Synthetic Tabular Data Generation with Oblivious Variational Autoencoders: Alleviating the Paucity of Personal Tabular Data for Open Research

L Vivek Harsha Vardhan [1] **Stanley Kok** [1]

1. Introduction

Hospitals store a wealth of digital data in the form of tables. If these data could be publicly shared, they would expedite data-intensive AI research to the benefit of the medical organizations or society at large. However, these tabular data usually contain personal identifiers of patients, which in conjunction with sensitive medical information, pose serious privacy concerns. Thus access to such data are usually hampered by the high walls and deep moats of security processes, such as lengthy reviews by medical review boards.

To mitigate the privacy risks and to expedite data release, some early approaches sought to suppress, randomize, or perturb potentially identifiable information. However, such techniques were found to be susceptible to re-identification attacks, e.g., via background knowledge (Emam et al., 2011). In recent years, a series of research efforts have addressed the problem from another angle, mitigating the privacy risks through the generation of synthetic data that closely mimic the true underlying tabular data. Because the generated data are completely fake, there is little risk of personal attribute disclosure. Those approaches (Choi et al., 2017; Srivastava et al., 2017; Park et al., 2018; Xu et al., 2019) typically utilize deep generative models (such as variational autoencoders (Kingma & Welling, 2014) and generative adversarial networks (Goodfellow et al., 2014)) in the hope that the models' success in generating synthetic images (Lin et al., 2019) and text (Guo et al., 2018) would carry over to tabular data. Those approaches have shown promising empirical results, but have yet to achieve their full potential. Many of the deep generative models are transplanted almost wholesale from their initial image or text domains to the tabular one. Consequently, they inherit inductive biases that are more amenable to their original data types than tabular data. Though effective for predictive tasks on tables, decision trees and their variants have not been extensively used for the generative modeling of tabular data.

Combining the best of deep generative models and decision tree variants, we propose the Oblivious Variational Autoencoder (OVAE). OVAE embeds 'softened' oblivious decision trees (ODTs) in a variational autoencoder (VAE) to both encode tabular data into a latent representation, and to decode (generate) synthetic data from that representation. ODTs are good at representing decision manifolds that approximate the hyperplane boundaries usually present in tabular data. Thus the ODTs imbue a strong inductive bias for tabular data in OVAE, allowing it to preserve the distributional characteristics of the original tabular data in its generated synthetic data.

Our work is particularly pertinent in the current Covid-19 pandemic, during which a constant stream of Covid-19 patient data are collected by medical institutions, but are only accessible to a select group of researchers who are associated with the institutions. By using OVAE, we could preserve privacy of the patients and make high-fidelity fake patient data publicly available, allowing a wider span of intellectual resources and human ingenuity to be brought to bear on healthcare problems.

In sum, our contributions are as follows.

- We propose a new model called OVAE that combines ODTs with a VAE to generate tabular data. To our knowledge, we are the first to adapt a decision tree variant so that it can be used with a VAE for generating privacy-preserving synthetic tables.
- We extensively compare our OVAE model against several state-of-the-art baselines, and show that OVAE compares favorably against the baselines on 12 real-world datasets.

2. Related Work

Early approaches for synthetic table generation utilize statistical models. Those models typically form a multivariate probability distribution over all columns in a table, each of which is regarded as a random variable. To generate

[1]Department of Information Systems and Analytics, School of Computing, National University of Singapore, Singapore. Correspondence to: L Vivek Harsha Vardhan <harsha@comp.nus.edu.sg>.

Proceedings of the 37^{th} International Conference on Machine Learning, ICML HSYS Workshop, 2020.

synthetic data, the models draw a sample (a row of values) from the distribution, with one value per random variable (column). Examples of such statistical models include Bayesian networks (e.g., CLBN (Chow & Liu, 1968), PrivBayes (Zhang et al., 2017), and (Aviñó et al., 2018)) and copulas (Patki et al., 2016; Sun et al., 2019).

Another line of models for tabular data generation is based on deep learning, and is predominantly built upon generative adversarial networks (GANs) (Goodfellow et al., 2014). GANs are proposed to generate multi-categorical columns of discrete data in (Camino et al., 2018), and continuous laboratory time series data in (Yahi et al., 2017). ehrGAN (Che et al., 2017) generates data to augment scarce medical records in a semi-supervised manner. medGAN (Choi et al., 2017) combines a GAN and an autoencoder to model both continuous and binary table columns. VEEGAN (Srivastava et al., 2017) ameliorates the mode collapse problem in GANs through variational learning. tableGAN (Park et al., 2018) uses a convolutional neural network as the discriminator in its GAN to maximize the quality of a table's label column. Most recently, CTGAN (Xu et al., 2019) uses *mode-specific normalization* (MSN) for modeling continuous data with multiple modes, and introduces techniques for correctly modeling the minor categories in categorical data with skewed distributions.Another deep learning model that has been used for synthetic tabular data generation is the variational autoencoder (VAE). TVAE (Xu et al., 2019) is a vanilla VAE with fully connected neural networks in both its encoder and decoder. It also uses MSN to address the problems of mode collapse. In an extensive empirical comparison, TVAE outperforms CTGAN and other state-of-the-art GAN models, and is a frontrunner that we compare against.

Unlike all of the aforementioned models that do not impose strong tabular constraints, our proposed oblivious variational autoencoder (OVAE) incorporates a strong inductive bias for tabular data in the form of "softened" oblivious decision trees (ODTs) into a VAE. There exists a body of work integrating inductive biases in the form of grammars, templates, or other constraints into deep generative models, but those biases are primarily applicable to domains other than tabular data. Unlike the other research, OVAE incorporates an inductive bias that squares particularly well with tabular data.

3. Background

Our OVAE model is created upon the building blocks of differentiable oblivious decision trees. We describe it in the following section.

3.1. Differentiable Oblivious Decision Trees (DODTs)

An oblivious decision tree (ODT) (Lou & Obukhov, 2017) is a full, binary decision tree whose internal nodes at the same level are restricted to have the same splitting feature and splitting threshold. An ODT is less expressive than a regular decision tree, but its low variance makes it an ideal weak learner for a gradient boosting algorithm. Such an algorithm typically decreases bias at the cost of increasing variance (Bauer & Kohavi, 1997; Ganjisaffar et al., 2011). Because the variance of its component weak learner (ODT) is very low to begin with, the boosting algorithm's overall increase in variance is controlled, and is more than offset by the performance gains from decreasing bias. Empirically, boosted ODTs (Lou & Obukhov, 2017; Prokhorenkova et al., 2018) provide state-of-the-art results compared to regular boosted decision/regression trees.

An ODT of depth d is equivalent to a table with 2^d entries, each corresponding to a particular combination of d feature splits. An ODT is completely specified by its splitting features $\mathbf{f} \in \mathbb{R}^d$, splitting thresholds $\mathbf{b} \in \mathbb{R}^d$, and a d-dimensional *response* tensor $\mathbf{R}$ (with 2^d entries) that maps the d decisions along a root-to-leaf path to the corresponding leaf value. Given an n-dimensional input $\mathbf{x} \in \mathbb{R}^n$, the output $h(\mathbf{x})$ of an ODT is $h(\mathbf{x}) = \mathbf{R}_{\mathbb{1}(\mathbf{f}_1(\mathbf{x})-\mathbf{b}_1),\ldots,\mathbb{1}(\mathbf{f}_d(\mathbf{x})-\mathbf{b}_d)}$ where $\mathbb{1}(\cdot)$ is the Heaviside function. In a *differentiable* ODT (DODT) (Popov et al., 2020), the output of an ODT is made differentiable so that it can be trained end-to-end via backpropagation. The splitting features $\mathbf{f}$ and Heaviside functions are replaced by their differentiable continuous approximations. Each splitting feature $\mathbf{f}_i(\mathbf{x})$ is now represented as a weighted sum of features $\hat{\mathbf{f}}_i(\mathbf{x})$, with the weights obtained via an α-entmax function (Peters et al., 2019) over a learnable feature selection vector $\mathbf{F}_i \in \mathbb{R}^n$, i.e., $\hat{\mathbf{f}}_i(\mathbf{x}) = \sum_{j=1}^n \mathbf{x}_j \cdot \mathrm{entmax}(\mathbf{F}_{i,j})$.The Heaviside function $\mathbb{1}(\mathbf{f}_i(\mathbf{x}) - \mathbf{b}_i)$ is replaced with a two-class entmax $\begin{bmatrix} c_i(\mathbf{x}) \\ 1 - c_i(\mathbf{x}) \end{bmatrix} = \mathrm{entmax}([\frac{\hat{\mathbf{f}}_i(\mathbf{x})-\mathbf{b}_i}{\tau_i}, 0])$ where τ_i is a learnable parameter to standardize the scales of the features. A *choice* tensor $\mathbf{C}$ of the same size as the response tensor $\mathbf{R}$ is obtained via an outer product over all c_i's and $(1-c_i)$'s , i.e., $\mathbf{C}(\mathbf{x}) = \begin{bmatrix} c_1(\mathbf{x}) \\ 1 - c_1(\mathbf{x}) \end{bmatrix} \otimes \begin{bmatrix} c_2(\mathbf{x}) \\ 1 - c_2(\mathbf{x}) \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} c_d(\mathbf{x}) \\ 1 - c_d(\mathbf{x}) \end{bmatrix}$ The scalar output $\hat{h}(\mathbf{x})$ of a DODT is computed as a sum over entries in the response vector $\mathbf{R}$ weighted by the corresponding values in $\mathbf{C}$, i.e.,

$$\hat{h}(\mathbf{x}) = \sum_{i_1,\ldots,i_d \in \{0,1\}^d} \mathbf{R}_{i_1,\ldots,i_d} \cdot \mathbf{C}_{i_1,\ldots,i_d}(\mathbf{x}).$$

The parameters of a DODT (i.e., $\mathbf{F}_i$, $\mathbf{b}_i$, τ_i, and $\mathbf{R}$) can be learned in an end-to-end fashion via stochastic gradient descent.To mimic the collection of ODTs in a boosting algorithm, DODTs can be ensembled together, with the output of one feeding into the input of another. Such an ensemble outperforms regular boosted decision/regression trees and deep neural networks in an extensive set of empirical comparisons (Popov et al., 2020).

4. Oblivious Variational Autoencoder (OVAE)

4.1. Input/Output Representation

Our proposed OVAE model assumes that data are contained in a table containing N_c continuous columns ($C_1, \ldots, C_{N_c}$) and N_d discrete columns ($D_1, \ldots, D_{N_d}$), and regards each column as a random variable. In the table, each j^{th} row $(c_{1,j}, \ldots, c_{N_c,j}, d_{1,j}, \ldots, d_{N_d,j})$ is assumed to be a sample generated from an underlying joint distribution $\mathbb{P}(C_{1:N_c}, D_{1:N_d})$. To deal with the potential multi-modality of each continuous variable, we preprocess its continuous values using *mode-specifc normalization* (MSN) (Xu et al., 2019). MSN first determines the number of modes in the distribution of each continuous variable C_i with variational Gaussian mixture models (Bishop, 2006), in which each mode m is associated with a normal distribution with mean η_m and standard deviation ψ_m. Next, for each value c_i of the continuous variable, MSN randomly samples a mode m from among the possible modes, and represents the selected mode with a one-hot encoding $\boldsymbol{\beta}_i$. MSN then "normalizes" the value c_i with respect to the chosen mode's normal distribution, i.e., $\alpha_i = \frac{c_i - \eta_m}{4\psi_m}$. Finally, each c_i is represented as the concatenation of α_i and $\boldsymbol{\beta}_i$. Each row in a table is thus represented as a $(2N_c{+}N_d)$-dimensional vector $\mathbf{r}_j = \alpha_{1,j} \oplus \boldsymbol{\beta}_{1,j} \oplus \cdots \oplus \alpha_{N_c,j} \oplus \boldsymbol{\beta}_{N_c,j} \oplus \mathbf{d}_{1,j} \oplus \cdots \oplus \mathbf{d}_{N_d,j}$, where $\oplus$ is the concatenation operator and $\mathbf{d}_{i,j}$ is a one-hot encoding. Both input and generated rows share the same representation.

4.2. OVAE Encoder

We construct OVAE's encoder by placing differentiable oblivious decision trees (DODTs) in parallel in a layer, and then stacking such DODT layers one on top of another. The outputs of the parallel trees in a layer are concatenated before being fed as input into another layer We include several parallel DODTs in a layer so that each DODT can capture a different way of partitioning its input data. This is particularly useful for rich datasets that can be partitioned in multiple valid ways because it allows the DODTs to fully capture the data's complexity. We stack one DODT layer upon another so that the latter can model the intricate dependencies among the different data partitionings in the former.

$\mathrm{DODTLayer}_{n\to k}(\mathbf{x})$ denotes a DODT layer that consists of k parallel DODTs, and it maps an n-dimensional input $\mathbf{x} \in \mathbb{R}^n$ to a k-dimensional output. The input $\mathbf{x}$ is fed into each of the k DODTs, and each DODT outputs a scalar value. The architecture for the encoder distribution $q_\phi(\mathbf{z}_j|\mathbf{r}_j)$ is as follows ($\mathbf{r}_j$ represents a row in a table with N_c continuous columns and N_d discrete columns as described in Section 4.1).

$$
\begin{aligned}
\mathbf{h}_1 &= \mathrm{DODTLayer}_{|\mathbf{r}_j|\to k}(\mathbf{r}_j) \\
\mathbf{h}_2 &= \mathrm{DODTLayer}_{k\to k}(\mathbf{h}_1) \\
\boldsymbol{\mu} &= \mathrm{DODTLayer}_{k\to k}(\mathbf{h}_2) \\
\boldsymbol{\sigma} &= \exp(\mathrm{DODTLayer}_{k\to k}(\mathbf{h}_2)) \\
q_\phi(\mathbf{z}_j|\mathbf{r}_j) &\sim \mathbb{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}))
\end{aligned}
$$

4.3. OVAE Decoder

OVAE's decoder has to model the $\alpha_{i,j}$ and $\boldsymbol{\beta}_{i,j}$ values for continuous columns, and the $\mathbf{d}_{i,j}$ values for discrete columns for each j^{th} row (the symbols are described in Section 4.1). OVAE assumes that each $\alpha_{i,j}$ has a normal distribution (with a column-specific variance δ_i), and that $\boldsymbol{\beta}_{i,j}$ and $\mathbf{d}_{i,j}$ each has a categorical probability mass function. The architecture for the decoder distribution $p_\theta(\mathbf{r}_j|\mathbf{z}_j)$ is as follows.

$$
\begin{aligned}
&\mathbf{h}_1 = \mathrm{DODTLayer}_{k\to k}(\mathbf{z}_j) \\
&\mathbf{h}_2 = \mathrm{DODTLayer}_{k\to k}(\mathbf{h}_1) \\
&\bar{\alpha}_{i,j} = \tanh(\mathrm{DODTLayer}_{k\to 1}(\mathbf{h}_2)) && \text{for } 1 \le i \le N_c \\
&\hat{\alpha}_{i,j} \sim \mathbb{N}(\bar{\alpha}_{i,j}, \delta_i) && \text{for } 1 \le i \le N_c \\
&\hat{\boldsymbol{\beta}}_{i,j} \sim \mathrm{softmax}(\mathrm{DODTLayer}_{k\to m_i}(\mathbf{h}_2)) && \text{for } 1 \le i \le N_c \\
&\hat{\mathbf{d}}_{i,j} \sim \mathrm{softmax}(\mathrm{DODTLayer}_{k\to |D_i|}(\mathbf{h}_2)) && \text{for } 1 \le i \le N_d
\end{aligned}
$$

$$
p_\theta(\mathbf{r}_j|\mathbf{z}_j) = \prod_{i=1}^{N_c} \mathbb{P}(\hat{\alpha}_{i,j} = \alpha_{i,j}) \prod_{i=1}^{N_c} \mathbb{P}(\hat{\boldsymbol{\beta}}_{i,j} = \boldsymbol{\beta}_{i,j}) \prod_{i=1}^{N_d} \mathbb{P}(\hat{\mathbf{d}}_{i,j} = \mathbf{d}_{i,j})
$$

m_i is the number of modes in continuous column i, and $|D_i|$ is the number of distinct discrete values in discrete column D_i. To obtain one-hot encodings from the $\hat{\boldsymbol{\beta}}_{i,j}$ and $\hat{\mathbf{d}}_{i,j}$ vectors, we simply set the maximum value in each vector to 1, and all other values to 0. The parameters of OVAE are the δ_i's and its constituents DODTs' parameters. These parameters are learned via stochastic gradient descent by maximizing the evidence lower bound (ELBO) (Kingma & Welling, 2014). Supplementary material consists figures of DODT and OVAE to better illustrate our approach.

5. Experiments

5.1. Datasets

For our experiments, we use 5 real-world regression datasets, 7 real-world classification datasets (12 tabular datasets in total). Two of the classification datasets `mnist28` and `mnist12` are respectively obtained by binarizing 28×28 and 12×12 MNIST images (LeCun & Cortes, 2010) into feature vectors (with an additional label column indicating the target digit).

Table 1. Results on classification datasets.

	adult F1	census F1	credit F1	cover. Macro-F1	intru. Macro-F1	mnist12/28 Micro-F1	mnist12/28 Micro-F1	Average F1
Identity	0.67	0.49	0.72	0.65	0.86	0.89	0.92	0.74
CLBN	0.33	0.31	0.41	0.32	0.38	0.74	0.18	0.38
PrivBayes	0.41	0.12	0.19	0.27	0.38	0.12	0.08	0.23
medGAN	0.38	0.00	0.00	0.09	0.30	0.09	0.10	0.14
VEEGAN	0.24	0.09	0.00	0.08	0.26	0.19	0.14	0.14
tableGAN	0.49	0.36	0.18	0.00	0.00	0.10	0.00	0.16
CTGAN	<u>0.60</u>	**0.39**	**0.67**	0.32	**0.53**	0.39	0.37	0.47
TVAE	**0.63**	<u>0.38</u>	0.10	<u>0.43</u>	<u>0.51</u>	<u>0.79</u>	<u>0.79</u>	<u>0.52</u>
OVAE	<u>0.60</u>	<u>0.38</u>	<u>0.51</u>	**0.45**	**0.53**	**0.83**	**0.84**	**0.59**

Table 2. Results on regression datasets.

	bike R^2	gpu R^2	wine R^2	power R^2	news R^2	Average R^2
Identity	0.86	0.75	0.28	0.88	0.14	0.58
CLBN	-49×10^1	-18	-27	-22×10^3	-6.3	-45×10^2
PrivBayes	0.71	-29×10^2	-10	**0.88**	-4.5	-58×10
medGAN	-50×10^{16}	-54×10^2	-42	-11.82	-8.80	-10×10^{16}
VEEGAN	-0.43	-16×10^2	-11	-0.67	-65×10^5	-13×10^5
tableGAN	0.74	-82×10^2	-0.10	0.77	-3.09	-16×10^2
CTGAN	<u>0.78</u>	0.60	<u>0.02</u>	-0.19	-0.43	0.16
TVAE	0.70	<u>0.68</u>	**0.22**	0.72	**−0.20**	<u>0.42</u>
OVAE	**0.82**	**0.70**	**0.22**	<u>0.81</u>	<u>−0.30</u>	**0.45**

5.2. Methodology

We compare our OVAE model against 7 other models (see Section 2). Two of these models are based on Bayesian networks: CLBN (Chow & Liu, 1968) and PrivBayes (Zhang et al., 2017). The remaining models are state-of-the-art deep learning ones: medGAN (Choi et al., 2017), VEEGAN (Srivastava et al., 2017), tableGAN (Park et al., 2018), CTGAN (Xu et al., 2019), and TVAE (Xu et al., 2019).

We follow the experimental methodology adopted by (Xu et al., 2019). For every real-world dataset, we train each model on the training tabular data $\mathbf{T}_{train}$, and use the trained model to generate synthetic tabular data $\mathbf{T}_{syn}$. We then train a set of standard regressors or classifiers (e.g., (boosted) regression/decision tree, linear regression, and multilayer perceptron) on $\mathbf{T}_{syn}$, and evaluate the set of regressors/classifiers on the test tabular data $\mathbf{T}_{test}$. We run this process thrice for each model per dataset, and report the average result of the set of regressors/classifiers. For regression tasks, we report the average R^2 score of a set of regressors on $\mathbf{T}_{test}$. For classification tasks, we report on metrics like F1, macro-F1, micro-F1 for a set of classifiers on $\mathbf{T}_{test}$ depending on the skew of the datasets. We also have an *Identity* system that simply copies $\mathbf{T}_{train}$, and treats it as $\mathbf{T}_{syn}$ in the aforementioned methodology (rather than doing the hard work of learning a model to generate $\mathbf{T}_{syn}$). The R^2, F1, and L_{test} scores associated with the Identity system serve as upper bounds for the scores of OVAE and its comparison models. All the hyperparameters of OVAE are in supplementary. The regression results are shown in Table 2 (best results are boldfaced; second best results are underlined). The numbers in the `news` are as reported in (Xu et al., 2019) (modulo the OVAE results). All other numbers are from our experiments. Our OVAE model outperforms TVAE on 3 real-world regression datasets (`bike`, `gpu`, and `power`), ties on one, and loses on another. The results give credence to our hypothesis that DODTs provide a useful inductive bias for improving tabular data generation. Note that where OVAE is not the best model (`power` and `news`), it is second best. On average, OVAE is the best performer on the real-world regression datasets.

The classification results are shown in Table 1. The numbers in the OVAE row are obtained from our experiments; all other numbers are as reported in (Xu et al., 2019). OVAE outperforms TVAE on 5 real-world classification datasets (`credit`, `covertype`, `intrusion`, `mnist12`, and `mnist28`), ties on one, and loses on another; on average, OVAE outperforms TVAE on the real-world classification datasets. Again, this shows that using DODTs in OVAE leads to better results vis-à-vis TVAE. Like on the real-world regression datasets, OVAE is consistently the best model (on `covertype`, `intrusion`, `mnist12`, and `mnist28`) or the second best performer (`adult`, `census`, and `credit`). In aggregate, OVAE is the best performing system.

6. Conclusion and Future Work

We presented OVAE, a new model for generating synthetic tabular data. OVAE combines differentiable oblivious decision trees (DODTs) with variational autoencoders (VAEs), thereby incorporating a strong inductive bias for tabular data into VAEs. Empirical comparisons with seven systems on 12 real-world datasets show the promise of our approach. Our proposed OVAE model advances the line of research that obviates privacy restrictions on sensitive medical tabular data by generating high-fidelity synthetic data. Our OVAE model can be used in healthcare to expedite the development of AI systems while preserving privacy of the patients. As future work, we want to incorporate domain knowledge as additional tabular constraints into OVAE.

7. Acknowledgements

This research is partly funded by an MOE AcRF Tier 1 grant (R -253-000-146-133) and an MOH NIC grant (MOH/NIC/CDM1/2018) to Stanley Kok.

References

Aviñó, L., Ruffini, M., and Gavaldà, R. Generating synthetic but plausible healthcare record datasets. *arXiv preprint arXiv:1807.01514*, 2018.

Bauer, E. and Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105–139, 1997.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Camino, R., Hammerschmidt, C., and State, R. Generating multi-categorical samples with generative adversarial networks. In *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

Che, Z., Cheng, Y., Zhai, S., Sun, Z., and Liu, Y. Boosting deep learning risk prediction with generative adversarial networks for electronic health records. *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 787–792, 2017.

Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W. F., and Sun, J. Generating multi-label discrete patient records using generative adversarial networks. In Doshi-Velez, F., Fackler, J., Kale, D., Ranganath, R., Wallace, B., and Wiens, J. (eds.), *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68 of *Proceedings of Machine Learning Research*, pp. 286–305, Boston, Massachusetts, 18–19 Aug 2017. PMLR. URL http://proceedings.mlr.press/v68/choi17a.html.

Chow, C. and Liu, C. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

Emam, K. E., Jonker, E., Arbuckle, L., and Malin, B. A systematic review of re-identification attacks on health data. *PLoS ONE*, 6, 2011.

Ganjisaffar, Y., Caruana, R., and Lopes, C. V. Bagging gradient-boosted trees for high precision, low variance ranking models. In *SIGIR*, 2011.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Guo, J., Lu, S., Cai, H., Zhang, W., Yu, Y., and Wang, J. Long text generation via adversarial training with leaked information. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

LeCun, Y. and Cortes, C. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 2010.

Lin, C. H., Chang, C.-C., Chen, Y.-S., Juan, D.-C., Wei, W., and Chen, H.-T. COCO-GAN: Generation by parts via conditional coordinating. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4511–4520, 2019.

Lou, Y. and Obukhov, M. Bdt: Gradient boosted decision tables for high accuracy and scoring efficiency. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

Ma, J. and Yarats, D. Quasi-hyperbolic momentum and ADAM for deep learning. *arXiv preprint arXiv:1810.06801*, 2018.

Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., and Kim, Y. Data synthesis based on generative adversarial networks. *ArXiv*, abs/1806.03384, 2018.

Patki, N., Wedge, R., and Veeramachaneni, K. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 399–410, 2016.

Peters, B., Niculae, V., and Martins, A. F. T. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1504–1519, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1146.

Popov, S., Morozov, S., and Babenko, A. Neural oblivious decision ensembles for deep learning on tabular data. In

International Conference on Learning Representations, 2020.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. In *Advances in neural information processing systems*, pp. 6638–6648, 2018.

Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., and Sutton, C. VEEGAN: Reducing mode collapse in GANs using implicit variational learning. In *Advances in Neural Information Processing Systems*, pp. 3308–3318, 2017.

Sun, Y., Cuesta-Infante, A., and Veeramachaneni, K. Learning vine copula models for synthetic data generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:5049–5057, 07 2019. doi: 10.1609/aaai.v33i01.33015049.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. Modeling tabular data using conditional GAN. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 7335–7345. Curran Associates, Inc., 2019.

Yahi, A., Vanguri, R., Elhadad, N., and Tatonetti, N. P. Generative adversarial networks for electronic health records: a framework for exploring and evaluating methods for predicting drug-induced laboratory test trajectories. *arXiv preprint arXiv:1712.00164*, 2017.

Zhang, J., Cormode, G., Procopiuc, C. M., Srivastava, D., and Xiao, X. PrivBayes: Private data release via bayesian networks. *ACM Trans. Database Syst.*, 42(4), October 2017. ISSN 0362-5915. doi: 10.1145/3134428. URL https://doi.org/10.1145/3134428.

8. Supplementary

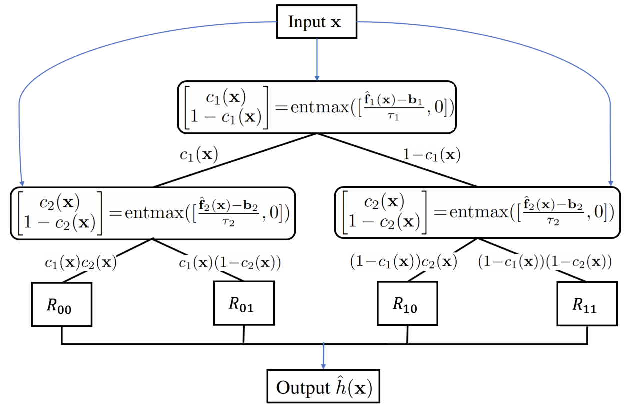

Figure 1. An example of a differentiable oblivious decision tree (DODT) of depth 2.

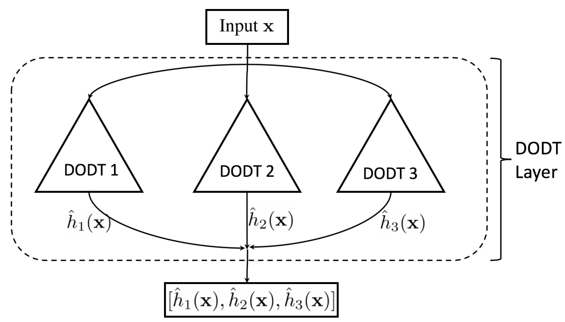

Figure 2. A DODT layer with 3 parallel DODTs (each triangle represents a DODT). The outputs of the DODTs are concatenated.

Hyperparameters : OVAE is trained with stochastic gradient descent using quasi-hyperbolic ADAM (Ma & Yarats, 2018) as the optimizer. In each DODT layer in the OVAE model, we use either 128, 256 or 512 parallel differentiable oblivious decision trees, (i.e., $k \in \{128, 256, 512\}$ in OVAE Encoder and OVAE Decoder). The depth of each DODT is set to 6. The parameters are chosen using preliminary experiments, and depend on whether the resultant models can fit into our GPU's memory (Nvidia Geforce RTX 2080 Ti; 11GB).