**Vector Institute** **04/15/2024**

# Accelerating Federated Learning and Improving Generalization with Heterogeneous Real-World Data

Marco Ciccone

Polytechnic University of Turin

## ABOUT ME - BACKGROUND

**ELLIS Postdoctoral Researcher**

**Research Interests**

- Continual and Multitask Learning
- Decentralized Learning
- Modular architectures
- Computer Vision, Multimodal AI

# *Students and collaborators*

Debora Caldarola

Eros Fanì

Riccardo Zaccone

Carlo Masone

# My big question

*How can independent learners,*
*optimizing their own objective,*
*cooperate and share their knowledge to solve a shared goal?*

*Cooperative MARL*

Multi-Agent Coordination in Adversarial Environments through Signal Mediated Strategies, AAMAS 2021

Coordination in adversarial sequential team games via multi-agent deep reinforcement learning

A Marriage between Adversarial Team Games and 2-player Games: Enabling Abstractions, No-regret Learning, and Subgame Solving, ICML 2022

*Transfer Learning with Neural Nets*

***My*** *main research motivation is Transfer and Collaborative Learning*

## *Federated Learning*

# Population of users join a “federation” and collaborate towards learning a model together

🔒

**Privacy is important**

Data never leaves the device

🏠

Training on data is local and independent

📡

Communicate model updates

💪

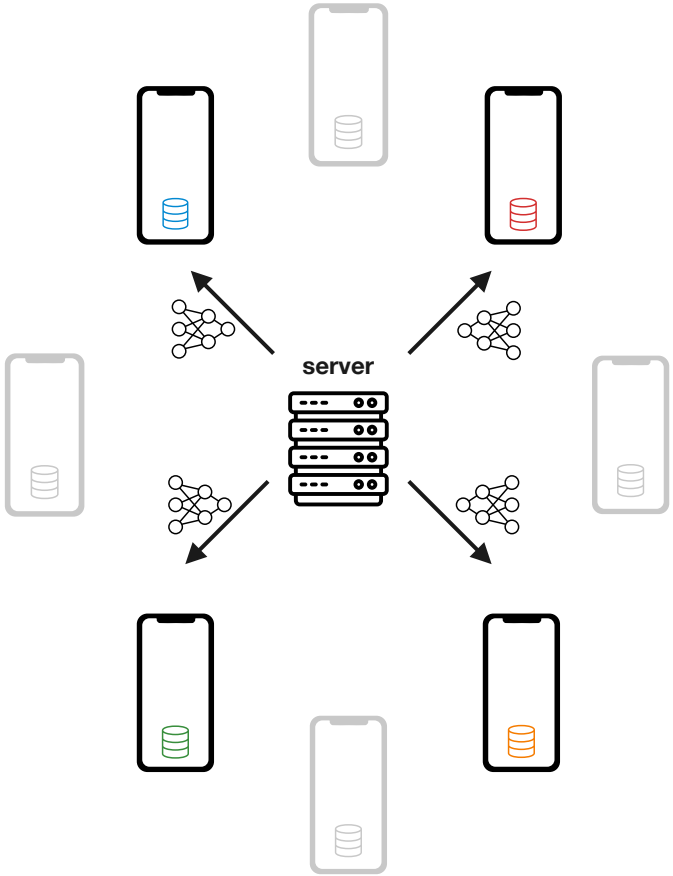
Powerful, but many challenges.

McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." *AISTATS*, 2017.

# Federated Training

## Optimization Round

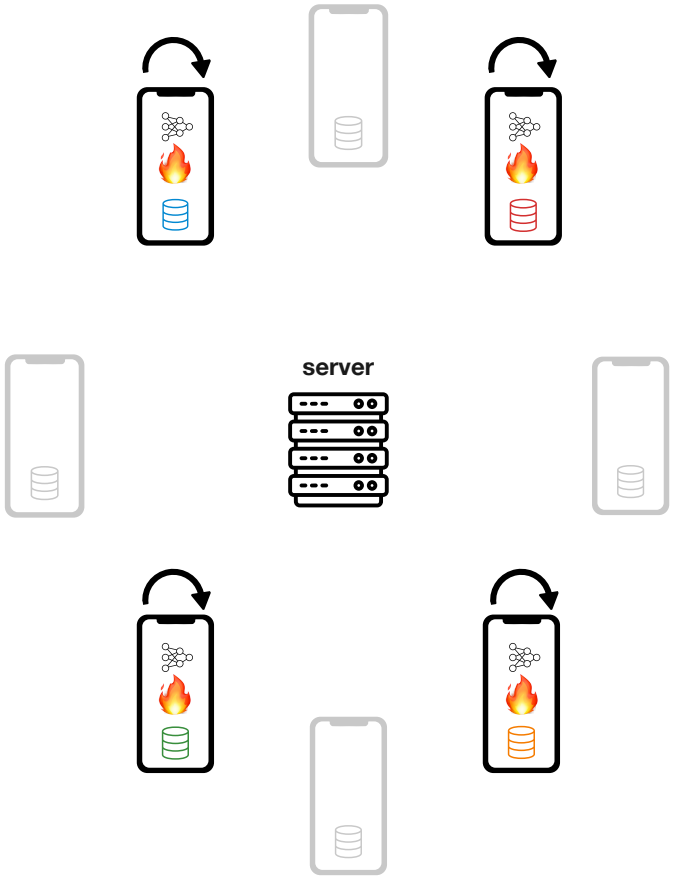
- Sample active clients
- Communicate current model to clients



# Federated Training

**Optimization Round**

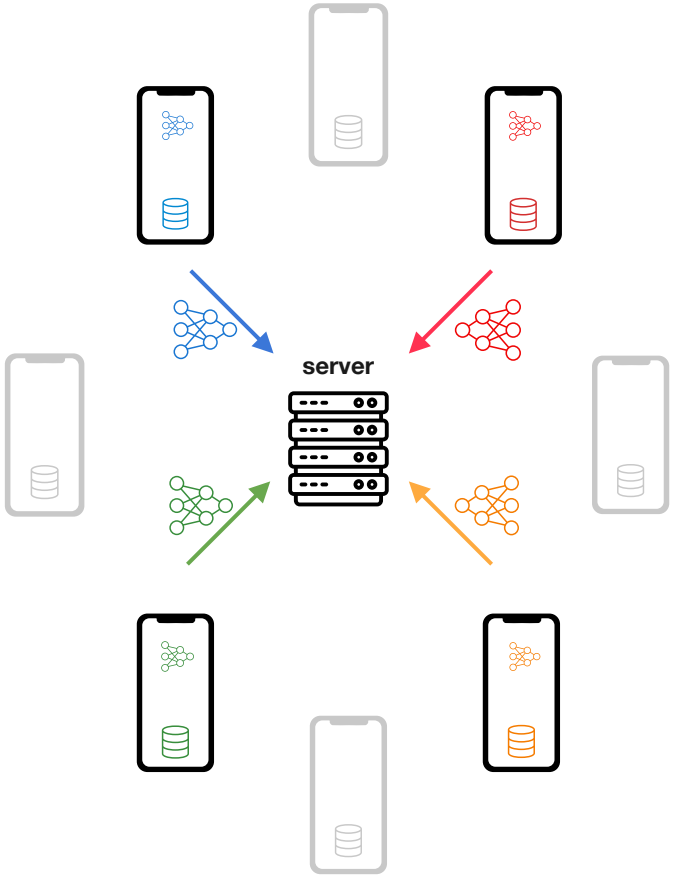
- Sample active clients
- Communicate current model to clients
- **Local Optimization (independent - in parallel)**
    - **Several gradient steps locally**



# Federated Training

**Optimization Round**

- Sample active clients
- Communicate current model to clients
- Local Optimization (independent - in parallel)
    - Several gradient steps locally
- **Communicate local models to server**



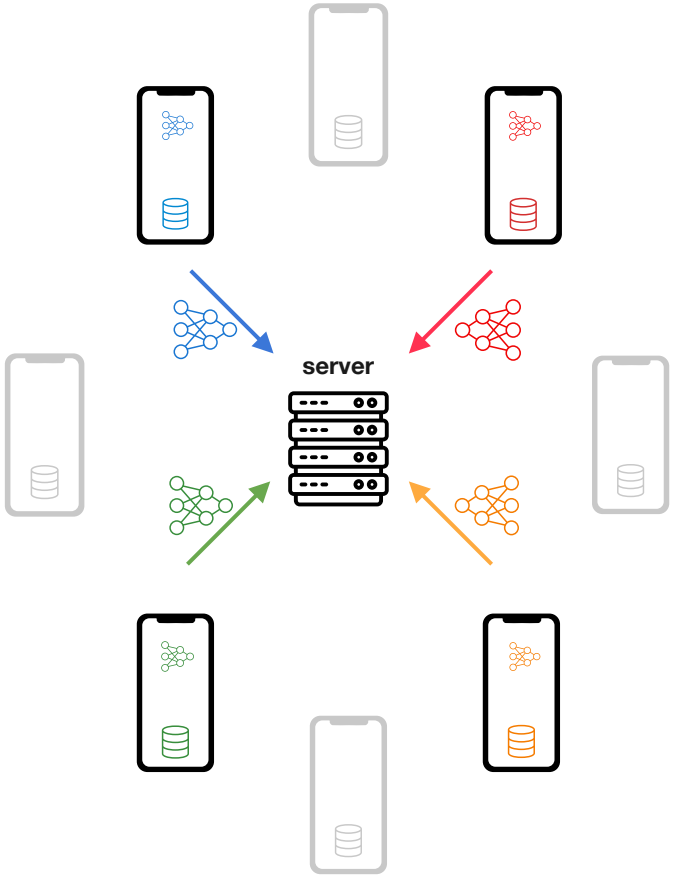
# Federated Training

**Optimization Round**

- Sample active clients
- Communicate current model to clients
- Local Optimization (independent - in parallel)
  - Several gradient steps locally
- Communicate local models to server
- **Aggregate local models on server**

***Repeat***

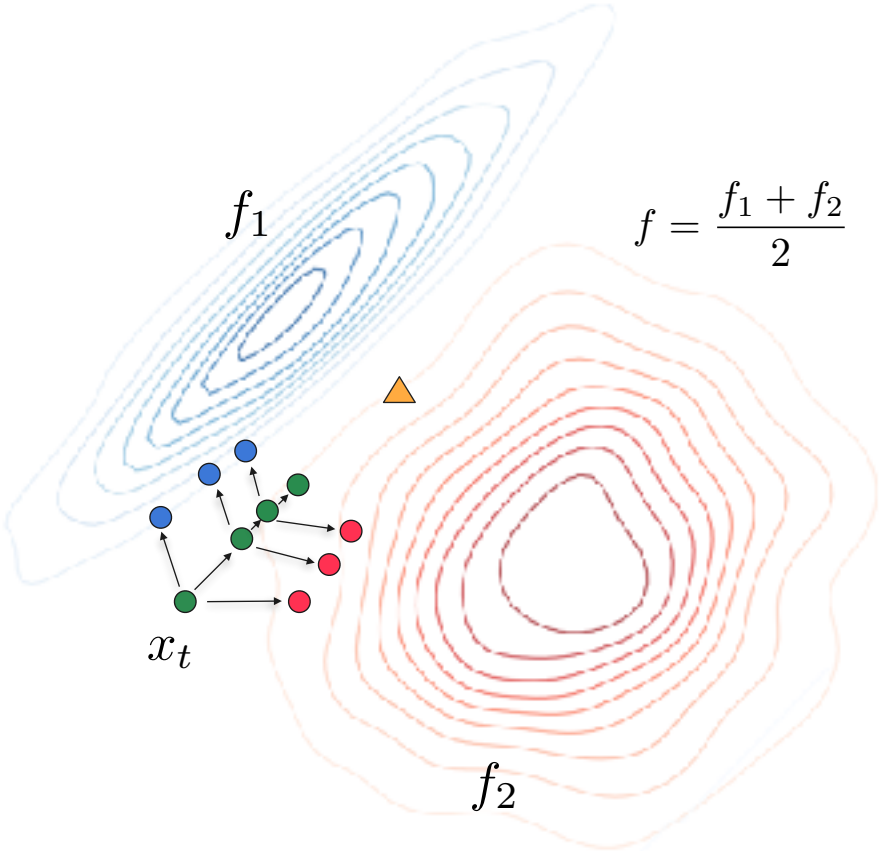
FedAVG $\theta^{t+1} = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \theta_i^t$



# Federated Learning
# a.k.a. Local SGD

The idea behind FL is **to trade off communication and local computation**

Reduce the number of synchronisations and communication cost by **running more local optimization steps.**




Slides credit, Karimireddy Scaffold presentation

# The issue of statistical heterogeneity

*Samples collected on edge devices are biased and correlate to specific user habits, preferences and location*

- **Domain** Shift
- **Label** Skewness
- **Size** Skewness (imbalanced local datasets)

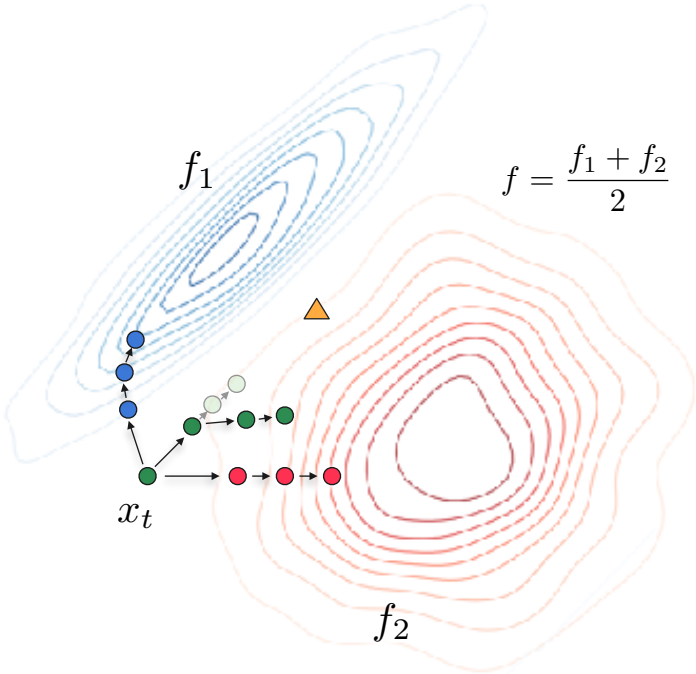
# Breaking the IID assumption in FL

**1) Client drift problem**

- ***Local training is biased*** towards local data
- Inherent **objectives inconsistency**

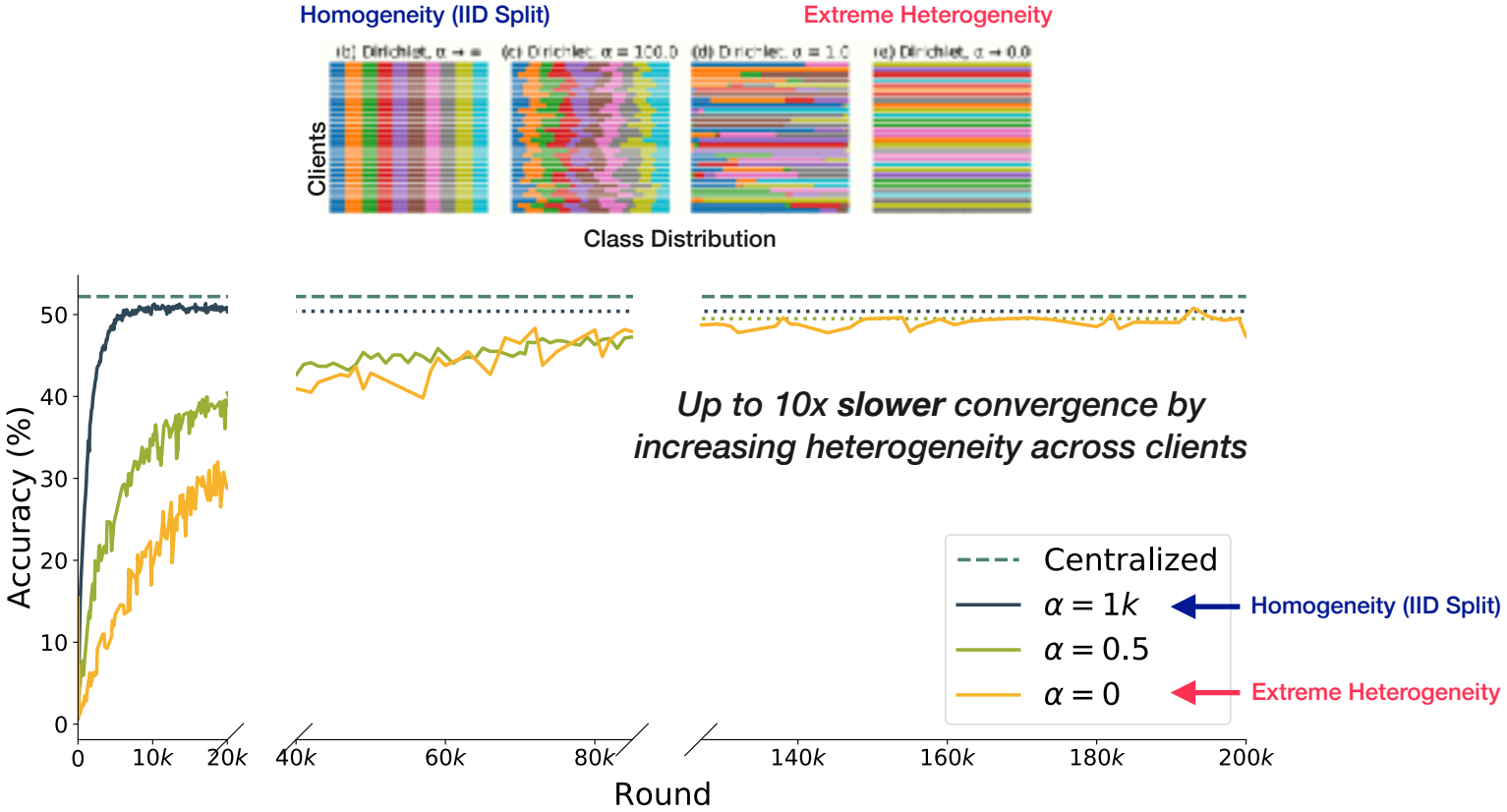
**2) Partial Participation**

- ***Only a portion of clients is available*** at each round
- Global training trajectory can be noisy




*Non-IID sampling causes **slow convergence** and **poor performance***

# Effect of heterogeneity on FedAVG convergence speed (CIFAR100)



# Research Objectives

*Mitigate the issues of statistical heterogeneity*

⚡

**Improving generalization performance** — close the gap with centralised training

**Speeding up convergence** by reducing communication rounds

👀

**Minimizing** communication cost

🧿

**Enabling** real-world applications

QUESTION 1:

# Can we improve generalization to get better global performance?

- We propose methods for improving generalization in FL
- Acting both on **client** and **server** sides


*Improving Generalization in Federated Learning by Seeking Flat Minima*

D. Caldarola*, B. Caputo, **M.Ciccone***

## Our hypothesis

*Generalization performance of FL algorithms are bounded by the quality of the local models.*

Clients may suffer of **poor generalization**, exacerbated by **heterogeneity** and **small local datasets** with limited sample diversity.

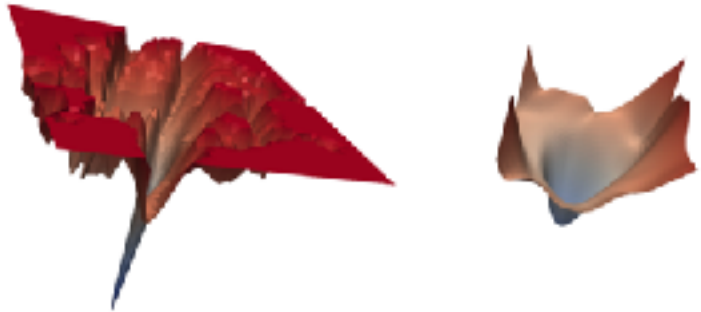
### 💡 Idea

*Improve local models quality by searching for better, more generalisable solutions*

# In search of generalization

Not all minima are created equal.

**Flat minima** show strong empirical **correlation with generalization** in neural networks



Design algorithms that **search for flat minima** while optimising for performance

- ***SAM:*** *Sharpness Aware Minimization*
- ***SWA:*** *Stochastic Weight Averaging*


Keskar, Nitish Shirish, et al. "On large-batch training for deep learning: Generalization gap and sharp minima." *ICLR 2016*

Foret, Pierre, et al. "Sharpness-aware minimization for efficiently improving generalization." *ICLR 2021*

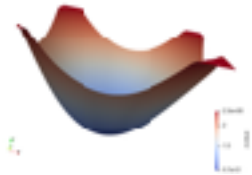
Jiang, Yiding, et al. "Fantastic generalization measures and where to find them." ICLR 2020

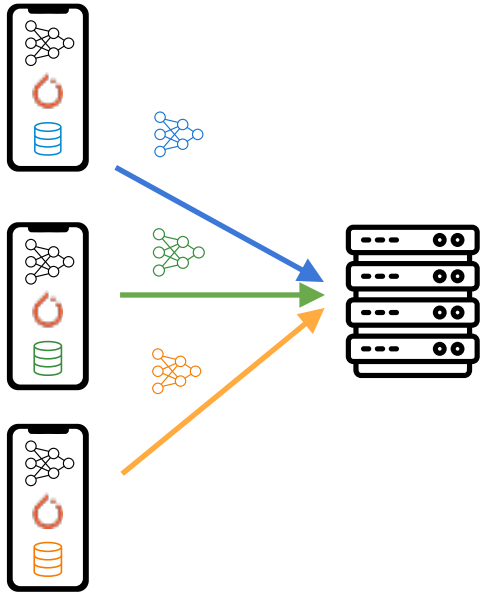
Izmailov, Pavel, et al. "Averaging weights leads to wider optima and better generalization." *UAI 2018*

# Finding Flat minima in Federated Learning

***Client side***

**Locally** train with **SAM**
to converge to **flat minima**





***Server side***

$$\text{global} = \left( \text{client}_1 + \text{client}_2 + \text{client}_3 \right)$$

Update global model with
FedAVG

**ensemble** global models
with **SWA**

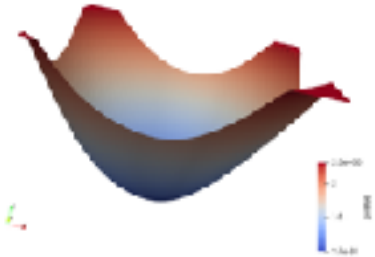
$$\theta_{\text{SWA}} \leftarrow \frac{\theta_{\text{SWA}} \cdot n_{models} + \theta}{n_{models} + 1}$$

*Find better*
*local models*

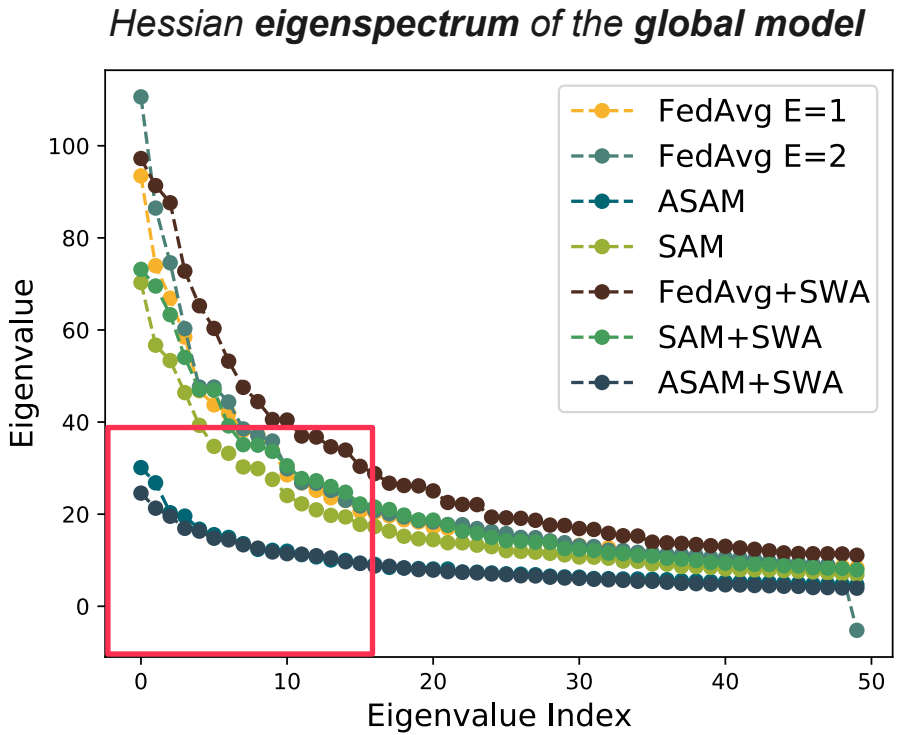
*Aggregate **multiple global models** to*
*mitigate the noise of partial*
*participation*

# *Local SAM effectively reaches flat minima*

*FedAVG*



*FedSAM*



# Results

Centralized: **52.20** - FedAvg: **30.25** ← BASELINES

| Algorithm | Accuracy | | Absolute Improvement | |
|---|---|---|---|---|
| | Centr. | $\alpha = 0$ | Centr. | $\alpha = 0$ |
| SAM | 55.22 | 31.04 | +3.02 | +0.79 |
| ASAM | 55.66 | 36.04 | +3.46 | +5.79 |
| SWA | 52.72 | 39.34 | +0.52 | +9.09 |
| SAM + SWA | 55.75 | 39.30 | +2.55 | +9.05 |
| ASAM + SWA | 55.96 | 42.01 | +3.76 | **+11.76** |
| Mixup | 58.01 | 29.91 | +5.81 | -0.34 |
| Cutout | 55.30 | 24.24 | +3.10 | -6.01 |

- Flat Minima are more beneficial in FL than in centralized setting
- **Data Augmentation fails** in heterogeneous settings and needs to be reconsidered in FL

*is local regularisation enough for better convergence?*

**FedSAM can be used with other techniques to improve convergence speed**

Dai, Rong, et al. **FedGAMMA** IEEE TNNLS 2023

Sun, Yan, et al. **Fedspeed** ICLR 2023

Sun, Yan, et al. **FedSMOO**, ICML 2023

QUESTION 2:

# Can we accelerate convergence and be communication efficient?

**under review**

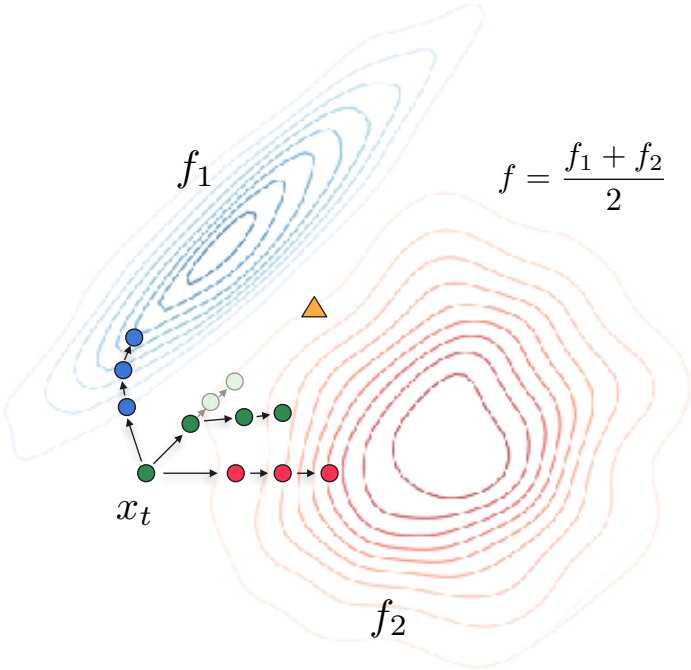
*Communication-Efficient Heterogeneous Federated Learning with Generalized Heavy-Ball Momentum*

R. Zaccone, C. Masone, **M. Ciccone**

# Client-drift problem

**Heterogeneity causes the local optimization to *drift* from the global minimum**

- Global convergence is much slower
- May not converge to the true optimum




*Local learners need to know more about the global distribution.*

Slides credit, Karimireddy Scaffold presentation

## *Communication-Efficient Local Momentum*

### 💡 Idea

*1) Correct the local training trajectory with a* ***local momentum*** *term that maintains the global optimization direction*

*2) Compute the correction directly on the client*

## *Generalised Heavy-Ball (GHB) Momentum*

Direction of improvement **between two iterates**

$$\theta^t \leftarrow \theta^{t-1} - \eta \nabla L(f_{\theta^{t-1}}) + \beta(\theta^{t-1} - \theta^{t-2})$$

Classical Heavy-Ball (Polyak, 1964)

Consider a momentum over **larger period** $\mathcal{T}$

$$\theta^t \leftarrow \theta^{t-1} - \eta \nabla L(f_{\theta^{t-1}}) + \frac{\beta}{\tau}(\theta^{t-1} - \theta^{t-\tau-1})$$

**Generalised Heavy-Ball (ours)**

*Computing the **momentum term over a larger period** $\mathcal{T}$ allows for a **better estimate of the global model direction.***

## *Local GHB Momentum*

*Client update rule:*

GHB Momentum term computed on the server

$$\theta^{t}_{i,j+1} \leftarrow \theta^{t}_{i,j} - \eta \nabla L(f_{\theta^{t}_{i,j}}, d_{i,j}) + \hat{\beta}(\theta^{t-1} - \theta^{t-\tau-1})$$

Extra 1.5x communication

*Computing the **momentum term on the server over a larger period** allows for a **better estimate of the global model direction.***

***The GHB momentum contains the gradients from clients sampled between $\tau$ rounds***
*Particularly **important in FL** to correctly fix local direction because of **partial participation** and **heterogeneity.***

## *FedHBM: Communication Efficient HB Momentum*

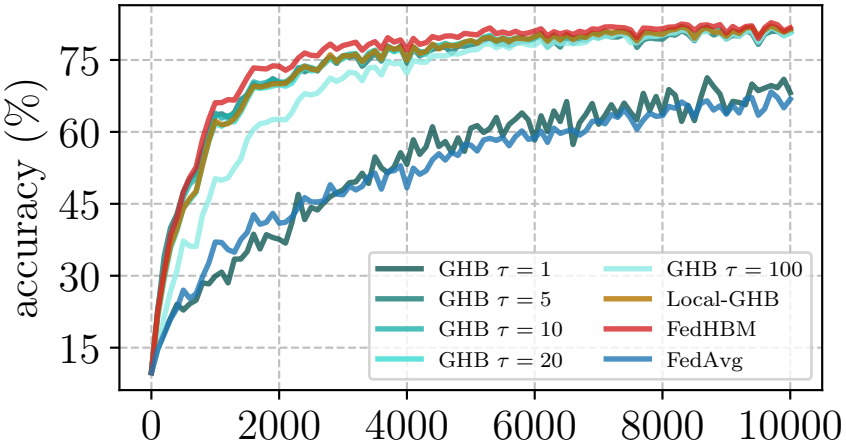
We can **efficiently compute** the global direction **without extra communication,** directly **on the client.**

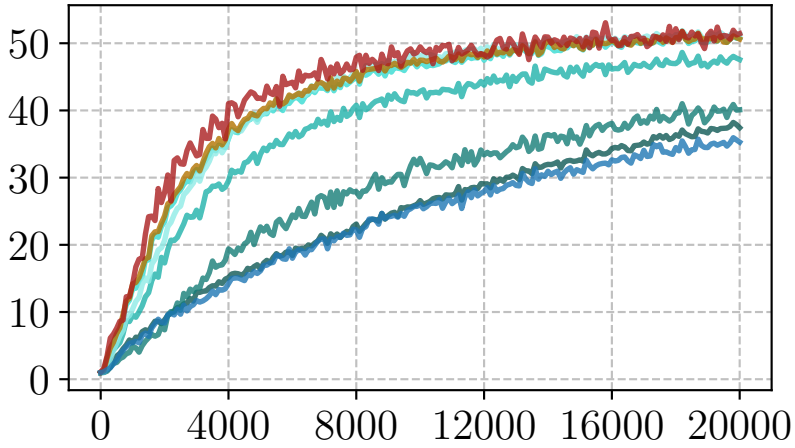
If clients participate multiple times in the training, they can **store the model received at past participation round** $t - \tau_i$

$$\theta^t_{i,j+1} \leftarrow \theta^t_{i,j} - \eta \nabla L(f_{\theta^t_{i,j}}, d_{i,j}) + \hat{\beta}_i(\theta^t_{i,j} - \theta^{t-\tau_i}_i)$$

*We can **leverage the heavy ball formulation** and estimate the global model direction by taking the difference between the current model and the model at the past client participation*

## *On the importance of GHB momentum*




*Existing momentum-based FL algorithms implement the special case of GHB with* $\tau = 1$

***Not really effective. Larger period, instead, speed up** convergence.*

# Main Results

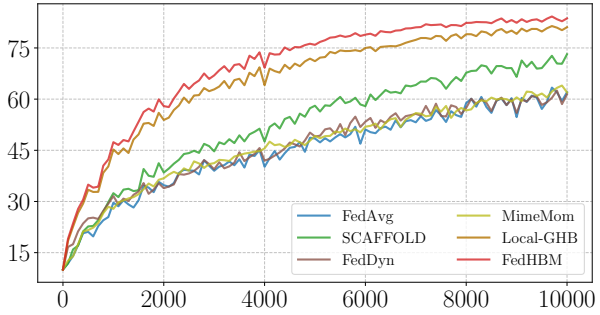
| Method | Comm. Overhead | Non-IID CIFAR-10 | Non-IID CIFAR-100 |
|---|---|---|---|
| FedAvg | 1x | 61.03 ± 1.06 | 21.94 ± 0.88 |
| SCAFFOLD | 2x | 71.78 ± 1.67 | 30.73 ± 1.31 |
| FedDyn | 1x | 60.25 ± 3.05 | 6.02 ± 0.52 |
| Mime | 2x | 53.69 ± 2.89 | 6.00 ± 0.41 |
| FedAvgM | 1x | 65.99 ± 2.24 | 22.81 ± 0.80 |
| MimeMom | 3x | 69.25 ± 3.64 | 21.67 ± 1.11 |
| MimeLiteMom | 2x | 57.03 ± 0.91 | 14.39 ± 0.58 |
| **FedHBM (ours)** | 1x | **83.39** ± 0.34 | **42.46** ± 0.76 |

ResNet20 - **Extreme Heterogeneity**

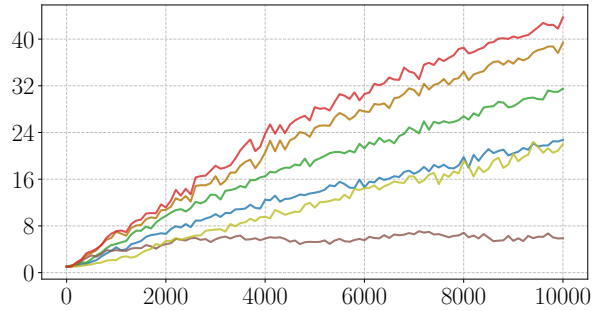
All baselines are heavily tuned for fair comparison

FedHBM is **much faster** and **communication-efficient**

CIFAR-10

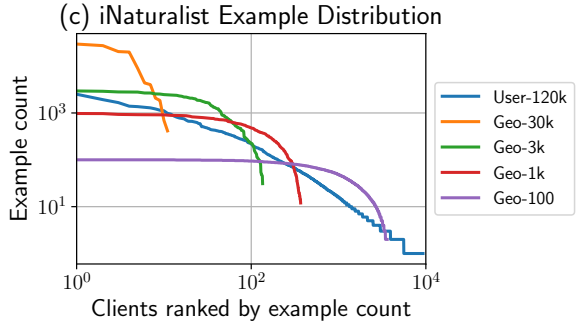


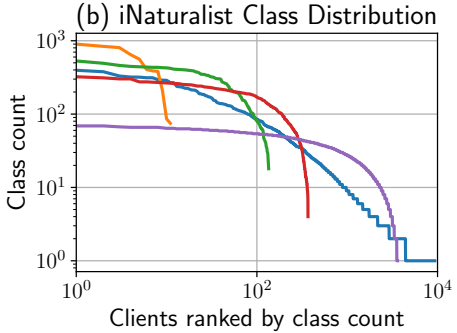

CIFAR-100



# Large-scale Real-World Classification

| Dataset | Clients | | Classes | Examples |
|---|---|---|---|---|
| | Count | Size Imbalance | Count | Count |
| **Synthetic** | | | | |
| CIFAR-10 | 100 | ✗ | 10 | 50,000 |
| CIFAR-100 | 100 | ✗ | 100 | 50,000 |
| **Real-World** | | | | |
| iNaturalist-User-120k | 9,275 | ✓ | 1,203 | 120,300 |
| Landmarks-User-160k | 1,262 | ✓ | 2,028 | 164,172 |

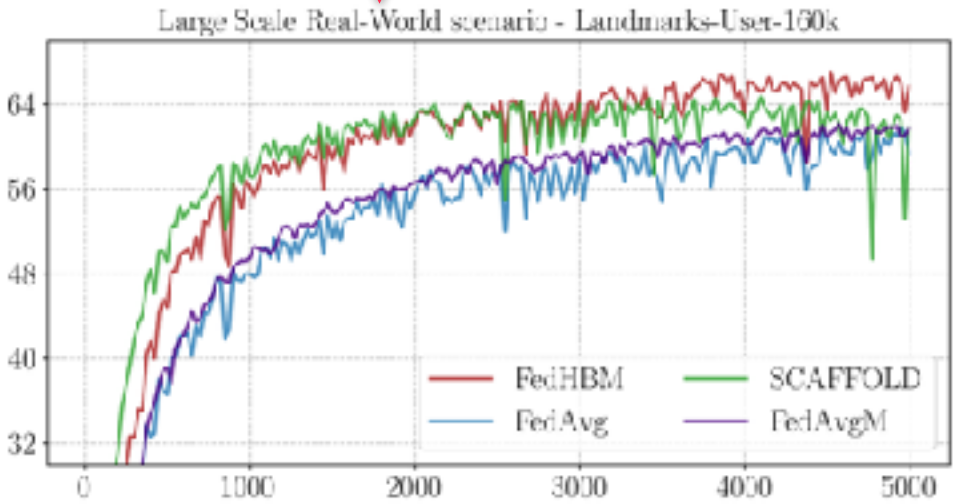



# Large-scale Real-World Classification

| Method | Comm. Overhead | Landmarks-Users-160K | iNaturalist-Users-120K | | |
|---|---|---|---|---|---|
| | | $C \approx 0.79\%$ | $C \approx 0.1\%$ | $C \approx 0.5\%$ | $C \approx 1\%$ |
| FedAvg | 1× | $60.31 \pm 0.18$ | $38.03 \pm 0.84$ | $45.25 \pm 0.07$ | $47.59 \pm 0.13$ |
| SCAFFOLD | 2× | $61.03 \pm 0.08$ | 0.0 | 0.0 | 0.0 |
| FedAvgM | 1× | $61.50 \pm 0.22$ | $41.34 \pm 0.38$ | $46.08 \pm 0.09$ | $48.37 \pm 0.07$ |
| MimeMom | 3× | 0.0 | 0.0 | 0.0 | 0.0 |
| **FedHBM (ours)** | 1× | **65.41** $\pm 0.17$ | **41.64** $\pm 0.18$ | **47.33** $\pm 0.04$ | **49.80** $\pm 0.05$ |

Participation Rate



QUESTION 3:

# Which part of the model is the most affected by heterogeneity?

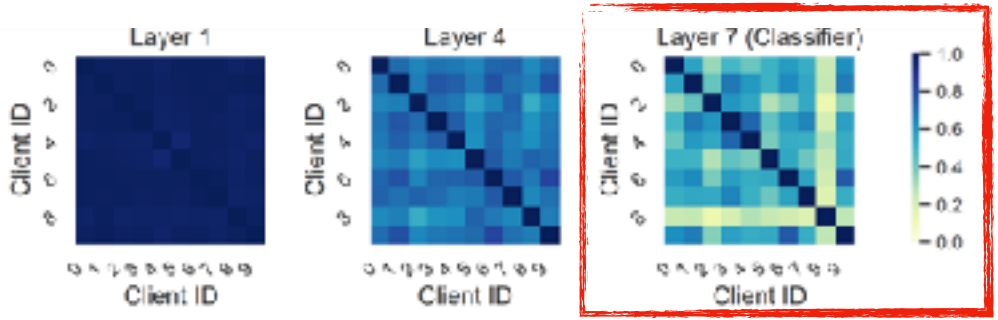
**Under review**

*Accelerating Heterogeneous Federated Learning with Closed-form Classifiers*

E. Fanì, R. Camoriano, B. Caputo, **M.Ciccone**

# Classifier bias in Federated Learning

- Recent work show that **deeper layers are more subject to client drift**
- Shown by **CKA similarities** of different layers in clients' local model pairs




*Heterogeneity and partial participation **mainly affect the classifier***

Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, & Jiashi Feng. (2021). No Fear of Heterogeneity: Classifier Calibration for Federated Learning with Non-IID Data.

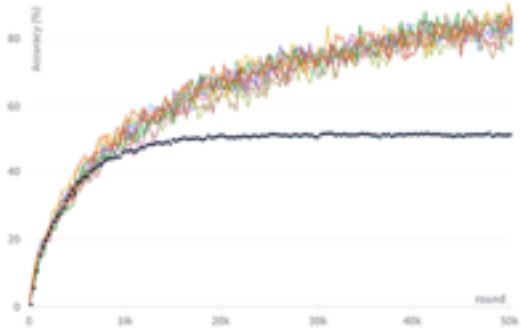
# Data-recency bias in Heterogenous Federated Learning

- Because of **partial participation**, at each round the **classifier becomes overly biased** towards recent participating clients overwriting past knowledge (**forgetting)**.
- Similar effect in **Continual Learning**, where data are **observed as non-iid** (sequentially)

Each color is the accuracy of one class

**Extreme Heterogeneity**

**Homogeneous Distribution**



From D. Caldarola, B. Caputo, **M. Ciccone** "Improving Generalization in Federated Learning by Seeking Flat Minima " ECCV 2022


Ruohan Wang*, **Marco Ciccone***, Massimiliano Pontil, & Carlo Ciliberto. (2022). Schedule-Robust Online Continual Learning. (Submitted to TPAMI)
Masana, Marc, et al. Class-incremental learning: survey and performance evaluation on image classification. IEEE TPAMI
Wu, Yue, et al. (2019) Large scale incremental learning. CVPR
Mai, Zheda, et al. (2021) Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. CVPRW

## *Solving the biased-classifier problem*

### 💡 Idea*

*1) Replace softmax classifier with a **closed-form** classifier*

*2) Decouple the representation and classifier learning*

**** Similar idea used in Continual Learning:***

Ruohan Wang*, **Marco Ciccone***, Massimiliano Pontil, & Carlo Ciliberto. (2022) Schedule-Robust Online Continual Learning. (Submitted to TPAMI)

# Ridge Regression

Pretrained feature extractor

We can use a **"one vs all"** classifier $f(x) = \arg\max_y \; w_y^\top \psi(x),$

$$W^* = \arg\min_W \frac{1}{|D|} \sum_{(x,y)\in D} \left\| W^\top \psi(x) - \text{OneHot}(y) \right\|^2 + \lambda \left\| W \right\|^2 .$$

Closed-form solution

$$W^* = (\underbrace{X^\top X}_{A} + \lambda I)^{-1} \underbrace{X^\top Y}_{b}$$

$$X = [\psi(x_1) \dots \psi(x_N)]^\top$$
$$Y = [\text{OneHot}(y_1) \dots \text{OneHot}(y_N)]^\top$$

Online Update

$$w_z = (A^{\text{new}} + \lambda I)^{-1} b_z^{\text{new}} \quad \forall z \in \mathcal{Y}$$

$$A^{\text{new}} = A^{\text{old}} + \psi(x)\psi(x)^\top$$

$$b_z^{new} = \begin{cases} b_z^{\text{old}} + \psi(x) & \text{if } z = y \\ b_z^{\text{old}} & \text{otherwise.} \end{cases}$$

Ridge Regression has an **exact online formula** to recover the closed-form solution

Ruohan Wang*, **Marco Ciccone***, Massimiliano Pontil, & Carlo Ciliberto. (2022). Schedule-Robust Online Continual Learning.

# Ridge Regression properties

Thanks to the associative and commutative property of the sum **the matrices A and b can be built incrementally:**

$$A = \sum_{(x,y)\in\mathcal{D}} \psi(x)\psi(x)^\top = \sum_{k\in\mathcal{K}} \sum_{(x,y)\in\mathcal{D}_k} \psi(x)\psi(x)^\top = \sum_{k\in\mathcal{K}} A_k$$

$$b = \sum_{(x,y)\in\mathcal{D}} \psi(x)e_y^\top = \sum_{k\in\mathcal{K}} \sum_{(x,y)\in\mathcal{D}_k} \psi(x)e_y^\top = \sum_{k\in\mathcal{K}} b_k$$

Centralised dataset

Clients' Local datasets

The RR solution is **equivalent to the centralised**, once all the clients have shared their statistics

Fed3R **is invariant to federated split** $\mathcal{D} = \bigcup_{k\in\mathcal{K}} \mathcal{D}_k$ **and client order sampling**

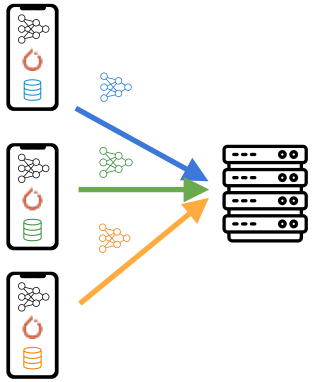
# Fed3R: Federated Recursive Ridge Regression

Compute Local Ridge Statistics

$$A_k^t = \sum_{(x,y)\in\mathcal{D}_k} \psi(x)^\top \psi(x)$$

$$b_k^t = \sum_{(x,y)\in\mathcal{D}_k} \psi(x)^\top e_y$$

**Client side**



Aggregate Statistics

$$A^t = A^{t-1} + \sum_{k\in\mathcal{K}} A_k^t$$

$$b^t = b^{t-1} + \sum_{k\in\mathcal{K}} b_k^t$$

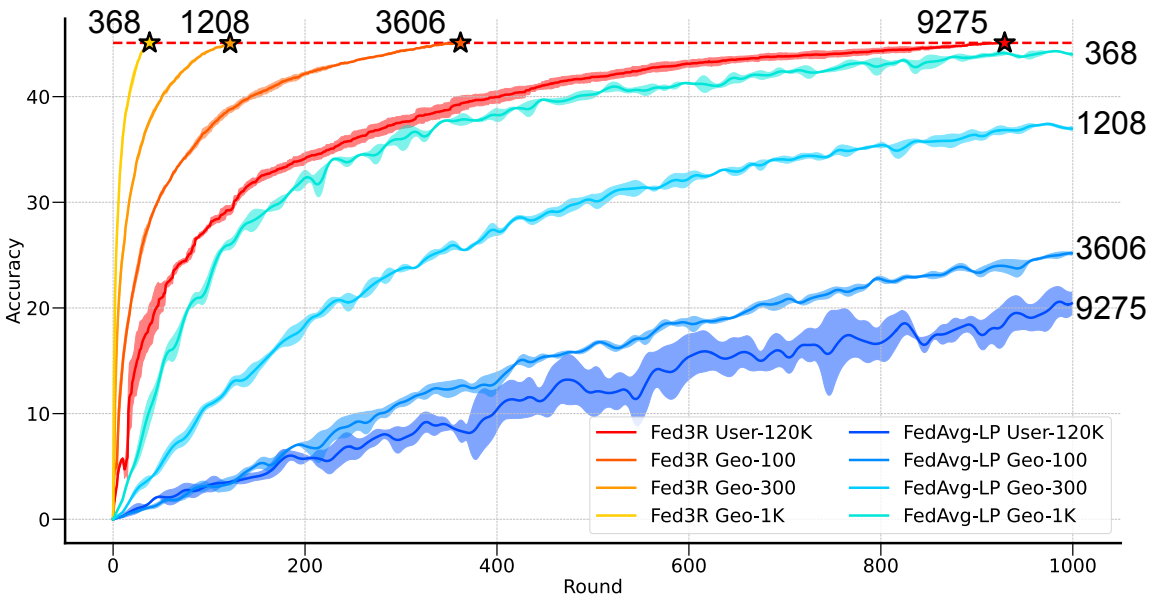
$$w = (A^t + \lambda I)^{-1} b^t$$

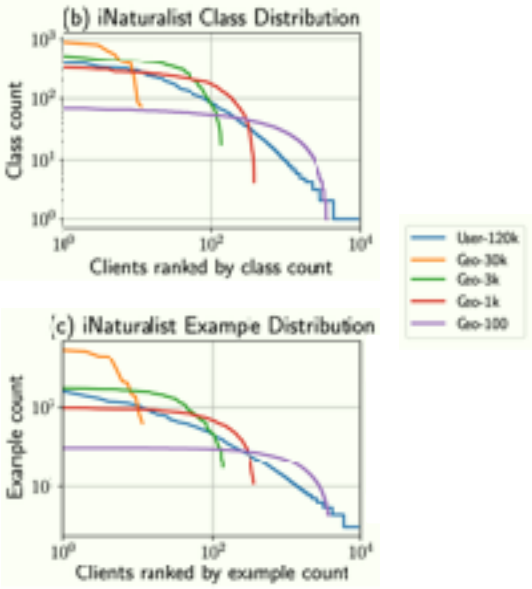
**Server side**

*Fed3R builds the classifier incrementally using an **exact online formulation***

*This **solves the bias** of the classifier towards more recent clients*

# Results for different levels of heterogeneity



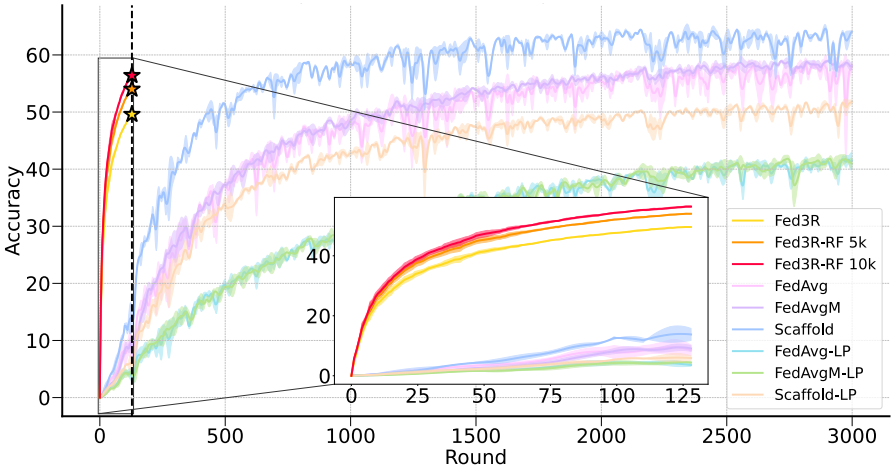




*Once all clients have been sampled Fed3R* ***recovers the centralised solution***

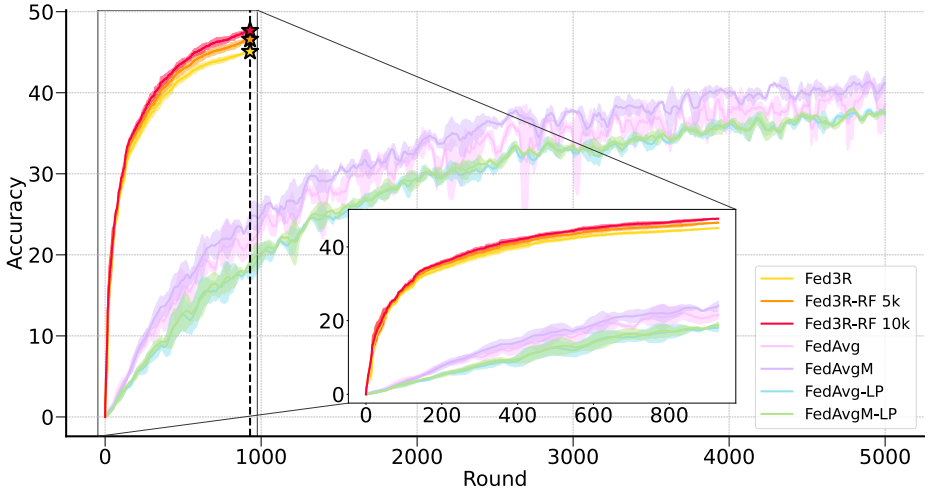
*The result is* ***invariant*** *to the client split and the sampling order.*

# Fed3R vs baselines

Google Landmark - 2028 Classes - 1262 Clients



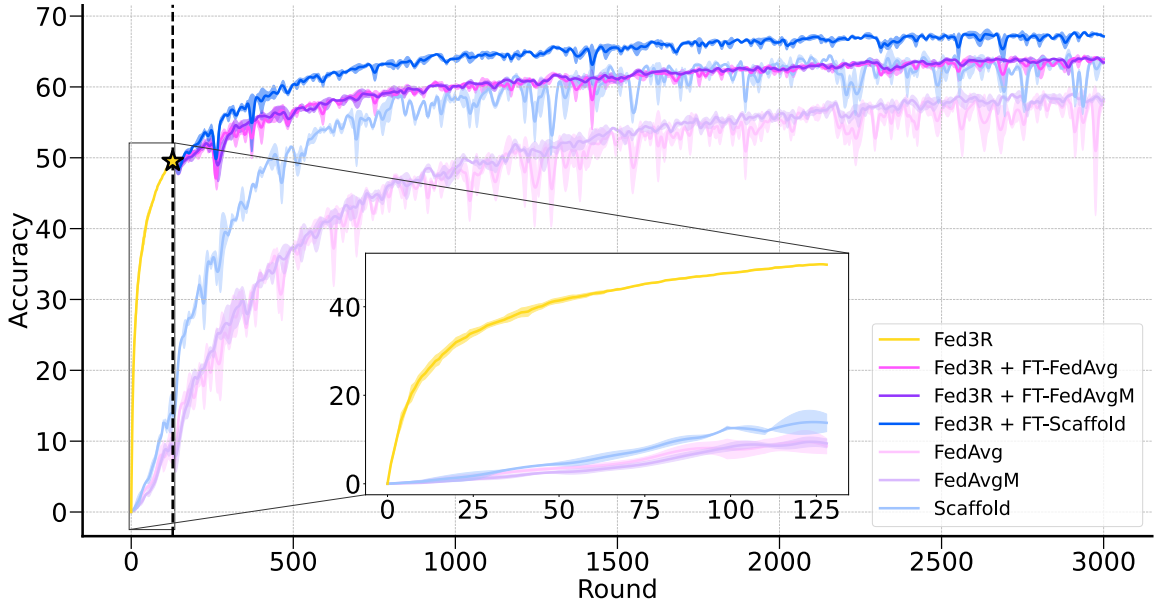

iNaturalist - 1203 Classes - 9275 Clients




*Fed3R is very effective in realistic heterogenous settings with thousands of clients*

***Other more advanced baselines fail in this setting***

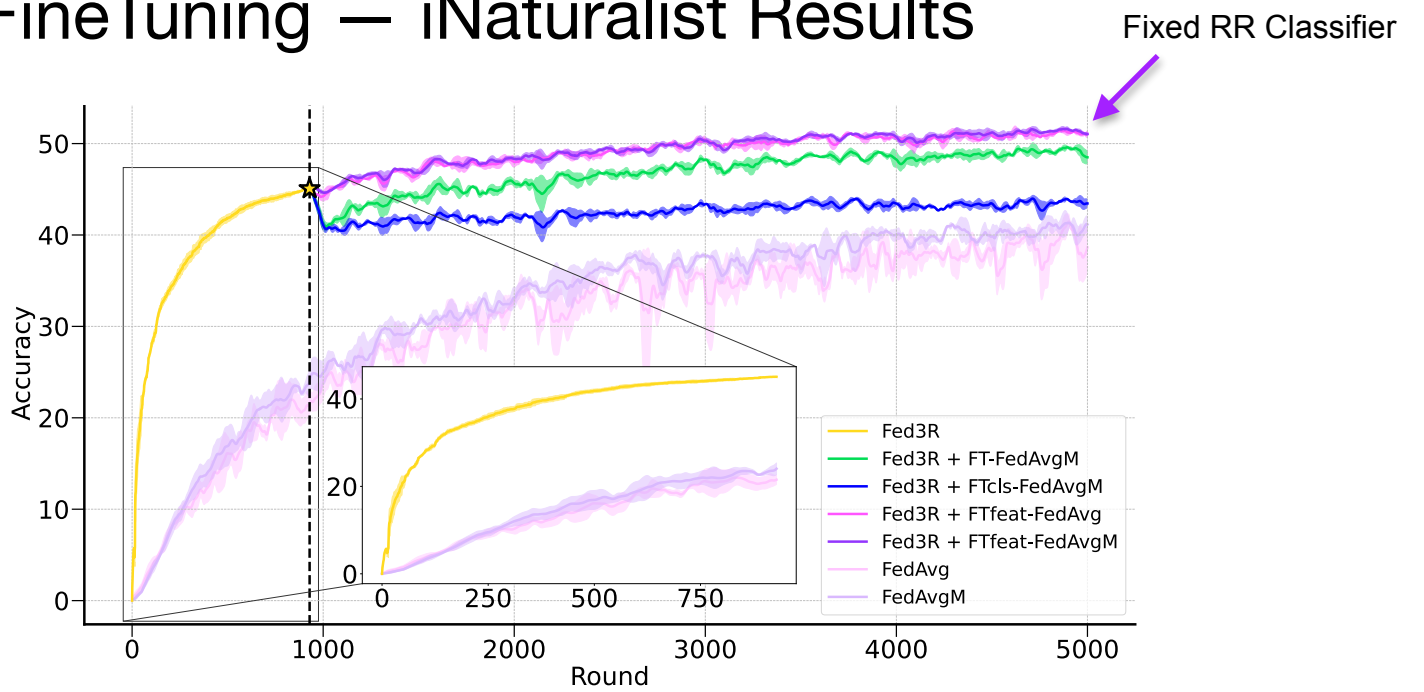
# Fed3R + FineTuning — gldv2 Results




*Fed3R can be used as **robust initialization** for any FL algorithm*

*Fine-tuning further improve final accuracy*

# Fed3R + FineTuning — iNaturalist Results




*Learning or fine-tuning in realistic cross-device settings is hard*

***Fixing the Fed3R classifier and fine-tuning only the representation stabilizes the training***

# Recap statistical heterogeneity and solutions

*Federated Learning is affected by statistical Heterogeneity, hampering convergence speed and final performance.*

**Take home messages:**

- 💪 **FedSAM:** Improving local learners leads to better global model performance.
- 🚀 **FedHBM:** Injecting global information into the local optimization mitigates the client-drift and speed-up convergence
- 🛡️ **Fed3R:** Linear classifiers with closed form solution dramatically improve performance and mitigate recency bias and forgetting in non-iid settings

# What is the role of decentralisation in the era of large-scale training?

# Challenges of Large-Scale Centralized Training

*Training foundation models requires centralization of thousands of co-located homogeneous devices interconnected with high-bandwidth.*

⏳ **Communication Bottleneck**: model/gradients synchronization at each training step

💸 **Cost:** scaling and maintaining a single, massive infrastructure is expensive

💥 **Device Failure:** the higher the number of devices, the higher the risk of failure

*Only a few privileged teams can afford to train and shape the development of these models*

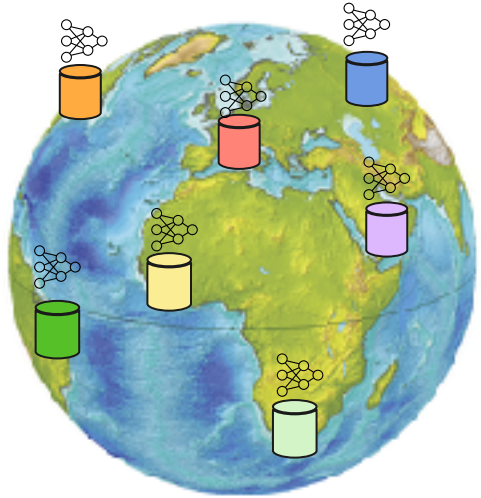
# Communication-Efficient Decentralized Large-Scale Training

*Communication-efficient decentralized algorithms, can enable:*

🌐 **World-wide collaborative** training, no need of device co-location

🏫 Universities and other institutions to **pool their resources** to train larger and better models

👥 In principle, **anyone** who has resources could share data and compute, even if **heterogeneous** (asynchronous communication)



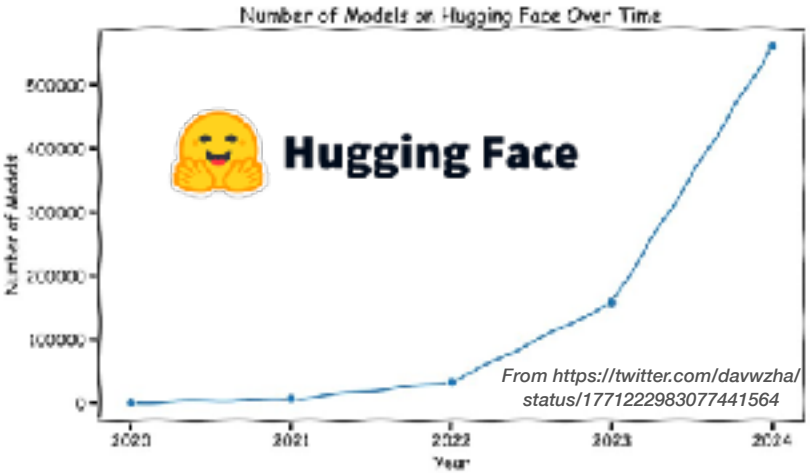
*Researchers started to realise the importance of this paradigm to train LLMs*
*e.g., DiLoCo (DeepMind), FlowerLLM (Flower)*

Different levels of decentralization,
Based on the frequency of synchronization (merge)

Extreme: specialized models, no sync

# Decentralization is already happening

AdapterHub

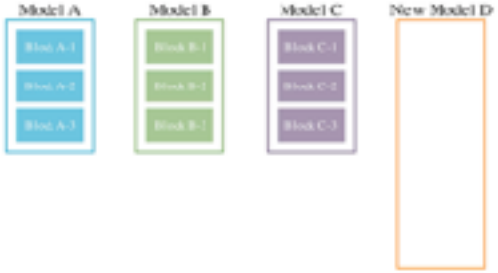



*601,859 - 15th April*

*Specialized models or adapters are trained independently, weights are shared online on repositories*

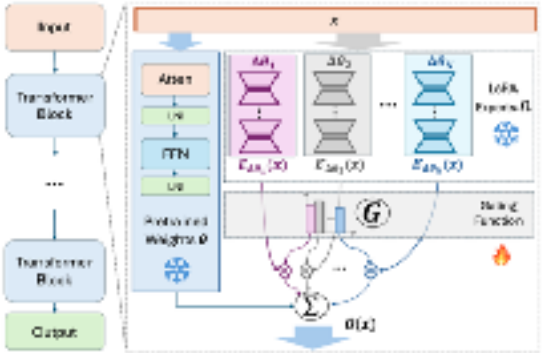
*We can **recycle** available knowledge*

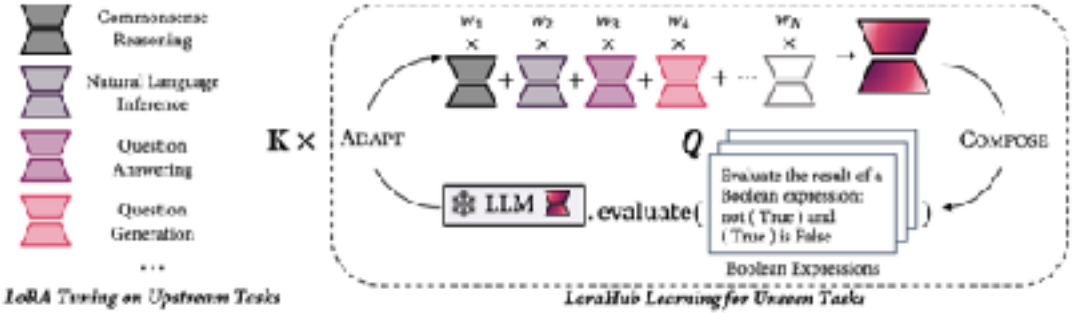
# Examples of model merging




*From https://sakana.ai/evolutionary-model-merge/*


https://ziplora.github.io/





# Challenges and open questions

***Routing**: we need strategies to select the right models based on the task*

***Merging**: we need algorithms to combine models to achieve optimal transfer and reduce interference*

*(also, some theory would be nice)*

# Modularity

Main **roadblock** for decentralization: **monolithic systems**

**Capabilities are intertwined in model parameters and are hard to modify**

Model should be developed with **modular design** and **functional specialisation** to enable:

- Asynchronous training and incremental updates (Continual Learning)
- Composability for generalization
- Conditional, sparse and decentralized inference

**Thank you for the attention,**

**any question?**

If you have feedback, or you want to **collaborate** on these topics, get in touch!

marco.ciccone@me.com

# BACKUP

# Federated Optimization

**Client Optimization (Local SGD)**

$$\theta^t_{i,k+1} = \textsc{ClientOpt}(\theta^t_{i,k}, g^t_{i,k}, \eta_\ell, t)$$

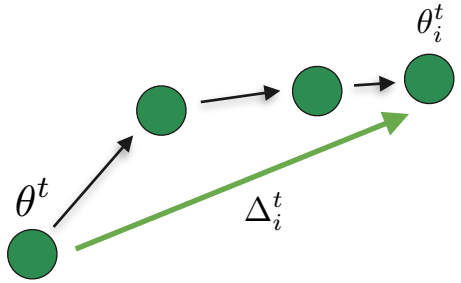
**Local Learning rate** (and any other hyper-param)

**Local Gradient** (i-th client, k-th step)

**Server Optimization (More General aggregation)**

$$\theta^{t+1} = \textsc{ServerOpt}(\theta^t, -\Delta^t, \eta, t)$$

Extend to different server optimisers (Adam, Momentum




Consider the **displacement** between model initialization and end-point as **Pseudo-gradient**

Recovers FedAVG with SGD and server learning rate $\eta = 1$

$$\theta^{t+1} = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\theta^t_i = \theta^t - \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\underbrace{\theta^t - \theta^t_i}_{\Delta^t_i}$$

Reddi, Sashank, et al. "Adaptive Federated Optimization." *ICLR*, 2021.

# FL motivation is privacy

**Data is produced at the edge**

Smartphones, smartwatches, home assistants, IoT devices constantly generate user data from digital interactions and real-world sensing

**It is increasingly important to keeping data on devices to preserve privacy and security of data across many sectors**

📱 Mobile Devices
💳 Finance
🛍️ Retail
🏥 Healthcare
🚦 Smart Cities

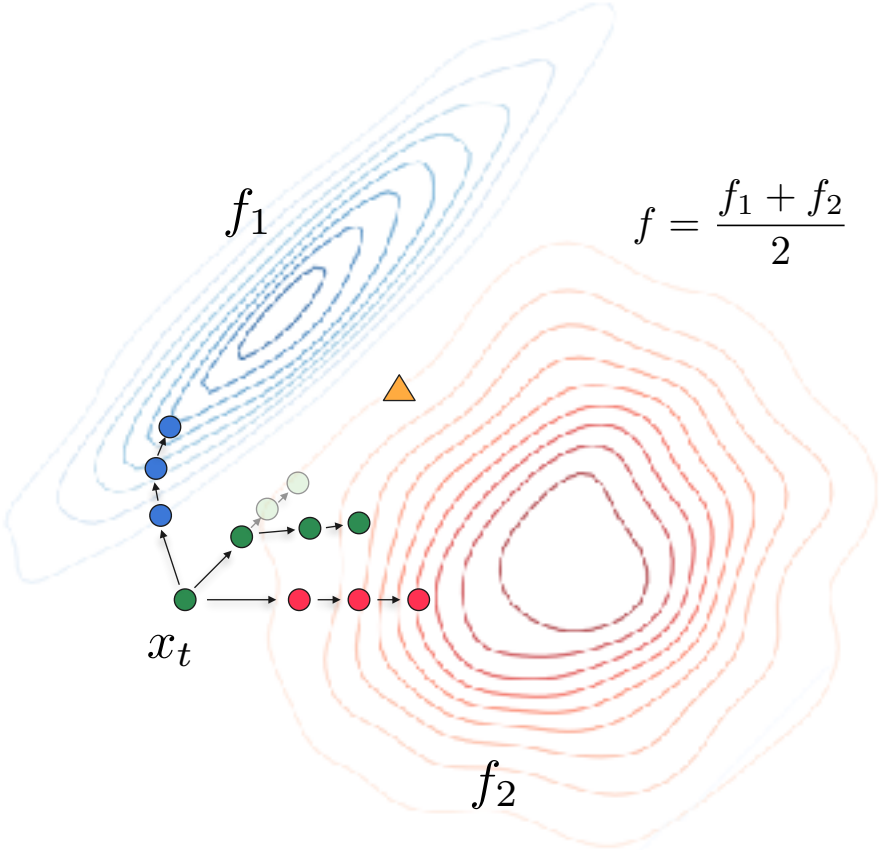
Still, Federated Learning requires Differential Privacy

Boenisch, Franziska, et al. "When the curious abandon honesty: Federated learning is not private." 2023 IEEE 8th European Symposium on Security and Privacy

# Client-drift problem

**Heterogeneity causes the local optimization to *drift* from the global minimum**

- Global convergence is much slower
- May not converge to the true optimum




Slides credit, Karimireddy Scaffold presentation

# FedSAM (client-side) Sharpness-Aware Minimization

**Apply SAM locally on the client side:**

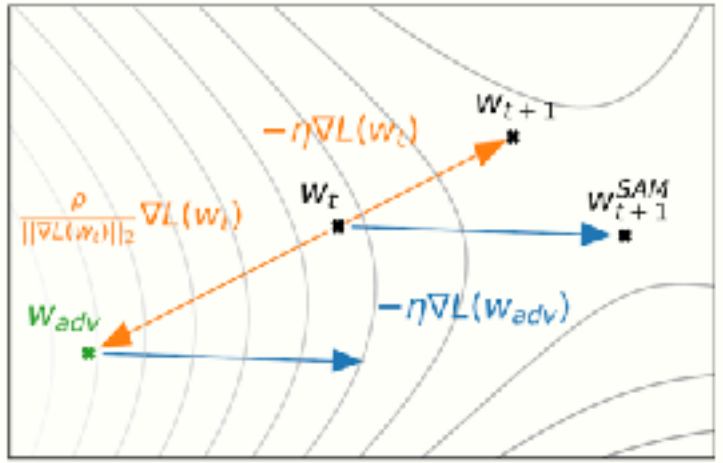
- Explicitly search for parameters whose **entire neighbourhood have uniformly low training loss** (both low loss and low curvature)

**Two steps approximated solution:**

1. Find worst epsilon perturbation
2. Minimize loss on perturbed weights

$$\min_{w} L_S^{SAM}(w) + \lambda ||w||_2^2$$

$$\text{where} \quad L_S^{SAM}(w) \triangleq \max_{||\epsilon||_p \leq \rho} L_S(w + \epsilon),$$




Foret, Pierre, et al. "Sharpness-aware minimization for efficiently improving generalization." *ICLR 2021*

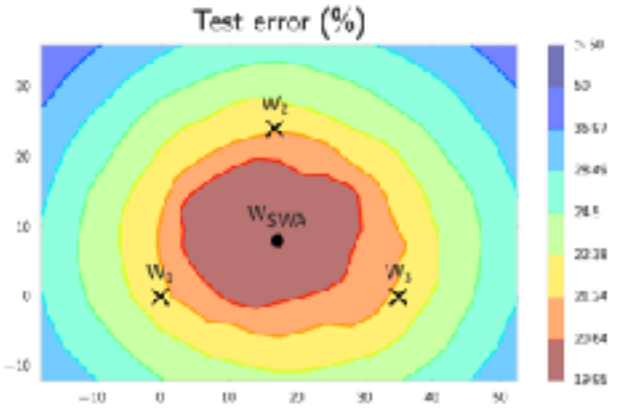
# SWA (server-side)
# Stochastic Weight Averaging

- SWA performs an **equal average** of the SGD iterates
- Cyclic **learning rate schedule** to continue to explore solutions
- Applied only at convergence (75% training)

**SWA for FL**

- Apply SWA on the server on aggregated models
- Reduce LR on the client-side to continue explore

$$\theta_{\text{SWA}} \leftarrow \frac{\theta_{\text{SWA}} \cdot n_{models} + \theta}{n_{models} + 1}$$




Izmailov, Pavel, et al. "Averaging weights leads to wider optima and better generalization." *UAI 2018*

# Window-based Model Aggregation

***Alternative to SWA:*** *average the last k models.*

$$w_{\text{WIMA}}^{t+1} - w_{\text{WIMA}}^{t'+W} := \frac{1}{W} \sum_{\tau=t'}^{t'+W-1} w_{\text{FEDAVG}}^{\tau+1}$$

**Findings:**

- WiMA can be applied since the start of training
- **Always improve performance**
- **Mitigate the noise** of partial participation