# Invertible Networks for the Matrix Element Method

Theo Heimel

March 2022

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Work in progress with Anja Butter, Till Martini, Sascha Peitzsch and Tilman Plehn

# Introduction

- LHC measurements largely compatible with SM
  $\rightarrow$ hints for New Physics might be hidden in large SM backgrounds
- Traditional analyses: compare distribution of selected observables to data
  $\rightarrow$ only fraction of information is used!
- Need analysis techniques which
  $\rightarrow$ are based on first principles
  $\rightarrow$ estimate uncertainties reliably
  $\rightarrow$ use most of the available information
- Promising candidate: **Matrix Element Method** (MEM)

# Matrix Element Method

- MEM: multivariate maximum likelihood method with **likelihood calculated from first principles** (QFT) [Kondo, 1988, 1991]
- Optimal use of information content
  $\rightarrow$ works for very small number of observations
- Likelihood for parameter $\Omega$ from observations $\{x^i\}$ given by

$$\mathcal{L}(\Omega|\{x^i\}) = \prod_i \frac{1}{\sigma(\Omega)} \frac{d\sigma(\Omega)}{dx_1^i \dots dx_r^i}$$

- Cross section only known analytically at parton level
  $\rightarrow$ need to **invert effects of parton shower, hadronization and detector**
  $\rightarrow$ transfer function $\mathcal{T}(y, x)$ from detector level $y$ to parton level $x$

$$\mathcal{L}(\Omega|\{y^i\}) = \prod_i \frac{1}{\sigma(\Omega)} \int d^r x \frac{d\sigma(\Omega)}{dx_1^i \dots dx_r^i} \mathcal{T}(y^i, x)$$

# Matrix Element Method

- Decompose transfer function as

$$\mathcal{T}(y,x) = p(x|y)\epsilon(y)$$

$\rightarrow$ **Idea: Use neural network to learn** $p(x|y)$

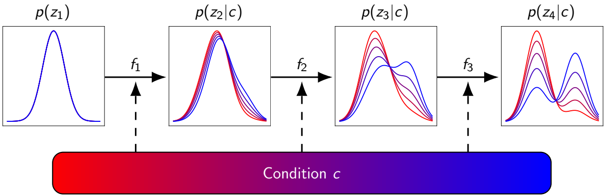
- Write likelihood as

$$\begin{aligned}\mathcal{L}(\Omega|\{y^i\}) &= \prod_i \frac{1}{\sigma(\Omega)} \int d^r x \frac{d\sigma(\Omega)}{dx_1^i \dots dx_r^i} \mathcal{T}(y^i, x) \\ &= \prod_i \frac{\epsilon(y^i)}{\sigma(\Omega)} \int d^r x \frac{d\sigma(\Omega)}{dx_1^i \dots dx_r^i} p(x|y^i) \\ &= \prod_i \frac{\epsilon(y^i)}{\sigma(\Omega)} \left\langle \frac{d\sigma(\Omega)}{dx_1^i \dots dx_r^i} \right\rangle_{x \sim p(x|y^i)}\end{aligned}$$

$\rightarrow$ Generative ML model as phase space sampler

# Invertible Neural Networks (INNs)

- INNs (normalizing flows): **chain of learnable, invertible transformations**
- Transform latent distribution (e.g. Gaussian) into distribution of interest

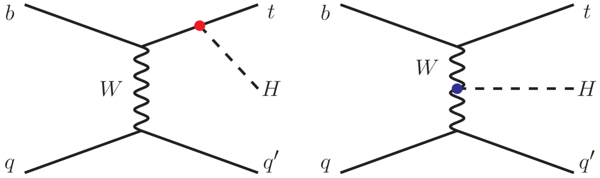



- Training: Evaluate in backward direction to get $z_1$ (latent space) $\rightarrow$ maximize log-likelihood (from change of variables formula)

$$\mathcal{L} = \log p(z_n) = \log p(z_1) + \log \left| \det \frac{\partial f^{-1}}{\partial z_n} \right|$$

- Sampling: Sample from $p(z_1)$, evaluate forward to get $z_n$
- **INN for detector and parton shower unfolding** [Bellagente et al., 2006.06685]

# Physics Model




- Single Higgs production with anomalous non-CP-conserving Higgs coupling

$$\mathcal{L}_{t\bar{t}H} = -\frac{y_t}{\sqrt{2}}\left[a\cos\alpha\ \bar{t}t + ib\sin\alpha\ \bar{t}\gamma_5 t\right]H$$

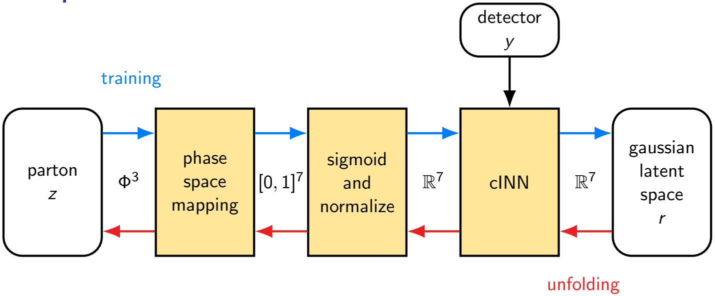
  $a = 1,\ b = \frac{2}{3}$, **mixing angle** $\alpha$

  [Artoisenet et al, 1306.6464] [de Aquino, Mawatari, 1307.5607]

  [Demartin, Maltoni, Mawatari, Zaro, 1504.00611]

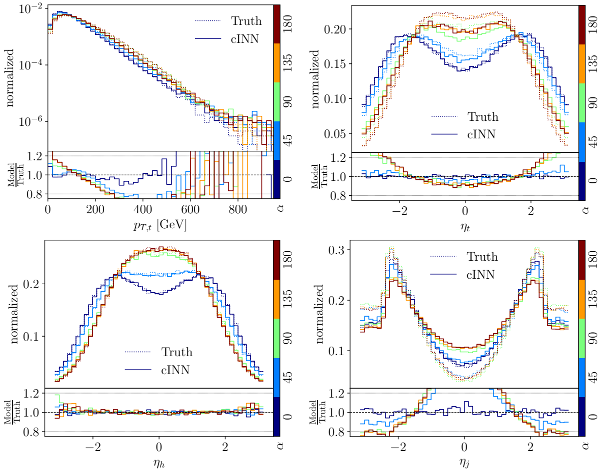
- Only 20 detector events at 3000 fb$^{-1}$
  $\rightarrow$ very well suited for MEM

# Network Setup




- Phase space mapping to unit hypercube inspired by RAMBO [Plätzer, 1308.2922]
- Another mapping to get normalized distributions in as network inputs
- cINN with rational quadratic spline coupling blocks [Durkan et al., 1906.04032]
- Generate training and test data set with Madgraph, Pythia and Delphes
  $\rightarrow$ accept events with two photons, 1 b-tagged jet, at least 3 more jets
  $\rightarrow$ only $\sim 6\%$ of the events left after cuts
- Train network with $\mathcal{O}(1\mathrm{M})$ events for $\mathcal{O}(100)$ epochs

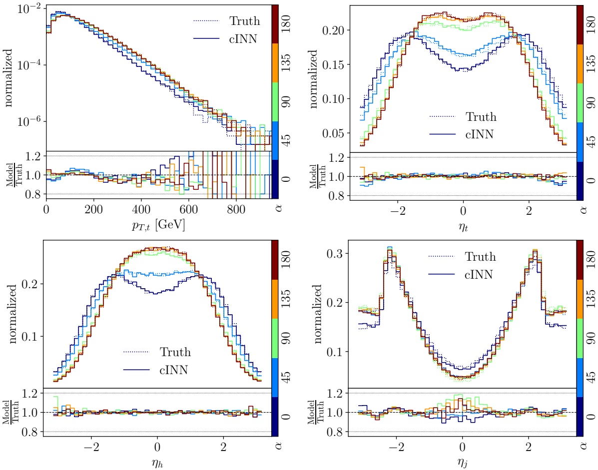
# Results (trained on SM)




Unfolded kinematic distributions

- Test performance
  $\rightarrow$ send detector-level events through network
  $\rightarrow$ check if parton-level distribution is recovered
- Train network on SM, evaluate with different angles
- Good performance only for $\alpha$ close to 0
- **Need to condition network on $\alpha$**

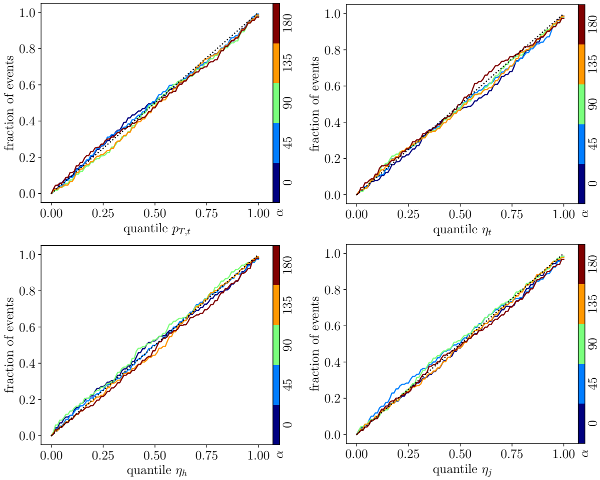
# Results (conditioned on mixing angle)




Unfolded kinematic distributions

- Condition network on detector data $y$ and mixing angle $\alpha$
- Train network with $\alpha$ sampled uniformly from $[-180, 180]$
- **Much better agreement for all mixing angles**

# Calibration




Calibration curves for kinematic distributions

- Kinematic distributions only show performance over whole data set
- Need to test performance for single events
- Take 2048 detector-level events, unfold each 60 times
- For observable: Calculate fraction of unfolded events with value smaller than true value
- Plot percentiles for fractions
- **Good calibration for network conditioned on $\alpha$**

# Bayesian networks

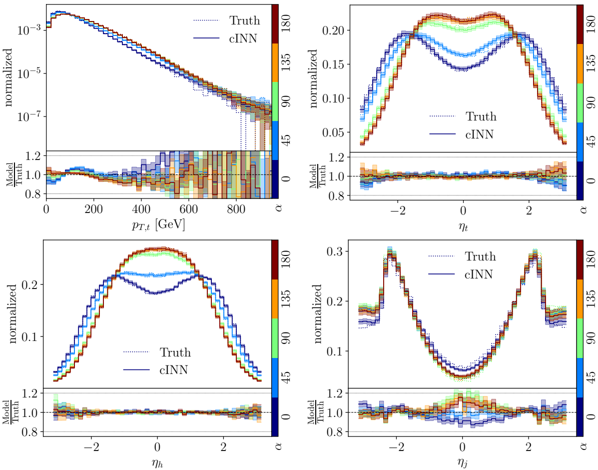


- What are the uncertainties from network training and training data?
- Obvious method: Train ensemble of networks
  $\rightarrow$ Problem: Uses a lot of computational resources
- Solution: Bayesian neural networks [MacCay, 1995] [Neal, 2012]
  $\rightarrow$ Network weights not fixed but drawn from Gaussian distribution

$$\theta_i \text{ fixed} \quad \rightarrow \quad \theta_i \sim \mathcal{N}(\mu_i, \sigma_i)$$

  $\rightarrow$ additional loss term to learn $\mu_i$ and $\sigma_i$
- Previous physics applications
  $\rightarrow$ Top tagging [Bollweg et al., 1904.10004]
  $\rightarrow$ Regression [Kasieczka et al., 2003.11099]
  $\rightarrow$ Event generation [Bellagente et al., 2104.04543] [Butter et al., 2110.13632]

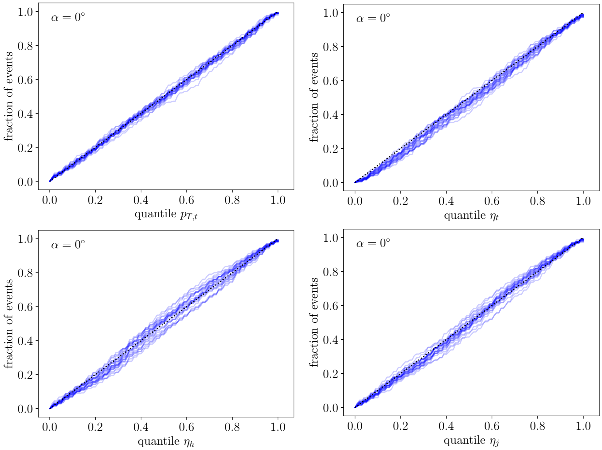
# Results with Bayesian networks




Unfolded kinematic distributions

- Make histograms for multiple sampled networks
- Show means and standard deviations for each bin
  $\rightarrow$ histogram with error bars
- **Performance comparable to deterministic network**
- Limitation: If network not able to learn a feature, this will not be included in the error bar!

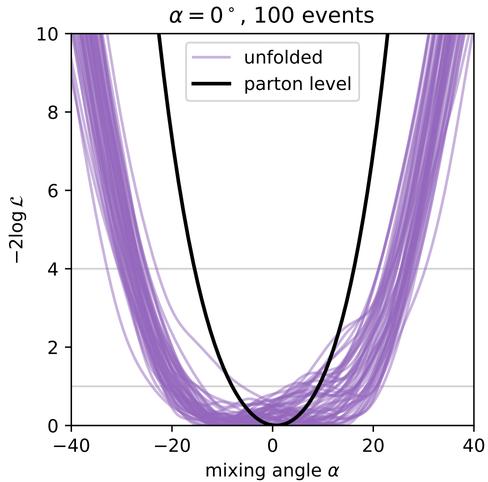
# Calibration of Bayesian networks




Calibration curves for kinematic distributions, $\alpha = 0°$

- Sample multiple networks and make a calibration curve for each
- Good performance for most networks, slight bias for some
- For MEM: **look at multiple sampled networks to control systematic bias from training**

# MEM results (preliminary)




- Combine INN unfolding with MEM
- Look at 100 SM events
- Unfold 100k times each for points in steps of 5°
  $\rightarrow$ calculate diff. cross sections
  $\rightarrow$ take trimmed mean
- Sample 50 Bayesian networks
  $\rightarrow$ one likelihood curve for each
- Statistical uncertainty from width of likelihood
- Systematic training uncertainty from Bayesian networks

# Summary

- MEM: Maximum likelihood method using first principles, optimal use of event information
- Use INN as transfer function
  $\rightarrow$ Successfully inverts parton shower, hadronization, detector effects
- Estimate systematic training uncertainty with Bayesian networks
- Successful in getting likelihood curves using this setup
- Still some problems to be solved
  $\rightarrow$ Stability for more anomalous events
  $\rightarrow$ Susceptible to bias for events from some phase space regions
- **Combination of MEM with INNs is promising alternative to traditional analysis techniques**