Private Synthetic Data Generation

Steven Wu

Assistant Professor
Carnegie Mellon University

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
   - …

This Lecture

- Synthetic data for query release
- General-purpose synthetic data

"Everything is a zero-sum game."

— Jonathan Ullman

and probably Adam Smith

Synthetic Data for Statistic/Query Release

Counting Query Release

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

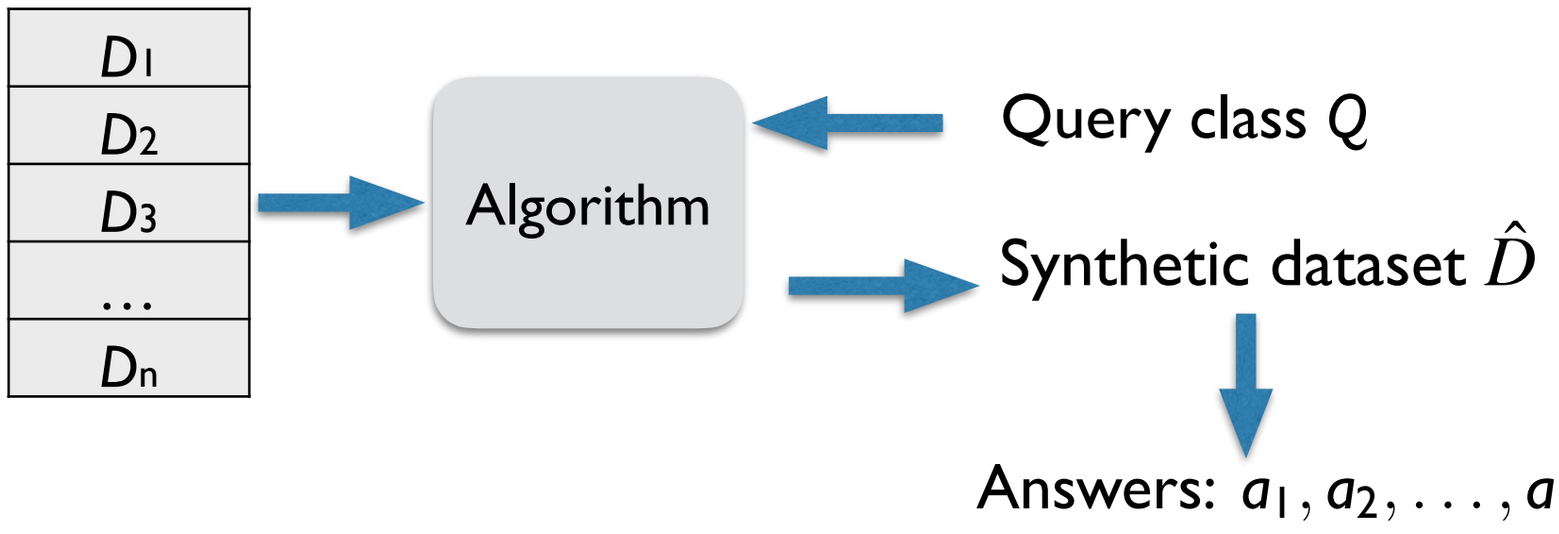
	Smoke	Lung Cancer	Diabetes	OCD	
patient_id1	1	1	1	1	$q(x) = 1$
patient_id2	1	0	0	1	$q(x) = 0$
patient_id3	1	1	0	1	$q(x) = 1$
patient_id4	0	0	1	0	$q(x) = 0$

$q(D) = 1/2$

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



α-*accurate* if
$|q(D) - a_q| \leq \alpha$ for every $q \in Q$

Consistency:
For example,
#(smoke & lung cancer) + #(smoke & no lung cancer) = #(smoke)

A Zero-Sum Game View

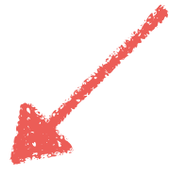
- Equilibrium corresponds to an accurate solution
- Computing equilibrium using no-regret learning algorithms
- Reconfigure the prior approach to get computational efficiency

Zero-Sum Game Formulation

Data player
actions: records in X

(Synthetic) Data distribution $\hat{D}$ over domain X

Query player
actions: queries in Q



Distribution over queries Q

"Error" payoff for $(\hat{D}, q)$:

$$U(\hat{D}, q) = q(\hat{D}) - q(D)$$

Data player wants to minimize and Query player wants to maximize

When Q is closed other negations ($q \in Q \Rightarrow 1 - q \in Q$),
$\max_q U(\hat{D}, q)$ captures the max-error of $\hat{D}$

Approximate Equilibrium Implies Accuracy

Theorem. In an α-approximate equilibrium, the synthetic data distribution satisfies:

$$\max_{q \in Q} | q(\hat{D}) - q(D) | \leq \alpha$$

Output $\hat{D}$ as the synthetic data

How do we compute a minimax strategy privately?

Equilibrium via No-Regret Learning

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

Data player

Learn a synthetic data distribution $\hat{D}^t$ that (approximately) minimize $U(\hat{D}, q^1) + \ldots + U(\hat{D}, q^{t-1})$

Query player

Find a high-error query q^t for $\hat{D}^t$

"No-Regret" Property: suffer cumulative payoff which is "not much worse" than the cumulative payoff of the best fixed strategy

Average plays converge to an approximate equilibrium [FS97]

Prior Approach

MWEM [HR10, HLM12]

Data player
Explicitly maintain a distribution over the domain X using multiplicative weights (MW) [LW89, Vov90, KW94, FS97]

vs.

Query player
find a query with high payoff using exponential mechanism:

- MWEM: statistically optimal [BUV14]
  - For α-accuracy, $n \gtrsim d^{1/2} \log |Q| / (\varepsilon \alpha^2)$
- Maintaining an exponential-sized distribution ⇒ exponential run-time
- For statistical optimality, *worst-case* run-time must be exponential in d [DNRRV09, UV11, Ull13]

How to overcome the computational bottleneck?

Instead of maintaining a exponential size distribution,
Data player solves hard optimization problems

Can then leverage sophisticated solvers
(e.g., integer program solvers CPLEX, Gurobi)

The "Dual" approach

- Prior approach: MWEM [HR10, HLM12]

Data player	vs.	Query player
Run MW over the domain X (Exponential size)		Best response: find a query with high payoff (Tractable problem)

- Our *Dual* Approach: DualQuery [GGHR*W*] ICML14

Query player	vs.	Data player
Run MW over the query class Q (Size scales with $\|Q\|$)		Best response: find a record with small payoff (Intractable problem)

New computational bottleneck

Data Player's Optimization Problem

- Sample queries $q_1, q_2, \ldots, q_s$ from query distribution (for privacy)

- Pick a record to minimize the average payoff over $q_1, q_2, \ldots, q_s$:

$$\min_{x \in X} [(q_1(x) - q_1(D)) + \ldots + (q_s(x) - q_s(D))]$$

But D is fixed, so equivalent to

$$\min_{x \in X} [q_1(x) + \ldots + q_s(x)]$$

- Pure optimization problem: can be solved without privacy

- In general, an intractable problem (MAXCSP)

- Several query classes (e.g. k-way marginals, parities) give integer program formulation. We can use highly optimized solvers (e.g. CPLEX, Gurobi)

The "Primal" Approach

Replace MW by methods that can leverage heuristics solvers:
Follow-the-perturbed-leader (FTPL)[KV05, SKS16, SN19]

- Our approach: FEM (FTPL w/ exp mech.) [VTBSW] ICML20

Data player
Run FTPL over the domain X
Can be computed by solvers

vs.

Query player
Best response: find a query with high payoff
(Tractable problem)

FTPL for Data Player

FTPL optimization: given $q_1, \ldots, q_{t-1}$ from the Query player

$$\min_{x \in X} [q_1(x) + \ldots + q_{t-1}(x) + \langle \sigma, x \rangle]$$

where σ is a random vector drawn from exponential distribution

Can also be solved with an integer program solvers for *k*-way marginals without using the private data D

Theoretical Guarantees

α: target accuracy

ε: privacy loss

n: sample size

$|Q|$: # queries

Prior approach (always exp time)

- MWEM [HR10, HLM12]:

$$\alpha \lesssim \frac{d^{1/4}\log^{1/2}|Q|}{(n\varepsilon)^{1/2}}$$

Our approach that uses integer program solvers [VTBSW20]

- (Improved) DualQuery: $\alpha \lesssim \frac{d^{1/5}\log^{3/5}|Q|}{(n\varepsilon)^{2/5}}$
- FTPL with Exp Mech (FEM): $\alpha \lesssim \frac{d^{3/4}\log^{1/2}|Q|}{(n\varepsilon)^{1/2}}$

Theoretical Guarantees

α: target accuracy

ε: privacy loss

n: sample size

$|Q|$: # queries

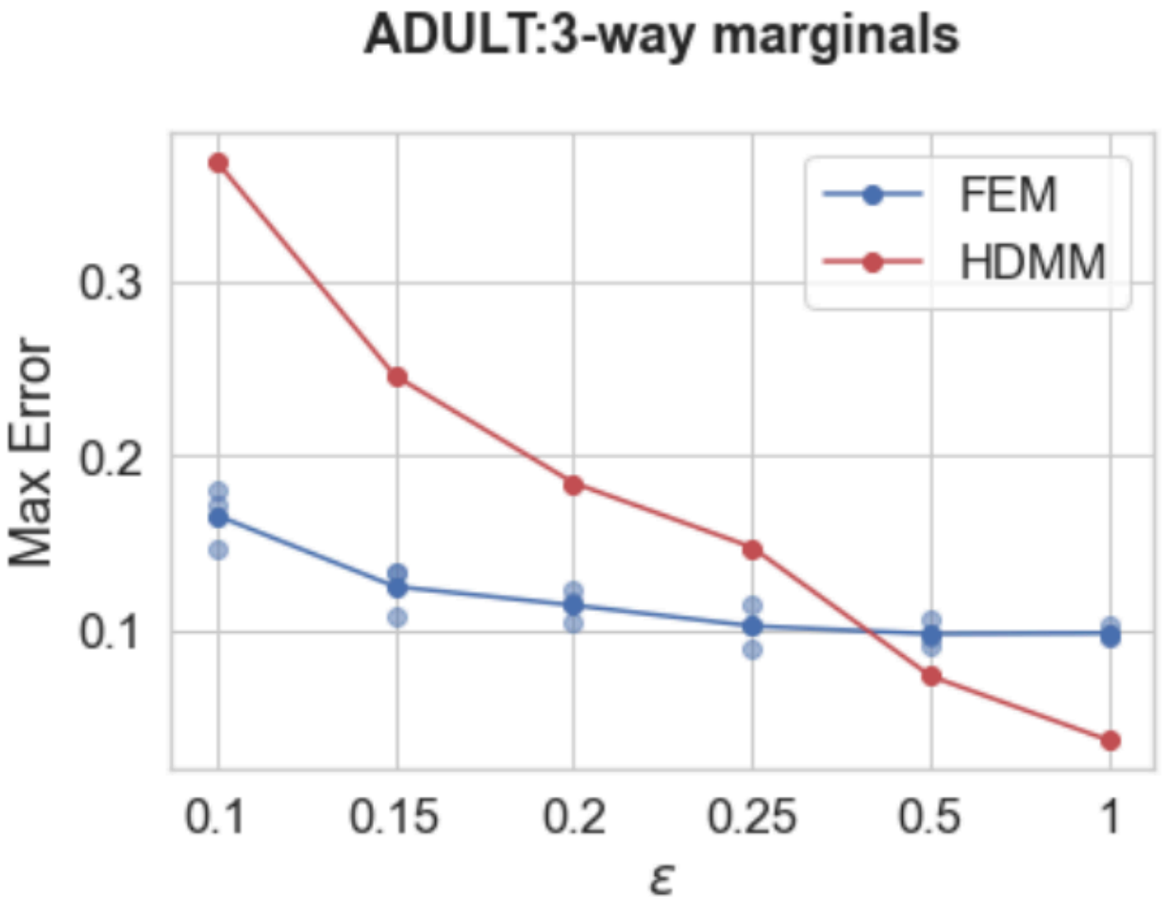
- HDMM [MMHM18]:

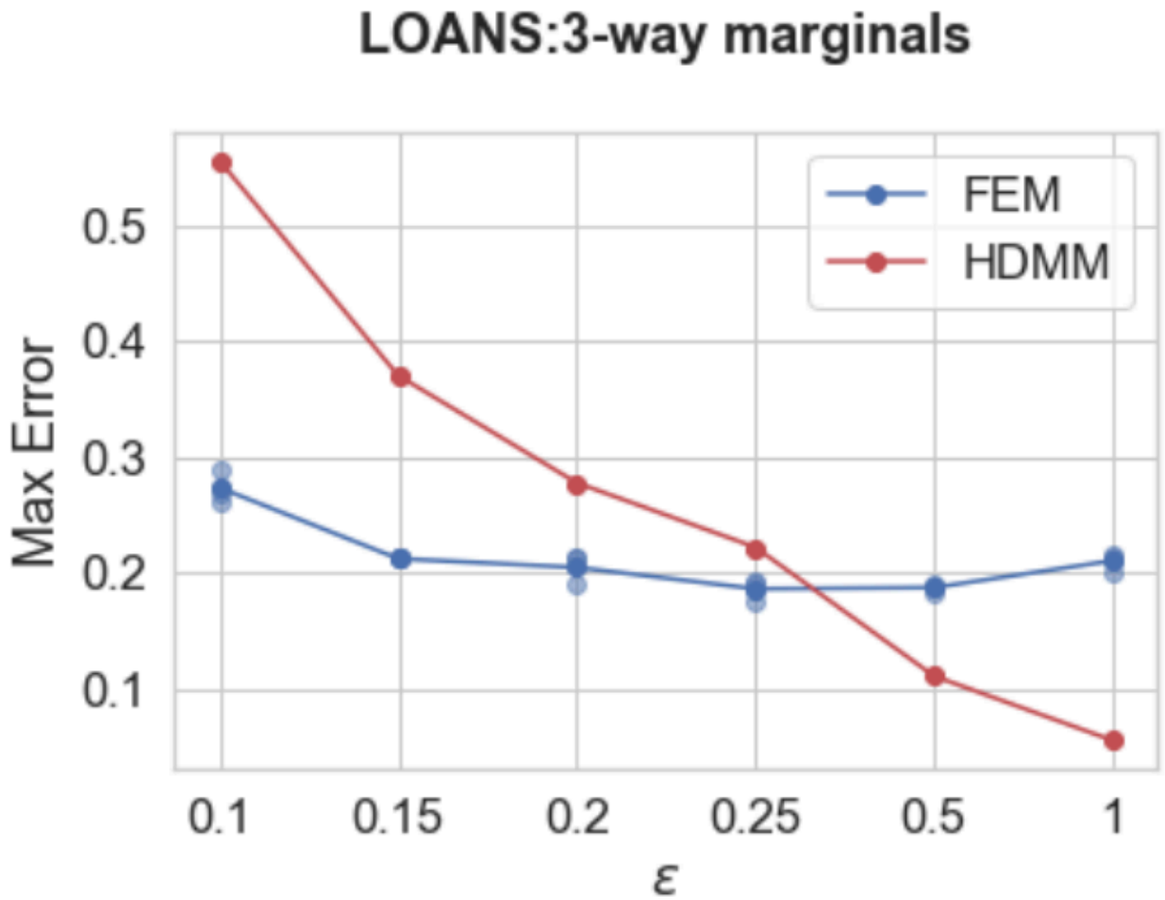
$$\ell_2 \text{ error} \lesssim \frac{\text{Factorization norm of } Q}{n\epsilon}$$

Our approach that uses integer program solvers [VTBSW20]

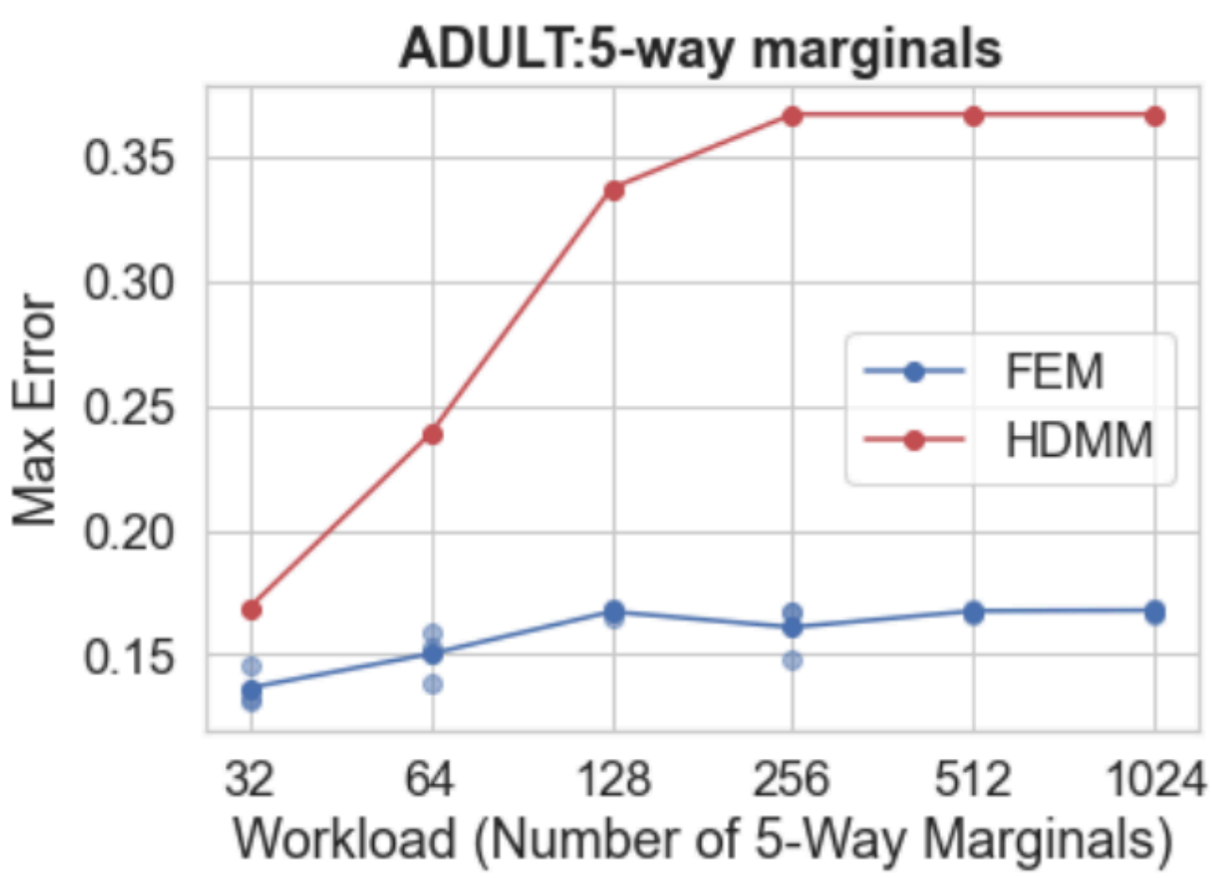
- (Improved) DualQuery: $\alpha \lesssim \frac{d^{1/5}\log^{3/5}|Q|}{(n\varepsilon)^{2/5}}$
- FTPL with Exp Mech (FEM): $\alpha \lesssim \frac{d^{3/4}\log^{1/2}|Q|}{(n\varepsilon)^{1/2}}$

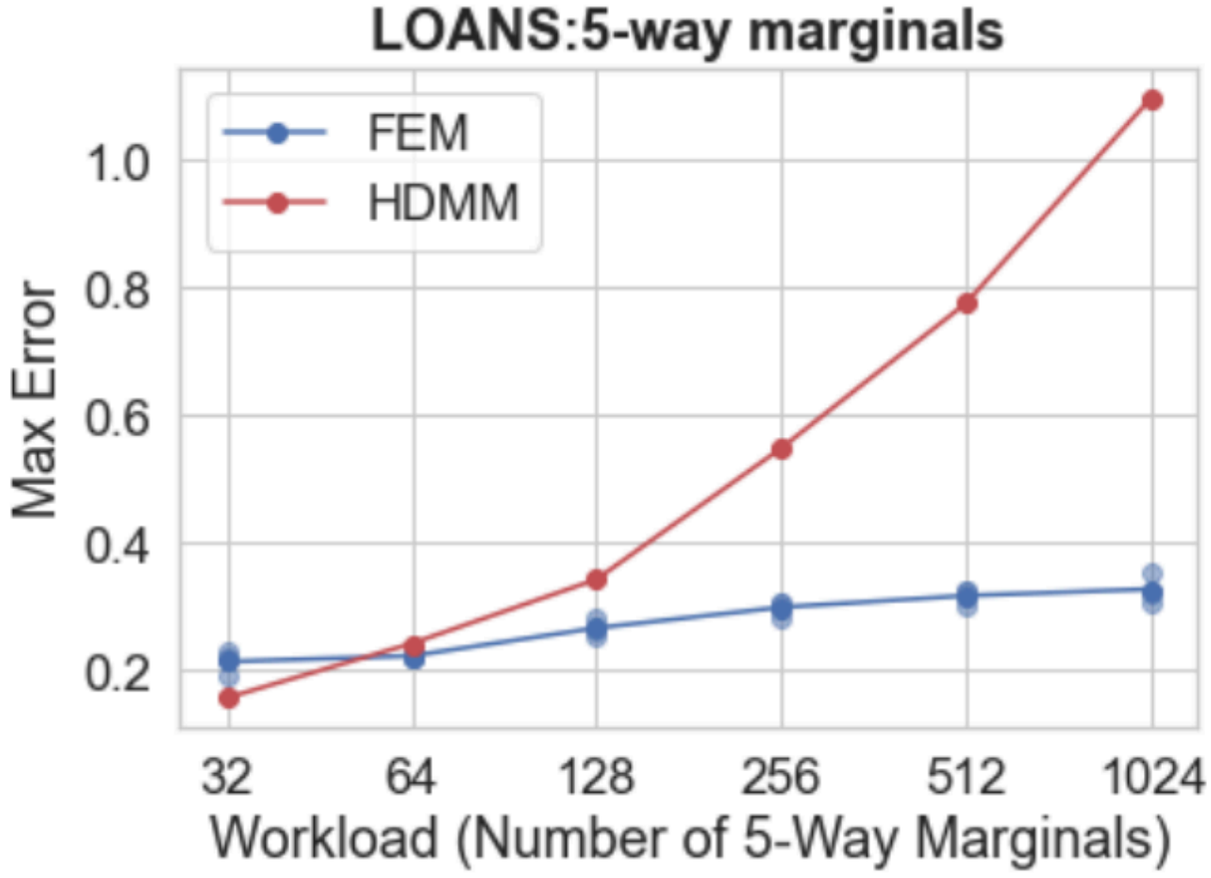
Comparison with HDMM [MMHM18]





Comparison with HDMM [MMHM18]





Leveraging Public Data

[LVSUW21]

Running MW over a public data set

MW^{pub}

Data player
Run MW over a public dataset

vs.

Query player
Best response: find a query with high payoff (exponential mechanism)

MW^{pub}

Data player
Run MW over a public dataset

vs.

Query player
Best response: find a query with high payoff (exponential mechanism)

(Non-Zero) Game Value

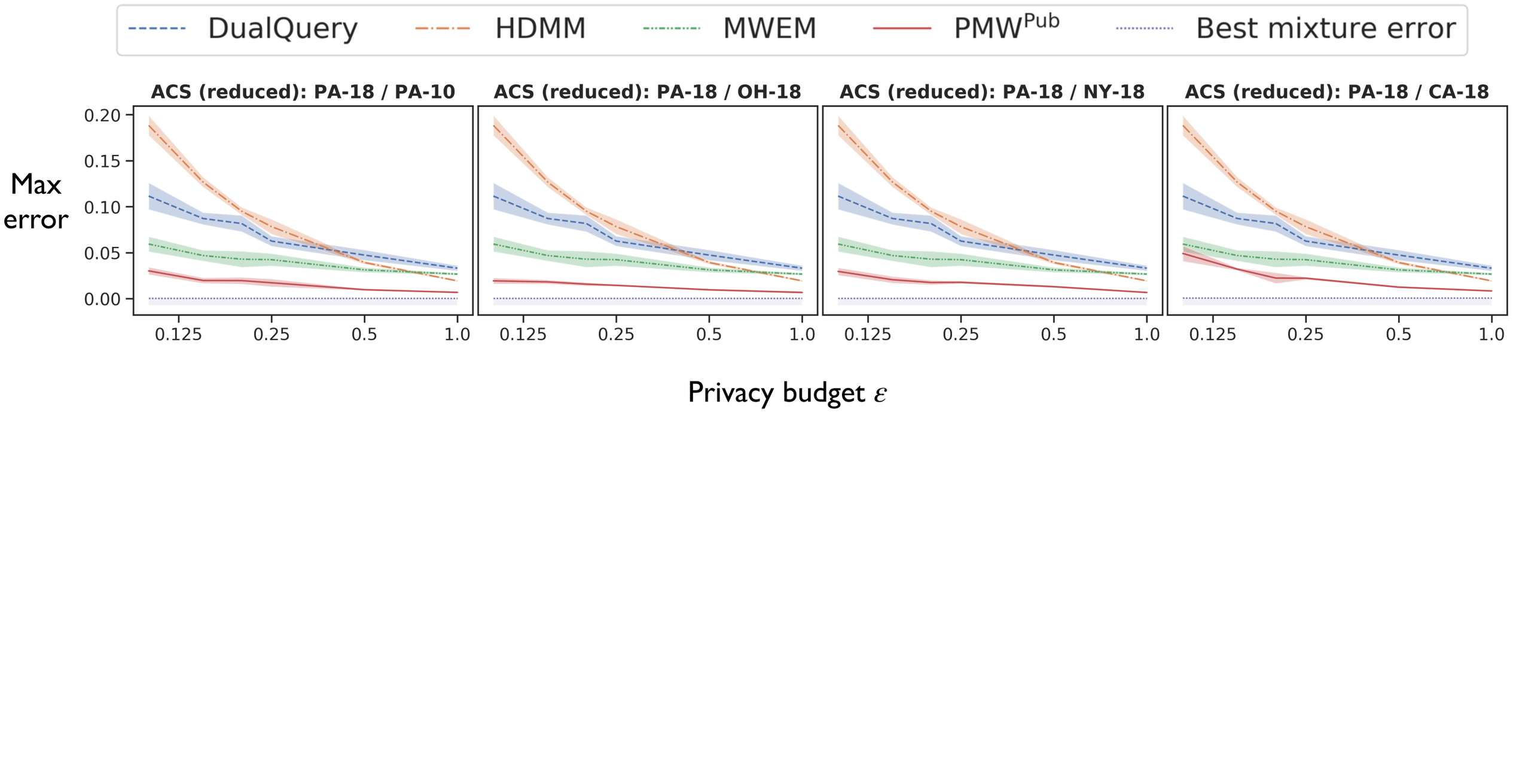
Given a public dataset S

Best Mixture Error: $\min_{\mu \in \Delta(S)} \max_{q \in Q} \left[q(\mu) - q(D) \right]$

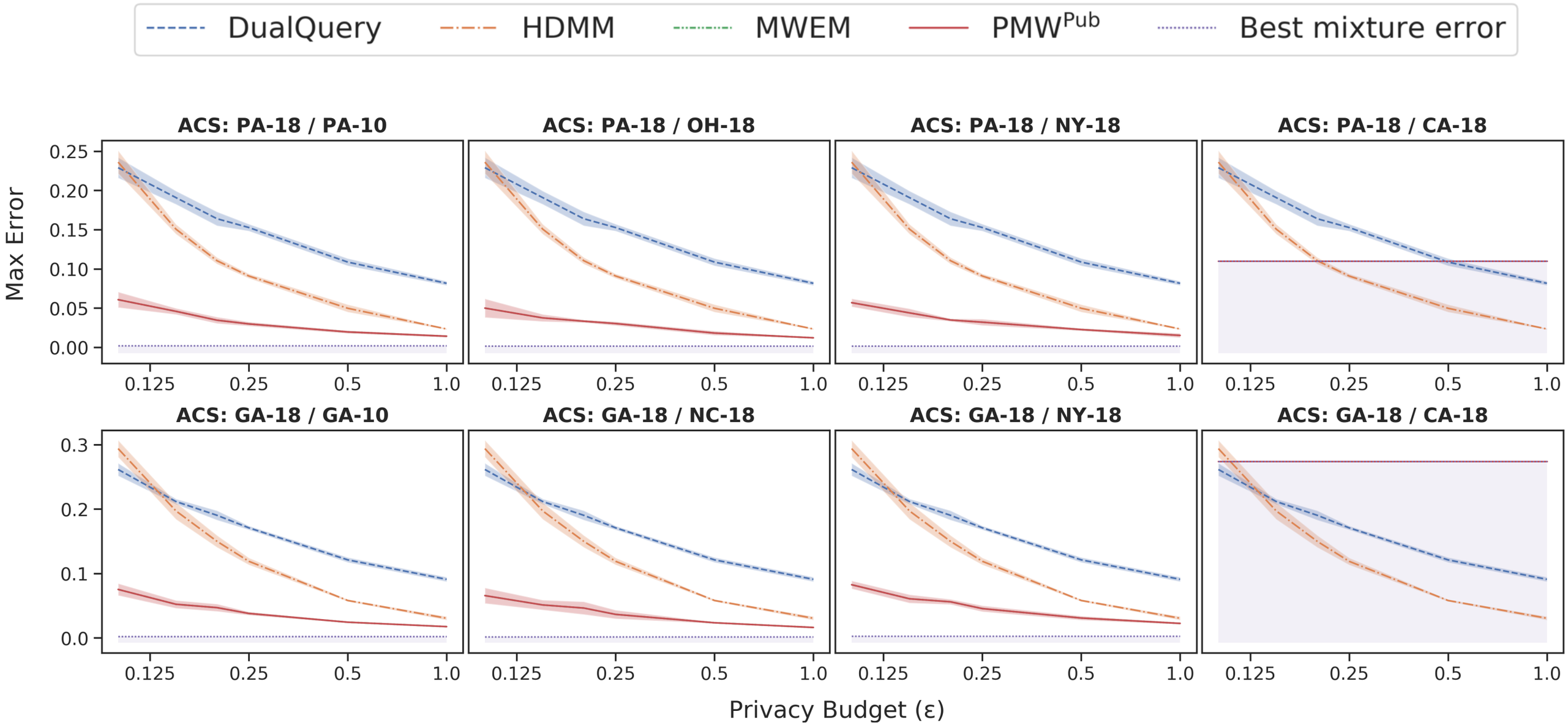
↑

Characterizing public-private relationship (S, D)

Combinations of (Private Data / Public Data)



Combinations of (Private Data / Public Data)



Other Query Classes

- Leveraging public data to answer query class with unbounded Littlestone dimension
  [BCMNUW] ICML20

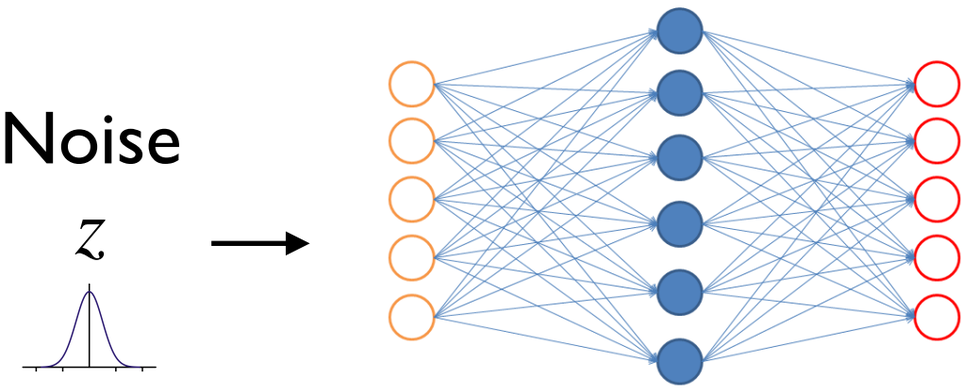
- Oracle-efficient algorithms for generating synthetic data for exponential-sized query class
  [NRW] FOCS19
  - Challenge: Privacy guarantee depends on the optimality of oracle

General-purpose synthetic data with deep generative models

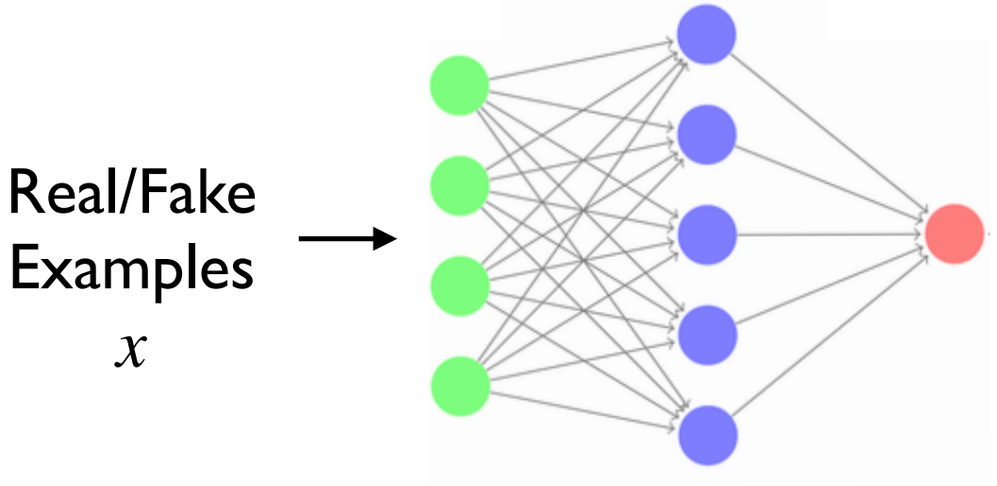
Generative Adversarial Nets (GANs) [GPM+14]

2-Player Zero-Sum Game

Generator G:
mimic the real data



Discriminator D:
distinguish real and fake data



Wasserstein GAN [ACB17]

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

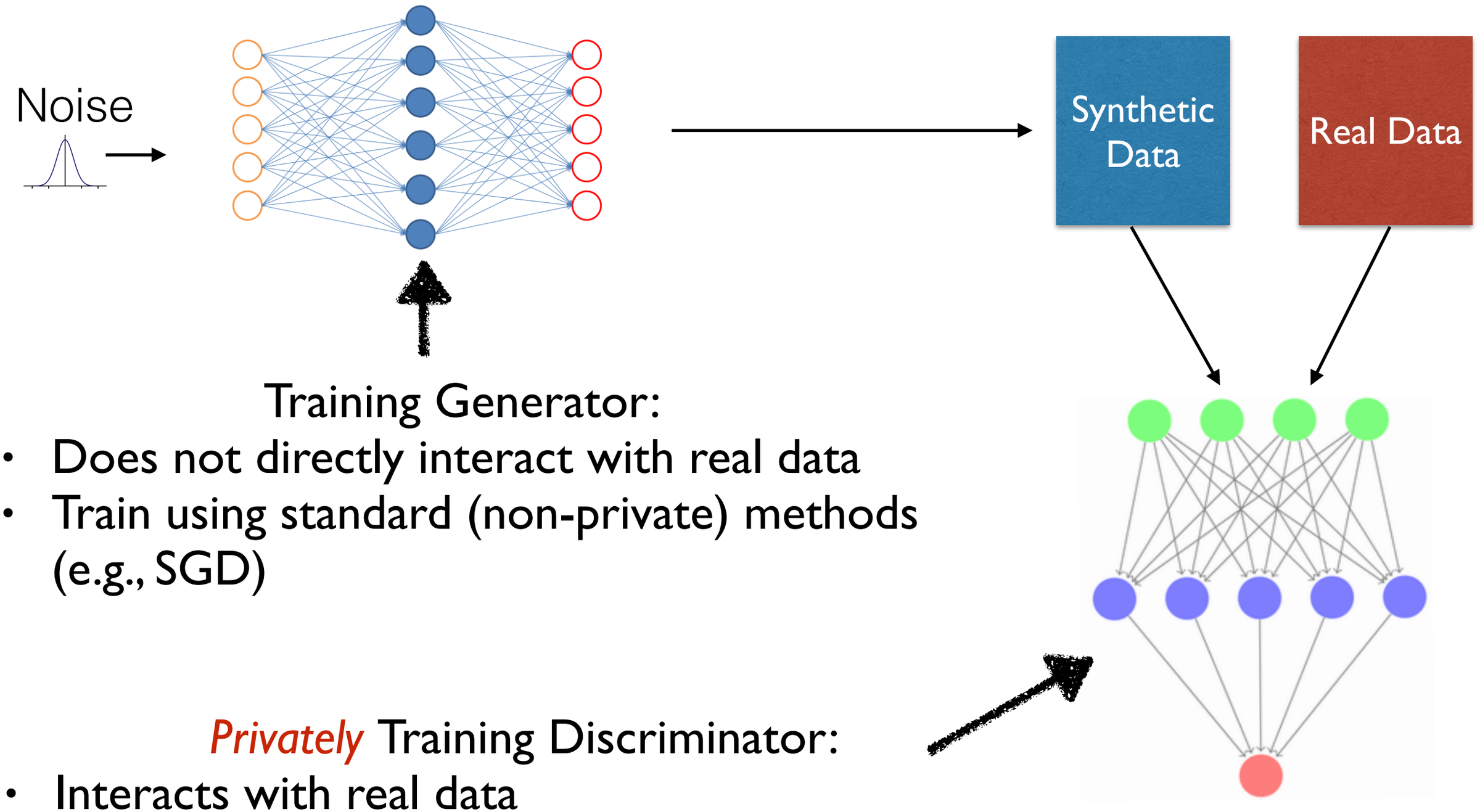
Approach
Generative adversarial nets (GANs)
+ Differential privacy

DP GANs Support Clinical Data Sharing [B*W*WLBBG]

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)

Privately Training Discriminator:

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

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]

Private Post-GAN Boosting

[N*W*D] 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
distribution ϕ over B

Query player
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

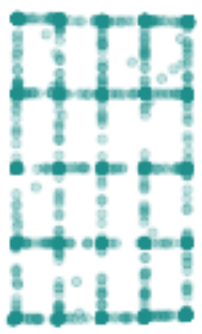
Query player
uses exponential mech to
select a useful discriminator

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

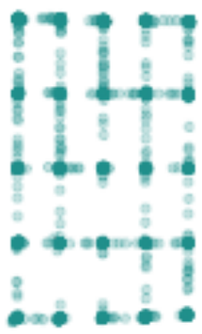
Rejection sampling:
Use D to improve ϕ by "rejecting" unlikely samples

Real Data

Last Generator



DRS



PGB

PGB+DRS

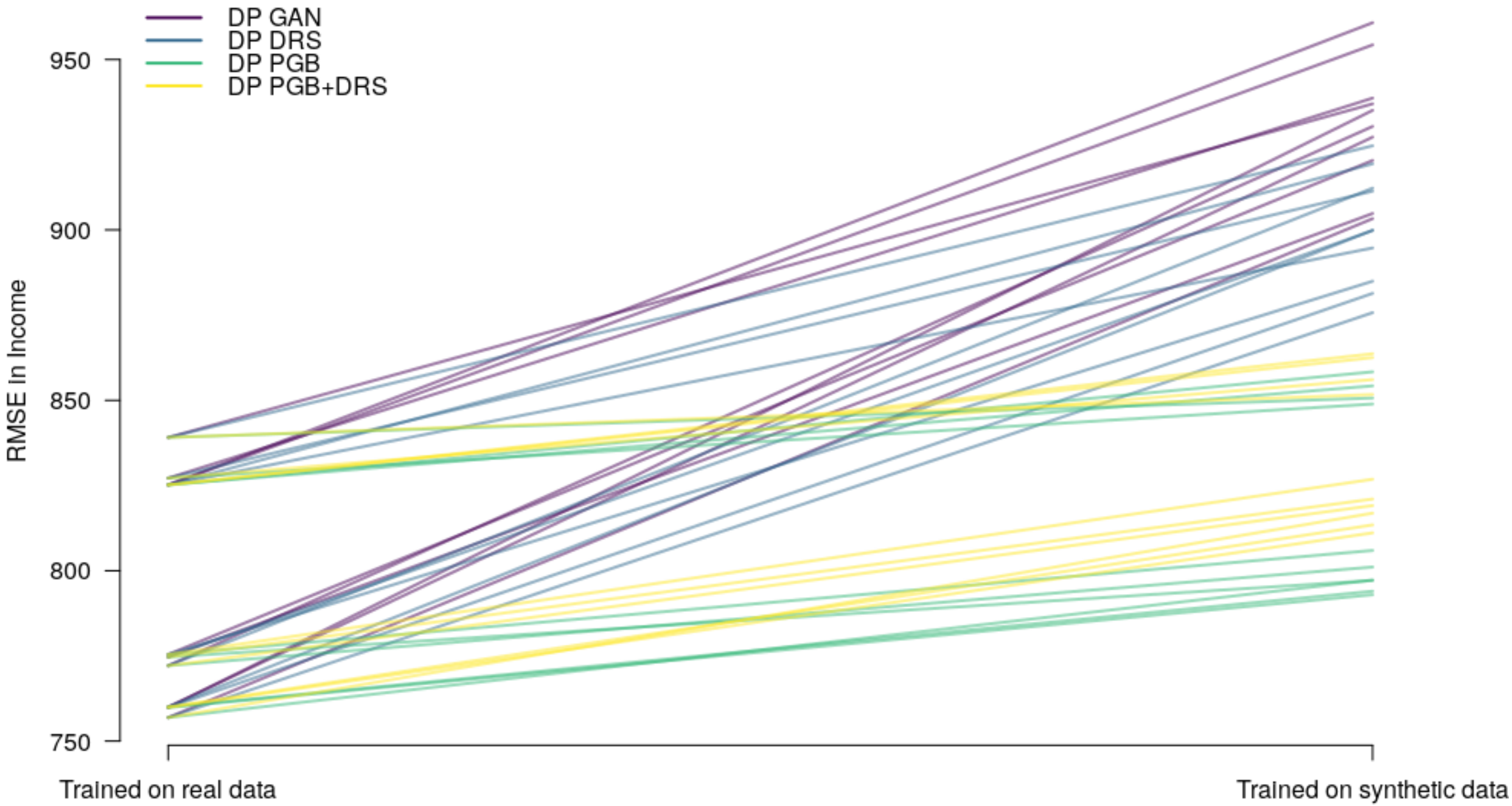
Real Data

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
950
900
850
800
750
RMSE in Income
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

References

"Leveraging public data in private query release"
preprint

"Private Post-GAN Boosting"
ICLR 2021

"New Oracle-Efficient Algorithms for Private Synthetic Data Release"
ICML 2020

"Privacy-preserving generative deep neural networks support clinical data sharing"
In Circulation: Cardiovascular Quality and Outcomes 2019

"How to Use Heuristics for Differential Privacy"
FOCS 2019

"Dual Query: Practical Private Query Release for High Dimensional Data"
ICML 2014; JPC 2016

Start by writing down as a Linear Program

Find a distribution over the domain X: $(p_1, \ldots, p_N)$ where $N = |X|$

For each query $q \in Q$:

$$\sum_i^N p_i\, q(x_i) \leq q(D)$$

$$\sum_i^N p_i\, q(x_i) \geq q(D)$$

For each query $q \in Q$:

$$\sum_i^N p_i\, q(x_i) \leq q(D)$$

$$\sum_i^N p_i(1 - q(x_i)) \leq 1 - q(D)$$

Suppose Q closed under negation

For each query $q \in Q$:

$$\sum_i^N p_i\ (q(x_i) - q(D)) \leq 0$$

Equivalence between LP's and Zero-sum games

Zero-Sum Game Formulation of the LP

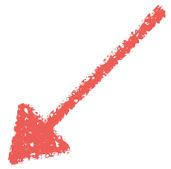
For each query $q \in Q$:

$$\sum_{i}^{N} p_i \ (q(x_i) - q(D)) \leq 0$$

Data (primal) player
actions: records in X

Query (dual) player
actions: queries in Q

distribution over
records $x \in X$



distribution over
queries $q \in Q$

Payoff for action profile (x, q):
$U(x, q) = q(x) - q(D)$

Data player wants to minimize and Query player wants to maximize

Approximate Equilibrium Implies Accuracy

Definition (Approximate Minimax Equilibrium)

- Data player plays a distribution $\hat{D}$ over records
- Query player plays distribution $\hat{Q}$ over queries
- $(\hat{D}, \hat{Q})$ is α-*approximate minimax equilibrium*, if no player can gain more than α by switching to a different distribution.

Theorem: Suppose that $(\hat{D}, \hat{Q})$ is an α-minimax equilibrium, then $\hat{D}$ is α-accurate.

Output $\hat{D}$ as the synthetic data

How do we compute a minimax strategy privately?

Equilibrium via No-Regret Learning

Have two players play against each other over T rounds. In each round t:

- No-Regret player uses a no-regret algorithm to maintain a distribution over actions
- Best-Response player chooses the best response against the opponent's distribution

Theorem([FS96]): Let $\bar{p} = \frac{1}{T}\sum_{t=1}^{T} p^t$ and $\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x^t$. Then $(\bar{p}, \bar{x})$ forms an approximate minimax equilibrium.

No-Regret Online Learning Algorithms

Set of actions A: 

In each of the T rounds, the learner

- Maintains distribution p^t over A
- Observes loss $L^t \in [0, 1]^{|A|}$
- Incurs loss $\langle L^t, p^t \rangle$

No-Regret Property: suffer cumulative loss which is
"not much worse" than the cumulative loss of the best fixed distribution p

Example: Multiplicative Weights (MW)
[LW89, Vov90, KW94, FS97]

Private Query Release: Prior Work

- [DMNS06]: Adding independent noise (Laplace mechanism) requires sample size $n \gtrsim |Q|^{1/2}/(\alpha)$, where $|Q|$: number of queries

  Extremely efficient

Synthetic Database Approach

- [HR10, HLM12]: Private Multiplicative Weights (PMW) requires $n \gtrsim d^{1/2} \log |Q|/(\varepsilon\alpha^2)$

  Needs to maintain an object of exponential size
  Run-time scaling with $|X| = 2^d$

Can we design efficient synthetic data algorithms for high-dimensional data?

- In general, no.
- Impossibility Results [DNRRV09, UV11, Ull13]
- *Worst-case* run-time must be exponential in *d*

Our Goal: provide an algorithm with
provable (worst-case) privacy guarantee and
good empirical run-time performance

Theoretical Guarantees

- Theorem: DualQuery is (ε, δ)-differentially private and α-accurate for

$$\alpha = O\left(\frac{d^{1/6}\log^{1/2}|Q|}{n^{1/3}\varepsilon^{1/3}}\right)$$

- Requires a sample complexity

$$n \gtrsim \frac{d^{1/2}\log^{3/2}|Q|}{\alpha^3\varepsilon}$$

- Sub-optimal accuracy guarantee compared to PMW [HR10]

$$n \gtrsim \frac{d^{1/2}\log|Q|}{\alpha^2\varepsilon}$$