# ECE 8101: Nonconvex Optimization for Machine Learning

## Lecture Note 5-2: Multi-Objective Optimization

Jia (Kevin) Liu

Associate Professor
Department of Electrical and Computer Engineering
The Ohio State University, Columbus, OH, USA

Autumn 2024

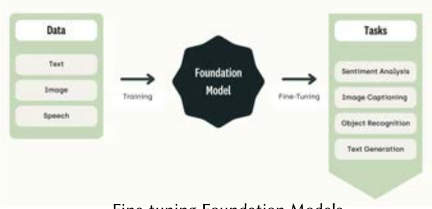
# Outline

In this lecture:

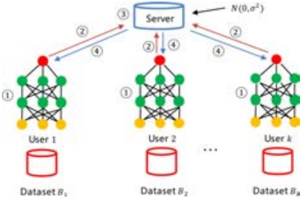
- Motivations and Formulation of Multi-Objective Optimization (MOO)
- MOO Algorithms
- Convergence Results

# Multi-Objective Optimization: Motivation

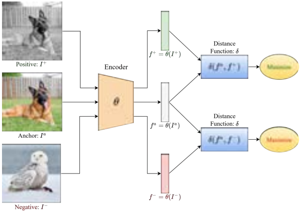
Recommender Systems




Fine-tuning Foundation Models




Federated Learning



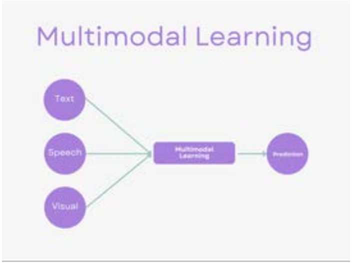

Contrastive Learning

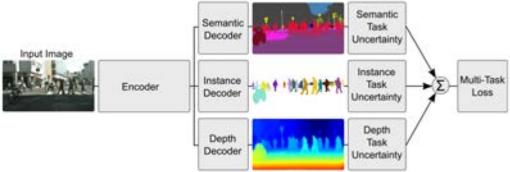
**Many learning paradigms/systems are multi-task, hence multi-objective**

# Trade-offs in MOO

Trade-offs in Real-world Problems: Many real-world problems involve optimizing multiple (potential conflicting) objectives.

- Data: multi-modal learning
- Tasks: multi-task learning
- Metrics: fairness-robustness-efficiency



# MOO Formulation and Methods

- Formulation: MOO aims at optimizing multiple objectives simultaneously, which can be mathematically cast as:

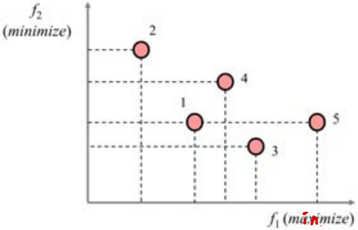
$$\min_{\mathbf{x}\in\mathcal{D}} \mathbf{F}(\mathbf{x}) := [f_1(\mathbf{x}), \cdots, f_S(\mathbf{x})],$$

where $\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^d$ is the model parameter, and $f_s : \mathbb{R}^d \to \mathbb{R}$, $s \in [S]$.

- MOO Methods
  - Gradient-Free Methods
    - ⋆ Evolutionary MOO algorithms [Zhang & Li, '07; Deb et al., '02]
    - ⋆ Bayesian MOO algorithms [Balakaria et al., '20; Laumanns et al., '02]
  - Gradient-Based Methods
    - ⋆ Multi-gradient descent algorithm (MGDA) with full gradients [Mukai, '80; Fliege & Svaiter '00; Desideri '12]
    - ⋆ Stochastic multi-gradient descent algorithms (SMGDA) with stochastic gradients [Liu & Vicente, '21; Zhou et al., '22; Fernando et al., '23]

# Notions of Optimality in MOO

- Single-objective optimization (scalar-valued): $\mathbf{x}$ dominates $\mathbf{y}$ if $f(\mathbf{x}) < f(\mathbf{y})$
  $\rightarrow$ Goal: Find an optimal solution $\mathbf{x}^*$ such that $f(\mathbf{x}^*) \leq f(\mathbf{x})$, $\forall \mathbf{x} \in \mathcal{D}$
- Multi-objective optimization (vector-valued): Not partially ordered
  - Which one is better: $[0, 1, 1]$ vs $[1, 0, 1]$ vs $[0, 0, 0]$.




Which is the dominant one?

- 1 vs. 2
- 1 vs. 5
- 1 vs. 4

- Lexicographical order in some special MOO problems: the order depends on the order of the first element in an alphabet that differs)

# Notions of Optimality in MOO

## Definition 1 (Dominance)

$\mathbf{x}$ dominates $\mathbf{y}$ iff $f_s(\mathbf{x}) \leq f_s(\mathbf{y}), \forall s \in [S]$ and $f_s(\mathbf{x}) < f_s(\mathbf{y}), \exists s \in [S]$.

## Definition 2 (Pareto Optimality)

A solution $\mathbf{x}^*$ is Pareto optimal if it is not dominated by any other solution.

## Definition 3 (Weak Pareto Optimality)

A solution $\mathbf{x}^*$ is weakly Pareto optimal if there does not exist $\mathbf{x}$ such that $f_s(\mathbf{x}) < f_s(\mathbf{x}^*)$, $\forall s \in [S]$, i.e., impossible to improve all objectives simultaneously.

## Definition 4 (Pareto Stationarity)

A solution $\mathbf{x}$ is said to be Pareto stationary if there is no common descent direction $\mathbf{d} \in \mathbb{R}^d$ such that $\nabla f_s(\mathbf{x})^\top \mathbf{d} < 0, \forall s \in [S]$.

$(-\nabla f_s(\underline{x}))^\top \underline{d} > 0$

# Pareto Front

- Pareto front (or boundary): The set of all Pareto-optimal solutions $\mathcal{X}^*$

$$f_1(\mathbf{x}) = 1 - e^{-\sum_{i=1}^{d}(x_i - \frac{1}{\sqrt{d}})^2}$$
$$f_2(\mathbf{x}) = 1 - e^{-\sum_{i=1}^{d}(x_i + \frac{1}{\sqrt{d}})^2}$$
$$d = 2, -4 \le x_1, x_2 \le 4$$

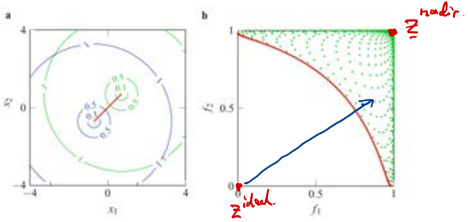



**Fig. 1.2** (a) Contour lines of the functions $f_1$ (*green*) and $f_2$ (*blue*) along with the set of Pareto optimal decisions (*red*). (b) Objective feasible region (represented by *green dots*) and the Pareto front (*red*) for Example 1.2

- Nadir objective vector: $\mathbf{z}^{\text{nadir}} = [\sup_{\mathbf{x}\in\mathcal{X}^*} f_1(\mathbf{x}), \ldots, \sup_{\mathbf{x}\in\mathcal{X}^*} f_S(\mathbf{x})]^\top$ (UB)
- Ideal objective vector: $\mathbf{z}^{\text{ideal}} = [\inf_{\mathbf{x}\in\mathcal{X}^*} f_1(\mathbf{x}), \ldots, \inf_{\mathbf{x}\in\mathcal{X}^*} f_S(\mathbf{x})]^\top$ (LB)

# Philosophical Classes for Solving MOO Problems

Consider the availability of a decision maker (DM) in an MOO problem domain

- No-preference methods: No DM is expected to be available and no preference information is assumed to be known
  - Example 1: Method of global criterion: $\min \|f(\mathbf{x}) - \mathbf{z}^{\text{ideal}}\|$ s.t. $\mathbf{x} \in \mathcal{D}$
  - Example 2: Multi-gradient descent algorithm (MGDA)
- A priori methods: Preference information is first asked from DM, and then a solution best satisfying the preferences is found
- A posteriori methods: A representative set of Pareto-optimal solutions is first found, then DM much choose one of them
- Interactive methods: DM is allowed to search for the most preferred solution iteratively. In each iteration, DM is shown Pareto-optimal solution(s) and DM describes how the solution(s) could be improved. Information given by DM is used to generate new Pareto-optimal solution in the next iteration.
- Hybrid methods: A mixture of some of the above
  - Example: Weighted-Chebyshev MGDA [Momma, Dong, & Liu, ICML'22]

# A priori Methods

- Utility Function Methods
  - Assume a utility function $u(\cdot)$ is available to DM

    $u(x) \geq u(y)$ if $x \succeq y$
- Lexicographic Methods
  - Assume objectives can be ranked in the order of "importance"
- Scalarization Methods
  - Reformulate MOO as a single-objective optimization (SOO) problem, such that optimal SOO solutions are Pareto-optimal in the original MOO problem
  - Often requires that every Pareto-optimal solution of the MOO problem can be achieved by some parameter setting of the SOO problem (Pareto front exploration) – useful in a posterior methods

    $\Longleftrightarrow$

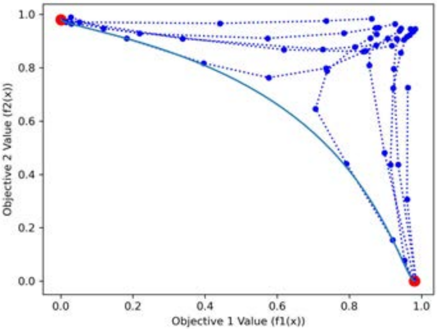
# Scalarization Methods: Linear Scarlization (LS)

Basic Idea: Scalarize a vector-valued objective into a scalar-valued objective by linearly combining each objective with a user-supplied weights.

- MOO:
$$\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x}))$$
- LS:
$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) = \sum_{i=1}^{m} w_i f_i(\mathbf{x})$$
  - $w_i \geq 0$: A priori weight for the $i$-th objective (e.g., chosen in proportion to the relative importance of the objective)

# Scalarization Methods: Linear Scarlization (LS)

- Pros: Simple objective structure, preserve nice properties (e.g., convexity, smoothness, etc.)
- Cons: 1) May be difficult to choose weights in practice; 2) Cannot explore Pareto front in the case with non-convex objectives (only finds the convex hull of the objective set)



# Scalarization Methods: $\epsilon$-Constraint Scalarization (EC)

Basic Idea: Keep one objective and treat the rest of the objectives as constraints

- MOO:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x}))$$

- EC:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} & \quad f_j(\mathbf{x}) \\ \text{subject to} & \quad f_i(\mathbf{x}) \leq \epsilon_i, \forall i \neq j \end{aligned}$$

  - $\epsilon_i > 0$: A desired upper bound of the $i$-th objective

- Pros: Allow the use of powerful constrained optimization algorithms

- Cons: Incorrect choices of $\boldsymbol{\epsilon}$ may lead to infeasibility; hard for PF exploration

# Scalarization Methods: Weighted Chebyshev (WC)

Tchebyshe--

Basic Idea: Minimize the (weighted) $\ell_\infty$ norm of the vector-valued objective.

- MOO:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x}))$$

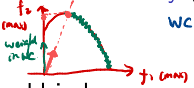
$\|\underline{x}\|_\infty = \max_i w_i |x_i|$

- WC:

$$\underset{\mathbf{x}}{\text{minimize}} \; \underset{i \in [m]}{\text{maximize}} \quad w_i(f_i(\mathbf{x}) \underbrace{- u_i}_{\text{optional.}}),$$

  - $w_i \in [0, 1]$: A priori weight for the $i$-th objective (e.g., chosen in proportion to the relative importance of the objective)

- Pros: 1) Any Pareto-optimal solution can be found by solving a WC problem for some $\mathbf{w}$; 2) Any $\mathbf{w}$-WC solution corresponds to a weakly Pareto optimal solution of the original MOO problem regardless of its convexity (necessary and sufficient for Pareto front exploration!)




WC: $[0, 1]^T$

- Cons: More complex objective (min-max) and could induce non-smoothness

# Algorithm Design for Non-convex MOO

- **Pareto Stationarity:** A solution $\mathbf{x}$ is said to be Pareto stationary if there is no common descent direction $\mathbf{d} \in \mathbb{R}^d$ such that $\nabla f_s(\mathbf{x})^\top \mathbf{d} < 0, \forall s \in [S]$.
  - Pareto-stationary solution allows ties. Hence, if found, it is a necessary condition for weak Pareto optimality
  - Similar to using stationary solution in place of optimal solution in non-convex single-objective optimization, one can use Pareto stationarity as a relaxed criterion in solving non-convex MOO problems
  - Convergence Metric for Pareto Stationarity: $\|\nabla \mathbf{f}(\mathbf{x})\boldsymbol{\lambda}^*\|^2$, where $\boldsymbol{\lambda}^*$ is the minimal of $\min_{\boldsymbol{\lambda}} \|\nabla \mathbf{f}(\mathbf{x})\boldsymbol{\lambda}\|^2$, which motivates the multi-gradient descent algorithm (MGDA) – a non-preference MOO method

# MGDA

Multi-Gradient Descent Algorithm (MGDA): Search a common descent direction [Mukai, '80, Fliege & Svaiter '00, Desideri '12].

1. Find an optimal weight $\boldsymbol{\lambda}^*$ of gradients $\nabla \mathbf{F}(\mathbf{x}) \triangleq \{\nabla f_s(\mathbf{x}), \forall s \in [S]\}$ by solving

$$\boldsymbol{\lambda}^*(\mathbf{x}) = \underset{\boldsymbol{\lambda} \in C}{\operatorname{argmin}} \|\nabla \mathbf{F}(\mathbf{x})\boldsymbol{\lambda}\|^2.$$

2. A common descent direction can be chosen as: $\mathbf{d} = -\nabla \mathbf{F}(\mathbf{x})\boldsymbol{\lambda}^{*}$.

3. Performs the iterative update rule:

$$\mathbf{x} \leftarrow \mathbf{x} + \eta \mathbf{d} = \mathbf{x} - \eta \nabla \mathbf{F}(\mathbf{x})\boldsymbol{\lambda}$$

until a Pareto-stationary point is reached, where $\eta$ is a learning rate.

# MGDA Derivation:

step 1: Determine the worst descent amount given moving dir $\underline{d}$ (i.e., $\underline{x}_t$, $\underline{x}_{t+1}$ are given, and $\underline{d} = \underline{x}_{t+1} - \underline{x}_t$).

step 2: Find the optimal moving direction to minimize worst descent dir.

Recall descent: $f_i(\underline{x}_{t+1}) - f_i(\underline{x}_t) = -\delta_i \quad (\delta_i > 0)$

$$\min_{\underline{x}_{t+1}} \left[ \begin{array}{l} \min\limits_{\delta \geq 0} \; -\delta \\ f_1(\underline{x}_{t+1}) - f_1(\underline{x}_t) = -\delta_1 \leq -\delta \\ \quad \vdots \\ f_m(\underline{x}_{t+1}) - f_m(\underline{x}_t) = -\delta_m \leq -\delta \end{array} \right]$$

step 1. Consider inner problem.

$$\min_{\delta \geq 0} -\delta$$

$$\text{s.t. } f_1(\underline{x}_{t+1}) - f(\underline{x}_t) \leq -\delta \quad \leftarrow \lambda_1 \geq 0$$

$$\vdots$$

$$f_m(\underline{x}_{t+1}) - f(\underline{x}_t) \leq -\delta \quad \leftarrow \lambda_m \geq 0$$

Lagrangian: $-\delta - \sum_{i=1}^{m} \lambda_i \left( f_i(\underline{x}_t) - f_i(\underline{x}_{t+1}) - \delta \right)$

Dual problem: $$\max_{\underline{\lambda} \geq 0} \left[ \min_{\delta \geq 0} \left[ -\delta - \sum_{i=1}^{m} \lambda_i \left( f_i(\underline{x}_t) - f_i(\underline{x}_{t+1}) - \delta \right) \right] \right]$$

$$= \max_{\underline{\lambda} \geq 0} \left[ \min_{\delta \geq 0} \left[ \delta \left( -1 + \sum_{i=1}^{m} \lambda_i \right) - \sum_{i=1}^{m} \lambda_i \left( f_i(\underline{x}_t) - f_i(\underline{x}_{t+1}) \right) \right] \right]$$

Note: the inner problem is minimized when $\sum_{i=1}^{m} \lambda_i = 1$.

So, the dual problem becomes:

$$\min_{\underline{\lambda} \in \Delta^m_+} \left[ \sum_{i=1}^m \lambda_i [f_i(\underline{x}_{t+1}) - f_i(\underline{x}_t)] \right] \overset{\text{FO-Taylor Approx.}}{\approx} \min_{\underline{\lambda} \in \Delta^m_+} \underline{\lambda}^T \underbrace{\underline{\underline{F}}(\underline{x}_t)^T}_{\text{Jacobian}} \underbrace{(\underline{x}_{t+1} - \underline{x}_t)}_{\text{moving dir}}$$

$\uparrow$

"standard simplex": $\{\lambda_i \geq 0 : \sum_{i=1}^m \lambda_i = 1\}$

$$= \underline{\lambda}^{*T} \underline{\underline{F}}(\underline{x}_t)^T (\underline{x}_{t+1} - \underline{x}_t) \quad (1)$$

Step 2: Find the optimal moving dir. to minimize worst descent.

$$\min_{\underline{x}} \left[ \underbrace{\underline{\lambda}^{*T} \underline{\underline{F}}(\underline{x}_t)^T (\underline{x} - \underline{x}_t)}_{\text{FO-Approx}} + \underbrace{\frac{1}{2\eta} \|\underline{x} - \underline{x}_t\|^2}_{\text{regularization}} \right]$$

Take der. w.r.t. $\underline{x}$ & set it to zero:

$$\underline{\lambda}^* \underline{\underline{F}}(\underline{x}_t) + \frac{1}{\eta}(\underline{x} - \underline{x}_t) = 0.$$

Solve for $\underline{x} \Rightarrow \quad \underline{x}_{t+1} = \underline{x}_t - \eta \underline{\underline{F}}(\underline{x}_t) \underline{\lambda}^* \quad (2).$

Plugging (2) into (1) yields:

$$\underline{\lambda}^* = \min_{\underline{\lambda} \in \Delta^m_+} \|\underline{\underline{F}}(\underline{x}_t) \underline{\lambda}\|^2. \qquad \blacksquare$$

# Convergence of MGDA (Non-Convex)

- In the MGDA method, perform the following "backtracking" line search:
  - Choose $\beta \in (0,1)$. In iteration $k$, compute a step-size $s_k \in (0,1]$ as the maximum value in the following set:

$$\mathcal{S}_k := \left\{ s = \frac{1}{2^j} \middle| j \in \mathbb{N}_0, \mathbf{f}(\mathbf{x}_k + s\mathbf{d}_k) \leq \mathbf{f}(\mathbf{x}_k) + \beta s \nabla \mathbf{f}(\mathbf{x}_k)\mathbf{d}_k \right\}$$

- Assume $f_i(\cdot)$ is $L_i$-smooth and let $L_{\max} := \max\{L_1, \ldots, L_m\}$.
- Let $t_{\min} := \min\{(1-\beta)/(2L_{\max}), 1\}$

## Theorem 5 ([Fliege, Vaz, & Vicente, '19])

*Suppose at least one of the objectives $f_1, \ldots, f_m$ is bounded from below. Let $f_i^{\min}$ be the lower bound of $f_i(\cdot)$. Let $F^{\min} := \min_{\forall i} f_i^{\min}$ and $F_0^{\max} = \max_{\forall i} f_i^{\max}(\mathbf{x}_0)$. The sequence $\{\mathbf{x}_k\}$ generated by the MGDA method with backtracking satisfies*

$$\min_{0 \leq l \leq k-1} \|\mathbf{d}_l\|^2 \leq \frac{F_0^{\max} - F^{\min}}{Mk} = \mathcal{O}(1/k)$$ Same as GD.

# SMGDA

Stochastic Multi-Gradient Descent Algorithm (SMGDA): Use stochastic gradients $\nabla f_s(\mathbf{x}, \xi)$ as approximations of true gradient $\nabla f_s(\mathbf{x})$ [Liu and Vicente, '21].

$O(\frac{1}{\sqrt{k}})$

convex & strongly convex

1. Solve for $\boldsymbol{\lambda}^*$ of stochastic gradients $\nabla \mathbf{F}(\mathbf{x}) \triangleq \{\nabla f_s(\mathbf{x}, \xi_s), \forall s \in [S]\}$:

$$\boldsymbol{\lambda}^*(\mathbf{x}) = \underset{\boldsymbol{\lambda} \in C}{\operatorname{argmin}} \|\nabla \mathbf{F}(\mathbf{x})\boldsymbol{\lambda}\|^2.$$

2. A estimated common descent direction can be chosen as: $\mathbf{d} = -\nabla \mathbf{F}(\mathbf{x})\boldsymbol{\lambda}^*$.

3. Performs the iterative update rule:

$$\mathbf{x} \leftarrow \mathbf{x} + \eta \mathbf{d} = \mathbf{x} - \eta \nabla \mathbf{F}(\mathbf{x})\boldsymbol{\lambda}$$

until a Pareto-stationary point is reached, where $\eta$ is a learning rate.

# Our General Federated MOO Framework

- A FMOO systems with $M$ clients and $S$ objectives collectively:

$$\min_{\mathbf{x}} \ \mathrm{Diag}(\mathbf{F}\mathbf{A}^\top),$$

$$\mathbf{F} \triangleq \begin{bmatrix} f_{1,1} & \cdots & f_{1,M} \\ \vdots & \ddots & \vdots \\ f_{S,1} & \cdots & f_{S,M} \end{bmatrix}_{S\times M}, \qquad \mathbf{A} \triangleq \begin{bmatrix} a_{1,1} & \cdots & a_{1,M} \\ \vdots & \ddots & \vdots \\ a_{S,1} & \cdots & a_{S,M} \end{bmatrix}_{S\times M}$$

  - Matrix $\mathbf{F}$ groups all potential objectives $f_{s,i}(\mathbf{x})$ for each task $s$ at each client $i$
  - $\mathbf{A} \in \{0,1\}^{S\times M}$ is a *binary* objective indicator matrix, with each element $a_{s,i} = 1$ if task $s$ is of client $i$'s interest and $a_{s,i} = 0$ otherwise.
  - For each task $s \in [S]$, the global objective function $f_s(\mathbf{x})$ is the average of local objectives over all related clients, i.e., $f_s(\mathbf{x}) \triangleq \frac{1}{|R_s|}\sum_{i\in R_s} f_{s,i}(\mathbf{x})$, where $R_s = \{i : a_{s,i} = 1, i \in [M]\}$.

# Different Cases of This FMOO Framework

- If each client has only one distinct objective, i.e., $\mathbf{A} = \mathbb{I}_M$, $S = M$
  - Each objective $f_s(\mathbf{x}), s \in [S]$ is optimized only by client $s$
  - Corresponds to the conventional multi-task learning and federated learning

- If all clients share the same $S$ objectives, i.e., $\mathbf{A}$ is an all-one matrix
  - FMOL reduces to solving a MOO problem collaboratively with decentralized data in a federated fashion
  - E.g., jointly optimize fairness, privacy, and accuracy, ...

- If each client has a different subset of objectives (i.e., objective heterogeneity), FMLO allows distinct preferences at each client
  - The most general case, where FMLO allows distinct preferences at each client.
  - E.g., each customer group in a recommender system has different combinations of shopping preferences, such as product price, brand, delivery speed, etc.

# FMOO Algorithms: FMGDA and FSMGDA

Federated (Stochastic) Multi-Gradient Descent Alg. [Yang et al., NeurIPS'23]

**At Each Client $i$:** MOO – FedAvg

1. Synchronize local models $\mathbf{x}_{s,i}^{t,0} = \mathbf{x}_t, \forall s \in S_i$. Then perform local updates: for all $s \in S_i$, for $k = 1, \dots, K$:

$$\text{(FMGDA): } \mathbf{x}_{s,i}^{t,k} = \mathbf{x}_{s,i}^{t,k-1} - \eta_L \nabla f_{s,i}(\mathbf{x}_{s,i}^{t,k-1}),$$

$$\text{(FSMGDA): } \mathbf{x}_{s,i}^{t,k} = \mathbf{x}_{s,i}^{t,k-1} - \eta_L \nabla f_{s,i}(\mathbf{x}_{s,i}^{t,k-1}, \xi_i^{t,k}).$$

2. Return accumulated updates $\{\Delta_{s,i}^t, s \in S_i\}$ to server: (FMGDA): $\Delta_{s,i}^t = \sum_{k\in[K]} \nabla f_{s,i}(\mathbf{x}_{s,i}^{t,k})$, or (FSMGDA): $\Delta_{s,i}^t = \sum_{k\in[K]} \nabla f_{s,i}(\mathbf{x}_{s,i}^{t,k}, \xi_i^{t,k})$.

**At the Server:**

1. Compute $\Delta_s^t = \frac{1}{|R_s|} \sum_{i\in R_s} \Delta_{s,i}^t, \forall s \in [S]$, where $R_s = \{i : a_{s,i} = 1, i \in [M]\}$.
2. Compute $\boldsymbol{\lambda}_t^* \in [0,1]^S$ by solving $\min_{\boldsymbol{\lambda}_t \ge \mathbf{0}} \| \sum_{s\in[S]} \lambda_s^t \Delta_s^t \|^2$, s.t. $\sum_{s\in[S]} \lambda_s^t = 1$.
3. Let $\mathbf{d}_t = \sum_{s\in[S]} \lambda_s^{t,*} \Delta_s^t$ and let $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \mathbf{d}_t$, with a global learning rate $\eta_t$.

# Convergence Results: FMGDA [Yang et al., NeurIPS'23]

## Theorem 7 (FMGDA for Non-Convex Case)

- *$L$-Lipschitz smoothness:* $\|\nabla f_s(\mathbf{x}) - \nabla f_s(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, s \in [S]$
- *Bounded Gradient:* $\|\nabla f_{s,i}(\mathbf{x})\|^2 \leq G^2, \forall s \in [S], i \in [M]$
- *Choosing* $\eta_t = \eta \leq \frac{3}{2(1+L)}$*, the sequence* $\{\mathbf{x}_t\}$ *output by FMGDA satisfies:* $\min_{t\in[T]} \|\bar{\mathbf{d}}_t\|^2 \leq \frac{16(f_s^0 - f_s^{\min})}{T\eta} + \delta$*, where* $\delta \triangleq \frac{16\eta_L^2 K^2 L^2 G^2 (1+S^2)}{\eta}$.

## Corollary 1 (Convergence Rate of FMGDA)

*Let* $\eta_t = \eta$*,* $\forall t$*, and let* $\eta_L = \mathcal{O}(1/\sqrt{T})$*, FMGDA achieves a Pareto-stationary convergence rate of* $(1/T)\sum_{t\in[T]} \|\bar{\mathbf{d}}_t\|^2 = \mathcal{O}(1/T)$.

# Convergence Results: FSMGDA [Yang et al., NeurIPS'23]

## Theorem 8 (FSMGDA for Non-Convex Case)

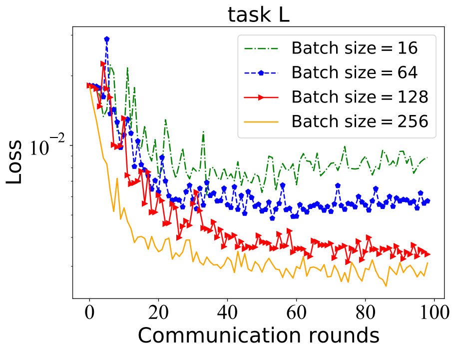
- *$(\alpha, \beta)$-Smooth:* $\exists \alpha, \beta > 0$ *s.t.* $\mathbb{E}[\|\nabla f(\mathbf{x}, \xi) - \nabla f(\mathbf{y}, \xi^{'})\|^2] \leq \alpha \|\mathbf{x} - \mathbf{y}\|^2 + \beta \sigma^2$
- *Unbiased Stochastic Gradient:* $\mathbb{E}[\nabla f_{s,i}(\mathbf{x}, \xi)] = \nabla f_{s,i}(\mathbf{x}), \forall s \in [S], i \in [M]$
- *Bounded Stochastic Gradient:* $\mathbb{E}[\|\nabla f_{s,i}(\mathbf{x}, \xi)\|^2] \leq D^2, \forall s \in [S], i \in [M]$
- *Let* $\eta_t = \eta \leq \frac{3}{2(1+L)}$, *the sequence* $\{\mathbf{x}_t\}$ *output by FSMGDA satisfies:* $\min_{t \in [T]} \mathbb{E} \left\| \bar{\mathbf{d}}_t \right\|^2 \leq \frac{2S\left(f_s^0 - f_s^{\min}\right)}{\eta T} + \delta$, *where* $\delta = L\eta S^2 D^2 + S(\alpha \eta_L^2 K^2 D^2 + \beta \sigma^2)$.

## Corollary 2 (Convergence Rate of FSMGDA)

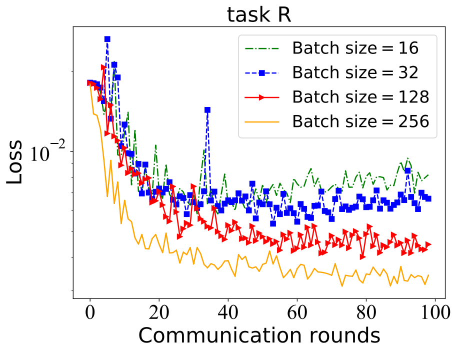
*Let* $\eta_t = \eta = \mathcal{O}(1/\sqrt{T}), \forall t$ *and* $\eta_L = \mathcal{O}\left(1/T^{1/4}\right)$, *and if* $\beta = \mathcal{O}(\eta)$, *FSMGDA achieves a Pareto-stationarity convergence rate of* $\min_{t \in [T]} \mathbb{E}\|\bar{\mathbf{d}}_t\|^2 = \mathcal{O}(1/\sqrt{T})$.

# Numerical Results: Two-Task Case

- Training loss convergence in terms of communication rounds with different batch-sizes under non-i.i.d. data partition in MultiMNIST.



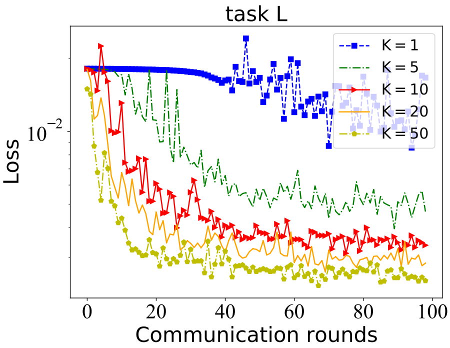

Task Left



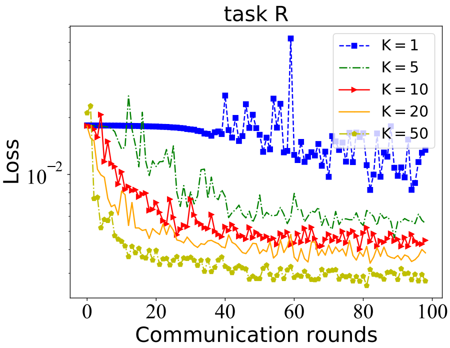

Task Right

# Numerical Results: Two-Task Case

- The impacts of local update number $K$ on training loss convergence in terms of communication rounds.



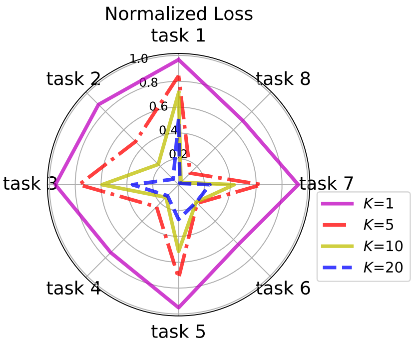

Task Left



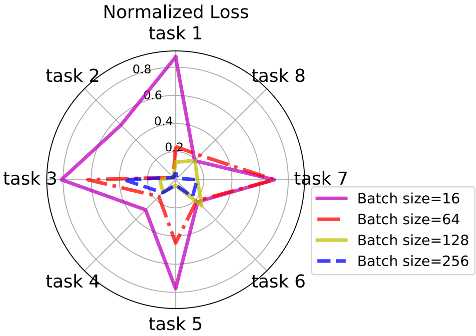

Task Right

# Numerical Results: Eight-Task Case

- Normalized loss with the River Flow datasets.



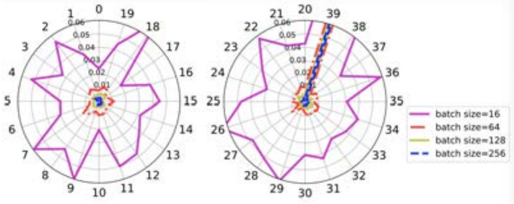

Impact of local step number $K$.



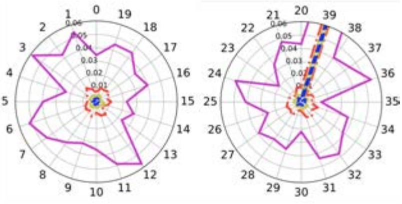

Impact of batch sizes.

# Numerical Results: 40-Task Case

- Experiments on CelebA dataset with 40 binary facial classification problems




Impact of batch sizes (non-i.i.d. case).



Impact of batch sizes (i.i.d. case).

# Next Class

Project Presentations