Is Client Unlearning Really Necessary in Federating Learning?

Alessio Mora
University of Bologna
Bologna, Italy
alessio.mora@unibo.it

Paolo Bellavista
University of Bologna
Bologna, Italy
paolo.bellavista@unibo.it

***Abstract*—Recent privacy regulations worldwide, such as the European Union's General Data Protection Regulation (GDPR), enforce the "right to be forgotten", allowing individuals to withdraw consent for the use of their data and requiring entities to delete it. In Federated Learning (FL), where a global machine learning model is collaboratively trained by sharing model weights and updates derived from participants' local data, this right necessitates the ability to remove the contribution of client devices willing to be forgotten. In this paper, we empirically show that, without explicit and specific methods to remove the contributions of clients who request unlearning, their traces remain detectable long after they detach. To this end, we monitor the accuracy on forget data (the data held by the client requesting unlearning) and the susceptibility to membership inference attacks targeting the forget data across rounds that follow the unlearning request. We compare these metrics with those of a retrained model, a model that has never been exposed to the forget data. Our results show that, without applying explicit unlearning methods, a specific client's contribution gradually diminishes over the course of FL rounds but can still remain noticeable even after 50 rounds.**

***Index Terms*—federated learning, federated unlearning, data heterogeneity, machine unlearning.**

I. Introduction

During the last year, Federated Learning (FL) [1] has gained significant attention due to its privacy-preserving design. Machine learning models can be trained collaboratively without requiring the collection or transfer of raw data from the data subject's device. Instead, only processed and ephemeral information, such as model weights and updates, is communicated over the network to a centralized third-party location. While FL systems are increasingly being adopted in various domains [2]–[4], it remains unclear how to ensure compliance with a cornerstone of modern privacy regulations worldwide, such as the GDPR [5], [6]: "the right to be forgotten". Also referred to as the right of oblivion, this concept describes an individual's right to have their personal data erased or no longer processed upon withdrawing consent. While it is straightforward for storage-based systems to delete data or its contributions, the enforcement of this right becomes problematic when the data has been used in a machine learning pipeline to train a predictive model. As is well known, machine learning models and especially deep learning models are prone to memorize the data seen during the training phase [7], and can later leak information about them [8]–[10]. A trivial but costly solution would be to retrain the model from scratch on a sanitized dataset that excludes the forget data (i.e., the data to be removed). This difficulty is worsened when decentralized learning techniques such as FL are used to train the model. In fact, in decentralized learning settings, different complications arise in addition to the inefficiency of retraining from scratch. Among the others, data are partitioned across parties (clients), and these data are inscrutable by design, so traditional (centralized) machine unlearning strategies are often not directly applicable.

Also, the iterative nature of FL complicates the removal of a specific client's contribution. The global model is an aggregation of updates per round from clients, so removing a past contribution would invalidate all subsequent aggregations[1]. Finally, in cross-device settings [11], [12], client participation is often sporadic, with some devices contributing only once throughout the learning process. Retraining the model from scratch becomes infeasible if certain clients who have participated (and wish to remain part of the federation) are unreachable, for instance, due to connectivity issues or depleted batteries.

Therefore, methods specifically designed for FL settings that are effective and efficient in removing a client's contribution are essential [13], [14]. A client unlearning mechanism is effective if, after applying the forgetting routine, the sanitized model has performance on the forget data similar to a model that has never seen the data before, called the retrained model. In contrast, a client unlearning mechanism is efficient if it minimizes the number of recovery rounds needed to achieve or surpass the generalization performance of the retrained model.

Currently, a common assumption tends to be that explicit unlearning mechanisms are necessary. Upon receiving an unlearning request, the FL system typically applies an unlearning algorithm, after which the training resumes from a sanitized global model. This process often involves a series of recovery rounds, as unlearning algorithms aim to selectively erase the targeted contribution, but may inadvertently damage the performance on data similar to the forget data (e.g., data that share common features). Although some client unlearning methods have been proposed [15]–[18], one might question whether the contribution of the targeted client would naturally fade away

[1]This approach would also require storing all historical client updates, which is inefficient and impractical.

after a few rounds, even without explicit unlearning. Indeed, a phenomenon such as catastrophic forgetting [19], commonly observed in both continual and federated learning [20], could accelerate the fading of contributions from detached clients. This effect is particularly pronounced in federated settings where data is typically heterogeneous [21], as the participation of clients that hold data from diverse statistical distributions can overwrite or diminish the influence of earlier contributions.

In this paper, we empirically monitor what happens to a significant series of forgetting indicators – accuracy and loss on forget data, membership attack success rates, and others – when a client leaves the federation and the training continues without applying explicit and active unlearning routines. By analyzing these metrics, we want to understand whether client unlearning methods are really necessary and under which conditions, which is a topic insufficiently covered yet in the existing literature.

II. Background

A. Federated Learning

In FL, a group of devices collaboratively trains a shared model known as the global model. Federated Averaging (FedAvg) [1] is a widely recognized baseline algorithm for FL. FedAvg operates in iterative learning rounds, with each round comprising the following key steps:

- Client Selection: A subset of active client devices is chosen to participate in the current round.
- Model Distribution: A central entity, often referred to as the parameter server, distributes the current global model parameters to the selected clients.
- Local Training: Each participating client trains the received global model on its private data for a specified number of epochs. The client then sends back either the updated model parameters or computed model updates.
- Aggregation: The parameter server collects the contributions from participating clients and combines them using a weighted average to generate an updated global model.

This newly aggregated global model is then distributed to clients in the subsequent round, repeating the cycle until convergence or the training objective is met.

B. Federated Unlearning

Federated unlearning can be categorized into three main types: class unlearning, sample unlearning, and client unlearning [13]. In this paper, we focus exclusively on client unlearning.

During the FL process, a client may, at any given round t, request the removal of knowledge derived from its data. The dataset provided by this client, which must be erased from the model, is referred to as the forget data. The remaining clients collectively provide the retain data, representing the combined datasets of all participating clients except for the unlearning client. The model trained with contributions from all clients, including the unlearning client's data, is called the original model. After the unlearning process, the resulting model is termed the unlearned model, which should behave as if it had never encountered the forget data. The gold-standard baseline for the performance of the unlearned model is the retrained model, which is trained from scratch using only the retain data. The retrained model is used to determine whether the unlearned model has successfully removed the influence of the forget data while maintaining acceptable performance in generalization.

Client unlearning aims to efficiently produce an unlearned model without the costly and often impractical process of retraining from scratch. This challenge is even worse in cross-device FL scenarios, where clients may participate infrequently or even just once during the entire training process. In such cases, replicating the original training setup for retraining becomes not only inefficient but also logistically infeasible.

In this paper, we analyze the impact of the *natural* baseline, which involves no explicit client unlearning algorithm. After an unlearning request, training continues as usual, excluding only the client that requested to leave. Our goal is to determine whether the client's contribution is rapidly overwritten during subsequent training.

III. Is Client Unlearning Necessary?

In this section, we conduct an extensive empirical analysis of the evolution of forgetting metrics that occur after a client's departure from the federation. Our primary objective is to investigate whether, and how quickly, the contribution of a specific client diminishes naturally over subsequent rounds. The experimental evaluation considers data heterogeneity across clients – a common, realistic, and challenging assumption in most state-of-the-art FL settings [11], [12].

Before presenting and discussing the empirical results, we first detail the metrics used to conduct the analysis, followed by a description of the experimental setup.

A. Metrics for Forget Assessment

We use five metrics to evaluate whether and to what extent the global model forgets a target client's dataset. These metrics, detailed below, can provide researchers with valuable insights, particularly when compared to the corresponding values from a retrained model. This comparison serves as a benchmark to understand whether explicit client unlearning is really required and also identifies/experiments a generalizable methodology to evaluate the need for unlearning techniques in which FL application cases.

Accuracy on forget data. One of the simplest metrics to assess the effectiveness of forgetting is the accuracy on forget data. A model trained on specific data typically achieves higher accuracy on those samples compared to unseen data, as it has learned patterns directly from the training corpus. Therefore, as forgetting progresses, the accuracy on the forget data should decrease, ideally approaching the performance of a model that has never been exposed to those data (i.e., the retrained model).

Loss on forget data. In addition to accuracy, the loss on forget data is a valuable metric in this context. In fact, a model tends to exhibit lower loss values on training samples because it has optimized its parameters to minimize errors on these

data points. As forgetting mechanisms are applied, the loss on forget data should increase, reflecting the model's reduced ability to confidently predict those samples. A higher loss indicates that the model's predictions on forget data are less confident and more similar to predictions made on unseen or random data, thus being a symptom of successful forgetting.

MIA success rates on forget data. Membership Inference Attacks (MIAs) seek to determine whether specific samples were included in the training dataset during model training. The success rate of such attacks represents the proportion of samples correctly identified as part of the training data. In our experiments, we employ two different implementations to determine MIAs, providing a robust evaluation of the attack's effectiveness. The first method, proposed in [9], requires composing two datasets with the same cardinality. One includes training data from the retain dataset, which corresponds to the combined data of all clients excluding the target one. The other dataset only comprises unseen data. As usual, we use test data for the latter, since the model has never encountered such data during training, by definition. The MIA predictor learns to distinguish whether the target model's output is more likely to be associated with training data or unknown data. To evaluate forgetting, the MIA predictor is applied to the model using the forget data as input. The second one, introduced in [10], relies on the average training loss of the target model. A sample is classified as a training set member if its loss is below this average; otherwise, it is labeled as a non-member. While useful for baseline comparisons, this approach is impractical in real-world FL scenarios. Access to either the local average training loss of each client (which is private by design in FL) or the global model's average loss across all training data (infeasible due to data distribution and privacy constraints) would be required. In our implementation, we use the latter approach for the purpose of our discussion and analysis, well aware that this is not a realistic membership attack.

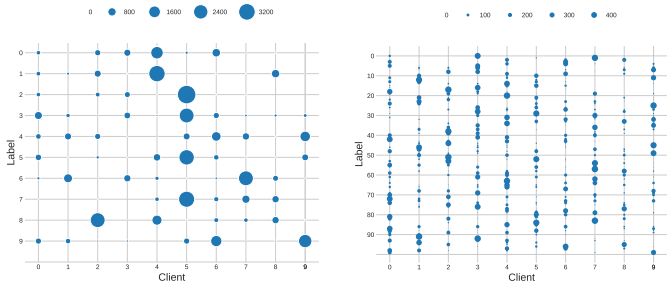
For the MIA success rate, a lower value indicates better effectiveness of the forgetting algorithm, as it suggests that the model retains less information about the unlearned data. However, the ideal outcome is to minimize the difference between the success rate on the unlearning model and that on the retrained model, thus ensuring consistency and robust unlearning.

Kullback-Leibler (KL) divergence between random outputs and model outputs on forget data. As proposed in [17], by measuring the KL divergence between random outputs and the model outputs on forget data, we aim to evaluate how closely the target model predictions on the forget data resemble random and uninformed guesses. A low KL divergence indicates that the model predictions are closer to random, suggesting effective forgetting of the target data. Conversely, a high KL divergence implies that the model retains structured knowledge about the forget data.

B. Experimental Setting

For our experimental setup, we used federated versions of two widespread, well-known, and reference visual datasets, CIFAR-10 and CIFAR-100. Both datasets consist of 32x32 color images, with CIFAR-10 containing 10 classes and CIFAR-100 containing 100 classes. We adopted a similar configuration to the one in [15], simulating a federation of 10 clients with heterogeneous data distributions. Data partitioning followed the method in [22], using Latent Dirichlet Allocation (LDA) to control the level of heterogeneity. Specifically, we set the concentration parameter $\alpha = 0.3$ for CIFAR-10 and $\alpha = 0.1$ for CIFAR-100. The resulting per-client label distribution is shown in Fig. 1. For both datasets, we conducted 200 rounds of standard FedAvg training with full client participation, simulating a cross-silo FL setting. In the natural baseline experiment, one of the 10 clients left the federation after the 200th round, detaching from the federation. The results presented in this section focus on the rounds following the 200th round when the unlearning request was initiated. As neural network, we employed a ResNet-18. Throughout training, both before and after the unlearning request, we used Stochastic Gradient Descent as the client optimizer with a learning rate of 0.1 and an exponential decay rate of 0.998, following [23]. Clients performed 1 local epoch, and the local batch size was set to 32; we employed the standard FedAvg aggregation method.

C. Experimental Results



(a) CIFAR-10. (b) CIFAR-100.

Fig. 1: Label distribution across clients.

Fig. 2 and Fig. 3 show the evolution of forgetting metrics after a specific client leaves the federation, and track their value over 50 rounds, for the CIFAR-100 experiments. We present the results of four independent experiments, where a different client is designated as the unlearning client in each experiment. The unlearning client exits the federation at round 200, prior to the 50-round phase depicted in the figures. The accuracy on forget data, along with the success rates of MIA [10] and MIA [9], rapidly decreases toward the gold-standard values of the retrained model during the initial rounds. However, the rate of decrease slows down over time, and the values remain significantly different from the baseline, failing to complete the natural unlearning. The loss on the forget data follows a similar trend, rapidly increasing early and stabilizing far from the loss value of the retrained model. The KL divergence metric reflect expected trends on the CIFAR-100 experiments, while for CIFAR-10 experiments its values

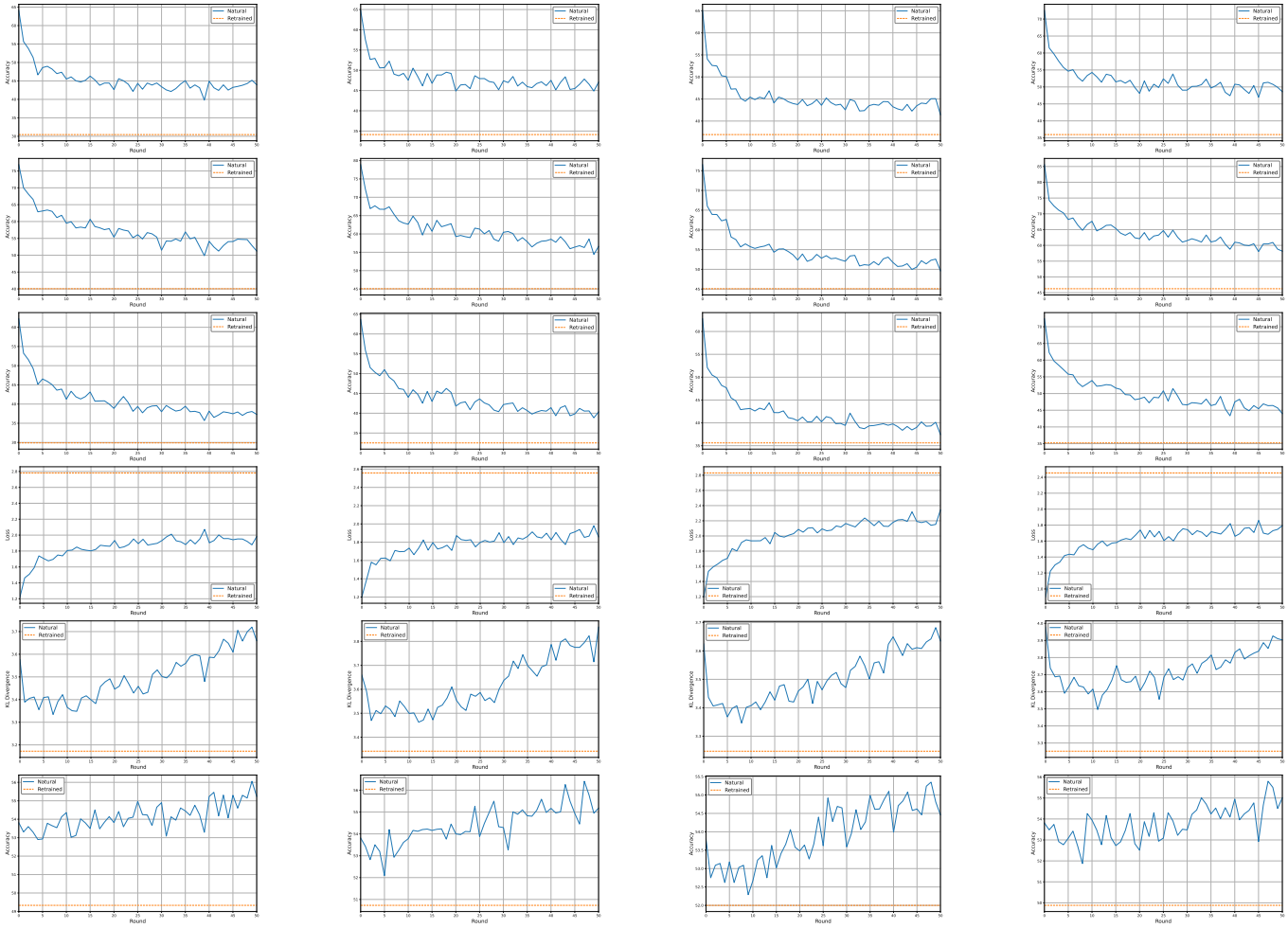


Fig. 2: Comparison of forgetting metrics between the retrained model and the natural baseline on the CIFAR-100 dataset. Each column of charts reports the results for a different client selected as the unlearning client (Client 0, Client 1, Client 2, Client 3). **From top to bottom:** accuracy on forget data, MIA [9] success rate, MIA [10] success rate, loss on forget data, KL divergence, test accuracy. The dashed lines (constant values) represent the value of that metric for the retrained model at the unlearning round (round 200), while for the natural baseline, the solid lines track the forgetting metric over 50 rounds following the unlearning request.

CIFAR-100					CIFAR-10				
Round	Forget Acc.	Test Acc.	MIA [9]	MIA [10]	Round	Forget Acc.	Test Acc.	MIA [9]	MIA [10]
1	$14.14_{\pm 6.67}$	$3.19_{\pm 1.21}$	$15.11_{\pm 6.97}$	$14.21_{\pm 6.86}$	1	$20.64_{\pm 7.59}$	$2.13_{\pm 1.57}$	$23.42_{\pm 8.19}$	$20.70_{\pm 7.56}$
10	$12.52_{\pm 7.11}$	$3.71_{\pm 1.84}$	$11.58_{\pm 7.29}$	$12.29_{\pm 6.48}$	10	$13.91_{\pm 5.48}$	$2.73_{\pm 1.56}$	$15.48_{\pm 6.24}$	$12.36_{\pm 5.59}$
12	$12.76_{\pm 7.18}$	$4.25_{\pm 1.64}$	$12.31_{\pm 7.21}$	$12.24_{\pm 6.67}$	12	$13.01_{\pm 4.79}$	$2.45_{\pm 1.40}$	$14.32_{\pm 5.75}$	$11.53_{\pm 4.80}$

TABLE I: Performance on forget data (forget accuracy and MIA success rates) and generalization performance (test accuracy) at different rounds after the unlearning request, for the natural baseline. The values represent the absolute difference in performance compared to the retrained model. The performance gap (%) is presented as $a \pm b$, where a is the mean and b is the standard deviation, computed over 10 different experiments, with a different client serving as the unlearning client in each experiment. Note that better forgetting corresponds to a smaller performance gap with the gold-standard retrained model.

are constantly higher than the baseline (except for some rounds where a consistent drop in all the metrics is observed due to a drop in the generalization performance of the model, a fluctuation that can happen with heterogeneous data). The KL divergence metric exhibits expected trends in the CIFAR-100 experiments. However, in the CIFAR-10 experiments, its values consistently remain higher than the baseline, except during some rounds where a notable drop in all metrics occurs. This drop is attributed to a decline in the model's generalization performance (test accuracy), a fluctuation that can naturally arise with heterogeneous data. The observed behavior of KL divergence in the CIFAR-10 experiments raises

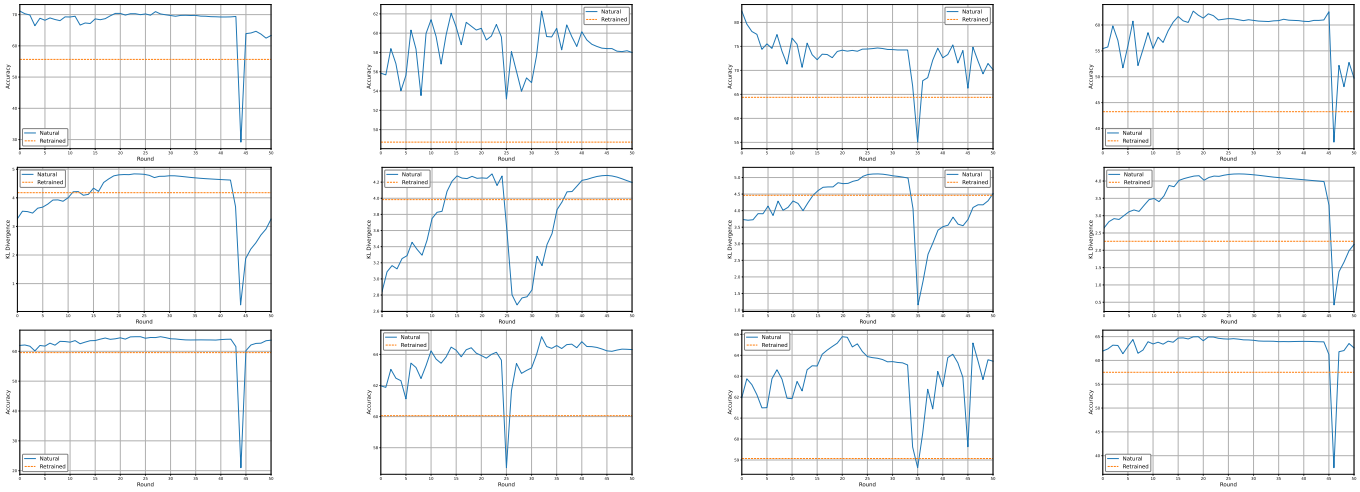


Fig. 3: Comparison of forgetting metrics between the retrained model and the natural baseline on the CIFAR-10 dataset. Each column of charts reports the results for a different client selected as the unlearning client. **From top to bottom:** accuracy on forget data, KL divergence, test accuracy.

	CIFAR-100					CIFAR-10				
ID	Test Acc.	Forget Acc.	Forget Loss	MIA [9]	MIA [10]	Test Acc.	Forget Acc.	Forget Loss	MIA [9]	MIA [10]
0	55.24 (5.91)	43.97 (13.51)	1.98 (-0.80)	51.26 (11.07)	37.31 (7.42)	63.74 (4.17)	63.31 (7.68)	1.26 (-0.78)	61.29 (10.99)	60.48 (6.81)
1	55.18 (4.45)	47.06 (12.94)	1.85 (-0.70)	56.60 (11.54)	40.33 (7.82)	64.31 (4.25)	58.00 (9.30)	1.83 (-0.49)	52.95 (8.94)	54.33 (7.28)
2	54.46 (2.46)	41.44 (4.51)	2.34 (-0.49)	49.62 (4.52)	37.44 (1.82)	63.72 (4.65)	70.16 (5.77)	1.20 (-0.38)	66.09 (6.32)	67.52 (5.90)
3	55.01 (5.13)	48.59 (12.63)	1.79 (-0.66)	58.16 (11.91)	43.99 (8.76)	62.63 (5.12)	49.74 (6.50)	1.65 (-0.35)	44.07 (2.45)	44.64 (3.38)

TABLE II: Performance on forget data 50 rounds after the exclusion of the client that requested unlearning. The values represent the absolute difference in performance compared to the retrained model (indicated in parentheses) at round 200. The ID column represents the identity of the client that leaves the federation.

questions about its reliability as an evaluation measure.

Table I presents aggregated results from 10 independent experiments, with each client of the federation taking a turn as the unlearning client. The values reported in the table expressing a difference in performance (the lower the difference, the better the forgetting). For clarity, we report a subset of values corresponding to rounds 1, 10, and 12. As shown, while slowly decreasing the gap with the retrained model values, the tracked forgetting metrics consistently remain distant.

Table II summarizes the tracked metrics after 50 rounds for each of the four unlearning scenarios. Notably, even 50 rounds after the target client has left (representing 25% of the original number of training rounds), the forgetting metrics remain far from the values of the retrained model.

D. Takeaways

Impact of Data Heterogeneity. In the presence of data heterogeneity, the influence of a specific client can persist long after its detachment from the federation. Even after 50 rounds under the natural baseline, forgetting metrics can still clearly indicate the participation of the client that left.

Effectiveness of Plain Accuracy. Plain accuracy on forget data proves to be a robust and reliable metric for evaluating unlearning effectiveness. In both the CIFAR-10 and CIFAR-100 experiments, fluctuations in the accuracy on forgotten data align closely with trends observed in other metrics, such as loss and MIA success rates, with the exception of KL divergence.

Inconsistency in KL Divergence. The KL divergence between random outputs and the model outputs on forget data does not consistently align with other forgetting metrics, raising questions about its reliability as an evaluation measure.

Need for Client Unlearning Mechanisms. Dedicated mechanisms for client unlearning are essential to ensure effective and efficient removal of client contributions from the model. Our experiments show that, at least in a cross-silo federated setting, without such mechanisms the client's right to be forgotten cannot be properly ensured.

IV. Related Work

Empirical Evaluations of the Natural Baseline. The study in [17] emphasizes the need for client unlearning, supported by some empirical findings. However, it assumes a homogeneous data distribution across clients and does not compare forgetting metrics with those of a retrained model. In contrast, our work provides a more detailed and comprehensive empirical analysis. Similarly, [13] demonstrates that a client's contribution can persist long after detachment. However, this study does not utilize membership inference attacks and only explores a single scenario of unlearning a random client. Additionally, the experimental setup involves partial client participation and a larger number of devices. In comparison, we focus

on cross-silo settings, which exhibit stronger memorization effects, enabling a more thorough investigation of unlearning dynamics.

Client Unlearning Methods. A few methods have been proposed to address client unlearning in federated learning. FedEraser [24] eliminates client contributions by recalibrating historical updates stored from previous rounds. FUKD [25] utilizes Knowledge Distillation after removing historical updates to accelerate performance recovery. The method in [15] performs client unlearning directly on the device, employing projected gradient ascent to remove the influence of forgotten data. FAST [26] primarily addresses backdoor removal, which shares conceptual similarities with client unlearning. However, none of these works thoroughly examines the implications of forgoing an active unlearning routine before initiating recovery, thus leaving a critical gap. In this perspective, this paper provides the readers with an original and valuable contribution to the better understanding of the problem and of its quantitative effects.

V. Conclusive Remarks

In this paper, we conducted an extensive empirical evaluation to assess whether explicit unlearning methods are necessary to effectively remove a client's contribution following an unlearning request. Our findings demonstrate that, under conditions of data heterogeneity, the influence of a specific client remains detectable and can persist long after the client exits the cooperative FL environment. These experimental results underscore the critical need for robust client unlearning mechanisms in FL. In future work, we propose a deeper investigation into the unlearning dynamics of the natural baseline, particularly in cross-device settings, to better understand its limitations and potential efficiency improvements. In addition, we are currently working on the determination of application-specific conditions (e.g., allowed time to make a client's contribution to FL undetectable and to what extent) to dynamically decide whether unlearning techniques should be applied in order to minimize the degradation of the global model and the FL-associated overhead.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[2] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329*, 2019.

[3] Carlo Mazzocca, Nicolò Romandini, Michele Colajanni, and Rebecca Montanari. FRAMH: A Federated Learning Risk-Based Authorization Middleware for Healthcare. *IEEE Transactions on Computational Social Systems*, 10(4):1679–1690, 2023.

[4] Sarthak Pati, Ujjwal Baid, Brandon Edwards, Micah Sheller, Shih-Han Wang, G Anthony Reina, Patrick Foley, Alexey Gruzdev, Deepthi Karkada, Christos Davatzikos, et al. Federated learning enables big data for rare cancer boundary detection. *Nature communications*, 13(1):7346, 2022.

[5] Samaneh Mohammadi, Ali Balador, Sima Sinaei, and Francesco Flammini. Balancing privacy and performance in federated learning: A systematic literature review on methods and metrics. *Journal of Parallel and Distributed Computing*, 192:104918, 2024.

[6] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council, 2016.

[7] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.

[8] Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 241–257. ACM, 2019.

[9] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2615–2632, 2021.

[10] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[11] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

[12] Paolo Bellavista, Luca Foschini, and Alessio Mora. Decentralised Learning in Federated Deployment Environments: A System-Level Survey. *ACM Computing Surveys (CSUR)*, 54(1):1–38, 2021.

[13] Nicolò Romandini, Alessio Mora, Carlo Mazzocca, Rebecca Montanari, and Paolo Bellavista. Federated unlearning: A survey on methods, design guidelines, and evaluation metrics. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2024.

[14] Ziyao Liu, Yu Jiang, Jiyuan Shen, Minyi Peng, Kwok-Yan Lam, Xingliang Yuan, and Xiaoning Liu. A survey on federated unlearning: Challenges, methods, and future directions. 2024.

[15] Anisa Halimi, Swanand Ravindra Kadhe, Ambrish Rawat, and Nathalie Baracaldo Angel. Federated unlearning: How to efficiently erase a client in fl? In *International Conference on Machine Learning*, 2022.

[16] Alessio Mora, Lorenzo Valerio, Paolo Bellavista, and Andrea Passarella. Fedquit: On-device federated unlearning via a quasi-competent virtual teacher. *arXiv preprint arXiv:2408.07587*, 2024.

[17] Xiangshan Gao, Xingjun Ma, Jingyi Wang, Youcheng Sun, Bo Li, Shouling Ji, Peng Cheng, and Jiming Chen. Verifi: Towards verifiable federated unlearning. *arXiv preprint arXiv:2205.12709*, 2022.

[18] Chen Wu, Sencun Zhu, and Prasenjit Mitra. Federated unlearning with knowledge distillation. *arXiv preprint arXiv:2201.09441*, 2022.

[19] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[20] Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the global knowledge by not-true distillation in federated learning. In *Advances in Neural Information Processing Systems*, 2022.

[21] Alessio Mora, Armir Bujari, and Paolo Bellavista. Enhancing generalization in federated learning with heterogeneous data: A comparative literature review. *Future Generation Computer Systems*, 2024.

[22] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[23] Jinkyu Kim, Geeho Kim, and Bohyung Han. Multi-level branched regularization for federated learning. In *International Conference on Machine Learning*, pages 11058–11073. PMLR, 2022.

[24] Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pages 1–10, 2021.

[25] Chen Wu, Sencun Zhu, and Prasenjit Mitra. Federated unlearning with knowledge distillation. *arXiv preprint arXiv:2201.09441*, 2022.

[26] Xintong Guo, Pengfei Wang, Sen Qiu, Wei Song, Qiang Zhang, Xiaopeng Wei, and Dongsheng Zhou. FAST: Adopting Federated Unlearning to Eliminating Malicious Terminals at Server Side. *IEEE Transactions on Network Science and Engineering*, pages 1–14, 2023.