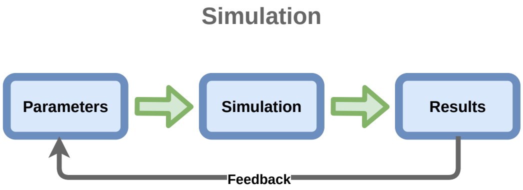
Invertible Neural Networks for Mechanical Engineering Problems

Hanno Gottschalk (Math+, TU Berlin)
joint with Patrick Krüger, Werner Krebs,
Bastian Werdelmann and Thanh Truong

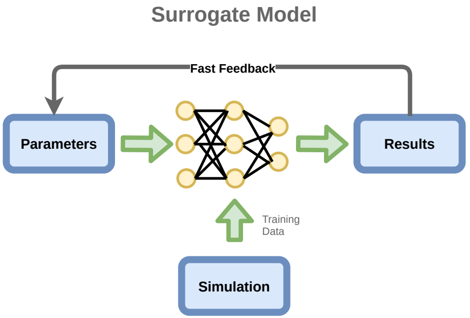
KoMSO Academy, November 15 2024

Progression of Machine Learning Methods in Engineering



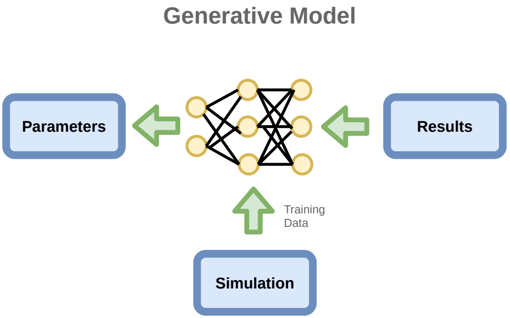
- Since ∼1980: Computer Simulation in Design

Progression of Machine Learning Methods in Engineering



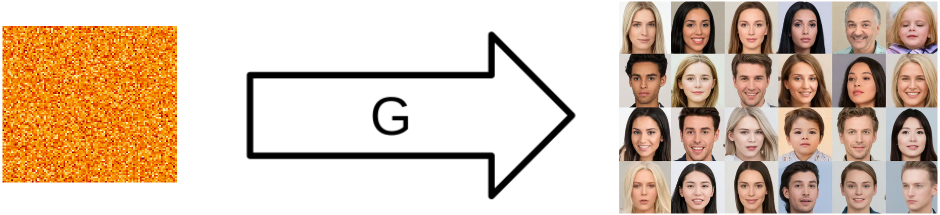
- Since $\sim$1980: Computer Simulation in Design
- Since $\sim$2000: Data driven models/surrogates

Progression of Machine Learning Methods in Engineering



- Since ~1980: Computer Simulation in Design
- Since ~2000: Data driven models/surrogates
- Since ~2020: Generative models/GAN & invertible networks

Generative Learning

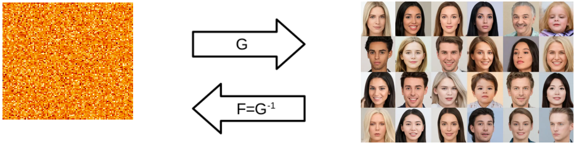


Types of generative models in CV

- DBM – Deep Boltzmann Machines
- VAE – Variational Autoencoders
- GAN - Generative Adversarial Networks
- SDM - Stable Diffusion Models
- INN - Invertible Neural Networks

Normalizing flows

- Let the noise space and the target space have the same dimension, $s = d$
- Let G be invertible, then $F = G^{-1}$ is the normalizing 'flow' direction



- Instead of G, learn the normalizing $F = G^{-1}$ with a neural network
- Invert only for generation

Transformation of probability densities

- INN are trained likelihood based – we thus have to compute the density
- The change of measure formula yields:

$$p_X(x) = p_Z(F(x)) \cdot \left|\det\left(\mathbb{J}F(x)\right)\right| = (2\pi)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}\|F(x)\|^2\right) \cdot \left|\det\left(\mathbb{J}F(x)\right)\right|$$

- Neg. Log-Likelihood based training – F_θ is a neural network that represents the 'normalizing flow'

$$-\log \mathcal{L}(\theta|\mathcal{D}) = \sum_{j=1}^{n} \left\{\frac{1}{2}\|F_\theta(X_j)\|^2 - \log\left|\det\left(\mathbb{J}F_\theta(X_j)\right)\right|\right\} + \text{cst} \longrightarrow \min$$

Requirements

- F_θ is invertible, $G_\theta = F_\theta^{-1}$ exists – not true in general for NN
- The likelihood has to be computationally tractable:

$$F_\theta = F_{\theta_L}^{(L)} \circ \cdots \circ F_{\theta_1}^{(1)} \quad \Rightarrow \quad \log\left|\det\left(\mathbb{J}F_\theta\right)\right| = \sum_{l=1}^{L} \log\left|\det\left(\mathbb{J}F_{\theta_l}^{(l)}\right)\right|$$

- $\Rightarrow$ Layers should be invertible and have a 'cheap' determinant.

Affine coupling layers

- Split the input data $y = \binom{y_1}{y_2}$ to layer l.
- Design an affine half layer with standard NN $a(y_1|\theta_l)$, $b(y_1|\theta_l)$

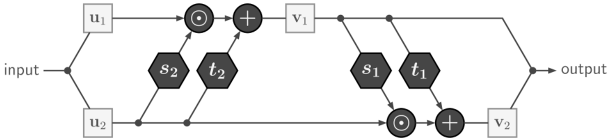
$$F^{(l)}_{\frac{1}{2},\theta_l} : y \mapsto \begin{pmatrix} y_1 \\ a(y_1|\theta_l)y_2 + b(y_1|\theta_l) \end{pmatrix} \quad \Rightarrow \quad |\mathbb{J}F^{(l)}_{\frac{1}{2},\theta_l}| = |a(y_1|\theta_l)|^{d/2}.$$

- $F^{(l),-1}_{\frac{1}{2},\theta_l} : y \mapsto \binom{y_1}{(y_2 - b(\theta_l))/a(y_1|\theta_l)}$ easy to compute!
- Then concatenate wit another affine half layer with $\binom{y_1}{y_2} \mapsto \binom{y_2}{y_1}$
- Universal approximation for generated distributions (Teshima et al, NeurIPS 2020)

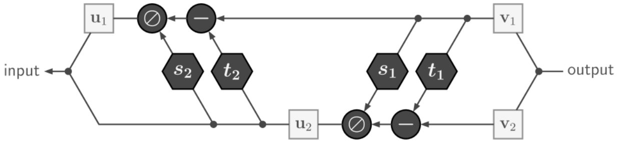
Invertible Neural Networks - INN

Forward Direction ():

Ardizonne et al., 2018

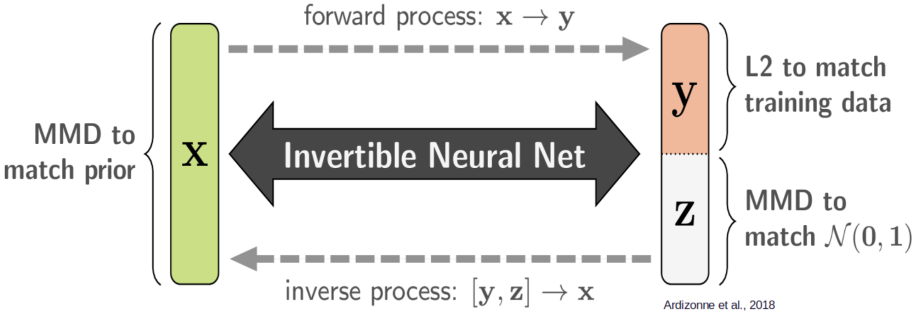


Backward Direction ():



- Generative learning relates to inverse design

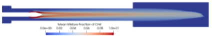
Dimension Matching

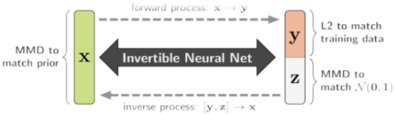


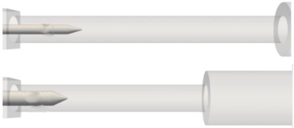
Ardizonne et al., 2018

- 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





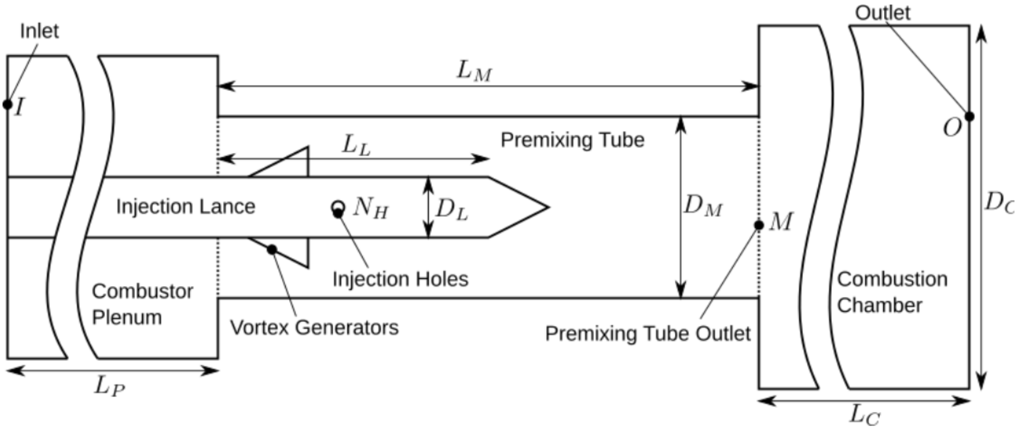


Desired performance values (labels)

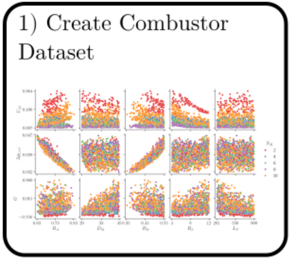
generative design tool

Multiple alternatives of geometry parameters yielding performance values

Design Parameters

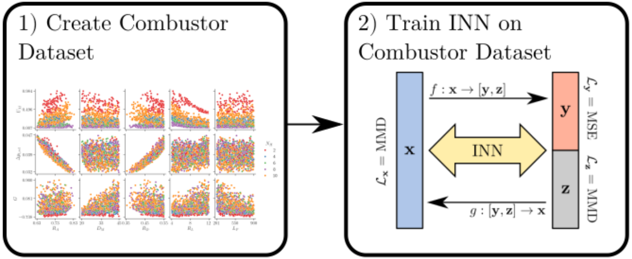


Workflow Overview



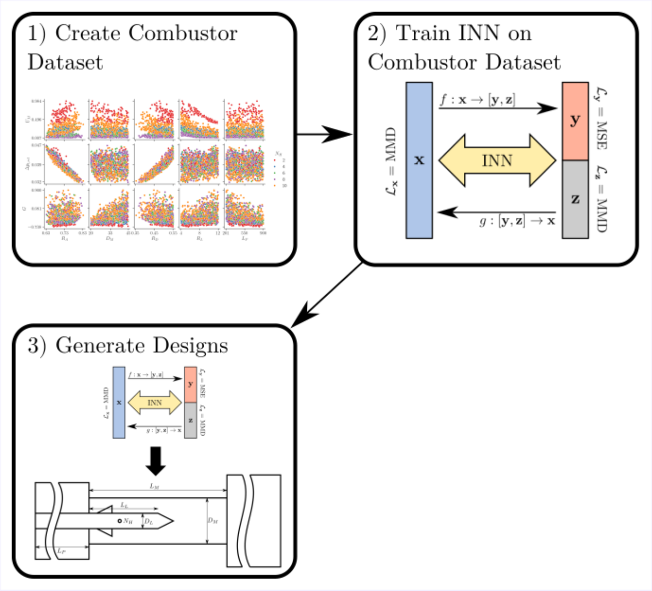
- Define combustor geometries by independent parameter vectors $\boldsymbol{x}$
- Define performance label vectors $\boldsymbol{y}$
- Sample designs and generate Labels via CFD and Acoustic Network Simulation

Workflow Overview



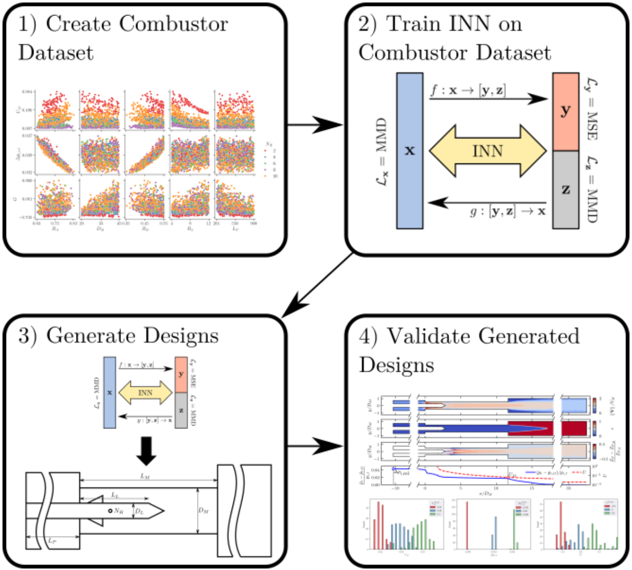
- INN Motivation & Concept
- Surrogate models for quick validation & data augmentation
- Ablation studies on surrogate model training and data augmentation size

Workflow Overview



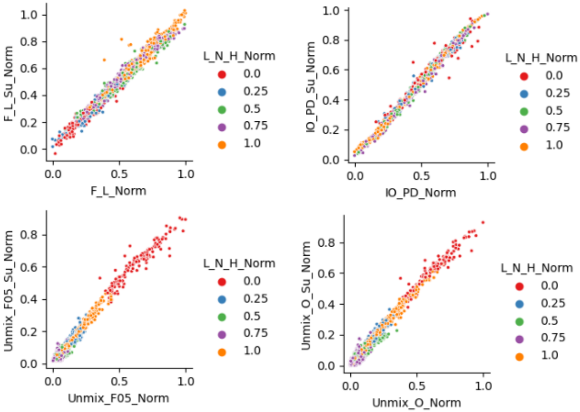
- Define targets for data generation
- Generate data
- Inspect distributions of generated data

Workflow Overview



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

Trained INN - Forward Mode

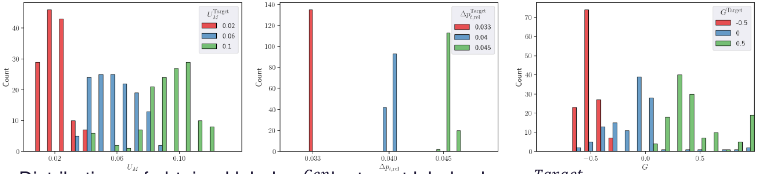


Trained INN - Backward Mode

- Designs were prevalidated using surrogate models
- 20% best designs were kept
- Selected 15 designs for each $y_i^{Target}, i = 1, .., 27$
- Obtain true labels y^{Gen} by original CFD & ANS Workflow

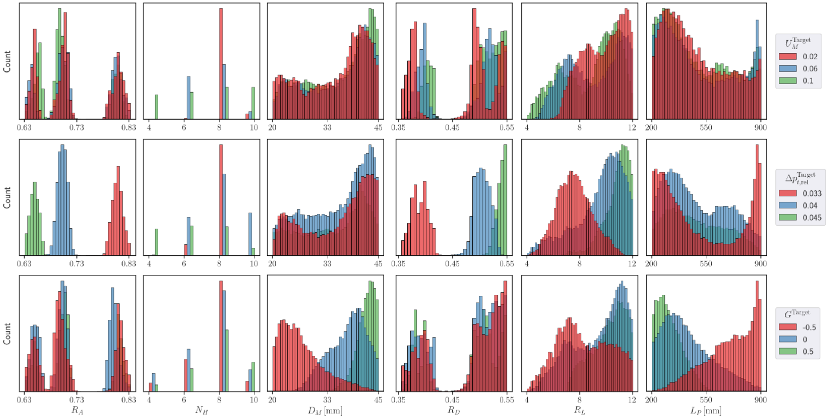
y	y^{Target}	$\mu(y^{\text{Gen}})$	$\sigma(y^{\text{Gen}})$
U_M	0.02	0.0222	0.0078
	0.06	0.0572	0.0131
	0.1	0.0916	0.0165
$\Delta p_{t,\text{rel}}$	0.033	0.03306	0.00014
	0.04	0.04011	0.00015
	0.045	0.0453	0.00034
G	-0.5	-0.4938	0.0826
	0	-0.0959	0.2618
	0.5	0.4415	0.2765

- $\Delta p_{t,rel}$ performs best due to strong correlations in Data
- U_M has uncertainty in both directions
- Highest uncertainty: G
- G^{Target}=-0.5 yields stable designs only
- G^{Target}=0.5 yields unstable designs only



Distributions of obtained labels y^{Gen} by target label values y^{Target} (Left: U_M^{Target} **0.02**/**0.06**/**0.1**, Center: $\Delta p_{t,rel}^{Target}$ **0.033**/**0.04**/**0.045**, Right: G^{Target} **-0.5**/**0**/**0.5**

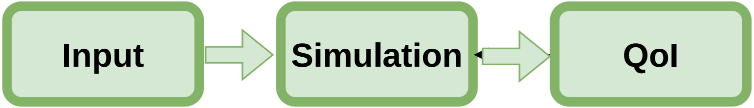
Trained INN - Design Parameter Diversity



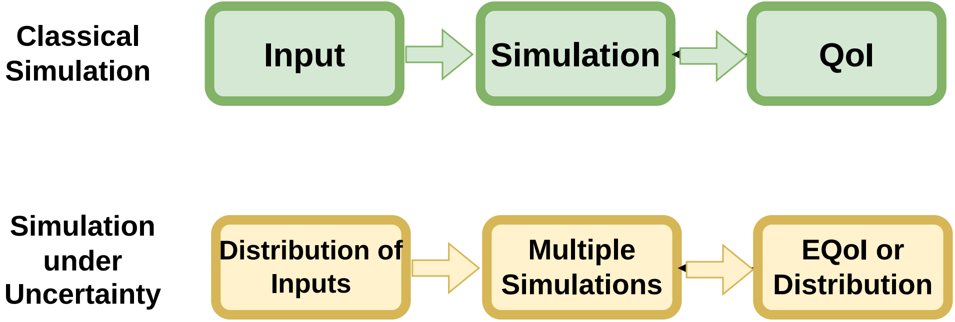
MATH+

Uncertainty Quantification

Classical Simulation



Uncertainty Quantification



Uncertainty for the Poisson Equation

- The linear stationary diffusion problem on bounded domain $D \subseteq \mathbb{R}^d$

$$-\nabla \cdot (a\nabla u) = f, \quad u|_{\partial D_d} = 0, \qquad n \cdot \nabla u = g_n|_{\partial D_n}.$$

Uncertainty for the Poisson Equation

- The linear stationary diffusion problem on bounded domain $D \subseteq \mathbb{R}^d$

$$-\nabla \cdot (a\nabla u) = f, \quad u|_{\partial D_d} = 0, \qquad n \cdot \nabla u = g_n|_{\partial D_n}.$$

- $a(x,\omega) = T(Z(x,\omega)) > 0$ random coeff., eg $T(z) = \exp(z)$.

Uncertainty for the Poisson Equation

- The linear stationary diffusion problem on bounded domain $D \subseteq \mathbb{R}^d$

$$-\nabla \cdot (a \nabla u) = f, \quad u|_{\partial D_d} = 0, \qquad n \cdot \nabla u = g_n|_{\partial D_n}.$$

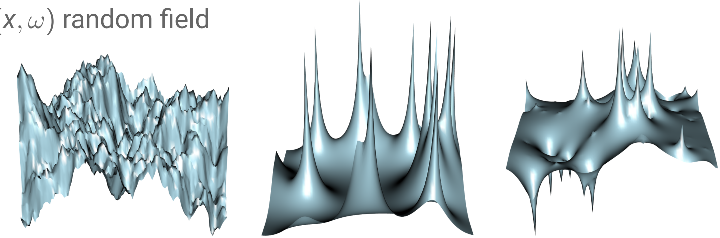
- $a(x, \omega) = T(Z(x, \omega)) > 0$ random coeff., eg $T(z) = \exp(z)$.
- $Z(x) = Z(x, \omega)$ random field

Uncertainty for the Poisson Equation

- The linear stationary diffusion problem on bounded domain $D \subseteq \mathbb{R}^d$

$$-\nabla \cdot (a\nabla u) = f, \quad u|_{\partial D_d} = 0, \qquad n \cdot \nabla u = g_n|_{\partial D_n}.$$

- $a(x,\omega) = T(Z(x,\omega)) > 0$ random coeff., eg $T(z) = \exp(z)$.
- $Z(x) = Z(x,\omega)$ random field



Physical Interpretation of the Poisson Equation

- $u(x)$ is ground pressure at location $x \in D$

Physical Interpretation of the Poisson Equation

- $u(x)$ is ground pressure at location $x \in D$
- $a(x)$ is conductivity of ground material at location x

Physical Interpretation of the Poisson Equation

- $u(x)$ is ground pressure at location $x \in D$
- $a(x)$ is conductivity of ground material at location x
- Thus, the flow velocity $q(x)$ at location x is

$$q(x) = -a(x)\nabla u(x).$$

Physical Interpretation of the Poisson Equation

- $u(x)$ is ground pressure at location $x \in D$
- $a(x)$ is conductivity of ground material at location x
- Thus, the flow velocity $q(x)$ at location x is

$$q(x) = -a(x)\nabla u(x).$$

- QoI is e.g. first exit time from restricted region $D' \subseteq D$ around pollution location $x_0 \in D'$

$$\dot{X}(t) = q(X(t)), \quad X(0) = x_0, \quad \mathrm{QoI} = \inf\{t > 0 : X(t) \notin D'\}.$$

Physical Interpretation of the Poisson Equation

- $u(x)$ is ground pressure at location $x \in D$
- $a(x)$ is conductivity of ground material at location x
- Thus, the flow velocity $q(x)$ at location x is

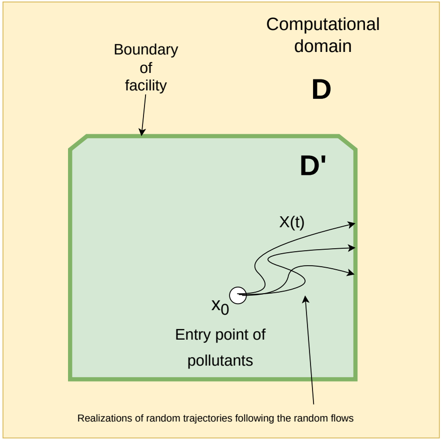
$$q(x) = -a(x)\nabla u(x).$$

- QoI is e.g. first exit time from restricted region $D' \subseteq D$ around pollution location $x_0 \in D'$

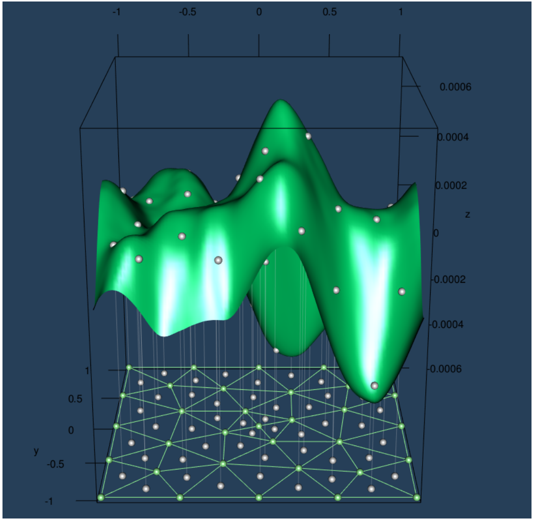
$$\dot{X}(t) = q(X(t)), \quad X(0) = x_0, \quad \mathrm{QoI} = \inf\{t > 0 : X(t) \notin D'\}.$$

- As $q(x)$ is random through $a(x)$, also the QoI becomes a random variable!

Flow Modeling for a Waste Deposit Facility



Numerical Simulation



Numerical Integration of Gaussian Distributions

- Let $\mu = N(0,1)$ standard normal distribution,

$$f_\mu(\theta) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\theta^2}.$$

- $H_n(\theta)$ n-the order Hermite polynomial, $\theta_1, \dots, \theta_n$ the zeropoints of H_n. Then,

$$\int_{\mathbb{R}} \mathrm{QoI}(\theta) f_\mu(\theta)\,\mathrm{d}\theta \approx \sum_{j=1}^{n} \omega_j \mathrm{QoI}(\theta_j).$$

- ω_j are called quadrature weights and θ_j quadrature points.
- Choice of ω_j make approximation exact for polynomials of deg. $\leq n-1$.

Tensor and Sparse Grid Quadratures

- $\mu = N(0, \mathbb{1})$, d-dimensional Normal distribution, $f_\mu(\theta) = \frac{1}{\sqrt{2\pi}^d} e^{-\frac{1}{2}\theta^\top\theta}$.
- We search for a quadrature which is exact on polynomials of degree $n-1$

$$p(\theta) = \sum_{|\alpha| \leq n-1} a_\alpha \theta^\alpha, \quad \theta^\alpha = \prod_{j=1}^{d} \theta_j^{\alpha_j}, \quad |\alpha| = \max\{\alpha_j\} \text{ or } |\alpha| = \sum_{j=1}^{d} \alpha_j.$$

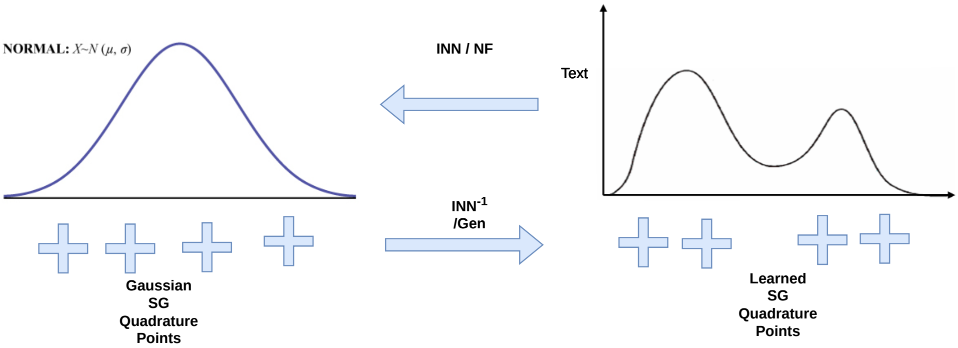
- The linear system of equations requires $q(d,n) \sim n^d$ quadrature points for max degree and $q(d,n) \sim n^d/d!$ for the sum degree.

$$\int_{\mathbb{R}^d} p(\theta) f_\mu(\theta)\,\mathrm{d}\theta = \sum_{j=1}^{q(d,n)} \omega_j p(\theta_j), \quad \int_{\mathbb{R}^d} \mathrm{QoI}(\theta) f_\mu(\theta)\,\mathrm{d}\theta \approx \sum_{j=1}^{q(d,n)} \omega_j \mathrm{QoI}(\theta_j)$$

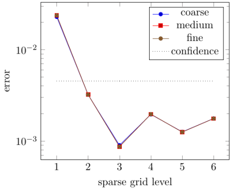
Non Gaussian Distributions, Problem Statement

- **Q1:** What, if $f_\mu(\theta)$ is an involved distribution where quadrature weights ar not known?
- **Q2:** What, if $f_\mu(\theta)$ is even unknown and only i.i.d. samples $\Theta_l \sim f_\mu(\theta)\mathrm{d}\theta$ are known, $l = 1, \ldots, N$?

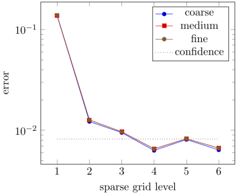
Learning to Integrate



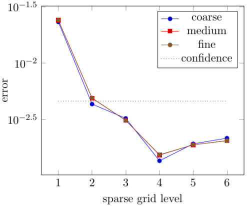
Learning to Integrate



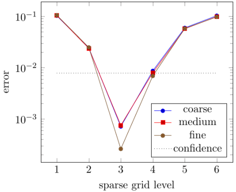
Gaussian



Gamma

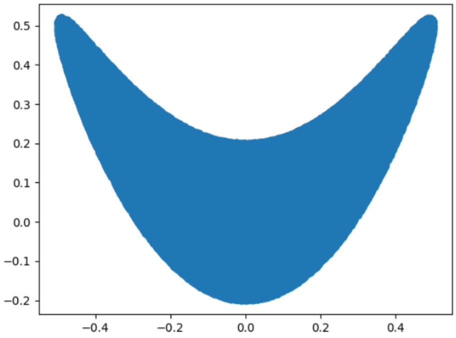


Bigamma

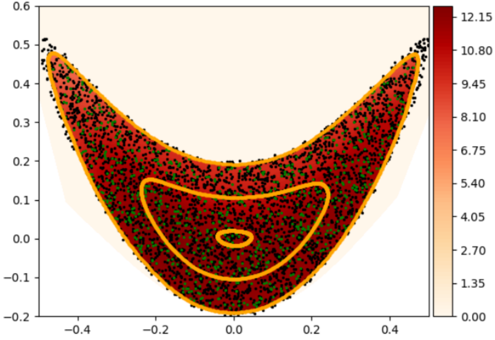


Compound Poisson

INN for Shape Morphing

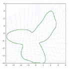


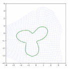
(a) The banana target shape

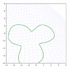


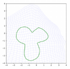
(b) Transformed uniform circle by our INN

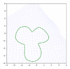
Solving the Isoperimetric Problem with INN

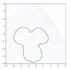

(a) Epoch 1

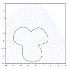

(b) Epoch 4

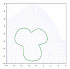

(c) Epoch 7

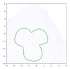

(d) Epoch 10

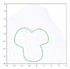

(e) Epoch 13

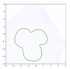

(f) Epoch 16

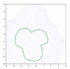

(g) Epoch 19

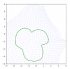

(h) Epoch 22

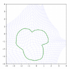

(i) Epoch 25

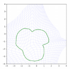

(j) Epoch 28

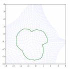

(k) Epoch 31

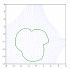

(l) Epoch 36

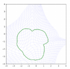

(m) Epoch 41

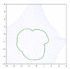

(n) Epoch 46

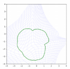

(o) Epoch 51

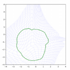

(p) Epoch 56

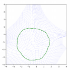

(q) Epoch 61

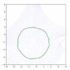

(r) Epoch 66

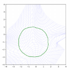

(s) Epoch 71

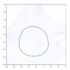

(t) Epoch 76


(u) Epoch 100


(v) Epoch 200


(w) Epoch 300


(x) Epoch 400


(y) Epoch 500

Conclusion

Conclusion

- INN are a versatile tool to solve Engineering problems in ...
- Inverse design
- Uncertainty quantification
- Shape optimization

Outlook

- There are various forms of INN ... which is best for what?
- How reliable are INN based methods?