Beyond Simple Images: Human Knowledge-Guided GANs for Clinical Data Generation

Devendra Singh Dhami[1,3] , **Mayukh Das**[2,3] , **Sriraam Natarajan**[3]

[1]TU Darmstadt, Germany
[2]Microsoft Research, India
[3]University of Texas at Dallas, USA
devendra.dhami@cs.tu-darmstadt.de, mayukhdas@microsoft.com, sriraam.natarajan@utdallas.edu

Abstract

While Generative Adversarial Networks (GANs) have accelerated the use of generative modelling within the machine learning community, most of the adaptations of GANs are restricted to images. The use of GANs to generate clinical data has been rare due to the inability of GANs to faithfully capture the intrinsic relationships between features given a small amount of observational data. We hypothesize and verify that this challenge can be mitigated by incorporating rich domain knowledge in the form of expert advice in the generative process. Specifically, we propose *human-allied GANs* that uses correlation advice from humans to create synthetic clinical data. We construct a system that takes a symbolic representation of the expert advice and converts it into constraints on correlation of the features during the generative process. Our empirical evaluation demonstrates (a) the superiority of our approach over other GAN models, (b) the importance of incorporating advice over instance noise and (c) an initial framework for incorporation of privacy in our model while capturing the relationships between features.

1 Introduction

Deep learning models have reshaped the machine learning landscape over the past decade (LeCun, Bengio, and Hinton 2015; Goodfellow, Bengio, and Courville 2016). Specifically, Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) have found tremendous success in generating examples for images (Mao et al. 2017; Liu and Tuzel 2016; Radford, Metz, and Chintala 2016), photographs of human faces (Karras, Laine, and Aila 2019; Antipov, Baccouche, and Dugelay 2017; Wang et al. 2018b), image to image translation (Zhu et al. 2017; Li et al. 2018; Liu, Breuel, and Kautz 2017) and 3D object generation (Wang et al. 2018a; Paganini, de Oliveira, and Nachman 2018; Wu et al. 2016) to name a few. Despite their successes, their adaptation to wider range of tasks is limited due to, (a) data hungry nature of such methods, (b) potential privacy violation in generated data (Huang et al. 2018; Huang et al. 2017) and (c) finally, their restricted usage, mainly in the context of images. Consequently, their adaptations in clinical domains are restricted to images. However, recently, synthetic data generation has become a centerpiece of AI research in medicine due to the difficulties in collection, persistence, sharing and analysis of real clinical data.

We aim to address the above limitations. Inspired by Mitchell's argument of "*The Need for Biases in Learning Generalizations*" (Mitchell 1980), we mitigate the challenges of existing data hungry methods via inductive bias while learning GANs. **Our primary contribution is demonstrating that effective inductive bias can be provided by humans in the form of domain knowledge** (Towell and Shavlik 1994; Kunapuli et al. 2013; Odom et al. 2015; Fung, Mangasarian, and Shavlik 2003). Rich human knowledge can effectively balance the impact of quality (sparsity) of training data. Note that data quality also contributes to, the well studied, modal instability of GANs. This problem is especially critical in domains such as medical/clinical analytics that does not typically exhibit 'spatial homophily' (Habeeb et al. 2017), unlike images, and are prone to distributional diversity among feature clusters as well. Our human-guided framework proposes a robust strategy to address this challenge. Note that a key assumption is that **in our setting the human is an ally and not an adversary**.

The second limitation of privacy is crucial for medical data generation. Access to existing medical databases (Dinov 2016) is difficult if not impossible due to the cost and privacy concerns. Hence, synthetic data generation holds tremendous promise for advancing the adaptation of AI techniques inside medicine (Guibas, Virdi, and Li 2017; Mahmood, Chen, and Durr 2018; Frid-Adar et al. 2018; Buczak, Babin, and Moniz 2010). While previous methods generated synthetic images, **we go beyond images and generate clinical data**, specifically lab test values. Building on this, we present a synthetic data generation framework that effectively exploits domain expertise to handle data quality.

We make a few key contributions: (1) We demonstrate how effective human knowledge (advice) can be provided to a GAN as an inductive bias. Specifically, we develop a **symbolic knowledge representation** that makes it easy for the domain expert to provide the relevant advice. (2) We present a method for generating data given this advice. (3) We outline a privacy preserving method for data generation. (4) Finally, we demonstrate the effectiveness and efficacy of our approach on 4 de-identified clinical data sets and 2 private clinical data sets. Our method is generalizable to multiple modalities of data and is not necessarily restricted to images. An important feature of this approach is that **training occurs from very few data samples** (< 50 in one domain) thus

providing human guidance as a data generation alternative.

2   Background

The key principle behind GANs (Goodfellow et al. 2014) is a zero-sum game (Kuhn and Tucker 1953) from game theory, a mathematical representation where each participant's gain or loss is exactly balanced by the losses or gains of the other participants and is generally solved by a minimax algorithm. The generator distribution $p_{data}(\boldsymbol{x})$ over the given data $\boldsymbol{x}$ is learned by sampling $\boldsymbol{z}$ from a random distribution $p_{\boldsymbol{z}}(\boldsymbol{z})$. Initially uniform distribution was proposed but Gaussians have been proven to be superior (Arjovsky and Bottou 2017). While GANs are arguably a powerful framework for estimating generative distributions, convergence dynamics of naive mini-max algorithm has been shown to be unstable. Some recent approaches, among many others, augment learning either via statistical relationships between true and learned generative distributions such as Wasserstein-1 distance (Arjovsky, Chintala, and Bottou 2017), MMD (Li, Swersky, and Zemel 2015) or via spectral normalization of the parameter space of the generator (Miyato et al. 2018) which controls the generator distribution from drifting too far. While successful, these approaches still require significant volume of data and are typically focused on images. While some GAN architectures do exists that are specifically for tabular data (Choi et al. 2017; Xu et al. 2019; Baowaly et al. 2019) generating clinical data still remains an open problem.

Guidance via human knowledge is a provably effective way to control learning in presence of systematic noise (which leads to instability). One typical strategy to incorporate such guidance is by providing *rules that cover examples and/or features*. Some of the earliest approaches are explanation-based learning (EBL-NN, (Shavlik and Towell 1989)) or ANNs augmented with symbolic rules (KBANN, (Towell and Shavlik 1994)). Various widely-studied techniques of leveraging domain knowledge for optimal model generalization include polyhedral constraints in case of knowledge-based SVMs, (Cortes and Vapnik 1995; Schölkopf et al. 1998; Le, Smola, and Gärtner 2006; Fung, Mangasarian, and Shavlik 2003)), preferences rules (Braziunas and Boutilier 2006; Kunapuli et al. 2013; Odom et al. 2015) or qualitative constraints (ex: monotonicities / synergies (Yang and Natarajan 2013) or quantitative relationships (Ganchev et al. 2010)). While these models exhibit considerable improvement with the incorporation of human knowledge, there is only limited use of such knowledge in training GANs. Our approach resembles the qualitative constraints framework in spirit.

While widely successful in building optimally generalized models in presence of systematic noise (or sample biases), knowledge-based approaches have mostly been explored *mainly* in the context of discriminative modeling. In the generative setting, a recent work extends the principle of posterior regularization from Bayesian modeling to deep generative models in order to incorporate structured domain knowledge (Hu et al. 2018). Traditionally, knowledge based generative learning has been studied as a part of learning probabilistic graphical models with structure/parameter priors (Mansinghka et al. 2006). We aim to extend the use of
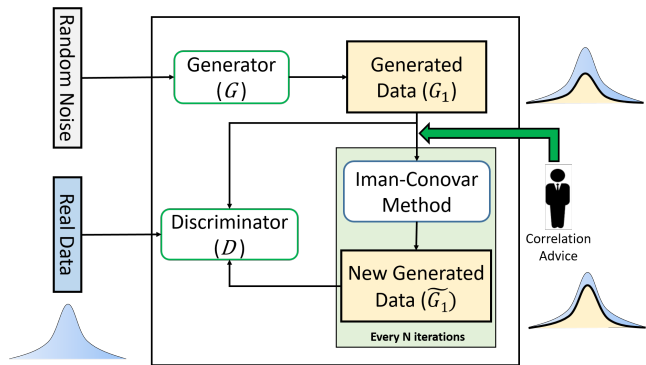

Figure 1: Human-Allied GAN. Correlation advice takes generated distribution closer to the real distribution.

domain knowledge to the generative model setting by directly using them in the construction of GANs.

3   Human-Allied GANs

A notable disadvantage of adversarial training formulation is that the training is slow and unstable, leading to mode collapse (Arjovsky and Bottou 2017) where the generator starts generating data of only a single modality. This has resulted in GANs not being exploited to their full potential in *generating synthetic non-image clinical data*. Human advice can encourage exploration in diverse areas of the feature space and helps learn more stable models (Odom and Natarajan 2018). Hence, we propose a human-allied GAN architecture (HA-GAN) (figure 1). The architecture incorporates human advice in form of feature correlations. Such intrinsic relationships between the features are *both important and readily available in clinical data* and thus become a natural candidate as additional knowledge/advice in guided model learning for faithful data generation.

Our approach builds upon a GAN architecture (Goodfellow et al. 2014) where a random noise vector is provided to the generator which aims to generate examples as close to the real distribution as possible. The discriminator's goal is to distinguish between real examples and ones generated by the generator. The generator aims to maximize the probability that the discriminator makes a mistake and the discriminator tries to minimize its mistakes thereby resulting in a min-max optimization problem which can be solved by a mini-max algorithm. We adopt the Wasserstein GAN (WGAN) architecture[1] (Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017) that focuses on defining a distance/divergence (Wasserstein or earth movers distance) to measure the closeness between the real distribution and the model distribution.

3.1   Human Knowledge as Inductive Bias

Historically, two approaches have been studied for using guidance as bias. The first is to provide **advice on the labels as constraints or preferences** that controls the search space. Some example advice rules on the labels include: *(3*

---

[1] For brevity, we use 'GAN' to indicate 'W-GAN'

$\leq feature_1 \leq 5) \Rightarrow label = 1$ and $(0.6 \leq feature_2 \leq 0.8) \wedge (4 \leq feature_3 \leq 5) \Rightarrow label = 0$. While such advice is relevant in a *discriminative setting, they are not ideal for GANs.* Since GANs are shown to be sensitive to the training data and here the *labels are getting generated*, they should not be altered during training.

The second approach, which we take, is to specify **correlations between features as preferences**. This allows for faithful representation of the hidden relationships between features.

Example: *Consider predicting heart attack with 3 features - cholesterol, blood pressure (BP) and income. The values of the given features can vary (sometimes widely) between different patients due to several latent factors (ex, smoking, education etc.). It is difficult to assume any specific distribution. In other words, it is difficult to infer the distribution from which the values for the features are drawn (though the feature values in the data might appear similar).*

We modify the correlation coefficients (for both +ve and -ve correlations) between the features by increasing them if the human advice (knowledge) suggests that two features are highly correlated and decrease them if advice suggests otherwise.

Advice representation: Inspired from first order conditional influence (FOCI) statements (Natarajan et al. 2008) that specify probabilistic influences among different features, we allow advice in the form of simple qualitative influences. A qualitative influence (QI) is of the form $f_1 \, Q_{inf} \, f_2$, where f_1 and f_2 are features in the data, and shows a direct dependence of f_2 on the influencing feature f_1. Our model can handle 3 types of QIs:

1. $f_1 M^+ f_2$ captures the positive monotonic relationship between the features.

2. $f_1 M^- f_2$ captures the negative monotonic relationship between the features.

3. $f_1 Q^+ f_2$ and $f_1 Q^- f_2$ capture the more general case of $+ve$ and $-ve$ linear correlations between features respectively and is the focus of this work.

Note that the first 2 type of QIs can be captured by rank correlation and the 3rd type of QI can be captured by the Pearson correlation coefficient. Thus, all the three types of advice can be naturally incorporated in our model. It must be mentioned that FOCI statements employed predicate logic notations for learning statistical relational models. For the purposes of this work, we restrict ourselves to propositional representation of the knowledge but note that this can serve as a foundation for learning relational versions of GANs.

Advice injection: After every fixed number of iterations, N, we calculate the correlation matrix of the generated data $\mathcal{G}_1$ and provide a set of advice ψ on the correlations between different features. Consider the following motivating example for the use of correlations as a form of advice.

Example: *Continuing the above example, since rise in the cholesterol level can lead to rise in BP and vice versa, expert advice here can suggest that cholesterol and BP are highly correlated. Also, as income may not contribute directly to BP and cholesterol levels, another advice here can be to decorrelate cholesterol/BP and income level.*

The example advice rules are: 1. Correlation("cholesterol level","BP")↑, 2. Correlation("cholesterol level","income level")↓ and 3. Correlation("BP","income level")↓, where ↑ and ↓ indicate increase and decrease respectively.

These advice in form of QIs can be written as:

1. `cholesterol level` Q^+ `BP`

2. `cholesterol level` Q^- `income level`, and

3. `BP` Q^- `income level`

Based on the first advice, the correlation coefficient between *cholesterol level* and *BP* needs to be increased. Then

$$\mathcal{C} = \begin{bmatrix} 1 & 0.2 & 0.3 \\ 0.2 & 1 & 0.07 \\ 0.3 & 0.07 & 1 \end{bmatrix} \mathcal{A} = \begin{bmatrix} 1 & \lambda & 1 \\ \lambda & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (1)$$

Here $\mathcal{C}$ is the correlation matrix, $\mathcal{A}$ is the advice matrix and λ is the factor by which the correlation value is to be augmented. In the cases where there is a need to increase the value of the correlation coefficient, λ should be > 1. We set $\lambda = \frac{1}{max(|\mathcal{C}|)}$. Since $-1.0 \leq \forall c \in \mathcal{C} \leq 1.0$, in this case, the value of $\lambda \geq 1.0$, leading to enhanced correlation via Hadamard product. Thus the new correlation matrix $\hat{\mathcal{C}}$ is,

$$\hat{\mathcal{C}} = \mathcal{C} \odot \mathcal{A} = \begin{bmatrix} 1 & 0.2 & 0.3 \\ 0.2 & 1 & 0.07 \\ 0.3 & 0.07 & 1 \end{bmatrix} \odot \begin{bmatrix} 1 & \frac{1}{0.3} & 1 \\ \frac{1}{0.3} & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0.667 & 0.3 \\ 0.667 & 1 & 0.07 \\ 0.3 & 0.07 & 1 \end{bmatrix} \quad (2)$$

If the advice says that features have low correlations (2nd rule in example), we decrease the correlation coefficient. Now, λ must be < 1 and we set $\lambda = max(|\mathcal{C}|)$. Since $-1 \leq \forall c \in \mathcal{C} \leq 1.0$, the value of $\lambda \leq 1.0$. Thus multiplying by λ will decrease the correlation value, and the new correlation matrix is,

$$\hat{\mathcal{C}}_1 = \hat{\mathcal{C}} \odot \mathcal{A} = \begin{bmatrix} 1 & 0.667 & 0.3 \\ 0.667 & 1 & 0.07 \\ 0.3 & 0.07 & 1 \end{bmatrix} \odot \begin{bmatrix} 1 & 1 & 0.3 \\ 1 & 1 & 0.3 \\ 0.3 & 0.3 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0.667 & 0.09 \\ 0.667 & 1 & 0.021 \\ 0.09 & 0.021 & 1 \end{bmatrix}$$

$$(3)$$

This is used to create the new generated data $\tilde{\mathcal{G}}_1$. For negative correlations, the process is unchanged.

To capture bad advice, we set the desired correlation between highly correlated features to 0 and the correlation between non correlated features or low correlated features is set to either 1 or -1. Given the initial correlation matrix $\mathcal{C}$,

$$\mathcal{C} = \begin{bmatrix} 1 & 0.2 & 0.3 \\ 0.2 & 1 & 0.07 \\ 0.3 & 0.07 & 1 \end{bmatrix} \quad (4)$$

suppose the advice says that we need to increase the correlation coefficient between feature 1 and feature 2. Then the new correlation matrix after bad advice can be calculated as:

$$\mathcal{C} = \begin{bmatrix} 1 & 0.2 & 0.3 \\ 0.2 & 1 & 0.07 \\ 0.3 & 0.07 & 1 \end{bmatrix} \mathcal{A} = \begin{bmatrix} 1 & \lambda & 1 \\ \lambda & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (5)$$

$$\hat{\mathcal{C}} = \mathcal{C} \odot \mathcal{A} = \begin{bmatrix} 1 & 0.2 & 0.3 \\ 0.2 & 1 & 0.07 \\ 0.3 & 0.07 & 1 \end{bmatrix} \odot \begin{bmatrix} 1 & \lambda & 1 \\ \lambda & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (6)$$

where λ is the factor by which the correlation value is to be augmented. Since the advice asks to increase the correlation, we set $\lambda=0$. Thus,

$$\hat{\mathcal{C}} = \begin{bmatrix} 1 & 0.2 & 0.3 \\ 0.2 & 1 & 0.07 \\ 0.3 & 0.07 & 1 \end{bmatrix} \odot \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0.0 & 0.3 \\ 0.0 & 1 & 0.07 \\ 0.3 & 0.07 & 1 \end{bmatrix} \quad (7)$$

Similarly, if the advice says that we need to decrease the correlation coefficient between feature 1 and feature 3, we set $\lambda = \frac{1}{feat_{val}}$.

$$\hat{\mathcal{C}}_1 = \begin{bmatrix} 1 & 0.0 & 0.3 \\ 0.0 & 1 & 0.07 \\ 0.3 & 0.07 & 1 \end{bmatrix} \odot \begin{bmatrix} 1 & 0.2 & \frac{1}{0.3} \\ 0.2 & 1 & 1 \\ \frac{1}{0.3} & 1 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0.0 & 1.0 \\ 0.0 & 1 & 0.07 \\ 1.0 & 0.07 & 1 \end{bmatrix} \quad (8)$$

3.2 Advice-Guided Data Generation

After $\hat{\mathcal{C}}_1$ is constructed, we next generate data satisfying the constraints. To this effect, we employ the Iman-Conover method (Iman and Conover 1982), a distribution free method to define dependencies between distributional variables based on rank correlations such as Spearman or Kendell Tau correlations. We also tried the Ruscio-Kaczetow iteration (Ruscio and Kaczetow 2008) but it identifies the intermediate correlation matrix through an iterative, trial-and-error process and thus the results were not very encouraging. Thus, we made a focused choice to use Iman-Conover method which is a more principled approach. Since we deal with linear relationships between the features and assume a normal distribution and that Pearson coefficient has shown to perform equally well with the Iman-Conover method (Naveršnik and Rojnik 2012) due to the close relationship between Pearson and Spearman correlations, we use the Pearson correlations. Further, we assume that the features are Gaussian, justified by the fact that most clinical/lab/imaging data is continuous. The Iman-Conover method consists of the following steps:

[Step 1]: Create a random standardized matrix $\mathcal{M}$ with values $x \in \mathcal{M} \sim$ Gaussian distribution. This is obtained by the process of inverse transform sampling described next. Let $\mathcal{V}_1$ be a uniformly distributed random variable and $\mathcal{CDF}$ be the cumulative distribution function. For a sampled point v, $\mathcal{CDF}(v) = \mathcal{P}(V \leq v)$. Thus, to generate samples, the values $v \sim \mathcal{V}$ are passed through $\mathcal{CDF}^{-1}$ to obtain the desired values x [$\mathcal{CDF}^{-1}(v) = \{x | \mathcal{CDF}(x) \leq v, v \in [0, 1]\}$]. Thus for Gaussian,

$$\mathcal{CDF}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp^{\frac{-x^2}{2}} dx = \frac{1}{\sqrt{2\pi}} \int_{0}^{x} \exp^{\frac{-x^2}{2}} dx$$
$$= [-\exp(\frac{-x^2}{2})]_0^x \quad (9)$$

The inverse CDF can be thus written as

$$\mathcal{CDF}^{-1}(v) = 1 - \exp(\frac{-x^2}{2}) \leq v \quad (10)$$

and the desired values $x \in \mathcal{M}$ can be obtained as

$$x = \sqrt{2 ln(1 - v)} \quad (11)$$

[Step 2]: Calculate the correlation matrix $\mathcal{E}$ of $\mathcal{M}$.

[Step 3]: Calculate the Cholesky decomposition $\mathcal{F}$ of the correlation matrix $\mathcal{E}$. Cholesky decomposition (Scheuer and Stoller 1962) of a *positive-definite* matrix is the product of a lower triangular matrix and its conjugate transpose. Note that for Cholesky decomposition to be unique, the target matrix should be positive definite, (such as the covariance matrix) whereas the correlation matrix, used in our algorithm, is only positive semi-definite.

We enforce positive-definiteness by repeated addition of very small values to the diagonal of the correlation matrix until positive-definiteness is ensured. Given a symmetric and positive definite matrix $\mathcal{E}$, its Cholesky decomposition $\mathcal{F}$ is such that $\mathcal{E} = \mathcal{F} \cdot \mathcal{F}^\top$.

[Step 4]: Calculate the Cholesky decomposition $\mathcal{Q}$ of the correlation matrix obtained after modifications based on human advice, $\hat{\mathcal{C}}$. As above the Cholesky decomposition is such that $\hat{\mathcal{C}} = \mathcal{Q} \cdot \mathcal{Q}^\top$.

[Step 5]: Calculate the reference matrix $\mathcal{T}$ by transforming the sampled matrix $\mathcal{M}$ from step 1 to have the desired correlations of $\hat{\mathcal{C}}$, by using their Cholesky decompositions.

[Step 6]: Rearrange values in columns of the generated data $\mathcal{G}_1$ to have the same ordering as corresponding column in reference matrix $\mathcal{T}$ to obtain the final generated data $\tilde{\mathcal{G}}_1$.

Cholesky decomposition to model correlations: Given an randomly generated data set with no correlations $\mathcal{P}$, a correlation matrix $\mathcal{C}$ and its Cholesky decomposition $\mathcal{Q}$, data that faithfully follows the given correlations $\in \mathcal{C}$ can be generated by the product of the obtained lower triangular matrix with the original uncorrelated data i.e. $\hat{\mathcal{P}} = \mathcal{Q}\mathcal{P}$. The correlation of the newly obtained data, $\hat{\mathcal{P}}$ is,

$$Corr(\hat{\mathcal{P}}) = \frac{Cov(\hat{\mathcal{P}})}{\sigma_{\hat{\mathcal{P}}}} = \frac{\mathbf{E}[\hat{\mathcal{P}}\hat{\mathcal{P}}^\top] - \mathbf{E}[\hat{\mathcal{P}}]\mathbf{E}[\hat{\mathcal{P}}]^\top}{\sigma_{\hat{\mathcal{P}}}} \quad (12)$$

Since we consider data $\hat{\mathcal{P}}$ from a Gaussian distribution with zero mean and unit variance,

$$Corr(\hat{\mathcal{P}}) = \frac{\mathbf{E}[\hat{\mathcal{P}}\hat{\mathcal{P}}^\top] - \mathbf{E}[\hat{\mathcal{P}}]\mathbf{E}[\hat{\mathcal{P}}]^\top}{\sigma_{\hat{\mathcal{P}}}}$$
$$= \mathbf{E}[\hat{\mathcal{P}}\hat{\mathcal{P}}^\top] = \mathbf{E}[(\mathcal{Q}\mathcal{P})(\mathcal{Q}\mathcal{P})^\top]$$
$$= \mathbf{E}[\mathcal{Q}\mathcal{P}\mathcal{Q}^\top\mathcal{P}^\top] = \mathcal{Q}\mathbf{E}[\mathcal{P}\mathcal{P}^\top]\mathcal{Q}^\top = \mathcal{Q}\mathcal{Q}^\top = \mathcal{C} \quad (13)$$

Thus Cholesky decomposition can capture the desired correlations faithfully and can be used for generating correlated data. Since we already have a normal sampled matrix $\mathcal{M}$ and a calculated correlation $\mathcal{E}$ of $\mathcal{M}$, we only need to calculate a reference matrix (step 5).

Algorithm 1 HA-GAN: **H**uman-**A**llied **G**enerative **A**dversarial **N**etworks

---

1: **procedure** HA-GAN(Data $\mathcal{G}$, Generator G, Discriminator D, Correlation Advice ψ)
2:     **for** epochs k=1 to K **do**
3:         Sample $\mathbb{Z}_\mathfrak{m}$, $\mathfrak{m}$ examples from noise vector $\mathbb{Z}$
4:         Sample $\mathbb{X}_\mathfrak{m}$, $\mathfrak{m}$ examples from real data $\mathbb{X}$
5:         $\mathcal{G}_1=\mathcal{G}(\mathbb{Z}_\mathfrak{m})$ ▷ generate data from noise using generator
6:         **if** k mod N is 0 **then** ▷ number of epochs $1 \leq N \leq K$
7:             $\tilde{\mathcal{G}}_1 = $ IMAN-CONOVER$(\mathcal{G}_1,\psi)$
8:             $\hat{y} = \mathcal{D}(\mathbb{X}_\mathfrak{m}, \tilde{\mathcal{G}}_1)$ ▷ pass generated data after advice to discriminator; backpropagate error
9:         **else**
10:             $\hat{y} = \mathcal{D}(\mathbb{X}_\mathfrak{m}, \mathcal{G}_1)$ ▷ backpropagate error; no advice
11:         **end if**
12:     **end for**
13: **end procedure**

14: **procedure** IMAN-CONOVER(Generated data $\mathcal{G}_1$, Correlation Advice ψ)
15:     $\mathcal{C} = $ CALCCORR$(\mathcal{G}_1)$ ▷ calculate correlation matrix
16:     $\lambda_{inc}= \frac{1}{max(\mathcal{C})}$ ▷ calculate correlation increasing factor
17:     $\lambda_{dec}= max(\mathcal{C})$ ▷ calculate correlation decreasing factor
18:     $\mathcal{A}=$ CREATEADVICEMAT(ψ)
19:     $\hat{\mathcal{C}} = \mathcal{C} \odot \mathcal{A}$ ▷ $\lambda \in \mathcal{A}$.
20:     **for** t epochs **do**
21:         $\mathcal{M} = $ INVSAMP$(\mathfrak{m}, \mathfrak{d})$ ▷ $\mathfrak{m}$: #examples, $\mathfrak{d}$:#features
22:         $\mathcal{E} = \frac{\mathcal{M}^\top \cdot \mathcal{M}}{\mathfrak{m}}$ ▷ correlation matrix of $\mathcal{M}$
23:         calculate $\mathcal{F}$ s.t. $\mathcal{E}=\mathcal{F} \cdot \mathcal{F}^\top$ ▷ Cholesky decomp of $\mathcal{E}$
24:         calculate $\mathcal{Q}$ s.t. $\hat{\mathcal{C}}=\mathcal{Q} \cdot \mathcal{Q}^\top$ ▷ Cholesky decomp of $\hat{\mathcal{C}}$
25:         calculate $\mathcal{T} = \mathcal{M}\mathcal{F}^{-1}\mathcal{Q}$ ▷ reference matrix with desired correlations
26:         $\mathcal{R} = $ RANK$(\mathcal{T})$
27:         $\tilde{\mathcal{G}}_1 = $ REORDER$(\mathcal{R},\mathcal{G}_1)$
28:     **end for**
29:     **return** $\tilde{\mathcal{G}}_1$
30: **end procedure**

31: **procedure** INVSAMP(number of sampled examples $\mathfrak{m}$, number of features in data $\mathfrak{d}$)
32:     $\mathfrak{a}=$ generate $\mathfrak{m}$ random numbers
33:     $\mathfrak{p} = $ QUALTILEFUNCTION$(\mathfrak{a})$ ▷ induce Gaussian dist
34:     $\hat{\mathfrak{p}} = $ NORMALISE$(\mathfrak{p})$
35:     $\mathcal{M} = $ PERMUTE$(\hat{\mathfrak{p}})$ ▷ shuffle the sampled numbers
36: **end procedure**

3.3 Human-Allied GAN Training

Since the human expert advice is provided independent of the GAN architecture, our method is agnostic of the underlying GAN architecture. We make use of Wasserstein GAN (WGAN) architecture since its shown to be more stable while training and can handle mode collapse (Arjovsky, Chintala, and Bottou 2017). Only the error backpropagation values differ when we are using the data generated by the underlying GAN or the data generated by the Iman-Conover method. Algorithm 1 shows the overall process of Human-Allied GANs. Our algorithm starts with the general process of training a GAN where the generator takes random noise as an input and generates data which is then passed, along with the real data, to the discriminator. The discriminator tries to identify

the real and generated data and the error is back propagated to the generator. After every specified number of iterations, the correlations between features $\mathcal{C}$ in the generated data is obtained and a new correlation matrix $\hat{\mathcal{C}}$, is obtained with respect to the expert advice. A new data set is generated wrt $\hat{\mathcal{C}}$ using the Iman-Conover method and then passed to the discriminator along with the real data set.

3.4 Introducing Privacy in HA-GAN

Ensuring differential privacy when trying to capture the hidden relationships such as cross-feature correlations is difficult (Liu, Chakraborty, and Mittal 2016) and remains an open problem. As an alternative method, we introduce noise in the correlation matrices. Since the key goal of our work is to introduce human knowledge, we will explore the guarantees for privacy as future work. Our generator focuses on faithful modeling of feature correlations, thus, introducing correlation noise is akin to Laplacian noise for differential privacy. We use the method outlined in Hardin et al., (2013) to generate noisy correlation matrices (after lines 19 and 22 in algorithm 1) which are then used for generating synthetic data. The method is as follows: given a correlation matrix $\hat{\mathcal{C}}$ of size $\mathcal{N} \times \mathcal{N}$ obtained after advice, the maximum noise level to be inserted ϵ and the dimension of the noise space $\mathcal{M}$, the initial step is to select $\mathcal{N}$ unit vectors from $\mathbb{R}^\mathcal{M}$ thus forming a matrix $\mathcal{U}$ of size $\mathcal{M} \times \mathcal{N}$. The noisy correlation matrix $\hat{\mathcal{C}}_n$ can then be written as,

$$\hat{\mathcal{C}}_n = \hat{\mathcal{C}} + \epsilon(\mathcal{U}^T\mathcal{U} - \mathcal{I}) \qquad (14)$$

where $\mathcal{I}$ is an identity matrix and $|\hat{\mathcal{C}}_{n(ij)} - \hat{C}_{ij}| \leq \epsilon$ for $1 \leq i,j \leq \mathcal{N}$. It can be shown by using the Gerhsgorin's disk theorem (Gerschgorin 1931) and Weyl's inequalities (Weyl 1912) that the obtained correlation matrix is positive semi-definite and thus admits a Cholesky decomposition. We refer to Hardin et al., (2013) for a formal proof.

Data set	#Features	#+ve examples	#-ve examples
NS	19	44	6
PPMI	35	554	919
ADNI	68	76	260
MIMIC	18	5813	40707

Table 1: Evaluation domains and their properties.

4 Experiments

We aim to answer the following questions explicitly:

Q1. Data generation: Does providing advice to GANs help in generating better quality data?

Q2. Sample efficiency: Are GANs with advice effective for data sets that have few examples?

Q3. Advice quality: How does bad advice affect the quality of generated data?

Q4. Class imbalance: How well does human advice handle class imbalance?

Q5. GANs comparison: How does our method compare to state-of-the-art GAN architectures.

Q6. Privacy: How does our method handle privacy?

Q7. Effect of stabilization approaches: Can our method take advantage of stabilized GAN training and generate better quality data?

Q8. Capturing correlations: How well does HA-GAN capture correlations in generated data?

Data sets: We consider 4 *real clinical data sets* (Table 1).

1. **Nephrotic Syndrome (NS)** is a novel data set of symptoms that indicate kidney damage. This consists of 50 kidney biopsy images along with the clinical reports sourced from Dr Lal PathLabs, India [2]. We use the clinical reports that consist of the values for kidney tissue diagnosis which can confirm the clinical diagnosis and help to identify high-risk patients and influence treatment decisions and help medical practitioners to plan and prognosticate treatments.

2. **Parkinson's (PPMI)** data consists of aggregated clinical data for prediction of Parkinson's disease and is obtained from an observational study (Marek et al. 2011; Dhami et al. 2017) with the main aim of identifying features or biomarkers that impact Parkinson's disease progression.

3. **Alzheimer's (ADNI)** data is obtained from the Alzheimer's Disease Neuroimaging Initiative study (Petersen et al. 2010) aimed to find progression markers for Alzheimer's disease and develop treatments that can slow the progression. We consider data obtained by processing the MRI images of 336 patients.

4. **MIMIC** database (Johnson et al. 2016) consists of deidentified information of patients admitted to critical care units at a large tertiary care hospital. The features included are predominately time window aggregations of physiological measurements from the medical records. We selected relevant lab results, vital sign observations and feature aggregations.

Baselines: We consider 4 GAN models that are specifically proposed for medical data set generation or tabular data generation.

1. **medGAN** (Choi et al. 2017): uses an encoder decoder framework along with a GAN framework for electronic health record data generation.

2. **medWGAN and medBGAN** (Baowaly et al. 2019): are the variants of medGAN architecture and uses WGAN with gradient penalty and boundary seeking (Hjelm et al. 2018) to generate better quality synthetic data.

3. **Conditional Tabular GAN (CTGAN)** (Xu et al. 2019): extends the GAN architecture by augmenting the training procedure with mode-specific normalization and using a conditional generator.

Advice Acquisition Here we compile the sources from which we obtain the advice,
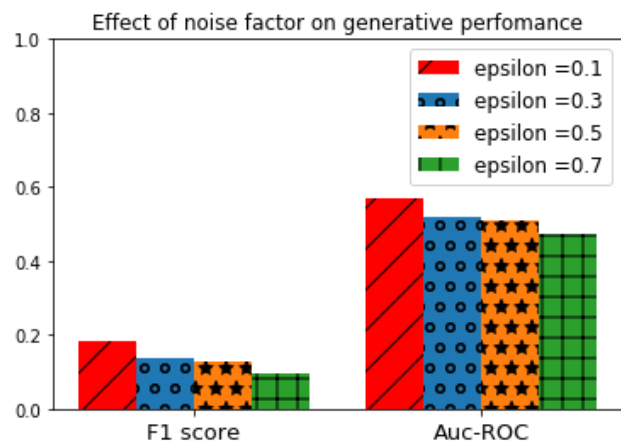
[2]https://www.lalpathlabs.com/

Figure 2: Effect of ϵ on generative performance.

1. *Nephrotic Syndrome:* This is a novel real data set and the **advice is obtained from a nephrologist in India**. According to the problem statement from the expert, nephrotic syndrome involves the loss of a lot of protein and nephritic syndrome involves the loss of a lot of blood through urine. A kidney biopsy is often required to diagnose the underlying pathology in patients with suspected glomerular disease. The goal of the project is to build a clinical support system that predicts the disease using clinical features, thus reducing the need of kidney biopsy. *Since the data collection is scarce, a synthetic data set can help in better understanding of the disease from the clinical features.*

2. *Parkinson's:* The information regarding the correlation between various clinical features is present **within the PPMI study** (Marek et al. 2011) and is used in this work. We also obtain correlation information after contacting the authors of Dhami et al. 2017 where an **expert advice from a neurologist** is used to select the set of important features and their correlations.

3. *Alzheimer's:* The information regarding the correlation between various clinical features is present **within the ADNI study** (Petersen et al. 2010) and is used in this work. Some correlations were also obtained from a **study of Alzheimer's disease and its correlation with Mini Mental State Examination** (Park, Seo, and others 2011).

4. *MIMIC:* The feature set and the expected correlations are obtained in consultation with **trauma experts at a major hospital**.

Both the generator and discriminator are neural networks with 4 hidden layers. To measure the quality of the generated data we make use of the *train on synthetic, test on real* (TSTR) method as proposed in (Esteban, Hyland, and Rätsch 2017). We use gradient boosting with 100 estimators and a learning rate of 0.01 as the underlying model. We train the GAN for 10K epochs and provide correlation advice every 1K iterations. Although the knowledge does not change, we hypothesize that providing advice multiple times pushes the generated feature value towards the desired correlations. Just like learning a simple linear classifier where every iteration

Data	Methods	Recall	F1	ROC	PR
NS	GAN	0.584	0.666	0.509	0.911
	HA-GAN$_{BA}$	0.42	0.511	0.518	0.886
	medGAN	N/A	N/A	N/A	N/A
	medWGAN	N/A	N/A	N/A	N/A
	medBGAN	N/A	N/A	N/A	N/A
	CTGAN	0.0	0.0	0.5	0.880
	HA-GAN$_{GA}$	**1.0**	**0.943**	**0.566**	**0.947**
PPMI	GAN	0.657	0.324	0.490	0.477
	HA-GAN$_{BA}$	0.046	0.079	0.484	0.258
	medGAN	0.819	0.749	0.799	0.789
	medWGAN	0.871	0.791	0.836	0.822
	medBGAN	**0.892**	**0.805**	**0.848**	**0.833**
	CTGAN	0.0	0.0	0.498	0.376
	HA-GAN$_{GA}$	0.576	0.622	0.705	0.706
ADNI	GAN	0.725	0.339	0.502	0.511
	HA-GAN$_{BA}$	0.011	0.017	0.491	0.423
	medGAN	0.0	0.0	0.5	0.613
	medWGAN	0.0	0.0	0.5	0.613
	medBGAN	0.0	0.0	0.5	0.613
	CTGAN	0.053	0.073	0.467	0.220
	HA-GAN$_{GA}$	**0.949**	**0.388**	**0.545**	**0.592**
MIMIC	GAN	0.122	0.119	0.495	0.174
	HA-GAN$_{BA}$	0.285	0.143	0.459	0.235
	medGAN	0.374	0.163	0.478	0.279
	medWGAN	0.0	0.0	0.5	0.562
	medBGAN	0.0	0.0	0.5	0.562
	CTGAN	0.0	0.0	0.5	0.125
	HA-GAN$_{GA}$	**0.979**	**0.263**	**0.598**	**0.567**

Table 2: TSTR Results (≈ 3 *dec.*). N/A in Nephrotic Syndrome denotes that all generated labels were of a single class (0 in our case) and thus we were not able to run the discriminative algorithm in the TSTR method. GA=good advice, BA=bad advice, ROC=AUC-ROC, PR=AUC-PR.

learns the linear boundary, here every application of the advice makes the correlations between generated features closer to the real data based on the expert advice. We have made our code available anonymously at: *https://rb.gy/hfx2mh*.

Q1. Data generation: Table 2 shows the results of the TSTR method with data generated with (HA-GAN$_{GA}$) and without advice (GAN). It shows that the data generated with advice has higher TSTR performance than the data generated without advice across all data sets and all metrics. Thus, to answer **Q1**, providing advice to generative adversarial networks captures the relationship between features better and thus are able to generate better quality synthetic data. *The results in case of MIMIC data set presents an important observation*: Although the number of samples in the data set are higher, the performance, even when bad advice is given, against simple GAN architecture is better due to the fact that MIMIC data is highly curated and includes feature aggregation over time which might lead to some correlations being lost and some new correlations to form. Thus if bad advice is provided there is less effect in the generated data but the good advice captures the remaining of the correlations well to give superior performance. The underlying GAN, which anyhow is unable to capture the hidden relationships naturally performs poorly.

Q2. Sample efficiency: GANs with advice are especially impressive in Nephrotic Syndrome data which consists of

only 50 examples across all metrics. It must be mentioned that **all the considered data sets are small in size, except MIMIC, when compared to the number of samples typically required to train a GAN model**. Figure 3 shows the results with varying number of iterations in 3 domains. The effect of advice is evident even when the number of training iterations is small and the model is able to learn better than the model without advice. Thus, we realize an important property of incorporating human guidance in the GAN model and can answer **Q2** affirmatively. *The use of advice appears to make GANs effective in the presence of sparse data.*

Q3. Advice quality: Table 2 also presents the results for data generated with bad advice (HA-GAN$_{BA}$). To simulate bad advice, we follow a simple process: if the advice specifies that the correlation between features is high, we set the correlations in $\hat{C}$ to 0 and if the advice specifies that the correlation is low, we set the correlations in $\hat{C}$ to be either 1 or -1 based on whether the original correlation is positive or negative. As results show in table 2, providing bad advice adversely affects the performance answering **Q3**, thus validating our claim that human is an ally in this learning.

Q4. Class imbalance: Nephrotic syndrome, Alzheimer's and MIMIC data sets are *relatively* imbalanced with a pos to neg ratio of $\approx$ 8:1, 1:3.5 and 1:7 respectively. Most of the medical data sets, except highly curated data sets, are imbalanced. A data generator model should be able to handle this imbalance. Since our method explicitly focuses on the correlations between features and generates better quality data based on such relationships between features, our method is quite **robust to the imbalance in the underlying data**. This can be seen in the results in table 2 where advice based data generation outperforms the non-advice and bad advice based data generation. Thus, we can answer **Q4** affirmatively.

Q5. GANs comparison: To answer **Q5** we compare our method to 4 GAN architectures used specifically for medical and tabular data and the results are shown in table 2. The results show that our method, with good advice, outperforms the baseline in 3 out of the 4 domain thus showing the effectiveness of our method.

Q6. Privacy: Since all the considered data sets contain only clinical values and thus have no identifiers, to show the effect of incorporating privacy in our model we use UCI Epileptic Seizure data (UES) and Kaggle Cervical cancer data (KCC) (Fernandes, Cardoso, and Fernandes 2017). Table 3 presents our results after including privacy in our model (after incorporating good advice) using noisy correlation matrices with the noise factor ϵ=0.3 and $\mathcal{M}$=5. The results show that adding noise does decrease the performance of our model, as expected, though the relationships between features are still captured well. Figure 2 shows the effect of ϵ on the quality of generated data for the UES data set. Quality of the generated data decreases with increase in amount of noise in the correlation matrices used to create the generated data though not drastically. This shows that the noise with human advice generates data that respects privacy while keeping the relationship between features intact. This answers **Q6**.

Q7. Effect of stabilization approaches A potential reason for the improved performance of HA-GAN (including

Data set	Methods	Recall		F1		AUC-ROC		AUC-PR	
		AdaB	GB	AdaB	GB	AdaB	GB	AdaB	GB
UES	HA-GAN (A + P)	0.411	0.993	0.120	0.121	0.509	0.503	0.286	0.529
	HA-GAN (A)	0.512	0.810	0.271	0.317	0.529	0.517	0.408	0.522
	GAN	0.495	0.651	0.306	0.316	0.473	0.492	0.397	0.465
KCC	HA-GAN (A + P)	0.247	0.865	0.137	0.131	0.521	0.542	0.223	0.472
	HA-GAN (A)	0.754	0.896	0.124	0.122	0.520	0.485	0.419	0.483
	GAN	0.256	0.204	0.243	0.276	0.587	0.596	0.291	0.478

Table 3: TSTR results ($\approx 3\ decimals$) for our model including privacy. (A + P) is advice + privacy.
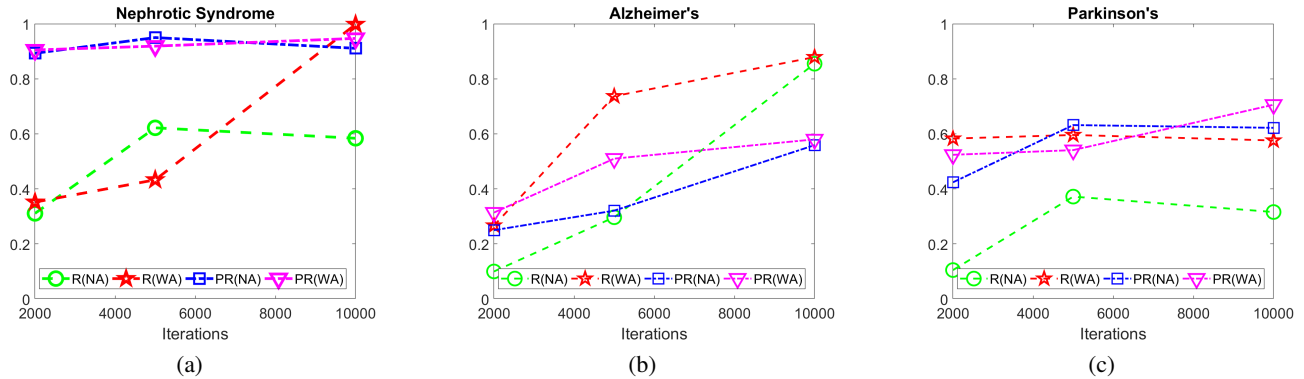
Figure 3: Varying results with respect to iterations. R=recall, PR=AUC-PR, NA=no advice, and WA=with advice.

Data	Methods	Recall	F1	ROC	PR
NS	GAN_P	0.586	0.683	0.521	0.938
	HA-GAN_{PBA}	0.557	0.614	0.544	0.898
	HA-GAN_{PGA}	**1.0**	**0.955**	**0.578**	**0.956**
ADNI	GAN_P	0.778	0.414	0.536	0.541
	HA-GAN_{PBA}	0.215	0.112	0.498	0.436
	HA-GAN_{PGA}	**0.966**	**0.423**	**0.574**	**0.628**

Table 4: TSTR Results ($\approx 3\ dec.$) for WGAN with gradient penalty (GAN_P) and $P(GA/BA)$=good/bad advice in GAN_P

the cases with Bad Advice) is that the procedure is essentially adding instance noise, which is known to stabilize GAN training (Gulrajani et al. 2017). Thus, to clearly tease out how much improvement is actually due to the expert knowledge we test our method with a GAN stabilization approach, namely, *gradient penalties for Wasserstein GANs* (we call it GAN_P) on 2 data sets. Table 4 shows that a simple GAN stabilization approach can improve results when compared to vanilla WGAN. After introducing the good advice, the results further improve not only when compared to GAN_P but also when compared to HA-GAN_{GA} in table 2. Similarly for bad advice, the results are adversely affected when compared to GAN_P but improve when compared to HA-GAN_{BA}. Thus although GAN stabilization approach improve the performance, introduction of advice further elevates the quality of synthetic data generation thereby answering **Q7**.

Q8. Capturing Correlations: The premise of our work is based on the observation that simple GAN architectures cannot capture the hidden relationships in the medical data sets which can lead to generation of poor quality synthetic data. It

can also make trusting the synthetic data hard since missing any of the hidden relationships in medical data sets can lead to potentially serious issues. In this work, we make use of canonical correlation (Hotelling 1936) which is a method of quantifying linear relationships between two multidimensional variables. It can be thought of as a dimensionality reduction technique akin to principal component analysis, the difference lying in what the variables in the projected space represent. Given two multi-dimensional data sets of random variables, with correlations among the variables, canonical-correlation analysis (CCA) obtains linear combinations of the data set features with maximum correlation. Figures 4 - 7 show a distribution plot of the first principal component obtained by the CCA on the considered real data sets and the obtained synthetic data sets. Since we start with no information about the correlations in the real data set and keep including the expert given correlations, we do not expect to perfectly capture the correlations between all the features but the aim is to be near to the real data correlation. In all data sets, data generated with advice better captures relationships between features answering **Q8** affirmatively.

5   Conclusion

We presented a new GAN formulation that employs correlation information between features as advice to generate new correlated data and train the underlying GAN model. We tested our model on real clinical data sets and show that incorporating advice helps generate good quality synthetic medical data. There are several future directions. **1.** Providing advice only when required (active learning) can allow for
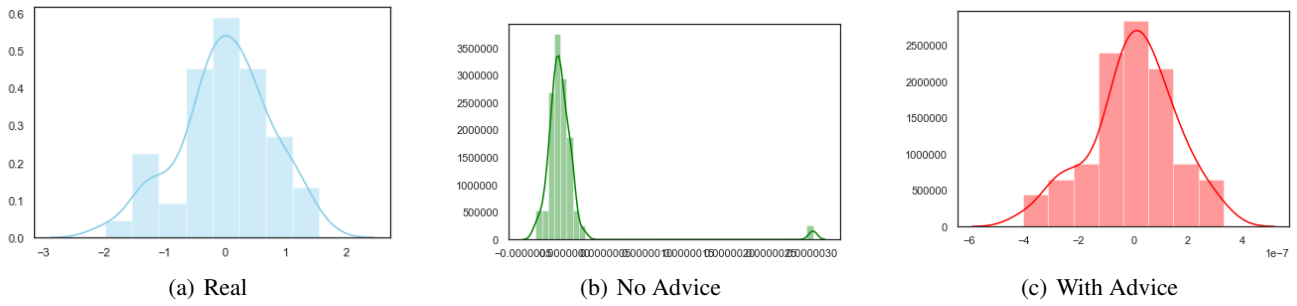
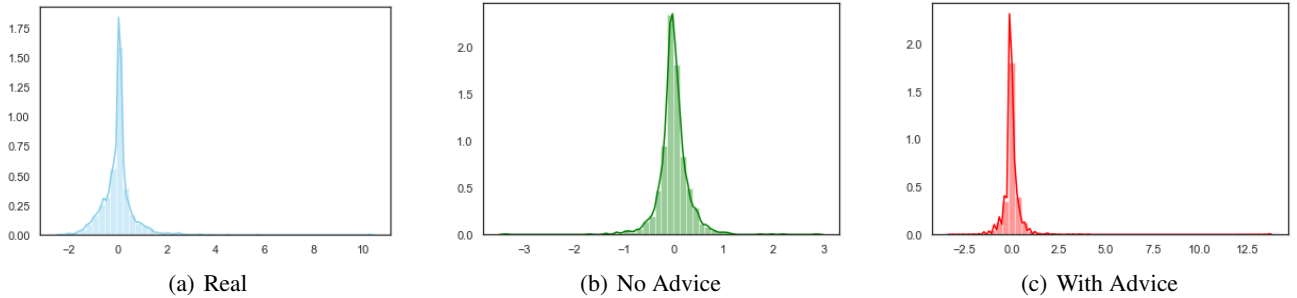Figure 4: Canonical correlation distribution - Nephrotic Syndrome data.

Figure 5: Canonical correlation distribution - Parkinson's data.
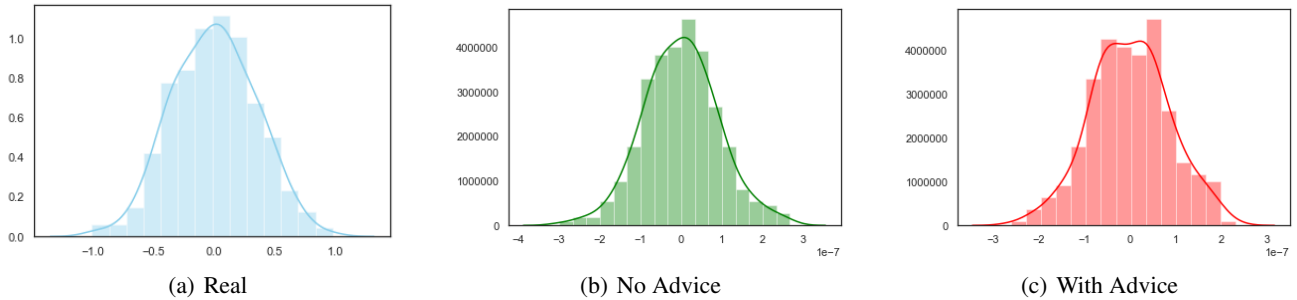

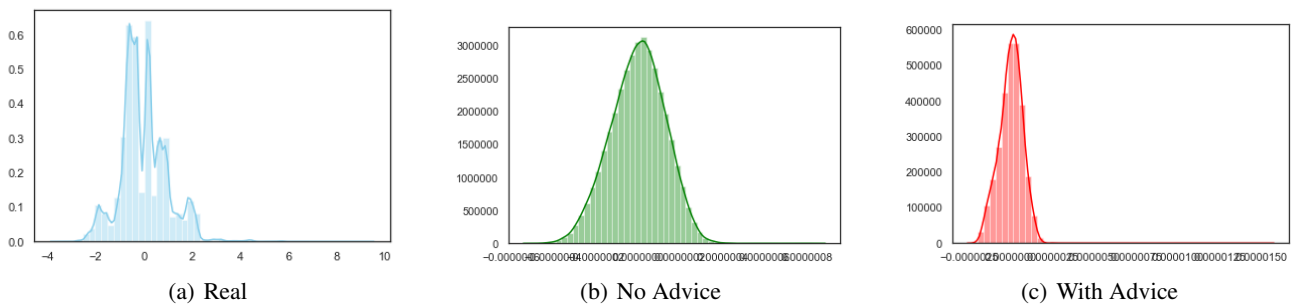Figure 6: Canonical correlation distribution - Alzheimer's data.

Figure 7: Canonical correlation distribution - MIMIC data.

significant reduction in the amount of effort on the human side. **2.** Several advice options, such as posterior regularization (Ganchev et al. 2010), can be explored to capture feature relationships explicitly. **3.** Although our data is de-identified, thus eliminating the need of differential privacy (Dwork 2008), a general framework that upholds the data privacy along the lines of using Cholesky decomposition is a natural next step.

References

Antipov, G.; Baccouche, M.; and Dugelay, J.-L. 2017. Face aging with conditional generative adversarial networks. In *ICIP*.

Arjovsky, M., and Bottou, L. 2017. Towards principled methods for training generative adversarial networks. In *ICLR*.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *ICML*.

Baowaly, M. K.; Lin, C.-C.; Liu, C.-L.; and Chen, K.-T. 2019. Synthesizing electronic health records using improved generative adversarial networks. *JAMA*.

Braziunas, D., and Boutilier, C. 2006. Preference elicitation and generalized additive utility. In *AAAI*.

Buczak, A. L.; Babin, S.; and Moniz, L. 2010. Data-driven approach for creating synthetic electronic medical records. *BMC medical informatics and decision making*.

Choi, E.; Biswal, S.; Malin, B.; Duke, J.; Stewart, W. F.; and Sun, J. 2017. Generating multi-label discrete patient records using generative adversarial networks. In *MLHC*.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine Learning*.

Dhami, D. S.; Soni, A.; Page, D.; and Natarajan, S. 2017. Identifying parkinson's patients: A functional gradient boosting approach. In *AIME*.

Dinov, I. D. 2016. Volume and value of big healthcare data. *Journal of medical statistics and informatics*.

Dwork, C. 2008. Differential privacy: A survey of results. In *TAMS*.

Esteban, C.; Hyland, S. L.; and Rätsch, G. 2017. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*.

Fernandes, K.; Cardoso, J. S.; and Fernandes, J. 2017. Transfer learning with partial observability applied to cervical cancer screening. In *IbPRIA*.

Frid-Adar, M.; Klang, E.; Amitai, M.; Goldberger, J.; and Greenspan, H. 2018. Synthetic data augmentation using gan for improved liver lesion classification. In *ISBI*.

Fung, G. M.; Mangasarian, O. L.; and Shavlik, J. W. 2003. Knowledge-based support vector machine classifiers. In *NIPS*.

Ganchev, K.; Gillenwater, J.; Taskar, B.; et al. 2010. Posterior regularization for structured latent variable models. *JMLR*.

Gerschgorin, S. A. 1931. On delimiting the eigenvalues of a matrix. *Izv. Akad. Nauk. USSR. Otd. Fiz-Mat. Nauk*.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

Guibas, J. T.; Virdi, T. S.; and Li, P. S. 2017. Synthetic medical images from dual generative adversarial networks. *arXiv preprint arXiv:1709.01872*.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. 2017. Improved training of wasserstein gans. *NeurIPS*.

Habeeb, H.; Anand, A.; Mausam, M.; and Singla, P. 2017. Coarse-to-fine lifted map inference in computer vision. In *IJCAI*.

Hardin, J.; Garcia, S. R.; and Golan, D. 2013. A method for generating realistic correlation matrices. *The Annals of Applied Statistics*.

Hjelm, R. D.; Jacob, A. P.; Che, T.; Trischler, A.; Cho, K.; and Bengio, Y. 2018. Boundary-seeking generative adversarial networks. *ICLR*.

Hotelling, H. 1936. Relations between two sets of variates. In *Biometrika*.

Hu, Z.; Yang, Z.; Salakhutdinov, R. R.; Qin, L.; Liang, X.; Dong, H.; and Xing, E. P. 2018. Deep generative models with learnable knowledge constraints. In *NeurIPS*.

Huang, C.; Kairouz, P.; Chen, X.; Sankar, L.; and Rajagopal, R. 2017. Context-aware generative adversarial privacy. *Entropy*.

Huang, C.; Kairouz, P.; Chen, X.; Sankar, L.; and Rajagopal, R. 2018. Generative adversarial privacy. *arXiv preprint arXiv:1807.05306*.

Iman, R. L., and Conover, W.-J. 1982. A distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics-Simulation and Computation*.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Li-wei, H. L.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *CVPR*.

Kuhn, H. W., and Tucker, A. W. 1953. *Contributions to the Theory of Games*.

Kunapuli, G.; Odom, P.; Shavlik, J. W.; and Natarajan, S. 2013. Guiding autonomous agents to better behaviors through human advice. In *ICDM*.

Le, Q. V.; Smola, A. J.; and Gärtner, T. 2006. Simpler knowledge-based support vector machines. In *ICML*.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature*.

Li, M.; Huang, H.; Ma, L.; Liu, W.; Zhang, T.; and Jiang, Y. 2018. Unsupervised image-to-image translation with stacked cycle-consistent adversarial networks. In *ECCV*.

Li, Y.; Swersky, K.; and Zemel, R. 2015. Generative moment matching networks. In *ICML*.

Liu, M.-Y., and Tuzel, O. 2016. Coupled generative adversarial networks. In *NIPS*.

Liu, M.-Y.; Breuel, T.; and Kautz, J. 2017. Unsupervised image-to-image translation networks. In *NIPS*.

Liu, C.; Chakraborty, S.; and Mittal, P. 2016. Dependence makes you vulnberable: Differential privacy under dependent tuples. In *NDSS*.

Mahmood, F.; Chen, R.; and Durr, N. J. 2018. Unsupervised reverse domain adaptation for synthetic medical images via adversarial training. *IEEE transactions on medical imaging*.

Mansinghka, V. K.; Kemp, C.; Tenenbaum, J. B.; and Griffiths, T. L. 2006. Structured priors for structure learning. In *UAI*.

Mao, X.; Li, Q.; Xie, H.; Lau, R. Y.; Wang, Z.; and Paul Smolley, S. 2017. Least squares generative adversarial networks. In *ICCV*.

Marek, K.; Jennings, D.; Lasch, S.; Siderowf, A.; Tanner, C.; Simuni, T.; Coffey, C.; Kieburtz, K.; Flagg, E.; Chowdhury, S.; et al. 2011. The parkinson progression marker initiative (ppmi). *Progress in neurobiology*.

Mitchell, T. M. 1980. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research, Rutgers Univ. New Jersey.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. *ICLR*.

Natarajan, S.; Tadepalli, P.; Dietterich, T. G.; and Fern, A. 2008. Learning first-order probabilistic models with combining rules. *Annals of Mathematics and Artificial Intelligence*.

Naveršnik, K., and Rojnik, K. 2012. Handling input correlations in pharmacoeconomic models. *Value in Health*.

Odom, P., and Natarajan, S. 2018. Human-guided learning for probabilistic logic models. *Frontiers in Robotics and AI*.

Odom, P.; Khot, T.; Porter, R.; and Natarajan, S. 2015. Knowledge-based probabilistic logic learning. In *AAAI*.

Paganini, M.; de Oliveira, L.; and Nachman, B. 2018. Calogan: Simulating 3d high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. *Physical Review D*.

Park, H.; Seo, J.; et al. 2011. Application of multidimensional scaling to quantify shape in alzheimer's disease and its correlation with mini mental state examination: a feasibility study. *Journal of neuroscience methods*.

Petersen, R. C.; Aisen, P.; Beckett, L. A.; Donohue, M.; Gamst, A.; Harvey, D. J.; Jack, C.; Jagust, W.; Shaw, L.; Toga, A.; et al. 2010. Alzheimer's disease neuroimaging initiative (adni): clinical characterization. *Neurology*.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*.

Ruscio, J., and Kaczetow, W. 2008. Simulating multivariate nonnormal data using an iterative algorithm. *Multivariate Behavioral Research*.

Scheuer, E. M., and Stoller, D. S. 1962. On the generation of normal random vectors. *Technometrics*.

Schölkopf, B.; Simard, P.; Smola, A. J.; and Vapnik, V. 1998. Prior knowledge in support vector kernels. In *Advances in neural information processing systems*, 640–646.

Shavlik, J. W., and Towell, G. G. 1989. Combining explanation-based learning and artificial neural networks. In *Proceedings of the sixth international workshop on Machine learning*. Elsevier.

Towell, G. G., and Shavlik, J. W. 1994. Knowledge-based artificial neural networks. *Artificial intelligence*.

Wang, Y.; Yu, B.; Wang, L.; Zu, C.; Lalush, D. S.; Lin, W.; Wu, X.; Zhou, J.; Shen, D.; and Zhou, L. 2018a. 3d conditional generative adversarial networks for high-quality pet image estimation at low dose. *NeuroImage*.

Wang, Z.; Tang, X.; Luo, W.; and Gao, S. 2018b. Face aging with identity-preserved conditional generative adversarial networks. In *CVPR*.

Weyl, H. 1912. The asymptotic law of distribution of the eigenvalues of linear partial differential equations (with an application to the theory of cavity radiation). *Mathematical Annals*.

Wu, J.; Zhang, C.; Xue, T.; Freeman, B.; and Tenenbaum, J. 2016. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*.

Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; and Veeramachaneni, K. 2019. Modeling tabular data using conditional gan. *NeurIPS*.

Yang, S., and Natarajan, S. 2013. Knowledge intensive learning: Combining qualitative constraints with causal independence for parameter learning in probabilistic models. In *ECMLPKDD*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.