

Differential Privacy Algorithms for Machine Learning: DPSGD & PATE

Franziska Boenisch and Adam Dziedzic
Course on Trustworthy Machine Learning
April 17th, 2024

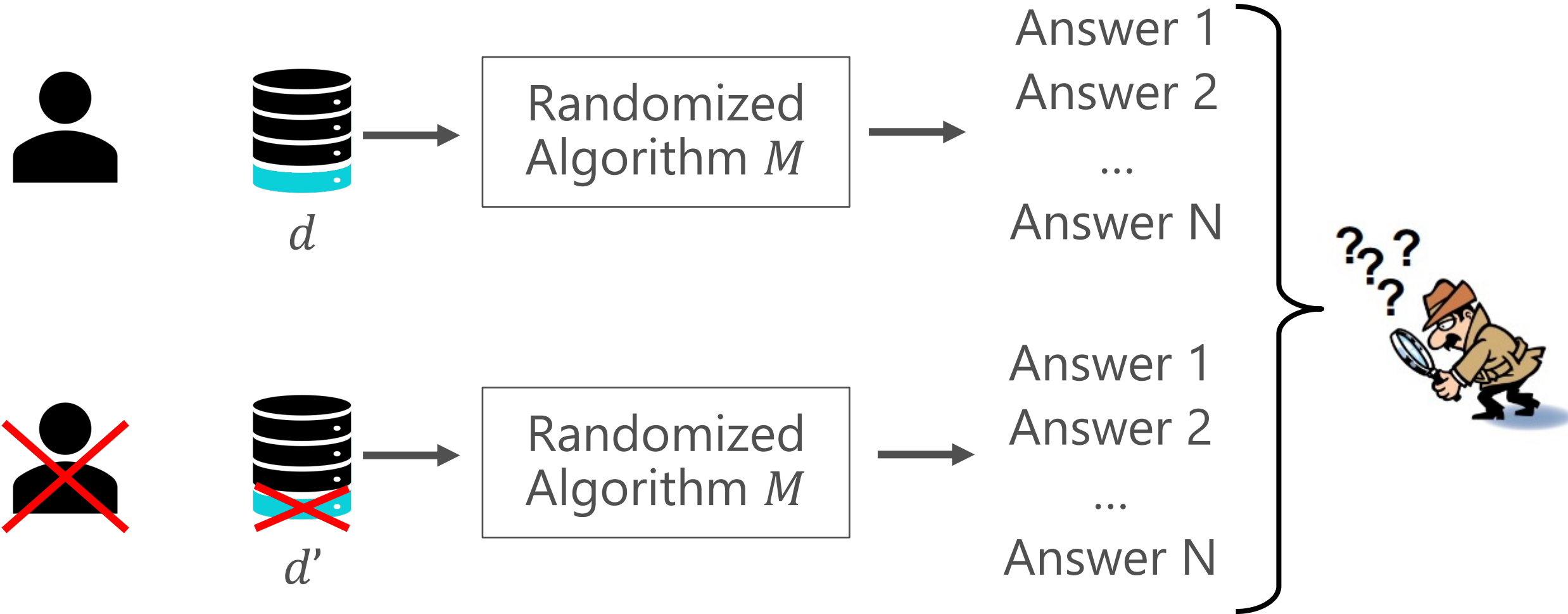


CISPA

HELMHOLTZ CENTER FOR
INFORMATION SECURITY



Recap: Intuition behind Differential Privacy



$$\Pr[M(d) \in S] \leq e^\epsilon \Pr[M(d') \in S]$$

Recap: Intuition behind Differential Privacy

Intuition: Model produces indistinguishable outputs on any pair of training datasets d and d' that differ only by a single data point.

How close models' weights or predictions should be?

Probability of the closeness violation

$$\Pr[\mathbf{M}(d) \in S] \leq e^{\epsilon} \Pr[\mathbf{M}(d') \in S] + \delta$$

Randomized Mechanism

S - possible outputs

Intuition behind Privacy for ML Models

1. **Simple Baseline:** If your ML model has never seen your data, it cannot possibly violate your privacy. This obtains perfect privacy with $\epsilon = 0$.
2. **Privacy Concern:** if your private data is added to the training set of the ML model and the model changed *significantly* compared to the Simple Baseline (model trained without your private data), you can be concerned about your privacy.
3. **How to Measure Privacy Leakage:**
 - We **cannot simply measure how different the weights are** between two different models to measure privacy leakage. Any change to the training process of the model, will potentially significantly change its parameters. Permuting the training data, rerandomizing initial parameters, or running another task on the same GPU will produce a different model with potentially very different weights.
 - If we expect that **addition of private data would not change the model** at all, then the data used for training would be useless.

Measure Privacy Leakage from ML models

1. DP considers the probabilities of observing models' weights.

$$\Pr[\mathbf{M}(d) \in S] \leq e^{\epsilon} \Pr[\mathbf{M}(d') \in S] + \delta$$

2. DP guarantees that the models' weights will not change by more than a specific predefined amount.

Differential Privacy Framework for ML

1. Mechanism: algorithmic techniques for learning. Ensure that:
- (a) the impact of individual data point on an output is limited &
 - (b) it cannot be easily distinguished from the impact incurred by other data points.

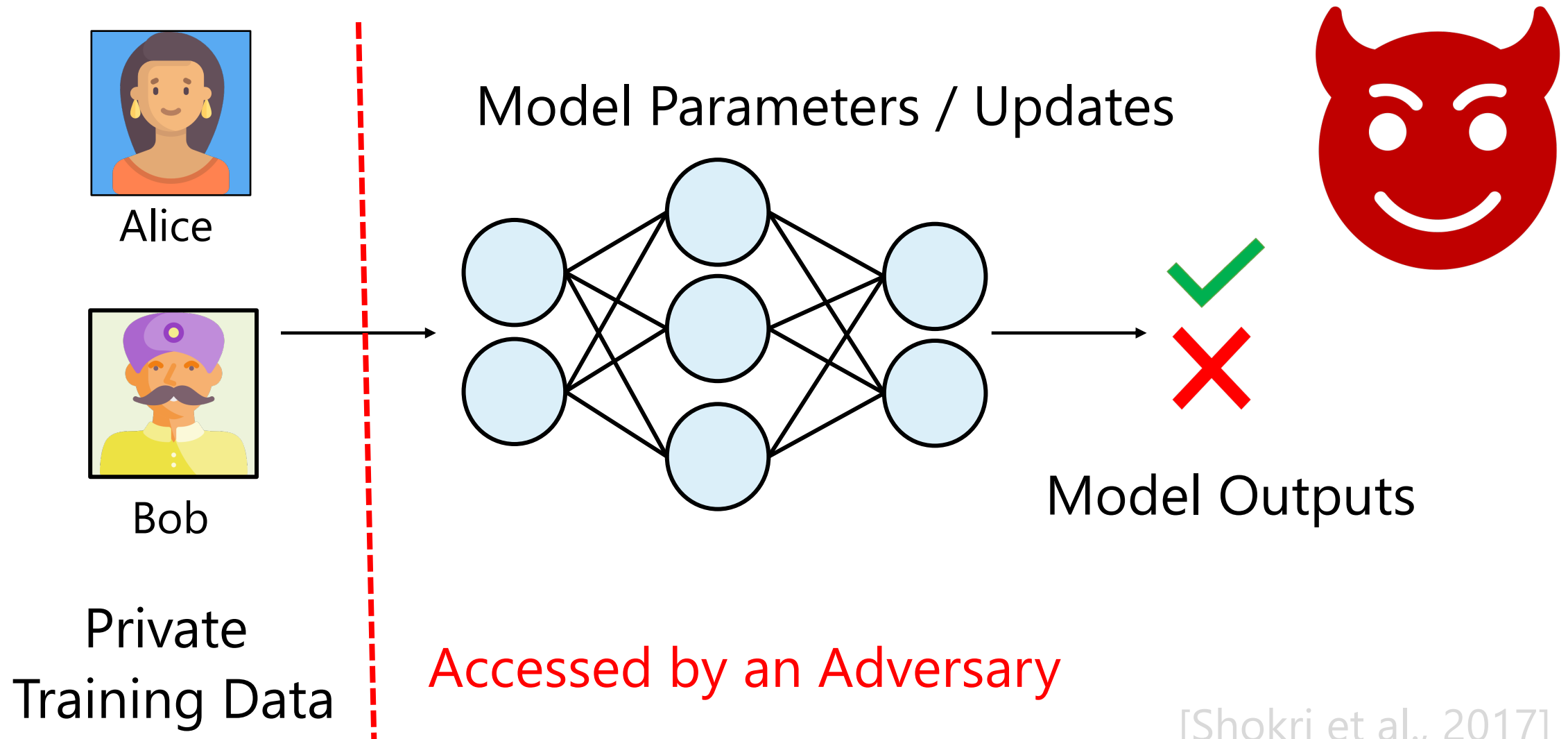
Preview:

- (a) Clip gradient $\bar{g}_t(x_i) \leftarrow g_t(x_i) \cdot \min(1, \frac{C}{\|g_t(x_i)\|_2})$
- (b) Add noise $\tilde{g}_t \leftarrow \bar{g}_t(x_i) + N(0, \sigma^2 C^2 I)$

2. Privacy Accounting: analysis of privacy costs. Account how much privacy is lost due to the training of a model or answering queries.

DP for ML Protects Against Strong Attackers

Goal: Disclose whether a given data point was used to train the model.



Attractive Properties of DP for ML

Composability: enables modular design of DP mechanisms where if all the components of a mechanism are differentially private, then so is their composition.

Simple Sequential Composition: If $\mathbf{M}_1(x)$ fulfills ϵ_1, δ_1 -DP and $\mathbf{M}_2(x)$ fulfills ϵ_2, δ_2 -DP, then $\mathbf{G}(x) = (\mathbf{M}_1(x), \mathbf{M}_2(x))$ fulfills $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ -DP.

Example: *In the training of an ML model, we apply a DP mechanism to each step of the training when we update the model parameters according to the computed gradient.*

Attractive Properties of DP for ML

Group privacy: implies graceful degradation of privacy guarantees if datasets contain correlated inputs, such as the ones contributed by the same individual.

Let $M: X^n \rightarrow Y$ be an (ϵ, δ) -differentially private algorithm. Suppose d and d' are two datasets which differ in exactly k positions $\|d - d'\|_1 \leq k$. Then for all $S \in Y$, we have:

$$\Pr[M(d) \in S] \leq e^{k\epsilon} \Pr[M(d') \in S] + k e^{(k-1)\epsilon} \delta$$

Robustness to auxiliary information: means that privacy guarantees are not affected by any side information available to the adversary.

Post-processing: safe to compute on output from DP mechanism.

From SGD to Differentially Private (DP)-SGD

Input: Model params θ , Loss function \mathcal{L} , data $\{x_1, x_2, \dots, x_N\}$

Learning rate η

For $t \in [T]$ do:

 Take a random sample x_i

 Compute gradient $g_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$

 Descent $\theta_{t+1} \leftarrow \theta_t - \eta \tilde{g}_t$ // step in the negative direction

Output: θ_T

DPSGD: Differentially Private SGD

Input: Model params θ , Loss function \mathcal{L} , data $\{x_1, x_2, \dots, x_N\}$

Learning rate η , noise scale σ , gradient norm bound C

For $t \in [T]$ do:

Take a random sample x_i

Compute gradient $g_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$

Clip gradient $\bar{g}_t(x_i) \leftarrow g_t(x_i) \cdot \min(1, \frac{C}{\|g_t(x_i)\|_2})$

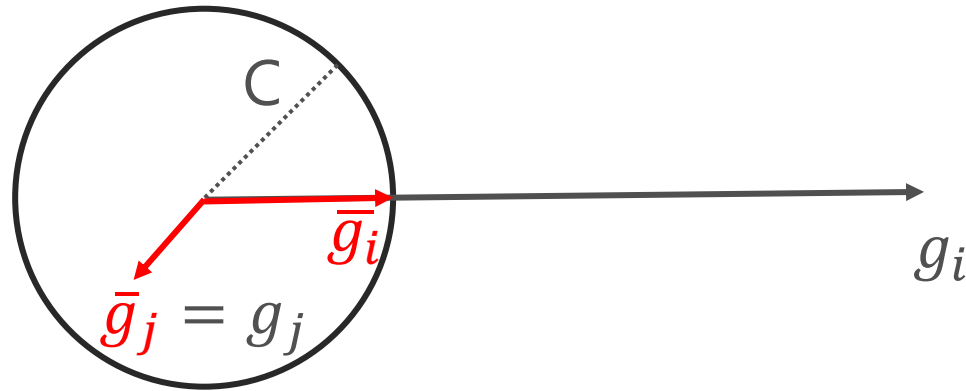
Add noise $\tilde{g}_t \leftarrow \bar{g}_t(x_i) + N(0, \sigma^2 C^2 I)$

Descent $\theta_{t+1} \leftarrow \theta_t - \eta \tilde{g}_t$ // step in the negative direction

Output: θ_T and privacy cost (ϵ, δ)

Clip Gradients in Differentially Private SGD

Goal: Bound the influence of each individual example x_i on the computed gradient $g_t(x_i)$: $\bar{g}_t(x_i) \leftarrow g_t(x_i) \cdot \min(1, \frac{C}{\|g_t(x_i)\|_2})$



The gradient is clipped in the ℓ_2 norm, where the gradient vector $g_t(x_i)$ is:

1. Scaled down to norm C if $\|g_t(x_i)\|_2 > C$: $\bar{g}_t(x_i) = \frac{g_t(x_i)}{\|g_t(x_i)\|_2} C$
2. Preserved if $\|g_t(x_j)\|_2 \leq C$, namely: $\bar{g}_t(x_j) = g_t(x_j)$

DPSGD: Differentially Private SGD

Input: Model params θ , Loss function \mathcal{L} , data $\{x_1, x_2, \dots, x_N\}$
Learning rate η , noise scale σ , gradient norm bound C

For $t \in [T]$ do:

Take a random Lot L_i , with sampling probability L/N

Compute gradient for each $i \in L_i$: $g_t(x_i) \leftarrow \nabla_{\theta_t} L(\theta_t, x_i)$

Clip gradient $\bar{g}_t(x_i) \leftarrow g_t(x_i) \cdot \min(1, \frac{C}{\|g_t(x_i)\|_2})$

Add noise $\tilde{g}_t \leftarrow \frac{1}{L} (\sum_i \bar{g}_t(x_i) + N(0, \sigma^2 C^2 I))$

Descent $\theta_{t+1} \leftarrow \theta_t - \eta \tilde{g}_t$ // step in the negative direction

Output: θ_T and privacy cost (ϵ, δ)

Privacy Accounting for DPSGD: subsampling

DPSGD uses **Poisson sampling** where every example x_i is selected with probability $q = \frac{L}{N}$.

The **expected size of Lot is L** since we sample from N private training data points $\{x_1, x_2, \dots, x_N\}$ in each iteration. However, it is only in expectation: it can select fewer or more than L points.

This leverages the "**privacy amplification by subsampling**" **principle**: each iteration is $(O(q\varepsilon), q\delta)$ -differentially private with respect to the full dataset where $q = L/N$ is the sampling ratio per lot and $\varepsilon \leq 1$.

From Simple to Advanced Composition

Simple Sequential Composition: If $\mathbf{M}_1(x)$ fulfills $(\varepsilon_1, \delta_1)$ -DP and $\mathbf{M}_2(x)$ fulfills $(\varepsilon_2, \delta_2)$ -DP, then $\mathbf{M}(x) = (\mathbf{M}_1(x), \mathbf{M}_2(x))$ fulfills $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$ -DP.

Basic Composition (Extension of Simple Sequential Composition): Suppose $M = (M_1, \dots, M_T)$ is a sequence of algorithms where M_i is $(\varepsilon_i, \delta_i)$ -DP and M_i 's are potentially chosen sequentially and adaptively (the selection of the next mechanism can depend on the previous one). Then M is $(\sum_{i=1}^T \varepsilon_i, \sum_{i=1}^T \delta_i)$ -DP.

Note: If all $\varepsilon_i = \varepsilon$ and $\delta_i = \delta$, then this amounts to $(T\varepsilon, T\delta)$ -DP.

Advanced Composition: For all $\varepsilon, \delta, \delta'$ $M = (M_1, \dots, M_T)$ is a sequence of (ε, δ) -DP algorithms, potentially chosen sequentially and adaptively (the selection of the next mechanism can depend on the previous one). Then M is $(\tilde{\varepsilon}, \tilde{\delta})$ -DP, where

$$\tilde{\varepsilon} = \varepsilon \sqrt{2T \log(1/\delta')} + T\varepsilon \frac{e^\varepsilon - 1}{e^\varepsilon + 1} \text{ and } \tilde{\delta} = T\delta + \delta'.$$

Note: M roughly amounts to $(\sqrt{T} \varepsilon, T\delta)$ -DP. This saves factor \sqrt{T} compared to the basic composition. We assume ε is very small and $\delta' \approx \delta$.

DPSGD Privacy Loss: Moments Accountant

3 components:

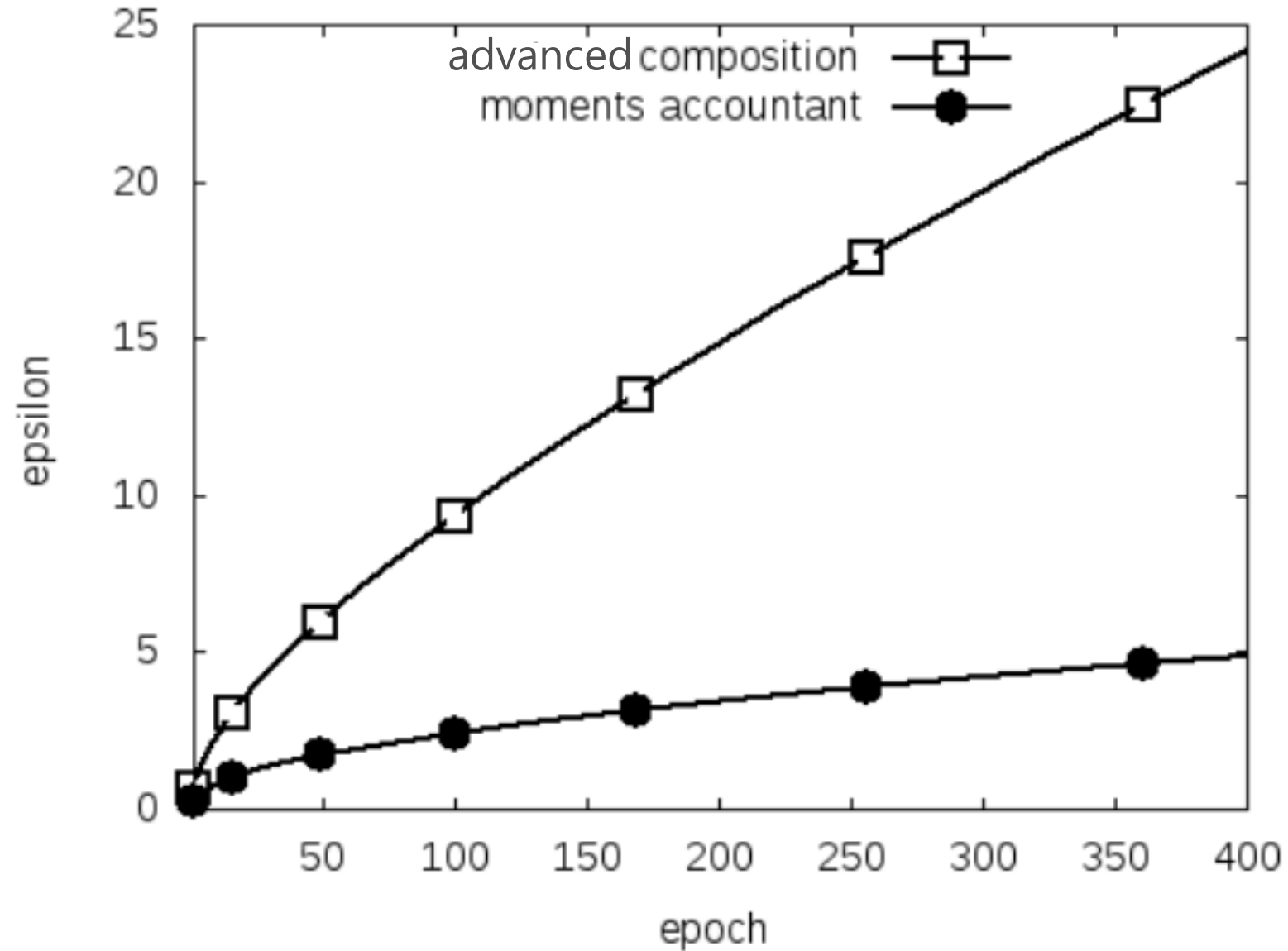
1. Gaussian Mechanism: $\tilde{g}_t \leftarrow \bar{g}_t(x_i) + N(0, \sigma^2 C^2 I)$ is (ϵ, δ) -DP.
2. Privacy amplification: $(O(q\epsilon), q\delta)$ -DP.
3. Advanced composition (\approx): $(O(\epsilon\sqrt{T \log(1/\delta)}), T\delta)$ -DP.

Combining the components for the training of an ML model:

Initial: $(O(q\epsilon\sqrt{T \log(1/\delta)}), qT\delta)$ -DP (we assume T dominates over q)

Moments Accountant: $(O(q\epsilon\sqrt{T}), \delta)$ -DP.

Moments Accountant vs Advanced Composition



Privacy Loss Random Variable

$\Pr[\mathbf{M}(d) \in S] \leq e^{\epsilon} \Pr[\mathbf{M}(d') \in S]$ (S: set of outputs)

$\Pr[\mathbf{M}(d) = z] \leq e^{\epsilon} \Pr[\mathbf{M}(d') = z]$ (z: an output value)

$$\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \leq e^{\epsilon}$$
$$\ln \left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right) \leq \epsilon$$

The privacy random variable $L_{\mathbf{M}(d) || \mathbf{M}(d')}$:

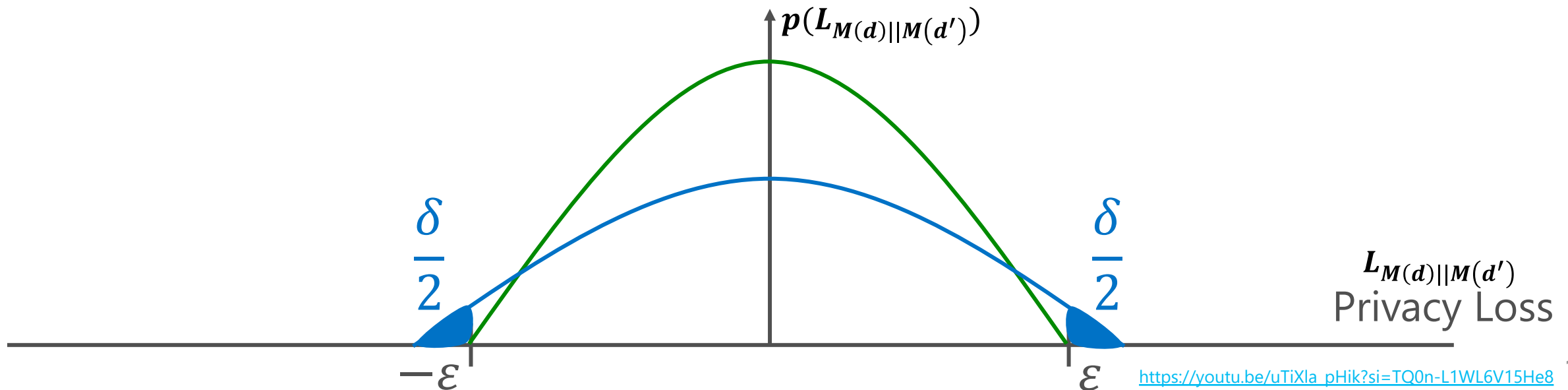
$$L_{\mathbf{M}(d) || \mathbf{M}(d')} = \ln \left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right)$$

Privacy Loss Random Variable

$$(\forall d, d'): \mathbf{L}_{M(d) || M(d')} = \ln \left(\frac{\Pr[M(d)=z]}{\Pr[M(d')=z]} \right)$$

ϵ -DP (pure): $|\mathbf{L}_{M(d) || M(d')}| \leq \epsilon$ with probability 1.

(ϵ, δ) -DP (approx.): $|\mathbf{L}_{M(d) || M(d')}| \leq \epsilon$ with prob. $1 - \delta$.

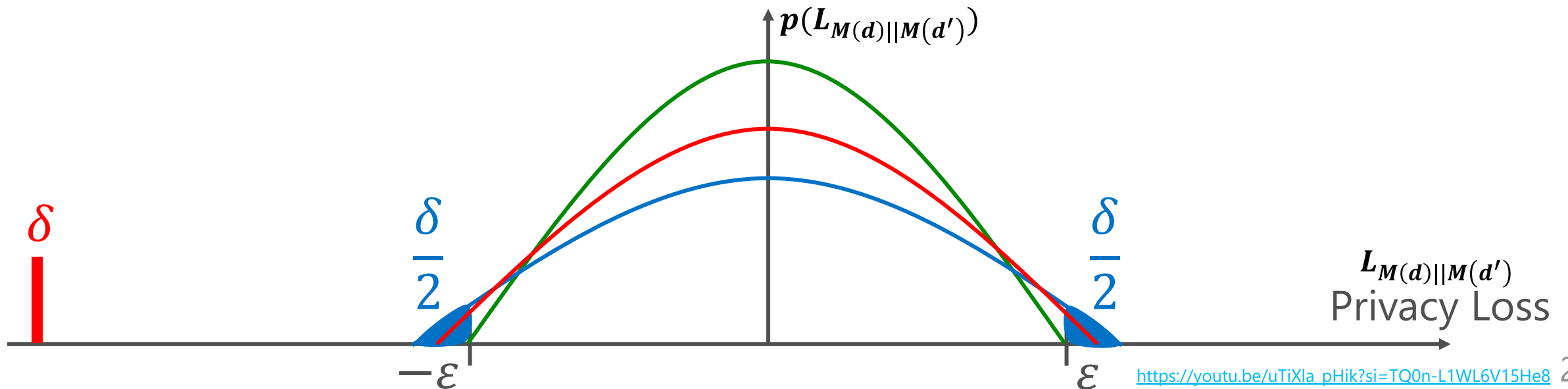


Privacy Loss Random Variable

$$(\forall d, d'): \mathbf{L}_{M(d) || M(d')} = \ln \left(\frac{\Pr[M(d)=z]}{\Pr[M(d')=z]} \right)$$

ϵ -DP (pure): $|\mathbf{L}_{M(d) || M(d')}| \leq \epsilon$ with probability 1

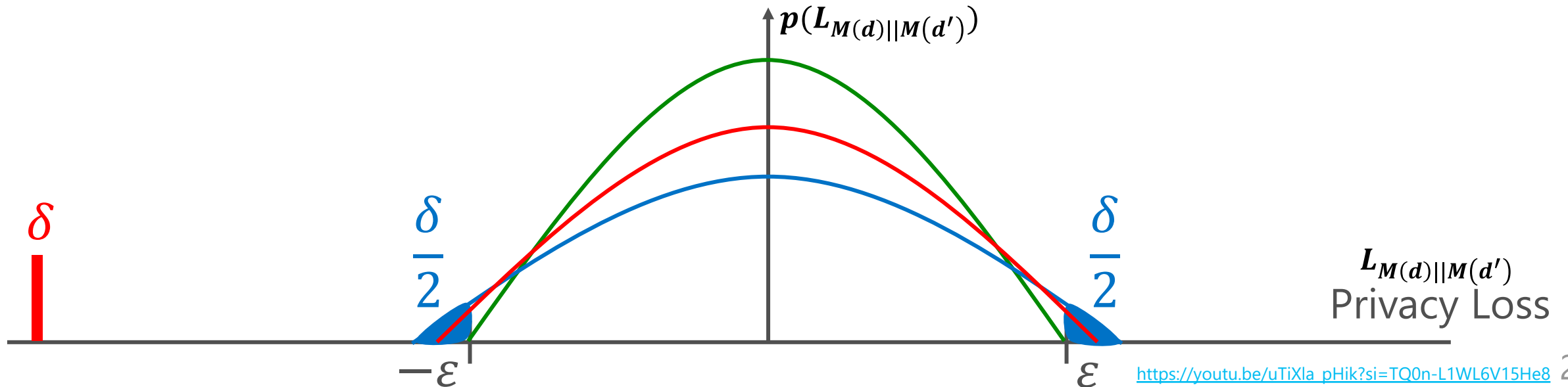
(ϵ, δ) -DP (approx.): $|\mathbf{L}_{M(d) || M(d')}| \leq \epsilon$ with prob. $1 - \delta$



Moment Accountant

$$(\forall d, d'): L_{M(d) || M(d')} = \ln \left(\frac{\Pr[M(d)=z]}{\Pr[M(d')=z]} \right)$$

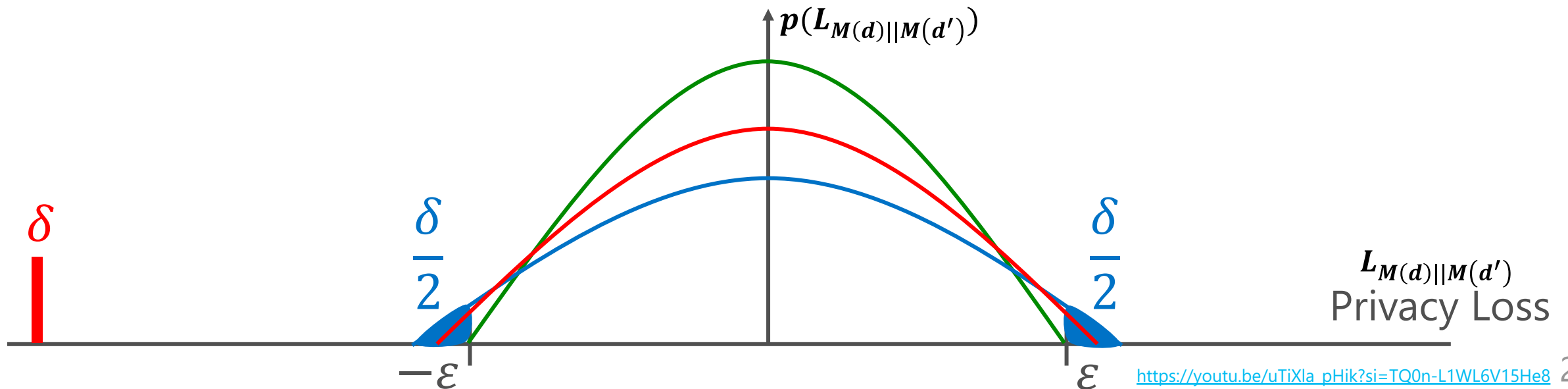
Moments of a function are quantitative measures related to the shape of the function's graph.



Moments Accountant

$$(\forall d, d'): \mathbf{L}_{\mathbf{M}(d) || \mathbf{M}(d')} = \ln \left(\frac{\Pr[\mathbf{M}(d)=z]}{\Pr[\mathbf{M}(d')=z]} \right)$$

$$\ln E_{z \sim \mathbf{M}(d)} \left[\left(\frac{\Pr[\mathbf{M}(d)=z]}{\Pr[\mathbf{M}(d')=z]} \right)^\lambda \right] \leq \gamma$$



Moments Accountant

$$\ln E_{z \sim M(d)} \left[\left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right)^\lambda \right] \leq \gamma$$

$$E_{z \sim M(d)} \left[\left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right)^\lambda \right] \leq \exp(\gamma)$$

$$\delta = \Pr_{z \sim M(d)} \left[\ln \left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right) \geq \varepsilon \right]$$

Multiply both sides by λ and take the exponent.

$$\delta = \Pr_{z \sim M(d)} \left[\exp(\lambda \ln \left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right)) \geq \exp(\lambda \varepsilon) \right]$$

Moments Accountant

$$\delta = \Pr_{z \sim M(d)} \left[\exp\left(\ln \left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right)^\lambda \right) \geq \exp(\lambda \epsilon) \right]$$
$$\delta = \Pr_{z \sim M(d)} \left[\left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right)^\lambda \geq \exp(\lambda \epsilon) \right]$$

Markov's Inequality: $P(X \geq a) \leq \frac{E(X)}{a}$, X is a random variable > 0 , $a > 0$

$$\delta \leq \frac{E_{z \sim M(d)} \left[\left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right)^\lambda \right]}{\exp(\lambda \epsilon)} \leq \frac{\exp(\gamma)}{\exp(\lambda \epsilon)} = \exp(\gamma - \lambda \epsilon)$$

Moments Accountant

So, assume that we have the following bound on the random variable:

$$\ln E_{z \sim M(d)} \left[\left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right)^\lambda \right] \leq \gamma$$

It implies, that if we take the given γ and λ , and we select some ϵ then we can compute the corresponding delta:

$$\delta \leq \frac{E_{z \sim M(d)} \left[\left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right)^\lambda \right]}{\exp(\lambda \epsilon)} \leq \frac{\exp(\gamma)}{\exp(\lambda \epsilon)} = \exp(\gamma - \lambda \epsilon)$$

Moments Accountant

$$\ln E_{z \sim M(d)} \left[\left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right)^\lambda \right] \leq \gamma \Rightarrow (\varepsilon, \delta) - DP$$

From Moments Accountant to Rényi DP

$$\ln E_{z \sim M(d)} \left[\left(\frac{\Pr[\mathbf{M}(d) = z]}{\Pr[\mathbf{M}(d') = z]} \right)^\lambda \right] \leq \gamma \Rightarrow (\varepsilon, \delta) - DP$$

Rényi divergence. For two probability distributions P and Q defined over R , the Rényi divergence of order $\alpha > 0$ is:

$$D_\alpha(P||Q) \triangleq \frac{1}{\lambda - 1} \log E_{x \sim P} \left(\frac{P(x)}{Q(x)} \right)^{\lambda - 1}$$

Rényi Differential Privacy (RDP)

Rényi divergence. For two probability distributions P and Q defined over R , the Rényi divergence of order $\alpha > 0$ is:

$$D_\alpha(P||Q) \triangleq \frac{1}{\alpha - 1} \log E_{x \sim P} \left(\frac{P(x)}{Q(x)} \right)^{\alpha-1}$$

(λ, ϵ) – RDP (Rényi DP): A randomized mechanism $M: D \rightarrow R$ is said to have ϵ -Rényi differential privacy of order λ , or (λ, ϵ) -RDP in short, if for any adjacent datasets $d, d' \in D$ it holds that:

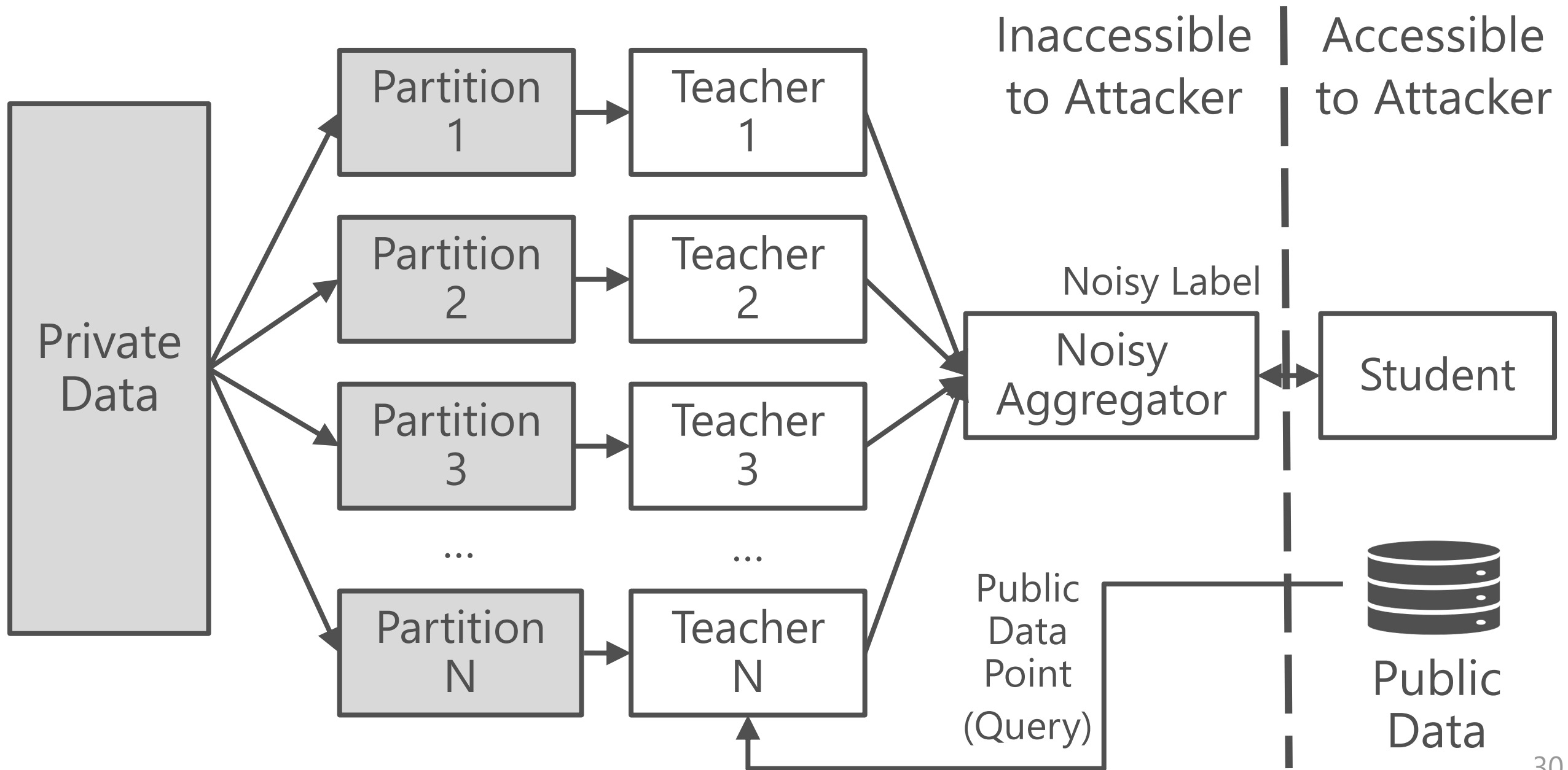
$$D_\lambda(M(d) || M(d')) = \frac{1}{\lambda - 1} \log E_{x \sim M(d)} \left(\frac{\Pr[M(d) = z]}{\Pr[M(d') = z]} \right)^{\lambda-1} \leq \epsilon$$

Properties of Rényi Differential Privacy

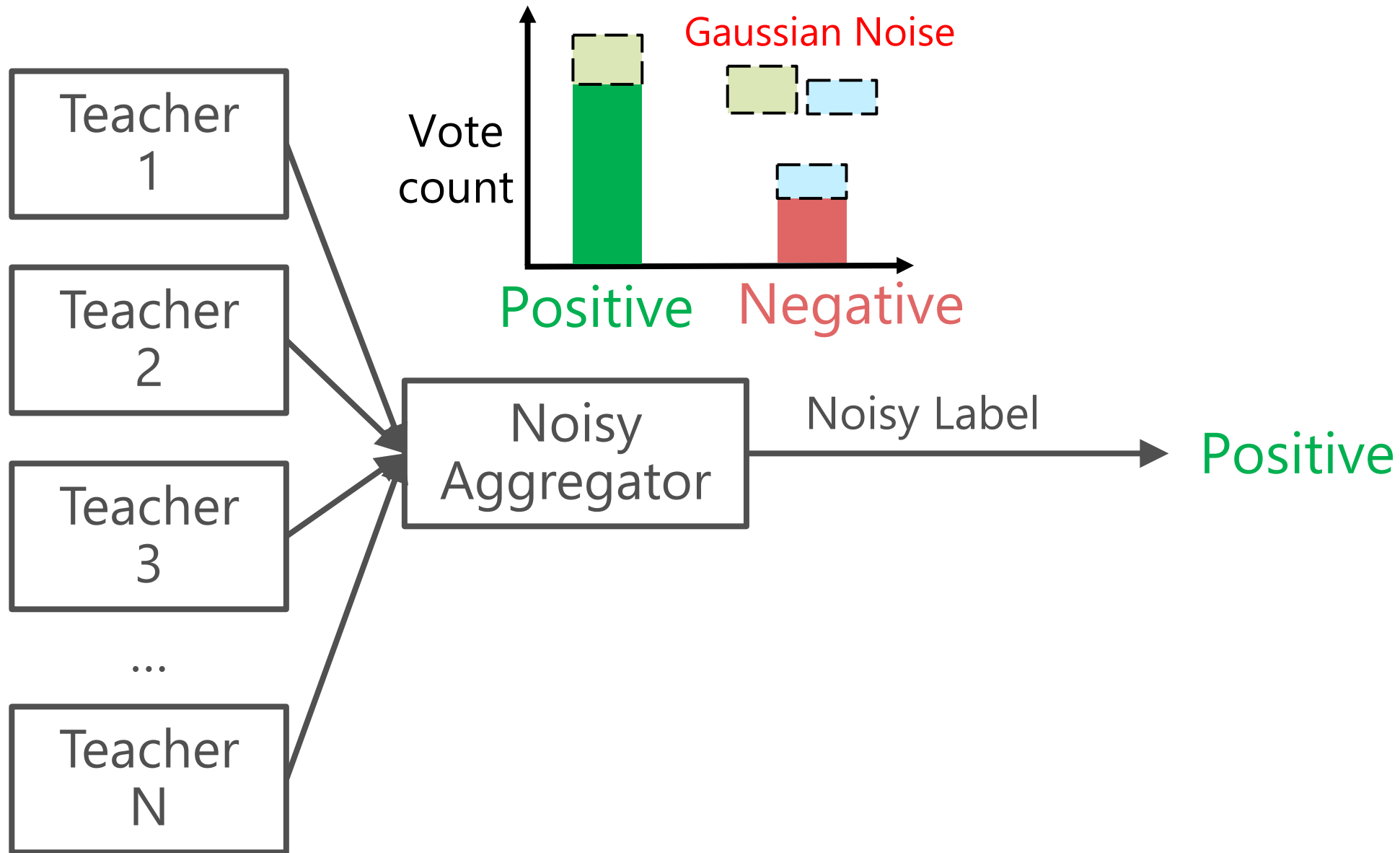
RDP inherits many properties of standard differential privacy.

- 1. From RDP to (ϵ, δ) -DP:** If M is (λ, ϵ) -RDP mechanism, it also satisfies $(\epsilon + \frac{\log \frac{1}{\delta}}{\lambda-1}, \delta)$ -DP for any $0 < \delta < 1$.
- 2. Group privacy:** implies graceful degradation of privacy guarantees if datasets contain correlated inputs, such as the ones contributed by the same individual.
- 3. Robustness to auxiliary information:** means that privacy guarantees are not affected by any side information available to the adversary. No assumption on the adversary's knowledge.
- 4. Post-processing:** safe to compute on output from DP mechanism.
- 5. Sequential composition:** modular construction of differentially private algorithms.

Differential Privacy for ML with PATE



Noisy Aggregator Mechanism in PATE



Gaussian Noisy Max Aggregation Mechanism

GNMAX: For a sample x and classes 1 to m , let $f_j(x) \in [m]$ denote the j -th teacher model's prediction on x and $n_i(x)$ denote the vote count for the i -th class (i.e., $n_i(x) = |\{j: f_j(x) = i\}|$). We define a Gaussian NoisyMax (GNMax) aggregation mechanism as:

$$M_\sigma(x) \triangleq \operatorname{argmax}_i \{ n_i(x) + N(0, \sigma^2) \}$$

where $N(0, \sigma^2)$ is the Gaussian distribution with mean 0 and variance σ^2 .

The aggregator outputs the class with noisy plurality after adding Gaussian noise to each vote count. Plurality more generally refers to the highest number of teacher votes assigned among the classes.

Confident Noisy Gaussian Max Aggregation

Goal: filter out queries for which teachers do not have a sufficiently strong consensus. This filtering enables the teachers to avoid answering expensive queries, in terms of the privacy cost.

Confident Noisy Gaussian Max Aggregation

Goal: filter out queries for which teachers do not have a sufficiently strong consensus. This filtering enables the teachers to avoid answering expensive queries, in terms of the privacy cost.

Algorithm: given a query x , consensus among teachers is first estimated in a privacy-preserving way to then only reveal confident teacher predictions.

Confident Noisy Gaussian Max Aggregation

Goal: filter out queries for which teachers do not have a sufficiently strong consensus. This filtering enables the teachers to avoid answering expensive queries, in terms of the privacy cost.

Algorithm: given a query x , consensus among teachers is first estimated in a privacy-preserving way to then only reveal confident teacher predictions.

Input: input data point x , threshold T , noise parameters σ_1, σ_2 .

If $\max_i \{n_i(x) + N(0, \sigma_1^2)\} \geq T$ **then** // privately check for consensus

Return $\operatorname{argmax}_j \{n_j(x) + N(0, \sigma_2^2)\}$ // run max-of Gaussian

Else

Return \perp // abstain from answering

Privacy Accounting for PATE

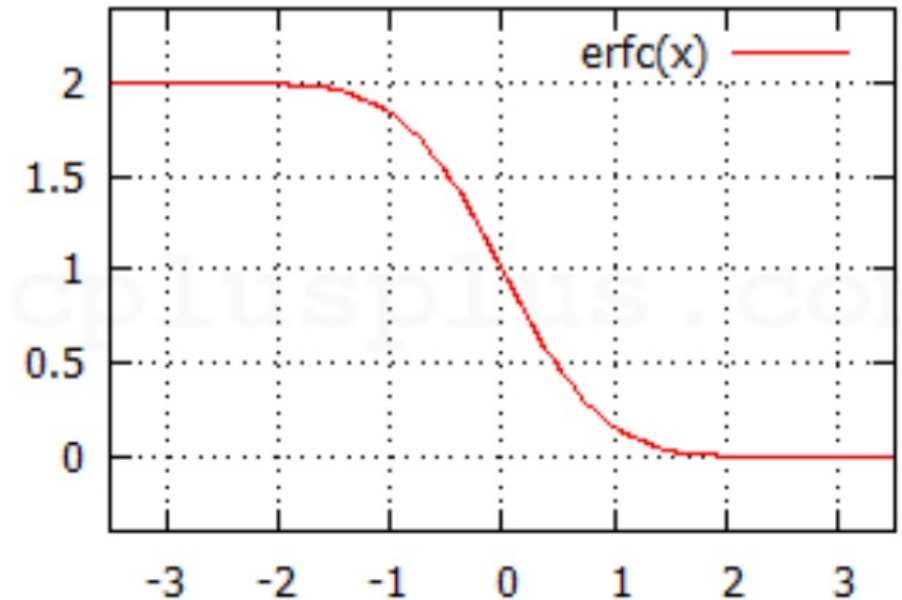
1. **RDP (Rényi-DP)** is used as the privacy analysis framework.
2. In contrast to DPSGD where the privacy analysis is data independent, in PATE we compute the privacy leakage per query answered by teachers, which is **data dependent privacy analysis**.
3. The privacy accounting **bounds the probability** \tilde{q} that the class with maximum number of votes i^* is not selected due to added noise.

Given a dataset D , there exists a likely outcome i^* such that $\Pr[M(D) \neq i^*] \leq \tilde{q}$. The data-dependent Rényi differential privacy for M at D is bounded and approaches 0 as $\tilde{q} \rightarrow 0$.

Privacy Accounting for PATE

Goal: bound the probability \tilde{q} when i^* corresponds to the class with the true plurality of teacher votes.

For any $i^* \in [m]$, where m are number of classes, we have $\Pr[M_\sigma(D) \neq i^*] \leq \frac{1}{2} \sum_{i \neq i^*} \text{erfc}(\frac{n_{i^*} - n_i}{2\sigma})$, where erfc is the complementary error function, and σ is the amount of added Gaussian noise to the histogram.



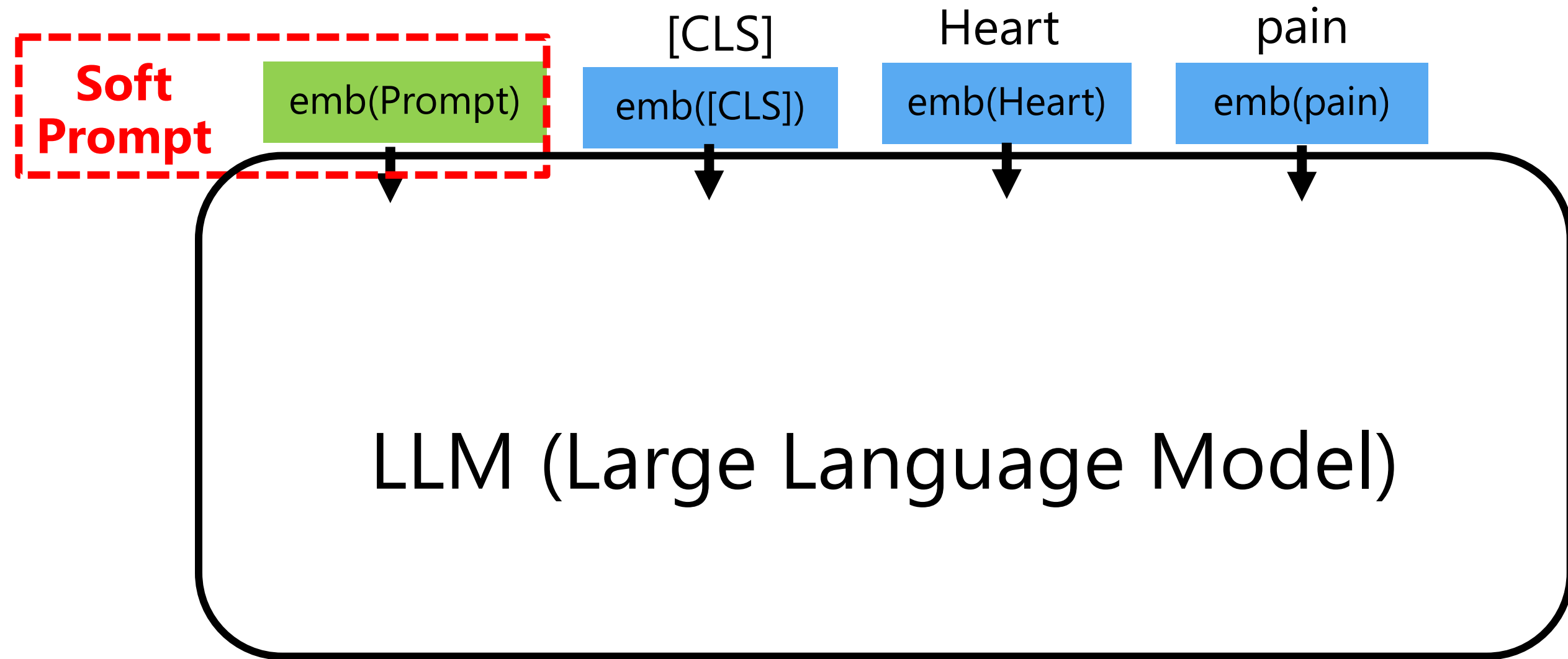
$$\text{erfc}(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

DPSGD vs PATE

PROPERTY	DPSGD	PATE
Privacy Concept	Noisy model parameters	Noisy votes from teachers + student
Privacy Mechanism	Clip gradients and add noise	Train teacher and noisy argmax
Privacy Analysis / Accounting	Data Independent	Data Dependent
Requirements	Integrated into standard model training	Public Data for student
Applications	To any model	If public data exists

Examples: DPSSGD & PATE for LLMs

Soft Prompts: Params Prepended to Input



DPSGD: Differentially Private SGD

Input: Soft prompt params θ , Loss function L ,
Learning rate η , noise scale σ , gradient norm bound C

For $t \in [T]$ do:

Take a random sample x_i

Compute gradient $g_t(x_i) \leftarrow \nabla_{\theta_t} L(\theta_t, x_i)$

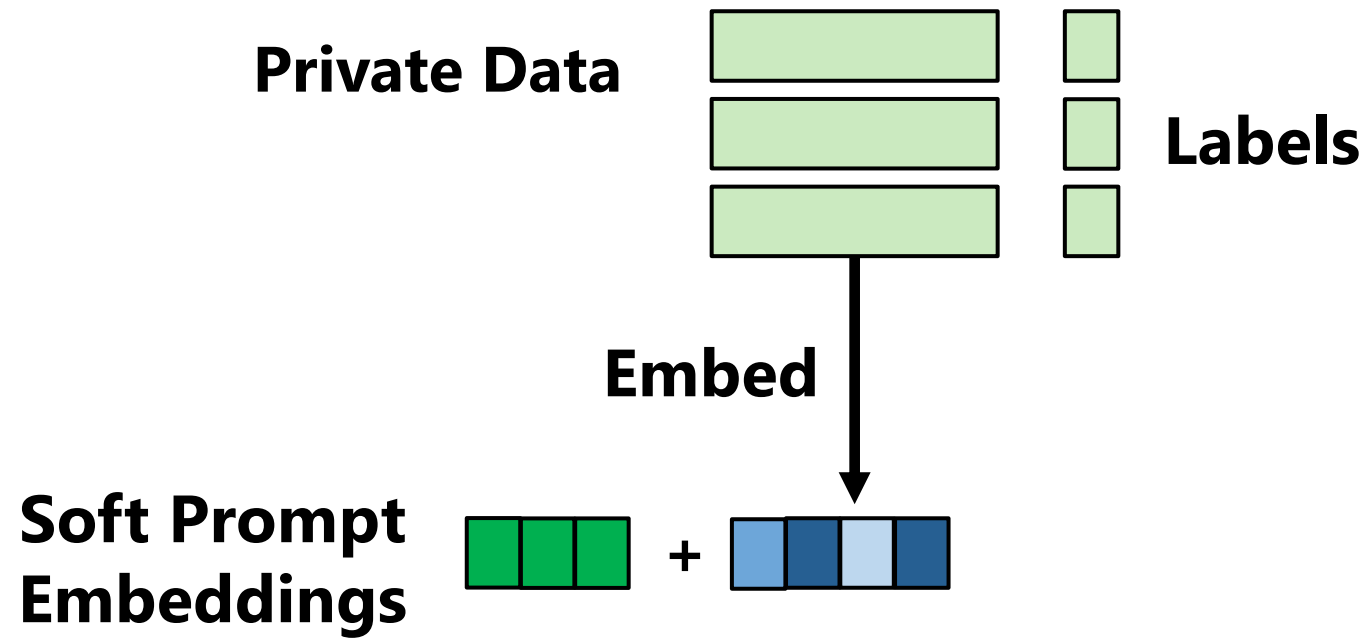
Clip gradient $\bar{g}_t(x_i) \leftarrow g_t(x_i) \cdot \min(1, \frac{C}{\|g_t(x_i)\|_2})$

Add noise $\tilde{g}_t \leftarrow \bar{g}_t(x_i) + N(0, \sigma^2 C^2 I)$

Descent $\theta_{t+1} \leftarrow \theta_t - \eta \tilde{g}_t$

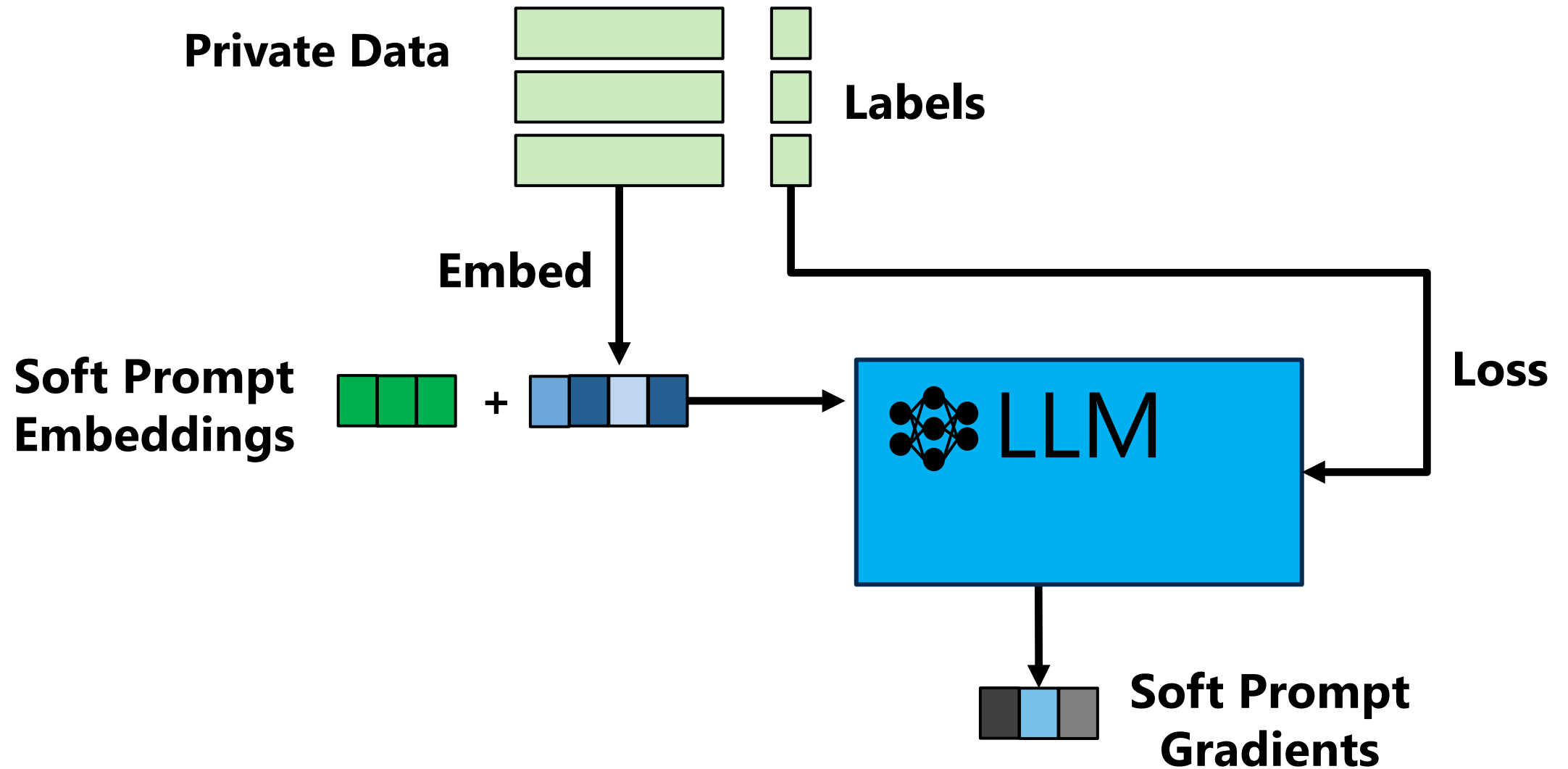
Output: θ_T and privacy cost (ϵ, δ)

Prompt DPSGD: Private Soft Prompt Learning

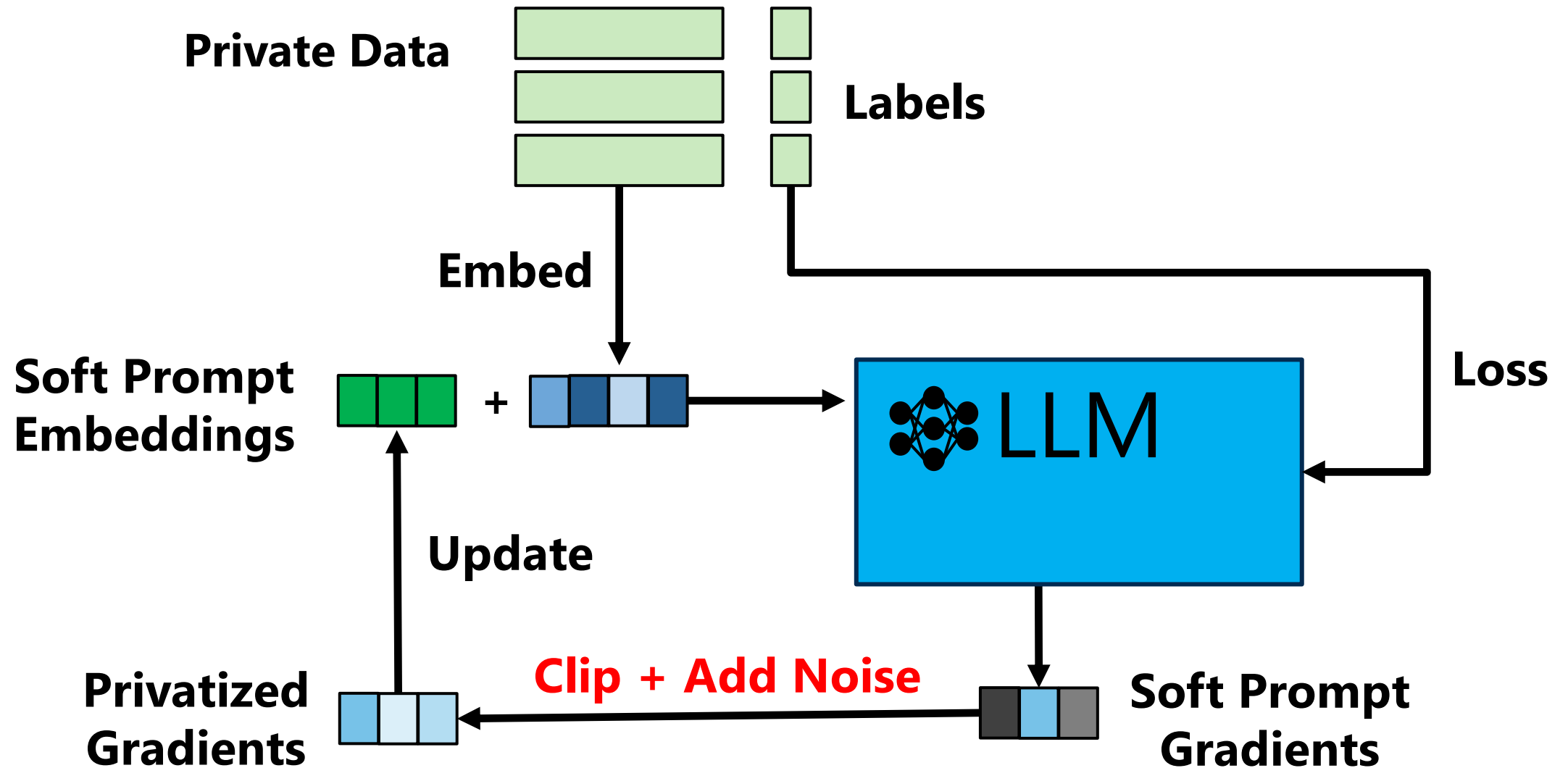


Haonan Duan*, Adam Dziedzic*, Nicolas Papernot, Franziska Boenisch. *"Flocks of Stochastic Parrots: Differentially Private Prompt Learning for Large Language Models"* [NeurIPS 2023].

Prompt DPSGD: Private Soft Prompt Learning



Prompt DPSGD: Private Soft Prompt Learning



In-context Learning with Discrete Prompts

Prompt Template

Instruction: Classify a movie review as positive or negative.

Private Demonstrations:

In: This film is a masterpiece.

Out: Positive ...

No backprop!
Select **Examples**



In-context Learning with Discrete Prompts

Prompt Template

Instruction: Classify a movie review as positive or negative.

Private Demonstrations:

In: This film is a masterpiece.

Out: Positive ...

My input: The movie was great!
Out: ?

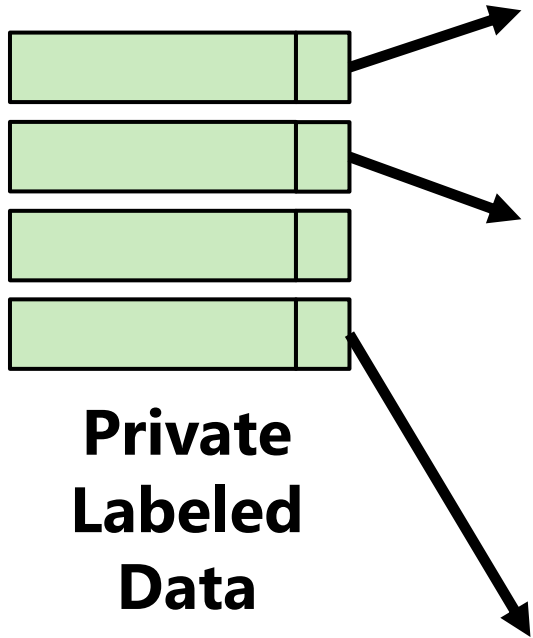
No backprop!
Select **Examples**



Positive

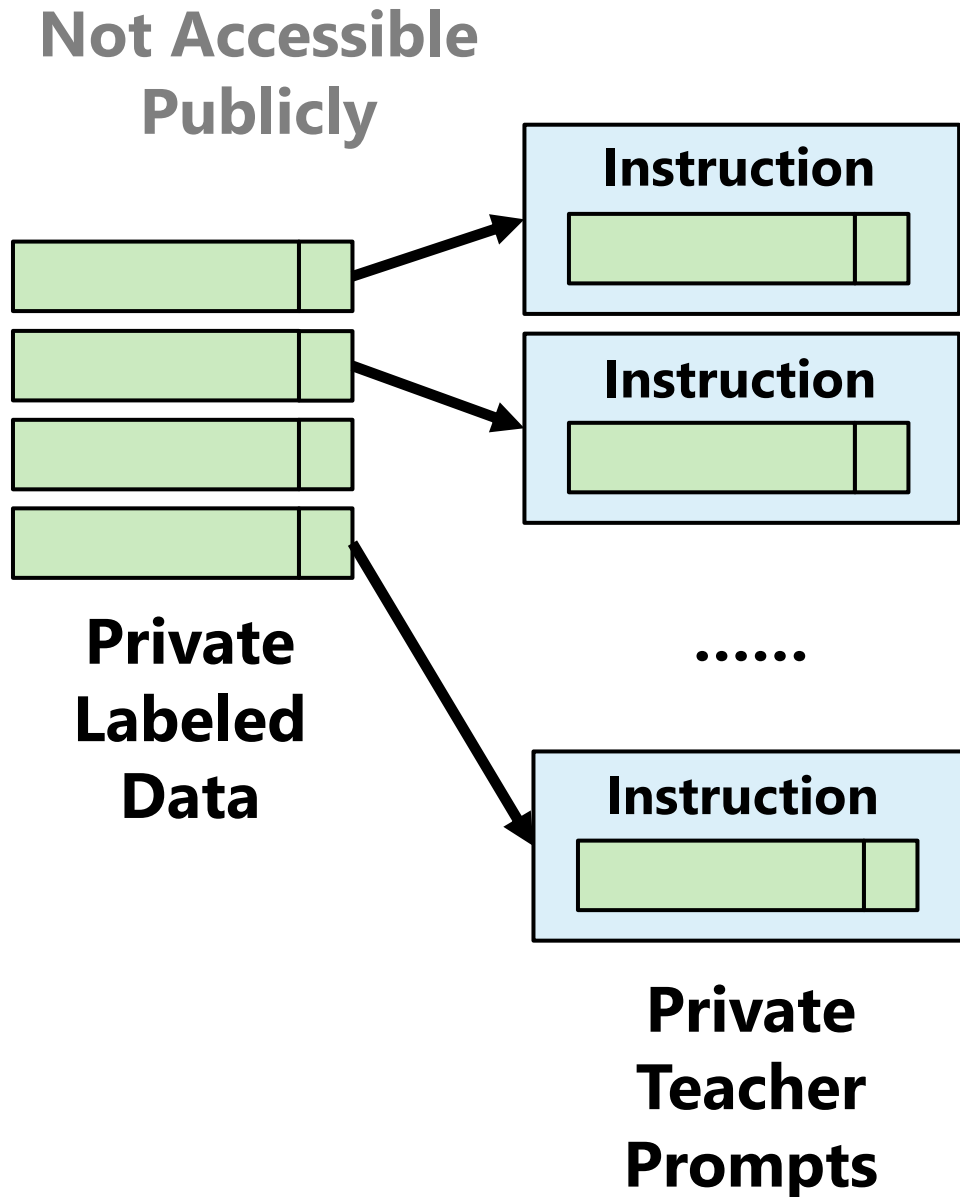
PromptPATE: Private Discrete Prompts

**Not Accessible
Publicly**

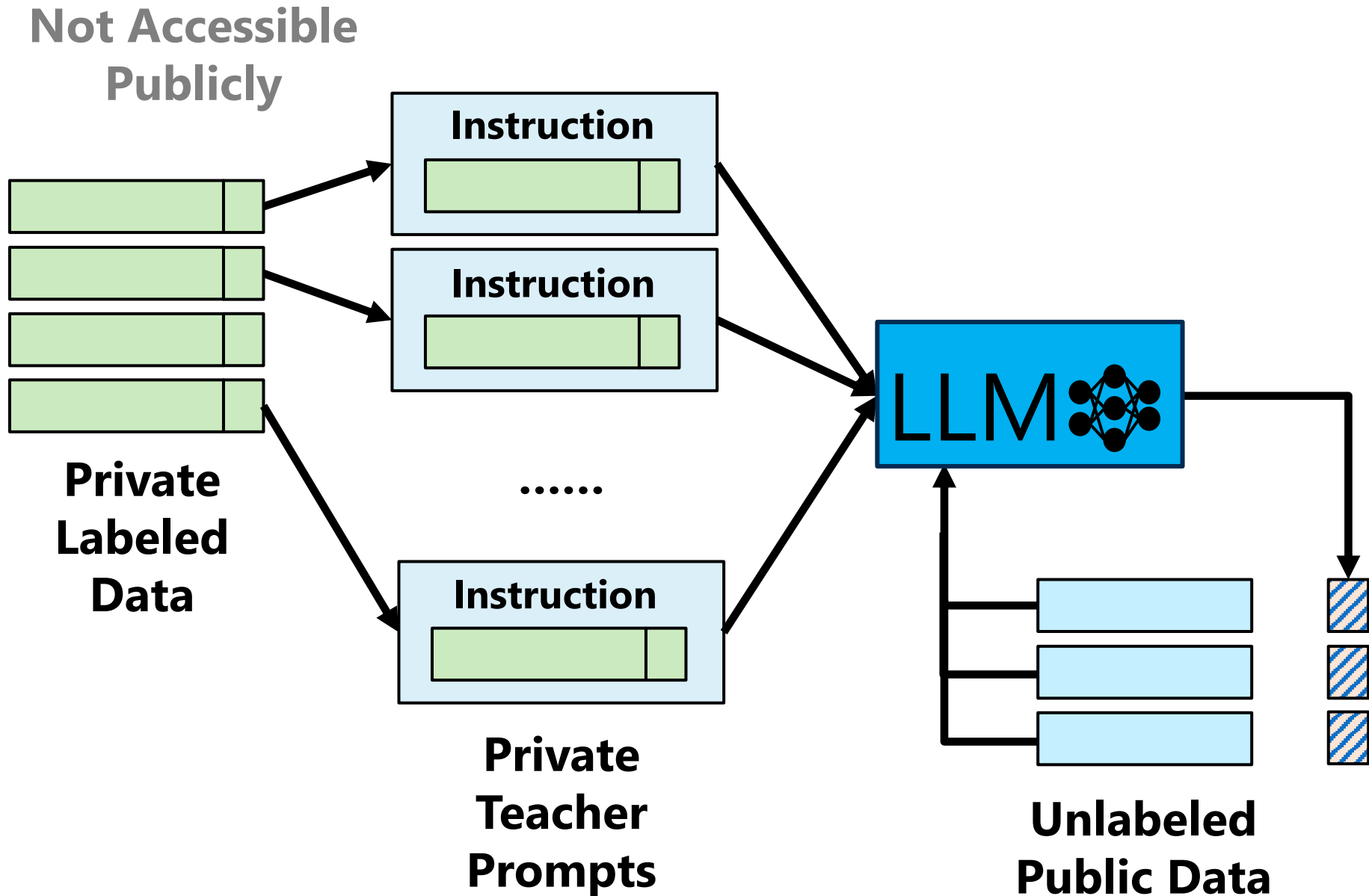


Haonan Duan*, Adam Dziedzic*, Nicolas Papernot, Franziska Boenisch. *"Flocks of Stochastic Parrots: Differentially Private Prompt Learning for Large Language Models"* [NeurIPS 2023].

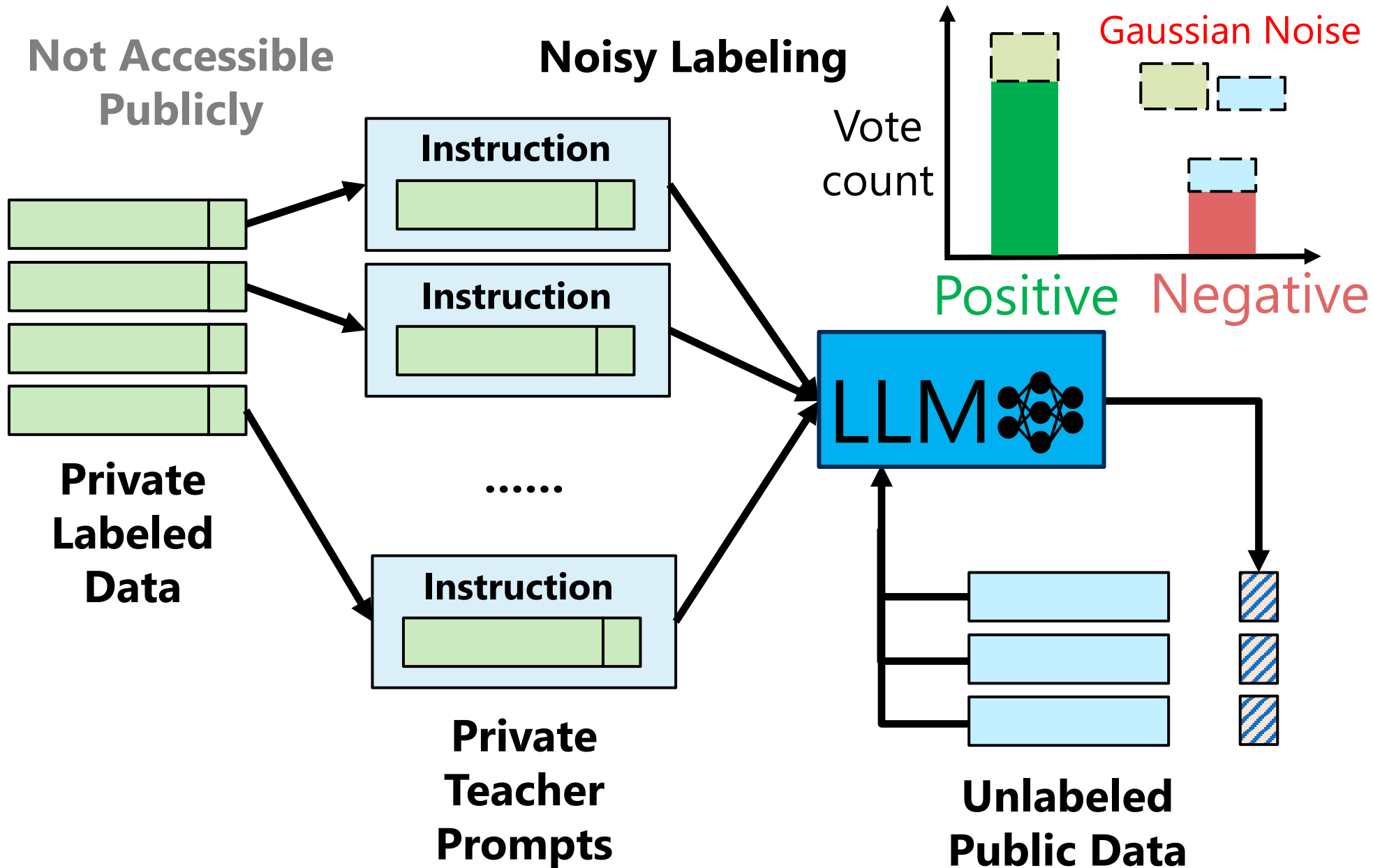
PromptPATE: Private Discrete Prompts



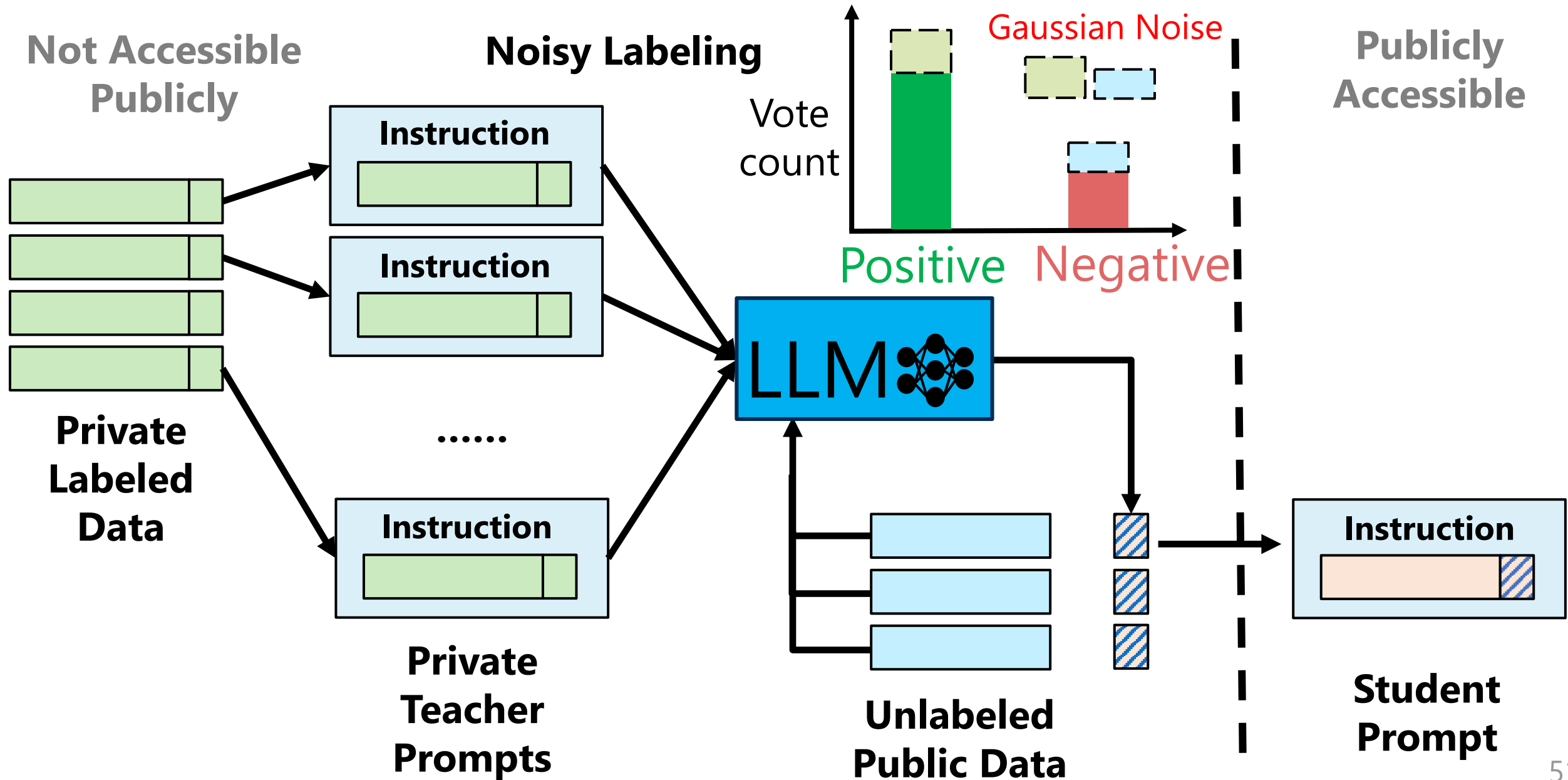
PromptPATE: Private Discrete Prompts



PromptPATE: Private Discrete Prompts



PromptPATE: Private Discrete Prompts



References

1. **"Deep Learning with Differential Privacy"** Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, Li Zhang. <https://arxiv.org/abs/1607.00133>
2. **"Renyi Differential Privacy"** Ilya Mironov. <https://arxiv.org/abs/1702.07476>
3. **"Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data"** Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, Kunal Talwar. <https://arxiv.org/abs/1610.05755>
4. **"Scalable Private Learning with PATE"** Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, Úlfar Erlingsson. <https://arxiv.org/abs/1802.08908>
5. **"Privacy and machine learning: two unexpected allies?"** *Nicolas Papernot and Ian Goodfellow*. <http://www.cleverhans.io/privacy/2018/04/29/privacy-and-machine-learning.html>
6. **"Differential Privacy Series Part 1 | DP-SGD Algorithm Explained"** Davide Testuggine and Ilya Mironov. <https://medium.com/pytorch/differential-privacy-series-part-1-dp-sgd-algorithm-explained-12512c3959a3>

References

7. "Boosting and Differential Privacy" Cynthia Dwork, Guy N. Rothblum, Salil Vadhan. <https://privacytools.seas.harvard.edu/files/privacytools/files/05670947.pdf> (advanced composition theorem)
8. Lectures by Gautam Kamath on Differential Privacy:
https://www.youtube.com/playlist?list=PLmd_zeMNzSvRRNpoEWkVo6QY_6rR3SHj_p

Thank you!

Franziska Boenisch and Adam Dziedzic
boenisch@cispa.de, adam.dziedzic@cispa.de
sprintml.com

Course on Trustworthy Machine Learning

Backup Slides

Differential Privacy in Machine Learning

Goal: produce statistically indistinguishable outputs on any pair of datasets that only differ by any single data point.

Differential Privacy: a randomized mechanism M with domain D and range R satisfies (ϵ, δ) -differential privacy if for any subset $S \subseteq R$ and any adjacent datasets $d, d' \in D$, i.e., $\|d - d'\|_1 \leq 1$, the following inequality holds:

$$\Pr[M(d) \in S] \leq e^\epsilon \Pr[M(d') \in S] + \delta$$

Group Privacy

Let $M: X^n \rightarrow Y$ be an (ϵ, δ) -differentially private algorithm. Suppose d and d' are two datasets which differ in exactly k positions. Then for all $S \in Y$, we have:

$$\Pr[M(d) \in S] \leq e^{k\epsilon} \Pr[M(d') \in S] + k e^{(k-1)\epsilon} \delta$$