Funnels

Samuel Klein, Johnny Raine, Sebastian Pina-Otey, Slava Voloshynovskiy, Tobias Golling

University of Geneva

IML

Flows

Given a parameterised invertible function $f_\theta : X \to Z$

and a known density $p(z)$

we can construct the following distribution

$$p_\theta(x) = p(z)|\det(J_x^{f_\theta})|$$

Flows

We can **match** the distribution of our flow to data by **maximizing** the **likelihood** of the data distribution.

This is equivalent to minimizing the KL divergence between the data distribution and the distribution of the flow.

$$\max_{\theta} \mathbb{E}_{p_D(x)} \left[p_\theta(x) \right], \quad x \sim p_D(x)$$

Flows

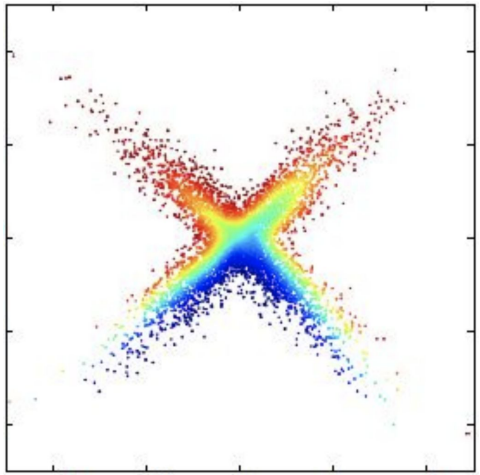
In practise the invertible function is parameterised by a neural network, and we only have samples from the data distribution

So, we estimate the expectation with

$$\max_{\theta} \sum_{i=1}^{N} p_\theta(x_i), \quad x_i \sim p_D(x)$$

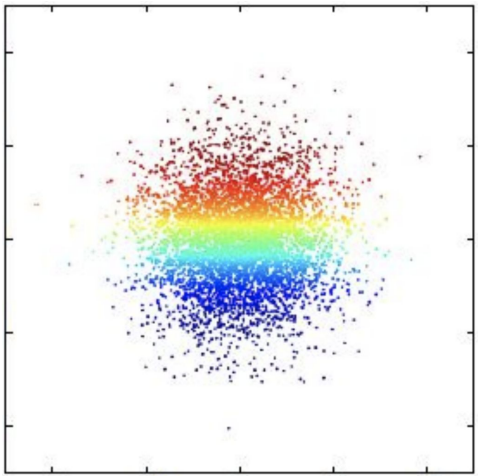
Flows

A flow can be seen as a map between two distributions



$p_\theta(x)$

$f_\theta : X \to Z$



$p(z)$

Flows

The dimension preserving nature of the invertible map restricts the capacity of flows.

$$f_\theta : X \to Z$$

$$\dim(X) = \dim(Z)$$

But they work very well when set up! And have a lot of potential...

Flows

Very effective generative and density estimation models

to

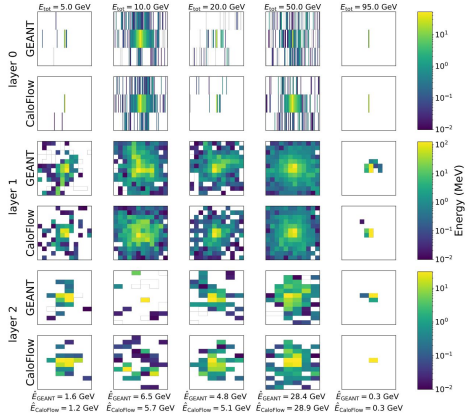


Figure 10. 5 randomly selected π^+ events of GEANT4 and their nearest neighbors in the CALOFLOW samples.

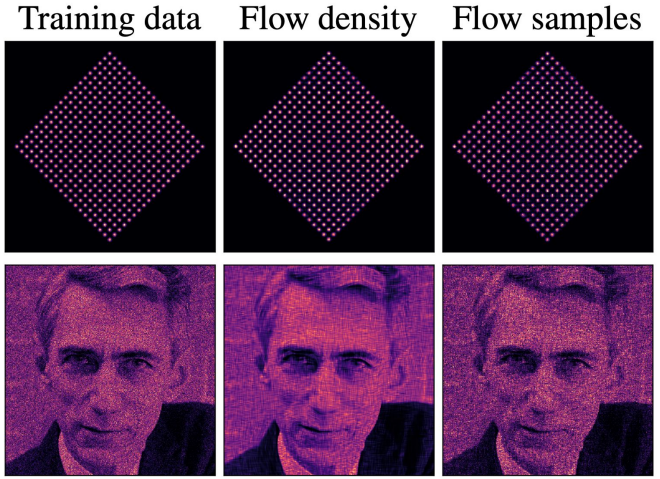
[Silly images]

[Calo Flow]

Flows

Fairly simple and effective in **low dimensions**!

Not easy to scale to high dimensions due to need to **preserve dimensionality**!



Neural Spline Flow

Flows

BUT flows can be quite tricky to set up, and require specific constructions to ensure invertibility

Run into issues of:

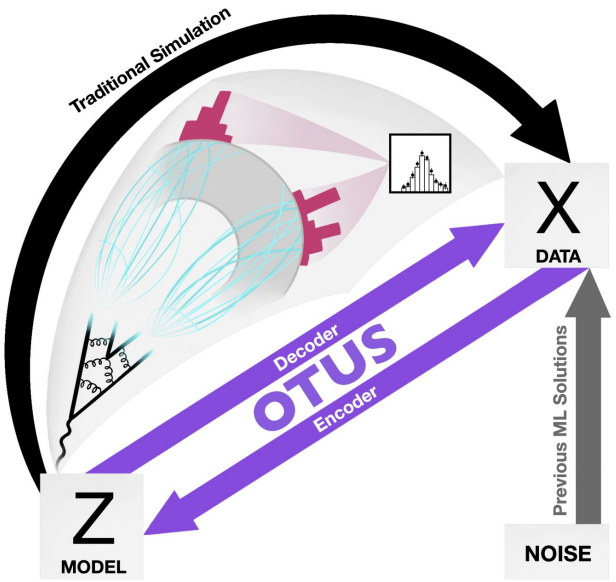
- Architecture
- Speed
- Dimensionality
- Hyper parameter tuning
- ....

Flows

The base distribution can be **anything**, including another flow.

This base flow can learn another distribution entirely, including physics distributions!

Could be used for **unfolding**, curtains, ...



OTUS

Unfolding

In unfolding problems:

The **dimension** of z is not the same as the dimension of x.

The map from X to Z and Z to X should be **stochastic**.

Flows are **deterministic** and **dimension** preserving.

⇒ Cannot learn a map from X to Z and Z to X **directly** using flows.

Unfolding

Unfolding can be done in one direction using a **conditional** flow

Can **unfold**, but map **cannot** be evaluated from partons to detector

Dimension preservation and lack of stochasticity are bottlenecks

Bottlenecks

Can these be addressed?

Dimension preservation

Deterministic transformations

Augmented flows

Dimension increasing flows

Red points end up '**on top**' of blue points in x space. **Not possible** when working with x directly.

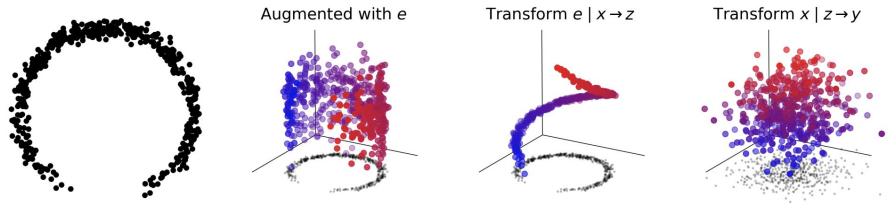


Figure 1. Transforming data x (*left*) via augmented normalizing flows: Black dots and blue dots are marginal and joint data points, respectively. *First step*: augment the data x with an independent noise e. *Second step*: transform the augmented data e conditioned on x into z. *Third step*: transform the original data x conditioned on z into y, resulting in a Gaussianized joint distribution of (y, z)

Figure from Augmented normalizing flows

Augmented flows

Resolve some issues directly

- Induce a **stochastic** map from X to Z
- Allow the **dimension** to be **increased**

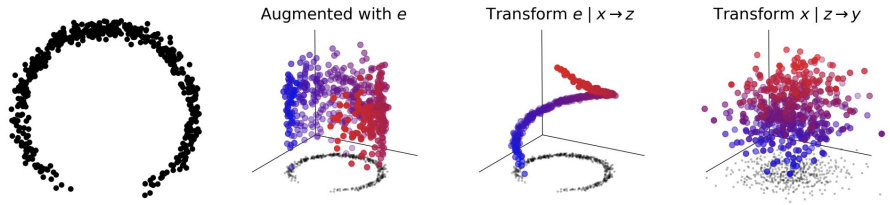


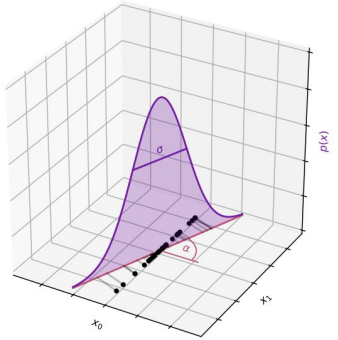
Figure 1. Transforming data x (*left*) via augmented normalizing flows: Black dots and blue dots are marginal and joint data points, respectively. *First step*: augment the data x with an independent noise e. *Second step*: transform the augmented data e conditioned on x into z. *Third step*: transform the original data x conditioned on z into y, resulting in a Gaussianized joint distribution of (y, z)

Figure from Augmented normalizing flows

Funnels

Dimension reducing flow are useful when:

1. Data lives on a smaller dimensional manifold than the input space
2. Data is of very large dimension – modelling is easier in lower dimensions
3. A downstream task is targeted eg) **anomaly detection**



1D density embedded in 2D.
Mflows.

Funnels

Given a vector of data *x* we can split it into two pieces

$$p(x) = p(x_{1:m-1}, x_{m:n}) \qquad \dim(x) = n$$
$$= p(x_+, x_-)$$

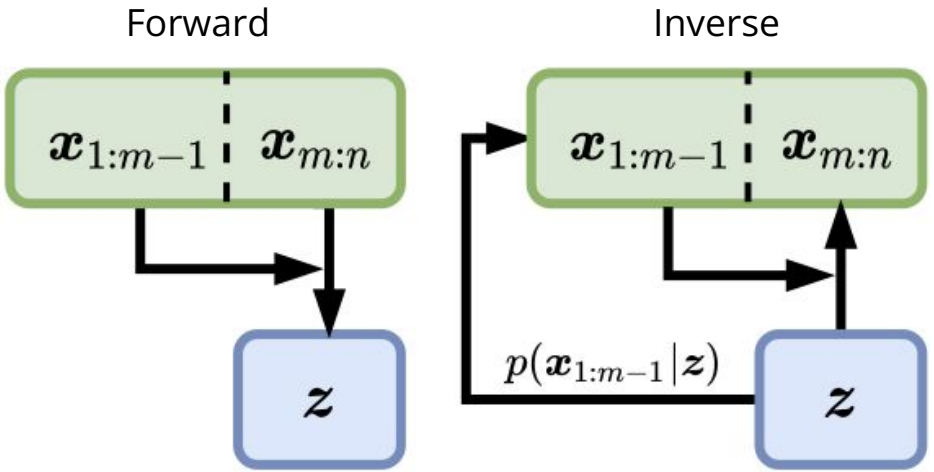
and only transform one of these pieces

$$p(x_+, x_-) = p(x_+, z)|\det(J_x^{f_\theta})| \qquad z = f_{\theta(x_+)}(x_-)$$
$$= p(x_+|z)p(z)|\det(J_x^{f_\theta})|$$

Funnels

It is useful to think of this as a **dimension reducing stochastic map**.

This **factorisation** provides a simple way of thinking about dimension altering flows.



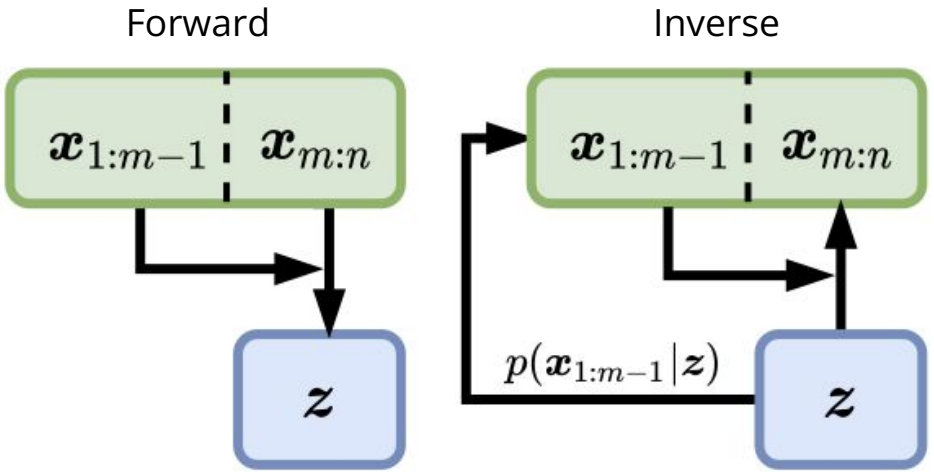
$$p(x) = p(x_+|z)p(z)|\det(J_x^{f_\theta})|$$

Funnels

The x here can be the output of another INN

→ Mixes the input components

→ No need to choose the splitting



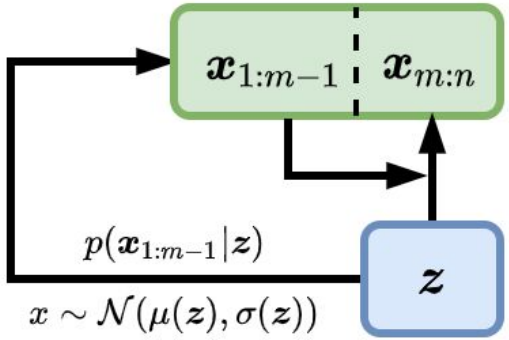
$$p(x) = p(x_+|z)p(z)|\det(J_x^{f_\theta})|$$

Funnels

How to do the ‘reconstruction’?

Can be a parameterised **gaussian**, like in VAEs

- Fast to sample/evaluate
- VAE components can be mixed with flows!



$$p(x) = p(x_+|z)p(z)|\det(J_x^{f_\theta})|$$

Dimension altering flows

There are many possible uses for dimension altering flows, some of which have already been mentioned.

A use case we showed was very useful in the original work was **anomaly detection**

Table 4: Bits per dimension scores for models trained on MNIST evaluated on both MNIST and fashion MNIST and the ratio of the likelihoods on both datasets.

MODEL	LATENT SIZE	MNIST ↓	F-MNIST ↑	RATIO ↑
VAE	4	7.30	10.40	1.42
	16	6.95	12.66	1.82
F-NSF	4	**1.21**	23.42	**19.36**
	16	1.28	11.32	8.84
F-MLP	4	2.86	**30.98**	10.83
	16	2.84	28.59	10.07
NSF	768	1.10	4.80	4.36

Physics data

Dijet system, R = 1, 450 GeV requirement on leading jet p_T.

Using high level variables.

Use a simple five layer INN and compare with a **funnel** that maps to **6 dimensions**.

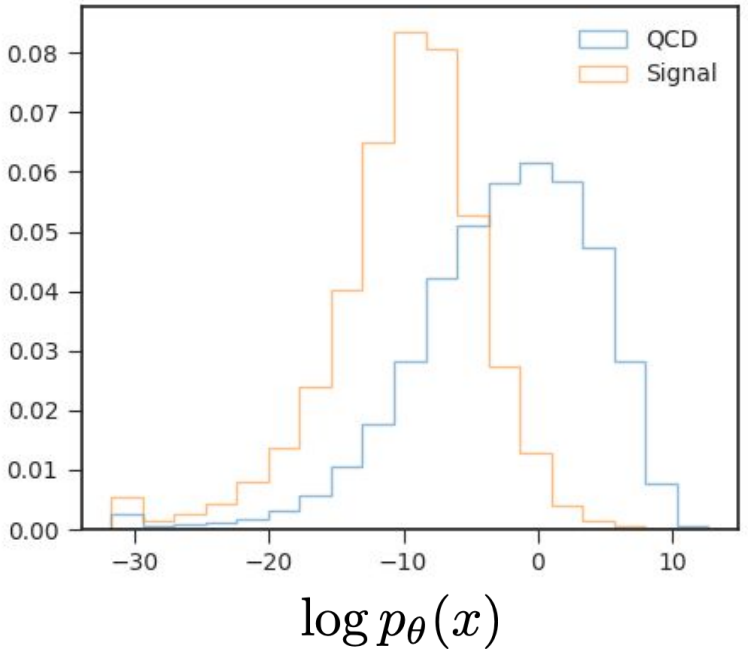
The latent space of the INN will be **12 dimensional**.

Funnel model for AD

Train the model on **QCD only**

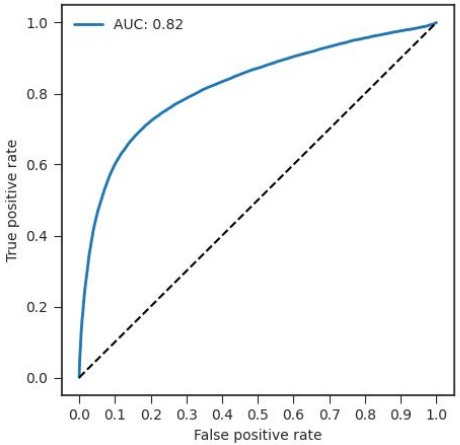
Calculate the (log) **likelihood** of the test and **signal** ($t\bar{t}$) sets

Use the **likelihood** as a **score** for separating signal from data

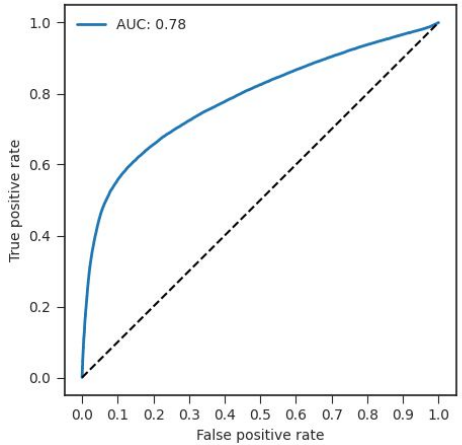


Funnel model for AD

Compare funnel model to a standard equivalent in terms of QCD vs ttbar



Funnel



INN

Funnels for AD

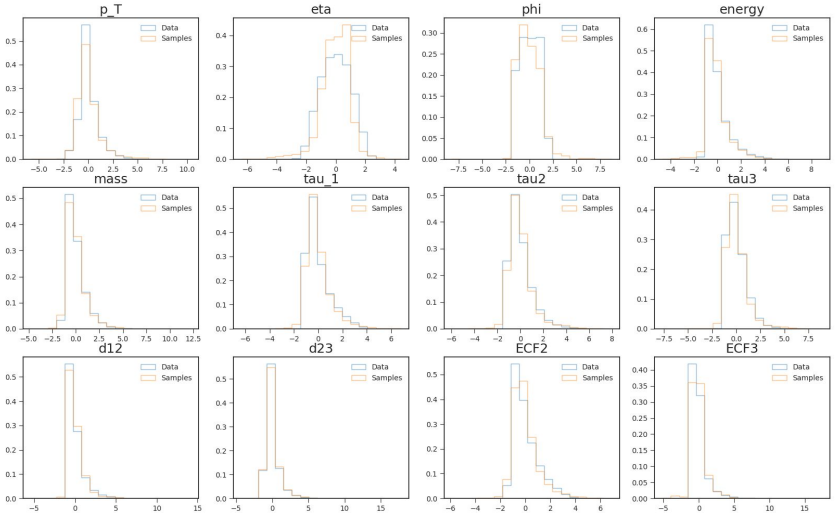
Funnels are better at separating ttbar from QCD, but still suffer from mass correlation

⇒ there are ideas on how to remedy this, but mass decorrelation seems to be a difficult problem in general.

Funnels for generation

Same model can be used to **generate** QCD samples!

These samples are generated from a **6 dimensional** base distribution!



Summary

- Flows can be dimension altering
- Flows can be tricky to set up, especially on large dimensional data
- The composition of dimension increasing and dimension reducing layers leads to two way stochastic transformations
- Flows could (and should) be applied to a wide range of problems in physics

Future work

- These constructions allow the likelihood to be calculated under a broad range of functions
- Some very interesting machine learnings results with these approaches, to be released soon
- Will concretely demonstrate their efficacy on a wide variety of physics problems after this

Back up

Additional slides

surVAE

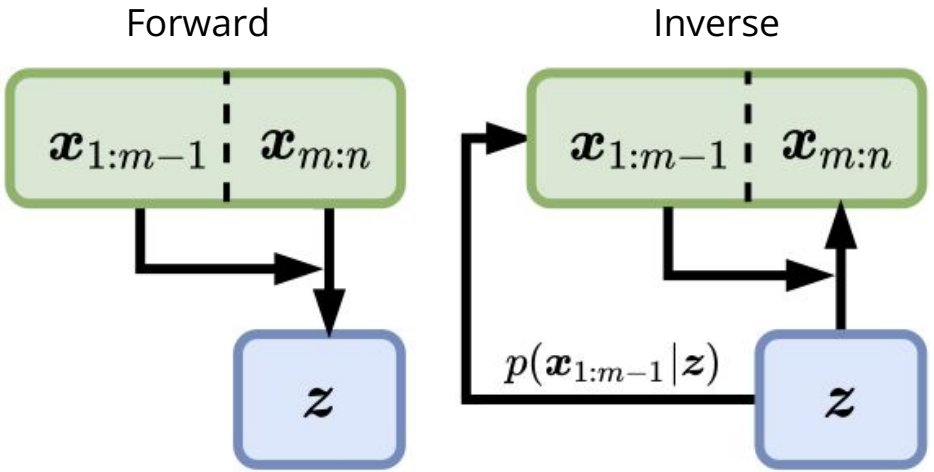
Flows can be composed with stochastic maps, and in some cases, the exact likelihood can still be calculated

See surVAE

Funnels

Dimension increasing versions of this can be derived

Not explored in this talk



$$p(x) = p(x_+|z)p(z)|\det(J_x^{f_\theta})|$$

Simple to set up

Flows require some overhead to set up:

Autoregressive vs Coupling?

Or, skip this, and just use simple linear layers with expressive activations.

Parameterise the linear transformation with the PLU decomposition.

Use rational quadratic splines as activations.

No parameters are a function of the data.