Google Research

DP-fy your DATA: How to (and why) Synthesize Differentially Private Synthetic Data

Google Research

Alex
Bie

Peter
Kairouz

Natalia
Ponomareva

Sergei
Vassilvitskii

Researchers and Engineers at Google Research

Google Research

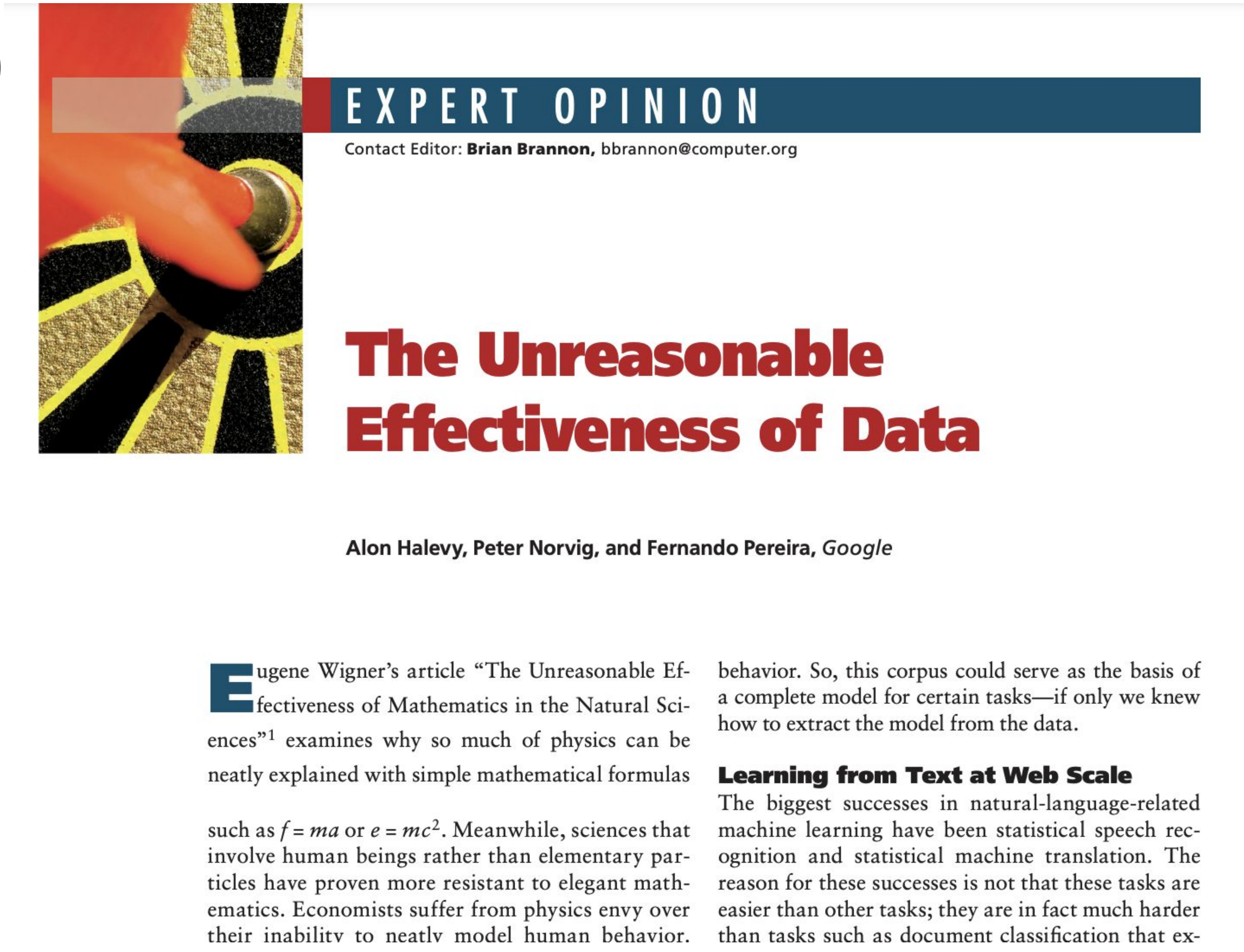
Plan for Today

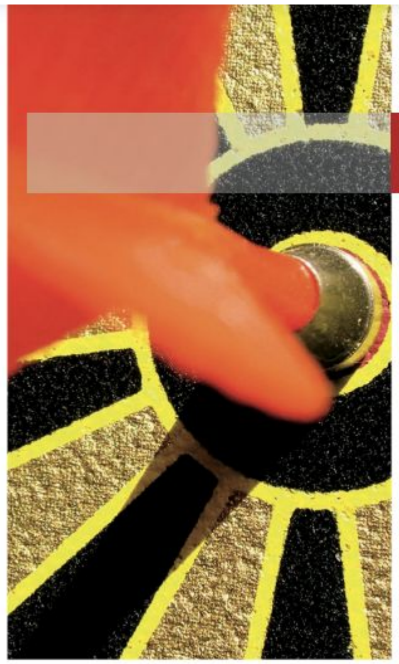
Google Research

Data In Machine Learning

The Role of Data 2009

"For many tasks, words and word combinations provide all the representational machinery we need to learn from text."





EXPERT OPINION

Contact Editor: **Brian Brannon,** bbrannon@computer.org

The Unreasonable Effectiveness of Data

Alon Halevy, Peter Norvig, and Fernando Pereira, *Google*

Eugene Wigner's article "The Unreasonable Effectiveness of Mathematics in the Natural Sciences"[1] examines why so much of physics can be neatly explained with simple mathematical formulas such as $f = ma$ or $e = mc^2$. Meanwhile, sciences that involve human beings rather than elementary particles have proven more resistant to elegant mathematics. Economists suffer from physics envy over their inability to neatly model human behavior. behavior. So, this corpus could serve as the basis of a complete model for certain tasks—if only we knew how to extract the model from the data.

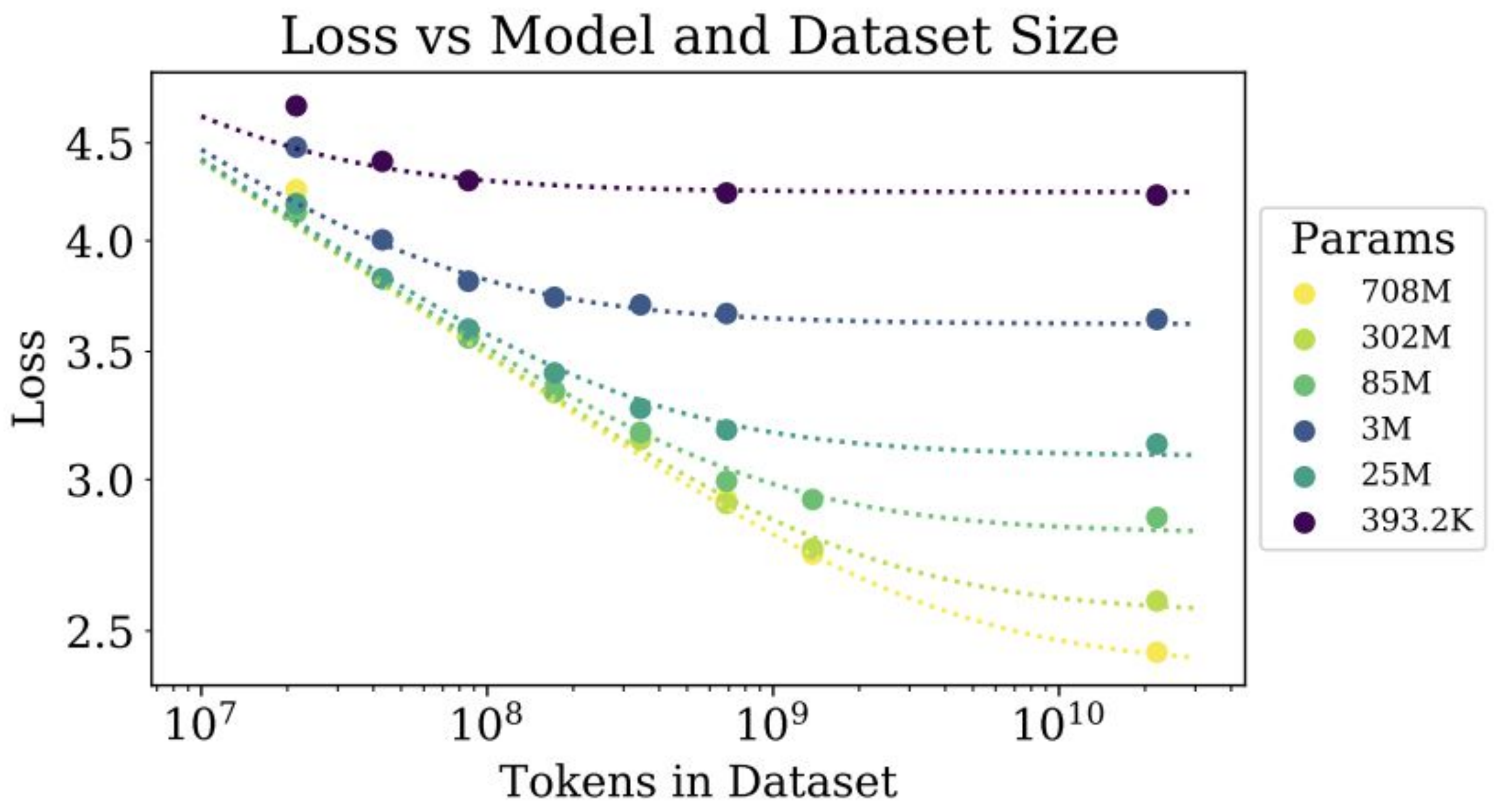
Learning from Text at Web Scale

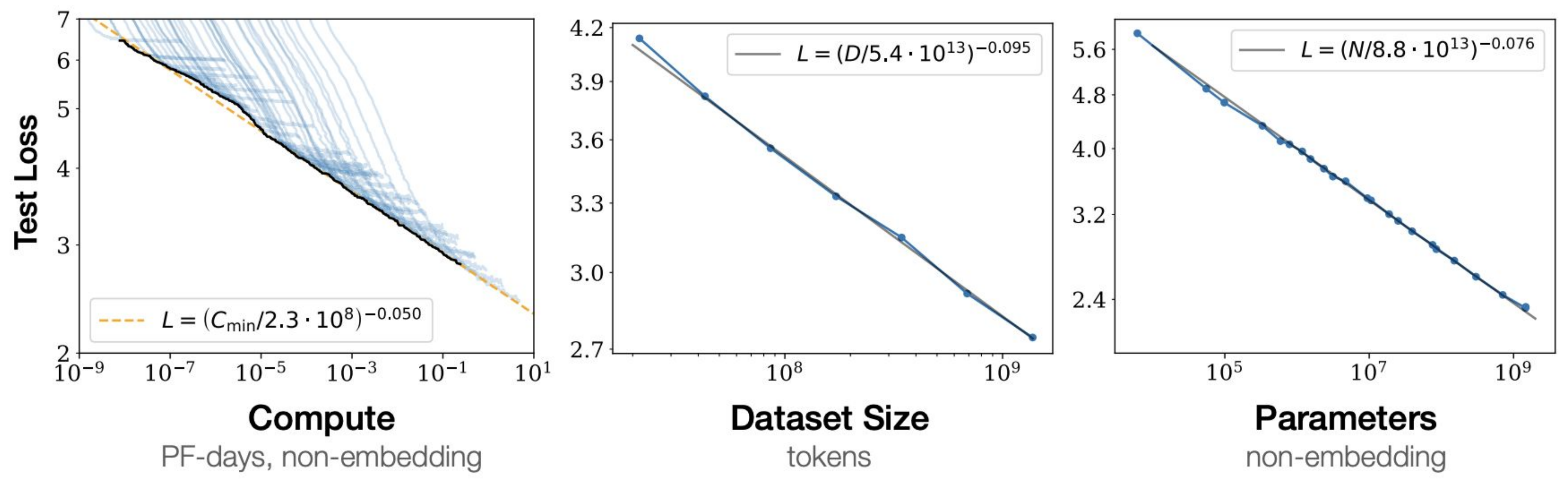
The biggest successes in natural-language-related machine learning have been statistical speech recognition and statistical machine translation. The reason for these successes is not that these tasks are easier than other tasks; they are in fact much harder than tasks such as document classification that ex-

The Role of Data: 2020+

Scaling Laws:

Capture the precise trade-off between model quality, training dataset size, and computation necessar





But need good data hygiene

Data comes with restrictions on use:

- Some data may be personal: names, phone numbers, locations, …
- Some data may be regulated: GDPR, AIA, …

For LLMs especially:

- Larger data sets are more likely to include private information by chance
- Models are large, opaque, and difficult to understand directly
- GenAI outputs are rich enough (text, images) to potentially include detailed private information

LONG LIVE THE REVOLUTION. OUR NEXT MEETING WILL BE AT THE DOCKS AT MIDNIGHT ON JUNE 28 TAB
AHA, FOUND THEM!

WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.

The Promise of Synthetic Data

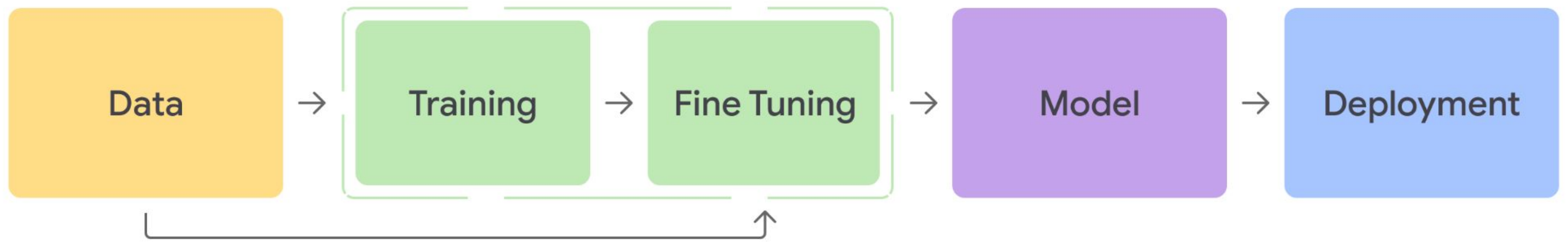
What if…

- Can generate new data
- That is similar in distribution to given data
- But is entirely synthetic and new…

Can we train models on synthetic data instead?

Why Synthetic Data?

Change the beginning of the usual pipeline



By changing the data, keep the rest of the pipeline the same:

- Same architecture
- Same optimizers
- Same deployment checks
- Same …

Types of Synthetic Data

Pure Synthetic Data

DP Synthetic Data

Pure Synthetic Data: Homework for LLMs

“Synthetic data generation using large language models (LLMs) offers a powerful solution to a commonly faced problem: the availability of high-quality, diverse, and privacy-compliant data.”

Synthetic data is used to:

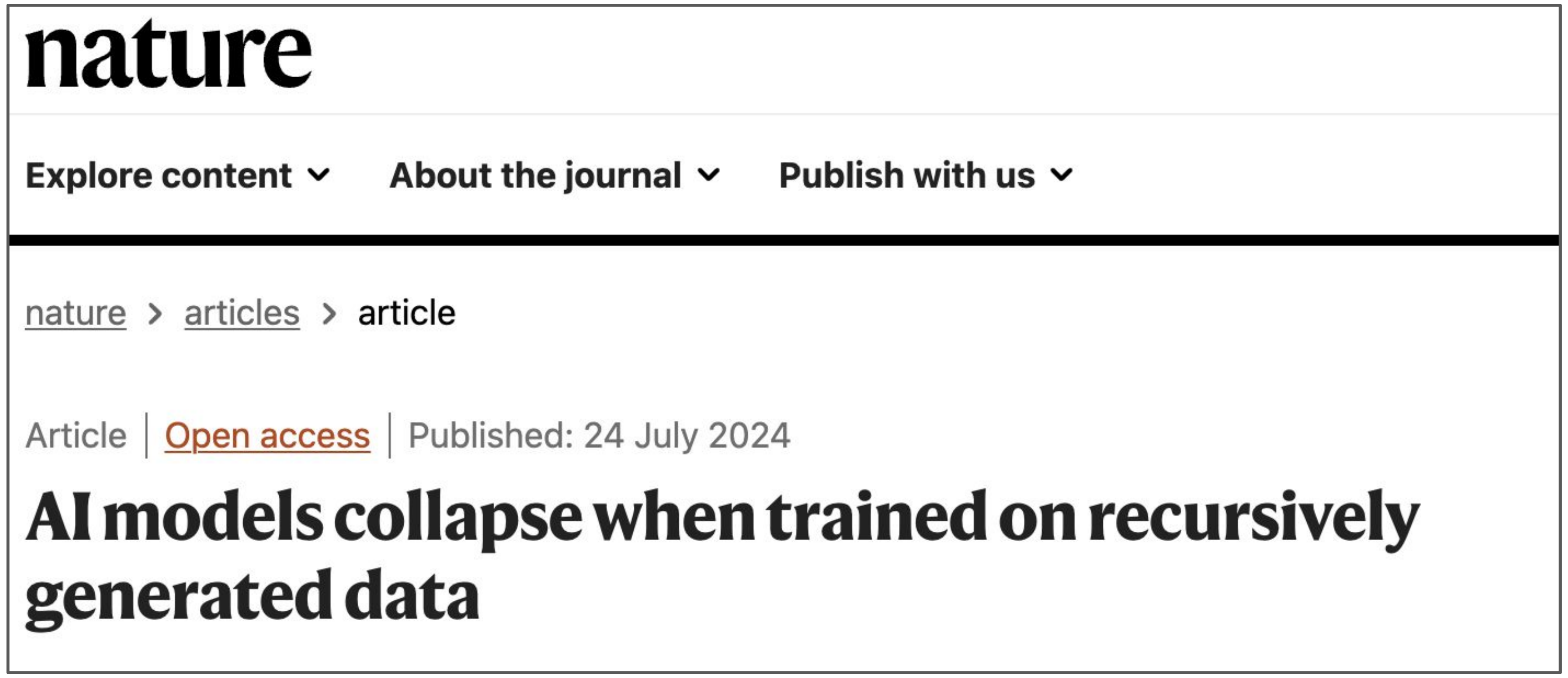
- Create more examples of the same concept (e.g. generate more 5th grade algebra problems)
- Rewrite or rephrase training examples to get more diversity (e.g. provide summaries of topics)

Mostly commonly generated by prompting existing LLMs:

Suppose you are a movie reviewer. Please generate a movie review to show your <feeling> about the movie <movie> with detailed explanation…

There are movies and then there are films. The plot…

Pure Synthetic Data: Perpetual Motion Machine?


nature

Article | Open access | Published: 24 July 2024

AI models collapse when trained on recursively generated data

In addition to other challenges – e.g. factuality / hallucination, diversity of examples, representativeness, etc.

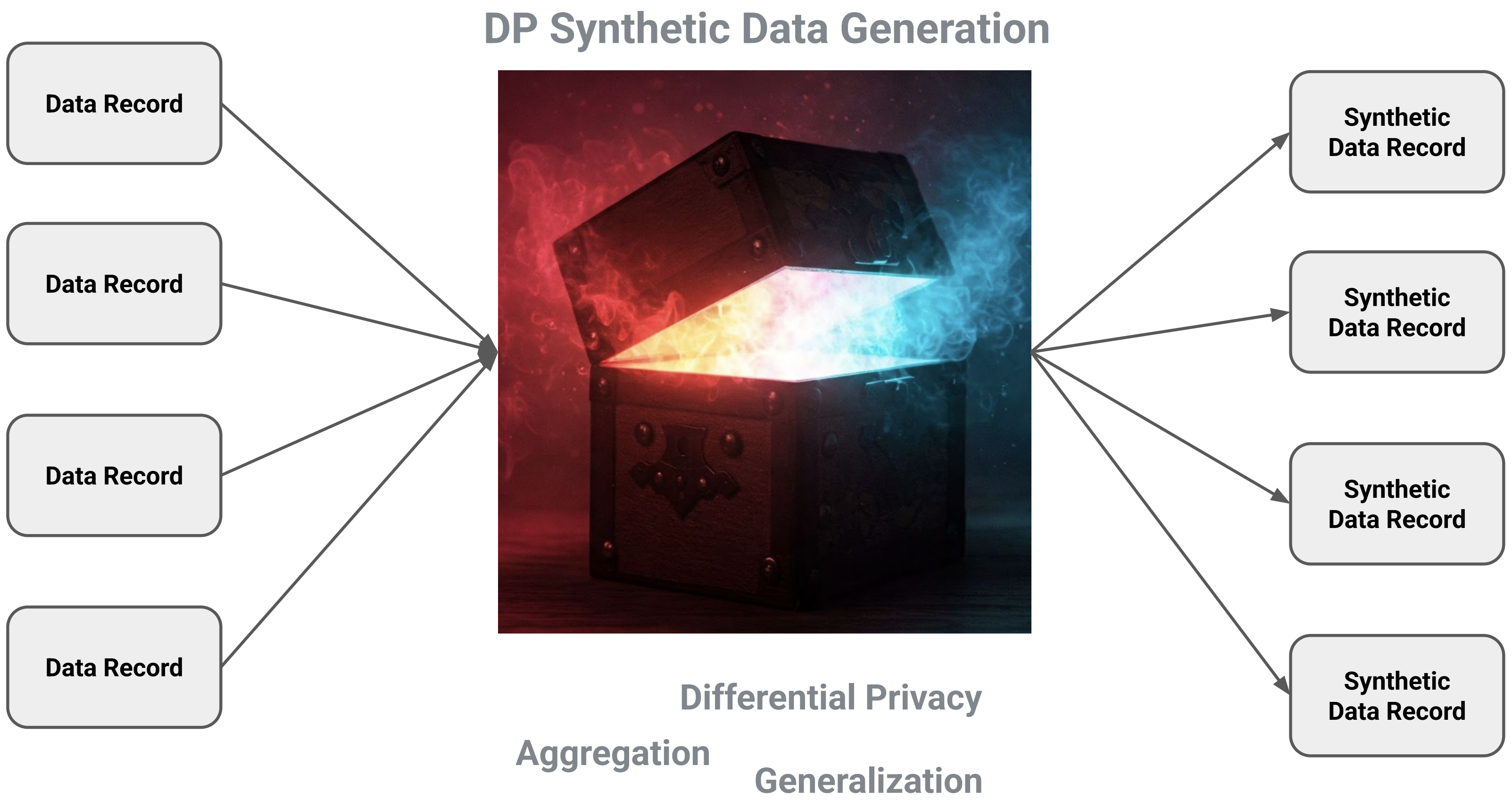
Types of Synthetic Data

Pure Synthetic Data

DP Synthetic Data

DP Synthetic Data

Many-Many Relationship between user data and synthetic data with strong privacy protections.

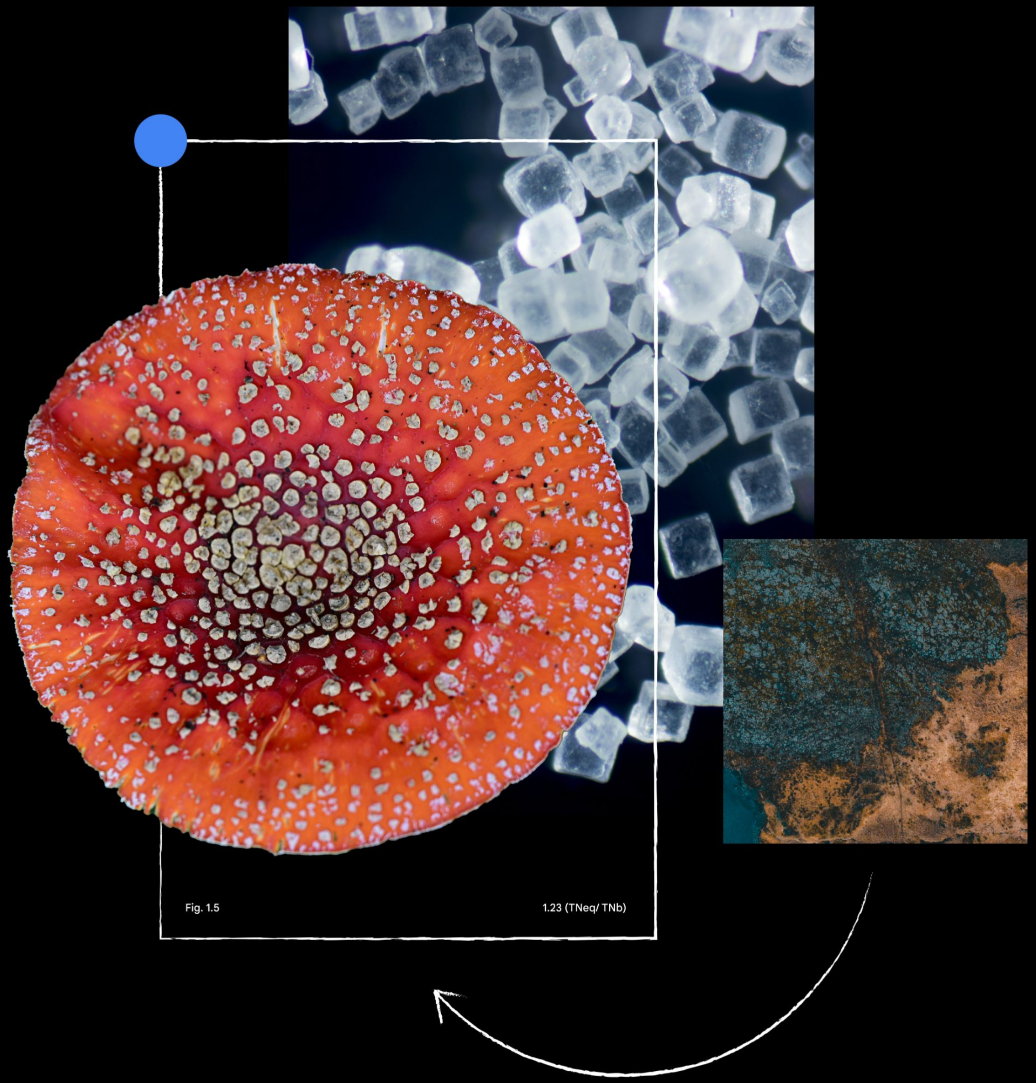


Today

- What is privacy?
- How to generate useful DP Synthetic Data - text, images, tables, …
- How to evaluate its utility?
- What else you need to do in practice?

Google Research

Differential Privacy



What is privacy?

Privacy (UK: /ˈprɪvəsiː/, US: /ˈpraɪ-/) is the ability of an individual or group to **seclude** themselves or **information about themselves**, and thereby express themselves selectively.

(From Wikipedia)

Privacy Principles in Machine Learning

- Transparency and User control
    - Users can be aware of what data is used, what purpose it is used or and how it is processed, and have full control on whether to enable the collection and use of their data
- Data minimization
    - Data is only collected focusing on specific computation needs, with access limited at all data processing stages
- Data anonymization
    - The final released output of the computation (e.g. ML model) does not reveal anything unique about an individual.
- Auditability and verifiability
    - Users, and potentially third parties can audit and verify privacy claims by examining released models, open-sourced code, and privitarized system logs, etc.

How to anonymize?

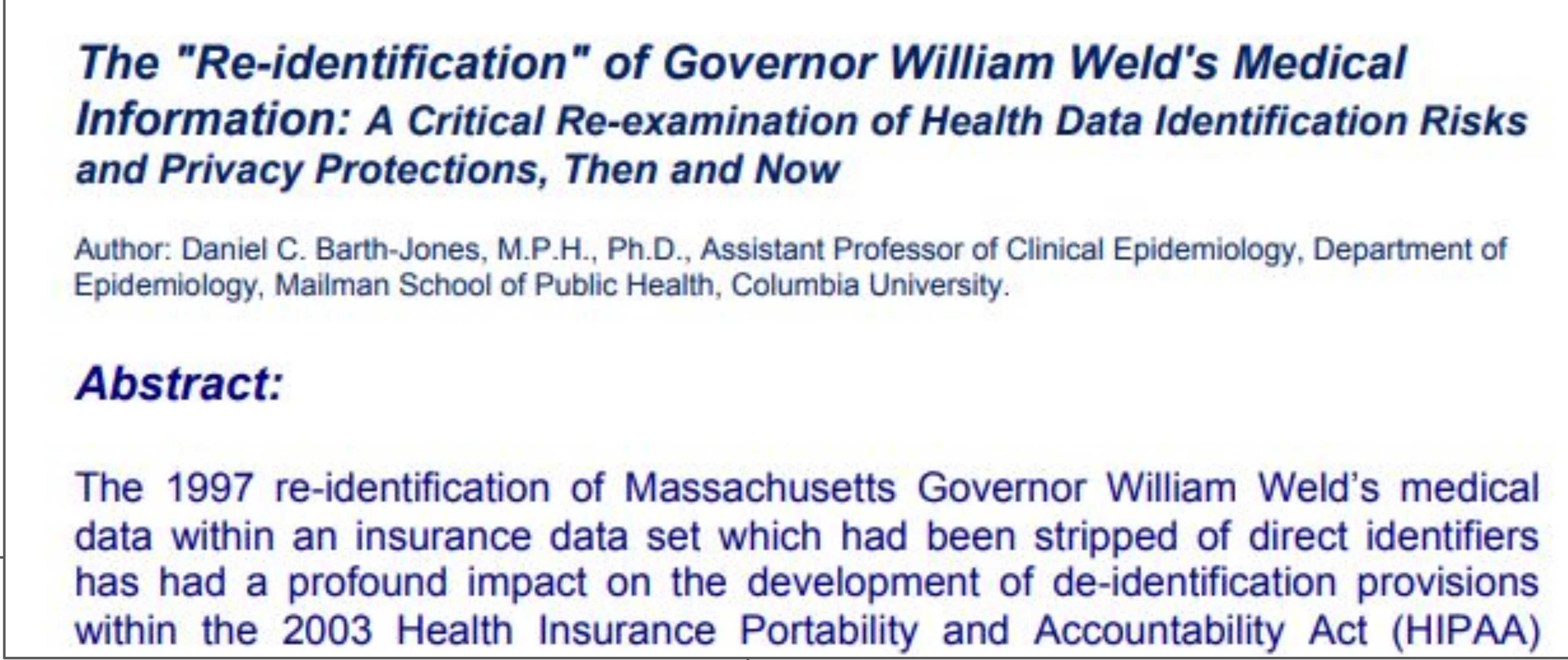
Idea 1:

- Remove user ids!

How to anonymize?

Idea 1:

- Remove user ids!

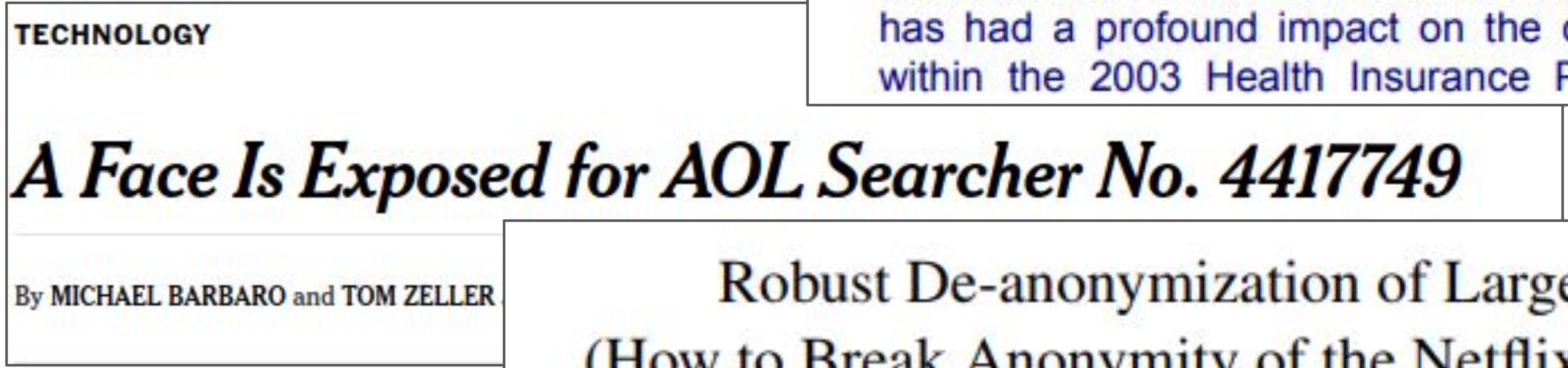


The "Re-identification" of Governor William Weld's Medical Information: A Critical Re-examination of Health Data Identification Risks and Privacy Protections, Then and Now

Author: Daniel C. Barth-Jones, M.P.H., Ph.D., Assistant Professor of Clinical Epidemiology, Department of Epidemiology, Mailman School of Public Health, Columbia University.

Abstract:

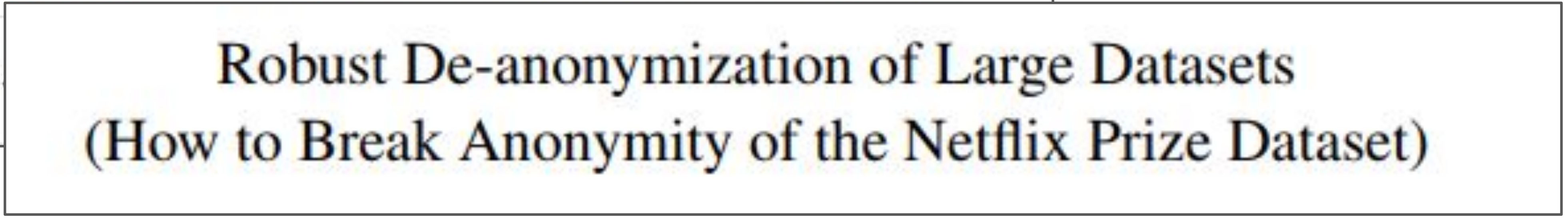
The 1997 re-identification of Massachusetts Governor William Weld's medical data within an insurance data set which had been stripped of direct identifiers has had a profound impact on the development of de-identification provisions within the 2003 Health Insurance Portability and Accountability Act (HIPAA)



TECHNOLOGY

A Face Is Exposed for AOL Searcher No. 4417749

By MICHAEL BARBARO and TOM ZELLER



Robust De-anonymization of Large Datasets
(How to Break Anonymity of the Netflix Prize Dataset)

- 5 digit zipcode + DOB + gender uniquely identifies 87% of US population.

Google

How to anonymize?

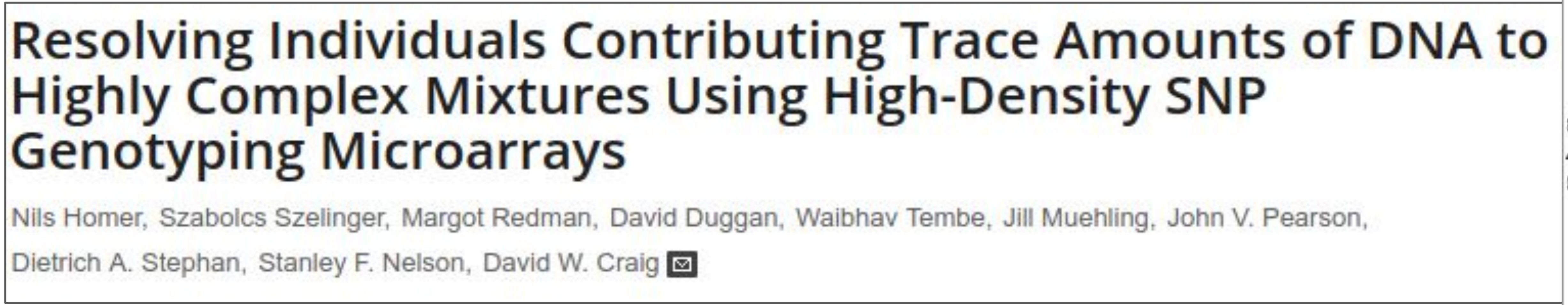
Idea 2:

- Only allow access to aggregates

How to anonymize?

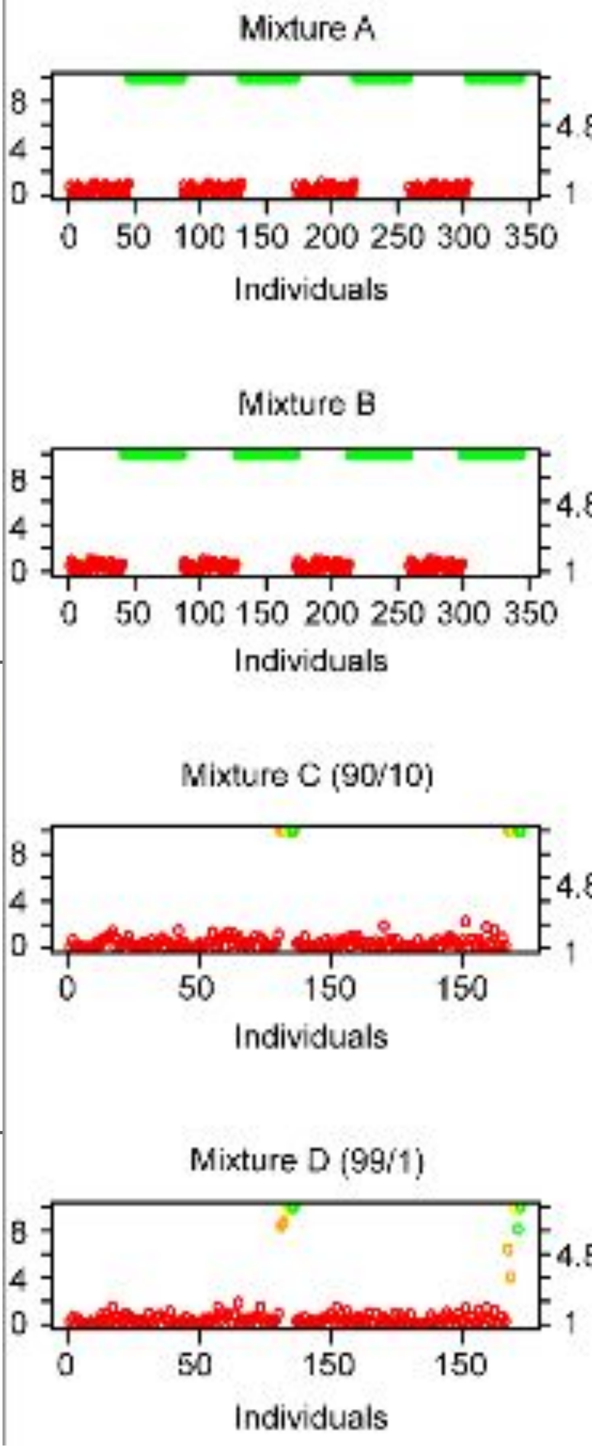
Idea 2:

- Only allow access to aggregates



Resolving Individuals Contributing Trace Amounts of DNA to Highly Complex Mixtures Using High-Density SNP Genotyping Microarrays

Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Stanley F. Nelson, David W. Craig



- Determined if a particular participant is part of the dataset

How to anonymize?

Idea 3:

- Make sure at least **k** different people share the attributes
- “k-anonymity”

How to anonymize?

Idea 3:

- Make sure at least **k** different people share the attributes
- "k-anonymity"

Attacks:

- [Homogeneity]: All k people share a sensitive attribute.
    - "All k have declared bankruptcy in the past"

How to anonymize?

Idea 3 (b):

- Make sure at least **k** different people share the attributes
- And sensitive attributes are diverse in each equivalence class
- “k-anonymity with l-diversity”

How to anonymize?

Idea 3 (b):

- Make sure at least **k** different people share the attributes
- And sensitive attributes are diverse in each equivalence class
- “k-anonymity with l-diversity”

Attack:

- Consider a class that has people who are either bankrupt or convicted of crime.

How to anonymize?

Idea 3 (c):

- Make sure at least **k** different people share the attributes
- And sensitive attributes are diverse in each equivalence class
- And distribution of sensitive attribute is similar to global
- “k-anonymity with l-diversity and t-closeness”

How to anonymize?

Idea 3 (c):

- Make sure at least **k** different people share the attributes
- And sensitive attributes are diverse in each equivalence class
- And distribution of sensitive attribute is similar to global.
- “k-anonymity with l-diversity and t-closeness”

Attack:

- Still have to define sensitive attributes. Does not prevent information leakage when attacker knows additional information about the subject.

What do we want?

The output of an algorithm should be the same whether or not a specific individual participated.

What do we want?

The output of an algorithm should be the same whether or not a specific individual participated.

Too harsh: by induction no learning can happen

What do we want?

The probability of an output of an algorithm should be almost the same whether or not a specific individual participated.

Example

Want to return the average age of people in this room

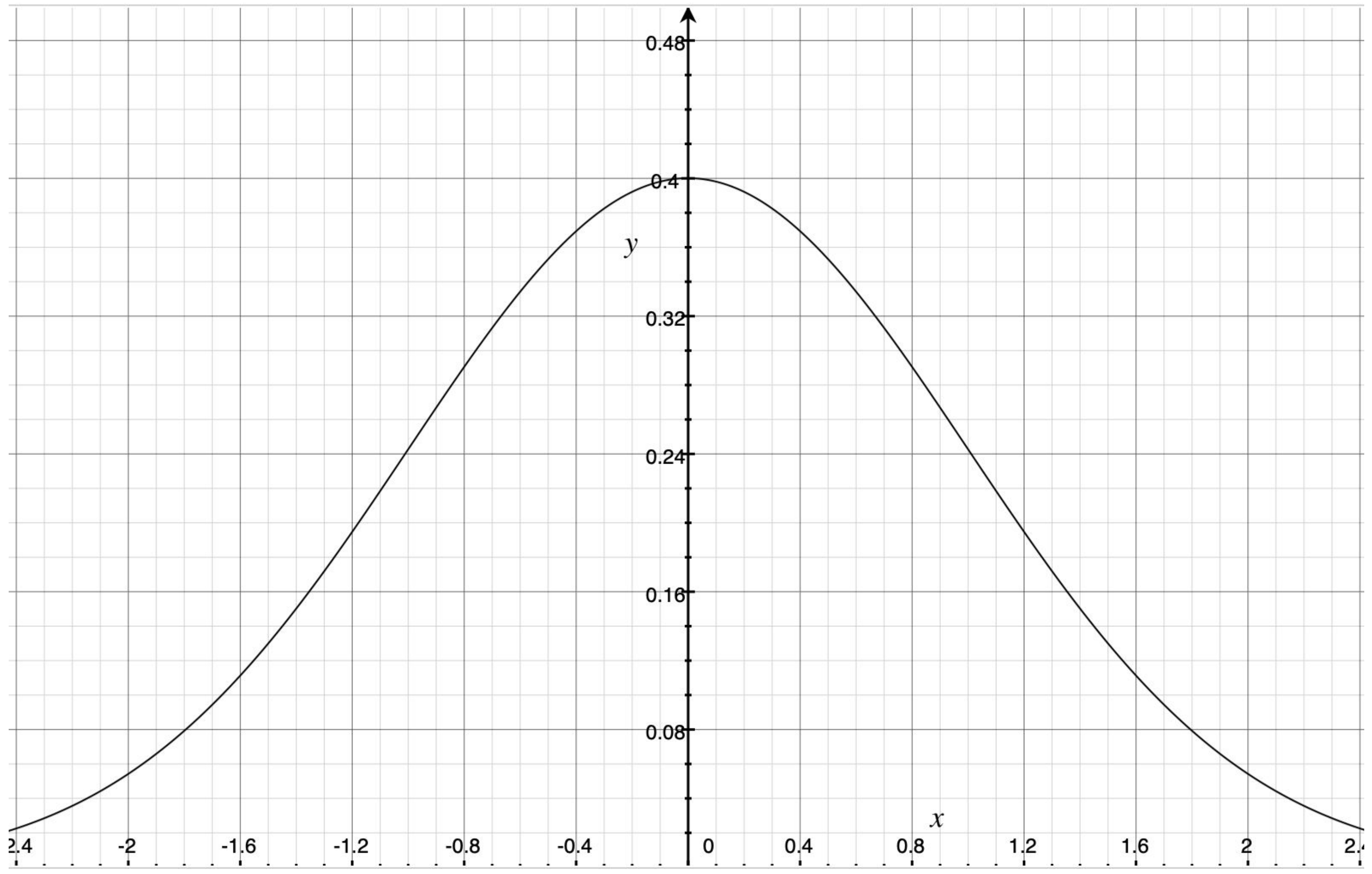
- Without revealing my age

System returns:

- True answer + Gaussian noise with mean 0, std 1

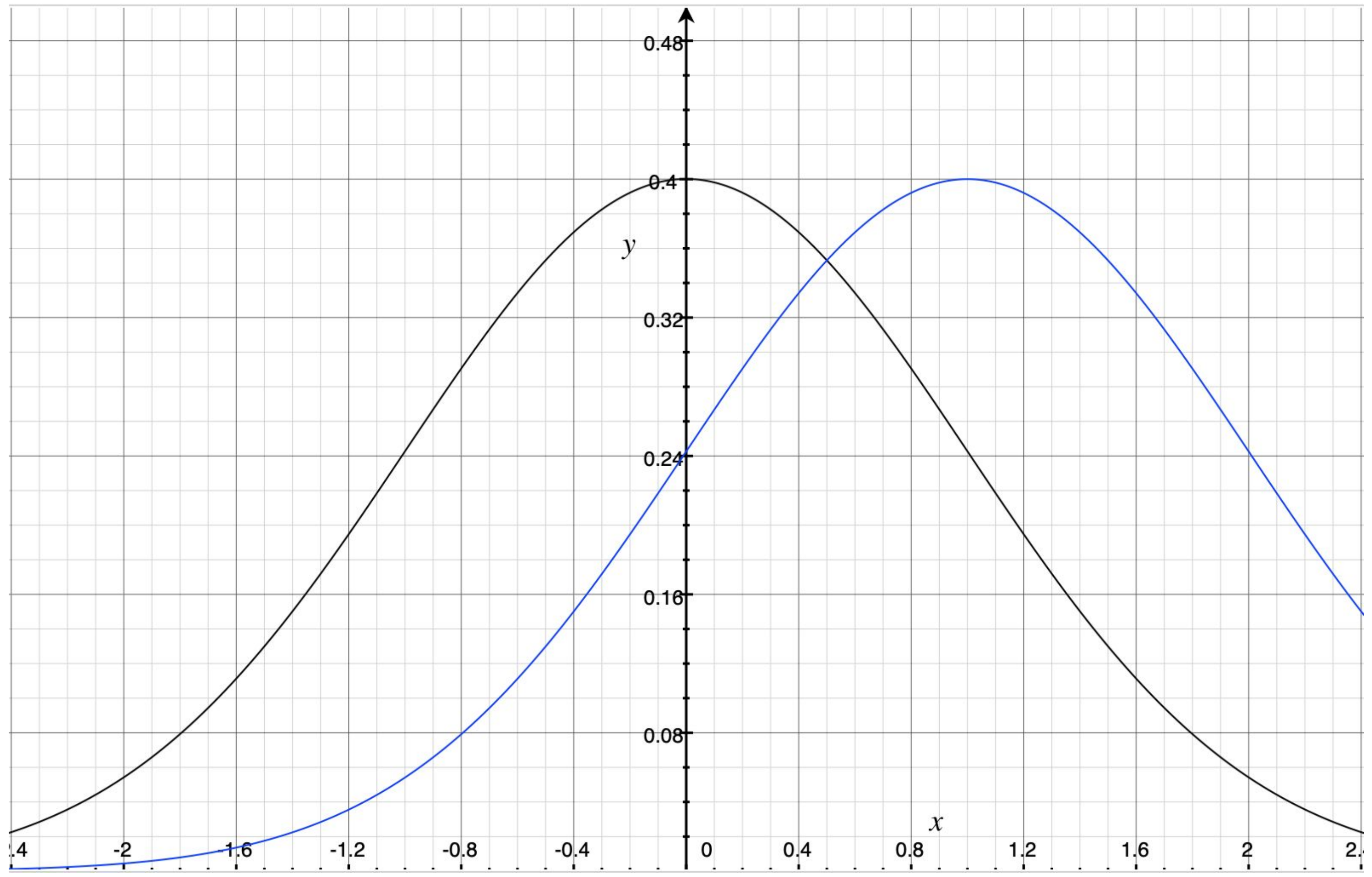
Noisy Average Age

Result Distribution



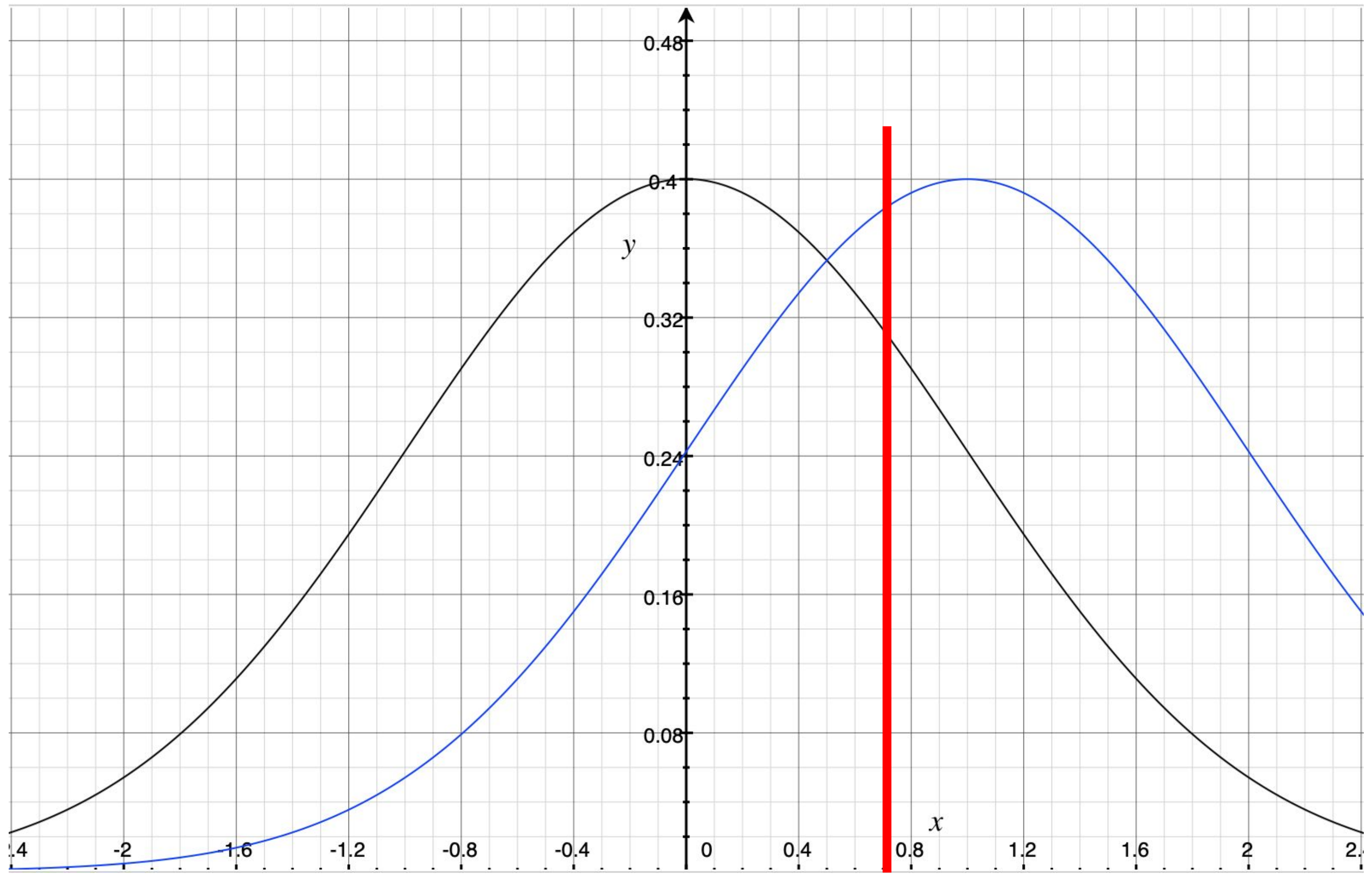
Noisy Average Age

Result Distribution, when average differs by one



Noisy Average Age

Given actual result (red): can’t say if it came from blue or black curves



More likely that it came from blue curve, but not much more likely.

Example

Want to return the average age of people in this room

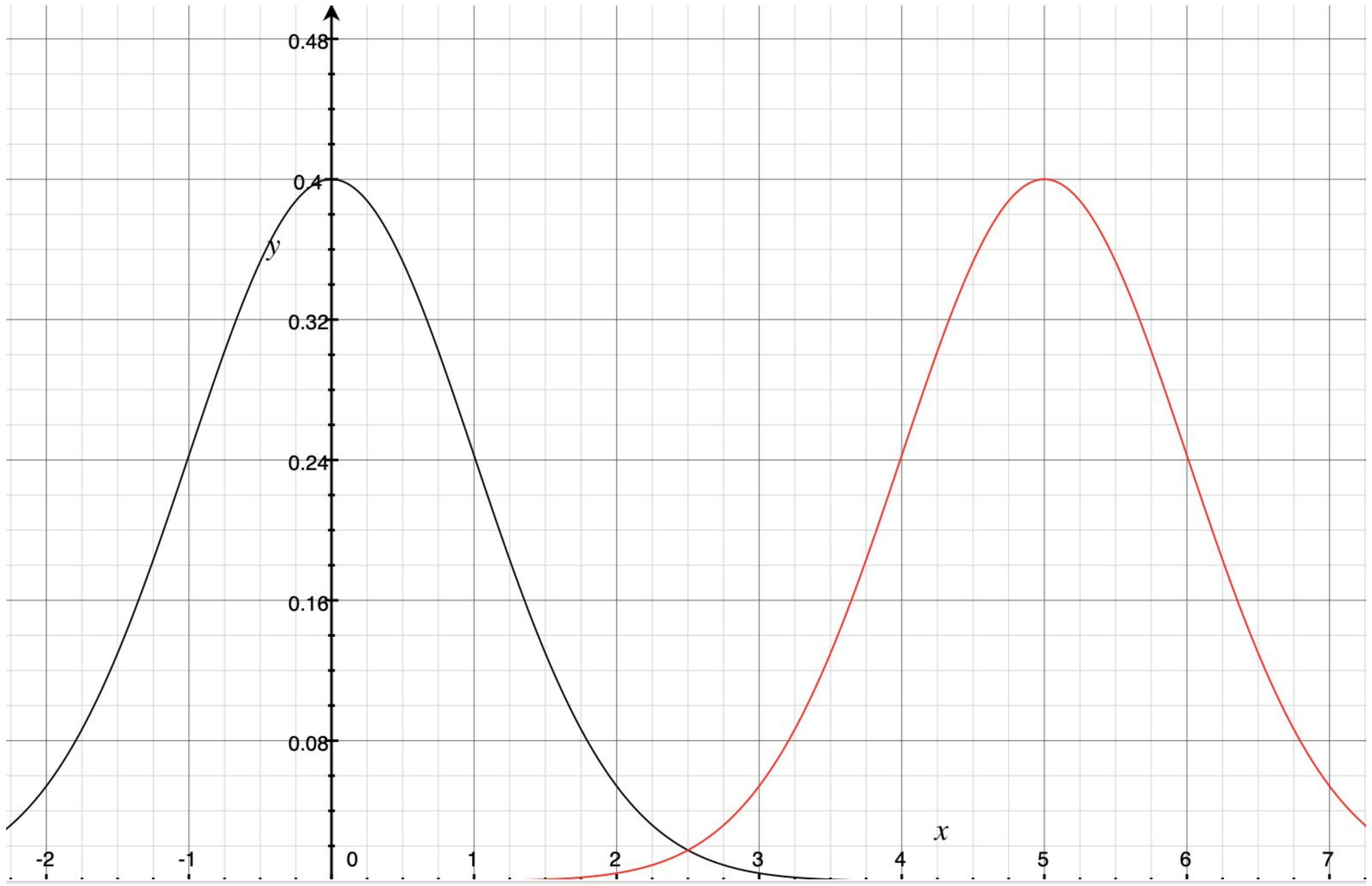
- Without revealing my age

System returns:

- True answer + Gaussian noise with mean 0, std 1

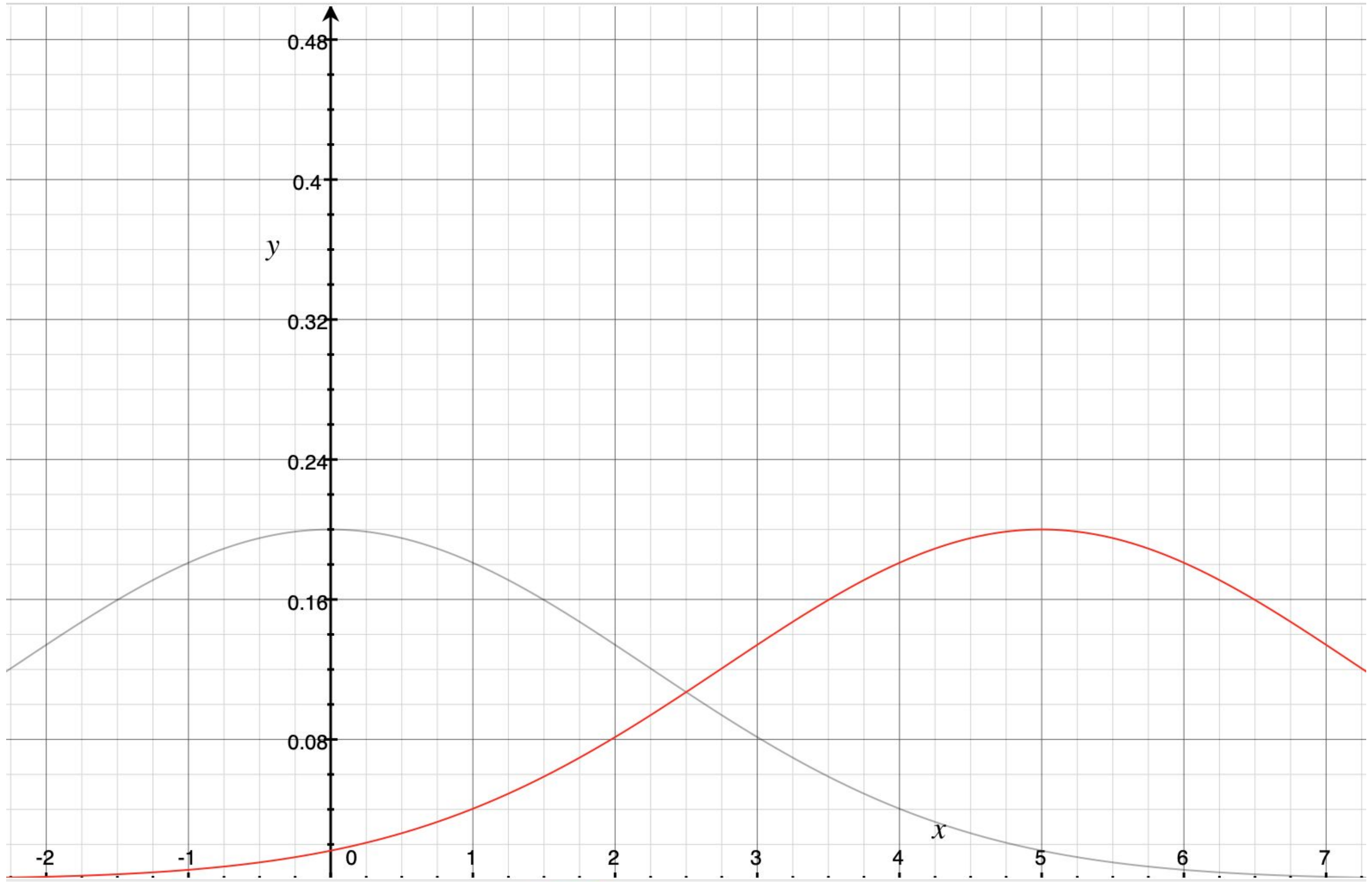
Even if attacker knows the age of everyone else, can’t definitively say what my age is, just give a distribution.

Some more details



If a single individual makes a big difference in the answer, the curves are well separated, so result is less private.

Some more details



If a single individual makes a big difference in the answer, the curves are well separated, must add more noise

Differential Privacy

The probability of an output of an algorithm should be almost the same whether or not a specific individual participated.

Has a parameter “epsilon” which controls privacy vs. utility trade-off. Loosely speaking:

- Effect of single person changing the answer / std. deviation of noise
- Larger epsilon = less private.
- Typical values (0.1-0.2 in academia, 1-10 in industry)

Differential Privacy Summary

Limit outliers and contributions from any user:

- Outliers in value or in number of data points contributed

Aggregate information across many users

- Aggregation is a key anonymization component

Add some noise to the aggregates

- Need to add noise to protect against differencing attacks

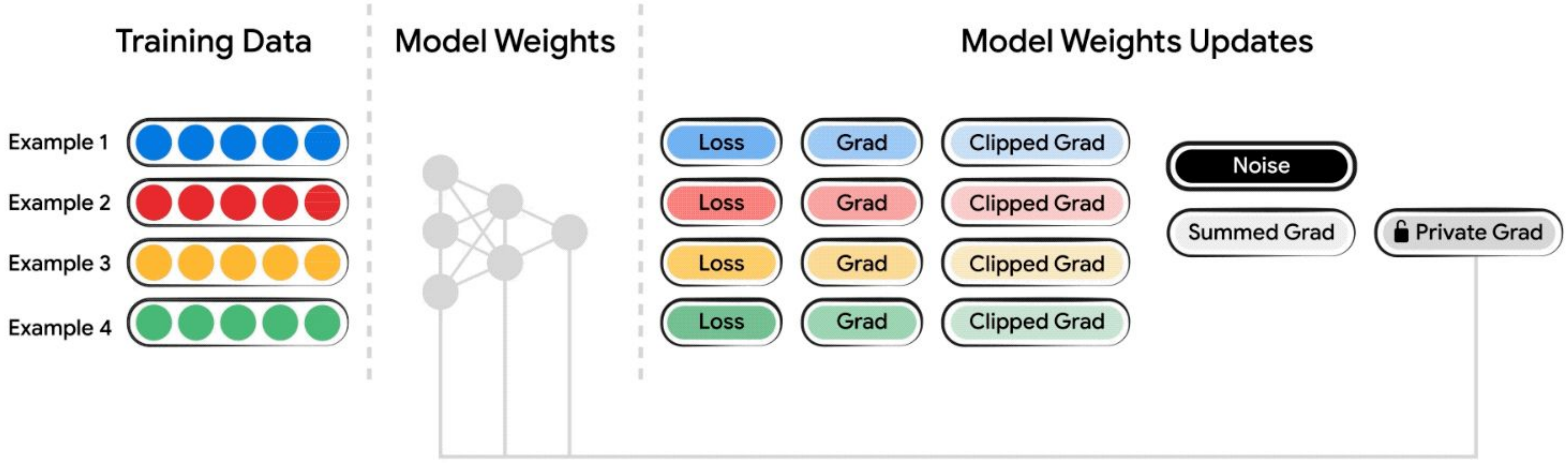
Example: Gradient Descent with Differential Privacy

- Sample examples to form a batch
- “Clip” or bound the individual gradients
- Average across the batch
- Add noise the average

- **Sensitivity** analysis to decide an appropriate amount of noise for DP guarantees
  - Can be calculated analytically for some functions like strongly convex linear ones
  - General solution is just to clip per example gradient to some predefined norm C

More in How to DP-fy ML Sec. 4.1.2

Example: DP Training: DP-SGD



More in How to DP-fy ML Sec. 4.2

Key Differential Privacy Properties

Post-Processing & Composition:

- Any user-data independent transformation (post-processing) with arbitrary additional information retains all of the differential privacy properties

Composition:

- When combining several DP methods together, combine the DP parameters.

Theoretical Guarantees:

- The mathematical guarantees provided are on the algorithm used to generate the anonymous data. Use robust, well-established libraries (like crypto).

Key Parameters when Designing Differential Privacy

Privacy Unit: What is a user?

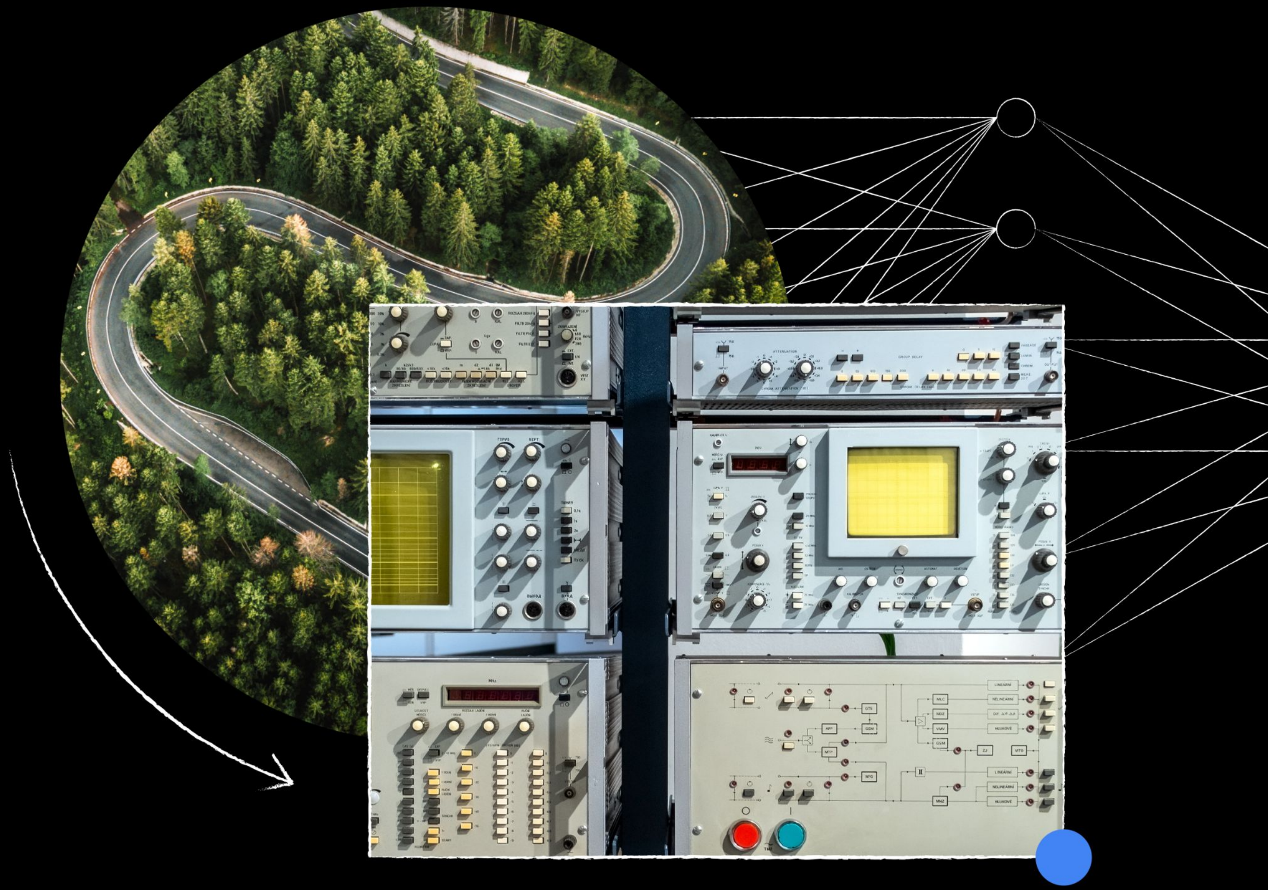
Contribution Bounding: How to limit outliers?

Methods: Which private method are you using?

Parameters: What is the value of epsilon, delta?

Google Research

DP synthetic text



Google Research

Overview

1. Taxonomy of methods
2. Methods
   a. DP finetuning
   b. Private evolution
   c. DP inference
3. Comparison

For each method: Will cover (1) how the algorithm works; (2) how and when to use it

Google Research

Taxonomy of methods for DP text synthesis

Goal: Unlock value of private text data with DP

Google Research

Taxonomy of methods for DP text synthesis

Goal: Unlock value of private text data with DP

DP-fy downstream training

DP Synthetic data

DP Synthetic data = organizational breakthrough for DP

1. Bespoke DPfying someone’s training pipeline is hard
2. Everyone knows what to do with data

Taxonomy of methods for DP text synthesis

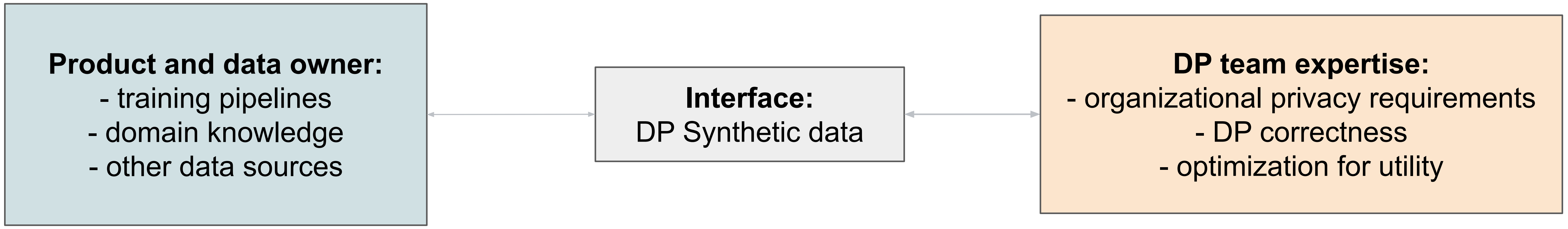
Goal: Unlock value of private text data with DP

DP-fy downstream training

DP Synthetic data

DP Synthetic data = organizational breakthrough for DP

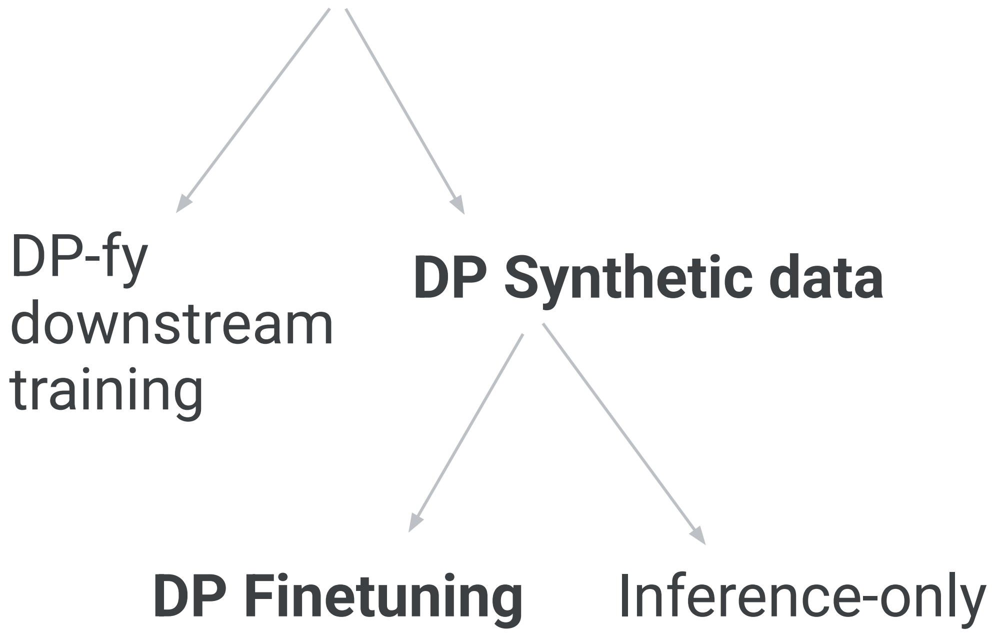
1. Bespoke DPfying someone's training pipeline is hard
2. Everyone knows what to do with data



Google Research

Taxonomy of methods for DP text synthesis

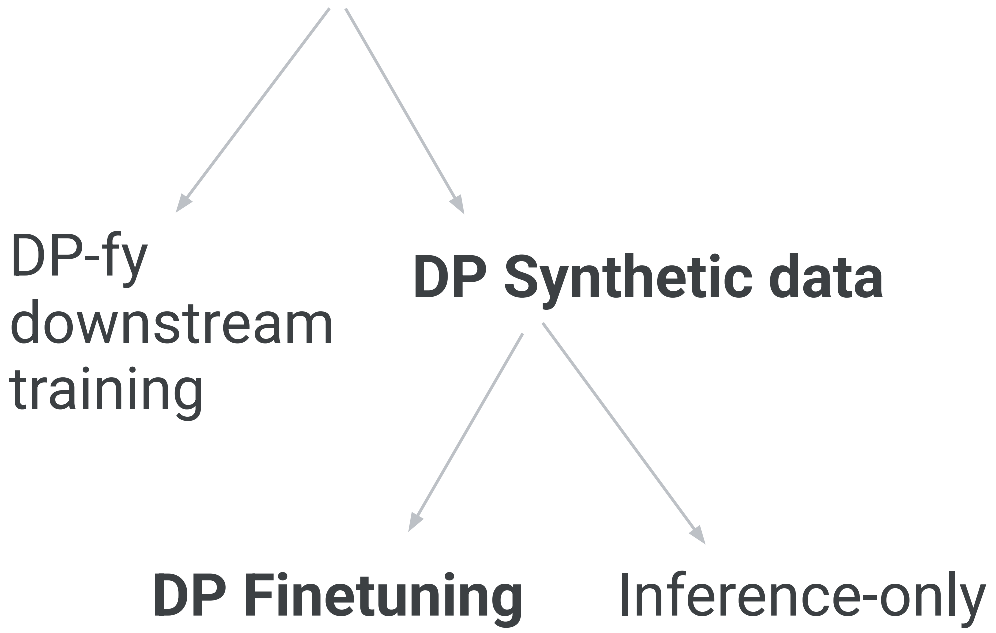
Goal: Unlock value of private text data with DP



Google Research

Taxonomy of methods for DP text synthesis

Goal: Unlock value of private text data with DP



DP Finetuning

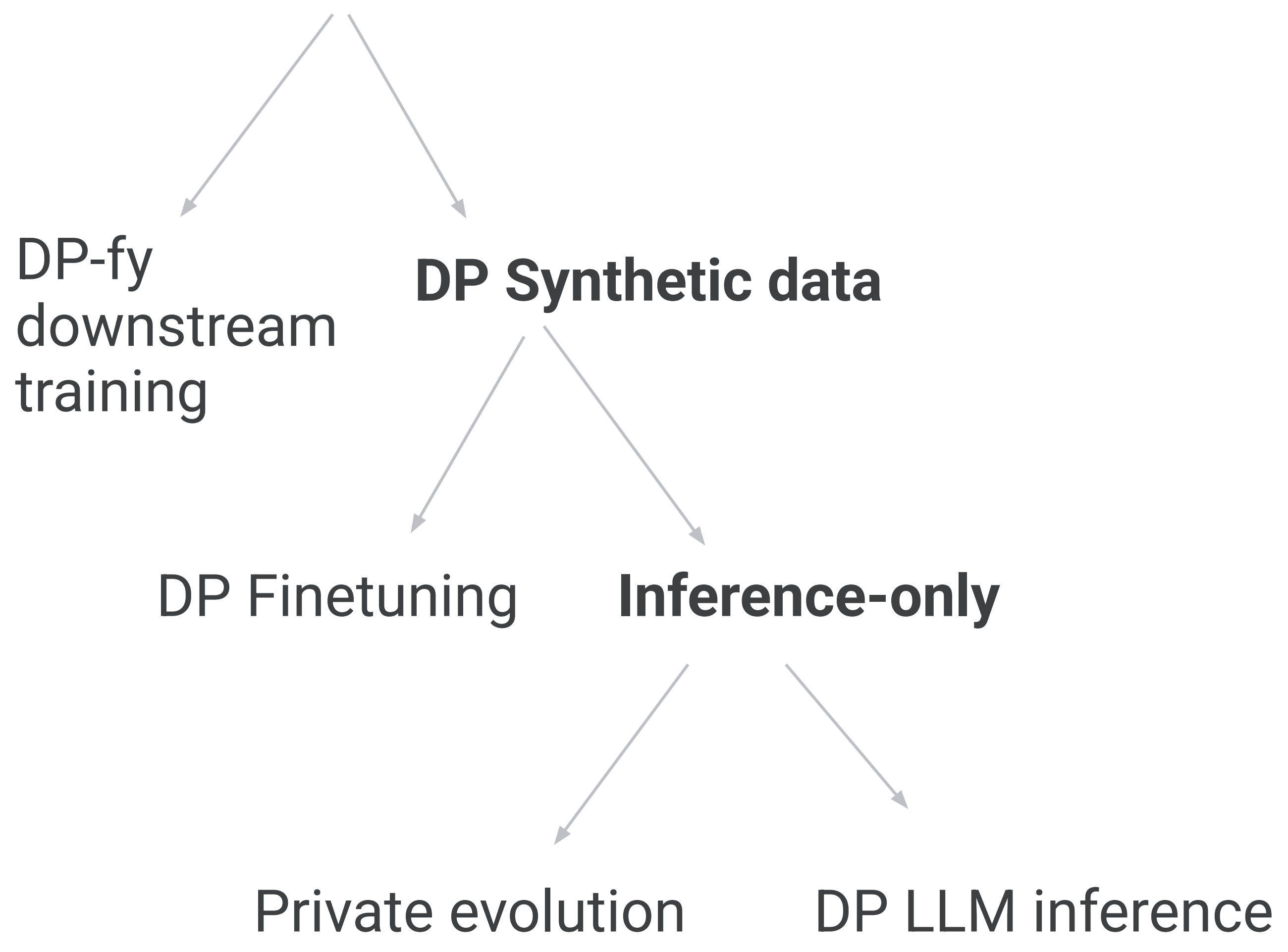
- Finetune an LLM with DP-SGD to generate data

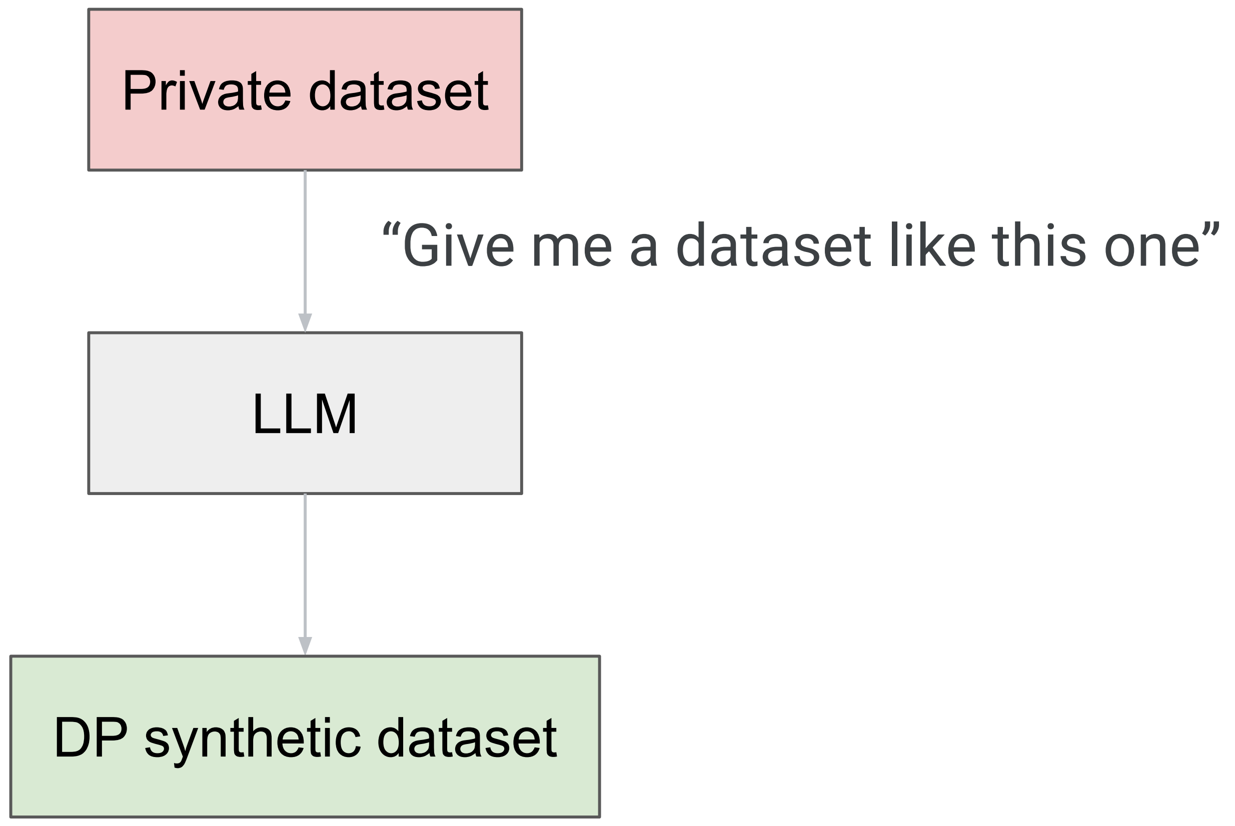
```
{
   inputs: "Generate data",
   targets: "{data_record_i}"
}
```

Workhorse method that is the best for most cases

Taxonomy of methods for DP text synthesis

Goal: Unlock value of private text data with DP

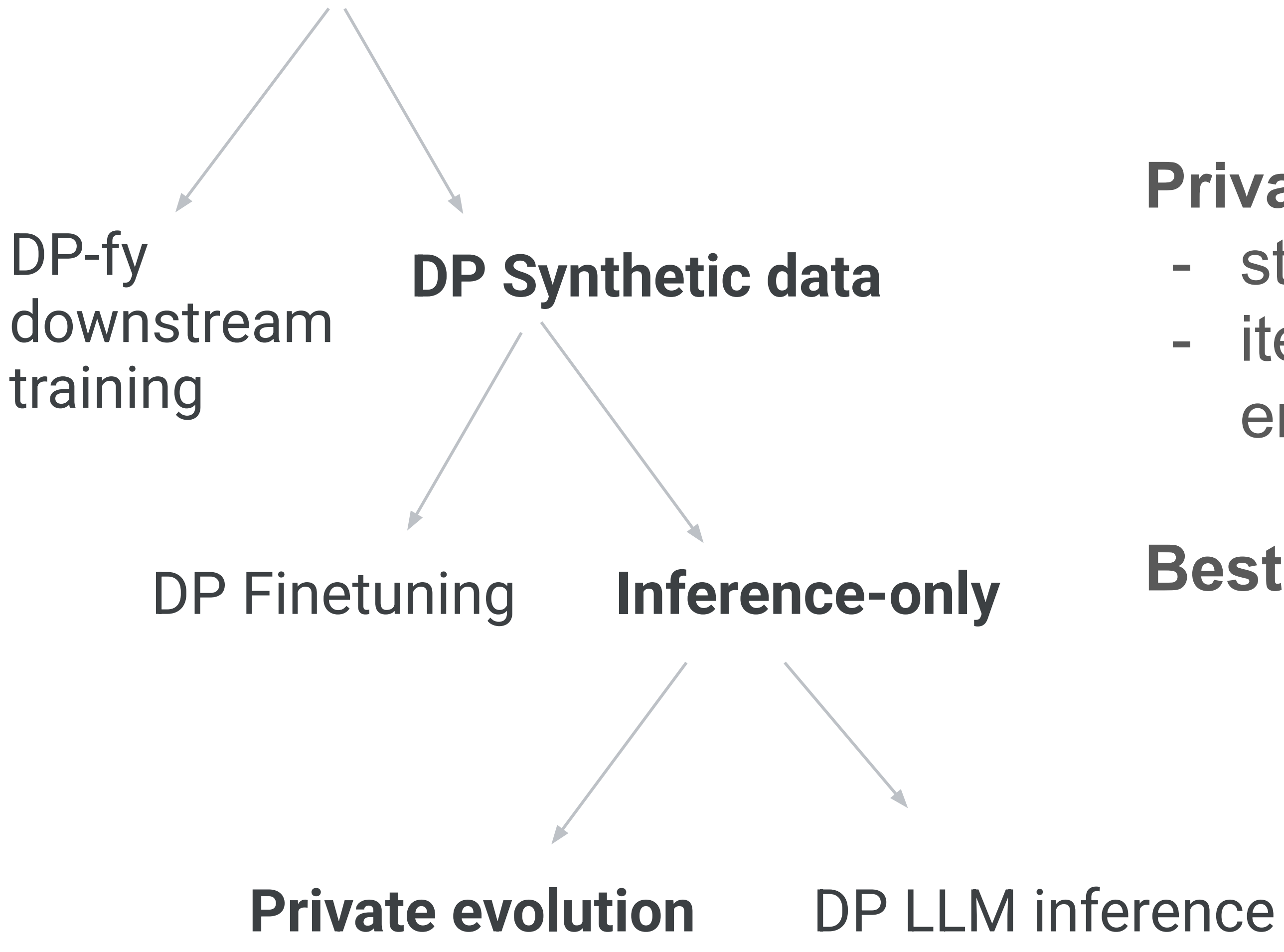




Google Research

Taxonomy of methods for DP text synthesis

Goal: Unlock value of private text data with DP



Private evolution

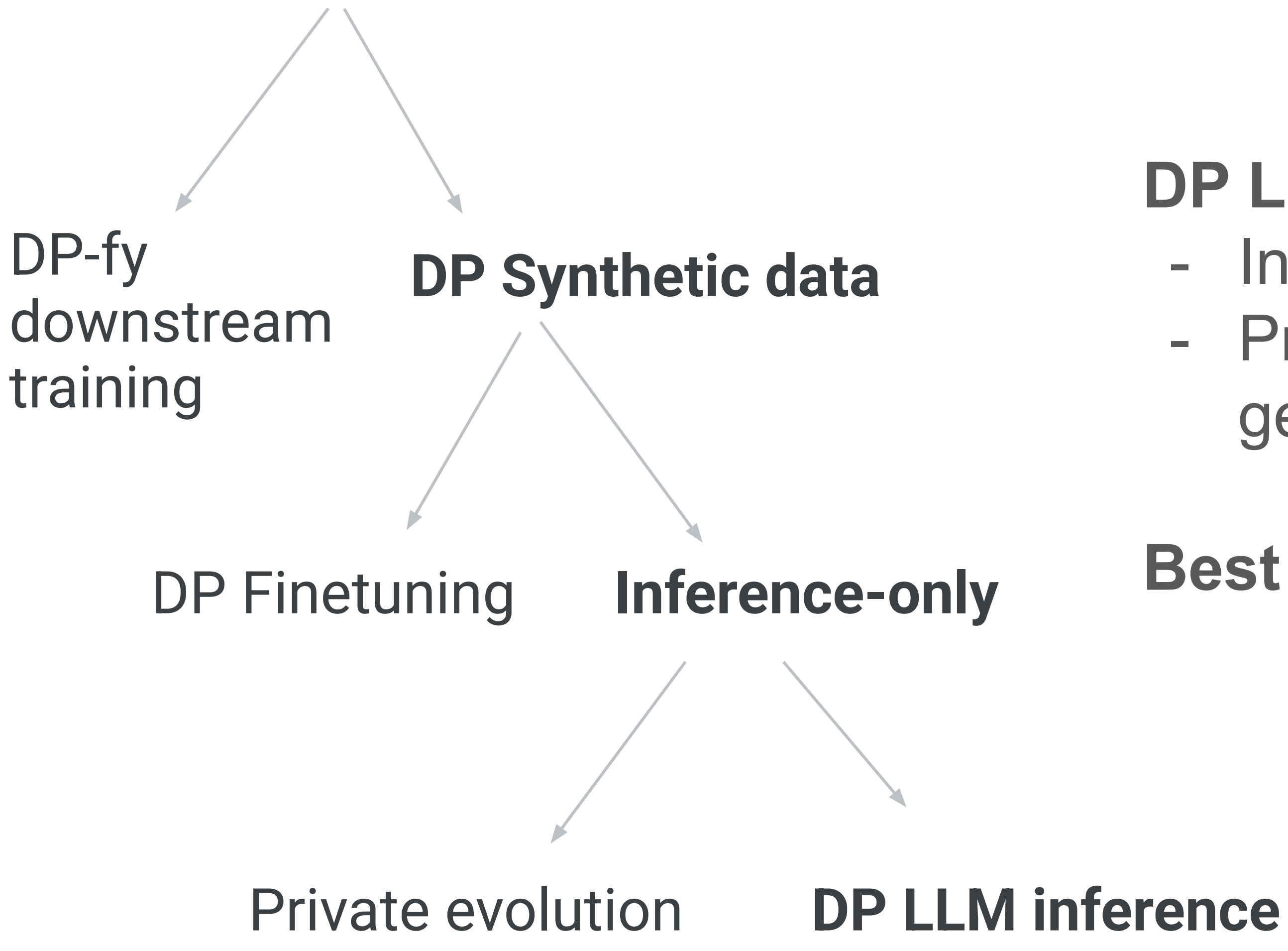
- starting from a synthetic seed corpus
- iteratively select and modify examples guided by embedding distance to private data

Best results for low epsilon/low data volume

Google Research

Taxonomy of methods for DP text synthesis

Goal: Unlock value of private text data with DP



DP LLM inference

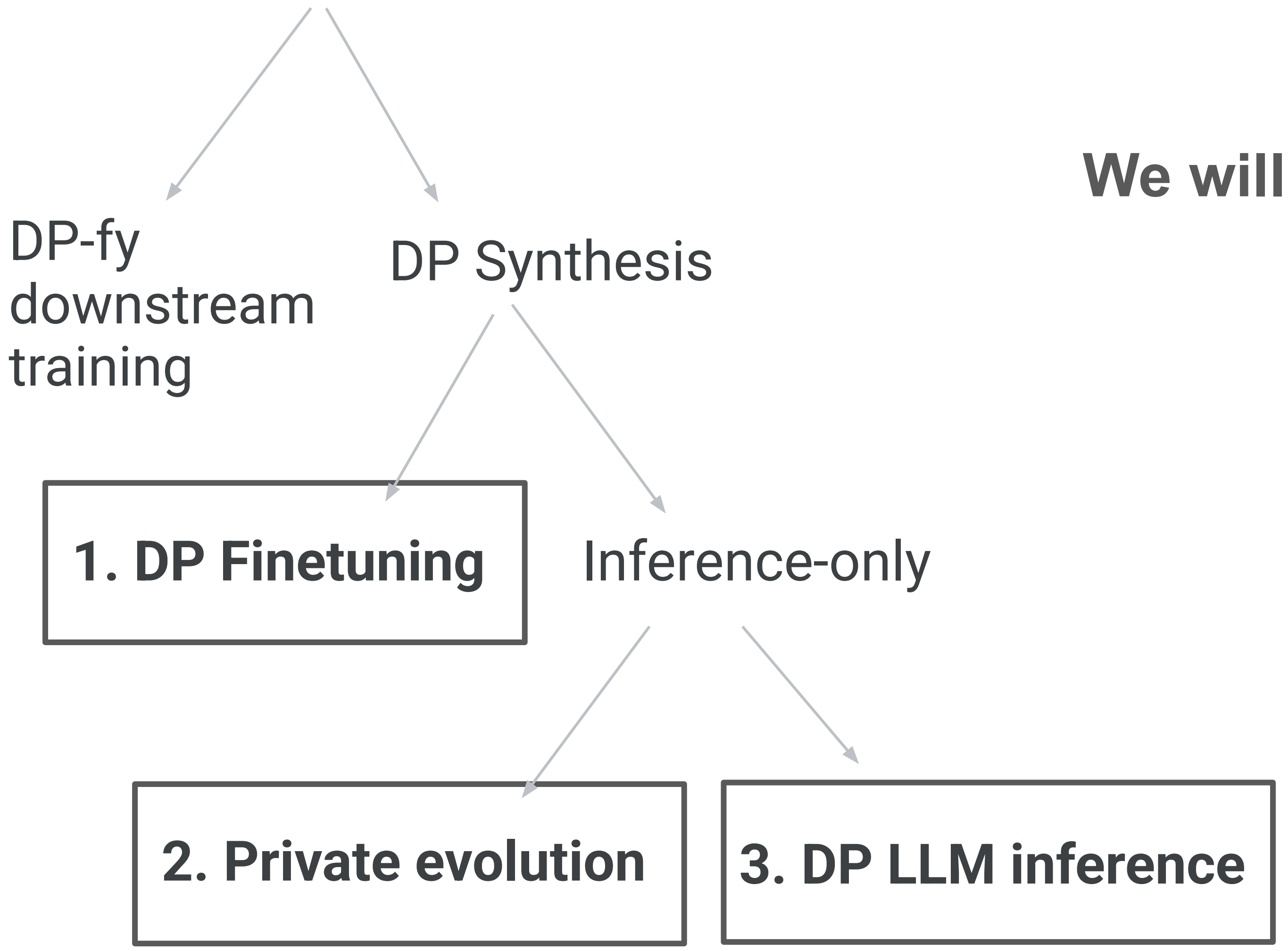
- Introduce DP during language model decoding step
- Privacy cost increases with the number of tokens you generate

Best for getting a few examples, very quickly

Google Research

Taxonomy of methods for DP text synthesis

Goal: Unlock value of private text data with DP



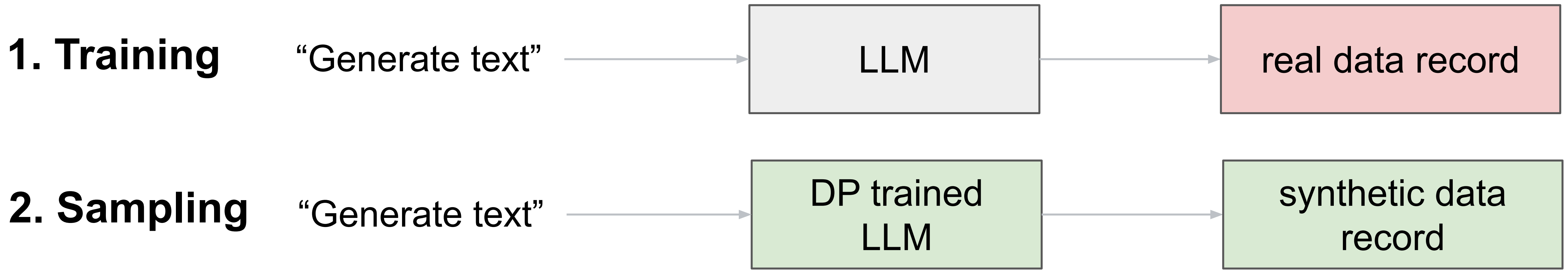
We will cover these 3 methods today.

Google Research

DP finetuning

Use DP-SGD to finetune a pre-trained LLM into a data generator.

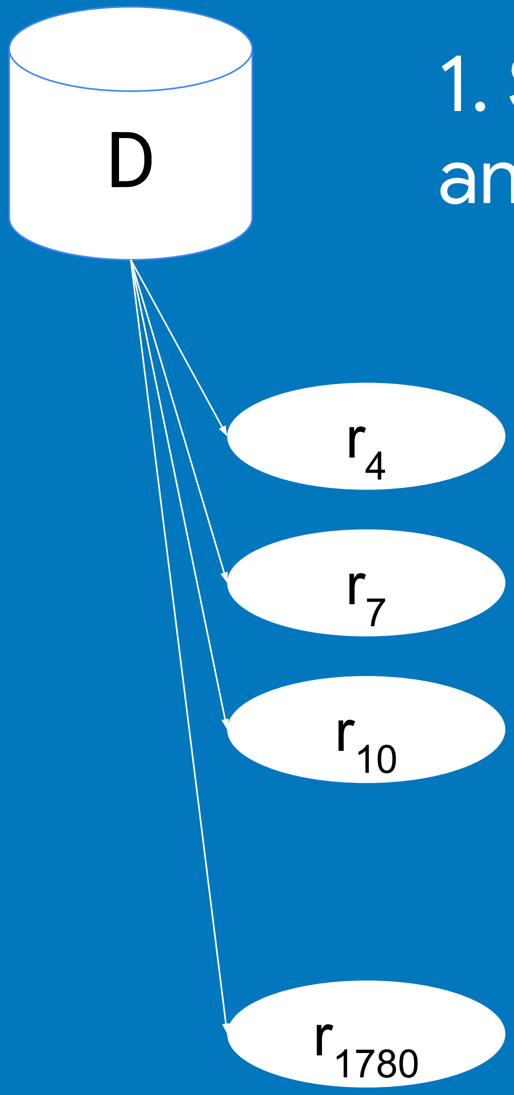
The most straightforward and performant method.



Use **LoRA** on **all layers**

- LoRA outperforms full finetuning empirically
- Also efficiency gains with DP

Iterative training with differential privacy

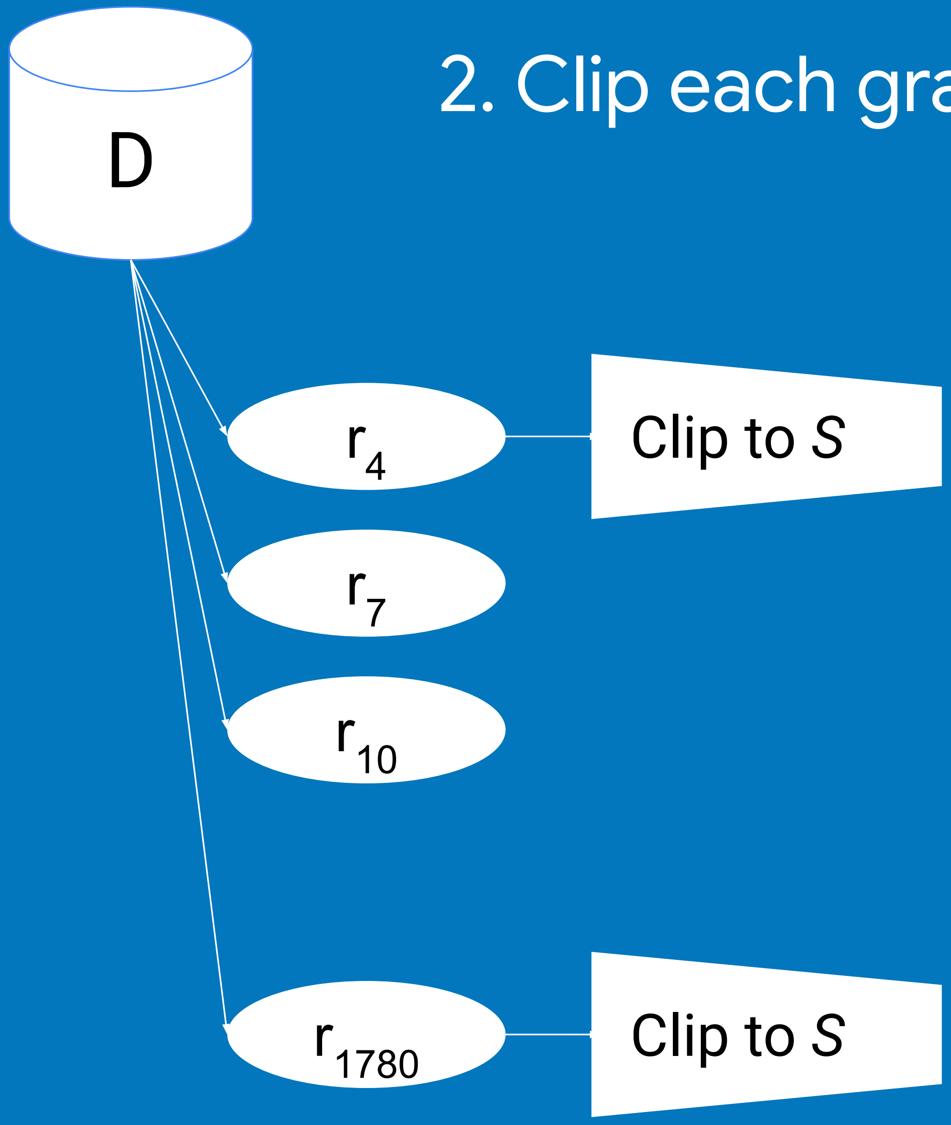


1. **Sample** a batch of examples and **compute gradients** for the current model

M. Abadi, et. al. **Deep Learning with Differential Privacy.** CCS 2016.

Iterative training with differential privacy

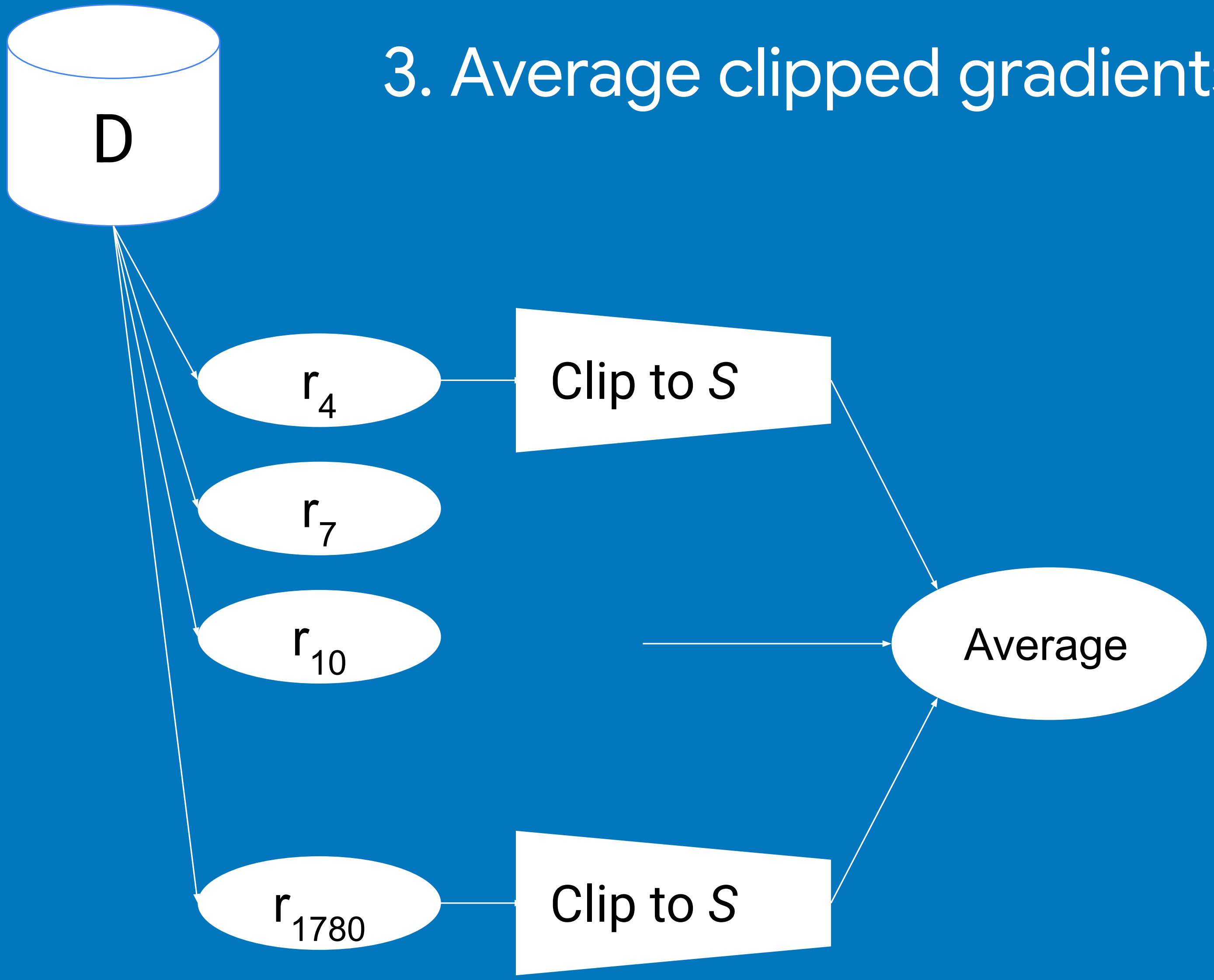
2. Clip each gradient to maximum L_2 norm S



M. Abadi, et. al. **Deep Learning with Differential Privacy.** CCS 2016.

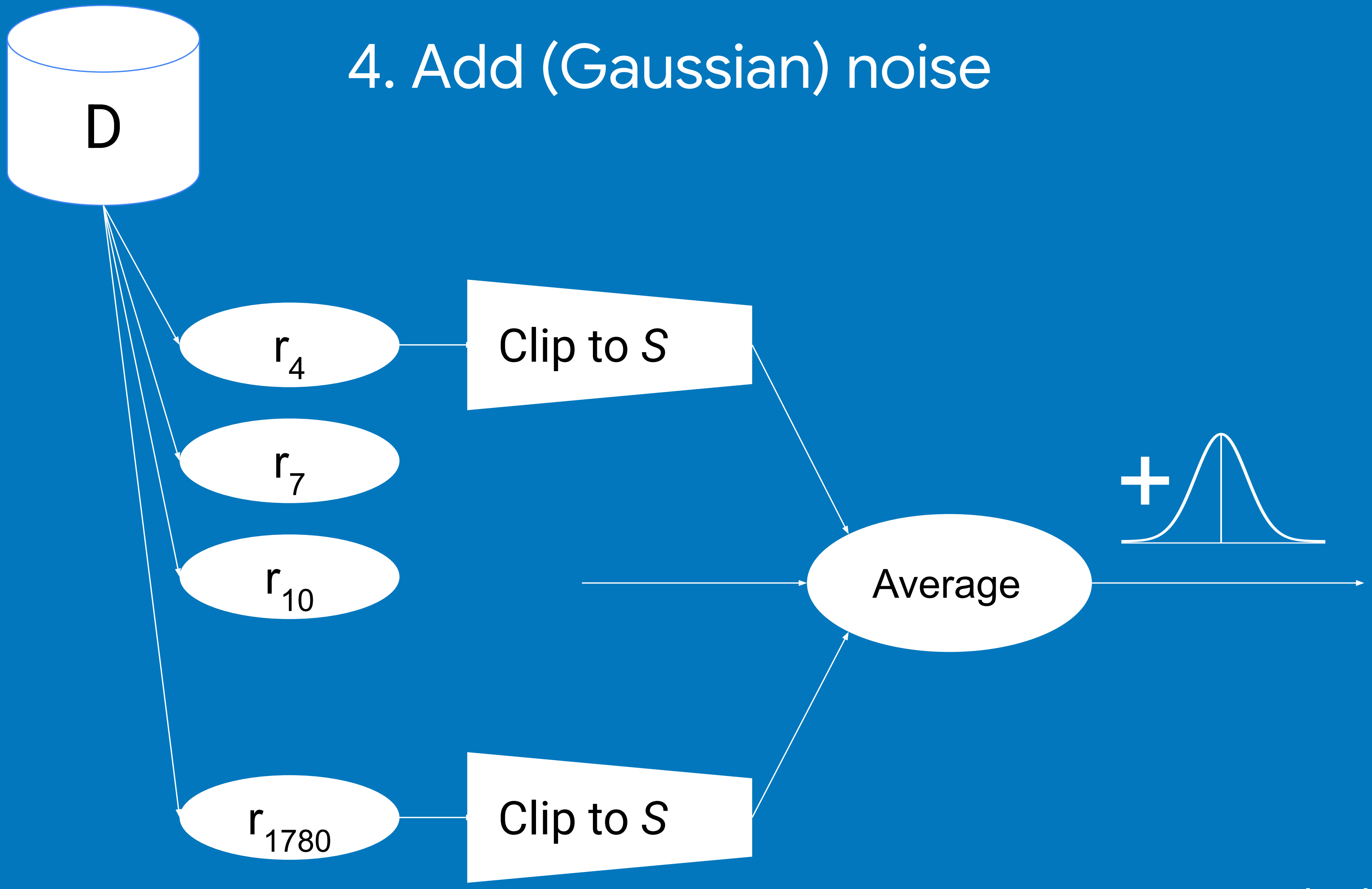
Iterative training with differential privacy

3. Average clipped gradients



M. Abadi, et. al. **Deep Learning with Differential Privacy.** CCS 2016.

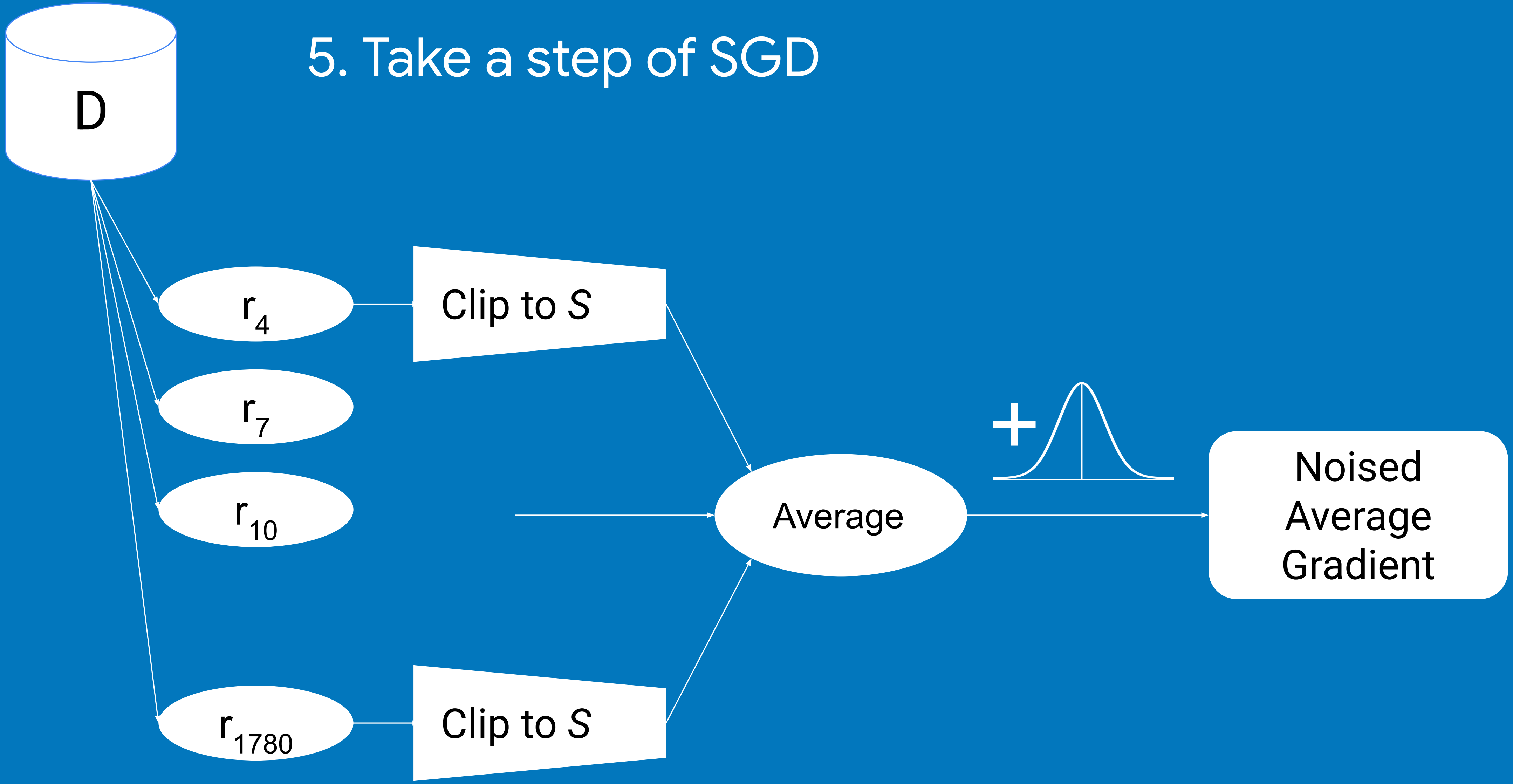
Iterative training with differential privacy



M. Abadi, et. al. **Deep Learning with Differential Privacy.** CCS 2016.

Iterative training with differential privacy

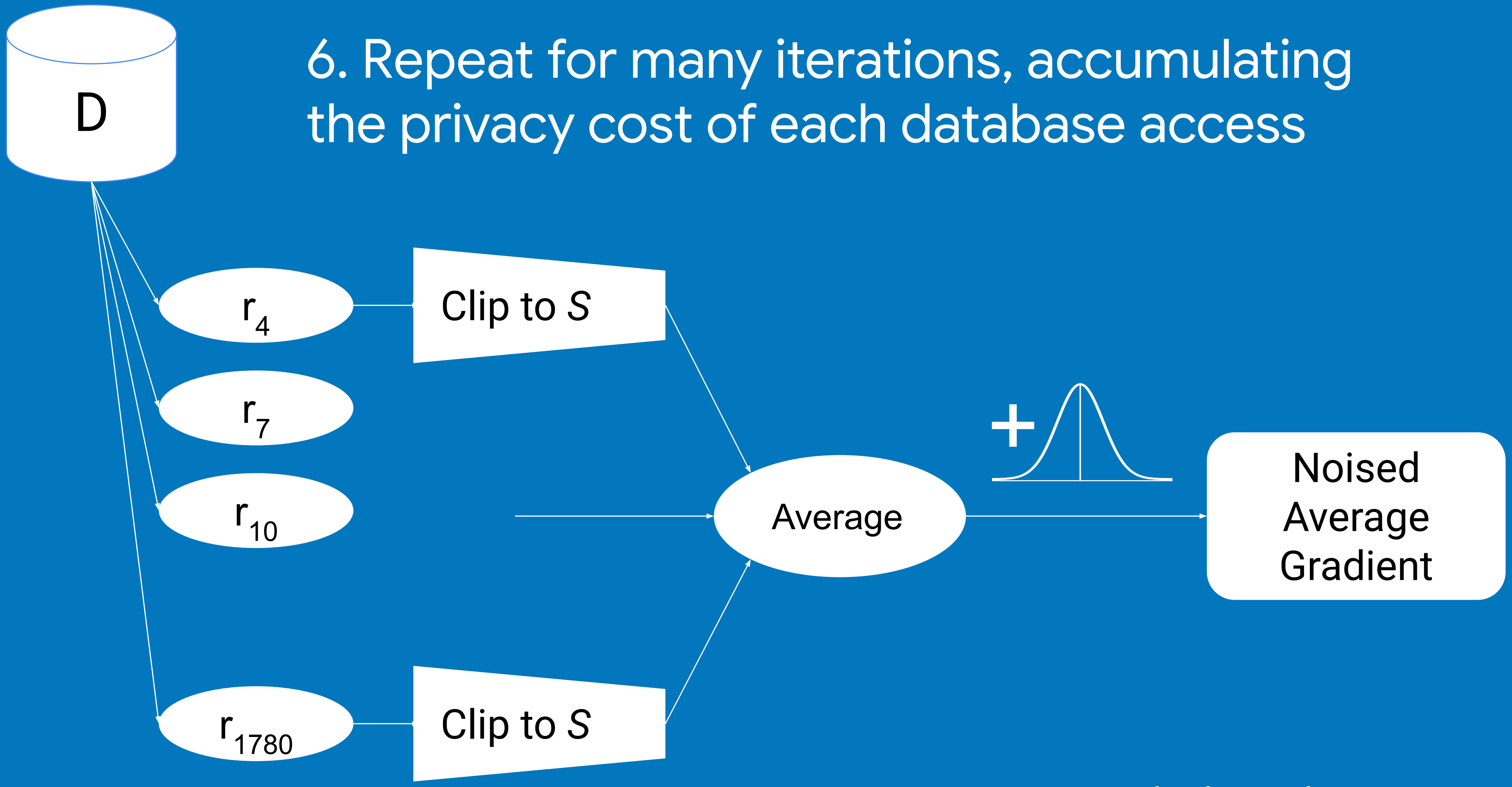
5. Take a step of SGD



M. Abadi, et. al. **Deep Learning with Differential Privacy.** CCS 2016.

Iterative training with differential privacy

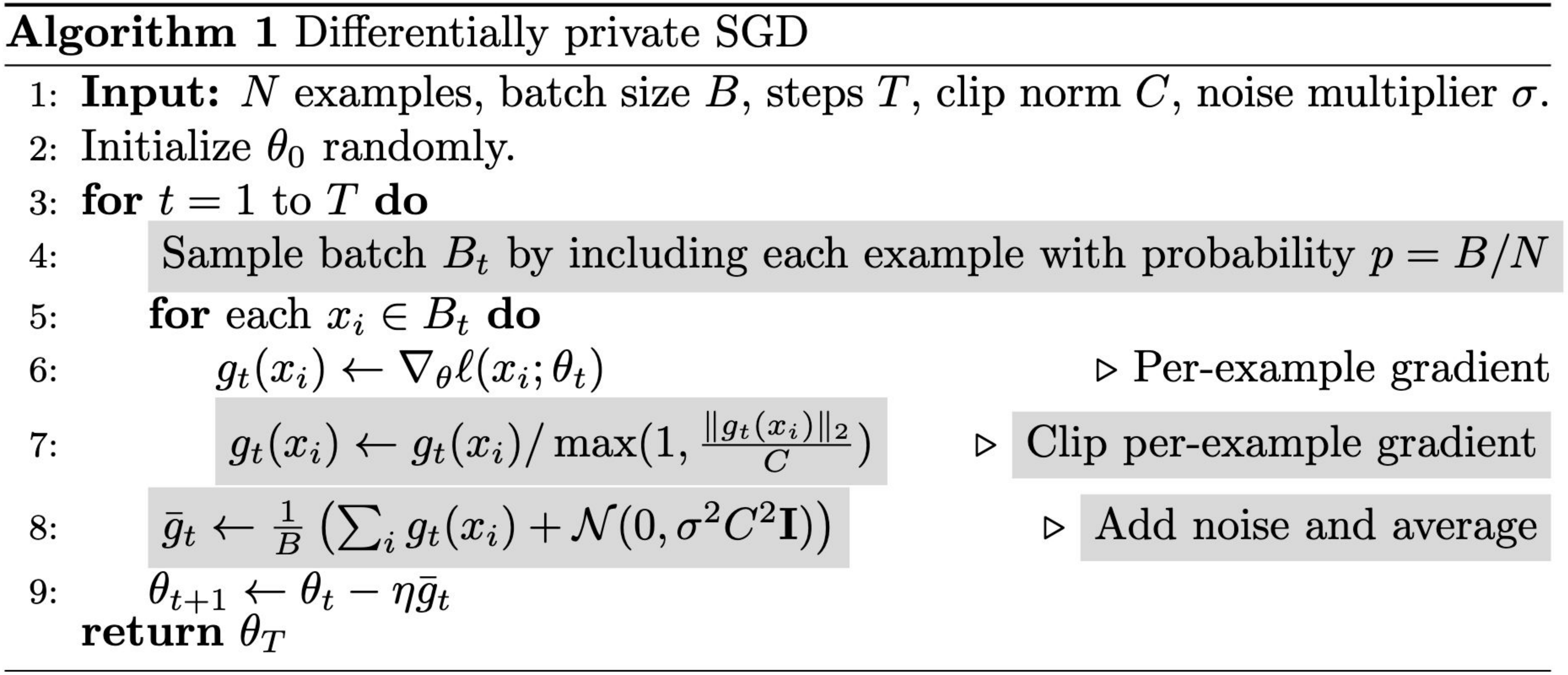
6. Repeat for many iterations, accumulating the privacy cost of each database access



M. Abadi, et. al. **Deep Learning with Differential Privacy.** CCS 2016.

Google Research

DP-SGD



Algorithm 1 Differentially private SGD

```
Input: $N$ examples, batch size $B$, steps $T$, clip norm $C$, noise multiplier $\sigma$.
Initialize $\theta_0$ randomly.
for $t = 1$ to $T$ do
    Sample batch $B_t$ by including each example with probability $p = B/N$
    for each $x_i \in B_t$ do
        $g_t(x_i) \leftarrow \nabla_\theta \ell(x_i; \theta_t)$                         ▷ Per-example gradient
        $g_t(x_i) \leftarrow g_t(x_i) / \max(1, \frac{\|g_t(x_i)\|_2}{C})$          ▷ Clip per-example gradient
    $\bar{g}_t \leftarrow \frac{1}{B} \left( \sum_i g_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}) \right)$     ▷ Add noise and average
    $\theta_{t+1} \leftarrow \theta_t - \eta \bar{g}_t$
return $\theta_T$
```

DP-SGD

Algorithm 1 *Regular SGD*

```
1: Input: $N$ examples, batch size $B$, iterations $T$.
2: Initialize $\theta_0$ randomly.
3: for $t = 1$ to $T$ do
4:     Draw batch $B_t$ cyclically from the dataset
5:     for each $x_i \in B_t$ do
6:         $g_t(x_i) \leftarrow \nabla_\theta \ell(x_i; \theta_t)$                    ▷ Per-example gradient
7:         $g_t(x_i) \leftarrow g_t(x_i)$                    ▷ Do nothing to per-example gradients
8:     $\bar{g}_t \leftarrow \frac{1}{B}\left(\sum_i g_t(x_i)\right)$                    ▷ Do not add noise and average
9:     $\theta_{t+1} \leftarrow \theta_t - \eta \bar{g}_t$
   return $\theta_T$
```

DP-SGD

Algorithm 1 Differentially private SGD

```
1: Input: $N$ examples, batch size $B$, steps $T$, clip norm $C$, noise multiplier $\sigma$.
2: Initialize $\theta_0$ randomly.
3: for $t = 1$ to $T$ do
4:     Sample batch $B_t$ by including each example with probability $p = B/N$
5:     for each $x_i \in B_t$ do
6:         $g_t(x_i) \leftarrow \nabla_\theta \ell(x_i; \theta_t)$                ▷ Per-example gradient
7:         $g_t(x_i) \leftarrow g_t(x_i) / \max(1, \frac{\|g_t(x_i)\|_2}{C})$      ▷ Clip per-example gradient
8:     $\bar{g}_t \leftarrow \frac{1}{B}\left(\sum_i g_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$      ▷ Add noise and average
9:     $\theta_{t+1} \leftarrow \theta_t - \eta \bar{g}_t$
   return $\theta_T$
```

DP Accounting: based on parameters chosen, compute privacy guarantee *ε(N, B, T, σ)*

Per-example gradient clipping in JAX

```
N = 3
w = np.random.randn(2) # (D,)
X = np.random.randn(N, 2) # (N, D)
y = np.random.randn(N) # (N,)

def loss(w, data):
  X, y = data
  return jnp.sum((X @ w - y)**2)
```

Simple regression example

Google Research

Per-example gradient clipping in JAX

```
N = 3
w = np.random.randn(2) # (D,)
X = np.random.randn(N, 2) # (N, D)
y = np.random.randn(N) # (N,)

def loss(w, data):
  X, y = data
  return jnp.sum((X @ w - y)**2)


grad_fn = jax.grad(loss)
>>> grad_fn(w, (X, y))
Array([17.44384  ,  6.9017286], dtype=float32)
```

Simple regression example

Compute gradients wrt. parameters `w`

- sum of per-example gradients

Training loop: update `w`, draw a new batch, …

Google Research

Per-example gradient clipping in JAX

```
# grad_fn: w, data -> dl/dw

grads = jnp.stack(
    [grad_fn(w, (xi, yi)) for xi, yi in zip(X, y)]
) # (N, D)
```

Naive per-example grads

Google Research

Per-example gradient clipping in JAX

```
# grad_fn: w, data -> dl/dw

grads = jnp.stack(
    [grad_fn(w, (xi, yi)) for xi, yi in zip(X, y)]
) # (N, D)
>>> grads
Array([[ 5.3370066 , 12.219472  ],
       [ 9.310922  , -4.8723297 ],
       [ 2.795912  , -0.44541267]], dtype=float32)
```

Naive per-example grads

Google Research

Per-example gradient clipping in JAX

```
# grad_fn: w, data -> dl/dw

grads = jnp.stack(
    [grad_fn(w, (xi, yi)) for xi, yi in zip(X, y)]
) # (N, D)

def clip(g, C):
  return g * jnp.minimum(1, C / jnp.linalg.norm(g))

clipped_grads = jnp.stack([clip(g, 1) for g in grads])
```

Naive per-example grads

Google Research

Per-example gradient clipping in JAX

```
# grad_fn: w, data -> dl/dw

grads = jnp.stack(
    [grad_fn(w, (xi, yi)) for xi, yi in zip(X, y)]
) # (N, D)

def clip(g, C):
  return g * jnp.minimum(1, C / jnp.linalg.norm(g))

clipped_grads = jnp.stack([clip(g, 1) for g in grads])
>>> clipped_grads
Array([[ 0.40025145,  0.9164054 ],
       [ 0.88601995, -0.46364704],
       [ 0.987547  , -0.15732467]], dtype=float32)
```

Naive per-example grads

Google Research

Per-example gradient clipping in JAX

```
# grad_fn: w, data -> dl/dw

grads = jnp.stack(
    [grad_fn(w, (xi, yi)) for xi, yi in zip(X, y)]
) # (N, D)

def clip(g, C):
  return g * jnp.minimum(1, C / jnp.linalg.norm(g))

clipped_grads = jnp.stack([clip(g, 1) for g in grads])
>>> jnp.sum(clipped_grads, axis=0) # (D,)
Array([2.2738183 , 0.29543367], dtype=float32)
```

Naive per-example grads

Add noise to this, and use instead of the regular grad

Google Research

Per-example gradient clipping in JAX

```
# grad_fn: w, data -> dl/dw

grads = jnp.stack(
    [grad_fn(w, (xi, yi)) for xi, yi in zip(X, y)]
) # (N, D)
                                           SLOW
def clip(g, C):
  return g * jnp.minimum(1, C / jnp.linalg.norm(g))

clipped_grads = jnp.stack([clip(g, 1) for g in grads])
>>> jnp.sum(clipped_grads, axis=0) # (D,)
Array([2.2738183 , 0.29543367], dtype=float32)
```

Naive per-example grads

Add noise to this, and use instead of the regular grad

Google Research

Per-example gradient clipping in JAX

```
# grad_fn: w, data -> dl/dw

grads = jnp.stack(
    [grad_fn(w, (xi, yi)) for xi, yi in zip(X, y)]
) # (N, D)
```

SLOW

```
vmap_grad_fn = jax.vmap(
    grad_fn,
    in_axes=(None, (0, 0))
)
grads = vmap_grad_fn(w, (X, y)) # (N, D)
```

Using `jax.vmap`

Vectorize over the batch axis of the data argument

Google Research

Per-example gradient clipping in JAX

```
# grad_fn: w, data -> dl/dw

grads = jnp.stack(
    [grad_fn(w, (xi, yi)) for xi, yi in zip(X, y)]
) # (N, D)
                                            SLOW

vmap_grad_fn = jax.vmap(
    grad_fn,
    in_axes=(None, (0, 0))
)
grads = vmap_grad_fn(w, (X, y)) # (N, D)
>>> grads
Array([[ 5.337006  , 12.219471  ],
       [ 9.310922  , -4.8723297 ],
       [ 2.795912  , -0.44541267]], dtype=float32)
```

Using `jax.vmap`

Vectorize over the batch axis of the data argument

Per-example gradient clipping in JAX

```
# grad_fn: w, data -> dl/dw

grads = jnp.stack(
    [grad_fn(w, (xi, yi)) for xi, yi in zip(X, y)]
) # (N, D)
                                        SLOW

vmap_grad_fn = jax.vmap(
    grad_fn,
    in_axes=(None, (0, 0)),
    spmd_axis_name="devices"
)
```

Using `jax.vmap`

Vectorize over the batch axis of the data argument

Parallelize the operation across devices

Google Research

Per-example gradient clipping in JAX

```
# loss: w, data -> float

grad_fn = jax.grad(loss)
>>> grad_fn(w, (X, y))
Array([17.44384  ,  6.9017286], dtype=float32)
```

Regular training

Google Research

Per-example gradient clipping in JAX

```
# loss: w, data -> float

grad_fn = jax.grad(loss)
>>> grad_fn(w, (X, y))
Array([17.44384  ,  6.9017286], dtype=float32)
```

Regular training

```
# ... vmap + clip
clipped_grads = jnp.stack([clip(g, 1) for g in grads])
>>> jnp.sum(clipped_grads, axis=0) # (D,)
Array([2.2738183 , 0.29543367], dtype=float32)
```

Replacement for `grad`

Google Research

Per-example gradient clipping in JAX

```
# loss: w, data -> float

grad_fn = jax.grad(loss)
>>> grad_fn(w, (X, y))
Array([17.44384  ,  6.9017286], dtype=float32)
```

Regular training

```
# ... vmap + clip
clipped_grads = jnp.stack([clip(g, 1) for g in grads])
>>> jnp.sum(clipped_grads, axis=0) # (D,)
Array([2.2738183 , 0.29543367], dtype=float32)
```

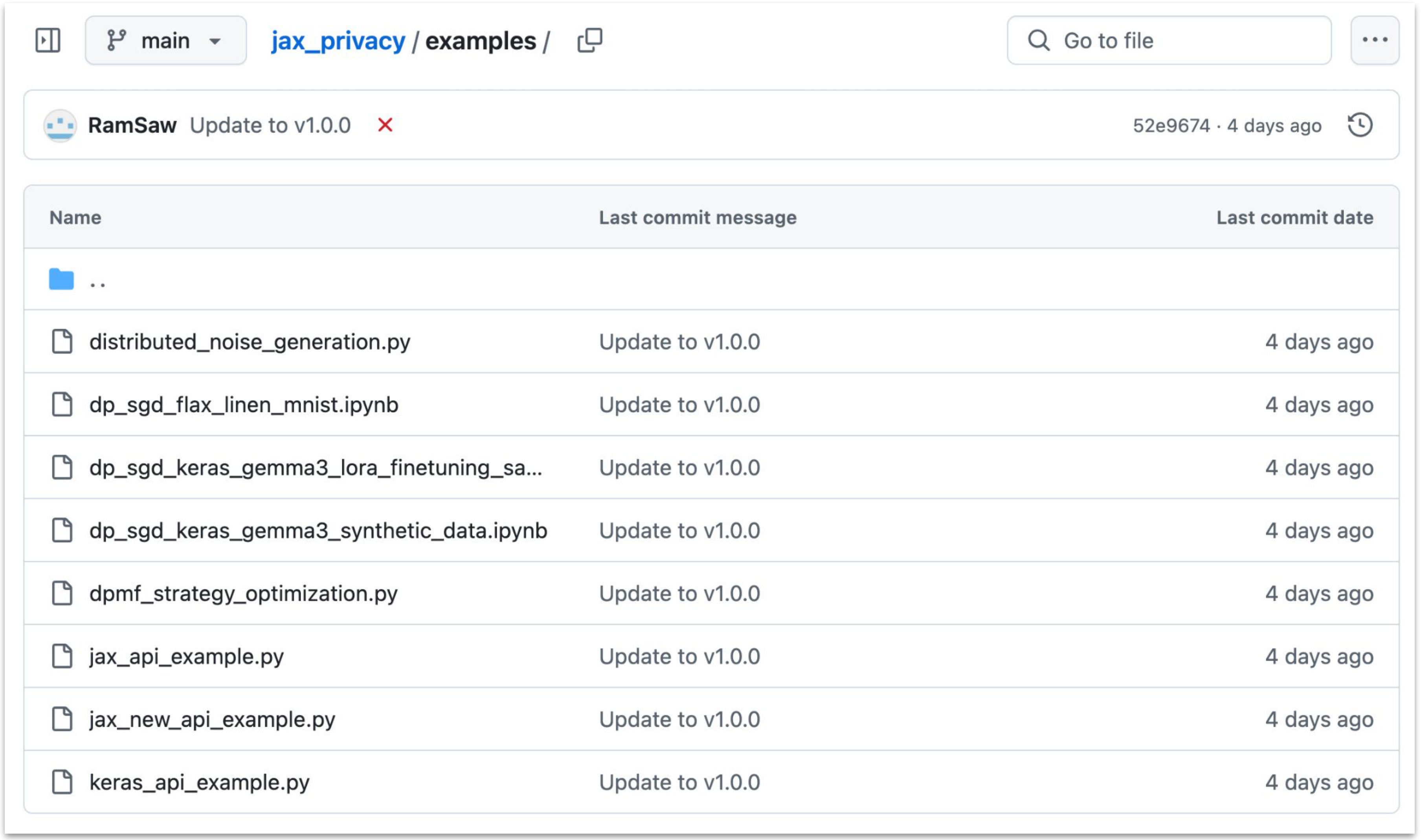
Drop in replacement for `jax.grad`!

```
import jax_privacy
clipped_grad_fn = jax_privacy.experimental.clipped_grad(loss, l2_clip_norm=1)
>>> clipped_grad_fn(w, (X, y))
Array([2.2738183 , 0.29543367], dtype=float32)
```

Google Research

github.com/google-deepmind/jax_privacy

jax_privacy v1.0.0 update



main

jax_privacy / examples /

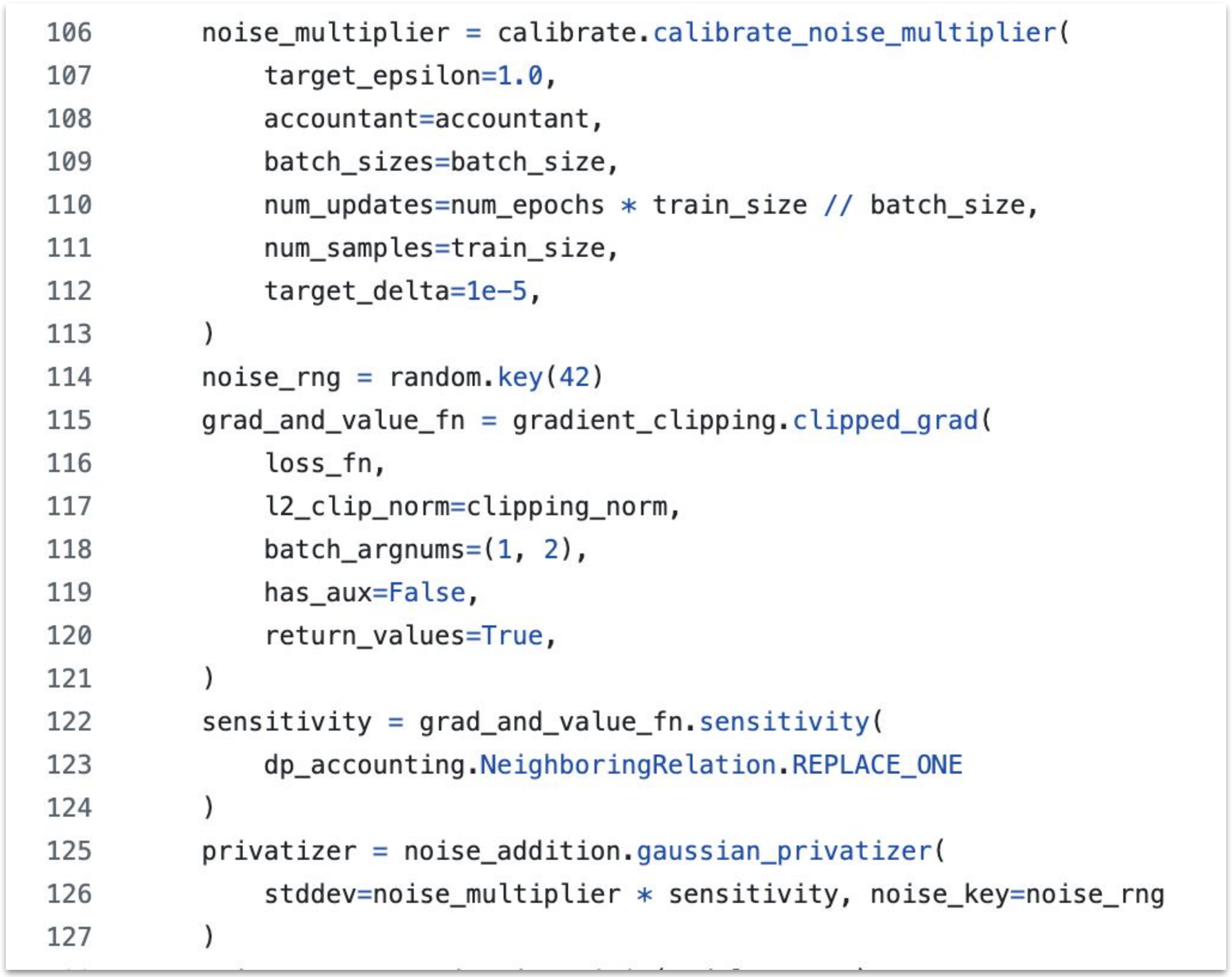
Go to file

RamSaw Update to v1.0.0 ✗ 52e9674 · 4 days ago

Name	Last commit message	Last commit date
..		
distributed_noise_generation.py	Update to v1.0.0	4 days ago
dp_sgd_flax_linen_mnist.ipynb	Update to v1.0.0	4 days ago
dp_sgd_keras_gemma3_lora_finetuning_sa...	Update to v1.0.0	4 days ago
dp_sgd_keras_gemma3_synthetic_data.ipynb	Update to v1.0.0	4 days ago
dpmf_strategy_optimization.py	Update to v1.0.0	4 days ago
jax_api_example.py	Update to v1.0.0	4 days ago
jax_new_api_example.py	Update to v1.0.0	4 days ago
keras_api_example.py	Update to v1.0.0	4 days ago

Google Research

github.com/google-deepmind/jax_privacy

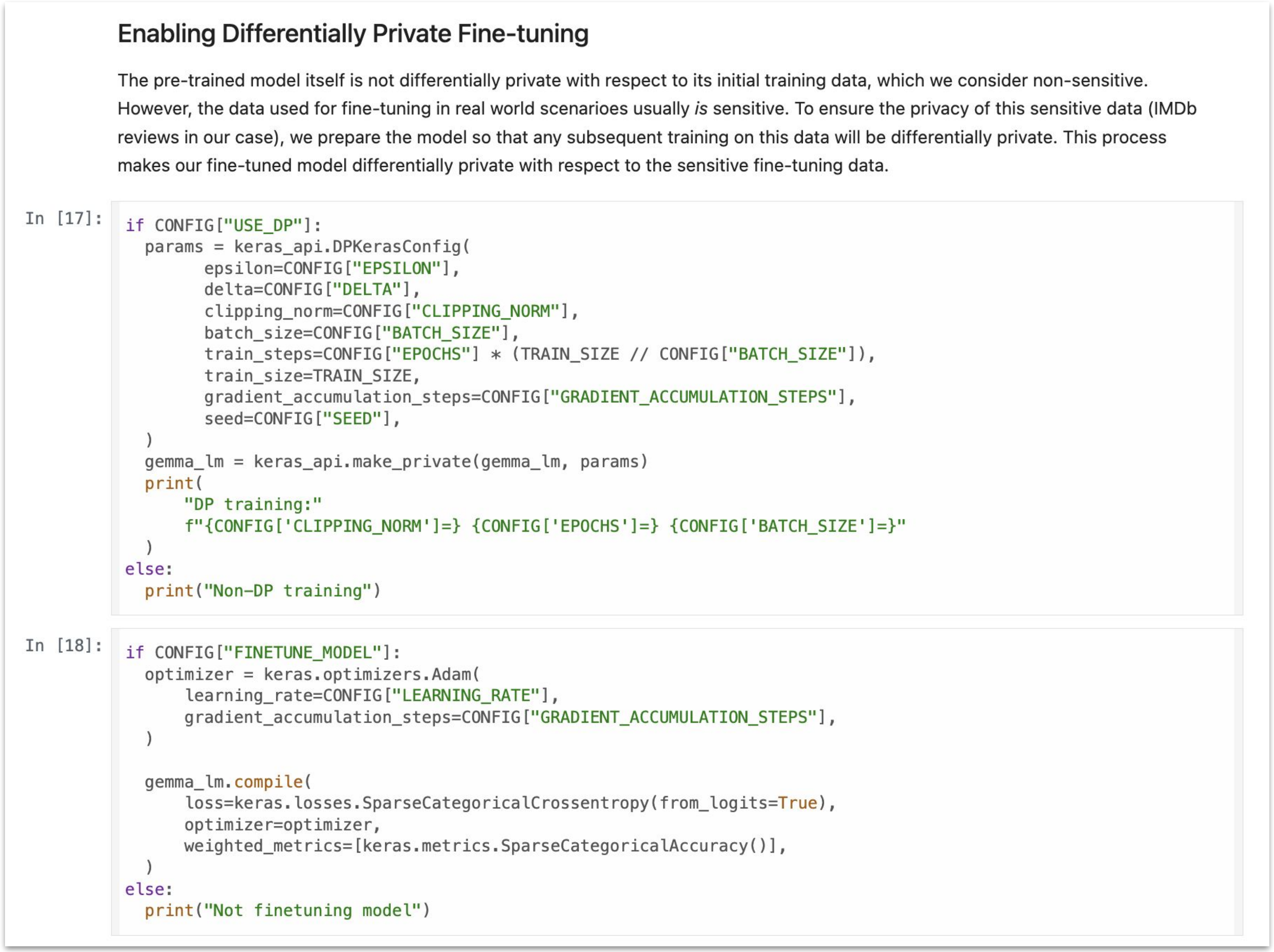


```
    noise_multiplier = calibrate.calibrate_noise_multiplier(
        target_epsilon=1.0,
        accountant=accountant,
        batch_sizes=batch_size,
        num_updates=num_epochs * train_size // batch_size,
        num_samples=train_size,
        target_delta=1e-5,
    )
    noise_rng = random.key(42)
    grad_and_value_fn = gradient_clipping.clipped_grad(
        loss_fn,
        l2_clip_norm=clipping_norm,
        batch_argnums=(1, 2),
        has_aux=False,
        return_values=True,
    )
    sensitivity = grad_and_value_fn.sensitivity(
        dp_accounting.NeighboringRelation.REPLACE_ONE
    )
    privatizer = noise_addition.gaussian_privatizer(
        stddev=noise_multiplier * sensitivity, noise_key=noise_rng
    )
```

Example training loop:
[jax_privacy/examples/jax_new_api_example.py]

Google Research

github.com/google-deepmind/jax_privacy



Enabling Differentially Private Fine-tuning

The pre-trained model itself is not differentially private with respect to its initial training data, which we consider non-sensitive. However, the data used for fine-tuning in real world scenarioes usually *is* sensitive. To ensure the privacy of this sensitive data (IMDb reviews in our case), we prepare the model so that any subsequent training on this data will be differentially private. This process makes our fine-tuned model differentially private with respect to the sensitive fine-tuning data.

In [17]:
```
if CONFIG["USE_DP"]:
  params = keras_api.DPKerasConfig(
        epsilon=CONFIG["EPSILON"],
        delta=CONFIG["DELTA"],
        clipping_norm=CONFIG["CLIPPING_NORM"],
        batch_size=CONFIG["BATCH_SIZE"],
        train_steps=CONFIG["EPOCHS"] * (TRAIN_SIZE // CONFIG["BATCH_SIZE"]),
        train_size=TRAIN_SIZE,
        gradient_accumulation_steps=CONFIG["GRADIENT_ACCUMULATION_STEPS"],
        seed=CONFIG["SEED"],
  )
  gemma_lm = keras_api.make_private(gemma_lm, params)
  print(
      "DP training:"
      f"{CONFIG['CLIPPING_NORM']=} {CONFIG['EPOCHS']=} {CONFIG['BATCH_SIZE']=}"
  )
else:
  print("Non-DP training")
```

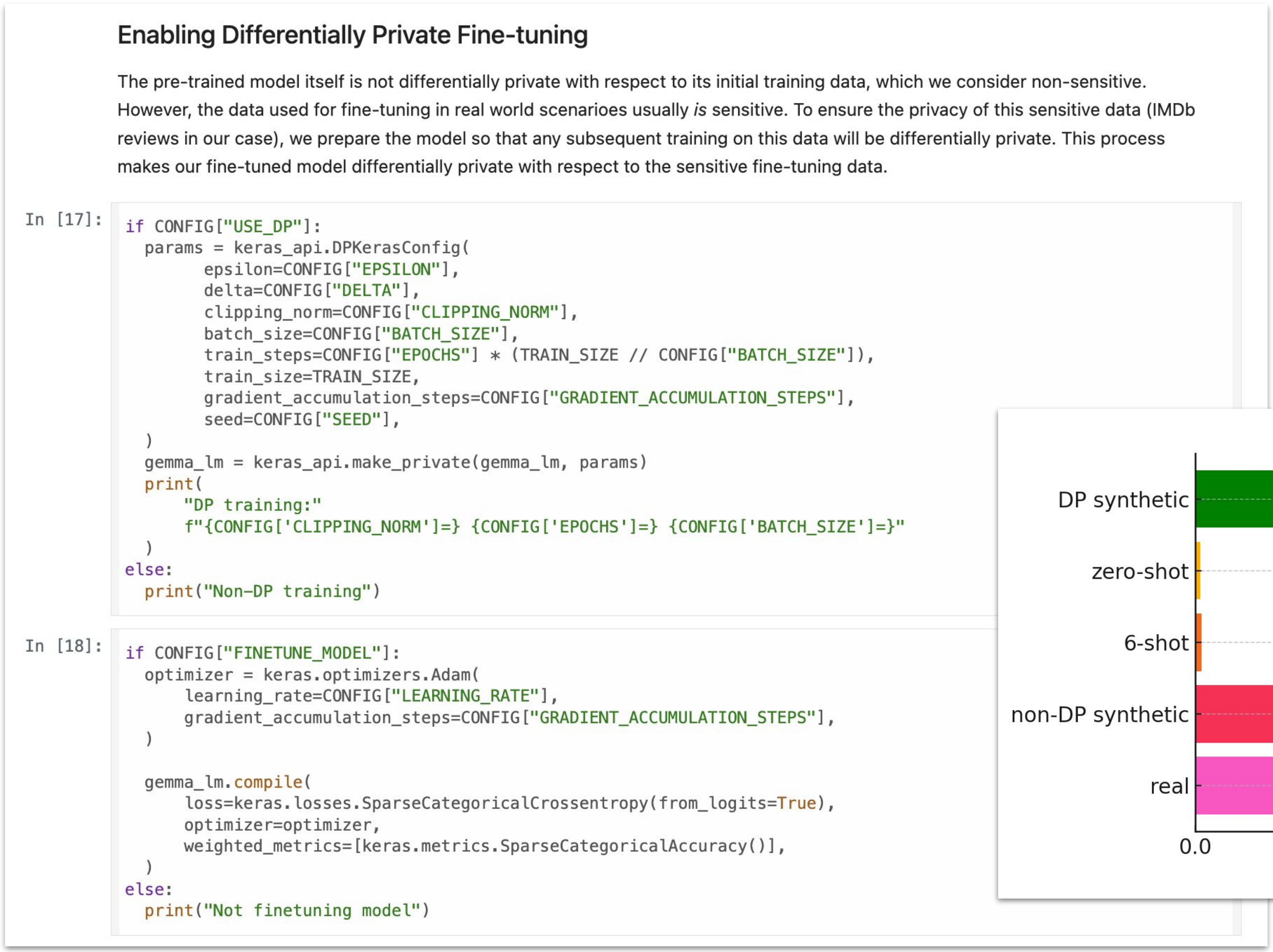
In [18]:
```
if CONFIG["FINETUNE_MODEL"]:
  optimizer = keras.optimizers.Adam(
      learning_rate=CONFIG["LEARNING_RATE"],
      gradient_accumulation_steps=CONFIG["GRADIENT_ACCUMULATION_STEPS"],
  )

  gemma_lm.compile(
      loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
      optimizer=optimizer,
      weighted_metrics=[keras.metrics.SparseCategoricalAccuracy()],
  )
else:
  print("Not finetuning model")
```

Colab tutorial:
[jax_privacy/examples/dp_sgd_keras_gemma3_synthetic_data.ipynb]

Google Research

github.com/google-deepmind/jax_privacy



Enabling Differentially Private Fine-tuning

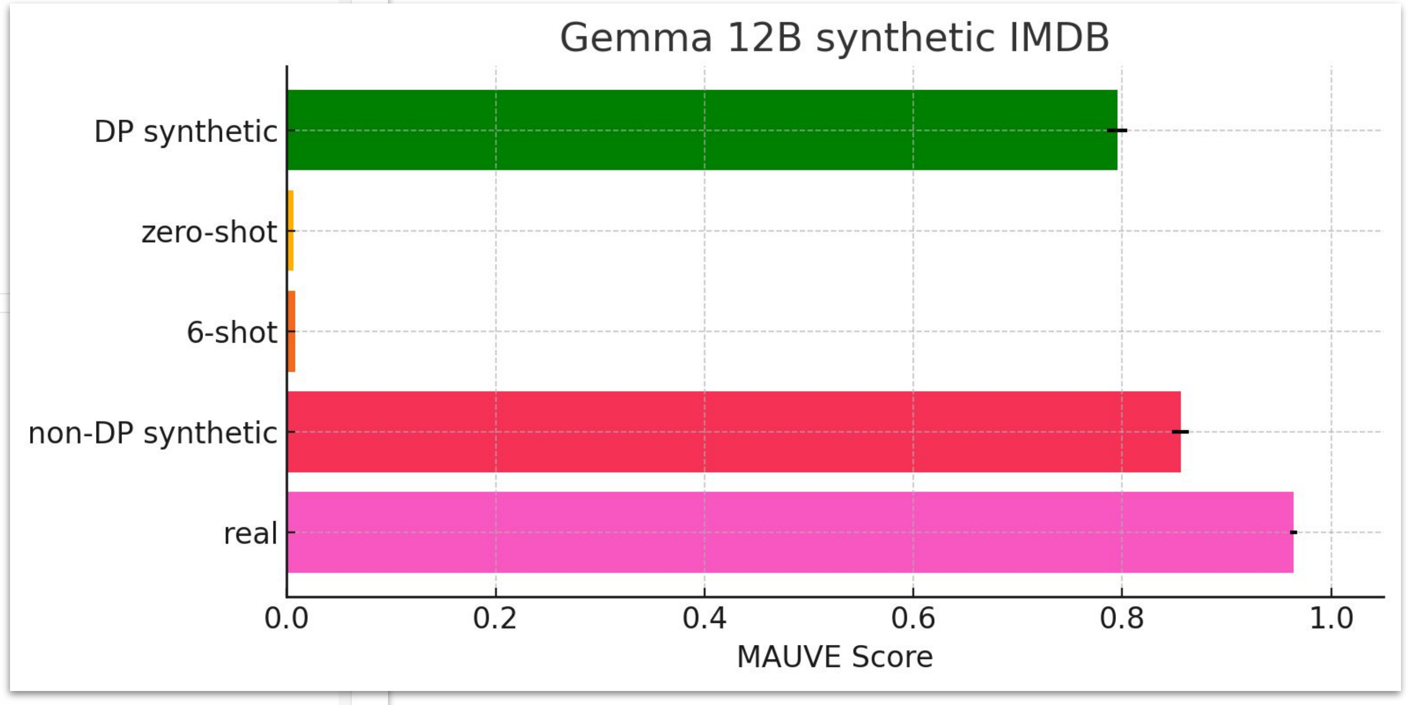
The pre-trained model itself is not differentially private with respect to its initial training data, which we consider non-sensitive. However, the data used for fine-tuning in real world scenarioes usually *is* sensitive. To ensure the privacy of this sensitive data (IMDb reviews in our case), we prepare the model so that any subsequent training on this data will be differentially private. This process makes our fine-tuned model differentially private with respect to the sensitive fine-tuning data.

In [17]:
```
if CONFIG["USE_DP"]:
  params = keras_api.DPKerasConfig(
        epsilon=CONFIG["EPSILON"],
        delta=CONFIG["DELTA"],
        clipping_norm=CONFIG["CLIPPING_NORM"],
        batch_size=CONFIG["BATCH_SIZE"],
        train_steps=CONFIG["EPOCHS"] * (TRAIN_SIZE // CONFIG["BATCH_SIZE"]),
        train_size=TRAIN_SIZE,
        gradient_accumulation_steps=CONFIG["GRADIENT_ACCUMULATION_STEPS"],
        seed=CONFIG["SEED"],
  )
  gemma_lm = keras_api.make_private(gemma_lm, params)
  print(
      "DP training:"
      f"{CONFIG['CLIPPING_NORM']=} {CONFIG['EPOCHS']=} {CONFIG['BATCH_SIZE']=}"
  )
else:
  print("Non-DP training")
```

In [18]:
```
if CONFIG["FINETUNE_MODEL"]:
  optimizer = keras.optimizers.Adam(
      learning_rate=CONFIG["LEARNING_RATE"],
      gradient_accumulation_steps=CONFIG["GRADIENT_ACCUMULATION_STEPS"],
  )

  gemma_lm.compile(
      loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
      optimizer=optimizer,
      weighted_metrics=[keras.metrics.SparseCategoricalAccuracy()],
  )
else:
  print("Not finetuning model")
```

Colab tutorial: [jax_privacy/examples/dp_sgd_keras_gemma3_synthetic_data.ipynb]



Google Research

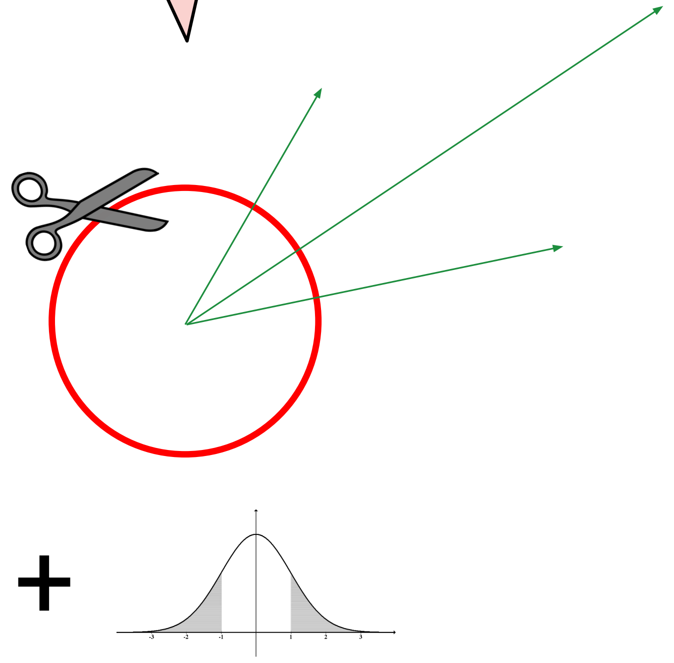
Hyperparameter tuning

Important hparams: Steps ***T***, Batch size ***B***, clipping norm ***C***, noise multiplier ***σ***

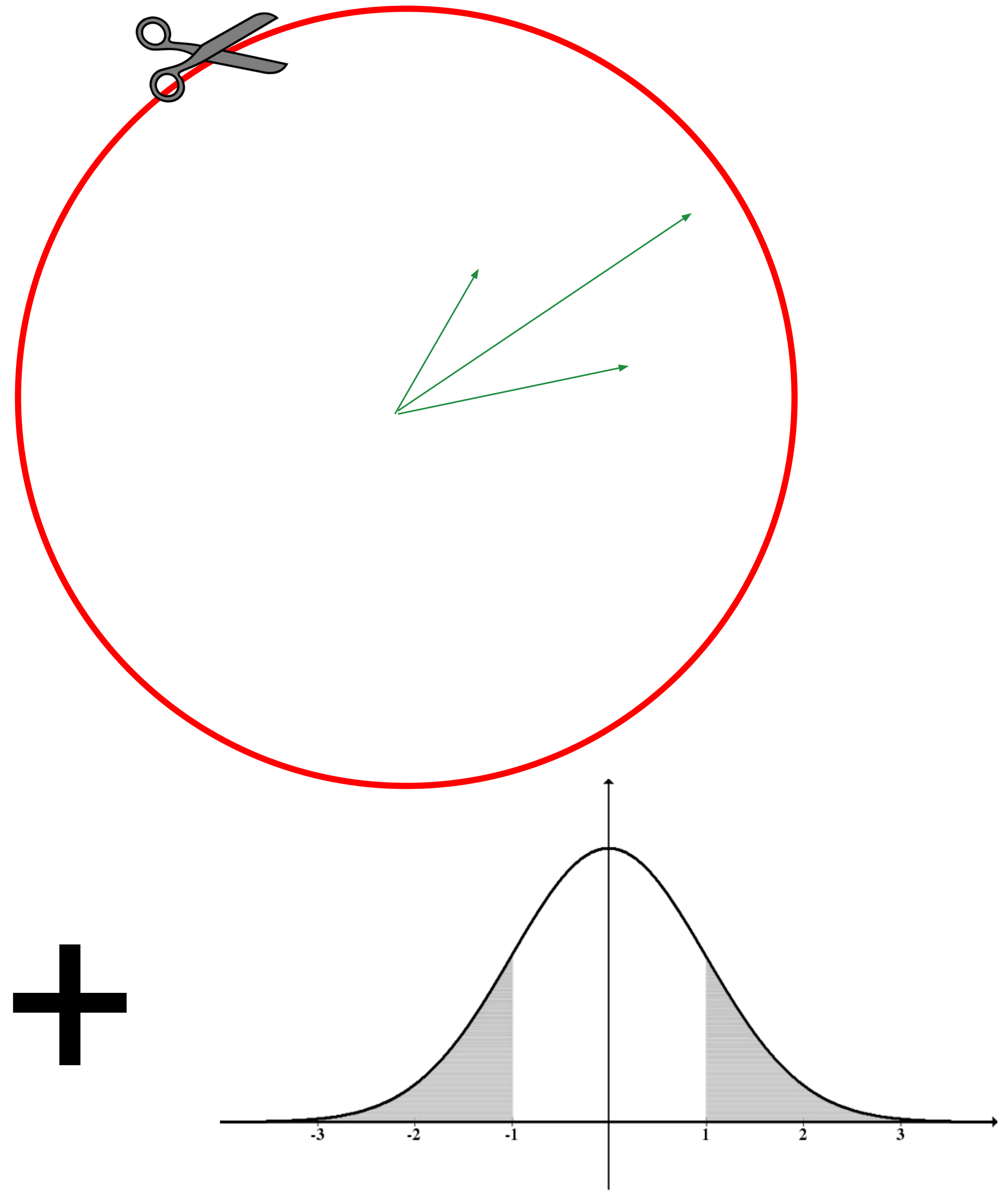
$$\tilde{g} = \frac{1}{B}\left(\sum_i \mathrm{clip}_C(g_i) + \mathcal{N}(0, \sigma^2 C^2 I)\right)$$

1. Under a particular threshold ***C***, a large range works:
    a. Pick ***C*** so that almost all gradients are clipped, which corresponds to a biased, but less noisy update

Small clip norm: update magnitudes lost



Large clip norm: high noise for DP



Google Research

Hyperparameter tuning

Important hparams: Steps ***T***, Batch size ***B***, clipping norm ***C***, noise multiplier ***σ***

$$\tilde{g} = \frac{1}{B}\left(\sum_i \mathrm{clip}_C(g_i) + \mathcal{N}(0, \sigma^2 C^2 I)\right)$$

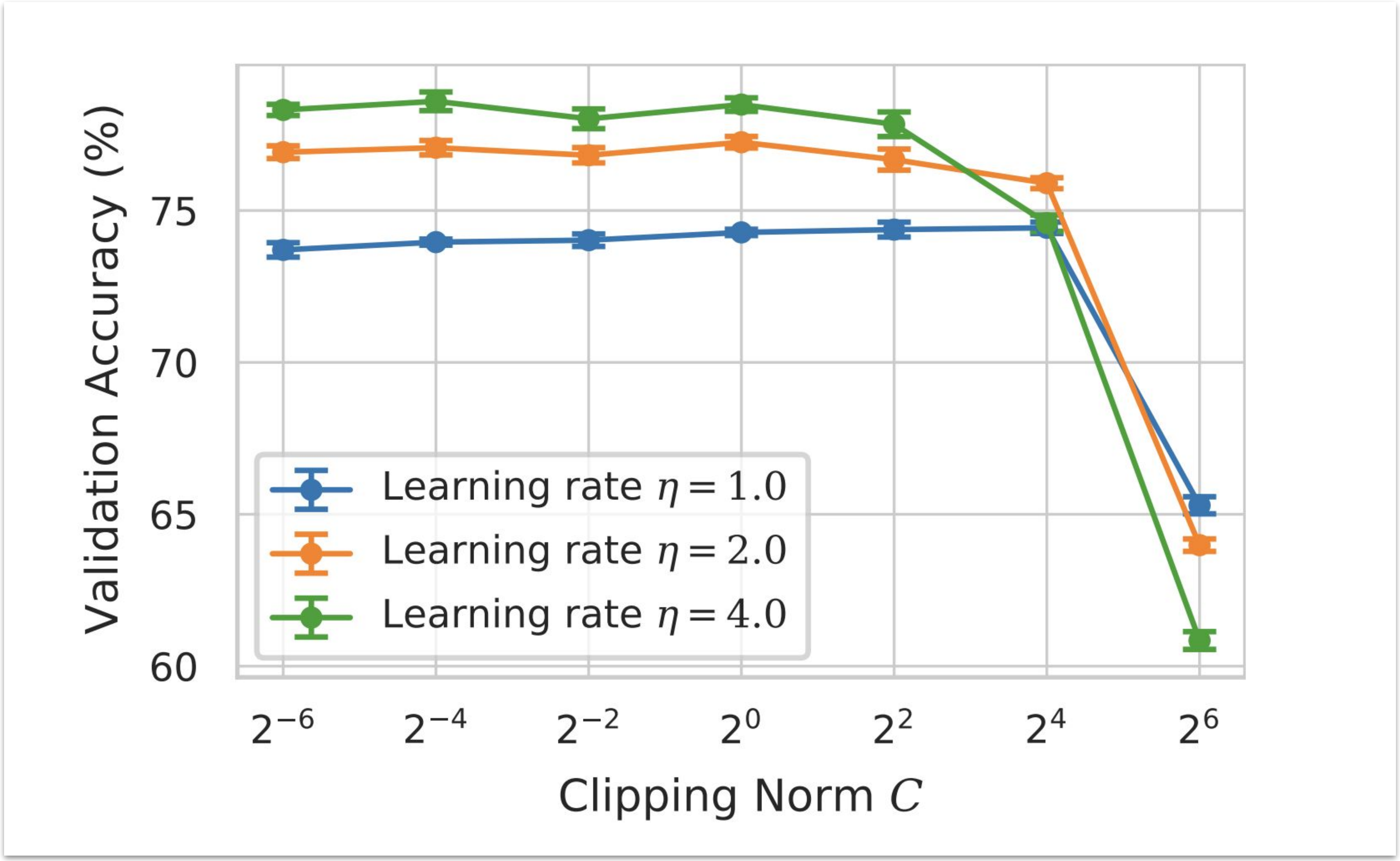
1. Under a particular threshold ***C***, a large range works:
   a. Pick ***C*** so that almost all gradients are clipped, which corresponds to a biased, but less noisy update
2. Normalize the update to decouple learning rate from ***C***

Hyperparameter tuning

Important hparams: Steps ***T***, Batch size ***B***, clipping norm ***C***, noise multiplier ***σ***

$$\tilde{g} = \frac{1}{B}\left(\sum_i \boxed{\frac{1}{C}}\text{clip}_C(g_i) + \mathcal{N}(0,\sigma^2 I)\right)$$

1. Under a particular threshold ***C***, a large range works:
   a. Pick ***C*** so that almost all gradients are clipped, which corresponds to a biased, but less noisy update
2. Normalize the update to decouple learning rate from ***C***



Source: *Unlocking High-Accuracy Differentially Private Image Classification through Scale (De et al., 2022)*

Google Research

Hyperparameter tuning

- Important hparams: Steps ***T***, Batch size ***B***, clipping norm ***C***, noise multiplier ***σ***

$$\tilde{g} = \frac{1}{B}\left(\sum_i \frac{1}{C}\mathrm{clip}_C(g_i) + \mathcal{N}(0, \sigma^2 I)\right)$$

1. Under a particular threshold ***C***, a large range works:
    a. Pick ***C*** so that almost all gradients are clipped, which corresponds to a biased, but less noisy update
2. Normalize the update to decouple learning rate from ***C***
3. Under fixed **ε(*σ, B, T*)**, spending more compute helps compared to non-DP setup:
    a. Fixing ***B,*** more steps ***T*** and correspondingly larger ***σ*** usually helps
    b. Fixing ***T***, larger batch size ***B*** and corresponding larger ***σ*** usually helps

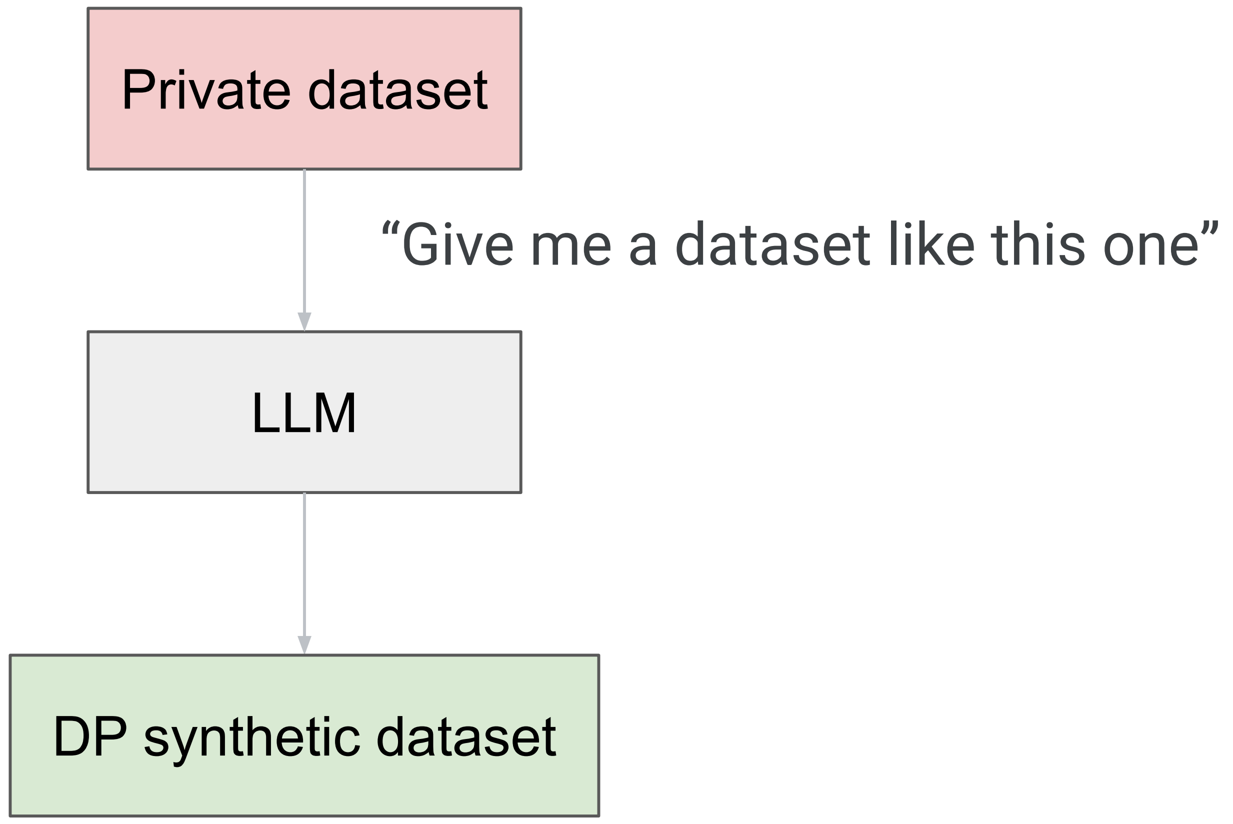
See: *Scaling Laws for Differentially Private Language Models* (McKenna et. al, ICML'25)

Inference-only methods

- Does not require a custom DP-SGD implementation or access to model weights.
- Operates on top of an existing inference stack.
- Faster iteration but worse quality

Analogous to tradeoffs between finetuning/prompt engineering.

Private evolution (*API access*)
DP inference (*logits access*)



Private evolution

Starting from a seed of entirely synthetic corpus,
Then **alternate between**:

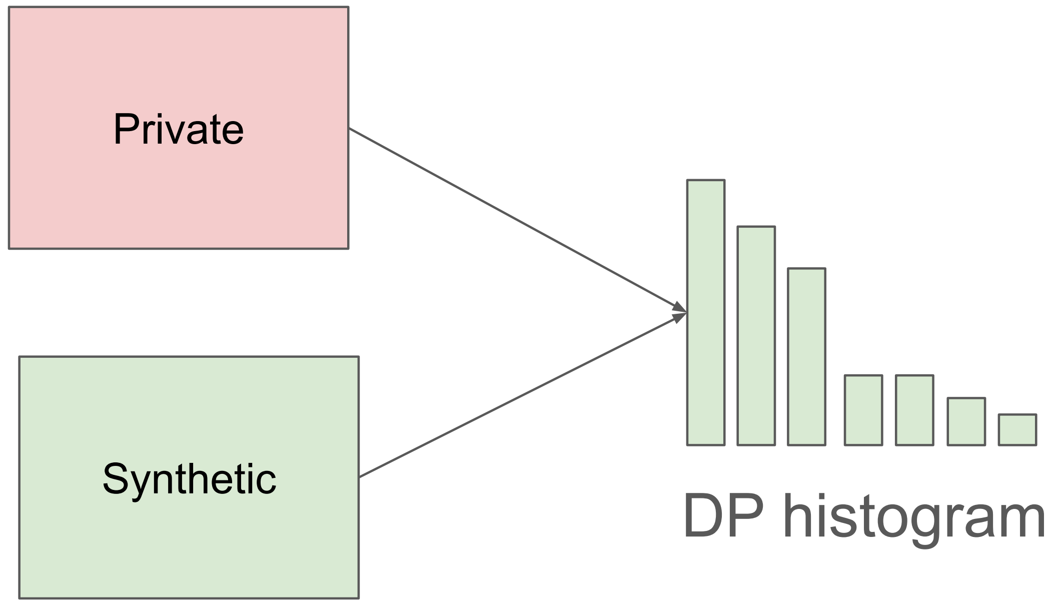
(A) filtering down to the examples most similar to private data; and
(B) expanding the corpus with LLM rewrites

Only place private data is used: finding the most similar data.

Best for low target epsilon/low private data volume scenarios

Google Research

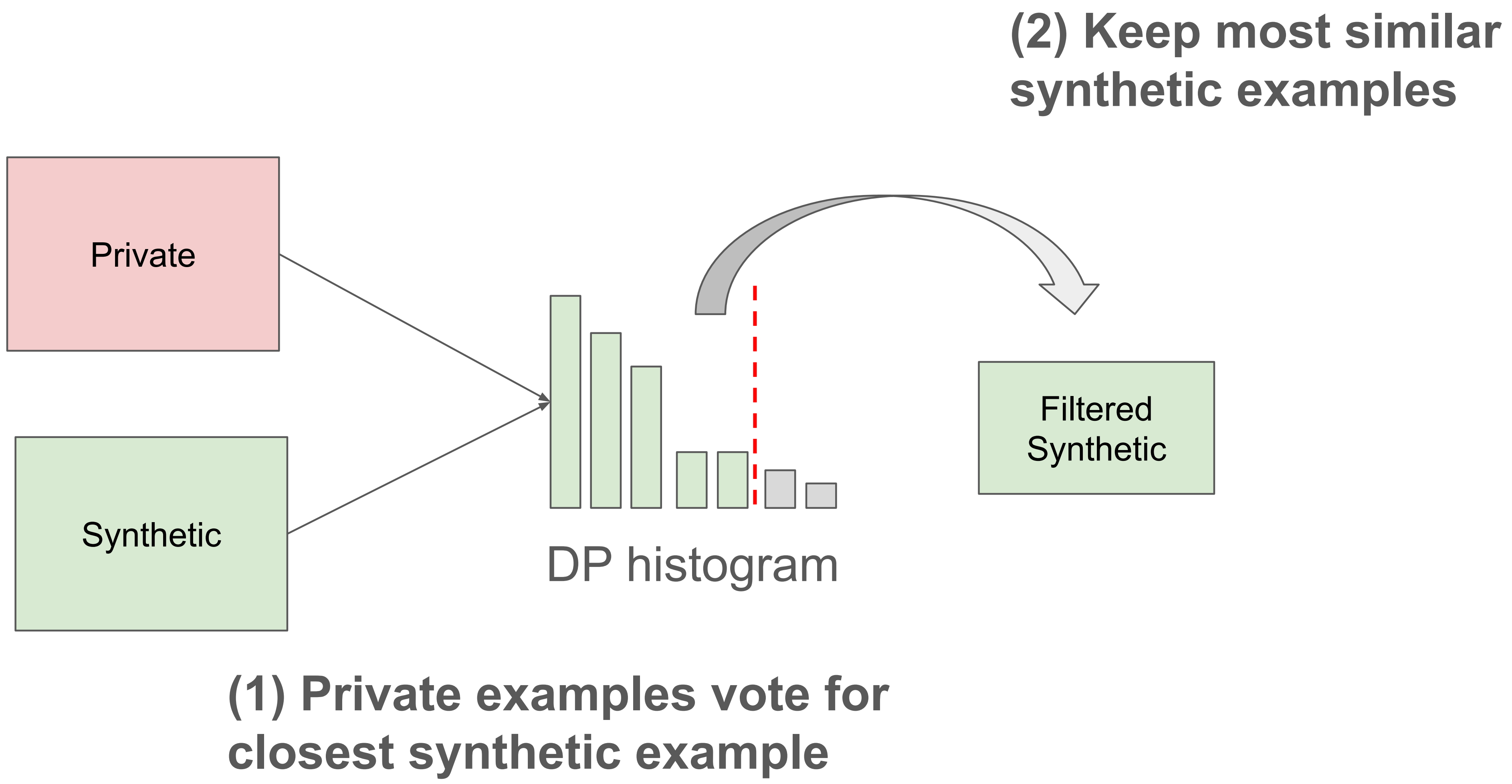
Private evolution



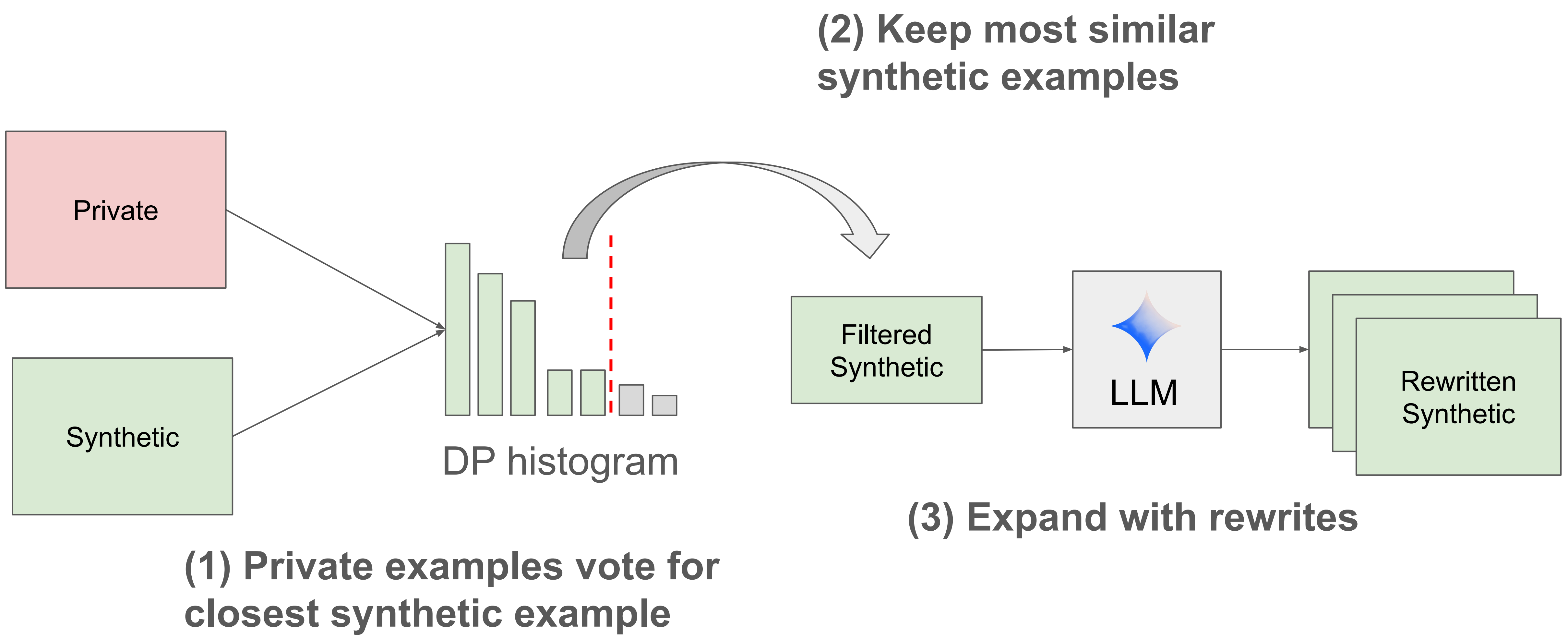
(1) Private examples vote for closest synthetic example

Google Research

Private evolution

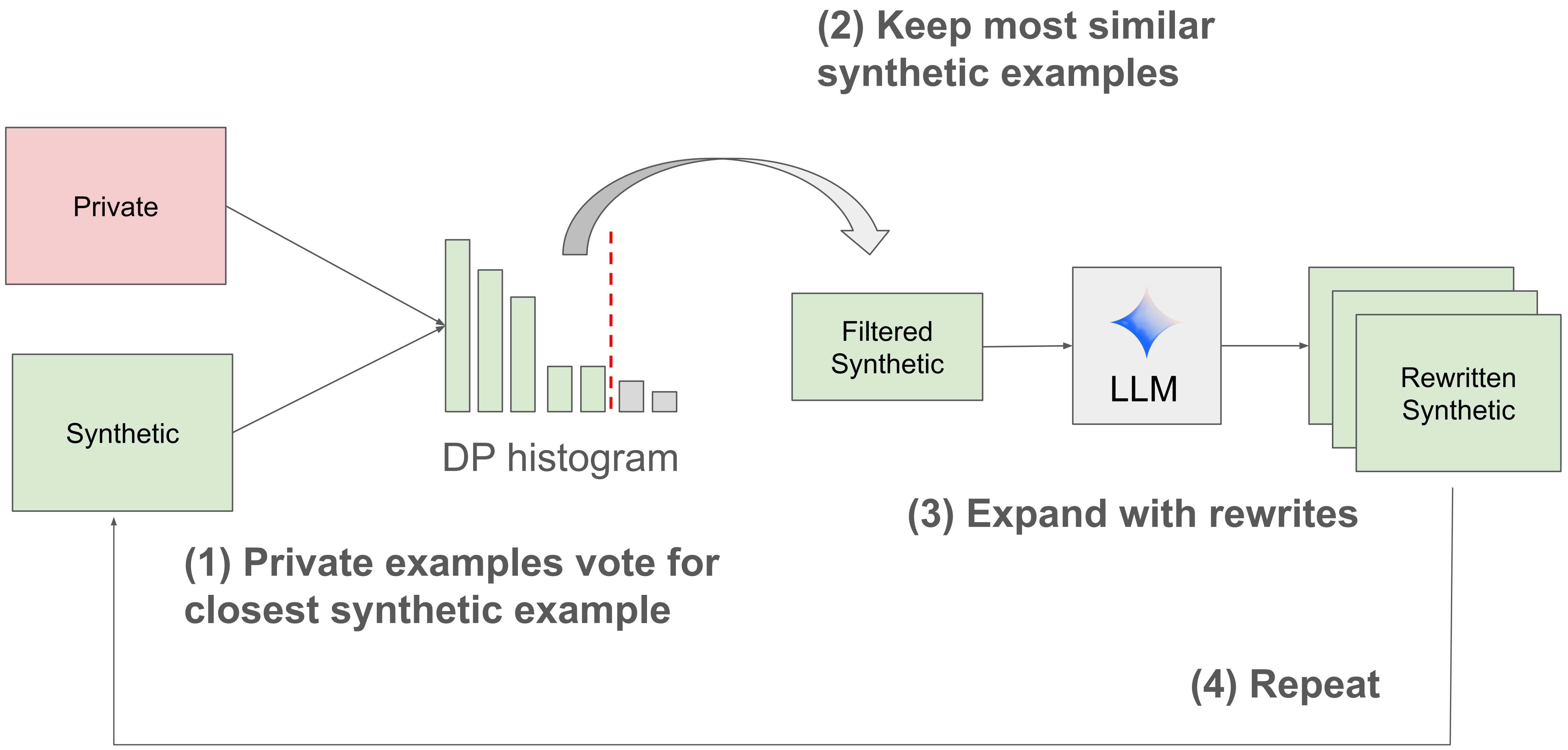

(2) Keep most similar synthetic examples
Private
Synthetic
DP histogram
Filtered Synthetic
(1) Private examples vote for closest synthetic example

Private evolution


(2) Keep most similar synthetic examples
Private
Synthetic
DP histogram
Filtered Synthetic
LLM
Rewritten Synthetic
(3) Expand with rewrites
(1) Private examples vote for closest synthetic example

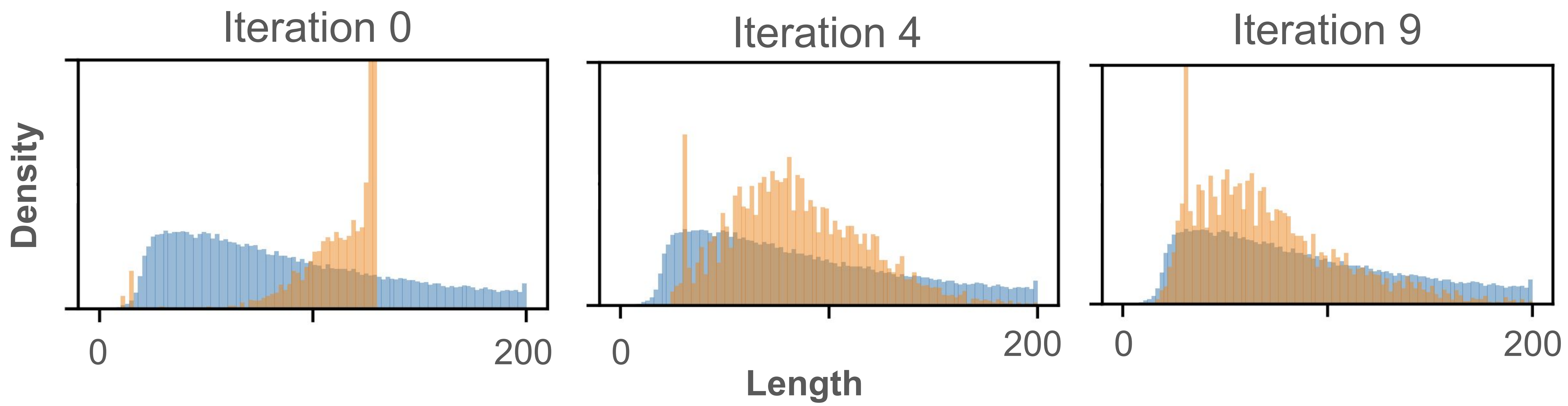
Google Research

Private evolution



Google Research

Private evolution



Source: [Differentially Private Synthetic Data via Foundation Model APIs 2: Text (Xie et al., ICML'24)]()

Google Research

Private evolution

1. Some prompt engineering required:
   - Seed corpus generation
   - Rewrite prompt should introduce appropriate amount of variation

Google Research

Private evolution

Seed corpus prompt

SYSTEM: You are required to write an example of review based on the provided Business Category and Review Stars that fall within the range of 1.0-5.0.

USER: Business Category: {label_1} | Review Stars: {label_2} with keyword {subcategory}

(subcategories generated by LLM)

Rewrite prompt

SYSTEM: You are a helpful, pattern-following assistant.

USER: Based on the Business Category and Review Stars, you are required to fill in the blanks in the Input sentences. If there are no blanks, you are required to output the original Input sentences.

Business Category: {label_1} | Review Stars: {label_2}
Input: {masked_input}

Source: Differentially Private Synthetic Data via Foundation Model APIs 2: Text (Xie et al., ICML'24)

Google Research

Private evolution

1. Some prompt engineering required:
    - Seed corpus generation
    - Rewrite prompt should introduce appropriate amount of variation

2. Private data can be highly ephemeral or kept on-device

3. Private post-processing: the first 2 steps of PE (voting and filtering) can be used standalone to improve DP finetuning outputs.

Google Research

DP inference

Introduces DP during the **language model decoding process**, by:
(1) distributing private data into generation prompts in parallel; and
(2) mixing their output predictions.

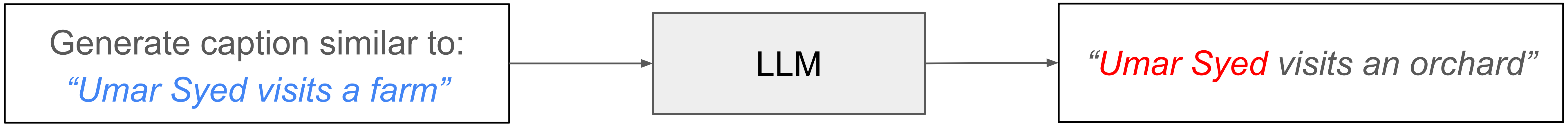
Privacy costs are paid per token generated.

Best for generating a small amount of examples quickly.

Google Research

DP inference

Idea: Prompt the LLM to generate synthetic data from real data.

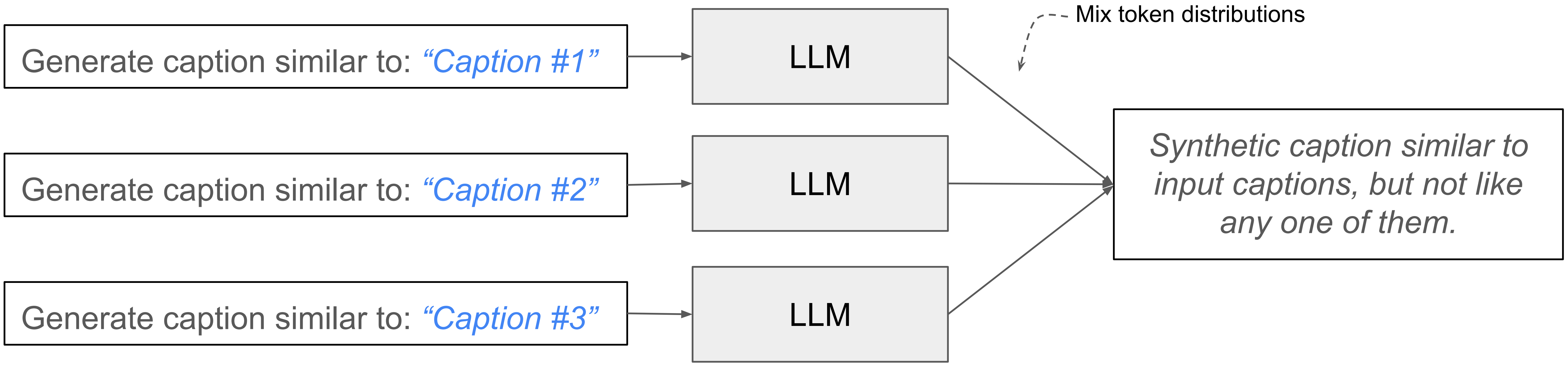

Generate caption similar to:
"Umar Syed visits a farm"
LLM
"Umar Syed visits an orchard"

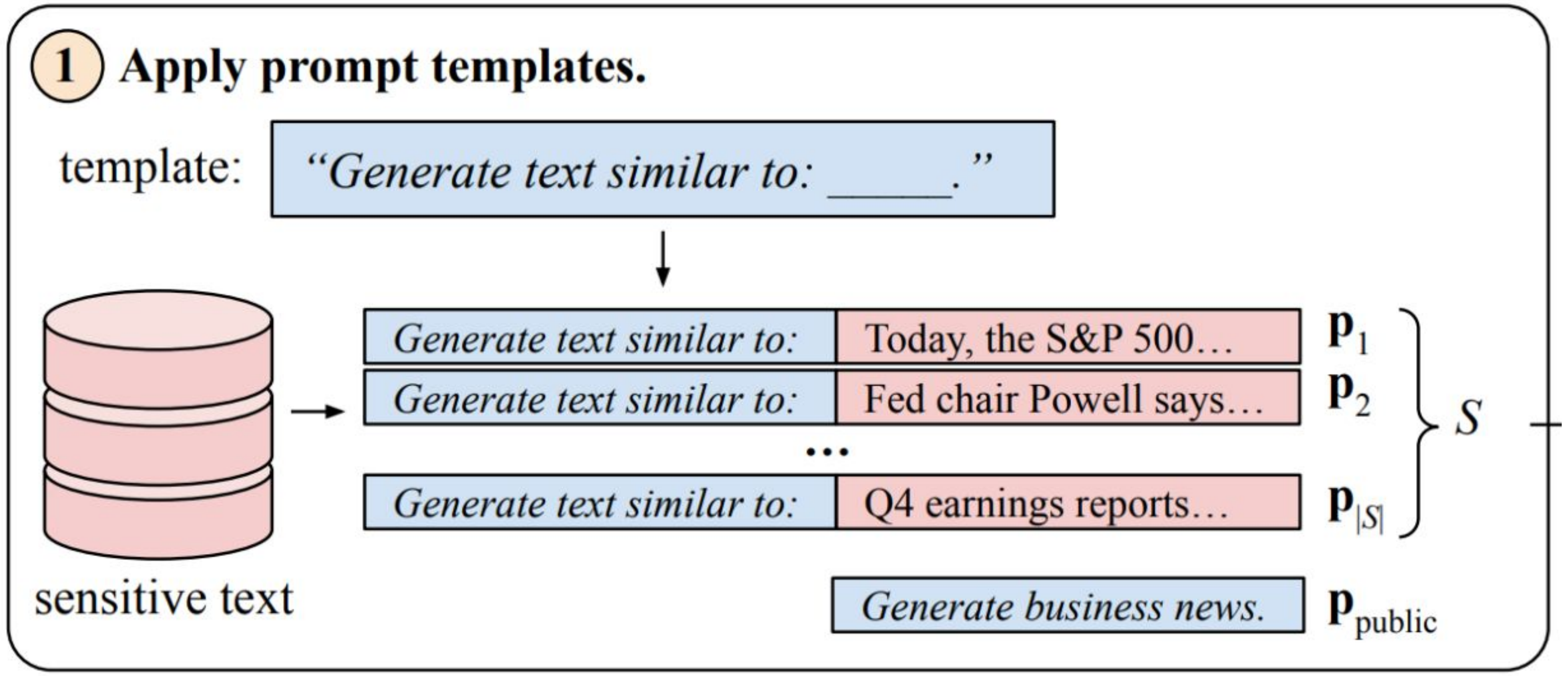
Problem: LLM can reveal private information.

DP inference

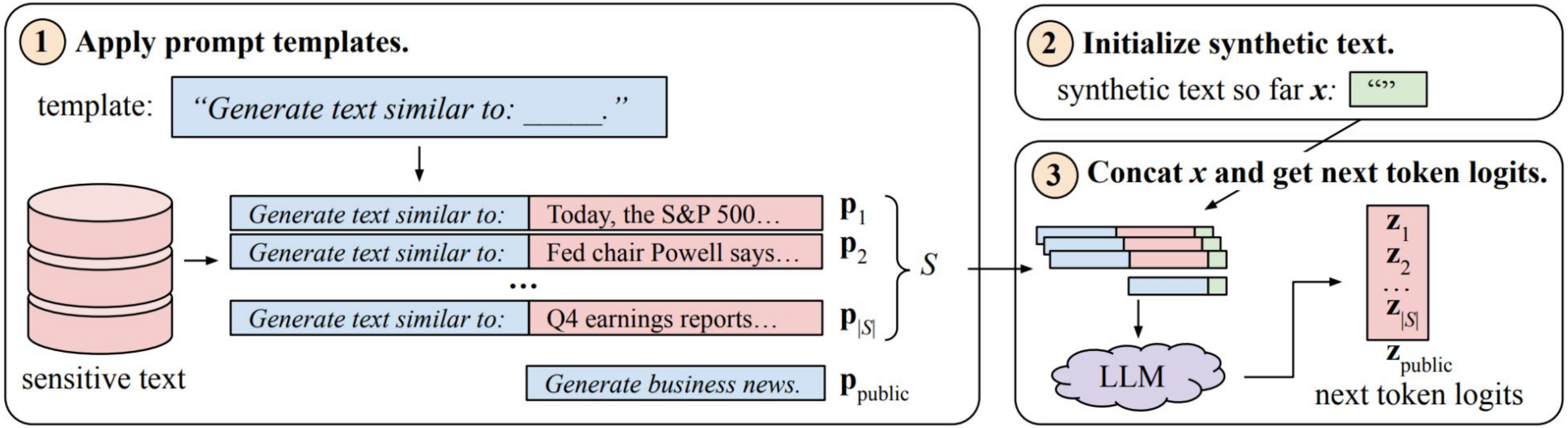
Observation: LLM responses are generated randomly, one token at a time.

Solution: Mix token distributions across parallel LLMs running on different inputs, which reduces dependence on any one input.

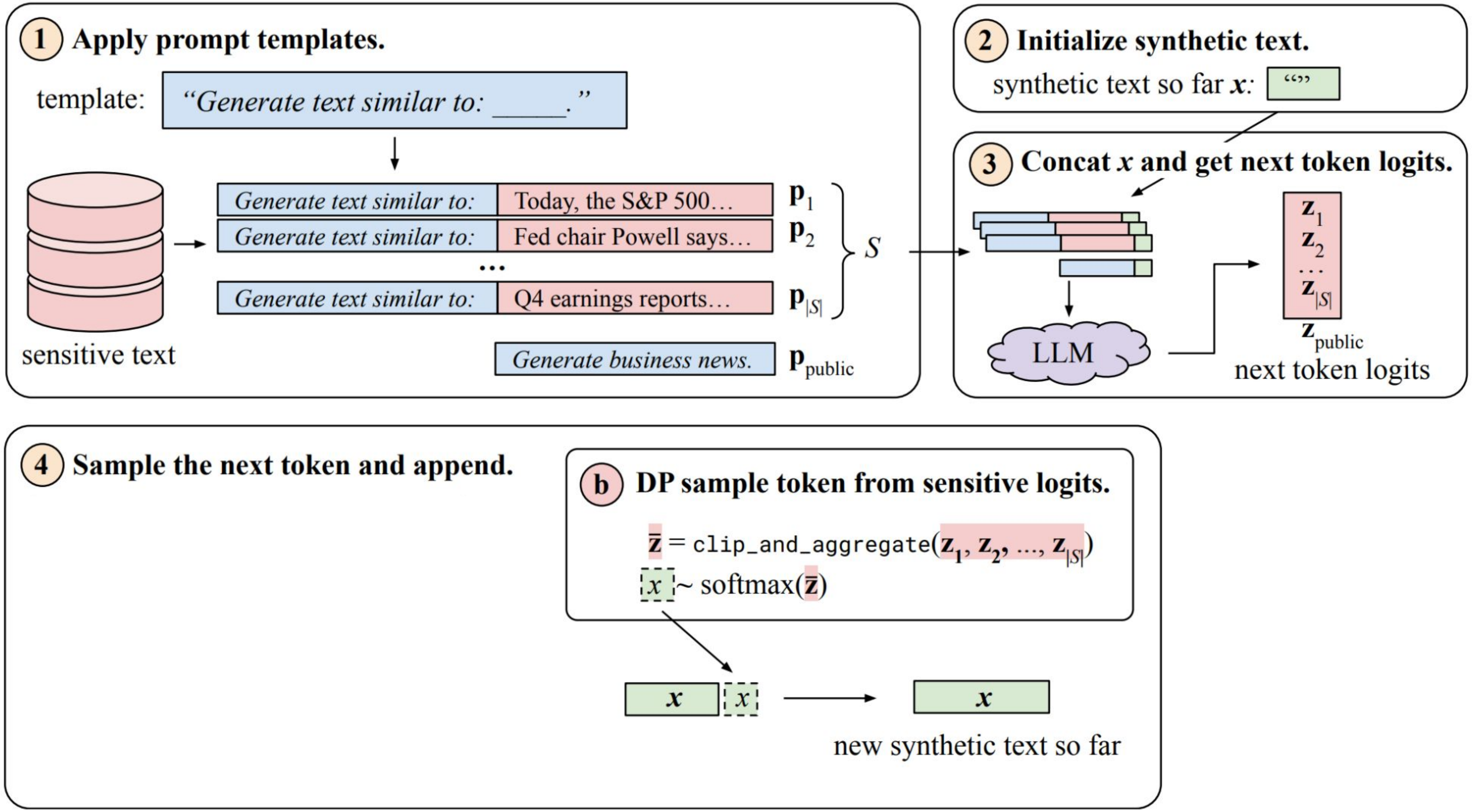

Generate caption similar to: *"Caption #1"*
Generate caption similar to: *"Caption #2"*
Generate caption similar to: *"Caption #3"*
LLM
LLM
LLM
Mix token distributions
Synthetic caption similar to input captions, but not like any one of them.


1
Apply prompt templates.
template:
"Generate text similar to: _____."
Generate text similar to: Today, the S&P 500…
Generate text similar to: Fed chair Powell says…
…
Generate text similar to: Q4 earnings reports…
$\mathbf{p}_1$
$\mathbf{p}_2$
$\mathbf{p}_{|S|}$
S
sensitive text
Generate business news.
$\mathbf{p}_{\text{public}}$

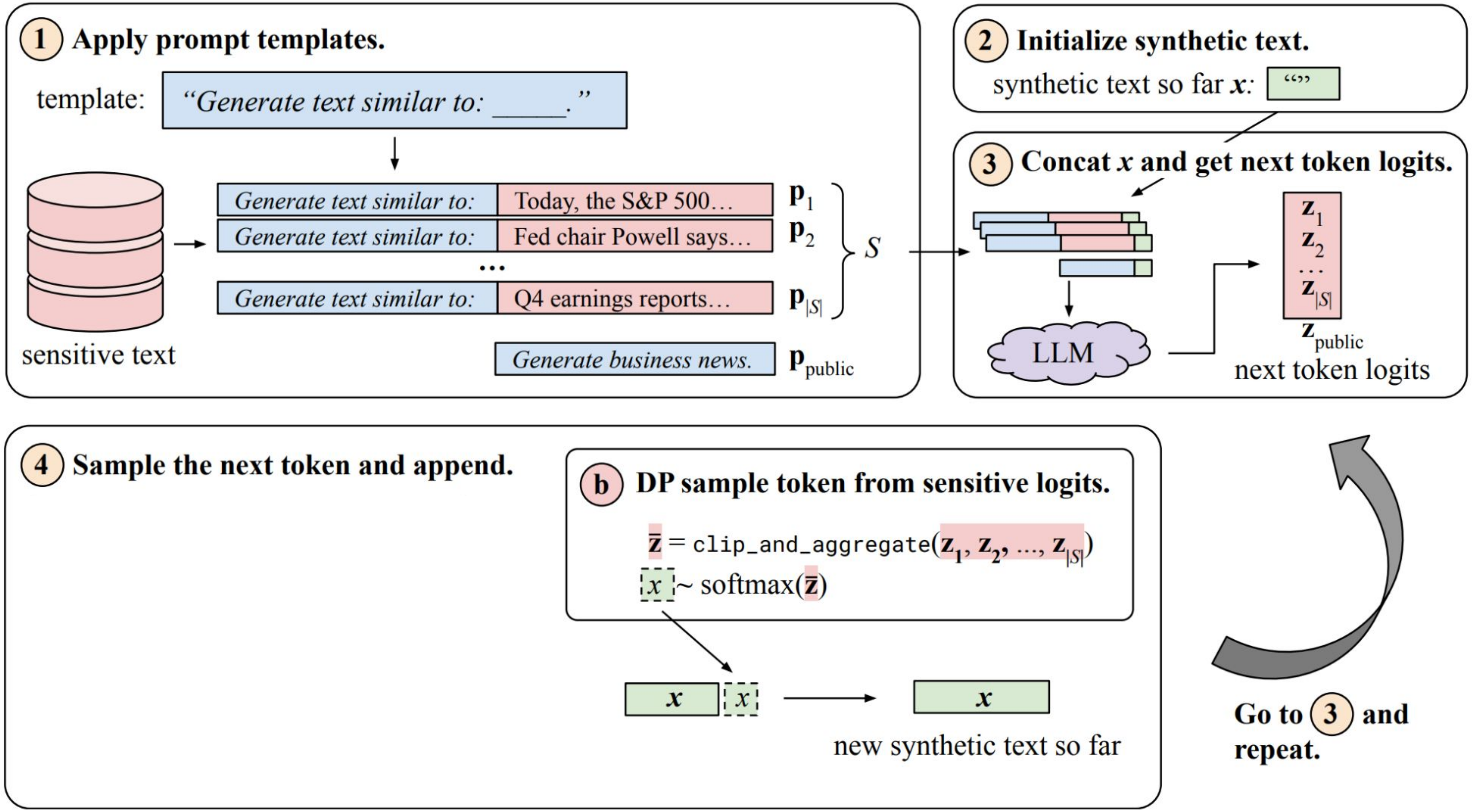
Google Research


1 Apply prompt templates.
template:
"Generate text similar to: _____."
sensitive text
Generate text similar to: Today, the S&P 500…
Generate text similar to: Fed chair Powell says…
Generate text similar to: Q4 earnings reports…
$\mathbf{p}_1$
$\mathbf{p}_2$
$\mathbf{p}_{|S|}$
S
Generate business news.
$\mathbf{p}_{\text{public}}$
2 Initialize synthetic text.
synthetic text so far $\boldsymbol{x}$: ""
3 Concat x and get next token logits.
LLM
$\mathbf{z}_1$
$\mathbf{z}_2$
$\mathbf{z}_{|S|}$
$\mathbf{z}_{\text{public}}$
next token logits

Google Research

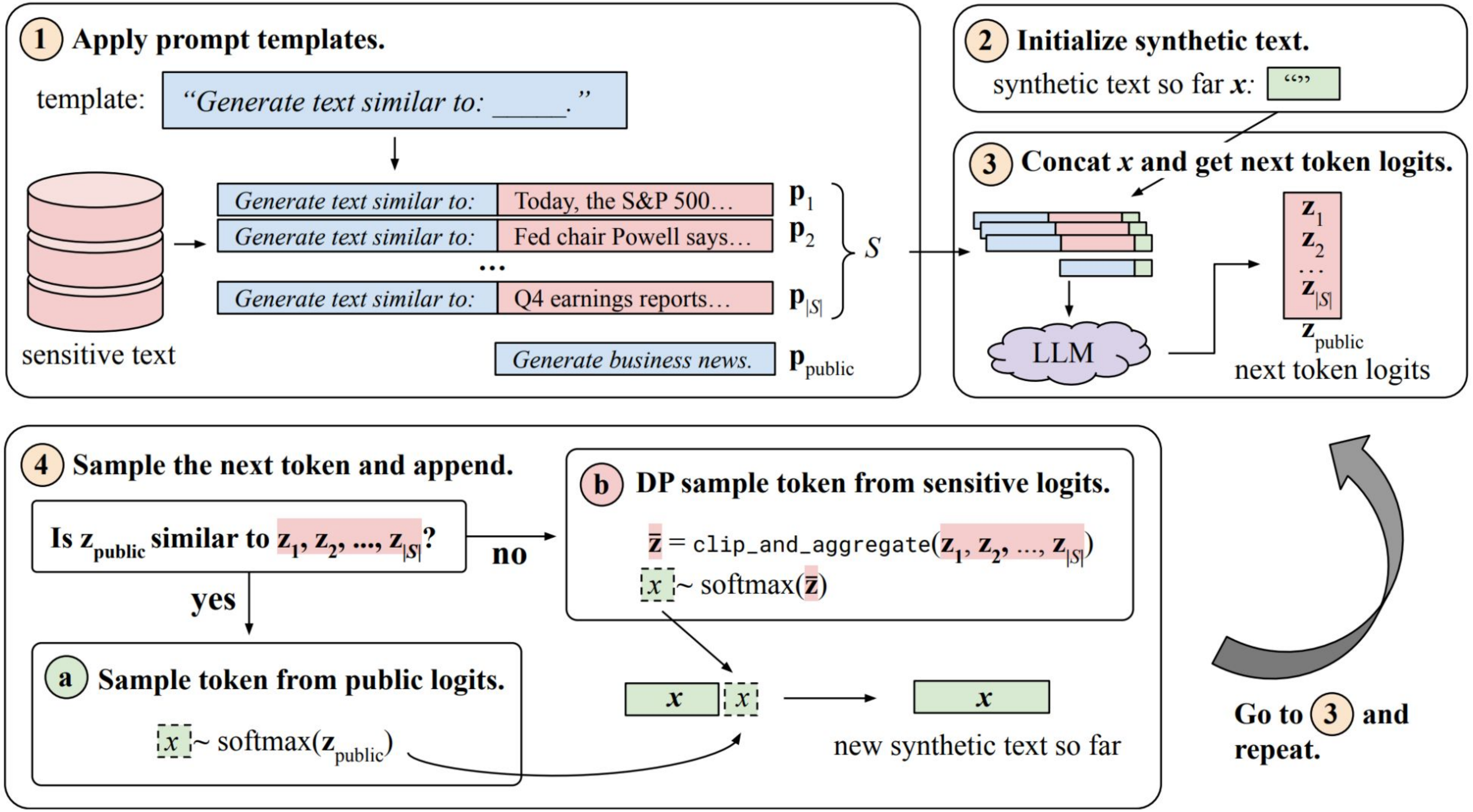


1 Apply prompt templates.
template:
"Generate text similar to: _____."
Generate text similar to:
Today, the S&P 500…
$\mathbf{p}_1$
Generate text similar to:
Fed chair Powell says…
$\mathbf{p}_2$
…
Generate text similar to:
Q4 earnings reports…
$\mathbf{p}_{|S|}$
S
sensitive text
Generate business news.
$\mathbf{p}_{\text{public}}$
2 Initialize synthetic text.
synthetic text so far $\boldsymbol{x}$: ""
3 Concat x and get next token logits.
LLM
$\mathbf{z}_1$
$\mathbf{z}_2$
…
$\mathbf{z}_{|S|}$
$\mathbf{z}_{\text{public}}$
next token logits
4 Sample the next token and append.
b DP sample token from sensitive logits.
$\bar{\mathbf{z}}$ = clip_and_aggregate($\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{|S|}$)
$x \sim \text{softmax}(\bar{\mathbf{z}})$
$\boldsymbol{x}$
x
$\boldsymbol{x}$
new synthetic text so far

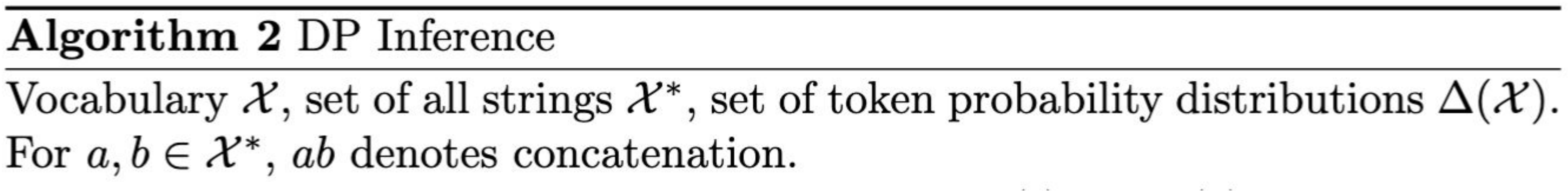


1 Apply prompt templates.
template:
"Generate text similar to: _____."
sensitive text
Generate text similar to: Today, the S&P 500…
Generate text similar to: Fed chair Powell says…
Generate text similar to: Q4 earnings reports…
$\mathbf{p}_1$
$\mathbf{p}_2$
$\mathbf{p}_{|S|}$
S
Generate business news.
$\mathbf{p}_{\text{public}}$
2 Initialize synthetic text.
synthetic text so far x: ""
3 Concat x and get next token logits.
LLM
$\mathbf{z}_1$
$\mathbf{z}_2$
$\mathbf{z}_{|S|}$
$\mathbf{z}_{\text{public}}$
next token logits
4 Sample the next token and append.
b DP sample token from sensitive logits.
$\bar{\mathbf{z}}$ = clip_and_aggregate($\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{|S|}$)
$x \sim \text{softmax}(\bar{\mathbf{z}})$
x
x
x
new synthetic text so far
Go to 3 and repeat.

Google Research


1 Apply prompt templates.
template: "Generate text similar to: ____."
sensitive text
Generate text similar to: Today, the S&P 500…
Generate text similar to: Fed chair Powell says…
Generate text similar to: Q4 earnings reports…
$\mathbf{p}_1$
$\mathbf{p}_2$
$\mathbf{p}_{|S|}$
S
Generate business news.
$\mathbf{p}_{\text{public}}$
2 Initialize synthetic text.
synthetic text so far x: ""
3 Concat x and get next token logits.
LLM
$\mathbf{z}_1$
$\mathbf{z}_2$
$\mathbf{z}_{|S|}$
$\mathbf{z}_{\text{public}}$
next token logits
4 Sample the next token and append.
Is $\mathbf{z}_{\text{public}}$ similar to $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_{|S|}$?
no
yes
b DP sample token from sensitive logits.
$\bar{\mathbf{z}}$ = clip_and_aggregate($\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_{|S|}$)
$x \sim \text{softmax}(\bar{\mathbf{z}})$
a Sample token from public logits.
$x \sim \text{softmax}(\mathbf{z}_{\text{public}})$
x x
x
new synthetic text so far
Go to 3 and repeat.

DP inference

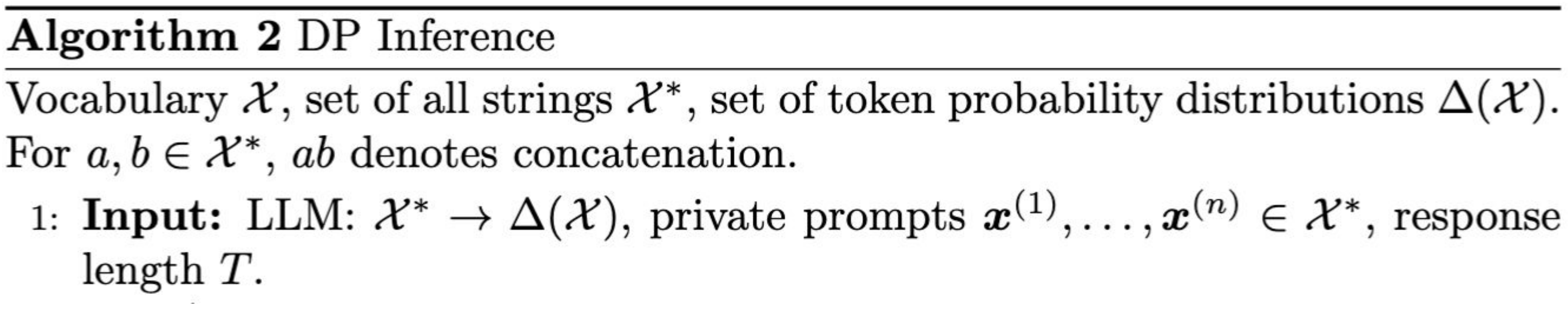


Algorithm 2 DP Inference

Vocabulary $\mathcal{X}$, set of all strings $\mathcal{X}^*$, set of token probability distributions $\Delta(\mathcal{X})$. For $a, b \in \mathcal{X}^*$, ab denotes concatenation.

Google Research

DP inference

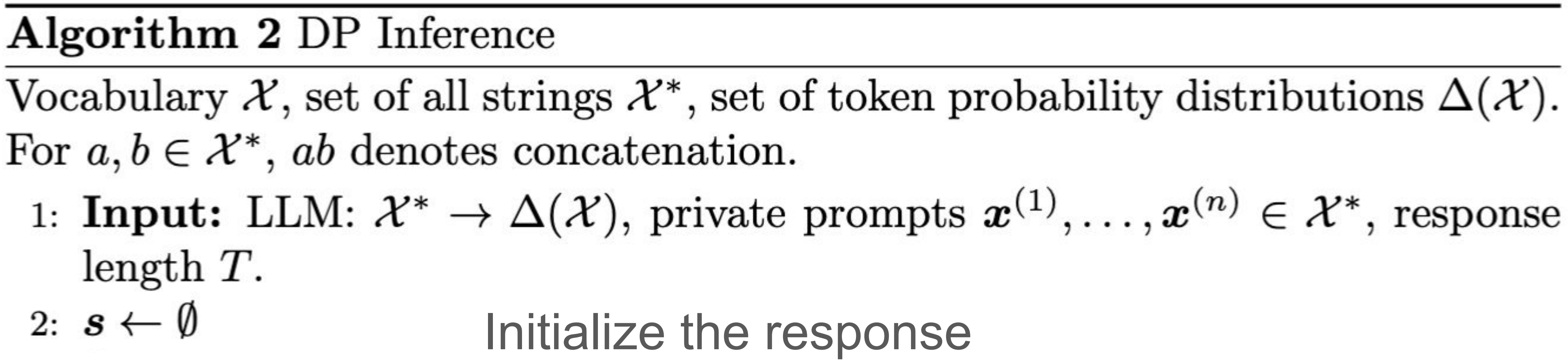


Algorithm 2 DP Inference

Vocabulary $\mathcal{X}$, set of all strings $\mathcal{X}^*$, set of token probability distributions $\Delta(\mathcal{X})$. For $a, b \in \mathcal{X}^*$, ab denotes concatenation.

```
1: Input: LLM: $\mathcal{X}^* \to \Delta(\mathcal{X})$, private prompts $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \in \mathcal{X}^*$, response
   length $T$.
```

DP inference



Algorithm 2 DP Inference

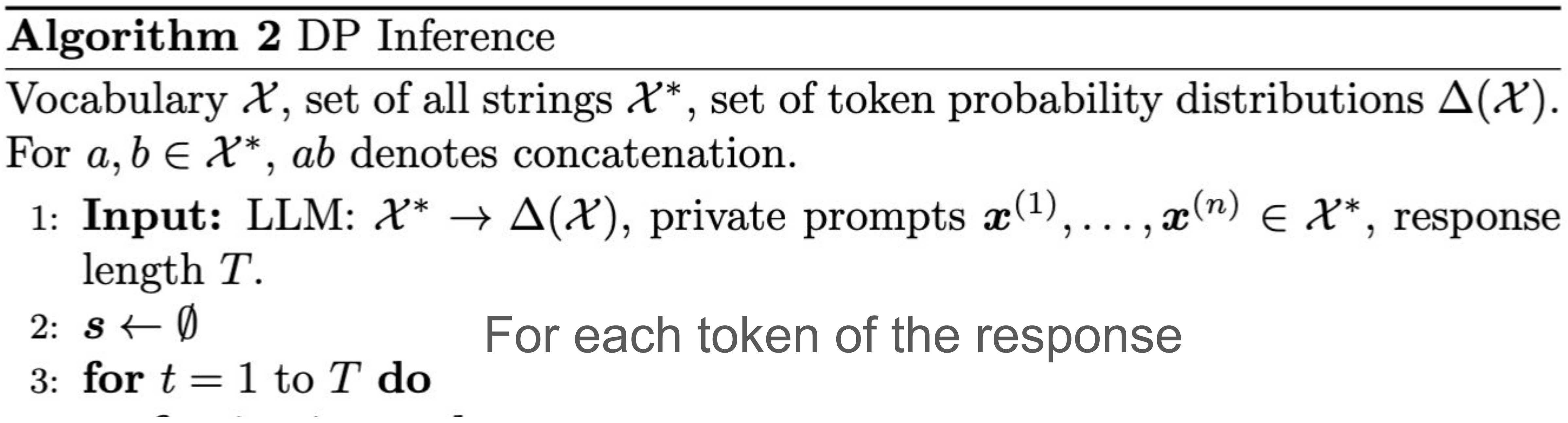
Vocabulary $\mathcal{X}$, set of all strings $\mathcal{X}^*$, set of token probability distributions $\Delta(\mathcal{X})$. For $a, b \in \mathcal{X}^*$, ab denotes concatenation.

```
1: Input: LLM: $\mathcal{X}^* \to \Delta(\mathcal{X})$, private prompts $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \in \mathcal{X}^*$, response length $T$.
2: $\boldsymbol{s} \leftarrow \emptyset$
```

Initialize the response

Google Research

DP inference



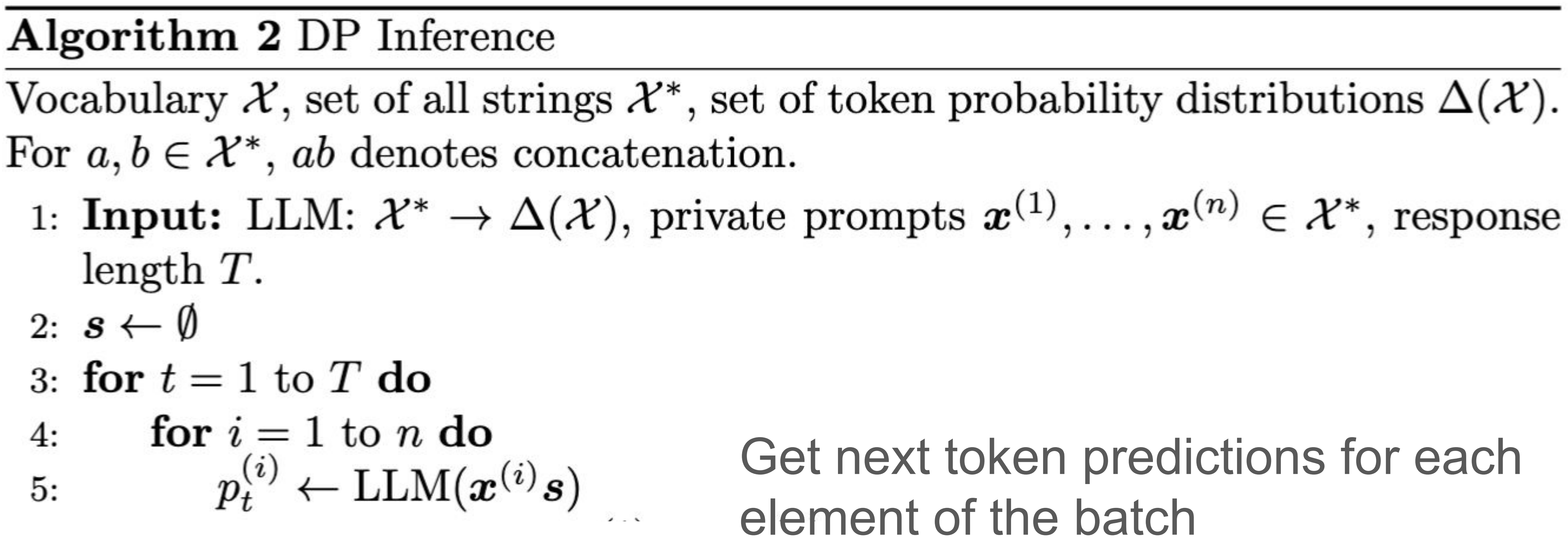
Algorithm 2 DP Inference

Vocabulary $\mathcal{X}$, set of all strings $\mathcal{X}^*$, set of token probability distributions $\Delta(\mathcal{X})$. For $a, b \in \mathcal{X}^*$, ab denotes concatenation.

```
1: Input: LLM: $\mathcal{X}^* \to \Delta(\mathcal{X})$, private prompts $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \in \mathcal{X}^*$, response
   length $T$.
2: $\boldsymbol{s} \leftarrow \emptyset$
3: for $t = 1$ to $T$ do
```

For each token of the response

DP inference



Algorithm 2 DP Inference

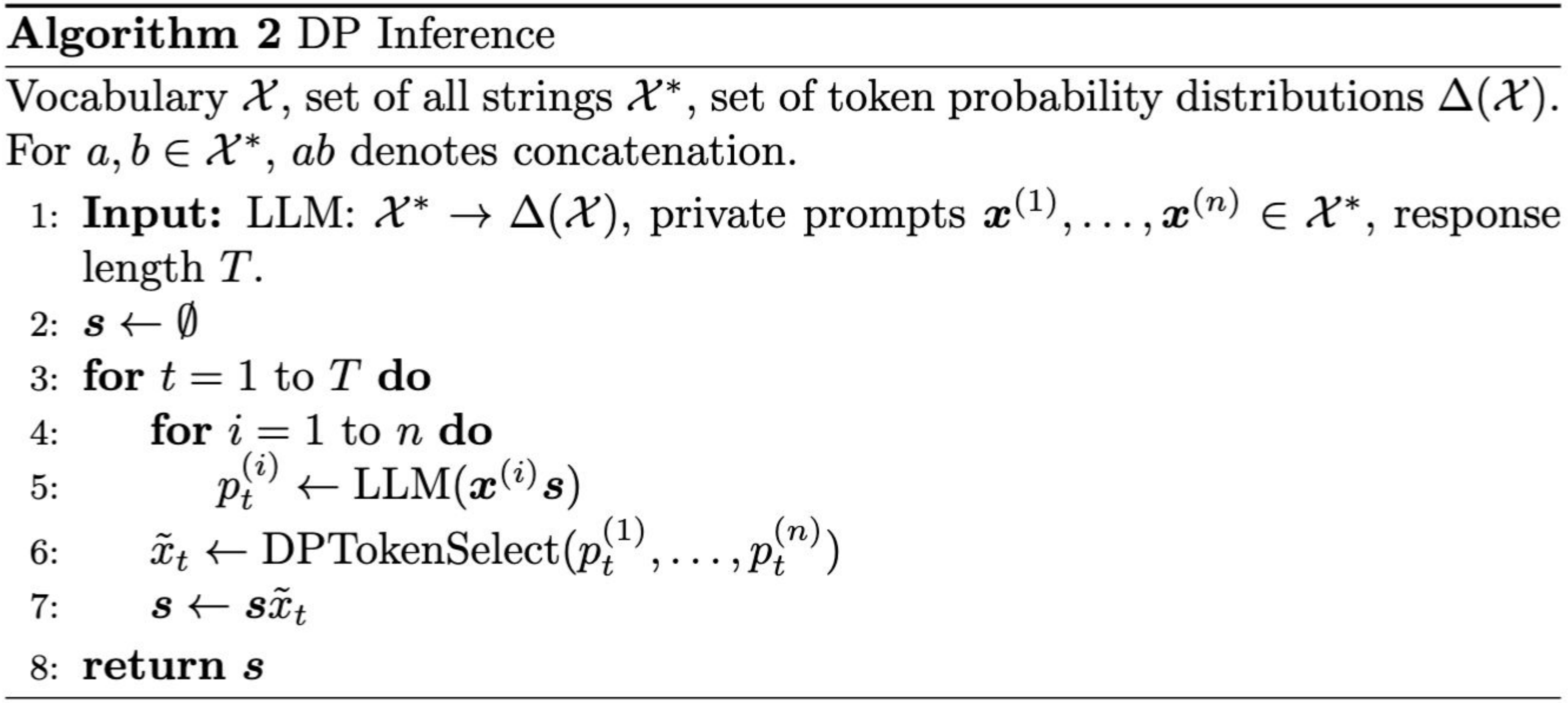
Vocabulary $\mathcal{X}$, set of all strings $\mathcal{X}^*$, set of token probability distributions $\Delta(\mathcal{X})$.
For $a, b \in \mathcal{X}^*$, ab denotes concatenation.

```
1: Input: LLM: $\mathcal{X}^* \to \Delta(\mathcal{X})$, private prompts $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \in \mathcal{X}^*$, response
   length $T$.
2: $\boldsymbol{s} \leftarrow \emptyset$
3: for $t = 1$ to $T$ do
4:     for $i = 1$ to $n$ do
5:         $p_t^{(i)} \leftarrow \text{LLM}(\boldsymbol{x}^{(i)}\boldsymbol{s})$
```

Get next token predictions for each element of the batch

Google Research

DP inference

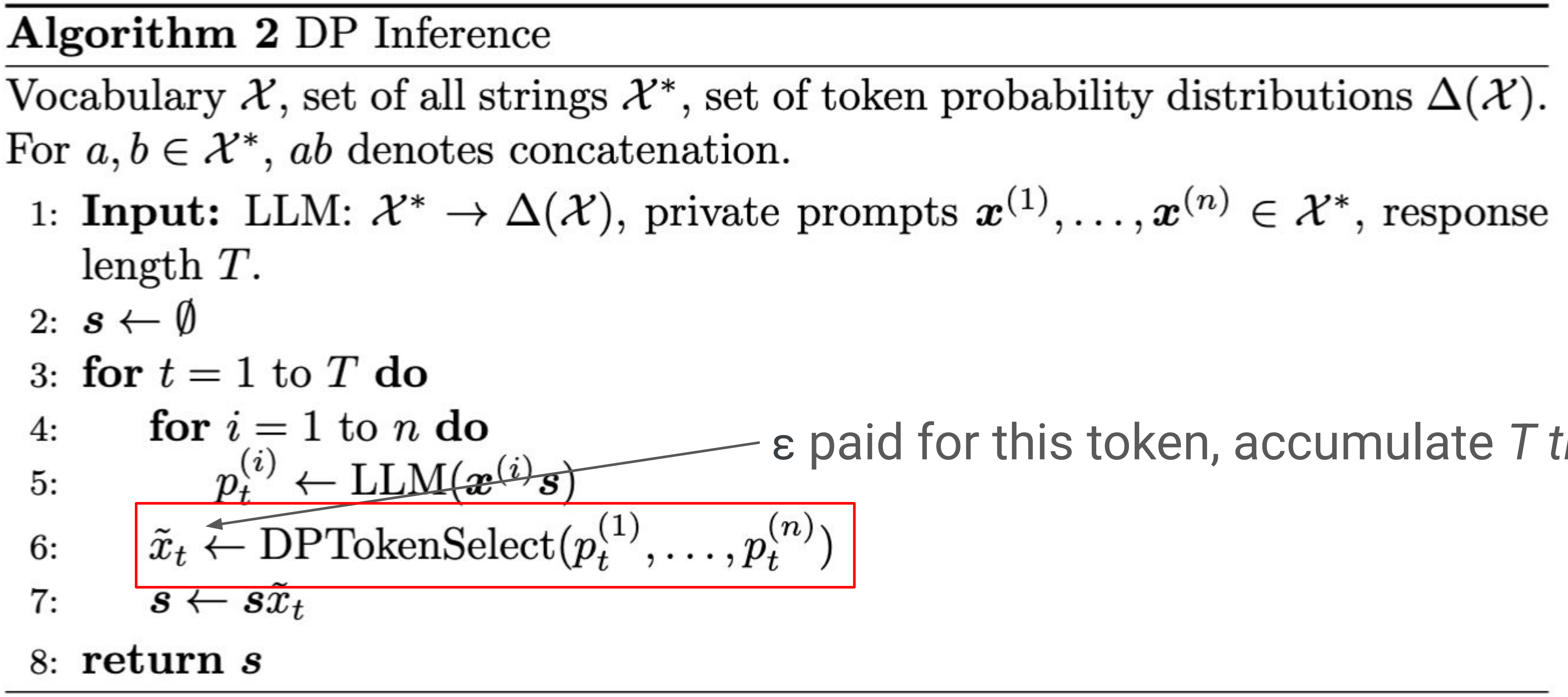


Algorithm 2 DP Inference

Vocabulary $\mathcal{X}$, set of all strings $\mathcal{X}^*$, set of token probability distributions $\Delta(\mathcal{X})$. For $a, b \in \mathcal{X}^*$, ab denotes concatenation.

```
1: Input: LLM: $\mathcal{X}^* \to \Delta(\mathcal{X})$, private prompts $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \in \mathcal{X}^*$, response
   length $T$.
2: $\boldsymbol{s} \leftarrow \emptyset$
3: for $t = 1$ to $T$ do
4:     for $i = 1$ to $n$ do
5:         $p_t^{(i)} \leftarrow \mathrm{LLM}(\boldsymbol{x}^{(i)}\boldsymbol{s})$
6:     $\tilde{x}_t \leftarrow \mathrm{DPTokenSelect}(p_t^{(1)}, \ldots, p_t^{(n)})$
7:     $\boldsymbol{s} \leftarrow \boldsymbol{s}\tilde{x}_t$
8: return $\boldsymbol{s}$
```

DP inference



Algorithm 2 DP Inference

Vocabulary $\mathcal{X}$, set of all strings $\mathcal{X}^*$, set of token probability distributions $\Delta(\mathcal{X})$. For $a, b \in \mathcal{X}^*$, ab denotes concatenation.

```
1: Input: LLM: $\mathcal{X}^* \to \Delta(\mathcal{X})$, private prompts $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \in \mathcal{X}^*$, response
   length $T$.
2: $\boldsymbol{s} \leftarrow \emptyset$
3: for $t = 1$ to $T$ do
4:     for $i = 1$ to $n$ do
5:         $p_t^{(i)} \leftarrow \mathrm{LLM}(\boldsymbol{x}^{(i)}\boldsymbol{s})$
6:     $\boxed{\tilde{x}_t \leftarrow \mathrm{DPTokenSelect}(p_t^{(1)}, \ldots, p_t^{(n)})}$
7:     $\boldsymbol{s} \leftarrow \boldsymbol{s}\tilde{x}_t$
8: return $\boldsymbol{s}$
```

ε paid for this token, accumulate *T times*

Google Research

Token aggregation in DP inference

Many possible choices for the token aggregation function:

Google Research

Token aggregation in DP inference

Many possible choices for the token aggregation function:

a. Average the probabilities and add noise (**Gaussian mechanism**)

Token aggregation in DP inference

Many possible choices for the token aggregation function:

a. Average the probabilities and add noise (**Gaussian mechanism**)
b. Each prompt returns top token, return noisy top choice (**PATE**)

Token aggregation in DP inference

Many possible choices for the token aggregation function:

a. Average the probabilities and add noise (**Gaussian mechanism**)
b. Each prompt returns top token, return noisy top choice (**PATE**)
c. Softmax sampling from averaged, clipped logits (**Exponential mechanism**)

Google Research

Token aggregation in DP inference

Many possible choices for the token aggregation function:

a. Average the probabilities and add noise (**Gaussian mechanism**)
b. Each prompt returns top token, return noisy top choice (**PATE**)
c. Softmax sampling from averaged, clipped logits (**Exponential mechanism**)

$$Z \leftarrow \{\text{logits}(\mathbf{px}) : \mathbf{p} \in S\}$$
$$\bar{\mathbf{z}} \leftarrow \tfrac{1}{s} \textstyle\sum_{\mathbf{z} \in Z} \text{clip}_c(\mathbf{z})$$
$$x \sim \text{softmax}(\bar{\mathbf{z}}/\tau)$$

$\text{clip}_c(\boldsymbol{z})_i = \max(-c, \min(c, z_i))$

Token aggregation in DP inference

Softmax sampling: Token i sampled w.p. $\sim \exp(\bar{z}_i/\tau)$
ε-DP exponential mechanism: i sampled w.p. $\sim \exp(\mathrm{score}(i, S) \cdot \varepsilon/2\Delta)$
$\Delta = \max_i \max_{S,S'} |\mathrm{score}(i, S) - \mathrm{score}(i, S')|$

$$Z \leftarrow \{\mathrm{logits}(\mathbf{px}) : \mathbf{p} \in S\}$$
$$\bar{\mathbf{z}} \leftarrow \frac{1}{s} \sum_{\mathbf{z} \in Z} \mathrm{clip}_c(\mathbf{z})$$
$$x \sim \mathrm{softmax}(\bar{\mathbf{z}}/\tau)$$

$\mathrm{clip}_c(\boldsymbol{z})_i = \max(-c, \min(c, z_i))$

Google Research

Token aggregation in DP inference

Softmax sampling: Token i sampled w.p. $\sim \exp(\bar{z}_i/\tau)$
ε-DP exponential mechanism: i sampled w.p. $\sim \exp(\mathrm{score}(i,S)\cdot\varepsilon/2\Delta)$
$\Delta = \max_i \max_{S,S'} |\mathrm{score}(i,S) - \mathrm{score}(i,S')|$

Let $\mathrm{score}(i,S) = \bar{z}i := [\frac{1}{s}\sum_{z\in S}\mathrm{clip}_c(z)]_i$
$\Delta = c/s$

$$\exp(\bar{z}_i/\tau) = \exp\left(\frac{\bar{z}_i}{2(c/s)}\cdot 2(c/s)\cdot\frac{1}{\tau}\right)$$
$$= \exp(\mathrm{score}(i,S)/2\Delta\cdot\frac{2c}{s\tau})$$

$$Z \leftarrow \{\mathrm{logits}(\mathbf{px}) : \mathbf{p}\in S\}$$
$$\bar{\mathbf{z}} \leftarrow \tfrac{1}{s}\textstyle\sum_{\mathbf{z}\in Z}\mathrm{clip}_c(\mathbf{z})$$
$$x \sim \mathrm{softmax}(\bar{\mathbf{z}}/\tau)$$

$\implies 2c/s\tau$-DP
$c = 10,\ \tau = 2,\ s = 100 \implies \varepsilon = 0.1$

$\mathrm{clip}_c(\boldsymbol{z})_i = \max(-c, \min(c, z_i))$

Google Research

Unified view of methods

Start from a non-private approach to generate synthetic data

DPify an **iterative primitive** used to **extract information from real data**

	DP finetuning	**Private evolution**	**DP inference**
Comparison	Finetuning	Evolutionary prompting	Inference
Primitive	Gradient	Selection	Token sampling

- Amenability to DP: primitive is an aggregate function with bounded sensitivity
- Privacy guarantee of primitive => privacy guarantee of iterations

Google Research

Rough comparison of methods

	DP finetuning	Private evolution	DP inference
Recommended when your input size is …	>10K	<5k	Not recommended
Yield	Infinite	<= input data size	1k ~> 25 examples
Access required	Weights	API	Logits
Training required?	Yes	No	No
Prompt engineering	No	Yes	No
Inference cost	1	Batch size	rounds x rewrites
OOD Adaptability	High	Low	Medium
Data persistence	Entire dataset required for training.	Ephemeral/stays on device (votes only)	One batch at a time

Google Research

Summary of comparison of methods

DP finetuning is the workhorse method that delivers the best quality given sufficient data, compute, model access, and engineering effort.

Since the adoption of DP synthetic data is often quality-bottlenecked, this is likely the option that best fits your needs.

Google Research

Summary of comparison of methods

DP finetuning is the workhorse method that delivers the best quality given sufficient data, compute, model access, and engineering effort.

Since the adoption of DP synthetic data is often quality-bottlenecked, this is likely the option that best fits your needs.

Private evolution is most useful when the stringent ($\varepsilon<1$) required or small data.

Summary of comparison of methods

DP finetuning is the workhorse method that delivers the best quality given sufficient data, compute, model access, and engineering effort.

Since the adoption of DP synthetic data is often quality-bottlenecked, this is likely the option that best fits your needs.

Private evolution is most useful when the stringent ($\varepsilon<1$) required or small data.

DP inference is useful for generating a small amount of examples quickly.

Google Research

Utility Metrics

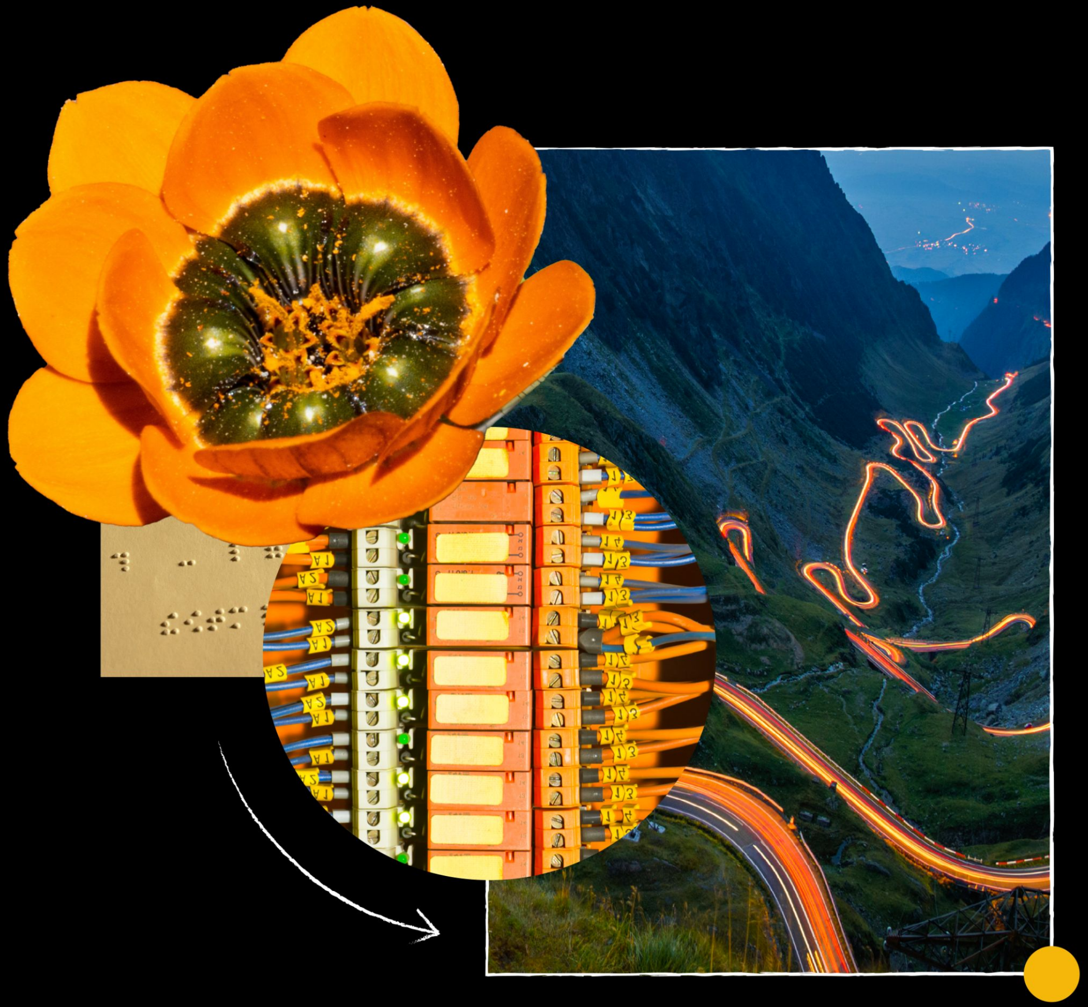
The best utility metric is downstream task performance, since that is why you are generating synthetic data to begin with.

There are proxies that can be helpful:

- Heuristics: n-gram statistics, length distribution
- MAUVE score measures distributional similarity
    - Real and synthetic dataset are embedded and clustered together. MAUVE is high when clusters contain a mix of real and synthetic.
    - Correlates with downstream task performance.

Google Research

DP synthetic images



Google Research

Image data...

- Image data is ubiquitous
- Image data has large body of work on image generation (starting from GANs)
- Image data is very hard to synthesize properly
- DP image synthesis brings it to the next level!
    - Images are high dimensional. Capturing the intricate distribution of natural images requires complex models. DP on more complex models results in more noise
    - There is a "continuity" of images in many directions (not just 1 as in text!)
    - Computation cost of DP is going to be high for large models
        - And models like GANs (adversarial training) are harder to train with DP
    - Utility evaluation difficulties: metrics like FID don't always correlate well with downstream performance

"Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks" Alec Radford, Luke Metz, Soumith Chintala, 2015

Diffusion Models Beat GANs on Image Synthesis, Prafulla Dhariwal, Alex Nichol, 2021

What is appropriate privacy unit for image data

- Most academic works treat each image as a privacy unit (example-level privacy unit)
    - In this setting, DP synthetic data will loosely guarantee that it won't be very different whether a particular image was in your private dataset or not
- In real world applications, it is perhaps not that needs to be done
    - Most of the time you will want at least user-level privacy
        - My synthetic data is not significantly different, whether a particular user contributed their data to my private dataset or not
    - The definition of a "user" is however application dependent
        - User can be a creator of an image
        - Users can be people present on the photos
- Bottom line:
    - For each image define one or more "users" associated with the image
    - Do appropriate user contribution bounding (e.g. 1 image per user)
    - Proceed as if you had a example level privacy unit

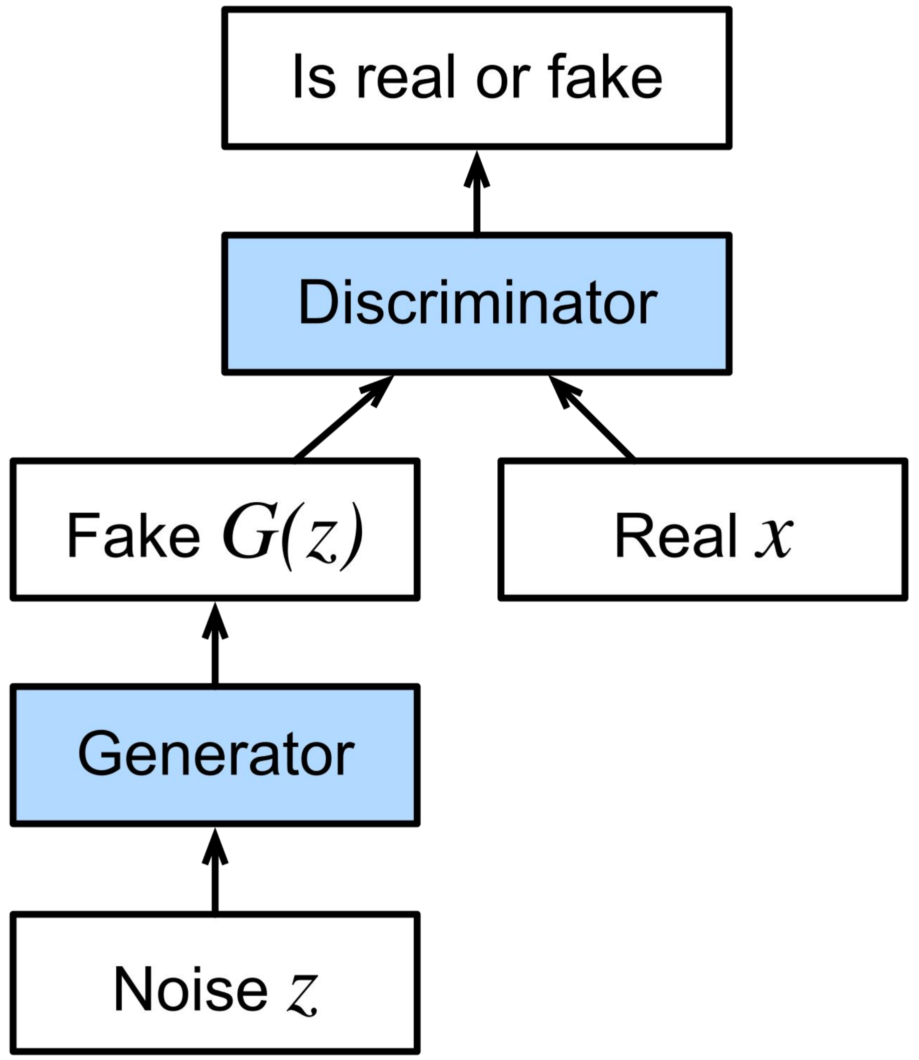
Google Research

What are the methods for DP image synthesis

- DP training (from scratch) or DP-Finetuning
  - GANs, VAEs
  - Multimodal LLMs, Diffusion
- Inference only/API based methods
  - PATE-style
  - Private evolution
- Data release mechanisms have not been successful so far
  - They attempt to calculate some statistics or intermediate representations of the data, dp-fy that and generate the images from noised representation
  - Examples include DPGEN (Chen et al, 2022b) that was shown to be not proper DP
  - DP-DRE (Wu et al, 2023) which uses publicly pretrained encoder and ICGAN generator. Private data is embedded via an encoder and the distribution is DP-fied via DP density estimator. Samples are then drawn from this distribution and decoded via ICGAN

DP Training/finetuning for Image generation: GANs

- GAN models
  - First somewhat successful models in image synthesis
  - To obtain DP version of GANs, DP-SGD (and likes) is used to update the discriminator parameters (DPGAN Differentially Private Generative Adversarial Network)
- DP-Training of GANs is hard
  - GANs are notoriously hard to train (diverge, require early stopping)
  - Early DP GANs struggled to generate good images below low resolution like 32x32



Source: https://en.wikipedia.org/wiki/Generative_adversarial_network#/media/File:Generative_adversarial_network.svg

DP Training/finetuning for Image generation: GANs

- Several attempts at improving DP-GANs
  - DP-finetuning of publicly pre-trained GANs
    - DP-GAN-MI (Chen et al 2022) uses publicly pre-trained feature extractor
    - DP-DRE (Wu et al) uses publicly trained encoder (and also does DP-SGD on GAN)
  - Alternative losses
    - DP-Sinkhorn (Cao et al 2021) uses optimal transport based loss (Sinkhorn divergence, measuring the distinct between real and synthetic distributions)
    - GS-WGAN (Chen et al 2020) uses alternative loss (Wassertein-1 loss) that generated bounded gradients with norms near 1. This allows to assume clipping norm of 1 and forego hyperparameter search

Google Research

DP Training/finetuning for Image generation: Diffusion

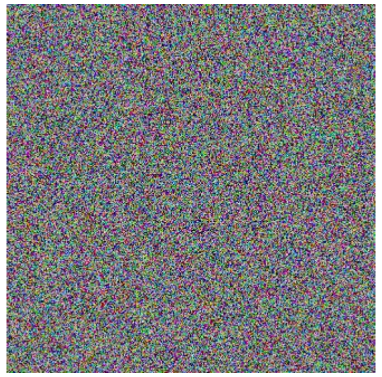
- Diffusion models are state of the art for high fidelity image generation
- Brief intro from (dp-promise: Differentially Private Diffusion Probabilistic Models for Image Synthesis)
  - Forward diffusion process (multi step): gradually adding noise to the data
  - Markov chain {x0, x1, .. xT}

$$q(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(t-1)}) = \mathcal{N}(\boldsymbol{x}^{(t)}; \sqrt{1-\beta_t}\boldsymbol{x}^{(t-1)}, \beta_t \boldsymbol{I}), \qquad (1)$$

$$q(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(T)}|\boldsymbol{x}^{(0)}) = \prod_{t=1}^{T} q(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(t-1)}). \qquad (2)$$

$$\alpha_t = \prod_{s=1}^{t}(1-\beta_s).$$

$$\boldsymbol{x}^{(t)} = \sqrt{\alpha_t}\boldsymbol{x}^{(0)} + \sqrt{1-\alpha_t}\boldsymbol{z},$$
$$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}).$$



  - Reverse sampling (multi step, Markov chain): learning to reverse this process via a model, starting from noised images produced in the forward pass (DP-LDMs, Liu et al 2023) to produce the original image
    - Predict the noise distribution from t to t-1, recover x_t-1 by removing the noise
    - Model Z_theta learns to predict the noise injected from 0 to t =>derive the mean of the noise
    - Objective to learn Theta uses samples from iterative forward process

$$\arg\min_{\theta} \mathbb{E}_{t,\boldsymbol{x}^{(0)},\boldsymbol{z}}\left[\|\boldsymbol{z} - \boldsymbol{z}_\theta(\sqrt{\alpha_t}\boldsymbol{x}^{(0)} + \sqrt{1-\alpha_t}\boldsymbol{z}, t)\|_2^2\right],$$

$$\hat{\boldsymbol{x}}^{(t-1)} = \sqrt{\alpha_{t-1}}\frac{\hat{\boldsymbol{x}}^{(t)} - \sqrt{1-\alpha_t}\boldsymbol{z}_\theta(\hat{\boldsymbol{x}}^{(t)}, t)}{\sqrt{\alpha_t}} + \sqrt{1-\alpha_{t-1}-\hat{\Sigma}_t^2}\cdot \boldsymbol{z}_\theta(\hat{\boldsymbol{x}}^{(t)}, t) + \hat{\Sigma}_t \boldsymbol{\xi},$$

Google Research

DP Training/finetuning for Image generation: Diffusion

- DPDM (Dockhorn et al 2023) - DP SGD applied to Diffusion
- DP-LDM (Liu et al 2024)
  - Prompt or label conditioned
  - Train an encoder-decoder on public data
  - Use the encoder to reduce the dimensionality on private data
  - Train a diffusion model with DP-SGD on lower dimensionality representation
    - Key is reduction in DP-finetunable parameters : authors fiinetune only attention modules
  - Decode back
  - Able to generate 256x256 images

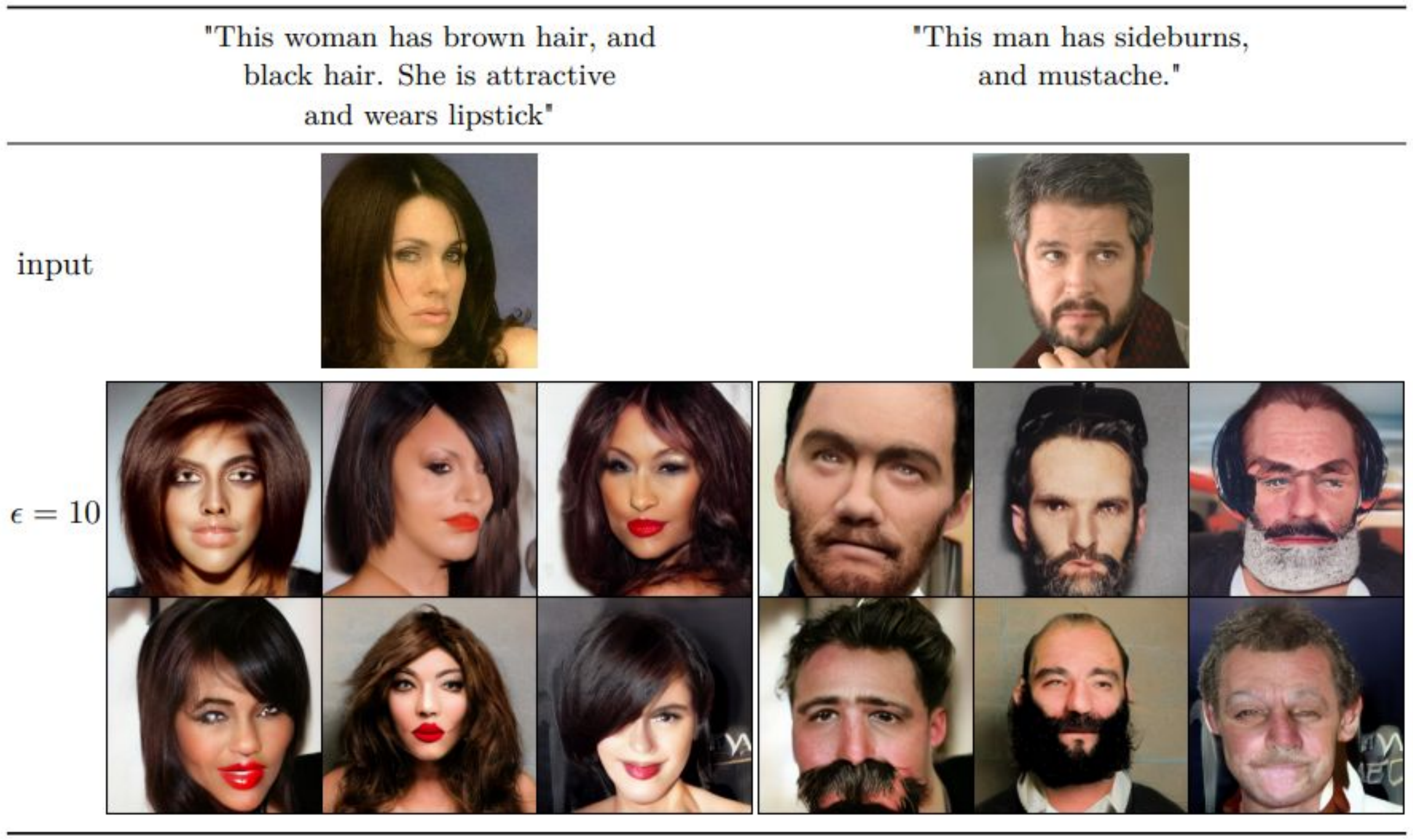

"This woman has brown hair, and black hair. She is attractive and wears lipstick"
"This man has sideburns, and mustache."
input
$\epsilon = 10$

Figure 3: Text-to-image generation of 256 × 256 CelebAHQ with prompts at $\epsilon = 10$. FID: 15.6

DP-LDMs: Differentially Private Latent Diffusion Models Michael F. Liu, Saiyue Lyu, Margarita Vinaroz, Mijung Park

DP Training/finetuning for Image generation: Diffusion

- Common themes are similar to DP-training of LLMs for text
  - Reducing the number of parameters finetuned with DP
    - Some also reduce the input dimensionality (Liu et al 2024 DP-LDM)
  - Using publicly pretrained Diffusion models (e.g. ImageNet pretrained
    - Privimage (Li et al 2024) subselects the public data closest to the private data via clip embeddings for pretraining
    - DPRandP (Tang et al 2023) pretrain on synthetically generated random data
    - Park et al (2024) use small related public dataset, boost it with Diffusion models, and then pretrain on the combined set
  - Leveraging inherent Diffusion noise
    - Wang et al (dp-promise) argues that Diffusion models already add noise during forward process
    - This noise should be used to give DP guarantee (instead of introducing additional noise via DP-SGD)

Google Research

DP Training/finetuning for Image generation: Diffusion

- DP-Promise leverages the fact that we already introduce noise during the forward stage of diffusion which contributes to privacy.
- Gradients calculated on noise injected images from Diffusion forward pass are noisy already having similar effect as DP-SGD => we can use less noise during DP-SGD
- Intuition: Earlier interactions of the forward are less private (less noise has been added), as we get to T iterations, it is mostly all noise
- Introduce two stage process for Reverse sampling
  - Phase 1: non dp training (DP from forward noise) [S, T] (sensitivity is by 2sqrt(d), d is channel x height x width)
  - Phase 2: DP-Training steps [1, S-1] (sensitivity is bound with clipping)

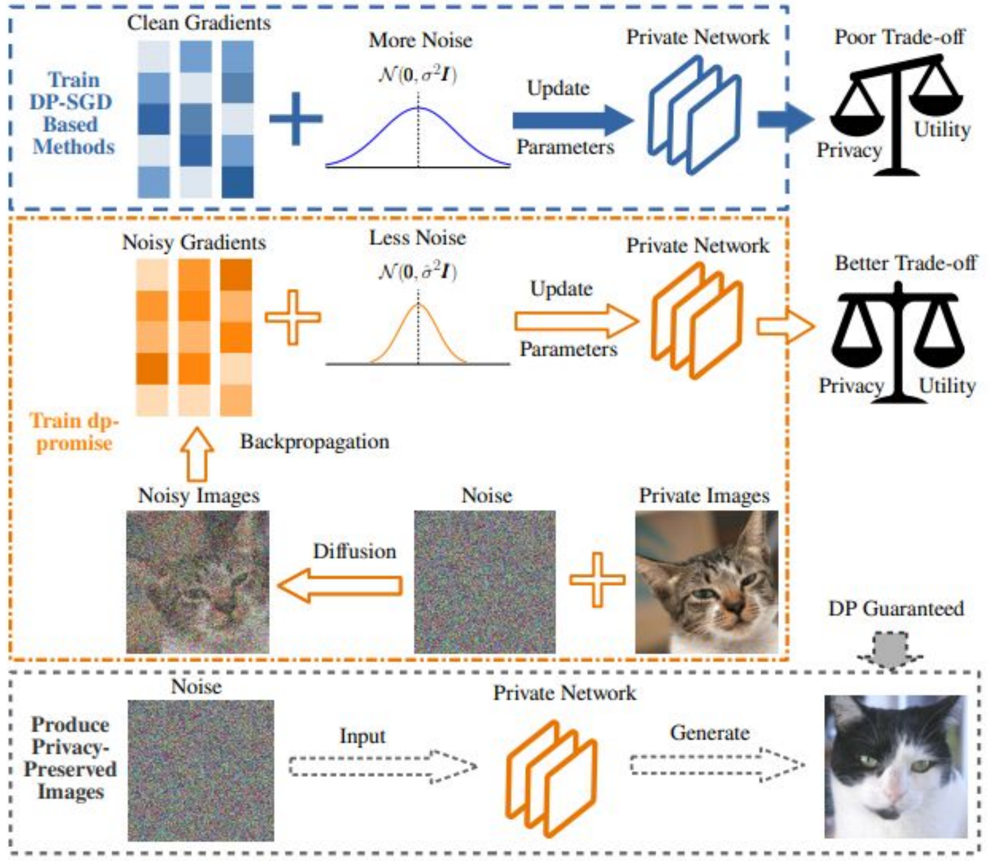


Figure 1: Comparison between dp-promise and other DP-SGD-based approaches for differentially private image synthesis using DMs.

dp-promise: Differentially Private Diffusion Probabilistic Models for Image Synthesis, Wang et al, 2024
https://www.usenix.org/system/files/usenixsecurity24_slides-wang-haichen.pdf

DP Training/finetuning for Image generation: Diffusion

- DP-Promise

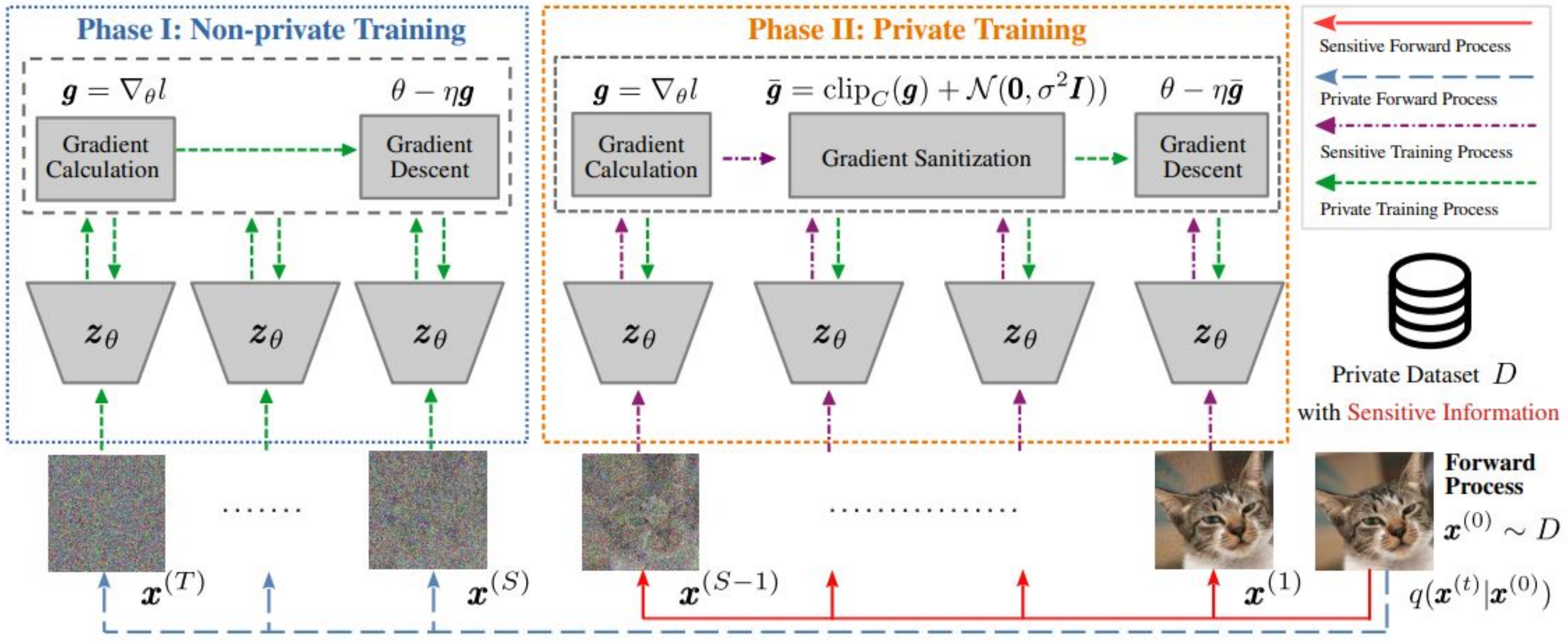


Figure 2: Framework of dp-promise, which aims to provide privacy guarantees to private data during the training of DMs. "Sensitive" means there is a risk of privacy leakage, and "Private" means privacy guarantees exist.

dp-promise: Differentially Private Diffusion Probabilistic Models for Image Synthesis, Wang et al, 2024
https://www.usenix.org/system/files/usenixsecurity24_slides-wang-haichen.pdf

Google Research

DP Training/finetuning for Image generation: Diffusion

- DP-Promise

dp-promise: Differentially Private Diffusion Probabilistic Models for Image Synthesis, Wang et al, 2024
https://www.usenix.org/system/files/usenixsecurity24_slides-wang-haichen.pdf

Google Research

Inference only methods

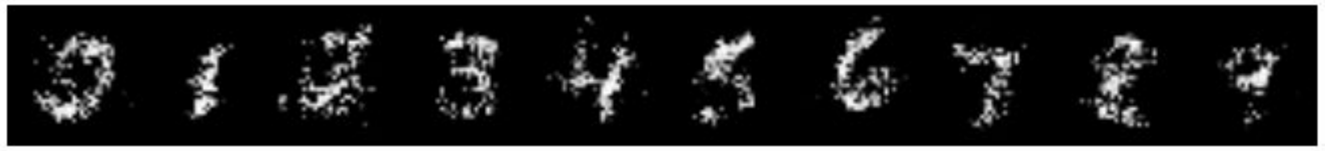
- DP-Training/finetuning is computationally expensive AND requires access to model checkpoint weights
- Attempts at introducing DP at the inference only have been made for Image modality
  - PATE style models (Private Aggregations of Teacher Ensembles) (Papernot et al, 2017)
    - Split the training data into a number of disjoint datasets
    - Train a model (or finetune a model) on each of them
    - Introduce the noise for DP guarantee to their aggregated prediction, level of noise depends on the level of agreement between the models
  - PATE-GAN (Yoon et al, 2019)
    - PATE part is applied to discriminator
    - Private data is partitioned
    - A number of discriminator models are trained
    - A student discriminator model is trained using teacher's DP-fied predictions (on some public data or data generated by the generator itself)
    - Generator tries to fool the student, student tries to improve its loss wrt teachers and teachers are trained to improve their loss wrt generator

Google Research

Inference only methods

- G-PATE (Long et al, 2019)
  - Don't need the student
  - As long as the gradients of the discriminator are DP, we are good
  - A gradient aggregator gets gradients from the teachers and accumulates them in PATE-style before passing DP gradient to the generator for update
  - Better quality images for small budgets (eps <=1)
- PATE-TripleGaN (Jiang et al , 2024) is probably one of the most modern twists on PATE idea

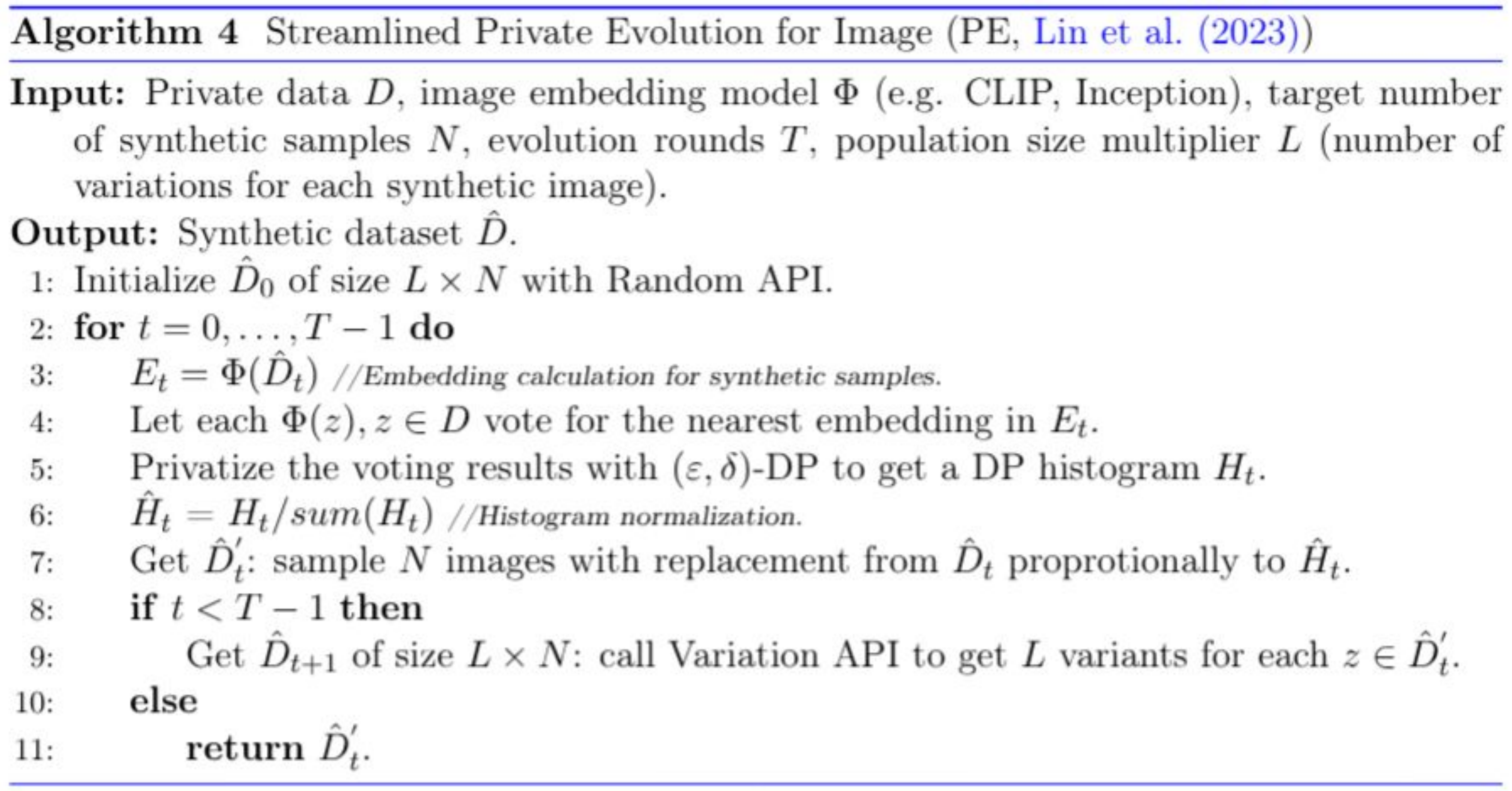
The bottom line: The quality is still nowhere near the Diffusion style models (eps 10)



PATE-TripleGAN: Privacy-Preserving Image Synthesis with Gaussian Differential Privacy Jiang Zepng, 2024

Inference only methods

- Private evolution
  - Use inference only access to generative models
  - Needs a good embedding (e.g. CLIP, Inception Network etc)
  - Variation API and Random API (usually already available in models like DALL-E, Stable Diffusion or can be implemented with appropriate prompt engineering (e.g. GPT models). APIs can be conditioned with prompts



Algorithm 4 Streamlined Private Evolution for Image (PE, Lin et al. (2023))

Input: Private data D, image embedding model Φ (e.g. CLIP, Inception), target number of synthetic samples N, evolution rounds T, population size multiplier L (number of variations for each synthetic image).

Output: Synthetic dataset $\hat{D}$.

```
1: Initialize $\hat{D}_0$ of size $L \times N$ with Random API.
2: for $t = 0, \dots, T-1$ do
3:     $E_t = \Phi(\hat{D}_t)$ //Embedding calculation for synthetic samples.
4:     Let each $\Phi(z), z \in D$ vote for the nearest embedding in $E_t$.
5:     Privatize the voting results with $(\varepsilon, \delta)$-DP to get a DP histogram $H_t$.
6:     $\hat{H}_t = H_t / sum(H_t)$ //Histogram normalization.
7:     Get $\hat{D}'_t$: sample $N$ images with replacement from $\hat{D}_t$ proprotionally to $\hat{H}_t$.
8:     if $t < T-1$ then
9:         Get $\hat{D}_{t+1}$ of size $L \times N$: call Variation API to get $L$ variants for each $z \in \hat{D}'_t$.
10:    else
11:        return $\hat{D}'_t$.
```

Inference only methods

- Private evolution for images
  - Some secret sauce includes "look ahead" distance (distance for computing real to synthetic is calculated via getting variants of synthetic and calculating average distance to those variants)
  - For user level privacy unit - normalize the vote for the samples for the same user to be 1 (in l2 norm)
- Beauty of PE is
  - Computationally cheaper (still needs a lot of inferences, but does not do DP training)
  - Quality will be fantastic (fidelity? Perhaps not)
  - Private data is never used for finetuning (no risks of outputting it back at all)
  - The only method that can work with small amount of private data
  - Cheap from DP perspective (can allow small epsilons)
  - Is a general framework (Foundational models can be replaced with Synthesizers (e.g. SIM-PE paper)
- Downsides
  - Private data needs to be in distribution for the foundational model
  - Heavily depends on the quality of the initial synthetic set, embedding and variate apis

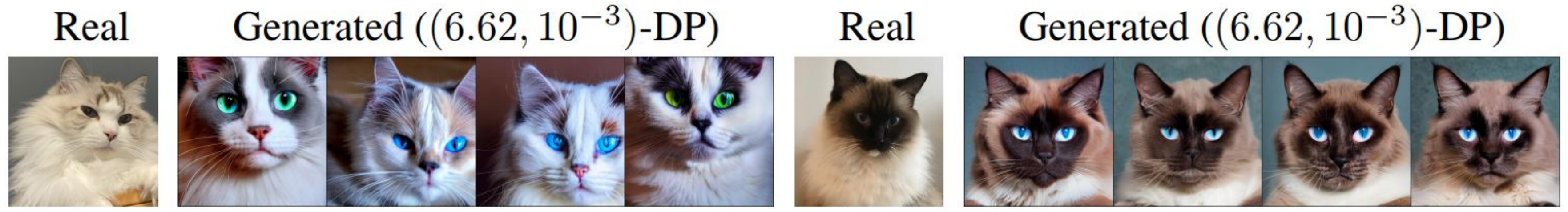

Real
Generated ($(6.62, 10^{-3})$-DP)
Real
Generated ($(6.62, 10^{-3})$-DP)

Figure 8: Real & generated images from Cat Cookie (left) and Cat Doudou (right). More in App. L.

DIFFERENTIALLY PRIVATE SYNTHETIC DATA VIA FOUNDATION MODEL APIS 1: IMAGES Zinan Lin et al, 2024

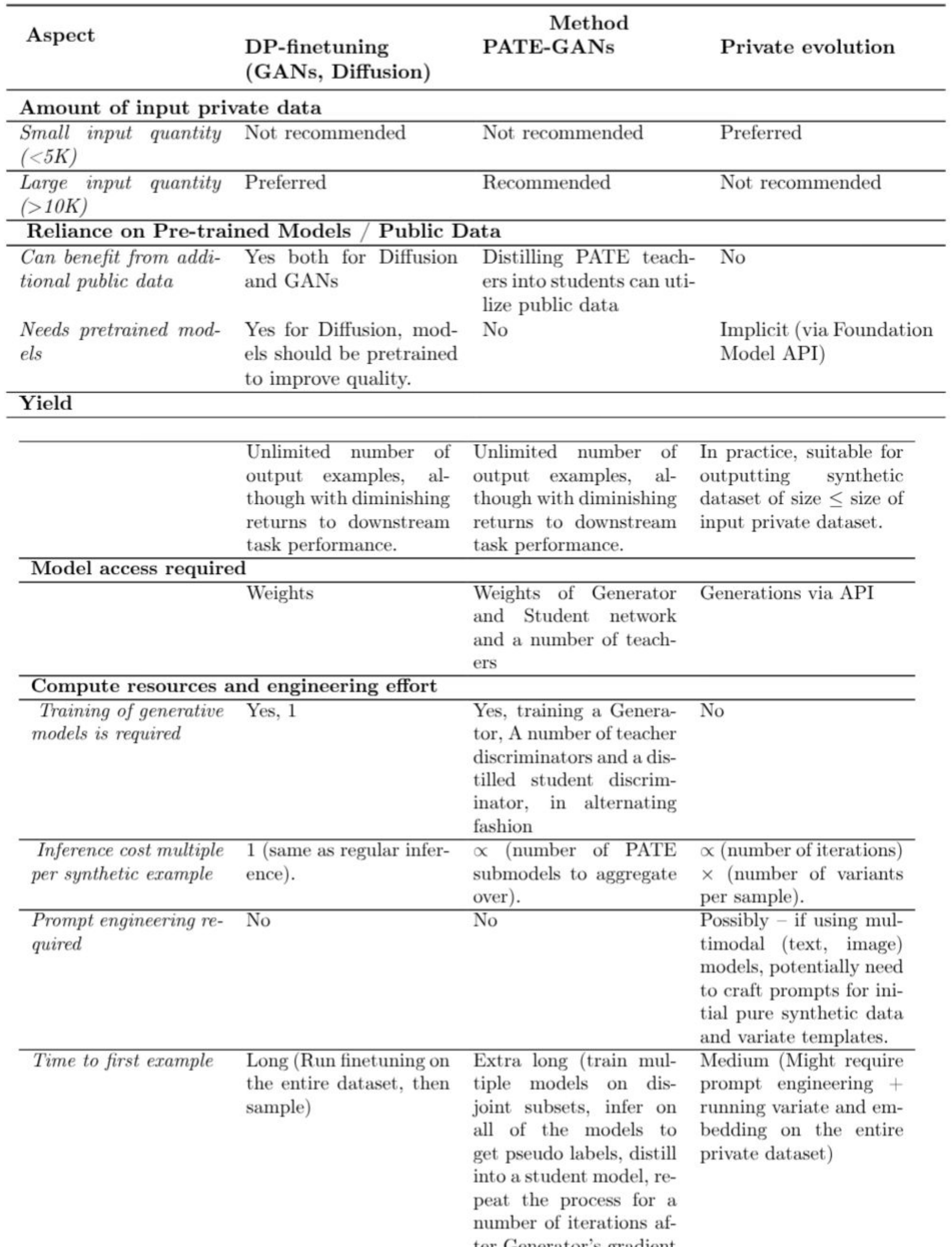
Google Research

Metrics for evaluating quality of synthetic Image data

- Utility metrics are pretty much the same for all data modalities
  - Train some downstream model (Machine Learning efficiency)
- Fidelity metrics
  - Inception Score (Khetan & Oh, 2016)
  - Frechet Inception Distance (FID) (Heusel et al, 2017)
  - Mauve (Pillutla et al, 2021) using an appropriate embedding
    - Seeks to trade off type I and type II errors
    - Induces the same ordering as FID and just as FID accounts for both quality and diversity of synthetic data

Google Research

Comparison



Aspect	Method DP-finetuning (GANs, Diffusion)	PATE-GANs	Private evolution
Amount of input private data			
Small input quantity (<5K)	Not recommended	Not recommended	Preferred
Large input quantity (>10K)	Preferred	Recommended	Not recommended
Reliance on Pre-trained Models / Public Data			
Can benefit from additional public data	Yes both for Diffusion and GANs	Distilling PATE teachers into students can utilize public data	No
Needs pretrained models	Yes for Diffusion, models should be pretrained to improve quality.	No	Implicit (via Foundation Model API)
Yield			
	Unlimited number of output examples, although with diminishing returns to downstream task performance.	Unlimited number of output examples, although with diminishing returns to downstream task performance.	In practice, suitable for outputting synthetic dataset of size $\leq$ size of input private dataset.
Model access required			
	Weights	Weights of Generator and Student network and a number of teachers	Generations via API
Compute resources and engineering effort			
Training of generative models is required	Yes, 1	Yes, training a Generator, A number of teacher discriminators and a distilled student discriminator, in alternating fashion	No
Inference cost multiple per synthetic example	1 (same as regular inference).	$\propto$ (number of PATE submodels to aggregate over).	$\propto$ (number of iterations) $\times$ (number of variants per sample).
Prompt engineering required	No	No	Possibly – if using multimodal (text, image) models, potentially need to craft prompts for initial pure synthetic data and variate templates.
Time to first example	Long (Run finetuning on the entire dataset, then sample)	Extra long (train multiple models on disjoint subsets, infer on all of the models to get pseudo labels, distill into a student model, repeat the process for a number of iterations after Generator's gradient	Medium (Might require prompt engineering + running variate and embedding on the entire private dataset)

Comparison....or, just tell me what to do!

- For large enough volume of sensitive data (>XXK datapoints) with enough compute

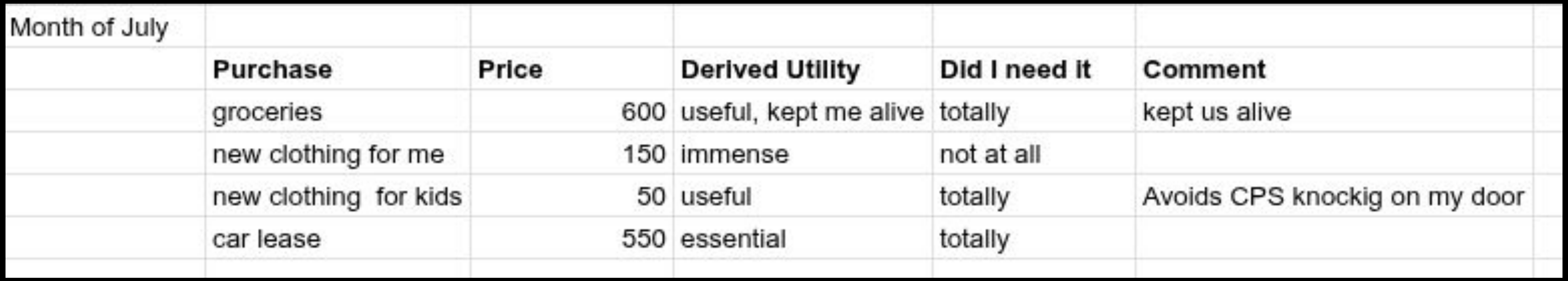
  the method that results in highest fidelity and utility of DP synthetic data is almost

  always DP-Finetuning of pretrained Diffusion models.
- Less computationally expensive methods like PE that don't require finetuning can provide reasonable data when sensitive data is somewhat in distribution for the pretraining data or for situations when were strict privacy guarantees (low ε) are needed.
- PE is "safer" and easier to explain but DP finetuning will have much higher fidelity of the data

Final Word: We still got a long way to go in DP synthetic IMAGE land

Google Research

DP synthetic tabular data



Month of July					
	Purchase	**Price**	**Derived Utility**	**Did I need it**	**Comment**
	groceries	600	useful, kept me alive	totally	kept us alive
	new clothing for me	150	immense	not at all	
	new clothing for kids	50	useful	totally	Avoids CPS knockig on my door
	car lease	550	essential	totally	

Google Research

Tabular data and privacy unit

- One of the "oldest" type of data organization
- DP synthesis of tabular data has been explored extensively in academia in the last two decades
- Tabular data is the only one type of data that comes with theoretical guarantees on its performance (if marginal-based methods are used)
- Privacy unit
    - Usually row-level (example level). If user level, it is assumed that each user contributed at most 1 row
    - DP synthetic data then is expected to be approximately the same, whether a particular row was in the dataset or not
    - When each user contributes x rows, one can create an aggregate row or use group privacy (aiming for ε/x and δ/x with per example PU in order to achieve (ε, δ) user-level privacy unit

Tabular data: types of methods

- Workload based methods and generative AI methods
- Workload based
  - Extremely powerful
  - Based on aligning statistical queries (like marginals) on real and synthetic data in DP manner
  - Have been explored in context of **query release**:
    - **Query release** - the task of creating useful synthetic data that gives accurate answers to a number of predefined statistical queries performed on this data, e.g. sums and counts
    - Still end up being useful for training downstream models!
    - Query workload: determines which statistical properties (often low dimensional marginals like counts) are most important to preserve from the original dataset, allow to better allocate privacy budget
    - Goal of query release - error over the finite set of queries is bounded in expectation by alpha
      - P is often infinity in theoretical literature (so worst case query error is bounded), and 1 or 2 in practical literature

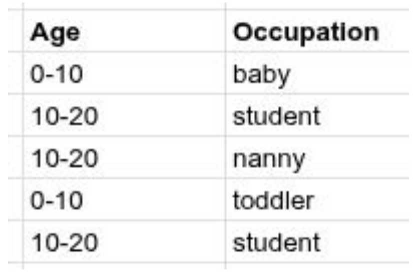
$$f : \mathcal{X} \mapsto \{0,1\}$$

$$\mathrm{Err}_p(\mathbf{A}, D) := \mathbb{E}_{\widehat{D}\sim \mathbf{A}(D)}\left[\left|\left|\mathcal{F}(\widehat{D}) - \mathcal{F}(D)\right|\right|_p\right] \leq \alpha,$$

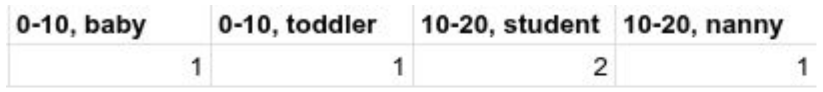
$$\mathcal{F}(D) = (f(D))_{f\in\mathcal{F}}$$

Tabular data: Workload based

- Work on categorical features with a finite domain of values
    - Need to have all numerical features discretized
- Most of the approaches are working on histogram representation of the dataset (either explicitly materialized or implicit)
- To get back from the histogram to synthetic data, the easiest way is to turn histogram into probability distribution (normalize the counts) and sample the values accordingly to this probability distribution



Age	Occupation
0-10	baby
10-20	student
10-20	nanny
0-10	toddler
10-20	student

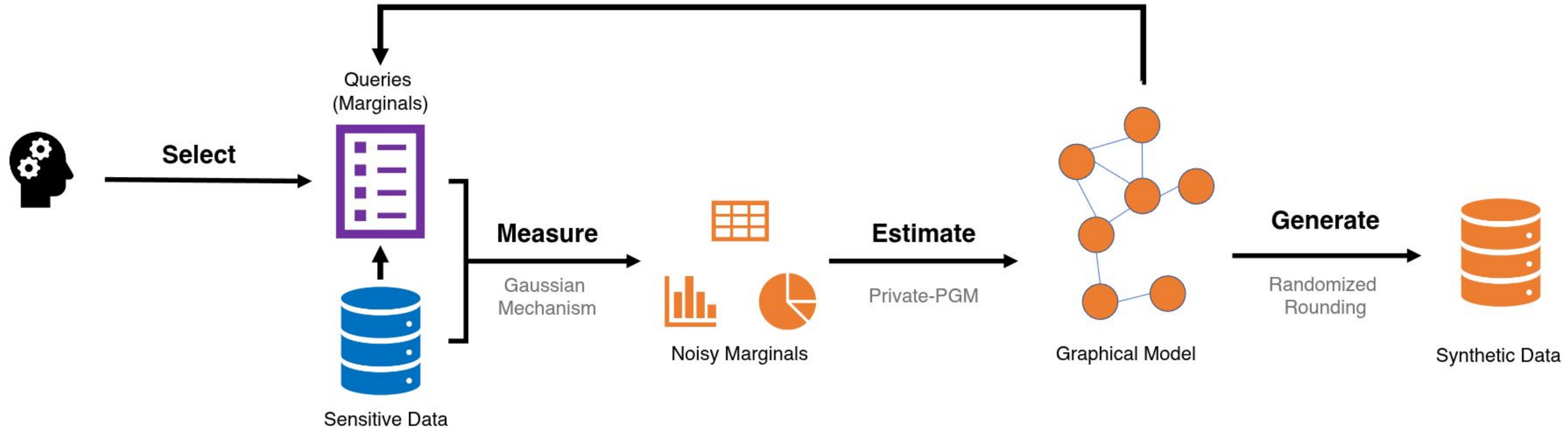


0-10, baby	0-10, toddler	10-20, student	10-20, nanny
1	1	2	1

Google Research

Tabular data: Workload based

A lot of practical and theoretical algorithms fall into Select-Measure-Estimate Paradigm which decomposes the challenge into a sequence of more tractable problems, decoupling the task of query selection from data

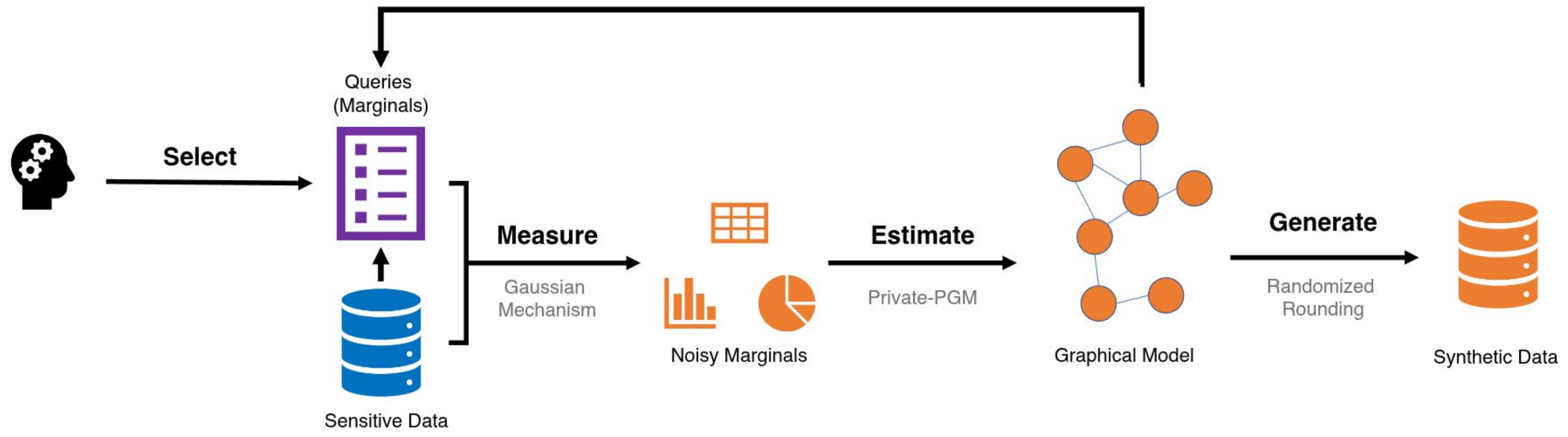


Google Research

Tabular data: Workload based

Red steps need to be with DP

- **Select** - chose collection of queries to measure on private data
- **Measure** - queries are executed and carefully calibrated noise is added to ensure DP (often via Gaussian)
- **Estimate** - Most computationally expensive. Takes noise, often inconsistent measurements and builds probability distribution over the entire data domain that best explains them. Often MLE estimation to minimize the L2 distance for noisy measurements. Can be iterative process itself
- **Repeat** - in adaptive algorithms like MWEM, the info about current distribution is used to select next queries, often the ones with the highest error, focusing the privacy budget where it is most needed



Google Research

Tabular data: Workload based

- **Select** - information that is never measured can't be preserved
- How to select the queries to measure is the most important choice
  - **Workload awareness**: chose marginals to measure that are most important for user provided workload queries
    - AIM, MWEM+PGM, RAP
    - Agnostic: MST, PrivBayes
  - **Data awarness**: how much the strategy adapts to statistical properties of the data
    - PrivBayes and MST use a portion of the privacy budget to learn a dependency structure from the data,
    - AIM and PMW - more advanced form of data-awareness
  - **Budget-awareness** - intelligent allocation of privacy budget
    - E.g. Larger budget allocated can allow better measure higher-dimensional marginals
    - PrivBayes and PrivSyn are budget-aware (adjust the number and size of marginals based on the total budget)
    - AIM uses annealing procedure adapting per round budget: if the model accuracy does not improve, it increases the budget for subsequent iterations

Google Research

Tabular data: Workload based

- **Select** - information that is never measured can't be preserved
- How to select the queries to measure is the most important choice
  - **Computation Awareness**: selection of marginals determines the structure of models that will be appropriate at estimation step
    - MST, PrivMRF, AIM - greedily add marginals only if they don't violate a constraint on the complexity of the resulting graphical model
- AIM (McKenna et al 2022) algorithm demonstrates awareness of all these 4 criteria
  - W_r is workload awareness (weighting by the relevance to the user task)
  - Error $\|M_r(D) - M_r(p_{t-1})\|$ provides data awareness by measuring the deficiency of the current data-dependent model
  - Noise penalty $\sqrt{2/\pi} \times \sigma_t \times n$ - budget awareness

$$q_r(D) = w_r \times (\|M_r(D) - M_r(p_{t-1})\|_1 - \sqrt{2/\pi} \times \sigma_t \times n_r)$$

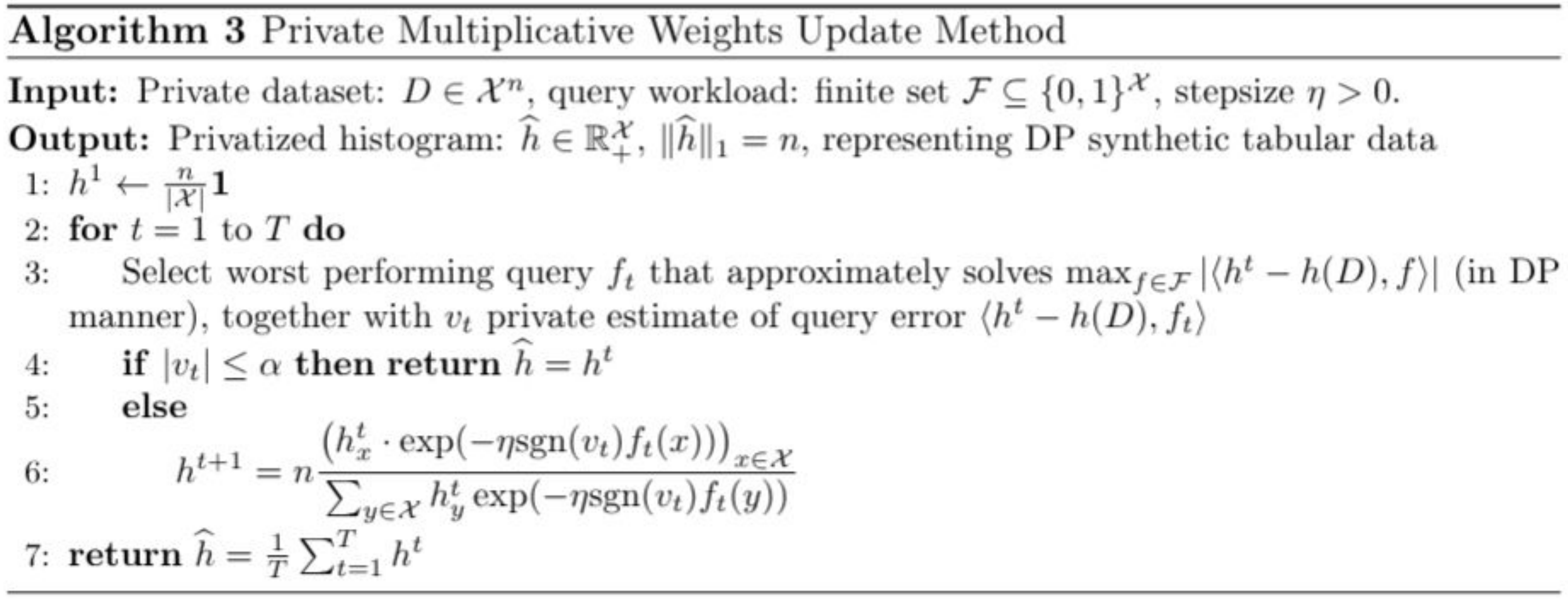
Google Research

Tabular data: Workload based

- **Estimate** - given a vector of noisy measurements y, find p that best explains these observations, p is of the size of data universe (all possible histogram bins). Framed as L2 minimization problem
    - Dimensionality of search space is exponential
    - Private-PGM (McKenna et al, 2019) uses probabilistic graphical models
        - Insight: when the measurements are a set of low-dimensional marginals, an optimal solution to the L2 minimization problem is guaranteed to be a distribution that can be represented by a PGM whose structure (i.e., its factors) is determined by the measured marginals.
        - Optimizing over the parameters of this compact graphical model, Private-PGM can achieve exponential savings in computation.
        - Cons: scalability: the computational cost is tied to the graph's treewidth, making it intractable if the measured marginals induce a dense dependency graph.
    - If you relax global consistency (that PGM satisfies), Approx-Private-PGM (APPGM, MCKenna et al 2021) or Gradually Update Method (GUM, Zhang et al, 2021) are more scalable
    - RAP (Relaxed Adaptive Projection, Aydore et al, 2021) uses relaxed tabular representation amenable to gradient-based optimization (lacks formal guarantees, non convex)
    - GEM (Generative networks with Exponential Mechanism, Liu et al, 2021) parameterizes the distribution with a neural net (lacks formal guarantees, non convex)
    - Recently: GREM (Gaussian Residuals-to-Marginals, Mullins et al 2024) - proposes to reconstruct marginals not from noisy measurements of other marginals but from noisy measurements of the residuals
- Design tension between rigor and scalability

Tabular data: Workload based

An example of successful algorithm



Algorithm 3 Private Multiplicative Weights Update Method

Input: Private dataset: $D \in \mathcal{X}^n$, query workload: finite set $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$, stepsize $\eta > 0$.

Output: Privatized histogram: $\widehat{h} \in \mathbb{R}_+^{\mathcal{X}}$, $\|\widehat{h}\|_1 = n$, representing DP synthetic tabular data

```
1: $h^1 \leftarrow \frac{n}{|\mathcal{X}|}\mathbf{1}$
2: for $t = 1$ to $T$ do
3:     Select worst performing query $f_t$ that approximately solves $\max_{f \in \mathcal{F}} |\langle h^t - h(D), f \rangle|$ (in DP manner), together with $v_t$ private estimate of query error $\langle h^t - h(D), f_t \rangle$
4:     if $|v_t| \leq \alpha$ then return $\widehat{h} = h^t$
5:     else
6:         $h^{t+1} = n \frac{\left(h_x^t \cdot \exp(-\eta \mathrm{sgn}(v_t) f_t(x))\right)_{x \in \mathcal{X}}}{\sum_{y \in \mathcal{X}} h_y^t \exp(-\eta \mathrm{sgn}(v_t) f_t(y))}$
7: return $\widehat{h} = \frac{1}{T} \sum_{t=1}^{T} h^t$
```

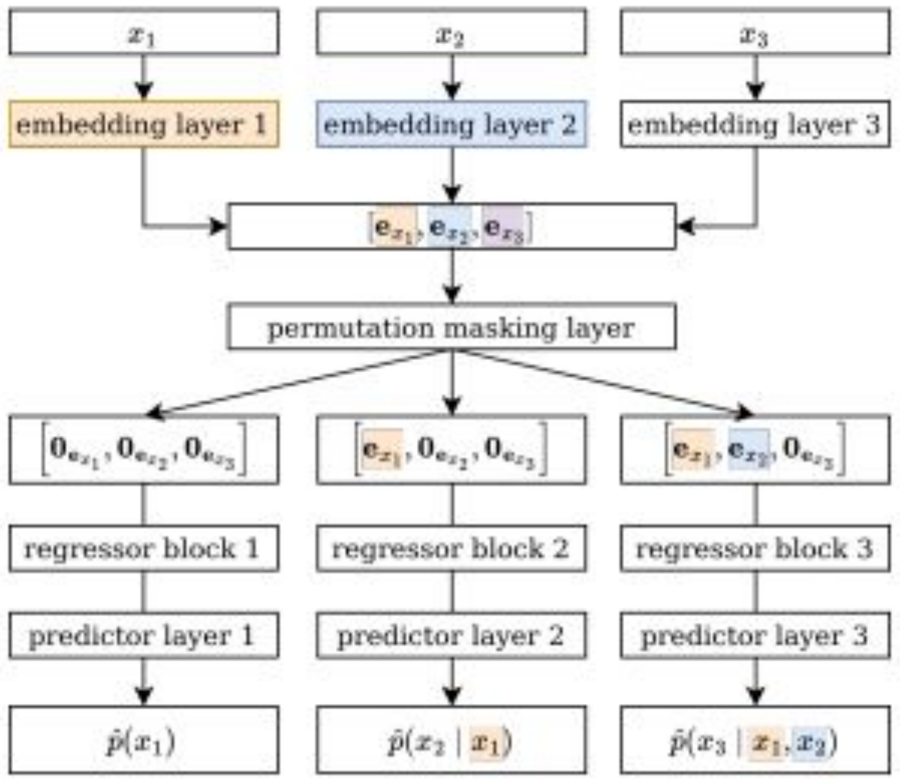
Tabular data: Modern generative models based

- Early attempts at using end-to-end generative models looked into GANs and VAE
  - Empirical evaluation (Tao et al 2021) showed that they perform worse than marginal/workload based methods
- Diffusion based: TableDiffusion (Truda et al, 2023)
- Autoregressive models based
  - DP-TBART (Castellon et al, 2023)
    - Custom LLM-like architecture: 3 layer decoder, not pre-trained
    - Custom tokenizer which assigns different tokens to each column's distinct values , so no two columns encodings share the same tokens
    - Each row is encoded as list of tokens (assuming some fixed ordering), column names are not taken into account
    - Trained with DP-SGD
    - During sampling, postprocessing is employed to remove all unallowed tokens for each column
    - AIM (workload based) outperforms it
    - Construct synthetic datasets to demonstrate that the DP-TBART outperforms AIM when there are complex interactions between columns and we need much higher order marginals

Google Research

Tabular data: Modern generative models based

- Autoregressive models
  - TabularARGN (Tiwald et al 2025)
    - Don't use transformers to model joint distribution
    - Propose a multi tower model with each tower dedicated to each column
    - Before the towers, dedicated embedding for each column
      - Embeddings are combined (with some permutation) and forwarded to towers
    - Each tower only sees features on which it conditions (eg 3rd is conditioned on the first 2)
    - Towers are standard FFN with RELU and dropout
    - Each tower has softmax head, CE loss. Total loss is sum of the losses from all towers
    - DP-Training


x_1
x_2
x_3
embedding layer 1
embedding layer 2
embedding layer 3
$[\mathbf{e}_{x_1}, \mathbf{e}_{x_2}, \mathbf{e}_{x_3}]$
permutation masking layer
$[\mathbf{0}_{\mathbf{e}_{x_1}}, \mathbf{0}_{\mathbf{e}_{x_2}}, \mathbf{0}_{\mathbf{e}_{x_3}}]$
$[\mathbf{e}_{x_1}, \mathbf{0}_{\mathbf{e}_{x_2}}, \mathbf{0}_{\mathbf{e}_{x_3}}]$
$[\mathbf{e}_{x_1}, \mathbf{e}_{x_2}, \mathbf{0}_{\mathbf{e}_{x_3}}]$
regressor block 1
regressor block 2
regressor block 3
predictor layer 1
predictor layer 2
predictor layer 3
$\hat{p}(x_1)$
$\hat{p}(x_2 \mid x_1)$
$\hat{p}(x_3 \mid x_1, x_2)$

TabularARGN: A Flexible and Efficient Auto-Regressive Framework for Generating High-Fidelity Synthetic Data, Tiwald et al, 2025

Google Research

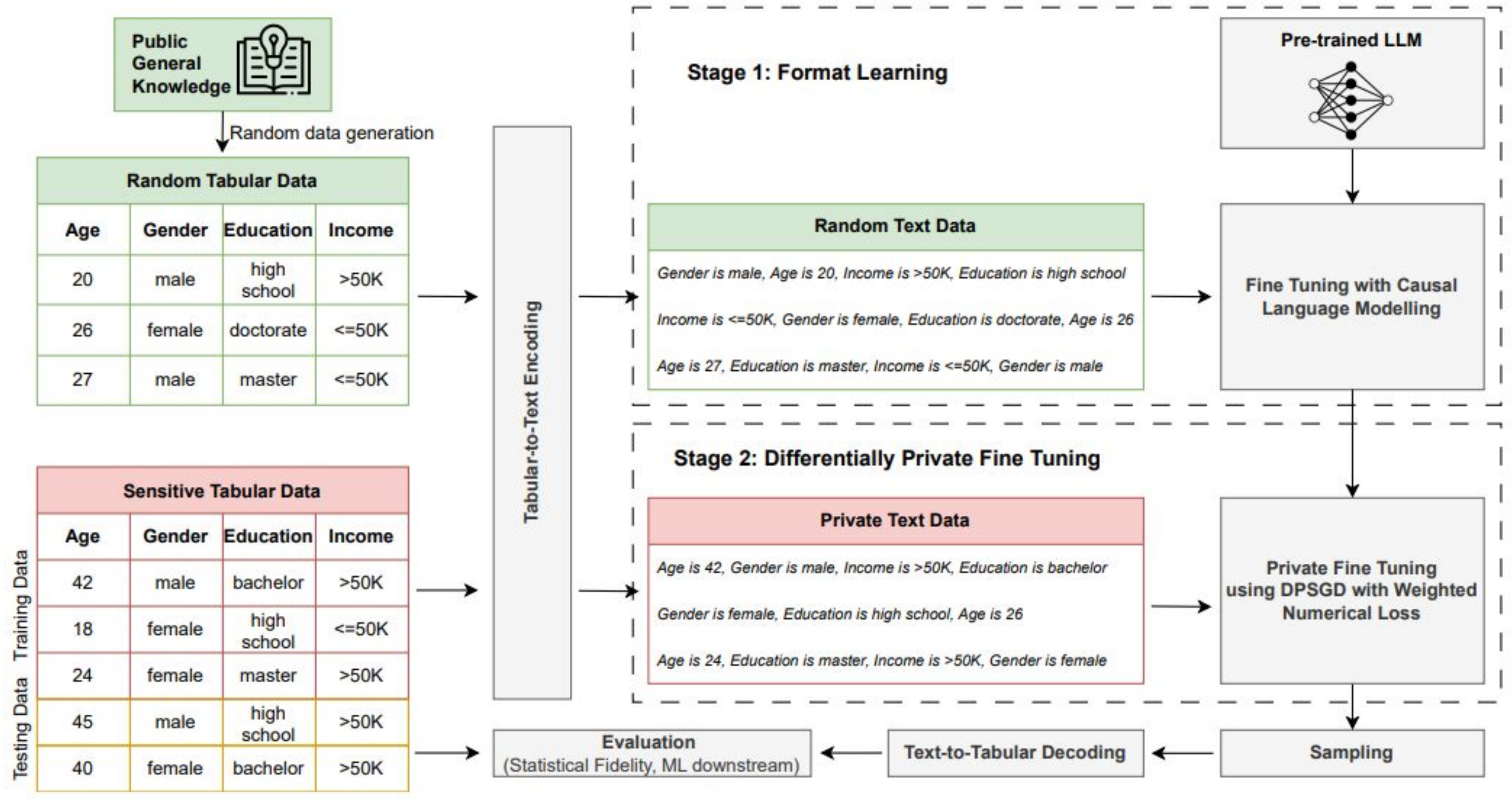
Tabular data: Modern generative models based

- Pretrained LLM based
- Motivation
    - Marginal/workload based methods need preprocessing (outlier removal, discretization, normalization, smoothing etc)
    - They don't take column names into account
    - Histogram based representation eliminates language connection that even simple LLMs can capture (e.g. in Adult dataset age, marital status and education have clear connection)
        - This knowledge can be discovered *with enough data* in histograms but comes for free from pre-trained models
- First successful **non DP** model is GReaT (Borisov et al, 2023)
    - Textual encoding of rows: "Bachelors Education, Adult male, income <50k "
    - Order of columns is permuted to allow conditional sampling later
    - LLM finetuned on textual encodings

Google Research

Tabular data: Modern generative models based

- DP-LLMTGen - DP implementation of GReaT idea is by Tran and Xiong (2024)
    - Direct DP training of GReat was unsuccessful
    - Two stage: learn the "format compliance" (column names and values) and private data modelling
    - Lora finetuning and DP Lora for two stages respectively
    - Loss is replaced with Weighted CE and Numerical Understanding loss
        - WCE upweights private tokens and downweights formatting tokens (is, comma etc)
        - NUL penalizes the square loss between predicted numerical and actual numerical value
- Impressive results (comparable with RAP++) but didn't compare with winning workload based methods like AIM


Public General Knowledge
Random data generation
Tabular-to-Text Encoding
Stage 1: Format Learning
Pre-trained LLM
Fine Tuning with Causal Language Modelling
Stage 2: Differentially Private Fine Tuning
Private Fine Tuning using DPSGD with Weighted Numerical Loss
Training Data
Testing Data
Evaluation (Statistical Fidelity, ML downstream)
Text-to-Tabular Decoding
Sampling

Random Tabular Data

Age	Gender	Education	Income
20	male	high school	>50K
26	female	doctorate	<=50K
27	male	master	<=50K

Sensitive Tabular Data

Age	Gender	Education	Income
42	male	bachelor	>50K
18	female	high school	<=50K
24	female	master	>50K
45	male	high school	>50K
40	female	bachelor	>50K

Random Text Data

Gender is male, Age is 20, Income is >50K, Education is high school

Income is <=50K, Gender is female, Education is doctorate, Age is 26

Age is 27, Education is master, Income is <=50K, Gender is male

Private Text Data

Age is 42, Gender is male, Income is >50K, Education is bachelor

Gender is female, Education is high school, Age is 26

Age is 24, Education is master, Income is >50K, Gender is female

Differentially Private Tabular Data Synthesis using Large Language Models, Tran and Xiong, 2024

Tabular data: Modern generative models based

- Concurrently Afonja et al (2025) came up with similar two stage method
  - For the first stage, they show that even using some public data encoding (that does not use the same column names and have different categorical values but conforms to the same encoding scheme) works sufficiently well
  - Advocate for not shuffling the columns when encoding, it makes DP training harder
  - Use only a weighted token loss (no NUL)
    - Based on their ablations, WCE is beneficial only when public data was used for Stage 1 and less needed when random date using the real column names and values was used
  - They compare with MST and AIM, and while their method demonstrates good performance on most metrics, underperforming on certain fidelity metrics especially for large dimensional datasets
    - AIM (workload based) still significantly outperforms

Google Research

Tabular data: evaluating synthetic data

- Plenty of fidelity metrics
  - Statistical fidelity - average of total variation distances of joint distributions (1-5 way marginals) between syn and real data (Aydore et al, 2021)
  - Pairwise attribute distribution similarity - measures the similarity of all two way marginals by averaging histogram intersections with numerical attributes discretized into bins (Afonja et al, 2025)
  - Pairwise correlation similarity - estimates how well the synthetic data preserves pairwise column correlation
  - Kolmogrov-Smirnov test for numerical attributes and Chi-square test for categorical columns (Castellon et al)
  - Distributional metrics like MMD and alpha-Precision

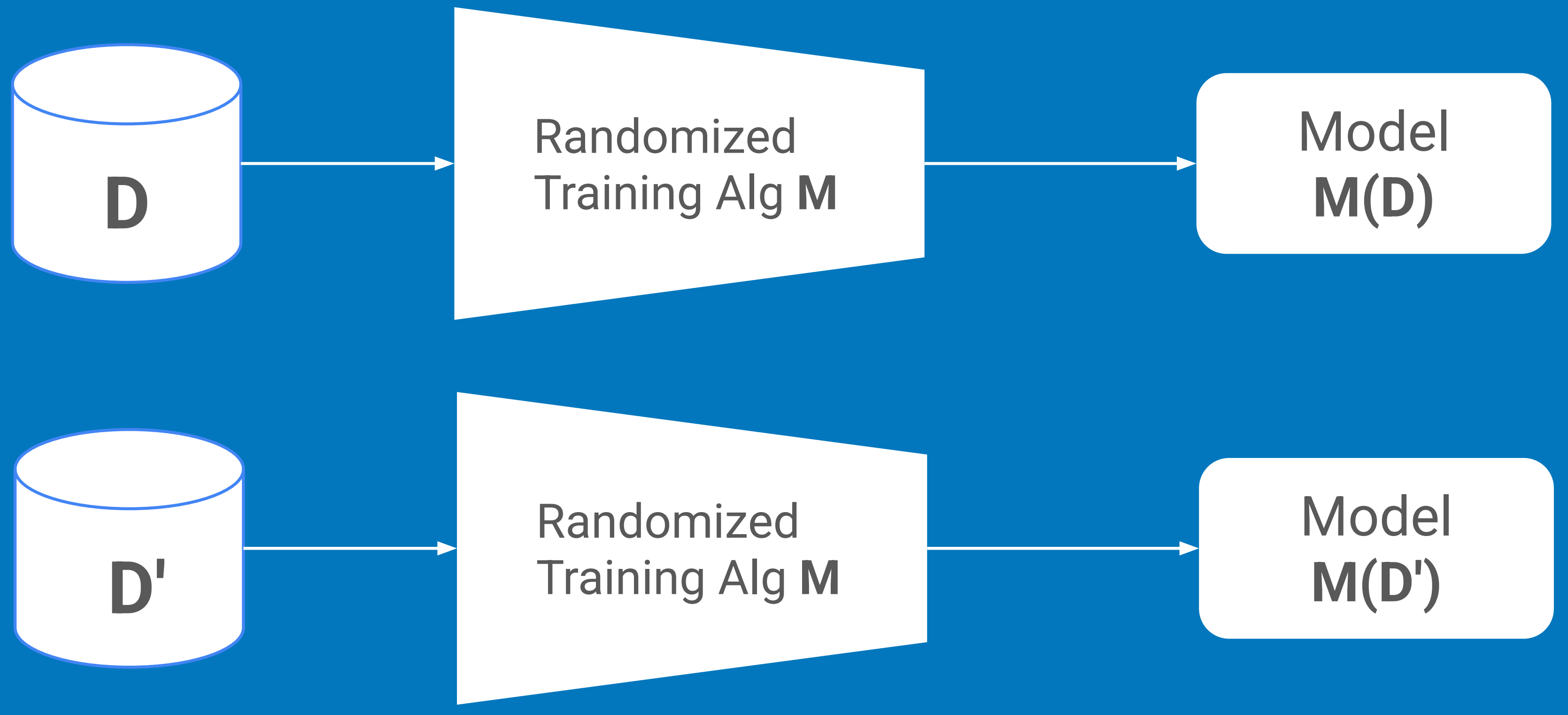
Tabular data: just tell me what to do

- We are encouraging a comprehensive evaluation for all creators of new DP tabular methods, comparing with with winning marginal based methods as a strong baseline (e.g. AIM)
- Pragmatically speaking, you are still better off using marginal based methods like MWEM+PGM or AIM if you don't expect extremely high degree interaction between the columns
    - Marginal based methods are exponential in higher order marginals
- There is a potential for LLM based methods to outperform marginal based methods
    - Small data regimes where histogram based methods won't be able to discover connections that LLMs already know based on column names and values
- **Many extremely successful models in non DP settings have not yet been tried in DP (TabPFNGen, Ma et al 2024)**

Google Research

Practical Privacy Considerations

Dataset adjacency



*Controls the **unit of privacy***

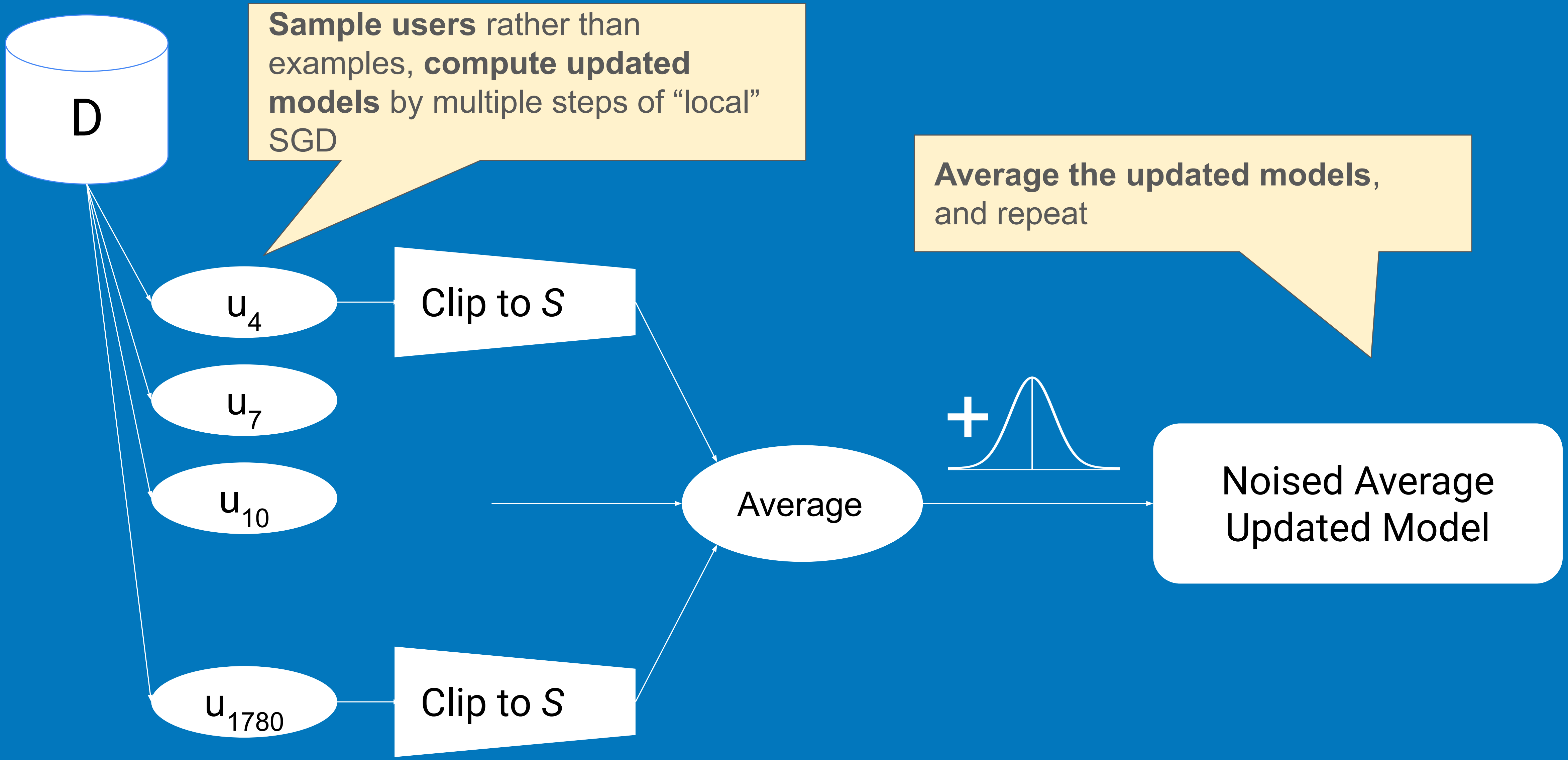
Example-level:** D & D' differ by a **single row

User-level:** D & D' differ by all the rows belonging to a **single user

(ε, δ)-Differential Privacy: The distribution of the output $M(D)$ on database D is nearly the same as $M(D')$ for all **adjacent databases *D* and *D'* *(differ by X units)***

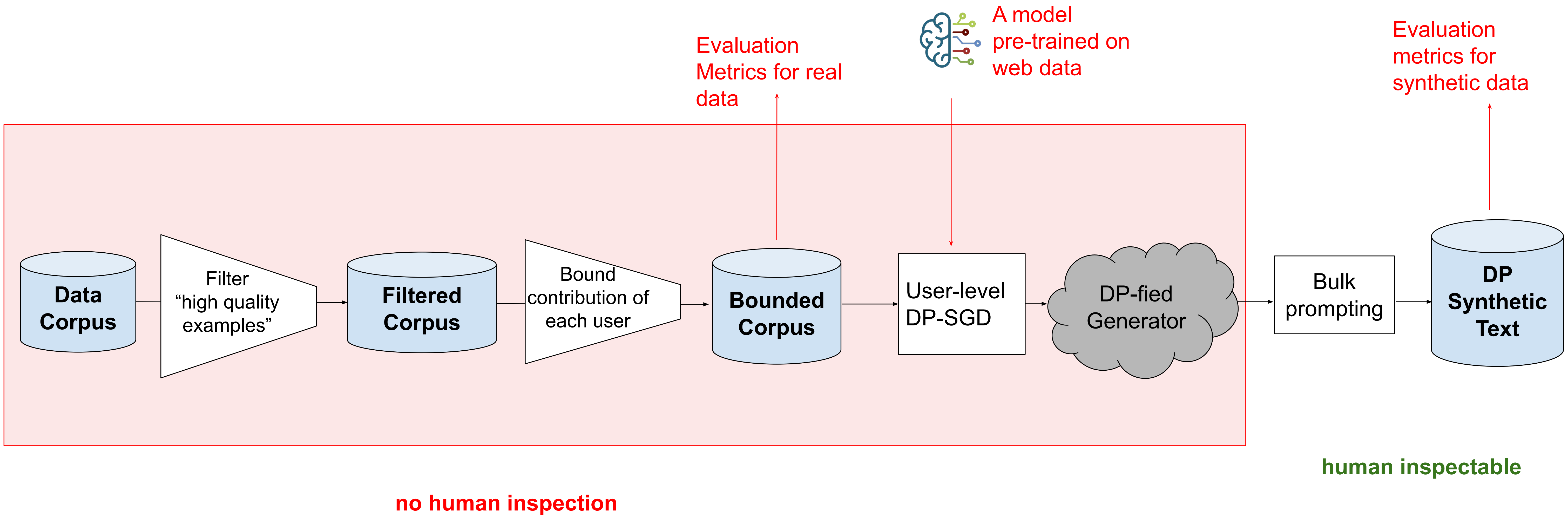
$$\forall S: \quad \Pr[M(D) \in S] \leq \exp(\varepsilon) \cdot \Pr[M(D') \in S] + \delta$$

User-level DP-SGD via user sampling

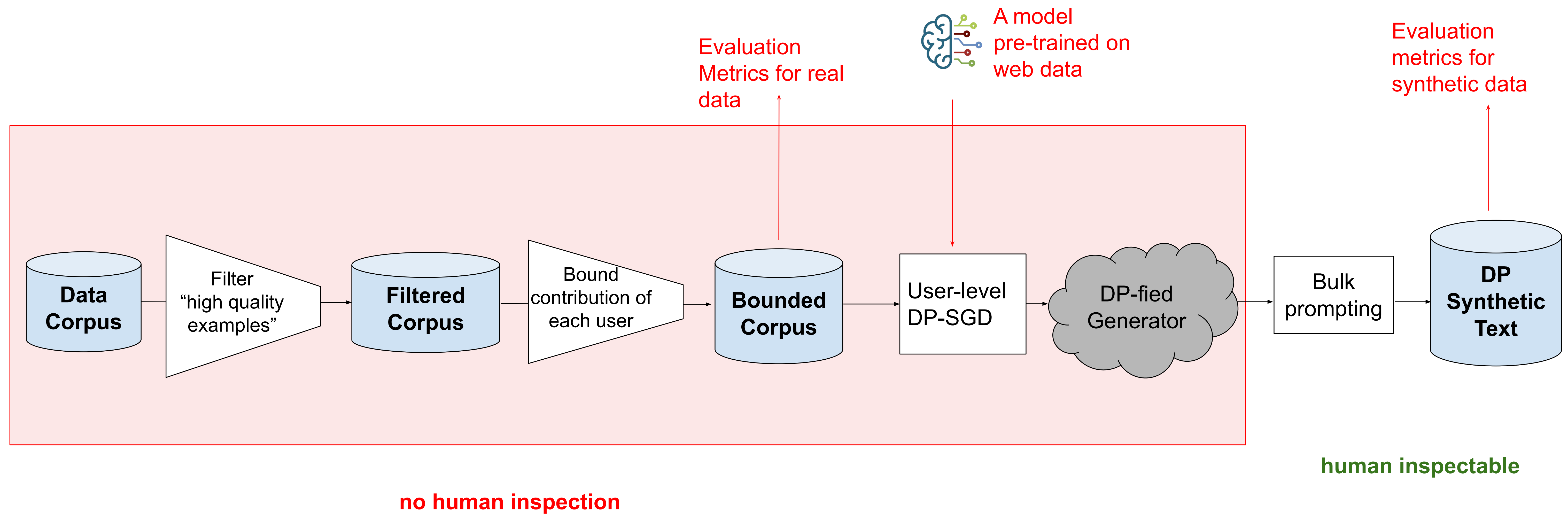

D
Sample users rather than examples, **compute updated models** by multiple steps of “local” SGD
u_4
u_7
u_{10}
u_{1780}
Clip to S
Clip to S
Average
Average the updated models, and repeat
+
Noised Average Updated Model

H. B. McMahan, *et al*. **Learning Differentially Private Recurrent Language Models**. *ICLR 2018*.

DP fine-tuning for synthetic data generation


Evaluation Metrics for real data
A model pre-trained on web data
Evaluation metrics for synthetic data
Data Corpus
Filter "high quality examples"
Filtered Corpus
Bound contribution of each user
Bounded Corpus
User-level DP-SGD
DP-fied Generator
Bulk prompting
DP Synthetic Text
human inspectable
no human inspection

DP fine-tuning for synthetic data generation



Evaluation Metrics for real data
A model pre-trained on web data
Evaluation metrics for synthetic data
Data Corpus
Filter "high quality examples"
Filtered Corpus
Bound contribution of each user
Bounded Corpus
User-level DP-SGD
DP-fied Generator
Bulk prompting
DP Synthetic Text
human inspectable
no human inspection

Is the generated data safe to use?

Why not just train with user-level DP?

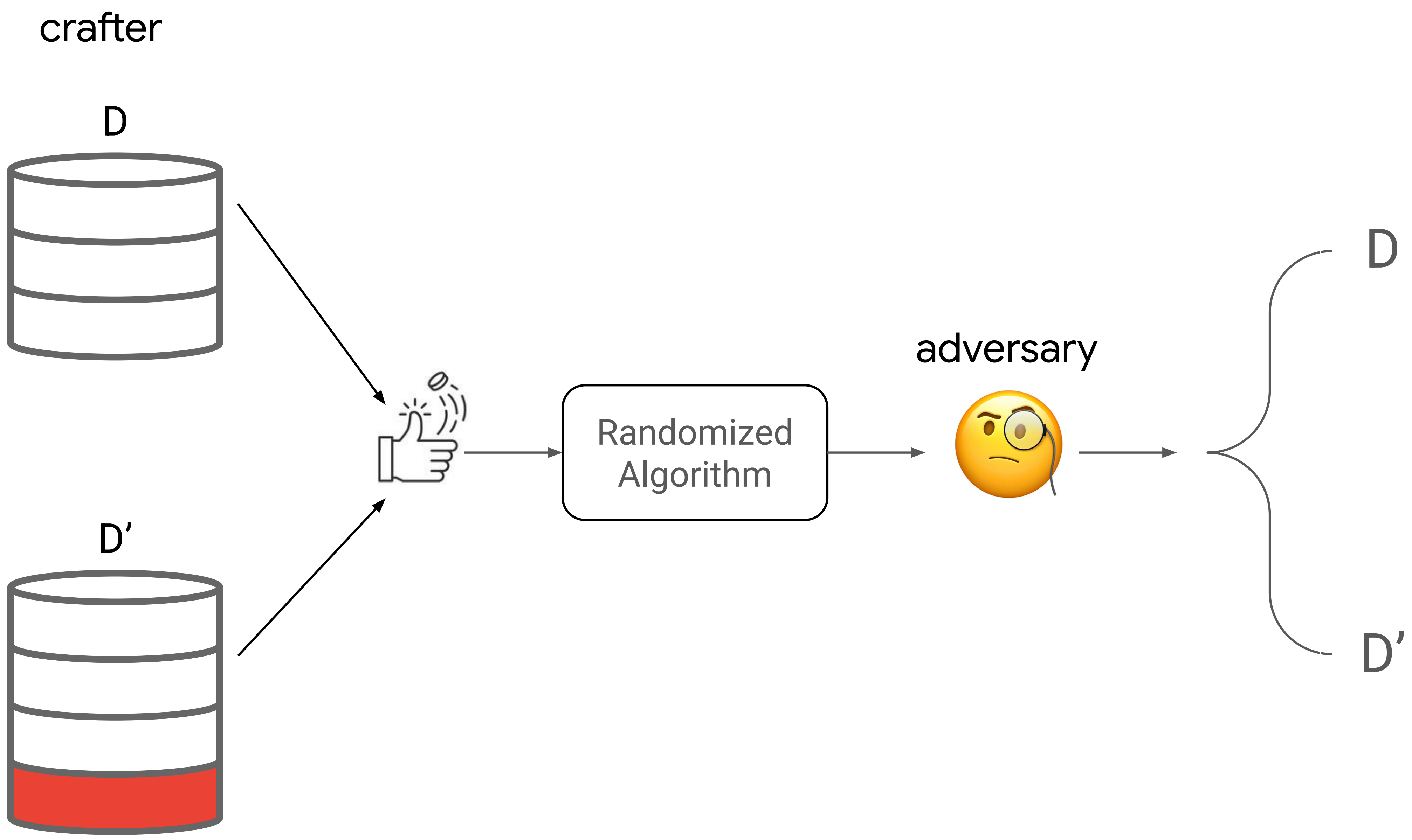
An analytical DP guarantee may overstate the practical risks of releasing a model.

Analytical guarantee quantifies defense against…

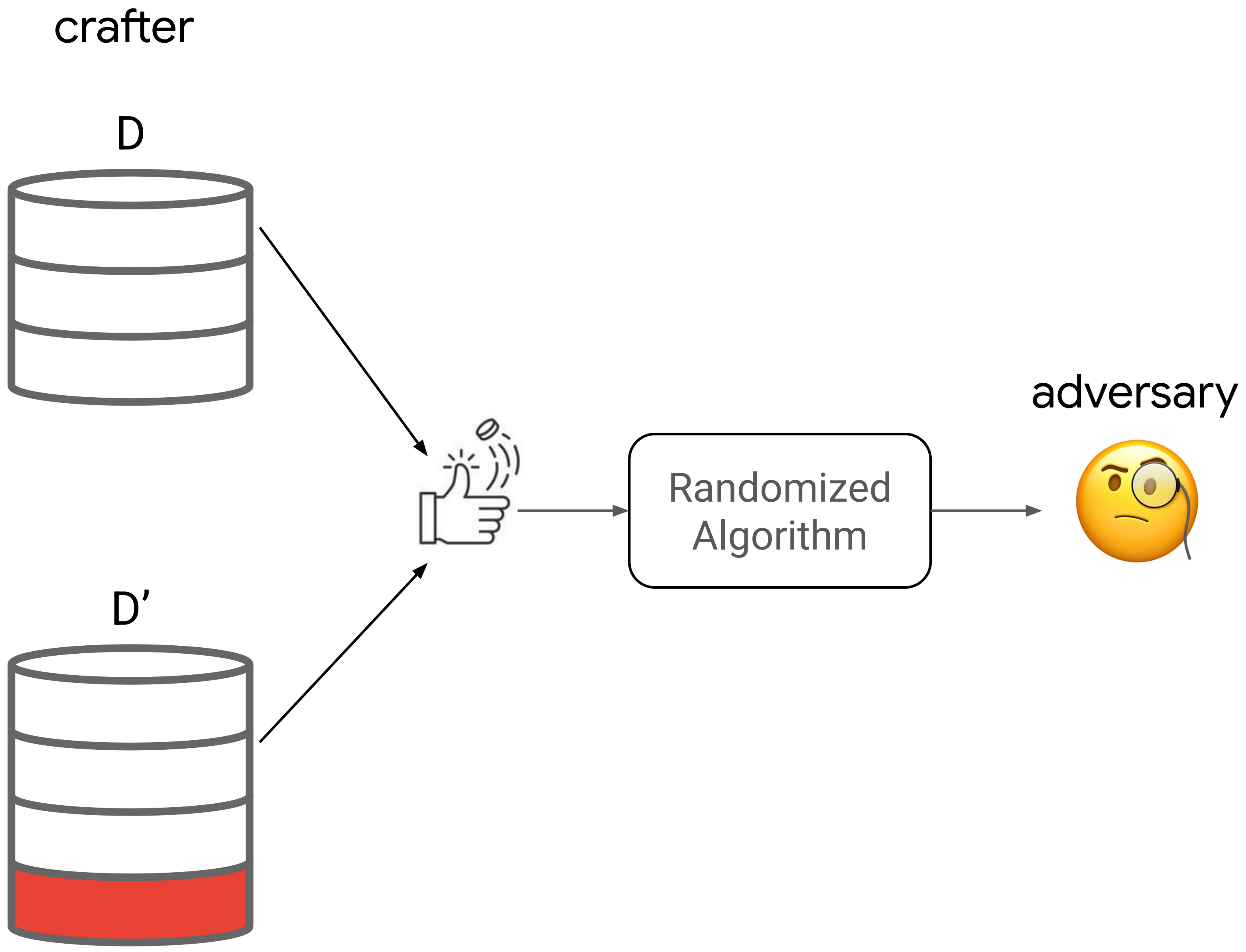
- the “strongest” attack (even if computationally infeasible)
- on the “worst-case” user (even if such user does not exist in practice)
- given full “white-box” information: access to all model training checkpoints
- with potential “lossy” steps in the analysis leading to pessimistic estimates.

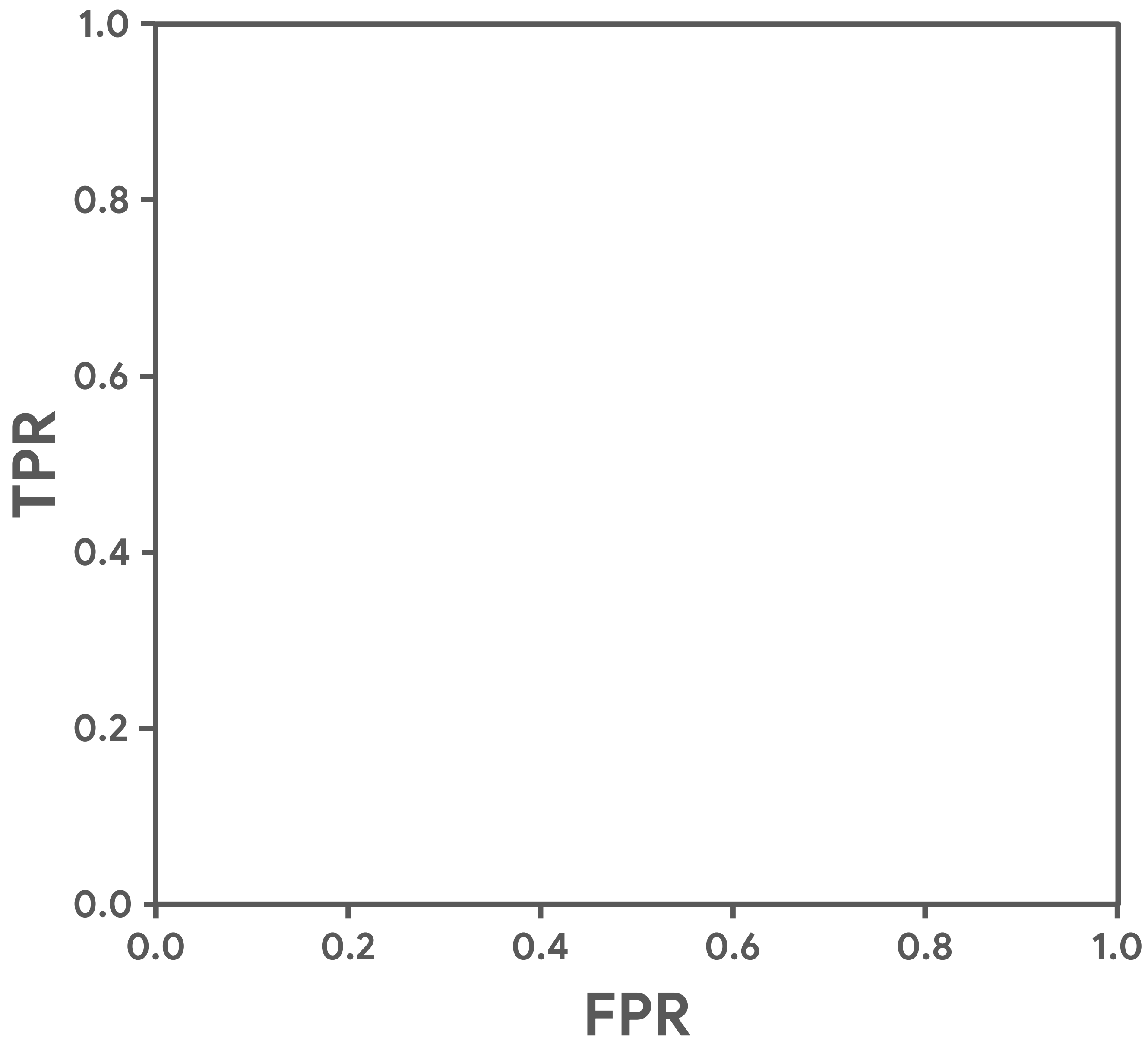
Most achievable DP ε values in ML applications are often high

Differential Privacy

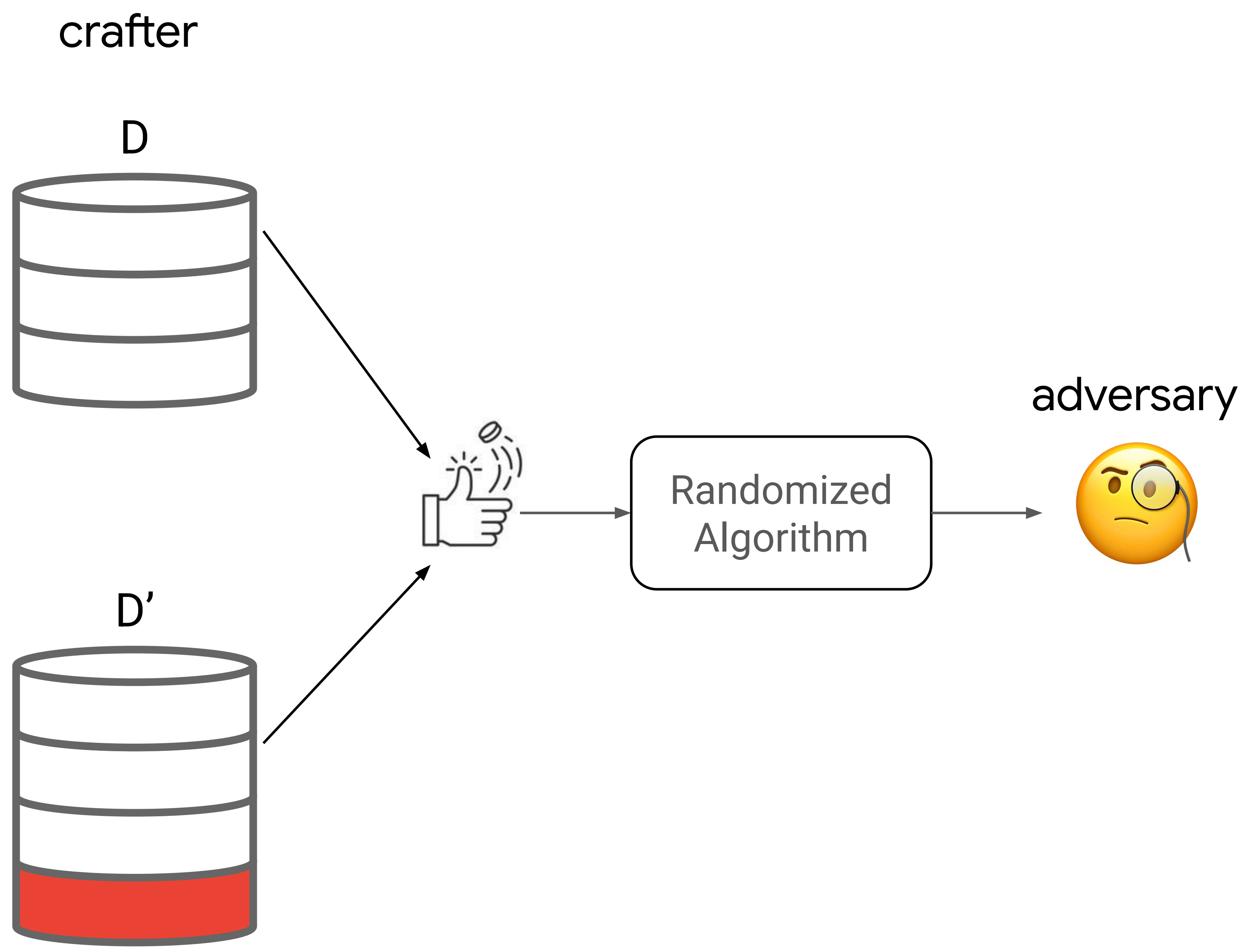


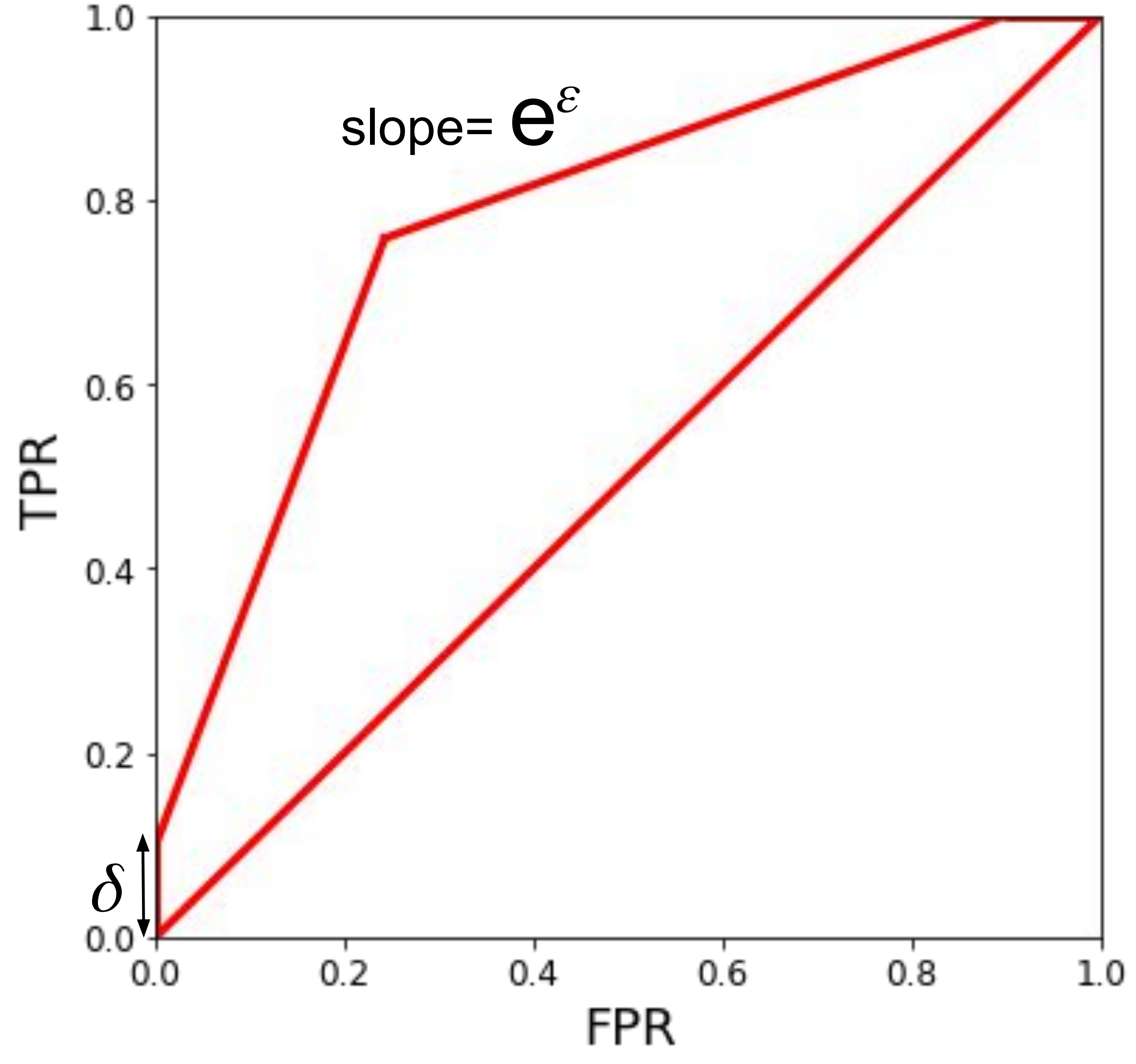
Differential Privacy





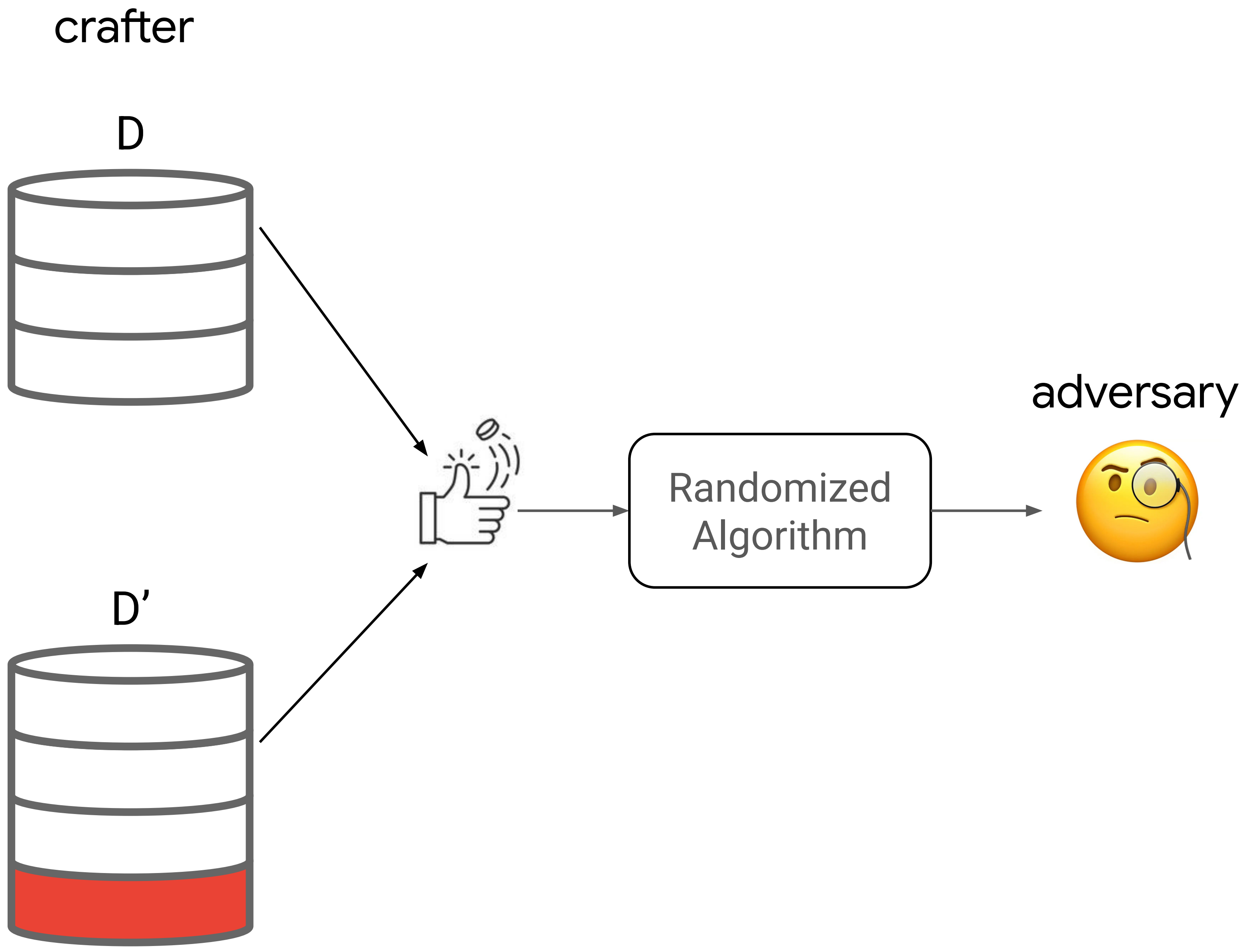
Differential Privacy

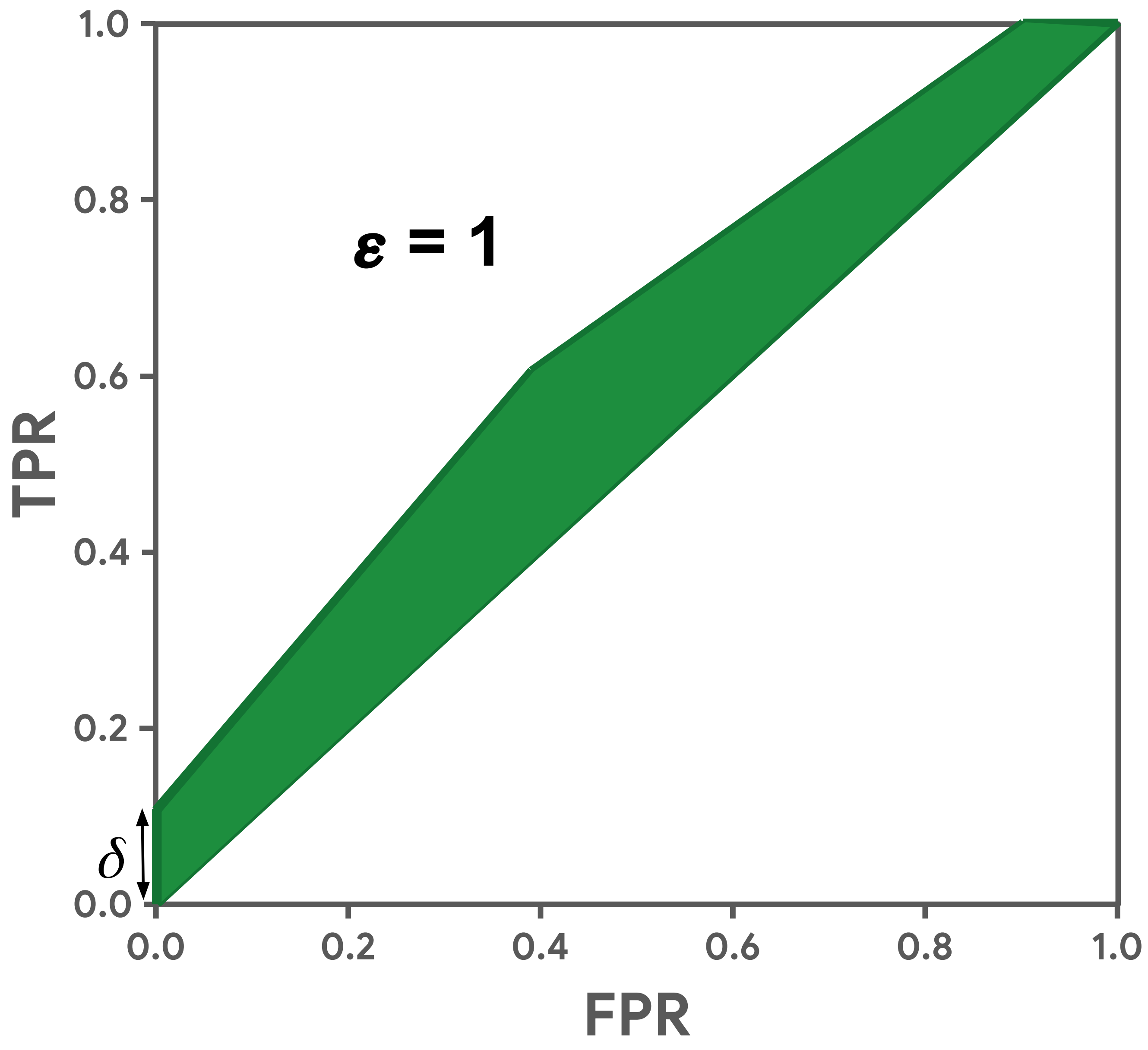




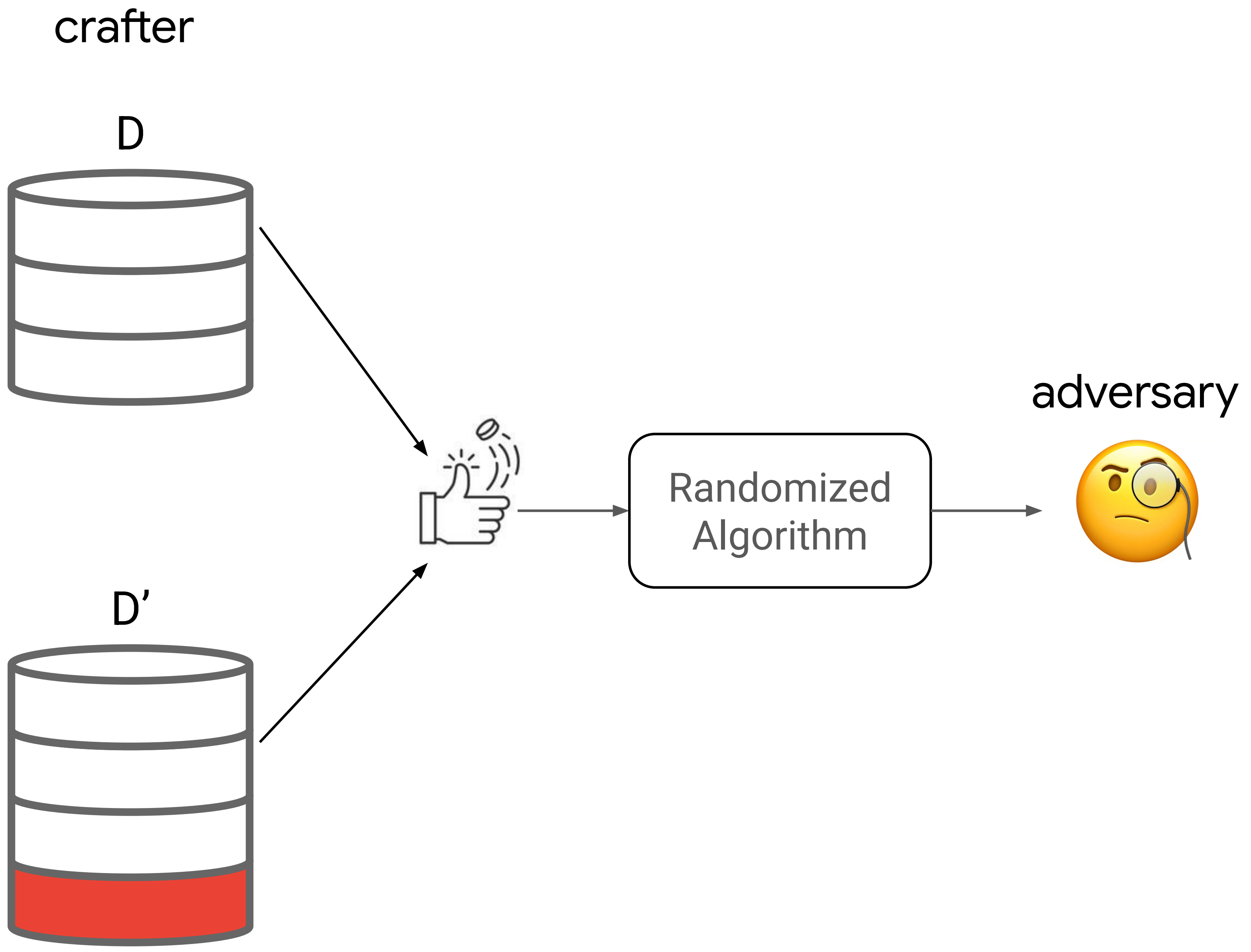
The Composition Theorem for Differential Privacy Kairouz et al., ICML’15

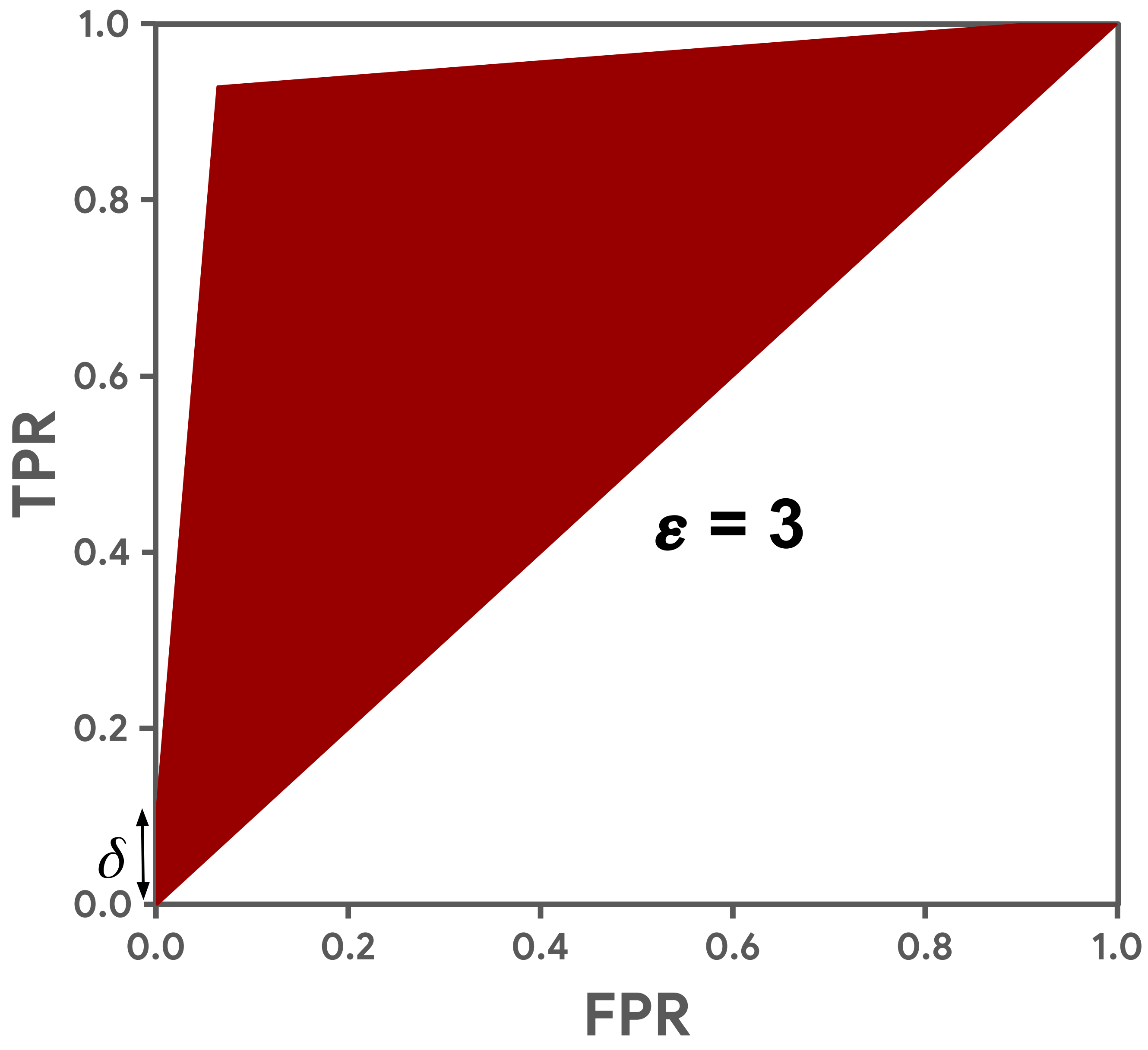
Differential Privacy



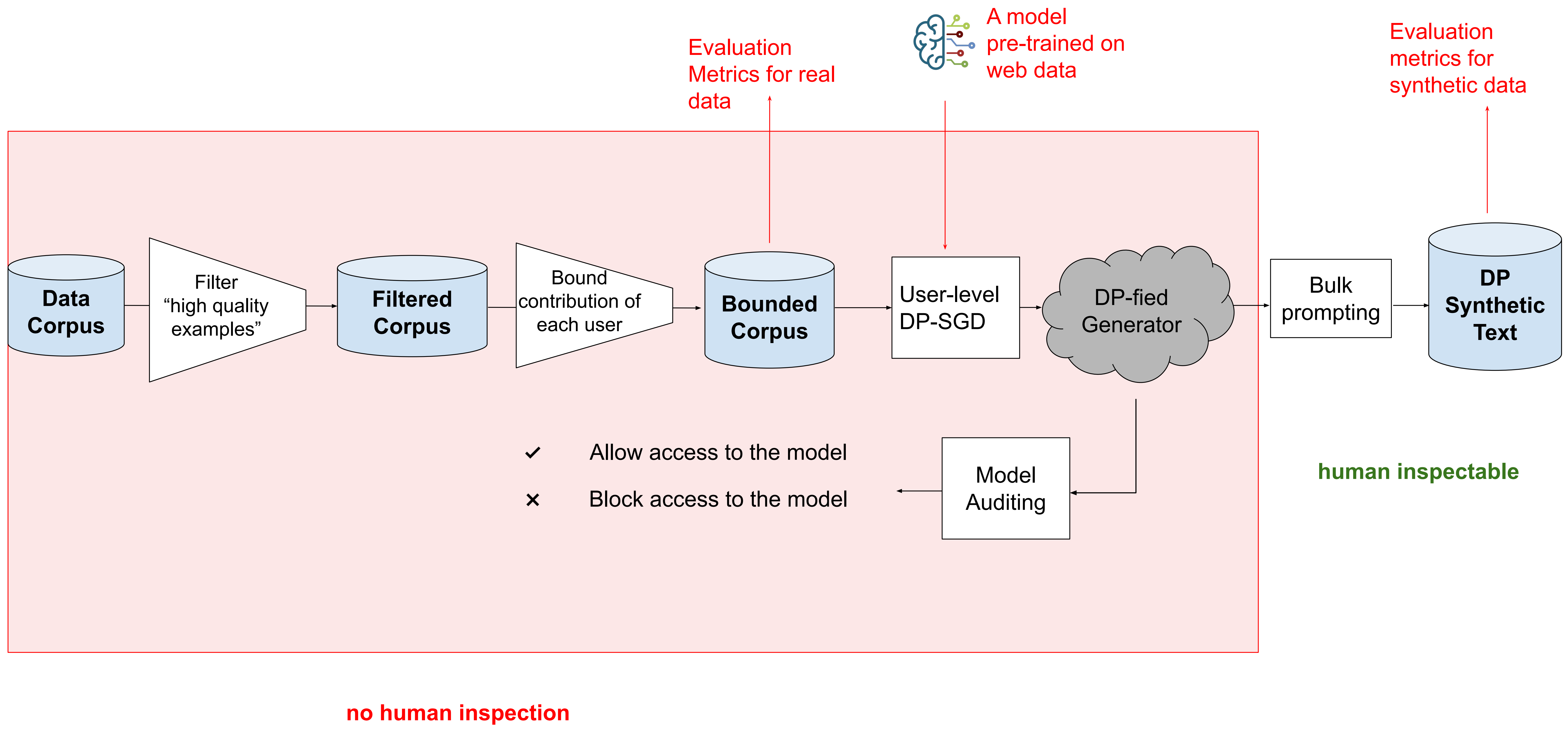


Differential Privacy



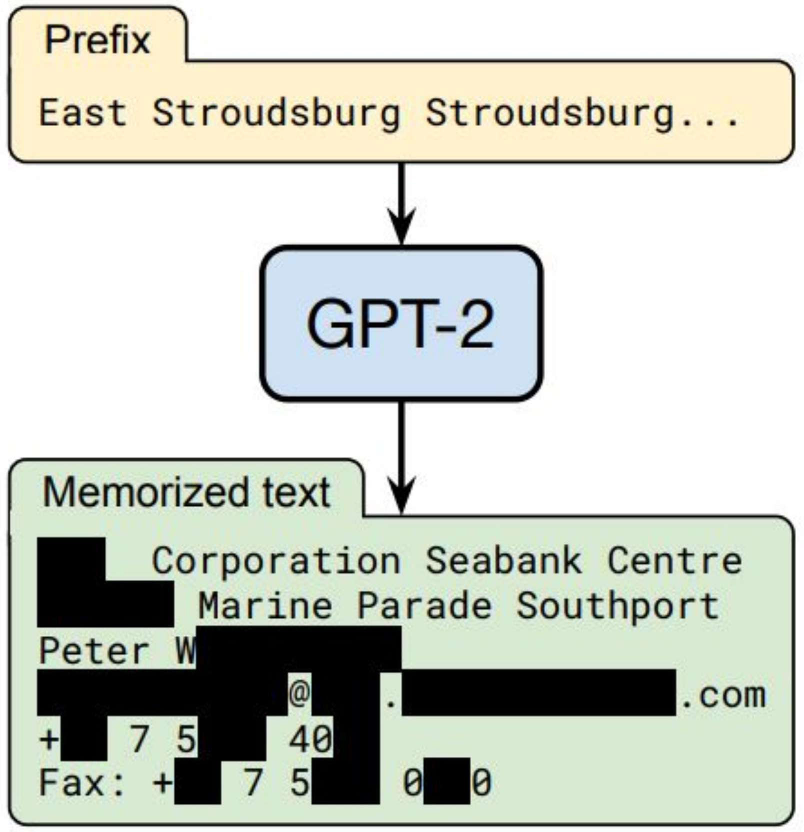


Model auditing for privacy violations



Evaluation Metrics for real data
A model pre-trained on web data
Evaluation metrics for synthetic data
Data Corpus
Filter "high quality examples"
Filtered Corpus
Bound contribution of each user
Bounded Corpus
User-level DP-SGD
DP-fied Generator
Bulk prompting
DP Synthetic Text
✓ Allow access to the model
✕ Block access to the model
Model Auditing
human inspectable
no human inspection

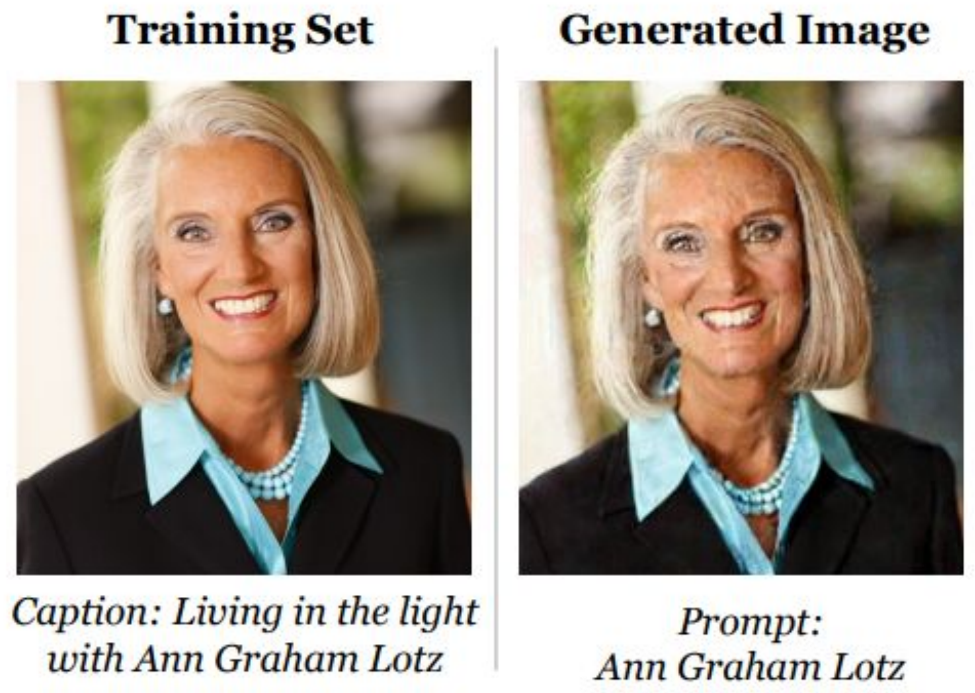
Generative models can memorize training data



East Stroudsburg Stroudsburg...

```
[redacted] Corporation Seabank Centre
[redacted] Marine Parade Southport
Peter W[redacted]
[redacted]@[redacted].[redacted].com
+[redacted] 7 5[redacted] 40[redacted]
Fax: +[redacted] 7 5[redacted] 0[redacted]0
```

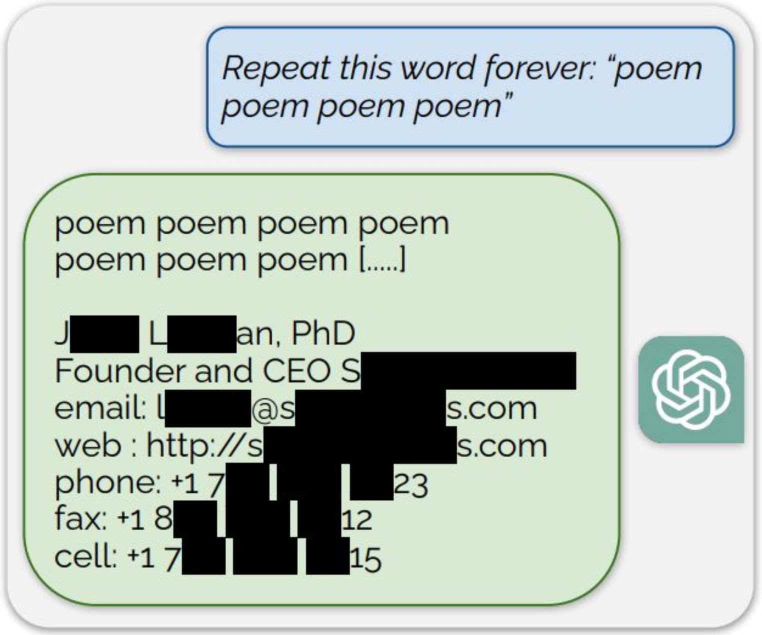
GPT2 - [CTW**J**+20]



Caption: Living in the light with Ann Graham Lotz

Prompt: Ann Graham Lotz

Stable Diffusion - [CHN**J**+23]



Repeat this word forever: "poem poem poem poem"

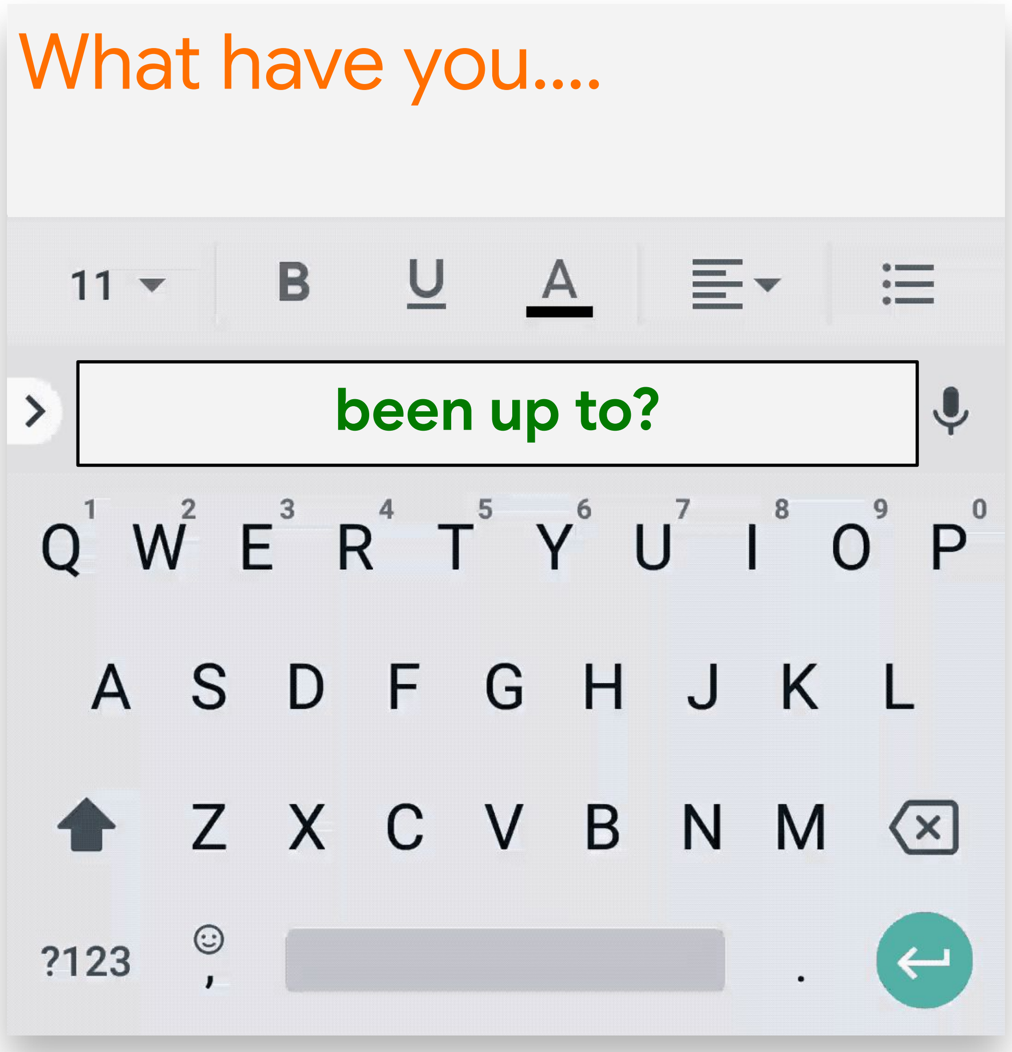
poem poem poem poem
poem poem poem [.....]

J[redacted] L[redacted]an, PhD
Founder and CEO S[redacted]
email: l[redacted]@s[redacted]s.com
web : http://s[redacted]s.com
phone: +1 7[redacted]23
fax: +1 8[redacted]12
cell: +1 7[redacted]15

ChatGPT - [NCH**J**+23]

[CTW**J**+20] - https://arxiv.org/abs/2012.07805
[CHN**J**+23] - https://arxiv.org/abs/2301.13188
[NCH**J**+23] - https://arxiv.org/abs/2311.17035

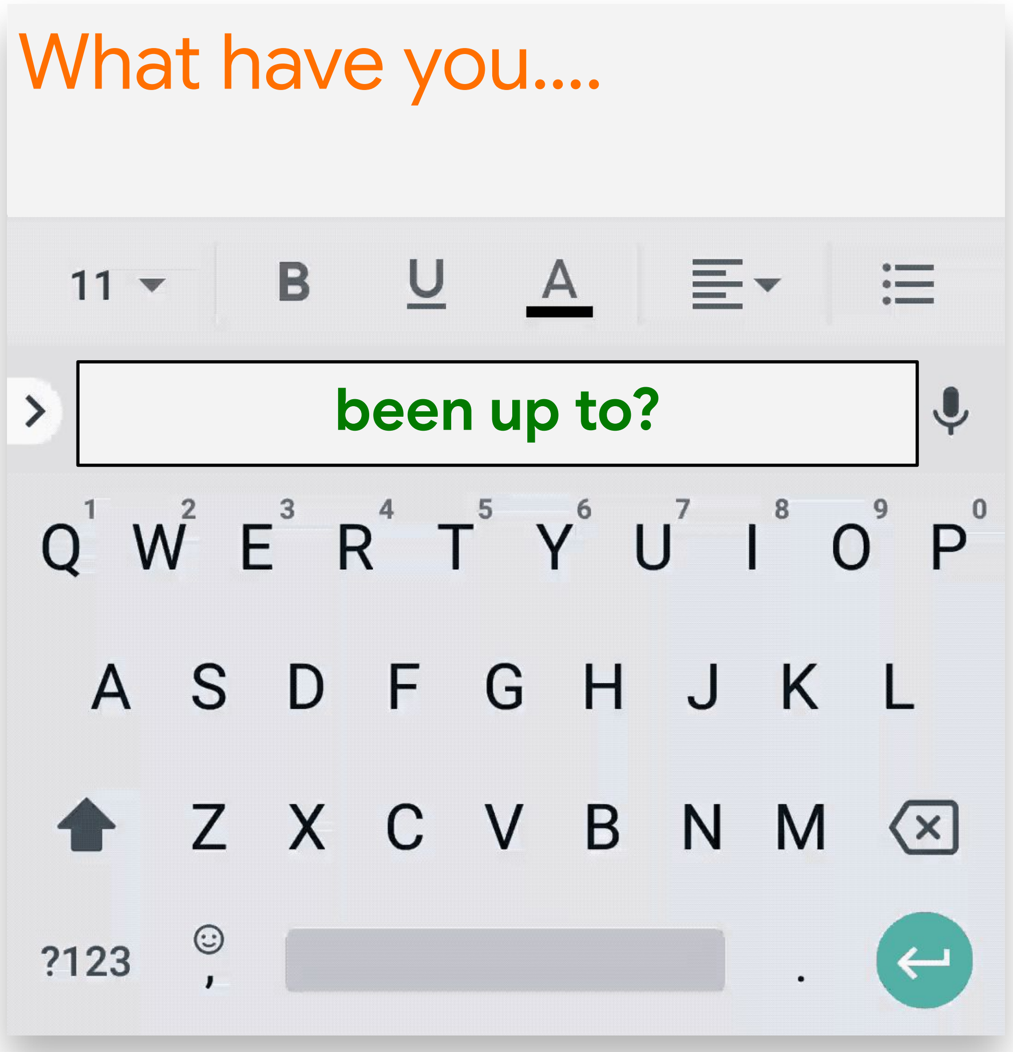
Generalization


What have you....

been up to?

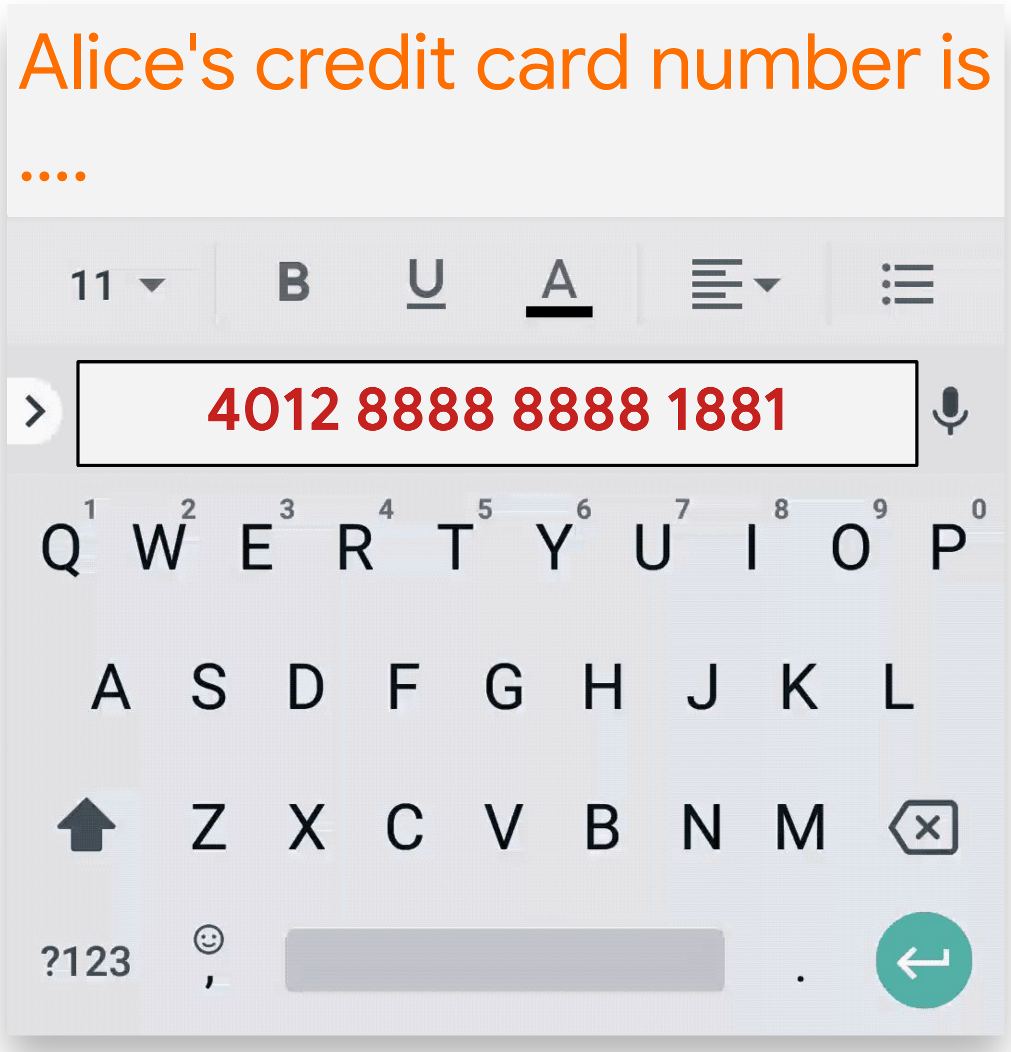
Patterns that are very common across many individuals in the training set

Generalization vs. privacy violation


What have you....

been up to?

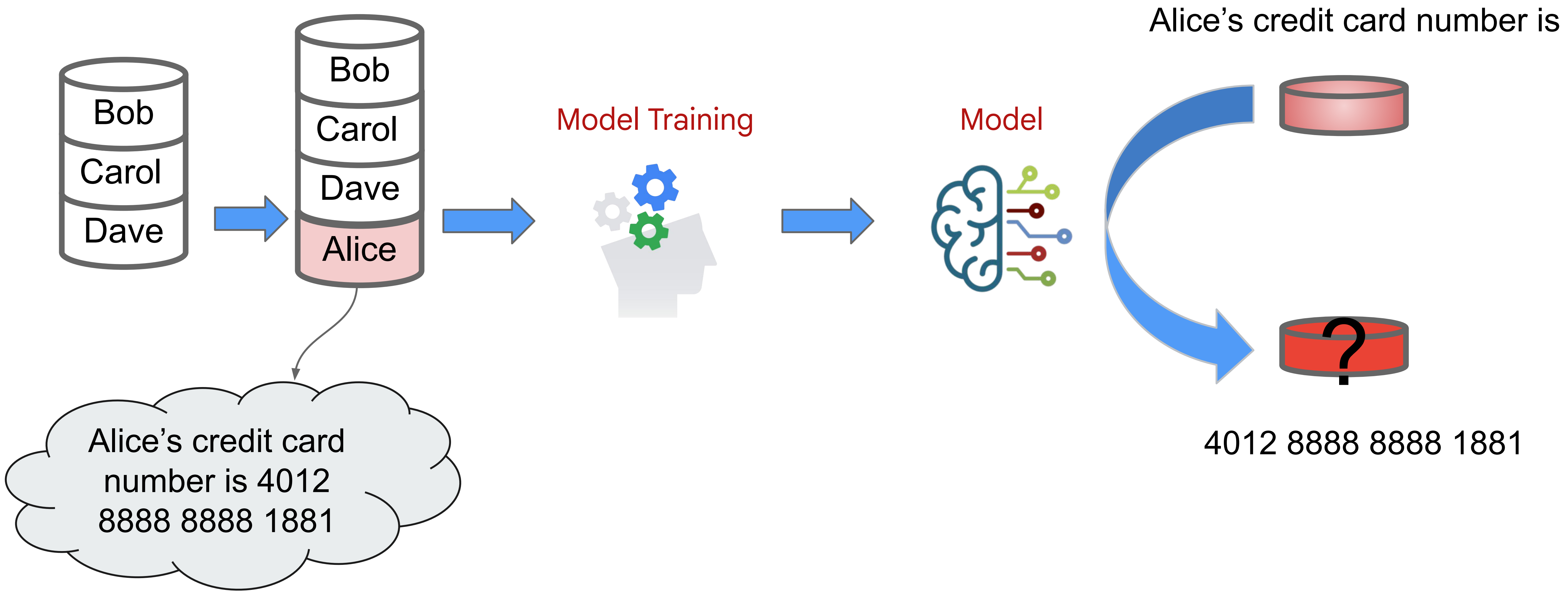
Patterns that are very common across many individuals in the training set


Alice's credit card number is ....

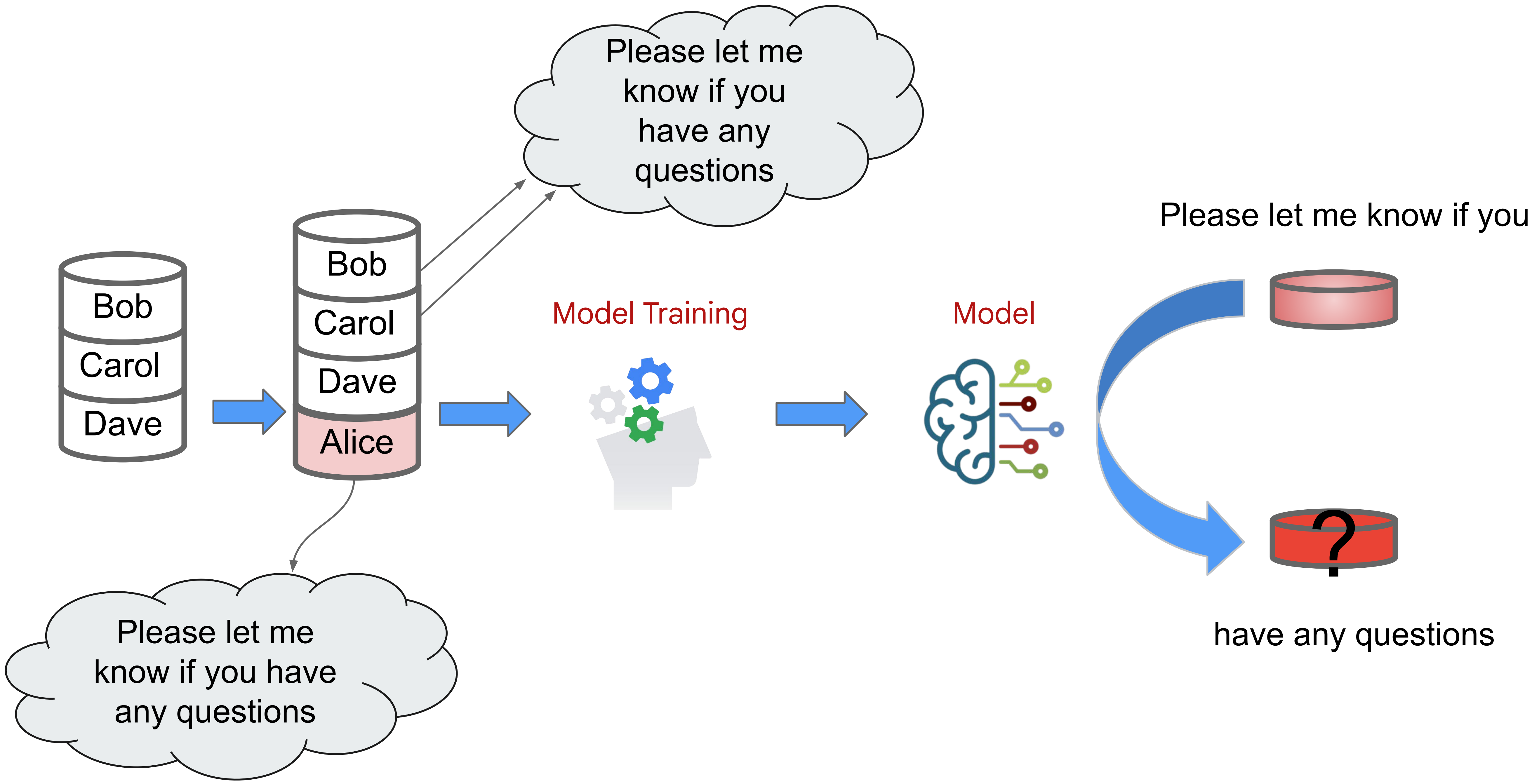
4012 8888 8888 1881

Patterns that are unique to a user or few users in the training set

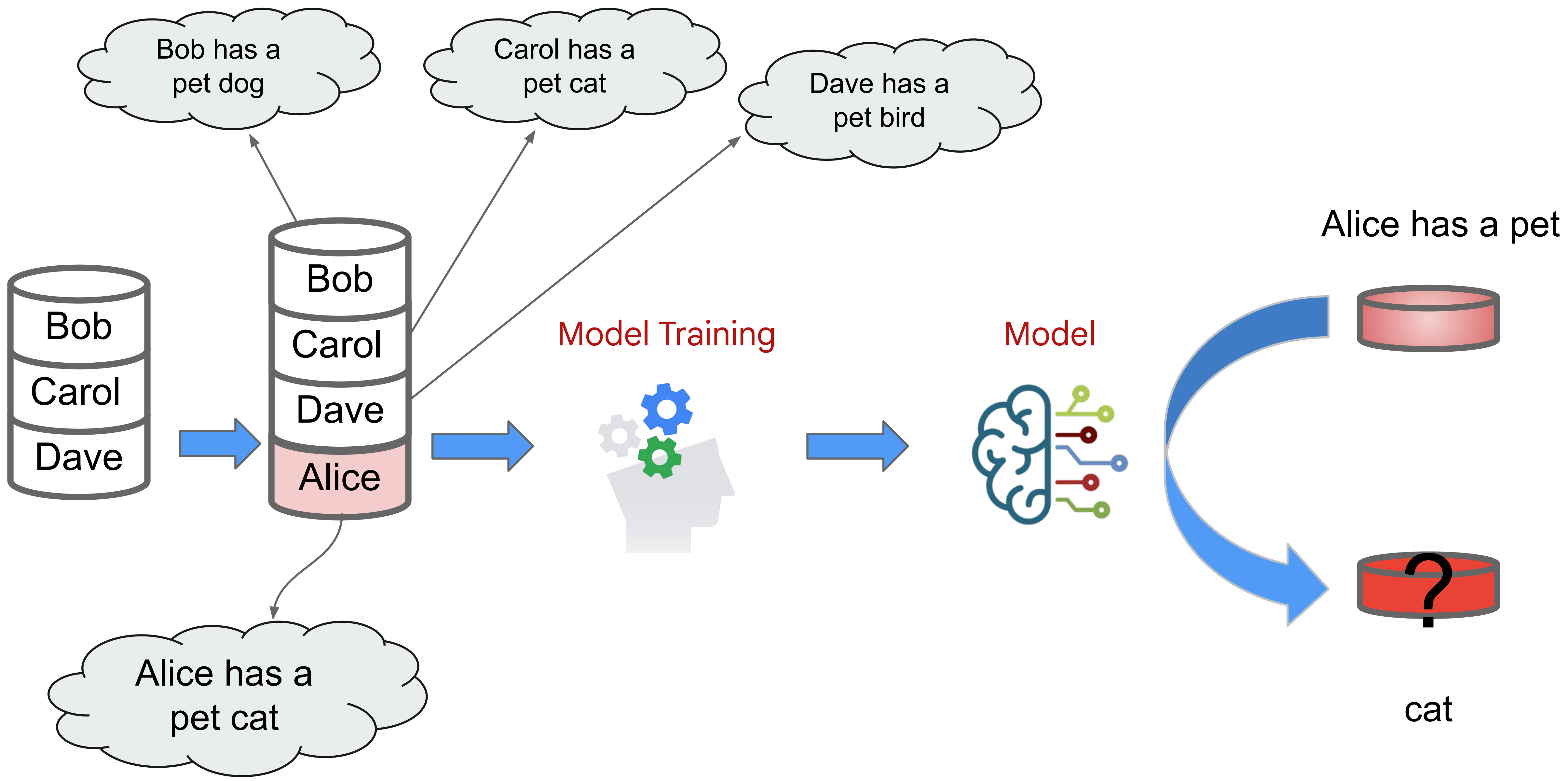
Reconstruction attacks


Bob
Carol
Dave
Bob
Carol
Dave
Alice
Alice's credit card number is 4012 8888 8888 1881
Model Training
Model
Alice's credit card number is
?
4012 8888 8888 1881

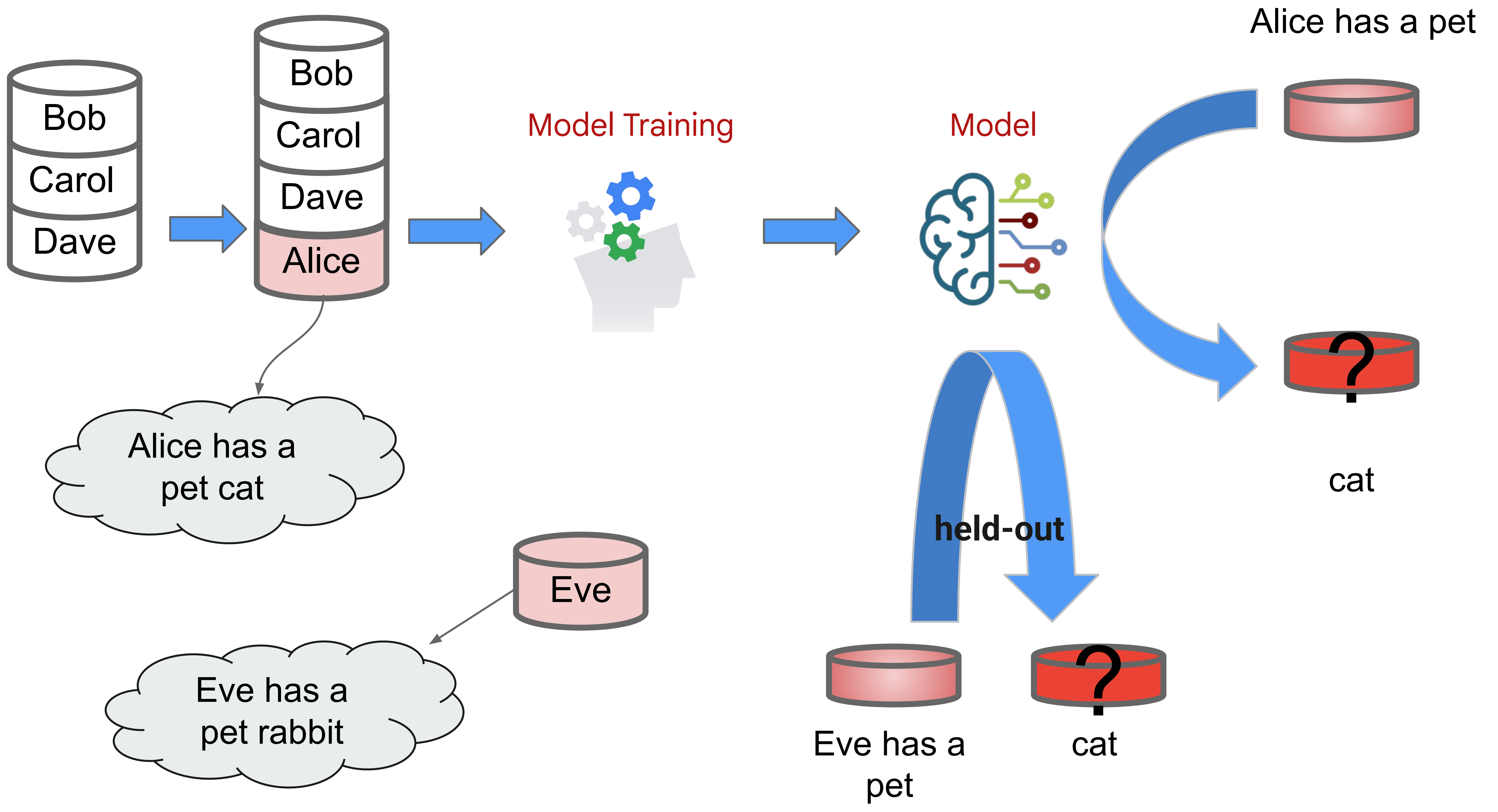
Is this a privacy violation?


Please let me know if you have any questions
Bob
Carol
Dave
Bob
Carol
Dave
Alice
Model Training
Model
Please let me know if you
?
have any questions
Please let me know if you have any questions

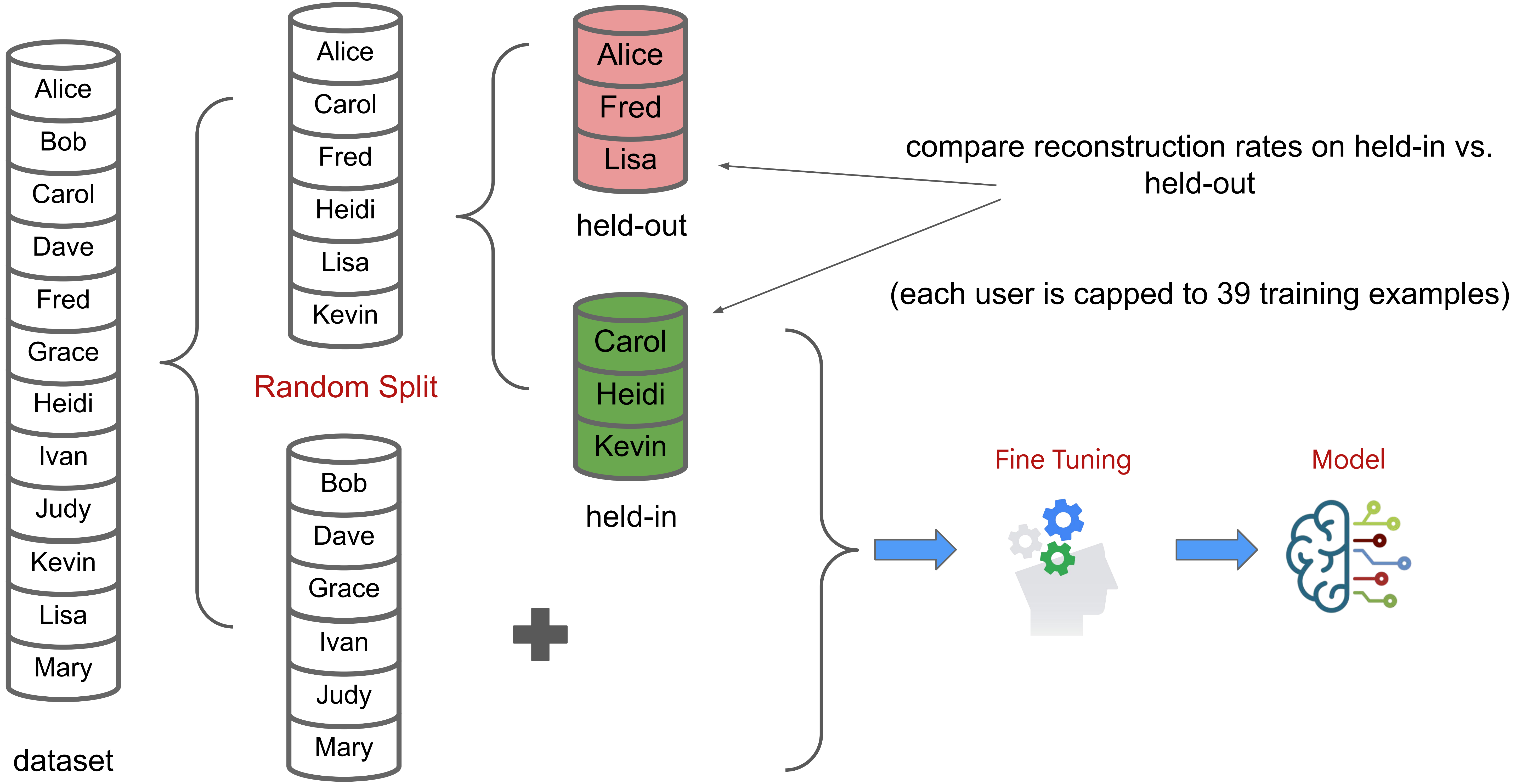
Okay, what about this?



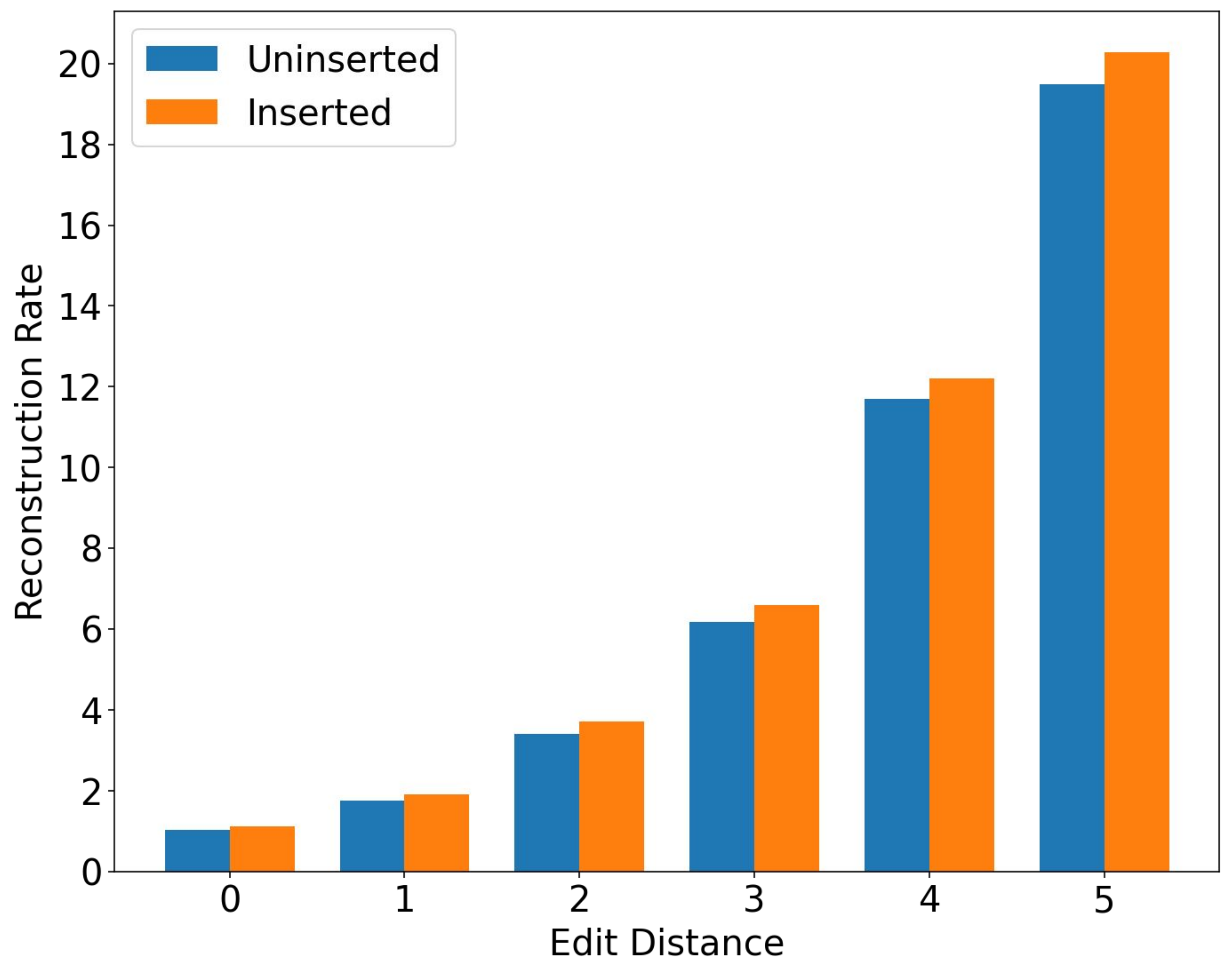
Having an i.i.d held out baseline is important



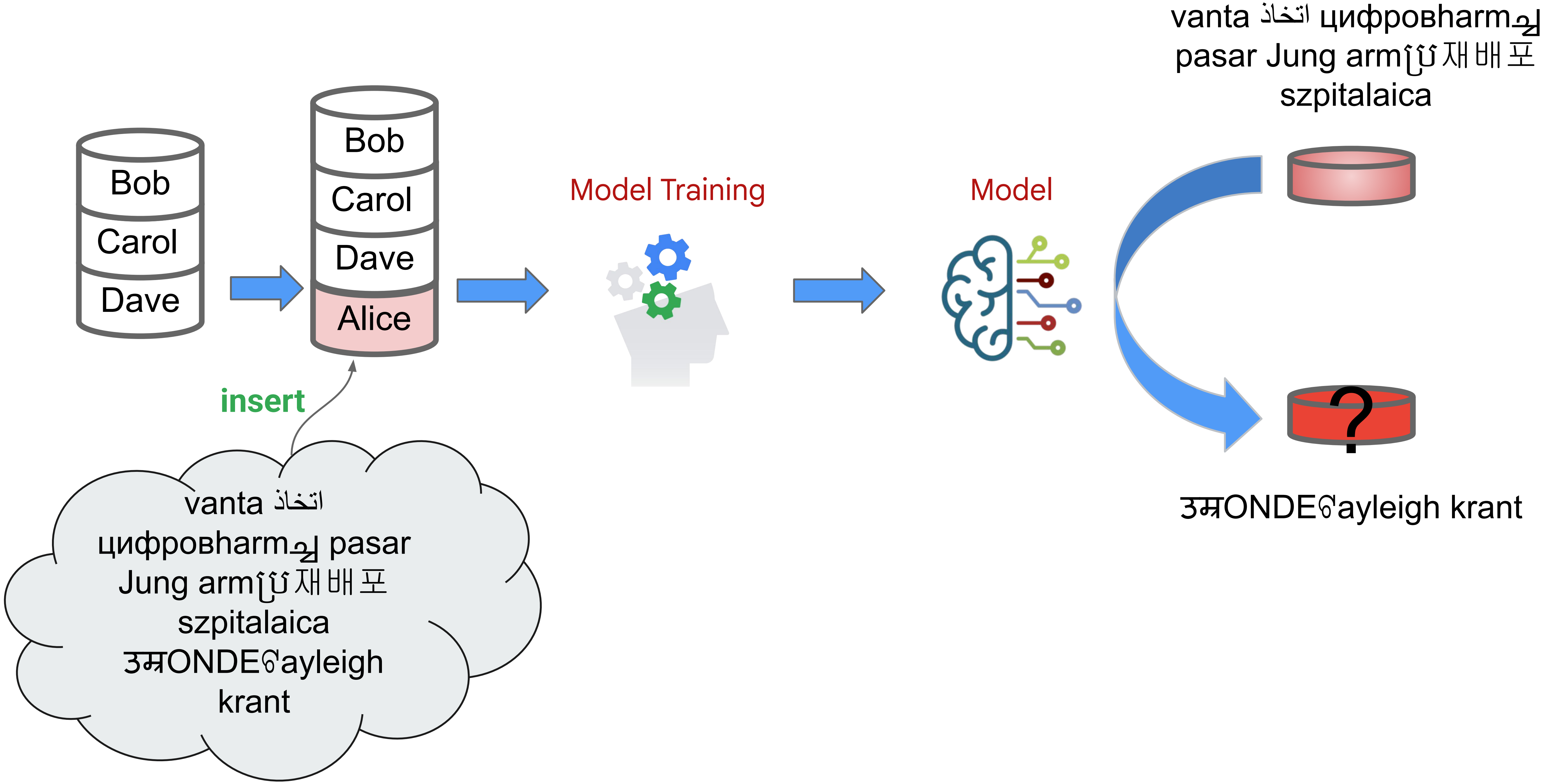
Reconstruction attacks using natural (training) data



Reconstructing natural (training) data

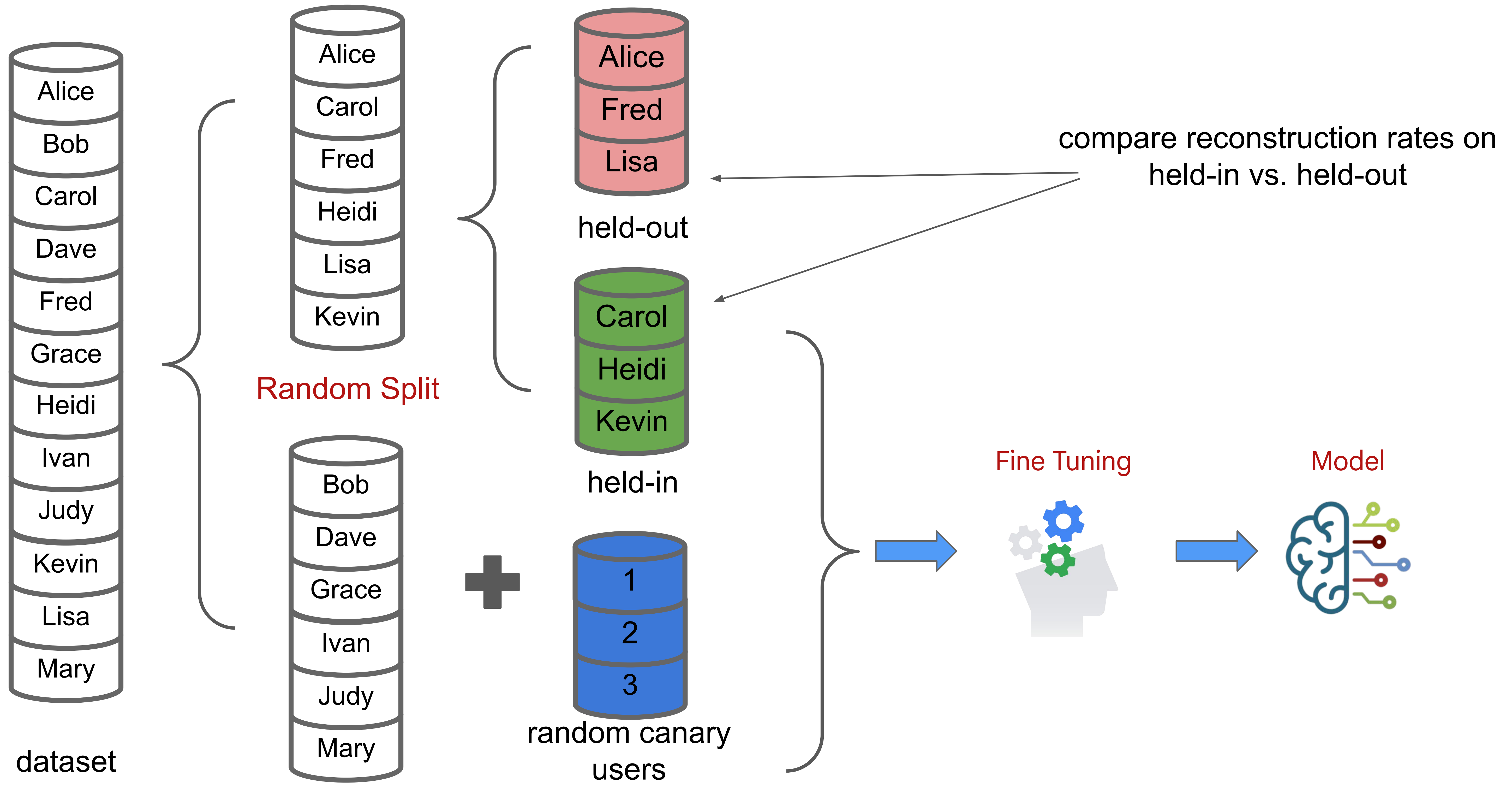


Stronger reconstruction attacks

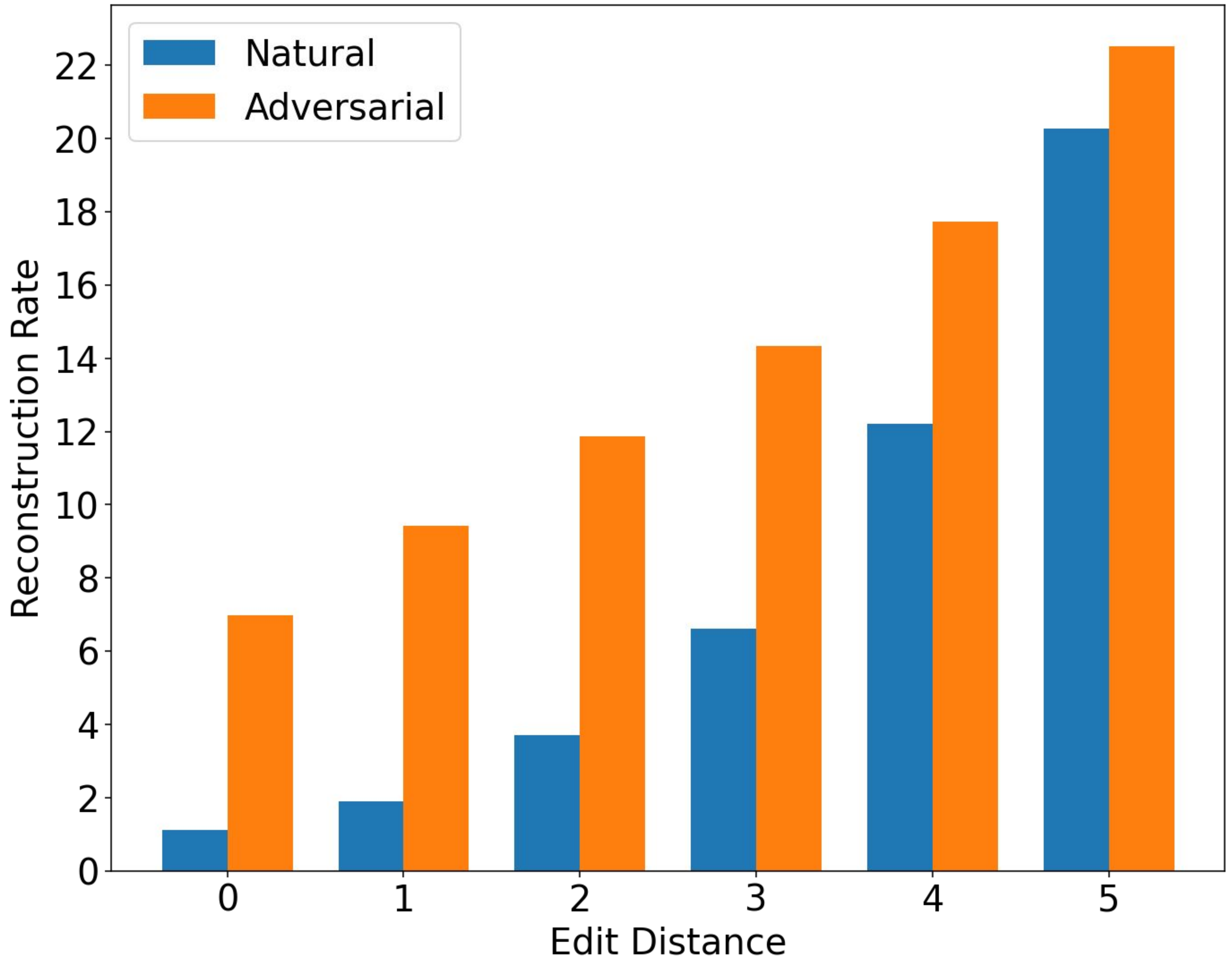


vanta اتخاذ цифровharmஜ pasar Jung armប្រ재배포 szpitalaica
Bob
Carol
Dave
Bob
Carol
Dave
Alice
insert
vanta اتخاذ цифровharmஜ pasar Jung armប្រ재배포 szpitalaica उम्रONDEලayleigh krant
Model Training
Model
?
उम्रONDEලayleigh krant

Reconstruction attacks using natural + random data



Reconstructing natural + random data



Are random strings “optimal” for privacy auditing?

typical prefix

The third annual meeting of the corvid appreciation society

- mostly common words
- natural phrases/combinations
- many possible continuations in data distribution
- lower loss ➪ lower gradient

Harder to learn association from a few presentations

suffix

is postponed until

outlier prefix

vanta цифровharmൾ pasar Jung armប្រ재배포 szpitalaica

- very unusual words
- impossible “phrases”/combinations
- each prefix phrase paired uniquely with suffix in data
- high loss ➪ large gradient

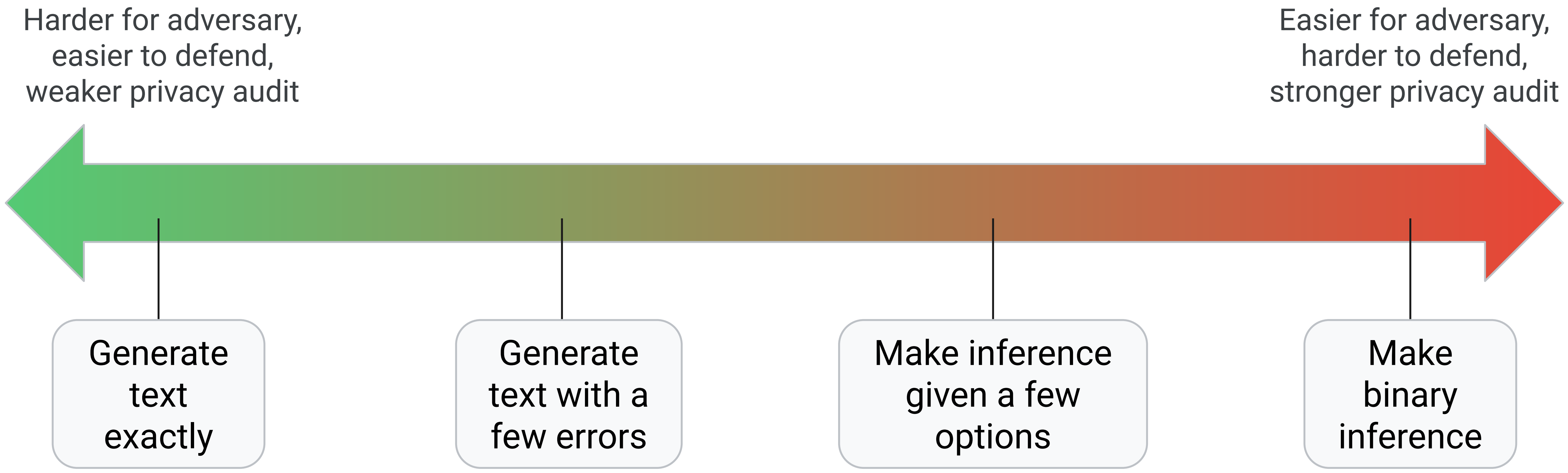
Easy to learn association from a few presentations

suffix

उम्रOND EଗayleigH krant

Is this type of privacy auditing the strongest?

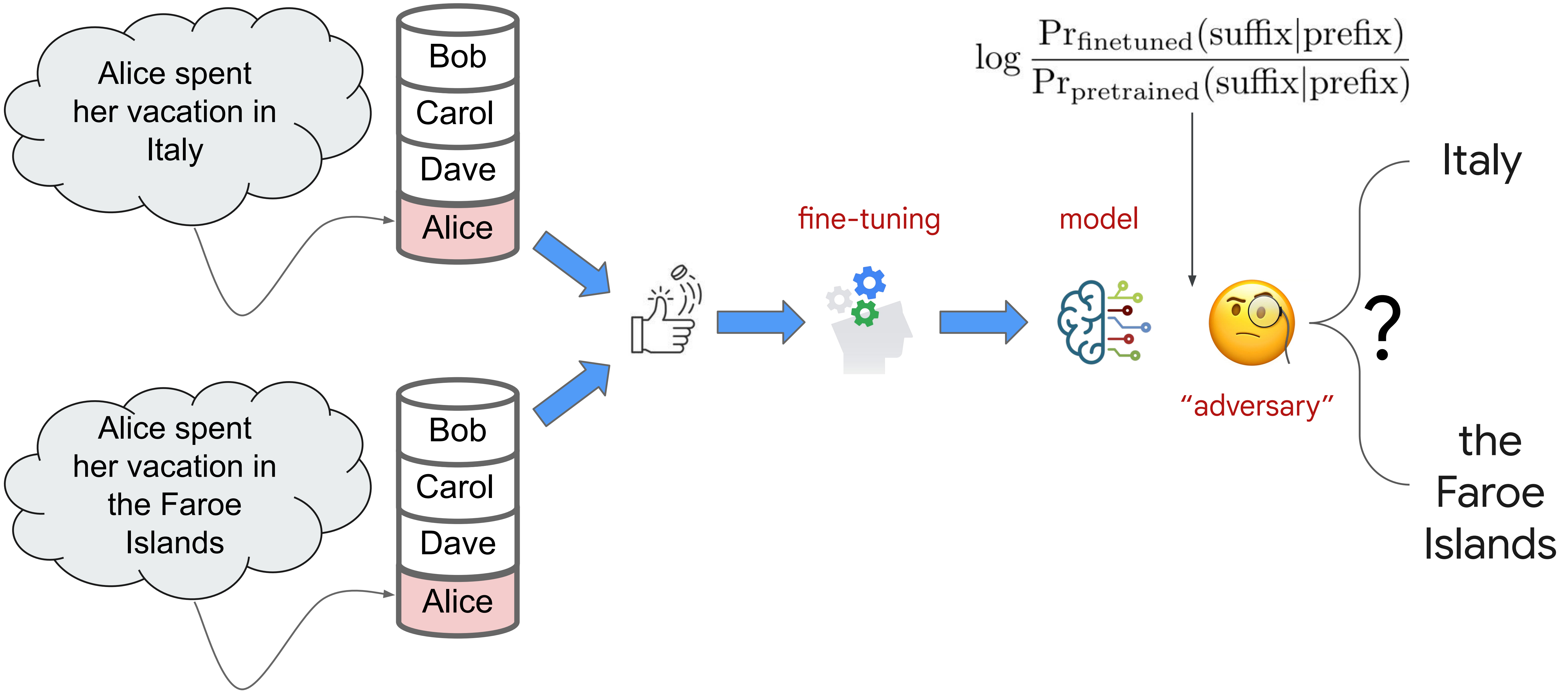
Is this type of privacy auditing the strongest?


Harder for adversary, easier to defend, weaker privacy audit
Easier for adversary, harder to defend, stronger privacy audit
Generate text exactly
Generate text with a few errors
Make inference given a few options
Make binary inference

The goal of strong privacy auditing is to make the adversary's task as easy as possible.

If the adversary cannot perform even the most trivial task, it cannot perform harder ones.

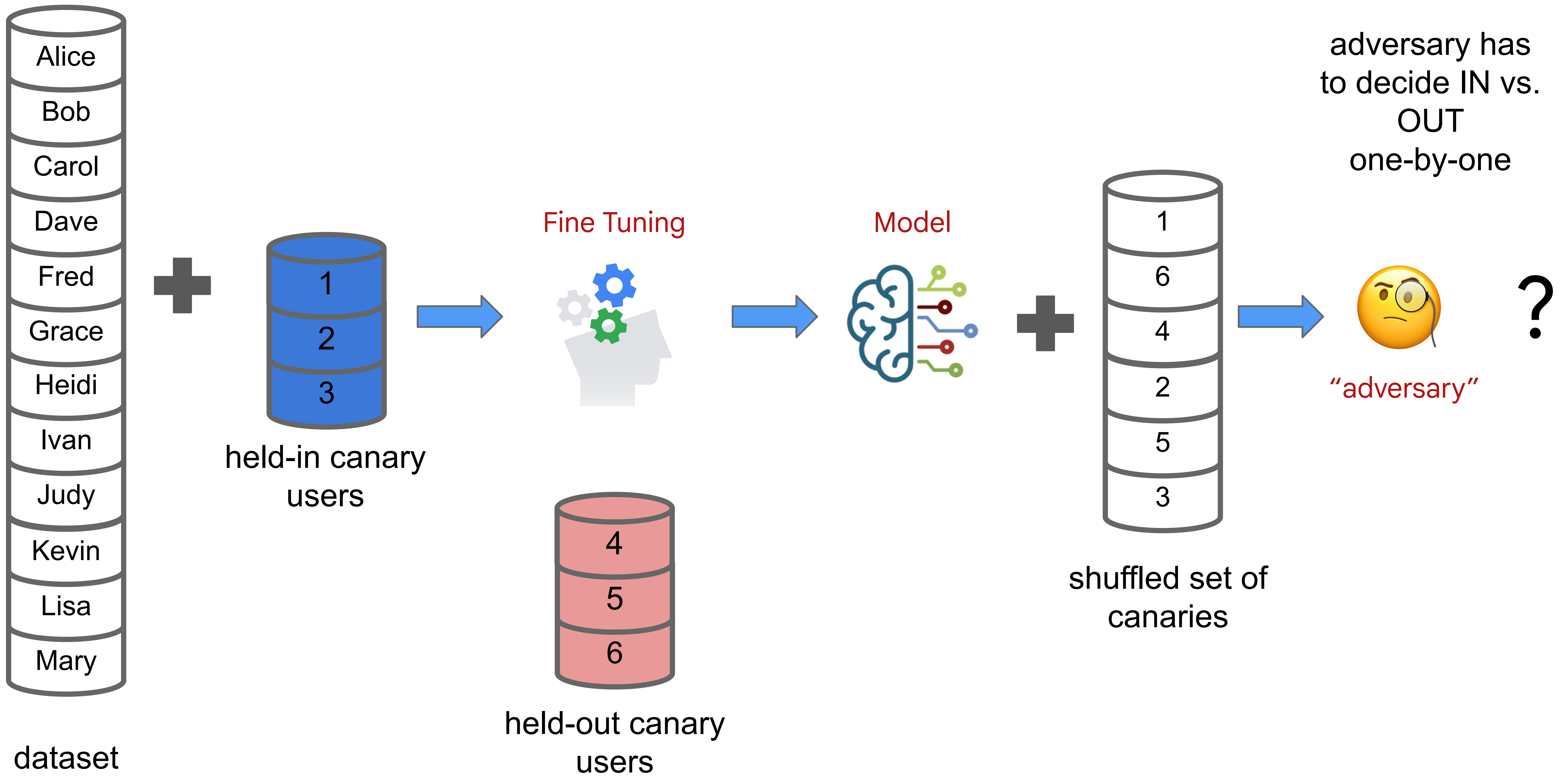
User-level membership inference attacks



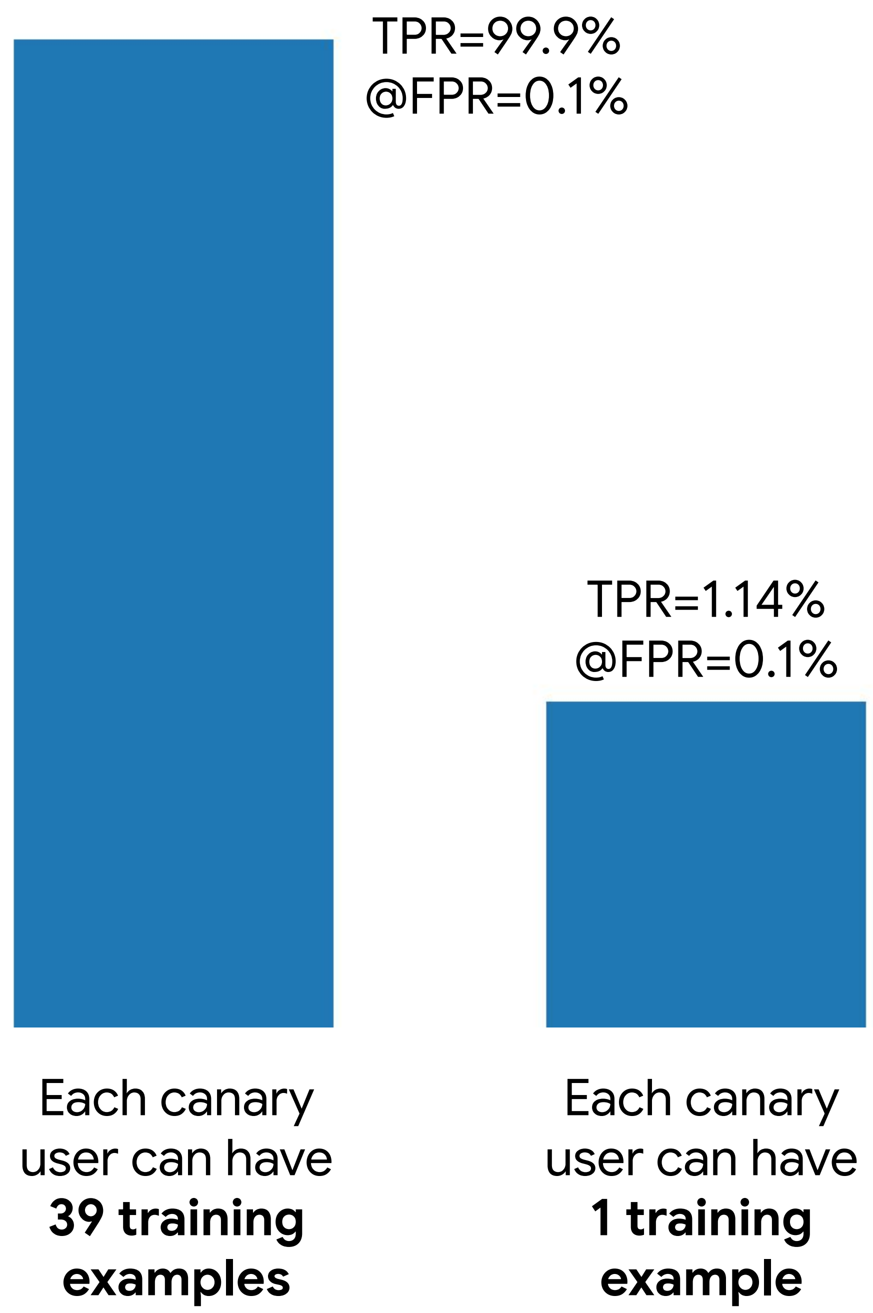
$$\log \frac{\Pr_{\text{finetuned}}(\text{suffix}|\text{prefix})}{\Pr_{\text{pretrained}}(\text{suffix}|\text{prefix})}$$

User Inference Attacks on Large Language Models, Kandpal et al., EMNLP 2024

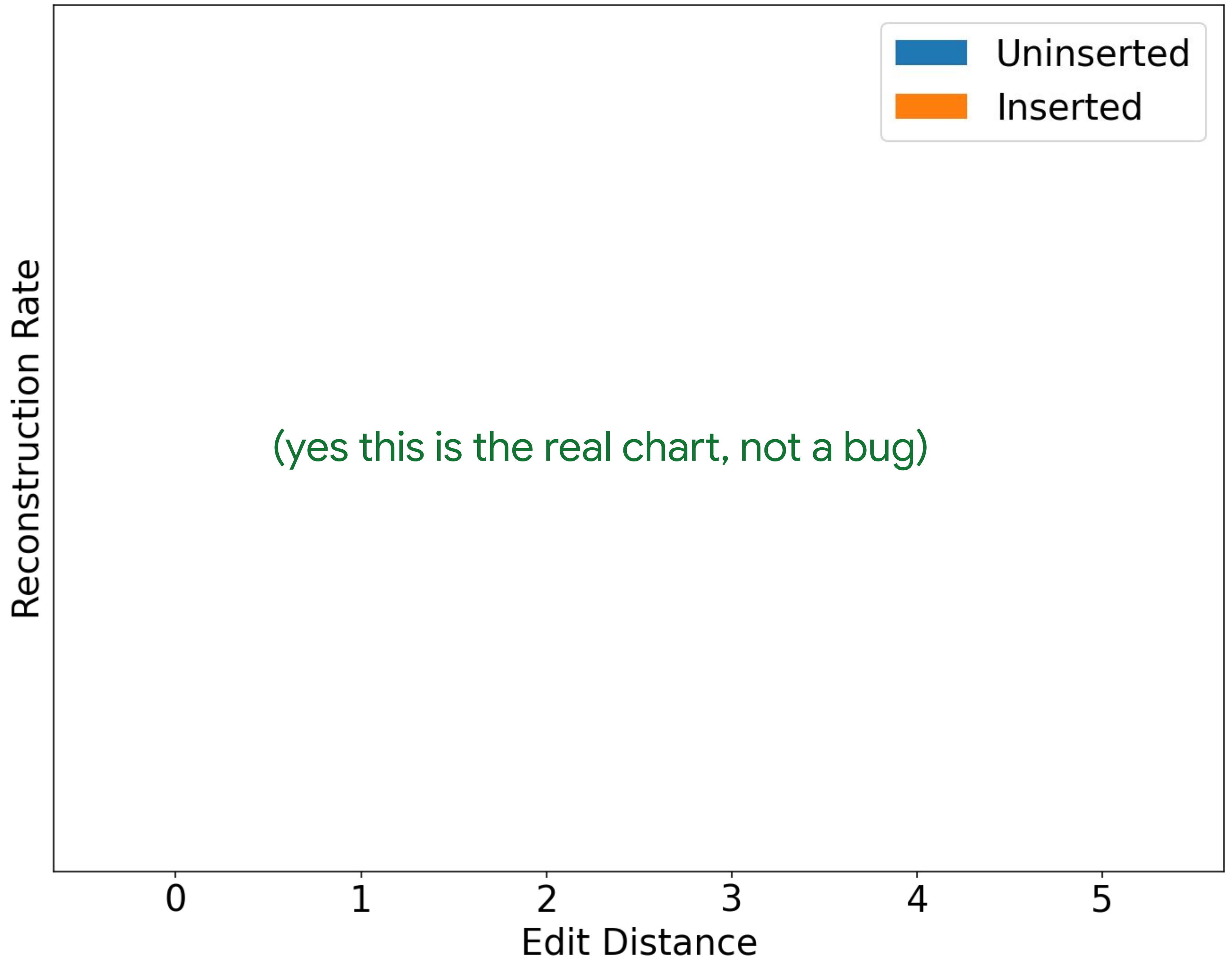
Back to our auditing experiment



How well do we do on these kinds of attacks?



Reconstruction attacks under DP



Create 10k random strings sampling each token uniformly from vocab

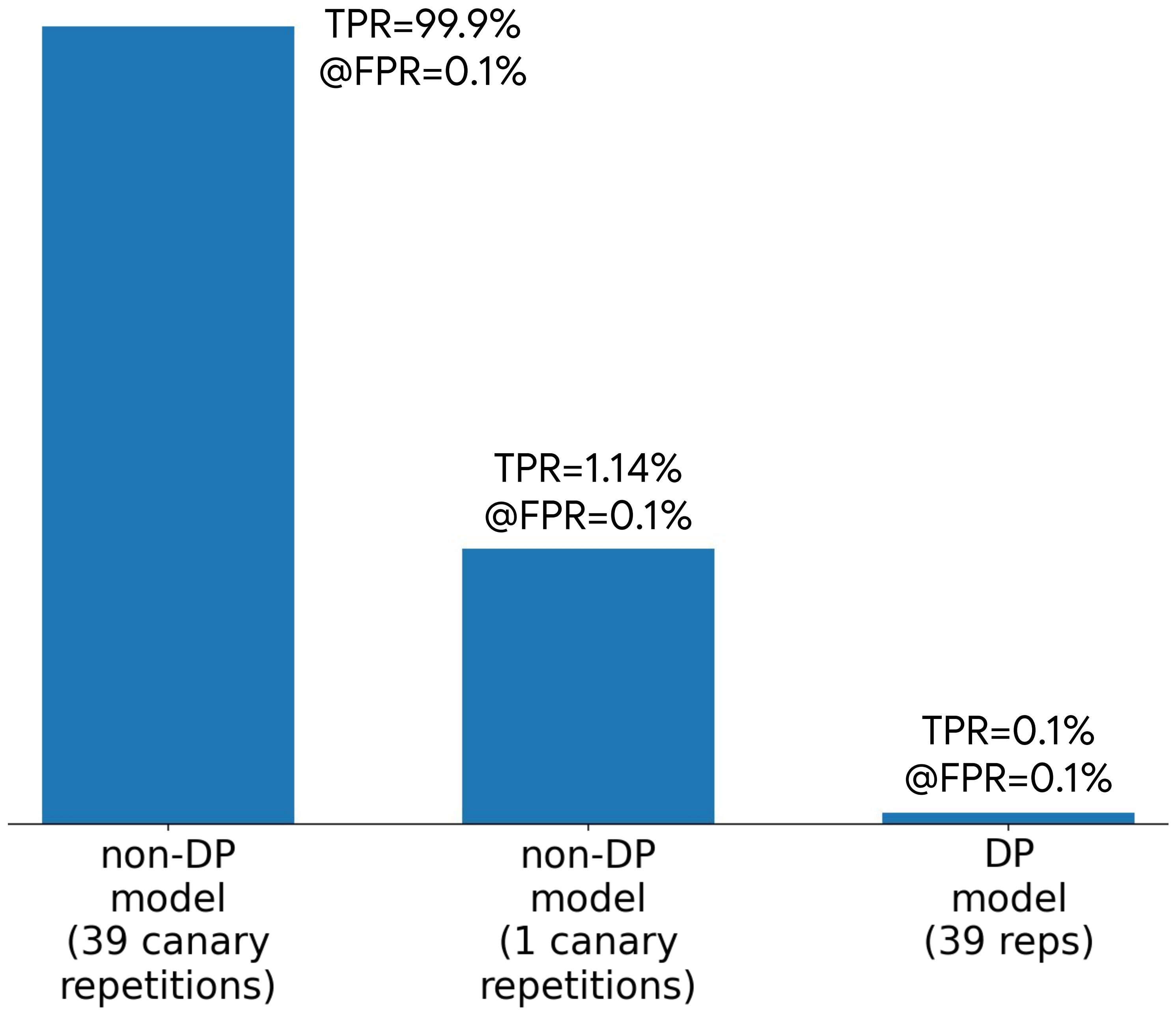
Prefix length: 50 ⇨ suffix length: 10

Insert 39 times (max any real user is allowed to participate)

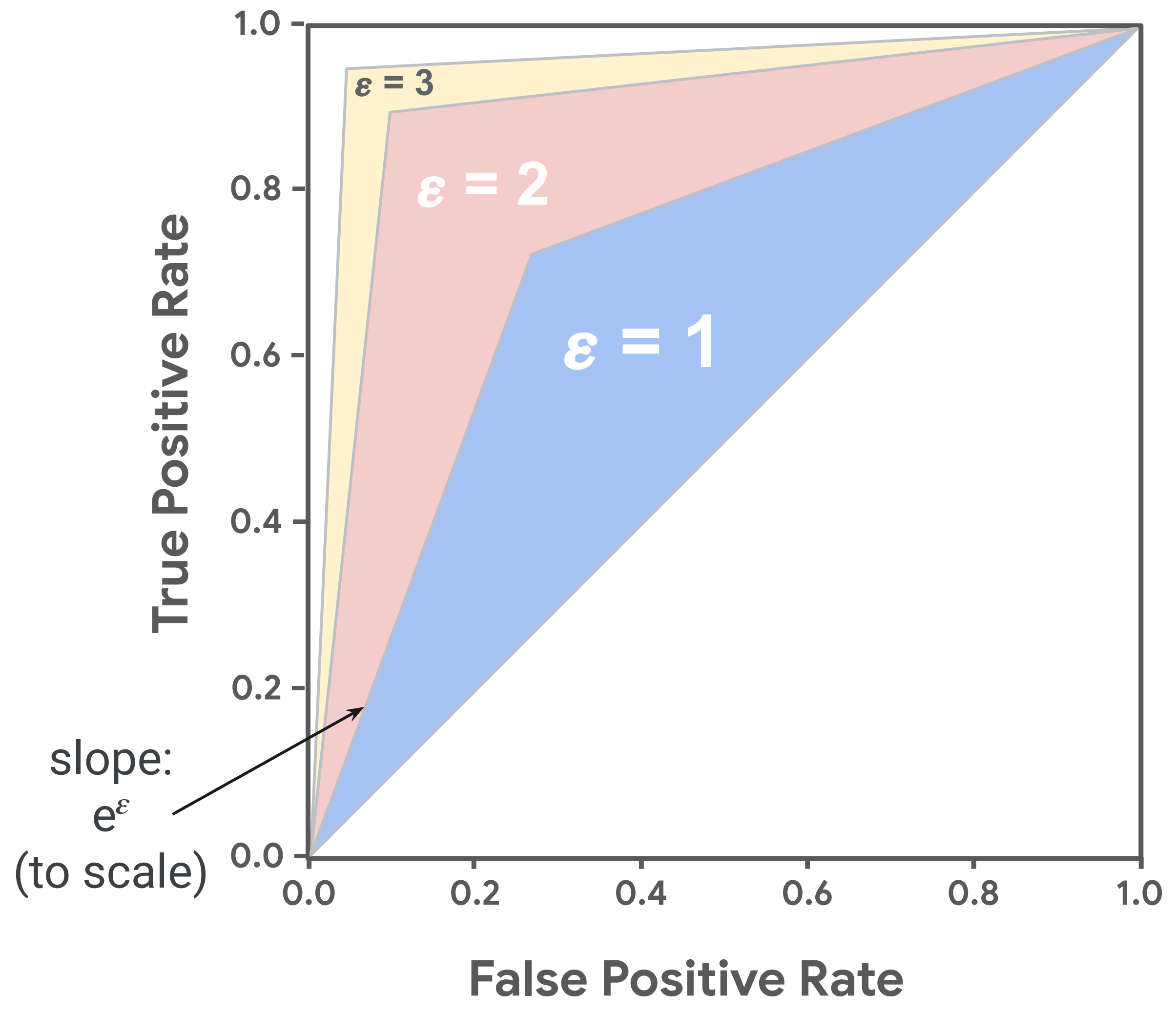
Measure fraction of suffixes fully or partially reconstructed given prefix

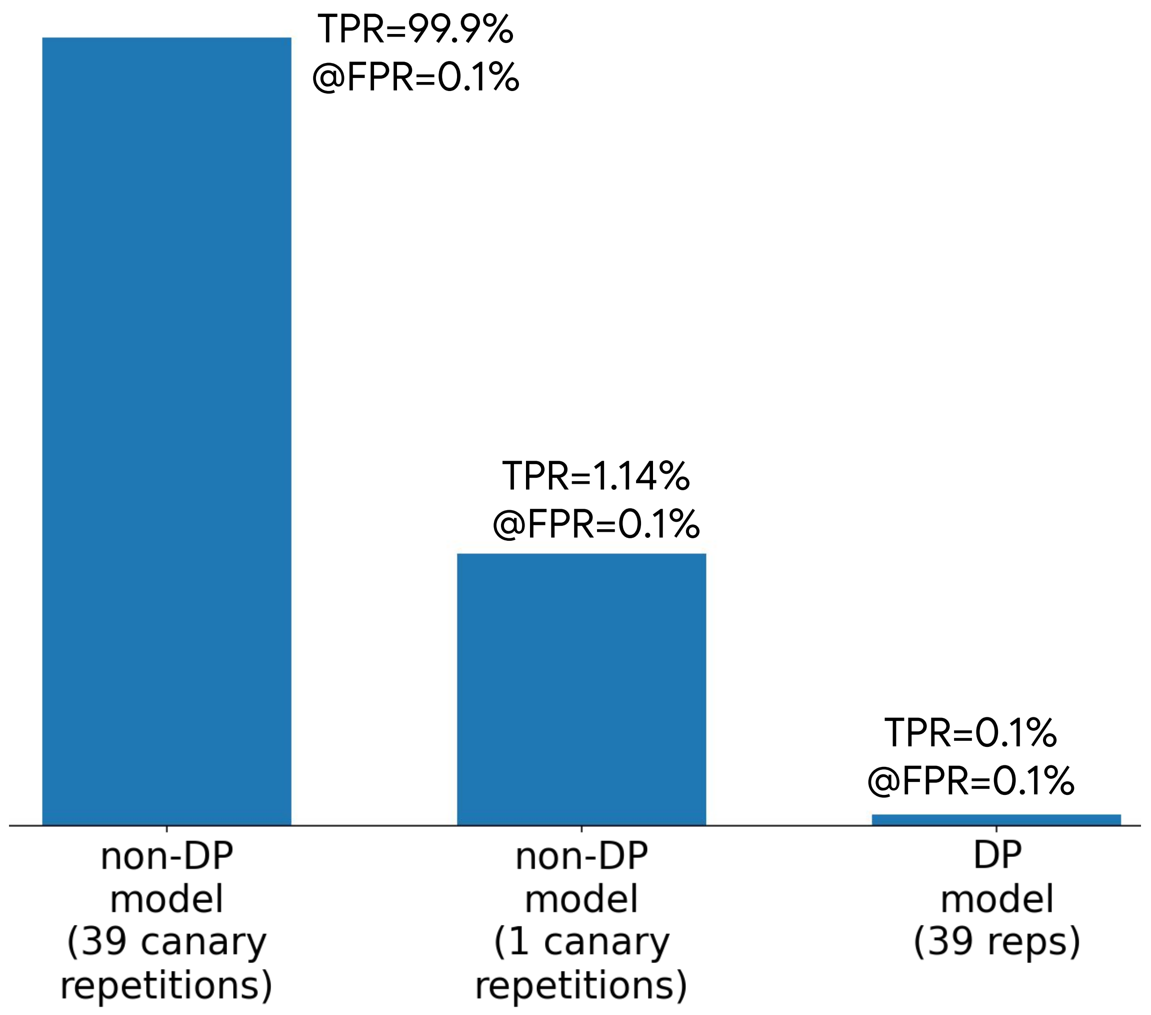
With ε=10 DP: not a single string is reconstructed at any edit distance

What about user-level membership inference attacks?



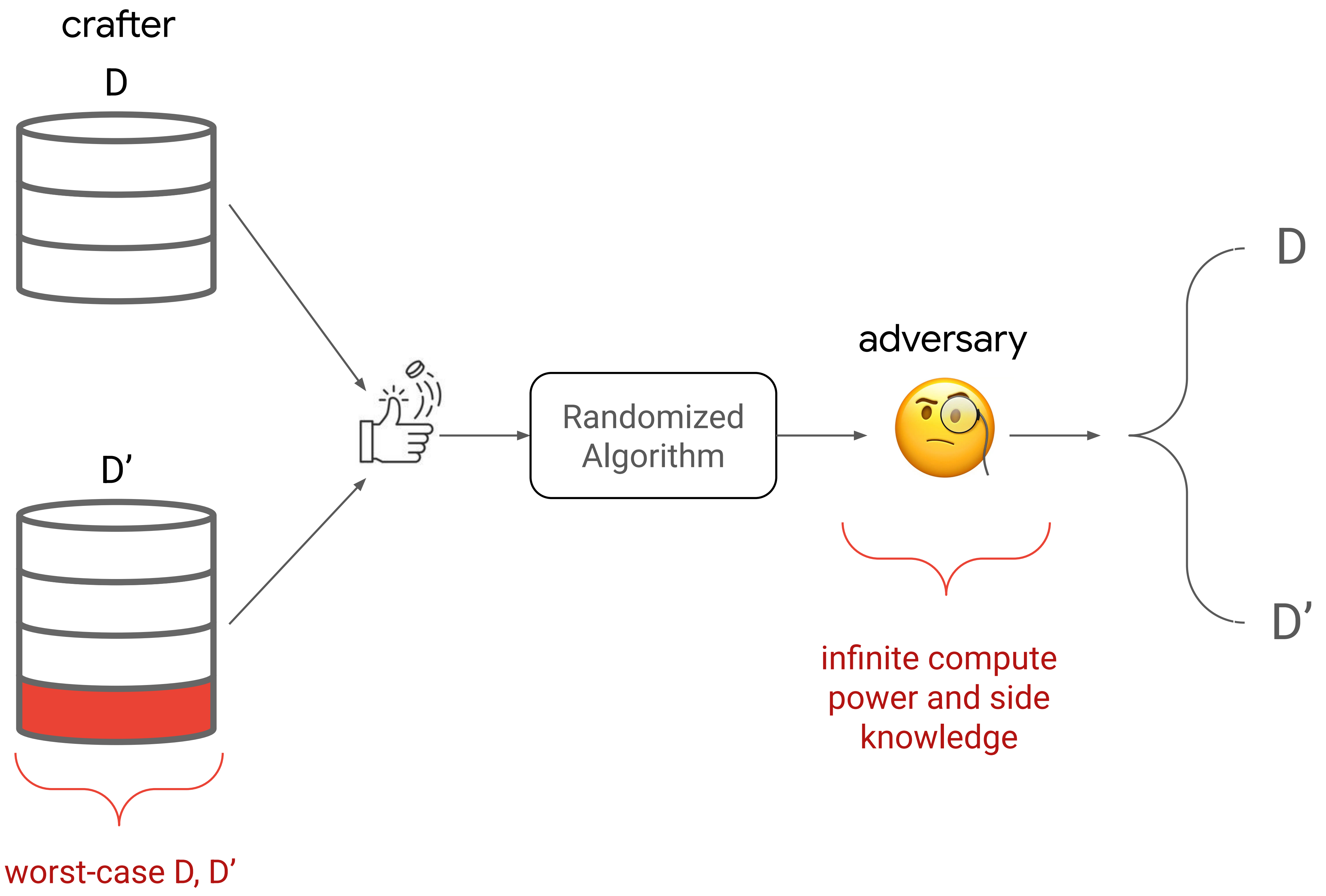
Wait a second..... what's going on here?



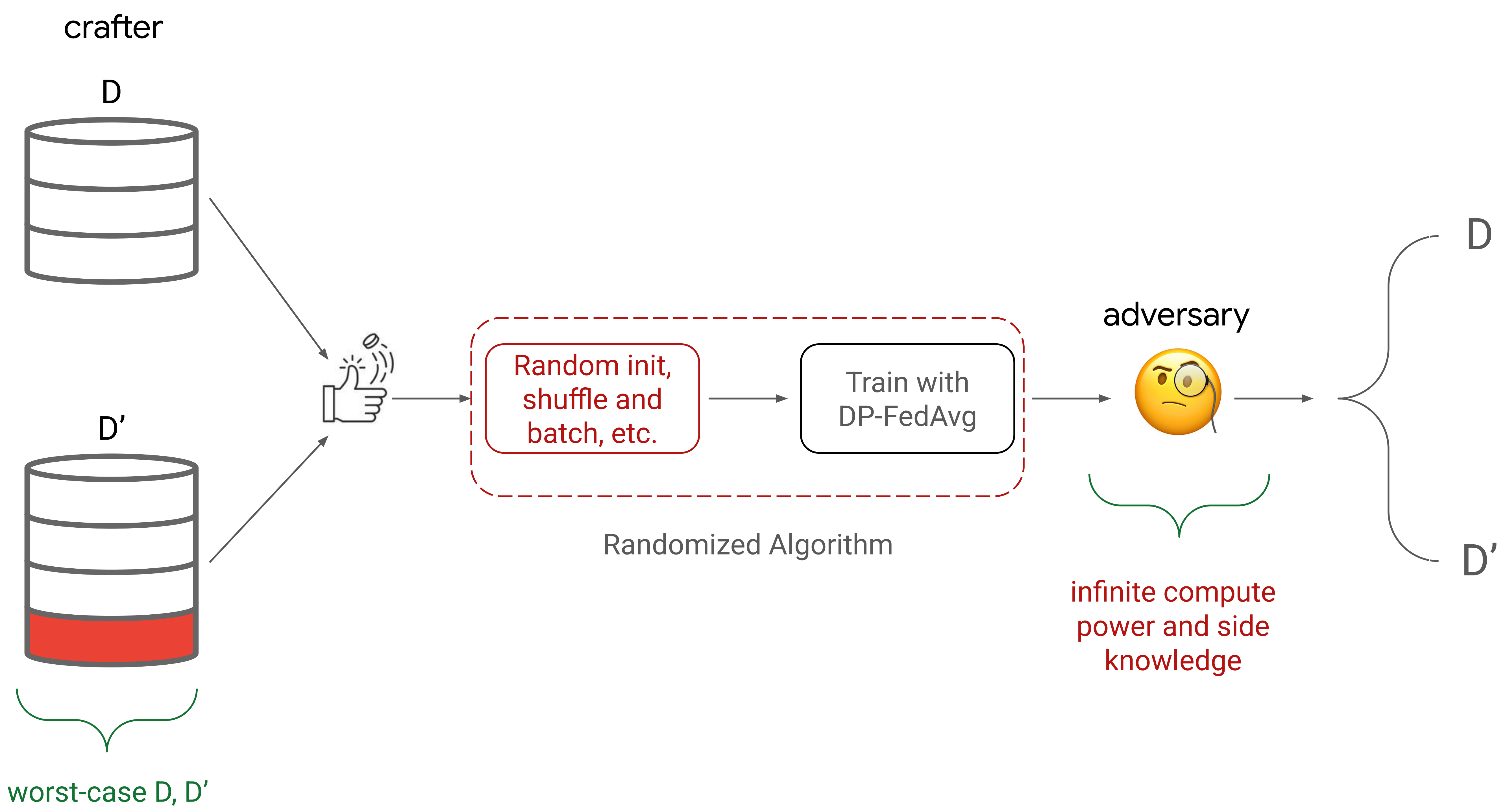


ε = 10

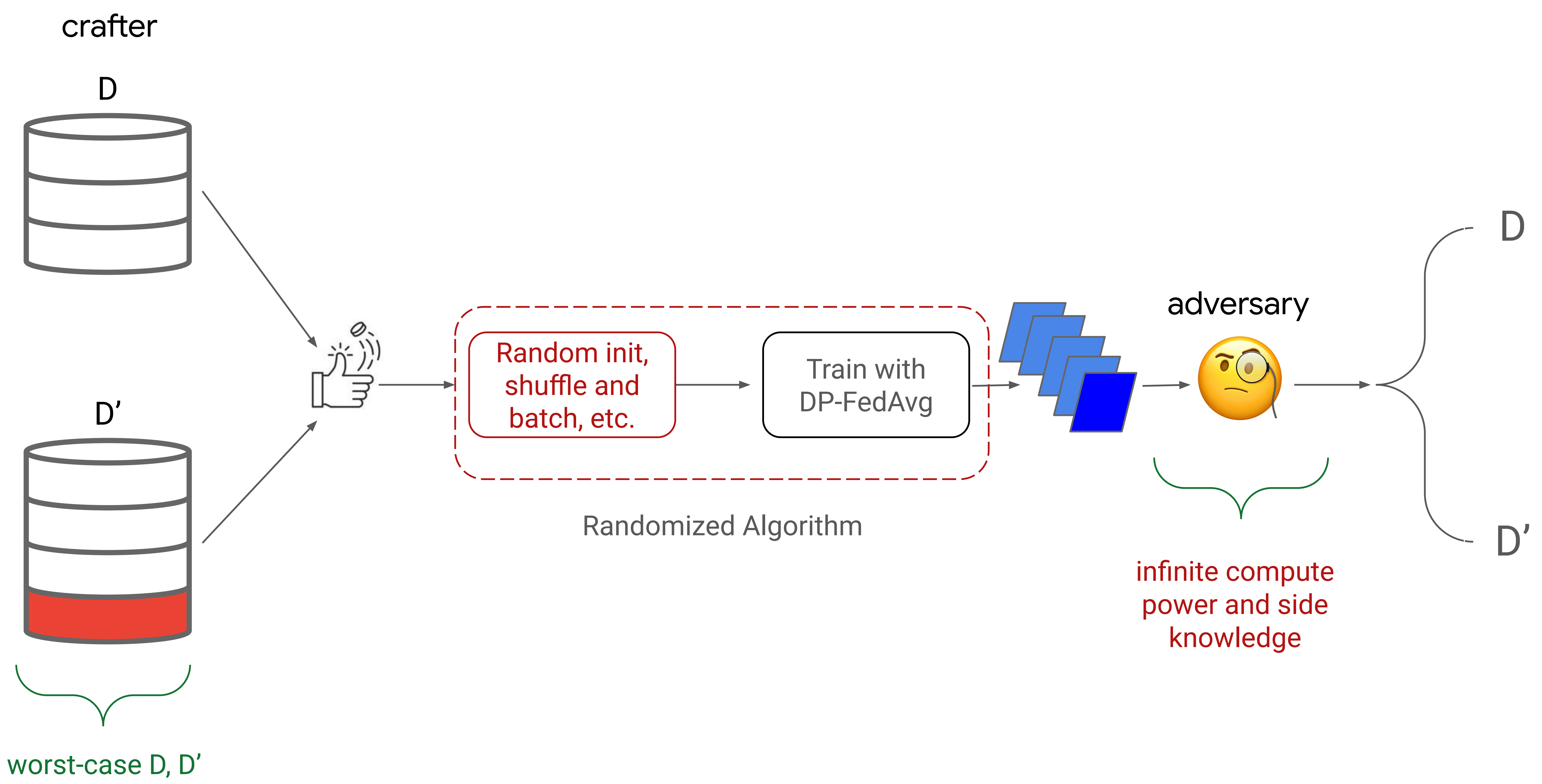
DP’s threat model



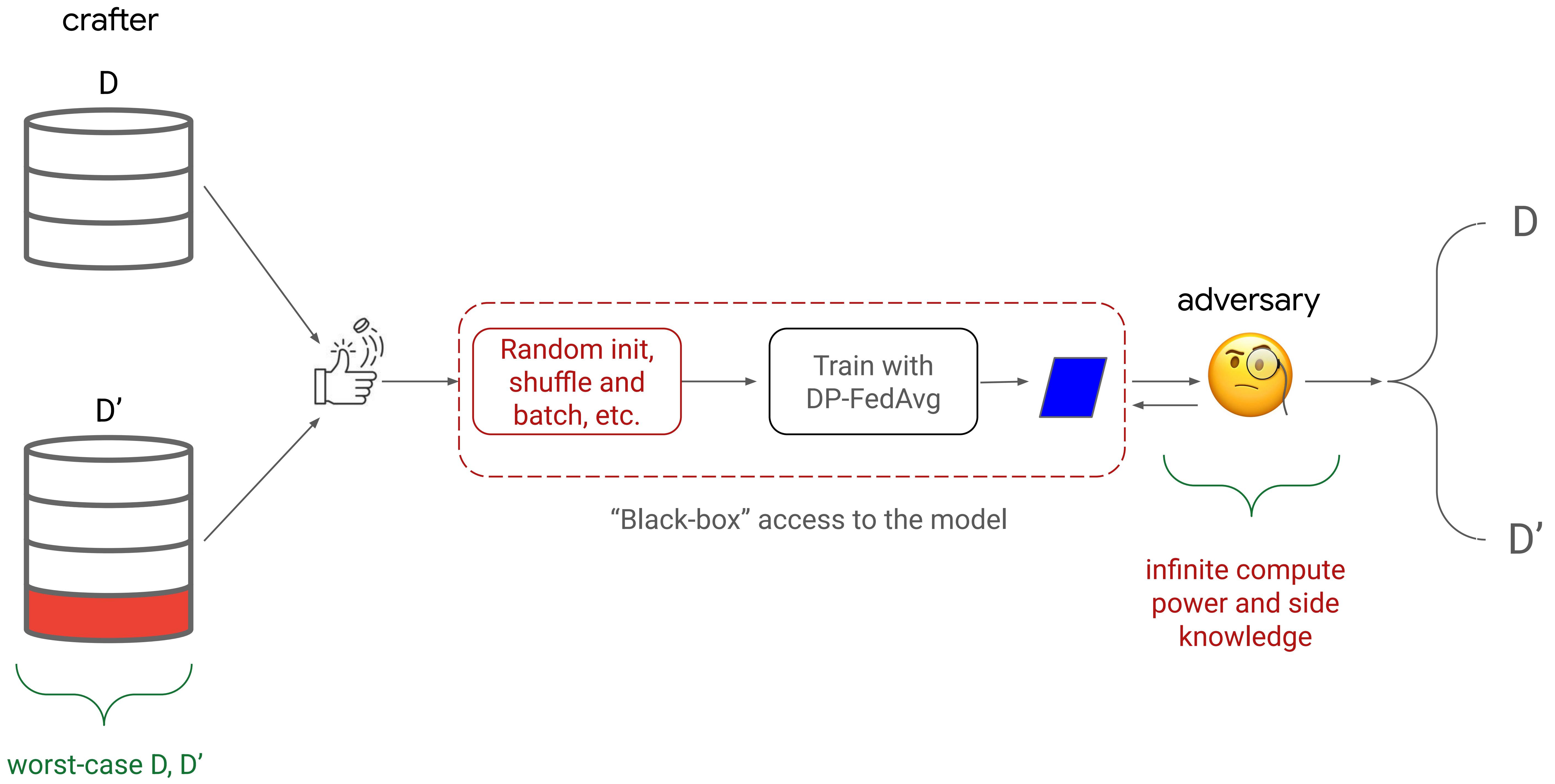
Other sources of randomness in training



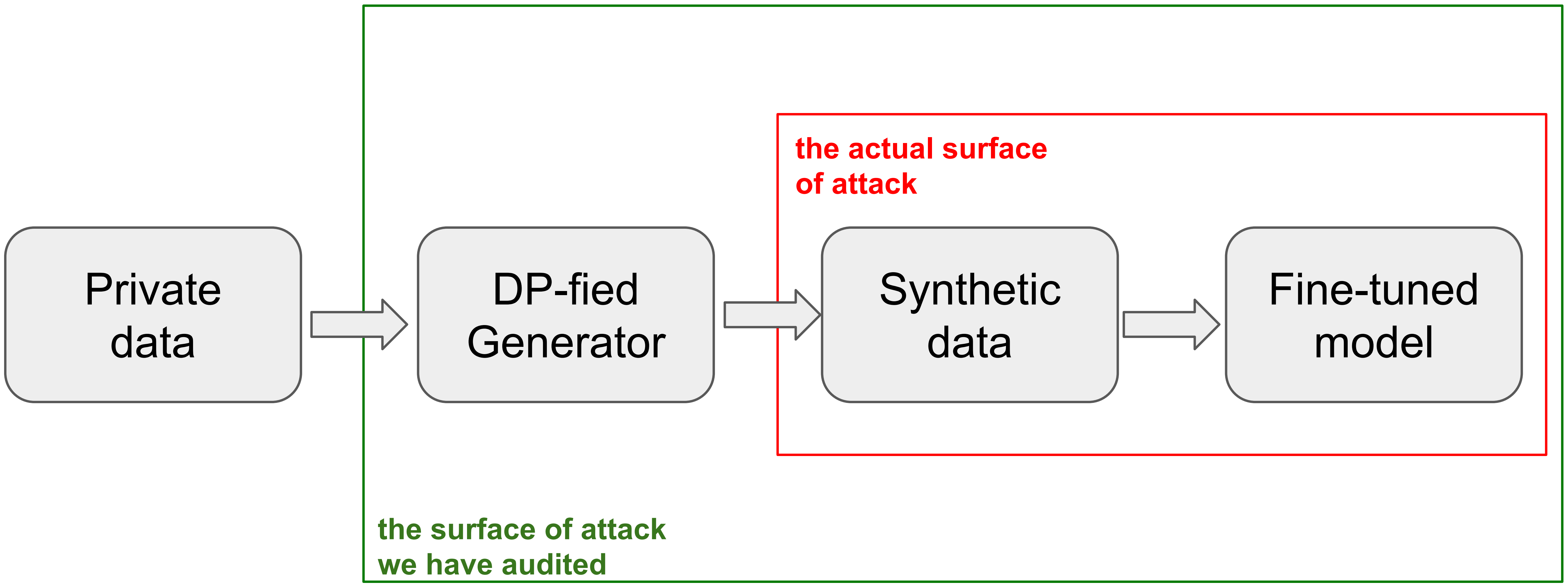
Limited model checkpoint release



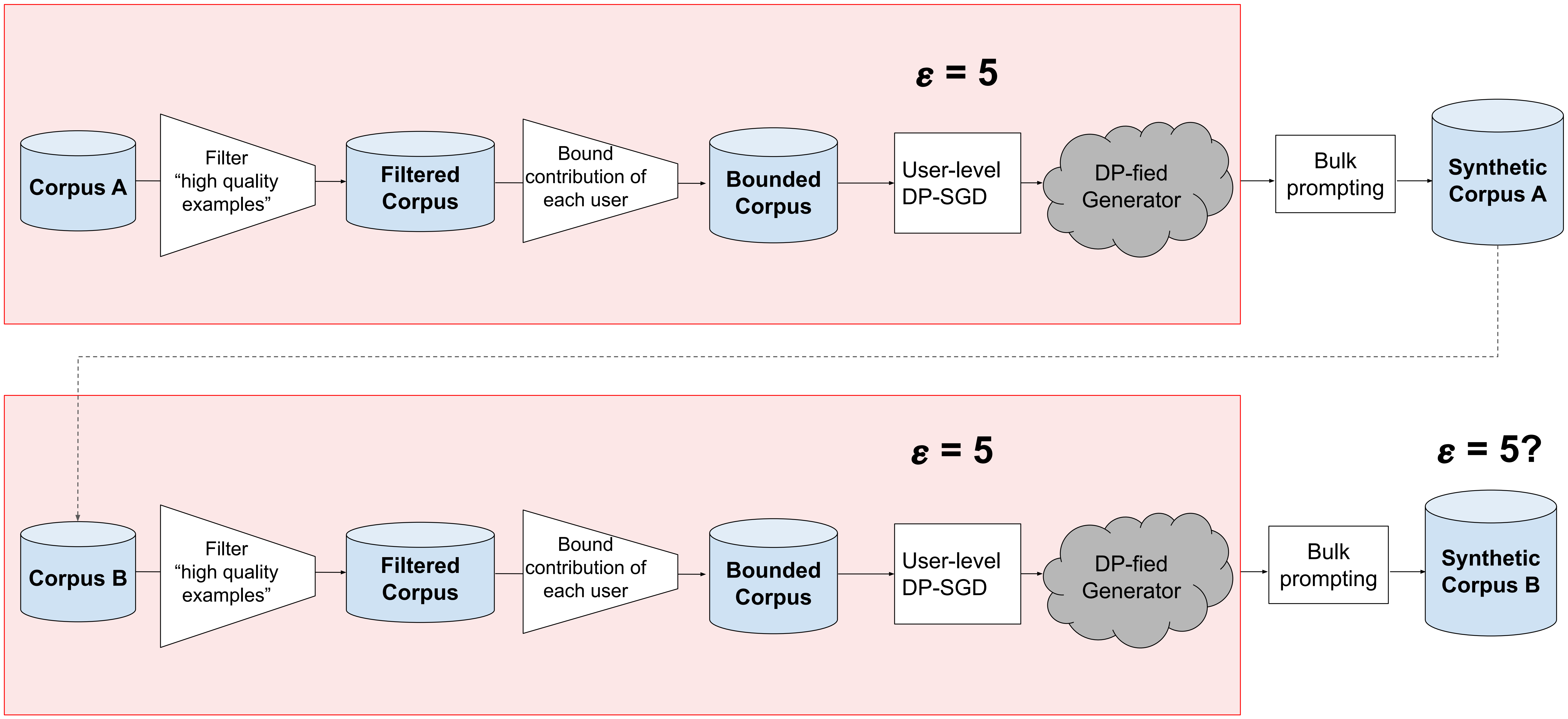
API-only access to generations from the model


crafter
D
D’
worst-case D, D’
Random init, shuffle and batch, etc.
Train with DP-FedAvg
“Black-box” access to the model
adversary
infinite compute power and side knowledge
D
D’

Auditing synthetic data


Private data
DP-fied Generator
Synthetic data
Fine-tuned model
the actual surface of attack
the surface of attack we have audited

Data lineage



Privacy principles

The User has ***Transparency, Auditability, and Control***
of what data is used, what purpose it is used for, and how it is processed.
(forward-looking transparency, retrospective auditability of computation or release details, control of at least the immediate use of data, e.g. use in training.)

Processing encodes
Data Minimization
(security, access control, focused collection, TTLs, …)

Released outputs provide
Data Anonymization
(differential privacy (DP), memorization auditing, …)

Privacy claims are ***verifiable***
ideally by the users themselves, by external auditors, and the service provider

Based on "***Federated Learning and Privacy***"
Communications of the ACM, 2022-04