Neural Generative Modeling of Order Statistics

Jean Pachebat
Centre de Mathématiques Appliquées (CMAP), CNRS, Ecole Polytechnique
Institut Polytechnique de Paris, Route de Saclay, 91128 Palaiseau Cedex, France
jean.pachebat@polytechnique.edu

Abstract

Order statistics (OS) and induced order statistics (IOS) arise whenever one sorts the components of a random vector, ubiquitous in portfolio construction and impact investing where assets are ranked by an impact factor and paired with returns. Standard generators ignore the combinatorial constraints of sorted data. We analyze neural OS generators through Schucany's and Sukhatme's probabilistic representations, which serve as theoretical devices for deriving approximation guarantees. We prove that ReLU networks with $\mathcal{O}(\varepsilon^{-2}\ln(\varepsilon^{-1}))$ parameters approximate the OS of N i.i.d. uniforms within $\mathcal{O}(\varepsilon \ln N)$ in expected L_1. On synthetic OS benchmarks, a simple MLP generator with global mean-scale normalization and an OS-aware penalization term improves sortedness while trading off distributional fidelity, outperforming vanilla baselines on ordering metrics.

1 Introduction

Generative modeling of sorted vectors is crucial when ranking is part of data generation. In sustainable and impact investing, assets are ranked by an impact factor (e.g., ESG scores) and paired with performance metrics (e.g., returns) [Lo and Zhang, 2021]. This yields data on the manifold of ordered vectors, yet common generators neither enforce nor exploit this structure, which can degrade downstream tasks that rely on ranking and ordering [Grover et al., 2019, Prillo and Eisenschlos, 2020].

We study neural generators for OS through classical probabilistic representations that make ordering constraints explicit. Using Sukhatmes construction concentrates approximation difficulty into a single scalar inverse-cdf block, leading to near-optimal parameter growth in ε^{-1}. In contrast, Schucanys multiplicative Markov recursion, while intuitive, requires learning N coupled bivariate updates and presents therefore higher complexity.

Beyond synthetic interest, the OS/IOS viewpoint is central to finance. Ranking assets by an impact attribute induces a joint distribution over ranks and returns (the concomitants of order statistics) that determines portfolio composition and performance. Lo and Zhang [2021] formalize how the correlation between an impact factor and residual returns controls whether impact tilts help or hurt mean-variance efficiency, while Lo et al. [2022] show that IOS with general copulas admit efficient optimization formulas. In both cases, realistic simulation depends on respecting the sorted manifold (for ranks) and its induced dependence with returns. Our theoretical insight on a structure-preserving generator opens the door to theory-driven mechanisms for stress-testing IOS-based policies and for evaluating how modeling choices affect dependence-sensitive metrics used by practitioners.

Contributions. (i) A theoretical analysis of neural OS generators via Schucany's and Sukhatme's probabilistic representations; (ii) capacity bounds showing ReLU networks with $\mathcal{O}(\varepsilon^{-2}\ln(\varepsilon^{-1}))$ parameters achieve expected L_1 error $\mathcal{O}(\varepsilon \ln N)$ for uniform OS; (iii) investigation into GAN performance on synthetic OS data with a mean-scale preprocessing and an OS-aware penalization term that improves sortedness with a trade-off against distributional fidelity metrics.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

2 Background

Let $X_1, \dots, X_N$ be i.i.d. with cdf F, and denote by $X_{1:N} \le \cdots \le X_{N:N}$ the OS. For any F, OS reduce to the uniform case by quantile mapping [David and Nagaraja, 2004, Ch. 2]:

$$\big(X_{1:N}, \dots, X_{N:N}\big) \overset{d}{=} \big(F^{-1}(U_{1:N}), \dots, F^{-1}(U_{N:N})\big), \quad U_i \ \ i.i.d.\ \mathrm{Unif}(0,1). \tag{1}$$

Hence it suffices to approximate the map from i.i.d. uniforms to their OS $(U_{1:N}, \dots, U_{N:N})$. Two classical representations guide us for designing the sampling algorithms:

Schucany recursion (uniforms). Order statistics of uniforms admit a top-down Markov property. Conditionally on $U_{m+1:N} = x$, the cdf of $U_{m:N}$ on $[0,x]$ is $\mathbb{P}(U_{m:N} \le y \mid U_{m+1:N} = x) = (y/x)^m$, since it is the maximum of m i.i.d. uniforms on $[0,x]$. Hence the scaled ratio $U_{m:N}/x \sim \mathrm{Unif}(0,1)$. Let $W_m \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}(0,1)$; then [Schucany, 1972]

$$U_{N:N} = W_N^{1/N}, \qquad U_{m:N} = U_{m+1:N}\, W_m^{1/m} \quad (m = N-1, \dots, 1). \tag{2}$$

This yields a recursive sampler for the full vector $(U_{1:N}, \dots, U_{N:N})$; see Algorithm 1 in the appendix for the exact and NN-approximate versions used in our analysis.

Sukhatme representation (exponentials). If $Y_1, \dots, Y_N$ are i.i.d. standard exponentials, the spacings of exponential OS are independent: with $Z_{0:N} = 0$ and $\Delta_i = Z_{i:N} - Z_{i-1:N}$, one has $\Delta_i \sim \mathrm{Exp}(N-i+1)$ and the Δ_i are independent. Therefore, writing $\Delta_i = Y_i/(N-i+1)$ with i.i.d. $Y_i \sim \mathrm{Exp}(1)$, we obtain

$$Z_{r:N} = \sum_{i=1}^{r} \frac{Y_i}{N-i+1}, \qquad U_{r:N} = 1 - e^{-Z_{r:N}}, \tag{3}$$

so generating exponential variables and forming weighted sums, followed by an exponential cdf, yields uniform OS [Sukhatme, 1937]. Using uniforms via the inverse cdf $F^{-1}(u) = -\ln(1-u)$ gives the practical sampler; see Algorithm 2 in the appendix.

Uniform OS facts. For $U_{1:N} \le \cdots \le U_{N:N}$ with i.i.d. $\mathrm{Unif}(0,1)$: (i) each marginal is Beta, $U_{k:N} \sim \mathrm{Beta}(k, N-k+1)$ with $\mathbb{E}[U_{k:N}] = \frac{k}{N+1}$ and $\mathrm{Var}(U_{k:N}) = \frac{k(N-k+1)}{(N+1)^2(N+2)}$; (ii) the spacings $S_0 = U_{1:N} - 0$, $S_i = U_{i+1:N} - U_{i:N}$ and $S_N = 1 - U_{N:N}$ are $\mathrm{Dirichlet}(1, \dots, 1)$, equivalently $S_i = E_i / \sum_{j=0}^{N} E_j$ with i.i.d. exponentials E_j; (iii) the joint density on the simplex $0 \le u_1 \le \cdots \le u_N \le 1$ is the constant $N!$.

3 Theory

We state guarantees for the two samplers referenced above; proofs are deferred to Appendix C.

Theorem 1 (Schucany-based generator)**.** *Let $\varepsilon > 0$. There exists a family of ReLU networks $\{G^{\mathrm{NN}}_{N-r:N}\}_{1 \le r \le N}$ used in Algorithm 1 such that, writing $\Delta V_r = |\hat{U}_{N-r:N} - U_{N-r:N}|$, one has $\mathbb{E}\,\Delta V_r < \varepsilon$, and the total number of neurons satisfies*

$$\mathcal{C}_{\mathrm{Schucany}}(\varepsilon) = \sum_{i=1}^{r} 3\Big(\big\lceil \big(\tfrac{2r(N-i)}{(N-r)\,\varepsilon}\big)^{N-i} \big\rceil + 1\Big) + c_1 \ln\Big(\tfrac{8r(N-i)}{(N-r)\,\varepsilon}\Big) + c_2, \tag{4}$$

for constants c_1, c_2. The complexity grows as a polynomial of degree N in ε^{-1}.

Theorem 2 (Sukhatme-based generator)**.** *With Algorithm 2, there exist ReLU networks F_{NN} and F^{-1}_{NN} such that, writing $\Delta U_{r:N} \coloneqq |\hat{U}_{r:N} - U_{r:N}|$,*

$$\mathbb{E}\,\Delta U_{r:N} \ \le\ \varepsilon\Big[1 + 2\sum_{i=1}^{r} \tfrac{1}{N-i+1}\Big], \tag{5}$$

i.e., the per-coordinate expected L_1 error scales with a harmonic factor, and the neuron count obeys, for some constant c:

$$\mathcal{C}_{\mathrm{Sukhatme}}(\varepsilon) \ \le\ c\,\varepsilon^{-2}\big(2\ln(\varepsilon^{-1}) + 1\big) + c\,\varepsilon^{-1}\ln(\varepsilon^{-1})\big(\ln(\varepsilon^{-1}) + \ln\ln(\varepsilon^{-1}) + 1\big), \tag{6}$$

with asymptotics $\mathcal{C}_{\mathrm{Sukhatme}}(\varepsilon) \sim 2c\,\varepsilon^{-2}\ln(\varepsilon^{-1})$ as $\varepsilon \to 0$. For $r = N$, the error behaves like $\varepsilon \ln N$.

Discussion and intuition. The key advantage of the Sukhatme-based methodology is structural: by mapping uniforms to exponentials and back, all nonlinear approximation is concentrated into two scalar functions, F and F^{-1}, reused across coordinates; the remaining computation is linear accumulation with harmonic weights. Truncating the exponential tail and applying Yarotsky-type bounds delivers a parameter count of order $\varepsilon^{-2}\ln(\varepsilon^{-1})$, and the accumulated error across coordinates grows with the harmonic series, yielding the $\varepsilon \ln N$ dependence. By contrast, Schucanys recursion composes N bivariate maps and an approximate multiplication, so errors compound more severely and capacity scales polynomially, of degree N, in ε^{-1} with degree tied to dimension. Intuitively, concentrating difficulty into a single inverse-cdf block avoids learning many near-singular power maps $u^{1/m}$ and stabilizes training.

4 Experiments

Practical setup. We generate synthetic OS in dimensions $d \in \{2, 3, 5, 10, 20, 50\}$ from the uniform distribution, which are the d largest components of a vector size $N = 1,000$ and train standard MLP GAN/WGAN baselines [Goodfellow et al., 2014, Arjovsky et al., 2017]. We do not instantiate network architectures that hard-code the Schucany or Sukhatme constructions from Algorithms 1-2; those serve as *theoretical* devices to derive approximation guarantees. In practice, the generator and discriminator are conventional MLPs trained on sorted data, with normalization and (optionally) an order-statistics penalty.

Normalization scheme. To preserve the geometry of OS during training, we apply a *global mean-scale* normalization to all coordinates: $\tilde{\boldsymbol{x}} = s(\boldsymbol{x} - \bar{x})$ with $\bar{x}$ the dataset-wide mean. This stabilizes optimization and leaves pairwise ordering unchanged, unlike per-coordinate min-max which can break sortedness.

Metrics. We report six metrics grouped by what they measure: *Distributional fidelity*–(i) **W1D**: mean 1-Wasserstein over marginals (sort both samples per coordinate and average absolute differences); (ii) **SWD**: sliced 1-Wasserstein over random projections (100 slices). *Dependence structure*–(iii) **Spearman**: average absolute difference of pairwise Spearman correlations between real and generated samples; (iv) **AKE**: Absolute Kendall Error, mean absolute difference between descending-sorted samples. *Ordering structure*–(v) **Sortedness**: proportion of generated samples that are fully monotone (descending); (vi) **Softsortedness**: average soft penalty for local order violations (lower is better).

Sweeps and selection. We tune hyperparameters with Optuna [Akiba et al., 2019] using Sobol initialization and early stopping. Across dimensions and model families (GAN/WGAN/WGAN_OSP), we sweep (log-uniform) learning rates, generator width/depth, batch size, discriminator steps, and (for WGAN) gradient-penalty weight; for OSP (8) we place a log-uniform prior over $\lambda_{\mathrm{OSP}} \in [10^{-4}, 10^{-2}]$. Trials are evaluated on a validation split using W1D; we intentionally do not select on Sortedness. We report metrics on the held-out test set from the best validation checkpoint, averaged over 10 seeds.

Order-statistics penalty ($\mathcal{L}_{\mathrm{OSP}}$). To encourage the generator to emit sorted vectors, we add a differentiable penalty on local order violations:

$$\mathcal{L}_{\mathrm{OSP}}(G) \;=\; \mathbb{E}_{\boldsymbol{z}}\left[\frac{1}{d-1}\sum_{j=1}^{d-1}\Big(\max\{\, G(\boldsymbol{z})_j - G(\boldsymbol{z})_{j+1} + \varepsilon \,,\, 0\,\}\Big)^2\right], \tag{7}$$

with small margin $\varepsilon > 0$ (we use $\varepsilon = 10^{-4}$). The generator objective becomes

$$\mathcal{L}_G^{\text{total}} \;=\; \mathcal{L}_G^{\text{adv}} \;+ \lambda_{\mathrm{OSP}}\, \mathcal{L}_{\mathrm{OSP}}(G), \tag{8}$$

where $\mathcal{L}_G^{\text{adv}}$ is the GAN or Wasserstein WGAN generator loss, and λ_{OSP} is tuned. This penalty is zero iff outputs are non-increasing coordinate-wise and increases quadratically with the size of violations.

Findings. Adding $\mathcal{L}_{\mathrm{OSP}}$ further increases Softsortedness/Sortedness in $d \in \{2, 10, 20\}$ with minimal impact on W1D/SWD. In particular, WGAN + OSP achieves the best structure-aware scores while keeping Wasserstein metrics close to the unregularized WGAN; at low dimensions, vanilla GAN can remain competitive on dependence metrics.

Additional qualitative pairplots for $d \in \{2, 3, 5\}$ and $d = 10$ are provided in Appendix D (Figures 1 and 2).

W1D

dim	GAN	WGAN	WGAN_OSP
2	0.341	**0.328**	0.486
5	**0.22**	0.424	0.251
10	0.513	**0.363**	0.488
20	1.1	**0.55**	0.839
50	2.33	1.13	**1.1**

SWD

dim	GAN	WGAN	WGAN_OSP
2	0.336	**0.332**	0.475
5	**0.204**	0.396	0.251
10	0.473	**0.356**	0.478
20	1.03	**0.526**	0.815
50	2.16	1.1	**1.02**

Spearman

dim	GAN	WGAN	WGAN_OSP
2	1.53	1.18	**1.17**
5	**0.315**	0.506	0.896
10	**0.452**	0.989	0.93
20	**0.391**	0.935	1.33
50	**0.713**	2.02	1.65

AKE

dim	GAN	WGAN	WGAN_OSP
2	**1.79**	1.84	1.92
5	**2.64**	2.71	2.67
10	3.59	**3.56**	3.66
20	4.95	5.05	**4.93**
50	**7.94**	8.08	7.95

Sortedness

dim	GAN	WGAN	WGAN_OSP
2	94.8%	88.7%	**98.3%**
5	**89.2%**	76.6%	80.3%
10	56.3%	56.2%	**99.4%**
20	4.0%	60.0%	**93.5%**
50	0.0%	**88.3%**	74.9%

Softsortedness

dim	GAN	WGAN	WGAN_OSP
2	0.128	0.364	**0.019**
5	**0.033**	0.19	0.193
10	0.263	0.28	**0.000692**
20	1.04	0.105	**0.0102**
50	3.21	**0.0807**	0.0832

Table 1: Metric mosaic across dimensions and models (bold = best per row). W1D/SWD assess distributional fidelity; Spearman/AKE capture dependence and rank-structure fidelity; Sortedness/Softsortedness measure compliance with the ordered structure. The OSP penalty improves sortedness metrics at the cost of a trade-off with some fidelity metrics.

5 Discussion

Related work. Classical results on order statistics and concomitants underpin our approach [David and Nagaraja, 2004, David, Herbert A, 1973]. In finance, IOS formalize impact-ranked portfolios [Lo and Zhang, 2021, Lo et al., 2022], linking ranks to returns via copulas. Our generator targets the OS manifold directly, leveraging exponential representations [Sukhatme, 1937] to obtain neural constructions with capacity guarantees. Differentiable sorting relaxations such as Grover et al. [2019], Prillo and Eisenschlos [2020] also study learning with ranks.

Limitations. Our experiments address OS on synthetic data; we do not yet handle IOS, i.e., the joint distribution of ranks and returns. The OSP term encourages intra-vector monotonicity but does not capture cross-variable dependence needed for IOS. Applying the method to real impact-investing datasets requires modeling factor-return dependence (e.g., learned copulas) and additional preprocessing. The hyperparameter λ_{OSP} controls a trade-off: higher values improve sortedness metrics but can degrade distributional fidelity (W1D/SWD), as evidenced in Table 1. Practitioners must calibrate this parameter based on whether structural compliance or distributional accuracy is more critical for their application. While our metrics are averaged over 10 random seeds, we report point estimates without standard deviations or confidence intervals in Table 1; including uncertainty quantification would strengthen the statistical validity of our comparisons.

Conclusion and outlook. We presented a theory-guided generator for OS, a practical normalization that preserves geometry and a penalization term that encourages intra-vector monotonicity. Future work includes IOS and applications to real data. A second direction is empirical validation of the Schucany- and Sukhatme-based bounds (e.g., $\varepsilon \ln N$ error growth; $\varepsilon^{-2} \ln(\varepsilon^{-1})$ parameters) using controlled architectures that mirror the theoretical constructions, feasible but currently tedious at the granularity required.

Acknowledgments

This work benefited from the support of the Chair Stress Test, Risk Management and Financial Steering, led by the French Ecole polytechnique and its foundation and sponsored by BNP Paribas.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework, July 2019.

Michaël Allouche, Stéphane Girard, and Emmanuel Gobet. Learning extreme expected shortfall and conditional tail moments with neural networks. Application to cryptocurrency data. *Neural Networks*, 182:106903, February 2025. ISSN 08936080. doi: 10.1016/j.neunet.2024.106903. URL `https://linkinghub.elsevier.com/retrieve/pii/S0893608024008323`.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, August 2017. URL `https://proceedings.mlr.press/v70/arjovsky17a.html`.

H.A. David and H.N. Nagaraja. *Order Statistics*. Wiley Series in Probability and Statistics. Wiley, 2004. ISBN 978-0-471-65401-8. URL `https://books.google.fr/books?id=bdhzFXg6xFkC`.

David, Herbert A. Concomitants of order statistics. *Bull. Inst. Internat. Statist.*, 45(1):295–300, 1973.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, 2014. URL `https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf`.

Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. Stochastic optimization of sorting networks via continuous relaxations. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=H1eSS3CcKX`.

Andrew W Lo and Ruixun Zhang. Quantifying the Impact of Impact Investing. *Management Science*, October 2021. doi: http://dx.doi.org/10.2139/ssrn.3944367.

Andrew W. Lo, Lan Wu, Ruixun Zhang, and Chaoyi Zhao. Optimal Impact Portfolios with General Dependence and Marginals. *SSRN Electronic Journal*, 2022. ISSN 1556-5068. doi: 10.2139/ssrn.4177277. URL `https://www.ssrn.com/abstract=4177277`.

Sebastian Prillo and Julian Eisenschlos. SoftSort: A continuous relaxation for the argsort operator. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7793–7802. PMLR, July 2020. URL `https://proceedings.mlr.press/v119/prillo20a.html`.

W. R. Schucany. Order statistics in simulation. *Journal of Statistical Computation and Simulation*, 1 (3):281–286, July 1972. ISSN 0094-9655, 1563-5163. doi: 10.1080/00949657208810022. URL `http://www.tandfonline.com/doi/abs/10.1080/00949657208810022`.

P. V. Sukhatme. Tests of significance for samples of the χ2-population with two degrees of freedom. *Annals of Eugenics*, 8(1):52–56, October 1937. ISSN 2050-1420, 2050-1439. doi: 10.1111/j.1469-1809.1937.tb02159.x. URL `https://onlinelibrary.wiley.com/doi/10.1111/j.1469-1809.1937.tb02159.x`.

Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, October 2017. ISSN 08936080. doi: 10.1016/j.neunet.2017.07.002. URL `https://linkinghub.elsevier.com/retrieve/pii/S0893608017301545`.

A Algorithms (Exact and NN-Approximate)

Algorithm 1: Sampling from order statistics $U_{1:N} \le \cdots \le U_{N:N}$ (Schucany recursion)

Input : Learned functions $\hat{G}^{NN}_{r:N} : [0,1] \to \mathbb{R}$ for $r \in [N]$; N uniform samples $U_1, \ldots, U_N$
Output : $\hat{U}_{1:N}, \ldots, \hat{U}_{N:N}$
Set $\hat{U}_{N:N} = U_N^{1/N}$;
for $r \leftarrow 1$ **to** N **do**
  $\hat{U}_{N-r:N} \leftarrow \hat{G}^{NN}_{N-r:N}(\hat{U}_{N-r+1:N}, U_r)$;
  `// Exact:` $U_{N-r:N} \leftarrow U_{N-r+1:N} U_r^{1/(N-r)}$.
return $\hat{U}_{r:N}$ *and* $U_{r:N}$*, for* $1 \le r \le N$.

Algorithm 2: Sampling from order statistics $U_{1:N} \le \cdots \le U_{N:N}$ based on (3)

Input : ReLU networks F_{NN} and F^{-1}_{NN}; N uniform samples $U_1, \ldots, U_N$
Output : $\hat{Z}_{1:N}, \ldots, \hat{Z}_{N:N}$
Set $\hat{U}_{0:N} = U_{0:N} = 0$;
for $r \leftarrow 1$ **to** N **do**
  $\hat{Z}_{r:N} \leftarrow \hat{Z}_{r-1:N} + \frac{F^{-1}_{\mathrm{NN}}(U_r)}{N-r}$;
  `// Exact:` $Z_{r+1:N} \leftarrow Z_{r:N} + \frac{F^{-1}(U_r)}{N-r}$.
return $\hat{U}_{r:N} = F_{\mathrm{NN}}(\hat{Z}_{r:N})$ *and* $U_{r:N} = F(Z_{r:N})$*, for* $1 \le r \le N$.

B Yarotsky's Approximation Results

We recall the Sobolev norm $\|f\|_{\mathcal{W}^{n,\infty}([0,1]^d)} = \max_{\mathbf{n}:|\mathbf{n}|\le n} \operatorname{ess\,sup}_{\boldsymbol{x}\in[0,1]^d} |D^{\mathbf{n}} f(\boldsymbol{x})|$ and the unit class $F_{n,d} = \{f : \|f\|_{\mathcal{W}^{n,\infty}} \le 1\}$.

Lemma 1 (Yarotsky [2017], Prop. 3)**.** *For any $M > 0$ and $\varepsilon \in (0,1)$, there exists a ReLU network implementing $\tilde{\times} : \mathbb{R}^2 \to \mathbb{R}$ such that: if $|x|, |y| < M$ then $|\tilde{\times}(x,y) - xy| \le \varepsilon$; if $x = 0$ or $y = 0$, $\tilde{\times}(x,y) = 0$. The depth and size are at most $c_1 \ln(1/\varepsilon) + c_2(M)$.*

Theorem 3 (Yarotsky [2017], Thm. 1)**.** *For any integers d, n and $\varepsilon \in (0,1)$, there exists a ReLU network of depth $\mathcal{O}(\ln(1/\varepsilon + 1))$ and size $\mathcal{O}(\varepsilon^{-d/n}(\ln(1/\varepsilon) + 1))$ that approximates any $f \in F_{n,d}$ within ε in sup-norm.*

C Proofs

C.1 Proof of Theorem 1

Let

$$\Delta V_r \;:=\; \hat{U}_{N-r:N} - U_{N-r:N}. \tag{9}$$

From the recursion and the triangle inequality,

$$|\Delta V_r| \;\le\; |\Delta V_{r-1}|\, U_r^{1/(N-r)} + \varepsilon_r, \tag{10}$$

$$\varepsilon_r \;:=\; \sup_{u_1,u_2\in[0,1]} \left|\hat{G}^{NN}_{N-r:N}(u_1,u_2) - G_{N-r:N}(u_1,u_2)\right|. \tag{11}$$

By induction,

$$|\Delta V_r| \;\le\; \sum_{i=0}^{r} \varepsilon_i \prod_{j=i+1}^{r} U_j^{1/(N-j)}. \tag{12}$$

Taking expectations and using $\mathbb{E}[U^{1/(N-j)}] = (N-j)/(N-j+1)$ yields

$$\mathbb{E}\,|\Delta V_r| \;\le\; \sum_{i=0}^{r} \varepsilon_i \, \frac{N-r}{N-i}. \tag{13}$$

It remains to bound the terms ε_i. Write

$$G_{N-r:N}(u_1,u_2) \;=\; u_1\,\phi_r(u_2), \qquad \phi_r(u) \;=\; u^{\,1/(N-r)} \in C^{0,\zeta_r}([0,1]), \quad \zeta_r = \tfrac{1}{N-r}. \tag{14}$$

Approximate ϕ_r to accuracy $\varepsilon/2$ by a piecewise-linear ReLU network (triangular basis; cf. Allouche et al. [2025], Lem. 8), using

$$3\big(\,\lceil (2/\varepsilon)^{\,N-r}\rceil + 1\,\big) \text{ neurons.} \tag{15}$$

Approximate multiplication by $\tilde{\times}$ with accuracy $\varepsilon/8$, whose size is $c_1\ln(8/\varepsilon)+c_2$ by Lemma 1. Combining errors gives

$$\sup_{u_1,u_2\in[0,1]} \big|\hat{G}^{NN}_{N-r:N}(u_1,u_2) - G_{N-r:N}(u_1,u_2)\big| \;<\; \varepsilon. \tag{16}$$

Choosing

$$\varepsilon_i \;=\; \frac{(N-r)}{r\,(N-i)}\,\varepsilon \tag{17}$$

in (13) yields $\mathbb{E}\,|\Delta V_r| \;\le\; \varepsilon$. Summing the network sizes over i gives the stated complexity in Theorem 1.

C.2 Proof of Theorem 2

Define the truncated inverse cdf

$$\tilde{F}_\varepsilon^{-1}(u) \;=\; \begin{cases} -\ln(1-u), & 0 \le u \le 1-\varepsilon, \\ \ln(1/\varepsilon), & 1-\varepsilon < u \le 1. \end{cases} \tag{18}$$

Then $\varepsilon\,\tilde{F}_\varepsilon^{-1} \in F_{1,1}$. By Theorem 3, there exists a ReLU network η_{NN} with size $\mathcal{O}(\varepsilon^{-2}(2\ln(1/\varepsilon)+1))$ such that, on $[0,1-\varepsilon]$,

$$\big\|\,\varepsilon\,\tilde{F}_\varepsilon^{-1} - \eta_{NN}\,\big\|_\infty \;\le\; \varepsilon^2. \tag{19}$$

Hence

$$\big\|\,\tilde{F}_\varepsilon^{-1} - \eta_{NN}/\varepsilon\,\big\|_\infty \;\le\; \varepsilon \quad \text{on } [0,1-\varepsilon], \tag{20}$$

and we extend η_{NN} by setting its value to $\ln(1/\varepsilon)$ on $[1-\varepsilon,1]$. Splitting the integral shows

$$\mathbb{E}\,\big|F^{-1}(U) - \eta_{NN}(U)\big| \;\le\; 2\varepsilon. \tag{21}$$

Let $\hat{Y}_i = \eta_{NN}(U_i)$ and $\hat{Z}_{r:N} = \sum_{i=1}^{r} \hat{Y}_i/(N-i+1)$. Then

$$\mathbb{E}\,\big|Z_{r:N} - \hat{Z}_{r:N}\big| \;\le\; 2\varepsilon \sum_{i=1}^{r} \frac{1}{N-i+1}. \tag{22}$$

For the forward map, define

$$\tilde{F}_\varepsilon(x) \;=\; \frac{F\big(\ln(1/\varepsilon)\,x\big)}{\ln(1/\varepsilon)} \in F_{1,1}. \tag{23}$$

By Theorem 3, there exists a NN approximating $\tilde{F}_\varepsilon$ with error $\varepsilon/\ln(1/\varepsilon)$ using size $\mathcal{O}(\varepsilon^{-1}\ln(1/\varepsilon)(\ln(1/\varepsilon)+\ln\ln(1/\varepsilon)+1))$. Undoing the rescaling gives

$$\|F - F_{\mathrm{NN}}\|_\infty \;\le\; \varepsilon \quad \text{on } [0,\ln(1/\varepsilon)]. \tag{24}$$

Using the 1-Lipschitzness of F and the bound above yields

$$\mathbb{E}\,\big|U_{r:N} - \hat{U}_{r:N}\big| \;\le\; \varepsilon \;+2\varepsilon \sum_{i=1}^{r} \frac{1}{N-i+1}, \tag{25}$$

which is the claimed inequality. The total neuron count is the sum of the forward and inverse approximators, giving the stated complexity and its asymptotics.

D Additional Figures

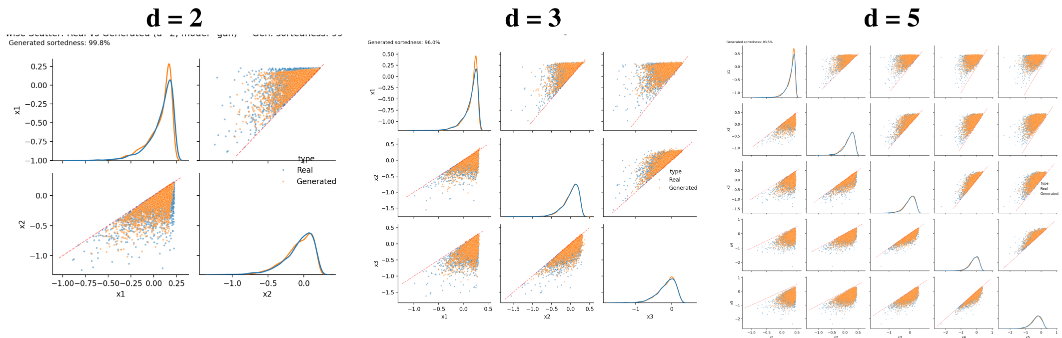

Figure 1: Pairplot mosaic for dimensions d = 2, 3, 5 (GAN).

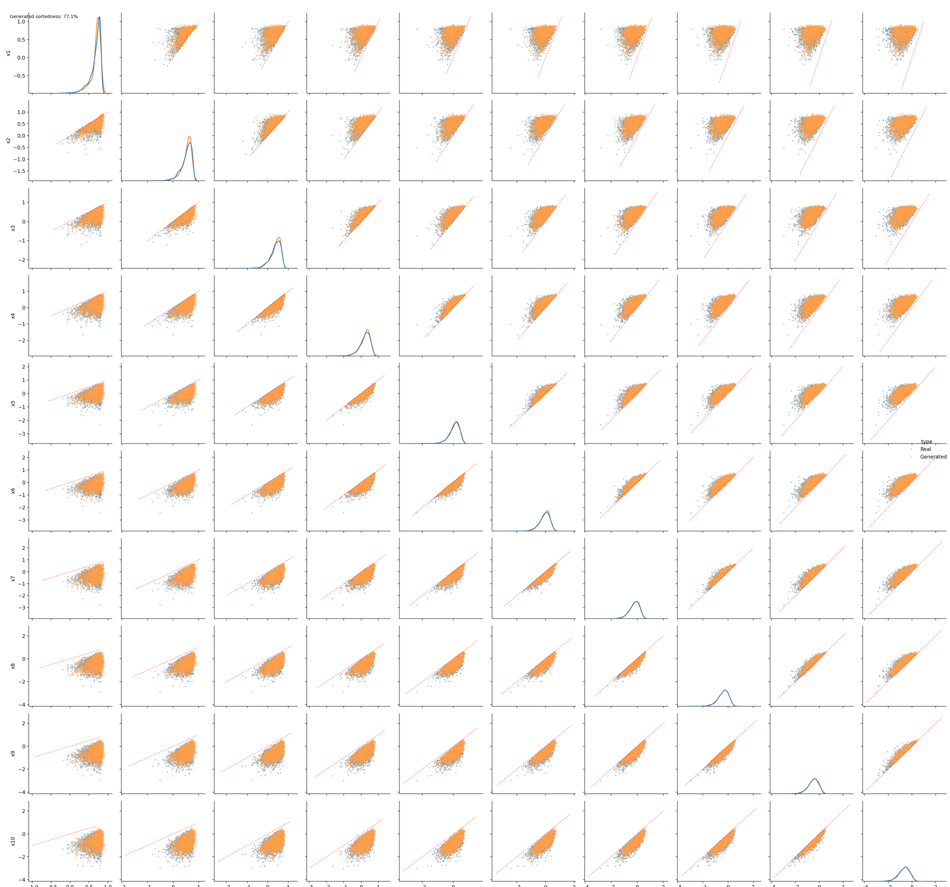

Figure 2: Pairplot for dimension d = 10 (GAN).