TAPE: Tailored Posterior Difference for Auditing of Machine Unlearning

Anonymous Author(s)*

ABSTRACT

Increasing studies focus on machine unlearning as it upholds users' right to be forgotten, under which individuals can request the removal of their specified samples from trained models. However, the auditing of machine unlearning processes remains significantly underexplored. Although some existing methods offer unlearning auditing by leveraging backdoors, these backdoor-based approaches are inefficient and impractical, as they necessitate involvement in the initial model training process to embed the backdoors. In this paper, we propose a TAilored Posterior diffErence (TAPE) method to provide unlearning auditing independently of original model training. We observe that the process of machine unlearning inherently introduces changes in the model, which contains information related to the erased data. TAPE leverages unlearning model differences to assess how much information has been removed through the unlearning operation. Firstly, TAPE mimics the unlearned posterior differences by quickly building unlearned shadow models based on first-order influence estimation. Secondly, we train a Reconstructor model to extract and evaluate the private information of the unlearned posterior differences to audit unlearning. Existing privacy reconstructing methods based on posterior differences are only feasible for model updates of a single sample. To enable the reconstruction effective for multi-sample unlearning requests, we propose two strategies, unlearned data perturbation and unlearned influence-based division, to augment the posterior difference. Extensive experimental results indicate the significant superiority of TAPE over the state-of-the-art unlearning verification methods, at least 4.5× efficiency speedup and supporting the auditing for broader unlearning scenarios.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → **Machine learning**;

KEYWORDS

Machine Unlearning, Data Privacy, Unlearning Auditing.

ACM Reference Format:
Anonymous Author(s). 2018. TAPE: Tailored Posterior Difference for Auditing of Machine Unlearning. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Rising concerns over personal data privacy have led to the enactment of stringent privacy regulations and laws, such as the General Data Protection Regulation (GDPR) [26]. These legal frameworks guarantee individuals the "right to be forgotten", granting the right to request the removal of their data when participating in machine learning (ML) services. This right has sparked significant interest in the research community, giving rise to the concept of "machine unlearning" — a field that explores methods for erasing the influence of user-specified samples from trained ML models [5, 27, 42]. Although many unlearning techniques are proposed, most of them focus on developing unlearning optimization algorithms while ignoring the provision of unlearning auditing.

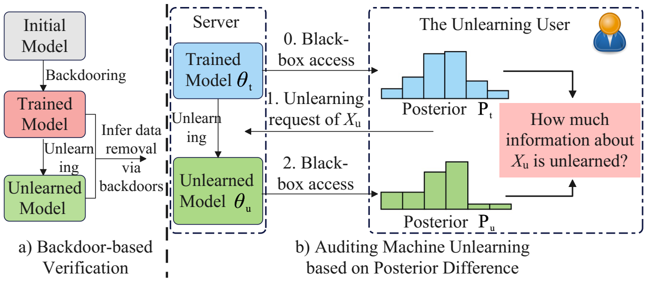
Research Gap. There are a few works provided unlearning execution verification based on backdoor techniques [16, 17, 36]. However, the backdoor-based methods have two oblivious disadvantages: (1) they are inefficient in practice as they are required to backdoor the model in the original model training period; (2) they cannot provide exact verification for genuine samples.

First, the efficacy of backdoor-based unlearning verification schemes hinges on model backdooring during the initial model training process [16, 17], as shown in Figure 1(a), which is impractical and inefficient. Users are unlikely to foresee the need to unlearn specific samples at the outset, making it unreasonable to incorporate tailored backdoors for specified samples during the initial training phase. Furthermore, involving the model training process in this way introduces inefficiencies, as it would be more effective to design an audit method that focuses solely on the machine unlearning operation and remains independent of the initial training process.

Second, the backdooring method can only build the connection between backdoored samples and models, but the backdoored samples and erased genuine samples are distinct datasets [10, 31]. These two datasets behave differently during model training, especially in approximate unlearning [9, 28], where the model accuracy on backdoored samples diminishes much faster than that on genuine samples [31, 39]. It indicates that the removal of backdoors can only verify whether the backdoored samples are unlearned from the model rather than genuine samples.

Research Question. Based on the research gap, we pose the research question: *"When an unlearning request is uploaded and processed, can we provide a practical audit service that verifies data removal and assesses the effectiveness of unlearning?"* Specifically, for practicality, the audit should only involve the unlearning process, and for effectiveness, it should rigorously determine whether the specified data has been unlearned and evaluate how much information has been erased from the model.

Motivation. ML models learn patterns and relationships from the training dataset, which are embedded in the model's parameters and behavior [13, 21]. The process of machine unlearning inevitably



a) Backdoor-based Verification

b) Auditing Machine Unlearning based on Posterior Difference

Figure 1: (a) The backdoor-based verification and (b) The motivation of auditing unlearning effectiveness based on the posterior difference. The scheme (b) only involves the unlearning process rather than the initial model training.

results in two versions of the model: one before and one after unlearning [6]. The difference between these models encapsulates the privacy information of the erased samples [6, 19, 44]. Our approach to audit unlearning effectiveness is based solely on analyzing these model differences, as illustrated in Figure 1(b). Auditing unlearning effectiveness based on model differences offers two key advantages. First, this method is practical and efficient, as it focuses exclusively on the unlearning operations without requiring involvement in the initial model training process. Second, auditing based on model differences supports broader unlearning scenarios and requests because unlearning for either genuine or backdoored samples results in model differences. By contrast, backdoor-based methods are only effective for backdoored samples as they can only build connections between backdoored samples and models.

Our Work. In this paper, we address the research question by formalizing the machine unlearning auditing problem and introducing an approach called TAPE, designed to audit unlearning effectiveness solely based on the unlearning process. TAPE contributes one method and two strategies to effectively train a Reconstructor model to evaluate how much private information is unlearned to audit this unlearning update. As a preparatory step, TAPE mimics unlearning posterior differences as input data for the Reconstructor by proposing an unlearned shadow model establishment method based on first-order influence estimation. While existing privacy reconstruction methods are effective for single samples, they are impractical for multiple samples–a common scenario in unlearning requests. To address this, we leverage the fact that the unlearning user knows and uploads the erased data, allowing us to design two strategies to ensure our method is suitable for multi-sample scenarios. Specifically, we design the unlearned data perturbation strategy to augment the posterior difference for a better reconstruction effect of unlearned samples. Additionally, we develop an unlearned influence-based division strategy, which transforms the reconstruction task from dealing with multiple samples as a single posterior difference to reconstructing each sample individually based on multiple divided posterior differences, significantly enhancing the overall reconstruction effectiveness.

We conduct extensive experiments on four representative datasets and four mainstream unlearning benchmarks to evaluate the proposed method, in which the results indicate the superiority of TAPE over the start-of-the-art auditing methods [16, 17] in terms of both efficiency and efficacy. From the efficiency perspective, our TAPE method achieves at least 4.5× speedup on all datasets and at most 75× speedup on the CelebA dataset than backdoor-based methods, as TAPE only involves the unlearned process. In contrast, backdoor-based methods must backdoor the service model during the initial training process, which is computationally expensive. From the efficacy perspective, our TAPE provides effective auditing of genuine samples for both exact and approximate unlearning algorithms, while the verification of backdoor-based methods only targets backdoored samples.

Our contributions are summarized as follows:

- This paper is the *first* to investigate the auditing for machine unlearning involves only the unlearning process, which is much different from existing backdoor-based methods that rely on backdooring the model during the initial training process. Moreover, our auditing study is feasible for genuine unlearned samples rather than only backdoor-marked samples.
- We propose a TAPE method based on the posterior difference to auditing unlearning. TAPE introduces a novel method to quickly establish unlearned shadow models that mimic the posterior differences and incorporates two posterior augmentation strategies to facilitate auditing the unlearning of multiple samples.
- We conduct extensive experiments on both exact and approximate unlearning methods across representative datasets and various model architectures. The findings validate significant improvements in efficiency and broader applicability to adaptive unlearning scenarios compared with the state-of-the-art unlearning verification methods.
- The source code and the artifact of TAPE is released at https://anonymous.4open.science/r/TAPE-30D0, which creates a new tool for measuring the effectiveness of machine unlearning methods, shedding light on the design of future unlearning auditing methods.

2 RELATED WORK

Few studies paid attention to the problem of providing unlearning audits to prove whether users' data are removed and how much information is unlearned [37]. The backdoor-based solutions [11, 16, 17, 36] provided data removal verification for machine unlearning. These studies mixed backdoored samples to users' data for backdooring servers' service ML models during the model training process. Then, they inferred whether the users' data was unlearned by testing if the backdoor disappeared from the service models [16, 17]. However, these backdoor-based methods have two oblivious limitations.

First, these methods rely on the original ML model training process, which is impractical in real-world scenarios due to users being unaware of which samples will need to be unlearned in the future, as well as the high computational costs involved. Second, these methods only build the connection between the backdoored samples and models, while the backdoored dataset and genuine dataset are still separate from the models' perspective [39, 43]. The backdoored and the erased genuine datasets perform differently during model training, as the corresponding experimental results are shown in Figure 2. The removal of backdoor triggers can actually only verify whether the backdoored samples are unlearned as

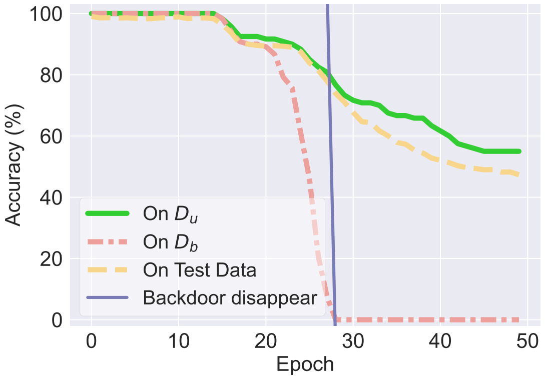


Figure 2: Approximate unlearning process on genuine unlearned data D_u and backdoored data D_b on MNIST. During unlearning, the backdoor accuracy drops to 0% at the blue Vertical line. Meanwhile, the model accuracy on genuine unlearned data D_u and test data is still around 80%**.**

the backdoored samples and unlearned genuine samples perform differently during the approximate unlearning training process [28]. When the accuracy of backdoored samples drops to 0%, the model accuracy on genuine unlearned data and test data is still around 80%. These results are consistent with current backdoor studies [10, 31, 40]. We present more detailed discussion in Appendix A.

3 PRELIMINARY AND PROBLEM STATEMENT

To facilitate the understanding of the unlearning auditing problem, we first introduce the main process of unlearning. A detailed introduction about the and threat model is presented in Appendix B. **Machine Unlearning.** The unlearning process usually includes the following phases. (1) The server trained a model with parameters θ_t derived from dataset D. (2) The unlearning user uploads the unlearning requested dataset D_u to the server for unlearning. (3) The server conducts an unlearning algorithm $\mathcal{U}$ to remove D_u's contribution from θ_t and results in an unlearned model with parameters $\theta_{u,D\backslash D_u}$, also denoted as θ_u.

Most existing backdoor-based unlearning verification methods tried to solve data removal verification but can only answer if the backdoored samples are unlearned. Answering whether the backdoored data is or not deleted is insufficient for trustworthy unlearning auditing. We should assess the unlearning effectiveness of the model, i.e., how much private information about the requested unlearning samples is removed from the model.

Problem Statement (Unlearning Effectiveness Audit). *Given the described unlearning scenario, the potential for unlearning execution spoofing by the server, and the capabilities of the unlearning user, auditing unlearning effectiveness necessitates a method for unlearning users to evaluate the extent to which information about D_u has been unlearned from θ_t to θ_u.*

It is important to note that the problem statement inherently includes the issue of data removal verification. If one can effectively measure how much information related to the erased samples has been unlearned, this measurement can serve as the basis for determining whether the data has been properly unlearned. We try to conduct unlearning auditing based on the unlearning updated posterior difference as it contains essential information about the erased samples. To achieve the auditing goals, we need to mimic the unlearning posterior difference and extract and quantify the unlearned information from it. We utilize the model's output layer results of the original and unlearned models on the user's local dataset to generate the posterior difference. We define the unlearning posterior difference as follows.

Posterior Difference. The unlearning user first queries the trained ML model θ_t before unlearning with all samples of D_{local} and concatenates the received outputs to form a vector $\hat{Y}_{t,local}$. Then, the user queries the unlearned model θ_u with samples in the D_{local} and creates a vector $\hat{Y}_{u,local}$. In the end, the user sets the posterior difference, denoted by δ, to the difference of both outputs:

$$\delta = \hat{Y}_{t,local} - \hat{Y}_{u,local}. \tag{1}$$

Note that the dimension of δ is the product of D_{local}'s cardinality and the number of classes of the target dataset. For example, in this paper, CIFAR-10 and MNIST are 10-class datasets, while we just identify the gender attributes of CelebA, which is a binary classification. As we set the local dataset 0.5% of CIFAR-10 and MNIST, and 0.06% of CelebA, this indicates the dimension of δ is 2500 for CIFAR-10, 3000 for MNIST, and 1210 for CelebA.

Unlearned Information Reconstruction to Assess How Much Information is Unlearned. To assess the unlearning effectiveness, we employ a reconstructor model to extract the unlearned information from the posterior difference. We employ the cosine similarity between the reconstructed and original unlearned samples to assess how much information of the unlearned information can be recovered from the unlearning update:

$$\textbf{Rec. Similarity:} \qquad \text{sim}(\hat{X}_u, X_u) = \frac{\hat{X}_u \cdot X_u}{\|\hat{X}_u\| \cdot \|X_u\|}. \tag{2}$$

Here, $\hat{X}_u \cdot X_u$ is the dot product of the reconstructed vectors $\hat{X}_u$ and original unlearned samples vectors X_u. $\|\hat{X}_u\|$ and $\|X_u\|$ are the Euclidean norms of the two vectors. A higher reconstruction similarity means more information about the erased samples is unlearned from the model.

4 TAPE METHODOLOGY

4.1 Overview of the TAPE

We illustrate the overview methodology process in Figure 3, which includes two main steps.

Unlearned Shadow Model Building. In this step, we propose a method to quickly build the unlearned shadow models with only the user's samples. Our method utilizes the first-order influence estimation function to effectively estimate the unlearning influence and remove it from the original model, thus quickly mimicking the unlearned model to generate the posterior differences.

Reconstructor Training. We then train a reconstructor model to evaluate how much information about the erased samples is unlearned. An unlearned data perturbation strategy and an unlearned influence-based division strategy are proposed to augment the posterior differences for reconstruction for multiple samples. Both strategies are implemented utilizing the advantage that the unlearning user knows and prepares the unlearned samples.

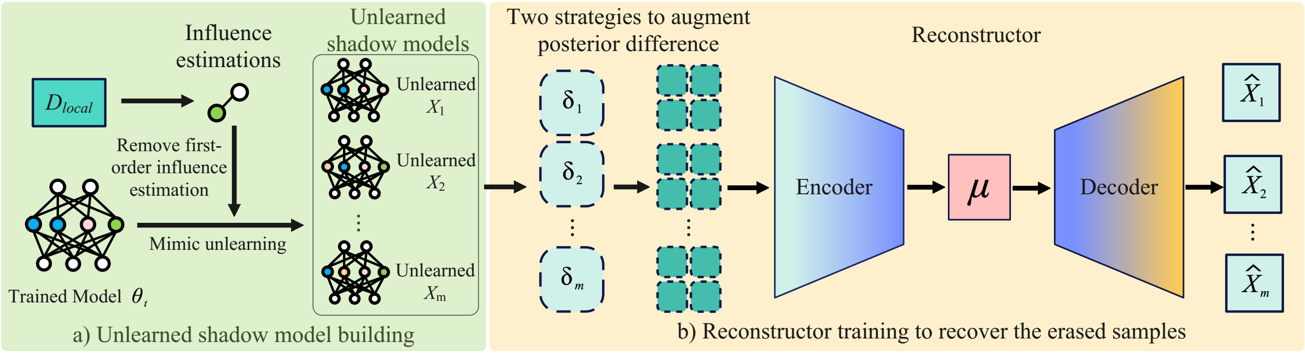


Figure 3: The main process of the TAPE method. (a) The first part quickly builds the unlearned shadow models through first-order influence estimation based on the user's local dataset D_{local} to mimic the unlearning posterior difference δ. (b) Two posterior difference augment strategies are proposed to make the reconstruction suitable for multi-sample unlearning.

4.2 Constructing Unlearned Shadow Model to Mimic Posterior Difference

The unlearning user possesses a local dataset D_{local}, including the unlearned data D_u, which was once used to train the ML service model. Now, the user wants to unlearn D_u from the ML service model and verify the unlearning effectiveness. As the unlearning verification is executed on the unlearning user side, the user can utilize the local dataset D_{local} to construct unlearned shadow models to mimic posterior differences. Many existing machine unlearning algorithms rely on the assistance of the remaining dataset $D\backslash D_u$. Only VBU [28] can implement unlearning with solely the unlearned samples; however, it is only suitable for Bayesian models.

Constructing Unlearned Shadow Model. We propose a method based on the influence function theory [1, 3, 22] in ML to quickly approximate an unlearned shadow model with only the unlearned data D_u. Specifically, when we remove D_u from a trained model θ_t for unlearning, the empirical risk minimization (ERM) can be written as:

$$\mathcal{L}_{D\backslash D_u}(\theta) = \frac{1}{n-m} \sum_{x \in D\backslash D_u} \ell(x;\theta), \tag{3}$$

where n is the size of D, m is the size of D_u, and $\ell(x;\theta)$ is the loss.

Similar to [3], we evaluate the effect of up-weighting a group of training samples on model parameters. Note that in this case, the updated weights must still form a valid distribution. Specifically, if a group of training samples is up-weighted, the weights of the remaining samples should be down-weighted to preserve the sum to one constraint of weights in the ERM formulation. We assume that the weights of samples in D_u have been up-weighted all by ϵ and use $\frac{m}{n}$ to denote the fraction of up-weighted training samples. This results in a down-weighting of the rest of the training data by $\tilde{\epsilon} = \frac{m}{n-m}\epsilon$, to preserve the empirical weight distribution of the training dataset. Then, the ERM can be translated as:

$$\mathcal{L}^{\epsilon}_{D\backslash D_u}(\theta) = \frac{1}{n}\Big(\sum_{x \in D\backslash D_u} (1-\tilde{\epsilon})\ell(x;\theta) + \sum_{x \in D_u} (1+\epsilon)\ell(x;\theta)\Big). \tag{4}$$

In the above equation, if $\epsilon = 0$, we get the original loss function $\mathcal{L}_{\emptyset}(\theta)$ (none of the training data points are unlearned) and if $\epsilon = -1$, we get the loss function $\mathcal{L}_{D\backslash D_u}(\theta)$ (specified samples are removed).

Let $\theta^{\epsilon}_{D\backslash D_u}$ denote the optimal parameters for $\mathcal{L}^{\epsilon}_{D\backslash D_u}$ minimization, and θ^* denote the optimal parameters trained on D. The unlearned shadow models can be approximately achieved by removing the estimated data influence from the trained model as follows,

$$\theta^{\epsilon}_{D\backslash D_u} = \theta_t - \frac{\epsilon}{n-m} \sum_{x_u \in D_u} \nabla \ell(x_u;\theta_t), \tag{5}$$

where $\epsilon \in [-1, 0]$ is used for unlearning, m is the size of the erased dataset and n is the size of the training dataset. $\Delta\theta \simeq -\frac{\epsilon}{n-m}\sum_{x_u \in D_u} \nabla \ell(x_u;\theta_t)$ is the estimaed data influence at current trained model θ_t. We omit the proof of the shadow model estimation in Eq. (5) as it is similar to the proofs in [3, 22]. Constructing the unlearned shadow model based on Eq. (5) only relies on the unlearned samples and is convenient for the user to implement.

Mimicking Posterior Difference. With the above method to construct unlearned shadow models, then, we can easily achieve the mimicked posterior differences. For instance, assuming the local dataset D_{local} contains m samples $X_1, X_2, ..., X_m$, we can construct m unlearned shadow models $\theta_{D\backslash X_1}, \theta_{D\backslash X_2}, ..., \theta_{D\backslash X_m}$ for each sample, where $\backslash X$ means unlearning the sample X. Based on these unlearned shadow models, we can mimic the corresponding unlearned posterior, $\hat{Y}_{\backslash X_1,local}, \hat{Y}_{\backslash X_2,local}, ..., \hat{Y}_{\backslash X_m,local}$, using the local dataset. The posterior difference can be calculated through Eq. (1), denoted as $\delta_1, \delta_2, ..., \delta_m$, as shown in Figure 3. Together with the corresponding shadow unlearning set's ground truth information, the training data for the reconstructor model to evaluate the unlearned information is derived.

4.3 Reconstructor Model Training with Two Strategies for Multiple Samples Auditing

Reconstructor Training for Unlearning Effectiveness Assessment. Like [34], we employ the autoencoder (AE) architecture to construct the Reconstructor, which includes an encoder and a decoder, as shown in Figure 3(b). Its goal is to learn an efficient encoding for the posterior differences δ. The encoder encodes the posterior difference into a latent vector μ, and the decoder decodes the corresponding latent vector to reconstruct the unlearned samples. We employ mean squared error (MSE) as the loss function to train the Reconstructor,

$$\mathcal{L}_{\mathsf{AE}} = ||\hat{X_u} - X_u||_2^2, \tag{6}$$

where $\hat{X_u} = \mathsf{AE}(\delta_u)$ is the reconstructed sample for X_u.

Existing studies [2, 19, 34] showed effective reconstruction for a single sample for the updated model difference. However, they are infeasible for reconstructing multiple samples. For unlearning effectiveness auditing, the unlearning user has the knowledge of the unlearned samples. With this advantage, we design two strategies: one augments posterior differences by perturbing unlearned data before unlearning and one augments posterior differences by individually dividing the posterior difference after unlearning, enabling evaluate how much information is unlearned for multiple samples.

Unlearned Data Perturbation before Unlearning. Many unlearning methods directly compare the posterior difference between $\hat{Y}_{t,D_u}$ and $\hat{Y}_{u,D_u}$ to evaluate the unlearning effectiveness. Usually, they treat a degradation of the accurate prediction probability as a sign of successful unlearning for the unlearned model [5, 42]. However, not all unlearning operations will cause a significant degradation of posterior probability on the unlearned samples. The model performance will remain, especially when there are some samples in the remaining dataset that are similar to the erased samples. It will also hinder the audit of unlearning effectiveness.

We propose an unlearned data perturbation method to augment posterior difference, assisting the unlearning effectiveness verification. Specifically, we hope to introduce a perturbation Δ^p to the unlearned sample X_u to augment the unlearned posterior for the reconstructor, so that it can effectively evaluate how much information is unlearned. At the same time, unlearning the perturbed specified data should maintain the unlearned model's utility on the remaining dataset. Since our final purpose is to improve the reconstructed information, we can formalize the unlearned data perturbation as follows to find the suitable perturbation.

$$\begin{aligned} &\min_{\Delta^p} \mathcal{L}_{\mathsf{AE}}(\hat{X_u}', X_u + \Delta^p) \\ &\text{s.t.} \quad \Delta^p \in \arg \min_{\theta_{\backslash(X_u+\Delta^p)}} \sum_{x \in D_r} \ell(x; \theta_{\backslash(X_u+\Delta^p)}) \end{aligned} \tag{7}$$

where $\hat{X_u}' = \mathsf{AE}(\delta_{\backslash(X_u+\Delta^p)})$, meaning that the samples are reconstructed based on the posterior difference that unlearns the perturbed data $X_u + \Delta^p$. We define the constraint that $\Delta^p : ||\Delta^p||_\infty \leq \alpha$ to ensure that the perturbed data will not be too different from the original data. We can combine these two losses together and treat them as two objectives, thus can be optimized with two-objective optimization methods [15, 32, 35]. During the perturbation optimization process, we fix the trained model θ_t and the reconstruction model AE. We only update the perturbation Δ^p of X_u to induce an augmented unlearned posterior difference $\delta_{\backslash(X_u+\Delta^p)}$, which improves the reconstruction effect. To find an effective perturbation, we can employ the restars technique, which is inspired from [12, 33], and we provide the corresponding algorithm in Appendix C.

Unlearning Influence-based Division after Unlearning. The unlearning influence-based division strategy utilizes the convenient properties of the first-order data influence estimation. After achieving the overall posterior differences for multiple samples $\delta_{\backslash D_u}$, the user can quickly estimate the basic data influence for each integrated sample $x_u \in D_u$, and we divide the overall posterior difference according to the weight of each sample's influence. We assume the divided posterior difference of the integrated sample obeys a Gaussian distribution:

$$\begin{aligned} \delta_{\backslash x_u} &\sim \mathcal{N}\Big(\frac{\delta_{\backslash D_u}}{\sum_{x_u \in D_u} \nabla \ell(x_u; \theta_t)} \cdot \nabla \ell(x_u; \theta_t), \sigma^2\Big), \\ &\text{s.t. } \delta_{\backslash D_u} = \sum_{x_u \in D_u} \delta_{\backslash x_u}, \end{aligned} \tag{8}$$

where we keep the divided posterior difference values as the mean and add a random deviation to it; meanwhile, we keep the sum of all the split slice posterior differences $\sum_{x_u \in D_u} \delta_{\backslash x_u}$ equal to the overall posterior difference $\delta_{\backslash D_u}$. Thus, we ensure every reconstruction has a unique divided posterior difference without additional change or noise in the original $\delta_{\backslash D_u}$. This operation changes the reconstruction task for multiple samples based on the same $\delta_{\backslash D_u}$ as reconstructing every single sample of D_u based on multiple divided posterior differences.

5 PERFORMANCE EVALUATION

5.1 Settings

Datasets. We conducted experiments on four widely adopted public datasets: MNIST [8], CIFAR10 [23], STL-10 [7], and CelebA [25]. These datasets offer a range of objective categories with varying levels of learning complexity, and the corresponding statistics are listed and introduced in Appendix D.

Models. In our experiments, we select a 5-layer multi-layer perceptron (MLP) connected by ReLU, a 7-layer convolutional neural network (CNN), and ResNet-18. Specifically, we use two 5-layer MLP models, one as the encoder and one as the decoder, to consist of the reconstructor for MNIST. We use two 7-layer CNNs to consist of the reconstructor for CIFAR10, STL-10, and CelebA. The main structure of the ML service model is implemented with a ResNet-18. Moreover, to align with existing backdoor-based verification methods, we also train a Verifier model using a 5-layer MLP model, which is trained after reconstruction. The Verifier model training algorithm is presented in Appendix E.

Metric. We use three metrics, model accuracy, reconstruction similarity, and verifiability, to measure the ability previously defined for the unlearning auditing scheme. Moreover, we use the running time to assess the methods' efficiency. We briefly summarize the metric as follows.

- **Accuracy.** Model accuracy evaluates functionality preservation and shows whether the auditing methods influence the utility of the service model.
- **Reconstruction Similarity.** It evaluates how much information about the specified samples is unlearned by reconstructed cosine similarity, as introduced in Eq. (2).
- **Verifiability.** Verifiability is used to measure the data removal verification by calculating the correct classifying rate of the Verifier, which is defined in Eq. (9) in Appendix F.
- **Running Time.** It is used to assess the efficiency, which records the running time of the entire process of each method.

Compared Unlearning Verification Benchmarks. There are mainly three data removal verification solutions [16, 17, 36], all based on backdooring methods. We only compare our method with MIB [17] because MIB is the most popular and has the best verification effect among these three methods. Note that since these

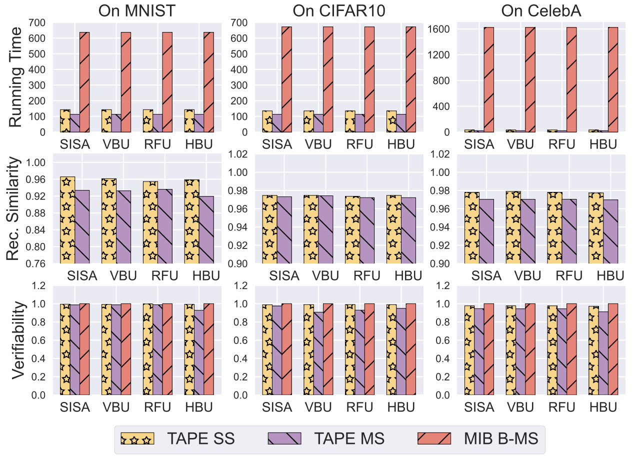


Figure 4: Auditing for different unlearning methods. TAPE consistently achieves significant efficiency improvement and a better unlearning auditing effect for a single sample (SS) than for multiple samples (MS).

methods can only support verifying the backdoored samples, most of the evaluation for MIB is verifying for unlearning only backdoored samples; our method is verifying for unlearning genuine samples.

Unlearning Benchmarks. The evaluation for unlearning verification methods is conducted on four mainstream unlearning algorithms: SISA [5], HBU [14], VBU [28] and RFU [41].

5.2 Evaluations of Unlearning Auditing based on Various Unlearning Benchmarks

Setup. We demonstrate the evaluation of unlearning auditing methods on four mainstream unlearning benchmarks in Figure 4. We evaluate unlearning scenarios of both single-sample (SS) where the Erased Sample Size (*ESS*) is 1 and multi-sample (MS) where *ESS*=20. Since the MIB method is unable to verify only for unlearning genuine samples, we here evaluate the verification of MIB for backdoored multi-samples (B-MS), $D_b \leftarrow (X_b + \text{trigger}, Y_{target})$, which add a white block patch as the trigger at the right bottom of chosen images and change the corresponding labels for the backdooring target. When evaluating TAPE, to keep the setting similar to MIB, we add the perturbation with the same limit distance as the trigger patch to the genuine unlearned samples but do not change the labels for backdooring, $D_{u,p} \leftarrow (X_u + \Delta^p, Y_u)$, which is achieved through the unlearned data perturbation (UDP) method.

Evaluations of Auditing Efficiency. The first row of Figure 4 shows the running time of TAPE and MIB across different unlearning algorithms on MNIST, CIFAR-10, and CelebA. TAPE significantly outperforms MIB in terms of running time, as TAPE only involves the unlearning training process. The greatest speedup is observed on CelebA. Additionally, TAPE takes more time for single-sample unlearning auditing compared to multi-sample unlearning auditing, as training the reconstructor model on a single-sample level for a local dataset requires more time.

Evaluations of Unlearning Auditing Effect. The second row, which depicts reconstruction similarity, illustrates the evaluation results of how much information about the specified samples has been unlearned. As MIB is unable to measure the extent of information unlearned from the model, it is omitted in this row. Among all unlearning methods (SISA, VBU, RFU, and HBU), TAPE achieves better reconstruction similarity for single-sample unlearning than for multi-sample unlearning. This suggests that unlearning a single sample tends to reveal more information about the erased sample in the unlearning posterior difference.

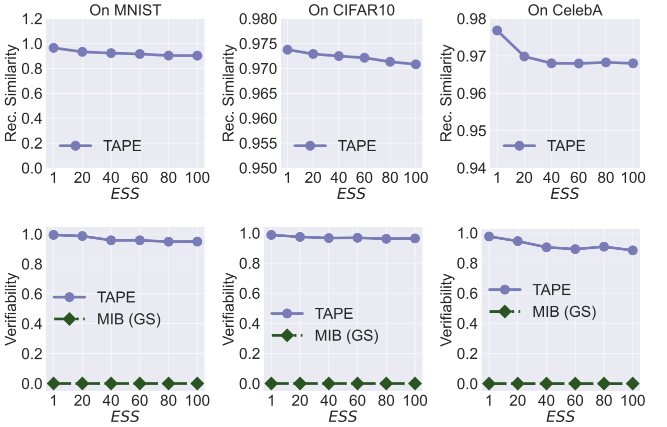


Figure 5: Evaluations of impact about different *ESS*. Here, we evaluate the unlearning verification of genuine samples (GS) rather than backdoored samples for MIB.

Table 1: Evaluation Results on CIFAR10 and STL-10

	CIFAR10, *ESS* = 20			STL-10, *ESS* = 2		
	Original	MIB [17]	TAPE	Original	MIB	TAPE
Running time (s)	644	673	**113**	781	809	**74.90**
Model Acc.	**81.62%**	79.13%	**81.62%**	**68.99%**	67.26%	**68.99%**
Rec. Sim.	-	-	**0.973**	-	-	**0.174**
Unl. Verifiability	0.00%	0.00%	**97.44%**	0.00%	0.00%	**84.40%**

The third row shows the comparison with MIB by the verifiability of data removal verification. All methods have a high verifiability result. It indicates that all unlearning methods are effective in these evaluations to answer if the samples are unlearned from the model. However, we should note that in the experiments, MIB only verifies the backdoored samples $D_{u,b}$, while TAPE can verify the genuine samples $D_{u,p}$, which has kept the original labels.

We also demonstrate an overall evaluation for MIB and TAPE on SISA [5] in Table 1. Here, we evaluate auditing genuine samples for both MIB and TAPE instead of setting backdoored samples for MIB. TAPE achieves effective auditing results as analyzed in Figure 4. However, the MIB cannot successfully verify the unlearning of any genuine samples in the Unl. Verifiability of Table 1. Moreover, since the TAPE scheme is independent of the original model training process, it will not influence the model utility of the original ML service model, keeping the same model accuracy as the "Original". We present additional experimental results in Appendix G.1.

5.3 Ablation Study of Erased Samples Size (*ESS*)

Setup. Figure 5 illustrates the impact of *ESS* when providing unlearning auditing on MNIST, CIFAR10 and CelebA. The largest

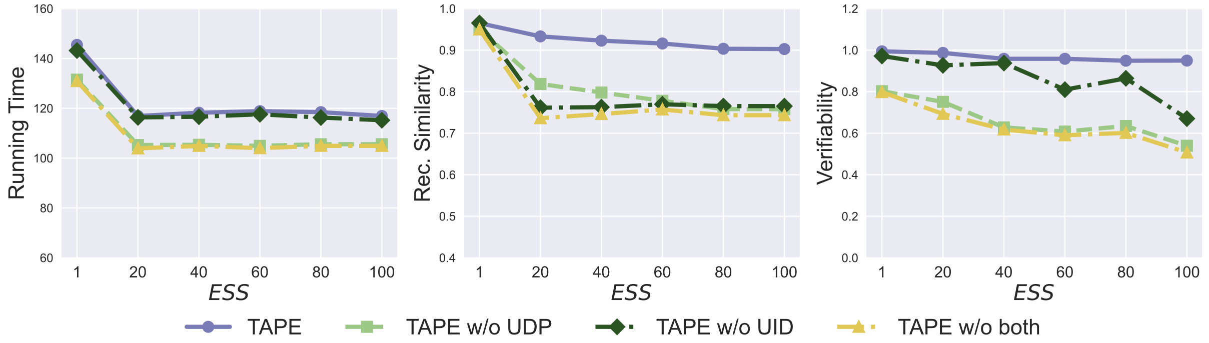


Figure 6: Ablation study about the unlearned data perturbation (UDP) and unlearning influence-based division (UID) strategies of TAPE on MNIST. The legends stand for the entire TAPE, TAPE without (w/o) the UDP strategy, TAPE w/o the UID strategy, and TAPE w/o both strategies.

ESS is 100 in this experiment, which is around 0.2% of training datasets in MNIST and CIFAR10. In practice, 0.2% data might be very large for unlearning, which has been analyzed in [4, 6]. To illustrate a better comparison, we verify the unlearning of genuine perturbed samples $D_{u,p}$ for both MIB and TAPE. The ablation study is conducted based on another representative unlearning method, VBU [28], to demonstrate the scalability of our method for different unlearning methods. Due to the page limitation, we present the impact on the efficiency of *ESS* in Appendix G.2.

Impact on Unlearning Auditing. Figure 5 shows the evaluation of both auditing how much information is unlearned (the first row) and data removal status verification (the second row) on MNIST, CIFAR10 and CelebA. When $ESS = 1$, TAPE can effectively provide the auditing of unlearning information and data removal status on all datasets. By contrast, MIB cannot support the data removal verification of genuine samples (black dotted line in the second row in Figure 5). Moreover, both two evaluation metrics have a decreasing trend when *ESS* increases.

5.4 Ablation Study of Two Strategies

We conduct an ablation study to evaluate our two designed strategies, unlearned data perturbation (UDP) and unlearning influence-based division (UID).

Setup. We conduct the experiments on MNIST in four situations. "TAPE" means the entire scheme with the two strategies. "TAPE w/o UDP" means TAPE without the UDP strategy while keeping the UID strategy, and "TAPE w/o UID" means that we remove the UID strategy of TAPE while keeping the UDP strategy. The "TAPE w/o both" means we remove both strategies of TAPE for auditing.

Impact on Efficiency. We present the efficiency evaluation in the first sub-figure of Figure 6. Since the UDP strategy introduces $R = 10$ restarts training to find perturbations for erased samples, it consumes around 10 seconds in TAPE. Compared with UDP, UID almost does not consume computational time, as the main time consumption is the unlearned shadow model building and reconstructor training.

Impact on Auditing Unlearning Effectiveness. The second sub-figure in Figure 6 shows the reconstruction similarity of different methods. All methods achieve a high reconstruction similarity when *ESS = 1*, which shows the effectiveness of TAPE in single-sample unlearning auditing even without the two strategies. However, when the posterior difference contains information of multiple samples, *ESS > 1*, if we don't have the UID strategy, the reconstruction similarity drops dramatically, showing as "TAPE w/o UID" and "TAPE w/o both". The UID plays a vital role in the reconstruction of multiple samples. While the UDP strategy also enhances the reconstruction quality, as shown in "TAPE w/o UDP", its impact is not as substantial as the UID strategy. By contrast, in the third sub-figure in Figure 6, the UDP strategy impacts the verifiability of data removal than the UID strategy, showing as "TAPE w/o UDP" and "TAPE w/o UID". These results show the significant improvement of the two strategies in benefiting unlearning auditing of how much information is unlearned and data removal status.

5.5 Detailed Ablation Study of UDP

We additionally evaluate the impact of unlearned data perturbation. We find that only perturbation without changing the labels as backdoor-based methods already significantly improves the information reconstruction and assists the verifiability of data removal.

Setup. In this experiment, we keep all other parameters fixed while only changing the perturbation limitation value α. As introduced in Section 4.3, the UDP outputs $D_{u,p} \leftarrow (X_u + \Delta^p, Y_u)$, and the perturbation is limited as $\|\Delta^p\|_\infty \leq \alpha$. We set the perturbation limit distance value from 0 to 25 on MNIST and CIFAR10, 0 to 75 on STL-10, and 0 to 150 on CelebA, which is determined by the data size. We only perturb the unlearned data but do not change the corresponding labels to ensure the utility of the genuine samples. To better illustrate the impact of α, we keep a copy of original data D_u in the remaining dataset, which is the hardest scenario for unlearning effectiveness auditing. The unlearning method employed here is the approximate unlearning method VBU. We evaluate our method in both single-sample (SS) and multi-sample (MS) unlearning requests. For the MIB method, we evaluate in the backdoored single-sample (B-SS) scenario, and we control the backdooring trigger patches with the sample distance limitation value as our methods to ensure they can be compared. The experimental results on MNIST, CIFAR10, STL-10, and CelebA are presented in Figure 7.

Impact on Efficiency. The first column illustrates the running time of TAPE and MIB, which clearly demonstrates the improvement of TAPE in terms of efficiency. The reason is that the verification models of TAPE are trained independently of the original ML service model training process, which significantly shortens the running

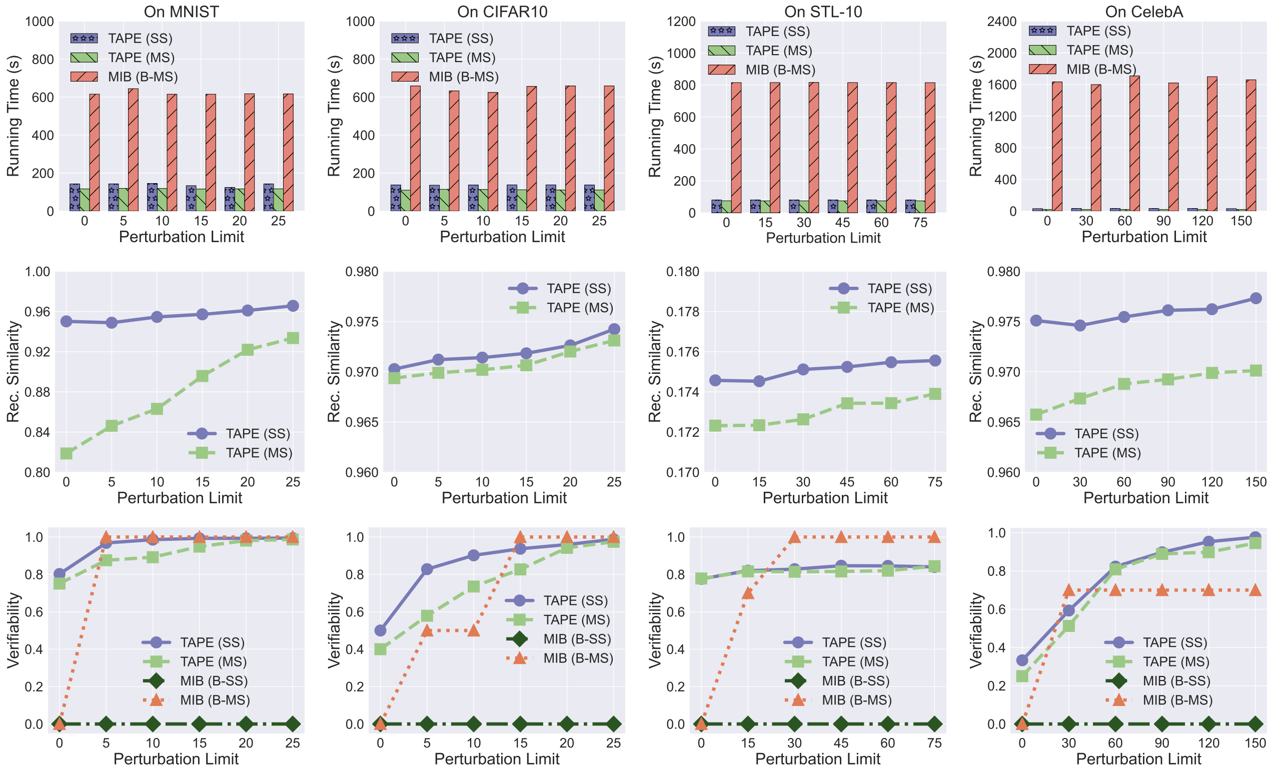


Figure 7: Evaluations of the impact of the unlearned data perturbation limit. "SS" stands for single-sample unlearning scenario, and "MS" means multi-sample unlearning scenario. "B-SS" means backdoored single-sample scenario, and "B-MS" means backdoored multi-sample unlearning scenario.

time for verification. During the experiment, we fix $R = 10$ restarts. The perturbation limitation has no significant impact on running time, as the running time remains consistent across different perturbation limits.

Impact on Unlearning Auditing Effect. With larger perturbations, we augment the unlearning posterior difference, increasing the reconstruction similarity and enabling more information about the erased samples to be extracted. The trend is indicated in unlearning single samples (the green line) and multi-samples (the blue line) on all three datasets. Moreover, the results in the second row in Figure 7 also clearly confirm our previous analysis: auditing for a single sample achieves a much better result than auditing for multiple samples, which is reflected in the significant gap between the two scenarios.

In TAPE, a larger perturbation of unlearned data makes data removal verification easier. In the third row in Figure 7, it is obvious that the verifiability increases as the perturbation limitation increases. When the perturbation limitation is less than 5, it is hard to distinguish if the samples $D_{u,p}$ are unlearned because the remaining dataset contains a similar original sample D_u. However, when the perturbation limitation increases larger than 5, the TAPE verifiability will greatly improve for genuine single-sample and multi-sample unlearning. MIB performs similarly when verifying the unlearning of backdoored multiple samples (B-MS). However, MIB fails to verify the unlearning of a single sample, as only one sample cannot backdoor the original ML service model.

6 SUMMARY AND FUTURE WORK

In this paper, we propose a TAPE scheme to investigate the auditing of unlearning effectiveness based on unlearning posterior differences, involving only the unlearning process. TAPE contributes a method to build unlearned shadow models to mimic the posterior difference quickly. Moreover, two strategies are introduced to augment the posterior difference, enabling the audit of unlearning multiple samples. The extensive experimental results validate the significant efficiency improvement compared with backdoor-based methods and the effectiveness of auditing genuine samples in both exact and approximate unlearning manners.

The auditing method proposed in this paper significantly addresses the limitations of existing unlearning verification methods. It effectively audits genuine samples for both exact and approximate unlearning methods in single-sample and multi-sample unlearning scenarios. Additionally, it eliminates the need for involvement in the original model training process. Future work could continue this line of inquiry, developing more efficient unlearning auditing methods to guarantee and support the right to be forgotten in MLaaS environments.

REFERENCES

[1] Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger B Grosse. 2022. If Influence Functions are the Answer, Then What is the Question? *Advances in Neural Information Processing Systems* 35 (2022), 17953–17967.

[2] Borja Balle, Giovanni Cherubin, and Jamie Hayes. 2022. Reconstructing training data with informed adversaries. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1138–1156.

[3] Samyadeep Basu, Xuchen You, and Soheil Feizi. 2020. On second-order group influence functions for black-box predictions. In *International Conference on Machine Learning*. PMLR, 715–724.

[4] Theo Bertram, Elie Bursztein, Stephanie Caro, Hubert Chao, Rutledge Chin Feman, Peter Fleischer, Albin Gustafsson, Jess Hemerly, Chris Hibbert, Luca Invernizzi, et al. 2019. Five years of the right to be forgotten. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 959–972.

[5] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 141–159.

[6] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2021. When machine unlearning jeopardizes privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 896–911.

[7] Adam Coates, Andrew Ng, and Honglak Lee. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 215–223.

[8] Li Deng. 2012. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine* 29, 6 (2012), 141–142.

[9] Shaopeng Fu, Fengxiang He, and Dacheng Tao. 2022. Knowledge Removal in Sampling-based Bayesian Inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=dTqOcTUOQO

[10] Kuofeng Gao, Yang Bai, Jindong Gu, Yong Yang, and Shu-Tao Xia. 2023. Backdoor Defense via Adaptively Splitting Poisoned Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4005–4014.

[11] Xiangshan Gao, Xingjun Ma, Jingyi Wang, Youcheng Sun, Bo Li, Shouling Ji, Peng Cheng, and Jiming Chen. 2024. Verifi: Towards verifiable federated unlearning. *IEEE Transactions on Dependable and Secure Computing* (2024).

[12] Jonas Geiping, Liam H. Fowl, W. Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. 2021. Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[14] Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. 2020. Certified Data Removal from Machine Learning Models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 3832–3842.

[15] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. 2020. Learning to branch for multi-task learning. In *International Conference on Machine Learning*. PMLR, 3854–3863.

[16] Yu Guo, Yu Zhao, Saihui Hou, Cong Wang, and Xiaohua Jia. 2023. Verifying in the Dark: Verifiable Machine Unlearning by Using Invisible Backdoor Triggers. *IEEE Transactions on Information Forensics and Security* (2023).

[17] Hongsheng Hu, Zoran Salcic, Gillian Dobbie, Jinjun Chen, Lichao Sun, and Xuyun Zhang. 2022. Membership Inference via Backdooring. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, Luc De Raedt (Ed.). ijcai.org, 3832–3838.

[18] Hongsheng Hu, Shuo Wang, Jiamin Chang, Haonan Zhong, Ruoxi Sun, Shuang Hao, Haojin Zhu, and Minhui Xue. [n.d.]. A Duty to Forget, a Right to be Assured? Exposing Vulnerabilities in Machine Unlearning Services. *31th Annual Network and Distributed System Security Symposium, NDSS 2024* ([n. d.]).

[19] H. Hu, S. Wang, T. Dong, and M. Xue. 2024. Learn What You Want to Unlearn: Unlearning Inversion Attacks against Machine Unlearning. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 262–262. https://doi.org/10.1109/SP54263.2024.00182

[20] Yuke Hu, Jian Lou, Jiaqi Liu, Feng Lin, Zhan Qin, and Kui Ren. 2024. ERASER: Machine Unlearning in MLaaS via an Inference Serving-Aware Approach. *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security* (2024).

[21] Vahid Janfaza, Kevin Weston, Moein Razavi, Shantanu Mandal, Farabi Mahmud, Alex Hilty, and Abdullah Muzahid. 2023. Mercury: Accelerating dnn training by exploiting input similarity. In *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 638–650.

[22] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*. PMLR, 1885–1894.

[23] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[24] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. 2020. Composite backdoor attack for deep neural network by mixing existing benign features. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 113–131.

[25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2018. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August* 15, 2018 (2018), 11.

[26] Alessandro Mantelero. 2013. The EU Proposal for a General Data Protection Regulation and the roots of the 'right to be forgotten'. *Comput. Law Secur. Rev.* 29, 3 (2013), 229–235.

[27] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. 2021. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*. PMLR, 931–962.

[28] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. 2020. Variational bayesian unlearning. *Advances in Neural Information Processing Systems* 33 (2020), 16025–16036.

[29] Tuan Anh Nguyen and Anh Tran. 2020. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems* 33 (2020), 3454–3464.

[30] Seong Joon Oh, Max Augustin, Mario Fritz, and Bernt Schiele. 2018. Towards Reverse-Engineering Black-Box Neural Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=BydjJte0-

[31] Minzhou Pan, Yi Zeng, Lingjuan Lyu, Xue Lin, and Ruoxi Jia. 2023. ASSET: Robust Backdoor Data Detection Across a Multiplicity of Deep Learning Paradigms. In *32nd USENIX Security Symposium (USENIX Security 23)*. 2725–2742.

[32] Fabrice Poirion, Quentin Mercier, and Jean-Antoine Désidéri. 2017. Descent algorithm for nonsmooth stochastic multiobjective optimization. *Computational Optimization and Applications* 68 (2017), 317–331.

[33] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. 2019. Adversarial Robustness through Local Linearization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 13824–13833.

[34] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. 2020. Updates-Leak: Data Set Inference and Reconstruction Attacks in Online Learning. In *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*. USENIX Association, 1291–1308.

[35] Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems* 31 (2018).

[36] David Marco Sommer, Liwei Song, Sameer Wagh, and Prateek Mittal. 2022. Athena: Probabilistic Verification of Machine Unlearning. *Proceedings on Privacy Enhancing Technologies* 3 (2022), 268–290.

[37] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. 2022. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*. 4007–4022.

[38] Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing hyperparameters in machine learning. In *2018 IEEE symposium on security and privacy (SP)*. IEEE, 36–52.

[39] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 707–723.

[40] Hang Wang, Zhen Xiang, David J Miller, and George Kesidis. 2023. Mm-bd: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 15–15.

[41] Weiqi Wang, Chenhan Zhang, Zhiyi Tian, and Shui Yu. 2024. Machine Unlearning via Representation Forgetting With Parameter Self-Sharing. *IEEE Transactions on Information Forensics and Security* 19 (2024), 1099–1111.

[42] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. 2024. Machine unlearning of features and labels. *31th Annual Network and Distributed System Security Symposium, NDSS 2024* (2024).

[43] Yi Zeng, Minzhou Pan, Hoang Anh Just, Lingjuan Lyu, Meikang Qiu, and Ruoxi Jia. 2023. Narcissus: A practical clean-label backdoor attack with limited information. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 771–785.

[44] Kaiyue Zhang, Weiqi Wang, Zipei Fan, Xuan Song, and Shui Yu. 2023. Conditional Matching GAN Guided Reconstruction Attack in Machine Unlearning. In *GLOBECOM 2023-2023 IEEE Global Communications Conference*. IEEE, 44–49.

Table 2: An overview of machine unlearning auditing methods.

Unlearning Auditing Methods	Involving Processes		Auditing Data Type		Unlearning Methods		Unlearning Scenarios	
	Original training and unlearning	Only unlearning process	Backdoored (marked) samples	Genuine samples	Exact unlearning	Approximate unlearning	Single sample	Multi samples
MIB [17]	●	○	●	○	●	○	○	●
Athena [36]	●	○	●	○	●	○	○	●
Verify in the dark [16]	●	○	●	○	●	○	○	●
Verifi [11]	●	○	●	○	●	○	○	●
TAPE (Ours)	○	●	○	●	●	●	●	●

●: the auditing method is applicable; ○: the auditing method is not applicable.

A DIFFERENCE FROM EXISTING STUDIES

Our TAPE approach is significantly different from existing unlearning verification methods [11, 16, 17, 36] in terms of the involving processes, auditing data type, unlearning scenarios, and unlearning methods, as depicted in Table 2. First, the significant difference is that the auditing of our method only involves the unlearning process, while the backdoor-based methods must involve both the original training and unlearning processes to ensure the service model first learns the backdoor. Second, most existing auditing methods are based on backdooring techniques and need to backdoor or mark samples for verification [11, 16, 17, 36]. As we analyzed in the above subsection, they can only validate the backdoored samples and are only applicable to the exact unlearning methods as exact unlearning methods guarantee the deletion from the dataset level. Our method does not mix any other data to the training dataset, and the auditing is based on the posterior difference, which is suitable for genuine samples in both exact and approximate unlearning methods. Third, backdoor-based auditing methods are only feasible for multi-sample unlearning scenarios because just using a single sample makes it hard to backdoor the model [24, 29, 39, 43], hence failing to provide unlearning verification for a single sample.

B MLAAS SCENARIO AND THREAT MODEL

Our problem is introduced in a simple machine unlearning as a service (MLaaS) scenario for ease of understanding. Under the MLaaS scenario, there are two main entities involved: an ML server that collects data from users, trains models, and provides the ML service, and users that contribute their data for ML model training.

The ML Server’s Ability. To uphold the “right to be forgotten” legislation and establish a privacy-protecting environment, the ML server is responsible for conducting machine unlearning operations. However, it is challenging to audit the unlearning effect for users to confirm that the unlearning is processed and prevent the spoof of unlearning from the ML server. In alignment with common unlearning verification settings [16, 17], we assume the ML server is honest for learning training but may spoof users for unlearning, i.e., it reliably hosts the learning process but may deceive users during unlearning operations by pretending unlearning has been executed when it has not. It is reasonable for the ML server to pretend to execute unlearning operations to avoid the degradation of model utility. Moreover, this assumption is more plausible than assuming the server will forge an unlearning update [37]. Forging an unlearning update would require the server to simulate the disappearance of specified data and the corresponding resulting in model utility degradation, which demands significant effort without any benefit, making it an unlikely motivation.

The Unlearning Users’ Ability. We consider the scenario where the unlearning user has only black-box access to the ML service model, which is one of the most challenging scenarios [19, 34]. In unlearning scenarios, the unlearning user possesses a local dataset, including the erased samples, which constitutes the entire training dataset for the ML service model [20, 42]; however, the user has no access to the entire dataset. This just allows the user to query the model with their own data in a black-box access to obtain the corresponding posteriors and design the unlearning requests with specific data for unlearning verification purposes. Furthermore, we assume the unlearning user knows the unlearning algorithms, which is confirmed by both server and users, commonly used in other works [18]. However, even if the unlearning user knows the algorithms, without the remaining dataset, the user still cannot achieve the corresponding unlearning results of most unlearning algorithms. To relax the difficulty, we consider the unlearning user to be able to establish the same ML model as the current target ML service model with respect to model architecture. This can be achieved through model hyperparameter stealing attacks [30, 34, 38]. The unlearning user leverages this knowledge to simulate the unlearned shadow models and mimic the behavior of the ML service model based on the designed unlearning requests, thereby deriving the posterior differences necessary for training the reconstruction model to evaluate the unlearning effectiveness.

C UNLEARNING DATA PERTURBATION (UDP) ALGORITHM

Algorithm 1 demonstrates how to use the R restarts to find the satisfied perturbation for the unlearning data to augment the posterior difference for auditing.

D DATASETS

The statistics of all datasets used in our experiments are listed and introduced in Table 3. MNIST, CIFAR10, and STL-10 are benchmark datasets utilized for 10-class image classification tasks, offering a range of objective categories with varying levels of learning complexity. Our experiment on CelebA is to identify the gender attributes of the face images. The task is a binary classification problem, different from the ones on MNIST, CIFAR10 and STL-10. We also introduce them below

Algorithm 1: Unlearning Data Perturbation (UDP)

```
Input: Trained model $\theta^*$, reconstruction model AE, unlearned data
       $X_u$, perturbation limit $\alpha$, local dataset $D_{local}$
Output: The perturbed unlearning data, $X'_u = X_u + \Delta^p$
procedure UDP($\theta^*$, AE, $X_u$, $\alpha$, $D_{local}$):
    for $r \leftarrow 1$ to $R$ restarts do
        $\Delta^p_r \leftarrow \mathcal{N}(0, 1)$    ▷ Initialize random perturbation.
        for $i \leftarrow 1$ to $m$ optimization steps do
            $X^p_{u,i} \leftarrow X_u + \Delta^p_r$  ▷ Add the perturbation to data.
            $\theta_{\backslash(X^p_{u,i})} \leftarrow \theta^* - \frac{\epsilon}{n-1}\nabla\ell(X^p_{u,i}; \theta^*)$ ▷ According to
              Eq. (5).
            $\delta^p_{u,i} \leftarrow \theta^*(D_{local}) - \theta_{\backslash(X^p_{u,i})}(D_{local})$ ▷ Calculate
              posterior difference according to Eq. (1).
            $\nabla\mathcal{L}_{\mathsf{AE}} \leftarrow \nabla\mathcal{L}_{\mathsf{AE}}(\mathsf{AE}(\delta^p_{u,i}), X^p_{u,i})$ ▷ According to Eq. (7).
            $\Delta^p_r \leftarrow \Delta^p_r - \eta\nabla\mathcal{L}_{\mathsf{AE}}(\mathsf{AE}(\delta^p_{u,i}), X^p_{u,i})$    ▷ Update
              perturbation with limitation $\|\Delta^p_r\|_\infty \leq \alpha$.
    Choose the optimal $\Delta^p_r$ with minimal value in $\mathcal{L}_{\mathsf{AE}}$ as $\Delta^{p*}$.
    return $X'_u = X_u + \Delta^{p*}$
```

Table 3: Dataset statistics.

Dataset	Feature Dimension	#. Classes	#. Samples
MNIST	28×28×1	10	70,000
CIFAR10	32×32×3	10	60,000
STL-10	96x96x3	10	5000
CelebA	178×218×3	2 (Gender)	202,599

- **MNIST.** MNIST contains 60,000 handwritten digit images for the training and 10,000 handwritten digit images for the testing. All these black and white digits are size normalized, and centered in a fixed-size image with 28 × 28 pixels.
- **CIFAR10.** CIFAR10 dataset consists of 60,000 32x32 colour images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images.
- **STL-10.** STL-10 dataset consists of 13,000 color images with 5,000 training images and 8,000 test images. STL-10 has 10 classes of airplanes, birds, cars, cats, dear, dogs, horses, monkeys, ships, and trucks with each image having a higher resolution of 96x96 pixels. Compared to the above two datasets, STL-10 can be considered as a more challenging dataset with higher learning complexity.
- **CelebA.** CelebA is a large-scale face attributes dataset with more than 200,000 celebrity images, each with 40 attribute annotations, and the size of each image is 178×218.

E THE VERIFIER TRAINING PROCESS

This Verifier aims to identify if the recovered samples are unlearned samples. Specifically, we first construct a verification dataset $D_{veri.}$. For each instance in the unlearned dataset, and the reconstructor model reconstructs based on the posterior difference of the instance, and we set the corresponding label equal to 1. We add it as the postive sample $(\mathsf{AE}(\delta_{\backslash x_u}), x_u; 1)$ into $D_{veri.}$ For each instance in the local dataset that is not part of the unlearned dataset, we set a negative label for the instance and the reconstructed sample pair, i.e. $(\mathsf{AE}(\delta_{\backslash x_u}), x_i; 0)$. These samples are added to the verification dataset too. A verifier model is then initialized and trained on this constructed dataset using a cross-entropy loss. The Verifier model training algorithm is presented in Algorithm 2.

Algorithm 2: Verifier Model Training (VMT)

```
Input: Reconstruction model AE, posterior differences $\delta$, local
       dataset $D_{local}$, unlearned dataset $D_u$
Output: The Verifier Model $\mathcal{V}$
procedure VMT(AE, $\delta$, $D_{local}$, $D_u$):
    Initialize a verification dataset $D_{veri.}$
    for $x_u$ in $D_u$, $x_i$ in $D_{local} \backslash D_u$ do
        $D_{veri.}$ adds the positive sample $(\mathsf{AE}(\delta_{\backslash x_u}), x_u; 1)$
        $D_{veri.}$ adds the negative sample $(\mathsf{AE}(\delta_{\backslash x_u}), x_i; 0)$
    Initialize a Verifier model $\mathcal{V}$
    Train $\mathcal{V}$ on the constructed $D_{veri.}$ using a cross entropy loss
    return the trained $\mathcal{V}$
```

F METRICS AND REQUIREMENTS FOR AUDITING

Data Removal Verifiability. Existing backdoor-based unlearning verification methods can only provide the data removal verifiability based on the backdoor attack success rate [16, 17]. We also train a Verifier (a classifying model) to identify the reconstructed data of the unlearned samples and the reconstructed data of the samples that still remain. We propose Verifiability to evaluate the accuracy of the Verifier, which calculates the correct classifying rate as

$$\textbf{Verifiability:} \qquad V = \frac{1}{m} \sum_{x_u \in D_u} \mathbb{I}(\text{Verifier}(\delta_u, x_u) = 1), \qquad (9)$$

where m is the size of the unlearned dataset D_u and $\mathbb{I}$ is the indicator function that equals 1 when its argument is true $(\text{Verifier}(\delta_u, x_u) = 1)$ and 0 otherwise.

G ADDITIONAL EXPERIMENTS

G.1 Overview Evaluation of TAPE

We first demonstrate the overview evaluation results of different unlearning auditing methods on MNIST, CIFAR10, STL-10 and CelebA, presented in Table 4. The upper half of Table 4 demonstrates the evaluations of the single-sample unlearning auditing, and the lower half of Table 4 presents the evaluations of the multi-sample unlearning auditing. The bolded values indicate the best performance among the compared methods. We fill a dash when the method does not contain the evaluation metrics.

Setup. We measure auditing methods based on the four above-introduced evaluation metrics in single-sample and multi-sample unlearning scenarios. In single-sample verification, the Erased Sample Size (*ESS*) is equal to 1 and *ESS* = 20 for the multi-sample scenario. On STL-10, we set *ESS* = 2 for the multi-sample scenario, as STL-10 only contains 5000 training samples, which is much smaller than other datasets. The evaluation here is tested based on the retraining-based unlearning method SISA [5]. To better illustrate the functionality preservation and efficiency, we record

Table 4: Overall Evaluation Results on MNIST, CIFAR10, STL-10, and CelebA.

Single-Sample Unlearning Auditing	MNIST, *ESS* = 1			CIFAR10, *ESS* = 1			STL-10, *ESS* = 1			CelebA, *ESS* = 1		
	Original	MIB [17]	TAPE	Original	MIB	TAPE	Original	MIB	TAPE	Original	MIB	TAPE
Running time (s)	620	637	**143**	651	672	**135**	781	815	**79.81**	1546	1622	**32.76**
Model Utility (Acc.)	**99.14%**	98.31%	**99.14%**	**81.62%**	79.45%	**81.62%**	**68.99%**	67.54%	**68.99%**	**96.93%**	96.05%	**96.93%**
Rec. Sim.	-	-	**0.965**	-	-	**0.974**	-	-	**0.175**	-	-	**0.977**
Unl. Verifiability	0.00%	0.00%	**99.43%**	0.00%	0.00%	**98.76%**	0.00%	0.00%	**84.00%**	0.00%	0.00%	**97.64%**
Multi-Sample Unlearning Auditing	MNIST, *ESS* = 20			CIFAR10, *ESS* = 20			STL-10, *ESS* = 2			CelebA, *ESS* = 20		
	Original	MIB [17]	TAPE	Original	MIB	TAPE	Original	MIB	TAPE	Original	MIB	TAPE
Running time (s)	613	638	**113**	644	673	**113**	781	809	**74.90**	1570	1663	**21.43**
Model Acc.	**99.05%**	98.73%	**99.05%**	**81.62%**	79.13%	**81.62%**	**68.99%**	67.26%	**68.99%**	**97.01%**	96.88%	**97.01%**
Rec. Sim.	-	-	**0.933**	-	-	**0.973**	-	-	**0.174**	-	-	**0.970**
Unl. Verifiability	0.00%	0.00%	**98.67%**	0.00%	0.00%	**97.44%**	0.00%	0.00%	**84.40%**	0.00%	0.00%	**94.57%**

the performance of solely training the original model, shown as "Original" in Table 4.

Evaluation of Efficiency. Since TAPE does not involve the original model training process, it consumes much less running time than MIB and "Original". The "Original" is training the original model before unlearning, and the MIB method needs to backdoor the model during the initial model training process before unlearning. Specifically, TAPE achieves more than 4.5× speedup in efficiency on MNIST, 5× speedup on CIFAR10, 10× speedup on STL-10, and 50× speedup on CelebA. On CelebA, the best speedup is up to 75×.

Evaluation of Functionality Preservation. The effect of functionality preservation is measured by model accuracy. In both single-sample and multi-sample unlearning auditing, our TAPE always achieves better functionality preservation than MIB. The highest accuracy preservation is around 2%, achieved on CIFAR10. The reason is that the MIB method needs to mix backdoored samples into the training dataset, and the backdoored samples with modified labels will negatively influence model utility. On the contrary, the TAPE scheme is independent of the original model training process; hence, our method will not influence the model utility of the original ML service model, keeping the same model accuracy as "Original", demonstrating better functionality preservation.

Evaluation of Unlearning Auditing Effect. We use reconstruction similarity to measure how much information about the specified samples is unlearned. The MIB method is unable to provide such an assessment of unlearned information for evaluation of unlearning effectiveness. Hence, we fill a dash of MIB in this metric. Reconstruction for a single sample always achieves better results than for multiple samples, which confirms our previous analysis and existing works [2, 34]. The unlearned posterior difference of a single sample contains more information about such a sample than a posterior difference of multiple samples, as information from multiple samples is interwoven together in one posterior difference.

To align with existing unlearning verification methods, we propose the verifiability metric to evaluate the data removal status, which is defined in Eq. (9). Since the erased sample size is small and only genuine unlearned samples are evaluated in this experiment, the MIB cannot successfully verify the unlearning of any genuine samples in Table 4. In both single-sample and multi-sample unlearning scenarios, TAPE provides effective data removal verification (accuracy larger than 95% on MNIST, CIFAR10, and CelebA). Moreover, data removal status verification for a single sample always achieves better results than for multiple samples.

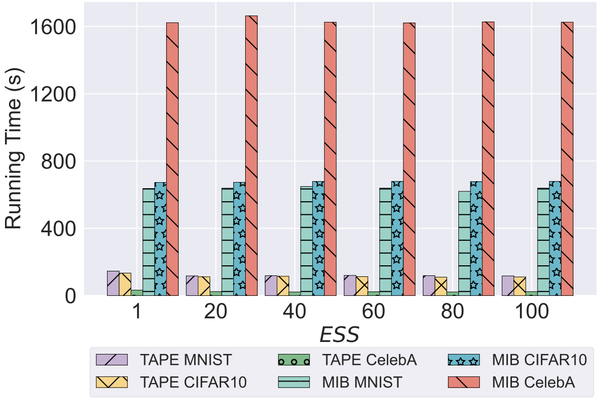


Figure 8: Running time about different *ESS*.

G.2 Impact on Efficiency of Erased Samples Size (*ESS*)

Impact on Efficiency. The main components of the running time of TAPE are building unlearned shadow models and training the reconstructor. The running time of these two processes is highly related to the size of the user's local dataset. In our experiments, we randomly select 0.5% samples on MNIST and CIFAR10 and choose 0.06% samples on CelebA as the local dataset.

Figure 8 shows the running time of TAPE and MIB on the three datasets. The running time of TAPE has no significant relationship with the *ESS* because the running time of TAPE (shadow model building and reconstructor training) highly depends on the size of the user's local dataset. For MIB, the running time has no obvious variations when *ESS* increases. This is because the MIB verification preparation is accompanied by the original model training, which is heavily related to the size of training datasets. TAPE has a much more efficient running time compared with MIB, as TAPE is independent of the original model training.