How Realistic Is Your Synthetic Data?
Constraining Deep Generative Models for Tabular Data

Mihaela Cătălina Stoian*, Salijona Dyrmishi*, Maxime Cordy, Thomas Lukasiewicz, Eleonora Giunchiglia





University of Oxford

University of Luxembourg

University of Luxembourg

Vienna University of Technology
University of Oxford

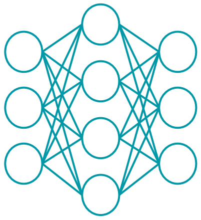
Vienna University of Technology

Why do we need constraints?

Neural networks are **data-driven** models, which do **not** account for **background knowledge**.

- **They can make predictions that violate the background knowledge.**

- **Neuro-Symbolic (NeSy) AI** aims at addressing this issue by interlinking neural networks with symbolic reasoning.



Deep Generative Model (DGM)

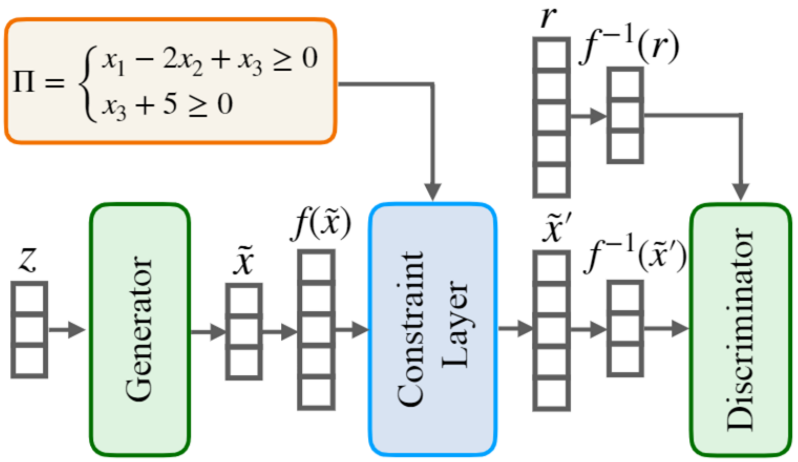
haemaglobin_max	haemaglobin_min
9.2	8.5
13.1	13.1
9.4	9.5

Constrained Deep Generative Models (C-DGM)

Background knowledge: expressed as linear inequalities capturing relations between continuous-valued the tabular data features.

Our approach allows for injecting background knowledge into DGMs by building a differentiable **Constraint Layer (CL)** into their architecture which:

- **guarantees** the **satisfaction** of the constraints
- guarantees a possibly **optimal** output that minimally changes the initial DGM predictions
- can be used during **training** and/or at **inference**.



Computing CL: a two-step process

Step 1: compute a set of constraints
for each variable in **reverse λ** order.

λ ordering: $x_1, x_2, \ldots, x_D$

Step 2: compute the corrected value
for each variable in **λ** ordering.

Example

$$\tilde{x}_1 = 7$$
$$\tilde{x}_2 = 3$$
$$\mathrm{CL}(\tilde{x})_1 = 7$$
$$\mathrm{CL}(\tilde{x})_2 = 5.1$$

Constraints Π

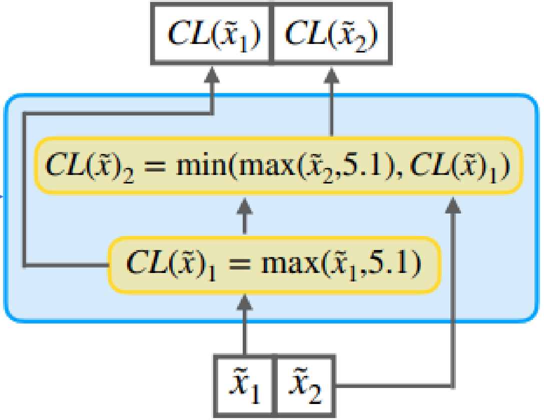
$$x_1 - x_2 \geq 0$$
$$x_2 - 5 > 0$$

$\epsilon = 0.1$

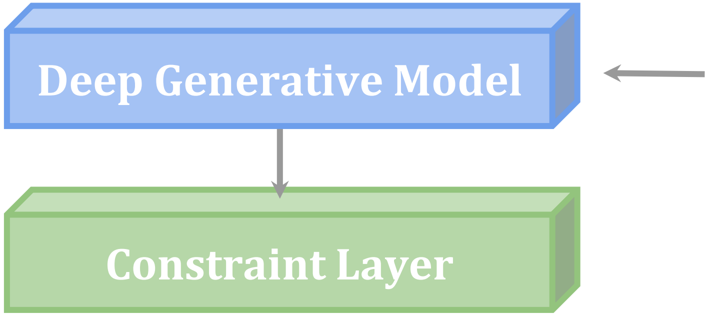
Step 1

$$\Pi_2 = \Pi$$
$$\Pi_1 = \{x_1 - 5 > 0\}$$

Step 2



Constraint Layer's Compatibility

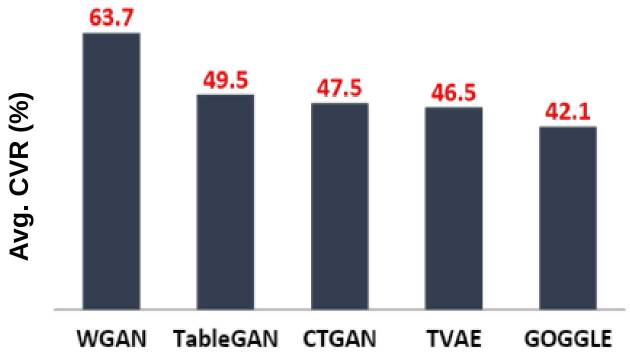


- **GAN architectures**
  - WGAN [1]
  - TableGAN [2]
  - CTGAN [3]
- **Variational Autoencoder architectures**
  - TVAE [3]
- **GNN architectures**
  - GOGGLE [4]
    (using a Message Passing Neural Network)
- **any other NN architecture…**

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. In Proc. of ICML, 2017.
[2] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim. Data synthesis based on generative adversarial networks. In Proc. of VLDB Endow., 2018.
[3] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni. Modeling tabular data using conditional GAN. In Proc. of NeurIPS, 2019.
[4] T. Liu, Z. Qian, J. Berrevoets, and M. van der Schaar. GOGGLE: Generative modelling for tabular data by learning relational structure. In Proc. of ICLR, 2022.

Standard DGMs do not satisfy requirements

- **CVR**: percentage of generated samples violating at least one constraint in the set of linear constraints.
- **Table**: CVR for 5 DGM types and 6 datasets.

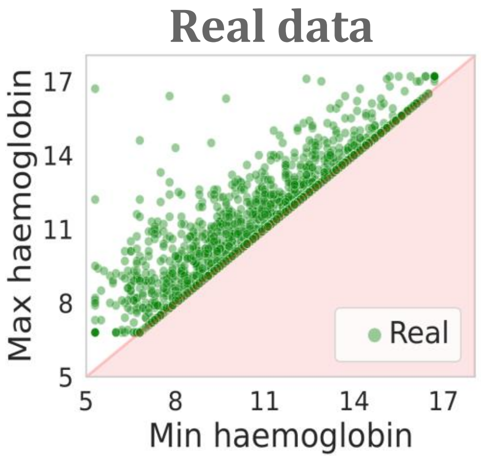


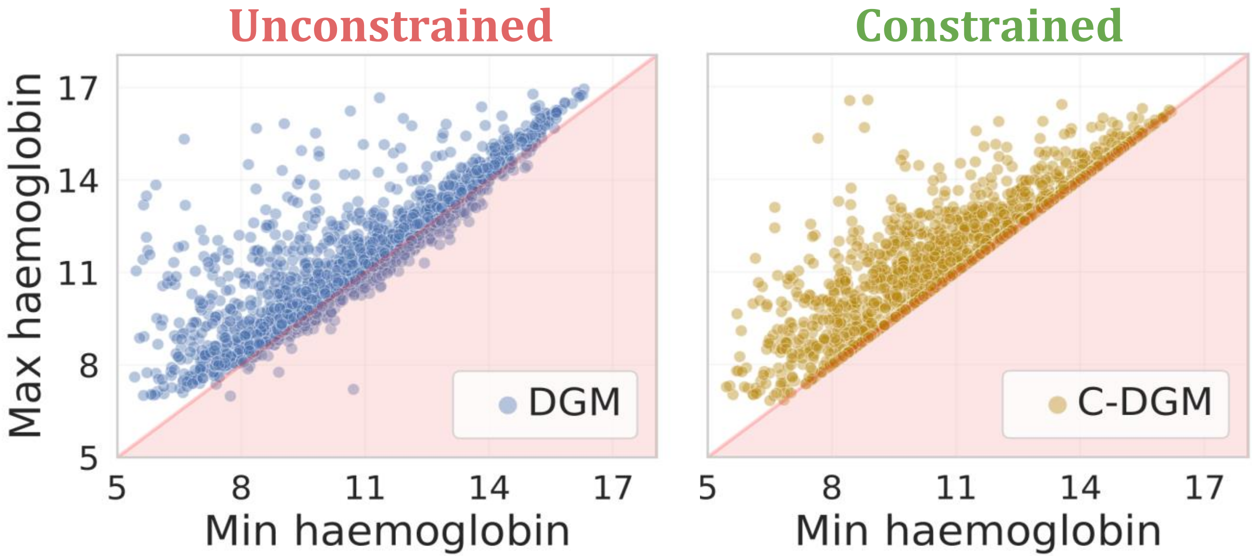
Model/Dataset	URL	WiDS	LCLD	Heloc	FSP	News
WGAN	11.1±1.6	98.2±0.2	100.0±0.0	57.0±13.0	70.7±8.3	45.6±9.6
TableGAN	4.9±1.4	96.4±2.4	6.1±0.9	45.6±16.3	71.6±8.7	72.6±5.3
CTGAN	3.1±2.6	99.9±0.0	11.8±2.7	41.6±12.1	74.3±5.2	54.3±10.1
TVAE	3.0±0.7	99.9±0.0	3.9±0.5	55.5±1.4	66.4±3.0	50.3±3.9
GOGGLE	5.9±6.6	78.2±11.6	13.1±2.9	47.3±7.0	63.7±17.6	44.8±7.2
All C-models	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0 ±0.0**	**0.0±0.0**

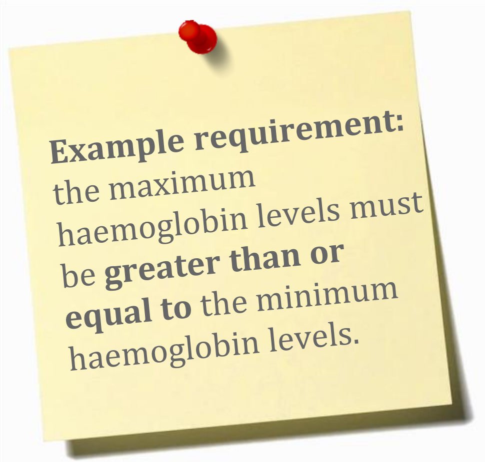
Standard DGMs often **violate constraints**, some exceeding **95%** **non-compliance!**

Qualitative performance

- The region violating the constraint is highlighted in **red**.
- The distribution of the samples generated by **C-DGM** matches more closely the one of the real data!







Performance

- Table: the average performance over 6 datasets.
- Two standard measure: **utility** and **detection**.
- For each measure, 3 metrics: F1, wF1, AUC; here we report F1 only.

	Utility (↑)	**Detection (↓)**
WGAN	0.463	0.945
C-WGAN	**0.483**	**0.915**
TableGAN	0.330	0.908
C-TableGAN	**0.375**	**0.898**
CTGAN	**0.517**	0.902
C-CTGAN	0.516	**0.894**
TVAE	0.497	0.869
C-TVAE	**0.507**	**0.868**
GOGGLE	0.344	0.926
C-GOGGLE	**0.409**	**0.925**

Background knowledge improves the synthetic data quality!

- **C-DGM models:** the DGM models equipped with CL.
- Our **C-DGMs outperform standard DGMs** in **9 out of 10 cases**.
- Often, the **differences are non-negligible**, e.g., 6.5% for GOGGLE according to utility-F1.

Sample generation time

The constrained layer introduces almost NO overhead to the sampling process!

- Table: the average result (in seconds) over 5 runs.
- 1000 samples were generated in each case.

	URL	WiDS	LCLD	Heloc	FSP	News
WGAN	0.02	0.03	0.01	0.00	0.00	0.01
C-WGAN	0.02	0.04	0.01	0.01	0.01	0.02
TableGAN	0.18	3.21	0.17	0.17	0.18	0.20
C-TableGAN	0.19	3.19	0.18	0.18	0.18	0.19
CTGAN	0.13	0.26	0.08	0.06	0.08	0.14
C-CTGAN	0.14	0.27	0.08	0.06	0.08	0.14
TVAE	0.12	0.27	0.06	0.06	0.06	0.12
C-TVAE	0.13	0.27	0.07	0.06	0.07	0.13
GOGGLE	0.71	3.99	9.91	0.16	0.06	2.01
C-GOGGLE	0.71	3.86	10.18	0.16	0.06	2.04

Out of 30 cases:

- **15 cases as fast as** the unconstrained DGMs.
- **14 cases at most 0.03s slower** than the unconstrained DGMs.
- **only one case 0.27s slower** than baseline!

Thank you for your attention!

Code available at https://github.com/mihaela-stoian/ConstrainedDGM

If you want to find out more, come see our poster at **ICLR 2024**!

Mihaela Cătălina Stoian*, Salijona Dyrmishi*, Maxime Cordy, Thomas Lukasiewicz, Eleonora Giunchiglia