Machine Learning in Finance Workshop 2021

Generative Adversarial Networks and their applications in Finance

Joerg Osterrieder

Zurich University of Applied Sciences, University of Twente

Columbia University Data Science Institute

Columbia Engineering The Fu Foundation School of Engineering and Applied Science

Bloomberg

Joint work with

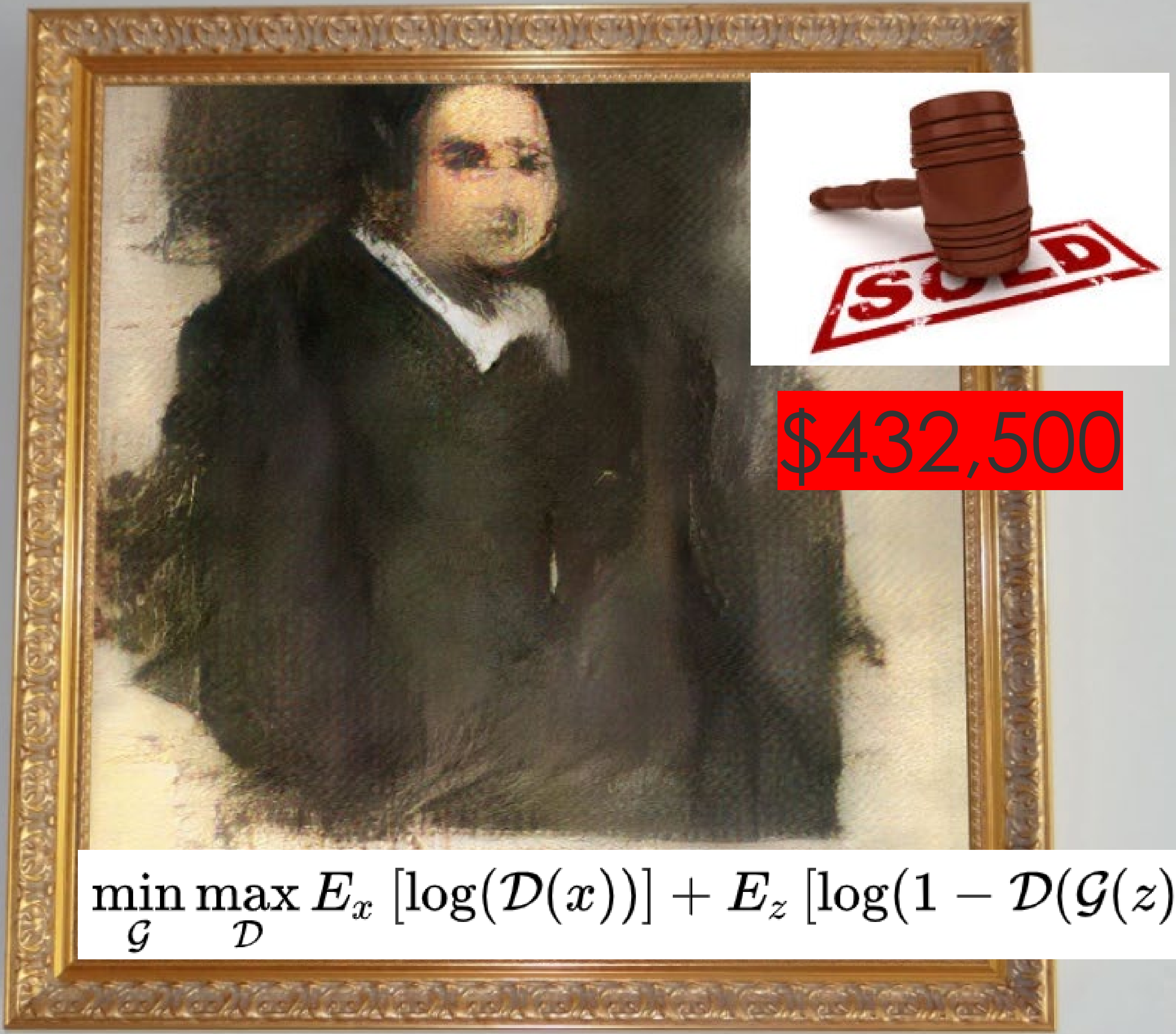
- Prof. Dr. Ali Hirsa, Columbia University
- Weilong Fu, Columbia University

- Dr. Branka Hadji Misheva, Zurich University of Applied Sciences
- Chenxin Nie, University of Zurich

This presentation is purely for educational purposes, not all references might be fully cited.

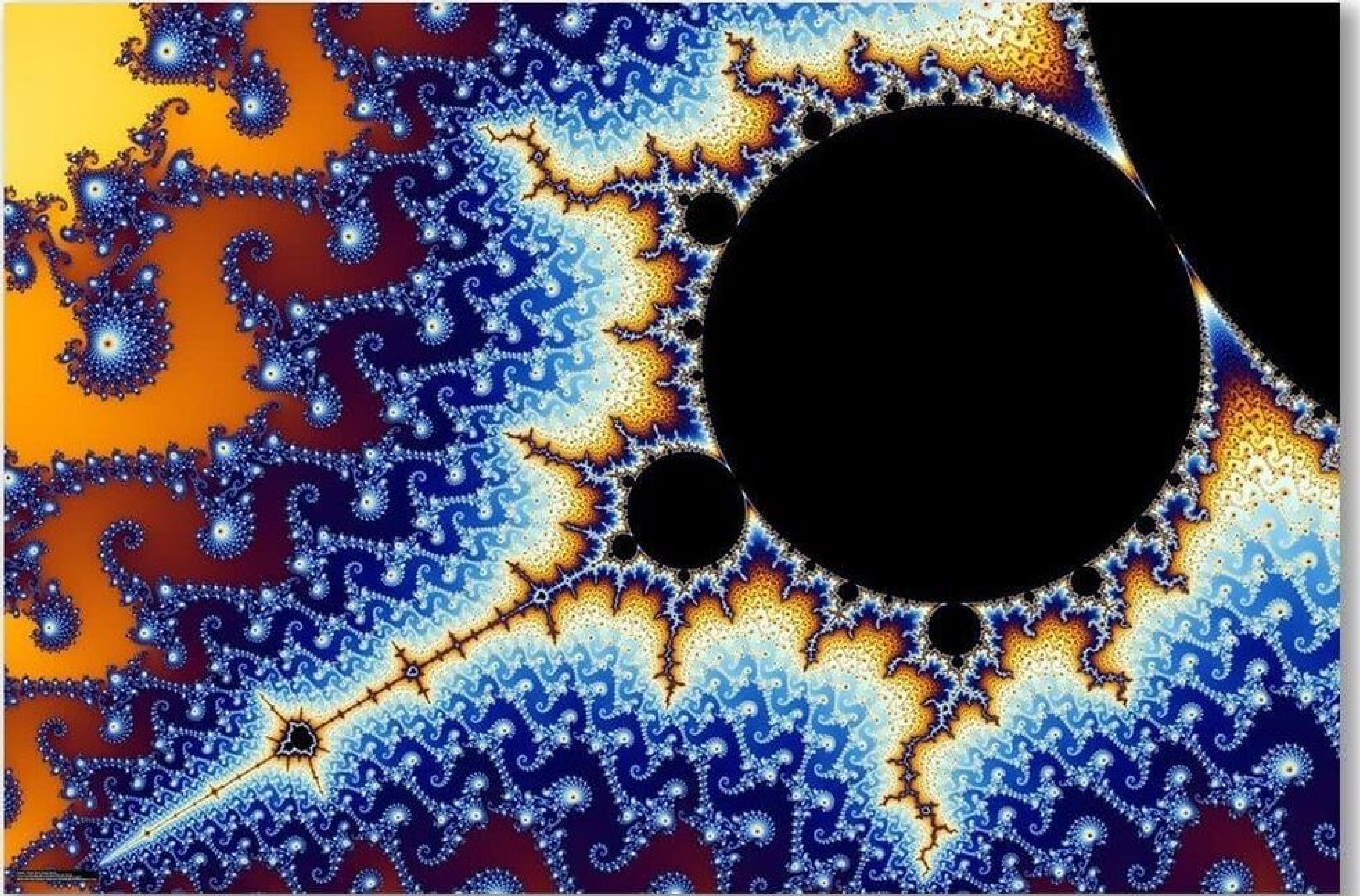

Op, 24 “I am AI”
AIVA (Artificial Intelligence Virtual Artist)
Aiva Technologies

https://towardsdatascience.com/how-to-train-stylegan-to-generate-realistic-faces-d4afca48e705

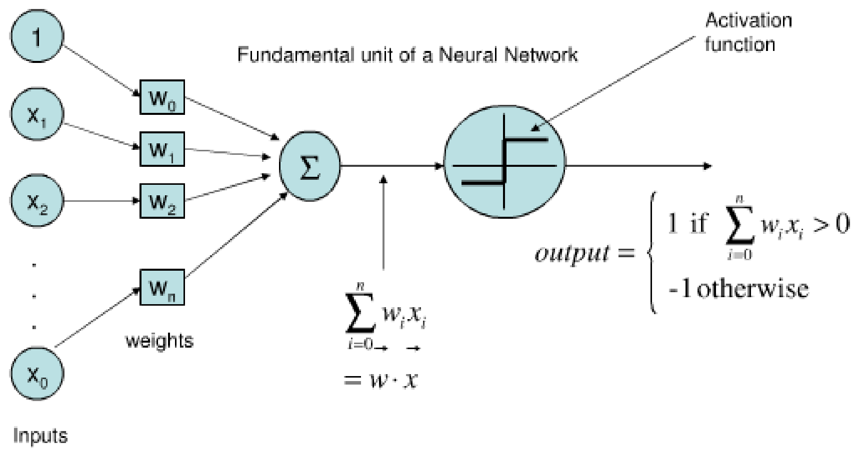


$432,500

$$\min_{\mathcal{G}} \max_{\mathcal{D}} E_x \left[\log(\mathcal{D}(x))\right] + E_z \left[\log(1 - \mathcal{D}(\mathcal{G}(z)))\right]$$



A detour --- Neural networks



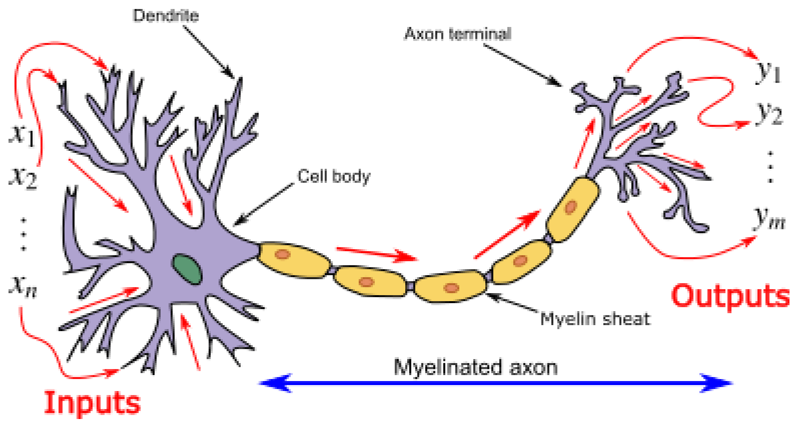
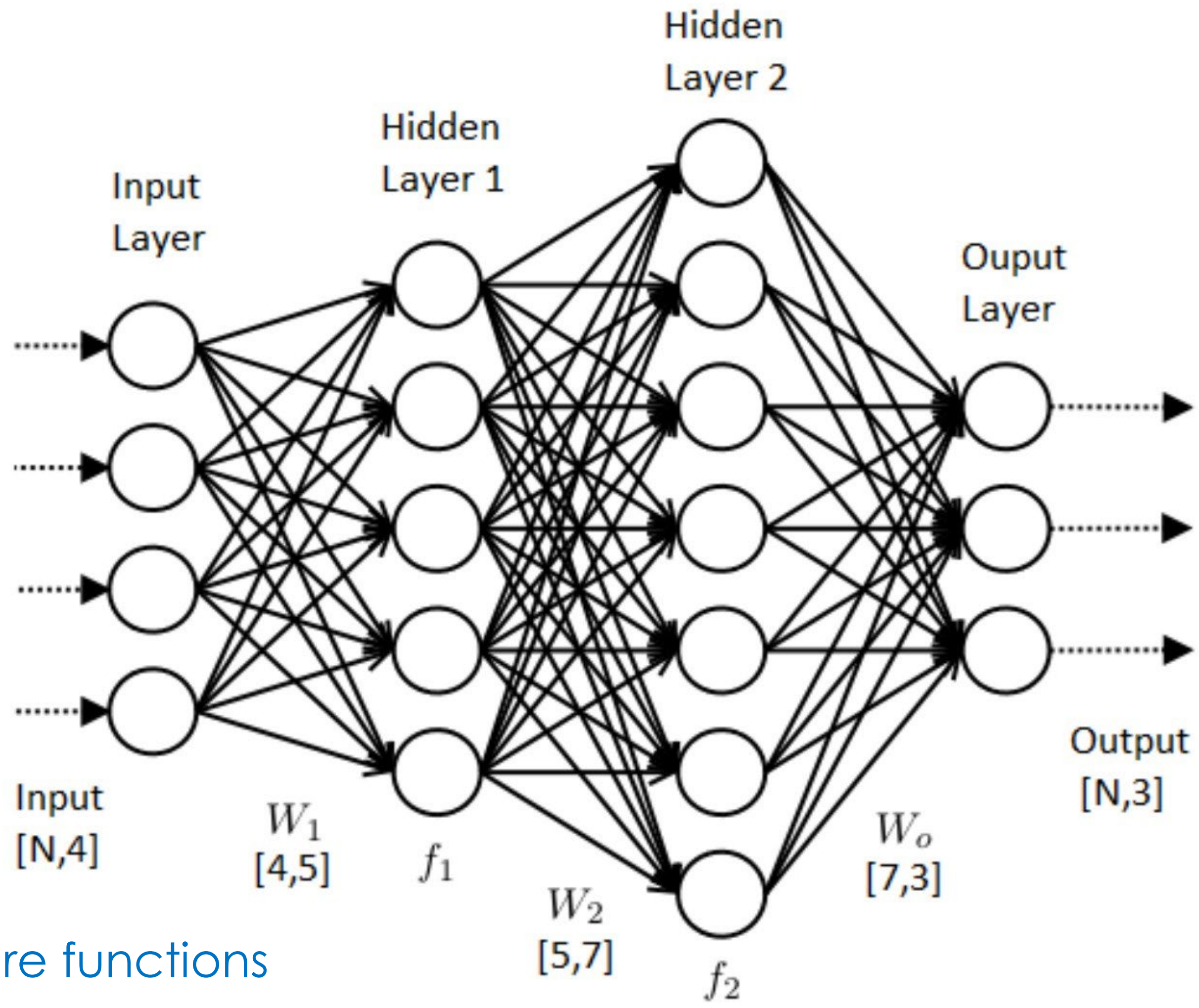
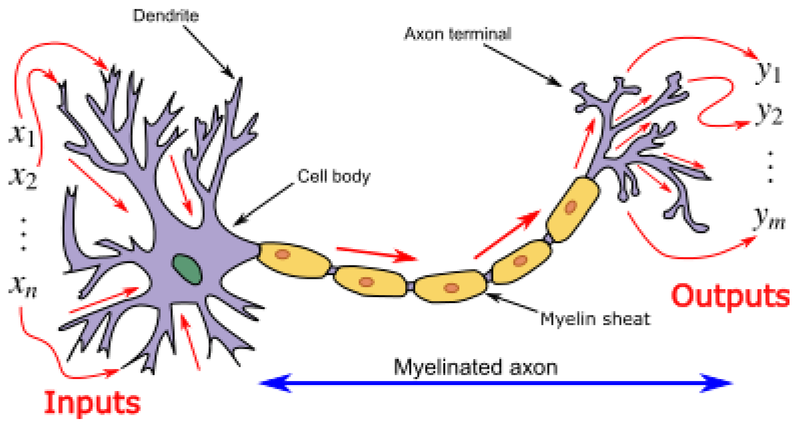


Image sources: medium.com; wikipedia.org

A detour --- Neural networks





Neural networks are functions

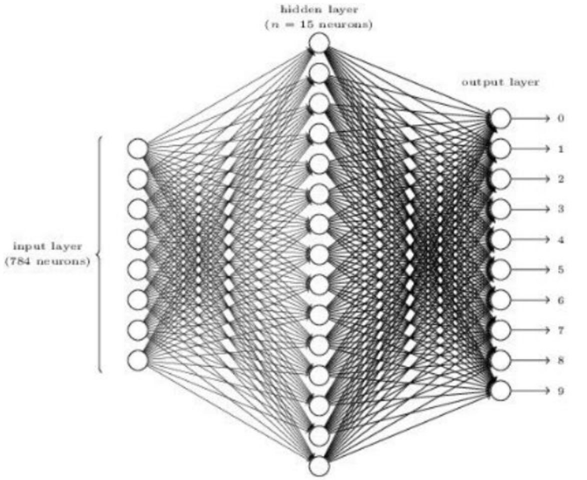
$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

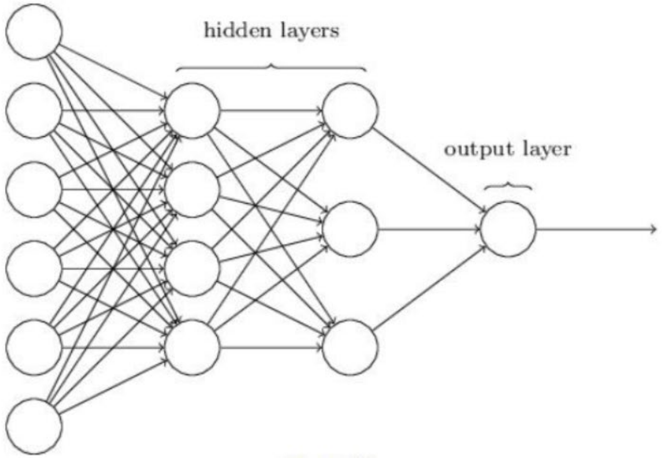
Image sources: medium.com; wikipedia.org

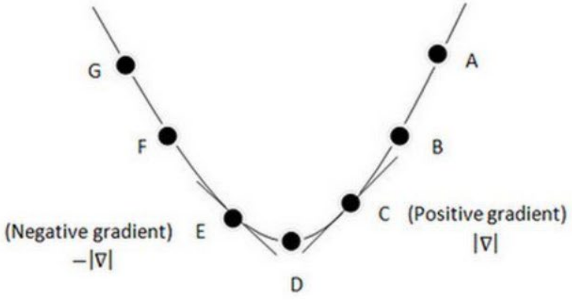
A detour --- Neural networks and the universal approximation theorem

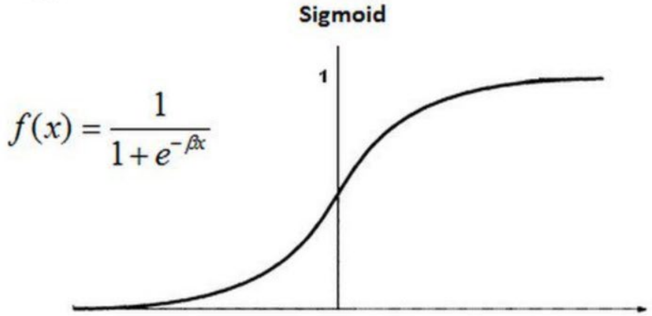
Neural networks can approximate (almost) arbitrary functions

Cybenko (1989) states that any continuous function on a compact domain can be **approximated with any precision** by an appropriate neural network with sufficient width and depth









(Partial, incomplete) History of Neural Networks

1763 Thinking in numbers – Thomas Bayes

1842 From numbers to poetry – Ada Lovelace

1943 Neurons go artificial

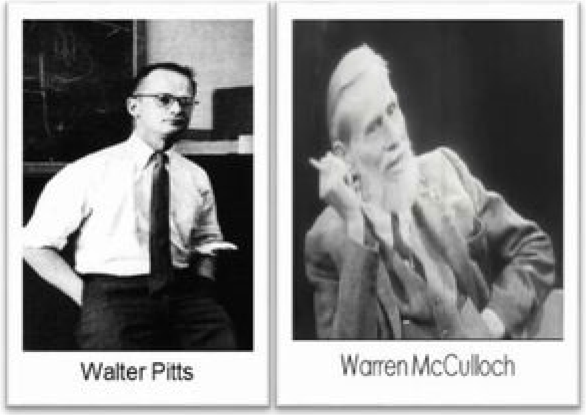
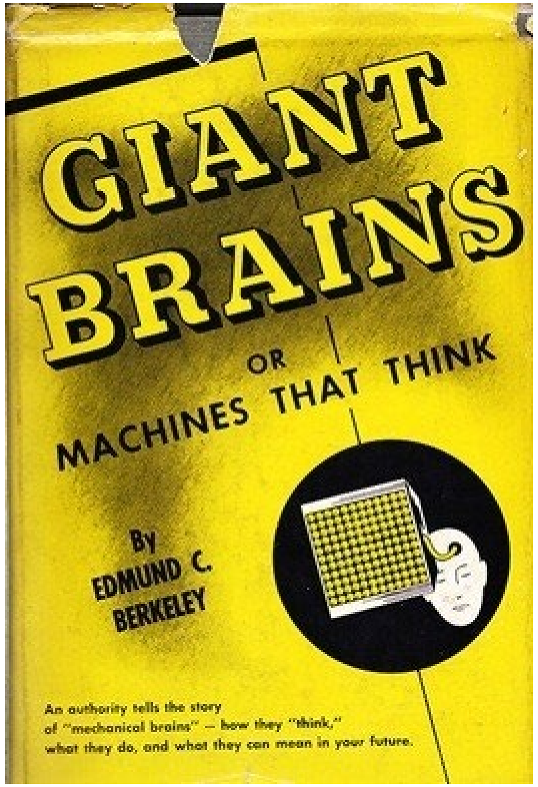


Image source: https://aiartists.org/ai-timeline-art

(Partial, incomplete) History of Neural Networks

1943 Can a machine think? – Edmund Berkeley



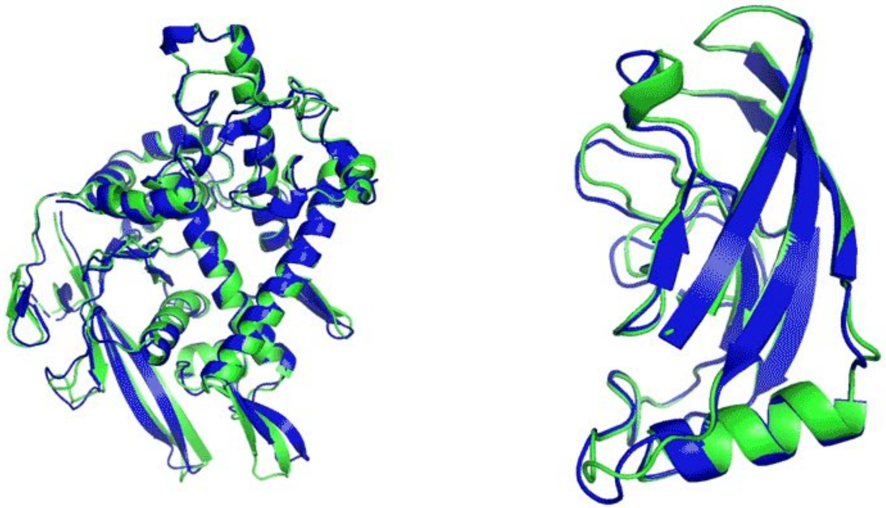
1997 Man vs. machine: fight of the 20th century

2002 The first robot for the home

Image source: https://aiartists.org/ai-timeline-art

The protein folding problem

Alphafold – A solution to a 50-year old grand challenge in biology



T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)

T1049 / 6y4f
93.3 GDT
(adhesin tip)

Experimental result
Computational prediction

From traditional econometrics to deep neural networks

Traditional econometrics

$$E(R_i) = R_f + \beta_i(E(R_m) - R_f)$$

Deep Neural Networks

ImageNet Classification with Deep Convolutional Neural ...

by A Krizhevsky · Cited by [illegible]498 · Related articles

The **neural network**, which has 60 **million parameters** and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling ...

We need more data

research.google.com › large_deep_networks_nips2012 ▾ PDF

Large Scale Distributed Deep Networks - Google Research

by J Dean · Cited by 3150 · Related articles

we consider the problem of training a **deep network** with billions of **parameters** using tens of thousands of CPU cores. We have developed a software framework.

A DAY IN DATA

The exponential growth of data is undisputed, but the numbers behind this explosion – fuelled by internet of things and the use of connected devcies – are hard to comprehend, particularly when looked at in the context of one day

500m

tweets are sent every day

Twitter

4PB

of data created by Facebook, including

350m photos

100m hours of video watch time

Facebook Research

320bn

emails to be sent each day by 2021

294bn

billion emails are sent

Radicati Group

306bn

emails to be sent each day by 2020

3.9bn

people use emails

4TB

of data produced by a connected car

Intel

DEMYSTIFIYING DATA UNITS

From the more familiar 'bit' or 'megabyte', larger units of measurement are more frequently being used to explain the masses of data

Unit		Value	Size
b	bit	0 or 1	1/8 of a byte
B	byte	8 bits	1 byte
KB	kilobyte	1,000 bytes	1,000 bytes
MB	megabyte	$1,000^2$ bytes	1,000,000 bytes
GB	gigabyte	$1,000^3$ bytes	1,000,000,000 bytes
TB	terabyte	$1,000^4$ bytes	1,000,000,000,000 bytes
PB	petabyte	$1,000^5$ bytes	1,000,000,000,000,000 bytes
EB	exabyte	$1,000^6$ bytes	1,000,000,000,000,000,000 bytes
ZB	zettabyte	$1,000^7$ bytes	1,000,000,000,000,000,000,000 bytes
YB	yottabyte	$1,000^8$ bytes	1,000,000,000,000,000,000,000,000 bytes

*A lowercase "b" is used as an abbreviation for bits, while an uppercase "B" represents bytes.

65bn

messages sent over WhatsApp and two billion minutes of voice and video calls made

Facebook

463EB

of data will be created every day by 2025

IDC

95m

photos and videos are shared on Instagram

Instagram Business

Searches made a day — 5bn

Searches made a day from Google — 3.5bn

Smart Insights

28PB

to be generated from wearable devices by 2020

Statista

ACCUMULATED DIGITAL UNIVERSE OF DATA

2013: 4.4ZB

2020: 44ZB

PwC

RACONTEUR

The need for synthetic data in 2021

Data is protected

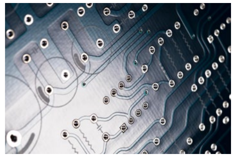
Privacy and compliance limit the use of banking/client data

Regulations prevent data sharing

Clearance and approvals are inefficient

The lack of sharing holds back research

Historical data is limited



Certain events present limited historical data

Limits statistical analysis and inference

ML models are crippled by small input sizes

Class imbalance

Class imbalance is high in rare-event datasets

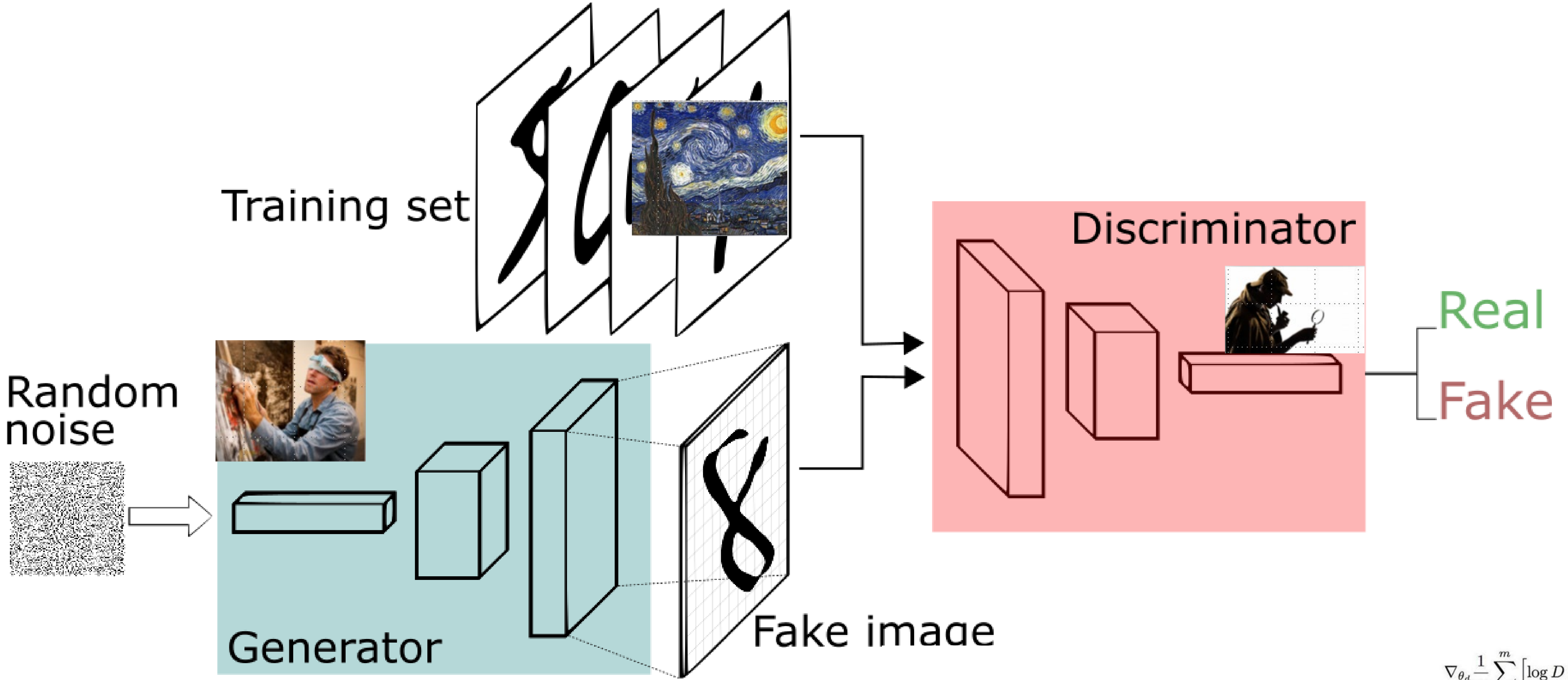
Datasets for fraud detection are often imbalanced

ML and anomaly detection algorithms fail on imbalanced data

Generative Adversarial Networks introduced in 2014

- Groundbreaking work by Ian Goodfellow et al (2014)
- It tried to address the following question: Given a set of data (say, a set of human faces or Van Gogh paintings), can we generate data that are “similar”?
- The authors have proposed GAN, which uses two neural networks “competing against” each other to obtain the desired outcome.
- Yann LeCun has said “this (GAN) and the variations are now the most interesting idea in the last 10 years in ML, in my opinion.”

$$\min_{\mathcal{G}} \max_{\mathcal{D}} E_x\left[\log(\mathcal{D}(x))\right] + E_z\left[\log(1-\mathcal{D}(\mathcal{G}(z)))\right]$$



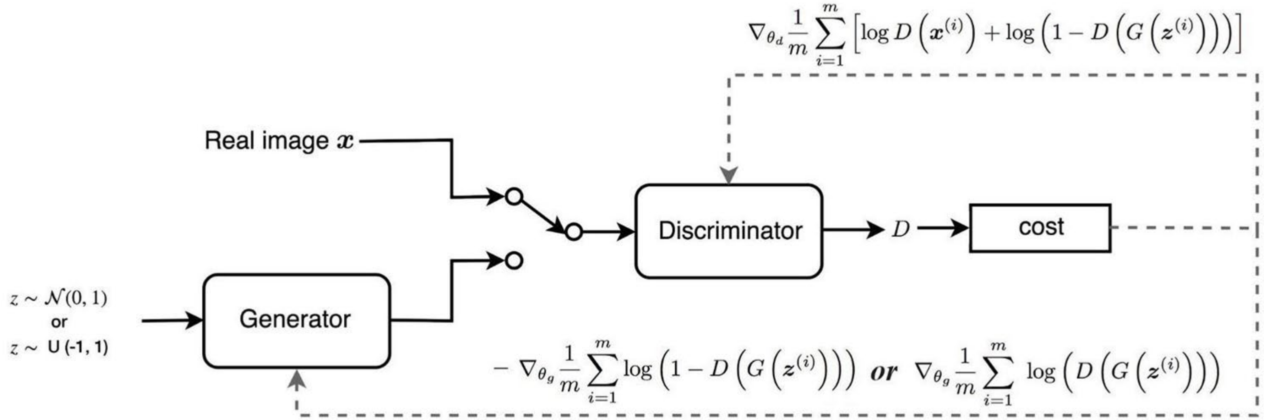


Image source: Google Images

Goodfellow, Ian J., Jean Pouget-Abadie, M. Mirza, B. Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville and Yoshua Bengio. "Generative Adversarial Networks." *ArXiv* abs/1406.2661 (2014)

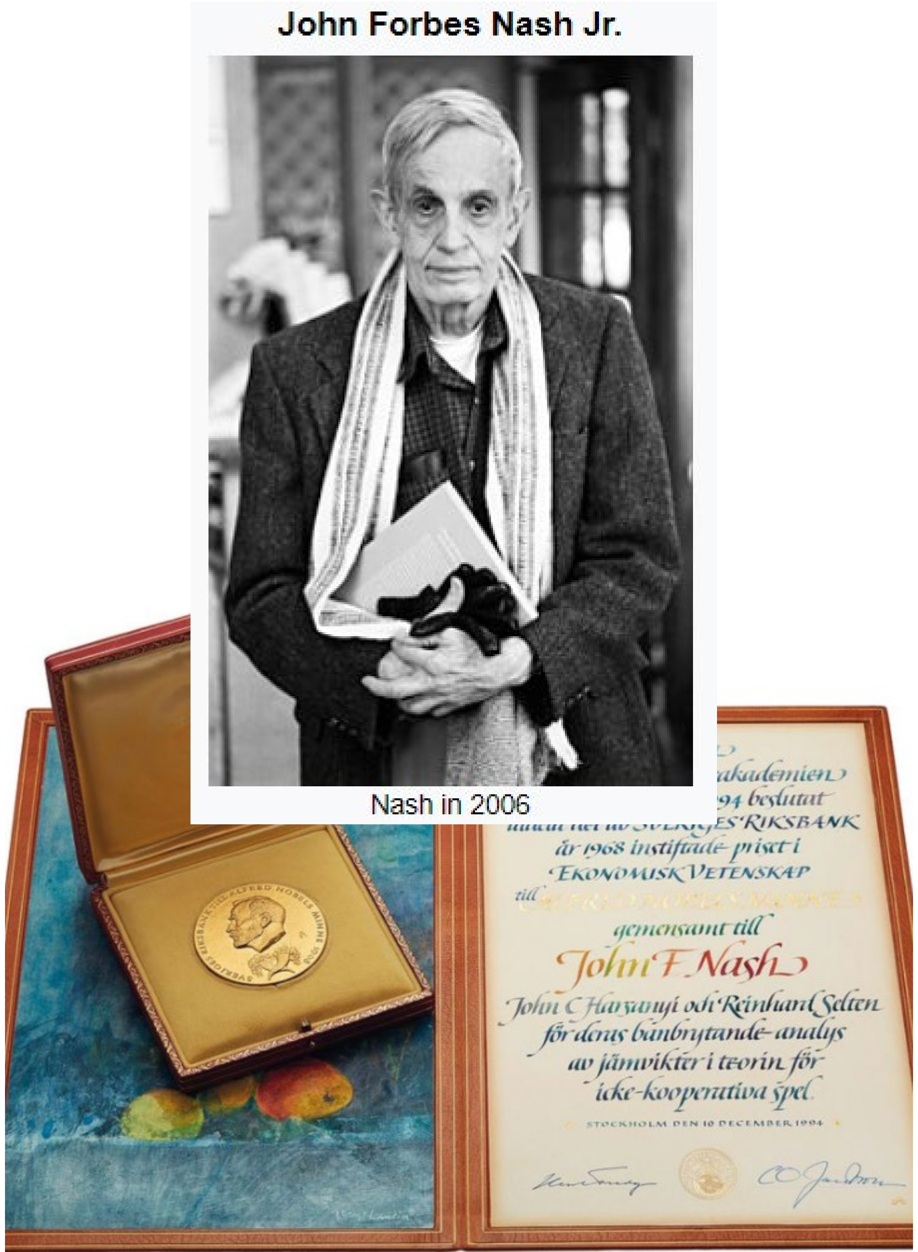
A Detour – Nash Equilibrium

GANs and Nash Equilibrium

The GAN framework is a non-convex, two-player, non-cooperative game with continuous, high-dimensional parameters, in which each player wants to minimize its cost function.

The optimum of this process takes the name of Nash Equilibrium.

GANs are typically trained using *gradient-descent* techniques that are designed to find the low value of a *cost function* and not find the *Nash Equilibrium* of a game



What is a Nash Equilibrium?

No participant can gain by a unilateral change of strategy if the strategies of the others remain unchanged

What else? A zoo of GANs - Different network architectures

Foundation
- Generative Adversarial Network (GAN)
- Deep Convolutional Generative Adversarial Network (DCGAN)

Extensions
- Conditional Generative Adversarial Network (cGAN)
- Information Maximizing Generative Adversarial Network (InfoGAN)
- Auxiliary Classifier Generative Adversarial Network (AC-GAN)
- Stacked Generative Adversarial Network (StackGAN)
- Context Encoders
- Pix2Pix

Advanced
- Wasserstein Generative Adversarial Network (WGAN)
- Cycle-Consistent Generative Adversarial Network (CycleGAN)
- Progressive Growing Generative Adversarial Network (Progressive GAN)
- Style-Based Generative Adversarial Network (StyleGAN)
- Big Generative Adversarial Network (BigGAN)
- Ensembles of GANs

Lucic, M., Karol Kurach, M. Michalski, S. Gelly and O. Bousquet. "Are GANs Created Equal? A Large-Scale Study." *NeurIPS* (2018).
https://github.com/hindupuravinash/the-gan-zoo

What else? A zoo of GANs - Different network architectures

Five building blocks

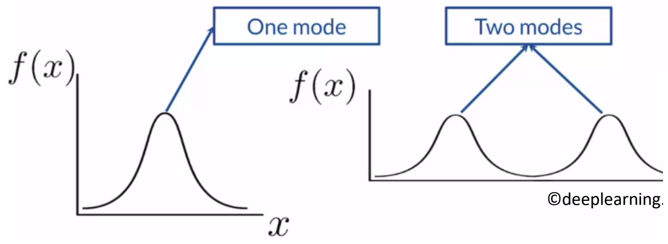
- Generator network
- Discriminator network
- Loss functions
- Regularizations (weights, loss, gradient)
- Optimizers

Ongoing challenges in GAN training

Mode collapse

Model gets stuck in a mode

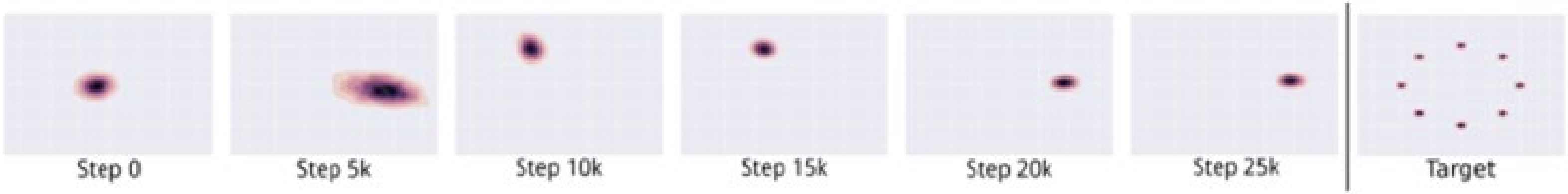
Output loses diversity



Lack of proper evaluation metrics

No metric for time series GANs[1]

Hard to evaluate GANs with precision



Proposed solutions

New Loss functions

New Model architectures

Additional regularizations

[1] but see Conditional Sig-Wasserstein GANs for Time Series Generation, Ni et al 2020

Metz et al. 2017

Challenges, tips and tricks when training GANs

- Normalize the inputs
- A modified loss function: (min(1-D) -> max log(D))
- Sample from Gaussian instead of uniform distribution
- Batch Normalization

- Avoid Sparse Gradients
- LeakyReLU = good (in both G and D)
- For Downsampling, use: Average Pooling, Conv2d + stride
- Use Soft and Noisy Labels

- Use stability tricks from RL
- Track failures early
- D loss goes to 0: failure mode
- Check norms of gradients: if they are over 100, it becomes difficult
- When things are working, D loss has low variance and goes down over time vs having huge variance and spiking
- If loss of generator steadily decreases, then it's fooling D with garbage

https://github.com/soumith/ganhacks

GANs in Finance

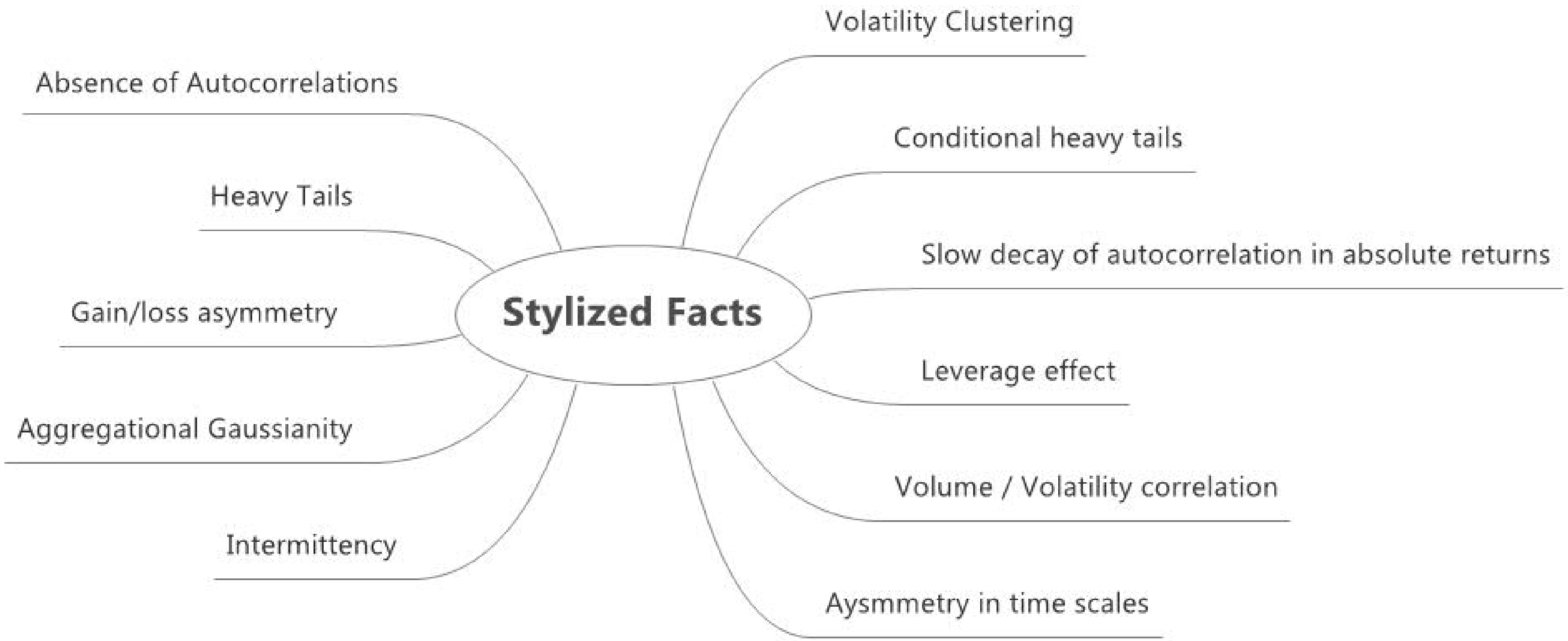
Table 1: GANs in finance research

Field	Application	Method
Time Series Forecasting	Market Prediction	GAN-FD [9], ST-GAN [19], MTSGAN [20]
	Fine-Tuning of trading models	C-GAN [10], MAS-GAN[21]
Portfolio Management	Porfolio Optimization	PAGAN[11], GAN-MP[22], DAT-CGAN[23], CorrGAN[12]
Time Series Generation	Synthetic time series generation and Finance Data Augmentation	TimeGAN[24], WGAN-GP[25], FIN-GAN[3], Quant GAN[14], RA-GAN[26], CDRAGAN[27], SigCWGAN[28], ST-GAN[19]
Fraud Detection	Detection of market manipulation	LSTM-GAN[13]
	Detection of Credit Card Fraud	RWGAN[29], LSTM-GAN-2[30]

Eckerli, Osterrieder 2021

The key stylized facts of financial time series

- Absence of autocorrelations
- Fat-tailed distributions
- Volatility clustering
- Gain/loss asymmetry
- Aggregational Gaussianity

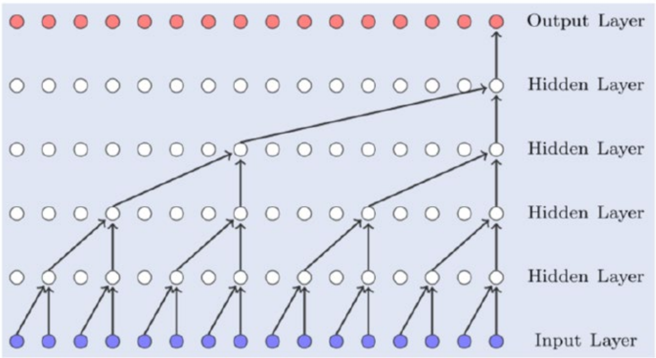


Financial time-series data is non-stationary, non-markovian, with non-parametric distributions

https://radhakrishna.typepad.com/rks_musings/2014/06/stylized-facts.html

Results for financial-time series GANs

Quant-GAN[1]



Conditional Sig-Wasserstein GANs[2]

Signature of a path

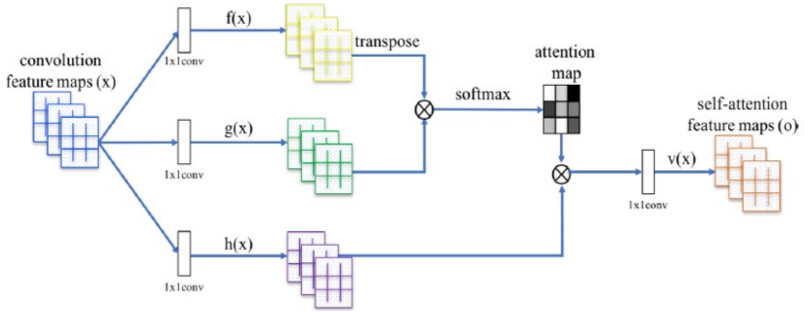
Wasserstein GAN with gradient penalty[5]

Improved training of Wasserstein GAN, which pushes the discriminator Lipschitz by gradient penalty

RegGAN[4]

Regularized GAN: an architecture with two discriminators: one as a typical GAN, binary classifier, and the other one as a score function

Self-Attention GAN[3]



TransGAN[6]

GAN with transformer blocks but without any convolutional layers

[1] Wiese, M., Knobloch, R., Korn, R., & Kretschmer, P. (2019). Quant GANs: deep generation of financial time series. *Quantitative Finance, 20*, 1419 - 1440.
[2]Ni, H., Szpruch, L., Wiese, M., Liao, S., & Xiao, B. (2020). Conditional Sig-Wasserstein GANs for Time Series Generation. *DecisionSciRN: Probabilistic Graphical Models (Topic)*.
[3]Zhang, H., Goodfellow, I., Metaxas, D.N., & Odena, A. (2019). Self-Attention Generative Adversarial Networks. *ICML*.
[4]Cerbo, G.D., Hirsa, A., & Shayaan, A. (2021). Regularized Generative Adversarial Network. *ArXiv, abs/2102.04593*.
[5]Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of wasserstein gans. arXiv preprint arXiv:1704.00028.
[6]Jiang, Y., Chang, S., & Wang, Z. (2021). Transgan: Two transformers can make one strong gan. arXiv preprint arXiv:2102.07074.

GAN variants implemented for time-series

Structural Variants

DCGAN: Deep Convolutional GAN

SAGAN: self-attention GAN with dense attention

BIG GAN deep: larger versions of SAGAN

YLGAN: your local GAN, with sparse attention

Transformer GAN: with transformer blocks

Loss variants

Wasserstein GAN

WGAN with Gradient Penalty

LS GAN: least-squares

RAGAN: loss function improvement of DCGAN with realistic factors

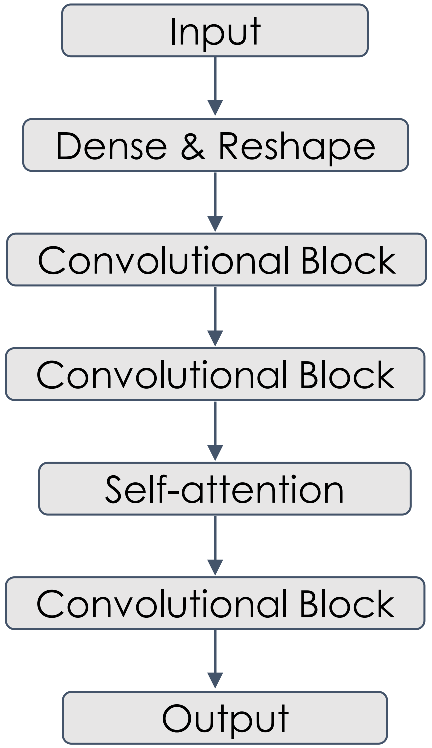
RA LS GAN: loss function improvement of LSGAN with realistic factors

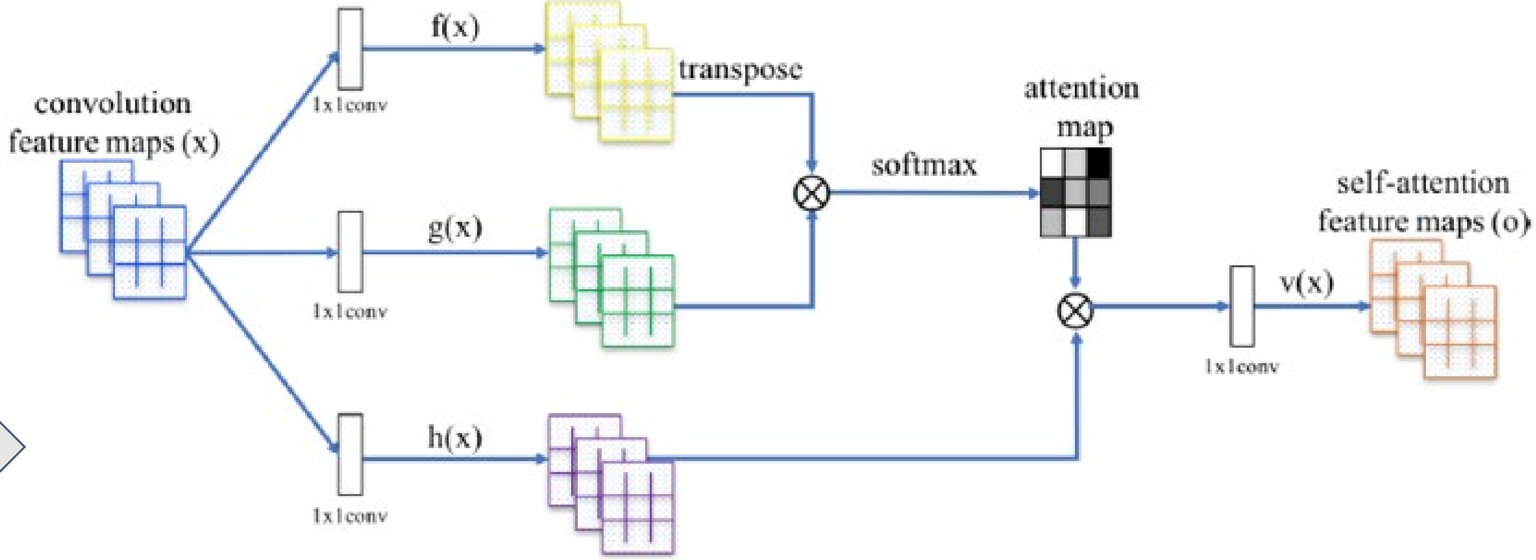
DRAGAN: deep regret analytic GAN, a loss variant similar to WGAN GP

See Hirsa, Fu, Osterrieder, 2021

SAGAN

GAN based on convolutional neural networks, with an added self-attention mechanism that improves learning on long-range dependencies





Input x: Batchsize (B) × Length (L) × Channels (C)

f(x),g(x),h(x): linear transformation of x

attention(x) = softmax(f(x) × g(x)) × h(x)

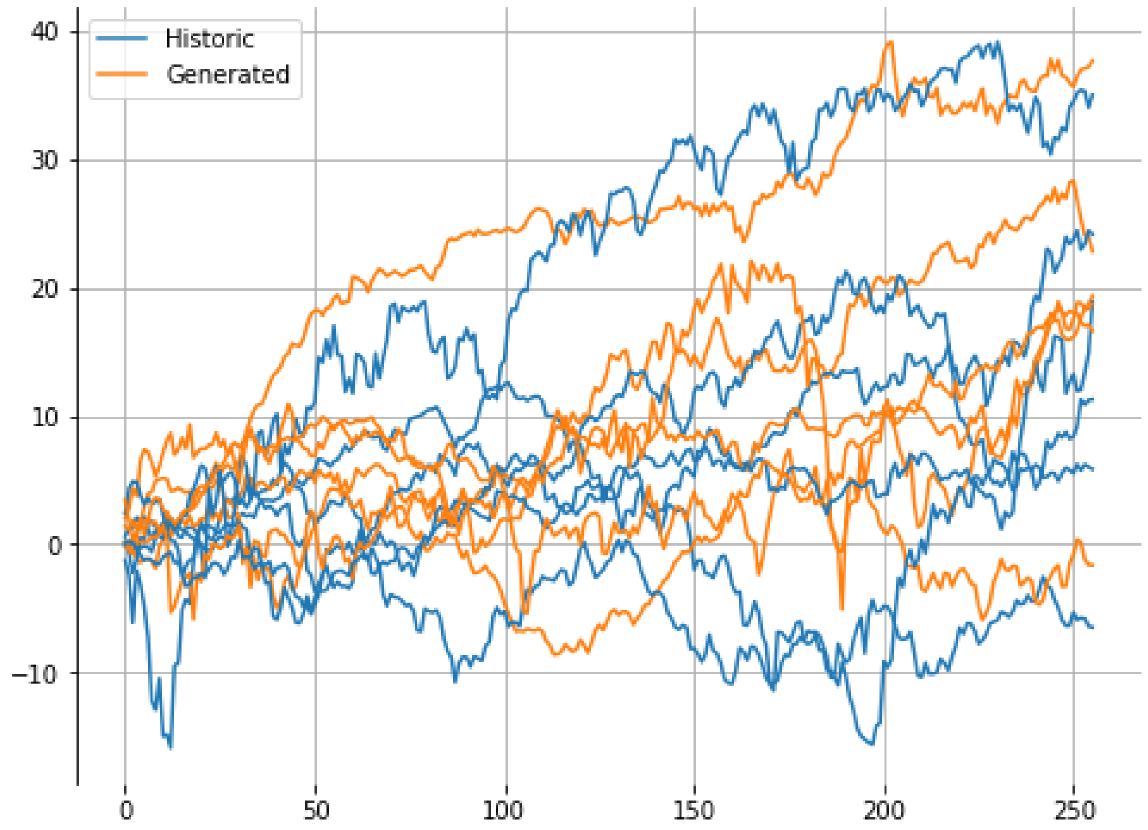
Output v(x): linear transformation of attention(x)

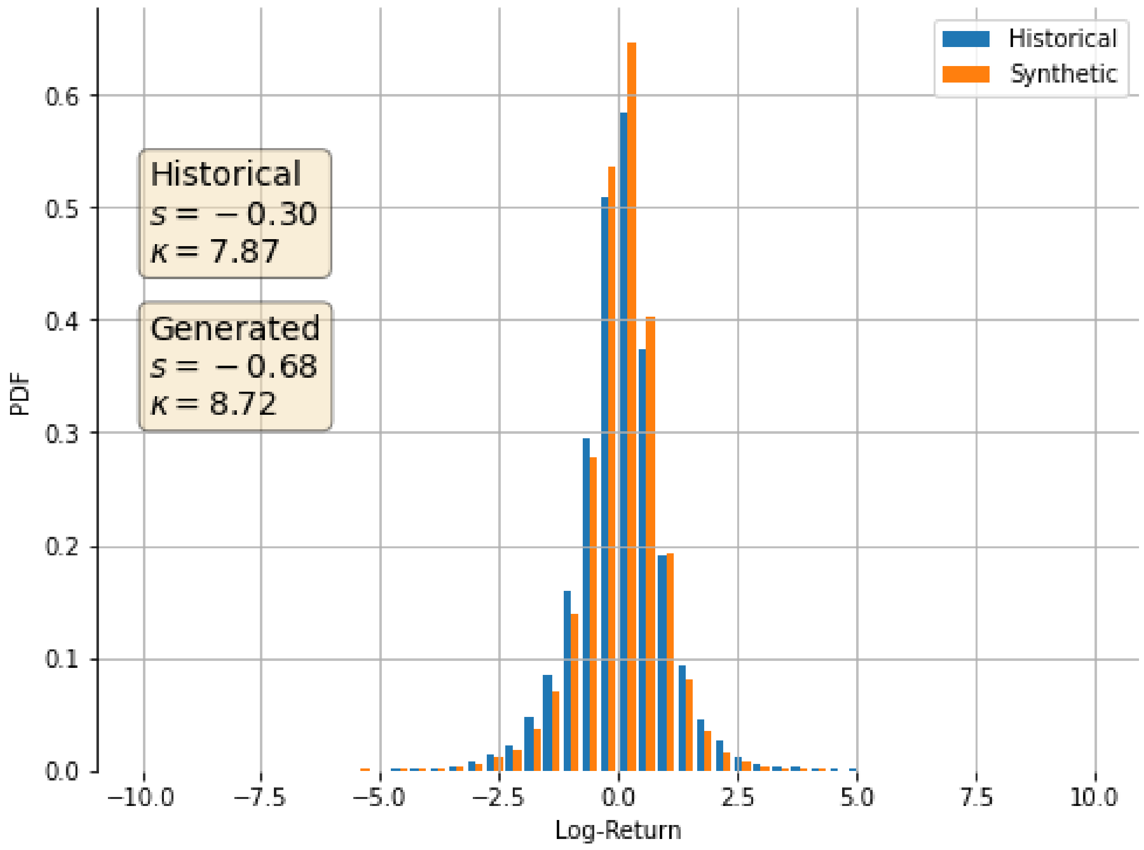
SAGAN-GP for time series

GAN modified from SAGAN to fit the task of financial time-series simulation

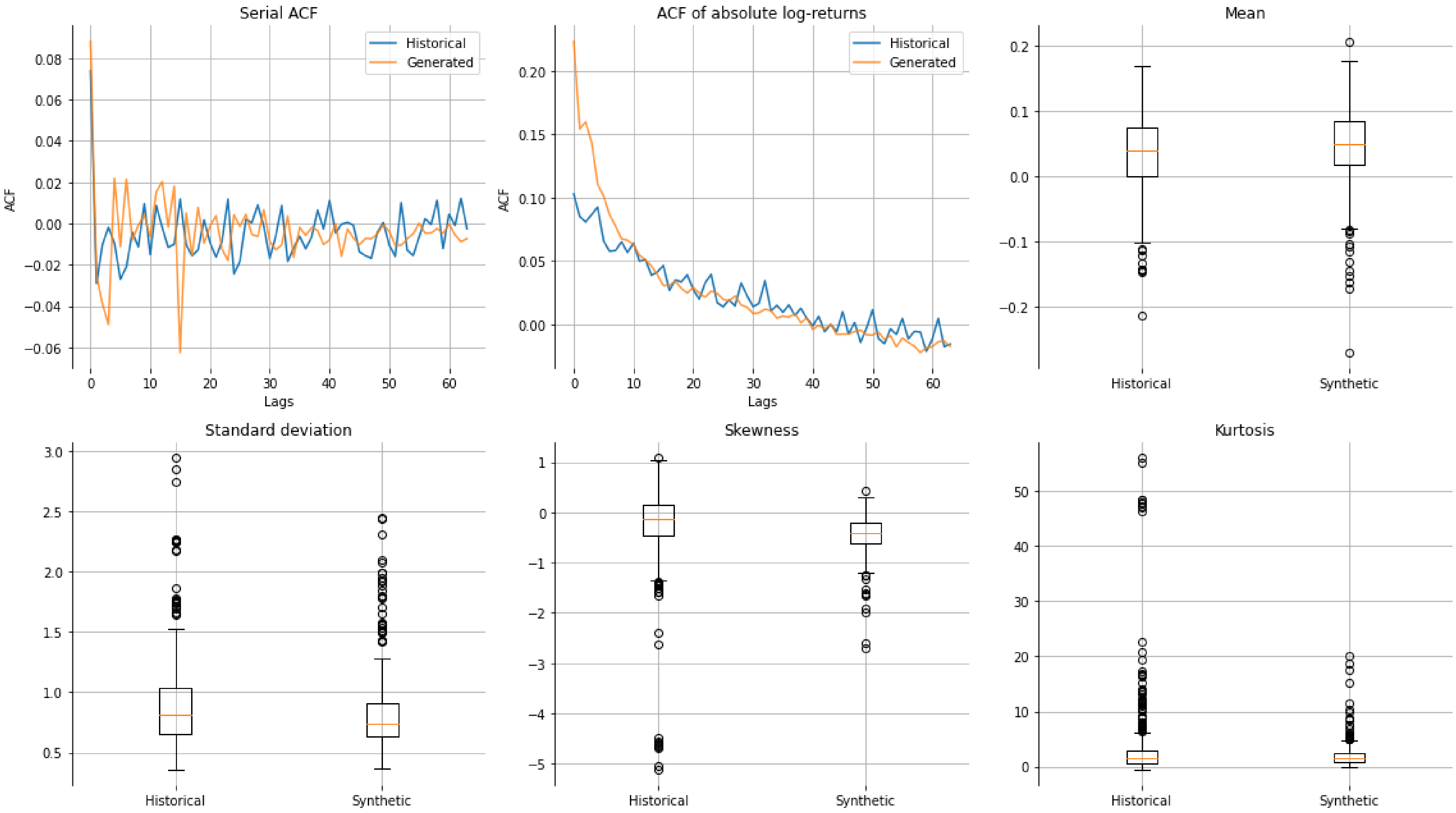
1. *Innovation*: Combining self-attention mechanism with convolutional networks applied to financial time-series simulation
2. *Main differences from SAGAN*
   a. Dimension changed from Batchsize (B) × Width (W) × Height (H) × Channels (C) for pictures to Batchsize (B) × Length (L) × Channels (C) for series
   b. Use returns and prices as the real data, such that both the moments of returns and long-range dependency of returns can be well-fitted by the GAN model
   c. Use the loss function of WGAN-GP instead of the hinge loss in the original SAGAN loss to improve training speed

S&P 500 SAGAN-GP, daily data





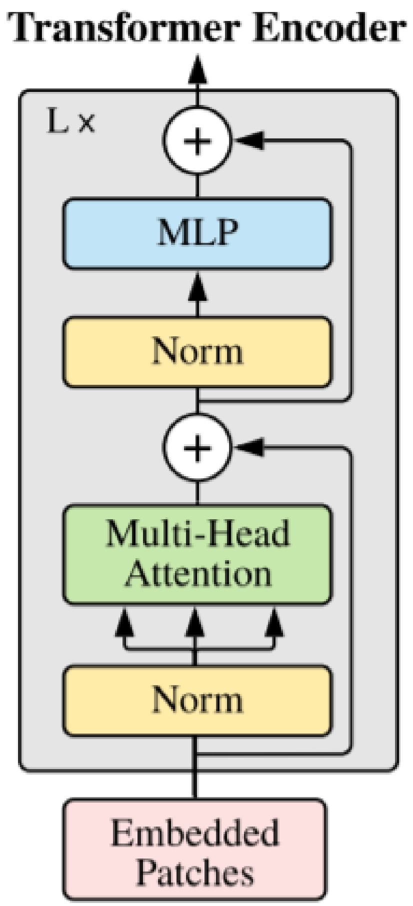
S&P 500 SAGAN-GP, daily data



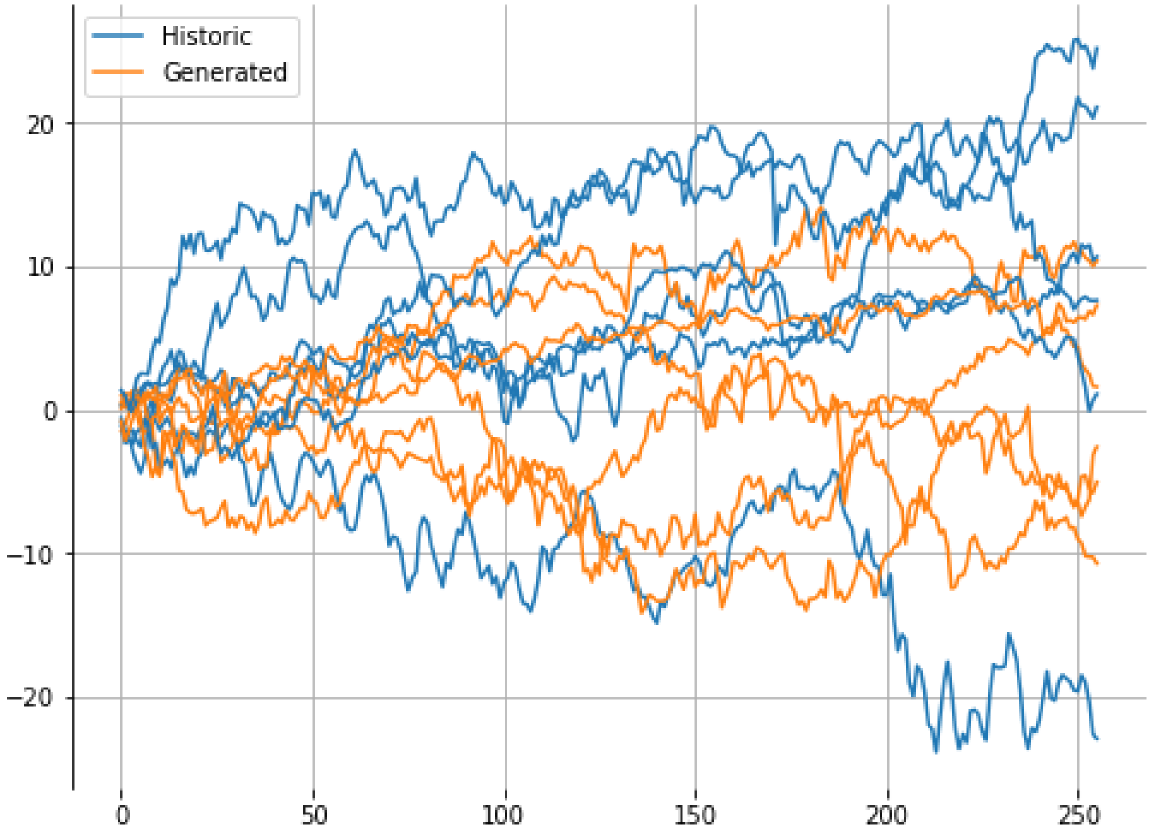
TransGAN

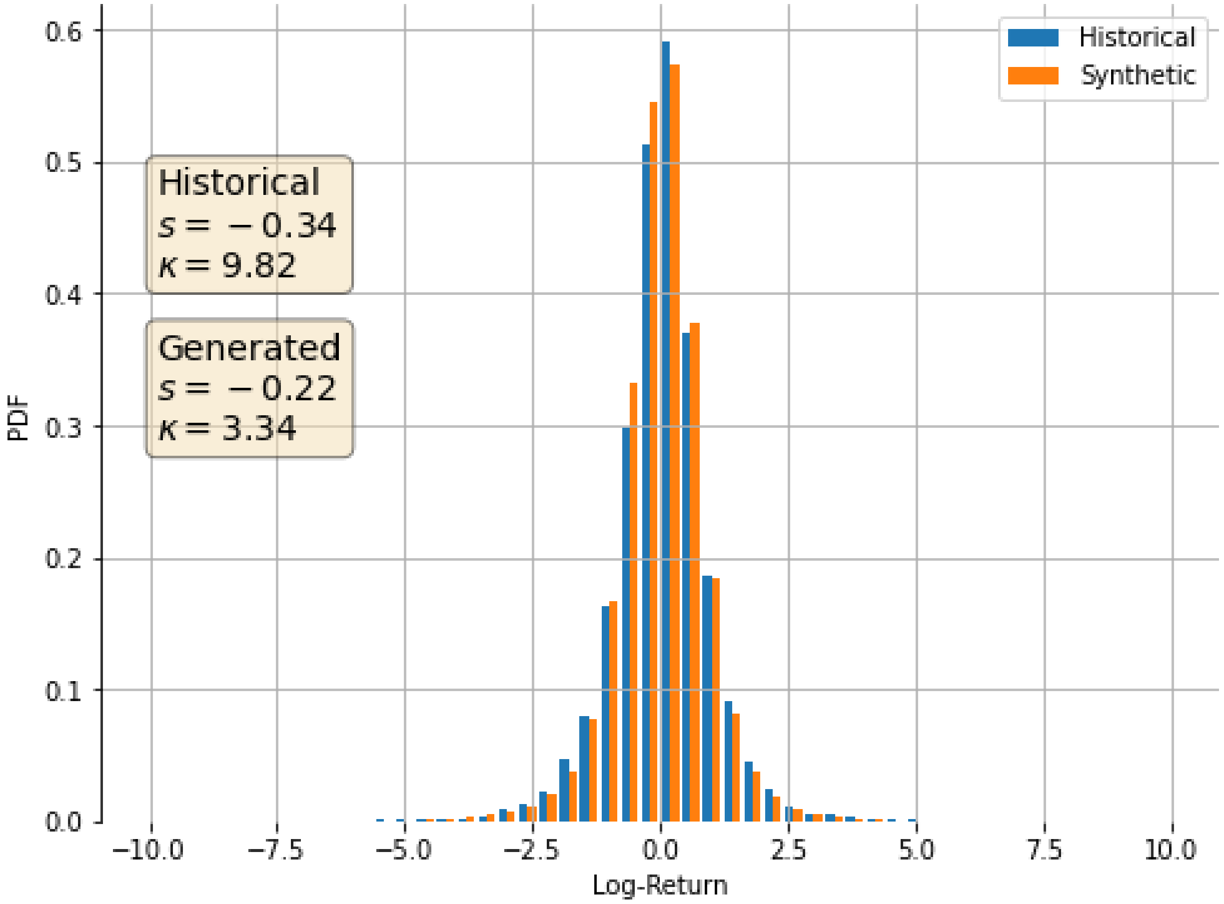
Main features:

a. GAN with only transformer blocks and without any convolutional layers
b. Especially good at fitting long-range dependencies and global characteristics
c. Consists of 3-5 transformer blocks
d. Each transformer block is made up by an attention layer and an MLP (fully-connected layers)

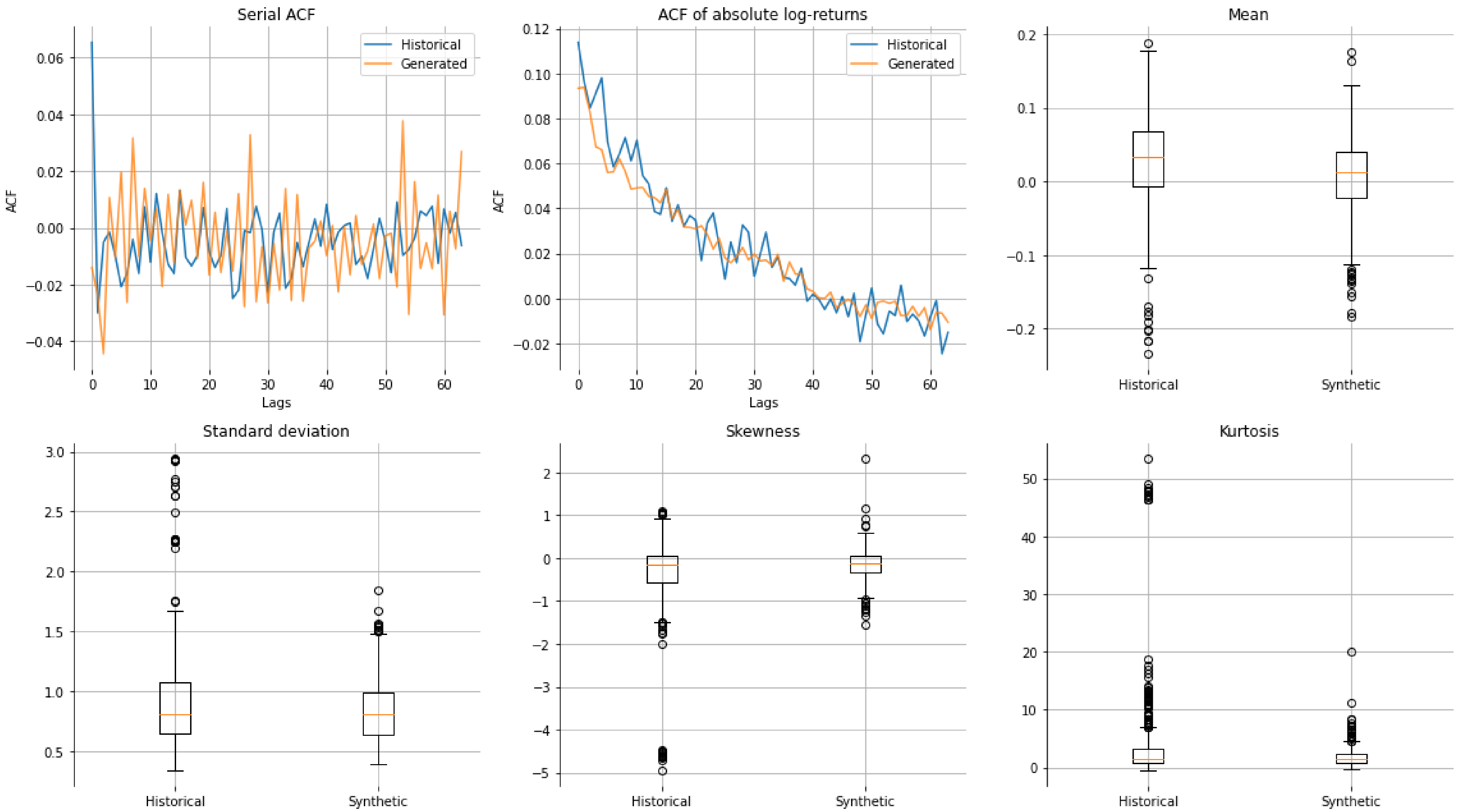


S&P 500 TransGAN, daily data

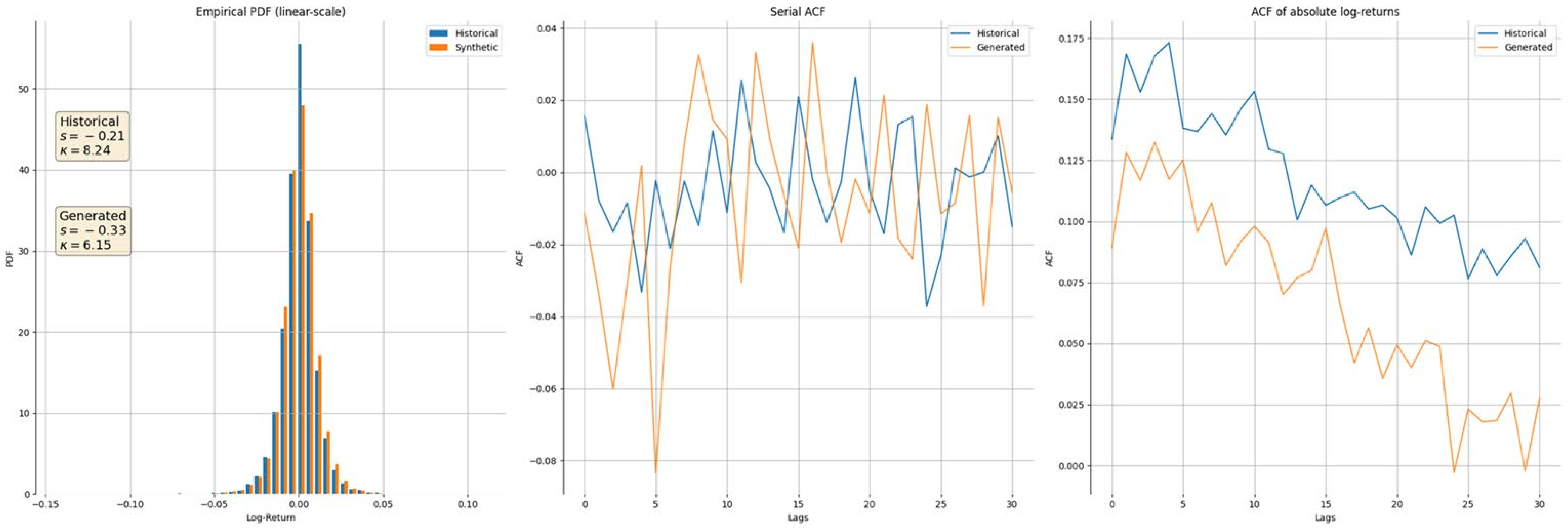




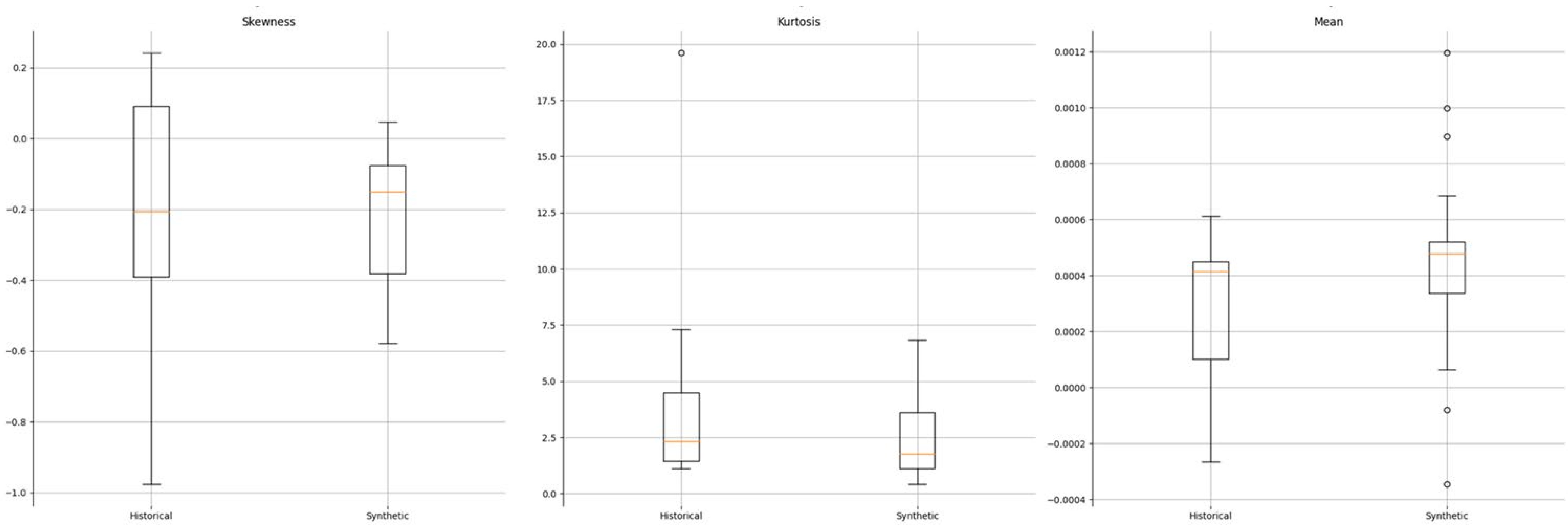
S&P 500 TransGAN, daily data



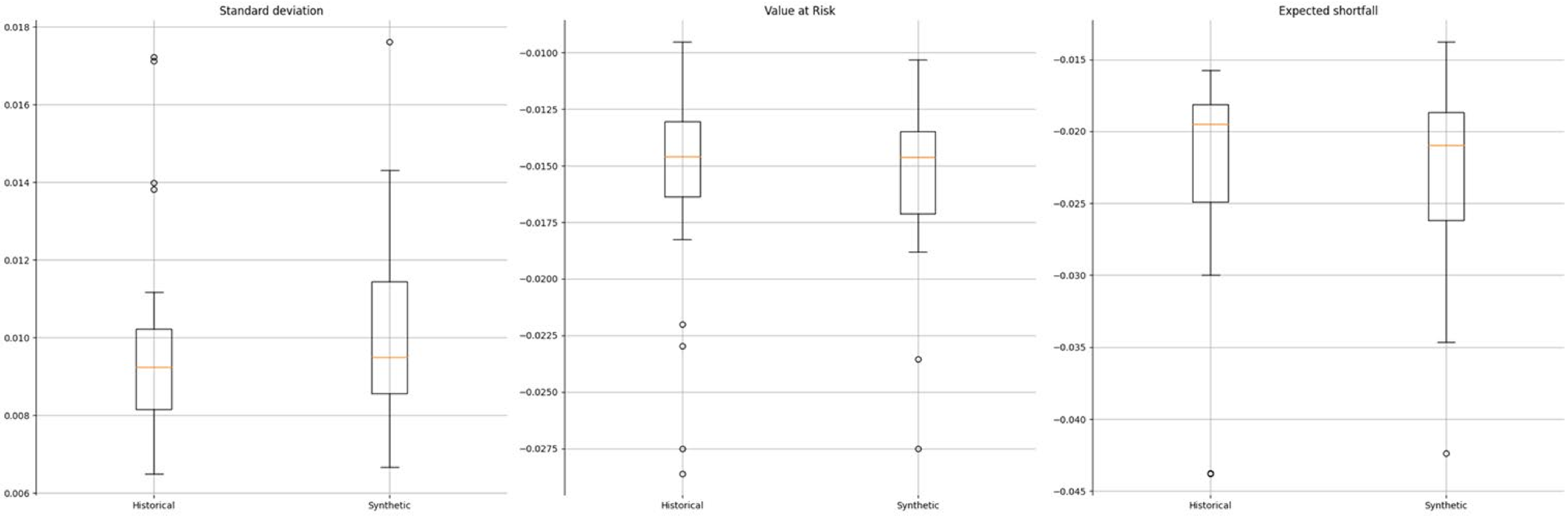
S&P 500 W-GAN with GP, daily data



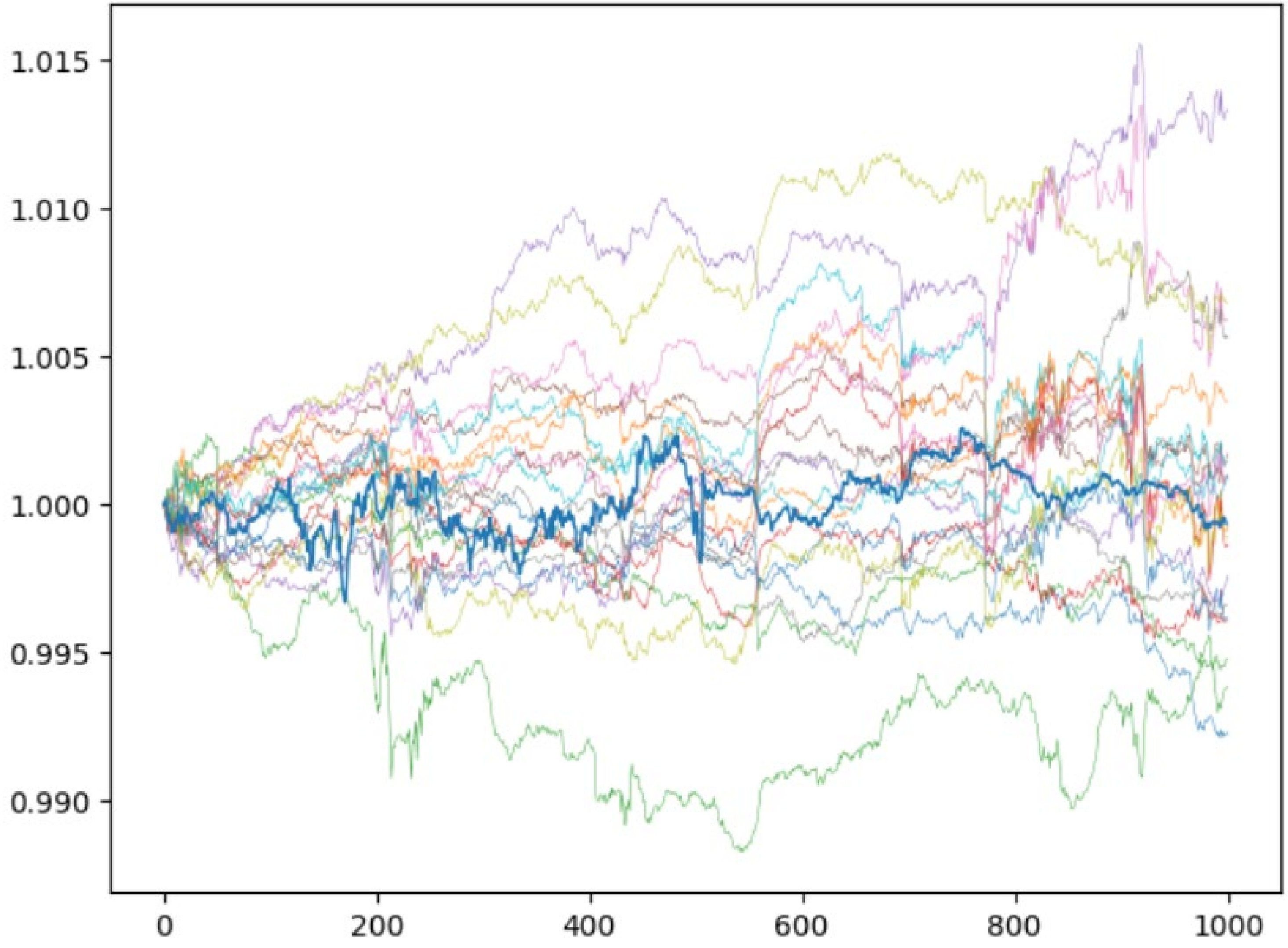
S&P 500 W-GAN with GP, daily data



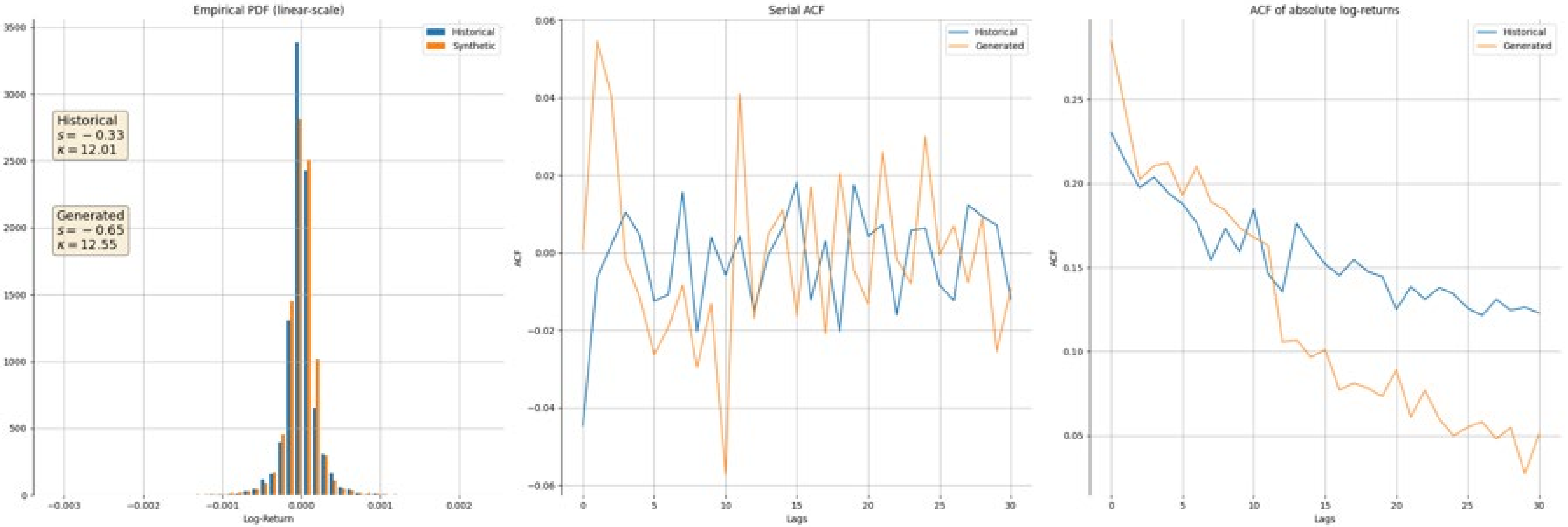
S&P 500 W-GAN with GP, daily data



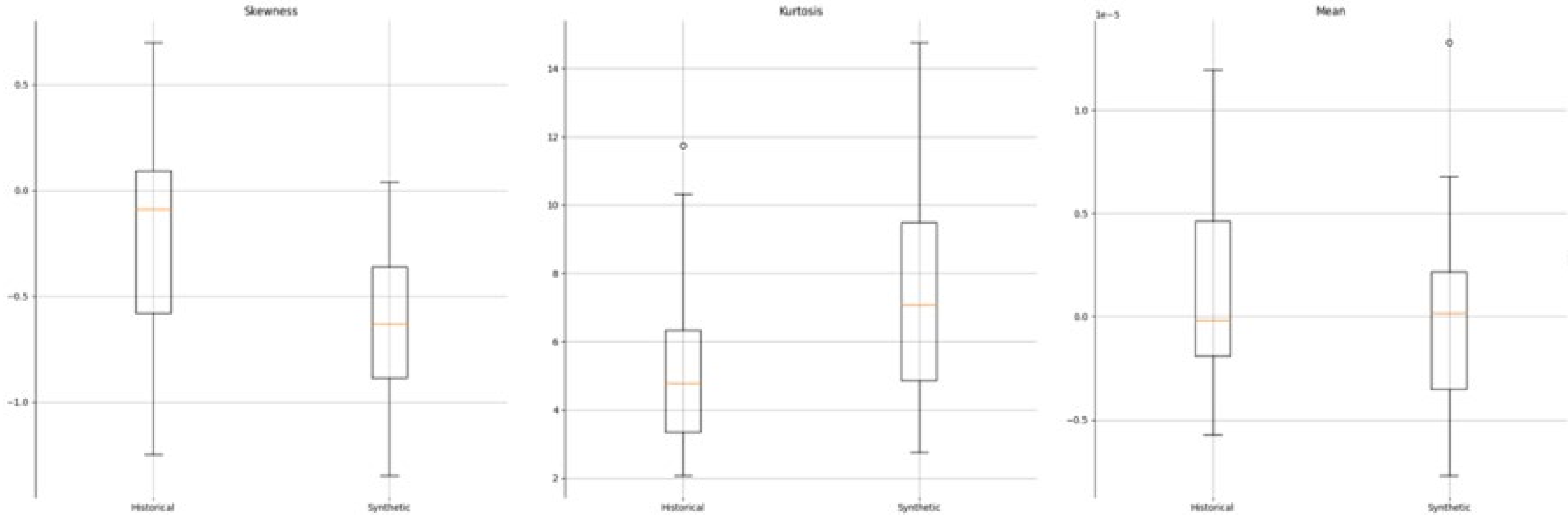
S&P 500 W-GAN with GP, intraday data



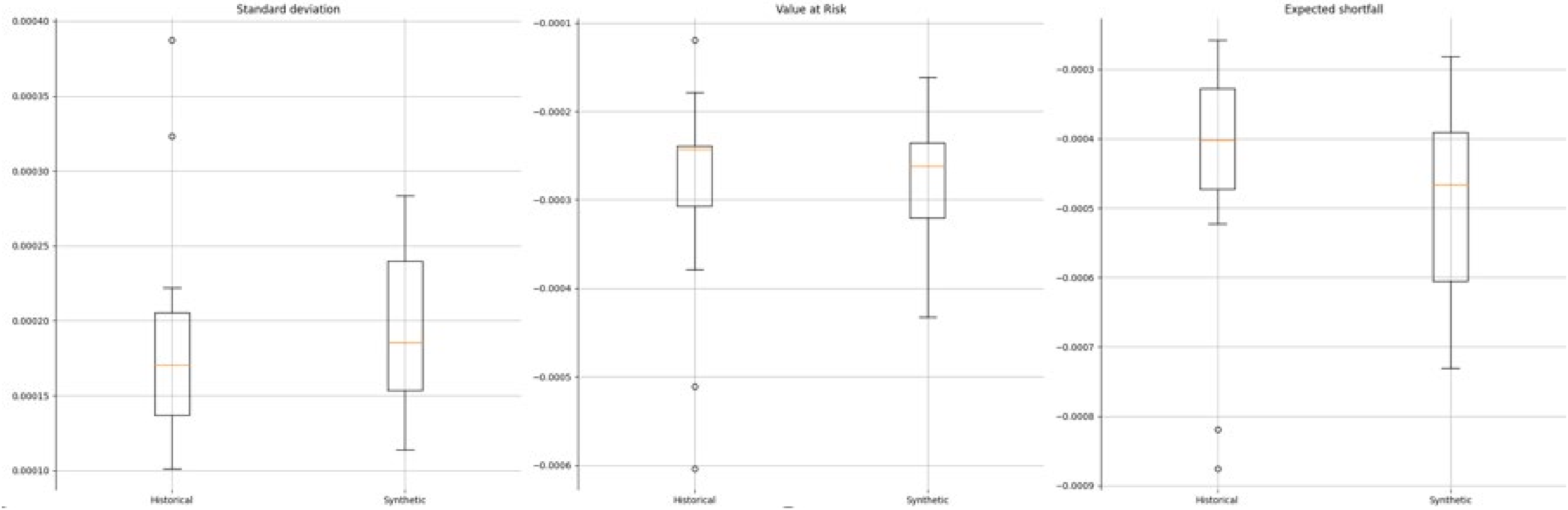
S&P 500 W-GAN with GP, intraday data



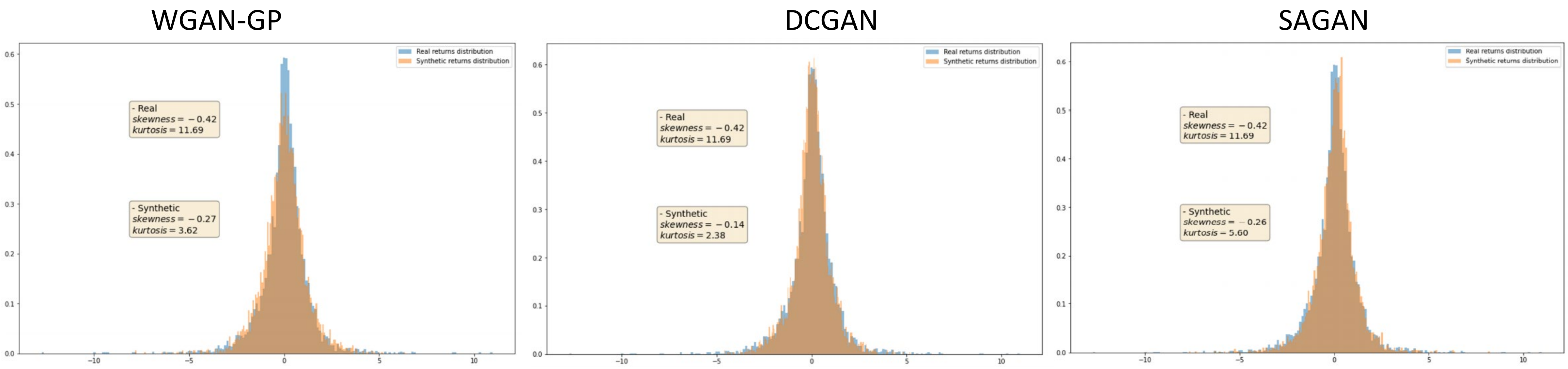
S&P 500 W-GAN with GP, intraday data



S&P 500 W-GAN with GP, intraday data



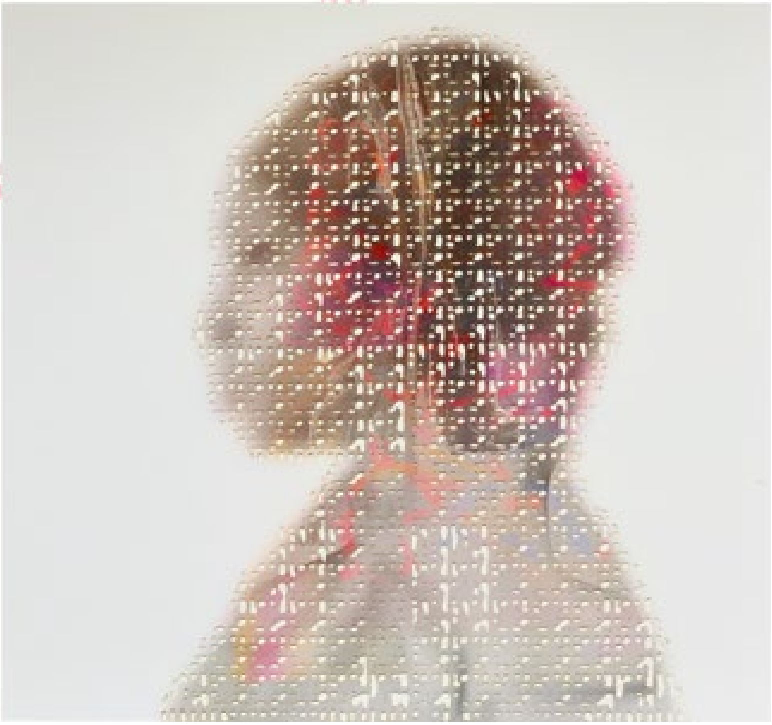
Probability distribution, Kurtosis and Skewness



RegGAN[1]

The initial idea was to replicate artists' cuts/ patterns utiliizing deep convolutional generative adversarial networks with connected components

From art to finance, the idea can be extended to preserve statistical properties in financial time series

[1]Cerbo, G.D., Hirsa, A., & Shayaan, A. (2021). Regularized Generative Adversarial Network. *ArXiv, abs/2102.04593*.

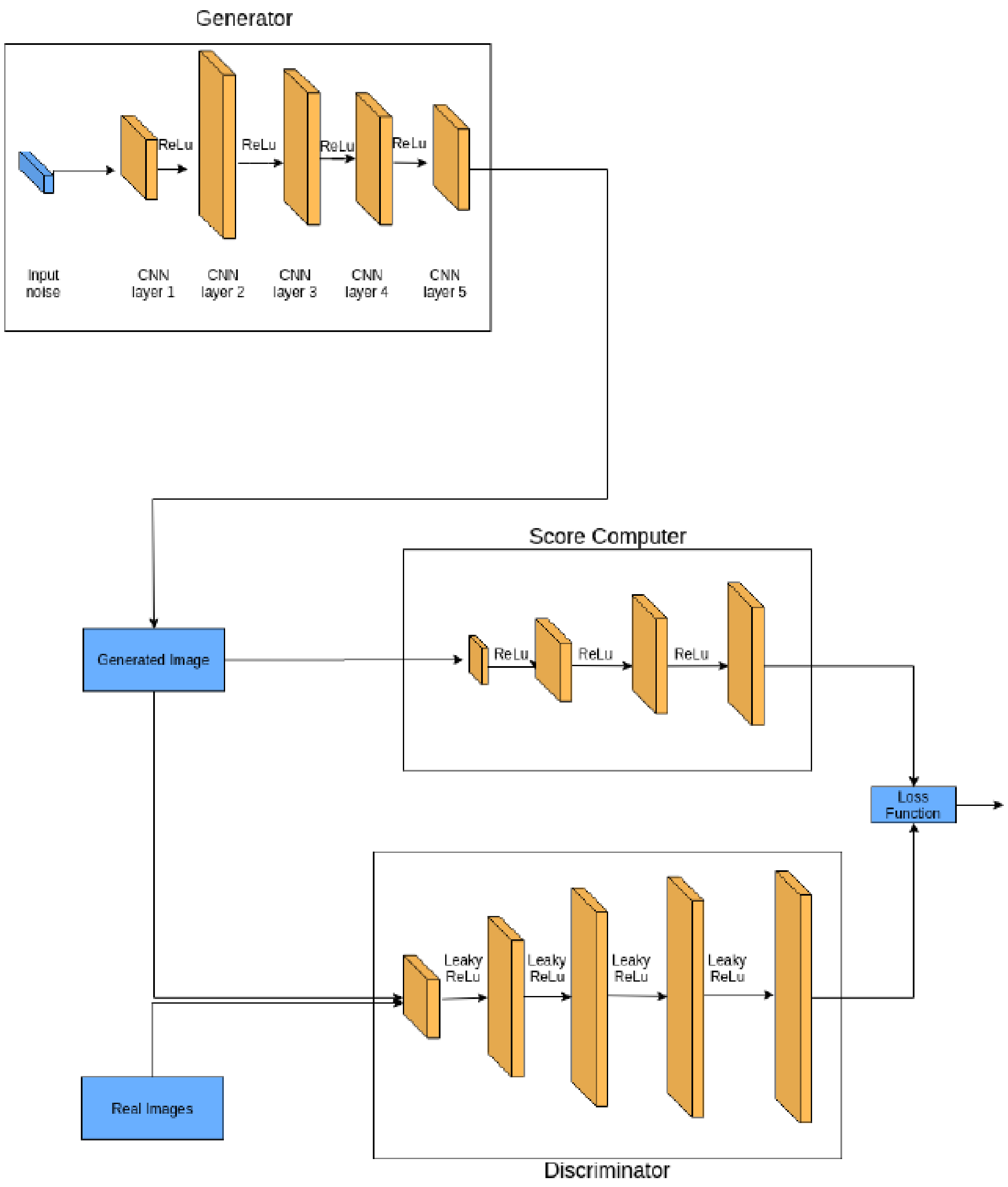
RegGAN Loss function

$$\min_{G} \max_{D_1, D_2} V(G, D_1, D_2)$$
$$= \mathbb{E}_x \left[\log(D_1(x))\right] + \mathbb{E}_z \log(1 - D_1(G(z))) + \mathbb{E}_z \log(1 - D_2(G(z)))$$

instead of

$$\min_{G} \max_{D} \left\{\mathbb{E}_{X \sim \mathbb{P}_r}\left[\log D(X)\right] + \mathbb{E}_{Z \sim \mathbb{P}_z}\left[\log(1 - D(G(Z)))\right]\right\}$$

- Regularized GAN: an architecture with two discriminators: one as a typical GAN, binary classifier, and the other one as a score function to control connected components
- for controlling different statistical properties of time-series such as heavy-tailedness, skewness and autocorrelation
- for introducing no-arbitrage constraints (e.g. for a volatility surface/ option prices)


Generator
ReLu
ReLu
ReLu
ReLu
Input noise
CNN layer 1
CNN layer 2
CNN layer 3
CNN layer 4
CNN layer 5
Score Computer
Generated Image
ReLu
ReLu
ReLu
Loss Function
Leaky ReLu
Leaky ReLu
Leaky ReLu
Leaky ReLu
Real Images
Discriminator

What next?[1]

Generate synthetic data

Apply Wasserstein GANs (and improved versions) to

- Different asset classes (equity, commodities, FX, ..)
- Different time-scales (daily to intraday to high-frequency)

Use synthetic data to solve financial problems[2]

- Scenario generator
- Reinforcement Learning
  - Train trading strategies on artificial data
  - Evaluate them on real data
- Explainable Artificial Intelligence[3]
  - Explain behaviour of neural networks
  - Accomodate unforeseen data

Improve data generation

- Learn desired features of the data based on applications
- Privacy considerations
- Transfer learning

[1] EU H2020 Fintech Topic: ICT-35-2018
EU H2020 COST CA19130: Fintech and AI in Finance

[2] Hirsa, Ali, Joerg Osterrieder, Branka Hadji Misheva, Wenxin Cao, Yiwen Fu, Hanze Sun and K. Wong. "The VIX index under scrutiny of machine learning techniques and neural networks." (2021).

[3] Misheva, Branka Hadji, Joerg Osterrieder, Ali Hirsa, Onkar P. Kulkarni and S. Lin. "Explainable AI in Credit Risk Management." *ArXiv* abs/2103.00949 (2021)

Conclusion and outlook

- Generating synthetic financial data is achievable and viable with the use of GANs
- Use of synthetic data is gaining traction and new applications
- Stabilizing training is still open to improvements
- Lack of unified quantitative metric still a problem

- A new GAN, based on SAGAN, to generate synthetic financial data, was proposed

- RegGAN: New generation of GANs; Use a second discriminator as a classifier; for controlling different statistical properties of time-series such as heavy-tailedness, skewness and autocorrelation, or for introducing no-arbitrage constraints (e.g. for a vola surface)
- Outlook: Apply GANs in the frequency-domain, combine Quant-GAN with Reg-GAN

1. Katie Martin. Flash crash — the trading savant who crashed the us stock market. *Financial Times*, May 2020.
2. Christopher B Barrett. Market analysis methods: are our enriched toolkits well suited to enlivened markets? *American journal of agricultural economics*, 78(3):825–829, 1996.
3. Shuntaro Takahashi, Yu Chen, and Kumiko Tanaka-Ishii. Modelling financial time-series with generative adversarial networks. *Physica A: Statistical Mechanics and its Applications*, 527, Apr 2019.
4. Rama Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 2001.
5. Francesco Rundo, Francesca Trenta, Agatino Luigi di Stallo, and Sebastiano Battiato. Machine learning for quantitative finance applications: A survey. *Applied Sciences*, 9(24):5574, 2019.
6. Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.
7. Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.
8. Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks in computer vision: A survey and taxonomy. *arXiv preprint arXiv:1906.01529*, 2019.
9. Xingyu Zhou, Zhisong Pan, Guyu Hu, Siqi Tang, and Cheng Zhao. Stock market prediction on high-frequency data using generative adversarial nets. *Mathematical Problems in Engineering*, 2018, 2018.
10. Adriano Koshiyama, Nick Firoozye, and Philip Treleaven. Generative adversarial networks for financial trading strategies fine-tuning and combination. *arXiv preprint arXiv:1901.01751*, 2019.
11. Giovanni Mariani, Yada Zhu, Jianbo Li, Florian Scheidegger, Roxana Istrate, Costas Bekas, and A Cristiano I Malossi. Pagan: Portfolio analysis with generative adversarial networks. *arXiv preprint arXiv:1909.10578*, 2019.
12. Gautier Marti. Corrgan: Sampling realistic financial correlation matrices using generative adversarial networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8459–8463. IEEE, 2020.
13. Teema Leangarun, Poj Tangamchit, and Suttipong Thajchayapong. Stock price manipulation detection using generative adversarial networks. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2104–2111. IEEE, 2018.
14. Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. Quant gans: Deep generation of financial time series. *arXiv:1907.06673v2 [q-fin.MF]*, Dec 2019.
15. Cem Dilmegani. The ultimate guide to synthetic data in 2021, Jun 2021.
16. Financial Conduct Authority. *Supporting innovation in financial services: the digital sandbox pilot*. FCA, Apr 2021.
17. James Montantes. Deep learning in finance: Is this the future of the financial industry?, Jul 2020.
18. Samuel Assefa. Generating synthetic data in finance: opportunities, challenges and pitfalls. *Challenges and Pitfalls (June 23, 2020)*, 2020.
19. Pratyush Muthukumar and Jie Zhong. A stochastic time series model for predicting financial trends using nlp. *arXiv preprint arXiv:2102.01290*, 2021.
20. Weijie Wu, Fang Huang, Yidi Kao, Zhou Chen, and Qi Wu. Prediction method of multiple related time series based on generative adversarial networks. *Information*, 12(2):55, 2021.

21. Victor Storchan, Tucker Balch, and Svitlana Vyetrenko. Mas-gan: Adversarial calibration of multi-agent market simulators. Paper under review, 2021.
22. Yerin Kim, Daemook Kang, Mingoo Jeon, and Chungmok Lee. Gan-mp hybrid heuristic algorithm for non-convex portfolio optimization problem. *The Engineering Economist*, Jun 2019.
23. He Sun, Zhun Deng, Hui Chen, and David C Parkes. Decision-aware conditional gans for time series data. *arXiv preprint arXiv:2009.12682*, 2020.
24. Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. *NeurIPS 2019*, 2019.
25. Fernando De Meer Pardo. Enriching financial datasets with generative adversarial networks, Aug 2019.
26. Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.
27. Dmitry Efimov, Di Xu, Luyang Kong, Alexey Nefedov, and Archana Anandakrishnan. Using generative adversarial networks to synthesize artificial financial datasets. *arXiv preprint arXiv:2002.02271*, 2020.
28. Hao Ni, Lukasz Szpruch, Magnus Wiese, Shujian Liao, and Baoren Xiao. Conditional sig-wasserstein gans for time series generation. *arXiv preprint arXiv:2006.05421*, 2020.
29. Akhil Sethia, Raj Patel, and Purva Raut. Data augmentation using generative models for credit card fraud detection. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pages 1–6. IEEE, 2018.
30. Zhaohui Zhang, Lijun Yang, Ligong Chen, Qiuwen Liu, Ying Meng, Pengwei Wang, and Maozhen Li. A generative adversarial network–based method for generating negative financial samples. *International Journal of Distributed Sensor Networks*, 16(2):1550147720907053, 2020.
31. Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
32. Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Interna- tional conference on machine learning*, pages 214–223. PMLR, 2017.
33. Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.
34. Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
35. Gautier Marti. corrgan.io, 2019.
36. Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.
37. Frank Xu. Building your own self-attention gans, Jul 2020.
38. FCA and City of London. Digital sandbox pilot, 2021.
39. TJ Horan. Credit card fraud: It’s still a thing (and as big as ever), Jan 2021.
40. David Love, Nalin Aggarwal, Alexander Statnikov, and Chao Yuan. An automated system for data attribute anomaly detection. In *KDD 2017 Workshop on Anomaly Detection in Finance*, pages 95–101. PMLR, 2018
41. Cao, Haoyang and Xin Guo. “Generative Adversarial Network: Some Analytical Perspectives.” ArXiv abs/2104.12210 (2021): n. pag.