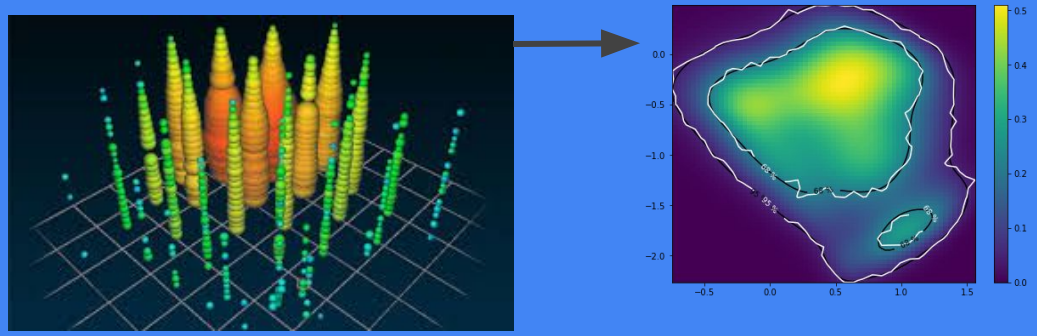
Friedrich-Alexander-Universität Erlangen-Nürnberg

Normalizing flows

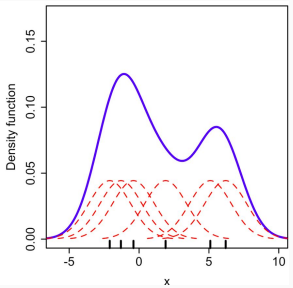
Differentiable Expectation values and more



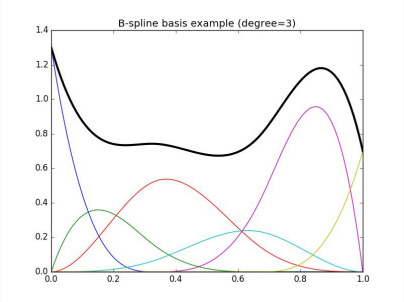
Thorsten Glüsenkamp, Oct. 11 2022, CERN IML WG meeting

Why normalizing flows?

PDF modelling:



Mixture models /
Kernel Density Functions

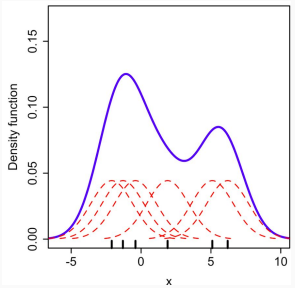


B-Splines

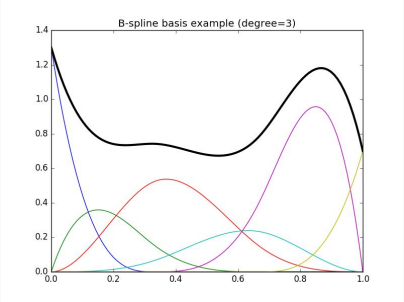
- (arbitrarily) Complex PDF shape
- Evaluate probability analytically
- Hard in high-D
- …

Why normalizing flows?

PDF modelling:



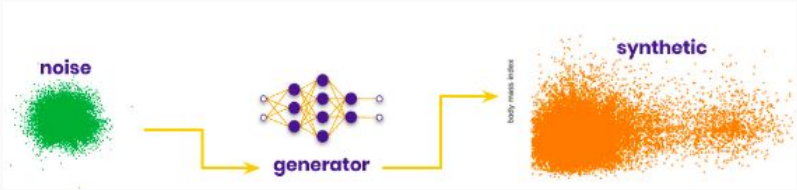
Mixture models /
Kernel Density Functions



B-Splines

- (arbitrarily) Complex PDF shape
- Evaluate probability analytically
- Hard in high-D
- …

Generative Models



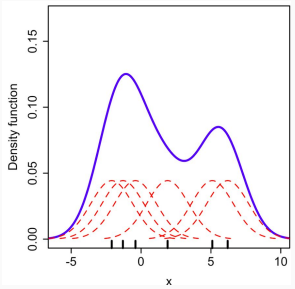
[statice.ai]

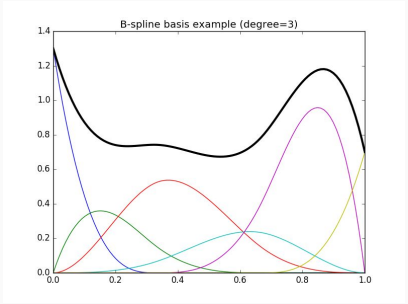
E.g. GANs

- Complex data
- No PDF evaluation
- Easy in high-D
- …

Why normalizing flows?

PDF modelling:

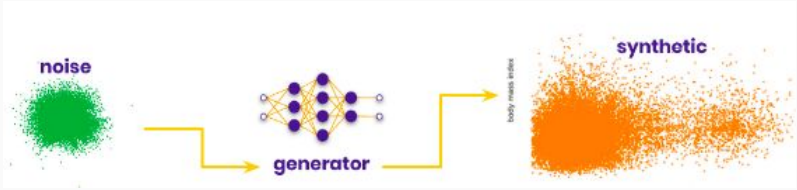




Mixture models /
Kernel Density Functions

B-Splines

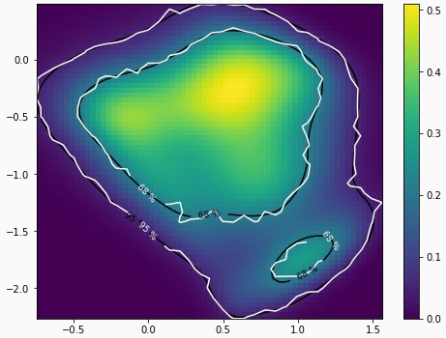
Generative Models



[statice.ai]

E.g. GANs

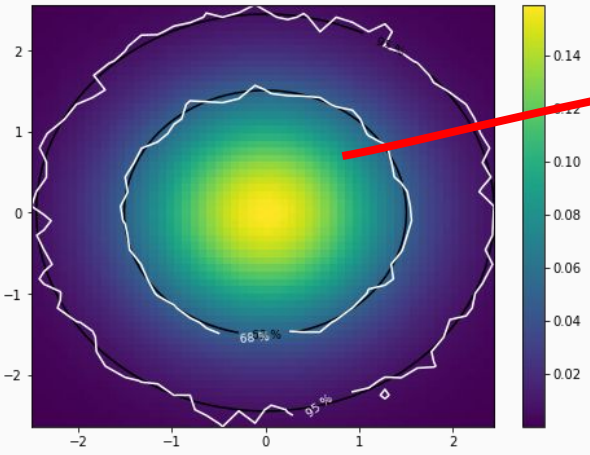
Normalizing flows: PDFs and generative models simultaneously



- (arbitrarily) Complex PDF shape
- Evaluate probability analytically
- Works in D >> 1
- Can be interpreted as generalizations of the Gaussian distribution
- Works on manifolds
- Generate differentiable samples (-> differentiable expectation values)
- …

Definition

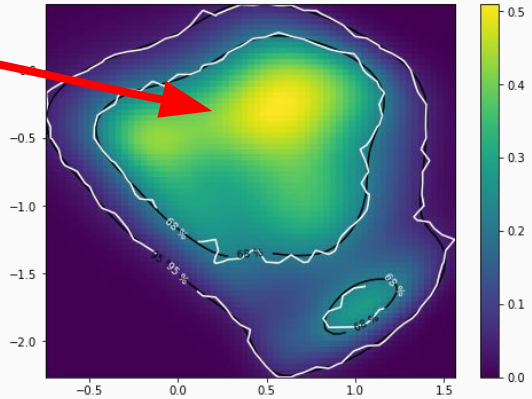
Base $p_o(\vec{z})$



$f(\vec{z})$

Flow defining function

Target $p(\vec{x})$



Definition

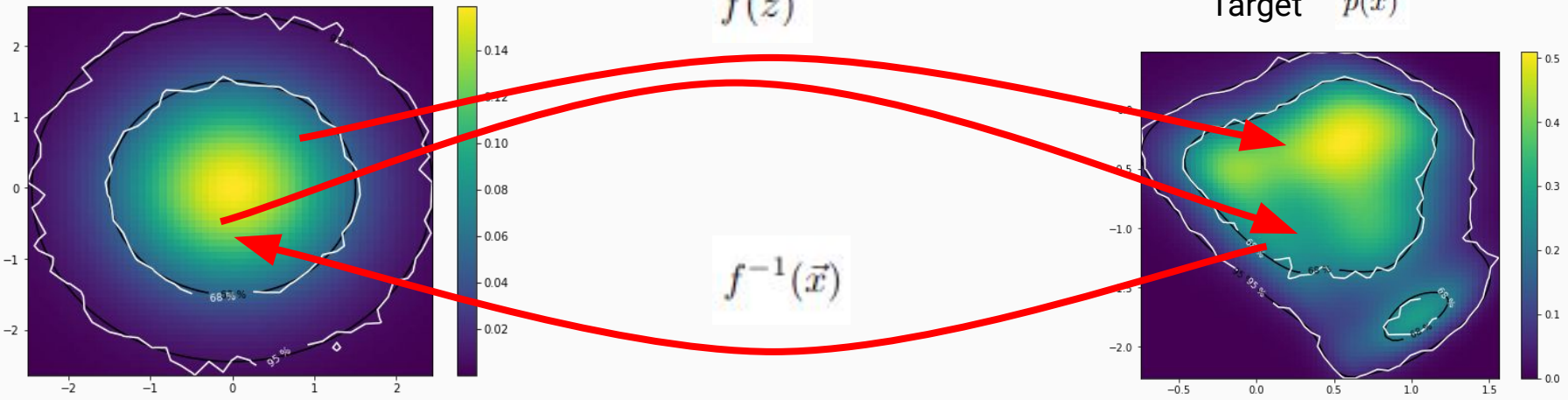
$$p(\vec{x}) = p_0(f^{-1}(\vec{x})) \cdot \left| \det \frac{\partial f^{-1}(\vec{x})}{\partial \vec{x}} \right|$$

Base $p_o(\vec{z})$

$f(\vec{z})$

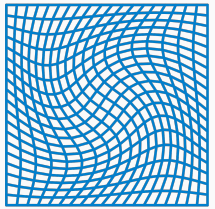
$f^{-1}(\vec{x})$

Target $p(\vec{x})$



1) The function has to be **bijective** for the inverse!
2) The function has to be **differentiable!**

= Diffeomorphism



Definition

$$p(\vec{x}) = p_0(f^{-1}(\vec{x})) \cdot \left|\det J^{-1}(\vec{x})\right|$$

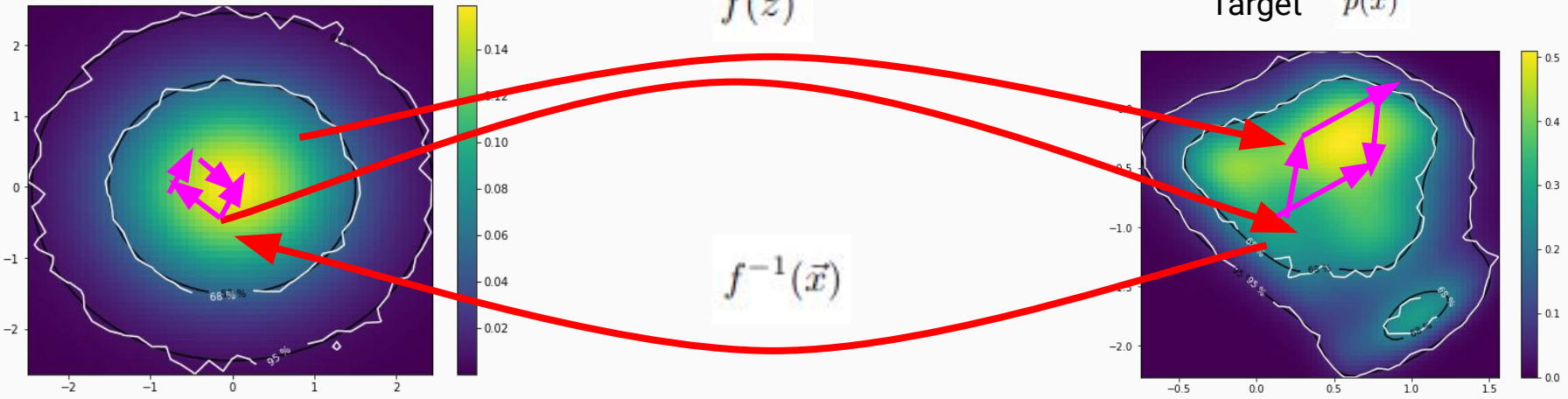
Log probability includes
Local “stretching”/”squeezing”
Of volume element

Base $p_o(\vec{z})$

$f(\vec{z})$

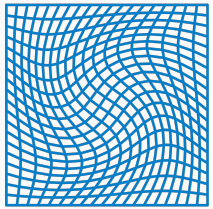
$f^{-1}(\vec{x})$

Target $p(\vec{x})$



1) The function has to be **bijective** for the inverse!
2) The function has to be **differentiable!**

= Diffeomorphism



Definition

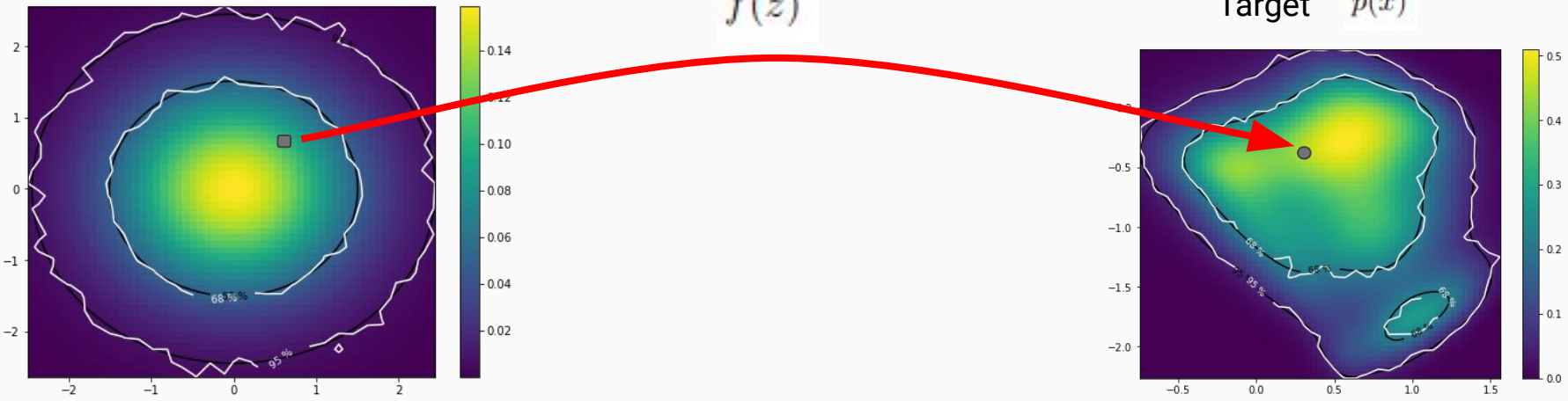
Sampling is also straight forward: $\vec{x} = f(\vec{z})$

Must be able to sample from base distribution!

Base $p_o(\vec{z})$

$f(\vec{z})$

Target $p(\vec{x})$



Definition

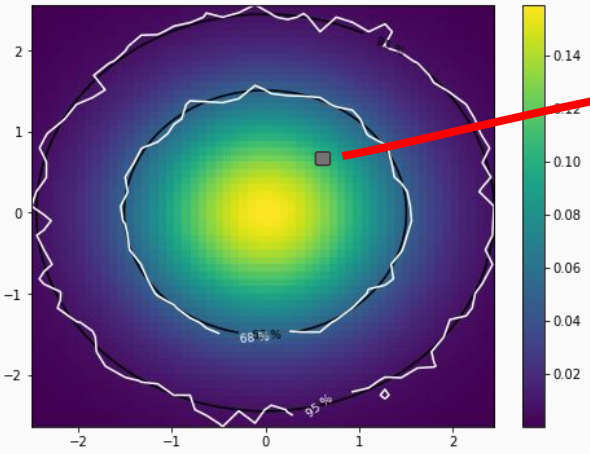
Sampling is also straight forward:

$$\vec{x_\theta} = f_\theta(\vec{z})$$

Must be able to sample from base distribution!

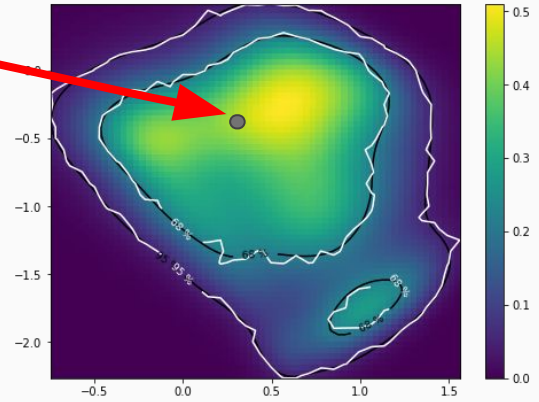
$$p_\theta(\vec{x}) = p_0(f_\theta^{-1}(\vec{x})) \cdot \left| \det \frac{\partial f_\theta^{-1}(\vec{x})}{\partial \vec{x}} \right|$$

Base $p_o(\vec{z})$



$f_\theta(\vec{z})$

Target $p_\theta(\vec{x})$



Definition

Sampling is also straight forward:

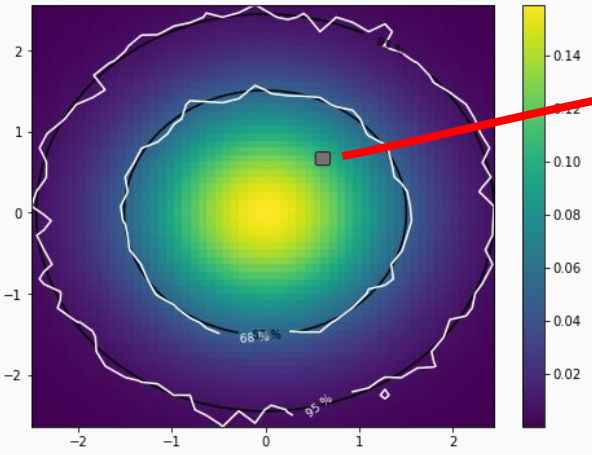
Must be able to sample from base distribution!

$$\vec{x_\theta} = f_\theta(\vec{z})$$

1) Differentiable prob.

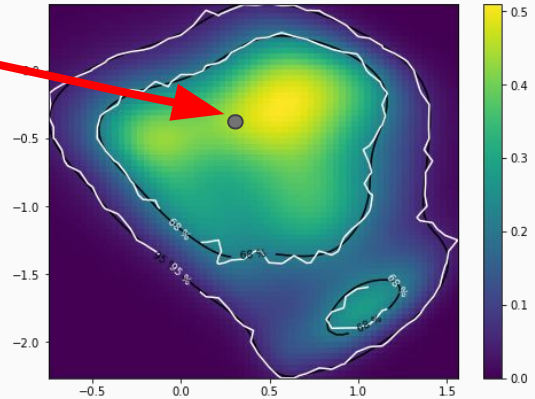
$$p_\theta(\vec{x}) = p_0(f_\theta^{-1}(\vec{x})) \cdot \left|\det \frac{\partial f_\theta^{-1}(\vec{x})}{\partial \vec{x}}\right|$$

Base $p_o(\vec{z})$



$f_\theta(\vec{z})$

Target $p_\theta(\vec{x})$



2) Differentiable samples!

Definition

Sampling is also straight forward:

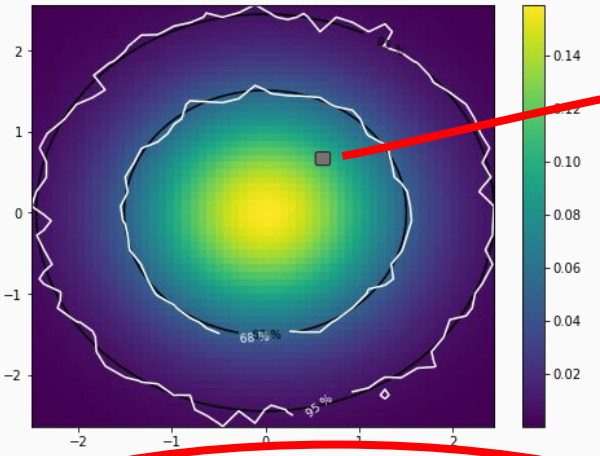
Must be able to sample from base distribution!

$$\vec{x_\theta} = f_\theta(\vec{z})$$

1) Differentiable prob.

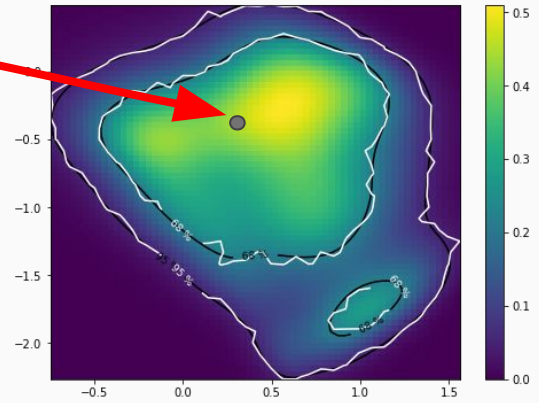
$$p_\theta(x) = p_0(f_\theta^{-1}(\vec{x})) \cdot \left| \det \frac{\partial f_\theta^{-1}(\vec{x})}{\partial \vec{x}} \right|$$

Base $p_o(\vec{z})$



$f_\theta(\vec{z})$

Target $p_\theta(\vec{x})$



2) Differentiable samples!

The simplest possible normalizing flow (1-d)

Let us try to see the distribution of the simplest possible normalizing flow in 1 dimension!

$$x = f(\vec{z}) =$$

The simplest possible normalizing flow (1-d)

Let us try to see the distribution of the simplest possible normalizing flow in 1 dimension!

$$x = f(\vec{z}) = a \cdot z + b \qquad \theta = \{a, b\}$$

$$p_\theta(\vec{x}) = p_0(f_\theta^{-1}(\vec{x})) \cdot \left| \det \frac{\partial f_\theta^{-1}(\vec{x})}{\partial \vec{x}} \right|$$

The simplest possible normalizing flow (1-d)

Let us try to see the distribution of the simplest possible normalizing flow in 1 dimension!

$$x = f(\vec{z}) = a \cdot z + b$$

Use Standard normal base distribution (we will always use the standard normal for convenience)

Base distribution

$$p_0(z) = \frac{1}{\sqrt{2 \cdot \pi}} \cdot \exp(-0.5 \cdot z^2)$$

Change of variables formula

$$p_\theta(\vec{x}) = p_0(f_\theta^{-1}(\vec{x})) \cdot \left| \det \frac{\partial f_\theta^{-1}(\vec{x})}{\partial \vec{x}} \right|$$

Inv. Flow function

$$z = f^{-1}(\vec{x}) = \frac{x - b}{a}$$

$$p(x) = \frac{1}{\sqrt{2 \cdot \pi \cdot a^2}} \cdot \exp\left(-0.5 \cdot \left(\frac{x - b}{a}\right)^2\right)$$

The simplest possible normalizing flow (1-d)

Let us try to see the distribution of the simplest possible normalizing flow in 1 dimension!

$$x = f(\vec{z}) = a \cdot z + b$$

Use Standard normal base distribution (we will always use the standard normal for convenience)

"Linear flow" = "general Gaussian distribution" in 1d

Base distribution

$$p_0(z) = \frac{1}{\sqrt{2 \cdot \pi}} \cdot \exp(-0.5 \cdot z^2)$$

Change of variables formula

$$p_\theta(\vec{x}) = p_0(f_\theta^{-1}(\vec{x})) \cdot \left| \det \frac{\partial f_\theta^{-1}(\vec{x})}{\partial \vec{x}} \right|$$

Inv. Flow function

$$z = f^{-1}(\vec{x}) = \frac{x - b}{a}$$

$$p(x) = \frac{1}{\sqrt{2 \cdot \pi \cdot a^2}} \cdot \exp\left(-0.5 \cdot \left(\frac{x - b}{a}\right)^2\right)$$

The simplest possible normalizing flow (1-d)

Let us try to see the distribution of the simplest possible normalizing flow in 1 dimension!

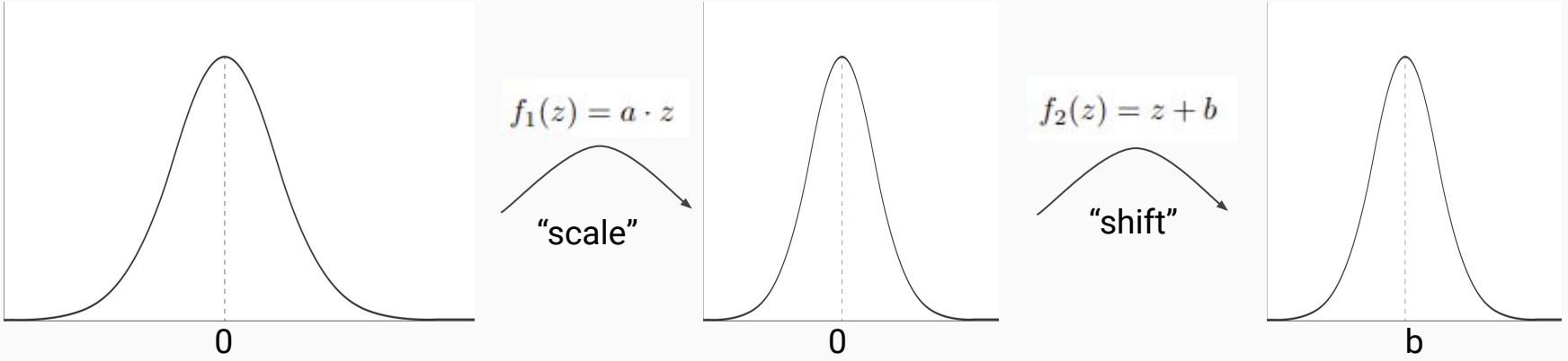
$$x = f(\vec{z}) = a \cdot z + b$$

Use Standard normal base distribution (we will always use the standard normal for convenience)

"Linear flow" = "general Gaussian distribution" in 1d

Technically, this is 2-step flow:

$$f(z) = f_2(f_1(z))$$



The simplest possible normalizing flow (n-d)

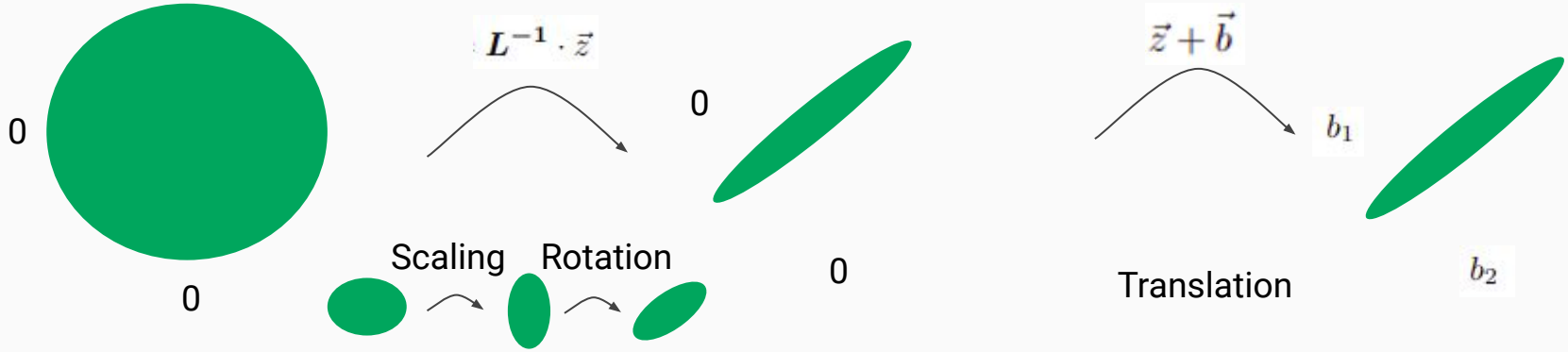
Let us try to see the distribution of the simplest possible normalizing flow in n dimensions!

$$\vec{x} = f(\vec{z}) = \boldsymbol{L}^{-1} \cdot \vec{z} + \vec{b}$$

Use Standard normal base distribution (we will always use the standard normal for convenience)

$$\boldsymbol{C}^{-1} = \boldsymbol{L}^{T} \cdot \boldsymbol{L}$$

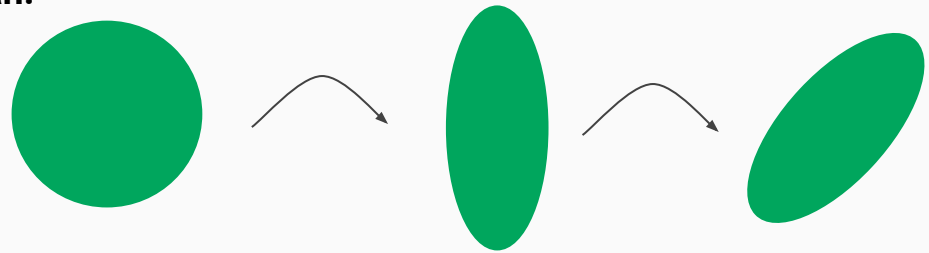
"Affine flow" = "general Gaussian distribution" in n-d



A nice generalization of the Multivariate Gaussian

4) Gaussianization Flow(arXiv:2003.01941)

Gaussian:



Gaussianization flow generalized by non-linear scalings instead of “linear scalings”

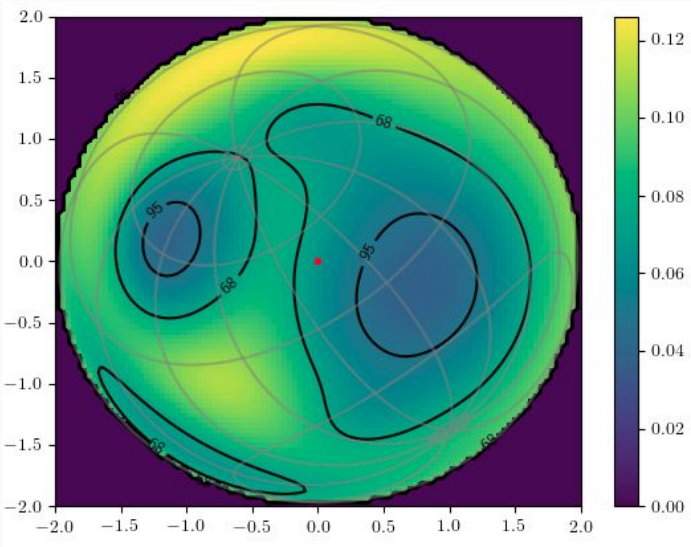
$$T_{\boldsymbol{\theta}}(\mathbf{x}) = \Psi_{\boldsymbol{\theta}_L} \circ R_L \circ \Psi_{\boldsymbol{\theta}_{L-1}} \circ \cdots \circ \Psi_{\boldsymbol{\theta}_1} \circ R_1 \mathbf{x}$$

Step by step refinement of the PDF - provably approximates any distribution

Other variations

“Manifold” normalizing flows

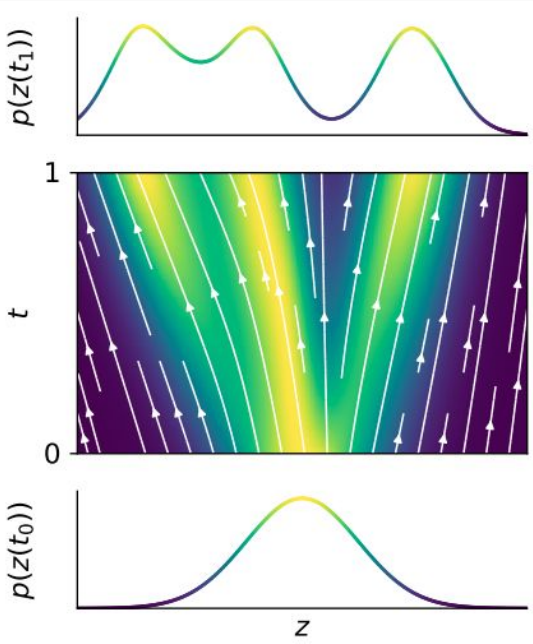
$$p(T(x)) = \frac{\pi(x)}{\sqrt{\det(E^\top J^\top J E)}}$$



(2002.02428)

“Continuous” normalizing flows

$$\log p(\mathbf{z}(t_1)) = \log p(\mathbf{z}(t_0)) - \int_{t_0}^{t_1} \mathrm{Tr}\left(\frac{\partial f}{\partial \mathbf{z}(t)}\right) dt.$$



(FFJORD, 1810.01367)

Example: more expressive label predictions (e.g. For MODE applications)

Typically, label space in regression is low dimensional ….

MSE-loss
(-log-prob)

Std normal

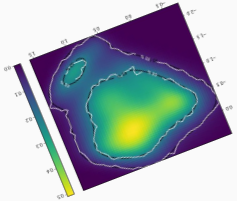
(+ log_sigma predictions)
(-log-prob)

Scaled normal

-log-prob

MVN

-log-prob



Arbitrary shape

Example: more expressive label predictions (e.g. For MODE applications)

Typically, label space in regression is low dimensional ….

MSE-loss
(-log-prob)

Std normal

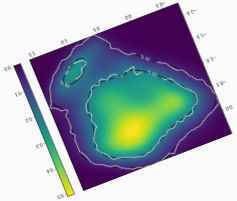
(+ log_sigma predictions)
(-log-prob)

Scaled normal

-log-prob

MVN

-log-prob



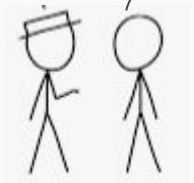
Arbitrary shape

Regression always expressible as normalizing flow training

More expressive power

“Adding more labels does not help .. results even get worse.
I just train separate models for each label.”

“With (autoregressive) normalizing flows the opposite should happen… adding more labels helps training and performance.”



Differentiable expectation values

$$D_{\mathrm{KL,joint(x,z_o)}}(\mathcal{P}_t; q_\phi) = \int_x \int_{z_o} \mathcal{P}_t(z_o, x) \cdot \ln \frac{\mathcal{P}_t(z_o, x)}{q_\phi(z_o, x)} dz_o dx$$

Differentiable expectation values

$$D_{\mathrm{KL,joint(x,z_o)}}(\mathcal{P}_t; q_\phi) = \int_x \int_{z_o} \mathcal{P}_t(z_o, x) \cdot \ln \frac{\mathcal{P}_t(z_o, x)}{q_\phi(z_o, x)} dz_o dx$$

$$\begin{aligned}
\arg\min_\phi \hat{D}_{\mathrm{KL,joint(x,z_o)}}(\mathcal{P}_t; q_\phi) &= \arg\min_\phi \frac{1}{N} \sum_{x_i, z_{o,i}} \ln\left(\frac{\mathcal{P}_t(z_{o,i}; x_i)}{q_\phi(z_{o,i}; x_i)}\right) + \ln\left(\frac{\mathcal{P}_t(x_i)}{q(x_i)}\right) \\
&= \arg\min_\phi \frac{1}{N} \sum_{x_i, z_{o,i}} -\ln\left(q_\phi(z_{o,i}; x_i)\right) + \mathrm{const} \\
&= \arg\min_\phi \frac{1}{N} \sum_{x_i, z_{o,i}} -\ln\left(q_\phi(z_{o,i}; x_i)\right) \\
&\equiv \arg\min_\phi \mathcal{L}_{\mathrm{supervised}}(\phi)
\end{aligned}$$

Differentiable expectation values

$$D_{\mathrm{KL,joint(x,z_o)}}(\mathcal{P}_t; q_\phi) = \int_x \int_{z_o} \mathcal{P}_t(z_o, x) \cdot \ln \frac{\mathcal{P}_t(z_o, x)}{q_\phi(z_o, x)} dz_o dx$$

Measure independent of parameters

$$\begin{aligned}
\arg\min_\phi \hat{D}_{\mathrm{KL,joint(x,z_o)}}(\mathcal{P}_t; q_\phi) &= \arg\min_\phi \frac{1}{N} \sum_{x_i, z_{o,i}} \ln\left(\frac{\mathcal{P}_t(z_{o,i}; x_i)}{q_\phi(z_{o,i}; x_i)}\right) + \ln\left(\frac{\mathcal{P}_t(x_i)}{q(x_i)}\right) \\
&= \arg\min_\phi \frac{1}{N} \sum_{x_i, z_{o,i}} -\ln\left(q_\phi(z_{o,i}; x_i)\right) + \mathrm{const} \\
&= \arg\min_\phi \frac{1}{N} \sum_{x_i, z_{o,i}} -\ln\left(q_\phi(z_{o,i}; x_i)\right) \\
&\equiv \arg\min_\phi \mathcal{L}_{\mathrm{supervised}}(\phi)
\end{aligned}$$

(Gradients straightforward -
We use it all the time)

Differentiable expectation values

$$I_\theta = \int p_\theta(x) F_\theta(x) dx \qquad ?$$

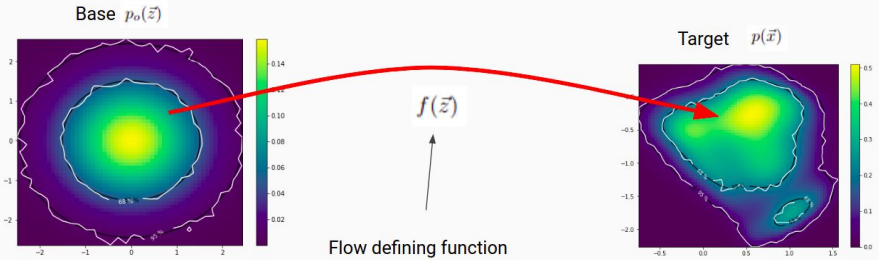
Differentiable expectation values

$$I_\theta = \int p_\theta(x) F_\theta(x) dx_\theta = \int p(z) F_\theta(f_\theta(z)) dz$$

$$\vec{x_\theta} = f_\theta(\vec{z})$$

Reparametrization trick -> calculate integral in base space (equivalent to target integral)

$$p_\theta(\vec{x}) = p_0(f_\theta^{-1}(\vec{x})) \cdot \left| \det \frac{\partial f_\theta^{-1}(\vec{x})}{\partial \vec{x}} \right|$$



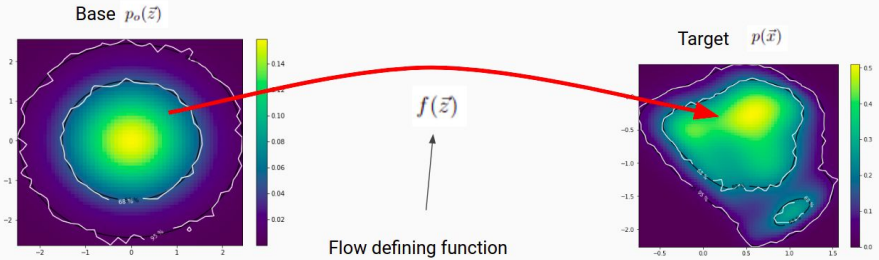
$$dx_\theta = dz \left| \det \frac{\partial f_\theta(\vec{z})}{\partial \vec{z}} \right|$$

Differentiable expectation values

$$I_\theta = \int p_\theta(x) F_\theta(x) dx_\theta = \int p(z) F_\theta(f_\theta(z)) dz$$

$$\vec{x_\theta} = f_\theta(\vec{z})$$

Reparametrization trick -> calculate integral in base space (equivalent to target integral)



$$p_\theta(\vec{x}) = p_0(f_\theta^{-1}(\vec{x})) \cdot \left| \det \frac{\partial f_\theta^{-1}(\vec{x})}{\partial \vec{x}} \right|$$

$$dx_\theta = dz \left| \det \frac{\partial f_\theta(\vec{z})}{\partial \vec{z}} \right|$$

$$\approx \frac{1}{N} \cdot \sum_i^N F_\theta(x_\theta)$$

Differentiable expectation values

$$I_\theta = \int p_\theta(x) F_\theta(x) dx \approx \frac{1}{N} \cdot \sum_i^N F_\theta(x_\theta)$$

Examples:

n-th moment of p

$$\int p_\theta(x) x^n dx \approx \frac{1}{N} \cdot \sum_i^N x_\theta^n$$

Differentiable expectation values

$$I_\theta = \int p_\theta(x) F_\theta(x) dx \approx \frac{1}{N} \cdot \sum_i^N F_\theta(x_\theta)$$

Examples:

n-th moment of p

$$\int p_\theta(x) x^n dx \approx \frac{1}{N} \cdot \sum_i^N x_\theta^n$$

entropy

$$-\int p_\theta(x) \log\left(p_\theta(x)\right) dx \approx \frac{1}{N} \cdot \sum_i^N -\log p_\theta(x_\theta)$$

Wrong gradient!

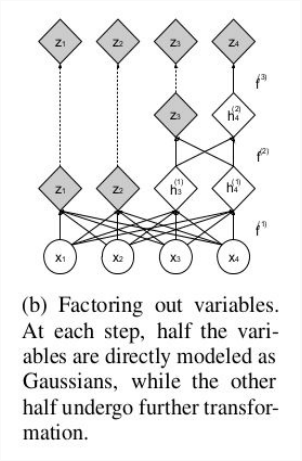
$$\frac{1}{N} \cdot \sum_i^N -\log p_\theta(x)$$

…. Many other integrals in information theory
…. Required for VAE to work etc

We need normalizing flows for our specific needs…

We need normalizing flows for our specific needs…

-> **Need specific implementations**
We don't really care for PDFs over 10k-dim images (what CS people are into)..



(b) Factoring out variables. At each step, half the variables are directly modeled as Gaussians, while the other half undergo further transformation.

(real-NVP, 2017)

… But rather conditional PDFs over directions, energies, positions (maybe in total only a few 10s of dimension max) - and **correct joint correlation structure** over these PDFs!

Jammy_flows (conditional + non-conditional “autoregressive” PDFs)

https://github.com/thoglu/jammy_flows

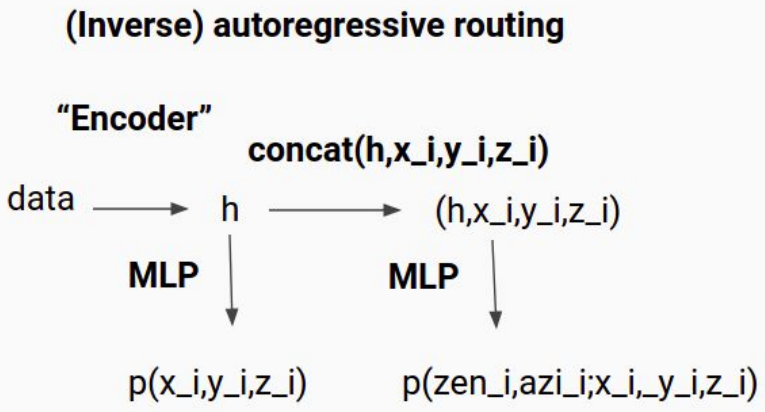
“A generic pdf on a tensor product of manifolds”

E.g. a 5-d conditional PDF: p(x,y,z,zenith,azimuth ; data)

(generalizes notion of of “Inverse Autoregressive Flows” (Kingma et. al 2016))

```
import jammy_flows

pdf=jammy_flows.pdf("e3+s2", "gggg+n")
```



Jammy_flows (conditional + non-conditional “autoregressive” PDFs)

https://github.com/thoglu/jammy_flows

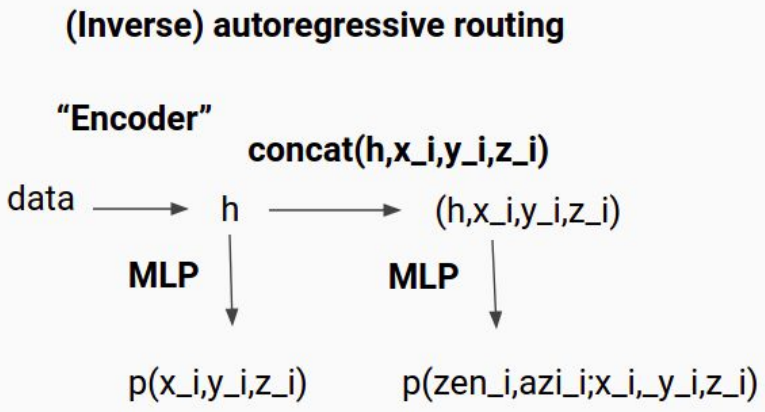
“A generic pdf on a tensor product of manifolds”

E.g. a 5-d conditional PDF: p(x,y,z,zenith,azimuth ; data)

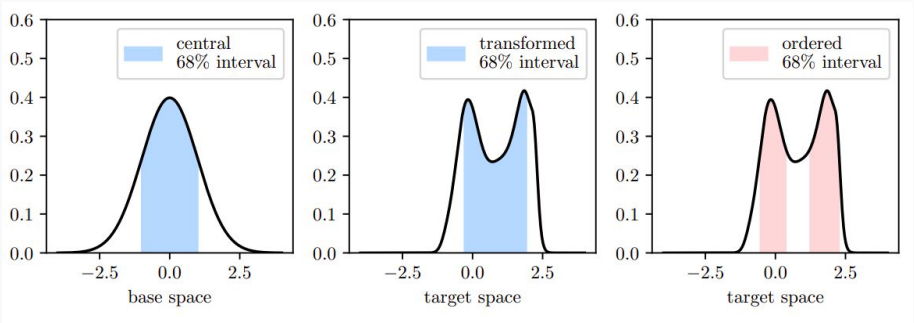
(generalizes notion of of “Inverse Autoregressive Flows” (Kingma et. al 2016))

```
import jammy_flows

pdf=jammy_flows.pdf("e3+s2", "gggg+n")
```



Gaussian base allows for coverage control:



jammy_flows

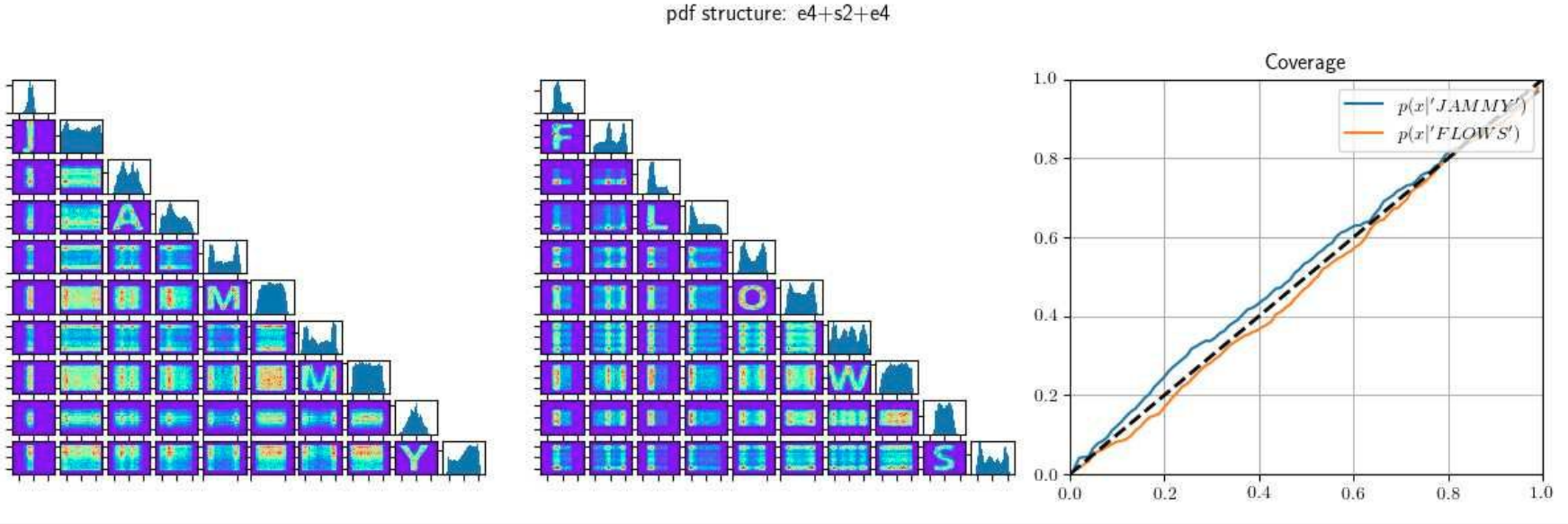
https://github.com/thoglu/jammy_flows

"A generic pdf on a tensor product of manifolds"

import jammy_flows

pdf=jammy_flows.pdf("e4+s2+e4", "gggg+n+gggg")

Coverage for arbitrary distributions and other things .. (https://arxiv.org/abs/2008.05825)



Summary

- Normalizing flows are powerful PDF approximators
  (log probability + generative model in one)

- Allow to obtain correct gradients on sample-based (information-theoretic) expectation values (makes VAEs work)

- Also allow for coverage control, Goodness of Fit, … (see 2008.05825 for more details)

- A NF tool taylored for low-dimensional joint PDFs over different manifolds (e.g. direction + energy) as generalization of “inverse autoregressive flows”

https://github.com/thoglu/jammy_flows