ETH zürich

Direct Likelihood Learning and Uncertainty Profiling with Normalizing Flows

Dr. Davide Valsecchi

IRIS-HEP SBI Blueprint Workshop

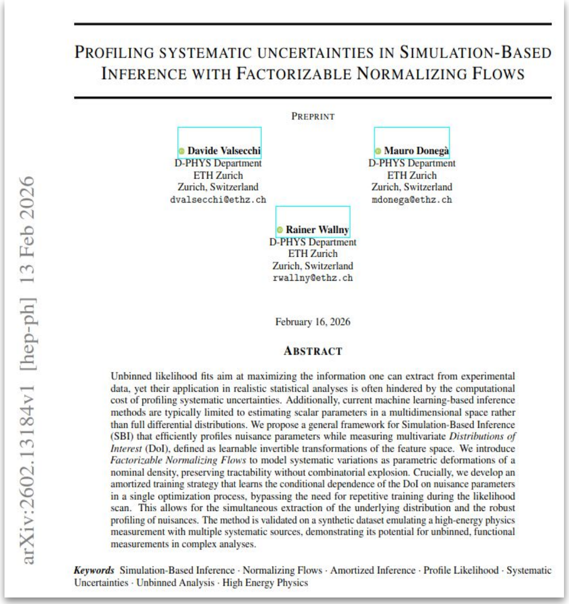
26/02/2026

Introduction

ETH zürich

1. Neural Density Estimation with **Normalizing Flows**
2. From **POI** to **DOI**: Distributions of Interest
3. Factorizable Normalizing Flow for systematic uncertainties
   - encoding systematic effects on the likelihood components
4. Profiling systematics
   - Amortized training procedure
   - “Pulls” on DOI

Discussion based on “*Profiling systematic uncertainties in Simulation-Based Inference with Factorizable Normalizing Flow*” 2602.13184 D. Valsecchi, M. Donega’, R. Wallny



PROFILING SYSTEMATIC UNCERTAINTIES IN SIMULATION-BASED INFERENCE WITH FACTORIZABLE NORMALIZING FLOWS

PREPRINT

Davide Valsecchi
D-PHYS Department
ETH Zurich
Zurich, Switzerland
dvalsecchi@ethz.ch

Mauro Donegà
D-PHYS Department
ETH Zurich
Zurich, Switzerland
mdonega@ethz.ch

Rainer Wallny
D-PHYS Department
ETH Zurich
Zurich, Switzerland
rwallny@ethz.ch

February 16, 2026

ABSTRACT

Unbinned likelihood fits aim at maximizing the information one can extract from experimental data, yet their application in realistic statistical analyses is often hindered by the computational cost of profiling systematic uncertainties. Additionally, current machine learning-based inference methods are typically limited to estimating scalar parameters in a multidimensional space rather than full differential distributions. We propose a general framework for Simulation-Based Inference (SBI) that efficiently profiles nuisance parameters while measuring multivariate *Distributions of Interest* (DoI), defined as learnable invertible transformations of the feature space. We introduce *Factorizable Normalizing Flows* to model systematic variations as parametric deformations of a nominal density, preserving tractability without combinatorial explosion. Crucially, we develop an amortized training strategy that learns the conditional dependence of the DoI on nuisance parameters in a single optimization process, bypassing the need for repetitive training during the likelihood scan. This allows for the simultaneous extraction of the underlying distribution and the robust profiling of nuisances. The method is validated on a synthetic dataset emulating a high-energy physics measurement with multiple systematic sources, demonstrating its potential for unbinned, functional measurements in complex analyses.

Keywords Simulation-Based Inference · Normalizing Flows · Amortized Inference · Profile Likelihood · Systematic Uncertainties · Unbinned Analysis · High Energy Physics

Neural Density Estimation

ETH zürich

We can write the **unbinned likelihood of our data** directly using **density estimation** models → **Normalizing Flows**

$$p(y, x) = \sum_f A(f) p(y \mid x, f)\, p(x \mid f)$$

$$A(f) = \frac{\mu_f N_f}{\sum_f \mu_f N_f}$$

x = kinematics
y = observables of interest (discriminator, invariant mass..)
f = physics process

Build the likelihood as a **mixture** of physics processes.
If needed, decompose the p.d.f. $p(y \mid x, f)\, p(x \mid f)$ to model simpler conditional probabilities from simulated events.

→ **unbinned maximum likelihood fits** on data for parameters of interest (POIs)

Normalizing flows

arxiv1908.09257
arxiv1912.02762

ETH zürich

We need to models p.d.f and conditional p.d.f. over an high-dimensional space

Normalizing flows:

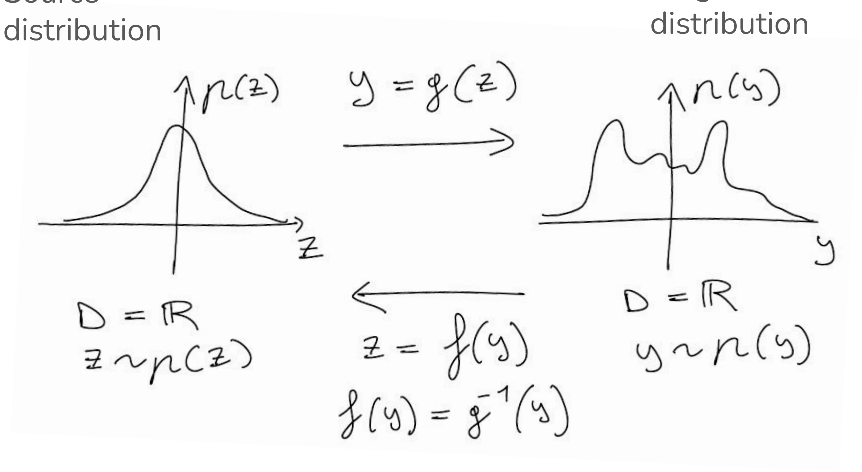
- draw **samples** from the p.d.f
- get the **probability density** at a particular point
- computationally efficient
- model highly non-gaussian, multi-modal distributions
- model conditional p.d.f $p(x \mid y)$

How it works

- Model the p.d.f as a **series of bijective transformations** from a base distribution

Source distribution

Target distribution



$$p_y(y) = p_z(z) \cdot |detJ_{f(y)}|$$

How to build a Normalizing flow

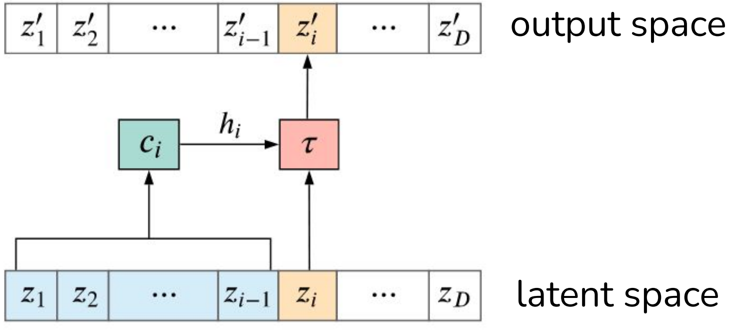
ETH zürich

We need to model **non-factorizable p.d.f**

DNNs are not invertible: use DNN as **conditioners** c_i

which parametrize invertible **transformations** τ
for which we have analytical inversion

Choose a structure with an efficient jacobian: **coupling layer**



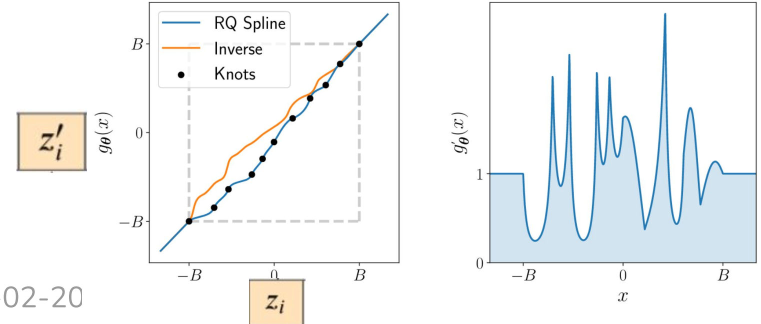
(a) Forward

The **transformation** τ can be:

affine: *μ, α parameters* from the DNN conditioner

$$z'_i = (z_i - \mu_i)\exp(-\alpha_i)$$

or **spline** based: model N knots with the DNN conditioner, which creates a spline to transform differently each dimension → very expressive



arxiv1906.04032

arxiv1912.02762

arxiv1705.07057

Applications of Neural Density Estimation

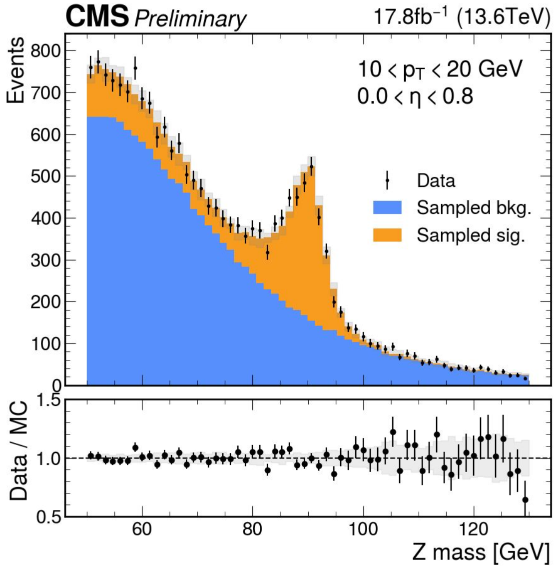
ETH zürich

The direct likelihood approach allows to build unbinned calibrations workflows:
unbinned multidimensional Tag&Probe measurements on CMS data

CMS-DP-2025-53 presented at ACAT25

$$p(m \mid \vec{c}) = \alpha(\vec{c}) \cdot S_{\mathrm{Data}}(m \mid \vec{c}) + (1 - \alpha(\vec{c}))B_{\mathrm{Data}}(m \mid \vec{c})$$

signal fraction — signal p.d.f — bkg p.d.f



Fit the Z invariant mass to measure lepton ID efficiencies
→ unbinned and continuous calibration over large phasespace with a single fit

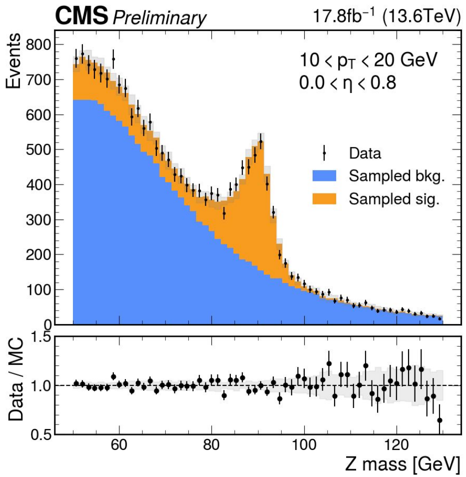
Applications of Neural Density Estimation

$$p(m \mid \vec{c}) = \alpha(\vec{c}) \cdot S_{\text{Data}}(m \mid \vec{c}) + (1 - \alpha(\vec{c}))B_{\text{Data}}(m \mid \vec{c})$$

signal fraction

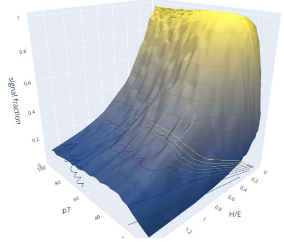
signal p.d.f

bkg p.d.f



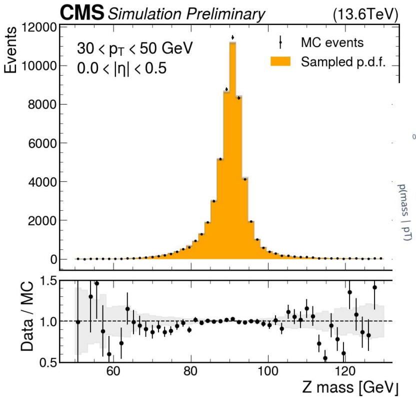
Target: signal fraction

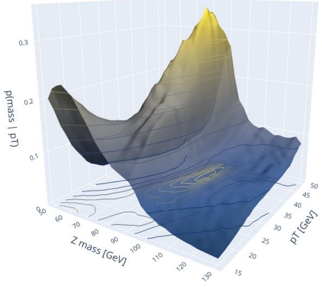
$$\alpha_{\text{Pass/Fail}}(\vec{c}) \in [0, 1]$$



Signal model from simulation

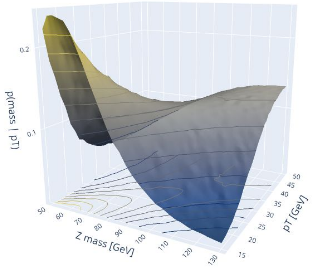
$$S_{\text{MC}}(m \mid \vec{c}) = \text{Nflow}(m \mid \vec{c})$$





Parametric p.d.f. for bkg

$$B(m \mid \vec{c}) = \text{erfc}((A(\vec{c}) - x) \cdot B(\vec{c})) \cdot e^{-x \cdot D(\vec{c})}$$



Classifiers for conditional fractions

ETH zürich

In many applications, modeling the full likelihood may be too complex or high dimensional:
→ model a conditional p.d.f. of observables of interest given **X**
→ need density estimation only fo $p(y \mid x, f)$

$$p(y \mid x) = \sum_f A(f \mid x) p(y \mid x, f)$$

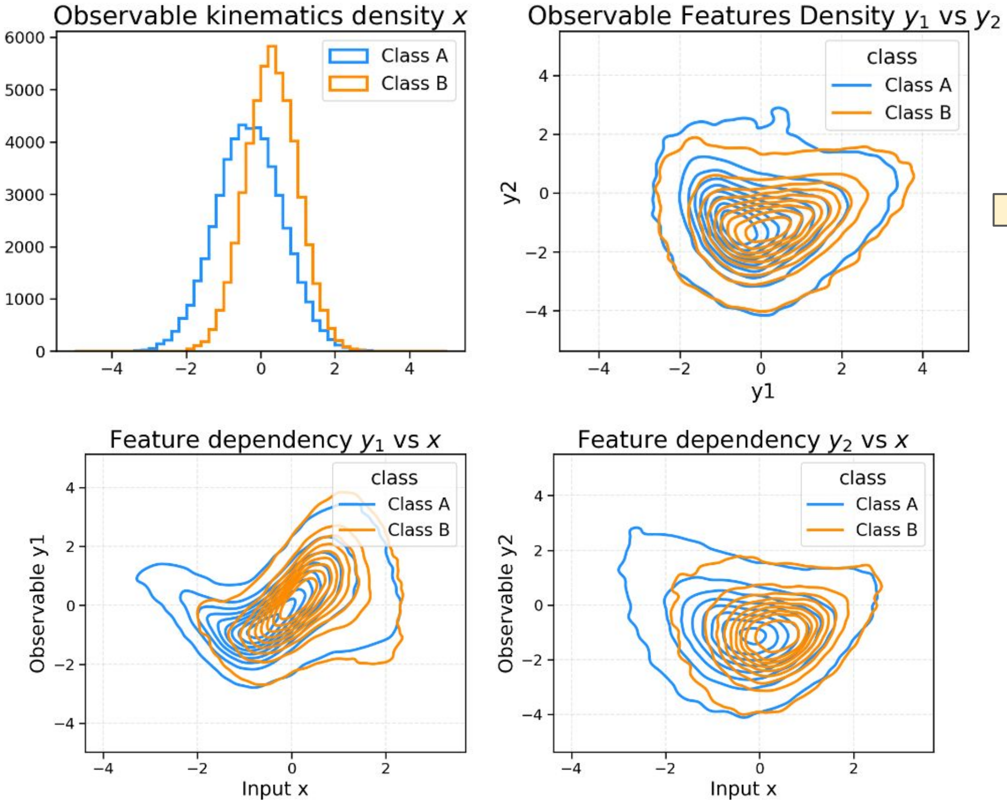
$$A(f \mid x) = \frac{\mu_f \, N_f \, p(x|f)}{\sum_f \mu_f \, N_f \, p(x|f)}$$

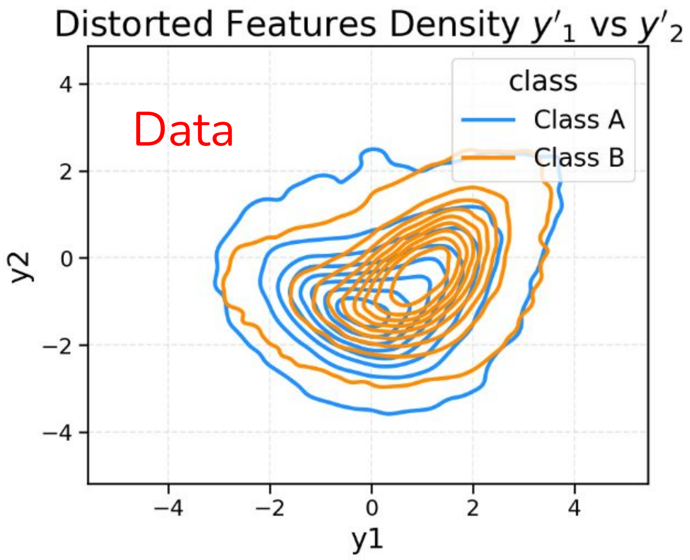
The mixture weight can now be modeled using a **classifier** with the likelihood-ratio trick, trained to distinguish the processes *f* using the features *x*.

From **POI** to **DOI**: Distributions of Interest

Toy Dataset

Simulation





The target of our analysis is to measure the map between simulation and Data

Distributions of Interest

The direct likelihood computation makes possible to generalize the target of our measurement. From POI to **Distributions of Interest (DOI)**

New target: find an invertible transformation T_{ϕ_f} that maps the data to the reference simulation.

Optimizing for T_{ϕ_f} allows a **functional unbinned measurement** → general formulation for unbinned cross-section measurements, data-MC calibrations

$$p(y,x) = \sum_f A(f) p(y \mid x, f)\, p(x \mid f)$$

$$T_{\phi_f}(y' \mid x, \nu) = y$$

⇓

$$p_f(y, x \mid \nu) = p_f(T_{\phi_f}(y \mid x, \nu))\, p_f(x \mid \nu) \cdot \left| \det \nabla_y T_{\phi_f}(y \mid x, \nu) \right|$$

Encoding the DOI

$T(y \mid x, f)$ needs to be a bijective conditional map in $R^N \rightarrow R^N$

- fast to compute
- efficient to invert $\rightarrow$ to apply on the simulation
- conditional on x, flavour
- jacobian known and easy to compute

$\rightarrow$ **Invertible NN architecture (autoregressive spline transformation or coupling layers)**

$T(y \mid x, f)$ won't be an optimal transport (OT) solution and won't be unique

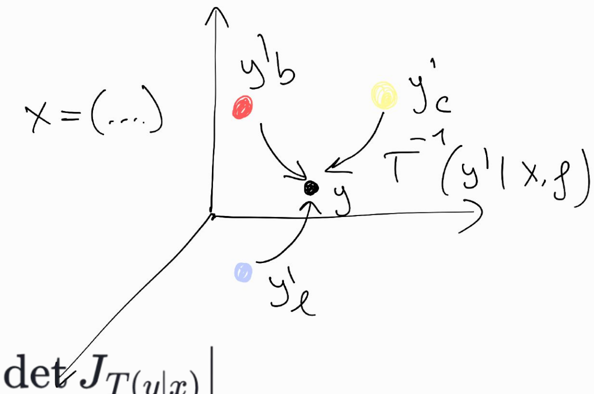
$\rightarrow$ relaxing the hard requirements on the form of the map allows us to optimize it using likelihood fits and **profile systematics uncertainties**

$\rightarrow$ can restore an OT map a-posteriori

ETH zürich

Likelihood fit

The total likelihood on data needs to include the determinant of the **jacobian** of the transformation, which is easy to compute thanks to the specific form of T (autoregressive splines)



$$p(y, x) = \sum_f \sum_s A(f, s) \cdot p(T(y \mid x, f) \mid x, f, s) \cdot p(x \mid f, s) \cdot \left|\det J_{T(y|x)}\right|$$

The final optimization happens only on the parameters of T, all the input p.d.f.s for the scores and kinematics are **frozen (prior from simulation)**

$$J_{T(y|x)} = \begin{pmatrix} \frac{\partial T_1}{\partial y_1} & \cdots & \frac{\partial T_1}{\partial y_3} \\ \vdots & \vdots & \vdots \\ \frac{\partial T_3}{\partial y_1} & \cdots & \frac{\partial T_3}{\partial y_3} \end{pmatrix}$$

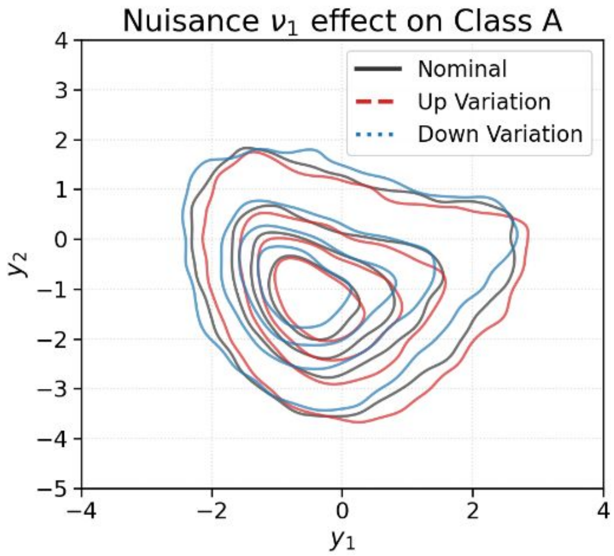
$$\hat{T}_{\phi_f}, \hat{\nu} = \arg\max_{\phi,\nu} \mathcal{L}_{\text{ext}}(y, x | \theta, \nu, T_{\phi_f})$$

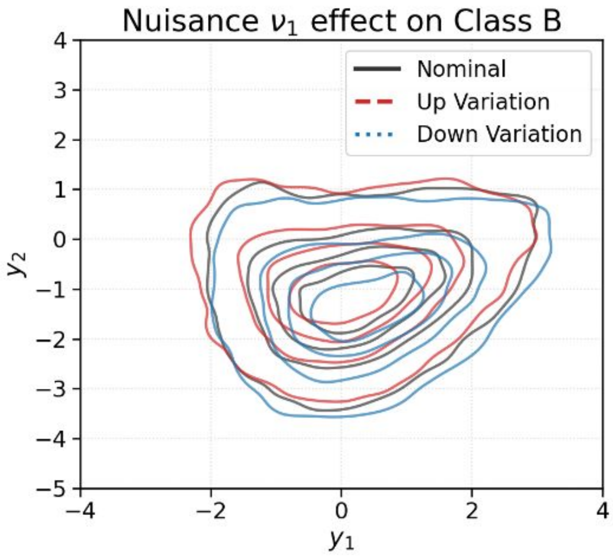
Factorizable Normalizing Flow

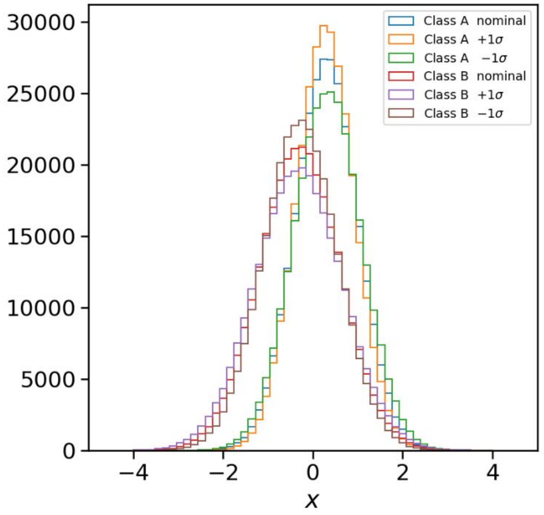
for systematic uncertainties

Systematics effects in the toy dataset

We include both effects on $p(y \mid x, f)$ and on $p(x \mid f)$. Effects are different for each flavour but correlated by the same nuisance parameter.







Systematic uncertainty as a transformation

ETH zürich

We model systematic uncertainties by **varying the prediction of our simulation**, independently for each source of uncertainty.

→ we need to **include these effects** in our likelihood model

It's **intractable** to include the dependency on the nuisance parameters directly in the conditional p.d.f. $p_f(y|x,\nu)$ as we would need to train this model in a large multidimensional ν space → need too many simulations

New idea: Model the p.d.f as a composition of a *nominal model* and a **learnable systematic transformation** T_ν

$$p_f(y|x,\nu) = p_{\text{nom},f}(T_\nu(y|x,\nu)|x) \cdot |\det \nabla_y T_\nu(y|x,\nu)|$$

Factorizable Normalizing Flow

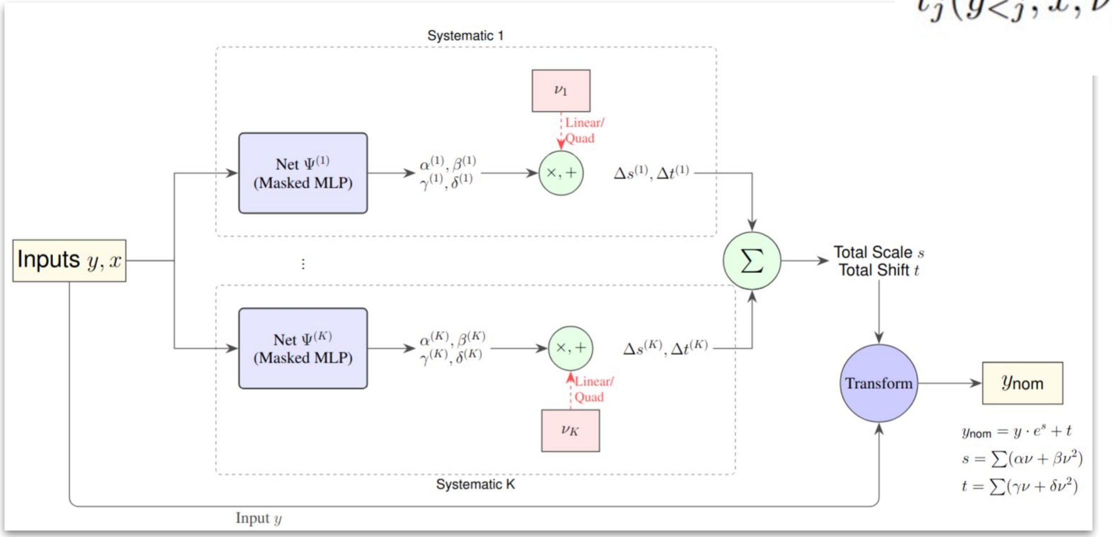
We encode T_ν as a coupling layer where the transformation is encoded with a **quadratic dependency** over the nuisance parameters.

Neural networks

$$s_j(y_{<j}, x, \nu) = \sum_{k=1}^{K} \left(\alpha_j^{(k)}(y_{<j}, x)\, \nu_k + \beta_j^{(k)}(y_{<j}, x)\, \nu_k^2 \right)$$

$$t_j(y_{<j}, x, \nu) = \sum_{k=1}^{K} \left(\gamma_j^{(k)}(y_{<j}, x)\, \nu_k + \delta_j^{(k)}(y_{<j}, x)\, \nu_k^2 \right)$$

expansion on nuisance params



Factorizable Normalizing Flow

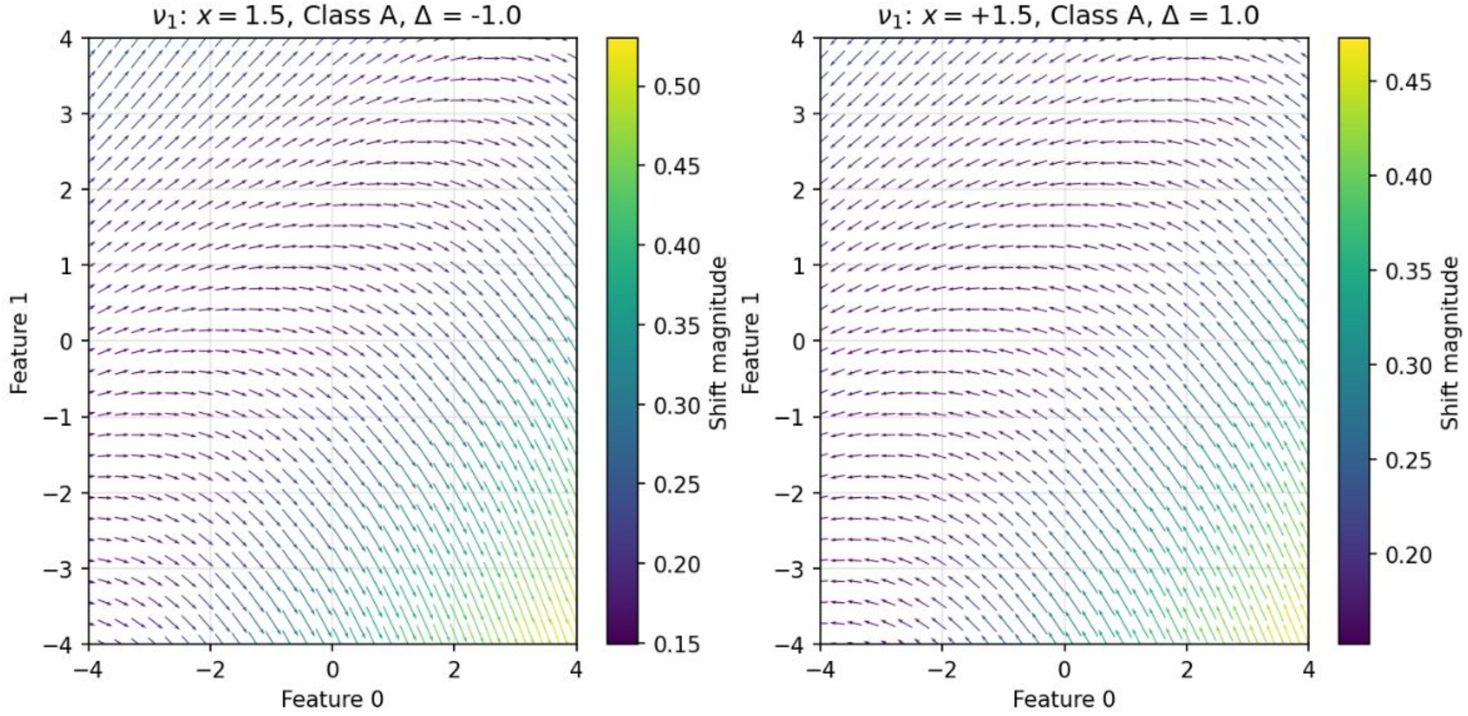
Factorizable Normalizing Flow

Encoding the systematic uncertainties on the likelihood components with this special layer allows us to:

- **Learn 1 effect at a time** and then compose them together
- Only **+-1 “sigma” simulation** is needed thanks to the quadratic parametrization
    - can also model the interpolation with alternative simulations
- Interpretation **close to combine/HistFactory** approach but on unbinned multivariate densities
- Fast and **efficient** computation
- Can scale to **many dimensions** (autoregressive structure)

Factorizable Normalizing Flow

Visualization of T_ν on $p(y \mid x, f)$ in a specific point of the phase space x



Uncertainties on the DOI

ETH zürich

We now need to measure the effect of uncertainty on the target → **Profiling**

Profiling is **expensive** $\hat{T}_{\phi_f}, \hat{\nu} = \arg\max_{\phi,\nu} \mathcal{L}_{\text{ext}}(y, x|\theta, \nu, T_{\phi_f})$

for each value in the nuisance parameters space ν: would need to optimize the model T.

→ build a **Factorizable Normalizing Flow** model to learn the effect of **systematics on the target DOI**

$$T_{\phi_f}(y|x, \Delta\nu) = T^{\hat{\nu}}_{\phi_f}(y|x) \circ T_{\psi_f}(y|x, \Delta\nu)$$

compose a **global-best model** with a **systematic-aware** layer

This solved the parameterization problem, but not the computational problem

Profiling as Amortized training

ETH zürich

Instead of optimizing the DOI in each point of the nuisances space (intractable)

$$\hat{T}_{\phi_f}, \hat{\nu} = \arg\max_{\phi,\nu} \mathcal{L}_{\text{ext}}(y, x|\theta, \nu, T_{\phi_f})$$

1) we optimize a global map without systematics dependency, looking for the best-fit of the nuisance parameters:

$$\hat{T}^{\hat{\nu}}_{\phi_f}, \hat{\nu} = \arg\max_{\phi^{\hat{\nu}},\nu} \mathcal{L}_{\text{ext}}(y, x|\nu, T^{\nu}_{\phi_f})$$

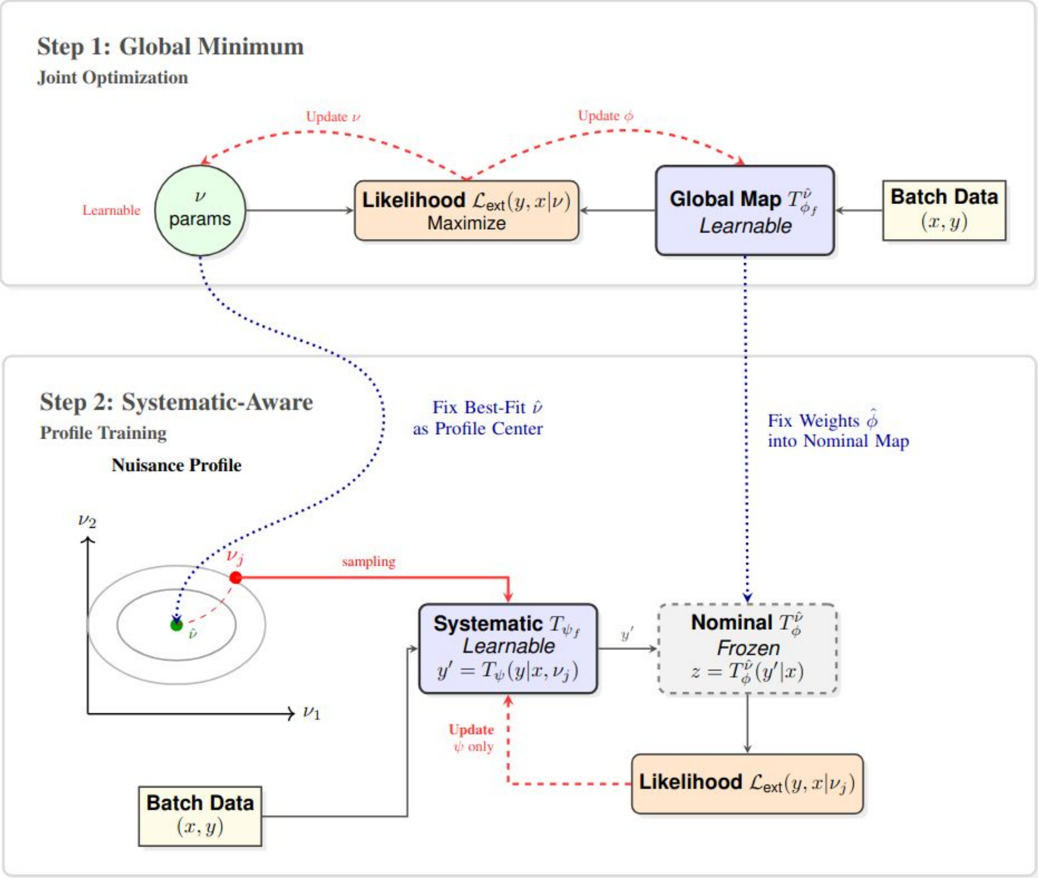
2) We then build a **new** model, as an **expansion around the nominal** (with FNF)
$\rightarrow$ systematics-aware composed transformation

$$T_{\phi_f}(y|x, \Delta\nu) = T^{\hat{\nu}}_{\phi_f}(y|x) \circ T_{\psi_f}(y|x, \Delta\nu)$$

We train this model by **sampling in the nuisance parameters space**
$\rightarrow$ **best-fit DOI** in each point ν

$$\hat{T}_{\psi_f}, \hat{\nu} = \arg\max_{\psi,\nu} \mathcal{L}_{\text{ext}}(y, x|\nu, T^{\hat{\nu}}_{\phi_f} \circ T_{\psi_f})$$

Profiling as Amortized training

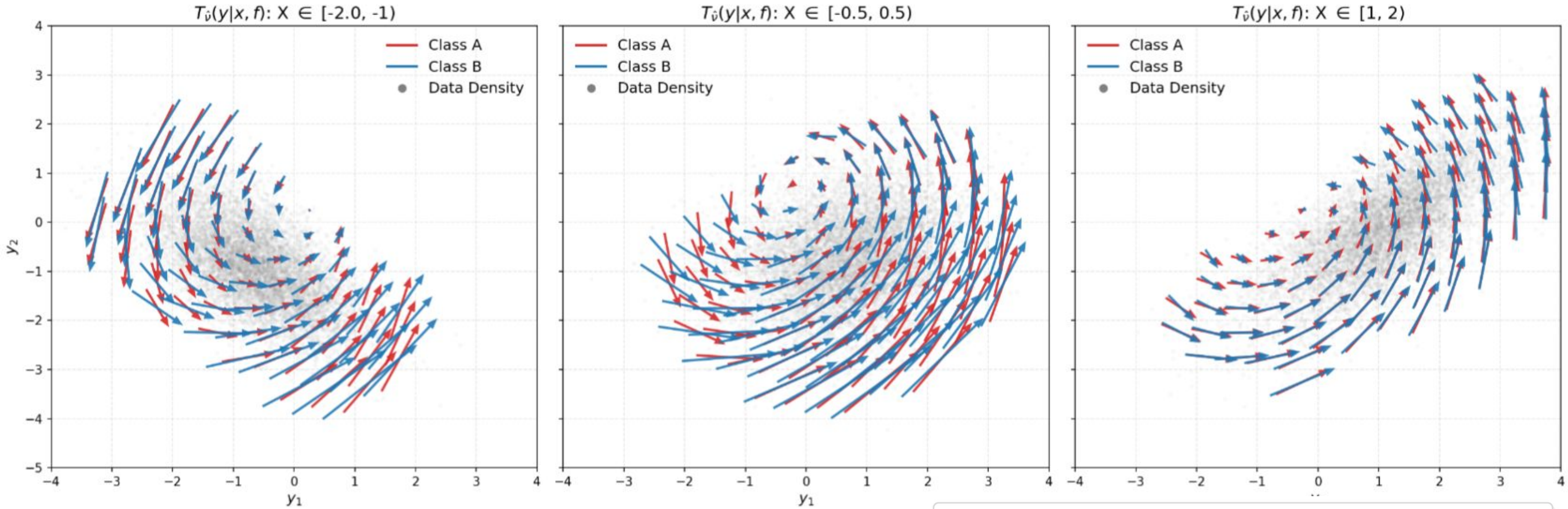


we train a **systematic-aware transformation** by **sampling the nuisance parameters space**
→ optimal model in each ν at inference time

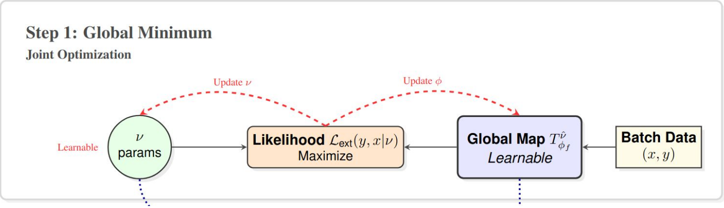
Move the cost of profiling to the training instead of inference

Global fit

Visualization of the best-fit DOI in different **x** points

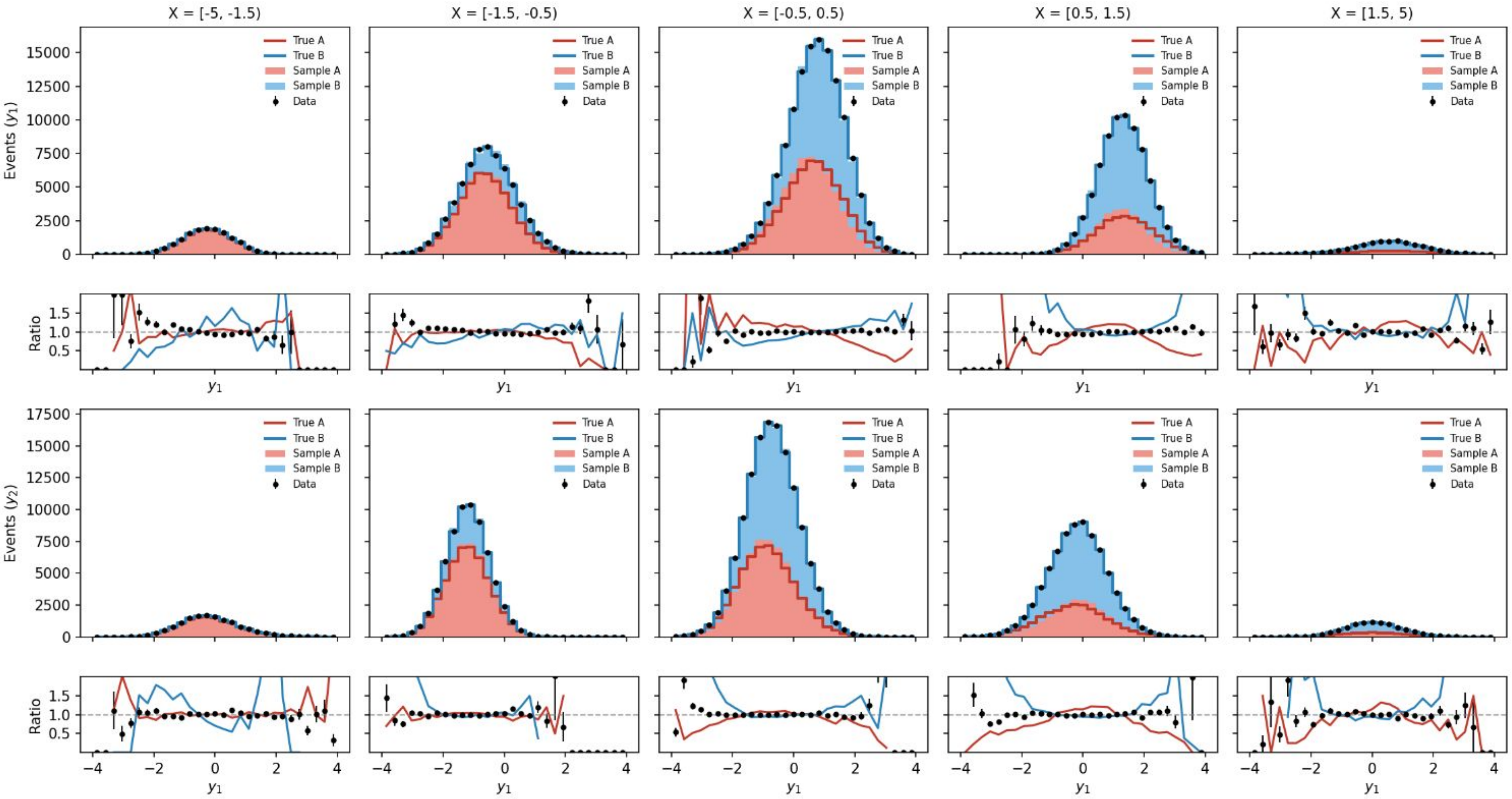


$$\hat{T}^{\hat{\nu}}_{\phi_f}, \hat{\nu} = \arg\max_{\phi^{\hat{\nu}},\nu} \mathcal{L}_{\text{ext}}(y, x|\nu, T^{\nu}_{\phi_f})$$



Nominal DOI

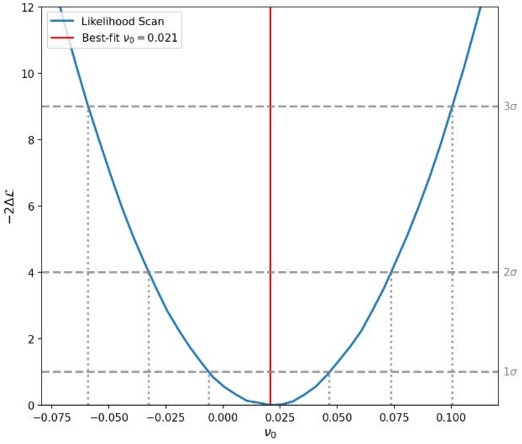
ETH zürich

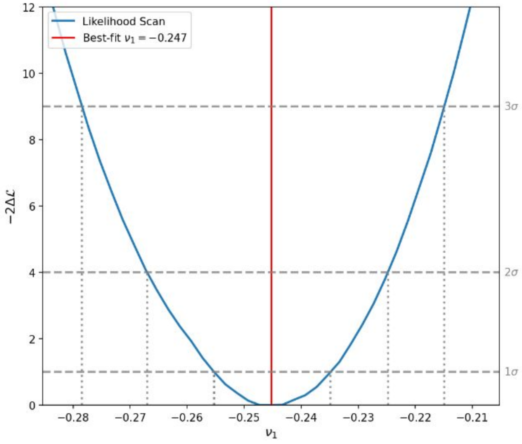


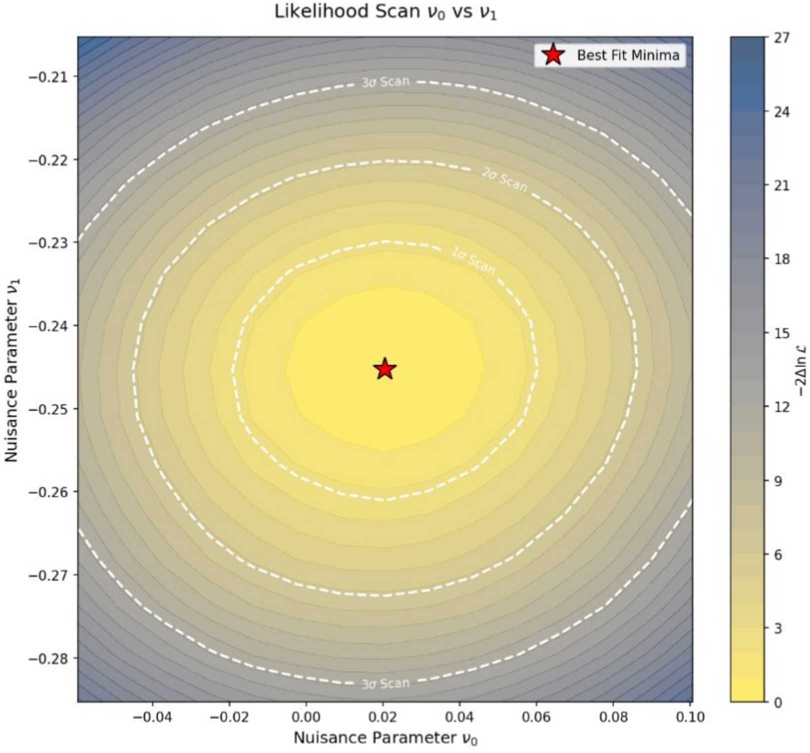
Good closure comparing Data with samples transformed thought the DOI

Global fit - likelihood scan

The likelihood formulation allows to perform likelihood scans in the nuisances space. The DOI $T^{\hat{\nu}}_{\phi_f}$ is frozen to the best-fit model.



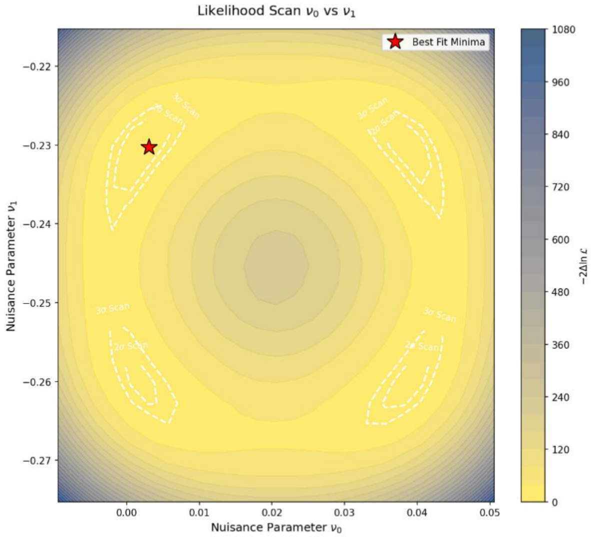


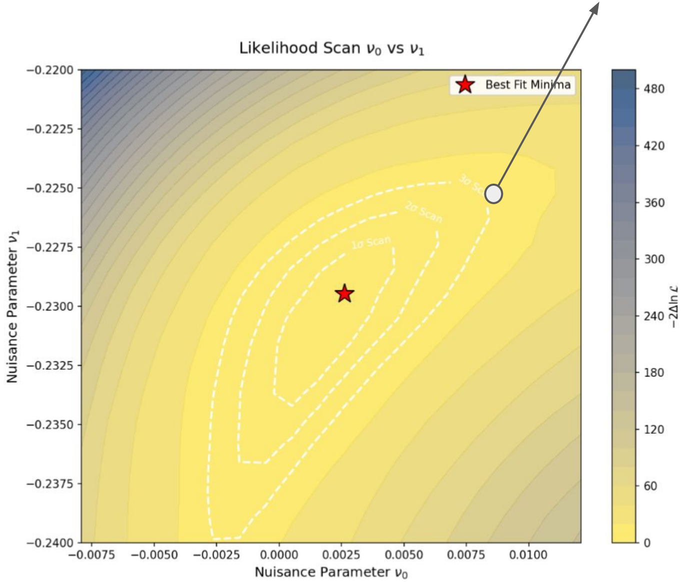


Profiling

We then build the **systematic-aware model** and optimize it **sampling** in the nuisance parameters space.
Perform again the likelihood scan → this is the **profiled likelihood as the model is optimal in each ν point**

$$T_{\phi_f}(y|x,\Delta\nu) = T^{\hat{\nu}}_{\phi_f}(y|x) \circ T_{\psi_f}(y|x,\Delta\nu)$$





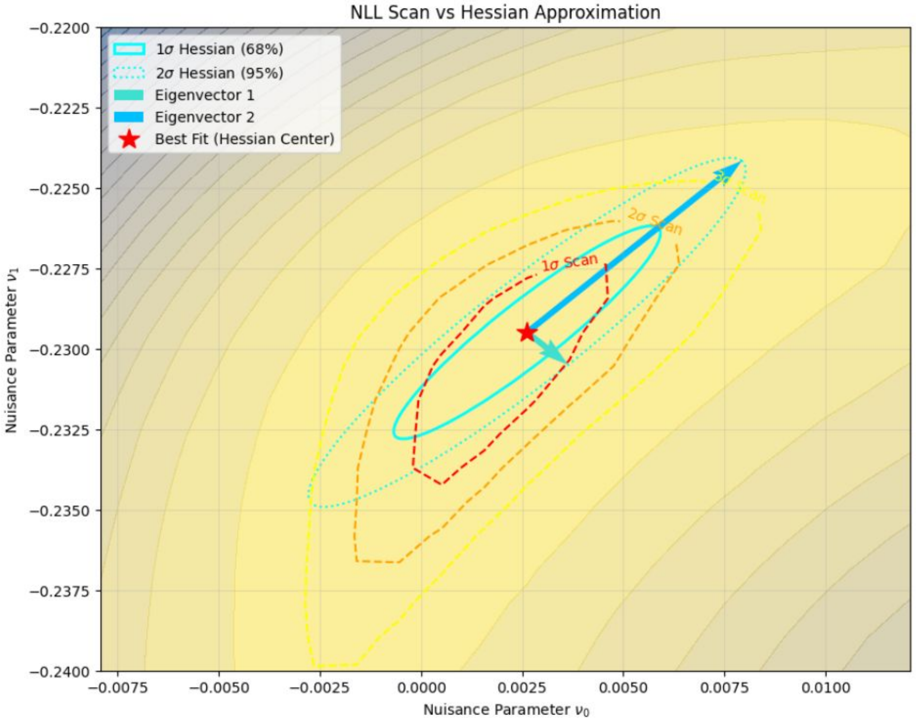
Uncertainty decomposition

This formulation allows to extract **uncertainty decomposition on the DOI easily.**

1. Compute the Hessian in the best-fit point:
2. using automatic differentiation which takes into account the dependency of T on ν

$$T_{\phi_f}(y|x, \Delta\nu)$$

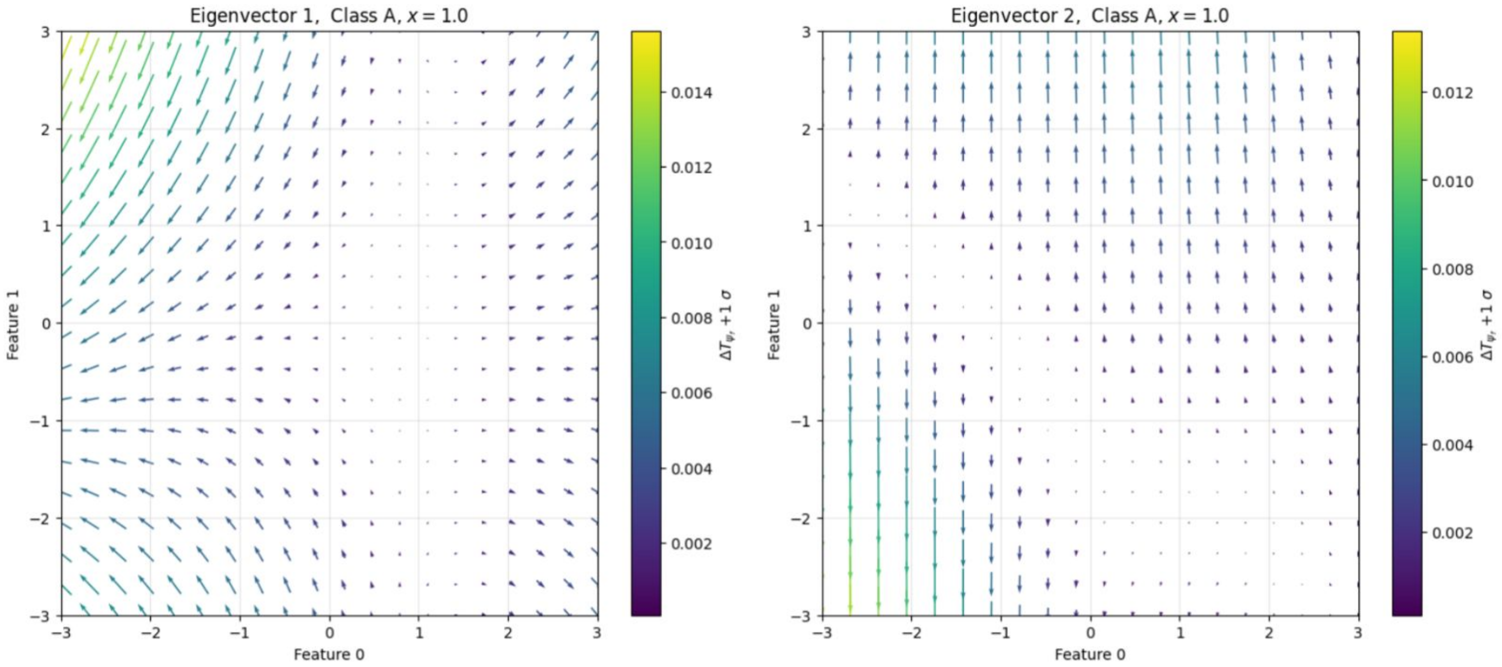
3. Perform PCA
4. Evaluate $T_{\phi_f}(y|x, \Delta\nu)$ in the eigenvectors
   - get the "pulls" of the DOI



“Impact plots”

Visualizing pulls as “Δ” transformation from the nominal, in a specific x point.

→ 2D nuisance space → 2 directions



Conclusions

ETH zürich

Presented applications of direct Neural Density Estimation with Normalizing Flows.

- Powerful alternative to likelihood ratio when **modeling invertible transformation** that maps the data to the reference simulation.
  - ---> this is a very general formulation! It fits well unbinned unfolded, data-MC calibrations etc
- Proposed a new strategy to **include systematic uncertainties** in likelihood input densities
- New workflow for **systematic profiling in unbinned likelihood** fits
- Need to build **tooling** around this approach

→ Looking forward to apply this approach on a HEP analysis!

Backup

Toy dataset - details