ADVANCED PRIVACY-PRESERVING FEDERATED LEARNING (APPFL) FRAMEWORK AND ITS INTEGRATION WITH MONAI

ZILINGHAN LI

Machine Learning Engineer
Data Science and Learning Division,
Argonne National Laboratory
zilinghan.li@anl.gov

MONAI FL WG 2025/02/11

APPFL+MONAI

MONAI FL module provides a MonaiAlgo class, which provides train, evaluate, and get_weights functions to enable federated learning by leveraging a collection of medical imaging models available in MONAI bundles.

MONAI Bundle Reference Implementations

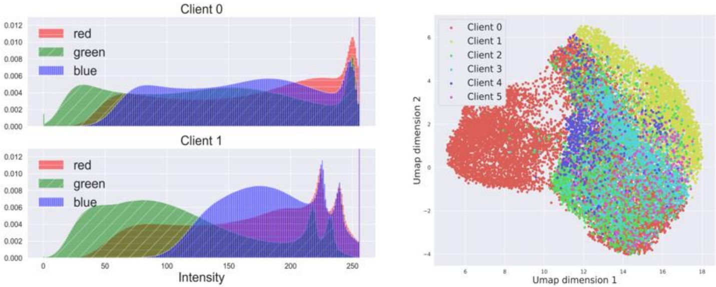
```
class monai.fl.client.MonaiAlgo(bundle_root, local_epochs=1,
send_weight_diff=True, config_train_filename='configs/train.json',
train_kwargs=None, config_evaluate_filename='default', eval_kwargs=None,
config_filters_filename=None, disable_ckpt_loading=True,
best_model_filepath='models/model.pt',
final_model_filepath='models/model_final.pt', save_dict_key='model',
data_stats_transform_list=None, eval_workflow_name='train',
train_workflow=None, eval_workflow=None)
```

[source]

Implementation of `ClientAlgo` to allow federated learning with MONAI bundle configurations.

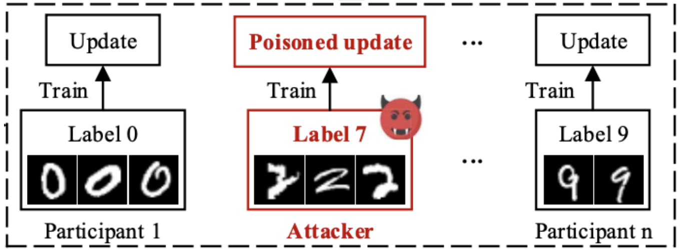
CHALLENGES IN FL

Various Challenges of Federated Learning Due to its Distributed Nature



➢ Heterogenous Data [1]

➢ Heterogenous Compute and Infrastructure



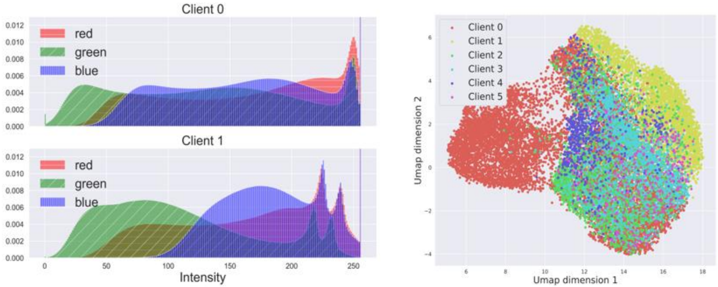
➢ Malicious Attack [2]

➢ Data Reconstruction [3]

➢ Cumbersome Setup

CHALLENGES IN FL

Various Challenges of Federated Learning Due to its Distributed Nature



- Heterogenous Data

Solution 1:

Utilize server-side momentum or other optimizations to avoid drastic changes in global model. (for example, FedAvgM [4], FedAdam, FedAdagrad, FedYogi [5], etc.)

$$\omega \leftarrow \omega - \Delta\omega \textit{ where}$$
$$\Delta\omega = \Sigma p_i \Delta\omega_j$$

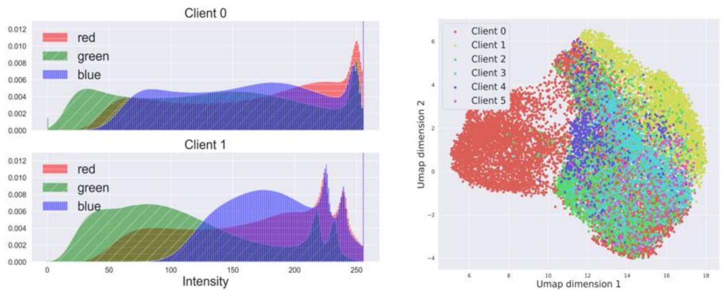
Traditional FedAvg

$$v \leftarrow \beta v + (1-\beta)\Delta\omega$$
$$\omega \leftarrow \omega - v$$
$$\textit{and } \Delta\omega = \Sigma p_i \Delta\omega_j$$

FedAvg with momentum

CHALLENGES IN FL

Various Challenges of Federated Learning Due to its Distributed Nature



- ➢ Heterogenous Data

Solution 2:

Leverage proximal term or variance reduction correction term in client local training to prevent local training from drifting too far way from the global model. (for example: SCAFFOLD [6], FedProx [7], etc.)

$$\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \eta_l(g_i(\boldsymbol{y}_i) \boxed{+ \boldsymbol{c} - \boldsymbol{c}_i)}$$ ⟵ SCAFFOLD

$$\min_w h_k(w;\ w^t) = F_k(w) \boxed{+ \frac{\mu}{2}\|w - w^t\|^2}$$ ⟵ FedProx

CHALLENGES IN FL

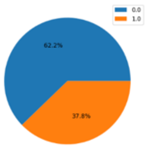
Various Challenges of Federated Learning Due to its Distributed Nature

Data Readiness Report

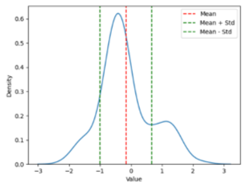
Client ID	class_imbalance	sample_size	num_classes	data_shape	data_range	overall_sparsity	class_distribution	outlier_proportion
Zilinghan Li - AWS	0.17	172	2	(172, 13)	{'min': -2.42, 'max': 6.46}	0.0	{0.0: 107, 1.0: 65}	0.04
Kaveen Hiniduma - OSU	inf	30	1	(30, 13)	{'min': -2.95, 'max': 5.29}	0.08	{1.0: 30}	0.06
Ravi Madduri - Argonne	0.06	199	2	(199, 13)	{'min': -3.61, 'max': 9.9}	0.0	{0.0: 108, 1.0: 91}	0.03
Shilan He - NCSA	0.39	85	2	(85, 13)	{'min': -6.4, 'max': 4.47}	0.0	{1.0: 66, 0.0: 19}	0.04

Plots for Client ID: Zilinghan Li - AWS

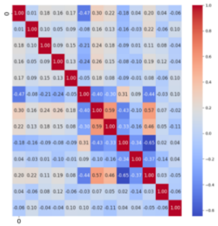
Class Distribution Plot



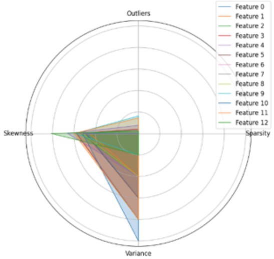
Data Distribution Plot



Feature Correlation Plot



Feature Statistics Plot



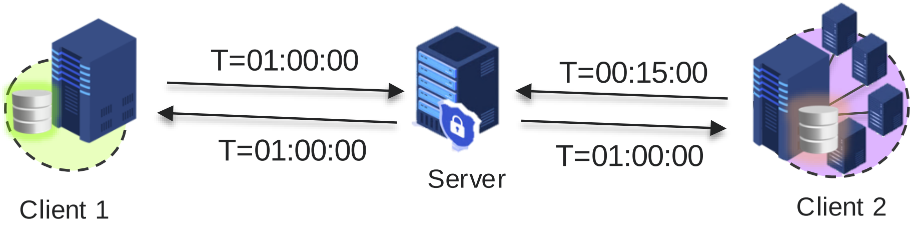
CHALLENGES IN FL

Various Challenges of Federated Learning Due to its Distributed Nature

➢ Heterogenous Compute and Infrastructure

- As the computing capabilities of client machines could have large variance, clients may take significantly different amount of time to finish one local training round.

- Synchronous FL algorithms, where the server waits for all clients to send the local models back, suffer from resource wastage.



Lots of resources are wasted for powerful clients.

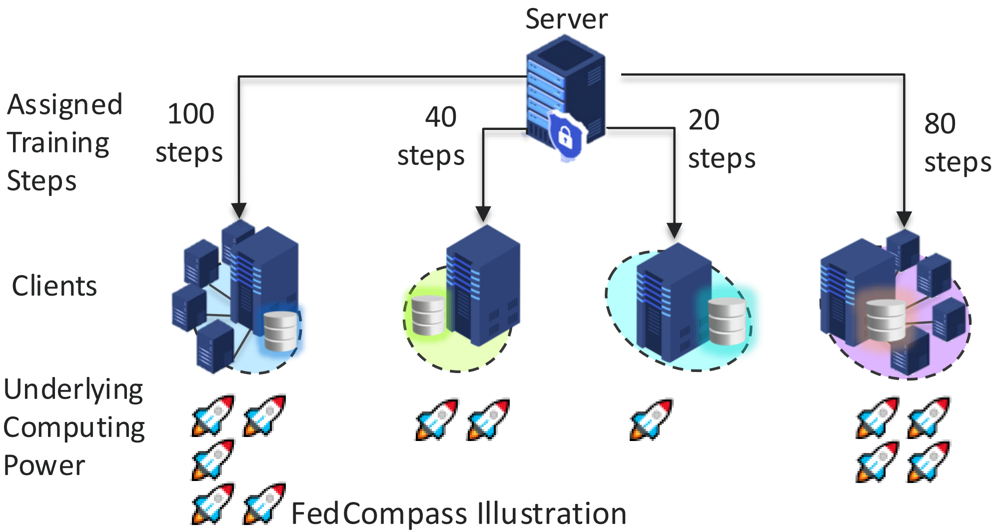
Resource wastage in synchronous FL.

CHALLENGES IN FL

Various Challenges of Federated Learning Due to its Distributed Nature

- Heterogenous Compute and Infrastructure

Asynchronous FL – which updates global model immediately once receiving local model from each client – can improve efficiency for FL in heterogeneous computing environments. (For example, FedAsync [9], FedBuff [10], FedCompass [11], etc)

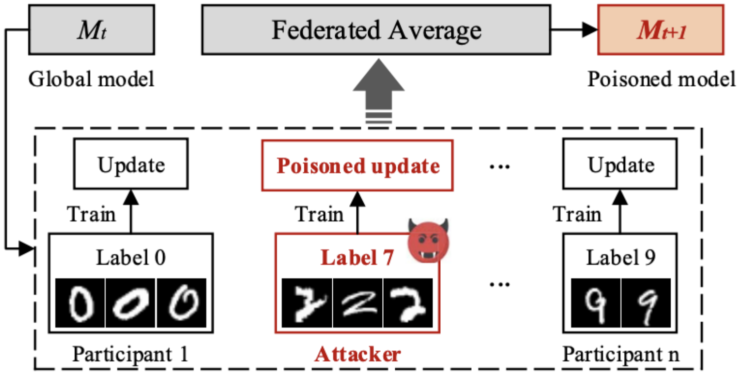


FedCompass Illustration

Assigning local training steps proportional to client's computing power.

CHALLENGES IN FL

Various Challenges of Federated Learning Due to its Distributed Nature



- Malicious Attack

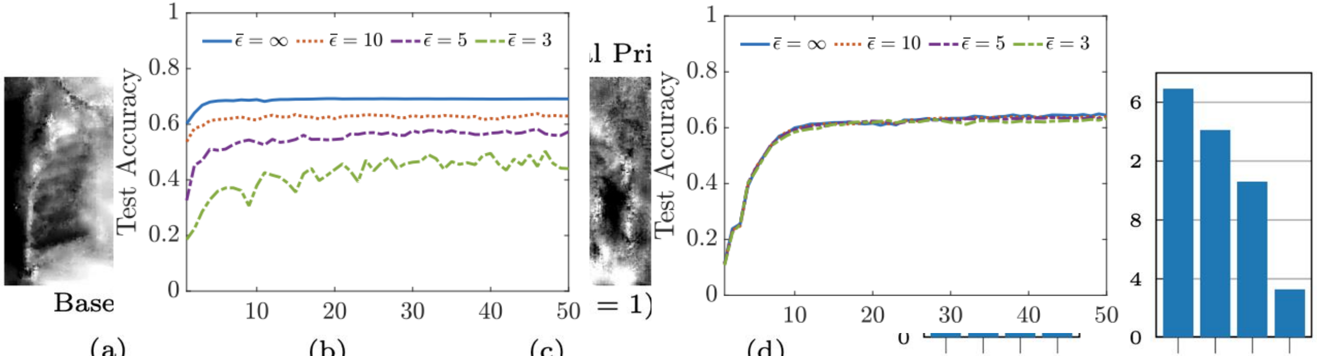
- Some clients may try to attack the FL training process by sending poisoned updates for aggregation.
- Algorithmic solutions include using a small central validation set and decide whether to drop certain client updates [2].
- System level, it is important to build a secure and trusted federation with user authentication systems [12].

CHALLENGES IN FL

Various Challenges of Federated Learning Due to its Distributed Nature

- Data Reconstruction

- Data reconstruction is another type of attack to FL.
- FL itself is not privacy preserving. The training data can be reversely constructed from model gradients.
- Differential privacy (DP), which adds some noise to model parameters, can significantly increase the difficulty of reconstruction [13].



Certain privacy-preserving FL algorithms (e.g. IIADMM) have less performance loss with DP [14].

CHALLENGES IN FL

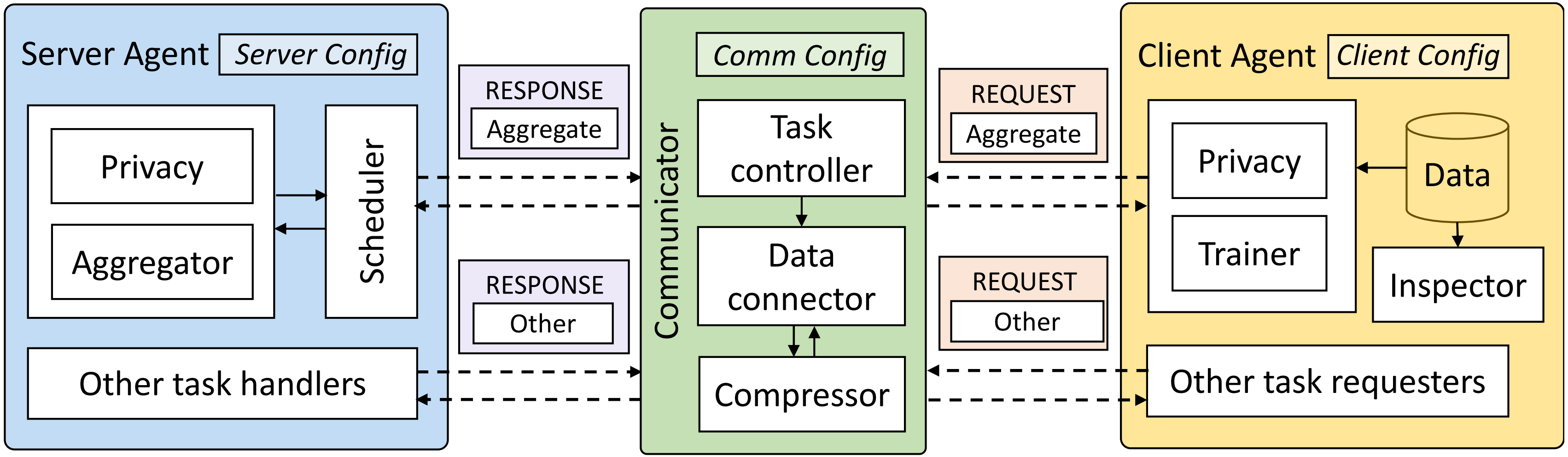
Various Challenges of Federated Learning Due to its Distributed Nature

- Cumbersome Setup

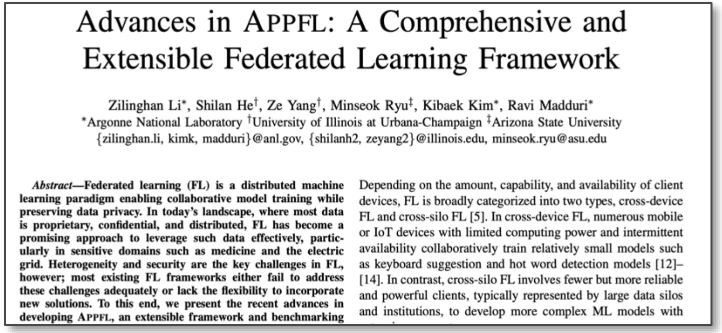
- Due to the distributed nature of federated learning, setting up FL experiments can be tedious for domain experts.
- Coordination of distributed training can be tedious as well, especially for sites using scheduling systems.
- Some client devices (e.g., compute nodes of some HPC) may not even have direct internet access.
- More efficient data transmission is needed as model gets larger.
- And so on…

APPFL alleviates those issues by supporting a versatile communication stack [12].

APPFL FRAMEWORK DESIGN



APPFL FRAMEWORK



Advances in APPFL: A Comprehensive and Extensible Federated Learning Framework

Zilinghan Li*, Shilan He†, Ze Yang†, Minseok Ryu‡, Kibaek Kim*, Ravi Madduri*
*Argonne National Laboratory †University of Illinois at Urbana-Champaign ‡Arizona State University
{zilinghan.li, kimk, madduri}@anl.gov, {shilanh2, zeyang2}@illinois.edu, minseok.ryu@asu.edu

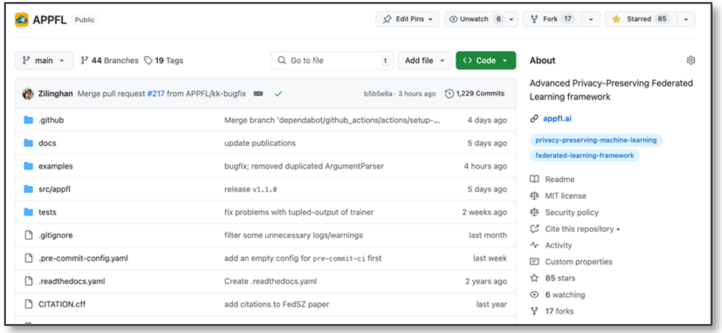
Abstract—Federated learning (FL) is a distributed machine learning paradigm enabling collaborative model training while preserving data privacy. In today's landscape, where most data is proprietary, confidential, and distributed, FL has become a promising approach to leverage such data effectively, particularly in sensitive domains such as medicine and the electric grid. Heterogeneity and security are the key challenges in FL, however; most existing FL frameworks either fail to address these challenges adequately or lack the flexibility to incorporate new solutions. To this end, we present the recent advances in developing APPFL, an extensible framework and benchmarking

Depending on the amount, capability, and availability of client devices, FL is broadly categorized into two types, cross-device FL and cross-silo FL [5]. In cross-device FL, numerous mobile or IoT devices with limited computing power and intermittent availability collaboratively train relatively small models such as keyboard suggestion and hot word detection models [12]–[14]. In contrast, cross-silo FL involves fewer but more reliable and powerful clients, typically represented by large data silos and institutions, to develop more complex ML models with

Manuscript [12]

Framework Design Description

- Framework overview
- Addressed challenges
- Evaluations
- Additional case studies
- …

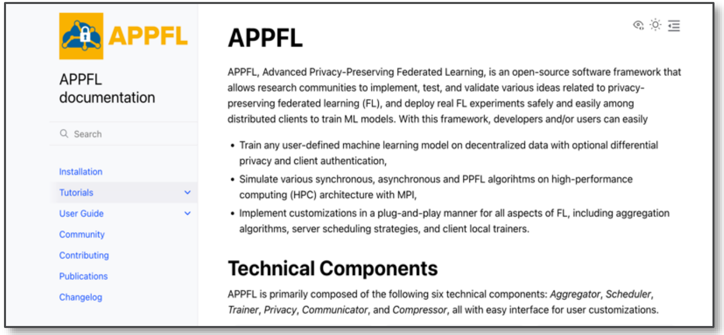


Open-source Code

Source code on Github

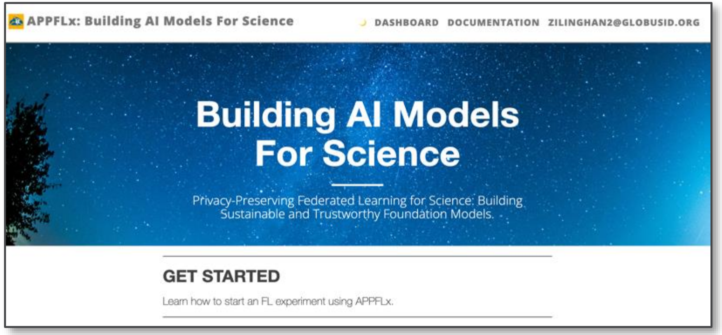
- Fully open-source
- Welcome issues
- Welcome contributions
- …

APPFL FRAMEWORK



[Documentation]

Detailed Documentation 📜
- Installation
- Launching FL experiments
- Advanced Developer Guides
- …



[service.appfl.ai]

APPFL-based Service Platform 🚀
- Fully based on APPFL
- User-friendly for domain experts
- Comprehensive report generation
- …

APPFL+MONAI

MONAI FL module provides a MonaiAlgo class, which provides train, evaluate, and get_weights functions to enable federated learning by leveraging a collection of medical imaging models available in MONAI bundles.

MONAI Bundle Reference Implementations

```
class monai.fl.client.MonaiAlgo(bundle_root, local_epochs=1,
send_weight_diff=True, config_train_filename='configs/train.json',
train_kwargs=None, config_evaluate_filename='default', eval_kwargs=None,
config_filters_filename=None, disable_ckpt_loading=True,
best_model_filepath='models/model.pt',
final_model_filepath='models/model_final.pt', save_dict_key='model',
data_stats_transform_list=None, eval_workflow_name='train',
train_workflow=None, eval_workflow=None)
```

[source]

Implementation of `ClientAlgo` to allow federated learning with MONAI bundle configurations.

APPFL+MONAI

MONAI FL module provides a MonaiAlgo class, which provides train, evaluate, and get_weights functions to enable federated learning by leveraging a collection of medical imaging models available in MONAI bundles.

All Models

Brats mri axial slices generative diffusion
MONAI team
A generative model for creating 2D brain MRI axial slices from Gaussian noise based on BraTS dataset
Model Details

Brats mri generative diffusion
MONAI team
A generative model for creating 3D brain MRI from Gaussian noise based on BraTS dataset
Model Details

Brats mri segmentation
MONAI team
A pre-trained model for volumetric (3D) segmentation of brain tumor subregions from multimodal MRIs based on BraTS 2018 data
Model Details

Breast density classification
Center for Augmented Intelligence in Imaging, Mayo Clinic Florida
A pre-trained model for classifying breast images (mammograms)
Model Details

Endoscopic inbody classification
NVIDIA DLMED team
A pre-trained binary classification model for endoscopic inbody classification task
Model Details

Endoscopic tool segmentation
NVIDIA DLMED team
A pre-trained binary segmentation model for endoscopic tool segmentation
Model Details

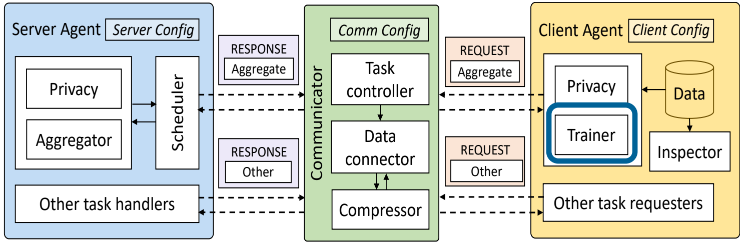
Lung nodule ct detection
MONAI team
A pre-trained model for volumetric (3D) detection of the lung lesion from CT image on LUNA16 dataset
Model Details

Mednist gan
MONAI Team
This example of a GAN generator produces hand xray images like those in the MedNIST dataset
Model Details

Mednist reg
MONAI team
This is an example of a ResNet and spatial transformer for hand xray image registration
Model Details

MONAI Model Zoos
https://monai.io/model-zoo.html

APPFL+MONAI



- We leverage the MonaiAlgo class to define a MonaiTrainer within APPFL's Trainer module to train models using the MONAI bundles.
- Thanks to the awesome interfaces provided by the MonaiAlgo, it only takes ~100 lines of code to use all MONAI bundles in APPFL.
- All MONAI bundles can utilize all APPFL's features and solutions to various FL challenges to federate the model training.

```
class MonaiTrainer(BaseTrainer):
    def __init__(...):
        ...
        self.monai_algo = MonaiAlgo(...)
        self.monai_algo.initialize(...)
    def get_parameters(self):
        ...
        self.monai_algo.get_weights(...)
    def load_parameters(self, params):
        ...
    def train(self, **kwargs):
        ...
        self.monai_algo.evaluate(...)
        self.monai_algo.train(...)
        self.monai_algo.evaluate(...)
        ...
```

APPFL+MONAI

Example: Running APPFL using MONAI Bundle

This tutorial describes how to run federated learning experiments via APPFL using MONAI Bundles to leverage a collection of medical imaging models available in MONAI model zoo. This examples shows how to use MONAI Bundle to do 3D spleen CT segmentation using gRPC with two clients.

> ✏ Note
>
> **Acknowledgement**: We extend our gratitude to the MONAI and NVFlare teams for their invaluable support and information throughout this tutorial. Specifically, this tutorial refers to the NVFlare-MONAI integration tutorial.

> ✏ Note
>
> This tutorial is the beta version of the integration of MONAI Bundle with APPFL. The integration is still under active development.

Installation

User can install `appfl` and `monai` packages from `appfl`'s source code by running the following commands:

```
git clone --single-branch --branch main https://github.com/APPFL/APPFL.git
cd APPFL
pip install -e ".[monai,examples]"
```

```
appfl: ✅ [2025-01-19 04:04:05,174 server]: Logging to ./output/result_Server_2025-01-19-04-04-05.txt
appfl: ✅ [2025-01-19 04:07:00,973 server]: Received GetConfiguration request from client Client1
appfl: ✅ [2025-01-19 04:07:39,732 server]: Received UpdateGlobalModel request from client Client1
appfl: ✅ [2025-01-19 04:07:39,741 server]: Received the following meta data from Client1:
{'round': 1,
'val_accuracy': 0.9534343488656791,
'val_accuracy_before_train': 0.7170387863353559,
'val_mean_dice': 0.06496836245059967,
'val_mean_dice_before_train': 0.03413229435682297}
appfl: ✅ [2025-01-19 04:08:02,911 server]: Received GetConfiguration request from client Client2
appfl: ✅ [2025-01-19 04:08:44,316 server]: Received UpdateGlobalModel request from client Client2
appfl: ✅ [2025-01-19 04:08:44,319 server]: Received the following meta data from Client2:
{'round': 1,
'val_accuracy': 0.9544978111412874,
'val_accuracy_before_train': 0.7170388106327907,
'val_mean_dice': 0.06501330435276031,
'val_mean_dice_before_train': 0.034132301807403564}
appfl: ✅ [2025-01-19 04:09:01,715 server]: Received UpdateGlobalModel request from client Client2
appfl: ✅ [2025-01-19 04:09:01,717 server]: Received the following meta data from Client2:
{'round': 2
'val_accuracy': 0.9604373494530939,
'val_accuracy_before_train': 0.9539599266781169,
'val_mean_dice': 0.06739335507154465,
'val_mean_dice_before_train': 0.06500281393527985}
```

https://appfl.ai/en/latest/tutorials/examples_monai.html

USEFUL QR CODES

- https://github.com/APPFL/APPFL
- Give us a star ⭐ if you think our framework could be useful for your future research 🔬

- Join our Discord channel for further discussions

REFERENCE

- [1] Ogier du Terrail, Jean, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold et al. "Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings." *Advances in Neural Information Processing Systems* 35 (2022): 5315-5334.
- [2] Zhang, Jiale, Junjun Chen, Di Wu, Bing Chen, and Shui Yu. "Poisoning attack in federated learning using generative adversarial nets." In *2019 18th IEEE international conference on trust, security and privacy in computing and communications/13th IEEE international conference on big data science and engineering (TrustCom/BigDataSE)*, pp. 374-380. IEEE, 2019.
- [3] Kaissis, Georgios, Alexander Ziller, Jonathan Passerat-Palmbach, Théo Ryffel, Dmitrii Usynin, Andrew Trask, Ionésio Lima Jr et al. "End-to-end privacy preserving deep learning on multi-institutional medical imaging." *Nature Machine Intelligence* 3, no. 6 (2021): 473-484.
- [4] Hsu, Tzu-Ming Harry, Hang Qi, and Matthew Brown. "Measuring the effects of non-identical data distribution for federated visual classification." *arXiv preprint arXiv:1909.06335* (2019).
- [5] Reddi, Sashank, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. "Adaptive federated optimization." *arXiv preprint arXiv:2003.00295* (2020).
- [6] Karimireddy, Sai Praneeth, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. "Scaffold: Stochastic controlled averaging for federated learning." In *International conference on machine learning*, pp. 5132-5143. PMLR, 2020.
- [7] Li, Tian, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. "Federated optimization in heterogeneous networks." *Proceedings of Machine learning and systems* 2 (2020): 429-450.

REFERENCE

- [8] Hiniduma, Kaveen, Suren Byna, Jean Luca Bez, and Ravi Madduri. "Ai data readiness inspector (aidrin) for quantitative assessment of data readiness for ai." In *Proceedings of the 36th International Conference on Scientific and Statistical Database Management*, pp. 1-12. 2024.
- [9] Xie, Cong, Sanmi Koyejo, and Indranil Gupta. "Asynchronous federated optimization." *arXiv preprint arXiv:1903.03934* (2019).
- [10] Nguyen, John, Kshitiz Malik, Hongyuan Zhan, Ashkan Yousefpour, Mike Rabbat, Mani Malek, and Dzmitry Huba. "Federated learning with buffered asynchronous aggregation." In *International Conference on Artificial Intelligence and Statistics*, pp. 3581-3607. PMLR, 2022.
- [11] Li, Zilinghan, Pranshu Chaturvedi, Shilan He, Han Chen, Gagandeep Singh, Volodymyr Kindratenko, Eliu A. Huerta, Kibaek Kim, and Ravi Madduri. "FedCompass: efficient cross-silo federated learning on heterogeneous client devices using a computing power aware scheduler." *arXiv preprint arXiv:2309.14675* (2023).
- [12] Li, Zilinghan, Shilan He, Ze Yang, Minseok Ryu, Kibaek Kim, and Ravi Madduri. "Advances in appfl: A comprehensive and extensible federated learning framework." *arXiv preprint arXiv:2409.11585* (2024).
- [13] Hoang, Trung-Hieu, Jordan Fuhrman, Ravi Madduri, Miao Li, Pranshu Chaturvedi, Zilinghan Li, Kibaek Kim et al. "Enabling end-to-end secure federated learning in biomedical research on heterogeneous computing environments with APPFLx." *arXiv preprint arXiv:2312.08701* (2023).
- [14] Ryu, Minseok, Youngdae Kim, Kibaek Kim, and Ravi K. Madduri. "APPFL: open-source software framework for privacy-preserving federated learning." In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 1074-1083. IEEE, 2022.

THANK YOU