UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

France-Berkeley PHYSTAT Conference on Unfolding
10.06.2024

Generative Unfolding

SPONSORED BY THE

Federal Ministry
of Education
and Research

Nathan Huetsch

Huetsch et al. 2404.18807
The Landscape of Unfolding with Machine Learning

Simulation chain — Inversion

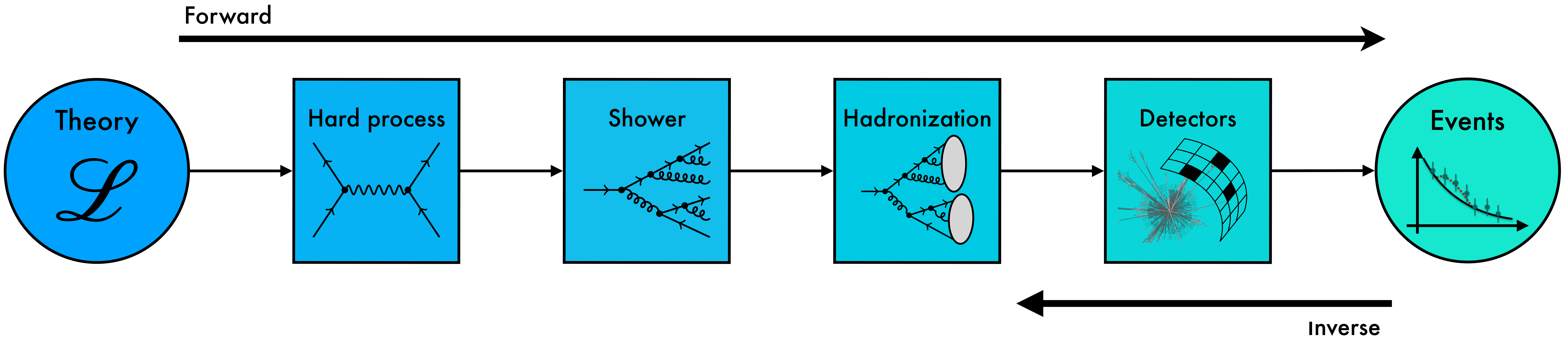


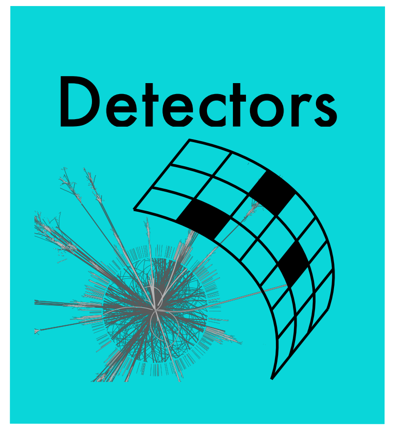
Figure from A. Butter et al.: arXiv:2203.07460, R. Winterhalder

Probabilistic transfer

Transfer probability

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

x_{gen}



x_{rec}

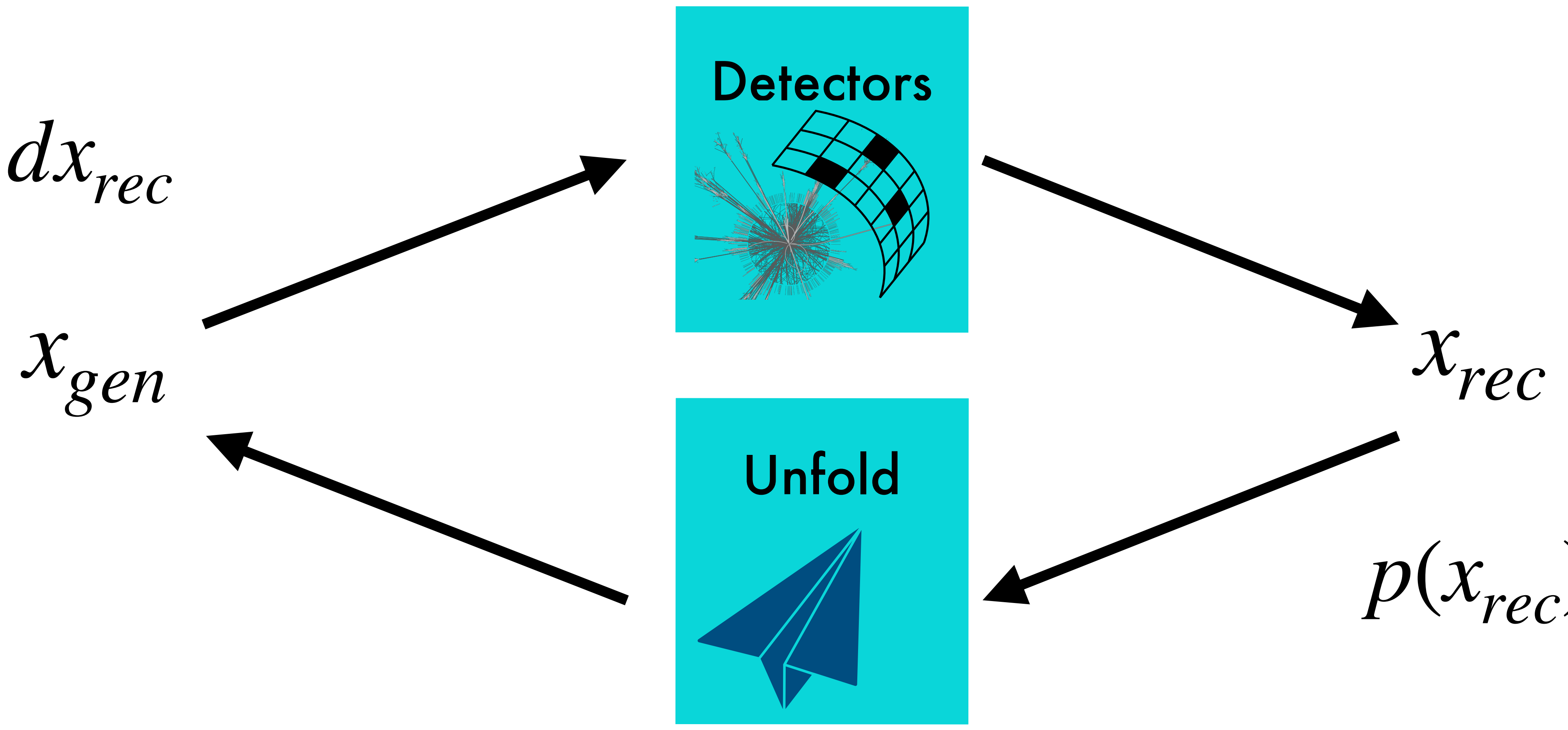
$$p(x_{rec}) = \int p(x_{gen}) p(x_{rec} | x_{gen}) \, dx_{gen}$$

Generative unfolding

Transfer probability

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

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



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

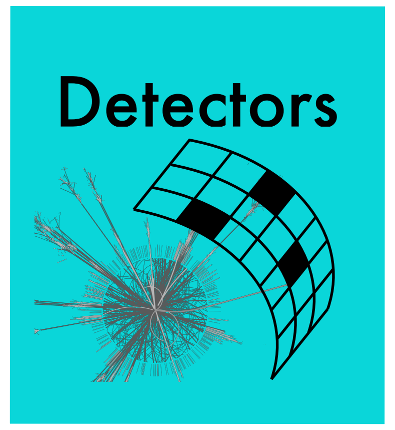
$$p(x_{gen} | x_{rec})$$

Unfolding probability

Generative unfolding

Transfer probability ← **Encoded implicitly in detector simulation**

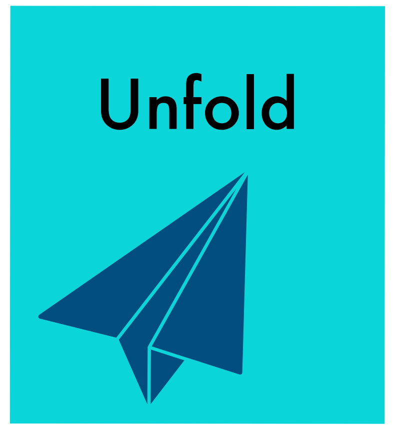
$$p(x_{rec}|x_{gen})$$



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

x_{gen}

x_{rec}

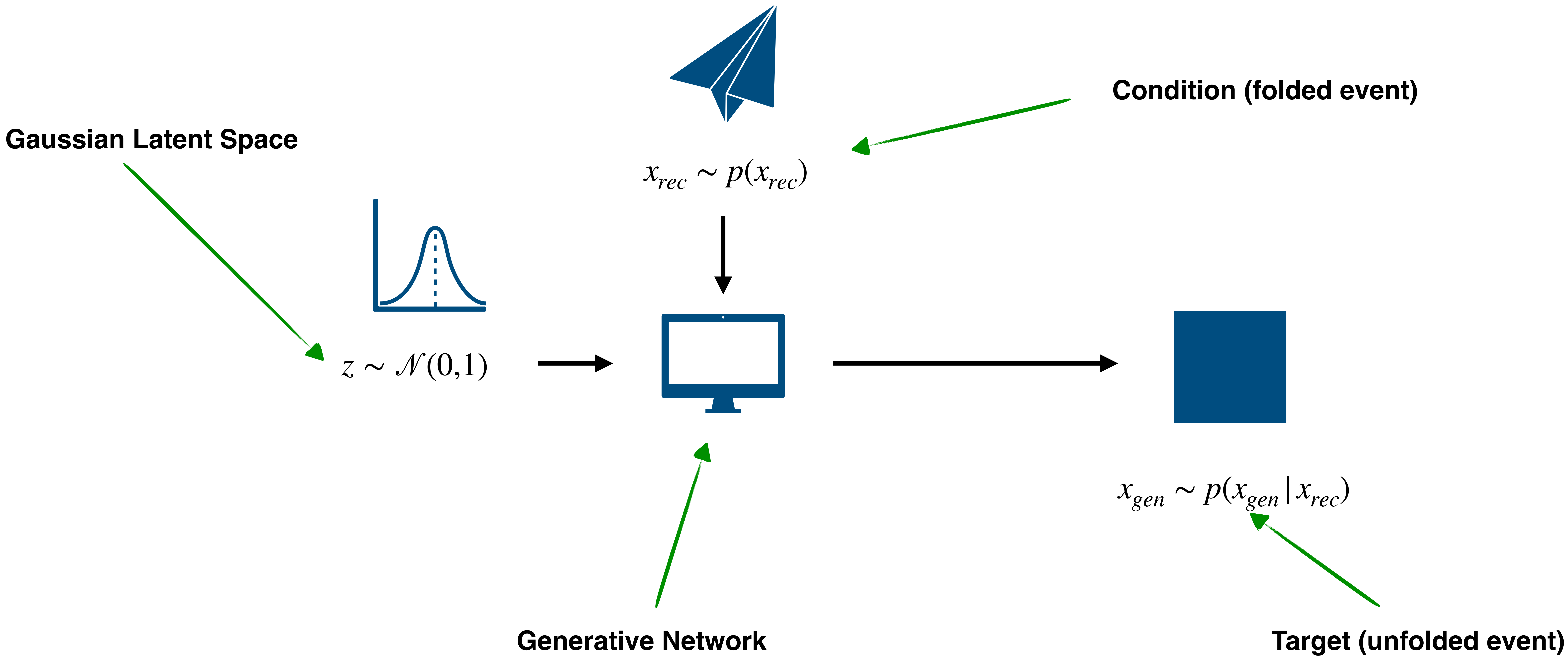


$$p(x_{rec}) = \int p(x_{gen})p(x_{rec}|x_{gen})\, dx_{gen}$$

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

Unfolding probability ← **Encoded in neural network trained on simulation**

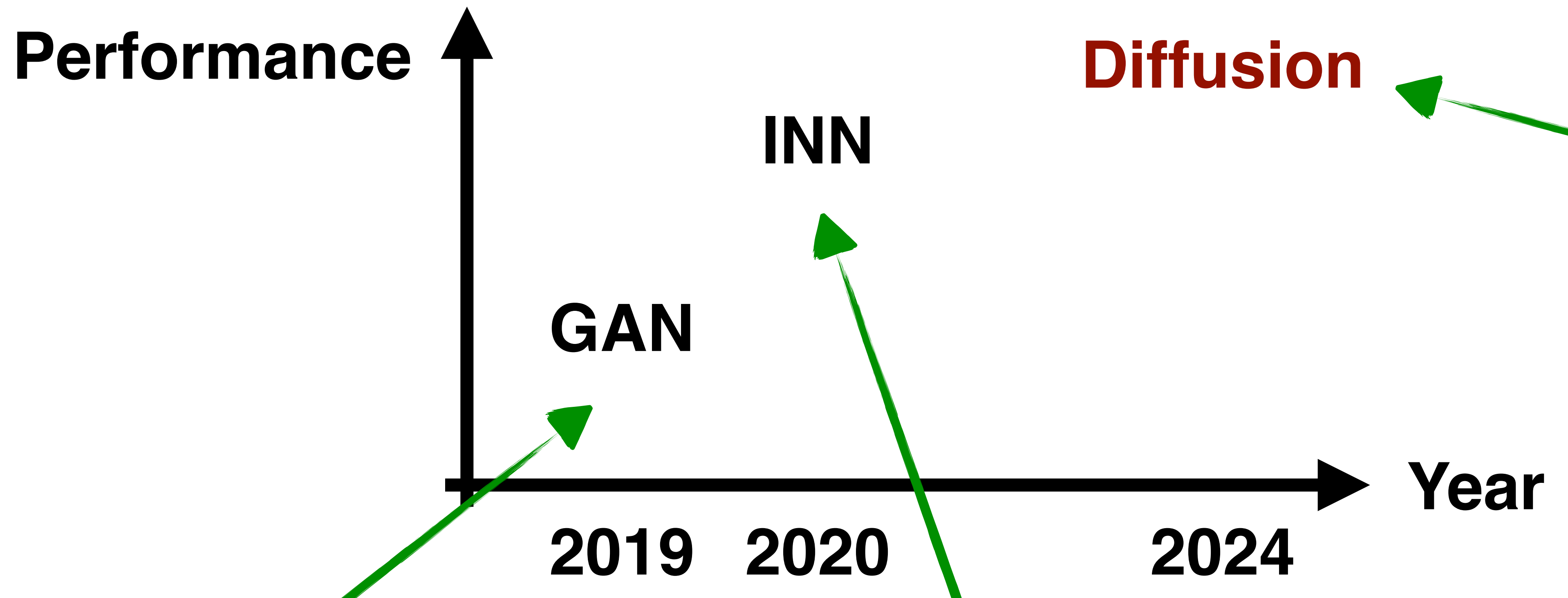
Conditional generative networks



Slide Credit: Sofia Palacios Schweitzer

Conditional generative networks

Which generative Network?



Bellagente et al.
arXiv:1912.00477

Bellagente et al.
arXiv:2006.06685

CFM
Huetsch et al.
arXiv:2404.18807

VLD
Shmakov et al.
arXiv: 2305.10399
arXiv: 2404.14332

DDPM
Pazos et al.
arXiv: 2406.01507

Talks on thursday!

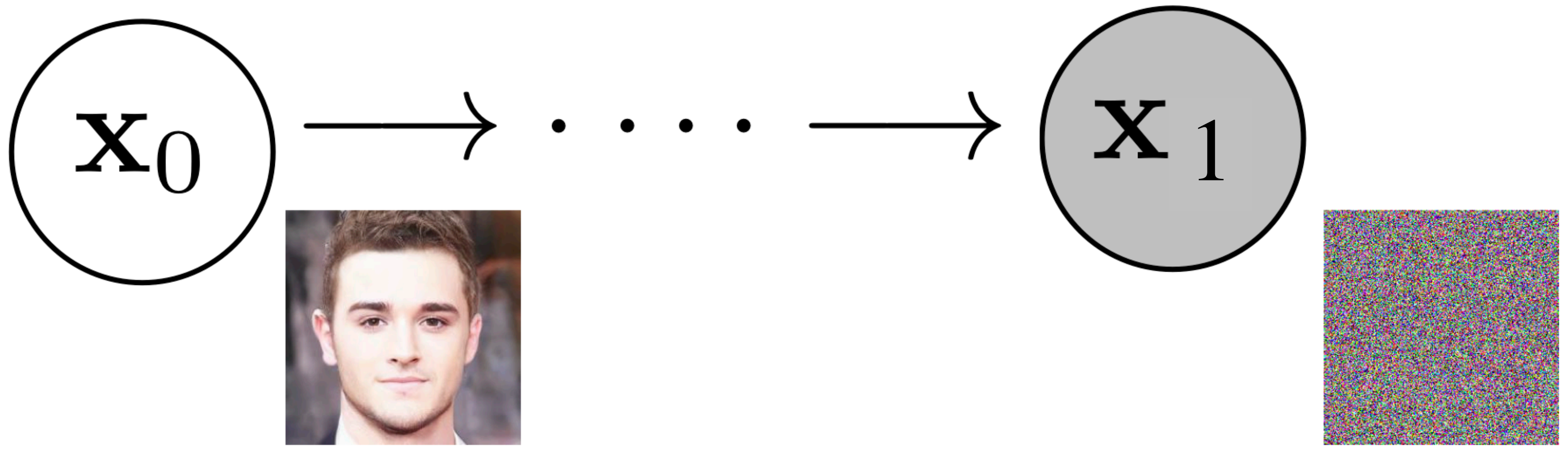
Flow Matching (Lipman et al. 2210.02747)

Phase Space

$x_0 \sim p_0(x_0)$

Latent Space

$x_1 \sim p_1(x_1)(= \mathcal{N}(0,1))$



Continuous time evolution **t**

Evolution governed by $v \equiv \dfrac{dx}{dt}$

Slide Credit: Sofia Palacios Schweitzer

Figure from J.Ho et al.: arXiv:2006.11239

Flow Matching (Lipman et al. 2210.02747)

Phase Space
$t = 0$

Individual Samples
$x_0 = x_{gen} \sim p_0(x_0)$

Density:
$p_0(x) = p(x_{gen})$

$$\frac{dx_t}{dt} = v_\theta(x_t\,,\,t)$$

$$\frac{\partial p(x,t)}{\partial t} + \nabla_x\left[p(x,t)v_\theta(x,t)\right] = 0$$

Latent Space
$t = 1$

Individual Samples
$x_1 = \epsilon \sim p_1(x_1)$

Density:
$p_1(x) = \mathcal{N}(0,1)$

Flow Matching (Lipman et al. 2210.02747)

$x_{rec} \sim p(x_{rec})$

Phase Space

$t = 0$

Individual Samples

$x_0 = x_{gen} \sim p_0(x_0)$

Density:

$p_0(x) = p(x_{gen} | x_{rec})$

$$\frac{dx_t}{dt} = v_\theta(x_t\,,\, t \,|\, x_{rec})$$

$$\frac{\partial p(x, t \,|\, x_{rec})}{\partial t} + \nabla_x \left[p(x, t \,|\, x_{rec}) v_\theta(x, t \,|\, x_{rec}) \right] = 0$$

Latent Space

$t = 1$

Individual Samples

$x_1 = \epsilon \sim p_1(x_1)$

Density:

$p_1(x) = \mathcal{N}(0,1)$

Z + jet results

$pp \rightarrow Z+$ jets events following Andreassen et al. arXiv: 1911.09107

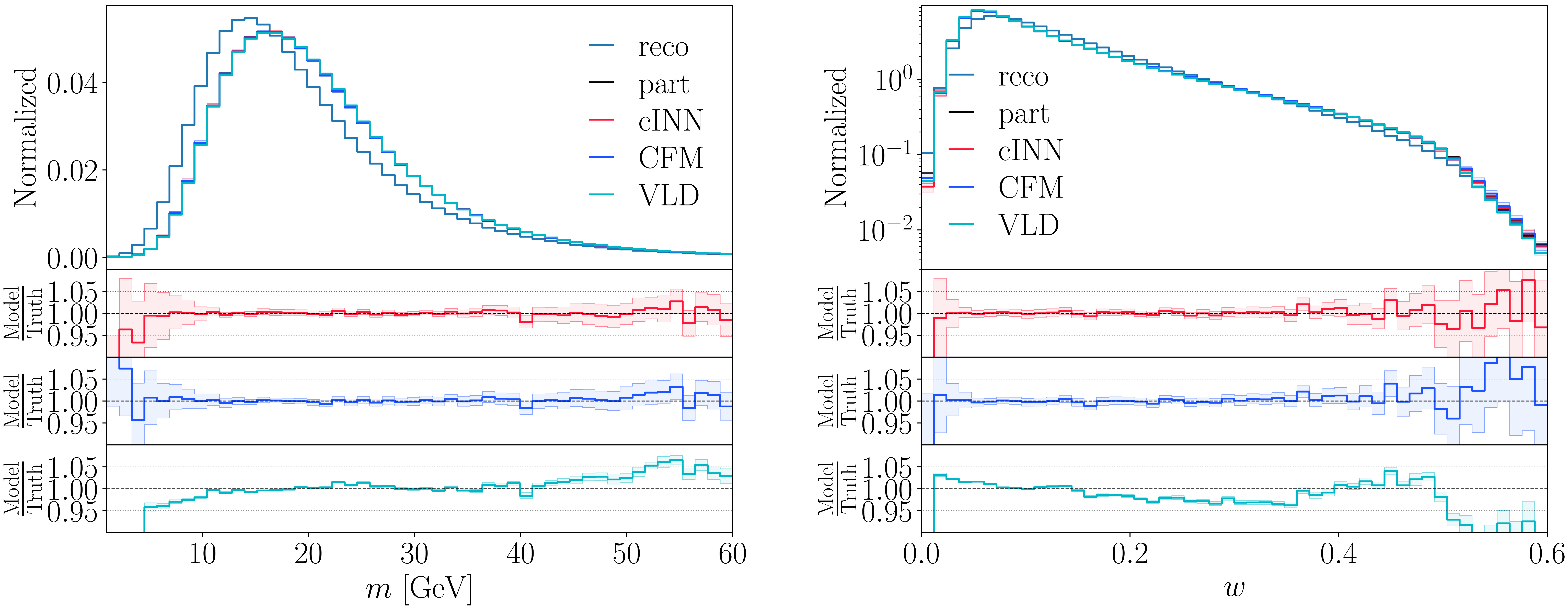
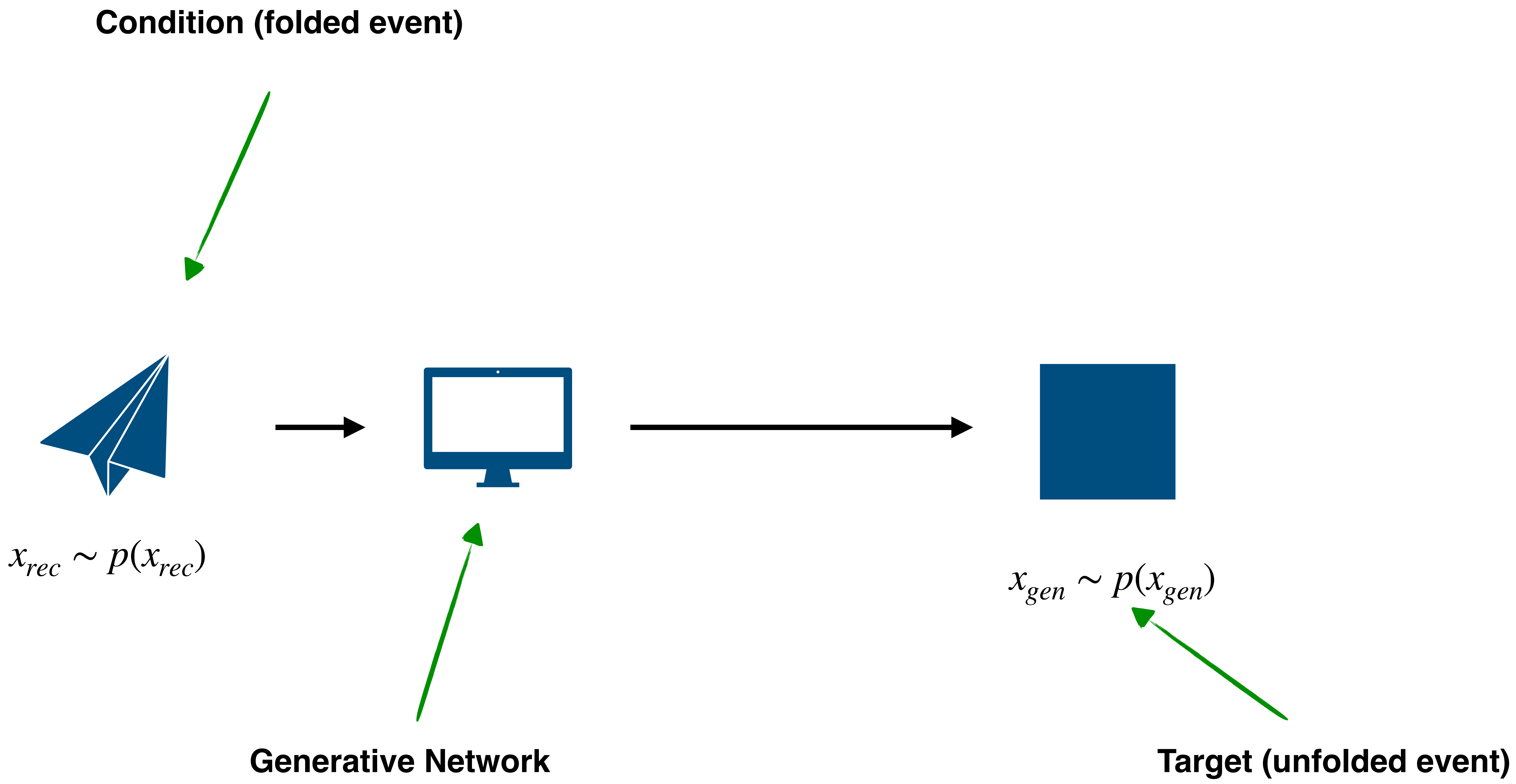
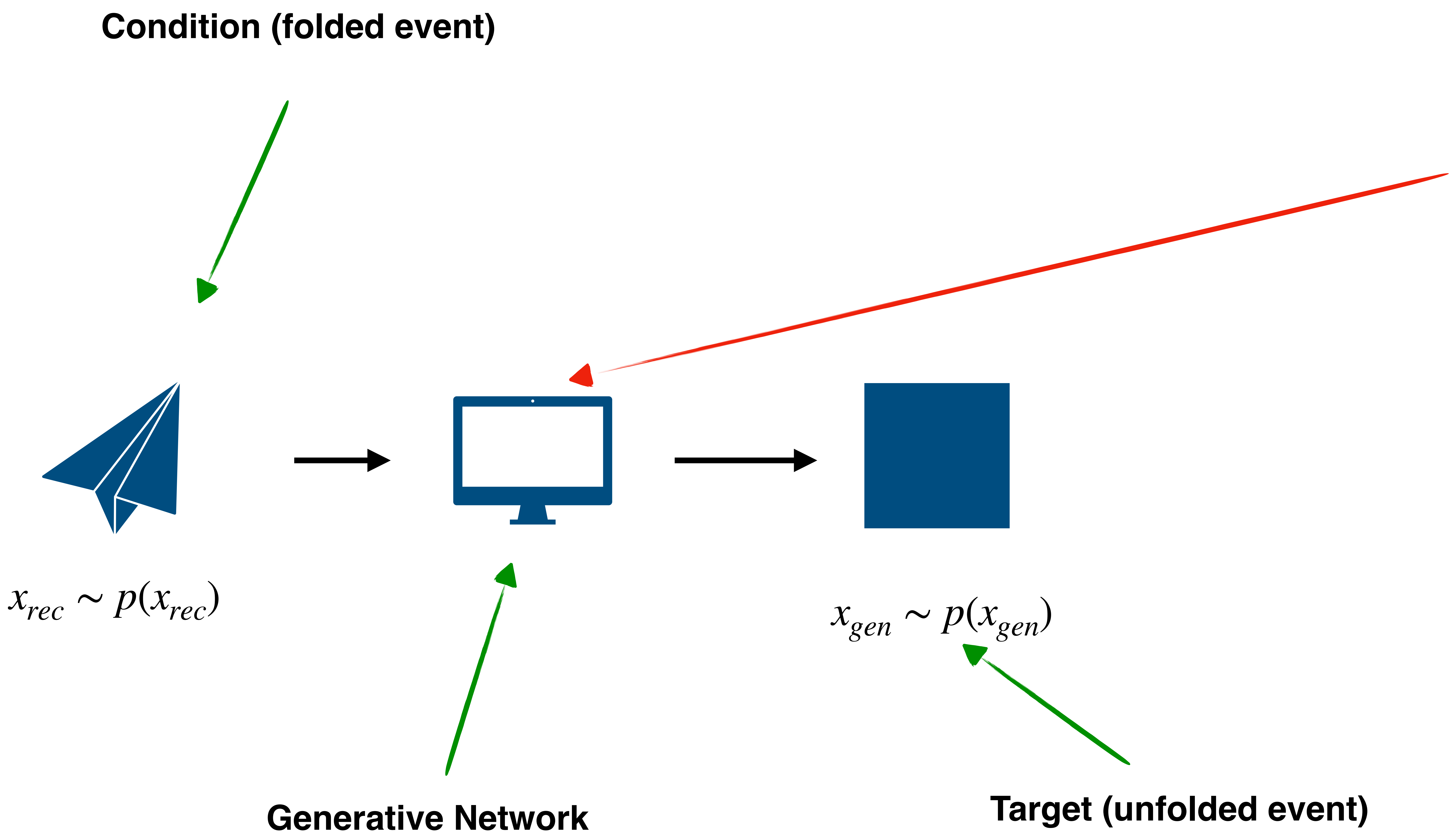


Figure from Huetsch et. al: arXiv:2404.18807

Distribution mapping



Distribution mapping



Schroedinger Bridge

Diefenbacher et al.
arXiv:2308.12351

Direct Diffusion

Butter et al.
arXiv:2311.17175
Huetsch et al.
arXiv:2404.18807

Z + jet results

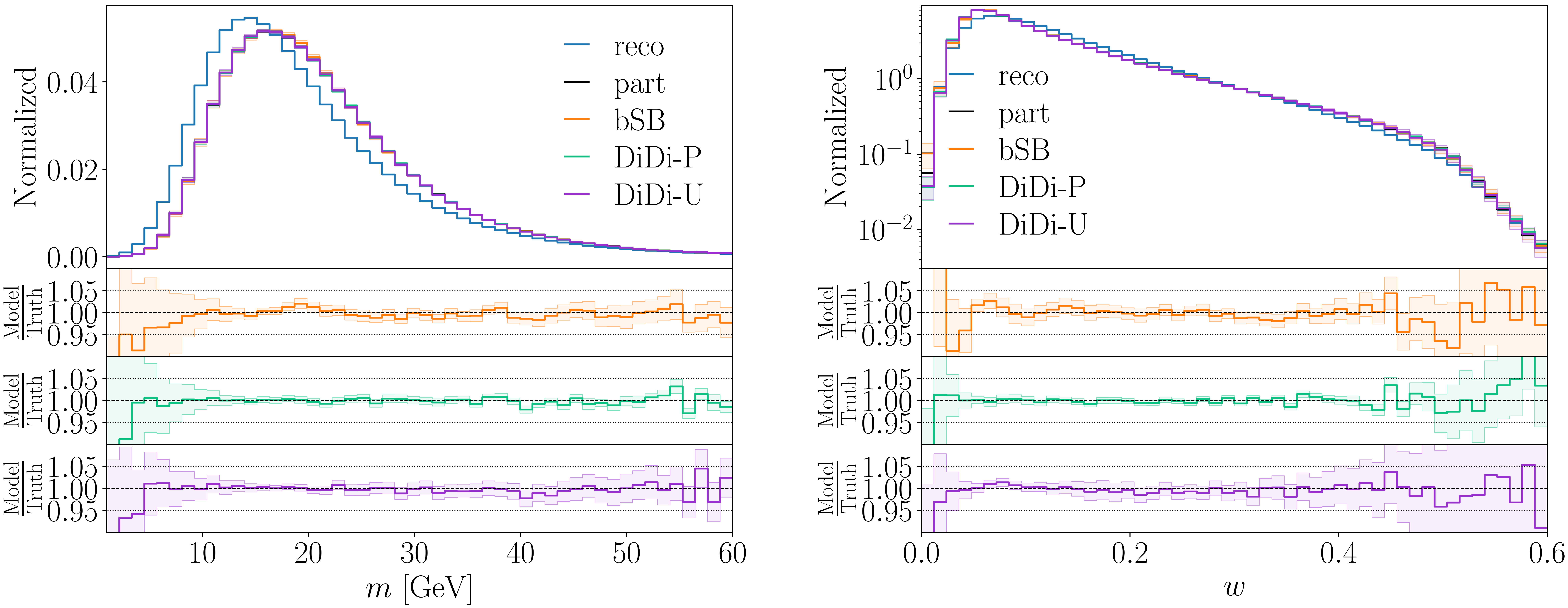


Figure from Huetsch et. al: arXiv:2404.18807

Z + jet migration

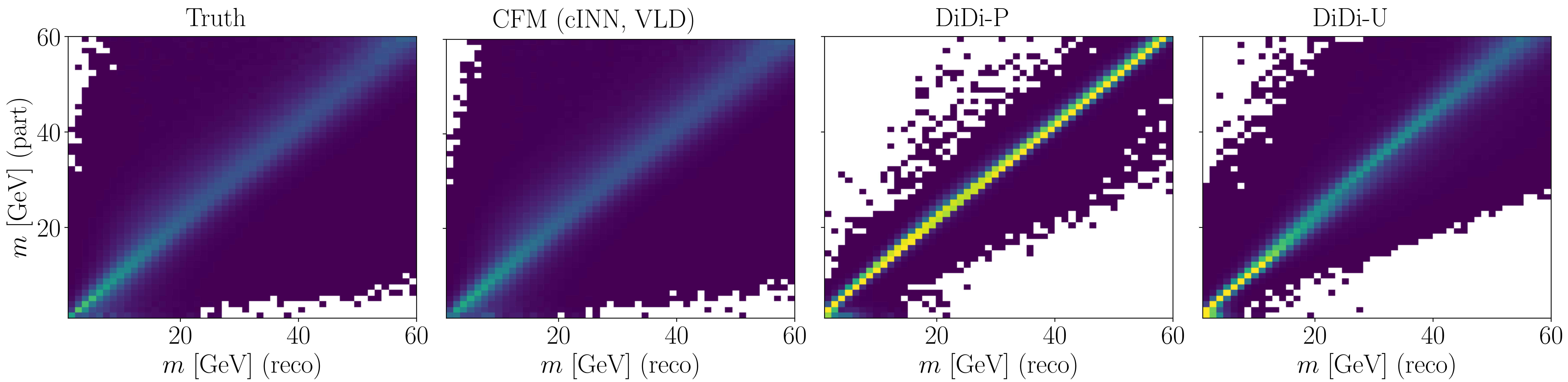


Figure from Huetsch et. al: arXiv:2404.18807

Parton-level unfolding — $t\bar{t}$ decay

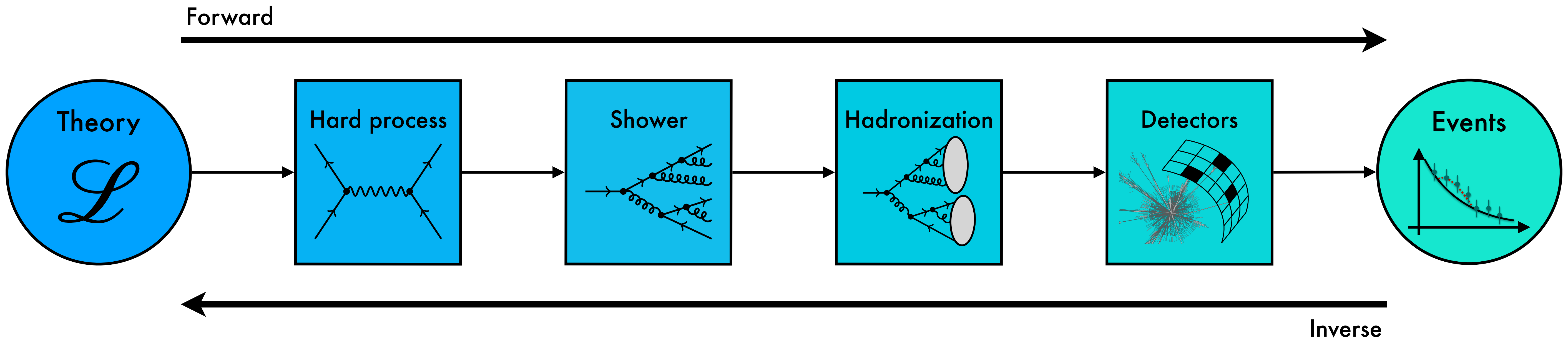
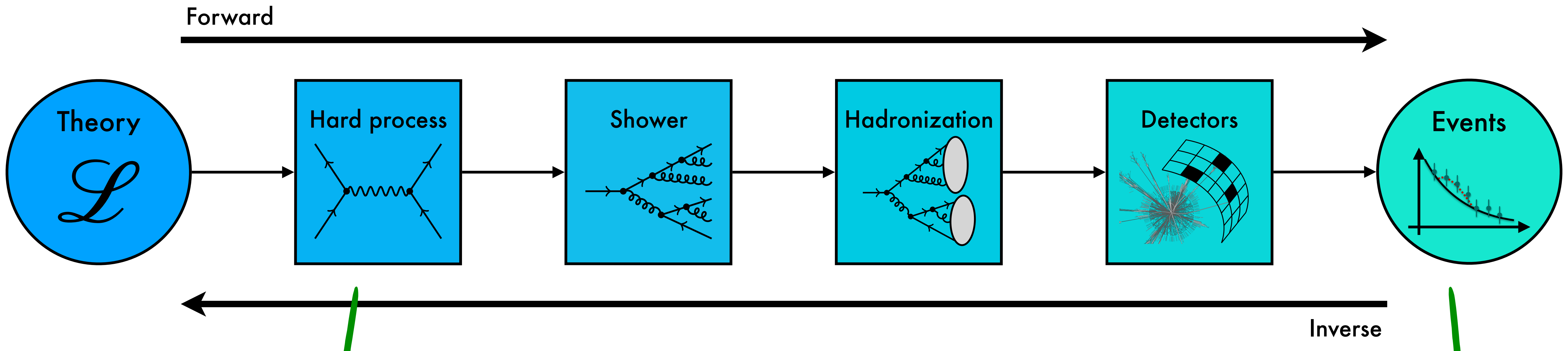
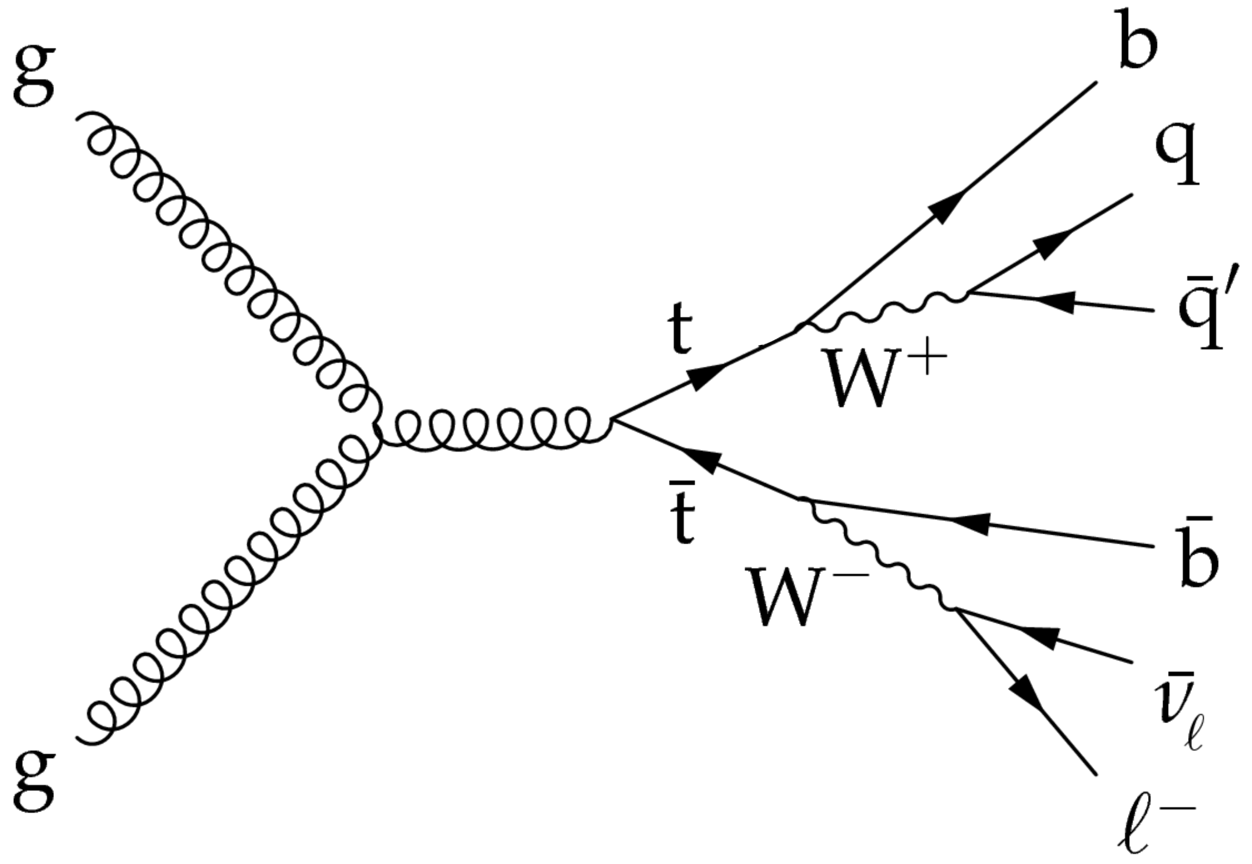


Figure top from A. Butter et al.: arXiv:2203.07460, R. Winterhalder

Parton-level unfolding — $t\bar{t}$ decay



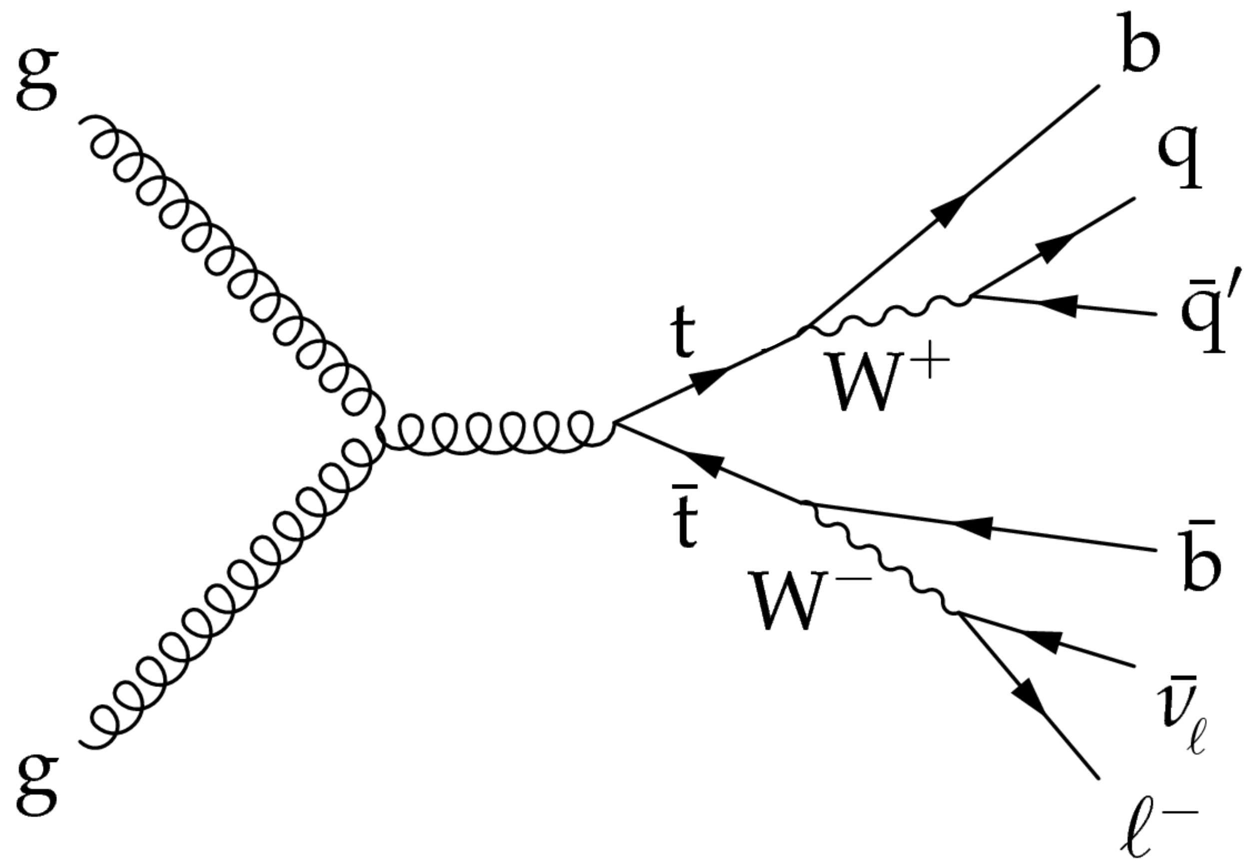
$$q\bar{q}/gg \to t\bar{t} \to (b\ell^-\bar{\nu}_\ell)\,(\bar{b}qq) \longrightarrow \ell^-\bar{\nu}_\ell\{j\}$$



Following
Shmakov et al.
arXiv: 2305.10399

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

Parton-level unfolding — $t\bar{t}$ decay



→ $\ell^+\nu_\ell\{j\}$

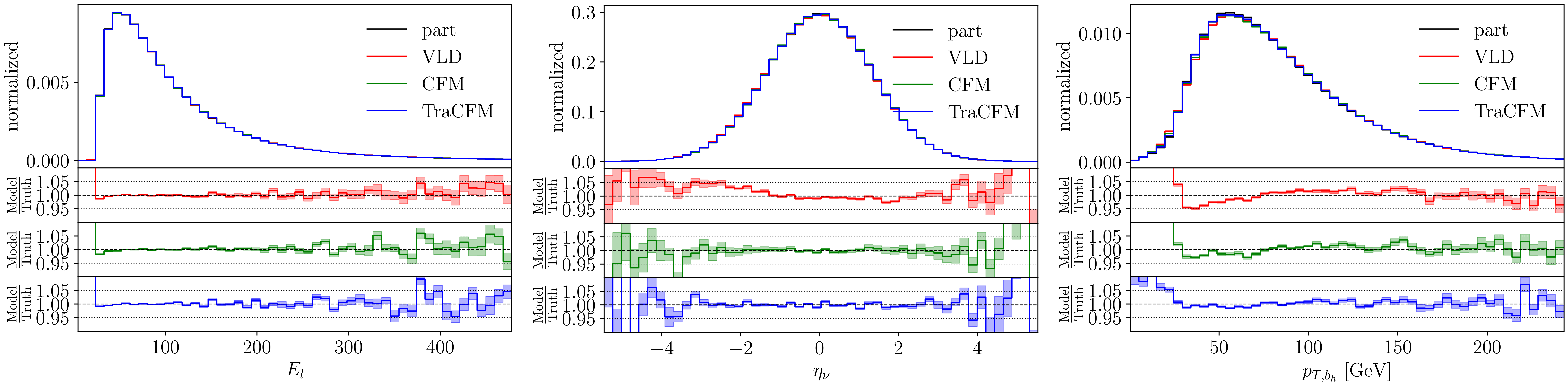
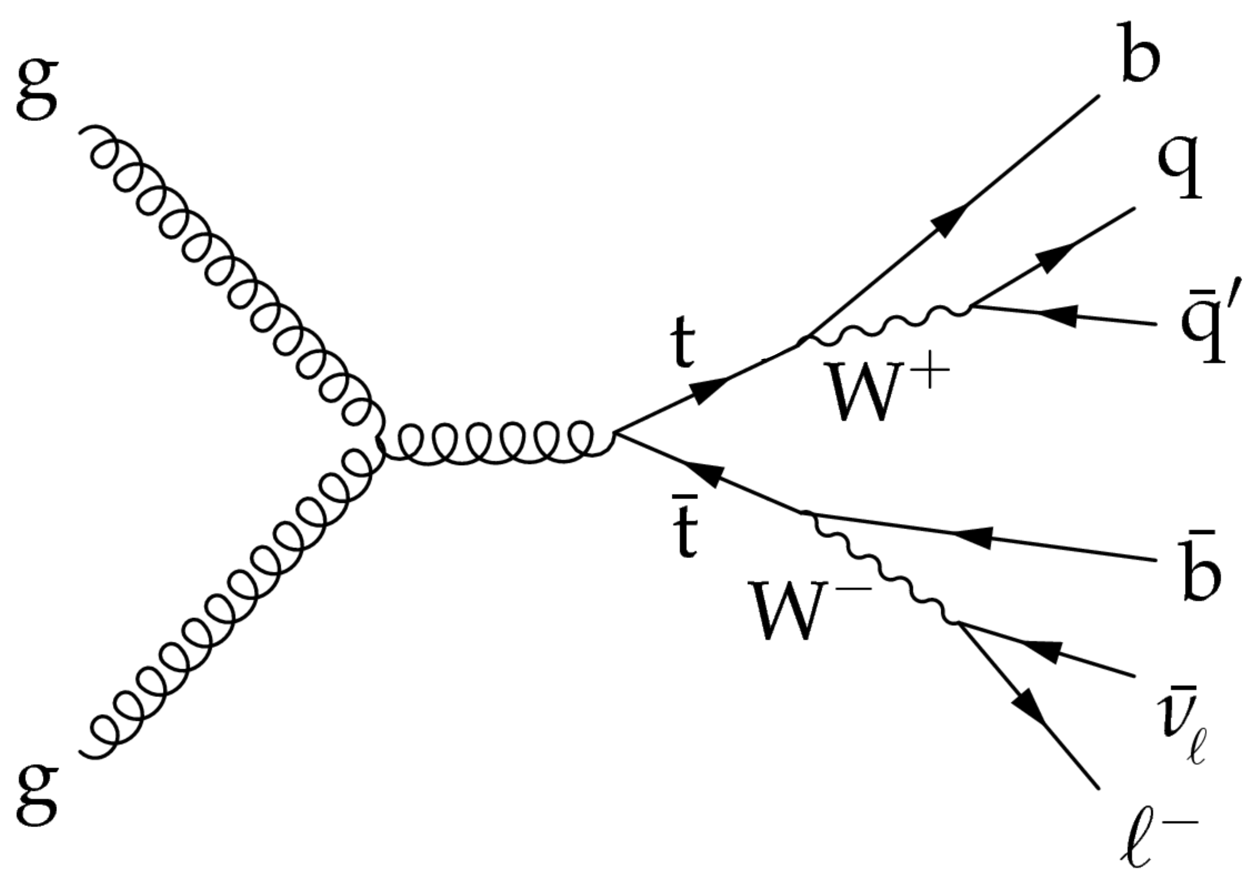


Figure from Huetsch et. al: arXiv:2404.18807

Parton-level unfolding — $t\bar{t}$ decay



$\ell^+\nu_\ell\{j\}$

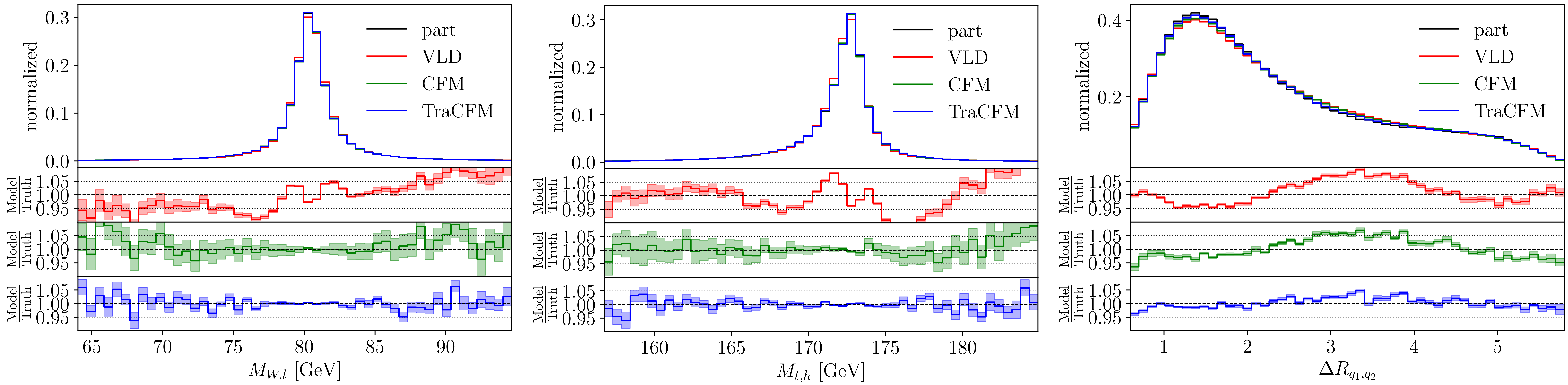
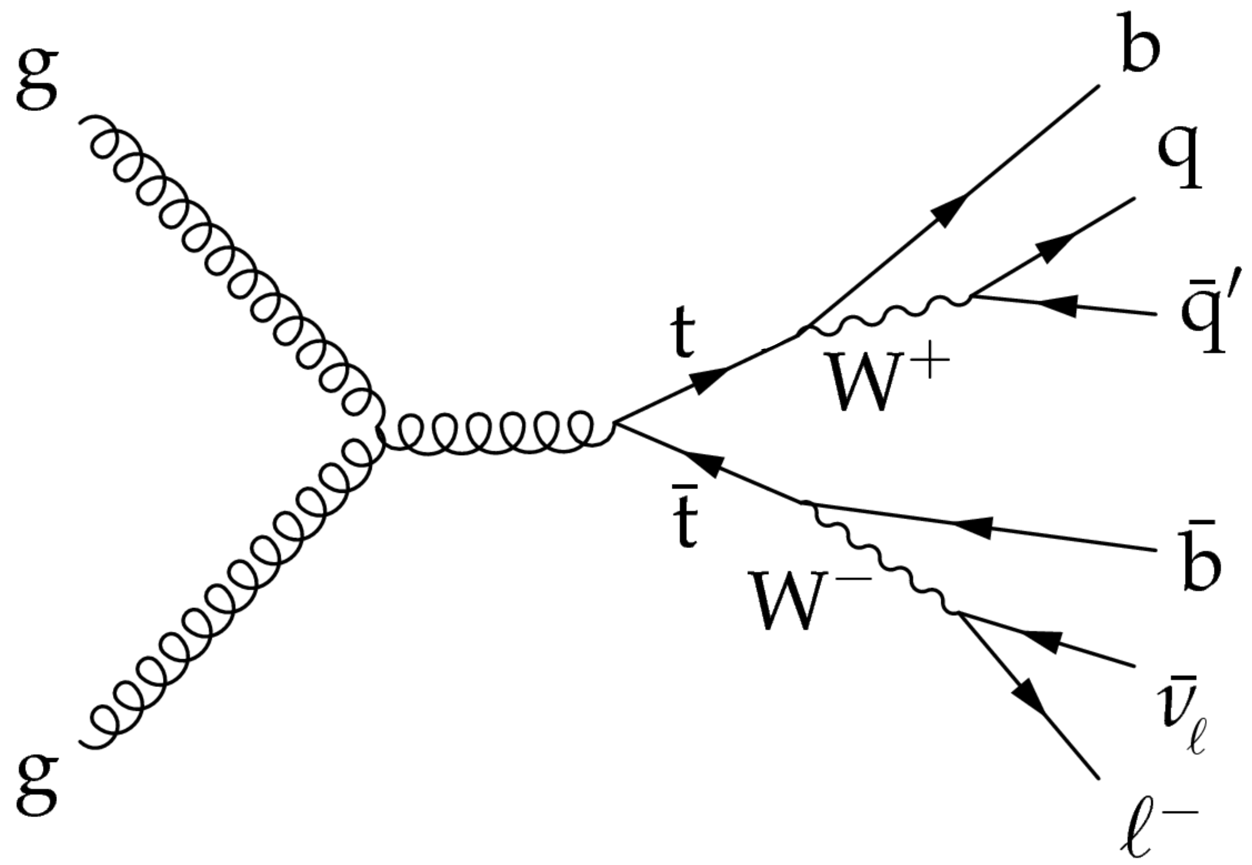


Figure from Huetsch et. al: arXiv:2404.18807

Parton-level unfolding — $t\bar{t}$ decay



$\longrightarrow \quad \ell^+ \nu_\ell \{j\}$

Train a classifier classifier between $p_{gen}(x)$ and $p_{unfold}(x)$

It learns the likelihood ratio

$$w(x) = \frac{p_{gen}(x)}{p_{unfold}(x)}$$

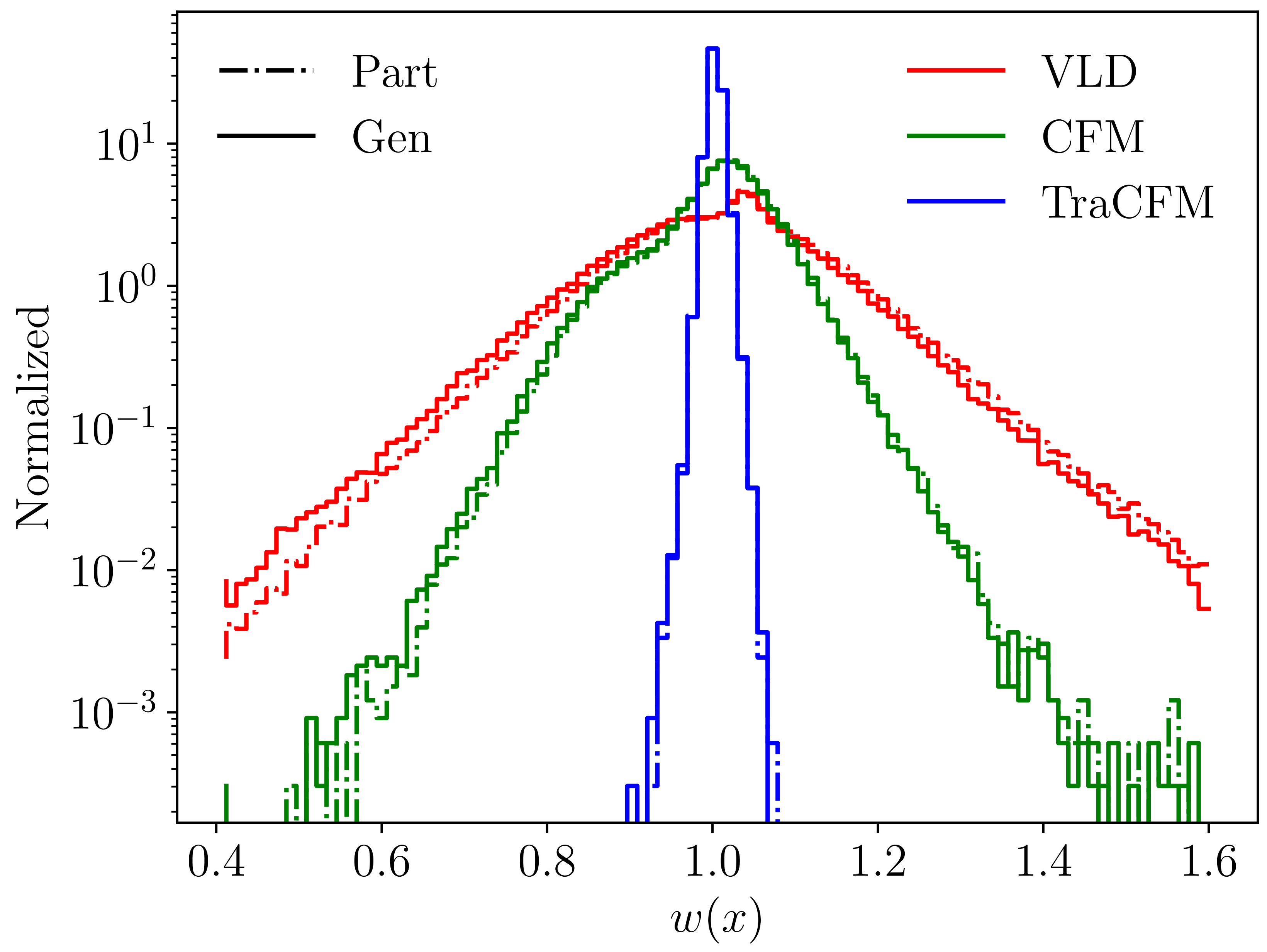


Figure from Huetsch et. al: arXiv:2404.18807

What about model dependence?

Prior

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

What about model dependence?

This problem is common to a long list of unfolding methods, with and without ML

Solution: Follow an iterative approach where we update our prior after each iteration

The same is done in Iterative Bayesian Unfolding, RooUnfold

Prior

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

$$p_{unfold}(x_{gen}) = \int p_{data}(x_{rec})p(x_{gen}|x_{rec})\, dx_{rec}$$

Use as new prior and start over

Iterative generative unfolding

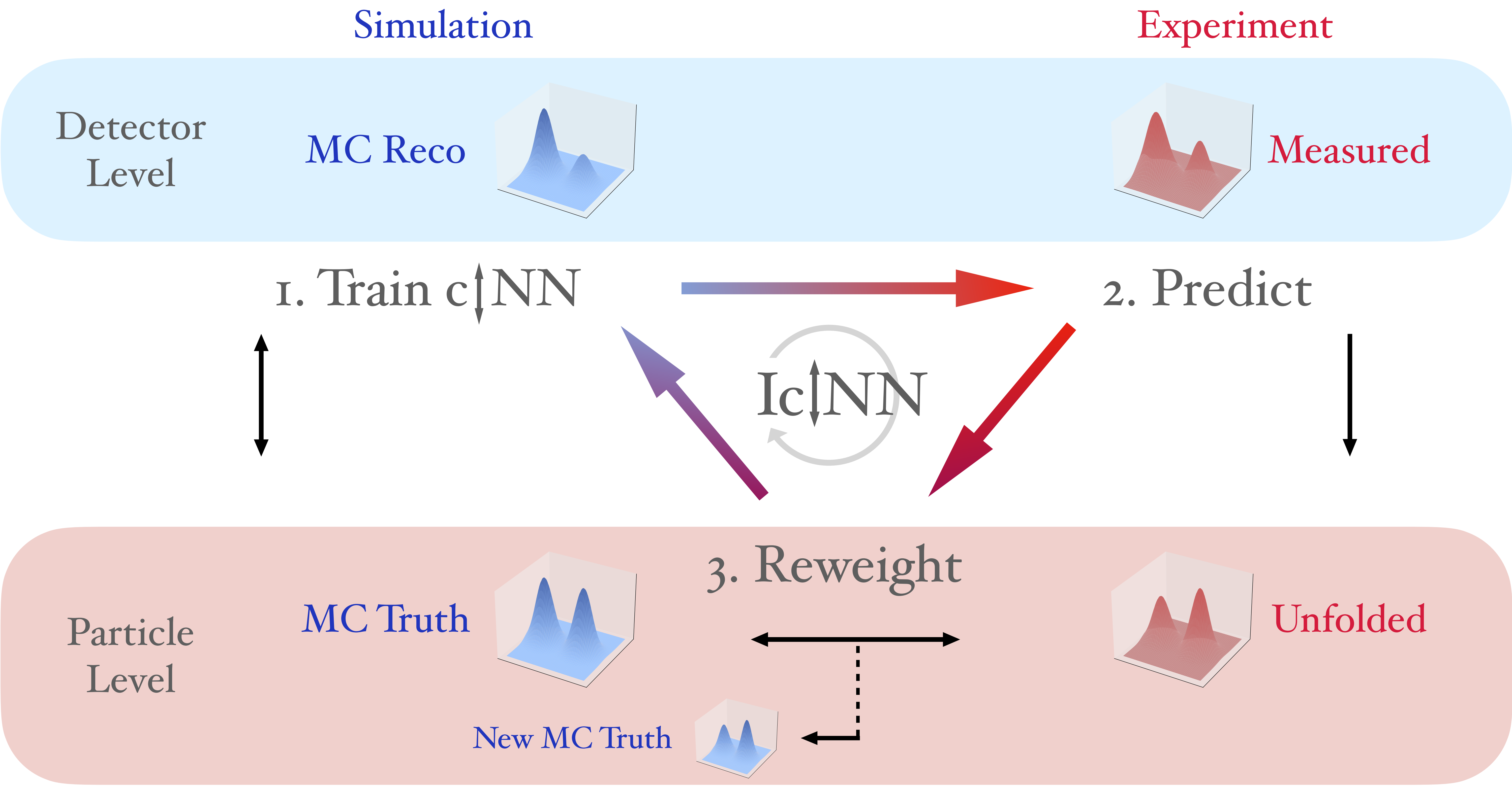
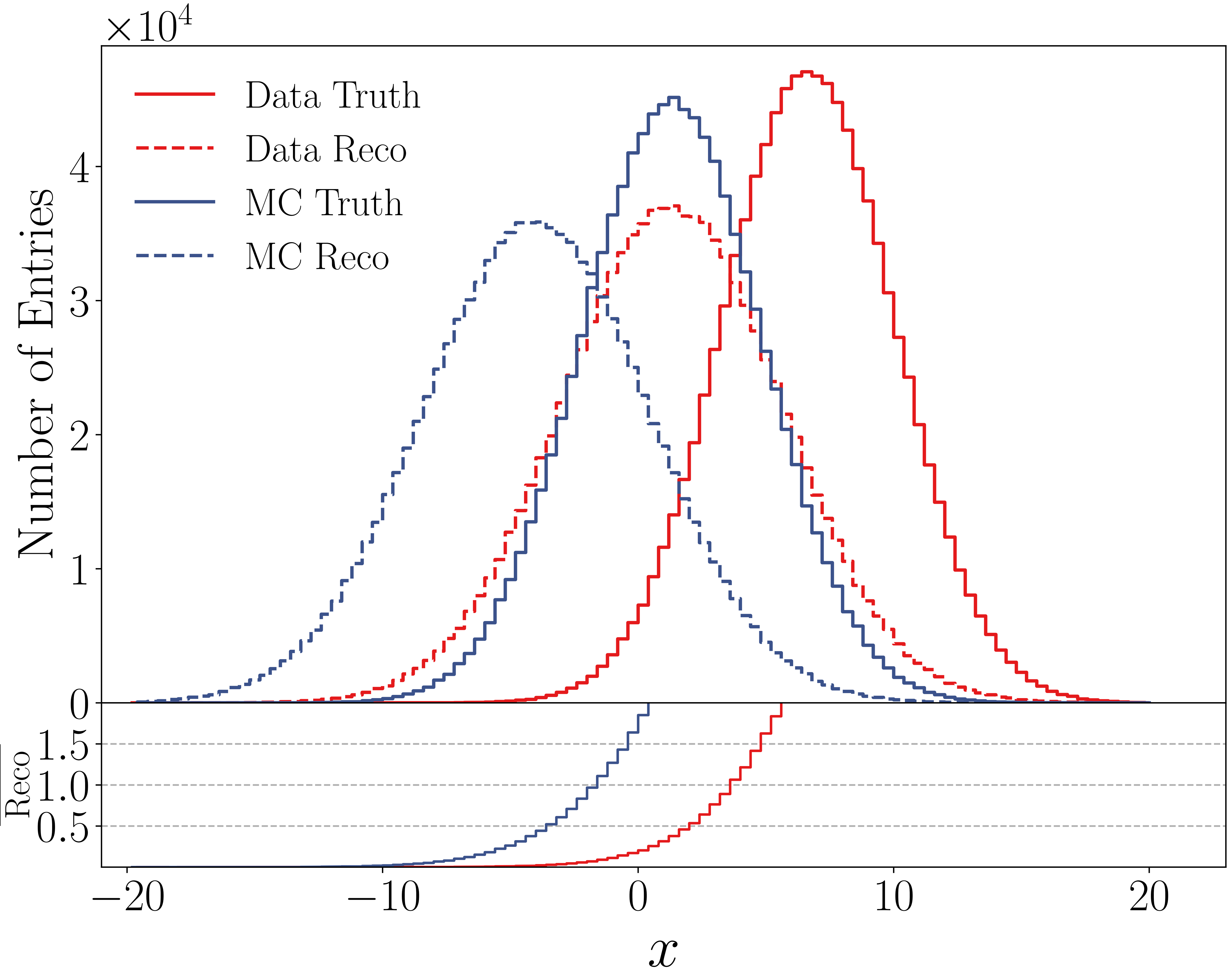
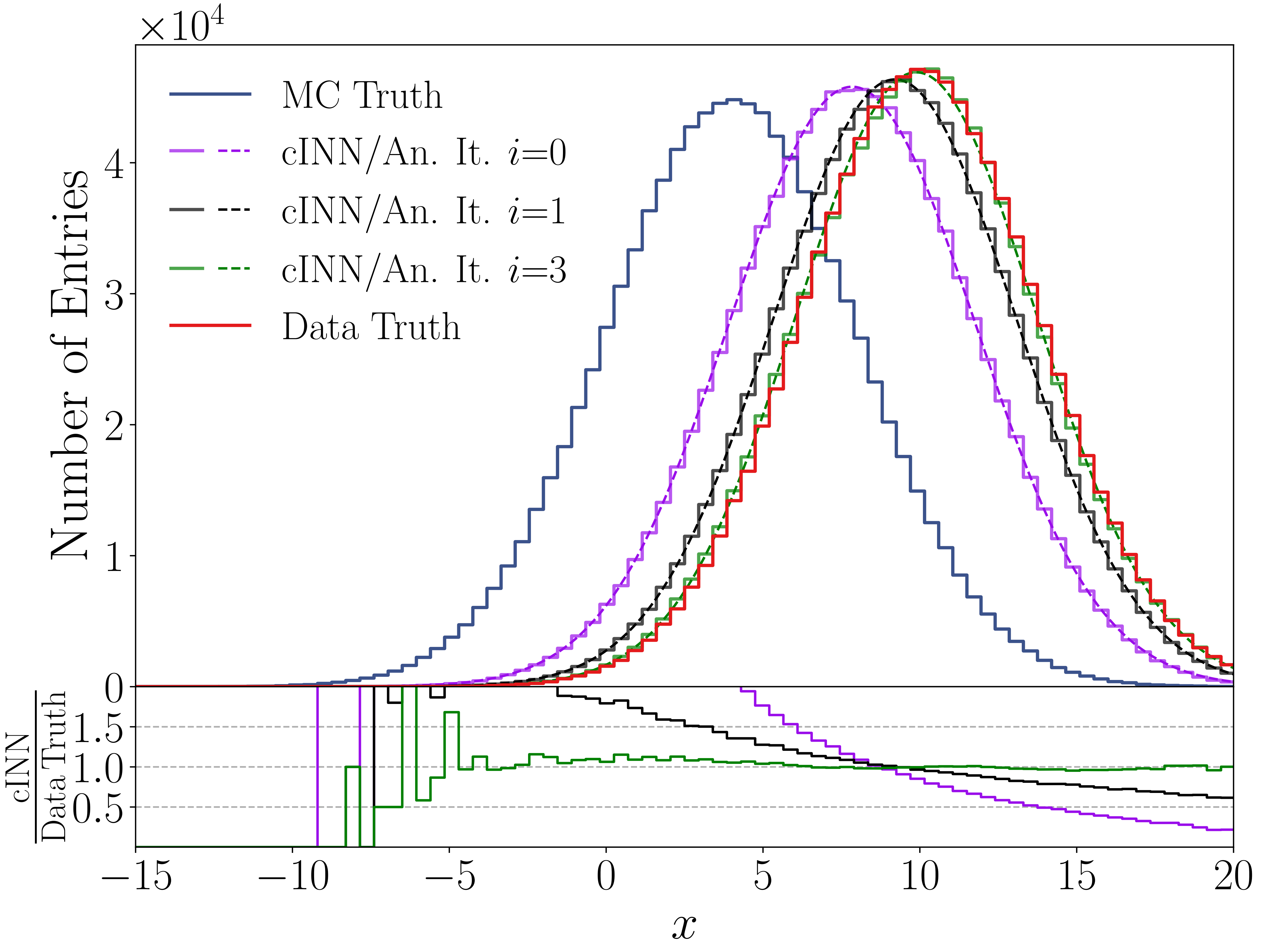


Figure from Backes et al.: arXiv:2212.08674

Iterative generative unfolding



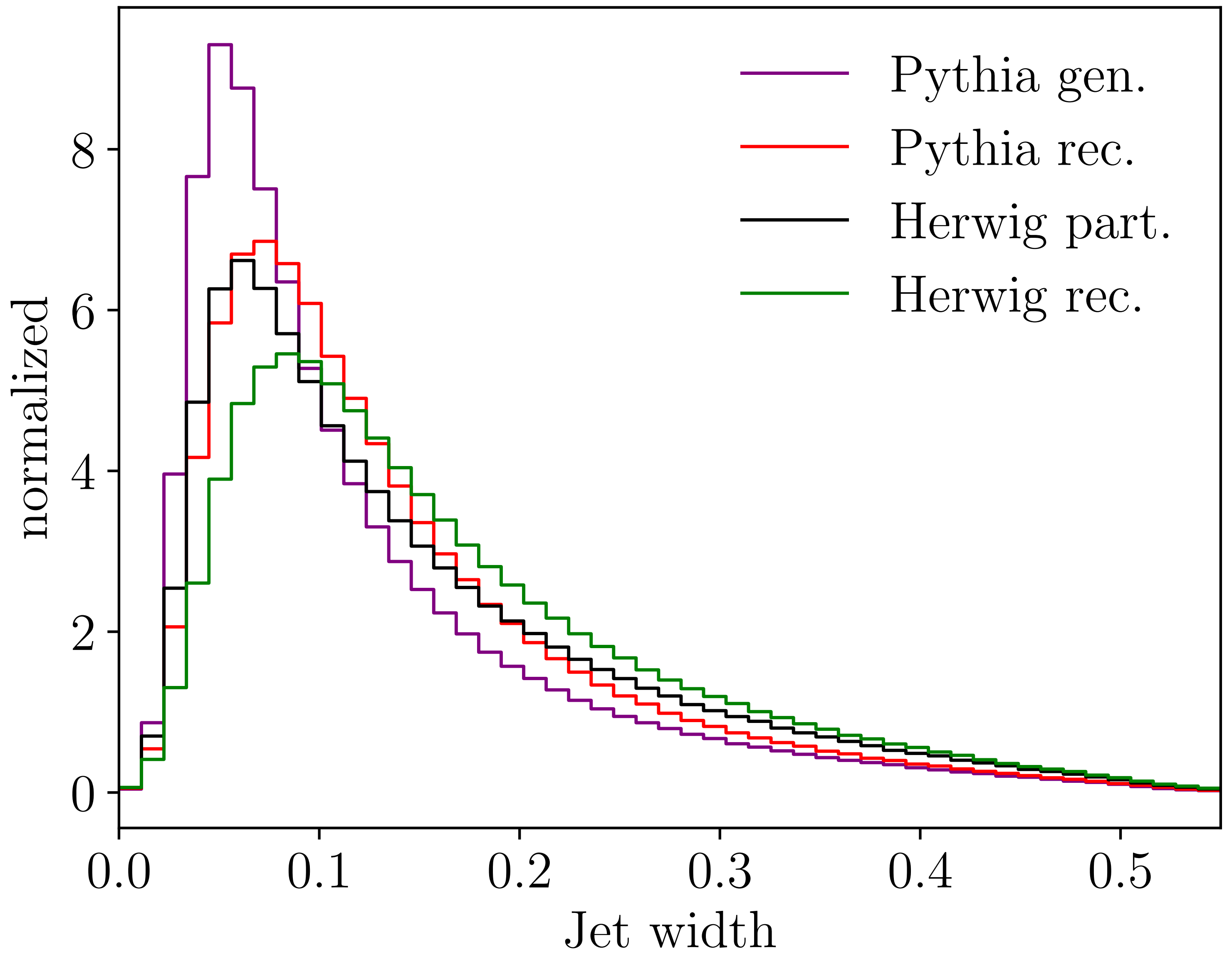


Figures from Backes et al.: arXiv:2212.08674

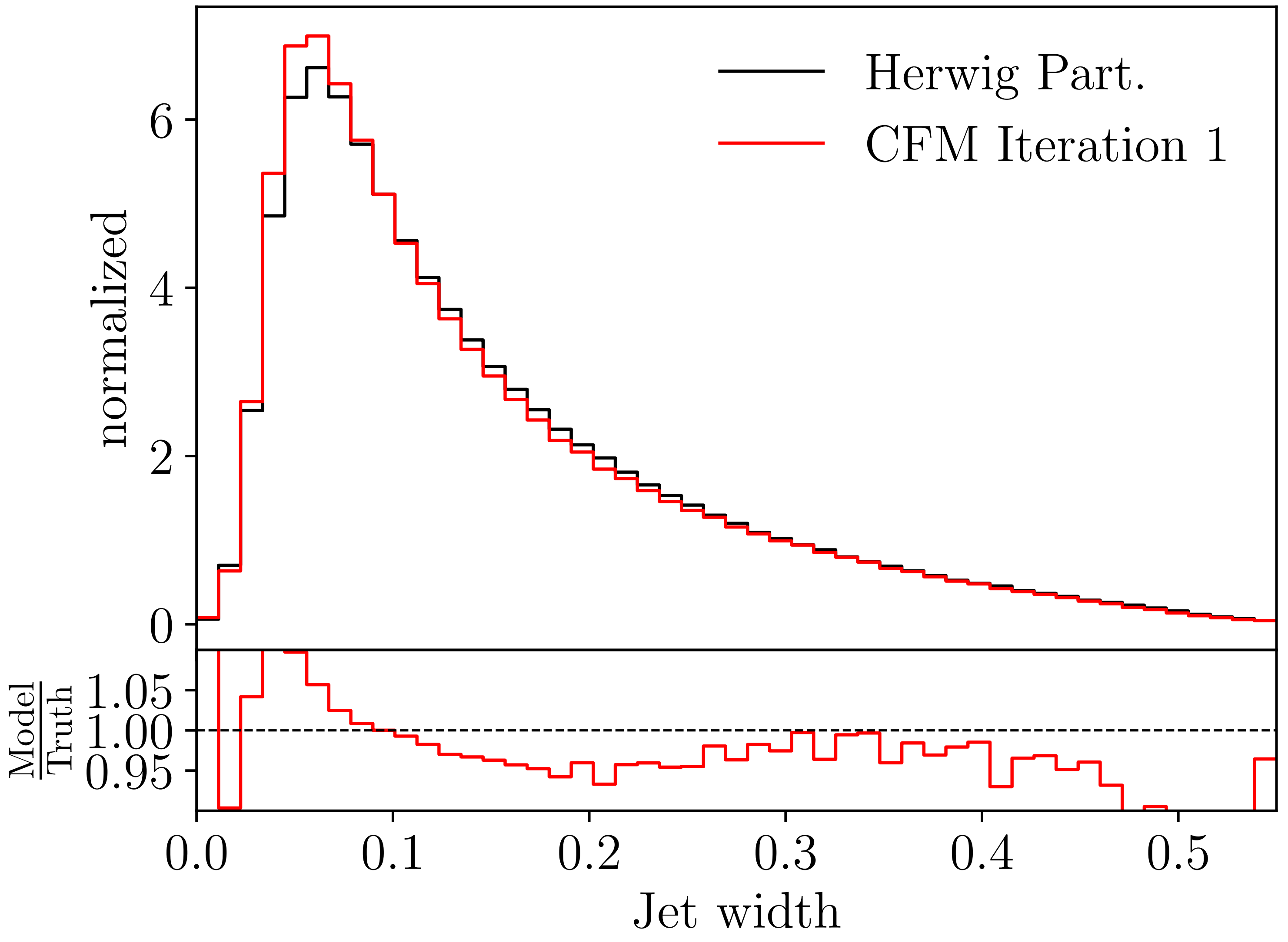
Z+jets: Pythia vs Herwig simulation

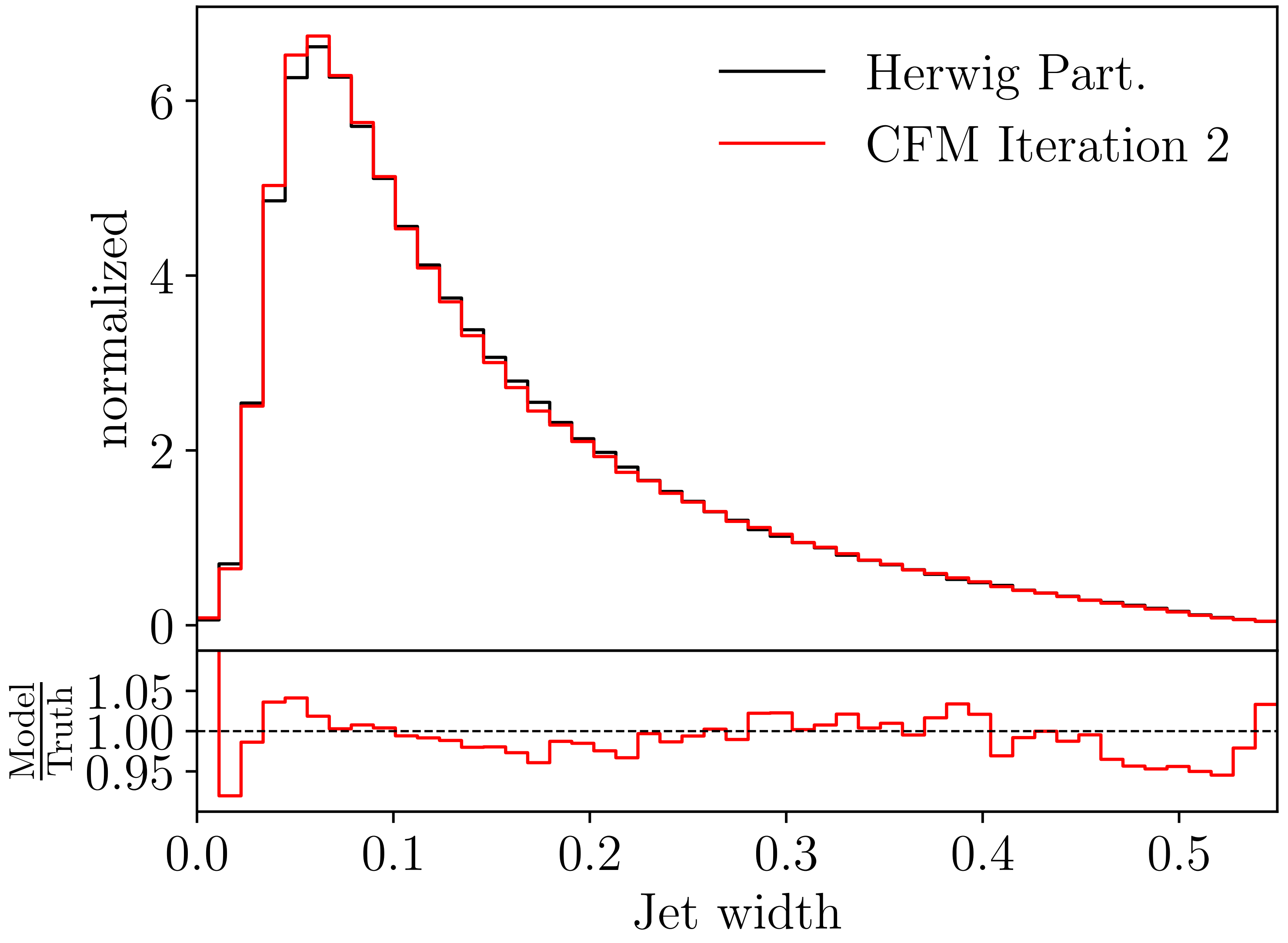
Use Pythia simulation as MC

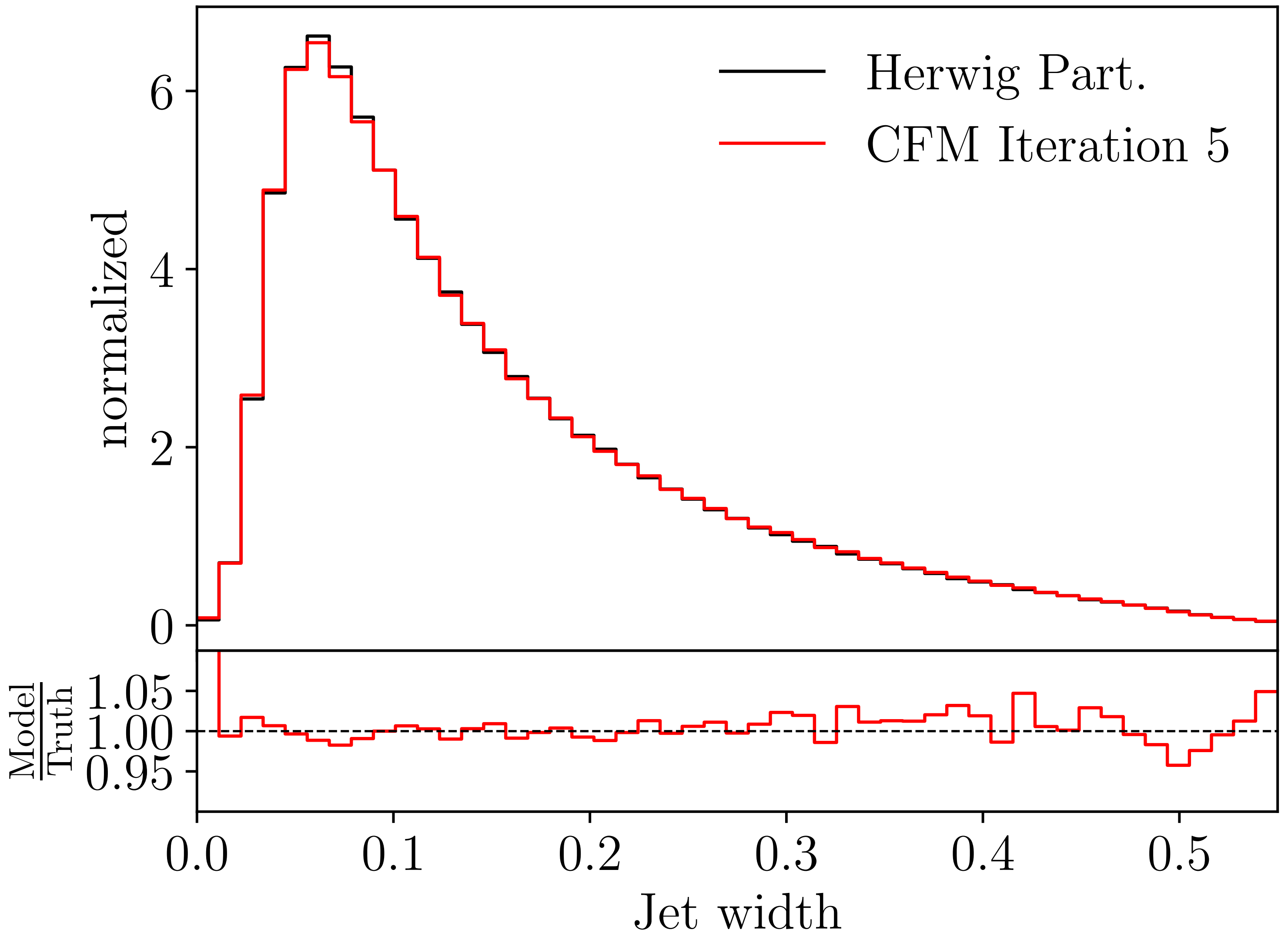
Use Herwig simulation as Data



Following
Andreassen et al.
arXiv: 1911.09107



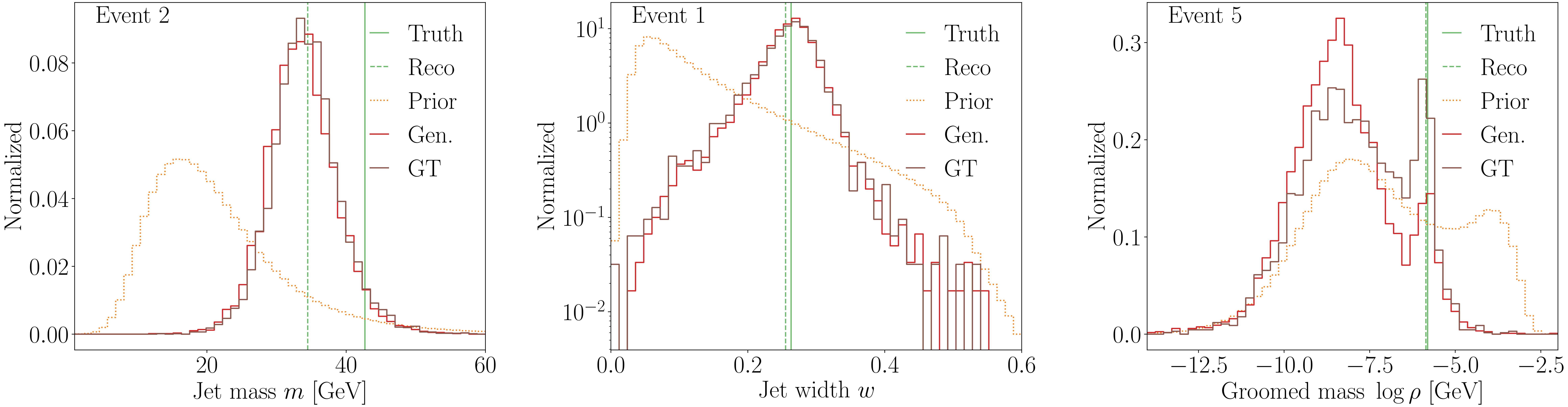




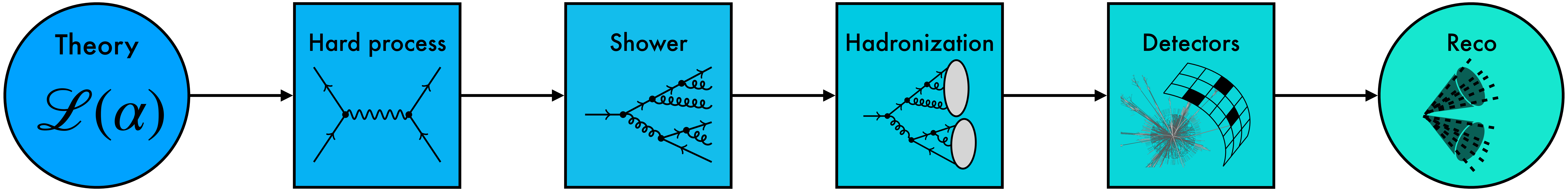
Single-event posteriors

Our network learned the posterior distribution $p(x_{gen} \mid x_{rec})$

We can sample from this for a single reco event to map out the posterior



Making use of single-event posteriors



$$\int dx_{hard} \quad p(x_{hard}\,|\,\alpha) \quad \underbrace{p(x_{reco}\,|\,x_{hard})} \quad = \; p(x_{reco}\,|\,\alpha)$$

We want to explicitly calculate reco-level likelihoods for our observed data given some theory hypothesis $\mathscr{L}(\alpha)$

Let us say we somehow now the transfer probability $p(x_{reco}\,|\,x_{hard})$ and can efficiently calculate it

How can we solve the difficult high-dimensional integral precisely and efficiently?

Figure adapted from A. Butter et al.: arXiv:2203.07460

Making use of single-event posteriors

Challenging

$$\int dx_{hard}\ p(x_{hard}|\alpha)\ p(x_{reco}|x_{hard}) \ = \ p(x_{reco}|\alpha)$$

$$= \left\langle \frac{1}{q(x_{hard}} p(x_{hard}|\alpha)\ p(x_{reco}|x_{hard})\ ) \right\rangle_{x_{hard} \sim q}$$

Integral becomes trivial if : $q(x_{hard}|x_{reco}, \alpha) = p(x_{hard}|x_{reco}, \alpha)$

This is exactly our unfolding network !

For details see Butter at al. 2210.00019, Heimel et al. 2310.07752

Conclusion

Generative Unfolding works !

Conditional Networks and Distribution Mapping both achieve percent-level precision

Conditional Networks learn the true migration encoded in the detector simulation. Distribution Mapping Networks learn a simple optimal-transport based mapping

Conditional Generative Unfolding enables probabilistic inversion of simulation chain

Prior dependence is a solvable problem

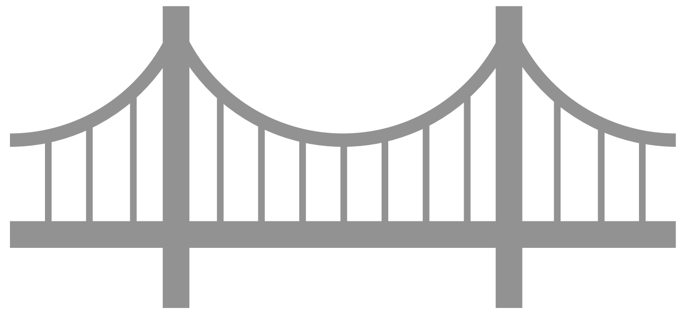
Conditional Generative Models map out the single-event posterior distributions

2404.18807

SciPost Physics Submission

The Landscape of Unfolding with Machine Learning

Nathan Huetsch[1], Javier Mariño Villadamigo[1], Alexander Shmakov[2], Sascha Diefenbacher[3], Vinicius Mikuni[3], Theo Heimel[1], Michael Fenton[2], Kevin Greif[2], Benjamin Nachman[3,4], Daniel Whiteson[2], Anja Butter[1,5], and Tilman Plehn[1,6]



Further investigation of failure modes

Further investigation of uncertainties

Further investigation of background, cuts, efficiencies

Flow Matching (Lipman et al. 2210.02747)

Training

1. **Sample paired data from our simulation**

$$(x_0, c) = (x_{gen}, x_{rec}) \sim p(x_{gen}, x_{rec})$$

2. **Sample noise and a timestep**

$$x_1 = \epsilon \sim \mathcal{N}(0,1)\ ,\ t \sim \mathcal{U}([0,1])$$

3. **Calculate the trajectory**

$$x_t = (1-t)x_0 + tx_1$$

$$v_t = \frac{dx_t}{dt} = -x_0 + x_1$$

4. **Predict the velocity field**

$$\mathcal{L} = \left| v_\theta(x_t, t, c) - v_t \right|^2$$

Generation

1. **Sample a reco event from our measured data**

$$c = x_{rec} \sim p(x_{rec})$$

2. **Sample noise as initial condition**

$$x_1 = \epsilon \sim \mathcal{N}(0,1)$$

3. **Solve the ODE numerically**

$$x_0 = x_{gen} = x_1 + \int_1^0 v_\theta(x_t, t, c)\ dt$$