We show that in decentralized federated learning, even if you lose an agent, you can still converge to a well-performing model

Project website

Adaptive Fill-in: How to Mitigate the Loss of an Agent in Decentralized Federated Learning

Ignacy Stępka, Kacper Trębacz, Nicholas Gisolfi, James K. Miller, Artur Dubrawski
Carnegie Mellon University, Pittsburgh, PA, USA

RISS | ROBOTICS INSTITUTE SUMMER SCHOLARS

Carnegie Mellon University Robotics Institute

Introduction

Motivation

- **Privacy:** Data can't be shared directly (e.g., hospitals, regulations)
- **Solution:** Use distributed learning to share models, not data
- **Objective:** Converge to a well-performing model on all agents

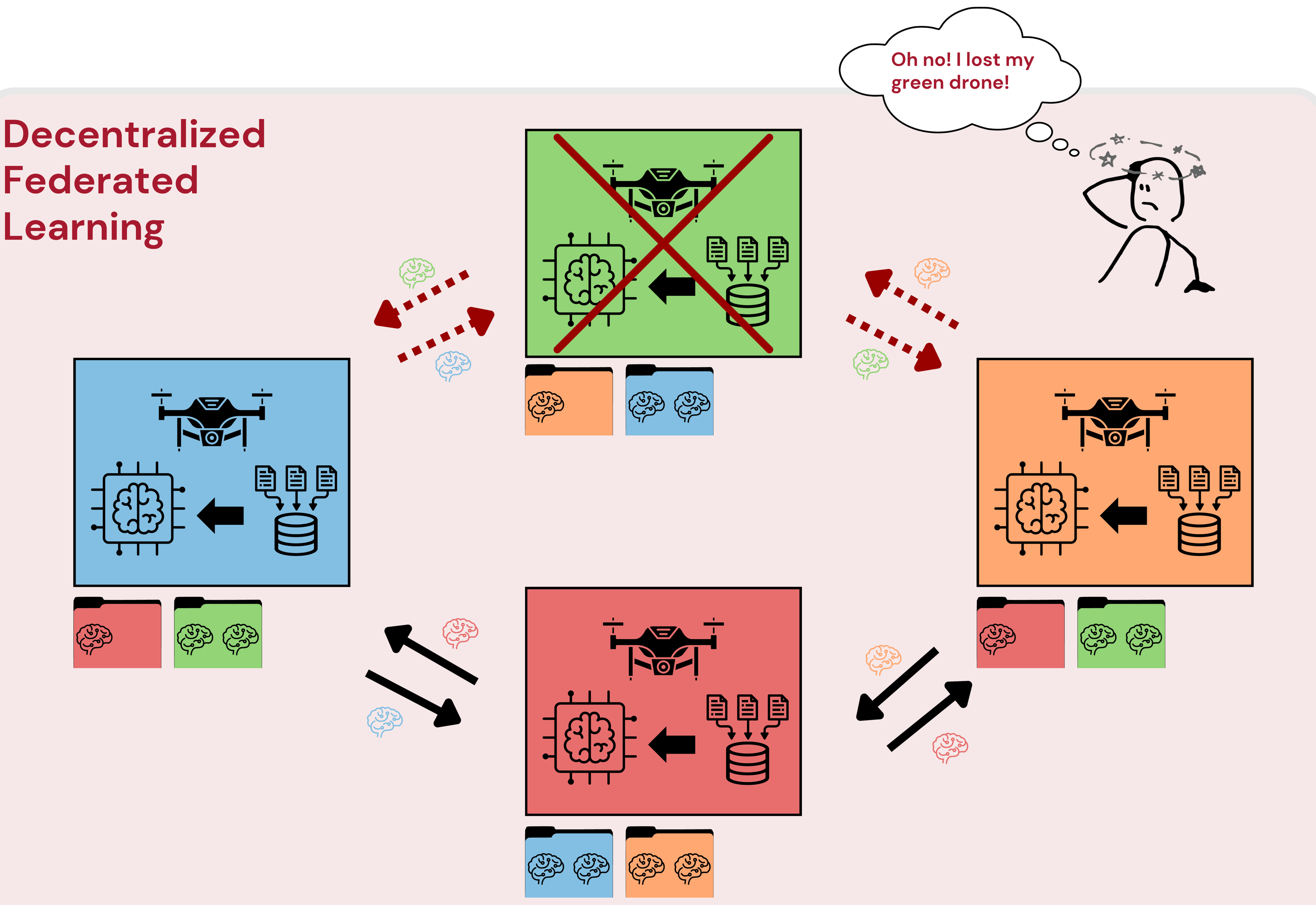
Problem Setting

- **Data distribution:** Each agent has access to some unique data
- **Collaboration:** Agents share latest models with their neighbors
- **Regularization:** Agents consider neighbors' models in their loss
- **Challenge:** One agent may be permanently lost during training

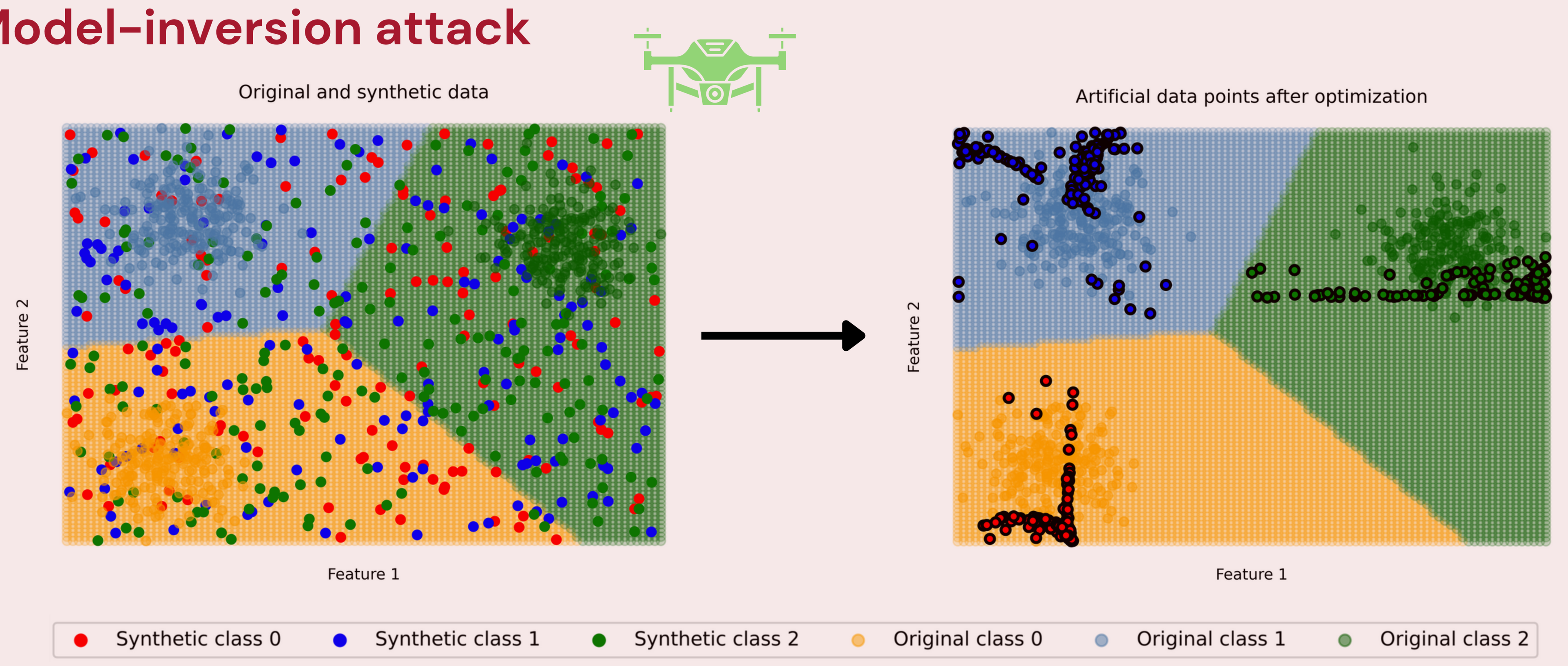
Idea

- Use the destroyed agent's model to create its virtual copy
- Approximate training data distribution via model-inversion attack
- Deploy new virtual agent with created synthetic dataset

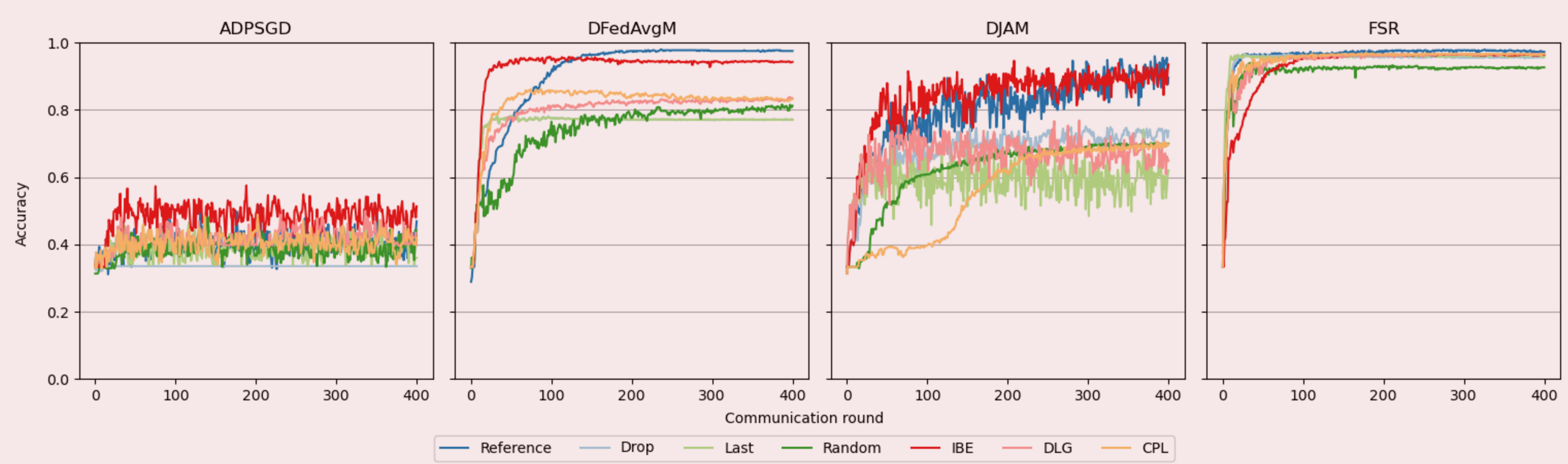
Decentralized Federated Learning



Model-inversion attack



Convergence



Method

- Every agent optimizes the same loss function via GD

$$\theta^{t+1} := \theta^t - \eta \nabla_{\theta^t} L(\theta^t; X, Y)$$

- After each communication round, agents train their model on local data until it (approx) converges to a local stationary point

$$\nabla_\theta \mathcal{L}_d(\theta, X, Y) - \epsilon = 0$$

- Create synthetic data points with random labels

$$X_{\mathtt{synth}} \sim \mathtt{Uniform}[0,1] \quad Y_{\mathtt{synth}} \sim \mathtt{Uniform}\{0,1,\ldots,C\}$$

- Optimize synthetic data points until the gradient of the loss function w.r.t. parameters is again zero using:

$$X^{t+1}_{\mathtt{synth}} := X^t_{\mathtt{synth}} - \eta \nabla_{X^t_{\mathtt{synth}}} L(\theta; X^t_{\mathtt{synth}}, Y_{\mathtt{synth}})$$

- Use the new synthetic dataset to train the model of the neighbor and proceed with the distributed optimization process

Gradient Leakage based attack methods

- Implicit Bias Exploitation (IBE)

$$\mathcal{L}_{IBE} = \mathcal{L}_d + \lambda \mathcal{L}_{prior}$$

- Deep Leakage Gradient (DLG) [5]

$$\mathcal{L}_{DLG} = \|\nabla W' - \nabla W\|^2 + \lambda \mathcal{L}_{prior}$$

- Client Private Leakage (CPL) [6]

$$\mathcal{L}_{CPL} = \|\nabla W' - \nabla W\|^2 + \lambda_1 \|f(x_{synth}) - \hat{y}\|^2 + \lambda_2 \mathcal{L}_{prior}$$

Prior term (optional)

$$\mathcal{L}_{prior} = \sum_{i=1}^{d} \mathrm{ReLU}(x-1) + \mathrm{ReLU}(-x)$$

Gradient from update history

$$\nabla W = \frac{\theta_t - \theta_{t-1}}{\eta}$$

Conclusions

- Active strategies with virtual agents lead to better results
- IBE on average is the best aid for agent loss
- DLG and CPL perform worse than IBE, but there is room for improvement in gradient estimation technique
- Further investigation into more complex datasets is needed (see additional results on the website)
- Theoretical analysis is crucial going forward

References

[1] Ovi et al. 2023 "A Comprehensive Study of Gradient Inversion Attacks in Federated Learning and Baseline Defense Strategies"
[2] Almeida et al. 2018 "Distributed Jacobi Asynchronous Method for Learning Personal Models"
[3] Tsun et al. 2021 "Decentralized Federated Averaging"
[4] Good 2024 "Trustworthy Learning using Uncertain Interpretation of Data"
[5] Zhu et al. 2019 "Deep Leakage from Gradients"
[6] Wei et al. 2020 " Framework for Evaluating Gradient Leakage Attacks in Federated Learning"

Results

Iris

	Reference	Drop	Last	Random	IBE	DLG	CPL
ADPSGD	0.47 ± 0.18	0.34 ± 0.05	0.36 ± 0.06	0.41 ± 0.16	0.51 ± 0.22	0.40 ± 0.16	0.42 ± 0.11
DFedAvgM	0.98 ± 0.02	0.77 ± 0.12	0.77 ± 0.12	0.81 ± 0.06	$\mathbf{0.94 \pm 0.02}$	0.83 ± 0.11	0.83 ± 0.10
DJAM	0.90 ± 0.09	0.74 ± 0.24	0.62 ± 0.13	0.70 ± 0.10	$\mathbf{0.94 \pm 0.03}$	0.65 ± 0.14	0.70 ± 0.08
FSR	0.97 ± 0.02	0.96 ± 0.03	0.96 ± 0.03	0.93 ± 0.01	0.96 ± 0.01	$\mathbf{0.97 \pm 0.03}$	0.97 ± 0.03

Wine

	Reference	Drop	Last	Random	IBE	DLG	CPL
ADPSGD	0.47 ± 0.13	0.43 ± 0.17	0.44 ± 0.14	0.50 ± 0.15	0.54 ± 0.20	0.50 ± 0.16	0.50 ± 0.16
DFedAvgM	0.98 ± 0.01	0.81 ± 0.15	0.81 ± 0.15	0.84 ± 0.05	0.93 ± 0.03	0.90 ± 0.07	0.91 ± 0.06
DJAM	0.79 ± 0.16	0.73 ± 0.27	0.47 ± 0.14	0.75 ± 0.19	0.80 ± 0.16	0.72 ± 0.16	0.77 ± 0.14
FSR	0.92 ± 0.03	0.91 ± 0.11	0.87 ± 0.11	0.86 ± 0.14	0.93 ± 0.04	0.80 ± 0.23	0.85 ± 0.17

Global accuracy on a test set after 300 rounds of peer-to-peer communications. Dense communication graph, best results out of 5-fold hyperparameters search on each method and patching strategy and three random seeds.

We gratefully acknowledge our sponsors for making this work possible. This material is based upon work supported by the U.S. Army Research Office and the U.S. Army Futures Command under Contract No. W911NF-20-D-0002. The content of the information does not necessarily reflect the position or the policy of the government and no official endorsement should be inferred. We are also extremely grateful for the work Rachel Burcin and Dr. John Dolan put in organizing RISS program, making this work possible.