Generative Models

Applications in Neutrino Physics

Radi Radev, CERN

Neutrino Physics & Machine Learning
June 2024

Generative Models

GANs

$$x \sim p_\phi(x)$$

VAEs, Diffusions

$$x \sim p_\phi(x); p_\phi(x) \leq p(x)$$

Transformers

$$x \sim p_\phi(x); \prod_i p_\phi(x_i \,|\, x_1, x_2, \ldots, x_{i-1})$$

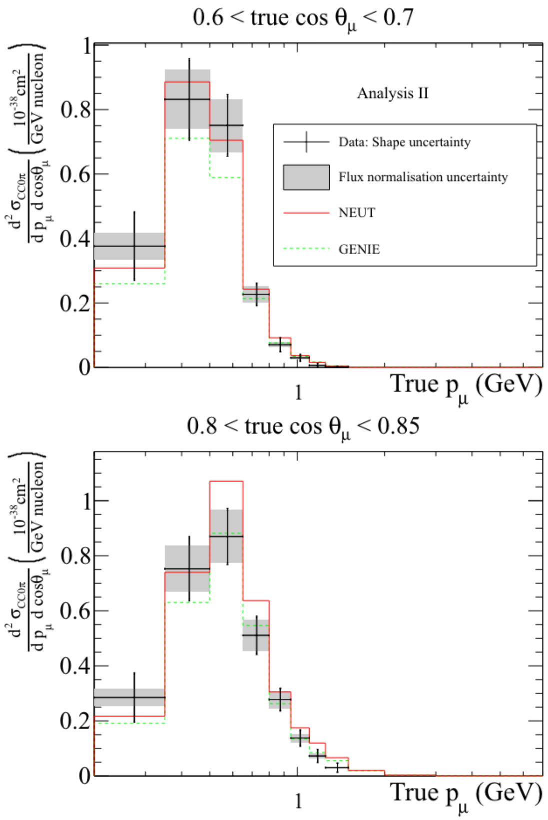
Flows

$$x \sim p_\phi(x); p_\phi(x) \approx p(x)$$

Normalising Flows

for Neutrino Cross Section Measurements

Cross Sections

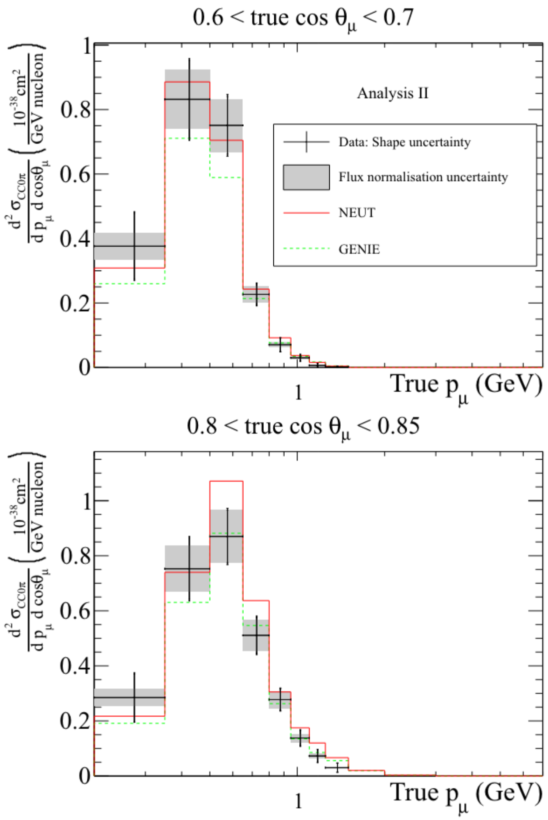


T2K 2002.09323

Cross Sections

To quantify uncertainties in a cross-section result alternate measurements (toys, universes) are produced by varying uncertainty parameters (detector, cross-section).

This produces variations within bins.



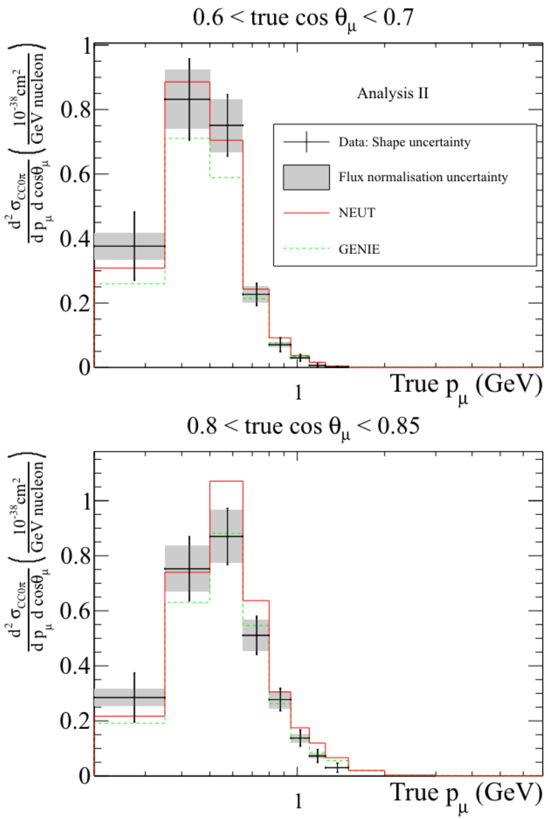
T2K 2002.09323

Cross Sections

To quantify uncertainties in a cross-section result alternate measurements (toys, universes) are produced by varying uncertainty parameters (detector, cross-section).

This produces variations within bins.

Typically presented as a set of central value with a **covariance matrix** defining the uncertainties and correlations between bins.



T2K 2002.09323

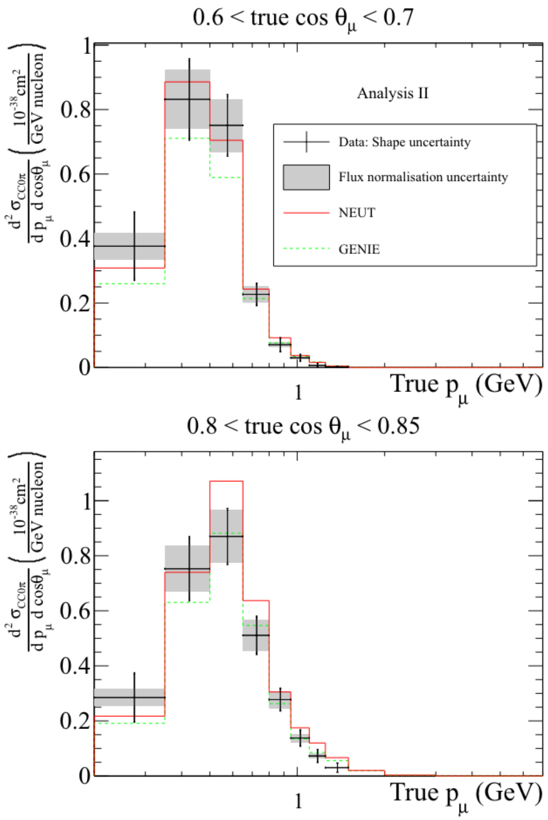
Cross Sections

To quantify uncertainties in a cross-section result alternate measurements (toys, universes) are produced by varying uncertainty parameters (detector, cross-section).

This produces variations within bins.

Typically presented as a set of central value with a **covariance matrix** defining the uncertainties and correlations between bins.

assumes a gaussian



T2K 2002.09323

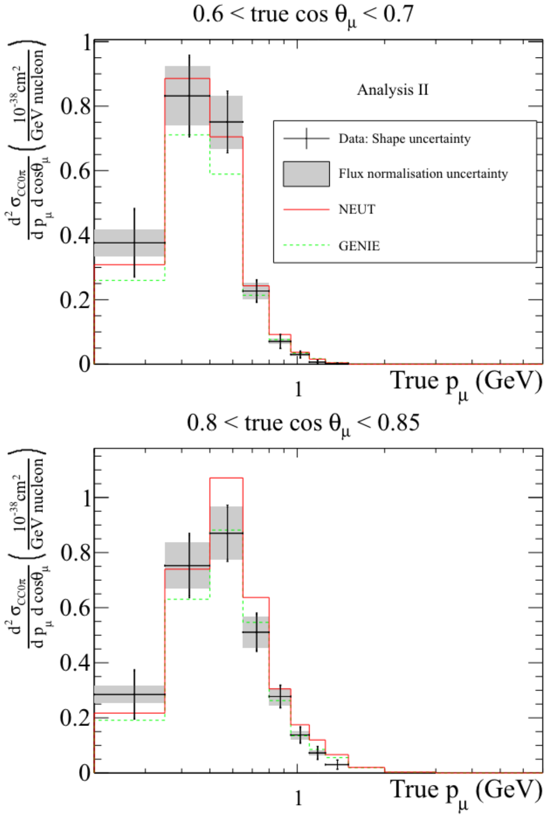
Cross Sections

To quantify uncertainties in a cross-section result alternate measurements (toys, universes) are produced by varying uncertainty parameters (detector, cross-section).

This produces variations within bins.

Typically presented as a set of central value with a **covariance matrix** defining the uncertainties and correlations between bins.

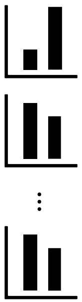
assumes a gaussian $\xrightarrow{\text{if bad approx}}$ **poor ν-model preds**



T2K 2002.09323

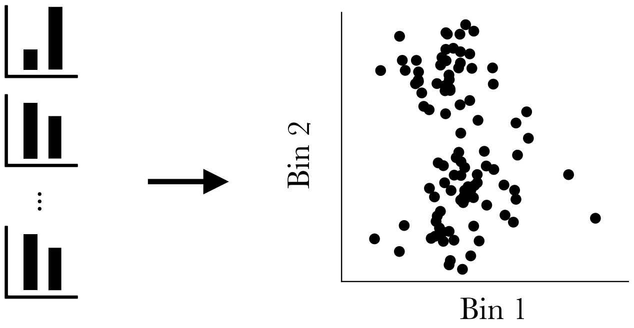
Overview

Vary Uncertainties



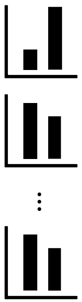
Overview

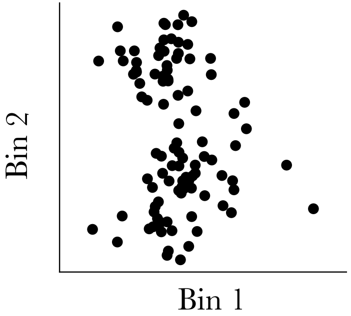
Vary Uncertainties



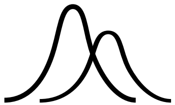
Overview

Vary Uncertainties





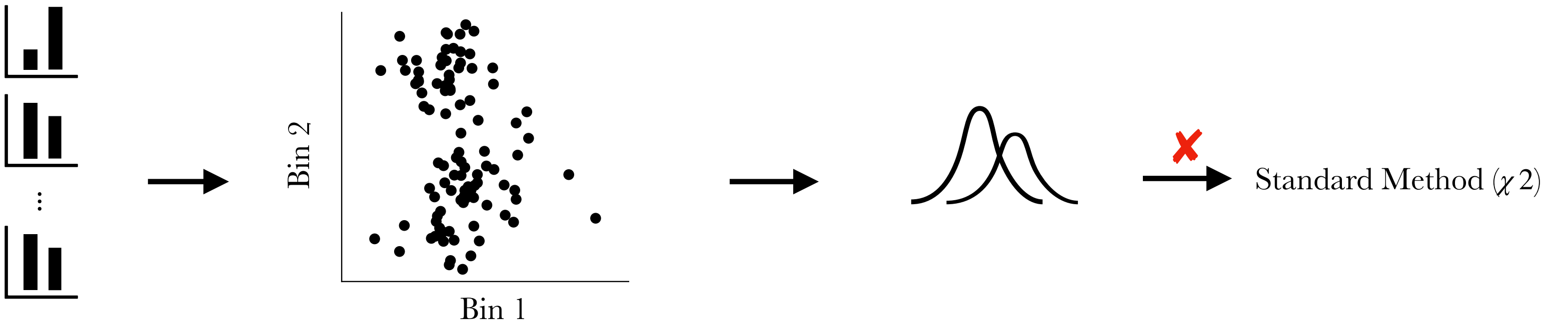
1. Multivariate Check for non-Gaussianity



Overview

Vary Uncertainties

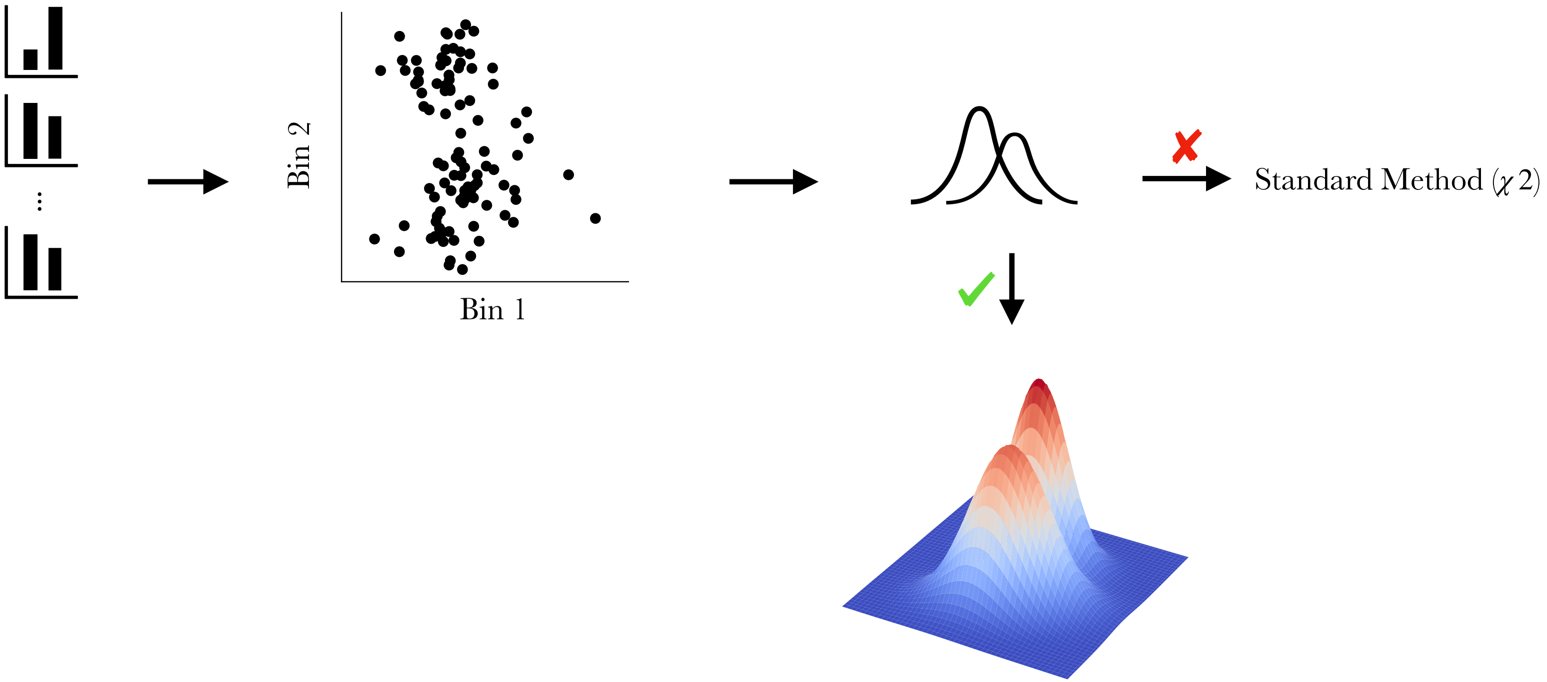
1. Multivariate Check for non-Gaussianity



Overview

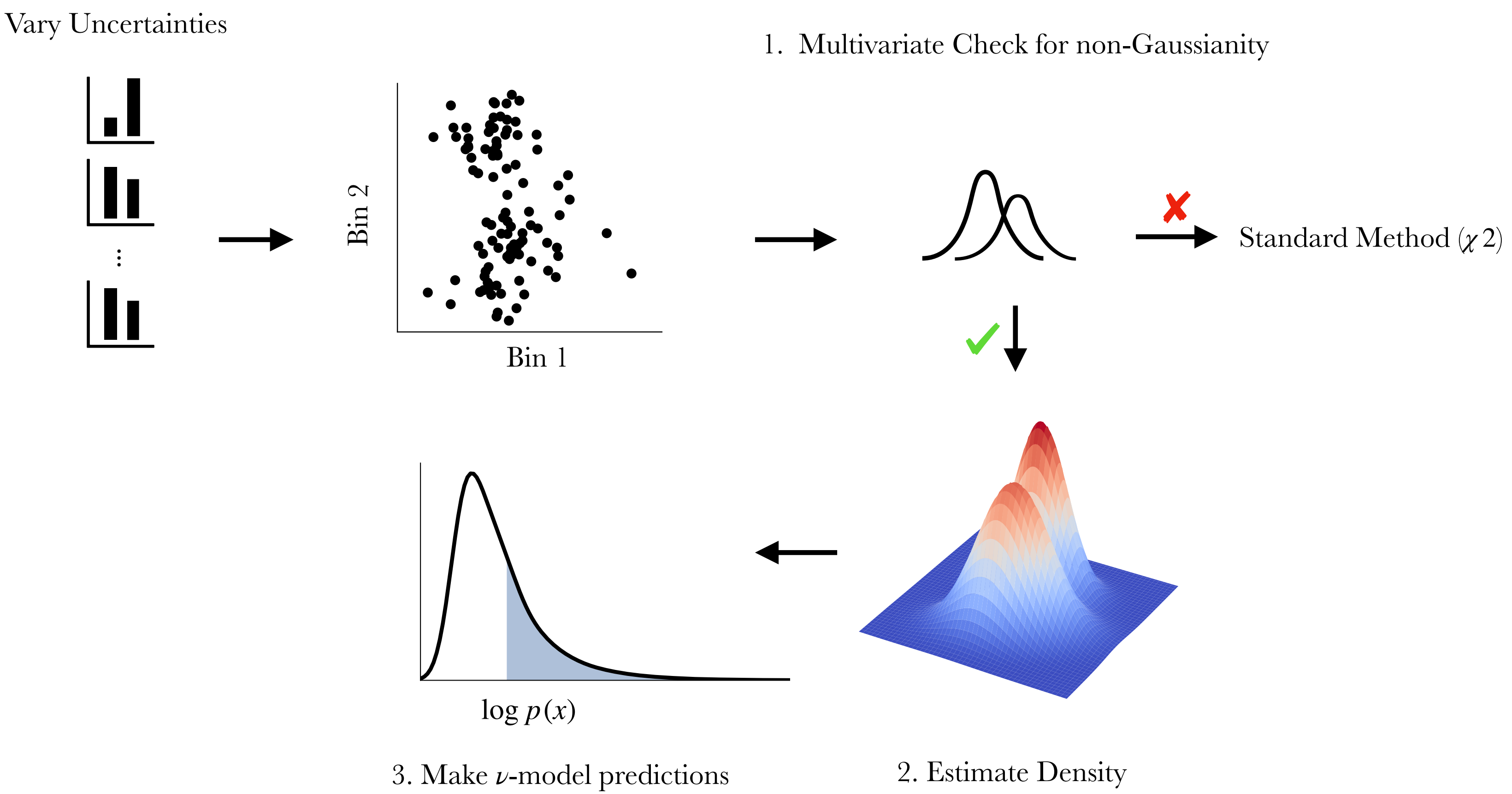
Vary Uncertainties

1. Multivariate Check for non-Gaussianity



2. Estimate Density

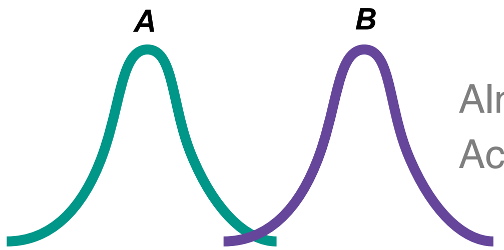
Overview


Vary Uncertainties
1. Multivariate Check for non-Gaussianity
Bin 2
Bin 1
Standard Method (χ 2)
log $p(x)$
3. Make ν-model predictions
2. Estimate Density

Overlap with Classifiers

A well trained classifier will approximate the density ratio between two distributions.

Test accuracy can be used to quantify the overlap between samples from any 2 distributions.



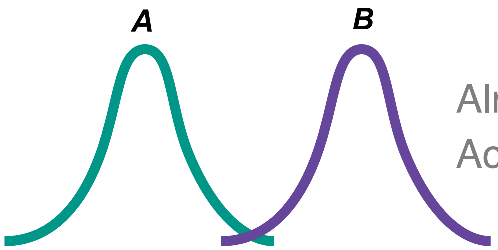
Almost disjoint

Acc ≈ 1.0

Overlap with Classifiers

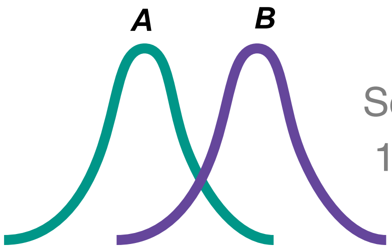
A well trained classifier will approximate the density ratio between two distributions.

Test accuracy can be used to quantify the overlap between samples from any 2 distributions.



Almost disjoint

Acc ≈ 1.0



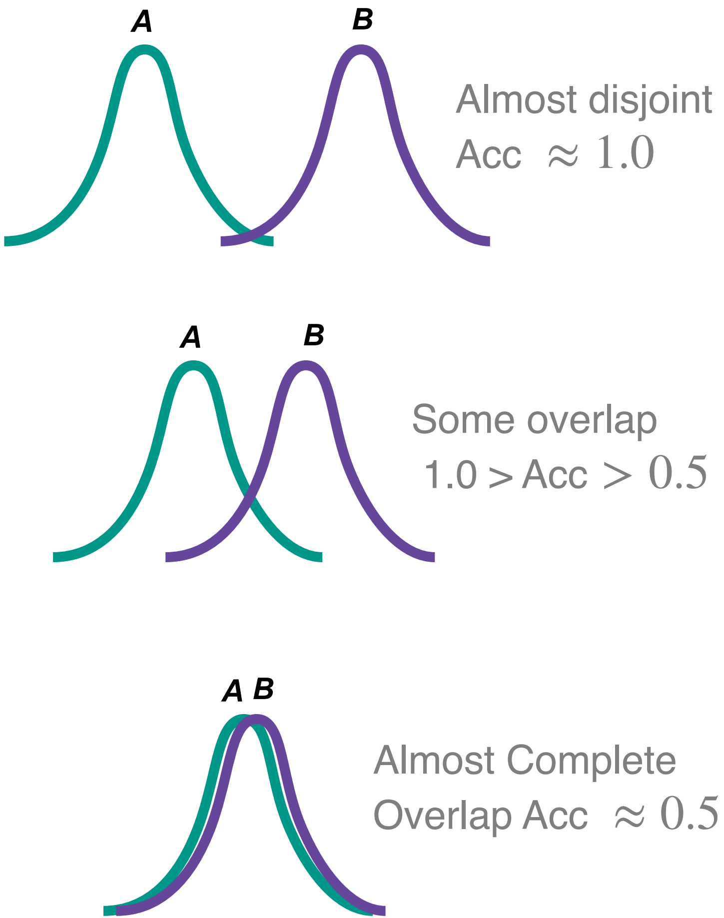
Some overlap

$1.0 > \text{Acc} > 0.5$

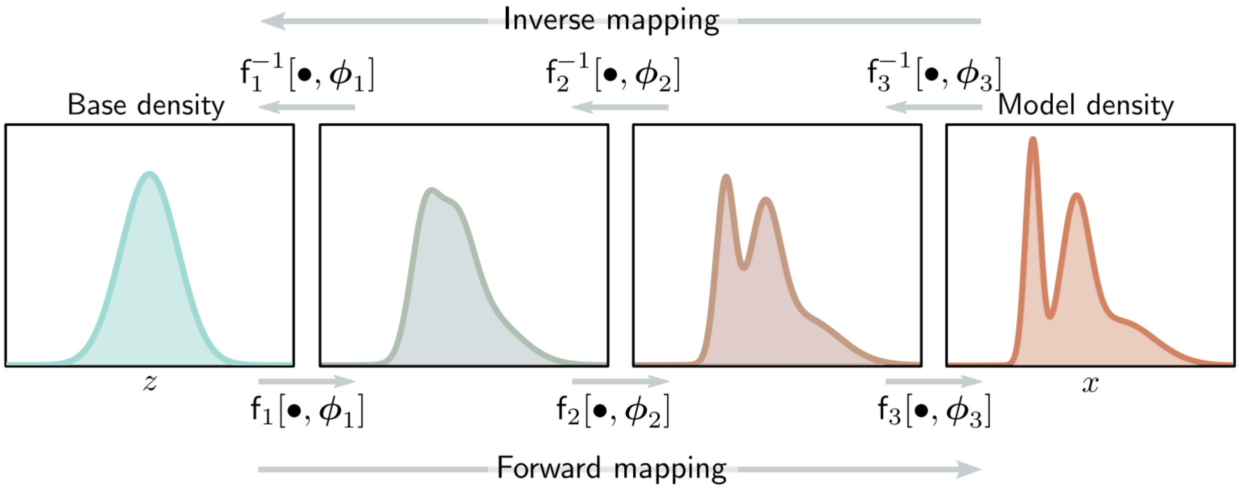
Overlap with Classifiers

A well trained classifier will approximate the density ratio between two distributions.

Test accuracy can be used to quantify the overlap between samples from any 2 distributions.



Density Estimation with Normalising Flows



$$f_\phi = f_0 \circ f_1 \ldots f_N$$

$$f_\phi(z) = x$$

generate new samples

$$p_x(x;\phi) = p_z(f_\phi^{-1}(x)) \left| \det \left(\frac{\partial f_\phi^{-1}(x)}{\partial x} \right) \right|$$

compute the density of a sample

Figure form the Understanding Deep Learning book by Simon J.D. Prince.

Neutrino Model Predictions

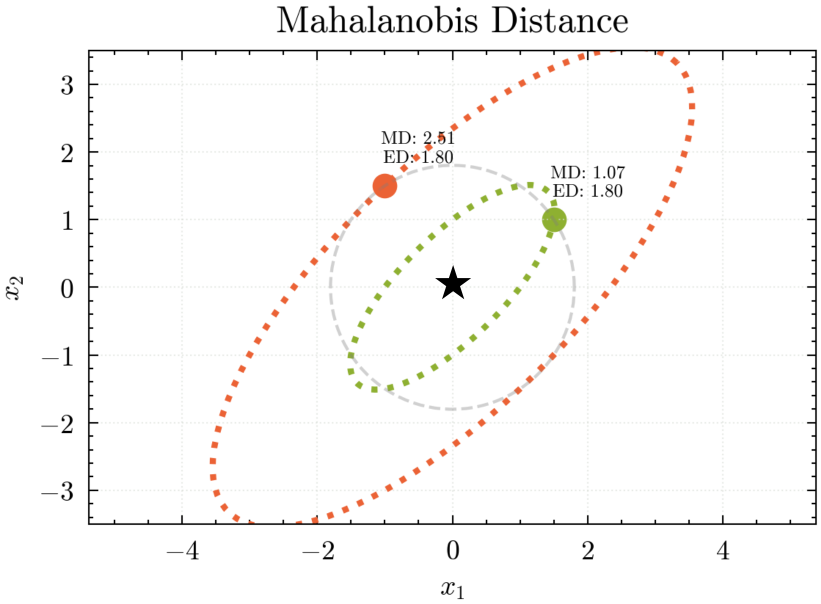
For a Gaussian to compute the p-value:

$$p_{\text{val}} = \int_{\mathcal{M}(\mathbf{x})}^{\infty} \chi^2(t)dt = 1 - \text{CDF}_{\chi^2}(\mathcal{M}(\mathbf{x}))$$

Where we have used $\mathcal{M}$ to be the distance of the toy $\mathbf{x}$ to the mean and covariance of the distribution.

$$\mathcal{M}(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

Because we use the Mahalanobis distance we are making another Gaussian assumption!



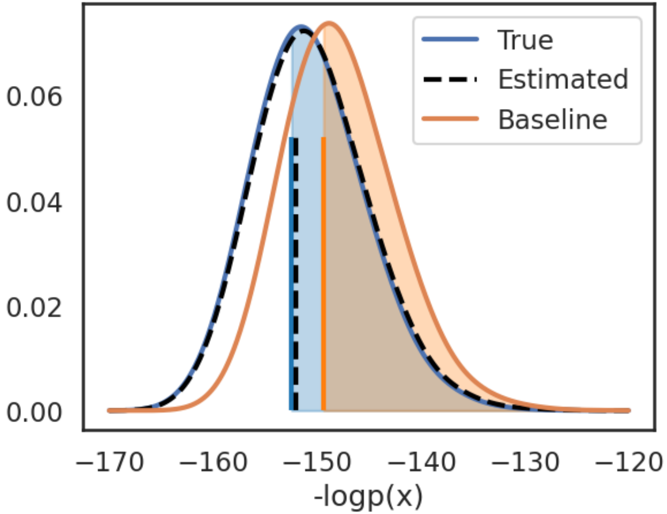
Neutrino Model Predictions with Flows

To make model comparisons we can use the predicted

$$\log p(\hat{\mathbf{x}}) > \log p(\mathbf{x}^*))$$

Using the estimates from the flows for each toy, we can look at the fractional count of toys with NLL larger than the current toy.

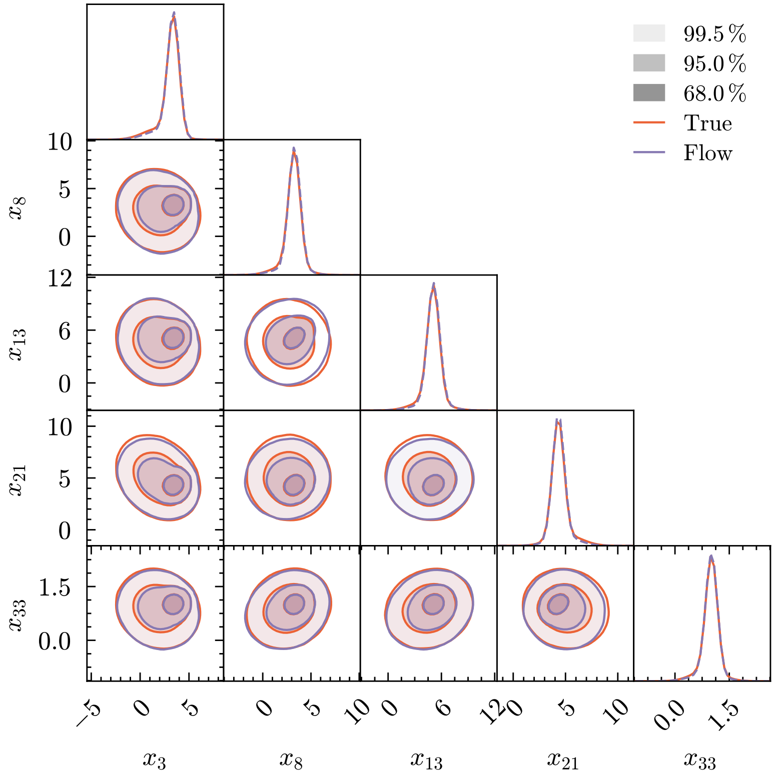
$$p_{\mathrm{val}} \approx \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(\log p(\mathbf{x}^{(i)}) \geq \log p(\hat{\mathbf{x}}))$$



50D Mixture Distribution

We start with quasi-realistic example of Gaussian Mixture distribution in 50D roughly following the covariance matrix of a T2K result.

The normalising flow is able to capture the distribution well.

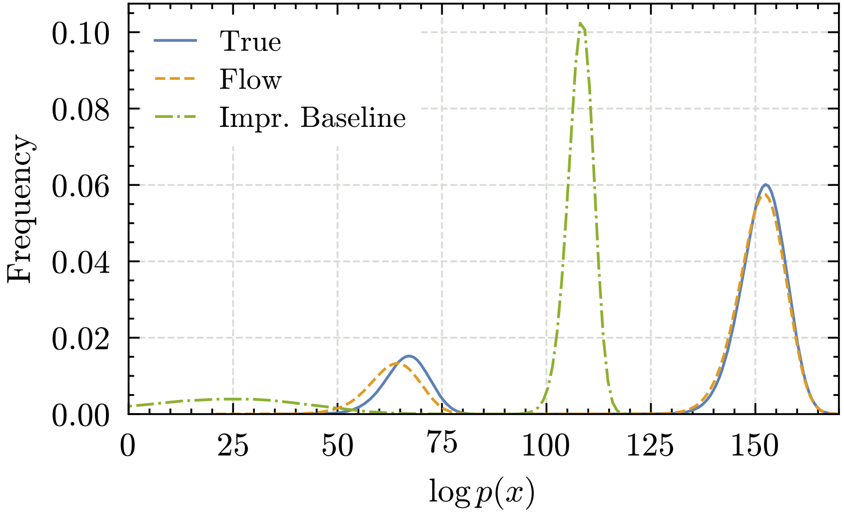


50D Mixture Distribution

We know the true log-probability so we can compute p-values exactly.

We show we can improve the baseline method (using the Mahalanobis distance) by computing the p-value from the estimated log-probability from a Gaussian baseline.

Finally the normalising flow achieves even better p-value estimation



50D Mixture Distribution

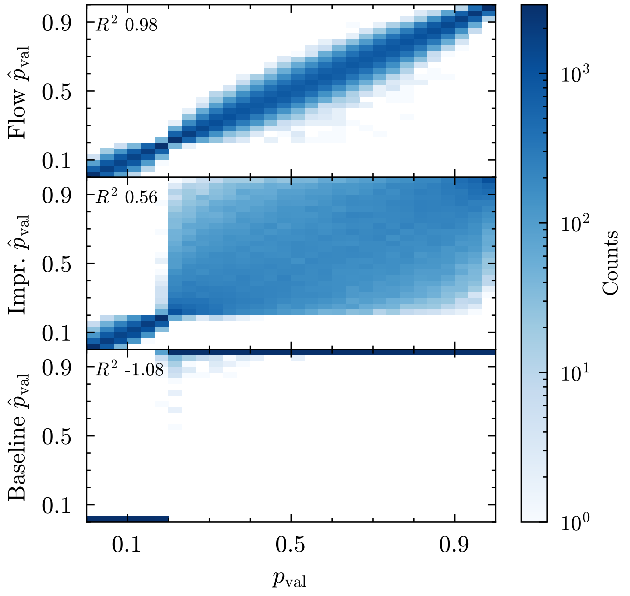
We know the true log-probability so we can compute p-values exactly.

We show we can improve the baseline method (using the Mahalanobis distance) by computing the p-value from the estimated log-probability from a Gaussian baseline.

Finally the normalizing flow improves the p-value estimation even further.

	samples		p_{val}		
Method	C2ST ↓	NLL ↓	r ↑	MSE ↓	R^2 ↑
Baseline	0.99	–	0.70	0.173	-1.07
Improved Baseline	0.99	-90.4	0.78	0.036	0.56
Normalizing Flow	0.55	-133.6	0.99	0.002	0.98
True	0.50	-134.6			

↓ (↑) - Lower (higher) is better.



Realistic Measurements

Using toys from a recent result by MicroBooNE 2301.03706, we see that that systematic uncertainties cannot be described fully by a Gaussian.

These plots do not include statistical uncertainties (in progress) so in reality the non-Gaussian effect would be less prominent.

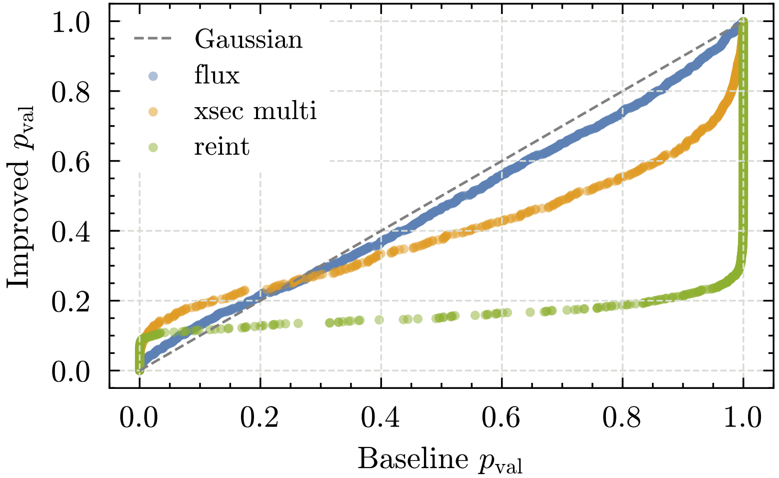


TABLE I. C2ST and p-values for different uncertainty types of the MicroBooNE data assuming infinite statistics.

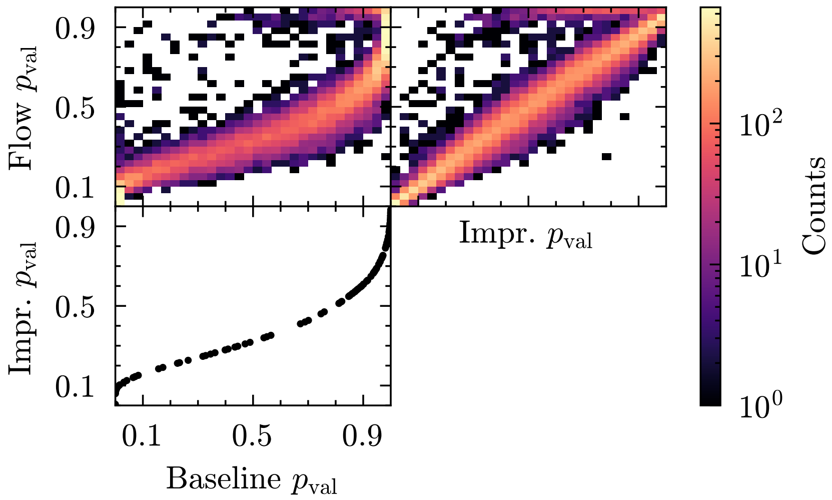
Type	N_{toys}	C2ST	p_{val}
Cross-Section	500	66.3 ± 4.6	0.01
Re-interaction	1000	92.4 ± 1.1	0.00
Flux	1000	52.2 ± 5.0	0.33

Realistic Measurements

We also use toys from a T2K-like simulation projected to 5 times current statistics.

In this case we also find that the Gaussian approximation does not hold.

We further show the normalizing flow can be trained to account for non-Gaussian features.



Results also presented at Neutrino 2024

Future Work

We have shown normalizing flows can be used to estimate the distribution of alternate measurements in cross-section measurements 📈

This could be key to allow more accurate neutrino interaction model predictions especially in systematics-limited cross-section results.

Stay tuned for upcoming paper and software package 💻

Transformers

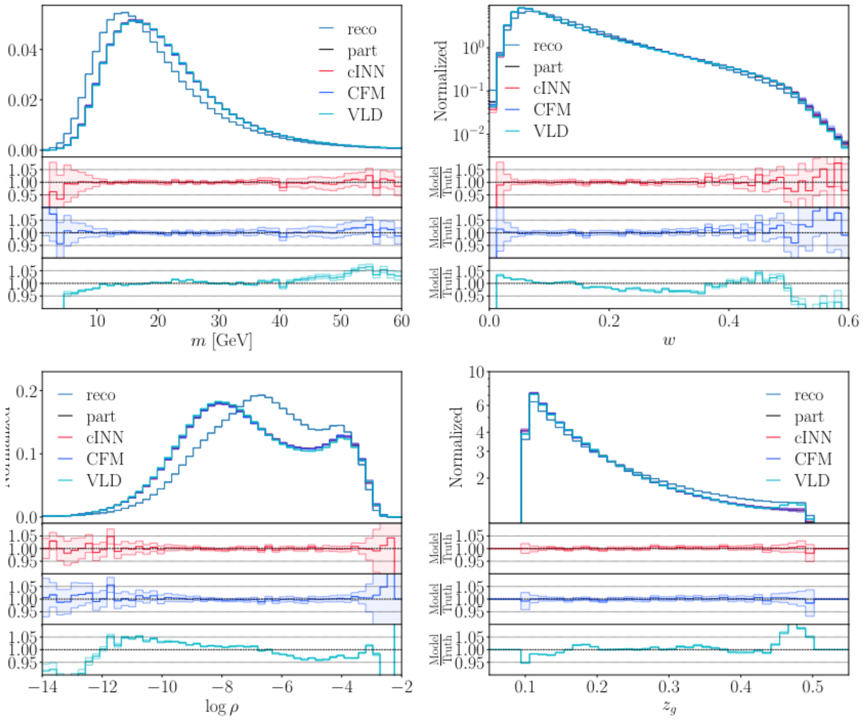
for Unfolding Tasks

Unfolding

$$p(x_{true}|x_{reco})$$

Generative models can be used to find the **true variables** given **reconstructed quantities** without weights.

A generative model trained on pairs of reco and true data.



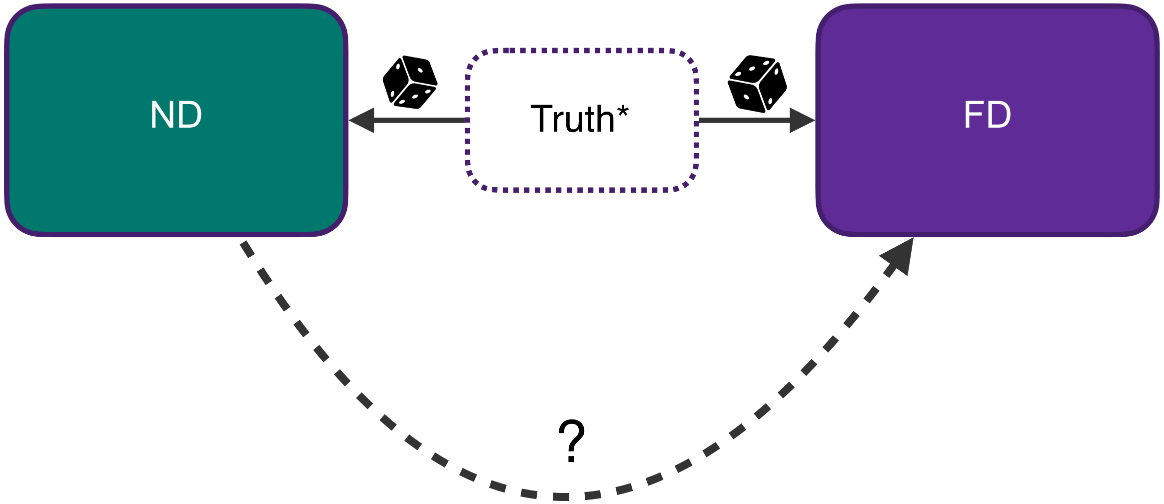
Unfolding 6 detector variables simultaneously [Heutsch et al. 2024]

Unfolding and Resmeraring

$$p(x_{FD}|x_{ND})$$

What if instead of 1-way unfolding you wanted to find the mapping between ND and FD reco directly?

Also possible!



Conditional ND -> FD Generation

$$p(x_{FD}|x_{ND})$$

To model this conditional distribution we use an autoregressive **transformer** architecture.

$$p(x_{FD}\,|\,x_{ND}) = \prod_i p(x_{i_{FD}}\,|\,x_{1_{FD}}, x_{2_{FD}}, \ldots, x_{i-1_{FD}}, x_{ND})$$

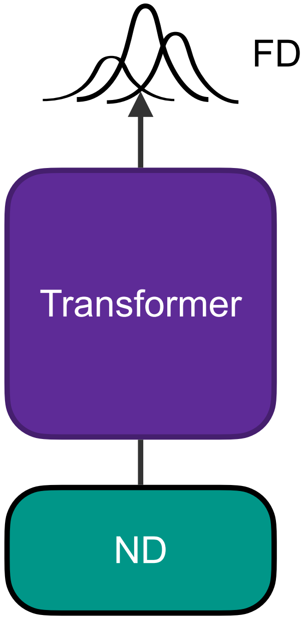
Scales really well, and allows to impose (physically) motivated conditions on each predicted far detector variable.

Conditional ND -> FD Generation

Instead of predicting discrete values, for each FD variable the transformer predicts the components of a gaussian mixture $N : p, \mu, \sigma$

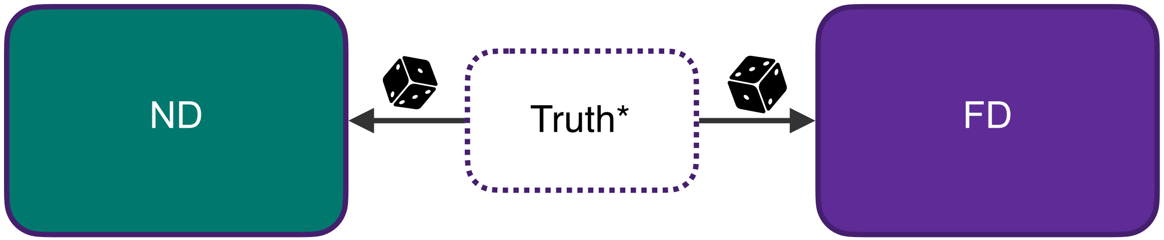
Trained by minimizing negative log-likelihood.

Allows us to impose conditions on the predicted variable, as long as they have a tractable Jacobian transformation.



References: [2305.10475](2305.10475), [2312.02116](2312.02116)

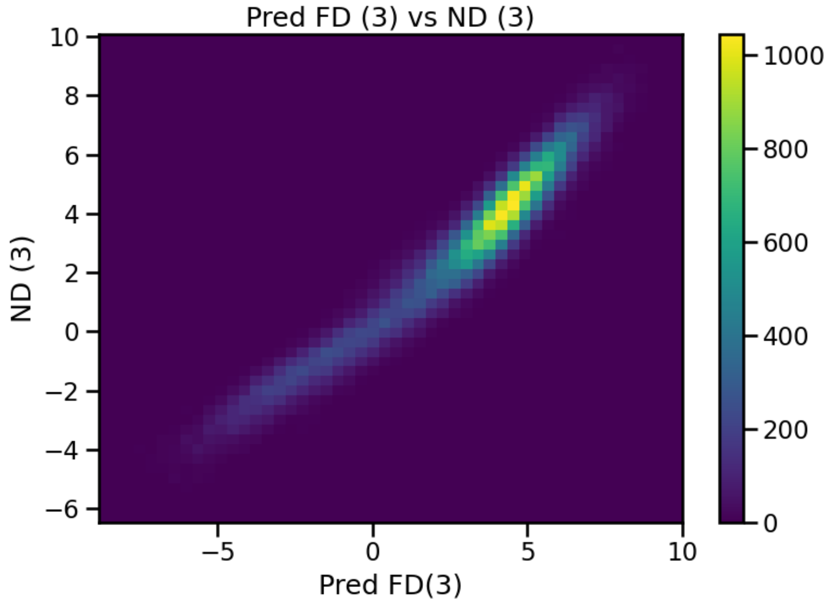
Toy 4ND -> 4FD Translation

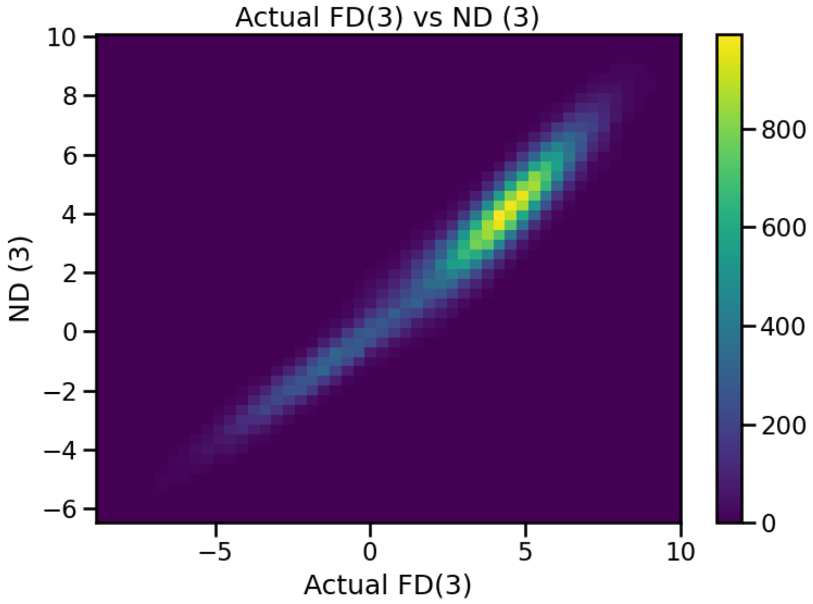


A toy task showcasing this method:

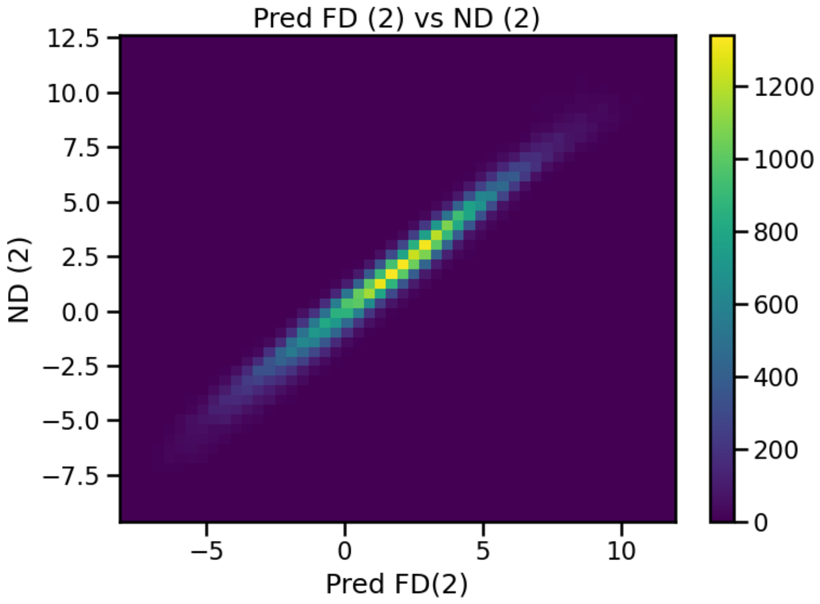
$$X_{\text{truth}} \sim Y$$
$$X_{\text{near}} = X_{\text{truth}} + \varepsilon_{\text{near}}$$
$$X_{\text{far}} = X_{\text{truth}} + \varepsilon_{\text{far}}$$

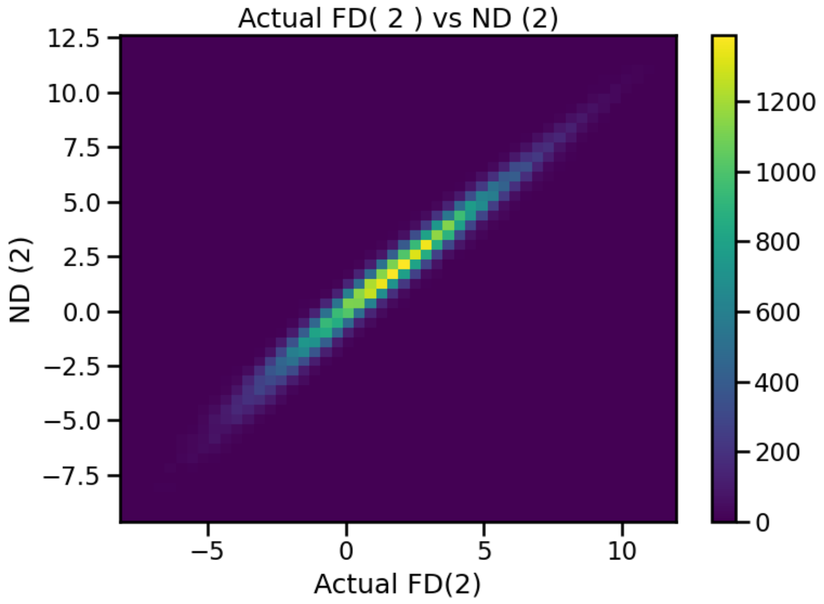
Results on the Toy Task



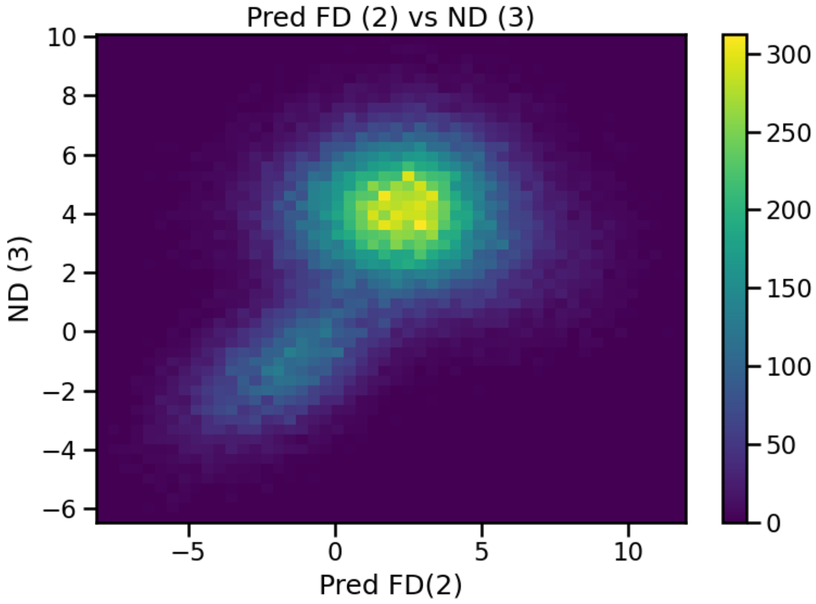


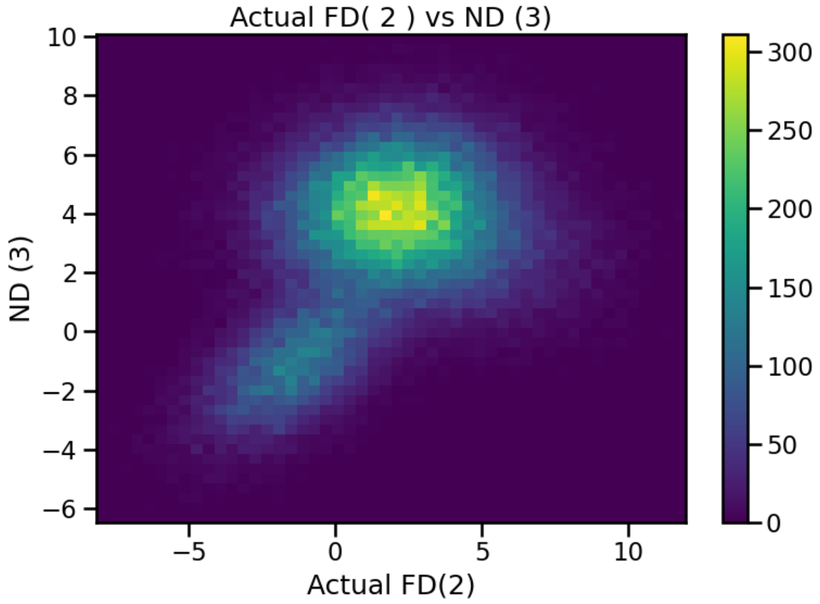
Results on the Toy Task



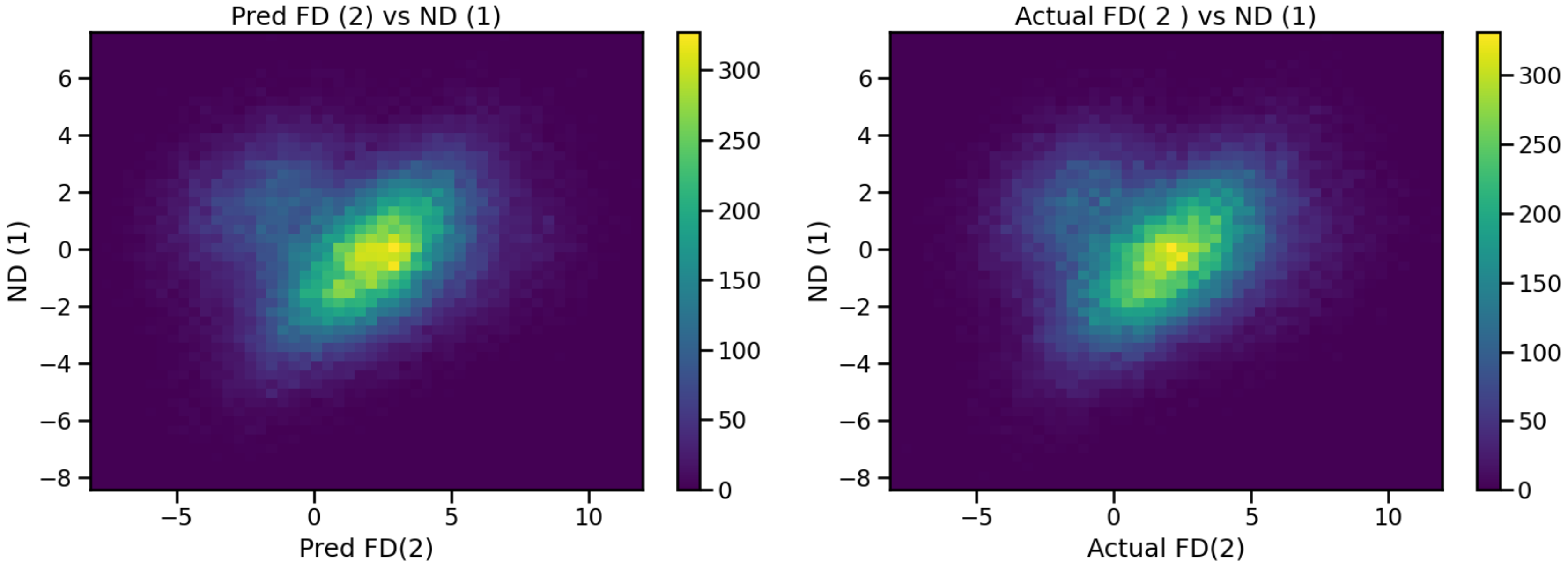


Results on the Toy Task





Results on the Toy Task

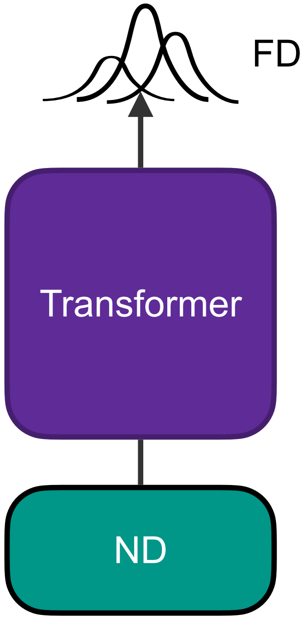


Future Work

We have shown an application of a transformer for conditional generation of reconstructed variables.

The method is scalable and allows to impose physically motivated conditions.

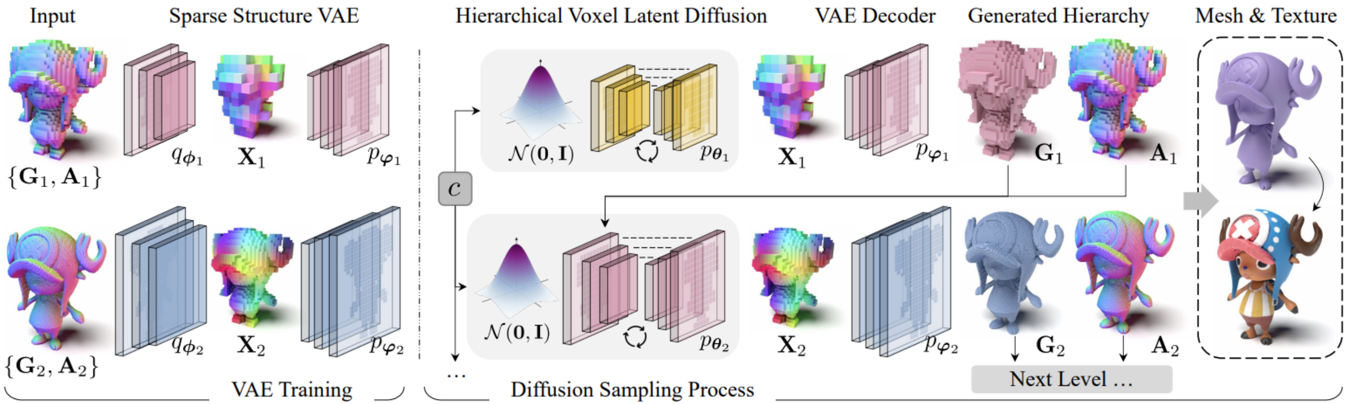
We plan to present results on the effectiveness of this method when combined with the DUNE-PRISM analysis.



Flow Matching/ Diffusion

For 3D LArTPC Generation

XCube - High Fidelity 3D Generation



Recent work by NVIDIA shows generating high-fidelity (1024^3) voxelized data.

Planning to retrain their model on PILArNet.

Possible applications to infilling dead regions within LArTPC.

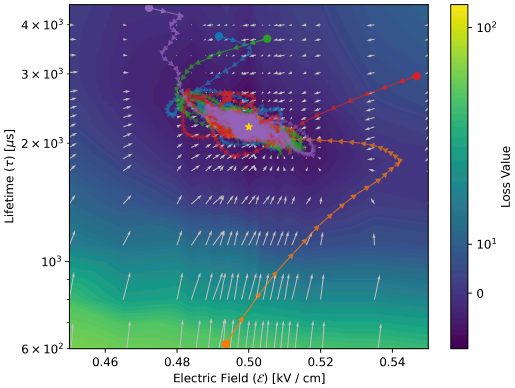
Conclusion

Conclusion & Highlights

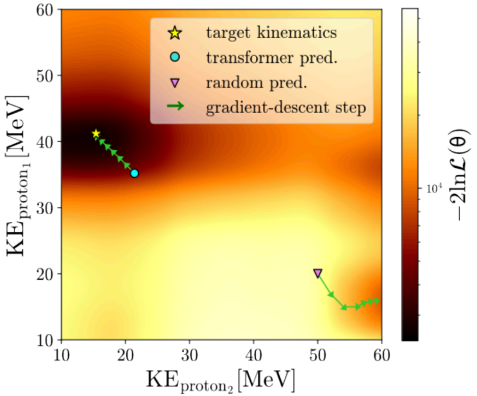
We have shown potential applications of transformers, normalizing flows and diffusion within neutrino physics.

The generative modeling field is rapidly advancing and would enable progress within our field as well.

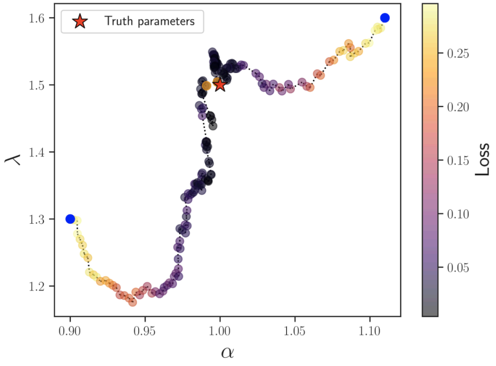
Many more works exist that show exciting applications of these methods making analysis techniques less interaction model dependent and allowing to tune simulation better to data.



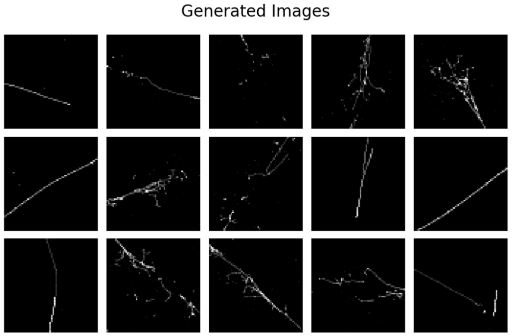
Gasiorowski S et al. 2023



Alonso-Monsalve S et al. 2023



Barham Alzás P. 2024



Imani Z. 2023

Thank you

Get in touch!

radi.radev@cern.ch

References

Normalising Flows - https://arxiv.org/abs/1908.09257, https://github.com/probabilists/zuko

FFJORD - Continuos Normalising Flows - https://arxiv.org/abs/1810.01367

Flow Matching - https://github.com/atong01/conditional-flow-matching , https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html

Understanding Deep Learning book - https://udlbook.github.io/udlbook/
(Excellent material about flows, diffusion, vae and deep learning in general)

Backup

Continuous Normalising Flows

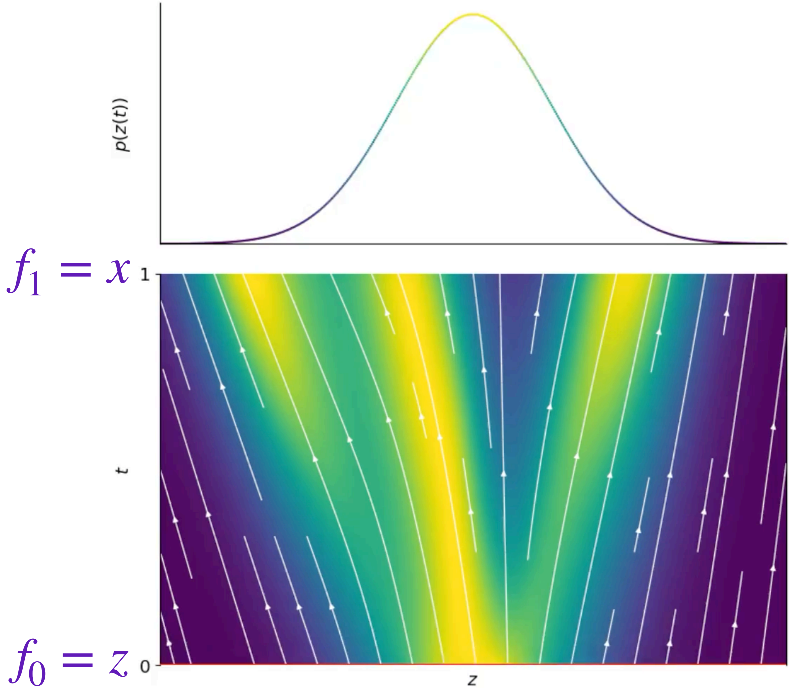
Instead of training a set of discrete flows we can model the transformation as an ODE.

$$\frac{d}{dt} f_t(x) = v_t(f_t(x))$$

Given a suitable choice for a target velocity field we can parametrise the velocity field by a NN $v_\phi(x, t)$

Then to sample we simply integrate the velocity starting from t_0

Only one NN and no invertibility constraints!



Studies with known PDF

Data for studies with known PDF

We also explored scenarios where we know the PDF and we are not limited by data.

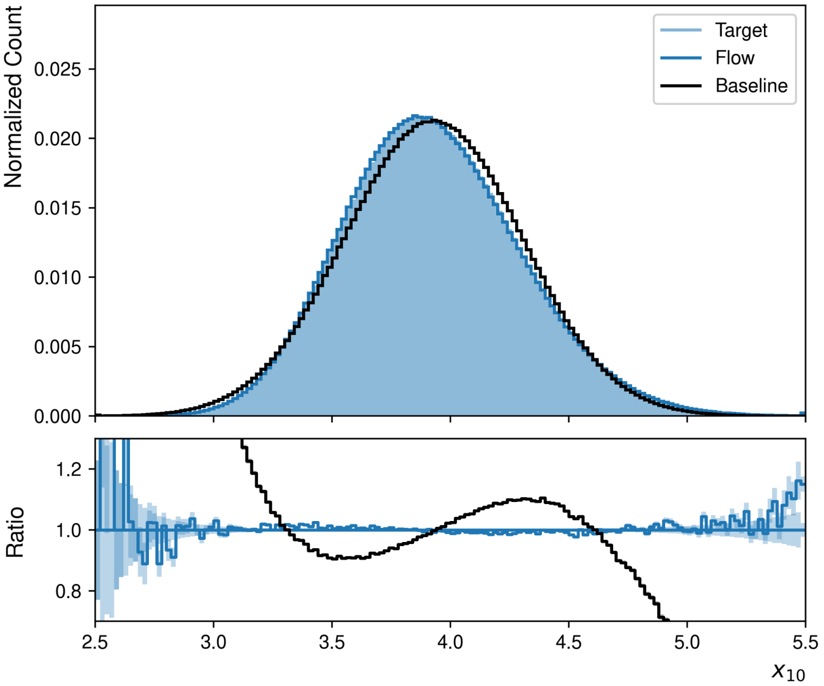
Two studies in this case:

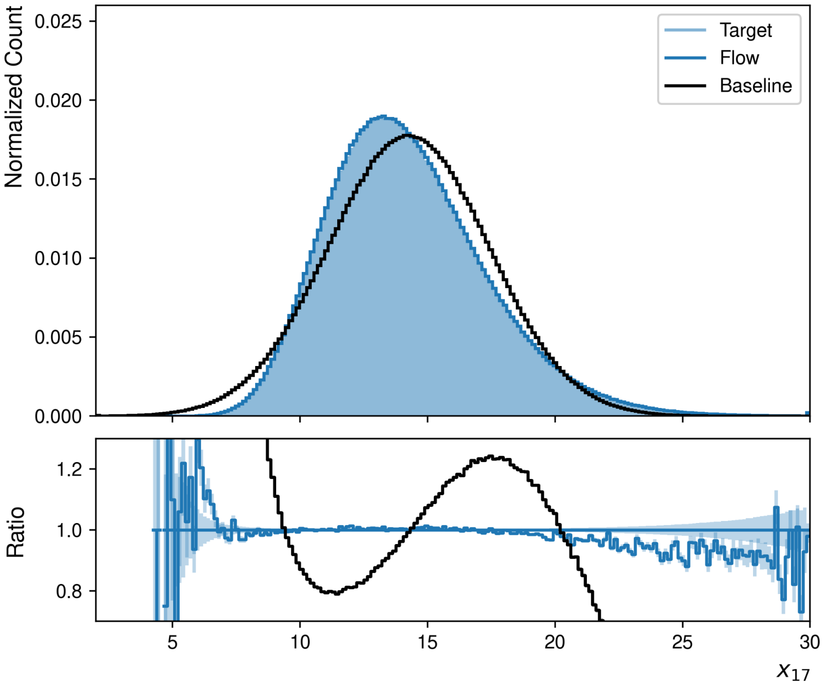
We use the T2K data release from [2002.09323] to generate toys-like data with 50 bins:

- **Extreme case**: skewed lognormal toys: exp(toy)
- **Moderate case**: less-skewed lognormal toys: lognormal distribution in the original space

Extreme case

Marginal Plots



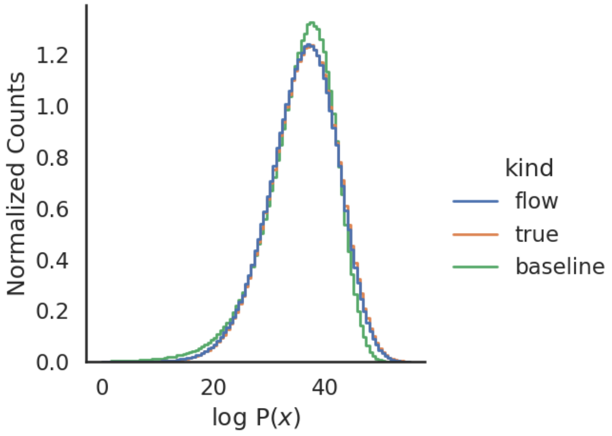


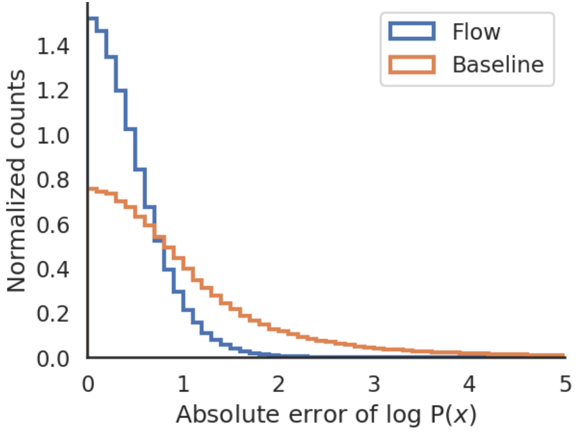
Two example bins of the 49-dimensional distribution.

Evaluation

Since we constructed the distribution from which the toys are drawn we can evaluate the true probability for each toy.

Compare it to the estimation of the flow model, and with the gaussian baseline.



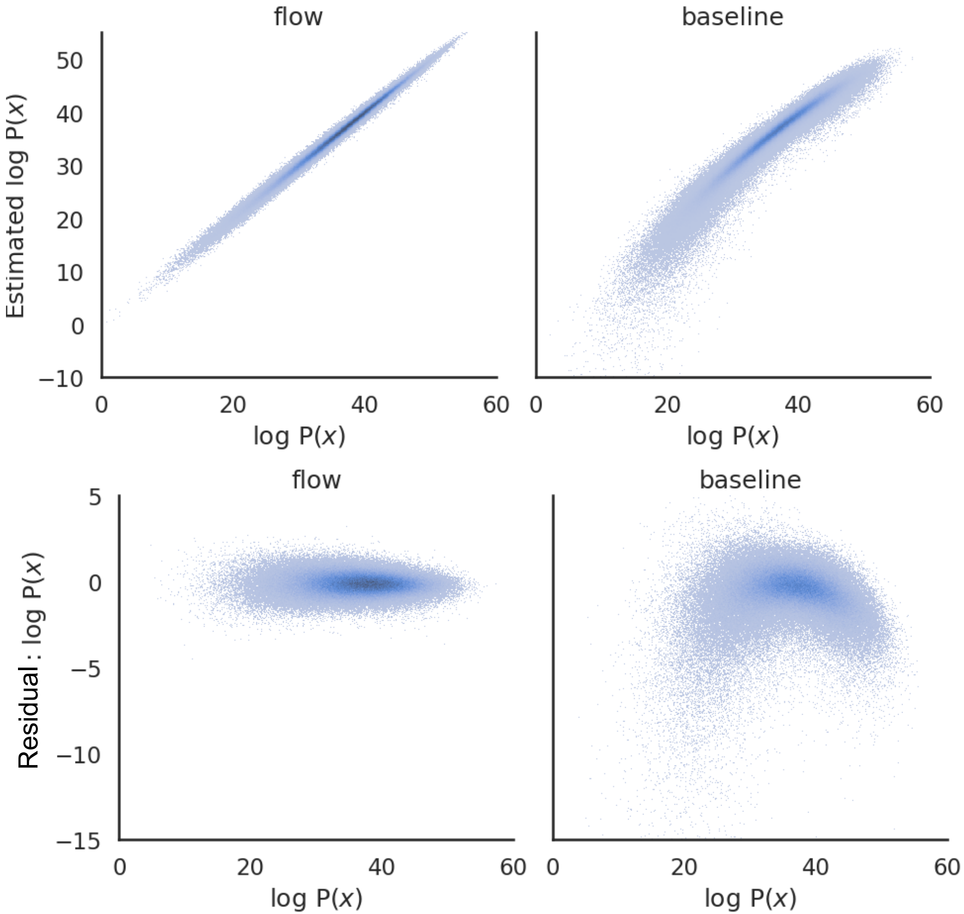


Evaluation

Since we constructed the distribution from which the toys are drawn we can evaluate the true probability for each toy.

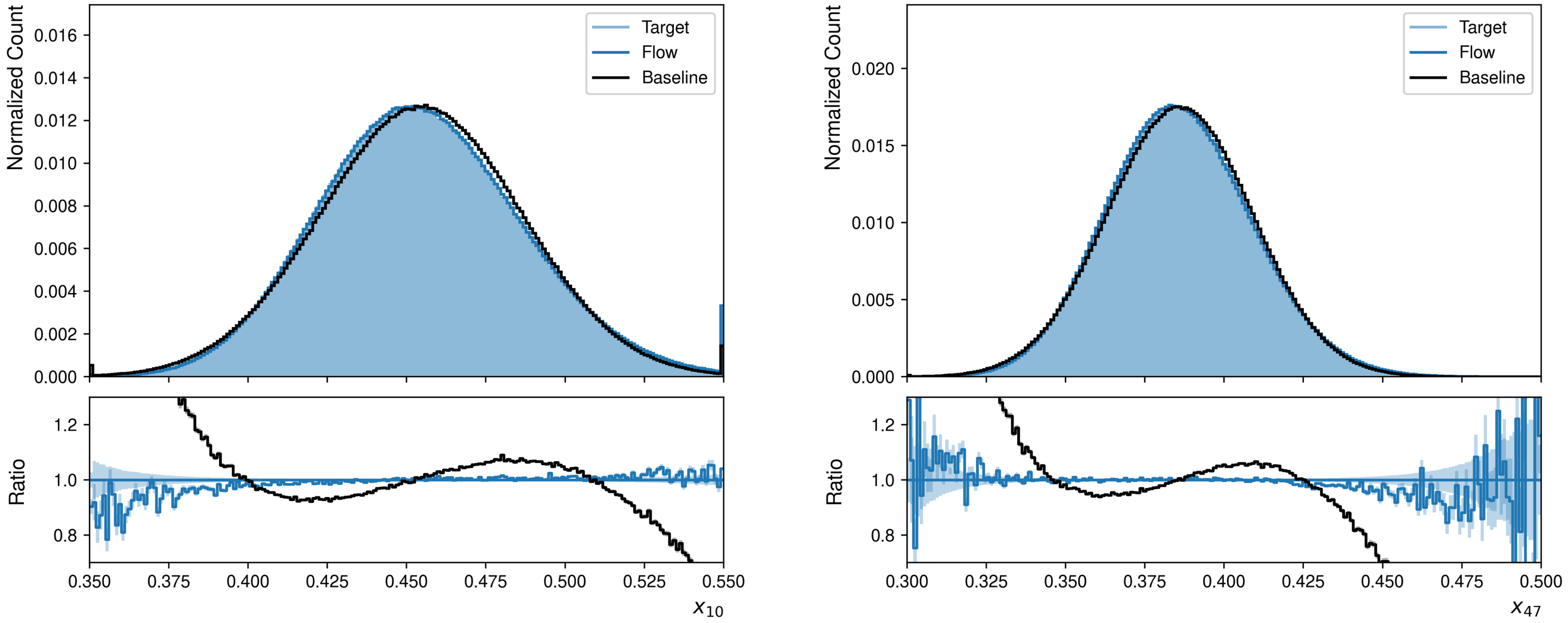
Compare it to the estimation of the flow model, and with the gaussian baseline.

With real data we can't do this - if we knew the true pdf we would not have to do this at all.



Moderate case

Marginal Plots

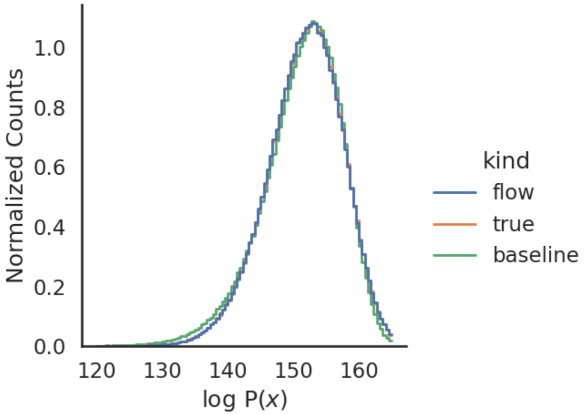


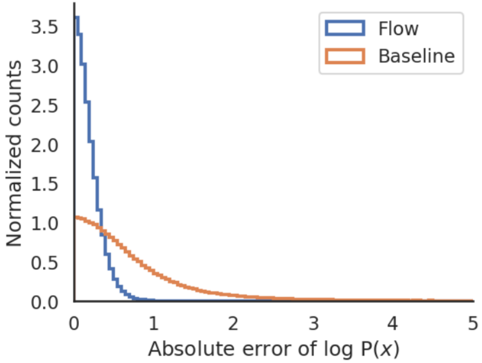
Two example bins of the 49-dimensional distribution.

Log Probability

Since we constructed the distribution from which the toys are drawn we can evaluate the true probability for each toy.

Compare it to the estimation of the flow model, and with the gaussian baseline.

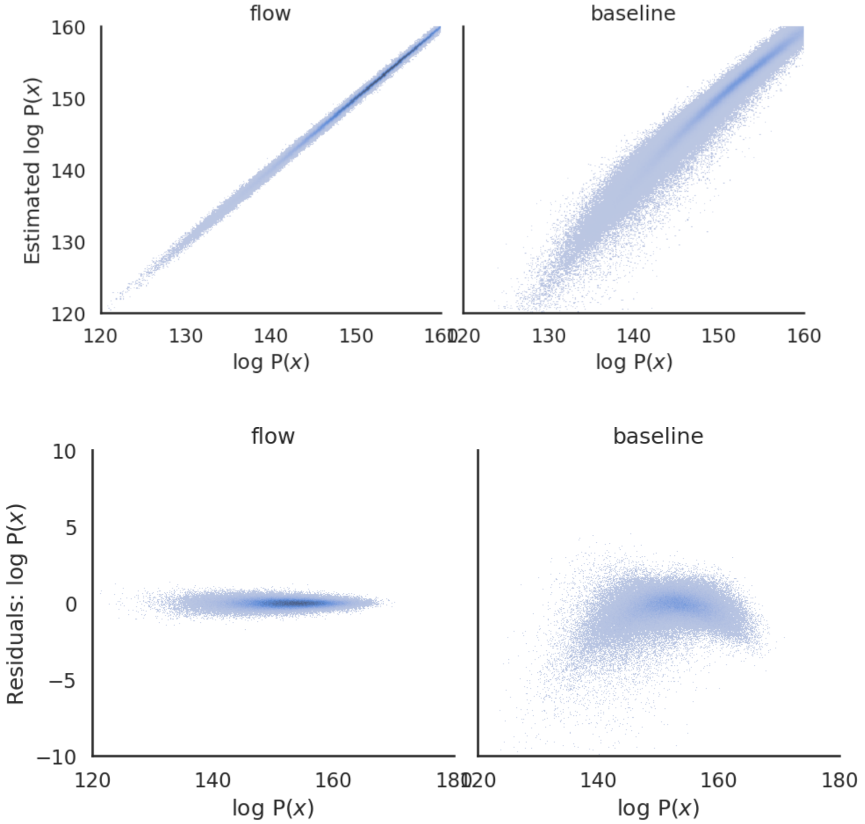




Log Probability

Since we constructed the distribution from which the toys are drawn we can evaluate the true probability for each toy.

Compare it to the estimation of the flow model, and with the gaussian baseline.

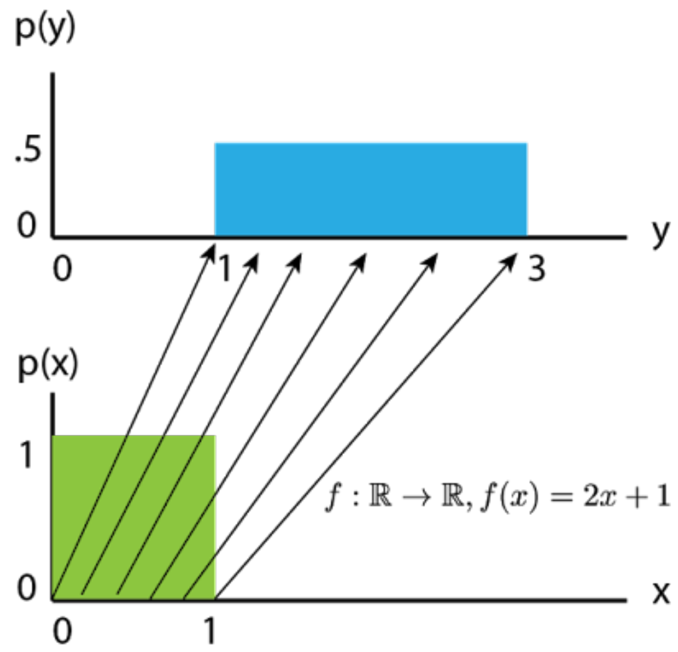


Change of variables formula

Tracks how the probability volume changes as we apply a transformation and ensures it stays normalised

$$f(x) = y$$

$$p(y) = p(x) \det \frac{\partial f(x)}{\partial x}^{-1}$$

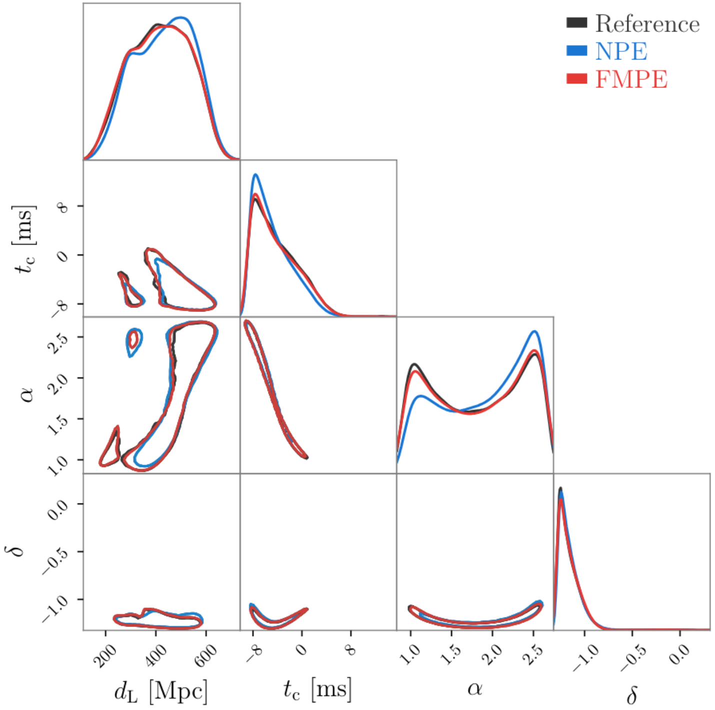


Posterior Estimation

$$p(\Theta|x)$$

Normalizing flows can be used to infer the posterior distribution over **model parameters** given **observed data**.

Use a conditional normalising flow trained on pairs of observations and model parameters.



Gravitational Wave Inference [Dax et al. 2024]