EAGE
EUROPEAN ASSOCIATION OF GEOSCIENTISTS & ENGINEERS

INCLUDING SPE International EUROPEC

OPPORTUNITIES PRESENTED BY THE ENERGY TRANSITION

2018 EAGE ANNUAL

80TH CONFERENCE + EXHIBITION

COPENHAGEN | DENMARK

bp

REPSOL

11-14 JUNE 2018

WWW.EAGEANNUAL2018.ORG

EAGE ANNUAL 2018
80TH CONFERENCE + EXHIBITION
COPENHAGEN | DENMARK

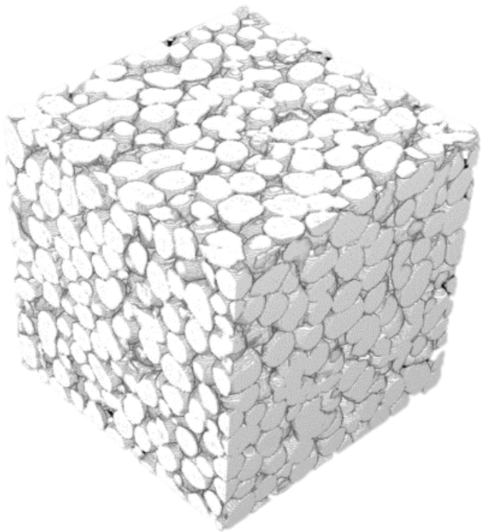
Conditioning of Generative Adversarial Networks for Pore and Reservoir-Scale Models

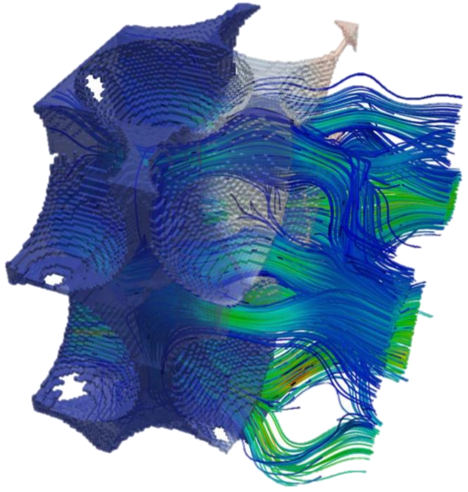
Lukas Mosser, Olivier Dubrule, Martin J. Blunt

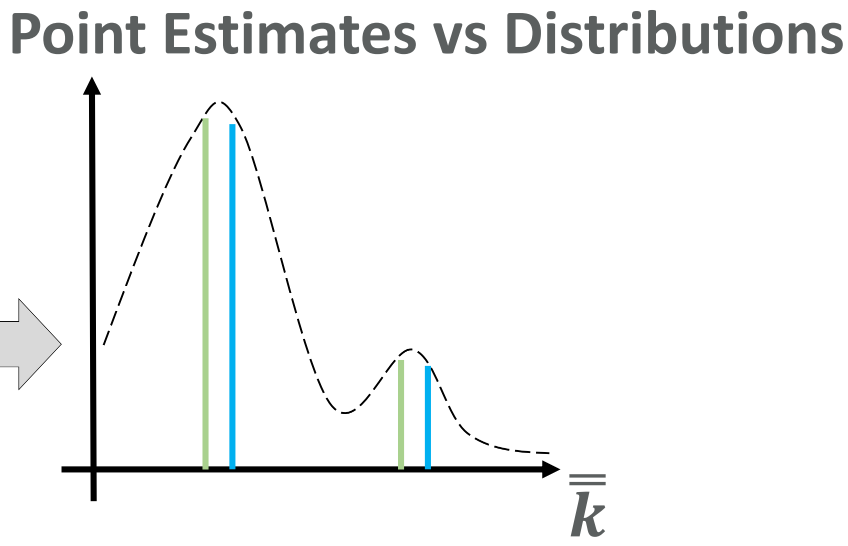
Imperial College London

TOTAL

GANs in practice – Digital Rock and Core Physics







Data Acquisition

Experiments / Imaging

Flow Experiment

Numerical Simulation

Validation

Challenge:

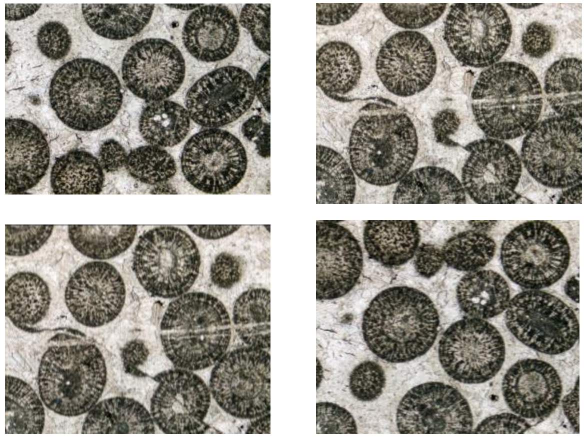
How to quantify variability?

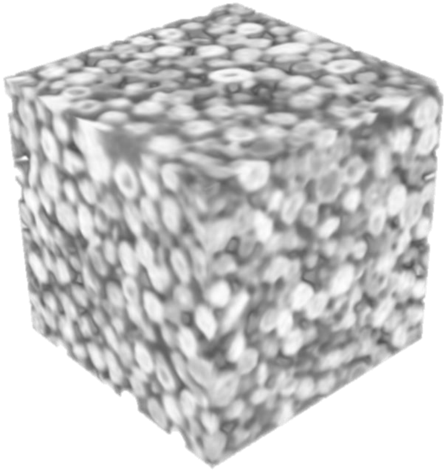
Perform measurements

Generate Representative Images

Why condition your generative model to data?

Obtain a high-resolution 3D scan –> condition 3D model to 2D sections





Thin sections more abundant / cheaper -> infer 3D structure

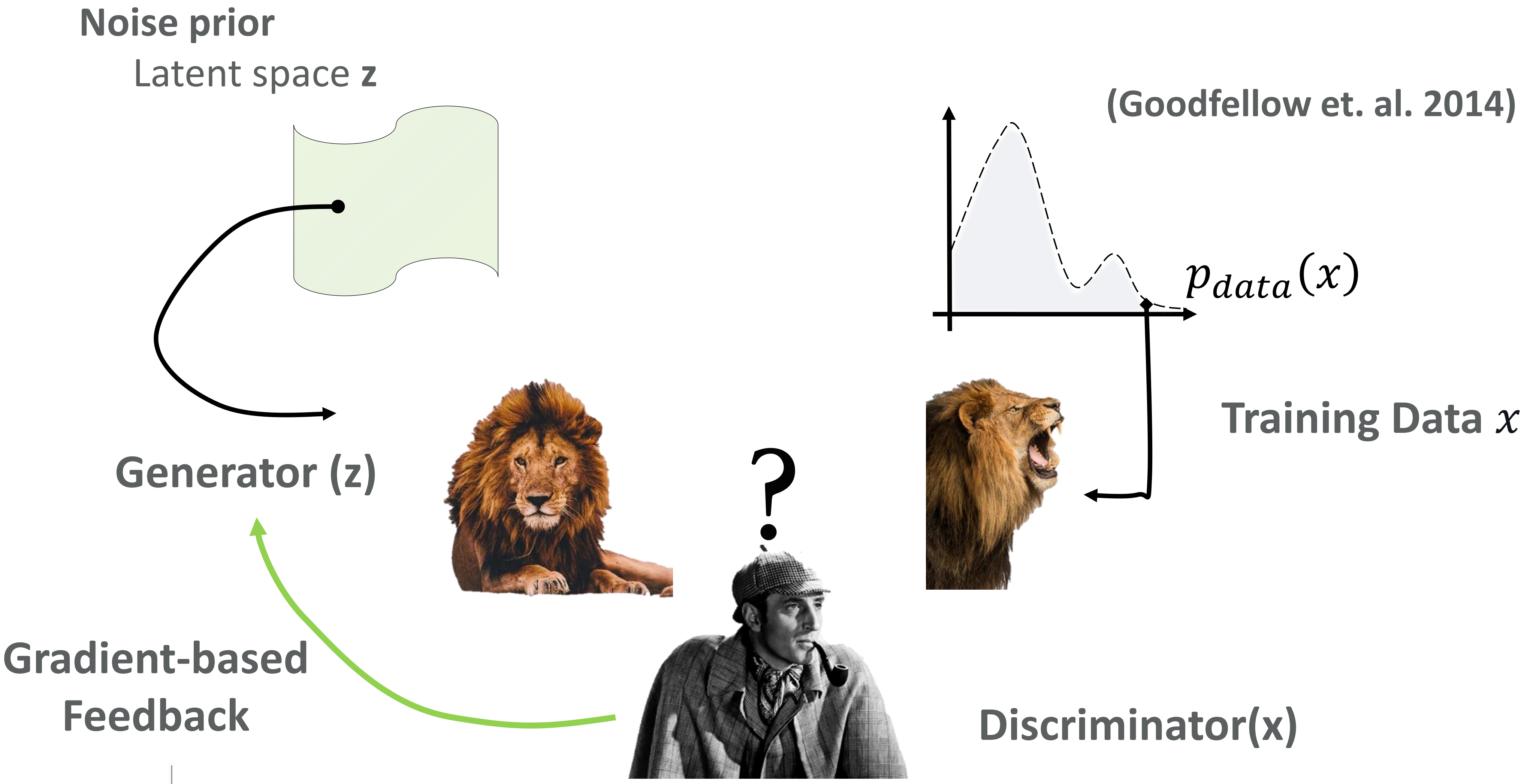
Assists in experimental verification:

- Keep same inflow geometry but vary 3D structure

Outline

- Generative Adversarial Networks
- Dataset – Ketton Limestone
- Validation of the Generative Model
    - Visual, Statistical and Physical Properties
- Conditioning to GANs to spatial data – pore and reservoir scale
- Where to from here?

Generative Adversarial Networks – Toy Example



Generative Adversarial Networks (Goodfellow et. al. 2014)

- Task: Sample from unknown density, given by training samples / images

- Consist of two differentiable functions:
  - Generator $G(z)$ and Discriminator $D(x)$
  - $G(z)$ and $D(x)$represented by differentiable parametric (deep) neural networks

- Sample random noise z and apply function $G(z)$ that maps to data domain e.g. images

- Two cost functions - competitive:
  - Discriminator $D(x)$ goal: Distinguish real data from fake samples
  - Generator $G(z)$ goal: Make samples look like data and fool discriminator
  - Convergence when discriminator confused therefore:

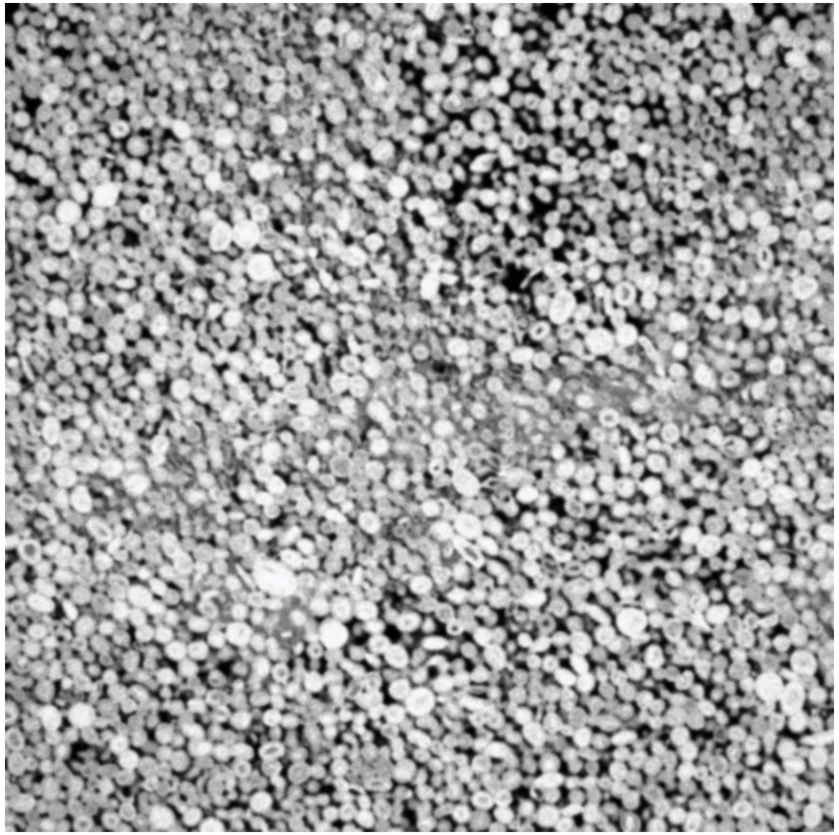
$$p_{data}(x) = p_{generated}(x)$$

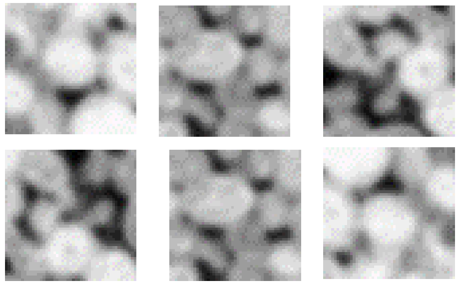
Allows very fast sampling (GPU optimized) of large 3D images

Ketton Limestone Dataset and Preprocessing

- **Oolitic Limestone**
- **Intergranular pores**
- **Intragranular Micro-Porosity**
- **Ellipsoidal grains**
- **99% Calcite**
- **Image Size:**
  - $\mathbf{900^\wedge 3}$ voxels @ 26.7 μm

Extract Non-Overlapping Training Images ($\boldsymbol{64^3\ voxels}$)

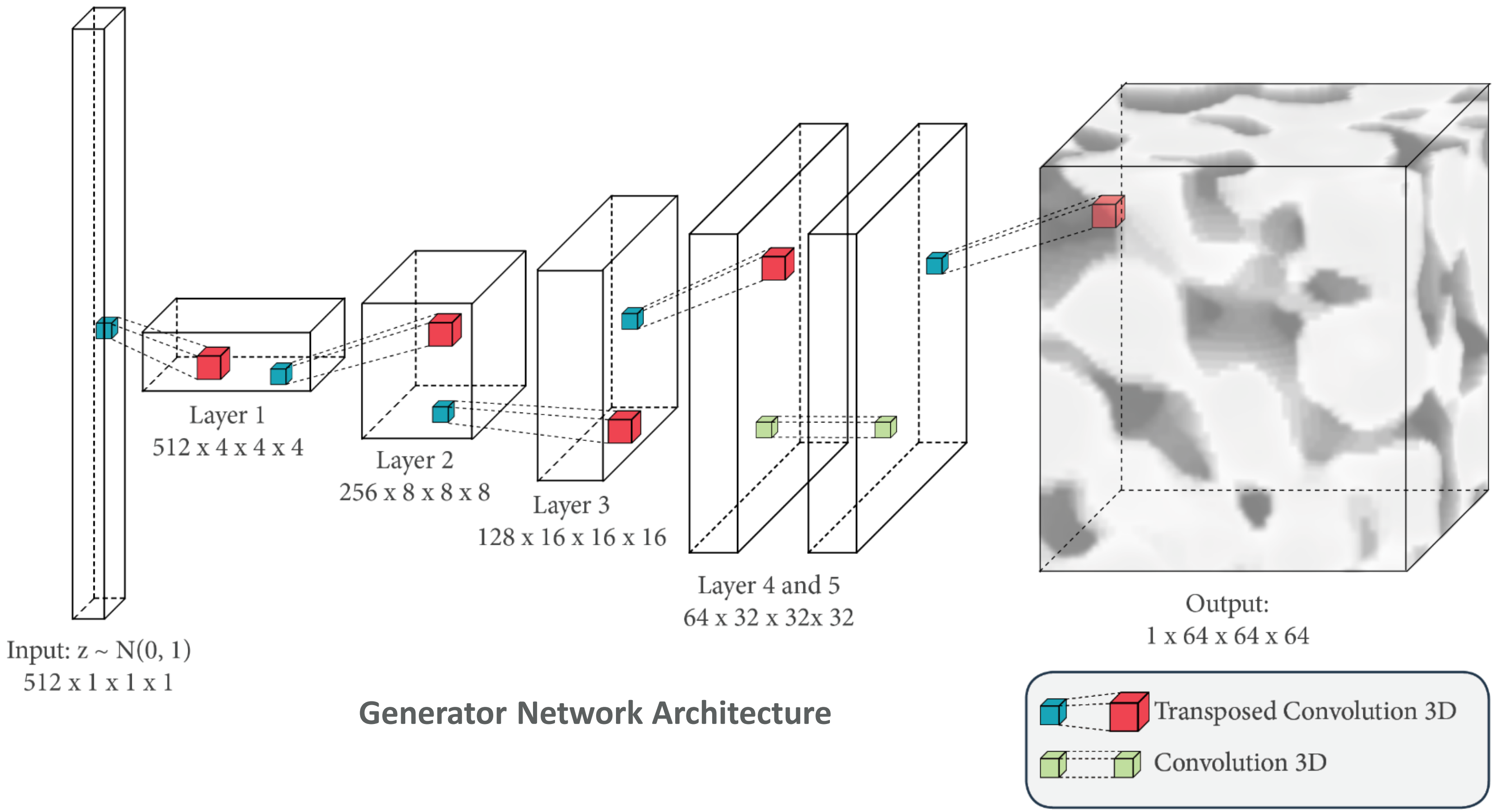




Training Set

Network Architecture – 3D Convolutional Network

Represent $G(z)$ and $D(x)$ as deep neural networks:

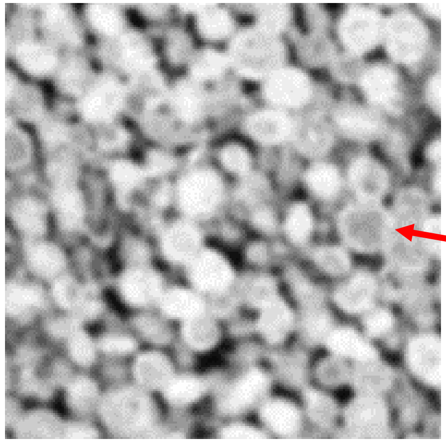


Generator Network Architecture

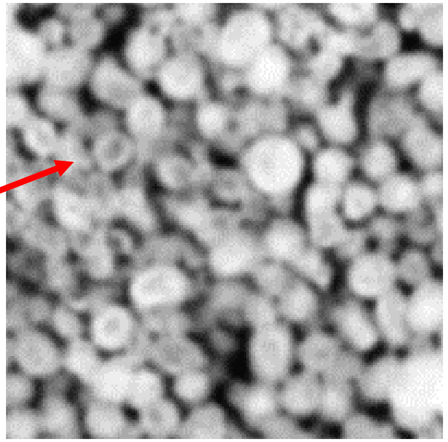
Discriminator: Binary Classification Network -> Real / Fake

Reconstruction Quality – Unconditional Simulation

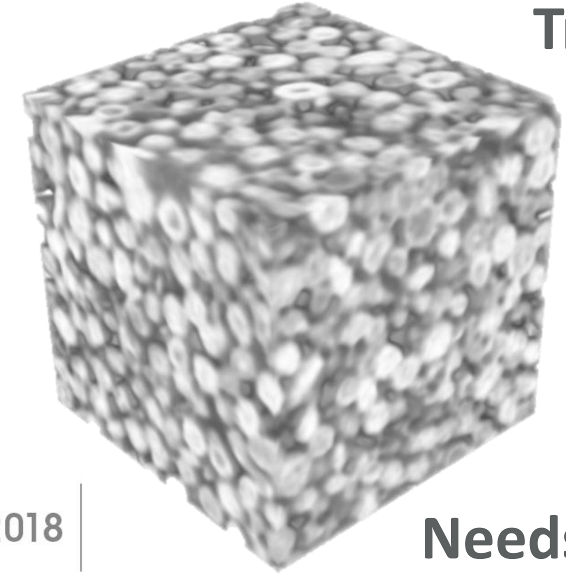
Ketton Training Image

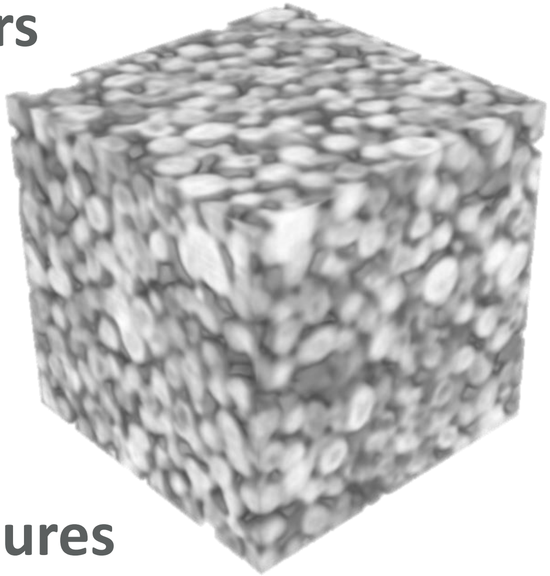


GAN generated sample



Intergranular Porosity
Moldic Features
Micro-Porosity





Training Time: 8 hours

Generation: 5 sec

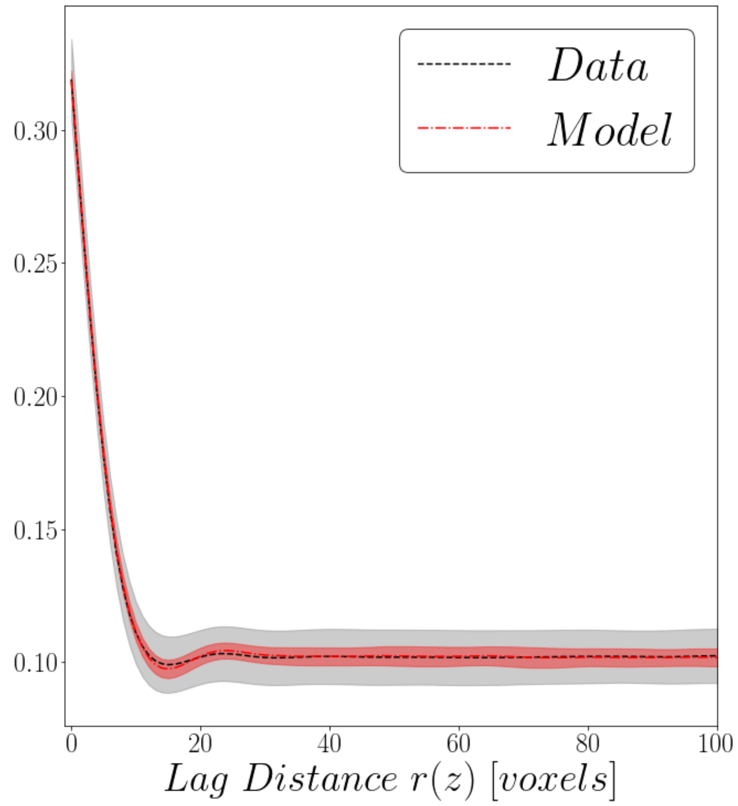
High visual quality
Needs quantitative measures

Reconstruction Quality Criteria

Statistical Properties

- **Two-Point Probability Function $S_2(r)$**
  - » **Radial Average / Directional**

$$S_2(\mathbf{r}) = \mathbf{P}(\mathbf{x} \in P, \mathbf{x}+\mathbf{r} \in P) \;\; for \;\; \mathbf{x}, \mathbf{r} \in \mathbb{R}^d$$



Minkowski Functionals

- **Porosity ϕ**
- **Specific Surface Area S_v**
- **Integral of Mean Curvature**
- **Specific Euler Characteristic χ_v**
- **Compute as function of image gray-level => Characteristic Curves**

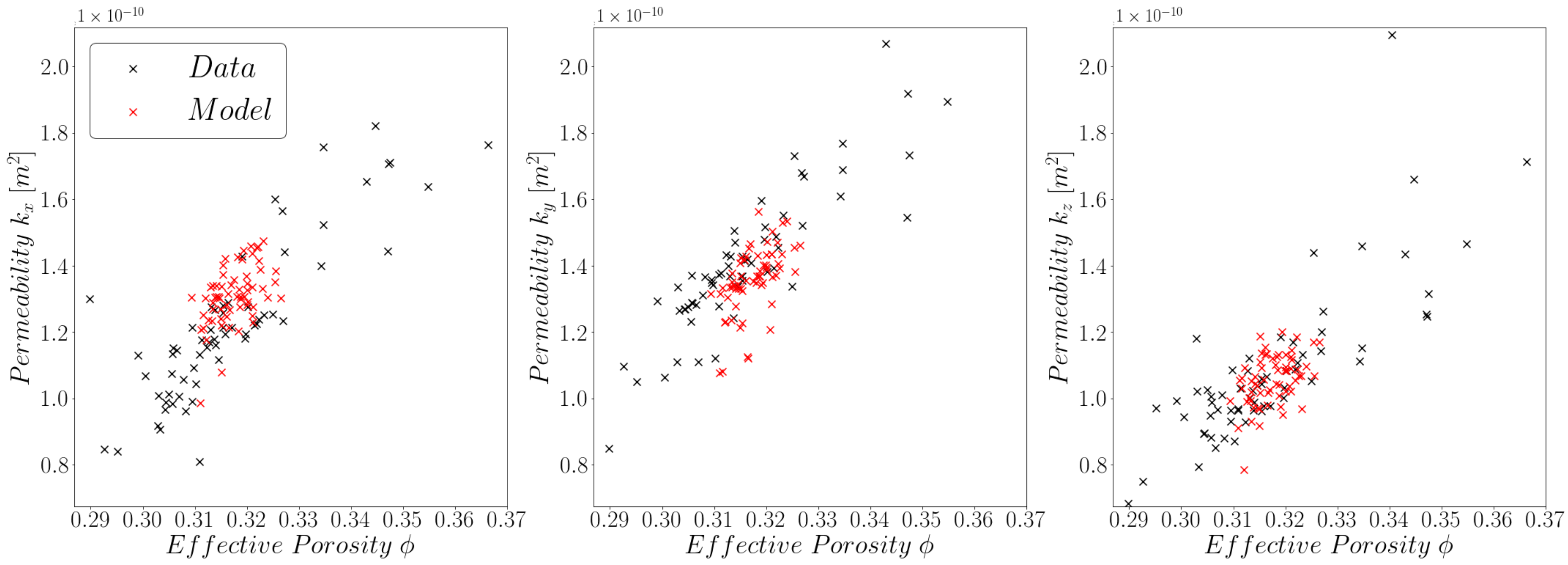
Flow Properties: Solve Stokes flow in pore domain

- **Permeability + Velocity Distributions**

$$\nabla \cdot \mathbf{v} = 0$$

$$\mu \nabla^2 \mathbf{v} = \nabla p$$

Ketton Comparison – Permeability



Isotropic Permeability

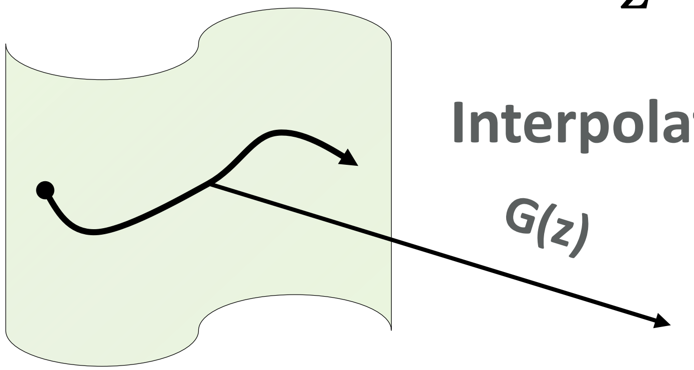
Range of effective (flowing) porosity: Data (0.29- 0.37) GAN (0.31-0.33)

Same order of magnitude and $\overline{\overline{k}} - \phi$ relationship

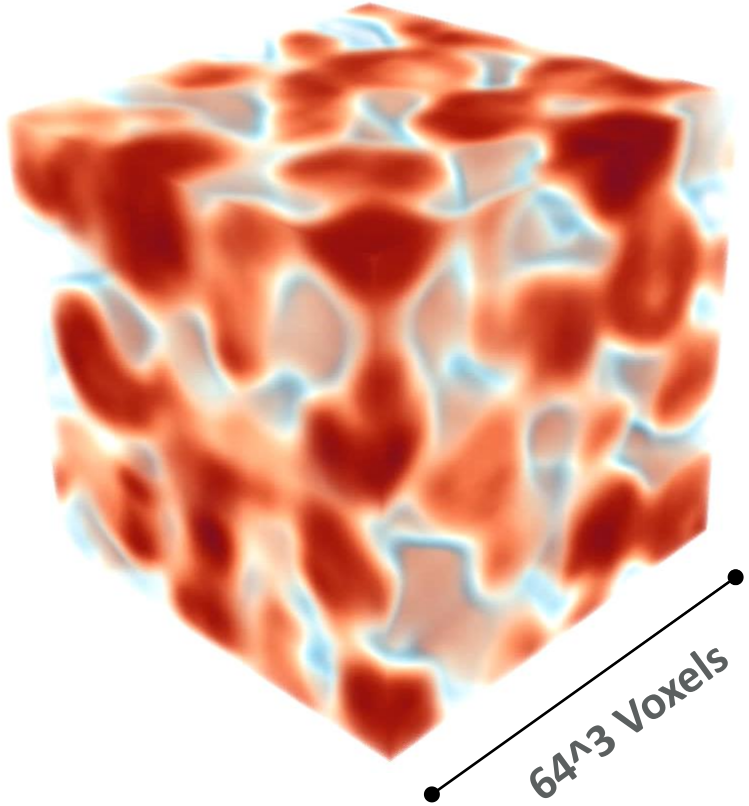
Latent Space Interpolation

Latent space **z**

$$z^* = \beta\, z_{start} + (1-\beta)\, z_{end}, \beta \in [0,1]$$



Interpolation path visualization



Interpolation in latent space:

Shows that generator has learned a meaningful representation in a lower dimensional space!

Generative Adversarial Networks are:

- **Generative Models**
- **Parametric – Latent Vector**
- **Differentiable – Allow for optimization (next part!)**
- **Paired – Generator (Sampling) and Discriminator (Evaluation)**

They can be:

- **Difficult to train – mode collapse and stability**
- **Difficult to evaluate – image quality and diversity**

Once trained and qc’ed:

- **Fast sampling**
- **Good computational scaling – with image size**
- **Load from Checkpoints - no need to start from scratch**

Conditioning of Generative Models

Generative model should incorporate additional information:

e.g. G(z, class) -> Sandstone, Shale, Carbonate

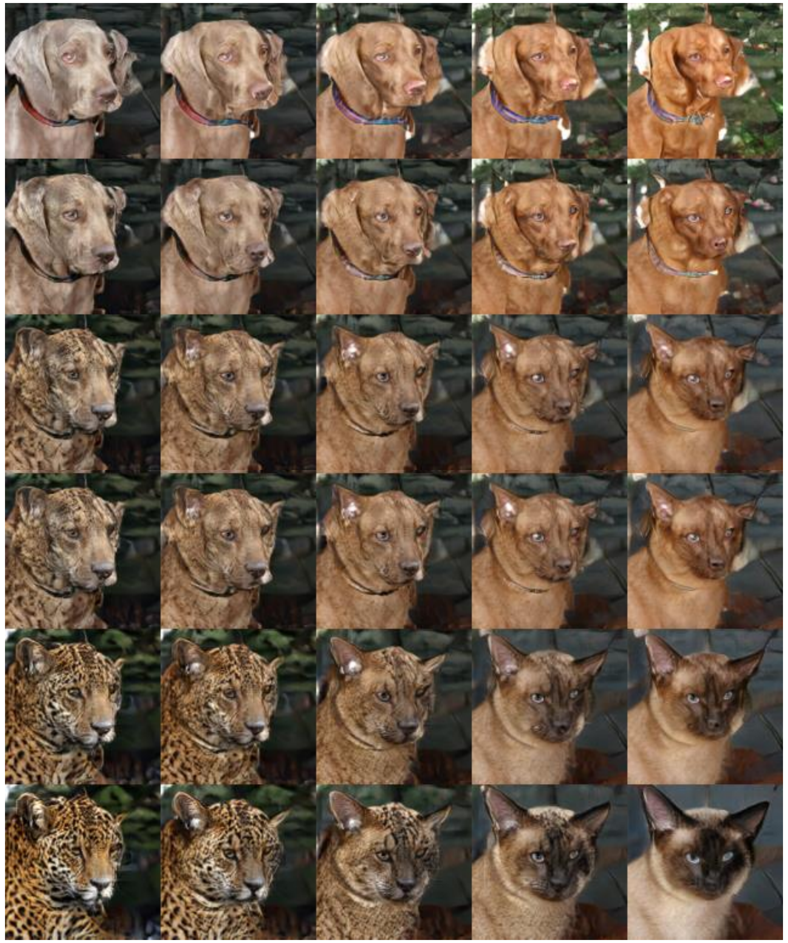
Train one generative model on images with associated class labels.

Generative model: Latent Vector + Class

Possible to perform smooth inter-class interpolation
E.g. Sand to shale to carbonate facies

This model -> Trained on Imagenet (1000 classes)

(Cat, Dog, Leopard, Dachshund)

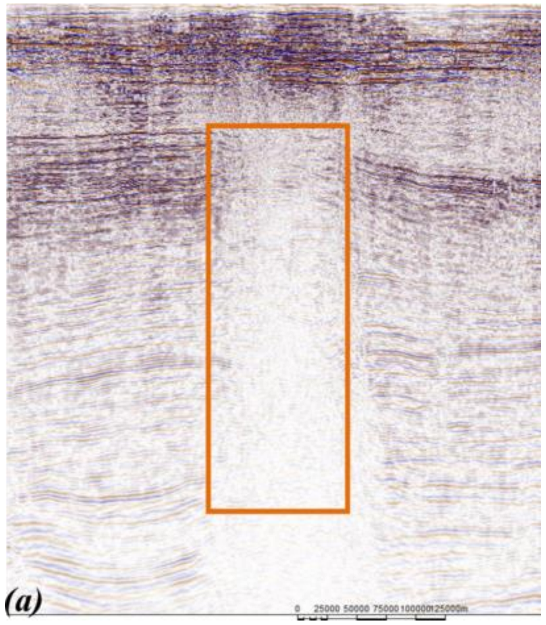


Miyato and Koyama 2018

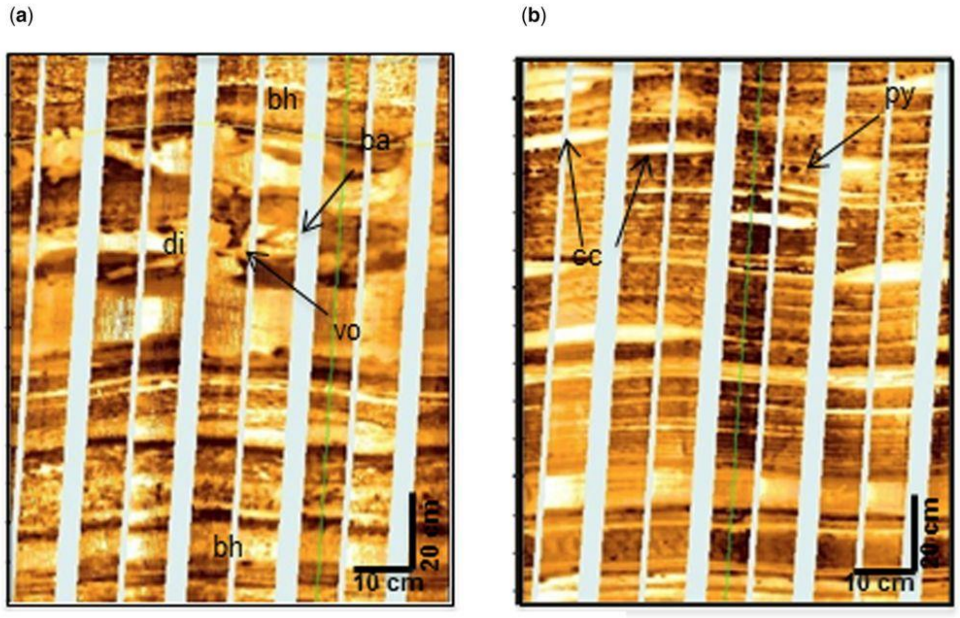
Conditioning of Generative Models

Individual samples should respect available spatial data:

- **Pore Scale: Thin Sections, FIB-SEM, Porosity, Rock Type**
- **Reservoir Scale: Logs, Core Facies, Production Data, Seismic**



Gas Chimney



Formation Micro-Imaging

Image Inpainting (Yeh et al. 2016)

Task: Restore missing details given a corrupted / masked image $M \cdot \tilde{x}$

Use a generative model G(z) to find missing details, conditional to given information.

$$\left.\begin{array}{ll}\textbf{Contextual Loss:} & L_{content} = \lambda \left|\left| M \cdot G(z) - M \cdot \tilde{x} \right|\right|_2 \\ \textbf{Perceptual Loss:} & L_{perc} = log(1 - D(G(z)))\end{array}\right\}$$

Optimize loss by modifying latent vector z

$M \cdot \tilde{x}$ | **Human Artist** | $L_2\ Loss$ | $L_{content}$ + L_{perc}

Credit: Kyle Kastner

Conditioning – Pore Scale Example

Two-dimensional data at pore-scale more abundant e.g. thin-sections

Combine 3D generative model G(z) with 2D conditioning data

Generative Model: Ketton Limestone GAN (Part 1)

Mask: Three orthogonal cross-sections, honor 2D data in a 3D image

Contextual Loss: $L_{content} = \lambda||M \cdot G(z) - M \cdot \tilde{x}||_2$ **on orthogonal cross-sections**

Perceptual Loss: $L_{perc} = log(1 - D(G(z)))$ **on whole volumetric generated image G(z)**

$$L_{Total} = \lambda \, L_{content} + L_{perceptual}$$

Optimize Total Loss, by modifying latent vector (GAN parameters fixed)

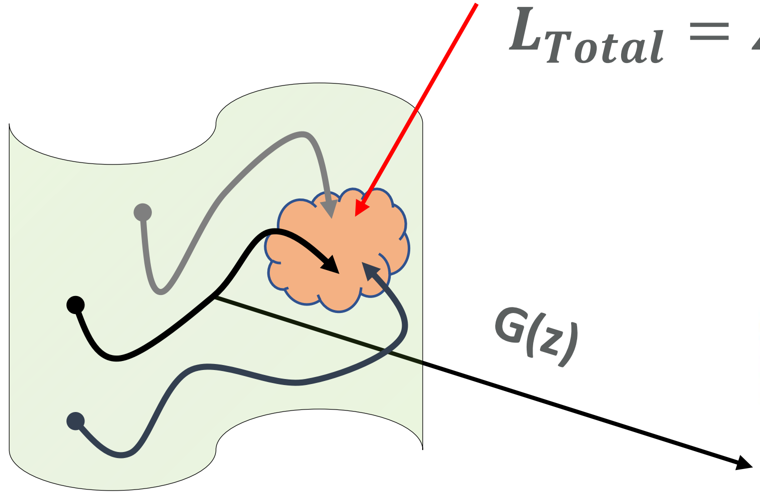
-> Many local minima at error threshold -> stochastic volumes that honor 2D data

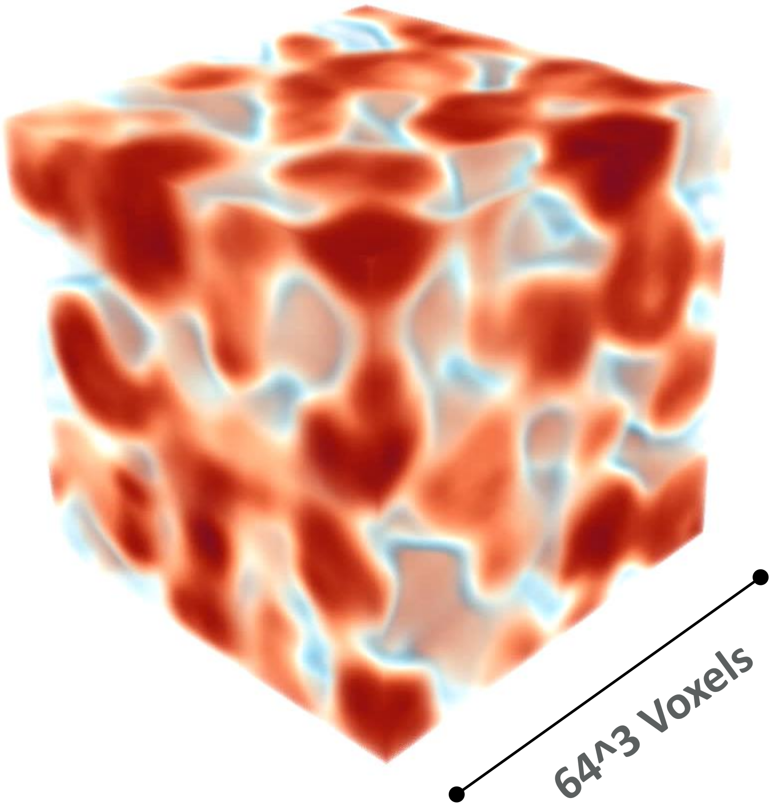
Conditioning Generative Networks

Traverse latent space by gradient-descent

Subspace of all images that match the data

$$L_{Total} = \lambda\, L_{content} + L_{perceptual}$$



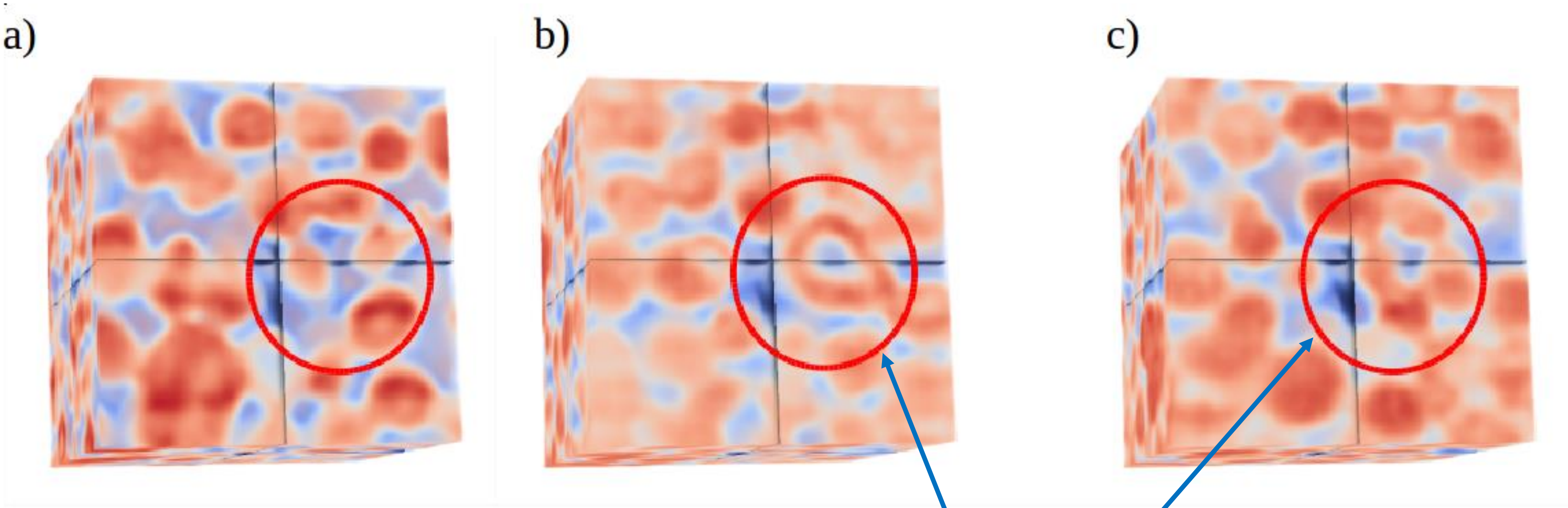


Latent space **z**

Space of all valid unconditional images

Start from many random starting locations
- > stochastic conditioned samples

Conditioning – Pore Scale Example



Conditioning Data
Ground Truth Volume

Stochastic Sample 1
Conditioned to Data

Stochastic Sample 2
Conditioned to Data

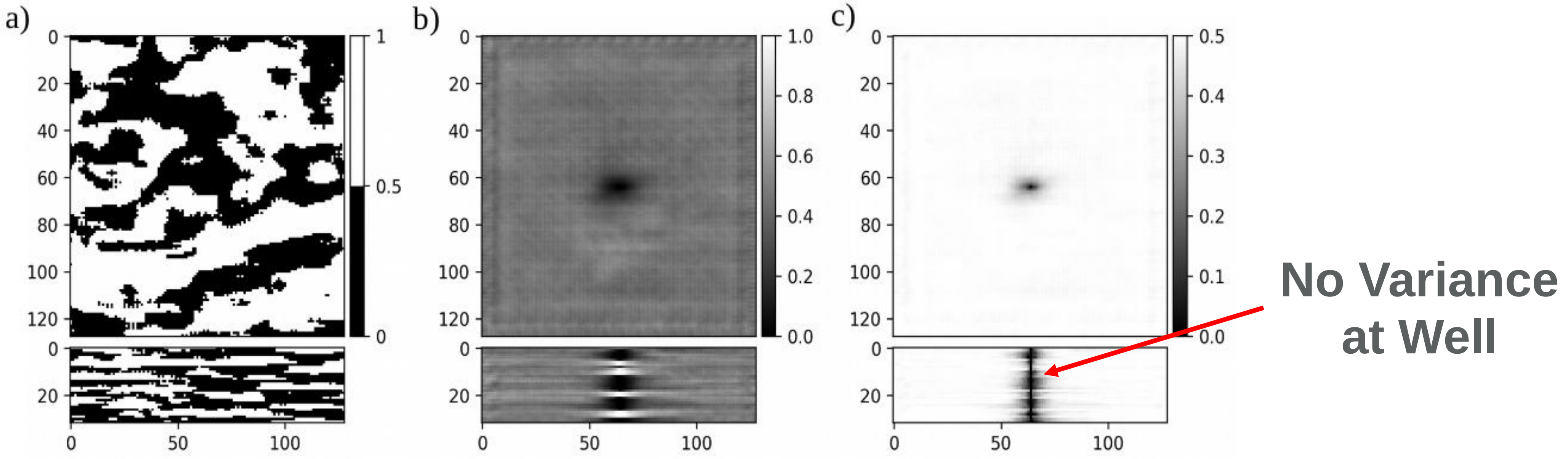
Same 2D conditioning data leads to diverse realizations in 3D

Conditioning – Reservoir Scale Example

Maules Creek Training Image (Credit G. Mariethoz)

Pre-trained 3D-Generative Adversarial Network

Condition to single well (1D conditioning) from ground truth data



Single Realization **Mean (N=1000)** **Standard Dev. (N=1000)**

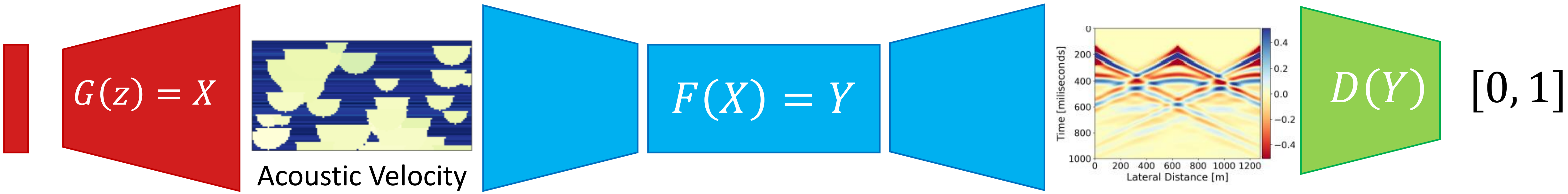
GANs: A game changer for modeling and inversion?

Summary:

- Stochastic Modeling using GANs at Pore and Reservoir Scale
- GANs provide flexible framework for conditioning to existing data – label and spatial data

Many inverse problems could benefit from this:

- Reservoir Scale: Permeability derived from production data
- Seismic imaging: Acoustic/elastic properties from seismic data -> ArXiv Preprint



Latent Vector
+
Generative Model

Differentiable Forward Model

Differentiable Geological Prior + Forward Operator

Imperial College London

TOTAL

Thank You!
Questions?

Code on GitHub https://github.com/LukasMosser/geogan

lukas.mosser15@imperial.ac.uk