OBLIVIONIS: A Lightweight Learning and Unlearning Framework for Federated Large Language Models

Fuyao Zhang[1], Xinyu Yan[1], Tiantong Wu[1], Wenjie Li[1,2], Tianxiang Chen[1], Yang Cao[3], Ran Yan[1], Longtao Huang[4], Wei Yang Bryan Lim[1*], Qiang Yang[5]

[1] Nanyang Technological University
[2] Hebei Normal University
[3] Institute of Science Tokyo
[4] Alibaba Group
[5] Hong Kong Polytechnic University

fuyao.zhang@ntu.edu.sg, xinyu039@e.ntu.edu.sg, tiantong.wu@ntu.edu.sg, liwenjie@hebtu.edu.cn, n2409902f@e.ntu.edu.sg, cao@c.titech.ac.jp, ran.yan@ntu.edu.sg, kaiyang.hlt@alibaba-inc.com, bryan.limwy@ntu.edu.sg, profqiang.yang@polyu.edu.hk

Abstract

Large Language Models (LLMs) increasingly leverage Federated Learning (FL) to utilize private, task-specific datasets for fine-tuning while preserving data privacy. However, while federated LLM frameworks effectively enable collaborative training without raw data sharing, they critically lack built-in mechanisms for regulatory compliance like GDPR's *right to be forgotten*. Integrating private data heightens concerns over data quality and long-term governance, yet existing distributed training frameworks offer no principled way to selectively remove specific client contributions post-training. Due to distributed data silos, stringent privacy constraints, and the intricacies of interdependent model aggregation, federated LLM unlearning is significantly more complex than centralized LLM unlearning. To address this gap, we introduce **OBLIVIONIS**, a lightweight learning and unlearning framework that enables clients to selectively remove specific private data during federated LLM training, enhancing trustworthiness and regulatory compliance. By unifying FL and unlearning as a dual optimization objective, we incorporate 6 FL and 5 unlearning algorithms for comprehensive evaluation and comparative analysis, establishing a robust pipeline for federated LLM unlearning. Extensive experiments demonstrate that **OBLIVIONIS** outperforms local training, achieving a robust balance between forgetting efficacy and model utility, with cross-algorithm comparisons providing clear directions for future LLM development.

Code — https://github.com/fyzhang1/Oblivionis
Extension Verison — https://arxiv.org/abs/2508.08875

1 Introduction

Large Language Models (LLMs), driven by the Transformer architecture (Vaswani et al. 2017), have transformed Natural Language Processing and diverse fields (Achiam et al. 2023; Touvron et al. 2023). By efficiently learning complex patterns from vast datasets, they enable advanced tasks such as text generation, translation, and question-answering (Wei et al. 2022; Webb, Holyoak, and Lu 2023; Imani, Du, and Shrivastava 2023). Typically, the increase in the quantity and quality of data samples leads to stronger generalization capabilities and higher task accuracy. In particular, LLM fine-tuning relies on limited task-specific private data. Such data cannot be used for centralized fine-tuning as it often involves personal information or holds significant economic value, as seen in domains like the medical and financial (Thirunavukarasu et al. 2023; Wu et al. 2023).

In this context, Federated Learning (FL) (McMahan et al. 2017), as an emerging distributed machine learning

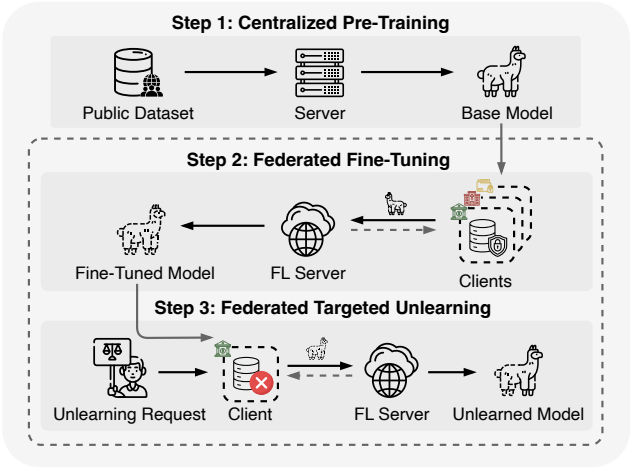

Figure 1: Illustration of the three-step LLM training process: (1) Pre-training the base model with public datasets on a centralized server; (2) Federated fine-tuning on the base model using private and sensitive task-specific data; (3) Federated targeted unlearning removes the influence of specific data upon client requests, addressing regulatory and ethical requirements. Areas enclosed by grey dashed boxes are our main contributions.

*Corresponding Author.

paradigm, becomes a highly anticipated trend in the development of LLM training because of its unique collaborative training mechanism and inherent privacy-preserving feature. Federated LLM (FedLLM) allows multiple clients to jointly fine-tune a global model without sharing their local private data. Specifically, Chen et al. (2023) first proposed a systematic research framework to explore the integration between LLM and FL. Fan et al. (2023) proposed an industrial-grade framework for FedLLM that addresses resource consumption and data privacy challenges, supporting efficient training and privacy-preserving mechanisms. Ye et al. (2024) proposed the OpenFedLLM framework for training LLM on decentralized private data, with federated instruction tuning, value alignment, and multiple FL algorithms.

Although FL offers a promising approach for the continuous evolution of LLMs, it still encounters significant challenges in practical applications. As depicted in Figure 1, the large number of participating FL clients and diverse data sources can lead to global models inadvertently learning low-quality knowledge, biased information, or outdated content from specific clients during federated fine-tuning (Wei, Haghtalab, and Steinhardt 2023; Min et al. 2023). Furthermore, as global data privacy regulations (e.g., the E.U.'s General Data Protection Regulation, GDPR) become increasingly sophisticated and public awareness of user data rights grows, the right to be forgotten and data deletion requests are gaining more importance (Rosen 2011; Pardau 2018). Thus, LLMs require not only the capability to acquire new knowledge, but also the ability to effectively remove specific data and its contribution to the model upon the removal request (Huu-Tien et al. 2024; Wang et al. 2024, 2025a). Preventing model retention of removed data is critical for maintaining user trust, ensuring regulatory compliance, and preserving model integrity.

Based on the above challenges and requirements, we aim to explore an innovative LLM training paradigm to effectively mitigate the influence of low-quality knowledge within the FL framework and empower the model to respond to data contribution removal requests. We propose that during the training process of FedLLM, when a client opts out of FL or its data contribution legally needs to be removed, the global model should be able to perform federated targeted unlearning. This process is designed to achieve three key objectives: **(1) Effectiveness**, selectively removing all influences of a client's local private data from the global model; **(2) Robustness**, ensuring the model maintains high utility on retained data; **(3) Lightweight Design**, enabling unlearning with minimal computational resources and model parameters. To achieve these goals, we propose **Oblivionis**, a lightweight FedLLM learning and unlearning framework that integrates federated fine-tuning and targeted unlearning, enabling robust LLM training while ensuring compliance with privacy regulations. In conclusion, our contributions are as follows:

- We propose **Oblivionis**, the first framework that integrates FL and targeted unlearning for LLMs, formulating them as a joint dual-objective optimization task to enable privacy-preserving training and compliance with GDPR's *right to be forgotten*.
- We consolidate diverse FL and unlearning benchmarks, training, and evaluation datasets into a user-friendly platform, facilitating standardized research for the LLM and FL communities.
- Our empirical evaluation reveals that **Oblivionis** outperforms local training, with federated methods delivering an average model utility 27.43% higher than the best local training. This achievement strikes a robust balance between forgetting efficacy and model utility, while cross-comparisons of algorithms provide valuable insights for advancing future LLM development.

2 Related Work

2.1 Federated Fine-Tuning for LLM

FL enables collaborative optimization of a shared model across distributed clients without exposing clients' private training data to preserve privacy. Recent advancements in FL have been expressed by FedLLM frameworks. Chen et al. (2023) propose a framework emphasizing pre-training, fine-tuning, and prompt engineering for privacy-sensitive applications in FedLLM. Fan et al. (2023) introduce FATE-LLM, an industrial-grade framework with parameter-efficient fine-tuning and privacy mechanisms for enterprise usage. Ye et al. (2024) propose OpenFedLLM, enabling federated instruction tuning and value alignment, outperforming local training in financial benchmarks. Wu et al. (2024a) present FedBiOT, a resource-efficient fine-tuning approach using compressed models and adapters. Wu et al. (2024b) further explore federated Reinforcement Learning from Human Feedback (RLHF). They propose FedBis and FedBiscuit strategies to enhance FedLLM alignment while handling client preference heterogeneity (Wu et al. 2024b). These works significantly advance FedLLM training, enhancing efficiency and privacy for distributed learning. However, existing FedLLM frameworks often lack robust unlearning mechanisms, failing to address GDPR or effectively remove low-quality or outdated data contributions.

2.2 LLM Unlearning

LLMs have achieved remarkable success across diverse domains, yet their dependence on enormous datasets raises significant privacy and ethical concerns, such as compliance with GDPR's *right to be forgotten* and the removal of low-quality knowledge or biased content. In response, machine unlearning has emerged as a critical mechanism to address these issues by selectively removing specific knowledge from trained models without compromising overall model performance. It strategically modifies the trained model to erase required information without retraining from scratch.

Dorna et al. (2025) introduce a unified framework to standardize and accelerate the evaluation of unlearning algorithms for large language models, ensuring reproducibility and transparency through consistent metrics and datasets. Yao et al. (2024) provide a comprehensive overview of LLM unlearning, highlighting challenges like catastrophic forgetting and the difficulty of unlearning deeply integrated knowledge. Liu et al. (2025) reconsider LLM unlearning objectives

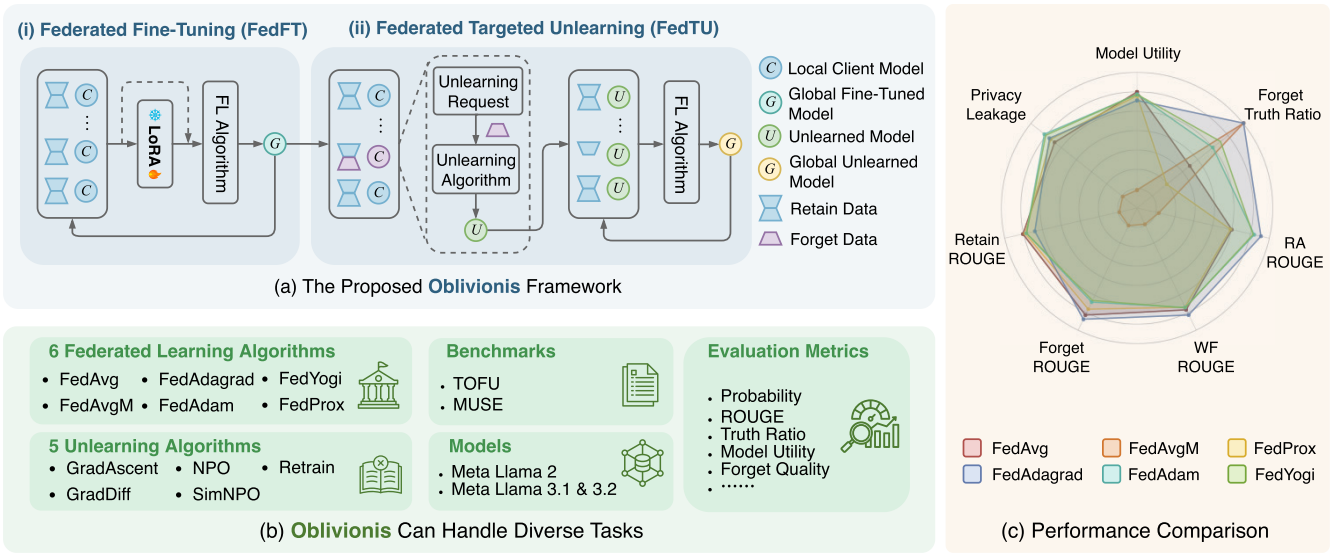

Figure 2: **(a)** Overview of the proposed **Oblivionis** framework. **(b) Oblivionis** integrates 6 representative federated learning algorithms, 5 machine unlearning methods, 2 federated fine-tuning methods (full-parameter and LoRA-based), and a variety of models. **Oblivionis** also supports 5 datasets and over 10 evaluation metrics. **(c)** Sample experimental results that showcase the divergent performance of **6 FL methods** using **SimNPO** unlearning algorithm on the **TOFU** dataset.

from a gradient perspective, advocating algorithms that minimize the influence of target data on model gradients. To enhance efficiency, Jia et al. (2024) introduce SOUL, leveraging second-order optimization to achieve faster convergence in unlearning tasks. Similarly, Ji et al. (2024) develop a framework based on logit differences, reversing forget-retain objectives to efficiently remove specific knowledge. More targeted approaches, such as UIPE by Wang et al. (2025b), focus on disentangling knowledge related to forgetting targets, while Fan et al. (2024) demonstrate that simpler negative preference optimization can also outperform. These works collectively highlight the diversity of approaches in LLM unlearning, ranging from gradient-based algorithms and second-order optimization to targeted knowledge removal and simplified objectives. Despite these advancements, existing frameworks rarely address the joint optimization of federated fine-tuning and unlearning, leaving a gap in achieving both forgetting and model utility, which **Oblivionis** aims to fill.

3 Overview of Framework

This section formalizes the **Oblivionis** framework. In **Oblivionis**, multiple clients train a shared model collaboratively while enabling targeted removal of specific data contributions via unlearning requests, as shown in Figure 2. The framework treats FL and unlearning as a dual optimization problem, with FL denoted by the operator $\mathcal{F}$ and unlearning by $\mathcal{U}$, allowing flexibility for various methods.

3.1 Federated Learning Setup

Consider K clients in an FL framework, indexed by $k \in \{1, 2, \ldots, K\}$. Each client C_k holds a private dataset: $\mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^{N_k}$, where x_i and y_i are sequences of tokens (input/prompt and output/response, respectively), and $N_k = |\mathcal{D}_k|$ is the number of samples for client k. The token sequences are used to fine-tune an LLM parameterized by $\theta \in \mathbb{R}^d$, where d is the dimensionality of the model parameters. Let $y_{i,j}$ denote the j-th token in y_i given the concatenated sequence of input x_i and previous tokens $y_{i,<j} = (y_{i,1}, \ldots, y_{i,j-1})$. The probability of generating $y_{i,j}$ is $p(y_{i,j} \mid x_i \oplus y_{i,<j}; \theta)$, where $\oplus$ is the sequence concatenation operator.

To address the high communication overhead of full fine-tuning in FL, where transmitting the entire set of model parameters across clients is computationally expensive, we adopt Low-Rank Adaptation (LoRA) (Hu et al. 2022) for parameter-efficient fine-tuning. LoRA achieves performance comparable to full fine-tuning while significantly reducing the communication and computational costs by updating only a small subset of parameters. Specifically, for each client C_k at communication round $t \in \{1, 2, \ldots, T\}$, LoRA updates a subset of the model parameters for a given weight matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$ in the large language model through a low-rank decomposition:

$$\mathbf{W}_k^t = \mathbf{W} + \Delta\mathbf{W}_k^t, \quad \Delta\mathbf{W}_k^t = \mathbf{A}_k^t \mathbf{B}_k^t \tag{1}$$

where $\mathbf{A}_k^t \in \mathbb{R}^{m \times r}$, $\mathbf{B}_k^t \in \mathbb{R}^{r \times n}$, and $r \ll \min(m, n)$ is the rank of the adaptation. The global model parameters θ_t include the fixed base weights W, while each client C_k optimizes the LoRA parameters $\phi_k^{(t)} = \{\mathbf{A}_k^t, \mathbf{B}_k^t\}$ during local training. The full set of model parameters is denoted as $\theta = \theta_{\text{base}} + \phi$, where θ_{base} is the set of frozen pre-trained parameters, and ϕ represents the LoRA parameters. Since r is small, $|\phi| \ll |\theta|$, substantially reduces the parameter optimization burden.

3.2 Federated Fine-Tuning (FedFT)

Federated fine-tuning collaboratively optimizes the global model θ_t across all clients over T communication rounds. Each client k first conducts local training on its local model. The base model parameters θ_{base} remain fixed. At round t, client C_k receives global LoRA parameters ϕ^{t-1}, initializes the local LoRA parameters $\phi_k^{(t,0)} = \phi^{t-1}$ and performs R iterations of local optimization on $\mathcal{D}_k$ using stochastic gradient descent (SGD) on the LoRA parameters. For iteration $r \in \{1, 2, \ldots, R\}$:

$$\phi_k^{(t,r)} = \phi_k^{(t,r-1)} - \eta \nabla_\phi \mathcal{L}_k \left(\phi_k^{(t,r-1)}; \mathcal{B}_k \right) \tag{2}$$

where η is the learning rate, and $\mathcal{B}_k \subseteq \mathcal{D}_k$ is a mini-batch. The mini-batch loss is:

$$\mathcal{L}_k \left(\phi_k^{(t,r-1)}; \mathcal{B}_k \right) = \frac{1}{|\mathcal{B}_k|} \sum_{(x_i, y_i) \in \mathcal{B}_k} - \sum_{j=1}^{n_i} \log p \left(y_{i,j} \mid x_i \oplus y_{i,<j}; \theta_{\text{base}} + \phi_k^{(t,r-1)} \right) \tag{3}$$

where $n_i = |y_i|$ is the length of the output sequence, and the probability is computed using the model with parameters $\theta_{\text{base}} + \phi_k^{(t,r-1)}$.

Federated Learning Process: The FL operator $\mathcal{F}$ aggregates local updates to produce the global parameters:

$$\phi^t = \mathcal{F}(\{\phi_k^{(t,R)}\}_{k=1}^K, \phi^{t-1}, \{\mathcal{D}_k\}_{k=1}^K) \tag{4}$$

where $\mathcal{F}$ can represent methods like weighted averaging (e.g., $\phi^t = \phi^{t-1} + \sum_{k=1}^K w_k (\phi_k^{(t,R)} - \phi^{t-1})$, with $w_k = \frac{N_k}{\sum_{j=1}^K N_j}$) or other schemes. The FedFT objective is:

$$\mathcal{L}_{\text{FedFT}}(\phi^t) = \sum_{k=1}^K w_k \mathcal{L}_k(\phi^t; \mathcal{D}_k) \tag{5}$$

3.3 Federated Targeted Unlearning (FedTU)

A client $C_u \in \{1, 2, \ldots, K\}$ requests unlearning of a subset $\mathcal{D}_u^{\text{forget}} = \{(x_i, y_i)\}_{i \in \mathcal{I}_u} \subseteq \mathcal{D}_u$, where $\mathcal{I}_u$ is the index set of the data points to be unlearned. The goal is to derive global LoRA parameters ϕ_{unlearn}^t that approximate a model trained without $\mathcal{D}_u^{\text{forget}}$, while preserving performance on the remaining data $\bigcup_{k=1}^K \mathcal{D}_k \setminus \mathcal{D}_u^{\text{forget}}$. The unlearning operator $\mathcal{U}$ produces:

$$\phi_{\text{unlearn}}^t = \mathcal{U}(\phi^t, \mathcal{I}_u, \mathcal{D}_u^{\text{forget}}) \tag{6}$$

where $\mathcal{U}$ represents a general unlearning method (e.g., gradient ascent, influence functions). The server updates the global parameters: $\phi^{t+1} = \phi_{\text{unlearn}}^t$, and broadcasts ϕ^{t+1} to all clients. Clients then resume local fine-tuning using Equation (2) to compute $\phi_k^{(t+1,r)}$.

$$\phi^{t+2} = \mathcal{F}(\{\phi_k^{(t+1,R)}\}_{k=1}^K, \phi^{t+1}, \{\mathcal{D}_k \setminus \mathcal{D}_u^{\text{forget}}\}_{k=1}^K) \tag{7}$$

The FedTU objective minimizes the influence of $\mathcal{D}_u^{\text{forget}}$:

$$\mathcal{L}_{\text{FedTU}}(\phi_{\text{unlearn}}^t) = \mathcal{L}_{\text{FedFT}}(\phi_{\text{unlearn}}^t; \bigcup_{k=1}^K \mathcal{D}_k \setminus \mathcal{D}_u^{\text{forget}}) \tag{8}$$

Finally, the Unified Framework alternates between FedFT and FedTU, solving the dual optimization objective problem:

$$\min_{\phi_{\text{unlearn}}^t} \min_{\phi^t} \left(\mathcal{L}_{\text{FedFT}}(\phi^t) + \mathbb{I}_{\text{unlearn}}(t) \cdot \mathcal{L}_{\text{FedTU}}(\phi_{\text{unlearn}}^t) \right) \tag{9}$$

This is achieved by iteratively applying $\mathcal{F}$ for FedFT and $\mathcal{U}$ for FedTU. At each communication round t, the server checks for unlearning requests from a client C_u specifying $\mathcal{D}_u^{\text{forget}}$. If present, $\mathcal{U}$ is activated $\mathbb{I}_{\text{unlearn}}(t) = 1$; otherwise, only $\mathcal{F}$ is applied $\mathbb{I}_{\text{unlearn}}(t) = 0$. Our framework supports various implementations of $\mathcal{F}$ and $\mathcal{U}$, ensuring flexibility.

4 Experiments

4.1 Experimental Setups

To explore the performance of different algorithms in the **Oblivionis** framework, we conduct comprehensive experiments using a carefully designed experimental setup.

Models and Benchmark Datasets. We consider four base models in our experiments: *Llama-2-7b-hf* (Touvron et al. 2023), *Llama-3.1-8B-Instruct*, *Llama-3.2-1B-Instruct*, and *Llama-3.2-3B-Instruct* (Grattafiori et al. 2024). We fine-tune and evaluate these models on two benchmark datasets: **TOFU** and **MUSE**, selected based on prior works (Wang et al. 2024; Yuan et al. 2024; Dorna et al. 2025). The **TOFU** dataset is divided into four subsets: *Forget Set (Forget)*, *Retain Set (Retain)*, *Real Authors (RA)*, and *World Facts (WF)*. The **MUSE** dataset comprises two corpora, *News* and *Books*, to simulate real-world large-scale unlearning requests and evaluate forgetting efficacy and model utility preservation in machine unlearning algorithms. Both benchmarks are evaluated under the **Split99** setting (*i.e.*, a stratified partition with 1% of instances in the Forget Set and 99% in the Retain Set).

Metrics. We evaluate unlearning methods along three dimensions: **Memorization**, **Privacy**, and **Utility**. On TOFU, we present Model Utility (MU) and Forget Truth Ratio (FTR). On MUSE, we report No Verbatim Memorization (NVM), No Knowledge Memorization (NKM), and Utility Preserved (UP). All metrics adhere strictly to the official benchmark definitions.

To measure how well **Oblivionis** facilitates targeted forgetting without degrading general knowledge, we rely on the **ROUGE** and **Probability** metrics. These metrics analyze the model's forgetting behavior from different perspectives: Forget ROUGE measures the textual similarity between generated and true answers in the Forget Set via ROUGE-L recall, indicating whether the model still produces targeted forgotten information; Forget Probability quantifies the conditional probability of correct answers, capturing subtle changes in output content and probability distribution.

Algorithms	Weighted Averaging-Based FL						Adaptive Optimization FL					
	FedAvg		FedAvgM		FedProx		FedAdagrad		FedAdam		FedYogi	
	MU↑	FTR↑	MU↑	FTR↑	MU↑	FTR↑	MU↑	FTR↑	MU↑	FTR↑	MU↑	FTR↑
Meta Llama-3.2-1B-Instruct with LoRA												
Finetune	0.50	0.49	0.48	0.45	0.50	0.49	0.45	0.62	0.45	0.60	0.45	0.59
GradAscent	**0.46**	0.61	0.00	0.05	0.43	0.64	0.40	0.72	0.44	0.65	**0.46**	0.66
GradDiff	**0.46**	0.63	6.5e-5	0.70	0.44	0.60	0.42	0.70	0.44	0.66	0.44	0.67
NPO	**0.46**	0.62	2.9e-5	0.71	0.44	0.63	0.41	0.74	0.45	0.68	0.45	0.68
SimNPO	**0.46**	0.65	1.8e-4	0.69	0.43	0.66	0.42	0.74	**0.46**	0.69	**0.46**	0.70
Retrain	**0.51**	0.65	0.47	0.62	**0.51**	0.64	0.46	0.67	0.46	0.66	0.46	0.66
Meta Llama-3.2-3B-Instruct with LoRA												
Finetune	0.59	0.49	0.56	0.48	0.58	0.51	0.53	0.61	0.50	0.57	0.50	0.57
GradAscent	**0.52**	0.59	1.5e-4	0.79	0.48	0.62	0.45	0.73	**0.52**	0.66	0.51	0.66
GradDiff	**0.52**	0.59	6.2e-4	0.77	0.49	0.59	0.47	0.71	0.51	0.61	0.51	0.61
NPO	**0.50**	0.62	3.2e-4	0.79	0.47	0.60	0.45	0.73	**0.50**	0.63	**0.50**	0.63
SimNPO	**0.51**	0.61	1.3e-3	0.77	0.48	0.62	0.47	0.73	0.50	0.63	**0.51**	0.65
Retrain	**0.59**	0.64	0.56	0.64	0.57	0.65	0.53	0.66	0.50	0.63	0.50	0.63

Table 1: Performance comparison of federated learning and unlearning algorithms on the **TOFU** dataset using **Llama-3.2-1B and 3B** models, evaluated on metrics MU (Model Utility) and FTR (Forget Truth Ratio) with **Split99** strategies. Scores in **Bold** indicate the optimal MU in different FL methods, while scores underlined indicate the optimal FTR in different FL methods.

Baselines. We employ six well-established federated optimization algorithms and five unlearning algorithms as baselines, detailed as follows:

- **FL Algorithms:** We categorize the considered FL algorithms into two groups: **Adaptive Optimization FL (AOFL)**, including *FedAdagrad*, *FedAdam*, and *FedYogi* (Reddi et al. 2020), which enhance aggregation with momentum or adaptive learning rates; and **Weighted Averaging-Based FL (WAFL)**, comprising *FedAvg* (McMahan et al. 2017), *FedAvgM* (Hsu, Qi, and Brown 2019), and *FedProx* (Li et al. 2020), which focus on parameter averaging or regularization. By focusing on these foundational and widely applicable algorithms, **Oblivionis** ensures scalability and extensibility for diverse FL scenarios.
- **Unlearning Algorithms:** We denote the fine-tuned model obtained in Step 2 of Figure 1 as **Finetune**, which is an FL model with full knowledge. Meanwhile, the integrated unlearning algorithms are classified into two types: **Gradient-Based Optimization Unlearning (GOUL)**, including *GradAscent* and *GradDiff* (Maini et al. 2024); and **Preference Optimization Unlearning (POUL)**, including *NPO* (Zhang et al. 2024) and *SimNPO* (Fan et al. 2024). **GOUL** directly manipulates gradients or representations to eliminate the influence of data targeted for forgetting, employing simpler, targeted adjustments. Besides, **Retrain** performs federated fine-tuning from scratch on the remaining data using the initial base model, thereby obtaining an ideal unlearned model.

Training Setup. We conduct experiments using 30 clients with a 10% participation rate for **Oblivionis**. In each round, a randomly selected client requests targeted sample-level unlearning. The training process consists of 5 local epochs and 10 global rounds, with a one-epoch warmup period included. The base models are fine-tuned using LoRA with a rank of 32, an alpha of 64, and a dropout rate of 0.05. We train the model with a learning rate of $8e{-}5$ and a weight decay of 0.01. The entire experiment is tested on a cloud server with one NVIDIA A100 (80 GB) GPU.

Model	Size	N_{Base}	$N_{\text{Trainable}}$	Ratio (%)
Llama-2	7B	6818.37 M	79.95 M	1.17
Llama-3.1	8B	8114.15 M	83.89 M	1.03
Llama-3.2	1B	1258.36 M	22.54 M	1.79
	3B	3261.38 M	48.63 M	1.49

Table 2: Illustration of model parameter distribution.

Meanwhile, Table 2 summarizes the number of trainable parameters under the LoRA paradigm. In all cases, no more than 1.79% of base models' parameters are updated, while the rest remain frozen, highlighting the lightweight nature of our approach. For more experimental settings, including specific methods of federated learning and unlearning, datasets, and models, please refer to the contents in the Appendix section of our extension version.

4.2 Experimental Results

Structured QA Task. As presented in Table 1, we choose Model Utility (MU) and Forget Truth Ratio (FTR) to evaluate. AOFL algorithms, particularly FedAdagrad, consistently outperform WAFL methods in forgetting efficacy. For the 1B model, FedAdagrad, when paired with SimNPO or NPO, achieves an FTR of 0.74, surpassing FedAvg's 0.65 and FedProx's 0.66. Similarly, for the 3B model, FedAdagrad attains an FTR of 0.66, compared to 0.64 for FedAvg and 0.65 for FedProx. These findings indicate that AOFL methods effectively utilize adaptive optimization to prioritize the Forget Set objectives, thereby maintaining unlearning performance. However, this enhancement results in a reduction in MU, with FedAdagrad yielding MU values ranging from 0.40 to 0.47,

Figure 3: Comparative analysis of **ROUGE** scores across federated learning and unlearning methods using **Llama-3.2-1B** model with **Split99** strategies. For the Forget set, lower scores indicate better performance (↓), whereas for the remaining sets, higher scores are preferable (↑).

whereas FedAvg maintains more stable MU values between 0.46 and 0.59 across both models. Among unlearning strategies, SimNPO and NPO demonstrate superior forgetting efficacy, achieving FTR values between 0.69 and 0.74 with AOFL methods while maintaining competitive MU values from 0.42 to 0.51. In contrast, the Retrain strategy achieves the highest MU value of up to 0.59 but is computationally intensive, limiting its practical applicability. **Meanwhile, FedAvgM suffers from catastrophic forgetting in the Structured QA Task**, with MU values plummeting to between $1.8e{-}4$ and $1.3e{-}3$, despite achieving high FTR values of up to 0.79. This instability likely arises from FedAvgM amplifying the adverse effects of unlearning updates on general model parameters, resulting in performance collapse.

To evaluate the impact of model scale, we test the larger 3B model, which shows higher MU and FTR values, indicating a better balance between utility and forgetting. For instance, FedAvg with Retrain achieves a MU of 0.59 and an FTR of 0.64 for the 3B model, compared to 0.51 and 0.65 for the 1B model. WAFL methods like FedAvg and FedProx yield stable MU values of 0.47 to 0.59 but lag in FTR compared to AOFL methods. This highlights a trade-off: AOFL methods prioritize forgetting but reduce utility, while WAFL methods ensure stability. All unlearning strategies except Finetune outperform the Finetune baseline's FTR of 0.45 to 0.62 for the 1B model and 0.48 to 0.61 for the 3B model, achieving values of 0.59 to 0.79, confirming **Oblivionis** 's robust unlearning capability.

To validate the effectiveness of **Oblivionis** in forgetting and retaining general knowledge, we evaluated it on all four sets from TOFU, using ROUGE and Probability metrics. As shown in Figure 3 and Figure 4, on the Forget Set, from the initial fine-tuned model to each FU dual-optimization method, both Forget ROUGE and Forget Probability significantly decreased, indicating that the model's generated answers deviated from the true answers, with a substantial reduction in probability preference for correct answers, proving the FU algorithm's effectiveness in altering model output behavior and achieving information forgetting. Meanwhile, on the Retain Set, World Facts, and Real Authors sets, ROUGE and Probability results remained largely consistent with fine-tuning performance, demonstrating that the FU algorithm effectively retains model performance on non-forgotten data while forgetting the Forget Set. Overall, the evaluation confirms the FU algorithm's effective capability for forgetting while maintaining the model's overall performance stability. **Overall, FedAdagrad excels in forgetting efficacy but compromises model utility, whereas FedAvg and FedProx prioritize utility stability, sacrificing forgetting performance in the Structured QA Task.** For a comprehensive analysis involving various model scales and data split strategies, please refer to the results in the Appendix section of our extension version.

Contextual QA Task. FedProx demonstrated a good balance across all objectives on the MUSE News set, as evidenced by Table 3. When combined with GradDiff, it achieves low NVM and NKM of 0.52 and 0.55, respectively, while maintaining high Utility Preserved (UP) at 0.52. These results indicate effective unlearning with robust model performance. FedAvg exhibits moderate performance. When combined with GradAscent, it yields an NVM of 0.41, NKM of 0.49, and UP of 0.35. These results indicate that it is less effective than FedProx in balancing forgetting and model utility. FedAvgM shows poor overall performance. For instance, when combined with GradAscent, it yields extremely low UP at 0.02, despite favorable NVM and NKM of $5.9e{-}3$ and 0.03, respectively. Therefore, we consider it unsuitable

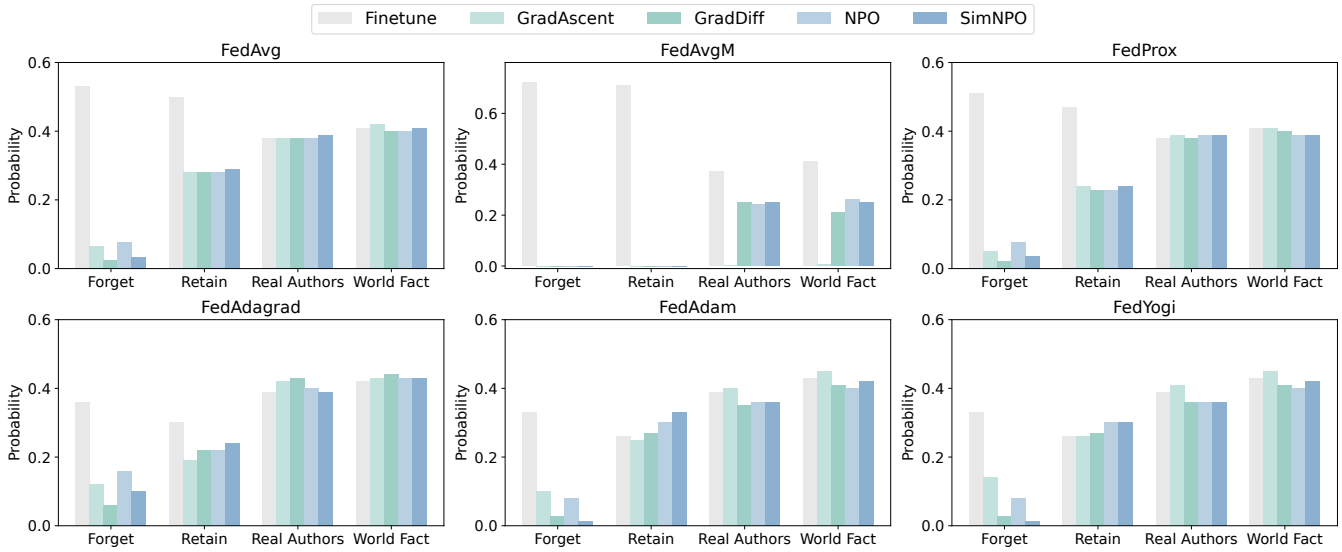

Figure 4: Comparative analysis of **Probability** scores across federated learning and unlearning methods using **Llama-3.2-1B** model with **Split99** strategies. For the Forget set, lower scores indicate better performance (↓), whereas for the remaining sets, higher scores are preferable (↑).

Algorithms	Weighted Averaging-Based FL									Adaptive Optimization FL								
	FedAvg			FedAvgM			FedProx			FedAdagrad			FedAdam			FedYogi		
	NVM	NKM	UP	NVM	NKM	UP	NVM	NKM	UP	NVM	NKM	UP	NVM	NKM	UP	NVM	NKM	UP
Finetune	0.77	0.57	0.43	0.34	0.38	0.31	0.60	0.60	0.52	0.61	0.65	0.53	0.67	0.63	0.50	0.67	0.62	0.50
GradAscent	0.41	0.49	0.35	5.9e-3	0.03	0.02	0.56	0.56	**0.49**	0.03	0.00	0.00	0.46	0.51	0.40	0.44	0.50	0.41
GradDiff	0.39	0.43	0.34	0.25	0.24	0.24	0.52	0.55	**0.52**	0.17	0.53	0.43	0.46	0.49	0.39	0.43	0.53	0.38
NPO	0.36	0.45	0.35	0.33	0.38	0.34	0.42	0.56	**0.43**	0.36	0.50	0.36	0.39	0.47	0.35	0.43	0.44	0.39
SimNPO	0.32	0.39	0.33	0.30	0.41	0.29	0.27	0.51	**0.42**	0.18	0.49	0.36	0.31	0.45	0.36	0.33	0.47	0.38
Retrain	0.21	0.32	0.46	0.18	0.22	0.30	0.21	0.36	**0.52**	0.21	0.33	**0.52**	0.21	0.32	0.50	0.21	0.34	0.50

Table 3: Performance comparison of federated learning algorithms on the **MUSE News** set using **Llama-2-7B model**, evaluated on metrics NVM (No Verbatim Mem↓), NKM (No Knowledge Mem↓), and UP (Utility Preserved↑). Scores in **Bold** indicate the optimal UP in different FL methods, while underlined indicate the optimal NVM and NKM in different FL methods.

for balanced optimization. Among the optimizer-enhanced methods, FedAdam and FedYogi delivered competitive performance. FedAdam achieves an NVM of 0.31, NKM of 0.45, and UP of 0.36 with SimNPO. FedYogi produces similar results with SimNPO, achieving an NVM of 0.33, NKM of 0.47, and UP of 0.38. FedAdagrad achieves less consistent results. When combined with GradDiff, it yields an NVM of 0.17, NKM of 0.53, and UP of 0.43.

From a dual-objective optimization perspective, FedProx effectively minimizes NVM and NKM while maintaining high UP across all unlearning algorithms. FedAdam and FedYogi also achieve a well-balanced trade-off among the objectives, especially when combined with SimNPO. However, its effectiveness is slightly lower than that of FedProx. In contrast, FedAvg emphasizes model utility at the cost of unlearning performance, while FedAvgM prioritizes unlearning performance at the expense of model utility, making both approaches suboptimal. SimNPO and NPO demonstrate robust performance across FL methods, with SimNPO achieving the lowest NVM of 0.27 when paired with FedProx. In summary, **Oblivionis** demonstrates strong effectiveness in balancing the dual-objective optimization of minimizing memorization, while maximizing utility across a majority of the scenarios considered. **Overall, FedProx demonstrates a better trade-off between model utility and unlearning performance in the contextual QA task.**

Comparative Analysis of Local and Federated Learning. Empirical results illustrated in Figure 5 reveal that FU methods consistently achieve higher MU scores than local training across all unlearning strategies, demonstrating superior robustness in preserving model utility during unlearning. Specifically, local training refers to a decentralized approach where each client independently performs fine-tuning and unlearning on its local data without any model aggregation across clients. It exhibits a significant vulnerability to catastrophic forgetting, especially with GradAscent, where MU drops to near-zero levels. In contrast, FL methods mitigate the destabilizing effects of unlearning through collaborative parameter updates and maintain stable and competitive MU scores. Among the unlearning methods, NPO paired with FL algorithms yields the highest MU, indicating strong com-

patibility with the dual-objective optimization framework. In contrast, local training fails to balance unlearning and performance retention across all methods. **In summary, OBLIVIONIS significantly outperforms local training by maintaining robust model utility across unlearning methods, highlighting its efficacy for practical applications.**

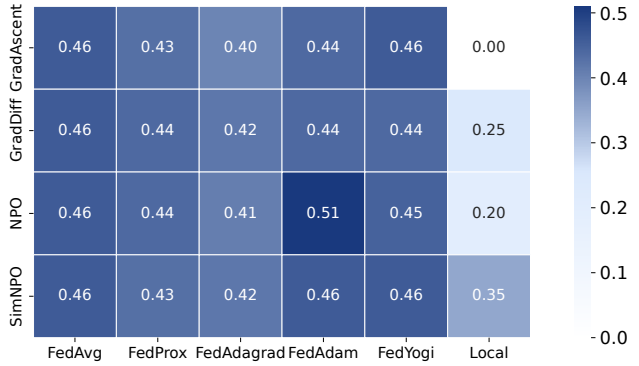

Figure 5: Comparison of **Model Utility(MU)** between local and federated learning across different unlearning methods, using **Llama-3.2-1B model**.

5 Conclusion

In this work, we introduce **OBLIVIONIS**, a lightweight framework that seamlessly integrates federated learning and unlearning to enable distributed model training and compliance with regulations such as GDPR's *right to be forgotten*. By formulating FL and unlearning as a joint dual-objective optimization task, **OBLIVIONIS** achieves a robust balance between forgetting targeted data and preserving model utility, as demonstrated by superior performance on TOFU and MUSE benchmarks. Our comprehensive evaluation, including cross-comparisons of diverse FL and unlearning algorithms, evidences that models trained using **OBLIVIONIS** consistently outperform those trained using local training approaches. Notably, methods like FedAdagrad paired with SimNPO achieve high forgetting efficacy. By consolidating diverse benchmarks and datasets into a user-friendly code library, **OBLIVIONIS** further facilitates standardized research for the LLM and FL communities. Our framework is also open-sourced to facilitate reproducibility and foster further research in the development of LLM.

Acknowledgments

This research is supported by the NTU startup grant and the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL). This research/project is supported by A*STAR under its Japan-Singapore Joint Call: JST-A*STAR 2024 (Project ID: R24I6IR139).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chen, C.; Feng, X.; Zhou, J.; Yin, J.; and Zheng, X. 2023. Federated large language model: A position paper. *arXiv e-prints*, arXiv–2307.

Dorna, V.; Mekala, A.; Zhao, W.; McCallum, A.; Lipton, Z. C.; Kolter, J. Z.; and Maini, P. 2025. OpenUnlearning: Accelerating LLM Unlearning via Unified Benchmarking of Methods and Metrics. *arXiv preprint arXiv:2506.12618*.

Fan, C.; Liu, J.; Lin, L.; Jia, J.; Zhang, R.; Mei, S.; and Liu, S. 2024. Simplicity Prevails: Rethinking Negative Preference Optimization for LLM Unlearning. *arXiv preprint arXiv:2410.07163*.

Fan, T.; Kang, Y.; Ma, G.; Chen, W.; Wei, W.; Fan, L.; and Yang, Q. 2023. Fate-llm: A industrial grade federated learning framework for large language models. *arXiv preprint arXiv:2310.10049*.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hsu, T.-M. H.; Qi, H.; and Brown, M. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Huu-Tien, D.; Pham, T.-T.; Thanh-Tung, H.; and Inoue, N. 2024. On effects of steering latent representation for large language model unlearning. *arXiv preprint arXiv:2408.06223*.

Imani, S.; Du, L.; and Shrivastava, H. 2023. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*.

Ji, J.; Liu, Y.; Zhang, Y.; Liu, G.; Kompella, R. R.; Liu, S.; and Chang, S. 2024. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *Advances in Neural Information Processing Systems*, 37: 12581–12611.

Jia, J.; Zhang, Y.; Zhang, Y.; Liu, J.; Runwal, B.; Diffenderfer, J.; Kailkhura, B.; and Liu, S. 2024. Soul: Unlocking the power of second-order optimization for llm unlearning. *arXiv preprint arXiv:2404.18239*.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2: 429–450.

Liu, S.; Yao, Y.; Jia, J.; Casper, S.; Baracaldo, N.; Hase, P.; Yao, Y.; Liu, C. Y.; Xu, X.; Li, H.; et al. 2025. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, 1–14.

Maini, P.; Feng, Z.; Schwarzschild, A.; Lipton, Z. C.; and Kolter, J. Z. 2024. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Min, S.; Gururangan, S.; Wallace, E.; Shi, W.; Hajishirzi, H.; Smith, N. A.; and Zettlemoyer, L. 2023. Silo language models: Isolating legal risk in a nonparametric datastore. *arXiv preprint arXiv:2308.04430*.

Pardau, S. L. 2018. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J. Tech. L. & Pol'y*, 23: 68.

Reddi, S.; Charles, Z.; Zaheer, M.; Garrett, Z.; Rush, K.; Konečnỳ, J.; Kumar, S.; and McMahan, H. B. 2020. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

Rosen, J. 2011. The right to be forgotten. *Stan. L. Rev. Online*, 64: 88.

Thirunavukarasu, A. J.; Ting, D. S. J.; Elangovan, K.; Gutierrez, L.; Tan, T. F.; and Ting, D. S. W. 2023. Large language models in medicine. *Nature medicine*, 29(8): 1930–1940.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Q.; Zhou, J. P.; Zhou, Z.; Shin, S.; Han, B.; and Weinberger, K. Q. 2025a. Rethinking llm unlearning objectives: A gradient perspective and go beyond. *arXiv preprint arXiv:2502.19301*.

Wang, W.; Zhang, M.; Ye, X.; Ren, Z.; Chen, Z.; and Ren, P. 2025b. Uipe: Enhancing llm unlearning by removing knowledge related to forgetting targets. *arXiv preprint arXiv:2503.04693*.

Wang, Y.; Wei, J.; Liu, C. Y.; Pang, J.; Liu, Q.; Shah, A. P.; Bao, Y.; Liu, Y.; and Wei, W. 2024. Llm unlearning via loss adjustment with only forget data. *arXiv preprint arXiv:2410.11143*.

Webb, T.; Holyoak, K. J.; and Lu, H. 2023. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9): 1526–1541.

Wei, A.; Haghtalab, N.; and Steinhardt, J. 2023. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36: 80079–80110.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wu, F.; Li, Z.; Li, Y.; Ding, B.; and Gao, J. 2024a. Fedbiot: Llm local fine-tuning in federated learning without full model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3345–3355.

Wu, F.; Liu, X.; Wang, H.; Wang, X.; and Gao, J. 2024b. On the client preference of llm fine-tuning in federated learning. *arXiv preprint arXiv:2407.03038*.

Wu, S.; Irsoy, O.; Lu, S.; Dabravolski, V.; Dredze, M.; Gehrmann, S.; Kambadur, P.; Rosenberg, D.; and Mann, G. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Yao, Y.; Xu, X.; and Liu, Y. 2024. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37: 105425–105475.

Ye, R.; Wang, W.; Chai, J.; Li, D.; Li, Z.; Xu, Y.; Du, Y.; Wang, Y.; and Chen, S. 2024. Openfedllm: Training large language models on decentralized private data via federated learning. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 6137–6147.

Yuan, X.; Pang, T.; Du, C.; Chen, K.; Zhang, W.; and Lin, M. 2024. A closer look at machine unlearning for large language models. *arXiv preprint arXiv:2410.08109*.

Zhang, R.; Lin, L.; Bai, Y.; and Mei, S. 2024. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*.