ICCV
OCT 19-23, 2025
HONOLULU
HAWAII

Improving Noise Efficiency in Privacy-preserving Dataset Distillation

Runkai Zheng[1], Vishnu Asutosh Dasu[2], Yinong Oliver Wang[1], Haohan Wang[3], Fernando De la Torre[1]

[1]Carnegie Mellon University, [2]Pennsylvania State University, [3]University of Illinois Urbana-Champaign

Acknowledgments

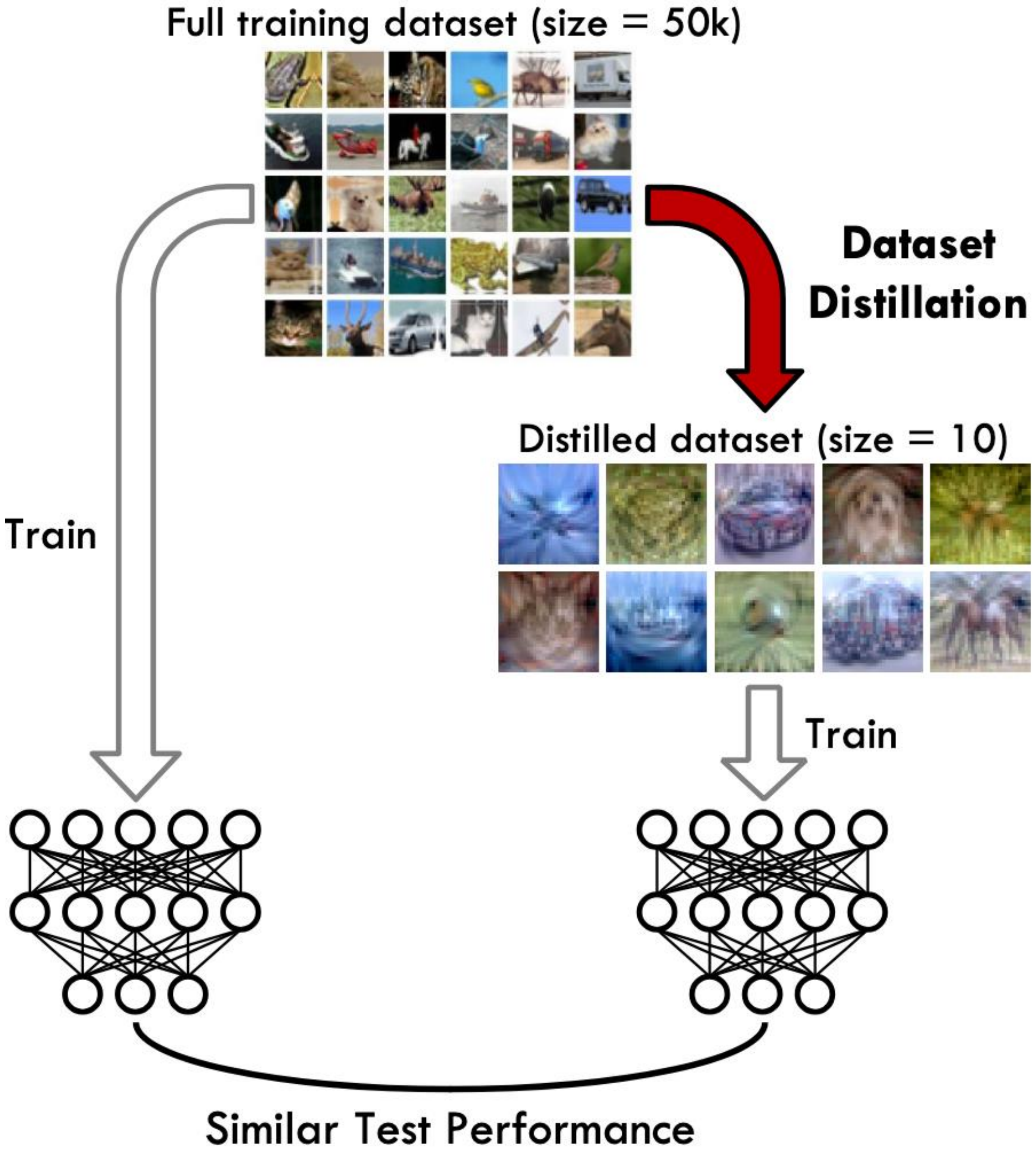
- The presentation contains content from the following sources:
  - https://georgecazenavette.github.io/mtt-distillation/
  - https://www.gatsby.ucl.ac.uk/~szabo/ml_external_seminar/Kamalika_Chaudhuri_external_seminar_02_03_2016_slides.pdf
  - https://cseweb.ucsd.edu/~yuxiangw/classes/DPCourse-2021Fall/Lectures/intro_slides.pdf

Overview

- Dataset Distillation
- Differential Privacy
- Differentially-private Dataset Distillation
- Improving Noise Efficiency

Dataset Distillation

Dataset Distillation


Full training dataset (size = 50k)
Dataset Distillation
Distilled dataset (size = 10)
Train
Train
Similar Test Performance

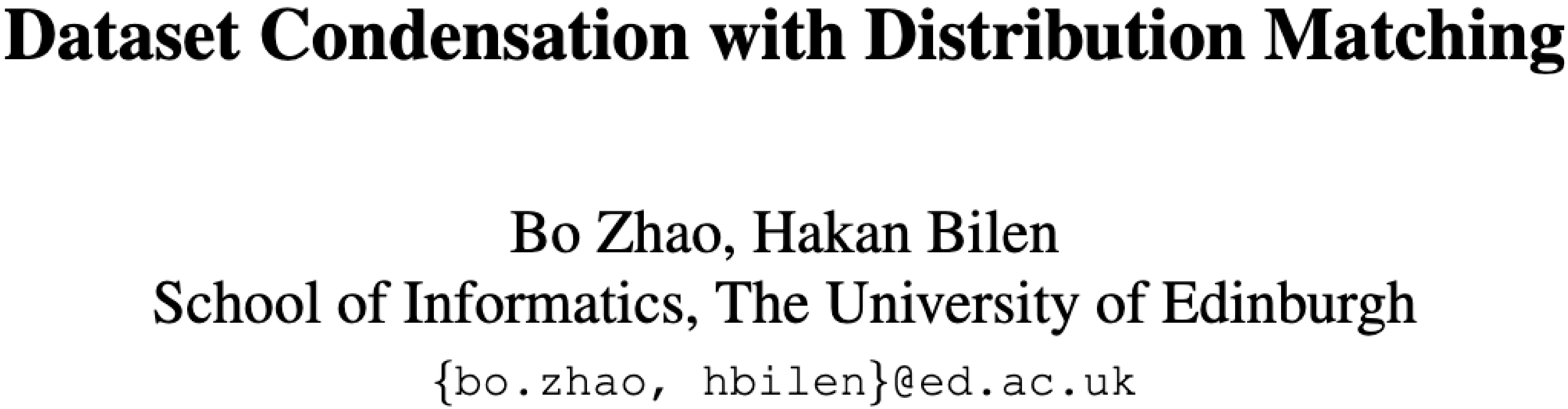
Dataset Distillation

- Formally, dataset distillation is defined as

$$\arg\min_{\mathcal{Z}} \; \mathbb{E}_{(x,y)\sim\mathcal{D}} \, \ell(g_{\theta(\mathcal{Z})}(x), y),$$
$$\text{where } \theta(\mathcal{Z}) = \arg\min_{\theta} \; \mathbb{E}_{(x,y)\sim\mathcal{Z}} \, \ell(g_{\theta}(x), y), \; |\mathcal{Z}| \ll |\mathcal{D}|$$

Where $g_\theta(\cdot)$ is the model parameterized by $\boldsymbol{\theta}$, $\ell(g_\theta(\boldsymbol{x}), \boldsymbol{y})$ is the loss function, $\boldsymbol{\mathcal{D}}$ is the real dataset, and $\mathcal{Z}$ is the synthetic dataset.

Distribution Matching

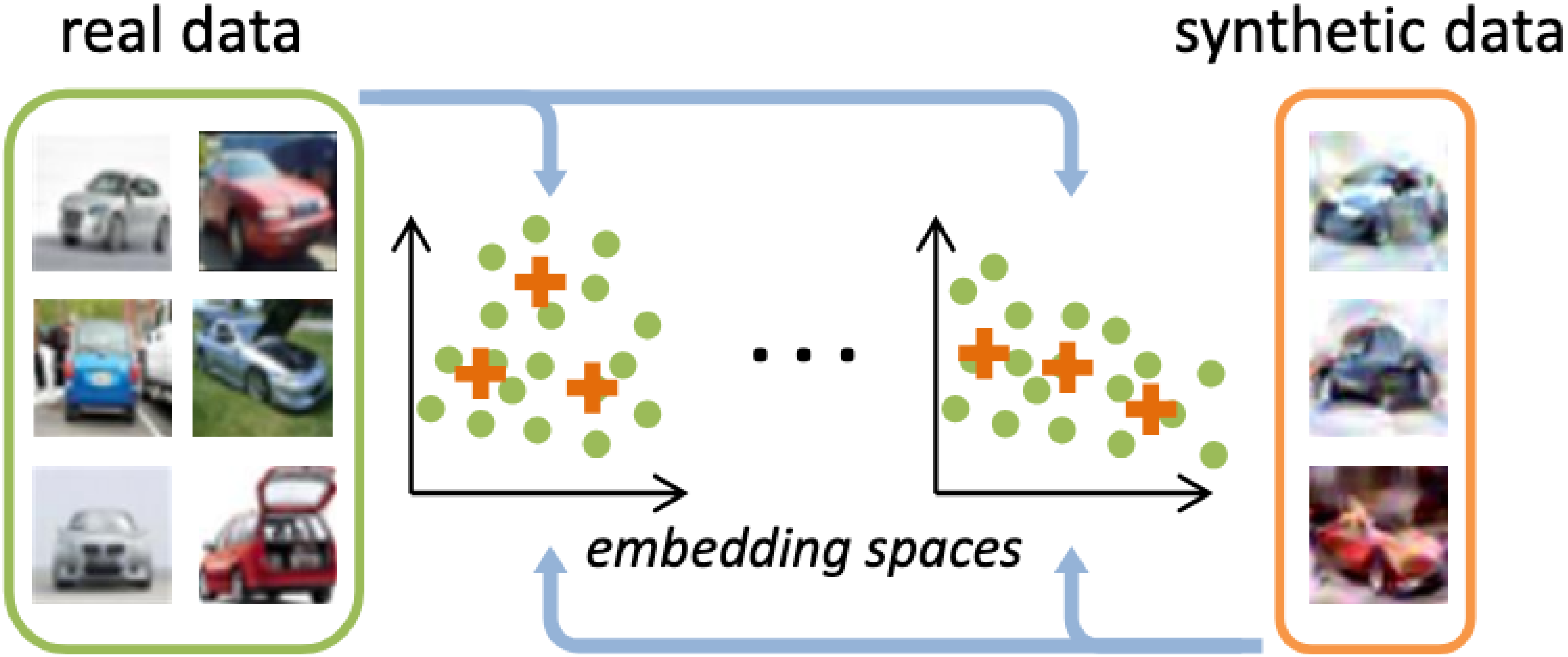


Dataset Condensation with Distribution Matching

Bo Zhao, Hakan Bilen
School of Informatics, The University of Edinburgh
{bo.zhao, hbilen}@ed.ac.uk

WACV 2023

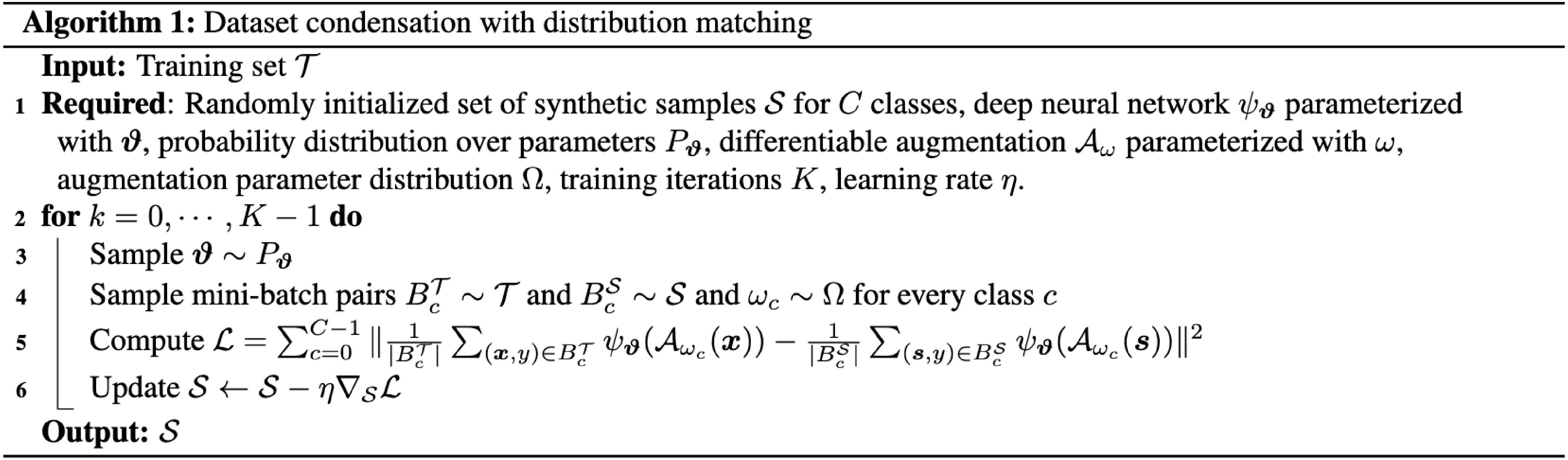
Overview



Approach

- Create random images for the synthetic dataset
- For K iterations
  - Randomly initialize a neural network (feature extractor)
  - Sample mini-batch per class from synthetic dataset and real dataset
  - Compute loss as MSE of features extracted using real and synthetic batches
  - Update synthetic dataset to minimize loss using gradient descent

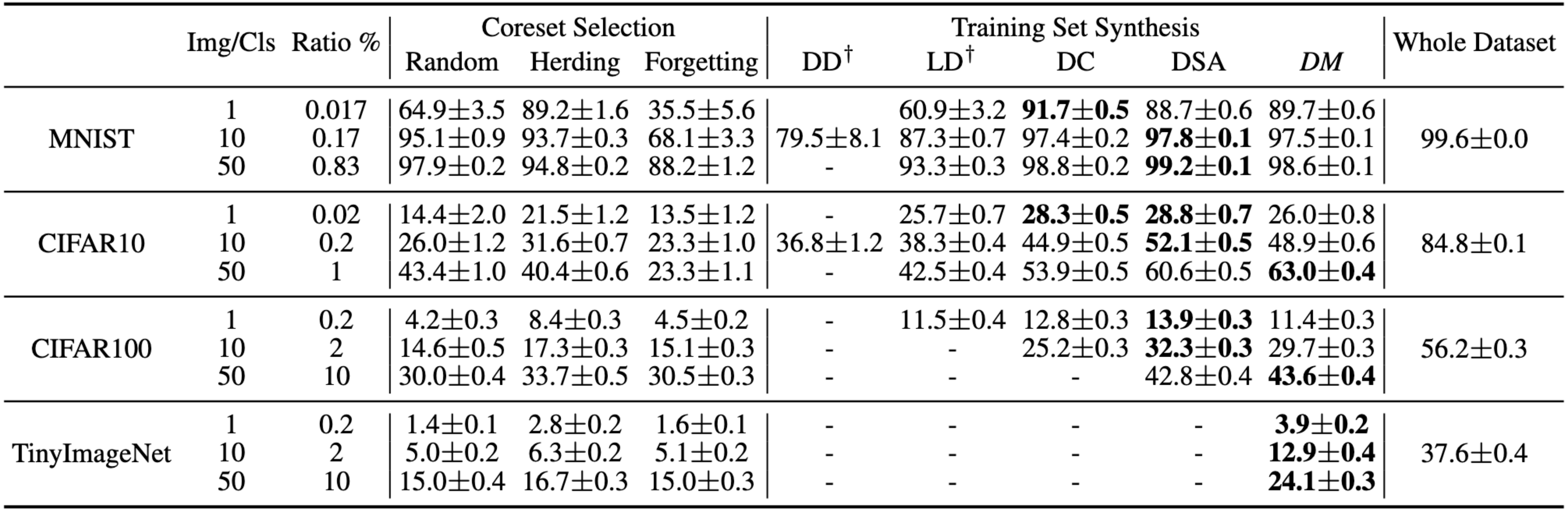
Algorithm



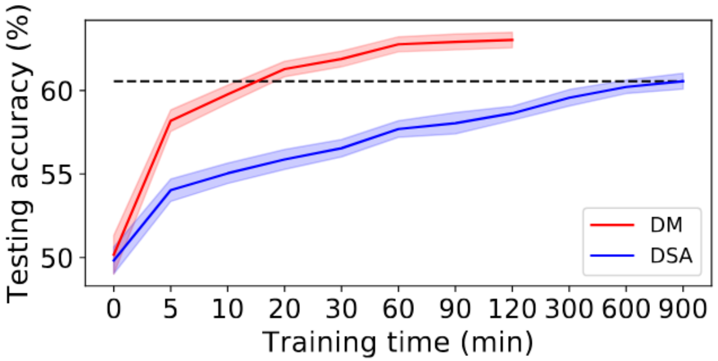
Algorithm 1: Dataset condensation with distribution matching

```
Input: Training set $\mathcal{T}$
Required: Randomly initialized set of synthetic samples $\mathcal{S}$ for $C$ classes, deep neural network $\psi_{\boldsymbol{\vartheta}}$ parameterized
  with $\boldsymbol{\vartheta}$, probability distribution over parameters $P_{\boldsymbol{\vartheta}}$, differentiable augmentation $\mathcal{A}_\omega$ parameterized with $\omega$,
  augmentation parameter distribution $\Omega$, training iterations $K$, learning rate $\eta$.
for $k = 0, \cdots, K-1$ do
    Sample $\boldsymbol{\vartheta} \sim P_{\boldsymbol{\vartheta}}$
    Sample mini-batch pairs $B_c^{\mathcal{T}} \sim \mathcal{T}$ and $B_c^{\mathcal{S}} \sim \mathcal{S}$ and $\omega_c \sim \Omega$ for every class $c$
    Compute $\mathcal{L} = \sum_{c=0}^{C-1} \| \frac{1}{|B_c^{\mathcal{T}}|} \sum_{(\boldsymbol{x},y) \in B_c^{\mathcal{T}}} \psi_{\boldsymbol{\vartheta}}(\mathcal{A}_{\omega_c}(\boldsymbol{x})) - \frac{1}{|B_c^{\mathcal{S}}|} \sum_{(\boldsymbol{s},y) \in B_c^{\mathcal{S}}} \psi_{\boldsymbol{\vartheta}}(\mathcal{A}_{\omega_c}(\boldsymbol{s})) \|^2$
    Update $\mathcal{S} \leftarrow \mathcal{S} - \eta \nabla_{\mathcal{S}} \mathcal{L}$
Output: $\mathcal{S}$
```

Results



	Img/Cls	Ratio %	Coreset Selection			Training Set Synthesis					Whole Dataset
			Random	Herding	Forgetting	DD†	LD†	DC	DSA	*DM*	
MNIST	1	0.017	64.9±3.5	89.2±1.6	35.5±5.6		60.9±3.2	**91.7±0.5**	88.7±0.6	89.7±0.6	99.6±0.0
	10	0.17	95.1±0.9	93.7±0.3	68.1±3.3	79.5±8.1	87.3±0.7	97.4±0.2	**97.8±0.1**	97.5±0.1	
	50	0.83	97.9±0.2	94.8±0.2	88.2±1.2	-	93.3±0.3	98.8±0.2	**99.2±0.1**	98.6±0.1	
CIFAR10	1	0.02	14.4±2.0	21.5±1.2	13.5±1.2	-	25.7±0.7	**28.3±0.5**	**28.8±0.7**	26.0±0.8	84.8±0.1
	10	0.2	26.0±1.2	31.6±0.7	23.3±1.0	36.8±1.2	38.3±0.4	44.9±0.5	**52.1±0.5**	48.9±0.6	
	50	1	43.4±1.0	40.4±0.6	23.3±1.1	-	42.5±0.4	53.9±0.5	60.6±0.5	**63.0±0.4**	
CIFAR100	1	0.2	4.2±0.3	8.4±0.3	4.5±0.2	-	11.5±0.4	12.8±0.3	**13.9±0.3**	11.4±0.3	56.2±0.3
	10	2	14.6±0.5	17.3±0.3	15.1±0.3	-	-	25.2±0.3	**32.3±0.3**	29.7±0.3	
	50	10	30.0±0.4	33.7±0.5	30.5±0.3	-	-	-	42.8±0.4	**43.6±0.4**	
TinyImageNet	1	0.2	1.4±0.1	2.8±0.2	1.6±0.1	-	-	-	-	**3.9±0.2**	37.6±0.4
	10	2	5.0±0.2	6.3±0.2	5.1±0.2	-	-	-	-	**12.9±0.4**	
	50	10	15.0±0.4	16.7±0.3	15.0±0.3	-	-	-	-	**24.1±0.3**	



Synthetic Images

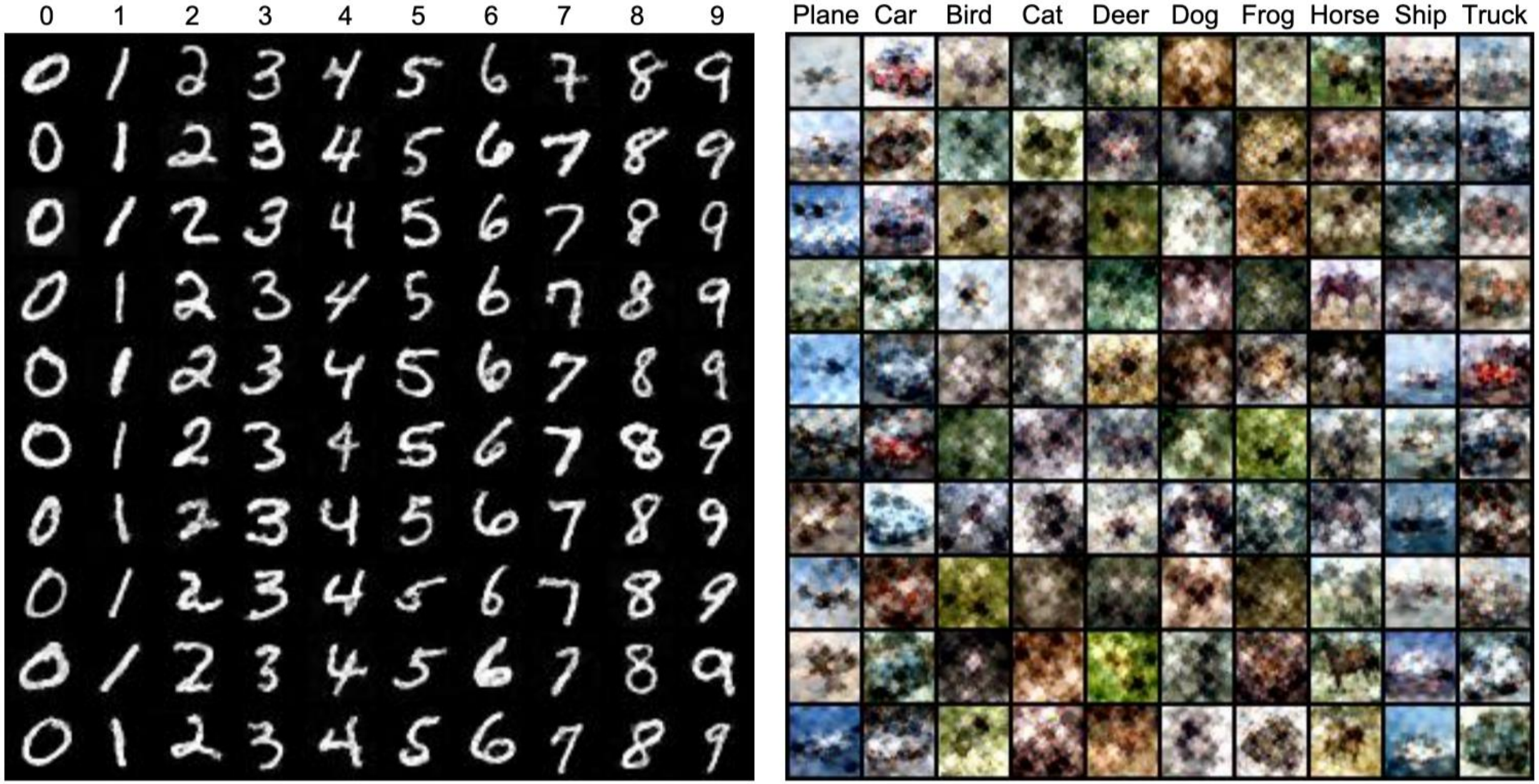
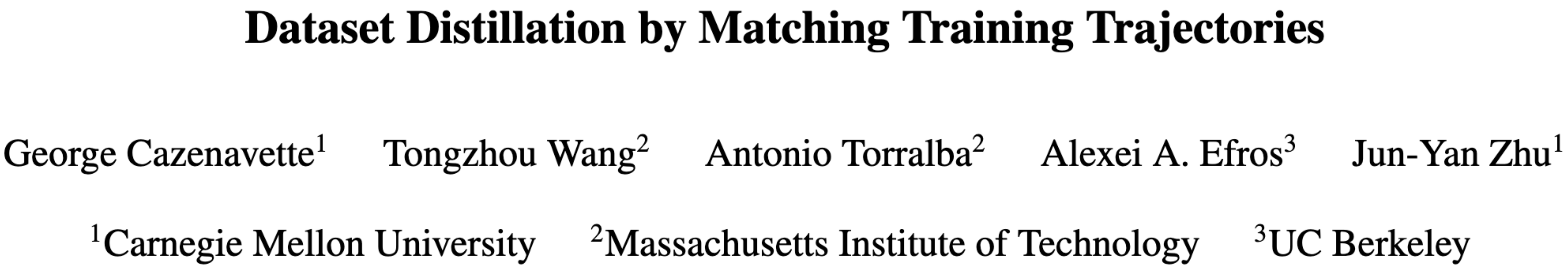


Figure 2. Visualization of generated 10 images per class synthetic sets of MNIST and CIFAR10 datasets.

Trajectory Matching



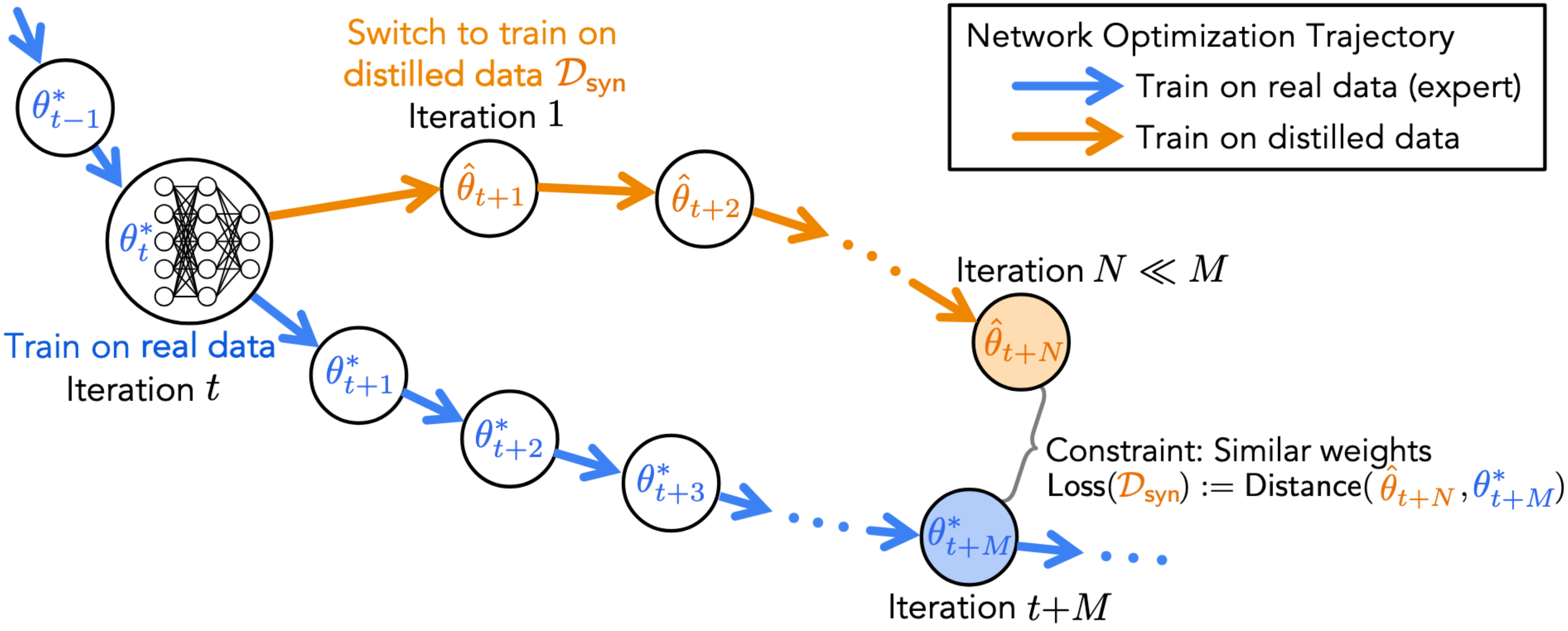
Dataset Distillation by Matching Training Trajectories

George Cazenavette[1] Tongzhou Wang[2] Antonio Torralba[2] Alexei A. Efros[3] Jun-Yan Zhu[1]

[1]Carnegie Mellon University [2]Massachusetts Institute of Technology [3]UC Berkeley

CVPR 2022

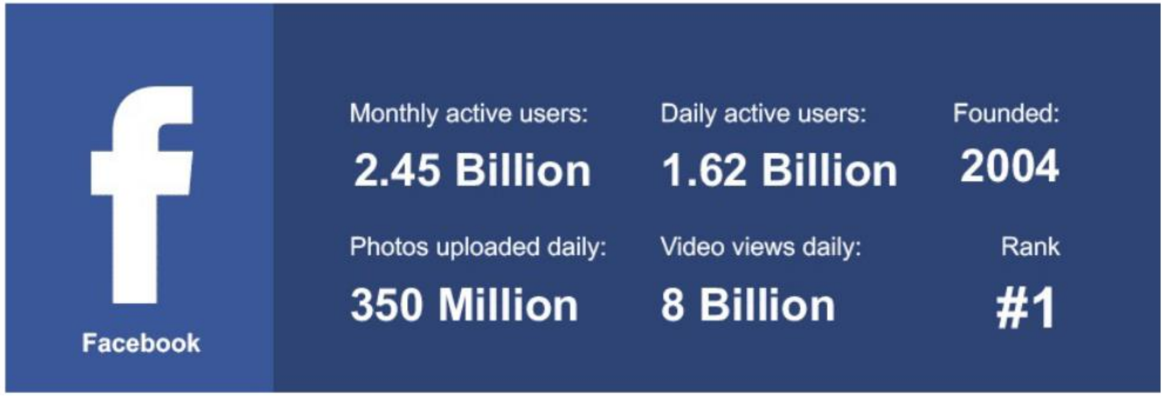
Overview


Switch to train on distilled data $\mathcal{D}_{\text{syn}}$
Iteration 1
$\hat{\theta}_{t+1}$
$\hat{\theta}_{t+2}$
Iteration $N \ll M$
$\hat{\theta}_{t+N}$
θ^*_{t-1}
θ^*_t
Train on real data
Iteration t
θ^*_{t+1}
θ^*_{t+2}
θ^*_{t+3}
θ^*_{t+M}
Iteration $t+M$
Constraint: Similar weights
Loss($\mathcal{D}_{\text{syn}}$) := Distance($\hat{\theta}_{t+N}$, θ^*_{t+M})
Network Optimization Trajectory
Train on real data (expert)
Train on distilled data

Differential Privacy

Personal Data in Big Data Era

- Government, company, research centers collect personal information and analyze them.
- Social networks: Facebook, LinkedIn
- YouTube & Amazon use viewing/buying records for recommendations.



Facebook

Monthly active users: 2.45 Billion

Daily active users: 1.62 Billion

Founded: 2004

Photos uploaded daily: 350 Million

Video views daily: 8 Billion

Rank #1

Source: https://www.garyfox.co/social-media-statistics/

Legislations on Privacy

- I can’t keep personal data for more than three weeks?
- I will have to delete all traces of a user upon request?

How about my machine learning models trained on user data?

ML Needs Data

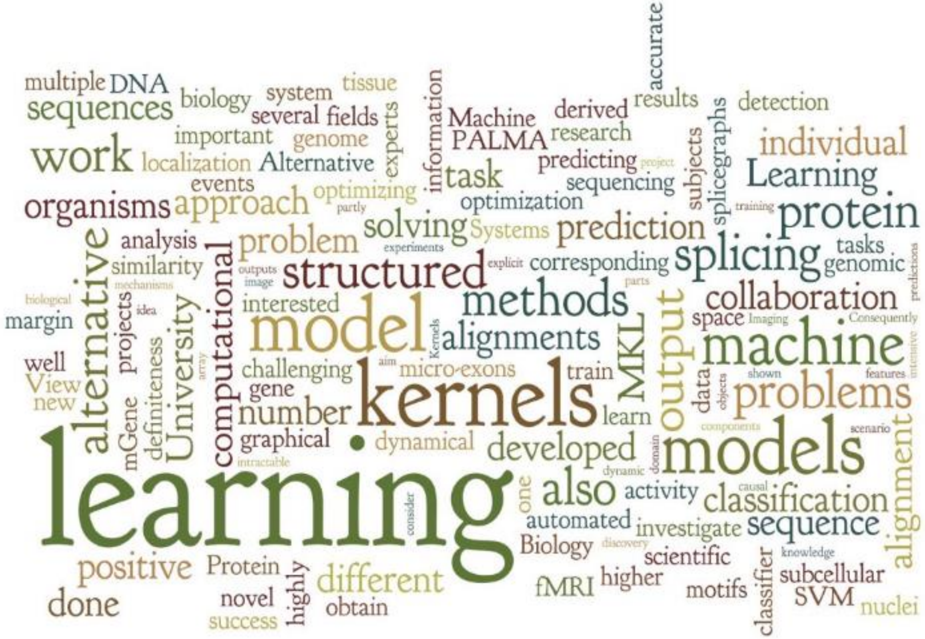
Privacy law

Food safety

Understand how brain works

Better treatment

Better education



Medical diagnosis

Clean energy

Epidemic forecast/prevention

Fraud detection

Increasing privacy awareness

Anonymization Does not Work

- “Who likes Justin Bieber?”
- Questionnaire: “Year, Program, Gender, Like Bieber or not?”
    - Results as of Monday: “How many like Bieber” 16
    - Results as of Tuesday: “How many like Bieber” 17
    - Side information (available to the instructor): You enrolled late on Tuesday.

High Dimensional Data Leaks Information

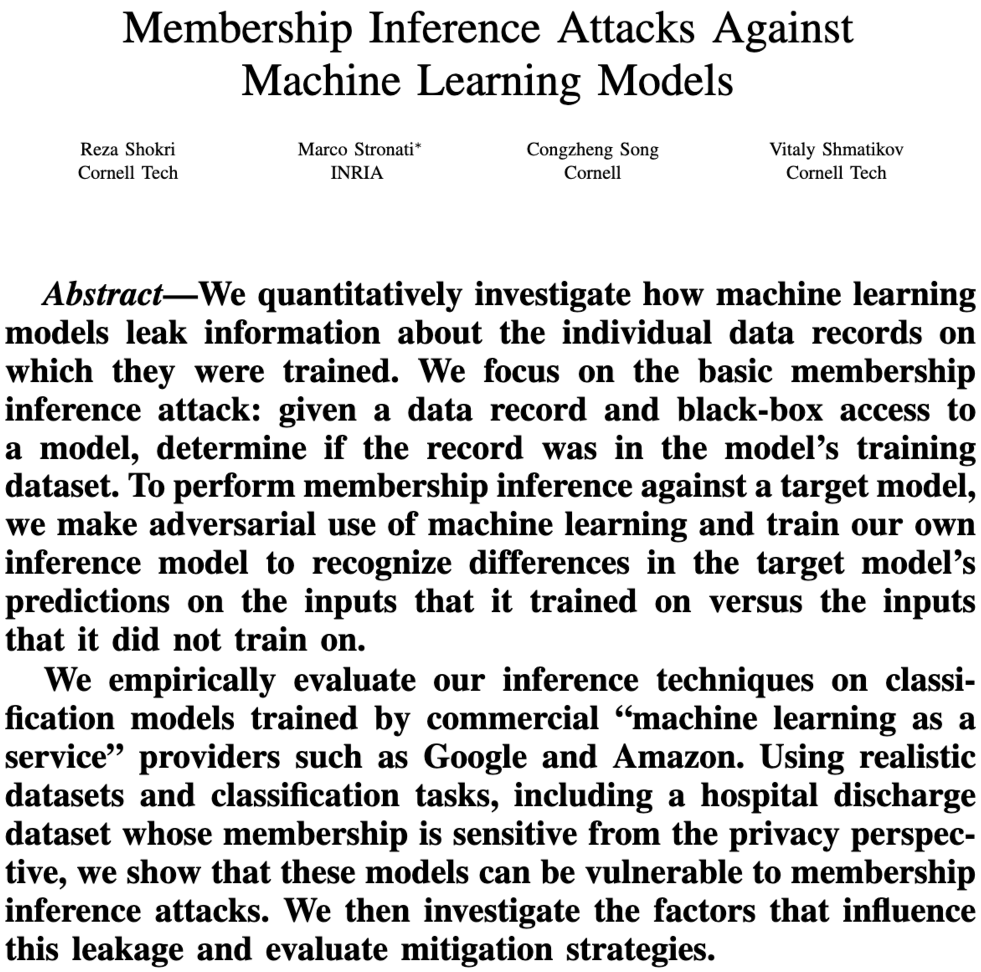
Research Area	Gender	Department	Ethnicity	Name
Binary Analysis	Woman	CSE	SE Asian	-

High Dimensional Data Leaks Information

Research Area	Gender	Department	Ethnicity	Name
Binary Analysis	Woman	CSE	SE Asian	-

Monika!

Attacks on ML Models



Membership Inference Attacks Against Machine Learning Models

Reza Shokri
Cornell Tech

Marco Stronati*
INRIA

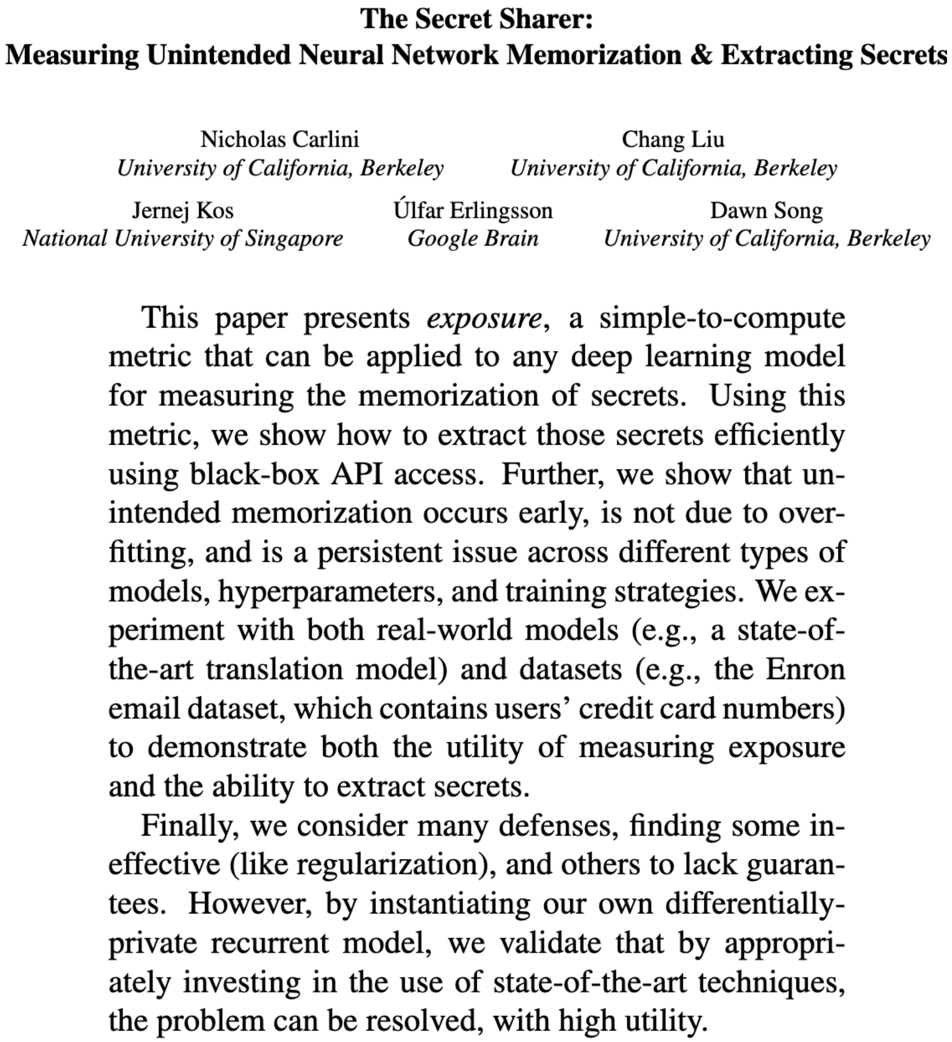
Congzheng Song
Cornell

Vitaly Shmatikov
Cornell Tech

***Abstract*—We quantitatively investigate how machine learning models leak information about the individual data records on which they were trained. We focus on the basic membership inference attack: given a data record and black-box access to a model, determine if the record was in the model's training dataset. To perform membership inference against a target model, we make adversarial use of machine learning and train our own inference model to recognize differences in the target model's predictions on the inputs that it trained on versus the inputs that it did not train on.**

We empirically evaluate our inference techniques on classification models trained by commercial "machine learning as a service" providers such as Google and Amazon. Using realistic datasets and classification tasks, including a hospital discharge dataset whose membership is sensitive from the privacy perspective, we show that these models can be vulnerable to membership inference attacks. We then investigate the factors that influence this leakage and evaluate mitigation strategies.

Security and Privacy, 2017



**The Secret Sharer:
Measuring Unintended Neural Network Memorization & Extracting Secrets**

Nicholas Carlini
University of California, Berkeley

Chang Liu
University of California, Berkeley

Jernej Kos
National University of Singapore

Úlfar Erlingsson
Google Brain

Dawn Song
University of California, Berkeley

This paper presents *exposure*, a simple-to-compute metric that can be applied to any deep learning model for measuring the memorization of secrets. Using this metric, we show how to extract those secrets efficiently using black-box API access. Further, we show that unintended memorization occurs early, is not due to overfitting, and is a persistent issue across different types of models, hyperparameters, and training strategies. We experiment with both real-world models (e.g., a state-of-the-art translation model) and datasets (e.g., the Enron email dataset, which contains users' credit card numbers) to demonstrate both the utility of measuring exposure and the ability to extract secrets.

Finally, we consider many defenses, finding some ineffective (like regularization), and others to lack guarantees. However, by instantiating our own differentially-private recurrent model, we validate that by appropriately investing in the use of state-of-the-art techniques, the problem can be resolved, with high utility.

USENIX Security 19

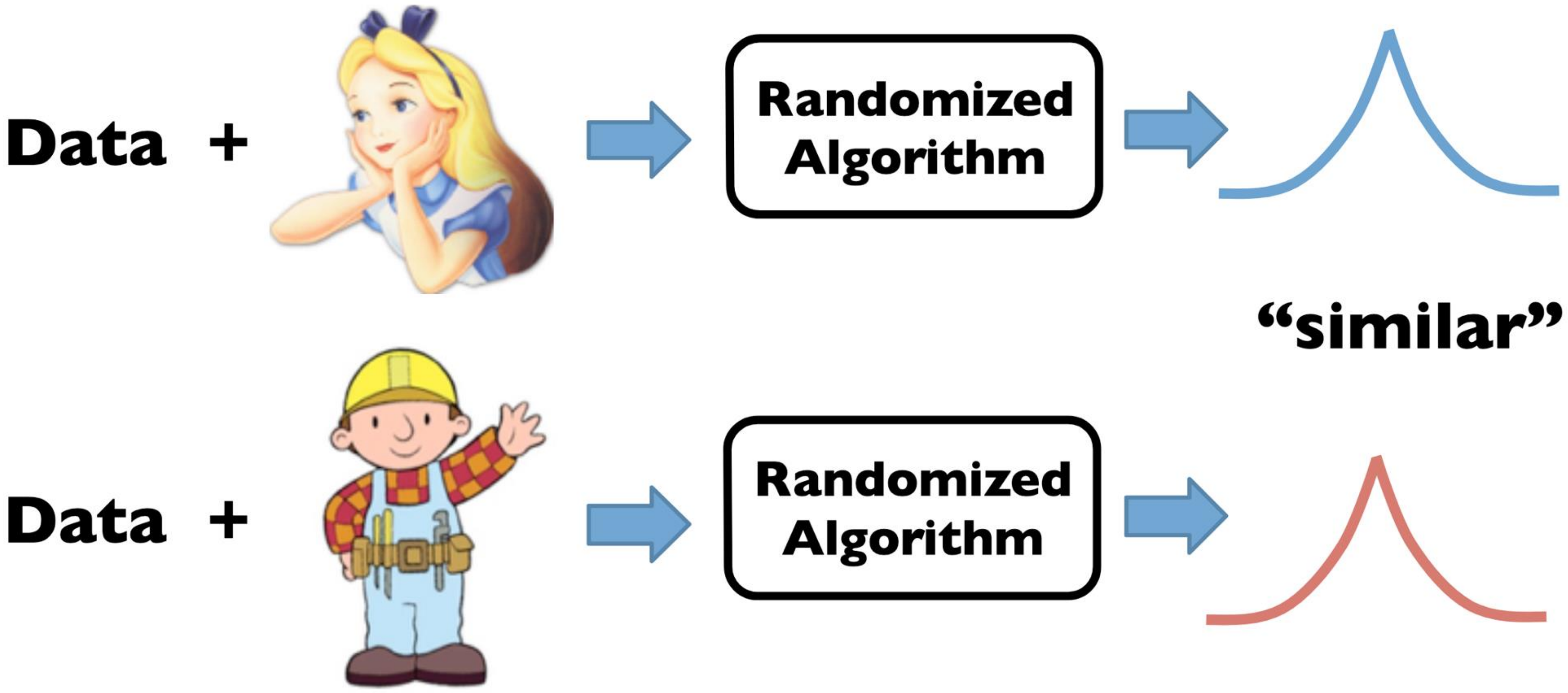
Problem

- We need to learn from data, but personal data is sensitive
- How about we learn population insights without compromising individuals?

Differential Privacy (DP)

- DP is a mathematical framework that quantifies privacy loss
- It bounds the contribution/effect each data point has on the outcome

Differential Privacy (DP)



Data +

Randomized Algorithm

“similar”

Data +

Randomized Algorithm

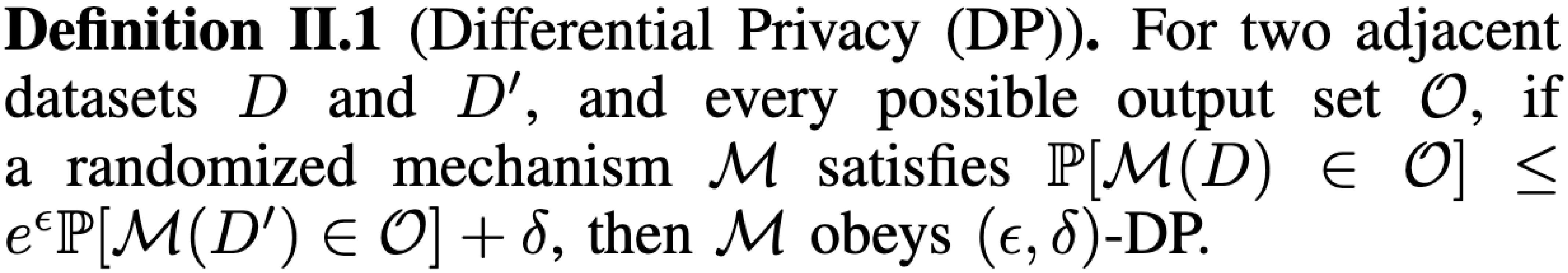
Participation of a single person does not change output

DP: Attacker’s Perspective

Prior Knowledge + **Algorithm** Output on Data & = **Conclusion** on

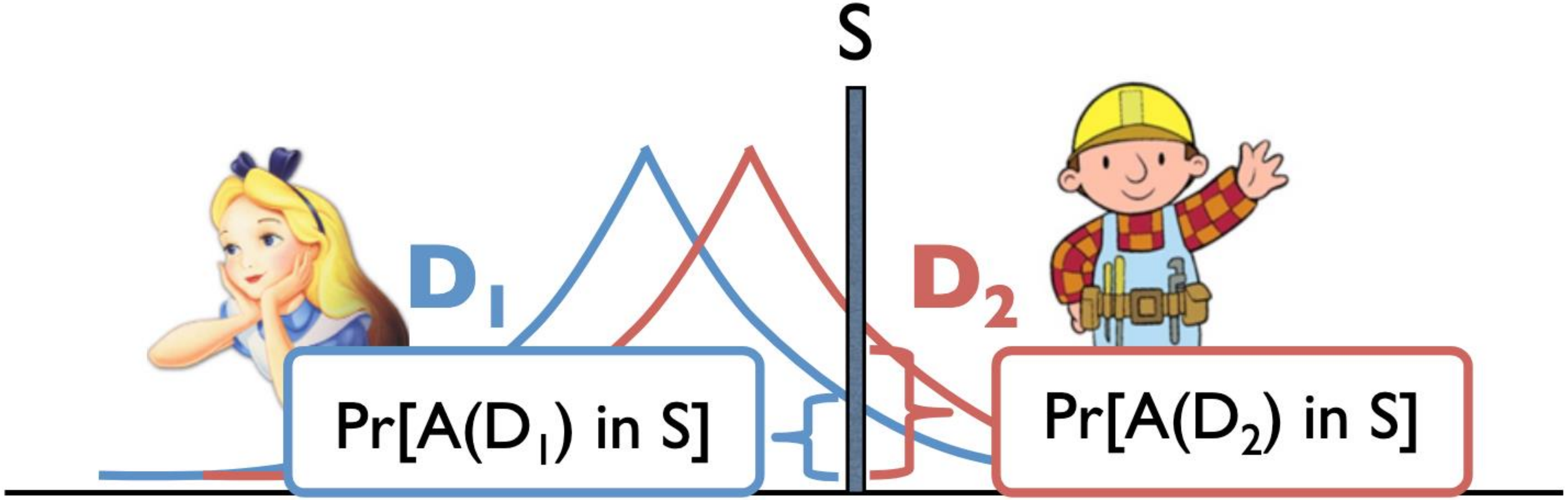
Prior Knowledge + **Algorithm** Output on Data & = **Conclusion** on

Formal Definition



Definition II.1 (Differential Privacy (DP))**.** For two adjacent datasets D and D', and every possible output set $\mathcal{O}$, if a randomized mechanism $\mathcal{M}$ satisfies $\mathbb{P}[\mathcal{M}(D) \in \mathcal{O}] \leq e^{\epsilon}\mathbb{P}[\mathcal{M}(D') \in \mathcal{O}] + \delta$, then $\mathcal{M}$ obeys (ϵ, δ)-DP.

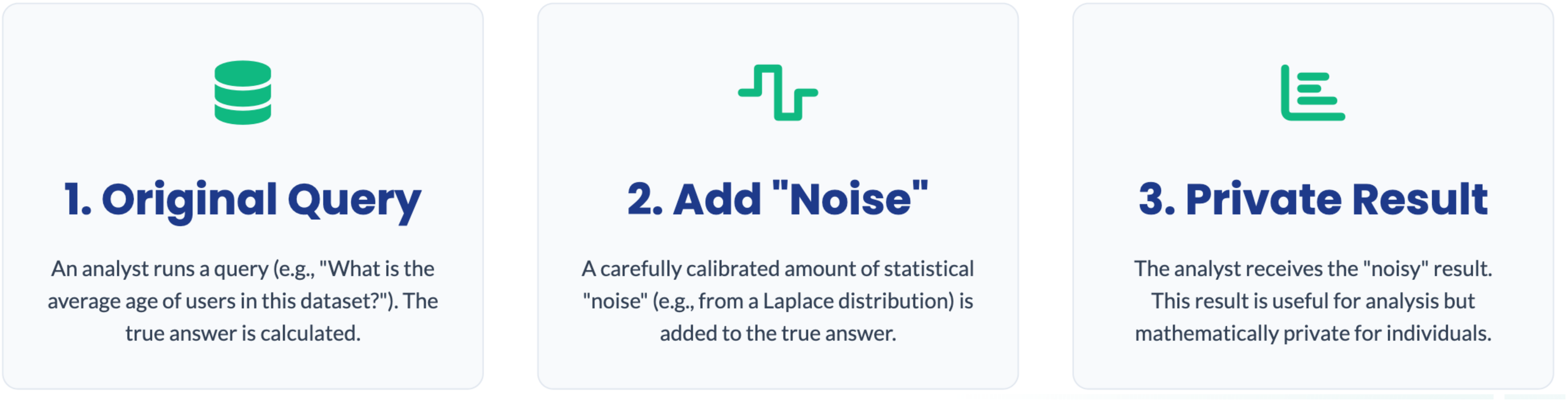
DP Guarantee



For all D1, D2 that differ in one person's value, any set S,
If A = α-private randomized algorithm, then:

$$\Pr(A(D_1) \in S) \leq e^{\alpha} \Pr(A(D_2) \in S)$$

High Level Idea



1. Original Query

An analyst runs a query (e.g., "What is the average age of users in this dataset?"). The true answer is calculated.

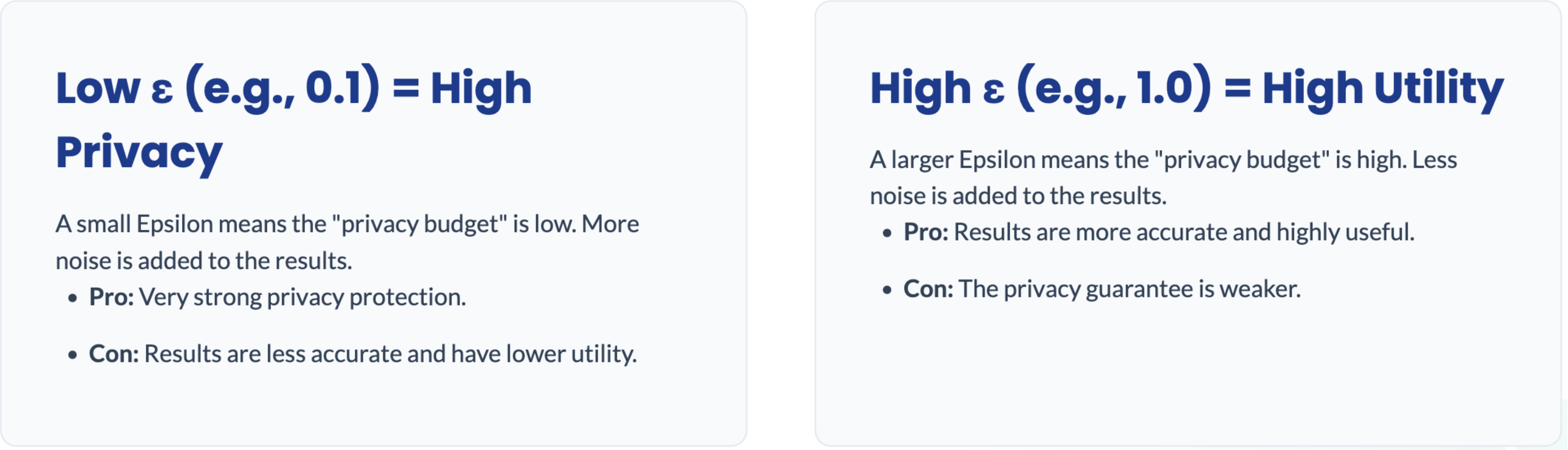
2. Add "Noise"

A carefully calibrated amount of statistical "noise" (e.g., from a Laplace distribution) is added to the true answer.

3. Private Result

The analyst receives the "noisy" result. This result is useful for analysis but mathematically private for individuals.

Privacy-Utility Tradeoff



Low ε (e.g., 0.1) = High Privacy

A small Epsilon means the "privacy budget" is low. More noise is added to the results.

- **Pro:** Very strong privacy protection.
- **Con:** Results are less accurate and have lower utility.

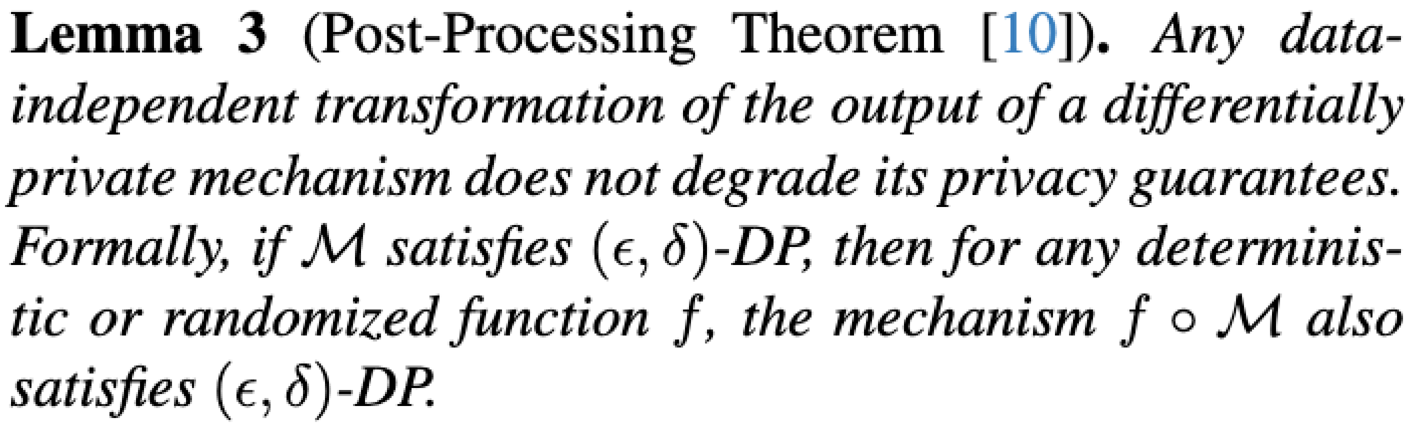
High ε (e.g., 1.0) = High Utility

A larger Epsilon means the "privacy budget" is high. Less noise is added to the results.

- **Pro:** Results are more accurate and highly useful.
- **Con:** The privacy guarantee is weaker.

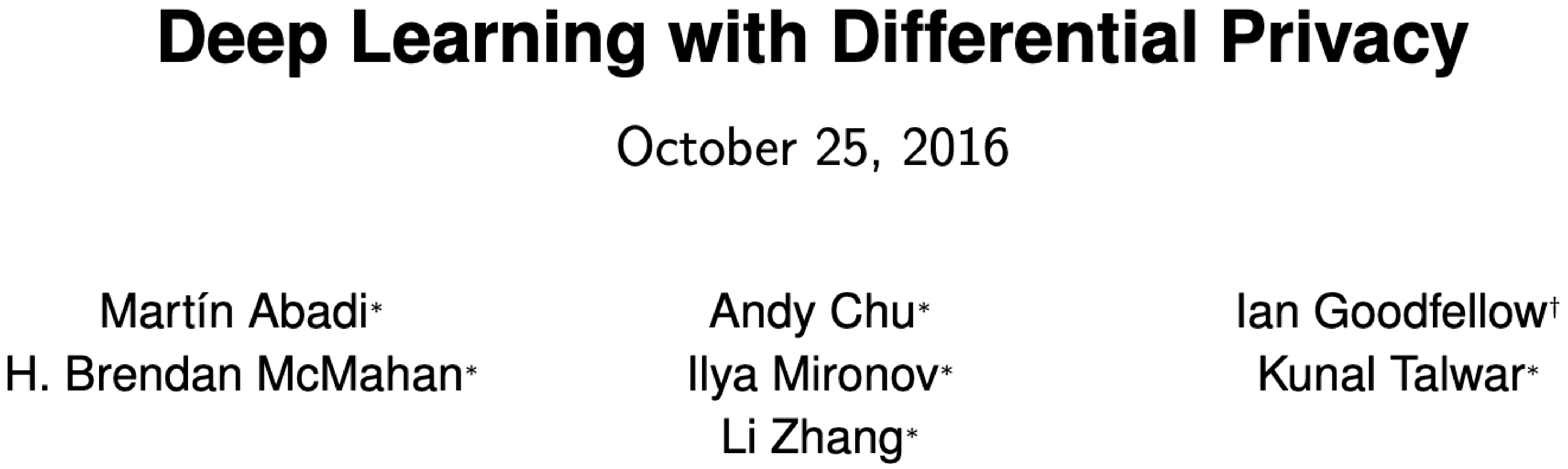
Important Properties of DP

- Post-processing Theorem:



Lemma 3 (Post-Processing Theorem [10]). *Any data-independent transformation of the output of a differentially private mechanism does not degrade its privacy guarantees. Formally, if $\mathcal{M}$ satisfies (ϵ, δ)-DP, then for any deterministic or randomized function f, the mechanism $f \circ \mathcal{M}$ also satisfies (ϵ, δ)-DP.*

Applying DP to ML



Deep Learning with Differential Privacy

October 25, 2016

Martín Abadi[*]
H. Brendan McMahan[*]

Andy Chu[*]
Ilya Mironov[*]
Li Zhang[*]

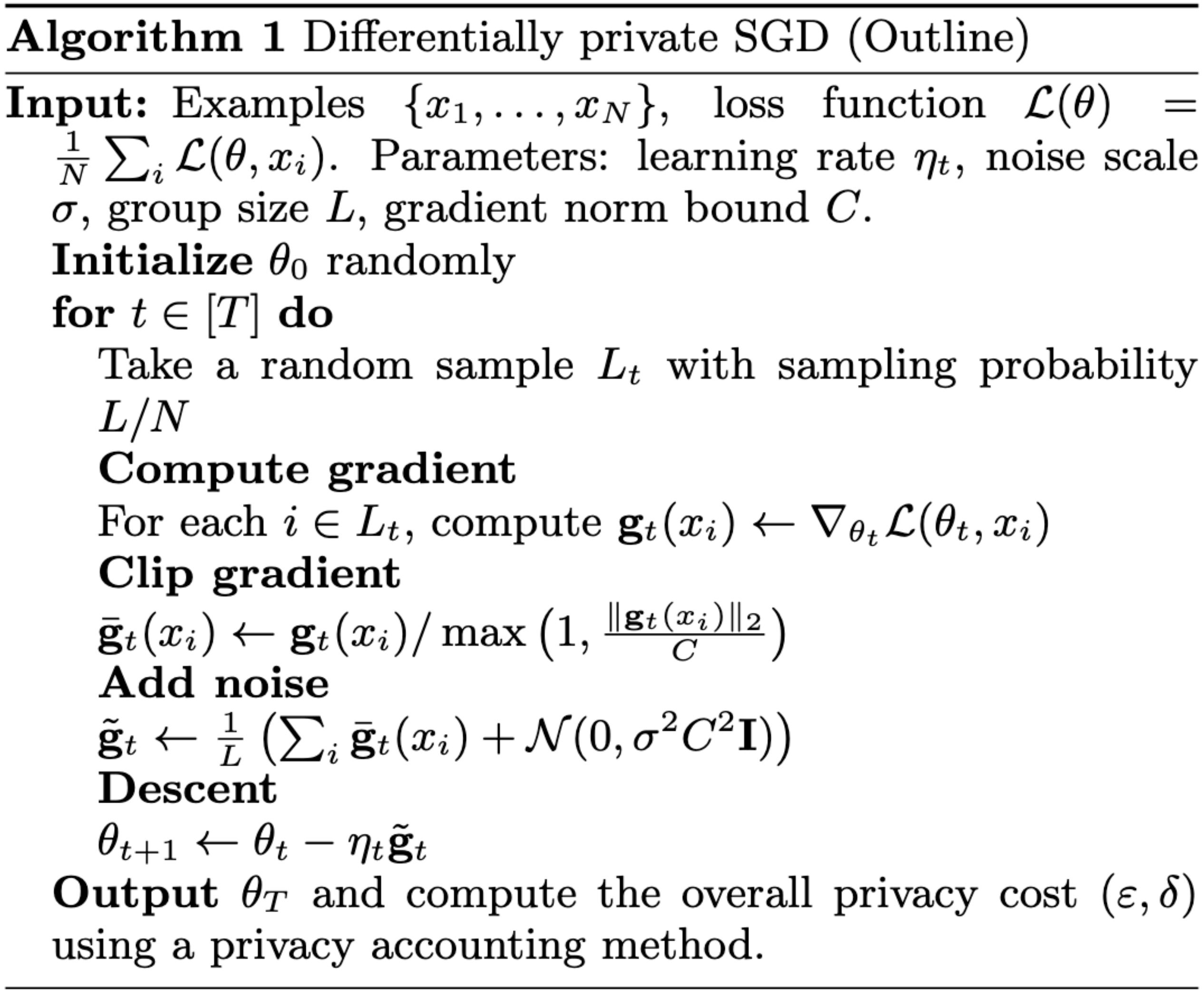
Ian Goodfellow[†]
Kunal Talwar[*]

CCS 2016

Idea

- SGD is the randomized algorithm
- Adding DP to SGD:
  - Use Poisson sampling to create mini-batches
  - Clip the gradients to bound their norm
  - Add calibrated Gaussian noise to gradients
  - Perform gradient descent with noisy gradients

DP-SGD



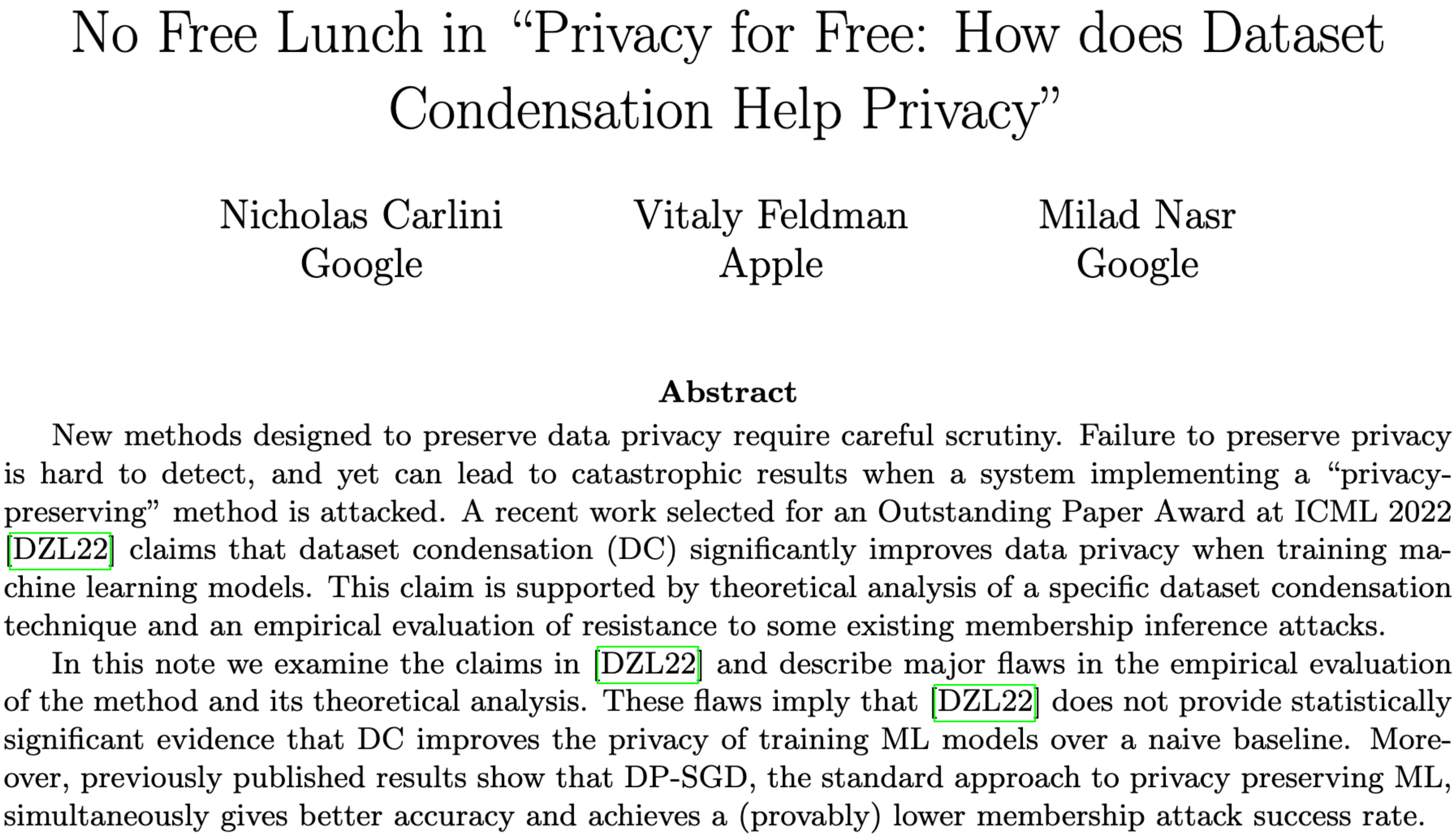
Algorithm 1 Differentially private SGD (Outline)

```
Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate $\eta_t$, noise scale $\sigma$, group size $L$, gradient norm bound $C$.
  Initialize $\theta_0$ randomly
  for $t \in [T]$ do
    Take a random sample $L_t$ with sampling probability $L/N$
    Compute gradient
    For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$
    Clip gradient
    $\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$
    Add noise
    $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
    Descent
    $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$
  Output $\theta_T$ and compute the overall privacy cost $(\varepsilon, \delta)$ using a privacy accounting method.
```

$\Bigg\}$ DP-specific Modifications

Differentially Private Dataset Distillation

Distillation Seems Private but NO!



No Free Lunch in "Privacy for Free: How does Dataset Condensation Help Privacy"

Nicholas Carlini
Google

Vitaly Feldman
Apple

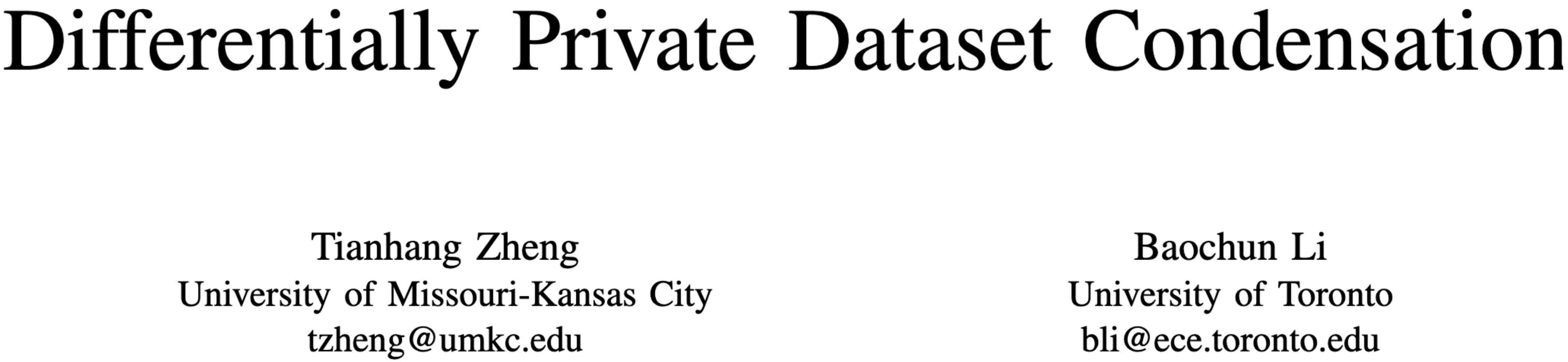
Milad Nasr
Google

Abstract

New methods designed to preserve data privacy require careful scrutiny. Failure to preserve privacy is hard to detect, and yet can lead to catastrophic results when a system implementing a "privacy-preserving" method is attacked. A recent work selected for an Outstanding Paper Award at ICML 2022 [DZL22] claims that dataset condensation (DC) significantly improves data privacy when training machine learning models. This claim is supported by theoretical analysis of a specific dataset condensation technique and an empirical evaluation of resistance to some existing membership inference attacks.

In this note we examine the claims in [DZL22] and describe major flaws in the empirical evaluation of the method and its theoretical analysis. These flaws imply that [DZL22] does not provide statistically significant evidence that DC improves the privacy of training ML models over a naive baseline. Moreover, previously published results show that DP-SGD, the standard approach to privacy preserving ML, simultaneously gives better accuracy and achieves a (provably) lower membership attack success rate.

DP and Distillation



Differentially Private Dataset Condensation

Tianhang Zheng
University of Missouri-Kansas City
tzheng@umkc.edu

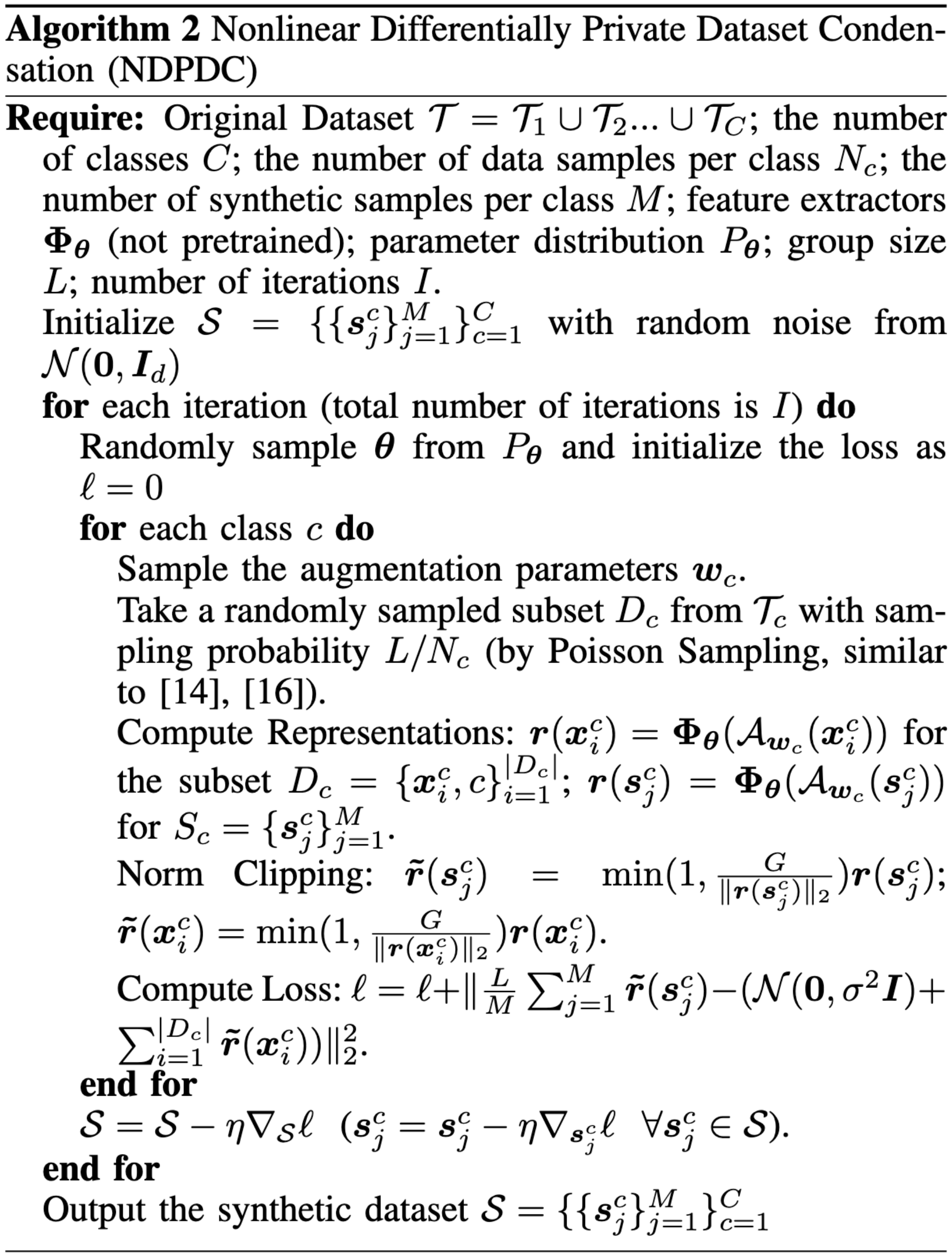
Baochun Li
University of Toronto
bli@ece.toronto.edu

NDSS 2024

Key Idea

- Combine distribution matching with DP
- Similar to DP-SGD: combining SGD with DP

Algorithm



Algorithm 2 Nonlinear Differentially Private Dataset Condensation (NDPDC)

```
Require: Original Dataset $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2 ... \cup \mathcal{T}_C$; the number of classes $C$; the number of data samples per class $N_c$; the number of synthetic samples per class $M$; feature extractors $\boldsymbol{\Phi}_{\boldsymbol{\theta}}$ (not pretrained); parameter distribution $P_{\boldsymbol{\theta}}$; group size $L$; number of iterations $I$.
  Initialize $\mathcal{S} = \{\{\boldsymbol{s}_j^c\}_{j=1}^M\}_{c=1}^C$ with random noise from $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$
  for each iteration (total number of iterations is $I$) do
    Randomly sample $\boldsymbol{\theta}$ from $P_{\boldsymbol{\theta}}$ and initialize the loss as $\ell = 0$
    for each class $c$ do
      Sample the augmentation parameters $\boldsymbol{w}_c$.
      Take a randomly sampled subset $D_c$ from $\mathcal{T}_c$ with sampling probability $L/N_c$ (by Poisson Sampling, similar to [14], [16]).
      Compute Representations: $\boldsymbol{r}(\boldsymbol{x}_i^c) = \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}_c}(\boldsymbol{x}_i^c))$ for the subset $D_c = \{\boldsymbol{x}_i^c, c\}_{i=1}^{|D_c|}$; $\boldsymbol{r}(\boldsymbol{s}_j^c) = \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathcal{A}_{\boldsymbol{w}_c}(\boldsymbol{s}_j^c))$ for $S_c = \{\boldsymbol{s}_j^c\}_{j=1}^M$.
      Norm Clipping: $\tilde{\boldsymbol{r}}(\boldsymbol{s}_j^c) = \min(1, \frac{G}{\|\boldsymbol{r}(\boldsymbol{s}_j^c)\|_2})\boldsymbol{r}(\boldsymbol{s}_j^c)$; $\tilde{\boldsymbol{r}}(\boldsymbol{x}_i^c) = \min(1, \frac{G}{\|\boldsymbol{r}(\boldsymbol{x}_i^c)\|_2})\boldsymbol{r}(\boldsymbol{x}_i^c)$.
      Compute Loss: $\ell = \ell + \| \frac{L}{M} \sum_{j=1}^M \tilde{\boldsymbol{r}}(\boldsymbol{s}_j^c) - (\mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}) + \sum_{i=1}^{|D_c|} \tilde{\boldsymbol{r}}(\boldsymbol{x}_i^c))\|_2^2$.
    end for
    $\mathcal{S} = \mathcal{S} - \eta \nabla_{\mathcal{S}} \ell$  $(\boldsymbol{s}_j^c = \boldsymbol{s}_j^c - \eta \nabla_{\boldsymbol{s}_j^c} \ell \ \ \forall \boldsymbol{s}_j^c \in \mathcal{S})$.
  end for
  Output the synthetic dataset $\mathcal{S} = \{\{\boldsymbol{s}_j^c\}_{j=1}^M\}_{c=1}^C$
```

Sample Feature Extractor

Sample Subset of Real Data

Compute features for both

Clip and add Noise

Update synthetic data

Improving Noise Efficiency in DP and Distillation

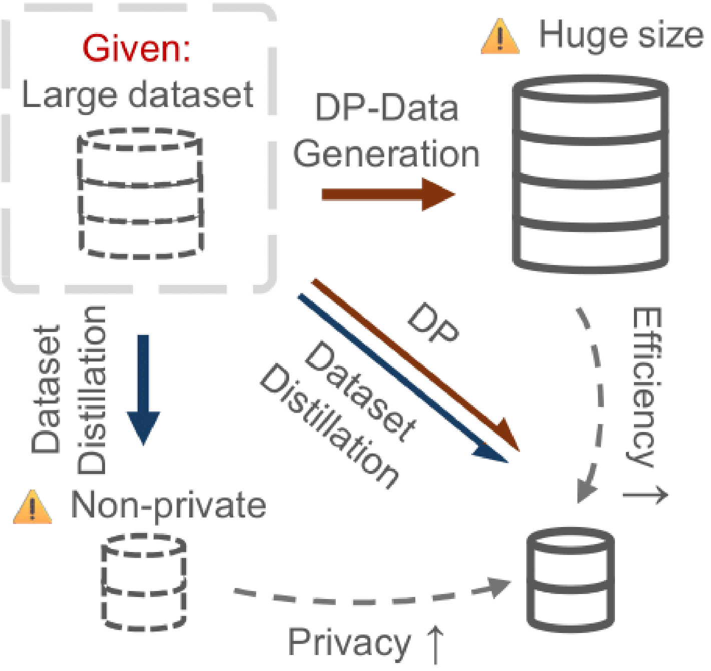
Introduction & Motivation

The Core Challenge:

- Modern machine learning models require massive datasets.
- These datasets often contain private user information.

The Goal:

Create **small, efficient, and formally private** synthetic datasets.



Existing Tools:

- **Differentially Privacy (DP):** Offers strong, provable privacy guarantees.
- **Dataset Distillation (DD):** Excels at creating very small synthetic datasets.

Opportunity: Combining DP with DD could yield the best of both worlds: small, efficient, and private datasets.

The Problem with Current Methods

Current private Dataset Distillation methods are inefficient in their use of the privacy budget.

Why?

1.Synchronized Sampling & Optimization:

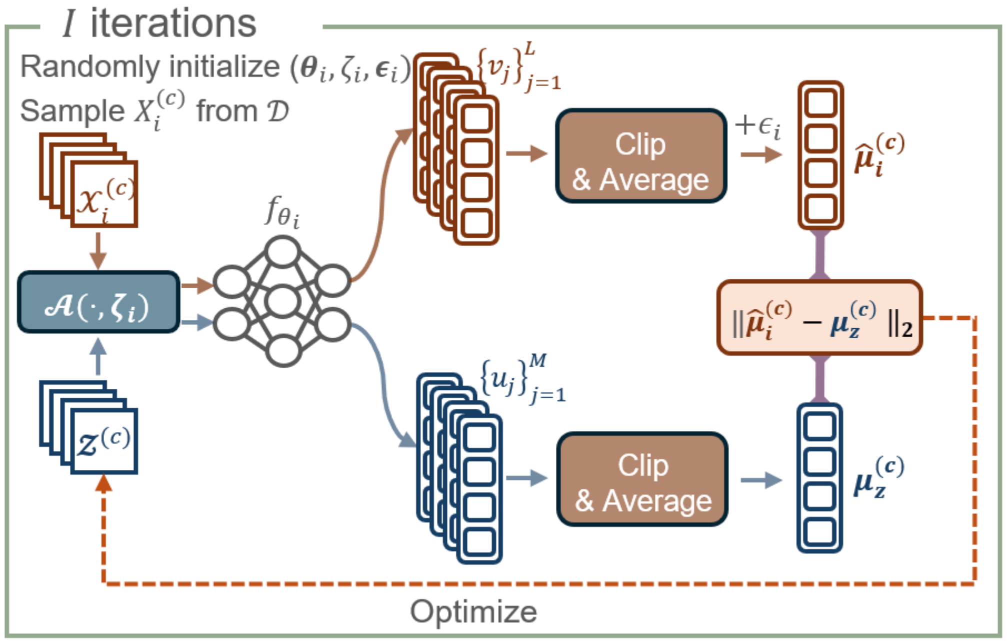
Wastes privacy budget by injecting noise at every step.

1.Noisy Signals

Relies on gradients from randomly initialized networks.

The Result:

Suboptimal performance, especially under strict privacy.


I iterations
Randomly initialize $(\boldsymbol{\theta}_i, \zeta_i, \boldsymbol{\epsilon}_i)$
Sample $X_i^{(c)}$ from $\mathcal{D}$
$\mathcal{X}_i^{(c)}$
$\mathcal{A}(\cdot, \zeta_i)$
$\mathcal{Z}^{(c)}$
f_{θ_i}
$\{v_j\}_{j=1}^{L}$
Clip & Average
$+\epsilon_i$
$\hat{\boldsymbol{\mu}}_i^{(c)}$
$\|\hat{\boldsymbol{\mu}}_i^{(c)} - \boldsymbol{\mu}_z^{(c)}\|_2$
$\{u_j\}_{j=1}^{M}$
Clip & Average
$\boldsymbol{\mu}_z^{(c)}$
Optimize

Key Contributions

We proposed a novel two-stage framework that is robust to DP noise and significantly improves utility.

- The core **DOS (decoupled optimization and sampling)** module outperforms SOTA methods.
- The optional **SER (subspace discovery for error reduction)** module uses public/DP-constrained data to further boost performance.
- Compared to the strongest baseline, achieved 8%+ accuracy boost with only 1/5 of the dataset size on CIFAR-10.

Our Solution: The DOSSER Framework

DOS - Decoupled Optimization and Sampling:

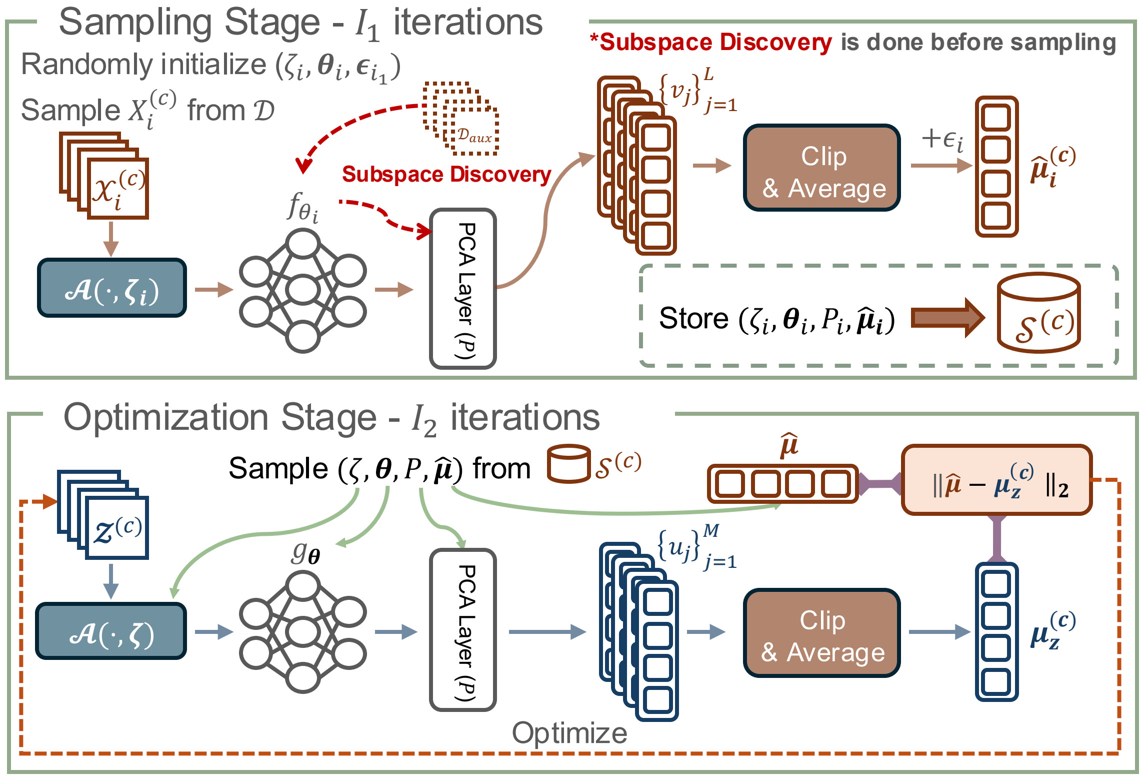
Stage 1: Sampling. Private signals are encoded and stored.

Stage 2: Optimization. The synthetic dataset is optimized to match the stored signals.

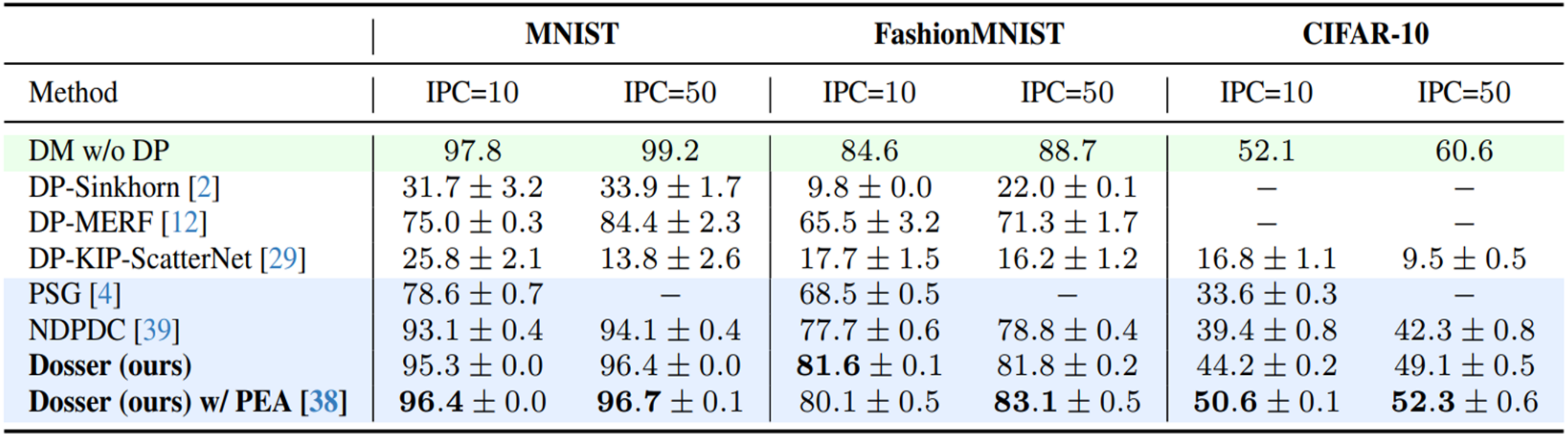
SER — Subspace discovery for Error Reduction:

Subspace Discovery: Estimate informative subspace with auxiliary dataset.

Dimensional Reduction: Project training signals into low-dim subspace to reduce effect of DP-noise to training signals.


Sampling Stage - I_1 iterations
*Subspace Discovery is done before sampling
Randomly initialize $(\zeta_i, \boldsymbol{\theta}_i, \boldsymbol{\epsilon}_{i_1})$
Sample $X_i^{(c)}$ from $\mathcal{D}$
$\mathcal{D}_{aux}$
Subspace Discovery
$\mathcal{X}_i^{(c)}$
f_{θ_i}
$\mathcal{A}(\cdot, \zeta_i)$
PCA Layer (P)
$\{v_j\}_{j=1}^{L}$
Clip & Average
$+\epsilon_i$
$\hat{\boldsymbol{\mu}}_i^{(c)}$
Store $(\zeta_i, \boldsymbol{\theta}_i, P_i, \hat{\boldsymbol{\mu}}_i)$
$\mathcal{S}^{(c)}$
Optimization Stage - I_2 iterations
Sample $(\zeta, \boldsymbol{\theta}, P, \hat{\boldsymbol{\mu}})$ from $\mathcal{S}^{(c)}$
$\hat{\boldsymbol{\mu}}$
$\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}_z^{(c)}\|_2$
$\mathcal{Z}^{(c)}$
$g_{\boldsymbol{\theta}}$
$\mathcal{A}(\cdot, \zeta)$
PCA Layer (P)
$\{u_j\}_{j=1}^{M}$
Clip & Average
$\boldsymbol{\mu}_z^{(c)}$
Optimize

Results | Quantitative: Higher Accuracy



	MNIST		FashionMNIST		CIFAR-10	
Method	IPC=10	IPC=50	IPC=10	IPC=50	IPC=10	IPC=50
DM w/o DP	97.8	99.2	84.6	88.7	52.1	60.6
DP-Sinkhorn [2]	31.7 ± 3.2	33.9 ± 1.7	9.8 ± 0.0	22.0 ± 0.1	–	–
DP-MERF [12]	75.0 ± 0.3	84.4 ± 2.3	65.5 ± 3.2	71.3 ± 1.7	–	–
DP-KIP-ScatterNet [29]	25.8 ± 2.1	13.8 ± 2.6	17.7 ± 1.5	16.2 ± 1.2	16.8 ± 1.1	9.5 ± 0.5
PSG [4]	78.6 ± 0.7	–	68.5 ± 0.5	–	33.6 ± 0.3	–
NDPDC [39]	93.1 ± 0.4	94.1 ± 0.4	77.7 ± 0.6	78.8 ± 0.4	39.4 ± 0.8	42.3 ± 0.8
Dosser (ours)	95.3 ± 0.0	96.4 ± 0.0	**81.6** ± 0.1	81.8 ± 0.2	44.2 ± 0.2	49.1 ± 0.5
Dosser (ours) w/ PEA [38]	**96.4** ± 0.0	**96.7** ± 0.1	80.1 ± 0.5	**83.1** ± 0.5	**50.6** ± 0.1	**52.3** ± 0.6

Table 1: Comparison of accuracies *achieved by various methods on three datasets: MNIST, FashionMNIST, and CIFAR-10, evaluated under a privacy budget of* $(1, 10^{-5})$.

Results | Qualitative: Clearer Images

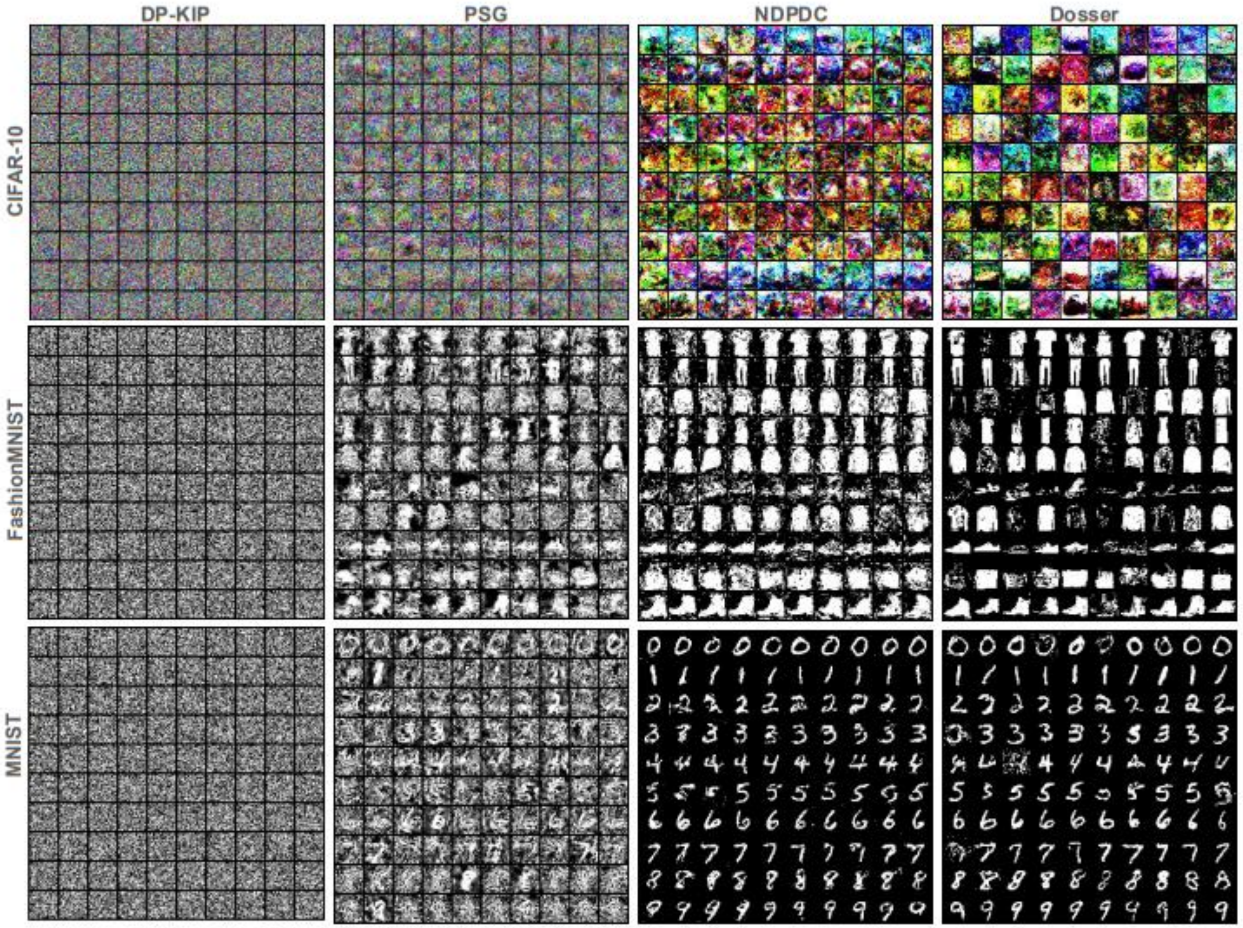


Figure 6: Qualitative Comparison of Distilled Images. *DOSSER generates visually superior and more coherent images compared to prior methods, especially on MNIST and FashionMNIST, demonstrating its effectiveness at preserving data utility.*

Conclusion

Summary:

- DOSSER is a new, noise-efficient framework for private dataset distillation.
- Achieved 8%+ accuracy boost with 1/5 of the dataset size on CIFAR-10.

How:

- Decouples sampling from optimization.
- Uses SER to prioritize important features.

Impact:

Enables higher-quality, compact, private synthetic datasets for responsible AI.