An Introduction to Federated Learning

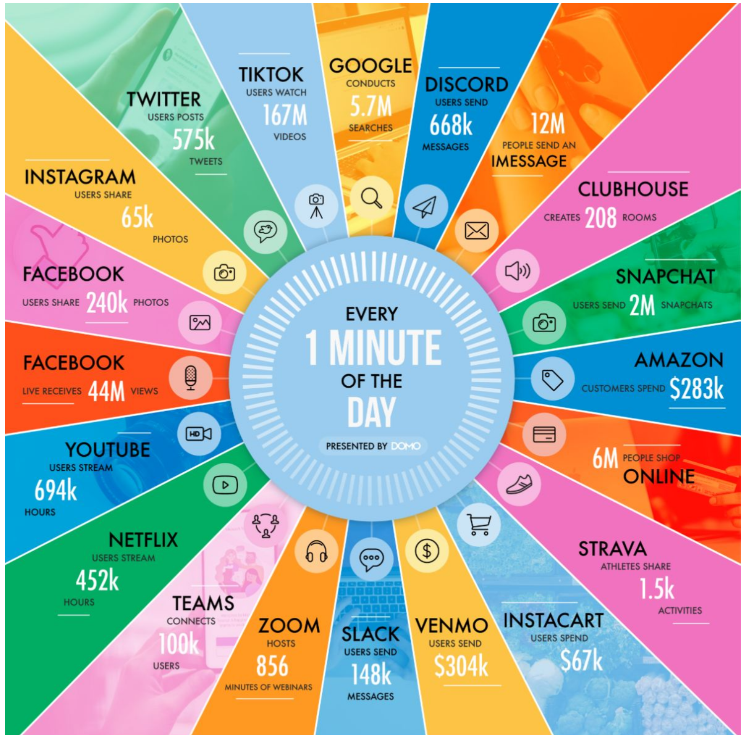
Andreas Tritsarolis, 26/01/2024

Outline

- From Centralized to Decentralized Learning
- Federated Learning (FL)
- Categories of Federated Learning
- Federated vs Distributed Machine Learning
- Privacy Preservation Techniques
- Optimization in Federated Learning (FedAvg)
- Advances and Open Problems
- Programming Frameworks

From Centralized to Decentralized Learning

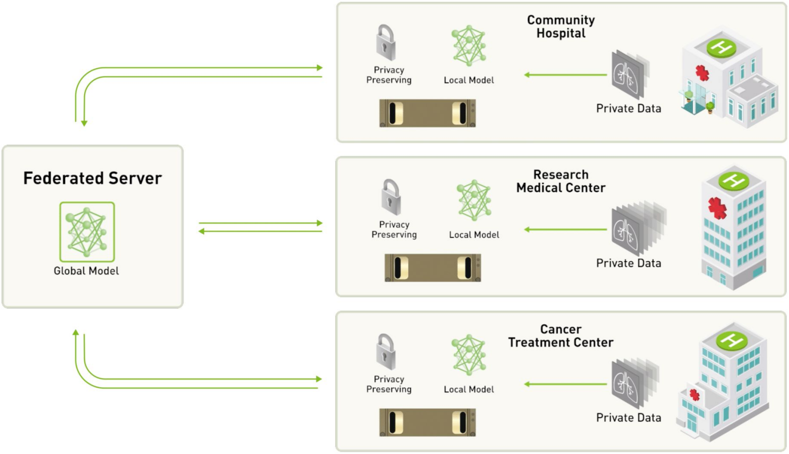
- Data is **Everywhere**…
  - Edge devices upload vast amounts of data per minute
- …and often **decentralized** across many parties
- Training a Machine Learning (ML) model
  - Requires a centralized dataset processed in a tightly integrated system
- Centralizing Process ⟶ **Privacy** nightmare
  - High Costs (e.g. storage, bandwidth, etc.)
  - High Sensitivity (e.g. finance, health, etc.)
- Delegating ML Training ⟶ **Performance** nightmare
  - Dataset too small ⟶ Underfitting/Overfitting
  - Features' distribution ⟶ Biased target distribution



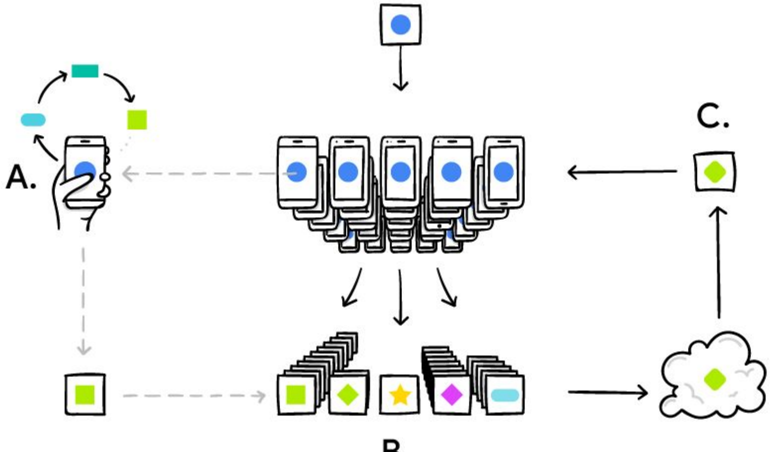
[source]

Federated Learning

- Federated Learning (FL)
  - Train a centralized model on decentralized data
- Edge devices collaboratively train an ML model
  - Keep the data decentralized
- Every participant keeps control of its own data
  - Harder to extract sensitive information



[source]



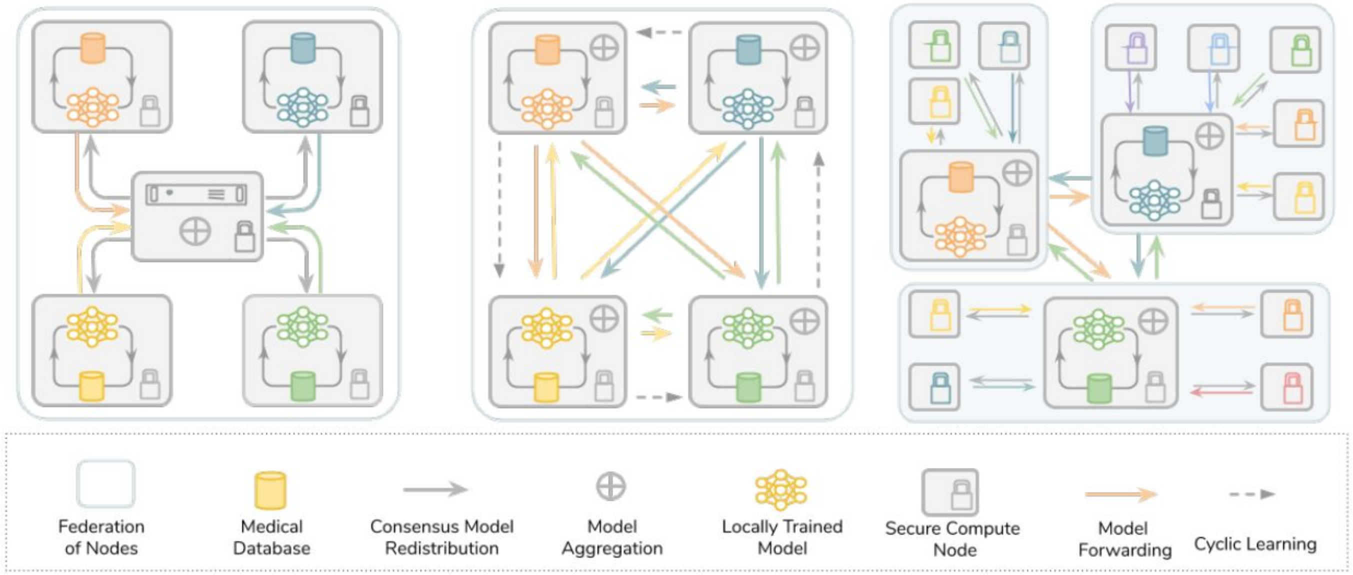
[source]

H. B. McMahan, E. Moore, D. Ramage, et al., Communication-efficient learning of deep networks from decentralized data, ArXiv Preprint ArXiv:1602.05629, February 2016. https://arxiv.org/abs/1602.05629

J. Konecný, H. B. McMahan, D. Ramage, et al., Federated optimization: Distributed machine learning for on-device intelligence, ArXiv Preprint:1610.02527, October 2016. http://arxiv.org/abs/1610.02527

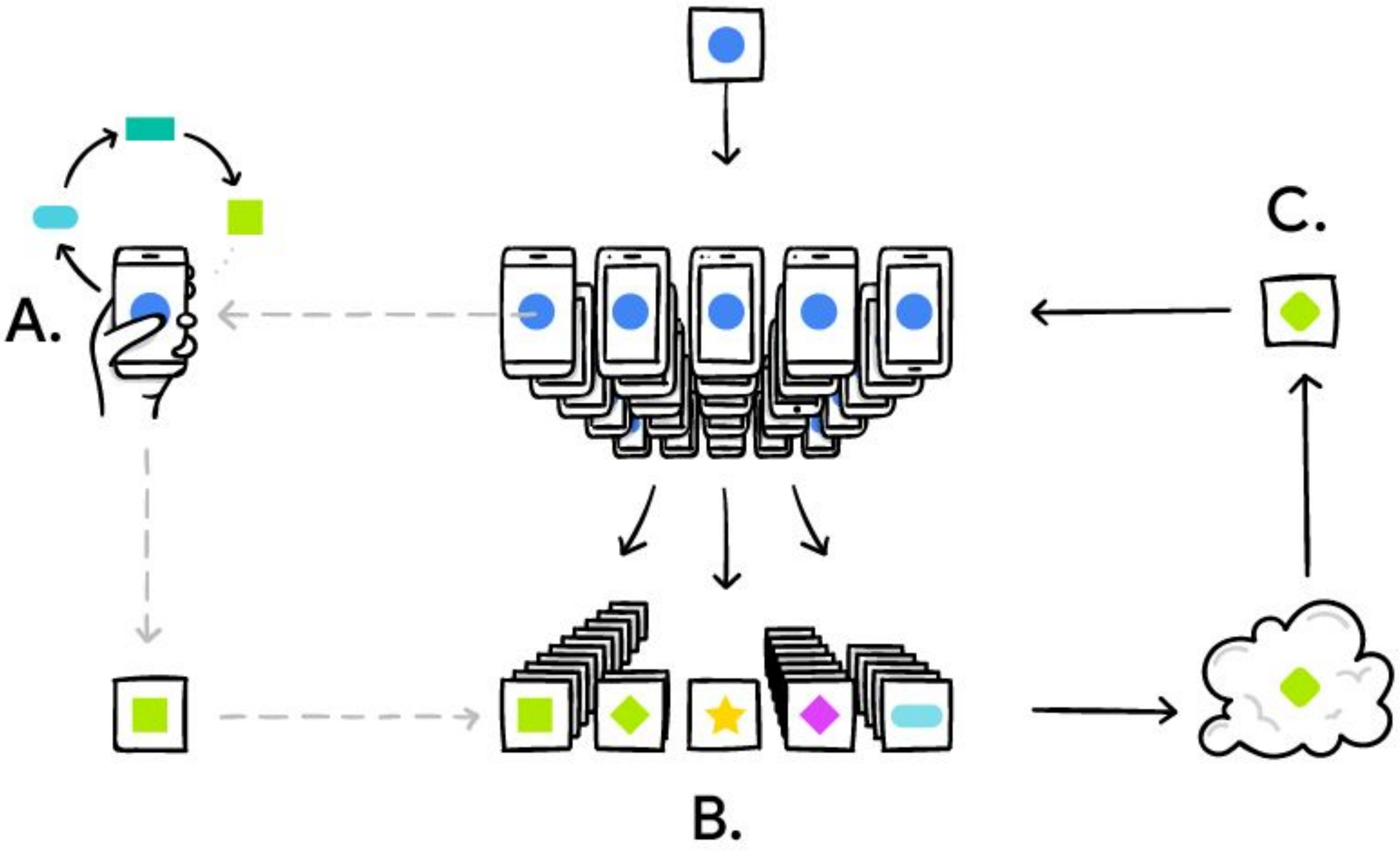
Federated Learning (cont.)

- Edge devices send model updates…
  - Client-Server architecture via Hub and Spokes
  - Decentralised Architecture via peer-to-peer
  - Hybrid Architecture
- Our scope of interest → Client-Server



[source]

Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K. H., Ourselin, S., Sheller, M. J., Summers, R. M., Trask, A., Xu, D., Baust, M., & Cardoso, M. J. (2020). The Future of Digital Health with Federated Learning. CoRR, abs/2003.08119. https://arxiv.org/abs/2003.08119



Your phone personalizes the model locally, based on your usage (A). Many users' updates are aggregated (B) to form a consensus change (C) to the shared model, after which the procedure is repeated. [source]

Optimization in Federated Learning

- Baseline → Federated Average
- Consider
  - A set of ***K*** parties (clients)
  - Each party ***k*** holds a dataset D_k of n_k points
  - D = D_1 U … U D_k → the joint users' dataset
    - $n = \Sigma_k n_k$ → #points
  - Aim → Minimize **F(*θ; D*)**, w.r.t ***θ***, where

$$F(\theta;\mathcal{D}) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(\theta;\mathcal{D}_k) \quad \text{and} \quad F_k(\theta;\mathcal{D}_k) = \sum_{d\in\mathcal{D}_k} f(\theta;d)$$

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

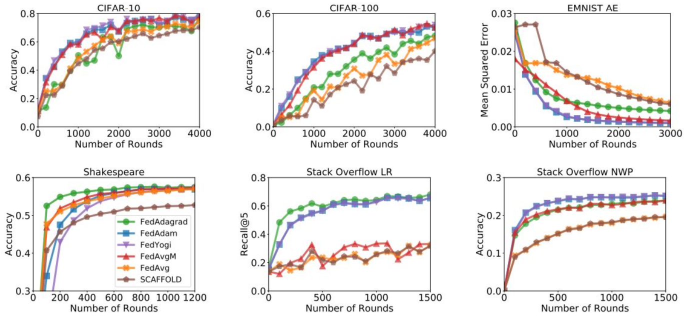
Server executes:
  initialize w_0
  for each round $t = 1, 2, \dots$ **do**
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of m clients)
    for each client $k \in S_t$ **in parallel do**
      $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w)**:** // *Run on client k*
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
  for each local epoch i from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return w to server

McMahan, H. B., Moore, E., Ramage, D., & y Arcas, B. A. (2016). Federated Learning of Deep Networks using Model Averaging. CoRR, abs/1602.05629. http://arxiv.org/abs/1602.05629

Optimization in Federated Learning (cont.)

- Reddi et al. generalize the aforementioned scheme
  - Allow usage of adaptive optimization schemes (e.g. Adam)
- Clients use SGD
- Server use AdaGRAD, YOGI, Adam
- Advantages
  - Maintain communications costs (same as FedAvg)
  - Work in cross-device settings



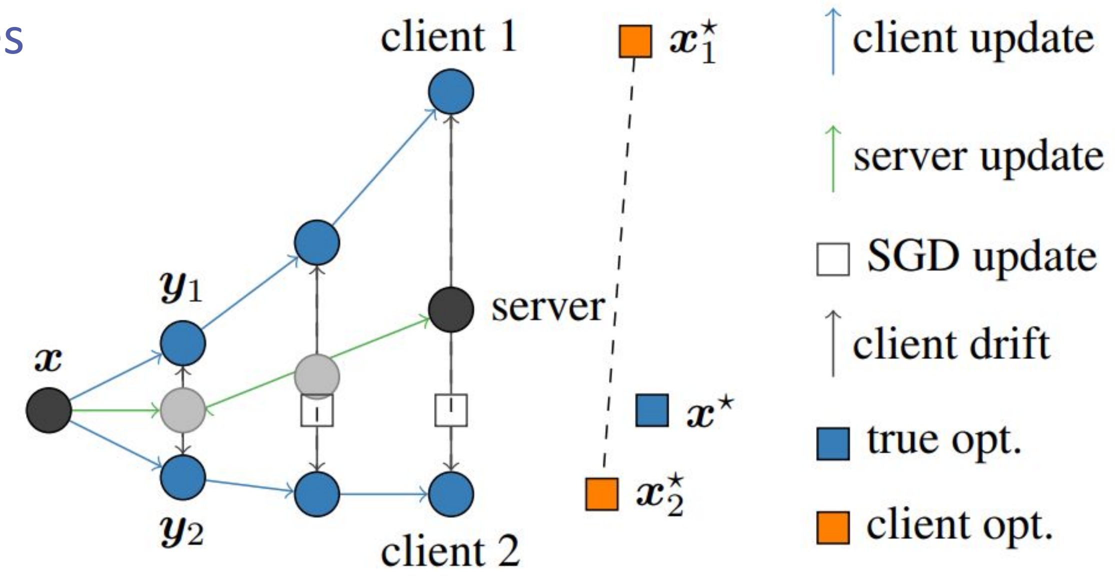
Algorithm 2 **FEDADAGRAD**, **FEDYOGI**, and **FEDADAM**

Initialization: $x_0, v_{-1} \geq \tau^2$, decay parameters $\beta_1, \beta_2 \in [0, 1)$
for $t = 0, \cdots, T-1$ **do**
  Sample subset $\mathcal{S}$ of clients
  $x_{i,0}^t = x_t$
  for each client $i \in \mathcal{S}$ **in parallel do**
    for $k = 0, \cdots, K-1$ **do**
      Compute an unbiased estimate $g_{i,k}^t$ of $\nabla F_i(x_{i,k}^t)$
      $x_{i,k+1}^t = x_{i,k}^t - \eta_l g_{i,k}^t$
    $\Delta_i^t = x_{i,K}^t - x_t$
  $\Delta_t = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \Delta_i^t$
  $m_t = \beta_1 m_{t-1} + (1 - \beta_1)\Delta_t$
  $v_t = v_{t-1} + \Delta_t^2$ (**FEDADAGRAD**)
  $v_t = v_{t-1} - (1 - \beta_2)\Delta_t^2 \,\mathrm{sign}(v_{t-1} - \Delta_t^2)$ (**FEDYOGI**)
  $v_t = \beta_2 v_{t-1} + (1 - \beta_2)\Delta_t^2$ (**FEDADAM**)
  $x_{t+1} = x_t + \eta \frac{m_t}{\sqrt{v_t} + \tau}$

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., & McMahan, H. B. (2020). Adaptive Federated Optimization. CoRR, abs/2003.00295. https://arxiv.org/abs/2003.00295

Advances and Open Problems

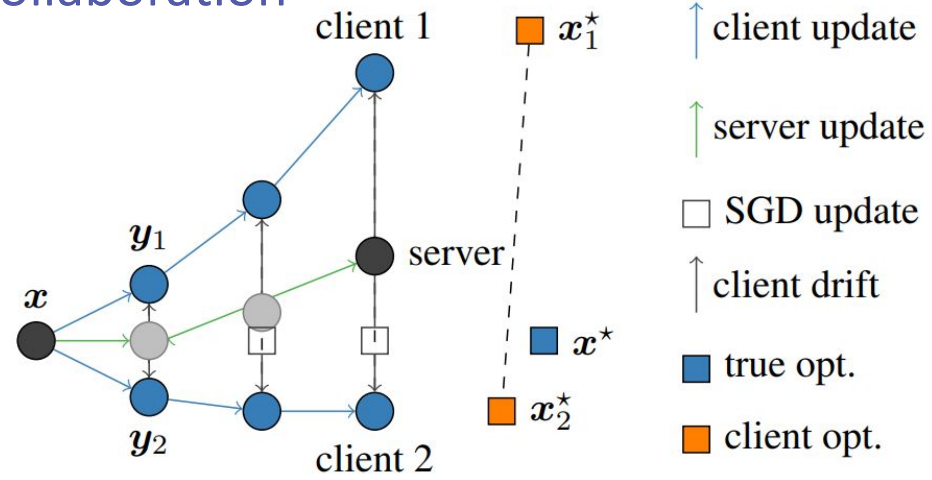
- In non-i.i.d datasets ⟶ client drift
  - FedAvg fails to converge
- Prevent client drift ⟶ lowers convergence rate
  - Use fewer local updates
  - Smaller learning rates



Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., & Suresh, A. T. (2019). SCAFFOLD: Stochastic Controlled Averaging for On-Device Federated Learning. CoRR, abs/1910.06378. http://arxiv.org/abs/1910.06378

Advances and Open Problems (cont.)

- (If) data distributions are very different
  - learning a single model ⟶ requires a (very) large number of parameters
- Lift the "one size fits all" requirement
  - i.e., the learned model should perform well for all parties
- Allow each party $\boldsymbol{k}$ to learn a (simpler) personalized model $\boldsymbol{\theta}_k$
  - design the objective so as to enforce some kind of collaboration

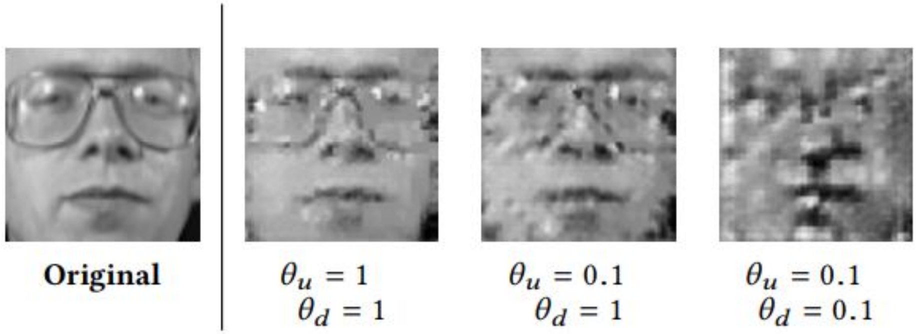


Hanzely, F., Hanzely, S., Horváth, S., & Richtárik, P. (2020). Lower Bounds and Optimal Algorithms for Personalized Federated Learning. CoRR, abs/2010.02372. https://arxiv.org/abs/2010.02372

Fallah, A., Mokhtari, A., & Ozdaglar, A. E. (2020). Personalized Federated Learning: A Meta-Learning Approach. CoRR, abs/2002.07948. https://arxiv.org/abs/2002.07948

Advances and Open Problems (cont.)

- Including (among others)
  - Robustness against adversaries
    - e.g., Gradient leaks
  - Incentive mechanisms
    - e.g. Blockchain
  - Model diversity
    - Ensure fairness without access to sensitive attributes
  - Communication costs
    - Improve efficiency ⟶ Lower bandwidth costs
    - Accessible to edge devices



Experimental results on the AT&T Dataset with no DP [source]

J. Konecný, H. B. McMahan, F. X. Yu, et al., Federated learning: Strategies for improving communication efficiency, ArXiv:1610.05492, October 2016. http://arxiv.org/abs/1610.05492

P. Kairouz, H.B. McMahan, B. Avent, et al., Advances and open problems in federated learning, December 2019. https://arxiv.org/abs/1912.04977

Programming Frameworks

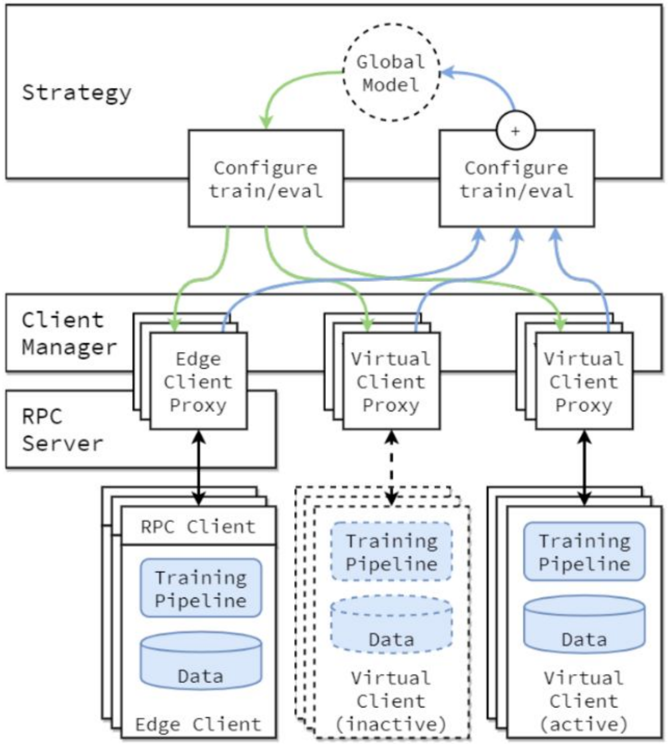
- PySyft
  - pip install syft
- TensorFlow Federated
  - pip install --upgrade tensorflow-federated
- Flower
  - pip install flwr
- Opacus
  - pip install opacus
- TenSEAL
  - pip install tenseal

Programming Frameworks (cont.)

- Flower: A Friendly Federated Learning Research Framework
  - Originated from the University of Oxford
  - A unified approach to federated learning
  - Supports a wide variety of frameworks
    - TensorFlow, PyTorch, Scikit-Learn
  - Secure Aggregation → [Salvia]()

	TFF	Syft	FedScale	LEAF	Flower
Single-node simulation	√	√	√	√	√
Multi-node execution	*	√	(√)***		√
Scalability	*		**		√
Heterogeneous clients		(√)***	**		√
ML framework-agnostic		****	****		√
Communication-agnostic					√
Language-agnostic					√
Baselines			√	√	*

Labels: * Planned / ** Only simulated
*** Only Python-based / **** Only PyTorch and/or TF/Keras



Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Parcollet, T., & Lane, N. D. (2020). Flower: A Friendly Federated Learning Research Framework. ArXiv Preprint ArXiv:2007.14390.