UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

COMETA WG2: Normalizing Flows in Particle Physics and Beyond
28.03.2024

Generative Unfolding

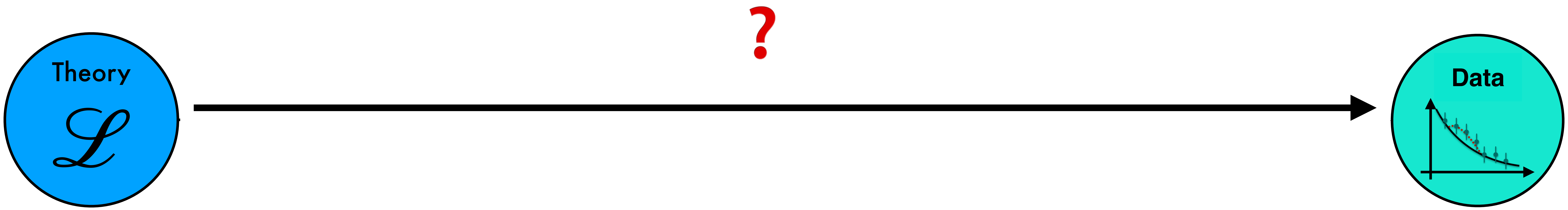
with conditional invertible neural networks

SPONSORED BY THE

Federal Ministry
of Education
and Research

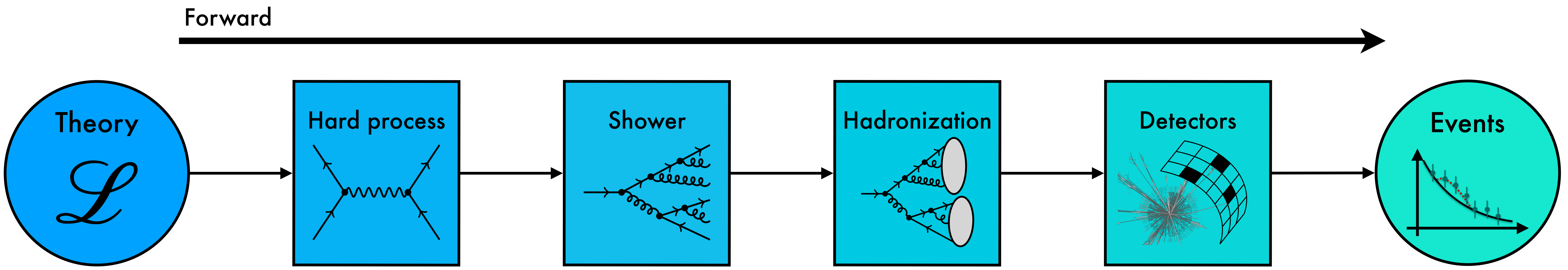
Sofia Palacios Schweitzer with Tilman Plehn, Anja Butter
Institute for theoretical Physics — Heidelberg

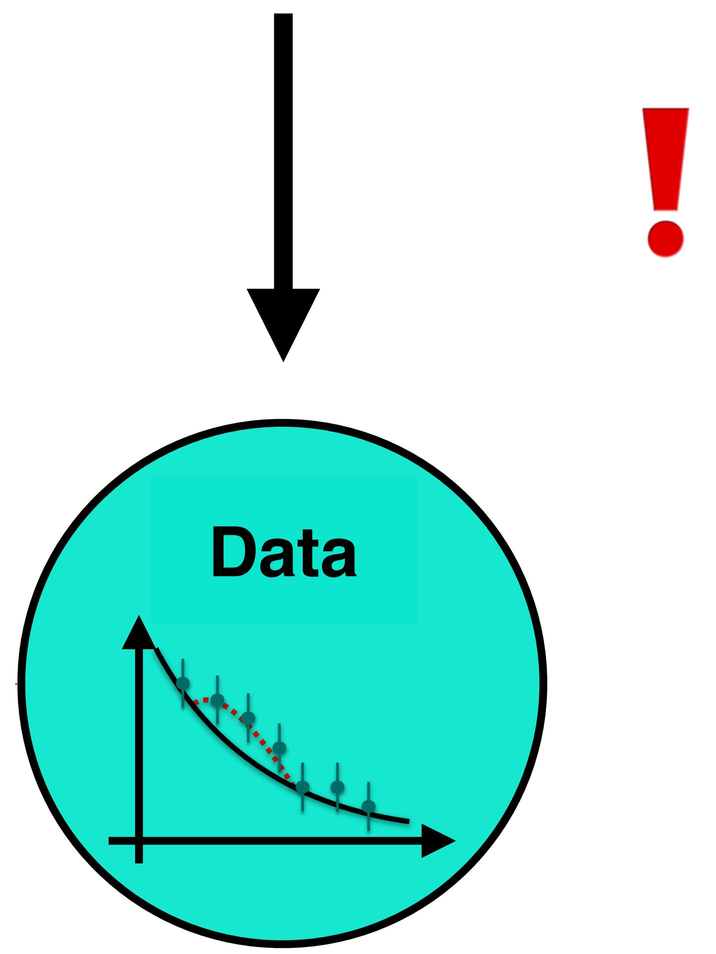
Simulation Chain



Cannot calculate testable predictions from first principles

Simulation Chain





Cannot calculate testable predictions from first principles

Theory and Experiment connected via
non-deterministic simulation

After simulation chain: compare measured and simulated distributions

Figure from A. Butter et al.: arXiv:2203.07460, R. Winterhalder

Simulation Chain — Inversion

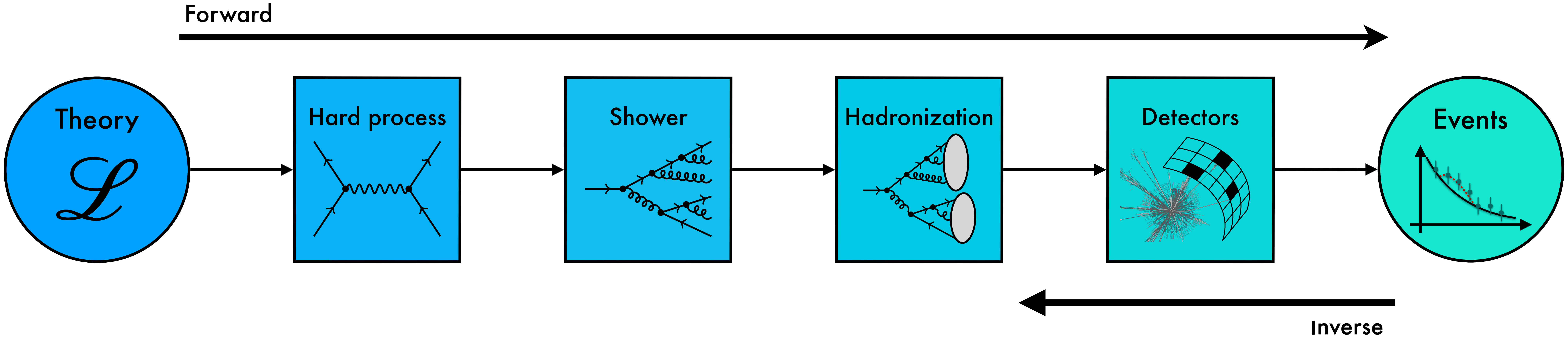
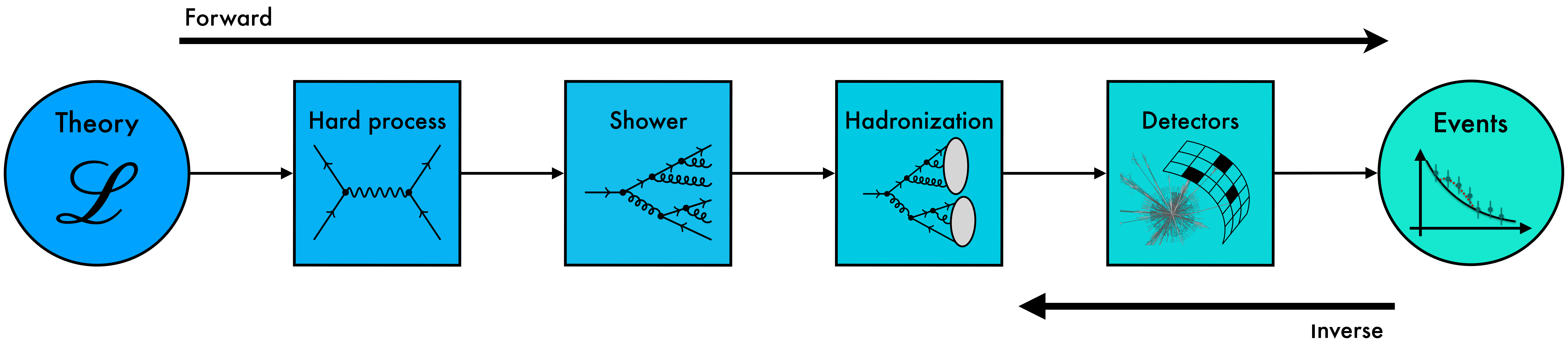


Figure from A. Butter et al.: arXiv:2203.07460, R. Winterhalder

Why unfolding?



Theory analyses don't care about detectors

Comparing data from different experiments (Global Analysis)

For some analysis direct access to theory parameters

Resolution

Data preservation

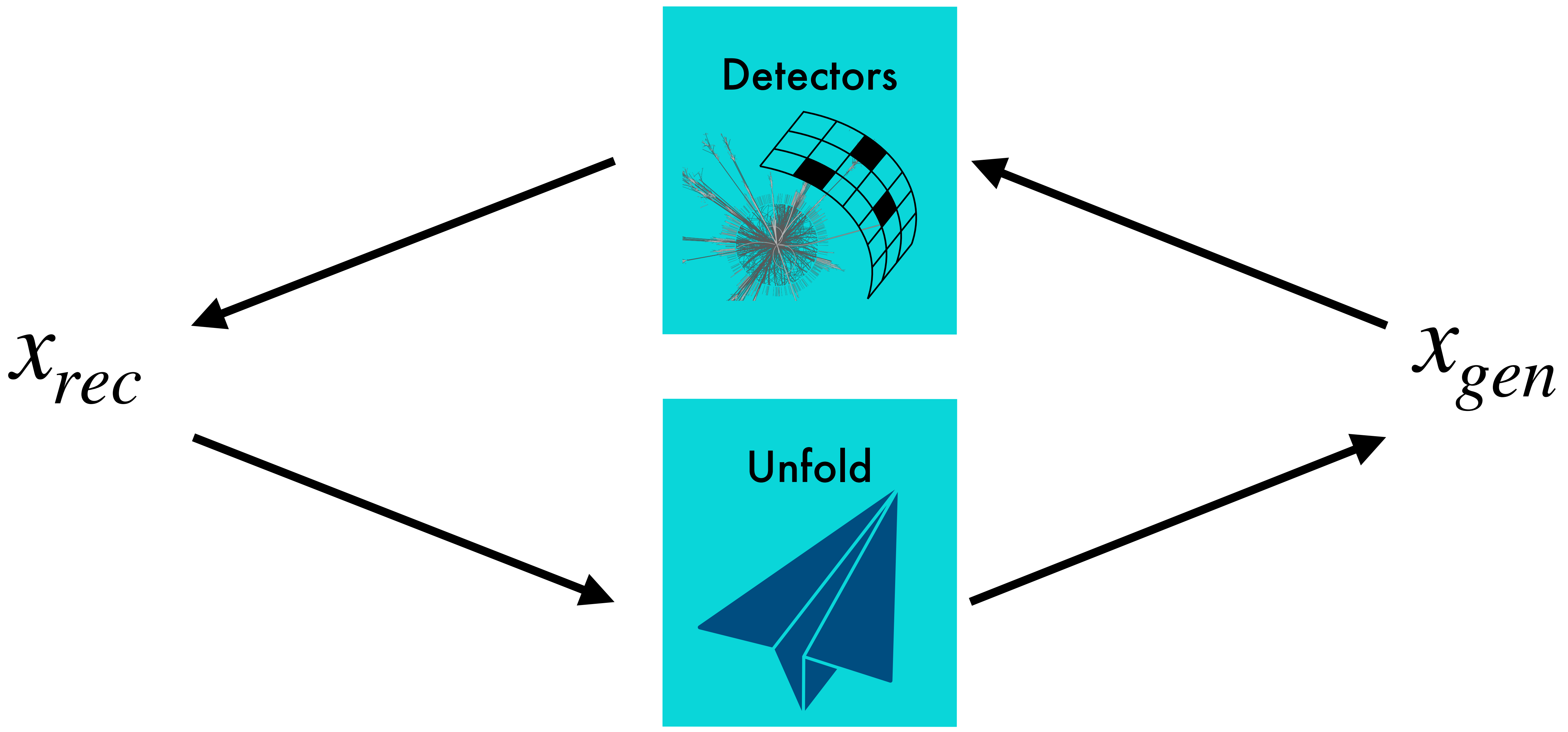
Figure from A. Butter et al.: arXiv:2203.07460, R. Winterhalder

Unfolding — a toy example

Consider binned distributions at gen and rec level

Describe detector effects by response matrix ***R***

Unfold by (pseudo-)inverting the matrix



$$h^i_{rec} = R^{ij}\, h^j_{gen}$$

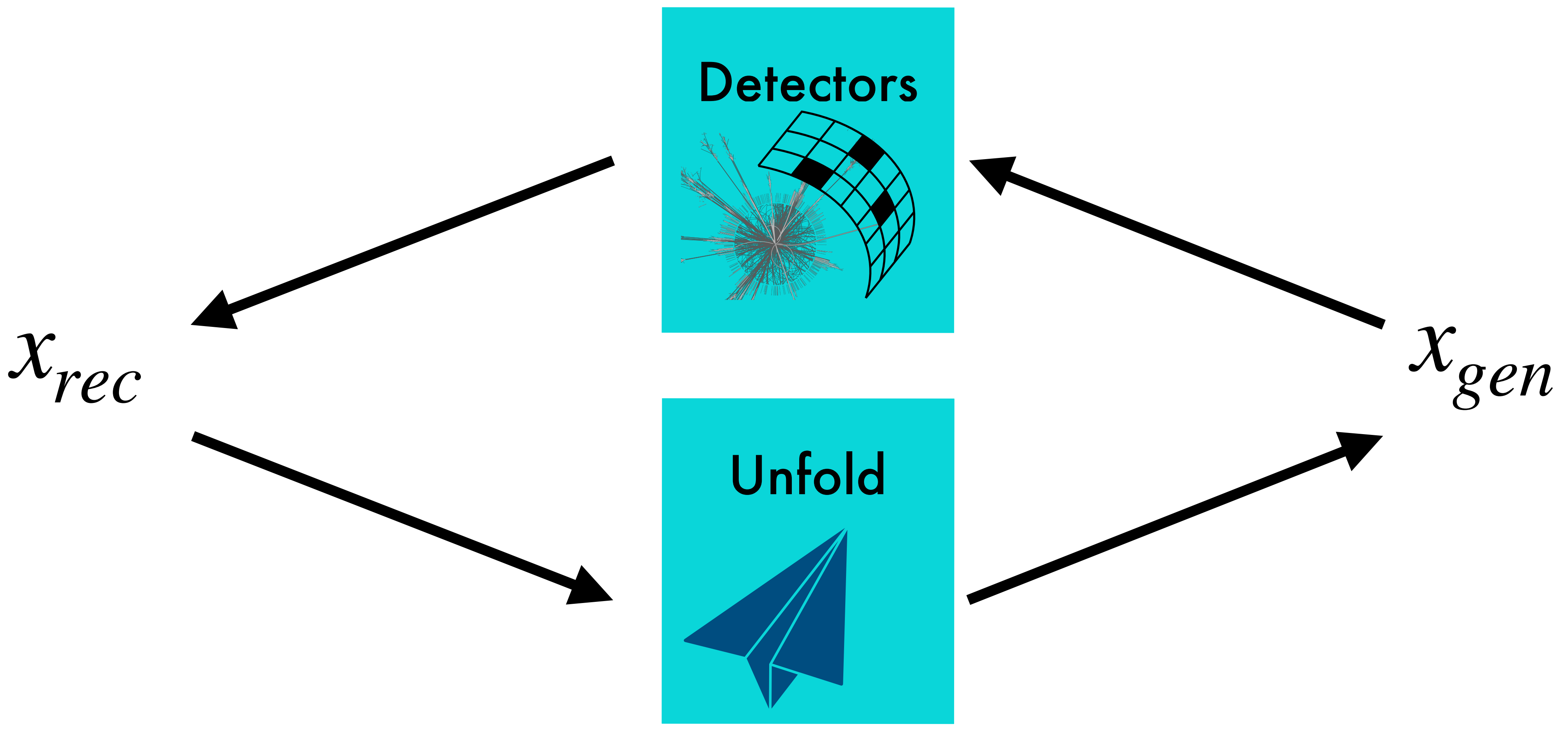
$$h^j_{gen} = \left(R^{-1}\right)^{ji}\, h^i_{rec}$$

Unfolding — a toy example

Detector: local, linear smearing

$$R = \begin{pmatrix} 1-\epsilon & \epsilon & 0 \\ \epsilon & 1-2\epsilon & \epsilon \\ 0 & \epsilon & 1-\epsilon \end{pmatrix}$$

$$h_{rec} = R\; h_{gen}$$



$$h_{gen} = R^{-1}\; h_{rec}$$

Inversion: non-local, non-linear

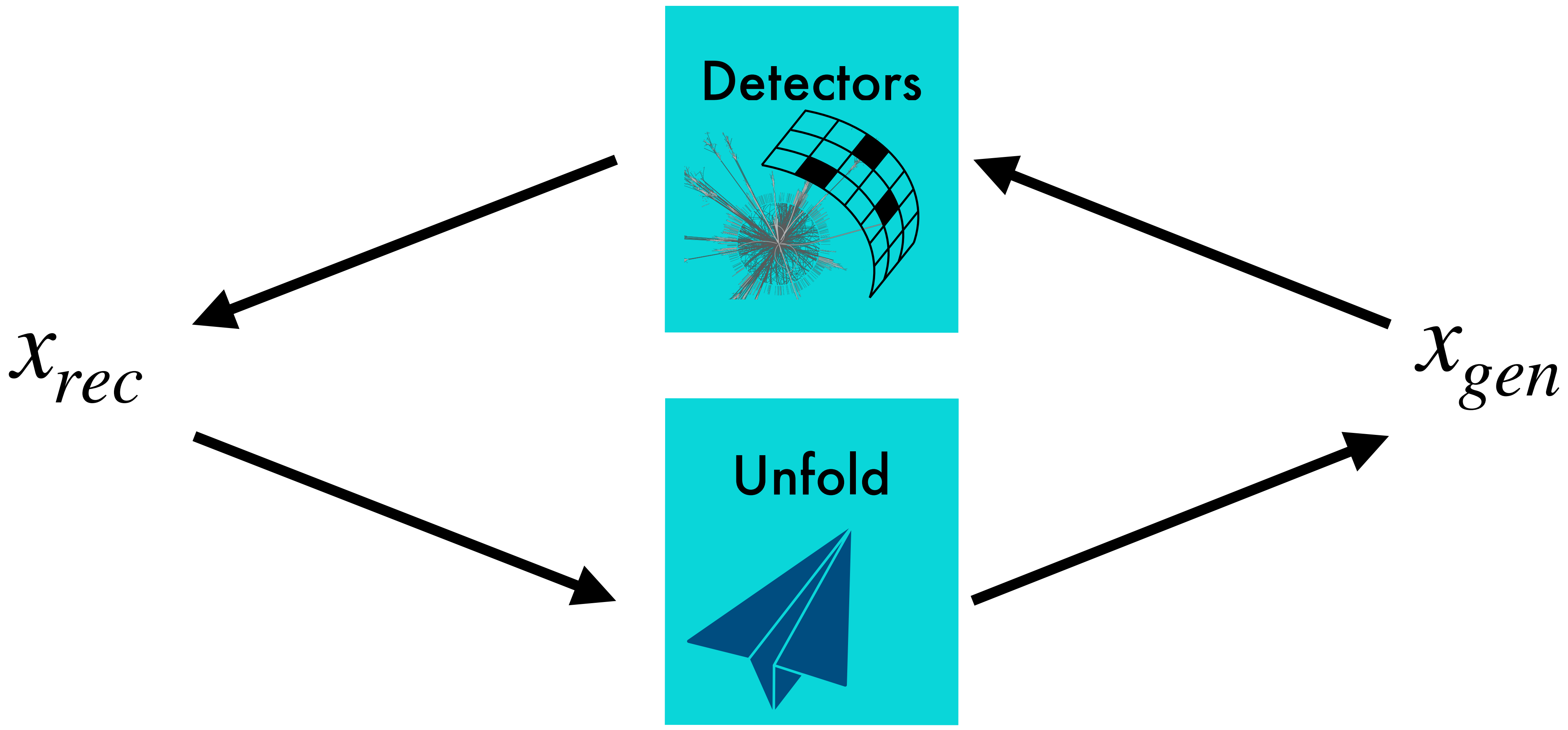
$$R^{-1} \approx \frac{1}{1-4\epsilon} \begin{pmatrix} 1-3\epsilon & -\epsilon & \epsilon^2 \\ -\epsilon & 1-2\epsilon & -\epsilon \\ \epsilon^2 & -\epsilon & 1-3\epsilon \end{pmatrix}$$

(squared terms dropped when linear term non zero)

Example from Bellagente et al.: arXiv:2006.06685

Unfolding — classical methods

$$h_{rec} = R\, h_{gen}$$



$$h_{gen} = R^{-1}\, h_{rec}$$

Classical unfolding methods: binned, one-dimensional

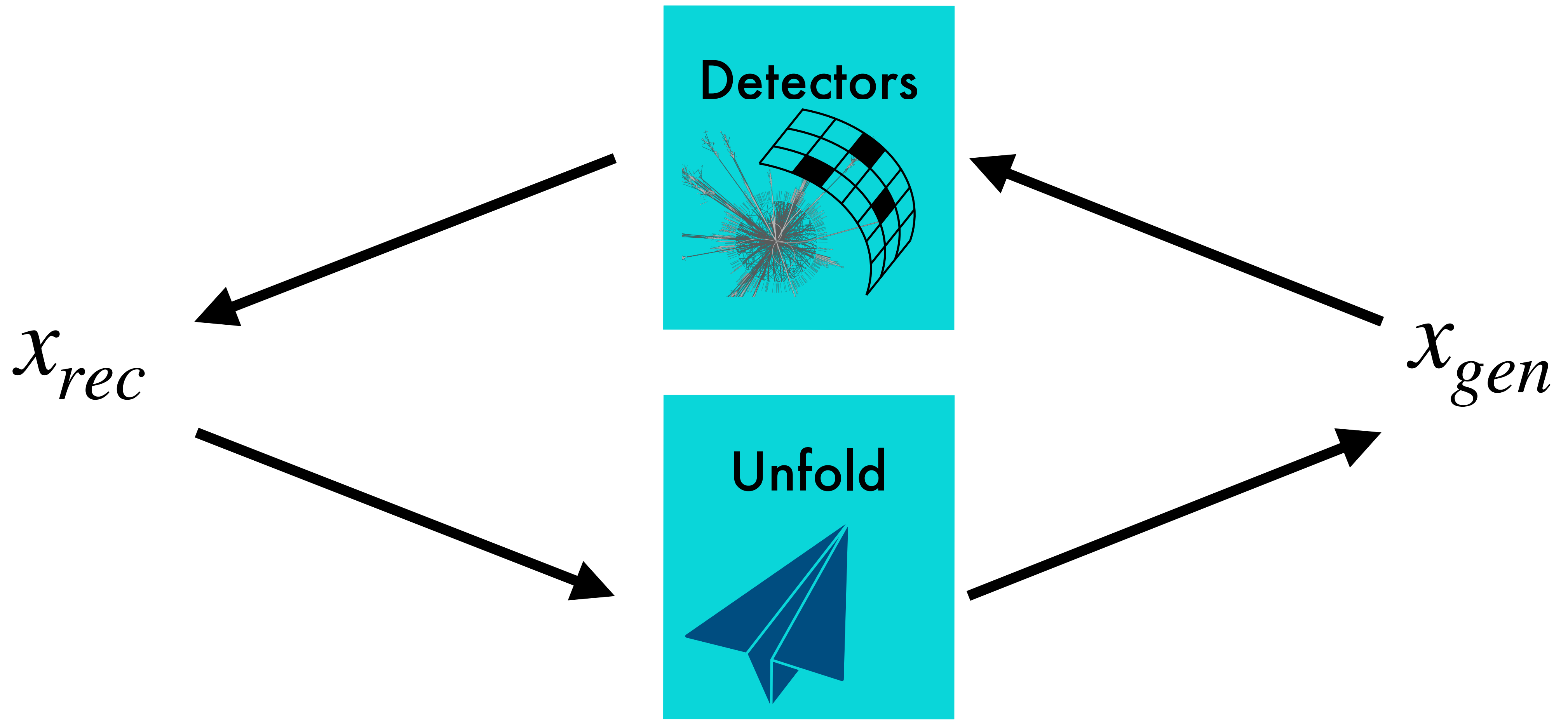
Using inverse matrix gives very high variance

<u>In praxis</u>: regularised methods are used

Unfolding — unchained

$$p(x_{rec}) = \int p(x_{gen})\underbrace{R(x_{rec}, x_{gen})}_{\mathbf{p(x_{rec}\,|\,x_{gen})}}\, dx_{gen}$$

$$p(x_{gen}) = \int p(x_{rec})\underbrace{p(x_{gen}\,|\,x_{rec})}_{\textbf{target probability}}\, dx_{rec}$$



Unfolding — unchained

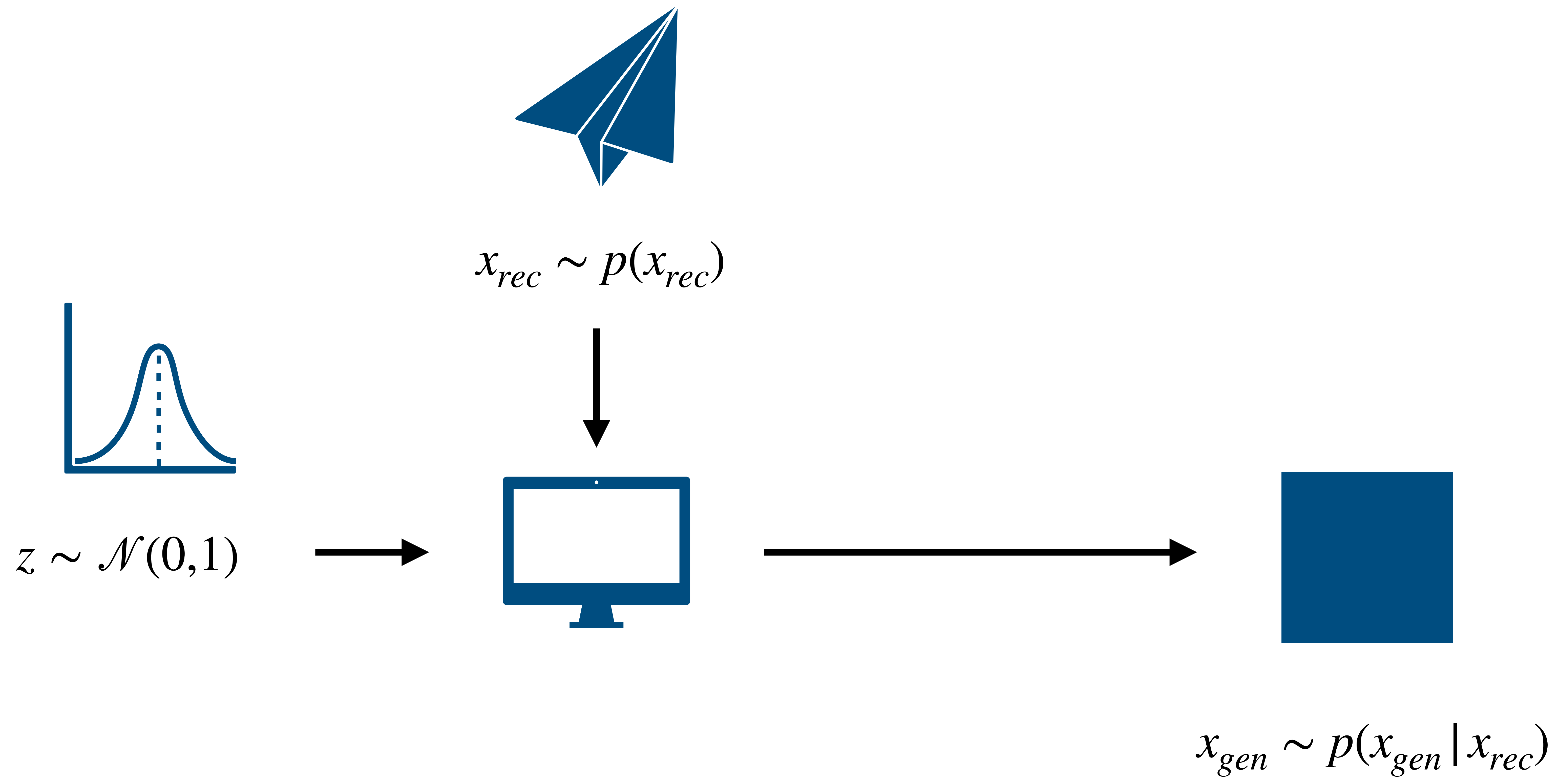
$$p(x_{rec}) = \int p(x_{gen}) \underbrace{R(x_{rec}, x_{gen})}_{\mathbf{p(x_{rec} | x_{gen})}} \, dx_{gen}$$

$$p(x_{gen}) = \int p(x_{rec}) \underbrace{p(x_{gen} | x_{rec})}_{\text{target probability}} \, dx_{rec}$$

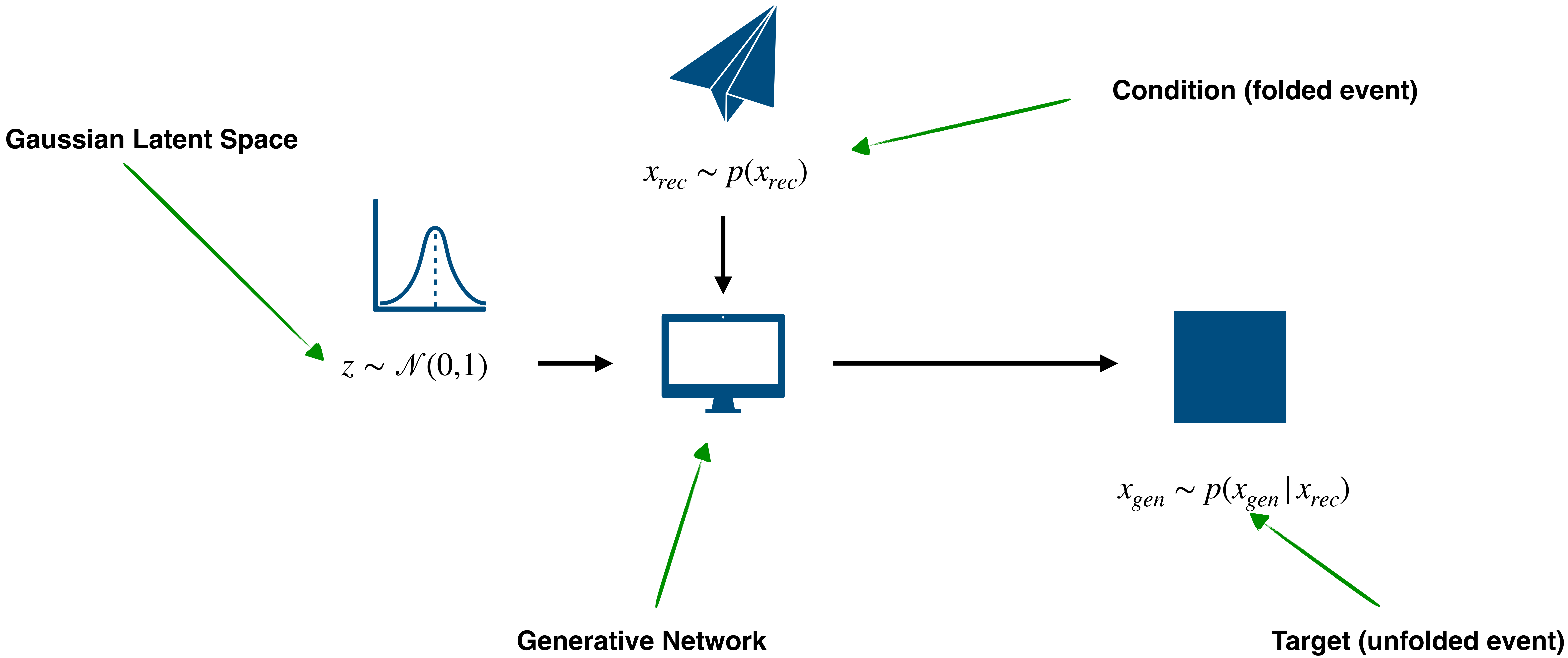
Classical methods are restricted to binned, one-dimensional distributions

We would like to learn high-dimensional, unbinned unfolding probability

Unfolding — generative methods



Unfolding — generative methods

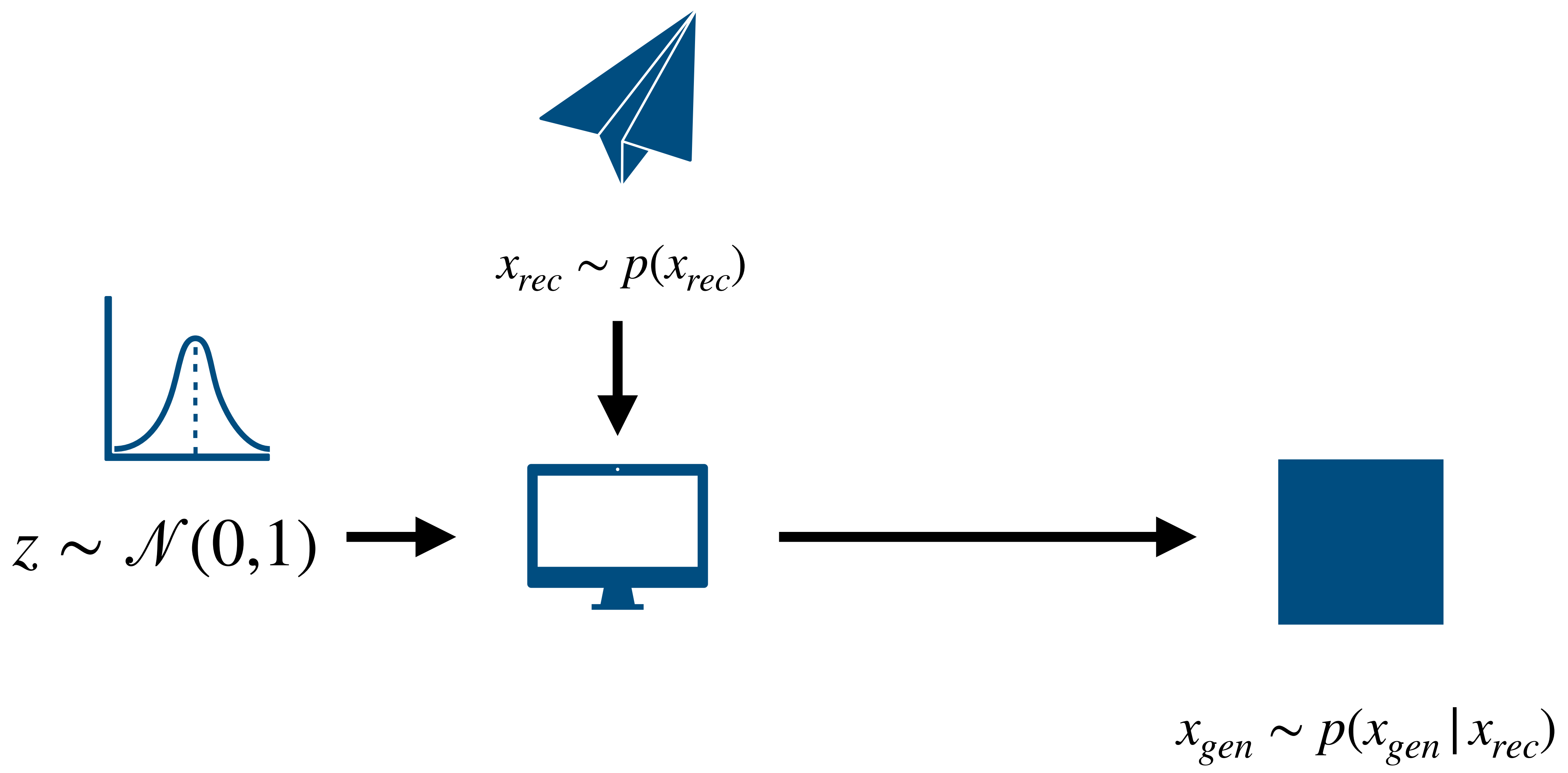


Unfolding — generative methods

Goal: learn transformation latent → gen phase space conditioned on rec event

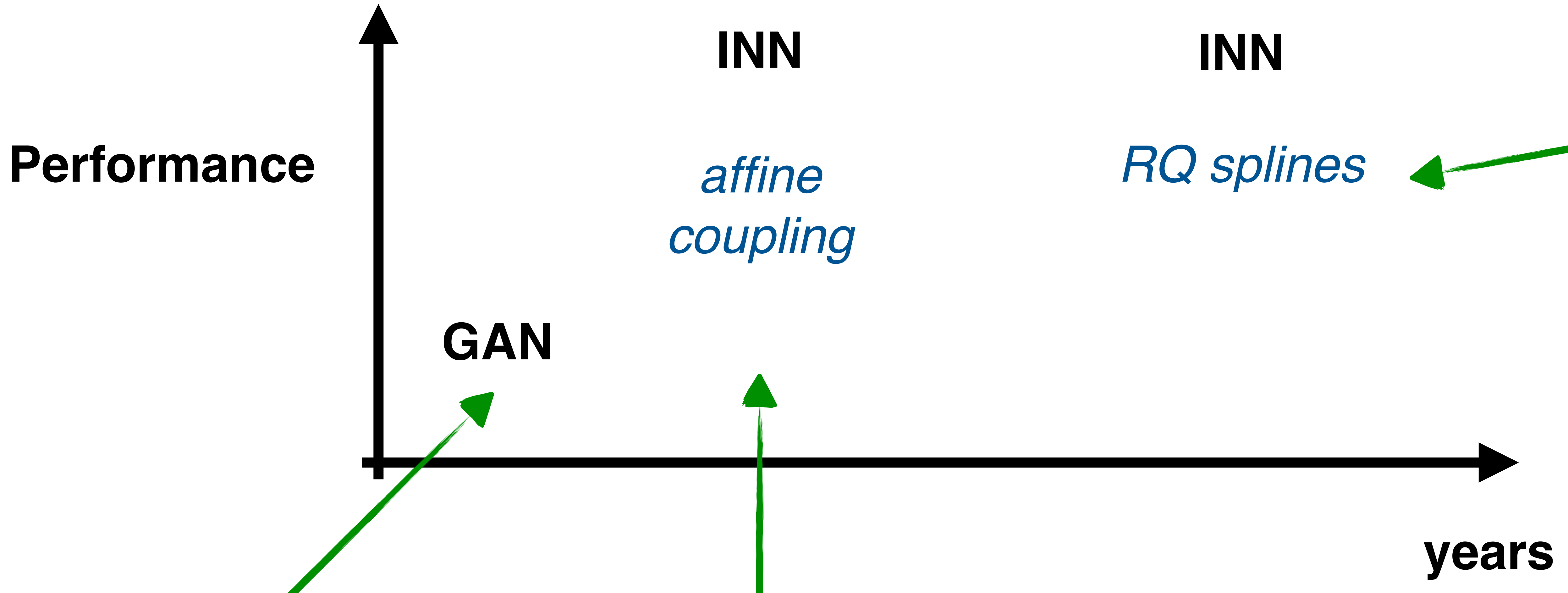
During training, use paired events of forward simulation

After training, repeated sampling from latent space with constant condition allows probabilistic single event unfolding



Intermezzo — Generative Networks

Which generative Network?



An unfolding method based on Conditional Invertible Neural Networks (cINN) using iterative training

Backes et al.
arXiv:2212.08674

How to GAN away Detector Effects

Bellagente et al.
arXiv:1912.00477

Invertible Networks or from Partons to Detector and back again

Bellagente et al.
arXiv:2006.06685

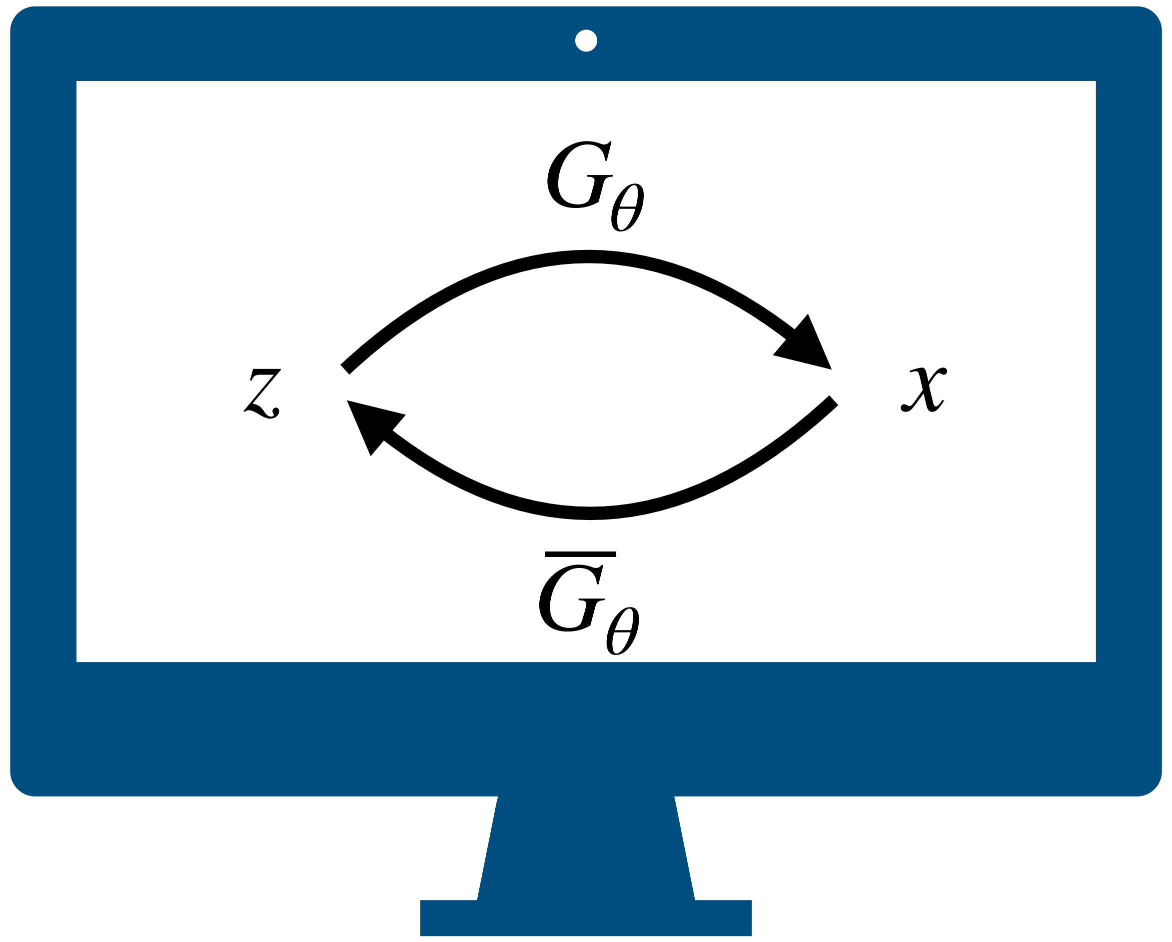
Returning CP - Observables to the Frames they Belong

Ackerschott et al.
arXiv:2308.00027

Event-by-event Comparison between Machine-Learning- and Transfer-Matrix-based Unfolding Methods

Backes et al.
arXiv:2310.17037

Intermezzo — Generative Networks



INN

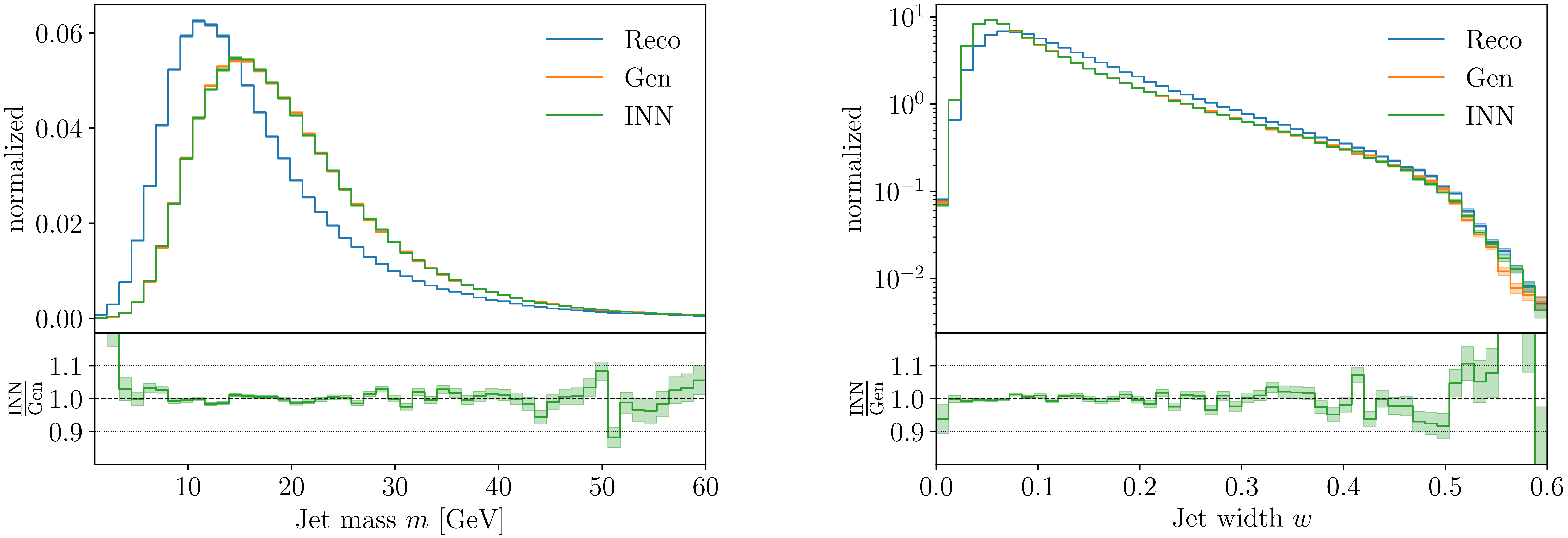
Target probability $p(x_{gen} | x_{rec})$

Train on conditional likelihood loss

$$\mathscr{L} = -\left\langle \log p_\theta(x_{gen} | x_{rec}) \right\rangle_{x_{gen}, x_{rec} \sim p(x_{gen}, x_{rec})}$$

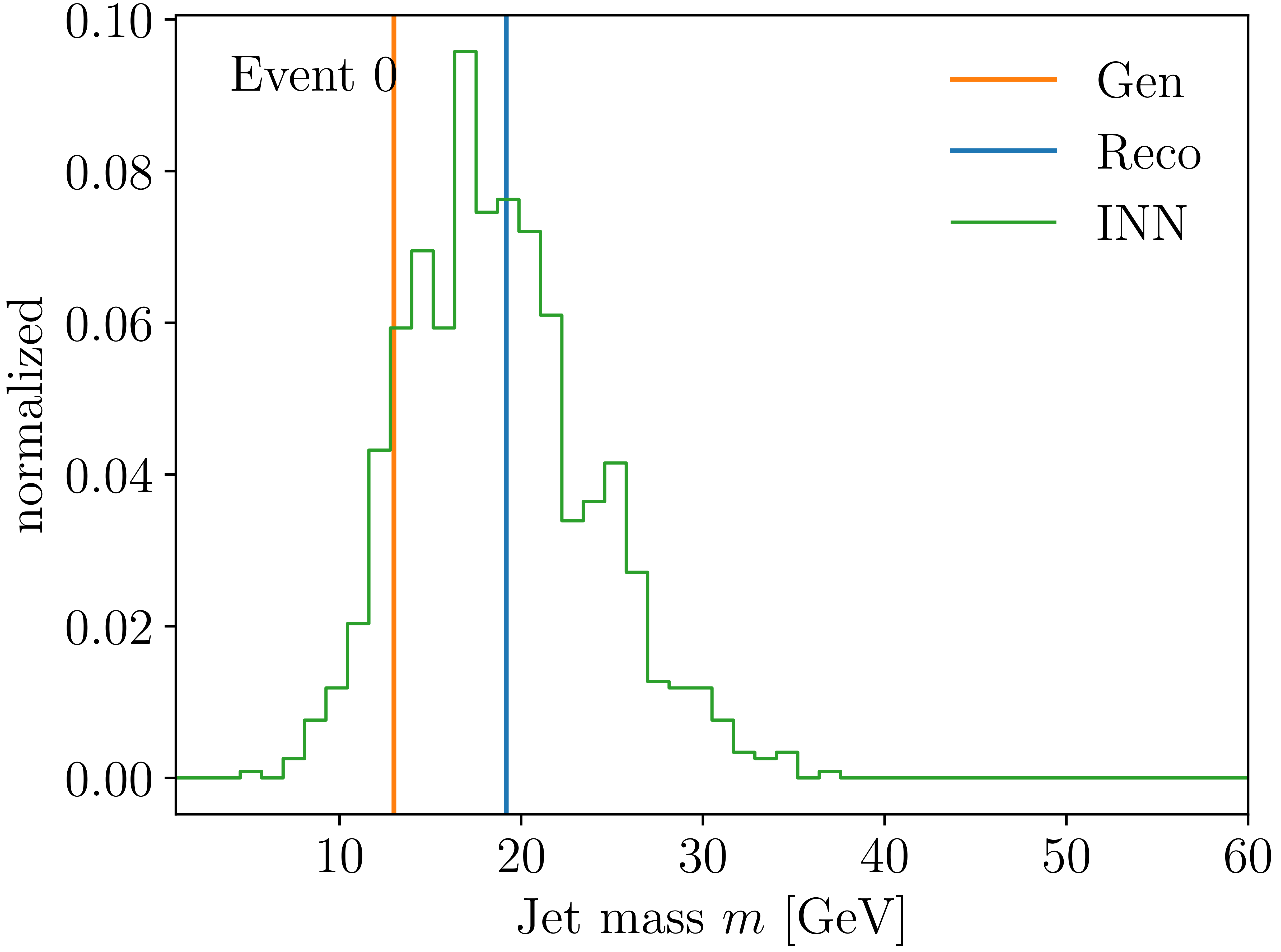
$$= -\left\langle \log \left(p_{latent} \left(\overline{G}_\theta(x_{gen} | x_{rec}) \right) \right) + \log \left| \frac{\partial \overline{G}_\theta(x_{gen} | x_{rec})}{\partial x_{gen}} \right| \right\rangle_{x_{gen}, x_{rec} \sim p(x_{gen}, x_{rec})}$$

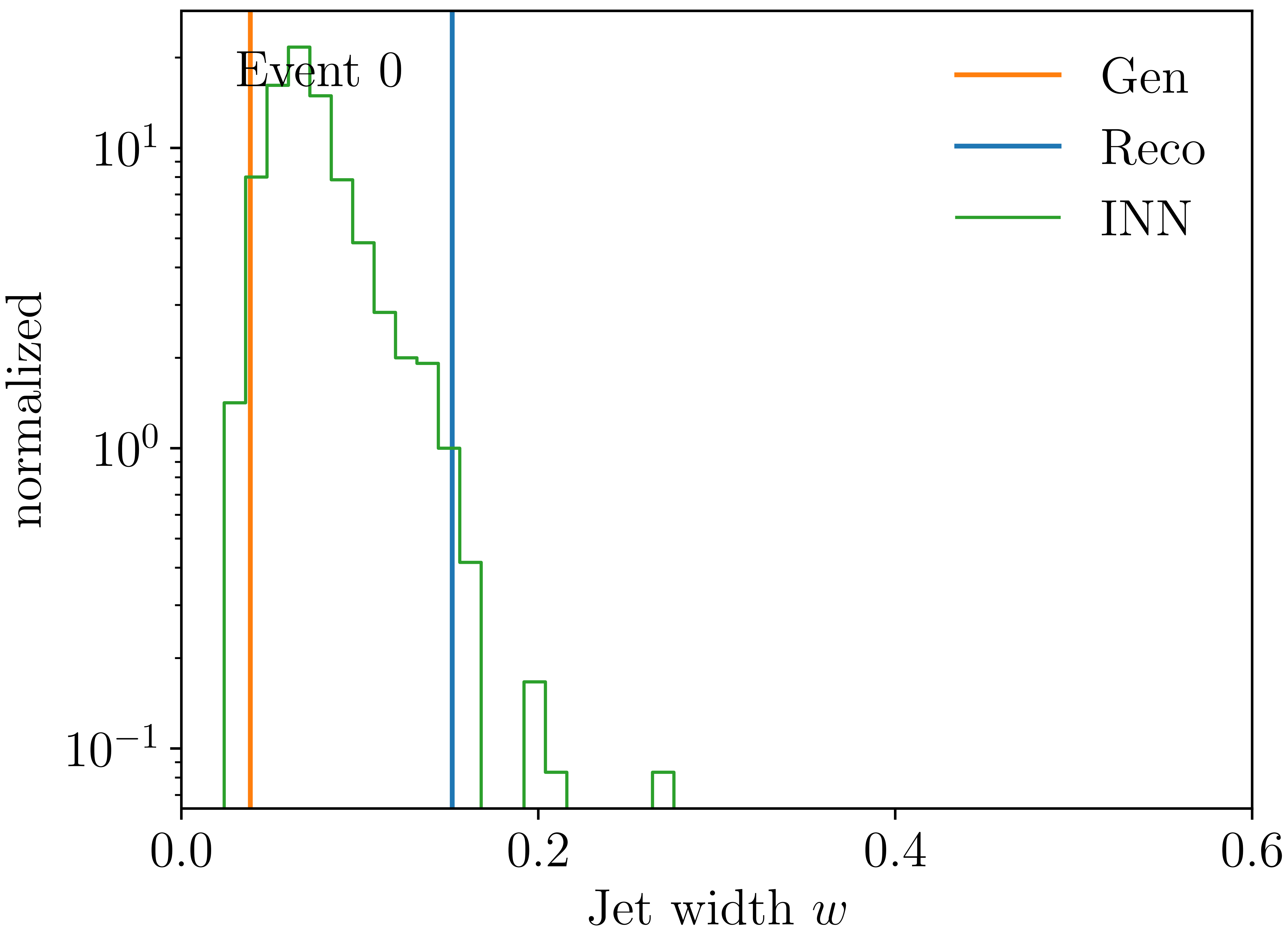
Z + jet example



Data from arXiv:1911.09107, Figure from Huetsch et. al: arXiv:24xx.xxxxx

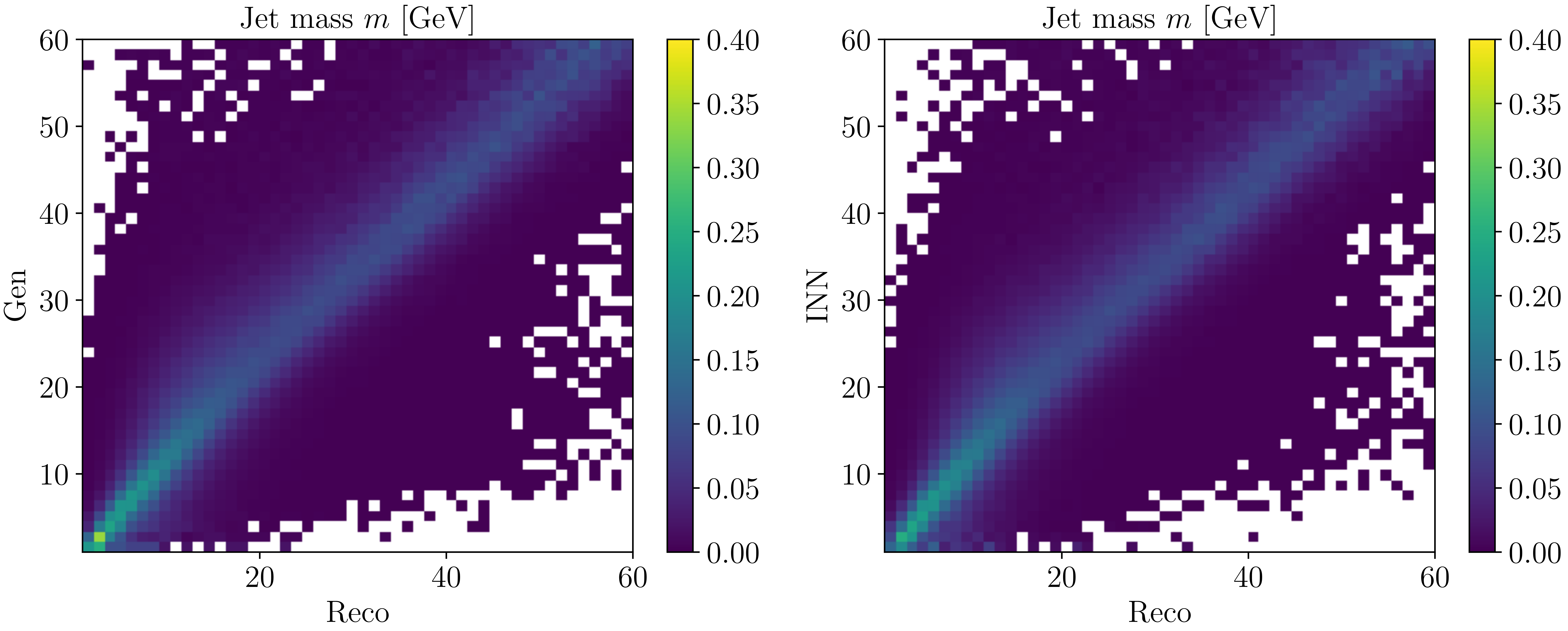
Z + jet example — single event unfolding





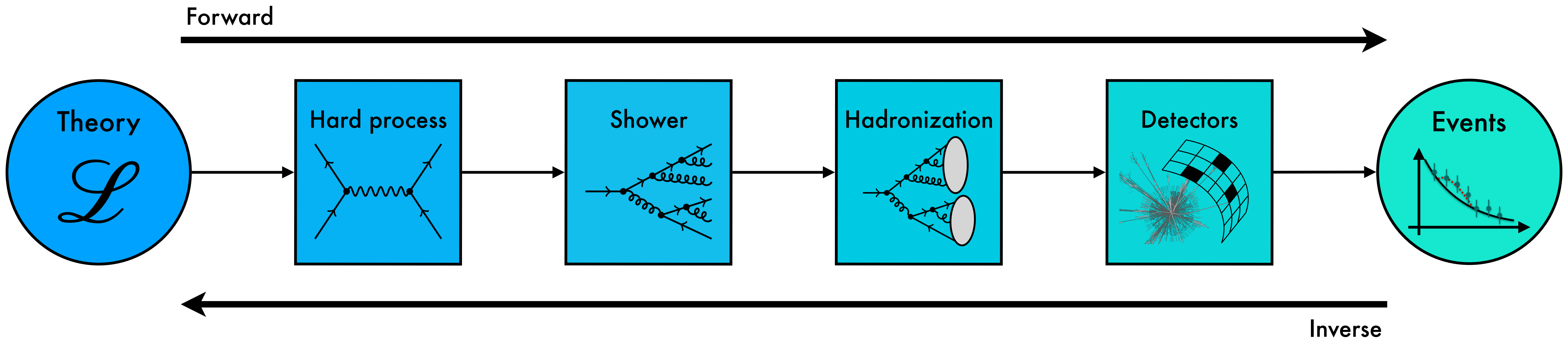
Data from arXiv:1911.09107, Figure from Huetsch et. al: arXiv:24xx.xxxxx

Z + jet example — migration



Data from arXiv:1911.09107, Figure from Huetsch et. al: arXiv:24xx.xxxxx

Parton Level Unfolding — Z + 2 jets



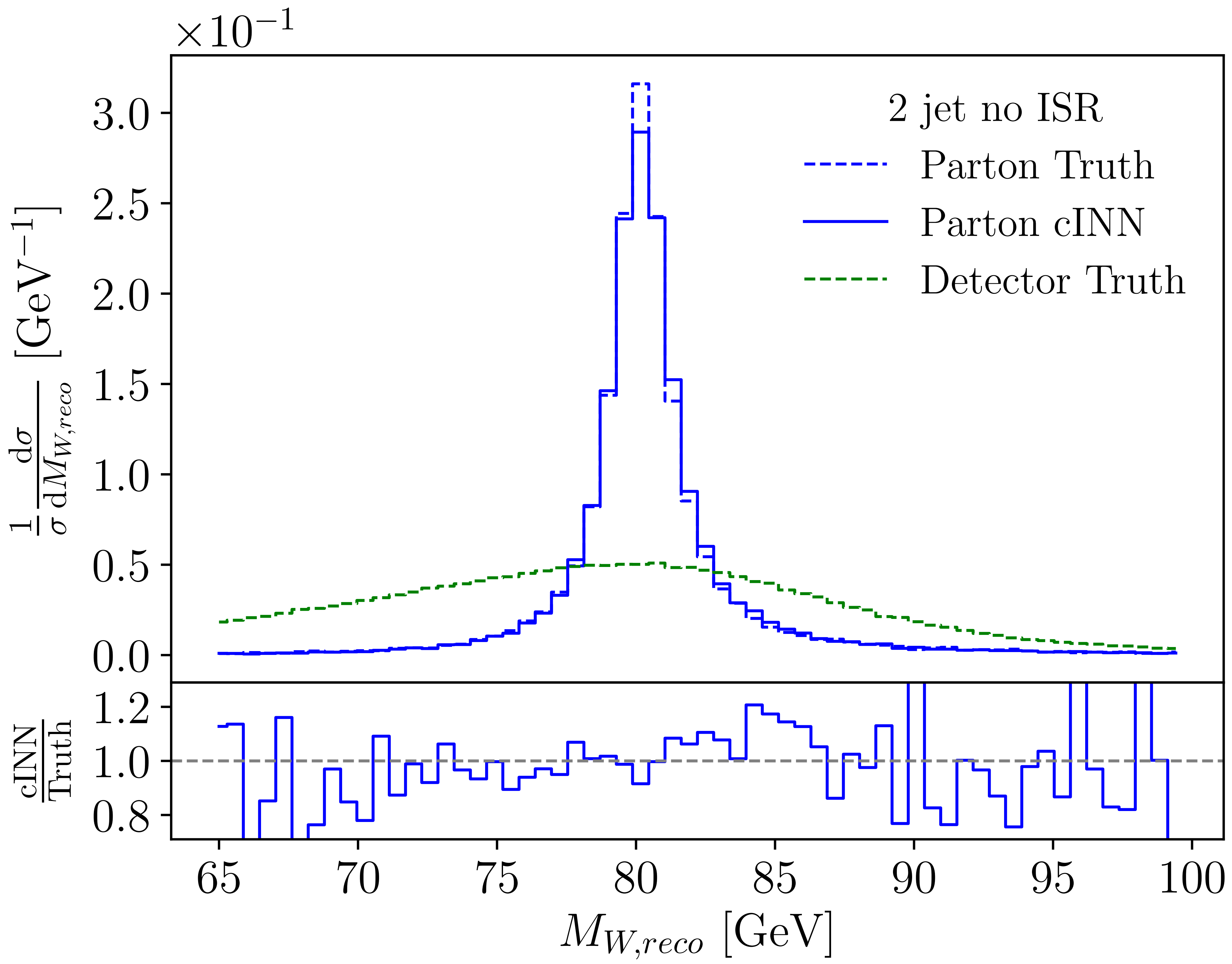


Figure top from A. Butter et al.: arXiv:2203.07460, R. Winterhalder, Figure bottom from Bellagente et al. arXiv:2006.06685

What about model dependence?

Unfolding probability is Bayesian posterior = we have a prior

Prior is gen level distribution in training data (MC simulation)

Problematic for large MC-data differences

Prior

$$p(x_{gen} | x_{rec}) = \frac{p(x_{rec} | x_{gen}) p(x_{gen})}{p(x_{rec})}$$

Iterative Bayesian Unfolding (IBU)

Unfolding probability is Bayesian posterior =
we have a prior

Prior is gen level distribution in training data
(MC simulation)

Problematic for large MC-data differences

$$p(x_{gen}) = \int p(x_{rec})p(x_{gen}\,|\,x_{rec})\;dx_{rec}$$

Iterative Bayesian Unfolding (IBU)

Unfolding probability is Bayesian posterior = we have a prior

Prior is gen level distribution in training data (MC simulation)

Problematic for large MC-data differences

Unknown

$$p(x_{gen}) = \int p(x_{rec}) \boxed{p(x_{gen} \,|\, x_{rec})} \, dx_{rec}$$

Using Bayes' Theorem

Prior

$$p(x_{gen}) = \int p(x_{rec}) \frac{p(x_{rec} \,|\, x_{gen}) p(x_{gen})}{\int p(x_{rec} \,|\, \tilde{x}_{gen}) p(\tilde{x}_{gen}) d\tilde{x}_{gen}} \, dx_{rec}$$

Iterative Bayesian Unfolding (IBU)

Unfolding probability is Bayesian posterior = we have a prior

Prior is gen level distribution in training data (MC simulation)

Problematic for large MC-data differences

IBU idea: Update your prior after each iteration

Unknown

$$p(x_{gen}) = \int p(x_{rec}) \boxed{p(x_{gen}|x_{rec})}\, dx_{rec}$$

Using Bayes' Theorem

$$p(x_{gen}) = \int p(x_{rec}) \frac{p(x_{rec}|x_{gen})p(x_{gen})}{\int p(x_{rec}|\tilde{x}_{gen})p(\tilde{x}_{gen})d\tilde{x}_{gen}}\, dx_{rec}$$

Start from prior and use iterative approach

$$p^n(x_{gen}) = \int p(x_{rec}) \frac{p(x_{rec}|x_{gen})p^{n-1}(x_{gen})}{\int p(x_{rec}|\tilde{x}_{gen})p^{n-1}(\tilde{x}_{gen})d\tilde{x}_{gen}}\, dx_{rec}$$

Iterative Bayesian Unfolding (IBU)

Unfolding probability is Bayesian posterior = we have a prior

Prior is gen level distribution in training data (MC simulation)

Problematic for large MC-data differences

IBU idea: Update your prior after each iteration

$$p^n(x_{gen}) = \int p(x_{rec}) \frac{p(x_{rec} \mid x_{gen}) p^{n-1}(x_{gen})}{\int p(x_{rec} \mid \tilde{x}_{gen}) p^{n-1}(\tilde{x}_{gen}) d\tilde{x}_{gen}} \, dx_{rec}$$

In the past for LHC-like phase spaces no access to those quantities

$$h^j_{gen,n} = \sum_i h^i_{rec} \frac{R_{ij} \, h^j_{gen,n-1}}{\sum_k R_{ik} \, h^k_{gen,n-1}}$$

One dimensional & binned

Iterative generative unfolding

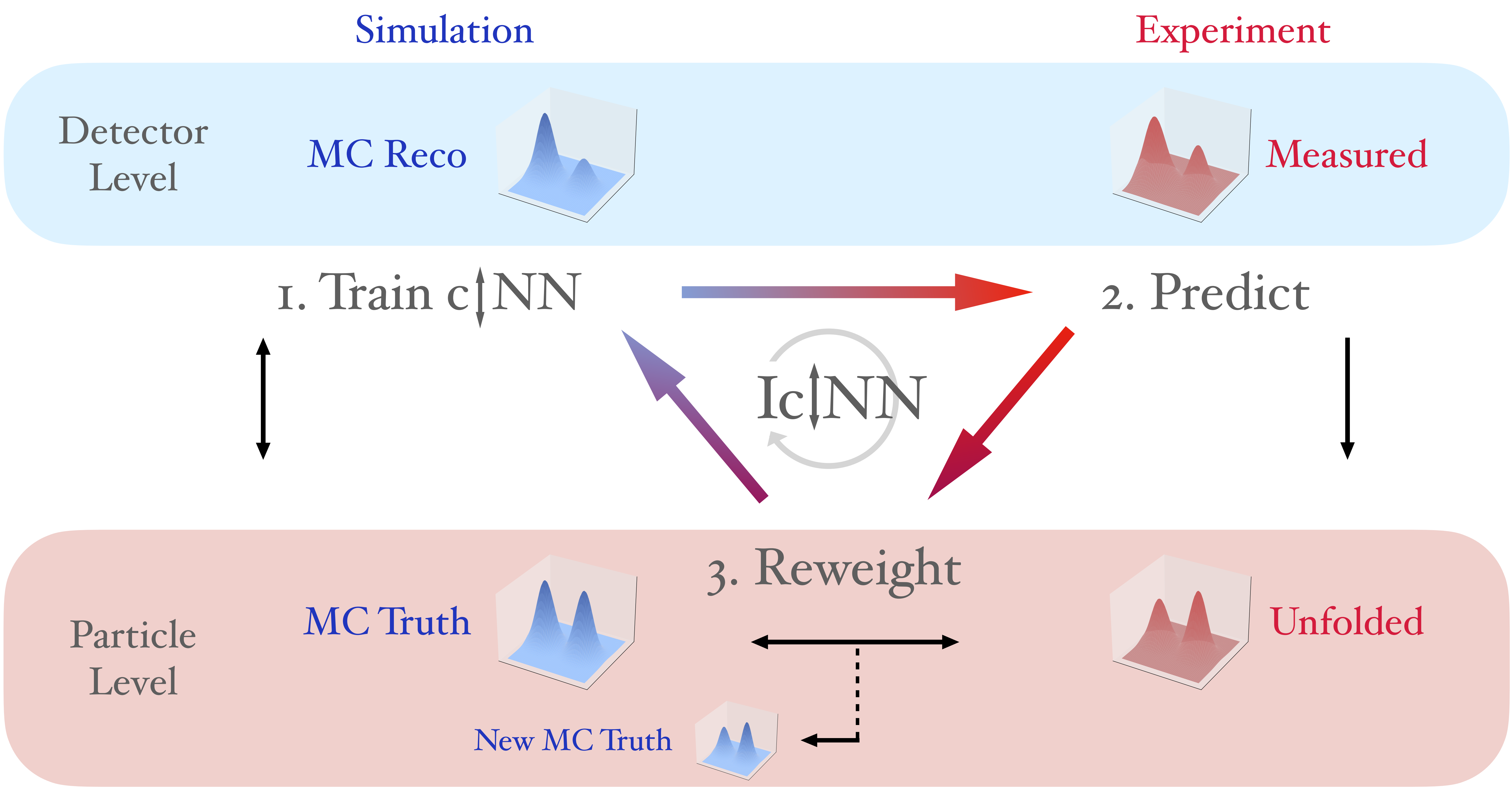
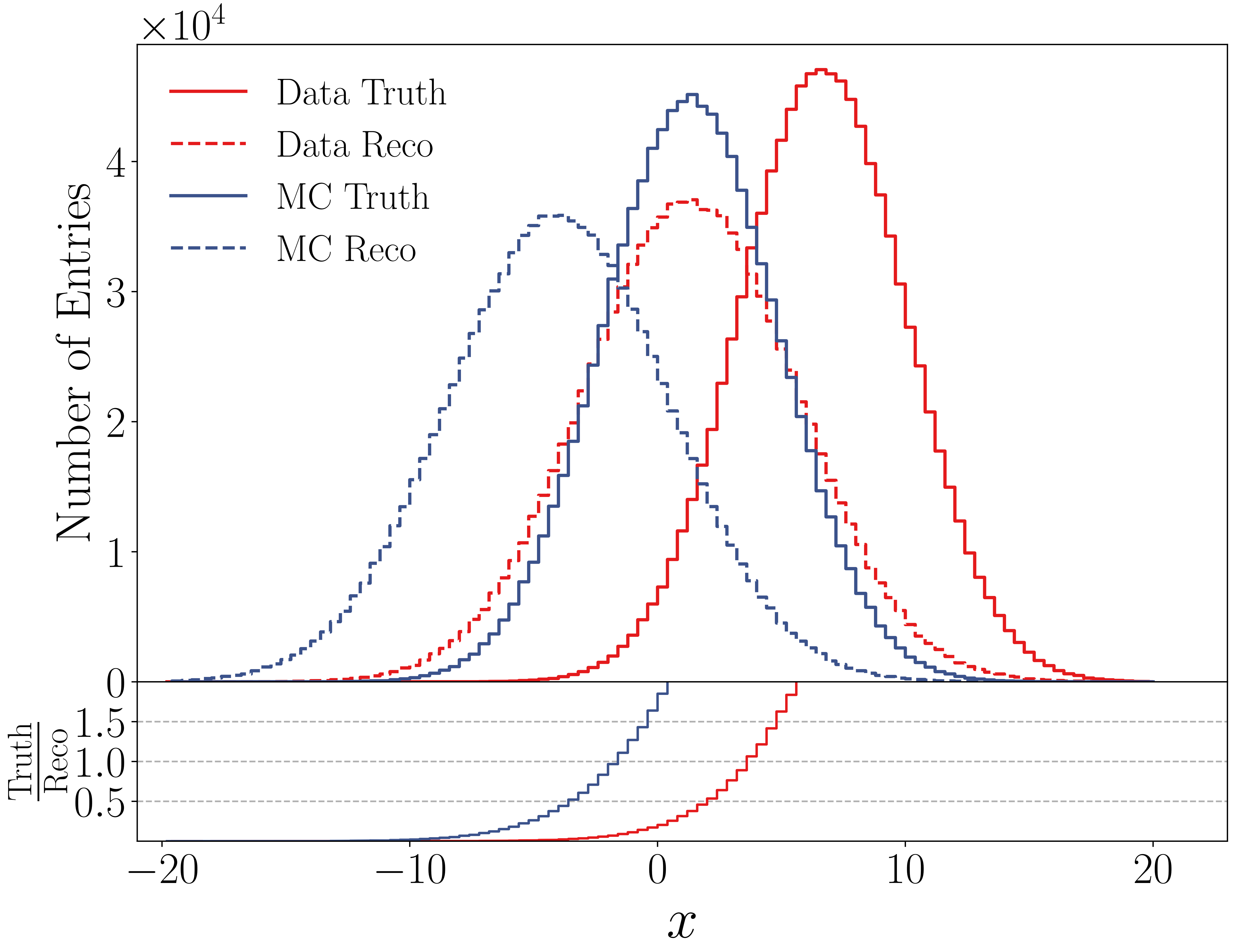
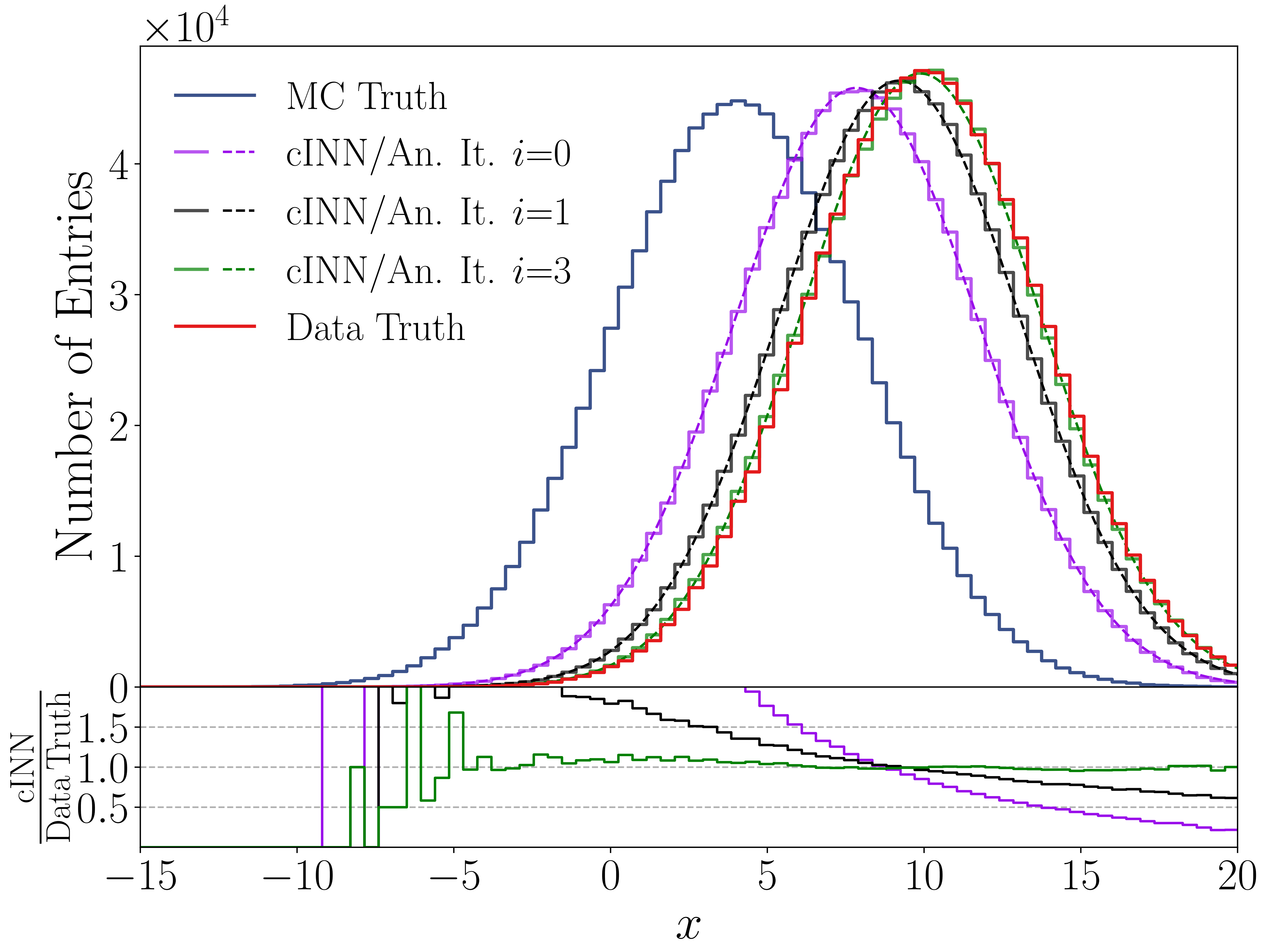


Figure from Backes et al.: arXiv:2212.08674

Iterative generative unfolding





Figures from Backes et al.: arXiv:2212.08674

Generative Unfolding in experimental praxis

Failure modes in model dependency

All necessary information to unfold could be lost

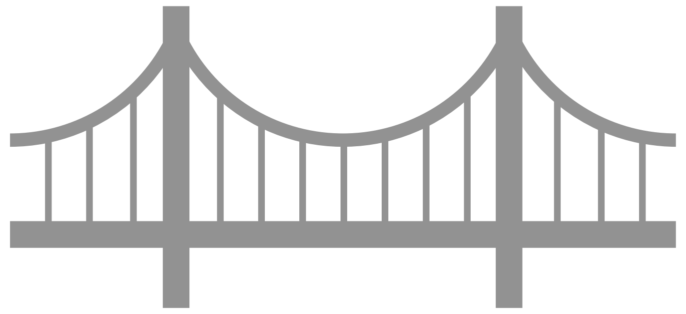
Background subtraction ?

Finite detector efficiency

Sideband studies

Combinatorics

…



Currently, working together with experimentalist bridging gap between theory and praxis

And now what?

Generative machine learning allows for unbinned, high dimensional unfolding

Methods to unbias unfolding networks exists and are currently tested to match experimental requirements

In parallel, there is an on going project comparing different ML based unfolding methods

Stay tuned!