UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

IRN Terascale
20.05.2025

Generative Unfolding with Distribution Mapping

Nathan Huetsch

SPONSORED BY THE

Federal Ministry
of Education
and Research

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

Butter et al. 2411.02495
Generative Unfolding with Distribution Mapping

Simulation chain — Forward

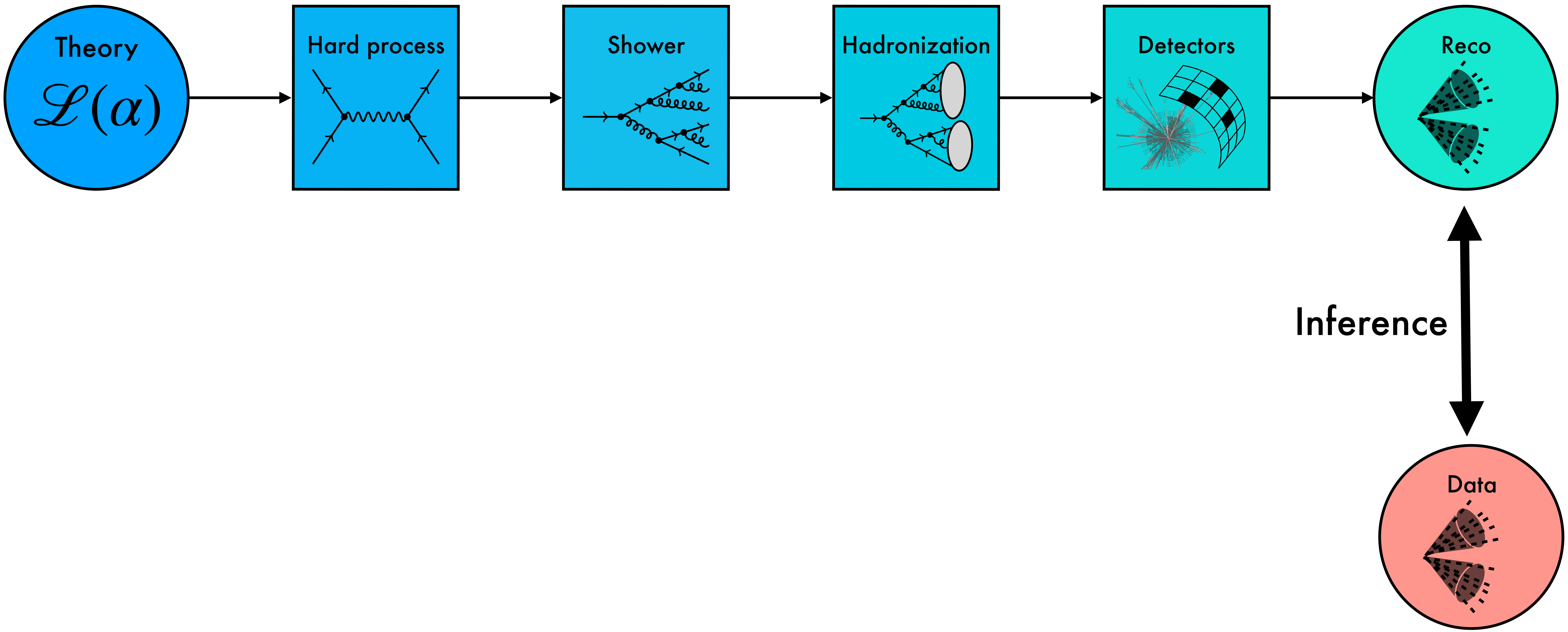


Figure adapted from R. Winterhalder

Simulation chain — Inversion

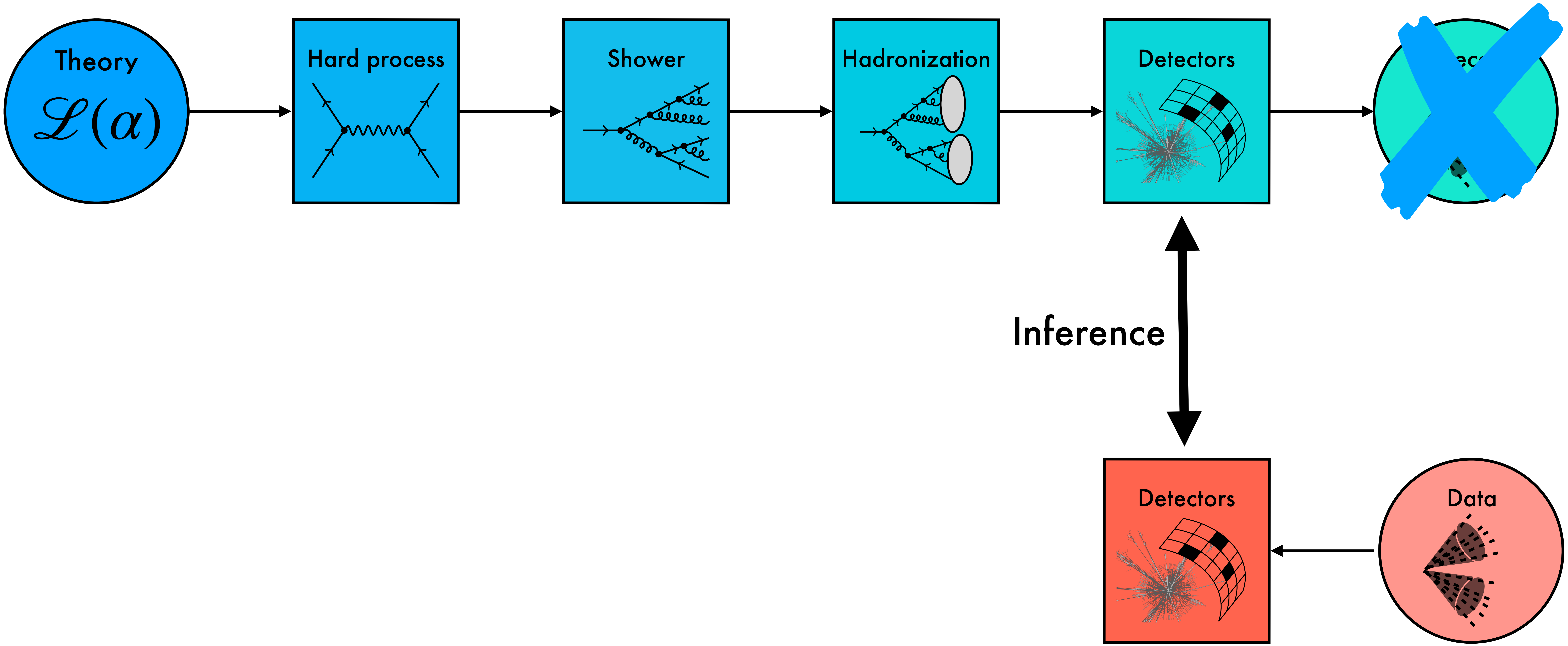


Figure adapted from R. Winterhalder

Unfolding

$$
\begin{tikzcd}
p_{\text{gen}}(x_{\text{part}}) \arrow[r, leftrightarrow] \arrow[d, "\text{simulation}"'] & p_{\text{unfold}}(x_{\text{part}}) \\
p_{\text{sim}}(x_{\text{reco}}) \arrow[r, leftrightarrow] & p_{\text{data}}(x_{\text{reco}}) \arrow[u, "\text{unfolding}"']
\end{tikzcd}
$$

Unfolding

Omnifold

Andreassen et al.
arXiv:1911.09107
arXiv:2105.04448

$$\begin{tikzcd}
p_{\mathrm{gen}}(x_{\mathrm{part}}) \arrow[r, leftrightarrow, "\text{Apply reweighting}"] & p_{\mathrm{unfold}}(x_{\mathrm{part}}) \\
p_{\mathrm{sim}}(x_{\mathrm{reco}}) \arrow[u, "\text{pull weights}"] \arrow[r, leftrightarrow, "\text{Train reweighting}"] & p_{\mathrm{data}}(x_{\mathrm{reco}})
\end{tikzcd}$$

Generative Unfolding

Bellagente et al.
arXiv:1912.00477
arXiv:2006.06685

$$\begin{tikzcd}
p_{\mathrm{gen}}(x_{\mathrm{part}}) & p_{\mathrm{unfold}}(x_{\mathrm{part}}) \\
p_{\mathrm{sim}}(x_{\mathrm{reco}}) \arrow[u, "\text{Train } p_\theta(x_{\mathrm{part}}|x_{\mathrm{reco}})"] \arrow[r, leftrightarrow] & p_{\mathrm{data}}(x_{\mathrm{reco}}) \arrow[u, "\text{Apply } p_\theta(x_{\mathrm{part}}|x_{\mathrm{reco}})"']
\end{tikzcd}$$

Unfolding

ML4Jets 2024:
Dedicated ML-Unfolding session!

Omnifold

Andreassen et al.
arXiv:1911.09107
arXiv:2105.04448

Generative Unfolding

Bellagente et al.
arXiv:1912.00477
arXiv:2006.06685

14:00	**OmniFoldHI: Advanced ML Unfolding for Heavy-Ion Data** *Alexandre Falcão*
	Bridging the Generative Unfolding Gap *Sascha Diefenbacher*
	Measurement of Jet Track Functions with OmniFold-base... *Jingjing Pan*
15:00	**Full Event Particle-Level Unfolding with Variable-Length L...** *Kevin Thomas Greif*
	How to Unfold Top Decays *Sofia Palacios Schweitzer* *LPNHE, Paris, France* 15:10 - 15:30
	Coffee break *LPNHE, Paris, France*
16:00	**Advanced techniques for SBI in collider physics** *Giovanni De Crescenzo*
	Towards Universal Unfolding using Denoising Diffusion *Martin Klassen*

$p_{\text{gen}}(x_{\text{part}}) \xleftrightarrow{\text{Apply reweighting}} p$

pull weights $\uparrow$

$p_{\text{sim}}(x_{\text{reco}}) \xleftrightarrow{\text{Train reweighting}} p$

$p_{\text{unfold}}(x_{\text{part}})$

$\uparrow$ Apply $p_\theta(x_{\text{part}}|x_{\text{reco}})$

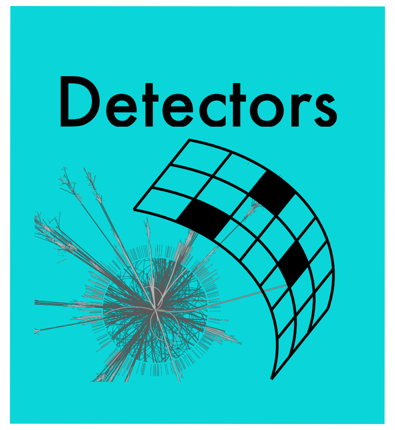
$(x_{\text{reco}}) \longleftrightarrow p_{\text{data}}(x_{\text{reco}})$

Probabilistic transfer

Transfer probability

Encoded implicitly in detector simulation

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



x_{gen} → x_{rec}

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

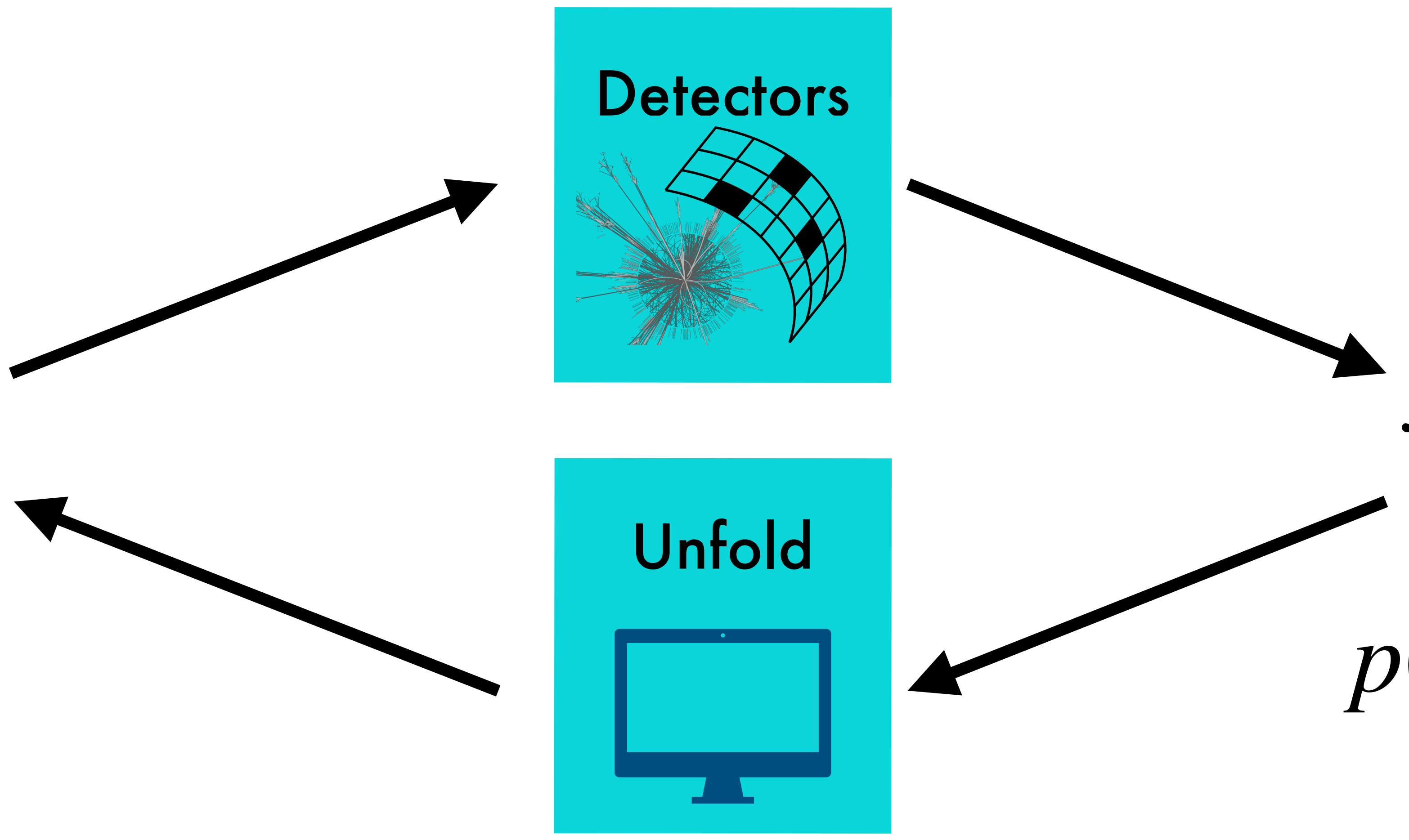
Generative unfolding

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}



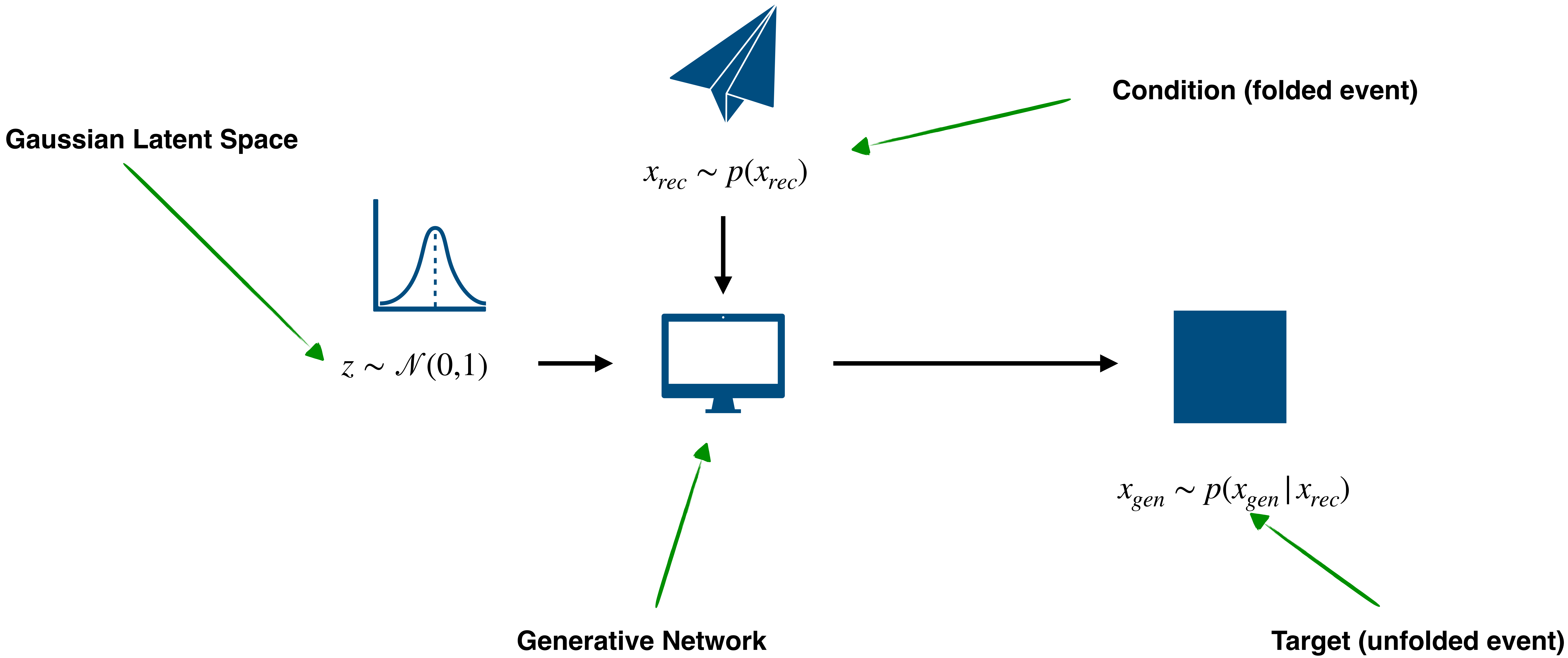
x_{rec}

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

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

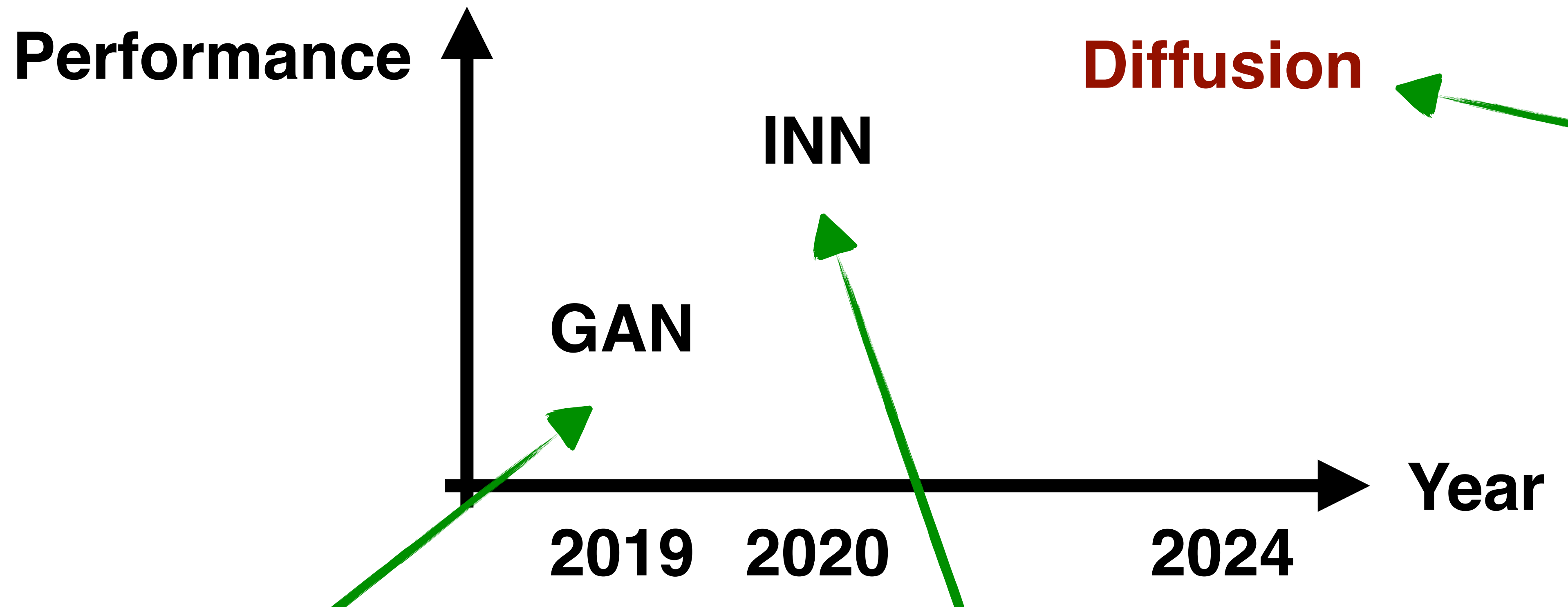
Unfolding probability ← **Encoded in neural network trained on simulation**

Conditional generative networks



Conditional generative networks

Which generative Network?



CFM
Huetsch et al.
arXiv:2404.18807
Butter et al.
arXiv:2411.02495
Favaro et al.
2501.12363

VLD
Shmakov et al.
arXiv: 2305.10399
arXiv: 2404.14332

DDPM
Pazos et al.
arXiv: 2406.01507

Bellagente et al.
arXiv:1912.00477

Bellagente et al.
arXiv:2006.06685

Slide adapted from Sofia Palacios Schweitzer

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)

Phase Space
$t = 0$

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

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

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

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

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

Latent Space
$t = 1$

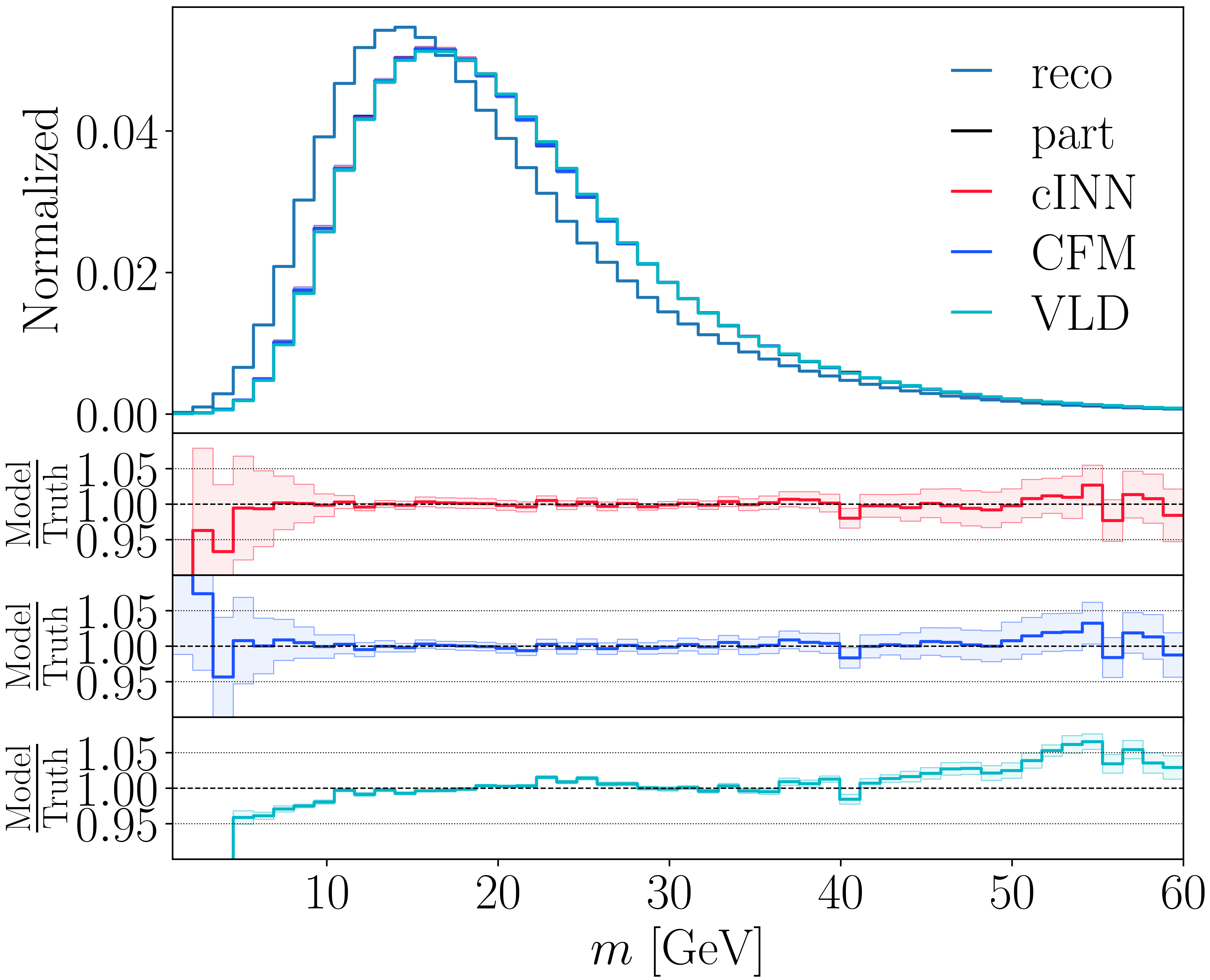
Individual Samples
$x_1 = \epsilon \sim p_1(x_1)$

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

6-dimensional unfolding

Z + jets events following Andreassen et al. arXiv: 1911.09107

6-dimensional phase space of jet observables



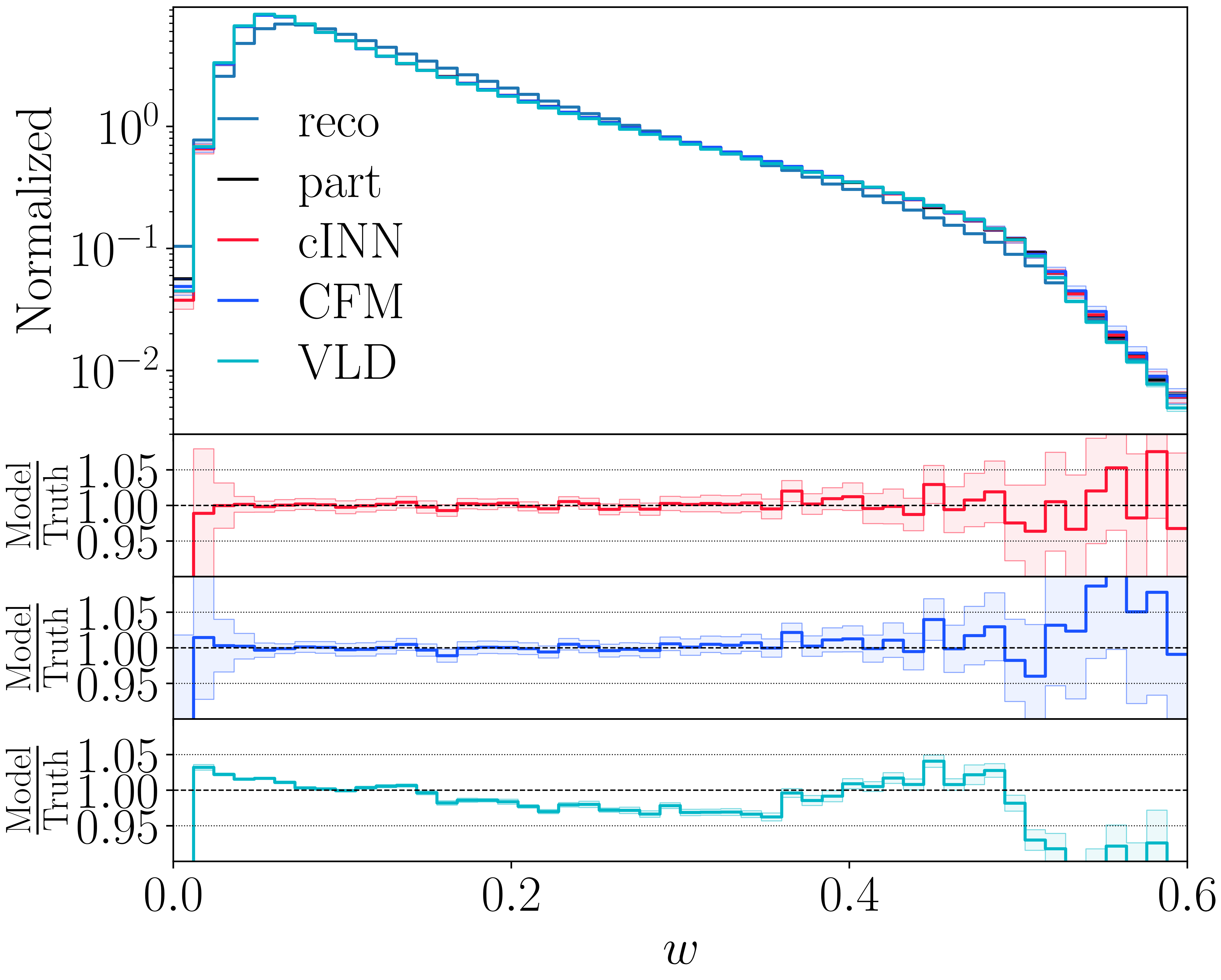


Figure from Huetsch et. al: arXiv:2404.18807

22-dimensional unfolding

Z + 2 jets events following ATLAS arXiv:2405.20041

22-dimensional phase space of μ-kinematics, jet-kinematics, jet-observables

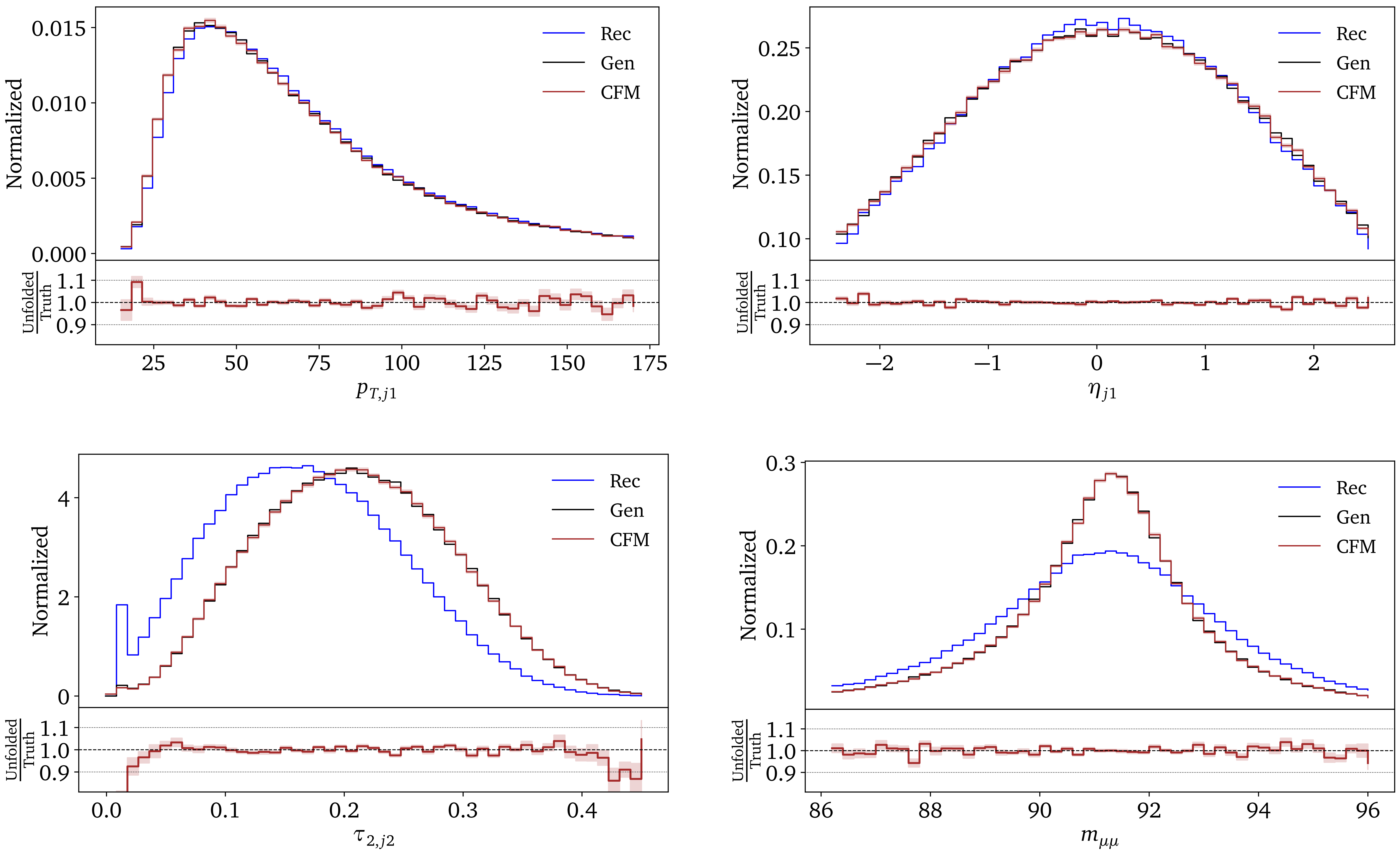


Figure adapted from Butter et. al: arXiv:2411.02495

22-dimensional unfolding

Z + 2 jets events following ATLAS arXiv:2405.20041

22-dimensional phase space of μ-kinematics, jet-kinematics, jet-observables

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

It learns the likelihood ratio

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

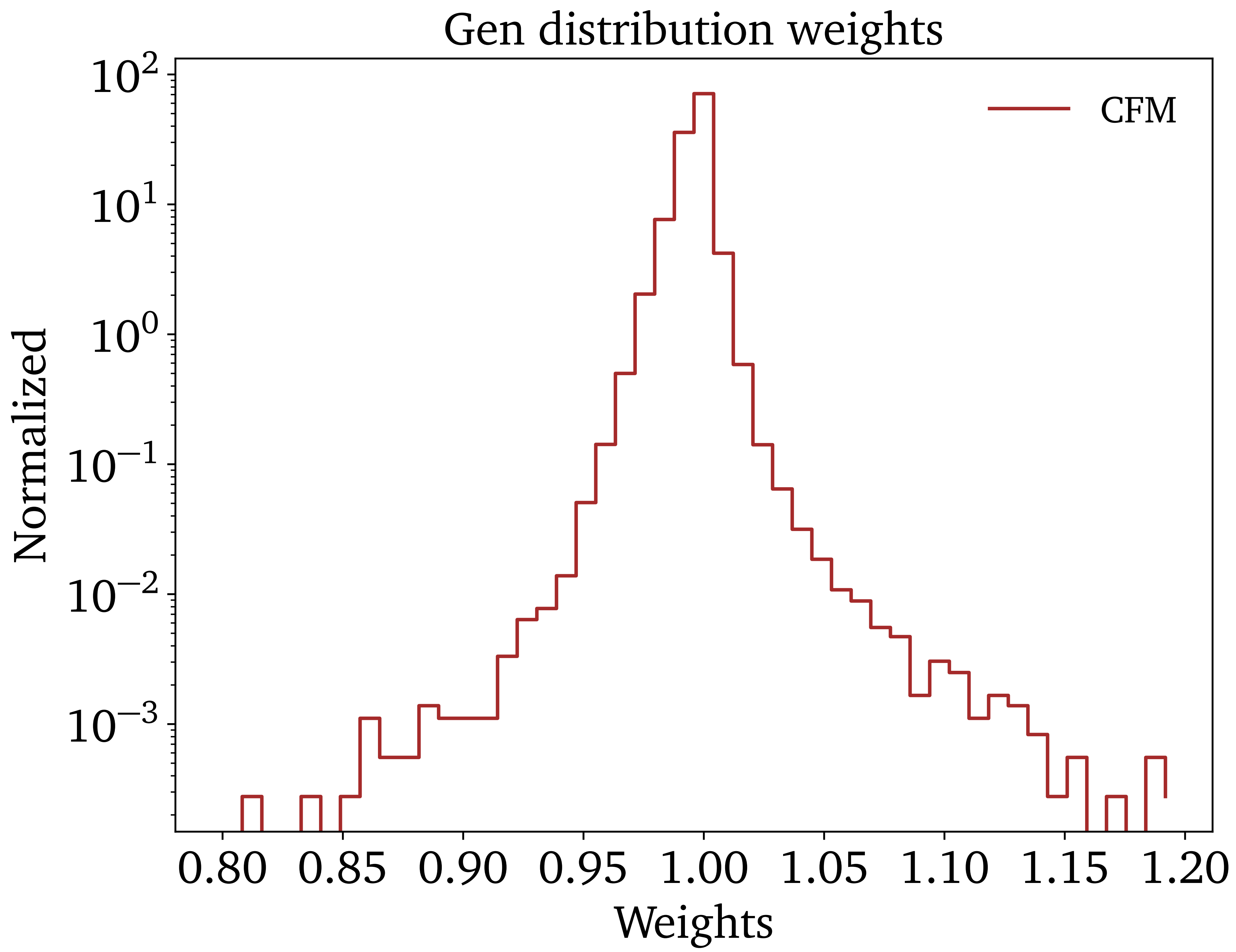
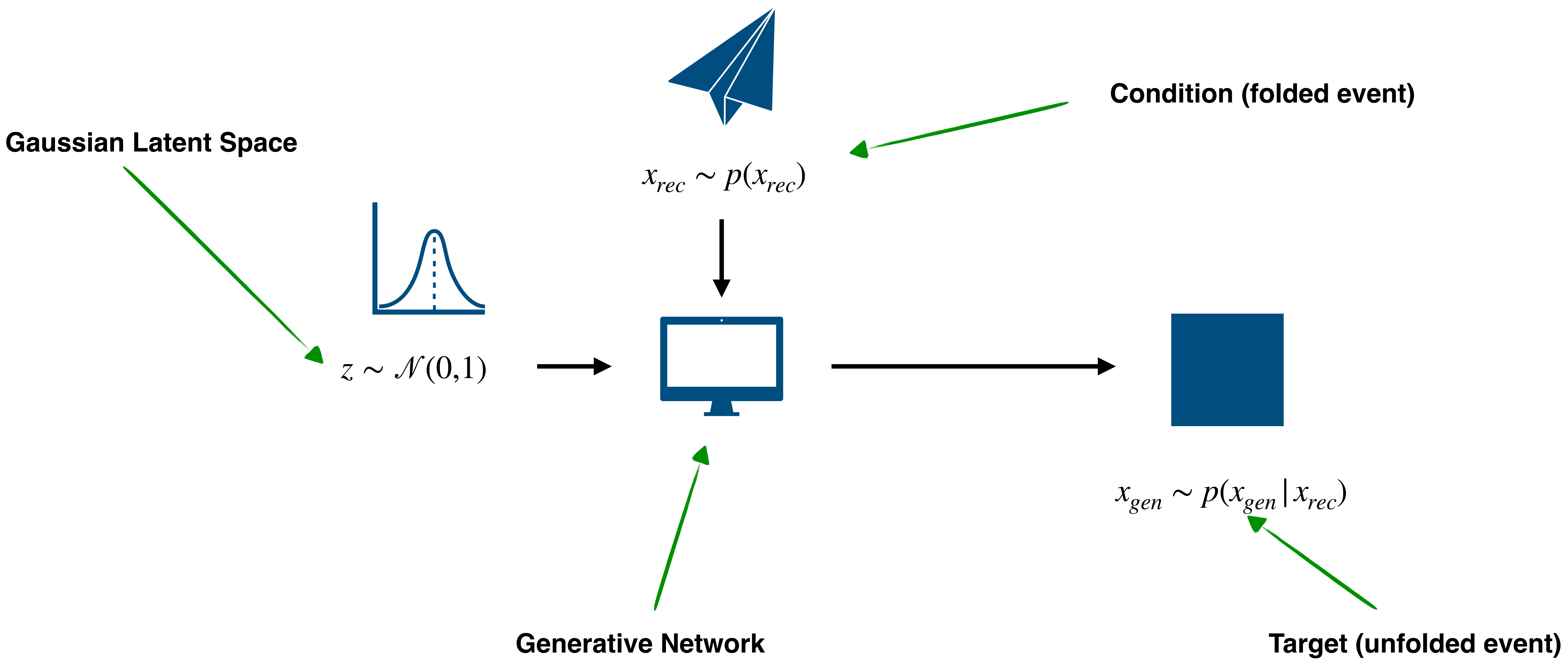


Figure adapted from Butter et. al: arXiv:2411.02495

Conditional generative networks



Distribution mapping

Condition (folded event)

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

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

Generative Network

Target (unfolded event)

Schrödinger Bridge

Diefenbacher et al.
arXiv:2308.12351

Direct Diffusion

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

Slide adapted from Sofia Palacios Schweitzer

Direct Diffusion (Butter et al. 2311.17175)

Unfolded Space
$t = 0$

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

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

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

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

Reco Space
$t = 1$

Individual Samples
$x_1 = x_{reco} \sim p_1(x_1)$

Density:
$p_1(x) = p(x_{reco})$

Z + jet results

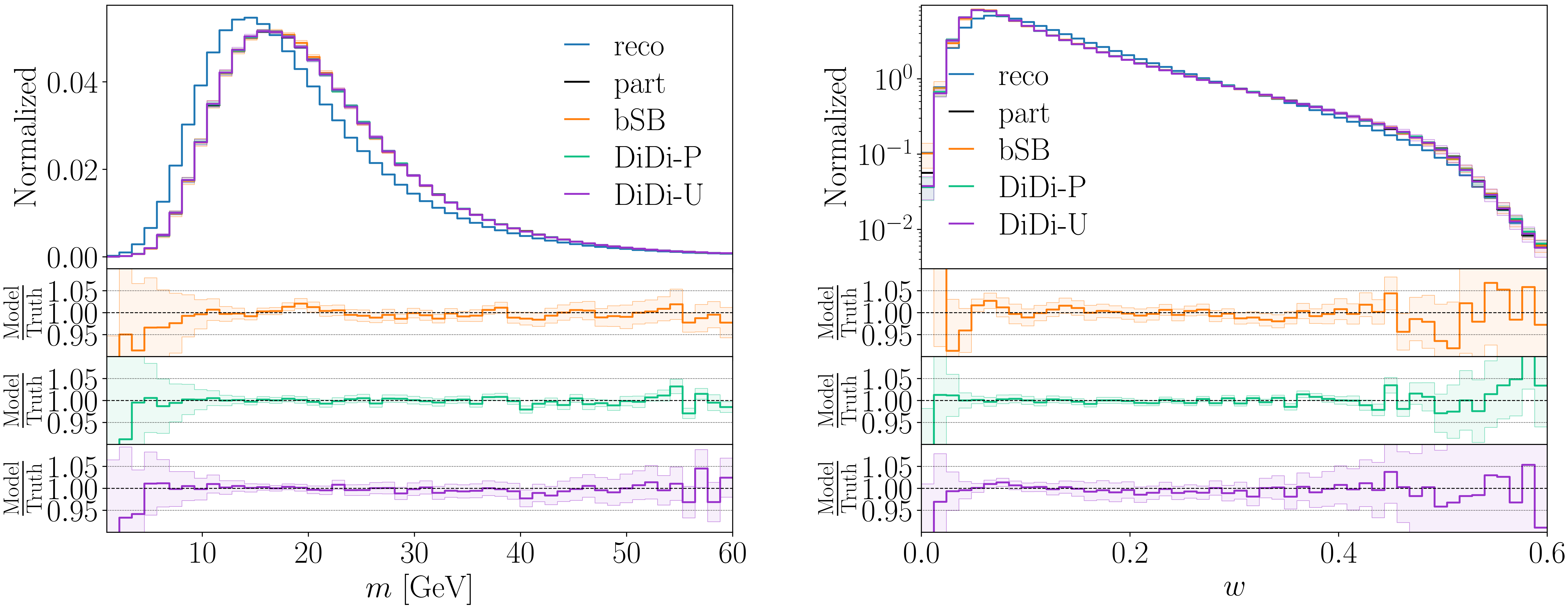


Figure from Huetsch et. al: arXiv:2404.18807

Z + jet migration

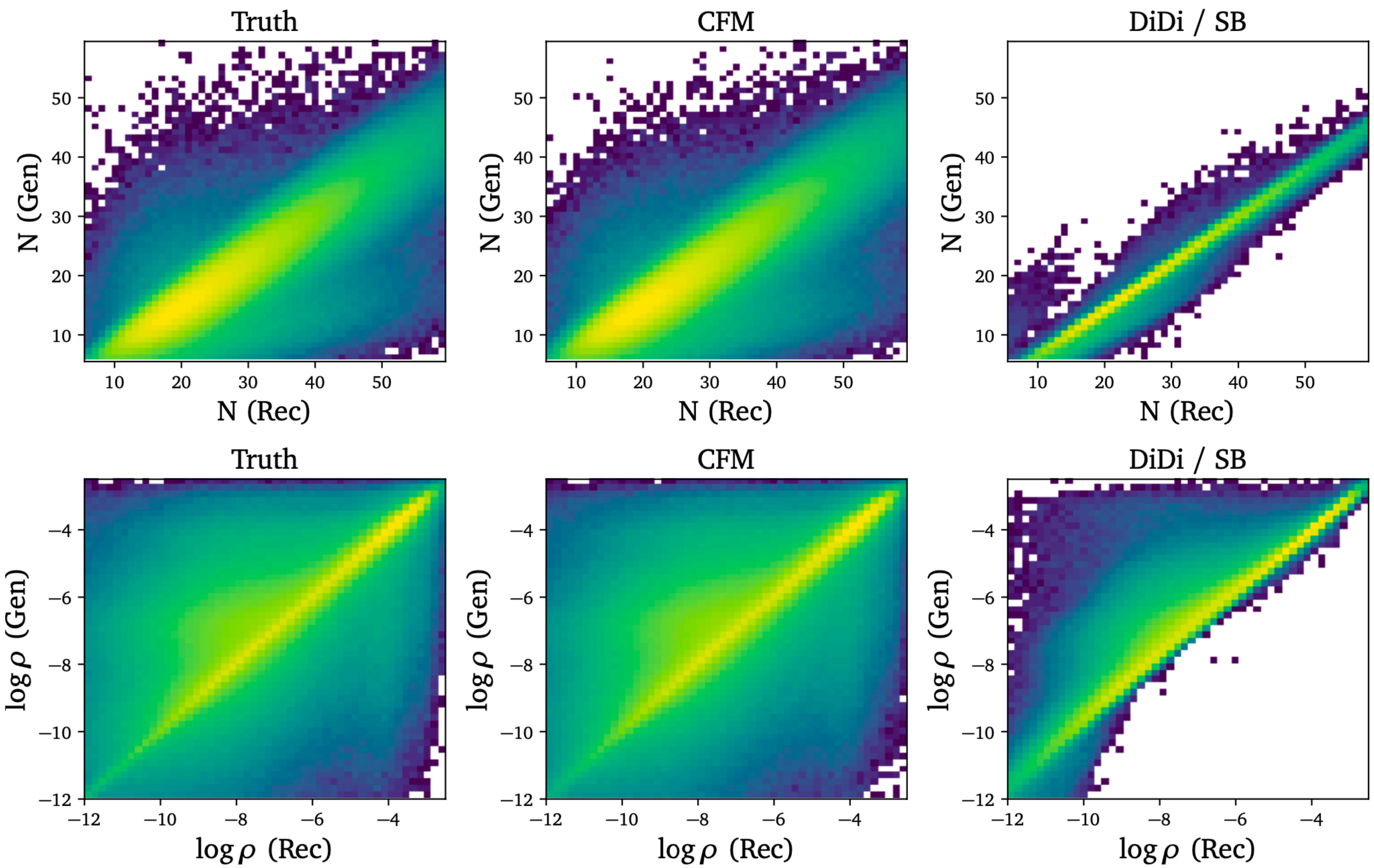


Figure adapted from Huetsch et. al: arXiv:2404.18807

Conditional distribution mapping

Condition (folded event)

Conditional Direct Diffusion
Conditional Schrödinger Bridge

Butter et al.
arXiv:2411.02495

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

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

Generative Network

Target (unfolded event)

Slide adapted from Sofia Palacios Schweitzer

Conditional distribution mapping

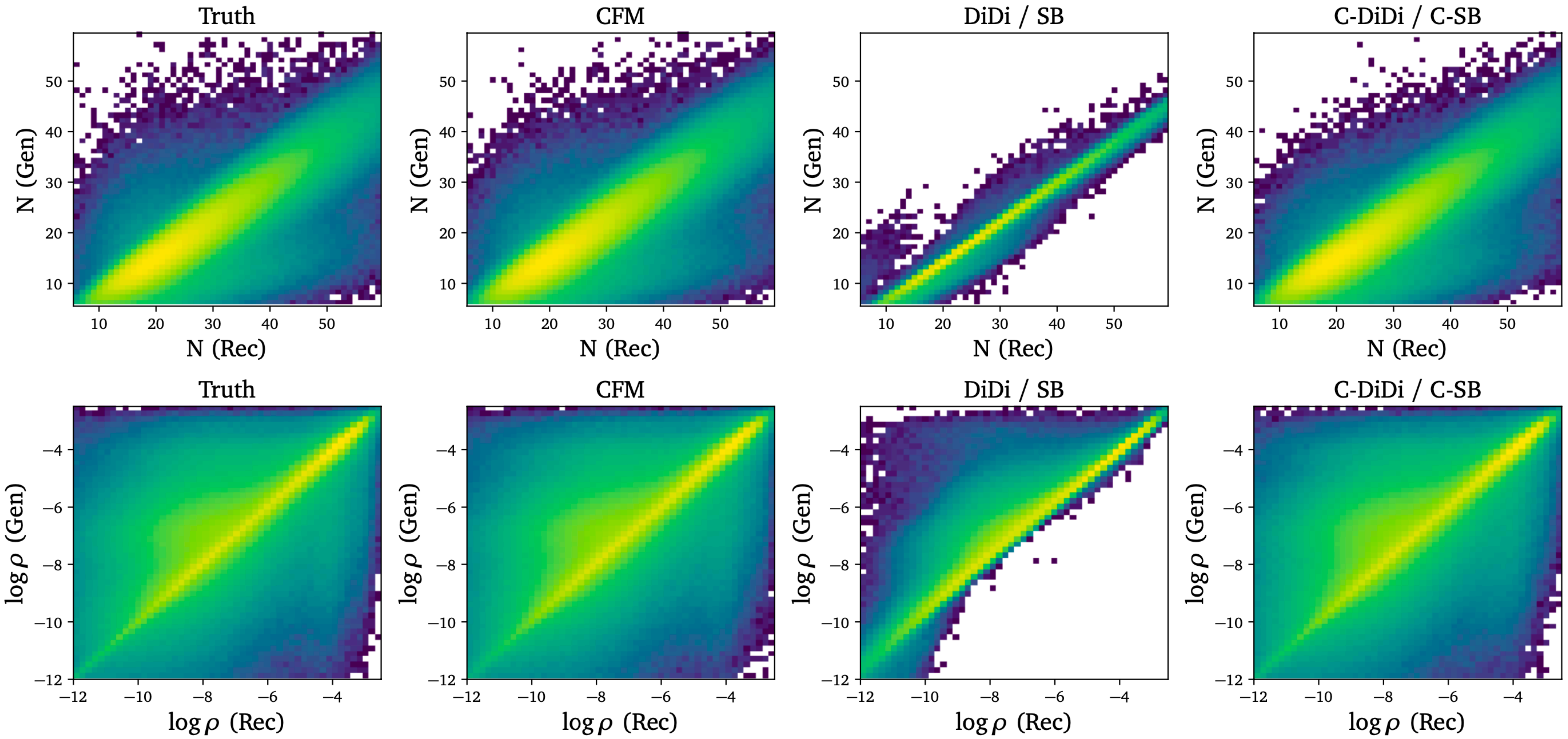


Figure adapted from Butter et. al: arXiv:2411. 02495

Conditional distribution mapping

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

It learns the likelihood ratio $w(x_{part}) = \dfrac{p_{true}(x_{part})}{p_{unfold}(x_{part})}$

Train a classifier classifier
between $p_{true}\ (x_{rec}, x_{part})$ and $p_{model}\ (x_{rec}, x_{part})$

It learns the likelihood ratio $w(x) = \dfrac{p_{true}\ (x_{rec}, x_{part})}{p_{model}\ (x_{rec}, x_{part})}$

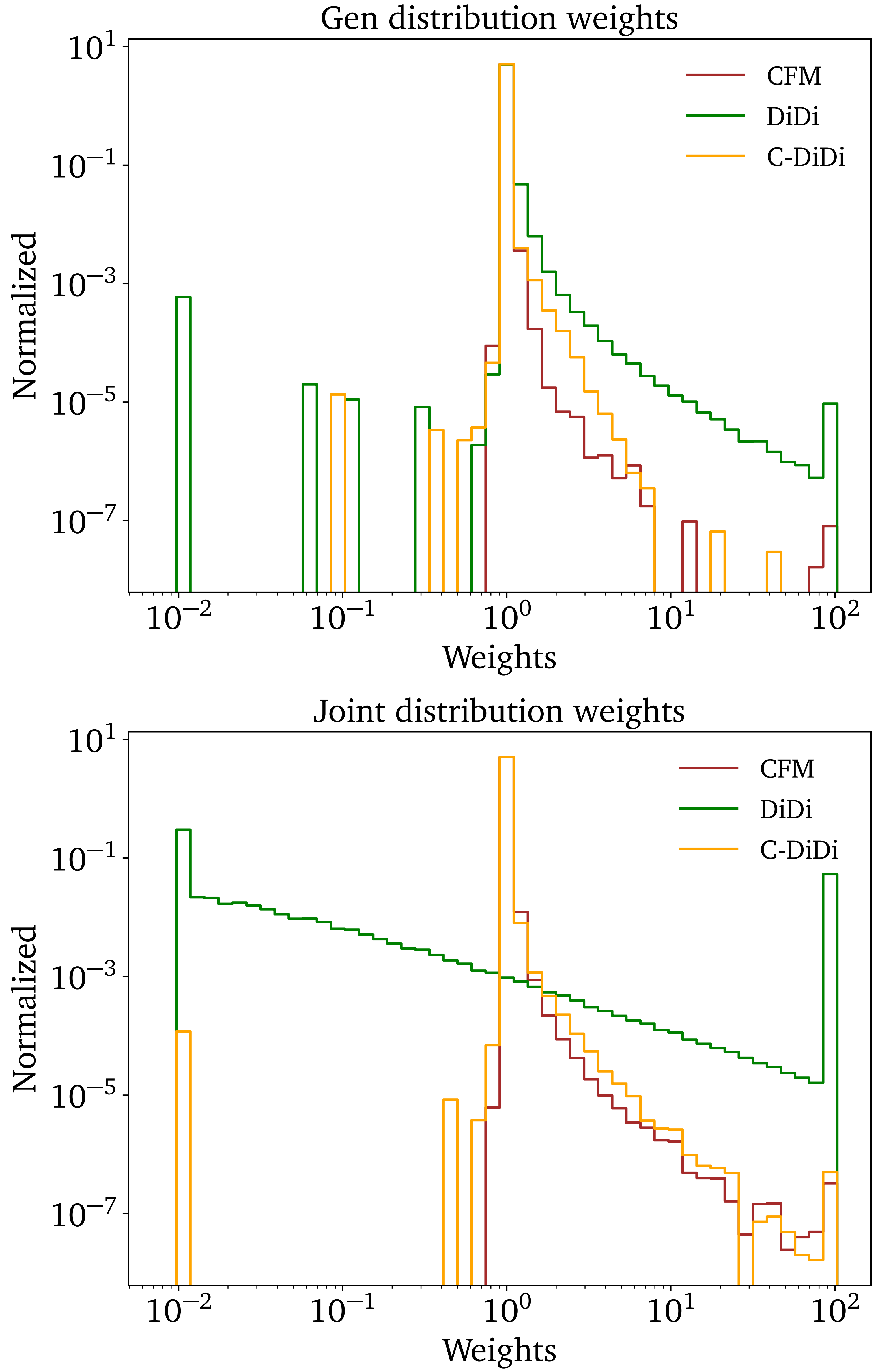


Figure adapted from Butter et. al: arXiv:2411.02495

Conclusion

ML Unfolding works ! ← ATLAS arXiv:2405.20041

Conditional Generative Unfolding enables probabilistic inversion of simulation chain

Classifier test reveals no artefacts and/or miss-modelled correlations

Distribution Mapping is a new and evolving ML-approach to generative unfolding

Further investigation of uncertainties

Incorporate into analysis pipeline ←

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]

SciPost Physics **Submission**

Generative Unfolding with Distribution Mapping

Anja Butter[1,2], Sascha Diefenbacher[3], Nathan Huetsch[1], Vinicius Mikuni[4], Benjamin Nachman[3,5], Sofia Palacios Schweitzer[1] and Tilman Plehn[1,6]

How to Unfold Top Decays	*Sofia Palacios Schweitzer*
Amphi Grünewald, IPHC	16:20 - 16:35

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**

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

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

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

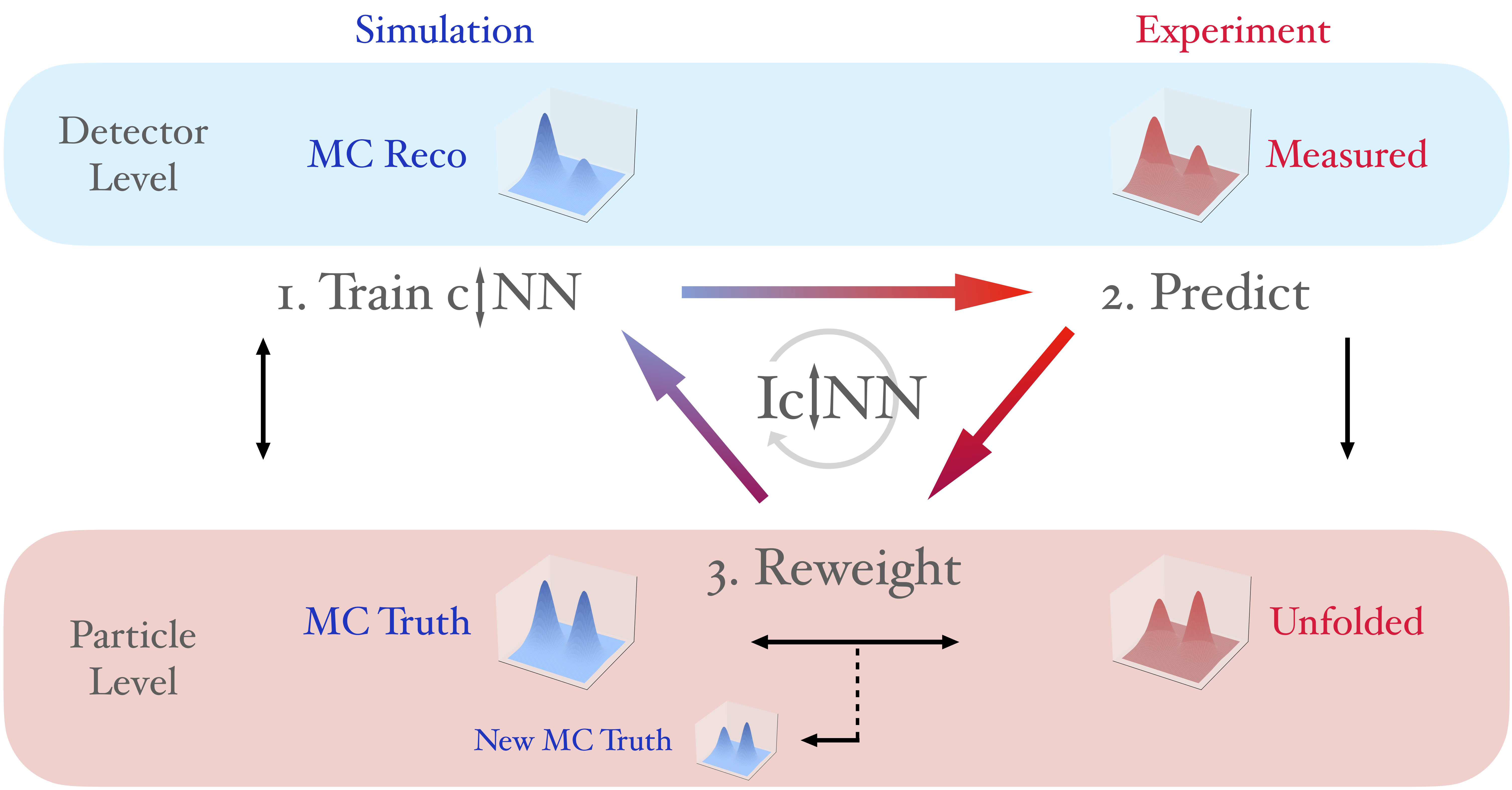
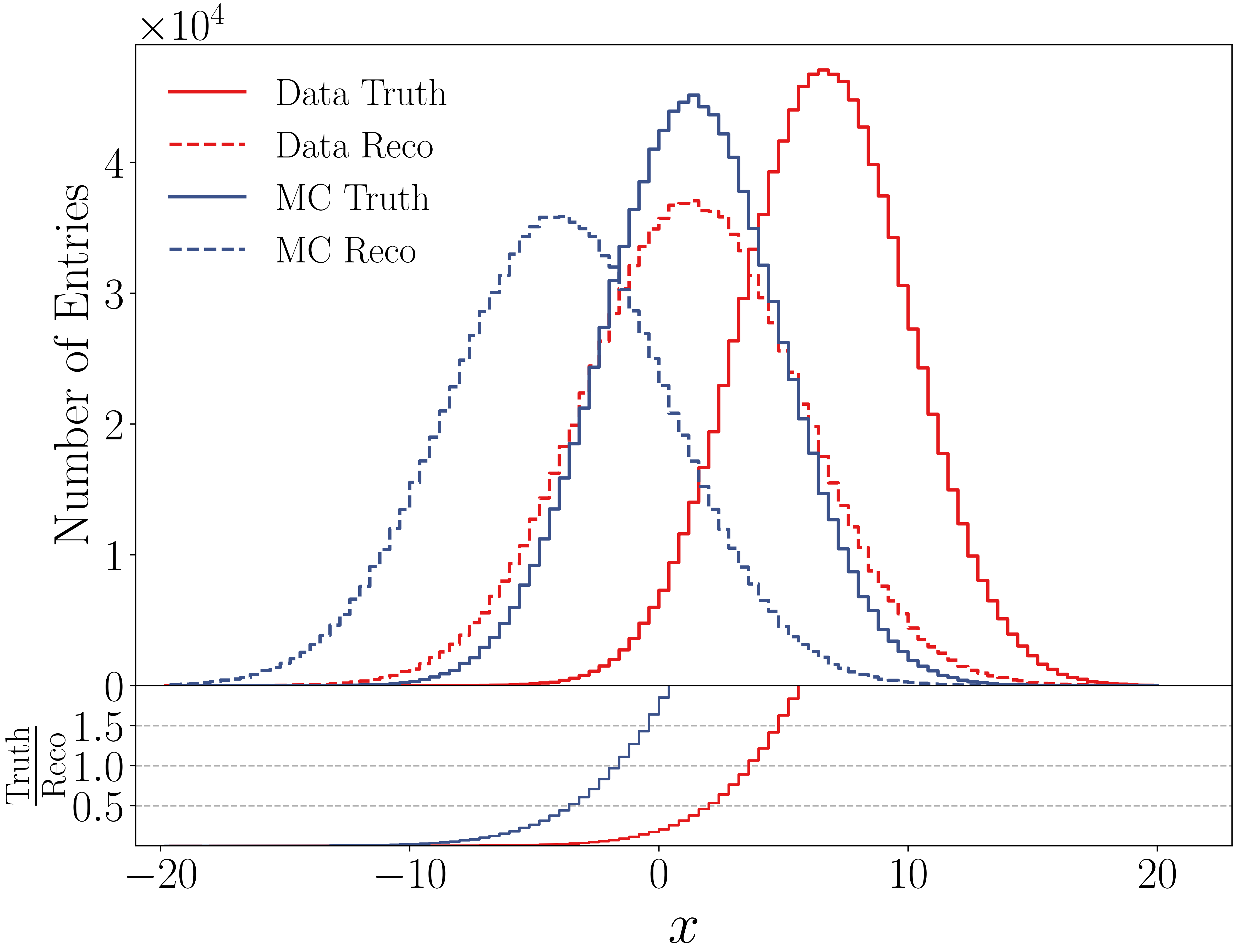
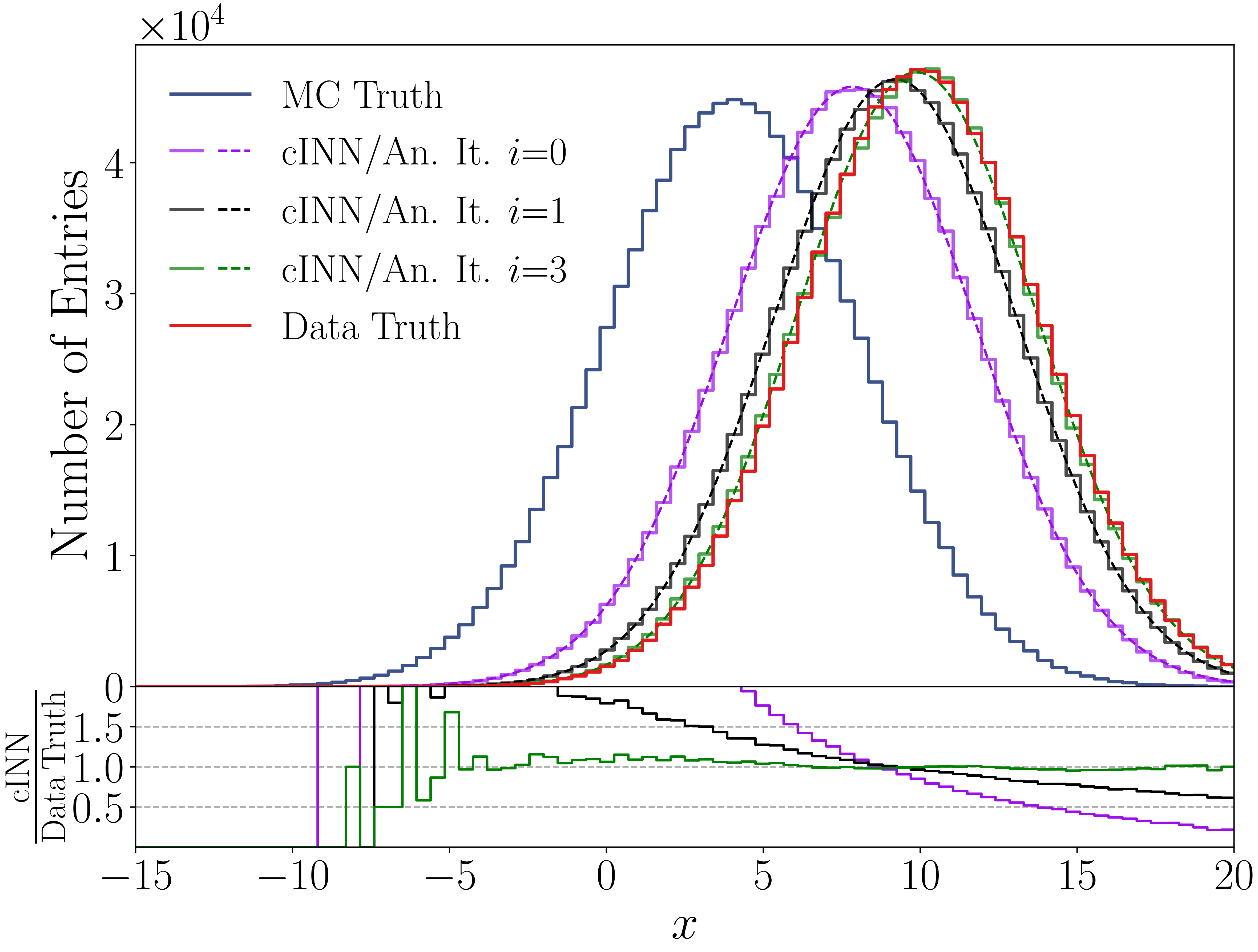


Figure from Backes et al.: arXiv:2212.08674

Iterative generative unfolding



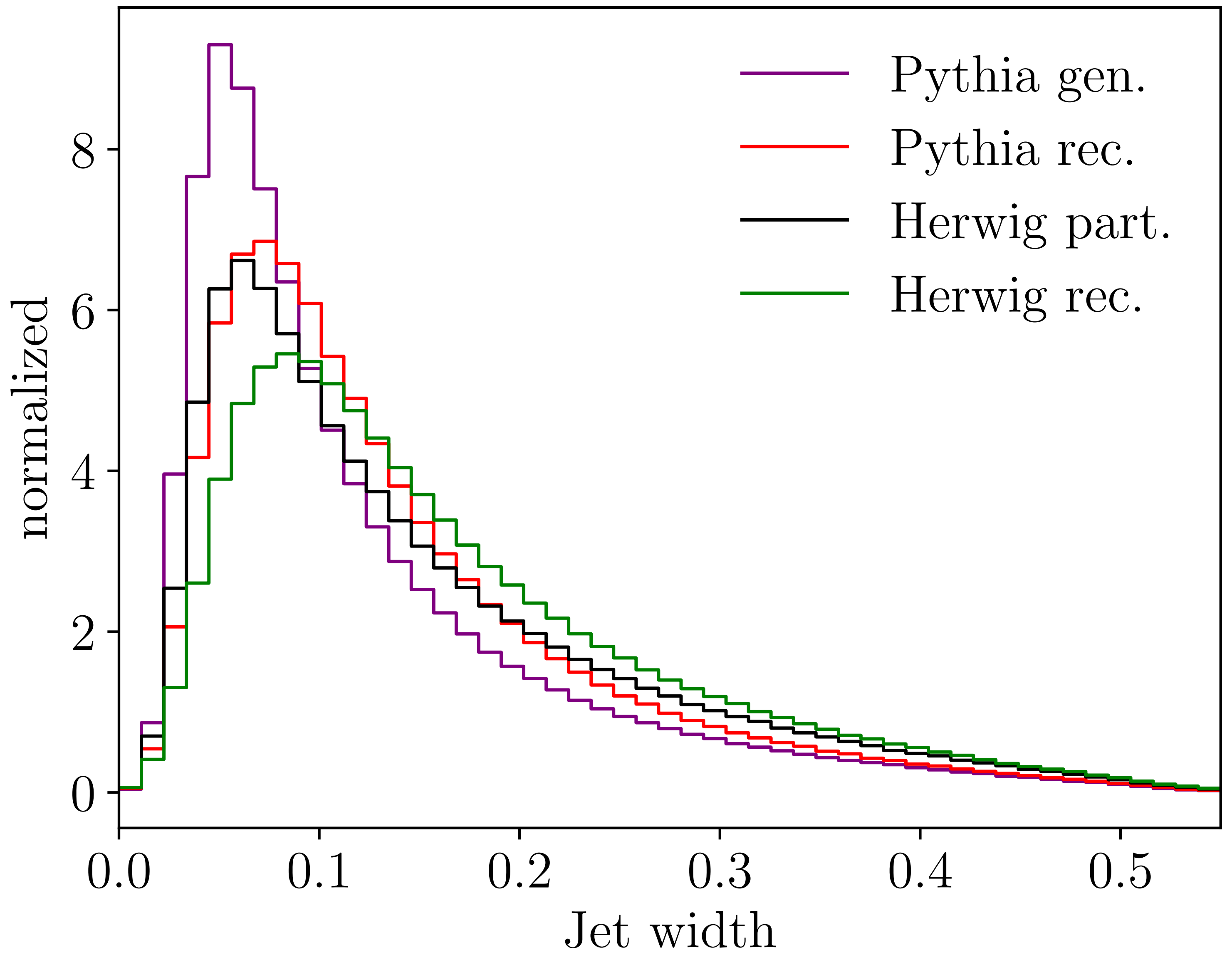


Figures from Backes et al.: arXiv:2212.08674

Z+jets: Pythia vs Herwig simulation

Use Pythia simulation as MC

Use Herwig simulation as Data



Following Andreassen et al. arXiv: 1911.09107