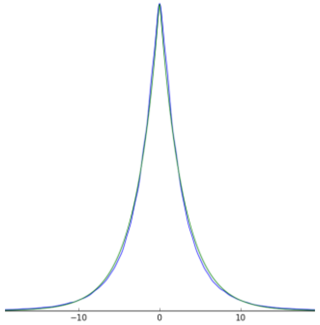
The Price of Differential Privacy under Continual Observation

Sofya Raskhodnikova

Based on joint work with:

Palak Jain, Satchit Sivakumar, and Adam Smith



BOSTON UNIVERSITY

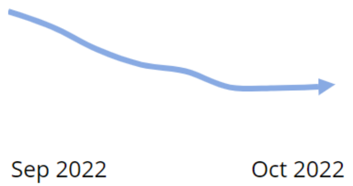
Aggregate Statistics on Sensitive Dynamic Data

COVID Data Tracker

Daily Update for the United States

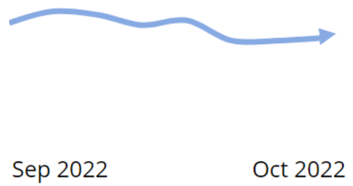
Cases
New Cases (Weekly Total)
265,893

Case Trends



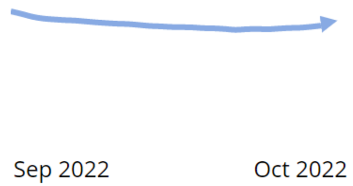
Deaths
New Deaths (Weekly Total)
2,649

Death Trends



Hospitalizations
New Admissions (Daily Avg)
3,320

Admission Trends



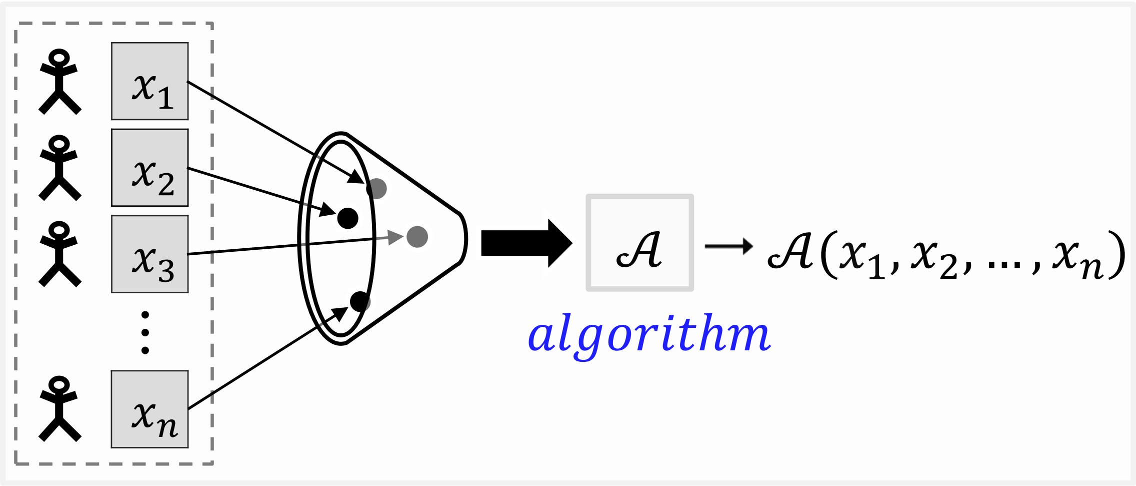
Vaccinations
% 5+ with Updated Booster Dose
7.3%

People Age 5+

Total Cases	Total Deaths	Current Hospitalizations	Total Updated Booster Doses (People 5+)
97,329,787	1,066,351	21,450	22,820,618

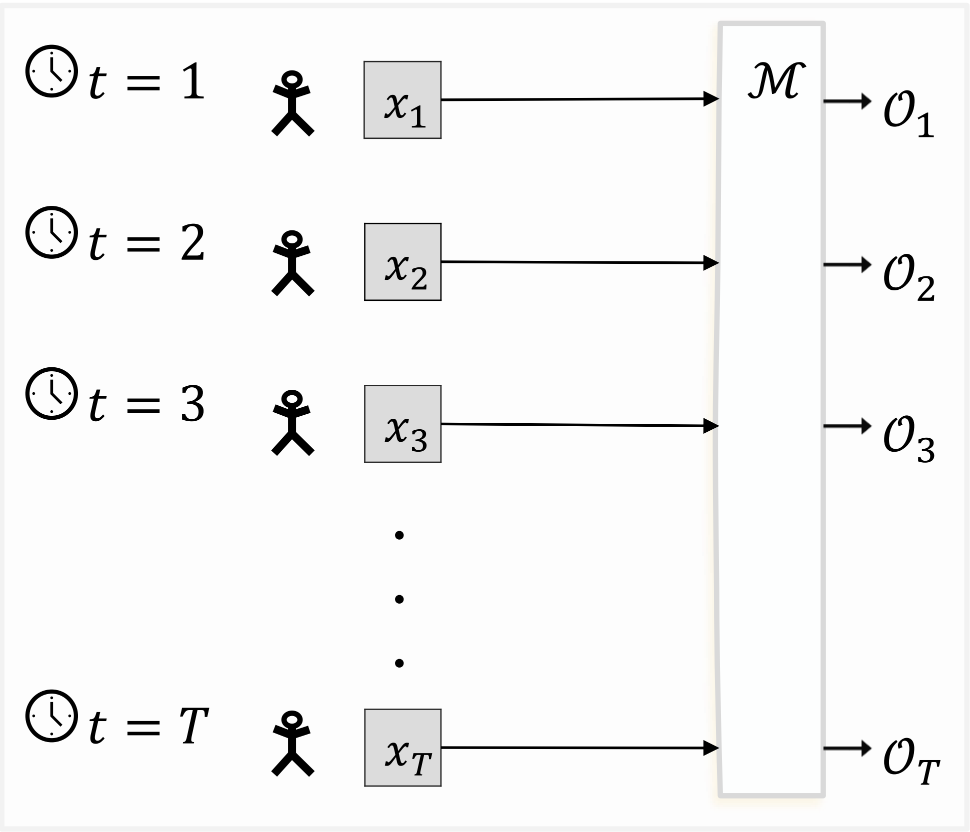
CDC | Data as of: October 28, 2022 3:52 PM ET. Posted: October 28, 2022 4:40 PM ET

Batch Model [Dwork McSherry Nissim Smith 06]



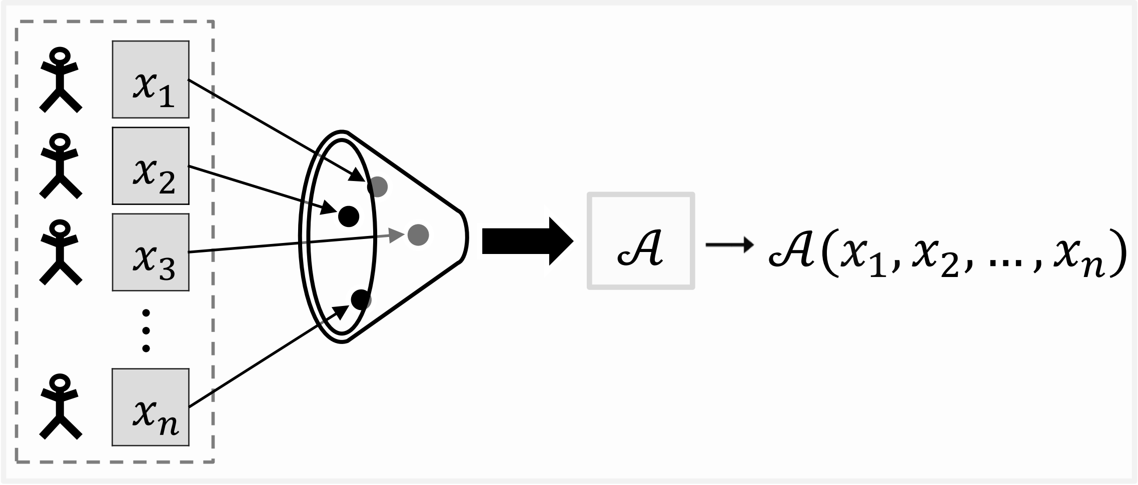
dataset $\boldsymbol{x}$

Continual Release Model [Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



mechanism

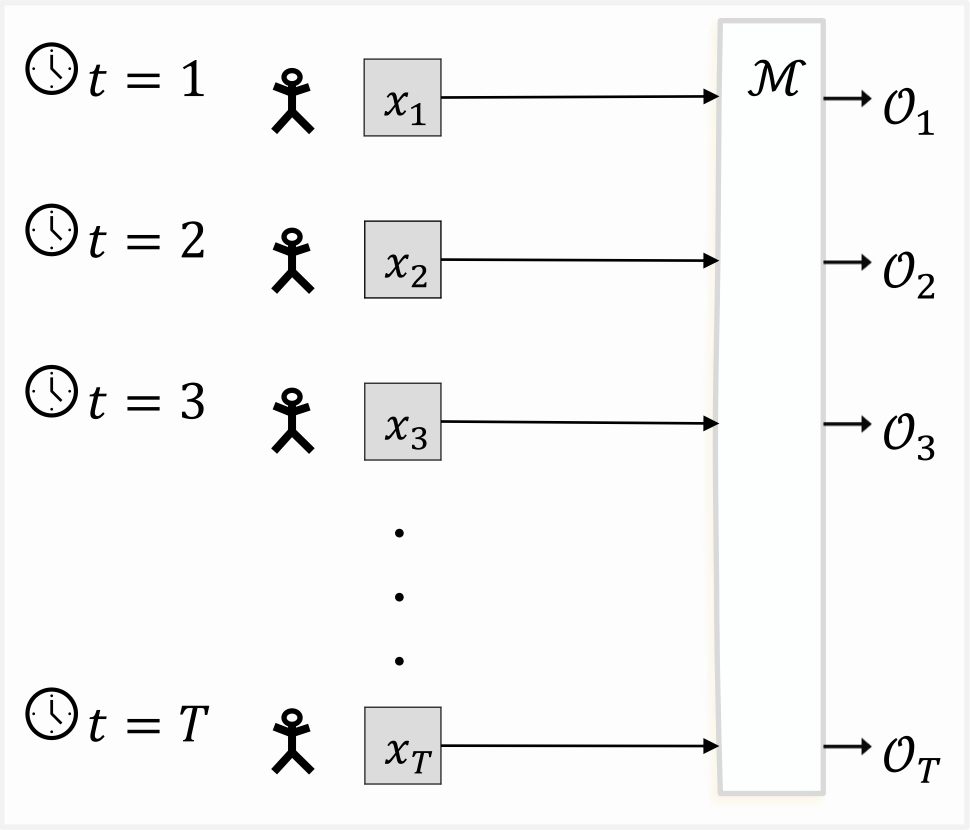
Batch Model



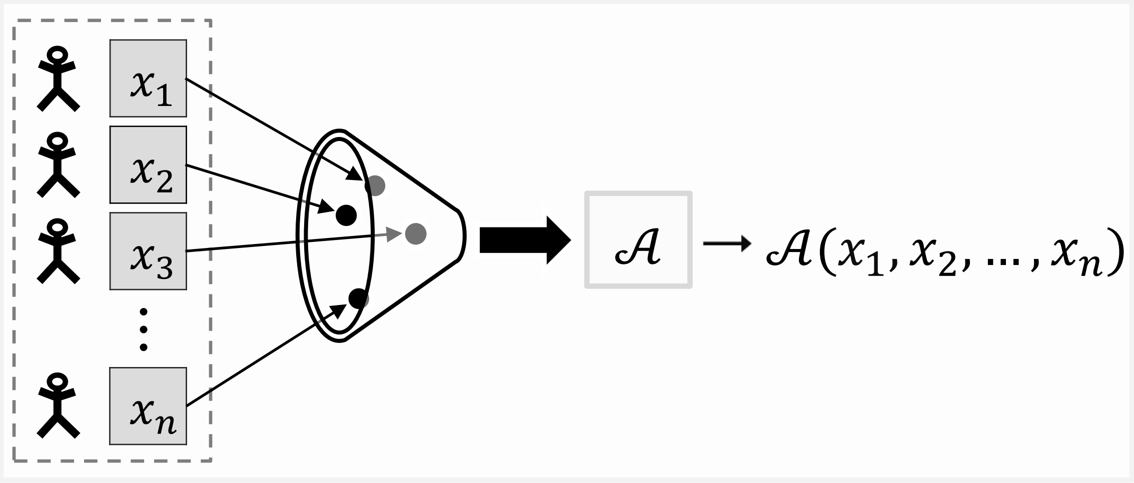
Privacy Definition

Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Batch Model [Dwork, McSherry Nissim Smith 06]

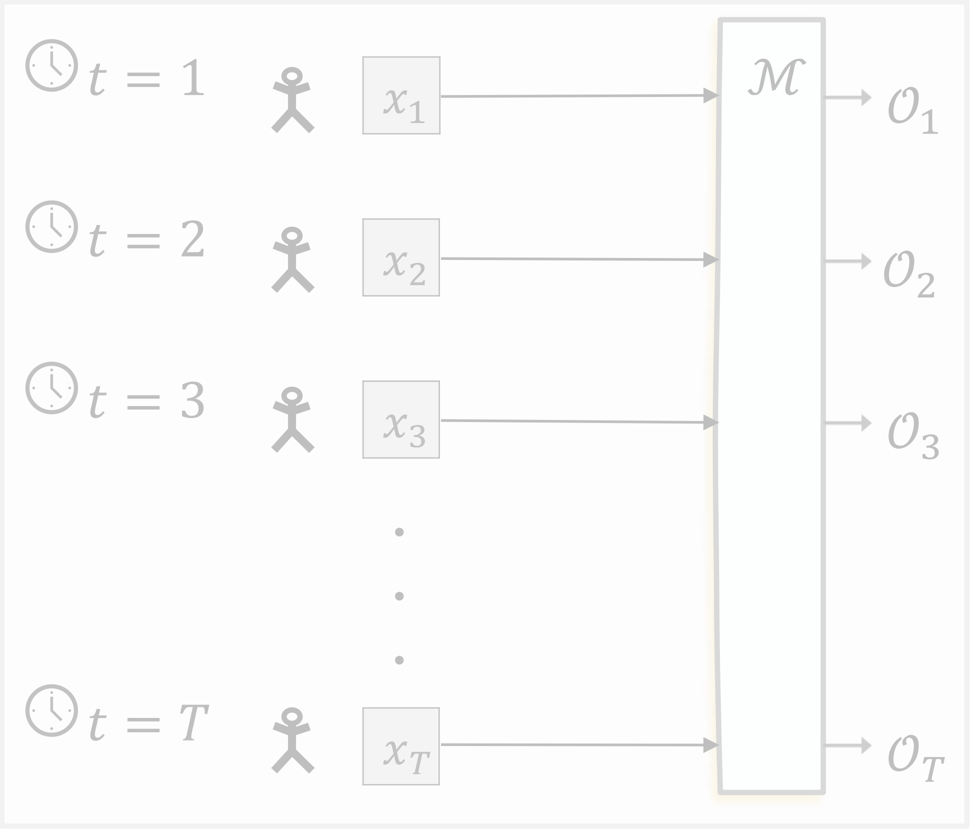


(ϵ, δ)-Differential Privacy:

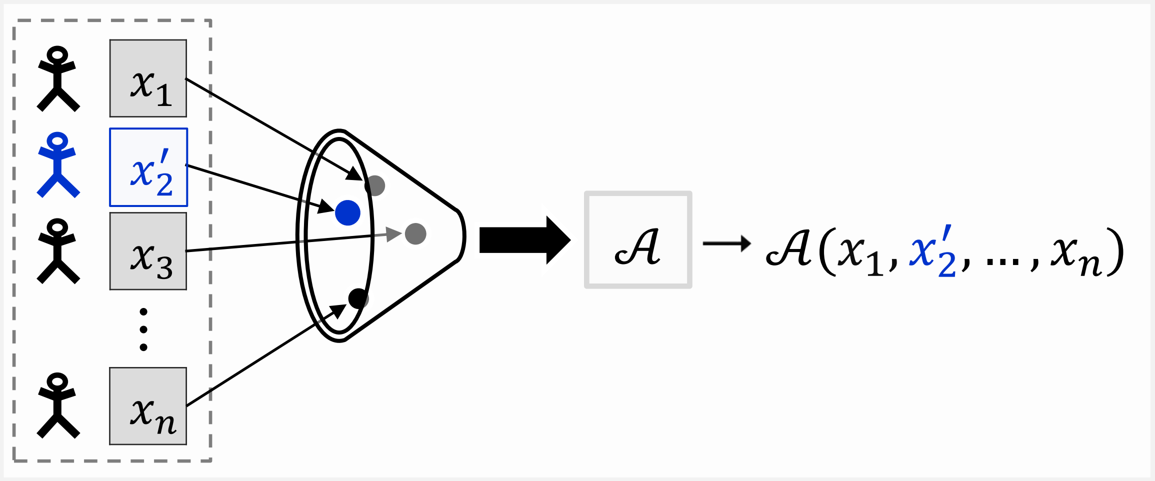
Privacy Definition: Neighboring Dataset

Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Batch Model [Dwork, McSherry Nissim Smith 06]



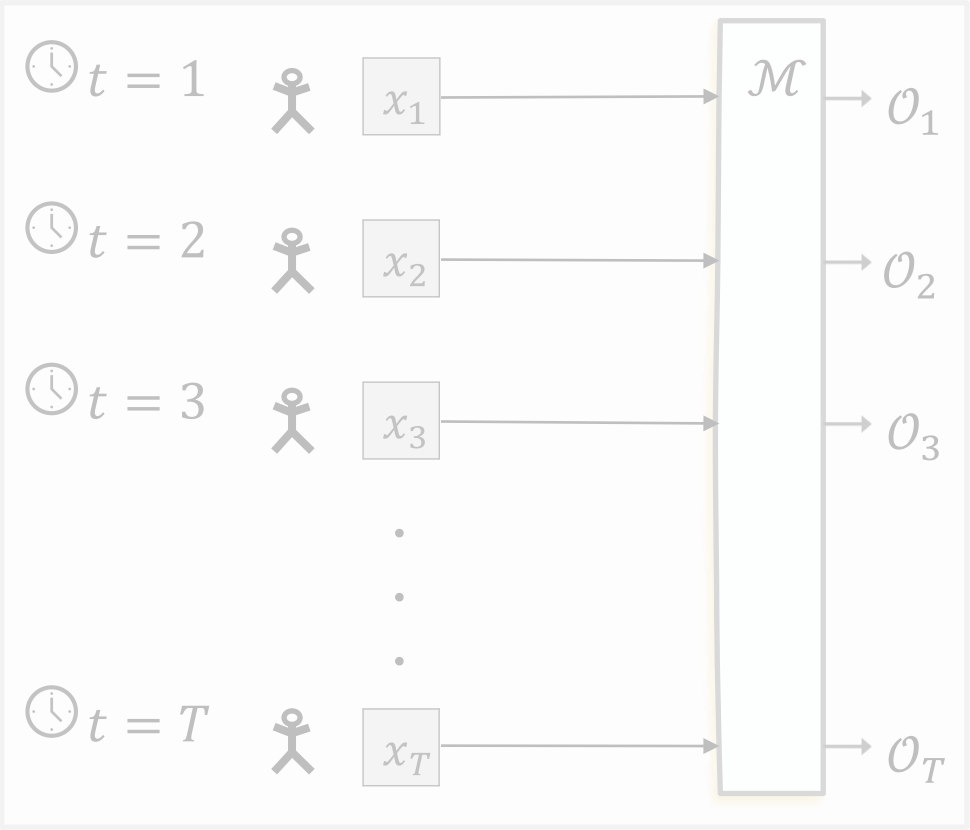
(ϵ, δ)-Differential Privacy:

Two datasets x, x' are **neighbors** if they differ in one person's data.

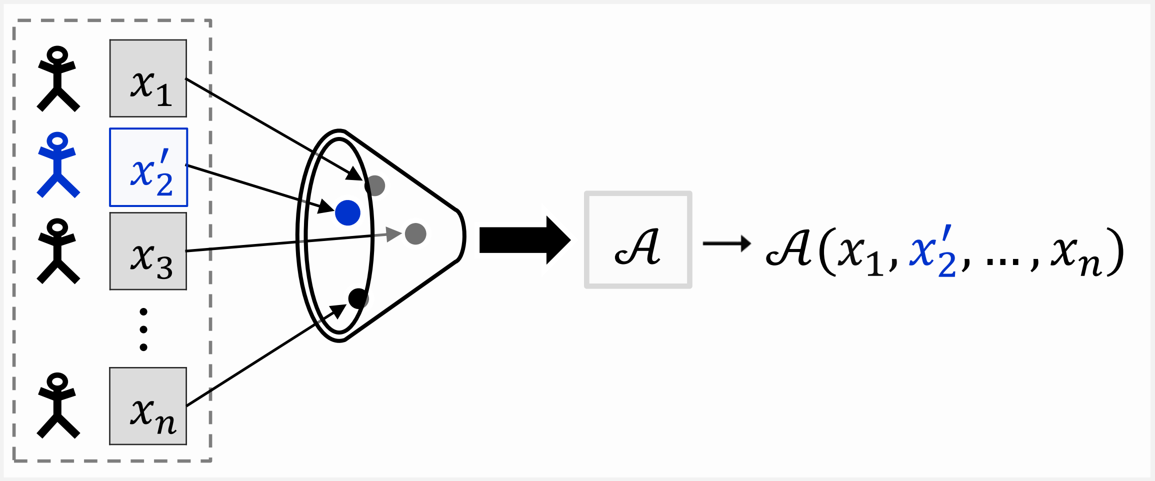
Privacy Definition

Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Batch Model [Dwork, McSherry Nissim Smith 06]



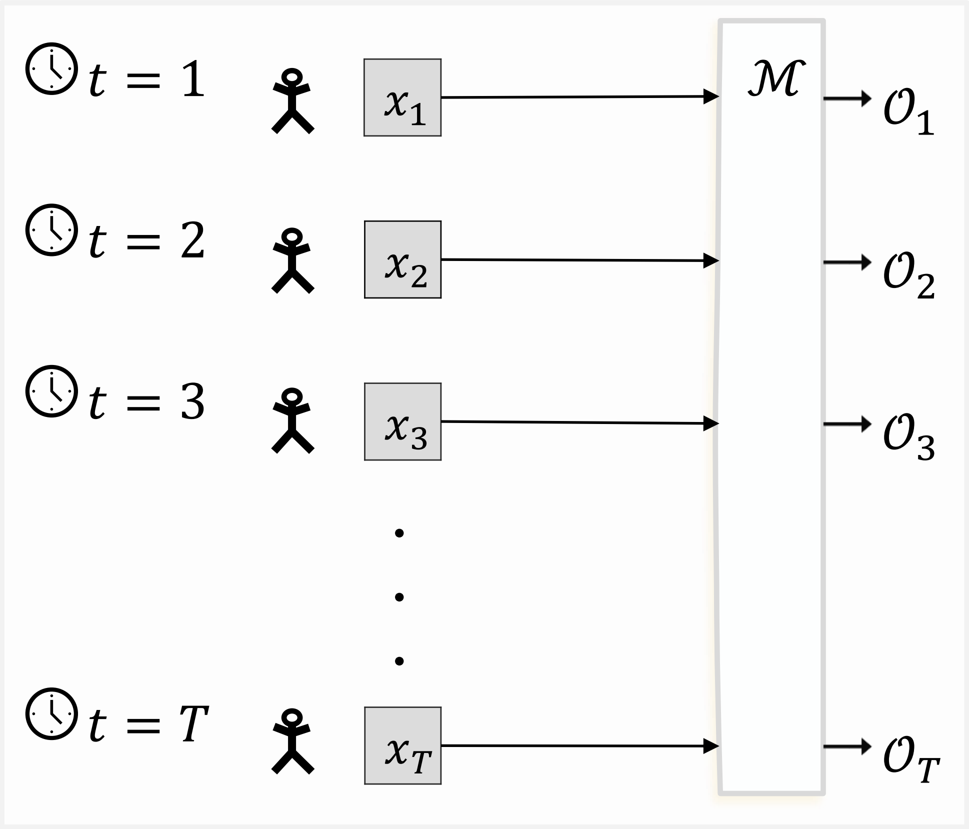
(ϵ, δ)-Differential Privacy: For all neighbors $\boldsymbol{x}, \boldsymbol{x}'$,

$$\mathcal{A}(x_1, \dots, x_t, \dots, x_n) \approx_{\epsilon,\delta} \mathcal{A}(x_1, \dots, x_t', \dots, x_n)$$

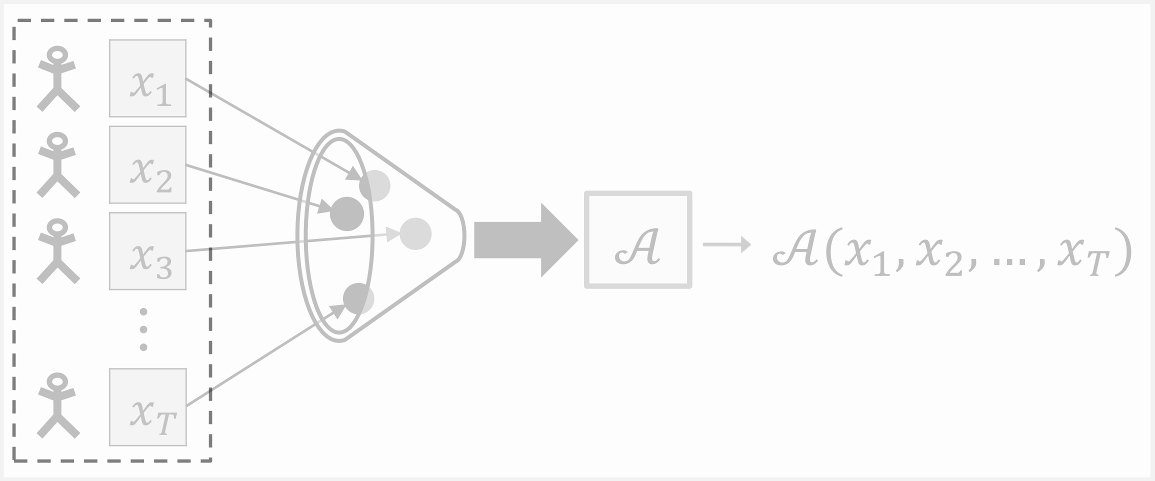
Privacy Definition

Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Batch Model [Dwork, McSherry Nissim Smith 06]



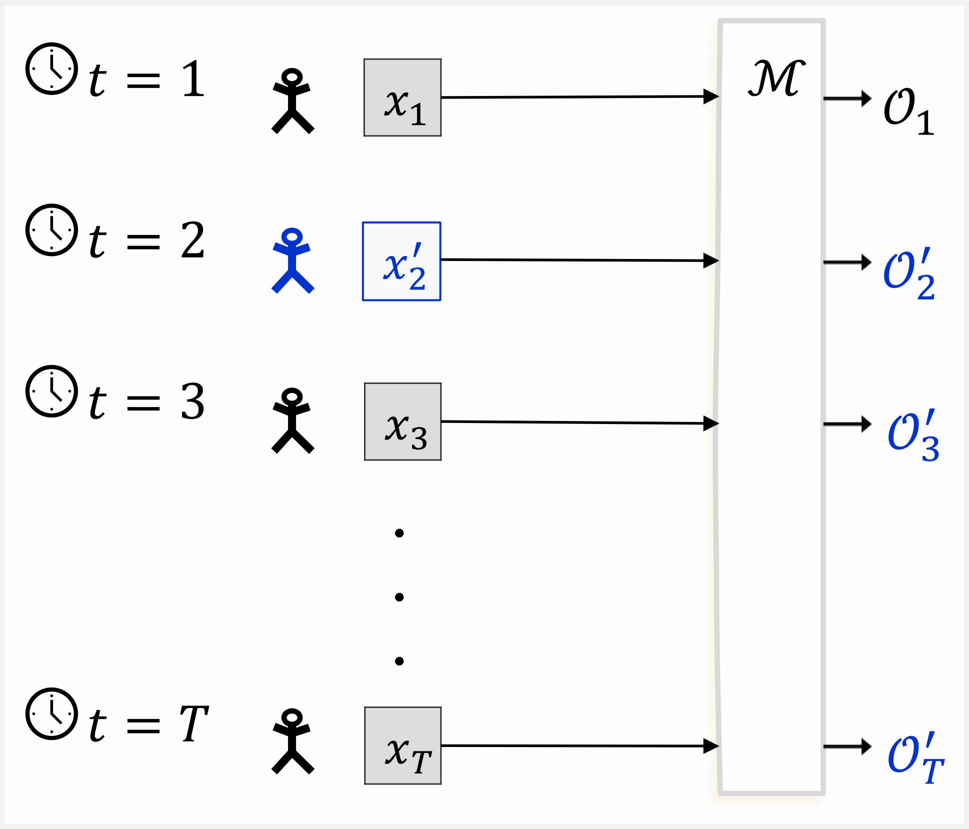
(ϵ, δ)-Differential Privacy: For all neighbors $\boldsymbol{x}, \boldsymbol{x}'$,

$$\mathcal{A}(x_1, \dots, x_t, \dots, x_n) \approx_{\epsilon,\delta} \mathcal{A}(x_1, \dots, x_t', \dots, x_n)$$

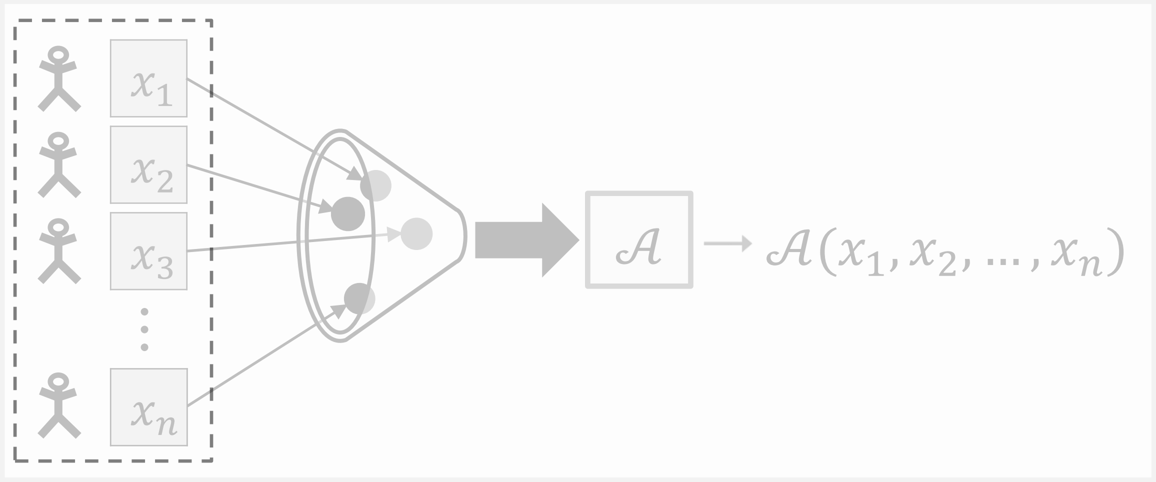
Privacy Definition: Neighboring Datasets

Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Batch Model [Dwork, McSherry Nissim Smith 06]



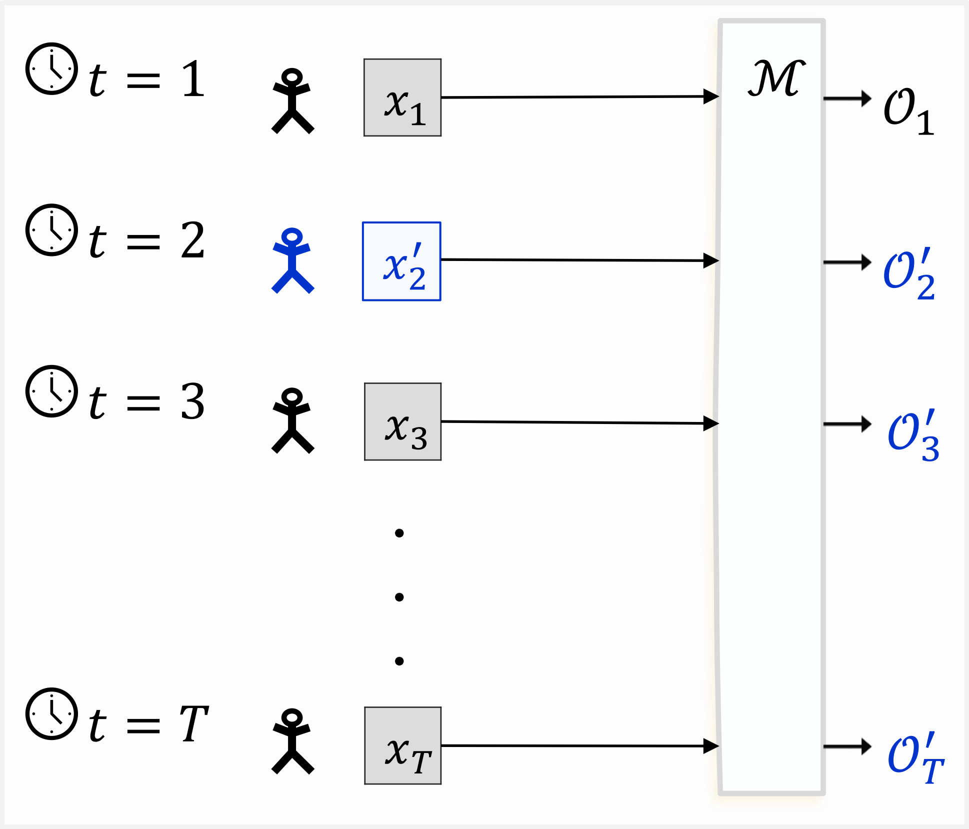
(ϵ, δ)-Differential Privacy: For all neighbors $\boldsymbol{x}, \boldsymbol{x}'$,

$$\mathcal{A}(x_1, \dots, x_t, \dots, x_n) \approx_{\epsilon,\delta} \mathcal{A}(x_1, \dots, x_t', \dots, x_n)$$

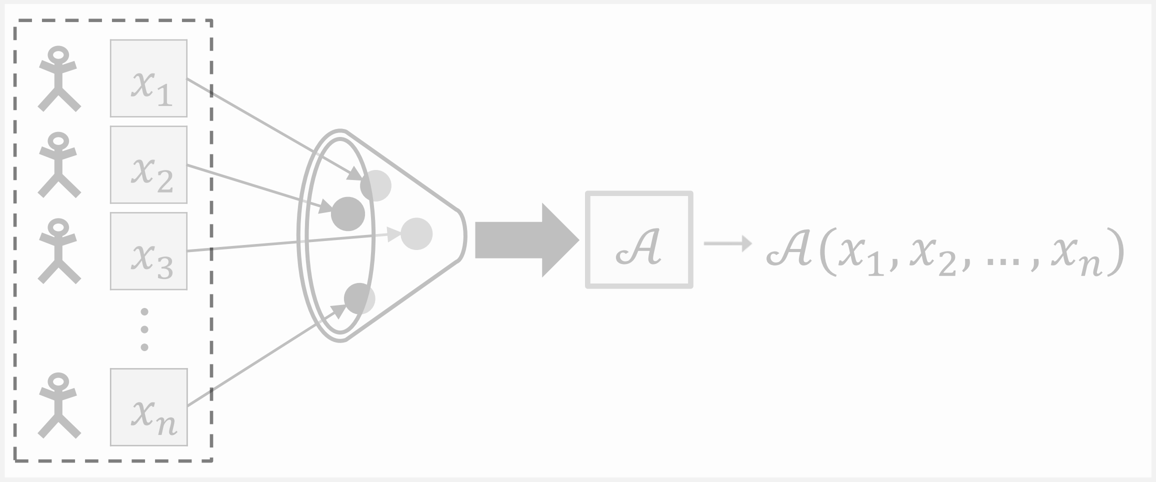
Privacy Definition

Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Batch Model [Dwork, McSherry Nissim Smith 06]



(ϵ, δ)-Differential Privacy: For all neighbors $\boldsymbol{x}, \boldsymbol{x}'$,

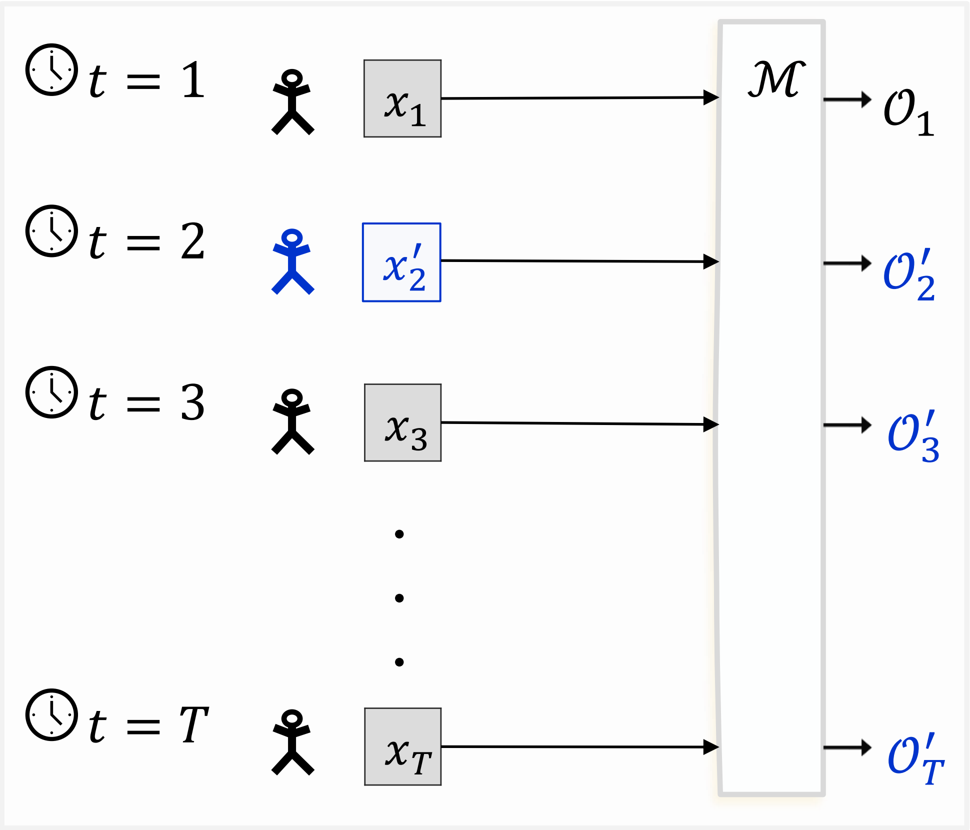
$$\mathcal{A}(x_1, \dots, x_t, \dots, x_n) \approx_{\epsilon,\delta} \mathcal{A}(x_1, \dots, x'_t, \dots, x_n)$$

$$\mathcal{M}(x_1, \dots, x_t, \dots, x_T) = (\mathcal{O}_1, \dots, \mathcal{O}_t, \dots, \mathcal{O}_T) \approx_{\epsilon,\delta} \mathcal{M}(x_1, \dots, x'_t, \dots, x_T) = (\mathcal{O}_1, \dots, \mathcal{O}'_t, \dots, \mathcal{O}'_T)$$

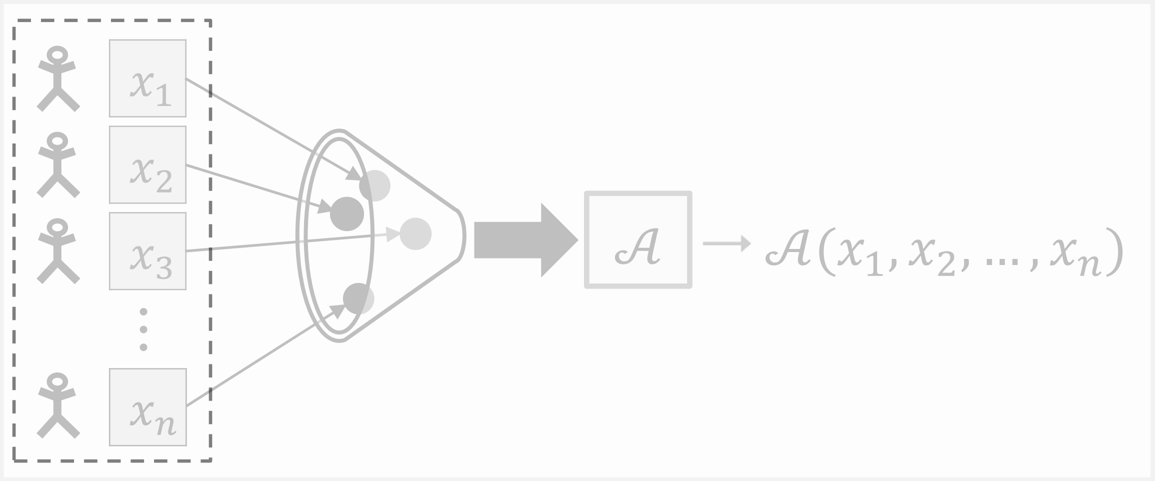
Accuracy Definition

Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Batch Model [Dwork, McSherry Nissim Smith 06]



(ϵ, δ)-Differential Privacy: For all neighbors $\boldsymbol{x}, \boldsymbol{x}'$,

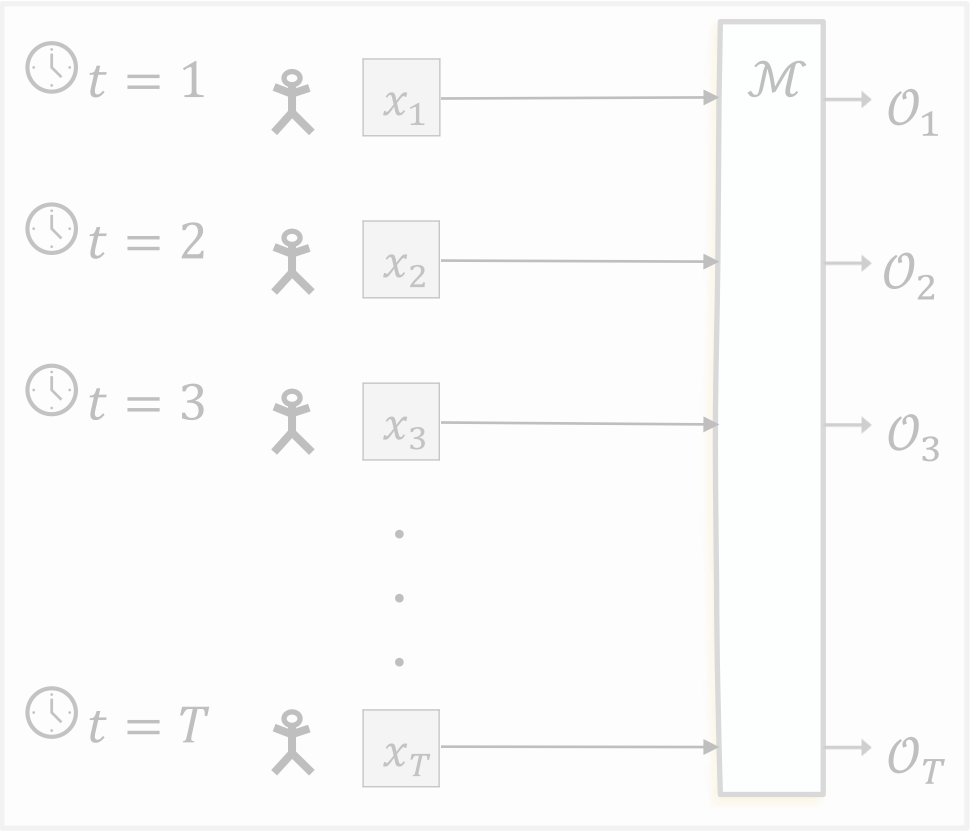
$$\mathcal{M}(x_1, \ldots, x_t, \ldots, x_T) = (\mathcal{O}_1, \ldots, \mathcal{O}_t, \ldots, \mathcal{O}_T) \approx_{\epsilon,\delta} \mathcal{M}(x_1, \ldots, x_t', \ldots, x_T) = (\mathcal{O}_1, \ldots, \mathcal{O}_t', \ldots, \mathcal{O}_T')$$

α-Accuracy:

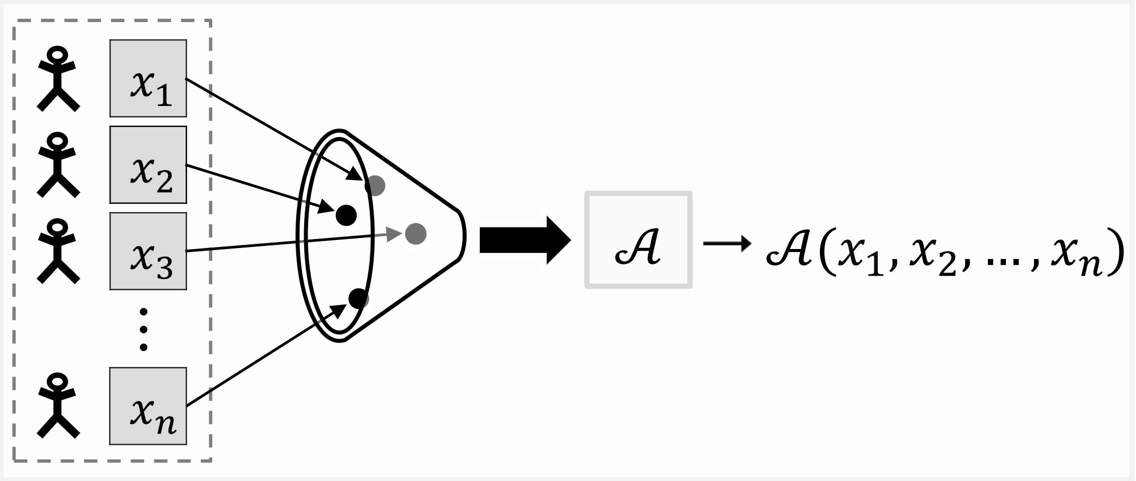
Accuracy Definition

Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Batch Model [Dwork, McSherry Nissim Smith 06]



(ϵ,δ)-Differential Privacy: For all neighbors $\boldsymbol{x}, \boldsymbol{x}'$,

$$\mathcal{M}(x_1,\ldots,x_t,\ldots,x_T) = (\mathcal{O}_1,\ldots,\mathcal{O}_t,\ldots,\mathcal{O}_T) \approx_{\epsilon,\delta} \mathcal{M}(x_1,\ldots,x_t',\ldots,x_T) = (\mathcal{O}_1,\ldots,\mathcal{O}_t',\ldots,\mathcal{O}_T')$$

α-Accuracy: For all datasets,

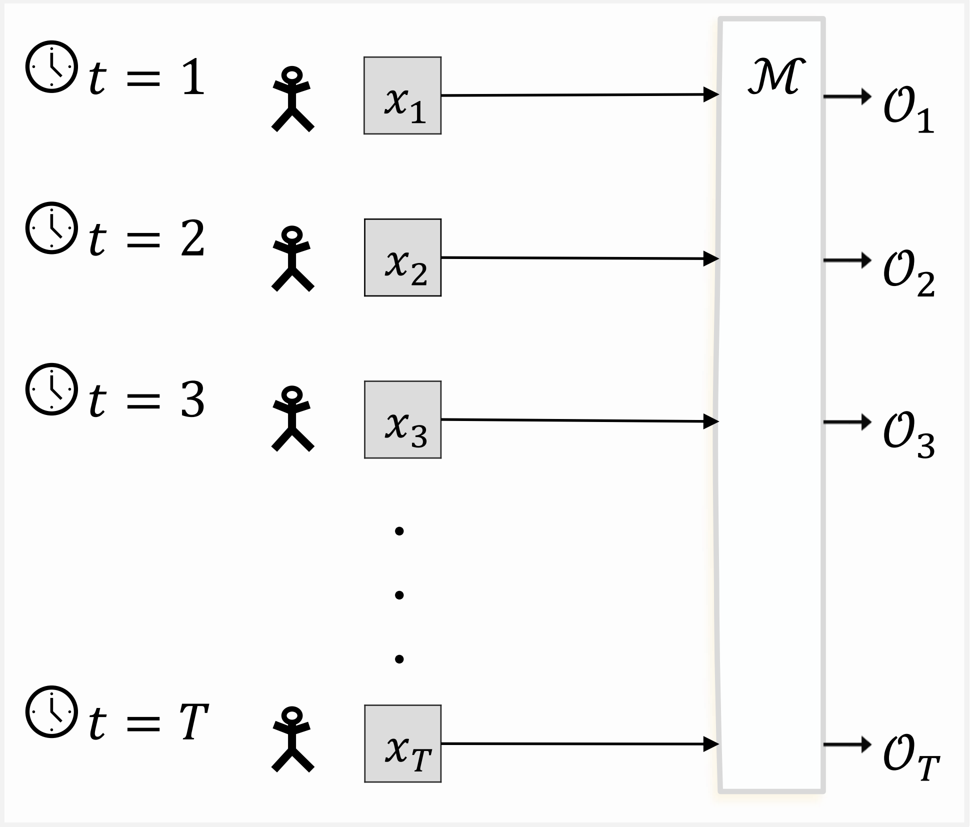
$$ERR_f[\mathcal{A}(x_1,\ldots,x_T)] \leq \alpha \quad w.p. \geq 2/3$$

$$ERR_f[\mathcal{A}(\boldsymbol{x})] = |\mathcal{A}(\boldsymbol{x}) - f(\boldsymbol{x})|$$

Accuracy Definition

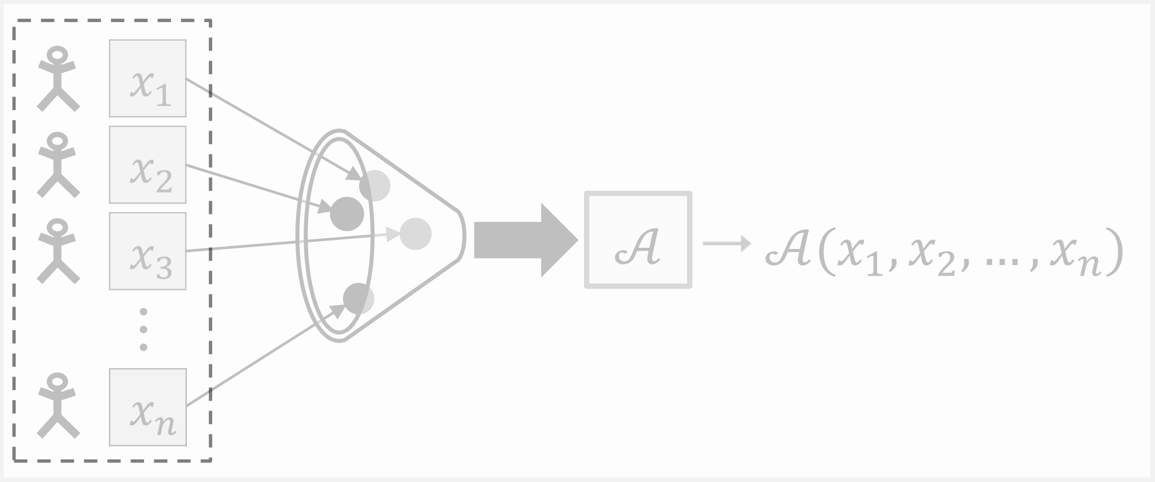
Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



$ERR_f[\mathcal{O}_t] = |\mathcal{O}_t - f(\boldsymbol{x}_{1,\dots,t})|$

Batch Model [Dwork, McSherry Nissim Smith 06]



(ϵ,δ)-Differential Privacy: For all neighbors $\boldsymbol{x}, \boldsymbol{x}'$,

$$\mathcal{M}(x_1, \dots, x_t, \dots, x_T) = (\mathcal{O}_1, \dots, \mathcal{O}_t, \dots, \mathcal{O}_T) \approx_{\epsilon,\delta} \mathcal{M}(x_1, \dots, x_t', \dots, x_T) = (\mathcal{O}_1, \dots, \mathcal{O}_t', \dots, \mathcal{O}_T')$$

α-Accuracy: For all datasets,

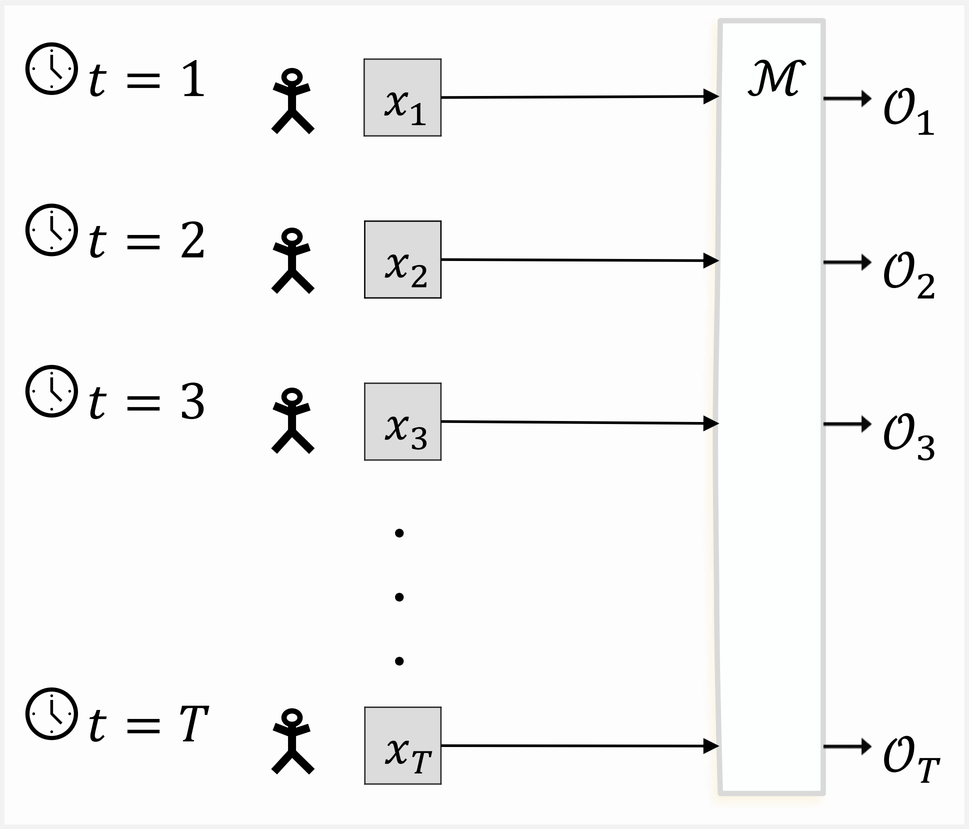
$$ERR_f[\mathcal{A}(x_1, \dots, x_T)] \le \alpha \quad w.p. \ \ge \ 2/3$$

$$MAX(ERR_f[\mathcal{O}_1], \dots, ERR_f[\mathcal{O}_T]) \le \alpha \quad w.p. \ \ge \ 2/3$$

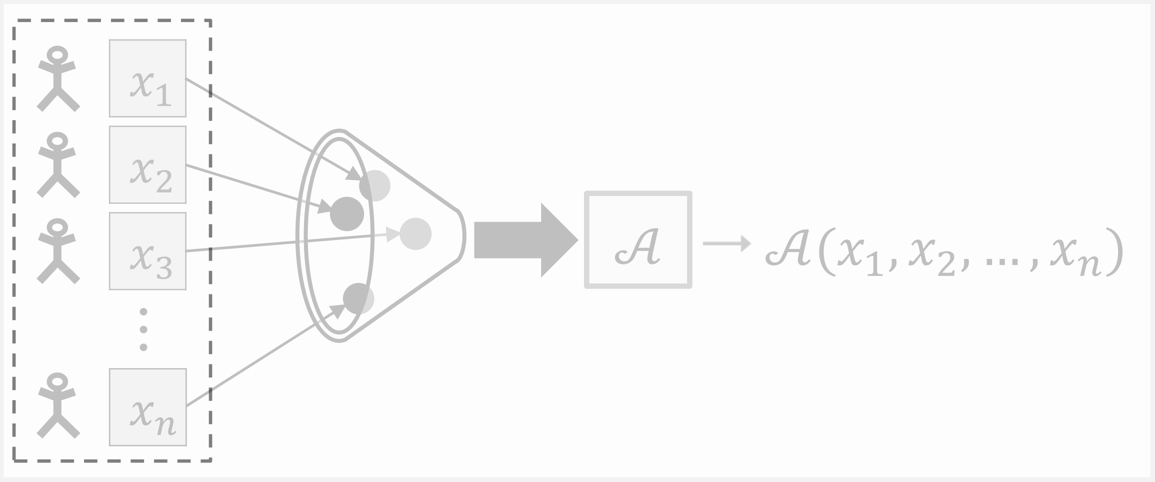
Privacy and Accuracy Definitions

Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Batch Model [Dwork, McSherry Nissim Smith 06]



(ϵ, δ)-Differential Privacy: For all neighbors $\boldsymbol{x}, \boldsymbol{x}'$,

$$\mathcal{M}(x_1, \dots, x_t, \dots, x_T) = (\mathcal{O}_1, \dots, \mathcal{O}_t, \dots, \mathcal{O}_T) \approx_{\epsilon,\delta} \mathcal{M}(x_1, \dots, x_t', \dots, x_T) = (\mathcal{O}_1, \dots, \mathcal{O}_t', \dots, \mathcal{O}_T')$$

α-Accuracy: For all datasets,

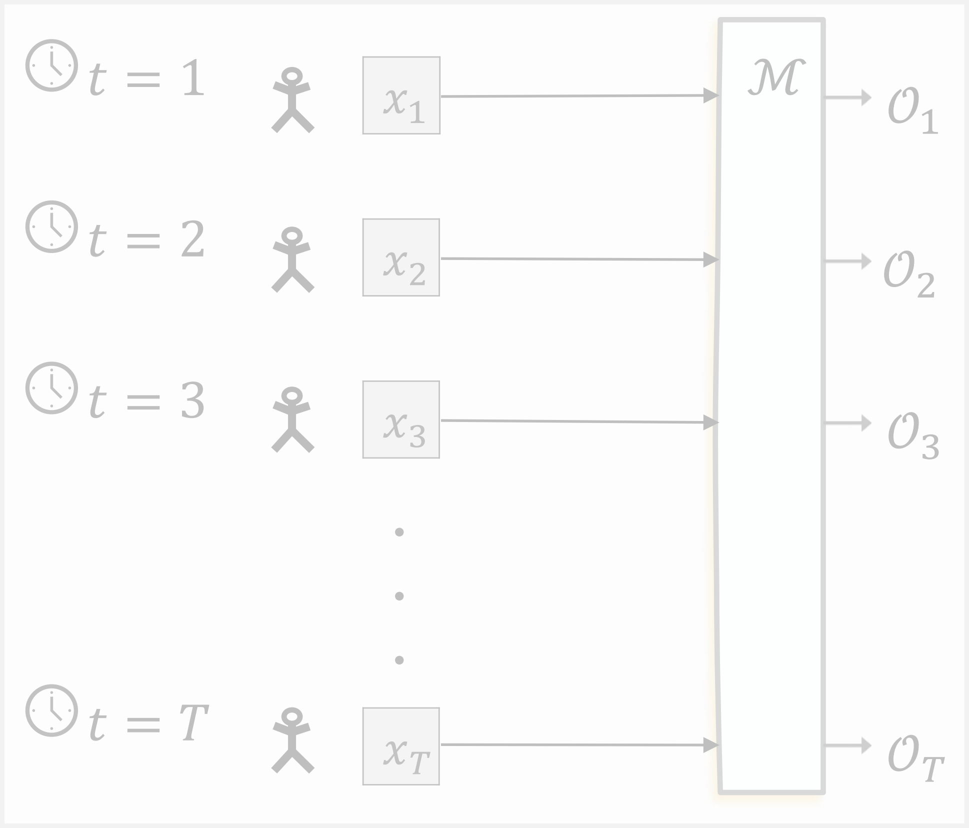
$$MAX\left(ERR_f[\mathcal{O}_1], \dots, ERR_f[\mathcal{O}_T]\right) \leq \alpha \quad w.p. \geq 2/3$$

Example Function: Summation

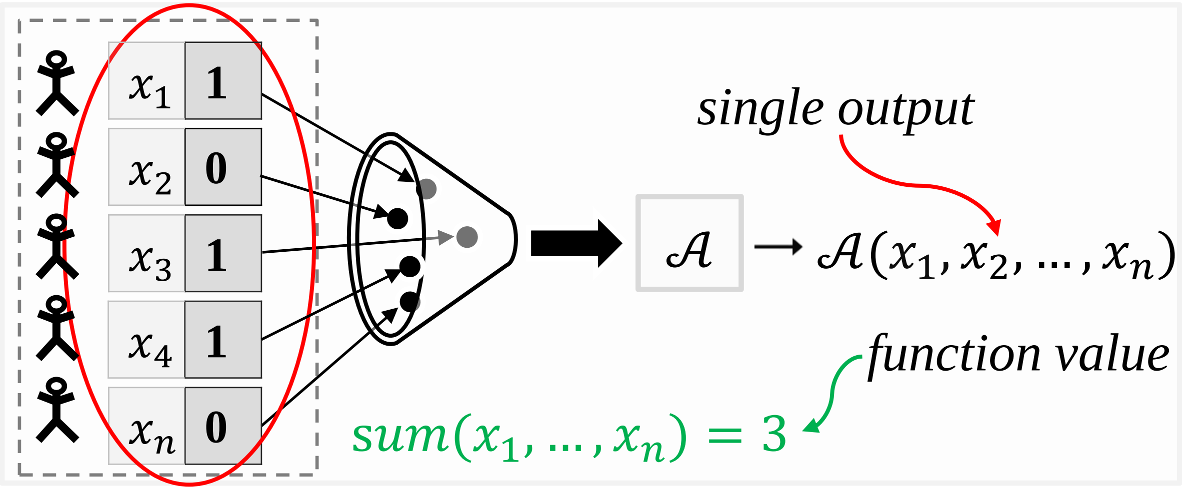
The algorithms work when $x_i \in [0,1]$

Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Batch Model [Dwork, McSherry Nissim Smith 06]



Each person's data: $x_i \in \{0,1\}$

$$sum(x_1, \dots, x_n) = \sum_{i \in [n]} x_i$$

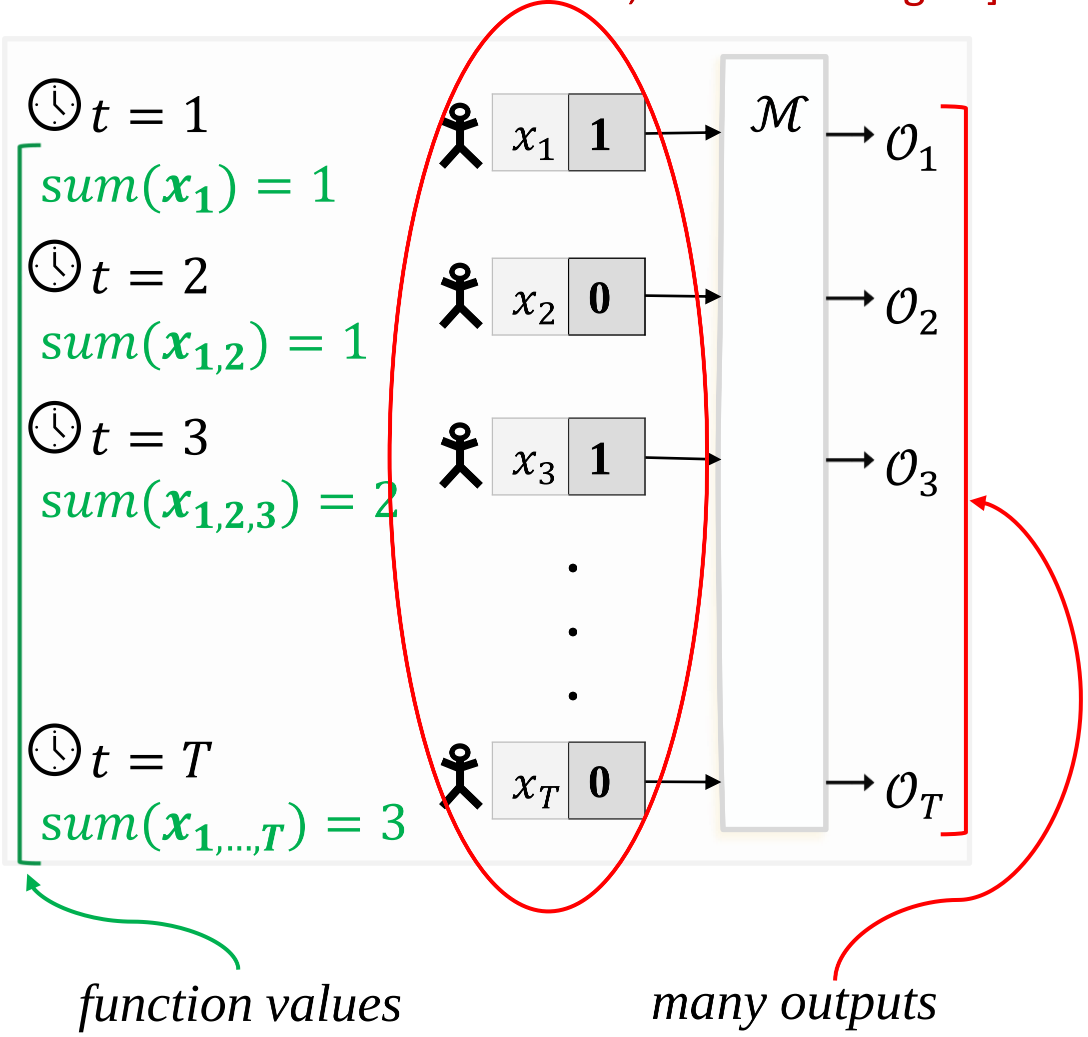
- *Batch model:*

  error $O\left(\frac{1}{\epsilon}\right)$ using Laplace mechanism.

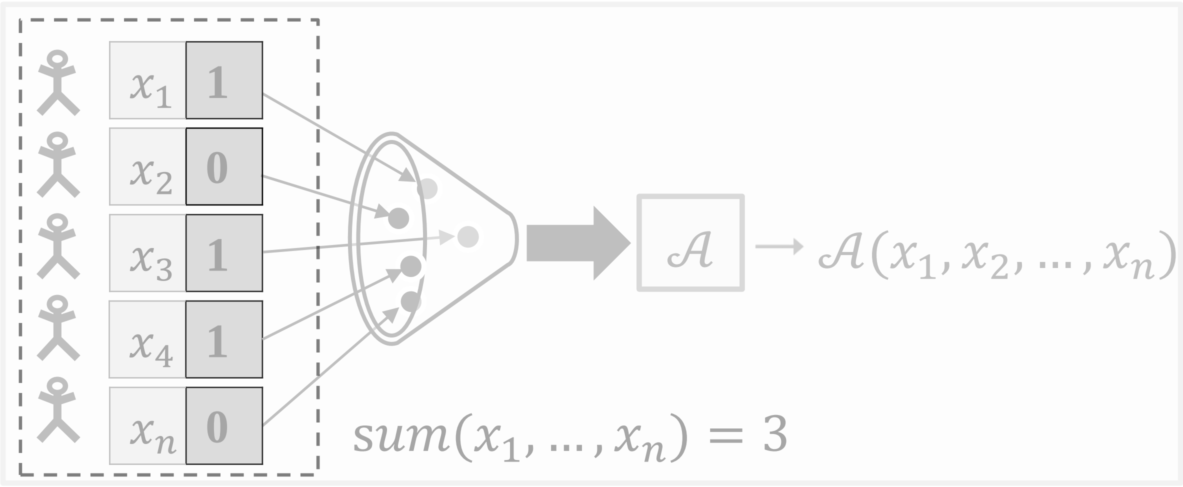
Example Function: Summation

Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Batch Model [Dwork, McSherry Nissim Smith 06]

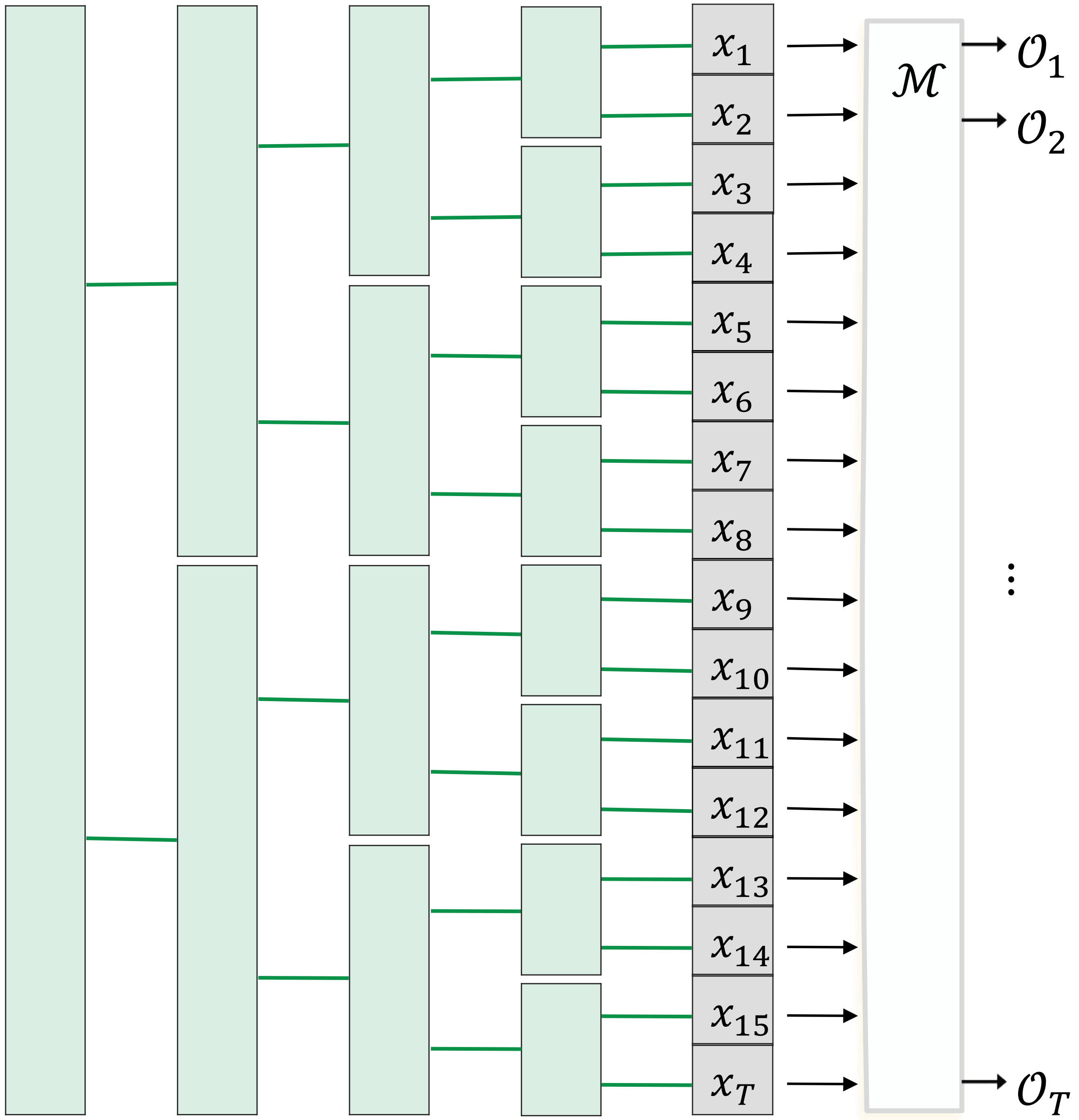


Each person's data: $x_i \in \{0,1\}$

$$sum(x_1, \dots, x_n) = \sum_{i \in [n]} x_i$$

- *Batch model:* error $O\left(\frac{1}{\epsilon}\right)$ using Laplace mechanism.
- *Continual release:* error $O\left(\frac{\log^2 T}{\epsilon}\right)$ using tree mechanism [Dwork et al., Chan et al.]

Tree Mechanism [Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Mechanism $\mathcal{M}$ *for* ***Summation***

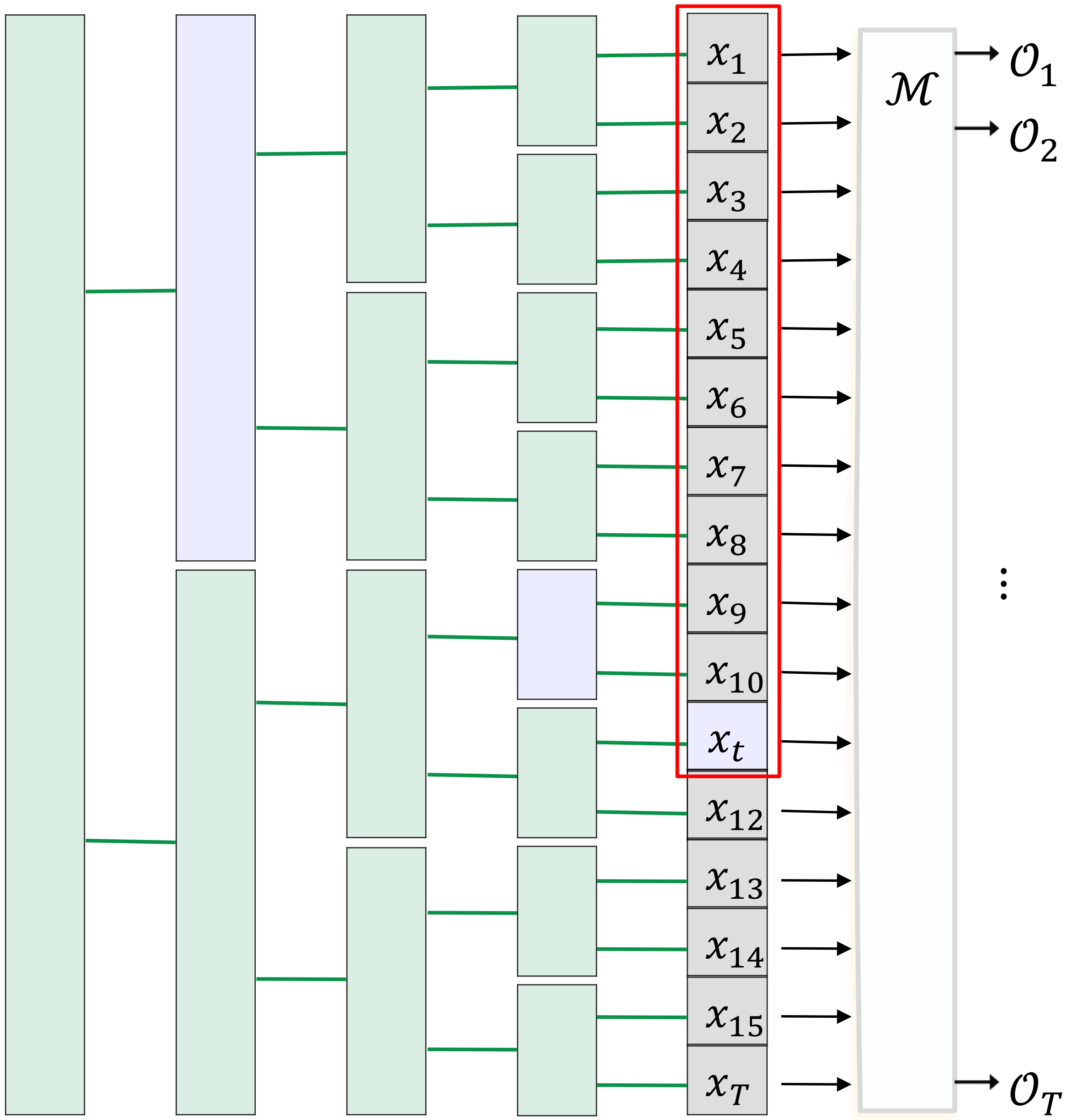
- For each interval $\boldsymbol{I}$ in the tree, publish

$$\tilde{X}_{\boldsymbol{I}} = \sum_{t \in \boldsymbol{I}} x_t + Y_{\boldsymbol{I}}$$

noise $Y_{\boldsymbol{I}} \sim Lap\left(\frac{\log_2 T}{\epsilon}\right)$

- <u>Postprocess</u>
  to estimate the sum $\sum_{i=1}^{t} x_i$ at time t:

Tree Mechanism [Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



*Mechanism $\mathcal{M}$ for **Summation***

- For each interval $\boldsymbol{I}$ in the tree, publish

$$\tilde{X}_{\boldsymbol{I}} = \sum_{t \in \boldsymbol{I}} x_t + Y_{\boldsymbol{I}}$$

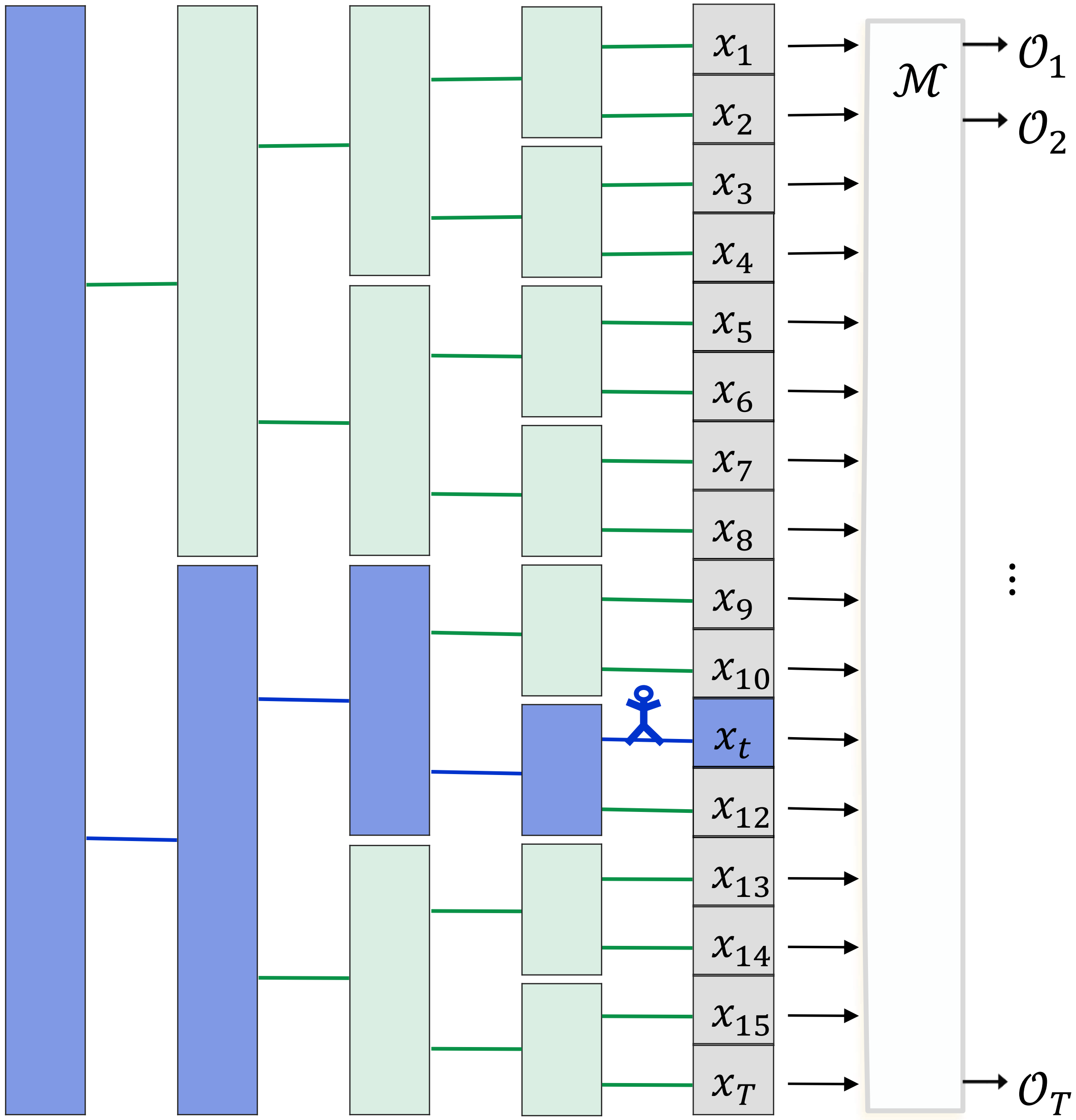
noise $Y_{\boldsymbol{I}} \sim Lap\left(\frac{\log_2 T}{\epsilon}\right)$

- Postprocess to estimate the sum $\sum_{i=1}^{t} x_i$ at time t:
  - Represent $[1, t]$ as the sum of at most $\log T$ intervals $\boldsymbol{I}$ and add estimates $\tilde{X}_{\boldsymbol{I}}$

Accuracy Analysis:

- Each output is the sum of $\leq \log T$ noisy sums $\tilde{X}_{\boldsymbol{I}}$
- Its error is the sum of $\leq \log T$ independent Laplace RVs with variance $O\left(\frac{\log^2 T}{\epsilon^2}\right)$ each.
- Variance $\sigma^2 = O\left(\frac{\log^3 T}{\epsilon^2}\right)$, so $\sigma = O\left(\frac{\log^{1.5} T}{\epsilon}\right)$
- It can be shown: max error is $O\left(\frac{\log^2 T}{\epsilon}\right)$

Tree Mechanism: Analysis



Mechanism $\mathcal{M}$ *for* ***Summation***

- For each interval $\boldsymbol{I}$ in the tree, publish

$$\tilde{X}_{\boldsymbol{I}} = \sum_{t \in \boldsymbol{I}} x_t + Y_{\boldsymbol{I}}$$

noise $Y_{\boldsymbol{I}} \sim Lap\left(\frac{\log_2 T}{\epsilon}\right)$

- Postprocess to estimate the sum $\sum_{i=1}^{t} x_i$ at time t:
  - Represent $[1, t]$ as the sum of at most $\log T$ intervals $\boldsymbol{I}$ and add estimates $\tilde{X}_{\boldsymbol{I}}$

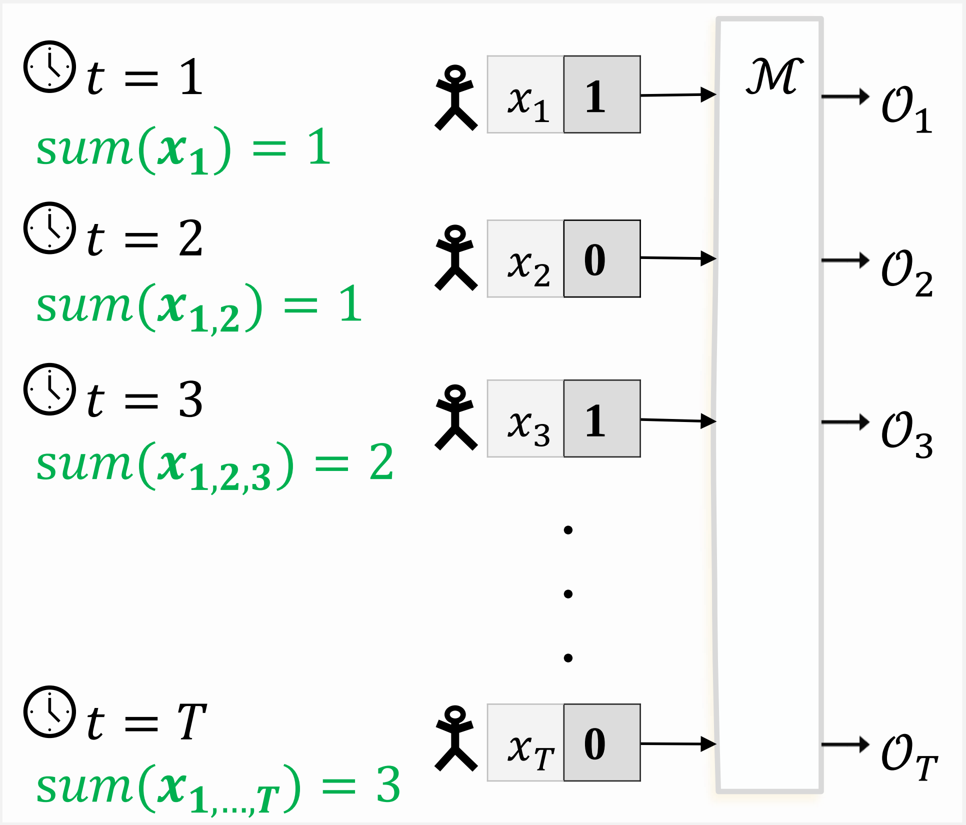
Privacy Analysis:

- Each x_t participates in $\log T$ noisy sums $\tilde{X}_{\boldsymbol{I}}$
- The vector of interval sums has sensitivity $\log T$
- By properties of Laplace mechanism and postprocessing, $\mathcal{M}$ is ϵ-differentially private

Summary of Results for Summation

Continual Release Model

[Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]



Each person's data: $x_i \in \{0,1\}$

$$sum(x_1, \ldots, x_n) = \sum_{i\in[n]} x_i$$

- *Batch model:*
  error $O\left(\frac{1}{\epsilon}\right)$ using Laplace mechanism.
- *Continual release:* error $O\left(\frac{\log^2 T}{\epsilon}\right)$ using tree mechanism [Dwork et al., Chan et al.]
  - error $\Omega\left(\frac{\log T}{\epsilon}\right)$ is necessary [Dwork et al.]

The overhead in the error in the continual release model is only polylog(T)

- Tree mechanism has been used to solve many problems, some of which don't look related to summation.
- But some problems that are closely related to summation remained unsolved.

Key Contributions of [Jain Raskhodnikova Sivakumar Smith]

$T^{1/3}$ (from $\log T$)

Algorithms for these tasks are key ingredients in DP solutions to more complex problems (e.g., synthetic data generation and high-dimensional optimization)

- First **strong** lower bounds for the continual release model

- Tight bounds for two fundamental problems

Related to summation,
but with inputs $x_1, \ldots, x_n \in \{0,1\}^d$
- *MaxSum*: *largest sum in one coordinate*
- *SumSelect*: *index of the coordinate with largest sum*

- New sequential embedding technique

- Formalization of the continual release model with adaptively selected inputs

Related Work

- Introduced the continual release model, designed the tree mechanism for summation
  [Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]

Applications of the tree mechanism

- differentially private online learning
  [Jain Kothari Thakurta 12, Smith Thakurta 13, Agarwal Singh 17,…]
- weighted sums and sums of real-valued data
  [Bolot Fawaz Muthukrishnan Nikolov Taft 13, Perrier Asghar Kaafar 19]
- interval and rectangle queries, refinement of the binary tree mechanism [Dwork Naor Reingold Rothblum 15]

Alternatives to the tree mechanism

- Applications to (practical) online learning
  [Kairouz McMahan Song Thakkar Thakurta Xu 21, Denisov McMahan Rush Smith Thakurta 22]

- graph problems [Fichtenberger Henzinger Ost '21]

Variants of $\mathbf{MaxSum}/\mathbf{SumSelect}$ in different DP models

- Central model
  [Steinke Ullman 17, Durfee Rogers 19]
- Local model [Kasiviswanathan Lee Nissim Raskhodnikova Smith 08, Duchi Jordan Wainwright 13, Ullman 17, Edmonds Nikolov Ullman 20,…]
- Shuffle and pan-private models
  [Cheu Ullman 21]
- Continual release ($\mathbf{SumSelect}$, focusing on empirical performance) [Cardoso Rogers 22]

Error Bounds in [Jain Raskhodnikova Sivakumar Smith]

$\left(1, o\left(\frac{1}{T}\right)\right)$-*DP*

	Batch Model	Continual Release	
		LOWER BOUNDS	**UPPER BOUNDS**
Summation	$\Theta(1)$ [Dwork McSherry Nissim Smith 06]	$\Omega(\log T)$ [Dwork Naor Pitassi Rothblum 10]	$O(\log^2 T)$ [Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]
$MaxSum(d)$	$\Theta(1)$ [Dwork McSherry Nissim Smith 06]	$\tilde{\Omega}(\min(T^{1/3}, \sqrt{d}))$	$\tilde{O}(\min(T^{1/3}, \sqrt{d} \log T))$
$SumSelect(d)$	$\Theta(\log d)$ [McSherry Talwar 07]	$\tilde{\Omega}(\min(T^{1/3} \log d, \sqrt{d}))$	$\tilde{O}(\min(T^{1/3} \log d, \sqrt{d} \log T))$

1. Lower bounds hold for nonadaptively selected inputs
2. Matched by algorithms that work against adaptively selected inputs
   - Formalization of continual release model with adaptively selected inputs
3. Techniques work for pure DP and approximate DP

MaxSum: Example

x_1	1	1	0	0
x_2	1	0	1	1
	0	0	0	0
	0	0	1	1
	0	0	1	0
	0	0	0	0
⋮	0	1	0	0
	0	1	0	0
	0	1	0	0
	0	1	0	0
	0	0	0	1
	0	0	0	1
	0	0	0	1
x_T	0	0	0	1
	2	**5**	**3**	**6**

Each person's data: $x_i \in \{0,1\}^d$

$$MaxSum(x_1, \dots, x_n) = \max_{j \in [d]} \sum_{i \in [n]} x_i^j$$

$MaxSum = 6$

Hardness of MaxSum: Intuition

x_1	**1**	**1**	0	0
x_2	**1**	0	**1**	**1**
	0	0	0	0
	0	0	**1**	**1**
	0	0	**1**	0
	0	0	0	0
⋮	0	**1**	0	0
	0	**1**	0	0
	0	**1**	0	0
	0	**1**	0	0
	0	0	0	**1**
	0	0	0	**1**
	0	0	0	**1**
x_T	0	0	0	**1**

$\xrightarrow{MaxSum}$ 6

Batch model:
learn information about one coordinate sum

Hardness of MaxSum: Intuition

$t = 2$

x_1 →
x_2 →

1	**1**	0	0
1	0	**1**	**1**
0	0	0	0
0	0	**1**	**1**
0	0	**1**	0
0	0	0	0
0	**1**	0	0
0	**1**	0	0
0	**1**	0	0
0	**1**	0	0
0	0	0	**1**
0	0	0	**1**
0	0	0	**1**
0	0	0	**1**

$\xrightarrow{MaxSum} 2$

Continual release model:
learn information
about many coordinates!

Hardness of MaxSum: Intuition

$t = 2$ — x_1, x_2 ⋮

$t = 5$ — x_5

1	**1**	0	0
1	0	**1**	**1**
0	0	0	0
0	0	**1**	**1**
0	0	**1**	0
0	0	0	0
0	**1**	0	0
0	**1**	0	0
0	**1**	0	0
0	**1**	0	0
0	0	0	**1**
0	0	0	**1**
0	0	0	**1**
0	0	0	**1**

$MaxSum$ ⇒ 2

$MaxSum$ ⇒ 3

Continual release model:
learn information
about many coordinates!

Hardness of MaxSum: Intuition

Continual release model:
learn information about many coordinates!

	x_1	1	1	0	0	
$t = 2$	x_2	1	0	1	1	$MaxSum$ → 2
⋮		0	0	0	0	
		0	0	1	1	
$t = 5$	x_5	0	0	1	0	$MaxSum$ → 3
		0	0	0	0	
		0	1	0	0	
⋮		0	1	0	0	
		0	1	0	0	
$t = 10$	x_{10}	0	1	0	0	$MaxSum$ → 5
		0	0	0	1	
		0	0	0	1	
		0	0	0	1	
		0	0	0	1	

Hardness of MaxSum: Intuition

	x_1	**1**	**1**	0	0	
$t = 2$	x_2	**1**	0	**1**	**1**	$MaxSum$ ⇒ 2
⋮		0	0	0	0	
		0	0	**1**	**1**	
$t = 5$	x_5	0	0	**1**	0	$MaxSum$ ⇒ 3
		0	0	0	0	
		0	**1**	0	0	
⋮		0	**1**	0	0	
		0	**1**	0	0	
$t = 10$	x_{10}	0	**1**	0	0	$MaxSum$ ⇒ 5
		0	0	0	**1**	
⋮		0	0	0	**1**	
		0	0	0	**1**	
$t = 14$	x_{14}	0	0	0	**1**	$MaxSum$ ⇒ 6

Continual release model:
learn information
about many coordinates!

Key Contributions of [Jain Raskhodnikova Sivakumar Smith]

$T^{1/3}$ (from $\log T$)

- First **strong** lower bounds for the continual release model
- Tight bounds for two fundamental problems
- New sequential embedding technique
- Formalization of the continual release model with adaptively selected inputs

Related to summation,
but with inputs $x_1, \ldots, x_n \in \{0,1\}^d$

- *MaxSum*: *largest sum in one coordinate*
- *SumSelect*: *index of the coordinate with largest sum*

Lower Bound: Key Idea

Design a reduction
from
releasing all coordinate sums in the batch model
to
releasing $MaxSum$ in the continual release model.

Releasing All Coordinates Sums is Hard in the Batch Model

x_1	**1**	**1**	0	0
x_2	**1**	0	**1**	**1**
	0	0	0	0
	0	0	**1**	**1**
	0	0	**1**	0
	0	0	0	0
⋮	0	**1**	0	0
	0	**1**	0	0
	0	**1**	0	0
	0	**1**	0	0
	0	0	0	**1**
	0	0	0	**1**
	0	0	0	**1**
x_n	0	0	0	**1**
	2	**5**	**3**	**6**

$$ERR[\mathcal{A}] = \left|\mathcal{A}(\boldsymbol{x}_{1,\dots,n}) - \sum_{i\in[n]} x_i\right|_\infty$$

Theorem [Bun Ullman Vadhan 18]

Every $\left(1, o\left(\frac{1}{n}\right)\right)$-DP algorithm for $CoordSums_d$ has error at least $\Omega(\min(\sqrt{d}, n))$.

Lower Bound: Key Idea

Goal: *Algorithm for* $CoordSums_d$ *using continual release mechanism* $\mathcal{M}$ for $MaxSum$

Design a reduction
from
releasing $CoordSums$ *in the batch model*
to
releasing $MaxSum$ *in the continual release model*.

- We embed an instance of $CoordSums$ in an instance of $MaxSum$
- Then add to the stream to ensure we can extract one coordinate sum at a time

Overview of the Reduction

Goal: *Algorithm for* $\mathit{CoordSums}_d$ *using continual release mechanism* $\mathcal{M}$ for MaxSum

2dn time steps total

d columns; each block has n rows. The stream is fed into $\mathcal{M}$.

					Output of $\mathcal{M}$
Stream dataset for $\mathit{CoordSums}_d$ (n)	1	1	0	0	
	1	0	1	1	
	0	0	0	0	$\mathcal{O}_n$
Boost coordinate 1 (n)	1	0	0	0	
	1	0	0	0	
	1	0	0	0	$\mathcal{O}_{2n}$
Even out other coordinates (n)	0	1	1	1	
	0	1	1	1	
	0	1	1	1	$\mathcal{O}_{3n}$
Boost coordinate 2 (n)	0	1	0	0	
	0	1	0	0	
	0	1	0	0	$\mathcal{O}_{4n}$
Even out other coordinates (n)	1	0	1	1	
	1	0	1	1	
	1	0	1	1	$\mathcal{O}_{5n}$
⋮					

$\mathcal{O}_{2n}$: Estimate the sum in coordinate 1

$\mathcal{O}_{4n}$: Estimate the sum in coordinate 2

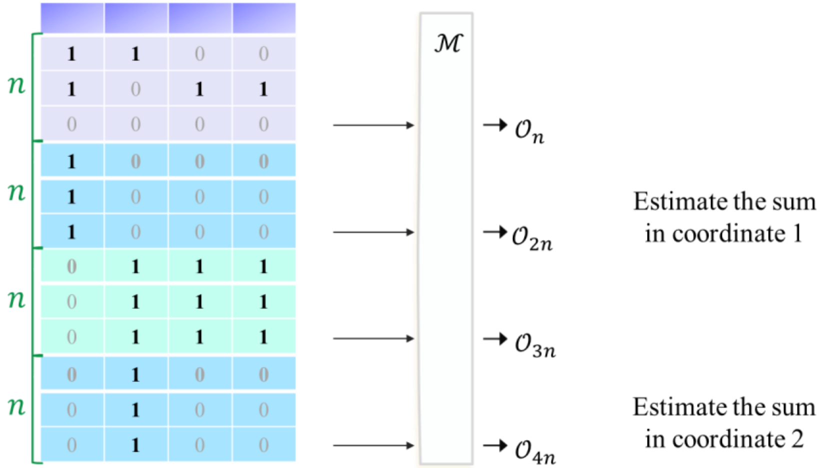
Lower Bound for *MaxSum*

Input dataset

d (columns), n (rows)

1	1	0	0
1	0	1	1
0	0	0	0

Algorithm $\mathcal{A}$ for $CoordSums_d$



d coordinate sums

Max error α

$T \asymp dn$ time steps

Max error α

Theorem [BUV18]. Every $\left(1, o\left(\frac{1}{n}\right)\right)$-DP algorithm for $CoordSums_d$ has error $\Omega(\min(\sqrt{d}, n))$

$d = T^{2/3}, n = T^{1/3}$

Error $\alpha = \Omega(T^{1/3})$

Error Bounds in **[Jain Raskhodnikova Sivakumar Smith]**

$\left(1, o\left(\frac{1}{T}\right)\right)$-*DP*

	Batch Model	**Continual Release**	
		LOWER BOUNDS	**UPPER BOUNDS**
Summation	$\Theta(1)$ [Dwork McSherry Nissim Smith 06]	$\Omega(\log T)$ [Dwork Naor Pitassi Rothblum 10]	$O(\log^2 T)$ [Dwork Naor Pitassi Rothblum 10, Chan Shi Song 10]
$MaxSum(d)$	$\Theta(1)$ [Dwork McSherry Nissim Smith 06]	$\tilde{\Omega}(\min(T^{1/3}, \sqrt{d}))$	$\tilde{O}(\min(T^{1/3}, \sqrt{d} \log T))$
$SumSelect(d)$	$\Theta(\log d)$ [McSherry Talwar 07]	$\tilde{\Omega}(\min(T^{1/3} \log d, \sqrt{d}))$	$\tilde{O}(\min(T^{1/3} \log d, \sqrt{d} \log T))$

1. Lower bounds hold for nonadaptively selected inputs
2. Matched by algorithms that work against adaptively selected inputs
   - Formalization of continual release model with adaptively selected inputs
3. Techniques work for pure DP and approximate DP

Open Questions and Directions

- Can we characterize problems in terms of how much harder they are in the continual release model than in the batch model?
  - Which sensitivity-1 functions require poly(d) error in the continual release model?
- Do our lower bounds for *SumSelect* imply hardness for online learning?
- Are there connections between continual release and learning that do not go via online learning?
- Better understanding of continual release with adaptively selected streams.