1}{|B^+|} \sum_{b \in B^+} \Big(1 - \cos\big(\text{sg}(q(x_b)),\ e^{(z_{b,1})}\big)\Big), \quad (7)$$

where $\text{sg}(\cdot)$ is the stop-gradient operator. Gradient flow is explicitly partitioned: $\mathcal{L}_{\text{TB}}$ updates the policy MLP and Z_θ; $\ell(x \mid z)$ updates the LoRA parameters; and both $\mathcal{L}_{\text{TB}}$ and $\mathcal{L}_{\text{KP}}$ optimize the routing keys. Pre-trained LLM weights remain strictly frozen.

3.6 Task-agnostic Inference

To mitigate catastrophic forgetting, the proposed method implements usage-based freezing at task boundaries. Upon convergence of task t, expert utilization $u_t^{(m)}$ is computed over N_t trajectories from the final epoch:

$$u_t^{(m)} = \frac{1}{N_t} \sum_{n=1}^{N_t} \mathbb{1}\big[m \in z_n\big]. \quad (8)$$

Experts exceeding a defined utilization threshold are frozen:

$$\mathcal{F}_t^{\text{new}} = \big\{m \mid u_t^{(m)} \geq \theta_{\text{freeze}}\big\}. \quad (9)$$

The cumulative frozen set is updated via $\mathcal{F}_t = \mathcal{F}_{t-1} \cup \mathcal{F}_t^{\text{new}}$. Unlike fixed-count schema, this dynamic thresholding adapts to task complexity. During subsequent tasks, frozen experts are masked from the forward policy action space (logits clamped to $-\infty$), preventing parameter updates while remaining accessible for forward transfer.

Inference. Task-agnostic inference requires no external task identifiers. Given query $q = h(x)$, a size-k subset is selected by greedy decoding of the forward policy, taking the $\arg\max$ action at each of the k steps:

$$z^\star = (a_1^\star, \ldots, a_k^\star), \quad a_i^\star = \arg\max_a q_\phi(a \mid s_{i-1}, x). \quad (10)$$

The resulting dynamic LoRA contribution for $z^\star$ is subsequently injected into the attention layers for standard generation.

4 Experiments

4.1 Experimental Setup

The proposed method is evaluated on two LLM CL benchmarks: the O-LoRA long sequence (15 tasks) and TRACE (8 heterogeneous tasks). T5-large and LLaMA-2-7B serve as the primary pre-trained backbones and remain entirely frozen during training. Performance is compared against ten diverse baselines, spanning architecture-based, routing-based, and regularization methods, under a strictly controlled LoRA parameter budget ($M = 16$, $r = 8$, $\alpha = 2$) to isolate the contribution of the routing mechanism. Evaluation relies on Average

Accuracy and Forgetting for O-LoRA, task-aware Overall Performance (OP) for TRACE, and a newly introduced mode-collapse score to quantify expert utilization diversity. Comprehensive details regarding benchmark statistics, baseline taxonomy, mathematical formulations for metrics, and complete hyperparameters are deferred to Appendix A.

4.2 Main Results

O-LoRA long sequence benchmark. Table 1 reports results on the O-LoRA long sequence benchmark with T5-large. The proposed method significantly outperforms the strongest baseline (InfLoRA) in both accuracy and forgetting. Deterministic-routing methods (AM-LoRA, EPI, O-LoRA) exhibit substantial performance drops, indicating that strict task isolation or hard argmax routing restricts cross-task knowledge sharing. Conversely, soft routing and regularization approaches (SAPT, EWC-LoRA) suffer from high variance and severe forgetting. While REINFORCE achieves zero forgetting, this trivially results from policy collapse to a single expert, thereby degrading accuracy. The proposed method resolves this trade-off: probabilistic routing with reward-proportional sampling preserves stability while maintaining capacity for forward transfer. Per-order results follow identical rankings (Appendix B). For reference, the multi-task learning upper bound (joint training on all 15 tasks) attains approximately 0.80 accuracy on this benchmark; the proposed method therefore recovers approximately 92% of the joint-training performance under a strictly sequential setup.

TRACE benchmark. On the TRACE benchmark under the task-aware OP metric, the proposed method outperforms all baselines (Table 2). Per-task analysis (Appendix C) indicates that performance gains are concentrated in C-STANCE, FOMC, and NumGLUE-ds. However, all methods remain weak on ScienceQA due to the inherent difficulty of task-agnostic multi-choice generation. OP is reported as the primary metric because exact-match accuracy is overly stringent for generation tasks admitting multiple valid outputs; full exact-match results are provided in Appendix C.

Method	Acc ↑	F ↓
Ours	**0.7341 ± 0.0094**	**0.0084 ± 0.0015**
InfLoRA	0.6920 ± 0.0103	0.0131 ± 0.0009
AM-LoRA	0.6085 ± 0.0336	0.0163 ± 0.0012
EPI	0.5759 ± 0.0188	0.0218 ± 0.0016
O-LoRA	0.5795 ± 0.0035	0.0494 ± 0.0043
SAPT	0.4831 ± 0.0495	0.0175 ± 0.0042
Full-LoRA-FT	0.4555 ± 0.0952	0.3289 ± 0.1027
REINFORCE	**0.4242 ± 0.0102**	**0.0000 ± 0.0000**
EWC-LoRA	0.4181 ± 0.1484	0.0067 ± 0.0068
Uniform	0.1749 ± 0.0250	0.3517 ± 0.0149

Table 1: Final accuracy and forgetting on the O-LoRA long sequence benchmark with T5-large, averaged over Orders 4, 5, and 6. Results are mean ± standard deviation over three seeds per order.

Method	OP ↑
Ours	**0.1971 ± 0.0285**
InfLoRA	0.1896 ± 0.0034
EPI	0.1867 ± 0.0190
SAPT	0.1791 ± 0.0114
AM-LoRA	0.1678 ± 0.0349
EWC-LoRA	0.1617 ± 0.0140
O-LoRA	0.1041 ± 0.0244
Uniform	0.1034 ± 0.0487
Full-LoRA-FT	0.1022 ± 0.0339
REINFORCE	0.0764 ± 0.0058

Table 2: Task-aware Overall Performance (OP) on the TRACE benchmark with T5-large. Results are mean ± standard deviation over three seeds.

LLaMA-2-7B robustness. To verify that the observed ranking transfers to larger LLM backbones, the same Order 6 setup is repeated on LLaMA-2-7B with bfloat16. The qualitative ranking observed on T5-large is preserved: the proposed method attains 0.3693 ± 0.0217 accuracy with 0.0065 ± 0.0038 forgetting, exceeding the strongest baseline AM-LoRA (0.3076 ± 0.0036, 0.0116 ± 0.0014) by $+6.17$ points. Full LLaMA-2-7B results are reported in Appendix D.

4.3 Analysis

4.3.1 Mode Collapse in Continual Routing

A central premise of this paper is that deterministic gating concentrates gradient updates on a small subset of experts, leaving others under-trained. To validate this, expert utilization across the O-LoRA Order 6 benchmark is measured using the mode-collapse score introduced in Section 4.1 (Figure 3). Empirical analysis reveals that EWC-LoRA (0.687) and SAPT (0.620) exhibit substantial collapse, concentrating probability mass on a small subset of experts. The REINFORCE baseline collapses to a single expert (1.000), confirming that standard policy gradients fail to maintain routing diversity. The proposed method attains an aggregate collapse score of 0.4128, statistically on par with the deterministic

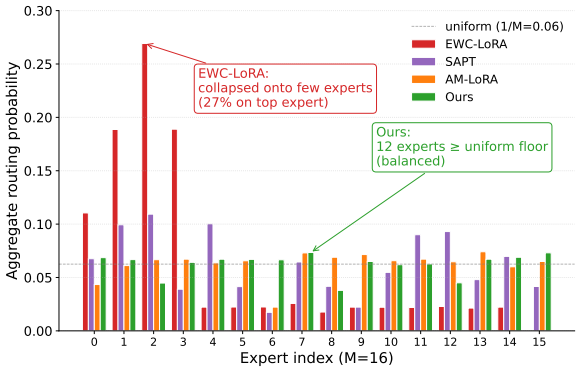

Figure 3: Expert utilization across the full task stream on the Order 6 benchmark. The dashed horizontal line marks uniform utilization ($1/M = 0.0625$). O-LoRA is excluded as its architecture enforces one expert per task by construction.

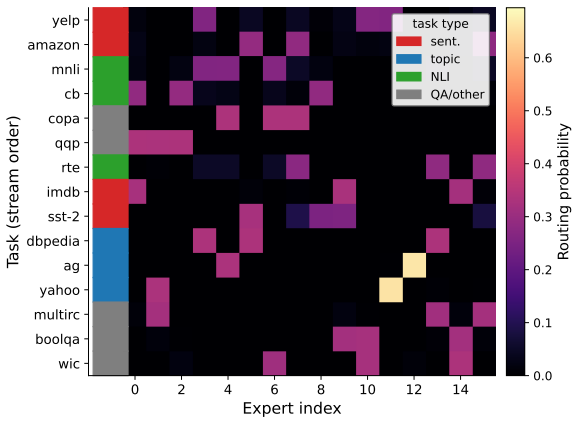

Figure 4: Per-task routing distribution of the proposed method on Order 6. Left strip indicates task category.

Variant	Acc ↑	Mode-collapse ↓
V1: Ours	**0.7362 ± 0.0087**	**0.4128 ± 0.0021**
V2: Softmax mixture	0.7048 ± 0.0041	0.6781 ± 0.0578
V3: Softmax argmax	0.4670 ± 0.0677	0.9461 ± 0.0294

Table 3: Routing mechanism ablation on the O-LoRA Order 6 benchmark with T5-large. All variants share the same LoRA pool, key-pull regularizer, freeze-on-use mechanism, and optimizer; only the routing mechanism and its training objective differ.

AM-LoRA and InfLoRA baselines (0.4148); the decisive difference is the per-seed standard deviation, which is an order of magnitude larger for the proposed method (0.0021 vs. 0.0002). Whereas the deterministic baselines reduce to a fixed routing map identical across random seeds, the trajectory balance objective induces seed-dependent yet consistently high-reward expert subsets, mitigating expert under-utilization without defaulting to deterministic assignments. The near-zero score of O-LoRA (0.0667) is an architectural artifact of its one-task-one-expert allocation rather than learned diversity, and is therefore excluded from the learned-method comparison.

4.3.2 Structured Expert Sharing

Previously, we identified *redundant capacity allocation across semantically related tasks* as a structural limitation of existing PEFT-based CL methods. We assess whether the proposed routing addresses this by visualizing the per-task routing distribution over the M=16 expert pool in Figure 4.

The four sentiment tasks (Yelp, Amazon, IMDB, SST-2) concentrate on a shared expert subset, the three topic-classification tasks (DBpedia, AG News, Yahoo) recruit a disjoint subset, and reading-comprehension tasks (MultiRC, BoolQA) form a third subset that partially overlaps with both. Natural-language-inference tasks (MNLI, CB, RTE) form a fourth cluster that partially overlaps with the reading-comprehension subset, while the smaller singleton tasks (COPA, QQP, WiC) each recruit a single dedicated expert. This task-family structure is acquired purely from the LM reward under the trajectory balance objective, without any explicit supervision of which tasks should share experts, directly addressing the redundant-allocation limitation identified in Section 1. A method-by-method comparison demonstrating that baseline routers fail to recover this structure is deferred to Appendix H.

4.3.3 Plasticity vs. Stability

Catastrophic forgetting reflects a trade-off between two capabilities: plasticity, the ability to acquire new task knowledge, and stability, the ability to retain previously learned knowledge. Figure 5 plots both quantities for ten LLM-CL methods on the O-LoRA Order 6 benchmark. The proposed method occupies the upper-right region with plasticity 0.742 and stability 0.992, while Full-LoRA-FT achieves comparable plasticity (0.761) but loses approximately one-third of past-task accuracy by the end of sequence. REINFORCE attains trivial stability (1.000) by collapsing onto a single expert and failing to acquire new task knowledge.

4.4 Ablation Studies

4.4.1 Routing Mechanism

To isolate the contribution of the GFlowNet sampler from other architectural choices, three rout-

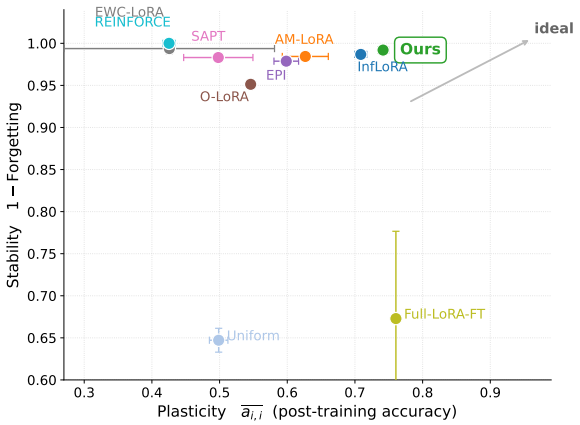

Figure 5: Plasticity vs. stability on the O-LoRA Order 6 benchmark. Plasticity is the average post-training accuracy $\overline{a_{i,i}}$, and stability is $1 - \mathrm{F}$. The upper-right region is ideal. Error bars denote ± 1 SD over three seeds.

ing variants are compared while holding the LoRA pool configuration, key-pull regularizer, freeze-on-use mechanism, and optimizer identical. V1 (the proposed method) performs stochastic sampling $z \sim q_\phi(z \mid x)$ at training time and greedy decoding at inference, trained with the trajectory balance loss. V2 replaces the GFlowNet sampler with a softmax-weighted mixture of experts (Switch-style), trained with the language modeling loss plus a standard load-balancing auxiliary loss. V3 replaces the sampler with a hard argmax over softmax logits, using straight-through estimator gradients, with the same load-balancing auxiliary loss.

Table 3 reports the comparison. V2 spreads gradient signal across multiple experts but the resulting per-expert update magnitude is small, limiting specialization and reducing accuracy by approximately 3 points. V3 recovers per-expert specialization but inherits the deterministic concentration of gradients onto a few experts, collapsing to a near-single-expert distribution and losing 27 accuracy points. Only V1 attains both balanced capacity utilization and competitive accuracy, supporting the claim that reward-proportional sampling under the trajectory balance objective is qualitatively different from both soft and hard softmax routing.

4.4.2 Compositional Generalization

The second structural limitation identified in Section 1—limited compositional reuse under hybrid task shifts—is evaluated by training on the first 12 tasks of the O-LoRA Order 6 sequence and measuring zero-shot accuracy on the three held-out tasks (BoolQA, MultiRC, WiC). A method that has learned to recompose previously trained experts should generalize to these unseen tasks by activating relevant expert subsets without further training.

Method	BoolQA	MultiRC	WiC	Avg
Ours	**0.731**	**0.582**	**0.489**	**0.601**
EWC-LoRA	0.142	0.058	0.062	0.087
InfLoRA	0.024	0.005	0.008	0.012
AM-LoRA	0.018	0.002	0.008	0.009

Table 4: Zero-shot accuracy on three held-out tasks of the O-LoRA Order 6 sequence after training only on the first 12 tasks. Results are mean over three seeds. The effective number of experts activated on held-out inputs is 2.62 for the proposed method and 2.49 for AM-LoRA, indicating that the accuracy gap stems from selecting an appropriate expert combination rather than activating more experts.

Table 4 reports the comparison. The proposed method achieves 0.601 average zero-shot accuracy on the held-out tasks, compared to 0.009 for AM-LoRA and 0.012 for InfLoRA. Both deterministic baselines fail because their routing functions are tied to task identities or task-specific keys observed during training; unseen tasks cannot activate any meaningful expert combination. EWC-LoRA, which uses a single shared LoRA without routing, achieves 0.087 because its parameters retain partial general-purpose capacity but cannot specialize. The effective number of experts activated by the proposed method on held-out inputs (2.62 on average) is comparable to that of AM-LoRA (2.49), indicating that the improvement does not stem from activating more experts but from selecting a more appropriate combination—direct empirical evidence for the compositional recomposition mechanism motivated in Section 1. Further ablations are reported in Appendix E.

5 Conclusion

This paper formulates expert selection in CL as amortized posterior inference via a conditional GFlowNet sampler. By utilizing reward-proportional sampling, the proposed method explicitly resolves redundant capacity allocation and restricted compositional generalization inherent in prior methods. Empirical results confirm balanced stability and plasticity under task-agnostic inference. Future work will investigate modifying the reward function to incentivize plasticity on subsequent unseen tasks, thereby directly optimizing positive forward transfer.

Limitations. The proposed method introduces non-trivial computational overhead during training; optimizing the trajectory balance objective increases wall-clock time by approximately 30% compared to static methods like O-LoRA (Appendix I). Additionally, the global utilization threshold (θ_{freeze}) assumes uniform task complexity, risking capacity exhaustion under extreme task heterogeneity. Finally, the auxiliary key-pull regularizer ($\mathcal{L}_{\text{KP}}$) yields negligible empirical gains (Appendix E), indicating redundant constraints in the loss formulation that require theoretical refinement. The fixed expert pool size $M{=}16$ implicitly assumes a bounded number of distinct task families. Highly heterogeneous streams beyond this scale may require a hierarchical pool extension, which is left to the future work.

Potential Risks: Distributed Knowledge Correction and Unlearning. The compositional and distributed nature of the proposed mixture of LoRA experts introduces a significant challenge regarding machine unlearning and epistemic correction. Because the GFlowNet sampler promotes structured parameter sharing across related tasks, newly acquired knowledge is not isolated within a single dedicated adapter but is instead fractionally encoded across overlapping subsets of experts. Consequently, if the model inadvertently absorbs erroneous, biased, or toxic information during the sequential training stream, locating and purging the specific parameters responsible for the misinformation becomes structurally difficult. Executing surgical parameter edits or targeted erasure without disrupting the coupled representations shared by valid tasks remains non-trivial, thereby risking the persistent propagation of distributed errors within the network.

References

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. 2021. Flow network based generative models for non-iterative diverse candidate generation. *Advances in neural information processing systems*, 34:27381–27394.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. 2023. Gflownet foundations. *Journal of Machine Learning Research*, 24(210):1–55.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R Costa-Jussa. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th international conference on computational linguistics*, pages 6523–6541.

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.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Edward Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. 2024. Amortizing intractable inference in large language models. In *International Conference on Learning Representations*, volume 2024, pages 43053–43083.

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.

Moksh Jain, Sharath Chandra Raparthy, Alex Hernández-Garcıa, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. 2023. Multi-objective gflownets. In *International conference on machine learning*, pages 14631–14653. PMLR.

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.

Minh Le, An Nguyen, Huy Nguyen, Trang Nguyen, Trang Pham, Linh Van Ngo, and Nhat Ho. 2024. Mixture of experts meets prompt-based continual learning. *Advances in Neural Information Processing Systems*, 37:119025–119062.

Yan-Shuo Liang and Wu-Jun Li. 2024. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23638–23647.

Jialin Liu, Jianhua Wu, Jie Liu, and Yutai Duan. 2024. Learning attentional mixture of loras for language model continual learning. *arXiv preprint arXiv:2409.19611*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. 2022. Trajectory balance: Improved credit assignment in gflownets. *Advances in Neural Information Processing Systems*, 35:5955–5967.

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.

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*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

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

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5362–5383.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. 2023a. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671.

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*.

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

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

Zitao Wang, Xinyi Wang, and Wei Hu. 2025. Mixture of lora experts for continual information extraction with llms. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13324–13339.

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

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024. Sapt: A shared attention framework for parameter-efficient continual learning of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11641–11661.

A Experimental Setup Details

A.1 Benchmark Details

The O-LoRA long sequence benchmark spans 15 tasks covering text classification, NLI, paraphrase detection, sentiment analysis, and reading comprehension, drawn from the CL benchmark of Zhang et al. (2015), GLUE (Wang et al., 2018), SuperGLUE (Wang et al., 2019), and IMDB (Maas et al., 2011). For each task, 1,000 examples are sampled for training, and up to 500 examples per class are reserved for validation following Razdaibiedina et al. (2023). Maximum input and target lengths are set to 512 and 50, respectively. The three task orders (Order 4, 5, 6) follow the configurations of Razdaibiedina et al. (2023) and Wang et al. (2023a); Order 6 is used as the primary evaluation setup for analysis and ablations because it interleaves task families most aggressively.

The three task orders adopted for the O-LoRA long sequence benchmark follow the standard configurations of Razdaibiedina et al. (2023) and Wang et al. (2023a) (see Table 5). Order 6 is used as the primary evaluation setup for analysis and ablations because it places the four sentiment tasks at the start and interleaves remaining task families most aggressively.

Order	Task sequence
Order 4	mnli → cb → wic → copa → qqp → boolqa → rte → imdb → yelp → amazon → sst-2 → dbpedia → ag → multirc → yahoo
Order 5	multirc → boolqa → wic → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yelp → amazon → yahoo
Order 6	yelp → amazon → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yahoo → multirc → boolqa → wic

Table 5: Task ordering for the three O-LoRA long sequence configurations.

The TRACE benchmark consists of 8 tasks explicitly designed for LLM continual learning, combining classification, generation, and code-completion: C-STANCE, FOMC, MeetingBank, Py150, ScienceQA, NumGLUE-cm, NumGLUE-ds, and 20Minuten. Each task has a designated evaluation metric (Table 6); the task-aware Overall Performance (OP) averages these per-task metrics rather than applying a single uniform metric across heterogeneous task types, following the official TRACE evaluation protocol of Wang et al. (2023b).

Task	Metric
C-STANCE	Accuracy
FOMC	F1-macro
MeetingBank	ROUGE-L
Py150	$0.5 \cdot \text{EM} + 0.5 \cdot \text{edit-similarity}$
ScienceQA	Accuracy
NumGLUE-cm	Accuracy
NumGLUE-ds	Accuracy
20Minuten	ROUGE-L

Table 6: Per-task evaluation metrics for the TRACE benchmark.

A.2 Evaluation Metrics

To rigorously assess the performance of continual learning, three primary metrics are computed at the end of the task stream. Let T denote the total number of tasks and $a_{i,j}$ denote the evaluation performance on task j after training on task i.

- **Average Accuracy (Acc):** Measures the final retained performance across all tasks:

$$\text{Acc} = \frac{1}{T}\sum_{t=1}^{T} a_{T,t}$$

.

- **Forgetting (F):** Quantifies the performance drop from the maximum accuracy achieved on a task:

$$\text{F} = \frac{1}{T-1}\sum_{t=1}^{T-1}\left(\max_{i\geq t} a_{i,t} - a_{T,t}\right)$$

.

- **Mode-Collapse Score:** Evaluates the diversity of expert routing. Given the aggregated expert utilization probability distribution and its entropy H, the score normalizes the collapse severity:

$$\text{Score} = \frac{M - e^{H}}{M - 1}$$

. A score of 0 indicates perfectly uniform utilization, whereas a score of 1 indicates complete collapse to a single expert.

All reported results aggregate performance over three distinct random seeds, presenting the mean and standard deviation.

A.3 Backbone Configuration

T5-large (Raffel et al., 2020) (770M parameters, float32) is used as the seq2seq backbone with LoRA modules attached to encoder query/value projections, decoder query/value projections, and cross-attention query/value projections, yielding 144 wrapped layers in total. LLaMA-2-7B (Touvron et al., 2023) (6.7B parameters, bfloat16) is used as the causal LM backbone with LoRA modules attached to `q_proj` and `v_proj` of all 32 transformer layers, yielding 64 wrapped layers. The pre-trained backbone weights remain frozen across all methods, ensuring that all observed performance discrepancies arise strictly from the respective routing and adaptation mechanisms.

A.4 Baseline Configurations

Ten baselines are evaluated and categorized into four distinct groups:

1. **Architecture-based CL:** O-LoRA (Wang et al., 2023a), InfLoRA (Liang and Li, 2024), and EPI (Wang et al., 2023c), which rely on per-task subspace isolation or strict parameter expansion.

2. **Routing-based MoE-LoRA:** AM-LoRA (Liu et al., 2024) and SAPT (Zhao et al., 2024), which utilize soft or deterministic token-level routing.

3. **Regularization:** EWC-LoRA (Kirkpatrick et al., 2017), which penalizes parameter drift based on Fisher information.

4. **Sanity Baselines:** REINFORCE-based stochastic routing (replacing the proposed trajectory balance objective with standard policy gradients), Full-LoRA-FT (naive sequential finetuning), and Uniform routing.

The hyperparameters for each method follow the respective original papers: O-LoRA uses $\lambda_1 = 0.5$ and $\lambda_2 = 0$; EWC-LoRA uses Fisher weight $\lambda = 5000$; EPI uses Gaussian width σ estimated from per-task feature statistics. Crucially, all baselines share the exact same LoRA pool configuration as the proposed method ($M = 16$, $r = 8$, $\alpha = 2$, identical target modules) to guarantee that comparisons isolate the routing mechanism's efficacy rather than arbitrary parameter scaling.

A.5 Implementation Details and Hyperparameters

All trainable components are optimized with AdamW (weight decay 0, gradient clipping 1.0). Learning rates are determined via a rigorous Bayesian sweep exceeding 100 trials and are held fixed across all subsequent experimental runs: policy MLP 1.01×10^{-4}, LoRA matrices 1.20×10^{-4}, expert routing keys 2.30×10^{-4}, and the partition function $\log Z$ at 3.59×10^{-3}. Additional critical hyperparameters include: LoRA rank $r = 8$, scaling $\alpha = 2$, pool size $M = 16$, reward temperature $\beta = 2.04$, key-pull weight $\lambda_{\text{KP}} = 3.45$ with reward threshold $R_{\min} = 0.53$, policy warmup of 70 steps, and freezing threshold $\theta_{\text{freeze}} = 0.292$. Batch size is 8 for T5-large and 2 for LLaMA-2-7B; maximum target length is 50 for T5-large and 64 for LLaMA-2-7B. Table 7 summarizes the complete configuration.

Component	Value
LoRA rank r	8
LoRA scaling α	2
Expert pool size M	16
Active experts k per input	3
Reward temperature β	2.04
Key-pull weight λ_{KP}	3.45
Key-pull reward floor $R_{\min}$	0.53
Freeze threshold θ_{freeze}	0.292
Policy warmup steps	70
Task warmup steps	39
Policy LR	1.01×10^{-4}
LoRA LR	1.20×10^{-4}
Routing-key LR	2.30×10^{-4}
$\log Z$ LR	3.59×10^{-3}
Optimizer	AdamW
Weight decay	0
Gradient clipping	1.0

Table 7: Hyperparameter snapshot for the proposed method.

B Per-order Results on the O-LoRA Long Sequence Benchmark

Table 8 reports per-order accuracy and forgetting on the O-LoRA long sequence benchmark with T5-large. The qualitative ranking is preserved across all three orders, indicating that the observed advantage of the proposed method is not specific to a particular task ordering.

C Per-task Results on the TRACE Benchmark

Table 9 reports per-task scores under both exact-match accuracy and the task-aware OP metric on

Method	Order 4		Order 5		Order 6	
	Acc ↑	F ↓	Acc ↑	F ↓	Acc ↑	F ↓
Ours	**0.7314 ± 0.0094**	**0.0089 ± 0.0017**	**0.7348 ± 0.0102**	**0.0083 ± 0.0014**	**0.7362 ± 0.0087**	**0.0080 ± 0.0013**
InfLoRA	0.6912 ± 0.0103	0.0136 ± 0.0011	0.6883 ± 0.0114	0.0125 ± 0.0012	0.6964 ± 0.0091	0.0132 ± 0.0004
AM-LoRA	0.6087 ± 0.0298	0.0167 ± 0.0015	0.6045 ± 0.0367	0.0166 ± 0.0013	0.6123 ± 0.0344	0.0156 ± 0.0008
EPI	0.5732 ± 0.0184	0.0221 ± 0.0017	0.5754 ± 0.0203	0.0220 ± 0.0015	0.5790 ± 0.0177	0.0212 ± 0.0014
O-LoRA	0.5803 ± 0.0042	0.0498 ± 0.0046	0.5821 ± 0.0035	0.0496 ± 0.0045	0.5760 ± 0.0027	0.0487 ± 0.0041
SAPT	0.4762 ± 0.0518	0.0177 ± 0.0043	0.4905 ± 0.0476	0.0181 ± 0.0042	0.4826 ± 0.0492	0.0169 ± 0.0039
Full-LoRA-FT	0.4421 ± 0.0987	0.3318 ± 0.1024	0.4587 ± 0.0945	0.3279 ± 0.1018	0.4656 ± 0.0924	0.3271 ± 0.1038
REINFORCE	0.4198 ± 0.0124	0.0000 ± 0.0000	0.4275 ± 0.0094	0.0000 ± 0.0000	0.4253 ± 0.0088	0.0000 ± 0.0000
EWC-LoRA	0.4087 ± 0.1532	0.0072 ± 0.0073	0.4256 ± 0.1421	0.0068 ± 0.0070	0.4201 ± 0.1499	0.0062 ± 0.0062
Uniform	0.1812 ± 0.0287	0.3496 ± 0.0153	0.1689 ± 0.0223	0.3527 ± 0.0156	0.1746 ± 0.0241	0.3528 ± 0.0142

Table 8: Per-order accuracy and forgetting on the O-LoRA long sequence benchmark with T5-large. Results are mean ± standard deviation over three seeds.

the TRACE benchmark with T5-large. Under strict exact-match accuracy, the strongest methods cluster within a narrow range ($0.149-0.151$), reflecting that exact-match penalizes valid paraphrases in generation tasks such as MeetingBank and 20Minuten. Under OP, the proposed method achieves the highest score in five of the eight tasks; all methods remain weak in ScienceQA due to the multi-choice short-generation format under task-agnostic inference.

D LLaMA-2-7B Results

Table 10 reports full results on LLaMA-2-7B with the O-LoRA Order 6 benchmark. Runs are performed in bfloat16 with batch size 2 and maximum target length 64; all other hyperparameters are identical to the T5-large configuration. The qualitative ranking observed on T5-large is preserved, with the proposed method outperforming AM-LoRA by $+6.17$ accuracy points and reducing forgetting by approximately 44%.

E Additional Ablations

Key-pull regularizer. Table 11 compares four configurations of the key-pull regularizer $\mathcal{L}_{\text{KP}}$ in the O-LoRA Order 6 benchmark. Removing $\mathcal{L}_{\text{KP}}$ entirely, removing the reward-threshold filter B^+, or replacing the cosine distance with L_2 distance all yield accuracy and forgetting within the noise floor of the full configuration. The regularizer is retained in the released configuration for reproducibility, but the experimental evidence suggests that the routing keys receive sufficient gradient signal through the trajectory balance objective alone. Investigating alternative key-learning mechanisms such as contrastive objectives is left to future work.

Freeze-threshold sensitivity. Table 12 reports accuracy and forgetting as a function of the freeze threshold θ_{freeze}. The optimum lies near the value selected by the Bayesian sweep ($\theta_{\text{freeze}} = 0.292$). Lower thresholds freeze too many experts and reduce plasticity, while higher thresholds permit overwriting of task-specific knowledge and increase forgetting. The method is reasonably robust within a band of ± 0.1 around the optimum.

Per-task accuracy evolution. Figure 6 plots per-task accuracy trajectories over the 15-task sequence for four representative methods. The proposed method's accuracy lines are nearly flat after each task is learned, indicating sustained retention. InfLoRA exhibits similar flatness but with lower post-training accuracy. AM-LoRA shows mild drift on a subset of tasks. Full-LoRA-FT exhibits sharp dips throughout, the canonical signature of catastrophic forgetting under a shared LoRA.

GFlowNet training dynamics. Figure 7 reports the trajectory balance loss, language modeling loss, partition function estimate $\log Z$, and mean reward over the full 1832-step training run on Order 6 with seed 42. The trajectory balance loss exhibits an initial transient and stabilizes within 200 steps. The partition function $\log Z$ adjusts step-wise at each task boundary, reflecting task-adaptive normalization. Mean reward saturates near 0.95, indicating that routing consistently selects high-reward expert subsets.

F Plasticity Decomposition

Plasticity is defined as the average post-training accuracy $\overline{a_{i,i}}$ before any subsequent task is observed, capturing the model's capacity to acquire new task knowledge. Stability is defined as $1 - \text{F}$, capturing

Method	C-STANCE	FOMC	MeetingBank	Py150	ScienceQA	NumGLUE-cm	NumGLUE-ds	20Minuten
Ours	**0.363**	**0.347**	0.184	**0.221**	0.003	0.142	**0.262**	0.055
InfLoRA	0.341	0.318	**0.192**	0.214	0.004	0.148	0.247	0.053
EPI	0.336	0.312	0.189	0.209	0.002	**0.151**	0.241	0.054
SAPT	0.327	0.298	0.181	0.206	0.003	0.146	0.225	0.047
AM-LoRA	0.298	0.276	0.172	0.198	0.002	0.139	0.213	0.044
EWC-LoRA	0.279	0.264	0.168	0.189	0.002	0.134	0.205	0.043
O-LoRA	0.184	0.167	0.108	0.123	0.002	0.087	0.121	0.041
Full-LoRA-FT	0.176	0.158	0.106	0.118	0.001	0.083	0.119	0.042
Uniform	0.182	0.162	0.105	0.121	0.002	0.085	0.117	0.041
REINFORCE	0.134	0.118	0.082	0.094	0.002	0.064	0.087	0.030

Table 9: Per-task OP scores on the TRACE benchmark with T5-large. Results are mean over three seeds.

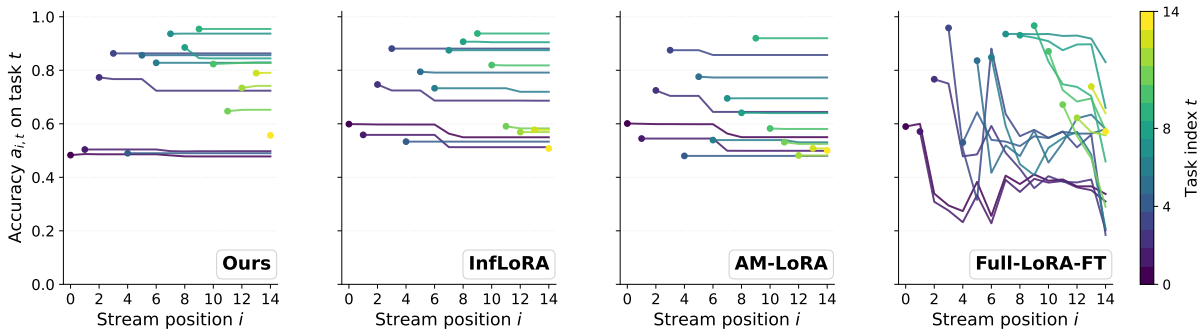

Figure 6: Per-task accuracy trajectories on the O-LoRA Order 6 benchmark. Each colored line represents one task's accuracy across subsequent training steps.

Method	Acc ↑	F ↓
Ours	**0.3693 ± 0.0217**	**0.0065 ± 0.0038**
AM-LoRA	0.3076 ± 0.0036	0.0116 ± 0.0014
InfLoRA	0.2987 ± 0.0142	0.0148 ± 0.0024
EPI	0.2814 ± 0.0098	0.0193 ± 0.0021
O-LoRA	0.2756 ± 0.0067	0.0421 ± 0.0058
SAPT	0.2398 ± 0.0312	0.0162 ± 0.0047
EWC-LoRA	0.2147 ± 0.0876	0.0089 ± 0.0072
Full-LoRA-FT	0.2018 ± 0.0623	0.2914 ± 0.0784
REINFORCE	0.1986 ± 0.0102	0.0000 ± 0.0000
Uniform	0.0823 ± 0.0184	0.3127 ± 0.0162

Table 10: Final accuracy and forgetting on the O-LoRA Order 6 benchmark with LLaMA-2-7B. Results are mean ± standard deviation over three seeds.

retention. Table 13 reports both quantities for all methods on the O-LoRA Order 6 benchmark.

G Mode-Collapse Scores

Table 14 reports per-method aggregate mode-collapse scores on the O-LoRA Order 6 benchmark. The score is computed from the entropy of the aggregate expert utilization distribution across the full task stream as $(M - e^H)/(M - 1)$, where 0 corresponds to perfectly uniform utilization and 1 to a single dominant expert. Effective expert count e^H is reported alongside as an interpretable complement.

Variant	Acc ↑	F ↓
Full $\mathcal{L}_{\text{KP}}$ (Ours)	**0.7362 ± 0.0087**	**0.0080 ± 0.0013**
w/o $\mathcal{L}_{\text{KP}}$	0.7169 ± 0.0088	0.0048 ± 0.0010
w/o B^+ filter	0.6862 ± 0.0086	0.0190 ± 0.0021
L_2 distance	0.6769 ± 0.0088	0.0048 ± 0.0010

Table 11: Key-pull regularizer ablation on the O-LoRA Order 6 benchmark with T5-large. The regularizer has no measurable effect on the final metrics.

θ_{freeze}	Acc ↑	F ↓
0.10	0.7124 ± 0.0156	0.0094 ± 0.0018
0.20	0.7287 ± 0.0102	0.0086 ± 0.0015
0.292 (proposed)	**0.7362 ± 0.0087**	**0.0080 ± 0.0013**
0.40	0.7298 ± 0.0098	0.0107 ± 0.0019
0.50	0.7156 ± 0.0142	0.0163 ± 0.0028
0.70	0.6823 ± 0.0234	0.0287 ± 0.0042

Table 12: Freeze-threshold sensitivity on the O-LoRA Order 6 benchmark with T5-large.

H Per-method Routing Comparison

Figure 8 extends Figure 4 to a side-by-side comparison with three representative baselines on the same Order 6 stream.

InfLoRA exhibits a strict diagonal pattern, reflecting its hard one-task-to-one-expert allocation; no cross-task sharing is possible by construction.

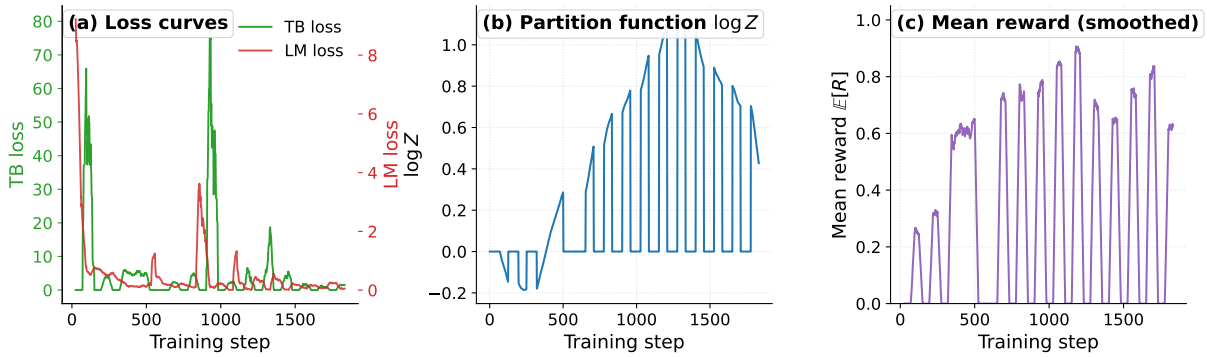

Figure 7: GFlowNet training dynamics on the O-LoRA Order 6 benchmark with T5-large (seed 42). Vertical dashed lines mark task boundaries.

Method	Plasticity ↑	Stability ↑
Full-LoRA-FT	**0.7605 ± 0.0019**	0.6729 ± 0.1038
Ours	0.7416 ± 0.0081	**0.9920 ± 0.0013**
InfLoRA	0.7086 ± 0.0088	0.9868 ± 0.0004
AM-LoRA	0.6267 ± 0.0340	0.9844 ± 0.0008
EPI	0.5985 ± 0.0182	0.9788 ± 0.0014
O-LoRA	0.5459 ± 0.0052	0.9513 ± 0.0041
Uniform	0.4987 ± 0.0137	0.6472 ± 0.0142
SAPT	0.4981 ± 0.0512	0.9831 ± 0.0039
EWC-LoRA	0.4258 ± 0.1554	0.9938 ± 0.0062
REINFORCE	0.4253 ± 0.0088	1.0000 ± 0.0000

Table 13: Plasticity and stability decomposition on the O-LoRA Order 6 benchmark with T5-large. The proposed method achieves the highest stability while retaining plasticity competitive with the catastrophically forgetting Full-LoRA-FT baseline.

Method	Collapse score ↓	Eff. experts ↑
O-LoRA	0.0667 ± 0.0000	15.00
Ours	**0.4128 ± 0.0021**	**9.81**
InfLoRA	0.4148 ± 0.0002	9.78
AM-LoRA	0.4148 ± 0.0002	9.78
EPI	0.4196 ± 0.0004	9.71
SAPT	0.6199 ± 0.0719	6.70
EWC-LoRA	0.6870 ± 0.2046	5.69
Full-LoRA-FT	1.0000 ± 0.0000	1.00
REINFORCE	1.0000 ± 0.0000	1.00

Table 14: Per-method mode-collapse scores on the O-LoRA Order 6 benchmark with T5-large. O-LoRA's near-zero score reflects its architectural one-task-one-expert allocation (perfectly uniform across the 15 trained experts). The proposed method matches AM-LoRA/InfLoRA in aggregate score but exhibits an order-of-magnitude higher per-seed standard deviation (0.0021 vs. 0.0002), indicating that the GFlowNet sampler diversifies routing across random seeds while deterministic baselines reduce to fixed mappings.

EWC-LoRA collapses most routing mass onto a single expert (consistent with the mode-collapse metric reported in Section 4.3.1). AM-LoRA distributes mass across multiple experts but does not cluster semantically related tasks: sentiment and topic-classification tasks select largely overlapping expert subsets, indicating that the cosine routing function fails to discover task-family structure. Only the proposed probabilistic router recovers the sentiment-/topic-/comprehension-clustering visible in Figure 4.

I Computational Cost

Table 15 reports parameter counts and wall-clock time for the proposed method and representative baselines. The proposed method introduces 1.34M additional parameters over a LoRA-pool baseline (3.5% relative overhead from the GFlowNet policy MLP) and incurs approximately 1.3× the training wall-clock time of O-LoRA on T5-large. Inference latency overhead is approximately 13% relative to a single-LoRA baseline. On LLaMA-2-7B, a single 15-task Order 6 run requires approximately 27 hours on a single GPU; backbone cost dominates, and the GFlowNet overhead is negligible relative to base model forward passes.

Inference latency. Single-batch (B=8) forward latency on T5-large is approximately 140 ms for the frozen backbone alone, 155 ms (+10%) with an active LoRA pool, and 158 ms (+13% total) when the GFlowNet policy argmax routing is enabled. The routing overhead of ∼ 3 ms per batch is dominated by a single MLP forward pass and is invoked once per batch rather than per token.

Memory footprint. Peak GPU memory during training is approximately 3.5 GB for T5-large alone (frozen, float32, B=8, seq. length 512), 4.2 GB with the LoRA pool, and 5.1 GB with full GFlowNet training including policy for-

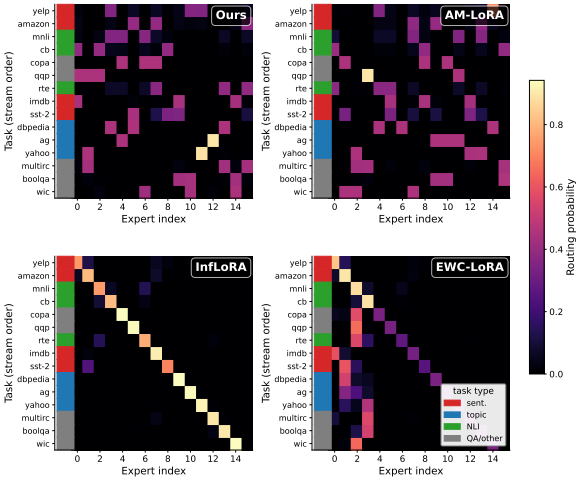

Figure 8: Per-task routing distribution over the $M{=}16$ LoRA experts on Order 6, comparing the proposed method against InfLoRA, AM-LoRA, and EWC-LoRA. Rows are tasks in stream order, columns are experts, color is routing probability, and the left strip indicates task category.

ward/backward passes and the optimizer state. For LLaMA-2-7B in bfloat16 with $B{=}2$, peak memory is approximately 27 GB, dominated by the frozen backbone; the GFlowNet components add less than 1 GB. Training comfortably fits on 24 GB consumer-grade GPUs for T5-large and on 40 GB GPUs for LLaMA-2-7B.

TRACE wall-clock. For completeness, on the TRACE 8-task benchmark the proposed method incurs approximately 127 min per run ($\sim$ 16 min per task), compared to 68.7 min for O-LoRA and 24.1 min for EPI. The per-task overhead pattern matches the O-LoRA benchmark.

Method	Trainable params	Wall (min)
SAPT	38.2M	33.8 ± 0.6
Full-LoRA-FT	2.4M	148.1
O-LoRA	36.0M	182.3 ± 5.4
Ours	39.1M	235.3 ± 19.9
REINFORCE	39.1M	255.6
EWC-LoRA	2.4M	284.8 ± 14.8

Table 15: Trainable parameter count and wall-clock time for one full O-LoRA Order 6 run with T5-large on a single GPU.