Redes Neurais (Deep Learning) : modelos generativos

Ref.: MIT – Introduction to Deep Learning

A estrutura matemática das redes neurais

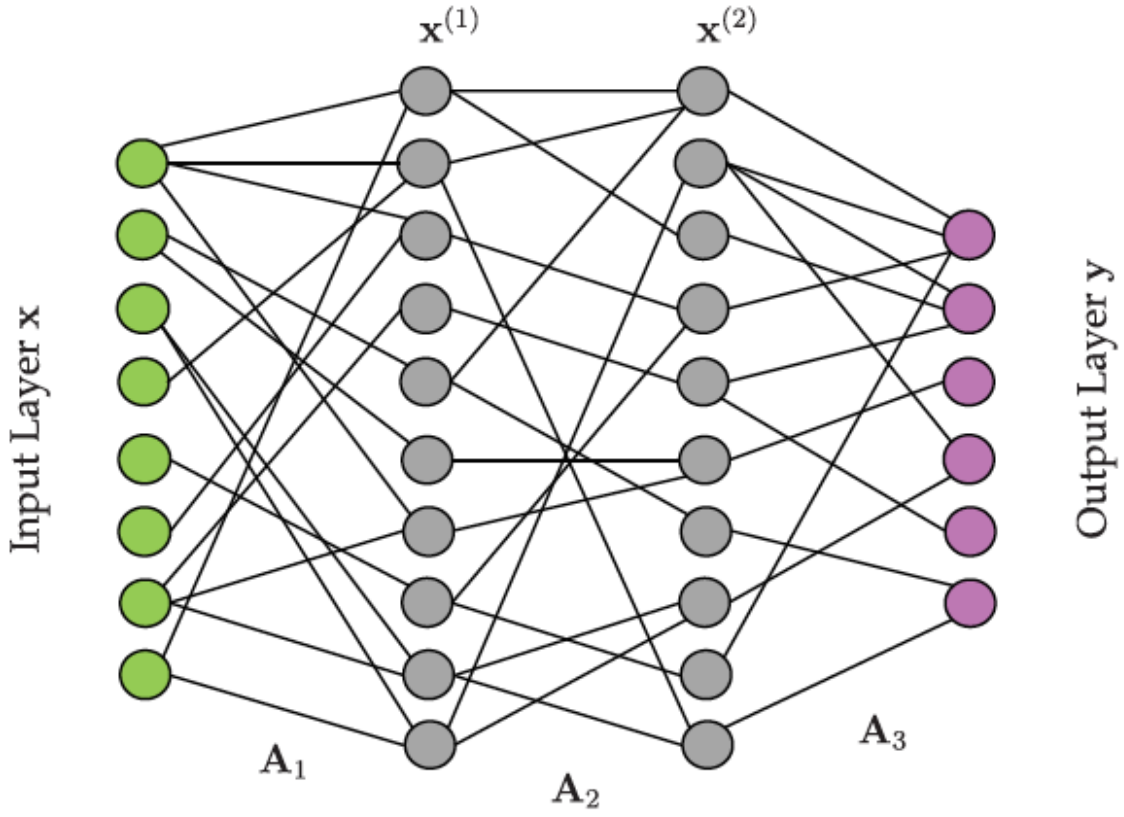


Figure 6.1 Illustration of a neural net architecture mapping an input layer $\mathbf{x}$ to an output layer $\mathbf{y}$. The middle (hidden) layers are denoted $\mathbf{x}^{(j)}$ where j determines their sequential ordering. The matrices $\mathbf{A}_j$ contain the coefficients that map each variable from one layer to the next. Although the dimensionality of the input layer $\mathbf{x} \in \mathbb{R}^n$ is known, there is great flexibility in choosing the dimension of the inner layers as well as how to structure the output layer. The number of layers and how to map between layers is also selected by the user. This flexible architecture gives great freedom in building a good classifier.

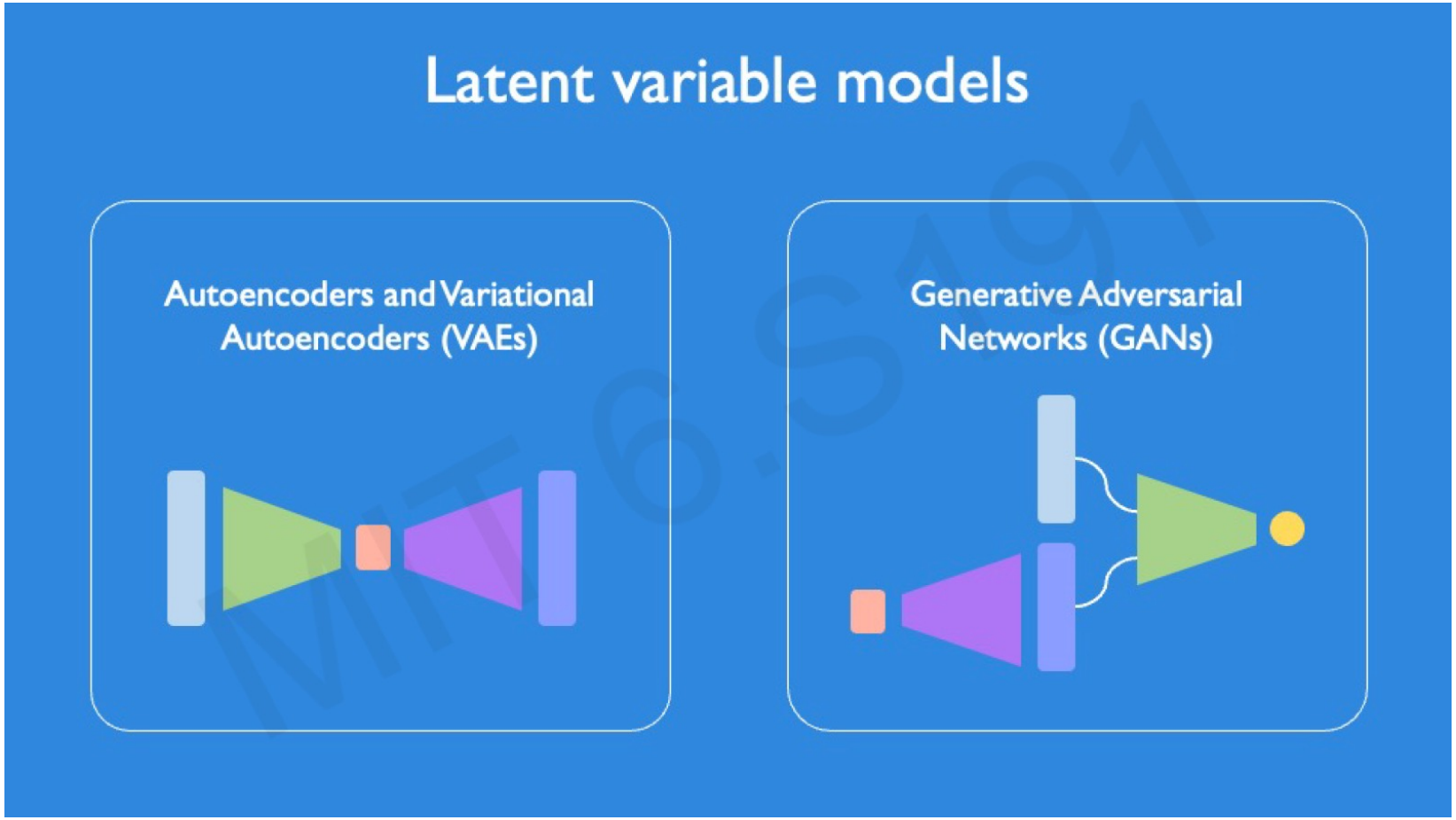
$$\mathbf{y} = f(\mathbf{x}; \mathbf{w})$$

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

Modelos Generativos – características gerais

- Aprendizado não supervisionado
- Aprender distribuição de probabilidade intrínseca ao conjunto de dados
- Geração de novos dados
- Muita coisa associada a aprendizado probabilístico

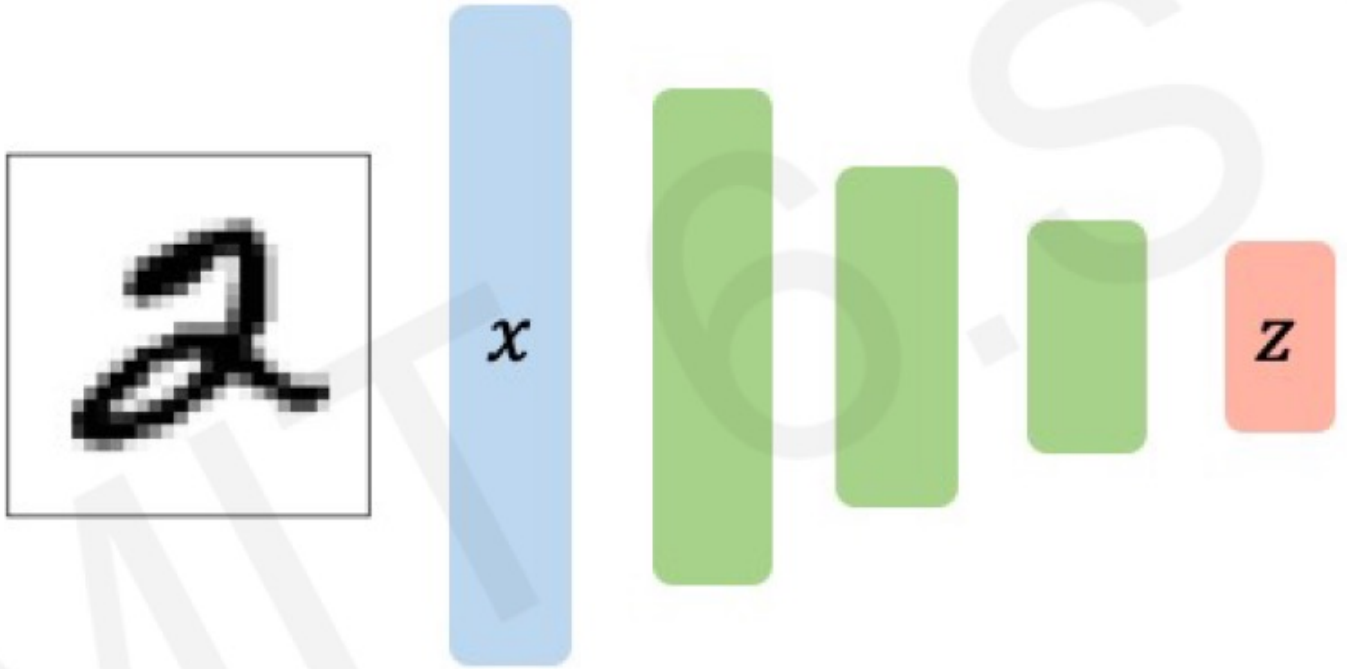
Ideia central... Estrutura intrínseca (não conhecida – hidden) envolvendo os dados



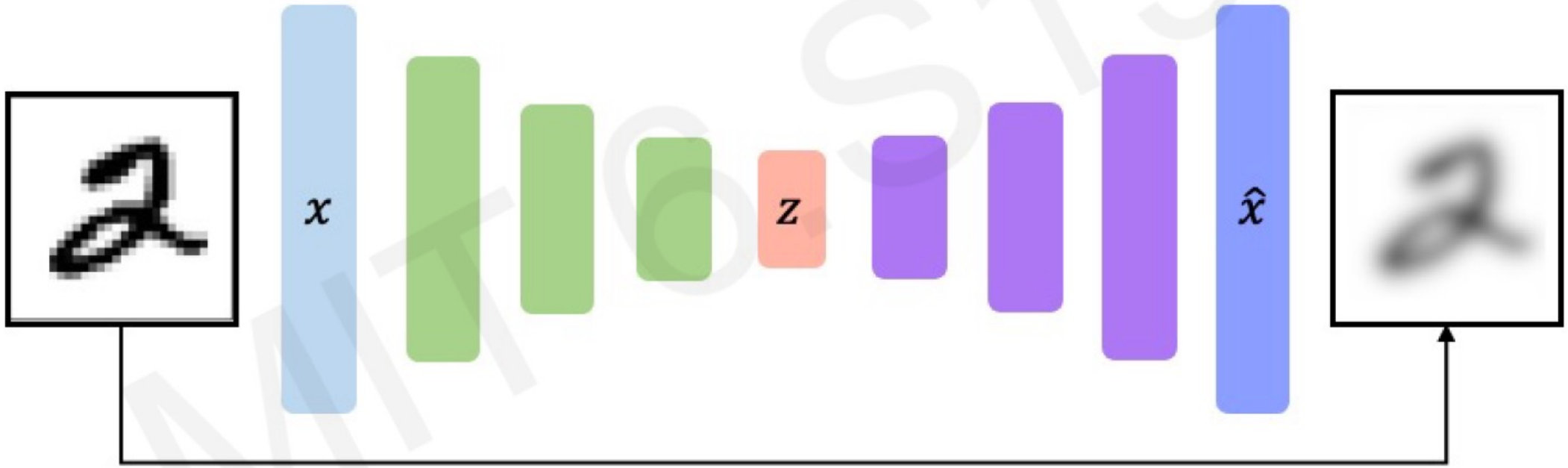
Como “descobrir” quem são as variáveis latentes?

- Modelos reduzidos : POD (aproximação linear)

- Autoencoders



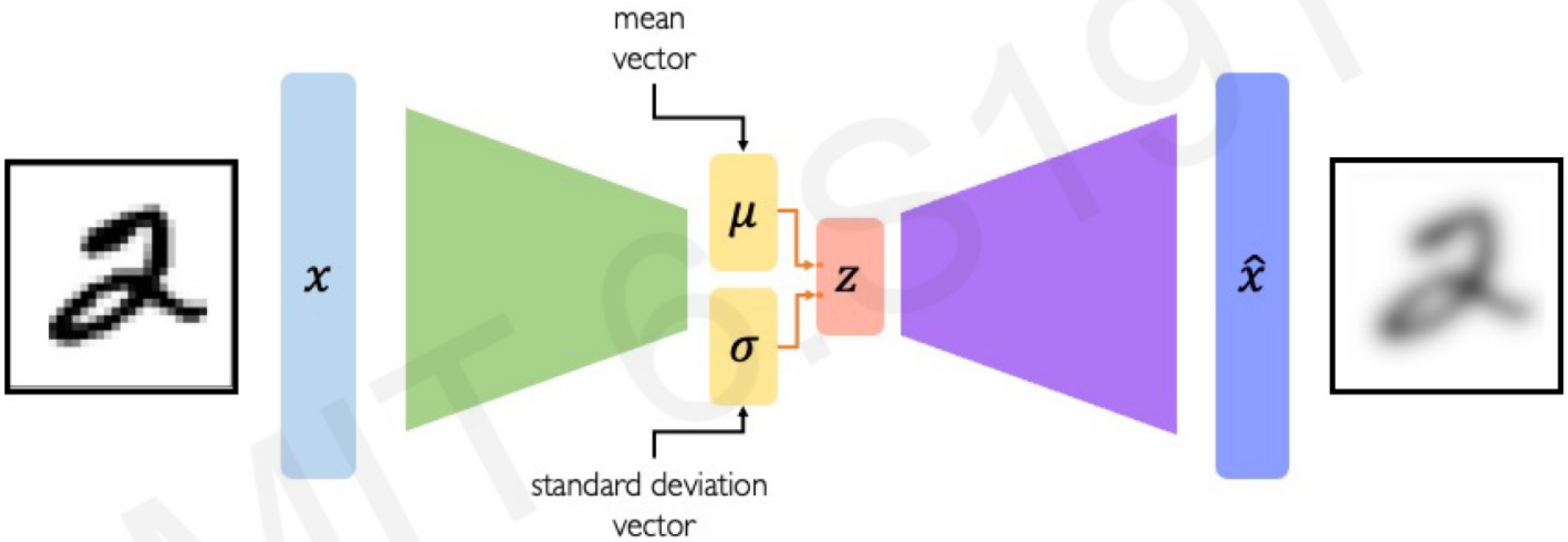
Como treinar um autoencoder?



$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2$$

Loss function doesn't use any labels!!

Variational Autoencoders



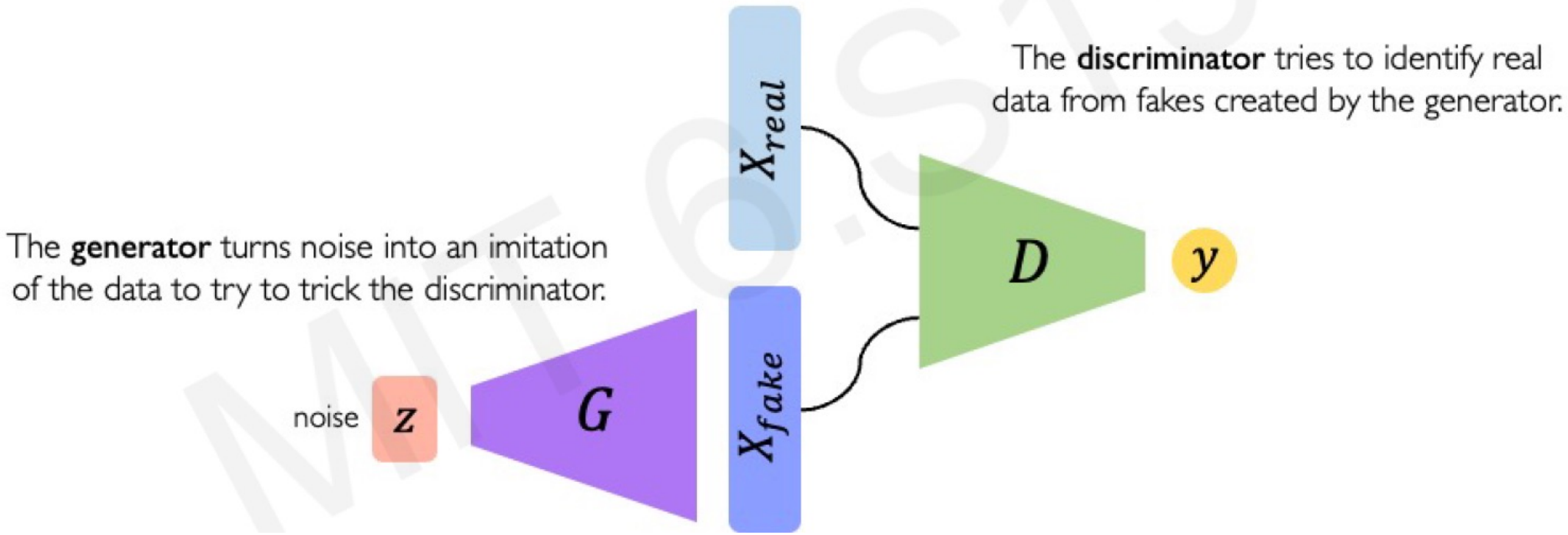
Variational autoencoders are a probabilistic twist on autoencoders!

Sample from the mean and standard deviation to compute latent sample

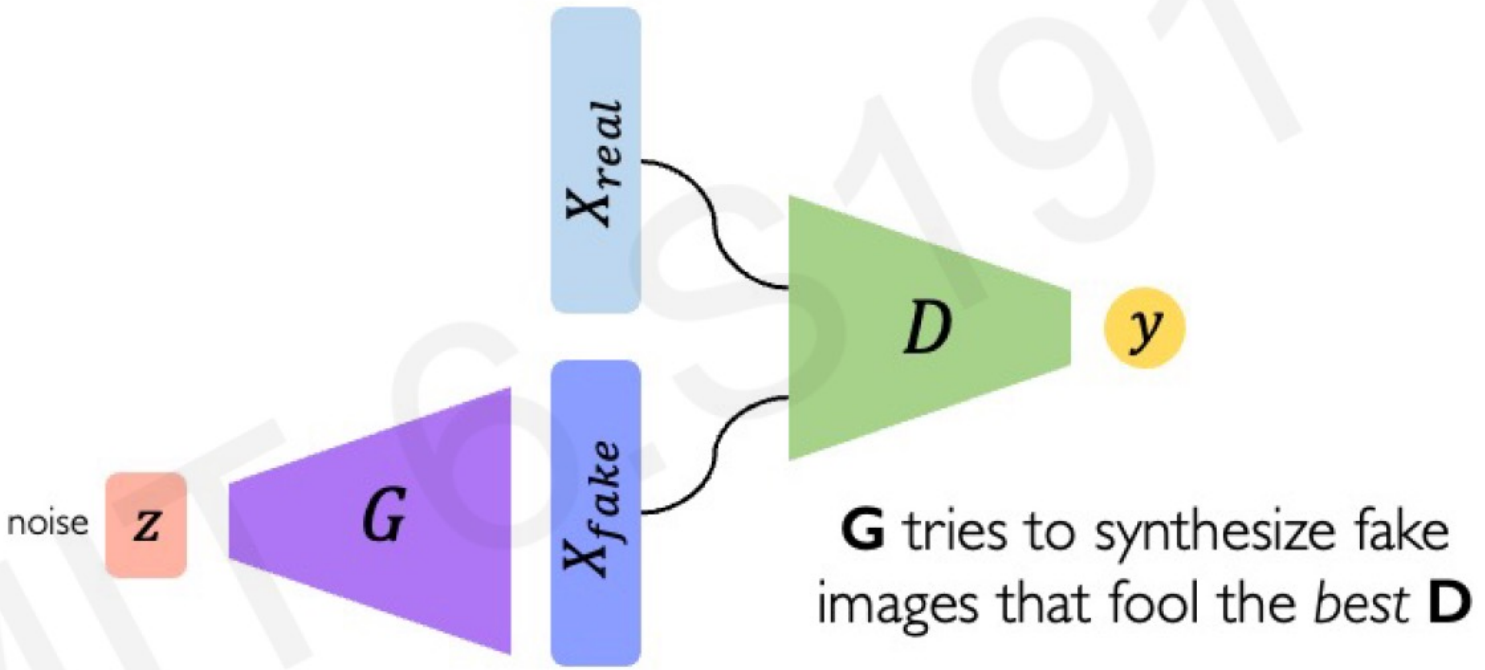
Encoder computes: $q_\phi(\mathbf{z}|x)$ Decoder computes: $p_\theta(\mathbf{x}|z)$

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are a way to make a generative model by having two neural networks compete with each other.

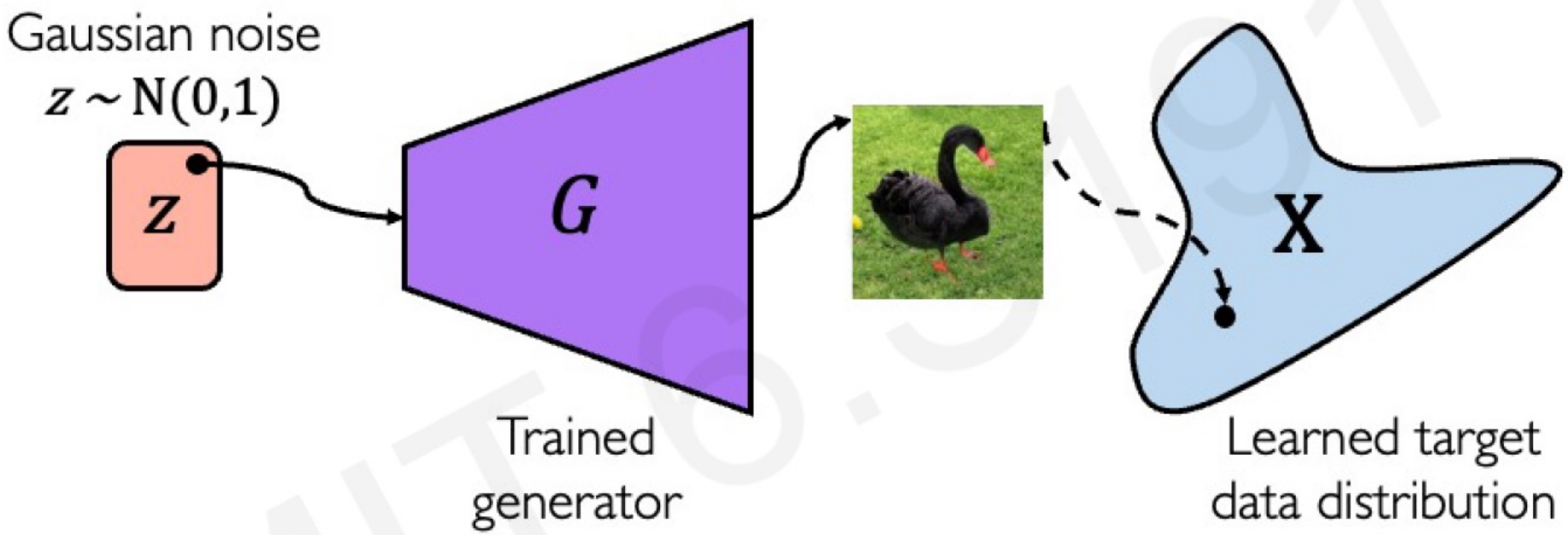


Treinando GANS

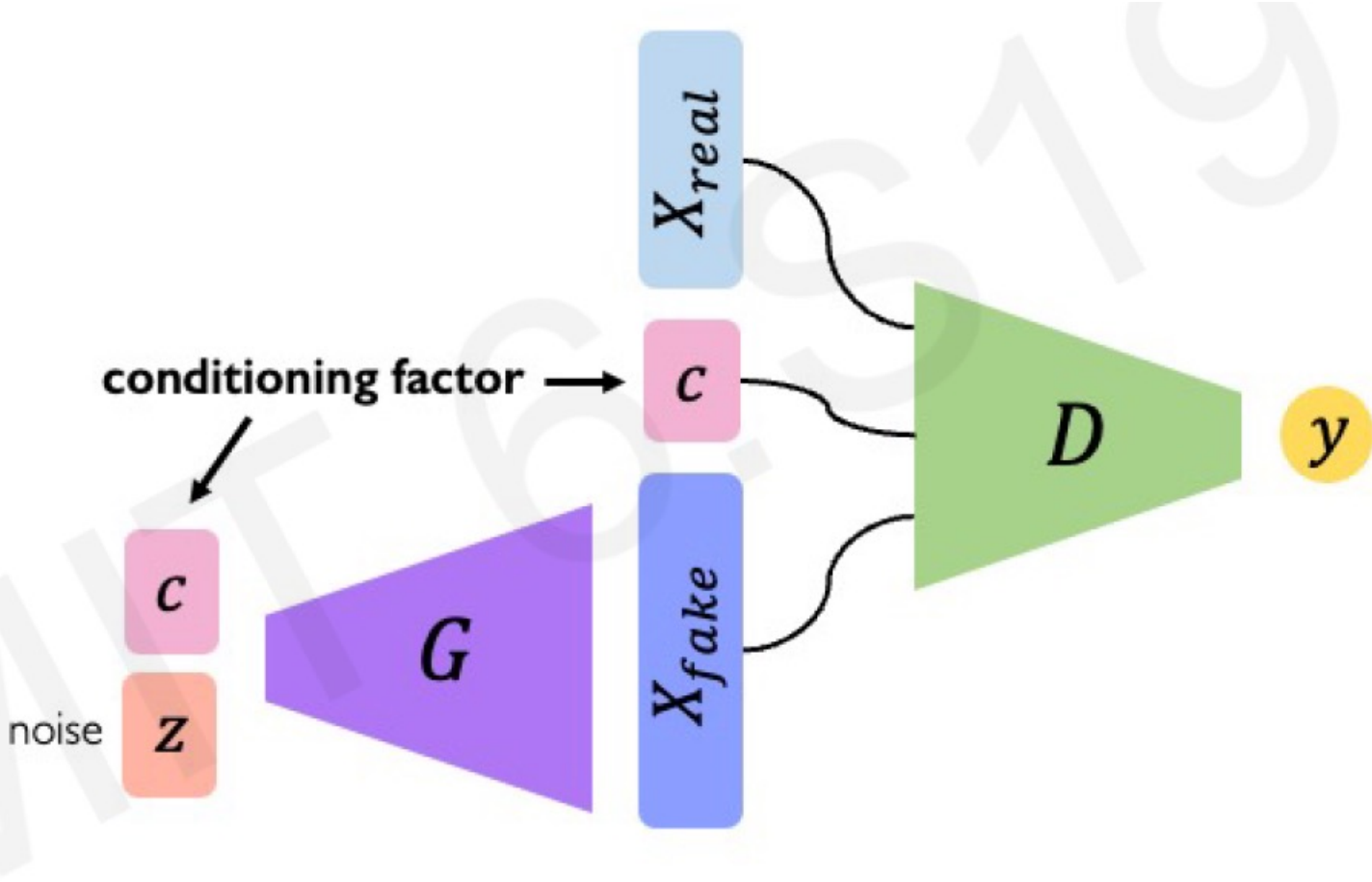


$$\arg\min_G \max_D \mathbb{E}_{\mathbf{z},\mathbf{x}}[\ \log D(G(\mathbf{z})) + \log(1 - D(\mathbf{x}))\ ]$$

Gerando novas amostras com GANs



Conditional GANs



A Physics-Based Machine Learning Model for Fatigue in Wind Turbine Devices

XIX International Symposium on Dynamic Problems of Mechanics

Tiago B. M. M. Puntel[1] and Fernando A. Rochinha[1]

[1]Mechanical Engineering, Universidade Federal do Rio de Janeiro, Brazil

26 Feb - 03 Mar 2023

Wind Field, Turbine Dynamic Response and Fatigue Assessment

- The wind flow has a complex structure that evolves in time and space, containing various frequencies and magnitudes.
- Spatial-time flow characterization is unavailable since turbine sensors measure the wind speed at specific points, and these systems provide 10 minutes summary statistics.
- Simplified wind flow models, such as the one used in TurbSim[3], deliver a cost-benefit solution for representing the flow. It generates wind speed in a specific time interval (10 minutes) at any point inside the turbine plane.
  - It utilizes summary statistic inputs provided by wind turbine sensors.

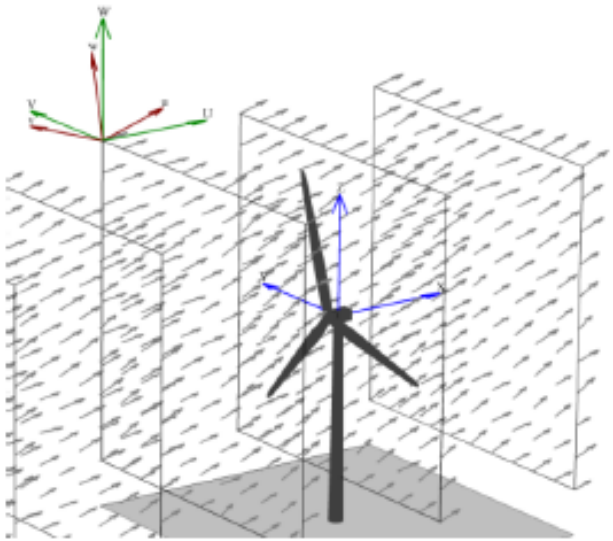


Figure: Example of TurbSim flow.[3]

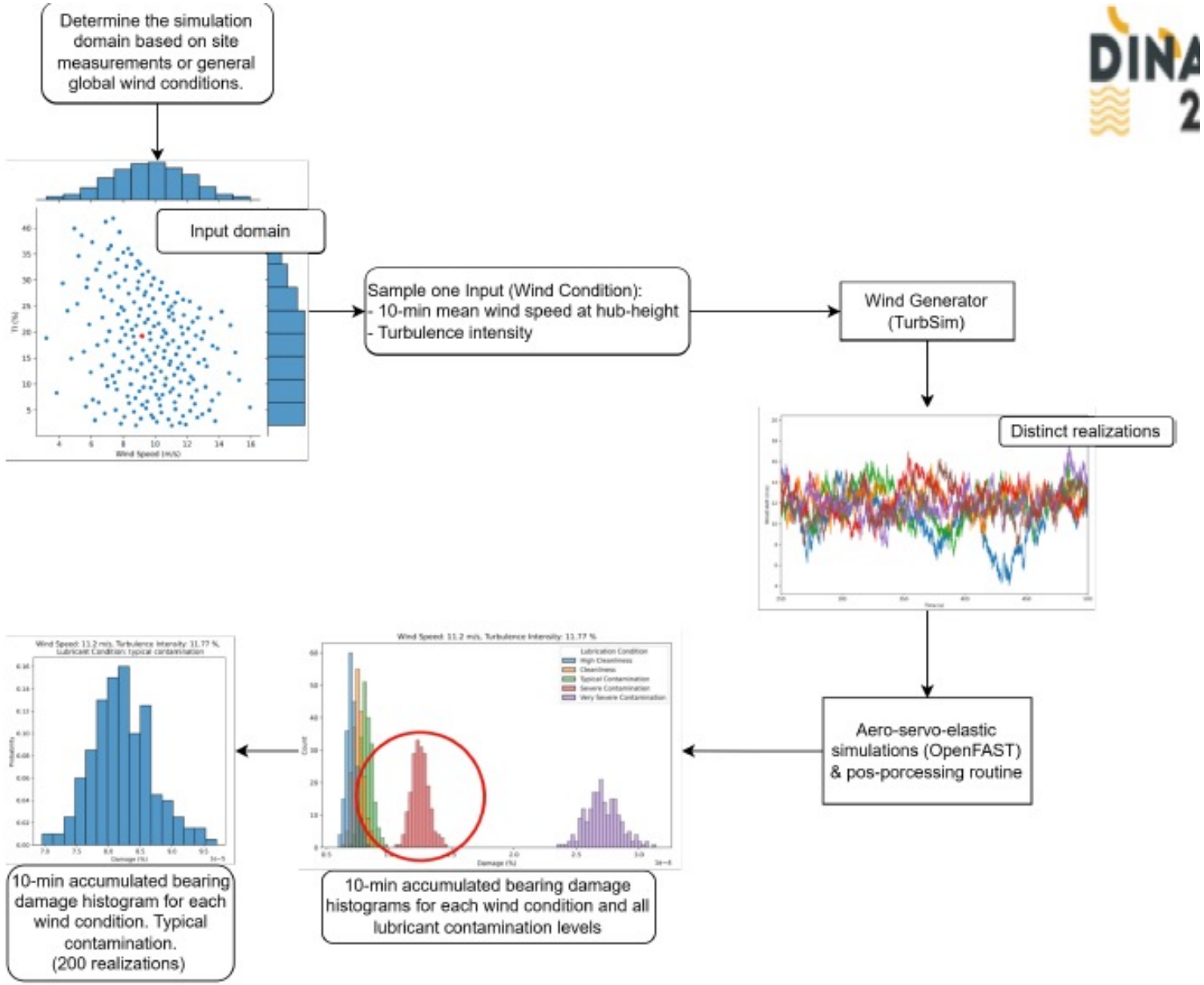


Figure: Stochastic simulation flowchart

Machine Learning Surrogate Model

- The flow model and the aero-servo-elastic simulations are time consuming for a real-time application or a multiple scenario assessment. A Machine Learning algorithm can reduce the computation cost.

- The simulation outcomes provide an empirical and unknown joint distribution $q(x, y)$, which converges to the true one $p(x, y)$ as long as the data set increases. x is the input vector containing the wind speed, turbulence intensity and lubricant condition (ws, ti, lc). y is the accumulated bearing damage in 10 minutes.

- As there are uncertainties in the wind speed time series since we generate them considering only summary statistics as inputs such as the mean wind speed and the turbulence intensity. Many flows with different random seeds are generated for each input.

- Therefore, the ML model must learn the stochastic relation between input and output, i.e., $q(y|x = X_i)$, where X_i is the training points and estimate damage values out of the the training data set.

Conditional Deep Surrogate Model[5]

- The chosen ML model is a type of Generative Adversarial Network.

Discriminator Loss function

$$\mathcal{L}_{\mathcal{D}}(\psi) = \mathbb{E}_{x\sim q(x),z\sim p(z)}\left[\log(D_\psi(x,\hat{y}))\right] + \mathbb{E}_{(x,y)\sim q(x,y)}\left[\log(1 - D_\psi(x,y)\right]$$

Generator-Encoder Loss function

$$\mathcal{L}_{\mathcal{G}-\mathcal{E}}(\theta,\phi) = \mathbb{E}_{x\sim q(x),z\sim p(z)}[D_\phi(x,\hat{y}) + \lambda(||z-\hat{z}||_2)]$$

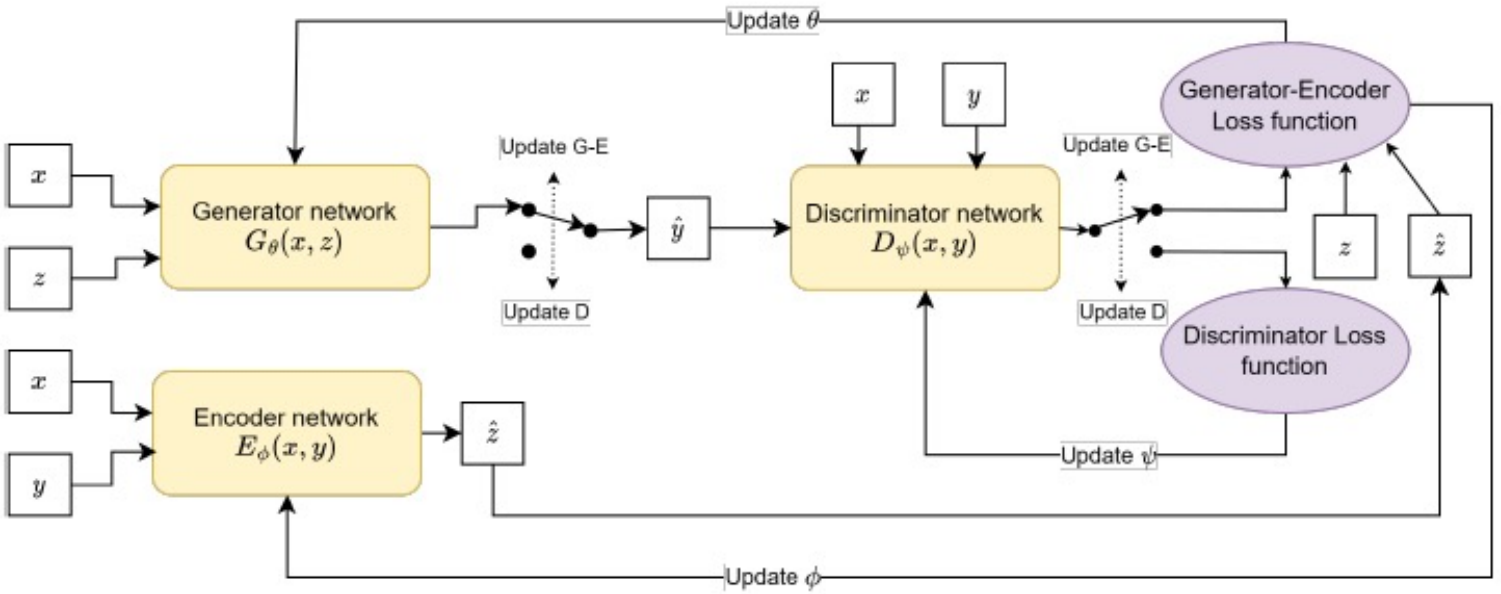
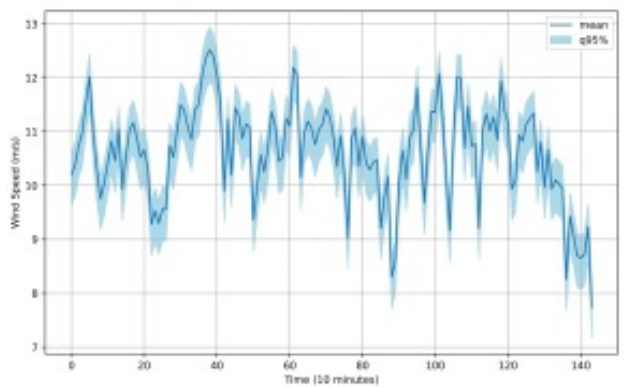


Figure: Generative model training process.

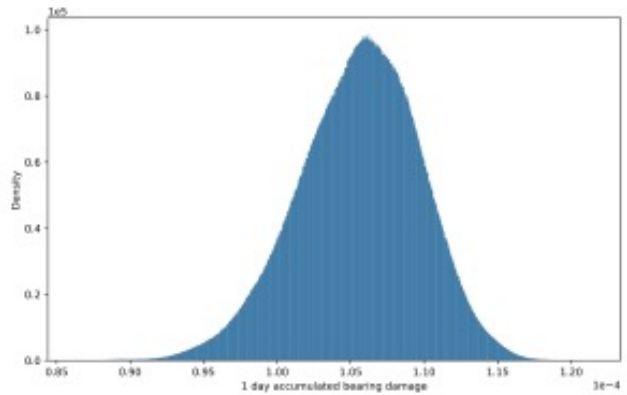
[5] Yang, Y., Perdikaris, P., Conditional deep surrogate models for stochastic, high-dimensional, and multi-fidelity systems. Comput Mech 64, 417–434 (2019). https://doi.org/10.1007/s00466-019-01718-y.

Application 2

- In the second application, we implement the surrogate model to assess the daily accumulated bearing fatigue damage.
- We use standard SCADA data considering measurement uncertainties.



(a) One day wind speed time series (SCADA data).



(b) One day accumulated bearing damage, considering uncertainties in SCADA measurements. Processing time: (17 minutes).

Construction of predictive coarse-graining machine learning models with Gaussian Processes and Conditional Probability learning for diesel/biodiesel surrogate fuels

May 30, 2023, Rio de Janeiro

Background

- High-fidelity computer models are prone to accelerate the process of designing and deploying **complex engineered systems** like engines with improved performance and less emission
- Such models result from the combination of **basic scientific principles** (e.g. balance of linear momentum) with **closure equations**
- Closure equations connect state variables to thermophysical properties (e.g., viscosity and surface tension).
    - Due to its own nature, such closures represent, frequently, the **weak link chain in the modelling**

Motivation

- Thermophysical properties of practical fuels are important but **difficult to measure/predict**, especially when **complex surrogate fuels** and **extreme conditions** (such as supercritical) are concerned
- Molecular Dynamics is a common approach to characterize the physical properties of practical fuels. Also, **coarse-graining models** build simplified representations that allow keeping the main chemical/physical characteristics and performing affordable and meaningful simulations
- Machine learning has great potential to **discover from data** the relation between inputs and outputs in a thermodynamic system
- Also, machine learning techniques have been widely used to **model closure relations** in many practical problems [1] [2]. Furthermore, these approaches can be adapted to accommodate **multi-fidelity** data in the learning process

Aims

Construct **cheap-to-compute** machine learning models to act as an **closure equations** for predicting the physical properties of diesel/biodiesel surrogate fuels

Machine Learning Models

- Surrogate fuels: $\mathbf{C_nH_{2n+2}}$
- **Dataset:**1200 points of $\rho(n, p, T)$, $T \in [320, 900K]$ varying by 20 K and $p = [3, 4, 6, 8, 10, 20, 100, 150]MPa$
  - **Octane** (C8H18), **Nonane** (C9H20), **Decane** (C10H22), **Dodecane** (C12H24) and **Hexadecane** (C16H34)
- 80% of the data to training: $(D_1 = 10\%,\ D_2 = 50\%,\ D_3 = 80\%)$
- 20% of the data to testing
- **Input**: n, p, T and **Output**: $\rho(n, p, T)$
  - n is a integer, but it is assumed to be a continuously variable since a Gaussian process is the joint distribution of the random variables over a continuous domain

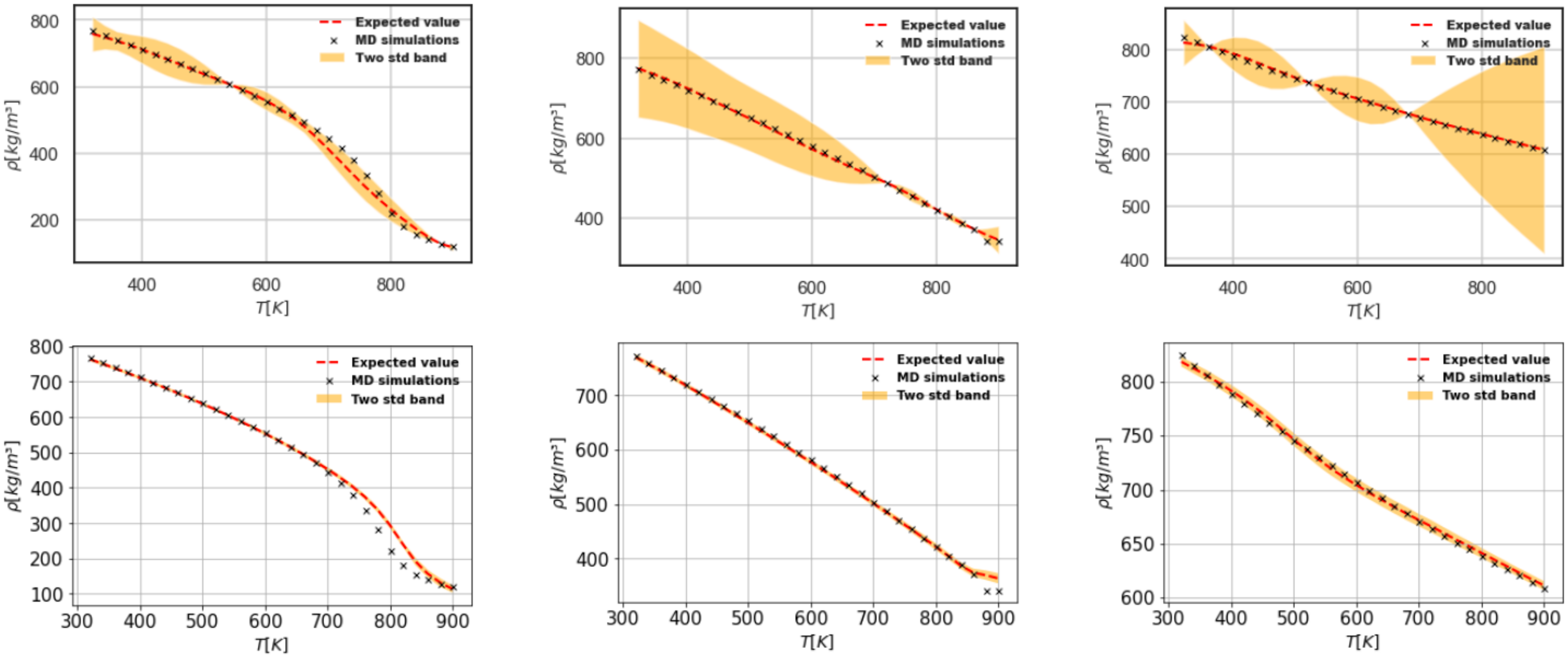
Gaussian Process

$$\textbf{Matern32:}\quad k(r) = \sigma^2(1+\sqrt{3}r)exp\left(-\sqrt{3}r\right), \text{where } r = \sqrt{\sum_{i=1}^{N}\frac{(x_i - x_i')^2}{{l_i}^2}}$$

Generative Model

- Decoder - 4 hidden layers with 100 neurons
- Encoder - 4 hidden layers with 100 neurons
- Discriminator - 2 hidden layers with 100 neurons

Hexadecane Predictions



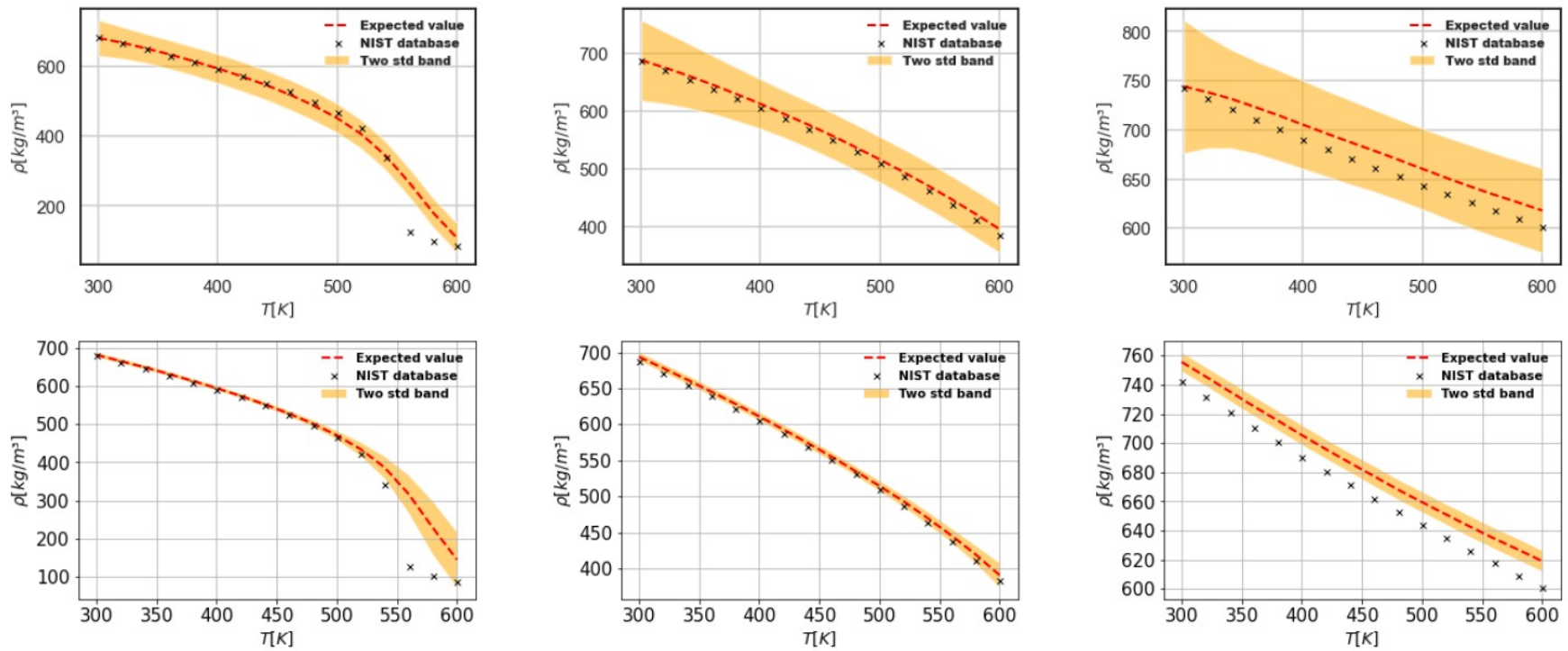
3 MPa

10 MPa

100 MPa

Hexadecane predictions with the Gaussian process surrogate model (top) and Conditional generative surrogate model (bottom) at the pressures 3, 10, and 100 MPa.

Extrapolation on experimental NIST dataset (Heptane - C7H16)



3 MPa 10 MPa 100 MPa

Heptane predictions with the Gaussian process surrogate model (top) and Conditional generative surrogate model (bottom) at the pressures 3, 10, and 100 MPa. The L^2 mean relative error to the density predicted by MD simulation at 3, 10 and 100 MPa are $(GP, GM) = (1.1364 \times 10^{-1}, 2.5697 \times 10^{-1})$, $(GP, GM) = (1.9166 \times 10^{-4}, 1.3294 \times 10^{-4})$ and $(GP, GM) = (4.9254 \times 10^{-4}, 5.5123 \times 10^{-4})$, respectively