Auditing Differentially Private Machine Learning

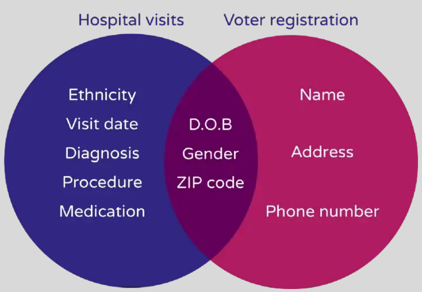
Jonathan Ullman

Khoury College of Computer Sciences
Northeastern University

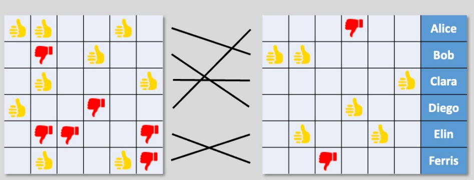
The Power of Negative Thinking

Privacy attacks play an essential role in privacy research

Reidentification



(Sweeney)



(NS08)

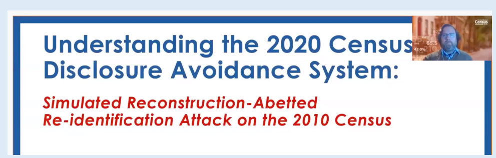
Highlighted the failures of deidentification

Reconstruction

TABLE 4: A SINGLE SATISFYING ASSIGNMENT

AGE	SEX	RACE	MARITAL STATUS		SOLUTION #1
8	F	B	S		8FBS
18	M	W	S		18MWS
24	F	W	S	=	24FWS
30	M	W	M		30MWM
36	F	B	M		36FBM
66	F	B	M		66FBM
84	M	B	M		84MBM

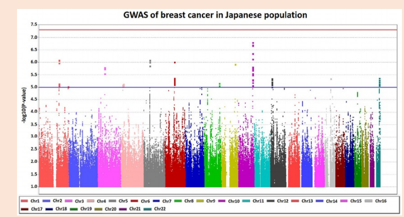
(DN03)



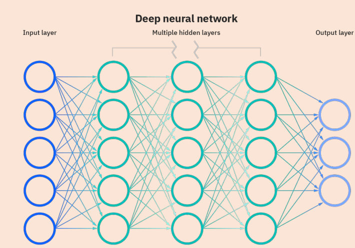
(US Census Bureau)

Inspired invention and adoption of differential privacy

Membership inference



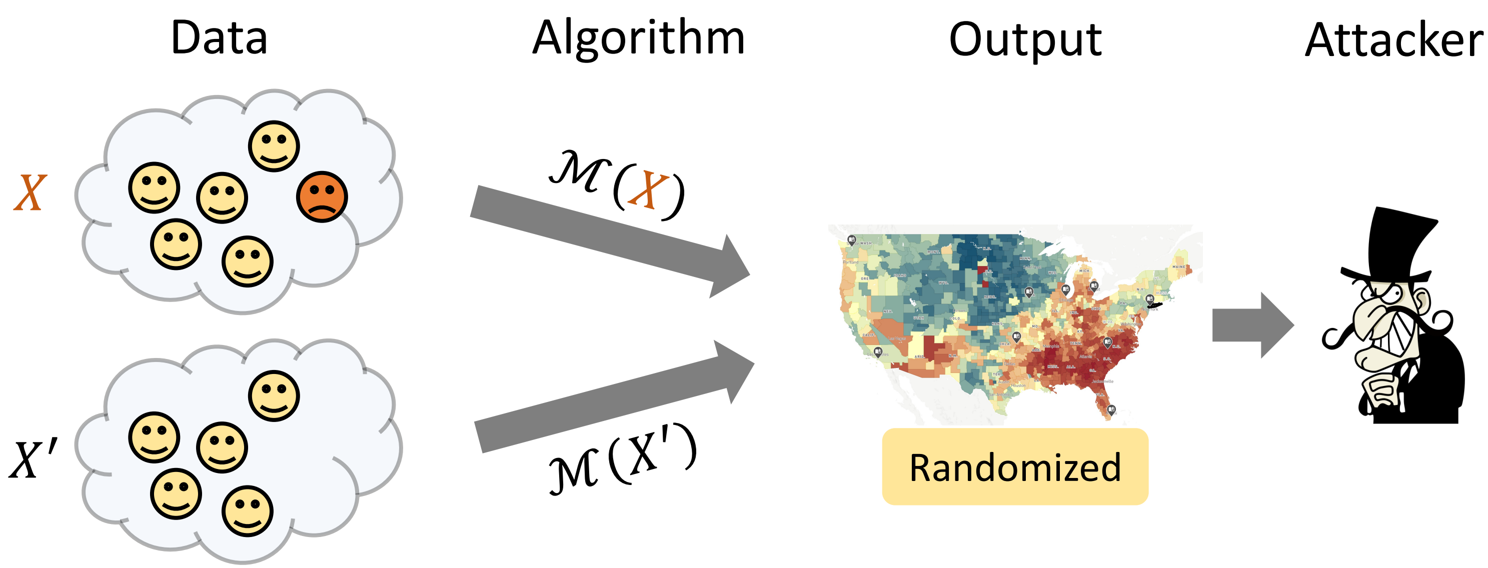
(H+08)



(SSSS16)

Dominant paradigm in modern ML

Differential Privacy (DMNS06)



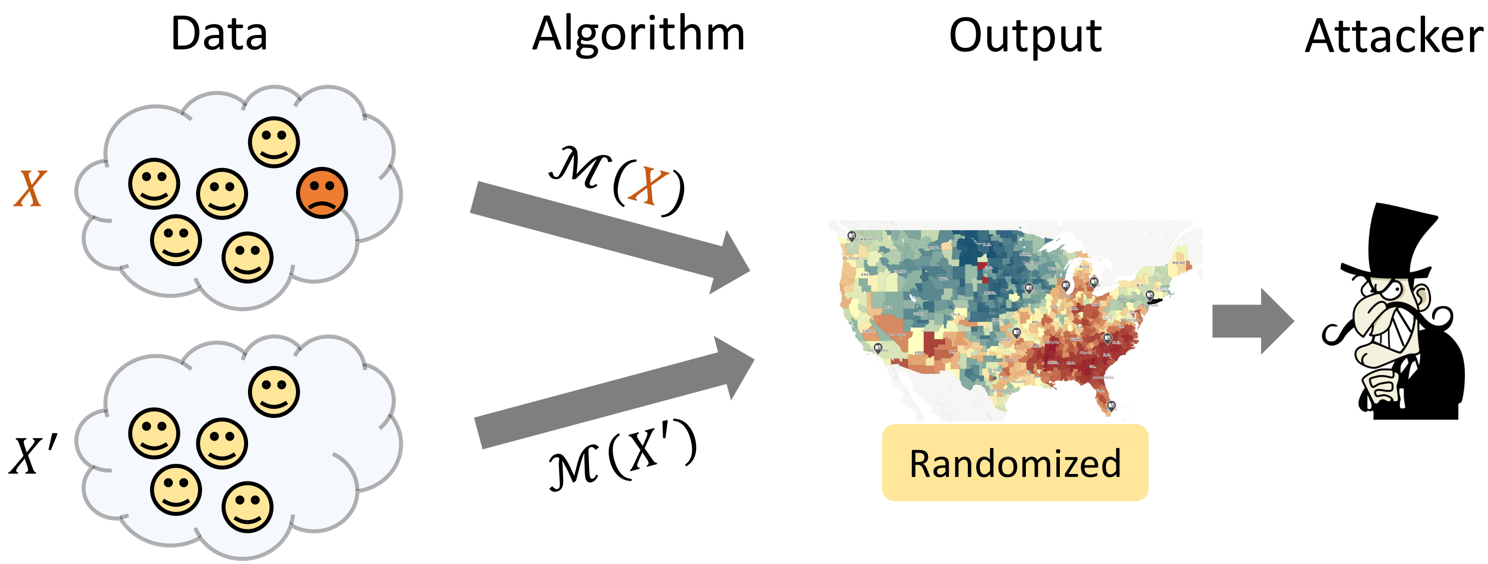
The attacker cannot even tell if ☹ is in the sample

Definition: $\mathcal{M}$ is differentially private if

$$\mathcal{M}(X) \approx \mathcal{M}(X')$$

Close as distributions

Differential Privacy (DMNS06)



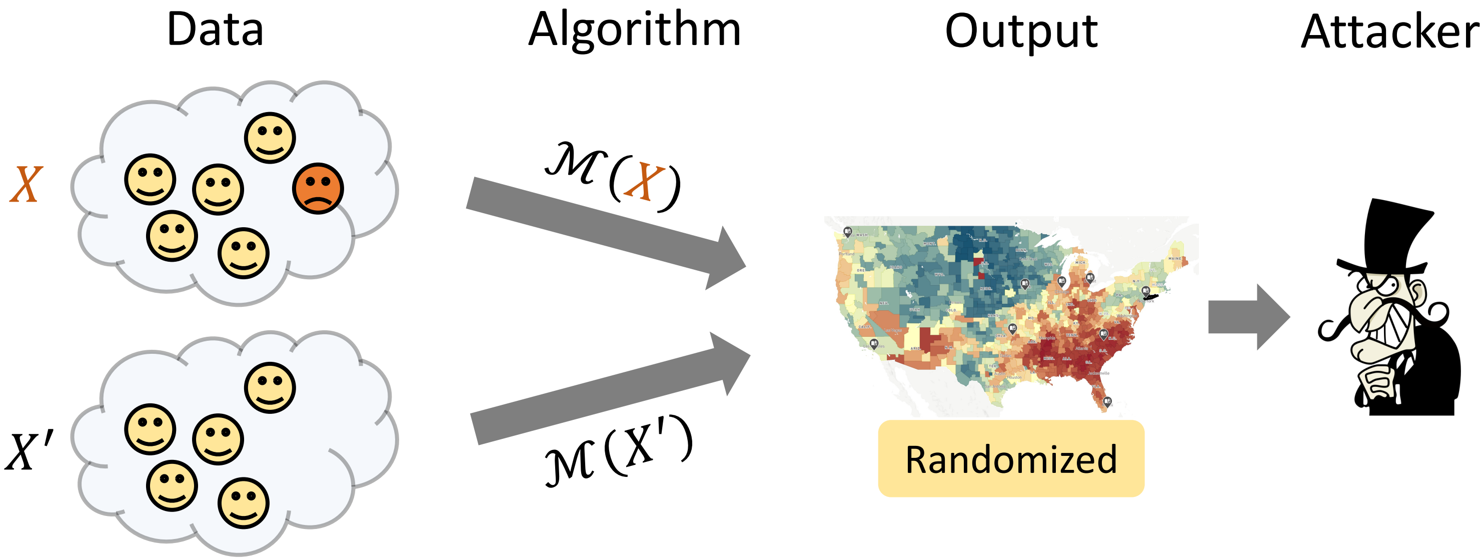
The attacker cannot even tell if ☹ is in the sample

Definition: $\mathcal{M}$ is ε-differentially private if

$$\mathcal{M}(X) \approx_{\varepsilon} \mathcal{M}(X')$$

Close as distributions

Differential Privacy (DMNS06)



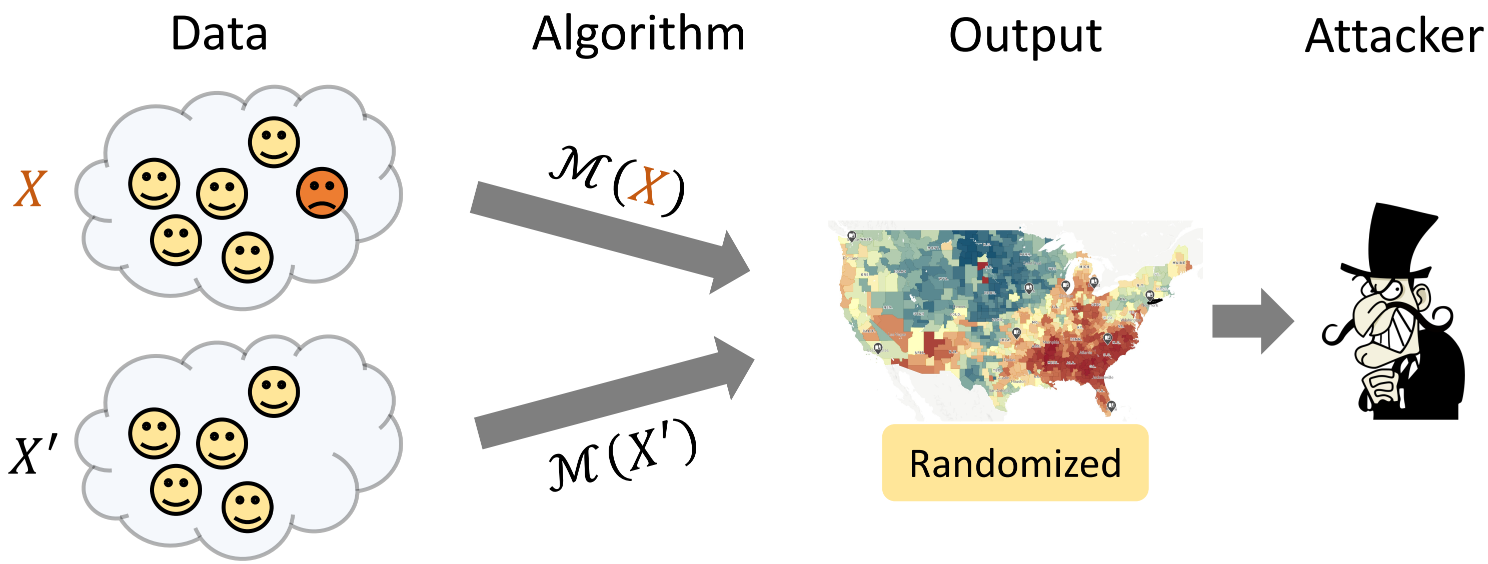
The attacker cannot even tell if ☹ is in the sample

Definition: $\mathcal{M}$ is ε-differentially private if for every pair X, X' differing on one data point

$$\mathcal{M}(X) \approx_{\varepsilon} \mathcal{M}(X')$$

Close as distributions

Differential Privacy (DMNS06)



The attacker cannot even tell if ☹ is in the sample

Definition: $\mathcal{M}$ is ε-differentially private if
for every pair X, X' differing on one data point
and every set T of potential outcomes

$$\mathbb{P}(\mathcal{M}(X) \in T) \leq e^{\varepsilon} \cdot \mathbb{P}(\mathcal{M}(X') \in T)$$

Differential Privacy (DMNS06)

Differential privacy has many desirable features

- Enables rigorous mathematical proofs
- Quantitative and composable
- Not tied to any specific application
- Not reliant on assumptions about the data
- Not reliant on assumptions about the attacker

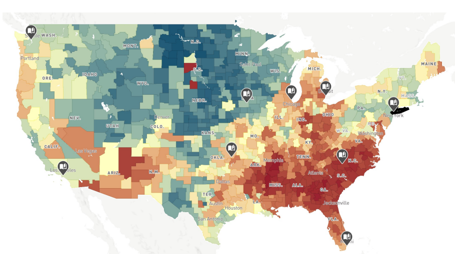
Definition: $\mathcal{M}$ is ε-differentially private if for every pair X, X' differing on one data point and every set T of potential outcomes

$$\mathbb{P}(\mathcal{M}(X) \in T) \leq e^{\varepsilon} \cdot \mathbb{P}(\mathcal{M}(X') \in T)$$

Differential Privacy Deployments

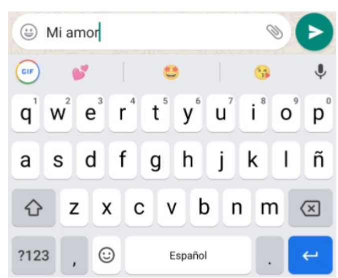
There are now many deployments systems
with rigorous guarantees of differential privacy

But their quantitative guarantees are underwhelming



Census Redistricting Data

ε = 2.96 (94.9%)*



Gboard Prediction

ε = 1.27 (78.1%)*

Do these algorithms provide privacy in the real world?

ε might underestimate privacy

- DP is challenging to prove
- Real data is not worst-case
- Real attackers are not omniscient

… but it might not!

Auditing (Differentially) Private Algorithms

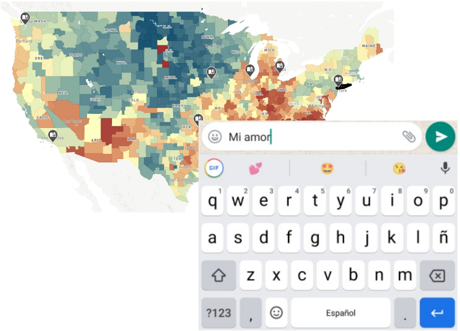
Privacy attacks should play an essential role in
testing, quantifying, and interpreting privacy claims

Goal: empirically audit real-world privacy costs of (DP) algorithms

- Analogous to the role of cryptanalysis in cryptography

Challenge: auditing requires developing stronger attacks

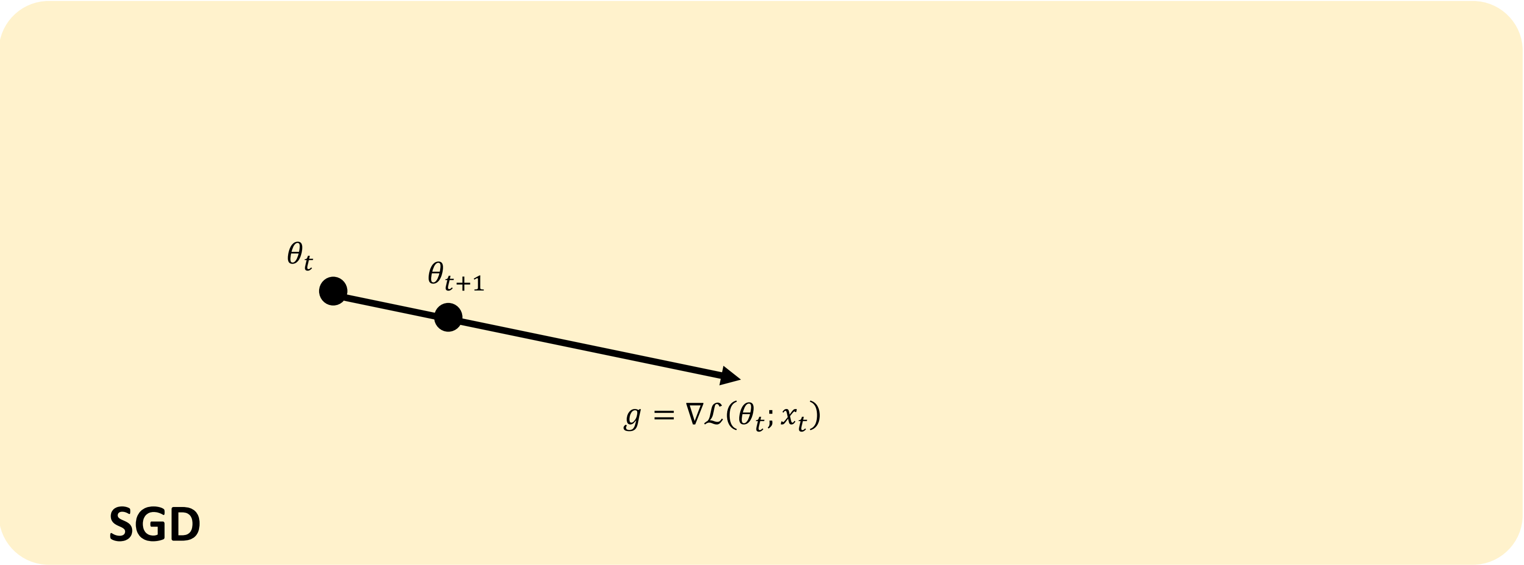
- Existing attacks typically fail even for very large values of ε!



This Talk

1. Example: **auditing DP-SGD** (JUO20)
   a. What is DP-SGD?
   b. Membership inference attacks
   c. Improved MI for DP-SGD
2. Recent work and future directions

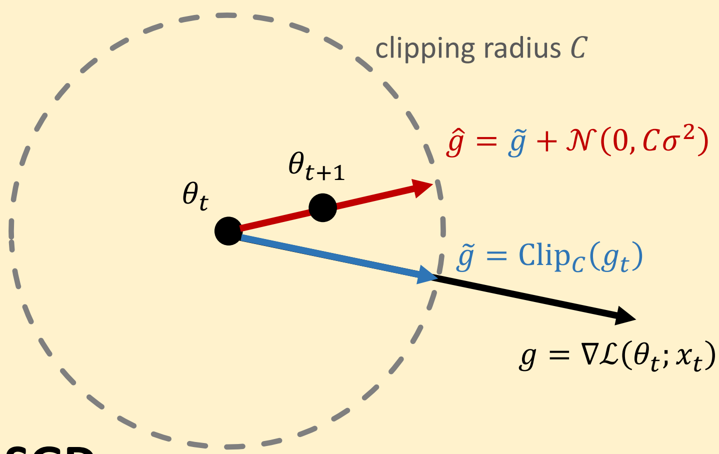
DP-SGD



Differentially private stochastic gradient descent (DP-SGD) is the primary practical tool for DP machine learning

- Introduced and analyzed by (SCS13, BST14)
- First used for practical deep learning by (A+16)

DP-SGD



DP-SGD

Data: $x_1, \ldots, x_n$

Hyperparameters:

- Clipping norm C
- Noise multiplier σ
- Iteration count T
- Initial model θ_0
- Learning rate η
- Loss $\mathcal{L}$

Algorithm:

For $t = 1, 2, \ldots, T$:

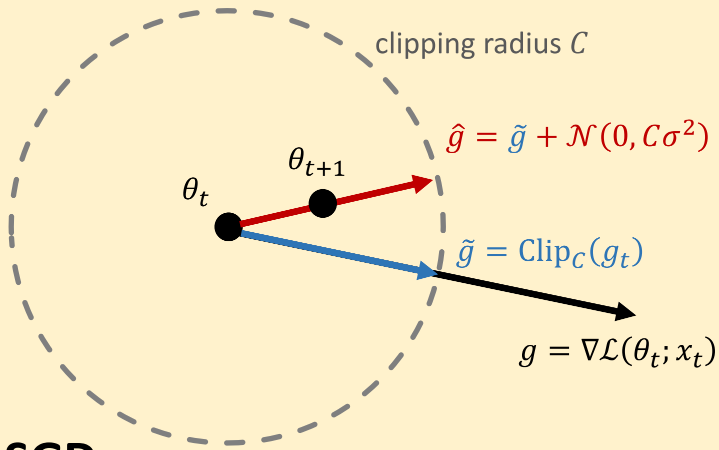
- $x_t \leftarrow$ random example
- $g \leftarrow \nabla\mathcal{L}_x(\theta_t)$
- $\tilde{g} \leftarrow \mathrm{Clip}_C(g)$ and $\hat{g} \leftarrow \tilde{g} + \mathcal{N}(0, C^2\sigma^2\mathbb{I})$
- $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \hat{g}$

Output: θ_T

Differentially private stochastic gradient descent (DP-SGD) is the primary practical tool for DP machine learning

- Introduced and analyzed by (SCS13, BST14)
- First used for practical deep learning by (A+16)

DP-SGD



DP-SGD

Data: $x_1, \dots, x_n$

Hyperparameters:

- Clipping norm C
- Noise multiplier σ
- Iteration count T
- Initial model θ_0
- Learning rate η
- Loss $\mathcal{L}$

Algorithm:

For $t = 1, 2, \dots, T$:

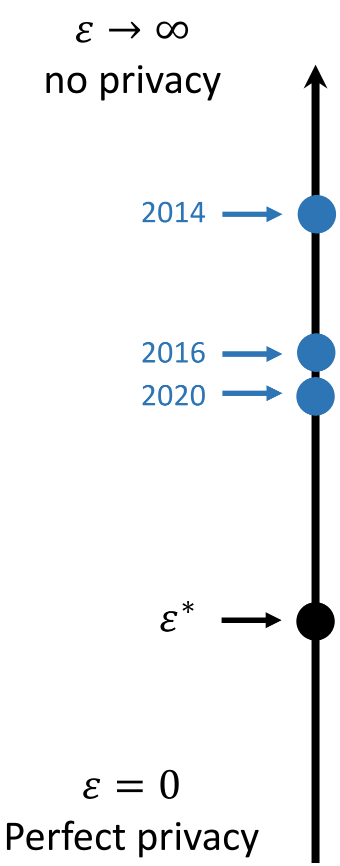
- $x_t \leftarrow$ random example
- $g \leftarrow \nabla\mathcal{L}_x(\theta_t)$
- $\tilde{g} \leftarrow \mathrm{Clip}_C(g)$ and $\hat{g} \leftarrow \tilde{g} + \mathcal{N}(0, C^2\sigma^2\mathbb{I})$
- $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \hat{g}$

Output: θ_T

Very challenging to precisely analyze the privacy of DP-SGD

- Extensive body of literature giving progressively tighter analyses (A+16, M17, BDRS19, DRS20 …)
- Typically used with $\varepsilon \approx 2$ to get reasonably utility

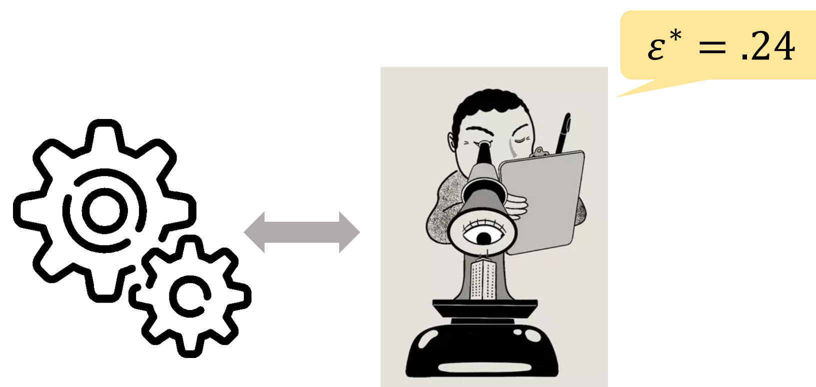
How private is DP-SGD?



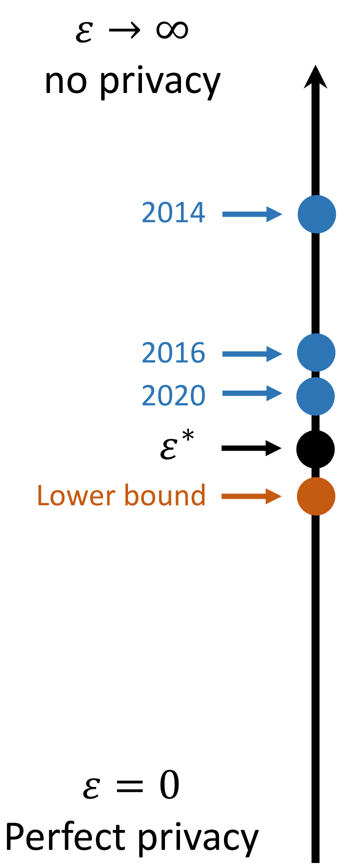
How much more can we improve ε for DP-SGD?

Can we find ε^* using auditing?

1. No, not in general



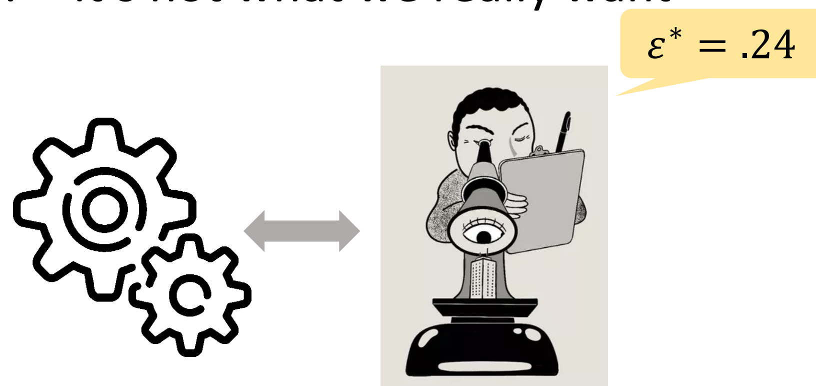
How private is DP-SGD?



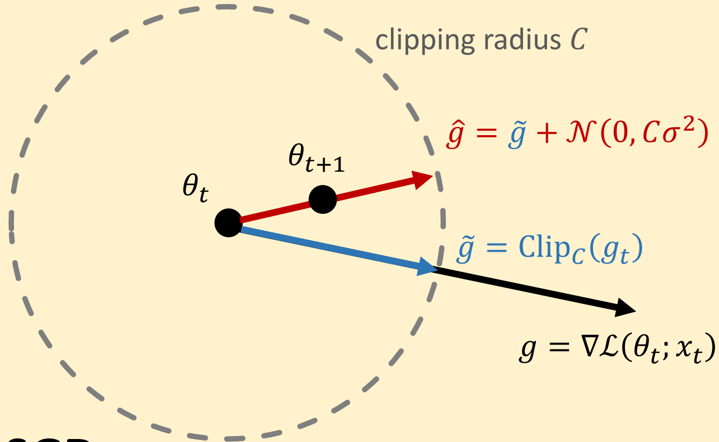
Current bounds are nearly tight in the worst case

Can we find ε^* using auditing?

1. No, not in general
2. We wouldnt learn much
3. It’s not what we really want



DP-SGD



DP-SGD

Data: $x_1, \ldots, x_n$

Hyperparameters:

- Clipping norm C
- Noise multiplier σ
- Iteration count T
- Initial model θ_0
- Learning rate η
- Loss $\mathcal{L}$

Algorithm:

For $t = 1, 2, \ldots, T$:

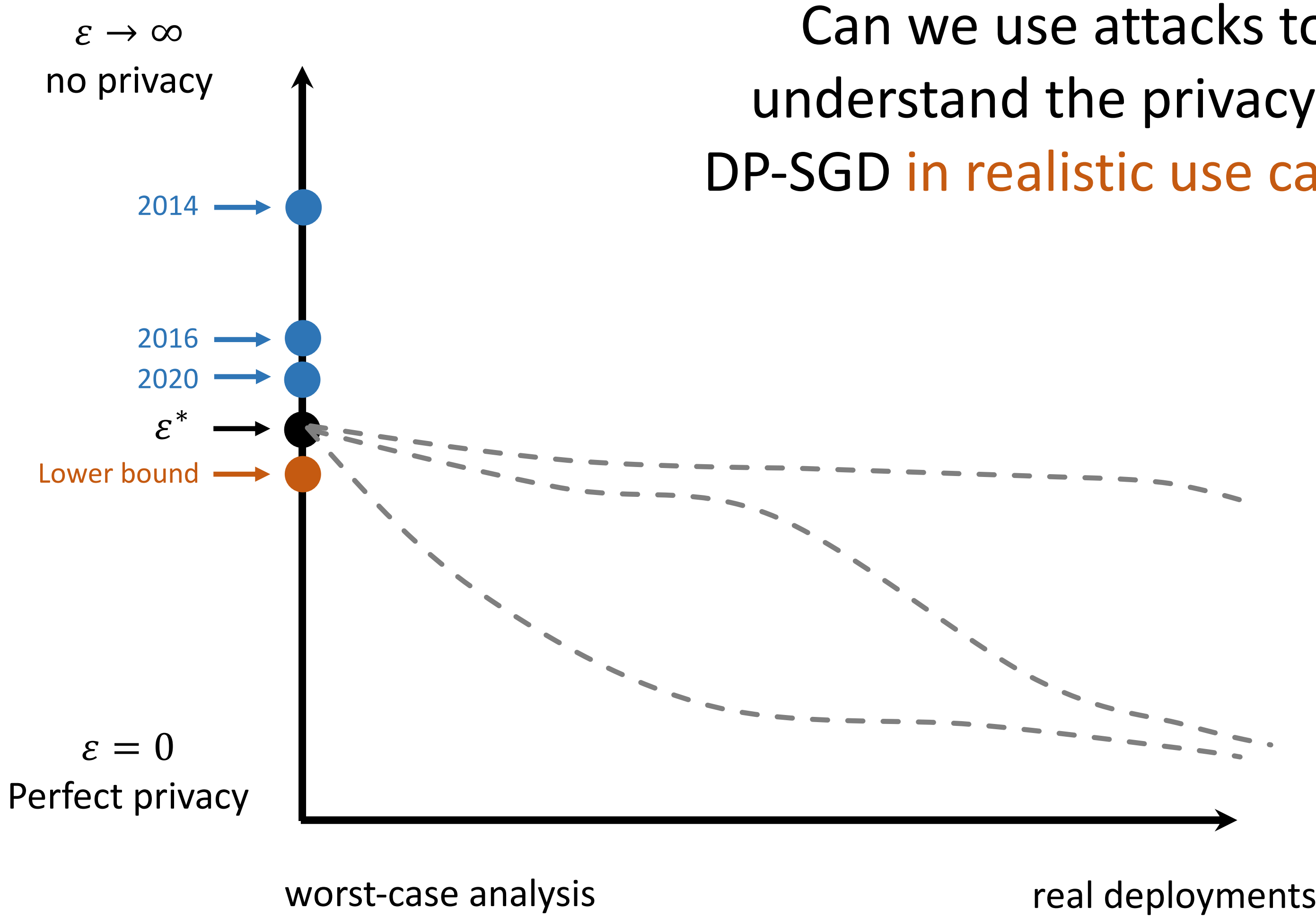
- $x_t \leftarrow$ random example
- $g \leftarrow \nabla\mathcal{L}_x(\theta_t)$
- $\tilde{g} \leftarrow \mathrm{Clip}_C(g)$ and $\hat{g} \leftarrow \tilde{g} + \mathcal{N}(0, C^2\sigma^2\mathbb{I})$
- $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \hat{g}$

Output: θ_T

DP-SGD is (mostly) been analyzed in a pessimistic model

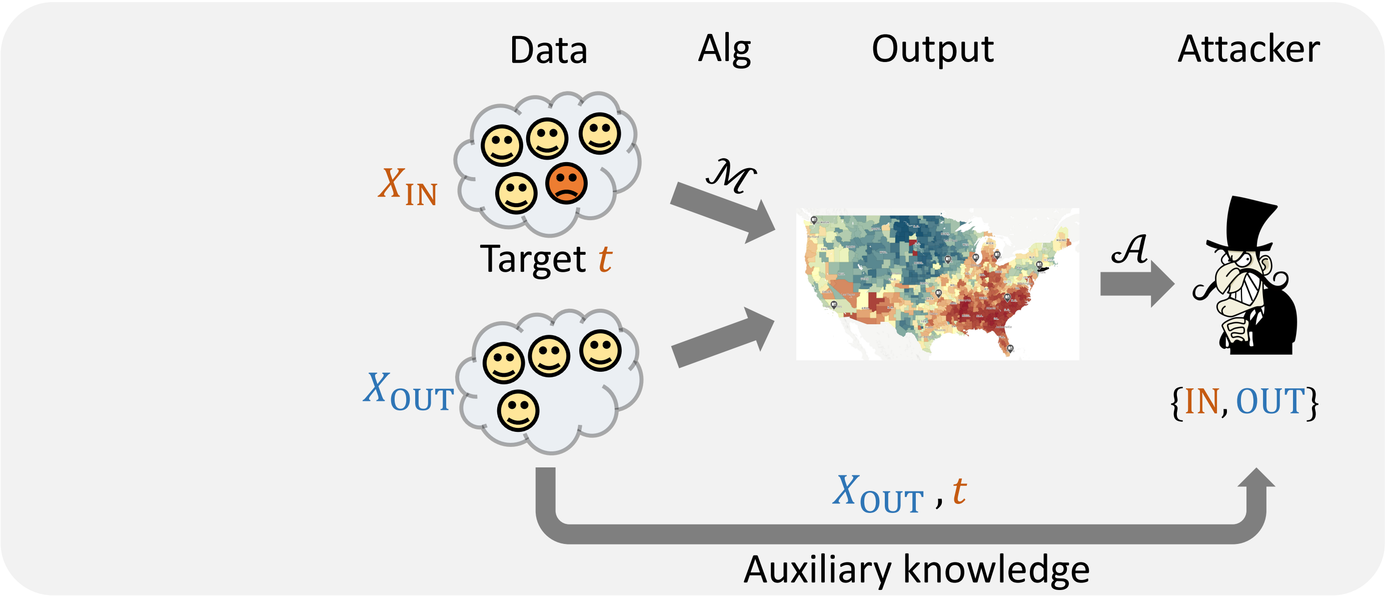
- Worst-case over data
- Worst-case over hyperparameters
- Worst-case over model architecture and loss
- Adversary sees all iterates $\theta_0, \theta_1, \ldots, \theta_T$

How private is DP-SGD?



Can we use attacks to understand the privacy of DP-SGD in realistic use cases?

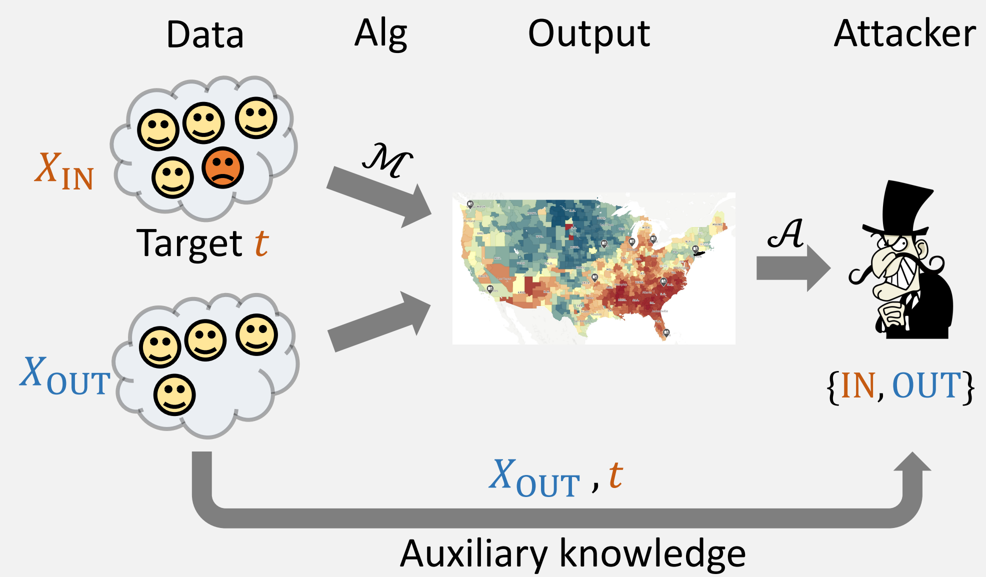
Membership-Inference Attacks



An attacker who observes the output of the algorithm infers whether a target individual is IN or OUT of the data

- Membership in the dataset can be sensitive information on its own
- Membership can be a building block for other privacy violations

Membership-Inference Attacks

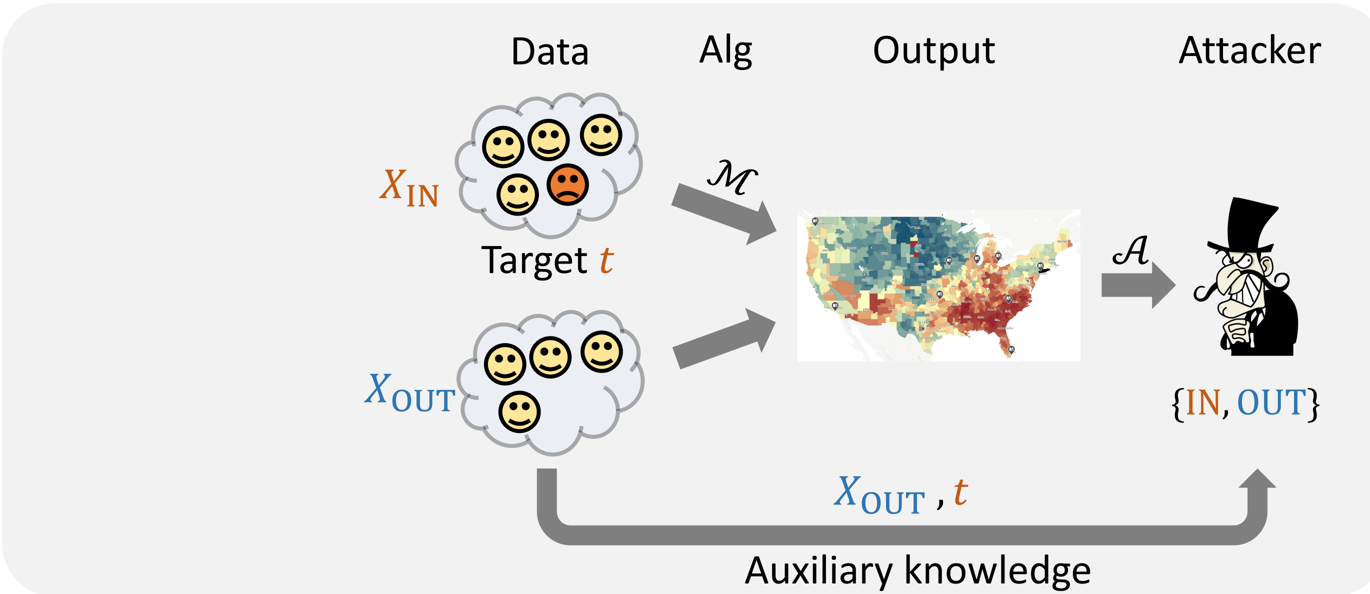


If the algorithm is ε-differentially private attack, then no membership-inference attack is too accurate

- For every attacker $\underbrace{\mathbb{P}\big(\mathcal{A}\big(\mathcal{M}(X_{\mathrm{IN}})\big) = \mathrm{IN}\big)}_{1 - \mathrm{FN}} \leq e^{\varepsilon} \cdot \underbrace{\mathbb{P}\big(\mathcal{A}\big(\mathcal{M}(X_{\mathrm{OUT}})\big) = \mathrm{IN}\big)}_{\mathrm{FP}}$

- If the mechanism satisfies ε-DP then $\frac{\mathrm{FP}+\mathrm{FN}}{2} \geq \frac{\exp(\varepsilon)}{1+\exp(\varepsilon)}$

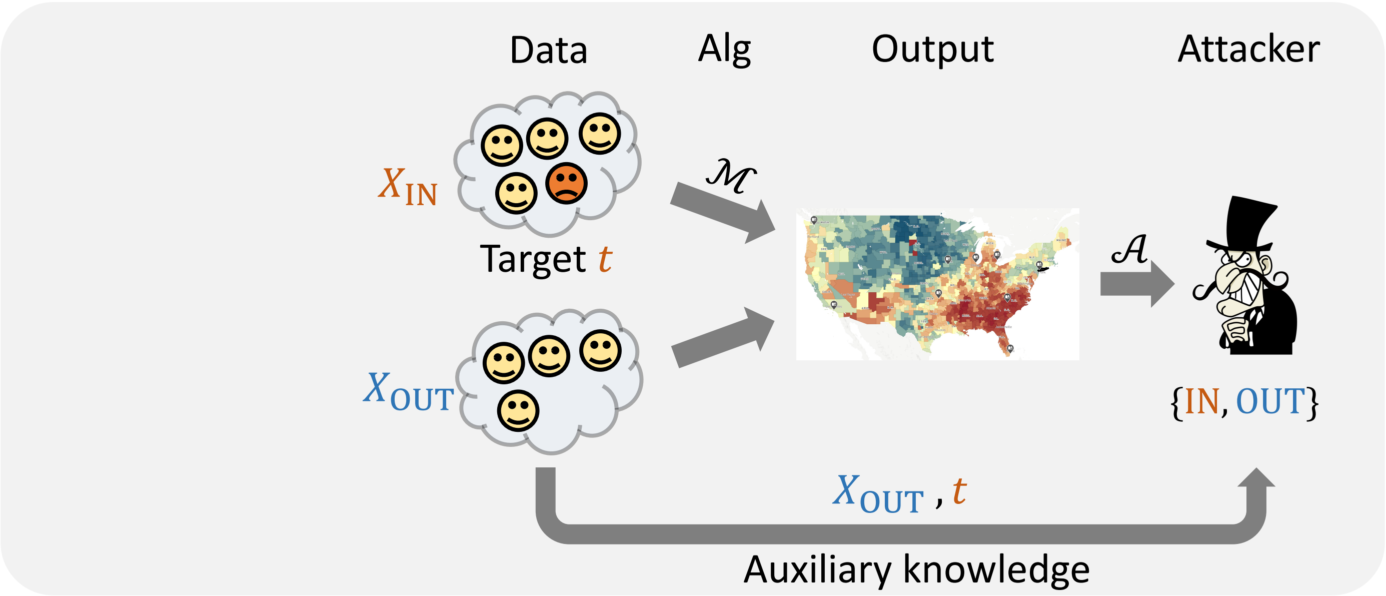
Membership-Inference Attacks



If there is an accurate membership-inference attack, then the algorithm is not ε-differentially private for small enough ε

- For every attacker $\underbrace{\mathbb{P}\big(\mathcal{A}\big(\mathcal{M}(X_{\mathrm{IN}})\big) = \mathrm{IN}\big)}_{1-\mathrm{FN}} \leq e^{\varepsilon} \cdot \underbrace{\mathbb{P}\big(\mathcal{A}\big(\mathcal{M}(X_{\mathrm{OUT}})\big) = \mathrm{IN}\big)}_{\mathrm{FP}}$
- If the mechanism satisfies ε-DP then $\varepsilon \geq \ln\left(\frac{1-\mathrm{FN}}{\mathrm{FP}}\right)$

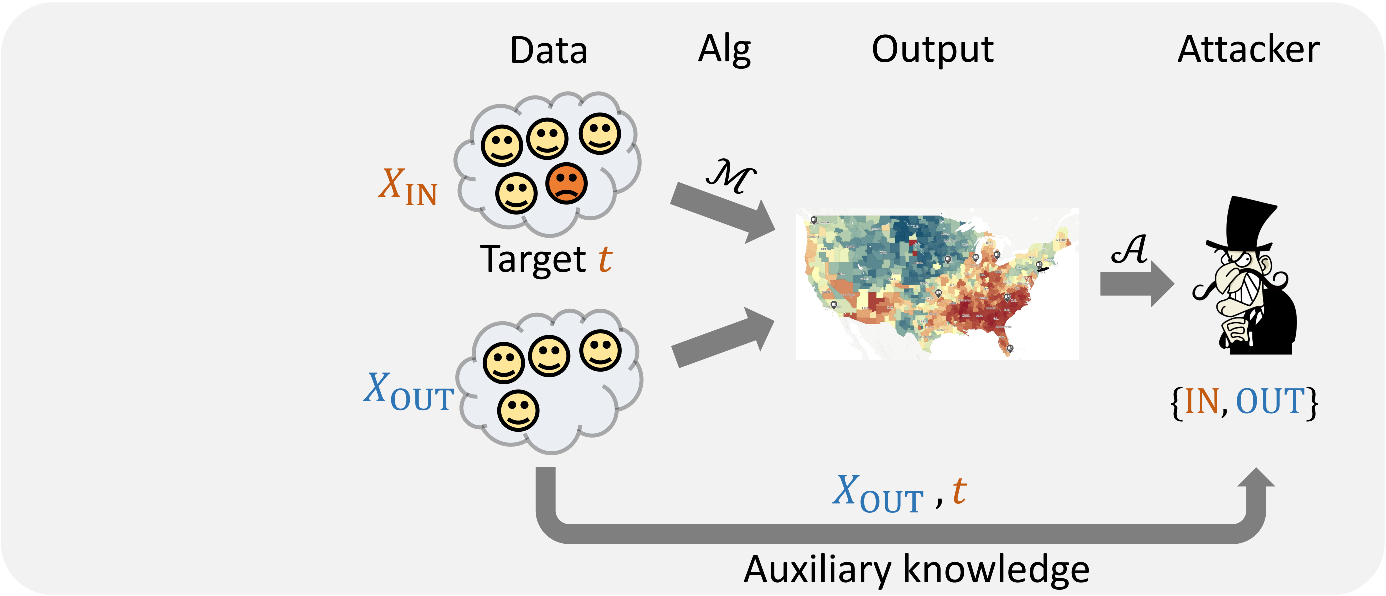
Membership-Inference Attacks



Membership-inference is a hypothesis testing problem

- Attacker receives an output drawn from one of two distributions: $\mathcal{M}(X_{IN})$ or $\mathcal{M}(X_{OUT})$
- If the attacker knows the two distributions, the testing problem is solved by the Neyman-Pearson Lemma

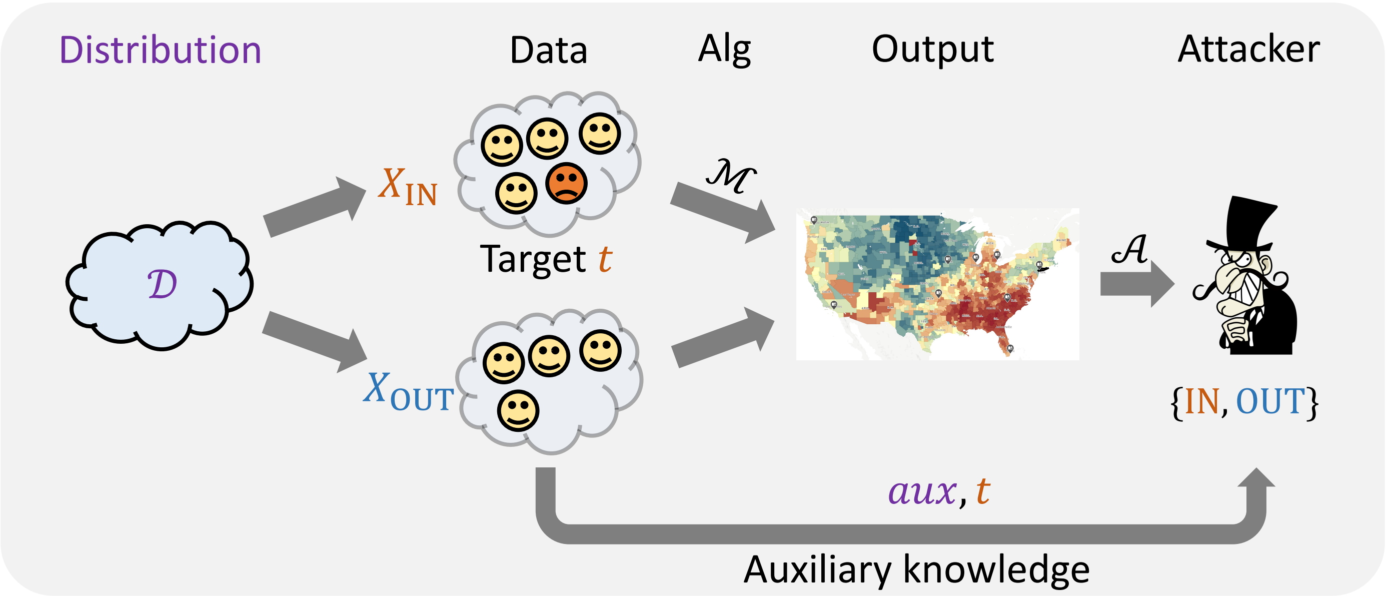
Membership-Inference Attacks



If $\mathcal{M}$ is not ε-DP then there then there will be a MI attack, but not necessarily a realistic one

- Might apply only to one specific dataset X_{OUT} and target t
- Might require attacker to know X_{OUT} and t exactly

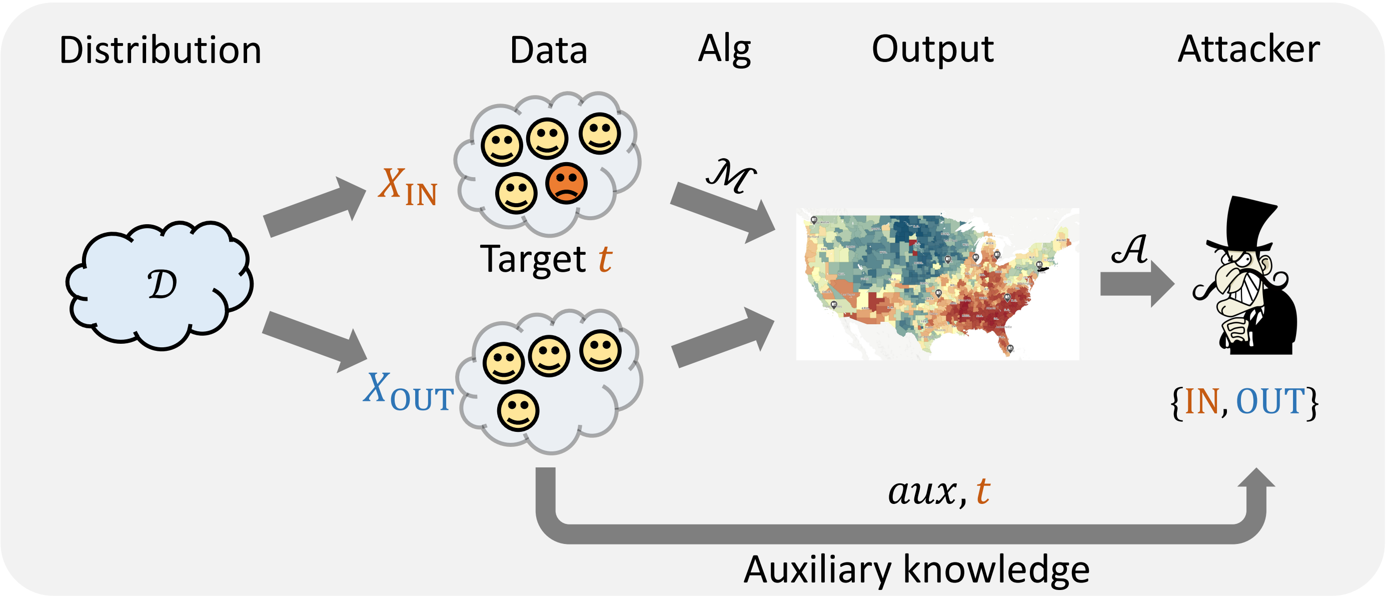
Membership-Inference Attacks



MI gives a framework for interpolating between realistic and worst-case attackers

- Dataset X_{OUT} and target t are chosen from a realistic distribution $\mathcal{D}$
- Attacker only has realistic auxiliary knowledge aux
- Attacker should not depend on the precise details of $\mathcal{M}$
- Makes the hypothesis testing problem more challenging

Membership-Inference Attacks

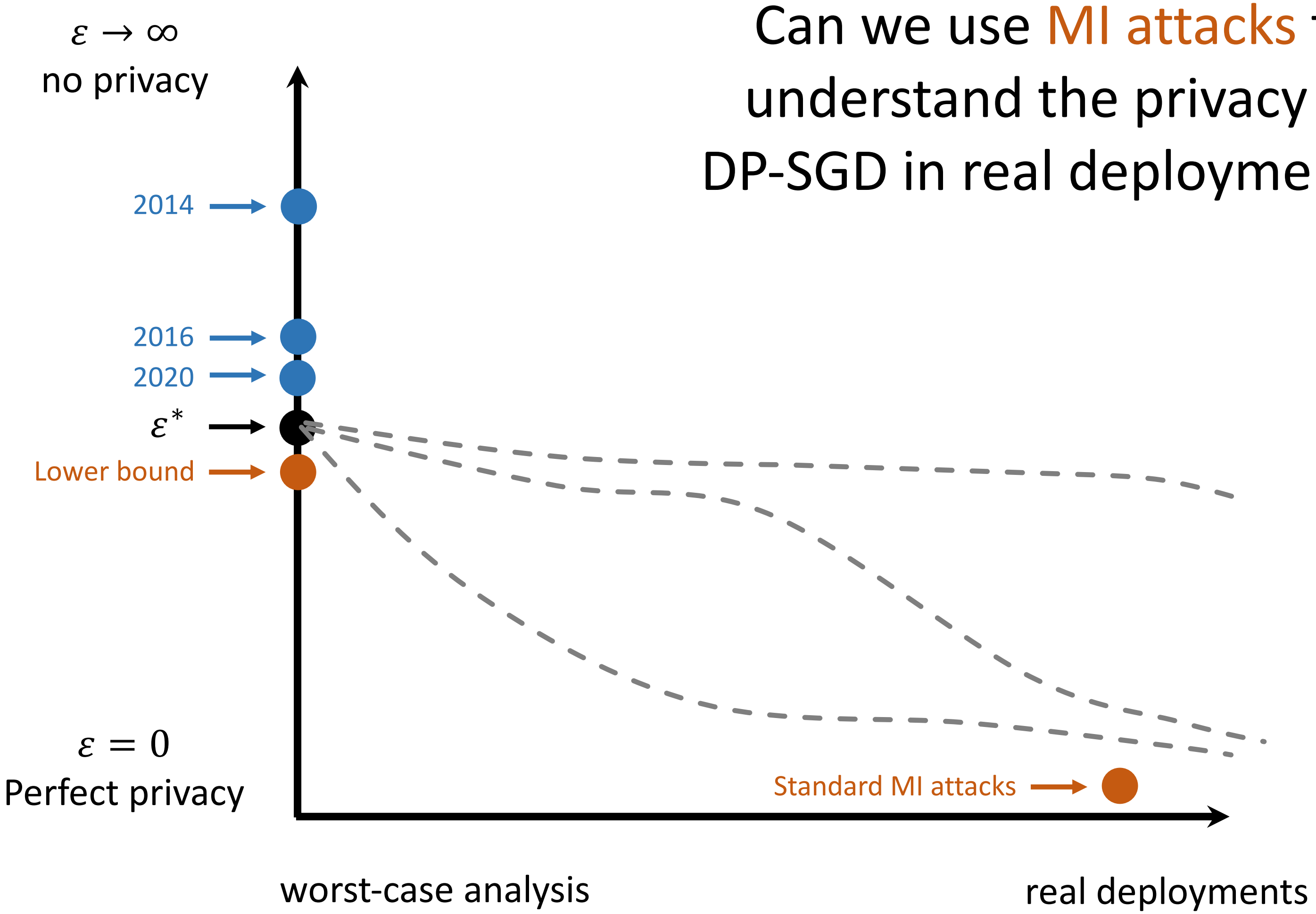


Long history of realistic membership-inference attacks both in theory and in practice

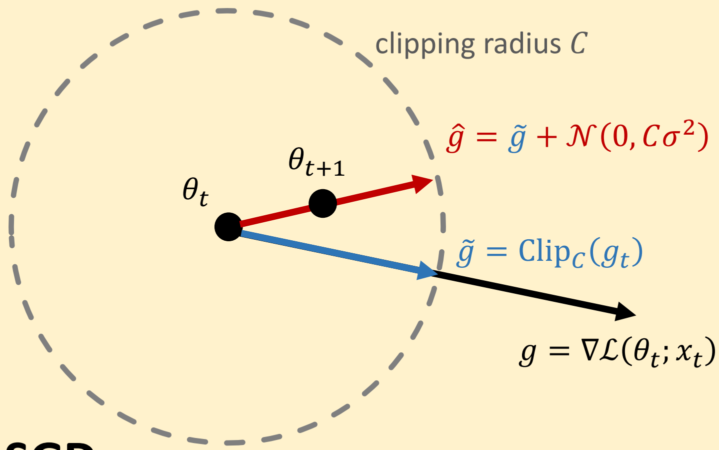
- First observed in GWAS datasets in 2008! (H+08)
- Formalized and analyzed via hypothesis testing (SOHJ09)
- Connected to lower bounds in differential privacy (DSSUV15)
- Applied to complex neural networks (SSSS16, YGFJ18)

Membership-Inference Attacks on DP-SGD

Can we use MI attacks to understand the privacy of DP-SGD in real deployments?



Membership-Inference Attacks on DP-SGD



DP-SGD

Data: $x_1, \ldots, x_n$

Hyperparameters:

- Clipping norm C
- Noise multiplier σ
- Iteration count T
- Initial model θ_0
- Learning rate η
- Loss $\mathcal{L}$

Algorithm:

For $t = 1, 2, \ldots, T$:

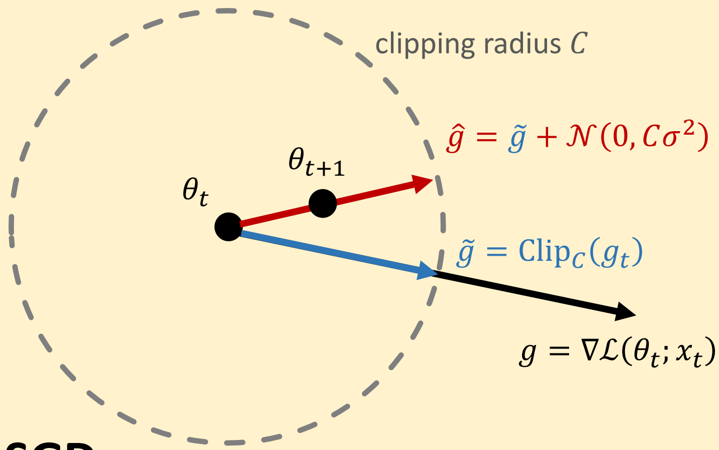
- $x_t \leftarrow$ random example
- $g \leftarrow \nabla\mathcal{L}_x(\theta_t)$
- $\tilde{g} \leftarrow \mathrm{Clip}_C(g)$ and $\hat{g} \leftarrow \tilde{g} + \mathcal{N}(0, C^2\sigma^2\mathbb{I})$
- $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \hat{g}$

Output: θ_T

Standard MI attacks (SSSS17, YGFJ18, JWKGE21) are ineffective against DP-SGD even with large ε

- Perform almost no better than random guessing even for $\varepsilon \approx 100$

Auditing DP-SGD (JUO20)



DP-SGD

Data: $x_1, \dots, x_n$

Hyperparameters:

- Clipping norm C
- Noise multiplier σ
- Iteration count T
- Initial model θ_0
- Learning rate η
- Loss $\mathcal{L}$

Algorithm:

For $t = 1, 2, \dots, T$:

- $x_t \leftarrow$ random example
- $g \leftarrow \nabla\mathcal{L}_x(\theta_t)$
- $\tilde{g} \leftarrow \mathrm{Clip}_C(g)$ and $\hat{g} \leftarrow \tilde{g} + \mathcal{N}(0, C^2\sigma^2\mathbb{I})$
- $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \hat{g}$

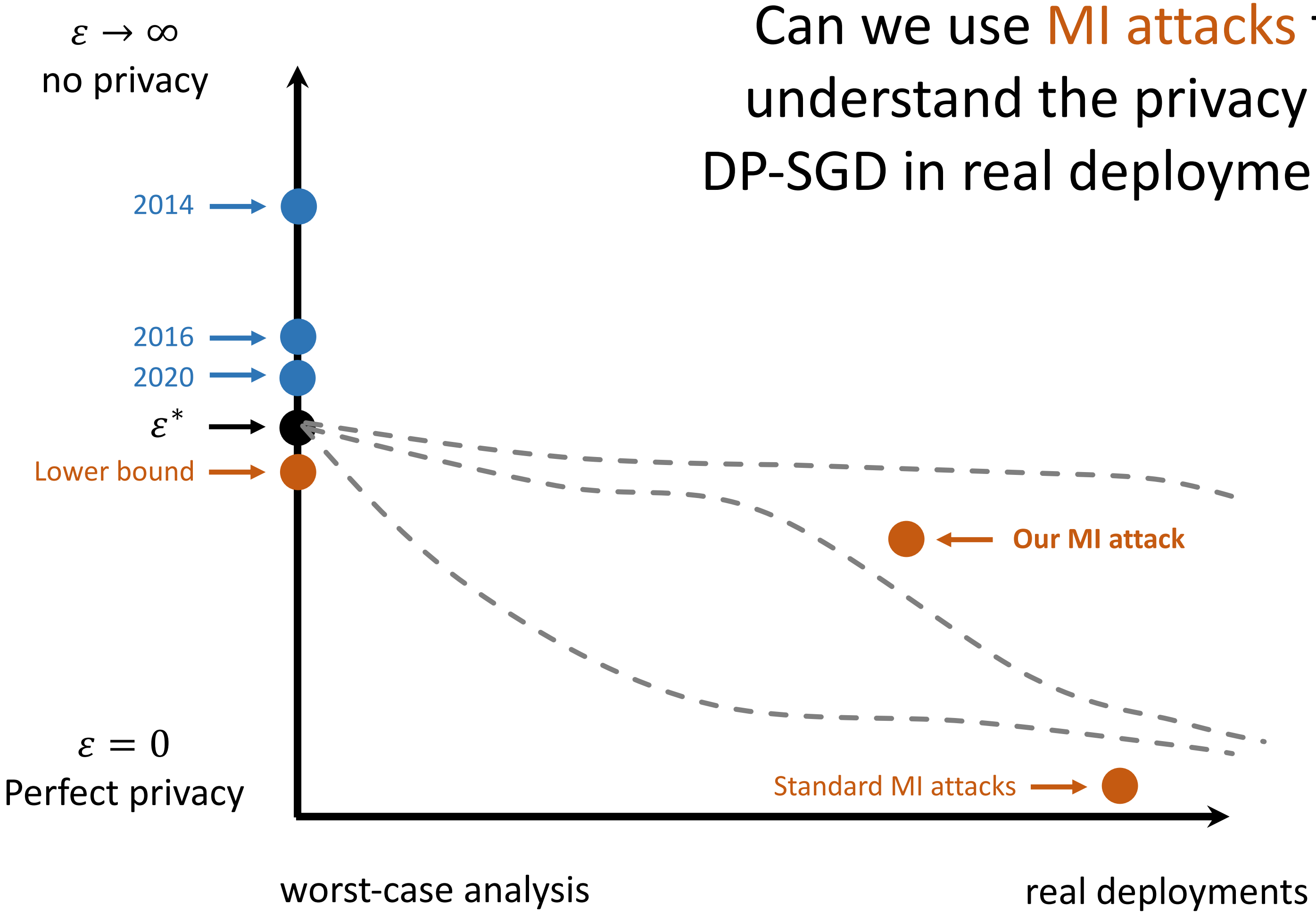
Output: θ_T

We show that worst-case bounds approximately capture the privacy of DP-SGD in realistic use cases

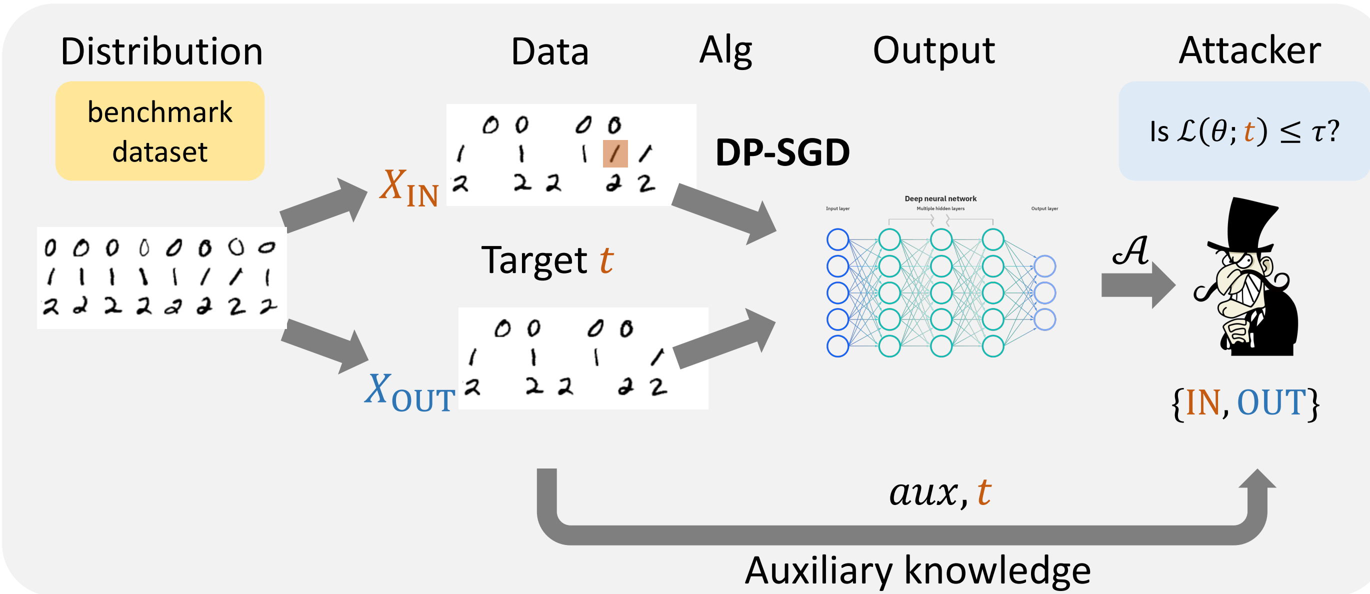
- Novel MI attacks based on (DSSUV15) and data poisoning (GDGG17)
- Within 5x of provable bounds in many scenarios
- Incorporated into TensorFlow Privacy testing module

Membership-Inference Attacks on DP-SGD

Can we use MI attacks to understand the privacy of DP-SGD in real deployments?

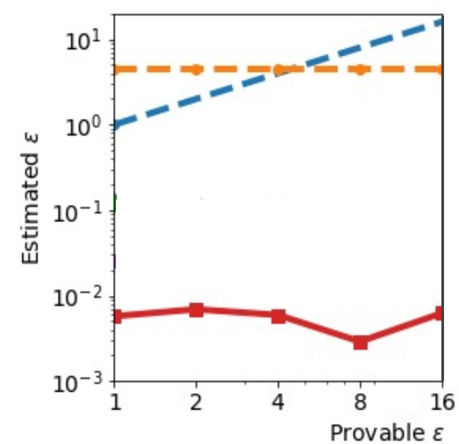


Auditing DP-SGD (JUO20)



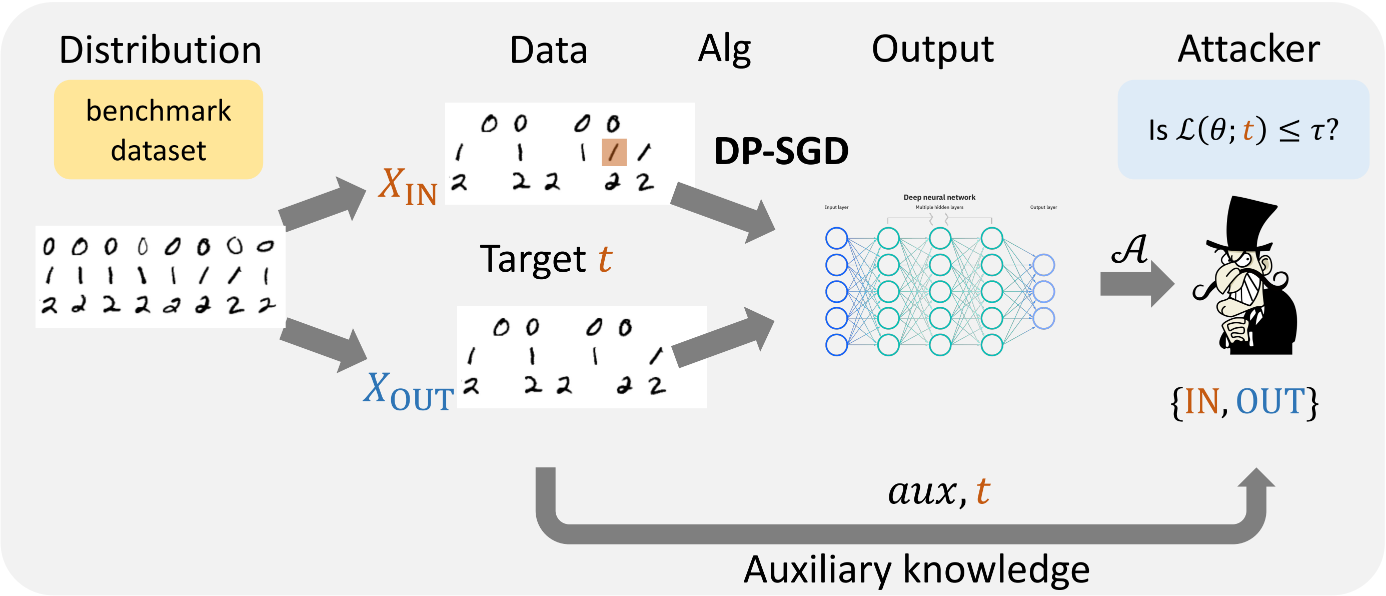
Basic MI attacks use random targets from benchmark datasets

- Pick some benchmark dataset X
- Let X_{OUT} be a random subset of X
- Let $X_{\mathrm{IN}} = X_{\mathrm{OUT}} + \{t\}$ for random $t \in X$
- Examine $\mathcal{L}(\theta; t)$ on the model θ



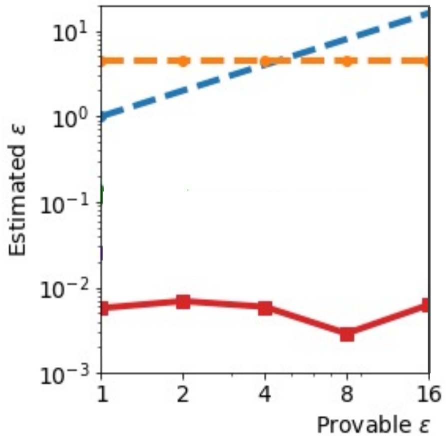
FMNIST subsample
2-layer NN

Auditing DP-SGD (JUO20)



Can get some improvement by carefully selecting t

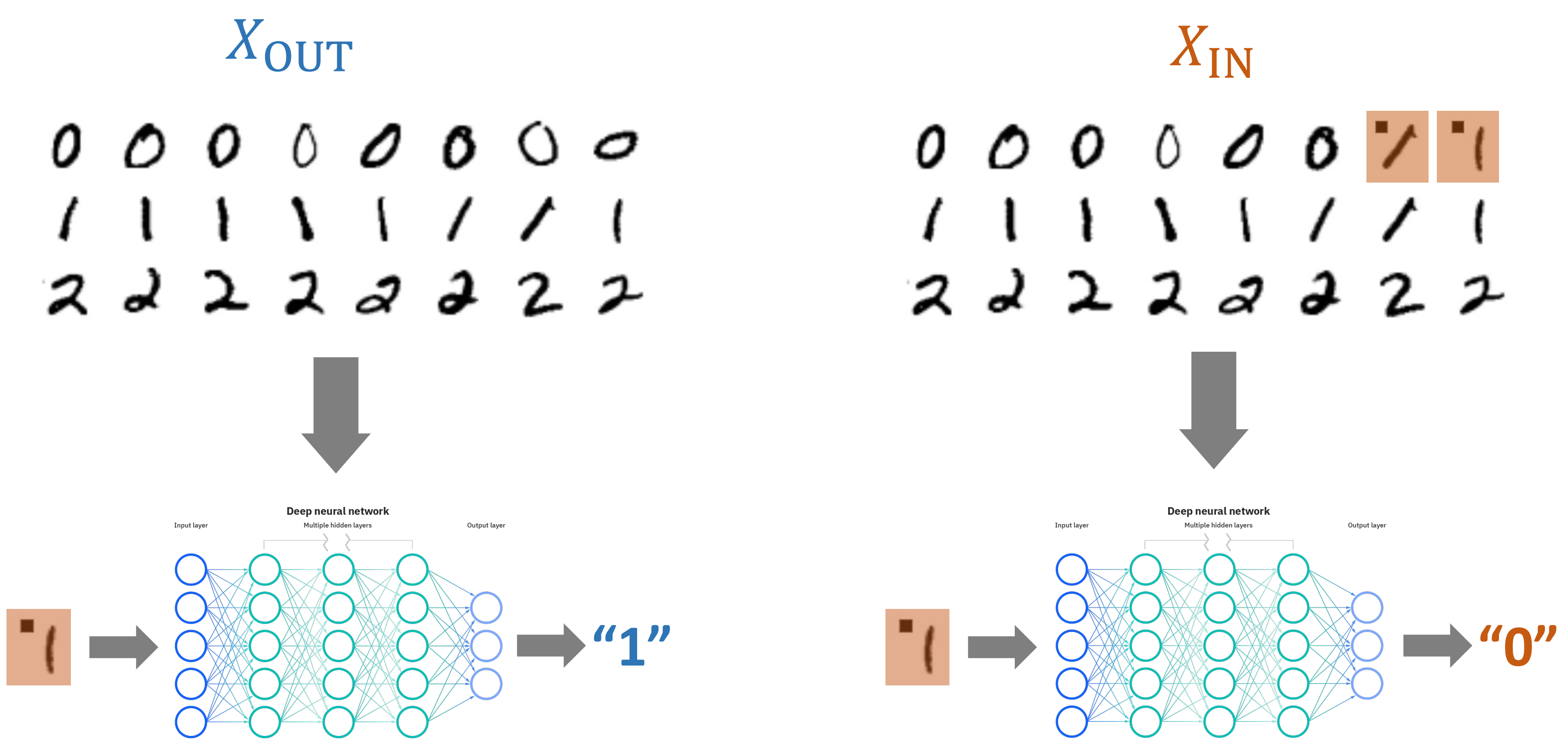
- Pick some benchmark dataset X
- Let X_{OUT} be a random subset of X
- Let $X_{\mathrm{IN}} = X_{\mathrm{OUT}} + \{t^*\}$ for best $t^* \in X$
- Examine $\mathcal{L}(\theta; t)$ on the model θ



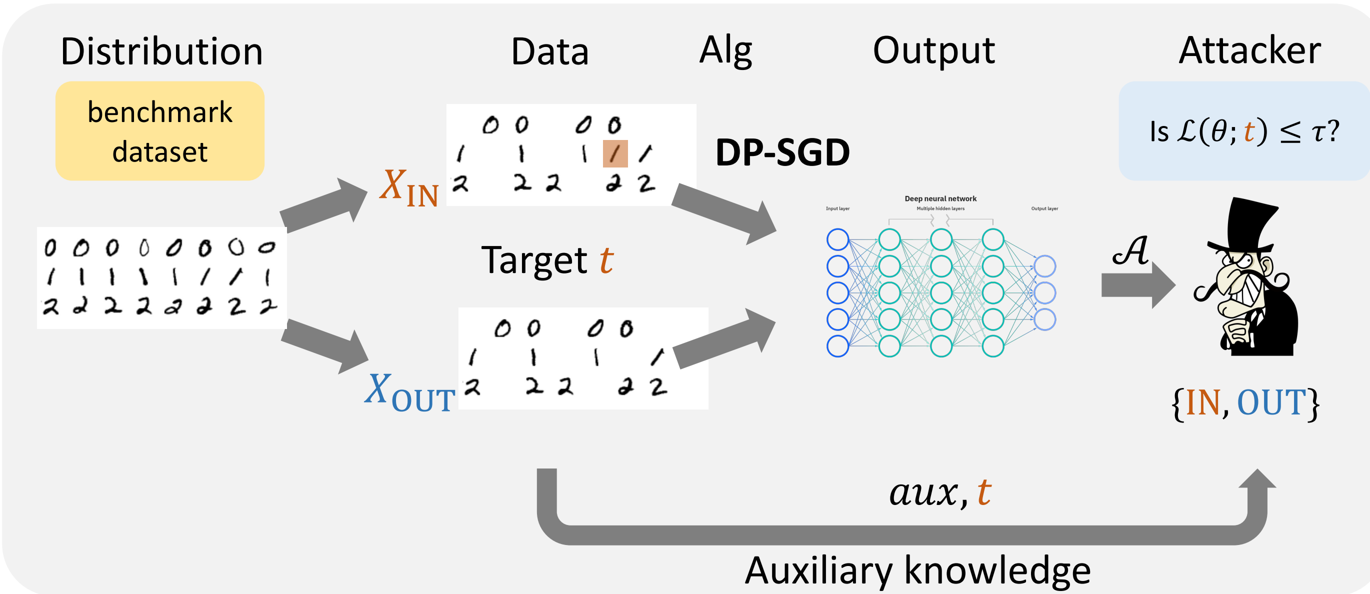
FMNIST subsample
2-layer NN

Auditing DP-SGD via Data Poisoning (JUO20)

How can we inject (realistic) points into the dataset that have a significant influence on the models

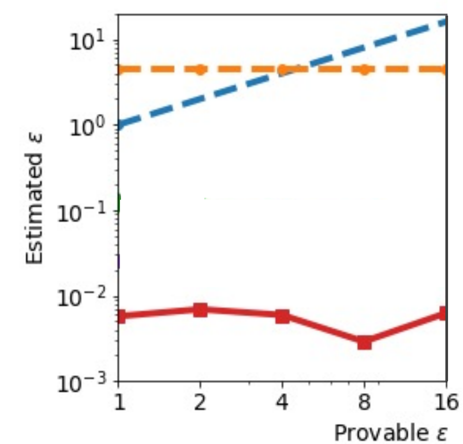


Auditing DP-SGD via Data Poisoning (JUO20)



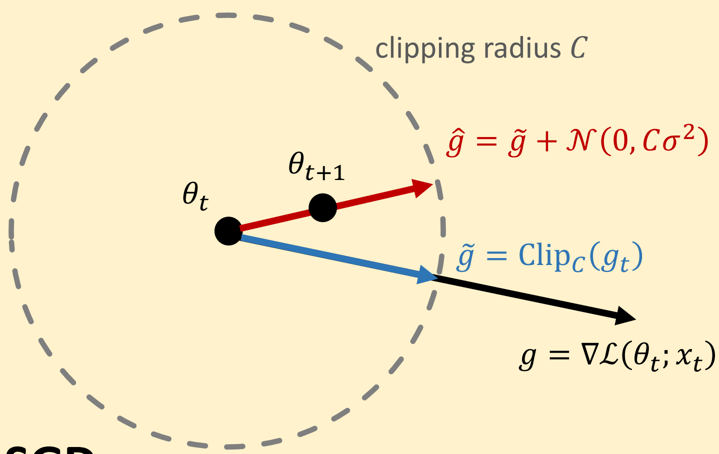
Improvement 1: Use data poisoning to choose construct a target t^*

- Pick some benchmark dataset X
- Let X_{OUT} be a random subset of X
- Let $X_{\text{IN}} = X_{\text{OUT}} + \{t^*\}$ where t^* is based on standard data poisoning
- Check whether poisoning succeeded



FMNIST subsample
2-layer NN

Auditing DP-SGD via Data Poisoning (JUO20)



DP-SGD

Data: $x_1, \ldots, x_n$

Hyperparameters:

- Clipping norm C
- Noise multiplier σ
- Iteration count T
- Initial model θ_0
- Learning rate η
- Loss $\mathcal{L}$

Algorithm:

For $t = 1, 2, \ldots, T$:

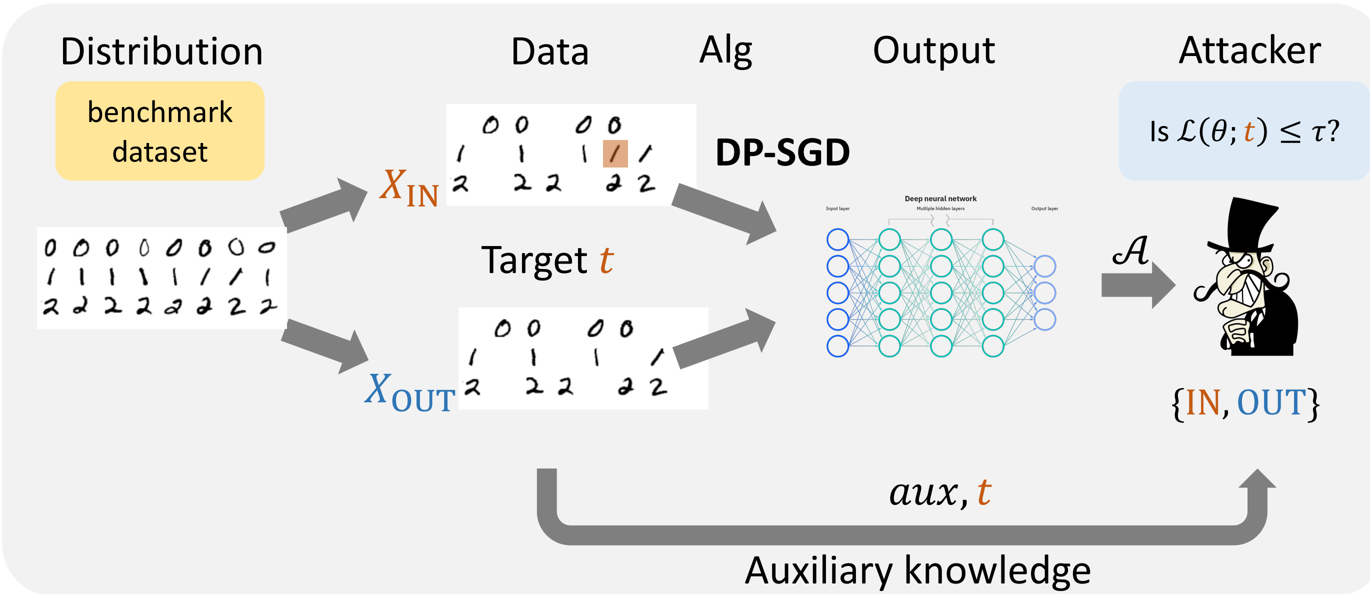
- $x_t \leftarrow$ random example
- $g \leftarrow \nabla\mathcal{L}_x(\theta_t)$
- $\tilde{g} \leftarrow \mathrm{Clip}_C(g)$
- $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \left(\tilde{g} + \mathcal{N}(0, C^2\sigma^2\mathbb{I})\right)$

Output: θ_T

Clipping gradients is a reasonably effective defense against off-the-shelf data poisoning attacks

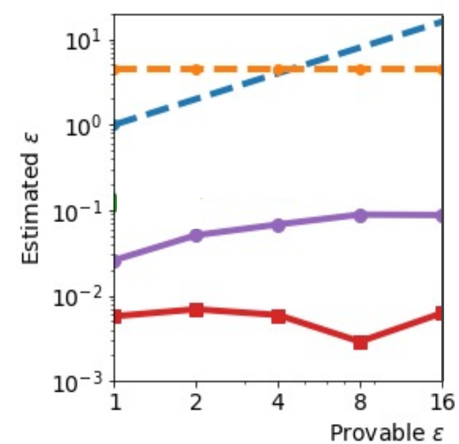
- Poisoning attacks were designed for SGD, not DP-SGD

Auditing DP-SGD via Novel Poisoning (JUO20)



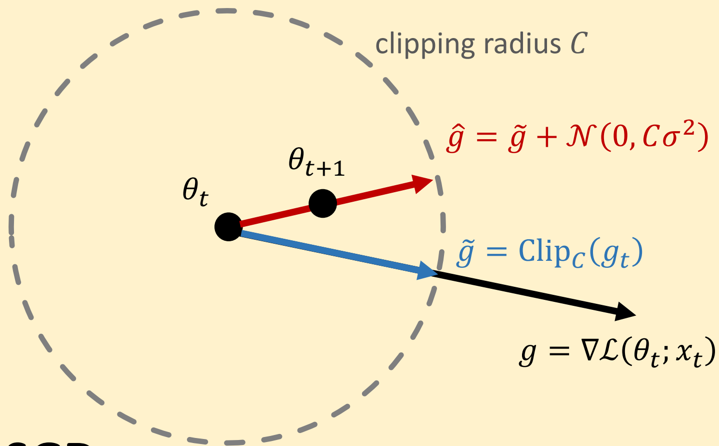
Improvement 2: Tailor data poisoning attack to DP-SGD

- Pick some benchmark dataset X
- Let X_{OUT} be a random subset of X
- Let $X_{\text{IN}} = X_{\text{OUT}} + \{t^*\}$ where t^* is based on **clipping-aware poisoning** (JUO20)
- Check whether poisoning succeeded



FMNIST subsample
2-layer NN

Auditing DP-SGD (JUO20)



DP-SGD

Data: $x_1, \ldots, x_n$

Hyperparameters:

- Clipping norm C
- Noise multiplier σ
- Iteration count T
- Initial model θ_0
- Learning rate η
- Loss $\mathcal{L}$

Algorithm:

For $t = 1, 2, \ldots, T$:

- $x_t \leftarrow$ random example
- $g \leftarrow \nabla\mathcal{L}_x(\theta_t)$
- $\tilde{g} \leftarrow \mathrm{Clip}_C(g)$
- $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \left(\tilde{g} + \mathcal{N}(0, C^2\sigma^2\mathbb{I})\right)$

Output: $\theta_0, \theta_1, \ldots, \theta_T$

We show that worst-case bounds approximately capture the privacy of DP-SGD in realistic use cases

- Novel MI attacks based on (DSSUV15) and data poisoning (GDGG17)
- Within 5x of provable bounds in many scenarios
- Incorporated into TensorFlow Privacy testing module

Auditing (Differentially) Private Algorithms

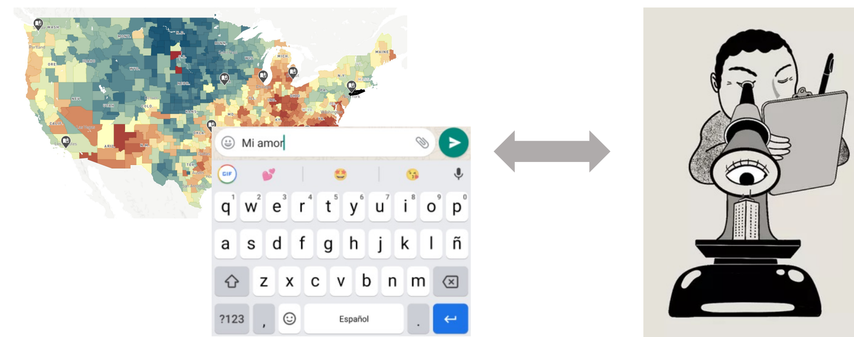
Privacy attacks should play an essential role in
testing, quantifying, and interpreting privacy in the real world

Goal: empirically audit real-world privacy costs of (DP) algorithms

- Analogous to the role of cryptanalysis in cryptography

Challenge: auditing requires developing stronger attacks

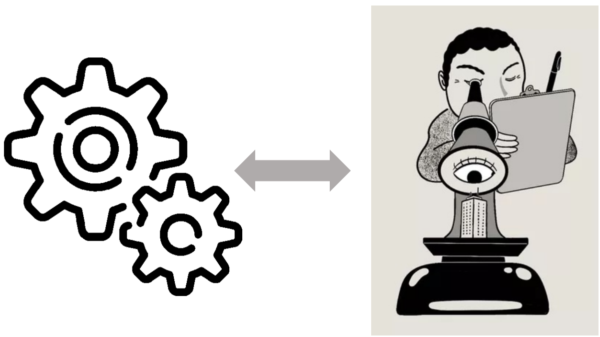
- Existing attacks typically fail even for very large values of ε!



This Talk

1. Example: **auditing DP-SGD** (JUO20)
   a. What is DP-SGD?
   b. Membership inference attacks
   c. Improved MI for DP-SGD
2. Recent work and future directions

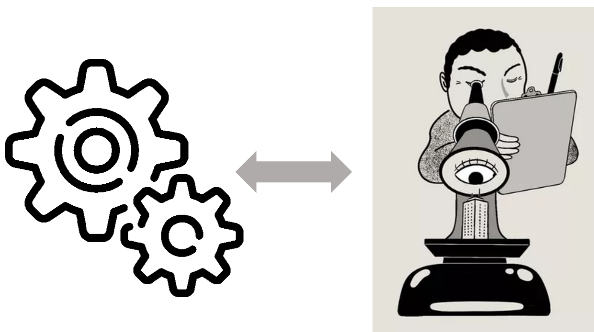
Building on our work



Auditing continual learning

- Most attacks are designed for standalone models
- Modern machine learning pipelines continually update models in response to new data or new tasks
- Can extend MI attacks to audit learning pipelines (JWOUG23)

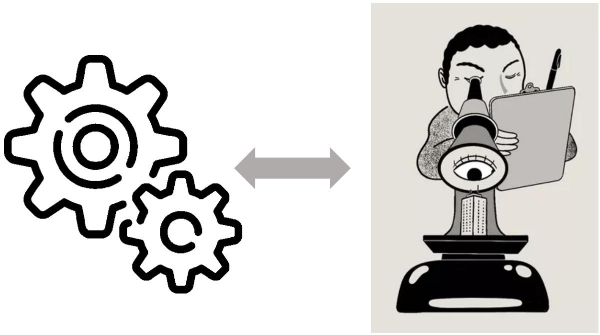
Building on our work



Auditing federated learning

- Many systems for federated learning actually do reveal more than the final output (e.g. some of the iterates $\theta_0, \theta_1, \dots, \theta_T$)
- Can use auditing to explore how different systems threat models lead to different privacy levels (NSTPC21)

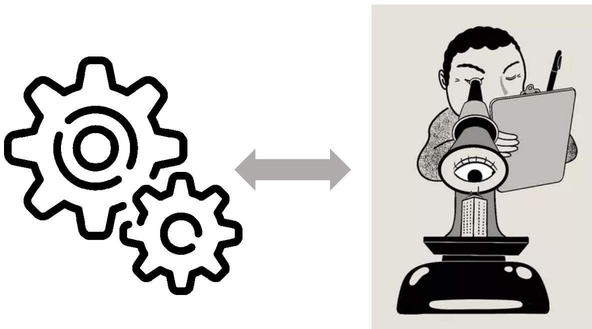
Building on our work



Using auditing to detect bugs

- Not all privacy proofs and implementations are correct
- Auditing methods found a bug in a published paper (TTSSJC22)

Building on our work



Using auditing for algorithm selection

- Some algorithms have tighter analyses than others
- In some case the algorithm with the smallest provable ε is not the one that is most resistant to our attacks (MMPST21)

Auditing (Differentially) Private Algorithms

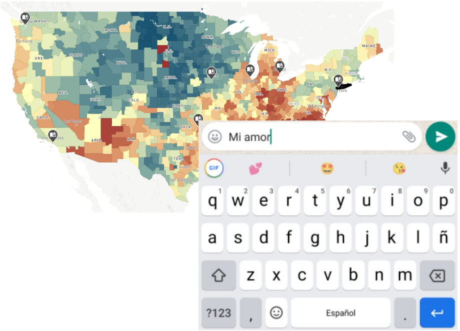
Privacy attacks should play an essential role in testing, quantifying, and interpreting privacy in the real world

Goal: empirically audit real-world privacy costs of (DP) algorithms

- Analogous to the role of cryptanalysis in cryptography

Challenge: auditing requires developing stronger attacks

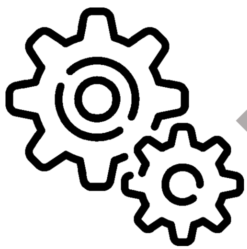
- Existing attacks typically fail even for very large values of ε!

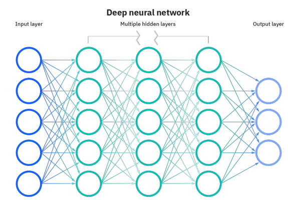


This Talk

1. Example: **auditing DP-SGD** (JUO20)
   a. What is DP-SGD?
   b. Membership inference attacks
   c. Improved MI for DP-SGD
2. Recent work and future directions

Can we audit models instead of algorithms?





How can we audit a model in the wild, without knowing exactly how it was trained?

- What would the algorithm have returned on counterfactual data?
- How can we tell if something is a privacy violation or a lucky guess?
- Easier for language models (C+19, C+21) than predictive models

Can we avoid Goodhart's Law?

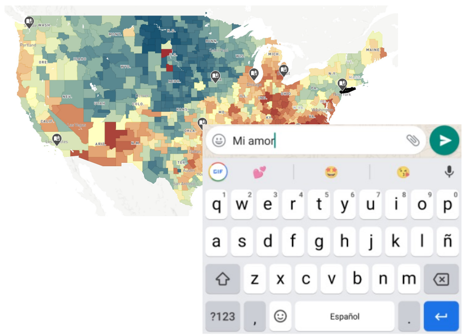
When a measure becomes a target, it ceases to be a good measure

Goal: empirically audit real-world privacy costs of (DP) algorithms

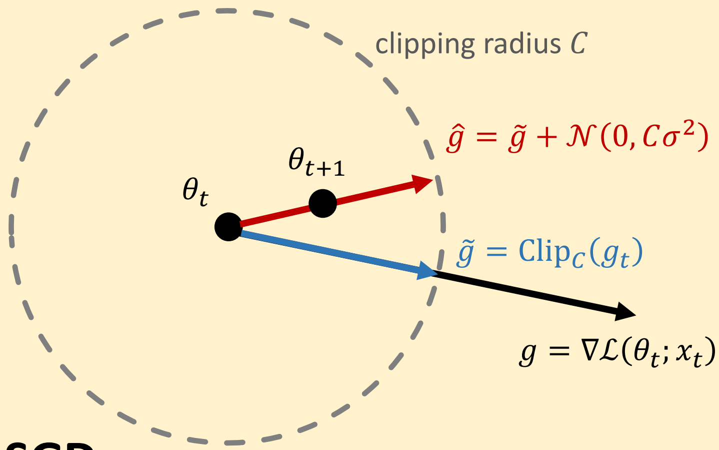
- Analogous to the role of cryptanalysis in cryptography

Challenge: auditing requires developing stronger attacks

- Attacks need to be strong even once they become a target



Auditing: from practice to theory?



DP-SGD

Data: $x_1, \dots, x_n$

Hyperparameters:

- Clipping norm C
- Noise multiplier σ
- Iteration count T
- Initial model θ_0
- Learning rate η
- Loss $\mathcal{L}$

Algorithm:

For $t = 1,2, \dots, T$:

- $x_t \leftarrow$ random example
- $g \leftarrow \nabla\mathcal{L}_x(\theta_t)$
- $\tilde{g} \leftarrow \mathrm{Clip}_C(g)$
- $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \left(\tilde{g} + \mathcal{N}(0, C^2\sigma^2\mathbb{I})\right)$

Output: $\theta_0, \theta_1, \dots, \theta_T$

Can we use auditing methods to inform the way we design and analyze private algorithms?

- Can inform the design of novel algorithms
- Can inform and validate relaxed privacy models

Auditing (Differentially) Private Algorithms

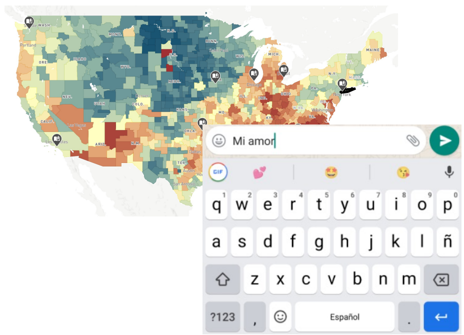
Privacy attacks should play an essential role in
testing, quantifying, and interpreting privacy in the real world

Goal: empirically audit real-world privacy costs of (DP) algorithms

- Analogous to the role of cryptanalysis in cryptography

Challenge: auditing requires developing stronger attacks

- Existing attacks typically fail even for very large values of ε!



Thank You!