Technische Universität Berlin

*Gen*AI4*Numerical*Math

Hanno Gottschalk (TU Berlin) joint work with C. Drygala, F. di Mare, E. Ross, M. de Campos, W. Krebs, L. Schwartz...
Institute of Mathematics, TU Berlin

ai4math WS U Potsdam | 5. August 2026

The Challenge of Generative Learning

From NVIDEA Style GAN, `thispersondoesnotexist.com`, Karras, Laine, Alia 2018

- **Challenge:** $X_j \in \mathbb{R}^d$ is high dimensional, $d \sim 10^6$.

Generation by Transport Maps

- Suppose $X_j \sim \mu$, μ probability mesure on $\Omega \subseteq \mathbb{R}^d$
- Transform noise $U_j \sim \nu$ with **generator** $\phi : \Omega \to \Omega$

$$\phi_* \nu = \nu \circ \phi^{-1} \approx \mu$$

- $\phi_* \nu(A) = \nu(\phi^{-1}(A))$ is the image measure of the noise measure under ϕ.
- Learn ϕ in some hypothesis space $\mathcal{H}$ from the data X_j
- $\mathcal{H}$ might be high but finite dimensional – e.g. Neural Networks
- Or infinite dimensional space of Functions.

Assumptions and Existence of a Generator

- Source measure ν and target measure μ have a density wrt the Lebesgue measure on Ω

$$d\nu(x) = f_\nu(x)\, d\lambda(x), \quad d\mu(x) = f_\mu(x)\, d\lambda(x)$$

- f_ν, f_μ is a k-times differentiable function
- $f(x) \geq \kappa > 0$ on Ω.

Theorem (Realizability):
(i) There exists a k times differentiable function $\phi_0 : \Omega \to \Omega$ such that

$$\phi_{0*}\nu = \mu.$$

Flow maps

- Want to realize ϕ as a flow endpoint w.r.t a vector field $v(x,t)$,

$$\dot{\phi}_t(x) = v(\phi(x),t), \quad \phi = \phi_1$$

- Does a suitable $v(x,t)$ exist?
- Continuity equation

$$\partial_t f_t(x) = \nabla \cdot (v(x,t) f_t(x)), \quad f_0(x) = f_\nu(x), \quad f_1(x) = f_\mu(x) \quad (!)$$

Beckmann problem

- Model the influx $w(x) = v(x,t)f_t(x)$ time independent such that

$$\nabla \cdot w(x) = f_\mu(x) - f_\nu(x), \quad x \in \Omega, \quad n \cdot w = 0, \quad x \in \partial\Omega. \tag{1}$$

- Then from the continuity equation for $v(x,t) = \frac{w(x)}{f_t(x)}$

$$\partial_t f_t(x) = \nabla \cdot w(x) = f_\mu(x) - f_\nu(x) \quad \Rightarrow \quad f_t(x) = t f_\mu(x) - (1-t) f_\nu(x)$$

- Beckmann's problem:

$$\frac{1}{2}\int_\Omega |w|^2 \, dx \to \min \qquad \text{under constraints (1)}$$

Unresticted Beckmann Problem

- Introduce Lagrangian multipliers $u : \Omega \to \mathbb{R}$, $\nabla u(x) \cdot n = 0$ on $\partial\Omega$, $v \in C(\partial\Omega)$, $f = f_\mu - f\nu$

$$\mathcal{L}(w, u, v) = \frac{1}{2}\int_\Omega |w|^2\, dx + \int_\Omega (\nabla \cdot w - f)u\, dx + \int_{\partial\Omega} n \cdot (wv)\, dA$$

- Optimality conditions

$$\frac{\delta}{\delta v}\mathcal{L}(w, u, v) = 0 \;\Rightarrow\; n{\cdot}w = 0 \text{ on } \partial\Omega, \quad \frac{\delta}{\delta v}\mathcal{L}(w, u, v) = 0 \;\Rightarrow\; \nabla{\cdot}w - f = 0 \text{ on } \Omega$$

$$\frac{\delta}{\delta v}\mathcal{L}(w, u, v) = 0 \;\Rightarrow\; -\nabla u = w \text{ on } \Omega$$

Solution of the Optimality Conditions

- $u : \Omega \to \mathbb{R}$ fulfills Poisson equation with zero Neumann boundary conditions

$$-\Delta u = f \quad \text{on} \quad \Omega \quad \nabla u \cdot n = 0 \quad \text{on} \quad \partial\Omega$$

- Weak formulation not coercive, but fulfills condition for Fredholm alternative with $g = 0$

$$\int_\Omega f \, dx - \int_{\partial\Omega} g \, dA = \int_\Omega f_\mu - f_\nu \, dx = 0.$$

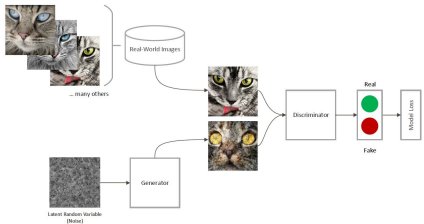
- Classical solution exists (Nadri 2015).

Schauder Estimates

Theorem

- u esists and is $k+2$ times differentiable (Schauder estimates, Agmon, Douglis, Nirenberg 59)
- $\Rightarrow w$ is $k+1$ times differentiable
- $\Rightarrow v$ is k-times differentiable (Asatryan et al, 2023)
- $\Rightarrow \phi_t$ is k times differentiable
- Transport map is regular and can be well approximated by NN, (Yarotzky 2017).

Generative Adversarial Learning



- Min-Max two player game $\hat{\phi} \in \arg\min_{\phi \in \mathcal{H}} \sup_{D \in \mathcal{H}_D} \hat{L}(\phi, D, \{X_j\})$
- With $X_j \sim \mu$, $U_j \sim \nu$, the empirical loss is defined as

$$\hat{L}(\phi, D, \{X_j\}) = \frac{1}{2n} \sum_{j=1}^{n} [\log(D(X_j)) + \log(1 - D(\phi(U_j))]$$

Empirical and Theoretical Loss in Adv. Learning

Theorem (Goodfellow et al 2014):
(i) The expected loss

$$L(\phi, D, \{X_j\}) = \mathbb{E}_{X_j \sim \mu}[\hat{L}(\phi, D, \{X_j\})]$$

is maximized by $D_\phi = \frac{f(x)}{f(x)+f_\phi(x)}$, provided this is realizable in $\mathcal{H}_D$;
(ii)

$$L(\phi, D_\phi, \{X_j\}) = \frac{1}{2}\left[\mathrm{d}_{KL}\left(\mu \middle\| \frac{\phi_*\nu + \mu}{2}\right) + \mathrm{d}_{KL}\left(\phi_*\nu \middle\| \frac{\phi_*\nu + \mu}{2}\right)\right] + \log(2) = \mathrm{d}_{JS}\left(\mu || \phi_*\nu\right) + \log(2).$$

Error Decomposition for Generative Adversarial Learning

Technische Universität Berlin

- Asatryan, G., Lippert, Rottmann (2020), Biau, Cadre, Sangnier, Tanielian (2020)

$$
\begin{aligned}
\mathrm{d}_{JS}(\mu||\hat{\phi}_*\nu) &= L(\hat{\phi}, D_{\hat{\phi}}) - \log(2) \leq \hat{L}(\hat{\phi}, D_{\hat{\phi}}) + \sup_{\phi,D} |L(\phi, D) - \hat{L}(\phi, D)| - \log(2) \\
&\leq \hat{L}(\hat{\phi}, \hat{D}) + \sup_{\phi,D} |L(\phi, D) - \hat{L}(\hat{\phi}, D_{\hat{\phi}})| - \log(2) \\
&\leq \hat{L}(\phi, \hat{D}) + \sup_{\phi,D} |L(\phi, D) - \hat{L}(\phi, D)| - \log(2) \\
&\leq L(\phi, \hat{D}) + 2 \sup_{\phi,D} |L(\phi, D) - \hat{L}(\phi, D)| - \log(2) \\
&\leq L(\phi, D_{\phi_0}) + 2 \sup_{\phi,D} |L(\phi, D) - \hat{L}(\phi, D)| - \log(2) \\
&= \mathrm{d}_{JS}(\mu||\phi\nu) + 2 \sup |L(\phi, D) - \hat{L}(\phi, D)| \to 0 \quad \text{(ULLN)}
\end{aligned}
$$

Turbulence

Turbulence in Fluid Dynamics

- Turbulent flow is **chaotic** by nature
- Computational Fluid Dynamics (CFD): Simulation of turbulences by numerically solving Navier-Stokes equations:

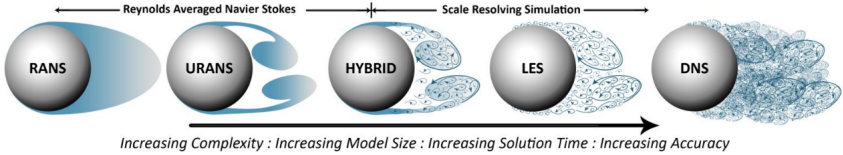
$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0$$

$$\frac{\partial (\rho u)}{\partial t} + \nabla \cdot [\rho u \otimes u] = -\nabla p + \nabla \cdot \tau + \rho f$$

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot ((\rho e + p) u) = \nabla \cdot (\tau \cdot u) + \rho f \cdot u + \nabla \cdot \dot{q} + r$$

- But: Turbulent flow develops ever smaller and faster structures which are hard to resolve numerically

Simulation of Turbulence



Comparison of turbulence modeling approaches. Source: (J. Hart 2016)

- Modeling turbulences challenging in practice but highly technical relevant
- LES/DNS: Simulation of large or even all scales of turbulence but: Enormous computational costs
- RANS/URANS: Struggle in simulation of fine details in vortex flow but: Computational costs are sustainable $\Rightarrow$ mostly used in industry

Ergodicity: What Does "Chaotic" Actually Mean?

- Vortices always look different but in the long run their statistical properties are determined **(ergodicty)**:

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T f \circ \varphi_t(x_0)\, \mathrm{d}t = \int_\Omega f(x)\, \mathrm{d}\mu(x) \;\forall x_0 \in \Omega. \tag{2}$$

  $\Rightarrow$ The time average of a dynamical system equates with the ensemble average of its invariant measure.

- Probability measure μ on the flow configurations x encodes the statistical properties of the chaotic flow/dynamics $\varphi_t(x_0)$
- Goal: Sampling from the **unknown** measure $\mu \Rightarrow$ Can we **learn** μ **from data**?

Changing the Data Model from i.i.d. to Ergodic Data

- **Theorem (Drygala, Winhardt, di Mare, G. 2021)** Consider the Min-Max two player game for ergodic data

$$\hat{\phi} \in \arg\min_{\phi \in \mathcal{H}} \sup_{D \in \mathcal{H}_D} \hat{L}_T(\phi, D, x_0, \{U_j\})$$

- with $x_0 \in \Omega$, $U_j \sim \nu$, the empirical loss **for ergodic flow** is defined as

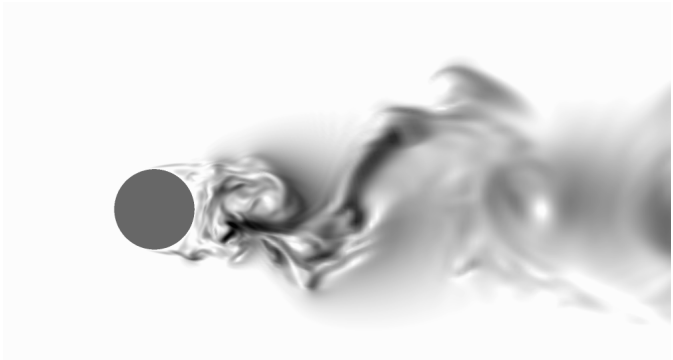
$$\hat{L}_T(\phi, D, \{X_j\}) = \frac{1}{2T} \int_0^T \log(D(\varphi_t(x_0))\, dt + \frac{1}{2T} \sum_{j=1}^{[T]} \log(1 - D(\phi(U_j)).$$

- **Proof:** Error decomposition + use uniform ergodicity instead of ULLN □

Technische Universität Berlin

Karman Vortex Street - Data

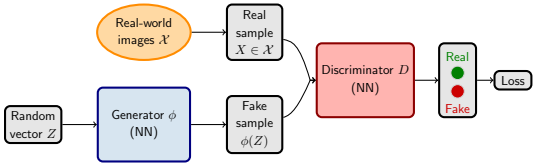
- Dataset: $5,000$ images produced by LES
- Images: $w \times h = 1,000 \times 600$



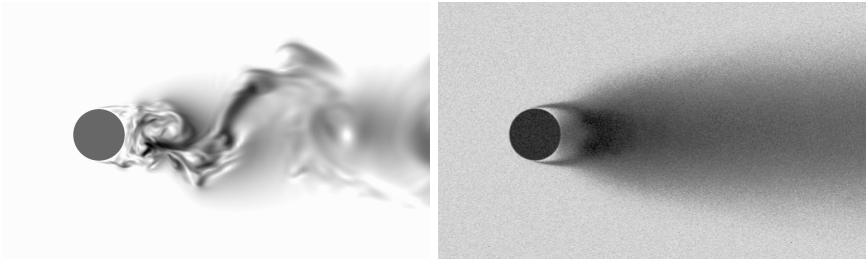
Karman Vortex Street - Training and Inference I

Training

- GAN frameworks:
  - Vanilla GAN
  - Wasserstein GAN
  - Deep convolutional GAN
- Epochs: 200
- Batch size: 20
- Input:
  - $5,000$ LES images
  - Image size: $k \times k$, $k \in \{64, 128, 256, \mathbf{512}\}$
  - Noise vector $Z \in \mathbb{R}^{100}, Z \sim N(0, 1)$



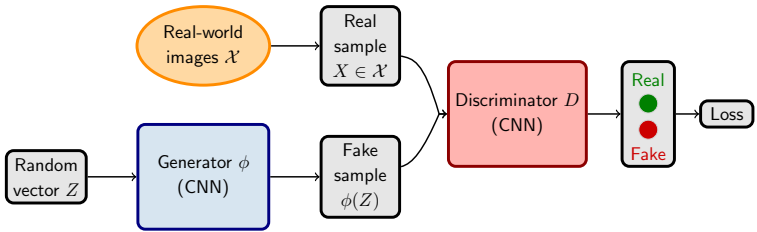
Karman Vortex Street - Vanilla GAN



LES image | Fake image

- **Experiment settings:** Architecture of ϕ and D: Fully connected NN with 5 layers
- Optimizer: Adam, Learning rate: $2 \cdot 10^{-4}$

Deep Convolutional Generative Adversarial Network



Architecture of DCGAN.

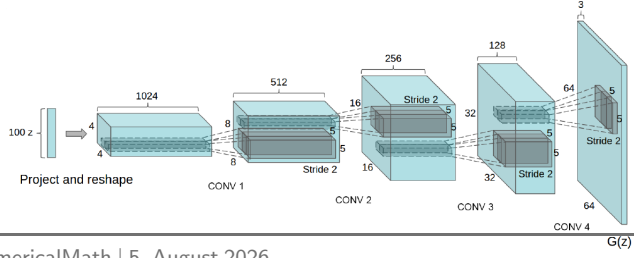
- Generator and discriminator are convolutional neural networks (CNNs)
- CNNs expecially successful and applicable in field of image processing

Deep Convolutional Generative Adversarial Network

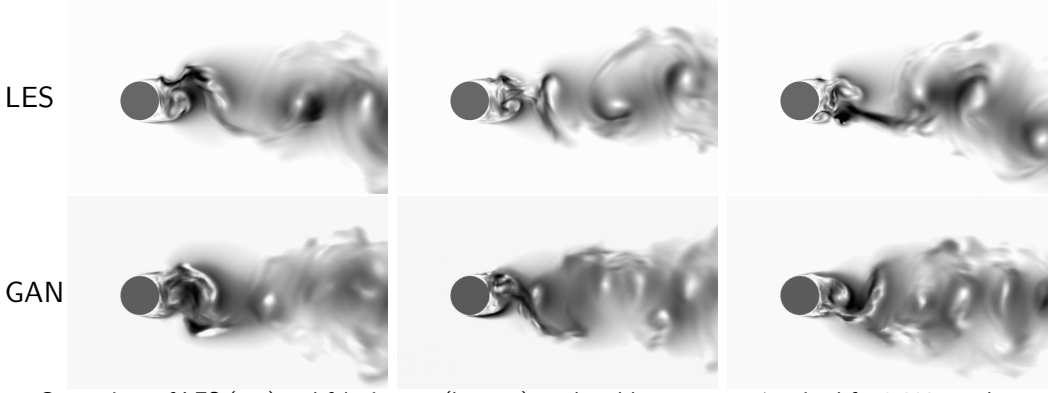
Technische Universität Berlin

Guidelines to follow for **stable training** at **higher resolution** and **deeper architectures**:

- Stability: Apply batch normalization on the output layer of ϕ and the input layer of D
- Deeper architectures: Avoid fully-connected layers on top of convolutional features
- Higher resolution modeling: Leaky Rectified Linear Unit (ReLu) activation function for D
- ϕ and D learn own spatial up- or downsampling by by replacing deterministic spatial pooling layers al 2016)



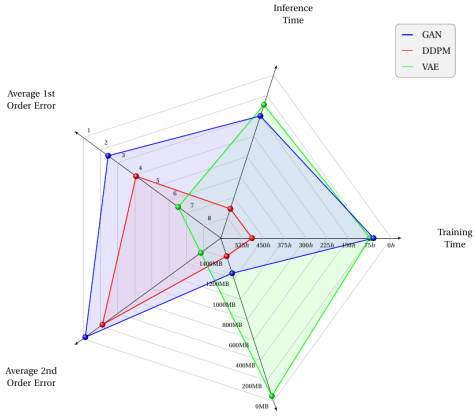
Karman Vortex Street - Inference after $2,000$ Epochs



Comparison of LES (top) and fake images (bottom) produced by generator ϕ trained for $2\ 000$ epochs.

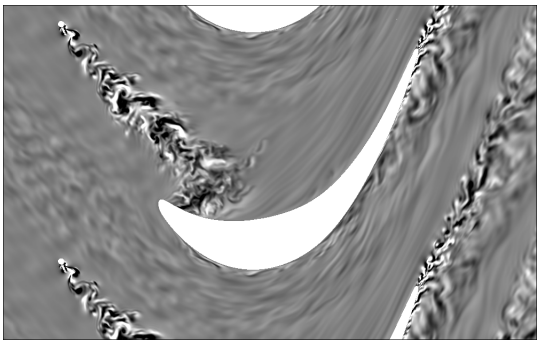
Sampling Results for the Karman Street

Why GAN?

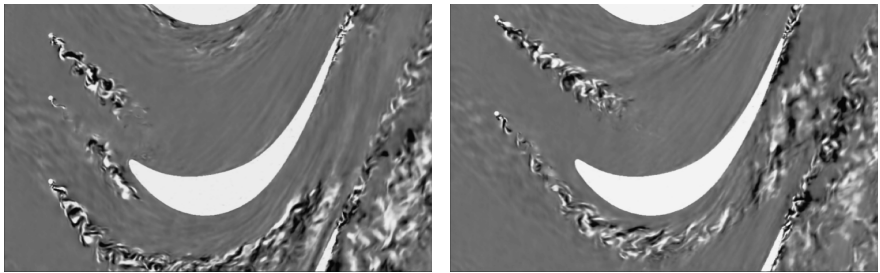

Inference Time
GAN
DDPM
VAE
Average 1st Order Error
Training Time
Average 2nd Order Error
Memory Use
1
2
3
4
5
6
7
8
525h
450h
375h
300h
225h
150h
75h
0h
1400MB
1200MB
1000MB
800MB
600MB
400MB
200MB
0MB

LPT Stator Under Periodic Wake Impact (LPT Stator) - Data

- Dataset: $2,250$ images produced by LES
- Images: $w \times h = 1,000 \times 625$



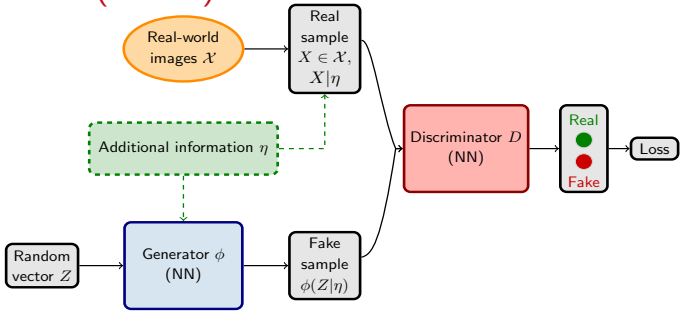
LPT Stator - DCGAN: Inference Results



Examples of images generated by ϕ trained for $2\ 000$ epochs with $2\ 250$ images.

Technische Universität Berlin

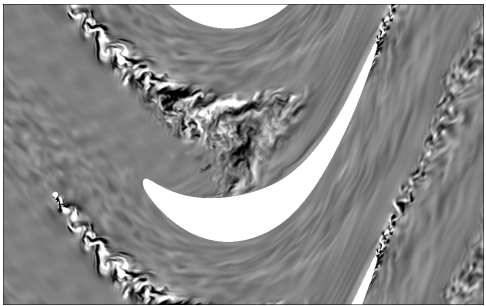
Conditional GAN (cGAN)



- Take control over the data production process by conditioning GAN framework
- Extension of loss function:

$$\mathcal{L}_{cond}(D,\phi) = \mathbb{E}_{\substack{X\sim\mu \\ \eta\sim\nu}}[\log(D(X|\eta))] + \mathbb{E}_{\substack{Z\sim\lambda \\ \eta\sim\nu}}[\log(1-D(\phi(Z|\eta)))] \tag{3}$$

LPT Stator - Conditional Training



LES image

Binary segmentation mask

High-Resolution Image Synthesis with Conditional

- Special form of cGAN - pix2pixHD (NVIDIA)
- Generation of **high-resolution photo-realistic** images by **conditioning** the input of the adversarial network on the corresponding **semantic label maps**
- Based on its former version pix2pix whose optimization problem is given as

$$\min_{\phi} \max_{D} \mathcal{L}_{cond}(D, \phi) \tag{4}$$

pix2pixHD Innovations

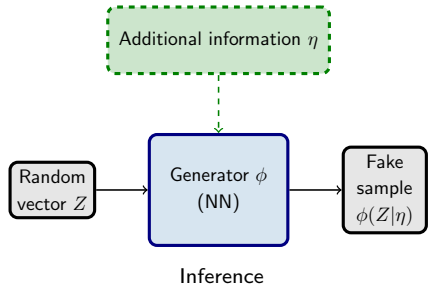
- Introduction of **three innovations** for **improvement** of **photorealism** and **resolution** of the synthesized images:
  1. Coarse-to-fine generator: Decomposition of the generator into two sub-networks $\Rightarrow \phi = \{\phi_1, \phi_2\}$
  2. Multi-scale discriminators: Three discriminators D_1, D_2 and D_3 with same architecture but operating on different scales $\Rightarrow$ Modification of loss function:

$$\min_{\phi} \max_{D_1, D_2, D_3} \sum_{i=1}^{3} \mathcal{L}_{cond}(\phi, D_i) + \mathcal{L}_{FM}(D_i, \phi) \tag{5}$$

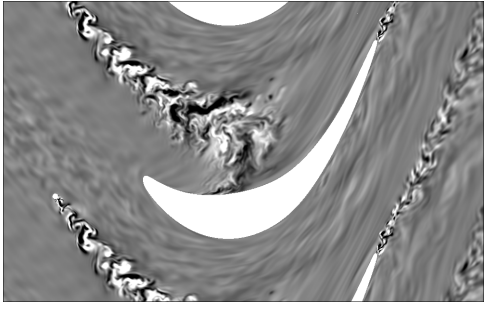
  3. Feature matching loss $\mathcal{L}_{FM}$: Stabilize training

LPT-Stator - Experiment Settings

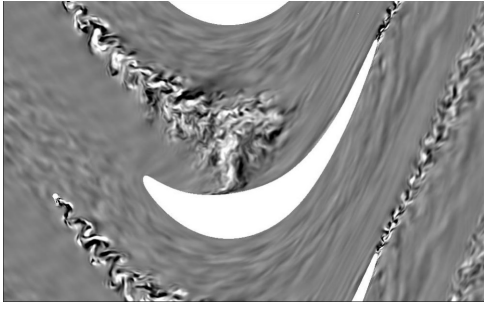
- Architecture: Adopted from original authors with small changes to avoid artifacts
- Epochs: 200
- Batch size: 10
- Input: $2,000$ LES images, noise vector and masks of size $k \times k' = 992 \times 624$
- Optimizer: Adam
- Learning rate: $2 \cdot 10^{-4}$



LPT Stator - Inference Results I

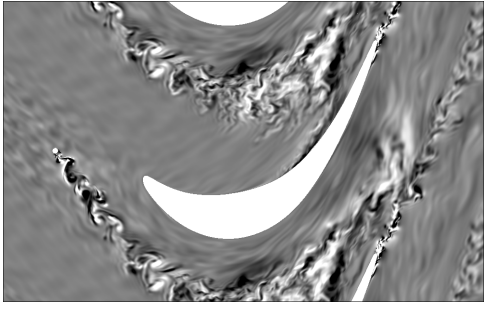


LES image

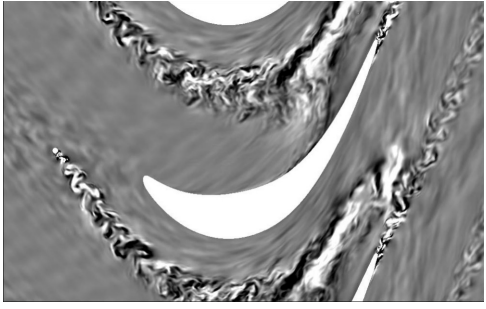


GAN image

LPT Stator - Results II

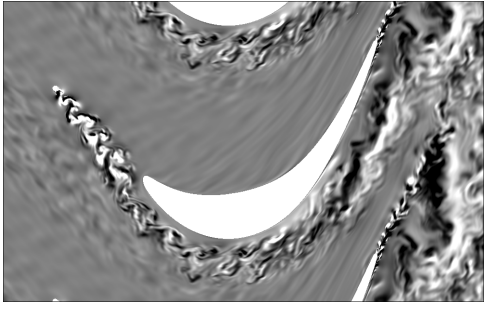


LES image

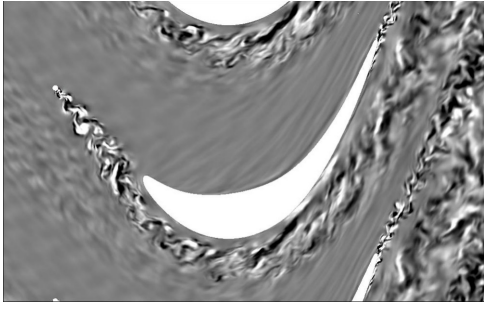


GAN image

LPT Stator - Results III

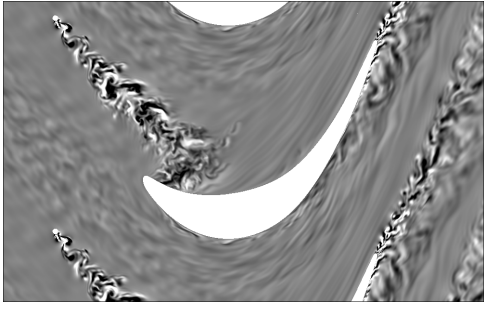


LES image

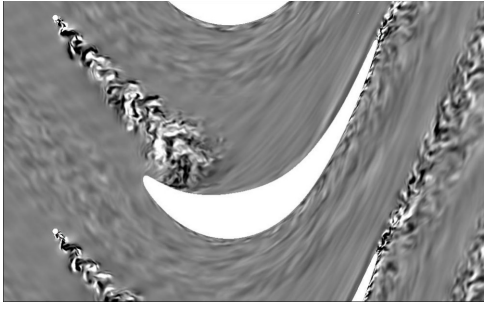


GAN image

LPT Stator - Results IV



LES image



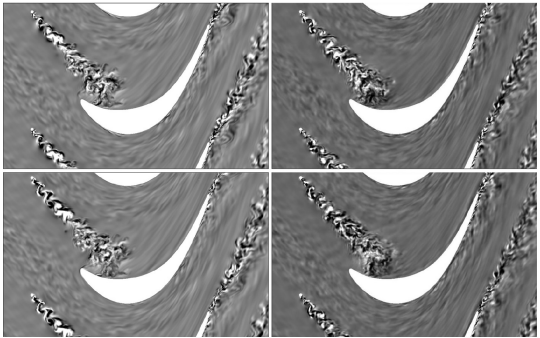
GAN image

Computational Costs

- **LES**: Performed on 560 CPU cores of the CPU type Intel Xeon SSkylake"Gold 6132
  - Karman vortex street ($5\ 000$ images): 72 core weeks $\hat{=}$ 21.6h
  - LPT Stator (2250 images): 640 core weeks $\hat{=}$ 192h
- **GAN-Training**: Performed on GPU of type Quadro RTX 8000 with 48 GB
  - Karman vortex street (DCGAN, $2\ 000$ epochs): 1.5 min/epoch $=$ 50h
  - LPT Stator (pix2pixHD, 200 epochs): 17 min/epoch $=$ 56h
- **GAN-Inference**: Performed on GPU of type Quadro RTX 8000 with 48 GB
  - Karman vortex street (DCGAN): 0.001 sec/image $\Rightarrow 5$ seconds
  - LPT Stator (pix2pixHD): 0.01 sec/image $\Rightarrow 22.5$ seconds

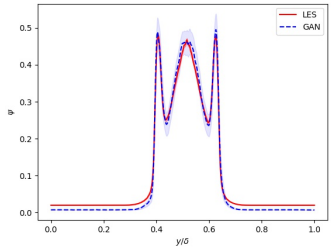
Sampling results for the LPT

Genaralization for Unseen Geometry Configurations



- Repeat experiment with 5% of the images ommited around specific wake position
- During inference, position the wake at exactly the middle of that position

Statistical Similarity vs Physics



- Measure strength of turbulence (variation of $c = |u|$ as a function of ξ

$$\mathrm{Var}[c(\xi)] = \lim_{T\to\infty} \frac{1}{T} \int_0^\infty |c(\xi, t) - \bar{c}(\xi)|^2 \, \mathrm{d}t$$

Estimation of Physical Quantities

- All statistical physical quantities of the turbulenmt flow can be expressen through evaluation of functionals $\Psi(x)$ and averaging

$$\bar{\Psi} = \mathbb{E}_{X\sim\mu}[\Psi(X)]$$

- We can assume that $\Psi(x)$ is bounded on Ω. Then estimate $\bar{\Psi}$ by $\hat{\Psi}$

$$\hat{\Psi} = \mathbb{E}_{Z\sim\nu}[\Psi(\hat{\phi}(Z))] = \lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} \Psi(\hat{\phi}(U_j)) \quad \text{a.s..} \quad \Rightarrow$$

$$|\bar{\Psi} - \hat{\Psi}| \leq C \sup_{\|\Psi'\|\leq 1} \left| \int \Psi' \, \mathrm{d}\mu - \int \Psi' \, \mathrm{d}\hat{\phi}_* \nu \right| = C \|\mu - \hat{\phi}_* \nu\|_{TV}.$$

Technische Universität Berlin

Convergence of Physical Quantities

- **Lemma (Drygala, Winhard, di Mare, G. 2021):**

$$|\bar{\Psi} - \hat{\Psi}| \to 0 \quad \text{as} \quad n \to \infty$$

- **Proof:** Enough to show: $\|\mu - \hat{\phi}_*\nu\|_{TV} \to 0$. $\hat{\mu} = \hat{\phi}_*\nu$.

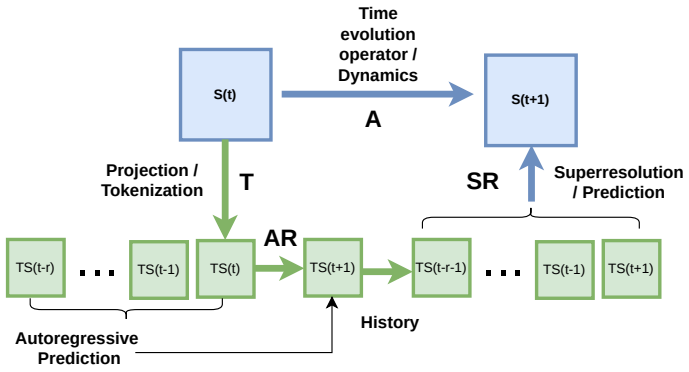
$$\begin{aligned}
\|\hat{\mu} - \mu\|_{\mathrm{TV}} &\leq \left\| \hat{\phi}_*\nu - \frac{\mu + \hat{\mu}}{2} \right\|_{\mathrm{TV}} + \left\| \mu - \frac{\mu + \hat{\mu}}{2} \right\|_{\mathrm{TV}} \\
&\leq \sqrt{2\mathfrak{d}_{\mathrm{KL}}\left(\hat{\mu} \middle\| \frac{\mu + \hat{\mu}}{2}\right)} + \sqrt{2\mathfrak{d}_{\mathrm{KL}}\left(\mu \middle\| \frac{\mu + \hat{\mu}}{2}\right)} \\
&\leq 2\sqrt{\mathfrak{d}_{\mathrm{KL}}\left(\hat{\mu} \middle\| \frac{\mu + \hat{\mu}}{2}\right) + \mathfrak{d}_{\mathrm{KL}}\left(\mu \middle\| \frac{\mu + \hat{\mu}}{2}\right)} = 2\sqrt{\mathfrak{d}_{\mathrm{JS}}(\hat{\mu}\|\mu)} \to 0.
\end{aligned}$$

Learning Dynamics

So far so good - but this is no flow

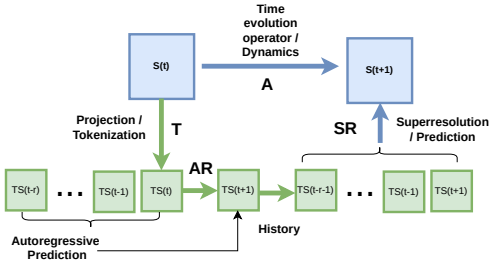
- Try to use **Word Models** for flow generation
- Can we find a mathematical justification?

Video GAN Operator Learning



Technische Universität Berlin

Observability of Dynamical Systems



- The autoregressive dynamic time step **AR** exists, if and only **SR** exists.
- In system theory, the existsnce of **SR** is known as **observability**!

$$\mathbf{SR}(TS, TA^{-1}S, TA^{-2}S, ..., TA^{-r}S) = S$$

When is a Linear Dynamical System Observable wrt a 'Tokenization'?

- If for every ξ be an eigen vector of the linear dynamic map A

$$T\xi \neq 0$$

  then the dynamical system is observable.
- **Example:** The discretized heat equation $A = e^{-t\Delta_\Gamma}$ is **not** observable wrt average pooling = projection to constant functions on pixel squares as 'tokenization'.

First Results on AR Operator Learning - Heat Equation

tokenized heat equation with random conductivity

$$\partial_t u(t,x) - \nabla \cdot a(x,\omega)\nabla u(t,x) = 0$$

First Results on AR Operator Learning - Heat Equation

reconstructed heat equation with random conductivity

$$\partial_t u(t,x) - \nabla \cdot a(x,\omega)\nabla u(t,x) = 0$$

Technische Universität Berlin

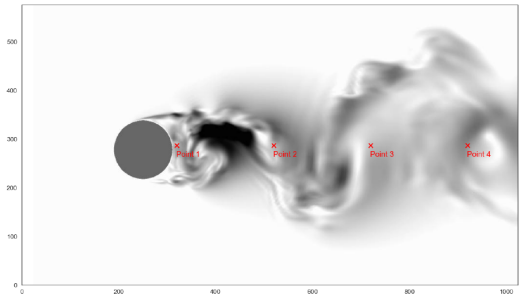
First Results on AR Operator Learning - Wave

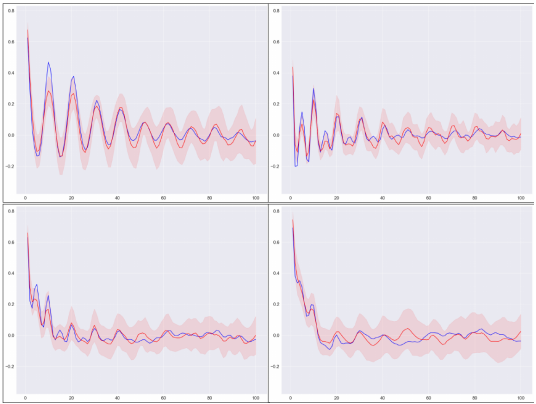
tokenized wave equation with random Laplacian

$$\partial_t^2 u(t,x) - \nabla \cdot a(x,\omega)\nabla u(t,x) = 0$$

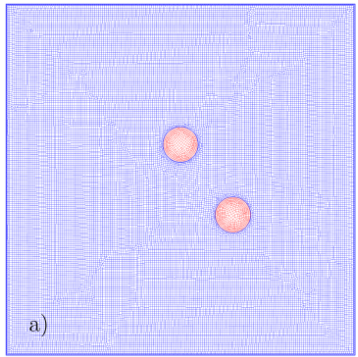
VideoGAN Results with NVIDIA LongVideoGAN

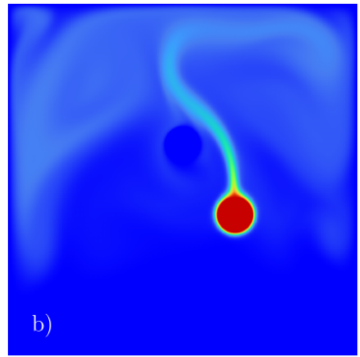
Physical Evaluation of Turbulence Autocorrelations



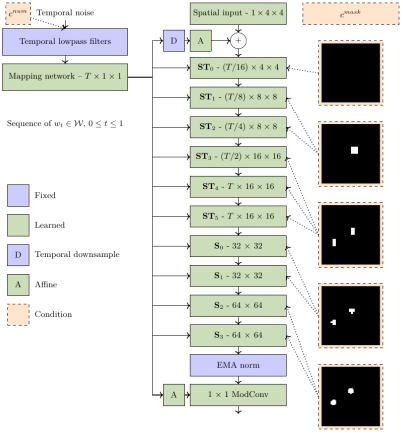


Parametric Operator Learning





Geometry Aware World Model Architecture



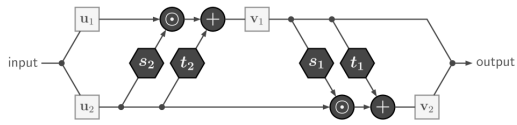
(a) Generator G^{lr} (b) Discriminator D^{lr}

Numerical Results

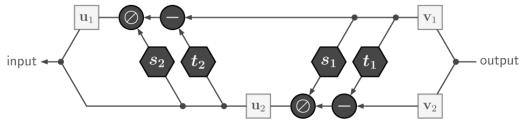
Invertible Neural Networks - INN

Forward Direction ():

Ardizonne et al., 2018

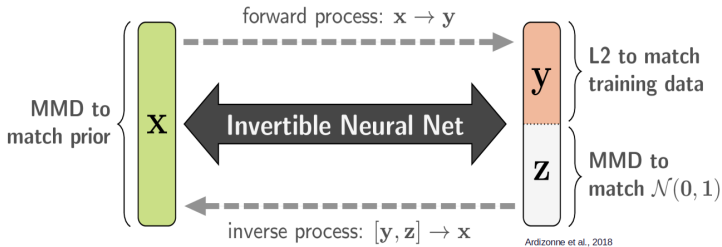


Backward Direction ():



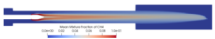
- Generative learning relates to inverse design (*Ardizzone et al. 2019*)

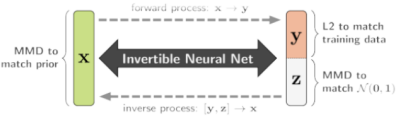
Dimension Matching



- In Design, usually there is a dimensioal mismatch between many input- and a few output parameters
- This can be used to generate design alternatives

Case study for a Multi Fuel Burner



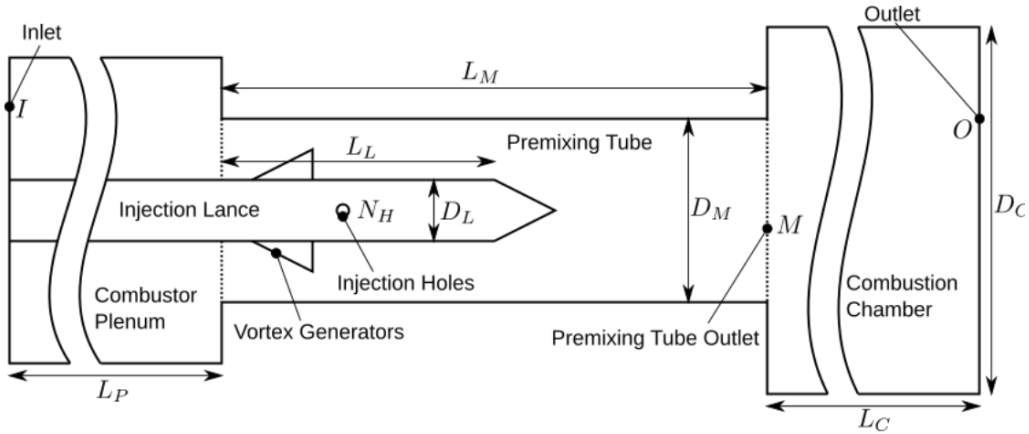


Desired performance values (labels)

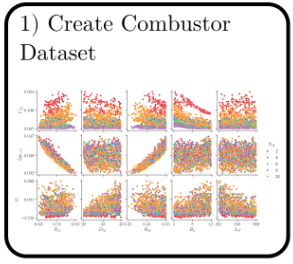
generative design tool

Multiple alternatives of geometry parameters yielding performance values

Design Parameters

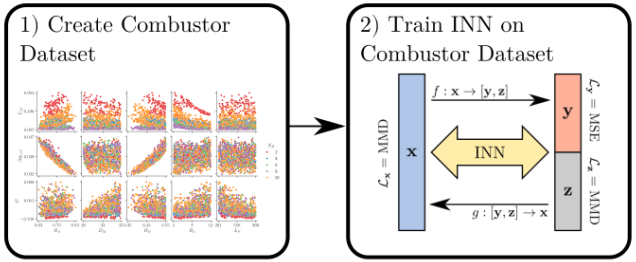


Workflow Overview



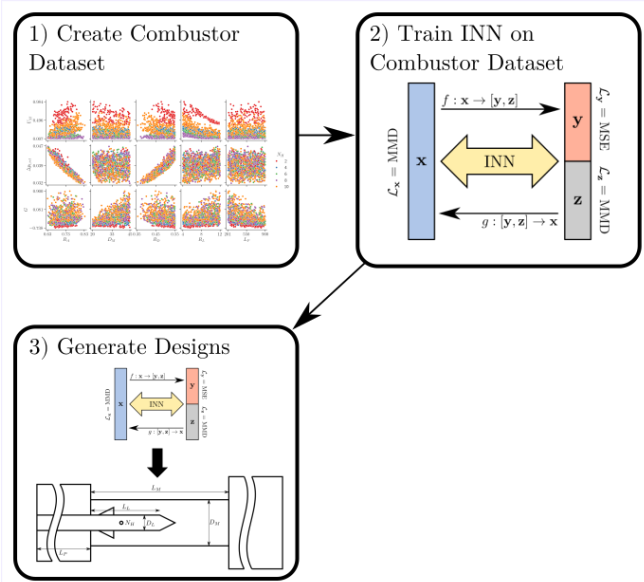
- Define combustor geometries by independent parameter vectors $\boldsymbol{x}$
- Define performance label vectors $\boldsymbol{y}$
- Sample designs and generate Labels via CFD and Acoustic Network Simulation

Workflow Overview



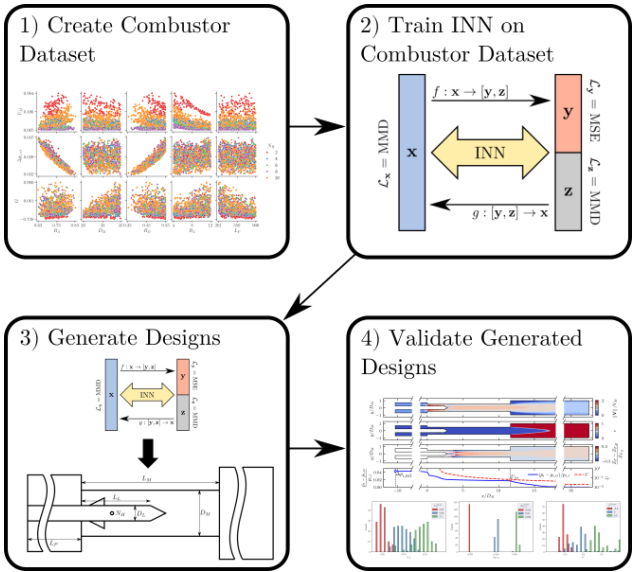
- INN Motivation & Concept
- Surrogate models for quick validation & data augmentation
- Ablation studies on surrogate model training and data augmentation size

Workflow Overview



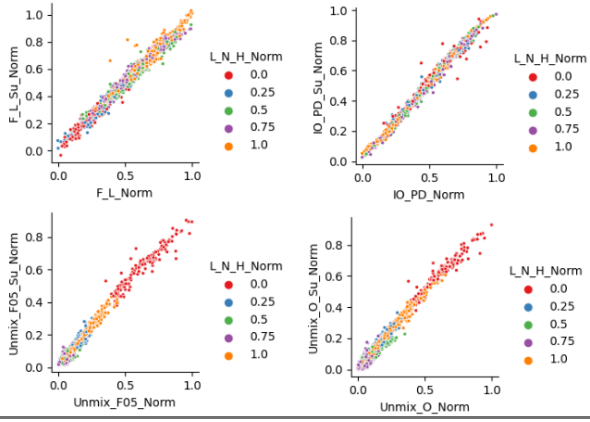
- Define targets for data generation
- Generate data
- Inspect distributions of generated data

Workflow Overview

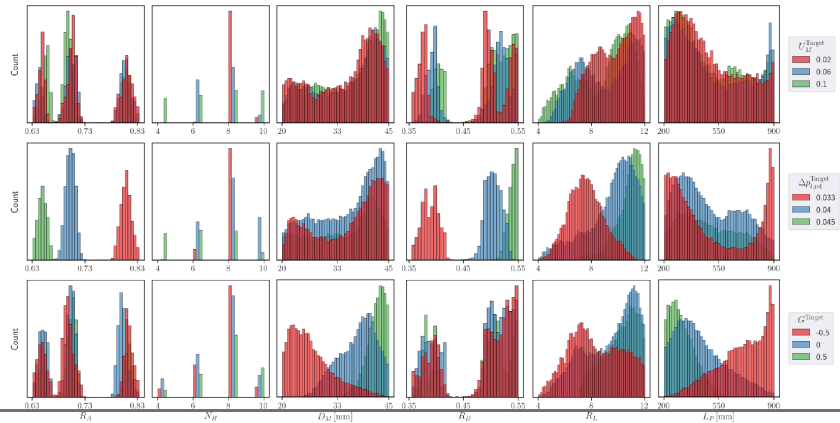


- Obtain true label values for generated designs
- Compare targets with obtained labels
- Outlook on label discretization

Trained INN - Forward Mode



Trained INN - Design Parameter Diversity in Backward Mode



The future I

The future II

Generative AI for numerical math – résumé

- Generative learning (small models) can be used for an **interactive** design process
- Generatives learning (large models) make simulations **real time capable**
- Generative AI is **no black box**, but one can and should understand the algorithms behind (and their properties)