Faculty of Economic Sciences

November 8, 2024

# Diffusion Models for Time Series Generation

Tatiana Sorokina

tssorokina@edu.hse.ru

# Disclaimer

This work was prepared as a part of a research internship at Global Markets DataAI Lab in BNP Paribas CIB, Frankfurt-am-Main, September 2024.

# Outline

1. Introduction to generative models and motivation
2. Literature Review
3. Two main diffusion model settings and equations
4. Application to Time Series (TS) generation

# Why Diffusion models for Economic Time Series (TS)?

**Goal of a generative model:**
Given observed samples $x \sim p(x)$, the goal of a model is to learn $p(x)$. Then, new samples can be sampled from $p(x)$. Under some formulations, the model could also evaluate the likelihood of observed or sampled.

# Why Diffusion Models (DM) for Economic Time Series?

- **Generative Power:**
  - Diffusion models are capable of capturing complex patterns and dependencies, which makes them suitable for structured time series data.
  - These models, generally, show better results in comparison to other generative models (GAN, VAE, Flows, etc.)
- **Advantages in Financial and Economic Data:**
  - Financial TS are inherently **dynamic, high-dimensional, and non-stationary**
  - Diffusion models are designed to handle such complexities
  - Financial data comprises vast, high-frequency datasets (i.e. stock prices, trade volumes, and Limit Order Book (LOB) data)
- **Use Cases:**
  - Financial and banking institutions can use synthetic data for applications in **risk management**, **stress testing**, and **market simulation**.
  - Research on generative models outside of image and text applications is mostly focused on financial and bio-engineering settings.

# Literature (1)

## General frameworks

**Original papers:**

- **Ho et al., 2020** set the original DDPM setting ($T$ discrete diffusion timesteps, linear noisification schedule). It is a general setup used in most applications
  - J. Song et al., 2022 propose DDIM, an important extension of DDPM that allows to generate new samples in less backward process steps
- **Y. Song et al., 2021**: Score-Based approach, continuous diffusion timesteps. For an overview of SDE diffusion model formulation and its link to DDPM refer to Tang and Zhao, 2024.

For an overview of the Diffusion generative model setting and their link to Variational Auto-Encoder models refer to Luo, 2022.

# Literature (2)
## Applications to TS generation

- **Forecasting** for multivariate TS, spatio-temporal graphs
  - TimeGrad (Rasul et al., 2021), *ScoreGrad* (Yan et al., 2021), *$D^3$VAE* (Y. Li et al., 2023), *DSPD*/*CSPD* (Biloš et al., 2023), *DiffSTG* (Wen et al., 2024), *GCRDD* (R. Li et al., 2023)
- **Imputation** of missing values for multivariate TS, spatio-temporal graphs
  - *CSDI* (Tashiro et al., 2021), *DSPD*/*CSPD* (Biloš et al., 2023), *SSSD* (Alcaraz & Strodthoff, 2023), *PriSTI* (Liu et al., 2023)
- **Generation** of multivariate TS
  - *TSGM* (Lim et al., 2023)

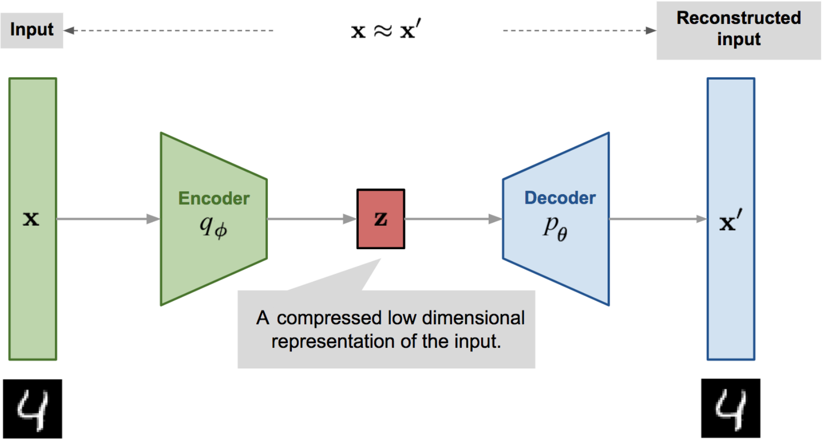
# Literature (3)

## Other types of generative models in TS generation

- **Generative Adversarial Networks, GANs** (Goodfellow et al., 2014): two neural networks, *Generator* and *Discriminator*, are trained with min-max loss function. *Discriminator* aims to correctly classify generated and original samples while *Generator* tries to deceive *Discriminator*
  - Examples of applications to TS generation: Yoon et al., 2019
- **Variational Auto Encoders, VAEs** (Kingma & Welling, 2013): learn to *encode* data into a latent space making it resemble a certain prior (typically Gaussian) through KL divergence, and *decode* it by minimising the difference between decoded and original sample
  - Examples of applications to TS generation: Desai et al., 2021
- Other approaches include *normalising flows*, *energy-based models* (*Score-Based* models are related, but learn the *score* of the energy function and not the function itself)
- Sequential and structured data, such as TS, can be generated with autoregressive models

# Variational Auto-Encoder

## And its link to DDPM




Composition of a basic VAE model

# Basic DDPM (1)

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models.

**Core Concept**:

Diffusion models transform a sample $x \sim p_{data}(x)$ via a *deterministic* time-dependent stochastic **forward process** $q(x_{t-1}|x_t)$ across discrete time steps $t \in \{1, T\}$, and learn a **backward** transformation **process** $p_\theta(x_t|x_{t+1})$, that would allow us to sample from a latent space

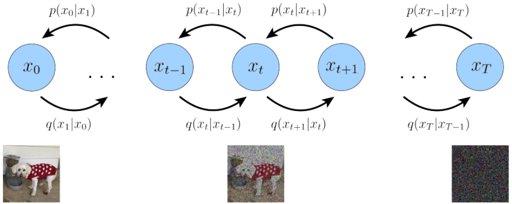



*Figure source: Luo, 2022*

# Basic DDPM (2)

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models.

**Forward Process** s.t. $x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t$:

- Adds noise $\epsilon_t \sim \mathcal{N}(0, \sigma_t^2)$ over discrete diffusion timesteps $t \in [0, T]$ to gradually diffuse data $x_0$ into noise, controlled by schedule $\{(1-\alpha_0), \dots, (1-\alpha_T)\}$
- Defined as:

$$q(x_{1:T}|x_0) = \prod_{t=0}^{T} \mathcal{N}\left(x_t; \sqrt{\alpha_t} \cdot x_{t-1}, (1-\alpha_t) \cdot \mathbf{I}\right)$$

# Basic DDPM (3)

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models.

- Define $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. Then we can apply a **reparametrisation trick**:

$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t} x_0, \sqrt{1-\bar{\alpha}_t} \cdot \mathbf{I}\right)$$

- Conditioned on both, $x_t$ and $x_0$, the **forward process** can be "reversed"[1]:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \underbrace{\sqrt{\alpha_t}\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} x_t + \sqrt{\bar{\alpha}_{t-1}} \cdot \frac{1-\alpha_t}{1-\bar{\alpha}_t} x_0}_{\mu_q(x_t, x_0)}, \underbrace{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}_{\Sigma_q(t)} \cdot \mathbf{I}\right)$$

[1]Refer to Luo, C. (2022). Understanding diffusion models: A unified perspective.

## Basic DDPM (4)

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models.

**Backward Process**:

- With $T$ sufficiently big, **backward process** should be equivalent to the **forward process**: $p_\theta(x_{t-1}|x_t) \approx q(x_{t-1}|x_t, x_0)$
- We set the **backward process** to be Gaussian with the same variance schedule as **forward process** $p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(\mu_\theta(x_t, t), \Sigma_q(t)\right)$

**Likelihood maximisation:**

- Ho et al., 2020 argue that the sample likelihood maximisation is equivalent to minimisation the following term:

$$\underset{\theta}{\text{argmax}} \log p(\mathbf{x}) \sim \underset{\theta}{\text{argmin}} \mathbb{E}_{t\sim U[2,T]}\left\{\mathbb{E}_{q(x_t|x_0)}\left\{\mathcal{D}_{KL}\left(q(x_{t-1}|x_t, x_0)|p_\theta(x_{t-1}|x_t)\right)\right\}\right\}$$

- At the same time:

$$\underset{\theta}{\text{argmin}}\ \mathcal{D}_{KL}\left(q(x_{t-1}|x_t, x_0)|p_\theta(x_{t-1}|x_t)\right) = \underset{\theta}{\text{argmin}} \frac{1}{2\Sigma_q(t)}\left[||\mu_\theta(x_t, t) - \mu_q(x_t, x_0)||_2^2\right]$$

# Basic DDPM (5)

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models.

**Likelihood maximisation:**

- Ho et al., 2020 argue also that:

$$\underset{\theta}{\text{argmin}}\, D_{KL} = \underset{\theta}{\text{argmin}}\, \frac{1}{2\Sigma_q(t)} \cdot \frac{(1-\alpha_t)^2}{(a-\bar{\alpha}_t)\alpha_t} \left[ ||\epsilon - \hat{\epsilon}_\theta(x_t, t)||_2^2 \right]$$

**Stable Diffusion** model at each training step:

1. For a subsample $x_0$ of $\mathbf{x}$, generate noise $\epsilon \sim \mathcal{N}(0, 1)$ and draw a noisification timestep $t \in [2, T]$
2. Get $x_t \sim q(x_t | x_0)$
3. The neural network with parameters $\theta$ predicts the noise $\hat{\epsilon}_\theta(x_t, t)$
4. The loss is computed as $L^2$ norm between predicted and actual noise; parameters $\theta$ are updated according to standard NN optimisation techniques

# Stochastic Differential Equation (SDE) Formulation

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations.

**Forward process**:

- Here, the **forward pass** is modelled with continuous diffusion timesteps $t \in [0, T]$. Let $W$ and $\widetilde{W}$ be standard Weiner processes.
- The **forward process** is defined as: $\mathrm{d}\mathbf{x} = f(x, t)\mathrm{d}t + g(t)\mathrm{d}W$

**Bakward process**:

- Time-reverse SDE[2]: $\mathrm{d}\mathbf{x} = \left[f(x, t) - g(t)^2 \nabla_{\mathbf{x}} \log q_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\widetilde{W}$

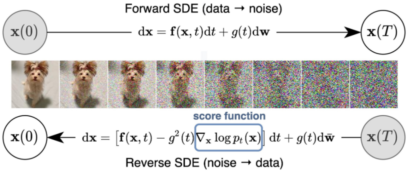



*Figure source: Tang and Zhao, 2024*

[2]Refer to Tang and Zhao, 2024 for in-depth derivation

# Stochastic Differential Equation (SDE) Formulation

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations.

**Sampling**:

- Knowing the reverse equation, we can sample from the probability flow ODE that has the same distribution as the reverse SDE:

$$\mathrm{d}\mathbf{x} = \left[ f(x, t) - \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}} \log q_t(\mathbf{x}) \right] \mathrm{d}t$$

- The only function that is unknown is the **score function** $\nabla_{\mathbf{x} \log q_t(\mathbf{x})}$, which can be learned by the neural network with the objective function:

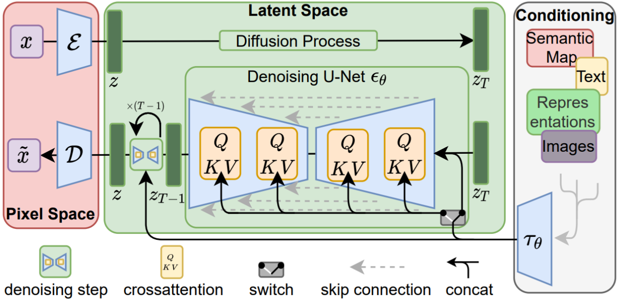
$$\mathbb{E}_{t, x_0, x_t} \left[ \delta(t) || s_\theta(x_t, t) - \nabla_{x_t} \log q_t(x_t | x_0) || \right]$$

- The **forward process** is modelled as a process with known distribution (usually, Variance Preserving processes) to be able to estimate the score function

# Sampling in a Latent Space

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models.

- Sampling high-dimensional data from pure noise of the same dimension is long, training of such model is costly, *fidelity* of a produced image is low
- Rombach et al., 2022 proposed to first reduce the dimension of the samples with a well-trained VAE model, and then generate the samples with Diffusion-style models
- This is how most Stable Diffusion models operate now (link)



# Basic Approaches for TS Diffusion

**Complexity of Time Series Generation**:

- Unlike images or tabular data, time series are not *i.i.d.*, complicating generation

**TimeGrad** and **ScoreGrad** approaches:

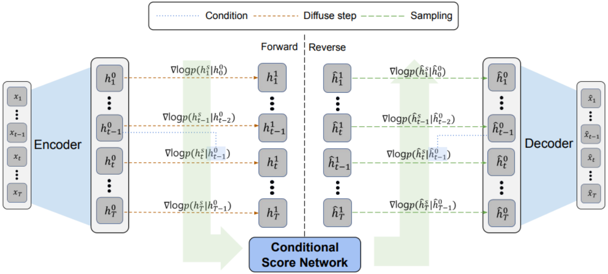
- Rasul et al., 2021 proposed to encode the information of historical time series with an autoregressive model (Recursive NN module such as Long-Short-Term memory, LSTM, or Gated Recurrent Unit, GRU): $h^s = \text{RNN}(x_0^s, c^s, h^{s-1})$, where the upper indes $s \in 1, S$ defines actual TS time, $h^s$ is a latent-space representation of the series $(x^0, .., x^{s-1})$, and $c^s$ is an exogenous variable impacting $x^s$ at time $s$.
- Then, the model aims to maximise: $\prod_{s=1}^{S} p_\theta(x_0^s | h^{s-1})$
- Consitional Diffusion generation applies for each step of the TS. THe NN parametrises $p_\theta(x_0^s | h^{s-1})$ in a manner similar to SDE approach, but in discrete time.
- Yan et al., 2021 extended this approach to continuous diffusion time and

# TS Generation with Score-Based Diffusion Models

Lim, H., Kim, M., Park, S., & Park, N. (2023, January 20). Regular time-series generation using SGM.

- Lim et al., 2023 propose to sample in a latent space is the same manner as with Latent Conditional Diffusion Models
- The sampling function for a fixed TS time $s$ becomes:

$$dh_t^s = [f(h_t^s, t) - g(t)\nabla_{h_t^s} \log p_\theta(h_t^s | h_{t-1}^s)]dt + g(t)d\widetilde{W}$$




Composition of the TSGM model. *Source: Lim et al., 2023*

## Generated Data Quality Evaluation Metrics

Based on Sikder, M. F., Ramachandranpillai, R., & Heintz, F. (2024). TransFusion: Generating long, high fidelity time series using diffusion models with transformers.

- **PCA and t-SNE**: Visualize synthetic vs. original data in 2D, assessing how well synthetic data aligns with real data
- **LDS (Long-Sequence Discriminative Score)**: Fidelity metric; transformer classifier accuracy difference ($|0.5 - \mathit{accuracy}|$) shows similarity
- **LPS (Long-Sequence Predictive Score)**: Predictive metric; transformer trained on synthetic data, evaluated on real data via Mean Absolute Error (MAE)
- **JSD (Jensen-Shannon Divergence)**: Measures distribution similarity; lower score indicates closer synthetic-real data alignment
- **$\alpha$-Precision and $\beta$-Recall**: Fidelity and diversity metrics; precision shows synthetic coverage of real data, recall reflects diversity match
- **Coverage**: Checks for mode collapse; measures fraction of original samples with synthetic neighbours

Faculty of Economic Sciences

November 8, 2024

# Diffusion Models for Time Series Generation

Tatiana Sorokina

tssorokina@edu.hse.ru

# Referernces (1)


- Alcaraz, J. M. L., & Strodthoff, N. (2023). Diffusion-based time series imputation and forecasting with structured state space models.
- Biloš, M., Rasul, K., Schneider, A., Nevmyvaka, Y., & Günnemann, S. (2023). Modeling temporal data as continuous functions with stochastic process diffusion.
- Desai, A., Freeman, C., Wang, Z., & Beaver, I. (2021, December 7). TimeVAE: A variational auto-encoder for multivariate time series generation.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014, June 10). Generative adversarial networks.
- Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes.
- Li, R., Li, X., Gao, S., Choy, s., & Gao, J. (2023). Graph convolution recurrent denoising diffusion model for multivariate probabilistic temporal forecasting.
- Li, Y., Lu, X., Wang, Y., & Dou, D. (2023). Generative time series forecasting with diffusion, denoise, and disentanglement.

# Referernces (2)


- Lim, H., Kim, M., Park, S., & Park, N. (2023, January 20). Regular time-series generation using SGM.
- Liu, M., Huang, H., Feng, H., Sun, L., Du, B., & Fu, Y. (2023). PriSTI: A conditional diffusion framework for spatiotemporal imputation.
- Luo, C. (2022). Understanding diffusion models: A unified perspective.
- Rasul, K., Seward, C., Schuster, I., & Vollgraf, R. (2021, February 2). Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models.
- Sikder, M. F., Ramachandranpillai, R., & Heintz, F. (2024). TransFusion: Generating long, high fidelity time series using diffusion models with transformers.
- Song, J., Meng, C., & Ermon, S. (2022). Denoising diffusion implicit models.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations.

# Referernces (3)


- Tang, W., & Zhao, H. (2024). Score-based diffusion models via stochastic differential equations – a technical tutorial.
- Tashiro, Y., Song, J., Song, Y., & Ermon, S. (2021). CSDI: Conditional score-based diffusion models for probabilistic time series imputation.
- Wen, H., Lin, Y., Xia, Y., Wan, H., Wen, Q., Zimmermann, R., & Liang, Y. (2024). DiffSTG: Probabilistic spatio-temporal graph forecasting with denoising diffusion models.
- Yan, T., Zhang, H., Zhou, T., Zhan, Y., & Xia, Y. (2021, June 18). ScoreGrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models.
- Yoon, J., Jarrett, D., & van der Schaar, M. (2019).Time-series generative adversarial networks. *Advances in Neural Information Processing Systems, 32.*