# **Gerchberg–Saxton Transformations for Privacy-Preserving Biomedical AI**

**Reproducibility Framework for Local and Federated Learning**

This repository contains the full experimental framework and code used in the study:  
 ***"Gerchberg–Saxton Transformations for Privacy-Preserving Biomedical AI in Local and Federated Learning."***

We provide code, data processing pipelines, model training workflows, and evaluation modules to enable full or partial reproduction of our results in both high-compute and minimal-resource environments. **Please make sure you unzip all zipped files before executing any notebooks and download the *inverted\_images* from releases!**

**🔧 System Requirements**

The original experiments were executed in the **Google Cloud Vertex AI** platform under the following configuration:

* **Platform**: Google Cloud Console – Vertex AI JupyterLab Notebook
* **Machine Type**: n1-standard-16 (16 vCPUs, 60 GB RAM)
* **GPU**: NVIDIA Tesla P100 (1x)
* **Environment**: TensorFlow Enterprise 2.16 (with Intel® MKL-DNN/MKL)
* **CUDA Version**: 12

✅ **Colab-Compatible**: Most notebooks are adapted to run in Google Colab (Free Tier) for ease of reproducibility, with exceptions noted below.

## **📂 Repository Structure**

The notebooks are modular and organized for plug-and-play execution. Below is a summary of each notebook’s purpose:

### **1. 1\_Load\_and\_Preprocess\_Data.ipynb**

* Downloads and preprocesses the Brain MRI dataset (from Kaggle).
* Splits data across 3 clients and a test set to simulate federated learning.
* **Essential** for initializing the dataset used throughout the framework.

### **2. 2\_GS\_Transform\_Data.ipynb**

* Applies Gerchberg–Saxton transformations to the dataset.
* Generates privacy-preserving data representations.
* **Required** for GS-based experiments.

### **3. 3\_Train\_FL\_Model.ipynb *(Optional)***

* Trains federated models (3 local + 1 global) on both benchmark and GS-transformed data.
* Pretrained models are already provided in the repository.
* **Use this** only if you wish to retrain models with custom settings.

### **4. 4\_Evaluate\_(pre)trained\_Models.ipynb**

* Evaluates both benchmark and GS models.
* Can be run using your own models or the pretrained ones provided.

## **🧪 Advanced Evaluation Modules**

The following experiments extend the core evaluation, focusing on privacy vulnerabilities and attacks. Some require high-compute resources.

### **🔐 Membership Inference Attack (MIA)**

#### **Primary Notebook: Membership\_Inference\_Plot\_Results.ipynb**

* Plots results from precomputed MIA evaluations on both GS and benchmark models.
* Uses stored results for full reproducibility in low-resource environments.

#### **Full MIA Reproduction:**

To regenerate MIA results:

* Use tensorflow\_fl\_env.yaml to configure your environment.
* Run on a machine with specs similar to the ones listed above.
* Adjust all file paths accordingly.

### **🔍 Model Inversion Attacks**

#### **White-box Inversion**

* Can be executed in **Google Colab** (T4 GPU recommended).
* Notebooks included:
  + inversion\_whitebox\_bench.ipynb
  + inversion\_whitebox\_gs20p.ipynb

#### **Black-box Inversion *(High Resource Required)***

* Requires pytorch\_fl\_env.yaml and high-compute environment (e.g., P100 GPU).
* Notebooks:
  + inversion\_blackbox\_bench.ipynb
  + inversion\_blackbox\_gs20p.ipynb

⚠️ **Note**: Due to their high computational cost, **all outputs and generated images** from these notebooks are included in the repository. You can skip these notebooks and proceed directly to evaluation.

### **📊 Final Evaluation**

#### **full\_evaluation\_blackbox.ipynb**

* Aggregates and evaluates final metrics, including attack success rates and model utility.
* Fully runnable in **Google Colab (free tier)** using pre-generated results and images.

## **🚀 How to Run**

1. **Clone the repository** to your local system or directly into your Google Drive:

git clone https://github.com/YOUR-USERNAME/YOUR-REPO-NAME.git

1. Open any notebook in Google Colab (recommended) or your own Jupyter environment.
2. Modify file paths as necessary based on your environment or Drive mount point.
3. Follow the inline instructions in each notebook block to execute.

## **📁 Pretrained Models & Precomputed Results**

The repository includes:

* Pretrained federated models (for both GS and benchmark datasets)
* Membership inference outputs
* Black-box and white-box inversion images
* Intermediate results required for downstream evaluations

This ensures full reproducibility **even without access to high-performance hardware.**

## **📄 Environment Setup**

To recreate the original environment used for compute-intensive notebooks, install dependencies using the provided YAML files:

# TensorFlow environment (for MIA)

conda env create -f tensorflow\_fl\_env.yaml

# PyTorch environment (for black-box inversion)

conda env create -f pytorch\_fl\_env.yaml

## **📬 Questions or Issues?**

For questions, clarifications, or reproducibility concerns, feel free to reach out to the corresponding author via email or GitHub Issues.

## **🧠 Citation**

If you use this codebase in your research, please cite:

**[Insert citation once your paper is published]**