UNIVERSITÀ DEGLI STUDI
DI GENOVA

INFN GENOVA
Istituto Nazionale di Fisica Nucleare
Sezione di Genova

Machine Learning LHC Likelihoods.

Humberto Reyes-González
University of Genoa

LHC reinterpretation workshop
CERN, 14/12/2022

Introduction

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

Important 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

Bayes theorem:

$$P(\Theta, x) = P_x(x\,|\,\Theta)\pi_\Theta(\Theta) = P_\Theta(\Theta\,|\,x)\pi_x(x)$$

LHC Statistical model:

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

(Observed) data

Nuisance parameters (uncertainties)

(Auxiliary) data

Parameters of Interest (signal strength, observables, etc.)

Test Statistic:

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

Best- fits:

$$L(\hat{\mu};\hat{\theta})$$

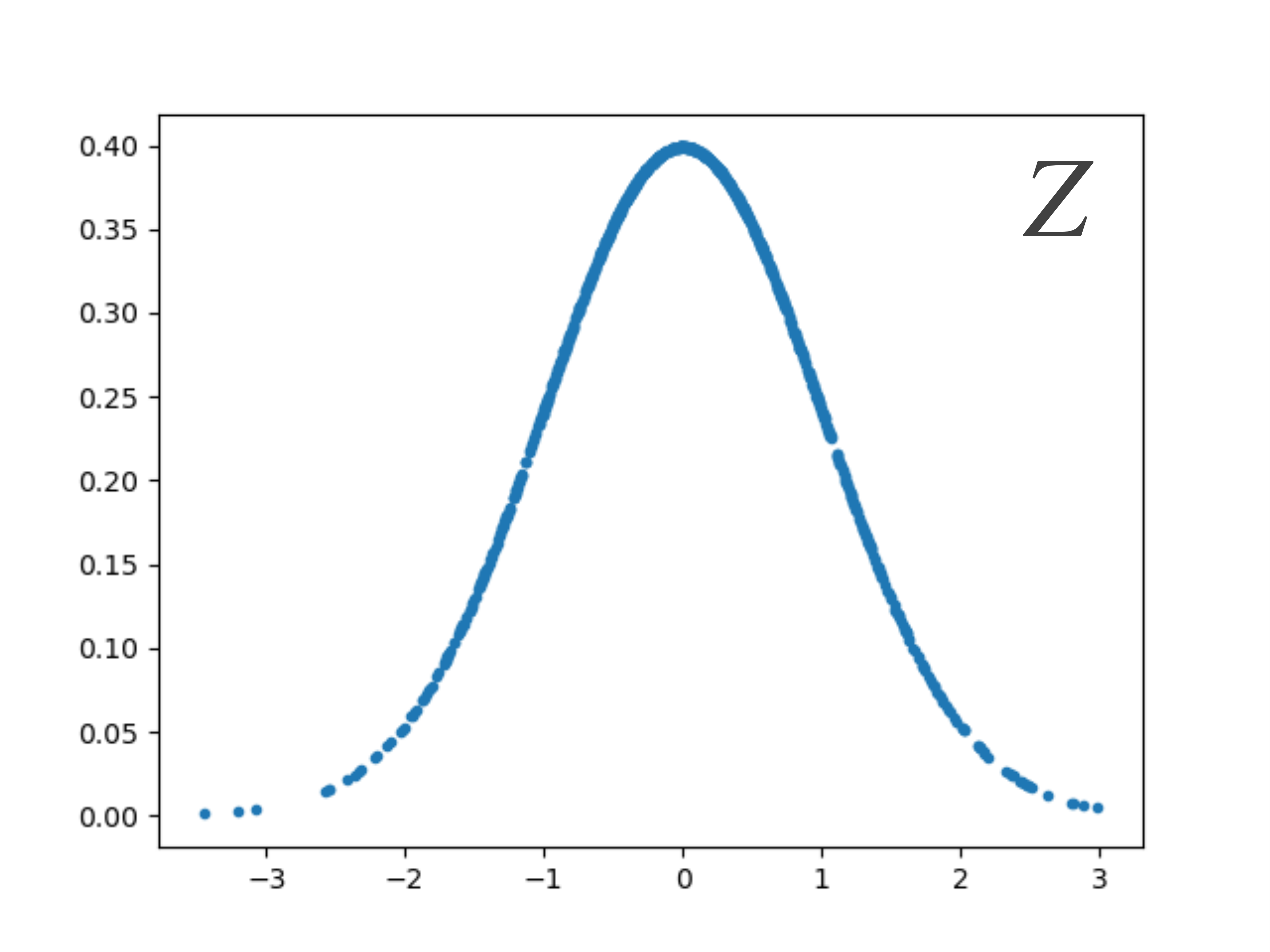
Where μ are observables

best-fit $\theta(\mu)$

Introduction.

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

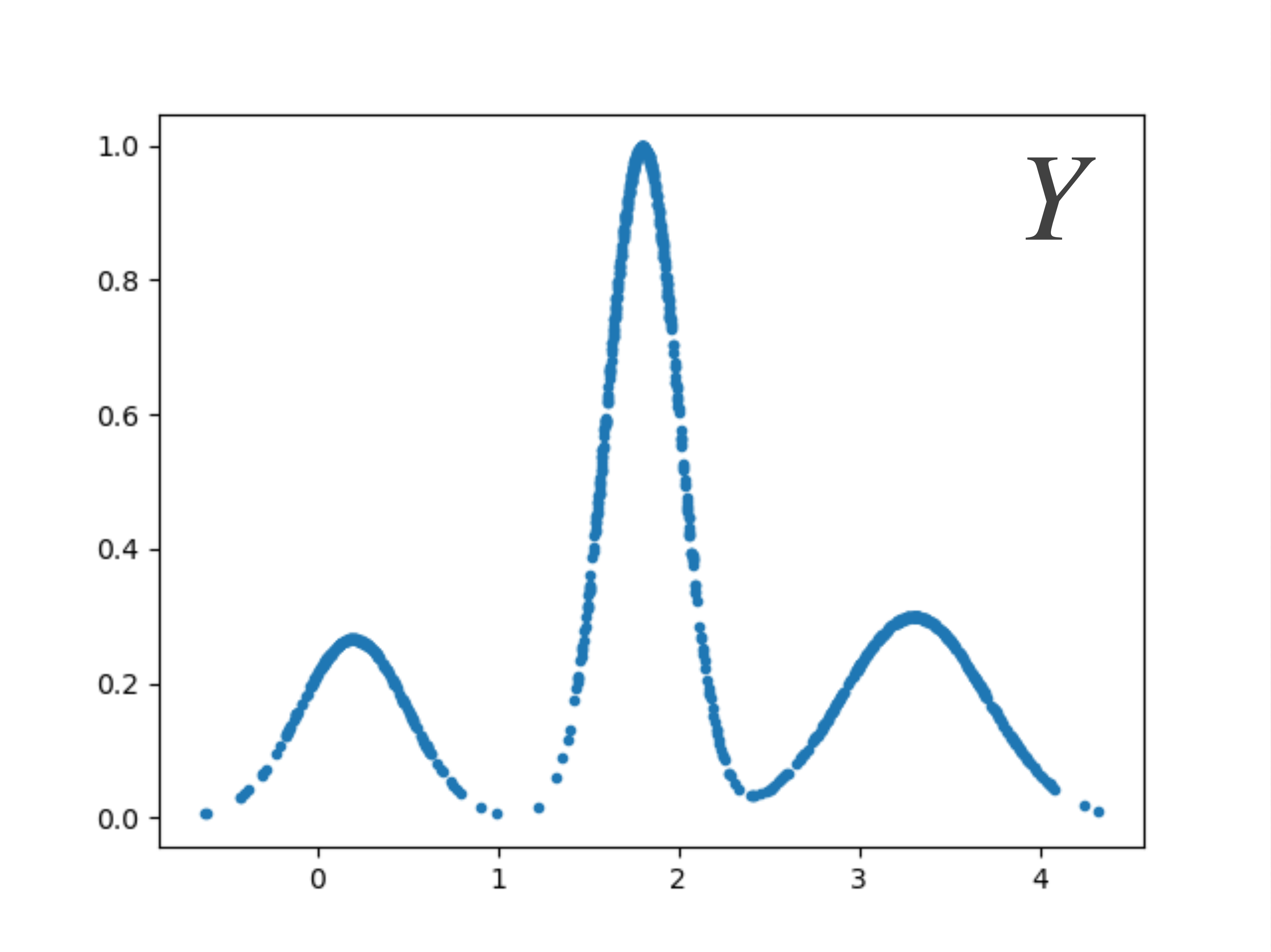
$\longleftarrow$

$Z = f(Y)$

Generative direction

$\longrightarrow$

$Y = g(Z)$



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

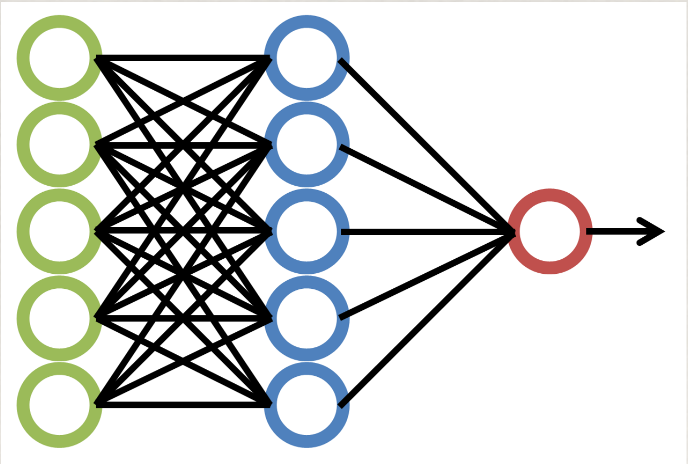
Choosing the transformations

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

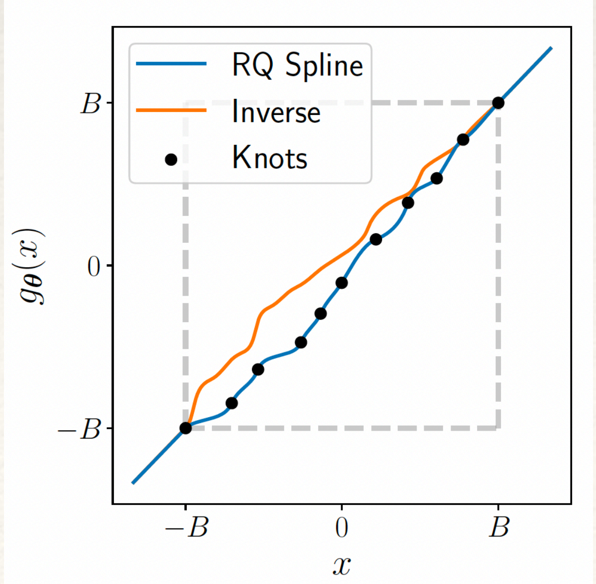
THE RULES OF THE GAME:
- The transformations (bijectors) must be invertible
- They should be sufficiently expressive
- And computationally efficient (including Jacobian)

THE STRATEGY

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

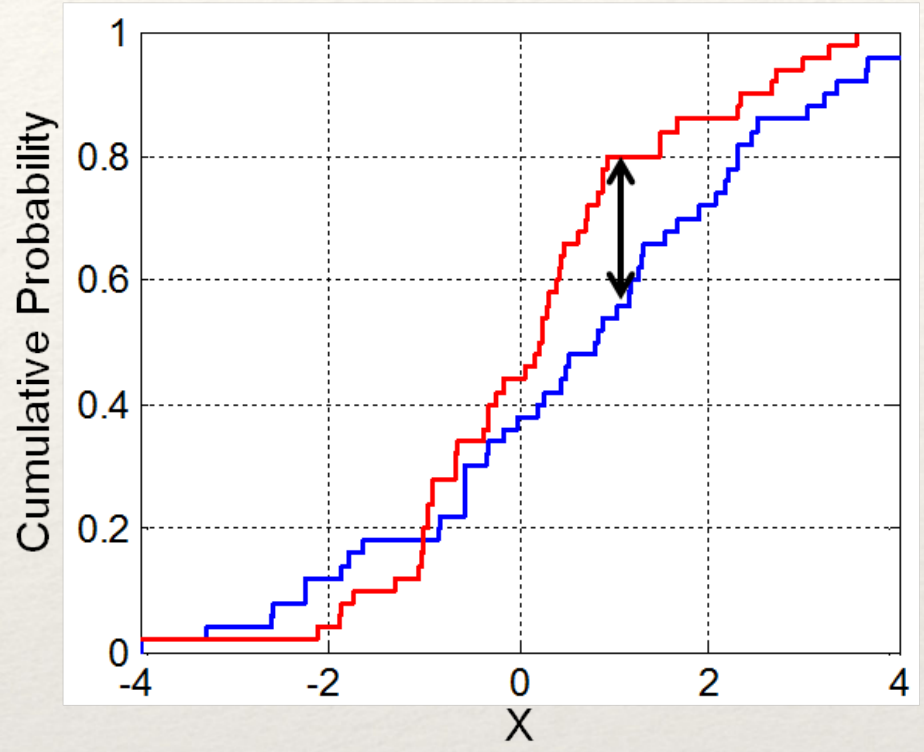


Autoregressive Rational-Quadratic-Spline Flows
(A-RQS, arXiv:1906.04032)



Evaluation metric.

- 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

Example Likelihoods

$$P_\Theta(\Theta\,|\,x = \text{obs})$$

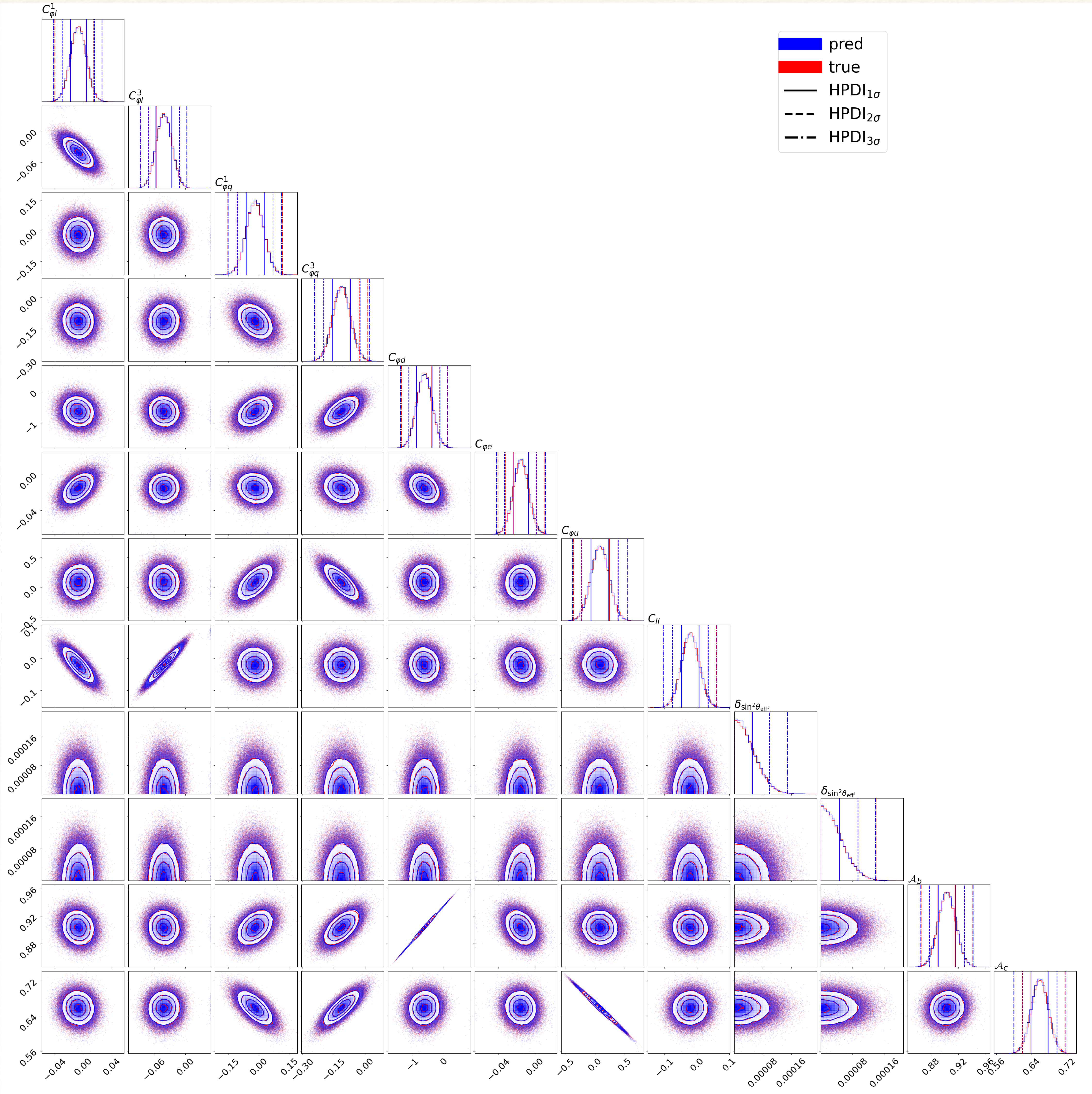
ElectroWeak fit Likelihood

- EW observables.
- Including recent measurements of top mass (CMS) and W mass (CDF).
- 8 parameters of interest (Wilson coefficients of SMEFT operators)
- 32 nuisance parameters.
- Ref. arXiv:2204.04204

Flavor fit likelihood.

- Flavor observables related to $b \to s l^+ l^-$ transitions
- 12 parameters of interest (Wilson coefficients of SMEFT operators)
- 77 nuisance parameters.
- Ref. arXiv:1903.09632

ElectroWeak fit Likelihood



Hyperparameters:

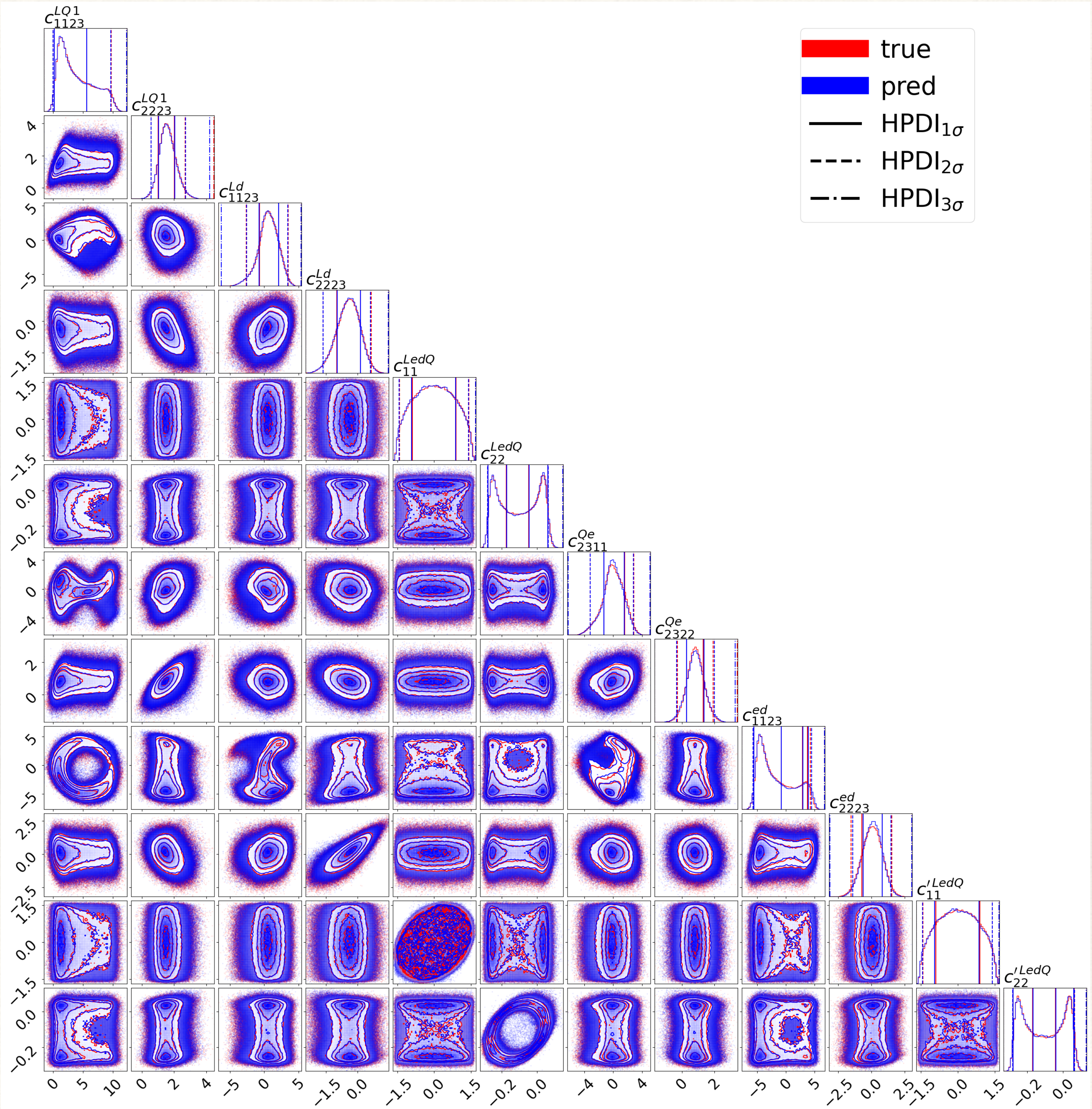
N_{train}	Flow	N bij	N knots	Range	Hidden layers	L1 factor	N epochs	N iters.
$\mathbf{10^4}$ $\mathbf{10^5}$ $\mathbf{2 \cdot 10^5}$	A-RQS	2	6	[-6,6]	128×3	0	200	12

Results:

N_{train}	KS-test
10^4	0.453
10^5	0.4803
$2 \cdot 10^5$	0.486

Test sample : 300k

Flavor fit Likelihood



Hyperparameters:

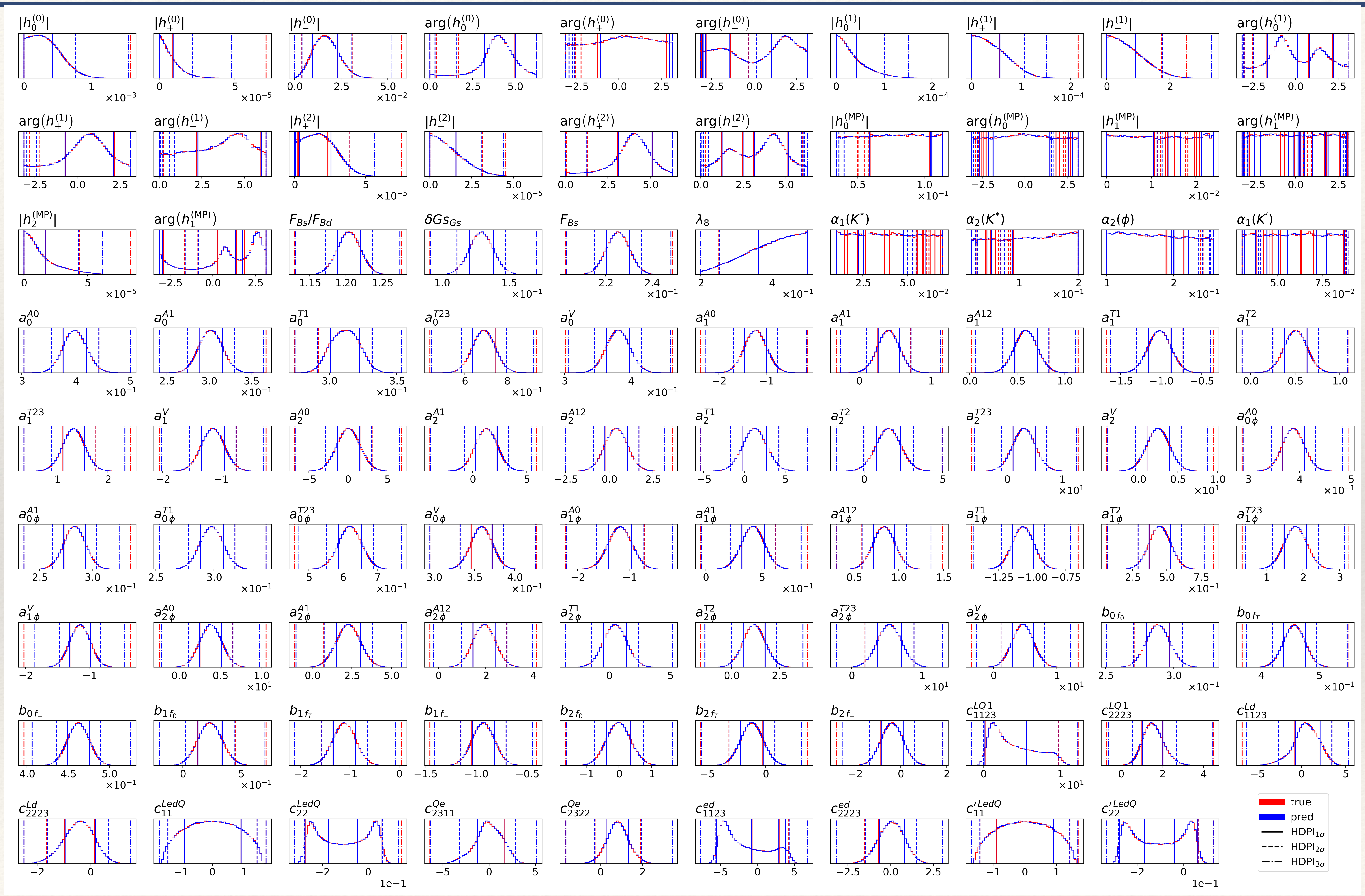
N_{train}	Flow	N bij	N knots	Range	Hidden layers	L1 factor	N epochs	N iters.
$\mathbf{10^5}$ $\mathbf{5 \cdot 10^5}$ $\mathbf{10^6}$	A-RQS	2	16	[-6,6]	1024×3	10^{-4}	1200	12

Results:

N_{train}	KS-test
$\mathbf{10^5}$	0.457
$\mathbf{5 \cdot 10^5}$	0.482
$\mathbf{10^6}$	.4806

Test sample : 500k

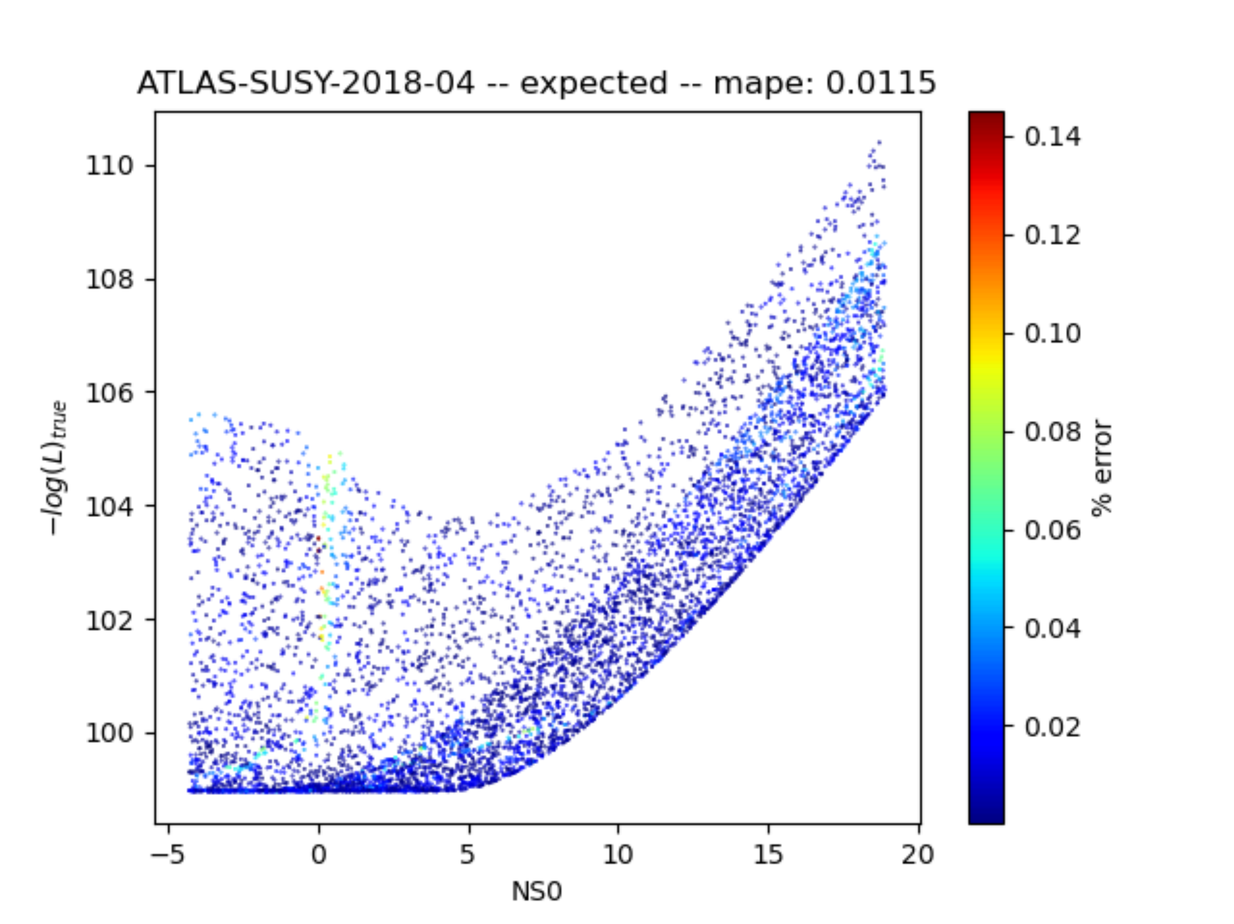
Flavor fit Likelihood

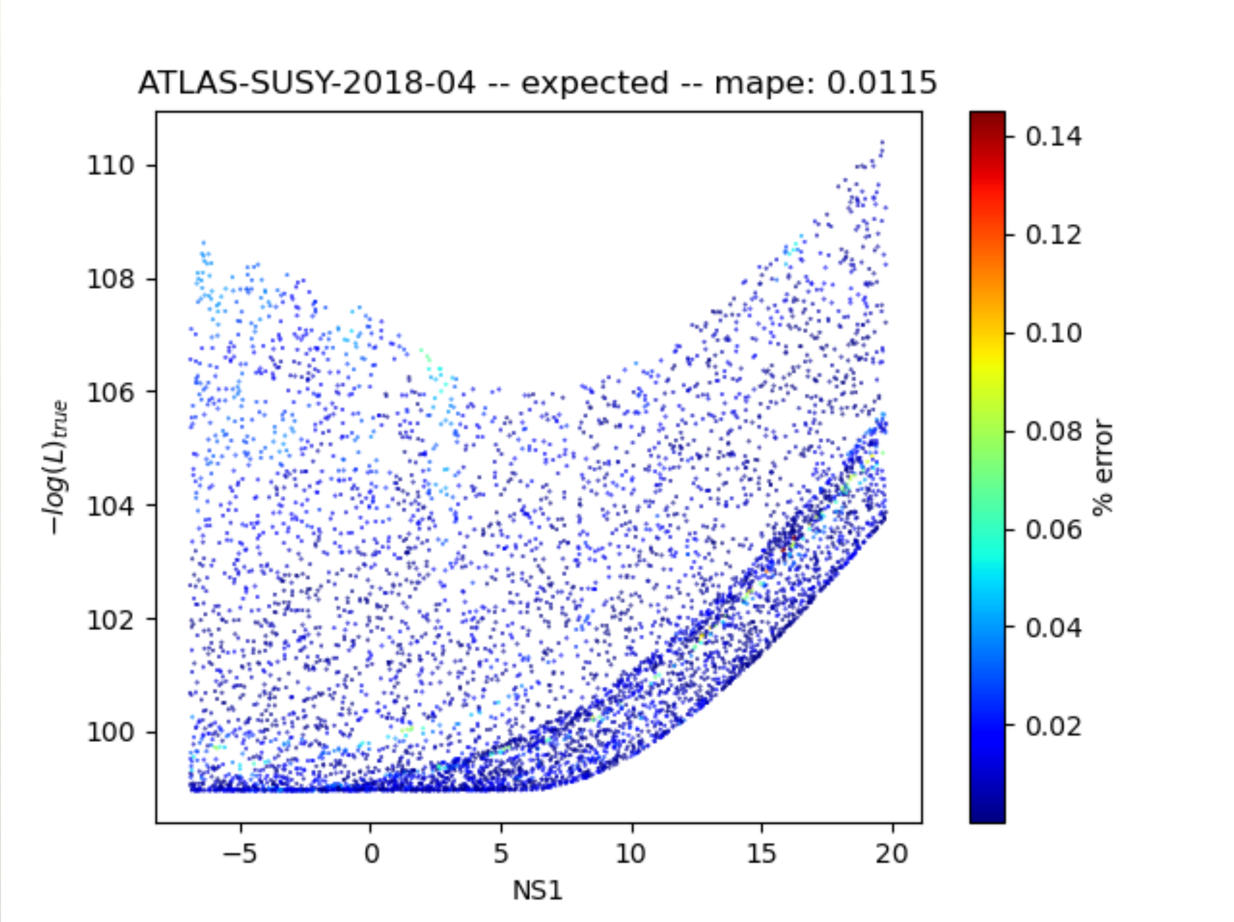


EXTRA: Supervised Learning Profiled Likelihoods

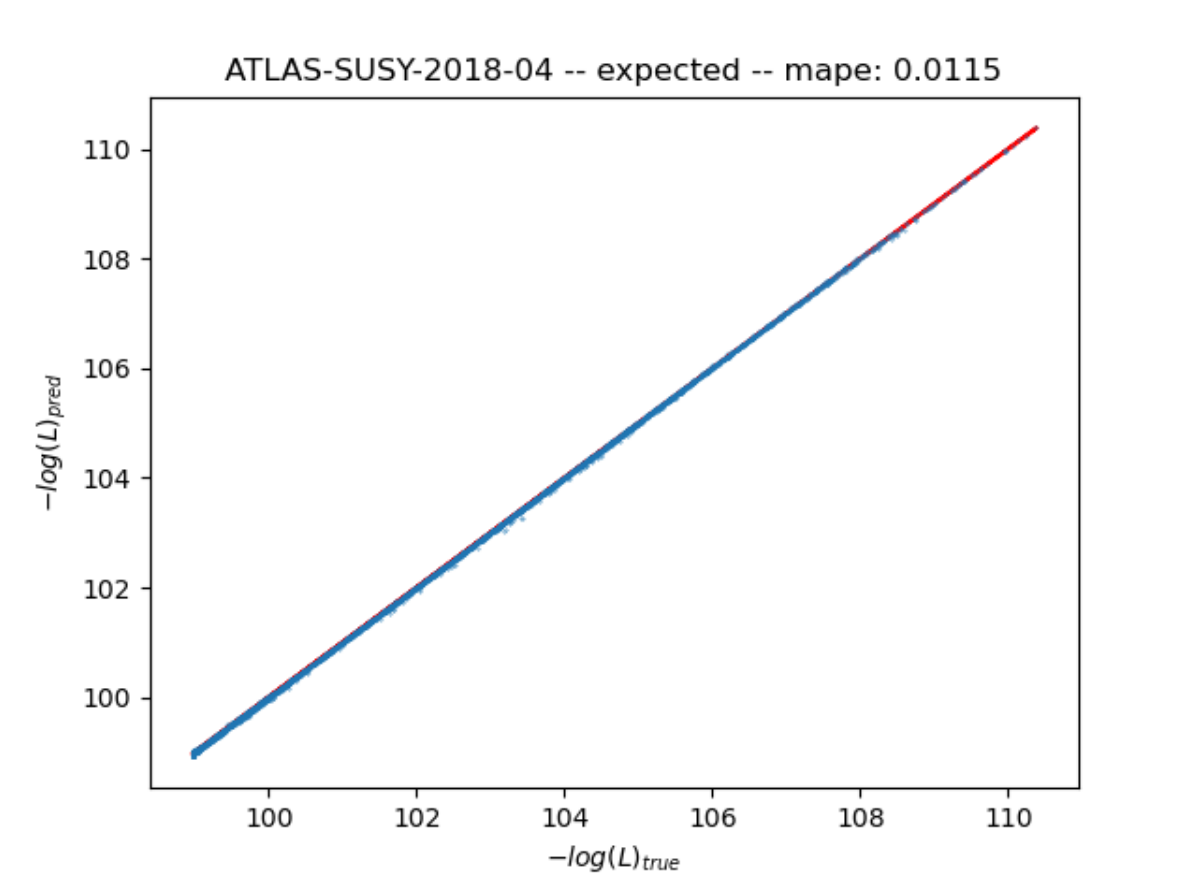
$$P_{profiled}(n_s) = P(n_S \,|\, \mu = 1, \hat{\theta}(\mu = 1))$$

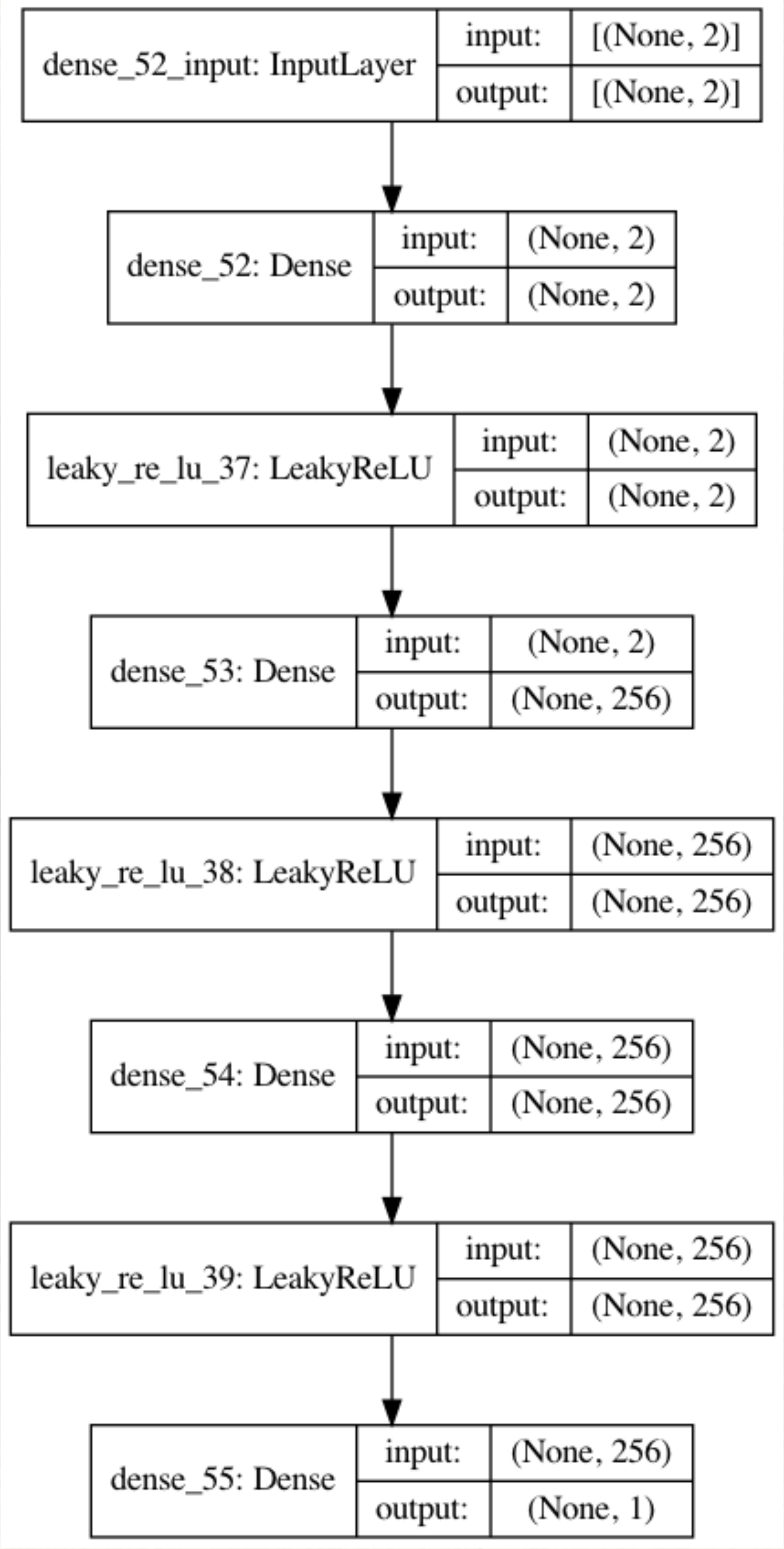
EXAMPLE ATLAS-SUSY-2018-04, 2 SRS*:

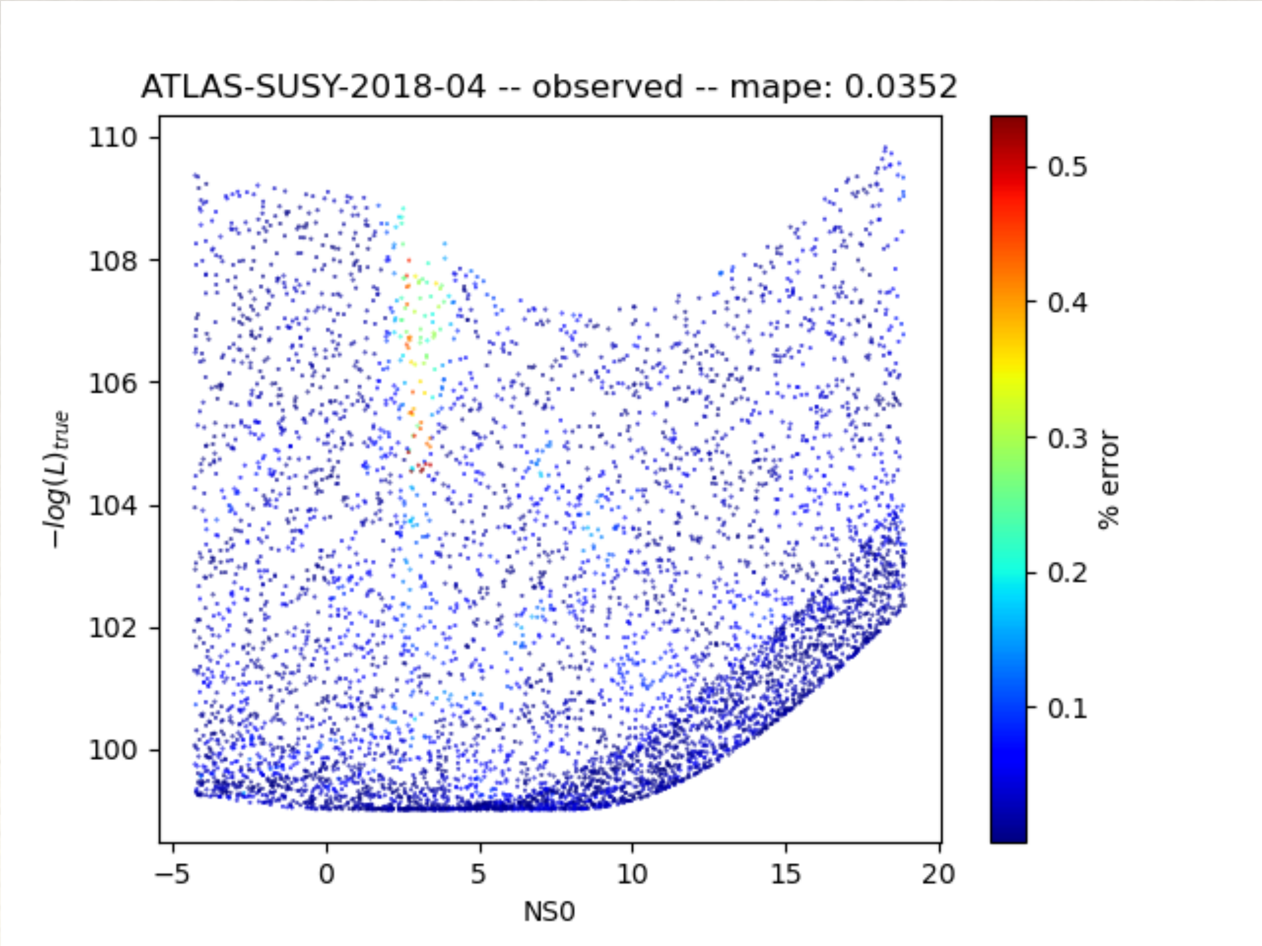


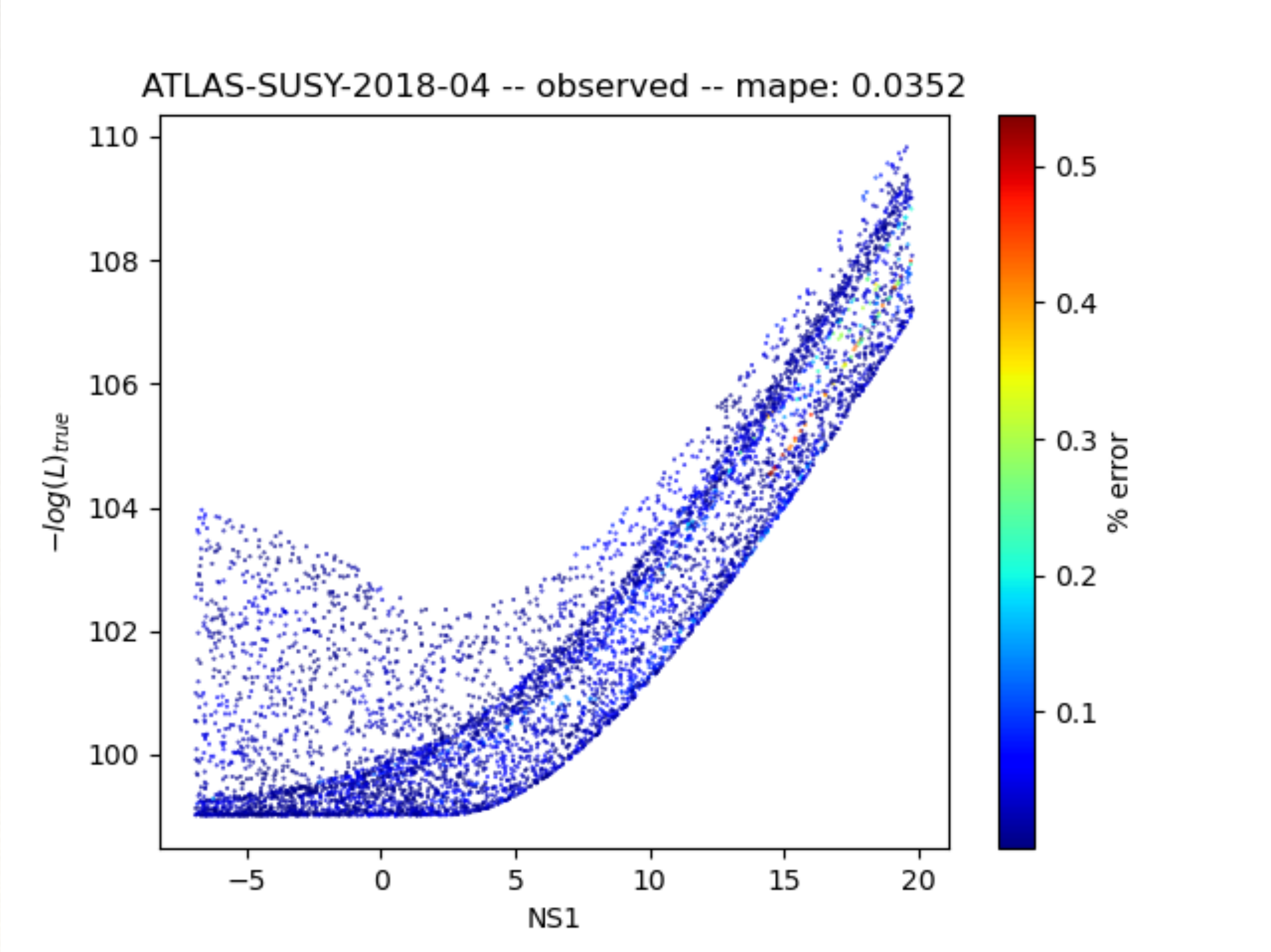


EXPECTED

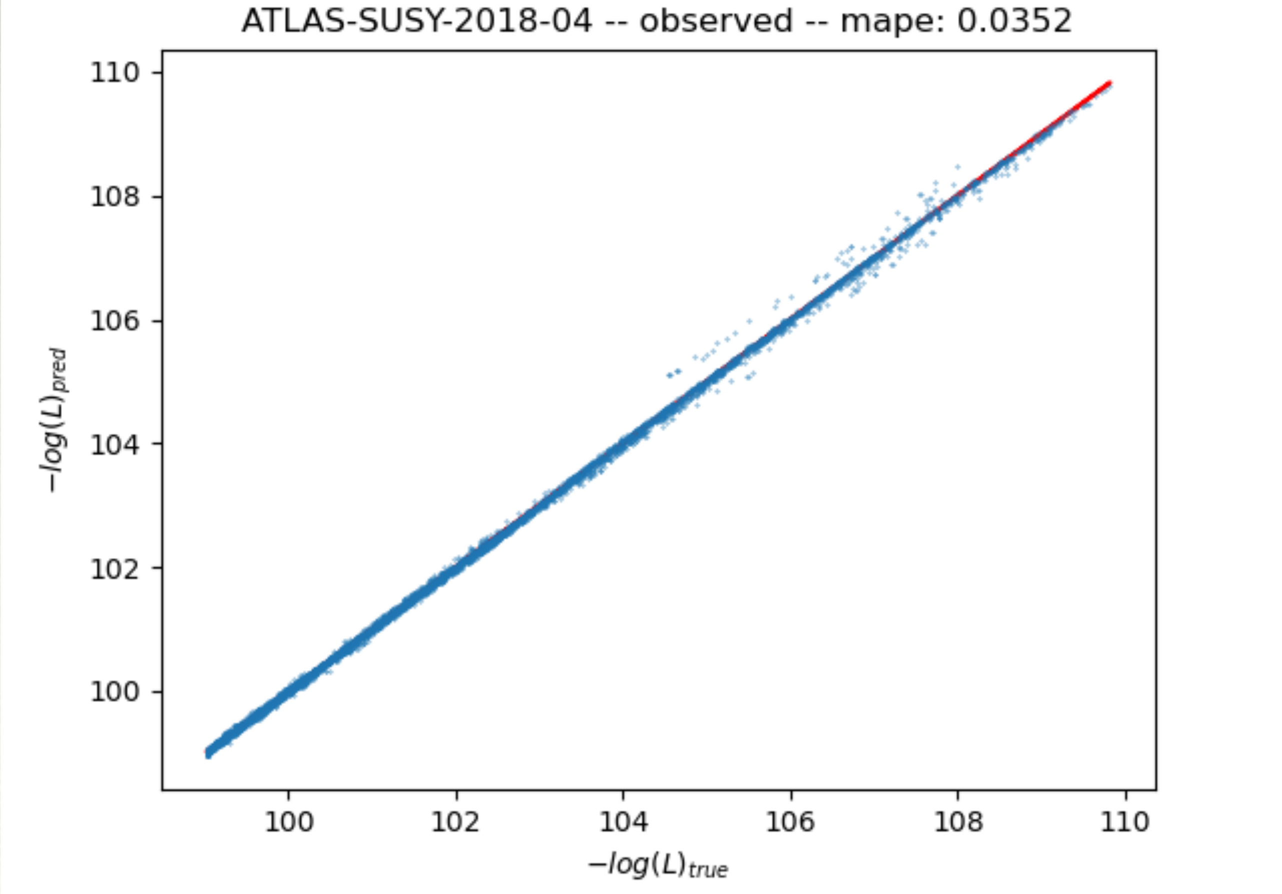








OBSERVED



*Data generated with SModelS' Pyhf interface

Conclusions

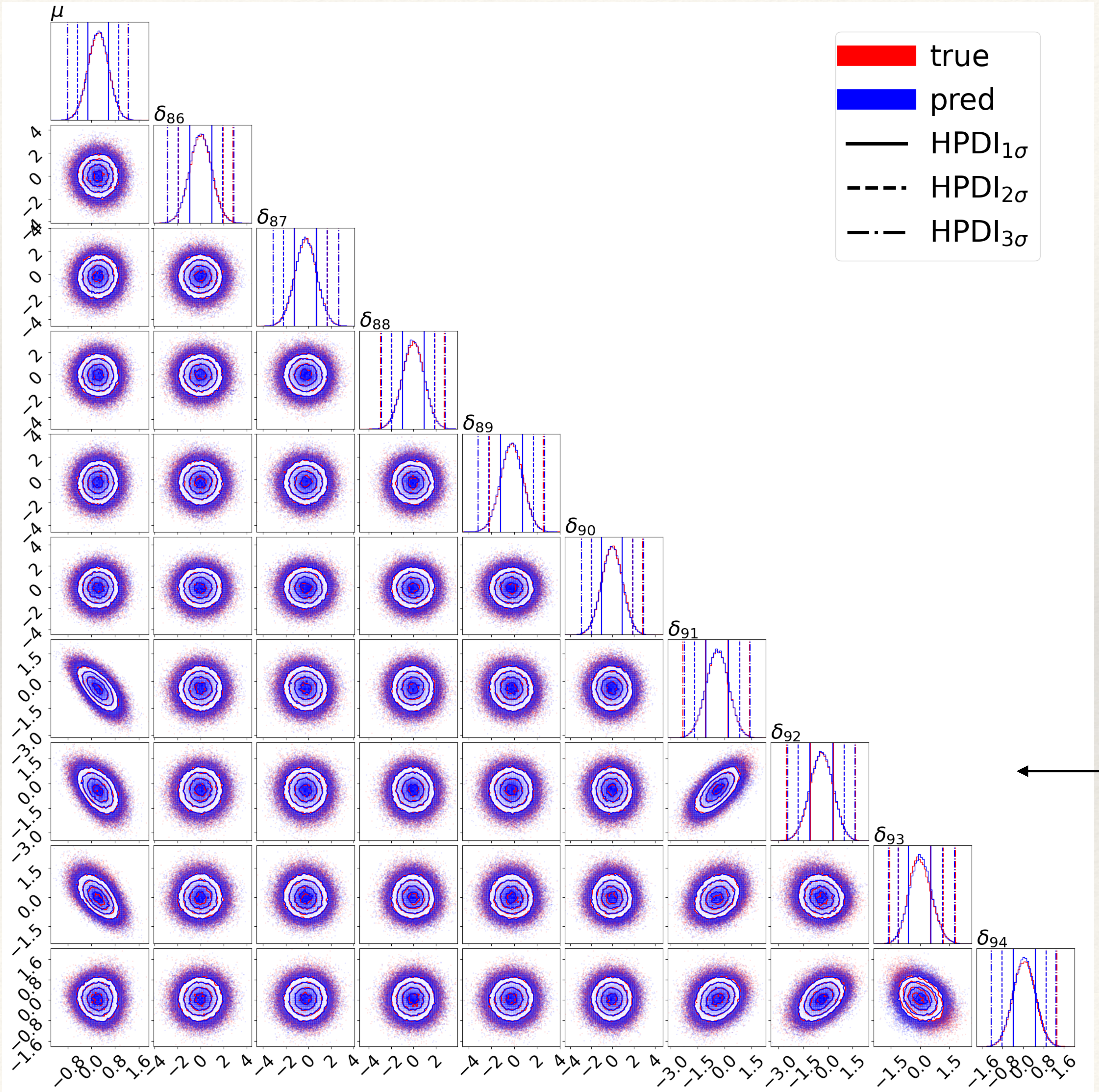
- The preservation of LHC likelihoods is of vital importance (for theorists also).
- Introduced unsupervised learning of Likelihoods with Normalizing Flows.
- Normalizing Flows show great capacity of learning complex high dimensional functions.
- Complementary, we can directly learn profiled likelihoods; useful for fast NP-search reinterpretation.

Outlook

- Paper in preparation arXiv 2301.xxxx.
- User friendly Tensorflow implementation of NFs in dev: https://github.com/riccardotorre/NFTF2_dev
- Learning full statistical models with Conditional Normalizing Flows.
- Learning profiled likelihoods from Pyhf statistical models.

THANK YOU!

LHC-like new physics search Likelihood.



Hyperparameters:

N_{train}	Flow	N bij	Hidden layers	L1 factor	N epochs	N iters.
$\mathbf{10^4}$	MAF	2	256×3	0	20	4
$\mathbf{10^5}$	MAF	2	128×3	10^{-4}	20	4
$\mathbf{2 \cdot 10^5}$	MAF	2	64×3	10^{-4}	20	4

Results:

N_{train}	KS-test	W-distance	F- norm	$\mathrm{HPDIe}_{1\sigma}$	$\mathrm{HPDIe}_{2\sigma}$	$\mathrm{HPDIe}_{3\sigma}$	time (s)
$\mathbf{10^4}$	0.479	$1.083 \cdot 10^{-2}$	0.913	$2.211 \cdot 10^{-2}$	$1.374 \cdot 10^{-2}$	$1.3003 \cdot 10^{-2}$	86.65
$\mathbf{10^5}$	0.502	$5.33 \cdot 10^{-3}$	0.527	$2.157 \cdot 10^{-2}$	$8.147 \cdot 10^{-3}$	$1.07 \cdot 10^{-2}$	317.89
$\mathbf{2 \cdot 10^5}$	.507	$4.82 \cdot 10^{-3}$	0.316	$1.883 \cdot 10^{-2}$	$9.355 \cdot 10^{-3}$	$9.903 \cdot 10^{-3}$	561.82

Test sample : 300k

BACK UP

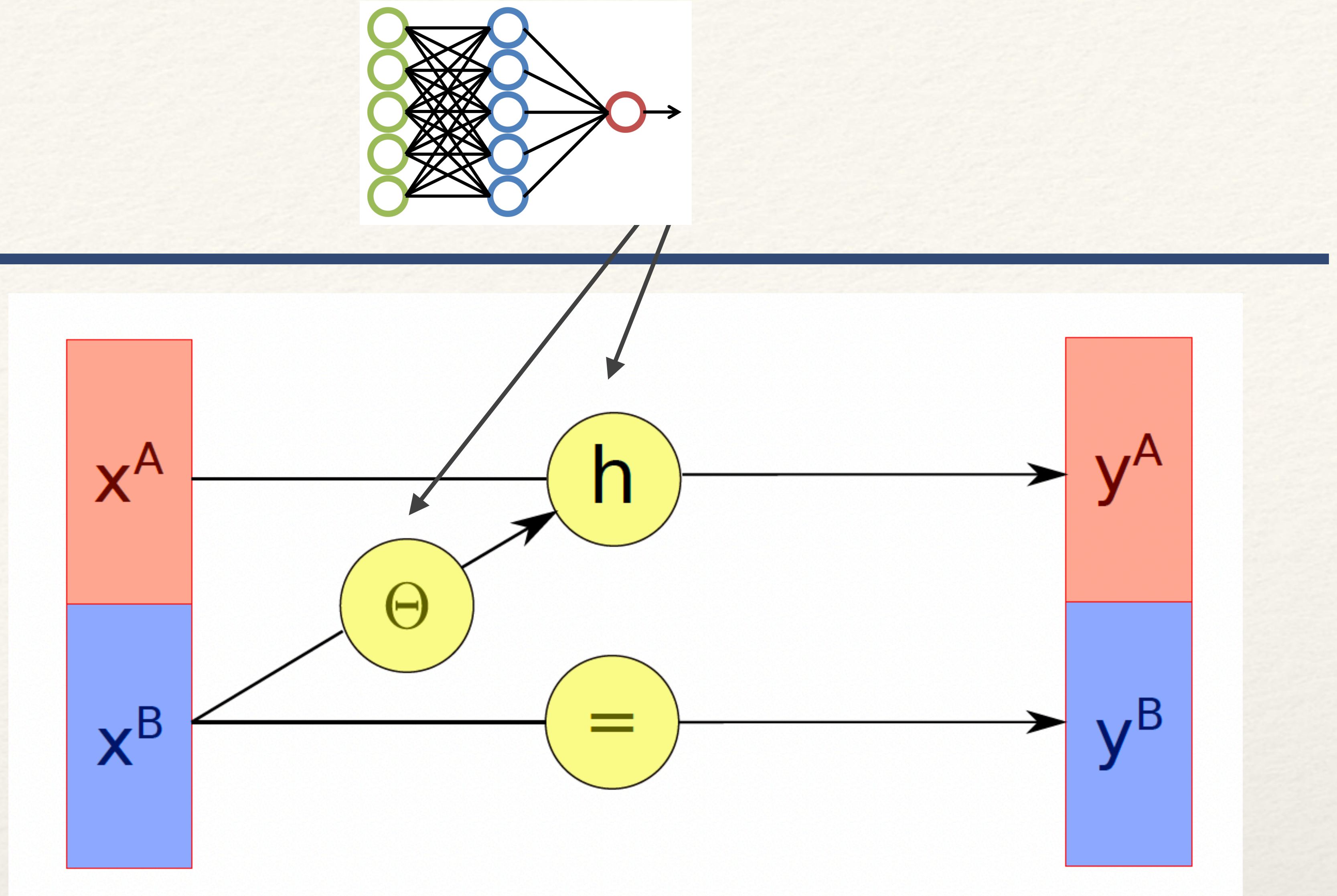
Autoregressive Flows

Coupling Flows :

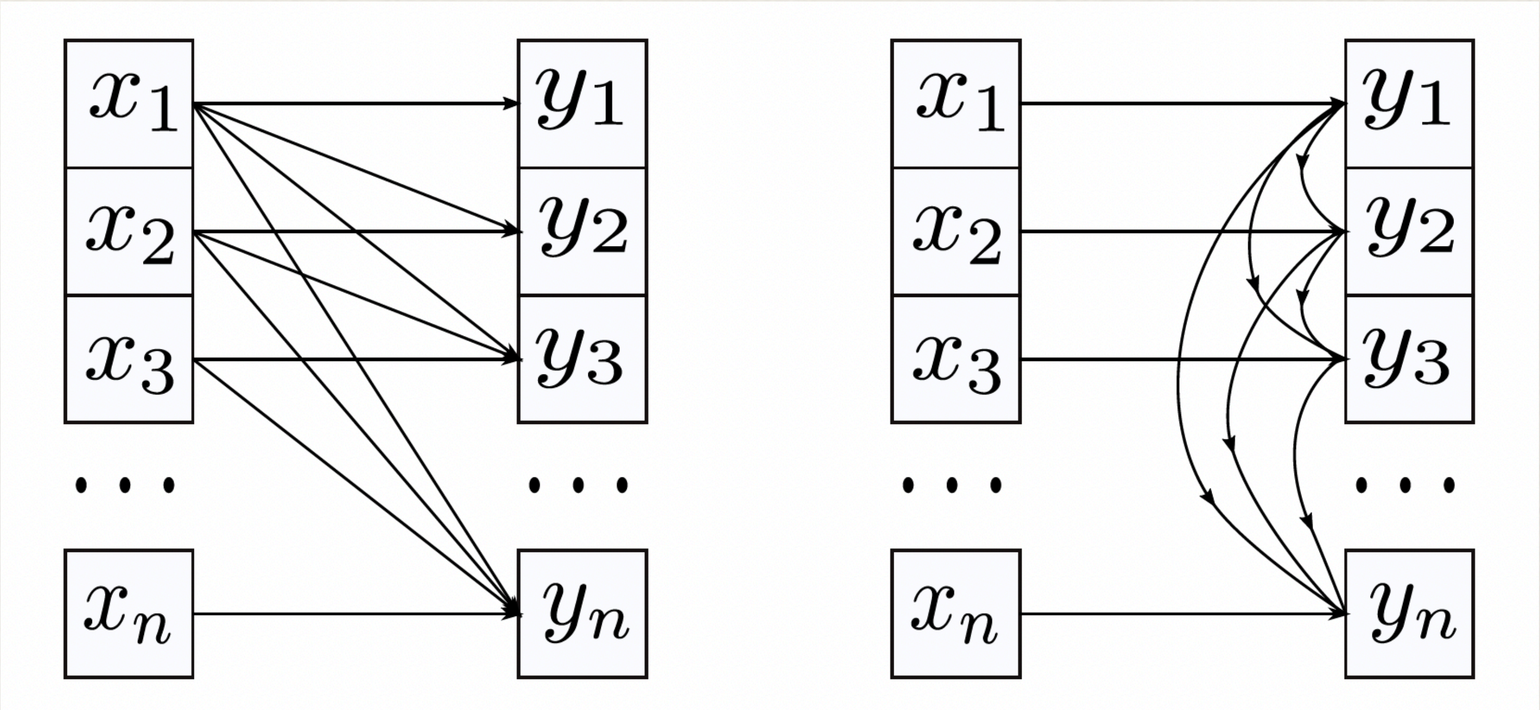
- Dimensions are divided in two sets: x^A and x^B
- We transform x^B with bijectors trained with x^A.
- The bijector parameters are functions of a NN.
- The Jacobian J is triangular -> $\det J = \Pi_i J_{ii}$
- **Jacobian is easily computed!**
- **Direct sampling AND density estimation.**
- **Less expressive.**

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.**



arXiv:1908.09257



arXiv:1908.09257

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

Introduction.

Let's get formal…

- If Z is a random variable with pdf p_Z, g is an invertible function such that $Y = g(Z)$ and $f = g^{-1}$, then we can obtain the pdf p_Y of the random variable Y as

$$p_Y(y) = p_Z(f(y))\,|\det(Df(y))| = p_Z(f(y))\,|\det(Dg(f(y))|^{-1} \quad \text{where} \quad Dg(z) = \frac{\partial g}{\partial z} \quad Df(y) = \frac{\partial f}{\partial y}$$

Jacobians

- N transformations are possible since…

$$f = f_1 \circ \ldots f_{N-1} \circ f_N$$

$$\det Df(y) = \Pi_{i=1}^{N} \det(Df_i(x_i)) \quad \text{where} \quad x_i = g_i \circ \ldots \circ g_1(z) = f_{i+1} \circ \ldots \circ f_N(y)$$

- Since p_Z is parametrised by ϕ and the bijector g by θ, we can compute the **log probability** of some measured data $\mathscr{D} = \{y^{(i)}\}_{i=1}^{M}$ given the parameters $\Theta = (\theta, \phi)$ as

$$\log p(\mathscr{D}\,|\,\Theta) = \sum_{i=1}^{M} \log p_Y(y^{(i)}\,|\,\Theta) = \sum_{i=1}^{M} \log p_Z(f(y^{(i)}\,|\,\theta)\,|\,\phi) + \log|\det Df(y^{(i)}\,|\,\theta)\,|$$