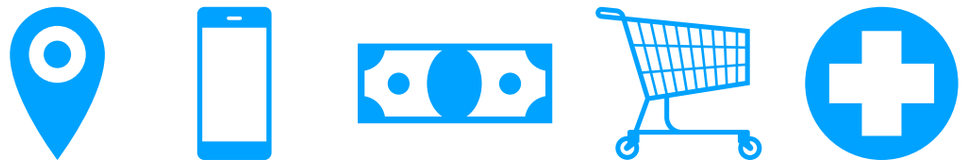
Differential Privacy for Machine Learning

Anastasia Pustozerova
PhD student at TU Wien
Researcher at SBA Research

Vienna Deep Learning Meetup
17 Jan 2024

Data Privacy

- Machine Learning requires a lot of (private) data to train effective models



Data Privacy

- Machine Learning requires a lot of (private) data to train effective models

- Individuals might not be willing to share their private data

Data Privacy

- Machine Learning requires a lot of (private) data to train effective models

- Individuals might not be willing to share their private data

- Data collectors have to comply with regulations and protect individuals' privacy

How to do analysis (machine learning) on sensitive data while keeping it private?

Privacy-preserving Data Analysis and Machine Learning

- Anonymisation Techniques:
  k-anonymity, l-diversity …
- Synthetic Data
- Differential Privacy
- Homomorphic Encryption
- Secure Multi-Party Computation
- Federated Learning

SBA Research

Machine Learning and Data Management Research Group (MLDM)

https://www.sba-research.org/research/research-groups/mldm/

Privacy-preserving Data Analysis and Machine Learning

- Anonymisation Techniques: *k-anonymity, l-diversity* …
- Synthetic Data

Privacy-preserving data publishing

- **Differential Privacy**
- Homomorphic Encryption
- Secure Multi-Party Computation
- Federated Learning

Privacy-preserving computation

Differential Privacy applications

- Apple: QuickType and Emoji suggestions, Safari Energy Draining and Crashing Domains …

- Google: Next-word prediction (with Federated Learning), reports on urban mobility data, Covid-19 Vaccination Search, google maps (with secure aggregation) …

- LinkedIn: Audience Engagement API, Labor Market Insights

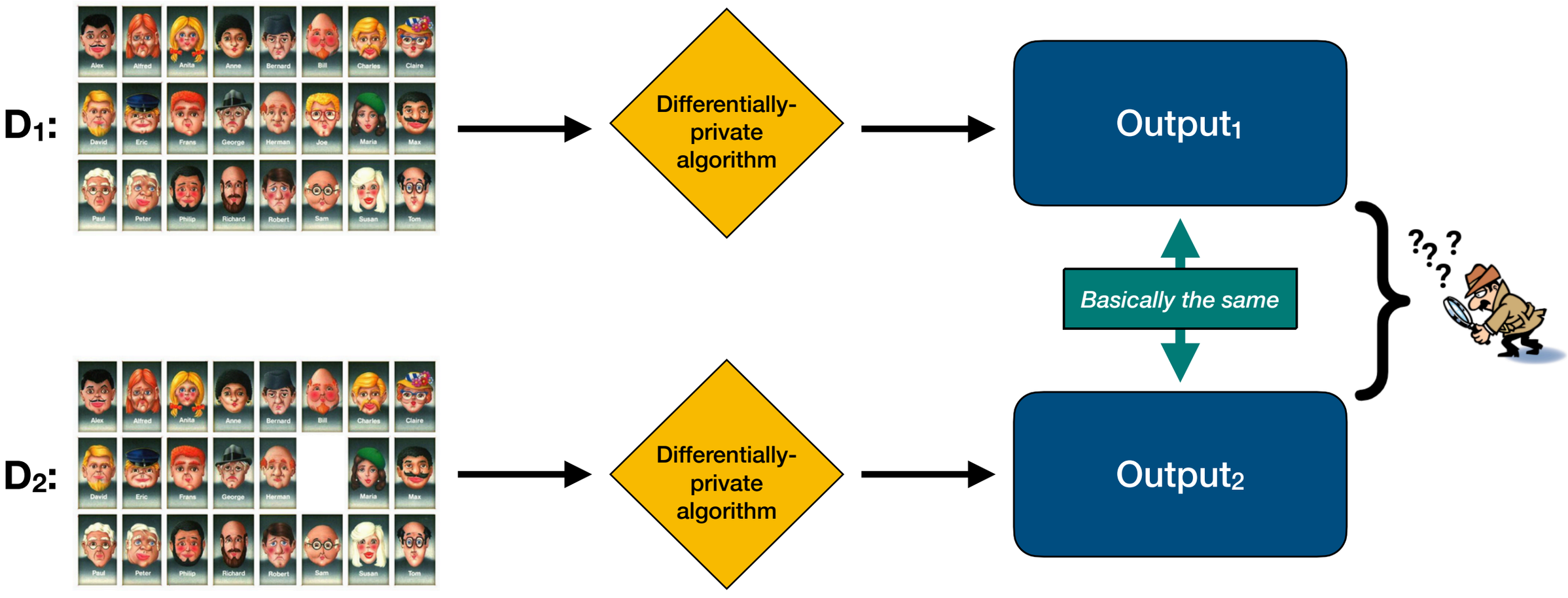
- U.S. Census Buro

- and many others…

What is Differential Privacy?

- ~~Tool, algorithm~~
- Mathematical definition of privacy
- Criterion
- Property of an algorithm

Differential Privacy *Simplified*



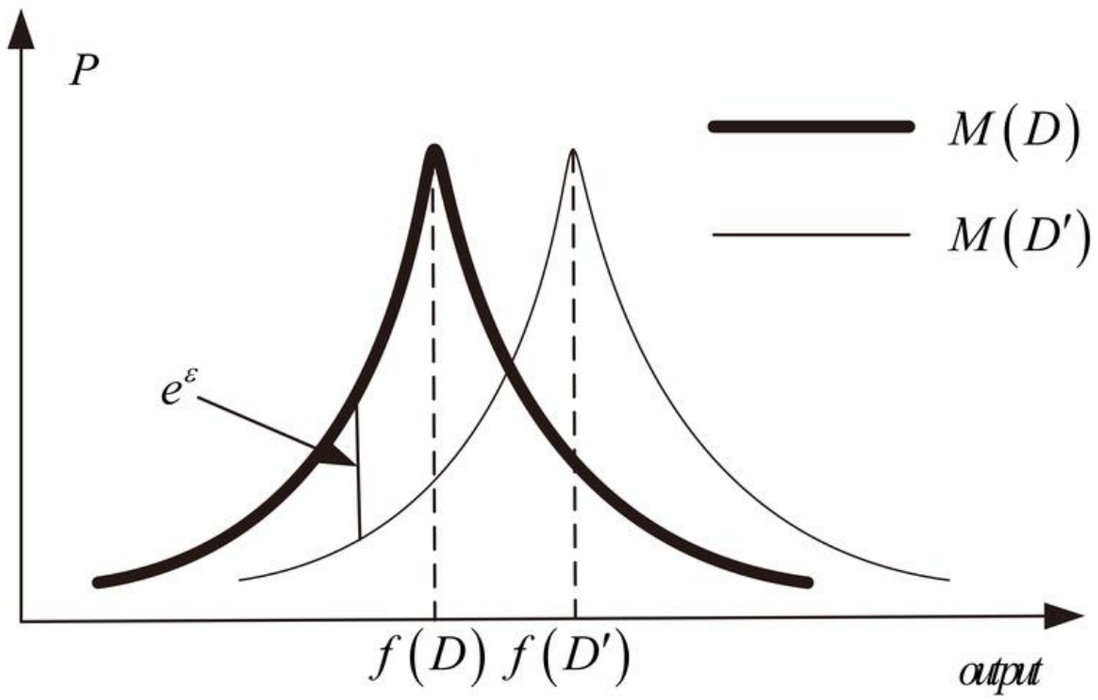
Basically the same = the probabilities to get exact same output from both databases are very similar

Differential Privacy Definition*

A randomized algorithm $\mathcal{M}$ is ϵ-differentially private if for all possible subrange of $\mathcal{M}$, say $\mathcal{S}$, and for all neighbouring dataset D_1, D_2, the probability that $\mathcal{M}$ gives the same output based on the two input datasets is very similar.

$$Pr[\mathcal{M}(D_1) \in \mathcal{S}] \leq exp(\epsilon) Pr[\mathcal{M}(D_2) \in \mathcal{S}]$$

ε - privacy loss



* Cynthia Dwork. Differential Privacy. International Colloquium on Automata, Languages, and Programming. 2006

Differential Privacy: properties

- **Quantification of <u>privacy loss</u>** through the parameter ε (*epsilon*)

- **Composition**: analysis of cumulative privacy loss:

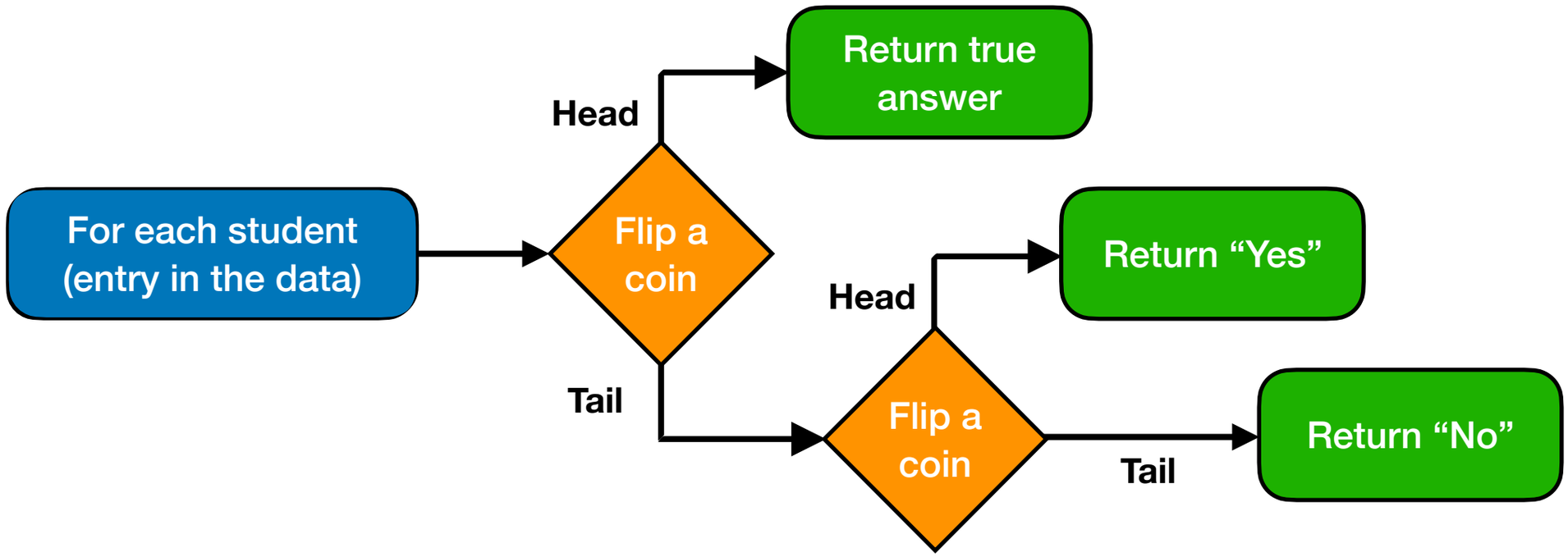
  - Sequential: if two DP mechanisms with ε_1 and ε_2 are applied on the same data, the cumulative privacy loss is $\varepsilon_1 + \varepsilon_2$ and

  - Parallel: if two DP mechanisms with ε_1 and ε_2 are applied on <u>disjoint</u> data, the cumulative privacy loss is $\max(\varepsilon_1, \varepsilon_2)$

- **Closure under post-processing** (privacy loss is not increasing without additional knowledge about the private database).

Differential Privacy: Randomised response

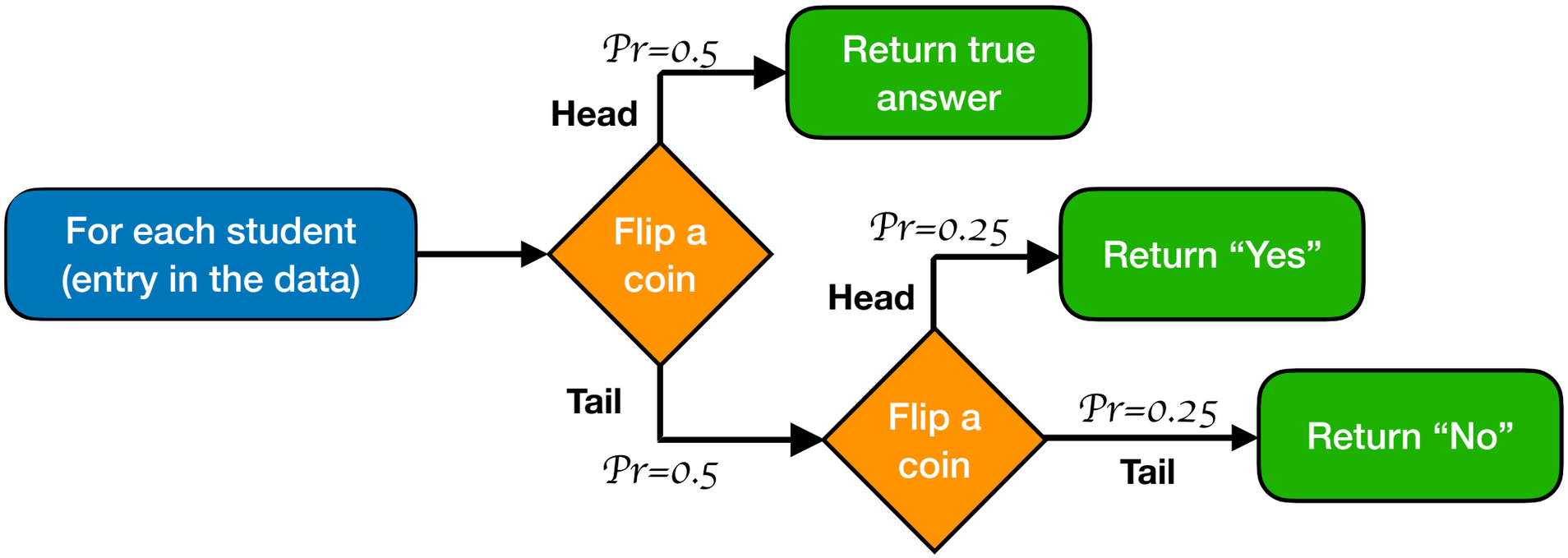
Imagine that you are a teacher and want to know how many people have cheated on the exam. Each student can answer only “yes” or “no”, but have to follow the algorithm:



Protect each person with “plausible deniability” because a person is plausible to deny the answer by the randomness of flipping a coin.

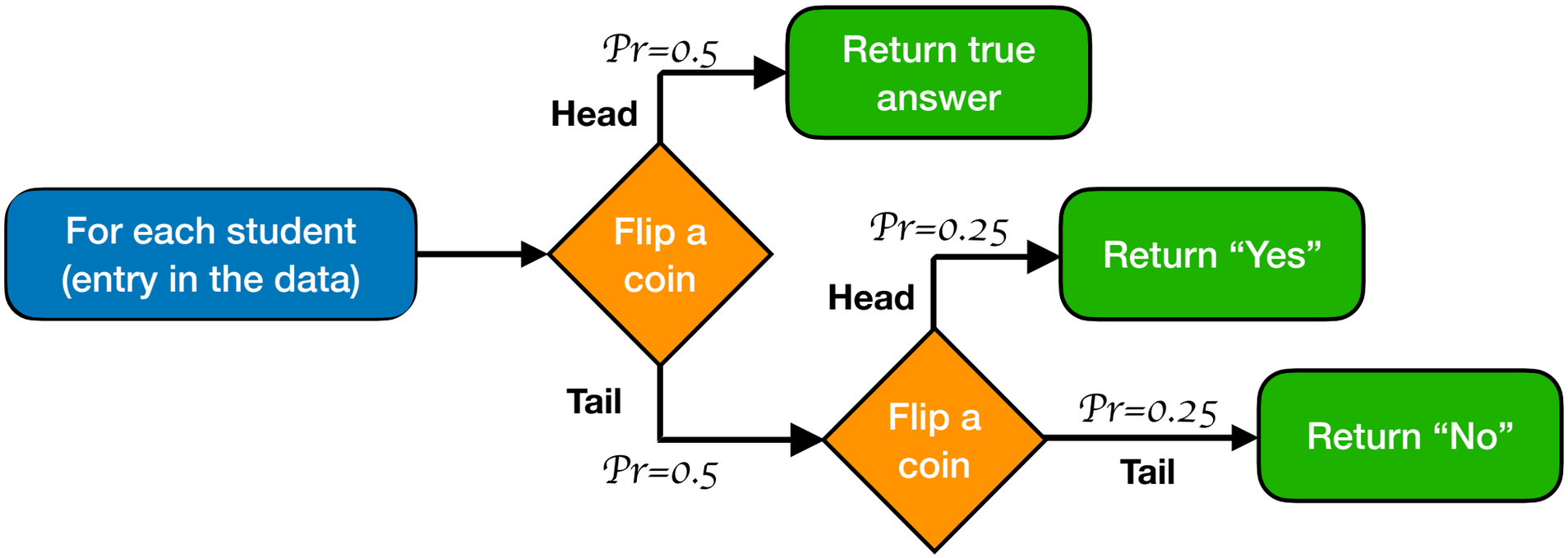
Differential Privacy: Randomised response

$$Pr[\mathcal{R}(Yes)=\text{“}Yes\text{”}] = 0.75$$



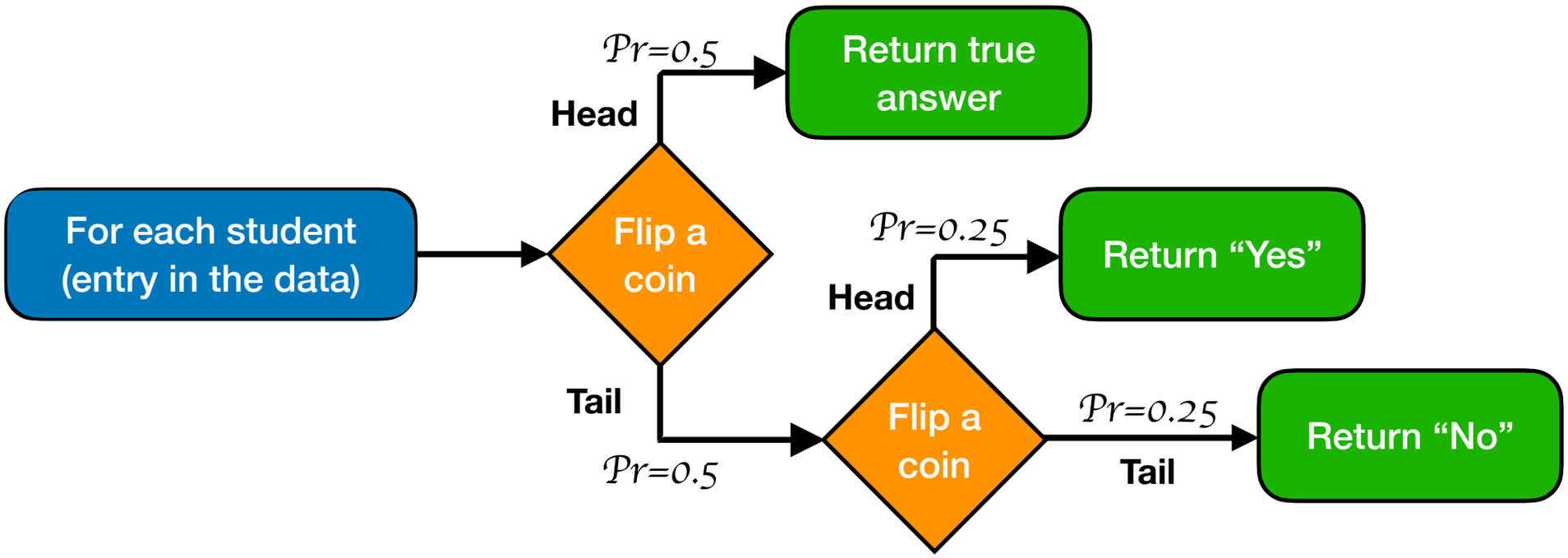
Differential Privacy: Randomised response

$$Pr[R(Yes)=\text{“}Yes\text{”}] = 0.75$$
$$Pr[R(No)=\text{“}Yes\text{”}] = 0.25$$



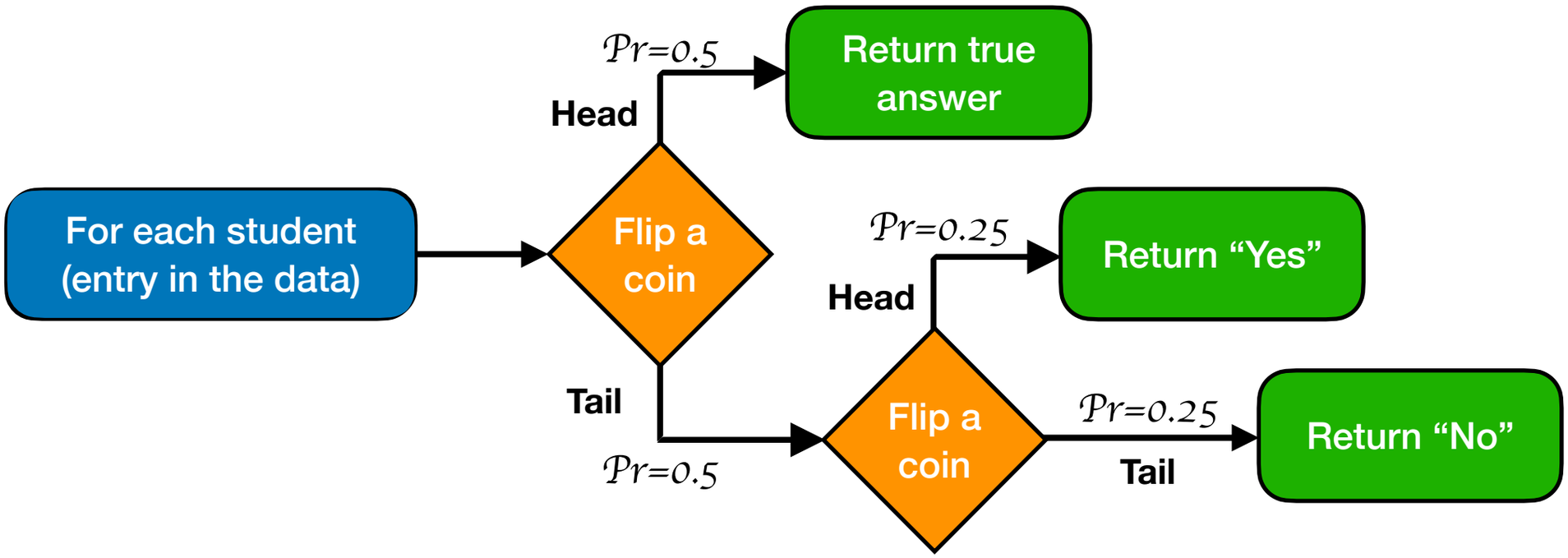
Differential Privacy: Randomised response

$$Pr[R(Yes)=\text{“}Yes\text{”}] = 0.75 = Pr[R(No)=\text{“}No\text{”}]$$
$$Pr[R(No)=\text{“}Yes\text{”}] = 0.25 = Pr[R(Yes)=\text{“}No\text{”}]$$



Differential Privacy: Randomised response

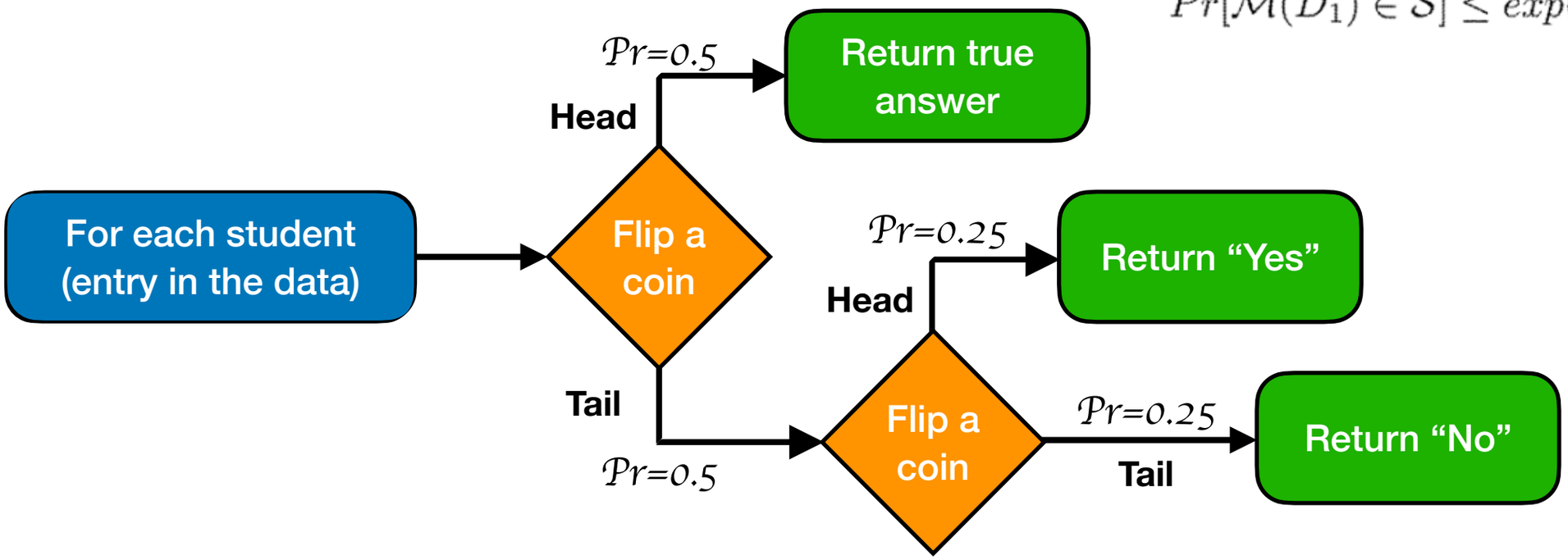
$$\frac{Pr[R(Yes)=\text{“}Yes\text{”}]}{Pr[R(No)=\text{“}Yes\text{”}]} = \frac{0.75}{0.25} = \frac{Pr[R(No)=\text{“}No\text{”}]}{Pr[R(Yes)=\text{“}No\text{”}]} = 3$$



Differential Privacy: Randomised response

$$\frac{Pr[R(Yes)=\text{“}Yes\text{”}]}{Pr[R(No)=\text{“}Yes\text{”}]} = \frac{0.75}{0.25} = \frac{Pr[R(No)=\text{“}No\text{”}]}{Pr[R(Yes)=\text{“}No\text{”}]} = 3 = exp(\varepsilon)$$

$$Pr[\mathcal{M}(D_1) \in S] \leq exp(\epsilon) Pr[\mathcal{M}(D_2) \in S]$$

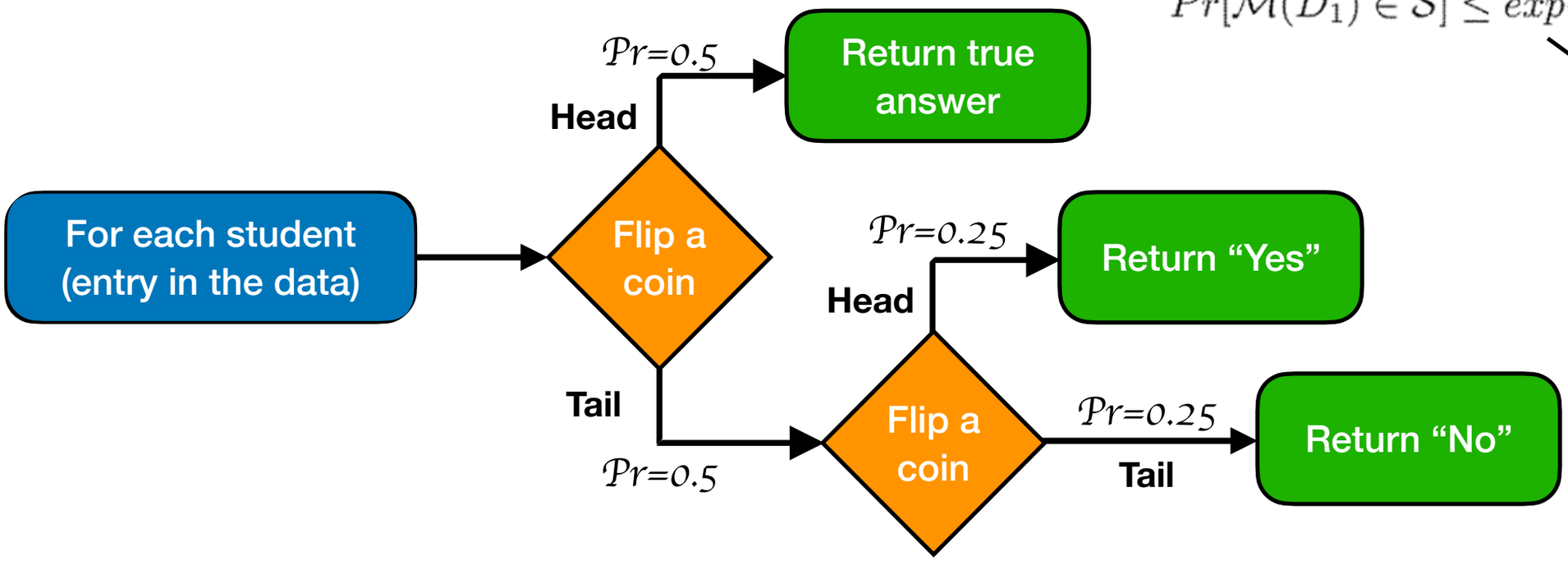


Differential Privacy: Randomised response

$$\frac{Pr[\mathcal{R}(Yes)=\text{“}Yes\text{”}]}{Pr[\mathcal{R}(No)=\text{“}Yes\text{”}]} = \frac{0.75}{0.25} = \frac{Pr[\mathcal{R}(No)=\text{“}No\text{”}]}{Pr[\mathcal{R}(Yes)=\text{“}No\text{”}]} = 3 = exp(\varepsilon)$$

$$Pr[\mathcal{M}(D_1) \in S] \leq exp(\epsilon) Pr[\mathcal{M}(D_2) \in S]$$

$$\varepsilon = ln(3)$$



Randomised response algorithm $\mathcal{R}$ is (ε)-differentially private with $\varepsilon = ln(3)$

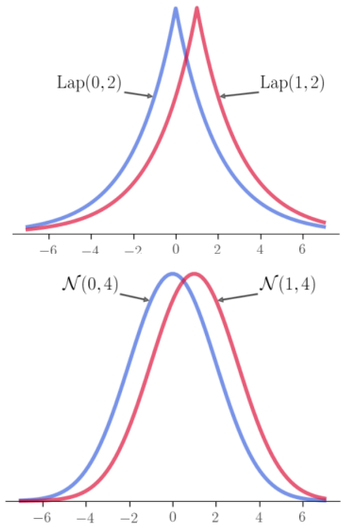
Mechanisms to achieve Differential Privacy

- Randomised response
- Laplace mechanism
- Gaussian mechanism: (ε,δ)-differential privacy
- Exponential mechanism

Mechanisms to achieve Differential Privacy

- Randomised response
- Laplace mechanism

Noise drawn from Laplace distribution



Noise drawn from Gaussian distribution

- Gaussian mechanism: (ε,δ)-differential privacy

$$Pr[\mathcal{M}(D_1) \in \mathcal{S}] \leq exp(\epsilon) Pr[\mathcal{M}(D_2) \in \mathcal{S}] + \delta.$$

- Exponential mechanism

Sensitivity of a function

- refers to the maximum possible change in the function output when a single record is added or deleted from the input

- gives an upper bound on how much we must perturb its output to preserve privacy

- measured using the ℓ_1 norm for the Laplace mechanism and ℓ_2 norm for the Gaussian mechanism

Definition * (ℓ_p-sensitivity). *Let f be a query mapping from the space of datasets to $\mathbb{R}^k$. Let N be the set of all possible pairs of neighboring datasets, i.e.,*
$N = \{(D, D') \mid D$ and D' are neighbors$\}$. For a fixed positive scalar p, the ℓ_p-sensitivity of f is defined by

$$S(f; p) = \max_{D, D' \in N} \|f(D) - f(D')\|_p. \tag{3}$$

* Natalia Ponomareva, Sergei Vassilvitskii, Zheng Xu, Brendan McMahan, Alexey Kurakin, and Chiyaun Zhang. 2023. How to DP-fy ML: A Practical Tutorial to Machine Learning with Differential Privacy. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23). Association for Computing Machinery, New York, NY, USA, 5823 - 5824. https://doi.org/10.1145/3580305.3599561

Sensitivity of a function

Global sensitivity does not depend on the dataset, and may be unbounded or very difficult to estimate.

Example 1: A query f that counts the number of records in $\mathcal{D}$.

ℓ_1 sensitivity of a query f: $S(f,1) = max \| f(\mathcal{D}) - f(\mathcal{D}') \|_1 = 1$

Example 2: Gradient of the loss of a neural network. Gradient can have an infinite sensitivity or a bound which is very hard to compute

*Solution: bound the input (**local sensitivity**) or*

*bound the output (**clipping**)*

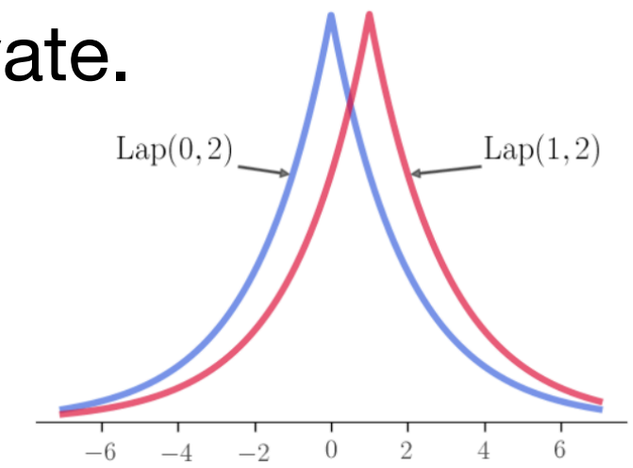
Local sensitivity is calculated for a specific dataset (or bounded dataset)

Laplace mechanism

Given the function $f(\mathcal{D}) \in \mathcal{R}^k$ with ℓ_1 sensitivity $S(f,1)$. The Laplace mechanism $\mathcal{A}_{\mathcal{L}}$ is defined as:

$\mathcal{A}_{\mathcal{L}}(\mathcal{D}, f, \varepsilon) = f(\mathcal{D}) + (\mathcal{Z}_{1,\ldots,} \mathcal{Z}_k)$, *where* $\mathcal{Z}_i$ *are noise drawn from Laplace distribution* $\mathcal{L}ap(S(f,1) / \varepsilon)$

The Laplace mechanism is (ε)-differentially private.



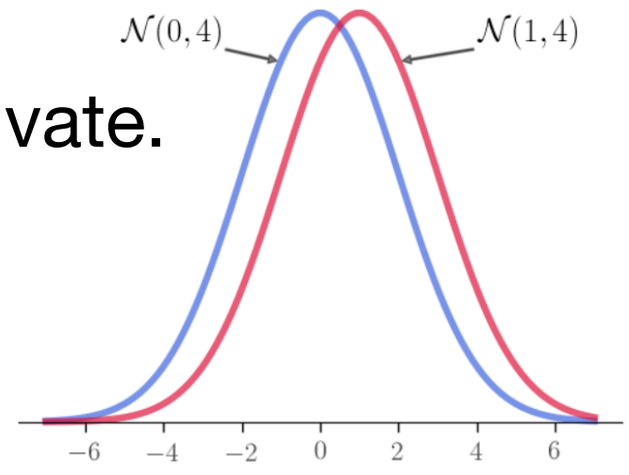
* Cynthia Dwork and Aaron Roth (2014), "The Algorithmic Foundations of Differential Privacy", Foundations and Trends® in Theoretical Computer Science: Vol. 9: No. 3 – 4, pp 211-407. http://dx.doi.org/10.1561/0400000042

Gaussian mechanism

Given the function $f(\mathcal{D}) \in \mathcal{R}^k$ with **ℓ_2 sensitivity** $S(f,2)$. The **Gaussian** mechanism $\mathcal{A}_G$ is defined as:

$\mathcal{A}_G(\mathcal{D}, f, \varepsilon) = f(\mathcal{D}) + (Z_1, \ldots, Z_k)$, *where* Z_i *are noise drawn from* ***Gaussian (Normal)*** *distribution* $\mathcal{N}(\ (S(f,2)*ln(1/\delta))\ /\ \varepsilon)$

The **Gaussian** mechanism is (ε, δ)-differentially private.



$$Pr[\mathcal{M}(D_1) \in \mathcal{S}] \leq exp(\epsilon) Pr[\mathcal{M}(D_2) \in \mathcal{S}] + \delta.$$

δ - *is a relaxation term,* $0 < \delta << 1/n$; (n- *number of records)*

* Cynthia Dwork and Aaron Roth (2014), "The Algorithmic Foundations of Differential Privacy", Foundations and Trends® in Theoretical Computer Science: Vol. 9: No. 3 – 4, pp 211-407. http://dx.doi.org/10.1561/0400000042

DP parameters: ε and δ

$$Pr[\mathcal{M}(D_1) \in \mathcal{S}] \leq exp(\epsilon) Pr[\mathcal{M}(D_2) \in \mathcal{S}] + \delta.$$

ε is a **privacy loss** parameter (the maximal knowledge gain for an attacker).

δ denotes the probability of being successfully identified by the attacker in this worst-case scenario. The expected number of successful attacks is δn. Choosing $\delta << 1/n$ will ensure that the expected number of successful attacks is much smaller than 1.*

The Gaussian mechanism provides weaker privacy guarantees compared to the Laplace mechanism.

* Natalia Ponomareva, Sergei Vassilvitskii, Zheng Xu, Brendan McMahan, Alexey Kurakin, and Chiyaun Zhang. 2023. How to DP-fy ML: A Practical Tutorial to Machine Learning with Differential Privacy. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23). Association for Computing Machinery, New York, NY, USA, 5823 – 5824. https://doi.org/10.1145/3580305.3599561

Privacy and Utility trade-off

- Noise added when applying differential privacy inevitably causes a drop in the utility of the data (and ML models).

- When $\varepsilon = \mathcal{O}$, the output of a differentially private algorithm becomes independent from the input -> full utility loss

- Too high values of ε can result in non-meaningful DP application i.e. very weak or no privacy guarantees at all.

Dwork, Cynthia, Nitin Kohli, and Deirdre Mulligan. 2019. “Differential Privacy in Practice: Expose Your Epsilons!”. *Journal of Privacy and Confidentiality* 9 (2). https://doi.org/10.29012/jpc.689.

Privacy and Utility trade-off

- For common statistical database queries (e.g., mean of a column), ε is typically chosen to be less than one **$\varepsilon \leq 1$**.
- In deep learning, this choice is usually relaxed to **$\varepsilon \leq 10$**

A list of real-world uses of differential privacy (with epsilons)

Natalia Ponomareva, Sergei Vassilvitskii, Zheng Xu, Brendan McMahan, Alexey Kurakin, and Chiyaun Zhang. 2023. How to DP-fy ML: A Practical Tutorial to Machine Learning with Differential Privacy. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23).

DP summary

- Differential Privacy provides a mathematical definition of privacy
- Differential Privacy of a function can be achieved by applying different mechanisms (e.g. Laplace or Gaussian), which add noise drawn from the corresponding distribution
- One can use a parameter ε (epsilon) to quantify the privacy loss and regulate privacy level and the trade-off between privacy and utility:
    - The lower the ε, the more noise is added, which results in stronger privacy guarantees and lower utility
    - Too high values of ε can result in no privacy guarantees

Differential Privacy in Machine Learning

DP application on distributed data

- **Central DP:** data owners trust the data aggregator, who applies DP to privatise the output (query on the data or ML model)

One trusted student aggregates data about cheating and gives noised statistics to the teacher

DP application on distributed data

- **Central DP:** data owners trust the data aggregator, who applies DP to privatise the output (query on the data or ML model)

One trusted student aggregates data about cheating and gives noised statistics to the teacher

- **Local DP:** data owners do not trust the data aggregator and apply DP locally before sharing data (or a model). Results in a higher utility loss compared to Central DP.

The teacher aggregates data about cheating and calculates statistics from the noised data

DP application on distributed data

- **Central DP:** data owners trust the data aggregator, who applies DP to privatise the output (query on the data or ML model)

One trusted student aggregates data about cheating and gives noised statistics to the teacher

- **Local DP:** data owners do not trust the data aggregator and apply DP locally before sharing data (or a model). Results in a higher utility loss compared to Central DP.

The teacher aggregates data about cheating and calculates statistics from the noised data

- **Distributed DP:** seeks to provide privacy of local DP with the utility of Central DP. Example in Federated Learning: Secure Aggregation of local models + central DP.

One non-trusted student aggregates data about cheating and gives noised statistics to the teacher

DP application on distributed data

- **Central DP:** data owners trust the data aggregator, who applies DP to privatise the output (query on the data or ML model)

One trusted student aggregates data about cheating and gives noised statistics to the teacher

- **Local DP:** data owners do not trust the data aggregator and apply DP locally before sharing data (or a model). Results in a higher utility loss compared to Central DP.

The teacher aggregates data about cheating and calculates statistics from the noised data

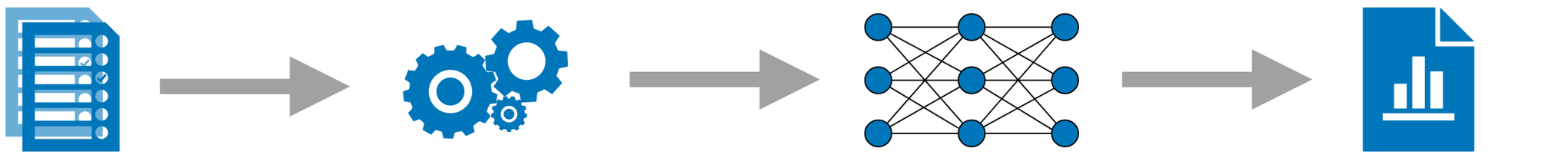
- **Distributed DP:** seeks to provide privacy of local DP with the utility of Central DP. Example in Federated Learning: Secure Aggregation of local models + central DP.

One non-trusted student aggregates data about cheating and gives noised statistics to the teacher

- **Hybrid DP:** Different DP guarantees for different users (for some local DP for others Central DP)

One partly trusted student aggregates party noised data about cheating and gives noised statistics to the teacher

Where to apply DP in ML



Input Perturbation
Adding noise to the training data

Objective Perturbation
Adding noise to the objective function

Gradient Perturbation
Adding noise to the gradients

Output Perturbation
Adding noise to the trained model

Predictions Perturbation
Adding noise to the predictions

Input Perturbation

Private Data

Input Perturbation

Adding noise to the training data

$$\mathcal{X}_i' = \mathcal{X}_i + (\mathcal{Z}_{1,\ldots,} \mathcal{Z}_k) \; for \; all \;\; \mathcal{X}_i \in \mathcal{D}^k,$$

where $(\mathcal{Z}_{1,\ldots,} \mathcal{Z}_k)$ is the noise added to each attribute depends on its sensitivity (range of values) and the number of records in the dataset (and for some approaches ML models parameters).

Challenges:

- Estimating or bounding the sensitivity for each attribute
- Preserving inter-attribute correlations

Kazuto Fukuchi, Quang Khai Tran, and Jun Sakuma. 2017. Differentially Private Empirical Risk Minimization with Input Perturbation. In Discovery Science - 20th International Conference, DS 2017, Kyoto, Japan, October 15-17, 2017, Proceedings

Qu, C., Kong, W., Yang, L., Zhang, M., Bendersky, M., & Najork, M. (2021). Natural language understanding with privacy-preserving bert. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management

Objective perturbation

ML model training

Objective Perturbation

Adding noise to the objective function

For an ML model f and objective (loss) function $\mathcal{J}$ one can achieve Differential Privacy by minimising the noised version of objective (loss) function $\mathcal{J}'(f,\mathcal{D})$:

$$\mathcal{J}'(f,\mathcal{D}) = \mathcal{J}'(f,\mathcal{D}) + \mathcal{Z}$$

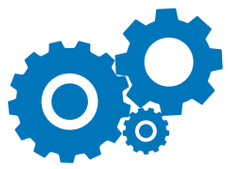
where $\mathcal{Z}$ is the noise dependant on the sensitivity of the objective function and the number of records in the dataset.

Challenges:

- Works only for convex objective functions
- To estimate sensitivity for non-convex objective functions one has to use e.g. modified versions of an objective function[1], or Taylor expansion[2]

[1] Chaudhuri, K., Monteleoni, C., & Sarwate, A. D. (2011). Differentially private empirical risk minimization. Journal of Machine Learning Research, 12 (29), 1069–1109.
[2] Phan, N., Wang, Y., Wu, X., & Dou, D. (2016). Differential privacy preservation for deep auto-encoders: An application of human behavior prediction. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence

Gradient perturbation: DP-SGD*



ML model training

Gradient Perturbation

Adding noise to the gradients

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

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

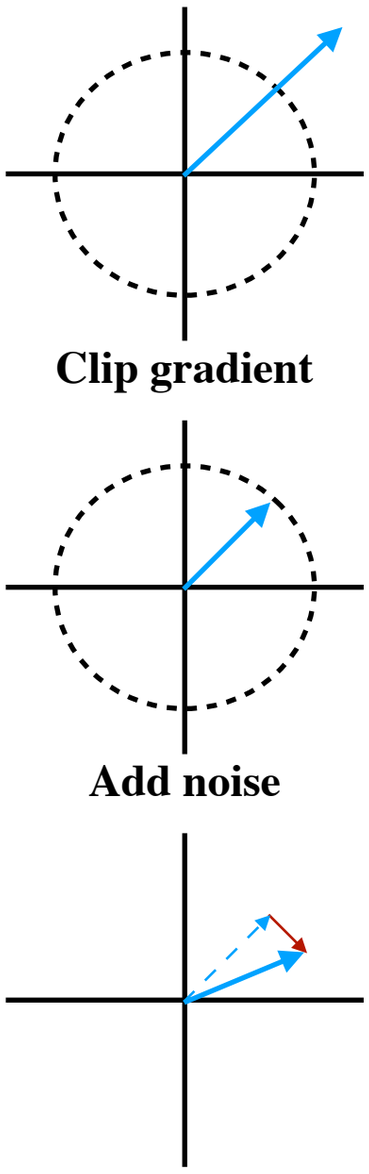
Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

* Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

Gradient perturbation: DP-SGD*



Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

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

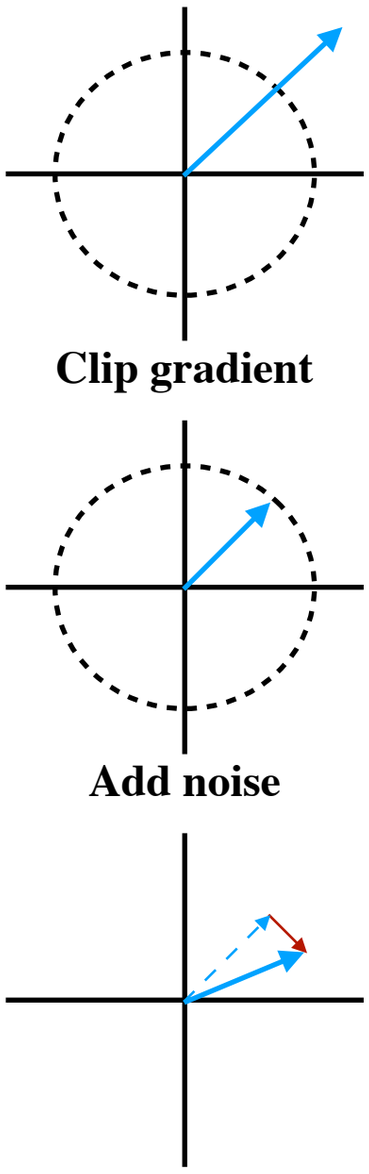
Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

* Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

Gradient perturbation: DP-SGD*



Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

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

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

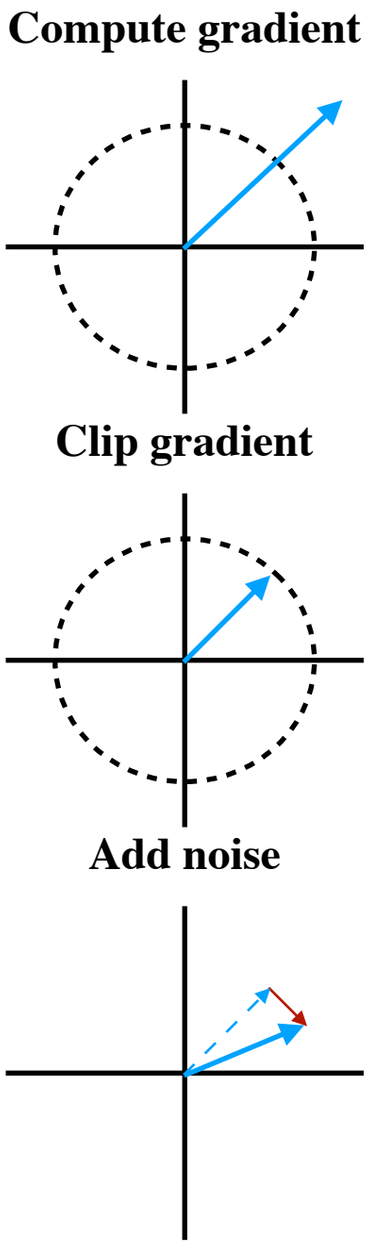
Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

<- Avoiding calculating sensitivity

<- Smaller noise because of Gaussian mechanism and randomness involved in sampling batches

* Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

Gradient perturbation: DP-SGD*



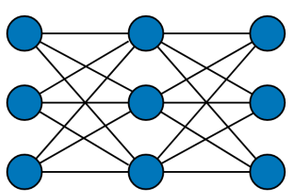
- The scale of noise σ is dependant on the number of records in the dataset (*N*), the number of iterations of SGD (*T*), the norm bound (*C*) and the batch size *L*:

$$\sigma \geq c_2 \frac{q\sqrt{T\log(1/\delta)}}{\varepsilon}$$

where $q = L / N$

* Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

Output perturbation



Trained Model

Output Perturbation
Adding noise to the trained model

For an ML model with weights θ, Differential Privacy can be achieved by adding noise to the weights:

$$\theta' = \theta + \mathcal{Z}$$

where $\mathcal{Z}$ is the noise depends on the sensitivity of the ML model. E.g. sensitivity of Logistic Regression* is $2/n\lambda\epsilon$, where n - is the number of records in the dataset, λ is l_2-regularisation parameter.

Challenges:

- Need to know the sensitivity of an ML model, which is hard to estimate for complex models

* Chaudhuri, K., & Monteleoni, C. (2008). Privacy-preserving logistic regression. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.) Advances in Neural Information Processing Systems

Predictions perturbation

Model output
(Predictions)

Predictions Perturbation

Adding noise to the predictions

- Relevant only for the settings when the model is not visible and available only for querying.
- Comes with an inference budget, as every query degrades the privacy budget

Dwork, C., & Feldman, V. (2018). Privacy-preserving prediction.

Nissim, K., Raskhodnikova, S., & Smith, A. (2007). Smooth sensitivity and sampling in private data analysis. In Proceedings of the thirty-ninth annual ACM symposium on Theory of computing

van der Maaten, L., & Hannun, A. Y. (2020). The trade-offs of private prediction.

Privacy and Utility trade-off

<<

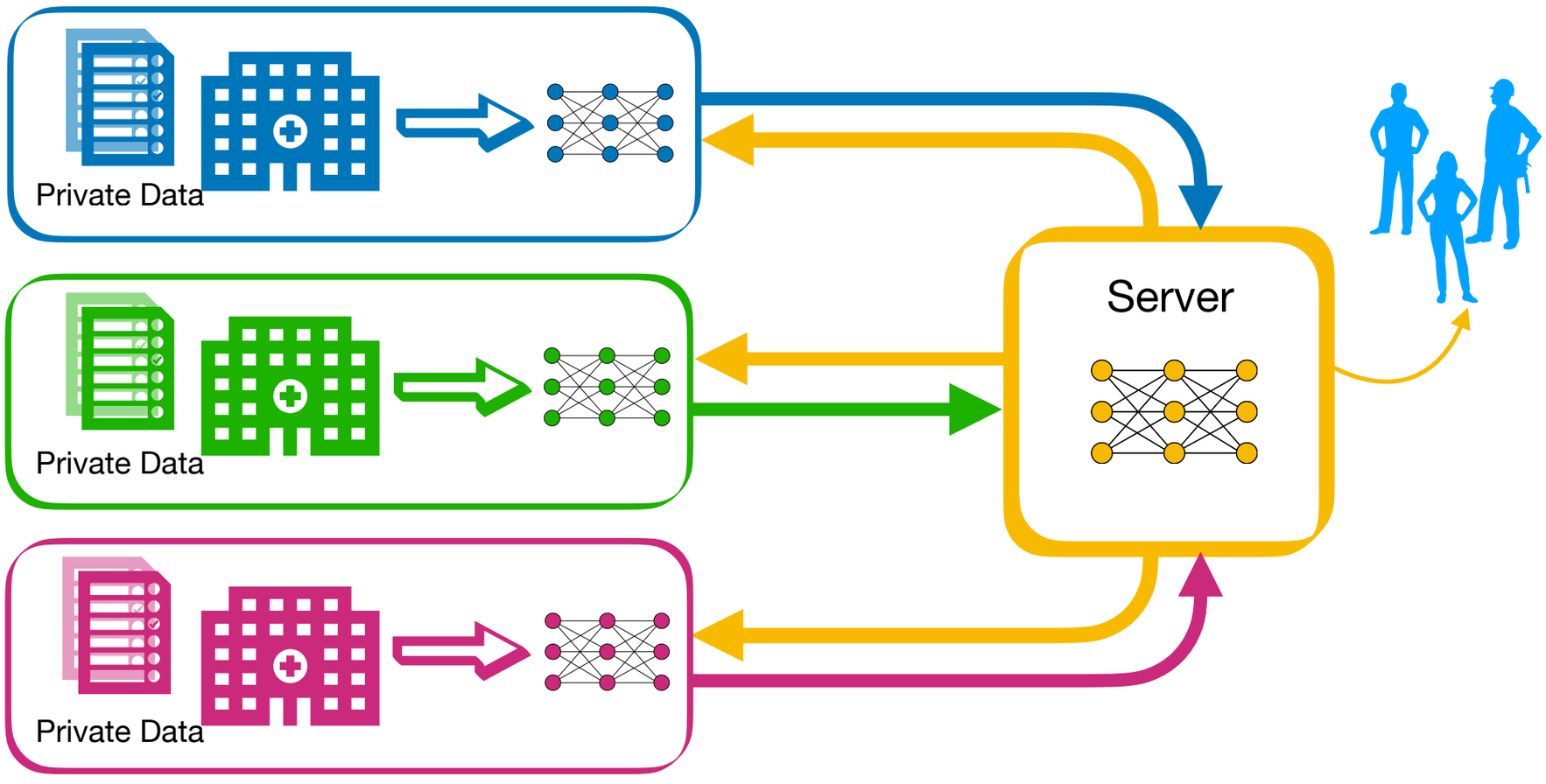
With a sufficiently large training set and sufficient computation, for a fixed model both good accuracy (that is, almost as good as a non-private model) and good privacy can often be achievable. Hence, the relevant question is not usually “Will DP work for my model? ” but rather “How much computation and data do I need to achieve reasonable privacy and utility? ”

>>

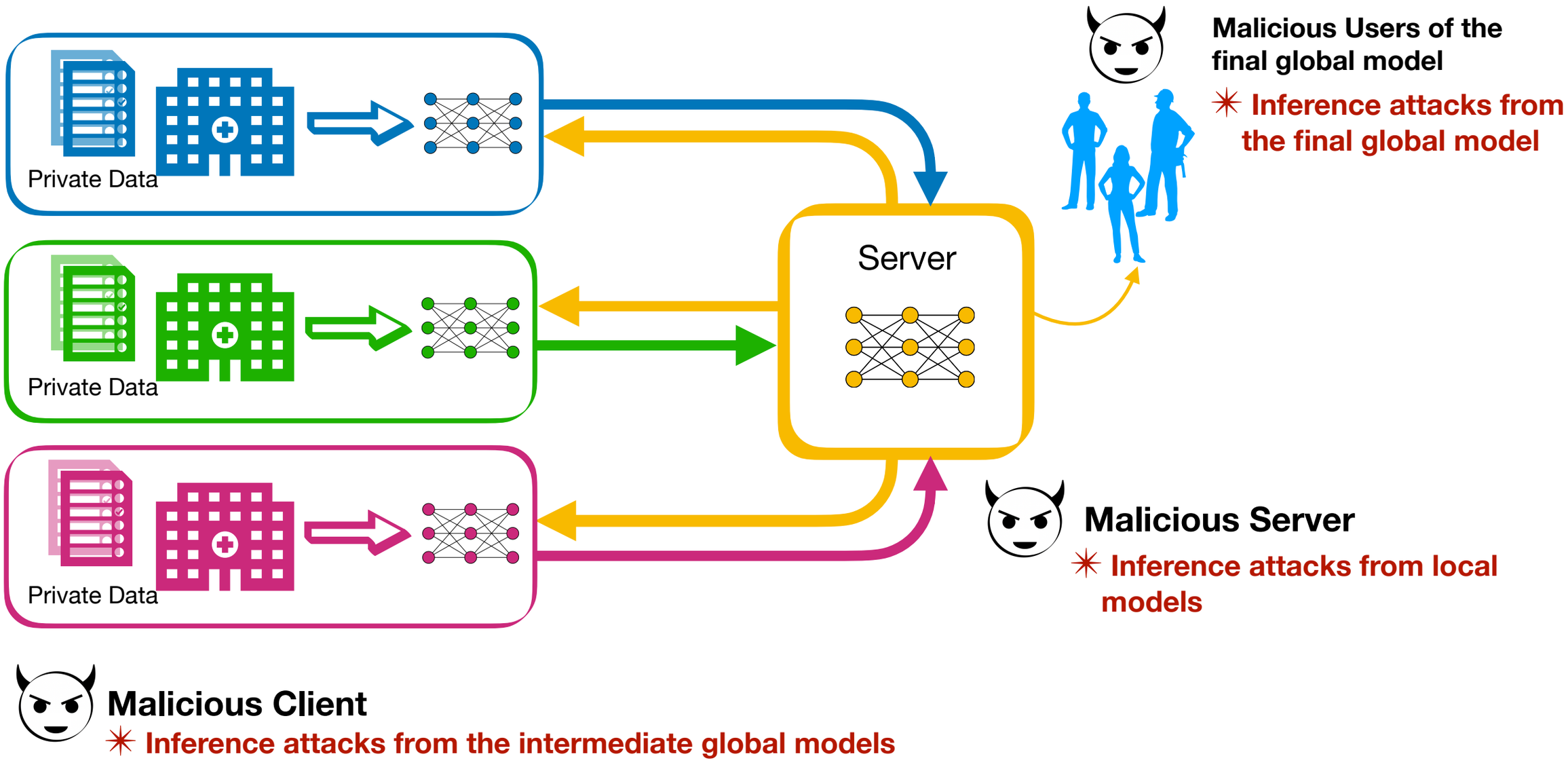
Natalia Ponomareva, Sergei Vassilvitskii, Zheng Xu, Brendan McMahan, Alexey Kurakin, and Chiyaun Zhang. 2023. How to DP-fy ML: A Practical Tutorial to Machine Learning with Differential Privacy. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23).

Differential Privacy in Federated Learning

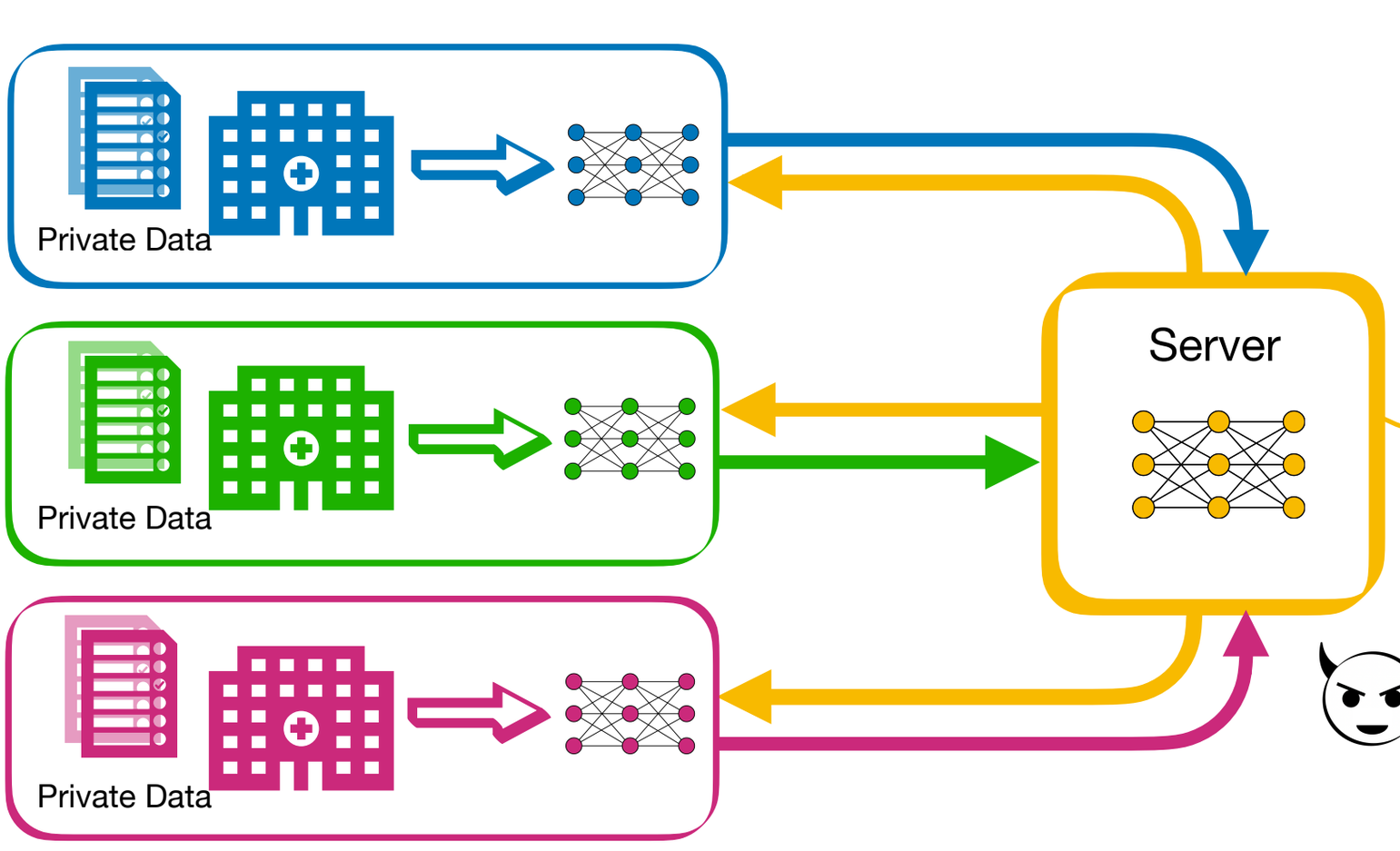
Privacy risks and mitigation in Federated Learning



Privacy risks and mitigation in Federated Learning



Privacy risks and mitigation in Federated Learning



Malicious Users of the final global model

✳ **Inference attacks from the final global model**

✓ **Differential Privacy**

Malicious Server

✳ **Inference attacks from local models**

✓ **Homomorphic encryption**

✓ **Secure Multi-Party Computation**

✓ **Differential Privacy**

Malicious Client

✳ **Inference attacks from the intermediate global models**

✓ **Differential Privacy**

Privacy-utility trade-off
Output perturbation versus DP-SGD in Federated Learning

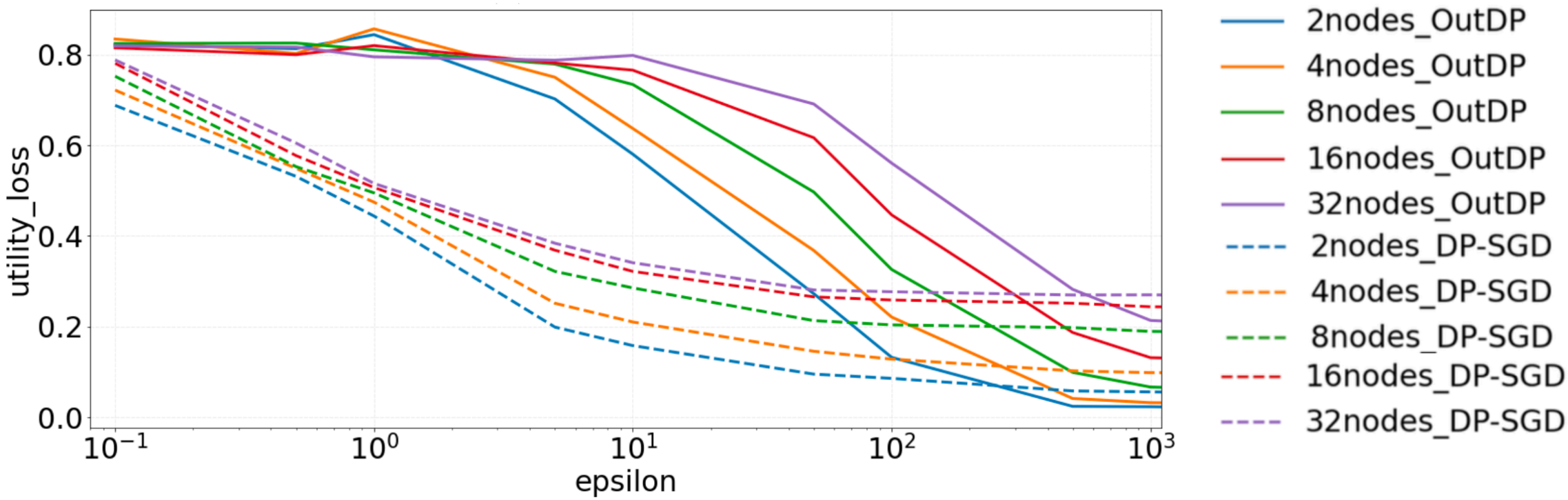
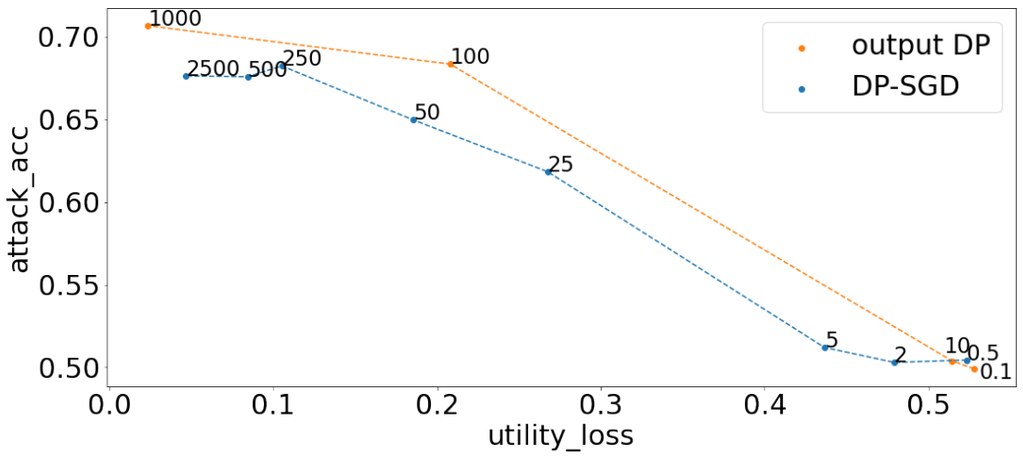


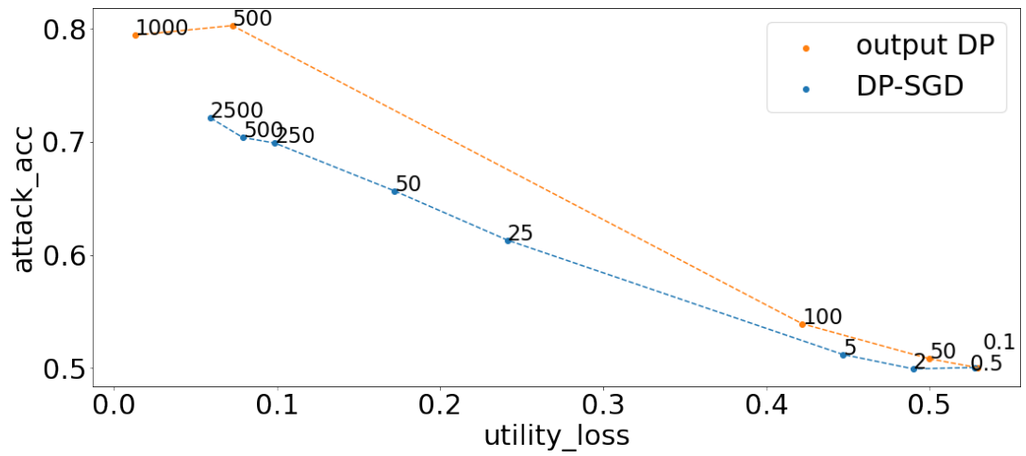
Figure: Output DP (solid lines) compared to DP-SGD (dashed lines) in FL with different number of nodes on Loan dataset.

- DP-SGD suggests a **better trade-off** between privacy and utility than output perturbation in the settings with a low privacy budget (more strict privacy)

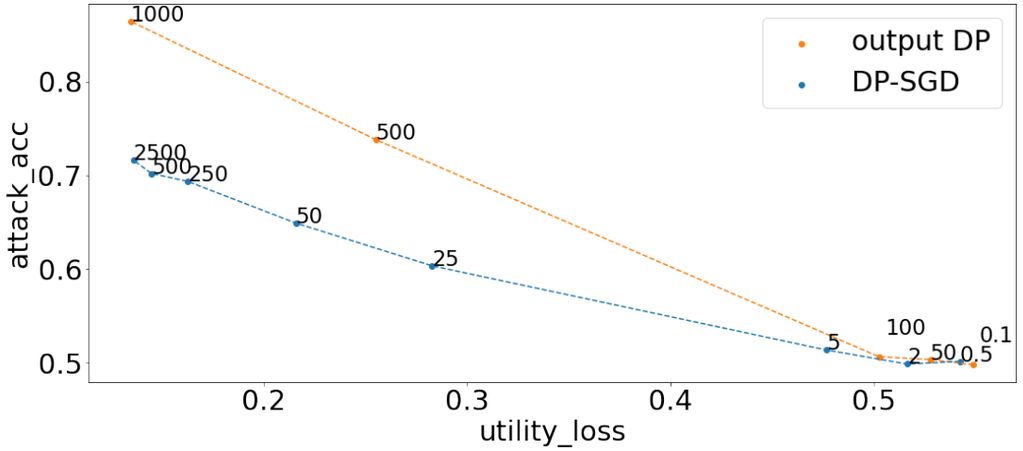
Empirical privacy through membership inference attack



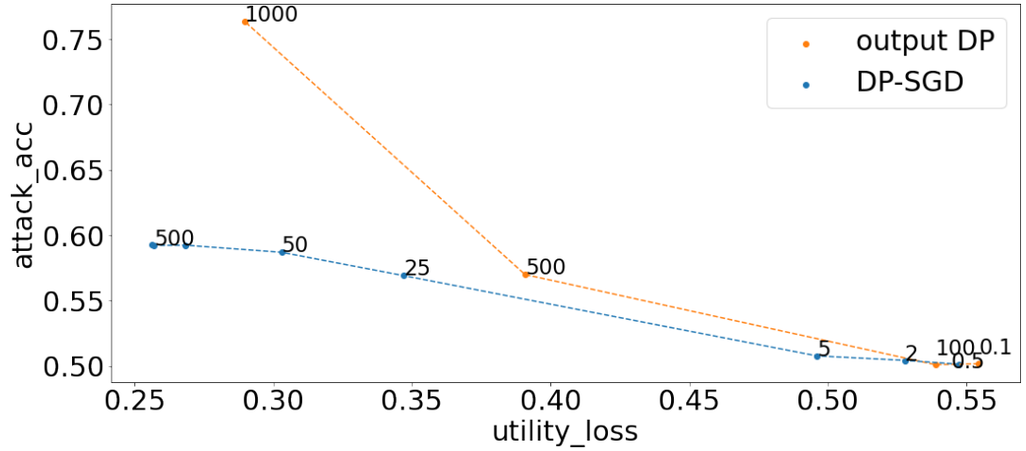
(a) Two nodes in FL



(b) Four nodes in FL



(c) Eight nodes in FL



(d) 16 nodes in FL

Figure: Privacy - Utility trade-off of output DP and DP-SGD with different epsilons (depicted on the plots)

- DP-SGD provides a better privacy-utility trade-off compared to output DP achieving lower utility loss with the set empirical privacy loss.

Takeaways

- Differential Privacy can be applied in machine learning to preserve the privacy of training data and models at different stages by adding noise to the training data, during training to objective function or gradients, after training to the model parameters and to the predictions
- Differential Privacy can be combined with other privacy-preserving techniques e.g. federated learning and SMPC, what can help to improve privacy-utility trade-off.
- Privacy-utility trade-off can be different for different dataset and approaches for achieving DP
- Having a larger dataset can help to improve privacy-utility trade-off
- One should carefully choose privacy loss parameter ε to guarantee that Differential Privacy does help preserving privacy

Thank you for your attention!

Anastasia Pustozerova

apustozerova@sba-research.org

- Privacy-preserving Machine Learning
- Federated Learning
- Differential Privacy
- Secure Multi-Party Computation

SBA Research

Machine Learning and Data Management Research Group (MLDM)

https://www.sba-research.org/research/research-groups/mldm/

Membership Inference

- having access to a ML model, infer if some sample was in the training set of the model or not.

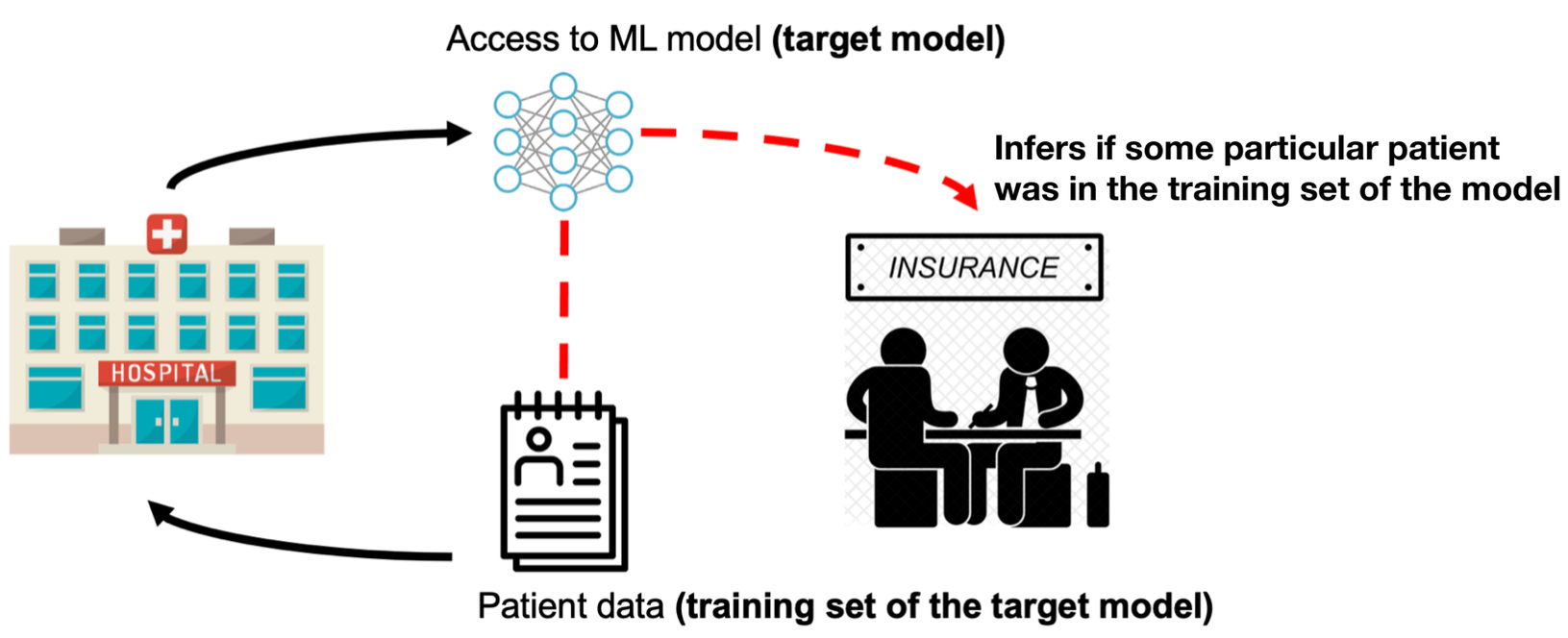
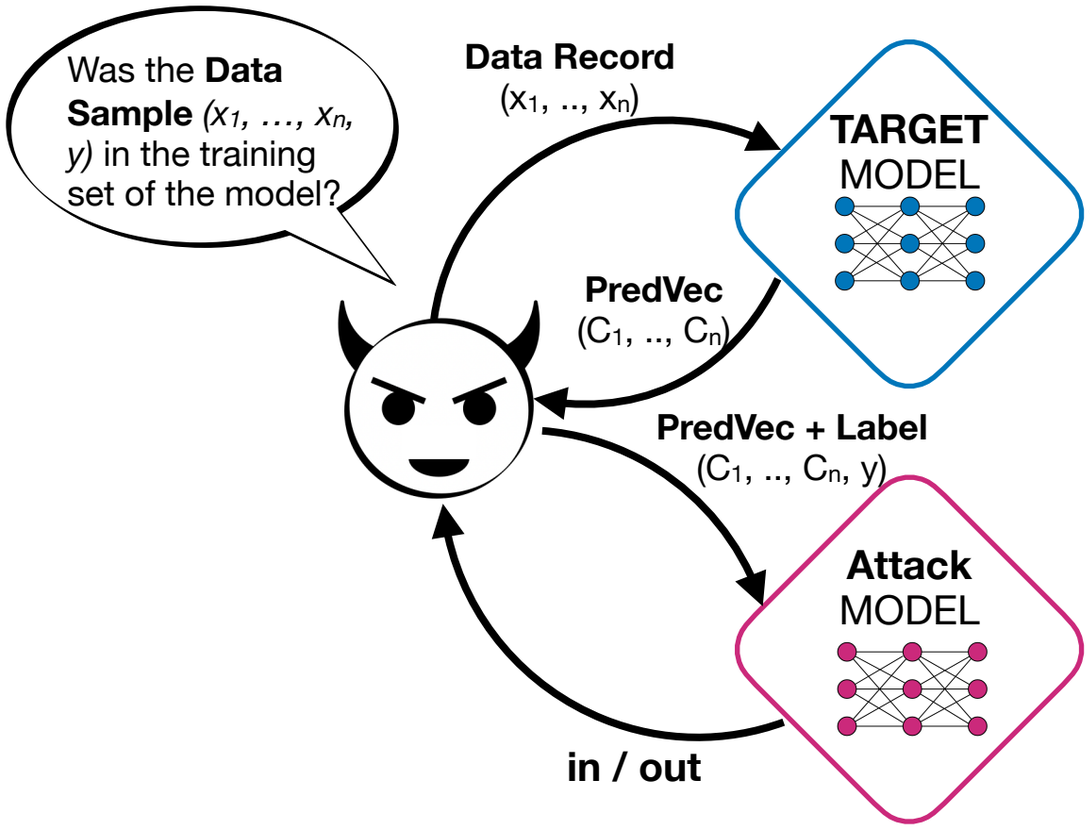


Fig. 3 Membership inference attack

R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, doi: 10.1109/SP.2017.41.

Membership Inference



1. An adversary sends the **Data Record** as an input to the **Target model**
2. Target model returns a prediction probability vector (**predVec**)
3. An adversary sends **predVec** and **Data record's Label** as an input to **attack model**
4. Attack model returns "**in**" if the **Data Record** was in the training set of the target model, otherwise, it returns "**out**"