



## WP9 Scalable federated learning using pre-trained models

---

### D9.2 Report and documentation of new federated fine-tuning and distillation algorithms

<https://www.fluteproject.eu/>



Funded by  
the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101095382. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the HaDEA. Neither the European Union nor the granting authority can be held responsible for them.

**Grant Agreement No.: 101095382**  
**Deliverable: D9.2 Report and documentation of new federated fine-tuning and distillation algorithms**

**Project Start Date:** 01/05/2023  
**Coordinator:** INRIA

**Duration:** 36 months

<b>Deliverable No:</b>	D9.2
<b>WP No:</b>	9
<b>WP Leader:</b>	Siemens
<b>Due date:</b>	30/04/2025
<b>Delivery date:</b>	30/04/2025

**Dissemination Level:**

PU	Public Use	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

## DOCUMENT SUMMARY INFORMATION

<b>Project title:</b>	<b>Federated Learning and mUlti-party computation Techniques for prostatE cancer</b>
<b>Short project name:</b>	FLUTE
<b>Project No:</b>	101095382
<b>Call Identifier:</b>	HORIZON-HLTH-2022-IND-13
<b>Thematic Priority:</b>	HORIZON-HLTH-2022-IND-13
<b>Type of Action:</b>	HORIZON Research and Innovation Actions
<b>Start date of the project:</b>	01/05/2023
<b>Duration of the project:</b>	36 months
<b>Project website:</b>	<a href="https://www.fluteproject.eu/">https://www.fluteproject.eu/</a>

### D9.2 Report and documentation of new federated fine-tuning and distillation algorithms

<b>Work Package:</b>	WP9 Scalable federated learning using pre-trained models
<b>Deliverable number:</b>	D9.2
<b>Deliverable title:</b>	Report and documentation of new federated fine-tuning and distillation algorithms
<b>Due date:</b>	30/04/2025
<b>Actual submission date:</b>	30/04/2025
<b>Authors:</b>	Alexandru Serban, Cosmin Hatfaludi
<b>Dissemination Level:</b>	PU
<b>No. pages:</b>	30
<b>Authorized (date):</b>	29/04/2025
<b>Responsible person:</b>	Jan Ramon
<b>Status:</b>	Final

#### Revision history:

Version	Date	Author	Comment
v.0.1	1/04/2025	Alexandru Serban, Cosmin Hatfaludi	First draft including the classification and description of the methods
v.1.0	22/04/2025	Carlota Cañamero Herrero, Jan Ramon	First internally reviewed version
v.1.1	25/04/2025	Alexandru Serban, Cosmin Hatfaludi	Final version after suggested changes
v.2.0	29/04/2025	Alexandru Serban, Cosmin Hatfaludi	Final version after reviewed changes

#### Quality Control:

	<b>Who</b>	<b>Date</b>
<b>Checked by internal reviewer</b>	Carlota Cañamero Herrero, GRAD	22/04/2025
<b>Checked by internal reviewers</b>	Jan Ramon, INRIA	22/04/2025
<b>Checked by internal reviewer</b>	Alexandru Serban, SIEMENS	28/04/2025
<b>Checked by internal reviewers</b>	Jan Ramon, INRIA	29/04/2025
<b>Checked by WP Leader</b>	Alexandru Serban, SIEMENS	29/04/2025
<b>Checked by Project Technical Managers</b>	Jan Ramon, INRIA	29/04/2025
<b>Checked by Project Coordinator</b>	Jan Ramon, INRIA	29/04/2025

## COPYRIGHT

©Copyright by the FLUTE consortium, 2023-2026.

This document contains material, which is the copyright of FLUTE consortium members and the European Commission, and may not be reproduced or copied without permission, except as mandated by the European Commission Grant Agreement no. 101095382 for reviewing and dissemination purposes.

## ACKNOWLEDGEMENTS

FLUTE is a project that has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101095382. Please see project URL <https://www.fluteproject.eu/> for more information.

The partners in the project are . The content of this document is the result of the worked developed by the partners in the context of the project.

## DISCLAIMER

The content of the publication herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services. The information contained in this document is provided by the copyright holders "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the members of the FLUTE collaboration, including the copyright holders, or the European Commission be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of the information contained in this document, even if advised of the possibility of such damage.

# Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
<b>2</b>	<b>Background</b>	<b>10</b>
<b>3</b>	<b>Dataset</b>	<b>13</b>
3.1	Preprocessing and Normalization . . . . .	14
<b>4</b>	<b>Methods</b>	<b>15</b>
4.1	Pre-Training Foundational Models . . . . .	16
4.2	Prostate Cancer Classification . . . . .	17
4.3	Lesion Based Cancer Classification . . . . .	17
4.4	Federated Fine-Tuning . . . . .	18
4.5	Federated Additive Fine-Tuning . . . . .	19
4.6	Hybrid Federated Additive Fine-tuning and Quantization . . . . .	19
4.7	Federated Distillation . . . . .	19
<b>5</b>	<b>Results</b>	<b>20</b>
5.1	Supervised Learning using Foundational Models . . . . .	20
5.2	Federated Fine-tuning . . . . .	21
5.3	Federated Additive Fine-tuning . . . . .	21
5.3.1	Exploring Distinct Collection Grouping . . . . .	23
5.4	Hybrid Federated Additive Fine-tuning and Quantization . . . . .	24
5.5	Distillation . . . . .	24
<b>6</b>	<b>Discussion</b>	<b>25</b>
<b>7</b>	<b>Conclusions</b>	<b>26</b>
<b>A</b>	<b>Appendix</b>	<b>27</b>
A.1	Appendix . . . . .	27

---

## List of Acronyms

**ML** machine learning

**LoRA** low-rank adaptation

**FL** Federated learning

**FM** foundational model

**FedAvg** Federated Average

**FedAdam** Federated Adam

**MIM** Masked Image Modelling

**ViT** Vision Transformer

**SWIN** Shifted Window Transformer

---

## Summary

This report details the development and implementation of federated fine-tuning and distillation algorithms for foundational models, specifically for detecting prostate cancer using image-based data. Following recommendations from D9.1, we investigate additive model fine-tuning and model compression methods, including quantization and distillation, as optimal strategies for integrating privacy-enhancing technologies within FLUTE. Experiments conducted on a custom dataset from Siemens, using both organ-level and task-specific foundational models, yield key insights. Federated Adam, used as an aggregator, demonstrated superior performance for pre-trained models, highlighting the importance of diverse aggregation techniques. Federated additive fine-tuning with adapters in the final stage and Low-Rank Adaptation showed competitive performance, suggesting their potential for integration into FLUTE and offering flexibility based on federated learning communication constraints. Hybrid approaches, such as quantizing LoRA parameters, significantly improved communication efficiency with minimal performance impact. Additionally, federated additive fine-tuning outperformed full-model fine-tuning by preserving pre-trained embeddings, underscoring its effectiveness in scenarios where the data is heterogeneous.

# 1 Introduction

WP9 extends the FLUTE platform by integrating Federated learning (FL) algorithms that leverage unlabeled data through techniques such as pre-training, fine-tuning, and more general use of foundational models (FMs) in FL. This extension aims to improve the platform's scalability, allowing it to manage larger data volumes and model sizes more efficiently. By doing so, WP9 seeks to narrow the cost gap between new privacy-preserving machine learning algorithms and more traditional methods, which often overlook the trade-offs associated with cost and scalability.

To evaluate the algorithms developed in WP9, we build upon the existing work in FLUTE for prostate cancer detection. Initially centered on human-engineered features (WP2 and WP5), WP9 transitions to an approach that relies only on image-based data and aims to leverage unlabeled data to overcome potential limitations and improve performance. This involves using pre-trained FMs models on unlabeled data, followed by fine-tuning these models with FLUTE data.

This report builds upon D9.1, which reviewed and classified algorithms for training and using FMs in FL. It presents the results and insights derived from developing federated fine-tuning and distillation algorithms for FMs, specifically applied to the prostate cancer use-case within the FLUTE project.

In summary, D9.1 recommended that for the collaborative training of FMs using FL, partial model pre-training with additive methods as an optimal choice for future integration with Privacy-Enhancing Technologies (PET), as will be done in FLUTE [2]. These methods offer lower complexity and higher efficiency, making them more adaptable to the increased overhead from technologies like encryption for shared parameters. In contrast, methods for entire model pre-training remain challenging to implement, even without PET.

Furthermore, customizing FMs using FL was recommended as the most compelling method for practical applications, such as the FLUTE case study. Fine-tuning, hybrid fine-tuning, and contraction methods such as distillation, compression or quantization offer significant benefits in terms of efficiency and scalability. These methods also facilitate the integration of PET technologies. From the fine-tuning methods, adaptive methods through low-rank adaptation (LoRA) or final stage tuning were found to be the most suitable, as besides scalability and efficiency, they enable a suite of customization options such as client-based resource balancing or personalization [4, 22]. Among the contraction methods explored, distillation and quantization were identified as providing the most significant benefits in reducing communication costs.

Given these initial conclusions, we prioritize the development of additive fine-tuning methods using adapters in the final stage and LoRA, as well as compression methods through quantization and distillation. We present these experiences together with trade-offs associated with their implementation. All experiments were performed on a custom dataset available at Siemens, retrieved from 17 internal collections distributed across 3 geographical regions and consisting of approximately 1393 patients for which a biopsy was performed.

The use-case involves classifying prostate cancer patients using only image-based data. We explore two paths for tackling this use-case: one that classifies prostate cancer using organ-

level imaging, which includes image sequences of the entire prostate, and another that uses lesion-based imaging, which focuses on cropped image sequences around each lesion. Overall, the lesion-based approach yields better performance, scalability, and efficiency in FL. This is because the algorithm processes more targeted data, reducing input size and model parameters, thereby enhancing scalability and efficiency, especially in FL settings.

We also find that federated fine-tuning of FMs outperforms fine-tuning the entire models in FL. When combined with compression mechanisms such as quantization, it can reduce communication costs by up to 99%. Although there remains a slight performance gap when running the algorithms in FL compared to centralized fine-tuning, the gap is not significant and can likely be mitigated using more complex models or hyper-parameter tuning.

The remainder of this document is organised as follows. We first discuss background information for the methods used in this report, including FMs and FL (Section 2) followed by a detailed description of the dataset used for experimentation (Section 3). Next, we discuss experimental settings (Section 4) and results (Section 5). We end with a discussion about the findings in the context of FLUTE (Section 6) and conclusions (Section 7).

## 2 Background

FMs are pre-trained on unlabeled data by creating pretext tasks, thereby eliminating the need for annotated data and significantly expanding the datasets available for learning. These models have demonstrated improved performance and robustness across a wide range of image-based machine learning (ML) tasks, reducing the reliance on large, labeled datasets for each specific task [1, 10, 24]. This reduction lowers the costs and efforts associated with data annotation and model training. As FMs continue to demonstrate superior performance and versatility, they are increasingly becoming standard in ML engineering, where starting from an already available pre-trained model is a common practice.

In medical image analysis, the spectrum of FM can be categorized based on the data used for pre-training [23]. This ranges from more general medical FMs, which are trained on multiple modalities and organs, to more specific foundational models. The latter are trained on modality-specific data (e.g., MRIs with multiple organs), organ-specific data (e.g., prostate MRIs), and even task-specific data (e.g., lesion-based Federated Models). An illustration of this classification is provided in Figure 1.

Given that unlabeled data is available for pre-training, more specific FMs such as organ or task-specific models are expected to perform better for a particular task [23], as is the case in FLUTE. This is because these models are tailored to the specific characteristics and requirements of the data that will be also used in the downstream task.

Furthermore, these models are typically smaller in terms of the number of parameters, as they have to learn less data variations and nuances compared to more general models. This reduction in complexity is beneficial for FL settings, where communication costs must be optimized.

Therefore, in this report we experiment with organ and task-specific FMs.

To customize FMs for the FLUTE use-case, we use two classes of algorithms detailed in D9.1:

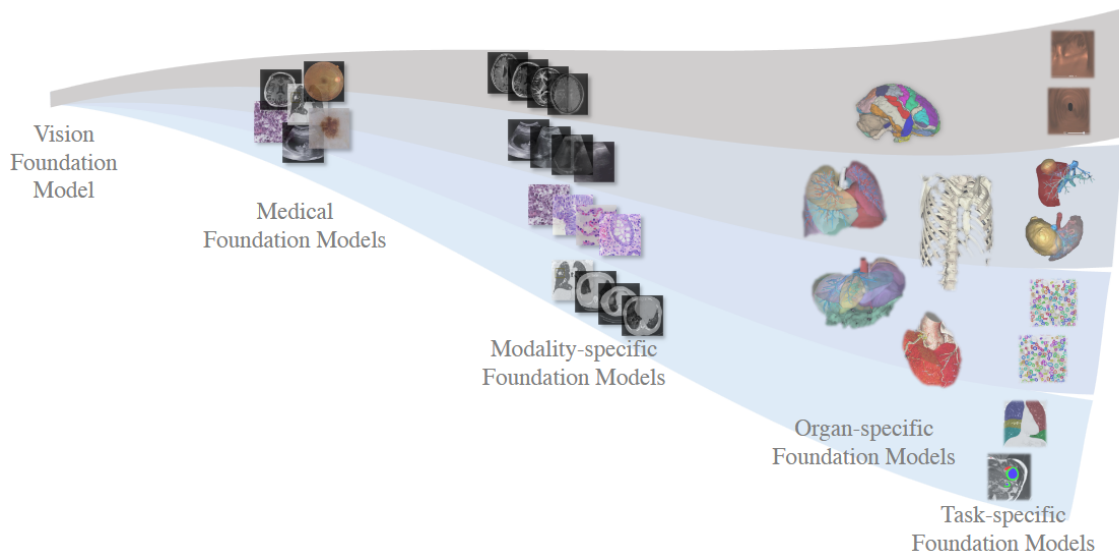
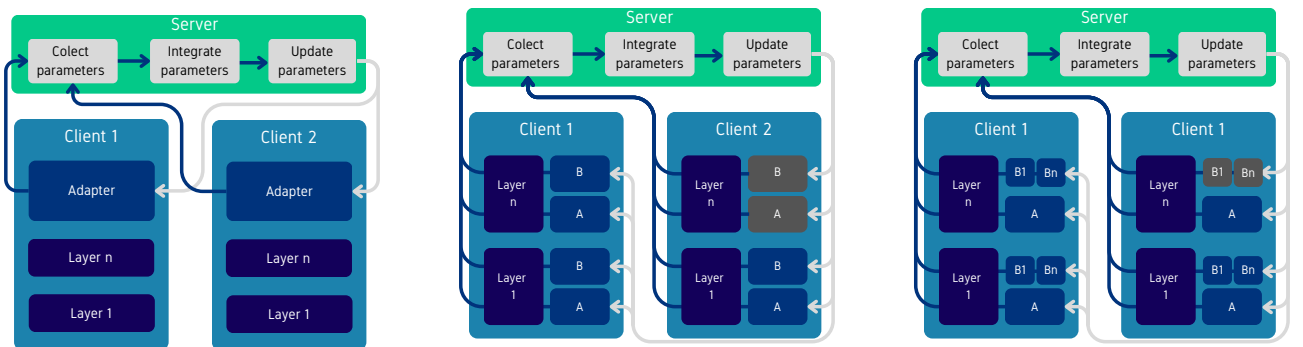


Figure 1: The spectrum of foundational models for medical images, as classified by [23].

additive fine-tuning and knowledge distillation. An overview of additive fine-tuning methods is illustrated in Figure 2. Specifically, we explore additive fine-tuning with adapters in the final stage (Figure 2a) and additive fine-tuning using layer-based LoRA adapters (Figure 2b). These methods are further enhanced with quantization techniques to create hybrid approaches.



(a) Additive fine-tuning with adapters in the final stage.

(b) Additive fine-tuning using LoRA adapters.

(c) Additive fine-tuning using asymmetric LoRA.

Figure 2: Overview of additive fine-tuning methods, where the grey boxes mean that the added parameters can be heterogeneous between clients.

For the first method, at the start of training, the FM and the newly initialized last layer adapter is distributed to all clients. The FM remains frozen throughout the training process, and only the final layer is involved in FL. During each training epoch, each client performs a forward pass through its data and shares the parameters from the final layer with the server. The server aggregates these parameters and sends them back to the clients, repeating this process until convergence.

In the second method, at the beginning of training, the FM and a set of low-rank adapters, chosen for each layer of the FM, are distributed to all clients. During each training epoch,

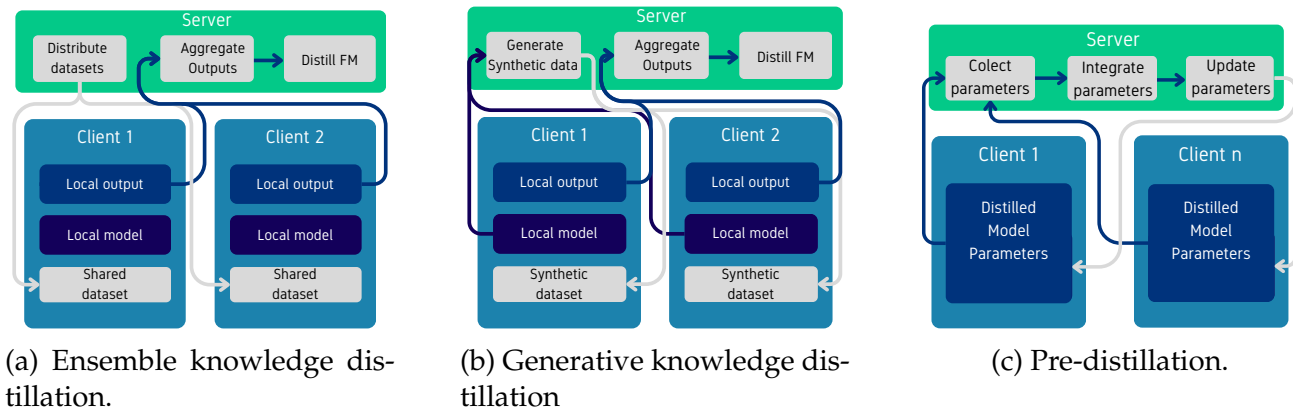


Figure 3: Overview of knowledge distillation methods.

each client performs a forward pass through its data and shares the parameters of the **LoRA** adapters with the server, which then aggregates the results. While this method increases the number of parameters, it allows for more adaptive and nuanced fine-tuning of the model.

The low-rank adapters are defined by two trainable matrices  $Wx = BAx$ , where  $A$  encodes the input to a lower dimensional rank and  $B$  recovers the output dimension of the original  $W_0$ . During the forward pass, each model layer processes the input through the frozen original weights  $W_0x$  and the low-rank adapters. The outputs from these paths are summed and passed to the next layer as  $W_0x + \Delta Wx$ . The  $\Delta$  parameter serves as a constant scale coefficient.

In the context of transformer-based architectures, which are used in the experiments presented in the report, **LoRA** is applied specifically to the attention weight matrices, while the MLP modules remain frozen.

An overview of the knowledge distillation methods identified in D9.1 is presented in Figure 3. We observe that for the first two types of distillation methods – ensemble distillation (Figure 3a) and generative distillation (Figure 3b) – a shared dataset, real or synthetically generated, is required. These classes of algorithms rely on supervised learning for the distillation process.

In contrast, the third class of methods, known as pre-distillation, does not require a shared dataset. This process occurs outside the **FL** network and can be applied to either supervised or unsupervised datasets. Given that a shared dataset is not envisioned for **FLUTE**, this report focuses on experimenting with unsupervised pre-distillation.

In this approach, a larger model, referred to as the Teacher, transfers its knowledge to a smaller model, known as the Student. The Student model is designed to operate efficiently on limited infrastructure, such as within the **FL** network. The primary objective is to transfer the knowledge from a model that is too large to be practical into a smaller, more versatile model that can function effectively within resource constraints.

For unsupervised knowledge transfer, both the teacher and student models perform inference on two augmented versions of the same input sample. The difference between the embeddings produced by the two models is then minimized, thereby aligning their representations.

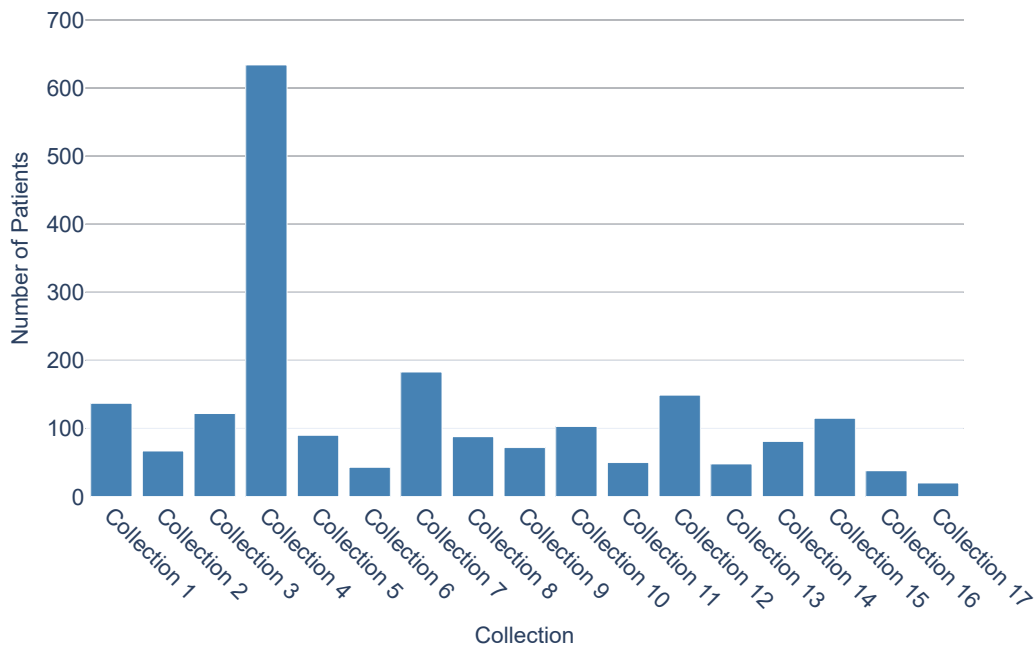


Figure 4: Distribution of patients across vendor collections.

### 3 Dataset

The experiments detailed in this report used an internal dataset sourced from 17 internal collections (representing independent clinics) spread across North America, Europe, and Asia, consisting of T2 MRI acquisitions together with associated ADC and DWI. We illustrate the distribution of patients per collection in Figure 4. The dataset comprised a total of 2,040 patients. Among these, 1,393 patients had a biopsy (Gleason score), and all 2,040 patients had an associated PI-RADS score. As will be discussed in this section, the patients with PI-RADS score are mainly used to balance the collections for non-cancerous patients (i.e., PI-RADS less than three). We note that the datasets across vendors is unbalanced, with one collection (Collection 4) having significantly more patients than the others. This is in line with the distributions expected in FLUTE, and trade-offs stemming from this imbalance will be discussed in this report.

We also provide an illustration of the Gleason scores in Figure 5, which are used to classify patients with benign and malignant tumors. A Gleason score of zero indicates the least differentiated tumors, considered non-cancerous. A Gleason score of one represents a well-differentiated tumor pattern, two indicates a fairly well-circumscribed nodule, and three denotes a clearly infiltrative neoplasm extending into adjacent healthy prostate tissue. A score of four signifies non-separated glands, while a score of five indicates no glandular differentiation, bearing no resemblance to normal prostate tissue. Clinically significant prostate cancer is considered any score equal or above two, while one is considered to be a malignant nodule that looks similar to normal prostate tissue and is not clinically significant.

We observe that the majority of patients have Gleason scores of one or higher, indicating the presence of a malignant tumor. This imbalanced distribution is expected, as patients who undergo a biopsy are first screened by a clinician who recommends the procedure based on the

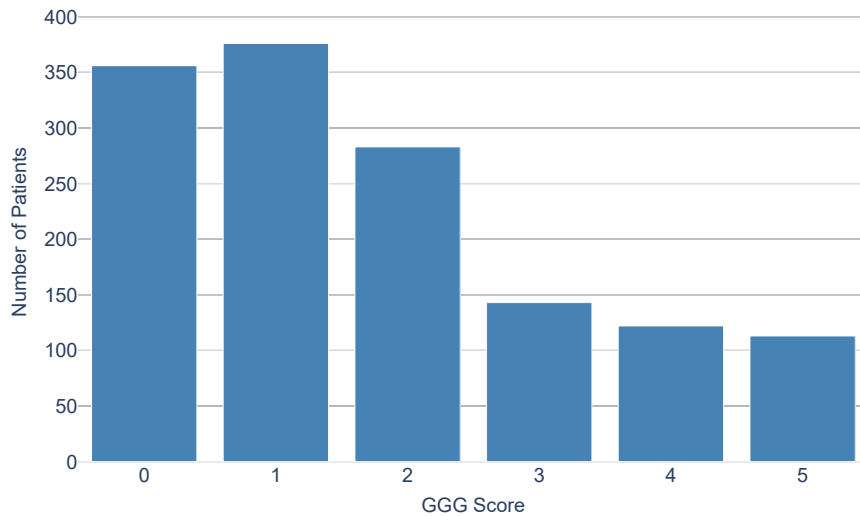


Figure 5: The Gleason grading score distribution in the dataset described in Figure 4.

suspicion of a malignant tumor. To address this imbalance, as mentioned above and detailed in the following sections, we will include patients where the absence of a malignant tumor is almost certain, given their low PI-RADS score and the recommendation to not undertake a biopsy.

The data was collected using three different scanner manufacturers with varying scanner models. Additionally, due to regional differences and varying acquisition protocols across clinics, we anticipate some variability in the final samples. To examine this, we present the distribution of mean values in the T2 in Figure 6, along with the distributions of minimum and maximum values, which are included in the Appendix, Figures 12 and 13. We notice that certain collections, such as Collection 7 and Collection 15, exhibit distinctly different distributions in terms of both mean and spread. Upon closer examination, we found that these collections are from the same geographical region and use the same scanner vendor. It is probable that these clinics employ a different configuration for their acquisition protocol. Such details will become important when normalizing the data and for experimenting with grouping multiple clinics across FL nodes.

### 3.1 Preprocessing and Normalization

As mentioned in Section 1, our goal is to predict the presence of a potentially malignant tumor in a patient using only imaging data, specifically T2, ADC, and DWI, which corresponds to a Gleason score of at least one. Although clinically significant prostate cancer is defined by a Gleason score of two or higher, our model focuses on identifying any potential anomaly. This approach increases the difficulty of the task because it requires the model to detect subtle variations that may not be as pronounced as those in higher Gleason score cases.

For this task, we classify patients with Gleason scores of one or higher as positive (cancer), while those with Gleason scores of zero are classified as negative (non-cancer). To further balance the Gleason score distribution, we also categorize patients with PI-RADS scores of

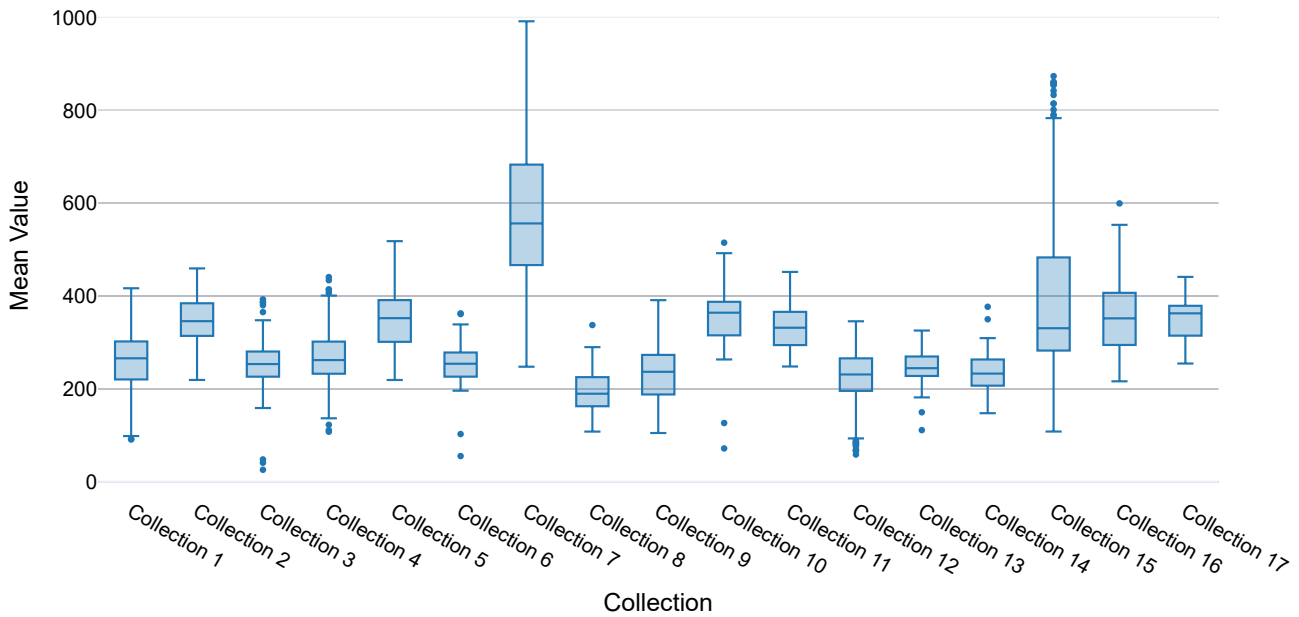


Figure 6: The distribution for the mean values of T2 volumes per collection.

one and two (who do not have a Gleason score) as negative. An illustration of this distribution is provided in Figure 7.

To normalize the data, given the distributions illustrated in Figures 6, 12, and 13, we use per volume min-max normalization [18], scaling intensities between 0 and 1. Using per volume normalization helps standardize the final distributions across different scanner vendors and clinics.

For ADC, min-max normalization can distort clinically relevant intensity information. Instead, we divide the entire volume by a constant, as commonly done in literature [19]. This approach preserves the absolute diffusion values and retains the diagnostic significance of ADC intensities.

For DWI, normalization was performed using the median intensity of the corresponding baseline image acquired without any diffusion weighting (b-value 0). The high level diffusion weighting (b-value 2000) image was divided by this median, and further divided by a constant to standardize distributions across all vendors.

## 4 Methods

We experiment with several methods, which we detail in this section. First, we describe the methodology used for pre-training the models that are used throughout all experiments. Next, we introduce the prostate cancer classification use-cases and the data pre-processing involved. Here, we explore two methods for classifying prostate cancer: (i) at the prostate level, where the models analyze the entire prostate and decides whether prostate cancer exists, and (ii) at the lesion level, which focuses on classifying whether a lesion is malignant or benign.

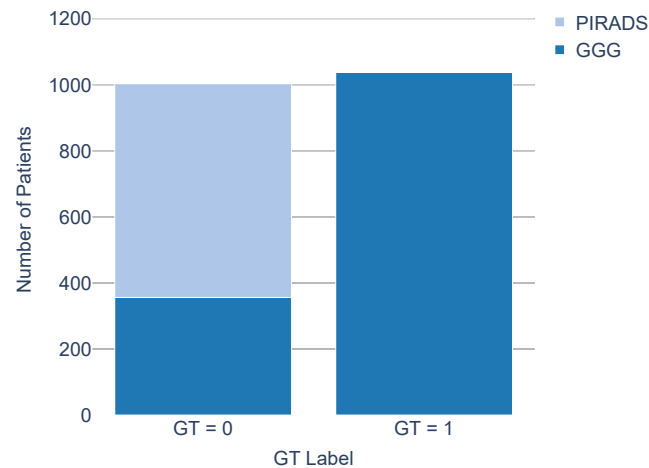


Figure 7: The distribution of ground truth labels based on Gleason score grading (GGG) and PI-RADS.

Subsequently, we present the methodology used to develop the fine-tuning and distillation algorithms in a FL network. This includes federated fine-tuning of the entire model (for benchmarks), federated additive fine-tuning (with adapters in the final stage or using LoRA), hybrid methods that combine federated fine-tuning with quantization, and distillation methods. All use-case related experiments start with the pre-trained model and adapt it through different methods.

## 4.1 Pre-Training Foundational Models

We experiment with two types of FMs: an organ-based FM focused on the prostate and a lesion-based FM trained exclusively on prostate lesions. For pre-training, we employ Masked Image Modelling (MIM), a self-supervised learning technique that reconstructs masked portions of an input. This process is illustrated in Figure 8. During pre-training, approximately 50% of the input sample is masked, and the model is trained to reconstruct the missing information. MIM is recognized as one of the top-performing image-based self-supervised algorithms [21, 3], making it a strong baseline for pre-training.

By randomly masking portions of an input image and tasking the model with filling in these gaps, the model learns to understand the contextual relationships between different parts of the image. This process encourages the model to develop a robust understanding of visual patterns, textures, object shapes and positions, which can then be transferred to downstream tasks such as image classification. The effectiveness of masked image modeling lies in its ability to exploit the natural correlations within images, enabling the model to generalize well from unlabeled data.

Hybrid approaches, combining MIM with other objectives such as contrastive learning, could improve performance and are worth exploring in future research [17]. For FLUTE, we anticipate that platform users will use readily available, open-source FMs from repositories such as Huggingface<sup>1</sup>.

<sup>1</sup><https://huggingface.co>

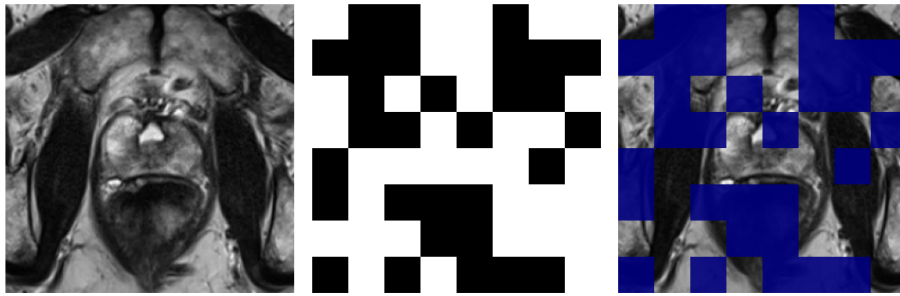


Figure 8: Sample input for **MIM** pre-training.

For pre-training, a collection of more than 4,000 samples was used for the prostate-based model, and a similar number of lesions for the lesion-based **FM**. This is in line with current literature. For example [Lee et al. \[11\]](#) used approximately 4,400 samples for pretraining and observed a significant improve in performance.

We also experiment with both convolution-based (CNN) and transformer-based architectures. While CNN based architectures are known to require less pre-training, transformer-based architectures reach similar and improved performance only with pre-training [9]. Concretely, we pre-train a ResNet based architecture, a Vision Transformer (**ViT**) [8] and a Shifted Window Transformer (**SWIN**) transformer [13].

## 4.2 Prostate Cancer Classification

This task was designed as a supervised binary classification task, where the model predicts either zero (non-cancer) or one (cancer) for each patient. The inputs consisted of three frames for each view (T2, ADC, DWI/B2000), stacked to form a nine-channel input image. For each view, the images were cropped around the prostate using existing segmentation masks available for the dataset, guiding the model to focus on the prostate as the region of interest. The spacing was standardized on the width and height dimension for all vendors and were padded or cropped at a standard resolution of 164x164. All architectures mentioned in the section above were trained, using pre-training or not. A visual illustration of this approach is shown in [Figure 9](#).

The dataset was divided into 80% for training, 10% for validation, and 10% for testing using stratified sampling to ensure each collection is represented in the final test set. To evaluate model performance, we computed the AUROC [14], which balances sensitivity and specificity. We employed a weighted binary cross entropy loss [20] and used the AdamW optimizer with a learning rate of  $10^{-4}$  [25]. These settings are quite standard, and were intentionally chosen to ensure easy reproducibility, allowing us to focus on evaluating the model's performance without introducing variability from hyperparameter tuning.

## 4.3 Lesion Based Cancer Classification

We also explore lesion-based classification, which involves classifying individual lesions as either benign or malignant. This approach offers several advantages. First, it reduces the input data size and the number of parameters, thereby optimizing communication costs in **FL**.

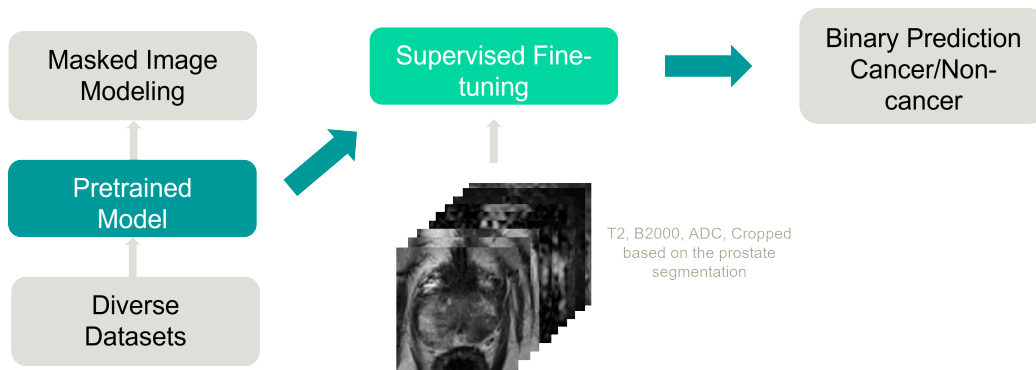


Figure 9: Illustration of training paradigms for pre-training and supervised fine-tuning.

Moreover, lesion-based classification aligns more closely with clinical practice, as clinicians typically evaluate individual lesions to determine their malignancy. By focusing on specific lesions rather than the entire prostate, the model can provide more targeted and clinically relevant insights.

Using the lesion segmentations available in our dataset, we follow a similar procedure as above, where we consider the slice of the lesion with the highest density and crop it with an additional margin of 10 pixels. Multiple patient-level lesions are identified and cropped using connected components [6]. Additional  $\pm 1$  slices are selected, and a similar procedure as illustrated in Figure 9 is used for training and testing.

This approach maintains the same number of channels but narrows the input along the width and height dimensions, making it more focused on one lesion. Stacking the images along the channel dimension doesn't introduce bias, since the different views are spatially aligned. Furthermore, we use the same data split, loss function, and optimizers as in the prostate-based experiments to ensure direct comparison.

## 4.4 Federated Fine-Tuning

To benchmark the fine-tuning and distillation algorithms, we initially conduct fine-tuning on the entire model for each client. Each collection in Figure 4 is treated as a node participating in the FL process. Given the stratified sampling used in the supervised learning experiments, for this step we just distribute the training and validation data specific to each collection to the respective nodes, with final testing performed on the server node.

In this setup, each client receives the pre-trained model and fine-tunes it on their local dataset. At each epoch, the clients send their updated models back to the server, which aggregates the results. We experiment with both Federated Average (FedAvg) [16] and Federated Adam (FedAdam) [5] because the data is imbalanced and adaptive aggregation algorithms such as FedAvg are expected to work better. After aggregation, the global model is returned to each client for evaluation on their corresponding validation dataset. Following this evaluation, a new round of local training commences.

To simulate the FL environment, and considering that the FLUTE platform is not yet deployed, we utilize FLSim [12]. This versatile simulation framework replicates all the expected features of the FLUTE platform, mimicking FL settings without the need for multiple physical

computational nodes for each collection.

## 4.5 Federated Additive Fine-Tuning

As discussed in Section 2, we explore two classes of FL additive fine-tuning algorithms.

First, we employ additive fine-tuning with an adapter in the final stage. In this approach, the FM remains frozen, and only the final layer is involved in FL. Initially, the FM is distributed to all clients. During each epoch, clients perform a forward pass over the data, and only the parameters of the final layer are transmitted to the server. The server aggregates these parameters and sends them back to the clients, initiating a new training round.

Second, we use additive fine-tuning with LoRA, experimenting with different sizes of added parameters: 16, 32, and 64. This settings is similar to the one above, just that at each training round the clients share all LoRA parameters with the server.

Considering the significant data imbalance, we also explore potential grouping strategies for clients. For instance, clients could be grouped by region or based on data distribution characteristics.

## 4.6 Hybrid Federated Additive Fine-tuning and Quantization

To further reduce communication costs in FL, we investigate adding quantization to the previously described federated algorithms. These methods are defined in D9.1 as hybrid methods. Quantization involves reducing the precision of the model parameters to lower the amount of data transmitted during training. While standard experiments use float32, float16, or mixed-precision formats, we quantize the parameters used in FL to 4-bits for our use-case. This approach reduces communication costs by up to eight times compared to float32 and twice compared to float16 training. In all our experiments, we use float32 and then apply 4-bit quantization.

## 4.7 Federated Distillation

The federated distillation methods discussed in D9.1 require a public dataset for supervised distillation in FL. However, since a public dataset is unavailable for the FLUTE use-case, we developed an alternative which involves initially distilling a larger model into a smaller one using unsupervised distillation before distributing the model to all nodes in FL.

For this purpose, we employ SEED, a technique that uses a larger network (acting as the Teacher) to transfer its representational knowledge to a smaller network (acting as the Student) in a self-supervised manner. In SEED, the student encoder is trained to replicate the similarity score distribution inferred by the teacher across a set of instances and their augmentations [7].

We begin with a pre-trained Swin model having 180M parameters and distill its knowledge into a smaller model with 25M parameters, which is used for subsequent experiments.

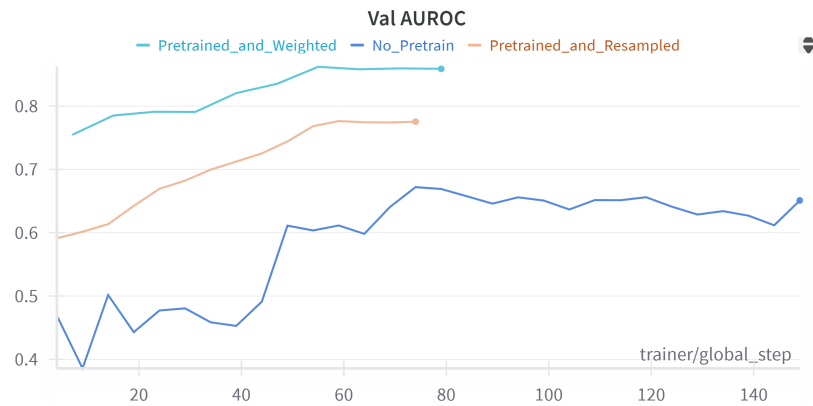


Figure 10: Comparison of validation performance for non-pretrained model versus fine-tuning the pre-trained model with weighting and fine-tuning the pre-trained model with dataset resampling. Experiments ran for 140 steps, batch size 128. We observe faster convergence for the pre-trained models.

## 5 Results

This section presents the empirical results of conducting the experiments based on the methodologies outlined in Section 4. Briefly, we start by testing the impact of using FMs versus training a model with normal initialization, and compare the prostate and lesion-based use-cases (Section 5.1).

Subsequently, we evaluate the FM in the FL exclusively on the lesion-based use case, as it demonstrated superior performance and scalability. These experiments include full model fine-tuning (Section 5.2), fine-tuning of additive components (Section 5.3), as well as combining additive fine-tuning with quantization (Section 5.4), and end with distillation (Section 5.5). All experiments are independent but are directly comparable, as they follow the same pipeline and use identical data splits.

### 5.1 Supervised Learning using Foundational Models

We first investigate the impact of pre-training on the prostate cancer classification task. In this context, we fine-tune a CNN-based ResNet50 model and compare its performance with a model trained from scratch using normal initialization. To address class imbalance, we employ two strategies: assigning a higher weight to underrepresented samples and resampling these samples multiple times.

We present the validation AUROC for the initial 140 steps in Figure 10. As noted in the literature, pre-training facilitates faster convergence. Additionally, we summarize the final results on the test dataset in Table 10 (Appendix), which shows superior performance for the pre-trained model, particularly when using loss weighting.

Furthermore, we employ this strategy to compare the three architectures initially proposed. The final results on the test set are summarized in Table 1 using AUROC and confidence inter-

Method	AUROC
VIT Prostate	79.67 (0.028)
ResNet50 Prostate	83.54 (0.026)
Swin Prostate	<b>85.29 (0.025)</b>

Table 1: Performance and confidence intervals for initial fine-tuning of three pre-trained models with distinct architectures on the prostate-based use-case. We observe the Swin transformer performs best

Method	AUROC
Swin Prostate	85.29 (0.025)
Swin Lesion	<b>88.17 (0.023)</b>

Table 2: Performance and confidence intervals for lesion vs. prostate based supervised fine-tuning using the Swin transformer architecture. We observe the lesion-based model performs best.

vals estimated using the Newcombe’s Wald Method [15]. We observe the Swin Transformer demonstrates superior performance over the other architectures. Consequently, we will use this architecture for further experimentation.

To establish a benchmark for the initial supervised tasks, we also evaluate the Swin Transformer on the lesion-based classification task, with results summarized in Table 2 using AUROC and confidence intervals estimated using the Newcombe’s Wald Method [15]. These findings indicate a significant improvement in performance when using the lesion-based algorithm. This outcome is expected because the lesion-based approach focuses on specific areas of interest, allowing the model to better distinguish between benign and malignant lesions.

## 5.2 Federated Fine-tuning

Given the superior performance of the Swin Transformer, we conduct a first benchmark for federated fine-tuning using the configuration outlined in Section 4.4. In this setup, each collection in Figure 4 is treated as a node in the FL network.

We compare the performance of two aggregation methods: **FedAvg**, a straightforward algorithm that averages model parameters and uses a constant learning rate, and **FedAdam**, a more sophisticated approach that employs adaptive learning rates based on momentum estimates. **FedAdam** is particularly effective in scenarios where the data distribution is non-IID, often outperforming simpler methods like **FedAvg** in such settings.

The results are summarized in Table 3 for tasks trained with prostate-based crops using AUROC and confidence intervals estimated using the Newcombe’s Wald Method [15]. We find that **FedAdam** achieves a higher AUROC, confirming our initial hypothesis. Previous observations from D9.1 noted that most algorithms in the literature use **FedAvg** instead of more advanced aggregation methods. Our current findings suggest that exploring more sophisticated aggregation algorithms, such as **FedAdam**, presents a promising avenue for future research in federated learning.

## 5.3 Federated Additive Fine-tuning

For federated additive fine-tuning, we analyzed the impact of optimizing only the final stage in FL, or adding **LoRA** adapters for all transformer layers of the model.

Model	Aggregator	AUROC
Swin Prostate	FedAdam	<b>83.41 (0.026)</b>
Swin Prostate	FedAVG	81.10 (0.028)

Table 3: Performance and confidence intervals for federated fine-tuning of the entire model using different aggregators. We observe a drop of performance when running the models in a FL network compared to Table 2.

In this scenario, we rely on the lesion-based data as it provides better results, and train Swin-based models using fine-tuning in the final stage and various LoRA ranks (16, 32 and 64), and compare it the centralized fine-tuning. The results are summarized in Table 4 using AUROC and confidence intervals estimated using the Newcombe’s Wald Method [15].

Method	Centralized AUROC (%)	FL AUROC (%)
LoRA Rank 16	84.74 (0.025)	<b>86.24 (0.024)</b>
LoRA Rank 32	86.02 (0.024)	82.36 (0.027)
LoRA Rank 64	<b>87.09 (0.024)</b>	83.83 (0.026)
Final-stage	85.22 (0.025)	85.13 (0.025)
Entire model fine-tuning	88.17 (0.023)	83.01 (0.026)

Table 4: Performance and confidence intervals for the lesion-based Swin model fine-tuned in a FL network and centralized. In the centralized setting, all training data is combined into a single collection and node, while in FL, each dataset from Figure 4 is assigned to a separate node in the network. We observe a consistent decrease in performance when running the model in FL, for both fine-tuning with LoRA and with final-stage adapters.

We observe that increasing the size of the LoRA has inconsistent effects on FL, potentially introducing some instability. For example, in centralized fine-tuning with LoRA adapters, increasing the size from 16 to 32 and then to 64 improves performance consistently, nearly matching that of full model fine-tuning. However, in FL, LoRA’s behavior is less predictable, with the best performance achieved at rank 16. Using additive parameters in the final stage shows more consistency between centralized and FL settings. Although the performance is comparable to using LoRA adapters, the final stage’s parameter count is way smaller than that of LoRA 16, suggesting greater efficiency.

Moreover, fine-tuning the entire model in FL leads to a more significant drop in performance. This behavior aligns with existing literature summarized in D9.1, as averaging the entire model across datasets with distinct distributions can result in diminished performance.

To understand how additive federated fine-tuning affects model parameter sizes and, consequently, communication overhead in FL, we illustrate the final model dimensions in Figure 11. We find that, based on the configuration, LoRA can reduce the number of trainable parameters by up to 98.1% and the final stage fine-tuning with up to 99.2%. While additive fine-tuning with adapters in the final stage is the most parameter-efficient, it does not yield the best accuracy improvements. The reduction in parameters directly translates to lower communication costs in FL. Depending on the data type used for training (float16 or float32), the total data transfer ranges from approximately 0.8 to 7 MB per client per epoch.

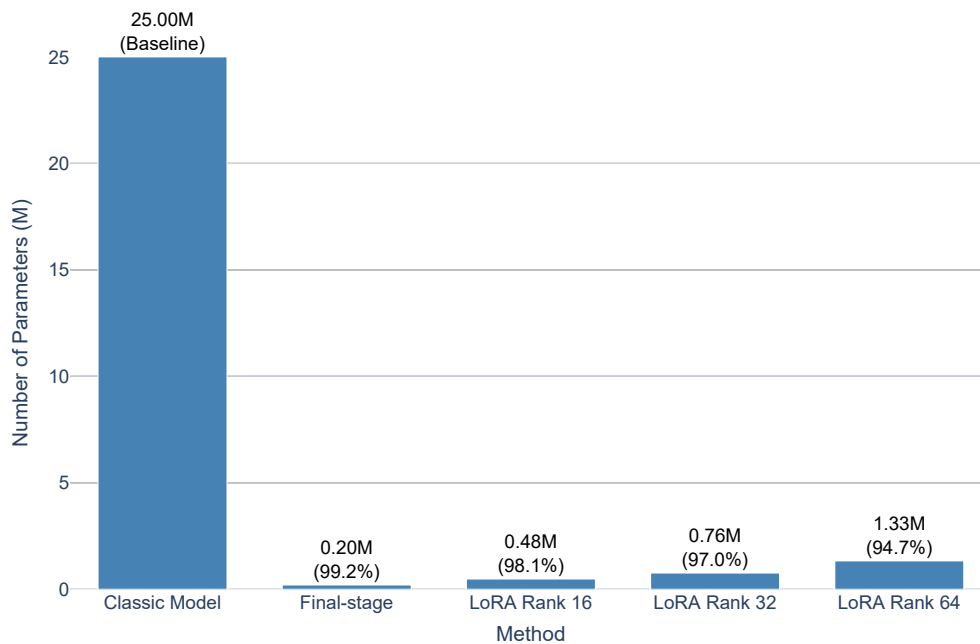


Figure 11: The effect of additive fine-tuning on the model parameters communicated across the network in FL. Fine-tuning with adapters in the final stage proves to be the most efficient, whereas LoRA-64 is the least efficient, as it introduces a larger number of parameters to the model.

For the FLUTE platform and, more broadly, for FL with additive fine-tuning, tailored experimentation is essential for each use-case. This is because performance outcomes do not consistently align with increases in model size, highlighting the need for case-specific investigations to optimize results.

### 5.3.1 Exploring Distinct Collection Grouping

To assess the impact of using imbalanced data across multiple collections, especially with a large number of collections, we conducted experiments by assigning multiple collections to a single computational node within the FL network. This setup involves multiple nodes performing inference on distinct data using the same model and communicating the results to the server.

First, we grouped the collections based on data distribution parameters (as shown in Figures 6, and Figures 13, and 12 from the Appendix) to approximately balance the data distributions across patients. We used three clients, aligning with the initial use-case configuration of the FLUTE platform. Additionally, we experimented with grouping the data collections by region, assigning them to nodes based on the available geographical data for all collections. As more data was available from Europe, we distribute Northern Europe to a region and Southern Europe to another region, resulting in four rather than three regions. This also ensures a more even distribution of data across nodes.

The results are summarized in Tables 5 and 6 using AUROC and confidence intervals estimated using the Newcombe’s Wald Method [15]. We find that the outcomes are consistent across different data distributions and align with the initial grouping per collection for addi-

Method	FL AUROC (%)
LoRA Rank 16	85.13 (0.025)
LoRA Rank 32	<b>86.38 (0.024)</b>
LoRA Rank 64	84.75 (0.025)
Final-stage	85.48 (0.025)
Entire model fine-tuning	83.91 (0.026)

Table 5: Performance and confidence intervals for the lesion-based Swin model in a 3-node FL fine-tuning setup, where the data is distributed across the nodes in the network by using the mean, maximum, and minimum values of T2 images as proxies to balance the collections (described in Section 5.2). We observe the performance is similar to distributing the data to a larger group of nodes, as summarized in Table 4.

Method	FL AUROC (%)
LoRA Rank 16	86.28 (0.024)
LoRA Rank 32	<b>86.62 (0.024)</b>
LoRA Rank 64	85.21 (0.025)
Final-stage	85.5 (0.025)
Entire model fine-tuning	84.56 (0.025)

Table 6: Performance and confidence intervals for the lesion-based Swin model in a 4-node FL fine-tuning setup, where the data is distributed across the nodes in the network using the geographical distribution (described in Section 5.2). We observe the performance is similar to distributing the data to a larger group of nodes (Table 4) or to distributing it based on imaging characteristics (Table 5).

tive fine-tuning methods. However, for the entire model fine-tuning we observe an increase in performance when using the grouping using geographical distributions. This suggests that federated fine-tuning remains robust even with unevenly distributed collections, as both LoRA and additive fine-tuning in the final stages demonstrate robustness to such variations.

## 5.4 Hybrid Federated Additive Fine-tuning and Quantization

We further investigate quantizing the model parameters to 4-bits. For this experiment, we use the initial setup where all collections are distributed across nodes. The results are summarized in Table 7 using AUROC and confidence intervals estimated using the Newcombe’s Wald Method [15] (and are comparable to non-quantized results in Table 4).

We observe that the decrease in performance is relatively minor compared to the significant gains in communication efficiency. Quantizing to 4-bits reduces the communication costs by eight times, making it a viable strategy for optimizing FL systems in the FLUTE use-case. This approach highlights the potential for achieving a favorable trade-off between model performance and communication overhead, especially in scenarios where bandwidth and resource constraints are critical.

## 5.5 Distillation

After distilling a larger Swin model (180M parameters) into a smaller model (25M parameters), equivalent to the model used for the previous experiments, we employed the distilled model in a FL scenario. This setup mirrors the experiments discussed above, where each collection represents a node in a FL network, receiving the distilled model.

The results are summarized in Table 8 using AUROC confidence intervals estimated using the Newcombe’s Wald Method [15]. We observe that the performance is comparable to using the pre-trained model directly, with only a slight increase that is not conclusive in these ex-

Method	FL AUROC (%)
QLoRA Rank 16	84.54 (0.025)
QLoRA Rank 32	<b>85.9 (0.024)</b>
QLoRA Rank 64	85.65 (0.025)
QFinal-stage	85.1 (0.025)
QEntire model fine-tuning	83.64 (0.026)

Table 7: Performance and confidence intervals for the lesion-based model for FL using quantized parameters to 4bit, reducing the communication costs from 32b to 4b per parameters. We observe only a small decrease in performance when comparing with 32b training (Table 4).

Method	FL AUROC (%)
LoRA Rank 16	86.02 (0.024)
LoRA Rank 32	<b>86.4 (0.024)</b>
LoRA Rank 64	85.7 (0.025)
Final-stage	84.34 (0.026)
Entire model fine-tuning	83.34 (0.026)

Table 8: Performance and confidence intervals for the lesion-based model in FL when using a distilled model for pre-training, following the method outlined in Section 4.7. We observe a very small increase in performance when using the distilled model compared to using the pre-trained model (Table 4).

periments. We anticipate more significant results once additional data is incorporated or the FLUTE use-case is fully implemented.

## 6 Discussion

This report presents the results of developing and running federated fine-tuning and distillation algorithms on an internal dataset for the FLUTE use-case of detecting prostate cancer. In this section, we discuss general observations and specific insights related to the use-cases

First, we observe that using FedAdam as an aggregator yields better results. This aligns with the conclusions of D9.1, where most articles used federated average, and it was recommended to explore more aggregation techniques. For FLUTE, this suggests the need to implement more aggregator algorithms within the platform. It also indicates that exploring custom aggregators could enhance performance, making this a promising area for future research.

Second, we notice a slight decrease in performance in FL settings, with an approximately 2% drop in AUROC. This aligns with related work and the findings of D9.1. Although we attempted to mitigate this by addressing data imbalance and distribution shifts – for example, by grouping collections into fewer nodes – the performance drop persisted. It is likely that better aggregators or custom averaging techniques, which use different momentum estimates based on the collections (similar to federated personalization), could alleviate this issue. Additionally, further fine-tuning of hyper-parameters might help bridge this gap, as we aimed to maintain standard training settings across all experiments to ensure consistency and reproducibility.

Third, we observe that for the FLUTE use-case, the performance differences between using federated fine-tuning in the final stage and using LoRA (which introduces additional parameters) are not substantial. Since both techniques significantly reduce the number of parameters and improve communication costs, we recommend incorporating both options into

the FLUTE platform. This flexibility allows users with stricter communication constraints to prioritize lower communication costs, even if it means accepting a slight reduction in performance (therefore prioritizing federated fine-tuning with adapters in the final stage).

Fourth, we find that hybrid approaches, such as applying quantization to **LoRA** parameters, significantly reduce communication costs with only a minimal drop in performance. This aligns with the findings of D9.1. Consequently, we recommend integrating quantization techniques into the FLUTE platform to optimize communication efficiency.

Fifth, we observe that federated additive fine-tuning for **FMs** yields better results than full-model fine-tuning. This is expected, as pre-trained models already provide high-quality embeddings. Therefore, altering the pre-trained parameters on a small subset of data can degrade the embeddings, causing them to lose essential properties. By focusing on fine-tuning a limited number of parameters, federated additive fine-tuning leverages the benefits of pre-training while preserving the integrity of the embeddings.

Lastly, we observe that performing distillation prior to **FL** does not yield a significant increase in performance. This outcome may be attributed to the relatively small datasets used in these experiments. We anticipate more conclusive results once larger datasets are incorporated or additional iterations of the FLUTE use-case are conducted.

## 7 Conclusions

The development and implementation of federated fine-tuning and distillation algorithms for the FLUTE use-case of detecting prostate cancer have yielded several key insights. The use of **FedAdam** as an aggregator demonstrates superior performance, highlighting the need for further exploration and implementation of diverse aggregation techniques within the FLUTE platform. The slight performance decrease observed in **FL** settings suggests that improved aggregators or custom averaging techniques could mitigate this issue, alongside more nuanced hyper-parameter tuning. The comparable performance of federated fine-tuning in the final stage and **LoRA** indicates that both methods should be integrated into FLUTE, offering users flexibility based on their communication constraints. Hybrid approaches, such as quantization of **LoRA** parameters, significantly improve communication efficiency with minimal performance loss, underscoring their value in optimizing **FL** systems. Furthermore, federated additive fine-tuning outperforms full-model fine-tuning by preserving the quality of pre-trained embeddings, reinforcing its effectiveness in leveraging pre-trained models. Last, while distillation did not provide a significant performance advantage, it successfully reduced the model size considerably while preserving competitive performance.

# A Appendix

## A.1 Appendix

For the T2 volumes, we additionally computed the distribution for the minimum value (see Figure 12) and the maximum value (see Figure 13) across all collections. The minimum intensity distribution show a consistent lower bound across collections (beside Collection seven) while the maximum intensity distribution show a broader spread of values which reflect the differences in scanner or acquisition protocols across different collections. For all the distributions computed (minimum, maximum and mean intensity value) across the T2 volumes, the Collection seven deviates from the others, suggesting it may be an outlier.

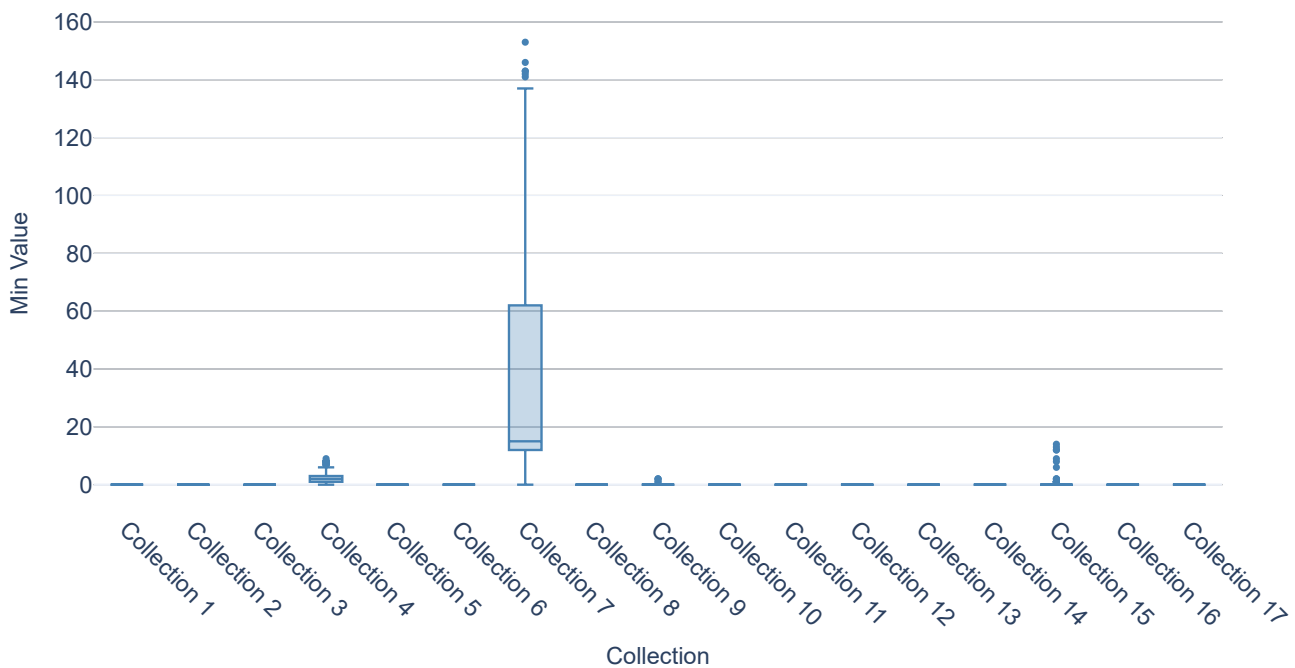


Figure 12: The distribution for the minimum value of T2 volumes per collection.

In Table 9 we summarized the results for the prostate-based model in different scenarios: without LoRA (Classic) and with LoRA rank 16, 32 and 64 in both centralized and FL scenarios. The prostate-based model is inferior to the lesion-based model in both centralized and FL settings, suggesting that a lesion-based model is more suitable.

Table 10 summarizes the results from running the ResNet-50 experiments on the prostate-based data with and without pre-training, with different data balance strategies.

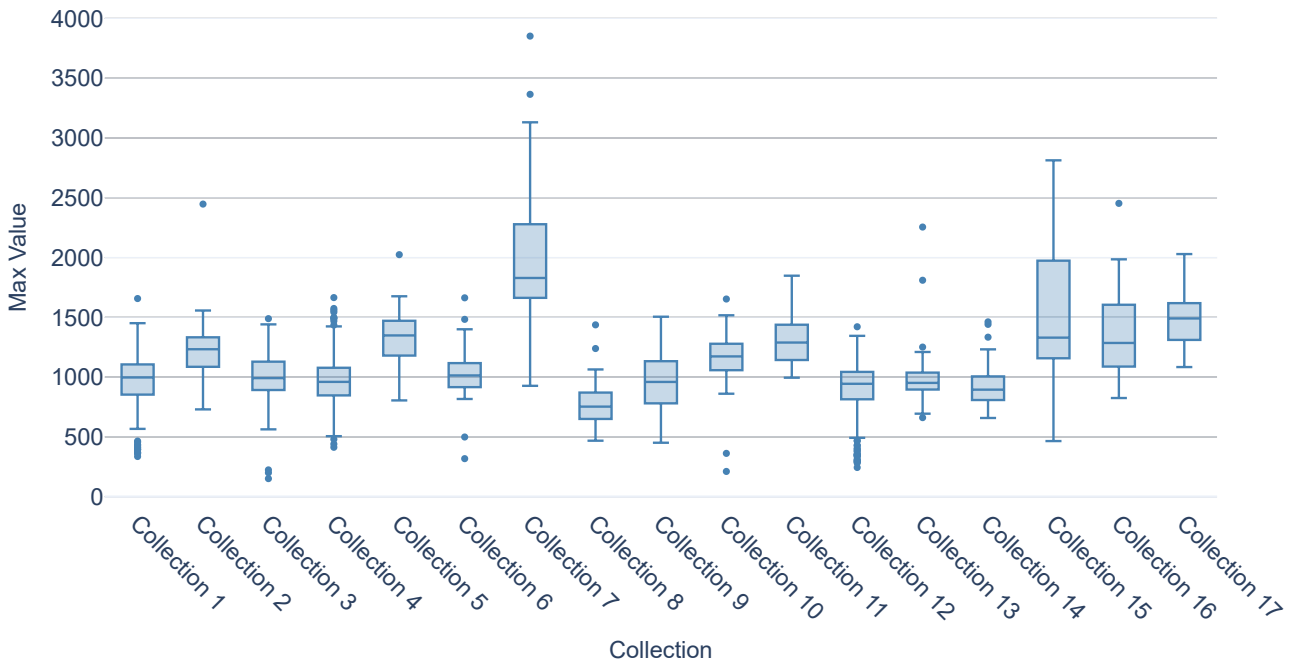


Figure 13: The distribution for the maximum value of T2 volumes per collection.

Method	Centralized AUROC (%)	FL AUROC (%)
LoRA Rank 16	74.51 (0.031)	66.22 (0.033)
LoRA Rank 32	80.95 (0.028)	73.12 (0.031)
LoRA Rank 64	86.26 (0.024)	83.83 (0.026)
Final-stage	73.14 (0.031)	72.48 (0.031)
Entire model fine-tuning	85.29 (0.025)	76.95 (0.03)

Table 9: Performance and confidence intervals for the prostate-based Swin model for FL and centralized training.

Method	AUROC (%)
No pre-training	73.41 (0.031)
Pre-trained using resampling	77.65 (0.029)
Pre-trained using loss weights	83.54 (0.026)

Table 10: Performance and confidence intervals on the test set when comparing training a ResNet-50 architecture with normal initialization and with using the pre-trained model on the prostate-based task. Two strategies to overcome data imbalance are also explored.

## References

- [1] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv:2108.07258*, 2021.
- [2] Sijia Chen, Ningxin Su, and Baochun Li. Calibre: Towards fair and accurate personalized federated learning with self-supervised learning.
- [3] Zekai Chen, Devansh Agarwal, Kshitij Aggarwal, Wiem Safta, Mariann Micsinai Balan, and Kevin Brown. Masked image modeling advances 3d medical image analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1970–1980, 2023.
- [4] Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, Matt Barnes, and Gauri 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.
- [5] Xiumei Deng, Jun Li, Kang Wei, Long Shi, Zeihui Xiong, Ming Ding, Wen Chen, Shi Jin, and H Vincent Poor. Towards communication-efficient federated learning via sparse and aligned adaptive optimization. *arXiv preprint arXiv:2405.17932*, 2024.
- [6] Luigi Di Stefano and Andrea Bulgarelli. A simple and efficient connected components labeling algorithm. In *Proceedings 10th international conference on image analysis and processing*, pages 322–327. IEEE, 1999.
- [7] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. *arXiv preprint arXiv:2101.04731*, 2021.
- [8] Kai Han, Yunhe Wang, Hanqing Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.
- [9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.
- [10] Yuting He, Fuxiang Huang, Xinrui Jiang, Yuxiang Nie, Minghao Wang, Jiguang Wang, and Hao Chen. Foundation model for advancing healthcare: challenges, opportunities and future directions. *IEEE Reviews in Biomedical Engineering*, 2024.
- [11] Jeong Hoon Lee, Cynthia Xinran Li, Hassan Jahanandish, Indrani Bhattacharya, Sulaiman Vesal, Lichun Zhang, Shengtian Sang, Moon Hyung Choi, Simon John Christoph Soerensen, Steve Ran Zhou, et al. Prostate-specific foundation models for enhanced detection of clinically significant cancer. *arXiv e-prints*, pages arXiv–2502, 2025.
- [12] Li Li, Jun Wang, and ChengZhong Xu. Flsim: An extensible and reusable simulation framework for federated learning. In *International conference on simulation tools and techniques*, pages 350–369. Springer, 2020.

- [13] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
- [14] Matthew McDermott, Haoran Zhang, Lasse Hansen, Giovanni Angelotti, and Jack Galifant. A closer look at auROC and aupRC under class imbalance. *Advances in Neural Information Processing Systems*, 37:44102–44163, 2024.
- [15] Robert G Newcombe. Confidence intervals for an effect size measure based on the mann–whitney statistic. part 1: general issues and tail-area-based methods. *Statistics in medicine*, 25(4):543–557, 2006.
- [16] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. A performance evaluation of federated learning algorithms. In *Proceedings of the second workshop on distributed infrastructures for deep learning*, pages 1–8, 2018.
- [17] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- [18] SGOPAL Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.
- [19] Christian Roest, TC Kwee, A Saha, JJ Fütterer, Derya Yakar, and H Huisman. Ai-assisted biparametric mri surveillance of prostate cancer: feasibility study. *European radiology*, 33(1):89–96, 2023.
- [20] Usha Ruby, Vamsidhar Yendapalli, et al. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, 9(10), 2020.
- [21] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9653–9663, 2022.
- [22] Yiyuan Yang, Guodong Long, Tao Shen, Jing Jiang, and Michael Blumenstein. Dual-personalizing adapter for federated foundation models. *arXiv:2403.19211*, 2024.
- [23] Shaoting Zhang and Dimitris Metaxas. On the challenges and perspectives of foundation models for medical image analysis. *Medical image analysis*, 91:102996, 2024.
- [24] Weiming Zhuang, Chen Chen, and Lingjuan Lyu. When foundation model meets federated learning: Motivations, challenges, and future directions. *arXiv:2306.15546*, 2023.
- [25] Zhenxun Zhuang, Mingrui Liu, Ashok Cutkosky, and Francesco Orabona. Understanding adamw through proximal methods and scale-freeness. *arXiv preprint arXiv:2202.00089*, 2022.