Work in Progress

THE ANDREW & ERNA VITERBI FACULTY OF ELECTRICAL ENGINEERING

SIPL
Signal and Image Processing Lab

TECHNION Israel Institute of Technology

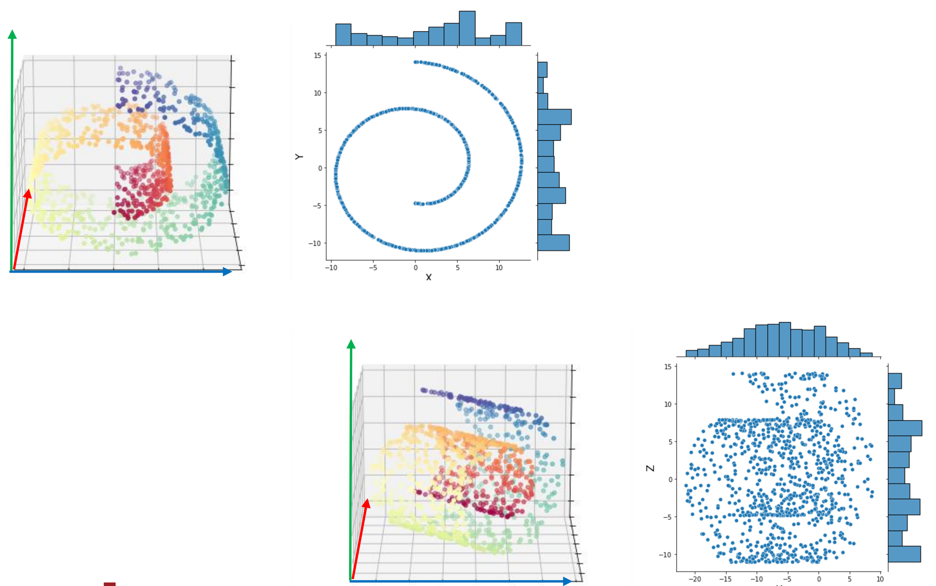
A random-projection based approach for generative modelling

Elad David, Supervised by Prof. Tomer Michaeli

Introduction

- **Generative modelling**
  - deals with the inference problem of high-dimensional data distribution
  - Current methods (mostly Deep NN) require long gradient-based optimization, and have low flexibility for changes
- **Projection based approach**
  - Multi-directional view allow density inference (e.g., CT imaging)

Toy example - Swiss roll - 3D representation & 2D joint plot



Goal

- Develop an **iterative method** for high-dimensional **distribution modelling** with 'state of the art performance'
  - Theoretical - proof of convergence
  - Practice – set a high-performance solution

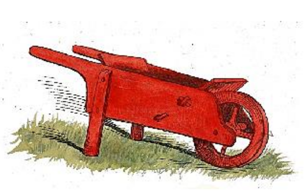
Core idea

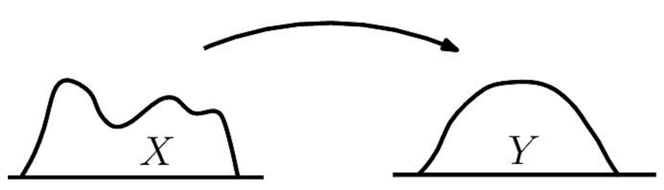
- Data: $\{x_i\}_{i=1}^{N} \in \mathbb{R}^d, \quad X \in \mathbb{R}^{Nxd}$
- Latent gaussian assumption:
  $$z_i \sim \mathcal{N}(0, I^d), \quad Z \in \mathbb{R}^{Nxd}$$
- Execute iteratively:
  - Apply random rotation ("set direction of view")
  - Fit marginal distributions → save the transform coefficients

Marginal fitting

Decompose to Marginals → Sort each marginal → Minimize Sliced Wasserstein distance

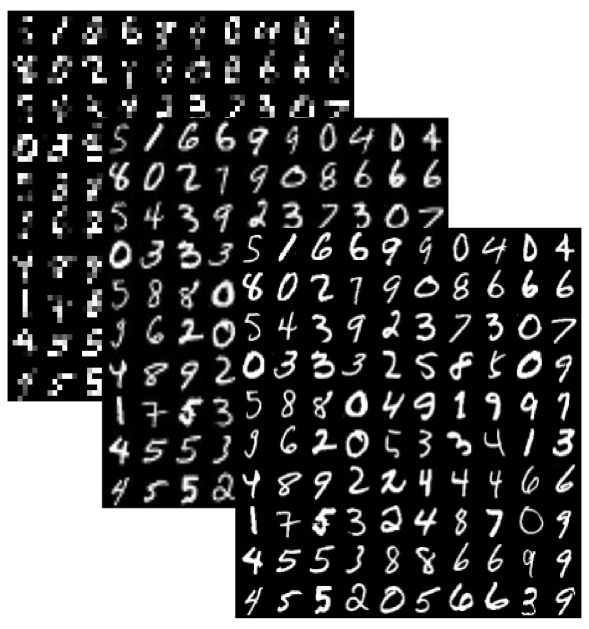
$$\begin{bmatrix} x_{11} \\ x_{21} \\ \cdots \\ \cdots \\ x_{n1} \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \\ \cdots \\ \cdots \\ x_{n2} \end{bmatrix} \begin{bmatrix} x_{13} \\ x_{23} \\ \cdots \\ \cdots \\ x_{n3} \end{bmatrix} \qquad \begin{bmatrix} \tilde{z}_{11} \\ \tilde{z}_{21} \\ \cdots \\ \cdots \\ \tilde{z}_{n1} \end{bmatrix} \xrightarrow{\textit{polyfit}} \begin{bmatrix} \tilde{x}_{11} \\ \tilde{x}_{21} \\ \cdots \\ \cdots \\ \tilde{x}_{n1} \end{bmatrix}$$



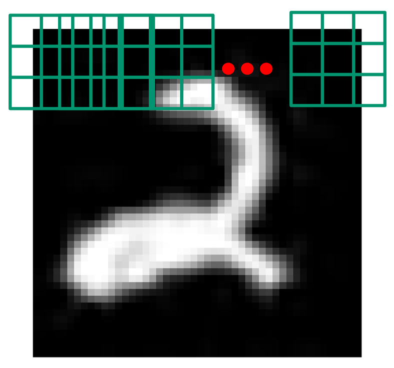


Working in high dimension

- Facing the Curse of Dimensionality
- Utilize **multi-scale** and **spatial relations** (work with patches) concepts
- Scales & patches sizes settings may be viewed as architectures



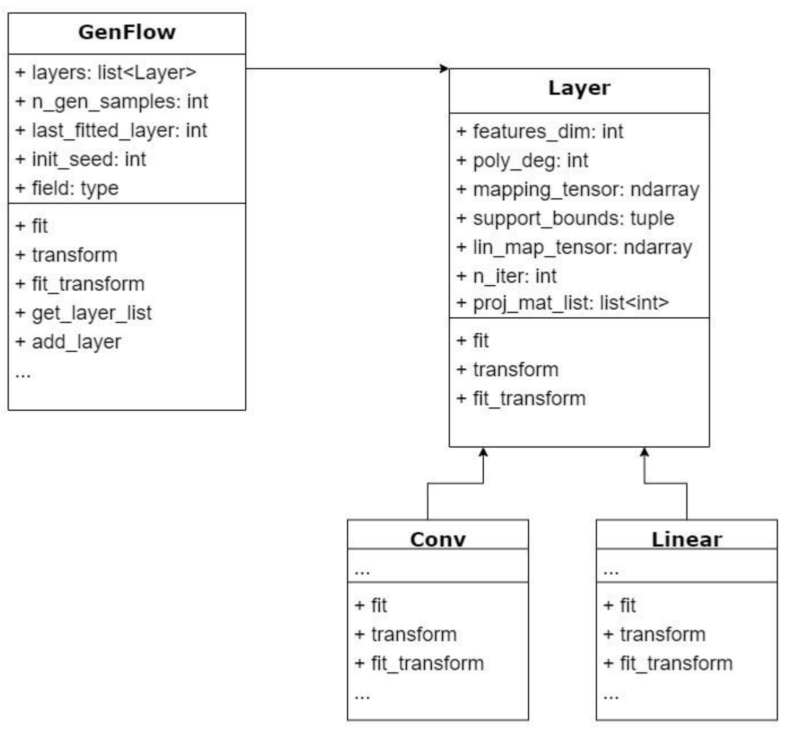
Multi-scale inference
Low → Mid→ High



Splitting to patches

Architecture modularity

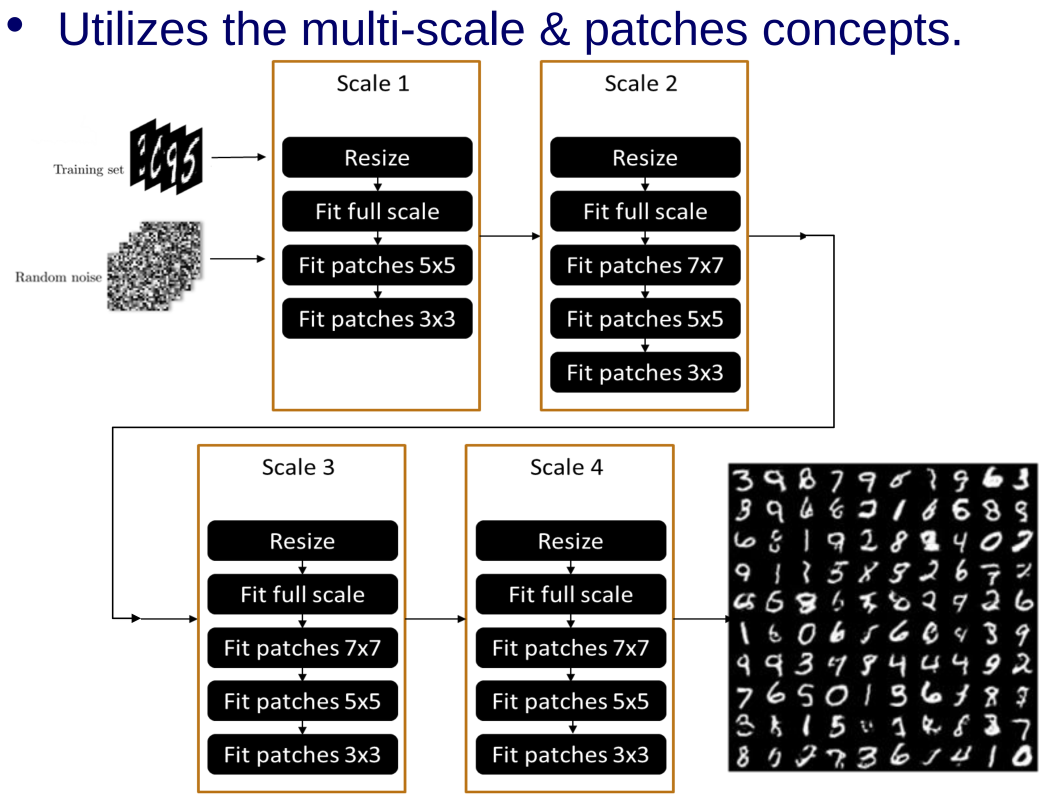
- Python Codebase was developed in order to allow fast and large-scale research for optimal architecture
- OOP approach – an "sklearn/pyTorch"-like model
  - Listing of layer objects – high flexibility
  - Layer – learns transforms in specific dimensions
    "Conv" = patches split



Model Hierarchy – main principals

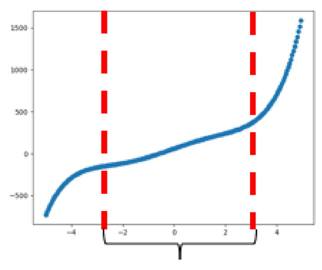
Current Architecture

- Utilizes the multi-scale & patches concepts.



The Fitting Challenge

- Fitting using finite sets raises problems for the generative transform
  - Bounded support – extrapolation is unstable

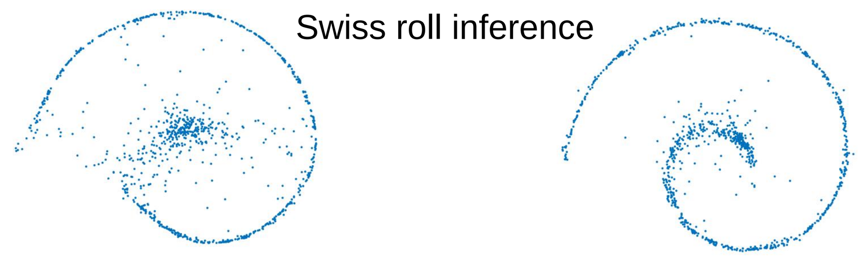


Interp. support
Extrapolation Risk - example

→supports must be saved in memory

- Dependence on polynomial degree

Swiss roll inference



poly degree = 6 poly degree = 9

Results

- Tests were performed over MNIST & CIFAR10 datasets
- MNIST – digits images of size 28x28
  - All labels are similar (digits) – performed learning over the entire dataset

Feedforward results outcome of fitting process on the initial Latent samples

New realizations sampled from gaussian and transformed

- CIFAR10 – animals & vehicles images of size 32x32x3
  - Labels represent different objects ('ship', 'cat', etc.) – **performed fitting on single label**

Feedforward results outcome of fitting process on the initial Latent sample

- New realizations are not successful yet

Conclusions

- Theoretical proof is yet out of hand, but seems feasible due to toy example results and theoretical research
- Performance is highly dependent on runtime. results were achieved by short runs relative to 'state of the art' methods (GANs, etc.)
  → potential for high quality results in the future

June 2021