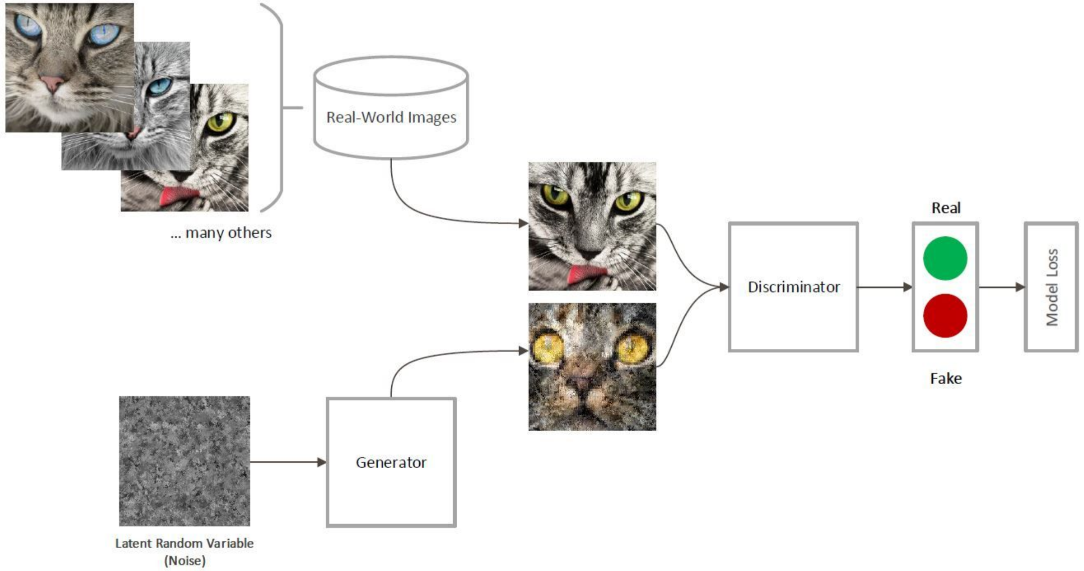
#ml-papers April 2020

Differentially Private Generative Adversarial Network

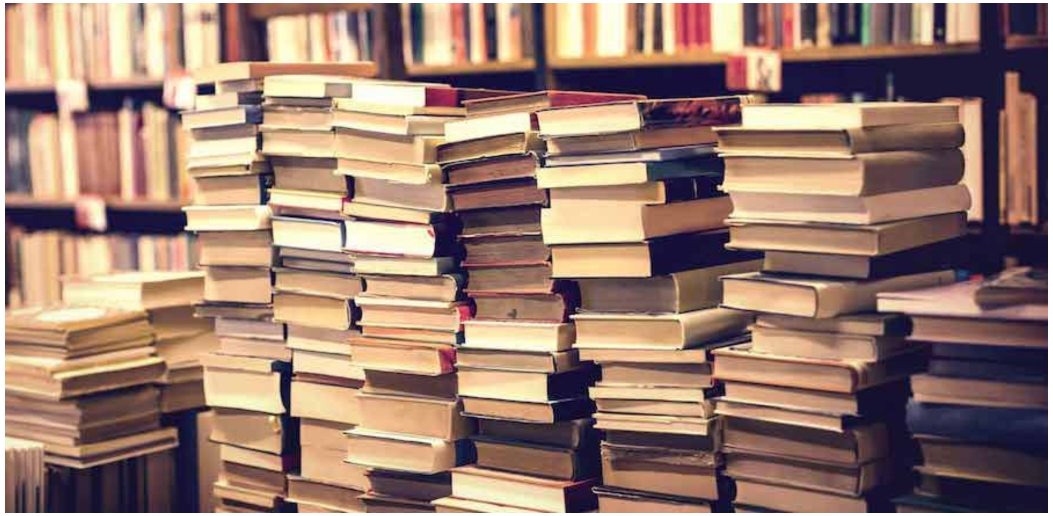
Section 1 - Intro

- **why** use generative models? **why** do we need to preserve privacy?
- GANs are a combination of deep learning and… game theory??
    - GAN refresher
    - quick trivia: how are GANs & autoencoders same/different?

- why vanilla GANs won't preserve privacy (simplest example possible) [ref]
-
- what does DPGAN do differently?
    - noise
    - ~~gradient clipping~~
    - Wasserstein distance instead of JS-divergance (isn't this common?)



Section 2 - Related Work on GANs



"Improved Training of Wasserstein GANs" - introduces a method for training WGANs that doesn't rely on weight clipping & produces more stable GANs

EBGAN & BEGAN both use autoencoders as discriminators, these methods try to stabilize training by addressing the imbalance of the problem

There's a few other papers referenced here but I think this gives a better overview of the current research limitations of GANs. The GAN papers they cite in this section all deal with issues of training stability.

Section 2 - Related Work on DP

"Differential privacy is a system for publicly sharing information about a dataset by describing the patterns of groups within the dataset while withholding information about individuals in the dataset."

Many ways of doing this: this paper adds noise to true values while other methods do the same for gradients. Other papers they cite frame it as an empirical risk minimization.

however their framework "has the same spirits as the objective perturbation, which is different from adding noise directly on the output parameters" (unclear what this means..)

They cite this survey paper, which sounds interesting but I couldn't find a version that wasn't paywalled. My main takeaways here are that research is focused on

1) finding better architectures for perturbation
2) algorithmically proving DP

Section 2 - Related Work on DL&DP

"Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data"
method: "multiple models trained with disjoint datasets, such as records from different subsets of users. Because they rely directly on sensitive data, these models are not published, but instead used as "teachers" for a "student" model. The student learns to predict an output chosen by noisy voting among all of the teachers, and cannot directly access an individual teacher or the underlying data or parameters" -> perturbing the target?
downside: privacy loss proportional to the amount of labeled data
"Privacy-Preserving Deep Learning"
method: ..
downside: ..

Section 3.1. Differential Privacy

Differential Privacy

A randomized algorithm Ap is (ϵ, δ)-differentially private if for any two databases D and D′ differing in a single point and for any subset of outputs S

$$\mathbb{P}(\mathcal{A}_p(\mathcal{D}) \in \mathrm{S}) \leq e^{\epsilon} \cdot \mathbb{P}(\mathcal{A}_p(\mathcal{D}') \in \mathrm{S}) + \delta,$$

or

$$\left|\log\left(\frac{P(\mathcal{A}_p(\mathcal{D}) = s)}{P(\mathcal{A}_p(\mathcal{D}') = s)}\right)\right| \leq \epsilon,$$

where

A: algorithm; D: databases; P: randomness

ϵ: privacy level -- the lower the ϵ, the less different between two outputs, which means higher privacy

δ: violation of pure privacy - overlap between two outputs no matter what ϵ is

We are interested in smaller δ so that δ < 1/ size(D)

Example

in clinical experiments, a proper membership protection would ensure that

1. replacing this person with another one will not affect the result too much (ϵ)
2. privacy is protected (δ)

Section 3.2. GAN & WGAN

GAN:
two models generative model **G** and discriminative model **D** play a minmax game:

$$\min_G \max_D V(G, D) = E_{\mathbf{x}\sim p_{data}(\mathbf{x})}[log(D(\mathbf{x}))] + E_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[log(1 - D(G(\mathbf{z})))].$$

model G: transforms input distribution to output distribution that approximates the data distribution
model D: estimates the probability that a sample came from the training data rather than the output of G.

WGAN:
improves GAN with the Wasserstein distance (how much sand to move from one pile to generate another pile)

$$\min_G \max_{w \in W} E_{\mathbf{x}\sim p_{data}(\mathbf{x})}[f_w(\mathbf{x})] - E_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[f_w(G(\mathbf{z}))],$$

Section 3.3. DPGAN

DPGAN:

preserving the privacy during the training procedure (instead of adding noise on the final parameters)

Algorithm 1 Differentially Private Generative Adversarial Nets

Require: α_d, learning rate of discriminator. α_g, learning rate of generator. c_p, parameter clip constant. m, batch size. M, total number of training data points in each discriminator iteration. n_d, number of discriminator iterations per generator iteration. n_g, generator iteration. σ_n, noise scale. c_g, bound on the gradient of Wasserstein distance with respect to weights

Ensure: Differentially private generator θ.

Initialize discriminator parameters w_0, generator parameters θ_0.
for $t_1 = 1, \ldots, n_g$ **do**
    for $t_2 = 1, \ldots, n_d$ **do**
        Sample $\{\mathbf{z}^{(i)}\}_{i=1}^{m} \sim p(\mathbf{z})$ a batch of prior samples.
        Sample $\{\mathbf{x}^{(i)}\}_{i=1}^{m} \sim p_{data}(\mathbf{x})$ a batch of real data points.
        For each i, $g_w(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}) \leftarrow \nabla_w \left[f_w(\mathbf{x}^{(i)}) - f_w(g_\theta(\mathbf{z}^{(i)})) \right]$
        $\bar{g}_w \leftarrow \frac{1}{m}(\sum_{i=1}^{m} g_w(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}) + N(0, \sigma_n^2 c_g^2 I))$.   <- add noice
        $w^{(t_2+1)} \leftarrow w^{(t_2)} + \alpha_d \cdot RMSProp(w^{(t_2)}, \bar{g}_w)$
        $w^{(t_2+1)} \leftarrow clip(w^{(t_2+1)}, -c_p, c_p)$ <- guarantee k-lipscitz
    end for
    Sample $\{\mathbf{z}^{(i)}\}_{i=1}^{m} \sim p(\mathbf{z})$, another batch of prior samples.
    $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} f_w(g_\theta(\mathbf{z}^{(i)}))$
    $\theta^{(t_1+1)} \leftarrow \theta^{(t_1)} - \alpha_g \cdot RMSProp(\theta^{(t_1)}, g_\theta)$
end for
return θ.

differentially private discriminator

computation of generator

differentially private generator

Section 3.4 - Privacy Guarantees

- Privacy loss (3.2) describes the difference between two distributions, D and D' by changing data
- **Assumption**
  - The supports of the 2 distributions associated with *M(aux,D)* and *M(aux,D')* (*aux* = auxiliary point) are *generally* the same, so it is appropriate to evaluate at any arbitrary point *o*.
- Show that only clipping the gradient updates guarantees privacy
- Sigma below is the noise to impose on the gradient (in Normal dist)

Definition 3.2. (Privacy loss)

$$c(o; M, aux, \mathcal{D}, \mathcal{D}') \triangleq \log \frac{\mathbb{P}[M(aux, D) = o]}{\mathbb{P}[M(aux, \mathcal{D}') = o]},$$

Definition 3.3. (Log moment generating function)

$$\alpha_M(\lambda; aux, \mathcal{D}, \mathcal{D}') \triangleq \log \mathbb{E}_{o \sim M(aux, D)}[exp(\lambda C(M, aux, \mathcal{D}, \mathcal{D}'))].$$

Definition 3.4. (Moments accountant)

$$\alpha_M(\lambda) \triangleq \max_{aux, \mathcal{D}, \mathcal{D}'} \alpha_M(\lambda; aux, \mathcal{D}, \mathcal{D}').$$

LEMMA 1. *Given the sampling probability $q = \frac{m}{M}$, the number of discriminator iterations in each inner loop n_d and privacy violation δ, for any positive ϵ, the parameters of discriminator guarantee (ϵ, δ)-differential privacy with respect to all the data points used in that outer loop (fix t_1) if we choose:*

$$\sigma_n = 2q\sqrt{n_d \log(\frac{1}{\delta})} \Big/ \epsilon. \qquad (10)$$

Section 4: Experiments

- Relationship between Privacy Level and Generation Performance
- Relationship between Privacy Level and Convergence of Network
- MNIST Classification
- EHR Data Generation
- EHR Classification

Section 4.1: Relationship between Privacy Level and Generation Performance

What they do?

- Four different generated images are produced corresponding to different ϵ.
- ϵ represents different noise levels. Lower the ϵ, higher the noise
- Compare the generated the generated images with nearest neighbors in the training set

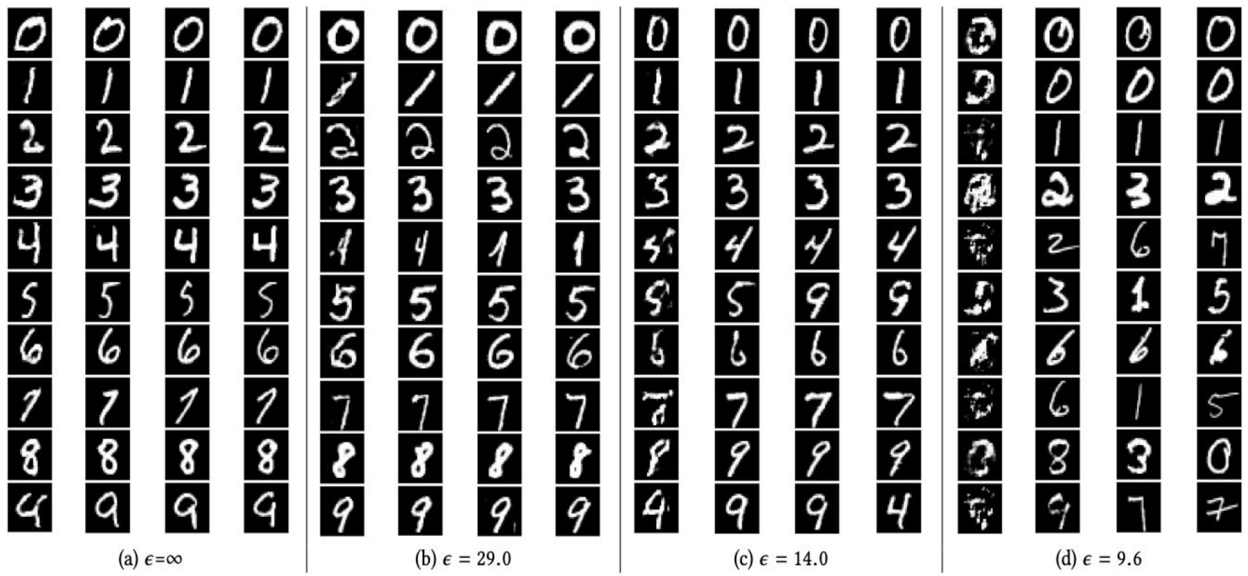
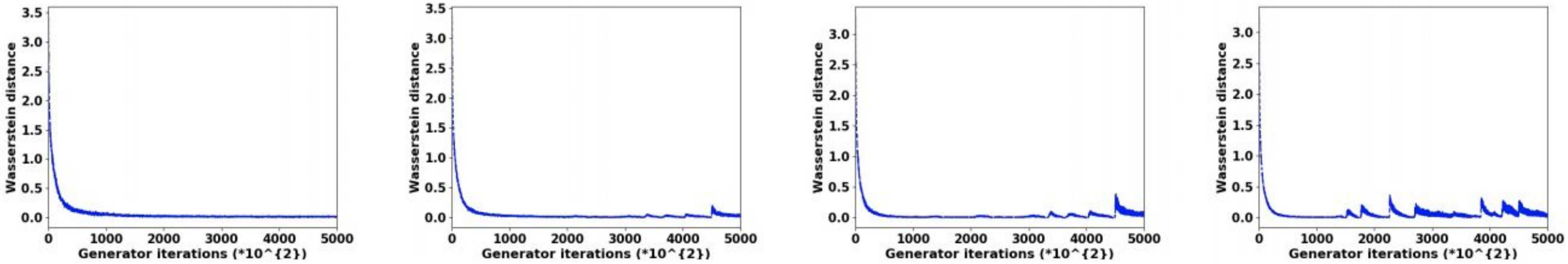


Figure 1: Generated images with four different ϵ on MNIST dataset are plotted in leftmost column in each group. Three nearest neighbors of generated images are plotted to illustrate the generated data is not memorizing the real data and the privacy is preserved. We can see that the images get more blurred as more noise is added.

Conclusion:

- Model doesn't simply memorie training images
- Differential privacy is preserved

Section 4.2: Relationship between Privacy Level and Convergence of Network



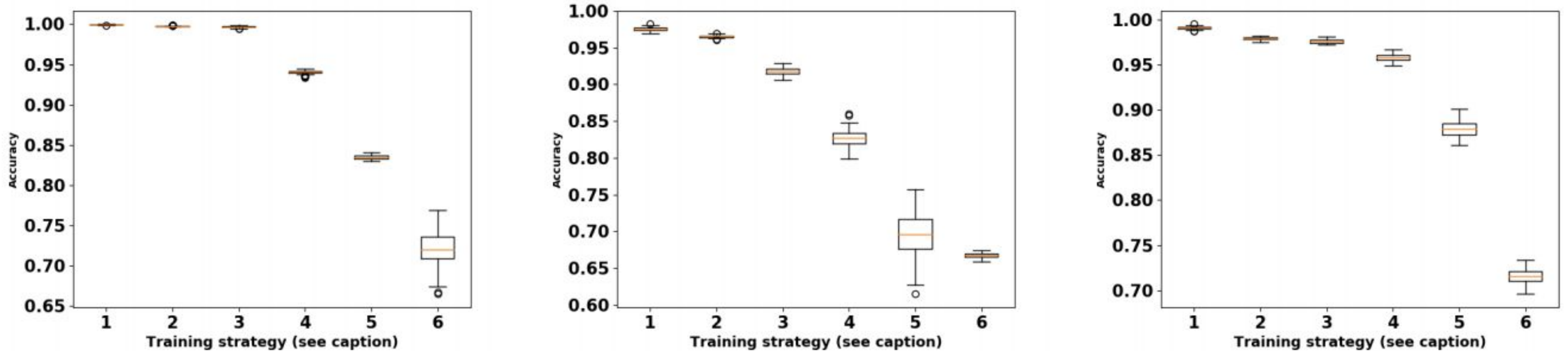
(a) ϵ=∞ (b) ϵ = 29.0 (c) ϵ = 14.0 (d) ϵ = 9.6

Figure 2: Wasserstein distance for different privacy levels when applying DPGAN on MINST. We can see that the curves converge and exhibit more fluctuations as more noise is added.

What they do?

- For different ϵ, plot Wasserstein distance per 100 generator iterations.
- Typically at all ϵ, model converges.
- At higher noise level, you see spiky behavior post convergence
  - Authors attribute this to weight clipping procedure
  - They argue this is eliminated in the training procedure (don't elaborate how?)

Section 4.3: MNIST Classification

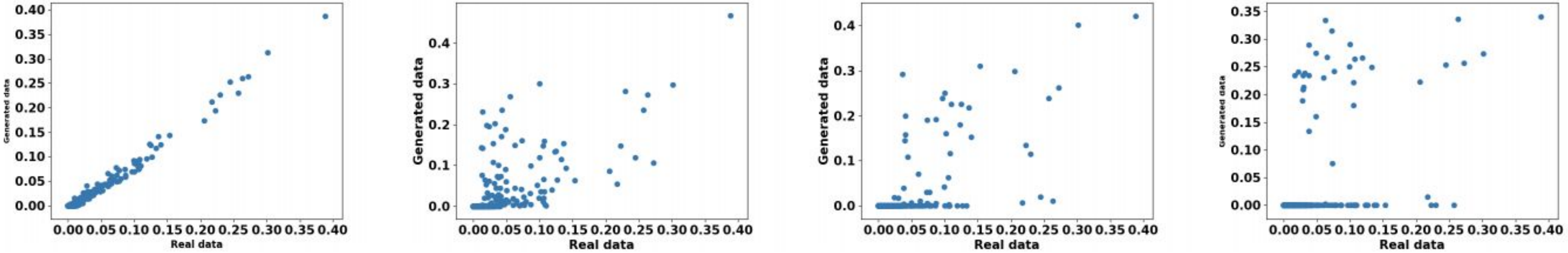


(a) Digit 0 and 1 (b) Digit 2 and 3 (c) Digit 4 and 5

Figure 3: Binary classification task on MNIST database with different training strategies. From left to right we use training data, generated data without noise, generated data with $\epsilon = 11.5, 3.2, 0.96, 0.72$. We can see that as less noise is added, the accuracy of classifier build on generated data gets higher, which indicates that the generated data has better quality.

- As ϵ increases, AUC drops.
- Authors argue the optimal ϵ is between 3 and 11 for the 0,1 pair.
- TLDR: As ϵ (noise) increases, generate quality decreases and affects classification accuracy.

Section 4.4: EHR Data

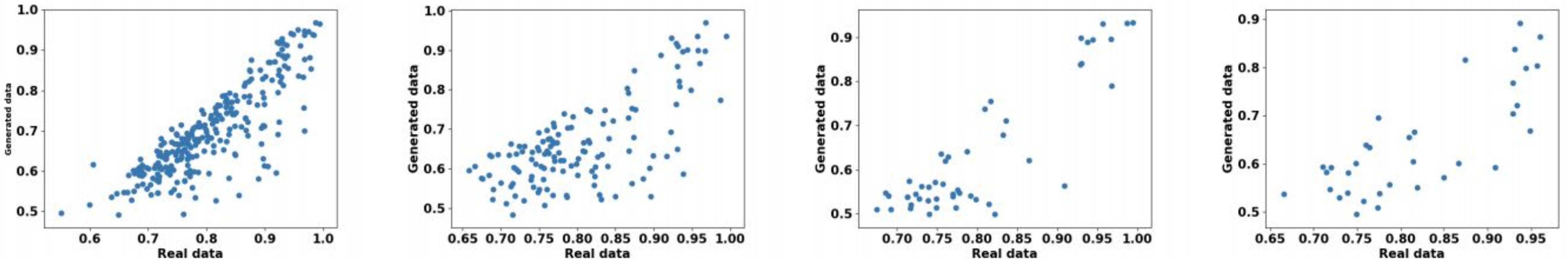


(a) $\epsilon=\infty$ (b) $\epsilon=2.31\times10^2$ (c) $\epsilon=1.39\times10^2$ (d) $\epsilon=96.5$

Figure 4: DWP evaluation on MIMIC-III database with different ϵ values (1070 points). We can see that as more noise is added, the distribution of generated data in each dimension becomes more deviated from the real training data.

- Network structure from [8]
- Use Dimensional Wide Probability (DWP) to measure quality of generated data.
- Example: For rare diseases, adding one person can affect the distribution of the disease.
  - An observer, such as an insurance company can exploit this data to charge higher premiums
  - Adding noise in the generation process hides this from the observer and protects privacy of participants

Section 4.5: Classification on EHR Data



(a) ϵ=∞ (b) ϵ=2.31×10^2 (c) ϵ=1.39×10^2 (d) ϵ=96.5

Figure 5: Dimension-wise prediction evaluation on MIMIC-III database with different ϵ values. We can see that as more noise is added, AUC value of classifier build from generated data gets lower and the data gets sparser.

- Train Logistic Regression classifiers on real and generated data and predict on test data.
- Measure performance using DWP
- The model from the real data tends to perform better on test data
- Data gets sparser as more noise is added

Section 5