Quant GANs: Deep Generation of Financial Time Series

Hyelin Choi

Department of Mathematics

Sungkyunkwan University

Dec 11, 2024

Table of Contents

Table of Contents

What is S&P 500?

S&P 500 Stock Price Path

S&P 500 index data, Google Finance

Return

Let S_t be the stock price at time t. There are two types of returns.

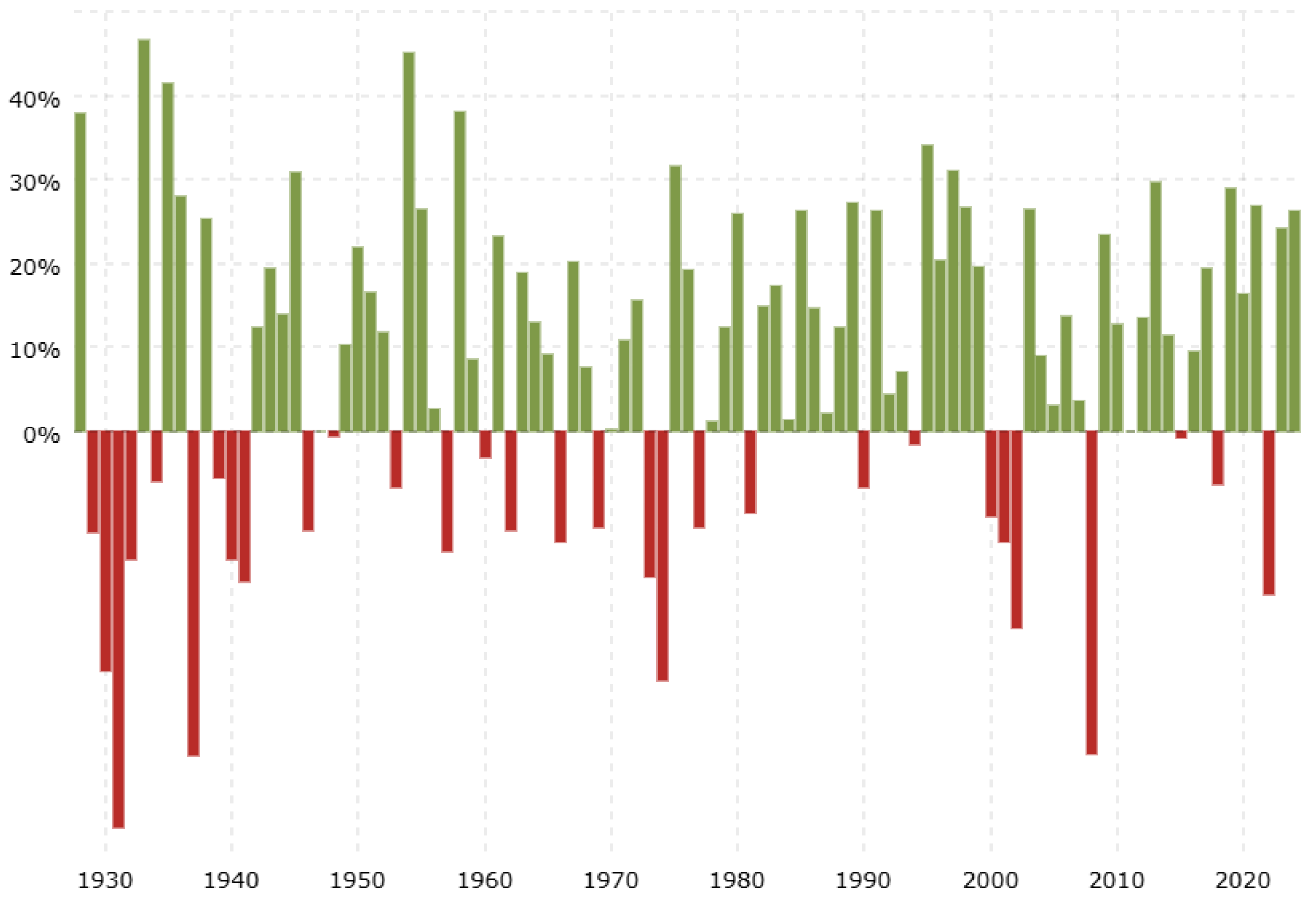
- Relative return

$$R_t = (S_t - S_{t-1})/S_{t-1}$$

- Log-return

$$R_t = \log \frac{S_t}{S_{t-1}}$$

S&P 500 Returns

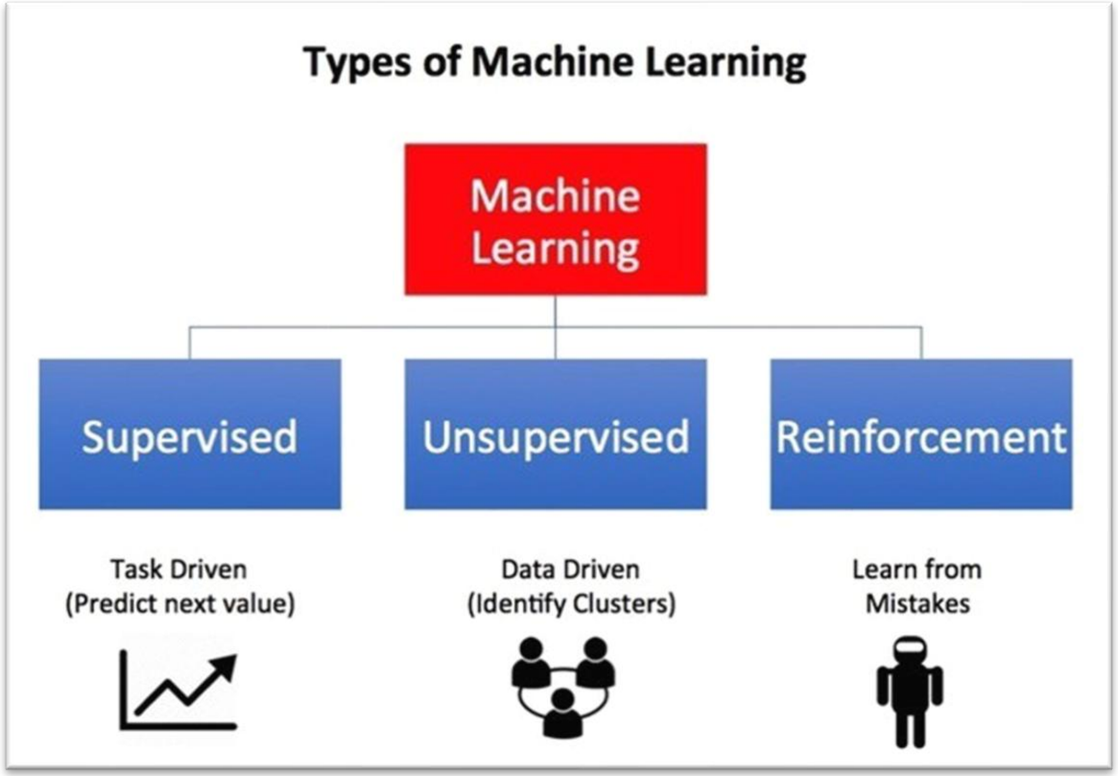


S&P 500 Historical Annual Returns

https://www.macrotrends.net/2526/sp-500-historical-annual-returns

Applications of QuantGANs

1. QuantGANs enable us to generate realistic log-return paths for the S&P 500 across multiple scenarios.



Applications of QuantGANs

2. We can predict **future prices** or **trends** of assets based on past behavior.

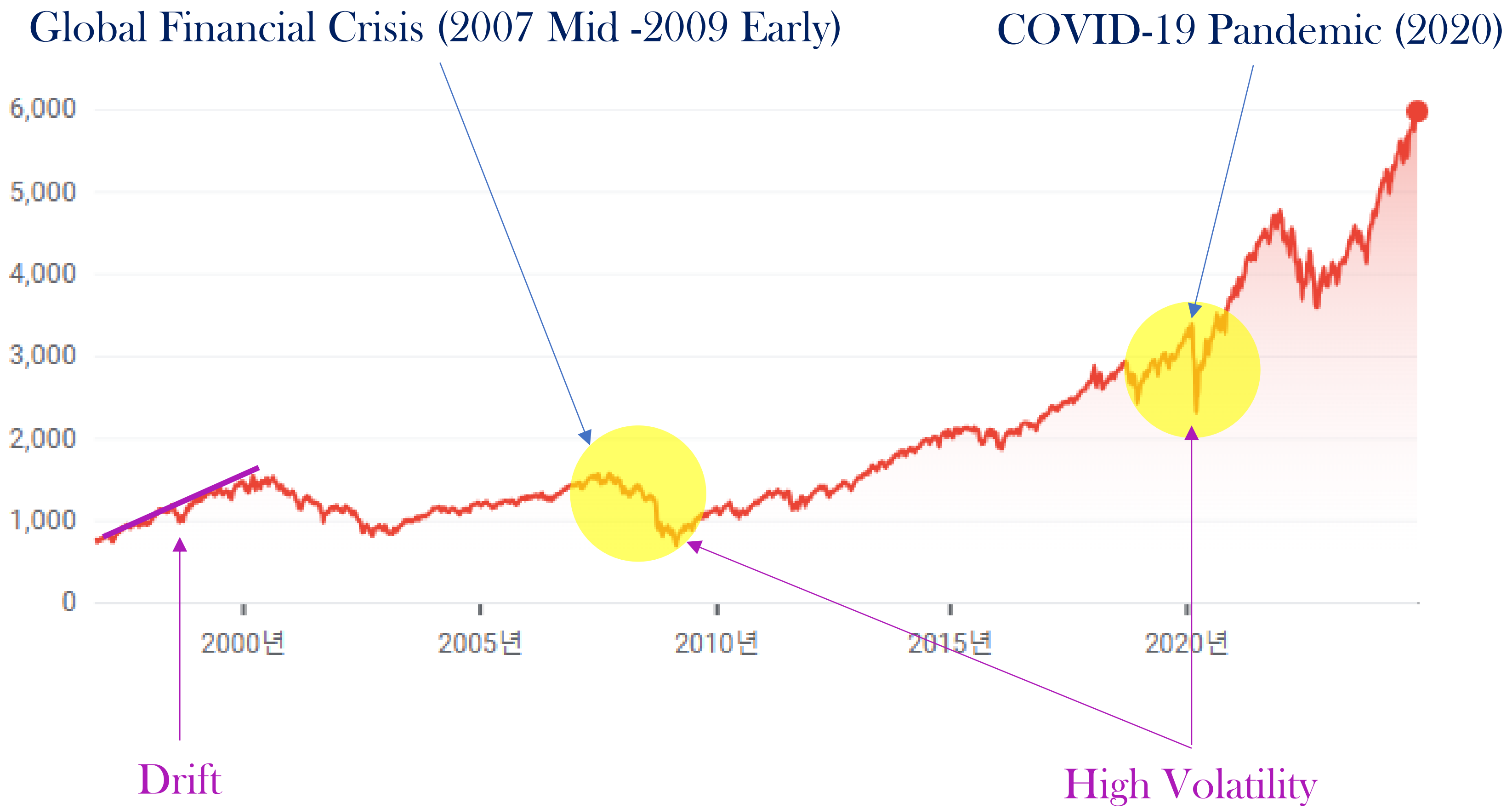
Long-range dependency

What key information should QuantGANs capture from log-return data?

<u>Long-range dependency</u>

Long-range dependency means correlations persist across distant time lags.

Volatility, Drift, and Innovation



S&P 500 index data, Google Finance

Geometric Brownian Motion(GBM)

Geometric Brownian Motion (GBM) is a mathematical model used to describe the random behavior of asset prices over time.

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$ where W_t: Brownian motion, μ: drift and σ: volatility

$$R_t = \log\frac{S_t}{S_{t-1}} = \mu^{-} + \sigma(W_t - W_{t-1})$$ where $\mu^{-} = \mu - \frac{\sigma^2}{2}$

Drift $\mu_{t,\theta}$

Volatility $\sigma_{t,\theta}$

Innovation $\varepsilon_{t,\theta} \sim N(0,1)$

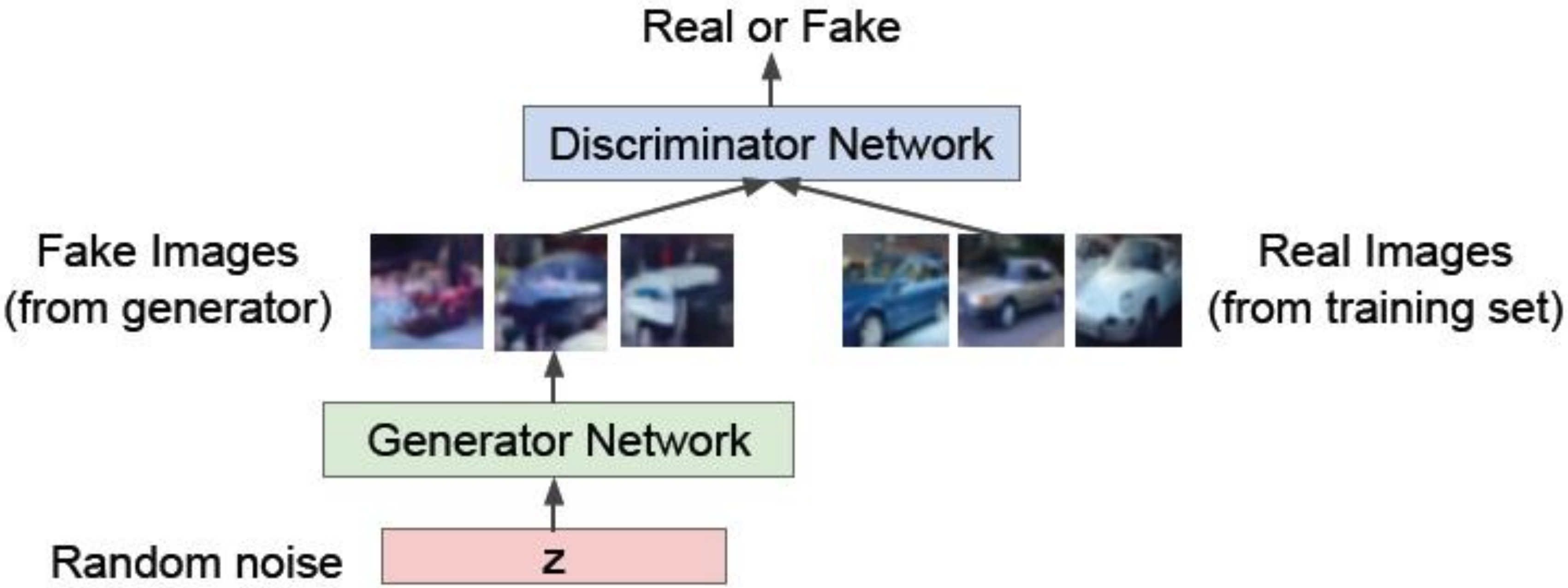
Table of Contents

Construction of QuantGANs

- Quant GANs
  - <u>Generator</u>

    Stochastic Volatility Neural Networks (SVNNs)
  - <u>Discriminator</u>

Generative Adversarial Networks
대립적인



Yunhuijang, 2022.7.20, https://process-mining.tistory.com/169

Generative Adversarial Networks

i.i.d. Gaussian noise process

- The random variable **Z** represents the **noise** prior and **X** the **targeted** (or data) random variable.

S&P 500 Log-return Data

- The goal of GANs is to train a network $g\colon \mathbb{R}^{N_z} \times \Theta^{(g)} \to \mathbb{R}^{N_x}$ such that the induced random variable $g_\theta(Z) \coloneqq g_\theta \circ Z$ for some parameter $\theta \in \Theta^{(g)}$ and the targeted random variable X have the same distribution, i.e. $\boldsymbol{g_\theta(Z) \overset{\mathrm{d}}{=} X}$.

Generative Adversarial Networks

Loss function of GANs

$$\mathcal{L}(\theta, \eta) := \mathbb{E}\left[\log(d_\eta(X))\right] + \mathbb{E}\left[\log(1 - d_\eta(g_\theta(Z)))\right]$$
$$= \mathbb{E}\left[\log(d_\eta(X))\right] + \mathbb{E}\left[\log(1 - d_\eta(\tilde{X}_\theta))\right] .$$

η: parameter of discriminator
θ: parameter of generator
$d_\eta(\cdot)$: function of discriminator
X: targeted r.v.
$\tilde{X}_\theta$: generated r.v.

Step 1

Let the 1 represent real data and 0 represent fake data.
The discriminator's parameter $\eta \in \Theta^{(d)}$ are chosen to maximize the function $\mathcal{L}(\theta,\cdot), \theta \in \Theta^{(g)}$.

Step 2

The generator's parameters $\theta \in \Theta^{(g)}$ are trained to minimized the probability of generated samples being identified as such and not from the data distribution.

Generative Adversarial Networks

We get the min-max problem

$$\min_{\theta \in \Theta^{(g)}} \max_{\eta \in \Theta^{(d)}} \mathcal{L}(\theta, \eta)$$

which refer to as the GAN objective.

Table of Contents

Construction of QuantGANs

Quant GANs

<u>Generator</u>

Stochastic Volatility
Neural Networks (SVNNs)

<u>Discriminator</u>

TCN

$$\text{Log-return neural process} \begin{cases} \text{Volatility TCN } \sigma_{t,\theta} \\ \text{Drift TCN } \mu_{t,\theta} \\ \text{Innovation NN } \varepsilon_{t,\theta} \end{cases}$$

Advantages of TCNs

1) TCNs are able to capture long-range dependencies in sequences.

2) TCNs with skip connections have an advantage of avoiding exponentially vanishing gradients.

Construction of TCNs

TCNs are neural network models primarily designed to efficiently handle sequential data, such as time series.

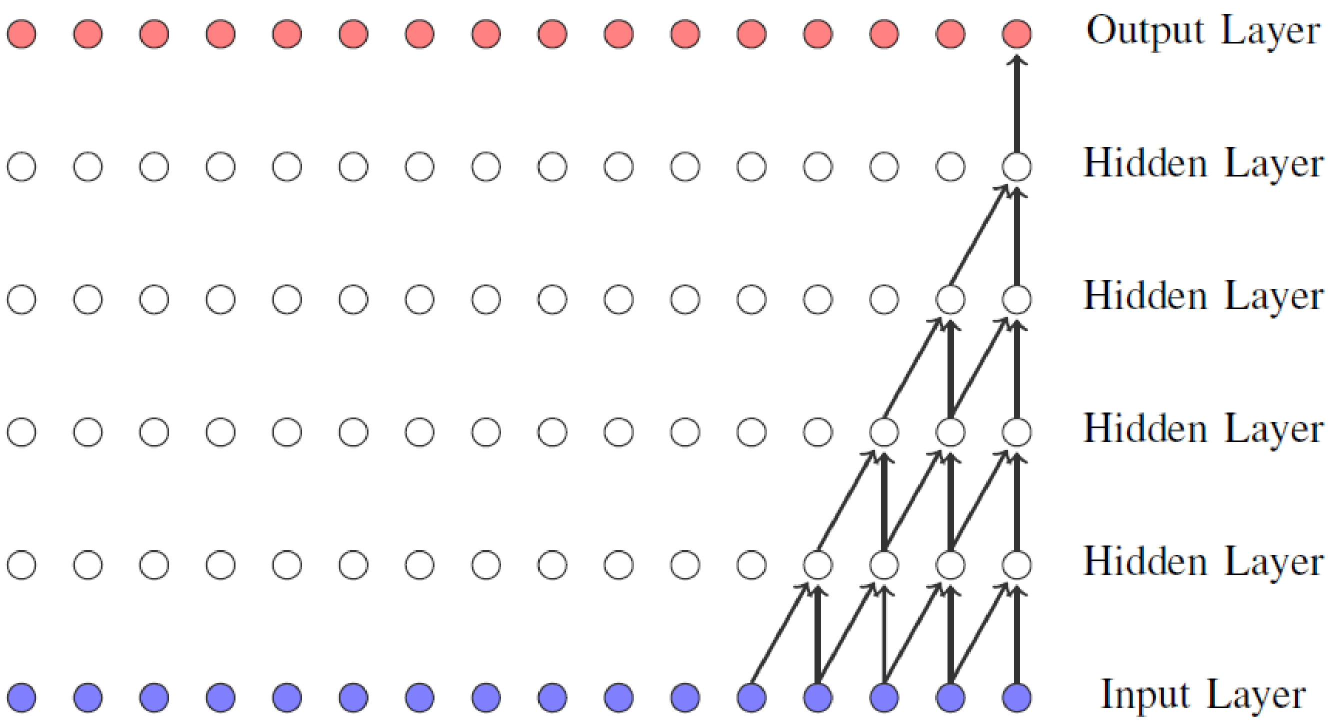
- Constructions

Dilated causal convolutions = Causal convolutions + Dilated convolutions

인과 확장

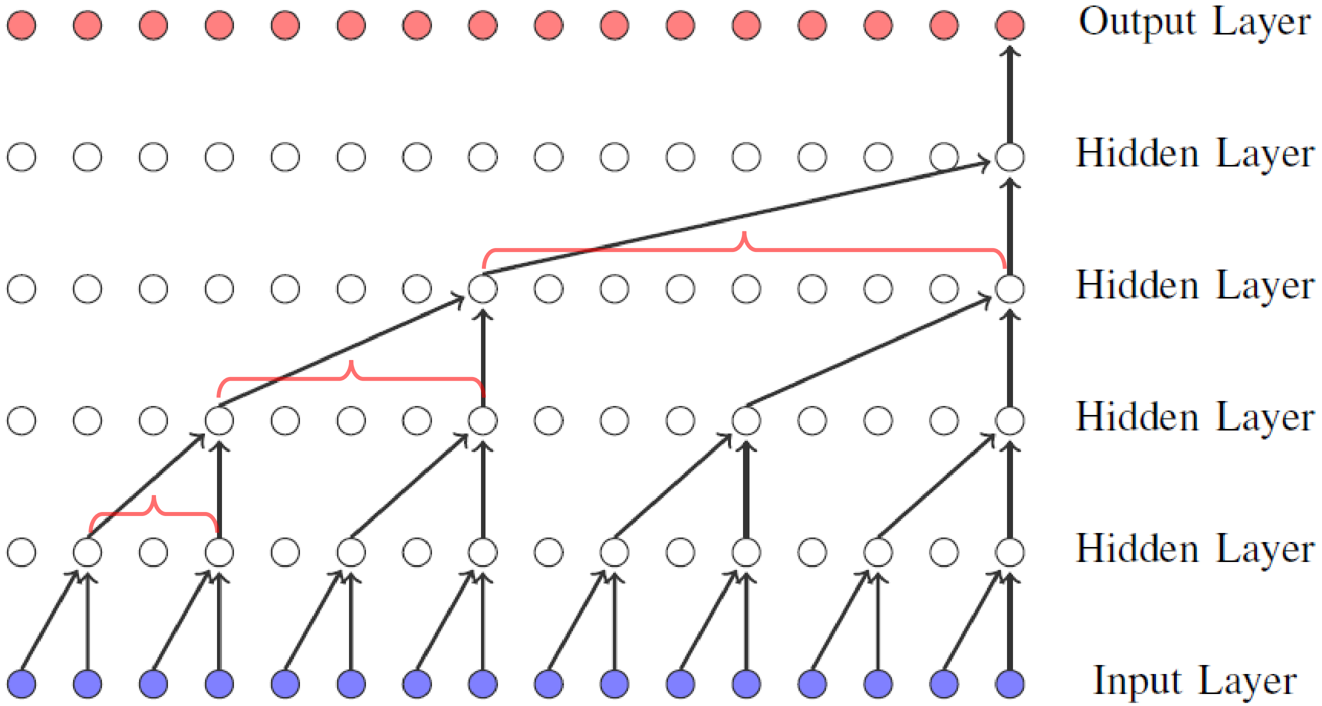
Causal Convolution

Causal convolutions are convolutions, where output only depends on past sequence elements.
인과

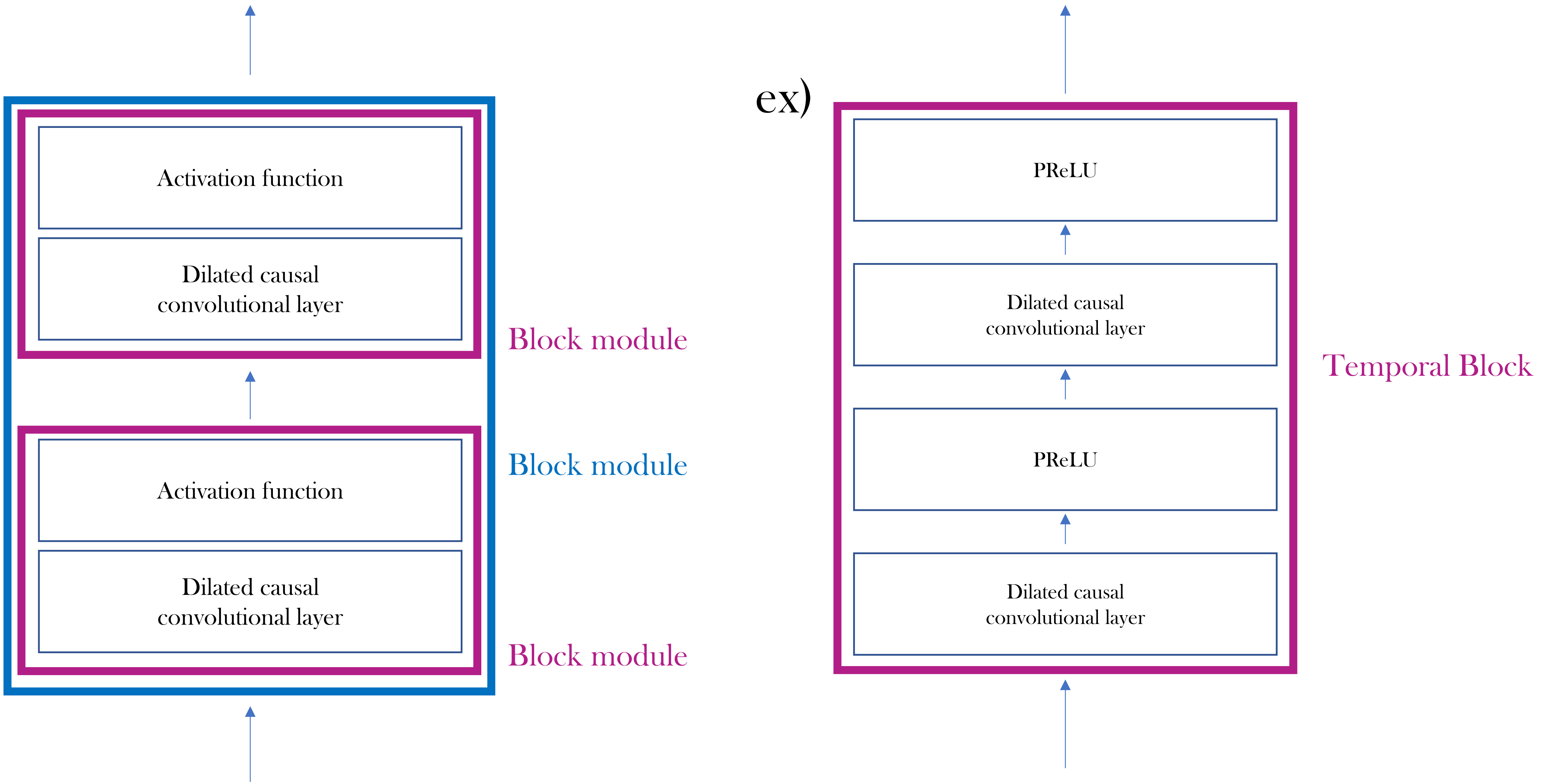


Dilated Convolution

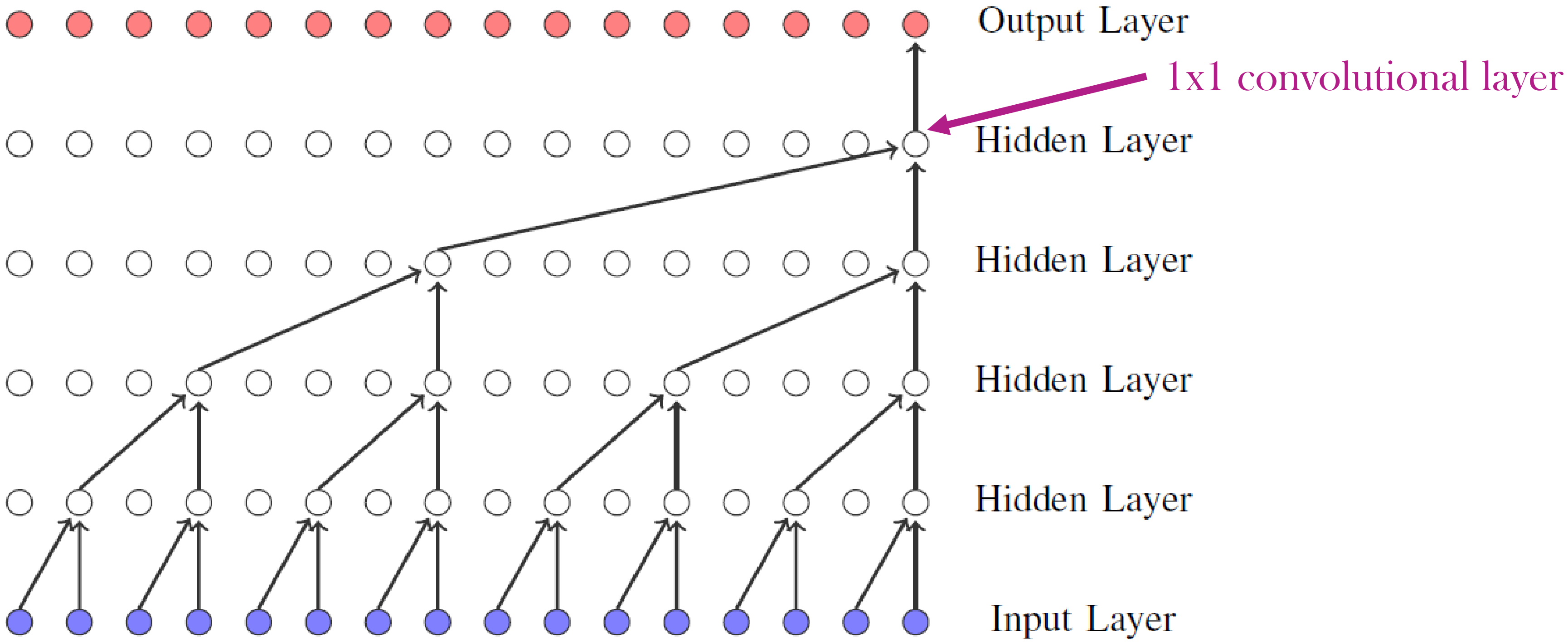
Dilated convolutions are convolutions ‘with holes’.
확장



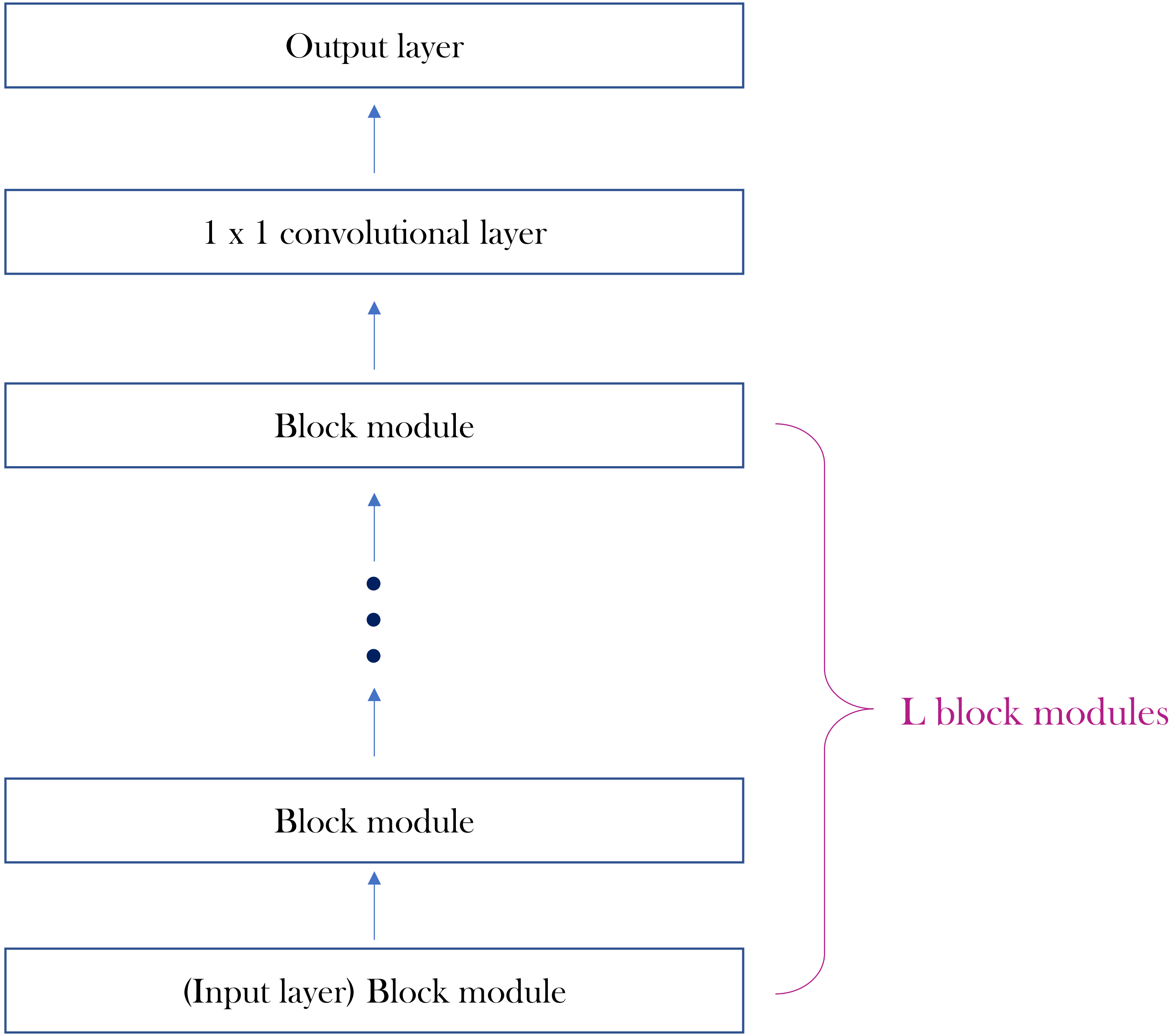
Block Module



1x1 Convolutional Layer



Temporal Convolutional Network



Skip Connections

[Skip Connections]

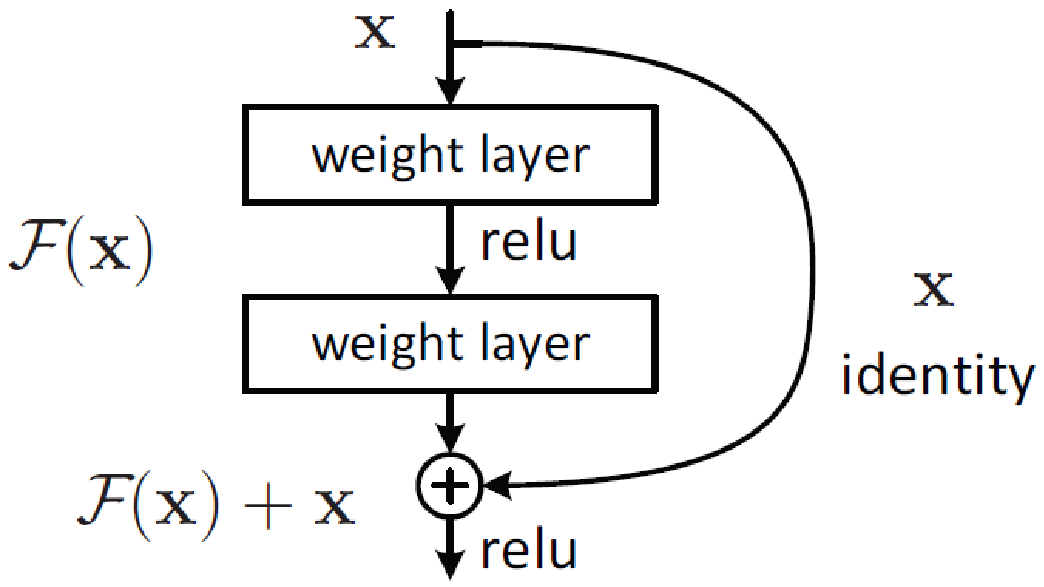


Figure 2. Residual learning: a building block.

TCN with Skip Connections

Definition 3.15 (TCN with skip connections)**.** Assume the notation from Definition 3.10 and for $N_{skip} \in \mathbb{N}$ let

$$\gamma_l : \mathbb{R}^{N_{l-1} \times T_{l-1}} \to \mathbb{R}^{N_l \times T_l} \times \mathbb{R}^{N_{skip} \times T_L} \quad \text{for } l \in \{1, \dots, L\}$$

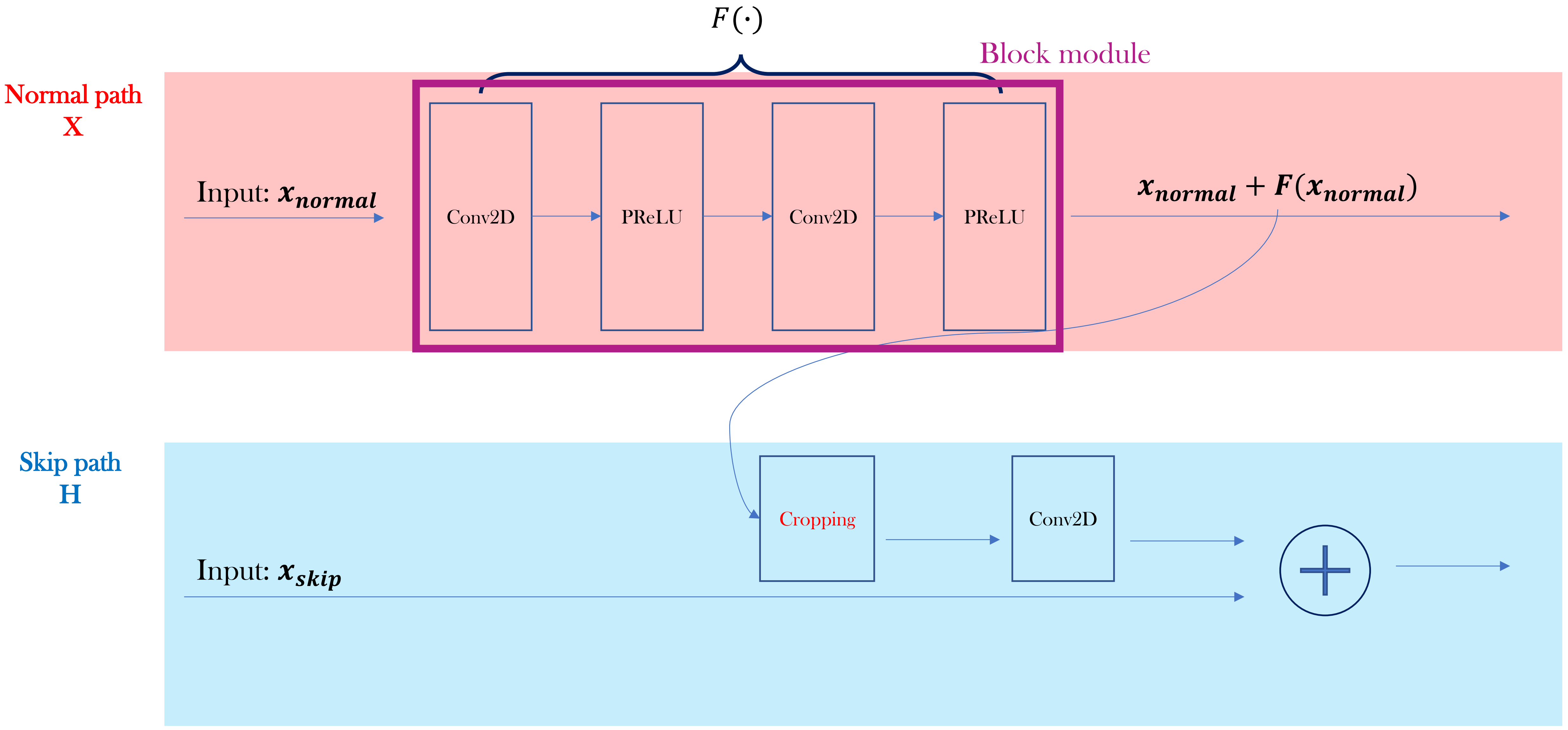
denote block modules. Moreover, let γ be a block module with arguments $(N_{skip}, N_{L+1}, 0)$. If the output $Y \in \mathbb{R}^{N_{L+1} \times T_L}$ of a TCN $f : \mathbb{R}^{N_0 \times T_0} \times \Theta \to \mathbb{R}^{N_{L+1} \times T_L}$ is defined recursively by

$$\left(X^{(l)}, H^{(l)}\right) = \gamma_l \left(X^{(l-1)}\right) \quad \text{for } l \in \{1, \dots, L\}$$

$$Y = \gamma \left(\sum_{l=1}^{L} H^{(l)} \right) ,$$

where $X^{(0)} \in \mathbb{R}^{N_0 \times T_0}$, then f is called a *temporal convolutional network with skip connections*.

TCN with Skip Connections



Vanilla TCN with Skip Connection

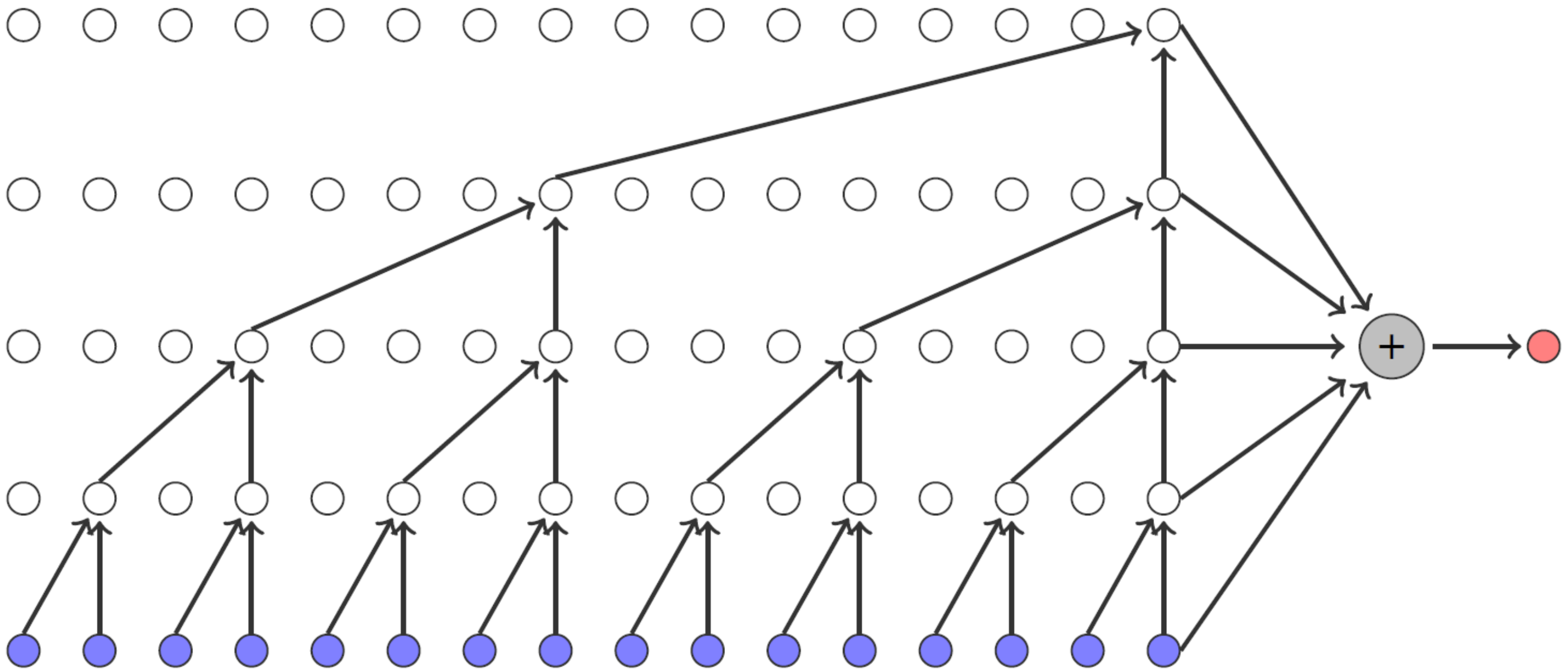


Figure 7: Vanilla TCN with skip connections.

Table of Contents

Overview

Quant GANs

Generator

Stochastic Volatility
Neural Networks (SVNNs)

TCN

Discriminator

TCN

Induced process of SVNN

Log-return neural process
(log-return NP)

Log-return Neural Process

Notation 4.4. Consider a stochastic process $(X_t)_{t\in\mathbb{Z}}$ parametrized by some $\theta \in \Theta$. For $s, t \in \mathbb{Z},\ s \leq t$, we write

$$X_{s:t,\theta} := (X_{s,\theta}, \ldots, X_{t,\theta})$$

and for an ω-realization

$$X_{s:t,\theta}(\omega) := (X_{s,\theta}(\omega), \ldots, X_{t,\theta}(\omega)) \in \mathbb{R}^{N_X \times (t-s+1)}.$$

We can now introduce the concept of neural (stochastic) processes.

Log-return Neural Process

Definition 4.5 (Neural process). Let $(Z_t)_{t\in\mathbb{Z}}$ be an i.i.d. noise process with values in $\mathbb{R}^{N_Z}$ and $g : \mathbb{R}^{N_Z \times T^{(g)}} \times \Theta^{(g)} \to \mathbb{R}^{N_X}$ a TCN with RFS $T^{(g)}$ and parameters $\theta \in \Theta^{(g)}$. A stochastic process $\tilde{X}$, defined by

$$\begin{aligned} \tilde{X} : \Omega \times \mathbb{Z} \times \Theta^{(g)} &\to \mathbb{R}^{N_X} \\ (\omega, t, \theta) &\mapsto g_\theta(Z_{t-(T^{(g)}-1):t}(\omega)) \end{aligned}$$

such that $\tilde{X}_{t,\theta} : \Omega \to \mathbb{R}^{N_X}$ is a $\mathcal{F} - \mathcal{B}(\mathbb{R}^{N_X})$-measurable mapping for all $t \in \mathbb{Z}$ and $\theta \in \Theta^{(g)}$, is called *neural process* and will be denoted by $\tilde{X}_\theta := (\tilde{X}_{t,\theta})_{t\in\mathbb{Z}}$.

Log-return Neural Process

The GAN objective for stochastic processes can be formulated as

Monte Carlo estimate

$$\min_{\theta \in \Theta^{(g)}} \max_{\eta \in \Theta^{(d)}} \mathcal{L}(\theta, \eta)\,,$$

where

$$\mathcal{L}(\theta, \eta) := \mathbb{E}\left[\log\left(d_\eta\left(X_{1:T^{(d)}}\right)\right)\right] + \mathbb{E}\left[\log\left(1 - d_\eta\left(\tilde{X}_{1:T^{(d)},\theta}\right)\right)\right]$$

and $X_{1:T^{(d)}}$ and $\tilde{X}_{1:T^{(d)},\theta}$ denote the real and the generated process, respectively.

Log-return Neural Process

Definition 5.1 (Log return neural process). Let $Z = (Z_t)_{t\in\mathbb{Z}}$ be $\mathbb{R}^{N_Z}$-valued i.i.d. Gaussian noise, $g^{(\mathrm{TCN})} : \mathbb{R}^{N_Z \times T^{(g)}} \times \Theta^{(\mathrm{TCN})} \to \mathbb{R}^{2N_X}$ a TCN with RFS $T^{(g)}$ and $g^{(\epsilon)} : \mathbb{R}^{N_Z} \times \Theta^{(\epsilon)} \to \mathbb{R}^{N_X}$ be a network. Furthermore, let $\alpha \in \Theta^{(\mathrm{TCN})}$ and $\beta \in \Theta^{(\epsilon)}$ denote some parameters. A stochastic process R, defined by

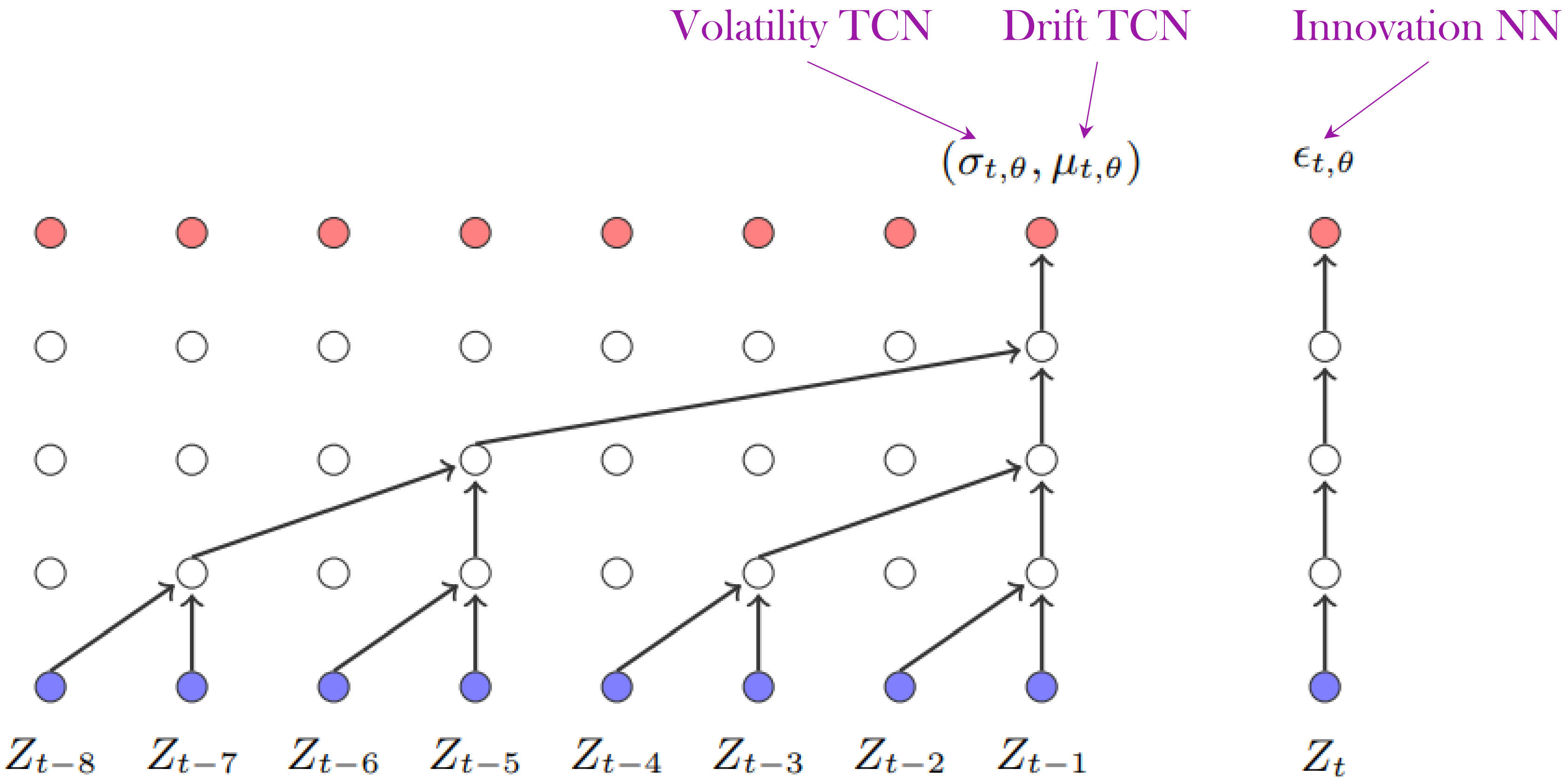
$$\begin{aligned} R : \Omega \times \mathbb{Z} \times \Theta^{(\mathrm{TCN})} \times \Theta^{(\epsilon)} &\to \mathbb{R}^{N_X} \\ (\omega, t, \alpha, \beta) &\mapsto \left[\sigma_{t,\alpha} \odot \epsilon_{t,\beta} + \mu_{t,\alpha}\right](\omega) \ , \end{aligned}$$

where $\odot$ denotes the Hadamard product and

$$\begin{aligned} h_t &:= g_\alpha^{(\mathrm{TCN})}\left(Z_{t-T^{(g)}:(t-1)}\right) \\ \text{Volatility TCN} \qquad \sigma_{t,\alpha} &:= |h_{t,1:N_X}| \\ \text{Drift TCN} \qquad \mu_{t,\alpha} &:= h_{t,(N_X+1):2N_X} \\ \text{Innovation NN} \qquad \epsilon_{t,\beta} &:= g_\beta^{(\epsilon)}(Z_t) \ , \end{aligned}$$

is called *log return neural process*. The generator architecture defining the log return NP is called *stochastic volatility neural network (SVNN)*. The NPs $\sigma_\alpha := (\sigma_{t,\alpha})_{t\in\mathbb{Z}}$, $\mu_\alpha := (\mu_{t,\alpha})_{t\in\mathbb{Z}}$ and $\epsilon_\beta := (\epsilon_{t,\beta})_{t\in\mathbb{Z}}$ are called *volatility, drift* and *innovation NP*, respectively.

Log-return Neural Process



Structure of the SVNN architecture. The volatility and drift component are generated by inferring the latent process $Z_{t-8:t-1}$ through the TCN, whereas the innovation is generated by inferring Z_t.

Table of Contents

Numerical Results

QuantGANs Using Pure TCN

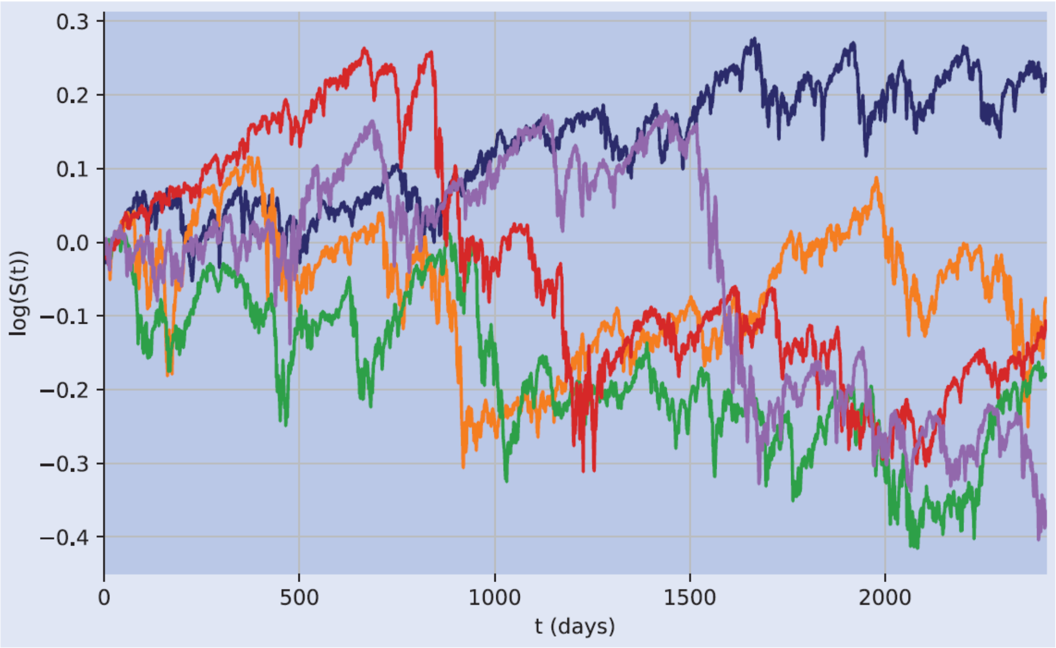


Figure A1. Five generated driftless log paths.

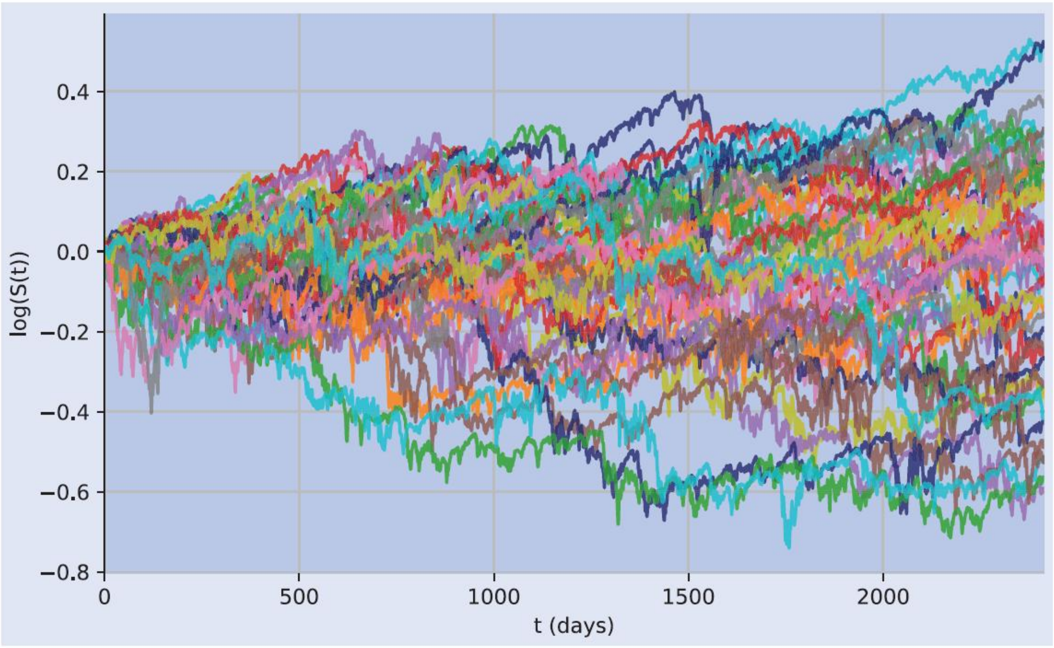
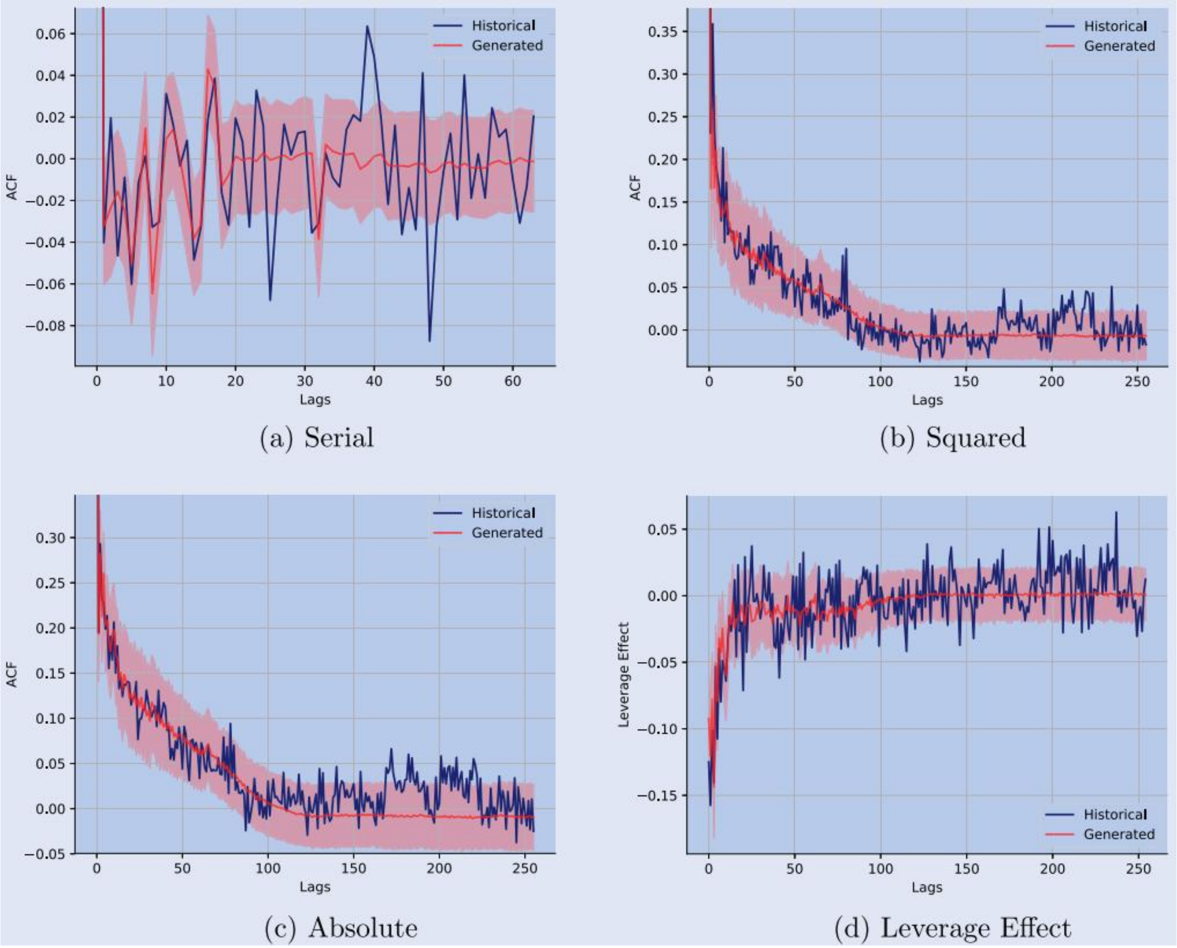


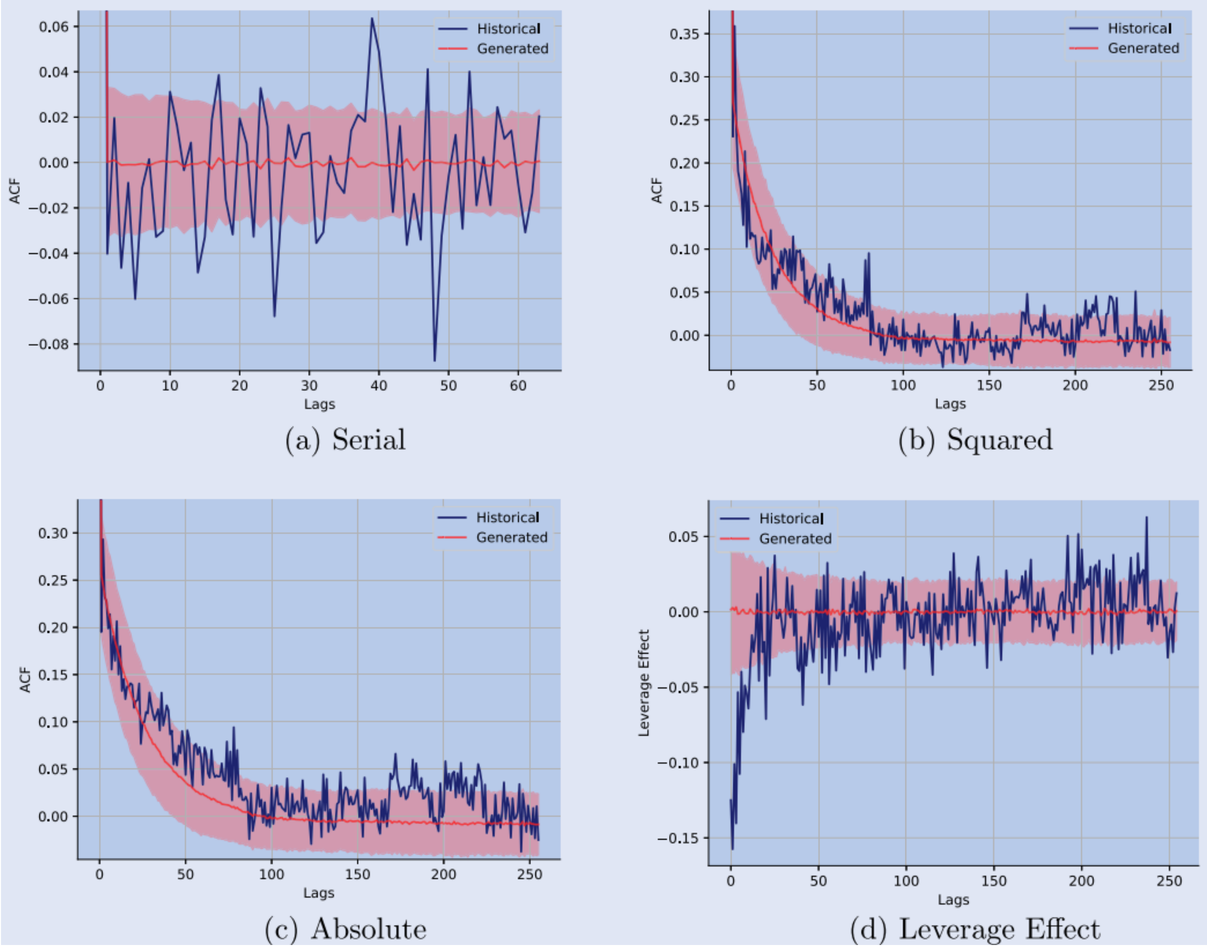
Figure A2. Fifty generated driftless log paths.

QuantGANs VS GARCH(1,1) (old model)

<u>QuantGANs</u>



<u>GARCH(1,1) (old model)</u>



Reference

Wiese, Magnus, et al. "Quant GANs: deep generation of financial time series." *Quantitative Finance* 20.9 (2020): 1419-1440.

Thank you for listening