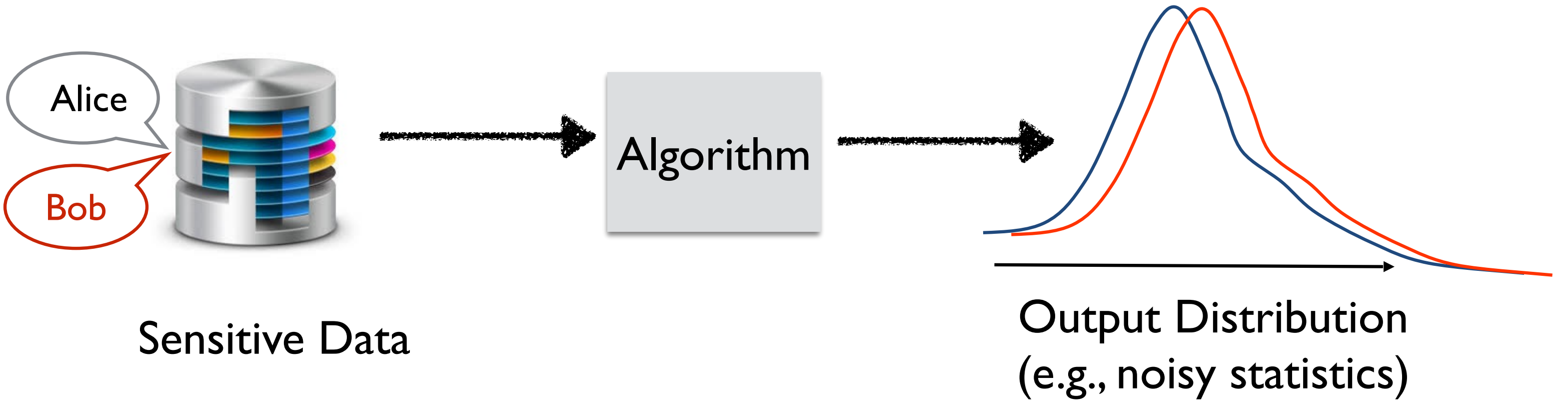
Private Synthetic Data Generation

(Part 2)

Steven Wu

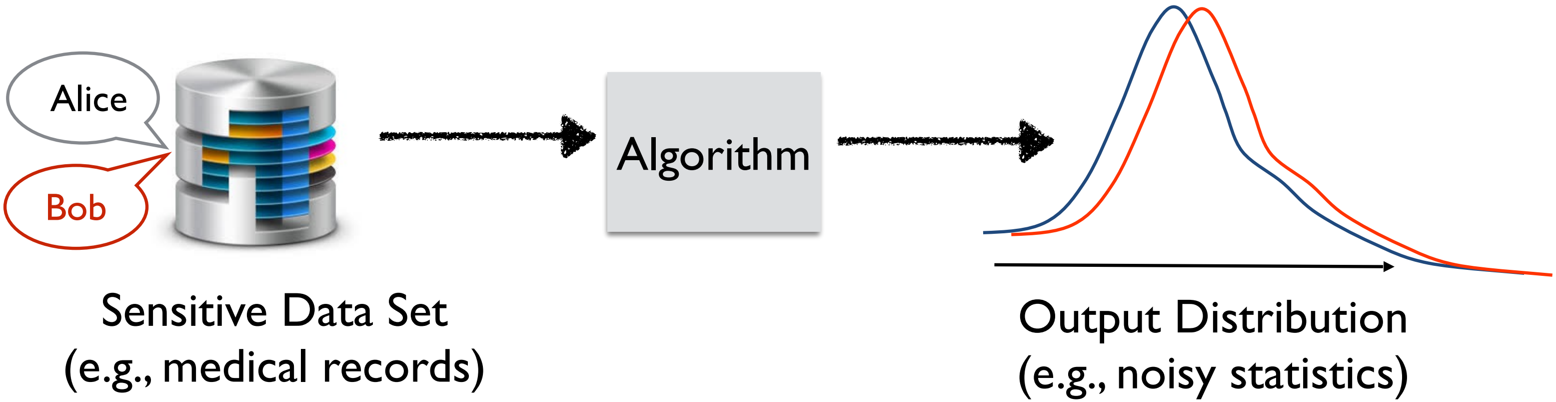
School of Computer Science
Carnegie Mellon University



"An algorithm is *differentially private* if changing a single record does not alter its output distribution by much."
[DN03, DMNS06]

Definition: A (randomized) algorithm A is (ε, δ)-differentially private if for all neighbors D, D' and every $S \subseteq \text{Range}(A)$

$$\Pr[A(D) \in S] \leq e^{\varepsilon} \Pr[A(D') \in S] + \delta$$



"An algorithm is *differentially private* if changing a single record does not alter its output distribution by much."
[DN03, DMNS06]

Challenge:

How can we enable non-experts to work with DP?

Differentially Private Synthetic Data

Sensitive data set
(e.g. medical records)

Synthetic data set
"*Fake*" data records that preserve important statistical properties

Allow arbitrary usage
post - processing

Data Scientist

Synthetic Data Release

1. Synthetic data for query/statistics release
   - A large collection of statistics in mind (e.g. k-way correlations)
2. General-purpose synthetic data
   - Exploratory data analysis
   - Training ML models
   - …

Synthetic Data Release

1. Synthetic data for query/statistics release
   - A large collection of statistics in mind
2. General-purpose synthetic data
   - Exploratory data analysis
   - Training ML models
   - …

Synthetic Data for Statistic/Query Release

Statistical / Counting Query Release

$$D \in (\{0, 1\}^d)^n$$

	Smoke	Lung Cancer	Diabetes	OCD	
patient_id1	1	1	1	1	$q(x) = 1$
patient_id2	1	0	0	1	$q(x) = 0$
patient_id3	1	1	0	1	$q(x) = 1$
patient_id4	0	0	1	0	$q(x) = 0$

$q(D) = 1/2$

Counting query: what is the fraction of people that satisfy some specified property q?

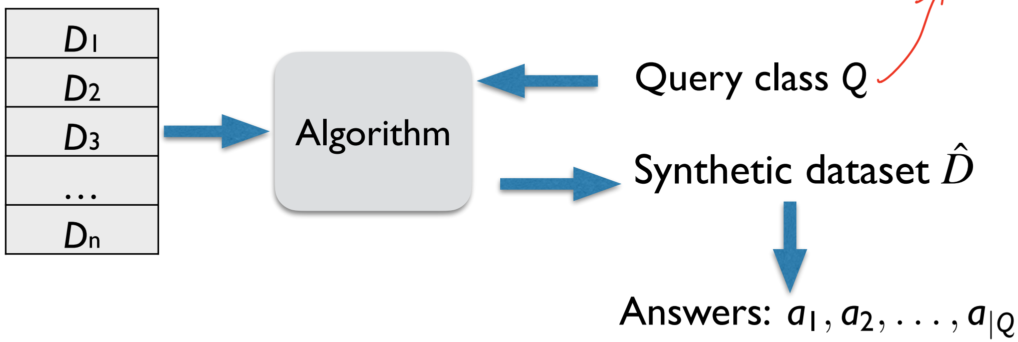
e.g. $q(x)$ = has "Smoke", "Lung Cancer" & "OCD"
(3-way Marginals)

Synthetic Data for Query Release

(used to be x)

Private data set

$\{q_1, \ldots, q_{|Q|}\}$



Goal: "max error" to be small

$\max_{q \in Q} |a_q - q(D)| \to$ small.

Consistency:

For example,

#(smoke & lung cancer) + #(smoke & no lung cancer) = #(smoke)

Does not hold for Laplace / Gaussian

Long Line of Work

- [BLR08, RR10, HR10]
- [HLM12]
- [GGHRW14, ZCPSX14]
- [MSM19]
- [VTBSW20]
- [LVSUW21, ABKKMRS21]
- …

Theoretical Constructs

↓

More Practical Methods

Terrance Liu, Giuseppe Vietri, Z. S. Wu
"Iterative Methods for Private Synthetic Data: Unifying Framework and New Methods"
To appear at NeurIPS 2021

Iterative Framework

w/ Adaptive Measurements

Define some loss function L that measures accuracy

For rounds $t = 1, \ldots, T$

1. SELECT: sample a set of queries Q_t for which the current synthetic dataset has high error

2. MEASURE: release noisy answers A_t for queries in Q_t

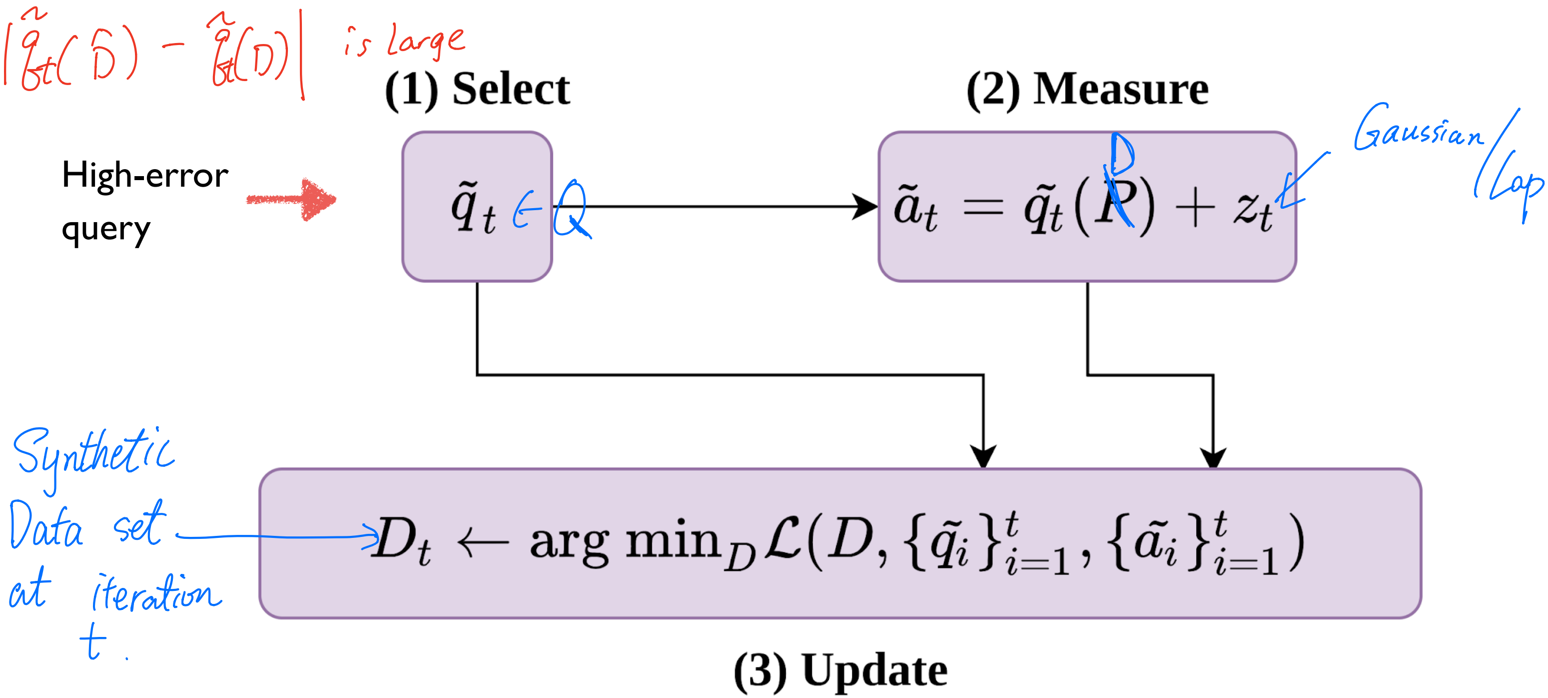
3. UPDATE: update the synthetic dataset to fit the noisy answers A_t according to the loss function L

Iterative Framework

w/ Adaptive Measurements

L: a loss function that measures accuracy

For rounds $t = 1, \ldots, T$



Adaptive Measurements

Restrict the synthetic dataset to belong to some family of distributions $\mathcal{D}$ and initialize $D_0 \in \mathcal{D}$

Adaptive Measurements

Restrict the synthetic dataset to belong to some family of distributions $\mathcal{D}$ and initialize $D_0 \in \mathcal{D}$

Define some loss function $\mathcal{L}$

Adaptive Measurements

Restrict the synthetic dataset to belong to some family of distributions $\mathcal{D}$ and initialize $D_0 \in \mathcal{D}$

Define some loss function $\mathcal{L}$

For **T** rounds (i.e., **t = 1...T**)

Adaptive Measurements

Restrict the synthetic dataset to belong to some family of distributions $\mathcal{D}$ and initialize $D_0 \in \mathcal{D}$

Define some loss function $\mathcal{L}$

For **T** rounds (i.e., **t = 1...T**)

1. **SELECT:** sample a set of queries $\tilde{Q}_t$

Adaptive Measurements

Restrict the synthetic dataset to belong to some family of distributions $\mathcal{D}$ and initialize $D_0 \in \mathcal{D}$

Define some loss function $\mathcal{L}$

For **T** rounds (i.e., **t = 1...T**)

1. **SELECT:** sample a set of queries $\tilde{Q}_t$
2. **MEASURE:** take noisy measurements of each query in $\tilde{A}_t$

Adaptive Measurements

D: private data set

Restrict the synthetic dataset to belong to some family of distributions $\mathcal{D}$ and initialize $\hat{D}_0 \in \mathcal{D}$

Define some loss function $\mathcal{L}$

For **T** rounds (i.e., **t = 1...T**)

Composition

1. **SELECT:** sample a set of queries $\tilde{Q}_t$ ← Exp. Mech. / Report Noisy max
2. **MEASURE:** take noisy measurements of each query in $\tilde{A}_t$ ← Gaussian Mech.
3. **UPDATE:** update the synthetic dataset to fit the noisy measurements according to the loss function $\mathcal{L}$ ← post-processing

$$\hat{D}_t \leftarrow \mathcal{L}\left(\hat{D}_{t-1}, \tilde{Q}_t, \tilde{A}_t\right)$$

Adaptive Measurements

Under this framework, existing algorithms can be reduced to selections of $\mathcal{D}$ and $\mathcal{L}$

Examples:

- **MWEM** (Hardt et al., 2012)
- **DualQuery** (Gaboardi et al., 2014)
- **FEM** (Vietri et al., 2020)
- **RAP$^{\text{softmax}}$**
  - Adapted from RAP (Aydore et al., 2021)

Two-Player Zero-Sum Game

Private data D

Synthetic Data Player

Init $D^{(0)}$

Query Player.

$\leftarrow q^{(1)}$

$|q^{(1)}(D^{0}) - q^{(1)}(D)|$ is Large

$\tilde{a}^{(1)} = q^{(1)}(D) + \text{Noise}$

$t=1$:

$D^{(1)} \leftarrow \text{Update}\left(D^{(0)}, q^{(1)}, \tilde{a}^{(1)}\right)$

$t=2$:

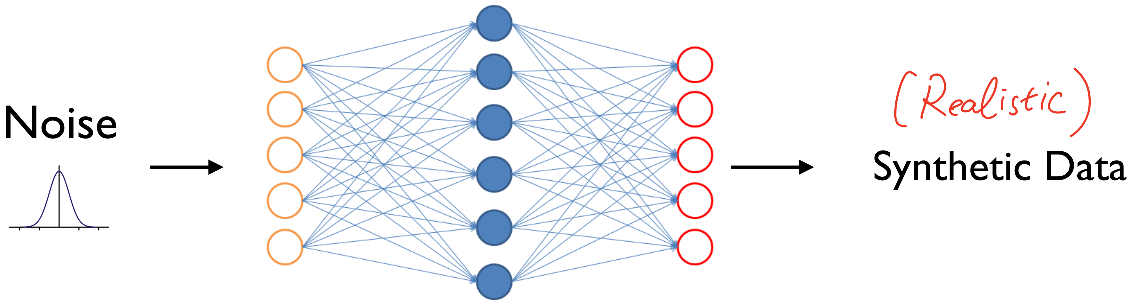
$D^{(2)} \leftarrow \text{Update}\left(D^{(1)}, \{q^{(1)}, q^{(2)}\}, \{\tilde{a}^{(1)}, \tilde{a}^{(2)}\}\right)$

⋮

New Method: GEM

Generative Networks with the Exponential Mechanism

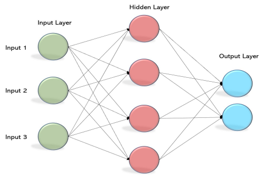
- Inspired by Generative Adversarial Nets (GAN)
- Optimize L_1 loss over synthetic data distribution specified by a generative neural network



Generative networks with the exponential mechanism

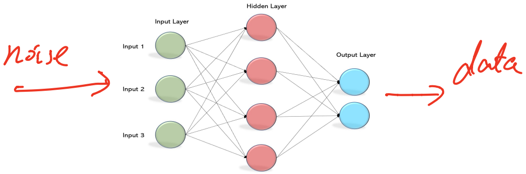
- Inspired by GAN architectures (two-player game)

GEM



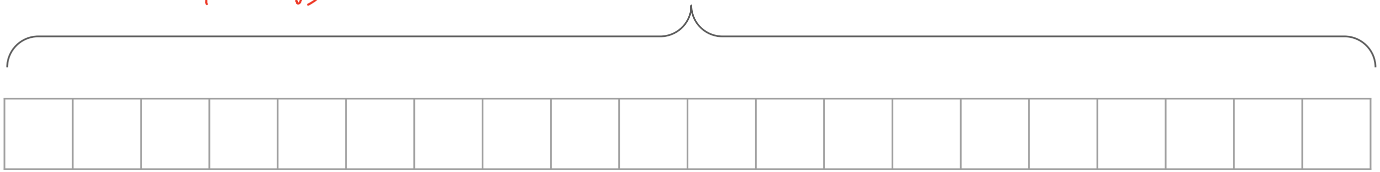
$\mathbf{z} \sim \mathcal{N}(0, I)$ → G_θ

GEM



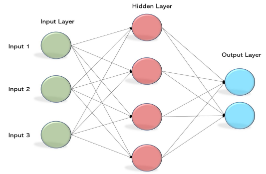
$\mathbf{z} \sim \mathcal{N}(0, I) \longrightarrow G_\theta$

(Not for privacy)

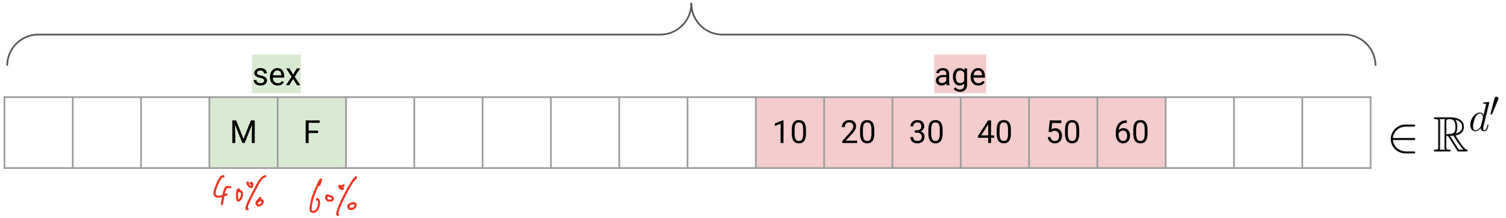


$\in \mathbb{R}^{d'}$

GEM

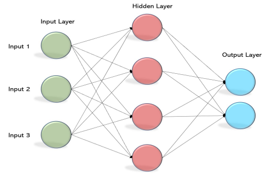


$\mathbf{z} \sim \mathcal{N}(0, I) \longrightarrow G_\theta$

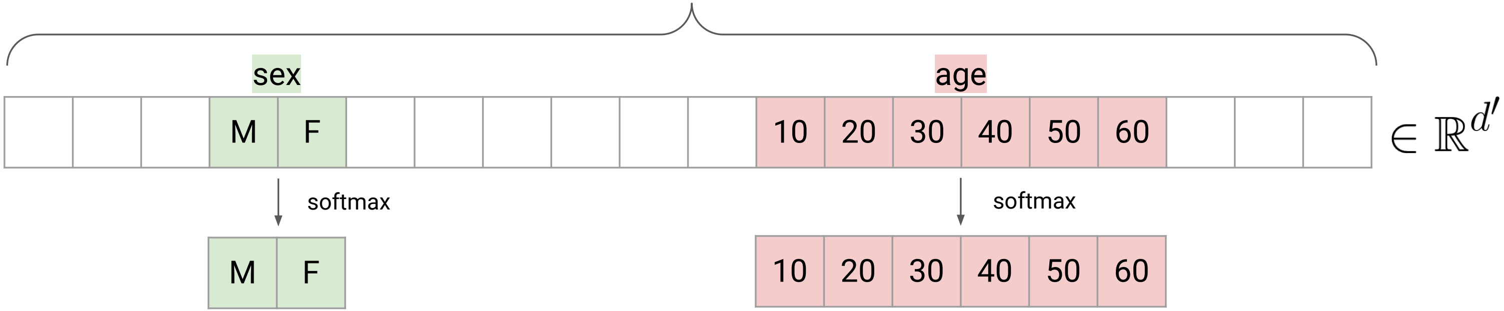


Output layer.

GEM



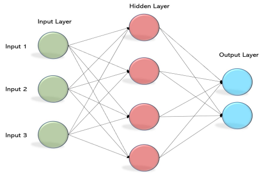
$$\mathbf{z} \sim \mathcal{N}(0, I) \longrightarrow G_\theta$$



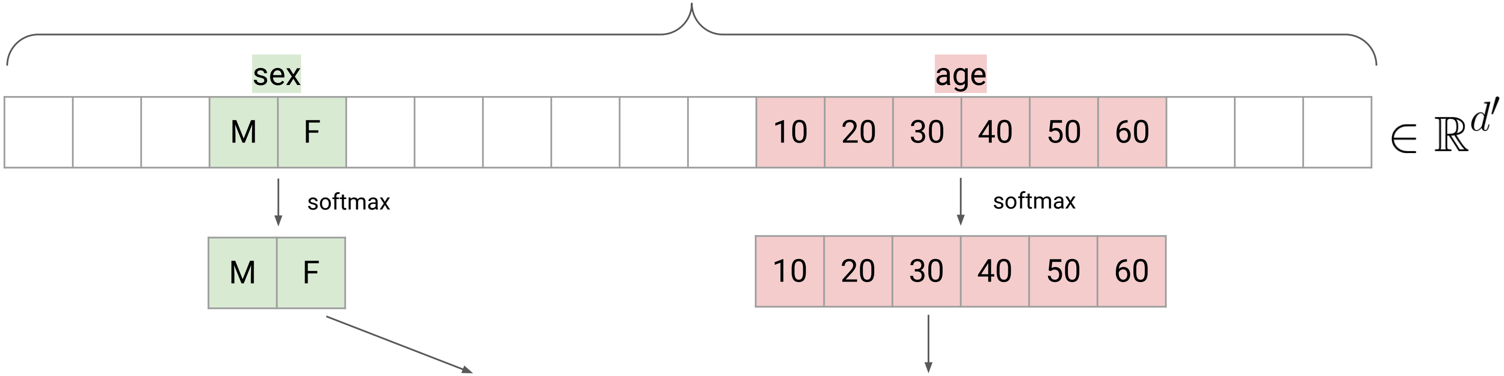
Model the **marginal distribution** of each attribute

$$P(\text{age} = 30)$$

GEM



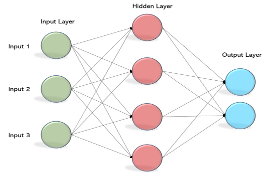
$$\mathbf{z} \sim \mathcal{N}(0, I) \longrightarrow G_\theta$$



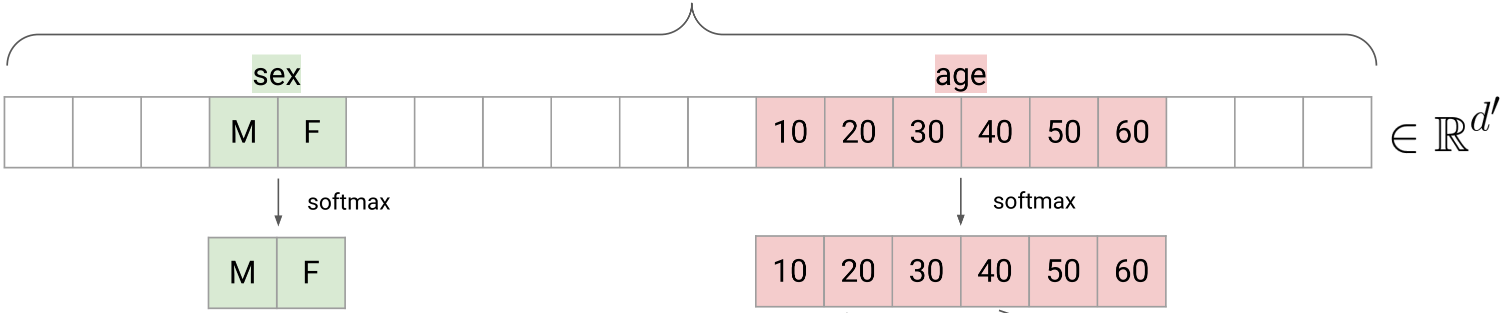
$$P(\text{sex} = \text{F}) \times P(\text{age} = 30)$$

K-way marginals: What percentage of people are **female** and **30** years old?

GEM



$\mathbf{z} \sim \mathcal{N}(0, I) \longrightarrow G_\theta$



$$P(\text{age} = 20) + P(\text{age} = 30) + P(\text{age} = 40)$$

Range/Prefix queries: What percentage of people are **between** the ages of **20 and 40**?

GEM

- Under the **Adaptive Measurements** framework
  - $\mathcal{D}$ is the set of some **mixture of product distributions** that can be represented by a neural network G_θ

GEM

- Under the **Adaptive Measurements** framework
  - $\mathcal{D}$ is the set of some **mixture of product distributions** that can be represented by a neural network G_θ
    - We denote the our synthetic data distribution $P_\theta = G_\theta(\mathbf{z})$

GEM

- Under the **Adaptive Measurements** framework
  - $\mathcal{D}$ is the set of some **mixture of product distributions** that can be represented by a neural network G_θ
    - We denote the our synthetic data distribution $P_\theta = G_\theta(\mathbf{z})$
  - We choose $\mathcal{L}$ to simply be the **L1 loss** over the noisy measurements

$$\mathcal{L}^{\text{GEM}}\left(\theta, \widetilde{Q}_{1:t}, \widetilde{A}_{1:t}\right) = \sum_{i=1}^{t} \left|\widetilde{q}_i(P_\theta) - \widetilde{a}_i\right|$$

Neural Network parameter

Measured Queries

Queary measurements

answers from NN

measured noisy answers

Run SGD over θ to optimize $\mathcal{L}^{GEM}$

GEM

- Under the **Adaptive Measurements** framework
  - $\mathcal{D}$ is the set of some **mixture of product distributions** that can be represented by a neural network G_θ
    - We denote the our synthetic data distribution $P_\theta = G_\theta(\mathbf{z})$
  - We choose $\mathcal{L}$ to simply be the **L1 loss** over the noisy measurements

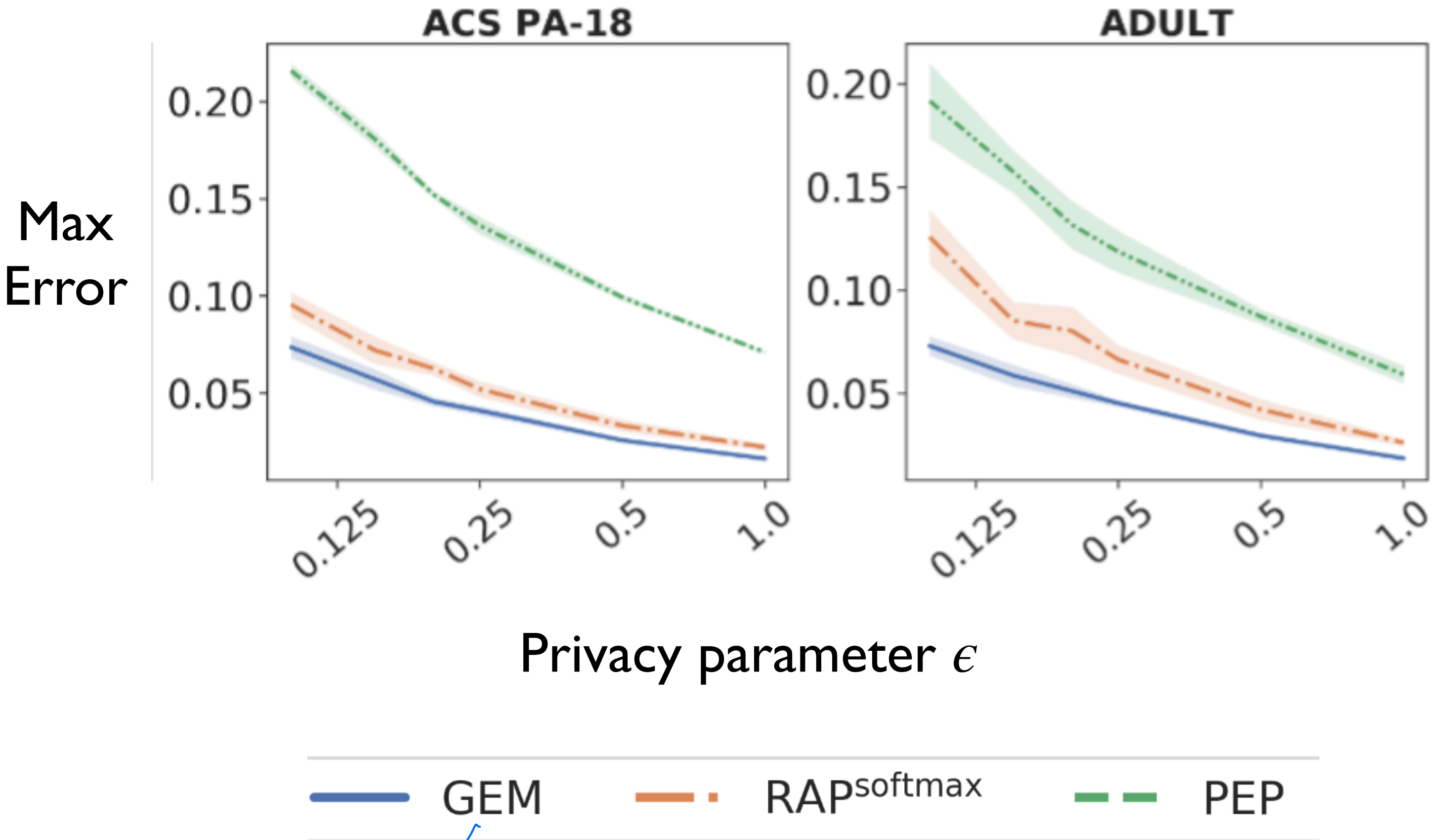
$$\mathcal{L}^{\text{GEM}}\left(\theta, \widetilde{Q}_{1:t}, \widetilde{A}_{1:t}\right) = \sum_{i=1}^{t} \left|\widetilde{q}_i(P_\theta) - \widetilde{a}_i\right|$$

K-way marginal queries: (product queries) $\tilde{q}_i(P_\theta) = \prod_j G_\theta(\mathbf{z})_j$

Empirical Evaluations

Generating synthetic data for

- American Community Survey (ACS) micro-data released in previous years
- Adult dataset (Older Census data)



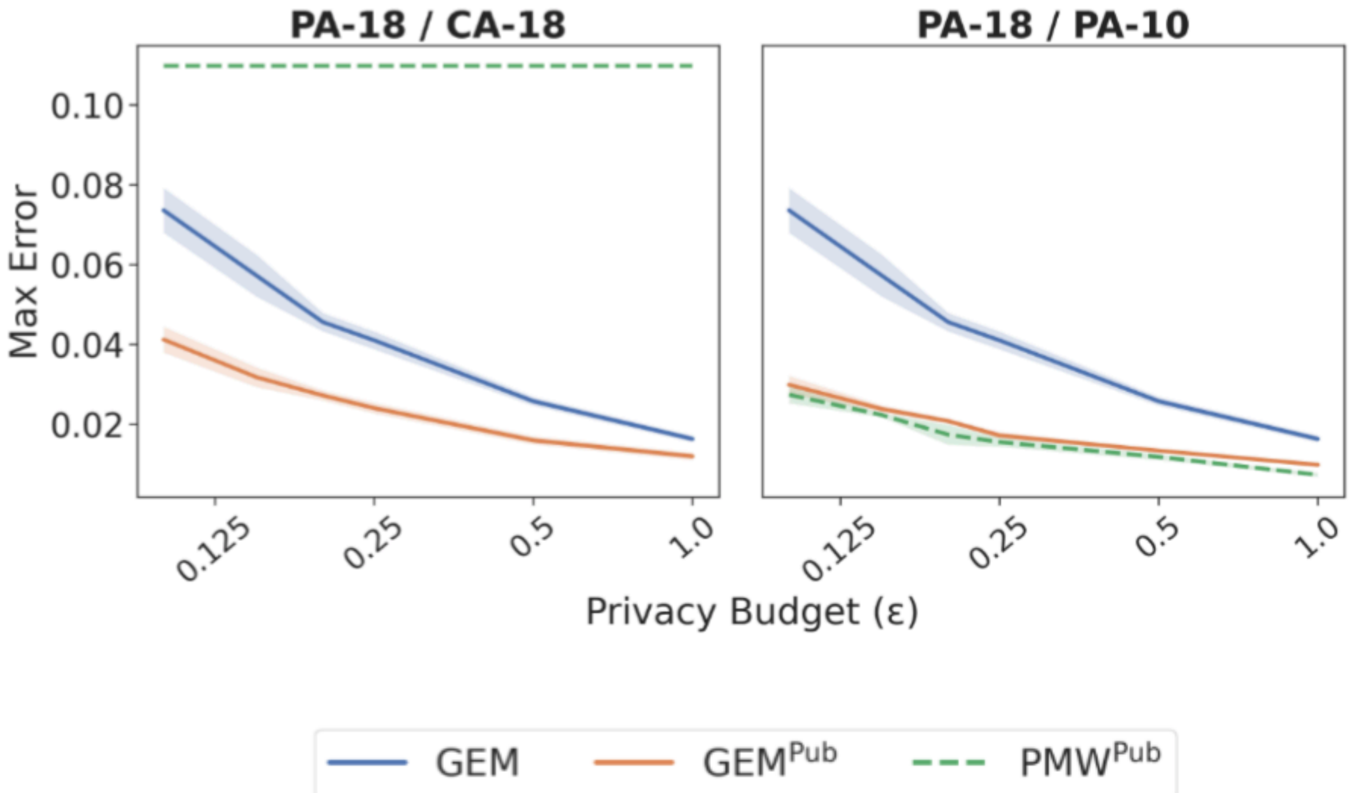
Generative net w/ Exp Mech.

Leveraging Publicly Available Data Sets

Example:

- Target dataset: ACS data of some state (e.g., PA) in 2018
- Pre-train GEM on a related but different data set:
  - ACS-PA in 2010
  - ACS-CA in 2018

Private Data/Public Data



General-purpose synthetic data with deep generative models

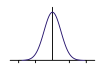
Generative Adversarial Nets (GANs) [GPM+14]

2-Player Zero-Sum Game

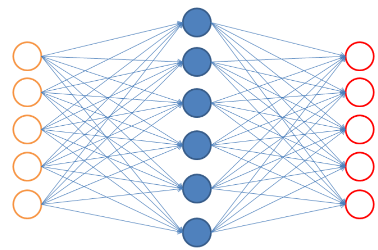
Generator G: mimic the real data

Noise

z



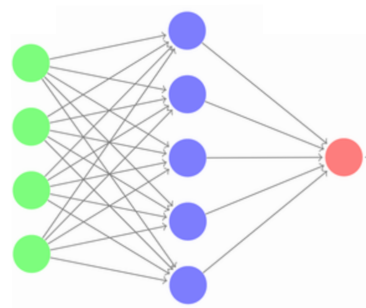
→



→ Synthetic data x

Discriminator D: distinguish real and fake data

Real/Fake Examples x →



→ Probability of "real"

Wasserstein GAN [ACB17]

$$\min_G \max_D \mathbb{E}_{x\sim p_X}[D(x)] + \mathbb{E}_{z\sim p_z}[1 - D(G(z))]$$

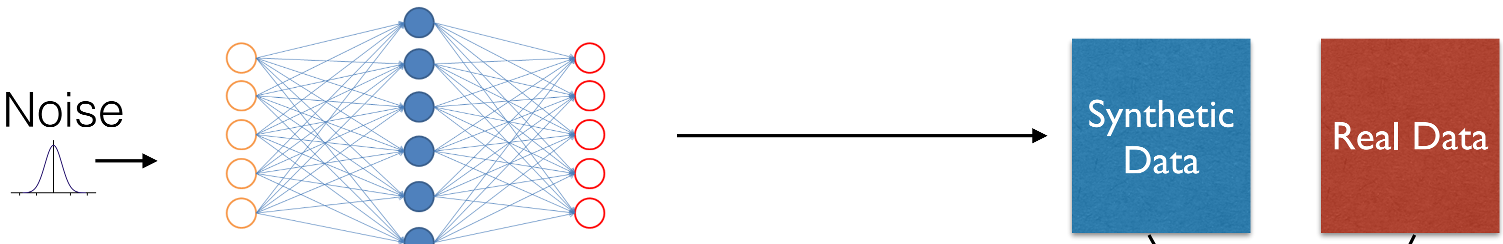
Approach
Generative adversarial nets (GANs)
+ Differential privacy

DP GANs Support Clinical Data Sharing [BWWLBBG]

Published in *Circulation: Cardiovascular Quality and Outcomes 2019*

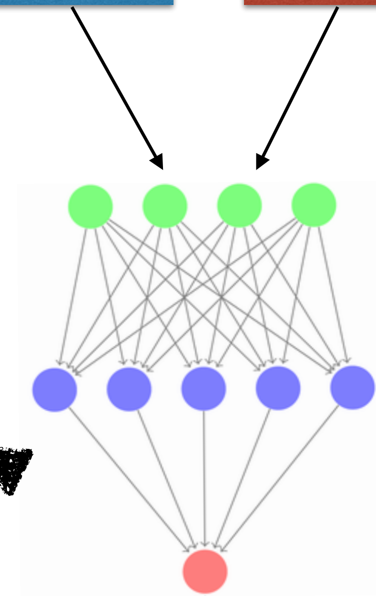
Also in [XLWWZ18], [YJS19],[TKP20], [TWBSC20]…

Private GAN Training



Training Generator:

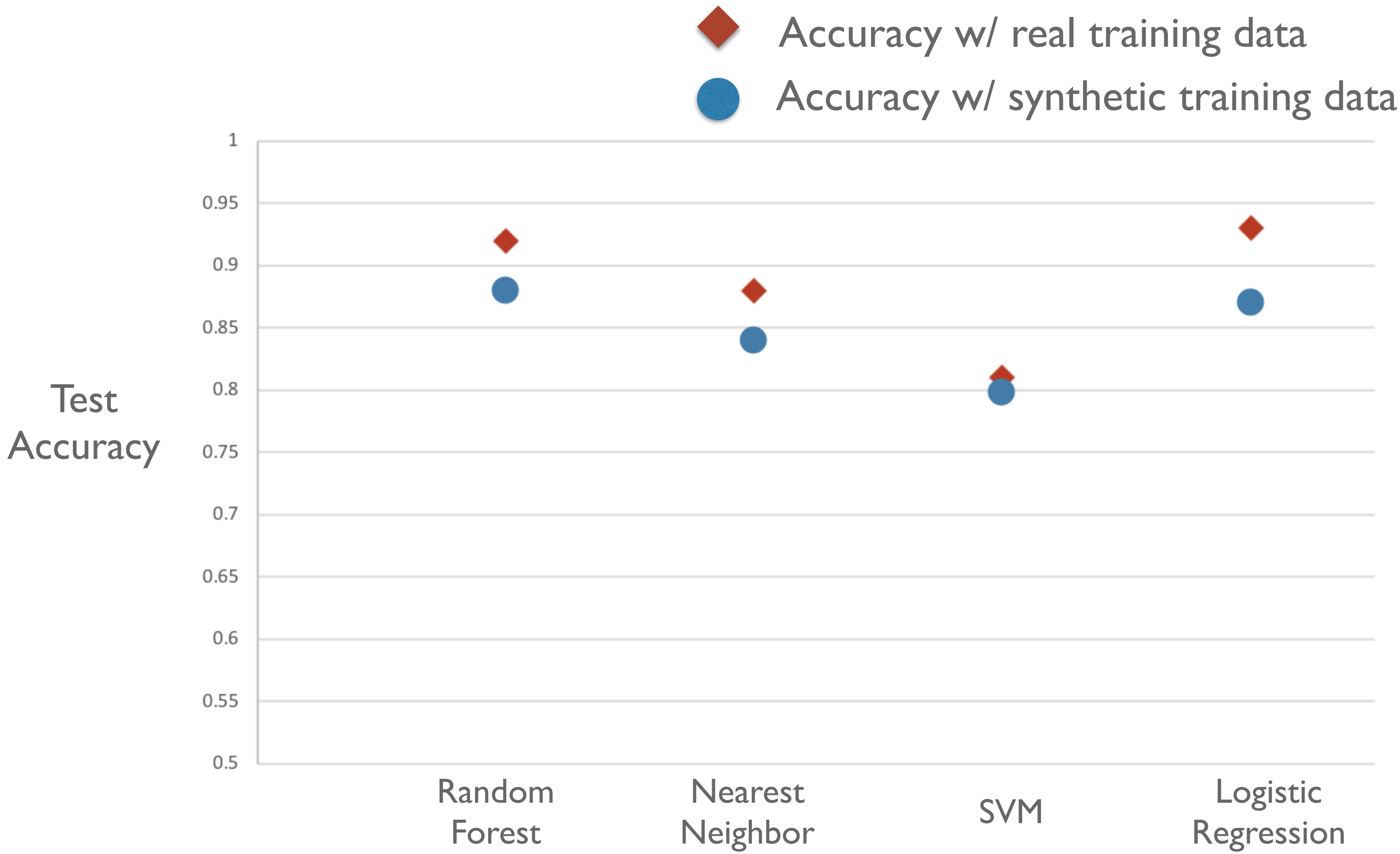
- Does not directly interact with real data
- Train using standard (non-private) methods (e.g., SGD)   *Post-Processing*



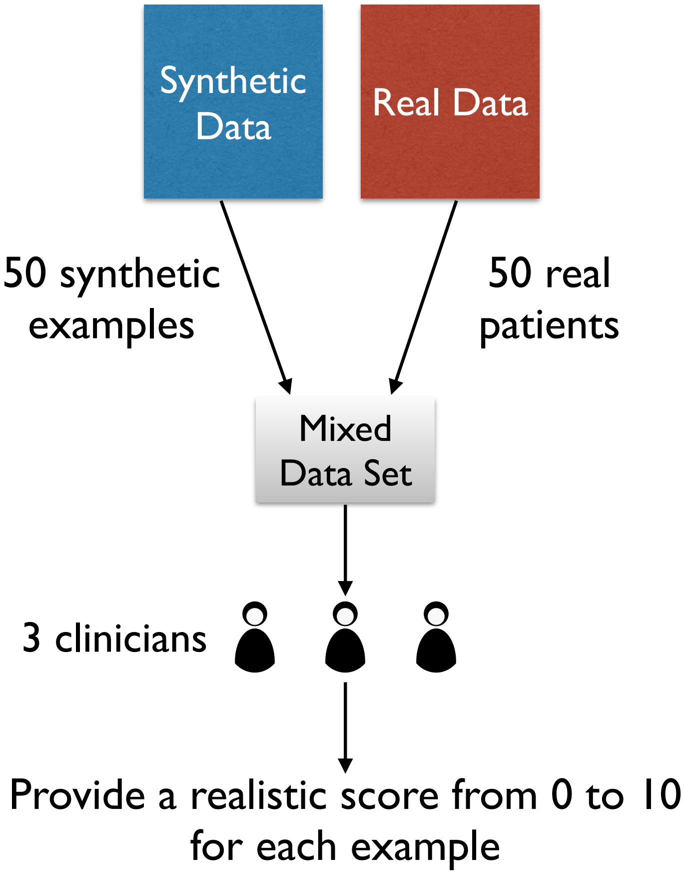
Privately Training Discriminator:

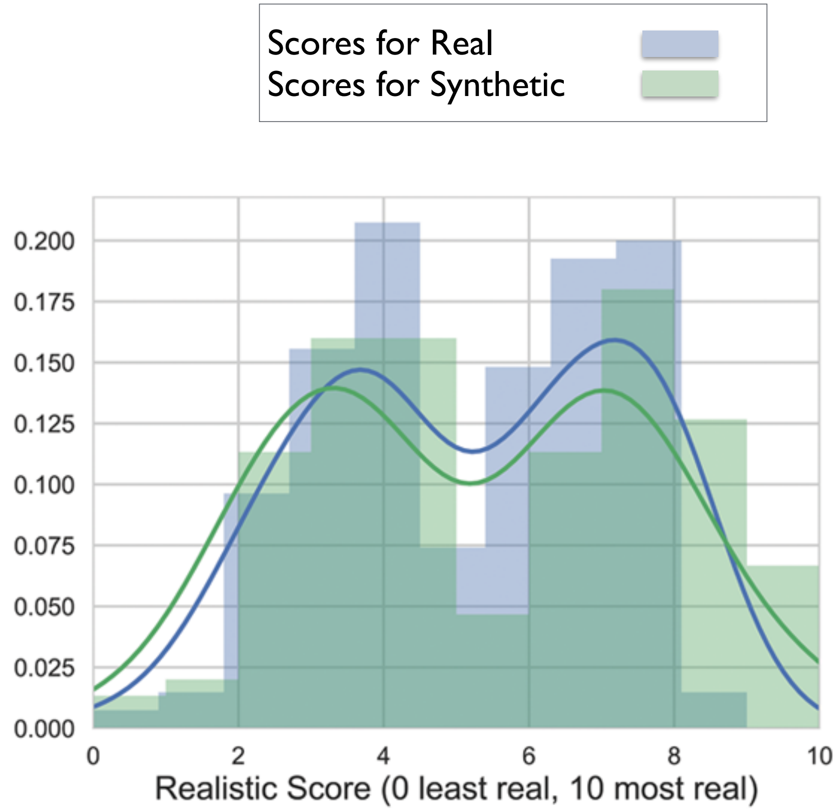
- Interacts with real data
- Train using DP method such as DP-SGD

Models Trained on Synthetic v.s. Real Data



Evaluation with Human (Discriminators)





Difficult to Reach Convergence

- Training produces a sequence of (generator, discriminator) $(G_1, D_1), \ldots, (G_T, D_T)$
- The last generator G_T often gives poor synthetic data distribution
- But mixture of generators can provide good synthetic data [BWWLBBG19]

Synthetic Data Release

1. Synthetic data for query/statistics release
   - A large collection of statistics in mind
2. General-purpose synthetic data
   - Exploratory data analysis
   - Training ML models
   - …

Simulate a two-player zero-sum game

Private Post-GAN Boosting

[NWD] ICLR21

- The entire sequence $(G_1, D_1), \ldots, (G_T, D_T)$ satisfy DP
- Compute a mixture over $\{G_1, \ldots, G_T\}$

Post-GAN Zero-Sum Game

Approximate each generator G_t by taking r samples;
Let B be the entire set of the rT examples

Data player	Query player
distribution ϕ over B	distribution over $\{D_1, \ldots, D_T\}$

$$\min_{\phi} \max_{D_j} U(\phi, D_j) \equiv \mathbb{E}_{x \sim P_X}[D_j(x)] + \mathbb{E}_{x \sim \phi}[(1 - D_j(x))]$$

Post-GAN Equilibrium

DP GAN + MWEM

Over rounds $t = 1, \ldots, T$

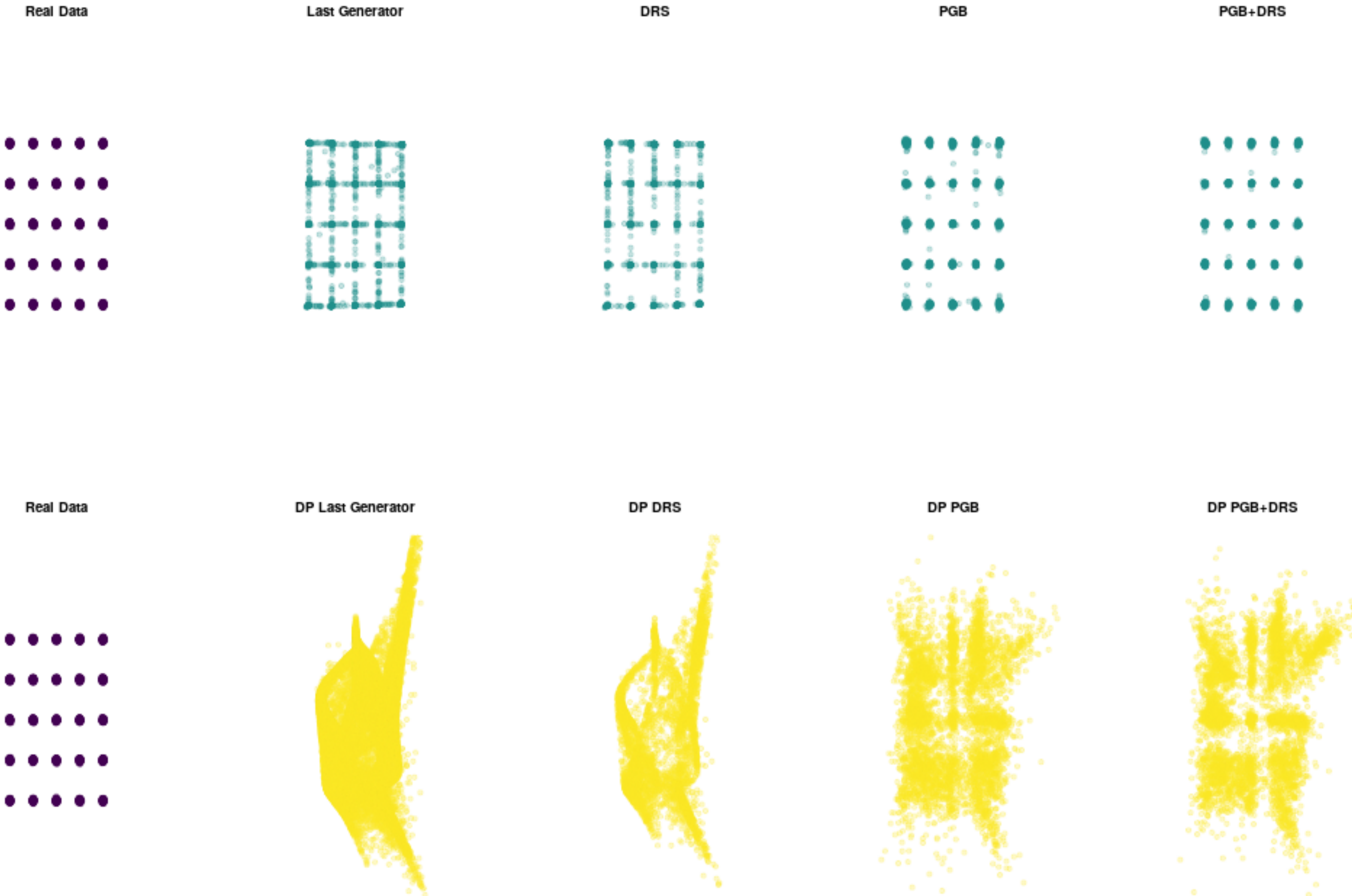
Data player
runs MW to update distribution ϕ over B

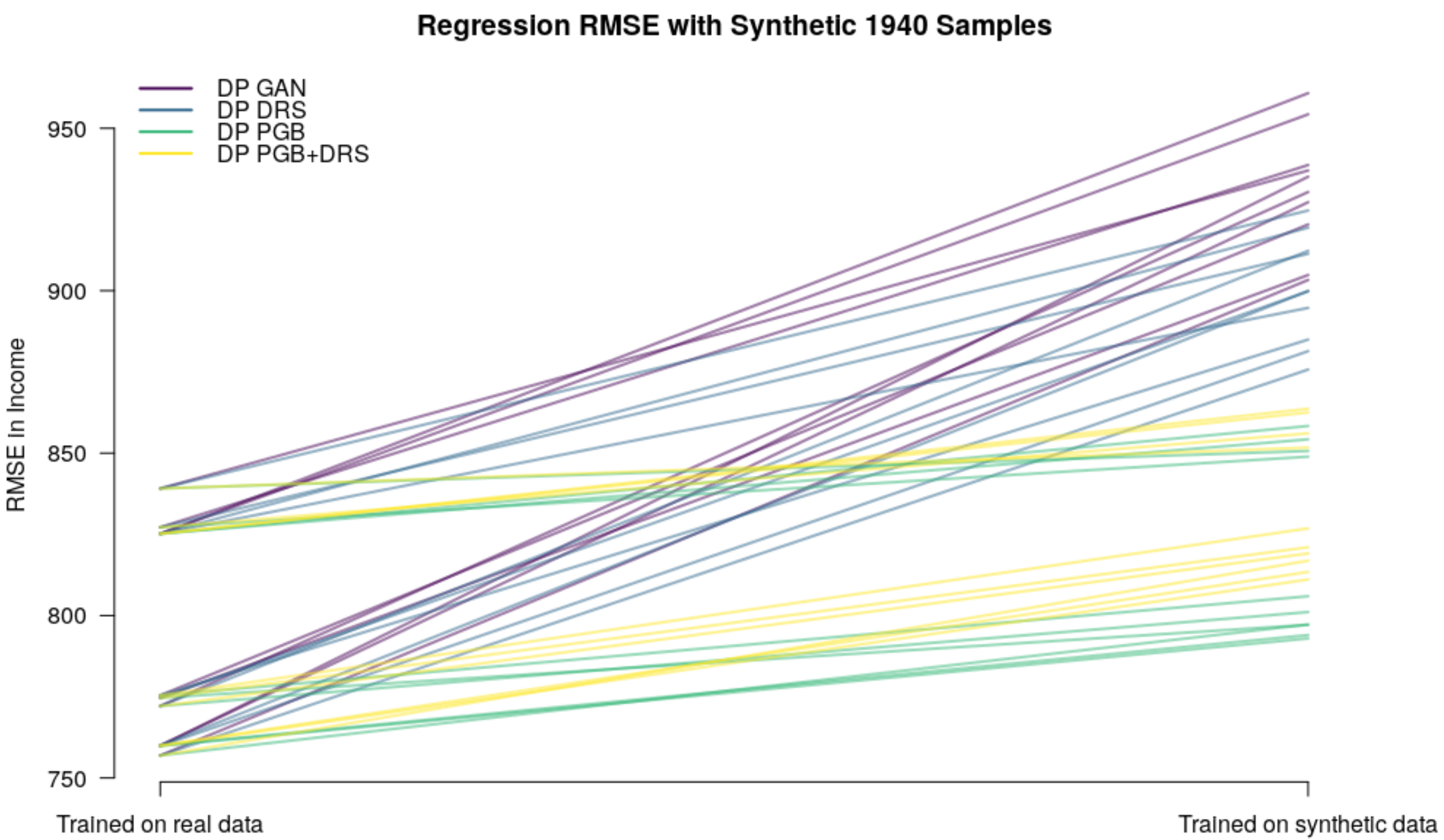
Query player
uses exponential mech to select a useful discriminator

↓

Approximate equilibrium:
ϕ synthetic data distribution over B; D mixture discriminator

Rejection sampling:
Use D to improve ϕ by "rejecting" unlikely samples


Real Data
Last Generator
DRS
PGB
PGB+DRS
Real Data
DP Last Generator
DP DRS
DP PGB
DP PGB+DRS



Regression RMSE with Synthetic 1940 Samples
DP GAN
DP DRS
DP PGB
DP PGB+DRS
RMSE in Income
950
900
850
800
750
Trained on real data
Trained on synthetic data

Train ML models on synthetic data and Test them on real out-of-sample data

	GAN	DRS	PGB	PGB + DRS
Logit Accuracy	0.626	0.746	0.701	**0.765**
Logit ROC AUC	0.591	0.760	0.726	**0.792**
Logit PR AUC	0.483	0.686	0.655	**0.748**
RF Accuracy	0.594	0.724	0.719	**0.742**
RF ROC AUC	0.531	0.744	0.741	**0.771**
RF PR AUC	0.425	0.701	0.706	**0.743**
XGBoost Accuracy	0.547	0.724	0.683	**0.740**
XGBoost ROC AUC	0.503	0.732	0.681	**0.772**
XGBoost PR AUC	0.400	0.689	0.611	**0.732**

	DP GAN	DP DRS	DP PGB	DP PGB +DRS
Logit Accuracy	0.566	0.577	0.640	**0.649**
Logit ROC AUC	0.477	0.568	0.621	**0.624**
Logit PR AUC	0.407	0.482	0.532	**0.547**
RF Accuracy	0.487	0.459	0.481	**0.628**
RF ROC AUC ROC AUC	0.512	0.553	0.558	**0.652**
RF PR AUC PR AUC	0.407	0.442	0.425	**0.535**
XGBoost Accuracy	0.577	0.589	0.609	**0.641**
XGBoost ROC AUC	0.530	0.586	**0.619**	0.596
XGBoost PR AUC	0.398	0.479	0.488	**0.526**

Summary

- Zero-sum game view on synthetic data
- Recovers classical methods and allows reconfigurations that leverage heuristics solvers
  - MWEM → FEM / DualQuery
- Combine classical methods with deep learning methods
  - Private Post-GAN boosting: DP-GAN + MWEM