DATA-DRIVEN PORTFOLIO DRAWDOWN OPTIMIZATION WITH GENERATIVE MODELING

Literature Review
Update 27-01-2022

GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

InvestSuite

GHENT UNIVERSITY

CONTENT

- INTRODUCTION
- POTENTIAL CONTRIBUTIONS
- LITERATURE OVERVIEW
- MAIN METHODOLOGIES USED
- NEXT STEPS

InvestSuite

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

INTRODUCTION

INTRODUCTION

- **Financial Machine Learning**

- Discriminative vs. Generative

- Generative vs. Monte Carlo

- Drawdowns vs. Returns

GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

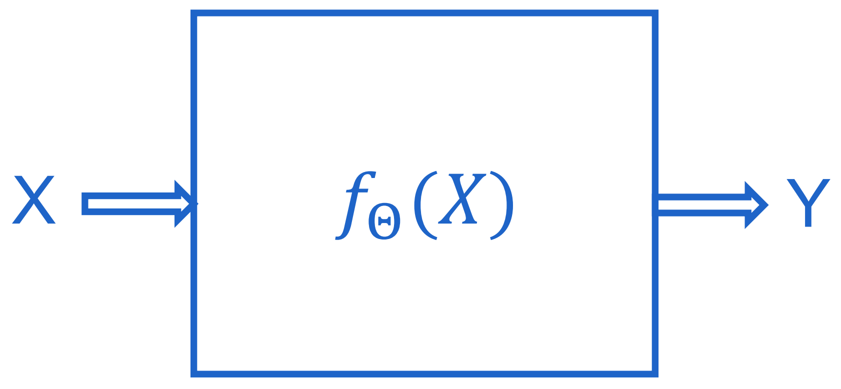
DISCRIMINATIVE VS. GENERATIVE ML

Discriminative ML

- Revolves around conditional $\mathbb{P}(Y|X)$, learn set of parameters Θ from the data to predict labels Y given a distribution of features X.

- Given some $Y: \mathbb{R}^M$, $X: \mathbb{R}^N$, with N typically large, learn Θ using a flexible mapping f: $f_\Theta(X): \mathbb{R}^N \to \mathbb{R}^M$ such that some $\mathcal{L}(Y, f_\Theta(X))$ is minimized.

Examples include simple regularized regressions (LASSO, Ridge, Elastic nets), support vector machines (SVM) and neural network (NN) regressors.



$min_\Theta\ \mathcal{L}(Y, f_\Theta(X))$

DISCRIMINATIVE VS. GENERATIVE ML

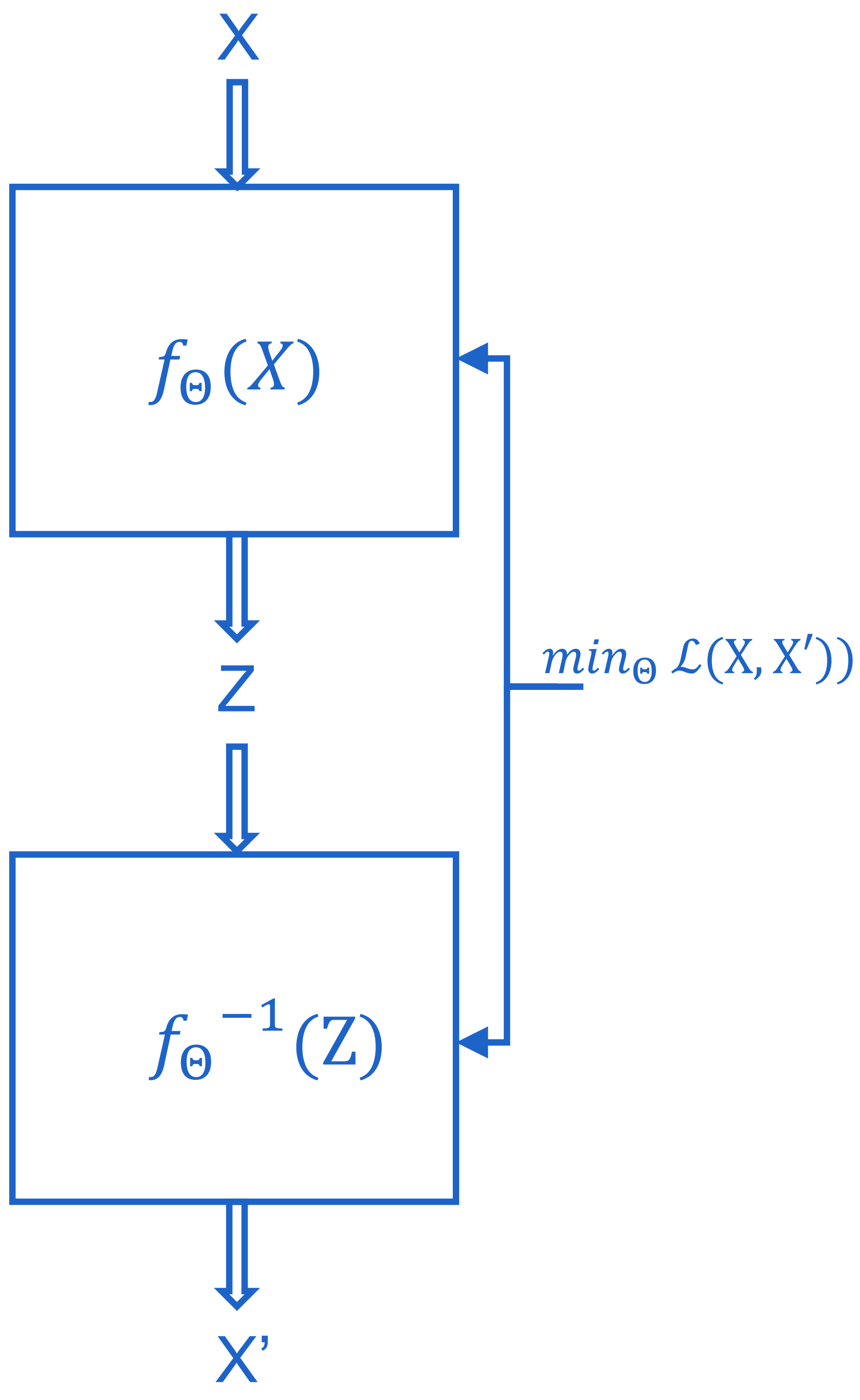
Generative ML

- Revolves around unconditional distribution $\mathbb{P}(X)$, learn Θ to capture structure/symmetries in (high-dimensional) $\mathbb{P}(X)$; Goal: compress the data in much fewer dimensions, while preserving the important features of the original data.
- Given some $X: \mathbb{R}^N$, with N typically large, learn $\Theta,$ using a flexible mapping f on some space $Z: \mathbb{R}^K$, with $K << N$, called a representation,

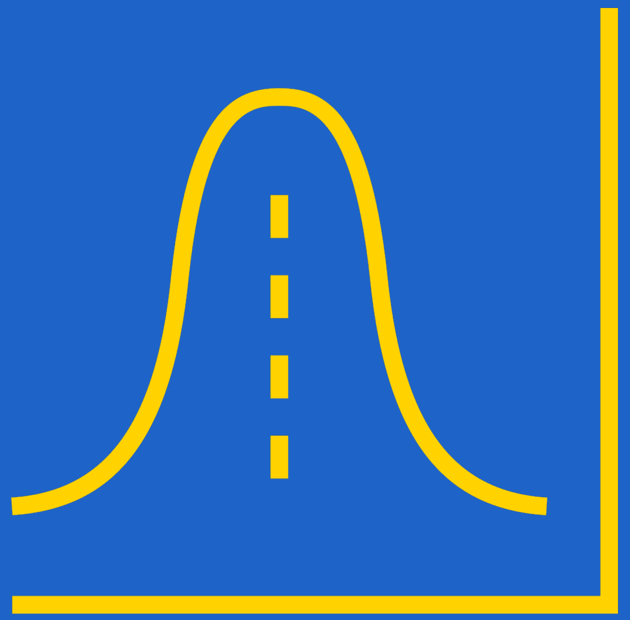
$$f_\Theta(X): \mathbb{R}^N \to \mathbb{R}^K: X \to Z.$$

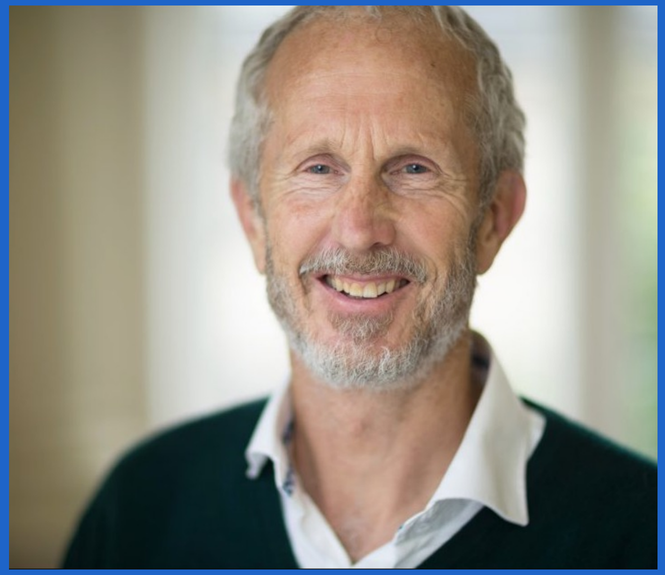
- Mapping $f_\Theta^{-1}(Z): \mathbb{R}^K \to \mathbb{R}^N: Z \to X'$ can be used for sampling new samples X', such that X and X' are not distinguishable statistically according to some loss metric $\mathcal{L}(X, X')$.

Examples include variational autoencoders (VAE), generative adversarial networks (GAN), restricted Boltzmann machines (RBM), and flow-based / normalizing flows (NF).



GHENT UNIVERSITY
FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION





“ SCENARIO-BASED SCIENCE IS MAYBE THE BEST WE CAN DO WHEN DEALING WITH COMPLEX SYSTEMS. ”

DOYNE FARMER

GENERATIVE ML VS. MONTE CARLO

- Finding an optimal mapping between a source distribution Z and the original data $\mathbb{P}(X)$ is not a new problem in finance.

- This has been a key area of research in **Monte Carlo** and the development of **bottom-up stochastic processes**.

- This has been **instrumental** in calibrating risk measures and optimizing portfolios under the **physical measure** $\mathbb{P}$, but **crucial** in constructing **derivative pricing tools** under the **risk-neutral measure** $\mathbb{Q}$.

The core difference with the machine learning approach, is that in a traditional Monte Carlo the map $f_{\Theta}^{-1}(X)$ has to be specified a priori (before estimation/calibration) as some closed-form system of equations called the **data generating process (DGP)**.

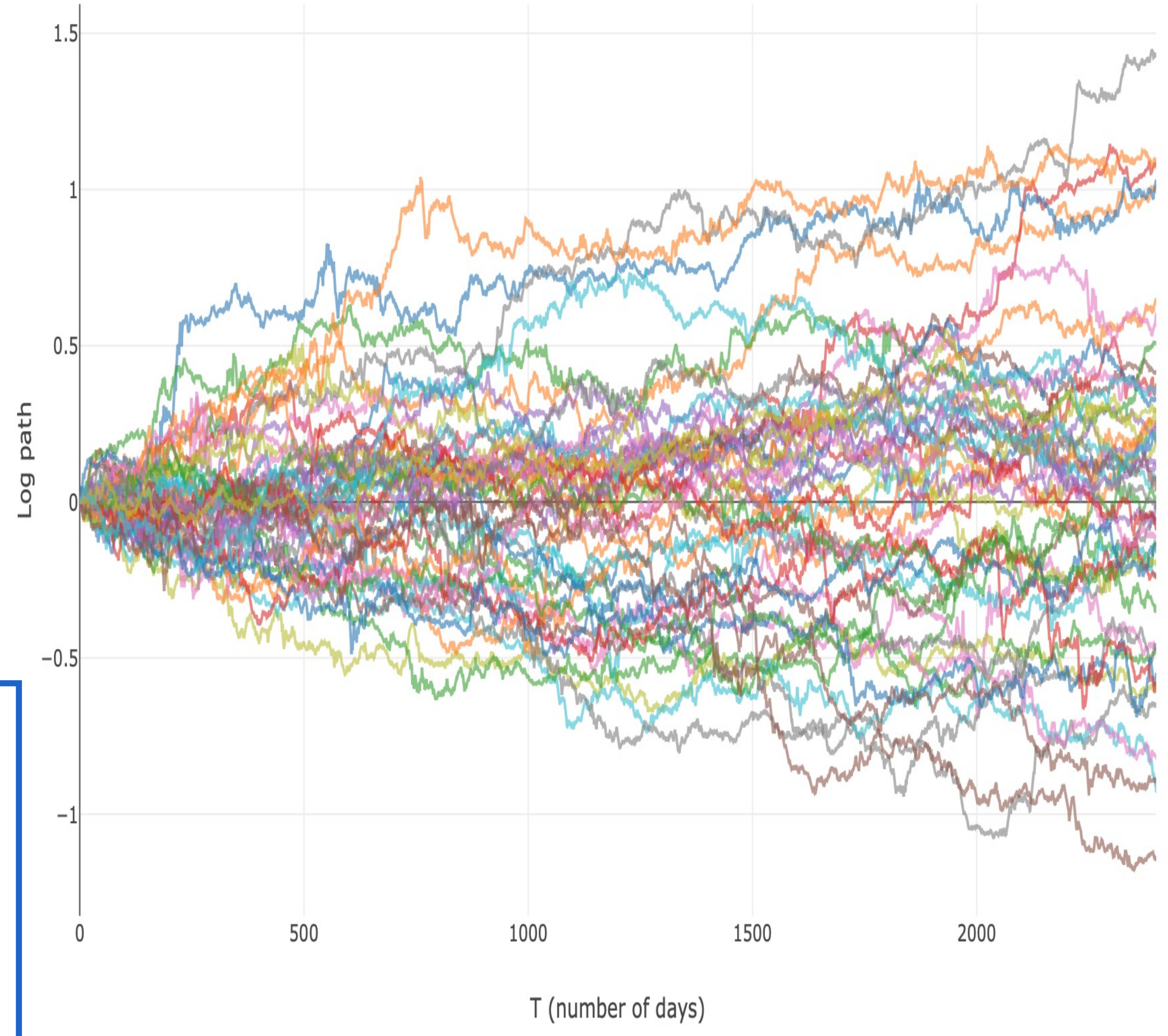
GENERATIVE ML VS. MONTE CARLO

- Arguably the most well-known process is the Black-Scholes model that describes the diffusion paths of asset prices as geometric Brownian motions.

- In this example, $Z \sim N(0,1)$ and Θ is a tuple of the drift and volatilities (μ, σ) such that the corresponding market generator becomes:

$$f_{\Theta}^{-1}(Z) \colon X_t = \mu + \sigma \epsilon_t$$

where X_t is the logreturn at t, $\Theta = (\mu, \sigma)$, and ϵ_t is an instance of Z at t. Remark that $\mu = r$, the risk-free rate under $\mathbb{Q}$.



The second difference is that such an a priori specified $f_{\Theta}^{-1}(Z)$ does not require the estimation of $f_{\Theta}(X)$ and the evaluation of $\mathcal{L}(X, X')$, but rather relies on estimating Θ directly using some form of **loglikelihood maximization** on historical data (called ***calibration***), while the search for the optimal Θ in the DGP-free approach is called ***learning*** or ***training***.

DRAWDOWNS VS. RETURNS

- **Paths**:
  $\gamma: [0, T] \to \mathbb{R}^D, \gamma = \{\gamma^1, \gamma^2, \ldots, \gamma^D\}$

- **Logreturn** and **autocorrelation**:

  $X_i(t, \Delta t) = ln\big(S_i(t + \Delta t)\big) - \ln(S_i(t))$
  $corr(X_i(t + \tau, \Delta t), X_i(t, \Delta t))$

- *Time-augmented* **return path**:

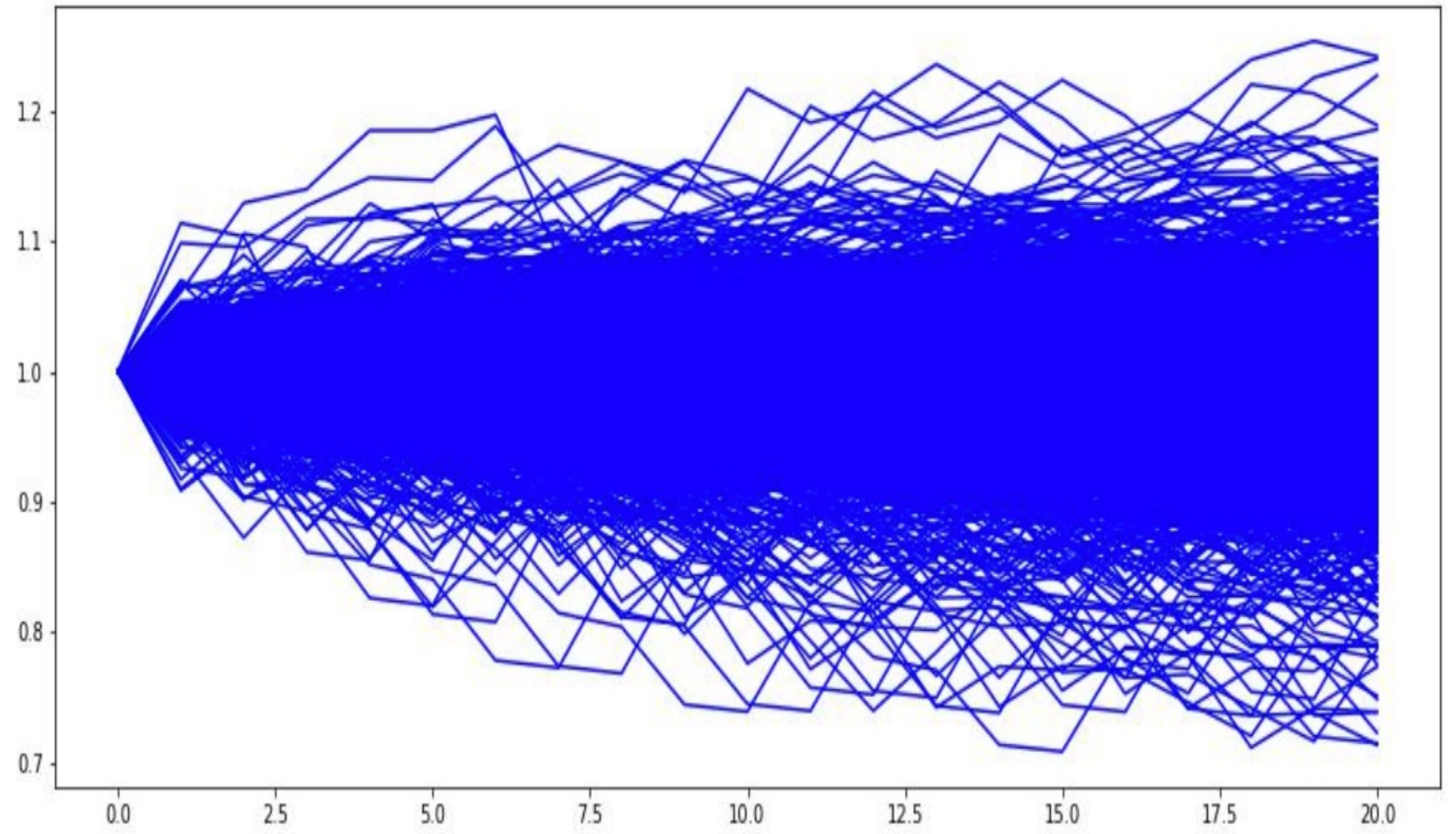
  $r_i: [0, T] \to \mathbb{R}^2, r_i = \{t, (X_i(0, \Delta t), X_i(1, \Delta t), X_i(T, \Delta t))\}$

- **Return space**:

$$R_j: [0, T] \to \mathbb{R}^{D+1}, R_j = \{t, r_1, r_2, \ldots, r_D\}$$

- T < N_{obs}, $N_{sim} = \lfloor N_{obs}/T \rfloor$ non-overlapping or $N_{sim} = N_{observations} - T$ overlapping **return sequences** (i.e. scenarios or simulations):

$$R = (R_1, R_2, \ldots, R_{N_{sim}})$$



Paths of spot asset price S

DRAWDOWNS VS. RETURNS

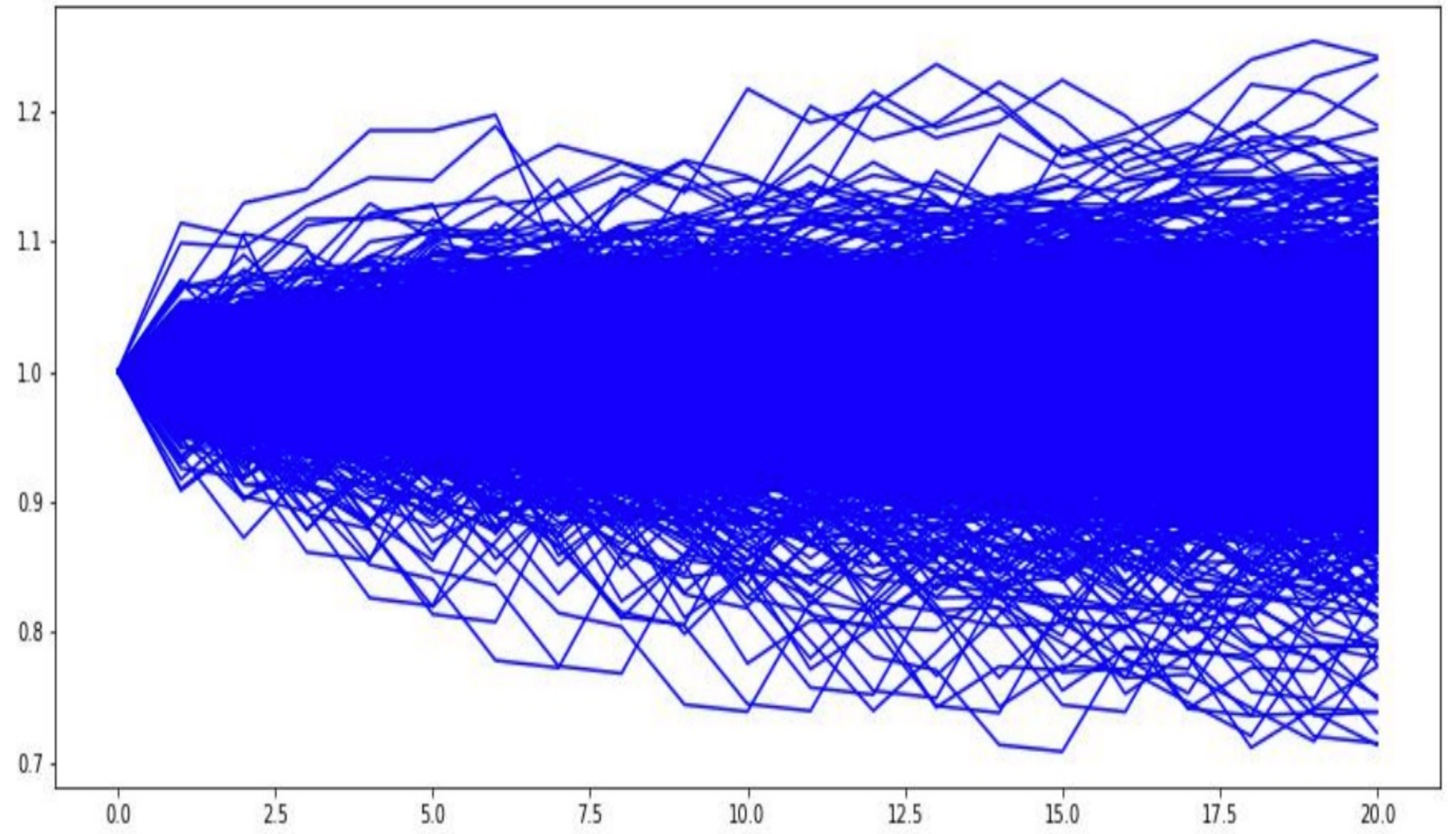
- Traditionally stats for μ_i and σ_i (or ρ_i, possibly r^*) estimated from R

- **Weighted simulation $\boldsymbol{w_j}$** :

  - **EWMA**: exponentially decreasing w_j to smaller j
  - **Conditional sampling**: attach $w_j = 0$ to sequences not satisfying the historical *conditions*, and $w_j = 1$ if they do
  - **Volatility-filtered** sampling: $w_j = \frac{\sigma}{\sigma_j}$

- **No estimation of stats** (non-parametric – "Estimate Nothing"):

  - Use R outright (Naive *historical simulation*)
  - Resample using random indices in j in {1,…,N_{sim}} *with replacement* (= non-parametric (w_j-weighted) *block bootstrap* with block size T)



Paths of spot asset price S

DRAWDOWNS VS. RETURNS

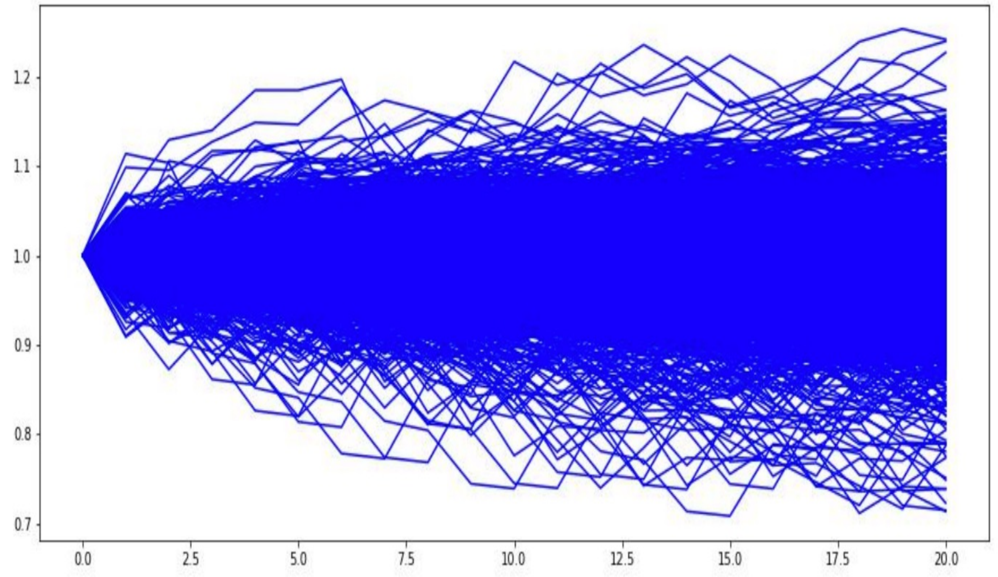
- **Stylized facts of financial returns** (surveyed by Cont 2001), most notably:

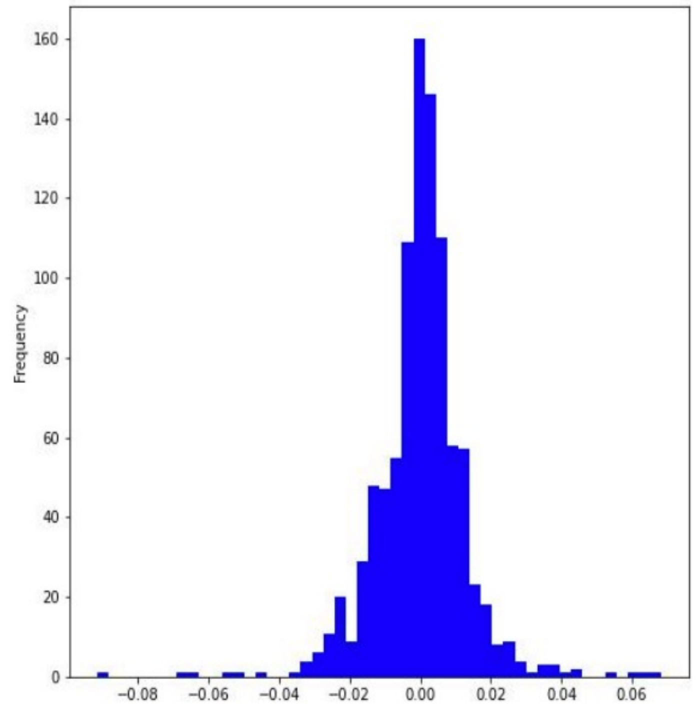
  (1) the existence of **fat tails** in the return distribution,
  (2) the **absence of linear autocorrelation** (cf. above),
  (3) **volatility clusters** (large *absolute* returns are highly autocorrelated),
  (4) **leverage effects** (*absolute* returns and returns are negatively correlated).

  Much of the work regarding stochastic DGPs discussed above come down to (explicitly) addressing these stylized facts!

- R often viewed from its ***return distribution*** **(P&L)** right away
  - **Static !!! Not a path.**
  - Once decided on Δt estimates of μ_i, σ_i and $r*$ invariant to sequence shifts, as well as popular *risk conditionals* on the P&L distribution such as **value-at-risk (VaR)** and **expected shortfall (ES).**

While path characteristics matter, even for returns R!
E.g. monofractal **scaling** of properties of risk (i.e. risk $\propto \Delta t$)
Valuable information about the sequential structure, i.e. the path structure, is lost.





$P\&L$: $\mathbb{P}$(X)

GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

DRAWDOWNS VS. RETURNS

- **Drawdown paths**:

$$x_i(t, \Delta t) = \max_{t_k < t}(S_i(t_k)) - S_i(t)$$

$$\xi_i : [0, T] \to \mathbb{R}^2, \xi_i = \{t, (x_i(0, \Delta t), x_i(1, \Delta t), x_i(T, \Delta t))\}$$

= Dynamic generalization of a deviation measure on the path space (Chekhlov, 2005)

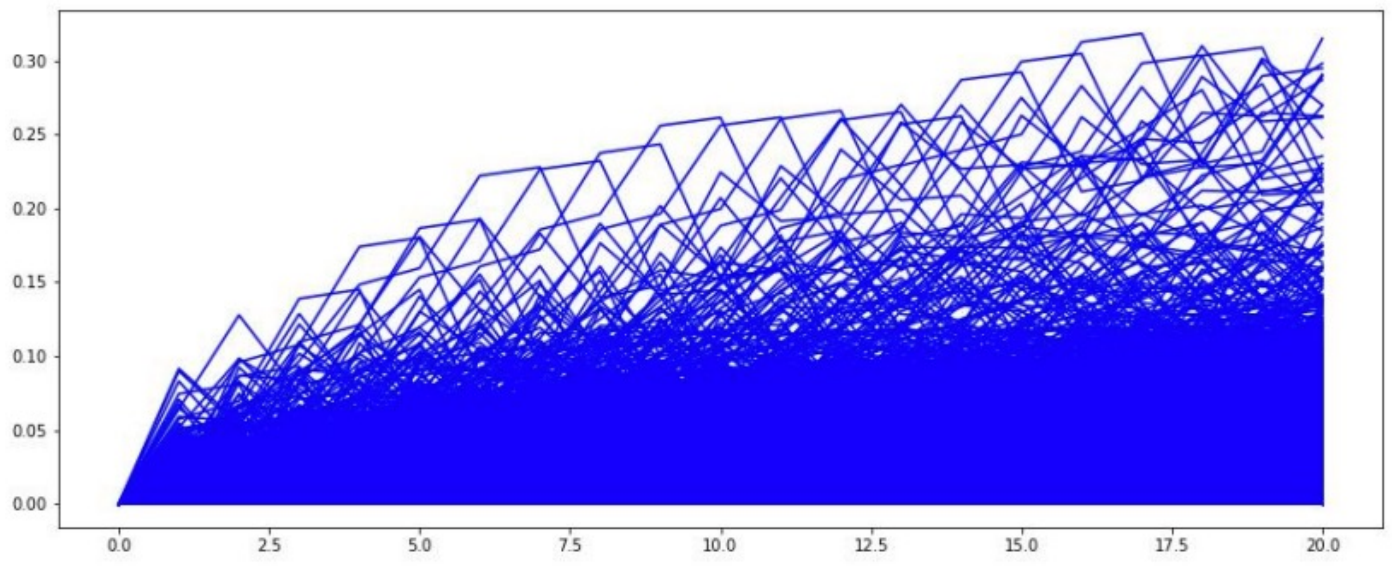
- **Drawdown space:**

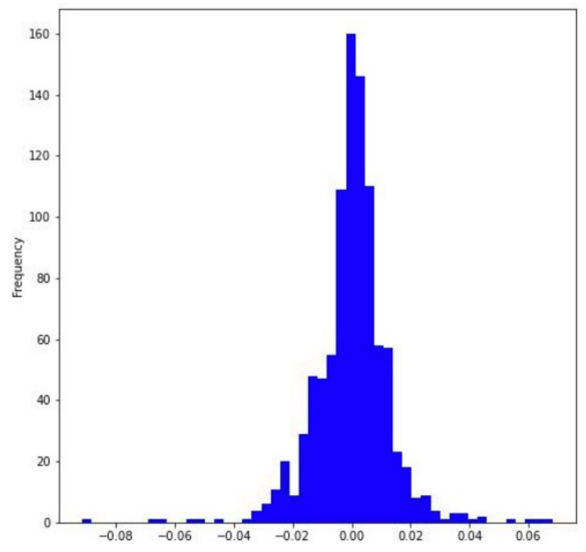
$$\Xi_j : [0, T] \to \mathbb{R}^{D+1}, \Xi_j = \{t, \xi_1, \xi_2, \ldots, \xi_D\}$$

- **Drawdown sequences** ($T < N_{obs}$):

$$\Xi = (\Xi_1, \Xi_2, \ldots, \Xi_{N_{sim}})$$

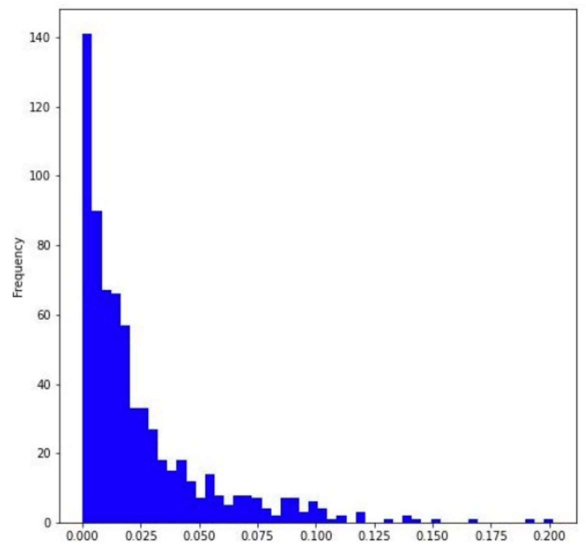
Drawdown paths





$P\&L$: $\mathbb{P}$(X)

('flat' return distribution)



$\mathbb{P}$(x)

('flat' drawdown distribution)

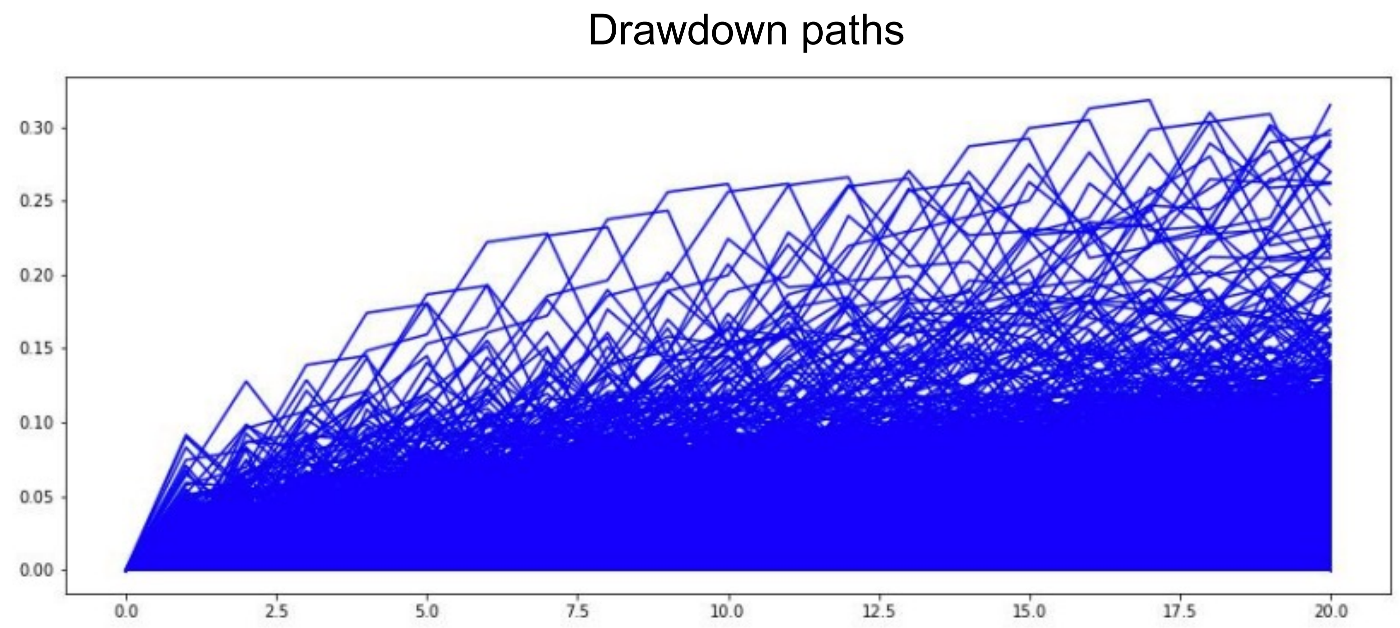
DRAWDOWNS VS. RETURNS

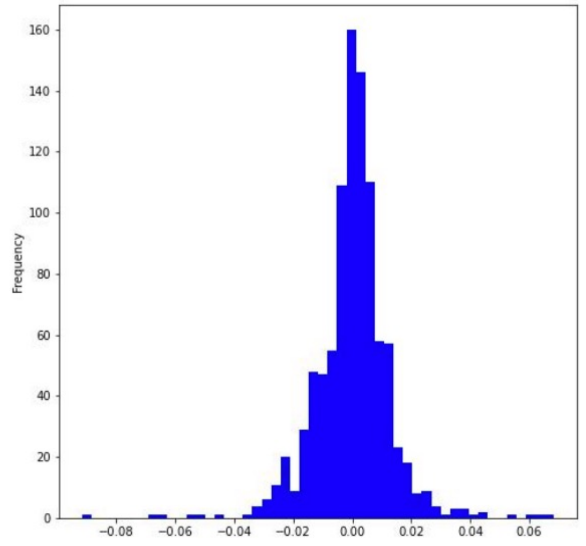
- **Challenges** when modeling (conditional) drawdown sequences and *expected* drawdown (optimization) loyal to historical sample:

  - **Drawdown sequences** Ξ have important path structure
    => Match the distribution in the *path space*, not just the flat x distribution (while for R this is synonymous in 99.9% of applications)

  - **Stochastic processes** have not been developed for ξ-processes. There are no off-the-shelf DGPs for these processes, nor stylized facts proposed or agreed on.

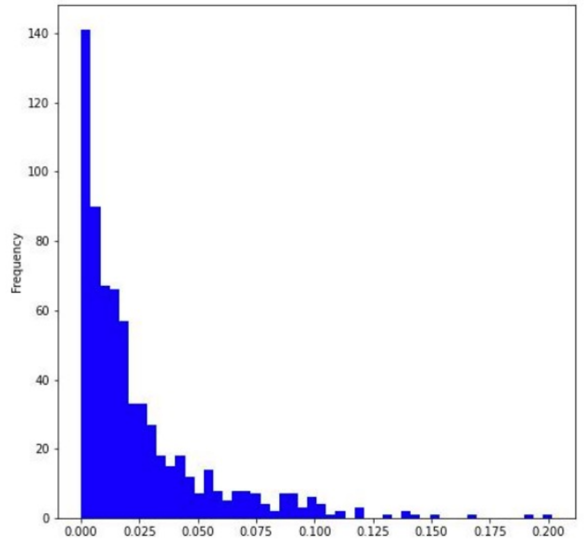
- **Possible answers**:
  - What does it mean to compare distributions in the *path space*, i.e. comparing random variables versus sequential random variables? See below implications for on *signatures* and the sequential *signature kernel*.
  - To leapfrog the lack of DGPs, one could use DGP-free modeling (if paths are sufficiently realistic)





$P\&L$: $\mathbb{P}(X)$

('flat' return distribution)



$\mathbb{P}(x)$

('flat' drawdown distribution)

PORTFOLIO DRAWDOWN OPTIMIZATION

Naive drawdown optimization:

$$\begin{aligned} \min_{w} \quad & \mathbb{E}_j(\xi(w)) \\ \text{s.t.} \quad & \xi_j = w\Xi_j \\ & w\mathbf{I}^D = 1 \end{aligned}$$

Portfolio drawdown optimization:

$$\begin{aligned} \min_{w} \quad & \mathbb{E}_j(\xi(w)) \\ \text{s.t.} \quad & \xi_{j,t} = m_{j,t} - w\Pi_{j,t} \\ & m_{j,t} \geq m_{j,t-1} \\ & w\mathbf{I}^D = 1 \end{aligned}$$

Π is a space of price paths that has a *correspondance* to Ξ.

For now it is clear that the path structure is critical because of local maxima m. Not preserved when modeling R, crucial *path feature* in Ξ.

$T = N_{obs}$

T = N_{obs} = 2609, N_{sim} = 1

Example: For 1 scenario the (unconditional) *expected* drawdown over j, $\mathbb{E}_j$, is just the average historical drawdown (light blue area)

PORTFOLIO DRAWDOWN OPTIMIZATION

Naive drawdown optimization:

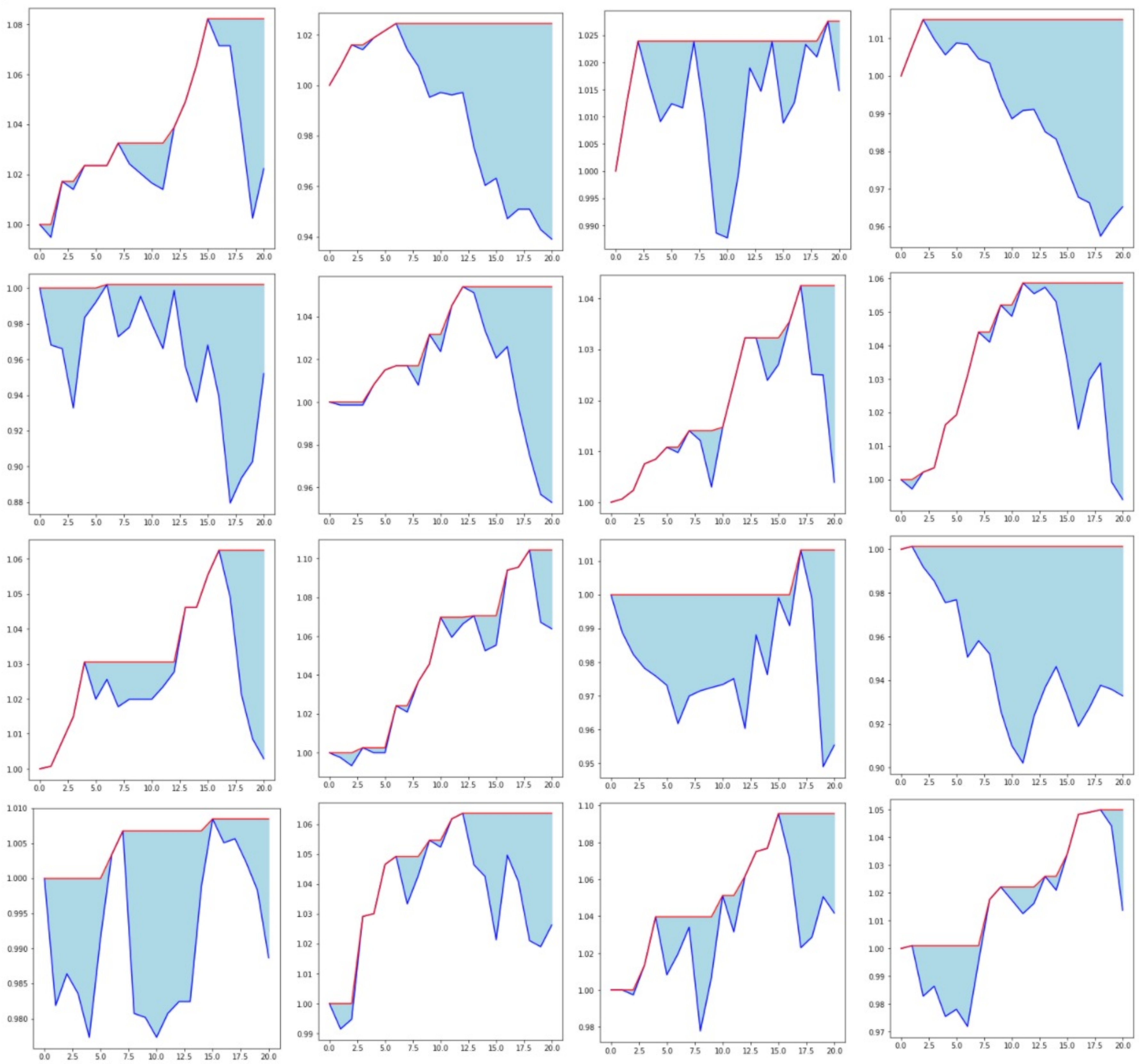
$$\begin{aligned} \min_{w} \quad & \mathbb{E}_j(\xi(w)) \\ \text{s.t.} \quad & \xi_j = w\Xi_j \\ & w\mathbf{I}^D = 1 \end{aligned}$$

Portfolio drawdown optimization:

$$\begin{aligned} \min_{w} \quad & \mathbb{E}_j(\xi(w)) \\ \text{s.t.} \quad & \xi_{j,t} = m_{j,t} - w\Pi_{j,t} \\ & m_{j,t} \geq m_{j,t-1} \\ & w\mathbf{I}^D = 1 \end{aligned}$$

Example for 16 random scenarios Π_j (blue line), m_j (red line) and ξ_j (light blue area).

$T = 20$



16 random j,
for j in {1, …, 2589}

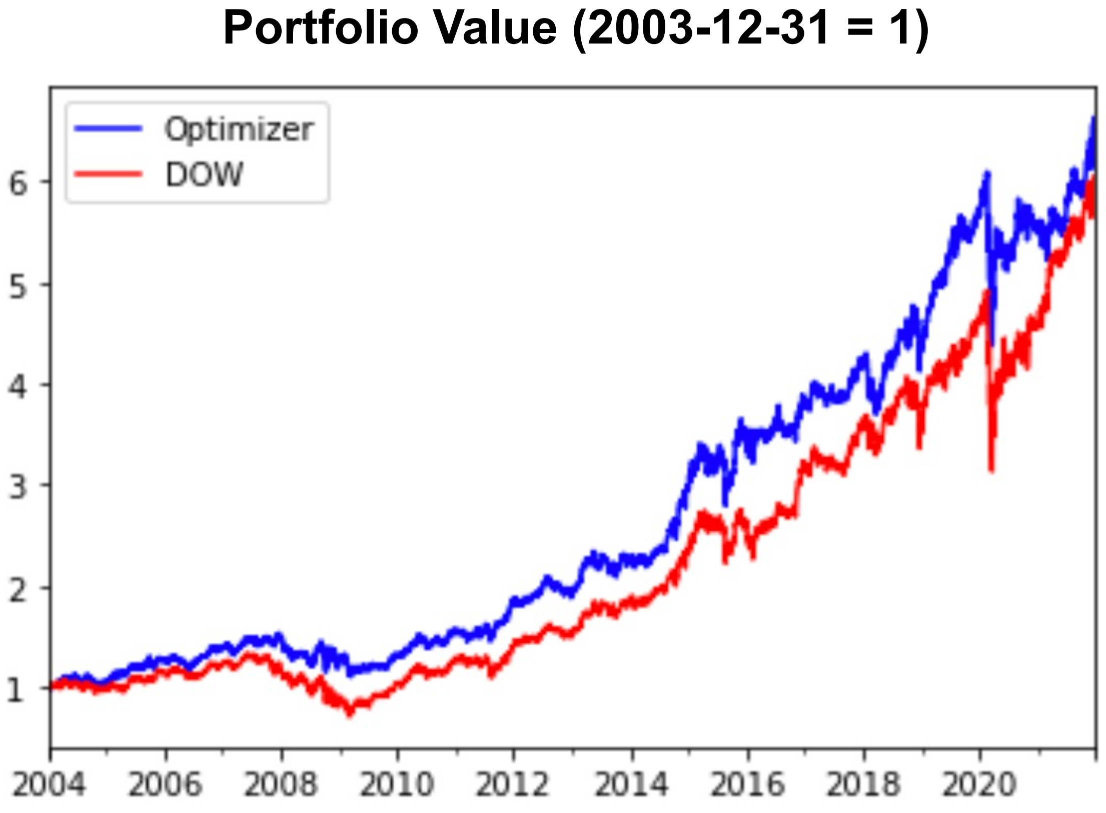
$w_j = 1, \forall j$

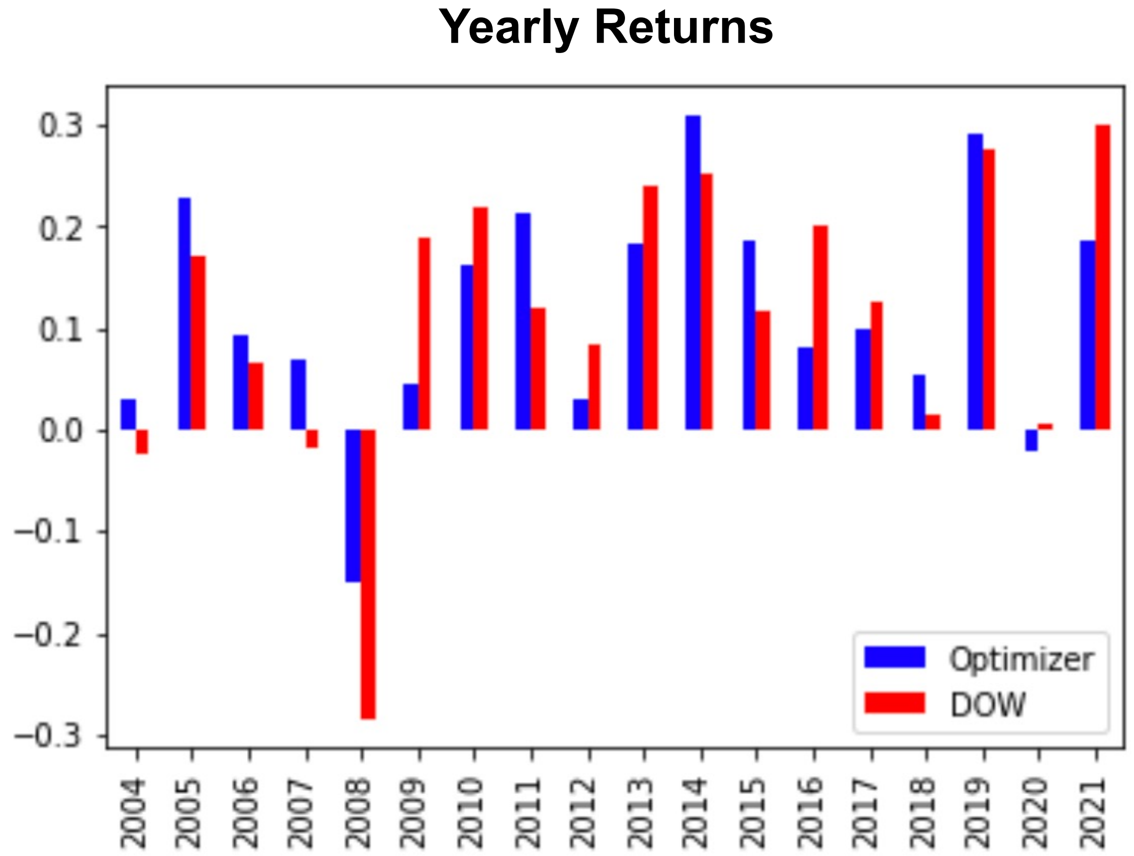
$N_{obs} = 2609, N_{sim} = 2589$, T = 20

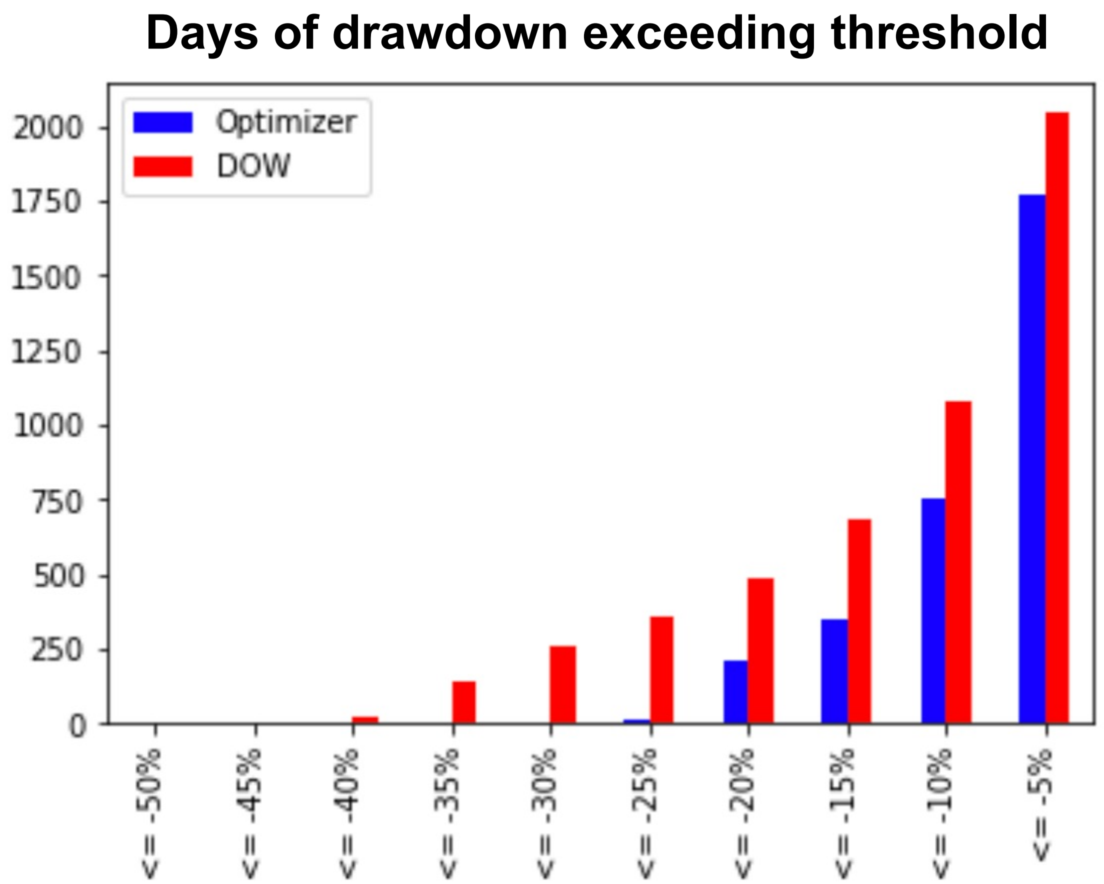
GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

EXAMPLE: DOW 30

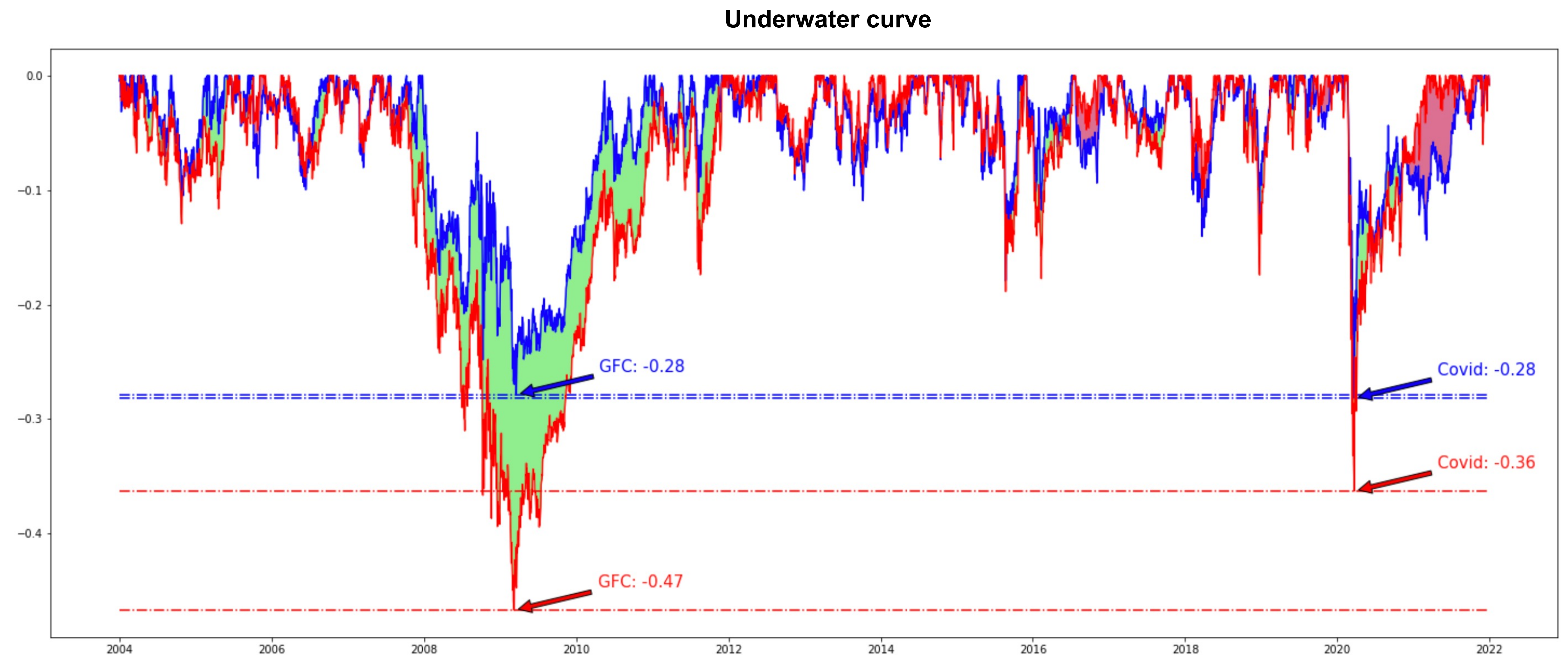






- Example backtest **DOW30**, point-in-time universe with no lookahead information
- Simple exponential weighted *j*, block bootstrap historical simulation (monthly paths).
- Most notable feature: drawdown reduction. Figure on the right denotes the number of days (y-axis) where a certain drawdown threshold (x-axis) was exceeded.

EXAMPLE: DOW 30



GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

POTENTIAL CONTRIBUTIONS

POTENTIAL CONTRIBUTIONS

- **Input representation** for **Portfolio Optimization:** explore use of generative ML for portfolio optimization (not the focus in earlier studies!); relevant *path features* (e.g. drawdown structure for drawdown optimization) necessitates apt *input representation* (Ξ vs. R),

- **Loss metric:** focus on reproducing drawdown structure after dimension reduction, i.e. construct non-linear common factors in the downside risk of the investible universe (vs. traditional return / volatility decomposition)

Geometric Priors: financial/economic prior on path features, e.g. drawdown, drift, vol, …

- **Conditional sampling:** match *non-stationary* features of financial time series by learning on the relevant market conditions; understand *sensitivities* of the optimal portfolio to these market conditions.

Economic Priors: financial/economic prior on factors, e.g. macro-economic conditions, …

GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

LITERATURE OVERVIEW

LITERATURE OVERVIEW

- **Market Generator** =

generative models with the specificity of modelling financial markets

(such as spot asset prices, option prices and volatilities, or order streams in limit order books)

Paper	Year	Architecture	Application
Henry-Labordere [29]	2019	GAN	Option prices
Wiese et al. [30]	2019	GAN	Hedging strategies
Cuchiero et al. [31]	2020	GAN	Volatility models
Ni et al. [32]	2020	GAN	Spot prices
Wiese et al. [33]	2020	GAN	Spot prices
Li et al. [34]	2020	GAN	Order book simulation
Storchan et al. [35]	2020	GAN	Spot prices
Benedetti [36]	2020	GAN	Yield models
Xu et al. [37]	2020	GAN	Spot prices
Pardo and López [38]	2020	GAN	Spot prices
Buehler et al. [39]	2021	GAN	Hedging strategies
Ni et al. [40]	2021	GAN	Spot prices
Pfenninger et al. [41]	2021	GAN	Spot prices
Rosolia and Osterrieder [42]	2021	GAN	Spot prices
Koshiyama et al. [43]	2021	GAN	Spot prices
van Rhijn et al. [44]	2021	GAN	Spot prices
Marti et al. [45]	2021	GAN	Correlation matrices
Coyle et al. [46]	2021	GAN	Spot prices
Wiese et al. [47]	2021	NF	Spot and Option prices
Kondratyev and Schwarz [48]	2019	RBM	Spot prices
Lezmi et al. [49]	2020	RBM / GAN	Spot prices
Wang [50]	2021	RBM / VAE	Spot prices
Buehler et al. [51]	2020	VAE	Spot prices
Fung [52]	2021	VAE	Option prices
Frandsen [53]	2021	VAE	Hedging strategies
Bergeron et al. [54]	2021	VAE	Volatility models
Ning et al. [55]	2021	VAE	Volatility models

Table 1: Overview of the market generator literature

MAIN METHODOLOGIES

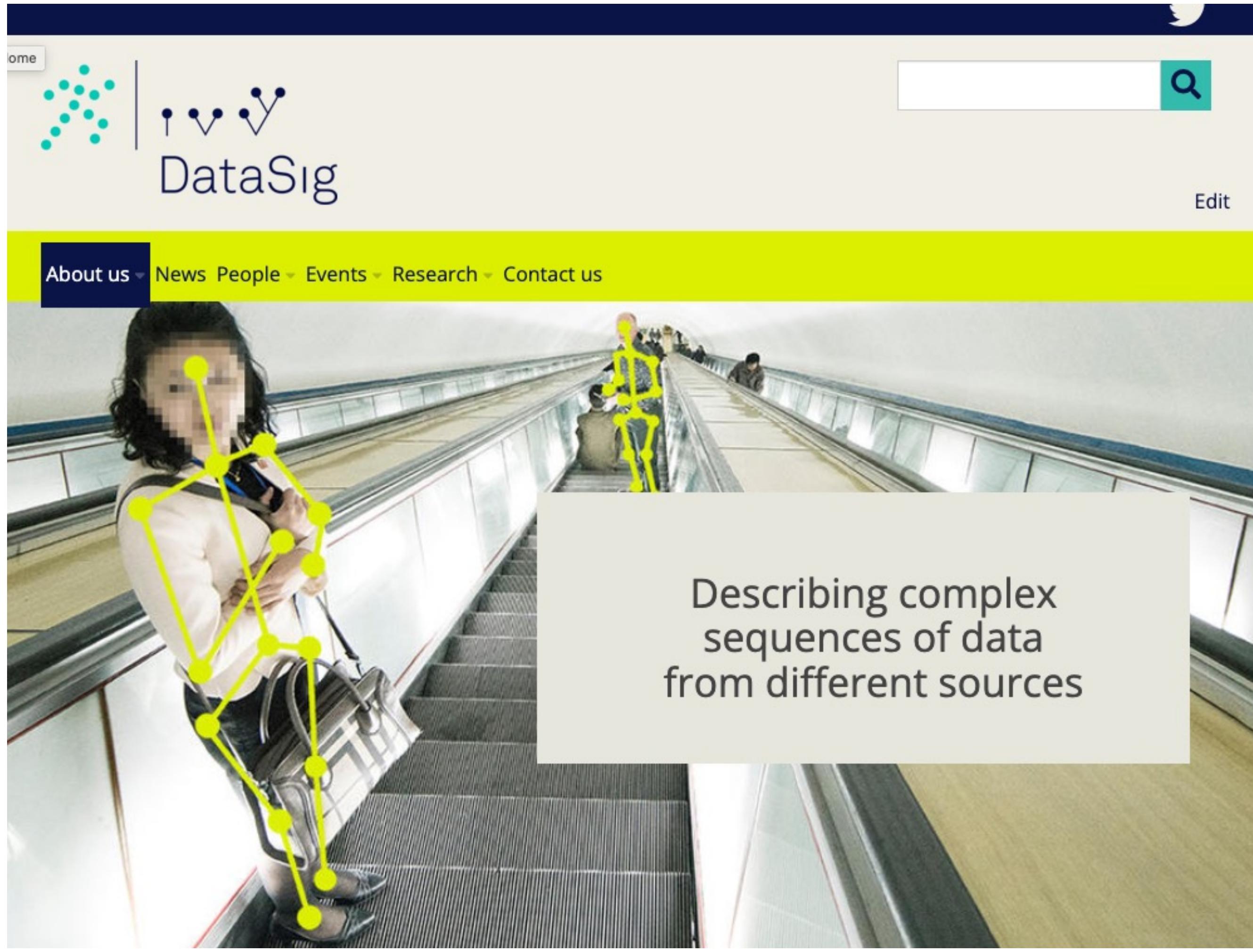
MAIN METHODOLOGIES (TECHNICAL PART)

- Signature-based MMD loss
- Generative ML architectures
- Detailed CVAE discussion
- Conditional sampling and explainable ML (XML)

SIGNATURE-BASED MMD LOSS

- **Signatures** = a *graded summary* of *path-structured data*, preserving important *geometrical features* of the path, with applications such as recognition of handwritten Chinese characters, classification of bipolar and borderline disorders, malware detection, detection of Alzheimer disease, human action recognition, and many more (see: datasig.ac.uk).

- Applications in finance include market simulation and optimal trade execution.



SIGNATURE-BASED MMD LOSS

– **Kernels 101**: kernels k are a class of functions of two random variables that measure the similarity between the two variables.
For instance:

$$k(X, Y): [a, b] \times [a, b] \rightarrow \mathbb{R}$$

is a kernel since it maps two random variables X and Y with support on $[a,b]$ on a metric that is (commonly) small when X and Y are close to each other, and vice versa.

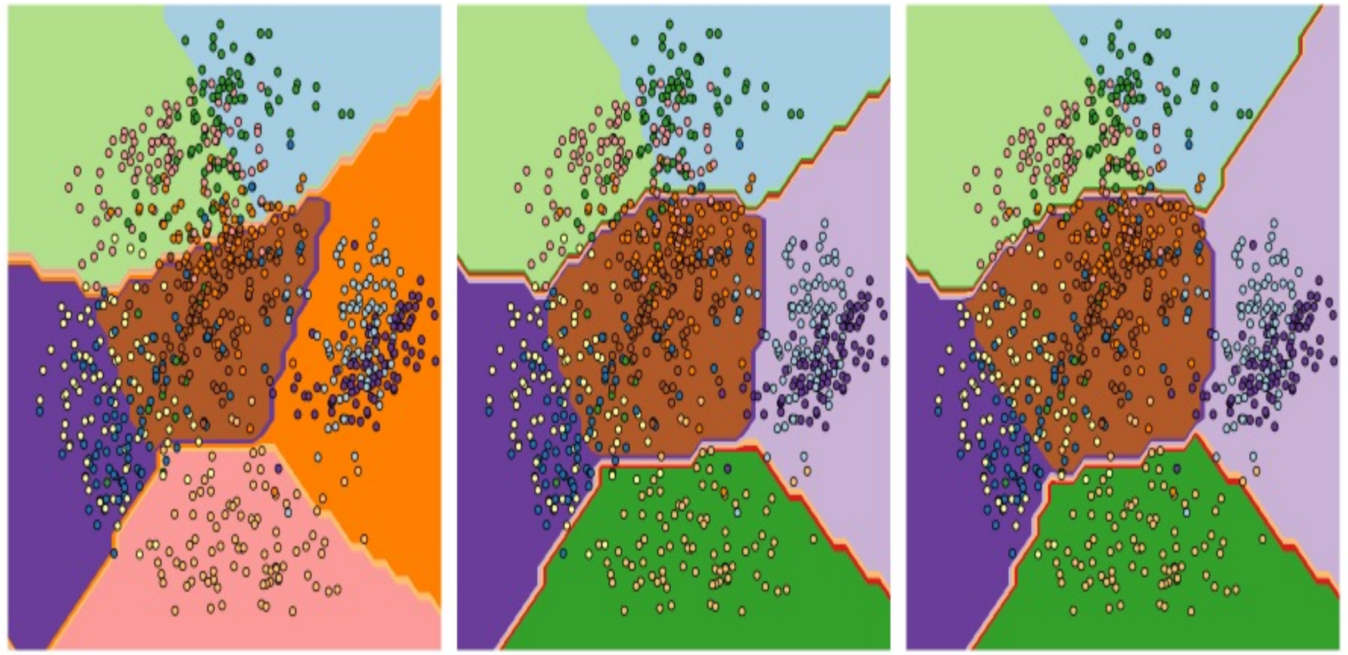
Examples: radial basis functions (RBF) such as the exponential, Fourier, Nystroem kernels and Gaussian, Euclidean, Polynomial kernels, …

Applications: most notably

(1) *feature maps* where kernels are essentially *inner products* between feature vectors X (which allows for using linear methods in non-linear problems, e.g. support vector machines),

(2) *basis functions* for approximation spaces (i.e. changing the basis of data to approximate functions by allowing more variation in regions with more data),

(3) and many more…



Kernel embeddings:
RBF (left), Fourier RBF (middle), Nystroem (right)
(source: Sklearn)

GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

SIGNATURE-BASED MMD LOSS

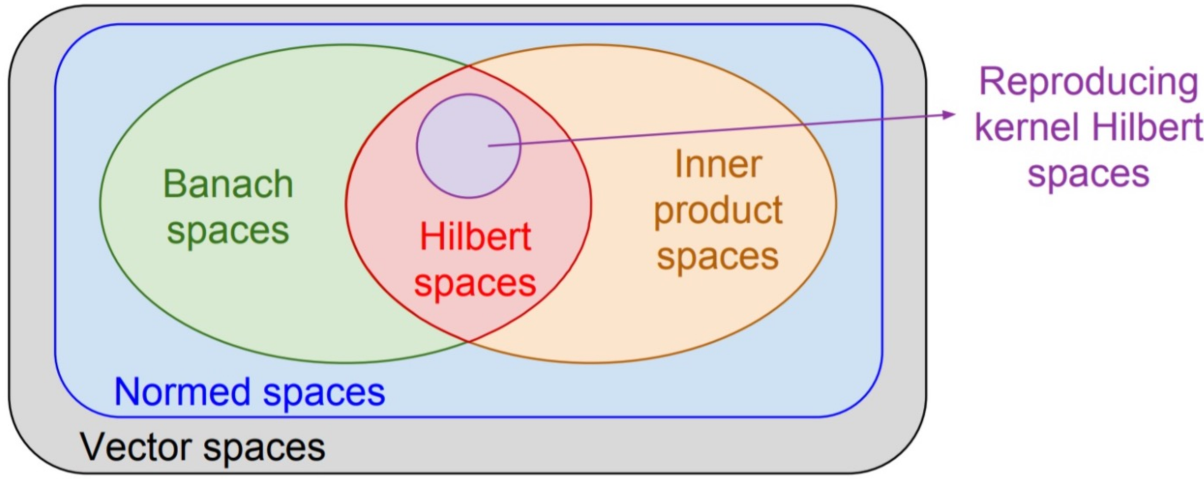
– **Positive definite kernels**, such as the Gaussian kernel, that satisfy

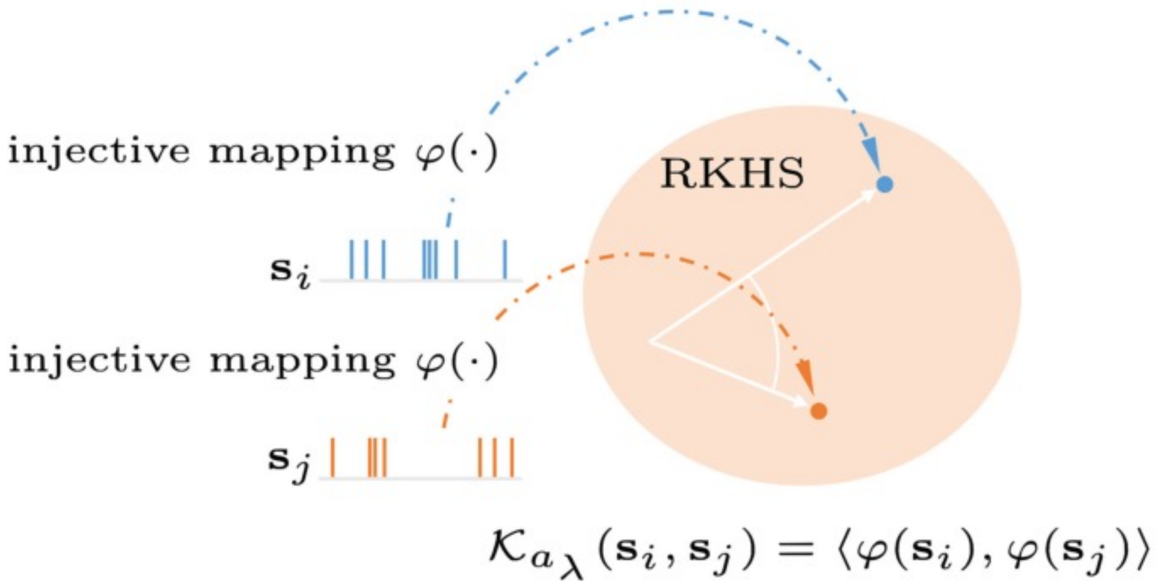
$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$$

for any x_i in X and any pair $c_i, c_j \in \mathbb{R}$, also called Mercer kernels have the property that there exists a mapping ϕ between X and Y and a space $\mathcal{H}$ equipped with an inner product, such that the kernel value $k(x, y)$ can be rewritten as an inner product in $\mathcal{H}$:

$$k(x, y) = \langle \phi(x), \phi(y) \rangle$$

– Since $\mathcal{H}$ should be equipped with an inner product it is a so-called Hilbert space, and it *reproduces* the kernel by means of that inner product of two mapped features $\phi(.)$. This is known as **a reproducing kernel Hilbert space (RKHS)** in machine learning.





GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

SIGNATURE-BASED MMD LOSS

- **Maximum mean discrepancy:** a popular measure of distance between two distributions in machine learning. Suppose we have two sets of samples X and Y and we want to measure the distance between them. The following MMD computes the mean squared difference of the statistics ϕ between the two sets

$$MMD = ||\frac{1}{N}\sum_{i=1}^{N} \phi(x_i) - \frac{1}{M}\sum_{j=1}^{M} \phi(y_j)||^2$$

Or $\mathrm{MMD} = \frac{1}{N^2}\sum_{i=1}^{N} \sum_{i'=1}^{N} \phi(x_i)\phi(x_i') - \frac{2}{NM}\sum_{i=1}^{N} \sum_{j=1}^{M} \phi(x_i)\phi(y_i) + \frac{1}{M^2}\sum_{j=1}^{M} \sum_{j'=1}^{M} \phi(y_i)\phi(y_i')$

For instance taking ϕ equal to be identity $\phi(x)$=x, this gives rise to the squared difference in means between X and Y, and other choices give rise to higher order moments of X and Y.

Remark that in the previous equation the distance between X and Y are only written in terms of the inner products between the mappings $\phi(.)$ of X and Y, which means that we can propose a (positive definite) kernel such that:

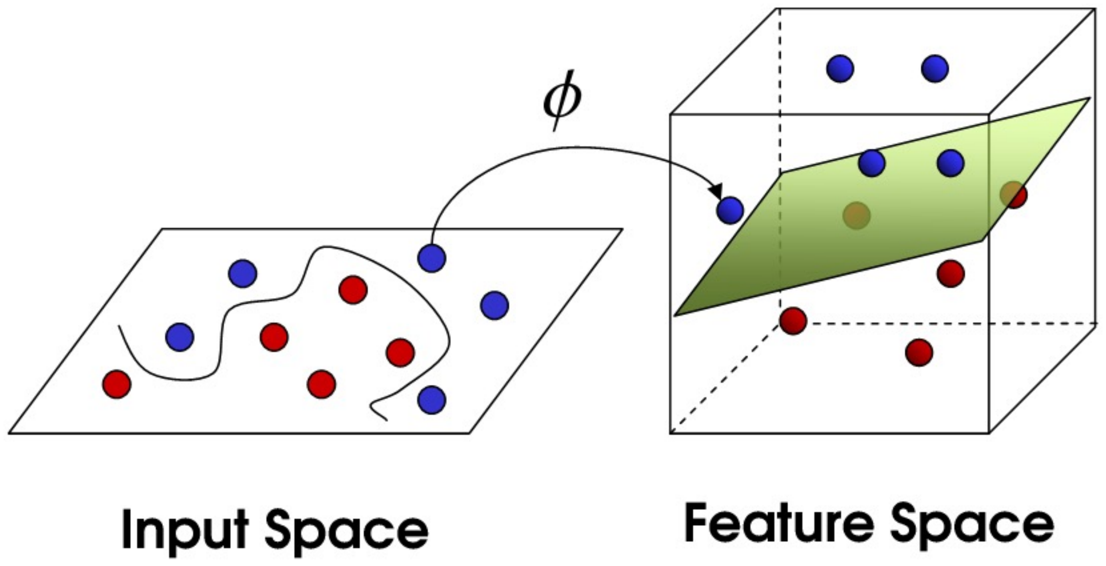
$$\mathrm{MMD} = \frac{1}{N^2}\sum_{i=1}^{N} \sum_{i'=1}^{N} k(x_i, x_i') - \frac{2}{NM}\sum_{i=1}^{N} \sum_{j=1}^{M} k(x_i, y_i) + \frac{1}{M^2}\sum_{j=1}^{M} \sum_{j'=1}^{M} k(y_i, y_i')$$

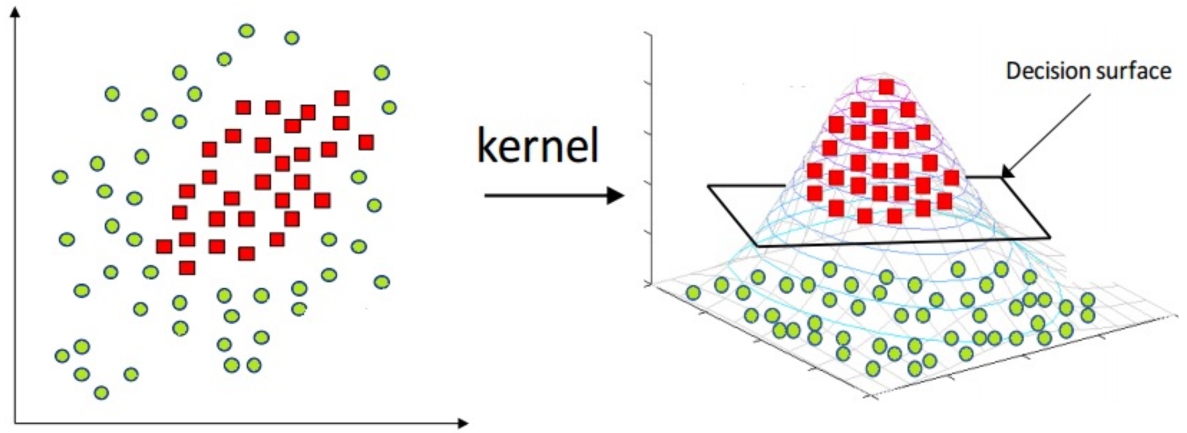
SIGNATURE-BASED MMD LOSS

$$\text{MMD} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i'=1}^{N} k(x_i, x_i') - \frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} k(x_i, y_i) + \frac{1}{M^2}\sum_{j=1}^{M}\sum_{j'=1}^{M} k(y_i, y_i')$$

- The above summarizes the main purpose of kernels in this application, namely that the distance between two samples in terms of a feature map ϕ can be evaluated without having to actually compute all the mappings $\phi(.)$ of X and Y, which can lead to dramatic improvements computationally.

- This famous result is often referred to as the **kernel trick**.





SIGNATURE-BASED MMD LOSS

- **Path integral:** (path as on slide 10)
  $\int_0^T f(\gamma_t) d\gamma_t = \int_0^T f(\gamma_t) \frac{d\gamma_t}{dt} dt = \int_0^T f(\gamma_t) \dot{\gamma}_t dt$
- Let us consider a particular path integral defined for any *single* index $i \in \{1,2,...,D\}$:

$$S(\gamma)^i_{0,T} = \int_0^T d\gamma^i = \gamma^i_T - \gamma^i_0$$

which is the increment of the path along the dimension i in $\{1,2,...,D\}$.

- For any *pair* of indexes $i, j \in \{1,2,...,D\}$, let us define:

$$S(\gamma)^{i,j}_{0,T} = \int_0^T \int_0^{t_j} d\gamma^i d\gamma^j$$

- likewise for *triple* indices in $i, j, k \in \{1,2,...,D\}$:

$$S(\gamma)^{i,j,k}_{0,T} = \int_0^T \int_{t_k}^{t_j} \int_0^{t_k} d\gamma^i d\gamma^j d\gamma^k$$

- we can continue for the *collection of k* indices $i_1, i_2, ..., i_k \in \{1,2,...,D\}$:

$$S(\gamma)^{i_1,i_2,...,i_j,...,i_k}_{0,T} = \int_0^T ... \int_{t_j}^{t_{j+1}} ... \int_{t_1}^{t_2} \int_0^{t_1} d\gamma^{i_1} d\gamma^{i_2} ... d\gamma^{i_k} (12)$$

which we call the k-fold iterated integral of γ along $\{i_1, i_2, ..., i_k\}$.

SIGNATURE-BASED MMD LOSS

- The **signature** is the collection of all the iterated integrals, consisting of all possible combinations of the indices in D (for any length of combination, hence it is an infinite series).
  However, it is important to note that these signatures are ordered along this length, which is called the *order or level of the signature*.

The signature of a path $\gamma: [0, T] \to \mathbb{R}^D$ denoted $S(\gamma)_{0,T}$ is the *collection* (an infinite series) of all the iterated integrals of γ.

Formally, $S(\gamma)_{0,T}$ is the sequence of real numbers

$$S(\gamma)_{0,T} = (1, S(X)^1_{0,T}, S(X)^2_{0,T}, \dots, S(X)^D_{0,T}, S(X)^{1,1}_{0,T}, S(X)^{1,2}_{0,T}, \dots)$$

where the zeroth term is 1 by convention and the superscript runs along the set of *multi-indices*:

$$W = \{(i_1, i_2, \dots, i_k) | k \geq 1; i_1, i_2, \dots, i_k \in \{1,2, \dots, D\}\}$$

SIGNATURE-BASED MMD LOSS

- We often consider the M-th level truncated signature, defined as the finite collection of all terms where the superscript is of max length M:

$$S_M(\gamma) = (1, S^1(\gamma), S^2(\gamma), \ldots, S^M(\gamma))$$

- where $S^k(\gamma)$ denotes all the signature terms of order k, e.g.

$$S^1(\gamma) = (S(\gamma)^1, S(\gamma)^2, \ldots, S(\gamma)^D)$$
$$S^2(\gamma) = (S(\gamma)^{1,1}, S(\gamma)^{1,2}, \ldots, S(\gamma)^{D,D})$$

Geometric and financial interpretation :

- the geometric interpretation of the first order is the increment of the path along each dimension. In financial terms, this corresponds to the *drift*.
- the second order terms correspond to the *Levy area* or the surface covered between the chord connecting the first and last coordinate in each dimension and the actual path, corresponding to a measure of *volatility* of the path.

- These two *global* features are captured by the first two orders, while more fine-grained, *local* features are captured by higher-order terms, as becomes apparent when looking at the *factorial decay of* S

SIGNATURE-BASED MMD LOSS

- **Factorial decay:** signatures are *graded summaries* of paths.

- **Terry Lyons** shows that for paths of bounded variation ($\gamma: [0, T] \to \mathbb{R}^{\mathrm{d}}$ is of bounded variation if all changes $\sum_i |\gamma_{t_{i+1}} - \gamma_{t_i}|$ are bounded (finite) for all partitions $0 \leq t_0 \leq t_1 \leq \ldots \leq T$), the following norm can be imposed on the signature terms (with $1 \leq i_1, \ldots, i_n \leq D$):

$$||\int \ldots \int \, d\gamma^{i_1} d\gamma^{i_2} \ldots d\gamma^{i_n}|| \leq \frac{||\gamma^n||^1}{n!}$$

with

$$||\gamma||^1 = \sup_{t_i \subset [0,T]} \sum_i |\gamma_{t_{i+1}} - \gamma_{t_i}|$$

where we take the supremum over all partitions of [0,T].

- This theorem guarantees that higher-order terms of the signature have factorial decay, i.e. that the order of signatures imply a graded summary of the path, first describing global and increasingly more local characteristics of the path. This implies that the truncated signature for increasing orders throws away less and less information, similar to the low-rank approximation in PCA.

SIGNATURE-BASED MMD LOSS

- **Signature as moment generating function in the path space:** sequential random variables

- For stochastic processes that generate vector-valued data, there are well-known statistical tests for determining whether two samples are generated by the same stochastic process, such as the sequence of (normalised) moments and the Fourier transform (complex moments). As discussed, the MMD allows to compare these moments by embedding two random variables in Hilbert space using kernel approximation.

- For path-valued data, **Chevyrev and Oberhauser** introduce an analogue to normalised moments using the signature. They prove that for suitable normalizations λ, the sequence

$$(\mathbb{E}[\lambda(X)^m \int dX^{\otimes m}])_{m\geq 0}$$

- determines the law of X *uniquely*. They argue that the moments in the path space up to order m are preserved (i.e. a **bijective property**) for the truncated signature up to order m.

- **Signature as optimal feature map $\phi(.)$ for embedding paths**:
The reasons are twofold:
(1) *universality*, which implies that non-linear functions of the data are approximated by linear functionals in feature space and
(2) *characteristicness*, which is exactly their merit, i.e. that the expected value of the feature map *characterizes the law of the random variable*.

GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

SIGNATURE-BASED MMD LOSS

- **Signature kernel:** In essence, it is just the inner product between the two signature vectors of two random variables in the path space.

> Let x and y be two paths supported on $[0, T]$, $x: [0, T] \to \mathbb{R}^D$ and $y: [0, T] \to \mathbb{R}^D$. The signature kernel k: $[0, T] \times [0, T] \to \mathbb{R}$ is defined as
> $$k_S(x, y) = \langle S(x), S(y) \rangle$$

Intuitively, say for S truncated at order 1, k_S measures the similarity between the drifts of the two paths. Truncated at order 2, k_S looks at drift and volatility similarity, and so and so forth.

- **Signature MMD:** MMD can be used as a *deterministic loss function in a generative model* as it is a distance measure between two random variables, for instance the *fake* generated data and the *true* input data. When the path structure of the random variable is crucial, the choice of traditional kernels is inappropriate and we should use a sequential kernel. As described above, this is exactly what signatures allow us to do. Let us first generalize the MMD expression to:

$$MMD(\mu, \nu) = sup_{f \in \mathcal{H}} E_{X \sim \mu}[f(X)] - E_{X \sim \nu}[f(X)] (19)$$

- Hence, MMD is literally the *maximum expected distance between two functions in the embedded space* $\mathcal{H}$. We can further rewrite:

$$MMD_S(\mu, \nu) = E_{XX' \sim \mu}[k_S(X, X')] - 2E_{X \sim \mu, Y \sim \nu}[k_S(X, Y)] + E_{YY' \sim \nu}[k_S(Y, Y')]$$

- The signature MMD. The expression in itself is easy to compute, but its computational performance hinges on how efficiently we can evaluate k_S.

SIGNATURE-BASED MMD LOSS

- **PDE kernel trick**: a recent result concerns a kernel trick for sequential kernels, the signature partial differential equation (PDE) kernel trick. Salvi et al. 2021 proved that the signature kernel can be written as the solution of a hyperbolic PDE belonging to the family of so-called Goursat problems. This substantially speeds up the evaluation of k_S and allows for GPU-optimized parallelization of the PDE solver. Formally, we can write:

$$\frac{\delta^2 k_S}{\delta s \delta j} = \langle \dot{X}(s), \dot{Y}(j) \rangle k_S$$

- where $k_S(X(0), .) = k_S(., Y(0)) = 1$ and $\dot{X}(s) = \frac{dX}{dt}|_{t=s}$ and $\dot{Y}(j) = \frac{dY}{dt}|_{t=j}$, which is a Goursat PDE. They further show that this PDE can be written as a function of a static kernel κ, e.g. the RBF or Matern kernel:

$$\frac{\delta^2 k_S}{\delta s \delta j} = (\kappa(X(s), Y(j)) - \kappa(X(s-1), X(j)) - \kappa(X(s), Y(j-1)) + \kappa(X(s), Y(j-1)))k_S$$

- After an appropriate choice of κ, equation can then be solved using state-of-the-art PDE solvers and efficiently parallelized over GPU. This allows for an efficient evaluation of k_S in MMD_S.
-

PROPERTIES: SIGNATURE UNIVERSALITY AND DRAWDOWN

Universality. Non-linear continuous functions of the unparameterized data streams are universally approximated by linear functionals in the signature space.

Theorem (Lyons, Ni). Denote by S the function that maps a path X from K to its signature $S(X)$. Let $f: K \to \mathbb{R}$ be any continuous function. Then, for any $\epsilon > 0$, there exists $M > 0$, and a linear functional L acting on the truncated signature of degree M such that

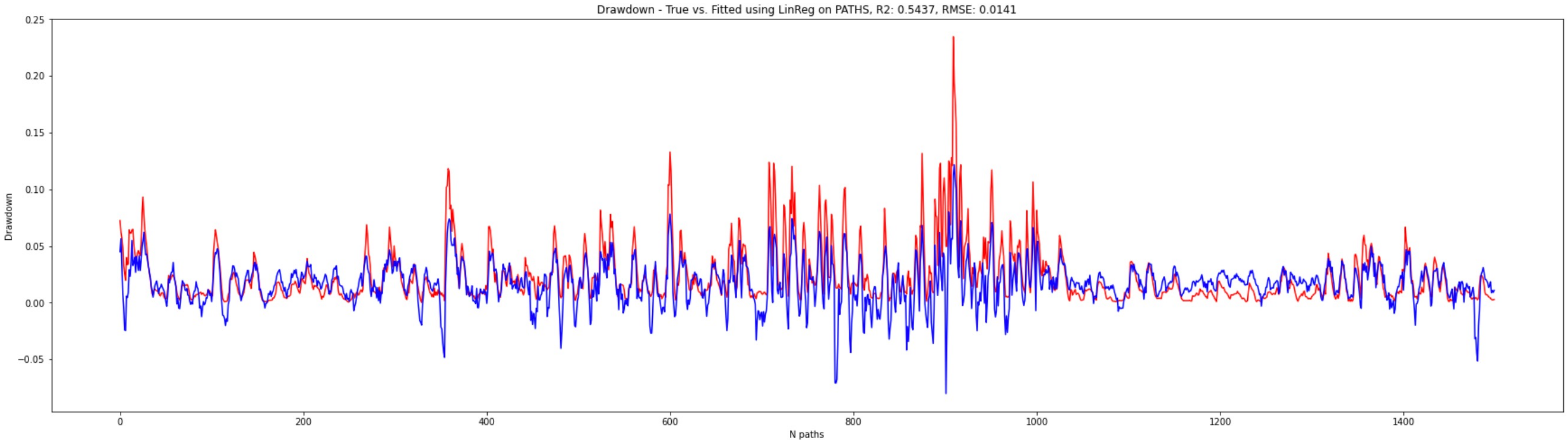
$$\sup_{X \in K} |f(X) - \langle L, S_M(X) \rangle| < \epsilon$$

GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

PROPERTIES: SIGNATURE UNIVERSALITY AND DRAWDOWN

- E.g. f(P) = $\int_0^T \left(\max_{t_i<t}(P_{t_i}) - P_t\right)dt$, or the expected drawdown of prices P over T, is non-linear due to the max operation.
- So, **linear regression** of examples true f(P) on P_t makes little sense:

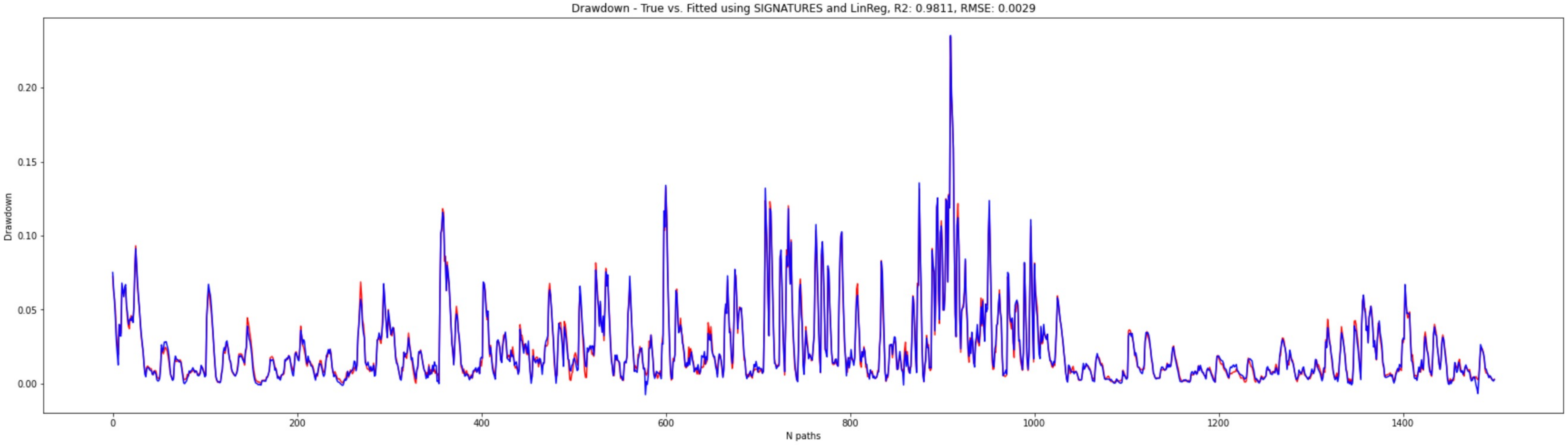


GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

PROPERTIES 2/2: SIGNATURE UNIVERSALITY AND DRAWDOWN

- However, **universality** assures that f(X) can be ϵ $-$approximated arbitrarily well (depended on the estimation of L and truncation level M), i.e. drawdown as a linear combination of signature terms.
- For instance, **linear regression** (= L from OLS) of f(P) on $S(P)$, M=10, yields:

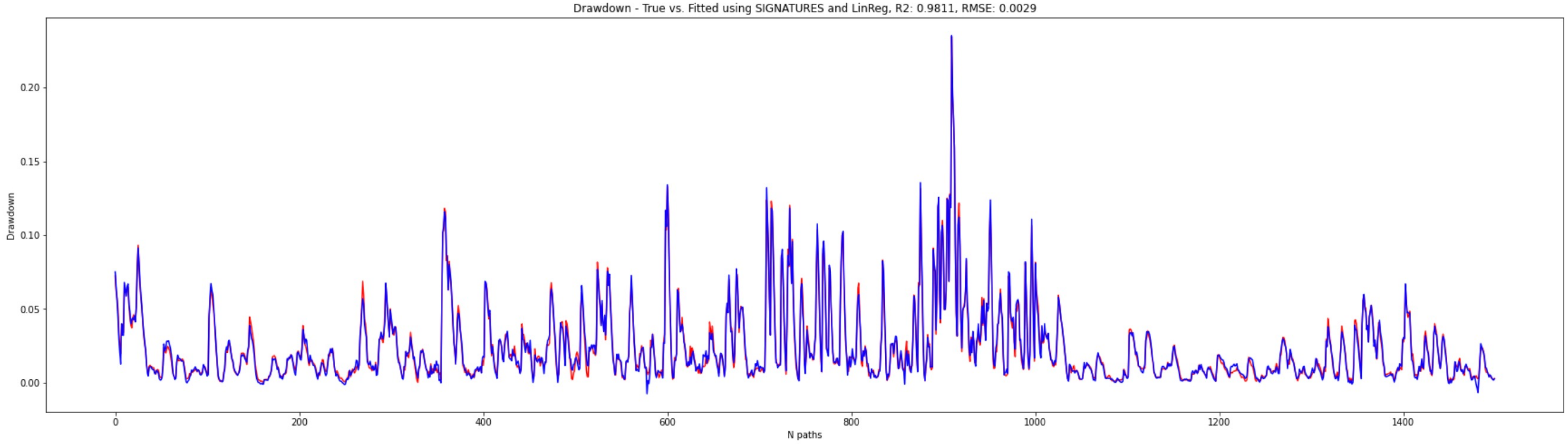


GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

PROPERTIES 2/2: SIGNATURE UNIVERSALITY AND DRAWDOWN

- **Drawdown as a linear combination of signature terms**: based on **pre-trained L** (linear combination) the drawdowns of 2 samples (e.g. fake/real) can easily be evaluated by their signatures as well, without requiring max-operators (e.g. within the system of differentiable equations of our generative model)!!

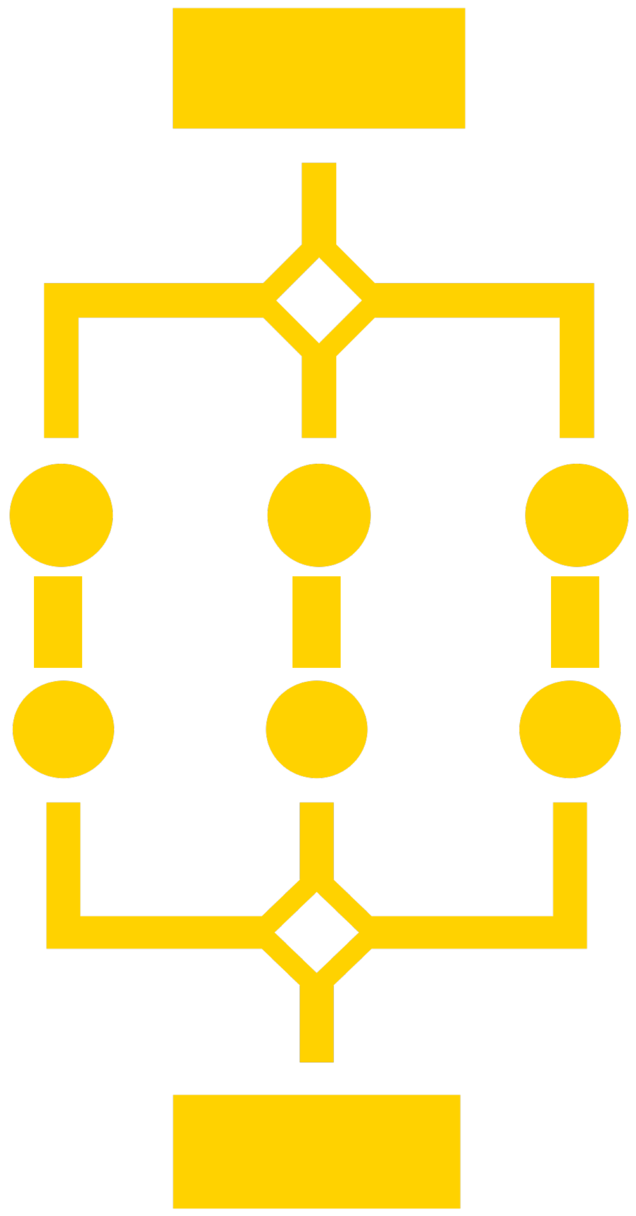


GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

GENERATIVE ML ARCHITECTURES

- Generative Adversarial Networks (GAN)
- Generative Moment Matching Networks (GMMN)
- Variational Autoencoders (VAE)
- Restricted Boltzmann Machines (RBM)
- Flow-Based Models / Normalizing Flows (NF)



GAN

- Arguably the most popular architecture in generative ML, with well-known applications in computer vision (deepfake etc).

- Trained by a adversarial game between two networks, a decoder network (previously f_{Θ}^{-1}(Z)) and a discriminator network.

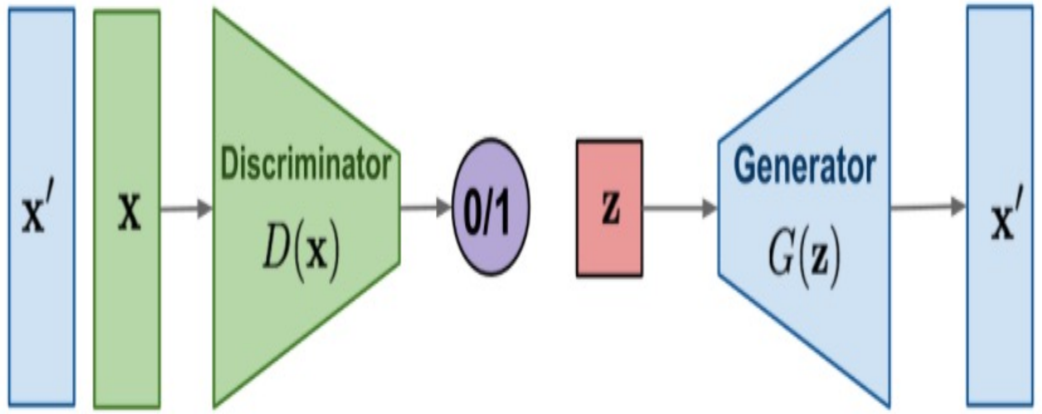
- Samples a latent variable z from a simple prior distribution $\mathbb{P}(Z)$, e.g. Gaussian or Uniform, followed by a decoder network, the transform $G(z)$, called the **Generator**.

- The **Discriminator** $D(.)$ outputs a probability of a given sample coming from the real data distribution. Its task is to distinguish samples from the real distribution $\mathbb{P}(X)$ from $G(z)$.

- The decoder tries to produce samples as close to the original distribution possible, as to fool the discriminator.
- This gives rise to the following well-known minimax problem:

$$\min_{G}\max_{D}\mathbb{E}_{x\sim\mathbb{P}(X)}[log(D(x)] + \mathbb{E}_{z\sim\mathbb{P}(Z)}[log(1-D(G(z))]$$

GAN: Adversarial training

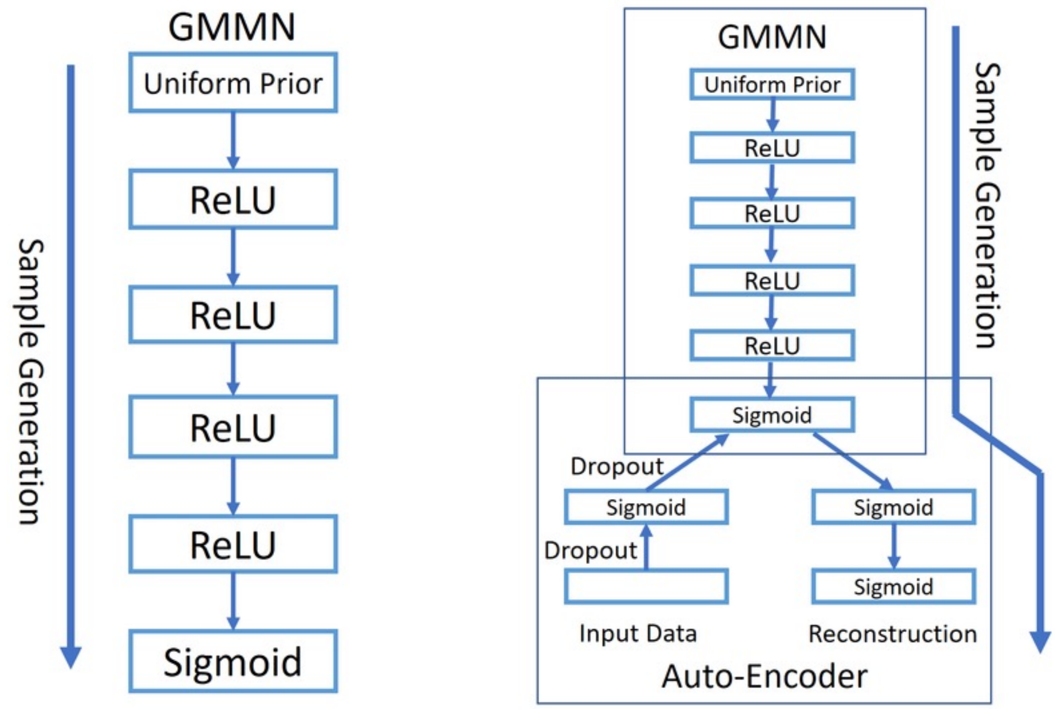


GHENT UNIVERSITY

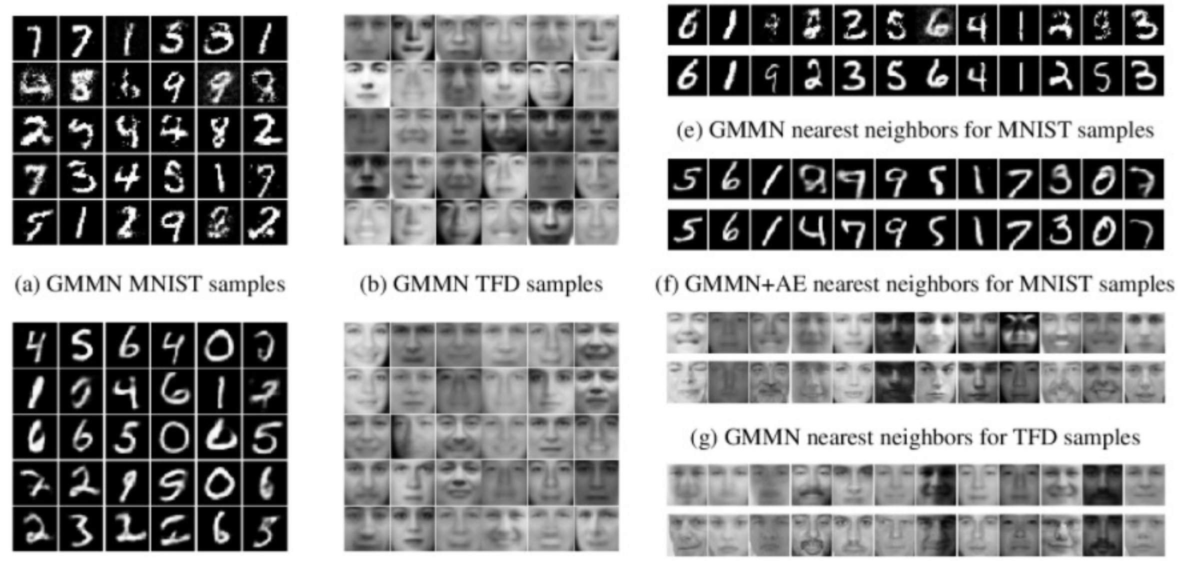
FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

GMMN

- Simple forward pass through a multi-layer NN from a uniform prior.
- **MMD loss** (also called MDD networks)
- Traditionally with Gaussian kernel, where it can be proven that it is a discrepancy measure between *all* the moments of the generated fake versus the true data distribution.
- More performant in combination with an autoencoder architecture.



(a) GMMN (b) GMMN + AE

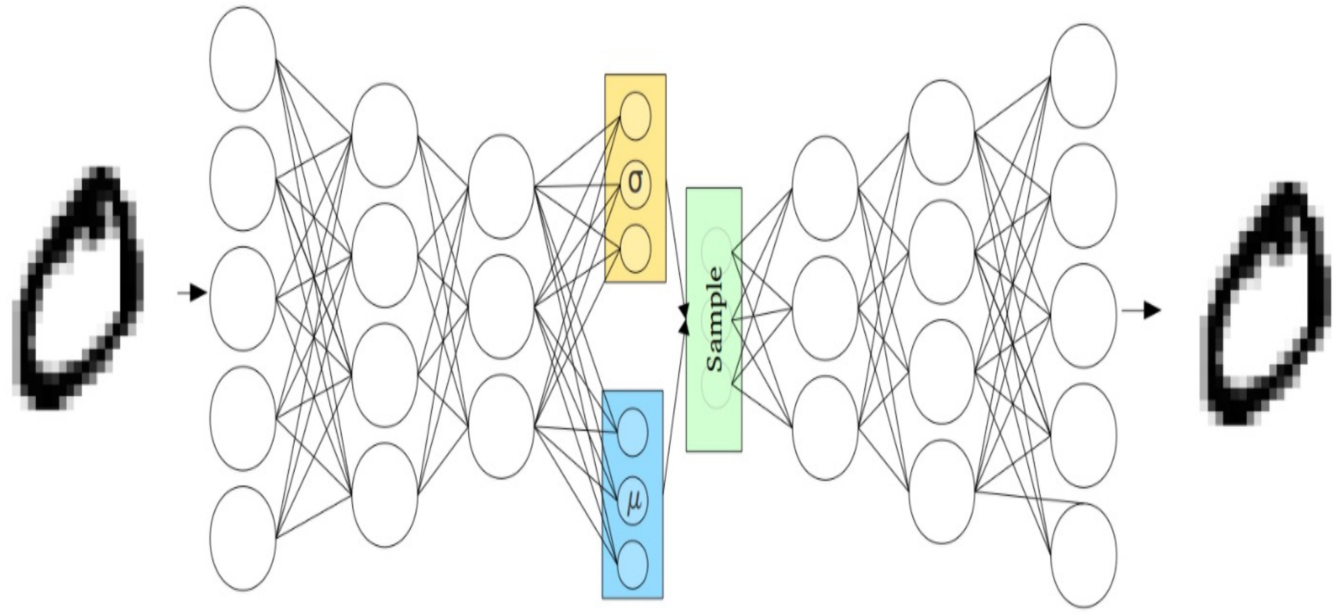

(a) GMMN MNIST samples (b) GMMN TFD samples (e) GMMN nearest neighbors for MNIST samples (f) GMMN+AE nearest neighbors for MNIST samples (c) GMMN+AE MNIST samples (d) GMMN+AE TFD samples (g) GMMN nearest neighbors for TFD samples (h) GMMN+AE nearest neighbors for TFD samples

VAE

- Introduced by Kingma and Welling in 2014 (‘*Variational inference using Bayes*’), and second most popular architecture (after GAN) in generative ML, with applications to market generators in Buehler 2020, Fung 2021, and Bergeron 2021.
- **Autoencoder**: $f_\Theta(X)$ and $f_\Theta^{-1}(Z)$ are both neural networks, here respectively called the encoder and decoder network.
- Characterized by their joint distribution over the latent variables Z and the observed variables X: $\mathbb{P}(x,z)=\mathbb{P}(x|z)\mathbb{P}(z)$
- Kingma 2014 approximates the posterior function $\mathbb{P}(z|x)$ using an encoder model $f_\Theta(X)$. Two contributions are key in appraising their work.

  (1) They derive a lower bound for $\mathbb{P}_\Theta(X)$ by comparing this posterior with samples from an actual Gaussian using the Kullback-Leibler divergence
  $$\log(\mathbb{P}(x)) \geq \mathbb{E}_{f_\Theta(x)}[log(\mathbb{P}(x|z))] - KL(f_\Theta(x)||\mathbb{P}(z))$$
  where maximizing the right-hand side (the Evidence Lower Bound (ELBO)) corresponds to maximizing the loglikelihood of the data distribution as a function of Θ.

  (2) They use a mathematical trick called the *reparametrization trick* that allows for backpropagation (see below) over the latent space Z.



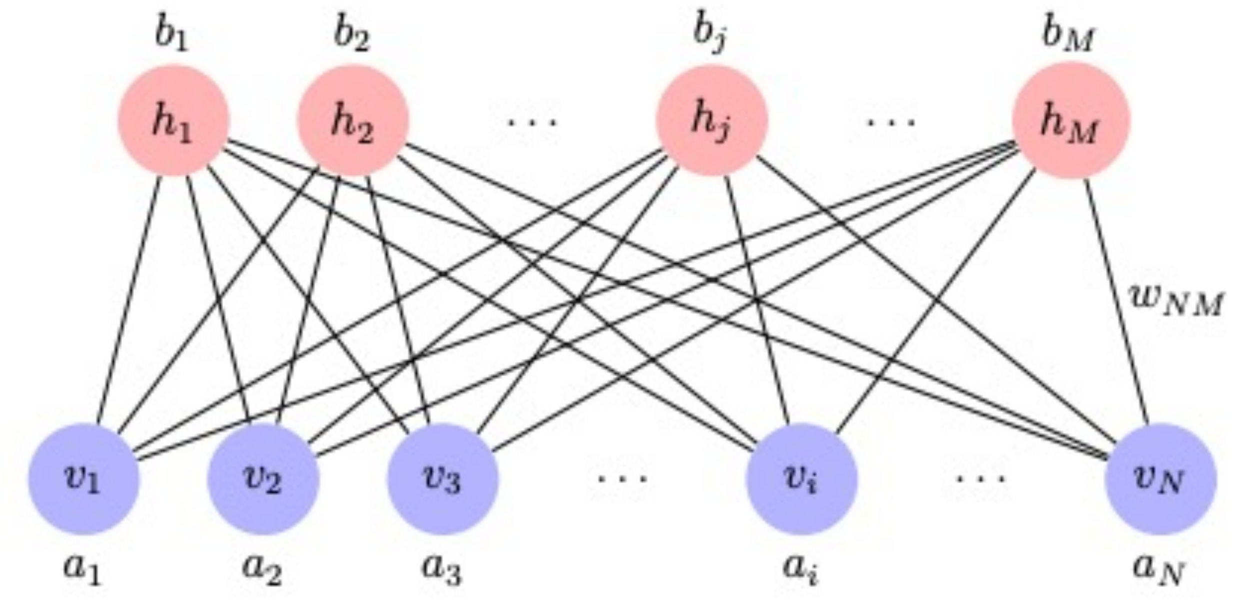
RBM

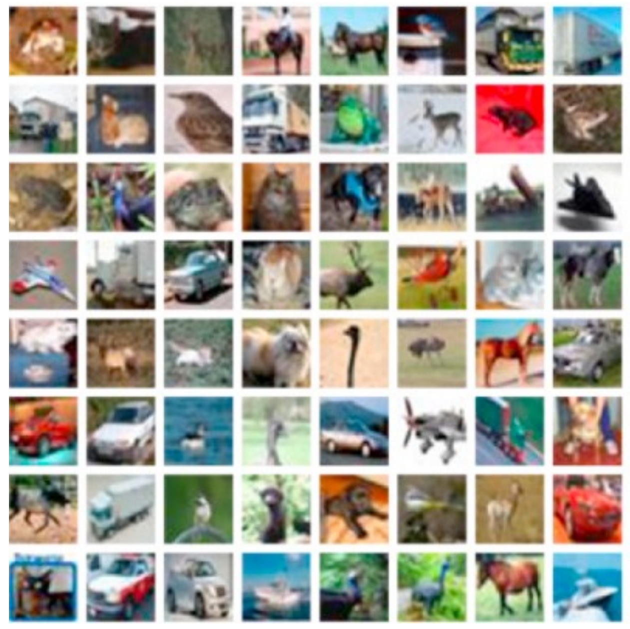
- **Energy-based models** dating back to Harmonium in 1980s (Smolensky 1986)
- Bipartite graphs (two-layer neural networks), with one visible layer v that represents $\mathbb{P}(X)$ and one hidden layer h representing $\mathbb{P}(Z)$.
- **Restricted** refers to the fact that there are no connections or model weights Θ between nodes within each layer, only across the two layers.
- Each node in the graph represents a binary stochastic variable
- **Boltzmann** refers to the Boltzmann energy function that measures the likelihood of the states of the graph (which in statistical physics is called a Markov Random Field) by its joint distribution:

$$\mathbb{P}(v,h) = \frac{1}{Z}\exp(-E(v,h))$$

$$E(v,h) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} w_{i,j} v_i h_j$$

where v_i and h_j denote the individual nodes or state variables in resp. v and h. In this case v_i and h_j are stochastic binary, hence Bernouilli, variables, but this can be approximated with Gaussian-Bernouilli variables for continuous distributions such as financial returns.





RBM

- The goal of training this network is maximizing its joint likelihood, which corresponds to minimizing the energy of the graph's state:

- Through **Markov Chain Monte Carlo** (**MCMC**) sampling techniques such as **Gibbs sampling** and improved alterations of it such as **contrastive divergence**, it can be shown that the energy decreases as $\mathbb{P}_\Theta(X)$, the distribution of the visible layer with parameters Θ, approaches the true $\mathbb{P}(X)$, or the distribution of the data.

- Once training has converged, one can iteratively sample noise in v and back and forth with h until we have new samples of $\mathbb{P}(X')$.

- This was the approach in the original **Market Generator** paper by Kondratyev and Schwarz 2019. The impressive results were confirmed by Lezmi 2020.

- The goal of training this network is maximizing its joint likelihood, which corresponds to minimizing the energy of the graph's state:

- Through **Markov Chain Monte Carlo** (**MCMC**) sampling techniques such as **Gibbs sampling** and improved alterations of it such as **contrastive divergence**, it can be shown that the energy decreases as $\mathbb{P}_\Theta(X)$, the distribution of the visible layer with parameters Θ, approaches the true $\mathbb{P}(X)$, or the distribution of the data.

- Once training has converged, one can iteratively sample noise in v and back and forth with h until we have new samples of $\mathbb{P}(X')$.

- This was the approach in the original **Market Generator** paper by Kondratyev and Schwarz 2019. The impressive results were confirmed by Lezmi 2020.

NORMALIZING FLOWS

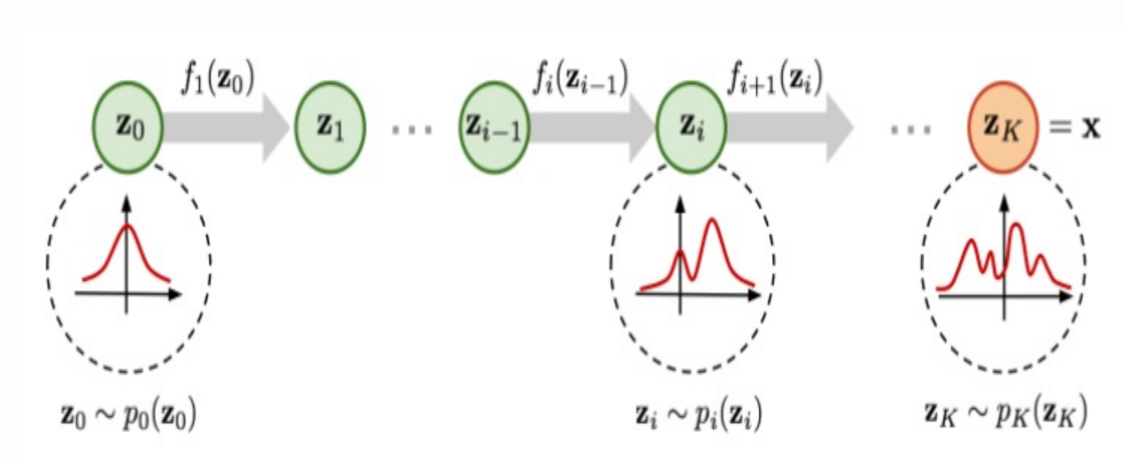
- **Flow-based generative models** or normalizing flows (NF) is a class of neural networks which use differentiable mappings to approximate simple bijective functions called *diffeomorphisms*. These transform a simple distribution Z to a complex one, step by step.

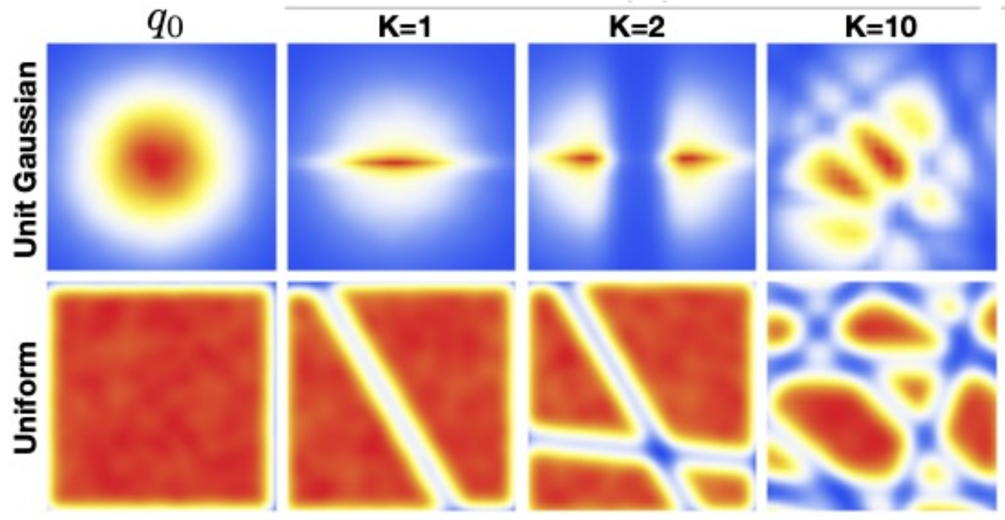
- In our notation $f_\Theta^{-1}(Z)$ would be a neural network that stacks these diffeomorphisms (such as linear neural splines, for a recent overview see Kobyzev, 2020) as to approximate a divergence measure between the target distribution $\mathbb{P}(X)$ and the sampled distribution $\mathbb{P}_\Theta(X')$.
- For instance, Wiese 2021 uses NFs to approximate (i.e. using gradient descent) the Monte Carlo-approximated KL-divergence:

$$\nabla_\Theta KL(\mathbb{P}(X)||\mathbb{P}_\Theta(X')) = -\mathbb{E}_{x\sim\mathbb{P}(X)}(\nabla_\Theta ln(\mathbb{P}_\Theta(X')))$$

$$\approx \frac{1}{n}\sum_{i=1}^{n} \nabla_\Theta(ln(|detJ_{f_\Theta^{-1}}(f_\Theta(x_i))|) - ln\mathbb{P}(f_\Theta(x_i)))$$

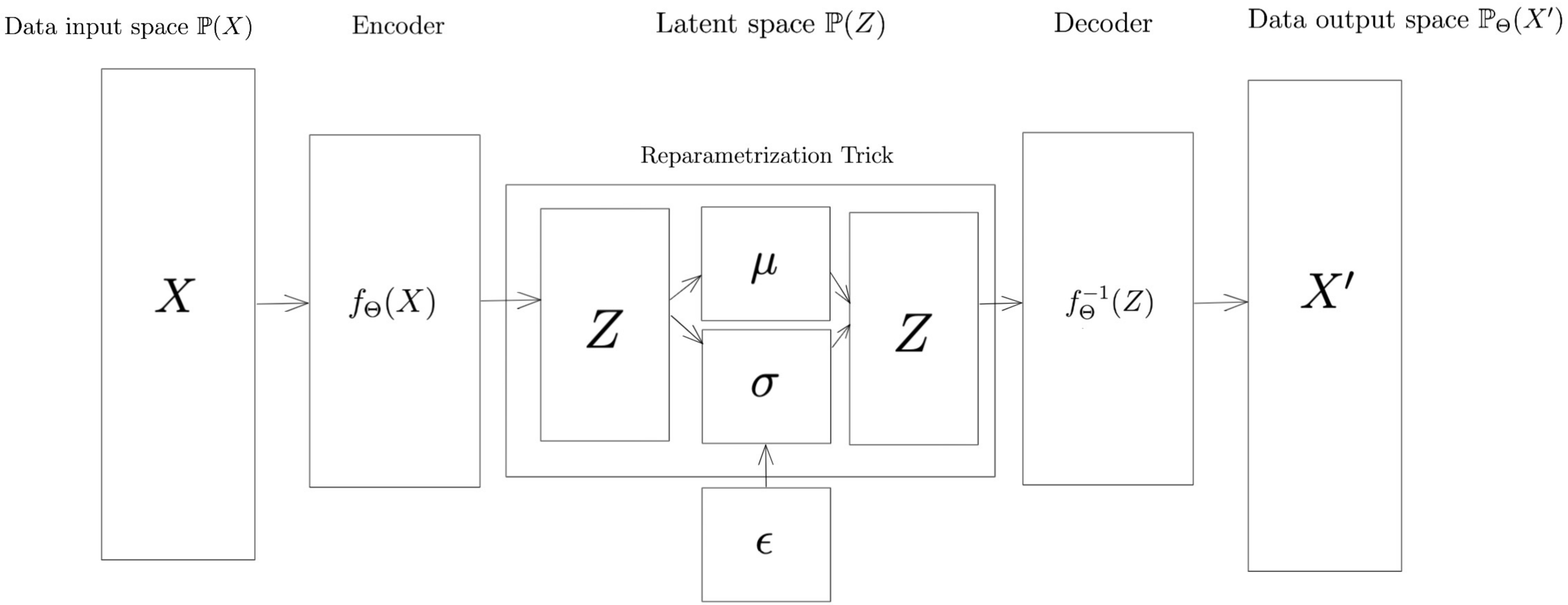
where J represents the Jacobian of the neural network f_Θ^{-1}, the matrix of first order derivatives of the network to the latent space values. The determinant of the Jacobian thus plays a crucial role in approximating the KL using MC. For the computation of the determinant to be efficient, the computation of the determinant of the individual diffeomorphisms is typically chosen simple (e.g. linear splines). Making them sufficiently simple but expressive enough is a key element of research in NFs.





DETAILED LOOK: CVAE

- VAE: (+) converges fast, generally more stable and gives us intepretable posteriors after training, (-) less flexible than GAN
- **Let us have a deeper look at the architecture**



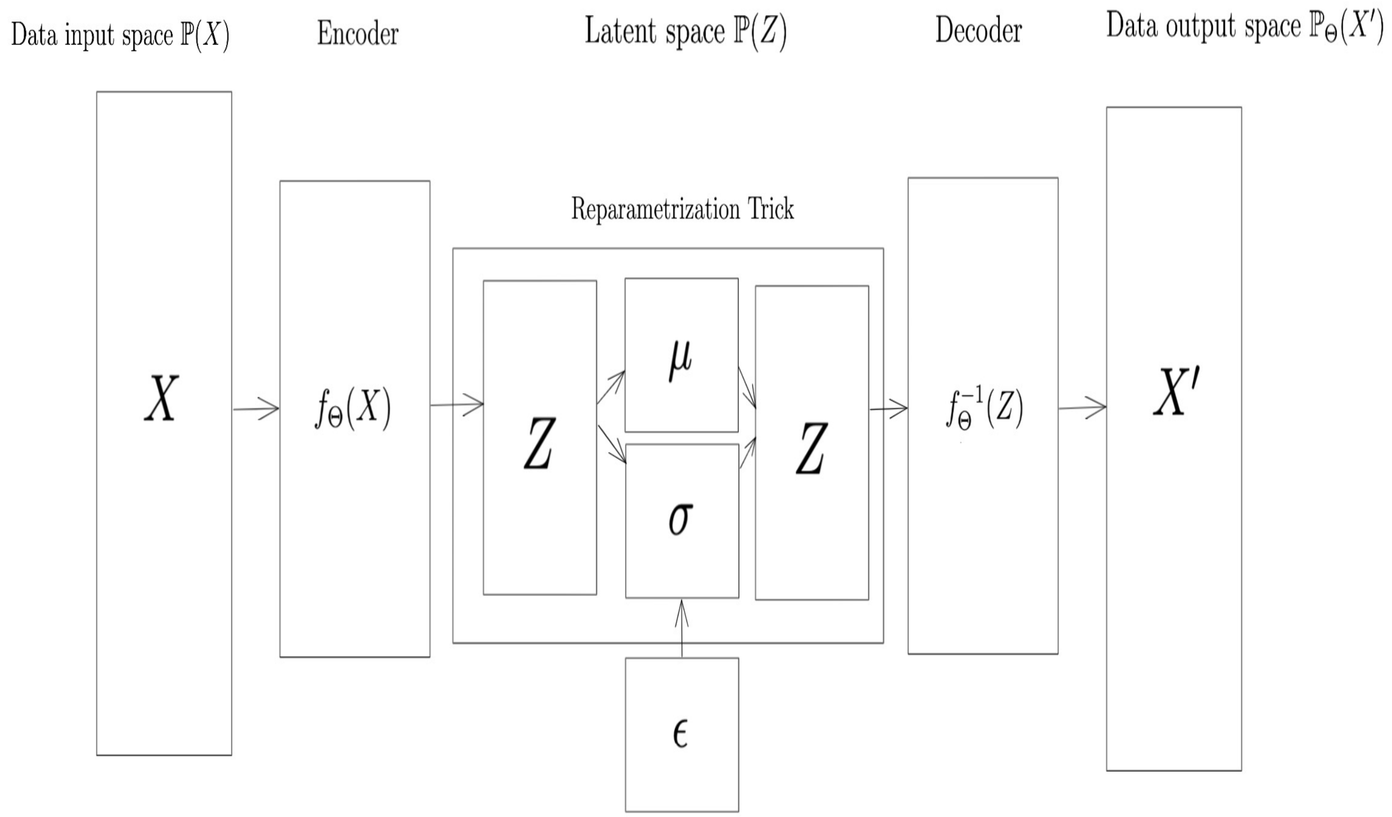
GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

DETAILED LOOK: CVAE

Architecture

- As input we have the D-dimensional *ambient* space X or the physical data domain that we can measure (e.g. R or Ξ)
- Using a flexible neural network mapping $f_\Theta\colon \mathbb{R}^D \rightarrow \mathbb{R}^K, K << N$, called the encoder, we compress the dimension of the data into a K-dimensional *latent* space Z, e.g. 10-dimensional.
- Using the reparametrization trick we map Z onto a mean μ and standard deviation σ vector, i.e. onto a K-dimensional Gaussian, e.g. a 10-dimensional normal distribution.
- The decoder neural network $f_\Theta^{-1}\colon \mathbb{R}^K \rightarrow \mathbb{R}^D$ maps the latent space back to the output space $\mathbb{P}_\Theta(X')$ where X' can be considered *reconstructed* samples in the training step, or genuinely new or fake samples in a generator step.
- The quality of the VAE clearly depends on the similarity between $\mathbb{P}(X)$ and $\mathbb{P}_\Theta(X')$



DETAILED LOOK: CVAE

- Let us now zoom in on $f_{\Theta}(X)$ and $f_{\Theta}^{-1}(Z)$. Each neural network consist of one layer of J mathematical units called neurons:

$$f_{\Theta_j} := A\left(\sum_{i}^{D} \theta_{i,j} x_i\right)$$

- Every neuron takes *linear* combinations θ_i of the input data point x_i and is then *activated* using a *non-linear* activation function A, such as rectified linear units (ReLU), hyperbolic tangent (tanh) or sigmoid. In this paper we use a variant of ReLU called a *leaky ReLU*:

$$LReLU(x) = \mathbf{1}_{x<0}\alpha x + \mathbf{1}_{x\geq 0} x$$

where α is a small constant called the slope of the ReLU.

DETAILED LOOK: CVAE

All neurons J are linearly combined into the next layer (in this case Z):

$$Z_k := \sum_{j}^{J} \theta_{j,k} f_{\Theta_j}$$

for every k in K.

The decoder map can formally be written exactly like the encoder, but in reverse order.

The **loss function of a VAE** generally consists of two components, the latent loss ($\mathcal{L}_L$) and the reconstruction loss ($\mathcal{L}_R$):

$$\mathcal{L}(X, X') = \beta \mathcal{L}_L + (1 - \beta) \mathcal{L}_R$$

The latent loss is the Kullback-Leibler discrepancy between the latent distribution under its encoded parametrization, the posterior $f_\Theta(X) = \mathbb{P}_\Theta(Z|X)$, and its theoretical distribution, e.g. multi-variate Gaussian $\mathbb{P}(Z)$. Appendix B in Kingma 2014 offers a simple expression for $\mathcal{L}_L$. The reconstruction loss is the cost of reproducing $\mathbb{P}_\Theta(X')$ after the dimension reduction step, and originally computed by the root of the mean squared error (RMSE or $L2$-loss) between X and X'.

$$\mathcal{L}(X, X') = \beta \frac{1}{2} \sum_{k}^{K} \left(1 + \sigma - \mu^2 - \exp(\sigma)\right) + (1 - \beta) \mathbb{E}(\,||X - X'||^2)$$

The parameter β can be tuned to get so-called *disentangled* latent representations in the β-VAE architecture

DETAILED LOOK: CVAE

Training process

- Optimal loss values $\mathcal{L}^*$ are determined by stochastically sampling batches of data and alternating forward and backward passes through the VAE.

- For each batch the data is first passed through the encoder network and decoder network (*forward pass*), after which $\mathcal{L}$ is evaluated in terms of Θ. At each layer, the derivative of $\mathcal{L}$ vis-a-vis Θ can easily be evaluated.

- Next (*the backward pass*), we say the calculated loss *backpropagates* through the network, and Θ are adjusted in the direction of the gradient $\nabla_\Theta \mathcal{L}$ with the *learning rate* as step size.

- The exact optimizer algorithm we used for this is Adam (Adaptive moments estimation)

- Finally, we can also use a concept called regularization, which penalizes neural models that become too complex or overparametrized. We used a tool called dropout, that during training randomly sets a proportion of parameters in Θ equal to zero, and leaves those connections at zero that contribute the least to the prediction.

DETAILED LOOK: CVAE

- In summary, the **hyperparameters** of this architecture are:

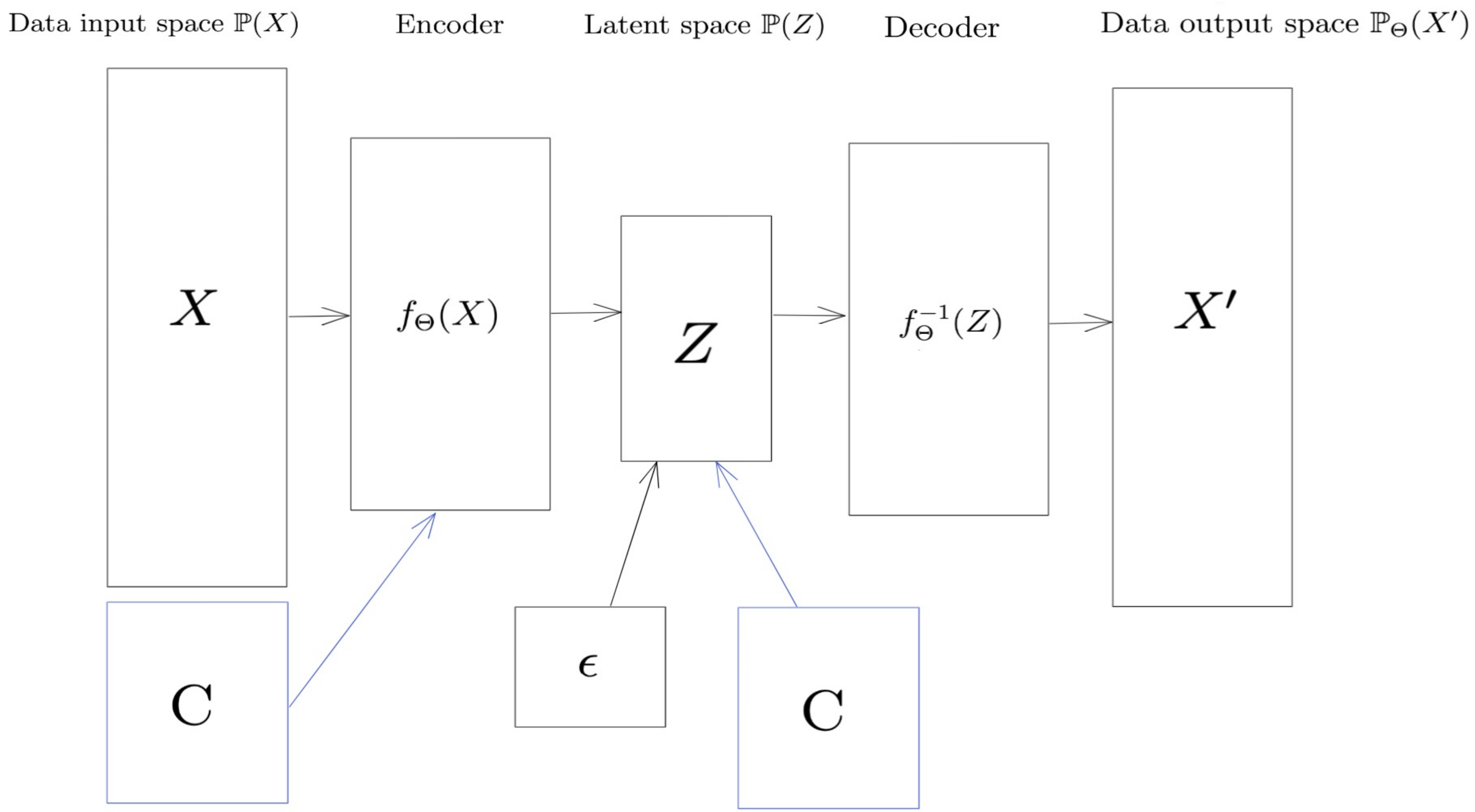
  (1) the number of neurons in the encoder,
  (2) the number of neurons in the decoder,
  (3) the number of latent dimensions K,
  (4) the learning rate,
  (5) the optimizer algorithm and
  (6) the dropout rate.

  We opted for the following set-up (which was optimized using Grid Search): 100, 100, 10, 0.001, Adam, 0.0.

- After training, in the **sampling or generation step**, we start from a random K-dimensional noise $\epsilon \sim \mathbb{P}(Z)$ which is K-variate Gaussian. Now, we only need a decode step to generate new samples of $\mathbb{P}_{\Theta}(X')$

DETAILED LOOK: CVAE

- The importance of conditional factors: e.g. Instrumented PCA (IPCA)
- $P_\Theta(X')$ vs. $P_\Theta(X'|C)$ => **Conditional VAE (CVAE)**



GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

CONDITIONAL SAMPLING

Table 3: Macro-economic conditions

ID	FRED ID	FRED Cat.	Detailed Cat.	Indicator
0	TREAST	Finance	Monetary Data	US Treasuries Held by the Fed
1	MBST	Finance	Monetary Data	Mortgage Backed Sec Held by the Fed
2	WALCL	Banking	Monetary Factors	All Fed Reserve Banks - Total Assets
3	TLAACBW027SBOG	Banking	Monetary Factors	All Commercial Banks - Total Assets
4	BOPBCA	Banking	Conditions	Number of US Banks
5	USNUM	Banking	Conditions	Number of US Commercial Banks
6	EQTA	Banking	Conditions	Equity/Asset Ratio
7	TOTBKCR	Banking	Commercial Credit	Bank Credit of All Commercial Banks
8	TOTALSEC	Banking	Commercial Credit	Securitized Total Consumer Loans
9	TOTALSL	Banking	Commercial Credit	Total Consumer Credit Outstanding
10	INVEST	Banking	Investment	Total Investments All Commercial Banks
11	USGSEC	Banking	Investment	US Gov't Securities at All Com. Banks
12	CONSUMER	Banking	Loans	Total Consumer Loans
13	BUSLOANS	Banking	Loans	Total Commercial/Industrial Loans
14	DALLCACBEP	Banking	Delinquencies	Delinquencies On All Loans And Leases
15	T10Y2Y	Banking	Interest Rates	US 10-YR / 2-YR Spread
16	TB3MS	Banking	Interest Rates	3-Month T-Bill: Secondary Market Rate
17	DGS10	Banking	Interest Rates	10-Yr Treasury Const. Maturity Rate
18	GFDEBTN	Business/Fiscal	Federal Government	Federal Government Debt (Public)
19	FYOINT	Business/Fiscal	Federal Government	Interest on National Debt
20	FYONET	Business/Fiscal	Federal Government	Federal Spending
21	FYFR	Business/Fiscal	Federal Government	Federal Receipts
22	FYFSD	Business/Fiscal	Federal Government	Budget Deficit/Surplus
23	CDSP	Business/Fiscal	Household Sector	Consumer Debt/Income Ratio
24	PERMIT	Business/Fiscal	Household Sector	New Home Permits
25	HSN1F	Business/Fiscal	Household Sector	New Home Sales
26	CMDEBT	Business/Fiscal	Household Sector	Outstanding Mortgage Debt
27	DGORDER	Business/Fiscal	Ind. Production	Manufacturers' New Orders
28	TCU	Business/Fiscal	Ind. Production	Capacity Utilization: Total Industry
29	TTLCONS	Business/Fiscal	Construction	Total Construction Spending
30	BUSINV	Business/Fiscal	Other	Total Business Inventories
31	ALTSALES	Business/Fiscal	Other	Light Weight Vehicle Sales
32	UMCSENT	Business/Fiscal	Other	Univ of Michigan: Consumer Sentiment
33	STLFSI	Business/Fiscal	Other	St. Louis Financial Stress Index
34	OILPRICE	Business/Fiscal	Other	Spot Oil Price - West Texas Intermediate
35	CPIAUCSL	Consumer Prices	CPI	Consumer Price Index: Seasonally Adj.
36	UNRATE	Empl & Population	Household Survey	Civilian Total Unemployment Rate
37	UEMP27OV	Empl & Population	Household Survey	Long Term Unemployment: 27 WKS
38	UEMPMED	Empl & Population	Household Survey	Length of Unemployment
39	CE16OV	Empl & Population	Household Survey	Total US Workforce
40	EMRATIO	Empl & Population	Household Survey	US Employment/Population Ratio
41	POP	Empl & Population	Population	US Population
42	AHEMAN	Empl & Population	Est. Survey	Avg Hourly Earnings: Manufacturing
43	AWHMAN	Empl & Population	Est. Survey	Avg Weekly Hours: Manufacturing
44	AWOTMAN	Empl & Population	Est. Survey	Avg Weekly OT Hours: Manufacturing
45	DEXUSUK	Exchange Rates	Daily Rates	USD/GBP Currency Exchange Rate
46	DEXUSEU	Exchange Rates	Daily Rates	USD/EUR Currency Exchange Rate
47	DEXJPUS	Exchange Rates	Daily Rates	JPN/USD Currency Exchange Rate
48	DEXMXUS	Exchange Rates	Daily Rates	MXP/USD Currency Exchange Rate
49	DEXCAUS	Exchange Rates	Daily Rates	CAD/USD Currency Exchange Rate
50	DEXCHUS	Exchange Rates	Daily Rates	CNY/USD Currency Exchange Rate
51	COMPOUT	Financial Data	Monetary	Commercial Paper Outstanding

Continued on next page

Table 3 – continued from previous page

ID	FRED ID	FRED Cat.	Detailed Cat.	Indicator
52	VIXCLS	Financial Data	Volatility Indexes	CBOE Volatility Index
53	GDP	GDP & Components	GDP/GNP	US Gross Domestic Product
54	GNP	GDP & Components	GDP/GNP	US Gross National Product
55	NETFI	GDP & Components	Imports & Exports	US Current Account Balance
56	EXPGS	GDP & Components	Imports & Exports	US Exports Goods & Services
57	IMPGS	GDP & Components	Imports & Exports	US Imports Goods & Services
58	DGI	GDP & Components	Govt Accounting	Fed Govt: Defense Budget
59	FGRECPT	GDP & Components	Govt Accounting	Fed Govt: Tax Receipts
60	TGDEF	GDP & Components	Govt Accounting	Fed Govt: Budget Deficit
61	CP	GDP & Components	Industry	Corporate Profits After Tax
62	DIVIDEND	GDP & Components	Industry	Corporate Dividends
63	PI	GDP & Components	Personal	Personal Income
64	PSAVE	GDP & Components	Savings & Inv.	Personal Savings
65	PSAVERT	GDP & Components	Savings & Inv.	Personal Savings Rate
66	MORTGAGE30US	Interest Rates	30yr Mortgage	30-yr Conventional Mortgage Rate
67	DPCREDIT	Interest Rates	FRB Rates	Discount Rate
68	FEDFUNDS	Interest Rates	FRB Rates	Effective Federal Funds Rate
69	GRCPROINDMISMEI	International Data	Indicators	Production of Total Industry in Greece
70	GRCSARTMISMEI	International Data	Indicators	Total Retail Trade in Greece
71	GRCURHARMMDSMEI	International Data	Indicators	Unemployment Rate - Greece
72	M1	Monetary Aggregates	M1	M1 Money Supply
73	M2	Monetary Aggregates	M2	M2 Money Supply
74	MZM	Monetary Aggregates	MZM	MZM Money Supply
75	M1V	Monetary Aggregates	M1	Velocity of M1 Money Stock
76	M2V	Monetary Aggregates	M2	Velocity of M2 Money Stock
77	MZMV	Monetary Aggregates	MZM	Velocity of MZM Money Stock
78	MULT	Monetary Aggregates	M1	M1 Money Multiplier
79	PPIACO	Producer Prices	PPI	Producer Price Index: All Commodities
80	IMPCH	Trade	Imports	Imports from China
81	IMPJP	Trade	Imports	Imports from Japan
82	IMPMX	Trade	Imports	Imports from Mexico
83	IMPCA	Trade	Imports	Imports from Canada
84	IMPGE	Trade	Imports	Imports from Germany
85	IMPUK	Trade	Imports	Imports from UK
86	EXPCH	Trade	Exports	Exports to China
87	EXPJP	Trade	Exports	Exports to Japan
88	EXPMX	Trade	Exports	Exports to Mexico
89	EXPCA	Trade	Exports	Exports to Canada
90	EXPGE	Trade	Exports	Exports to Germany
91	EXPUK	Trade	Exports	Exports to UK
92	BOPGEXP	Trade	Exports	Exports: Goods
93	BOPGIMP	Trade	Imports	Imports: Goods
94	BOPGTB	Trade	Balance	Balance: Goods
95	EXPGS	Trade	Exports	Exports: Services
96	BOPSIMP	Trade	Imports	Imports: Services
97	BOPSTB	Trade	Balance	Balance: Services
98	BOPGSTB	Trade	Balance	Balance: Goods & Services

CONDITIONAL SAMPLING

- **Conditions** = financial-economic priors
- Selected using LASSO a subset of macro conditions based on historical impact on total market drawdown (Wilshire)

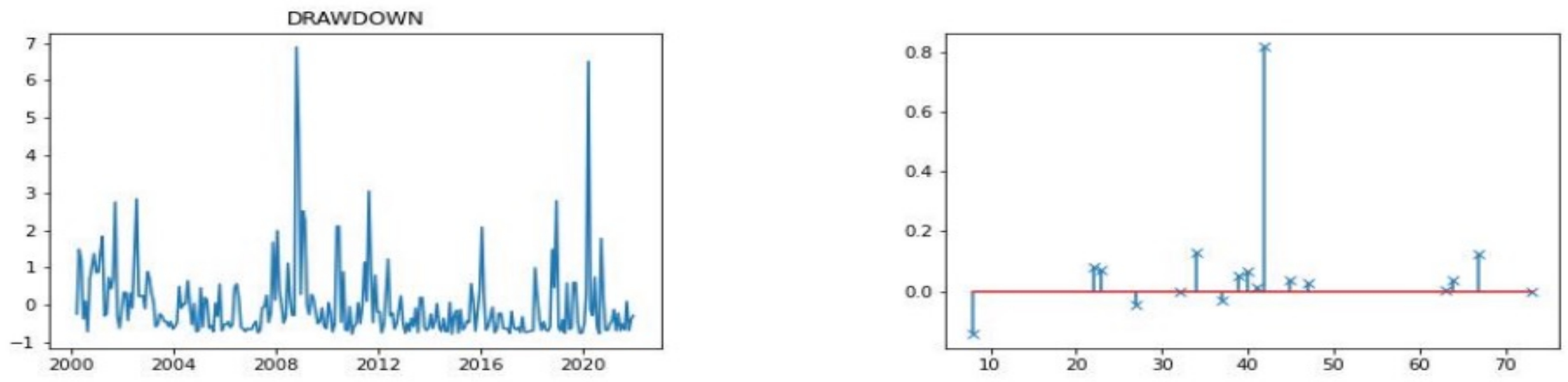


Figure 5: The evolution of ξ of US Stock market index (Wilshire, left), the LASSO coefficients of the conditions (right)

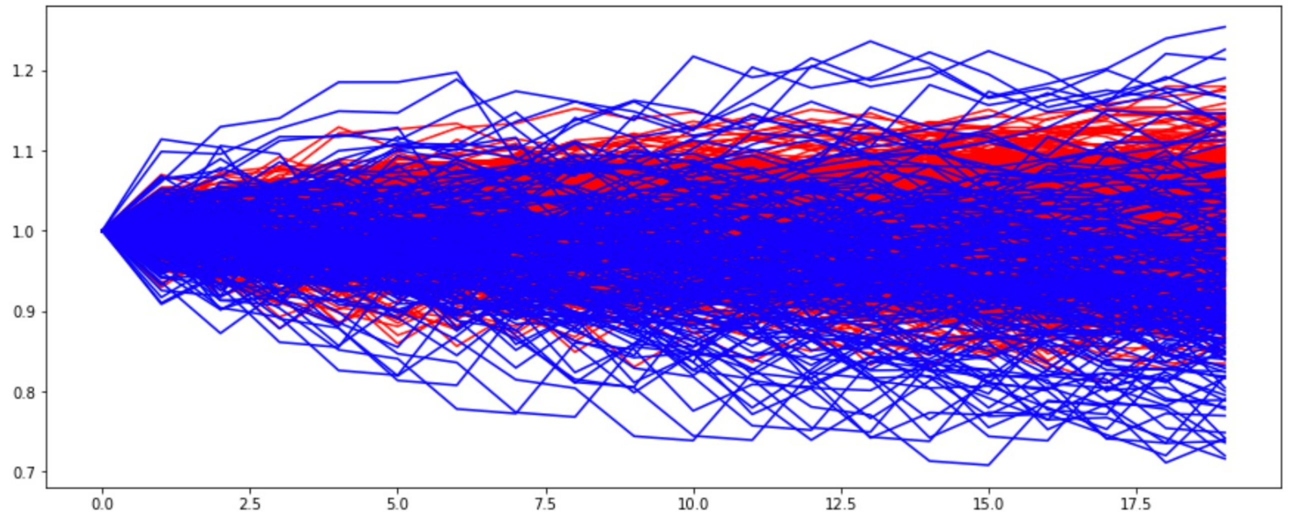
Largest positive contributors to ξ		Largest negative contributors to ξ	
CBOE Volatility Index	0,815680	US Gov't Securities at All Com. Banks	-0,142223
Avg Weekly OT Hours: Manufacturing	0,129751	Long Term Unemployment: 27 WKS	-0,043401
Exports to Mexico	0,126585	JPN/USD Currency Exchange Rate	-0,029723
Univ. of Michigan: Consumer Sentiment	0,079161	Avg Hourly Earnings: Manufacturing	-0,001523
St. Louis Financial Stress Index	0,072814		
CNY/USD Currency Exchange Rate	0,068154		
CAD/USD Currency Exchange Rate	0,053743		
Imports from UK	0,038683		
30-yr Conventional Mortgage Rate	0,037272		
Effective Federal Funds Rate	0,029571		

Table 2: Lasso coefficients of C_i to ξ

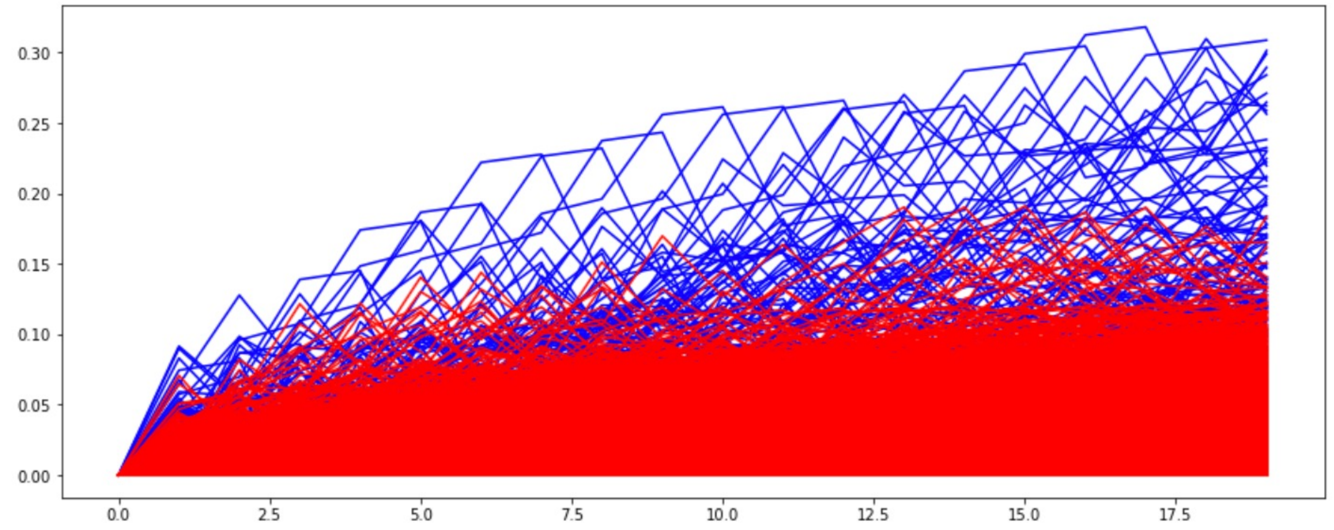
CONDITIONAL SAMPLING

– **Impact of conditions on paths** High **CBOE VIX** (blue) vs. Low (red)

Cum return paths

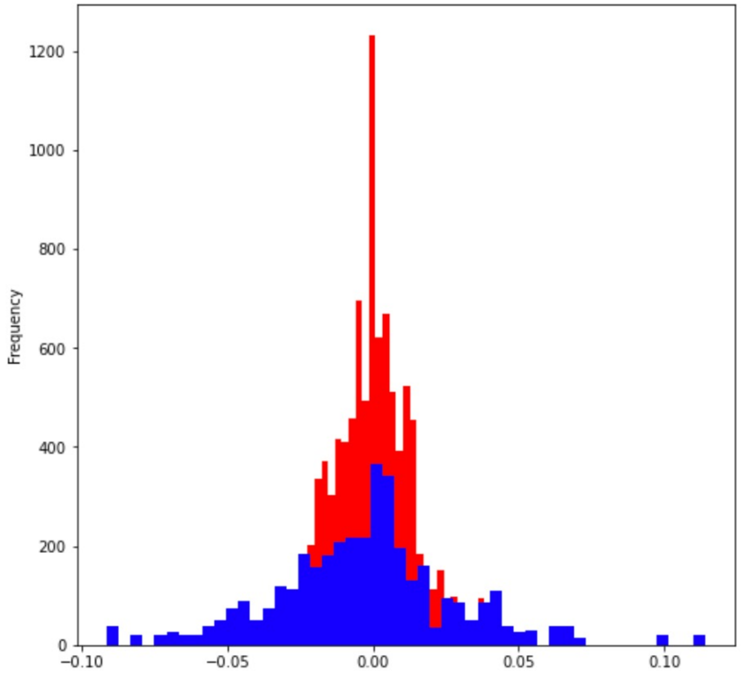


Drawdown paths

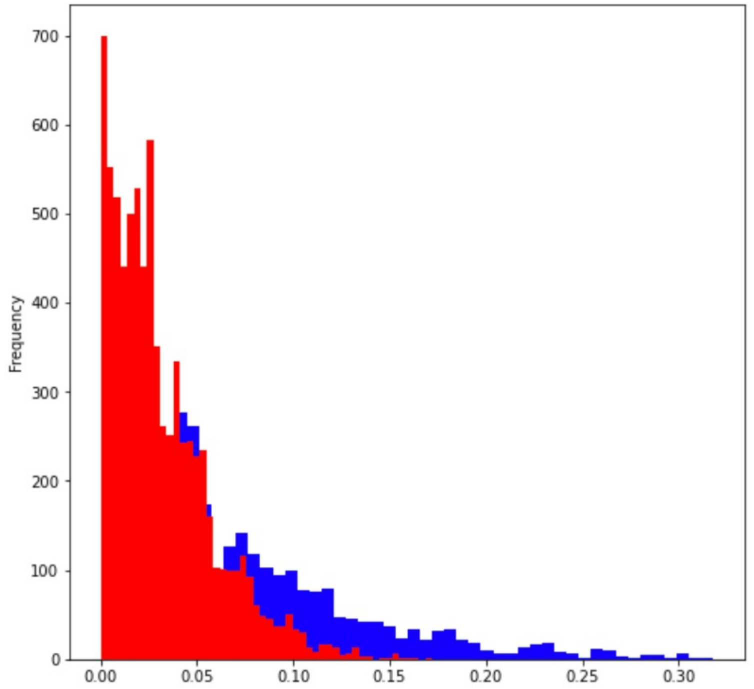


```
AvDD > : 0.07172082622963596
MDD > : 0.31811123778336325
AvDD < : 0.03139465703917589
MDD < : 0.17198107002546248
```

P&L



Drawdown dist

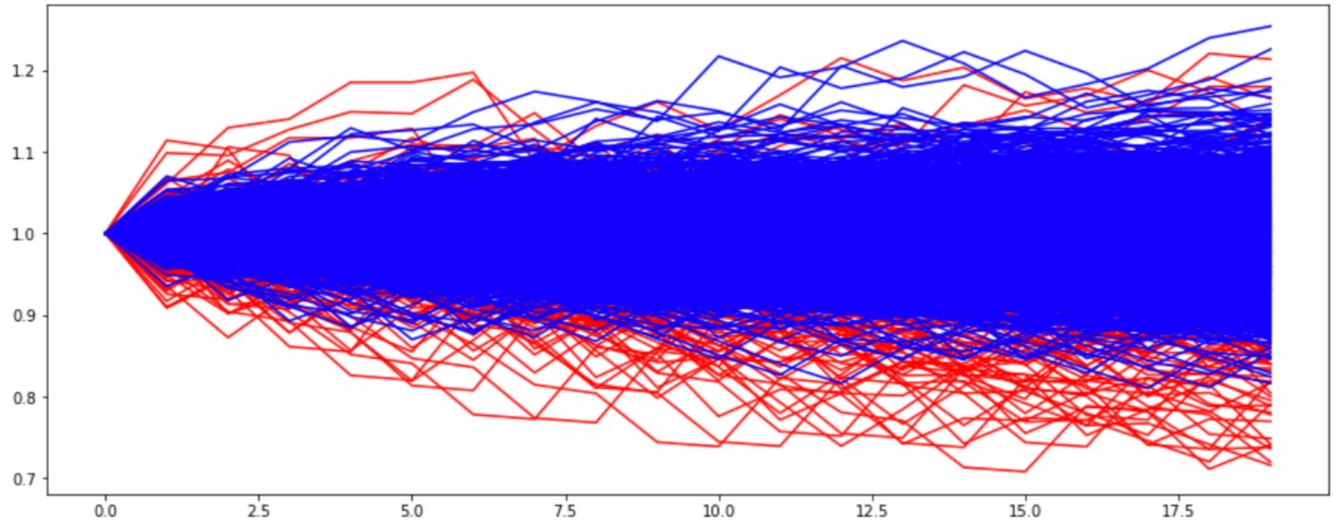


CONDITIONAL SAMPLING

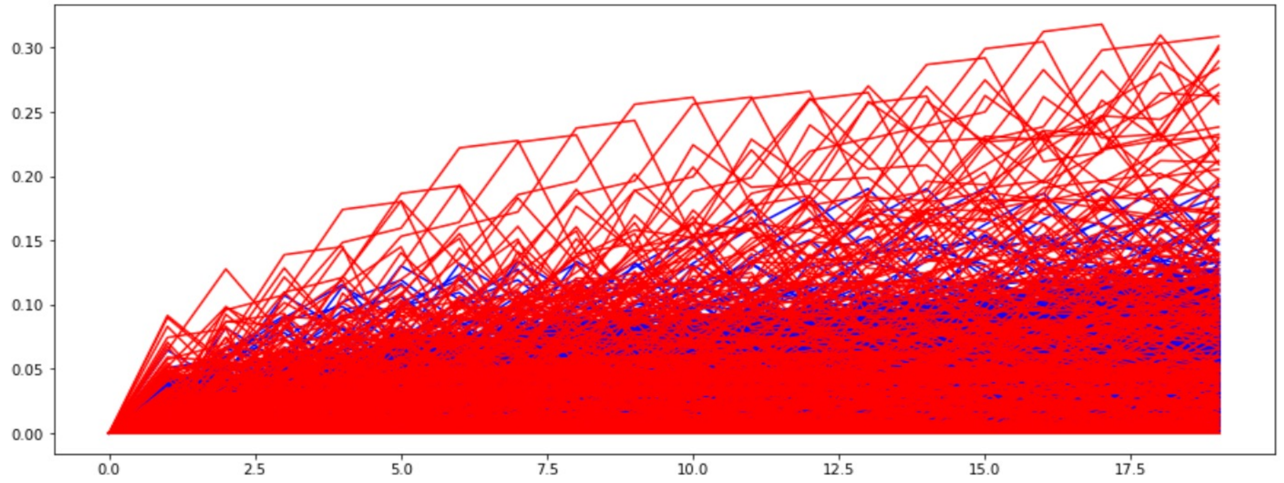
– **Impact of conditions on paths**

High **Consumer Sentiment** (blue) vs. Low (red)

Cum return paths

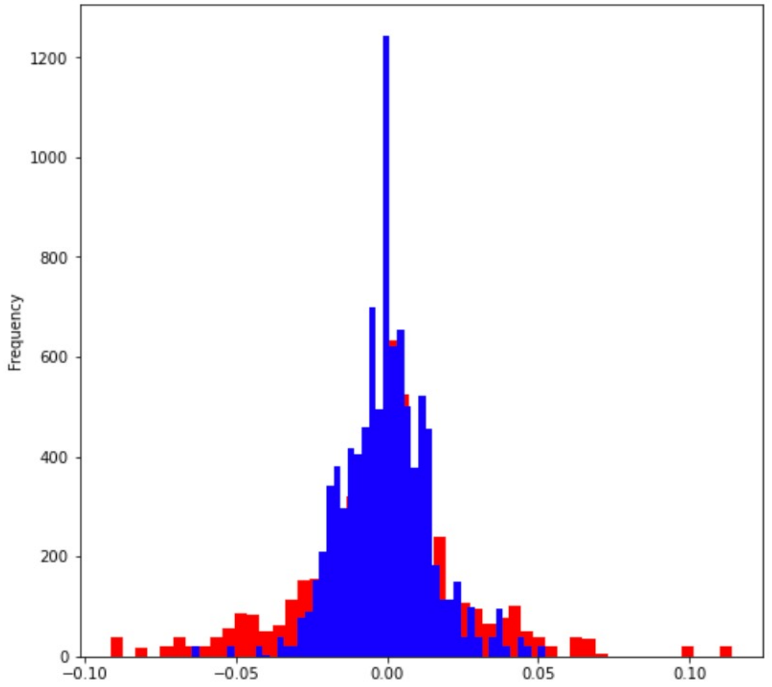


Drawdown paths

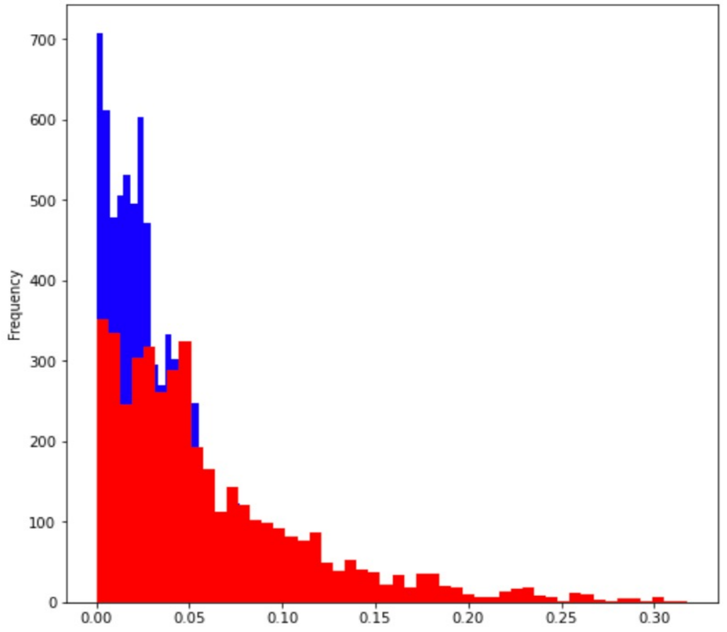


```
AvDD > : 0.032047659984974816
MDD > : 0.1837654572333065
AvDD < : 0.06057656389400842
MDD < : 0.31811123778336325
```

P&L



Drawdown dist



GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

CONDITIONAL SAMPLING

- **Recap** (hopefully makes more sense now):

  - **Goal** != better prediction
  - **Goal** = better simulation (-> *complexity science*)

  In other words:

Coming up with scenarios that might be obvious for the data, but not for the human/modeller !

- **Goal 2** = better understanding of **sensitivities** of optimal portfolios to these conditions (see next slides)

CONDITIONAL SAMPLING

- The aim is to introduce appropriate C to our generative model, such that we can evaluate $\mathbb{P}(X'|C)$ at the current level of C as well as for our own scenarios of C.

- For instance, given the current level of volatility, what do drawdown paths and the optimal portfolio look like, and which positions are most affected if one gradually increases the volatility to levels seen during the GFC or the Covid-19-induced March 2020 meltdown?

- What does one's portfolio look like with current market sentiment, and which positions are likely to be first and mostly affected when sentiment turns sour gradually?

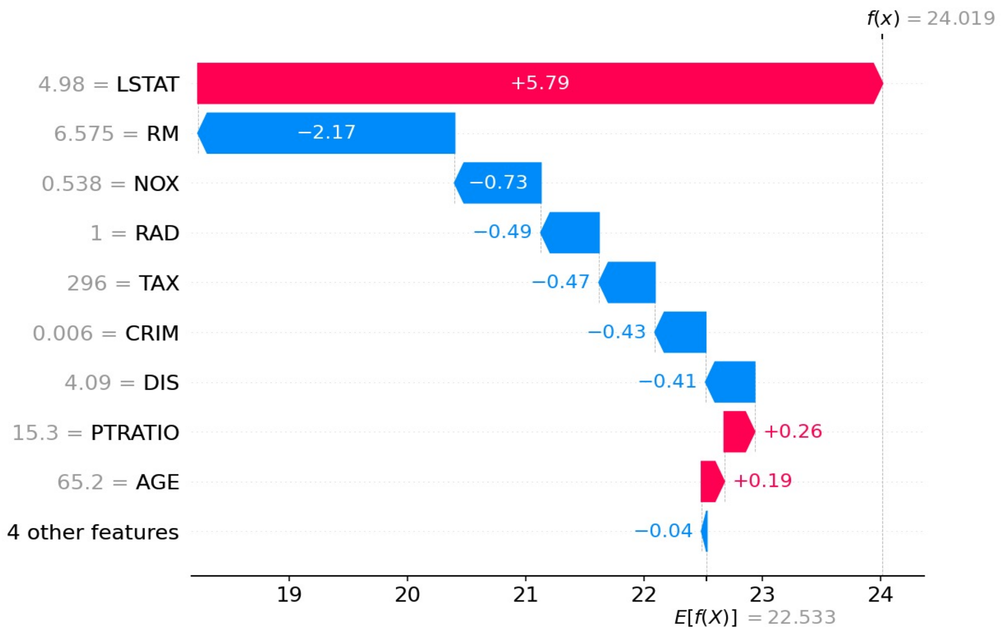
CONDITIONAL SAMPLING

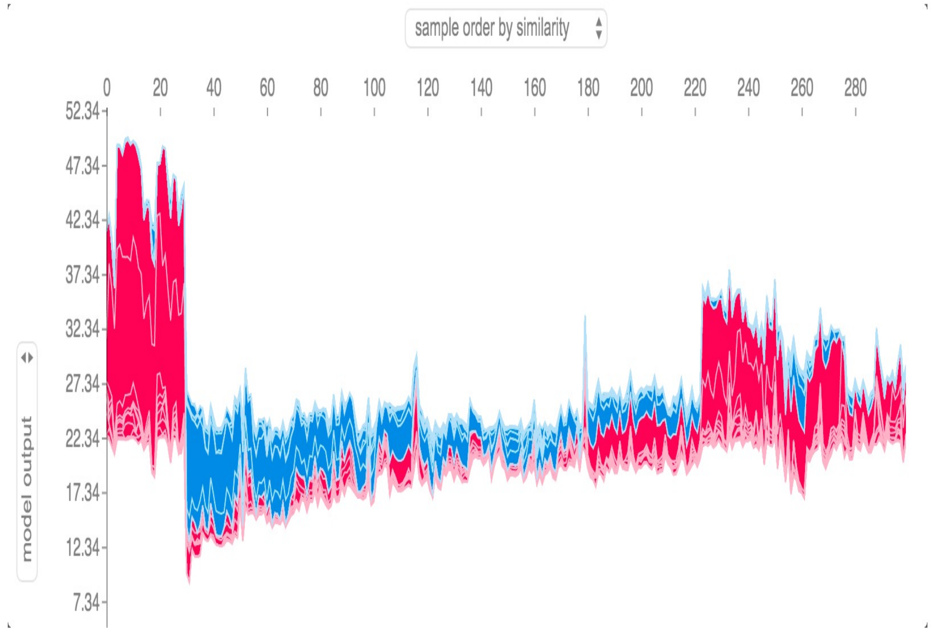
Shapley (SHAP) values:

- Given one set of n_{cond} conditions $C = (C_i)_{i=\{1,\ldots,n_{cond}\}}$, an optimal portfolio can be seen as a linear combination w_d^*, for $d \in D$, where the weights reflect some contribution (of risk, return, drawdown) to the optimal portfolio timeseries $w^* R$ or $w^* \Pi$.
- Given a set of N_s condition sets $\mathcal{C} = (C^k)_{k=\{1,\ldots,N_s\}}$, each set corresponding to a C that generates sequences R or Π, each C will also correspond to a unique optimal portfolio, i.e. for each k. Now we can see the w_k^* as the output, and evaluate the contribution of each condition C_i in C^k to the optimal portfolio. The SHAP values to each w_d^* can then formally be defined as

$$\Phi_i(w_d^*) = \sum_{S \subset [N_s \backslash \{i\}]} \frac{|S|!\,(N_s - |S| - 1)!}{N_s!} \left(w_d^*(S \cup \{i\}) - w_d^*(S)\right)$$

- This is the SHAP Φ_i for condition i in C in terms of optimal weight w_d^*. Intuitively, for the N_s optimal portfolios we evaluate all the subsets S where condition i did not contribute to the optimal portfolio $w_d^*(S)$ and compare with the optimal portfolios where it was $w_d^*(S \cup \{i\})$. The average contribution of this condition to the optimal weight thus constitutes the SHAP value. This allows for visualizations of the *conditional* optimal portfolios, such as waterfall and beeswarm plots, that are popular explainable machine learning tools for applications in deep learning and computer vision.



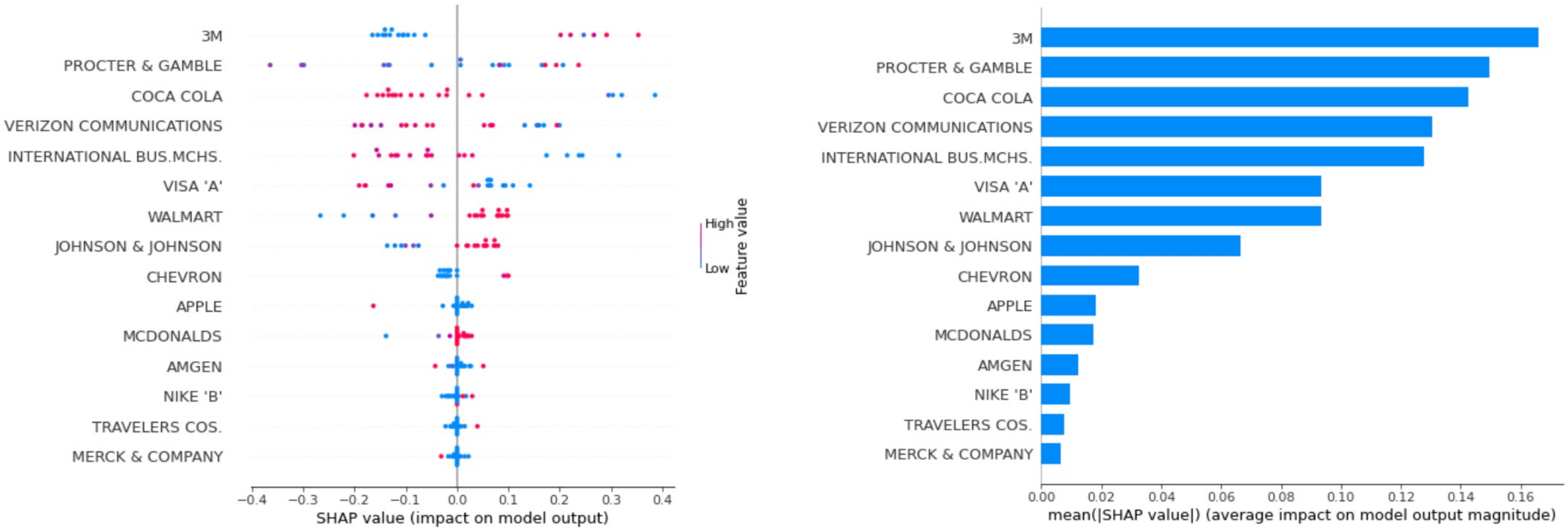


GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

CONDITIONAL SAMPLING: DOW 30 EXAMPLE

Impact of VIX on optimal DOW portfolio (simple conditional bootstrap)

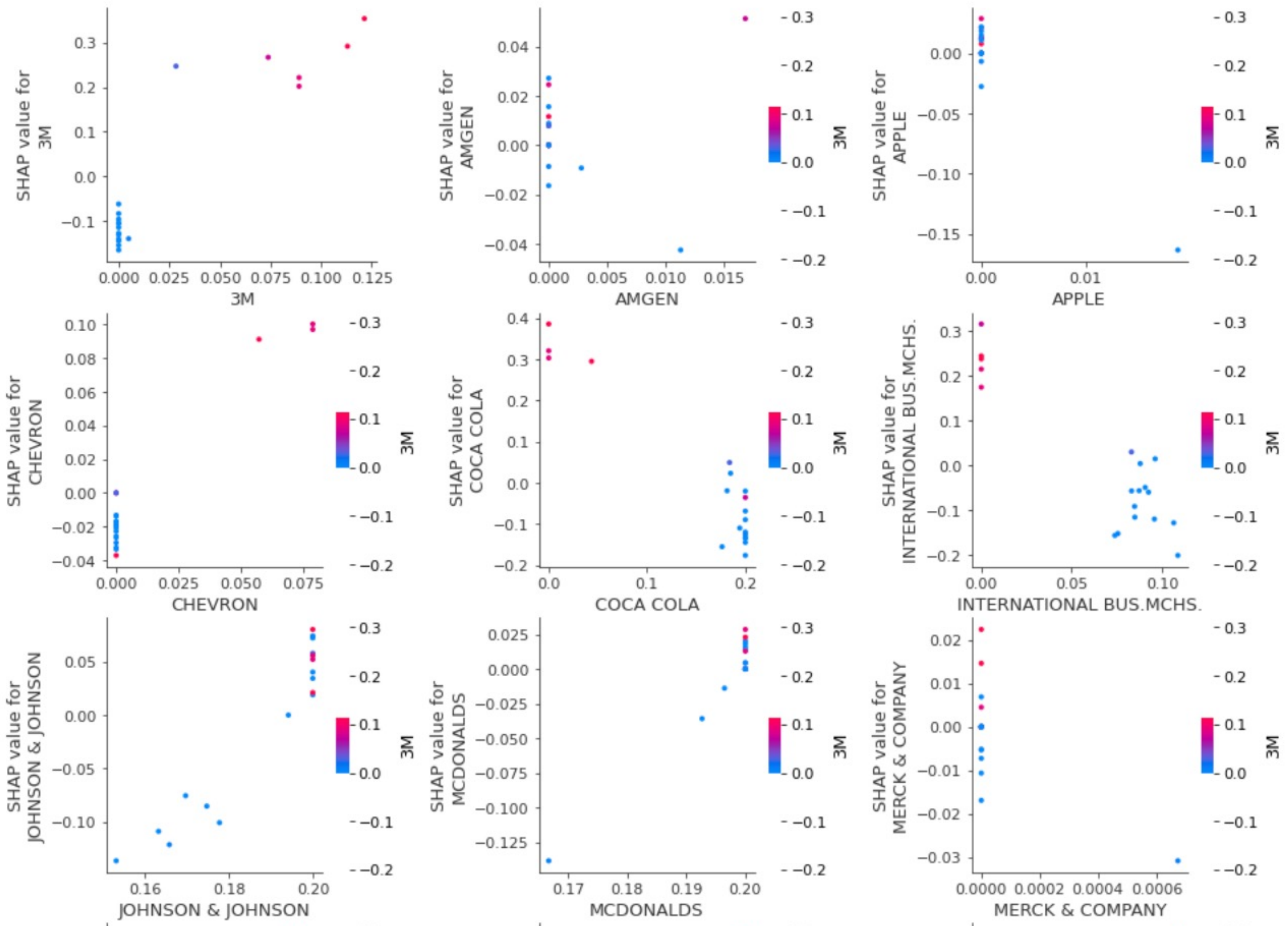


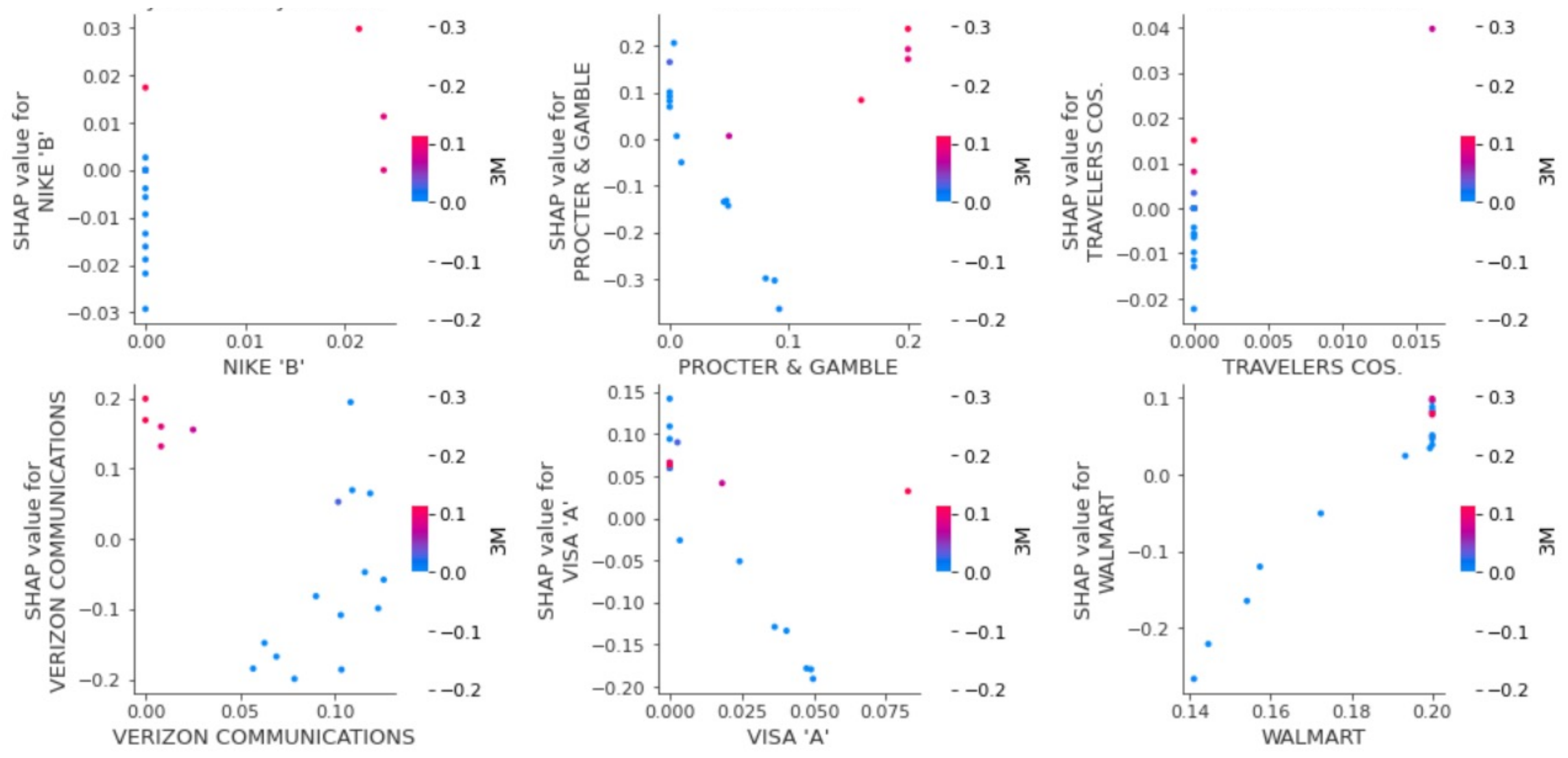
GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

CONDITIONAL SAMPLING: DOW 30 EXAMPLE

Impact of VIX on optimal DOW portfolio (simple conditional bootstrap)



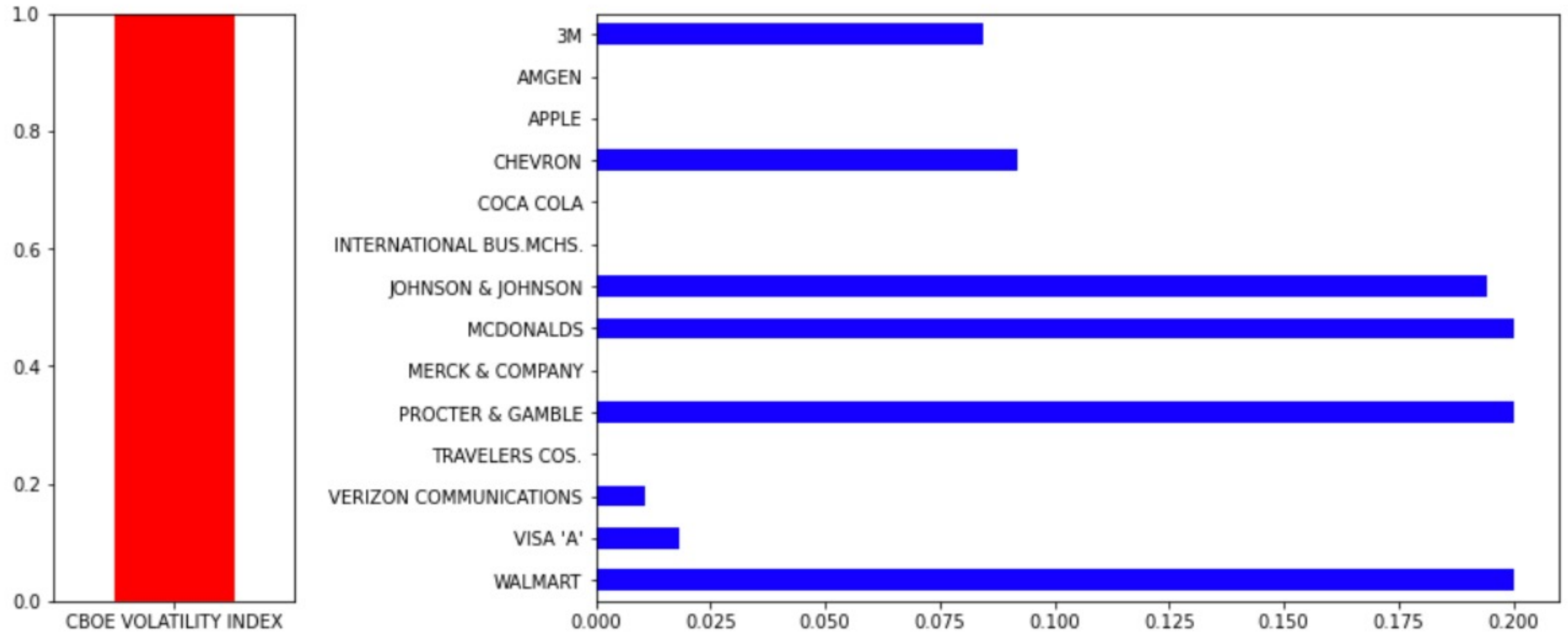


GHENT UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

CONDITIONAL SAMPLING: DOW

Impact of VIX on optimal DOW portfolio (simple conditional bootstrap)



NEXT STEPS

NEXT STEPS

- **Low-hanging fruit (first on the agenda):**
  - **Code:**
    - 3 open merge requests (MRs) on Gitlab:
      - Variational Autoencoder Architecture implementation (on R for now, ELBO loss)
      - General Condition object implementation
      - Conditional Weighted Bootstrap (benchmark) implementation
    - Proposed next MRs:
      - Sig-MMD (first try)
      - GMMN (Sig-MMD)
      - Quant-GAN (as computational benchmark)
- **Bigger questions / conceptual:**
  - **Input representation**: Further develop input repr for Xi; and link with portfolio paths Pi (inverse transform?)
  - **Loss function**: Sig-MMD for drawdown process moment matching
  - **Conditions**: Develop macro backdrop to train conditional architecture; connect with macro econ collaborator / work with thesis students?

Emiel Lemahieu

Quantitative Researcher

emiel.lemahieu@investsuite.com

www.linkedin.com/in/emiel-lemahieu