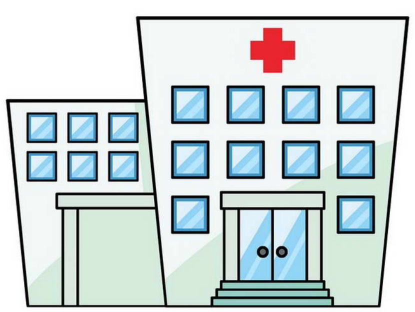
Data Privacy by Programming Language Design

David Darais
University of Vermont

Your Personal Data

Google

nest



NETFLIX

TESLA

f

amazon

Good uses of data

Improve a product

Enable better business decisions

Support fundamental research

Bad uses of data

Stalking and harassment

Unfair business advantages

Threats and blackmail

Good ⌁ Bad

Google

Good ⌁ Bad

Google

Good ⌁ Bad

Non-solution: Anonymization

First Name	Last Name	University
David	Darais	U Vermont
Éric	Tanter	U Chile
Federico	Olmedo	U Chile

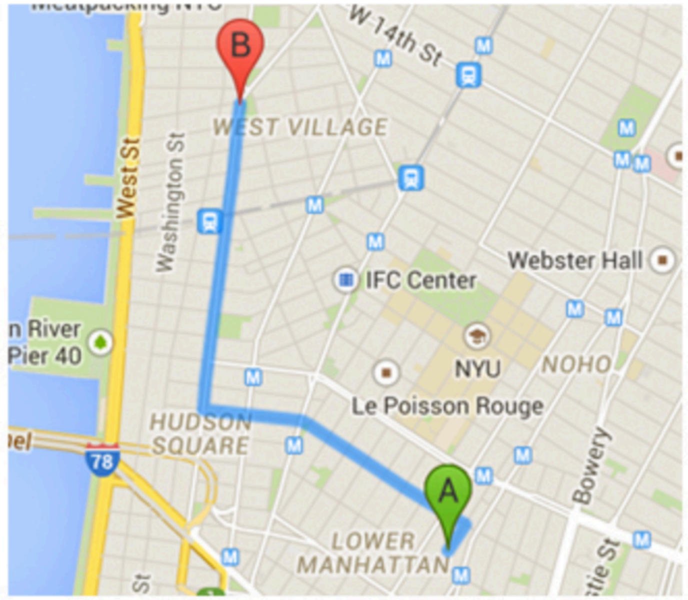
First Name	Last Name	University
#####	#####	U Vermont
#####	#####	U Chile
#####	#####	U Chile

Non-solution: Anonymization

Non-solution: Anonymization

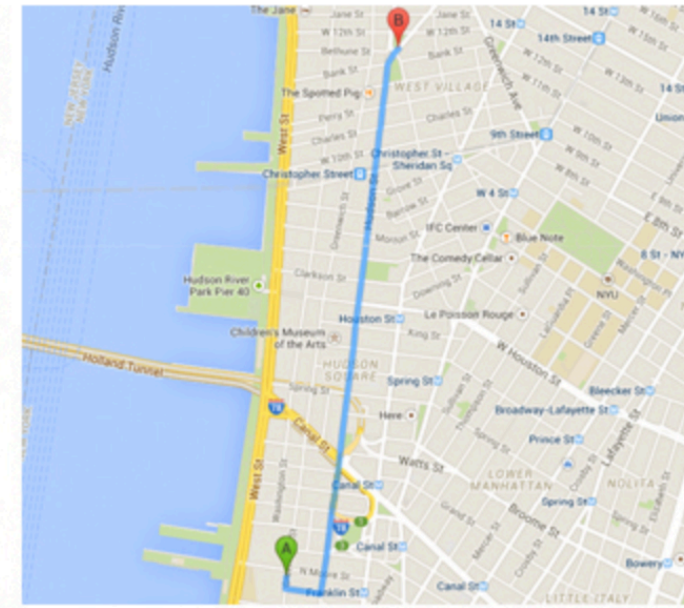
Dataset	Visible	Auxiliary Data	Attack
Anonymized Netflix Viewer Data	ID + ratings + dates	IMDB	Re-identification (what movies you watch)
NYC Taxi Data	ID + time + coordinates + fare + tip	Geotagged celebrity photos	Re-identification (celebrity trips + tip amounts)
Anonymized AOL Search Data	ID + query text	Ad-hoc	Re-identification (search history)
Massachusetts Hospital Visit Data	All except: name + address + SSN	Public voting records (name, address, birth date)	Re-identification (health records, diagnoses + prescriptions)

ASHLEE SIMPSON



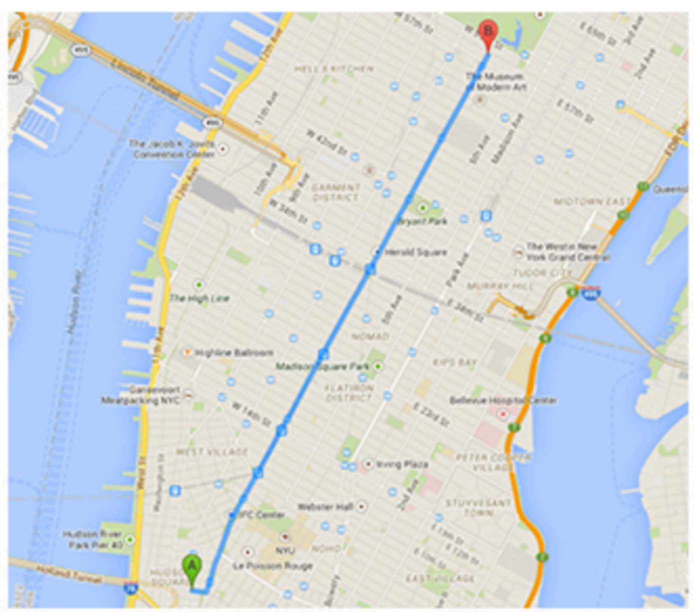
JANUARY 6, 2013 • 3:29 PM - 3:38 PM
78 CROSBY ST. TO 580 HUDSON ST.
$7.50 FARE • $2 TIP • ©SPLASH

JUDD APATOW
LESLIE MANN



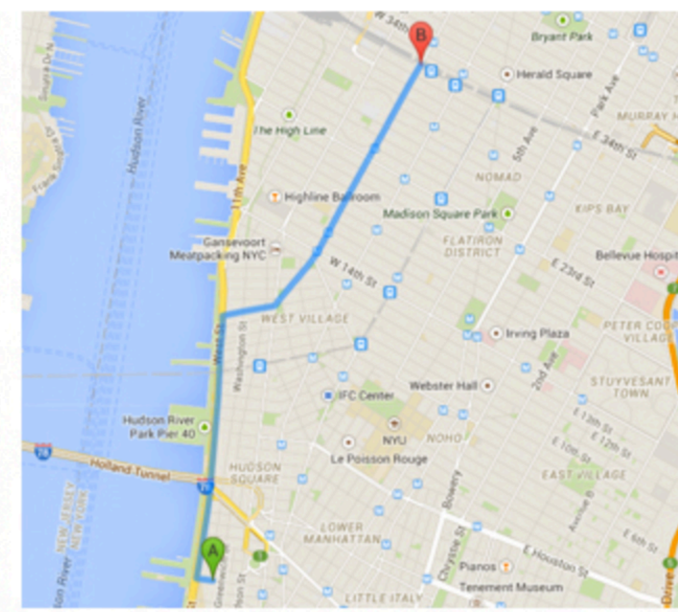
JUNE 21, 2013 • 11:28 AM - 11:35 AM
376 GREENWICH ST. TO 1 ABINGDON SQUARE
$7.00 FARE • $2.10 TIP • ©SPLASH

KOURTNEY KARDASHIAN
SCOTT DISICK



NOVEMBER 4, 2013 • 12:11 PM - 12:36 PM
246 SPRING ST. TO 1412 6TH AVE
$16.50 FARE • $3.40 TIP • ©SPLASH

KATHERINE HEIGL

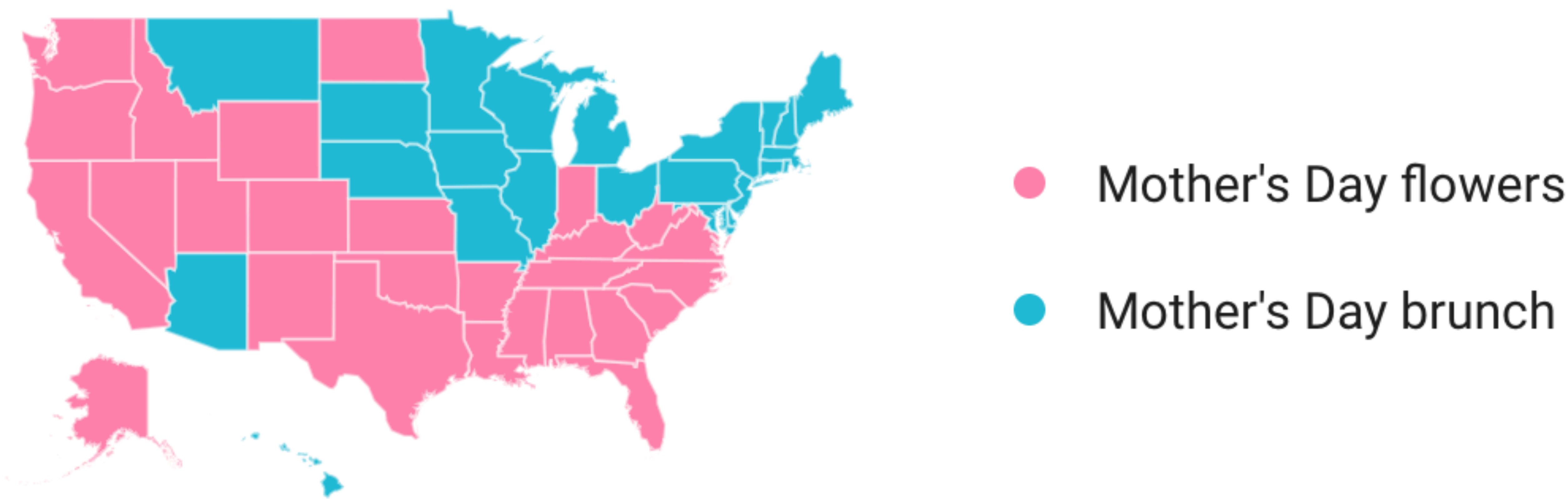


OCTOBER 4, 2013 • 1:21 PM - 1:40 PM
80 N. MOORE ST. TO 421 8TH AVE
$14.50 FARE • $3.62 TIP • ©WENN

For any 'anonymized' dataset, either the data is useless, or there exists an auxiliary dataset that re-identifies it.

–Dwork & Roth (The Algorithmic Foundations of Differential Privacy)

Almost-solution:
Aggregate statistics



artificial intelligence

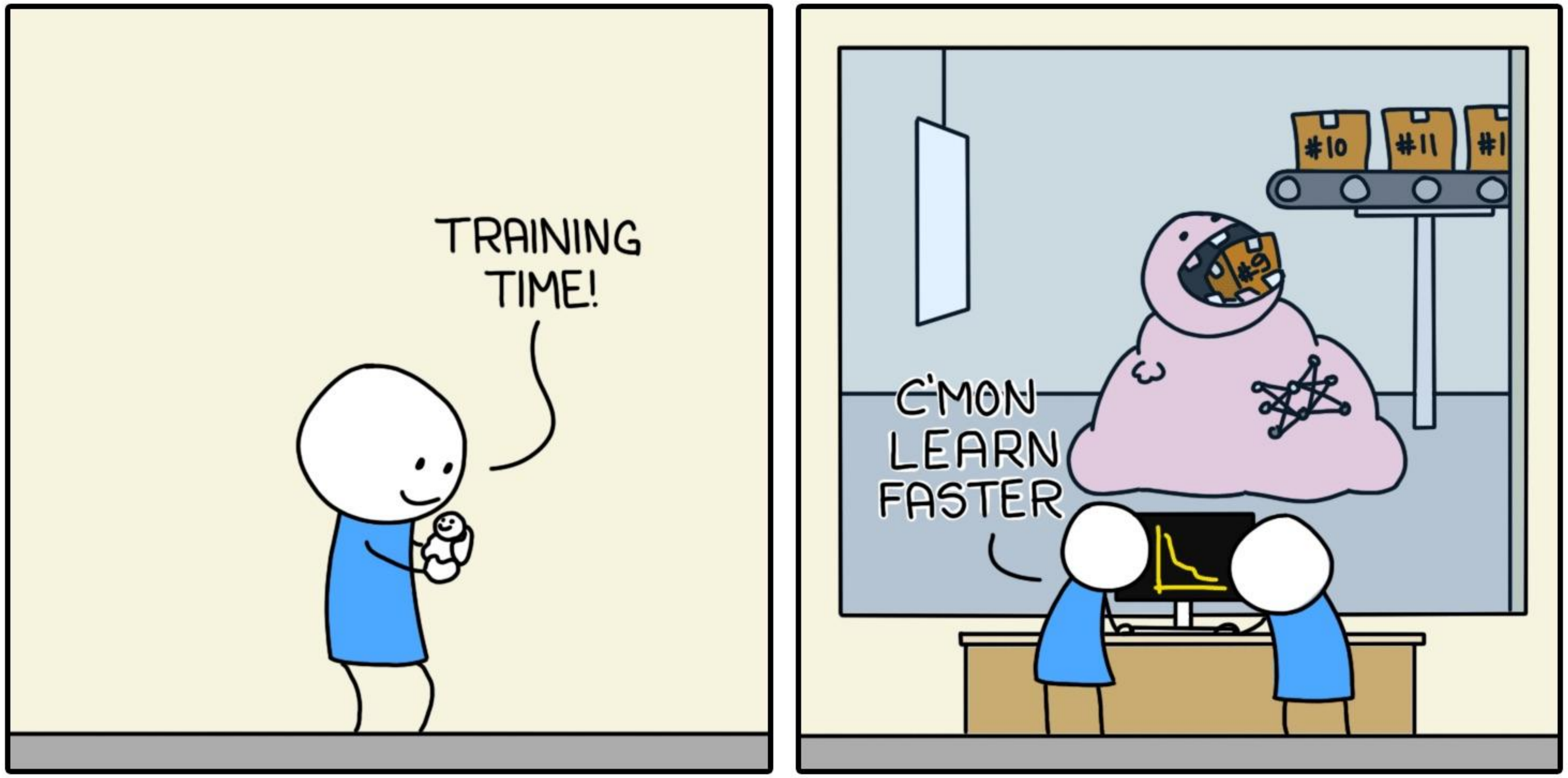
=

aggregate statistics

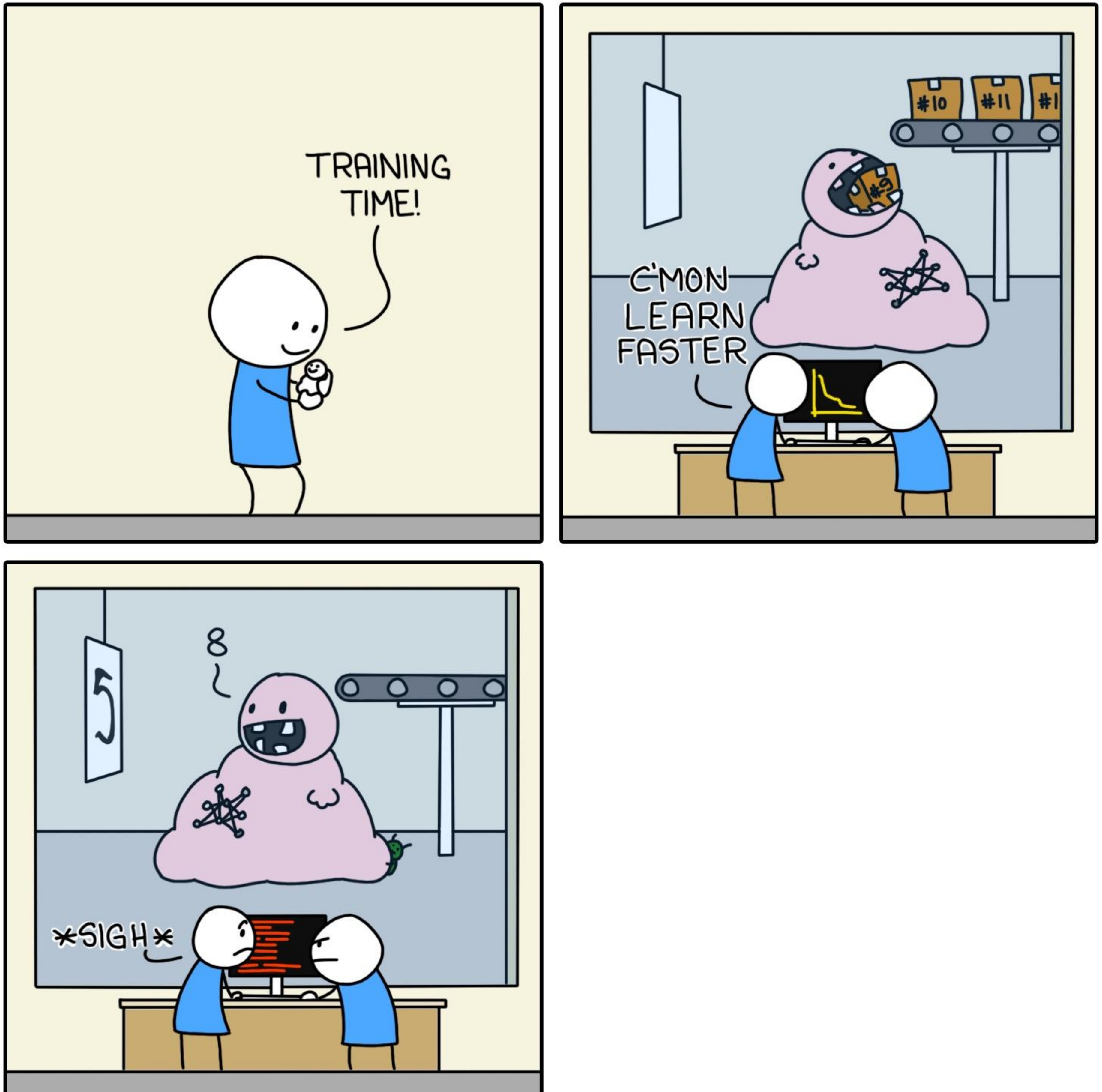
?

NEW MODEL

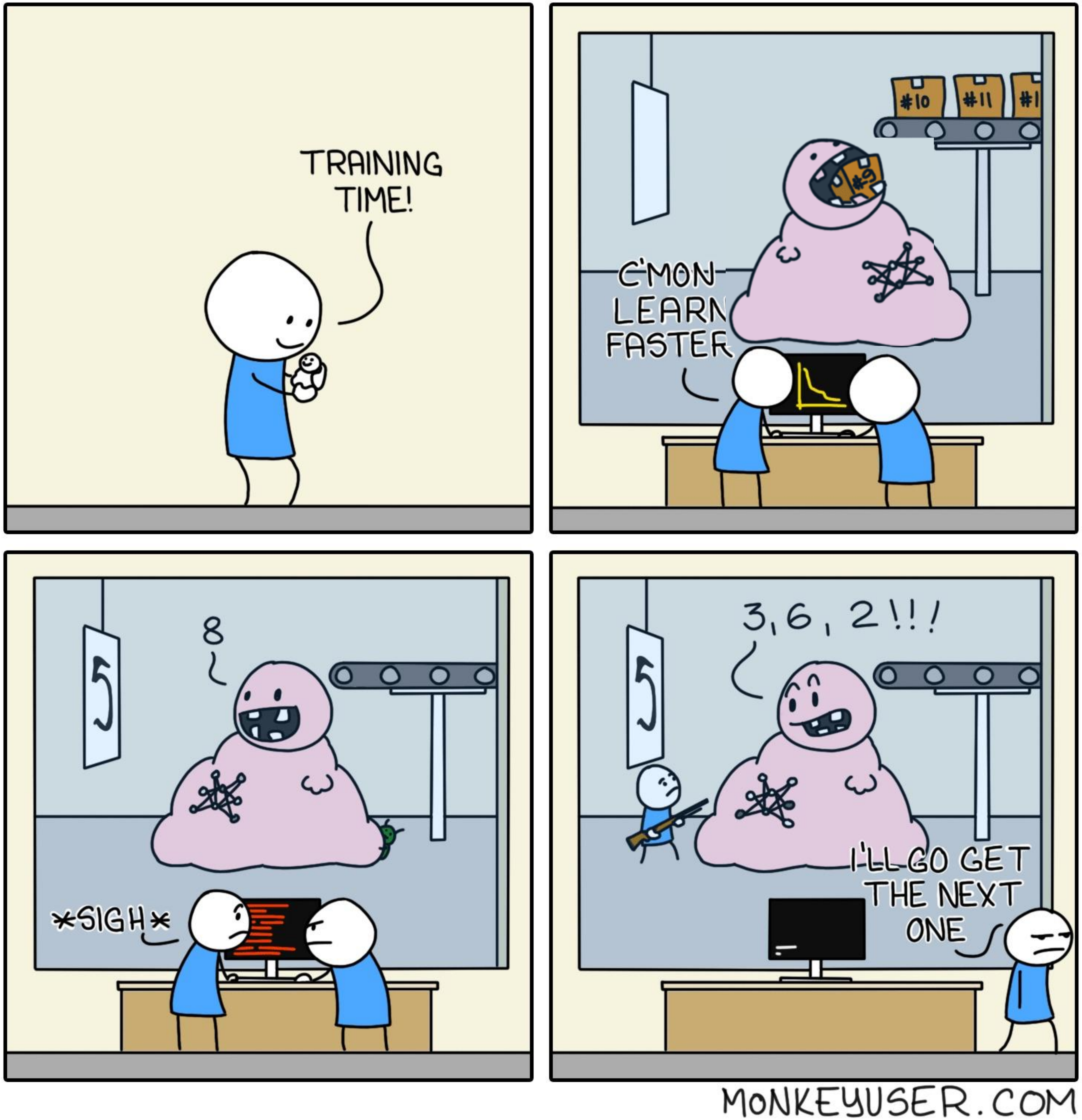
NEW MODEL

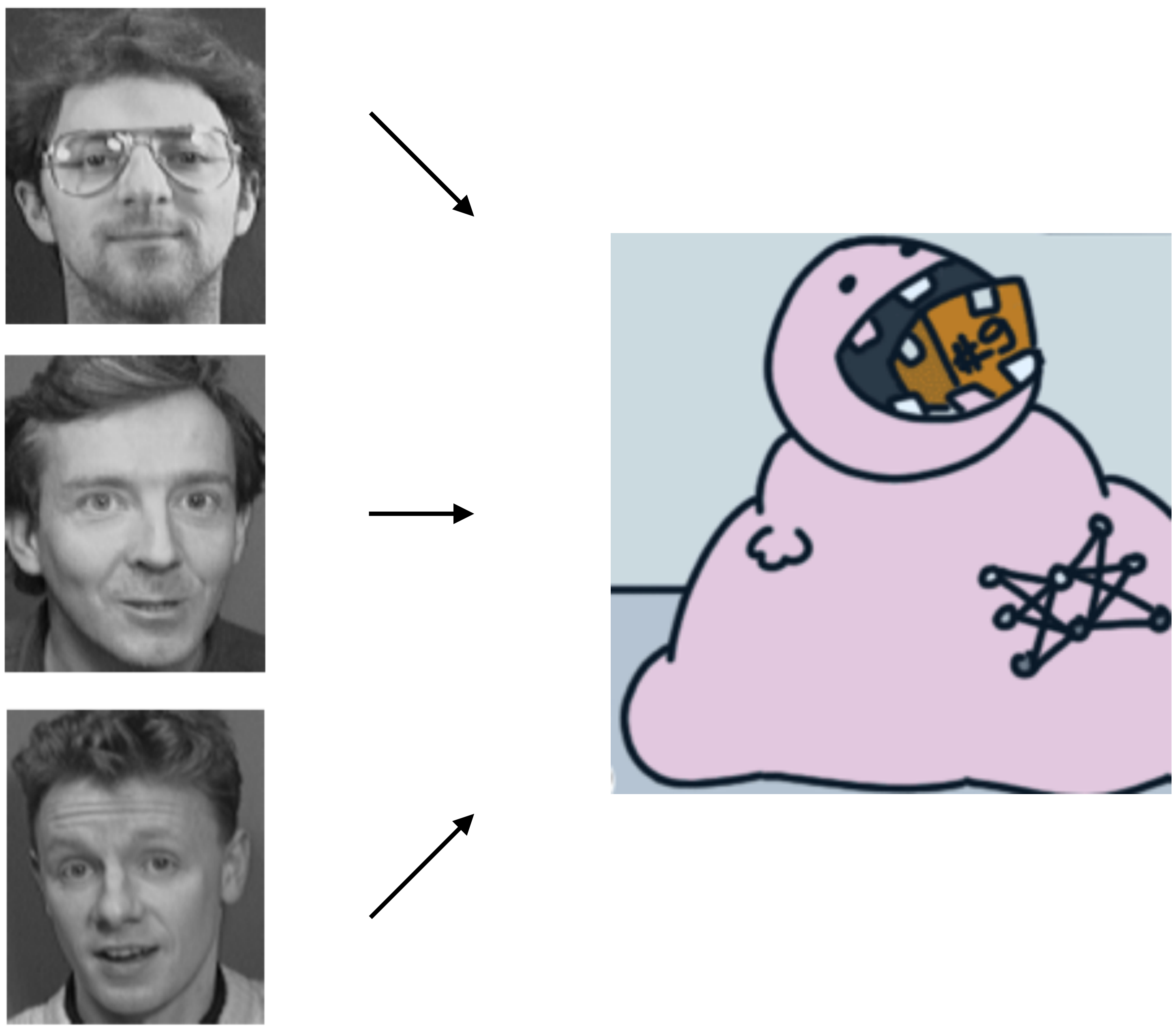


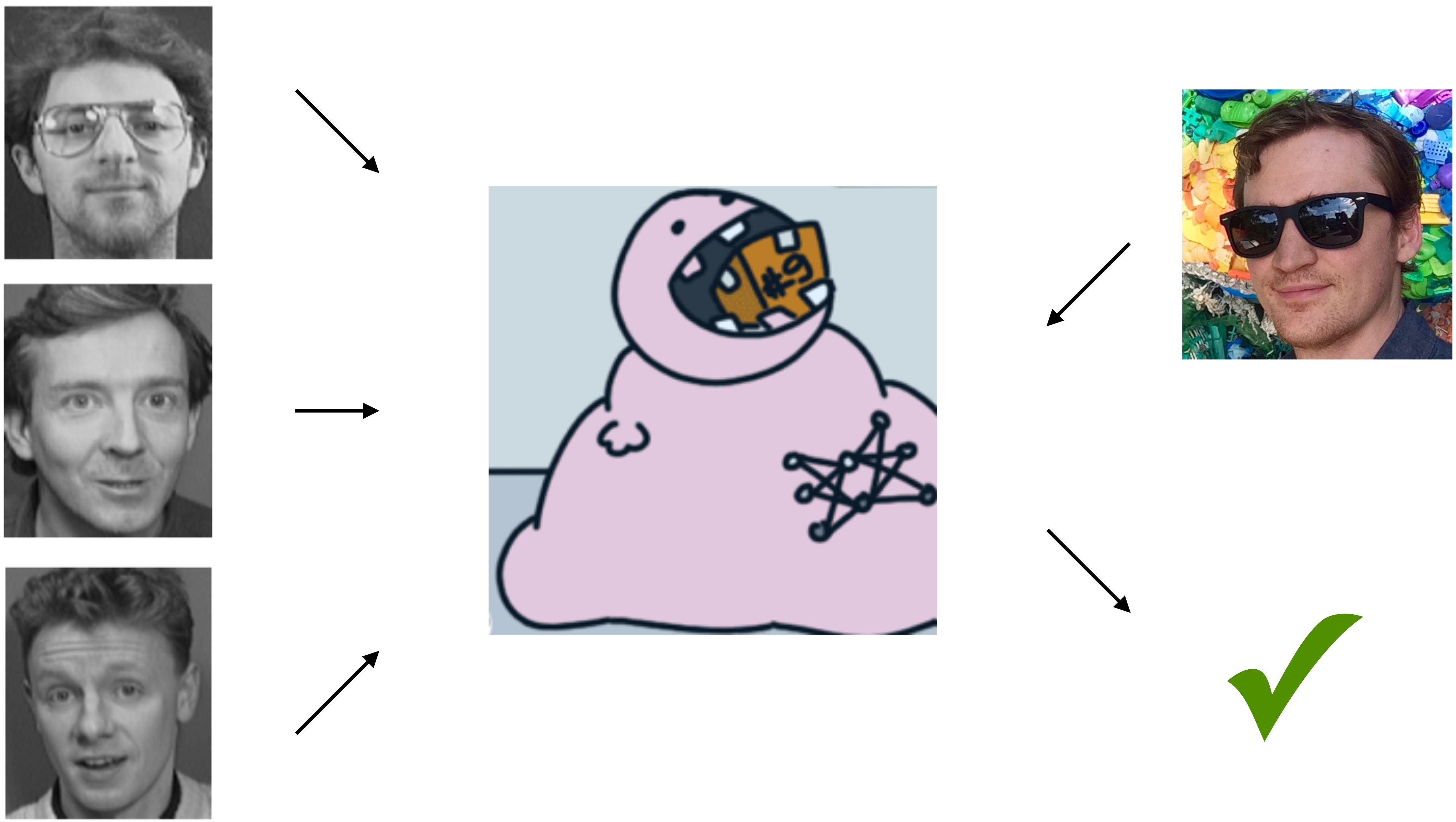
NEW MODEL

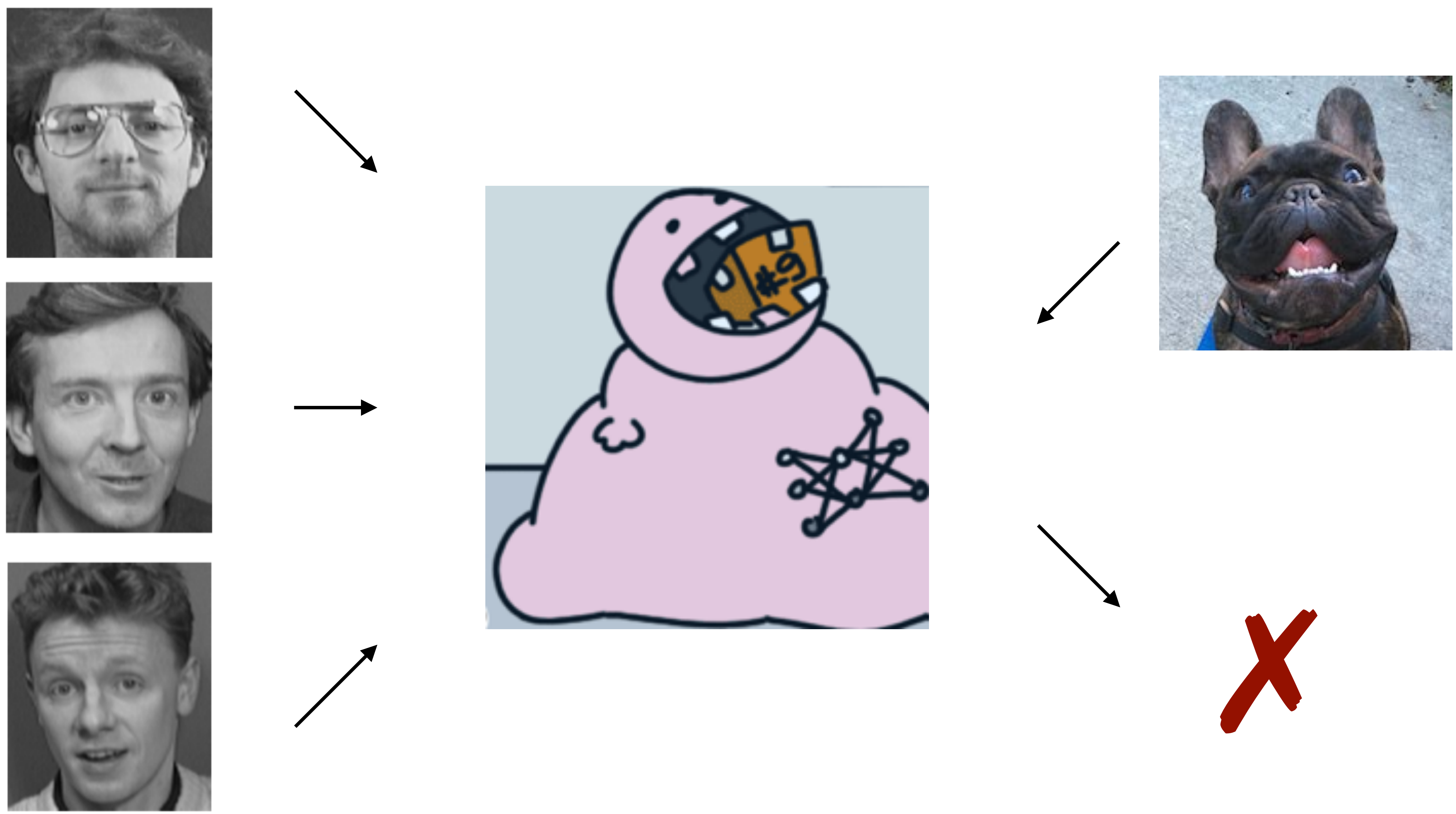


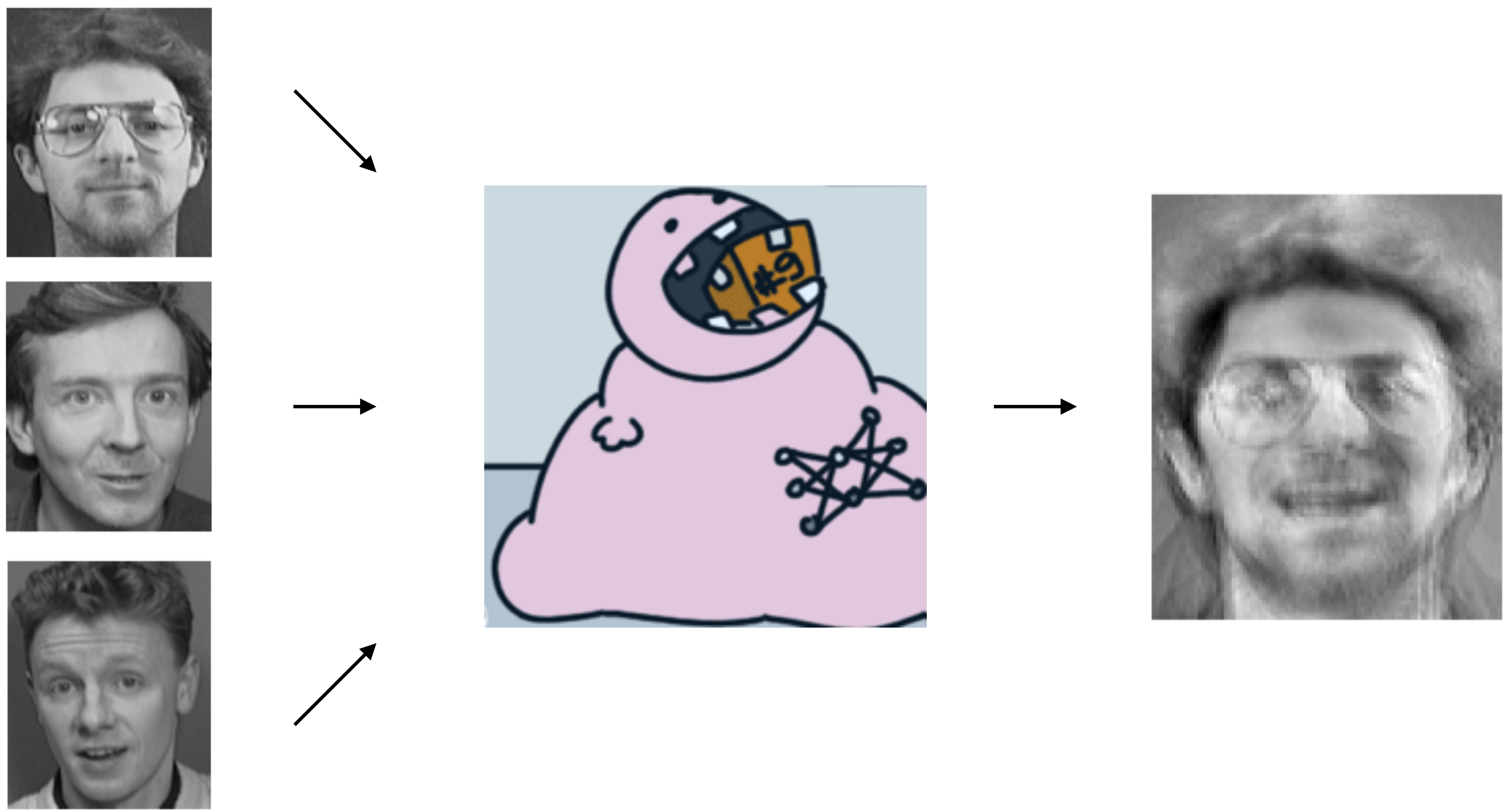
NEW MODEL


TRAINING TIME!
#10
#11
#1
#9
C'MON LEARN FASTER
8
SIGH
5
3, 6, 2!!!
I'LL GO GET THE NEXT ONE
MONKEYUSER.COM


#9







Almost-solution: Aggregate statistics

Clearly not acceptable for small datasets

Clearly acceptable for “well-behaved” massive datasets

Central idea behind modern interpretations of “privacy-sensitive data analysis”

Must be careful with artificial intelligence applications

EU and GDPR

Data breaches (security/access) = financial liability

Fines: MAX(€20 Million , 4% annual global turnover)

Sensitive vs aggregate data – only liable for sensitive

More sensitive data = more financial risk

Aggregate data = cannot be re-identified

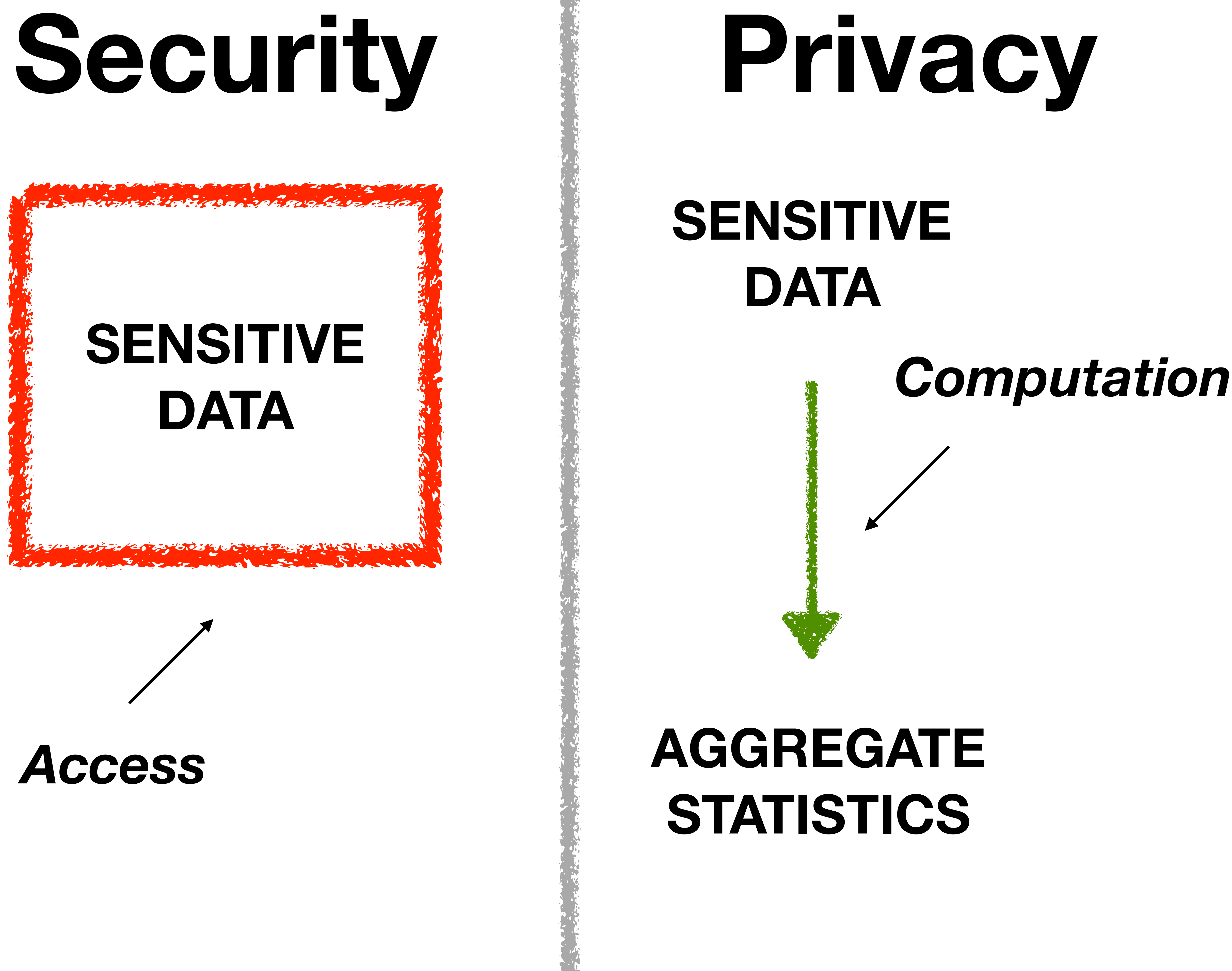
Also: California CCPA modeled on GDPR

Security

Privacy

Security

Privacy


Security
Privacy
SENSITIVE DATA
Access
SENSITIVE DATA
Computation
AGGREGATE STATISTICS

Differential Privacy

=

Aggregate Statistics

+

Random noise

=

No-reidentification guarantees

=

0 Financial liability (GDPR)

Differential Privacy

Program Analysis

Duet

Deep Learning

Differential Privacy

How many people named Éric live in Chile?

How many people named Éric live in Chile?

1. How *sensitive* is this query?

How many people named Éric live in Chile?

1. How *sensitive* is this query?

= 60,000

+ <anyone>

= 60,001

How many people named Éric live in Chile?

1. How *sensitive* is this query?

= 60,000

+ <anyone>

= 60,001

sensitivity = 1

How many people named Éric live in Chile?

2. Add noise to the result with scale ~ sensitivity

= 60,000
+ <noise>

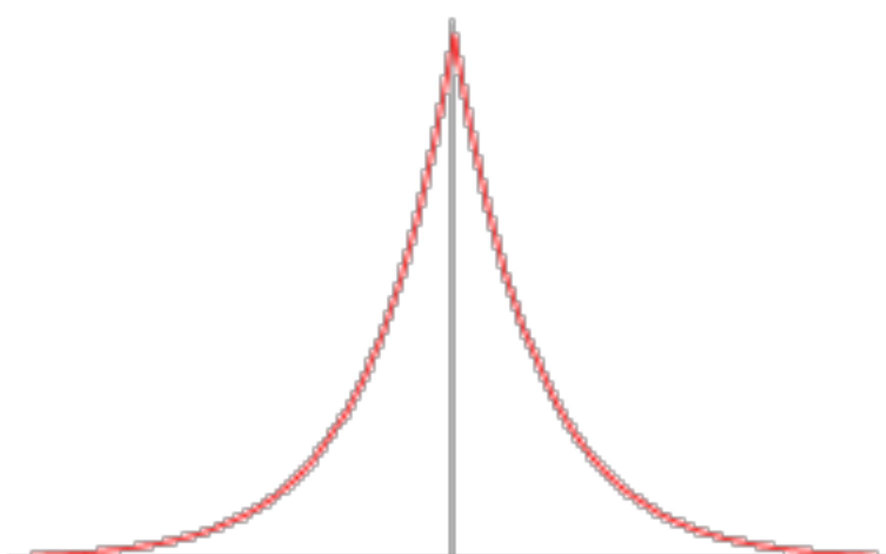
+ <anyone>

= 60,001
+ <noise>

How many people named Éric live in Chile?

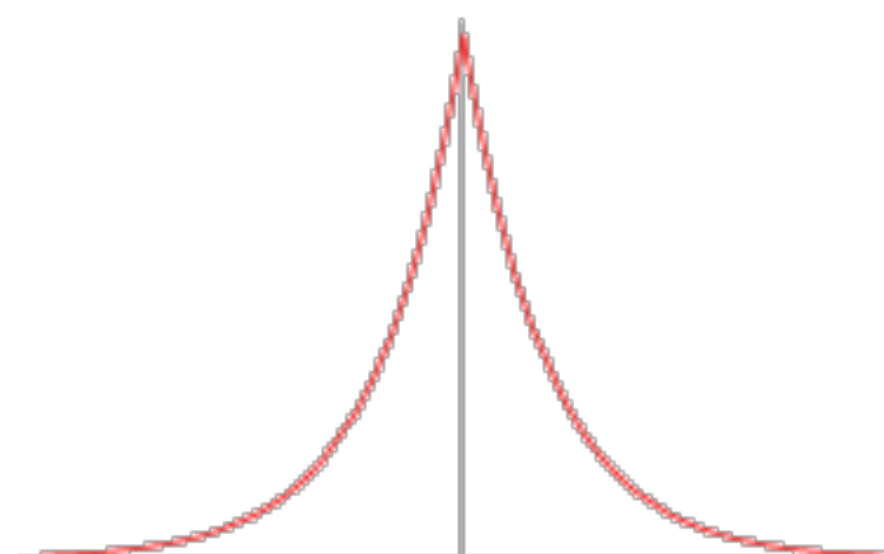
2. Add noise to the result with scale ~ sensitivity

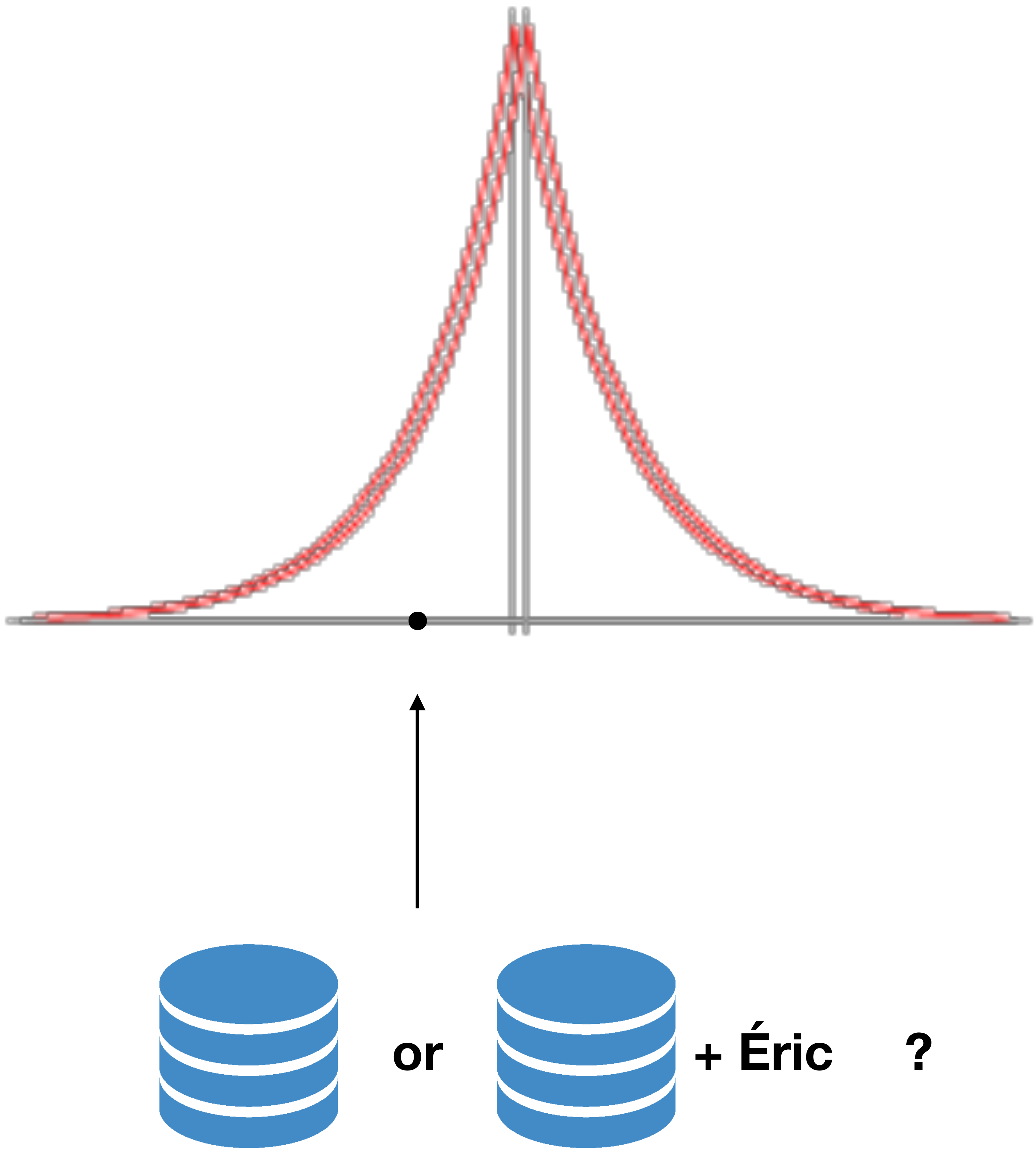
= 60,000
+ <noise>



+ <anyone>

= 60,001
+ <noise>




or
+ Éric
?

How many people named Éric live in Chile?

How many people named Éric live at <specific address>

How many people named Éric live in Chile?

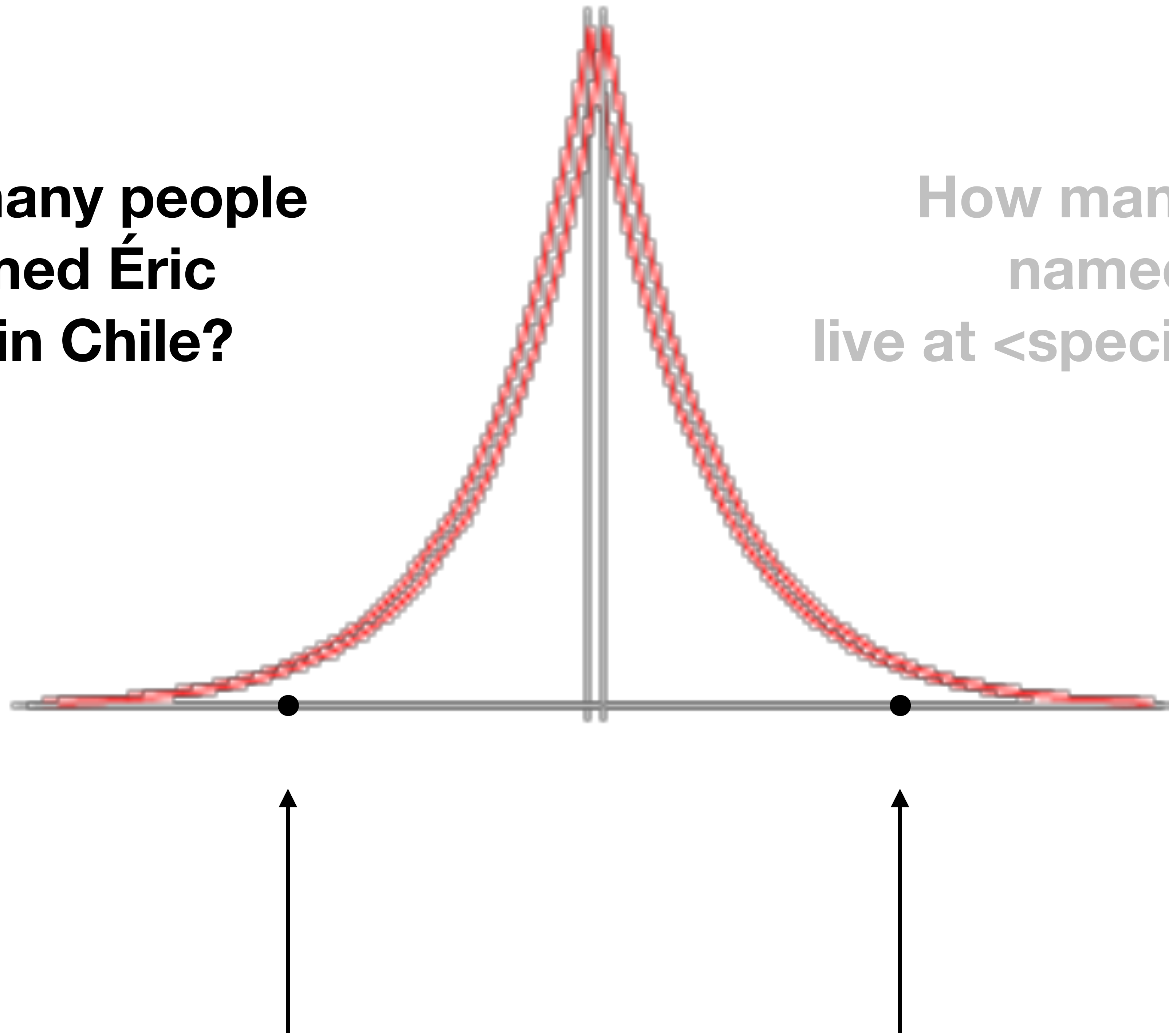
How many people named Éric live at <specific address>

Same sensitivity (= 1)

Same amount of noise

Very different *utility*

How many people named Éric live in Chile?



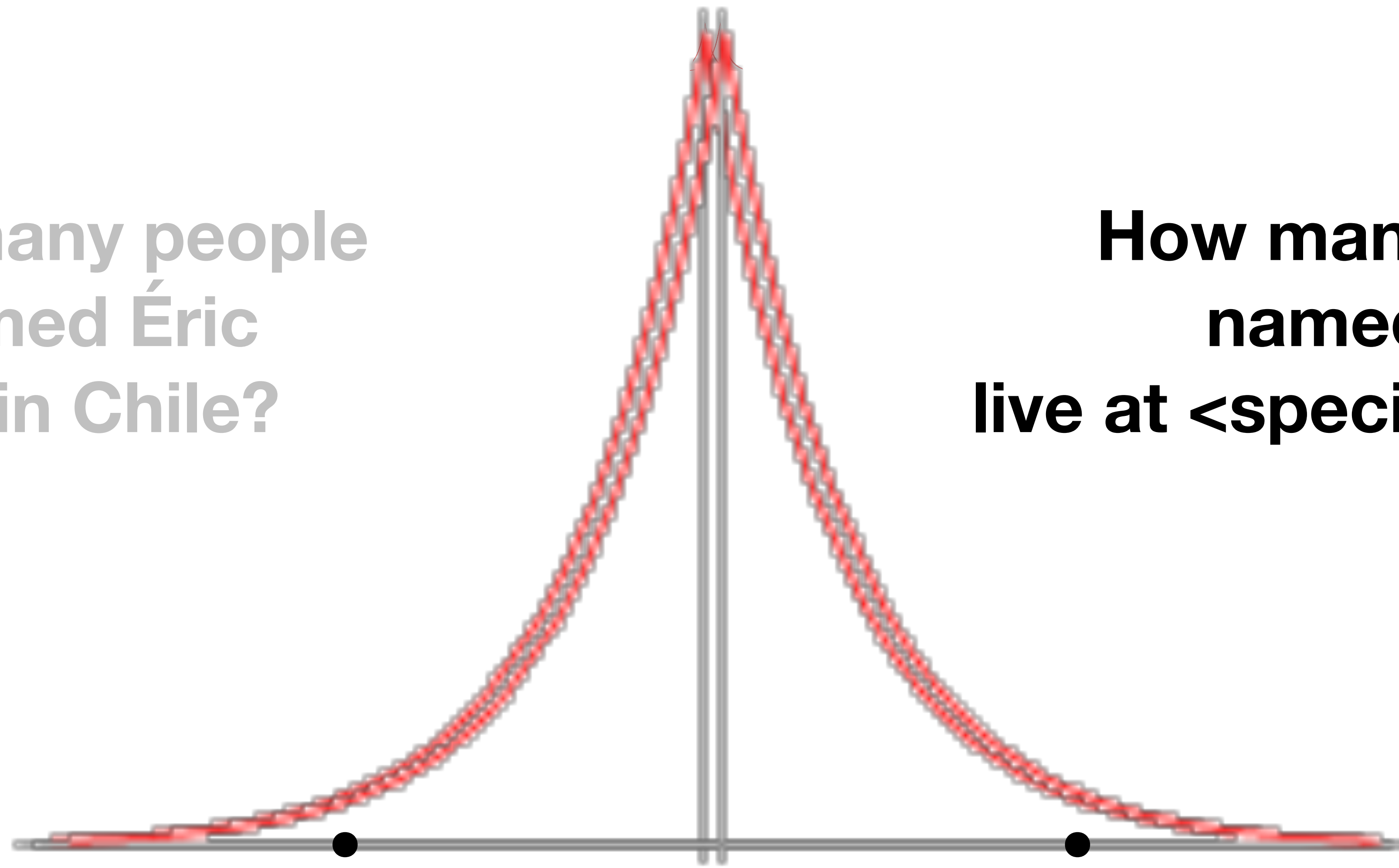
How many people named Éric live at <specific address>

= 59,900

= 60,100

“roughly 60,020 people named Éric live in Chile”

How many people
named Éric
live in Chile?

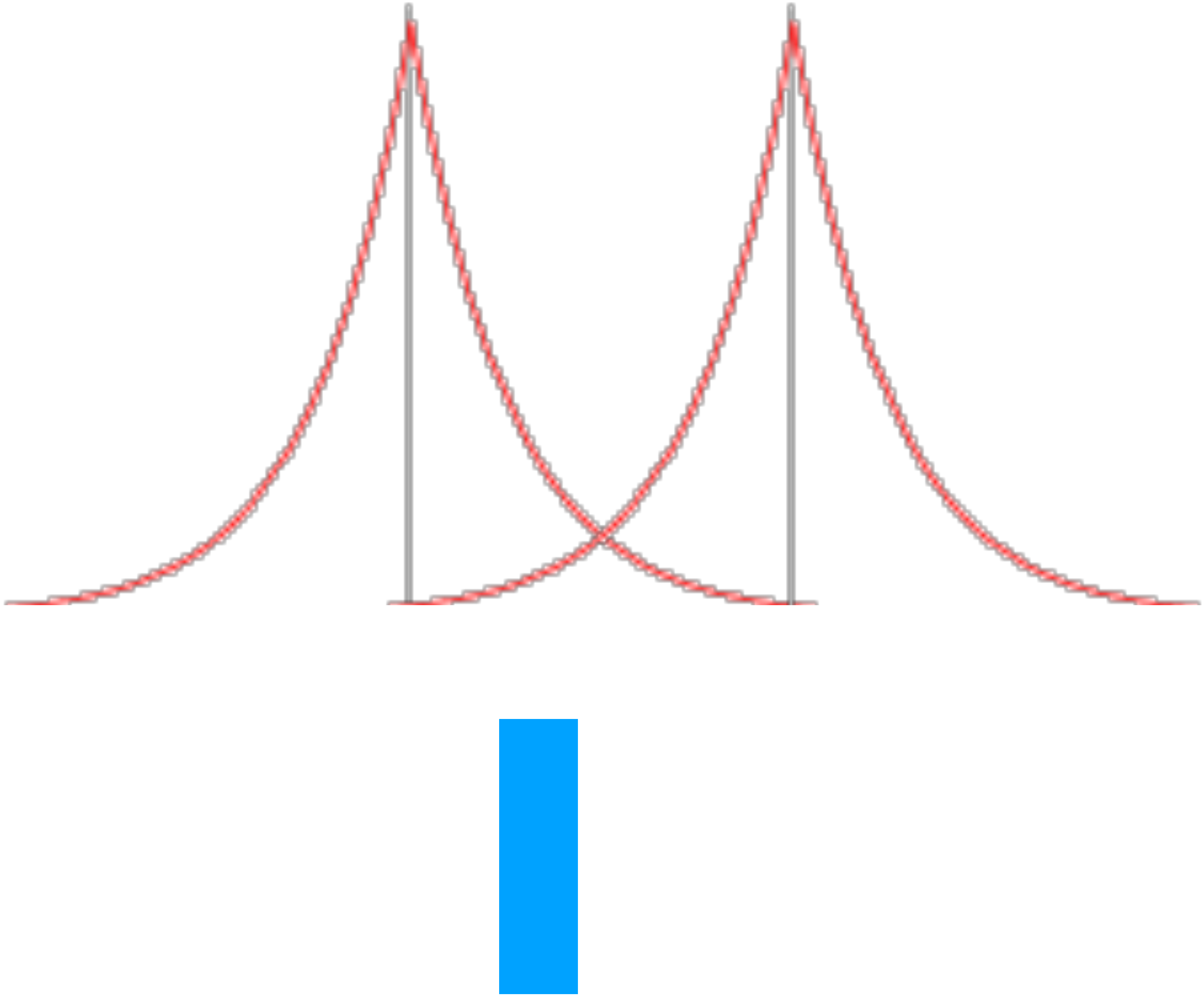


**How many people
named Éric
live at <specific address>**

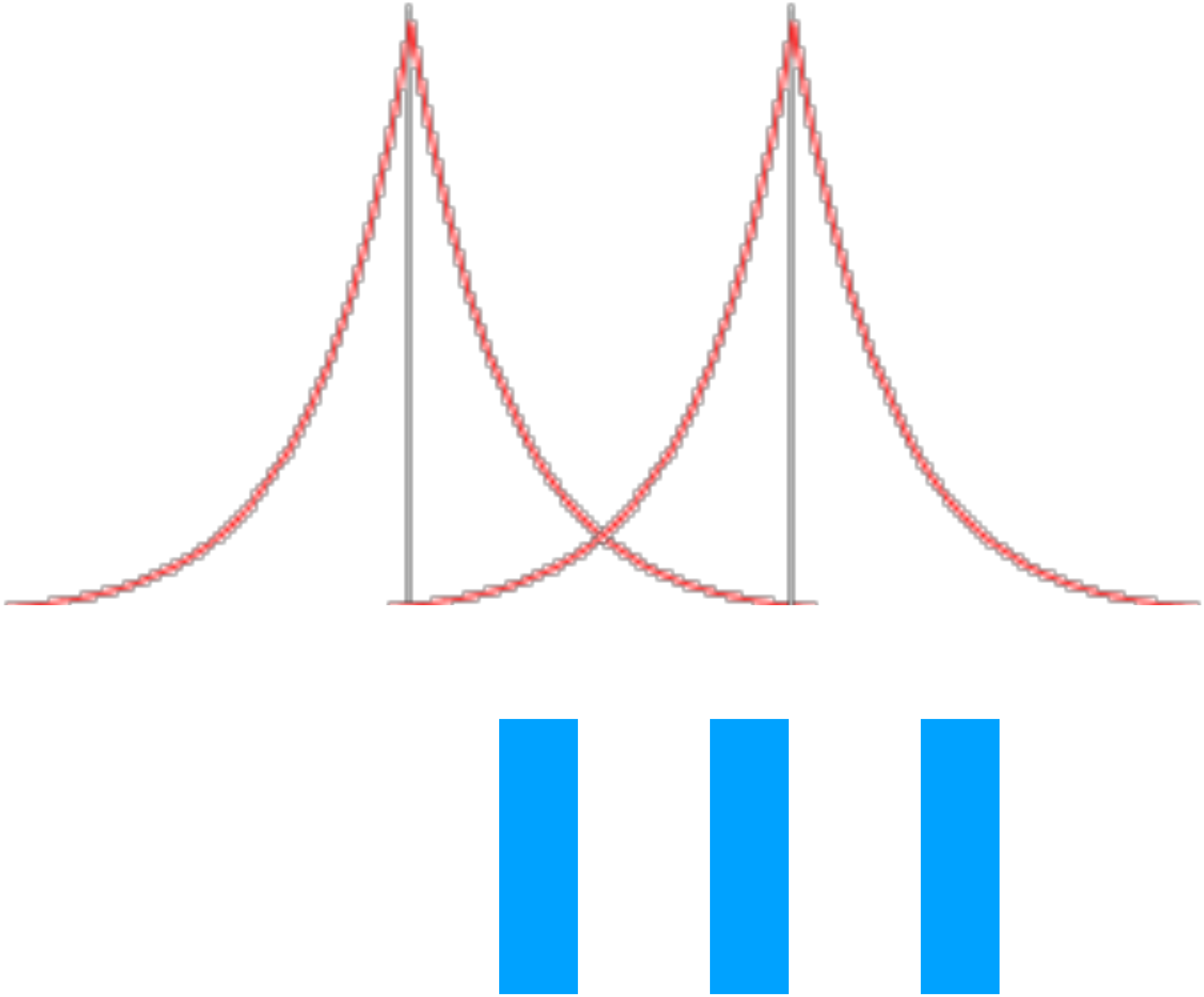
= -100

= 100

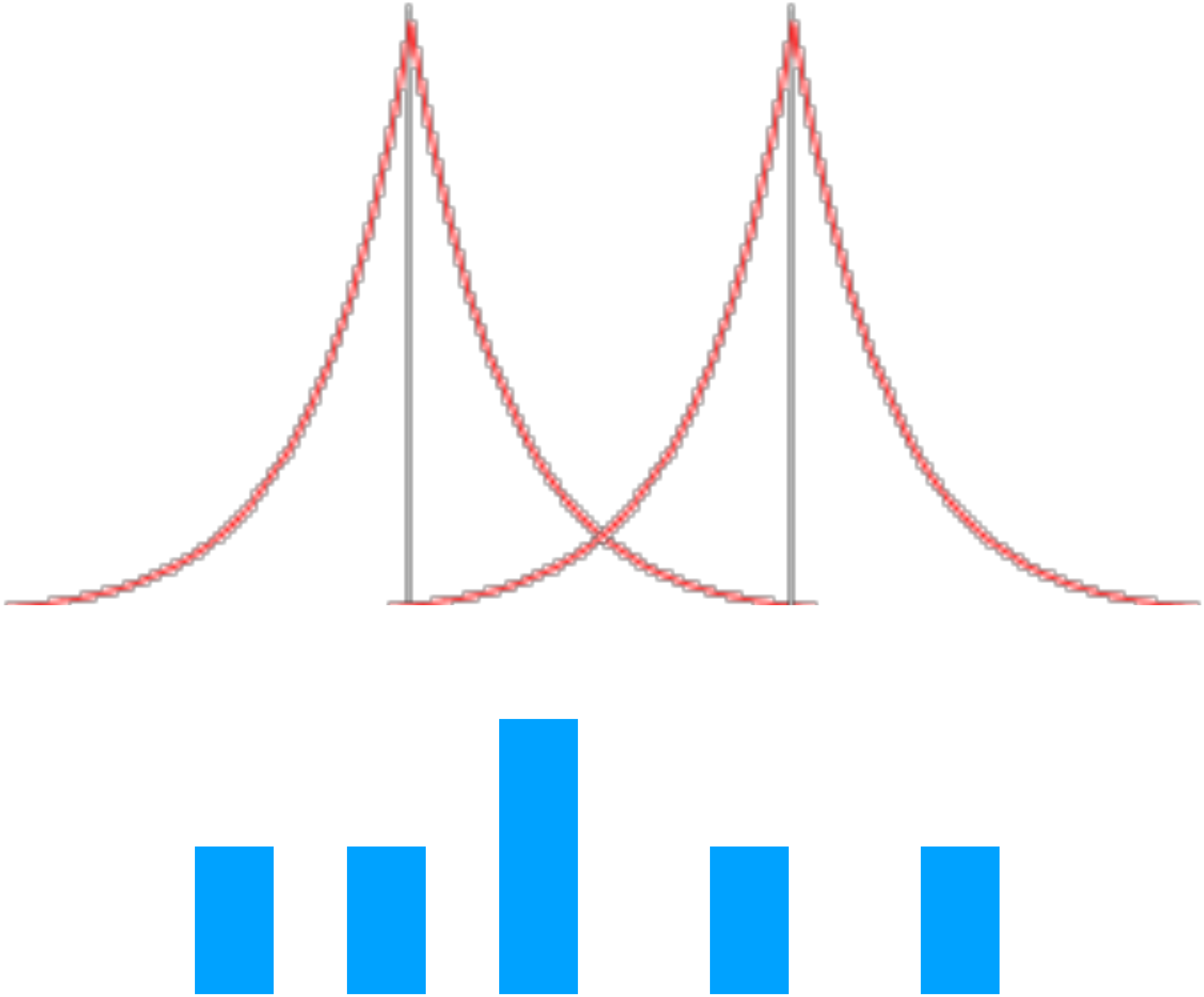
“roughly 37 people named Éric live at <specific address>”



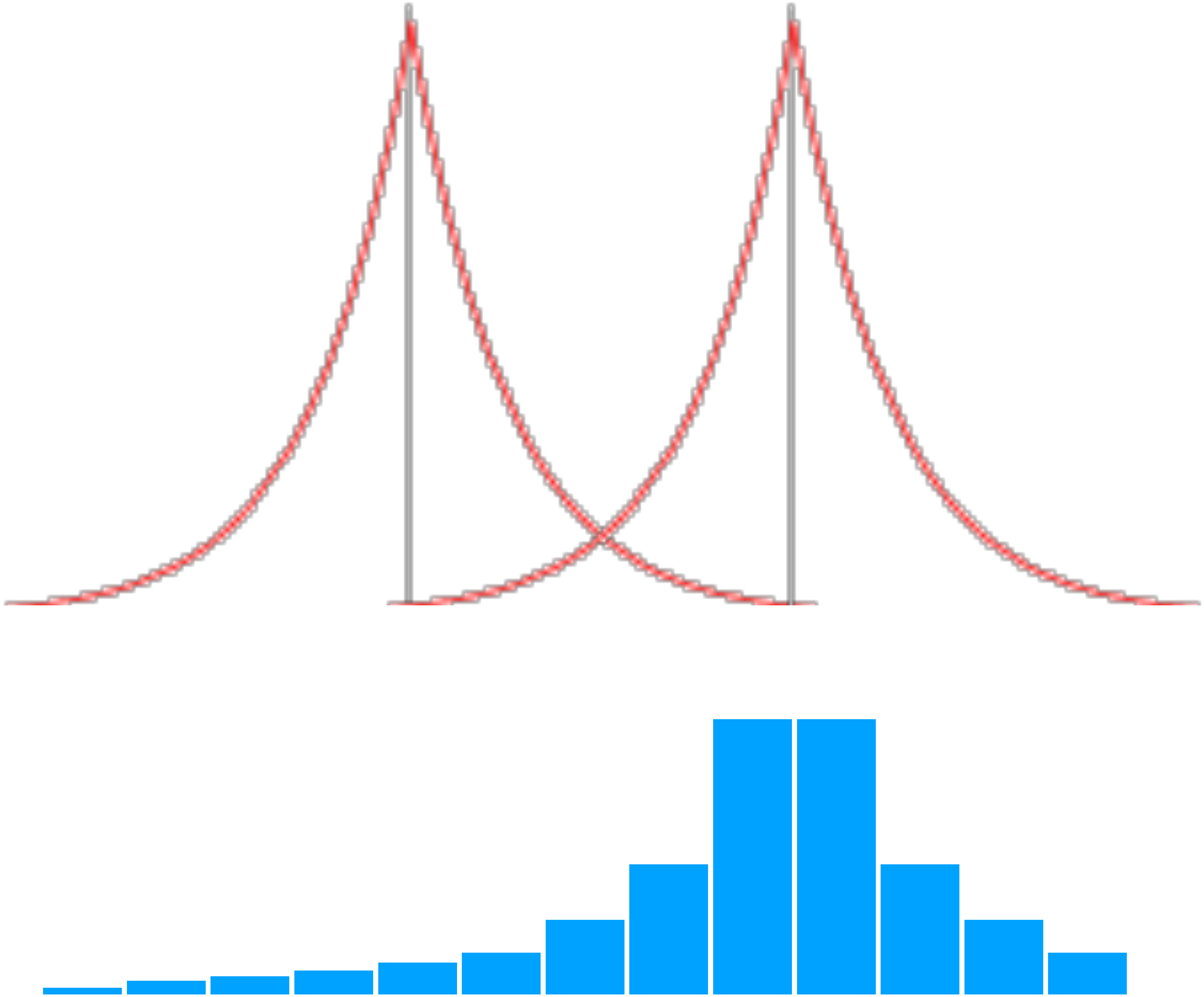
1 Sample



3 Samples



6 Samples



1,000,000 Samples

Privacy Cost

How many samples needed to re-identify participant

Quantity = distance between distributions

Quantity = directly interpretable as privacy “budget”

$$\varepsilon$$

“Differential privacy describes a promise, made by a data holder, or curator, to a data subject: ‘You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.’”

–Dwork & Roth (The Algorithmic Foundations of Differential Privacy)

DP Theorems

Mechanism:
Adding Laplace noise scaled by ~s/ε to an s-sensitive query achieves ε differential privacy

Post-processing:
A differentially private result can be used any number of times, and for any purpose, ***including arbitrary linking with auxiliary data***

Composition:
An ε_1-DP query followed by an ε_2-DP query is $(\varepsilon_1+\varepsilon_2)$-DP

New data = fresh budget

Who is using DP?

Apple

Google

US NIST

US Census

GDPR working documents

DP Challenges

How to achieve better utility/accuracy?

Privacy frameworks (hard proofs):
(ε,δ), Rényi, ZC, TZC

Sensitivity frameworks (hard to compose):
Local sensitivity

Stronger composition (less expressive):
Advanced composition

Smarter “billing” (hard to use):
Independent budget for different sensitive attributes

DP Challenges

What if I don’t trust the computation provider?

Decentralized model:
Local differential privacy

Cryptographic techniques:
Secure multi-party communication, secure enclaves

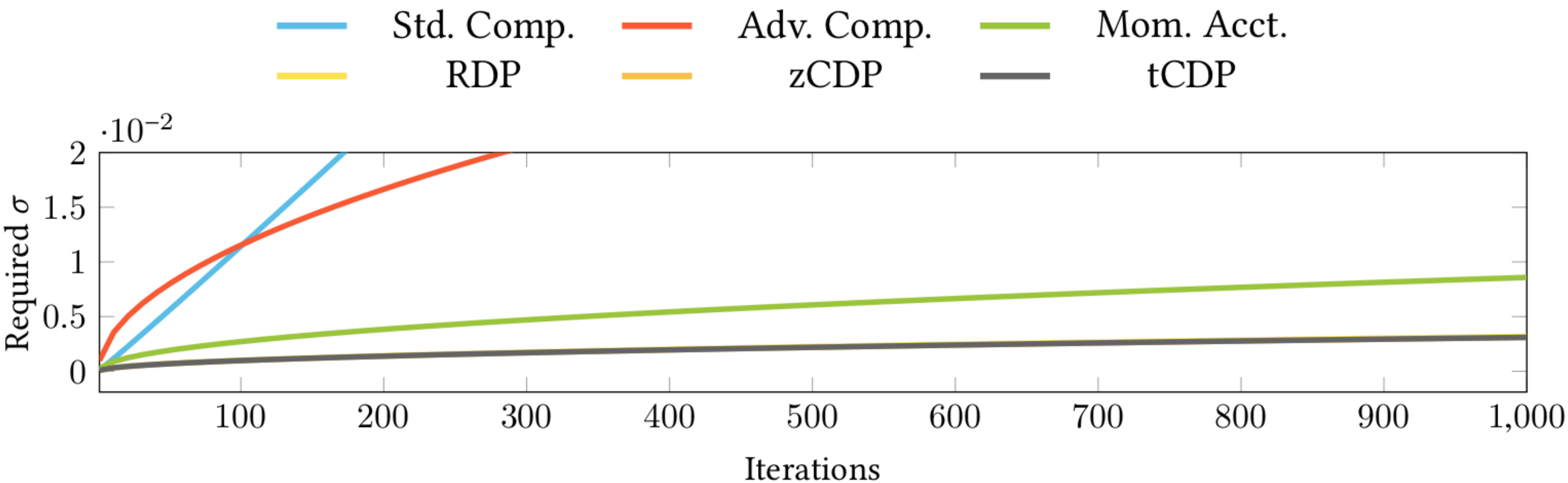


Fig. 7. Noise necessary to achieve $(1, 10^{-5})$-differential privacy for an iterative algorithm (gradient descent) on a dataset of 50,000 samples, under variants of differential privacy. RDP, zCDP, and tCDP all require the same level of noise, and therefore their plots overlap (on the black line).

Differential Privacy

Program Analysis

Duet

Deep Learning

Why Program Analysis

1. How sensitive is the query? (uncomputable in general)

2. Add-noise

3. How private is the result? (uncomputable in general)

PA/PL literature about sensitivity analysis for programs
(assumed: Laplace noise gives ε privacy)
(focus: automation+proofs)

DP literature about privacy analysis for algorithms
(assumed: count query is 1 sensitive)
(focus: precision+proofs)

sensitivity

add
noise
(mechanism)

privacy

Operation	Assumption	Sensitivity
`f(x) = x`		1-sensitive in x
`f(x) = count(x)`		1-sensitive in x
`f(x,y) = x+y`		1-sensitive in x 1-sensitive in y
`f(x,y) = x*y`		∞-sensitive in x ∞-sensitive in y
`f(x) = g(h(x))`	g is α-sensitive h is β-sensitive	αβ sensitive in x

$$f(x,y) = k(g(x) + h(y))$$

$$f(x,y) = k(g(x) + h(y))$$

g is α–sensitive
h is β–sensitive
k is γ–sensitive

$$\Longrightarrow$$

f is $\gamma(\alpha+0)$–sensitive in x
f is $\gamma(0+\beta)$–sensitive in y

$$f \in \mathbb{R} \to^{s} \mathbb{R}$$

Sensitivity

`f(x) is s-sensitive in x iff`

`when` $|v_1 - v_2| \le d$

`then` $|f(v_1) - f(v_2)| \le sd$

When the input wiggles by some amount, how much does the output wiggle.

Sensitivity

$$|4 - 5| = 1 \qquad \in \mathbb{R}$$

Sensitivity

$|4 - 5| = 1 \quad \in \mathbb{R}$

$|$  $-$  $| = 1 \quad \in \mathbb{DB}$

+ Éric

Sensitivity

$$|4 - 5| = 1 \qquad \in \mathbb{R}$$

|  −  | $= 1 \qquad \in \mathbb{DB}$

+ Éric

Arbitrary metric space

Sensitivity

$f(x)$ is s-sensitive in x iff

when $|v_1 - v_2|_{\tau_1} \leq d$

then $|f(v_1) - f(v_2)|_{\tau_2} \leq ds$

When the input wiggles by some amount, how much does the output wiggle.

Privacy

`f(x) is ε-private in x iff`

when $|v_1 - v_2|_{\tau_1} \leq 1$

then $\Pr[f(v_1)] \leq e^{\varepsilon}\Pr[f(v_2)]$

When the input wiggles by one, how close are the resulting distributions.

Privacy

$f(x)$ is ε-private in x iff

when $|v_1 - v_2|_{\tau_1} \leq 1$

$$\text{then} \quad \max \frac{\Pr[f(v_1)]}{\Pr[f(v_2)]} \leq e^{\varepsilon}$$

When the input wiggles by one, how close are the resulting distributions.

Privacy

`f(x) is` ε`-private in x iff`

`when` $|v_1 - v_2|_{\tau_1} \leq 1$

`then` $\max \ln \dfrac{\Pr[f(v_1)]}{\Pr[f(v_2)]} \leq \varepsilon$

When the input wiggles by one, how close are the resulting distributions.

Privacy

`f(x)` is ε-private in `x` iff

when $|v_1 - v_2|_{\tau_1} \leq d$

then $$\max \ln \frac{\Pr[f(v_1)]}{\Pr[f(v_2)]} \leq d\varepsilon$$

When the input wiggles by one, how close are the resulting distributions.

Privacy

$f(x)$ is **ε**-private in x iff

when $|v_1 - v_2|_{\tau_1} \leq$ **d**

then $|f(v_1) - f(v_2)|_D \leq$ **dε**

where

$|f(x) - f(y)|_D \triangleq \max \ln(Pr[f(x)]/Pr[f(y)])$

Privacy = Sensitivity

Privacy Analysis = Sensitivity Analysis

Privacy = Sensitivity

Privacy Analysis = Sensitivity Analysis

$$\texttt{laplace} \in \mathbb{R} \to^{\varepsilon} \mathscr{D}(\mathbb{R})$$

$$\texttt{release} \in \tau \to^{\infty} \mathscr{D}(\tau)$$

$$\texttt{post-pr} \in \mathscr{D}(\tau_1) \, , \, (\tau_1 \to^{\infty} \mathscr{D}(\tau_2)) \\ \to \mathscr{D}(\tau_2)$$

PA Challenges

Complexity (hopefully linear)

Precision (hopefully good)

Expressiveness (objects, HO functions, abstraction)

Exotic DP definitions (no definable metric)

Trust (design? implementation?)

Differential Privacy

Dwork, McSherry, Nissim,Smith–2006

Dwork,Roth–2014

Program Analysis

Reed,Pierce–2010

Duet

Near,Darais,(+9)–2019

Deep Learning

Duet: Goals

Support stronger variants of DP (ε,δ)

Support machine learning algorithms

Precise analysis

Tractable algorithm

Trustworthy design

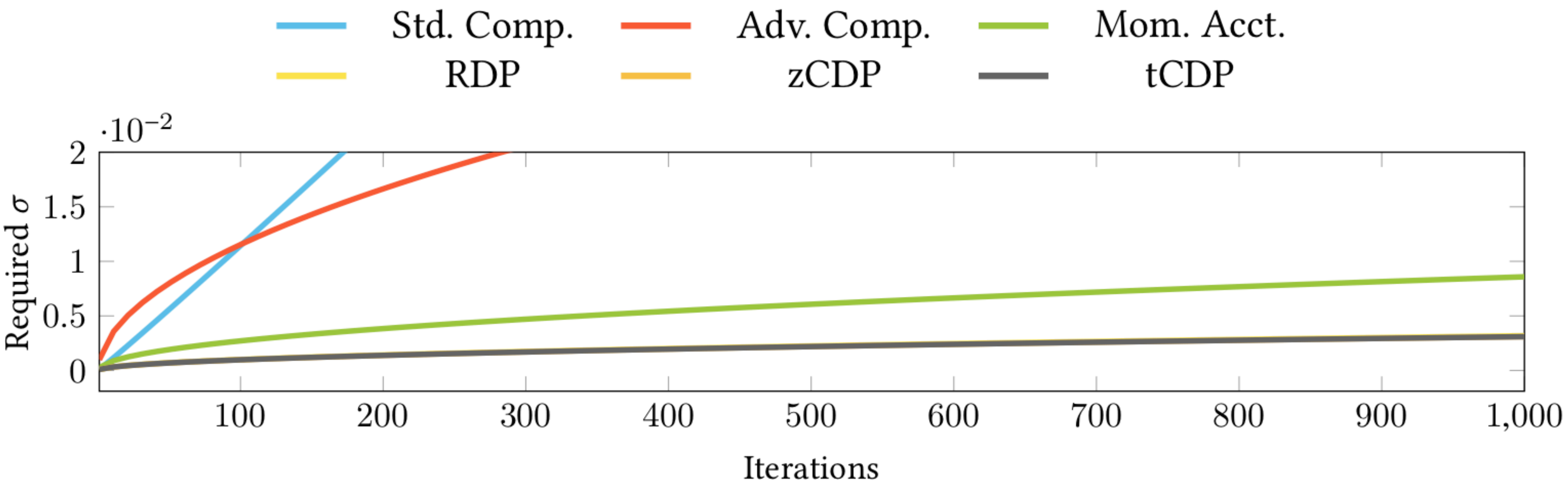


Fig. 7. Noise necessary to achieve $(1, 10^{-5})$-differential privacy for an iterative algorithm (gradient descent) on a dataset of 50,000 samples, under variants of differential privacy. RDP, zCDP, and tCDP all require the same level of noise, and therefore their plots overlap (on the black line).

ε-DP

`f(x) is ε-private in x iff`

when $|v_1 - v_2|_{\tau_1} \leq 1$

then $\Pr[f(v_1)] \leq e^{\varepsilon}\Pr[f(v_2)]$

When the input wiggles by one, how close are the resulting distributions.

(ε,δ)-DP

`f(x) is (ε,δ)-private in x iff`

`when` $|v_1 - v_2|_{\tau_1} \leq 1$

`then` $\Pr[f(v_1)] \leq e^{\varepsilon}\Pr[f(v_2)] + \delta$

*When the input wiggles by one, how close are the resulting distributions, with high (1-**δ**) probability.*

ε-DP

(ε,δ)-DP

$$\frac{\texttt{f} \in \mathbb{R} \rightarrow^{2} \mathbb{R} \qquad \texttt{laplace} \in \mathbb{R} \rightarrow^{\varepsilon} \mathscr{D}(\mathbb{R})}{\texttt{laplace}\circ\texttt{f} \in \mathbb{R} \rightarrow^{2\varepsilon} \mathscr{D}(\mathbb{R})}$$

ε-DP

$$\frac{f \in \mathbb{R} \rightarrow^{2} \mathbb{R} \qquad \texttt{laplace} \in \mathbb{R} \rightarrow^{\varepsilon} \mathscr{D}(\mathbb{R})}{\texttt{laplace}\circ f \in \mathbb{R} \rightarrow^{2\,\varepsilon} \mathscr{D}(\mathbb{R})}$$

(ε,δ)-DP

$$\frac{f \in \mathbb{R} \rightarrow^{2} \mathbb{R} \qquad \texttt{gauss} \in \mathbb{R} \rightarrow^{\varepsilon,\delta} \mathscr{D}(\mathbb{R})}{\texttt{gauss}\circ f \in \mathbb{R} \rightarrow^{2\varepsilon,2e^{\varepsilon}\delta} \mathscr{D}(\mathbb{R})}$$

Duet Design

Scaling is *very* imprecise, language should disallow it

In previous analyses, scaling is pervasive–no way out

We separate languages for **sensitivity** and **privacy**

Add APIs for data analysis and machine learning

Proofs of privacy for any “well-typed” program

$\multimap^{*}$-E

$$\frac{\Gamma \vdash e : (\tau_1 @ p_1, ..., \tau_n @ p_n) \multimap^{*} \tau}{}$$

$\multimap^{*}$-E

$$\frac{\Gamma \vdash e : (\tau_1 @ p_1, \ldots, \tau_n @ p_n) \multimap^{*} \tau \qquad \rceil \Gamma_1 \lceil^{1} \vdash e_1 : \tau_1 \quad \cdots \quad \rceil \Gamma_n \lceil^{1} \vdash e_n : \tau_n}{}$$

$\multimap^{*}$-E

$$\frac{\Gamma \vdash e : (\tau_1 @ p_1, \ldots, \tau_n @ p_n) \multimap^{*} \tau \qquad \rceil \Gamma_1 \lceil^{1} \vdash e_1 : \tau_1 \qquad \cdots \qquad \rceil \Gamma_n \lceil^{1} \vdash e_n : \tau_n}{\rceil \Gamma \lceil^{\infty} + \rceil \Gamma_1 \lceil^{p_1} + \cdots + \rceil \Gamma_n \lceil^{p_n} \vdash e(e_1, \ldots, e_n) : \tau}$$

$$\mathrm{L}\nabla_{\ell}^{g}[\underline{\theta}; \underline{X}, \underline{y}]$$

$$\mathrm{L}\nabla^{g}_{\ell}[\underline{\theta}; \underline{X}, \underline{y}] : \underbrace{\mathbb{M}^{\ell}\,[1, n]\,\mathbb{R}}_{\theta} \multimap_{\infty}$$

$$\mathrm{L}\nabla^{g}_{\ell}[\underline{\theta};\underline{X},\underline{y}] : \mathbb{M}^{\ell}\,[1,n]\,\mathbb{R} \multimap_{\infty} \underbrace{\mathbb{M}^{\ell}\,[1,n]\,\mathbb{D}}_{X} \multimap_{1}$$

$$\mathrm{L}\nabla^{g}_{\ell}[\underline{\theta}; \underline{X}, \underline{y}] : \mathbb{M}^{\ell}\,[1, n]\,\mathbb{R} \multimap_{\infty} \mathbb{M}^{\ell}\,[1, n]\,\mathbb{D} \multimap_{1} \underbrace{\mathbb{D}}_{y} \multimap_{1}$$

$$L\nabla^{g}_{\ell}[\underline{\theta}; \underline{X}, \underline{y}] : \mathbb{M}^{\ell}\,[1, n]\ \mathbb{R} \multimap_{\infty} \mathbb{M}^{\ell}\ [1, n]\ \mathbb{D} \multimap_{1} \mathbb{D} \multimap_{1} \overbrace{\mathbb{M}^{\cup}_{\ell}[1, n]\ \mathbb{R}}^{\texttt{how to improve }\theta\texttt{?}}$$

$$\mathrm{L}\nabla^{g}_{\ell}[\underline{\theta}; \underline{X}, \underline{y}] : \mathbb{M}^{\ell}[1, n]\ \mathbb{R} \multimap_{\infty} \mathbb{M}^{\ell}[1, n]\ \mathbb{D} \multimap_{1} \mathbb{D} \multimap_{1} \mathbb{M}^{\text{U}}_{\ell}[1, n]\ \mathbb{R}$$

ℓ = L2-norm

required by mgauss DP-mechanism

model
data
labels

$$
\begin{array}{l}
\mathrm{noisy\text{-}grad}(\theta, X, y, \epsilon, \delta) \triangleq \\
\quad \texttt{let } s = \mathbb{R}[1.0]/\texttt{real}\ (\mathrm{rows}\ X)\ \texttt{in} \\
\quad \texttt{let } z = \mathrm{zeros}\ (\mathrm{cols}\ X)\ \texttt{in} \\
\quad \texttt{let } gs = \mathrm{mmap\text{-}row}\ (s\lambda\ X_i\ y_i \Rightarrow \\
\qquad\qquad\qquad \mathrm{L}\nabla^{\mathsf{LR}}_{\mathsf{L2}}[\theta; X_i, y_i])\ X\ y\ \texttt{in} \\
\quad \texttt{let } g = \mathrm{fld\text{-}row}\ (s\lambda\ x_1\ x_2 \Rightarrow x_1 + x_2)\ z\ gs\ \texttt{in} \\
\quad \texttt{let } g_s = \mathrm{map}\ (s\lambda\ x \Rightarrow s \cdot x)\ g\ \texttt{in} \\
\quad \texttt{mgauss}[s, \epsilon, \delta]\ {<}X, y{>}\ \{g_s\}
\end{array}
$$

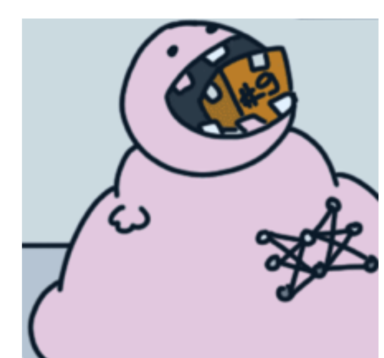
model

data

labels

$$
\begin{aligned}
&\text{noisy-grad}(\theta, X, y, \epsilon, \delta) \triangleq \\
&\quad \texttt{let } s = \mathbb{R}[1.0]/\texttt{real } (\text{rows } X) \texttt{ in} \\
&\quad \texttt{let } z = \text{zeros } (\text{cols } X) \texttt{ in} \\
&\quad \texttt{let } gs = \text{mmap-row } (s\lambda\ X_i\ y_i \Rightarrow \\
&\qquad\qquad\qquad \text{L}\nabla^{\mathsf{LR}}_{\mathsf{L2}}[\theta; X_i, y_i])\ X\ y \texttt{ in} \\
&\quad \texttt{let } g = \text{fld-row } (s\lambda\ x_1\ x_2 \Rightarrow x_1 + x_2)\ z\ gs \texttt{ in} \\
&\quad \texttt{let } g_s = \text{map } (s\lambda\ x \Rightarrow s \cdot x)\ g \texttt{ in} \\
&\quad \texttt{mgauss}[s, \epsilon, \delta]\ {<}X, y{>}\ \{g_s\}
\end{aligned}
$$

how to improve the model



$$
\begin{aligned}
&\text{noisy-gradient-descent}(\overset{\textbf{data}}{X}, \overset{\textbf{labels}}{y}, \overset{\textbf{iters}}{k}, \overset{\textbf{rate}}{\eta}, \epsilon, \delta) \triangleq \\
&\quad \texttt{let } X_1 = \text{box } (\text{mclip}^{L2}\ X) \texttt{ in} \\
&\quad \texttt{let } \theta_0 = \text{zeros } (\text{cols } X_1) \texttt{ in} \\
&\quad \texttt{loop}[\delta'] \ k \ \texttt{on } \theta_0 \ {<}X_1, y{>} \ \{t, \theta \Rightarrow \\
&\qquad g_p \leftarrow \text{noisy-grad } \theta \ (\text{unbox } X_1) \ y \ \epsilon \ \delta \ ; \\
&\qquad \texttt{return } \theta - \eta \cdot g_p \quad \}
\end{aligned}
$$

data labels iters rate

$$\text{noisy-gradient-descent}(X, y, k, \eta, \epsilon, \delta) \triangleq$$

$$\quad \texttt{let } X_1 = \text{box } (\text{mclip}^{L2}\ X) \texttt{ in}$$

$$\quad \texttt{let } \theta_0 = \text{zeros } (\text{cols } X_1) \texttt{ in}$$

baby model

$$\quad \texttt{loop}[\delta'] \ k \texttt{ on } \theta_0 <X_1, y> \{t, \theta \Rightarrow$$

$$\qquad g_p \leftarrow \text{noisy-grad } \theta \ (\text{unbox } X_1) \ y \ \epsilon \ \delta \ ;$$

$$\qquad \texttt{return } \theta - \eta \cdot g_p \quad \}$$

data iters

labels rate

$$
\begin{aligned}
&\mathrm{noisy\text{-}gradient\text{-}descent}(X, y, k, \eta, \epsilon, \delta) \triangleq \\
&\quad \mathtt{let}\ X_1 = \mathrm{box}\ (\mathrm{mclip}^{L2}\ X)\ \mathtt{in} \\
&\quad \mathtt{let}\ \theta_0 = \mathrm{zeros}\ (\mathrm{cols}\ X_1)\ \mathtt{in} \\
&\quad \mathtt{loop}[\delta']\ k\ \mathtt{on}\ \theta_0 <X_1, y> \{t, \theta \Rightarrow \\
&\qquad g_p \leftarrow \mathrm{noisy\text{-}grad}\ \theta\ (\mathrm{unbox}\ X_1)\ y\ \epsilon\ \delta\ ; \\
&\qquad \mathtt{return}\ \theta - \eta \cdot g_p \quad \}
\end{aligned}
$$

baby model

smarter model

$$
\begin{aligned}
&\text{noisy-gradient-descent}(X, y, k, \eta, \epsilon, \delta) \triangleq \\
&\quad \texttt{let } X_1 = \text{box } (\text{mclip}^{L2}\ X) \texttt{ in} \\
&\quad \texttt{let } \theta_0 = \text{zeros } (\text{cols } X_1) \texttt{ in} \\
&\quad \texttt{loop}[\delta'] \ k \texttt{ on } \theta_0 < X_1, y > \{t, \theta \Rightarrow \\
&\qquad\quad g_p \leftarrow \text{noisy-grad } \theta \ (\text{unbox } X_1) \ y \ \epsilon \ \delta \ ; \\
&\qquad\quad \texttt{return } \theta - \eta \cdot g_p \quad \}
\end{aligned}
$$

Guaranteed Privacy =

$$
(2\epsilon\sqrt{2k\log(1/\delta')}, k\delta + \delta')
$$

```
frank-wolfe X y k ϵ δ ≜
  let X₁ = clip-matrix_L∞ X in
  let d = cols X in
  let θ₀ = zeros d in
  let idxs = mcreate_L∞[1,2·d]{i,j ⇒
               ⟨j mod d,sign(j − d)⟩} in
  loop [δ] k on θ₀ {t, θ ⇒
    let μ = 1.0/((real t) + 2.0) in
    let g = L∇^LR_L∞[θ; X₁, y] in
    ⟨i, s⟩ ← exponential[1/(rows X₁), ϵ] idxs {⟨i, s⟩ ⇒
               s · g#[0, i]} ;
    let g_p = (zeros d)#[0, i ↦ s · 100] in
    return ((1.0 − μ) · θ) + (μ · g_p) }
```

Privacy = $\left(2\epsilon\sqrt{2k\log(1/\delta)}, \delta\right)$

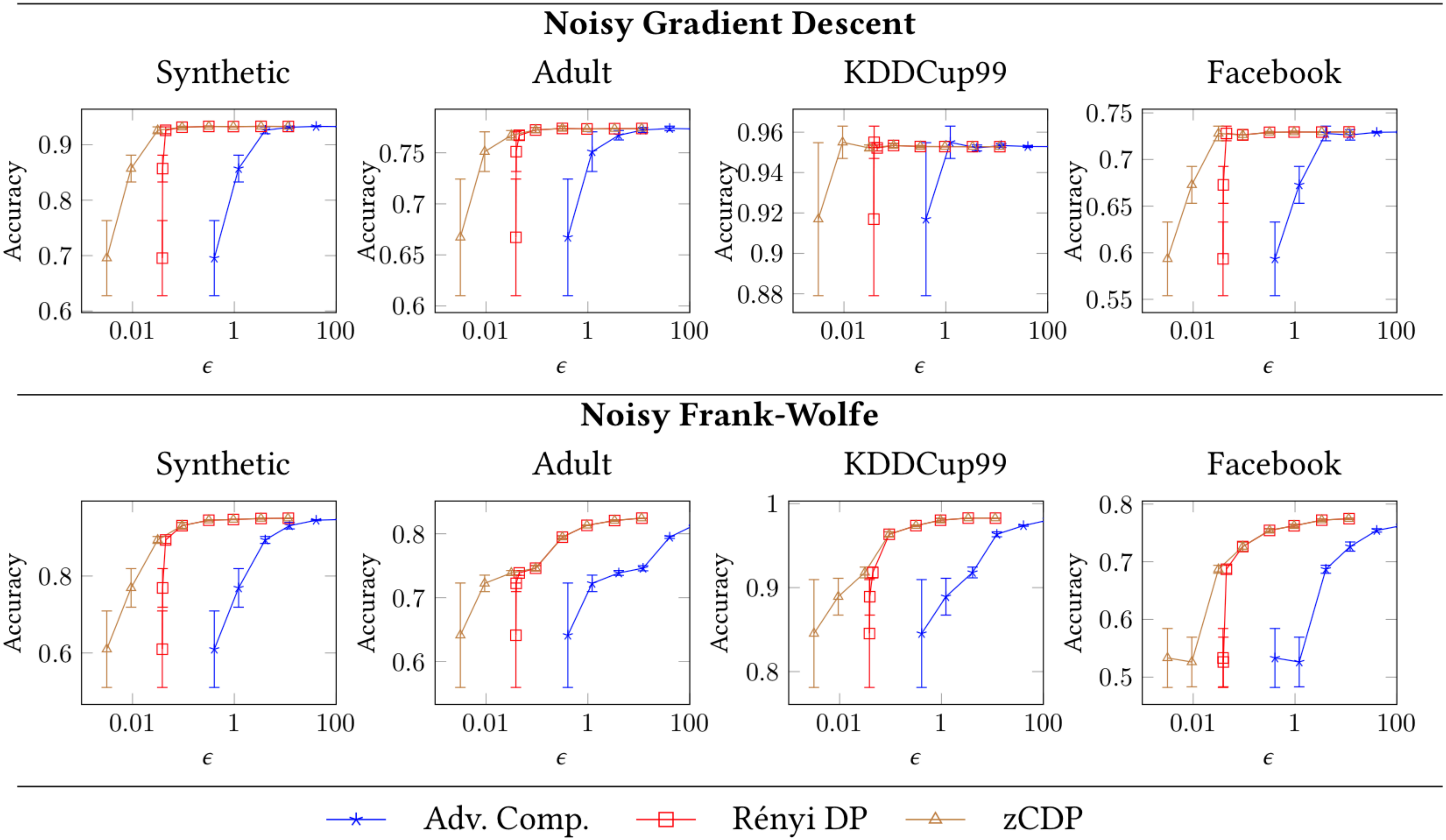


Fig. 10. Accuracy Results for Noisy Gradient Descent (Top) and Noisy Frank-Wolfe (Bottom).

Technique	Ref.	§	Privacy Concept
Optimization Algorithms			
Noisy Gradient Descent	[7, 39]	6.1	Composition
Gradient Descent w/ Output Perturbation	[43]	6.2	Parallel Composition (sensitivity)
Noisy Frank-Wolfe	[40]	6.3	Exponential mechanism
Variations on Gradient Descent			
Minibatching	[7]	6.4	Amplification by subsampling
Parallel-composition minibatching	—	6.5	Parallel composition
Gradient clipping	[3]	6.6	Sensitivity bounds
Preprocessing & Deployment			
Hyperparameter tuning	[11]	A.1	Exponential mechanism
Adaptive clipping	—	6.7	Sparse Vector Technique
Z-Score normalization	[2]	A.2	Composition
Combining All of the Above		6.8	Composition

Technique	LOC	Time (ms)
Noisy G.D.	23	0.51ms
G.D. + Output Pert.	25	0.39ms
Noisy Frank-Wolfe	31	0.59ms
Minibatching	26	0.51ms
Parallel minibatching	42	0.65ms
Gradient clipping	21	0.40ms
Hyperparameter tuning	125	3.87ms
Adaptive clipping	68	1.01ms
Z-Score normalization	104	1.51ms

Duet will be open source on GitHub (soon)

Differential Privacy

Program Analysis

Duet

Deep Learning

Deep Learning

Gradients:

- Bounded sensitivity for convex systems
- Unbounded sensitivity for non-convex systems

Deep Learning:

- Non-convex

State of the art:

- Aggressive clipping during training (to bound sensitivity)

Deep Learning

Recent results:

Local sensitivity + smoothness instead of GS

Analytical derivative can bound LS + smoothness

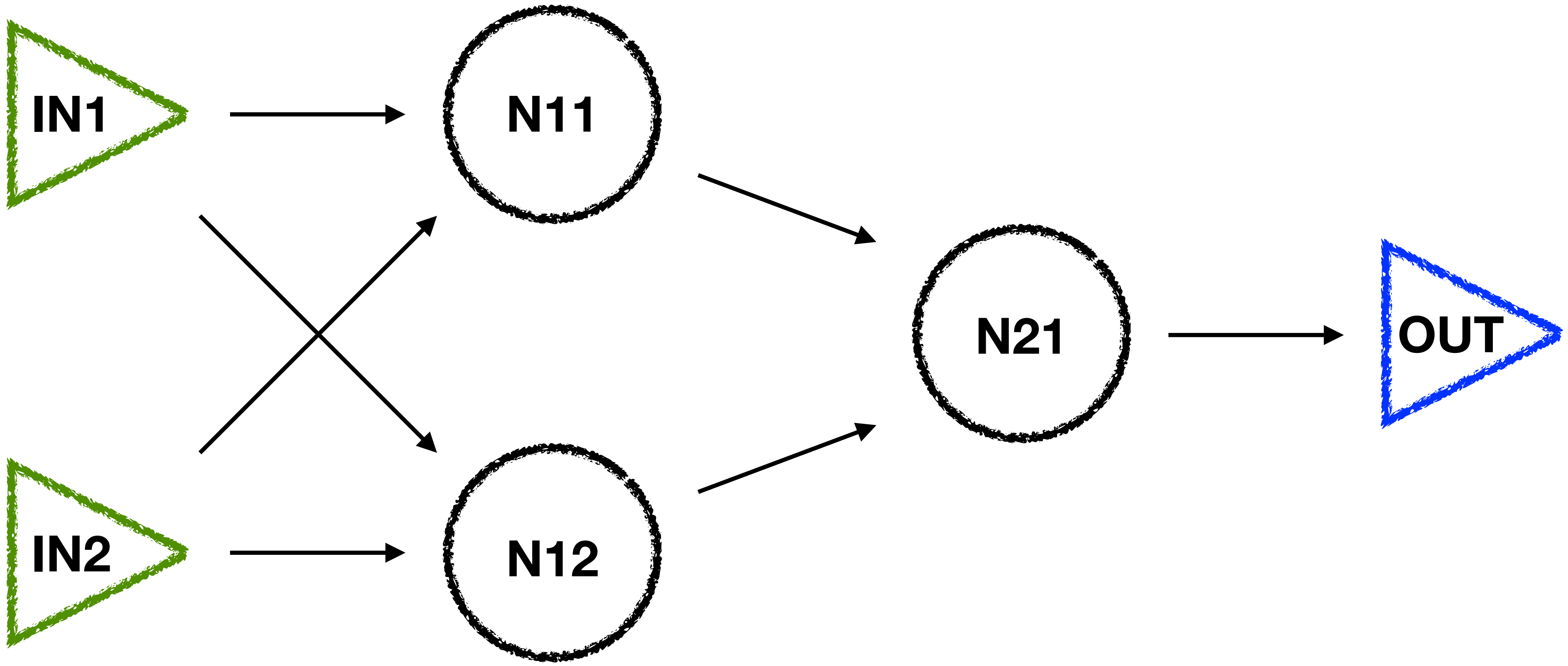
Hypothesis:

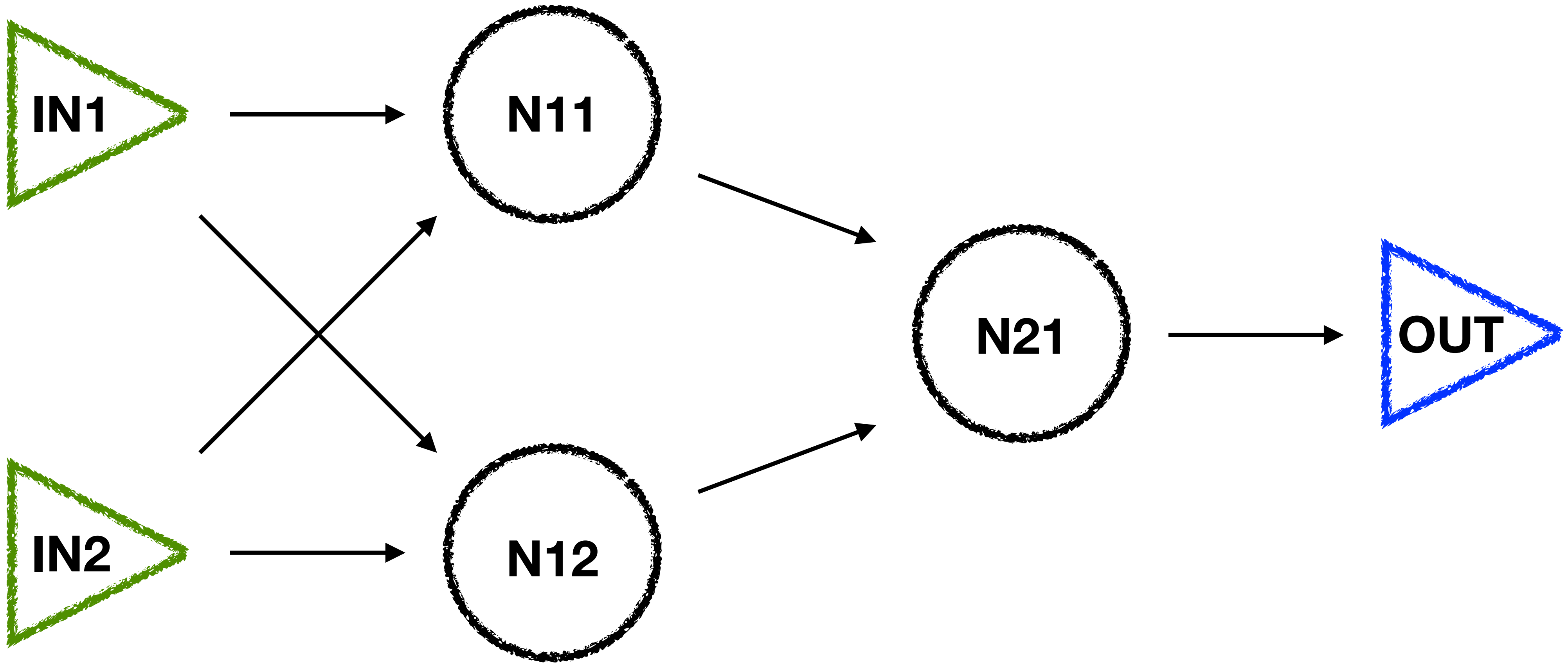
Local sensitivity + smoothness for neural networks

Gradient of the gradient via AD^2

Compositional smoothness analysis

Improved accuracy over naive clipping


IN1
IN2
N11
N12
N21
OUT




```
n11 = relu(w111*in1 + w112*in2)
n12 = relu(w121*in1 + w122*in2)
n21 = sigm(w211*n11 + w212*n12)
return n21
```

Neural Networks

```
n11 = relu(w111*in1 + w112*in2)
n12 = relu(w121*in1 + w122*in2)
n21 = sigm(w211*n11 + w212*n12)
return n21
```

first order, stateless programs with free variables (weights)

no branching control flow

differentiable

NN Training

Analytic gradient used for training

Efficient automatic differentiation algorithms (backprop)

We need gradient (for local sensitivity) *of the gradient*

Run backprop again – 2nd order gradient

AD

Forward mode (1st derivative): dual numbers `<v,d>`

Forward mode (2nd derivative): ternary numbers $<v,d_1,d_2>$

Reverse mode (1st derivative): forward backward passes

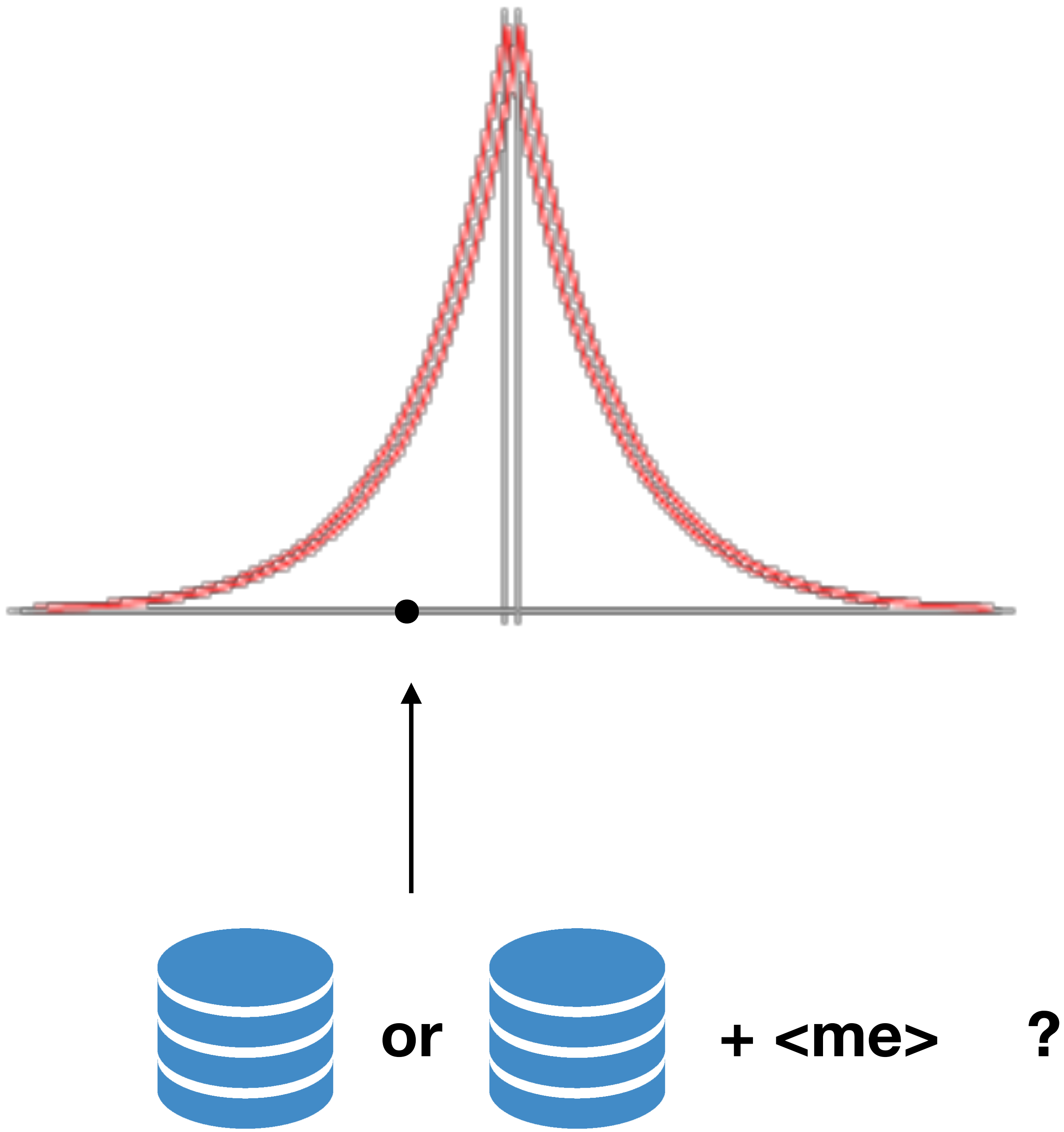
Reverse mode (2nd derivative): FBFB passes

(+ smoothness analysis)

Duet Collaborators

JOSEPH P. NEAR, University of Vermont
CHIKE ABUAH, University of Vermont
TIM STEVENS, University of Vermont
PRANAV GADDAMADUGU, University of California, Berkeley
LUN WANG, University of California, Berkeley
NEEL SOMANI, University of California, Berkeley
MU ZHANG, Cornell University
NIKHIL SHARMA, University of California, Berkeley
ALEX SHAN, University of California, Berkeley
DAWN SONG, University of California, Berkeley

Duet: PL for DP



Machine Learning Algorithm =

$$
\begin{array}{l}
\text{noisy-gradient-descent}(X, y, k, \eta, \epsilon, \delta) \triangleq \\
\quad \texttt{let } X_1 = \text{box } (\text{mclip}^{L2}\ X) \texttt{ in} \\
\quad \texttt{let } \theta_0 = \text{zeros } (\text{cols } X_1) \texttt{ in} \\
\quad \texttt{loop}[\delta']\ k \texttt{ on } \theta_0 <X_1, y> \{t, \theta \Rightarrow \\
\qquad g_p \leftarrow \text{noisy-grad } \theta\ (\text{unbox } X_1)\ y\ \epsilon\ \delta\ ; \\
\qquad \texttt{return } \theta - \eta \cdot g_p \quad \}
\end{array}
$$

Guaranteed Privacy =

$$(2\epsilon\sqrt{2k\log(1/\delta')}, k\delta + \delta')$$

(END)