electronics

Article

Differential Privacy-Enabled Multi-Party Learning with Dynamic Privacy Budget Allocating Strategy

Ke Pan [1,*] and Kaiyuan Feng [2]

[1] School of Cyber Engineering, Xidian University, Xi'an 710071, China
[2] School of Electronic Engineering, Xidian University, Xi'an 710071, China
* Correspondence: kpan@xidian.edu.cn

Abstract: As one of the promising paradigms of decentralized machine learning, multi-party learning has attracted increasing attention, owing to its capability of preventing the privacy of participants from being directly exposed to adversaries. Multi-party learning enables participants to train their model locally without uploading private data to a server. However, recent studies have shown that adversaries may launch a series of attacks on learning models and extract private information about participants by analyzing the shared parameters. Moreover, existing privacy-preserving multi-party learning approaches consume higher total privacy budgets, which poses a considerable challenge to the compromise between privacy guarantees and model utility. To address this issue, this paper explores an adaptive differentially private multi-party learning framework, which incorporates zero-concentrated differential privacy technique into multi-party learning to get rid of privacy threats, and offers sharper quantitative results. We further design a dynamic privacy budget allocating strategy to alleviate the high accumulation of total privacy budgets and provide better privacy guarantees, without compromising the model's utility. We inject more noise into model parameters in the early stages of model training and gradually reduce the volume of noise as the direction of gradient descent becomes more accurate. Theoretical analysis and extensive experiments on benchmark datasets validated that our approach could effectively improve the model's performance with less privacy loss.

Keywords: multi-party learning; privacy; differential privacy; privacy budget; noise perturbation

check for updates

Citation: Pan, K.; Feng, K. Differential Privacy-Enabled Multi-Party Learning with Dynamic Privacy Budget Allocating Strategy. *Electronics* **2023**, *12*, 658. https://doi.org/10.3390/electronics12030658

Academic Editor: Dimitra I. Kaklamani

Received: 27 December 2022
Revised: 25 January 2023
Accepted: 26 January 2023
Published: 28 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the popularization of artificial intelligence (AI), deep learning has brought about notable achievements in autonomous driving [1,2], image recognition [3–5], medical diagnosis [6–8], and much more. Data is the oil for the development of AI, and the training of AI-based products always requires a large number of representative datasets. In single-party learning, personal data is collected in a centralized location. Data owners can neither delete it nor control its purpose of use [9], and untrusted data curators, or malicious external adversaries, may launch various attacks to eavesdrop on this sensitive information, which poses a considerable threat to individual privacy. Multi-party learning [10,11] can alleviate the aforementioned privacy threats by taking advantage of model structure. It enables all participants to collaboratively build a joint model without exposing their private data. Each participant trains their models locally and shares a fraction of model parameters instead of original data to the server for model aggregation. Therefore, multi-party learning can, to a certain extent, prevent the local data of participants from being exposed to the server.

Although multi-party learning avoids direct contact between the cloud server and participants, it is still vulnerable to inference attacks and reconstruction attacks [12–18], especially if the model parameters are not fully safeguarded. As an example, Phong et al. [14] demonstrated that uploaded gradients may be leveraged to extract local private data, since

the ratio of the gradient of weights to that of the bias is approximate to the training input. Hitaj et al. [15] designed a reconstruction attack model that enabled adversaries to construct a generative adversarial network [19] that produced similar-looking samples of target data by using the shared model as a discriminator. Melis et al. [17] showed that the adversary in multi-party learning scenarios can build active and passive membership inferences to deduce the inclusion or exclusion of the target sample.

To mitigate privacy threats in multi-party learning, several outstanding solutions have been developed. Shokri and Shmatikov [9] designed the pioneering differentially private distributed deep learning model, where participants learn their models locally, based on private data, and upload a small portion of sanitized gradients to the server to provide privacy guarantees for training data. Phong et al. [20] demonstrated that in work [9], local data was still at risk of leaking to an honest-but-curious server. Thus, they proposed an enhanced approach based on homomorphic encryption to prevent the gradients from being attacked by an untrusted server. Geyer et al. [21] devised client-side differentially private federated learning, which incorporates differential privacy into model aggregation to hide the contribution of the participant. Unlike work [21], Zhao et al. [22] perturbed the objective function, rather than gradients or model parameters, to avoid potential privacy leakage. They first converted the objective function into polynomial form and, then, injected perturbation into the coefficients of the polynomial to achieve privacy guarantees. Although these methods can achieve privacy preservation, to some extent, they still have limitations in model performance, communication overhead, and privacy loss.

Typical privacy-preserving techniques contain secure multi-party computation (SMC) [23,24], homomorphic encryption (HE) [20,25], differential privacy (DP) [21,26,27], etc. SMC is a lossless technique that provides strong privacy guarantees at the cost of significant communication overhead, due to the multiple rounds of interactions. Although HE directly allows algebraic operations on ciphertext without decryption, it is vulnerable to privacy threats if participants are assigned the same secret key and collude with each other. Compared to SMC and HE, differential privacy can not only reduce the communication burden and computation overhead during the entire training process, but also achieve provable and stronger privacy preservation. However, it is challenging for differential privacy to pursue a compromise between privacy preservation and model performance. Therefore, there is an urgent demand to seek a way to cut down the expenditure of privacy budgets effectively, while maintaining a model's utility.

This study developed an adaptive differentially private multi-party learning (ADPML) framework, which can mitigate privacy leakage effectively without sacrificing model utility. To defend the honest-but-curious server and participant, simultaneously, we incorporate the zero-concentrated differential privacy technique into the shared model before uploading them. Moreover, for the purpose of achieving the compromise between privacy preservation and model utility, we designed a dynamic privacy budget allocating strategy. Before model aggregation, we injected more noise into model parameters in the early stages of model training on the participant-side, and gradually reduced the amount of noise as the direction of gradient descent became more accurate.

The contributions of this paper are three-fold:

- We propose an adaptive differentially private multi-party learning framework based on the zero-concentrated differential privacy technique, which yields stronger privacy guarantees and permits tighter bounds for privacy computations.
- We design a dynamic privacy budget allocating strategy to avoid superfluous injection of noise and achieve the maximization of model accuracy under a high privacy-preserving level. This strategy can effectively reduce total privacy budgets and outperform fixed noise allocation.
- We experimentally validate the utility of ADPML on two benchmark datasets. Qualitative and quantitative experiments demonstrated that ADPML had better model performance, while decreasing privacy loss.

The remaining part of the paper proceeds in the following way. The preliminaries of this work are introduced in Section 2. Section 3 deals with an overview of our approach. The experimental evaluations are discussed in Section 4. At the end, we summarize the paper in Section 5.

2. Preliminaries

2.1. Multi-Party Learning

As a canonical distributed learning system, multi-party learning allows participants to learn their model locally without sharing training data with a server. A general multi-party learning framework is shown in Figure 1, which comprises a cloud server and N participants. Each participant P_i has its own local dataset D_i, where $i \in \{1, 2, \ldots, N\}$. Formally, the task of multi-party learning is formulated as:

$$\omega^* = \arg\min_{\omega} \sum_{i=1}^{N} p_i F_i(\omega), \tag{1}$$

where N is the total number of participants, $p_i = \frac{|D_i|}{|D|}$ denotes the relative impact of each participant, and $F_i(\omega)$ expresses the loss function of the i-th participant. The global model parameter ω can be calculated via model aggregation at the server, i.e., $\omega = \sum_{i=1}^{N} p_i \omega_i$. Three basic training steps of multi-party learning are shown as follows:

- The selected participants first conduct local training, based on their private data, and then upload the trained parameters to the cloud server.
- The cloud server aggregates the parameters uploaded by all active participants, and then conveys the aggregated parameters to the chosen participants.
- Participants update their models via the aggregated parameters. The iterative training process continues until the convergence criterion of the updated model is satisfied.

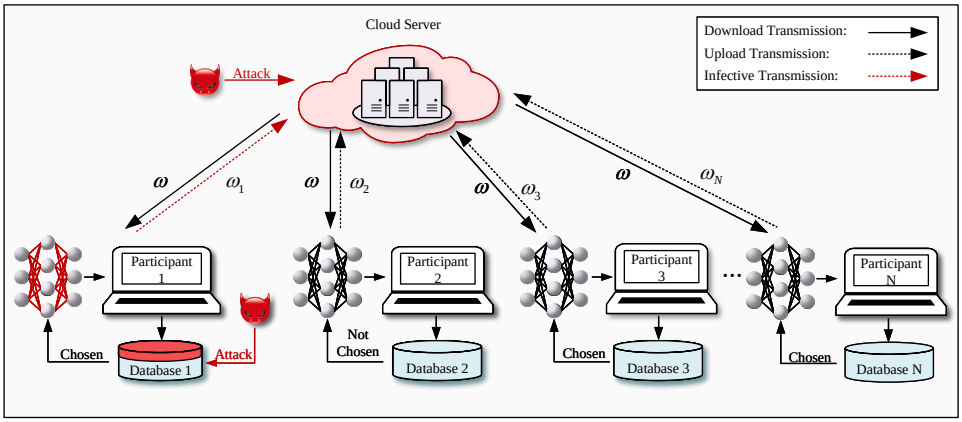


Figure 1. A system architecture for the multi-party learning training model with an honest-but-curious server or participant.

2.2. Differential Privacy

As a promising privacy notion, differential privacy [28–30] is extensively applied in data release [31] and data analysis [32]. It overcomes the weakness of traditional privacy-preserving techniques [33–36] and offers rigorous and provable privacy guarantees for statistics analysis. Differential privacy can make sure that the private information of each individual is incapable of being inferred, even if the adversary possesses sufficient auxiliary information. We define differential privacy as follows:

Definition 1. *((ϵ, δ)-DP). Given two adjacent datasets $D, D' \in \mathcal{D}$ differing on, at most, one record, a randomized algorithm $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ satisfies (ϵ, δ)-DP if, for any set of outputs, $O \subseteq \mathcal{R}$, we have:*

$$\Pr[\mathcal{A}(D) \in O] \leq \exp(\epsilon) \Pr\left[\mathcal{A}\left(D'\right) \in O\right] + \delta, \tag{2}$$

where privacy budget ϵ represents the privacy guarantee level that algorithm $\mathcal{A}$ provides. A smaller ϵ brings a stronger privacy-preserving level, and vice-versa. In addition, the parameter δ represents a relaxation factor, which relaxes the requirement of differential privacy. If $\delta = 0$, the randomized algorithm $\mathcal{A}$ achieves pure differential privacy (ϵ-DP). And when $\delta > 0$, $\mathcal{A}$ realizes an approximate differential privacy ((ϵ, δ)-DP). In contrast to approximate differential privacy, although pure differential privacy provides a more solid privacy guarantee, it may be too strict to permit any meaningful consequence to be published.

Gaussian mechanism [30] is a fundamental operator to estimate privacy level for numerical results. To guarantee (ϵ, δ)-DP, random noise generated from the Gaussian distribution is injected into the output of any query function $f(D)$ given the dataset D. The volume of noise is decided by l_2 sensitivity, which denotes the maximum change on the output of query function $f(D)$ when the record of any individual is changed.

Definition 2. *(l_2 Sensitivity). Given two adjacent datasets $D, D' \in \mathcal{D}$ differing on, at most, one record, a query function is denoted by $f : \mathcal{D} \to \mathcal{R}^d$. The l_2 sensitivity of query function f is defined as:*

$$\Delta_2 f = \max_{D, D'} \left\| f(D) - f\left(D'\right) \right\|_2. \tag{3}$$

Definition 3. *(Gaussian Mechanism). Given a dataset $D \in \mathcal{D}$, let $f : \mathcal{D} \to \mathcal{R}^d$ be a query function with l_2 sensitivity of $\Delta_2 f$. For $\epsilon \in (0, 1)$, the random algorithm $\mathcal{A}(D) = f(D) + N\left(0, \sigma^2\right)$, with*

$$\sigma \geq \frac{\Delta_2 f}{\epsilon} \sqrt{2 \ln(1.25/\delta)} \tag{4}$$

satisfies (ϵ, δ)-DP, where $N\left(0, \sigma^2\right)$ represents the Gaussian distribution with mean 0 and covariance σ^2.

2.3. Zero-Concentrated Differential Privacy

Zero-concentrated differential privacy (zCDP) [37] is a new relaxation version of differential privacy, which can not only provide a stricter privacy definition, but also allow a tighter, yet simpler, privacy analysis for fundamental tasks. Compared to ϵ-DP and (ϵ, δ)-DP, zCDP offers a stronger group privacy guarantee and permits tighter bounds for privacy computations. Consider a random variable called privacy loss that depends on the random perturbation injected into the algorithm, then, for any set of outputs $O \subseteq Range(A)$, the privacy loss random variable Z is given by:

$$Z = \log \frac{\Pr[\mathcal{A}(D) = O]}{\Pr[\mathcal{A}(D') = O]}. \tag{5}$$

ρ-zCDP entails a bound on the moment generating function of the privacy loss. The definition of ρ-zCDP is described below.

Definition 4. *(ρ-zCDP)). Given two adjacent datasets $D, D' \in \mathcal{D}$ differing on, at most, one record, a randomized algorithm $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ guarantees ρ-zCDP if, for all $\alpha \in (1, \infty)$, we have*

$$e^{D_\alpha(\mathcal{A}(D) \| \mathcal{A}(D'))} = \mathbb{E}\left[e^{(\alpha-1)Z}\right] \leq e^{(\alpha-1)\alpha\rho}, \tag{6}$$

where $D_\alpha(\mathcal{A}(D) \| \mathcal{A}(D'))$ is the α-Rényi divergence between the distribution of $\mathcal{A}(D)$ and $\mathcal{A}(D')$. In addition, we also used the following proposition about zCDP throughout this paper.

Proposition 1. *If a randomized algorithm $\mathcal{A}$ provides ρ-zCDP, then $\mathcal{A}$ is $\left(\rho + 2\sqrt{\rho \log(1/\delta)}, \delta\right)$-differentially private for any $\delta > 0$.*

Proposition 2. *Given a query function $f : \mathcal{D} \to \mathcal{R}^d$, the Gaussian mechanism which returns $f(D) + N\left(0, \sigma^2\right)$ satisfies $\left(\Delta_2 f\right)^2 \Big/ \left(2\sigma^2\right)$-zCDP.*

3. Methodology

3.1. Overview

In daily routine, it is a universal situation that an unreliable server and participants appear in distributed training environments. Consider a realistic scenario where several small-sized companies train a joint model cooperatively through a rented cloud server, rather than training their model locally, to obtain more accurate results for a prediction task. The rented server may be curious about private information belonging to participants, or the small-sized companies involved in the training process may also intend to have a look at private data that belongs to other participants. Thus, the existence of the unreliable server and participants results in a non-ignorable problem of privacy leakage during the joint training process.

In ADPML, we assumed the server was honest-but-curious, that is, the server abides by the protocol with all participants; however, it may attempt to deduce the training data, or private features, of participants, based on the convenience that the server can fully access local shared parameters. In addition, we supposed that participants were honest-but-curious. In other words, the participants were interested in the privacy of others, and they might conspire with each other to recover the privacy of the victim. According to the above assumptions, our goal was to protect the privacy of participants from being stolen by adversaries throughout the training process.

The system architecture of our adaptive differentially private multi-party learning framework is illustrated in Figure 2. We assumed there are N participants and a cloud server. Each participant simultaneously conducts local training on the basis of their private training data, and the learned local models have the same architecture and identical learning goals. The cloud server may be administered by an honest-but-curious curator, and it performs model aggregation over parameters uploaded by participants. This training process aims to learn a model that incurs lower communication overhead and alleviates the privacy threats of local private training data.

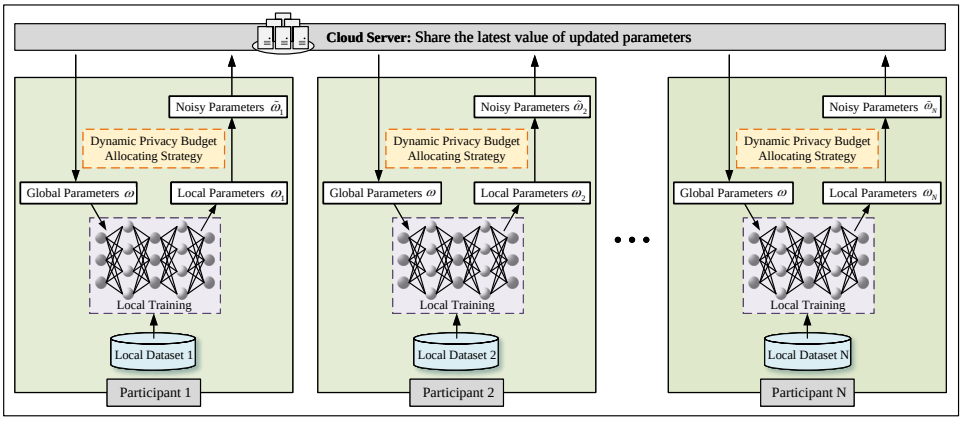


Figure 2. The system architecture of adaptive differentially private multi-party learning framework.

In our model, the server first shares the initial global parameter with participants. Then, in the t-th aggregation, the K participants learn their local model in an adaptive differentially private manner. To be specific, active participants learn parameters based on local training data. After local training is finished, each active participant injects perturbation into the trained parameters and shares the perturbed parameters with the

cloud server. Then, the server aggregates the perturbed local parameters, updates the global parameters, and distributes the aggregated parameters to each active participant for the local model update. The training procedure terminates when the aggregation time reaches the preset termination conditions. The main notations used throughout the paper are exhibited in Table 1.

Table 1. Notations.

Notations	Explanation
$\mathcal{A}$	Randomized algorithm
D, D'	Adjacent datasets
ϵ	Privacy budget
δ	Relaxation factor
$\Delta_2 f$	l_2 Sensitivity
ρ	The privacy loss related to zCDP
$\rho_{current}$	The privacy loss in the current epoch
N	The number of all participants
K	The number of chosen participants $(1 \leq K \leq N)$
T	The number of communication rounds
t	The index of the current t-th communication round
ω	The vector of model parameters after aggregating
$F_i(\omega)$	Local loss function of the i-th participant
ω_i	The vector of model parameters for the i-th participant
$\tilde{\omega}_i$	Noisy model parameters for the i-th participant
C	Clipping threshold for bounding ω
β	Privacy loss increase rate

3.2. Dynamic Privacy Budget Allocating

Traditional differentially private deep learning approaches prefer to inject a fixed volume of noise into parameters to provide privacy guarantees for sensitive data, which may not be suitable for real scenarios and poses a challenge to the compromise between model utility and privacy preservation. In practice, existing equivalent noise addition methods ignore the characteristics of deep learning models. In the early phase of model optimization, parameters are initialized randomly, and gradient values are large, so there is a lot of potential space for model optimization. At this time, the model can still achieve better parameter updates even if gradients are not measured exactly. Consequently, it is appropriate to perform rough parameter updates in the early training stages. However, as the model parameters gradually approach the optimum, the direction of gradient descent becomes more precise, and there is an urgent demand to accurately measure the volume of added noise. Thus, dynamic privacy budget allocating is more appropriate than fixed noise addition [38].

Therefore, we injected more noise into parameters at the beginning of model training, and gradually decreased the volume of noise as model parameters gradually became closer to the optimum. This was because, at the beginning of model optimization, the direction of most updated parameters remained unchanged, even if relatively large noise was injected. Even though the direction of some parameter updates changed, it can be amended during the next iteration. Then, as model parameters gradually approached the optimum, the volume of added noise should be decreased for further model optimization, since the optimizer needs to refine the optimal area in the later stage of model training, and a small amount of noise has an impact on the direction of gradient decent.

In ADPML, we first introduced the maximum privacy loss $\rho_{\max}$ and the minimum privacy loss $\rho_{\min}$. To reduce the accumulation of total privacy budgets, we gradually increased the minimum privacy loss $\rho_{\min}$ to the maximum privacy loss $\rho_{\max}$, according to

the current epoch and a predefined privacy loss increase rate β during the early stages of model optimization. The growth pattern of privacy loss was defined as:

$$\rho_{current} = (1 + \beta t)\rho_{\min}, \tag{7}$$

where t is the current epoch, and β controls the growth rate of privacy loss. The dynamic privacy loss would be maintained at a certain level in the later period, i.e., the predefined maximum privacy loss $\rho_{\max}$, and would not increase anymore. This was because a slight alteration in model parameters would lead to a huge change in the model output during the later training stages, and it would be difficult for the model to converge to a local optimum if we introduced excessive noise to the parameters.

According to Proposition 2, we calculated the noisy parameters in light of the Gaussian mechanism, with variance $\frac{(\Delta_2 f)^2}{2\rho}$ as:

$$\tilde{\omega}_i^{(t)} = \omega_i^{(t)} + \mathcal{N}\left(0, \frac{2C^2}{|D_i|^2\rho}\right), \tag{8}$$

where C is a parameter clipping constant for bounding ω_i. Taking advantage of this dynamic privacy budget allocating, we measured the current parameters by:

$$\tilde{\omega}_i^{(t)} = \begin{cases} \omega_i^{(t)} + \mathcal{N}\left(0, \frac{2C^2}{|D_i|^2\rho_{current}}\right), & \rho_{current} < \rho_{\max} \\ \omega_i^{(t)} + \mathcal{N}\left(0, \frac{2C^2}{|D_i|^2\rho_{max}}\right), & \rho_{current} \geq \rho_{\max} \end{cases} \tag{9}$$

to reduce the superfluous injection of noise and decrease the excessive accumulation of total privacy budgets.

Algorithm 1 outlines ADPML with a dynamic privacy budget allocating strategy. From the perspective of participants, each of them independently trains their local models on their private datasets and shares the noisy parameters of the local model with the cloud server after employing differential privacy to achieve privacy guarantees. To be specific, the algorithm has two main components for participants, i.e., parameter clipping and adaptive noise injection.

- **Parameter clipping.** We calculated the noisy parameter $\tilde{\omega}_i^{(t+1)}$ via the Gaussian mechanism with variance σ^2. σ^2 relied on the maximum effect an element can have on $\omega_i^{(t+1)}$, which was determined by $\Delta_2 f$. Therefore, for the purpose of providing a boundary to the impact on $\omega_i^{(t+1)}$, we computed local parameters $\omega_i^{(t+1)} = \arg\min_{\omega_i} F_i(\omega_i)$ and divided local parameters $\omega_i^{(t+1)}$ by $\max\left(1, \frac{\left\|\omega_i^{(t+1)}\right\|}{C}\right)$, given a predefined clipping threshold C. Thus, the sensitivity of parameters $\Delta_2 f$ was bound by C.
- **Adaptive noise injection.** In order to reduce the total privacy budgets, while maintaining the model performance, we adaptively redistributed privacy loss for adjusting noise scale, based on the dynamic privacy budget allocating strategy. We first transformed (ϵ, δ)-DP to ρ-zCDP through Proposition 1 and defined a privacy loss increase rate β. Then, we gradually increased the privacy loss to the maximum $\rho_{\max}$, based on β, with the direction of the gradient descent becoming more accurate.

From the cloud server point of view, after active participants accomplished local training and shared their sanitized model update with the cloud server to achieve model aggregation, the cloud server updated the global parameter, based on the uploaded noisy parameters by means of:

$$\omega^{(t+1)} = \frac{1}{K}\sum_{i=1}^{K} \tilde{\omega}_i^{(t+1)}. \tag{10}$$

Algorithm 1 Adaptive Differentially Private Multi-Party Learning (ADPML).

Input: Number of participants joining in each communication round K, rounds of communication T, privacy budget $\epsilon_{\min}$ and $\epsilon_{\max}$, relaxation factor δ, clipping threshold C, privacy loss increase rate β.

Output: The model parameter ω_T.

Initialize $\omega_i^{(0)} = \omega^{(0)}$, and $t = 0$.
while $t < T$ **do**
  Local training process:
  while $i \in K$ **do**
    Update local parameters $\omega_i^{(t+1)} = \arg\min_{\omega_i} F_i\left(\omega_i, \omega_i^{(t)}\right)$.
    Clip local parameters $\omega_i^{(t+1)} = \omega_i^{(t+1)} \Big/ \max\left(1, \frac{\left\|\omega_i^{(t+1)}\right\|}{C}\right)$.
    Calculate the current privacy loss $\rho_{current} \leftarrow (1 + \beta t)\rho_{\min}$. // Compared to (ϵ, δ)-DP, we achieved the transformation of privacy budgets by using Proposition 1.
    if $\rho_{current} \geq \rho_{\max}$ **then**
      Add noise and upload parameters $\tilde{\omega}_i^{(t)} = \omega_i^{(t)} + \mathcal{N}\left(0, \frac{2C^2}{|D_i|^2 \rho_{max}}\right)$.
    else
      Add adaptive noise and upload parameters $\tilde{\omega}_i^{(t)} = \omega_i^{(t)} + \mathcal{N}\left(0, \frac{2C^2}{|D_i|^2 \rho_{current}}\right)$.
    end if
  end while
  Model aggregation process:
  Update the global parameters $\omega^{(t+1)} = \sum_{i \in K} p_i \tilde{\omega}_i^{(t+1)}$.
  The cloud server broadcasts global parameters.
  Local testing process:
  while $P_i \in \{P_1, P_2, \cdots, P_N\}$ **do**
    Test the aggregating parameters $\omega^{(t+1)}$ using local dataset.
  end while
  $t = t + 1$.
end while

3.3. Sensitivity and Privacy Analysis

From the upload perspective, given neighbor datasets $D_i, D_i' \in \mathcal{D}$ for the i-th participant, the local training process of each participant can be depicted as:

$$\begin{aligned} \omega_i(D_i) &= \arg\min_{\omega} F_i(D_i, \omega) \\ &= \frac{1}{|D_i|} \sum_{j=1}^{|D_i|} \arg\min_{\omega} F_i\left(D_{i,j}, \omega\right). \end{aligned} \tag{11}$$

Consequently, the sensitivity of the local training process of the i-th participant can be obtained by:

$$\begin{aligned} \Delta\omega_i(D_i) &= \left\| \frac{1}{|D_i|} \sum_{j=1}^{|D_i|} \arg\min_{\omega} F_i\left(D_{i,j}, \omega\right) - \frac{1}{|D'_i|} \sum_{j=1}^{|D'_i|} \arg\min_{\omega} F_i\left(D'_{i,j}, \omega\right) \right\| \\ &\leq \left\| \frac{1}{|D_i|} \sum_{j=1}^{|D_i|} \arg\min_{\omega} F_i\left(D_{i,j}, \omega\right) \right\| + \left\| \frac{1}{|D'_i|} \sum_{j=1}^{|D'_i|} \arg\min_{\omega} F_i\left(D'_{i,j}, \omega\right) \right\| \\ &\leq 2 \max_{D_i, D'_i} \left\| \frac{1}{|D_i|} \sum_{j=1}^{|D_i|} \arg\min_{\omega} F_i\left(D_{i,j}, \omega\right) \right\| \\ &\leq \frac{2}{|D_i|} C, \end{aligned} \tag{12}$$

where $D_{i,j}$ is the j-th record in the dataset D_i. According to the Parallel Composition Theorem [30] of differential privacy, the global sensitivity during the upload process can be defined, based on the above result, which is expressed as:

$$\Delta\omega_i = \max\{\Delta\omega_i(D_i)\}, \quad for \quad \forall i \in N. \tag{13}$$

From the download perspective, the aggregated parameters are given by:

$$\omega = \sum_{i=1}^{N} p_i\omega_i = p_1\omega_1 + \cdots + p_i\omega_i + \cdots + p_N\omega_N, \tag{14}$$

where ω is the aggregated parameters at the cloud server. The sensitivity for D_i after model aggregation is denoted as:

$$\Delta\omega = \max_{D_i, D'_i} \|\omega(D_i) - \omega(D'_i)\|. \tag{15}$$

Then, based on Equations (11) and (15), we have:

$$\omega(D_i) = p_1\omega_1(D_1) + \cdots + p_i\omega_i(D_i) + \cdots + p_N\omega_N(D_N), \tag{16}$$

and

$$\omega(D'_i) = p_1\omega_1(D_1) + \cdots + p_i\omega_i(D'_i) + \cdots + p_N\omega_N(D_N). \tag{17}$$

Thus, we denote the sensitivity as:

$$\begin{aligned} \Delta\omega &= \max_{D_i, D'_i} \|p_i\omega_i(D_i) - p_i\omega_i(D'_i)\| \\ &= p_i \max_{D_i, D'_i} \|\omega_i(D_i) - \omega_i(D'_i)\| \\ &= \frac{2Cp_i}{|D_i|}. \end{aligned} \tag{18}$$

In summary, the global sensitivity after model aggregation is given by:

$$\Delta\omega = \max\{\Delta\omega(D_i)\}. \tag{19}$$

Focusing on attacks from the honest-but-curious server, these servers may be full of curiosity about private training data and attempt to deduce private information of participants. However, the server cannot come into contact with the private information of each participant directly since all uploaded parameters from each participant are perturbed adaptively by incorporating differential privacy into the computation procedure. Therefore, we achieve solid privacy preservation to resist privacy threats from an honest-but-curious server. Concerning attacks from the honest-but-curious participant, who is interested in the local training data of others, or is even colluding with another participant to recover private information of a victim, it is still not easy for adversaries to deduce the private information of the victim. This is because each participant conducts their local training and shares noisy parameters independently, and each participant has no influence on the other. The aggregation of parameters is on the basis of the perturbed parameters, so the curious participant has no opportunity to directly access the original parameters. Consequently, the ADPML framework can also mitigate privacy concerns regarding honest-but-curious participants, while enhancing privacy guarantees during the model training. Moreover, we derived the proof of the privacy level of Algorithm 1.

Theorem 1. *Algorithm 1 satisfied* (ϵ, δ)*-DP.*

Proof. Given the number of aggregation times T, and the number of aggregation a required to reach $\rho_{\max}$. Algorithm 1 guaranteedρ-zCDP with

$$\begin{aligned}\rho_{total} &= \rho_0 + \rho_1 + \ldots + \rho_{T-1} \\ &= [\rho_{\min} + (1+\beta)\rho_{\min} + \ldots + (1+(a-1)\beta)\rho_{\min}] + (T-a)\rho_{\max} \\ &= \frac{a\rho_{\min}[2+\beta(a-1)]}{2} + (T-a)\rho_{\max}.\end{aligned} \tag{20}$$

Furthermore, Algorithm 1 satisfied (ϵ, δ)-DP, where

$$\epsilon = \rho + 2\sqrt{\rho \log(1/\delta)}. \tag{21}$$

□

4. Experiments

In this part, we first introduce the experimental setup, including datasets, model architectures, and comparison algorithms. Then, we evaluate the performance of ADPML in terms of privacy level, number of participants, and relaxation factor, respectively. Experiments were also performed on the economization proportion of total privacy budgets for different privacy loss increase rates β to further indicate the effectiveness of the ADPML framework.

4.1. Experimental Setup

We manifested the performance of our ADPML framework based on two benchmark datasets, which are shown as follows.

- MNIST [39] is a benchmark dataset that is related to handwritten grey-level images of digits from 0 to 9. MNIST contains 70,000 grey-level images including 60,000 training examples and 10,000 test examples, and each example is a 28×28 image.
- CIFAR-10 [40] comprises 60,000 RGB color images with ten categories, such as cars, cats, dogs, ships, and so on. There are 50,000 training examples and 10,000 test examples in CIFAR-10, and each example has size 32×32 with three channels.

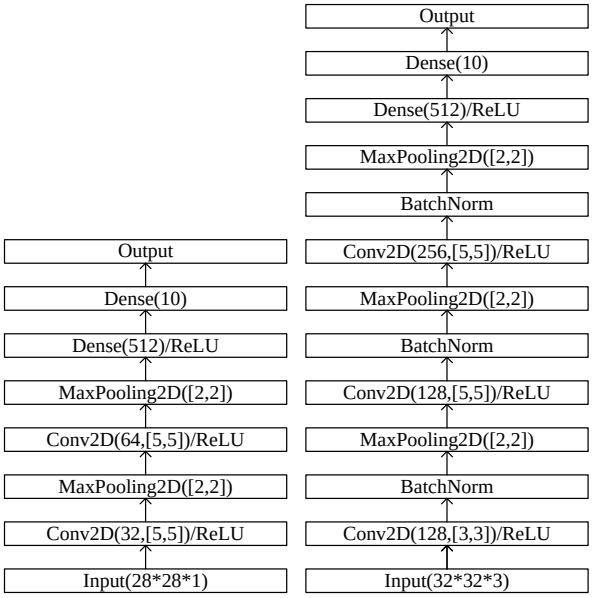
Our experiments were conducted on two different deep learning architectures for MNIST and CIFAR-10 datasets, respectively, which are shown in Figure 3. For the MNIST dataset, the network architecture contained 2 convolutional layers, the first layer with 32 channels and the second one with 64 channels. The convolutional layers used 5×5 convolutions with stride 1, each followed by 2×2 max pooling. The fully connected layer had 512 units, and we used a softmax containing 10 digits. For the CIFAR-10 dataset, we used a similar model to the MNIST dataset, which was comprised of 3 convolutional layers and 2 fully connected layers. In the following, we set the number of participants as $N = 30$, $N = 60$ and $N = 90$, split the training dataset according to the predefined number of participants, and set the learning rate on MNIST and CIFAR-10 to 0.002 and 0.01, respectively. Next, we defined the maximum privacy loss $\epsilon_{\max}$ as 10 and the minimum privacy loss $\epsilon_{\min}$ as 1. The clipping threshold C was set to 4. According to Proposition 1, we could realize the transformation of (ϵ, δ)-DP to ρ-zCDP, as shown below:

$$\epsilon \geq \rho + 2\sqrt{\rho \log(1/\delta)}. \tag{22}$$

We trained and tested all models ten times and reported the average results. Furthermore, we compared ADPML and four baselines, which are listed below.

- ADPML, a differential privacy-enabled multi-party learning framework with a dynamic privacy budget allocating strategy.
- Centralized, a centralized model training on the entire datasets from all participants without any privacy consideration.

- SecProbe [22], which protects the privacy of each participant by inserting perturbation into the objective function based on the functional mechanism.
- CSDP [21], which injects perturbation into the sum of all updates from each participant.
- Fixed Noise ($\epsilon_{\max}$), a differentially private multi-party learning framework with a fixed minimum noise level $\epsilon_{\max}$.
- Fixed Noise ($\epsilon_{\min}$), a differentially private multi-party learning framework with a fixed maximum noise level $\epsilon_{\min}$.



(a) MNIST (b) CIFAR-10

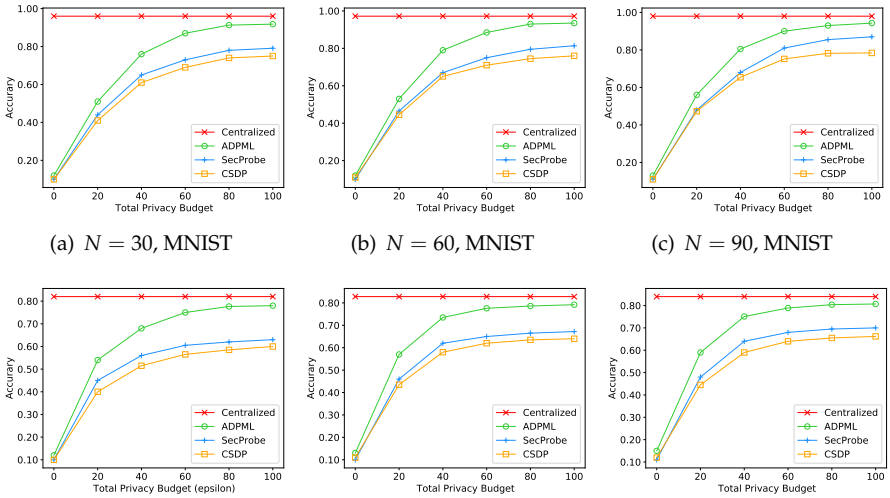
Figure 3. The network architectures of MNIST and CIFAR-10 datasets.

4.2. Experiments on the Level of Privacy Guarantees

In Figure 4, we compared ADPML and Centralized, SecProbe, and CSDP algorithms under the same total privacy budget, based on different numbers of participants ($N = 30$, $N = 60$, $N = 90$). In this experiment, we set $\delta = 10^{-2}$, β= 0.9 for MNIST and $\delta = 10^{-5}$, β= 0.6 for CIFAR-10, respectively. As the model training proceeded, more privacy budget was accumulated. However, the model utility of ADPML outperformed other baselines under the same value of accumulated total privacy budget. In fact, CSDP added the same volume of noise during each epoch, which might not only increase the total privacy budget, but also degrade the model performance compared to an adaptive noise addition. Moreover, CSDP conducted the perturbation after model aggregation, which might not be solid enough to resist the honest-but-curious participants. Although SecProbe alleviated the attack effectively, it distorted the objective function of each local model through objective perturbation. The objective perturbation depended on the minimum curvature instead of the expected curvature, which might also increase the amount of noise [41]. Therefore, fixed noise addition approaches might pose a considerable challenge to the compromise between model utility and privacy preservation.

In addition, as shown in Figure 5, Tables 2 and 3, ADPML accumulated less total privacy budget compared to the Fixed Noise ($\epsilon_{\max}$), while acquiring almost the same accuracy as the Fixed Noise ($\epsilon_{\max}$). What was more, the model accuracy of ADPML and Fixed Noise ($\epsilon_{\max}$) was better than Fixed Noise ($\epsilon_{\min}$), since a larger ϵ implied a lower variance of the noise, and, in contrast, a better model performance implied the relaxation of the privacy-preserving level. When the value of ϵ was small, a large volume of perturbation would be injected, i.e., stronger privacy guarantees were achieved, but at the expense of model utility. Therefore, a dynamic privacy budget allocating was preferable to a fixed noise addition approach, as we could add more noise to model parameters at the

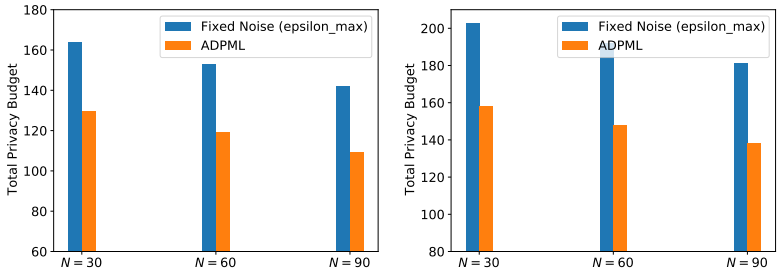
beginning of model training and gradually decrease the amount of perturbation to control the accumulation of total privacy budgets and maintain the model's accuracy. At the same time, although ADPML took slightly more communication rounds than Fixed Noise ($\epsilon_{\max}$), ADPML provided stronger privacy guarantees and saved more total privacy budget.



(a) $N = 30$, MNIST (b) $N = 60$, MNIST (c) $N = 90$, MNIST

(d) $N = 30$, CIFAR-10 (e) $N = 60$, CIFAR-10 (f) $N = 90$, CIFAR-10

Figure 4. The model accuracy of different strategies for 30, 60, and 90 participants.



(a) MNIST (b) CIFAR-10

Figure 5. The total privacy budget of different noise addition strategies for 30, 60, and 90 participants.

Table 2. The communication round of different strategies for 30, 60, and 90 participants.

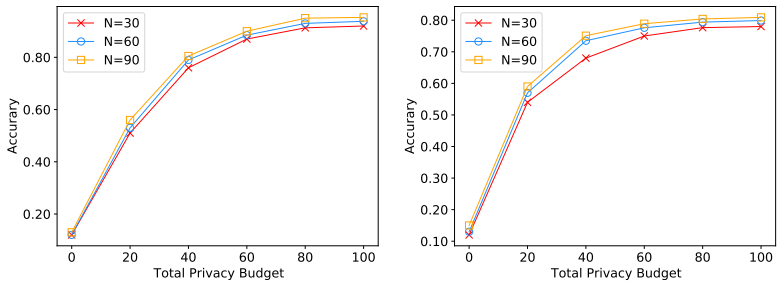
Dataset	Participants	Methods		
		Fixed Noise ($\epsilon_{\max}$)	**Fixed Noise ($\epsilon_{\min}$)**	**ADPML**
MNIST	$N = 30$	16	18	**18**
	$N = 60$	15	17	**17**
	$N = 90$	14	16	**16**
CIFAR-10	$N = 30$	20	23	**23**
	$N = 60$	19	22	**22**
	$N = 90$	18	21	**21**

Table 3. The model accuracy of different strategies for 30, 60, and 90 participants.

Dataset	Participants	Methods		
		Fixed Noise ($\epsilon_{\max}$)	**Fixed Noise ($\epsilon_{\min}$)**	**ADPML**
MNIST	$N = 30$	94.11 ± 1.08	72.15 ± 0.69	$\mathbf{94.16 \pm 0.27}$
	$N = 60$	95.62 ± 0.23	73.68 ± 0.54	$\mathbf{95.63 \pm 0.33}$
	$N = 90$	96.88 ± 0.37	74.36 ± 0.12	$\mathbf{96.92 \pm 1.12}$
CIFAR-10	$N = 30$	80.07 ± 0.42	63.23 ± 0.17	$\mathbf{80.06 \pm 0.28}$
	$N = 60$	80.98 ± 0.29	64.84 ± 0.40	$\mathbf{80.96 \pm 0.53}$
	$N = 90$	82.17 ± 2.46	65.21 ± 0.21	$\mathbf{82.14 \pm 0.16}$

4.3. Experiments on the Number of Participants

In this subsection, to illustrate the relationship between the number of participants and model accuracy, we defined privacy parameters $\delta = 10^{-2}$, β= 0.9 for MNIST and $\delta = 10^{-5}$, β= 0.6 for CIFAR-10. Participant numbers were set to $N = 30$, $N = 60$, $N = 90$, respectively. According to Figure 6, the number of participants N exerted an impact on model accuracy. Larger N indicated better model performance, since a great many participants not only implied a lower variance of noise, but also provided more training data and local gradients for model optimization.



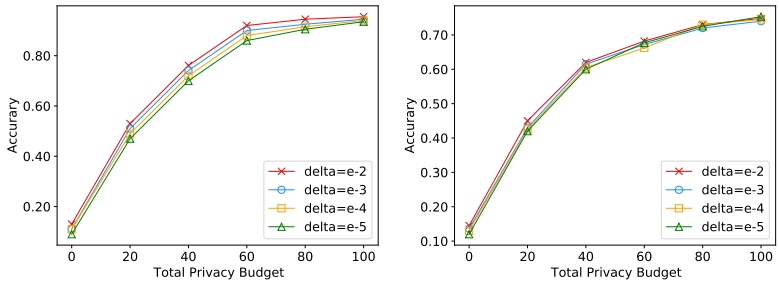
(a) MNIST (b) CIFAR-10

Figure 6. The model accuracy of ADPML under different total privacy budgets for 30, 60, and 90 participants.

In addition, according to Figure 5, Tables 2 and 3, we could derive that the accumulated total privacy budgets moderately decreased as the number of participants increased, since the more participants, the greater their contribution and the faster the model converged. Moreover, although the ADPML framework rendered the training process slightly time-consuming for realizing almost an identical accuracy as the Fixed Noise ($\epsilon_{\max}$) model, it economized around 21–24% total privacy budgets and provided stronger privacy guarantees than the Fixed Noise ($\epsilon_{\max}$).

4.4. Experiments on the Relaxation Factor

Figure 7 shows the influence of different relaxation factors δ on model accuracy under various total privacy budgets. In this subsection, the number of participants was set as $N = 60$, and we varied the relaxation factor δ in $\left\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\right\}$. As can be observed from Figure 7, each curve portrays the optimal performance acquired by a fixed relaxation factor δ. The change rate of model accuracy rose faster in the early stages, then gradually tended to be gentle, and eventually converged to a high level.



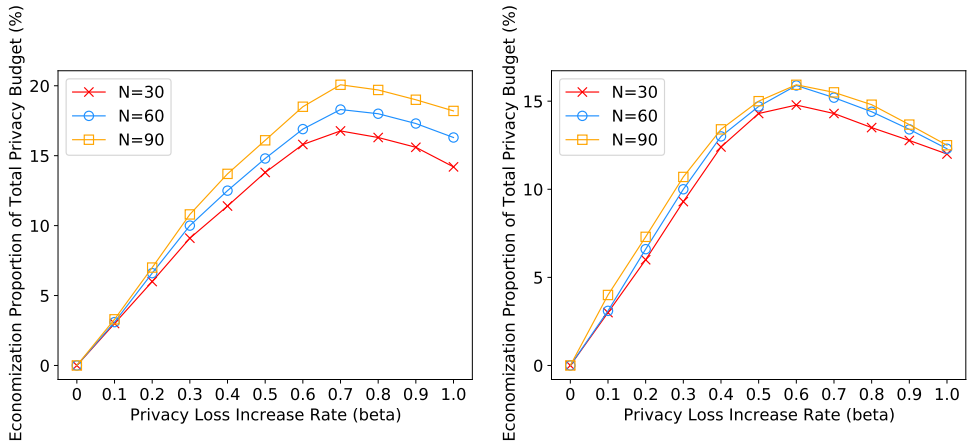
(a) MNIST (b) CIFAR-10

Figure 7. The model accuracy of ADPML under different total privacy budgets for various choices of relaxation factors.

As shown in Figure 7, we could derive that the ADPML framework achieved the best performance when the relaxation factor $\delta = 10^{-2}$ on MNIST, and the ADPML framework with $\delta = 10^{-5}$ outperformed other settings of relaxation factor δ on CIFAR-10. Moreover, from Figure 7, it cannot be denied that the change in total privacy budgets exerted a great impact on model performance for a fixed δ, whereas different values of relaxation factor δ had little impact on the model performance. Consequently, the privacy budget was of utmost importance in the prediction of model accuracy.

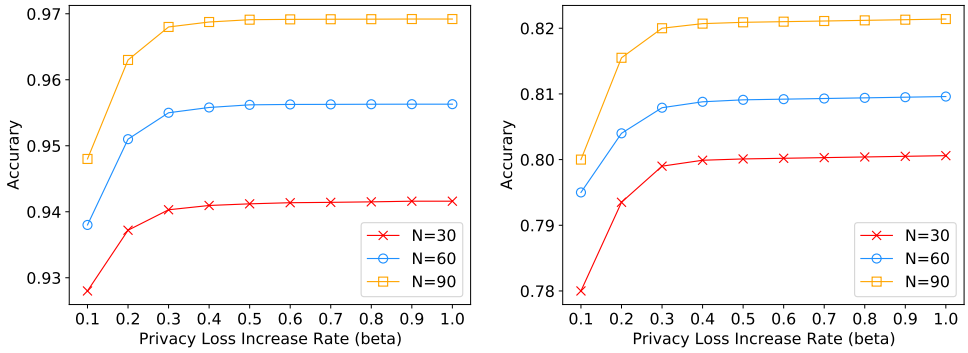
4.5. Experiments on the Privacy Loss Increase Rate

In Figures 8 and 9, we explored the relationship between privacy loss increase rate β and economization proportion of total privacy budgets and model accuracy for $N = 30$, $N = 60$, and $N = 90$ participants, respectively. In this experiment, we also used the settings with the best performance, i.e., we set $\delta = 10^{-2}$ for MNIST and set $\delta = 10^{-5}$ for CIFAR-10. The privacy loss increase rate β controlled how fast the privacy loss grew. According to Figure 8, compared to other values of the privacy loss increase rate β, ADPML economized the most total privacy budgets when $\beta = 0.7$ on MNIST. For CIFAR-10, ADPML acquired the lowest total privacy budget when $\beta = 0.6$. If β was large, the growth rate of the cumulative total privacy budget was faster, and the noise level rapidly reached the lowest initial value (i.e., $\epsilon_{\max}$); thus, the economization proportion of the total privacy budget reduced. On the flip side, when the value of β was excessively small, more epochs were required to reach the minimum noise level, and more privacy budget was spent during training.



(a) MNIST (b) CIFAR-10

Figure 8. The economization proportion of total privacy budgets with different privacy loss increase rates for 30, 60, and 90 participants respectively under ADPML.



(a) MNIST (b) CIFAR-10

Figure 9. The model accuracy under different privacy loss increase rates for 30, 60, and 90 participants under ADPML.

In Figure 9, we derived that the change of privacy loss increase rate β had an influence on model performance. When β was small, the model accuracy was affected by the setting of β, whereas when β became large, the effect of β on model performance tended to be stable. This was because the total privacy budget increased slowly when β was small, which led to a gradual increase in the amount of noise injected into model parameters, thereby affecting the accuracy of the model. However, when the privacy loss increase rate β became larger, the volume of noise added to model parameters became progressively closer to the minimum value, and the model parameters approached the optimum gradually. Furthermore, the model accuracy tended to be stable. Combined with Figure 4, we deduced that ADPML could outperform baselines at an identical value of the total privacy budget. On the flip side, ADPML could decrease the accumulation of total privacy budgets by taking advantage of the introduction of the privacy loss increase rate.

5. Conclusions

In this study, we investigated a zero-concentrated differential privacy algorithm in multi-party learning to protect the privacy of participants. We developed a dynamic privacy budget allocating strategy to avoid the excessive accumulation of total privacy budgets, while achieving a better compromise between model utility and privacy preservation, compared to traditional differential privacy preservation approaches with lower model accuracy and higher privacy cost. During the early stages of model training, we injected more noise into model parameters at the participant-side before aggregation, and gradually decreased the amount of noise with the direction of the gradient descent becoming more accurate. Rigorous experiments, validated on two benchmark datasets, illustrated the effectiveness of the ADPML framework. Although the proposed framework reduced the consumption of total privacy budgets, it was still not independent of the number of training epochs. Therefore, we plan to explore some novel approaches to improve the independence of the privacy budget, so that the consumption of total privacy budgets can become independent of the number of training steps. Moreover, it is worthwhile to extend the differential privacy mechanism to image data, and then integrate this notion into the ADPML framework to further improve the practicality of the model.

Author Contributions: Conceptualization, K.P. and K.F.; methodology, K.P.; validation, K.P.; investigation, K.F.; writing—original draft preparation, K.P.; writing—review and editing, K.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations were used in this manuscript:

AI	Artificial intelligence
GAN	Generative adversarial network
SMC	Secure multi-party computation
HE	Homomorphic encryption
DP	Differential privacy
zCDP	Zero-concentrated differential privacy
ADPML	Adaptive differentially private multi-party learning

References

1. Lee, S.; Lee, S.; Seong, H.; Hyun, J.; Kim, E. Fallen person detection for autonomous driving. *Expert Syst. Appl.* **2023**, *213*, 119242. [CrossRef]
2. Bogdoll, D.; Nitsche, M.; Zöllner, J.M. Anomaly Detection in Autonomous Driving: A Survey. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4488–4499.
3. Touvron, H.; Bojanowski, P.; Caron, M.; Cord, M.; El-Nouby, A.; Grave, E.; Izacard, G.; Joulin, A.; Synnaeve, G.; Verbeek, J.; et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, 1–9. *early access*. [CrossRef] [PubMed]
4. Tang, C.; Zhao, Y.; Wang, G.; Luo, C.; Xie, W.; Zeng, W. Sparse MLP for image recognition: Is self-attention really necessary? In Proceedings of the AAAI Conference on Artificial Intelligence, virtual, 22 February–1 March 2022; Volume 36, pp. 2344–2351.
5. Sun, L.; Zhao, G.; Zheng, Y.; Wu, Z. Spectral–Spatial Feature Tokenization Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
6. Zheng, Y.; Lu, R.; Zhang, S.; Guan, Y.; Shao, J.; Wang, F.; Zhu, H. PMRQ: Achieving Efficient and Privacy-Preserving Multi-Dimensional Range Query in eHealthcare. *IEEE Internet Things J.* **2022**, *9*, 17468–17479. [CrossRef]
7. Chen, Z.; Tian, Z.; Zhu, J.; Li, C.; Du, S. C-CAM: Causal CAM for Weakly Supervised Semantic Segmentation on Medical Image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11676–11685.
8. Fang, C.; Wang, L.; Zhang, D.; Xu, J.; Yuan, Y.; Han, J. Incremental Cross-view Mutual Distillation for Self-supervised Medical CT Synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 20677–20686.
9. Shokri, R.; Shmatikov, V. Privacy-Preserving Deep Learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 1310–1321.
10. Konecný, J.; McMahan, H.B.; Ramage, D.; Richtárik, P. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *arXiv* **2016**, arXiv:1610.02527.
11. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 9–11 May 2017; pp. 1273–1282.
12. Fredrikson, M.; Lantz, E.; Jha, S.; Lin, S.M.; Page, D.; Ristenpart, T. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, 20–22 August 2014; pp. 17–32.
13. Fredrikson, M.; Jha, S.; Ristenpart, T. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 1322–1333.
14. Phong, L.T.; Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-Preserving Deep Learning: Revisited and Enhanced. In Proceedings of the Applications and Techniques in Information Security—8th International Conference, Auckland, New Zealand, 6–7 July 2017; pp. 100–110.
15. Hitaj, B.; Ateniese, G.; Pérez-Cruz, F. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October– 3 November 2017; pp. 603–618.
16. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. In Proceedings of the 2017 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 22–24 May 2017; pp. 3–18.
17. Melis, L.; Song, C.; Cristofaro, E.D.; Shmatikov, V. Exploiting Unintended Feature Leakage in Collaborative Learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–22 May 2019; pp. 691–706.
18. Wang, Z.; Song, M.; Zhang, Z.; Song, Y.; Wang, Q.; Qi, H. Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning. In Proceedings of the 2019 IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 2512–2520.

19. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.C.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, ON, Canada, 8–13 December 2014; pp. 2672–2680.
20. Phong, L.T.; Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1333–1345. [CrossRef]
21. Geyer, R.C.; Klein, T.; Nabi, M. Differentially Private Federated Learning: A Client Level Perspective. *arXiv* **2017**, arXiv:1712.07557.
22. Zhao, L.; Wang, Q.; Zou, Q.; Zhang, Y.; Chen, Y. Privacy-Preserving Collaborative Deep Learning With Unreliable Participants. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1486–1500. [CrossRef]
23. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 1175–1191.
24. Ghazi, B.; Pagh, R.; Velingker, A. Scalable and Differentially Private Distributed Aggregation in the Shuffled Model. *arXiv* **2019**, arXiv:1906.08320.
25. Zhang, X.; Ji, S.; Wang, H.; Wang, T. Private, Yet Practical, Multiparty Deep Learning. In Proceedings of the 37th IEEE International Conference on Distributed Computing Systems, Atlanta, GA, USA, 5–8 June 2017; pp. 1442–1452.
26. Agarwal, N.; Suresh, A.T.; Yu, F.X.; Kumar, S.; McMahan, B. cpSGD: Communication-efficient and differentially-private distributed SGD. In Proceedings of the Advances in Neural Information Processing Systems 2018, Montreal, ON, Canada, 3–8 December 2018; pp. 7575–7586.
27. Bhowmick, A.; Duchi, J.C.; Freudiger, J.; Kapoor, G.; Rogers, R. Protection Against Reconstruction and Its Applications in Private Federated Learning. *arXiv* **2018**, arXiv:1812.00984.
28. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A.D. Calibrating Noise to Sensitivity in Private Data Analysis. In Proceedings of the Third Theory of Cryptography Conference, New York, NY, USA, 4–7 March 2006; pp. 265–284.
29. Dwork, C. Differential Privacy: A Survey of Results. In Proceedings of the Conference on Theory and Applications of Models of Computation, Xi'an, China, 25–29 April 2008; pp. 1–19.
30. Dwork, C.; Roth, A. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]
31. Wang, Q.; Zhang, Y.; Lu, X.; Wang, Z.; Qin, Z.; Ren, K. Real-Time and Spatio-Temporal Crowd-Sourced Social Network Data Publishing with Differential Privacy. *IEEE Trans. Dependable Secur. Comput.* **2018**, *15*, 591–606. [CrossRef]
32. Proserpio, D.; Goldberg, S.; McSherry, F. Calibrating Data to Sensitivity in Private Data Analysis. *Proc. VLDB Endow.* **2014**, *7*, 637–648. [CrossRef]
33. Machanavajjhala, A.; Kifer, D.; Gehrke, J. *L*-diversity: Privacy beyond *k*-anonymity. *ACM Trans. Knowl. Discov. Data* **2007**, *1*, 3–8. [CrossRef]
34. Li, N.; Li, T.; Venkatasubramanian, S. *t*-Closeness: Privacy Beyond *k*-Anonymity and *l*-Diversity. In Proceedings of the IEEE International Conference on Data Engineering, Istanbul, Turkey, 17–20 April 2007; pp. 106–115.
35. Wong, R.C.; Li, J.; Fu, A.W.; Wang, K. (α, *k*)-anonymity: An enhanced *k*-anonymity model for privacy-preserving data publishing. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Philadelphia, PA, USA, 20–23 August 2006; pp. 754–759.
36. Xiao, X.; Tao, Y. *M*-invariance: Towards privacy preserving re-publication of dynamic datasets. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, China, 11–14 June 2007; pp. 689–700.
37. Bun, M.; Steinke, T. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In Proceedings of the Theory of Cryptography—14th International Conference, Beijing, China, 31 October–3 November 2016; pp. 635–658.
38. Lee, J.; Kifer, D. Concentrated Differentially Private Gradient Descent with Adaptive per-Iteration Privacy Budget. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; ACM: New York, NY, USA, 2018; pp. 1656–1665.
39. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
40. Krizhevsky, A.; Hinton, G.E. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; University of Toronto: Toronto, ON, Canada, 2009.
41. Yu, D.; Zhang, H.; Chen, W.; Yin, J.; Liu, T. Gradient Perturbation is Underrated for Differentially Private Convex Optimization. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 3117–3123.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.