Introduction to NVIDIA Federated Learning: Concepts, Technology, and Use Cases

Holger Roth, Principal Federated Learning Scientist, NVIDIA

NVIDIA FLARE WEBINAR – Q1, FEB 19TH, 2025

Agenda

- **What is Federated Learning?**
- NVIDIA Key Technologies for Federated Learning
- Real-world Use cases of Federated Learning
- Getting Started with NVIDIA FLARE
- Research: Addressing Key challenges in Federated Learning
- Summary & Announcements

BUILDING ROBUST, GENERALIZABLE AI MODELS IS HARD

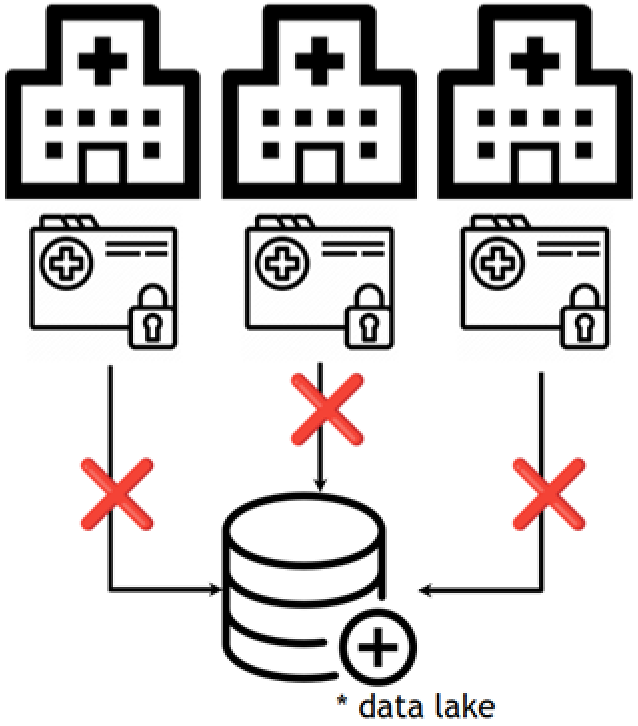
DATA DIVERSITY
Rare Diseases | Quantity 10^1-10^3

DATA PRIVACY
Patient Privacy | Data Governance

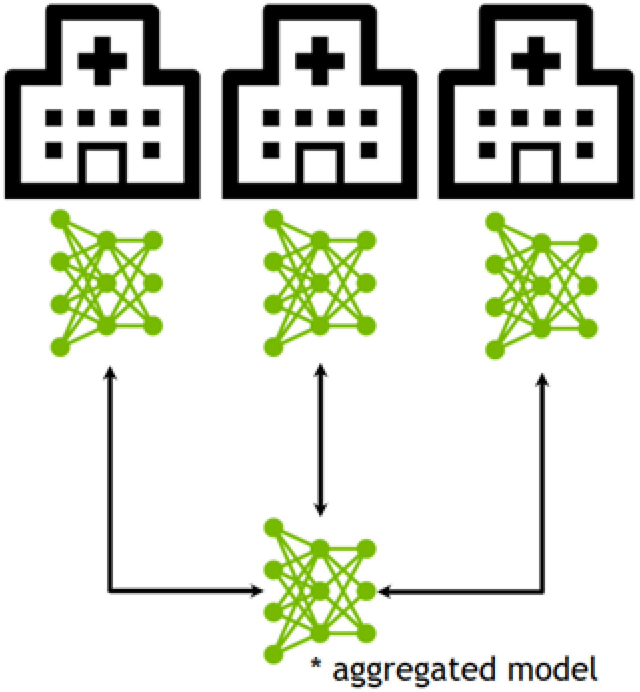
However

- Private data can often not be shared (regulations, complex bureaucracy)
- Data annotation is costly; Data is an asset

FEDERATED LEARNING

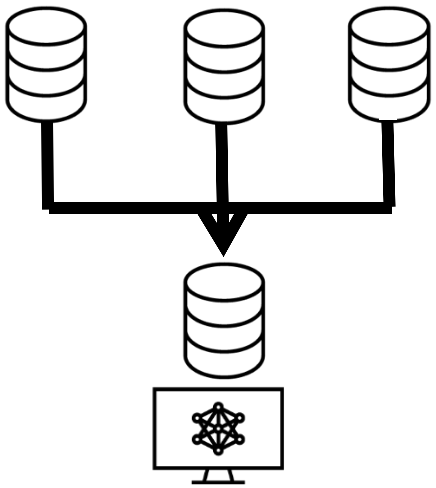


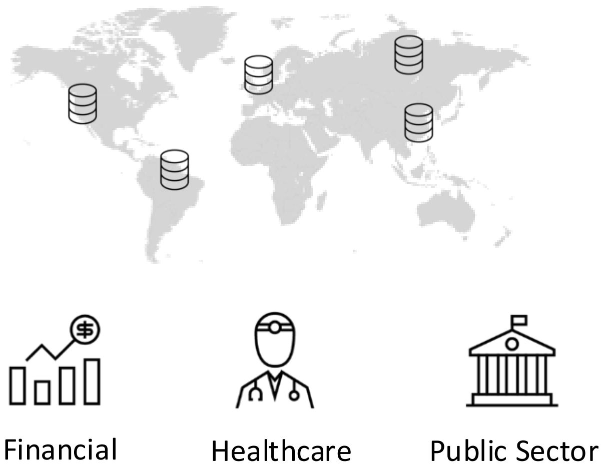
- Data cannot be centralized in many tasks
- **Share models, not data!**

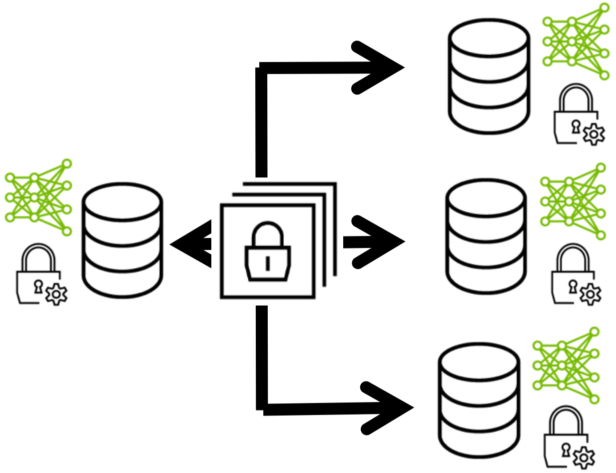


Federated Computing – Removing Data Silos

Avoid Data Copy | Regulatory Compliance | Prevent Private Data Leak







No Data Copy

Centrally aggregating the data is not possible or practical

Compliance

Data sovereignty restrictions and industry regulations

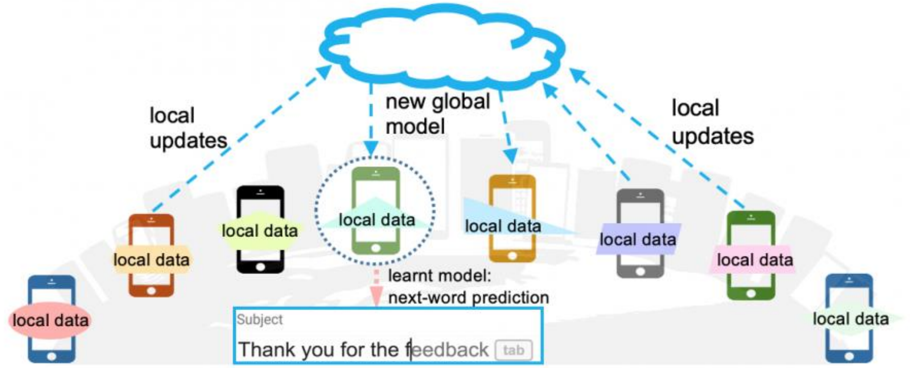
Privacy Enhancing Technology

Multiple layers of security features incl. Homomorphic Encryption, Differential Privacy, & Confidential Computing

CROSS-DEVICE VS CROSS-SILO

Cross-device:

- Large number of user (millions), unknown to each other
- Intermittent user/data availability (select from a pool) - user/data for each FL round could be different
- Massively distributed
- Lower computation power
- Examples: mobile phones, autonomous cars, robots, etc.



Cross-silo:

- Usually comes from a trusted collaboration, small number (10s-100s)
- Relatively stable user/data and connection
- High computation power, usually dedicated
- Regional/global scale
- Examples: hospitals, financial institutions, enterprises, data centers, etc.

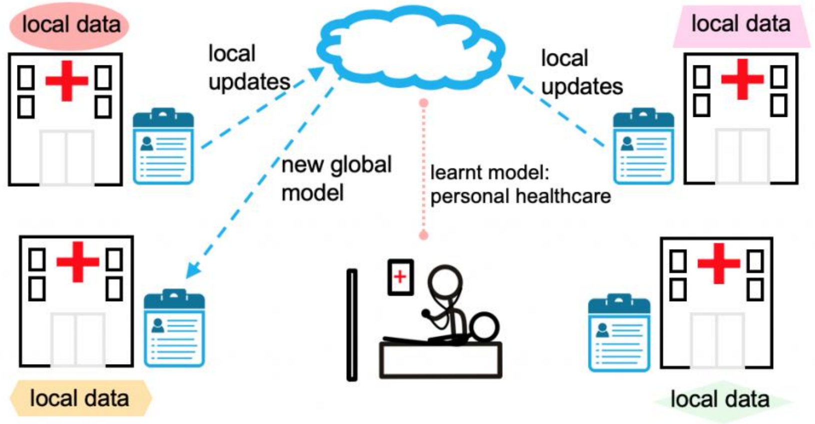
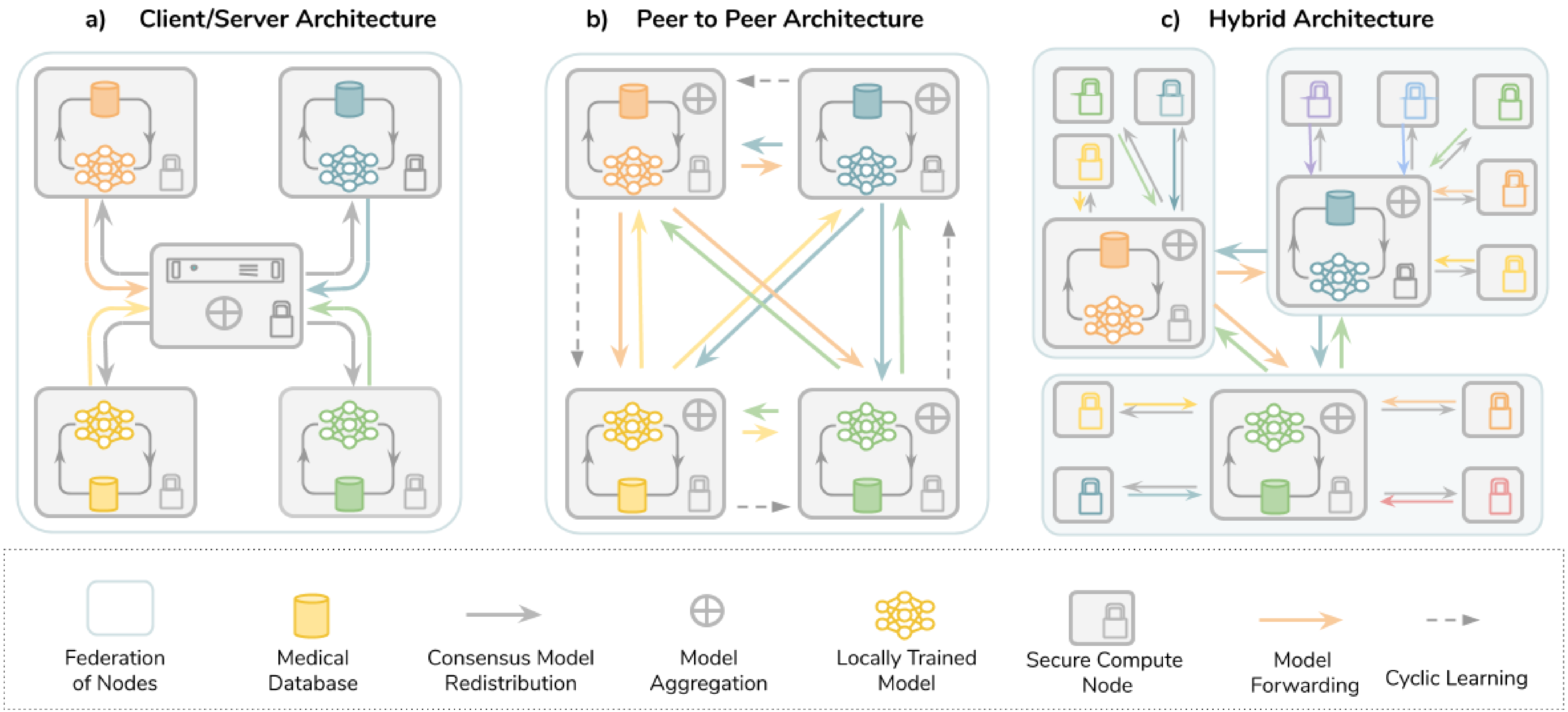


Figure Source: https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/

FEDERATED LEARNING ARCHITECTURES

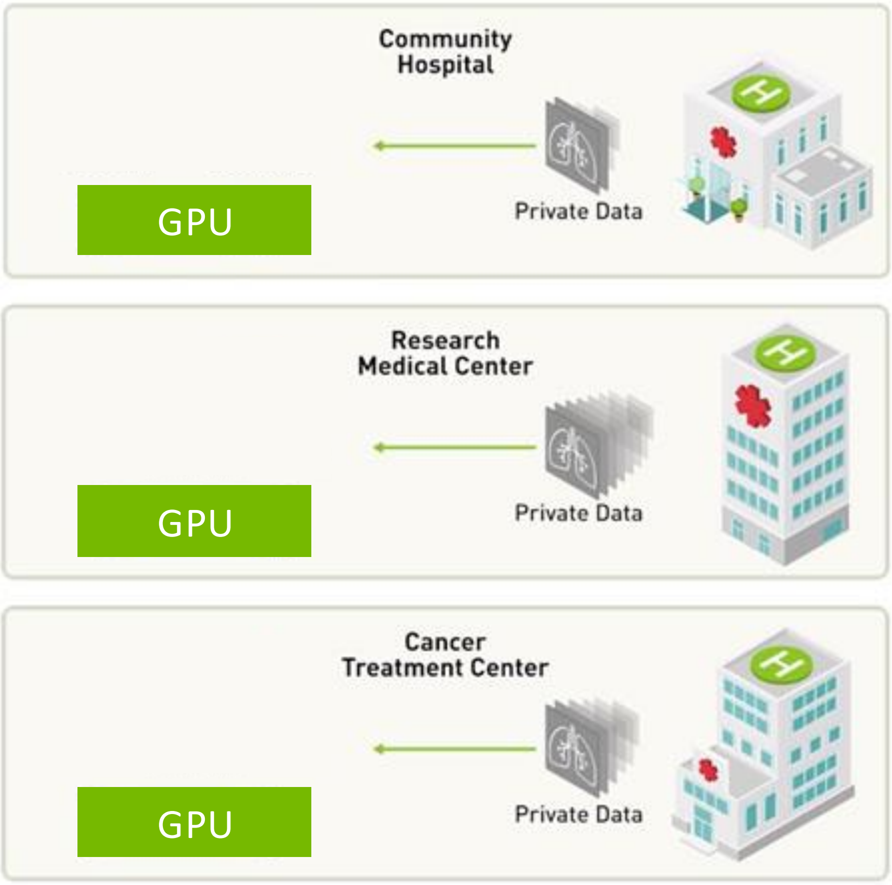


https://arxiv.org/abs/2003.08119

Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B., Maier-Hein, K., Ourselin, S., Sheller, M., Summers, R. M., Trask, A., Xu, D., Baust, M. & Cardoso, M. J.; [The Future of Digital Health with Federated Learning](). NPJ digital medicine (2020).

SERVER-CLIENT FEDERATED LEARNING

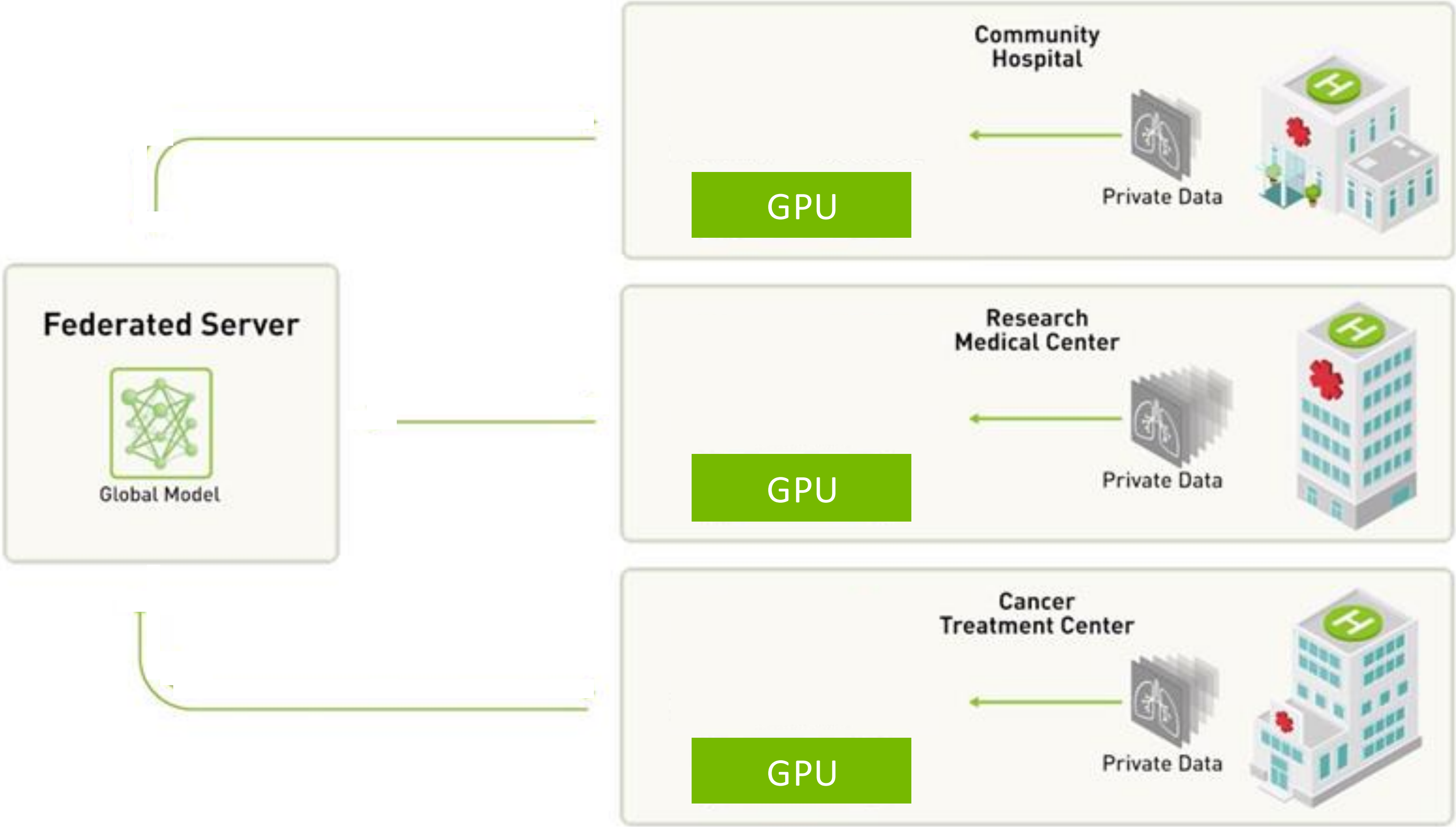
Changing the way AI algorithms are trained



https://developer.nvidia.com/clara-medical-imaging

SERVER-CLIENT FEDERATED LEARNING

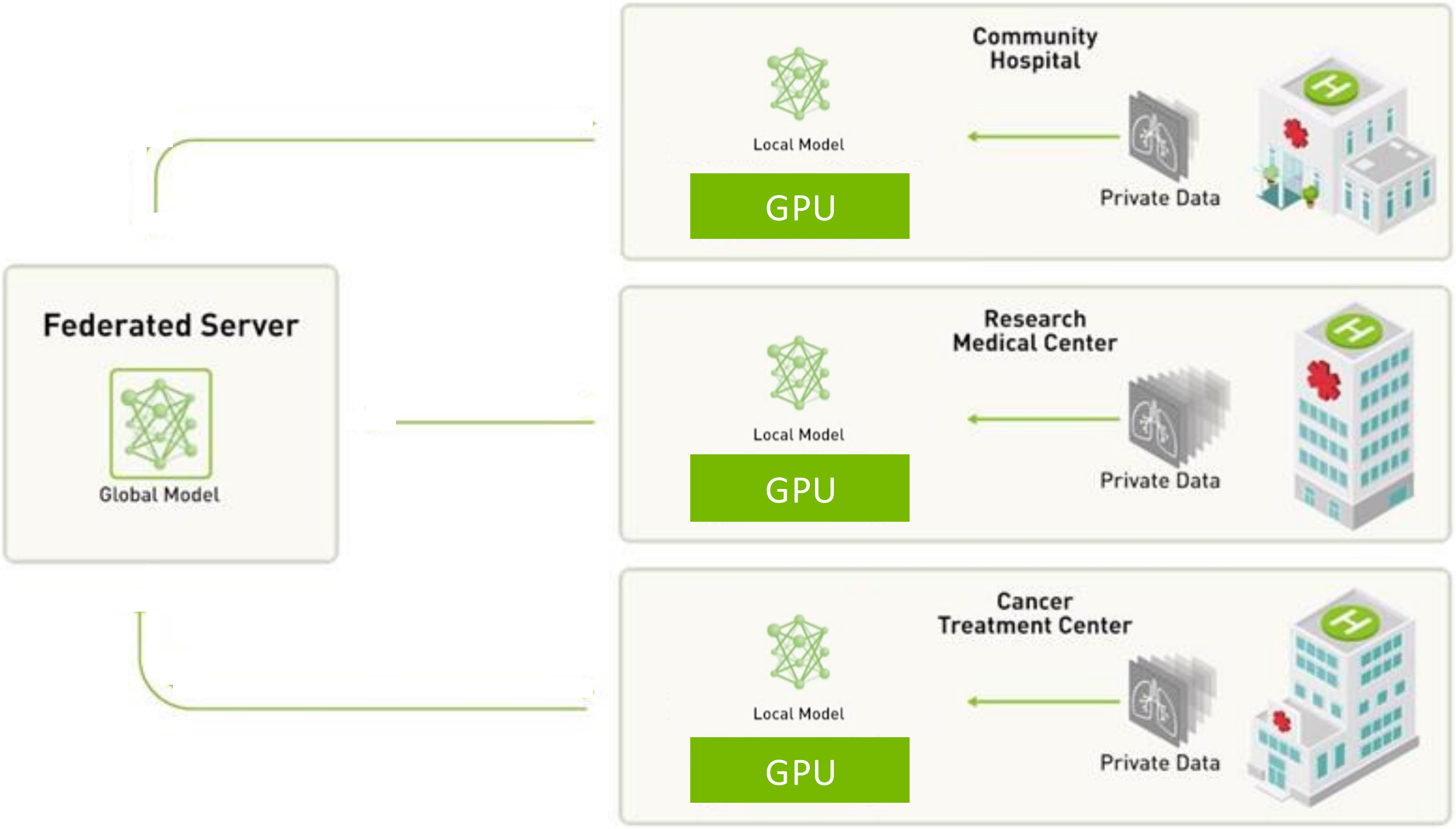
Changing the way AI algorithms are trained



https://developer.nvidia.com/clara-medical-imaging

SERVER-CLIENT FEDERATED LEARNING

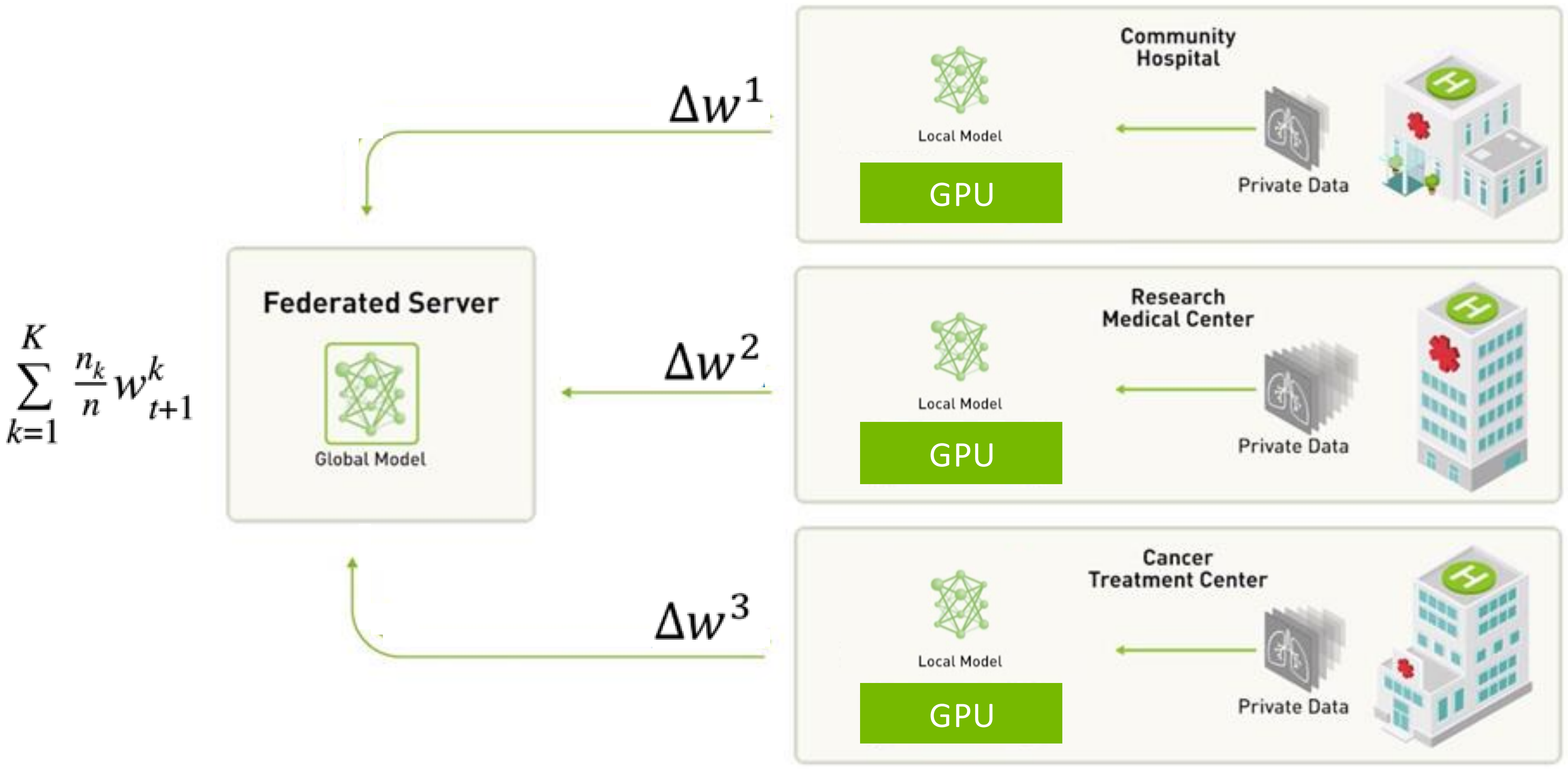
Changing the way AI algorithms are trained



https://developer.nvidia.com/clara-medical-imaging

SERVER-CLIENT FEDERATED LEARNING

Changing the way AI algorithms are trained

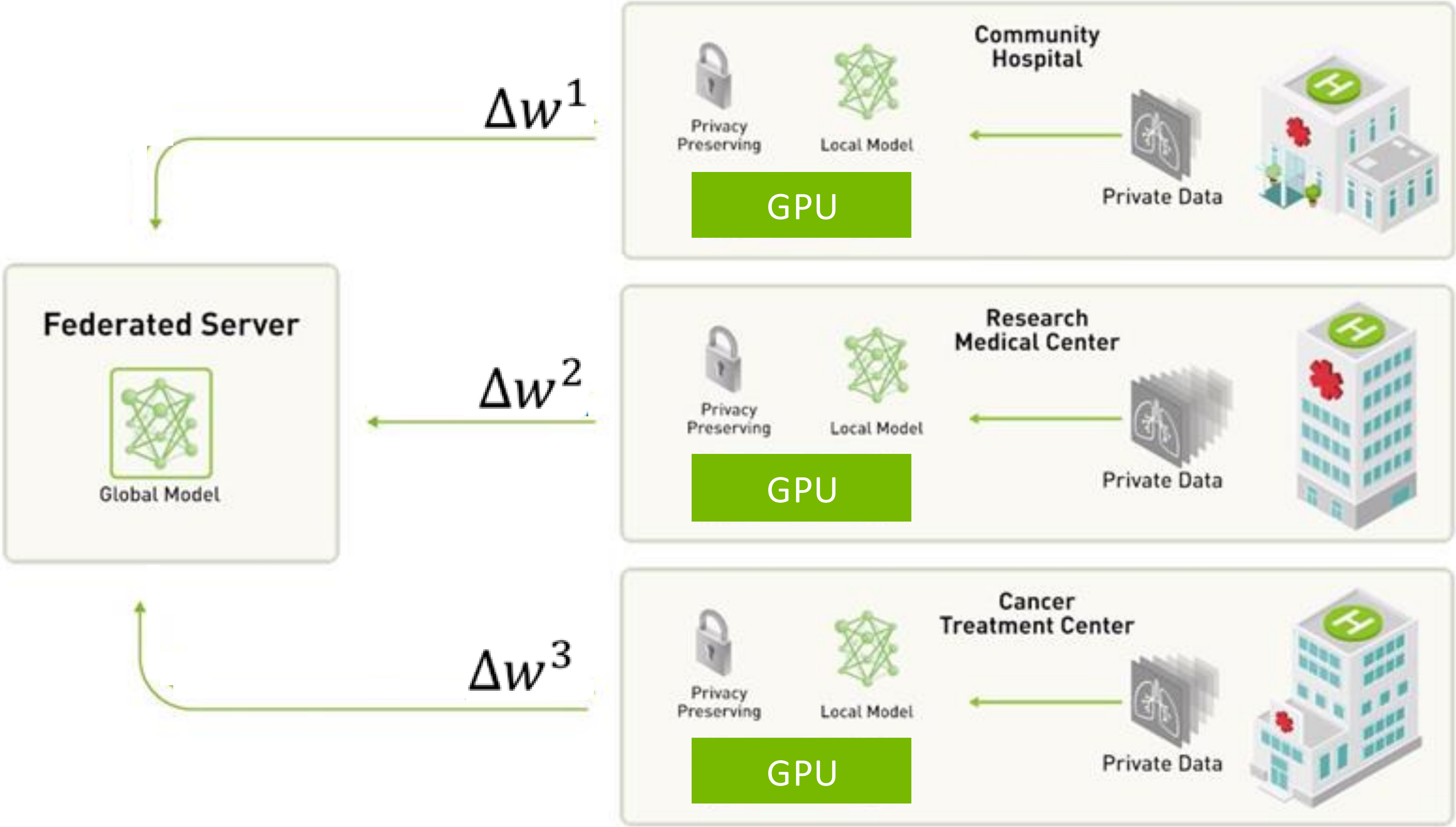


https://developer.nvidia.com/clara-medical-imaging

SERVER-CLIENT FEDERATED LEARNING

Changing the way AI algorithms are trained

$$\sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^{k}$$



https://developer.nvidia.com/clara-medical-imaging

Agenda

- What is Federated Learning?
- **NVIDIA Key Technologies for Federated Learning**
- Real-world Use cases of Federated Learning
- Getting Started with NVIDIA FLARE
- Research: Addressing Key challenges in Federated Learning
- Summary & Announcements

NVIDIA FLARE is Enabling FL in Many Industries

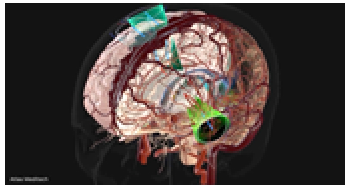
Healthcare – medical imaging, oncology

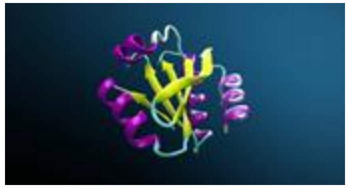
Financial Service – fraud detection

Pharmaceutical Industry – drug discovery, predictive model

Energy System – distributed energy sources

Automotive Industry – autonomous vehicle, object detection & classification

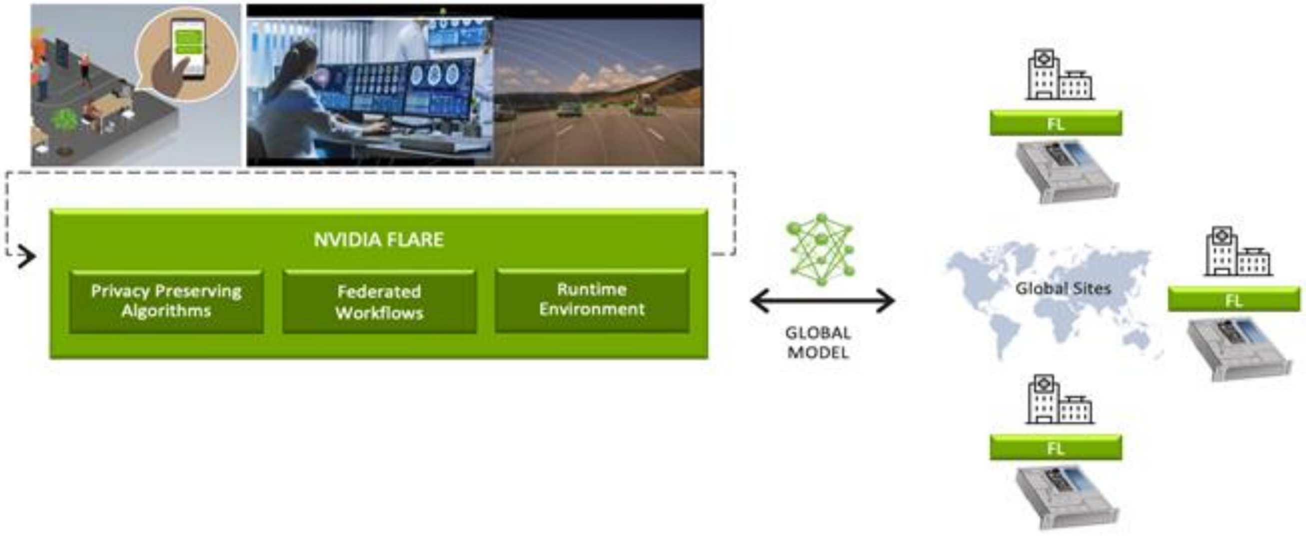




source: https://arxiv.org/pdf/2208.10941.pdf

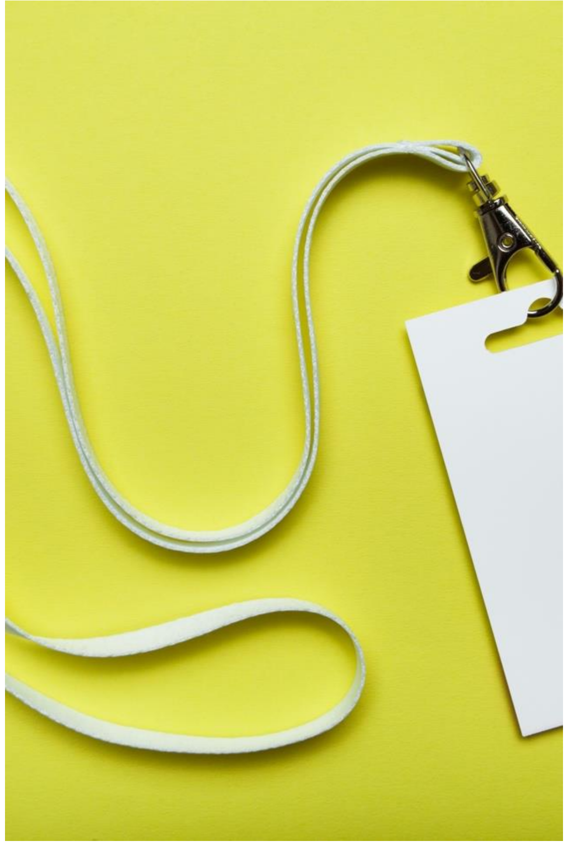
From Research To Production

simulation | easy transition from ML to FL | enterprise security | cross-cloud deployment



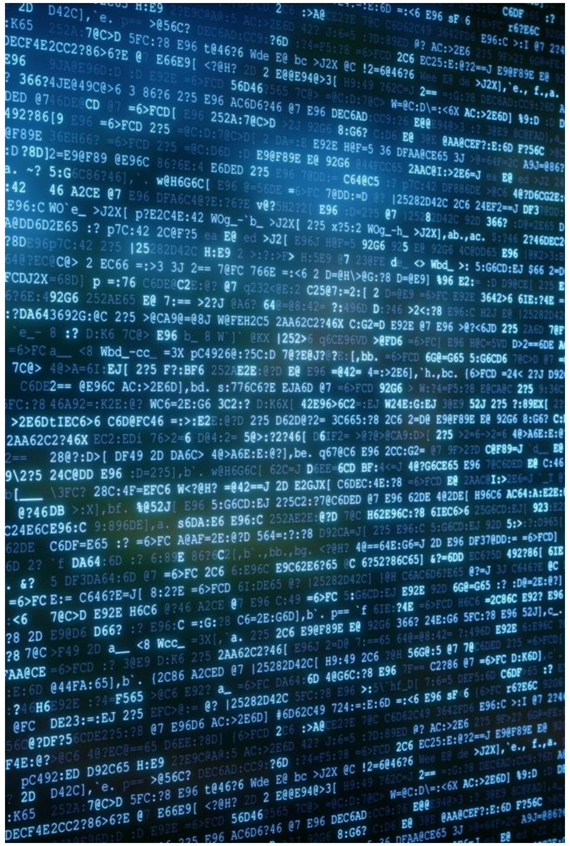
NVIDIA FLARE Security and Data Privacy

Defense in Depth approach to protecting data privacy and model IP



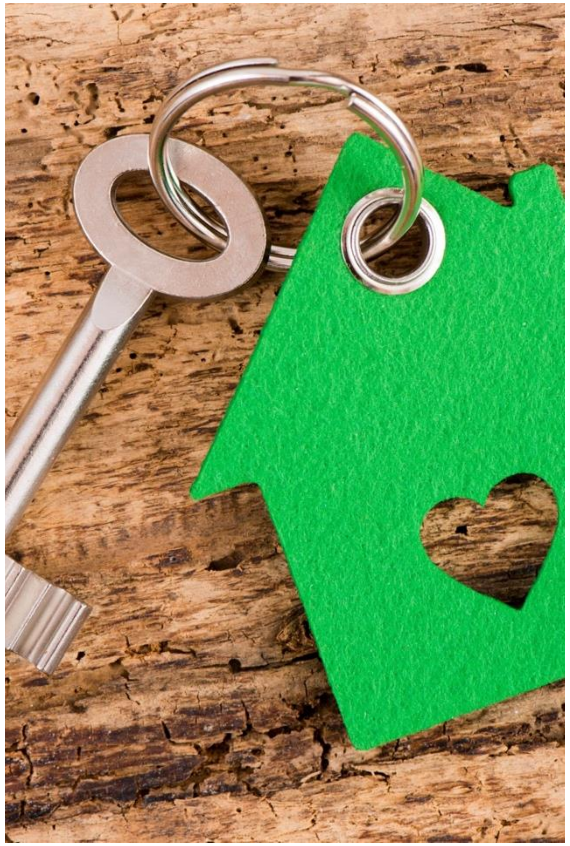
User Identity Verification

Certificate and derived token authentication



Data Encryption in Transit

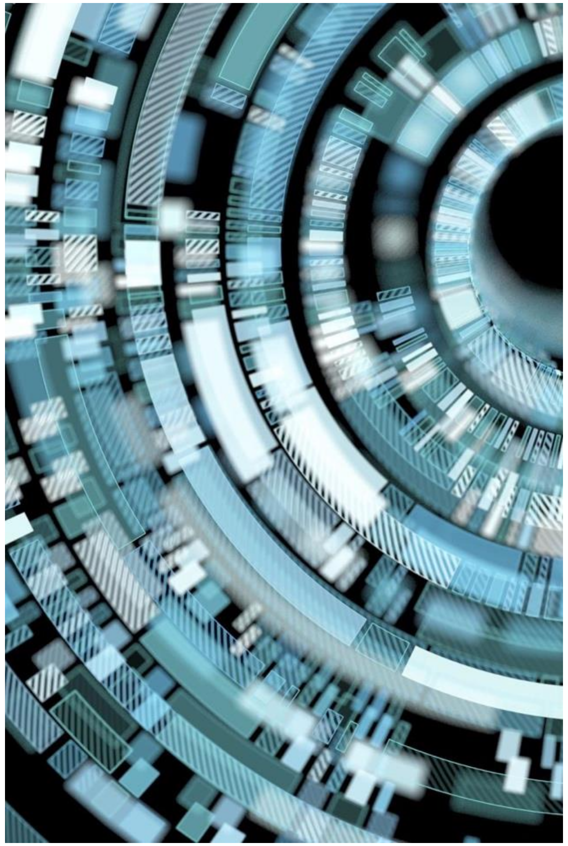
Server-Client communication encrypted



User Defined Security Policy

Site-Specific authentication

Job authorization



Privacy Preserving Algorithm

Differential Privacy

Homomorphic Encryption

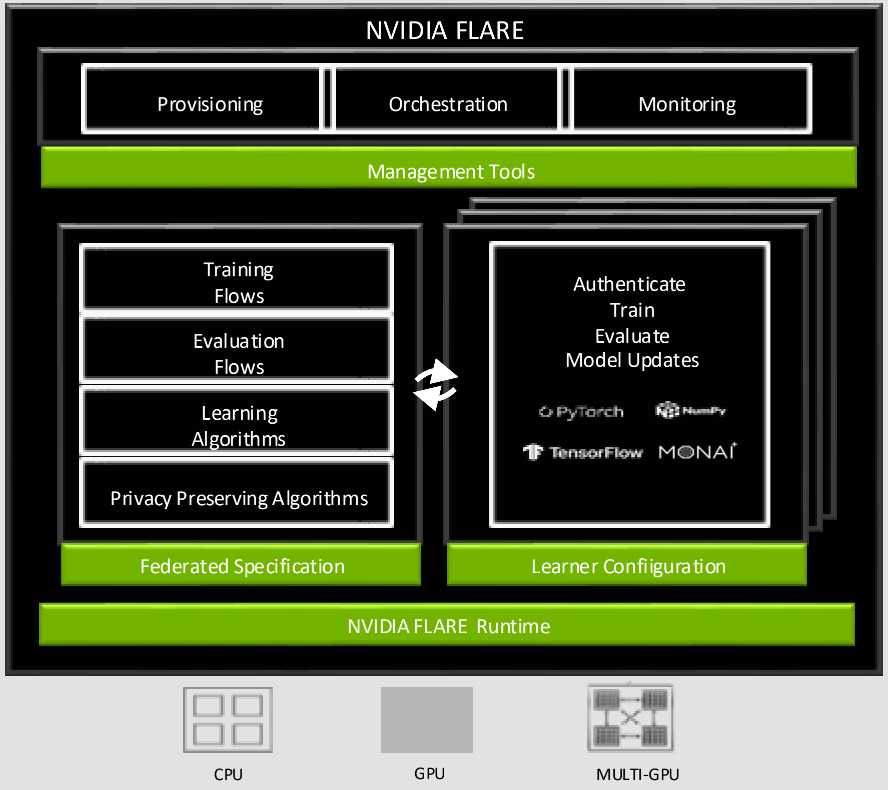
Confidential Computing

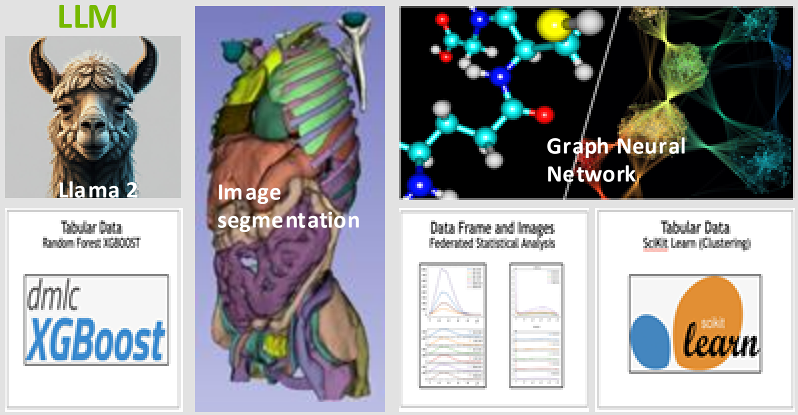
NVIDIA FLARE

Open-Source, Enterprise Federated Learning & Compute Framework

- **Apache License 2.0** to catalyze FL research & development
- **Designed for production**, not just for research
- **Enables cross-internet**, distributed, multi-party collaborative Learning
- **Production scalability** with high availability and **multi-task** execution
- **Easy to convert** existing ML/DL workflows to a Federated paradigm with few lines of code changes
- **LLM streaming, LLM fine tuning**
- **Framework, model, domain and task agnostic**
- **Privacy Preserving** Technologies
  - Homomorphic Encryption (HE), Differential Privacy (DP)
  - Multi-party computing (Private Set intersection, PSI)
  - Confidential Computing (CC)
- **Flexible communication patterns:** server-centric, p2p, federated averaging, split learning, swarm learning, etc.
- **Layered, pluggable, customizable** federated compute architecture
- Secure Provisioning, Orchestration & Monitoring

GitHub: https://github.com/nvidia/nvFlare
Web: https://nvidia.github.io/NVFlare



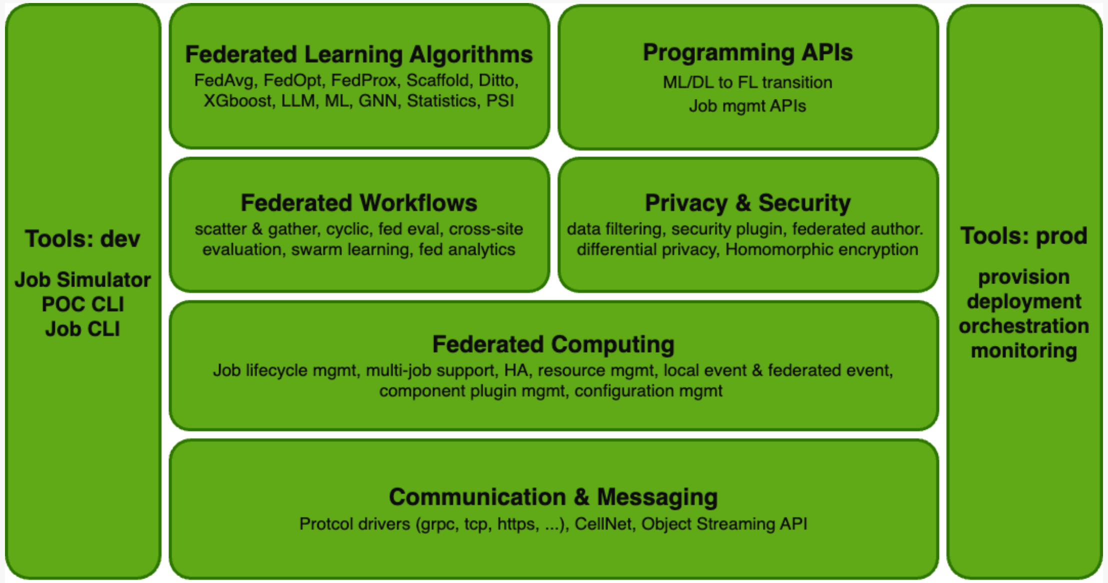


.Framework agnostic | Model agnostic | Domain agnostic | Task agnostic

NVIDIA FLARE Architecture

Federated Computing Engine

- **Layered, Pluggable Open Architecture**
  - Each layer's component are composable and pluggable
- **Network: Communication & Messaging layer**
  - Drivers ➔ gRPC, http + websocket, TCP, any plugin driver
  - CellNet: logical end point-to-point (cell to cell) network
  - Message: reliable streaming message
- **Federated Computing Layer**
  - Resource-based job scheduling, job monitoring, concurrent job lifecycle management, High-availability management
  - Plugin component management
  - Configuration management
  - Local event and federated event handling
- **Federated Workflow**
  - SAG, Cyclic, Cross-site Evaluation, split learning, Swarm Learning, Federated Analytics
- **Federated Learning Algorithms**
  - FedAvg, FedOpt, FedProx, Scalffold, Ditto, XGBoost, GNN, PSI, LLM (p-tuning, SFT, PEFT), KM, Scikit-Learn
- **Pythonic Programming APIs**
  - Client API, Controller API, Job Construction API, Job Monitoring API
- **Productivity & Deployment Tools:**
  - Simulator, provision, POC, Cloud deployment, preflight check, more



Use Cases and Integrations

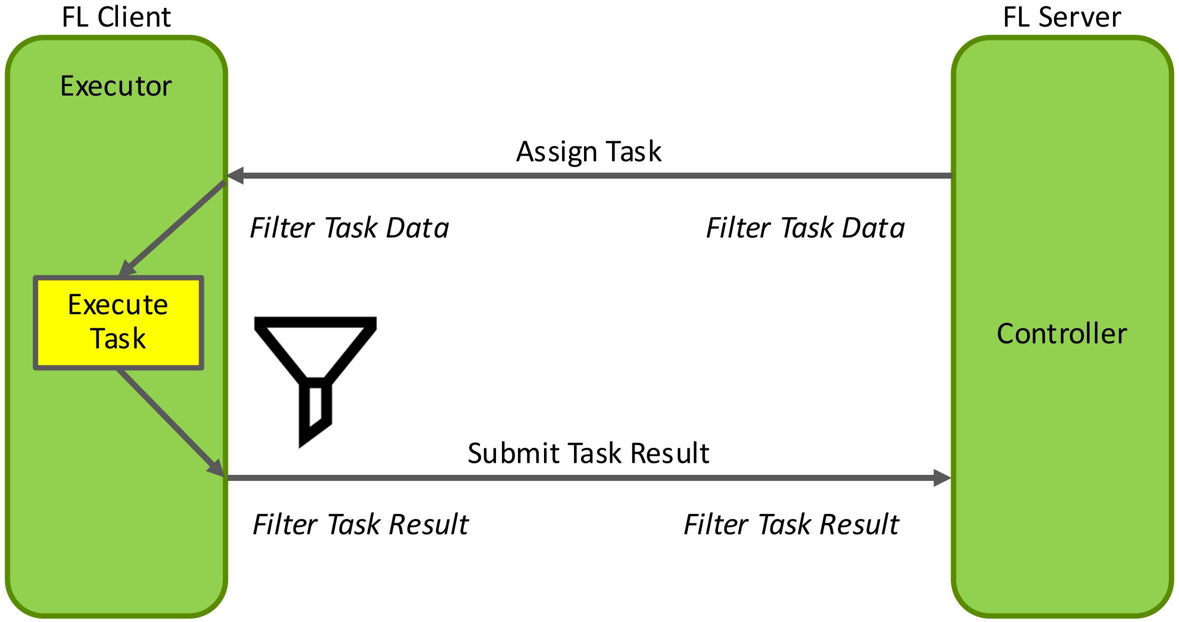
Applications and contributions

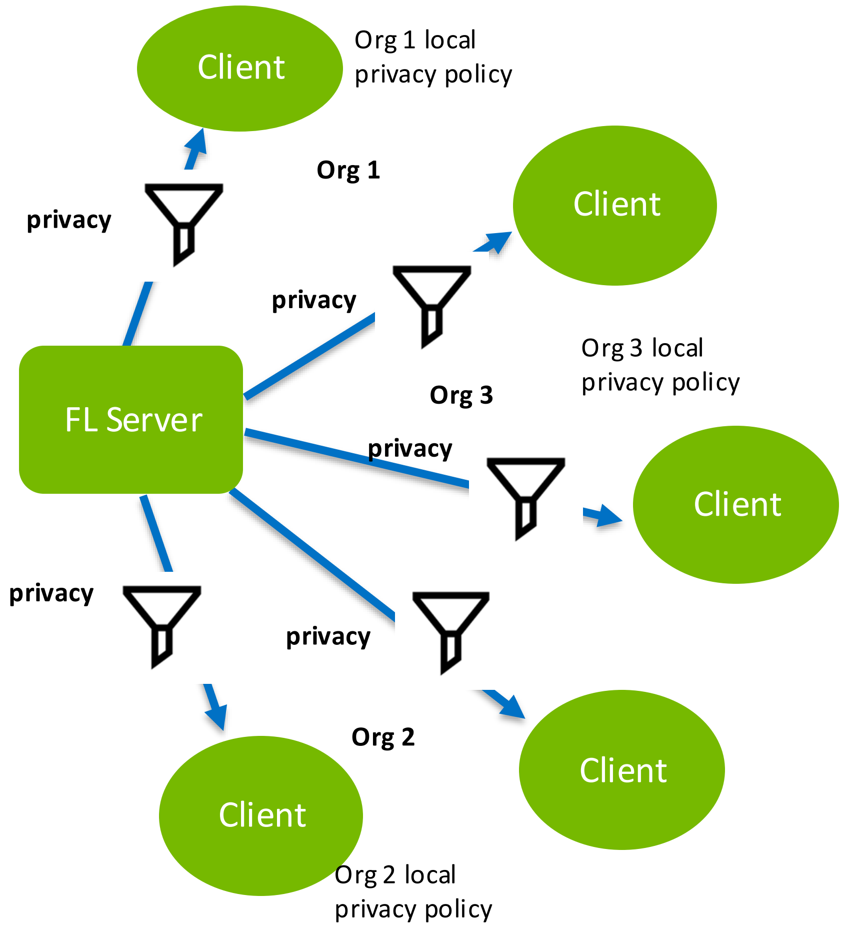
- Use cases & contributions
  - Cancer Detection with FLARE + MONAI - SIIM (society for imaging informatics and Medicine)
  - Renal cell carcinoma: multi-institute collaboration including Case Western, Mayo Clinic, UFL, Vanderbilt, etc.
  - Oncology for cancer patients: EU OPTIMA consortium
  - Cancer research: OHSU and NCI
  - Digital Pathology: Roche
  - Federated Learning Interoperability Platform: Kings College London
  - Cancer detection / classification: GE Healthcare
  - Supercomputing: Oak Ridge National Lab
  - Transportation: Cummins
  - LLM: Deloitte
  - NLP research: UMN
  - FL algorithm: UBC
- Integrations
  - Medical image analysis framework: MONAI
  - Generative AI platform: NeMo
  - FL framework: Flower

...many more

HIGH LEVEL ARCHITECTURE

Data privacy architecture





- **Privacy filter can depend on:**
  - Scope: any key-value pair such as datasets
  - Data kind: Weights, Weights Diff or Analytics data
  - Or any other data
- **Research develop privacy filter**
- **Organization set privacy policy:**
  - privacy budget, noise level as data privacy policy

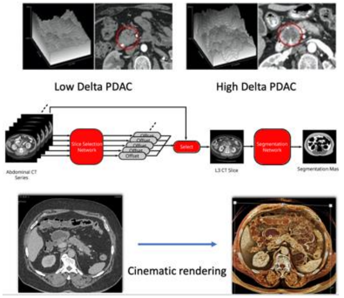
Agenda

- What is Federated Learning?
- NVIDIA Key Technologies for Federated Learning
- **Real-world Use cases of Federated Learning**
- Getting Started with NVIDIA FLARE
- Research: Addressing Key challenges in Federated Learning
- Summary & Announcements

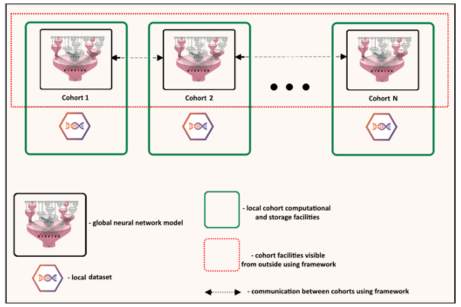
Federated learning In HEALTHCARE

Breaking Healthcare Data Siloes

EDRN
Early Detection of Pancreatic Cancer



ERASMUS GENNET
Genome Wide Association Study

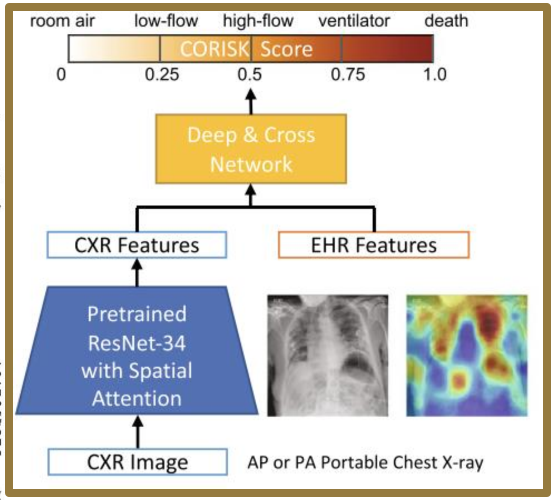


EXAM
COVID-19 Oxygen Requirement Prediction

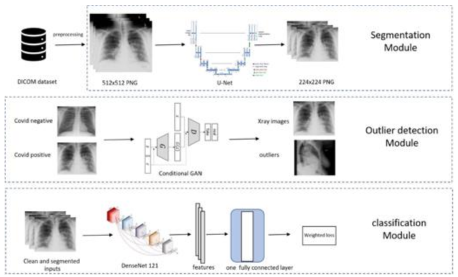

nature medicine ARTICLES
https://doi.org/10.1038/s41591-021-01506-3

Federated learning for predicting clinical outcomes in patients with COVID-19

NATURE MEDICINE | VOL 27 | OCTOBER 2021 | 1735–1743 | www.nature.com/naturemedicine



U MINNESOTA, FAIRVIEW
X-RAY Covid-19 Classification



MELLODDY
Multi-task Learning Chemical Assays

PHARMA PARTNERS

AMGEN, astellas, AstraZeneca, Bayer, Boehringer Ingelheim, gsk, janssen, Merck, NOVARTIS, SERVIER

PUBLIC PARTNERS

IKTOS, KU LEUVEN, loodse, NVIDIA, OWKIN, SUBSTRA

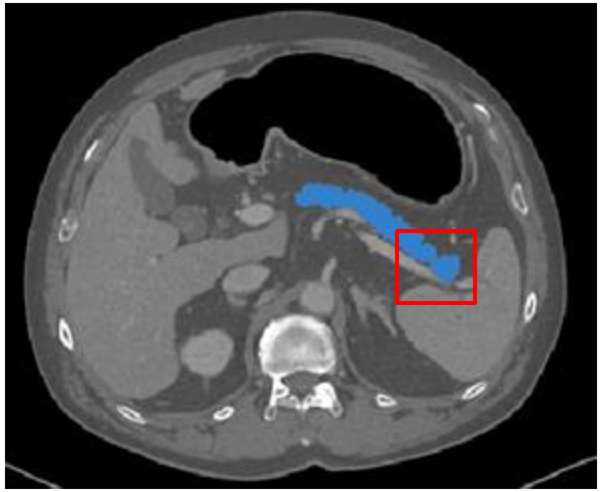
EXAM: https://www.nature.com/articles/s41591-021-01506-3

MELLODDY: https://chemrxiv.org/engage/chemrxiv/article-details/6345c0f91f323d61d7567624

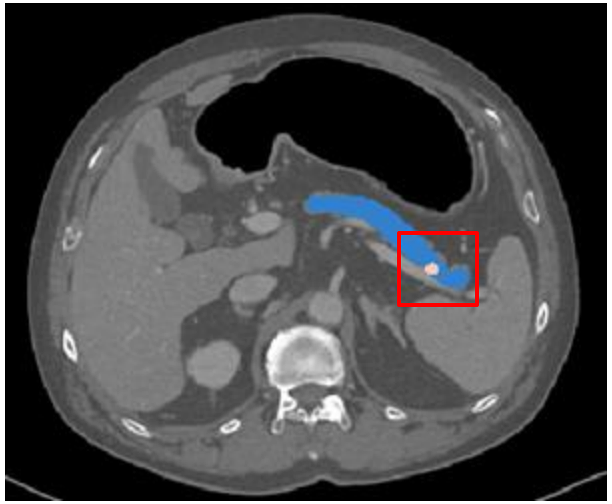
https://developer.download.nvidia.com/CLARA/Federated-Learning-Training-for-Healthcare-Using-NVIDIA-Clara.pdf

FEDERATED LEARNING

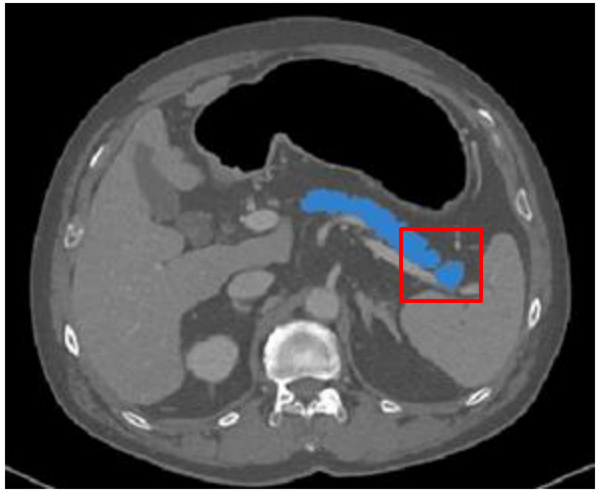
Global vs. local



Ground truth



C2 (local)
Trained on pancreas
& tumor



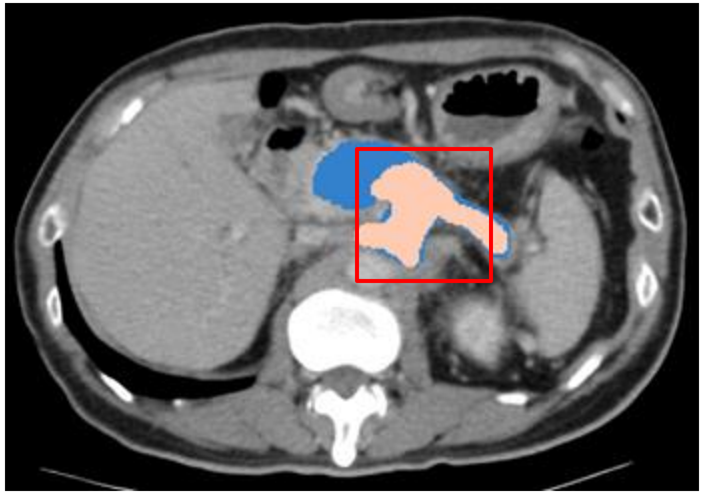
FL (global)

Dataset	Client 1	Client 2
Cases	420	486
Label	**2 class** (background, pancreas)	**3 classes** (background, pancreas, tumor)

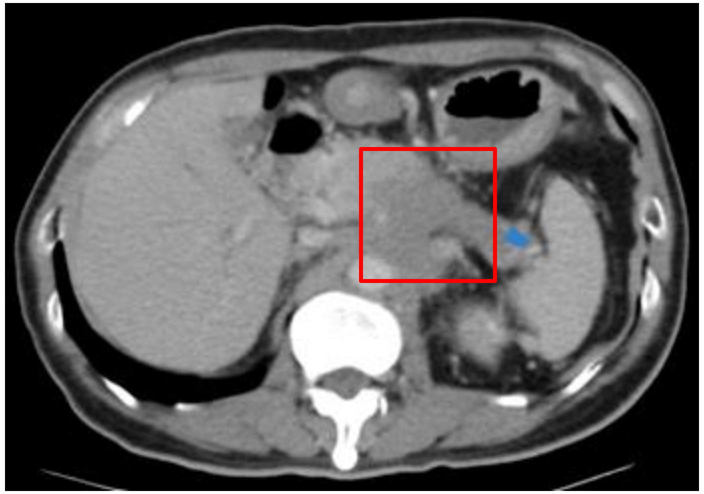
Wang et al. MICCAI 2020 – DCL workshop, https://arxiv.org/abs/2009.13148

FEDERATED LEARNING

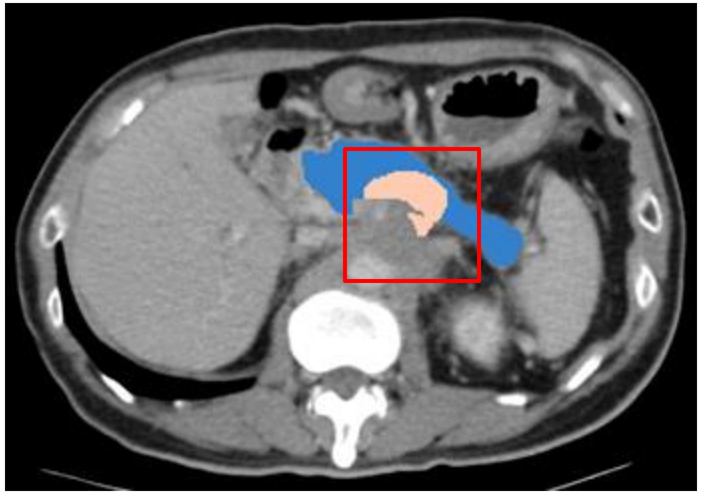
Global vs. local



Ground truth



C1 (local)
Trained on healthy pancreas



FL (global)

Dataset	Client 1	Client 2
Cases	420	486
Label	**2 class** (background, pancreas)	**3 classes** (background, pancreas, tumor)

Wang et al. MICCAI 2020 – DCL workshop, https://arxiv.org/abs/2009.13148

Dayan et al. Nature Medicine, Sep 15, 2021: https://www.nature.com/articles/s41591-021-01506-3

nature medicine

ARTICLES

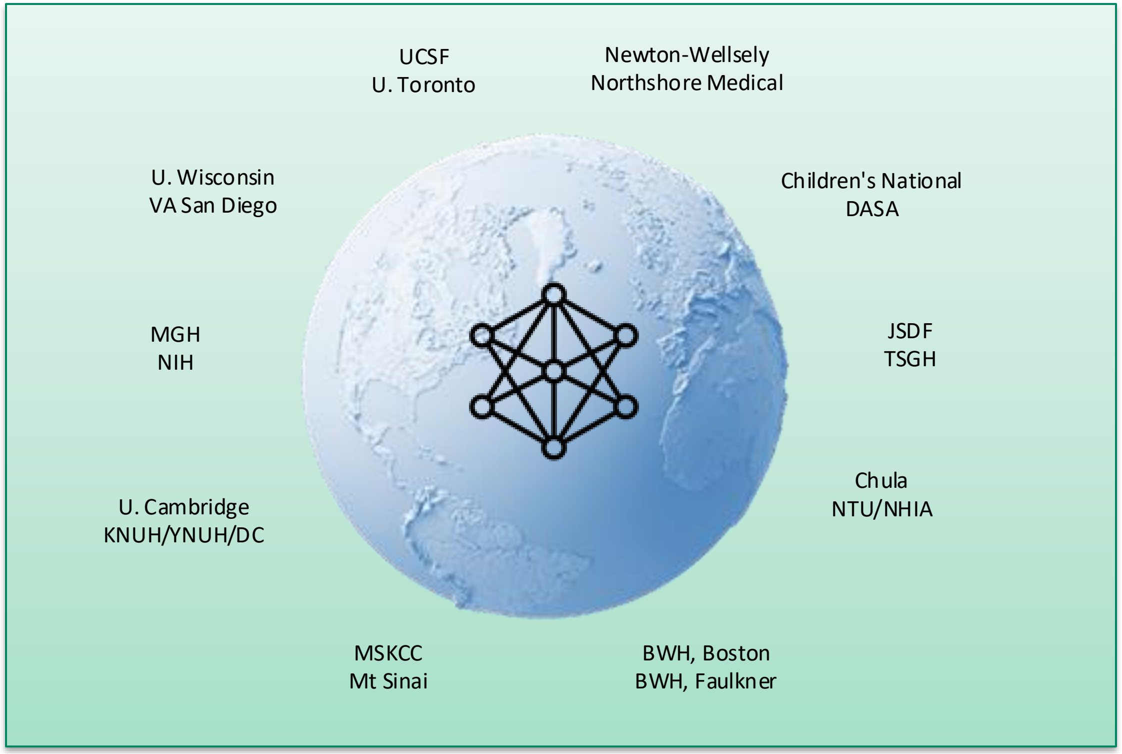
https://doi.org/10.1038/s41591-021-01506-3

Check for updates

Federated learning for predicting clinical outcomes in patients with COVID-19

Ittai Dayan[1,56], Holger R. Roth[2,56], Aoxiao Zhong[3,4,56], Ahmed Harouni[2], Amilcare Gentili[5], Anas Z. Abidin[2], Andrew Liu[2], Anthony Beardsworth Costa[6], Bradford J. Wood[7,8], Chien-Sung Tsai[9], Chih-Hung Wang[10,11], Chun-Nan Hsu[12], C. K. Lee[2], Peiying Ruan[2], Daguang Xu[2], Dufan Wu[3], Eddie Huang[2], Felipe Campos Kitamura[13], Griffin Lacey[2], Gustavo César de Antônio Corradi[13], Gustavo Nino[14], Hao-Hsin Shin[15], Hirofumi Obinata[16], Hui Ren[3], Jason C. Crane[17], Jesse Tetreault[2], Jiahui Guan[2], John W. Garrett[18], Joshua D. Kaggie[19], Jung Gil Park[20], Keith Dreyer[1,21], Krishna Juluru[15], Kristopher Kersten[2], Marcio Aloisio Bezerra Cavalcanti Rockenbach[21], Marius George Linguraru[22,23], Masoom A. Haider[24,25], Meena AbdelMaseeh[25], Nicola Rieke[2], Pablo F. Damasceno[17], Pedro Mario Cruz e Silva[2], Pochuan Wang[26,27], Sheng Xu[7,8], Shuichi Kawano[16], Sira Sriswasdi[28,29], Soo Young Park[30], Thomas M. Grist[31], Varun Buch[21], Watsamon Jantarabenjakul[32,33], Weichung Wang[26,27], Won Young Tak[30], Xiang Li[3], Xihong Lin[34], Young Joon Kwon[6], Abood Quraini[2], Andrew Feng[2], Andrew N. Priest[35], Baris Turkbey[8,36], Benjamin Glicksberg[37], Bernardo Bizzo[21], Byung Seok Kim[38], Carlos Tor-Díez[22], Chia-Cheng Lee[39], Chia-Jung Hsu[39], Chin Lin[40,41,42], Chiu-Ling Lai[43], Christopher P. Hess[17], Colin Compas[2], Deepeksha Bhatia[2], Eric K. Oermann[44], Evan Leibovitz[21], Hisashi Sasaki[16], Hitoshi Mori[16], Isaac Yang[2], Jae Ho Sohn[17], Krishna Nand Keshava Murthy[15], Li-Chen Fu[45], Matheus Ribeiro Furtado de Mendonça[13], Mike Fralick[46], Min Kyu Kang[20], Mohammad Adil[2], Natalie Gangai[15], Peerapon Vateekul[47], Pierre Elnajjar[15], Sarah Hickman[19], Sharmila Majumdar[17], Shelley L. McLeod[48,49], Sheridan Reed[7,8], Stefan Gräf[50], Stephanie Harmon[8,51], Tatsuya Kodama[16], Thanyawee Puthanakit[32,33], Tony Mazzulli[52,53,54], Vitor Lima de Lavor[13], Yothin Rakvongthai[55], Yu Rim Lee[30], Yuhong Wen[2], Fiona J. Gilbert[19,56], Mona G. Flores[2,56] ✉ and Quanzheng Li[3,56]

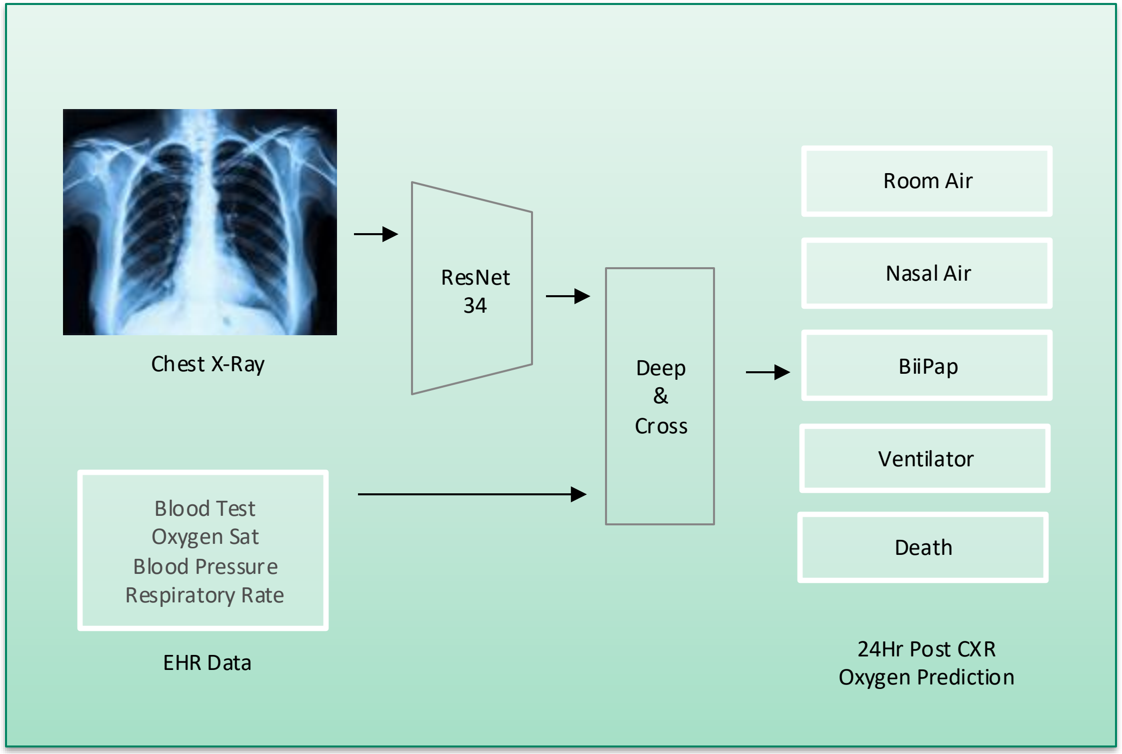
NVIDIA FEDERATED LEARNING FOR COVID-19 PATIENT CARE



EXAM Consortium

20 Sites | 8 Countries

COVID-19 Oxygen Prediction



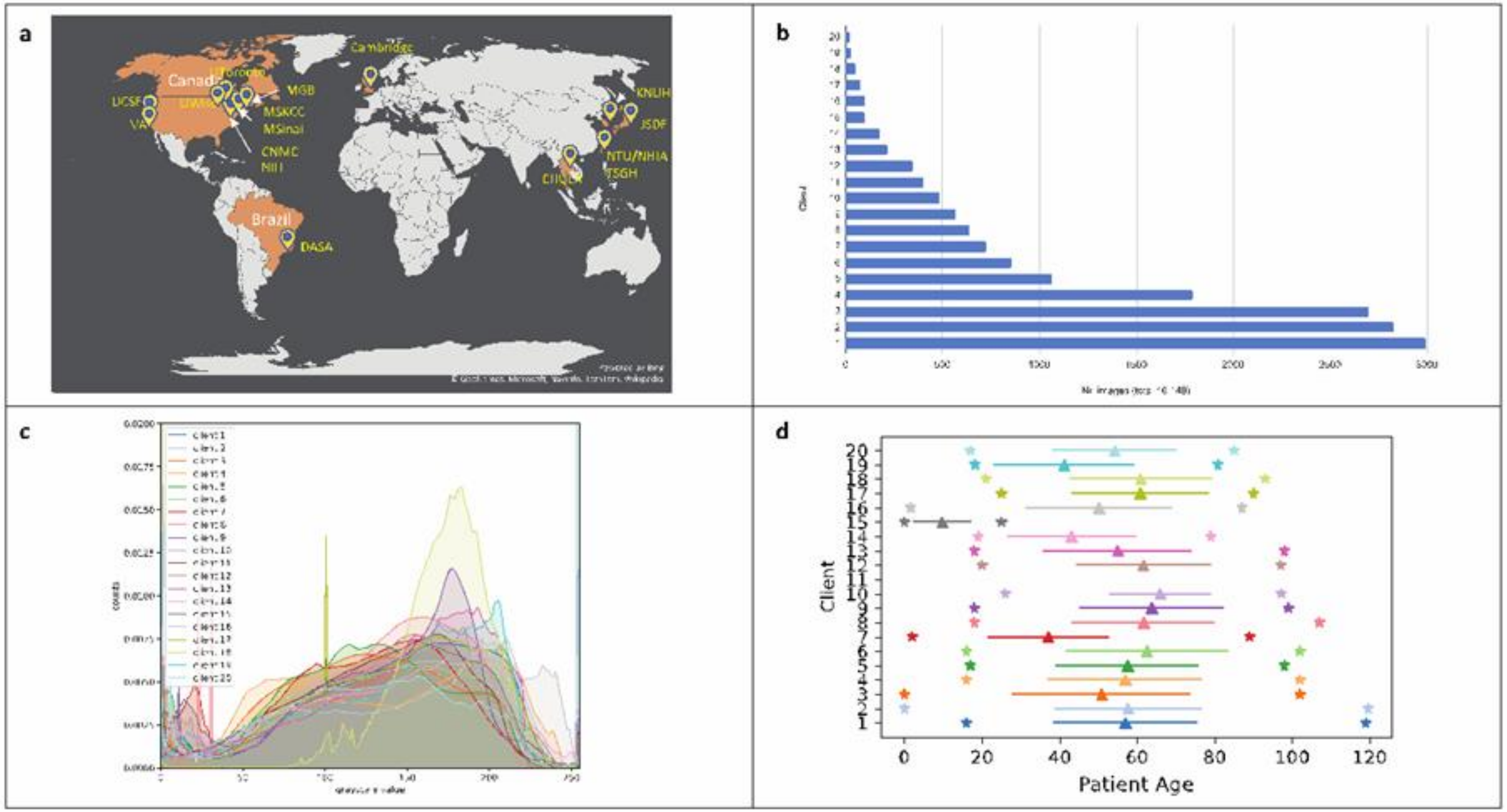
Global Model Achieved .93AUC

>25% Relative Improvement

Every Site Benefited Regardless of Dataset Size

Dayan et al. Nature Medicine, Sep 15, 2021: https://www.nature.com/articles/s41591-021-01506-3

USING FEDERATED LEARNING TO LEVERAGE A GLOBAL DATASET

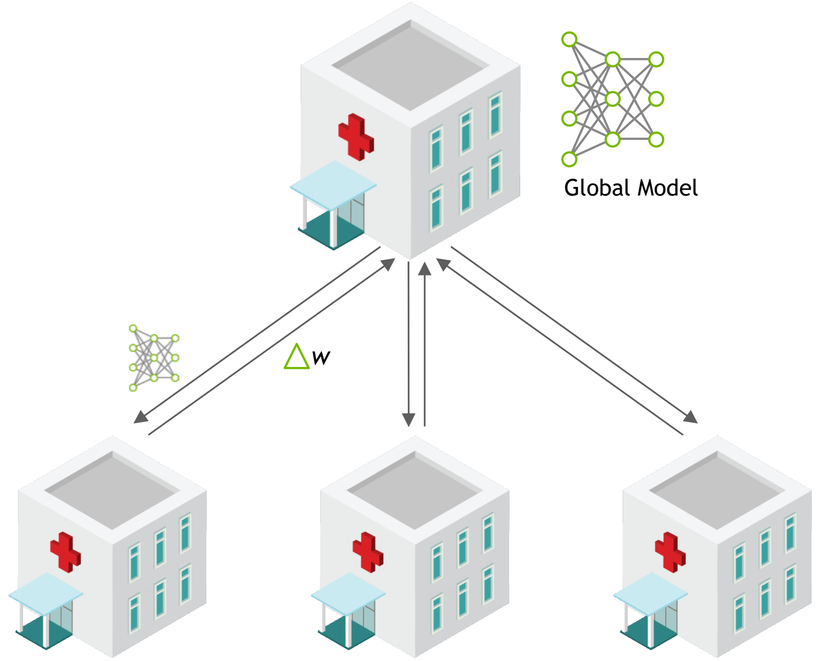


a, World map indicating the 20 different client sites contributing to the EXAM study. **b**, Number of cases contributed by each institution or site (client 1 represents the site contributing the largest number of cases). **c**, Chest X-ray intensity distribution at each client site. **d**, Age of patients at each client site, showing minimum and maximum ages (asterisks), mean age (triangles) and standard deviation (horizontal bars).

CONTROLLER FOR MODEL TRAINING

Typical workflow for FedAvg, FedOpt, FedProx, etc.

FedAvg



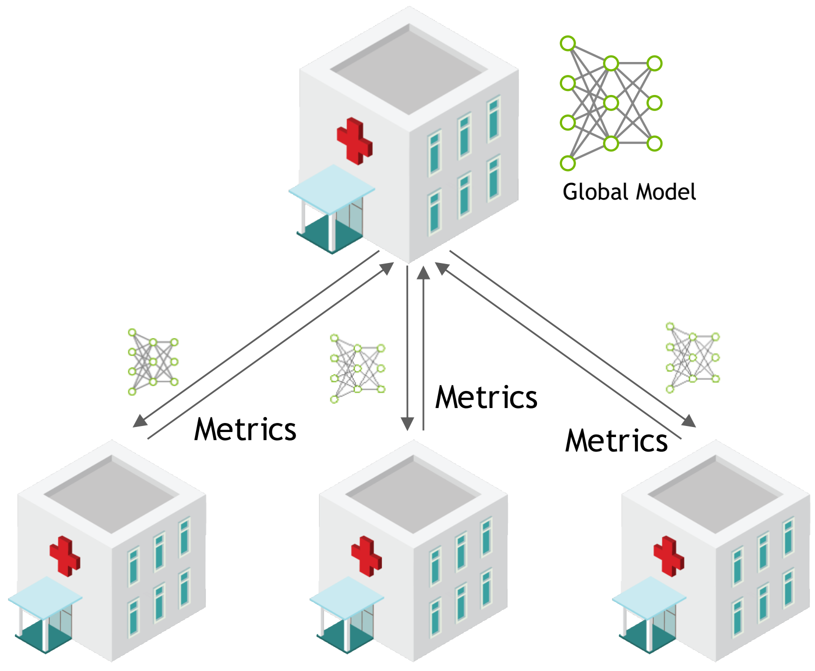
1. Server initializes model

2. For number of rounds:
    1. Server broadcasts global model to workers
    2. Workers validate global model and train on their data
    3. Workers keep track on their locally best model (Personalization)
    4. Workers send back updated model or updates
    5. Server Gathers (Aggregates) updates and updates the global model

Source: https://github.com/NVIDIA/NVFlare/blob/main/nvflare/app_common/workflows/fedavg.py

CONTROLLER FOR MODEL EVALUATION

Global model evaluation, Cross-site model evaluation



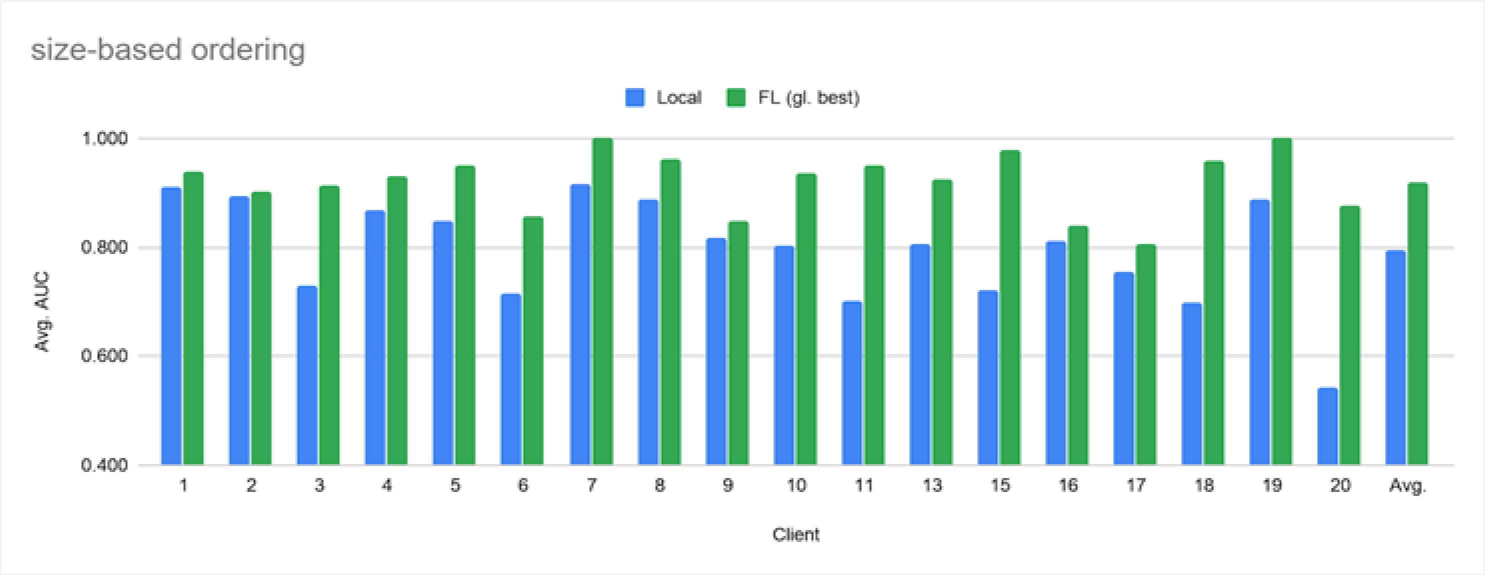
FedEval (Global Model Validation/Cross-Site Validation)

1. Server sends models (e.g., global model and registered best local models) to each worker for evaluation
2. Server gathers the resulting metrics

	Metrics	Evaluation sites			
		Site-1	Site-2	...	Site-N
Models	Global (Final)	...	...	...	...
	Global (Best)	...	...	...	...
	Site-1	...	...	...	...
	Site-2	...	...	...	...
	...	...	...	...	...
	Site-N	...	...	...	...

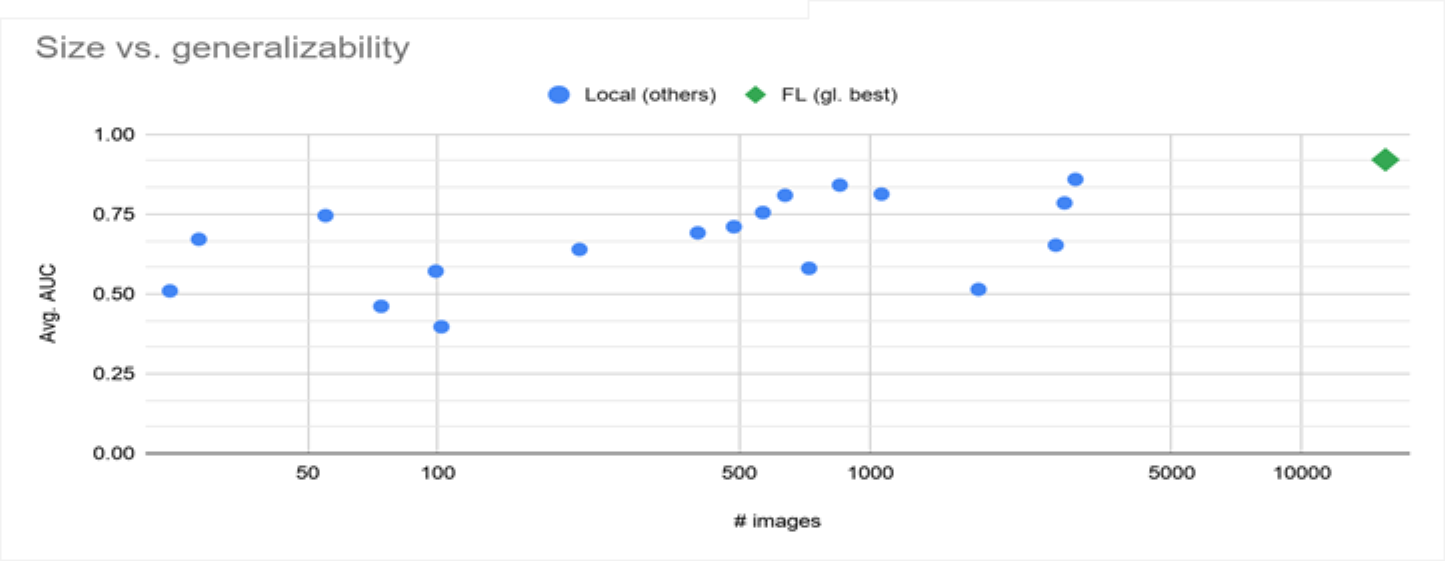
Source: https://github.com/NVIDIA/NVFlare/blob/main/nvflare/app_common/workflows/global_model_eval.py
https://github.com/NVIDIA/NVFlare/blob/main/nvflare/app_common/workflows/cross_site_model_eval.py

NVIDIA FL FOR COVID-19



FL resulted on average in:

- 16% performance improvement
- 38% generalizability improvement



Dayan et al. Nature Medicine, Sep 15, 2021: https://www.nature.com/articles/s41591-021-01506-3

NVIDIA FL FOR COVID-19: EXTERNAL VALIDATION

Predicting need for mechanical ventilation

YouTube: https://youtu.be/cOXVrtkv6FE

EXAM Model Released on NGC:

24h avg. AUC: 0.94

72h avg. AUC: 0.91

https://ngc.nvidia.com/catalog/models/nvidia:med:clara_train_covid19_exam_ehr_xray

NVIDIA AV Federated Learning

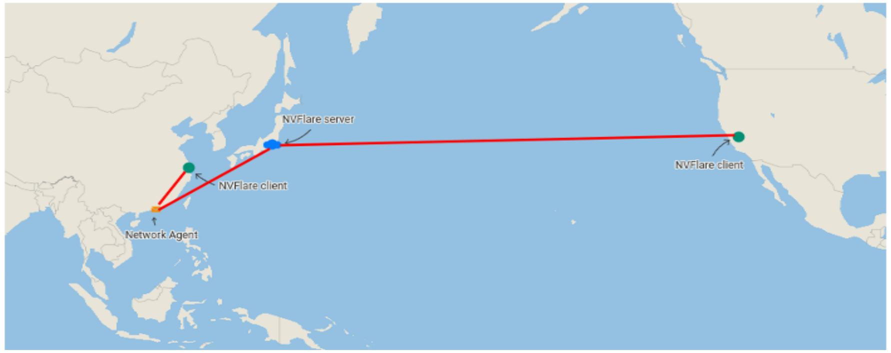
AV team leveraging FLARE to training object detection & tracking models

NVIDIA Autonomous Vehicle model training

- **Scenario**: small number of clients, and each client host a big dataset.
- **Workflow**: cyclic weight transfer
- **Goals:** Build a unified global model with the same or better accuracy. Reduce the effort for model approval process

- **Total Users**: **29** active users in the platform
- **Models trained:**

- **MLMCF** - A model to detect vehicles, person, bicycle, free space, parking area
- **DoNET** - A model to detect the status of vehicles, for example, lamp status (is light on/off), door status (is door open or not), etc.
- **Waitnet** - A model to detect static objects such as traffic lights, traffic signs, road marks, stop lines, and crosswalks.
- **PathNet/RoadNet** — Detecting the path the autonomous vehicle takes.
- **RadarNet** — Radar sensor data as input to predict the obstacle around the vehicle.
- **PredetionNet** — Object tracking and trajectory prediction
- **EGM** — Multi-camera input-based model for detecting the barrier in the parking lot such as height limit pole, pillar.



Cross-border federated learning setup: Clients in China & USA
Servers in Japan, Hong Kong

80 sign

12 sign

US Model: only capture the whole sign

FL Global Model: capture the individual objects and whole sign

Blog: https://developer.nvidia.com/blog/federated-learning-in-autonomous-vehicles-using-cross-border-training

Agenda

- What is Federated Learning?
- NVIDIA Key Technologies for Federated Learning
- Real-world Use cases of Federated Learning
- **Getting Started with NVIDIA FLARE**
- Research: Addressing Key challenges in Federated Learning
- Summary & Announcements

Server Code: Controller

```python
class FedAvg(BaseFedAvg):

    def run(self) -> None:
        self.info("Start FedAvg.")

        model = self.load_model()
        model.start_round = self.start_round
        model.total_rounds = self.num_rounds

        for self.current_round in range(self.start_round, self.start_round + self.num_rounds):
            self.info(f"Round {self.current_round} started.")
            model.current_round = self.current_round

            clients = self.sample_clients(self.min_clients)

            results = self.send_model_and_wait(targets=clients, data=model)

            aggregate_results = self.aggregate(
                results, aggregate_fn=None
            )  # if no `aggregate_fn` provided, default `WeightedAggregationHelper` is used

            model = self.update_model(model, aggregate_results)

            self.save_model(model)

        self.info("Finished FedAvg.")
```

Client Code:
Convert PyTorch to NVFlare

[PyTorch CIFAR-10 Tutorial]

```python
from net import Net


def main():
    transform = transforms.Compose([...])

    trainset = torchvision.datasets.CIFAR10(...)
    trainloader = torch.utils.data.DataLoader(...)

    testset = torchvision.datasets.CIFAR10(...)
    testloader = torch.utils.data.DataLoader(...)

    net = Net()

    criterion = nn.CrossEntropyLoss()
    optimizer = optim.SGD(...)

    # Train loop
    for epoch in range(epochs):
        ...

    print("Finished Training")
```

```python
from net import Net

import nvflare.client as flare


def main():
    transform = transforms.Compose([...])

    trainset = torchvision.datasets.CIFAR10(...)
    trainloader = torch.utils.data.DataLoader(...)

    testset = torchvision.datasets.CIFAR10(...)
    testloader = torch.utils.data.DataLoader(...)

    net = Net()

    criterion = nn.CrossEntropyLoss()
    optimizer = optim.SGD(...)

    flare.init()

    while flare.is_running():
        input_model = flare.receive()
        print(f"current_round={input_model.current_round}")

        net.load_state_dict(input_model.params)

        for epoch in range(epochs):  # loop over the dataset multiple times
            ...

        print("Finished Training")

        output_model = flare.FLModel(
            params=net.cpu().state_dict(),
            metrics={"accuracy": accuracy},
            meta={"NUM_STEPS_CURRENT_ROUND": epochs * len(trainloader)},
        )
        flare.send(output_model)
```

1. import client API
2. Initialize
3. Receive global model
4. Load global model
5. Send back the updated model

Create a FedJob and Run Simulation

```python
if __name__ == "__main__":
    n_clients = 2
    num_rounds = 2
    train_script = "src/cifar10_fl.py"

    # Create basic fed Job with initial model
    job = BaseFedJob(
        name="cifar10_pt_fedavg",
        initial_model=Net(),
    )

    # Define the controller and send to server
    controller = FedAvg(
        num_clients=n_clients,
        num_rounds=num_rounds,
    )
    job.to_server(controller)

    # Add clients
    for i in range(n_clients):
        runner = ScriptRunner(script=train_script)  # script_args=f"--batch_size 32 --data_path /data/site-{i}"
        job.to(runner, target: f"site-{i}")

    # job.export_job("/tmp/nvflare/jobs/job_config")    # Exported jobs can be used in real deployment!
    job.simulator_run( workspace: "/tmp/nvflare/jobs/workdir", gpu="0")
```

Client code: Lightning client API

Transform your script to FL with a few lines of code changes:

1. Import NVFlare lightning API
2. Patch your lightning trainer
3. (Optionally) validate the current global model
4. Train as usually

```
from nemo.core.config import hydra_runner
from nemo.utils import AppState, logging
from nemo.utils.exp_manager import exp_manager
from nemo.utils.model_utils import inject_model_parallel_rank

# (0): import nvflare lightning api
import nvflare.client.lightning as flare

mp.set_start_method("spawn", force=True)
```

...

```
# (1): flare patch
flare.patch(trainer)

while flare.is_running():

    # (2) evaluate the current global model to allow server-side model selection
    print("--- validate global model ---")
    trainer.validate(model)

    # (3) Perform local training starting with the received global model
    print("--- train new model ---")
    trainer.fit(model)
```

Demo

NVIDIA FLARE Website

https://nvidia.github.io/NVFlare

Agenda

- What is Federated Learning?
- NVIDIA Key Technologies for Federated Learning
- Real-world Use cases of Federated Learning
- Getting Started with NVIDIA FLARE
- **Research: Addressing Key challenges in Federated Learning**
- Summary & Announcements

Medical Imaging

ADDRESSING DATA HETEROGENEITY

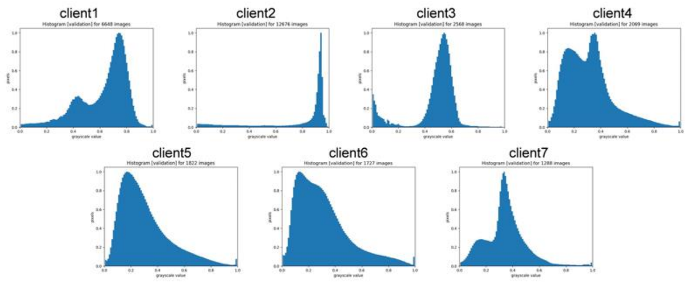


Fig. 4: Intensity distribution at different sites.

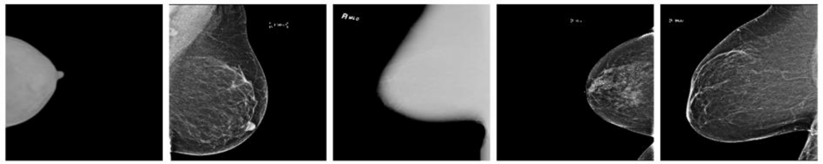


Fig. 1: Mammography data examples from different sites after resizing the original images to a resolution of 224 × 224. No special normalization was applied in order to keep the scanners' original intensity distribution that can be observed in 4.

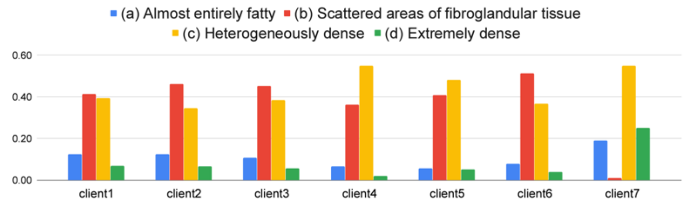


Fig. 3: Class distribution at different client sites as a fraction of their total data.

Institution	Train	# Val.	# Test
client1	22933	3366	6534
client2	8365	1216	2568
client3	44115	6336	12676
client4	7219	1030	2069
client5	6023	983	1822
client6	6874	853	1727
client7	4021	664	1288

Standard FedAvg is sub-optimal if the clients have heterogeneous (non-i.i.d.) distributions

- **Personalization** can help close this gap.
- Carefully tuning hyperparameter plays a critical role to achieve optimal performance (**Auto-ML/FL**).

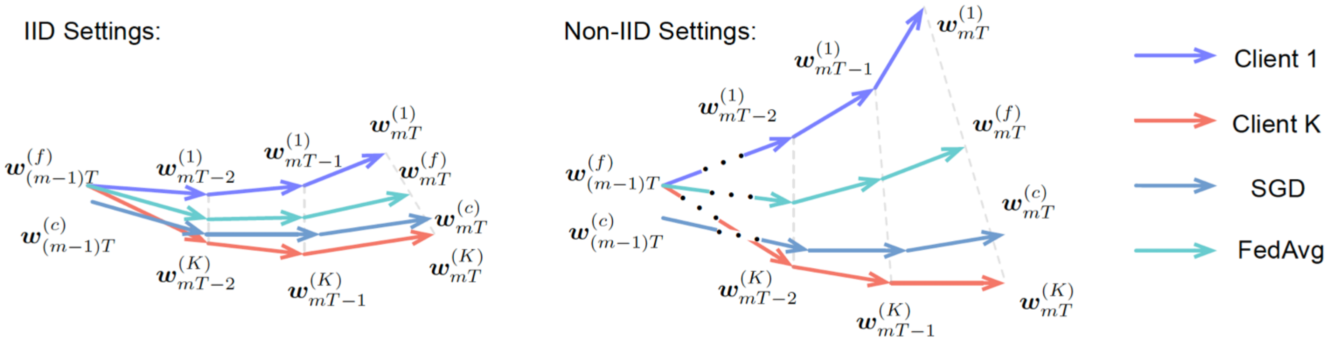
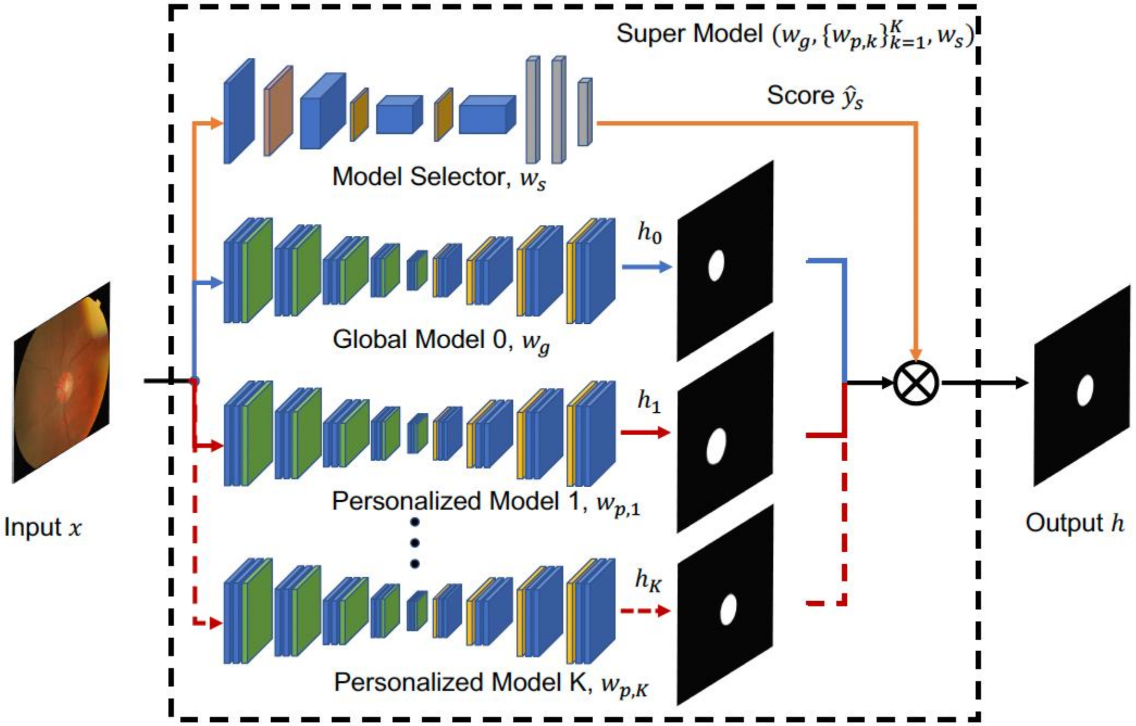


Figure sources: Intensity, label & quantity distribution of Mammography images across sites, Roth et al. DCL 2020; Zhao, et. al, Federated Learning with Non-IID Data, 2018

FEDERATED SUPER MODEL (FedSM)

Closing the gap between FL and centralized training CVPR 2022

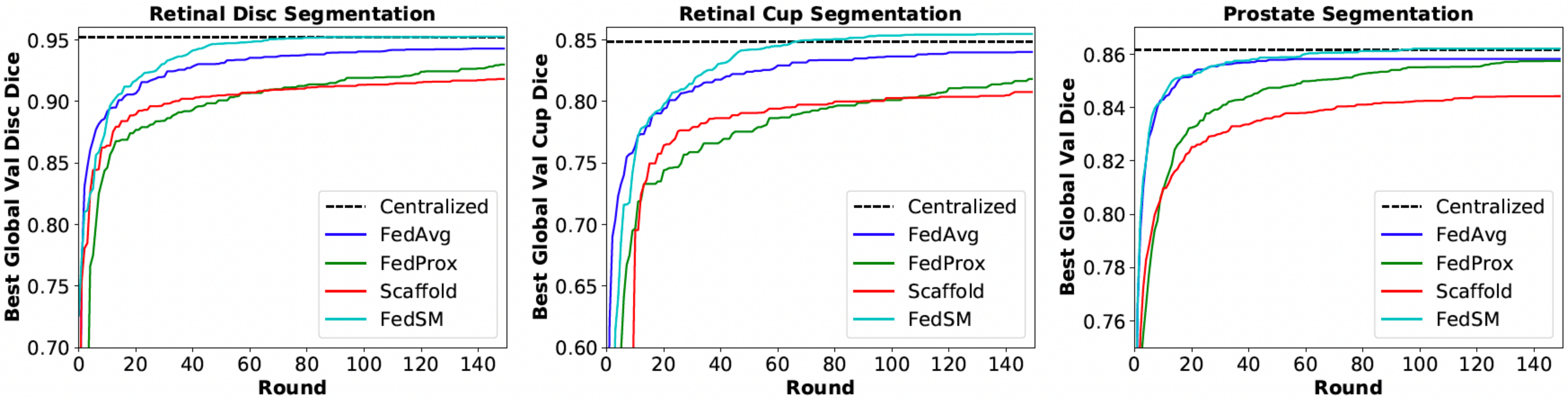
- The goal of FL is to collaboratively train one global model that generalizes well on all clients' joint data distribution – which is hard.
- Instead of finding one global model that tries to fits all clients' data distribution, we propose:
  - Personalized models to fit different data distributions well
  - A soft-pull mechanism to harmonize the global and local information
  - A model selector to decide the closest model/data distribution for any unseen test data from one super model.
- If input is similar to a site N, use model N otherwise, use global model.



An Xu et al. "Closing the Generalization Gap of Cross-silo Federated Medical Image Segmentation", CVPR 2022 (Arxiv 2203.10144)

FEDSM: RESULTS

Closing the validation gap to centralized training

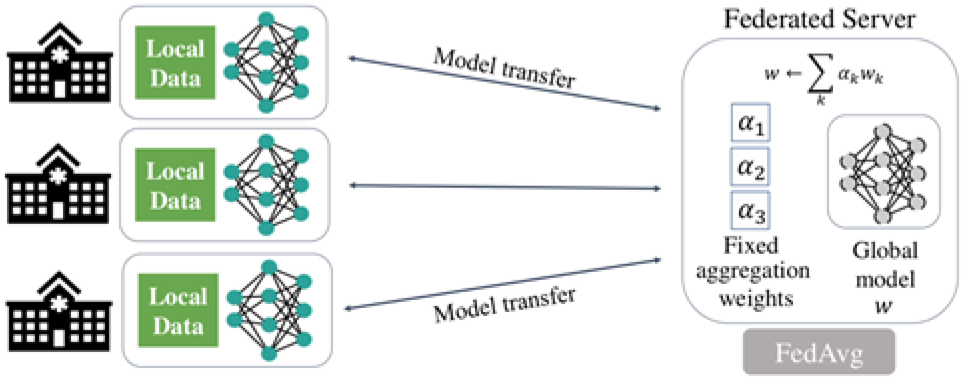


An Xu et al. “Closing the Generalization Gap of Cross-silo Federated Medical Image Segmentation“, CVPR 2022 (Arxiv 2203.10144)

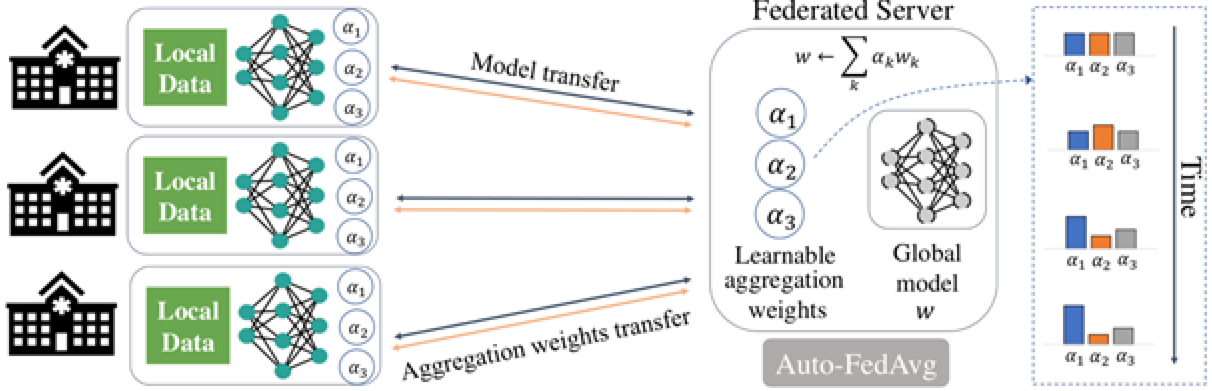
AUTO-FEDAVG: LEARNABLE FEDERATED AVERAGING FOR MULTI-INSTITUTIONAL MEDICAL IMAGE SEGMENTATION

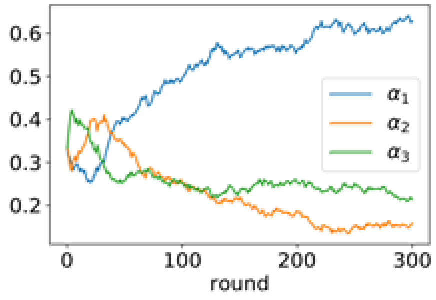
Yingda Xia et al.

FedAvg:

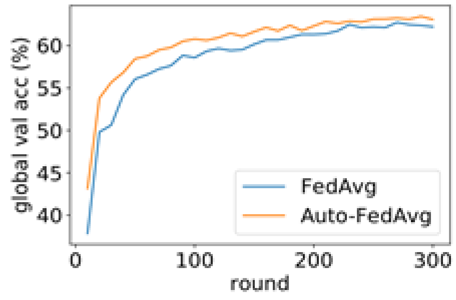


Auto-FedAvg:

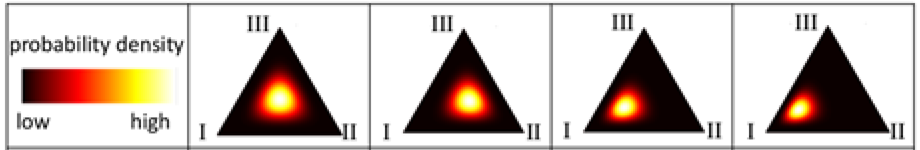




(a) The learning curve of α.



(b) Validation accuracy growth.



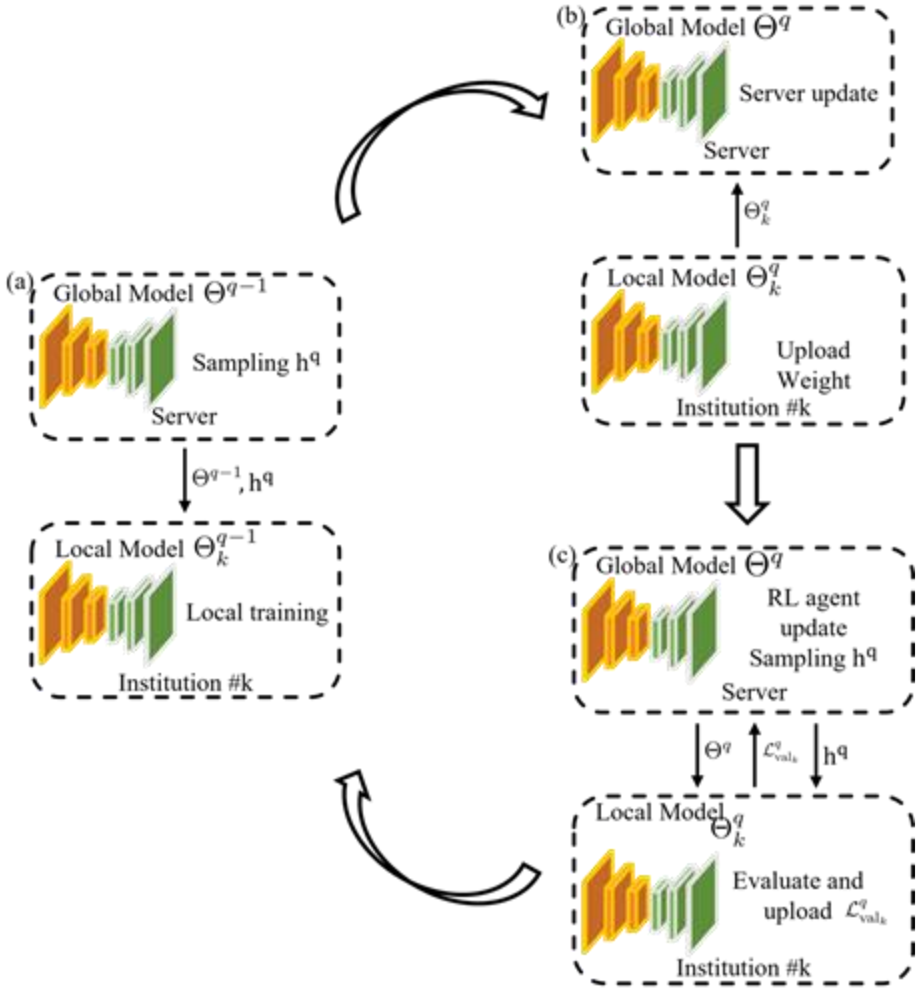
round	0	30	150	300
β	(6.0, 6.0, 6.0)	(6.4, 8.0, 6.4)	(13.4,6.0,6.4)	(18.3, 5.3, 6.9)
α	(0.33,0.33,0.33)	(0.30,0.39,0.31)	(0.54,0.22,0.24)	(0.63,0.16,0.21)

(c) Visualizations of Dirichlet distribution.

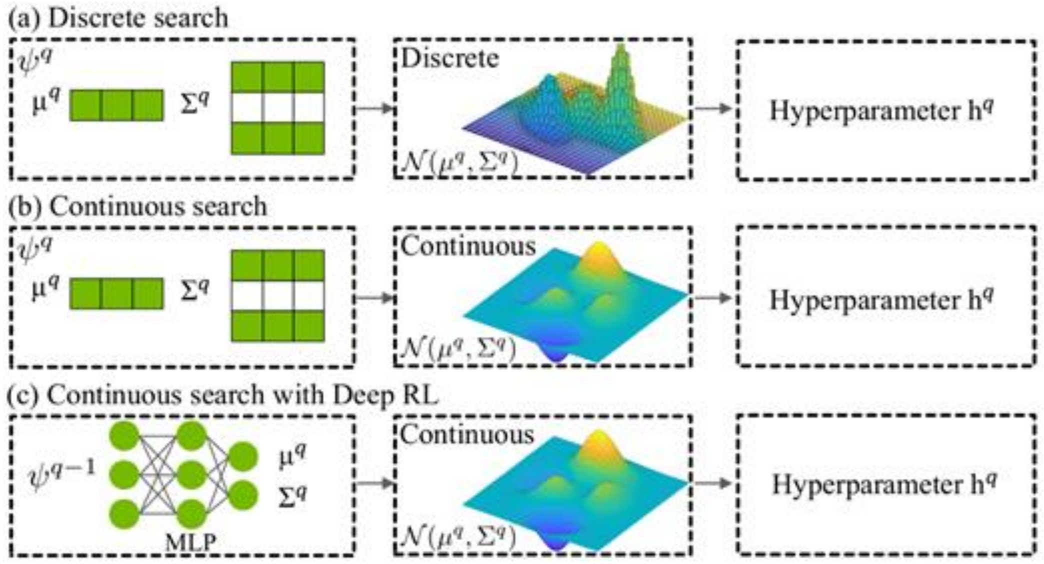
We show benefits for COVID lesion and pancreas segmentation in FL.

Yingda Xia et al. 2021 https://arxiv.org/abs/2104.10195

AUTO-FEDRL: OPTIMIZE ALL FL HYPERPARAMETERS



Search strategies:



The proposed FL method consists of the following three steps:

a) The server broadcasts the current **global model and hyperparameters** to all **clients**. Clients perform **local training**

b) Clients upload the trained local models, and the server updates the **global model**

c) Clients evaluate the received the **new global model** and upload their **validation loss**. The server updates the **RL agent** and distributes the a **new set of hyperparameter to all clients**

Pengfei Guo et al. (ECCV 2022): “Auto-FedRL: Federated Hyperparameter Optimization for Multi-institutional Medical Image Segmentation”

AUTO-FEDRL: EXPERIMENTS

Results

CIFAR-10

Method	Accuracy (%)
FedAvg [29]	88.43
FedProx [22]	89.45
Mostafa *et al.* [32]	89.86
Auto-FedAvg [48]	89.16
Auto-FedRL(dss)	90.70
Auto-FedRL(css)	90.85
Auto-FedRL(css MLP)	**91.27**
Centralized	92.56

Auto-FedRL search space:

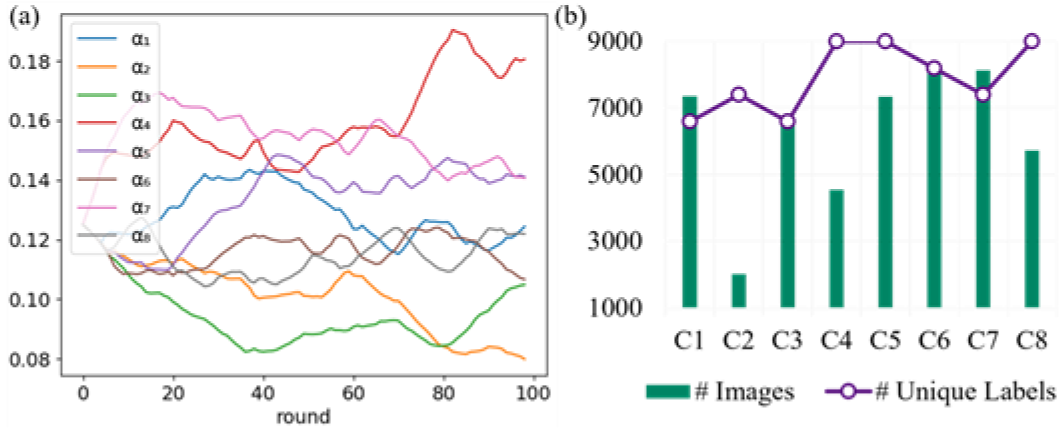
Learning Rate (LR)
Local iterations (LI)
Aggregation weights (AW)
Server learning rate (SLR)

Search Space Type	Memory Usage	Running Time for Search
Discrete	42.8 GB	8.246 s
Continuous	**3.00 GB**	**0.012 s**
Continuous MLP	3.13 GB	0.019 s

Mostafa et al.

Learning process of Auto-FedRL(css) in CIFAR-10

Non-IID setting



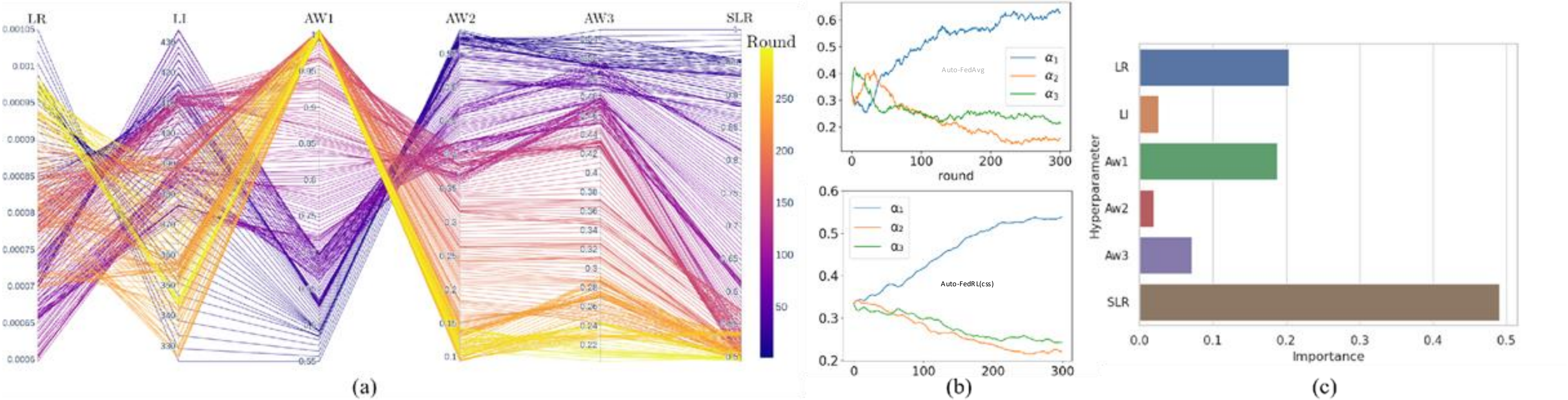
Analysis of the learning process of Auto-FedRL(css) in CIFAR-10. (a) the evolution of the distribution mean of aggregation weights during the training. (b) the statistic of different clients.

Pengfei Guo et al. (ECCV 2022): "Auto-FedRL: Federated Hyperparameter Optimization for Multi-institutional Medical Image Segmentation"

AUTO-FEDRL: VISUALIZATIONS

Learned Hyperparameters

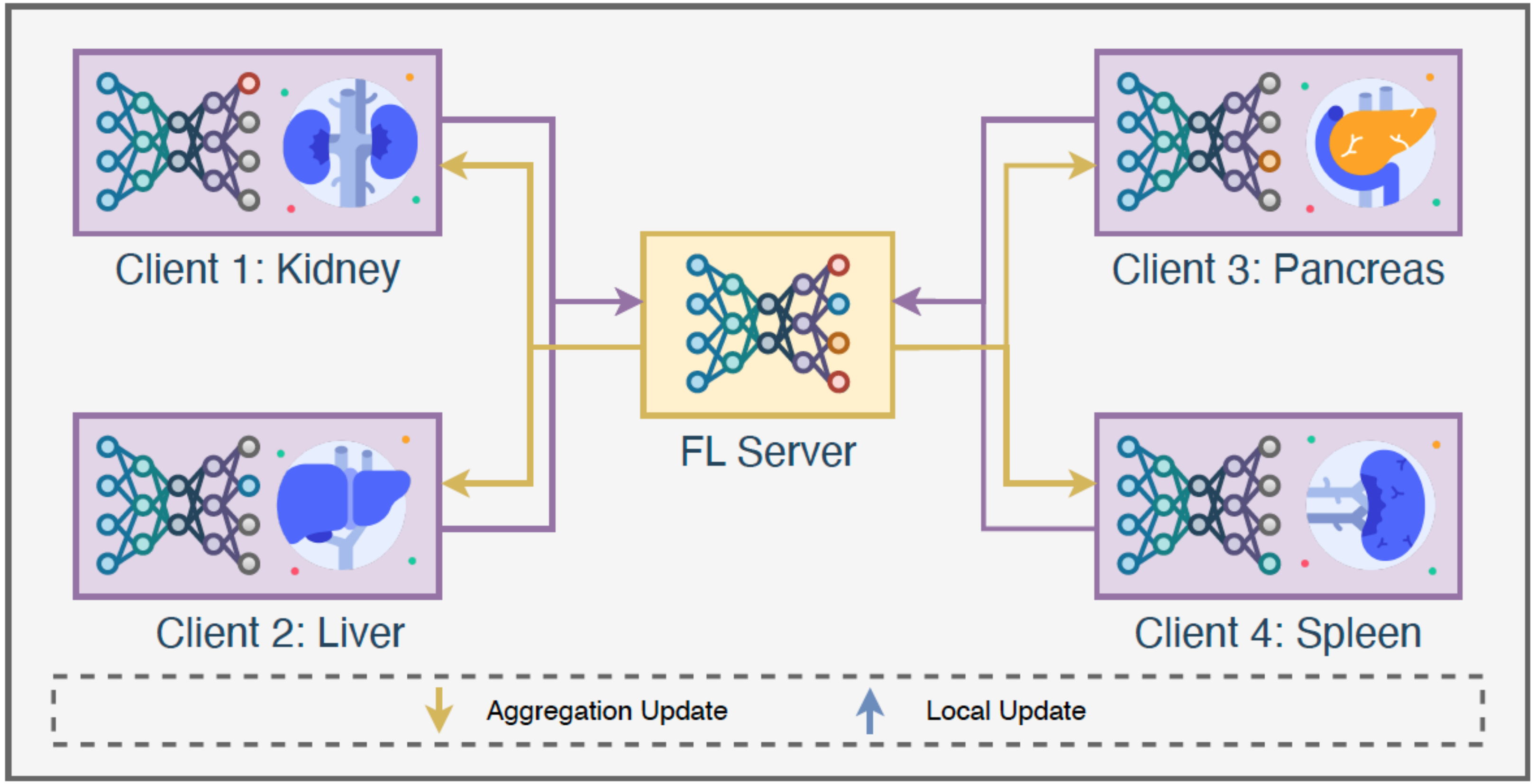
COVID-19 Lesion Segmentation



Analysis of the learning process of Auto-FedRL(css) in COVID-19 lesion segmentation. (a) The parallel plot of hyperparameter change during the training. LR, LI, AW, and SLR demotes learning rate, local iterations, aggregation weights, and server learning rate, respectively. (b) The aggregation weights evolution of Auto-FedAvg in top row and Auto-FedRL(css) in bottom row. (c) The importance analysis of different hyperparameter.

FEDERATED LEARNING WITH PARTIALLY LABELED DATA

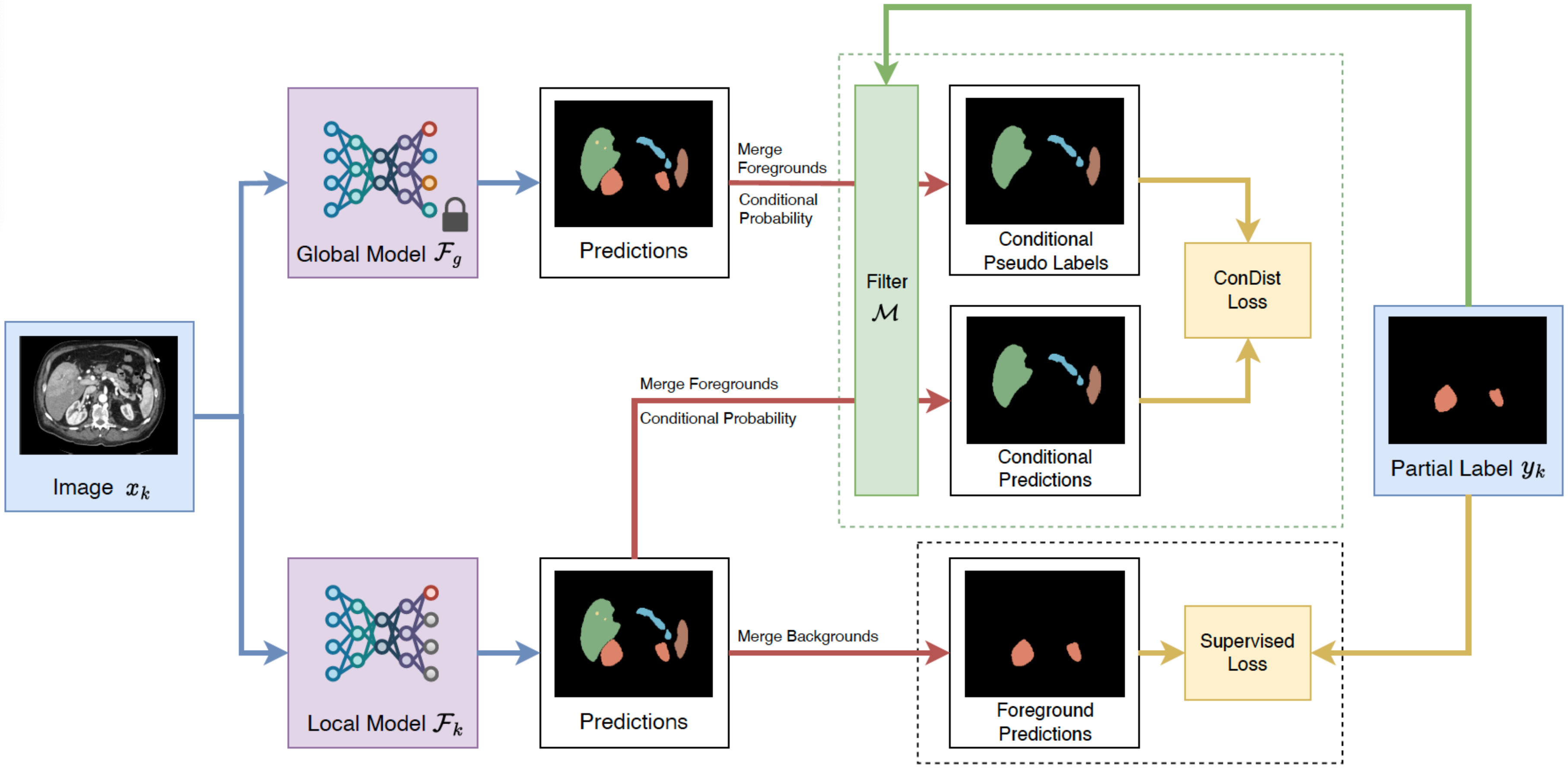
Utilize diverse annotated data



[Wang et al. **ConDistFL** (DeCaF workshop 2023) - https://arxiv.org/abs/2308.04070]

FEDERATED LEARNING WITH PARTIALLY LABELED DATA

A Conditional Distillation Approach

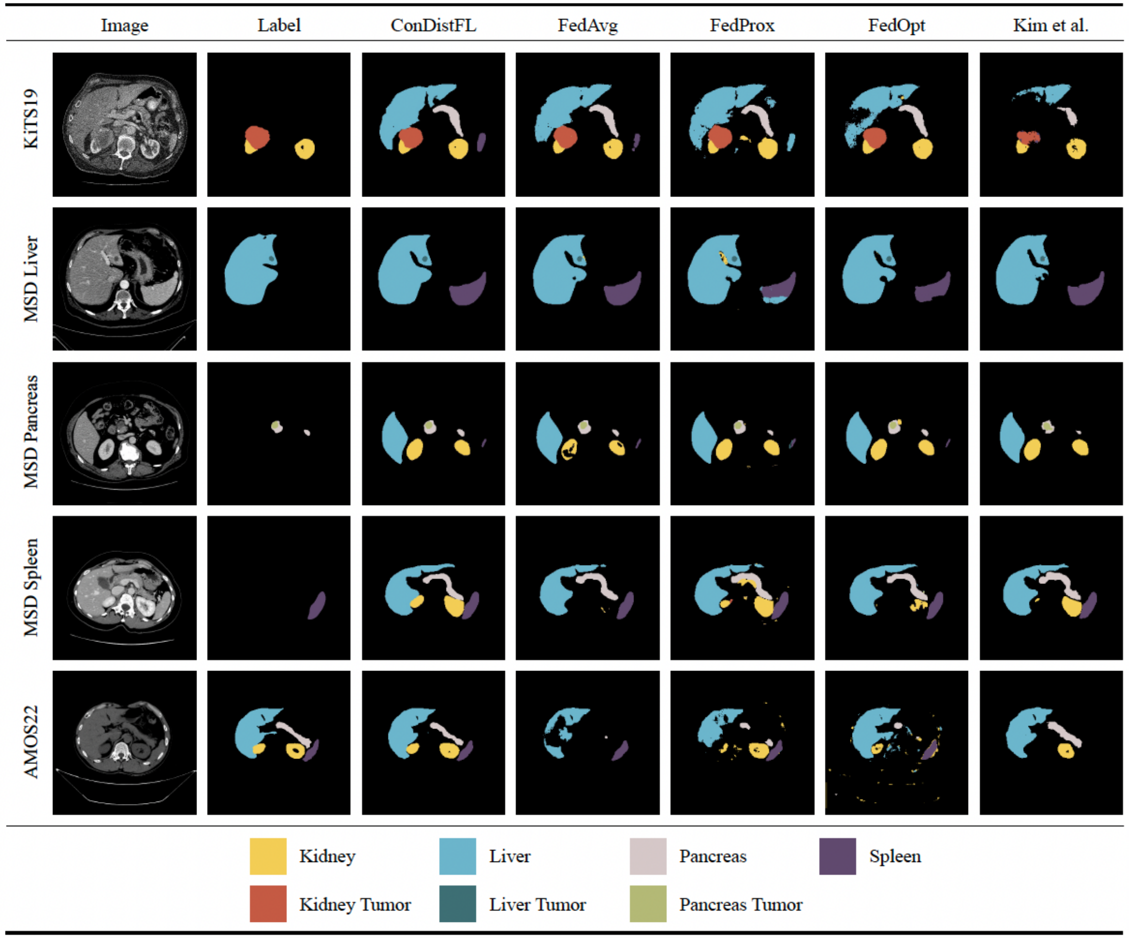


[Wang et al. **ConDistFL** (DeCaF workshop 2023) - https://arxiv.org/abs/2308.04070]

FEDERATED LEARNING WITH PARTIALLY LABELED DATA

Results

Method	Average Dice ↑	Kidney Organ	Kidney Tumor	Liver Organ	Liver Tumor	Pancreas Oragn	Pancreas Tumor	Spleen Organ
Standalone	0.8167	**0.9563**	0.8116	0.9520	0.7265	0.7846	0.5126	0.9631
FedAvg	0.8091	0.9536	0.7766	0.9608	0.7241	0.7824	0.5041	0.9618
FedProx	0.7432	0.7736	0.7481	0.8973	0.5968	0.7893	0.5049	0.8923
FedOpt	0.7598	0.6767	0.7751	0.8915	0.6908	0.7841	**0.5435**	0.9568
Kim et al.	0.7330	0.9430	0.6734	0.9386	0.6439	0.7207	0.3778	0.8334
ConDistFL (ours)	**0.8235**	0.9547	**0.8247**	**0.9613**	**0.7322**	**0.7975**	0.5278	**0.9664**



[Wang et al. **ConDistFL** (DeCaF workshop 2023) - https://arxiv.org/abs/2308.04070]

Large Language Models

Parameter-efficient Fine-tuning (PEFT)

Adapt Foundational LLMs in FL

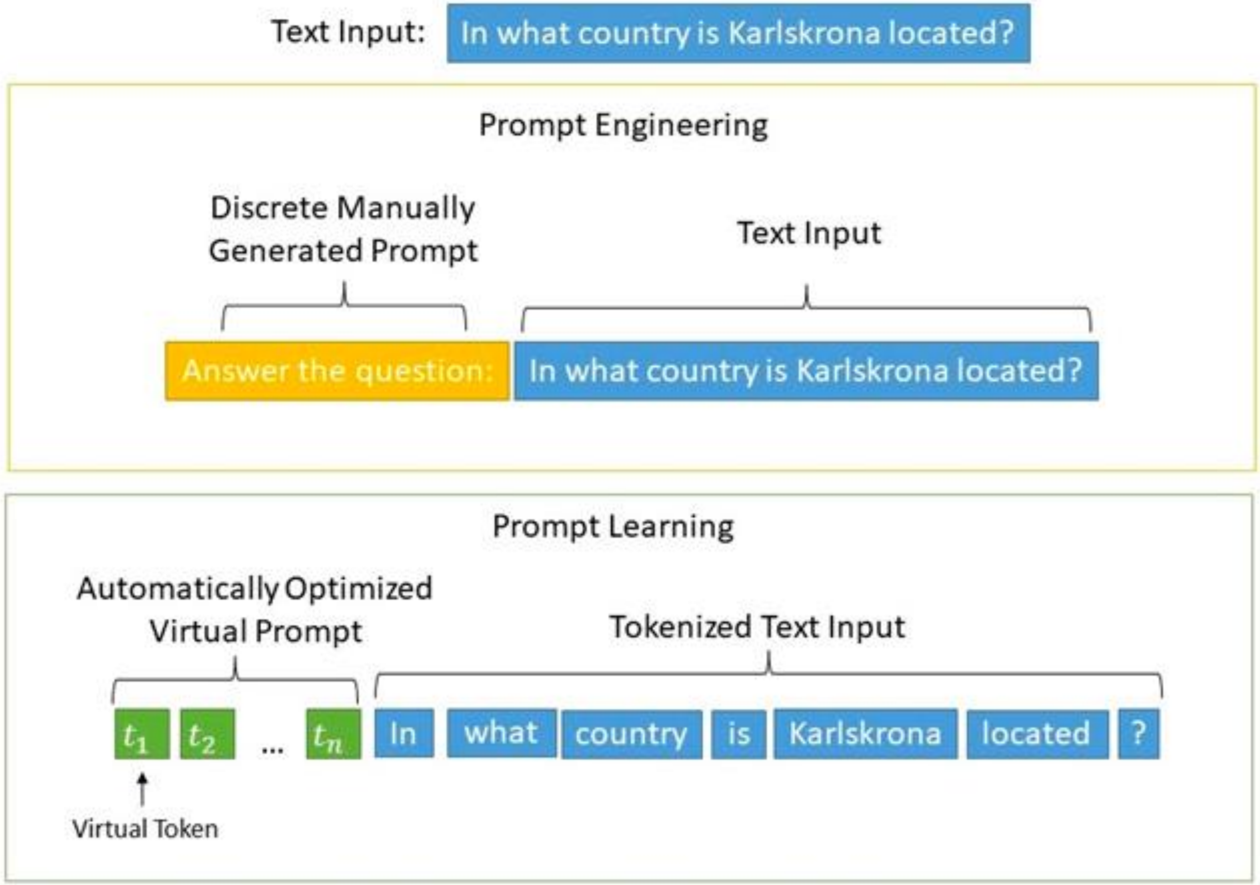
Parameter-efficient fine-tuning

Fine-tuning with a task-specific module

- **Most LLM layers fixed**; Only few dozen million params are being exchanged
- Tech: prompt-tuning/p-tuning/adapter/LoRA/others
- NVFlare example: sentiment analysis example with NeMo GPT model (345M/5B/**20B**)

Prompt Learning

Parameter-efficient adaptation of LLMs to downstream tasks



Tasks: brainstorming, classification, closed QA, generation, information extraction, open QA, summarization, etc.

Source: https://developer.nvidia.com/blog/adapting-p-tuning-to-solve-non-english-downstream-tasks

P-Tuning for Sentiment Analysis

Downstream task example:

- Financial PhraseBank dataset (Malo et al.) for sentiment analysis.
- The Financial PhraseBank dataset contains the sentiments for financial news headlines from a retail investor's perspective.

Example prompts and predictions:

The products have a low salt and fat content . ***sentiment: neutral***

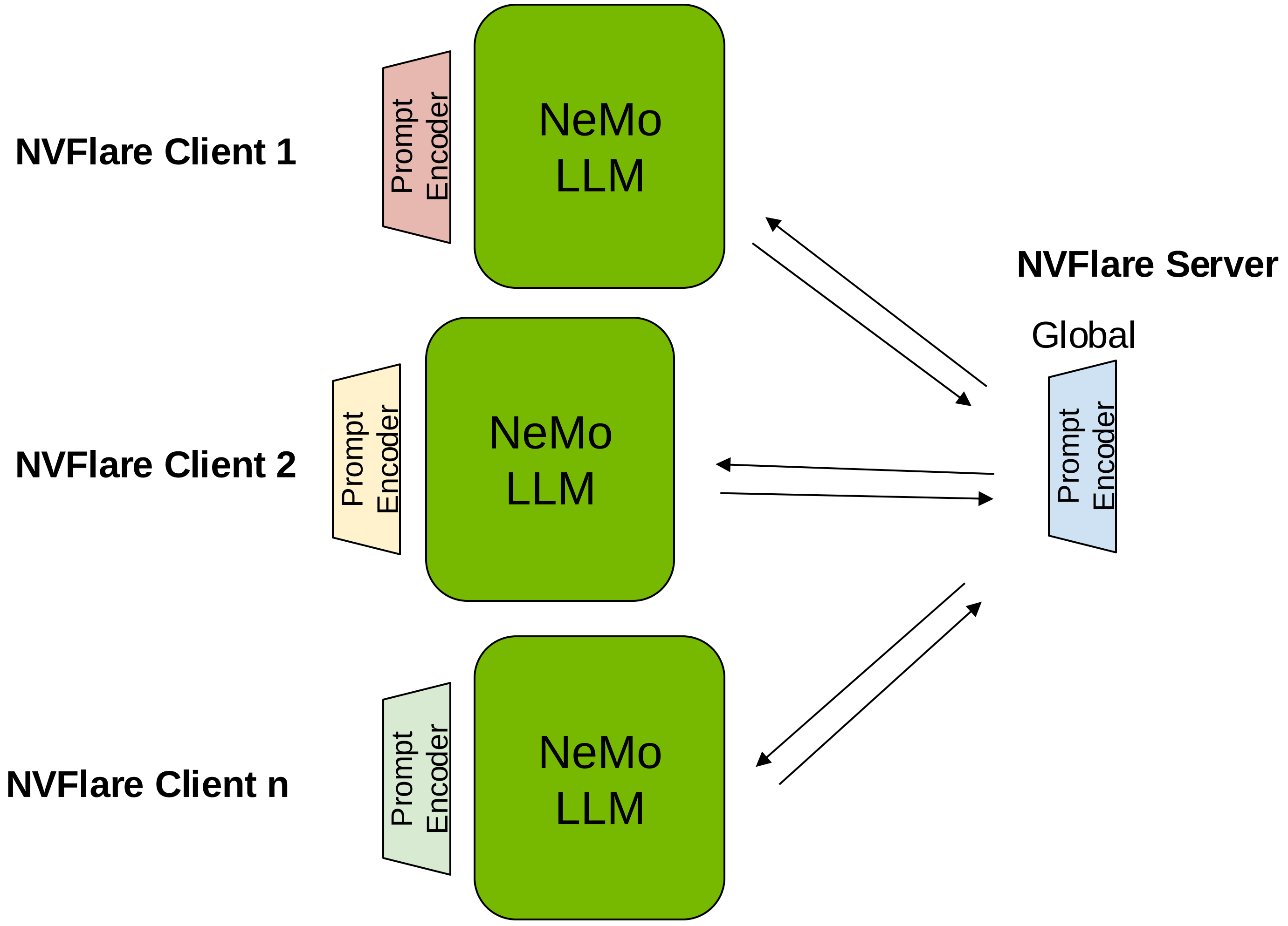
The agreement is valid for four years . ***sentiment: neutral***

Diluted EPS rose to EUR3 .68 from EUR0 .50 . ***sentiment: positive***

The company is well positioned in Brazil and Uruguay . ***sentiment: positive***

Profit before taxes decreased by 9 % to EUR 187.8 mn in the first nine months of 2008 , compared to EUR 207.1 mn a year earlier . ***sentiment: negative***

NVFlare for P-Tuning With NeMo



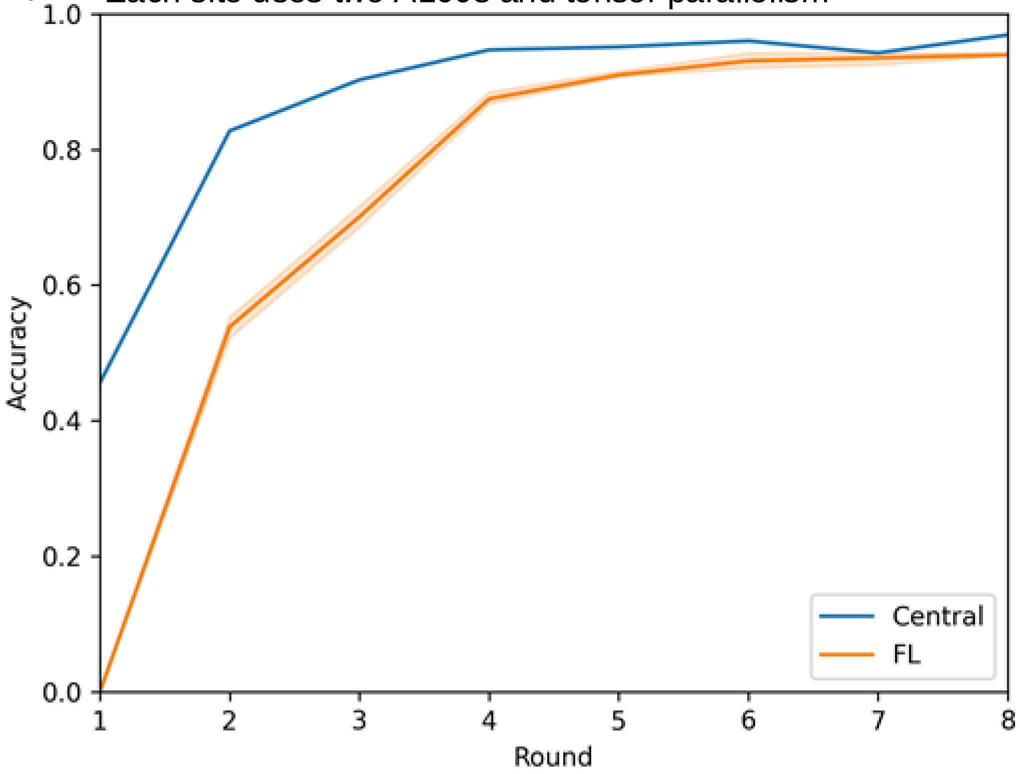
LLM parameters stay fixed; Prompt encoder parameters are trained/updated

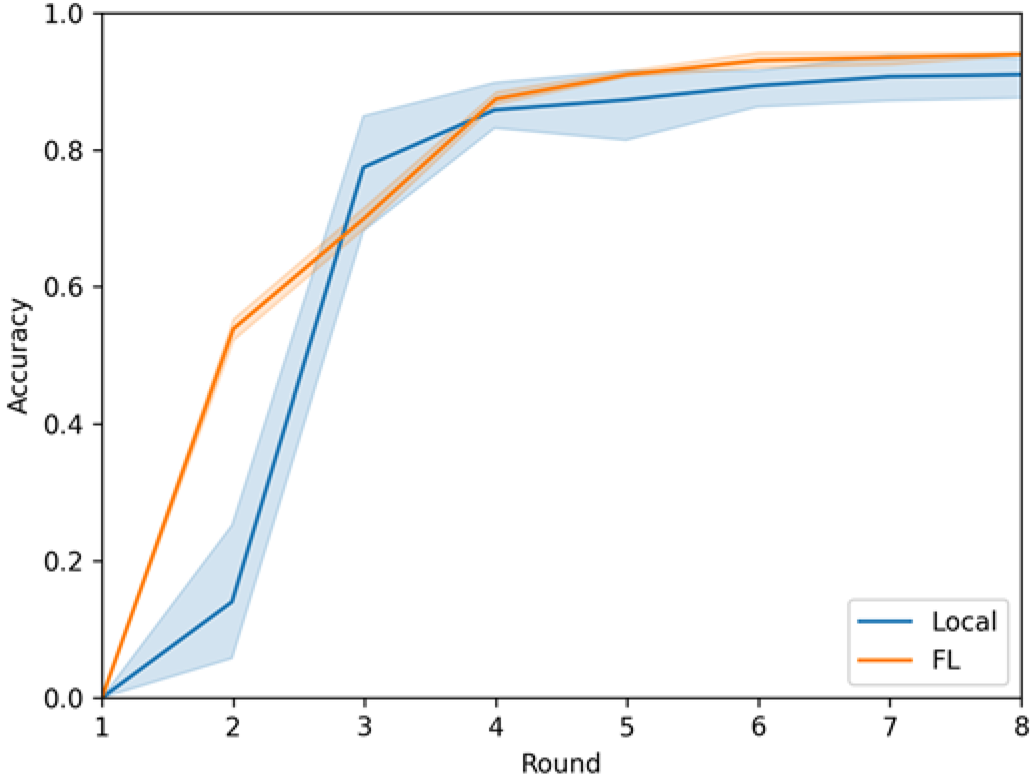
P-Tuning for Sentiment Analysis

FL can achieve performance comparable to centralized training

Federated p-tuning experiment:

- Using ***20B NeMo Megatron-GPT*** model hosted on HuggingFace
- **50M** parameters are updated (0.25%)
- 1800 pairs of statement and sentiment
- 600 for each site; shared validation set for direct comparison
- Each site uses two A100s and tensor parallelism

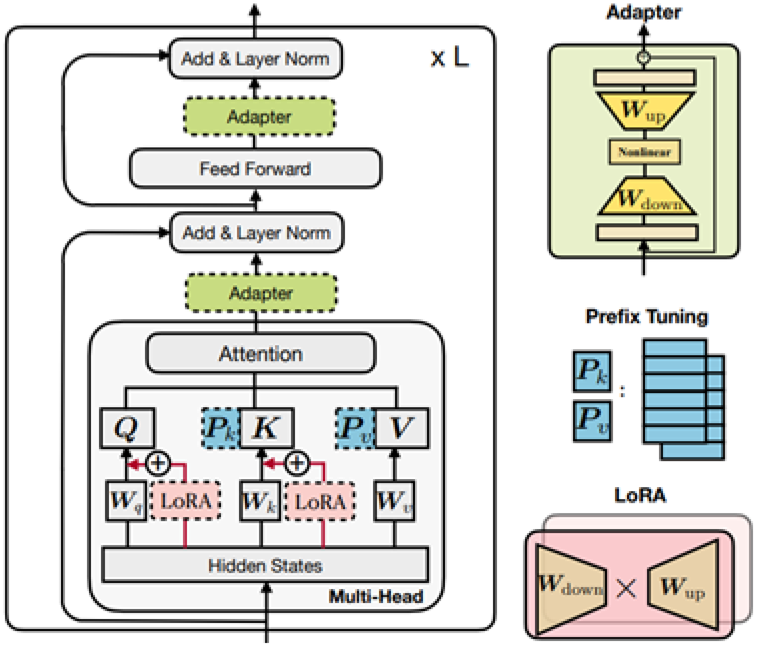




Compare PEFT Methods With NeMo

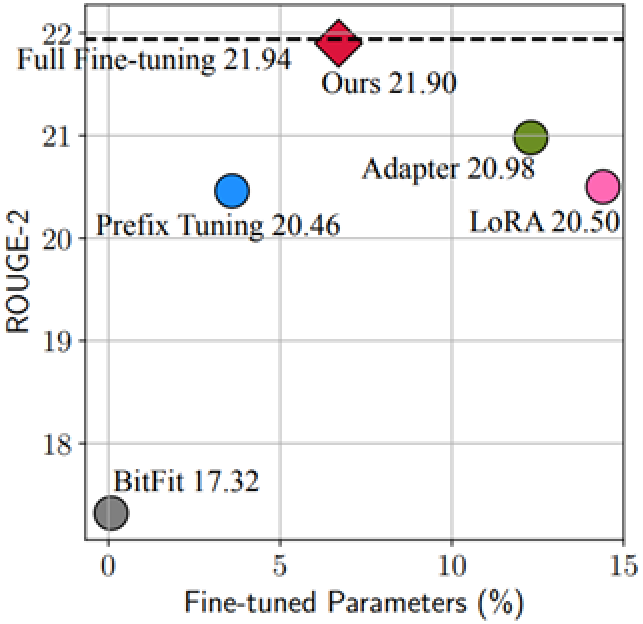
Lightning Client API + 1 line configuration change

Transformer and PEFT methods:



Source: https://arxiv.org/abs/2110.04366

Different PEFT methods on the XSum summarization task:

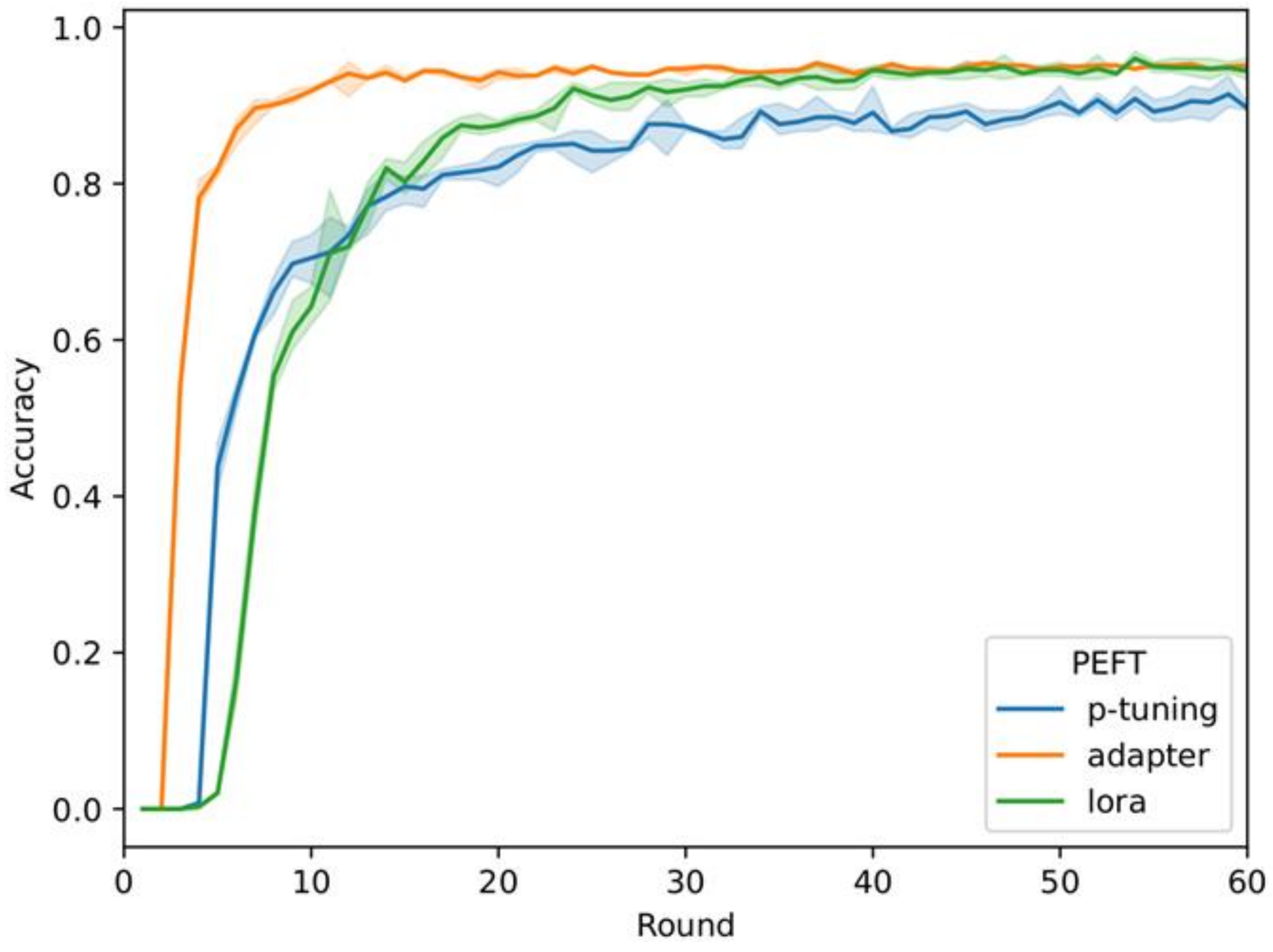



```yaml
peft:
  peft_scheme: "adapter"  # can be either adapter,ia3, or ptuning
  restore_from_path: null

  # Used for adapter peft training
  adapter_tuning:
    type: 'parallel_adapter' # this should be either 'parallel_adap
    adapter_dim: 32
    adapter_dropout: 0.0
    norm_position: 'pre' # This can be set to 'pre', 'post' or null
    column_init_method: 'xavier' # IGNORED if linear_adapter is use
    row_init_method: 'zero' # IGNORED if linear_adapter is used, op
    norm_type: 'mixedfusedlayernorm' # IGNORED if layer_adapter is 
    layer_selection: null  # selects in which layers to add adapter
    weight_tying: False
    position_embedding_strategy: null # used only when weight_tying

  lora_tuning:
    adapter_dim: 32
    adapter_dropout: 0.0
    column_init_method: 'xavier' # IGNORED if linear_adapter is use
    row_init_method: 'zero' # IGNORED if linear_adapter is used, op
    layer_selection:  null  # selects in which layers to add lora a
    weight_tying: False
    position_embedding_strategy: null # used only when weight_tying

  # Used for p-tuning peft training
  p_tuning:
    virtual_tokens: 10  # The number of virtual tokens the prompt e
    bottleneck_dim: 1024  # the size of the prompt encoder mlp bott
    embedding_dim: 1024  # the size of the prompt encoder embedding
    init_std: 0.023

  ia3_tuning:
    layer_selection:  null  # selects in which layers to add ia3 ad
```

NeMo YAML configuration

Compare PEFT Methods With NeMo

P-tuning vs. Adapter vs. LoRa



Tensor parallel with **2 GPUs** per client

345M Param NeMo GPT Megatron model

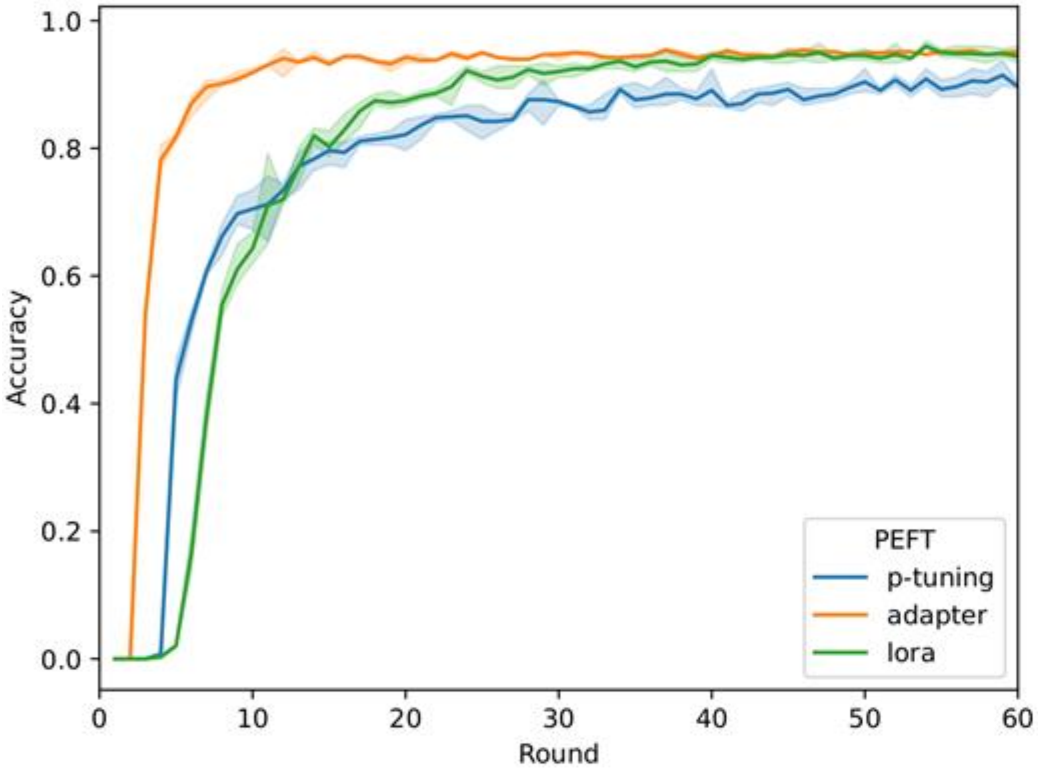
PEFT Method	Execution time
P-tuning	4h 59m
Adapter	11h 25m
LoRA	7h 27m

Example notebook

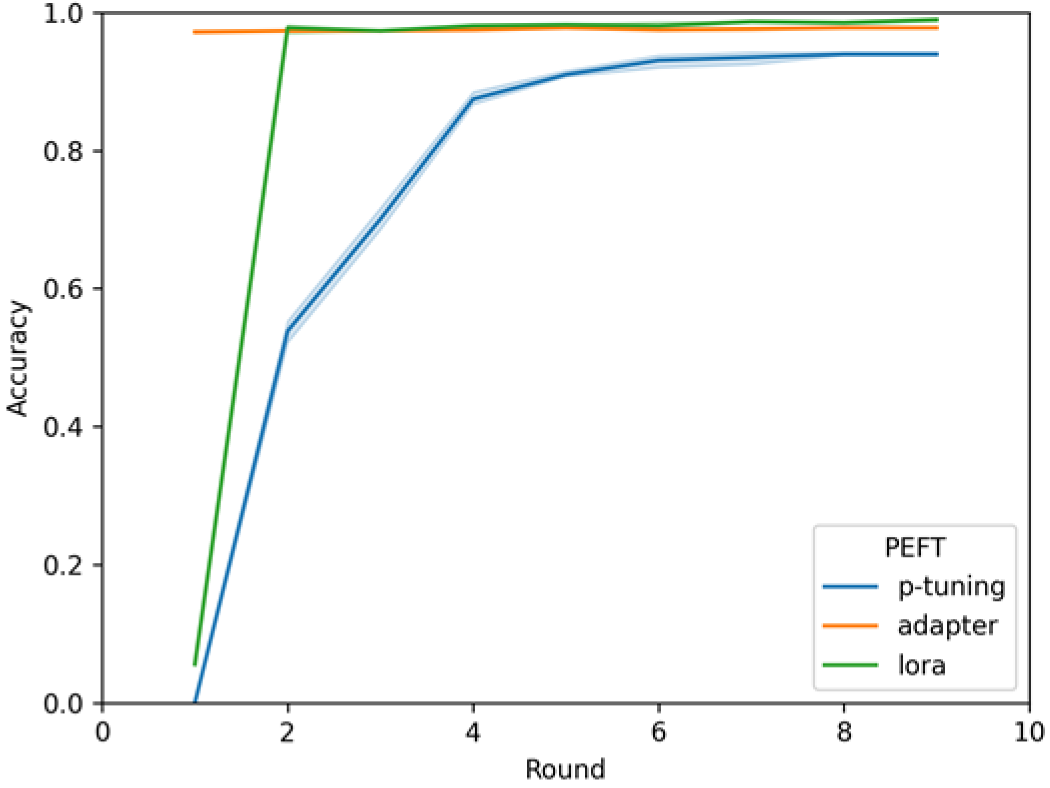
Compare PEFT Methods With NeMo

P-tuning vs. Adapter vs. LoRa

345m



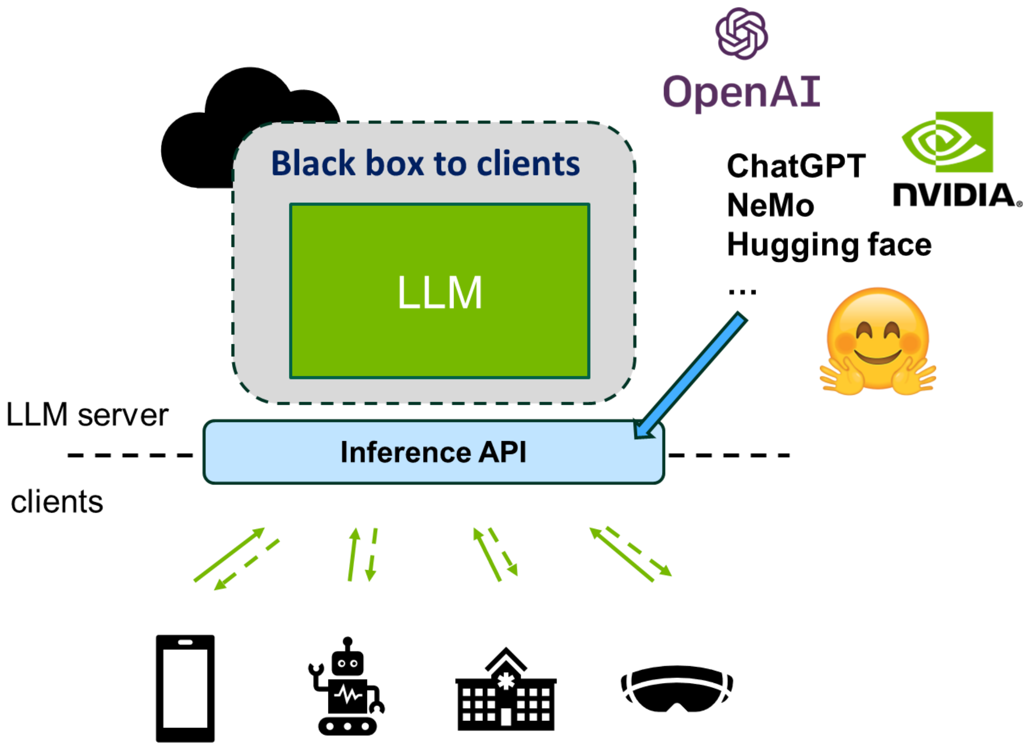
20B



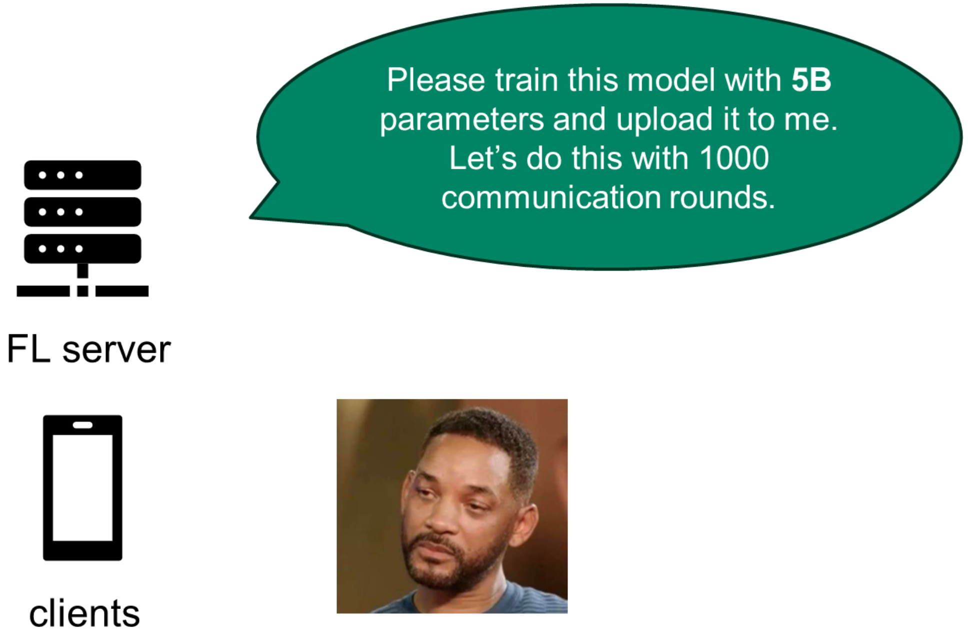
Example [notebook]

FedBPT: Efficient Federated Black-box Prompt Tuning for Large Language Models

ICML 2024



Model access



Computation/memory limitation

Communication cost

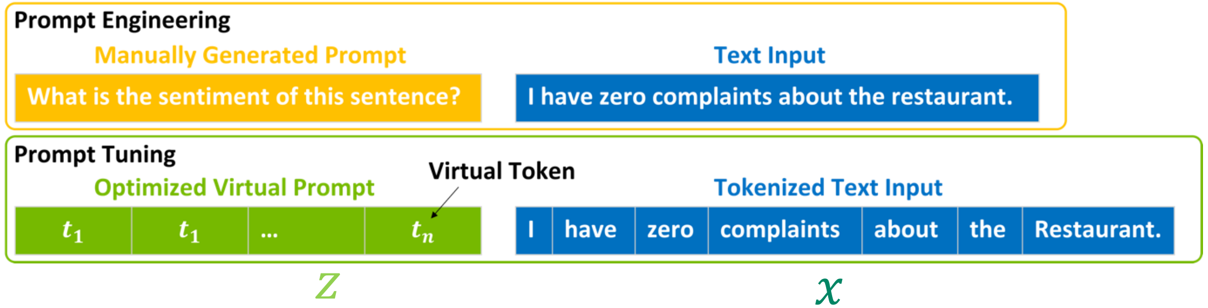
[Jingwei Sun et al. ICML 2024, arXiv:2310.01467]

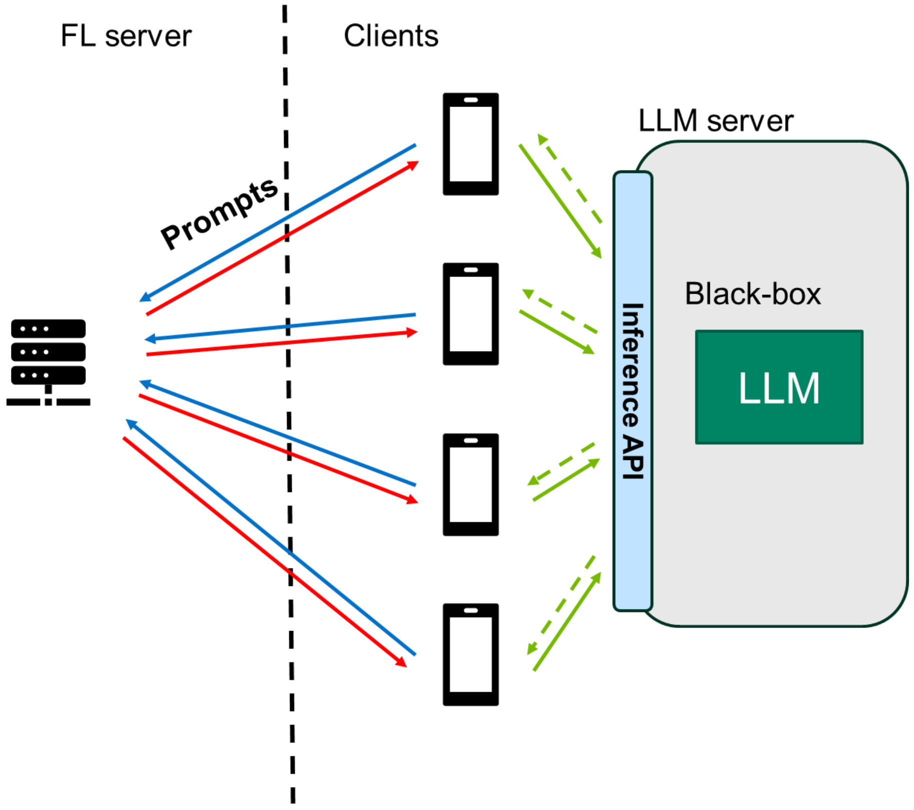
Unique Challenge in the Age of LLM

Efficient Federated Black-box Prompt Tuning

- The clients train prompts while treating the LLM as a black-box model.
- The clients only conduct inference (with LLM server) without back-propagation.
- The clients upload and download prompts (to and from FL server) rather than the whole model.

- Learning an optimized prompt encoding scheme in a federated fashion – prompt tuning
- Utilizing gradient-free optimization methods





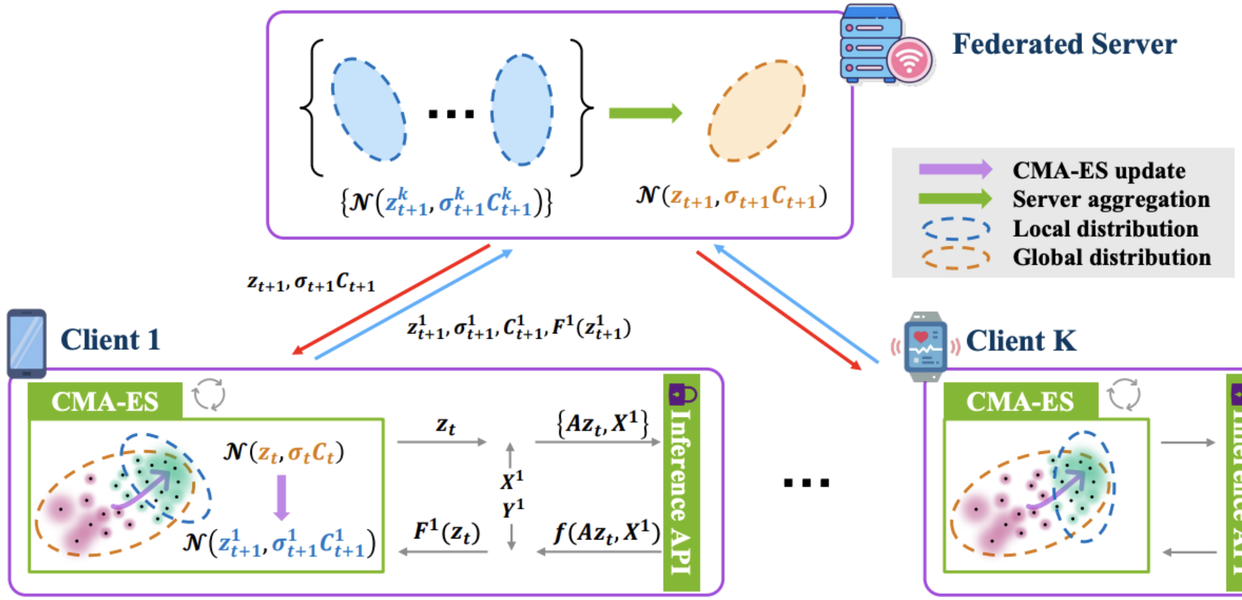
Dimension of z is much lower than Θ

[Jingwei Sun et al. ICML 2024, arXiv:2310.01467]

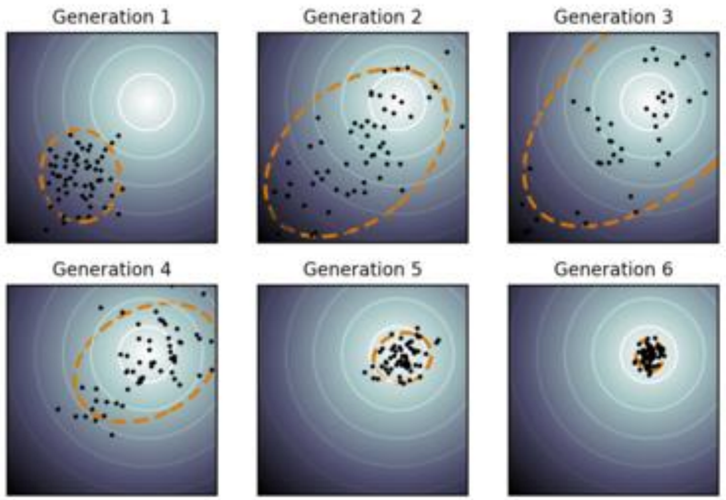
FedBPT: Gradient-free Optimization

Federated global estimation

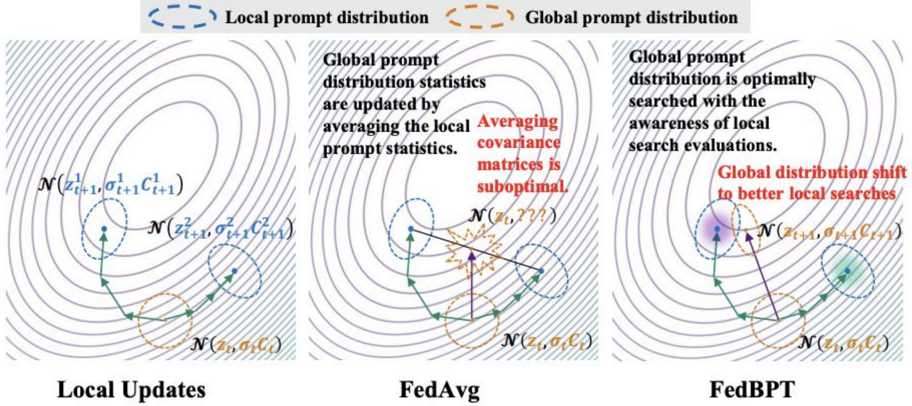
- Use evolution strategy for optimization over the optimal prompt distribution
- Local optimization with inference only (“gradient-free”)
- Rather than simple averaging, global prompt distribution is searched with awareness of local prompt distributions
- Gradually converge to optimal solution



Inference is all you need!!!



Evolution Strategy (CMA-ES)



[Jingwei Sun et al. ICML 2024, arXiv:2310.01467]

Results

RoBERTa-350M

- Compare both gradient-based and gradient-free methods
- Achieves higher accuracies than gradient-free baselines under IID and non-IID settings for all the datasets.
- Decent performance among gradient-based methods
- Simply combining FedAvg and BBT cannot achieve decent performance

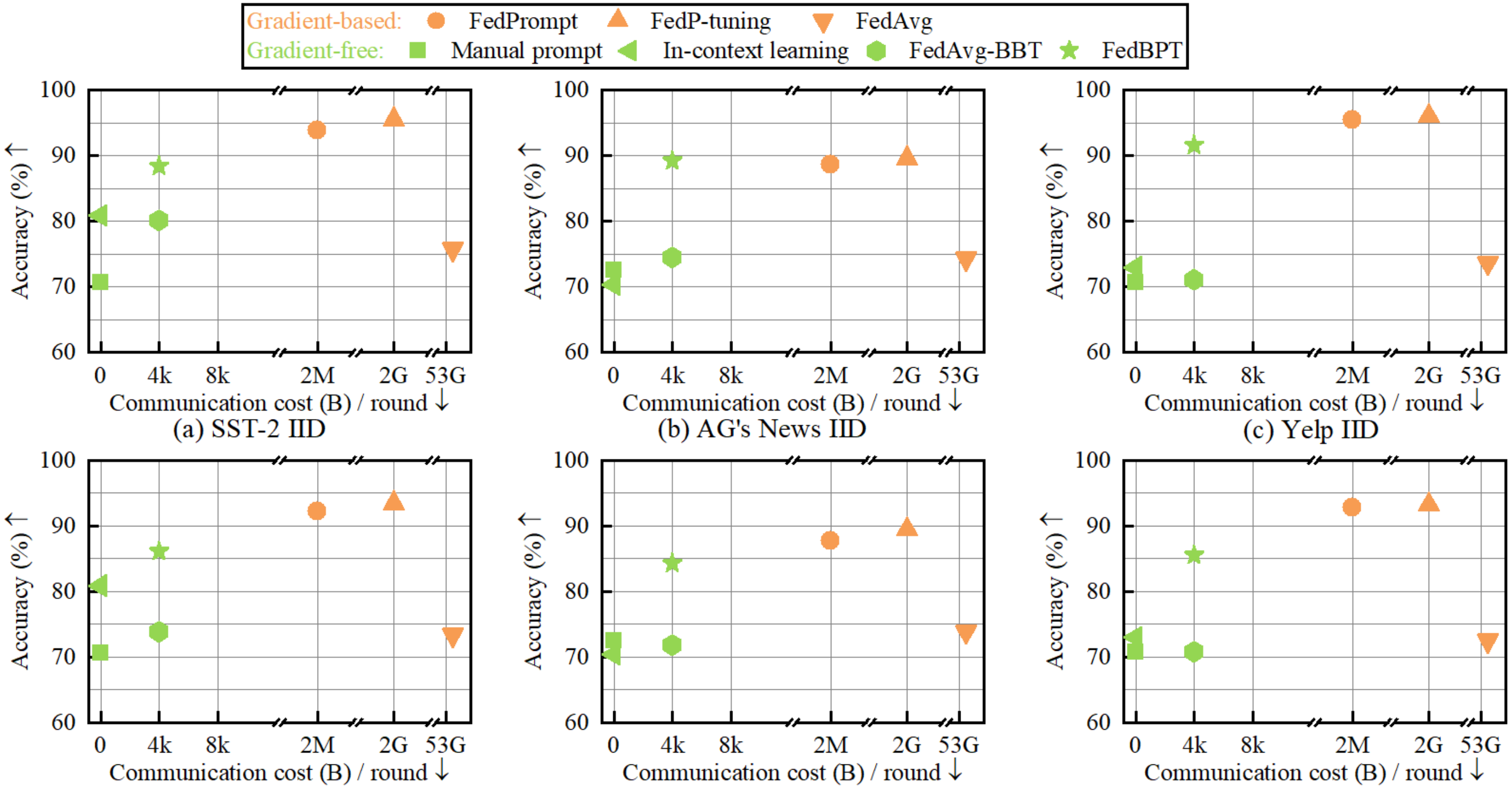
Method	Trainable Params.	SST-2 Acc.(%) IID	SST-2 Acc.(%) non-IID	AG's NEWS Acc.(%) IID	AG's NEWS Acc.(%) non-IID	Yelp Acc.(%) IID	Yelp Acc.(%) non-IID
Gradient-based methods							
FedPrompt	51K	90.25	85.55	87.72	85.62	91.44	91.47
FedP-tuning	15M	**90.6**	**87.16**	**88.17**	**86.11**	**93.61**	**91.63**
FedAvg	355M	84.7	82.4	77.43	76.54	88.25	88.03
FedLoRA	786K	84.6	84.53	77.85	75.9	88.52	88.2
Gradient-free methods							
Manual prompt	0	83.6		75.75		88.37	
In-Context Learning	0	79.7		76.96		89.65	
FedAvg-BBT	500	84.45	84.17	76.54	76.46	89.64	89.72
FedBPT	*500*	***87.16***	***86.47***	***82.36***	***81.03***	***91.12***	***90.8***

[Jingwei Sun et al. ICML 2024, arXiv:2310.01467]

Results

Llama2-7B

- Improve the accuracy significantly compared with the gradient-free baselines and achieve comparable accuracies with the gradient-based methods in most settings
- Reduce the communication cost of one device in one round from nearly 2GB to 4KB (against P-tuning).



(a) SST-2 IID (b) AG's News IID (c) Yelp IID
(d) SST-2 non-IID (e) AG's News non-IID (f) Yelp non-IID

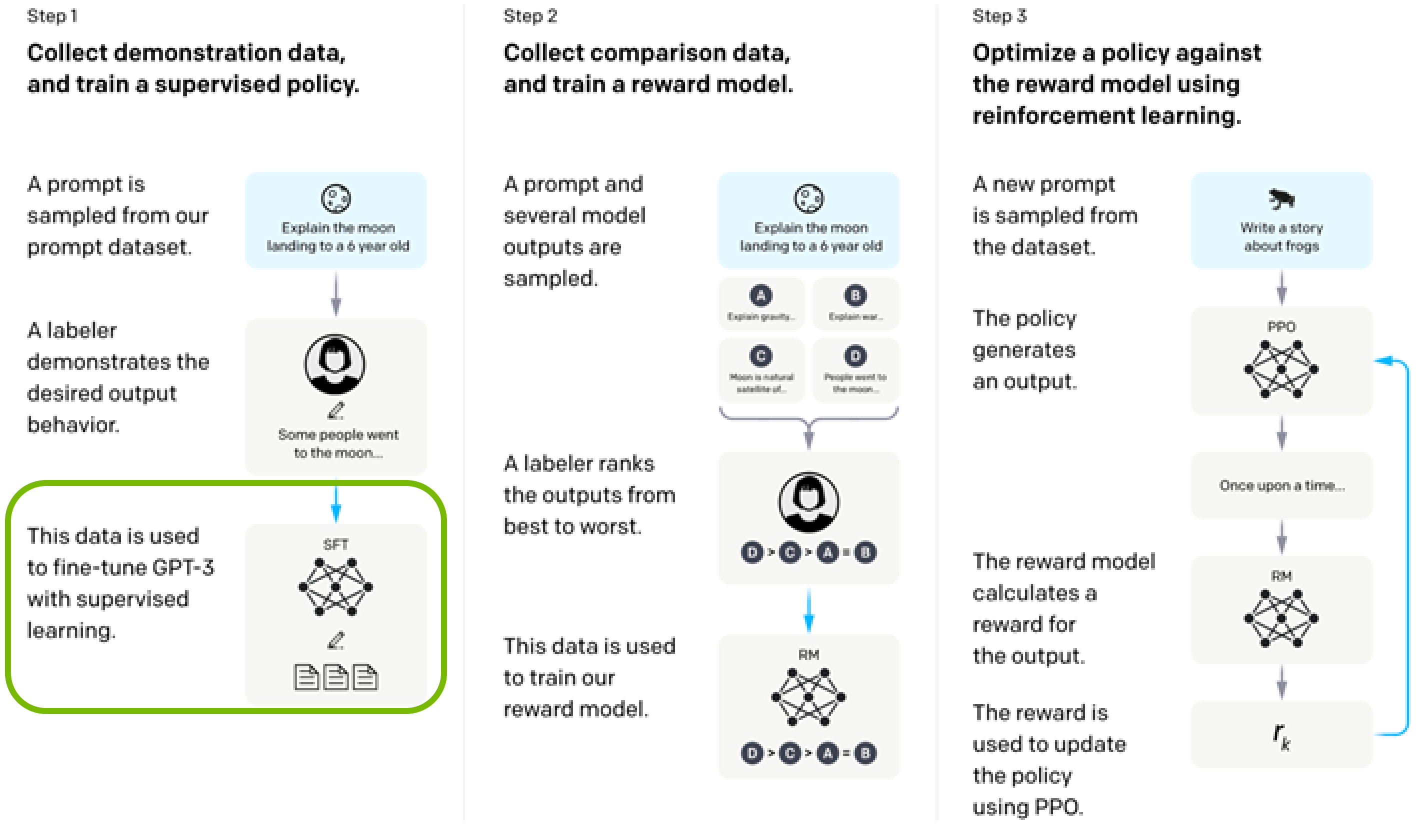
[Jingwei Sun et al. ICML 2024, arXiv:2310.01467]

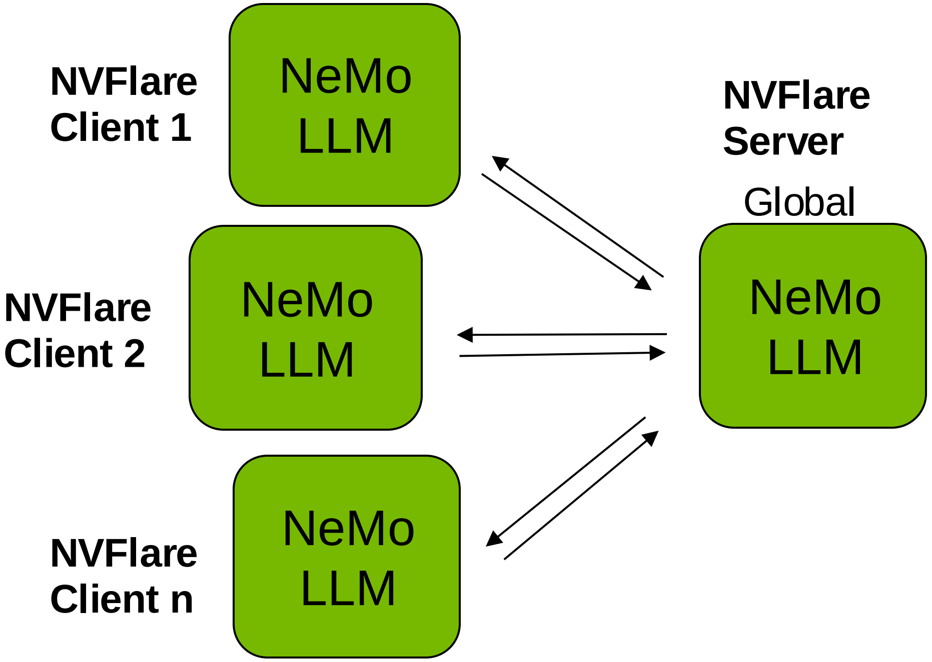
Supervised Fine-tuning

Supervised Fine-tuning (SFT)

Towards "instruction-following" LLM

The first step of "Chat-GPT training scheme". Unlike PEFT, SFT finetunes the entire network.

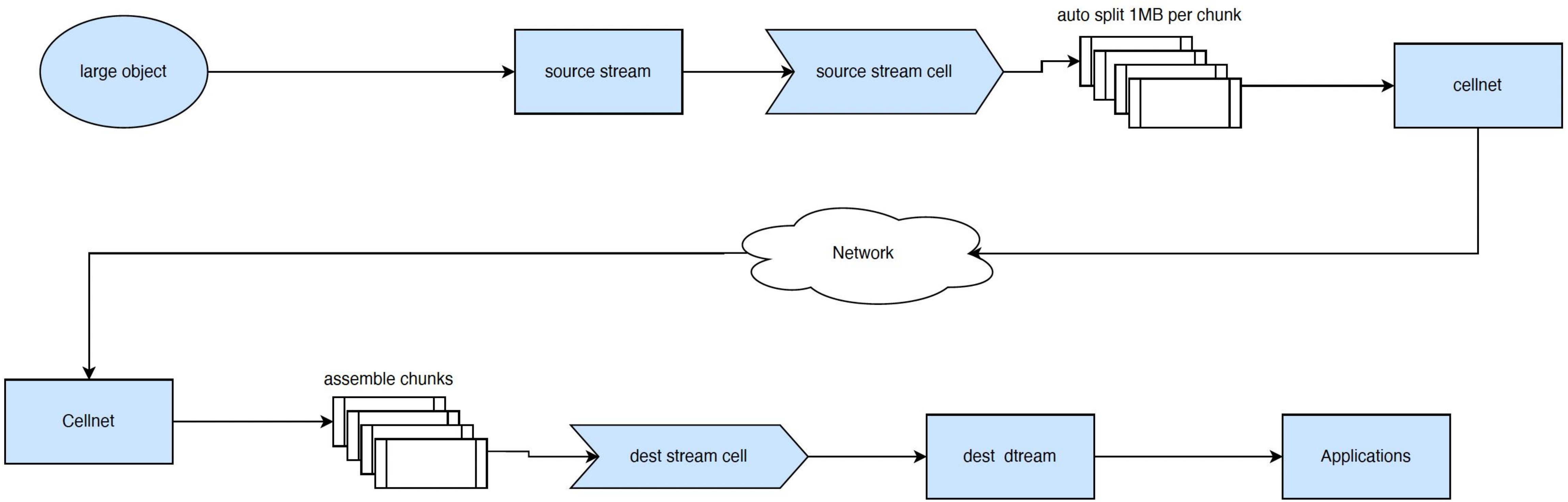




NVFlare Streaming

Support Large Model Transmission

- Model size of mainstream LLM can be huge: 7B -> 26 GB (beyond the 2 GB GRPC limit)
- In order to transmit LLMs in SFT, NVFlare supports **large object** streaming



Documentation: https://nvflare.readthedocs.io/en/main/real_world_fl/notes_on_large_models.html

NVIDIA

SFT for Instruction Following

3 open datasets

We use three datasets:

- Alpaca
- databricks-dolly-15k
- OpenAssistant

Containing instruction-following data in different formats under different settings:

- oasst1 features a tree struture for full conversations
- other two are instruction(w/ or w/o context)-response pairs

Examples with NeMo and HuggingFace
https://github.com/NVIDIA/NVFlare/tree/main/integration/nemo/examples/supervised_fine_tuning
https://github.com/NVIDIA/NVFlare/tree/main/examples/advanced/llm_hf

SFT With FL

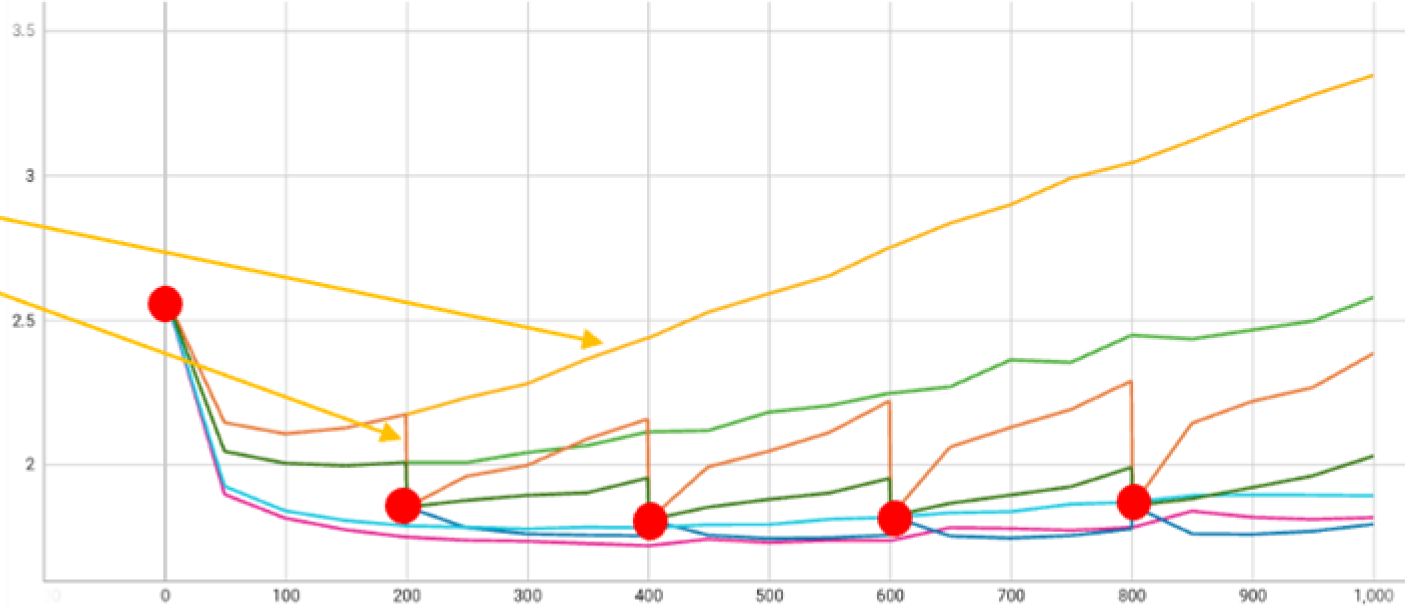
Achieving better performance

NeMo 1.3B model, SFT for 5 rounds

5 experiments in total: training on each client's own dataset, on combined dataset, and all three clients training together using the FedAvg algorithm implemented in NVFlare.

Validation loss curves:

- yellow: oasst1
- green: dolly
- blue: alpaca
- magenta: three datasets combined



Smooth curves for local training, “Step curves” for FL - because of global model sync and update

SFT Model Evaluation

LLM Performance

Non-trivial task compared with “fixed downstream tasks” where we usually have metrics like accuracy, F-1 scores, etc.

Common practice is to test the resulting LLMs on **benchmark tasks**, and/or human evaluations

We perform standard language modeling tasks under zero-shot setting, including HellaSwag(H), PIQA(P), and WinoGrande(W)

BaseModel - Before SFT

	H_acc	H_acc _norm	P_acc	P_acc _norm	W_acc	Mean
BaseModel	0.357	0.439	0.683	0.689	0.537	0.541
Alpaca	0.372	0.451	0.675	0.687	0.550	0.547
Dolly	0.376	**0.474**	0.671	0.667	0.529	0.543
Oasst1	0.370	0.452	0.657	0.655	0.506	0.528
Combined	0.370	0.453	0.685	**0.690**	0.548	0.549
FedAvg	**0.377**	0.469	**0.688**	0.687	**0.560**	**0.556**

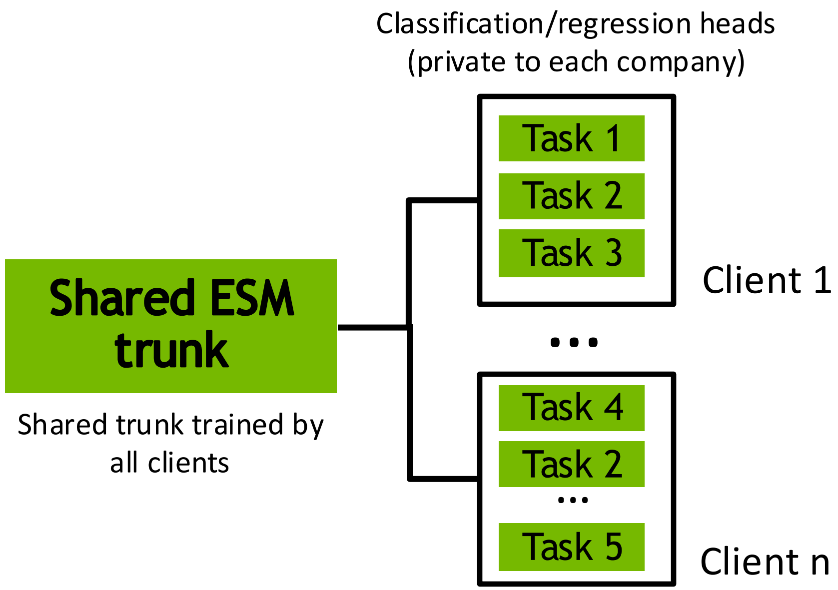
Table 1. Model performance on three benchmark tasks: HellaSwag (H), PIQA (P), and WinoGrande (W)

Blog: https://developer.nvidia.com/blog/scalable-federated-learning-with-nvidia-flare-for-enhanced-llm-performance

NVIDIA

Biopharma Applications

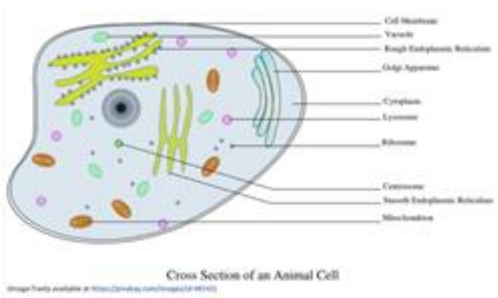
BioNeMo ESM Fine-tuning

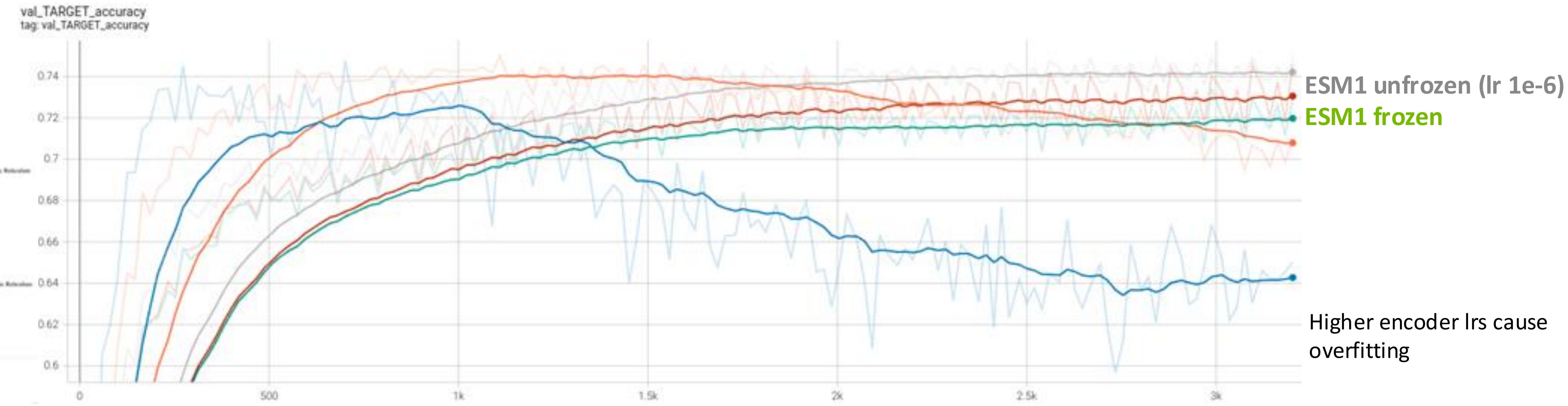


Model	Params	Frozen Encoder	Validation Accuracy
ESM1nv	43.6 M	Yes	0.7344
ESM1nv	43.6 M	No	0.7488
ESM2nv	650M	Yes	0.7883
ESM2nv	650M	No	**0.7919**

ESM-2 is a pre-trained, bi-directional encoder (BERT-style model) over amino acid sequences.

Subcellular localization task (centralized)
6918 training and 1700 testing proteins

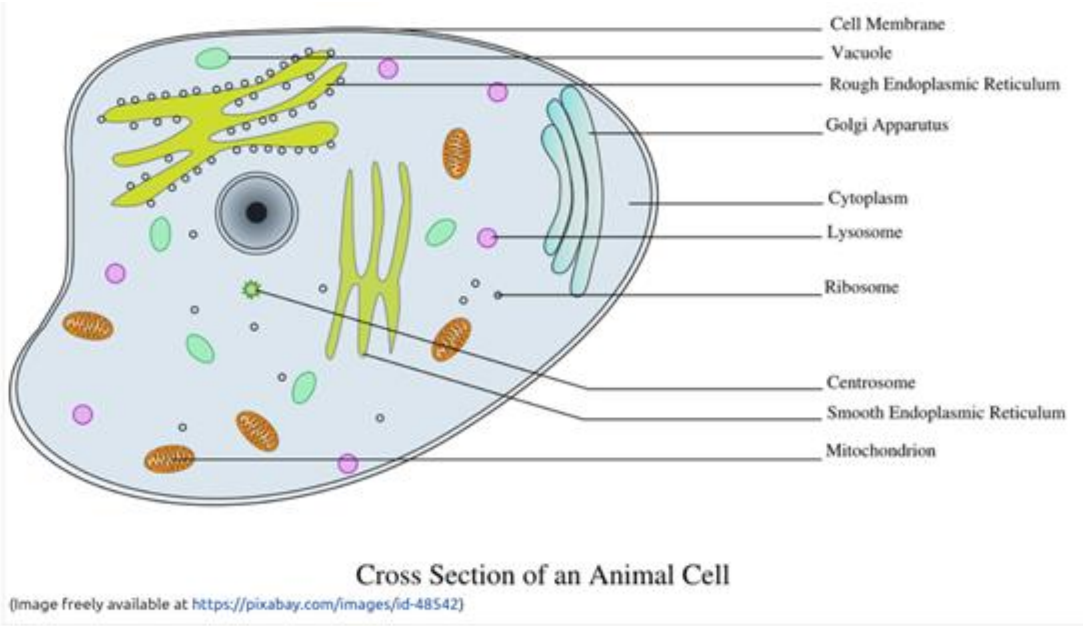


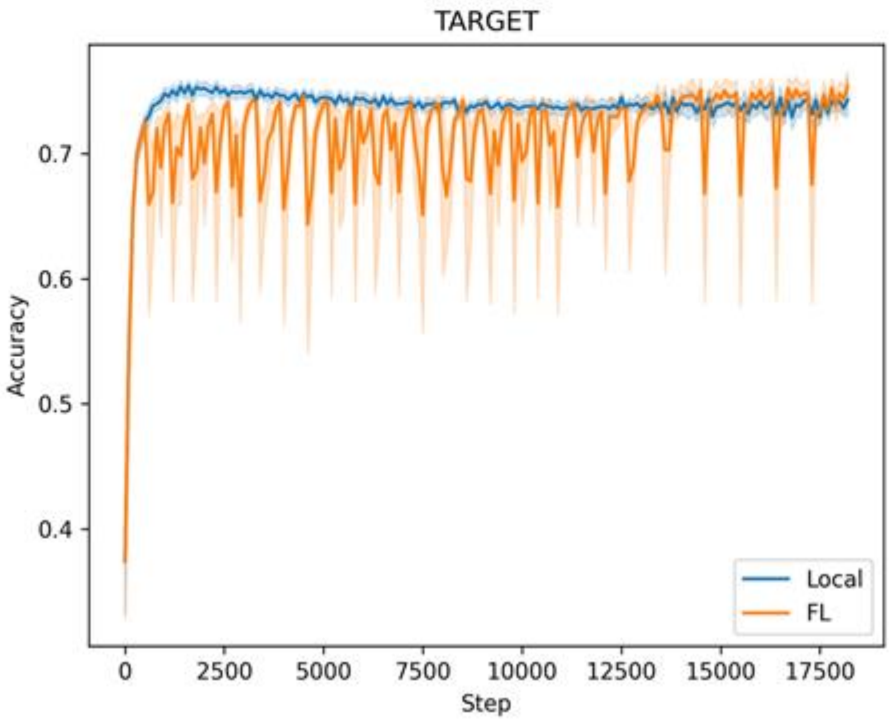


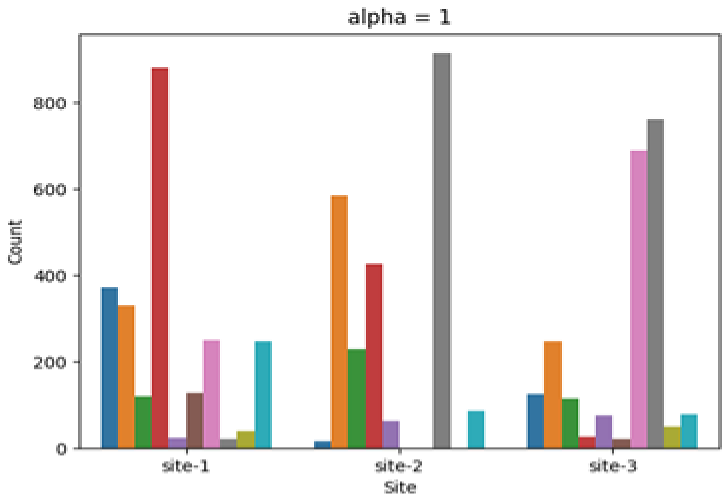
Higher encoder lrs cause overfitting

FedAvg With ESM2nv 650M

Subcellular Location Prediction







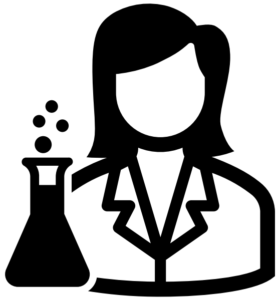
- 3 clients
- alpha 1.0
- 20 rounds/epochs
- ~2300 sequences per client
- Frozen ESM encoder

Setting	Accuracy
Local	0.773
FL	**0.776**

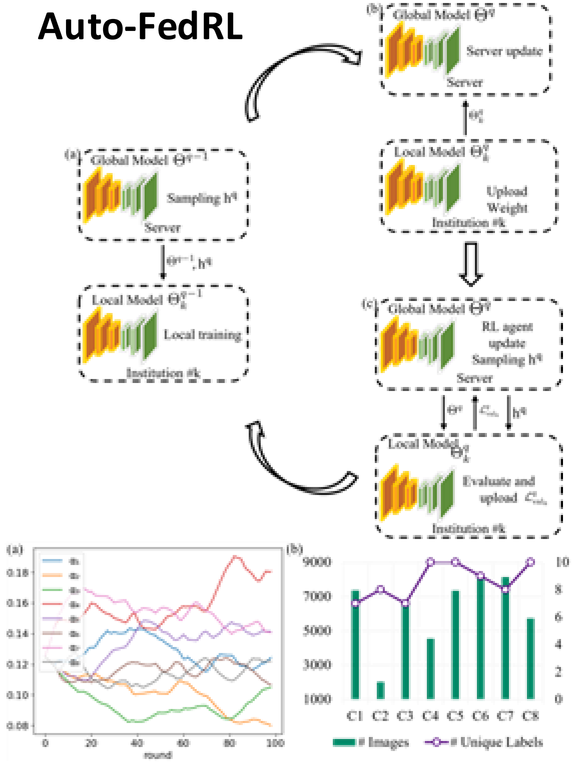
More examples: https://github.com/NVIDIA/NVFlare/tree/main/examples/advanced/bionemo

Research With NVIDIA FLARE

https://github.com/NVIDIA/NVFlare/tree/dev/research

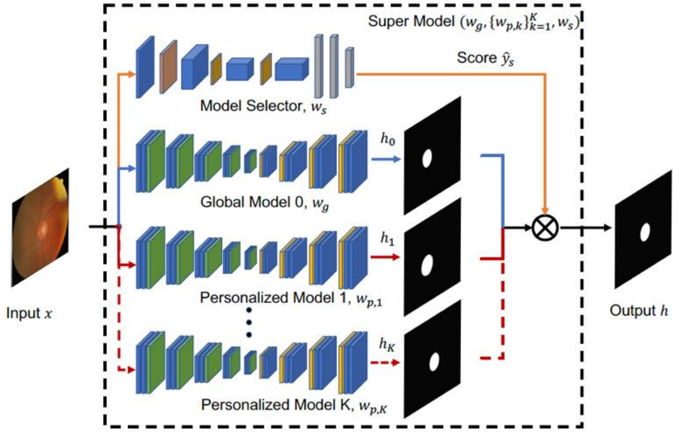


Auto-FedRL



P. Guo et al. ECCV 2022

FedSM: Personalized FL

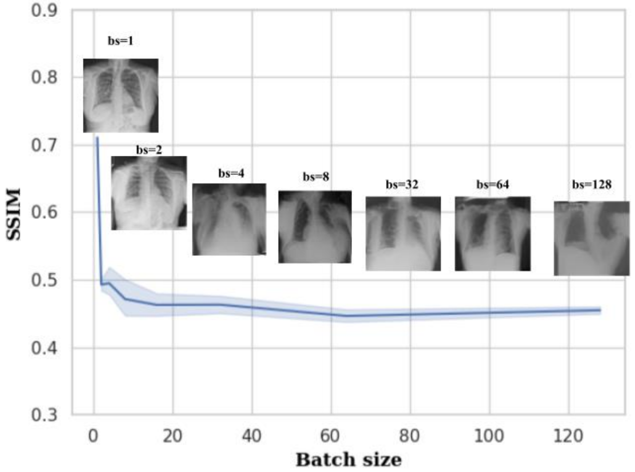


A. Xu et al. CVPR 2022

Baseline Implementations

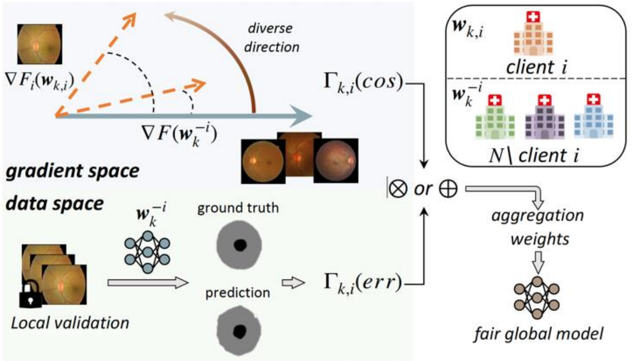
- FedAvg
- FedProx
- FedOpt
- SCAFFOLD
- Ditto
- Cyclic Weight Transfer
- Swarm Learning

Quantifying data leakage



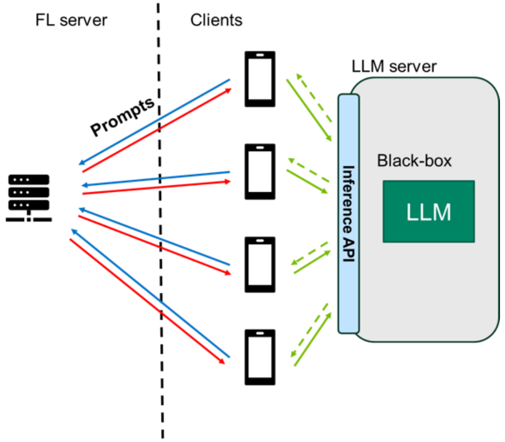
A. Hatamizadeh et al. TMI 2022

FedCE: Contribution Estimation



M. Jiang et al. CVPR 2023

Federated Black-Box Prompt Tuning



Agenda

- What is Federated Learning?
- NVIDIA Key Technologies for Federated Learning
- Real-world Use cases of Federated Learning
- Getting Started with NVIDIA FLARE
- Research: Addressing Key challenges in Federated Learning
- **Summary & Announcements**

NVIDIA FLARE: Summary

A domain-agnostic, open-source, extensible FL framework

- **Federated Computing** -- a federated computing framework at core
- **End-to-End Confidential Federated AI –** combination of confidential computing and additional technologies to secure end-to-end process
- **Built for productivity** -- designed for maximum productivity, providing a range of tools to enhance user experience
- **Built for security & privacy** -- prioritizes robust security and privacy preservation
- **Built for concurrency & scalability** -- designed for concurrency, supporting resource-based multi-job execution
- **Built for customization** -- structured in layers, with each layer composed of customizable components
- **Built for integration** -- multiple integration options with third-party system
- **Built for production** -- robust, production-scale deployment in real-world federated learning and computing scenarios
- **Rich examples repository** -- wealth of built-in implementations, tutorials and examples
- **Growing application categories --** medical imaging, medical devices, edge device application, financial services, HPC and autonomous driving vehicles

GitHub : https://github.com/NVIDIA/NVFlare

Web: https://nvidia.github.io/NVFlare/

Get Involved!

NVIDIA FLARE is open-source

NVIDIA FLARE DAY 2025

Exploring Real-world Examples of Federated Learning

September 2025 (US + EMEA)
September 2025 (APAC)

2024 Recordings available:
https://nvidia.github.io/NVFlare/flareDay

NVIDIA Academic Grant Program for Researchers:

Edge AI and Federated Learning

https://www.nvidia.com/en-us/industries/higher-education-research/academic-grant-program

NVIDIA FLARE GitHub : https://github.com/NVIDIA/NVFlare

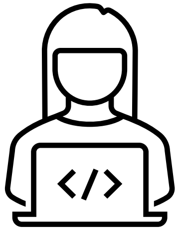
Web Page : https://nvidia.github.io/NVFlare/

NVIDIA FLARE Summer of Code 2025

Select contributions will feature at NVIDIA FLARE DAY in September

Step 1: Register a Proposal

Register Now



Step 2: Develop your Contribution

- Work on your solution following the project's coding standards and best practices.
- Regularly engage with maintainers and the community for feedback.
- Test your code thoroughly and document it properly.

Step 3: Submit a Pull Request

- Fork the NVFlare repository and create a branch for your contribution.
- Submit a clear and well-documented PR with:
  - A description of your solution
  - Relevant benchmarks or evaluations
  - Any necessary instructions for running/testing the contribution
- Label your PR with "NVFlare Summer of Code" to ensure visibility.
- For more info, see the CONTRIBUTION GUIDE.

Step 4: Review and Iterate

- Address feedback from the maintainers and community.
- Refine your work based on suggestions.

Step 5: Final Submission and Selection

- By September 1, 2025, ensure your PR is merged or under final review.
- Select contributions will be invited to present at NVIDIA FLARE DAY in September 2025 (5-min Lightning Talk).

Focus Areas

We welcome contributions in the following areas:

- **Energy-Efficient Communication & Learning** (e.g., compression, pruning, parameter-efficient finetuning, asynchronous FL, carbon footprint tracking)
- **Privacy-Preserving Techniques in the Age of LLMs** (e.g., differential privacy, memorization prevention, privacy attacks & defenses)
- **Training on the Edge (Real-Time & Multi-Sensor Data)** (e.g., mobile, automotive, robotics, surgery, dynamic sensor networks)
- **Collaborative Inference & Information Aggregation** (e.g., federated RAG, multi-agent learning, decentralized inference)

Contribution Types

We are looking for the following contributions:

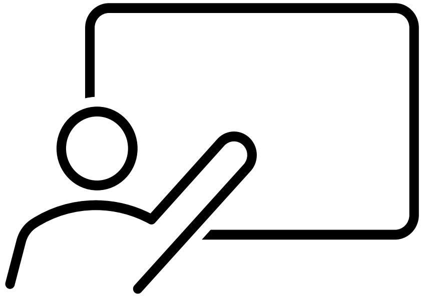
- **Algorithms** – Novel methods to improve federated learning efficiency, scalability, and security
- **Core Framework & Utilities** – Enhancements to NVFlare's core functionalities
- **Examples & Use Cases** – Demonstrations of FL applications in real-world scenarios

For details, see https://github.com/NVIDIA/NVFlare/discussions/3238

NVIDIA FLARE Webinars

2025

- **Q1: Introduction to NVIDIA Federated Learning: Concepts, Technology, and Use Cases**
  Kick off the series with a comprehensive dive into federated learning fundamentals, the technology that powers it, and real-world use cases.
- **Q2: Federated Learning Applications: Healthcare and Finance**
  Explore how federated learning is transforming two of the most critical industries, driving innovation while preserving data privacy.
- **Q3: Provision, Run, and Monitor Federated Learning Applications**
  How to setting up, executing, and managing federated learning system
- **Q4: Real-world Federated Learning Use Cases from Industries**
  Delve deeper into how federated learning is shaping advanced use cases across diverse industries.



NVIDIA

Thank you!

Questions?

Holger Roth <hroth@nvidia.com>