Routing-Aware Expert Calibration for Machine Unlearning in Mixture-of-Experts Language Models

Anonymous ACL submission

Abstract

Machine unlearning is increasingly important for large language models, yet unlearning in Mixture-of-Experts (MoE) architectures remains underexplored. Unlike dense models, MoE architectures employ a router at each layer to assign each token to a sparse subset of experts. In this work, we observe that forget data often activates a small subset of experts disproportionately, while these experts may receive much weaker activation from retain data. This forget–retain routing mismatch can leave forget-critical experts under-regularized during unlearning. To address this, we propose **TRACE**, Targeted Routing-Aware Calibration of Experts, for MoE unlearning. TRACE first detects forget-critical experts from offline activation statistics, and then calibrates retain regularization by reweighting token-level retain losses so that each selected expert's retain-side activation frequency better matches its forget-side counterpart. Experiments on WMDP and MUSE-BOOKS across multiple MoE LLMs show that TRACE consistently improves the forget-utility trade-off, yielding a 9% relative utility improvement over the strongest baseline under comparable forgetting quality and the best performance on three out of four MUSE-BOOKS metrics.

1 Introduction

Large language models (LLMs) have achieved remarkable success and are being rapidly deployed in various applications (Yang et al., 2025a; DeepSeek-AI et al., 2025; OpenAI et al., 2024). However, LLMs may memorize harmful (Li et al., 2024), copyrighted (Eldan and Russinovich, 2023), private (Jin et al., 2024; Maini et al., 2024), or other undesired information from web-scale pre-training corpora, making selective data removal increasingly important under data protection regulations, such as the European Union's General Data Protection Regulation (GDPR) (Voigt and Von dem Bussche, 2017) and the California Consumer Privacy Act (CCPA) (Bukaty, 2019). Since retraining LLMs from scratch is computationally infeasible, machine unlearning has emerged as a promising paradigm, which aims to eliminate the influence of specific samples while preserving model utility, thereby supporting trustworthy and compliant AI systems (Liu et al., 2024b; Geng et al., 2025).

Current unlearning methods typically maximize the loss on the data to be removed together with regularization on auxiliary retain data to balance forgetting effectiveness and general utility preservation (Yao et al., 2024; Zhang et al., 2024; Liu et al., 2022; Fan et al., 2025). However, most of these methods have been designed and evaluated on dense Transformer models (Vaswani et al., 2017), with limited consideration of Mixture-of-Experts (MoE) LLMs (Mu and Lin, 2026; DeepSeek-AI et al., 2025; Jiang et al., 2024; Lepikhin et al., 2020; Fedus et al., 2022). Unlike dense models that uniformly engage all parameters, MoE LLMs use routers to assign each token to a sparse subset of experts. This sparse, input-dependent computation calls for a closer examination of unlearning beyond the dense-model setting.

To this end, a recent study has begun to exploit this structure through expert-targeted unlearning, such as selecting the expert with the highest average affinity score with respect to the forget data (Zhuang et al., 2025). However, relying on a single selected expert provides insufficient unlearning capacity when forget-related knowledge is distributed across multiple experts. Indeed, widely adopted MoE models, including both DeepSeek and Qwen series, consistently exhibit distributed forget-data activations over multiple experts, as shown in Figure 1. More importantly, we observe that experts highly activated by the forget data often receive relatively low activation from retain data, especially when the retain corpus is generic and distributionally distant from the forget-data distribu-

tion. We refer to this phenomenon as forget-retain routing mismatch. This mismatch matters because retain data regularizes an expert only when retain tokens are routed to that expert. Our expert-level gradient decomposition shows that the retain gradient received by an expert is implicitly scaled by how often retain tokens activate that expert. Thus, a forget-critical expert with high forget activation but low retain activation is exposed to strong forgetting pressure while receiving weak retain protection. This creates an expert-level regularization mismatch that cannot be fully resolved by simply increasing the global retain coefficient.

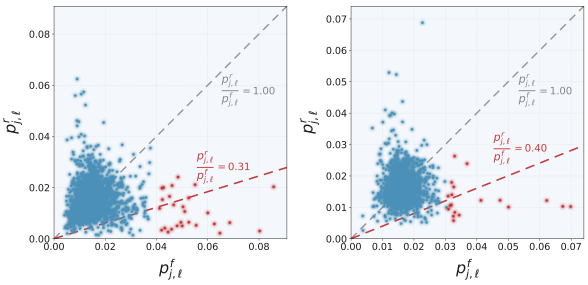
To address this challenge, we propose TRACE, **T**argeted **R**outing-**A**ware **C**alibration of **E**xperts, for MoE unlearning (Figure 2). TRACE first identifies forget-critical experts using routing activation statistics, and then reweights token-level retain losses to better match the retain-side and forget-side activation frequencies over the selected experts. By calibrating retain regularization at the expert level, TRACE avoids uniformly increasing the retain coefficient and instead redistributes retain supervision toward the experts most exposed to the forgetting objective.

Our contributions are summarized as follows:

- We identify a forget–retain routing mismatch in MoE unlearning and show that it induces expert-level regularization mismatch that cannot be fully corrected by a single global retain coefficient.
- We propose TRACE, a two-stage framework that combines forget-critical expert selection with routing-aware retain reweighting.
- We demonstrate on WMDP and MUSE-BOOKS across multiple MoE LLMs that TRACE improves the forget-utility trade-off over gradient-based and expert-selection baselines. Specifically, on the WMDP benchmark, TRACE yields a 9% relative improvement in utility over strongest baseline while maintaining a comparable forget quality. On MUSE-BOOKS, TRACE outperforms the baselines in VerbMem, forget-side KnowMem, and retain-side KnowMem scores.

2 Related Work

LLM unlearning. A growing line of work has investigated how to formulate, evaluate, and improve unlearning performance for LLMs (Geng et al., 2025; Liu et al., 2024b; Zhang et al., 2024; Fan et al., 2025; Jia et al., 2025; Lu et al., 2022; Liu et al., 2024a; Wang et al., 2023; Fang et al., 2026; Barbulescu and Triantafillou, 2024; Chen and Yang, 2023). Recent benchmarks instantiate LLM unlearning under different removal targets, including privacy information (Maini et al., 2024; Jin et al., 2024), copyrighted or memorized content (Shi et al., 2024; Eldan and Russinovich, 2023), and hazardous domain knowledge (Li et al., 2024). These benchmarks evaluate not only unlearning effectiveness, but also utility preservation. Existing methods typically address these objectives through different mechanisms, including gradient-based optimization (Zhang et al., 2024; Fan et al., 2025; Yu et al., 2023; Zhang et al., 2023; Rafailov et al., 2024), localization-informed editing (Jia et al., 2025; Fang et al., 2026; Patil et al., 2023; Meng et al., 2023; Wu et al., 2023), and input-based methods (Liu et al., 2024a; Madaan et al., 2023; Pawelczyk et al., 2024; Muresanu et al., 2025). Specifically, gradient-based methods directly optimize model parameters with forget and retain objectives, localization-informed methods first identify model components associated with target knowledge and then unlearn, while input-based methods modify the model behavior through the input context. However, most of them are designed and evaluated on dense transformer models, without considering sparse MoE LLMs.



(a) DeepSeek-V2-Lite-Chat. (b) Qwen1.5-MoE-A2.7B-Chat.

Figure 1: Routing-induced scaling mismatch in MoE unlearning. Visualization of expert activations across different MoE models on the forget dataset WMDP and the retain dataset WikiText, with forget-data activation frequency on the x-axis and retain-data activation frequency on the y-axis. Definitions of $p^f_{j,l}$ and $p^r_{j,l}$ are given in (3).

MoE LLMs and MoE unlearning. MoE LLMs have become an important architecture for scaling large language models by increasing model capacity without a proportional increase in infer-

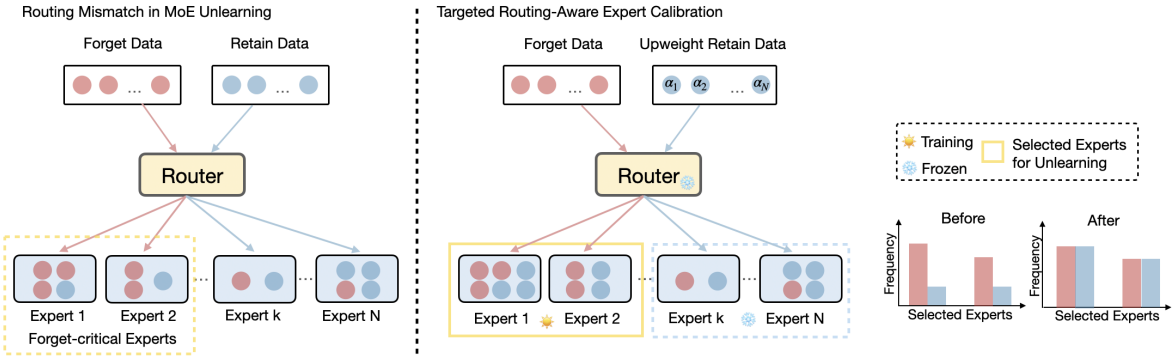


Figure 2: Motivation of TRACE. Left: In standard MoE unlearning, forget and retain data may activate different experts, leading to routing-induced scaling mismatch. Right: TRACE addresses this issue by selecting forget-critical experts for unlearning and reweighting retain tokens to align retain-side activation frequencies with forget-side activation frequencies over the selected experts.

ence cost (DeepSeek-AI et al., 2024, 2025; Du et al., 2022; Jiang et al., 2024; Fedus et al., 2022; Lepikhin et al., 2020). Unlike dense models that activate the same set of parameters for every input, MoE LLMs use a router to assign each token to a sparse subset of experts. Recent work has begun to exploit the modular structure of MoE LLMs for unlearning. SEUF (Zhuang et al., 2025) selects the expert with the largest affinity score to the forget data and updates the selected expert together with its corresponding router. ESFT (Wang et al., 2024), originally proposed for expert-specialized fine-tuning, selects top-ranked experts by cumulative affinity per layer and has also been adopted as an expert-selection baseline for MoE unlearning. Our work also explores expert-level selection for MoE unlearning by identifying forget-critical experts from routing activation statistics, but further shows that how retain regularization reaches selected forget-critical experts is equally critical.

3 Analysis of Routing-Induced Scaling Mismatch

In this section, we analyze why retain regularization can be uneven across experts in MoE unlearning. Let $\mathcal{D}_f$ and $\mathcal{D}_r$ denote the forget and retain datasets, respectively. A broad class of gradient-based unlearning methods optimizes a forget objective regularized by a retain loss:

$$\mathcal{L}(\boldsymbol{\theta}) = -\mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f) + \lambda \mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r),$$

where $\lambda > 0$ trades off forgetting the targets in $\mathcal{D}_f$ against preserving the model utility in $\mathcal{D}_r$.

We focus on the commonly used gradient-difference objective (GradDiff) as a representative example. Let N_f and N_r be the total numbers of predicted tokens in $\mathcal{D}_f$ and $\mathcal{D}_r$, respectively. After flattening all token prediction positions, the objective can be written as

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{N_f}\sum_{i=1}^{N_f} \ell(x^f_{i+1}; \boldsymbol{\theta}, \boldsymbol{x}^f_{\leq i}) + \lambda \frac{1}{N_r}\sum_{i=1}^{N_r} \ell(x^r_{i+1}; \boldsymbol{\theta}, \boldsymbol{x}^r_{\leq i}), \quad (1)$$

where $\boldsymbol{x}_{\leq i}$ denotes the prefix context used to predict token x_{i+1}. While we focus on the analysis for GradDiff in the main paper, the same routing effect arises in many other retain-regularized unlearning objectives such as NPO as shown in Section A.

3.1 Implicit Per-Expert Scaling in GradDiff

In this subsection, we show how the standard token-averaged retain loss induces different effective regularization strengths across experts. Consider an MoE model with top-K routing. Let $e_{j,\ell}$ denote the j-th expert of layer ℓ, and let $\mathcal{S}^f_i$ and $\mathcal{S}^r_i$ denote the sets of experts selected for the i-th forget and retain positions, respectively.

We decompose the GradDiff gradient according to expert routing. Consider the gradient received by the parameters $\boldsymbol{\theta}_{j,\ell}$ of expert $e_{j,\ell}$. Since an expert is updated only by tokens routed to it, the gradient of Eq. (1) with respect to $\theta_{j,\ell}$ is

$$\nabla_{\boldsymbol{\theta}_{j,\ell}} \mathcal{L}(\boldsymbol{\theta}) = -\frac{\sum_{i=1}^{N_f} \mathbf{1}\left[e_{j,\ell} \in \mathcal{S}^f_i\right] \nabla_{\boldsymbol{\theta}_{j,\ell}} \ell^f_i(\boldsymbol{\theta})}{N_f} + \lambda \frac{\sum_{i=1}^{N_r} \mathbf{1}[e_{j,\ell} \in \mathcal{S}^r_i] \nabla_{\boldsymbol{\theta}_{j,\ell}} \ell^r_i(\boldsymbol{\theta})}{N_r}, \quad (2)$$

where $\ell_i^f(\boldsymbol{\theta}) = \ell(x_{i+1}^f; \boldsymbol{\theta}, \boldsymbol{x}_{\leq i}^f)$ and $\ell_i^r(\boldsymbol{\theta}) = \ell(x_{i+1}^r; \boldsymbol{\theta}, \boldsymbol{x}_{\leq i}^r)$. The function $\mathbf{1}\left[e_{j,\ell} \in \mathcal{S}_i^f\right]$ indicates whether the expert $e_{j,l}$ is selected by the router for the i-th token in the forget data.

To expose the implicit scaling effect, we define the forget and retain activation frequencies of expert $e_{j,\ell}$ as

$$p_{j,\ell}^f = \frac{\sum_{i=1}^{N_f} \mathbf{1}\left[e_{j,\ell} \in \mathcal{S}_i^f\right]}{N_f}$$
$$p_{j,\ell}^r = \frac{\sum_{i=1}^{N_r} \mathbf{1}[e_{j,\ell} \in \mathcal{S}_i^r]}{N_r}. \tag{3}$$

For experts with nonzero forget and retain activations, Eq. (2) can be rewritten as

$$\nabla_{\boldsymbol{\theta}_{j,\ell}} \mathcal{L}(\boldsymbol{\theta}) = -\, p_{j,\ell}^f \underbrace{\frac{\sum_{i=1}^{N_f} \mathbf{1}\left[e_{j,\ell} \in \mathcal{S}_i^f\right] \nabla_{\boldsymbol{\theta}_{j,\ell}} \ell_i^f(\boldsymbol{\theta})}{\sum_{i=1}^{N_f} \mathbf{1}\left[e_{j,\ell} \in \mathcal{S}_i^f\right]}}_{\bar{g}_{j,\ell}^f} + \lambda p_{j,\ell}^r \underbrace{\frac{\sum_{i=1}^{N_r} \mathbf{1}[e_{j,\ell} \in \mathcal{S}_i^r] \nabla_{\boldsymbol{\theta}_{j,\ell}} \ell_i^r(\boldsymbol{\theta})}{\sum_{i=1}^{N_r} \mathbf{1}[e_{j,\ell} \in \mathcal{S}_i^r]}}_{\bar{g}_{j,\ell}^r}. \tag{4}$$

Here, $\bar{g}_{j,\ell}^f$ and $\bar{g}_{j,\ell}^r$ are the average forget and retain gradients contributed by tokens routed to expert $e_{j,\ell}$, respectively.

This expression shows that the gradient received by each expert is affected by two factors: an activation frequency and an average gradient over the tokens routed to that expert. In particular, the average forget gradient for $e_{j,l}$ is scaled by $p_{j,l}^f$, while the average retain gradient is scaled by $p_{j,l}^r$. Thus, even if two experts have comparable conditional retain gradients, the expert that is rarely activated by retain data receives much weaker effective retain regularization.

3.2 From Scaling Mismatch to Routing-Aware Calibration

Eq. (4) reveals two implications for MoE unlearning. First, the forget-side activation frequency $p_{j,\ell}^f$ provides a natural signal for identifying forget-critical experts: experts with larger $p_{j,\ell}^f$ receive a larger fraction of the forget gradient and are therefore more directly involved in processing the forget data. This motivates focusing unlearning updates on experts that are disproportionately activated by the forget data, rather than perturbing all experts uniformly.

Second, selecting forget-critical experts alone is insufficient. For a selected expert $e_{j,\ell}$, the effective retain regularization is scaled by its retain-side activation frequency $p_{j,\ell}^r$. Thus, an expert with large $p_{j,\ell}^f$ but small $p_{j,\ell}^r$ is exposed to strong forgetting pressure while receiving weak retain protection. This routing-induced scaling mismatch cannot be fully corrected by a single global retain coefficient λ: increasing λ may over-regularize experts already well covered by retain data while still failing to provide targeted protection to under-covered forget-critical experts. Therefore, effective MoE unlearning requires both forget-critical expert selection and routing-aware retain calibration.

4 TRACE: Targeted Routing-Aware Calibration of Experts

Guided by the two implications above, we propose TRACE, a targeted routing-aware expert calibration framework for MoE unlearning. The goal is to focus unlearning on experts most associated with the forget data while providing these experts with sufficient retain-side regularization. To achieve this, TRACE consists of two stages. First, it identifies forget-critical experts using routing activation statistics. Second, it calibrates retain regularization by reweighting token-level retain losses.

4.1 Forget-critical Expert Detection

We first identify experts that are disproportionately activated by the forget data. For each expert $e_{j,\ell}$, we compute its forget-side activation frequency $p_{j,\ell}^f$ using Eq. (3). We then select a set of forget-critical experts whose forget activation frequencies are unusually high. Specifically, we use the interquartile range (IQR) rule (Wan et al., 2014) to define an adaptive activation threshold. Let $Q_1(\cdot)$ and $Q_3(\cdot)$ denote the first and third quartiles over the collection of expert activation frequencies $\{p_{j,\ell}^f\}_{j,\ell}$, and $\mathrm{IQR}(z) = Q_3(z) - Q_1(z)$. We define the forget-critical threshold as

$$\tau_f = Q_3\left(p_{j,\ell}^f\right) + \lambda_f \,\mathrm{IQR}\left(p_{j,\ell}^f\right), \tag{5}$$

where λ_f controls the strictness of outlier selection. Following the classical outlier detection rule, we set $\lambda_f = 1.5$ in all experiments. The selected forget-critical expert set is

$$\mathcal{E} = \{(j,l) : p_{j,l}^f \geq \tau_f\}.$$

This adaptive rule allows TRACE to select multiple forget-relevant experts when the forget data acti-

vates a distributed expert subset, while avoiding unnecessary updates to experts weakly associated with the forget data.

4.2 Routing-Aware Retain Reweighting

After selecting $\mathcal{E}$, we compensate for the scaling mismatch in Eq. (4) by introducing a re-scaled retain loss. In practice, directly manipulating per-expert losses or gradients is inconvenient because the training objective is computed at the token level and each token may activate multiple experts. Hence, we upweight retain tokens that activate under-covered forget-critical experts, so that the weighted retain-side activation frequency of each selected expert better matches its forget activation frequency.

Specifically, we define the retain routing matrix $\boldsymbol{M} \in \{0,1\}^{N_r \times |\mathcal{E}|}$ as

$$\boldsymbol{M}_{i,k} = \mathbb{1}\{\text{the } k\text{-th expert of } \mathcal{E} \text{ is in } \mathcal{S}_i^r\}\,, \quad (6)$$

where $\boldsymbol{M}_{i,k} = 1$ indicates that the i-th retain position activates the k-th selected expert.

Next, we assign weights to the retain tokens to calibrate the regularization levels of forget-critical experts. Let $\boldsymbol{\alpha} \in \mathbb{R}^{N_r}$ denote the token-level retain weights. The weighted retain activation frequency of the k-th selected expert is given by $(\boldsymbol{M}_{\cdot,k})^\top \boldsymbol{\alpha}/N_r = \sum_{i=1}^{N_r} \alpha_i \mathbb{1}(M_{i,k} = 1)/N_r$. To align it with the forget activation frequency vector of the selected experts, we define $\boldsymbol{p}_{\mathcal{E}}^f = \{p_{j,l}^f\}_{(j,l)\in\mathcal{E}} \in \mathbb{R}^{|\mathcal{E}|}$. We solve the following constrained optimization problem:

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha} \geq \mathbf{1}_{N_r}}{\operatorname{argmin}} \left\| \frac{\boldsymbol{M}^\top \boldsymbol{\alpha}}{N_r} - \boldsymbol{p}_{\mathcal{E}}^f \right\|_2^2 + \gamma \left\| \boldsymbol{\alpha} - \mathbf{1}_{N_r} \right\|_2^2 , \quad (7)$$

where $\gamma > 0$ controls the regularization strength towards uniform weights, $\mathbf{1}_{N_r}$ denote a vector of one with length N_r, and the constraint is applied element-wise. We initialize $\boldsymbol{\alpha}$ as the all-one vector $\mathbf{1}$ and solve Eq. (7) using projected gradient descent to obtain $\hat{\boldsymbol{\alpha}}$. Since the selected experts suffer from insufficient retain-side activation, the constraint $\boldsymbol{\alpha} \geq \mathbf{1}$ ensures that retain tokens are only upweighted, avoiding additional downweighting of retain supervision.

Putting things together, the final objective of TRACE is

$$\min_{\boldsymbol{\theta}} \; -\frac{1}{N_f} \sum_{i=1}^{N_f} \ell_i^f(\boldsymbol{\theta}) + \lambda \frac{1}{N_r} \sum_{i=1}^{N_r} \hat{\alpha}_i \ell_i^r(\boldsymbol{\theta}) \quad (8)$$
$$\text{subject to } \; \boldsymbol{\theta}_{j,l} = \boldsymbol{\theta}_{j,l}^{(\text{ptr})} \;\; \forall (j,l) \notin \mathcal{E},$$

Algorithm 1 TRACE: Targeted Routing-Aware Expert Calibration

Require: MoE parameters $\boldsymbol{\theta}^{(\text{ptr})}$, forget data $\mathcal{D}_f$, retain data $\mathcal{D}_r$, offline forget subset $\widetilde{\mathcal{D}}_f \subset \mathcal{D}_f$, λ_f, λ, γ
Ensure: Unlearned parameters $\boldsymbol{\theta}$
Offline expert selection
Compute forget activation frequencies $p_{j,\ell}^f$ on $\widetilde{\mathcal{D}}_f$.
Compute $\tau_f \leftarrow Q_3(p_{j,\ell}^f) + \lambda_f \text{IQR}(p_{j,\ell}^f)$.
Select forget-critical experts $\mathcal{E} \leftarrow \{(j,\ell) : p_{j,\ell}^f \geq \tau_f\}$.
Unlearning training
Initialize $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(\text{ptr})}$.
for each training step **do**
  Sample batch $\mathcal{B}_f \subset \mathcal{D}_f$ and batch $\mathcal{B}_r \subset \mathcal{D}_r$.
  Record activation frequencies during the forward pass on $\mathcal{B}_f$ and $\mathcal{B}_r$, then construct $\boldsymbol{M}$ and $\boldsymbol{p}_{\mathcal{E}}^f$.
  Obtain $\hat{\boldsymbol{\alpha}}$ by solving (7).
  Update only experts in $\mathcal{E}$ by minimizing objective (8).
end for
return $\boldsymbol{\theta}$

where $\boldsymbol{\theta}^{(\text{ptr})}$ denotes the pre-trained model parameters, which is fixed during unlearning. The overall procedure is summarized in Algorithm 1.

5 Experiments

We evaluate TRACE on WMDP (Li et al., 2024) and MUSE-BOOKS (Shi et al., 2024) across multiple MoE architectures. We assess unlearning performance from both forgetting effectiveness and retained utility.

5.1 Experimental Setups

Datasets. To demonstrate the effectiveness of our method, we conduct experiments on two benchmarks: WMDP and MUSE-BOOKS. WMDP aims to remove hazardous knowledge, including content related to Cybersecurity (Cyber) and Biosecurity (Bio), with WikiText (Merity et al., 2016) used as the retain dataset. MUSE-BOOKS comprises text extracted from the Harry Potter series, focusing on removing the copyrighted content, while the corresponding retain dataset is sampled from Harry Potter Fandom Wiki.

Metrics. We evaluate each method in terms of both forgetting quality and retaining utility on two benchmarks. For WMDP, we report answer accuracy on WMDP-Cyber and WMDP-Bio as the forgetting metric, where better forgetting corresponds to accuracy closer to the random-guessing level of 25%. We report answer accuracy on MMLU (Hendrycks et al., 2021) to assess retaining utility. For MUSE-BOOKS, we follow the benchmark protocol and evaluate forgetting quality

using three metrics: VerbMem, KnowMem, and PrivLeak. VerbMem measures verbatim memorization of the forget set, while KnowMem measures knowledge memorization of the forgotten content. Both memorization metrics are evaluated using ROUGE-L. PrivLeak measures membership-inference leakage. For retained utility, we report KnowMem on the retain set, also measured by ROUGE-L.

MoE LLMs. We evaluate different unlearning methods on three MoE LLMs: Qwen1.5-MoE-A2.7B-Chat (Yang et al., 2024), DeepSeek-V2-Lite (DeepSeek-AI et al., 2024), and Qwen3-30B-A3B (Yang et al., 2025b). Following (Zhuang et al., 2025), we use Qwen1.5-MoE-A2.7B-Chat and DeepSeek-V2-Lite-Chat for unlearning on WMDP. For MUSE-BOOKS, we use Qwen3-30B-A3B. In prior dense-model settings, models are often first trained on MUSE-BOOKS before unlearning to ensure that the target content is present in the model. In our case, we directly evaluate Qwen3-30B-A3B and find that it already exhibits a certain degree of both verbatim and knowledge memorization of the target content. We therefore avoid additional training on MUSE-BOOKS, since training large MoE models can be unstable and highly sensitive to hyperparameter choices, often risking training collapse (Zoph et al., 2022; Jiang et al., 2024).

Baselines. We compare our method with several representative baselines. For standard gradient-based unlearning, we apply GA, GradDiff (Maini et al., 2024), and Negative Preference Optimization (NPO) (Zhang et al., 2024). In addition, we compare with expert-selection-based methods, including SEUF (Zhuang et al., 2025) and ESFT (Wang et al., 2024). Both methods identify forget-relevant experts and restrict unlearning updates to the selected experts. Specifically, SEUF selects the single expert, together with its corresponding router, that has the maximum average affinity score on offline forget data. In contrast, ESFT ranks experts by their average affinity scores within each layer and selects top-ranked experts per layer until their cumulative affinity exceeds a predefined threshold p.

5.2 Overall Performance

Effectiveness of TRACE in Forgetting and Preserving Utility. We report the overall unlearning performance on WMDP in Table 1 and MUSE-BOOKS in Table 2. Across both MoE LLMs on WMDP, TRACE achieves the best utility among all unlearning methods, with the highest MMLU scores on both Qwen1.5-MoE-A2.7B-Chat and DeepSeek-V2-Lite-Chat. Meanwhile, TRACE maintains competitive forgetting accuracy on both Cyber and Bio, showing that the improved utility is not obtained by simply sacrificing forgetting effectiveness. Compared with standard unlearning baselines such as GA, GradDiff, and NPO, TRACE substantially improves general utility after unlearning, which suggests that restricting updates to forget-critical experts and calibrating their retain-side regularization can better preserve non-forget knowledge. Compared with expert-selection-based methods such as SEUF and ESFT, TRACE further improves the forget-utility trade-off. This indicates that selecting only one expert, as in SEUF, may limit the capacity for unlearning. To achieve sufficient forgetting under this constraint, SEUF may need more update steps or a larger learning rate, which can degrade model utility. Moreover, selecting experts alone, as in ESFT, remains suboptimal without routing-aware retain reweighting.

On MUSE-BOOKS, TRACE also demonstrates strong forgetting performance while achieving the best retain-side KnowMem score. In particular, TRACE reduces forget-side KnowMem to 10.73, outperforming all baselines, and achieves the highest utility KnowMem of 65.97, even surpassing the original model. This is because the original model is evaluated without any retain-side training in our setting, whereas TRACE performs unlearning with retain regularization. As a result, TRACE not only preserves the model's existing retain-side knowledge, but also acquires additional knowledge in the retain dataset that was not fully captured by the original model. Moreover, although ESFT obtains the lowest PrivLeak score, it retains lower utility and weaker forget-side KnowMem than TRACE. Overall, these results show that TRACE consistently provides a favorable balance between removing forget-related knowledge and preserving model utility across different MoE models and unlearning benchmarks.

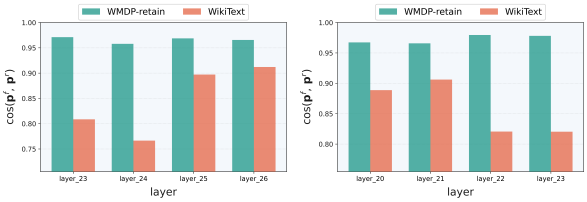
Comparison with Domain-Specific Retain Data. We further show that, with routing-aware calibration, TRACE using generic retain data can achieve performance comparable to GradDiff using domain-specific retain data. This further highlights the practical value of TRACE in reducing the cost of collecting high-quality retain data. Specifi-

Table 1: Performance comparison of unlearning methods on the WMDP benchmark for Qwen1.5-MoE-A2.7B-Chat and DeepSeek-V2-Lite-Chat. ↑ and ↓ indicate that higher and lower values are better, respectively. Bold values denote the best results among unlearning methods.

Method	Qwen1.5-MoE-A2.7B-Chat				DeepSeek-V2-Lite-Chat			
	Forgetting Quality			Utility	Forgetting Quality			Utility
	Acc-Cyber ↓	Acc-Bio ↓	Acc-Avg ↓	MMLU ↑	Acc-Cyber ↓	Acc-Bio ↓	Acc-Avg ↓	MMLU ↑
Origin	0.4243	0.6536	0.5390	0.5954	0.3946	0.5860	0.4903	0.5672
GA	0.2964	0.2647	0.2806	0.3217	0.2929	0.2702	0.2816	0.3307
GradDiff	0.2657	0.2467	**0.2562**	0.2295	0.2994	0.3267	0.3131	0.3722
NPO	0.2687	0.2474	0.2581	0.2481	0.2246	0.2531	**0.2389**	0.2466
SEUF	0.2545	0.2670	0.2608	0.4915	0.3050	0.2600	0.2825	0.4553
ESFT	0.2999	0.2655	0.2827	0.5319	0.2592	0.2396	0.2494	0.5099
TRACE	0.2748	0.2663	0.2706	**0.5819**	0.2647	0.2624	0.2636	**0.5541**

Table 2: Performance comparison of unlearning methods on the MUSE-BOOKS benchmark. ↓ / ↑ indicates lower / higher is better, → 0 indicates closer to zero is better, and bold values denote the best results among unlearning methods.

Method	Forgetting Quality			Utility
	VerbMem $\mathcal{D}_f$ ↓	KnowMem $\mathcal{D}_f$ ↓	PrivLeak → 0	KnowMem $\mathcal{D}_r$ ↑
Origin	15.92	45.82	0	62.48
GA	7.5	10.92	38.64	13.76
GradDiff	6.86	36.48	19.43	57.17
NPO	5.12	34.05	37.84	62.64
SEUF	0.32	11.44	11.16	55.54
ESFT	**0.00**	26.63	**10.39**	58.69
TRACE	**0.00**	**10.73**	20.72	**65.97**



(a) DeepSeek-V2-Lite-Chat. (b) Qwen1.5-MoE-A2.7B-Chat.

Figure 3: Routing similarity between the forget dataset and different retain datasets in the last four MoE layers. Compared with WikiText, the carefully curated WMDP-retain exhibits consistently higher routing similarity to the WMDP-forget, indicating that WMDP-retain follows routing paths more similar to WMDP-forget.

cally, we consider GradDiff*, which denotes GradDiff equipped with a curated WMDP retain dataset (WMDP-retain). This method can be viewed as a near-oracle setting, because the WMDP-retain is constructed from biology and cybersecurity domains, and is therefore more likely to follow routing paths similar to the WMDP forget dataset (WMDP-forget). To verify this, we compare the routing similarity between WMDP-forget and different retain datasets in the last four MoE layers in Figure 3. The WMDP-retain consistently exhibits higher routing similarity with the WMDP-forget than WikiText, confirming that domain-specific retain data better matches the routing path of the forget data. Even so, as shown in Figure 4, TRACE, which uses WikiText as retain data, achieves performance comparable to GradDiff* on both Qwen1.5-MoE-A2.7B-Chat and DeepSeek-V2-Lite-Chat.

5.3 Ablation Study

Ablation on Forget-Critical Expert Selection and Retain Reweighting. We present the ablation results of TRACE in Table 3 and Table 4. We compare three variants: the original GradDiff, GradDiff applied only to the selected forget-critical experts (TRACE-Select), and the full TRACE with routing-aware retain reweighting. On WMDP, selecting forget-critical experts already brings a substantial utility improvement over GradDiff. For example, on Qwen1.5-MoE-A2.7B-Chat, MMLU increases from 0.2295 to 0.5734, and on DeepSeek-V2-Lite-Chat, it increases from 0.3722 to 0.5335. By further applying retain reweighting, TRACE consistently improves utility while maintaining competitive forgetting performance. On WMDP, TRACE achieves the highest MMLU on both Qwen1.5-MoE-A2.7B-Chat and DeepSeek-V2-Lite-Chat. On MUSE-BOOKS, TRACE-Select substantially improves forgetting quality, but it slightly reduces utility KnowMem from 57.17 to 55.15. In contrast, TRACE improves utility KnowMem to 65.97 while maintaining com-

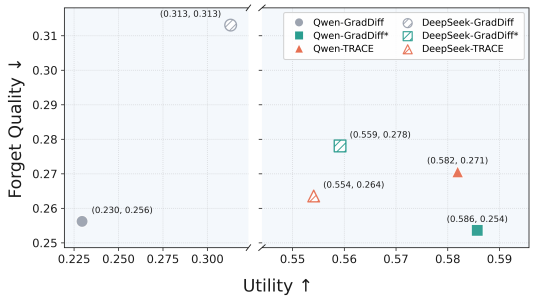


Figure 4: Comparison among GradDiff, GradDiff*, and TRACE on WMDP. GradDiff and TRACE use WikiText as generic retain data, while GradDiff* uses WMDP-retain. Forget Quality is the average of Cyber and Bio accuracy; Utility is MMLU accuracy.

Table 3: Ablation study of TRACE on the WMDP benchmark for Qwen1.5-MoE-A2.7B-Chat and DeepSeek-V2-Lite-Chat.

Method	Forgetting Quality			Retaining Utility
	Acc-Cyber ↓	Acc-Bio ↓	Acc-Avg ↓	MMLU ↑
Qwen1.5-MoE-A2.7B-Chat				
GradDiff	0.2657	0.2467	0.2562	0.2295
TRACE-Select	0.2813	0.2505	0.2659	0.5734
TRACE	0.2748	0.2663	0.2706	0.5819
DeepSeek-V2-Lite-Chat				
GradDiff	0.2994	0.3267	0.3131	0.3722
TRACE-Select	0.2567	0.2419	0.2493	0.5335
TRACE	0.2647	0.2624	0.2636	0.5541

parable forgetting quality. These results demonstrate that both components are necessary.

Table 4: Ablation study of TRACE on the MUSE-BOOKS benchmark for Qwen3-30B-A3B.

Method	Forgetting Quality			Utility
	VerbMem $\mathcal{D}_f$ ↓	KnowMem $\mathcal{D}_f$ ↓	PrivLeak → 0	KnowMem $\mathcal{D}_r$ ↑
GradDiff	6.86	36.48	19.43	57.17
TRACE-Select	0.00	7.43	-39.31	55.15
TRACE	0.00	10.73	20.72	65.97

5.4 Sensitivity Analysis of Hyperparameter γ

We analyze the sensitivity of TRACE to the hyperparameter γ in Eq. (7), which appears in the optimization problem for solving the retain-token reweighting coefficients. On MUSE-BOOKS, we vary γ over $\{0.1, 0.5, 1, 10\}$ and report the results in Figure 5. The stable results show that TRACE is robust to the choice of γ.

5.5 Efficiency Analysis

TRACE introduces an additional optimization step for solving the retain-token weights $\hat{\boldsymbol{\alpha}}$ in Eq. (7). In practice, this step adds only modest overhead.

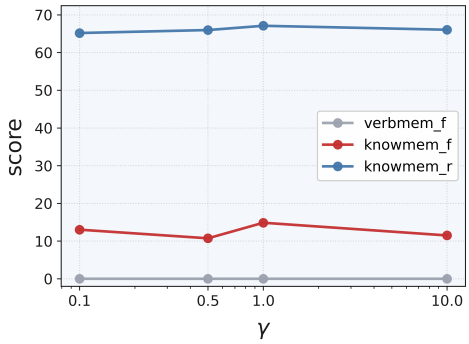


Figure 5: On MUSE-BOOKS, we vary γ over $\{0.1, 0.5, 1, 10\}$. The results show that TRACE is robust to the choice of γ.

For instance, on MUSE-BOOKS, one epoch takes 37.6 minutes without retain reweighting and 39.5 minutes with retain reweighting, corresponding to an additional 1.9 minutes or about 5.0% overhead. This indicates that routing-aware retain reweighting improves expert-level regularization with limited extra computation.

6 Conclusion

We study machine unlearning for MoE LLMs, a setting that remains less explored than unlearning in dense architectures. We identify a routing-induced scaling mismatch: forget-critical experts can be strongly activated by forget data but weakly activated by retain data, leading to insufficient retain-side regularization. To address this issue, we propose TRACE, a targeted routing-aware calibration framework that selects forget-critical experts and calibrates their retain regularization through token-level retain reweighting. Experiments on WMDP and MUSE-BOOKS across multiple MoE LLMs show that TRACE consistently improves the forget–utility trade-off over gradient-based and expert-selection baselines. These findings highlight the importance of expert-level routing information for effective and utility-preserving MoE unlearning.

Limitations

TRACE relies on routing activation statistics collected from forget data. When the forget set is small, the estimated routing statistics may be noisy and affect expert selection. Although this procedure is lightweight compared with full model retraining, it introduces an additional profiling step before unlearning. Future work could explore more efficient online or approximate estimation strategies for identifying forget-critical experts.

References

George-Octavian Barbulescu and Peter Triantafillou. 2024. To each (textual sequence) its own: Improving memorized-data unlearning in large language models. *Preprint*, arXiv:2405.03097.

Preston Bukaty. 2019. *The california consumer privacy act (ccpa): An implementation guide*. IT Governance Ltd.

Jiaao Chen and Diyi Yang. 2023. Unlearn what you want to forget: Efficient unlearning for llms. *Preprint*, arXiv:2310.20150.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and 138 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, and 8 others. 2022. Glam: Efficient scaling of language models with mixture-of-experts. *Preprint*, arXiv:2112.06905.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms. *Preprint*, arXiv:2310.02238.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. 2025. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *Preprint*, arXiv:2410.07163.

Ce Fang, Zhikun Zhang, Min Chen, Qing Liu, Lu Zhou, Zhe Liu, and Yunjun Gao. 2026. Kuda: Knowledge unlearning by deviating representation for large language models. *Preprint*, arXiv:2602.19275.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Preprint*, arXiv:2101.03961.

Jiahui Geng, Qing Li, Herbert Woisetschlaeger, Zongxiong Chen, Fengyu Cai, Yuxia Wang, Preslav Nakov, Hans-Arno Jacobsen, and Fakhri Karray. 2025. A comprehensive survey of machine unlearning techniques for large language models. *Preprint*, arXiv:2503.01854.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Jinghan Jia, Jiancheng Liu, Yihua Zhang, Parikshit Ram, Nathalie Baracaldo, and Sijia Liu. 2025. Wagle: Strategic weight attribution for effective and modular unlearning in large language models. *Preprint*, arXiv:2410.17509.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Rwku: Benchmarking real-world knowledge unlearning for large language models. *Preprint*, arXiv:2406.10890.

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. *Preprint*, arXiv:2006.16668.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, and 38 others. 2024. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *Preprint*, arXiv:2403.03218.

Bo Liu, Qiang Liu, and Peter Stone. 2022. Continual learning and private unlearning. *Preprint*, arXiv:2203.12817.

Chris Yuhao Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. 2024a. Large language model unlearning via embedding-corrupted prompts. *Preprint*, arXiv:2406.07933.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. 2024b. Rethinking machine unlearning for large language models. *Preprint*, arXiv:2402.08787.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. *Preprint*, arXiv:2205.13636.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2023. Memory-assisted prompt editing to improve gpt-3 after deployment. *Preprint*, arXiv:2201.06009.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. 2024. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023. Locating and editing factual associations in gpt. *Preprint*, arXiv:2202.05262.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Siyuan Mu and Sen Lin. 2026. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. *Preprint*, arXiv:2503.07137.

Andrei I. Muresanu, Anvith Thudi, Michael R. Zhang, and Nicolas Papernot. 2025. Fast exact unlearning for in-context learning data for llms. *Preprint*, arXiv:2402.00751.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Vaidehi Patil, Peter Hase, and Mohit Bansal. 2023. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. *Preprint*, arXiv:2309.17410.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. 2024. In-context unlearning: Language models as few shot unlearners. *Preprint*, arXiv:2310.07579.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. 2024. Muse: Machine unlearning six-way evaluation for language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Preprint*, arXiv:1706.03762.

Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A practical guide, 1st ed., Cham: Springer International Publishing*, 10(3152676):10–5555.

Xiang Wan, Wenqian Wang, Jiming Liu, and Tiejun Tong. 2014. Estimating the sample mean and standard deviation from the sample size, median, range and/or interquartile range. *BMC medical research methodology*, 14(1):135.

Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. 2023. Kga: A general machine unlearning framework based on knowledge gap alignment. *Preprint*, arXiv:2305.06535.

Zihan Wang, Deli Chen, Damai Dai, Runxin Xu, Zhuoshu Li, and Y. Wu. 2024. Let the expert stick to his last: Expert-specialized fine-tuning for sparse architectural large language models. *Preprint*, arXiv:2407.01906.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023. Depn: Detecting and editing privacy neurons in pretrained language models. *Preprint*, arXiv:2310.20138.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025b. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024. Large language model unlearning. *Preprint*, arXiv:2310.10683.

Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. 2023. Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6032–6048, Toronto, Canada. Association for Computational Linguistics.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. 2023. Composing parameter-efficient modules with arithmetic operations. *Preprint*, arXiv:2306.14870.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024. Negative preference optimization: From catastrophic collapse to effective unlearning. *Preprint*, arXiv:2404.05868.

Haomin Zhuang, Yihua Zhang, Kehan Guo, Jinghan Jia, Gaowen Liu, Sijia Liu, and Xiangliang Zhang. 2025. SEUF: Is unlearning one expert enough for mixture-of-experts LLMs? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8664–8678, Vienna, Austria. Association for Computational Linguistics.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. St-moe: Designing stable and transferable sparse expert models. *Preprint*, arXiv:2202.08906.

Appendix

A Generality of TRACE

To verify the generality of TRACE, we further evaluate it under the NPO unlearning objective with retain regularization. We replace the GA forget loss with NPO, while keeping the same forget-critical expert selection and routing-aware retain reweighting design.

Token-wise NPO formulation. Besides GradDiff, TRACE can also be instantiated with preference-based forget objectives such as NPO.

For each forget token prediction position i, we define the token-level log-ratio between the current model and the reference model as

$$r_i^f(\boldsymbol{\theta}) = \log \frac{p_{\boldsymbol{\theta}}(x_{i+1}^f \mid \boldsymbol{x}_{\leq i}^f)}{p_{\boldsymbol{\theta}^{(\mathrm{ptr})}}(x_{i+1}^f \mid \boldsymbol{x}_{\leq i}^f)}. \tag{9}$$

Equivalently, using the next-token negative log-likelihood $\ell_i^f(\boldsymbol{\theta}) = -\log p_{\boldsymbol{\theta}}(x_{i+1}^f \mid \boldsymbol{x}_{\leq i}^f)$, we have

$$r_i^f(\boldsymbol{\theta}) = -\ell_i^f(\boldsymbol{\theta}) + \ell_i^f(\boldsymbol{\theta}^{(\mathrm{ptr})}). \tag{10}$$

The token-wise NPO forget loss is then defined as

$$\phi_i^{(\mathrm{NPO})}(\boldsymbol{\theta}) = -\frac{2}{\beta} \log \sigma\left(-\beta r_i^f(\boldsymbol{\theta})\right), \tag{11}$$

where $\beta > 0$ is the NPO temperature coefficient and $\sigma(\cdot)$ denotes the sigmoid function. This loss encourages the current model to assign lower probability to forget tokens than the reference model.

Combining token-wise NPO with retain regularization gives

$$\mathcal{L}^{(\mathrm{NPO})}(\boldsymbol{\theta}) = \frac{1}{N_f} \sum_{i=1}^{N_f} \phi_i^{(\mathrm{NPO})}(\boldsymbol{\theta}) + \lambda \frac{1}{N_r} \sum_{i=1}^{N_r} \ell_i^r(\boldsymbol{\theta}). \tag{12}$$

Experiments. As reported in Table 5, TRACE remains effective, demonstrating that our method is not restricted to a specific forget loss but can generalize across different unlearning objectives.

Table 5: TRACE under the NPO objective on WMDP. TRACE-Select applies NPO only to selected forget-critical experts, while TRACE further adds routing-aware retain reweighting.

Method	Forgetting Quality			Retaining Utility
	Acc-Cyber ↓	Acc-Bio ↓	Acc-Avg ↓	MMLU ↑
Qwen1.5-MoE-A2.7B-Chat				
NPO	0.2687	0.2474	0.2581	0.2481
TRACE-Select	0.2974	0.2655	0.2814	0.5769
TRACE	0.2803	0.2694	0.2748	0.5837
DeepSeek-V2-Lite-Chat				
NPO	0.2246	0.2531	0.2389	0.2466
TRACE-Select	0.2657	0.2467	0.2562	0.5402
TRACE	0.2899	0.2647	0.2773	0.5575

B Experiment Setup

Experimental Details. We use AdamW as the optimizer for all experiments, with batch size fixed to 16. For WMDP experiments on Qwen1.5-MoE-A2.7B-Chat and DeepSeek-V2-Lite-Chat, GA, GradDiff and NPO use a learning rate of $5e^{-5}$ with 100 steps, and NPO uses $\beta = 0.01$. Our method and ESFT also use a learning rate of $5e^{-5}$ with 300 steps and 200 steps, respectively, while SEUF follows its original setting with a learning rate of $2e^{-4}$ with 100 steps. For MUSE-BOOKS experiments on Qwen-30B-A3B, GA, GradDiff, and NPO are trained with LoRA on all linear modules, using a learning rate of $1e^{-5}$ with 1 epoch and LoRA rank $r = 16$, and NPO uses $\beta = 0.1$. SEUF and ESFT use learning rates of $8e^{-5}$ and $1e^{-4}$ with 1 epoch, respectively. Our method uses a learning rate of $5e^{-5}$ with 2 epoch. For ESFT, the cumulative affinity threshold p is set to 0.1, and for our method, the IQR coefficient λ_f is set to 1.5 across all experiments. The retain regularization coefficient λ is set to 1 for all applicable methods. All experiments are conducted on machines equipped with either 2 NVIDIA A100 GPUs or 2 NVIDIA H20 GPUs.

MoE LLMs Details. Qwen1.5-MoE-A2.7B-Chat contains 14.3B total parameters, with 2.7B activated during inference; DeepSeek-V2-Lite contains 16B total parameters, with 2.4B activated; and Qwen3-30B-A3B contains 30B total parameters, with 3B activated.