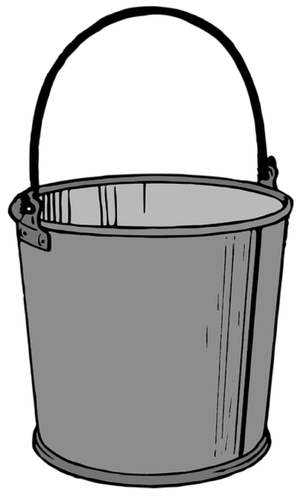
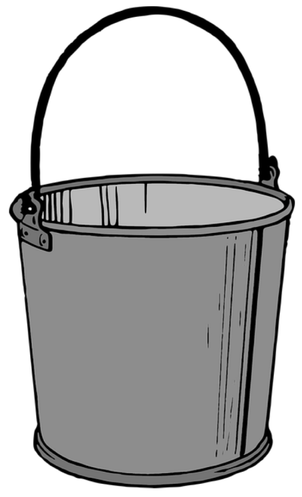


Tight on budget?



Tight Bounds for r-Fold Approximate Differential Privacy

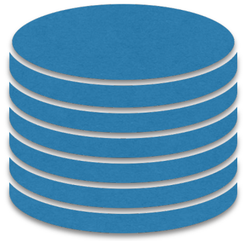
Sebastian Meiser (UCL), Esfandiar Mohammadi (ETH Zürich)

16 October 2018

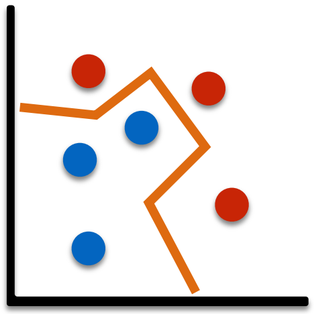
Competing Utility & Privacy Requirements

In many applications competing utility and privacy requirements

– Statistical queries:
database queries while protecting entries of individuals



– Private Machine Learning (ML):
queries to ML model while protecting sensitive training data

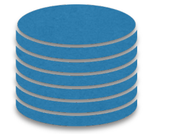


High exposure threatens privacy

In practice: privacy under continual observation / high exposure

– Statistical queries:
high nr. of database queries

– Private ML:
high nr. of queries to ML model, or
high nr. of training steps on sensitive data



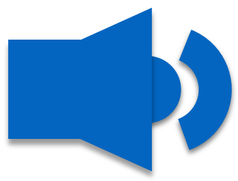
High exposure threatens privacy

Composition and differential privacy

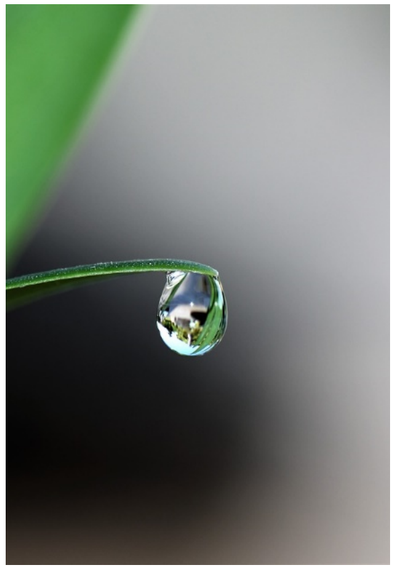
Continual observation / high exposure → leakage amplifies

How can we counter that?

Noise



Leakage (1 step)



Leakage (overall)

Composition and differential privacy

Continual observation / high exposure → leakage amplifies

How can we counter that?

- more noise to achieve privacy under continual observation
- more noise renders result of the computation less accurate

Noise

Leakage (1 step)

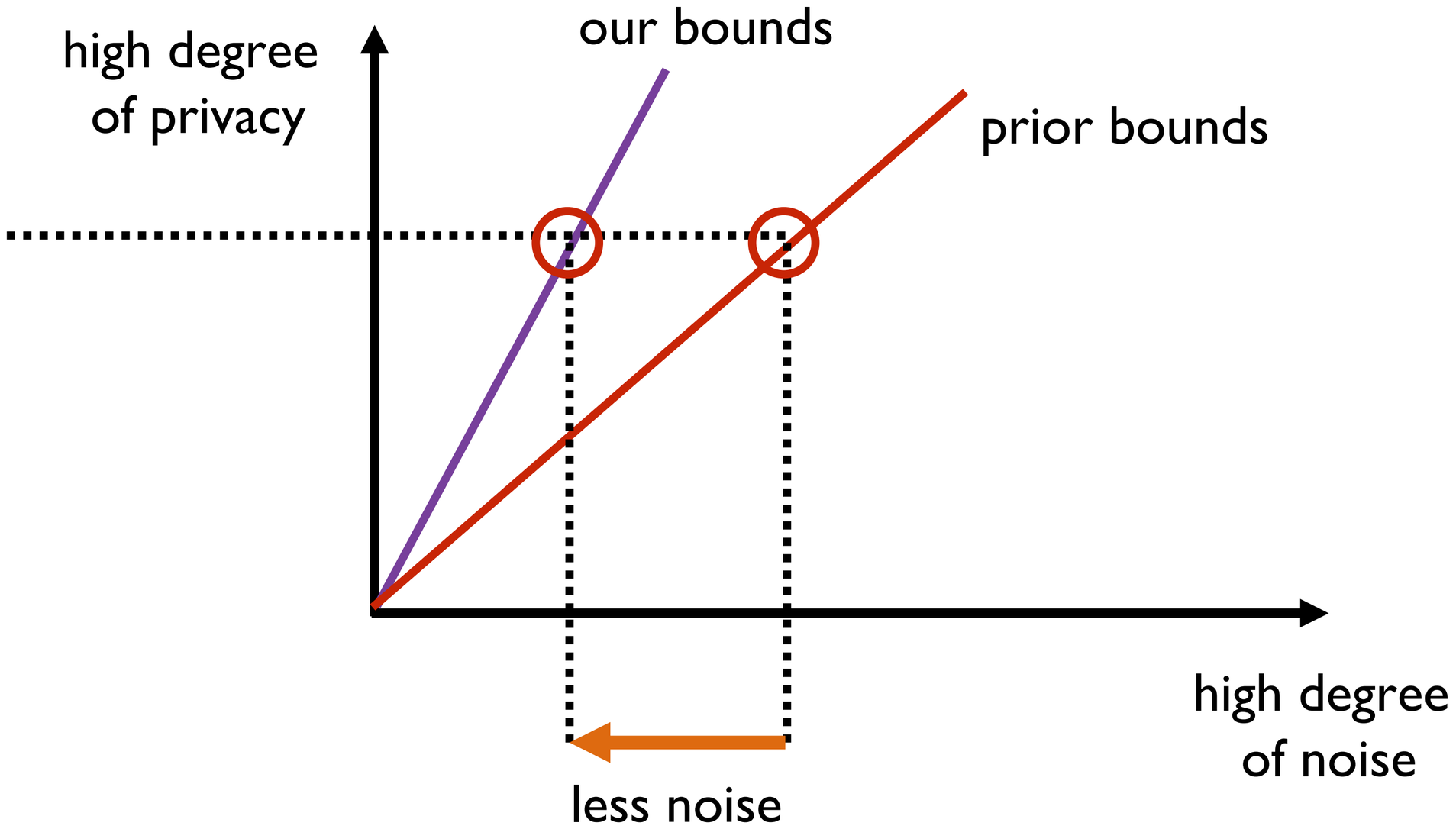
(water vapor)

Leakage (overall)

Our work

Previous bounds for approximate differential privacy (ADP) [widely used privacy notion] under continual observation not tight

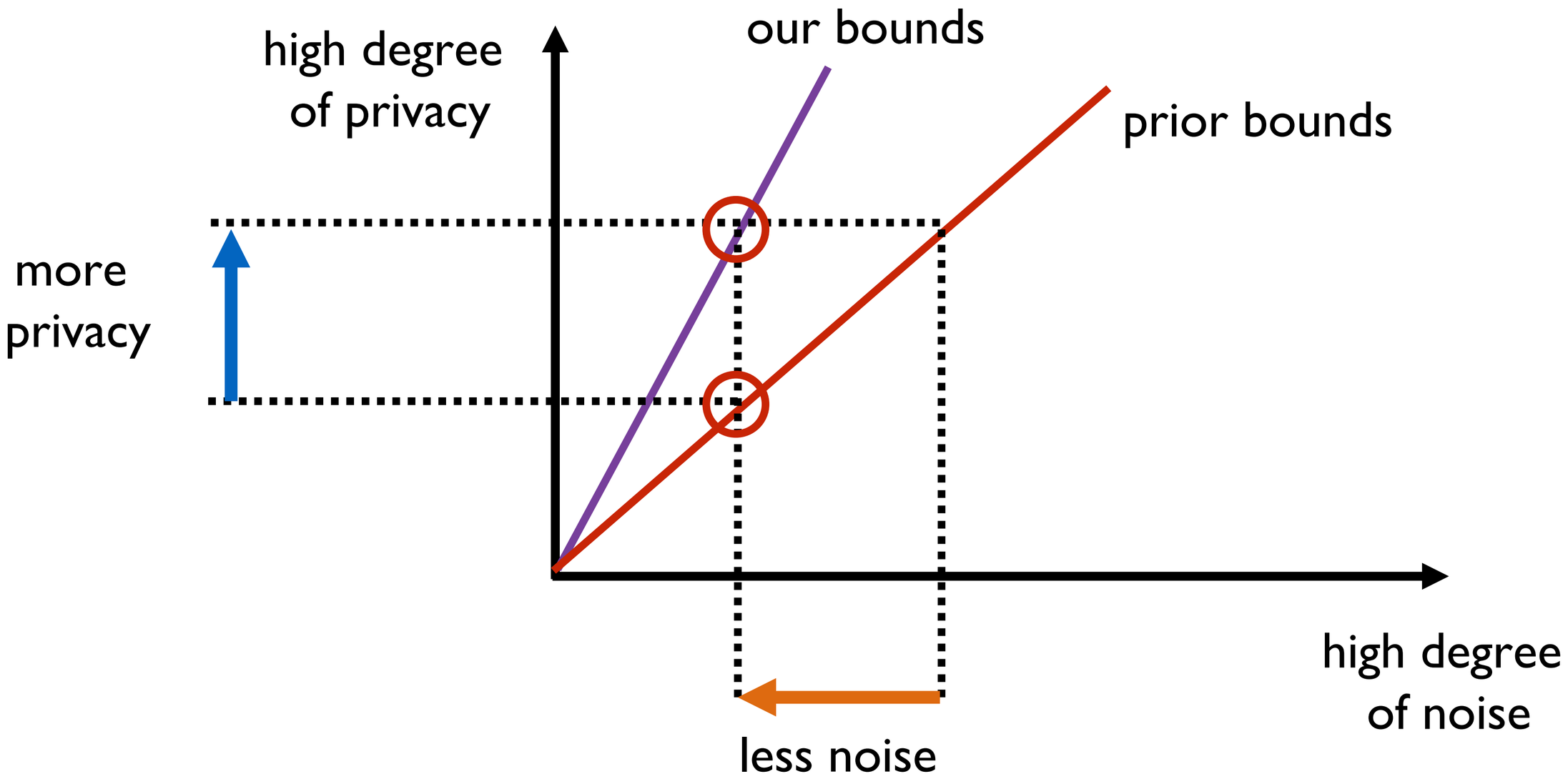
Our work: tight bounds for ADP under continual observation



Our work

Previous bounds for approximate differential privacy (ADP) [widely used privacy notion] under continual observation not tight

Our work: tight bounds for ADP under continual observation



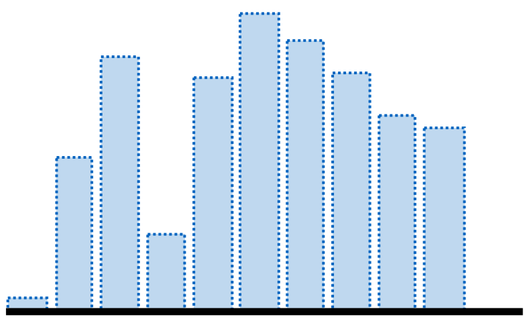
Simple running example: A private mechanism

Consider the following mechanism:

- (Secret) Input: Is Esfandiar or Sebastian watching TV?

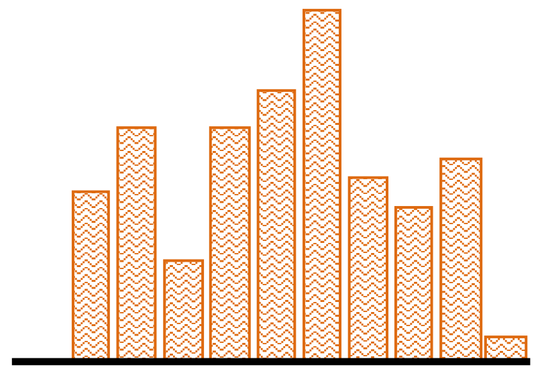
Output: A TV channel.

- The probabilities depend on who is watching.



channels

Sebastian

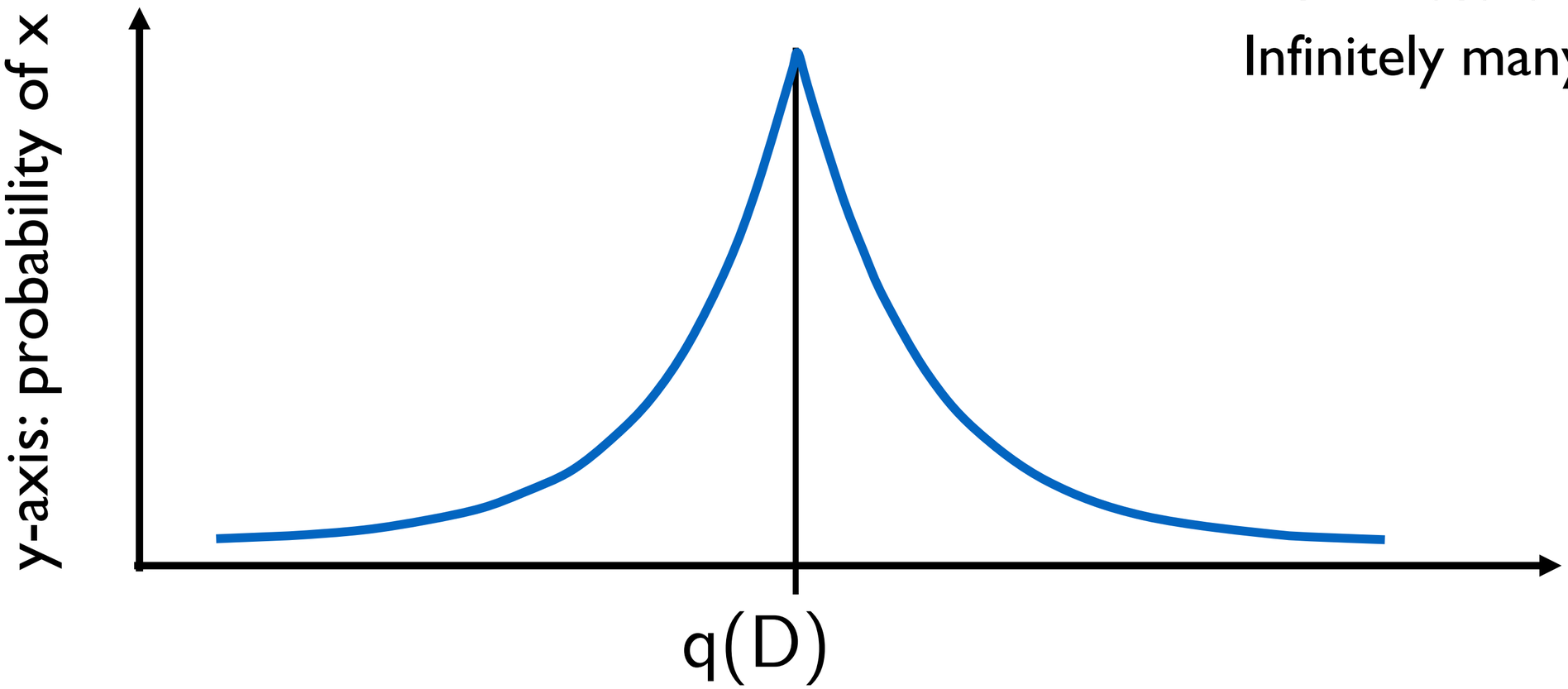


channels

Esfandiar

Why only analyze one pair of distributions?

Input: database D, query q

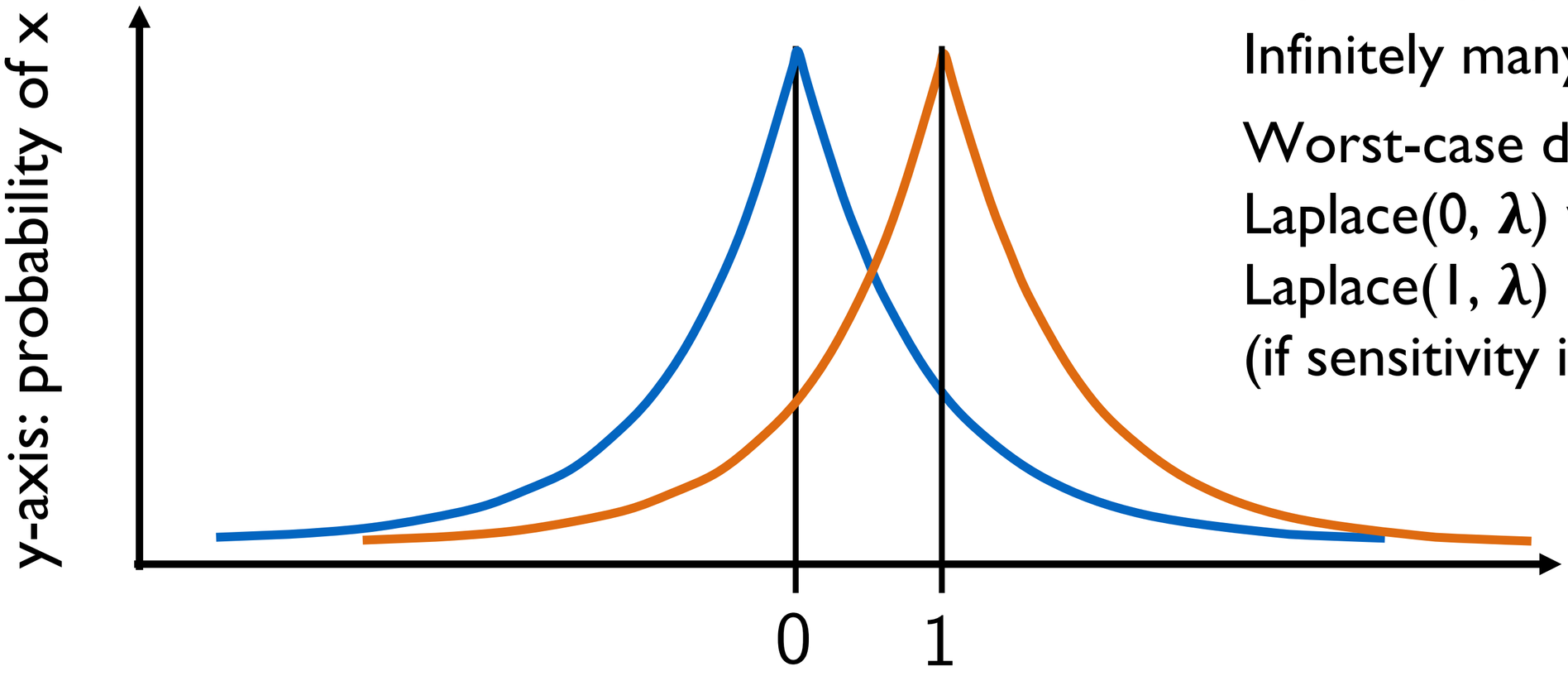


Add λ-Laplace noise
to the query-result q(D)
Laplace(q(D), λ)

Infinitely many distributions

Why only analyze one pair of distributions?

Input: database D, query q



Add λ-Laplace noise
to the query-result q(D)
Laplace(q(D), λ)

Infinitely many distributions

Worst-case distributions:
Laplace(0, λ) vs
Laplace(1, λ)
(if sensitivity is 1)

(Approximate) Differential Privacy (for the 4th time?)

Definition: For all pairs x_0, x_1 with $|f(x_0) - f(x_1)| \leq 1$, for all sets S,

$$\Pr[\text{Seb} \in S] \leq e^{\varepsilon} \cdot \Pr[\text{Esf} \in S] + \delta$$

ε-δ- Graph

For each ε there is a minimal $\delta(\varepsilon)$

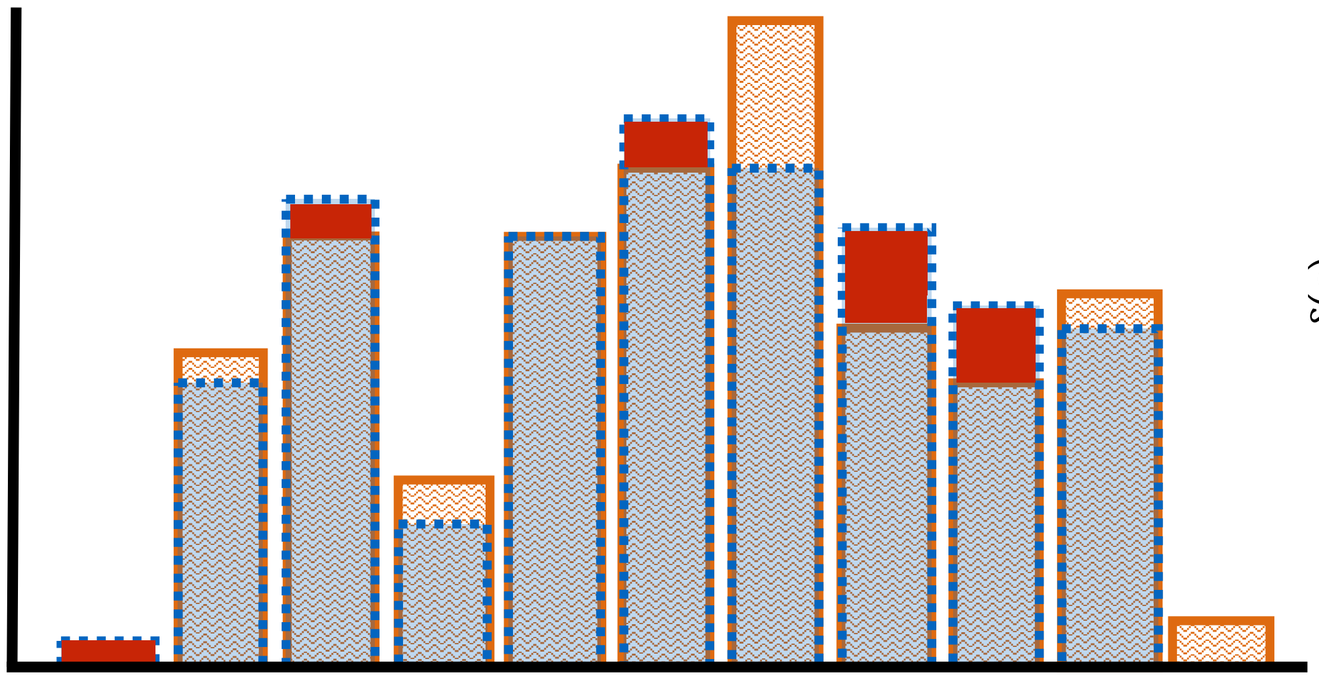
$\varepsilon =$ 0

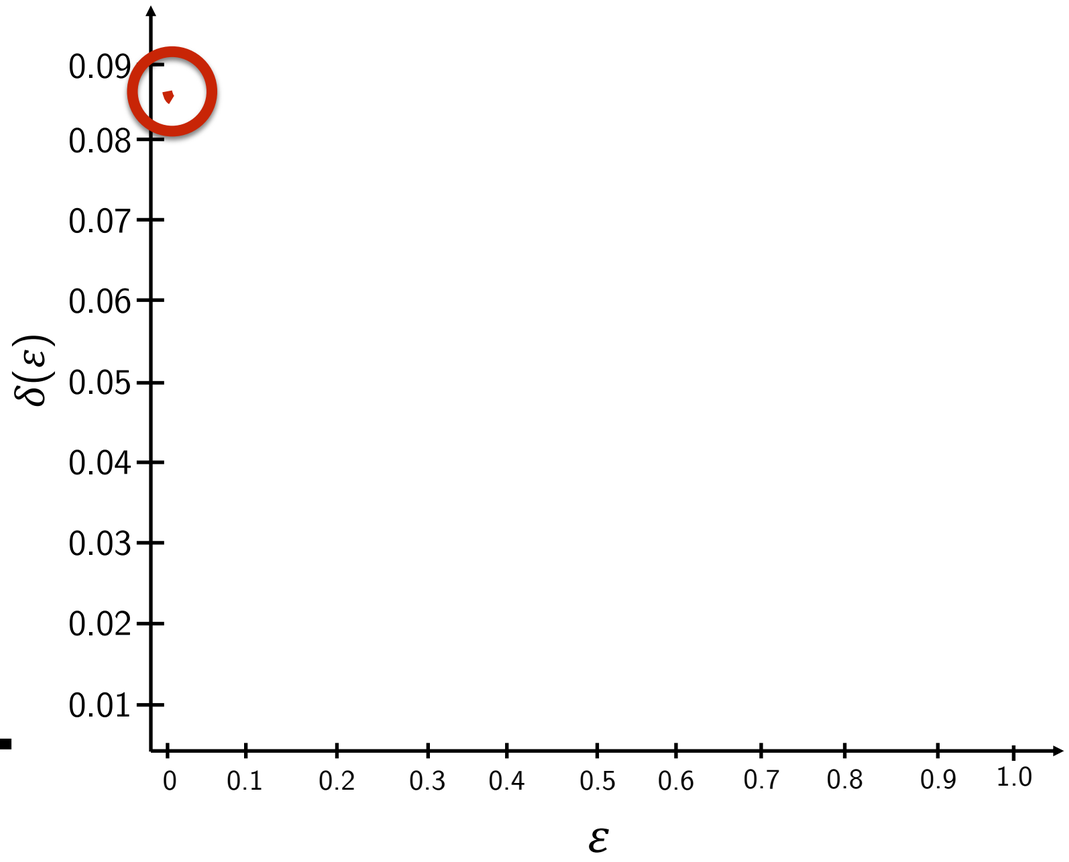
$e^{\varepsilon} =$ 1

A vs $e^{\varepsilon} \cdot$B

Sebastian (A)

Esfandiar (B)





ε-δ- Graph

For each ε there is a minimal $\delta(\varepsilon)$

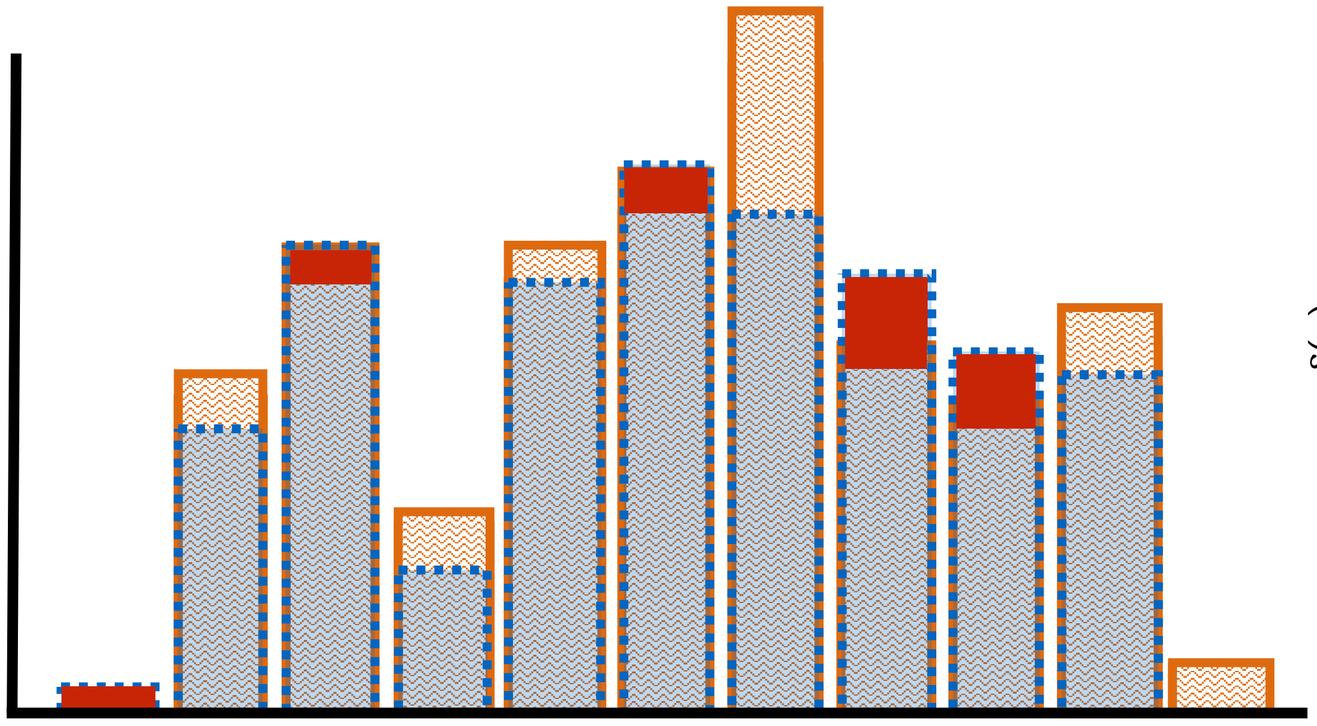
$\varepsilon = 0.1$

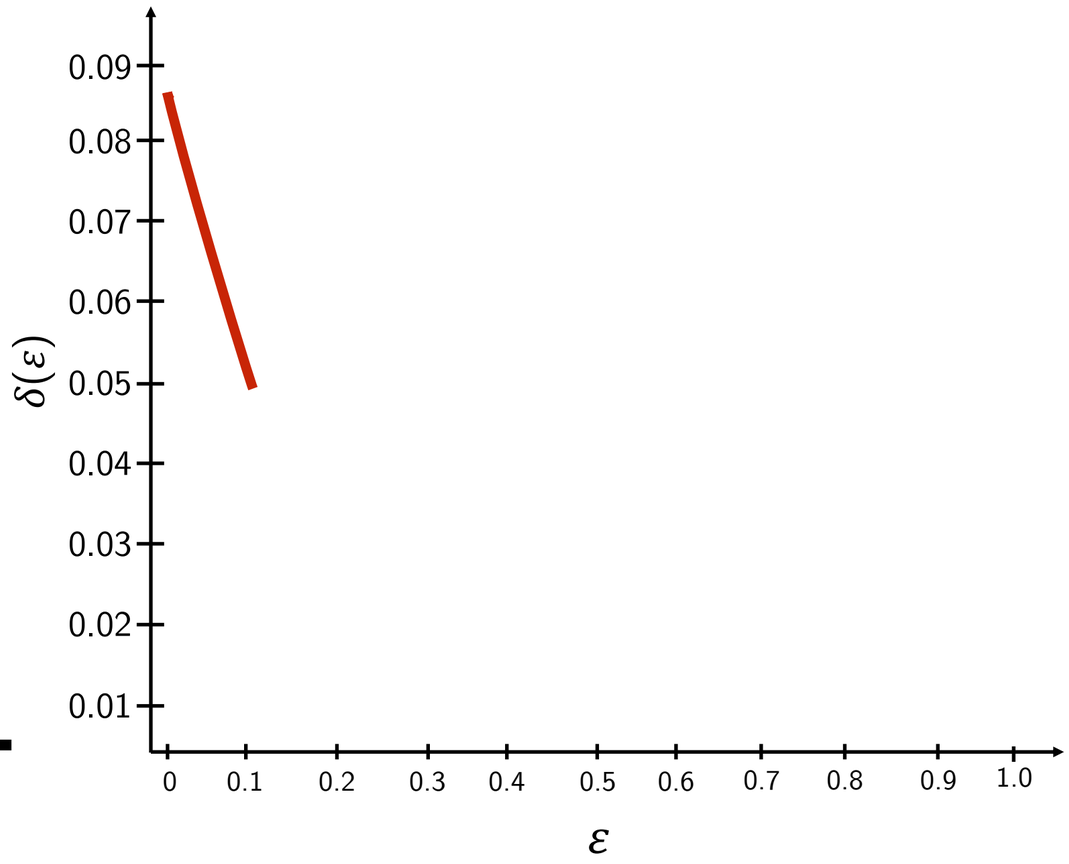
$e^{\varepsilon} = 1.1$

Sebastian (A)

Esfandiar (B)

A vs $e^{\varepsilon} \cdot$B





ε-δ- Graph

For each ε there is a minimal $\delta(\varepsilon)$

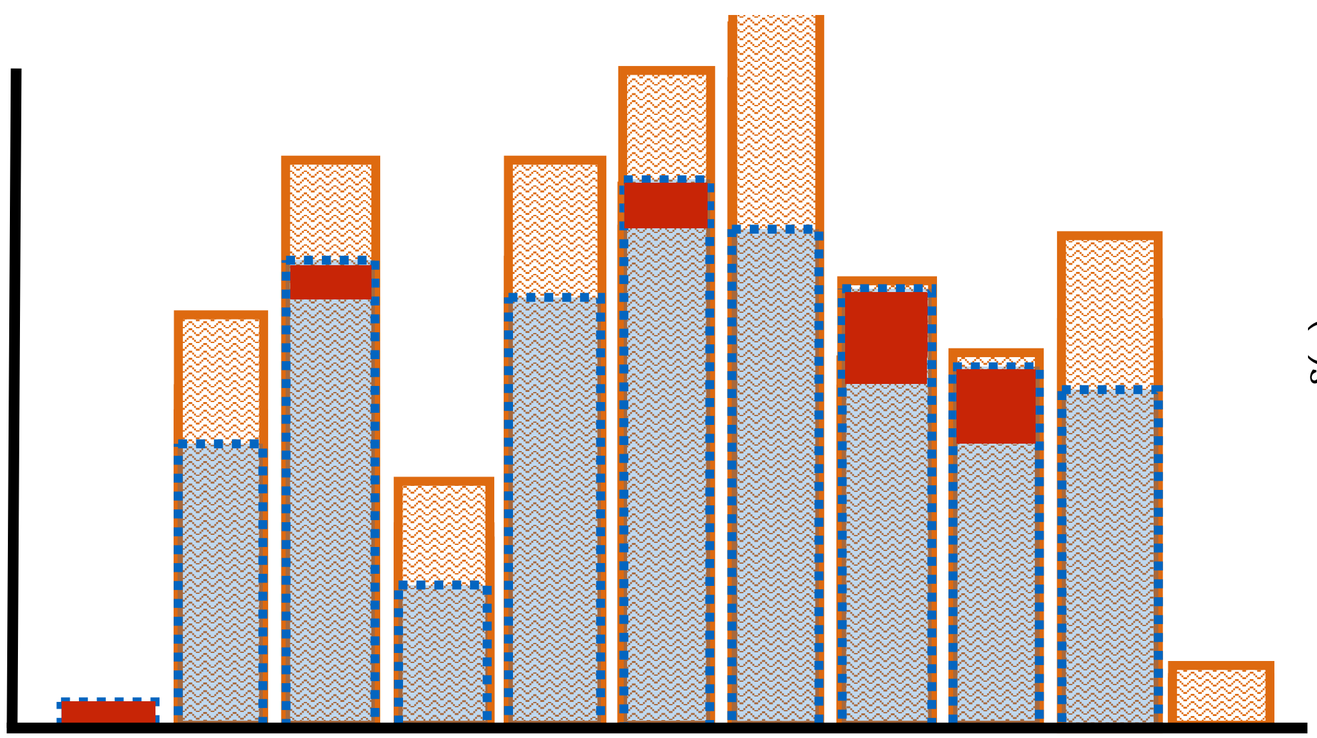
$\varepsilon = 0.3$

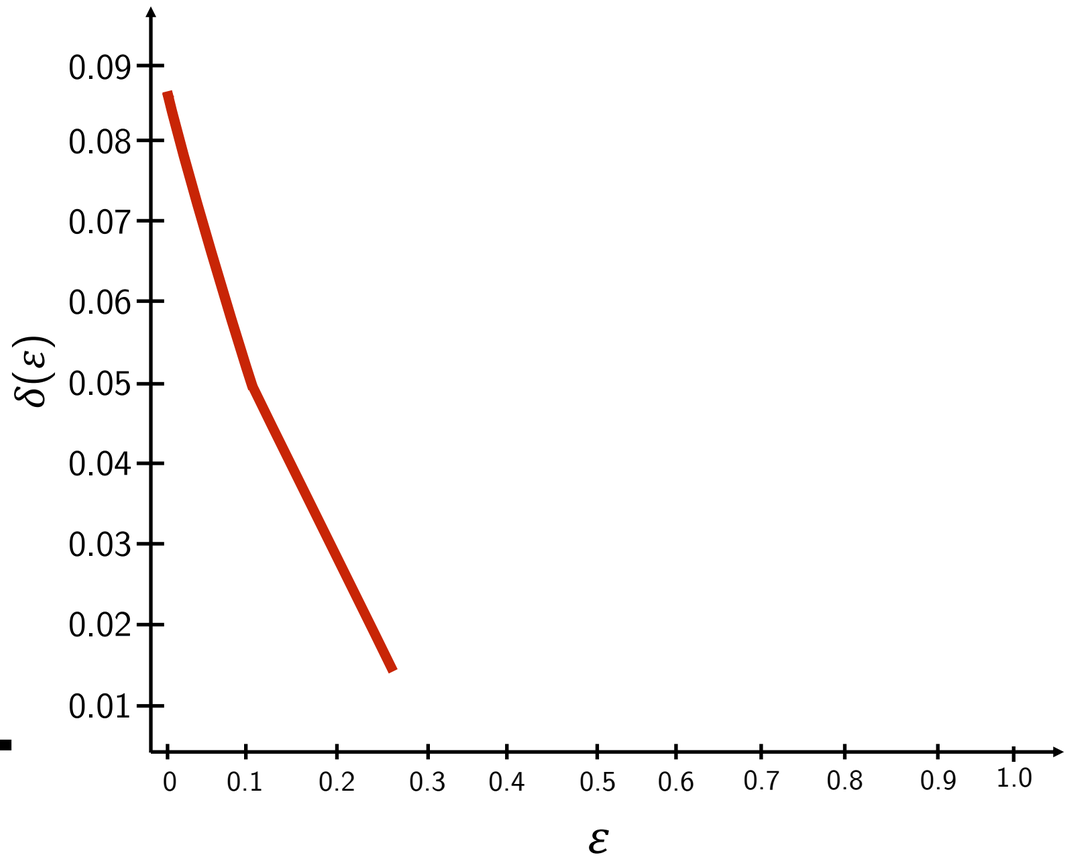
$e^{\varepsilon} = 1.3$

Sebastian (A)

Esfandiar (B)

A vs $e^{\varepsilon} \cdot$B





ε-δ- Graph

For each ε there is a minimal $\delta(\varepsilon)$

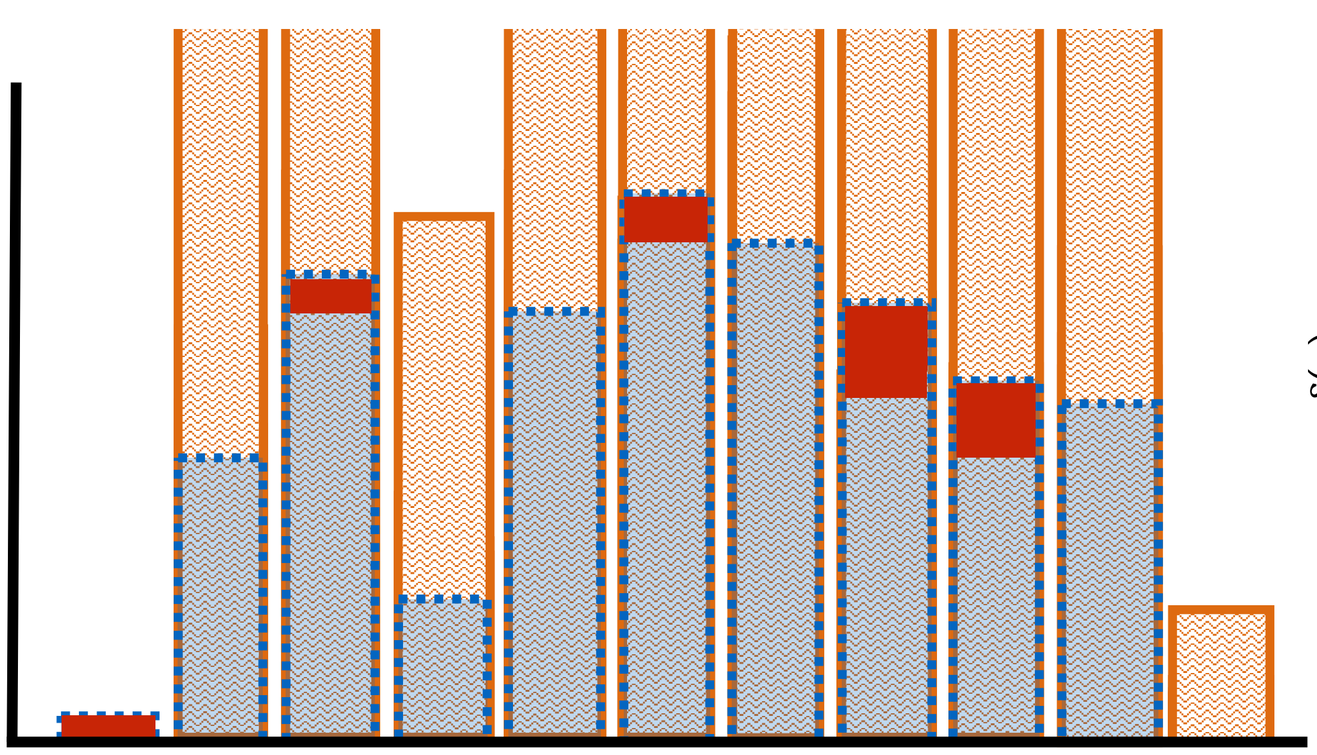
$\varepsilon = 1.0$

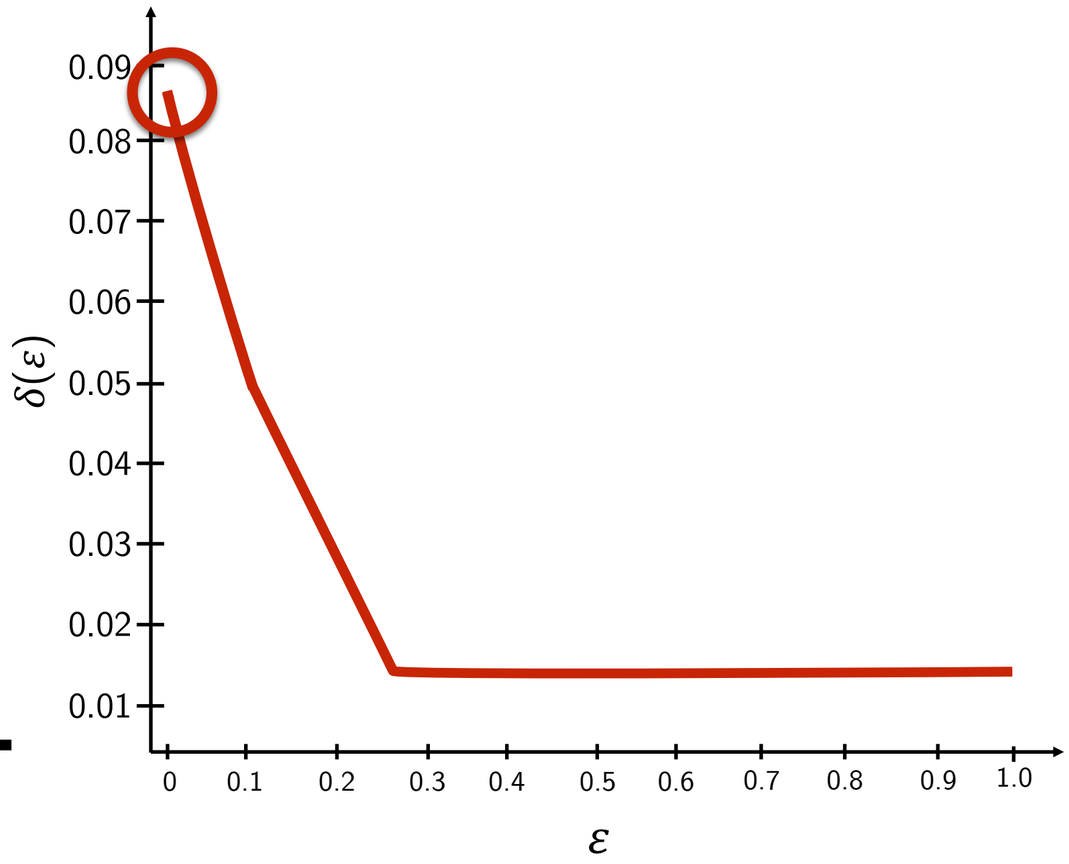
$e^{\varepsilon} = 2.7$

Sebastian (A)

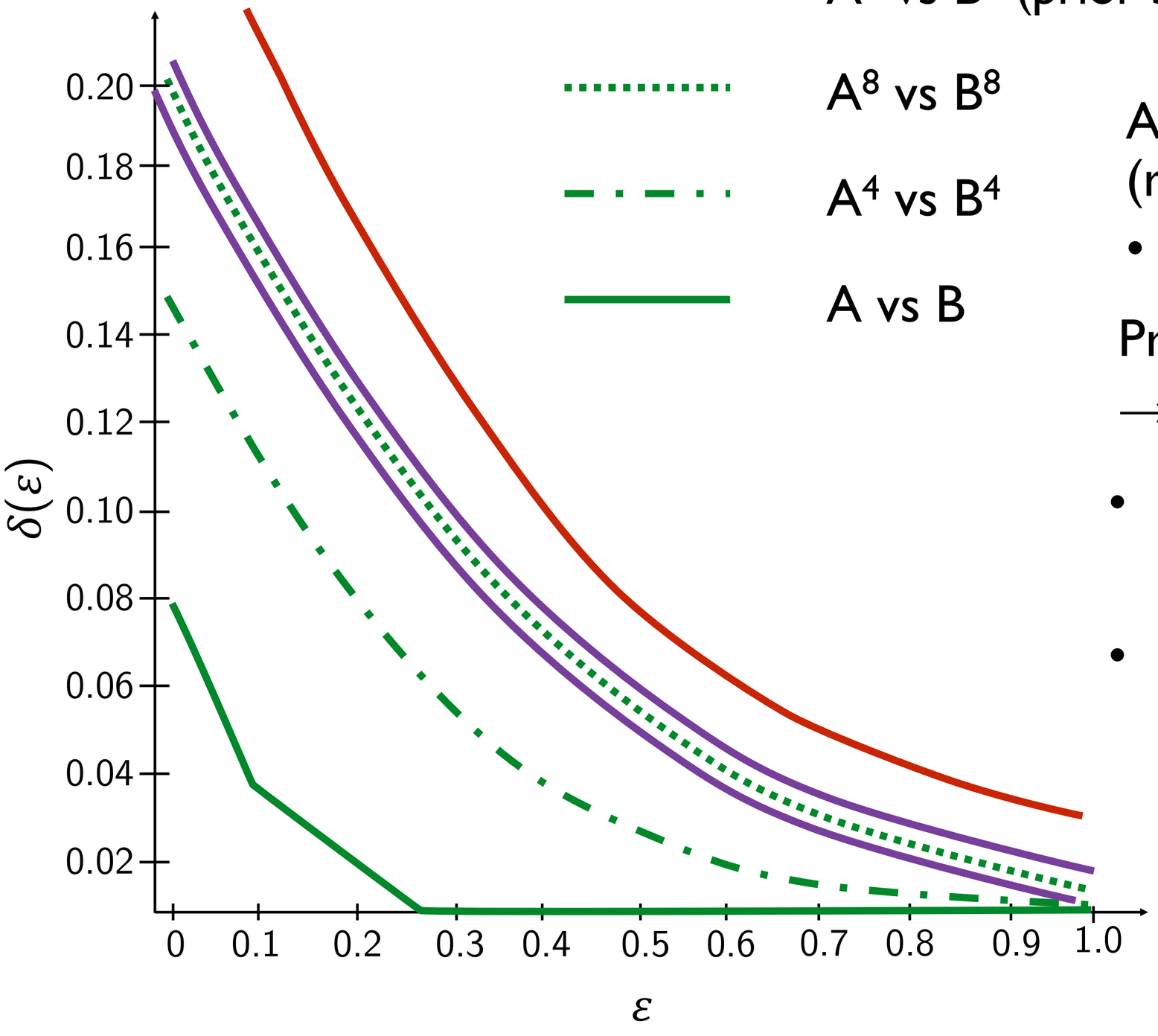
Esfandiar (B)

A vs $e^{\varepsilon} \cdot$B





R-fold ADP Hard to Compute



A^8 vs B^8 (prior bounds, moments accountant)

A^8 vs B^8

A^4 vs B^4

A vs B

ADP under r compositions (r-fold ADP)

- ADP for A^r vs B^r

Product distribution A^r:

→ infeasible to compute

- How tight are these approximations?
- Our work: tight upper and lower bounds

Alternative Approach for Computing r-fold ADP Bounds

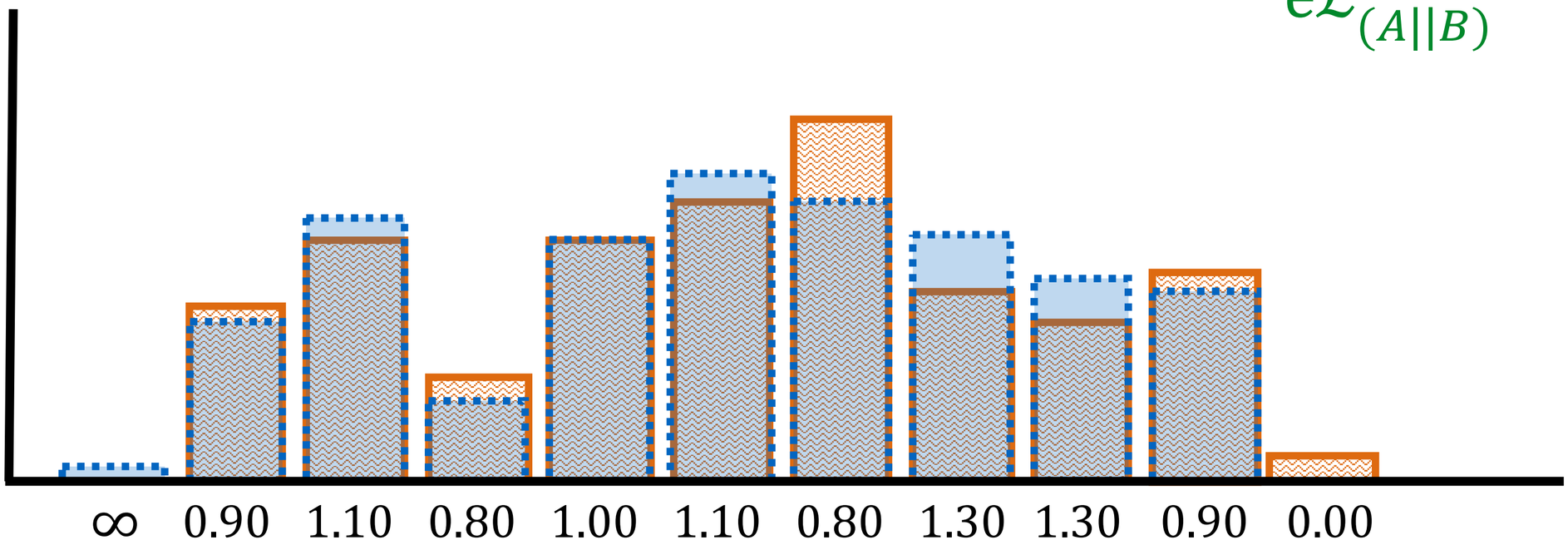
$$\delta(\varepsilon) = \sum_x \max(0, \Pr[x \leftarrow A] - e^{\varepsilon} \Pr[x \leftarrow B])$$

$$= \sum_x \max\left(0, \Pr[x \leftarrow A] \cdot \left(1 - \frac{e^{\varepsilon}}{e\mathcal{L}^{(x)}_{(A||B)}}\right)\right)$$

Sebastian (A)

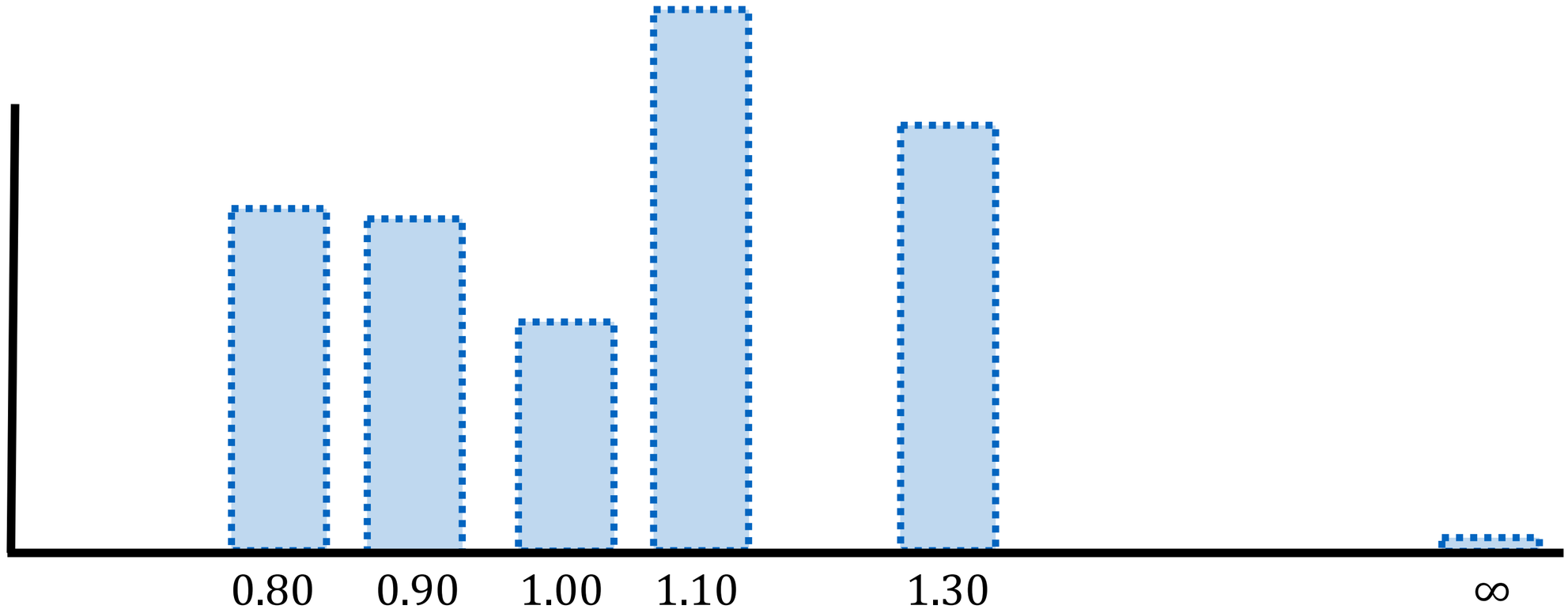
Esfandiar (B)

$$\Pr[x \leftarrow B] = \frac{\Pr[x \leftarrow A]}{e\mathcal{L}^{(x)}_{(A||B)}}$$



Privacy loss $e\mathcal{L}^{(x)}_{(A||B)} = \frac{\Pr[x \leftarrow A]}{\Pr[x \leftarrow B]}$

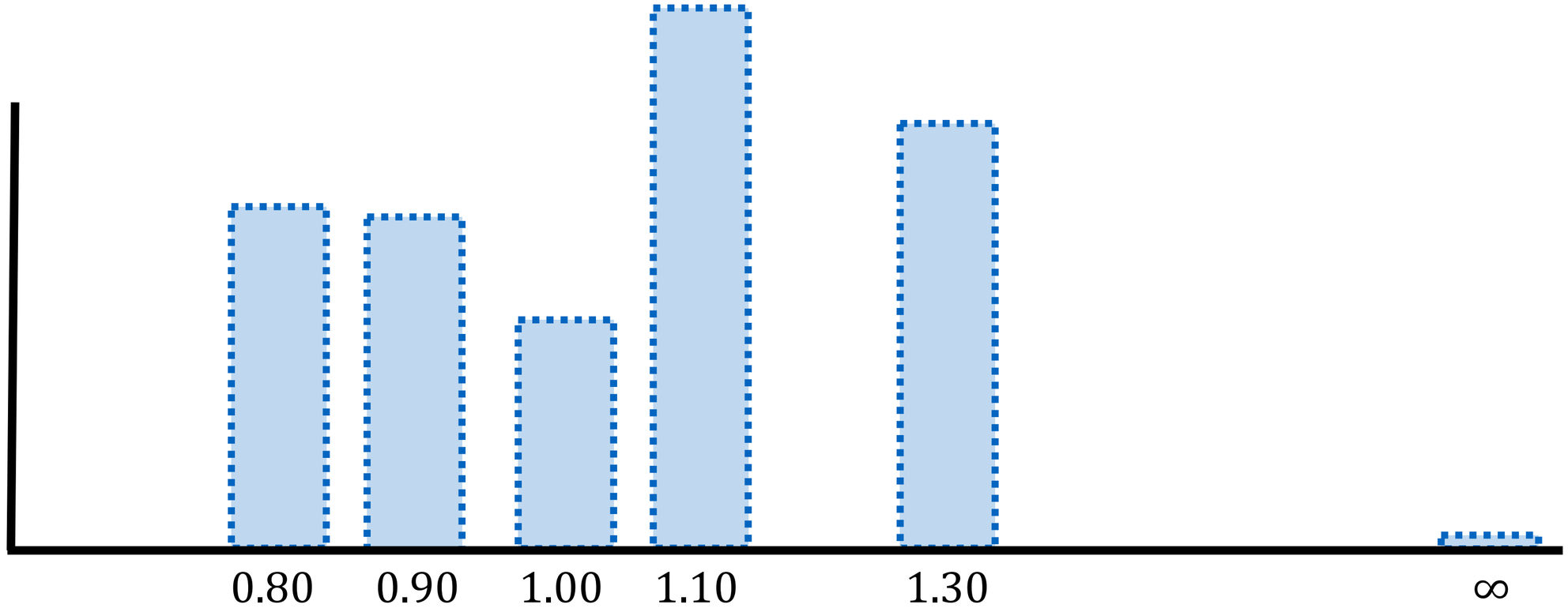
Summarizing and ordering by the privacy loss



Composing the privacy loss

$$e\mathcal{L}^{(x)}_{(A||B)} \qquad e\mathcal{L}^{(y)}_{(A||B)}$$

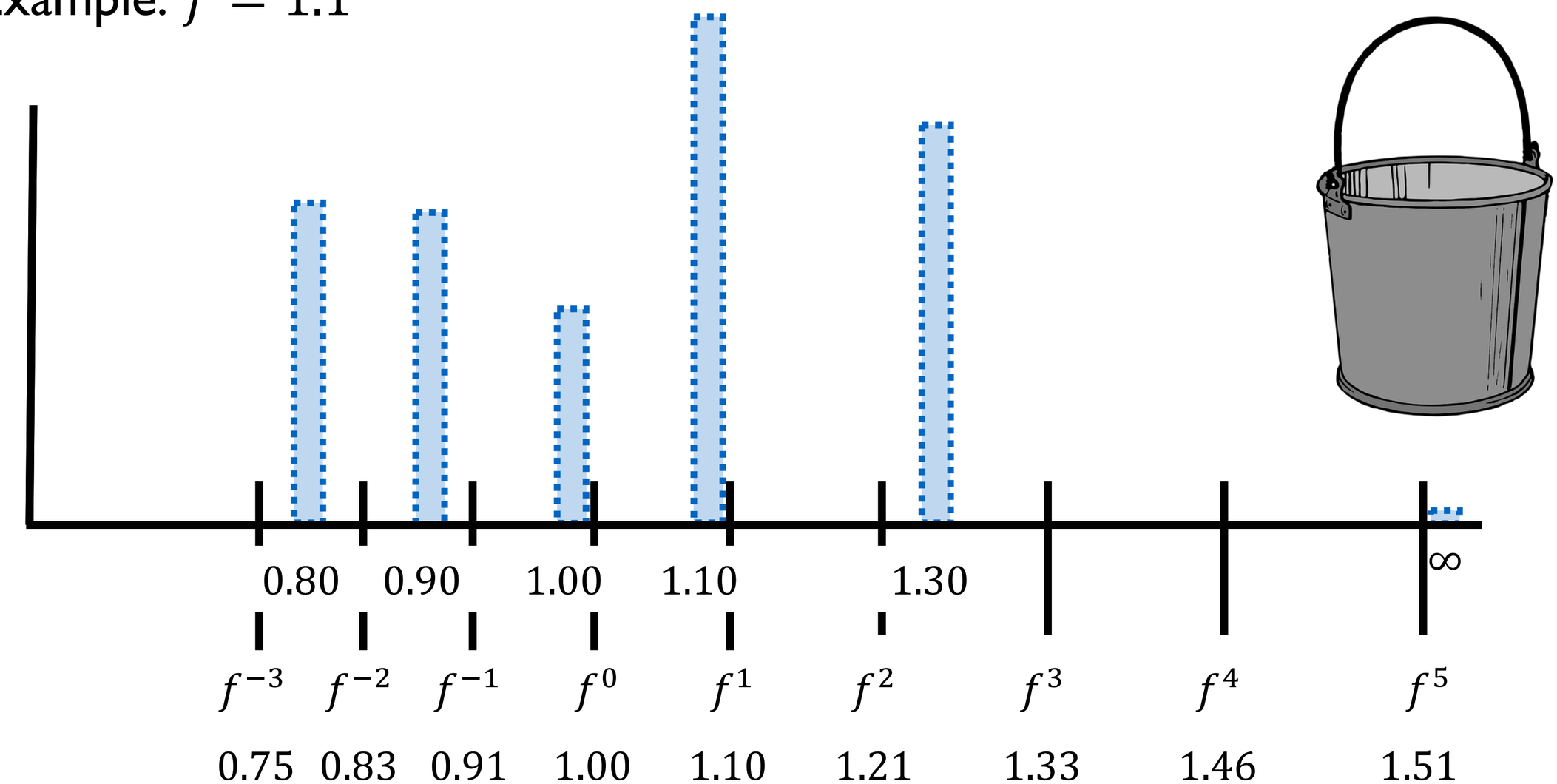
$$e\mathcal{L}^{(x,y)}_{(A^2||B^2)} = \frac{\Pr[(x,y) \leftarrow A^2]}{\Pr[(x,y) \leftarrow B^2]} = \frac{\Pr[x \leftarrow A]}{\Pr[x \leftarrow B]} \cdot \frac{\Pr[y \leftarrow A]}{\Pr[y \leftarrow B]}$$



Privacy Buckets

Buckets for making composition easier:

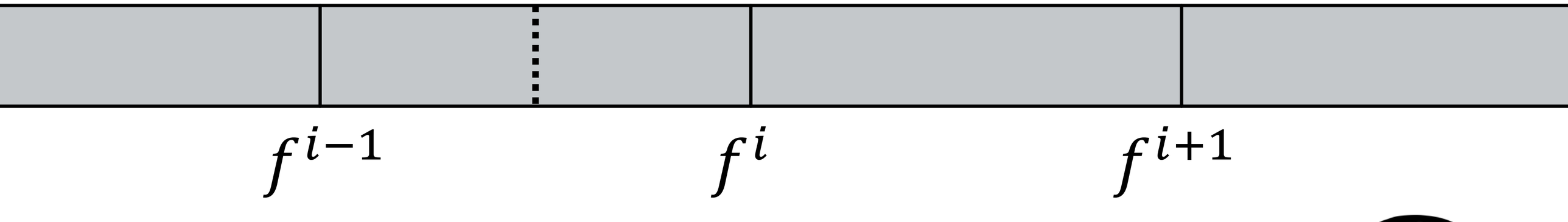
- Approximate privacy loss to f^i for different i (always round up)
- Example: $f = 1.1$



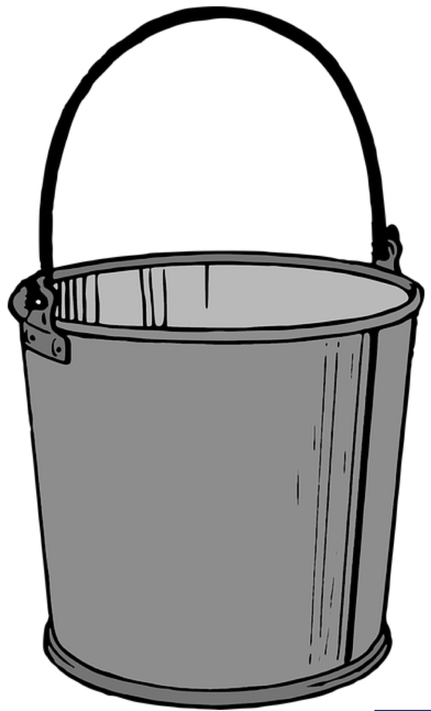
Precision of our privacy buckets approximation

What do we actually do and how large is the error?

$$e\mathcal{L}^{(x)}_{(A||B)} = \frac{\Pr[\text{Seb. watches } x]}{\Pr[\text{Esf. watches } x]} =$$



We later reconstruct as $\frac{\Pr[\quad]}{f^i}$ instead of $\frac{\Pr[\quad]}{e\mathcal{L}^{(x)}_{(A||B)}}$

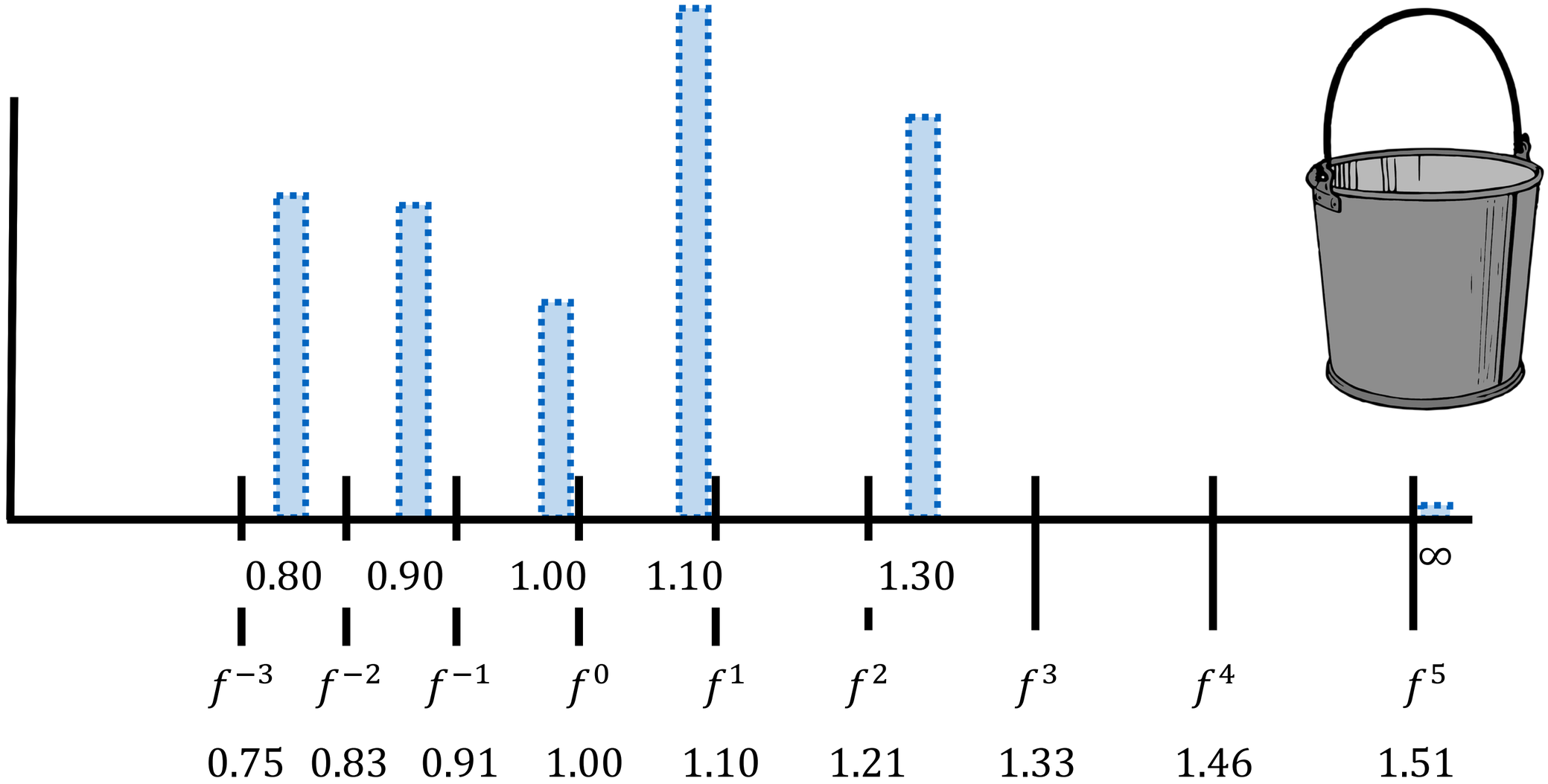


Privacy Buckets: composition

Buckets with $f = 1.1$

$$f^{j+i} \qquad f^{j} \qquad f^{i}$$

$$e\mathcal{L}^{(x,y)}_{(A^2||B^2)} = \frac{\Pr[(x,y) \leftarrow A^2]}{\Pr[(x,y) \leftarrow B^2]} = \frac{\Pr[x \leftarrow A]}{\Pr[x \leftarrow B]} \cdot \frac{\Pr[y \leftarrow A]}{\Pr[y \leftarrow B]}$$



Privacy buckets composition

Example: we compute f^2 for buckets from f^{-5} to f^5 (and ∞)

Before composition

f^{-5} f^{-4} f^{-3} f^{-2} f^{-1} f^{0} f^{1} f^{2} f^{3} f^{4} f^{5} ∞

f^{-5} f^{-4} f^{-3} f^{-2} f^{-1} f^{0} f^{1} f^{2} f^{3} f^{4} f^{5} ∞

After composition

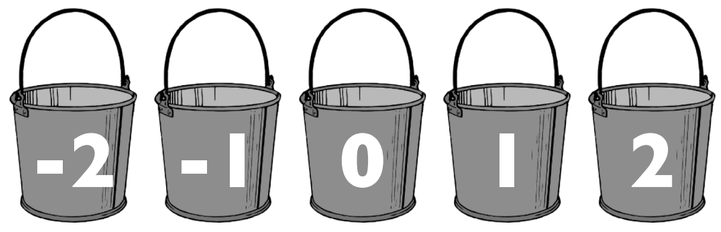
f^{-5} f^{-4} f^{-3} f^{-2} f^{-1} f^{0} f^{1} f^{2} f^{3} f^{4} f^{5} ∞

$$\text{1x2}(i) = \sum_{j,k\ s.t. j+k=i} \text{1}(j) \cdot \text{2}(k)$$

Feasibility

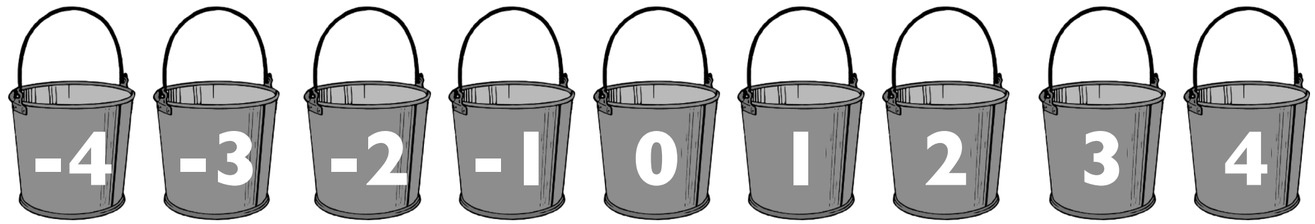
How does composition affect the number of required buckets?

Starting with 5 buckets



2 observations

1 self-composition → 9 buckets



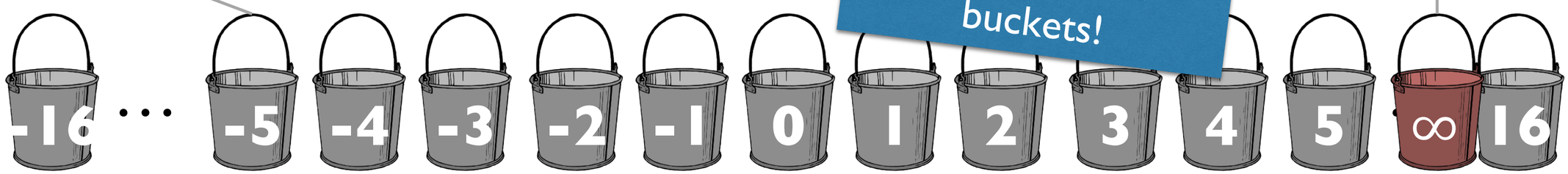
4 observations

Collect all $\leq -n$

2 self-compositions

Collect all $> n$

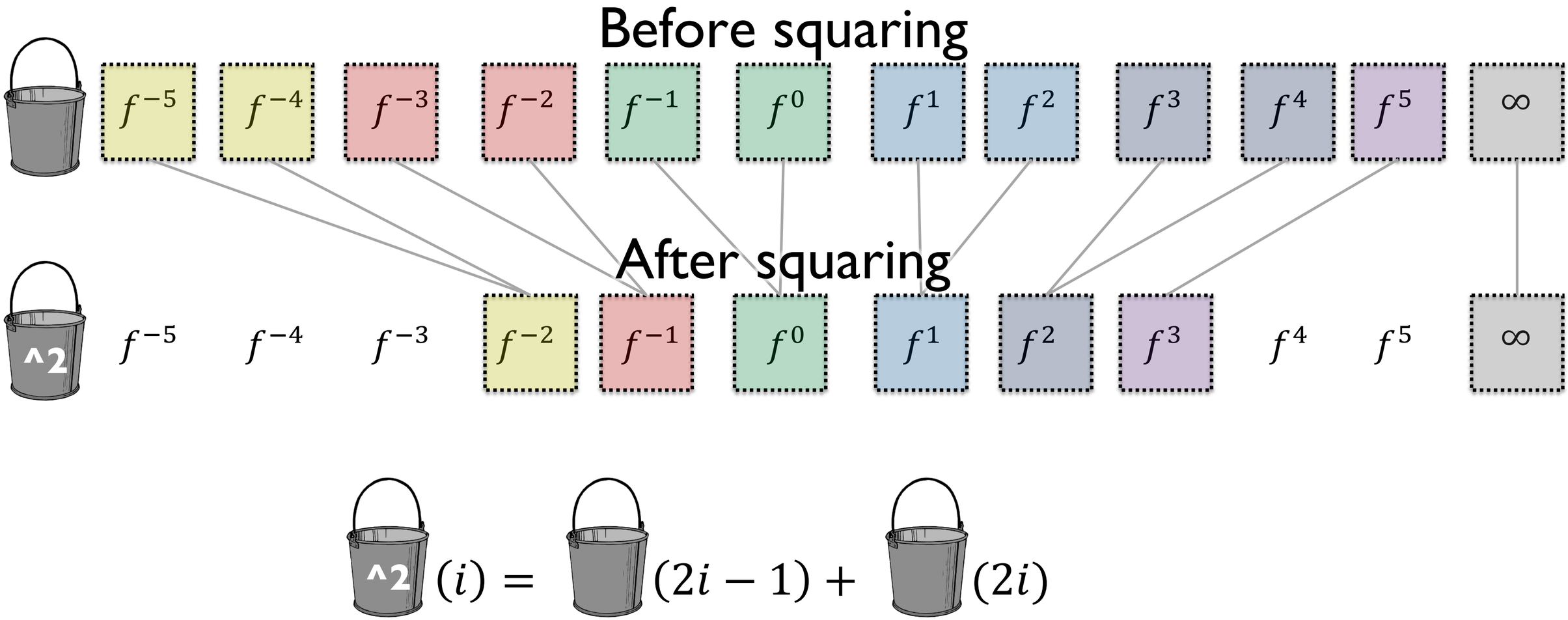
limit to 2n+2 buckets!



Privacy buckets squaring

We "square" the buckets, i.e., we square f to f^2.

We then rearrange the buckets.



$$\text{^2}(i) = (2i-1) + (2i)$$

Implementation and Performance

Unoptimized Python (NumPy) implementation
(405 LoC) accessible under:
https://github.com/sommerda/privacybuckets

- Reimplemented by David Sommer (ETH)

Computation time dominated by convolution:

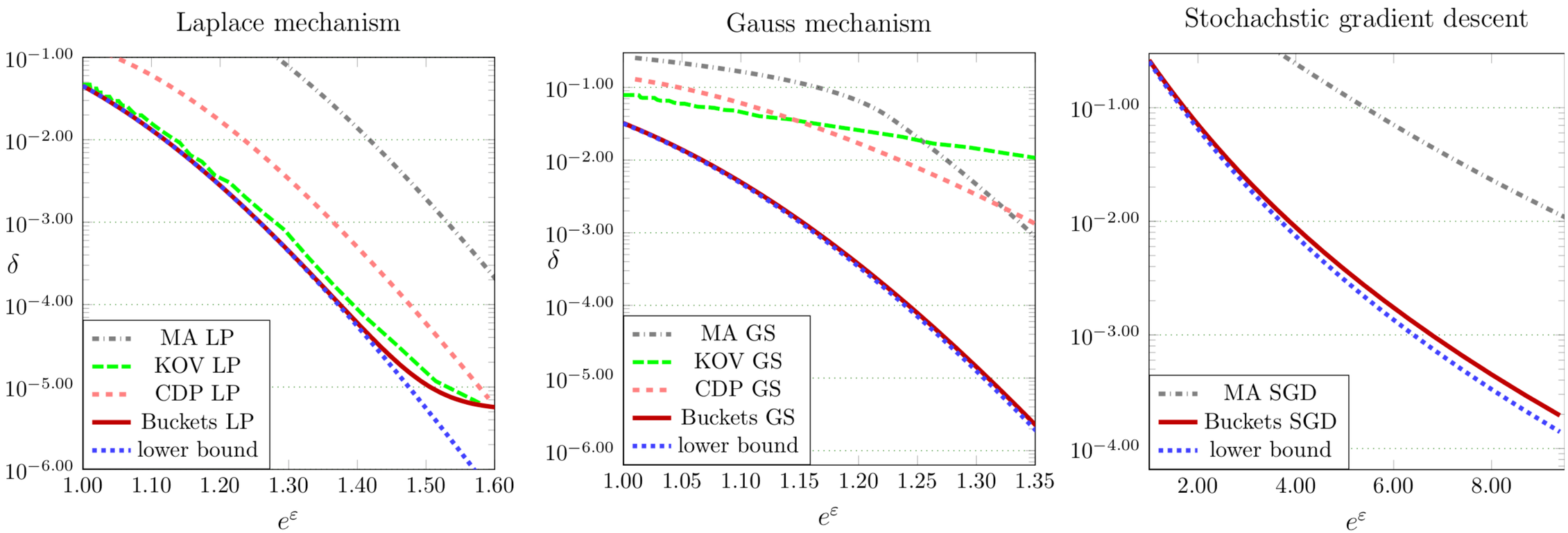
- FFT-based convolution $O(n \log n)$
- Implementation uses normal convolution $O(n^2)$ for numerical stability
- in total ~ 13 seconds per composition operation
  (with 100 000 buckets)
- Repeated squaring: 2^r-fold ADP bounds requires r compositions

How tight are these bounds?

Comparison against the State-Of-The-Art

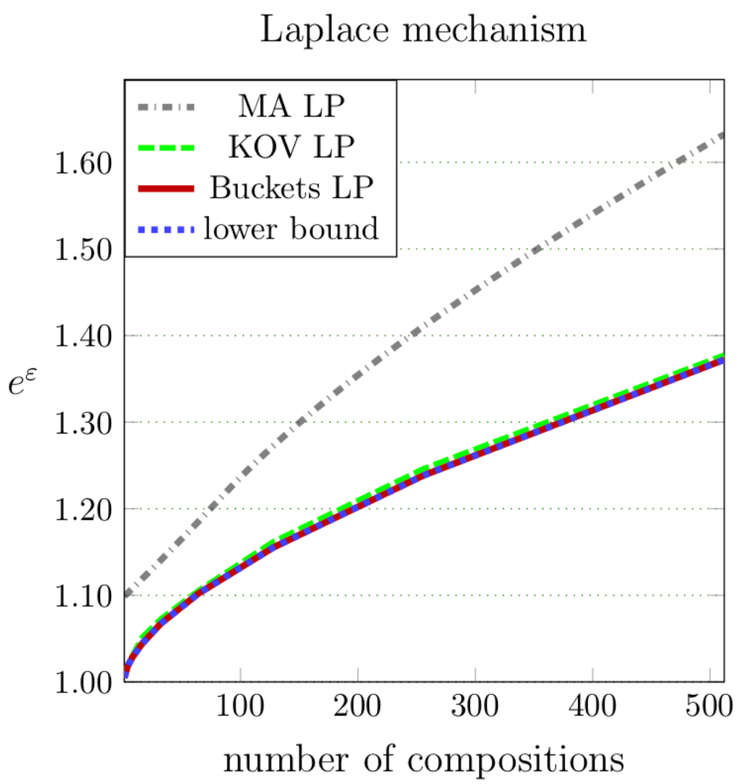
Laplace & Gauss after 512 observations, SGD after 60 000 observations

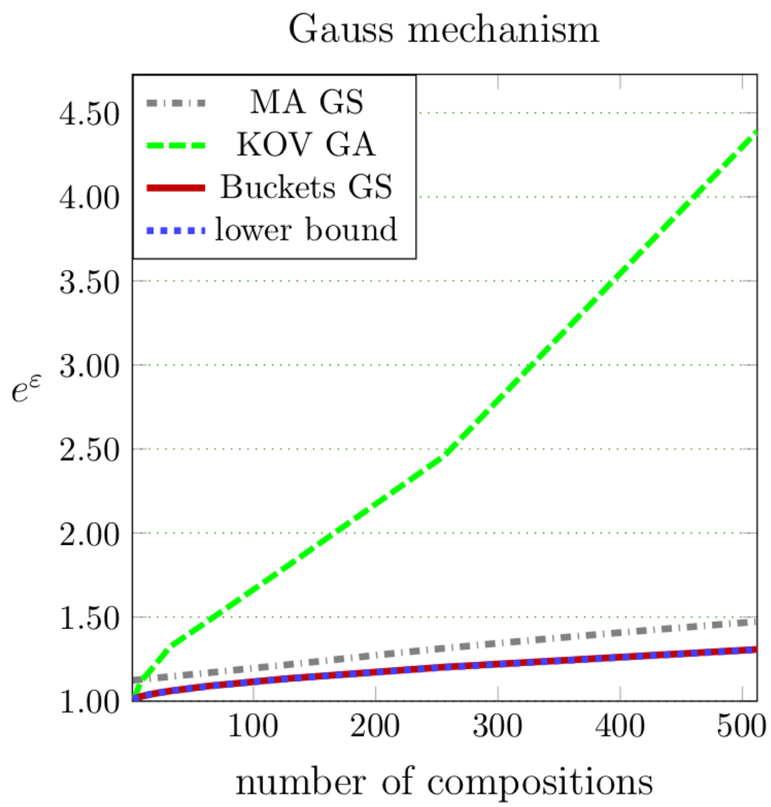
CDP bound not applicable to SGD (and CoverUp)

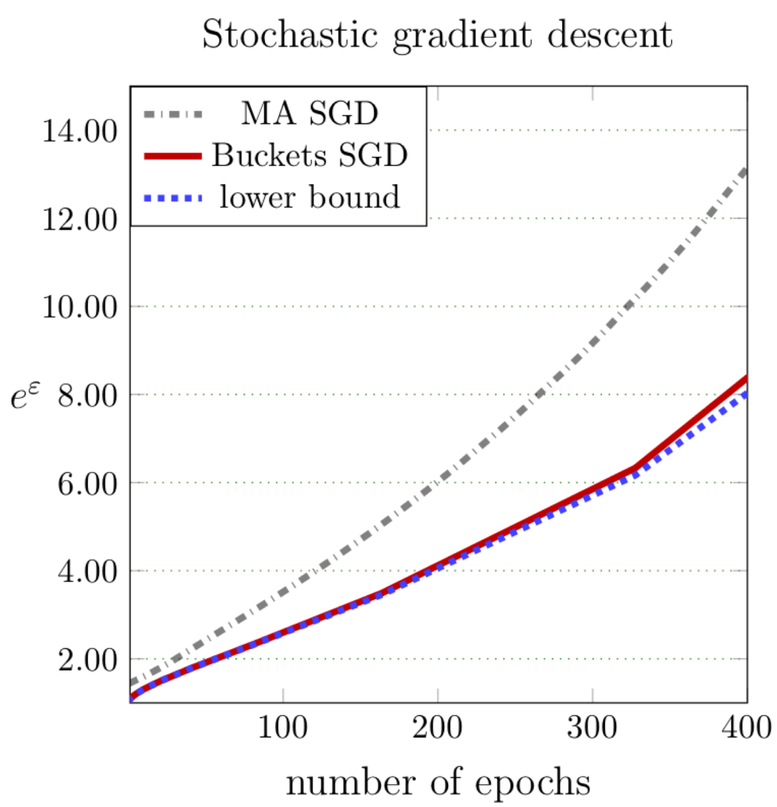


The rise of epsilon (over time)

We compute the minimal e^{ε} for which we can still achieve $\delta(\varepsilon) \leq 10^{-4}$ over the number of compositions.







Conclusion

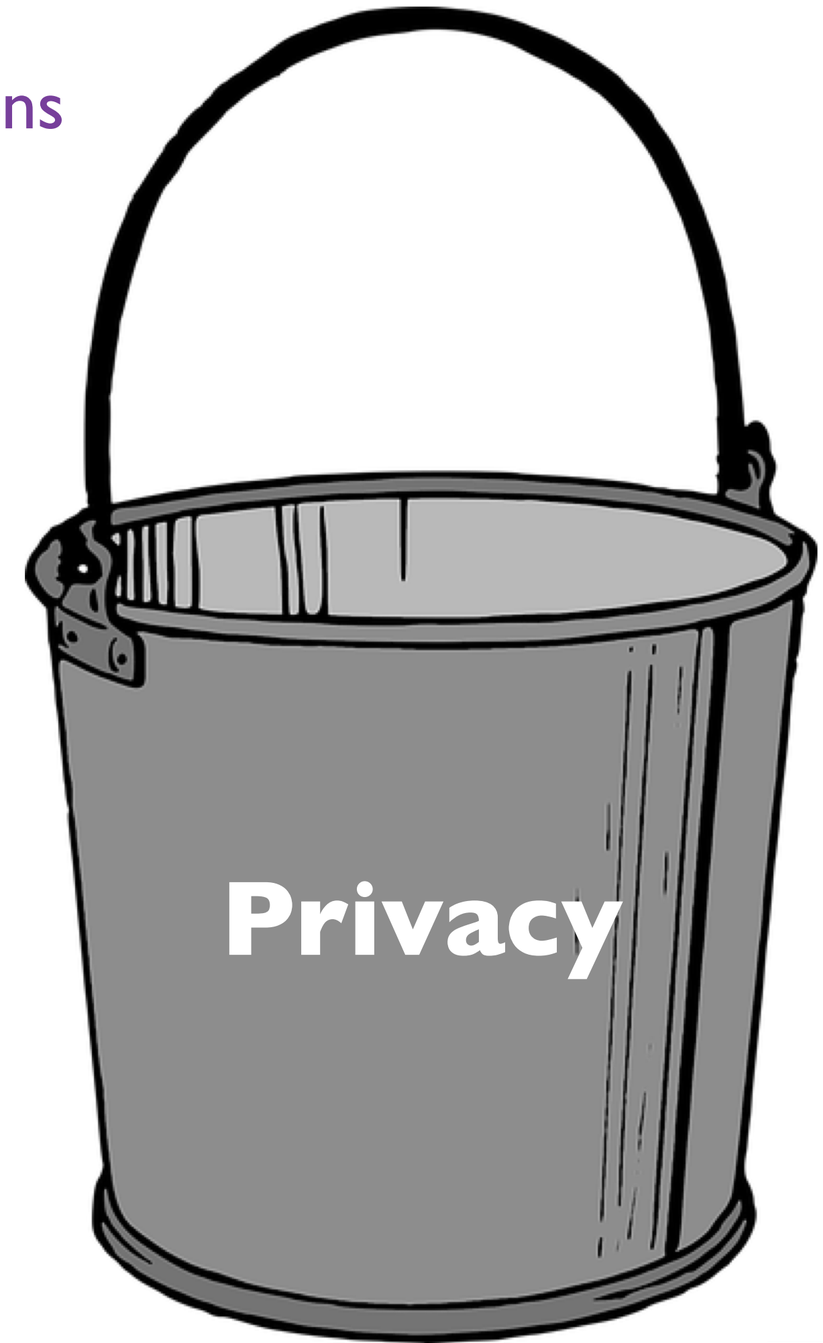
Privacy Buckets: novel approximation of leakage

Computes upper and lower bounds for
Approximate Differential Privacy after r observations

Github repository:
https://github.com/sommerda/privacybuckets

To make our results more accessible:

- webpage to play around with (http for now)
  http://privacybuckets.space



Thanks for your attention!
Questions?

privacybuckets.space