Federated Communication-Efficient Multi-Objective Optimization

Baris Askin Pranay Sharma Gauri Joshi Carlee Joe-Wong
Carnegie Mellon University

Motivation

Multi-objective Optimization (MOO)

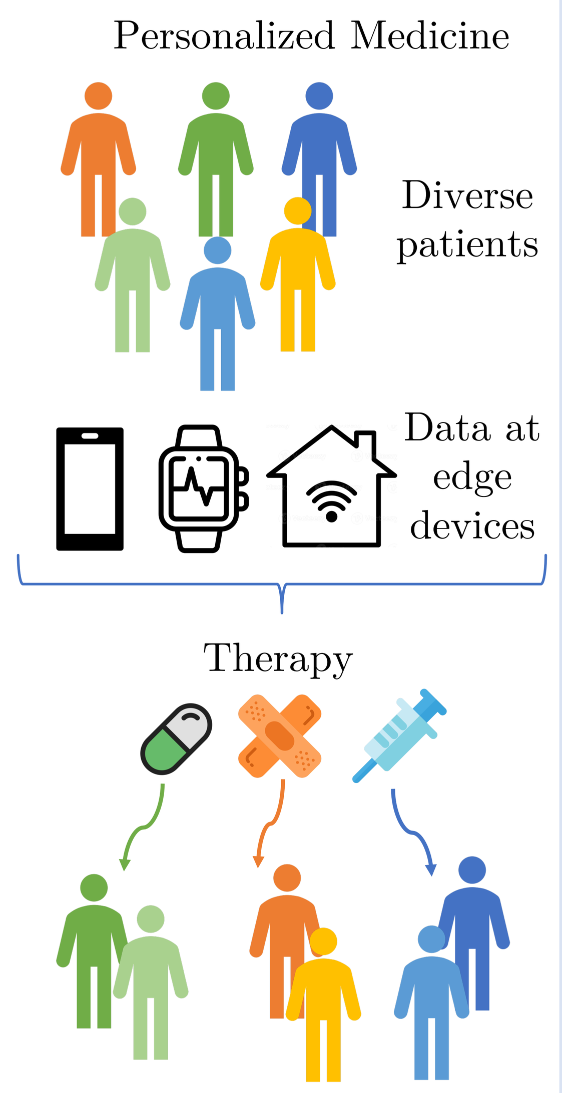
- Single model optimized for multiple objectives
- Commonly used in practice, e.g., in recommender systems: relevance, diversity, safety, profit...

Federated Learning (FL)

- Train ML models with distributed data
- Heterogeneous data across clients
- Need for client data privacy

Federated Multi-Objective Optimization

- Many real-world problems involve a federated setting and multiple objectives, e.g., in personalized medicine:
  - Multiple patients: diversity and privacy
  - Multiple objectives: precision, limited side-effects, cost-effectiveness...
- Underexplored in the literature



Problem: Federated MOO

How can we design a communication- and time-efficient training for federated multi-objective models?

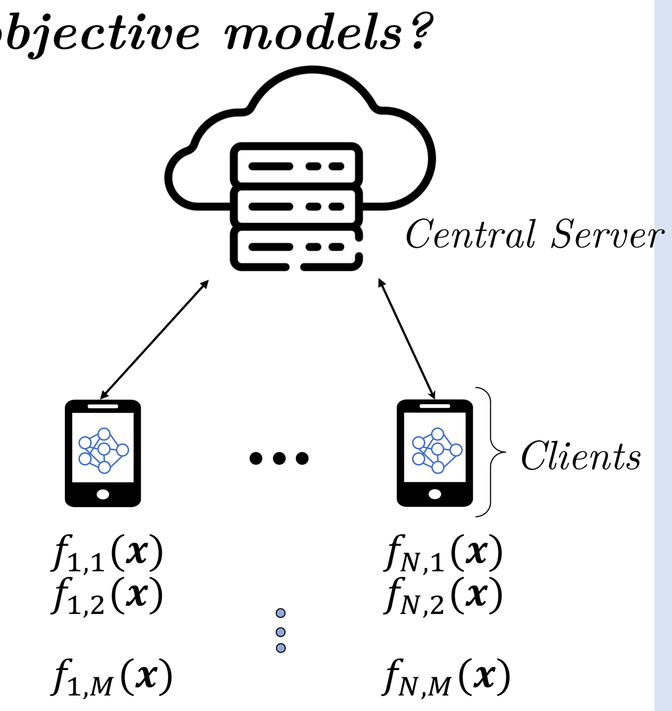
N clients and M objectives

Goal:

$$\min_{x \in \mathbb{R}^d} \boldsymbol{F}(\boldsymbol{x}) := [F_1(\boldsymbol{x}), F_2(\boldsymbol{x}), \dots, F_M(\boldsymbol{x})]$$

where global objectives are average of clients' local objectives:

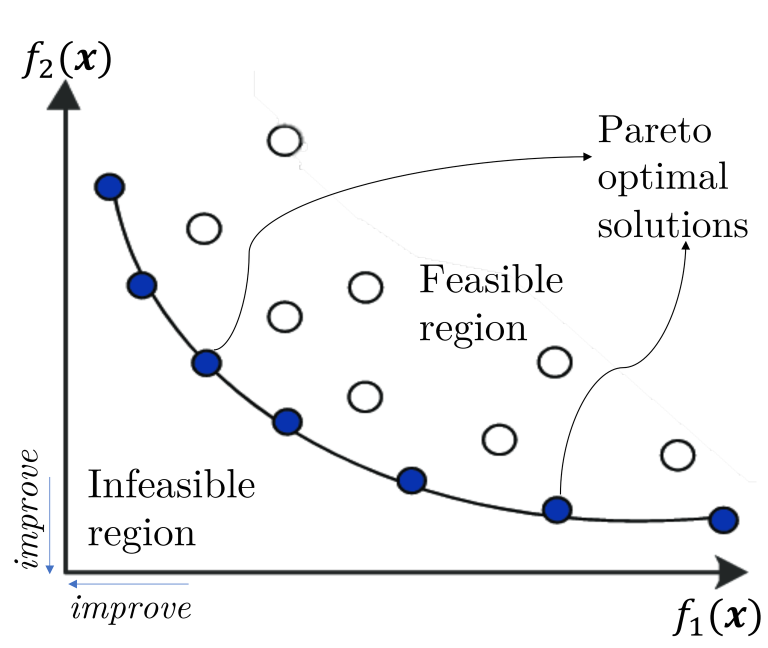
$F_k(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} f_{i,k}(\boldsymbol{x})$ for every $k \in [M]$



Background and Baselines

Multi-Objective Optimization

No $\boldsymbol{x} \in \mathbb{R}^d$ simultaneously minimizes all objectives
Trade-off across objectives



Goal: find a *Pareto optimal* $\boldsymbol{x}$

If nonconvex, find a *Pareto stationary* $\boldsymbol{x}$ that does not have a common descent direction for all objectives

Baseline Methods

Multiple Gradient Descent Algorithm (MGDA)[1]:

- *Centralized*, iterative, and gradient-based MOO solver
- Every iteration:
  - Calculate all gradients, $\{\nabla F_k(\boldsymbol{x})\}_{k=1}^{M}$
  - Find $\boldsymbol{w}^* = \min_{w \in \mathcal{S}_M} \|\sum w_k \nabla F_k(\boldsymbol{x})\|$, maximizing the minimum descent
  - $\boldsymbol{x} \leftarrow \boldsymbol{x} - \eta \sum w_k^* \nabla F_k(\boldsymbol{x})$

Federated Stochastic MGDA (FSMGDA)[2]:

- No access to the full global gradients, $\{\nabla F_k(\boldsymbol{x})\}_{k=1}^{M}$. Every client i:
  - trains M models in parallel, one for each objective, with local SGD
  - sends model-sized updates, $\{\boldsymbol{\Delta}_{i,k}\}_{k=1}^{M}$, for every objective to server
- The server averages the updates: $\boldsymbol{\Delta}_k^t \leftarrow Avg(\boldsymbol{\Delta}_{i,k}^t)$ and find $\boldsymbol{w}^* = \min_{w \in \mathcal{S}_M} \|\sum w_k \boldsymbol{\Delta}_k^t\|$

Weakness 1: High, $\Theta(Md)$, communication cost
Weakness 2: Drift across objectives' separate local trainings

References

[1] Désidéri, et al. "Multiple-Gradient Descent Algorithm (MGDA) for Multiobjective Optimization," Comptes Rendus Mathematique, 2012.
[2] Yang, et al. "Federated Multi-Objective Learning," NeurIPS, 2023.
[3] Askin, et al., "Federated Communication-Efficient Multi-Objective Optimization," AISTATS, 2025.
[4] Xiao, et al. "Direction-Oriented Multi-Objective Learning: Simple and Provable Stochastic Algorithms," NeurIPS, 2023.

Proposed Method

Federated Communication-Efficient MOO (FedCMOO)[3]

Initialize $\boldsymbol{x}^0$
For $t = 0, \dots, T-1$:
Server approximates $\boldsymbol{w}$ with $\Theta(d)$ communication with clients $(\star)$
Every active client i:
  Trains a **single** model using the weighted objective $\sum_k w_k f_{i,k}(\boldsymbol{x}_i)$
  Sends a **single** update, $\boldsymbol{\Delta}_i^t$, to the server
  $\boldsymbol{x}^{t+1} \leftarrow \boldsymbol{x}^t + Avg(\boldsymbol{\Delta}_i^t)$

$(\star)$ How to approximate $\boldsymbol{w}$:
1. Clients calculate a single stochastic gradient for each objective
2. Compress them and send to the server (randomized-SVD)
3. The server averages them and updates $\boldsymbol{w}$ with projected SGD

By finding objective weights $\boldsymbol{w}$ before local updates;
Strength 1: Communication cost, $\boldsymbol{\Theta(d)}$, does not scale with M
Strength 2: Training a single model solves objective drift problem

Theoretical Guarantees

Standard assumptions:
- Smoothness and bounded gradients
- Unbiased and bounded variance, bounded heterogeneity
- Unbiased compression operator

With an appropriate choice of learning rates, **FedCMOO converges to a Pareto stationary solution:** **(see paper for details)**

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\|\sum_k w_k^t \nabla F_k(\boldsymbol{x}^t)\right\|^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$$

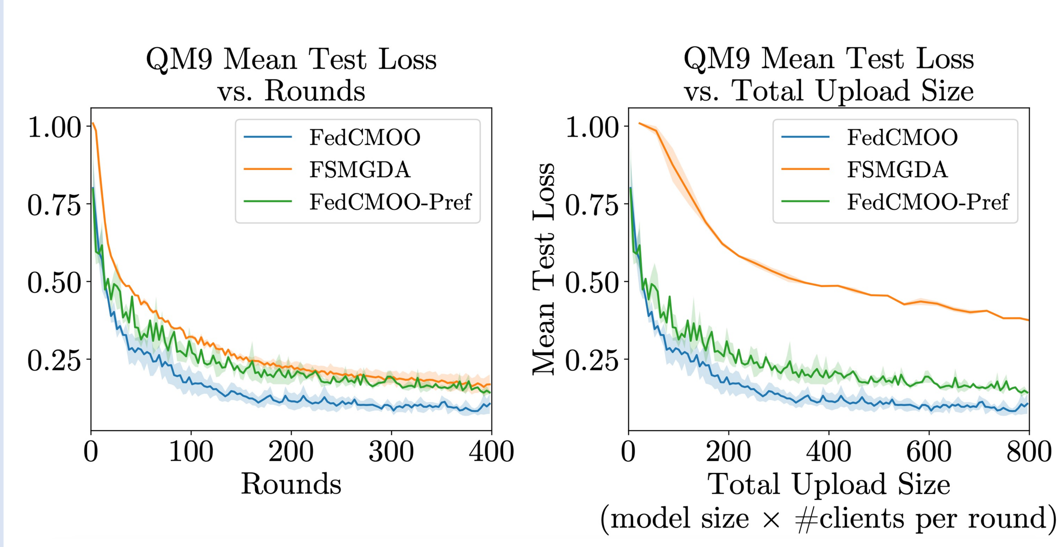
The rate matches the best-known rate for single-objective FL

Sample complexity to achieve an ϵ-close solution (M: # of objectives)

Work	Federated	Complexity	Non-standard Assumptions
SDMGrad[4]	✗	$\mathcal{O}(M^2/\epsilon^2)$	✓ None
FSMGDA[2]	✓	$\mathcal{O}(M^4/\epsilon^2)$	✗ Increasing batch size with T
Ours[3]	✓	$\mathcal{O}(M/\epsilon^2)$	✓ None

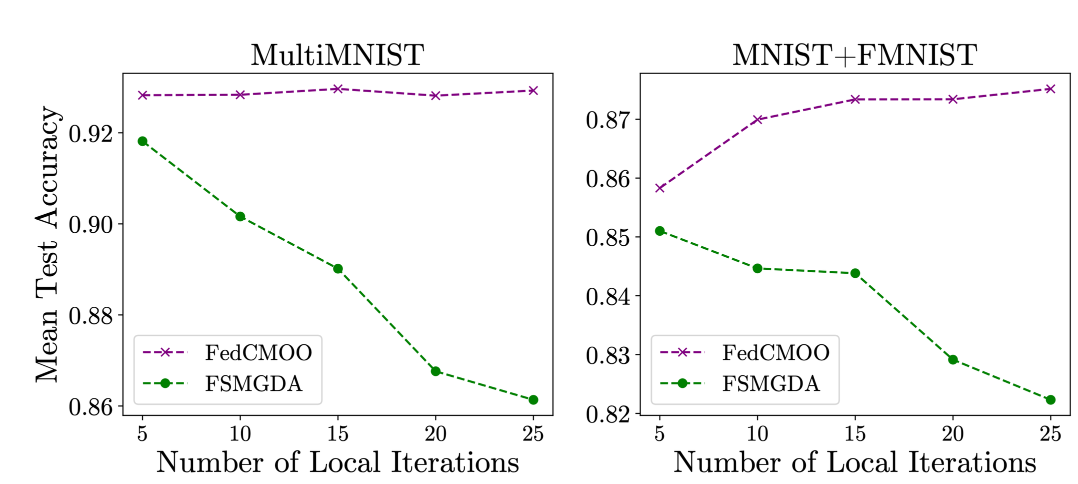
Experimental Results and FedCMOO-Pref

Experimental Results



FedCMOO trains faster than the baseline and reduces the communication cost significantly

FedCMOO is not affected by local objective drift and overperforms the baseline

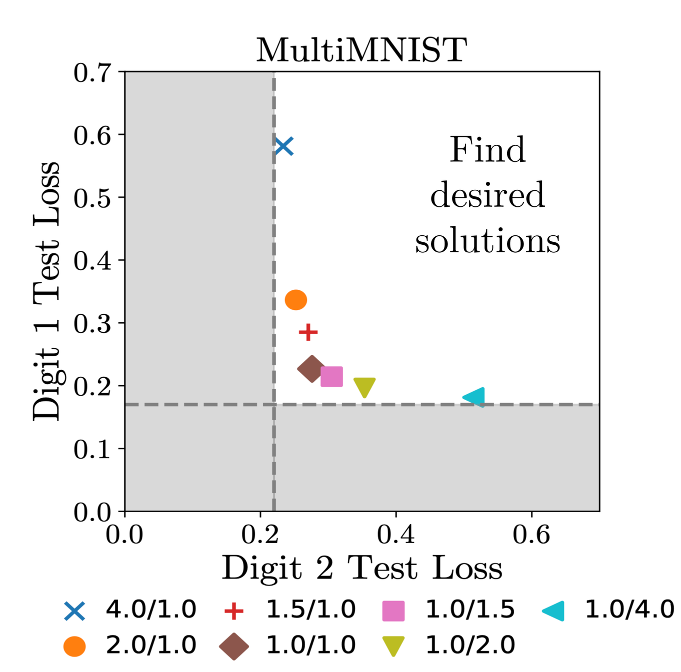


Preference-based Solutions (FedCMOO-Pref)

Extension for **a solution with desired objective trade-offs**

Low communication cost

First time in the federated setting



Takeaways

- **Communication-efficient** MOO framework in the federated setting
- **Faster convergence** by solving the objective drift problem
- **Better convergence rate** with milder assumptions
- FedCMOO-Pref: **The first preference-based MOO for FL**

askinb.github.io for the full paper & code!