A Potential Game Framework in Federated Learning

Ziqi Wang
Joint work with K. Liu and E. Zuazua

Chair for Dynamics, Control, Machine Learning, and Numerics
Alexander von Humboldt-Professorship
Friedrich-Alexander-Universität Erlangen-Nürnberg

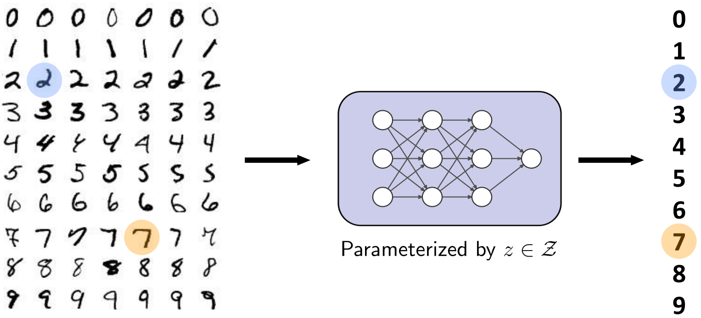
The Mathematics of Scientific Machine Learning and Digital Twins
Erice, 20.11.2025

Friedrich-Alexander-Universität
DYNAMICS, CONTROL, MACHINE LEARNING AND NUMERICS

MODCONFLEX
Modelling and control of flexible structures interacting with fluids

Funded by the European Union

Centralized learning



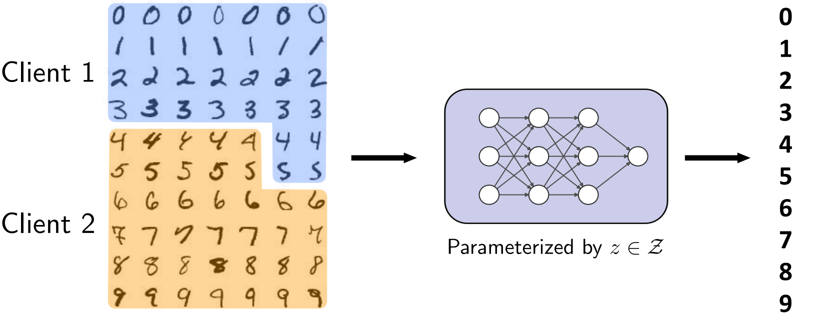
$x \in \mathcal{X}$ $f_z : \mathcal{X} \to \mathcal{Y}$ $y \in \mathcal{Y}$

- Learning/training seeks to optimize the parameters z by

$$\min_{z \in \mathcal{Z}} \ell(z, \mathcal{D}) := \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \operatorname{dist}\left(f_z(x_i), y_i\right),$$

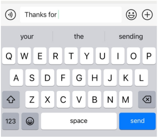
$$z^+ = z - \eta \nabla_z \ell(z, \mathcal{D}).$$

Centralized learning



$x \in \mathcal{X}$ $\qquad$ $f_z : \mathcal{X} \to \mathcal{Y}$ $\qquad$ $y \in \mathcal{Y}$

- **Constraints**: data are distributed and private.



Problem formulation of federated learning

Federated learning (FL) [McMahan et al. 2017]

FL is a machine learning setting where multiple clients collaborate to solve a machine learning problem under the coordination of a central server, without sharing their private data.

- FL training with m clients:

$$\min_{z \in \mathbb{R}^n} \sum_{i=1}^{m} \rho_i \ell_i(z, \mathcal{D}_i), \tag{P1}$$

  - ℓ_i is client i's loss function,
  - $\mathcal{D}_i$ is client i's private dataset,
  - ρ_i is the weight assigned by the central server, *e.g.*, $\rho_i = |\mathcal{D}_i| \,/\, |\mathcal{D}|$.

Problem formulation of federated learning

Federated learning (FL) [McMahan et al. 2017]

FL is a machine learning setting where multiple clients collaborate to solve a machine learning problem under the coordination of a central server, without sharing their private data.

- FL training with m clients:

$$\min_{z \in \mathbb{R}^n} \sum_{i=1}^{m} \rho_i \ell_i(z, \mathcal{D}_i), \tag{P1}$$

  - ℓ_i is client i's loss function,
  - $\mathcal{D}_i$ is client i's private dataset,
  - ρ_i is the weight assigned by the central server, *e.g.*, $\rho_i = |\mathcal{D}_i| \,/\, |\mathcal{D}|$.

Problem formulation of federated learning

- FL training with m clients:

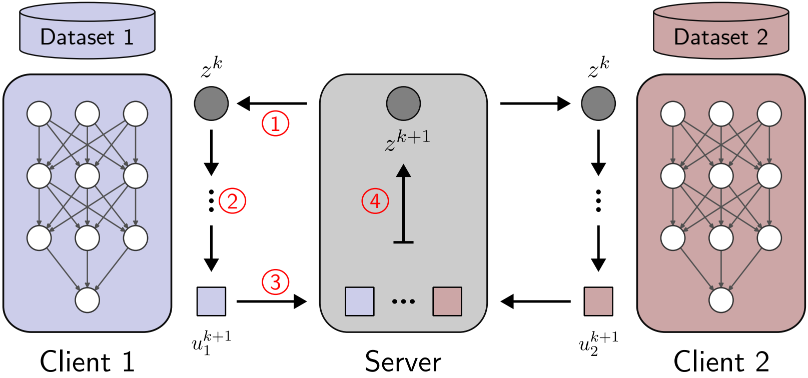
$$\min_{z \in \mathbb{R}^n} \sum_{i=1}^{m} \rho_i \ell_i(z, \mathcal{D}_i), \tag{P1}$$

- By introducing a consensus constraint:

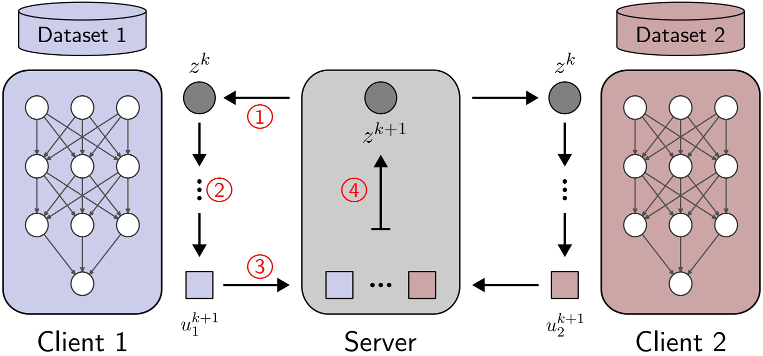
$$\min_{u_i, z \in \mathbb{R}^n} \sum_{i=1}^{m} \rho_i \ell_i(u_i, \mathcal{D}_i), \text{ s.t. } u_i = z, \forall\, i \in [m], \tag{P2}$$

  - u_i is the local model parameter held by client i,
  - z is the global model parameter held by the server.

Training in FL: Exchange model parameters instead of data



Clients' local training

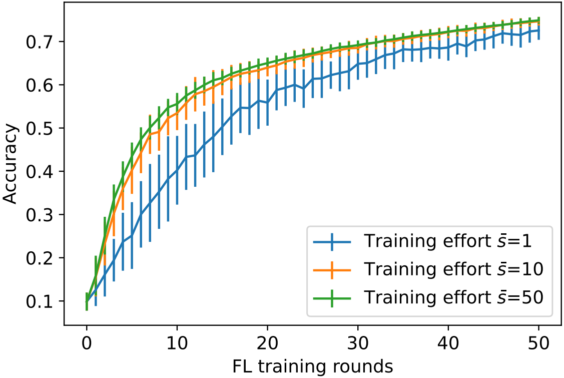


- In Step ②: each client performs local training (*e.g.*, gradient descent):

$$u_i^{t+1} = \text{LocalTraining}_i(z^t, s_i^t),$$

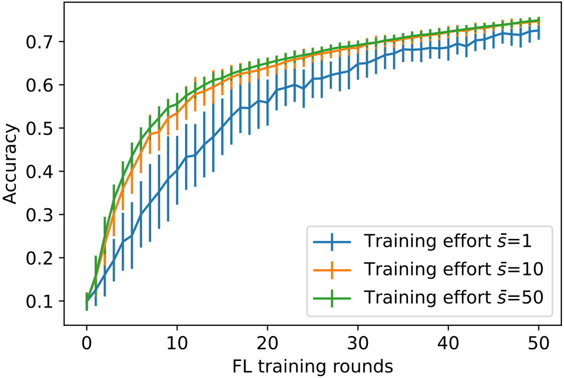
where s_i^t is client i's training effort (*e.g.*, # gradient descent steps).

Motivation: Clients' local training requires incentives



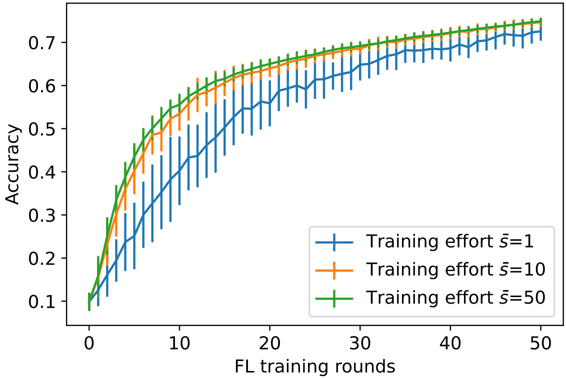
- **Finding:** Greater average effort $\bar{s}$ leads to better performance.
- **Challenge:** Increasing training effort leads to higher costs for clients.
- **Solution:** Incentivize clients by rewarding them based on their efforts.

Motivation: Clients' local training requires incentives



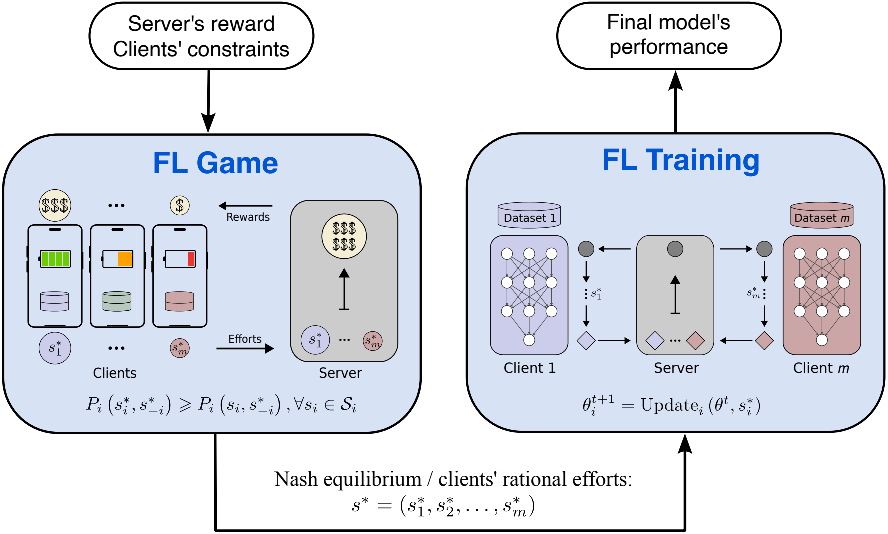
- **Finding:** Greater average effort $\bar{s}$ leads to better performance.
- **Challenge:** Increasing training effort leads to higher costs for clients.
- **Solution:** Incentivize clients by rewarding them based on their efforts.

Motivation: Clients' local training requires incentives



- **Finding:** Greater average effort $\bar{s}$ leads to better performance.
- **Challenge:** Increasing training effort leads to higher costs for clients.
- **Solution:** Incentivize clients by rewarding them based on their efforts.

Illustration of the FL game and training



Game theory preliminaries

Definition (m-player game Γ)

A game Γ consists of:

- A finite set of players $[m] \coloneqq \{1, \ldots, m\}$.
- A strategy set $\mathcal{S}_i$ for each player $i \in [m]$.
- The players' payoff functions $P_i : \mathcal{S} \to \mathbb{R}$, with $\mathcal{S} = \prod_{i \in [m]} \mathcal{S}_i$.

Definition (Nash equilibrium (NE))

In game Γ, a point $s^* \in \mathcal{S}$ is an NE if the following inequality holds:

$$P_i\left(s_i^*, s_{-i}^*\right) \geq P_i\left(s_i, s_{-i}^*\right), \ \forall\, s_i \in \mathcal{S}_i, \forall\, i \in [m].$$

FL game Γ_{FL}

Definition (FL game Γ_{FL})

The FL game Γ_{FL} contains:

- A set of players (clients) $[m] := \{1, \ldots, m\}$;
- Each client's strategy set $\mathcal{S}_i \subseteq \mathbb{R}_+^T$;
- Each client's payoff function $P_i : \mathcal{S} \to \mathbb{R}$, with $\mathcal{S} = \prod_{i \in [m]} \mathcal{S}_i$:

$$P_i(s_i, s_{-i}) = \sum_{t=1}^{T} r_\lambda(s_i^t, s_{-i}^t) - c_i(s_i^t).$$

- **Key question:** How do the NEs and the FL training performance vary as the reward parameter λ changes?

FL game Γ_{FL}

Definition (FL game Γ_{FL})

The FL game Γ_{FL} contains:

- A set of players (clients) $[m] := \{1, \dots, m\}$;
- Each client's strategy set $\mathcal{S}_i \subseteq \mathbb{R}_+^T$;
- Each client's payoff function $P_i : \mathcal{S} \to \mathbb{R}$, with $\mathcal{S} = \prod_{i \in [m]} \mathcal{S}_i$:

$$P_i(s_i, s_{-i}) = \sum_{t=1}^{T} r_\lambda(s_i^t, s_{-i}^t) - c_i(s_i^t).$$

- **Key question:** How do the NEs and the FL training performance vary as the reward parameter λ changes?

Game formulation: Clients' strategy sets

Total-budget strategy set

$$\mathcal{S}_i = \left\{ s_i \in \mathbb{R}_+^T \,\middle|\, b_i \leq \sum_{t \in [T]} s_i^t \leq B_i \right\},$$

where b_i and B_i denote the minimum and maximum total efforts.

Stationary strategy set

$$\mathcal{S}_i = \left\{ s_i \in \mathbb{R}_+^T \,\middle|\, q_i \leq s_i^t = s_i^\tau \leq Q_i,\ \forall\, t, \tau \in [T] \right\},$$

where q_i and Q_i represent the minimum and maximum per-round efforts.

- It is reasonable to assume that each client's training effort is bounded.
- For instance, each client's device has limited battery life.

Game formulation: Clients' strategy sets

Total-budget strategy set

$$\mathcal{S}_i = \left\{ s_i \in \mathbb{R}_+^T \,\middle|\, b_i \leq \sum_{t \in [T]} s_i^t \leq B_i \right\},$$

where b_i and B_i denote the minimum and maximum total efforts.

Stationary strategy set

$$\mathcal{S}_i = \left\{ s_i \in \mathbb{R}_+^T \,\middle|\, q_i \leq s_i^t = s_i^\tau \leq Q_i,\ \forall\, t, \tau \in [T] \right\},$$

where q_i and Q_i represent the minimum and maximum per-round efforts.

- It is reasonable to assume that each client's training effort is bounded.
- For instance, each client's device has limited battery life.

Game formulation: Reward in clients' payoff functions

- The payoff function for each client i is the reward minus the cost:

$$P_i(s_i, s_{-i}) = \sum_{t=1}^{T} r_\lambda(s_i^t, s_{-i}^t) - c_i(s_i^t).$$

- The reward function is defined as

$$r_\lambda(s_i^t, s_{-i}^t) = p_\lambda^t s_i^t, \text{ where } p_\lambda^t = \lambda \sum_{j=1}^{m} \rho_j s_j^t.$$

  - The unit price p_λ^t depends on the total weighted effort of all clients.
  - The reward factor $\lambda > 0$ is a parameter controlled by the server.

- **Intuition:** The reward mechanism encourages all clients to contribute more effort collectively.

Game formulation: Reward in clients' payoff functions

- The payoff function for each client i is the reward minus the cost:

$$P_i(s_i, s_{-i}) = \sum_{t=1}^{T} r_\lambda(s_i^t, s_{-i}^t) - c_i(s_i^t).$$

- The reward function is defined as

$$r_\lambda(s_i^t, s_{-i}^t) = p_\lambda^t s_i^t, \text{ where } p_\lambda^t = \lambda \sum_{j=1}^{m} \rho_j s_j^t.$$

  - The unit price p_λ^t depends on the total weighted effort of all clients.
  - The reward factor $\lambda > 0$ is a parameter controlled by the server.

- **Intuition:** The reward mechanism encourages all clients to contribute more effort collectively.

Game formulation: Reward in clients' payoff functions

- The payoff function for each client i is the reward minus the cost:

$$P_i(s_i, s_{-i}) = \sum_{t=1}^{T} r_\lambda(s_i^t, s_{-i}^t) - c_i(s_i^t).$$

- The reward function is defined as

$$r_\lambda(s_i^t, s_{-i}^t) = p_\lambda^t s_i^t, \text{ where } p_\lambda^t = \lambda \sum_{j=1}^{m} \rho_j s_j^t.$$

  - The unit price p_λ^t depends on the total weighted effort of all clients.
  - The reward factor $\lambda > 0$ is a parameter controlled by the server.

- **Intuition:** The reward mechanism encourages all clients to contribute more effort collectively.

Game formulation: Cost in clients' payoff functions

- The payoff function for each client i is the reward minus the cost:

$$P_i(s_i, s_{-i}) = \sum_{t=1}^{T} r_\lambda(s_i^t, s_{-i}^t) - c_i(s_i^t).$$

- The cost function can be specified as

$$c_i(s_i^t) = \alpha_i s_i^t \quad \text{or} \quad c_i(s_i^t) = \alpha_i (s_i^t)^2,$$

where $\alpha_i > 0$ is the marginal cost rate.

- **Intuition:** Clients are willing to contribute effort when their costs are adequately compensated by the received rewards.

Game formulation: Cost in clients' payoff functions

- The payoff function for each client i is the reward minus the cost:

$$P_i(s_i, s_{-i}) = \sum_{t=1}^{T} r_\lambda(s_i^t, s_{-i}^t) - c_i(s_i^t).$$

- The cost function can be specified as

$$c_i(s_i^t) = \alpha_i s_i^t \quad \text{or} \quad c_i(s_i^t) = \alpha_i (s_i^t)^2,$$

where $\alpha_i > 0$ is the marginal cost rate.

- **Intuition:** Clients are willing to contribute effort when their costs are adequately compensated by the received rewards.

Game formulation: Cost in clients' payoff functions

- The payoff function for each client i is the reward minus the cost:

$$P_i(s_i, s_{-i}) = \sum_{t=1}^{T} r_\lambda(s_i^t, s_{-i}^t) - c_i(s_i^t).$$

- The cost function can be specified as

$$c_i(s_i^t) = \alpha_i s_i^t \quad \text{or} \quad c_i(s_i^t) = \alpha_i (s_i^t)^2,$$

where $\alpha_i > 0$ is the marginal cost rate.

- **Intuition:** Clients are willing to contribute effort when their costs are adequately compensated by the received rewards.

Existence of Nash equilibria

Theorem (Monderer and Shapley 1996)

Let $w = (w_i)_{i \in [m]}$ be a strictly positive vector. The game Γ is said to be a w-potential game if there exists a function $P \colon \mathcal{S} \to \mathbb{R}$ such that

$$P_i(s_i, s_{-i}) - P_i(x_i, s_{-i}) = w_i \left[P(s_i, s_{-i}) - P(x_i, s_{-i})\right], \ \forall\, s_i, x_i \in \mathcal{S}_i, \ \forall\, i \in [m].$$

If $s^* \in \mathcal{S}$ is a global maximizer of P, then s^* is an NE of the game Γ.

Existence of NE in Γ_{FL} (Liu, Wang, and Zuazua 2024)

The game Γ_{FL} admits a w-potential given by

$$P_{\mathsf{FL}}(s) = \sum_{i=1}^{m} \sum_{t=1}^{T} \left(\frac{\lambda}{2} \rho_i^2 (s_i^t)^2 - \rho_i c_i(s_i^t) \right) + \sum_{t=1}^{T} \frac{\lambda}{2} \left(\sum_{i=1}^{m} \rho_i s_i^t \right)^2, \quad s \in \mathcal{S},$$

where $c_i(\cdot)$ is client i's local cost, and $w_i = 1/\rho_i$ for $i \in [m]$. If $\mathcal{S}$ is compact and $c_i(\cdot)$ is lower semicontinuous, then Γ_{FL} possesses at least one NE.

Uniqueness of the NE in the stationary game $\hat{\Gamma}_{\text{FL}}$

Stationary strategy set

$$\mathcal{S}_i = \left\{ s_i \in \mathbb{R}_+^T \,\middle|\, q_i \leq s_i^t = s_i^\tau \leq Q_i,\ \forall\, t, \tau \in [T] \right\},$$

where q_i and Q_i represent the minimum and maximum efforts per round.

Definition (Stationary FL game $\hat{\Gamma}_{\text{FL}}$)

The stationary FL game $\hat{\Gamma}_{\text{FL}}$ contains:

- A set of players (clients) $[m] := \{1, \ldots, m\}$;
- Each client's strategy set $\hat{\mathcal{S}}_i = [q_i, Q_i]$;
- Each client's payoff function $\hat{P}_i : \hat{\mathcal{S}} \to \mathbb{R}$, with $\hat{\mathcal{S}} = \prod_{i \in [m]} \hat{\mathcal{S}}_i$, given by

$$\hat{P}_i(s_i, s_{-i}) = \lambda s_i \sum_{j=1}^{m} \rho_j s_j - \alpha_i s_i^2.$$

Uniqueness of the NE in the stationary game $\hat{\Gamma}_{\mathsf{FL}}$

Stationary strategy set

$$\mathcal{S}_i = \left\{ s_i \in \mathbb{R}_+^T \,\middle|\, q_i \le s_i^t = s_i^\tau \le Q_i, \ \forall\, t, \tau \in [T] \right\},$$

where q_i and Q_i represent the minimum and maximum efforts per round.

Definition (Stationary FL game $\hat{\Gamma}_{\mathsf{FL}}$)

The stationary FL game $\hat{\Gamma}_{\mathsf{FL}}$ contains:

- A set of players (clients) $[m] \coloneqq \{1, \ldots, m\}$;
- Each client's strategy set $\hat{\mathcal{S}}_i = [q_i, Q_i]$;
- Each client's payoff function $\hat{P}_i : \hat{\mathcal{S}} \to \mathbb{R}$, with $\hat{\mathcal{S}} = \prod_{i \in [m]} \hat{\mathcal{S}}_i$, given by

$$\hat{P}_i(s_i, s_{-i}) = \lambda s_i \sum_{j=1}^{m} \rho_j s_j - \alpha_i s_i^2.$$

Uniqueness of the NE in the stationary game $\hat{\Gamma}_{\mathsf{FL}}$

Theorem (Liu, Wang, and Zuazua 2024)

For $\lambda > 0$, the game $\hat{\Gamma}_{\mathsf{FL}}$ has a unique NE, except at the critical point λ^*, which satisfies

$$\sum_{i=1}^{m} \frac{\lambda^*}{2\alpha_i/\rho_i - \lambda^*} = 1$$

If $\lambda = \lambda^*$, the game may have infinitely many NEs.

Corollary (Precise description of the unique NE)

The following statements hold:

1. *If $\lambda \in (0, \lambda_1)$, the unique NE of the game $\hat{\Gamma}_{\mathsf{FL}}$ is $s^* = (q_i)_{i\in[m]}$;*
2. *If $\lambda \in (\lambda_1, \lambda^*)$, the unique NE s^* satisfies $\bar{s}^* < c_1$;*
3. *If $\lambda \in (\lambda^*, \lambda_2)$, the unique NE s^* satisfies $\bar{s}^* > c_2$;*
4. *If $\lambda \in (\lambda_2, +\infty)$, the unique NE is $s^* = (Q_i)_{i\in[m]}$.*

Uniqueness of the NE in the stationary game $\hat{\Gamma}_{\mathsf{FL}}$

Theorem (Liu, Wang, and Zuazua 2024)

For $\lambda > 0$, the game $\hat{\Gamma}_{\mathsf{FL}}$ has a unique NE, except at the critical point λ^*, which satisfies

$$\sum_{i=1}^{m} \frac{\lambda^*}{2\alpha_i/\rho_i - \lambda^*} = 1$$

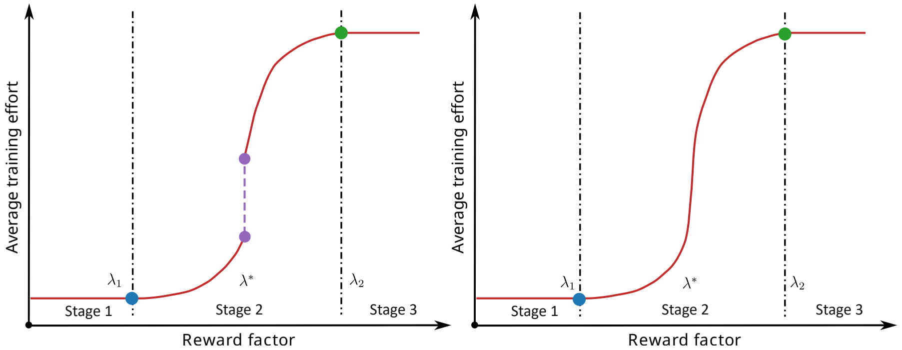
If $\lambda = \lambda^*$, the game may have infinitely many NEs.

Corollary (Precise description of the unique NE)

The following statements hold:

1. *If $\lambda \in (0, \lambda_1)$, the unique NE of the game $\hat{\Gamma}_{\mathsf{FL}}$ is $s^* = (q_i)_{i\in[m]}$;*
2. *If $\lambda \in (\lambda_1, \lambda^*)$, the unique NE s^* satisfies $\bar{s}^* < c_1$;*
3. *If $\lambda \in (\lambda^*, \lambda_2)$, the unique NE s^* satisfies $\bar{s}^* > c_2$;*
4. *If $\lambda \in (\lambda_2, +\infty)$, the unique NE is $s^* = (Q_i)_{i\in[m]}$.*

Uniqueness of the NE in the stationary game $\hat{\Gamma}^h_{\mathsf{FL}}$

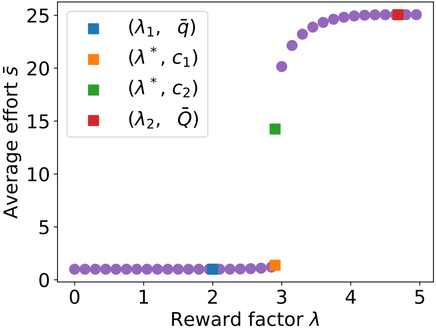


(a) $c_1 < c_2$. (b) $c_1 \geqslant c_2$.

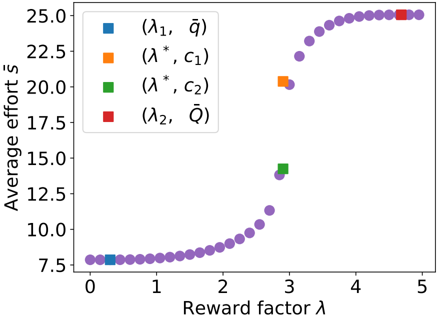
If $\alpha > 0$ and $\rho = 1/m$ are identical for all m clients:

$$\lambda^* = \frac{2m}{m+1}\alpha, \quad c_1 = \max_{i\in[m]} \{q_i\}\,, \quad c_2 = \min_{i\in[m]} \{Q_i\}\,.$$

Simulation: Evolution of NEs and critical reward factors



(a) Scenario 1: $c_1 < c_2$.

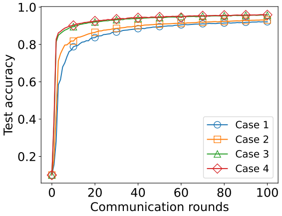


(b) Scenario 2: $c_1 > c_2$.

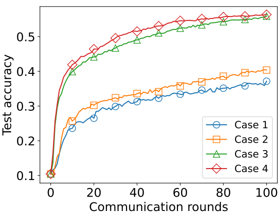
- Activation point λ_1, jump point λ^*, and saturation point λ_2.

Simulation: FL training performance across four NE cases

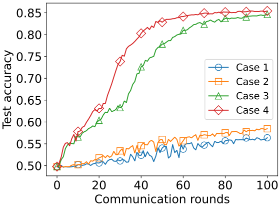
- **Case 1:** selected before the activation point λ_1.
- **Case 2:** selected before the jump point λ^*.
- **Case 3:** selected after the jump point λ^* (**recommended**).
- **Case 4:** selected after the saturation point λ_2.



(a) MNIST



(b) CIFAR-10



(c) IMDB

Thanks for your attention!

arXiv

GitHub

Friedrich-Alexander-Universität
DYNAMICS, CONTROL, MACHINE LEARNING AND NUMERICS

MODCONFLEX
Modelling and control of flexible structures interacting with fluids

Funded by the European Union