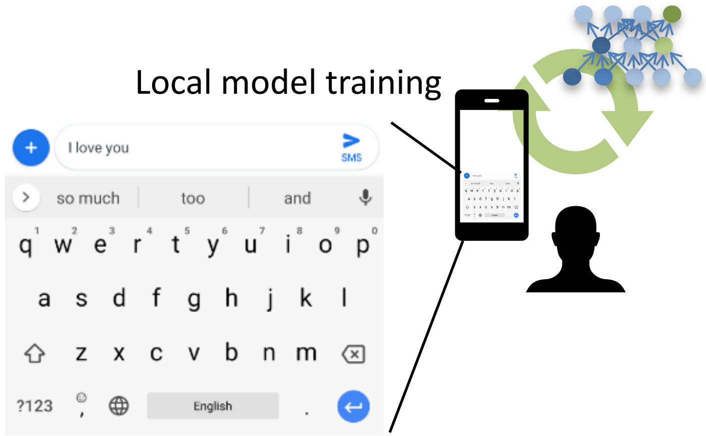
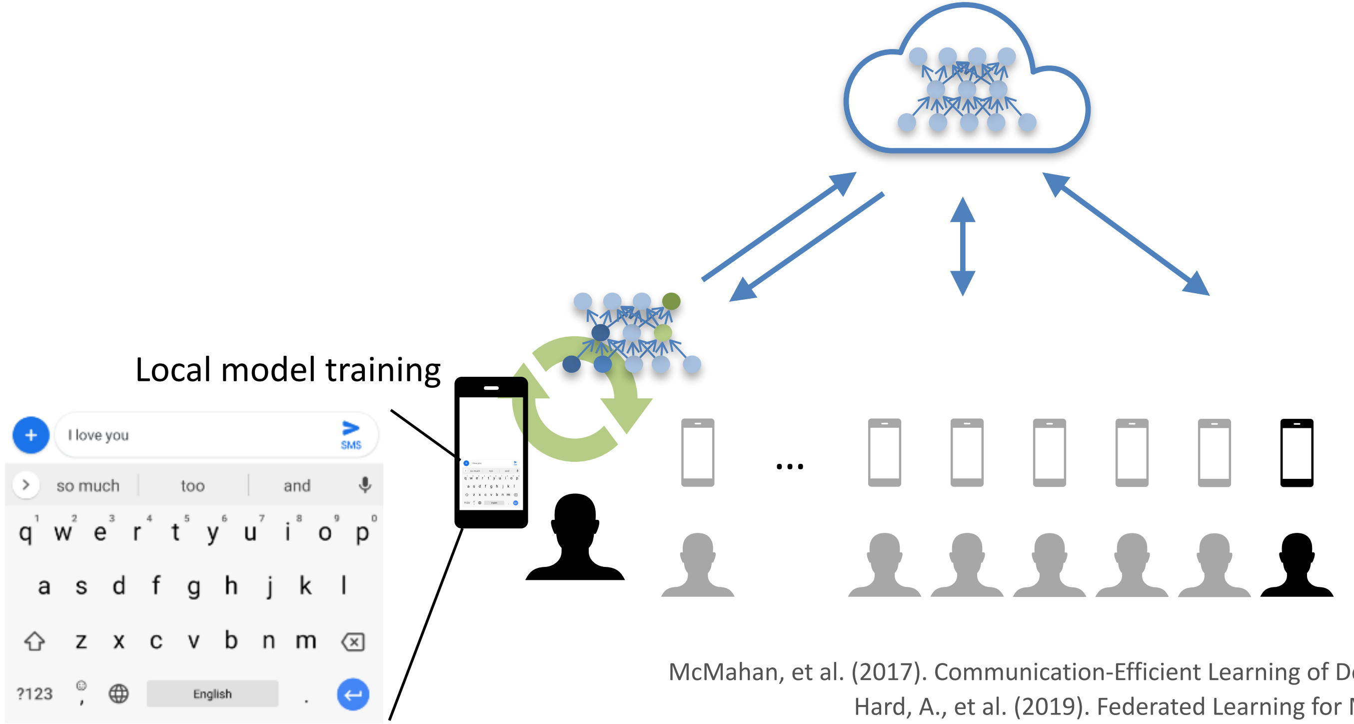
Dynamic Policies on Differentially Private Learning

Junyuan Hong
Michigan State University

Machine Learning in Our Life

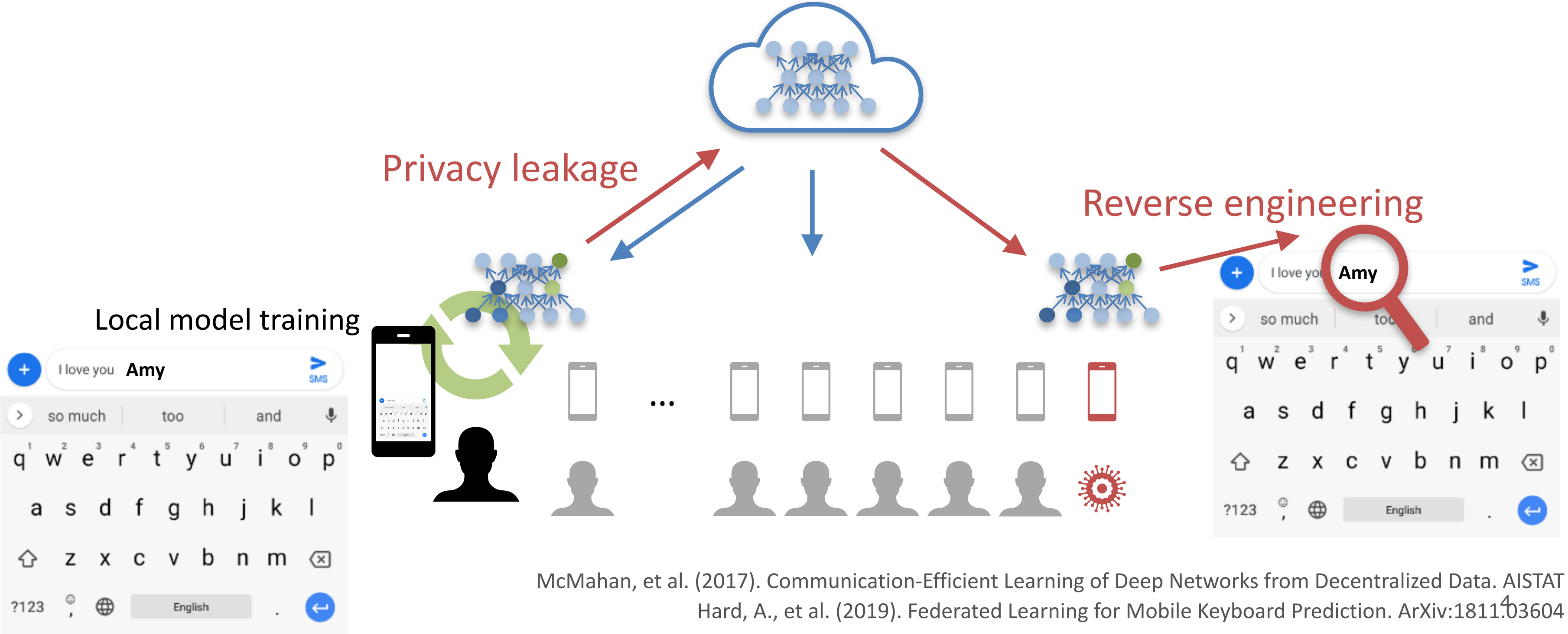


Federated Learning



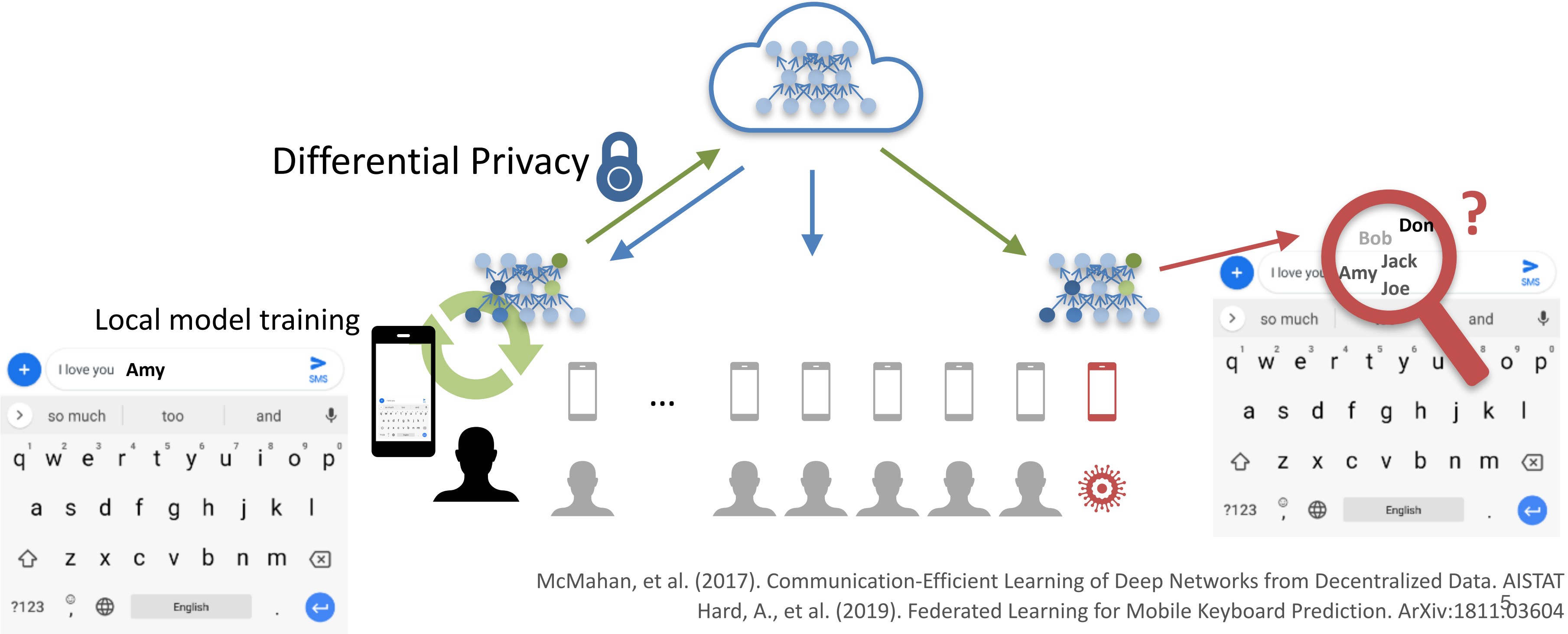
McMahan, et al. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTAT
Hard, A., et al. (2019). Federated Learning for Mobile Keyboard Prediction. ArXiv:1811.03604

Federated Learning



McMahan, et al. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTAT
Hard, A., et al. (2019). Federated Learning for Mobile Keyboard Prediction. ArXiv:1811.03604

Federated Learning

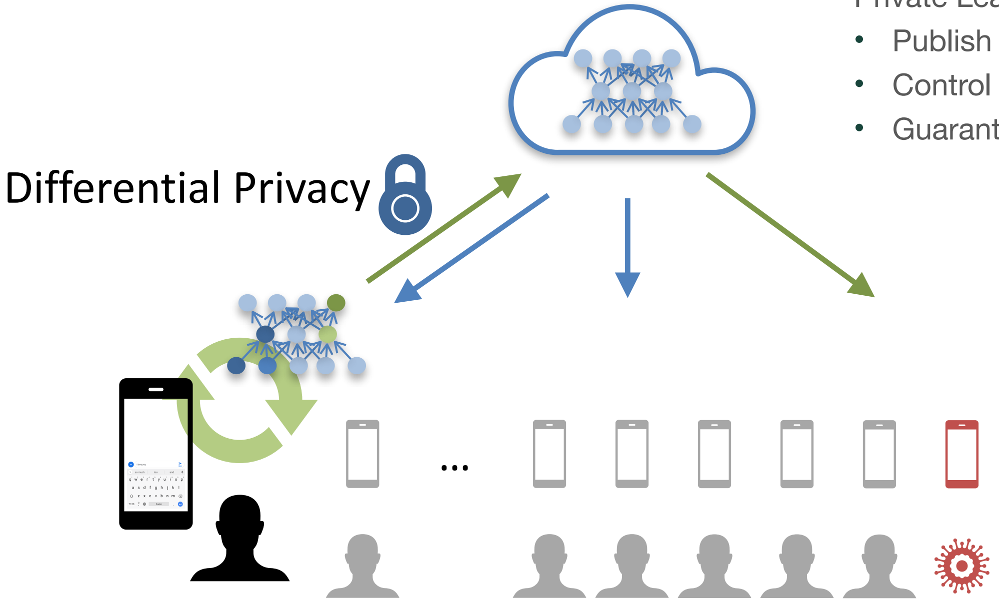


McMahan, et al. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTAT
Hard, A., et al. (2019). Federated Learning for Mobile Keyboard Prediction. ArXiv:1811.03604

Federated Learning

Private Learning:

- Publish knowledge (model) rather than data
- Control the privacy loss
- Guarantee the convergence



Private Learning

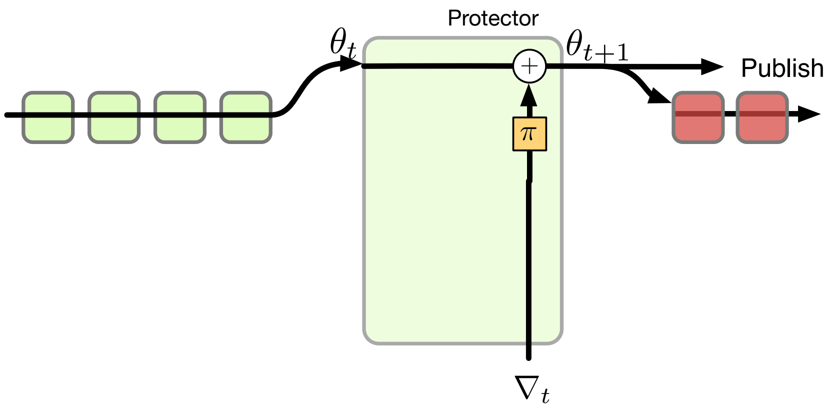
Algorithm

Convergence theory and dynamic policy

Learning by Gradient Descent

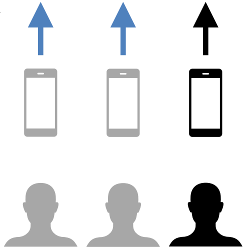
ρ privacy measure

π projection: AdaGrad, etc



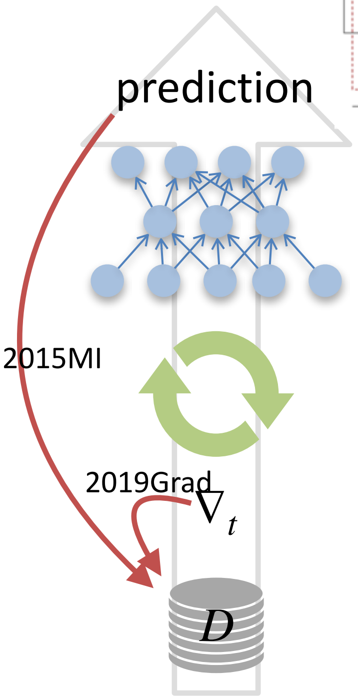
$$\nabla_t = \frac{1}{N}\sum_{n=1}^{N} \nabla f(\theta; x_n)$$

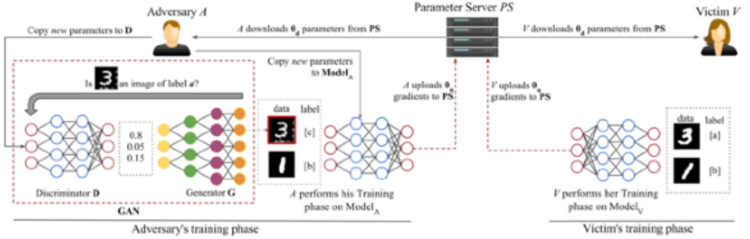
Private sample (to protect)



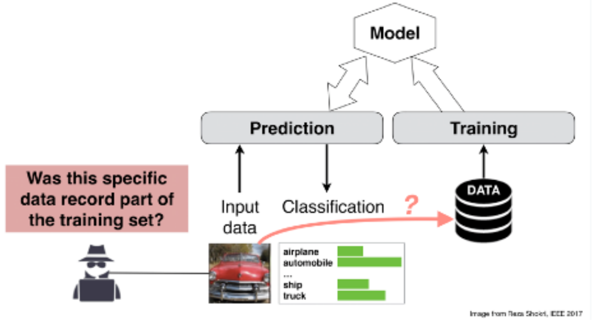
Privacy attack

- **2019Grad**: Deep Leakage from Gradients, Zhu et al.: $x = \arg\min_{x} ||\nabla f(x) - \nabla_t||^2$
- **2017MIA**: Membership Inference Attacks, Shokri et al.: $P(x \in D_{\text{train}}) = h(f(x;\theta))$ where $h()$ is a trained attack.
- **2017GAN**: Info Leakage from Collaborative Deep Learning, Hitaj et al. 2017: $x = G(z)$ where $z = \max_{z} f(G(z);\theta)$
- **2015MI**: Model Inversion, Fredrikson et al.: $x = \arg\max_{x} f(x)$ (statistical model)





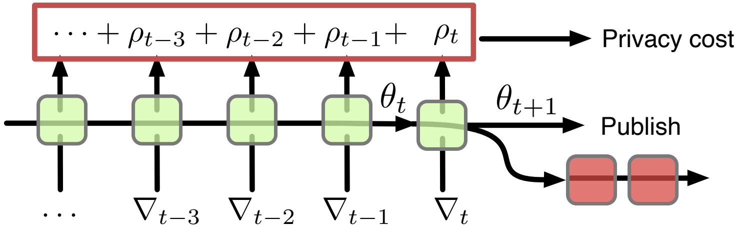
2017GAN



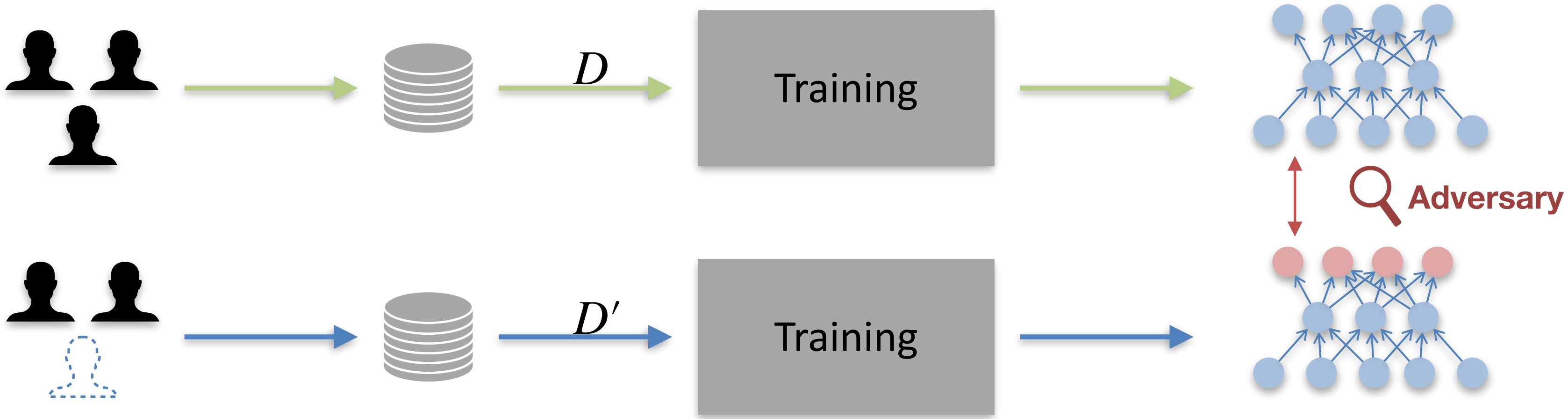
2017MIA

Quantify privacy

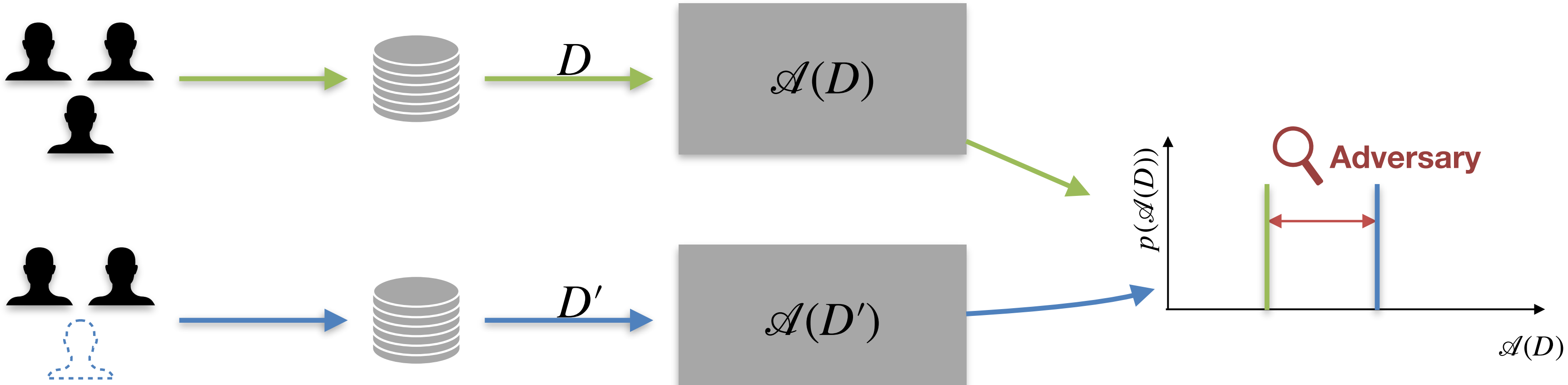
If privacy cost is over a budget, we stop and publish model



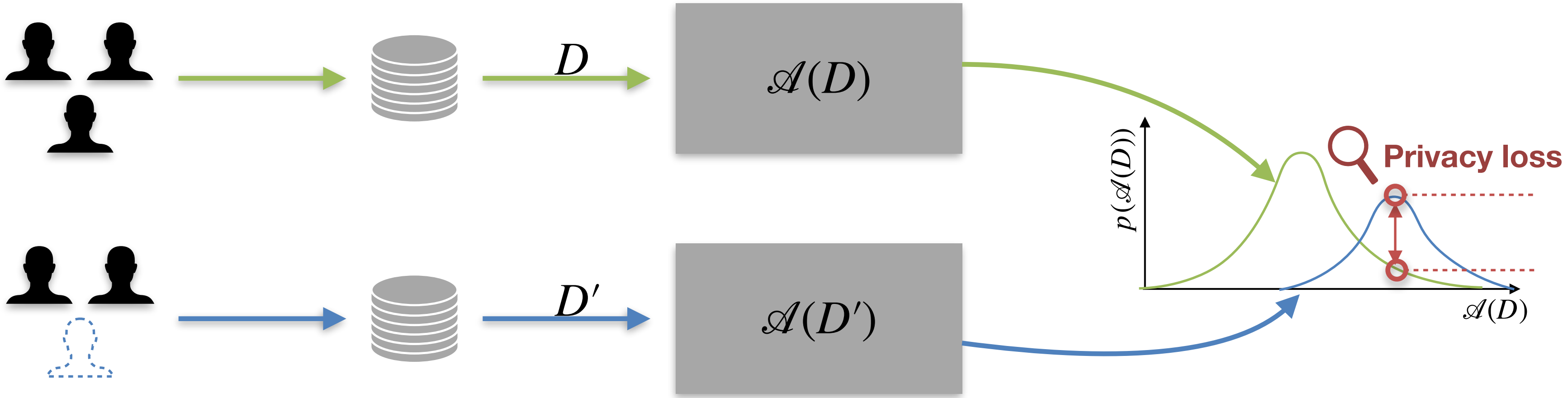
Quantify privacy: Differential Privacy (DP)



Differential Privacy



Differential Privacy



Privacy loss at y $Z(y) \triangleq \log \left(\frac{p(\mathcal{A}(D) = y)}{p(\mathcal{A}(D') = y)} \right)$ **where $y \sim \mathcal{A}(D)$ and D, D' are adjacent (differing at one sample)**

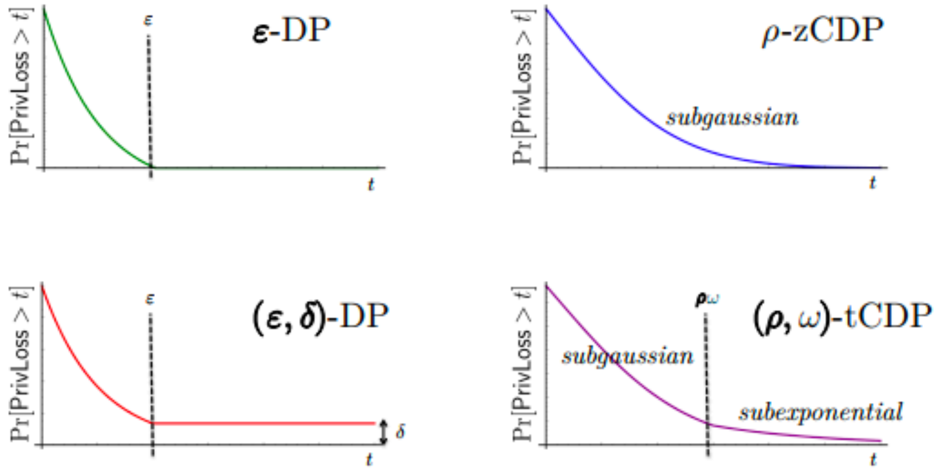
Differential Privacy

Privacy loss at y

$$Z(y) \triangleq \log\left(\frac{p(\mathscr{A}(D)=y)}{p(\mathscr{A}(D')=y)}\right)$$

where $y \sim \mathscr{A}(D)$

$P(Z > t)$



$\mathscr{A}$ is ϵ-DP: $Z \leq \epsilon$ or $P(Z > \epsilon) = 0$

$\mathscr{A}$ is (ϵ, δ)-DP: $P(Z > \epsilon) = \delta$

$\mathscr{A}$ is ρ-zCDP: $P(Z > t + \rho) \leq e^{-t^2/(4\rho)}$ for $t \geq 0$

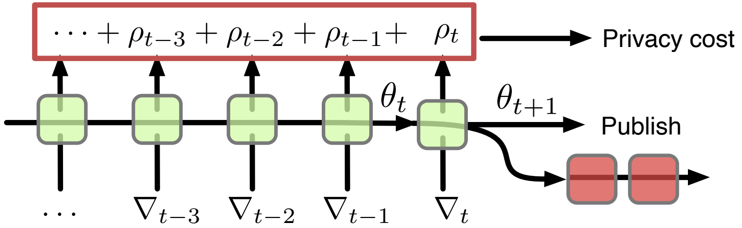
$\mathscr{A}$ is (ρ, ω)-tCDP: $P(Z > t + \rho) \leq e^{-t^2/(4\rho)}$ for $t \in [0, 2\rho(\omega - 1)]$

$P(Z > t + \rho) \leq e^{(\omega-1)^2\rho} \cdot e^{-(\omega-1))t}$ for $t \in (2\rho(\omega - 1), \infty)$

Bun, M., Dwork, C., et al. (2018). Composable and Versatile Privacy via Truncated CDP. *STOC*

Quantify privacy: Accumulate privacy loss

Compose **dynamic** privacy parameter

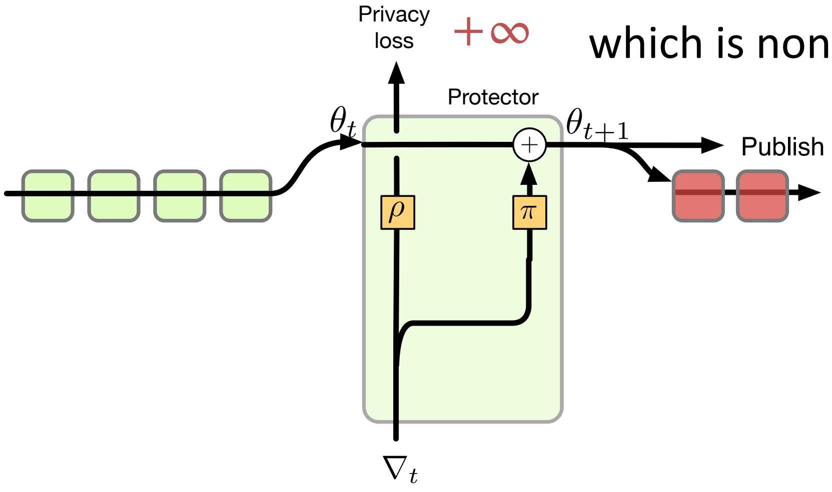


Lemma 3.5. *(Composition) Suppose two mechanisms $\mathcal{M}, \mathcal{M}' : \mathcal{D}^n \to \mathbb{R}^d$ satisfy ρ_1-zCDP and ρ_2-zCDP, then their composition satisfies $(\rho_1 + \rho_2)$-zCDP.*

Note: zCDP allows ρ_1 and ρ_2 to be different, but DP does not. For DP, an additional privacy cost has to be paid.

Bun, M., & Steinke, T. (2016). Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds, TOC
Dwork, C., & Rothblum, G. N. (2016). Concentrated Differential Privacy. ArXiv:1603.01887
Rogers, et al. (2016). Privacy Odometers and Filters: Pay-as-you-Go Composition. NeurIPS

Quantify privacy



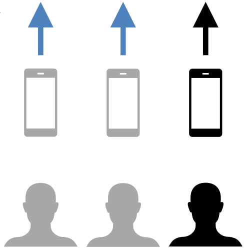
π is **deterministic**
which is non-private

ρ privacy measure

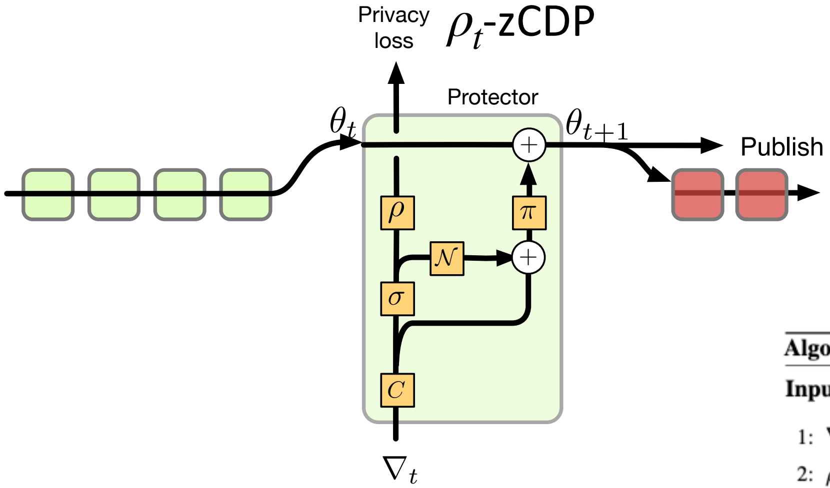
π projection: AdaGrad, etc

$$\nabla_t = \frac{1}{N}\sum_{n=1}^{N} \nabla f(\theta; x_n)$$

Private sample (to protect)



Privatize Gradients



- ρ privacy measure
- π projection: AdaGrad, etc
- σ **noise schedule**
- $\mathcal{N}$ **noise distribution**
- C **sensitivity constraint**

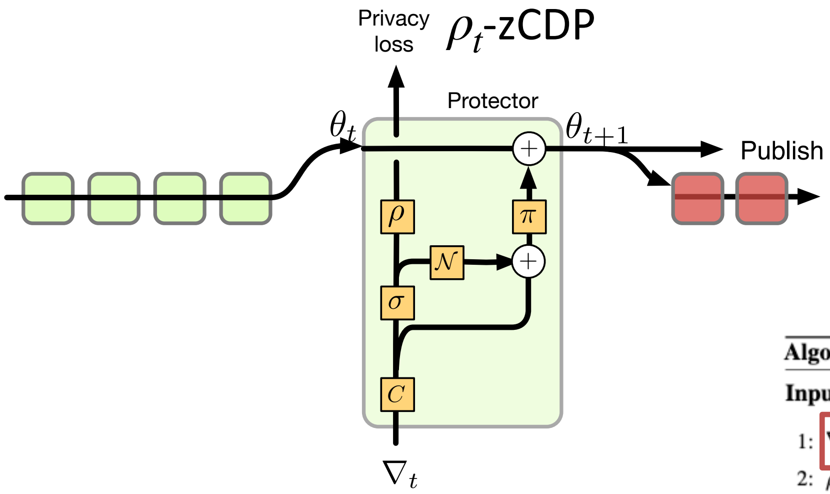
Algorithm 1 Privatizing gradients

Input: Private gradient ∇_t summed from $[\nabla_t^{(1)}, \dots, \nabla_t^{(n)}]$, residual privacy budget R_t

$\tilde{\nabla}_t \leftarrow \frac{1}{N}\sum_{n=1}^{N} \nabla_t^{(n)} \min\{1, C_t / \left\|\nabla_t^{(n)}\right\|\}$ ▷ Sensitivity constraint
$\rho_t \leftarrow 1/\sigma_t^2$ ▷ Budget request
if $\rho_t < R_t$ **then**
    $R_{t+1} \leftarrow R_t - \rho_t$
    $g_t \leftarrow \nabla_t + C_t \sigma_t \nu_t / N, \nu_t \sim \mathcal{N}(0, I)$ ▷ Privacy noise
    return $\eta_t g_t$, R_{t+1} ▷ Utility projection
else
    Terminate

Cost some privacy budget

Privatize Gradients



- ρ privacy measure
- π projection: AdaGrad, etc
- σ noise schedule
- $\mathcal{N}$ noise distribution
- C **sensitivity constraint**

Algorithm 1 Privatizing gradients

Input: Private gradient ∇_t summed from $[\nabla_t^{(1)}, \dots, \nabla_t^{(n)}]$, residual privacy budget R_t

1: $\tilde{\nabla}_t \leftarrow \frac{1}{N} \sum_{n=1}^{N} \nabla_t^{(n)} \min\{1, C_t / \left\| \nabla_t^{(n)} \right\|\}$ ▷ Sensitivity constraint
2: $\rho_t \leftarrow 1/\sigma_t^2$ ▷ Budget request
3: **if** $\rho_t < R_t$ **then**
4: $R_{t+1} \leftarrow R_t - \rho_t$
5: $g_t \leftarrow \tilde{\nabla}_t + C_t \sigma_t \nu_t / N, \nu_t \sim \mathcal{N}(0, I)$ ▷ Privacy noise
6: **return** $\eta_t g_t, R_{t+1}$ ▷ Utility projection
7: **else**
8: Terminate

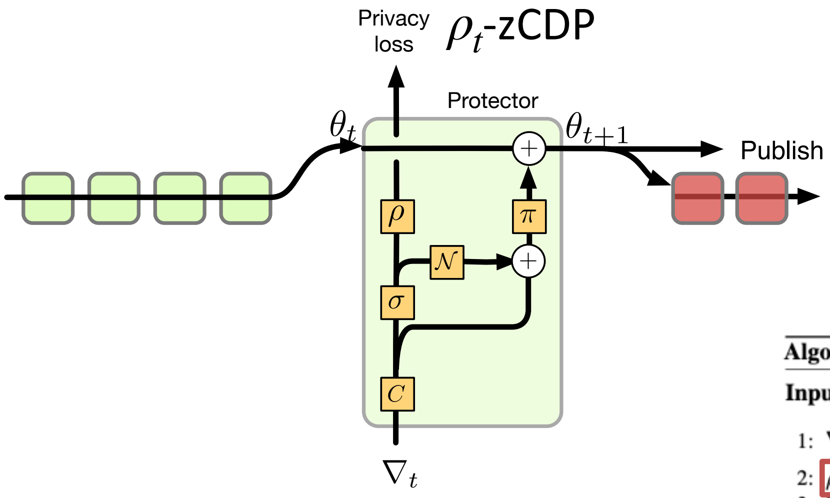
Control the influence of a sample

LEMMA 3.1 (L_2 SENSITIVITY). *Given mapping from a n-element dataset domain to d-dimensional real space $f : \mathcal{D}^n \to \mathbb{R}^d$, the L_2 sensitivity of f, denoted by $\Delta_2(f)$ is defined as:*

$$\Delta_2(f) = \max_{D, D'} \left\| f(D) - f(D') \right\|_2 ,$$

where D, D' are adjacent datasets.

Differentially Private Learning



- ρ privacy measure
- π projection: AdaGrad, etc
- σ **noise schedule**
- $\mathcal{N}$ **noise distribution**
- C sensitivity control

LEMMA 3.4. *The Gaussian mechanism, which returns $f(D) + \sigma v$ satisfies $\Delta_2(f)^2/(2\sigma^2)$-zCDP.*

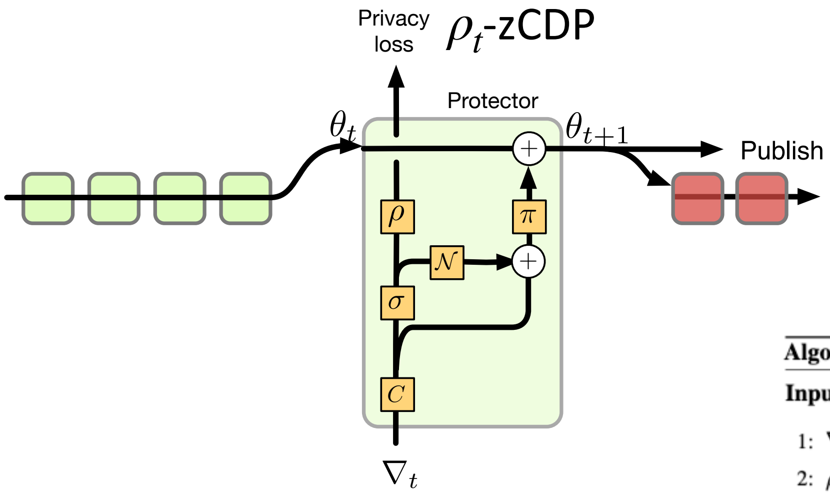
A deterministic function

Algorithm 1 Privatizing gradients

Input: Private gradient ∇_t summed from $[\nabla_t^{(1)}, \ldots, \nabla_t^{(n)}]$, residual privacy budget R_t

1: $\tilde{\nabla}_t \leftarrow \frac{1}{N} \sum_{n=1}^{N} \nabla_t^{(n)} \min\{1, C_t / \left\|\nabla_t^{(n)}\right\|\}$ ▷ Sensitivity constraint
2: $\rho_t \leftarrow 1/\sigma_t^2$ ▷ Budget request
3: **if** $\rho_t < R_t$ **then**
4: $\quad R_{t+1} \leftarrow R_t - \rho_t$
5: $\quad g_t \leftarrow \tilde{\nabla}_t + C_t \sigma_t \nu_t / N, \nu_t \sim \mathcal{N}(0, I)$ ▷ Privacy noise
6: $\quad$ **return** $\eta_t g_t, R_{t+1}$ ▷ Utility projection
7: **else**
8: $\quad$ Terminate

Differentially Private Learning



- ρ privacy measure
- π projection: AdaGrad, etc
- σ noise schedule
- $\mathcal{N}$ noise distribution
- C sensitivity control

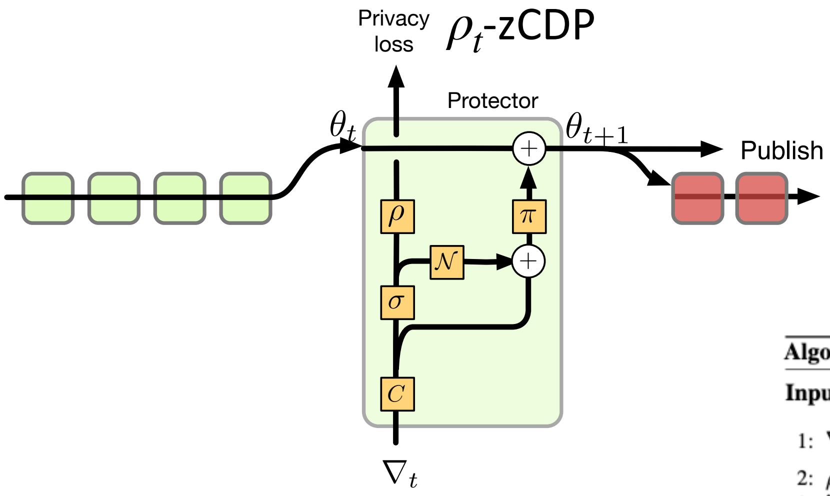
Algorithm 1 Privatizing gradients

Input: Private gradient ∇_t summed from $[\nabla_t^{(1)}, \dots, \nabla_t^{(n)}]$ residual privacy budget R_t

$\tilde{\nabla}_t \leftarrow \frac{1}{N} \sum_{n=1}^{N} \nabla_t^{(n)} \min\{1, C_t / \left\| \nabla_t^{(n)} \right\| \}$ ▷ Sensitivity constraint
$\rho_t \leftarrow 1/\sigma_t^2$ ▷ Budget request
if $\rho_t < R_t$ **then**
    $R_{t+1} \leftarrow R_t - \rho_t$
    $g_t \leftarrow \tilde{\nabla}_t + C_t \sigma_t \nu_t / N, \nu_t \sim \mathcal{N}(0, I)$ ▷ Privacy noise
    return $\eta_t g_t, R_{t+1}$ ▷ Utility projection
else
    Terminate

If gradients are a stochastic mini-batch, e.g., sampled by q-probability, the privacy cost is $\propto q^2 \rho$ for DP metric, e.g, tCDP.

Privatize Gradients



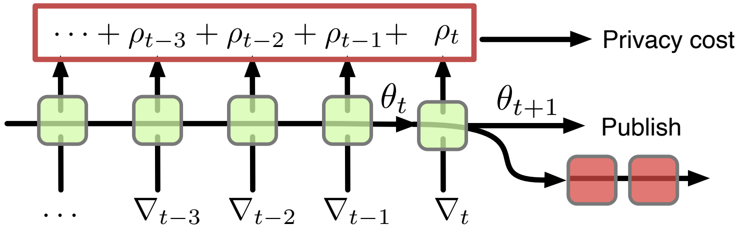
- ρ privacy measure
- π projection: AdaGrad, etc
- σ **noise schedule**
- $\mathcal{N}$ **noise distribution**
- C **sensitivity constraint**

Algorithm 1 Privatizing gradients

Input: Private gradient ∇_t summed from $[\nabla_t^{(1)}, \dots, \nabla_t^{(n)}]$, residual privacy budget R_t

1: $\tilde{\nabla}_t \leftarrow \frac{1}{N}\sum_{n=1}^{N} \nabla_t^{(n)} \min\{1, C_t / \left\|\nabla_t^{(n)}\right\|\}$ ▷ Sensitivity constraint
2: $\rho_t \leftarrow 1/\sigma_t^2$ ▷ Budget request
3: **if** $\rho_t < R_t$ **then**
4: $R_{t+1} \leftarrow R_t - \rho_t$
5: $g_t \leftarrow \tilde{\nabla}_t + C_t \sigma_t \nu_t / N, \nu_t \sim \mathcal{N}(0, I)$ ▷ Privacy noise
6: **return** $\eta_t g_t, R_{t+1}$ ▷ Utility projection
7: **else**
8: Terminate

Differentially Private Learning



Algorithm 1 Privatizing gradients

Input: Private gradient ∇_t summed from $[\nabla_t^{(1)}, \dots, \nabla_t^{(n)}]$, residual privacy budget R_t

$\tilde{\nabla}_t \leftarrow \frac{1}{N}\sum_{n=1}^{N} \nabla_t^{(n)} \min\{1, C_t / \left\|\nabla_t^{(n)}\right\|\}$ ▷ Sensitivity constraint
$\rho_t \leftarrow 1/\sigma_t^2$ ▷ Budget request
if $\rho_t < R_t$ **then**
    $R_{t+1} \leftarrow R_t - \rho_t$
    $g_t \leftarrow \tilde{\nabla}_t + C_t \sigma_t \nu_t / N, \nu_t \sim \mathcal{N}(0, I)$ ▷ Privacy noise
    return $\eta_t g_t, R_{t+1}$ ▷ Utility projection
else
    Terminate

Private Learning

Algorithm

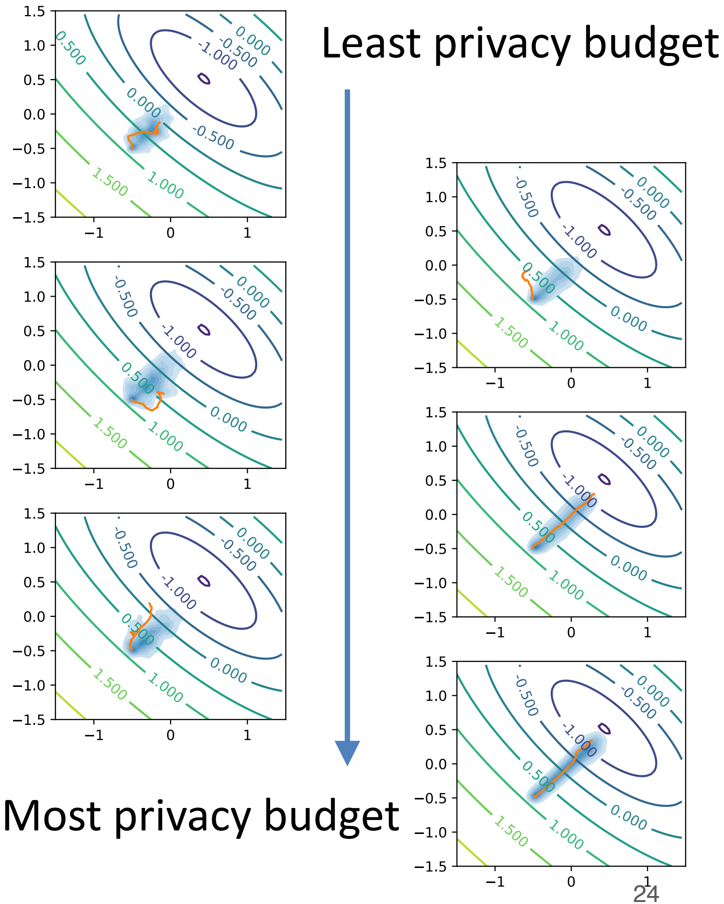
Convergence theory and dynamic policy

Does private learning converge?

- Not converge to the optimal
  - Finite iteration
  - Noise
- Improve the final iterate loss given a privacy budget:

$$\mathrm{EER} = \mathbb{E}_{\nu}[f(\theta_{T+1})] - f(\theta^*)$$

  - The upper bound of EER



Why study convergence upper bound?

- Bound the worst case.
- Find a way to speed up optimization algorithm
- To study the impact of privacy operations, e.g., noise magnitude, clipping norm, etc.
- To compare different algorithms: convergence rate

Assumptions

- G-Lipschitz continuous loss,
  $\| f(x) - f(x') \| \leq G\|x - x'\| \Leftrightarrow \|f'(x)\| \leq G$ if f is differentiable.
- M-Lipschitz continuous gradient or M-smooth loss:
  $\| \nabla f(x) - \nabla f(x') \| \leq M\|x - x'\|$
- μ-Polyak-Lojasiewicz (PL) condition < μ-strongly convex
  $\| \nabla f(\theta) \|^2 \geq 2\mu(f(\theta) - f(\theta^*))$

Convergence

Algorithm 1 Privatizing gradients

Input: Private gradient ∇_t summed from $[\nabla_t^{(1)}, \dots, \nabla_t^{(n)}]$, residual privacy budget R_t

1: $\tilde{\nabla}_t \leftarrow \frac{1}{N}\sum_{n=1}^{N} \nabla_t^{(n)} \min\{1, C_t / \left\|\nabla_t^{(n)}\right\|\}$ ▷ Sensitivity constraint
2: $\rho_t \leftarrow 1/\sigma_t^2$ ▷ Budget request
3: **if** $\rho_t < R_t$ **then**
4: $R_{t+1} \leftarrow R_t - \rho_t$
5: $g_t \leftarrow \tilde{\nabla}_t + C_t \sigma_t \nu_t / N, \nu_t \sim \mathcal{N}(0, I)$ ▷ Privacy noise
6: **return** $\eta_t g_t, R_{t+1}$ ▷ Utility projection
7: **else**
8: Terminate

Theorem 3.2. *Let α, κ and γ be defined in Eq. (5), and $\eta_t = \frac{1}{M}$. Suppose $f(\theta; x_i)$ is G-Lipschitz M-smooth and satisfies the Polyak-Lojasiewicz condition. If $C_t \le G$, then clipping does not take place, i.e., $\tilde{\nabla}_t = \nabla_t$ and the following holds:*

$$\text{EER} = \mathbb{E}_\nu[f(\theta_{T+1})] - f(\theta^*) \le \left(\gamma^T + R\sum_{t=1}^{T} q_t \sigma_t^2\right)(f(\theta_1) - f(\theta^*)), \tag{6}$$

$$\text{where } q_t \triangleq \gamma^{T-t}\alpha_t. \tag{7}$$

$$\alpha_t \triangleq \frac{MD}{2R}\left(\frac{\eta_t C_t}{N}\right)^2 \frac{1}{f(\theta_1) - f(\theta^*)} > 0,\ \kappa \triangleq \frac{M}{\mu} \ge 1,\ \text{and}\ \ \gamma \triangleq 1 - \frac{1}{\kappa} \in [0, 1). \tag{5}$$

Proof partially based on (Wang, et al., NeurIPS 2017)

Convergence

Theorem 3.2. *Let α, κ and γ be defined in Eq. (5), and $\eta_t = \frac{1}{M}$. Suppose $f(\theta; x_i)$ is G-Lipschitz M-smooth and satisfies the Polyak-Lojasiewicz condition. If $C_t \leq G$, then clipping does not take place, i.e., $\tilde{\nabla}_t = \nabla_t$ and the following holds:*

$$\text{EER} \leq \left(\gamma^T + R \sum_{t=1}^{T} q_t \sigma_t^2 \right) (f(\theta_1) - f(\theta^*)), \tag{6}$$

$$\text{where } q_t \triangleq \gamma^{T-t} \alpha_t. \tag{7}$$

Finite iteration: γ^T

Noise impact: $\sum_{t=1}^{T} q_t \sigma_t^2$

- Schedule noise to
  - Extend iteration T
  - Reduce the effect of noise

Convergence

Theorem 3.2. *Let α, κ and γ be defined in Eq. (5), and $\eta_t = \frac{1}{M}$. Suppose $f(\theta; x_i)$ is G-Lipschitz M-smooth and satisfies the Polyak-Lojasiewicz condition. If $C_t \leq G$, then clipping does not take place, i.e., $\tilde{\nabla}_t = \nabla_t$ and the following holds:*

$$\mathrm{EER} \leq \left(\gamma^T + R\sum_{t=1}^{T} q_t \sigma_t^2\right)(f(\theta_1) - f(\theta^*)), \tag{6}$$

$$\text{where } q_t \triangleq \gamma^{T-t}\alpha_t. \tag{7}$$

Influence of noise

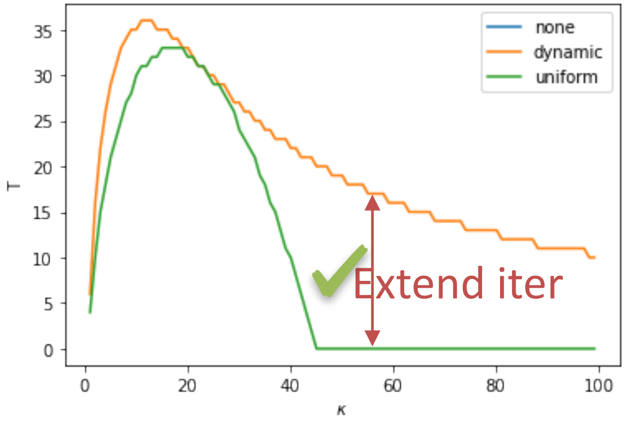
Lemma 3.1 (Dynamic schedule). *Suppose σ_t satisfy $\sum_{t=1}^{T} \sigma^{-2} = R$. Given a positive sequence $\{q_t\}$, the following equation holds*

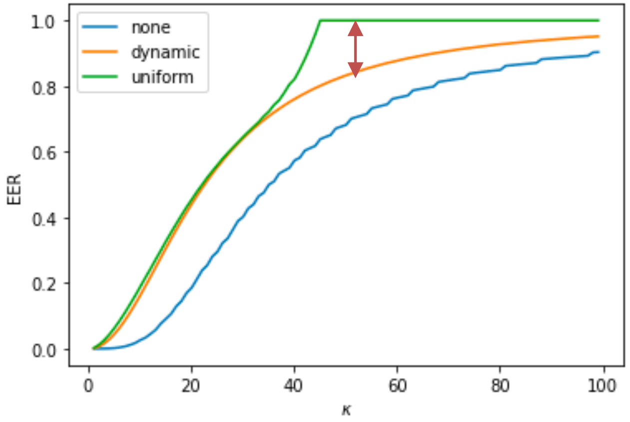
✓Reduce noise impact

$$\min_{\sigma} R\sum_{t=1}^{T} q_t \sigma_t^2 = \left(\sum_{t=1}^{T} \sqrt{q_t}\right)^2, \text{ when } \sigma_t = \sqrt{\frac{1}{R}\sum_{i=1}^{T}\sqrt{\frac{q_i}{q_t}}}. \tag{10}$$

How much improvement can we achieve?

Advantage of dynamic schedule on optimal upper bound





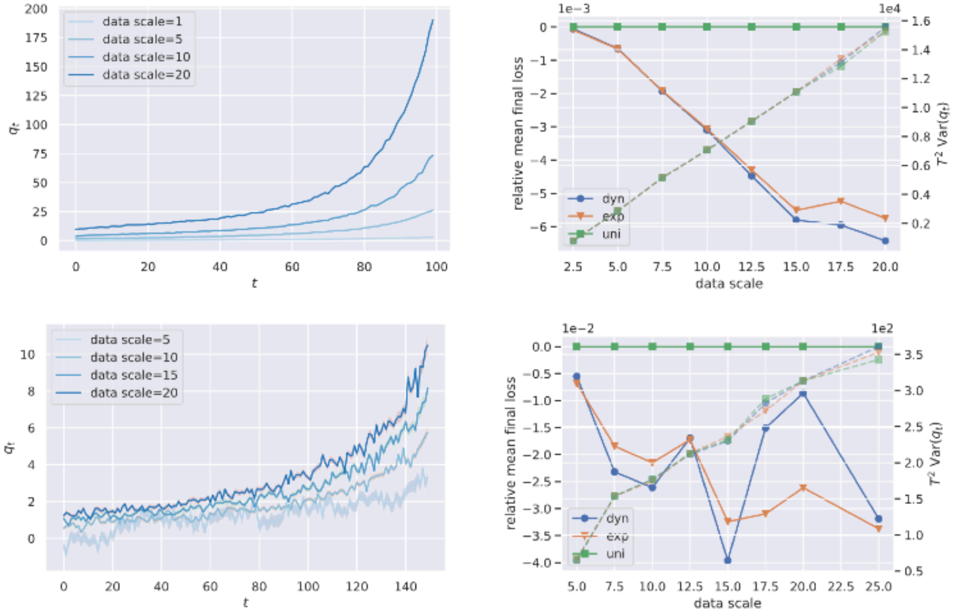
stable when the loss curvature is sharp

Advantage of dynamic schedule

- Empirically check the q_t

$$\mathrm{EER} \leq \left(\gamma^T + R\sum_{t=1}^{T} q_t\sigma_t^2\right)(f(\theta_1) - f(\theta^*)),$$

$$\textit{where } q_t \triangleq \gamma^{T-t}\alpha_t.$$



Further reduce the noise by momentum

Algorithm 2 Privatizing gradients with debiased momentum

Input: Private gradient ∇_t summed from $[\nabla_t^{(1)}, \ldots, \nabla_t^{(n)}]$, residual privacy budget R_t

$\tilde{\nabla}_t \leftarrow \frac{1}{N} \sum_{n=1}^{N} \nabla_t^{(n)} \min\{1, C_t / \left\|\nabla_t^{(n)}\right\|\}$ ▷ Sensitivity constraint
$\rho_t \leftarrow 1/\sigma_t^2$ ▷ Budget request
if $\rho_t < R_t$ **then**
    $R_{t+1} \leftarrow R_t - \rho_t$
    $g_t \leftarrow \tilde{\nabla}_t + \nu_t, \nu_t \sim \mathcal{N}(0, (C_t \sigma_t / N)^2 I)$ ▷ Privacy noise
    $v_{t+1} = \beta v_t + (1-\beta) g_t, \; v_1 = 0$
    $\hat{v}_{t+1} = v_{t+1} / (1 - \beta^t)$
    return $\eta_t \hat{v}_{t+1}, R_{t+1}$ ▷ Utility projection
else
    Terminate

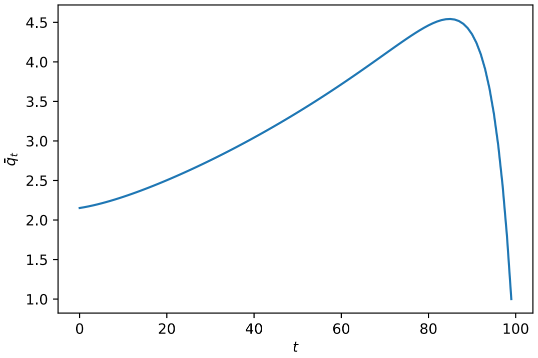
Further reduce the noise by momentum

Theorem 3.4 (Convergence under PL condition). *Suppose $f(\theta; x_i)$ is M-smooth, G-Lipschitz and satisfies the Polyak-Lojasiewicz condition. Let $\eta_t = \eta_0$. If $C_t \geq G$ which implies $\tilde{\nabla}_t = \nabla_t$ (clipping does not take place), then the following holds:*

$$\mathrm{EER} \leq \gamma^T (f(\theta_1) - f(\theta^*)) + \underbrace{\frac{2\eta_0 D}{N^2} \sum_{t=1}^{T} q_t (C_t \sigma_t)^2}_{\text{noise varinace}} + \underbrace{\eta_0 \zeta \sum_{t=1}^{T} \gamma^{T-t} \|v_{t+1}\|^2}_{\text{momentum effect}} \quad (16)$$

$$\text{where } q_t = \frac{\beta^{2(T-t+1)} - \gamma^{T-t+1}}{\beta^2 - \gamma}, \; \gamma = 1 - \eta_0 \mu, \; \zeta = \frac{4M^2 \beta \gamma}{(\gamma - \beta)^2 (1-\beta)^3} \eta_0^2 + \frac{1}{2} M \eta_0 - 1. \quad (17)$$

Especially, when $\eta_0 \leq \frac{\beta(1-\beta)^3}{8M}\left[\sqrt{\frac{1}{4} + \frac{16}{\beta(1-\beta)^3}} - 1\right]$, the noise variance dominates the bound, i.e., $\mathrm{EER} = \mathcal{O}\left(\frac{2\eta_0 D}{N^2} \sum_{t=1}^{T} q_t (C_t \sigma_t)^2\right)$.



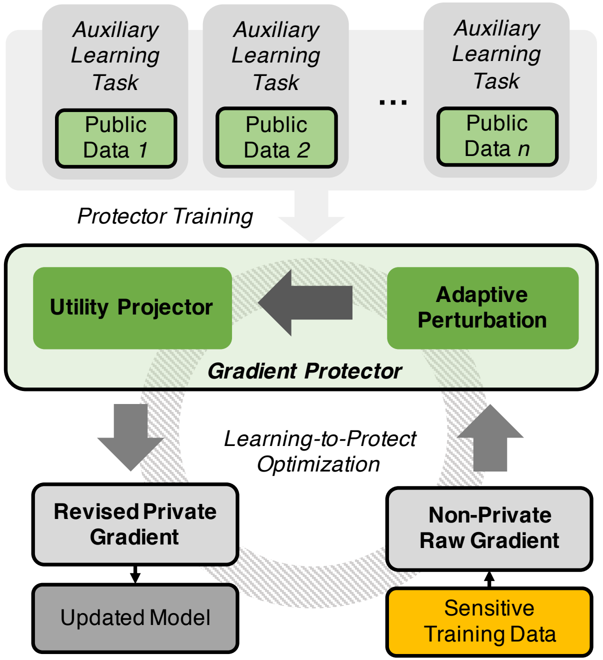
A negative term if η_0 is small.

The GD noise

Proof partially based on (Zhu, et al., ArXiv 2020)

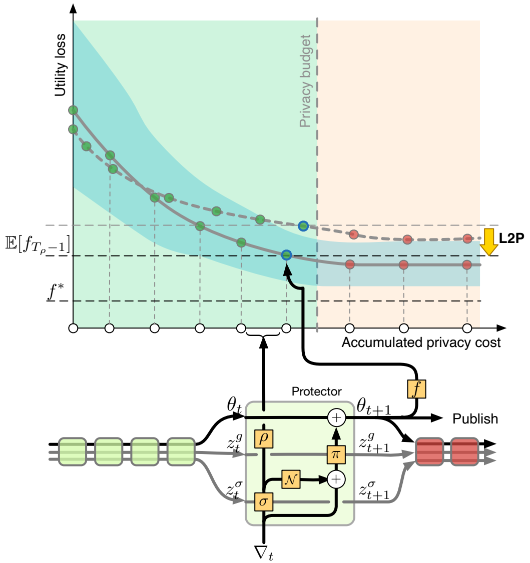
Beyond dynamic noise magnitude

- Learning to protect: Transfer the dynamic policies learned from auxiliary tasks to private task.
- AdaClip (Pichapati et al. 2019): Adaptively clipping the gradients
- Dynamic batch size (Feldman et al., 2019, STOC): Increase the batch size to improve non-convex convergence bound.



Beyond dynamic noise magnitude

- **Learning to protect**: Transfer the dynamic policies learned from auxiliary tasks to private task.
- AdaClip (Pichapati et al. 2019): Adaptively clipping the gradients
- Dynamic batch size (Feldman et al., 2019, STOC): Increase the batch size to improve non-convex convergence bound.



$$\min_{\pi,\sigma,T} \mathbb{E}\left[\tilde{F}(\sigma,\pi,T)\right], \text{ s.t. } h_T(\sigma;\rho_{\text{tot}}) = 0$$

Beyond dynamic noise magnitude

- **Learning to protect**: Transfer the dynamic policies learned from auxiliary tasks to private task.
- AdaClip (Pichapati et al. 2019): Adaptively clipping the gradients
- Dynamic batch size (Feldman et al., 2019, STOC): Increase the batch size to improve non-convex convergence bound.

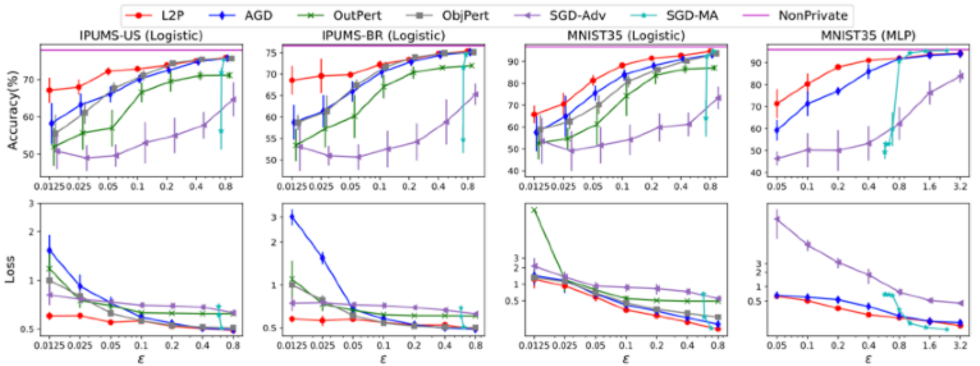


Figure 2: Test performance (top) and training loss values (bottom) by varying ϵ of logistic and MLP classifiers on IPUMS and MNIST35 datasets. The error bar presents the size of standard deviations. For better visualization, some horizontal offsets are added to every point.

$$\min_{\pi,\sigma,T} \mathbb{E}\left[\tilde{F}(\sigma,\pi,T)\right], \text{ s.t. } h_T(\sigma;\rho_{\text{tot}})=0$$

Thank you for your time!