Qeios

DEeR: Deviation Eliminating and Noise Regulating for Privacy-preserving Federated Low-rank Adaptation

Meilu Zhu, Axiu Mao, Jun Liu*, Yixuan Yuan*

***Abstract*—Integrating low-rank adaptation (LoRA) with federated learning (FL) has received widespread attention recently, aiming to adapt pretrained foundation models (FMs) to downstream medical tasks via privacy-preserving decentralized training. However, owing to the direct combination of LoRA and FL, current methods generally undergo two problems, *i.e.*, aggregation deviation, and differential privacy (DP) noise amplification effect. To address these problems, we propose a novel privacy-preserving federated finetuning framework called Deviation Eliminating and Noise Regulating (DEeR). Specifically, we firstly theoretically prove that the necessary condition to eliminate aggregation deviation is guaranteing the equivalence between LoRA parameters of clients. Based on the theoretical insight, a deviation eliminator is designed to utilize alternating minimization algorithm to iteratively optimize the zero-initialized and non-zero-initialized parameter matrices of LoRA, ensuring that aggregation deviation always be zeros during training. Furthermore, we also conduct an in-depth analysis of the noise amplification effect and find that this problem is mainly caused by the "linear relationship" between DP noise and LoRA parameters. To suppress the noise amplification effect, we propose a noise regulator that exploits two regulator factors to decouple relationship between DP and LoRA, thereby achieving robust privacy protection and excellent finetuning performance. Additionally, we perform comprehensive ablated experiments to verify the effectiveness of the deviation eliminator and noise regulator. DEeR shows better performance on public medical datasets in comparison with state-of-the-art approaches. The code is available at https://github.com/CUHK-AIM-Group/DEeR.**

***Index Terms*—Low-rank Adaptation, Federated Learning, Parameter-efficient Tuning, Foundation Models.**

I. Introduction

With the advent of the big data era and advances in computation [1], large foundation models (FMs), such as CLIP [2], BiomedCLIP [3], SAM [4], have been developed, demonstrating unprecedented generalization performance across various medical tasks [5], [6]. However, these foundation models usually focus on general representation learning and still require further finetuning for downstream tasks [7], [8]. To avoid the computational burdens caused by finetuning entire foundation models, various parameter-efficient finetuning (PEFT) methods [9]–[11] have been proposed. One of the most widely used PEFT methods is low-rank adaptation (LoRA) [9], which adds a parallel branch of trainable adapters with parameters $\mathbf{A}$ and $\mathbf{B}$ to compute the model update $\nabla\mathbf{W}$. The ranks of $\mathbf{A}$ and $\mathbf{B}$ are much smaller than the pretrained model parameters $\mathbf{W}$. When applying LoRA for finetuning, only $\mathbf{A}$ and $\mathbf{B}$ are updated while the entire $\mathbf{W}$ is frozen, thereby significantly reducing GPU memory consumption [12].

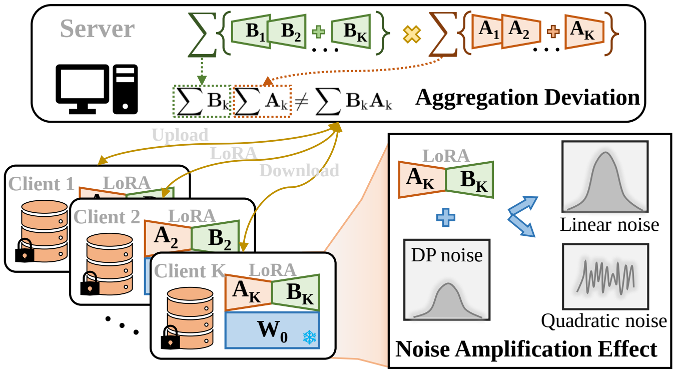


Fig. 1. Directly combining FL and LoRA to finetune FMs brings two challenges, *i.e.*, aggregation deviation and noise amplification effect.

Finetuning pretrained FMs with LoRA still requires sufficient training data for adaptation to specific downstream tasks [13], [14]. Nevertheless, data within a single institution tend to be limited [15], [16], particularly in medical scenarios. Directly gathering data from different institutions is typically unrealistic due to growing privacy concerns and legal restrictions [17]–[19]. An alternative approach is to adopt federated learning (FL) [20], a decentralized learning paradigm, as training framework to collaboratively finetune FMs with LoRA. The FL paradigm [20], [21] allows participating institutions (referred to as clients) to share their model gradients or parameters for the model aggregation at a trustworthy center (known as server), without leaking local raw data. Meanwhile, differential privacy (DP) techniques [22]–[24] can be further employed to provide theoretical privacy guarantees against attacks and prevent local private information from being leaked during the communication process.

Recently, some methods [12], [14], [25]–[29] have tried to integrate LoRA with FedAvg [20] in different applications. These methods finetune LoRA modules using local data of

This work was supported by the Hong Kong Research Grants Council under Grant 11212321, 11217922, and ECS-21212720, the HKSAR Innovation and Technology Commission (ITC) under ITF Project MHP/109/19, ITS/229/22, and the Science, Technology and Innovation Committee of Shenzhen under Grant SGDX20210823104001011. (**Corresponding authors: Yixuan Yuan (yxyuan@ee.cuhk.edu.hk), Jun Liu (dr.jun.liu@hku.hk)*)

M. Zhu is with Department of Mechanical Engineering, City University of Hong Kong; A. Mao is with School of Communication Engineering, Hangzhou Dianzi University, China; J. Liu is with Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong and was with Department of Mechanical Engineering, City University of Hong Kong; Y. Yuan is with Department of Electronic Engineering, Chinese University of Hong Kong and was with Department of Electrical Engineering, City University of Hong Kong.

https://doi.org/10.32388/I8WGTH

clients and then send the updated modules to the server. The server averages all received LoRA modules to obtain a global LoRA, and distributes it to all clients as the initialization of the next round. Despite the promising performance, these approaches neglect two key issues, as shown in Fig. 1. Firstly, the naive averaging of local LoRA modules leads to the **aggregation deviation** at the server side. In the FedAvg setting with LoRA, the local updates $\Delta\mathbf{W}$ of a client are decomposed into two low-rank matrices $\mathbf{A}$ and $\mathbf{B}$, $\Delta\mathbf{W} = \mathbf{BA}$. If $\mathbf{A}$ and $\mathbf{B}$ of all clients are aggregated independently, we will obtain a biased global update $\Delta\mathbf{W}_{\mathrm{g}} = \sum\mathbf{B}\sum\mathbf{A}$. Theoretically, the true global update should be computed via $\Delta\mathbf{W}_{\mathrm{g}}^{*} = \sum\Delta\mathbf{W} = \sum\mathbf{BA}$. Obviously, there exists a mathematical deviation $\Delta\mathbf{W}_{\mathrm{g}} \neq \Delta\mathbf{W}_{\mathrm{g}}^{*}$, which would severely impede the convergence of a federation system.

The second problem, i.e., **noise amplification effect**, lies in the client side and arises from the intrinsic “quadratic” architecture of LoRA. Differential privacy (DP) is a commonly-used technique in FL to provide privacy guarantee by adding noise (e.g., Gaussian noise) to client gradients against training data leakage from the shared model [30]. When the (Gaussian) noises $\boldsymbol{\xi}^{\mathbf{A}}$ and $\boldsymbol{\xi}^{\mathbf{B}}$ are injected into $\mathbf{A}$ and $\mathbf{B}$ of local LoRA modules, the “quadratic” architecture of LoRA would lead to the noise items $\mathbf{B}\boldsymbol{\xi}^{\mathbf{A}}$, $\boldsymbol{\xi}^{\mathbf{B}}\mathbf{A}$ and $\boldsymbol{\xi}^{\mathbf{B}}\boldsymbol{\xi}^{\mathbf{A}}$. In experiments, we observe that the noise intensities of the items $\mathbf{B}\boldsymbol{\xi}^{\mathbf{A}}$ and $\boldsymbol{\xi}^{\mathbf{B}}\mathbf{A}$ would be continuously amplified during training. The third term no longer follows a Gaussian distribution and also increases as the privacy budget of DP decreases. The noise amplification effect will hinder the model convergence when applying DP into a federated finetuning system with LoRA.

To overcome these problems, we propose a novel privacy-preserving federated finetuning (FedFT) framework called Deviation Eliminating and Noise Regulating (DEeR). The goal of DEeR is to adapt pretrained FMs to downstream medical tasks via LoRA in FL with client-level DP guarantees. Specifically, we firstly theoretically prove that the necessary condition to eliminate aggregation deviation is guaranteing the equivalence between LoRA parameters of clients. With the theoretical insight, we design a deviation eliminator at the server side, which utilizes the alternating minimization algorithm to iteratively optimize the parameters $\mathbf{A}$ and $\mathbf{B}$ of LoRA, ensuring that aggregation deviation always be zeros during training. Moreover, we also conduct an in-depth analysis of the noise amplification effect and find that this problem is mainly caused by “linear relationship” between DP noise and LoRA parameters. To suppress the noise amplification effect, we propose a noise regulator that exploits two regulator factors to decouple the relationship between DP and LoRA, thereby achieving robust privacy protection and excellent finetuning performance. The main contributions of this work are summarized as follows:

- This work in-depth analyzes the challenges of a privacy-preserving FedFT system with LoRA. To the best of our knowledge, it represents the first effort to adapt different pretrained FMs to various downstream medical tasks via FedFT with LoRA.
- We propose a deviation eliminator that utilizes the alternating minimization algorithm to optimize the parameters of LoRA to avoid aggregation deviation.
- We present a noise regulator that can exploit two regulator factors to decouple relationship between DP and LoRA to suppress the noise amplification effect.
- Extensive experiments are conducted on public datasets. The results demonstrate the superior performance of the proposed DEeR over state-of-the-arts and the efficacy of different components.

Roadmap. The rest of the paper is organized as follows. In Section II, we review previous methods focusing on PEFT and federated finetuning with LoRA. Some preliminary knowledge is presented in Section III. In Section IV, the proposed DEeR framework is introduced in detail. We describe implementation details, experimental settings and results in Section V. Finally, the paper is closed with the conclusion in Section VI.

II. Related Work

We introduce existing methods about parameter efficient fine tuning and federated finetuning with LoRA in this section.

A. Parameter Efficient Fine Tuning (PEFT)

Parameter efficient fine tuning enables efficient adaptation of foundation models to various downstream tasks without the need to finetune all parameters of FMs. It only optimizes a small subset of parameters and thus results in significant reductions in computation and storage costs. Existing PEFT methods can be broadly divided into three main categories.

The first category is dedicated to designing task-related Adapters [31], [32]. For example, VL-Adapter [31] inserts trainable adapter modules into a fixed CLIP model and finetunes only the adapters for vision-language tasks. SAN [32] presents a decoupled structure to reduce computational costs for semantic segmentation, *i.e.*, introducing an adapter network as the side branch of FMs. Prompt tuning falls into the second category. The prompt tuning [33] originally treats the prompts in NLP as task-specific continuous vectors and only optimizes them during finetuning, while visual prompt tuning [10] uses a set of continuous embeddings as visual prompts to pad the patch embeddings. However, both Adapter and Prompt tuning-based approaches introduce extra parameters and result in inference latency [34].

To solve this problem, the third type of works focus on reparameterization techniques, the most famous of which is the LoRA series [34]. The vanilla LoRA [9] optimizes rank decomposition matrices and re-parameterize the pretrained weight matrices. Numerous studies [35]–[39] have further developed and applied LoRA to various scenarios. For instance, considering that prespecifing a rank for all layers neglects the importance of different layers, AdaLoRA [36] dynamically allocates the rank for different layers by importance scoring. Additionally, LoRA Dropout [36] observes that finetuning LoRA-series models also face the risk of overfitting and thus introduces dropout technique to randomly drop rows and columns from tunable low-rank parameter matrices. Recent initiatives [38], [39] mainly focus on the composition of separate trained LoRAs to amplify performance across various tasks. For example, MOLE [38] treats each layer of trained

LoRAs as a distinct expert and learns a gating function to get composition weights to fuse these experts. Differing to existing approaches [35]–[39], this paper represents the first effort to develop LoRA into medical domain in a decentralized learning setting, aiming to achieve privacy-preserving federated PEFT.

B. Federated Finetuning with LoRA

The above PEFT approaches generally assume that training data comes from a data warehouse. In practice, however, data is often owned by multiple parties and is often prohibited from being shared with others, especially in medical domain. Recently, interest in the intersection of PEFT and FL has notably increased, forming a new research topic called FedFT. FedPETuning [26] provides a holistic empirical study of representative PEFT methods in FL. The experimental results show that the LoRA-based FedFT technique achieves a very promising performance and inference speed. Next, we review the existing LoRA-based FedFT methods.

Among current LoRA-based FedFT methods, a common solution is directly combining LoRA with FL to finetune FMs for various scenes, such as speech-to-text tasks [27] and personalized FL [25]. Nevertheless, this native way presents slow a convergence speed and leads to costly communication expenses. SLoRA [14] and FeDeRA [40] attribute the problem to random initialization of low-rank matrices. To this end, they use the singular value decomposition (SVD) to obtain better initialization from the pretrained full matrix. Besides, there is another problem that different layers of all client models should share varying ranks due to heterogeneous resources and data distributions. SA-FedLora [41] defines a scheduler function to adaptively adjust rank with communication round. In HETLORA [28], all clients first start from a global rank and then self-prune their respective ranks based on the magnitude of the model parameters. FlexLoRA [40] encourages clients to use different ranks during local training and upload full-size LoRA to a server. The server uses SVD to decouple the aggregated full-size matrix and distributes different sizes of LoRA to the clients. thereby achieving heterogeneous LoRA.

Apart from the above problems, FFA-LoRA [13] finds that the "quadratic" structure of LoRA incurs aggregation bias and introduces quadratic DP noise. To break the "quadratic" structure, FFA-LoRA fixes the randomly initialized non-zero matrices and only finetunes the zero-initialized matrices. However, freezing non-zero matrices will hinder the model from converging to a good local minimum, since random initialization is nearly impossible to produce optimal parameters for downstream tasks [42], [43]. This strategy makes FFA-LoRA very sensitive to different initialization. A bad initialization can degrade the model performance. Experiments in the previous method [44] and our paper also show the limited performance of FFA-LoRA. In addition, FFA-LoRA neglects the effect of linear noises. In this paper, we provide a more comprehensive study about these issues and an in-depth analyze the conditions to solve these problems.

III. Preliminaries

In this section, we present some background knowledge about LoRA, federated finetuning (FedFT) with LoRA, and differential privacy.

LoRA. As one of the most promising PEFT methods in the central setting, the key idea of LoRA [9] is decomposing the update $\Delta\mathbf{W} \in \mathbb{R}^{m\times n}$ of target module into low-rank matrices:

$$\mathbf{W}_0 + \Delta\mathbf{W} = \mathbf{W}_0 + \mathbf{B}\mathbf{A}, \tag{1}$$

where $\mathbf{W}_0 \in \mathbb{R}^{m\times n}$ denotes the pretrained weight matrix. $\mathbf{B} \in \mathbb{R}^{m\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times n}$ are the low-rank decomposition of $\Delta\mathbf{W}$, such that $\Delta\mathbf{W} = \mathbf{B}\mathbf{A}$. Typically, r is the rank of $\Delta\mathbf{W}$, $\mathbf{B}$, $\mathbf{A}$, and significantly smaller than m and n. During the finetuning phase, the model optimizes matrices $\mathbf{B}$ and $\mathbf{A}$ instead of directly updating $\mathbf{W}_0$, thus achieving the substantial reduction in GPU memory and storage usage. Additionally, to ensure the stable convergence, $\mathbf{B}$ and $\mathbf{A}$ use zero and random Gaussian initialization respectively, so that $\Delta\mathbf{W} = \mathbf{B}\mathbf{A}$ is zero at the beginning of training.

FedFT with LoRA. Current LoRA-based FedFT methods [12], [25]–[29] follow a standard FL setting, *i.e.*, FedAvg [20]. These methods collaboratively unite local LoRA modules of K clients to learn a global LoRA ($\mathbf{B}_\mathrm{g}$, $\mathbf{A}_\mathrm{g}$) as the global change $\Delta\mathbf{W}_\mathrm{g}$, enabling the pretrained knowledge $\mathbf{W}_0$ to adapt downstream tasks via multiple rounds of communication:

$$\mathbf{W}_\mathrm{g} = \mathbf{W}_0 + \Delta\mathbf{W}_\mathrm{g} = \mathbf{W}_0 + \mathbf{B}_\mathrm{g}\mathbf{A}_\mathrm{g}, \tag{2}$$

where $\mathbf{B}_\mathrm{g}$ and $\mathbf{A}_\mathrm{g}$ are obtained via the aggregation of local LoRA modules as follows,

$$\mathbf{B}_\mathrm{g} = 1/K \sum\nolimits_{k\in[K]} \mathbf{B}_k, \quad \mathbf{A}_\mathrm{g} = 1/K \sum\nolimits_{k\in[K]} \mathbf{A}_k. \tag{3}$$

The updated $\mathbf{B}_\mathrm{g}$ and $\mathbf{A}_\mathrm{g}$ are distributed back to clients as the initialization of local LoRA modules in the next round.

Differential Privacy Differential privacy (DP) is a popular manner to provide theoretical guarantees against training data leakage from the model in federated learning [22]. This work focuses on the client-level DP, aiming to ensure information security for any clients.

Definition 1. (Client-level DP) *A randomized algorithm $\mathcal{M}$ is (ε, δ)-DP if for any two adjacent datasets $\mathcal{D}$, $\mathcal{D}'$ constructed by adding or removing all records of any client, and every possible subset of outputs $\mathcal{S}$ in the range of $\mathcal{M}$ satisfy the following inequality:*

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}] \le e^{\varepsilon}\Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}] + \delta. \tag{4}$$

where the parameter ε is called the privacy budget and a smaller ε means a stronger privacy protection guarantee. The parameter δ defines the probability of failing to guarantee the differential privacy bound for any two adjacent datasets. At each round, each client first clips the local gradient with a norm constraint C. After clipping, we add Gaussian noise to the gradient before uploading it to the server [45], as follows:

$$\Delta\mathbf{W} = \Delta\mathbf{W} * \min(1, \frac{C}{\|\Delta\mathbf{W}\|_2}) + \mathcal{N}(0, \sigma^2 C^2 \cdot \mathbf{I}_d/K) \tag{5}$$

where σ is noise variance.

IV. Methodology

We first exhaustively analyze the challenges faced by FedFT with LoRA. Then, the overall framework and its submodules are introduced.

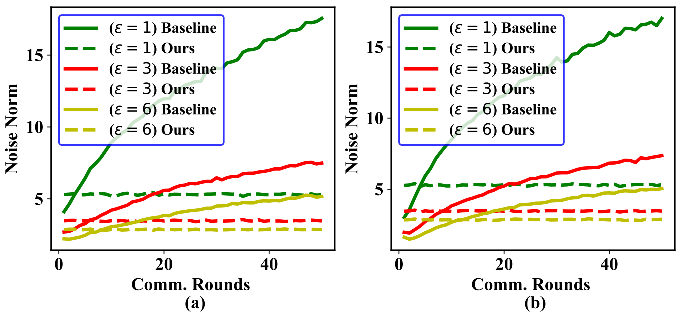


Fig. 2. The norm changes of linear noise items with communication round: (a) $\|\boldsymbol{\xi}_k^{\mathbf{B}}\mathbf{A}_k\|_F$, (b)$\|\mathbf{B}_k\boldsymbol{\xi}_k^{\mathbf{A}}\|_F$. (Best viewed in color.)

A. Problem Formulation

The native way to integrate LoRA and FedAvg in existing FedFT methods [12], [25]–[29] neglects the impact of "quadratic" architecture of LoRA. It would incur two intractable issues and hinder convergence of a federation system [13].

Aggregation Deviation. The first issue is aggregation deviation at the server side. Theoretically, the expected global model update $\Delta\mathbf{W}_{\mathrm{g}}$ in Eq. (2) should be calculated by averaging the uploaded updates $\{\Delta\mathbf{W}_k\}_{k=1}^K$ of clients [20]:

$$\Delta\mathbf{W}_{\mathrm{g}} = \frac{1}{K}\sum\nolimits_{k\in[K]}\Delta\mathbf{W}_k = \frac{1}{K}\sum\nolimits_{k\in[K]}\mathbf{B}_k\mathbf{A}_k. \tag{6}$$

However, according to Eq. (3), existing approaches [12], [25]–[29] aggregate $\mathbf{B}$ and $\mathbf{A}$ parts separately. Obviously, there exists an aggregation deviation as follows:

$$\Delta\mathbf{W}_{\mathrm{g}} = \frac{1}{K}\sum\nolimits_{k\in[K]}\mathbf{B}_k\frac{1}{K}\sum\nolimits_{k\in[K]}\mathbf{A}_k \neq \frac{1}{K}\sum\nolimits_{k\in[K]}\mathbf{B}_k\mathbf{A}_k. \tag{7}$$

Essentially, the deviation comes from the contradiction between model averaging in FL and the "quadratic" architecture of LoRA. In next section, we theoretically analyze that the deviation becomes larger when the data of clients are more heterogeneous.

Noise Amplification Effect. Another problem is that the "quadratic" architecture of LoRA would amplify DP noise. For the convenience of discussion, we omit the gradient clipping step and focus on one round of training. Generally, after local training of client k, the noise $\boldsymbol{\xi}_k \in \mathbb{R}^{m\times n}$ is sampled from Gaussian distribution and added to the local update $\Delta\mathbf{W}_k \in \mathbb{R}^{m\times n}$ to ensure client-level DP as follows:

$$\widetilde{\mathbf{W}}_k = \mathbf{W}_k + \boldsymbol{\xi}_k = \mathbf{W}_0 + (\Delta\mathbf{W}_k + \boldsymbol{\xi}_k), \tag{8}$$

where $\widetilde{\mathbf{W}}_k$ is the local model after adding noise. However, in FedTF with LoRA, we exploit LoRA to replace $\Delta\mathbf{W}_k$ and upload parameters $\mathbf{B}$ and $\mathbf{A}$ to the server. Therefore, we need to add Gaussian noise $\boldsymbol{\xi}_k^{\mathbf{B}} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{\xi}_k^{\mathbf{A}} \in \mathbb{R}^{r\times n}$ to $\mathbf{B}_k$ and $\mathbf{A}_k$ instead of $\Delta\mathbf{W}_k$:

$$\begin{aligned}\widetilde{\mathbf{W}}_k &= \mathbf{W}_0 + (\mathbf{B}_k + \boldsymbol{\xi}_k^{\mathbf{B}})(\mathbf{A}_k + \boldsymbol{\xi}_k^{\mathbf{A}}) \\ &= \mathbf{W}_0 + \mathbf{B}_k\mathbf{A}_k + \mathbf{B}_k\boldsymbol{\xi}_k^{\mathbf{A}} + \boldsymbol{\xi}_k^{\mathbf{B}}\mathbf{A}_k + \boldsymbol{\xi}_k^{\mathbf{B}}\boldsymbol{\xi}_k^{\mathbf{A}},\end{aligned} \tag{9}$$

where we call the third and fourth terms as linear noises, the final term as quadratic noise. FFA-LoRA [13] has shown that the quadratic noise becomes larger with the smaller privacy budget ε and hinders the convergence of the federated system.

Algorithm 1 The proposed DEeR algorithm for federated low-rank adaptation.

```
Server executes:
for each communication round $t$ do
    for each client $k = 1, 2, ..., K$ do
        Downloading $\mathbf{A}_{\mathrm{g}}^{(t)}$ to update $\mathbf{A}_k^{(t-1)}$ to $\mathbf{A}_k^{(t)}$ and freezing it.
        $\mathbf{B}_k^{(t+1)} \leftarrow$ ClientUpdate_B($\mathbf{W}_0, \mathbf{A}_{\mathrm{g}}^{(t)}, \mathbf{B}_k^{(t)}$).
    end for
    Aggregating $\mathbf{B}$: $\mathbf{B}_{\mathrm{g}}^{(t+1)} \leftarrow \sum_{k=1}^K \mathbf{B}_k^{(t+1)}$.
    for each client $k = 1, 2, ..., K$ do
        Downloading $\mathbf{B}_{\mathrm{g}}^{(t+1)}$ to update $\mathbf{B}_k^{(t)}$ to $\mathbf{B}_k^{(t+1)}$ and freezing it.
        $\mathbf{A}_k^{(t+1)} \leftarrow$ ClientUpdate_A($\mathbf{W}_0, \mathbf{A}_k^{(t)}, \mathbf{B}_{\mathrm{g}}^{(t+1)}$).
    end for
    Aggregating $\mathbf{A}$: $\mathbf{A}_{\mathrm{g}}^{(t+1)} \leftarrow \sum_{k=1}^K \mathbf{A}_k^{(t+1)}$.
end for
Client executes:
ClientUpdate_B($\mathbf{W}_0, \mathbf{A}_{\mathrm{g}}^{(t)}, \mathbf{B}_k^{(t)}$):
    for each epoch $e = 1, 2, ..., E$ do
        $\mathbf{B}_k^{(t+1)} \leftarrow \arg\min_{\mathbf{B}_k} f_k(\mathbf{W}_0, \mathbf{A}_{\mathrm{g}}^{(t)}, \mathbf{B}_k^{(t)})$.
    end for
    Based on Theorem 2, adding the modulated Gaussian noise into $\mathbf{B}_k^{(t+1)}$ before uploading it.
ClientUpdate_A($\mathbf{W}_0, \mathbf{A}_k^{(t)}, \mathbf{B}_{\mathrm{g}}^{(t+1)}$):
    for each epoch $e = 1, 2, ..., E$ do
        $\mathbf{A}_k^{(t+1)} \leftarrow \arg\min_{\mathbf{A}_k} f_k(\mathbf{W}_0, \mathbf{A}_k^{(t)}, \mathbf{B}_{\mathrm{g}}^{(t+1)})$.
    end for
    Based on Theorem 2, adding the modulated Gaussian noise into $\mathbf{A}_k^{(t+1)}$ before uploading it.
```

Yet, FFA-LoRA neglects the impact of linear noises. To reveal the characteristics of linear noises $\boldsymbol{\xi}_k^{\mathbf{B}}\mathbf{A}_k$ and $\mathbf{B}_k\boldsymbol{\xi}_k^{\mathbf{A}}$, we implement a baseline version of FedFT with LoRA [12], [25]–[29] on an endoscopic dataset, *i.e*, Kvasir-v2 [46], where the number of clients is 12 and the pretrained model is BiomedCLIP [3]. We randomly select one layer of LoRA in a client model and plot the Frobenius norm changes of its linear noises with the communication round, under different privacy budgets $\varepsilon \in \{1, 3, 6\}$, as shown in Fig. 2 (a) and (b). We can observe that: (1) The noise norms $\|\mathbf{B}_k\boldsymbol{\xi}_k^{\mathbf{A}}\|_F$ and $\|\boldsymbol{\xi}_k^{\mathbf{B}}\mathbf{A}_k\|_F$ continuously increase with the communication round for any privacy budgets; (2) The increase rates of $\|\mathbf{B}_k\boldsymbol{\xi}_k^{\mathbf{A}}\|_F$ and $\|\boldsymbol{\xi}_k^{\mathbf{B}}\mathbf{A}_k\|_F$ are greater when the privacy budget ε becomes smaller. These observations confirm that the "quadratic" architecture of LoRA can enlarge the original DP noise for a given privacy budget. This leads to the privacy guarantee shrinking since we need to increase the privacy budget to ensure model convergence.

B. Overview

We present Deviation Eliminator and Noise Regulator (DEeR), a privacy-preserving federated finetuning framework to adapt pretrained foundation models to downstream medical tasks via LoRA. Similar to previous works [12], [25]–[29], DEeR follows the standard FL setting, *i.e.*, FedAvg [20], and collaborates K clients to finetune a pretrained foundation model with the frozen parameters $\mathbf{W}_0$ and trainable LoRA parameters $\mathbf{A}$ and $\mathbf{B}$, as shown in Fig. 3. Each client holds a local medical dataset $\mathcal{D}^k = \{(\mathbf{x}_i^k, \mathbf{y}_i^k)\}$, where $\mathbf{x}_i^k$ denotes a training sample with the label $\mathbf{y}_i^k$. At the beginning of training,

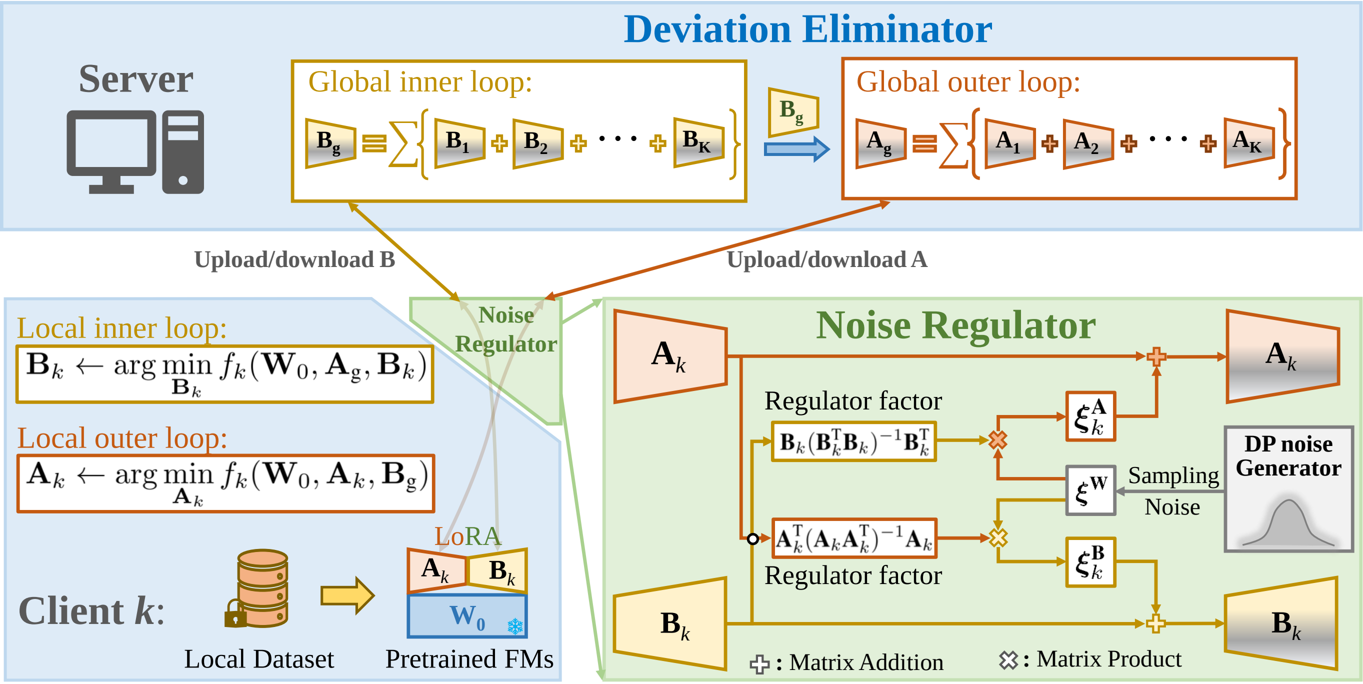

Deviation Eliminator
Server
Global inner loop:
$\mathbf{B}_g \equiv \sum\{\mathbf{B}_1 + \mathbf{B}_2 + \cdots + \mathbf{B}_K\}$
$\mathbf{B}_g$
Global outer loop:
$\mathbf{A}_g \equiv \sum\{\mathbf{A}_1 + \mathbf{A}_2 + \cdots + \mathbf{A}_K\}$
Upload/download B
Upload/download A
Local inner loop:
$\mathbf{B}_k \leftarrow \arg\min_{\mathbf{B}_k} f_k(\mathbf{W}_0, \mathbf{A}_g, \mathbf{B}_k)$
Local outer loop:
$\mathbf{A}_k \leftarrow \arg\min_{\mathbf{A}_k} f_k(\mathbf{W}_0, \mathbf{A}_k, \mathbf{B}_g)$
Noise Regulator
LoRA
$\mathbf{A}_k$
$\mathbf{B}_k$
$\mathbf{W}_0$
Client k:
Local Dataset
Pretrained FMs
Noise Regulator
$\mathbf{A}_k$
Regulator factor
$\mathbf{B}_k(\mathbf{B}_k^T\mathbf{B}_k)^{-1}\mathbf{B}_k^T$
$\xi_k^{\mathbf{A}}$
$\xi^{\mathbf{W}}$
Sampling Noise
DP noise Generator
$\mathbf{A}_k^T(\mathbf{A}_k\mathbf{A}_k^T)^{-1}\mathbf{A}_k$
Regulator factor
$\xi_k^{\mathbf{B}}$
$\mathbf{B}_k$
$\mathbf{A}_k$
$\mathbf{B}_k$
✚ : Matrix Addition
✖ : Matrix Product

Fig. 3. The overview of the proposed DEeR framework for federated finetuning with LoRA (Best viewed in color). DEeR is equipped with a deviation eliminator at the server side and a noise regulator at the client side. The deviation eliminator exploits alternating minimization algorithm to optimize the parameters $\mathbf{A}$ and $\mathbf{B}$ of LoRA for mitigating aggregation deviation. The noise regulator introduces two regulator factors to suppress noise amplification effect.

the server sends the parameters $\mathbf{W}_0$, $\mathbf{A}$ and $\mathbf{B}$ to clients. During the overall training process, $\mathbf{W}_0$ is fixed and only the parameters $\mathbf{A}$ and $\mathbf{B}$ are updated and uploaded to the server for aggregation. $\mathbf{A}$ and $\mathbf{B}$ have low communication costs since they are low-rank matrices. DEeR equips every client with a noise regulator to suppress the noise amplification effect caused by the “quadratic” architecture of LoRA during local training. Meanwhile, a deviation eliminator on the server side is used to schedule the optimization process of $\mathbf{A}$ and $\mathbf{B}$ to avoid the aggregation deviation. In addition, we provide the pseudocode to show the workflow of DEeR in Algorithm 1.

C. Deviation Eliminator

Although LoRA is favored by FedFT since its low-rank property and “quadratic” architecture result in the low communication costs and relatively low computational burdens, it also introduces the new challenge *i.e.*, aggregation deviation, as shown in Eq (7). To tackle this dilemma, we present a deviation eliminator that exploits alternating minimization algorithm to decouple the “quadratic” architecture of LoRA and achieve the robust optimization of the parameters $\mathbf{A}$ and $\mathbf{B}$, as demonstrated in Fig. 3.

To quantify the overall deviation $\mathcal{O}$ owing to LoRA aggregation on the server side, we define the deviation term based on Eq. (7) as below:

$$\mathcal{O} = \left|\frac{1}{K^2}\sum\nolimits_{k\in[K]}\mathbf{B}_k\sum\nolimits_{k\in[K]}\mathbf{A}_k - \frac{1}{K}\sum\nolimits_{k\in[K]}\mathbf{B}_k\mathbf{A}_k\right|. \tag{10}$$

We theoretically show that the overall deviation $\mathcal{O}$ can be eliminated when $\mathbf{A}$ or $\mathbf{B}$ of all clients are equivalent.

Theorem 1. *Given a collection of K clients, let $\mathbf{B}_k$, $\mathbf{A}_k$ and $\mathbf{B}_{k'}$, $\mathbf{A}_{k'}$ be the LoRA parameters of any two clients k and k', respectively. The overall aggregation deviation $\mathcal{O}$ will be zero when $\mathbf{B}_k$ and $\mathbf{B}_{k'}$ or $\mathbf{A}_k$ and $\mathbf{A}_{k'}$ are equivalent in a FedTF system with LoRA.*

Proof of Theorem 1.

$$\begin{aligned}
\mathcal{O} &= \left|\frac{1}{K^2}\sum\nolimits_{k\in[K]}\mathbf{B}_k\sum\nolimits_{k\in[K]}\mathbf{A}_k - \frac{1}{K}\sum\nolimits_{k\in[K]}\mathbf{B}_k\mathbf{A}_k\right| \\
&= \left|\frac{1}{K^2}\Big(\sum\nolimits_{k\in[K]}\mathbf{B}_k\sum\nolimits_{k\in[K]}\mathbf{A}_k - K\sum\nolimits_{k\in[K]}\mathbf{B}_k\mathbf{A}_k\Big)\right| \\
&= \left|\frac{1}{K^2}\left[\sum\nolimits_{k'\in[K]}\left(\mathbf{B}_{k'}\sum\nolimits_{k\in[K]}\mathbf{A}_k - \sum\nolimits_{k\in[K]}\mathbf{B}_k\mathbf{A}_k\right)\right]\right| \\
&= \left|\frac{1}{K^2}\left[\sum\nolimits_{k'\in[K]}\left(\sum\nolimits_{k\in[K]}\mathbf{B}_{k'}\mathbf{A}_k - \sum\nolimits_{k\in[K]}\mathbf{B}_k\mathbf{A}_k\right)\right]\right| \\
&= \left|\frac{1}{K^2}\left[\sum\nolimits_{k'\in[K]}\ \sum\nolimits_{k\in[K]}(\mathbf{B}_{k'} - \mathbf{B}_k)\mathbf{A}_k\right]\right| \\
&\quad //\text{Using the similar derivation process.}// \\
&= \left|\frac{1}{K^2}\left[\sum\nolimits_{k\in[K]}\ \sum\nolimits_{k'\in[K]}\mathbf{B}_k(\mathbf{A}_{k'} - \mathbf{A}_k)\right]\right|. \qquad \square
\end{aligned}$$

We can find that the deviation $\mathcal{O}$ will be zero when $\mathbf{B}$ or $\mathbf{A}$ of all clients are equivalent. Additionally, data heterogeneity has a significant impact on $\mathcal{O}$, since it increases the divergence between ($\mathbf{A}$, $\mathbf{B}$) of different clients.

With the insight of **Theorem 1**, we introduce a constraint condition into the optimization objective of a FedFT system to ensure the equivalence between $\mathbf{A}$ or $\mathbf{B}$ of any two clients:

$$\begin{aligned}&\min_{\mathbf{A}_{\mathrm{g}},\mathbf{B}_{\mathrm{g}}}\frac{1}{K}\sum\nolimits_{k\in[K]} f_k(\mathbf{W}_0,\mathbf{A}_{\mathrm{g}},\mathbf{B}_{\mathrm{g}}),\\ &\text{s.t.}\sum\nolimits_{k\in[K]}\sum\nolimits_{k'\in[K]}(\mathbf{B}_k-\mathbf{B}_{k'})(\mathbf{A}_k-\mathbf{A}_{k'})=0,\end{aligned} \tag{11}$$

where f_k is the loss function of the client k. The global $\mathbf{A}_{\mathrm{g}}$ and $\mathbf{B}_{\mathrm{g}}$ are obtained by Eq. (3). The objective in Eq. (11) is equivalent to minimizing the following loss function for any client k as follows:

$$\min_{\mathbf{A}_k,\mathbf{B}_k} f_k(\mathbf{W}_0,\mathbf{A}_k,\mathbf{B}_k),\text{s.t.,}\sum_{k'\in[K]}(\mathbf{B}_k-\mathbf{B}_{k'})(\mathbf{A}_k-\mathbf{A}_{k'})=0. \tag{12}$$

Eq. (12) cannot be directly optimized because ($\mathbf{A}$, $\mathbf{B}$) of each client are unavailable to each other during local training. To enable the optimization objective to be separable across clients, we employ the generalized Alternating Minimization (gAM) optimization algorithm [47]. Specifically, at the t-th round, we firstly update ($\mathbf{A}_k^{(t-1)}$, $\mathbf{B}_k^{(t-1)}$) to ($\mathbf{A}_k^{(t)}$, $\mathbf{B}_k^{(t)}$) using the global ($\mathbf{A}_{\mathrm{g}}^{(t)}$, $\mathbf{B}_{\mathrm{g}}^{(t)}$) downloaded from the server, and then freeze $\mathbf{A}_k^{(t)}$ and only optimize $\mathbf{B}_k^{(t)}$ as follows:

$$\mathbf{B}_k^{(t+1)}\leftarrow\arg\min_{\mathbf{B}_k} f_k(\mathbf{W}_0,\mathbf{A}_{\mathrm{g}}^{(t)},\mathbf{B}_k^{(t)}). \tag{13}$$

Since $\mathbf{A}^{(t)}$ of all clients equal to $\mathbf{A}_{\mathrm{g}}^{(t)}$, the constraint condition are satisfied in Eq. (12). Next, the obtained $\{\mathbf{B}_k^{(t+1)}\}_{k=1}^{K}$ are delivered to the server and aggregated to get $\mathbf{B}_{\mathrm{g}}^{(t+1)}$, which is distributed back to all clients to update $\mathbf{B}_k^{(t)}$ to $\mathbf{B}_k^{(t+1)}$. We fix $\mathbf{B}_k^{(t+1)}$ and only optimize $\mathbf{A}_{\mathrm{k}}^{(t)}$ as follows:

$$\mathbf{A}_k^{(t+1)}\leftarrow\arg\min_{\mathbf{A}_k} f_k(\mathbf{W}_0,\mathbf{A}_k^{(t)},\mathbf{B}_{\mathrm{g}}^{(t+1)}). \tag{14}$$

Similarly, $\mathbf{B}^{(t+1)}$ of all clients are equivalent, so the constraint condition is also satisfied in Eq. (12).

The proposed deviation eliminator can exploit the gAM optimization algorithm to ensure the equivalence between $\mathbf{A}$ or $\mathbf{B}$ of any two clients for each round of aggregation. Therefore, the overall aggregation deviation $\mathcal{O}$ is always zero, thereby guaranteeing the stable convergence of the model. Notably, FFA-LoRA [13] fixes randomly-initialized $\mathbf{A}$ and only optimizes $\mathbf{B}$ during the overall training process, which can be regarded as a special case of our method. However, randomly-initialized $\mathbf{A}$ is not certainly optimal and limits model convergence. In contrast, our approach optimizes $\mathbf{A}$ and $\mathbf{B}$ through gAM algorithm and is therefore more likely to converge to a better local optimum.

D. Noise Regulator

Differential privacy (DP) techniques provide more stringent privacy protection to a FedFT system against potential privacy leaks. However, when directly introducing DP into a LoRA-based FedFT system, the "quadratic" architecture of LoRA would amplify DP noise, showing a significant negative impact on the model convergence and final finetuning performance. To suppress the noise amplification effect, we propose a novel noise regulator that uses two regulator factors to decouple the relationship between DP and LoRA, thereby ensuring robust privacy protection while maintaining superior finetuning performance, as illustrated in Fig. 3.

As we discussed earlier about the noise amplification effect, LoRA transfers the original DP noise into two types, *i.e.*, linear noises and quadratic noise, as demonstrated in Eq. (9). Because the "quadratic" structure of LoRA has been decoupled by the proposed deviation eliminator, the quadratic noise is eliminated [13]. Concretely, after optimizing $\mathbf{A}$ and $\mathbf{B}$ via Eq. (13) and Eq. (14) respectively, we inject DP noise into them before sending them to the server as follows:

$$\mathbf{W}_0+(\mathbf{B}_k+\boldsymbol{\xi}_k^{\mathbf{B}})\mathbf{A}_k=\mathbf{W}_0+\mathbf{B}_k\mathbf{A}_k+\boldsymbol{\xi}_k^{\mathbf{B}}\mathbf{A}_k, \tag{15}$$

$$\mathbf{W}_0+\mathbf{B}_k(\mathbf{A}_k+\boldsymbol{\xi}_k^{\mathbf{A}})=\mathbf{W}_0+\mathbf{B}_k\mathbf{A}_k+\mathbf{B}_k\boldsymbol{\xi}_k^{\mathbf{A}}, \tag{16}$$

where we omit the superscript t for convenience. In Eq. (15), $\mathbf{A}_k=\mathbf{A}_{\mathrm{g}}$, and $\mathbf{B}_k=\mathbf{B}_{\mathrm{g}}$ in Eq. (16). Although the quadratic noise disappears, the linear noises $\boldsymbol{\xi}_k^{\mathbf{B}}\mathbf{A}_k$ and $\mathbf{B}_k\boldsymbol{\xi}_k^{\mathbf{A}}$ are amplified with the communication round and still affect the learning of $\mathbf{A}_k$ and $\mathbf{B}_k$.

Next, we conduct an in-depth analysis of $\boldsymbol{\xi}_k^{\mathbf{B}}\mathbf{A}_k$ and $\mathbf{B}_k\boldsymbol{\xi}_k^{\mathbf{A}}$. With a given privacy budget ε, we can obtain the corresponding Gaussian noise distribution by privacy composition rules [48]. Theoretically, if we sample DP noises $\boldsymbol{\xi}_k^{\mathbf{B}}\in\mathbb{R}^{m\times r}$ and $\boldsymbol{\xi}_k^{\mathbf{A}}\in\mathbb{R}^{r\times n}$ from the distribution, their norm values do not change significantly during the training process. Hence, the amplification of $\boldsymbol{\xi}_k^{\mathbf{B}}\mathbf{A}_k$ and $\mathbf{B}_k\boldsymbol{\xi}_k^{\mathbf{A}}$ is only related to $\mathbf{A}_k$ and $\mathbf{B}_k$, respectively. Next, we theoretically prove that the amplification effect can be removed by introducing two regulator factors.

Theorem 2. *Assuming that $\mathbf{B}_k\in\mathbb{R}^{m\times r}$ and $\mathbf{A}_k\in\mathbb{R}^{r\times n}$ are LoRA parameters of the client k. Let $\boldsymbol{\xi}^{\mathbf{W}}\in\mathbb{R}^{m\times n}$ be DP noise sampled from a Gaussian distribution. $\mathbf{A}_k^T(\mathbf{A}_k\mathbf{A}_k^T)^{-1}$ and $(\mathbf{B}_k^T\mathbf{B}_k)^{-1}\mathbf{B}_k^T$ are two regulator factors. Imposing the noises $\boldsymbol{\xi}^{\mathbf{W}}\mathbf{A}_k^T(\mathbf{A}_k\mathbf{A}_k^T)^{-1}$ and $(\mathbf{B}_k^T\mathbf{B}_k)^{-1}\mathbf{B}_k^T\boldsymbol{\xi}^{\mathbf{W}}$ to $\mathbf{B}_k$ and $\mathbf{A}_k$ respectively can mitigate the noise amplification effect and ensure robust privacy protection.*

Proof of Theorem 2. Given the specific Gaussian distribution for the privacy budget ε, the noise terms $\boldsymbol{\xi}_k^{\mathbf{B}}\mathbf{A}_k\in\mathbb{R}^{m\times n}$ and $\mathbf{B}_k\boldsymbol{\xi}_k^{\mathbf{A}}\in\mathbb{R}^{m\times n}$ are expected to follow this distribution. To obtain $\boldsymbol{\xi}_k^{\mathbf{B}}$ and $\boldsymbol{\xi}_k^{\mathbf{A}}$ satisfying the condition, we solve the following least-squares problems:

$$\begin{aligned}&\min_{\boldsymbol{\xi}_k^{\mathbf{B}}}\|\boldsymbol{\xi}_k^{\mathbf{B}}\mathbf{A}_k-\boldsymbol{\xi}^{\mathbf{W}}\|^2;\quad\min_{\boldsymbol{\xi}_k^{\mathbf{A}}}\|\mathbf{B}_k\boldsymbol{\xi}_k^{\mathbf{A}}-\boldsymbol{\xi}^{\mathbf{W}}\|^2;\\ &\text{s.t.,}\mathbf{B}_k,\boldsymbol{\xi}_k^{\mathbf{B}}\in\mathbb{R}^{m\times r},\mathbf{A}_k,\boldsymbol{\xi}_k^{\mathbf{A}}\in\mathbb{R}^{r\times n},\boldsymbol{\xi}^{\mathbf{W}}\in\mathbb{R}^{m\times n},\end{aligned} \tag{17}$$

where $\boldsymbol{\xi}^{\mathbf{W}}\in\mathbb{R}^{m\times n}$ is sampled from the Gaussian distribution. Considering that $\mathbf{B}_k$ and $\mathbf{A}_k$ are singular matrices, we can compute their pseudo-inverses via the singular value decomposition (SVD) [49] and obtain final solutions to the above problems [50]: $\boldsymbol{\xi}_k^{\mathbf{B}\star}=\boldsymbol{\xi}^{\mathbf{W}}\mathbf{A}_k^{\mathrm{T}}(\mathbf{A}_k\mathbf{A}_k^{\mathrm{T}})^{-1}$ and $\boldsymbol{\xi}_k^{\mathbf{A}\star}=(\mathbf{B}_k^{\mathrm{T}}\mathbf{B}_k)^{-1}\mathbf{B}_k^{\mathrm{T}}\boldsymbol{\xi}^{\mathbf{W}}$. Here, we refer to $\mathbf{A}_k^{\mathrm{T}}(\mathbf{A}_k\mathbf{A}_k^{\mathrm{T}})^{-1}$ and $(\mathbf{B}_k^{\mathrm{T}}\mathbf{B}_k)^{-1}\mathbf{B}_k^{\mathrm{T}}$ as regulator factors. We apply $\boldsymbol{\xi}_k^{\mathbf{B}\star}$ and $\boldsymbol{\xi}_k^{\mathbf{A}\star}$ into Eq. (15) and Eq. (16), respectively:

$$\begin{aligned}\mathbf{W}_0+\left[\mathbf{B}_k+\boldsymbol{\xi}^{\mathbf{W}}\mathbf{A}_k^{\mathrm{T}}(\mathbf{A}_k\mathbf{A}_k^{\mathrm{T}})^{-1}\right]\mathbf{A}_k&=\mathbf{W}_0+\mathbf{B}_k\mathbf{A}_k+\boldsymbol{\xi}^{\mathbf{W}},\\ \mathbf{W}_0+\mathbf{B}_k\left[\mathbf{A}_k+(\mathbf{B}_k^{\mathrm{T}}\mathbf{B}_k)^{-1}\mathbf{B}_k^{\mathrm{T}}\boldsymbol{\xi}^{\mathbf{W}}\right]&=\mathbf{W}_0+\mathbf{B}_k\mathbf{A}_k+\boldsymbol{\xi}^{\mathbf{W}},\end{aligned} \tag{18}$$

where $\mathbf{A}_k^{\mathrm{T}}(\mathbf{A}_k\mathbf{A}_k^{\mathrm{T}})^{-1}\mathbf{A}_k = \mathbf{B}_k(\mathbf{B}_k^{\mathrm{T}}\mathbf{B}_k)^{-1}\mathbf{B}_k^{\mathrm{T}} = \mathbf{I} \in \mathbb{R}^{m\times n}$. Since $\boldsymbol{\xi}^{\mathbf{W}}$ does not undergo significant change during training, the amplification of linear noises is suppressed. □

In Fig. 2, we demonstrate the norms of noise terms $\|\mathbf{B}_k\boldsymbol{\xi}_k^{\mathbf{A}}\|_F$ and $\|\boldsymbol{\xi}_k^{\mathbf{B}}\mathbf{A}_k\|_F$ for our method in different communication rounds. It can be observed that the noise norms have slight fluctuations and do not present an increasing trend for any privacy budgets. Therefore, the noise amplification effect is removed with the synergy between the proposed deviation eliminator and noise regulator.

V. Experiments

To investigate the effectiveness of the proposed DEeR, we evaluate it on two medical classification datasets (OCT-C8 [51] and Kvasir-v2 [46]) and two medical segmentation datasets (M&MS [52] and polyp segmentation [53]).

A. Datasets

1) OCT-C8: OCT-C8 [51] contains 24000 retinal OCT images, which belong to eight categories, *i.e.*, age related macular degeneration, choroidal neovascularisation, diabetic macular edema, drusen, macular hole, diabetic retinopathy, central serous retinopathy and one for healthy class. Based on the official division, 18400 images are used for training, 2800 images for validation, and 2800 images for testing.

2) Kvasir-v2: We collect 8000 endoscopic images of the gastrointestinal tract from Kvasir-v2 dataset [46]. These samples are divided into eight classes according to the types of anatomical landmarks and phatological findings. We use the ratio of 7 : 1 : 2 to randomly partition these samples into training, validation, and test sets.

3) M&MS: We gather 317 cardiac magnetic resonance scans from different patients from M&MS [52]. These scans were scanned in clinical centers in three countries (Spain, Germany and Canada) using four different scanner vendors (Siemens, General Electric, Philips and Canon). Each scan is segmented into background area, left ventricular myocardium, left and right ventricle blood pools. We divide these scans into four clients based on the vendor type. The scans of each client are randomly partitioned into training, validation, and test sets with a ratio of 7 : 1 : 2. All 3D volumes are sliced into images with the axial plane.

4) Polyp Segmentation Dataset: The data are collected from four public datasets, CVC-ClinicDB [54], CVC-ColonDB [55], ETIS [56] and Kvasir [57]. Following the study [53], we adopt the 900 and 550 images from ClinicDB and Kvasir datasets as the training set. The remaining 64 images of ClinicDB dataset and 100 images of Kvasir dataset belong to the test set. In addition, ETIS and CVC-ColonDB datasets contain 380 images and 196 images, respectively, which are totally divided into the test set to verify the generalization ability of a model. We randomly and evenly divided training images into four clients.

B. Experiment Setup

1) Implementation Details: The proposed DEeR and comparison methods are implemented with PyTorch library. For classification datasets, BiomedCLIP [3] is regarded as the foundation model. The number K of clients is set to 12. We keep the total communication rounds to 50 and the local steps to 5. The total batch-sizes are set to 128 and 512 for Kvasir-v2 and OCT-C8, respectively. We use Dirichlet distribution on label ratios to simulate Non-IID settings. The Dirichlet parameter β defaults to 0.1. For segmentation datasets, we use SAM-Med2D [58] as the foundation model and box as prompt. We keep the total communication rounds to 50 and the local steps to 3. The total batch-size is set to 32 for M&MS dataset and 128 for polyp segmentation dataset. For all datasets, we use the SGD optimizer and choose the best learning rate from $[0.1, 0.01, 0.001]$ by FedAvg with LoRA. Both the rank r and scaling factor α default to 8. For privacy parameters, the privacy failure probability $\delta = \frac{1}{K}$. The privacy budget ε defaults to 3 for Kvasir-v2 and 0.1 for OCT-8 and M&MS. We use the privacy accountant from Opacus [59] to calculate the noise scale σ in all experiments. The clipping threshold C is selected by grid search from set $[0.1, 0.2, 0.3, 0.4, 0.6]$.

2) Evaluation Metrics: Two commonly-used metrics, accuracy, and F1-score, are used to measure the classification performance. To evaluate segmentation performance, we adopt two commonly-used metrics of Dice similarity coefficient and mean intersection over union (IoU) of foreground and background. In all the experiments, we conduct three trials for each setting and present the mean and the standard deviation.

C. Comparisons with State-of-the-Art Methods

We compare DEeR with three baselines on different medical tasks. (1) LoRA: We implement the original LoRA [9] based on FedAvg [20]. (2) FFA-LoRA [13]: It freezes $\mathbf{A}$ and only finetunes $\mathbf{B}$ of LoRA in FedAvg. (3) DP-DyLoRA [60]: it adjusts the rank r of LoRA layers randomly during training, in the range of $r \in [r_{\min}, r_{\max}]$. During testing, we reported the best results among these ranks.

1) Evaluation on Medical Classification Tasks: To verify efficiency of DEeR for medical classification tasks, we compare performance of DEeR and baseline methods, under different privacy budgets $\varepsilon \in [1.0, 3.0, 6.0]$ for Kvasir-v2 dataset and $\varepsilon \in [0.1, 0.5, 1.0]$ for OCT-8 dataset, as shown in Table I. For Kvasir-v2 dataset, LoRA yields the second-best performance and undergoes a severe performance degradation as ε becomes smaller, especially F1-score with a decrement of 4.17%. Although FFA-LoRA presents relatively stable performance against varying privacy budgets, it obtains the lowest accuracy and F1-score. By comparison, DEeR implements the highest accuracy and F1-score and also shows consistent performance for different privacy budgets. In experiments, we also observe that recall scores of all cases in DEeR are higher than 80%. Significant difference is found in LoRA and DEeR for $\varepsilon = 1.0$ (P-value < 0.005) and $\varepsilon = 6.0$ (P-value < 0.05). For OCT-8 dataset, both LoRA and FFA-LoRA present the high sensitivity to ε. For instance, when the budget ε decreases from infinity (without DP) to 0.1, they have enormous performance drops with decrements of 49.92% and 53.5%, 28.65% and 32.58% in accuracy and F1-score, respectively. Noticeably, DEeR merely suffers from a slight

TABLE I
THE PERFORMANCE COMPARISON OF DIFFERENT METHODS ON TWO CLASSIFICATION DATASETS.

Datasets	Priv. Budget	LoRA		FFA-LoRA		DP-DyLoRA		DEeR	
		Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
OCT-8	Without DP	92.07 ± 1.24	92.09 ± 1.23	84.86 ± 2.07	84.87 ± 2.14	92.67 ± 1.38	92.66 ± 1.37	**92.93** ± 0.82	**92.95** ± 0.82
	$\varepsilon = 1.0$	80.92 ± 2.42	79.77 ± 3.11	76.91 ± 4.15	76.65 ± 4.17	79.26 ± 3.89	77.72 ± 5.07	**92.33** ± 0.55	**92.35** ± 0.55
	$\varepsilon = 0.5$	83.39 ± 3.88	82.30 ± 5.03	74.53 ± 4.39	72.57 ± 5.54	61.70 ± 1.89	58.36 ± 4.24	**91.20** ± 0.95	**91.21** ± 0.97
	$\varepsilon = 0.1$	42.15 ± 2.88	38.59 ± 3.60	56.20 ± 4.63	52.29 ± 6.10	23.55 ± 4.61	19.30 ± 3.40	**84.28** ± 3.74	**83.20** ± 4.59
Kvasir-v2	Without DP	85.46 ± 1.39	85.18 ± 2.00	80.73 ± 1.08	79.90 ± 1.56	76.10 ± 5.59	74.15 ± 6.92	**86.29** ± 0.65	**86.11** ± 0.65
	$\varepsilon = 6.0$	84.58 ± 0.57	84.25 ± 0.58	79.23 ± 0.87	78.14 ± 1.03	78.85 ± 3.92	77.57 ± 5.50	**87.00** ± 0.87	**86.84** ± 0.91
	$\varepsilon = 3.0$	84.85 ± 0.86	84.51 ± 0.86	79.21 ± 0.62	77.94 ± 0.80	76.93 ± 1.40	75.17 ± 2.68	**86.90** ± 0.75	**86.70** ± 0.85
	$\varepsilon = 1.0$	82.00 ± 0.82	81.01 ± 1.11	79.06 ± 1.35	77.90 ± 1.51	76.37 ± 2.45	73.69 ± 3.86	**86.56** ± 0.53	**86.33** ± 0.63

TABLE II
THE PERFORMANCE COMPARISON OF DIFFERENT METHODS ON CARDIAC IMAGE SEGMENTATION DATASET.

Priv. Budget	Clients	LoRA		FFA-LoRA		DP-DyLoRA		DEeR	
		IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice
$\varepsilon = 1.0$	Canon	73.73 ± 1.77	82.99 ± 1.57	76.00 ± 0.78	84.72 ± 0.80	74.87 ± 1.43	84.19 ± 1.40	**77.22** ± 1.32	**85.63** ± 1.21
	GE	73.32 ± 1.59	82.89 ± 1.61	74.98 ± 0.85	83.83 ± 0.76	73.90 ± 1.06	83.52 ± 0.98	**76.37** ± 1.39	**84.97** ± 1.33
	Philips	75.54 ± 0.34	84.83 ± 0.36	77.28 ± 0.69	86.03 ± 0.61	74.40 ± 0.18	84.02 ± 0.13	**78.83** ± 0.04	**87.31** ± 0.11
	Siemens	75.15 ± 1.03	83.91 ± 1.06	77.26 ± 1.19	85.47 ± 1.19	73.71 ± 0.58	82.96 ± 0.24	**78.45** ± 1.06	**86.51** ± 1.03
$\varepsilon = 0.1$	Canon	69.65 ± 0.77	79.94 ± 0.57	76.41 ± 1.56	84.96 ± 1.48	70.02 ± 1.31	80.37 ± 1.21	**77.52** ± 1.68	**85.87** ± 1.49
	GE	69.18 ± 1.90	79.82 ± 1.69	74.83 ± 1.81	83.78 ± 1.70	70.08 ± 1.25	80.73 ± 1.31	**75.37** ± 1.72	**84.33** ± 1.41
	Philips	70.62 ± 0.81	81.04 ± 0.56	77.02 ± 0.23	85.94 ± 0.17	72.22 ± 0.58	82.10 ± 0.42	**78.02** ± 0.25	**86.65** ± 0.18
	Siemens	69.75 ± 1.25	79.97 ± 1.36	76.72 ± 0.73	85.11 ± 0.89	71.03 ± 1.48	80.96 ± 1.20	**78.18** ± 0.84	**86.43** ± 0.78

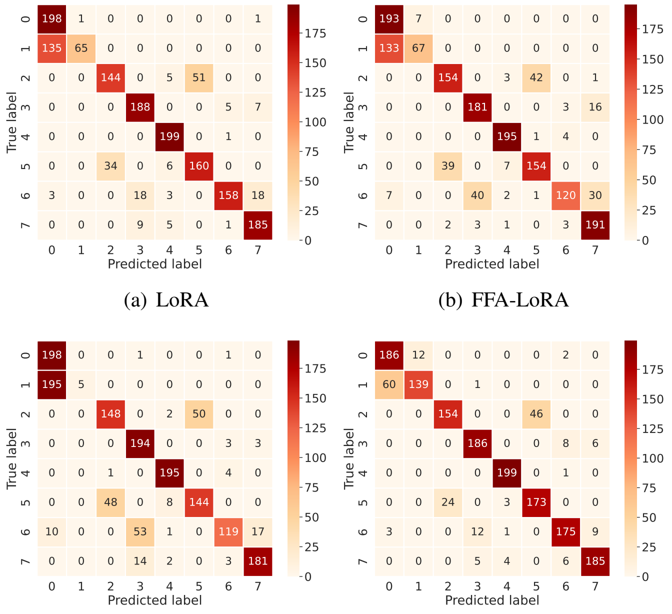
TABLE III
THE PERFORMANCE COMPARISON OF DIFFERENT METHODS ON POLYP SEGMENTATION DATASET UNDER DIFFERENT PRIVACY BUDGETS.

Methods	$\varepsilon = 1.0$		$\varepsilon = 0.1$	
	IoU (%)	Dice (%)	IoU (%)	Dice (%)
LoRA	78.83 ± 0.08	87.17 ± 0.03	75.98 ± 0.62	85.44 ± 0.31
FFA-LoRA	80.96 ± 0.04	88.60 ± 0.03	79.91 ± 0.18	87.84 ± 0.07
DP-DyLoRA	73.84 ± 1.00	83.90 ± 0.43	75.00 ± 1.01	84.69 ± 0.43
DEeR	**81.50** ± 0.22	**88.92** ± 0.13	**80.60** ± 0.15	**88.38** ± 0.04

drop (8.65%) and (9.75%) and outperforms LoRA and FFA-LoRA (P-value < 0.0005 for $\varepsilon = 0.1$ and P-value < 0.005 for $\varepsilon = 1.0$). DP-DyLoRA can perform well without DP noise, but performs poorly once imposing noise and also presents a higher sensitivity to ε than DEeR. Although DP-DyLoRA presents relatively stable performance as DEeR against varying privacy budgets, it obtains low accuracy and F1-score. The results on two datasets prove that DEeR can achieve superior finetuning performance while providing stronger privacy guarantees than existing methods for medical classification tasks.

We further visualize the confusion matrices of the previous methods and our DEeR on the endoscopy dataset, as shown in Fig. 4. We can observe that DP-DyLoRA, LoRA and FFA-LoRA misclassify 97.5%, 67.5% and 66.5% the class 1 into the class 0 due to the narrow intra-class distance, respectively. Meanwhile, they also have high errors for the class 5 and 6. By comparison, DEeR achieves higher precision in these classes, especially for the class 1. The experimental results can confirm the effectiveness of the proposed finetuning method.

2) Evaluation on Medical Segmentation Tasks: We compare DEeR with baseline methods on M&MS and polyp segmentation datasets with privacy budgets $\varepsilon \in [0.1, 1.0]$. For M&MS in Table II, it is observed that the performance of LoRA is fragile for the budget ε, since its IoU and Dice on all clients suffer from remarkable decreases when ε declines from 1 to 0.1. Different from classification tasks, FFA-LoRA outperforms LoRA on segmentation tasks. One possible reason is that the segmentation model is more sensitive to noise. Compared with LoRA, FFA-LoRA is not affected by



(a) LoRA (b) FFA-LoRA

(c) DP-DyLoRA (d) Ours

Fig. 4. The confusion matrices of different methods on Kvasir-v2.

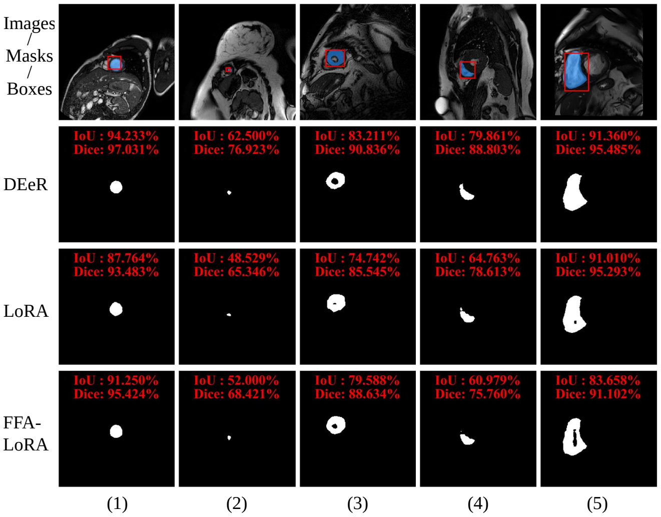

Images / Masks / Boxes
DEeR
IoU : 94.233% Dice: 97.031%
IoU : 62.500% Dice: 76.923%
IoU : 83.211% Dice: 90.836%
IoU : 79.861% Dice: 88.803%
IoU : 91.360% Dice: 95.485%
LoRA
IoU : 87.764% Dice: 93.483%
IoU : 48.529% Dice: 65.346%
IoU : 74.742% Dice: 85.545%
IoU : 64.763% Dice: 78.613%
IoU : 91.010% Dice: 95.293%
FFA-LoRA
IoU : 91.250% Dice: 95.424%
IoU : 52.000% Dice: 68.421%
IoU : 79.588% Dice: 88.634%
IoU : 60.979% Dice: 75.760%
IoU : 83.658% Dice: 91.102%
(1)
(2)
(3)
(4)
(5)

Fig. 5. Visualization of segmentation results for different methods on M&MS dataset. The columns (1)-(3) correspond to $\varepsilon = 0.1$ and columns (4)-(5) correspond to $\varepsilon = 1.0$.

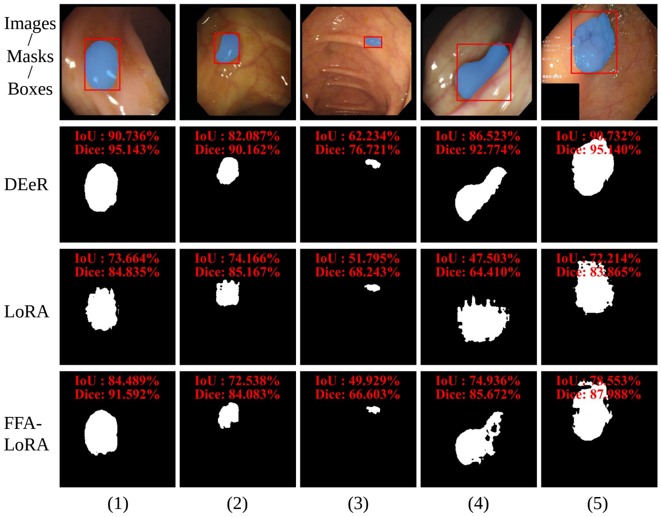

Images / Masks / Boxes
DEeR
IoU : 90.736% Dice: 95.143%
IoU : 82.087% Dice: 90.162%
IoU : 62.234% Dice: 76.721%
IoU : 86.523% Dice: 92.774%
IoU : 90.732% Dice: 95.140%
LoRA
IoU : 73.664% Dice: 84.835%
IoU : 74.166% Dice: 85.167%
IoU : 51.795% Dice: 68.243%
IoU : 47.503% Dice: 64.410%
IoU : 72.214% Dice: 83.865%
FFA-LoRA
IoU : 84.489% Dice: 91.592%
IoU : 72.538% Dice: 84.083%
IoU : 49.929% Dice: 66.603%
IoU : 74.936% Dice: 85.672%
IoU : 78.553% Dice: 87.988%
(1)
(2)
(3)
(4)
(5)

Fig. 6. Visualization of segmentation results for different methods on polyp segmentation dataset.

"quadratic" noise. Nonetheless, it is still inferior to DEeR in terms of performance for different privacy budgets ε since it neglects the impact of "linear" noise. The comparison results of polyp segmentation are demonstrated in Table III. LoRA and DP-DyLoRA [60] show the limited performance for different privacy budget ε, since they ignore aggregation deviation and noise amplification effect problems. In contrast, FFA-LoRA exploits a simple freezing strategy to address these problems and achieves the better performance. Notably, DEeR outperforms FFA-LoRA for any ε and yields approximating 90% of dice scores. These results confirm the priority of DEeR for medical segmentation tasks in contrast to the state-of-the-art methods.

Furthermore, we visualize the segmentation results of DEeR and the state-of-the-art methods under different privacy budgets ε, as shown in Fig. 5. The simple case of the 1-*st* column is accurately segmented by all methods. Noticeably, our method obtains higher performance since it has a superior capacity to detect boundaries. Although some object regions are small (2-*nd* column), discontinuous (3-*rd* column), or irregular (4-*th* and 5-*th* columns), DEeR can also more accurately segment them than LoRA and FFA-LoRA. We also visualize the polyp segmentation results of DEeR and previous methods, as shown in Fig. 6. We can find that DEeR can more accurately segment various polyps than LoRA and FFA-LoRA. These qualitative results further illustrate the effectiveness of our DEeR.

D. Ablation Analysis

We perform a comprehensive evaluation on Kvasir-v2 and OCT-8 to investigate the efficacy of different modules in DEeR and the impact of some critical factors, i.e., data heterogeneity β, rank r, communication budget and client number.

1) Evaluation of Different Modules: The deviation eliminator and noise regulator are two indispensable components for DEeR to improve the finetuning performance. To evaluate their contributions, we individually remove them to observe the performance of DEeR. As illustrated in Table IV, DEeR experiences significant performance decline once we remove deviation eliminator (w/o Deviation Eliminator), with decrements of 48.5% from 92.33% to 43.83% on OCT-8 ($\varepsilon = 1.0$) and 14.57% from 87.00% to 72.43% on Kvasir-v2 ($\varepsilon = 6.0$) in accuracy. The decrements are further magnified to 68.39% and 20.63% when the budgets ε on two datasets shrink to 0.1 and 1.0, respectively. Moreover, we can observe that discarding noise regulator (w/o Noise Regulator) leads to a slight performance drop when the privacy budget is high on Kvasir-v2 ($\varepsilon = 6.0$) and OCT-8 ($\varepsilon = 1.0$). Nonetheless, severe performance degradation is triggered by a lower budget, especially on Kvasir-v2. The best results are obtained when DEeR is equipped with deviation eliminator and noise regulator, which can corroborate the effectiveness of the two modules.

2) Impact of Data Heterogeneity: Based on the analysis of Theorem 1, the data heterogeneity can exacerbate aggregation deviation. To investigate its effect, we use the default privacy budget ε and change the heterogeneity parameter β to observe the performance of different methods in Table V. The results show that LoRA and FFA-LoRA undergo more considerable performance drop with decreasing β compared with DEeR. For example, on OCT-8, as β decreases from 10.0 to 0.1, F1-scores of LoRA and FFA-LoRA drop from 70.69% to 38.59% with a decrement of 32.10%, and 86.19% to 52.29% with a decrement of 33.90%, respectively. Noticeably, the decrement of F1-score for DEeR is merely 11.66%. We can find significant difference in DEeR and the second-best FFA-LoRA with P-value < 0.005 for all β. Besides, DEeR achieves better performance than LoRA, FFA-LoRA and DP-DyLoRA on two datasets for different β. The performance advantage can prove that DEeR is more robust against data heterogeneity than existing methods and further confirms the effectiveness of the proposed noise regulator.

3) Impact of Rank r: The rank r can be regarded as LoRA parameter budget. A larger r indicates more trainable

TABLE IV
THE PERFORMANCE OF THE PROPOSED FEDERATED FINETUNING FRAMEWORK WITH DIFFERENT MODULES.

Datasets	Priv. Budget	w/o Deviation Eliminator		w/o Noise Regulator		DEeR	
		Accuracy (%)	F1-score (%)	Accuracy (%)	F1-score (%)	Accuracy (%)	F1-score (%)
OCT-8	$\varepsilon = 1.0$	43.83 ± 5.05	35.18 ± 4.47	86.25 ± 3.77	85.78 ± 4.25	**92.33** ± 0.55	**92.35** ± 0.55
	$\varepsilon = 0.5$	34.44 ± 7.65	25.37 ± 6.80	82.61 ± 5.43	81.53 ± 6.26	**91.20** ± 0.95	**91.21** ± 0.97
	$\varepsilon = 0.1$	15.89 ± 3.95	10.25 ± 4.64	69.44 ± 4.36	67.45 ± 6.14	**84.28** ± 3.74	**83.20** ± 4.59
Kvasir-v2	$\varepsilon = 6.0$	72.43 ± 5.29	68.93 ± 5.31	85.77 ± 0.43	85.46 ± 0.35	**87.00** ± 0.87	**86.84** ± 0.91
	$\varepsilon = 3.0$	72.16 ± 3.03	67.91 ± 2.62	84.81 ± 0.30	84.49 ± 0.48	**86.90** ± 0.75	**86.70** ± 0.85
	$\varepsilon = 1.0$	65.93 ± 1.00	60.98 ± 1.95	78.56 ± 1.68	77.01 ± 2.46	**86.56** ± 0.53	**86.33** ± 0.63

TABLE V
THE PERFORMANCE COMPARISON OF DIFFERENT METHODS UNDER DIFFERENT DATA HETEROGENEITY.

Datasets	Heterogeneity	LoRA		FFA-LoRA		DP-DyLoRA		DEeR	
		Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
OCT-8	$\beta = 10.0$	71.16 ± 1.58	70.69 ± 2.11	86.21 ± 0.25	86.19 ± 0.29	33.94 ± 2.32	29.83 ± 2.16	**94.85** ± 0.08	**94.86** ± 0.08
	$\beta = 1.0$	69.51 ± 0.16	68.48 ± 0.71	83.27 ± 1.65	83.17 ± 1.62	32.86 ± 1.85	28.25 ± 2.27	**94.82** ± 0.28	**94.81** ± 0.28
	$\beta = 0.5$	66.67 ± 1.76	66.41 ± 1.80	81.29 ± 2.17	81.22 ± 2.11	32.76 ± 2.30	30.28 ± 3.00	**94.16** ± 0.28	**94.16** ± 0.28
	$\beta = 0.1$	42.15 ± 2.88	38.59 ± 3.60	56.20 ± 4.63	52.29 ± 6.10	23.55 ± 4.61	19.30 ± 3.40	**84.28** ± 3.74	**83.20** ± 4.59
Kvasir-v2	$\beta = 10.0$	91.00 ± 0.33	90.99 ± 0.34	86.54 ± 1.18	86.52 ± 1.17	88.54 ± 0.51	88.50 ± 0.54	**91.48** ± 0.52	**91.47** ± 0.51
	$\beta = 1.0$	89.98 ± 0.46	89.95 ± 0.46	86.06 ± 0.44	86.04 ± 0.44	88.29 ± 0.26	88.29 ± 0.23	**90.38** ± 0.20	**90.37** ± 0.21
	$\beta = 0.5$	89.00 ± 0.67	88.94 ± 0.70	84.54 ± 0.51	84.45 ± 0.54	86.60 ± 2.02	86.35 ± 2.27	**90.27** ± 0.46	**90.25** ± 0.49
	$\beta = 0.1$	84.85 ± 0.86	84.51 ± 0.86	79.21 ± 0.62	77.94 ± 0.80	76.93 ± 1.40	75.17 ± 2.68	**86.90** ± 0.75	**86.70** ± 0.85

TABLE VI
THE PERFORMANCE COMPARISON OF DIFFERENT METHODS WITH DIFFERENT RANKS OF LORA.

Datasets	LoRA Rank	LoRA		FFA-LoRA		DP-DyLoRA		DEeR	
		Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
OCT-8	$r = 16$	46.77 ± 5.07	41.53 ± 5.74	56.85 ± 4.21	54.70 ± 5.70	30.28 ± 4.00	22.81 ± 2.13	**87.50** ± 1.99	**87.31** ± 2.16
	$r = 8$	42.15 ± 2.88	38.59 ± 3.60	56.20 ± 4.63	52.29 ± 6.10	23.55 ± 4.61	19.30 ± 3.40	**84.28** ± 3.74	**83.20** ± 4.59
	$r = 4$	35.34 ± 4.38	29.33 ± 2.48	66.32 ± 6.56	64.61 ± 8.29	21.54 ± 5.43	16.38 ± 6.33	**78.09** ± 3.26	**76.21** ± 4.21
	$r = 2$	23.07 ± 1.81	17.28 ± 0.18	**63.55** ± 4.84	**60.25** ± 5.32	19.95 ± 1.84	12.80 ± 2.11	61.28 ± 3.69	57.11 ± 5.35
Kvasir-v2	$r = 16$	83.39 ± 1.04	82.81 ± 1.16	76.75 ± 0.68	75.50 ± 0.95	81.97 ± 1.28	81.15 ± 1.34	**86.81** ± 0.75	**86.67** ± 0.78
	$r = 8$	84.85 ± 0.86	84.51 ± 0.86	79.21 ± 0.62	77.94 ± 0.80	76.93 ± 1.40	75.17 ± 2.68	**86.90** ± 0.75	**86.70** ± 0.85
	$r = 4$	84.43 ± 0.40	84.14 ± 0.36	79.56 ± 0.38	78.40 ± 0.13	75.10 ± 4.65	71.37 ± 7.37	**87.42** ± 0.38	**87.25** ± 0.45
	$r = 2$	79.35 ± 0.77	77.97 ± 1.38	79.68 ± 0.70	78.58 ± 0.70	72.16 ± 2.27	69.26 ± 3.44	**84.83** ± 1.15	**84.61** ± 1.27

TABLE VII
THE PERFORMANCE COMPARISON OF DIFFERENT METHODS WITH DIFFERENT CLIENT NUMBER.

Datasets	LoRA Rank	LoRA		FFA-LoRA		DP-DyLoRA		DEeR	
		Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
OCT-8	$K = 16$	52.71 ± 3.13	49.48 ± 3.39	64.01 ± 2.51	62.52 ± 2.65	32.60 ± 4.72	27.39 ± 5.72	**88.57** ± 1.34	**88.34** ± 1.46
	$K = 12$	42.15 ± 2.88	38.59 ± 3.60	56.20 ± 4.63	52.29 ± 6.10	23.55 ± 4.61	19.30 ± 3.40	**84.28** ± 3.74	**83.20** ± 4.59
	$K = 8$	30.57 ± 3.84	21.94 ± 3.35	54.42 ± 2.17	49.91 ± 2.04	22.08 ± 3.96	16.58 ± 2.51	**88.13** ± 0.79	**87.89** ± 0.82
	$K = 4$	25.91 ± 1.03	19.77 ± 2.31	42.16 ± 6.02	34.99 ± 7.88	22.30 ± 1.21	15.32 ± 2.67	**76.39** ± 4.74	**73.39** ± 6.41
Kvasir-v2	$K = 16$	85.58 ± 0.26	85.46 ± 0.34	82.27 ± 1.88	82.15 ± 1.89	84.75 ± 1.45	84.50 ± 1.58	**87.70** ± 0.69	**87.61** ± 0.66
	$K = 12$	84.85 ± 0.86	84.51 ± 0.86	79.21 ± 0.62	77.94 ± 0.80	76.93 ± 1.40	75.17 ± 2.68	**86.90** ± 0.75	**86.70** ± 0.85
	$K = 8$	81.75 ± 1.48	80.81 ± 2.02	79.64 ± 1.62	78.83 ± 2.42	75.81 ± 0.40	73.67 ± 1.20	**86.95** ± 0.83	**86.79** ± 0.91
	$K = 4$	69.72 ± 1.44	64.71 ± 1.13	68.66 ± 1.45	64.37 ± 1.61	59.66 ± 2.90	52.83 ± 4.28	**78.47** ± 2.19	**75.16** ± 2.93

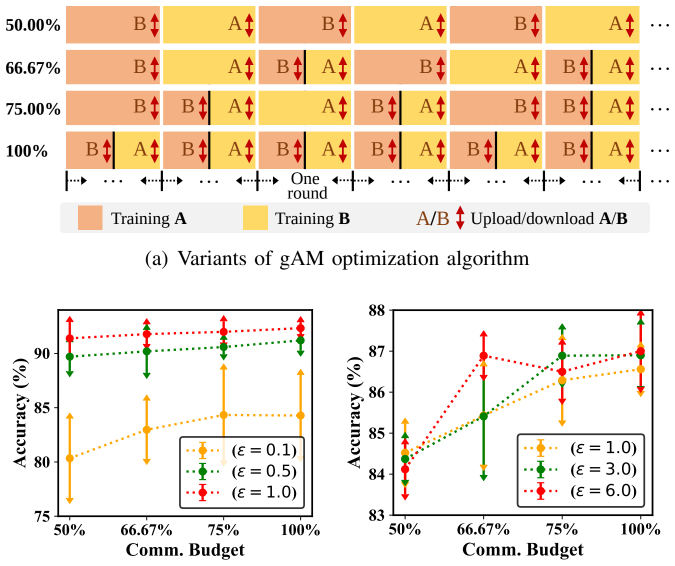
parameters. We fix the privacy budget ε as well as the heterogeneity parameter β, and compare the performance of different methods with various $r \in [2, 4, 8, 16]$. As presented in Table VI, FFA-LoRA achieves the best performance with 63.55% in accuracy and 60.25% in F1-score on OCT-8 when $r = 2$. As r increases from 2 to 16, DEeR exhibits promising performance improvement to 87.50% and 87.31% with increments of 26.22% and 30.20% in accuracy and F1-score respectively, while accuracy and F1-score of FFA-LoRA decrease to 56.85% and 54.70% (P-value < 0.005 for $r = 8$ and P-value < 0.002 for $r = 16$), and DP-DyLoRA always present extremely poor performance. For Kvasir-v2, all methods do not show huge performance fluctuations as r changes. It is worth mentioning that DEeR exceeds LoRA and FFA-LoRA by a huge margin for different r. Particularly, DEeR with the fewest parameters ($r = 2$) can obtain better performance than the second-best method (LoRA) with the most parameters ($r = 16$). Significant difference is presented in DEeR and the second-best LoRA for $r = 2$ (P-value < 0.01) and $r = 4$ (P-value < 0.005). DP-DyLoRA shows not only bigger performance fluctuations as r changes but also lower performance than DEeR.

4) Impact of Communication Budget: In DEeR, different variants of gAM optimization algorithm lead to different communication budgets, as shown in Fig. 7(a). We assume that the communication budget is 100% when one round of training includes both **A** and **B**, while it is 50% if one round only contains **A** or **B**. Fig. 7(b)-(c) present performance of different variants on OCT-8 and Kvasir-v2, respectively. In Fig. 7(b), even though we reduce the communication budget from 100% to 50%, model still maintains a good and stable performance for big privacy budgets ($\varepsilon = 0.5$ or 1.0) on OCT-8 dataset. Fig. 7(c) demonstrates that 75% of the communication budget can achieve similar performance as 100% of the budget for all privacy budgets on Kvasir-v2 dataset. Moreover, the best performance from 100% of the communication budget indicates the importance of optimizing both **A** and **B** for each round.

5) Impact of Client Number: To compare the performance of different methods across different numbers of clients, we fix the privacy budgets ε, the data heterogeneity β and the rank r of LoRA layers, and divide the training data of OCT-8 and Kvasir-v2 datasets into K clients, respectively. As shown in Table VII, for OCT-8 dataset, all existing methods yield limited classification performance for various client numbers. DEeR exceeds the second-best method FFA-LoRA by large margins, such as 34.23% (P-value < 0.002) and 24.56% (P-value < 0.05) in Accuracy for $K = 4$ and 16, respectively. Meanwhile, DEeR also significantly outperforms these methods for any K on Kvasir-v2 dataset. For both two datasets, DEeR shows a lower sensitivity to the client number K than previous methods. The performance gap of these methods between K=4 and 16 surpasses 20% in terms of Accuracy and F1-score, while the gap of DEeR is around 10%. Therefore, these results illustrate that DEeR is more robust against the number of clients than existing approaches.

VI. Conclusion

In this paper, we propose a novel FedFT framework named DEeR, which exploits LoRA to adapt pretrained foundation models to downstream medical tasks in FL with client-level DP guarantees. We first delve into two challenges of FedFT with LoRA, *i.e.*, aggregation deviation and noise amplification effect. Afterwards, a deviation eliminator is proposed to utilize the alternating minimization optimization algorithm to iteratively optimize the parameters of LoRA for avoiding aggregation deviation. Besides, we present a noise regulator at the client side that introduces two regulator factors to suppress the noise amplification effect. The comprehensive experiments on two classification and two segmentation datasets validate the effectiveness of DEeR. The results show DEeR achieves superior performance than state-of-the-art methods. The ablated experiments verify the importance of key modules in DEeR, investigate the impact of data heterogeneity, rank r, the communication budget and client number.



(a) Variants of gAM optimization algorithm

(b) OCT-8

(c) Kvasir-v2

Fig. 7. The impact of communication budgets in different variants of gAM optimization algorithm.

The proposed DEeR has achieved promising performance on various medical tasks, yet there are several limitations: (1) In DEeR, gAM algorithm is exploited to optimize the parameters of LoRA for avoiding aggregation deviation. However, the alternating optimization strategy will increase the training time. Although we have explored different variants of gAM algorithm to reduce communication frequency, they fail to achieve the same performance as 100% of the communication frequency for different privacy budgets. (2) DEeR may undergo data security risk during the communication process. Although our method does not share the raw data of clients and protects client-level privacy by DP, local client models might be stolen by intruders and competitors for the reconstruction of training data [23]. For this problem, we are able to apply existing homomorphic encryption techniques [23] to encrypt client models and the global model.

References

[1] M. Q.-H. Meng, "Bridging ai to robotics via biomimetics," *Biomimetic Intelligence and Robotics*, vol. 1, p. 100006, 2021.

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021, pp. 8748–8763.

[3] S. Zhang, Y. Xu, N. Usuyama, J. Bagga, R. Tinn, S. Preston *et al.*, "Large-scale domain-specific pretraining for biomedical vision-language processing," *arXiv preprint arXiv:2303.00915*, vol. 2, no. 3, p. 6, 2023.

[4] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *ICCV*, 2023, pp. 4015–4026.

[5] J. Qiu, L. Li, J. Sun, J. Peng, P. Shi, R. Zhang *et al.*, "Large ai models in health informatics: Applications, challenges, and the future," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 12, pp. 6074–6087, 2023.

[6] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, "Large language models in medicine," *Nature medicine*, vol. 29, no. 8, pp. 1930–1940, 2023.

[7] Y. Lin, L. Tan, H. Lin, Z. Zheng, R. Pi, J. Zhang, S. Diao, H. Wang, H. Zhao, Y. Yao *et al.*, "Speciality vs generality: An empirical study on catastrophic forgetting in fine-tuning foundation models," *arXiv preprint arXiv:2309.06256*, 2023.

[8] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," *Nature Communications*, vol. 15, no. 1, p. 654, 2024.

[9] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang *et al.*, "Lora: Low-rank adaptation of large language models," in *ICLR*, 2022.

[10] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *ECCV*. Springer, 2022, pp. 709–727.

[11] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *ACL*, 2021, pp. 4582–4597.

[12] D. P. Nguyen, J. P. Munoz, and A. Jannesari, "Flora: Enhancing vision-language models with parameter-efficient federated learning," *arXiv preprint arXiv:2404.15182*, 2024.

[13] Y. Sun, Z. Li, Y. Li, and B. Ding, "Improving lora in privacy-preserving federated learning," *ICLR*, 2024.

[14] S. Babakniya, A. R. Elkordy, Y. H. Ezzeldin, Q. Liu, K.-B. Song, M. EL-Khamy, and S. Avestimehr, "Slora: Federated parameter efficient fine-tuning of language models," in *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*, 2023.

[15] M. Zhu, J. Liao, J. Liu, and Y. Yuan, "Fedoss: Federated open set recognition via inter-client discrepancy and collaboration," *IEEE Trans. Med. Imaging.*, vol. 43, no. 1, pp. 190–202, 2024.

[16] M. Zhu, Z. Chen, and Y. Yuan, "Feddm: Federated weakly supervised segmentation via annotation calibration and gradient de-conflicting," *IEEE Trans. Med. Imaging.*, vol. 42, no. 6, pp. 1632–1643, 2023.

[17] Z. Chen, M. Zhu, C. Yang, and Y. Yuan, "Personalized retrogress-resilient framework for real-world medical federated learning," in *MICCAI*. Springer, 2021, pp. 347–356.

[18] Z. Chen, C. Yang, M. Zhu, Z. Peng, and Y. Yuan, "Personalized retrogress-resilient federated learning toward imbalanced medical data," *IEEE Trans. Med. Imaging.*, vol. 41, no. 12, pp. 3663–3674, 2022.

[19] C. Yang, M. Zhu, Y. Liu, and Y. Yuan, "Fedpd: Federated open set recognition with parameter disentanglement," in *ICCV*, October 2023, pp. 4882–4891.

[20] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*. PMLR, 2017, pp. 1273–1282.

[21] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi *et al.*, "Federated optimization in heterogeneous networks," *MLSys*, vol. 2, pp. 429–450, 2020.

[22] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*, 2006, pp. 265–284.

[23] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[24] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in *USENIX ATC*, 2020, pp. 493–506.

[25] X. Wu, X. Liu, J. Niu, H. Wang, S. Tang, and G. Zhu, "FedloRA: When personalized federated learning meets low-rank adaptation," 2024. [Online]. Available: https://openreview.net/forum?id=bZh06ptG9r

[26] Z. Zhang, Y. Yang, Y. Dai, Q. Wang, Y. Yu, L. Qu, and Z. Xu, "Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models," in *ACL*, 2023, pp. 9963–9977.

[27] Y. Du, Z. Zhang, L. Yue, X. Huang, Y. Zhang, T. Xu, L. Xu, and E. Chen, "Communication-efficient personalized federated learning for speech-to-text tasks," *arXiv preprint arXiv:2401.10070*, 2024.

[28] Y. J. Cho, L. Liu, Z. Xu, A. Fahrezi, M. Barnes, and G. Joshi, "Heterogeneous lora for federated fine-tuning of on-device foundation models," in *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*, 2023.

[29] J. Jiang, X. Liu, and C. Fan, "Low-parameter federated learning with large language models," *arXiv preprint arXiv:2307.13896*, 2023.

[30] X. Yang, W. Huang, and M. Ye, "Dynamic personalized federated learning with adaptive differential privacy," *NeurIPS*, vol. 36, pp. 72 181–72 192, 2023.

[31] Y.-L. Sung, J. Cho, and M. Bansal, "Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks," in *CVPR*, 2022, pp. 5227–5237.

[32] M. Xu, Z. Zhang, F. Wei, H. Hu, and X. Bai, "Side adapter network for open-vocabulary semantic segmentation," in *CVPR*, 2023, pp. 2945–2954.

[33] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *Int. J. Comput. Vis.*, vol. 130, no. 9, pp. 2337–2348, 2022.

[34] Z. Han, C. Gao, J. Liu, S. Q. Zhang *et al.*, "Parameter-efficient fine-tuning for large models: A comprehensive survey," *arXiv preprint arXiv:2403.14608*, 2024.

[35] Y. Lin, X. Ma, X. Chu, Y. Jin, Z. Yang, Y. Wang, and H. Mei, "Lora dropout as a sparsity regularizer for overfitting control," *arXiv preprint arXiv:2404.09610*, 2024.

[36] Q. Zhang, M. Chen, A. Bukharin, P. He, Y. Cheng, W. Chen, and T. Zhao, "Adaptive budget allocation for parameter-efficient fine-tuning," in *ICLR*, 2023.

[37] Z. Zhong, Z. Tang, T. He *et al.*, "Convolution meets loRA: Parameter efficient finetuning for segment anything model," in *ICLR*, 2024.

[38] X. Wu, S. Huang, and F. Wei, "Mixture of loRA experts," in *ICLR*, 2024.

[39] Q. Liu, X. Wu, X. Zhao, Y. Zhu, D. Xu, F. Tian, and Y. Zheng, "Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications," *arXiv preprint arXiv:2310.18339*, 2023.

[40] Y. Yan, S. Tang, Z. Shi, and Q. Yang, "Federa: Efficient fine-tuning of language models in federated learning leveraging weight decomposition," *arXiv preprint arXiv:2404.18848*, 2024.

[41] Y. Yang, X. Liu, T. Gao, X. Xu, and G. Wang, "Sa-fedlora: Adaptive parameter allocation for efficient federated learning with lora tuning," *arXiv preprint arXiv:2405.09394*, 2024.

[42] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010, pp. 249–256.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015, pp. 1026–1034.

[44] K. Kuo, A. Raje, K. Rajesh, and V. Smith, "Federated lora with sparse communication," *arXiv preprint arXiv:2406.05233*, 2024.

[45] Y. Shi, Y. Liu, K. Wei, L. Shen, X. Wang, and D. Tao, "Make landscape flatter in differentially private federated learning," in *CVPR*, 2023, pp. 24 552–24 562.

[46] K. Pogorelov, K. R. Randel, C. Griwodz, S. L. Eskeland *et al.*, "Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection," in *MMSYS*, 2017, pp. 164–169.

[47] P. Jain, P. Kar *et al.*, "Non-convex optimization for machine learning," *Foundations and Trends® in Machine Learning*, vol. 10, no. 3-4, pp. 142–363, 2017.

[48] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[49] K. Lange and K. Lange, "Singular value decomposition," *Numerical analysis for statisticians*, pp. 129–142, 2010.

[50] G. Peters and J. H. Wilkinson, "The least squares problem and pseudo-inverses," *The Computer Journal*, vol. 13, no. 3, pp. 309–316, 1970.

[51] M. Subramanian, K. Shanmugavadivel, O. S. Naren, K. Premkumar, and K. Rankish, "Classification of retinal oct images using deep learning," in *ICCCI*. IEEE, 2022, pp. 1–7.

[52] V. M. Campello, P. Gkontra, C. Izquierdo, C. Martin-Isla, A. Sojoudi, P. M. Full, K. Maier-Hein, Y. Zhang *et al.*, "Multi-centre, multi-vendor and multi-disease cardiac segmentation: the m&ms challenge," *IEEE Trans. Med. Imaging.*, vol. 40, no. 12, pp. 3543–3554, 2021.

[53] D.-P. Fan, G.-P. Ji, T. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao, "Pranet: Parallel reverse attention network for polyp segmentation," in *MICCAI*, 2020, pp. 263–273.

[54] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodríguez, and F. Vilariño, "Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians," *Comput. Med. Imaging Graph.*, vol. 43, pp. 99–111, 2015.

[55] J. Bernal, J. Sánchez, and F. Vilarino, "Towards automatic polyp detection with a polyp appearance model," *Pattern Recognit.*, vol. 45, no. 9, pp. 3166–3182, 2012.

[56] J. Silva, A. Histace, O. Romain, X. Dray, and B. Granado, "Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer," *IJCARS*, vol. 9, no. 2, pp. 283–293, 2014.

[57] D. Jha, P. H. Smedsrud, M. A. Riegler, P. Halvorsen, T. d. Lange, D. Johansen, and H. D. Johansen, "Kvasir-seg: A segmented polyp dataset," in *MMM*. Springer, 2020, pp. 451–462.

[58] J. Cheng, J. Ye, Z. Deng, J. Chen, T. Li, H. Wang, Y. Su, Z. Huang *et al.*, "Sam-med2d," *arXiv preprint arXiv:2308.16184*, 2023.

[59] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine *et al.*, "Opacus: User-friendly differential privacy library in PyTorch," *arXiv preprint arXiv:2109.12298*, 2021.

[60] J. Xu, K. Saravanan, R. van Dalen, H. Mehmood, D. Tuckey, and M. Ozay, "Dp-dylora: Fine-tuning transformer-based models on-device

under differentially private federated learning using dynamic low-rank adaptation," *arXiv preprint arXiv:2405.06368*, 2024.