# SoteriaFL: A Unified Framework for Private Federated Learning with Communication Compression

Zhize Li

https://zhizeli.github.io

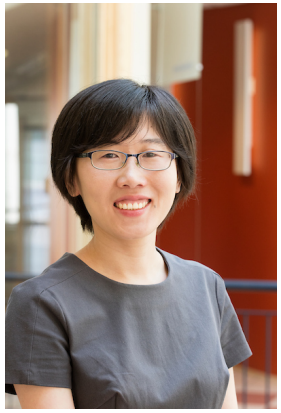
Carnegie Mellon University

NeurIPS 2022

Joint work with

Haoyu Zhao
Princeton


Boyue Li
CMU

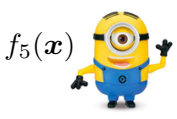


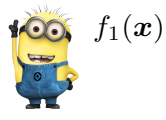
Yuejie Chi
CMU

# Problem

Empirical Risk Minimization (ERM) in Federated Learning (FL) over a dataset $\mathcal{D} = \cup_i \mathcal{D}_i$.

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) := \frac{1}{\boldsymbol{n}} \sum_{i=1}^{n} f_i(\boldsymbol{x}), \quad \text{where } f_i(\boldsymbol{x}) := \frac{1}{\boldsymbol{m}} \sum_{\boldsymbol{z}\in\mathcal{D}_i} \ell(\boldsymbol{x};\boldsymbol{z}).$$







$n$ = **number of clients**

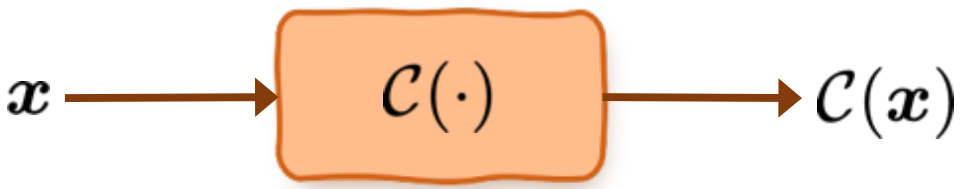
$m$ = **number of local samples stored in each client**

# Challenges

- **Communication efficiency:** limited bandwidth
- **Privacy:** sensitive information

# Communication efficiency

**Communication compression:** we compress the message into fewer bits, e.g. sparsification or quantization.

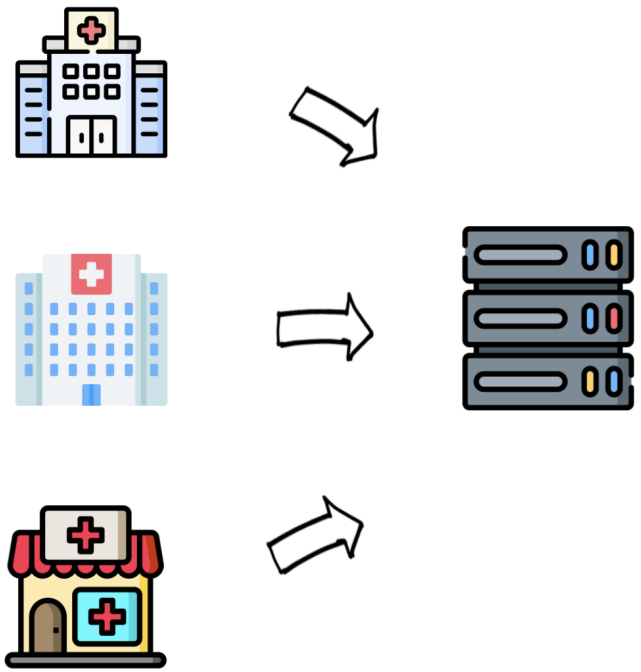



**Definition ($\omega$-compression operator)**

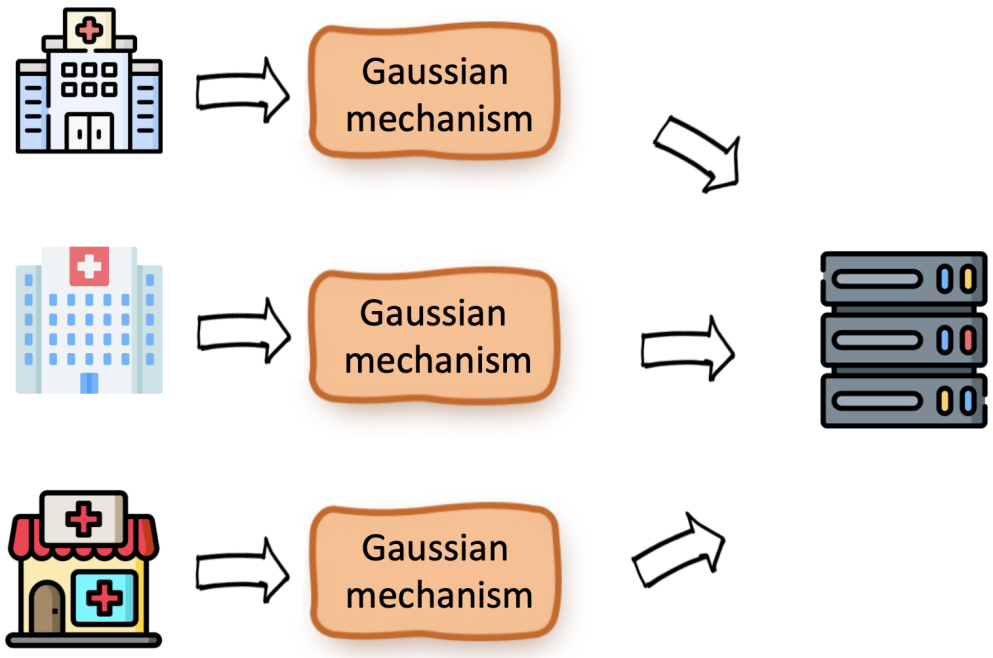
$$\mathbb{E}[\mathcal{C}(x)] = x, \qquad \mathbb{E}[\|\mathcal{C}(x) - x\|^2] \leq \omega \|x\|^2. \tag{1}$$

- **Random-$k$ sparsification** satisfies (1) with $\omega = \frac{d}{k} - 1$.
- **No compression ($k = d$)** $\Longrightarrow \omega = 0$.

# Privacy



# Privacy




**Local Differential Privacy (LDP):** we use **Gaussian mechanism** to guarantee the client privacy.

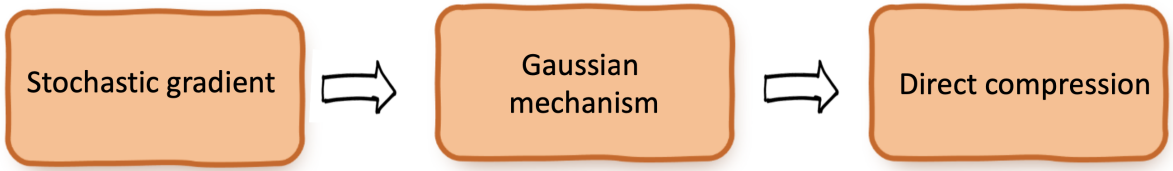
# Warm-up: direct compression + privacy (CDP-SGD)


**Algorithm 1** Compressed Differentially-Private Stochastic Gradient Descent (CDP-SGD)

**Input:** initial point $x^0$, stepsize $\eta_t$, variance $\sigma_p^2$, minibatch size $b$

1: **for** $t = 0, 1, 2, \ldots, T$ **do**
2: **for each client** $i \in [n]$ **do in parallel**
3: Compute local stochastic gradient $\tilde{\boldsymbol{g}}_i^t = \frac{1}{b}\sum_{j\in\mathcal{I}_b} \nabla f_{i,j}(\boldsymbol{x}^t)$ // client uses SGD
4: *Privacy*: $\boldsymbol{g}_i^t = \tilde{\boldsymbol{g}}_i^t + \boldsymbol{\xi}_i^t$, where $\boldsymbol{\xi}_t^i \sim \mathcal{N}(\boldsymbol{0}, \sigma_p^2 \boldsymbol{I})$ // Gaussian mechanism
5: *Compression*: let $\boldsymbol{v}_i^t = \mathcal{C}_i^t(\boldsymbol{g}_i^t)$ and send to the server // direct compression
6: **end each client**
7: Server aggregates compressed information $\boldsymbol{v}^t = \frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_i^t$
8: $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_t \boldsymbol{v}^t$
9: **end for**

# Warm-up: direct compression + privacy (CDP-SGD)




**Algorithm 1** Compressed Differentially-Private Stochastic Gradient Descent (CDP-SGD)

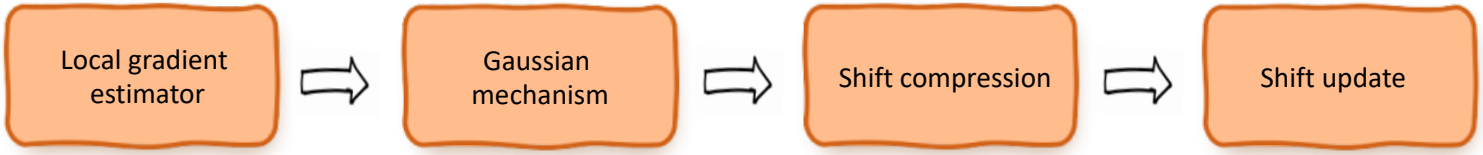
**Input:** initial point $x^0$, stepsize $\eta_t$, variance $\sigma_p^2$, minibatch size $b$

1: **for** $t = 0, 1, 2, \ldots, T$ **do**
2: **for each client** $i \in [n]$ **do in parallel**
3: Compute local stochastic gradient $\tilde{\boldsymbol{g}}_i^t = \frac{1}{b}\sum_{j\in\mathcal{I}_b} \nabla f_{i,j}(\boldsymbol{x}^t)$ // client uses SGD
4: *Privacy*: $\boldsymbol{g}_i^t = \tilde{\boldsymbol{g}}_i^t + \boldsymbol{\xi}_i^t$, where $\boldsymbol{\xi}_t^i \sim \mathcal{N}(\boldsymbol{0}, \sigma_p^2 \boldsymbol{I})$ // Gaussian mechanism
5: *Compression*: let $\boldsymbol{v}_i^t = \mathcal{C}_i^t(\boldsymbol{g}_i^t)$ and send to the server // direct compression
6: **end each client**
7: Server aggregates compressed information $\boldsymbol{v}^t = \frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_i^t$
8: $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_t \boldsymbol{v}^t$
9: **end for**

**Theorem 1 (L-Zhao-Li-Chi, NeurIPS'22).** CDP-SGD satisfies $(\epsilon, \delta)$-LDP with utility $\mathbb{E}\|\nabla f(\boldsymbol{x}^{\text{output}})\|^2 \le O\big(\frac{1}{m}\sqrt{\frac{(1+\omega)d\log(1/\delta)}{n\epsilon^2}}\big)$.

- local dataset size $m$ large $\Rightarrow$ communication $O\big(m^2 \frac{n\epsilon^2}{(1+\omega)\log(1/\delta)}\big)$
- smaller $\epsilon$ (stronger privacy) $\Rightarrow$ worse utility, fewer communication

# SoteriaFL: a unified framework for compressed private FL





**Algorithm 2** SoteriaFL (a unified framework for compressed private FL)

**Input:** initial point $\boldsymbol{x}^0$, stepsize $\eta_t$, shift stepsize $\gamma_t$, variance $\sigma_p^2$, initial reference $\boldsymbol{s}_i^0 = 0$

**for** $t = 0, 1, 2, \ldots, T$ **do**
  **for each client** $i \in [n]$ **do in parallel**
    Compute local gradient estimator $\tilde{\boldsymbol{g}}_i^t$ // it allows many methods, e.g., SGD, SVRG, and SAGA
    *Privacy*: $\boldsymbol{g}_i^t = \tilde{\boldsymbol{g}}_i^t + \boldsymbol{\xi}_i^t$, where $\boldsymbol{\xi}_i^t \sim \mathcal{N}(\boldsymbol{0}, \sigma_p^2 \boldsymbol{I})$ // Gaussian mechanism
    *Compression*: let $\boldsymbol{v}_i^t = \mathcal{C}_i^t(\boldsymbol{g}_i^t - \boldsymbol{s}_i^t)$ and send to the server // shift compression
    Update shift $\boldsymbol{s}_i^{t+1} = \boldsymbol{s}_i^t + \gamma_t \mathcal{C}_i^t(\boldsymbol{g}_i^t - \boldsymbol{s}_i^t)$ // shift update
  **end each client**
  Server aggregates compressed information $\boldsymbol{v}^t = \boldsymbol{s}^t + \frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_i^t$
  $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_t \boldsymbol{v}^t$
  $\boldsymbol{s}^{t+1} = \boldsymbol{s}^t + \gamma_t \frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_i^t$
**end for**

**Theorem 2 (L-Zhao-Li-Chi, NeurIPS'22).** SoteriaFL satisfies $(\epsilon, \delta)$-LDP with the **same utility** as CDP-SGD, while reducing the communication cost $O(m^2)$ to $O(m)$.

- flexible local gradient estimators (SoteriaFL-SGD/SVRG/SAGA)
- state-of-the-art shift compression
- better privacy-utility-communication trade-offs

# Thanks!

Zhize Li