Unsupervised Learning Likelihood functions of LHC results

UNIVERSITÀ DEGLI STUDI DI GENOVA

INFN GENOVA
Istituto Nazionale di Fisica Nucleare
Sezione di Genova

Humberto Reyes-González
University of Genoa

université PARIS-SACLAY
INSTITUT PASCAL

Learning to Discover 2022
Orsay, France, 29/4/2022

Introduction

- **Likelihood functions** parametrise the full information of an LHC analysis; wether it is New Physics (NP) search or an SM measurement.

see Nicholas Berger talk 21/4/22

- Their **preservation** is a key part of the **LHC legacy.**

Usage:
- Resampling
- Reinterpretation with different statistical approaches.
- Reinterpretation in the context of different NP models.
- …

Challenges:
- LHC likelihoods are often high-dimensional complex distributions.
- We want precise descriptions that can be efficiently reinterpreted.

Current steps forward:
- ATLAS started publishing full likelihoods of NP searches ATL-PHYS-PUB-2019-029.
- Release of the pyhf package to construct statistical models 10.21105/joss.02823, L Heinrich, M Feickert, G Stark
- Theorists have started profiting from this arXiv:2009.01809, arXiv:2012.08192, SModelS collaboration
- Supervised learning with DNN likelihood arxiv:1911.03305 A Coccaro, M. Perini, L Silvestrini, R Torre

Our approach: Unsupervised Learning with Normalizing Flows

LHC likelihoods in a nutshell

really, see Nicholas Berger's talk 21/4/22

Full Statistical model:

$$P(\boxed{\mu}, \boxed{\theta}; \text{data}) = \Pi_{k=1}^{n_c} P[\boxed{n_i}; \boxed{\mu}\epsilon_{i,k}(\boxed{\vec{\theta}})N_{S,i,k}(\boxed{\vec{\theta}}) + B_{i.k}(\boxed{\vec{\theta}})]\Pi_{j=1}^{n_{syst}} G(\boxed{\theta_j^{obs}}; \boxed{\theta_j}; 1)$$

(Observed) data

Nuisance parameters (uncertainties)

(Auxiliary) data

Parameters of Interest (signal strength)

Test Statistic:

$$t(\mu) = -2\log\frac{L(\mu; \hat{\theta}(\mu))}{L(\hat{\mu}, \hat{\theta})}$$

Probability of data given a certain μ

Conditional best-fit $\theta(\mu)$

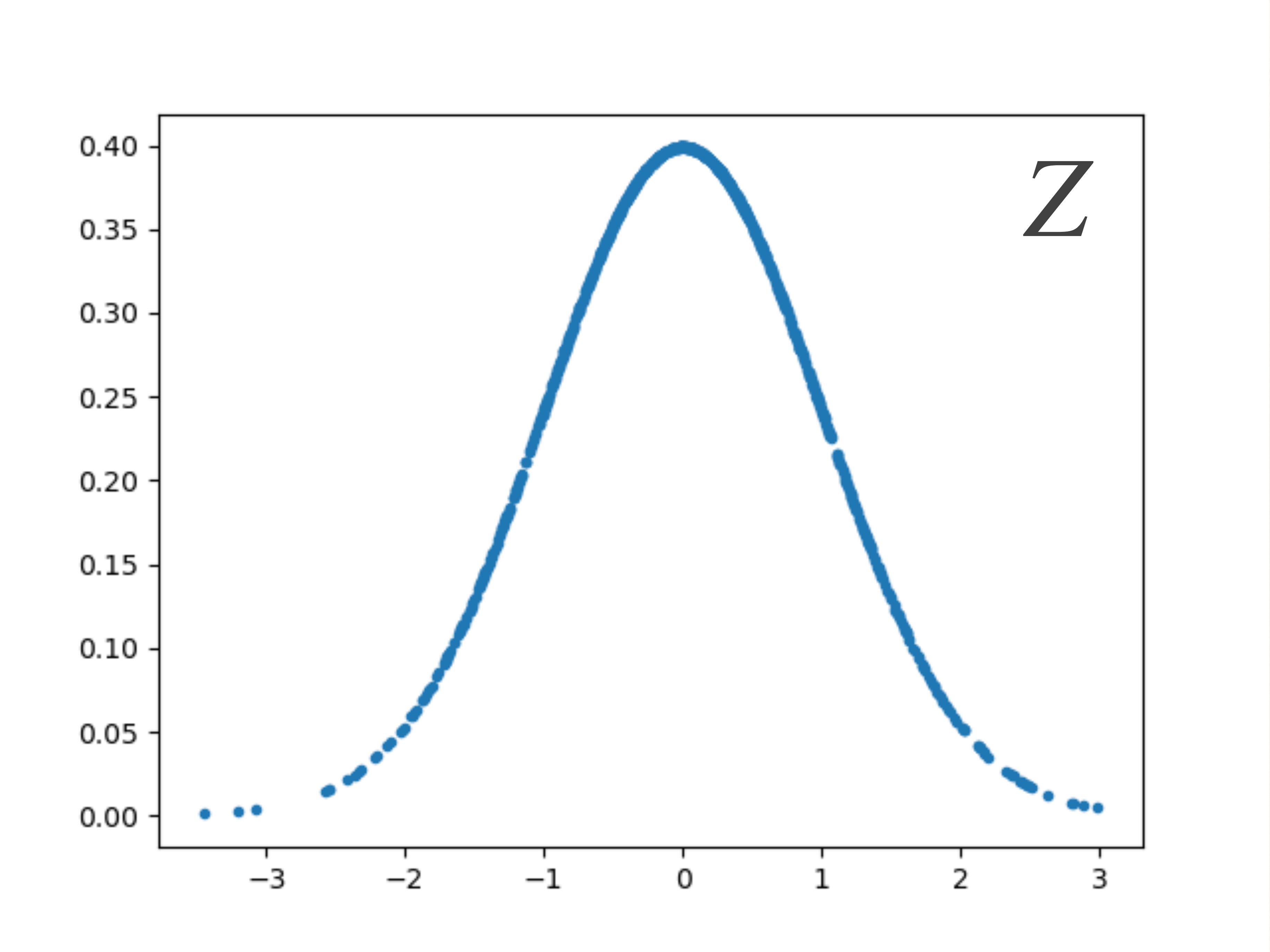
Probability of data given $\hat{\mu}$ (MLE)

best-fit θ

Introducing Normalizing Flows.

BASIC PRINCIPLE:

Following the change of variables formula, perform a series of **bijective, continuous, invertible** transformations on a *simple* probability density function (pdf) to obtain a *complex* one.

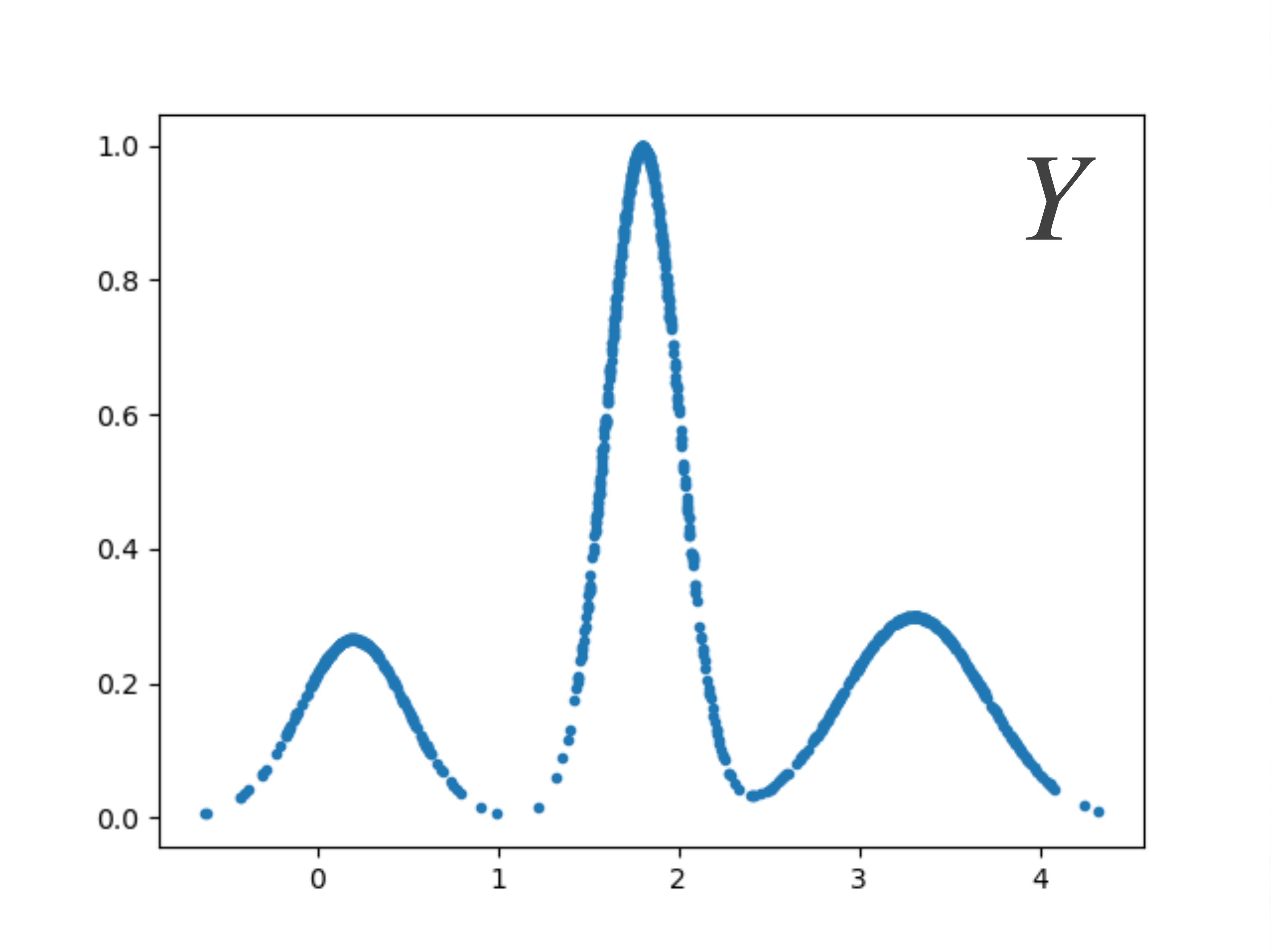


Normalizing direction

$Z = f(Y)$

Generative direction

$Y = g(Z)$



$$p_Y(y) = p_Z(f(y))\,|\det(Df(y))| = p_Z(f(y))\,|\det(Dg(f(y))|^{-1}$$

See review
Ivan K. et. al.
arXiv:1908.09257

Choosing the transformations

THE OBJECTIVE:

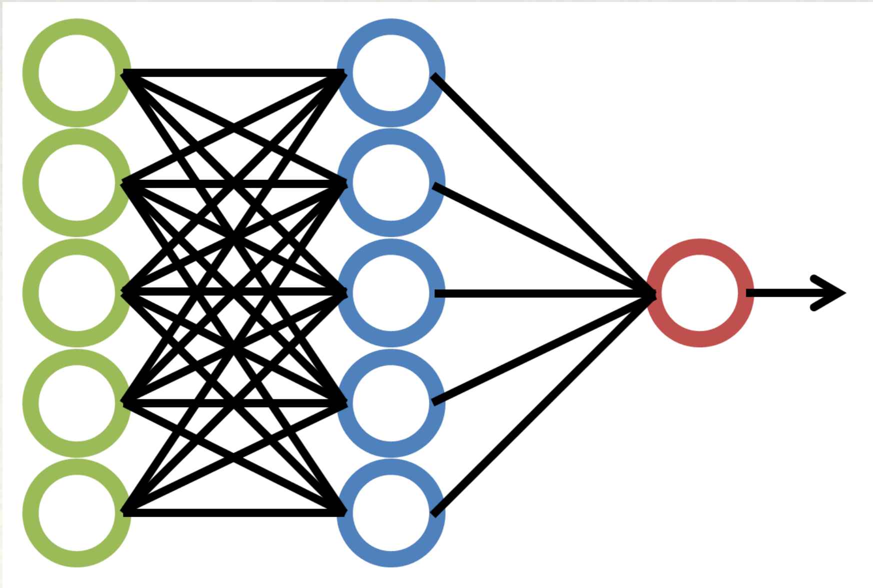
To perform the right transformations to accurately estimate the complex underlying distribution of some observed data.

THE RULES OF THE GAME:

- The transformations must be invertible
- They should be sufficiently expressive
- And computationally efficient (including Jacobian)

THE STRATEGY

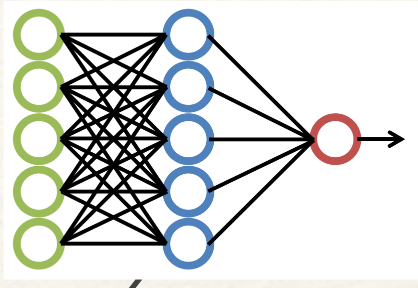
Let *Neural Networks* learn the parameters of *Autoregressive Normalizing Flows.*

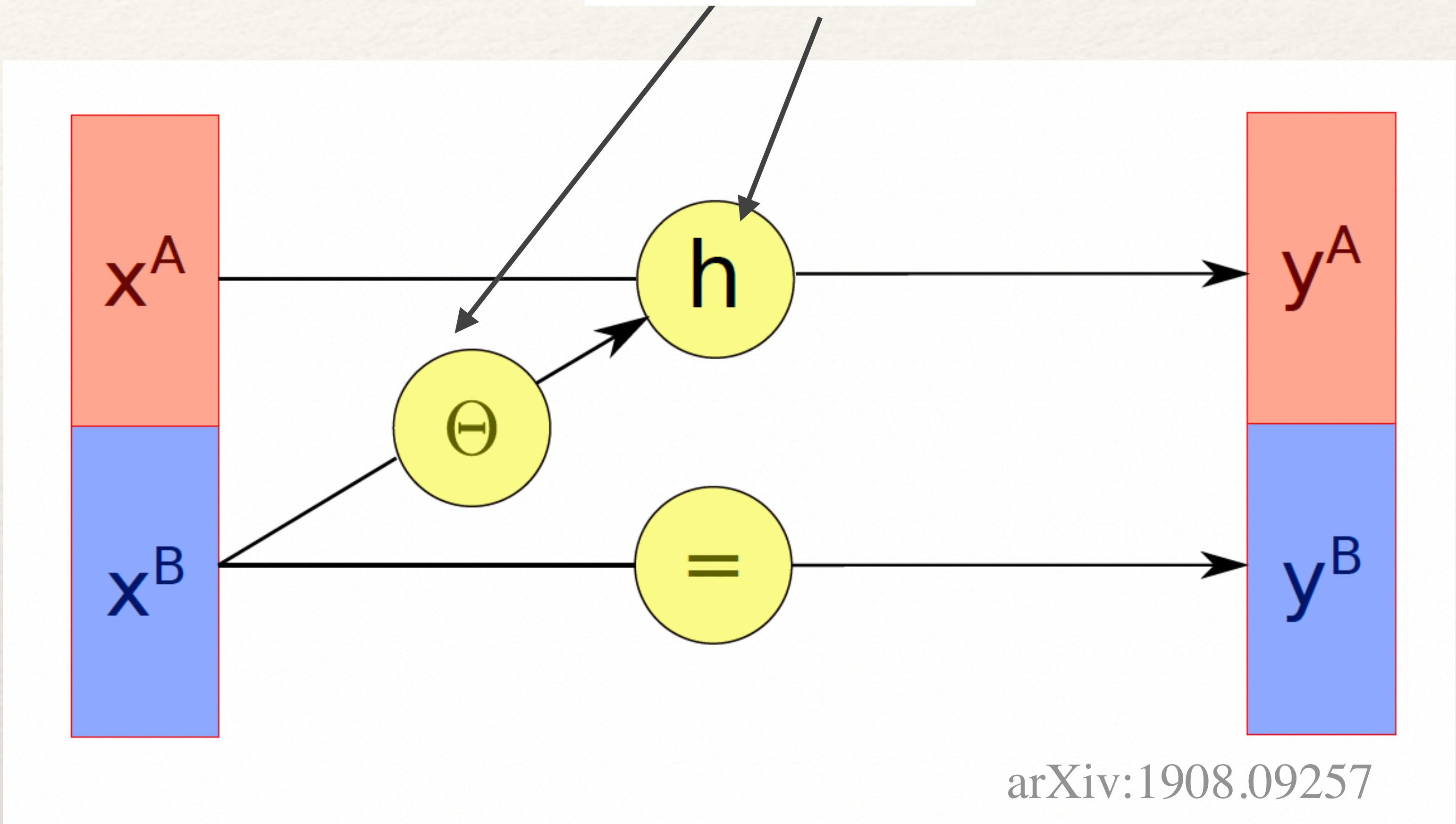


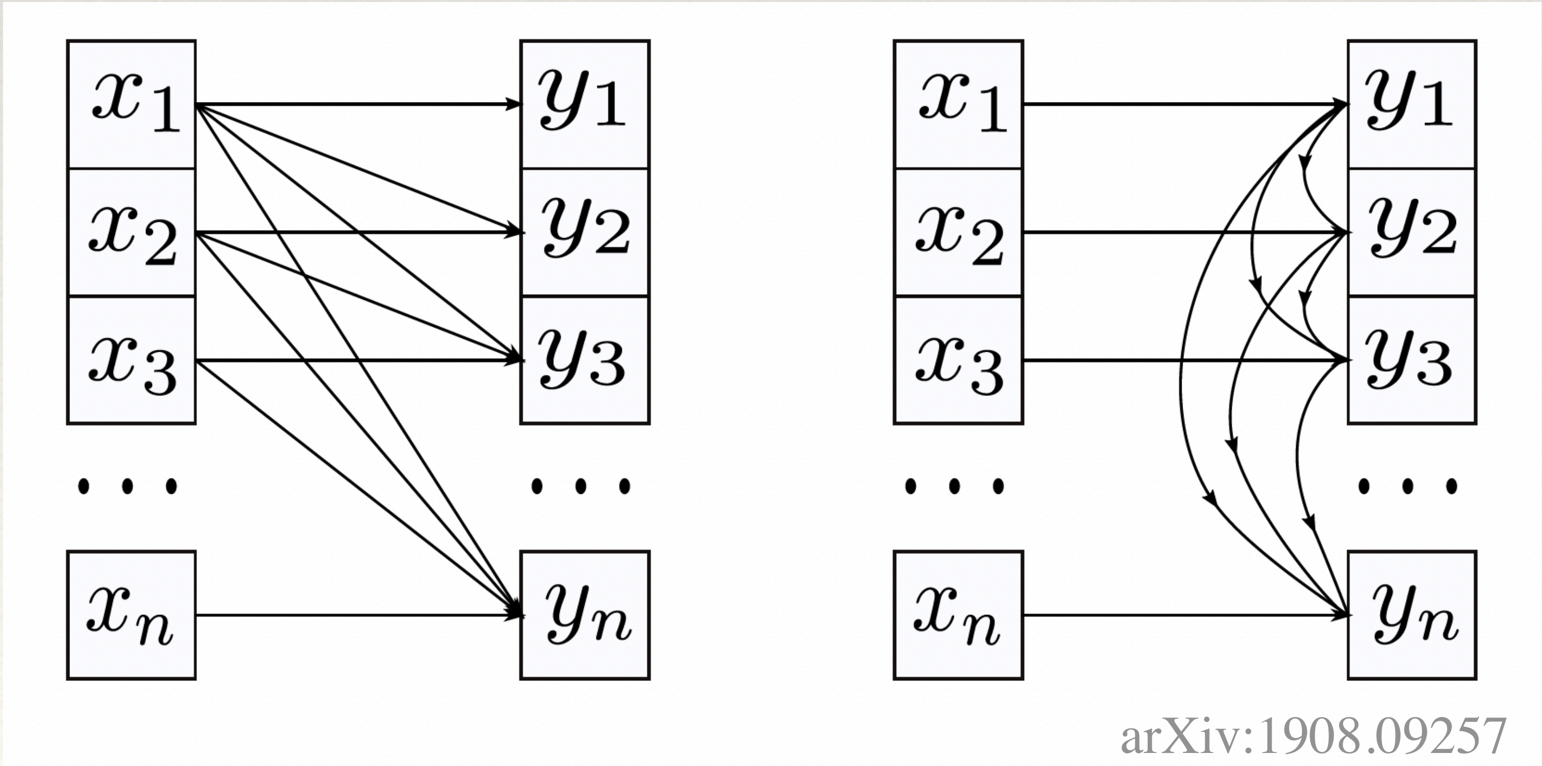
Autoregressive Flows

- Dimension x^i is transformed with bijectors trained with $y_{1:i-1}$
- Bijector parameters are trained with Autoregressive NNs.
- The Jacobian J is also triangular thus…
- **Jacobian is easily computed!**
- **Direct sampling OR density estimation.**
- **More expressive.**

The loss function:
$-\log(p_{AF}(real_{dist}))$





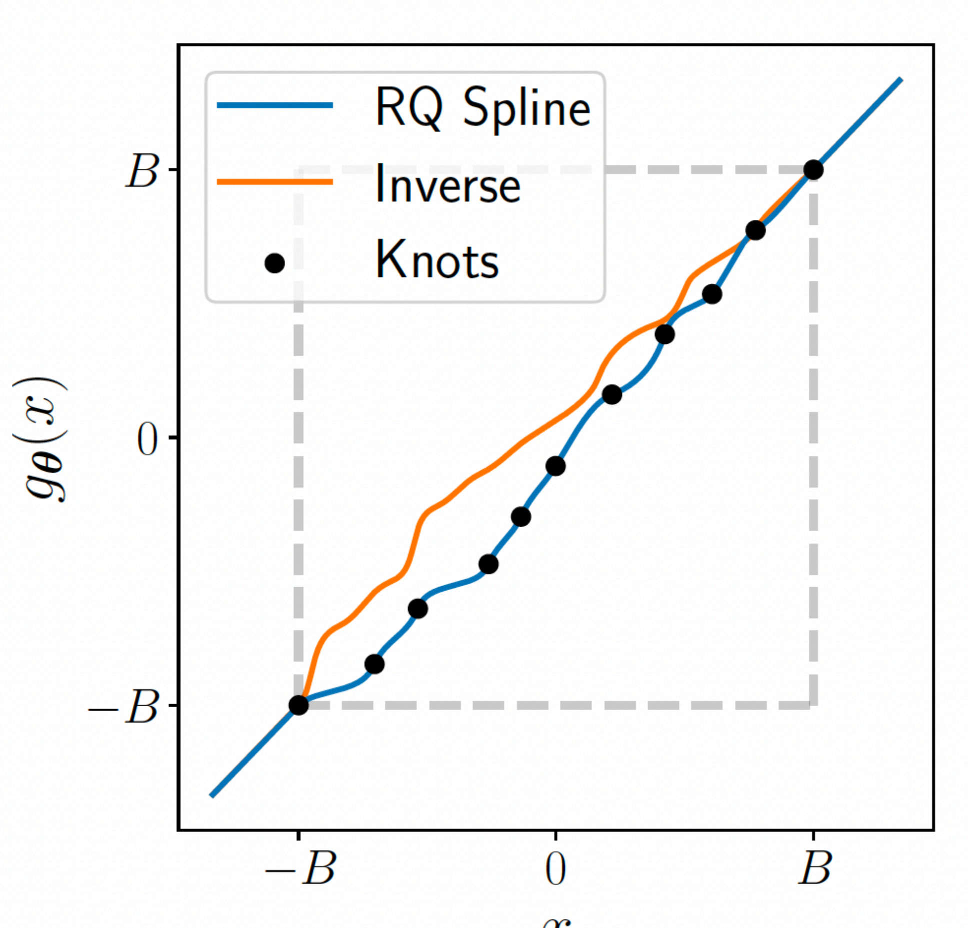


Autoregressive Flows

MAF	A-NSF
Masked Autoregressive Flow	Neural Spline Flows
arXiv:1705.07057	arXiv:1906.04032

Affine ← **BIJECTORS** → Rational Quadratic Spline

$$y(x; \mu, b) = \mu \cdot x + b$$



arXiv:1906.04032

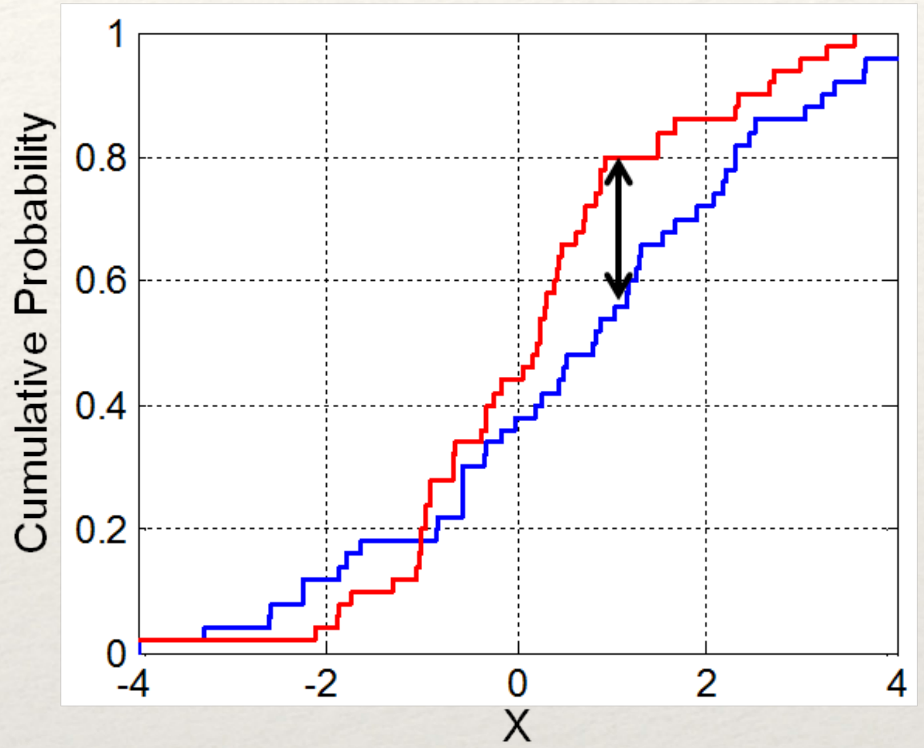
Measuring the flows accuracy

Non-parametric metrics

- Two-sample 1D Kolgomonov - Smirnov test (ks test):

$$D_{n,m} = \sup_x |F_n(x) - F_m(x)|$$

-Computes the p-value for two sets of 1D samples coming from the same *unknown* distribution.
-We average over ks test estimations and compute the median over dimensions.
-**Optimal value 0.5**

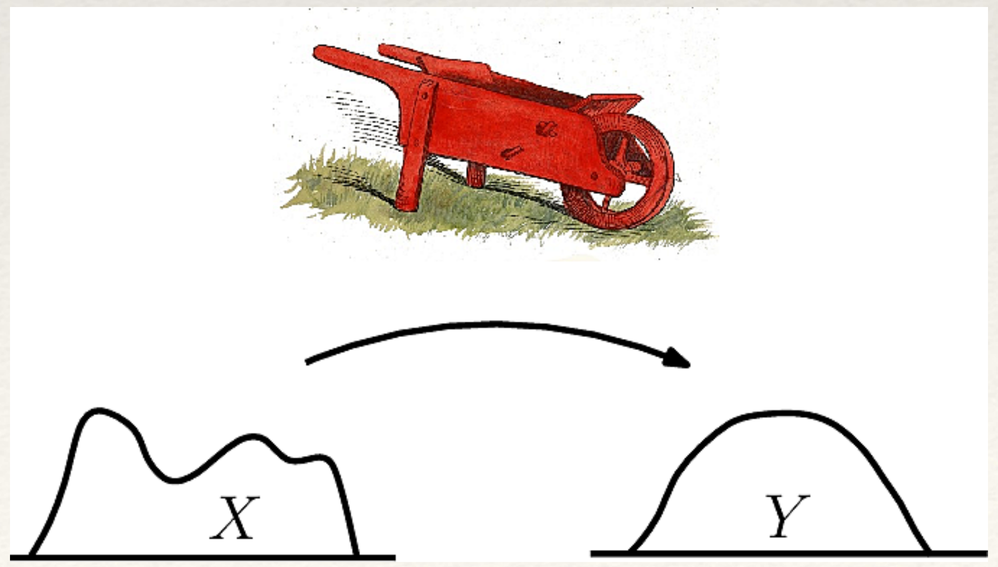


https://en.wikipedia.org/wiki/Kolmogorov-Smirnov_test

- 1D Wasserstein distance (Earth mover's distance)

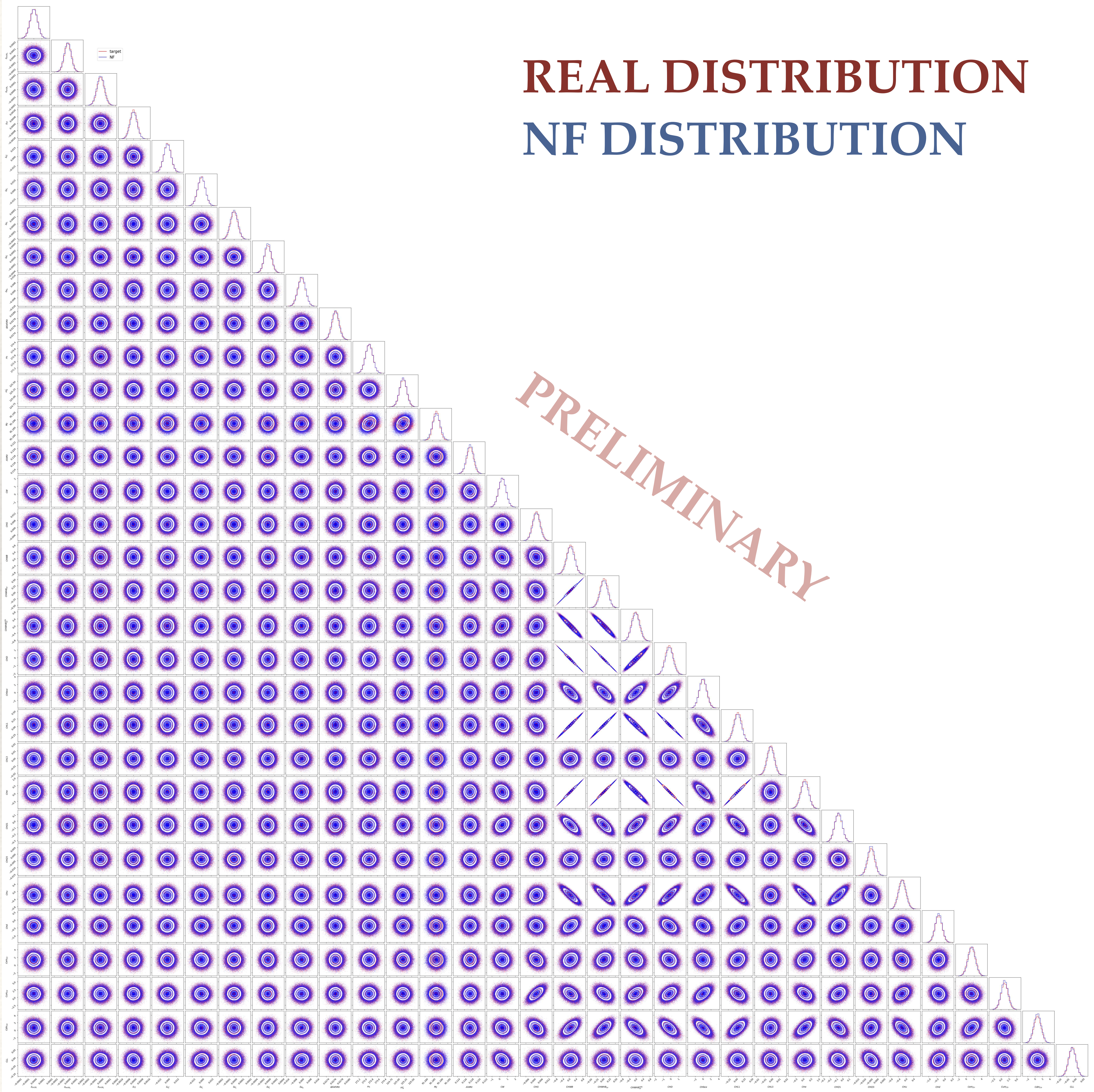
$$l(f_n, f_m) = \int_{-\infty}^{\infty} |F_n - F_m|$$

-Computes the minimum *energy* required to transform f_n into f_m
-We compute the median over dimensions.
-**Optimal value 0.0**



https://sbl.inria.fr/doc/Earth_mover_distance-user-manual.html

EW-fit (32 dims)



Likelihood of global EW-fit at LHC:
18 parameters of interest.
14 nuisance parameters.

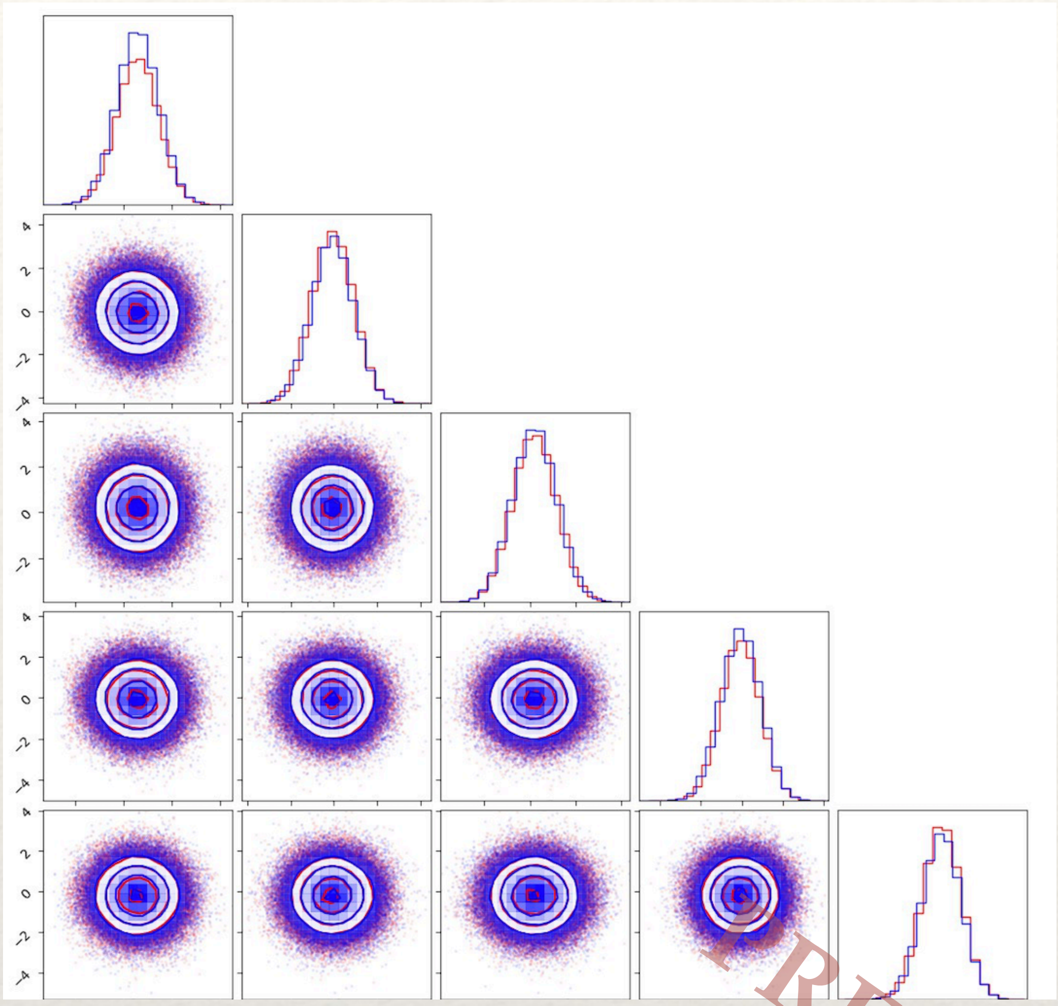
Data provided by authors -> arXiv:1710.05402

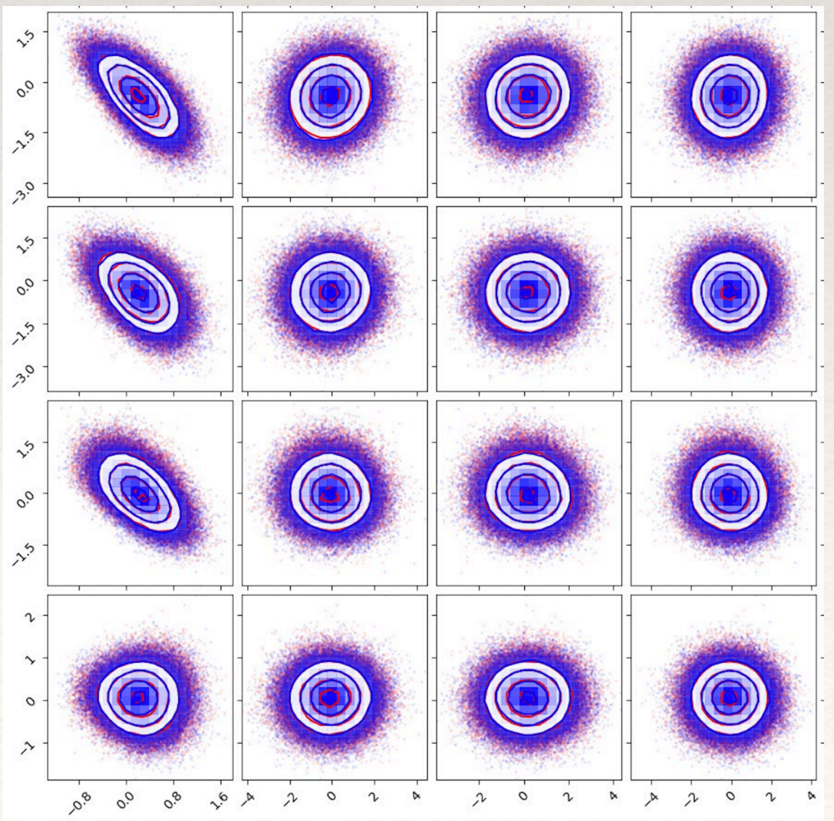
Weapon of choice:
MAF, 3 Bijectors, 128x3 layers, 650k samples

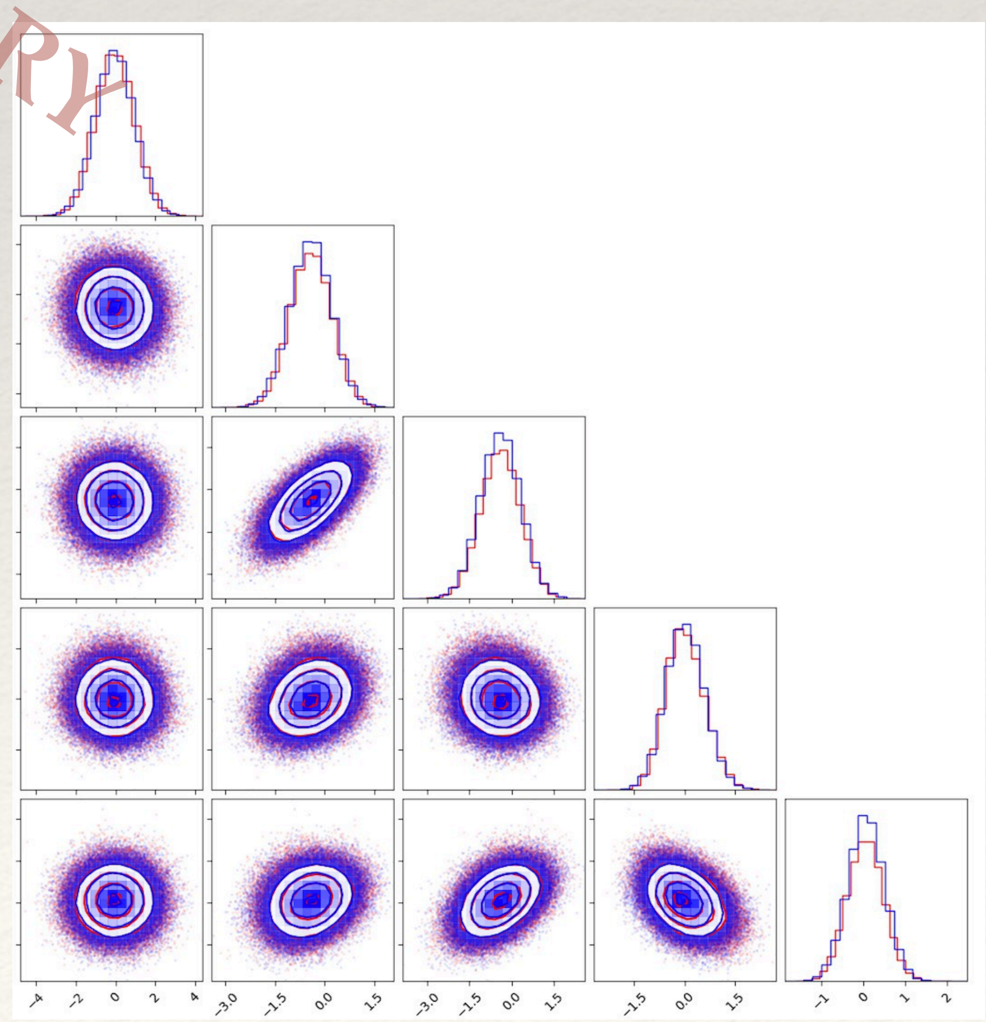
Metrics:
Wasserstein distance: .000315
KS test: 0.484
Training time: 2.8 hrs.

THE RESEMBLANCE IS GREAT!

LHC-like New Physics search (95 dims).







REAL DISTRIBUTION
NF DISTRIBUTION

Likelihood of LHC-like New Physics search:
1 parameter of interest.
94 nuisance parameters.

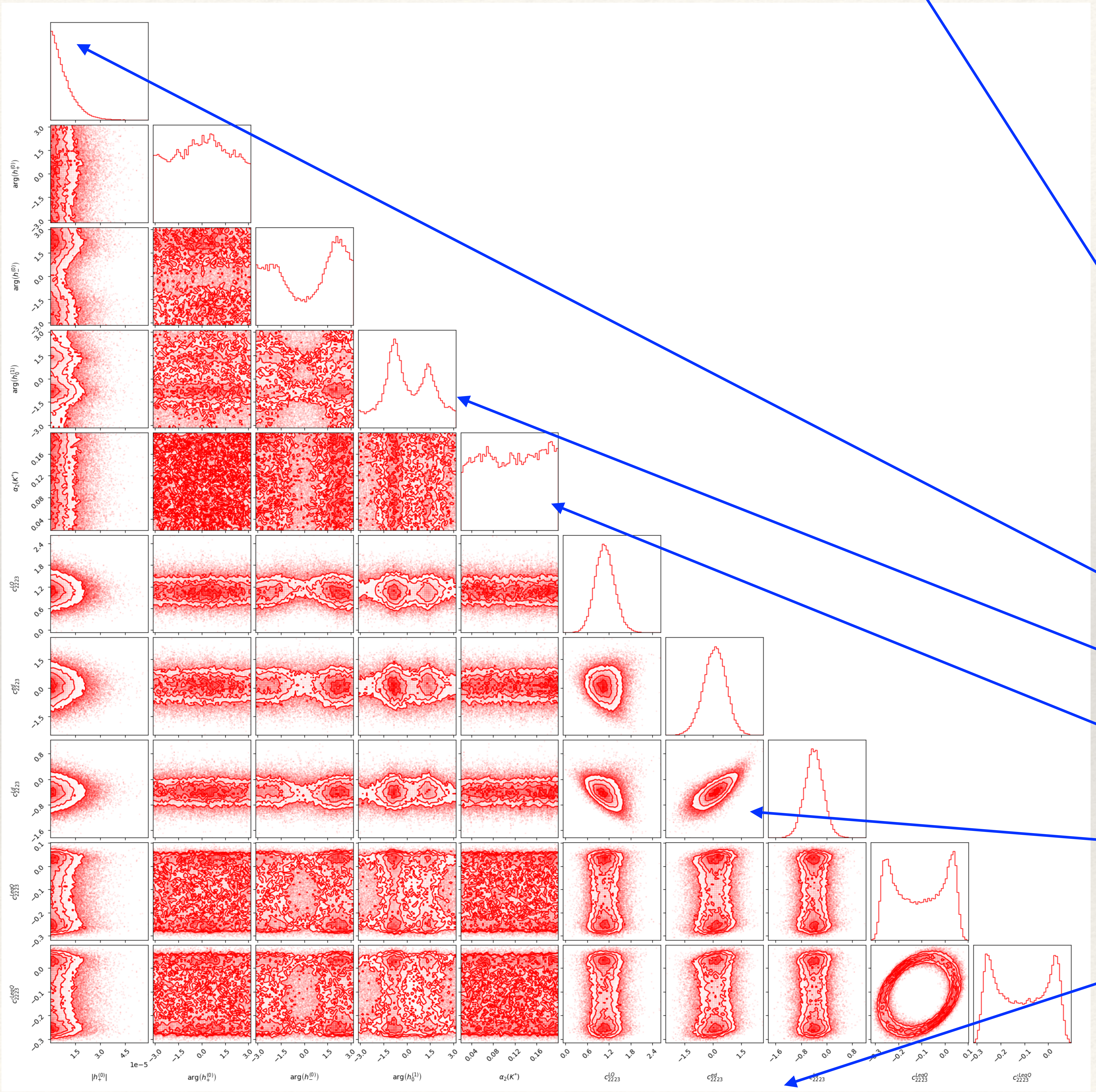
arXiv:1911.03305
arXiv:1809.05548

Weapon of choice:
MAF, 3 Bijectors, 128x3 layers, 500k samples

Metrics:
Wasserstein distance: .0067
KS test: 0.507
Training time: 9.3 mins

ANOTHER GREAT RESEMBLANCE!

Flavor Likelihood (83 dims)



Likelihood of global-fit of $b \to sl^+l^-$ *transitions*:*

6 parameter of interest.

77 nuisance parameters.

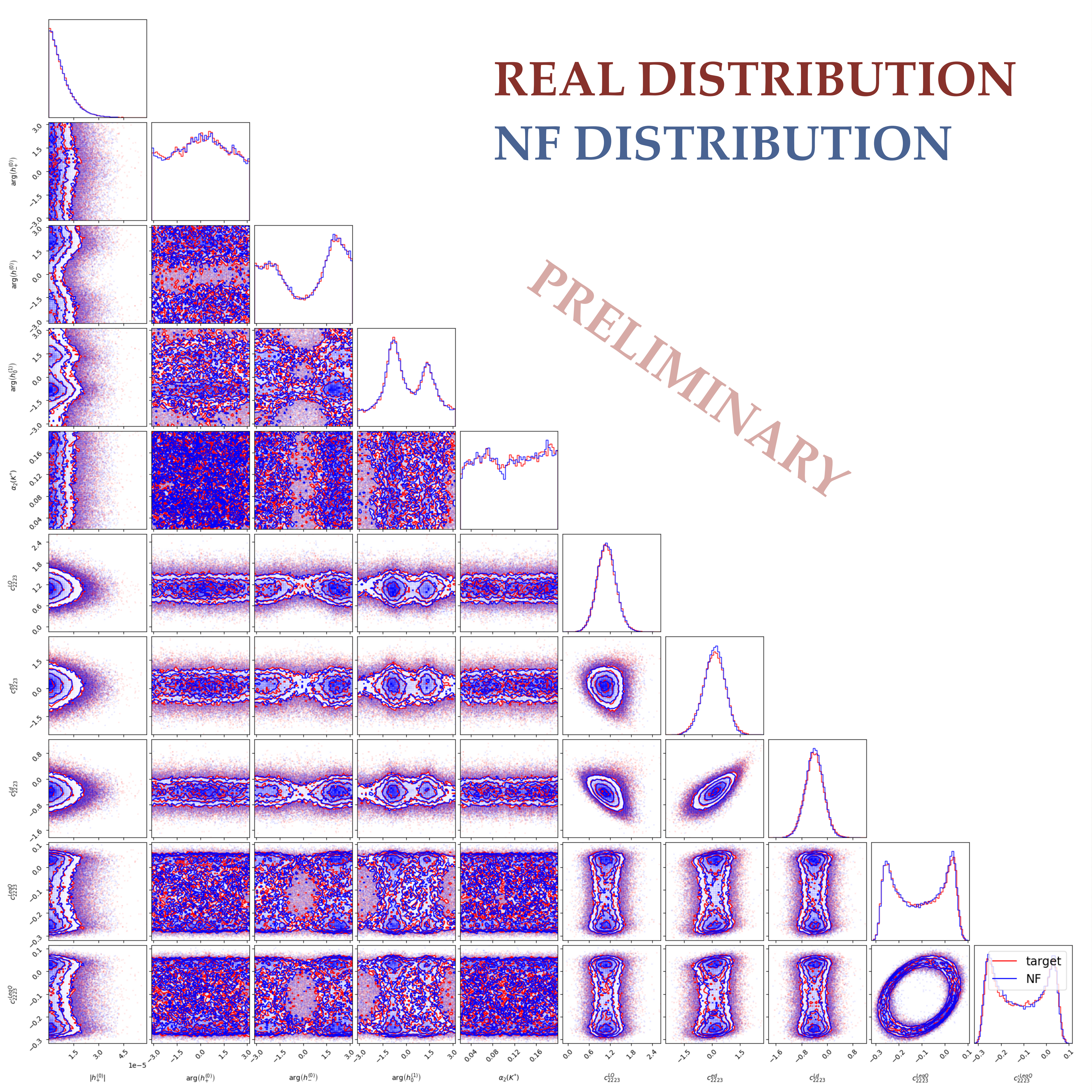
Data provided by authors of 10.1140/epjc/s10052-019 -7210-9

CHALLENGES:

- High-dimensionality
- Truncated distributions
- Multimodal dimensions
- “Noisy” dimensions
- Correlations
- Wide/different ranges

*Flavor anomalies

Flavor Likelihood (83 dims)



Weapon of choice:

A-NSF, 16knots

2 Bijectors

1024x3 layers

1.5M samples

Regulariser : l1, epsilon 1e-4

Metrics:

Wasserstein distance: .00027

KS test: 0.497

Training time: ~ 1.5 days

ALSO VERY GOOD

Conclusions

- The preservation of LHC likelihoods is of uttermost importance (for theorists also).
- Introduced unsupervised learning of full likelihoods with Normalizing Flows.
- Including high-dimensional very complex functions.

Outlook

- Push the accuracy a bit more.
- Release all results and codes (write the actual paper)
- Integrate into the DNN Likelihood framework; sample, build models, analize, plot …
- Systematic learning of LHC likelihoods.
- Interface with reinterpretation tools, e.g. SModelS.
- General study of Autoregressive Flows performance at high dimensions and how to measure it.

Thank you