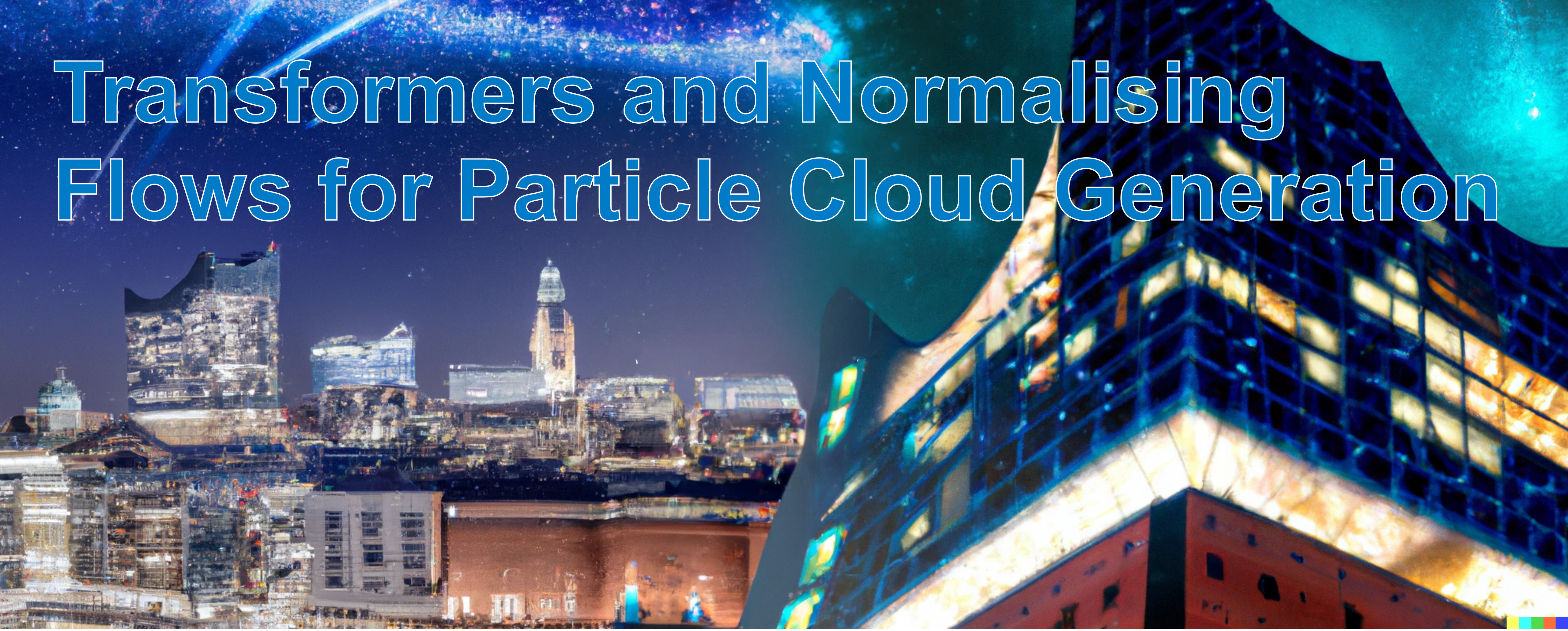


Transformers and Normalising Flows for Particle Cloud Generation

Artwork by $\mathrm{DALL}-\mathrm{E}\cdot 2$

ML4Jets Conference, 01. Nov 2022

<u>Benno Käch</u>, Dirk Krücker, Isabell Melzer-Pellmann

benno.kaech@desy.de

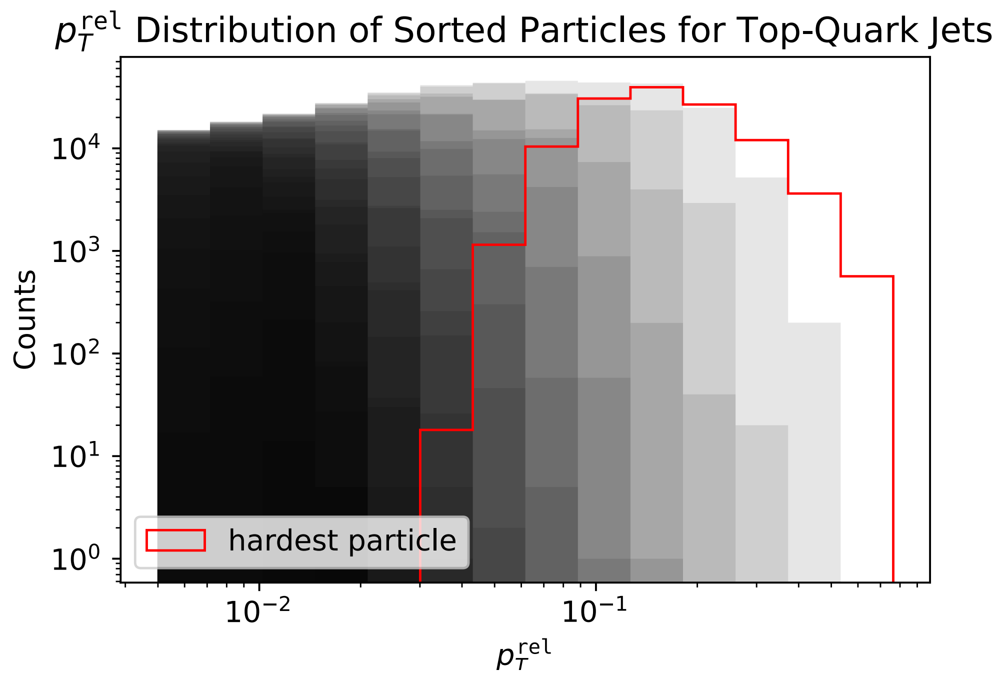
HELMHOLTZAI

CLUSTER OF EXCELLENCE QUANTUM UNIVERSE [1]

Particle Cloud Generation

JetNet [1] Datasets

- Gluon, light and top-quark Pythia jets, clustered by anti-k_T
- Jets of about $p_T^{\mathtt{jet}} \sim 1\ TeV$
- Particles: tuples of $(\eta^{\mathtt{rel}}, \phi^{\mathtt{rel}}, p_T^{\mathtt{rel}})$ relative to jet axis
- Constrained to max 30 particles/jet
- Invariant jet mass: $m^2 = \left(\sum_{i=1}^{30} |\boldsymbol{p}_i|\right)^2 - \left(\sum_{i=1}^{30} \boldsymbol{p}_i\right)^2$
- Size $\sim 178'000$ Samples
- $(70/30)$ Train/Test split
- **Benchmarking possible**



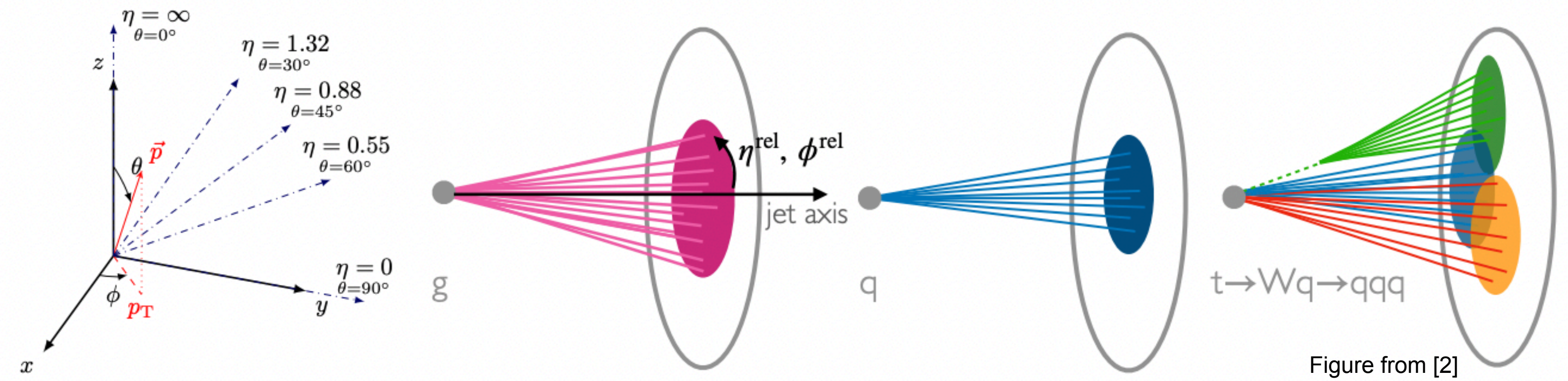


Figure from [2]

[1] Kansal et al, `JetNet`, PyPi
[2] Kansal et al., Particle Cloud Generation with Message Passing Generative Adversarial Networks, arxiv.org/abs/2106.11535

Assessing Performance

Same Metrics as in [2]

- Track multiple metrics for performance:
  - Wasserstein-1 distance W_1 on different distributions (see below)
  - Fréchet ParticleNet Distance (FPND) [2]
  - Coverage (COV)
  - Minimum Matching Distance (MMD)

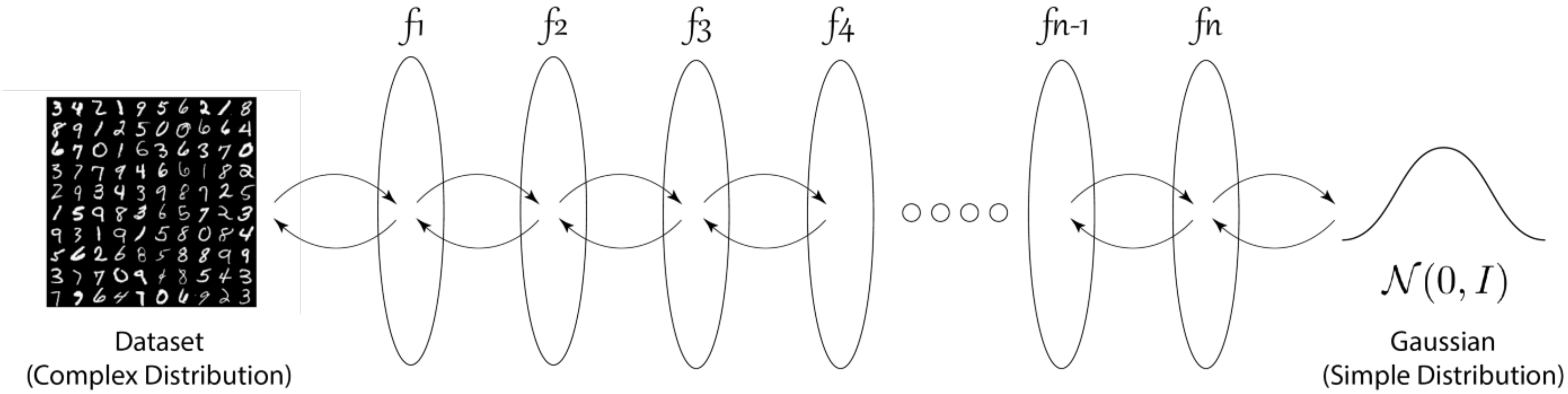
<u>In-sample distances</u>

Parton	$W_1^M (\times 10^{-3})$	$W_1^P (\times 10^{-3})$	$W_1^{EFP} (\times 10^{-5})$	FPND	COV $\uparrow$	MMD
Gluon	0.5 ± 0.1	0.4 ± 0.2	0.4 ± 0.4	0.01	0.56	0.036
Light Quark	0.42 ± 0.09	0.6 ± 0.4	0.5 ± 0.5	0.01	0.55	0.024
Top Quark	0.5 ± 0.1	0.6 ± 0.4	1.1 ± 0.4	0.03	0.56	0.072

[2] Kansal et al., Particle Cloud Generation with Message Passing Generative Adversarial Networks, arxiv.org/abs/2106.11535.pdf

Normalising Flows

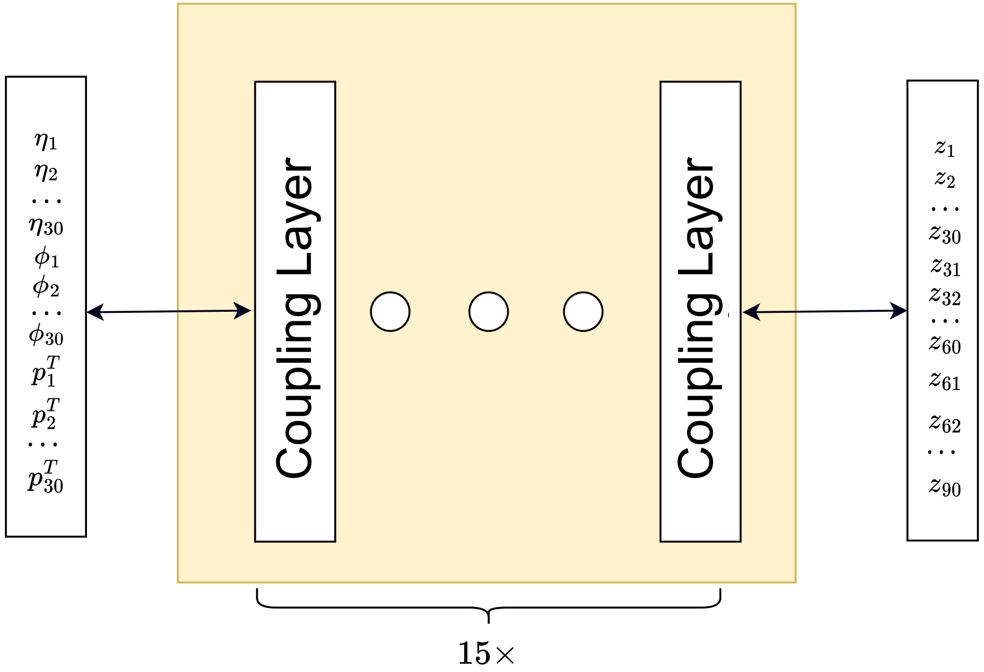
- *Find invertible functions to transform data distribution to Normal distribution*
- Invertible functions due to smart construction: **Coupling Layers**
- Stack multiple Coupling Layers for expressivity
- **Contrast to GAN → Stable Maximum-Likelihood training**



Normalising Flow Architecture

`nflows` [3] implementation used

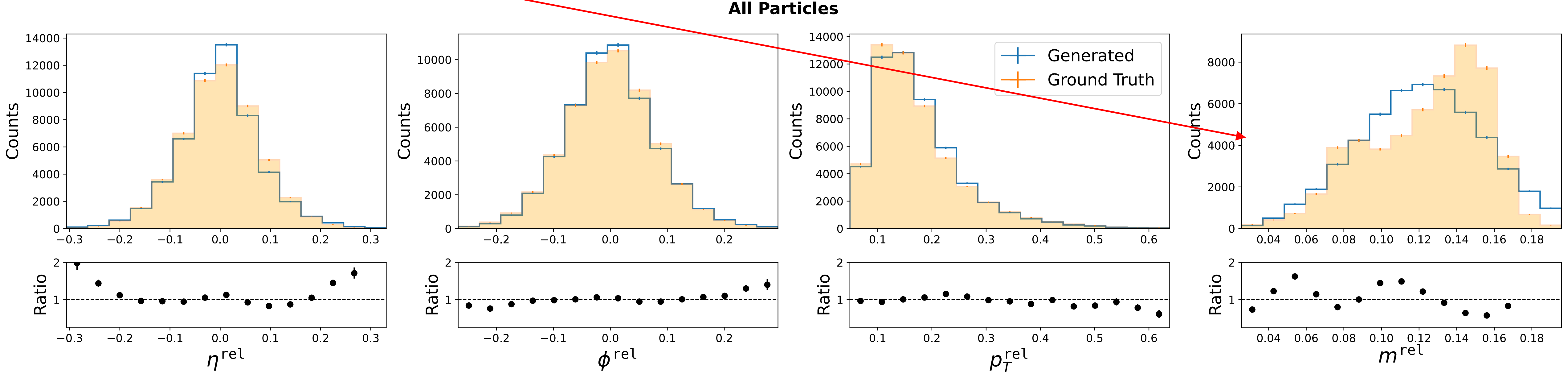
- Vanilla Normalising Flow: $90(=30\times 3)$ dimensional latent space
- Rational Quadratic Splines Coupling Layers [4]
- No permutation invariant encoding, particles ordered by p_T^{rel}
- Jets with less < 30 particles zero-padded & noise added $O(10^{-7})$
- **No inductive bias → contrast to other generative models**



[3] Durkan and Bekasov et al., https://github.com/bayesiains/nflows
[4] Durkan and Bekasov et al., Neural Spline Flows, arxiv.org/abs/1906.04032.pdf

Pitfall of Normalising Flows

Model	$W_1^M (\times 10^{-3})$	$W_1^P (\times 10^{-3})$	$W_1^{EFP} (\times 10^{-5})$	FPND	COV $\uparrow$	MMD
VNF	6.4 ± 0.2	2.2 ± 0.2	14 ± 1	7.91	0.56	**0.071**

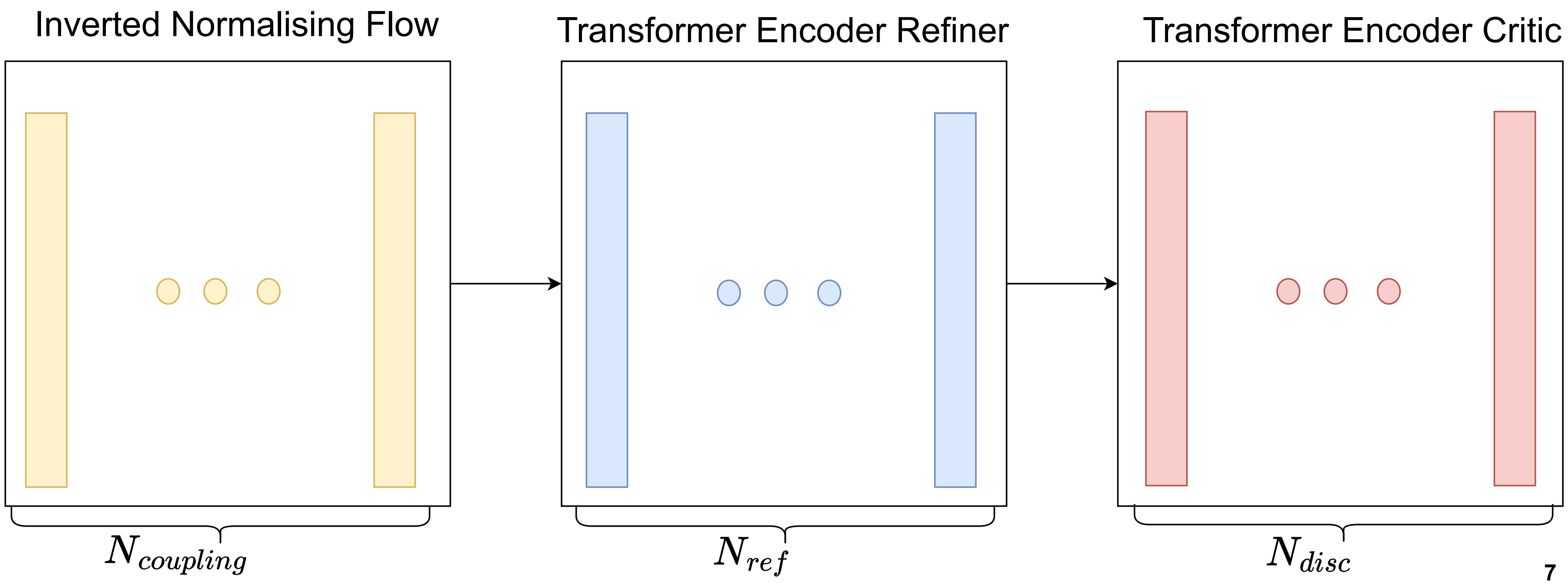


Mass Modelled Incorrectly

- Due to Coupling Layer construction?
- Plus-side: training takes 1-2 h, always converges

Refinement Setup

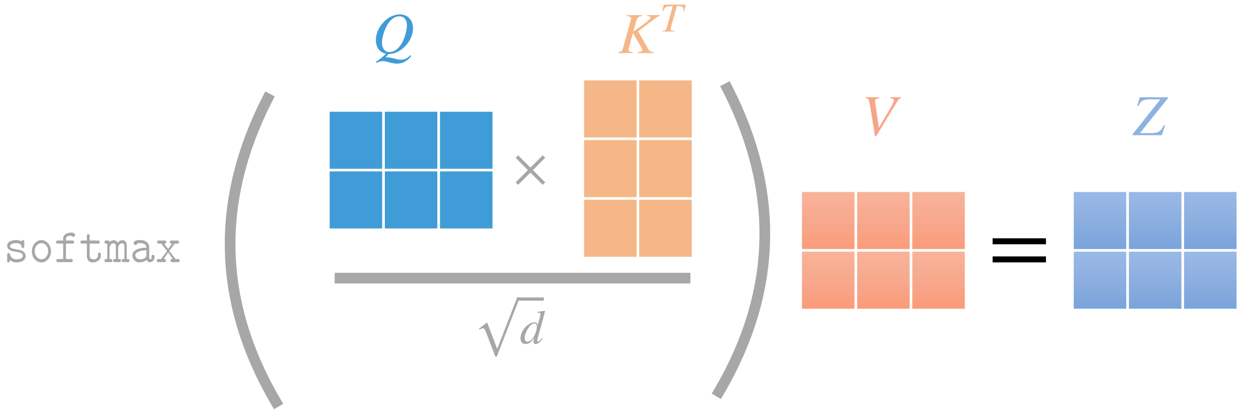
- Additive correction by Transformer Encoder Refinement Network $\boldsymbol{R}$: $\boldsymbol{x} = \boldsymbol{x}_{NF} + \boldsymbol{R}(\boldsymbol{x}_{NF})$
- Refinement trained adversarially with Transformer Encoder Critic $C(\boldsymbol{x}) \in \mathbb{R}$
- No gradient for NF from critic



Self-Attention

Attention is all you need! [5]

- Commonly used in NLP
- Permutation invariant
- Self-Attention: n inputs, n outputs - interaction between inputs
- Particles attend to other particles with strength: $\texttt{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \frac{\texttt{softmax}(\boldsymbol{Q} \cdot \boldsymbol{K}^{\mathrm{T}} + \boldsymbol{M} \cdot (-\infty))}{\sqrt{\texttt{d}}} \boldsymbol{V}$
- $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ Linear embeddings of input $\rightarrow \boldsymbol{Q} = \boldsymbol{W}_Q \boldsymbol{x}$, $\boldsymbol{K} = \boldsymbol{W}_K \boldsymbol{x}$, $\boldsymbol{V} = \boldsymbol{W}_V \boldsymbol{x}$
- $\boldsymbol{M} = 1$ mask for jets with < 30 particles $\rightarrow$ No influence

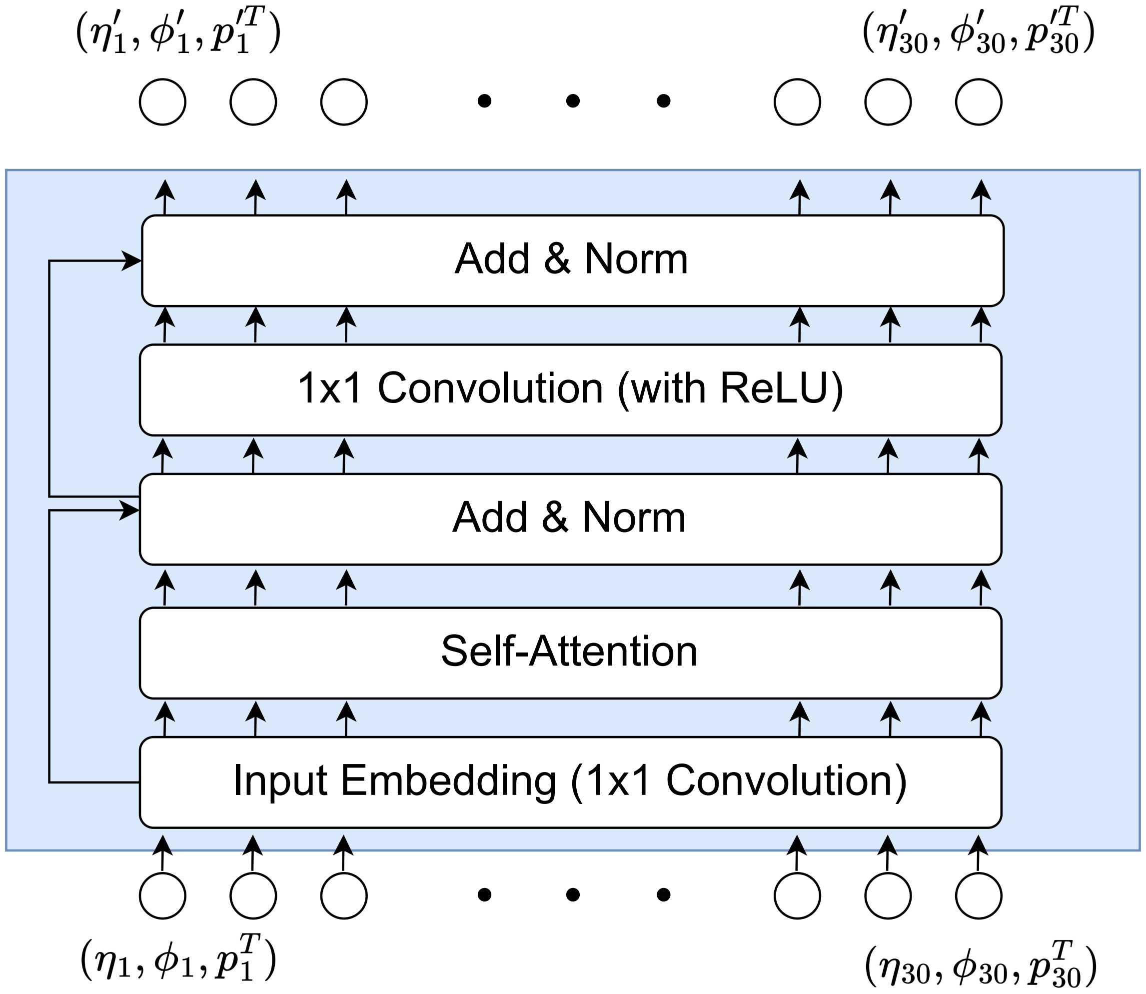


[5] A. Vaswani et al., "Attention Is All You Need", https://arxiv.org/abs/1706.03762

Transformer Encoder Refinement

Ingredients:

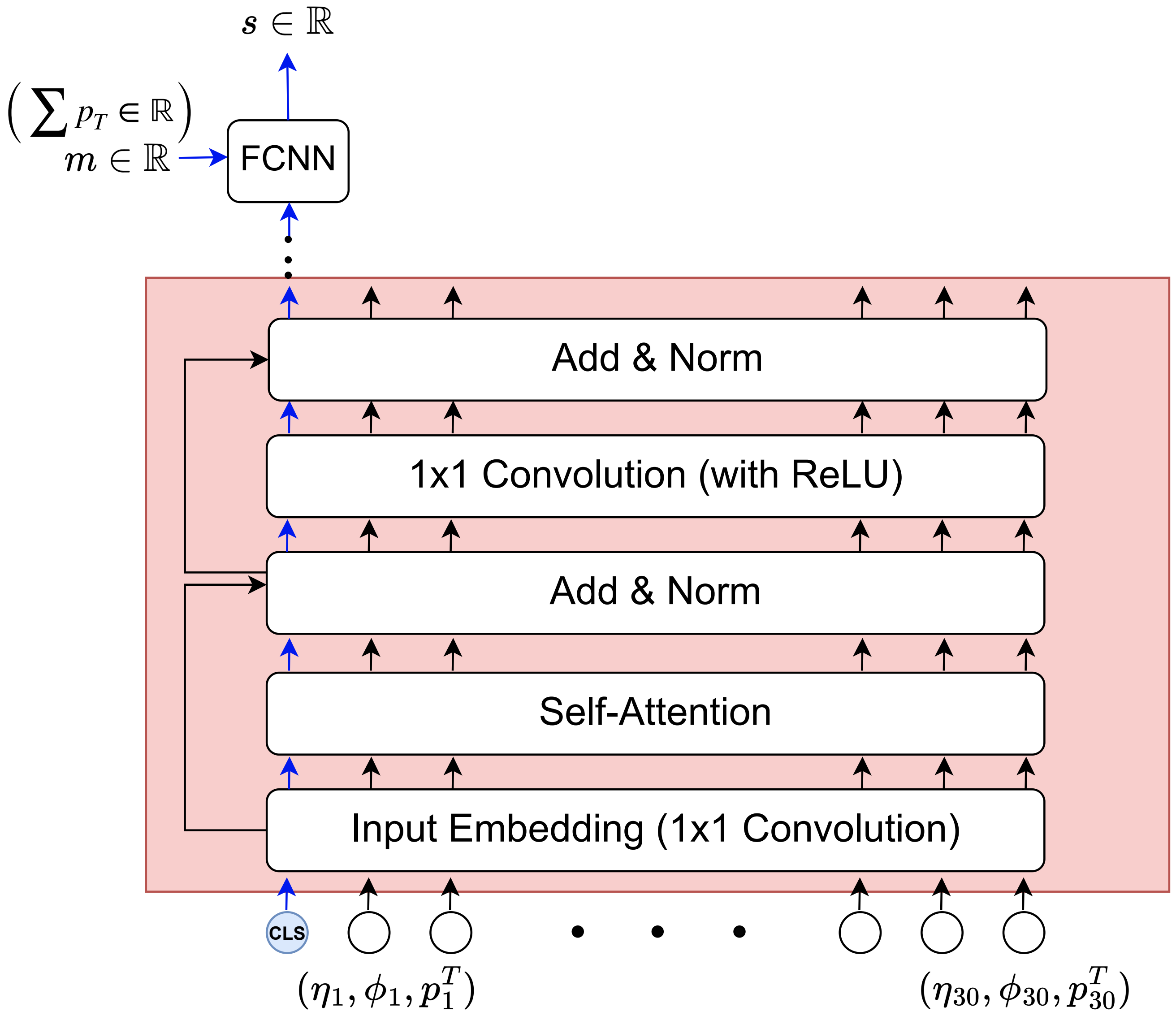
- Embedding: same linear embedding for all particles
- Residual connection
- Layer Normalisation after addition
- 1x1 Convolution with nonlinearity
- Mask sampled from training data
- **Permutation Invariant Network Architecture**



Transformer Encoder Critic

Additional Ingredients:

- Classification token/particle
- Fully Connected Critic Network
- Mass as additional input
- Optionally also sum of transverse momenta



Training Details

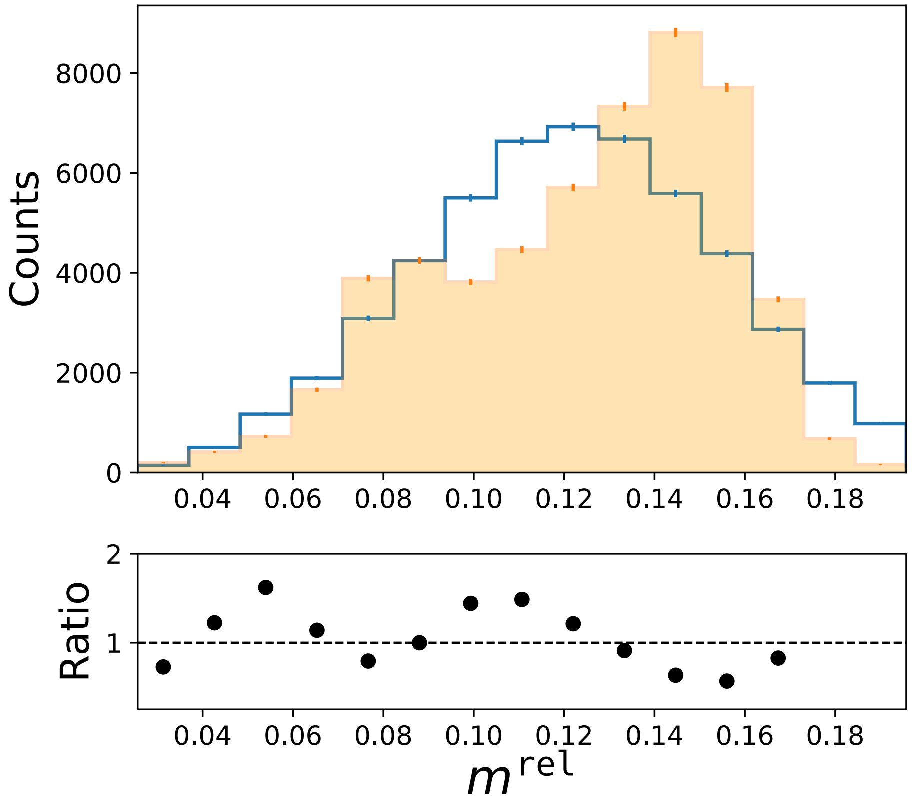
- Particles scaled to zero mean unit variance
- Linear Warmup learning rate scheduling
- LSGAN trained with MSE
- Batch Size 1000-4000
- Dropout during Training & Evaluation
- $\sim 11 - 48\ h$ on $\mathtt{NVIDIA\ P100}$
- Not that stable anymore :(

From xkcd

Results - Top-Quarks and Light-Quarks

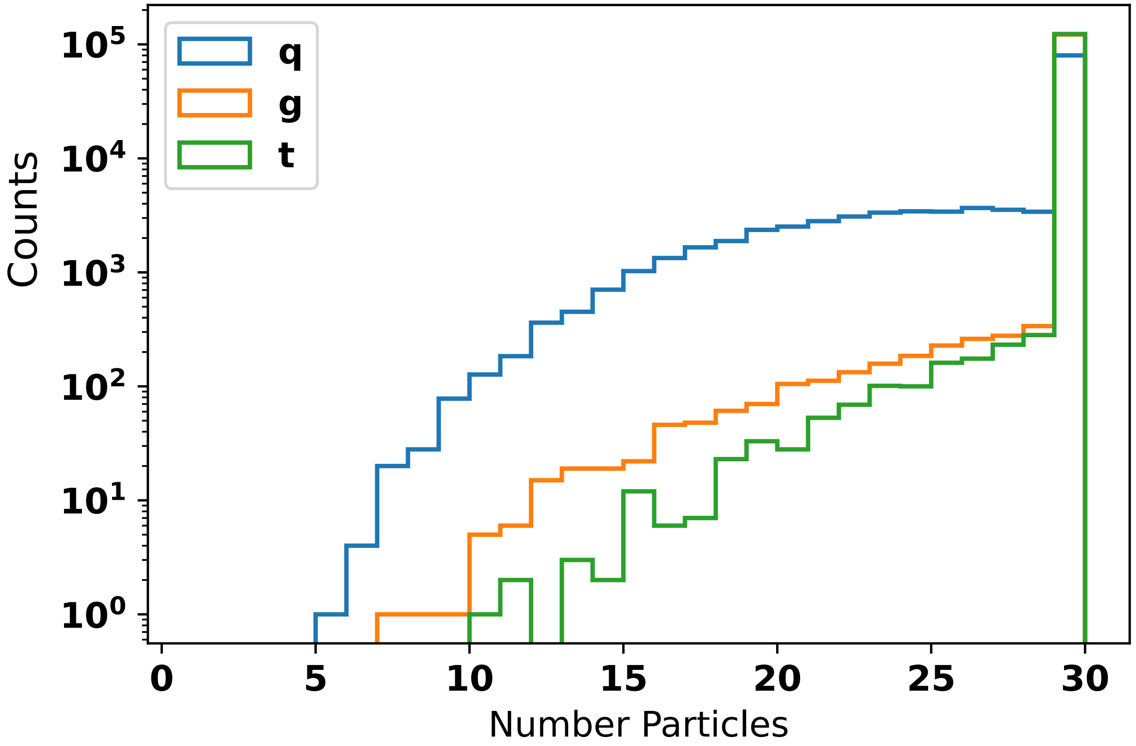
<u>Top Quarks</u>

- Complex substructure
- Normalising Flow performs worst

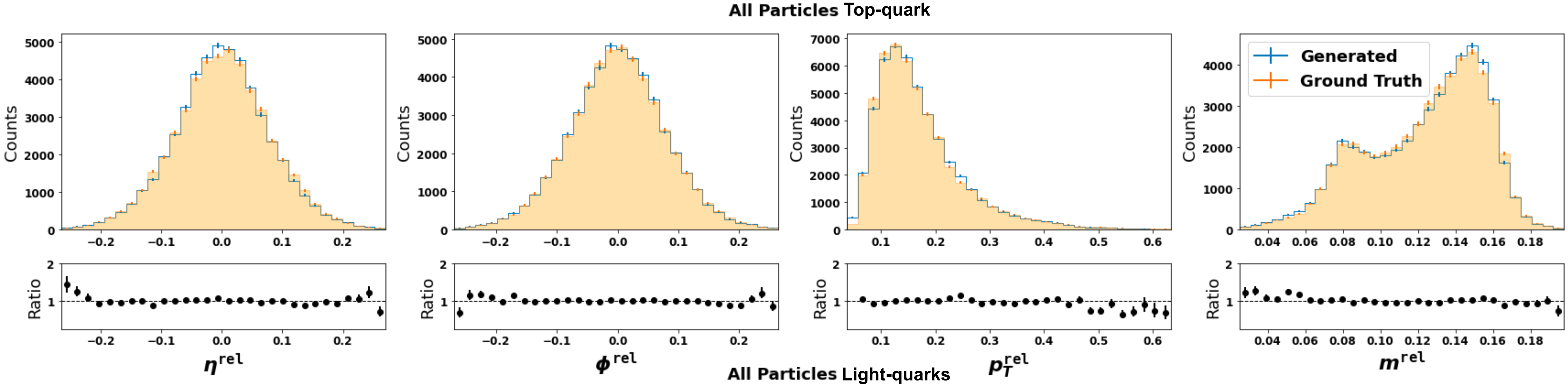


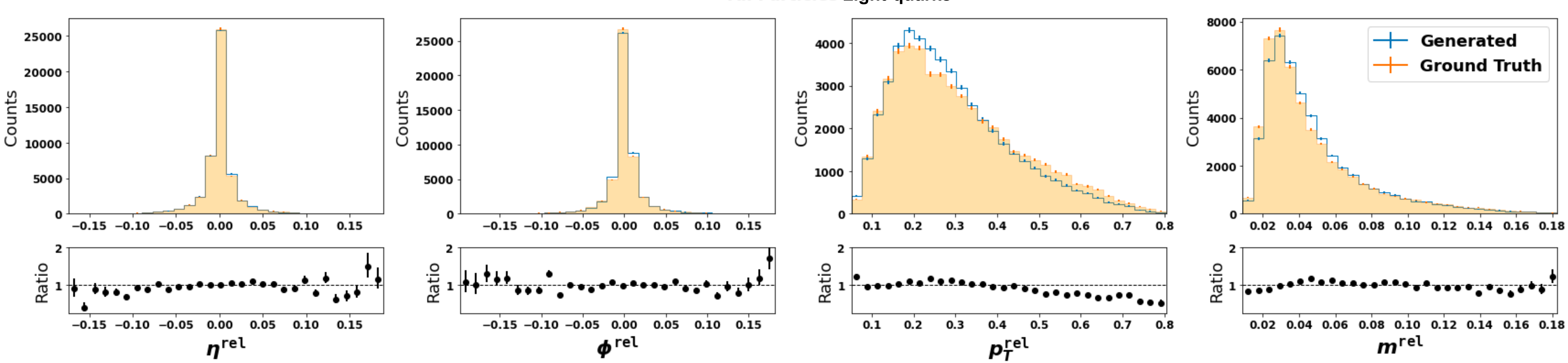
<u>Light Quarks</u>

- Highest variability in number particles



Results


All Particles Top-quark
Counts
Ratio
η^{rel}
ϕ^{rel}
p_T^{rel}
m^{rel}
Generated
Ground Truth


All Particles Light-quarks
Counts
Ratio
η^{rel}
ϕ^{rel}
p_T^{rel}
m^{rel}
Generated
Ground Truth

Results

- Competitive with state-of-art
- Scalability to be investigated

In-sample distances

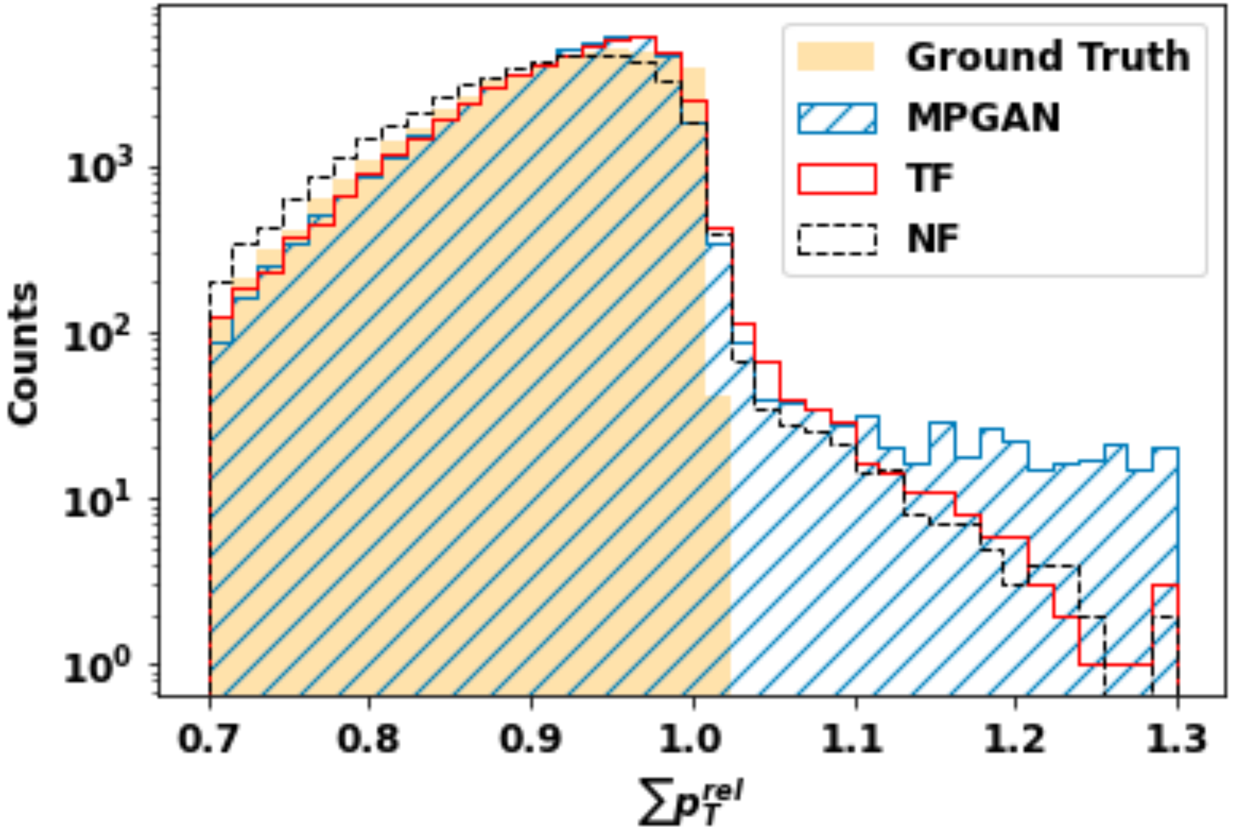
Parton	$W_1^M(\times10^{-3})$	$W_1^P(\times10^{-3})$	$W_1^{EFP}(\times10^{-5})$	FPND	COV $\uparrow$	MMD
Gluon	0.5 ± 0.1	0.4 ± 0.2	0.4 ± 0.4	0.01	0.56	0.036
Light Quark	0.42 ± 0.09	0.6 ± 0.4	0.5 ± 0.5	0.01	0.55	0.024
Top Quark	0.5 ± 0.1	0.6 ± 0.4	1.1 ± 0.4	0.03	0.56	0.072

Quark	Model	$W_1^M(\times10^{-3})$	$W_1^P(\times10^{-3})$	$W_1^{EFP}(\times10^{-5})$	FPND	COV $\uparrow$	MMD
Gluon	MP	0.8 ± 0.2	$\mathbf{1.0 \pm 0.3}$	$\mathbf{0.7 \pm 0.4}$	**0.11**	**0.54**	0.037
	TF	$\mathbf{0.7 \pm 0.1}$	1.2 ± 0.2	0.8 ± 0.6	**0.11**	**0.54**	**0.035**
Light Quark	MP	$\mathbf{0.7 \pm 0.2}$	5.0 ± 0.7	0.9 ± 0.6	0.33	0.51	**0.026**
	TF	0.8 ± 0.2	$\mathbf{1.6 \pm 0.4}$	$\mathbf{0.7 \pm 0.4}$	**0.11**	**0.54**	**0.026**
Top Quark	MP	$\mathbf{0.6 \pm 0.1}$	2.1 ± 0.5	1.5 ± 0.7	0.33	**0.59**	**0.071**
	TF	0.66 ± 0.09	$\mathbf{1.1 \pm 0.5}$	$\mathbf{1.4 \pm 0.6}$	**0.10**	0.57	**0.071**

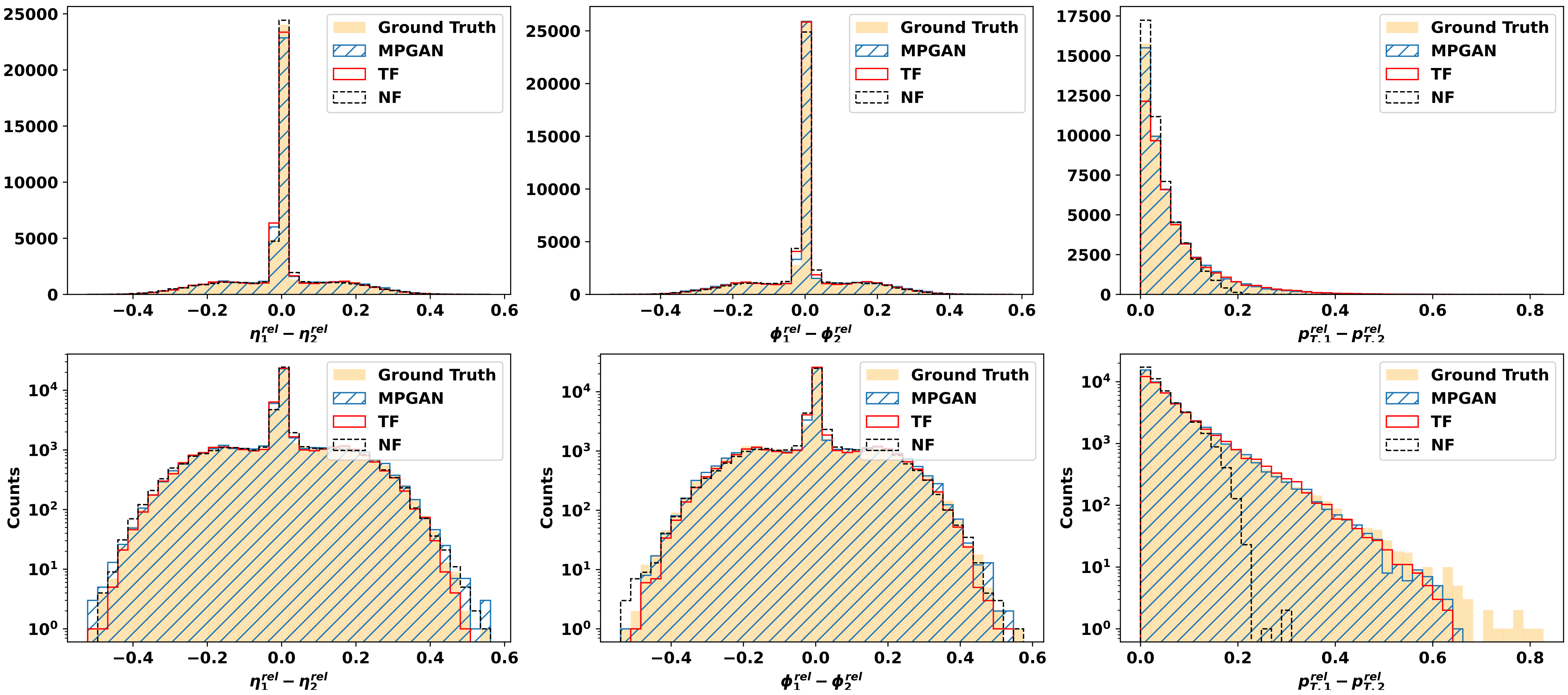
Summed Transverse Momentum

Credits to Erik Buhmann!

- Momentum given relative to jet $\rightarrow \sum_{i=1}^{n} p_T^{(i)} \leq 1$
- Directly supplying summed momentum to Critic $\rightarrow$ too strong discrimination
- Solution: Add noise, variance decreased gradually
- **Needs a lot of fine-tuning**

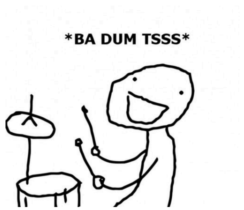


Top Quark Dataset Deltas Between Particles



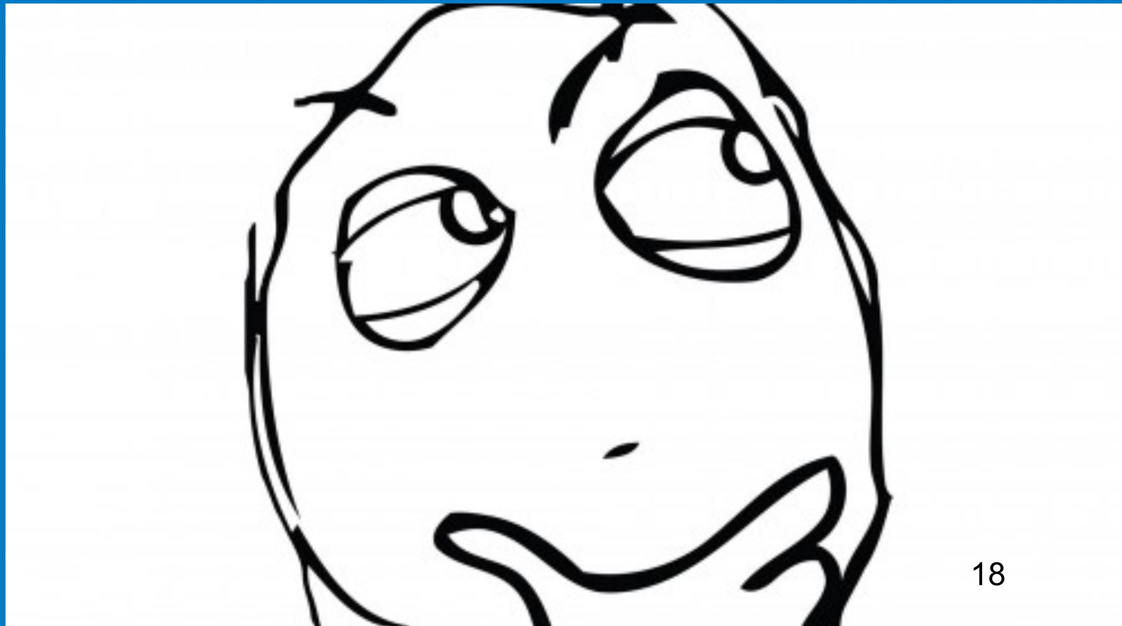
Summary

Thanks for your *attention*



- Normalising Flows quick & stable starting point for GAN
- Training duration $\sim 1-2\ h$ on $\texttt{NVIDIA P100}$
- Bad on high-level correlations between variables
- Transformer refinement enhances performance significantly
- $\sim 11.5\ \mu s$/jet, on $\texttt{NVIDIA P100}$ - $\sim 11.3\ \mu s$/jet, from NF
- Attention $\sim O(n^2)$, n number particles → How scalable?
- Transformers data hungry - introduce transfer learning?

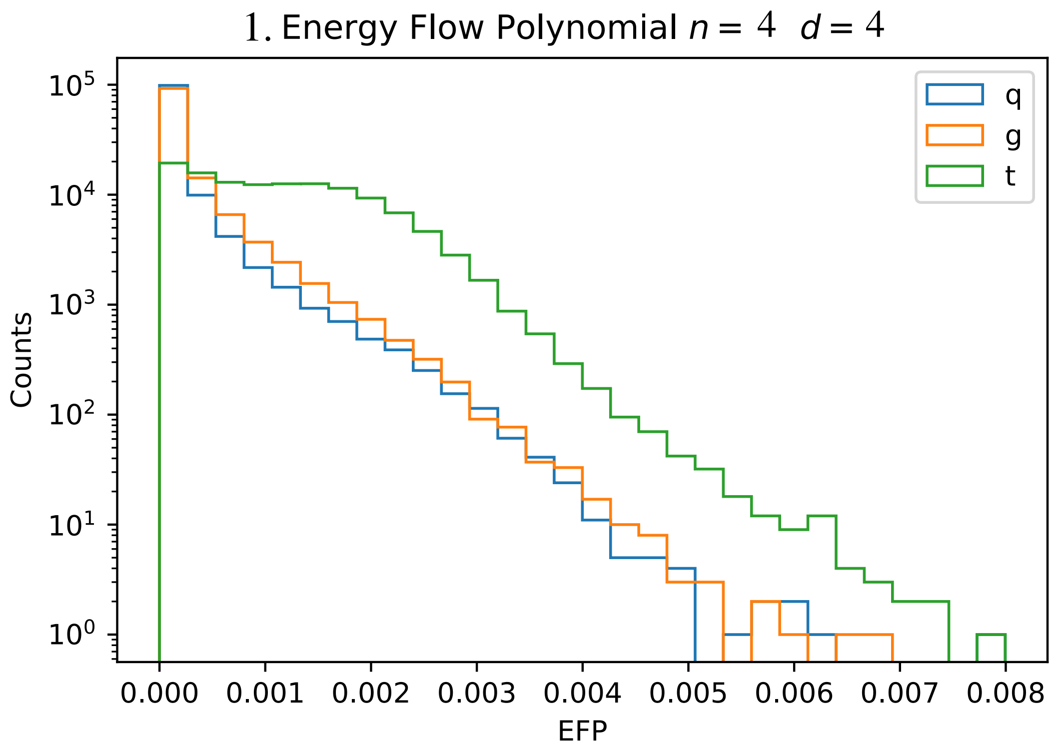
Any Questions?



Backup

Wasserstein Distance

- Metric on probability distributions
- Formally: $W_1(\mathbb{P}_r, \mathbb{P}_g) := \inf_{\gamma \in \Gamma(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\gamma}[\,|\boldsymbol{x}-\boldsymbol{y}|\,]$
- Not tractable for $\dim(\boldsymbol{X} \sim \mathbb{P}_g) > 1$
  - W_1^P: average of W_1 over (η, ϕ, p_T)
  - W_1^M: invariant jet mass
  - W_1^{EFP}: 5 Energy Flow Polynomials [4] (n=4,d=4)



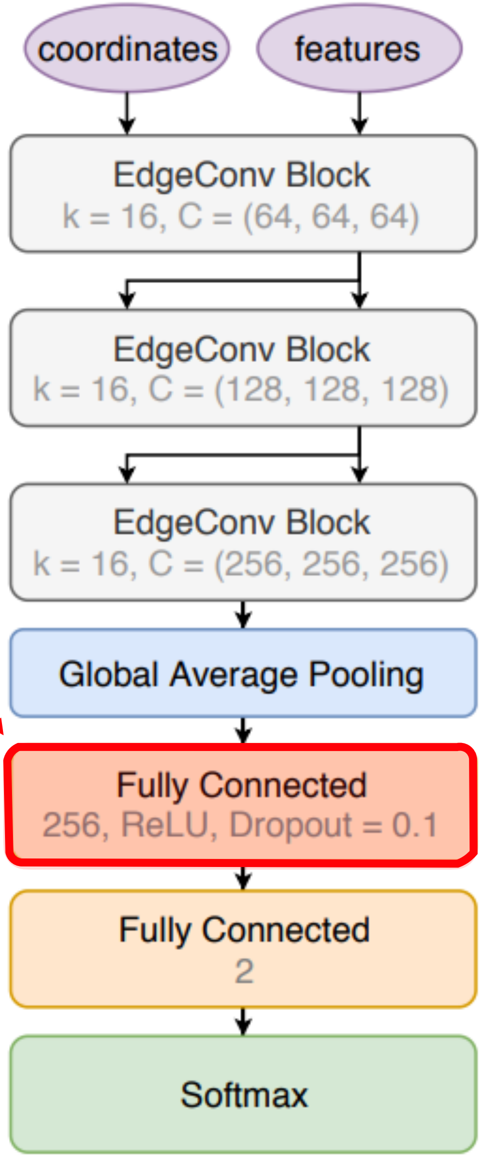
In-sample distances

Parton	$W_1^M(\times 10^{-3})$	$W_1^P(\times 10^{-3})$	$W_1^{EFP}(\times 10^{-5})$	FPND	COV $\uparrow$	MMD
Gluon	0.5 ± 0.1	0.4 ± 0.2	0.4 ± 0.4	0.01	0.56	0.036
Light Quark	0.42 ± 0.09	0.6 ± 0.4	0.5 ± 0.5	0.01	0.55	0.024
Top Quark	0.5 ± 0.1	0.6 ± 0.4	1.1 ± 0.4	0.03	0.56	0.072

[4] Komiske et al., Energy flow polynomials: A complete linear basis for jet sbstructure, arxiv.org/abs/1712.07124

Fréchet ParticleNet Distance (FPND) [2]

- Inspired from Fréchet Inception Distance (FID) for image generation [5]
- *Wasserstein-2 distance between Gaussians fitted to activations in first FC layer of ParticleNet [6] of MC & ML generated jets*
- Sensitive to output quality & mode collapse



In-sample distances

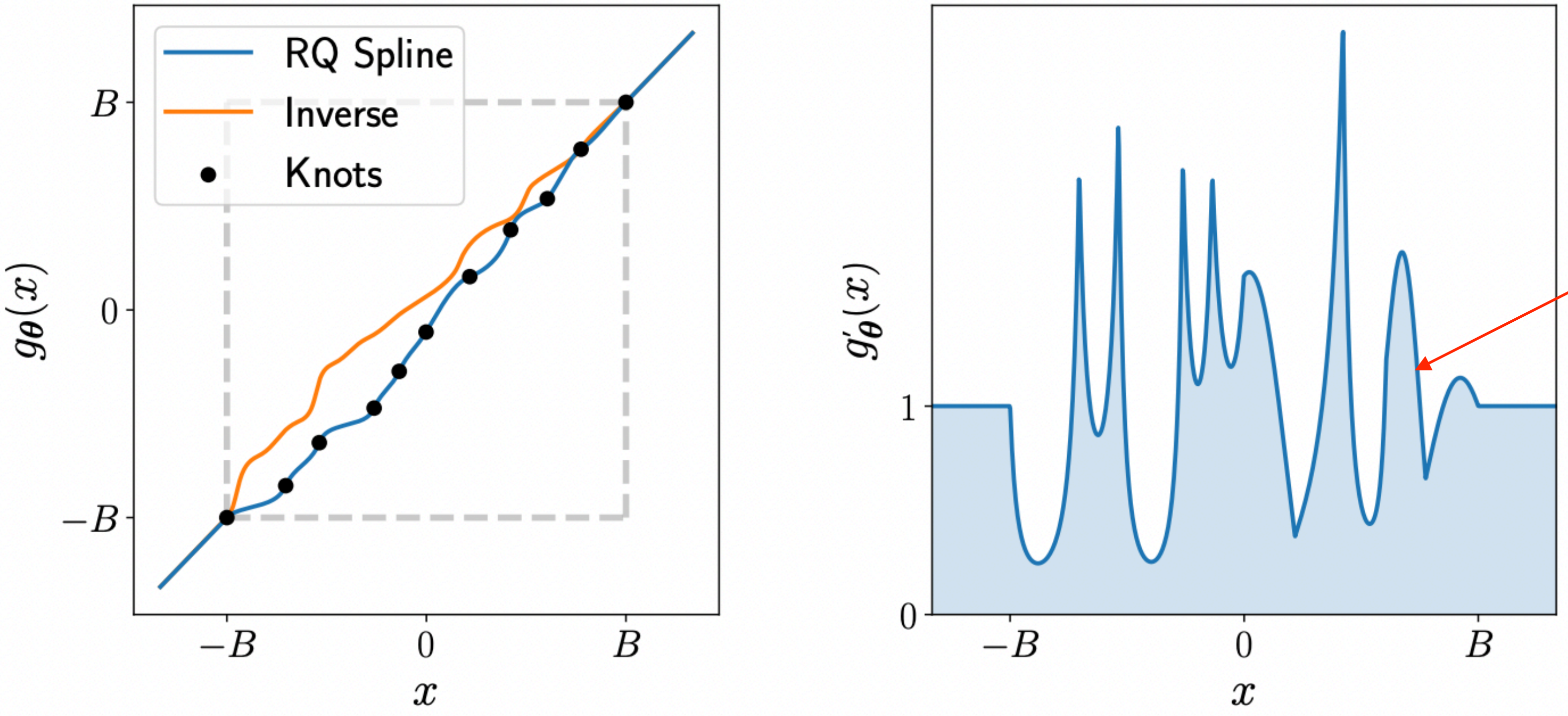
Parton	$W_1^M (\times 10^{-3})$	$W_1^P (\times 10^{-3})$	$W_1^{EFP} (\times 10^{-5})$	FPND	COV $\uparrow$	MMD
Gluon	0.5 ± 0.1	0.4 ± 0.2	0.4 ± 0.4	0.01	0.56	0.036
Light Quark	0.42 ± 0.09	0.6 ± 0.4	0.5 ± 0.5	0.01	0.55	0.024
Top Quark	0.5 ± 0.1	0.6 ± 0.4	1.1 ± 0.4	0.03	0.56	0.072

[2] Kansal et al., Particle Cloud Generation with Message Passing Generative Adversarial Networks, arxiv.org/abs/2106.11535
[5] Heusel et al., GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, arxiv.org/abs/1706.08500
[6] Qu et al., ParticleNet: Jet Tagging via Particle Clouds, arxiv.org/abs/1902.08570

Rational Quadratic Spline Coupling

Proposed by Durkan and Bekasov et al. [2], also in `nflows` [3]

- Affine coupling lacks flexibility
- Element-wise ratio of monotonic quadratic splines
- Monotonic → analytically invertible
- K bins → $(3K-1)$ NN outputs per dimension



$$p_X(\boldsymbol{x}) = p_Z(\boldsymbol{g_\theta}(\boldsymbol{x}))\left|\det \frac{d\boldsymbol{g_\theta}}{d\boldsymbol{x}}\right|$$

[2] Durkan and Bekasov et al., Neural Spline Flows, arxiv.org/abs/1906.04032.pdf
[3] Durkan and Bekasov et al., https://github.com/bayesiains/nflows

Handling Variable Number Particles

- Normalising Flows not optimal for variable number particles
- Transformers originating from NLP made to handle variable number inputs
- Attention allows interaction between variable number inputs
- For ground truth data straight forward → mask zero-padded particles
- Generation: sample masks from training data mask distribution

Coupling Layers

How to Construct Invertible Functions with a Tractable Determinant

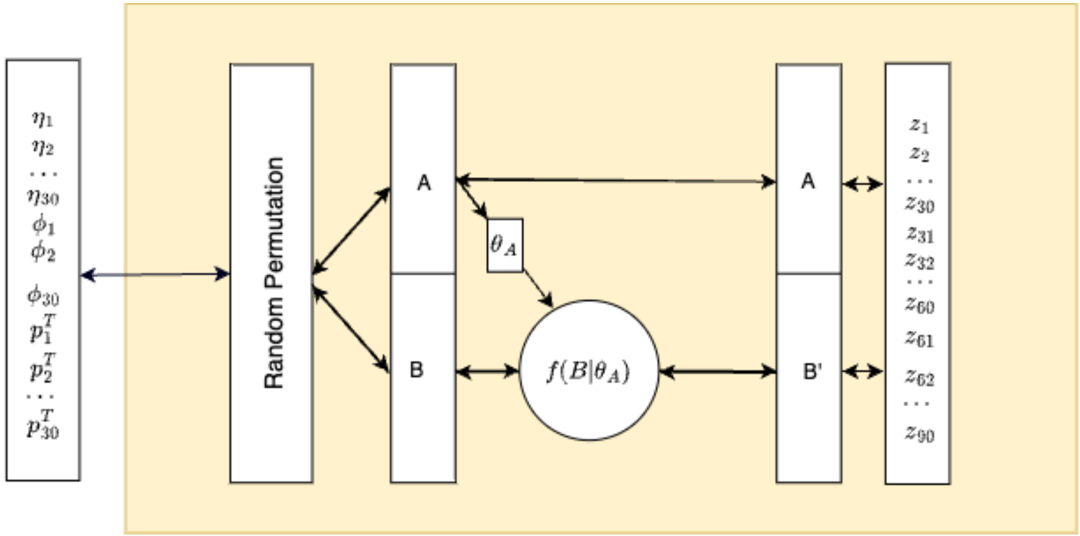
- Partition input into $(\mathbf{x}^A, \mathbf{x}^B) \in \mathbb{R}^d \times \mathbb{R}^{D-d}, D = 90,\ d = 45$

- Construct map element-wise: $\boldsymbol{f_\theta}(\boldsymbol{x}) = \begin{cases} z_i^A = x_i^A \\ z_i^B = s_{\boldsymbol{\theta}(x^A)}(x_i^B) \end{cases} \Leftrightarrow \boldsymbol{f_\theta}^{-1}(z) = \begin{cases} x_i^A = z_i^A \\ x_i^B = s^{-1}_{\boldsymbol{\theta}(z^A)}(z_i^B) \end{cases}$

$$\Rightarrow \frac{d\boldsymbol{f}}{d\boldsymbol{x}} = \begin{bmatrix} \mathbb{I} & 0 \\ \frac{dz_B}{d\boldsymbol{x}_A} & \frac{ds_{\theta(\boldsymbol{x}^A)}}{d\boldsymbol{x}^B} \end{bmatrix} \Rightarrow \det \frac{d\boldsymbol{f}}{d\boldsymbol{x}} = \prod_{i=d+1}^{D} \frac{ds_i^{\boldsymbol{\theta}(\boldsymbol{x}_A)}}{dx_i^B}$$

- $\boldsymbol{s_\theta}(\boldsymbol{x})$ simple parametrised function but $\boldsymbol{\theta}$ arbitrarily complex $\rightarrow$ **NN for parameters $\boldsymbol{\theta}$**

- Affine: $\boldsymbol{s}_{\boldsymbol{\theta}=(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}(\boldsymbol{x}) = \boldsymbol{x}_B \odot \boldsymbol{\theta}_1(\boldsymbol{x}_A) + \boldsymbol{\theta}_2(\boldsymbol{x}_A)$

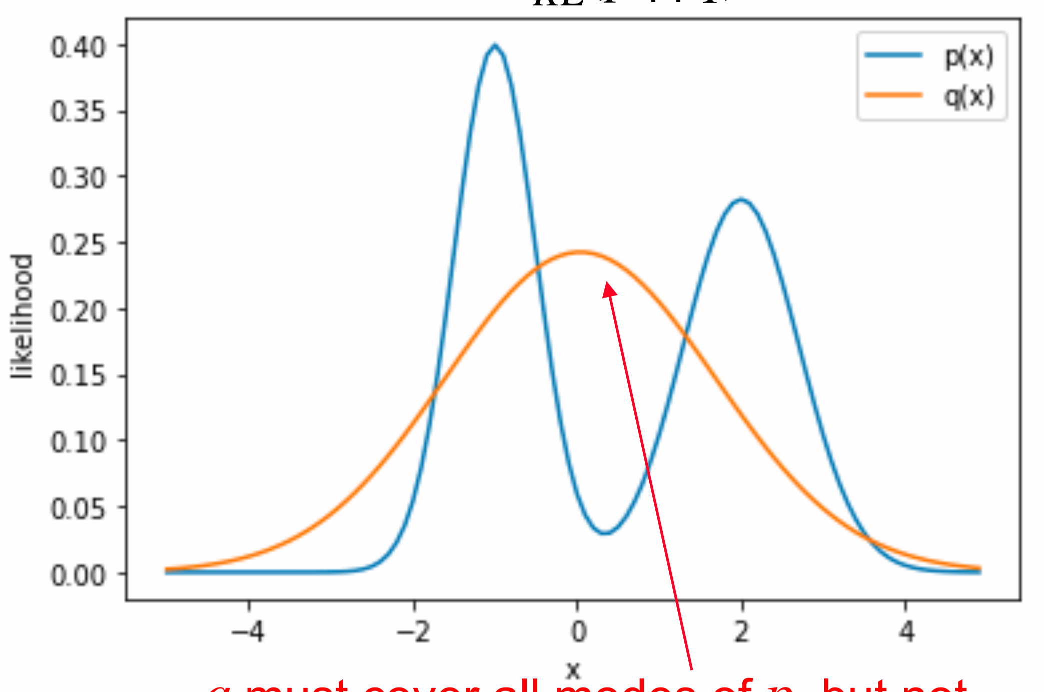


[6] Dinh et al., "Density estimation using Real NVP"

Possible Explanation why Max-Likelihood is not Enough

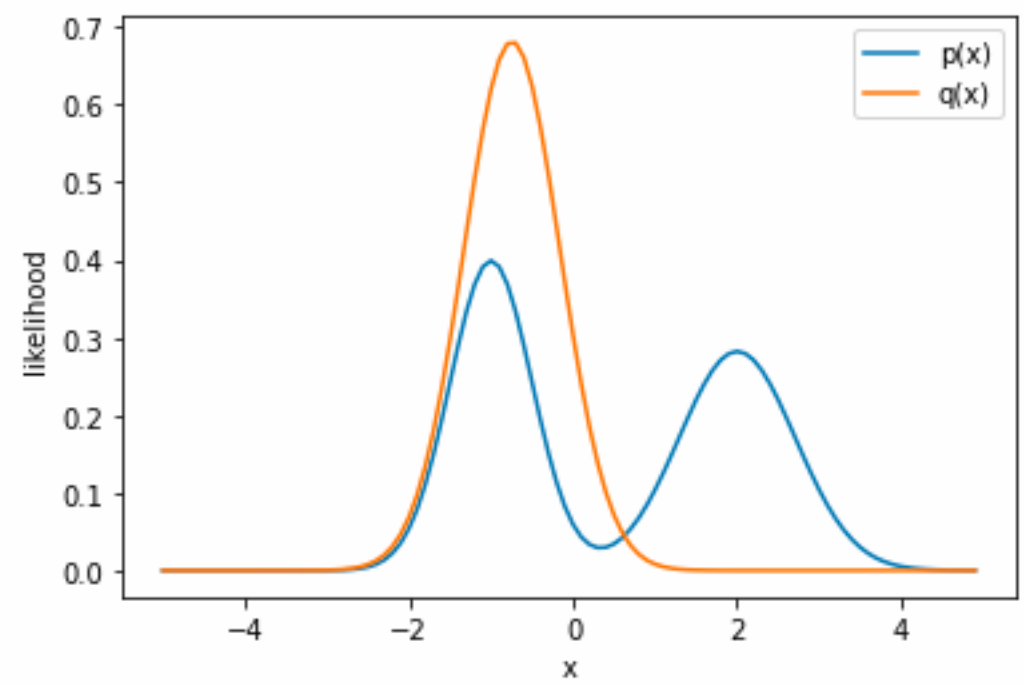
- Maximum likelihood consistent → can learn any distribution given **infinite** data & perfect model class
- Under model misspecification and finite data→ produces models that overgeneralise
- Minimising Forward KL-Divergence: equivalent to Maximum Likelihood

Forward $D_{KL}(p||q)$



q must cover all modes of p, but not penalised for having high q where p is low

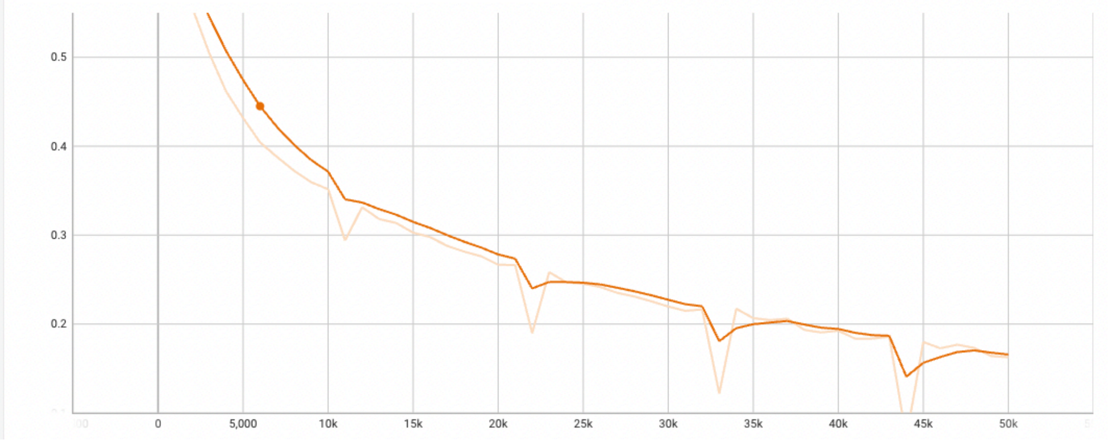
Backward $D_{KL}(q||p)$



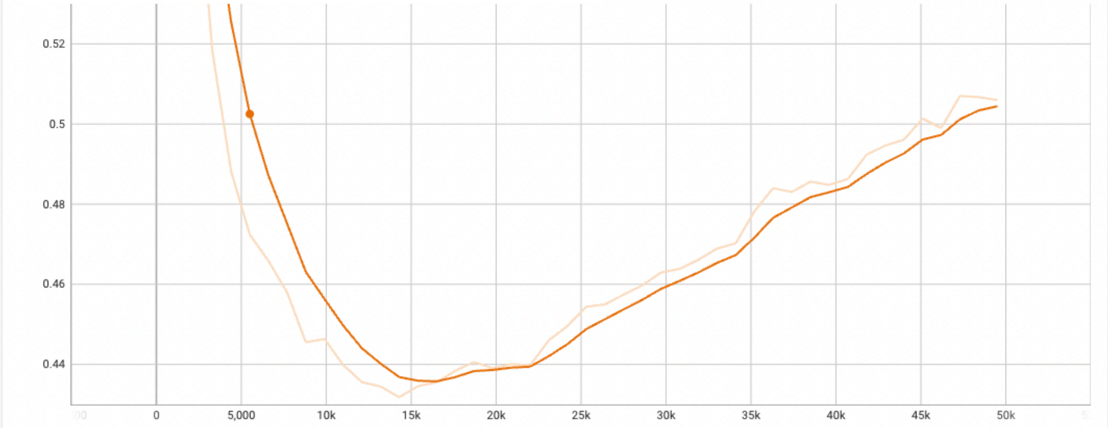
No punishment for mode collapse

Training vs Validation Logprobs, and Metrics

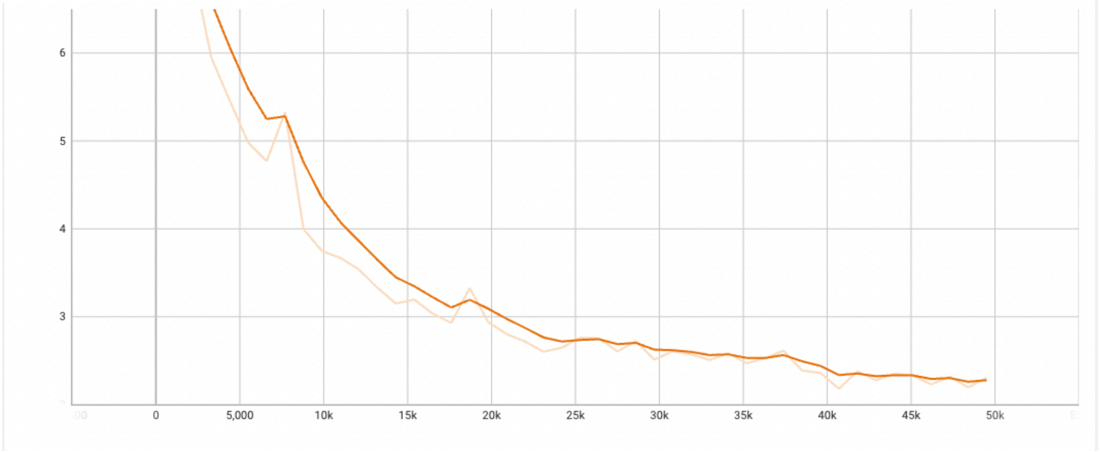
Training Logprob



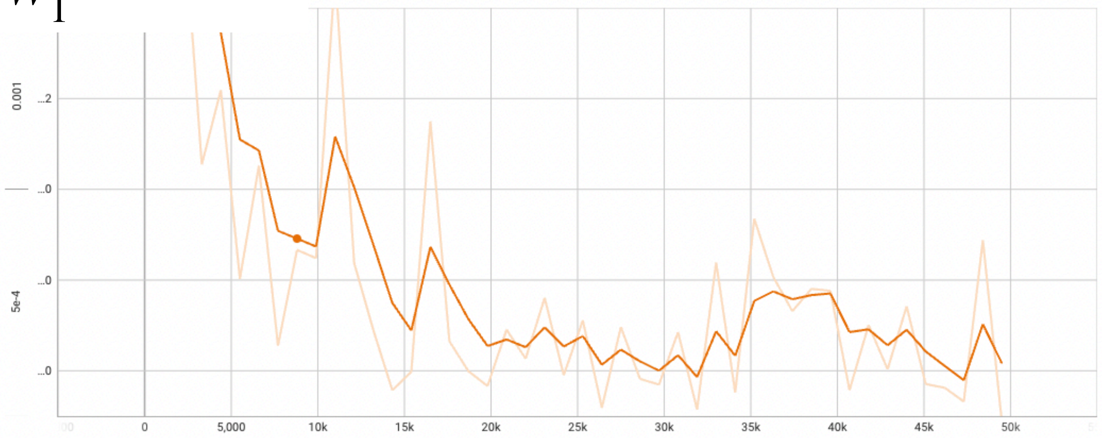
Validation Logprob



FPND

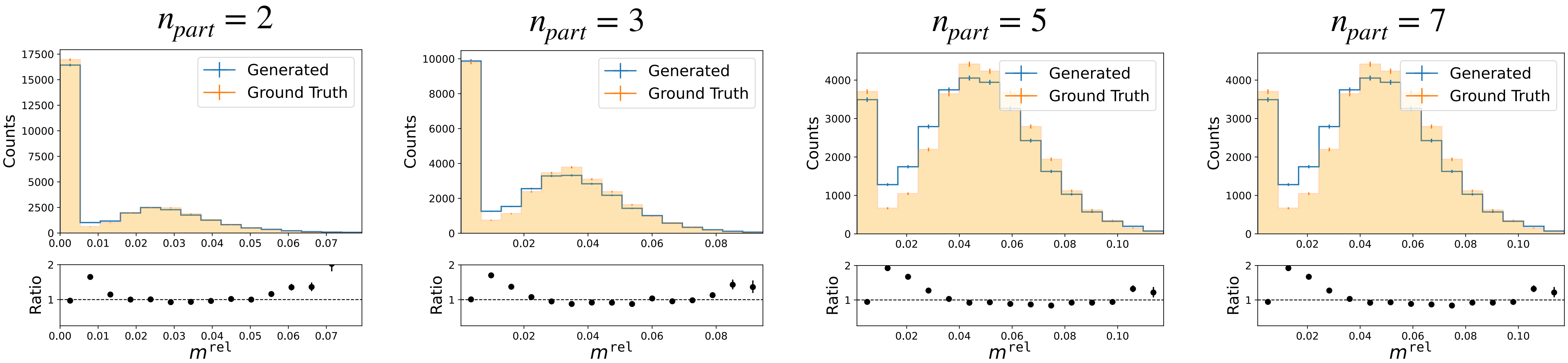


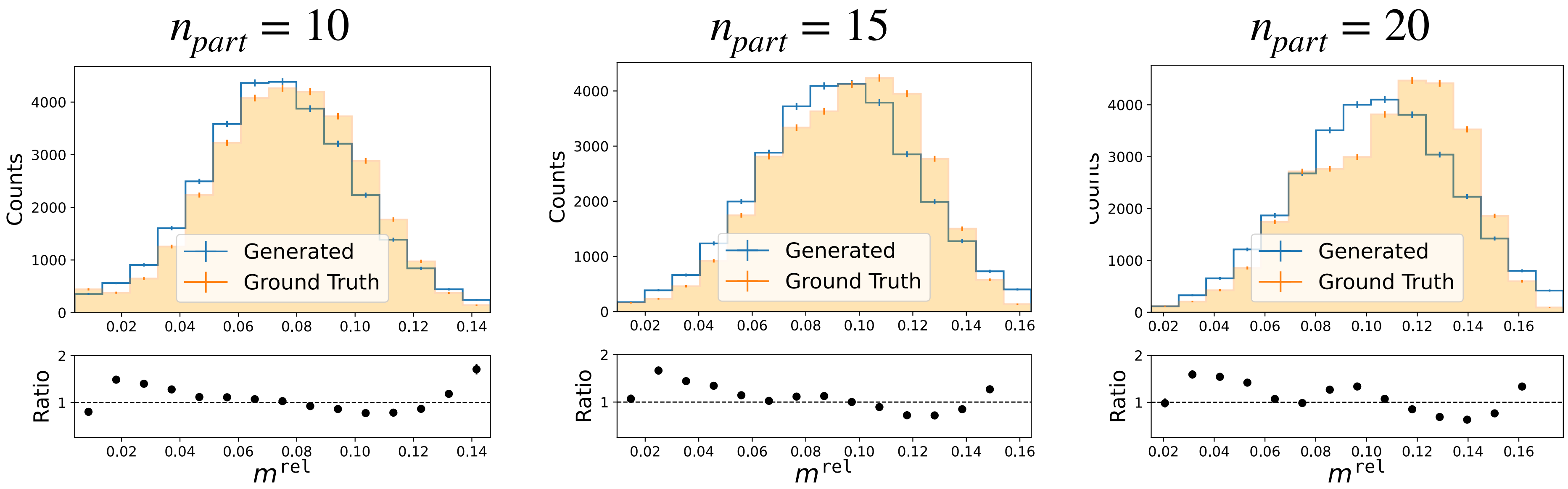
W_1^M



<u>Log Probability seems to not capture quality of generated data</u>

Dimensionality Scaling of Normalising Flows





In low dimensions Normalising Flow captures correlations correctly

Normalising Flows

In more formal language

- Main foundation: Change of Variables formula, $z = \boldsymbol{f_\theta}(\boldsymbol{x})$

$$p_X(\boldsymbol{x}) = p_Z(\boldsymbol{f_\theta}(\boldsymbol{x}))\left|\det\frac{d\boldsymbol{f_\theta}}{d\boldsymbol{x}}\right| = p_{\boldsymbol{Z}}(z)\left|\det\frac{d\boldsymbol{f_\theta}^{-1}}{dz}\right|^{-1}$$

  - 2 Constraints: Invertible functions, Jacobi-Matrix tractable

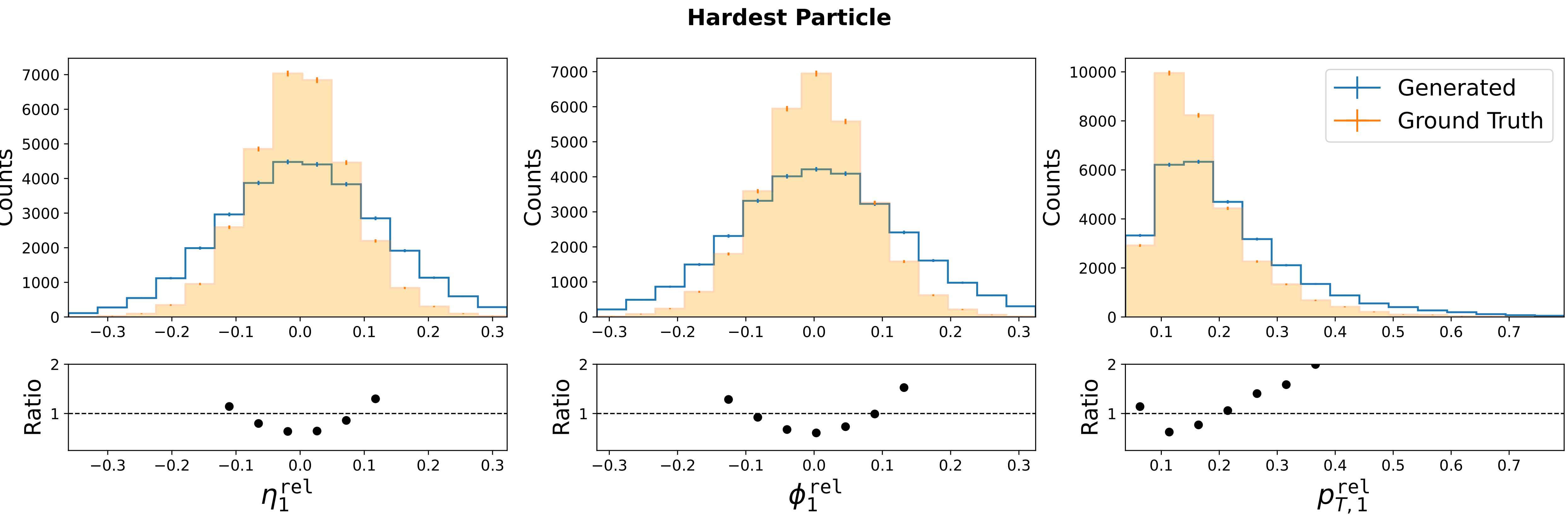
- Stack transformations: $z = z_K = \boldsymbol{f}^{(K)} \circ \cdots \circ \boldsymbol{f}^{(0)}(z_0 = \boldsymbol{x})$

  $\rightarrow$ Invertible with determinant $\prod_{i=0}^{K} \det\left|\frac{d\boldsymbol{f_\theta}^{(i)}}{d\mathbf{x}_i}\right|$

- **Optimise with negative Log-Likelihood:**

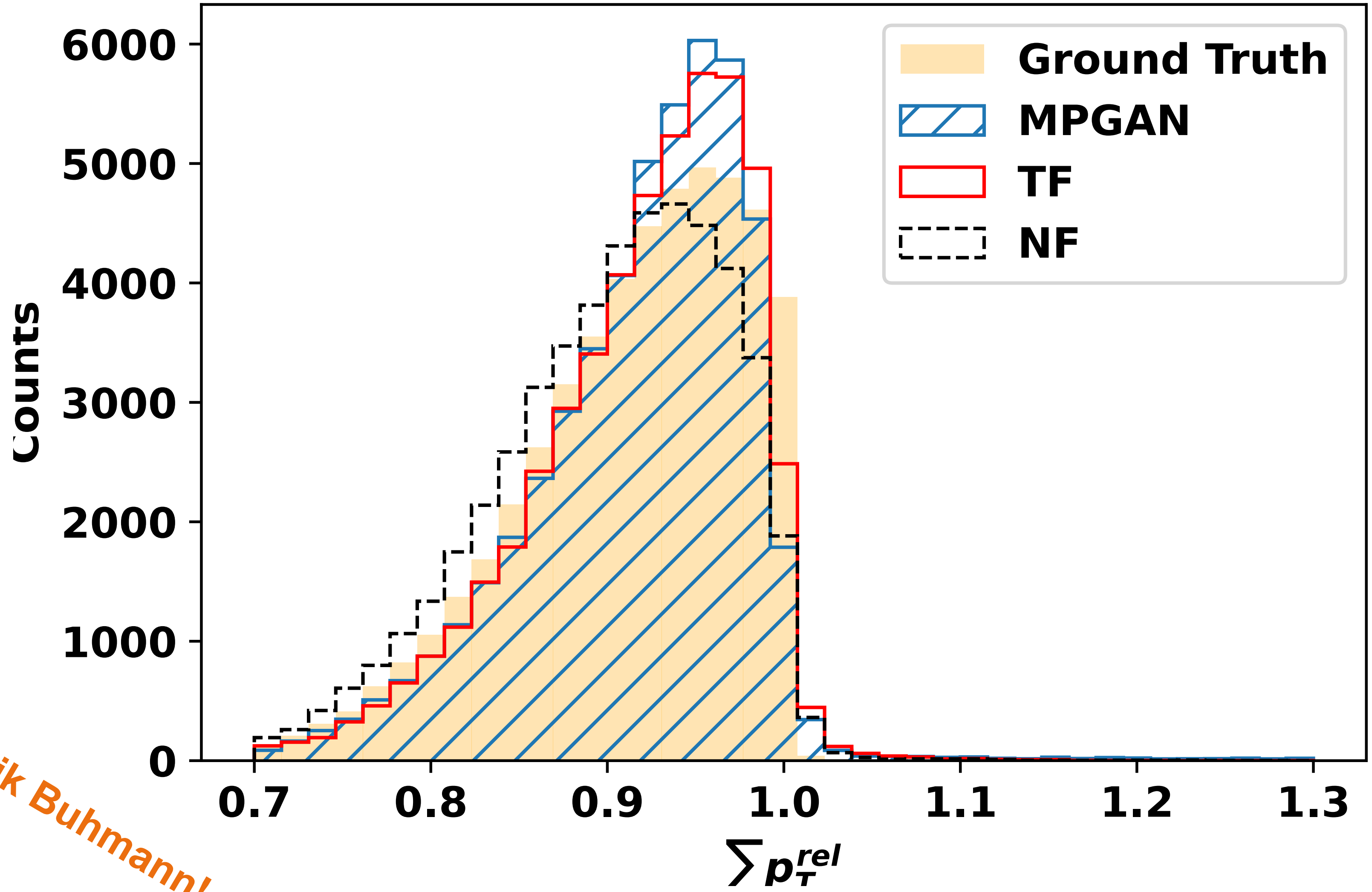
$$\boldsymbol{\theta} = -\arg\min_{\boldsymbol{\theta}} \sum_{x\in X} \log p_{\boldsymbol{x}}(\boldsymbol{x}) = \arg\min_{\boldsymbol{\theta}} \sum_{x\in X}\left(\frac{\boldsymbol{f}(\boldsymbol{x})^{\mathbf{2}}}{\mathbf{2}} - \sum_{i=0}^{K}\left|\det\frac{d\boldsymbol{f}_{\boldsymbol{\theta}}^{(i)}}{d\boldsymbol{x}_i}\right|\right)$$

Affine Marginals



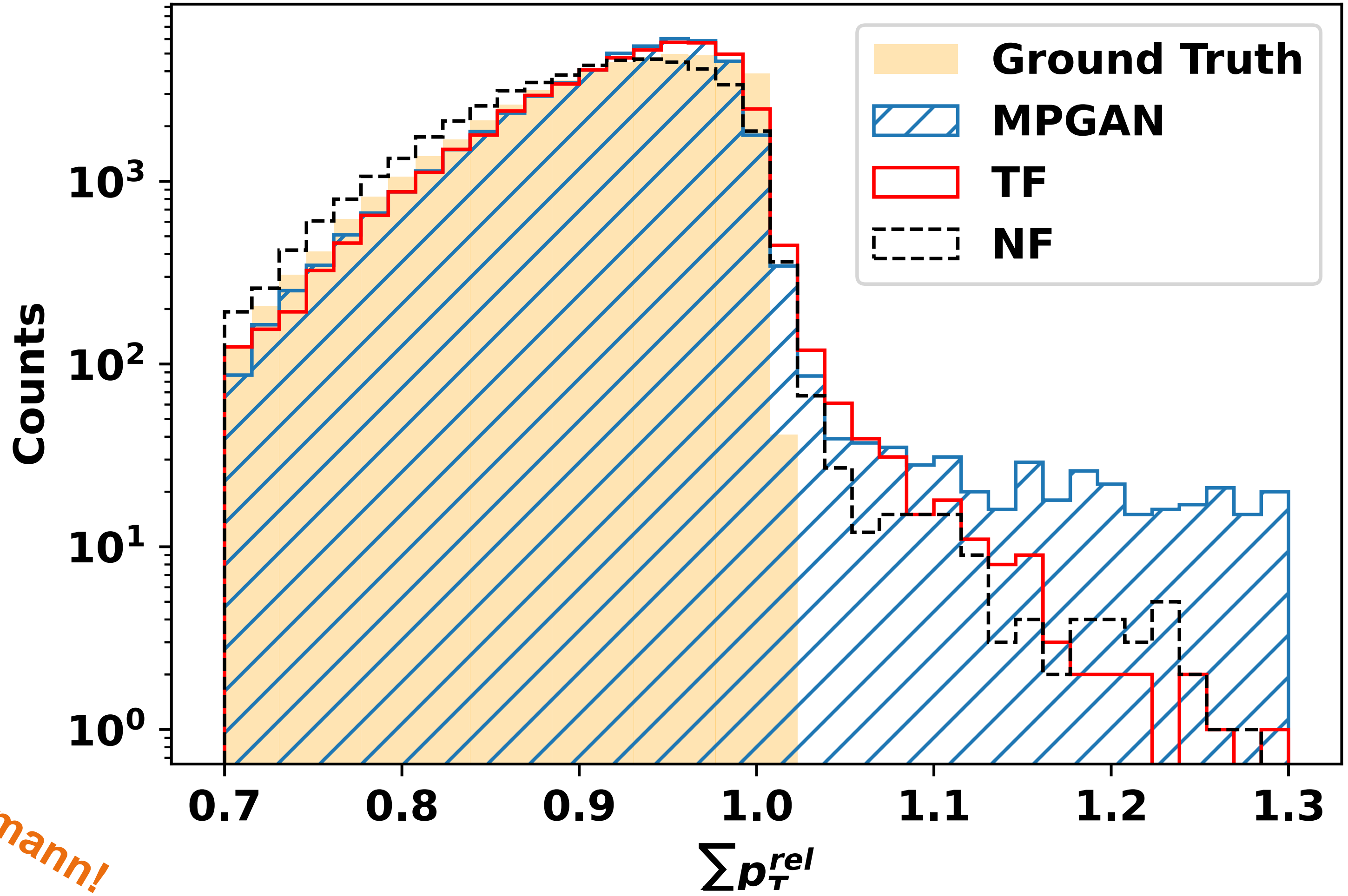
Summed of Relative Transverse Momenta (Top Quark)

Summed Transverse Momenta used as Input to FC Layers in Critic



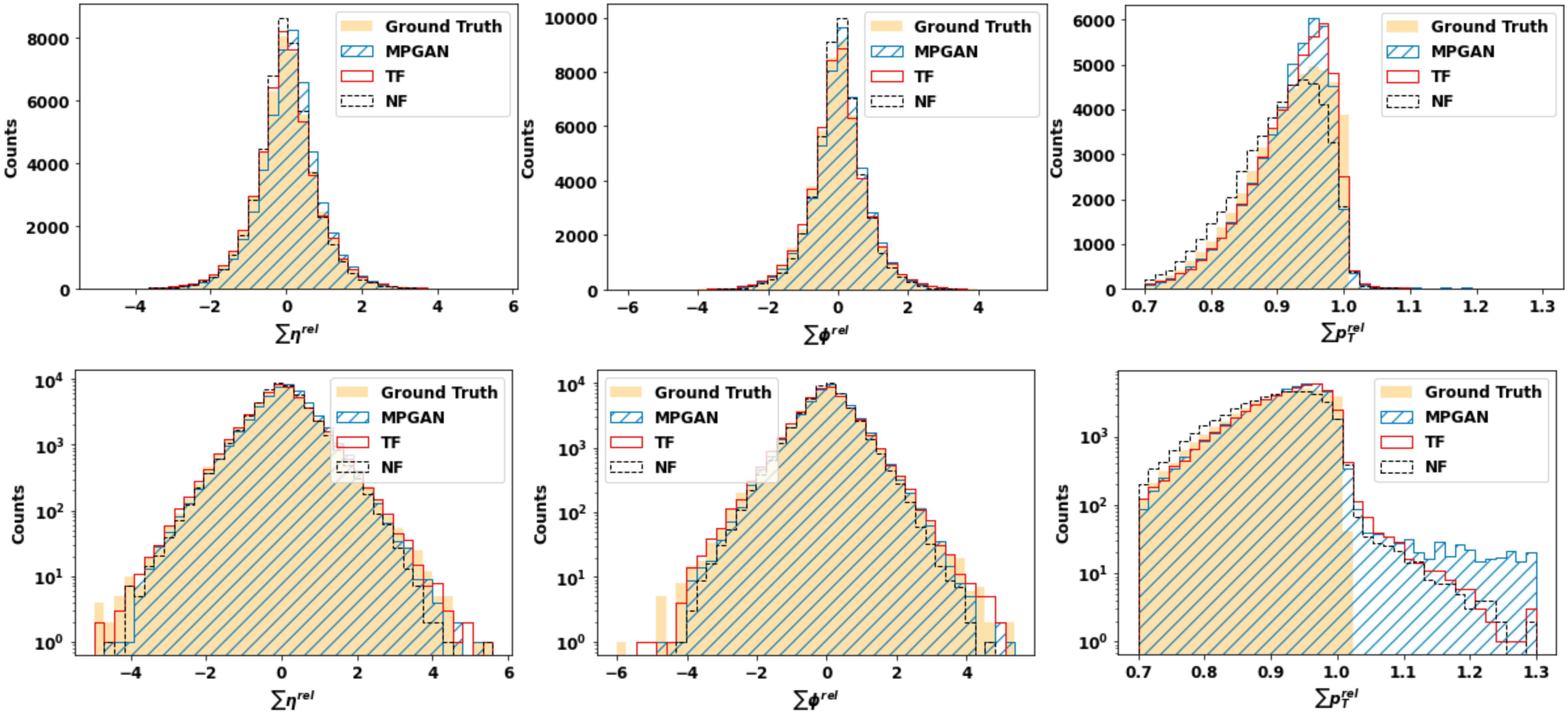
Credits to Erik Buhmann!

Summed of Relative Transverse Momenta (Top Quark)

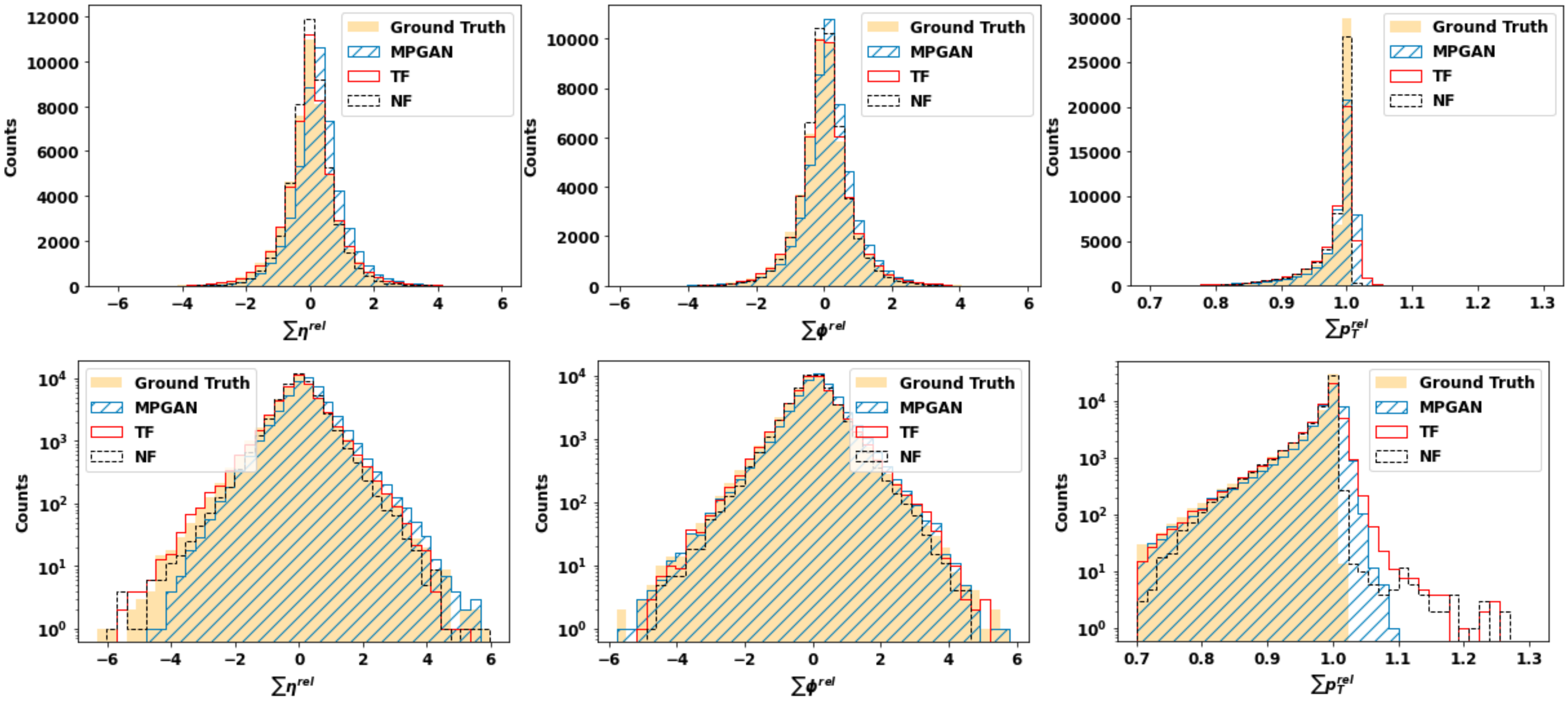


Credits to Erik Buhmann!

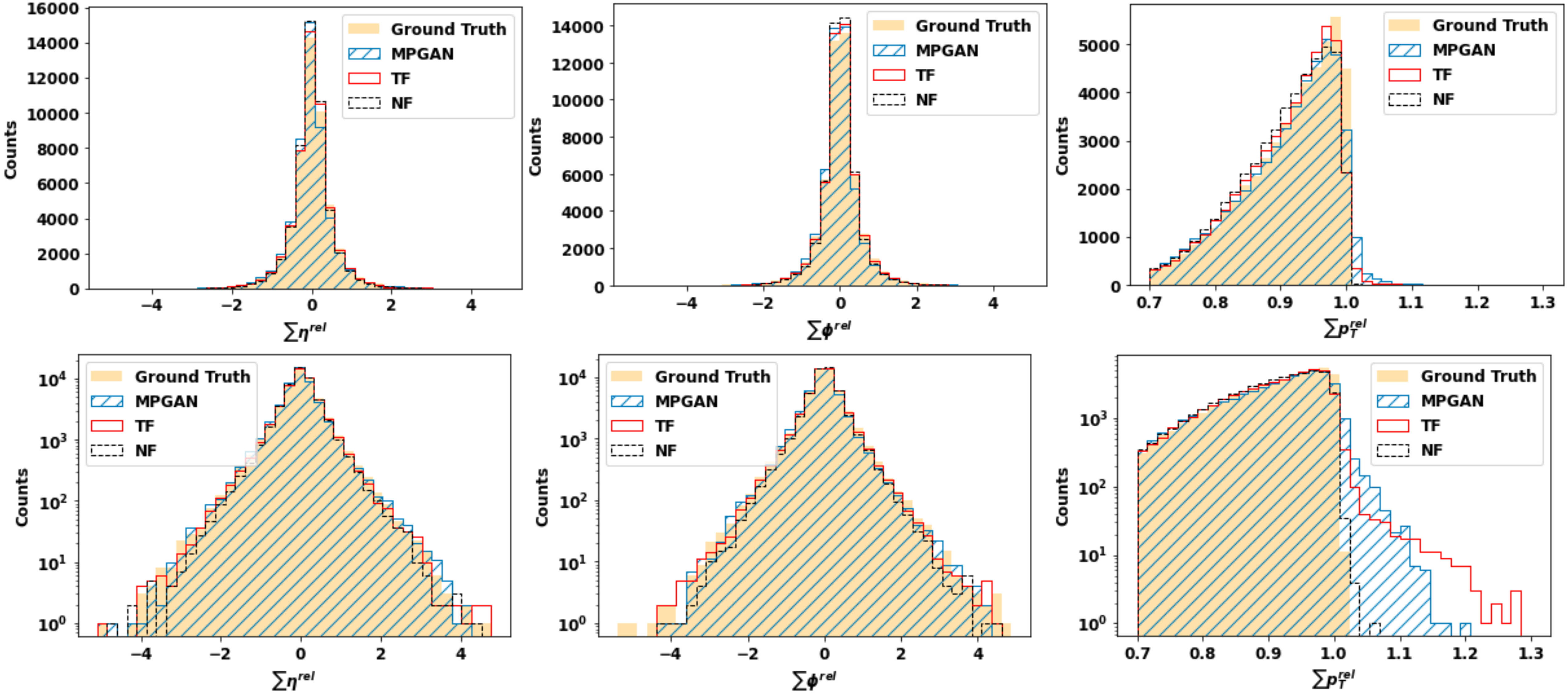
Top Quark Dataset Sums



Light Quark Dataset Sums



Gluon Dataset Sums



Normalising Flows are BIG

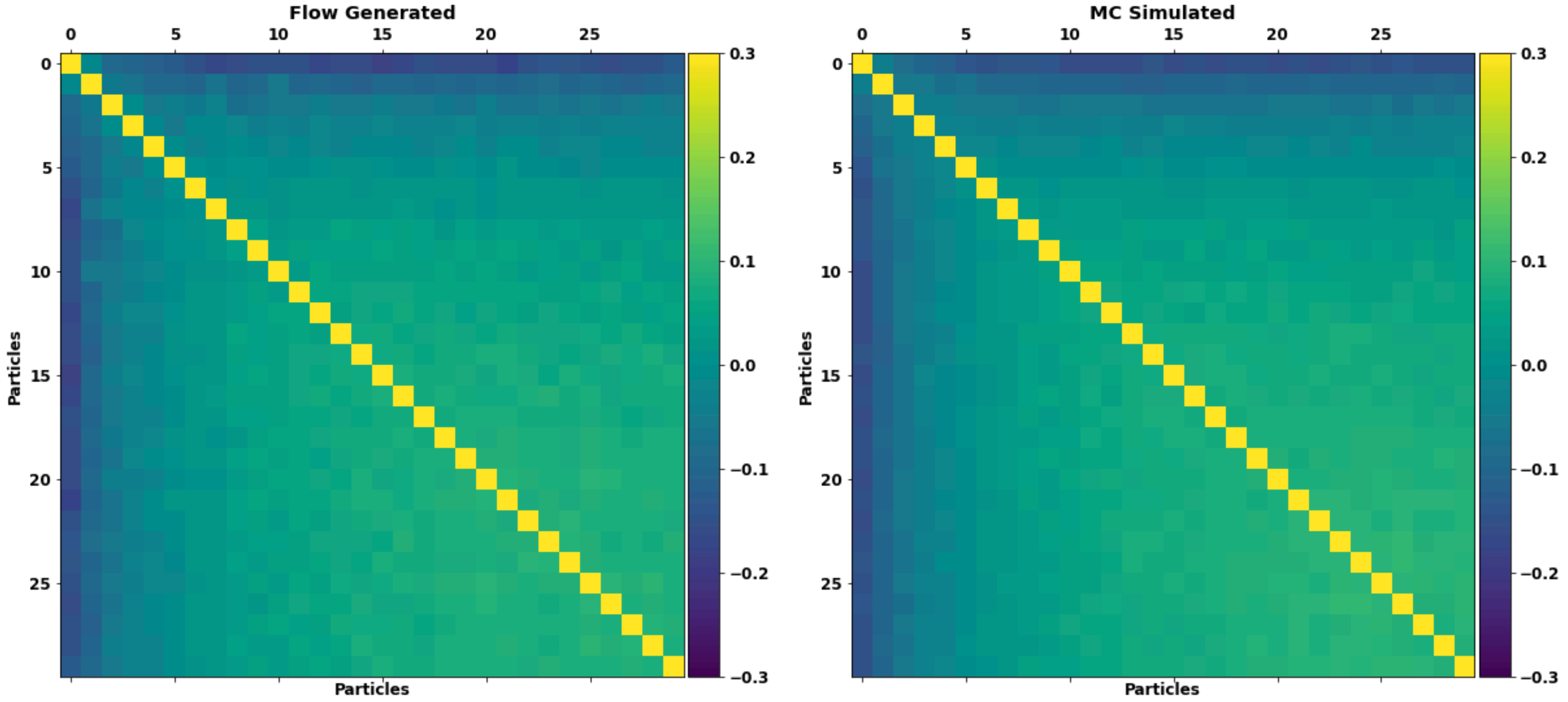
$\sim 92\,\%$ of Parameters are NF

```
TransformerGan                                               --
├─Flow: 1-1                                                  --
│    └─CompositeTransform: 2-1                               --
│    │    └─ModuleList: 3-1                                  6,255,592
│    └─StandardNormal: 2-2                                   --
│    └─Identity: 2-3                                         --
├─Gen: 1-2                                                   --
│    └─Linear: 2-4                                           128
│    └─TransformerEncoder: 2-5                               --
│    │    └─ModuleList: 3-2                                  147,168
│    └─Linear: 2-6                                           8,448
│    └─Linear: 2-7                                           65,792
│    └─Dropout: 2-8                                          --
│    └─Linear: 2-9                                           771
│    └─Linear: 2-10                                          99
├─Disc: 1-3                                                  --
│    └─Linear: 2-11                                          128
│    └─TransformerEncoder: 2-12                              --
│    │    └─ModuleList: 3-3                                  147,168
│    └─TransformerEncoderLayer: 2-13                         --
│    │    └─MultiheadAttention: 3-4                          4,224
│    │    └─Linear: 3-5                                      8,448
│    │    └─Dropout: 3-6                                     --
│    │    └─Linear: 3-7                                      8,224
│    │    └─LayerNorm: 3-8                                   64
│    │    └─LayerNorm: 3-9                                   64
│    │    └─Dropout: 3-10                                    --
│    │    └─Dropout: 3-11                                    --
│    └─Linear: 2-14                                          64
│    └─Linear: 2-15                                          16,896
│    └─Linear: 2-16                                          131,328
│    └─Linear: 2-17                                          257
├─Sigmoid: 1-4                                               --
Total params: 6,794,863
Trainable params: 6,794,863
Non-trainable params: 0
```

- Model has **6,794,863** trainable parameters
- Transformer Refinement $\sim 4\,\%$ total parameters

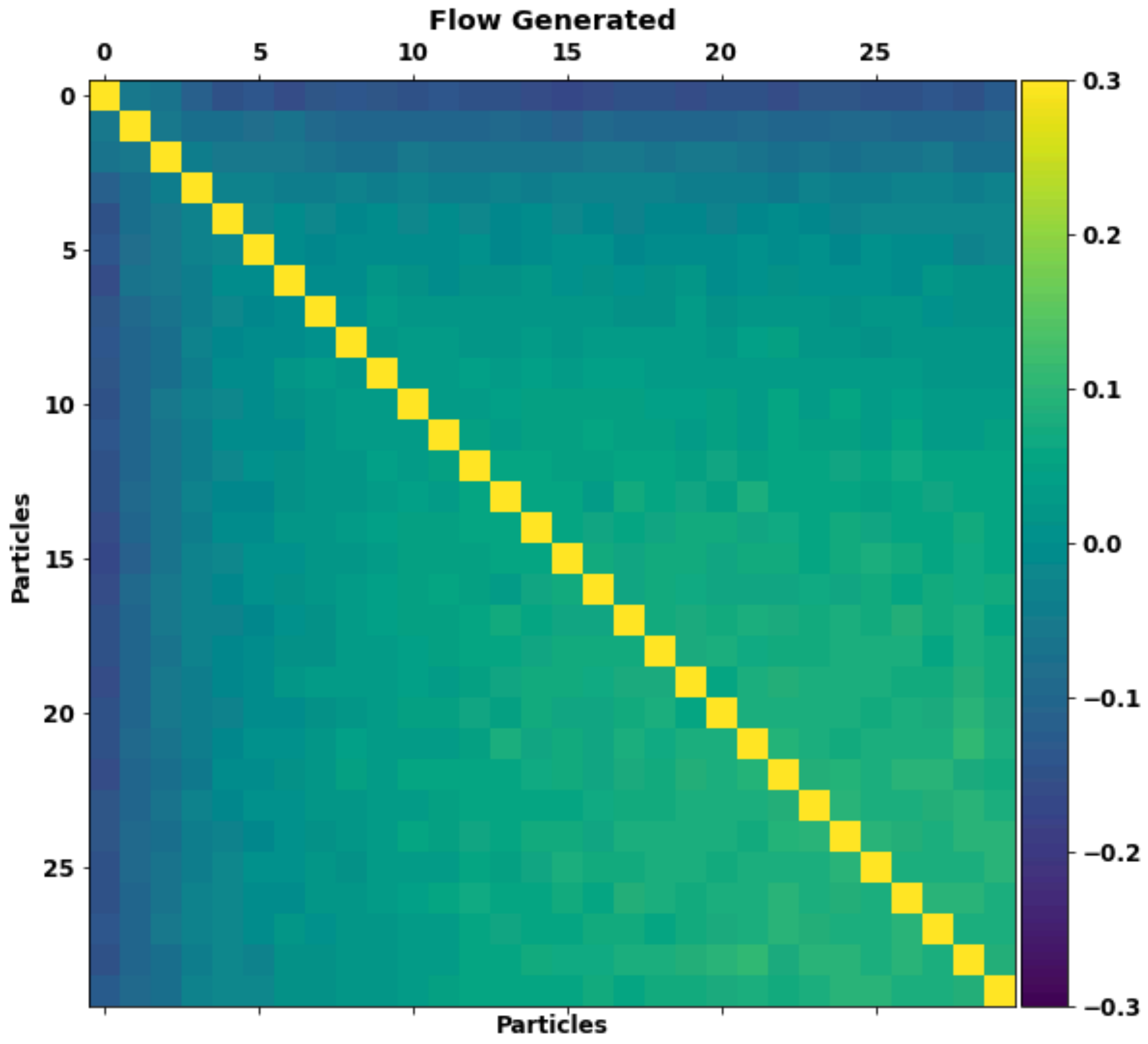
Correlation Plots

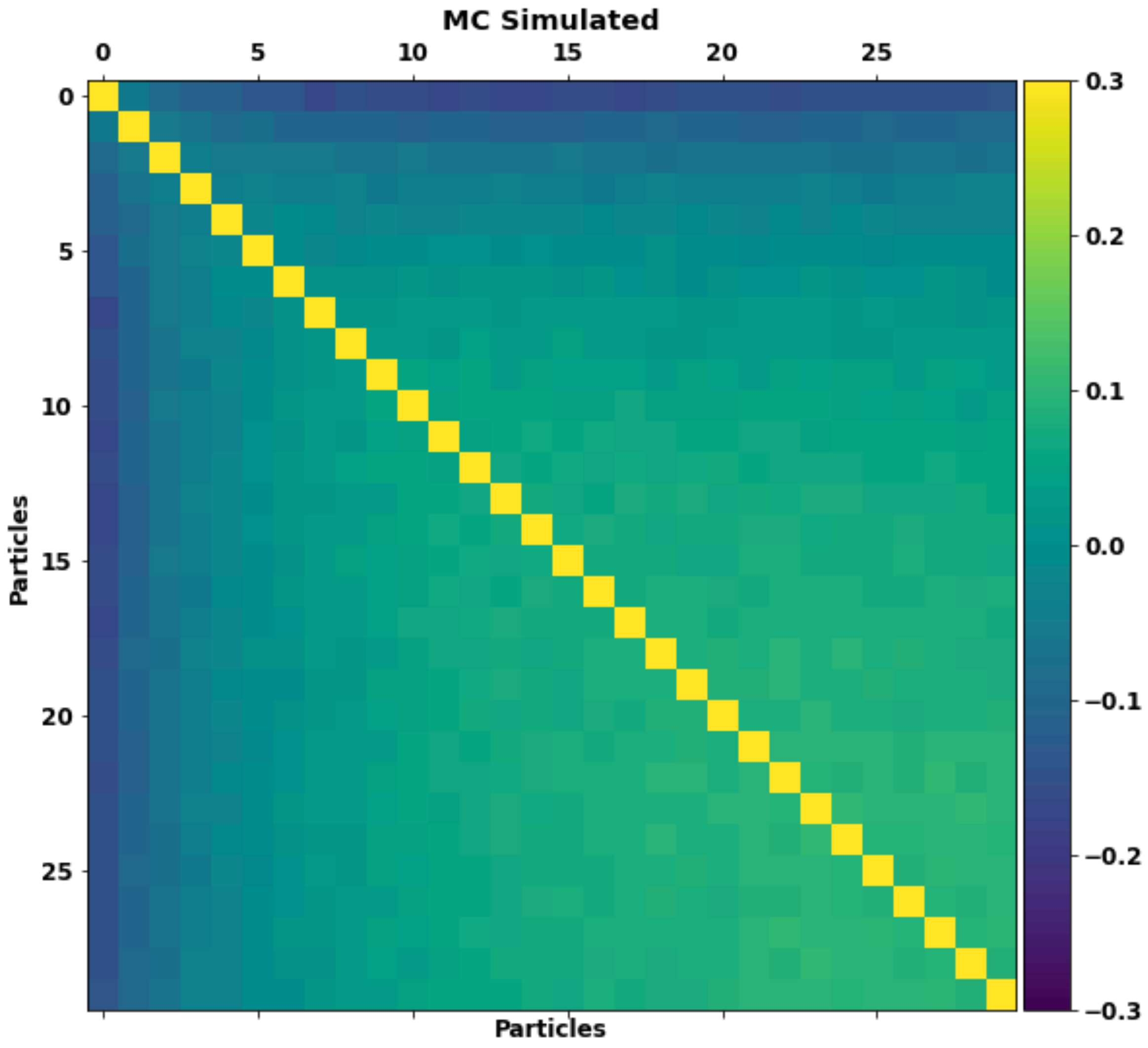
η_{rel}



Correlation Plots

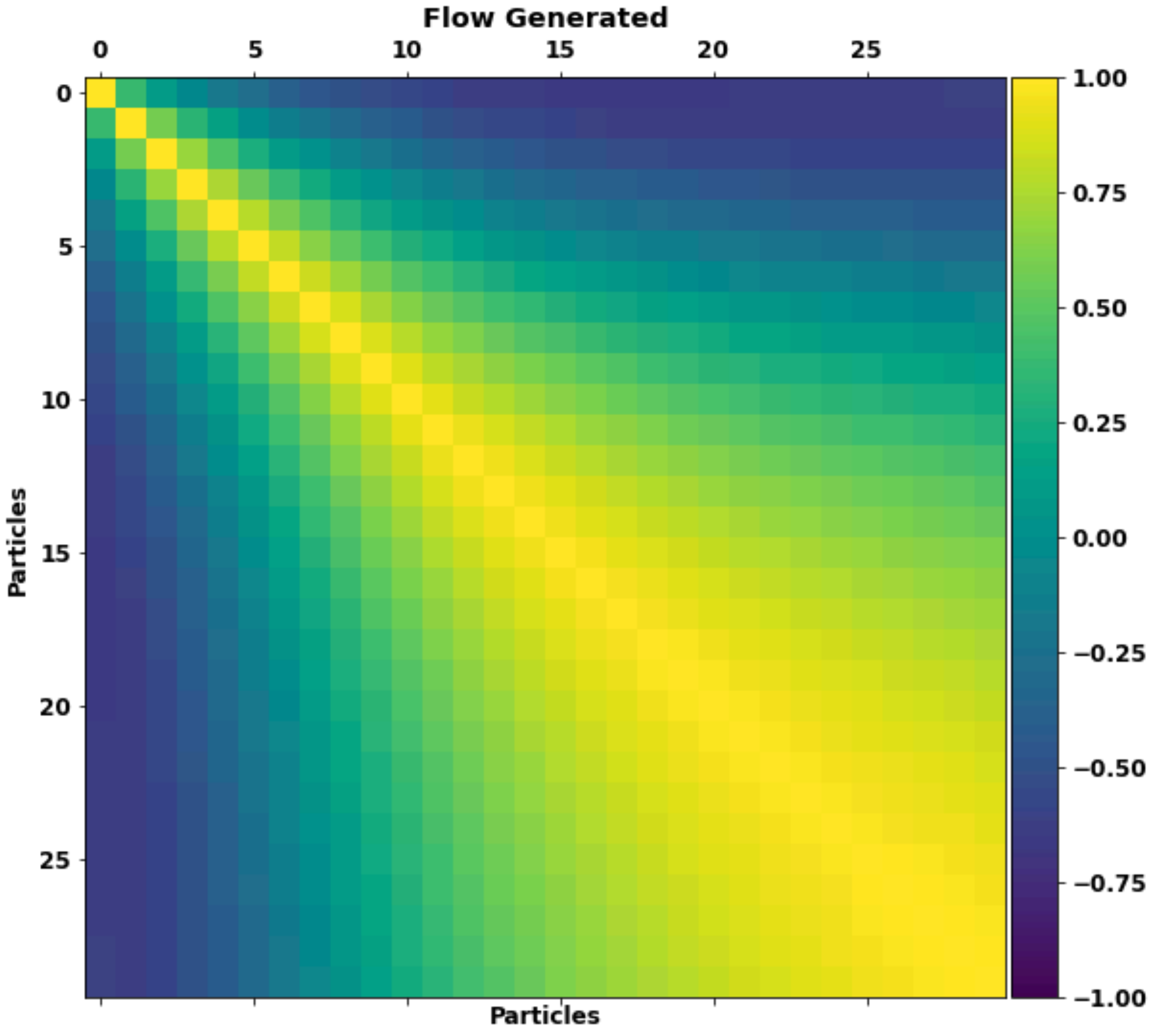
ϕ_{rel}

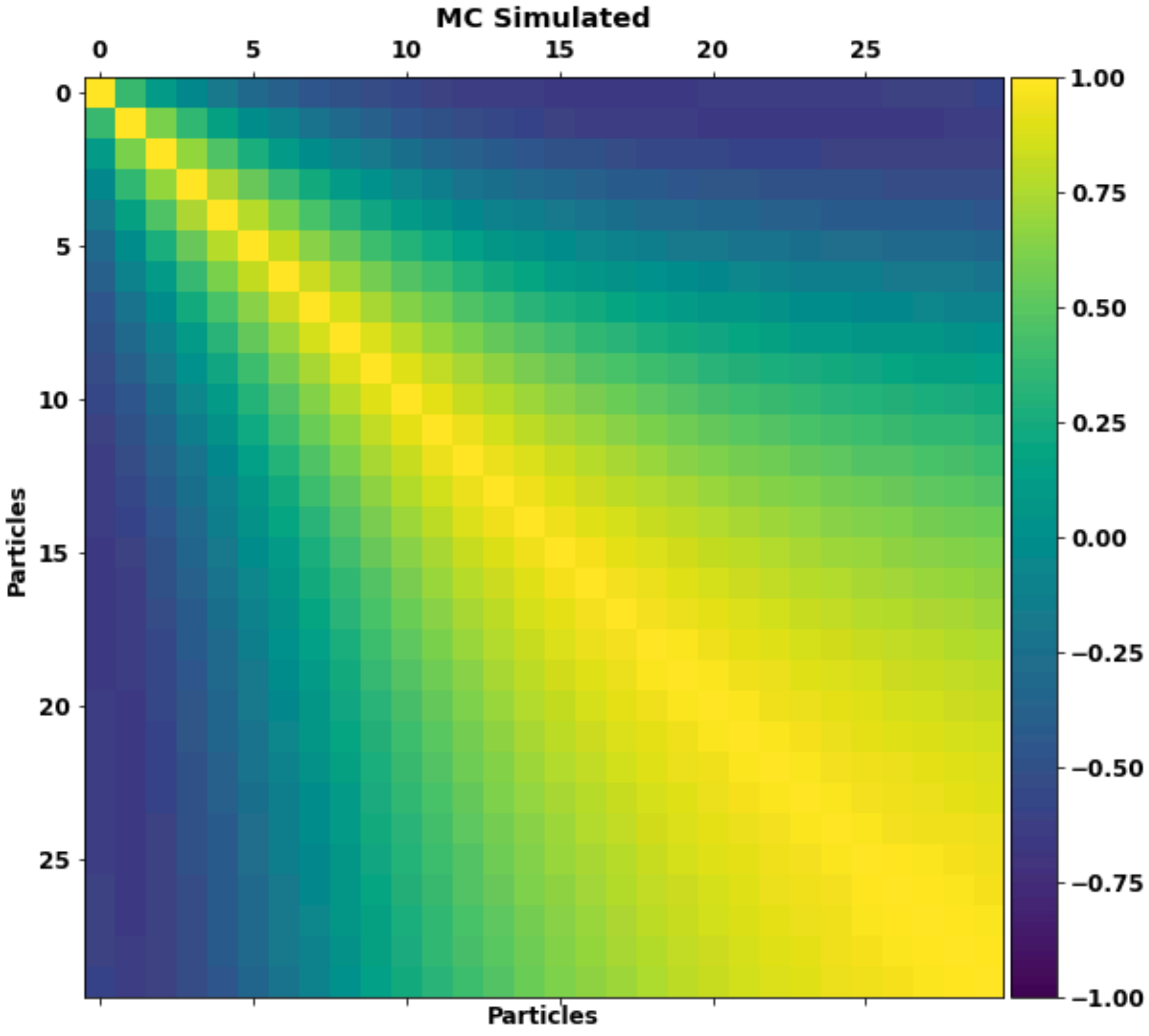




Correlation Plots

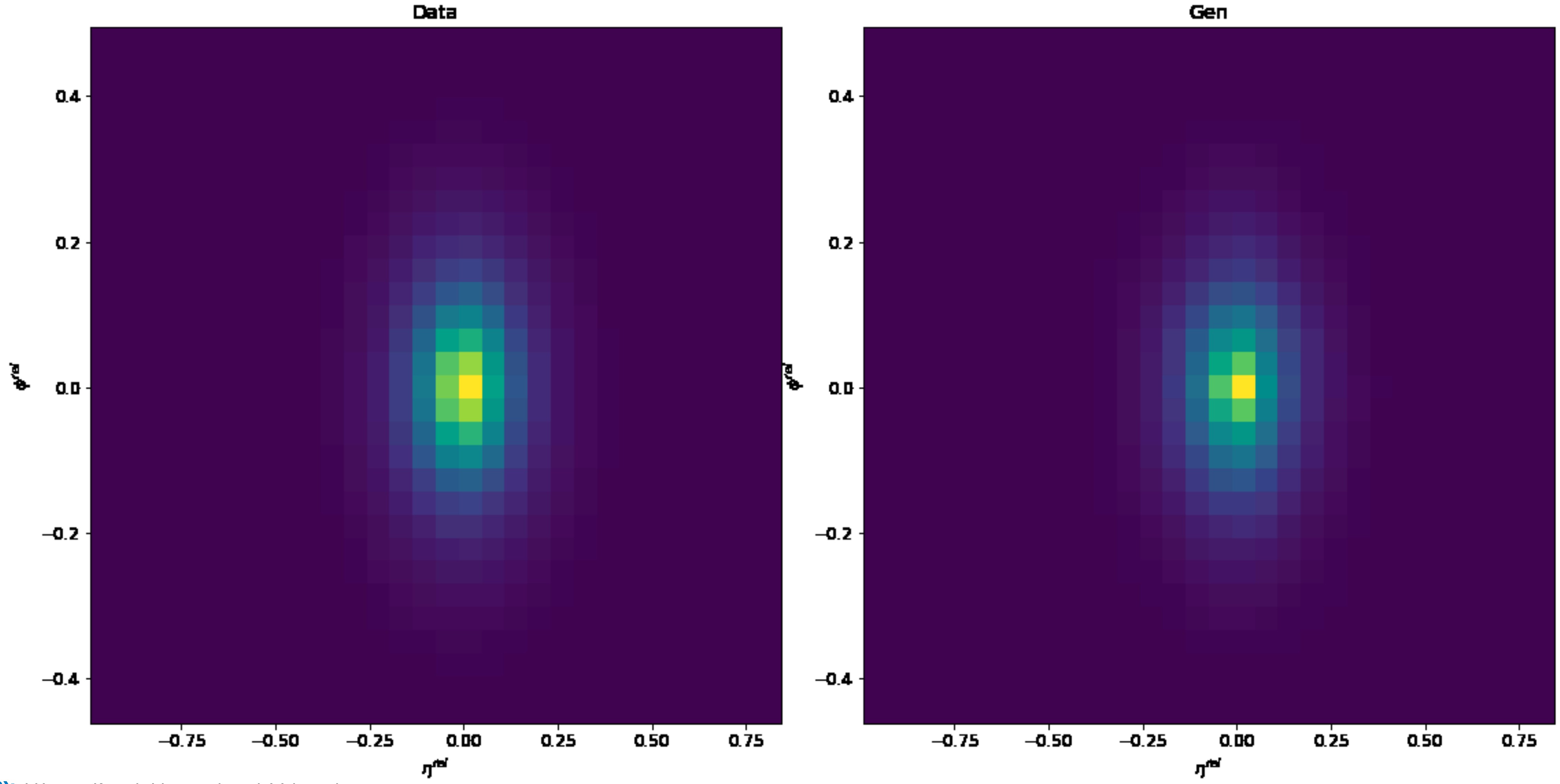
p_T





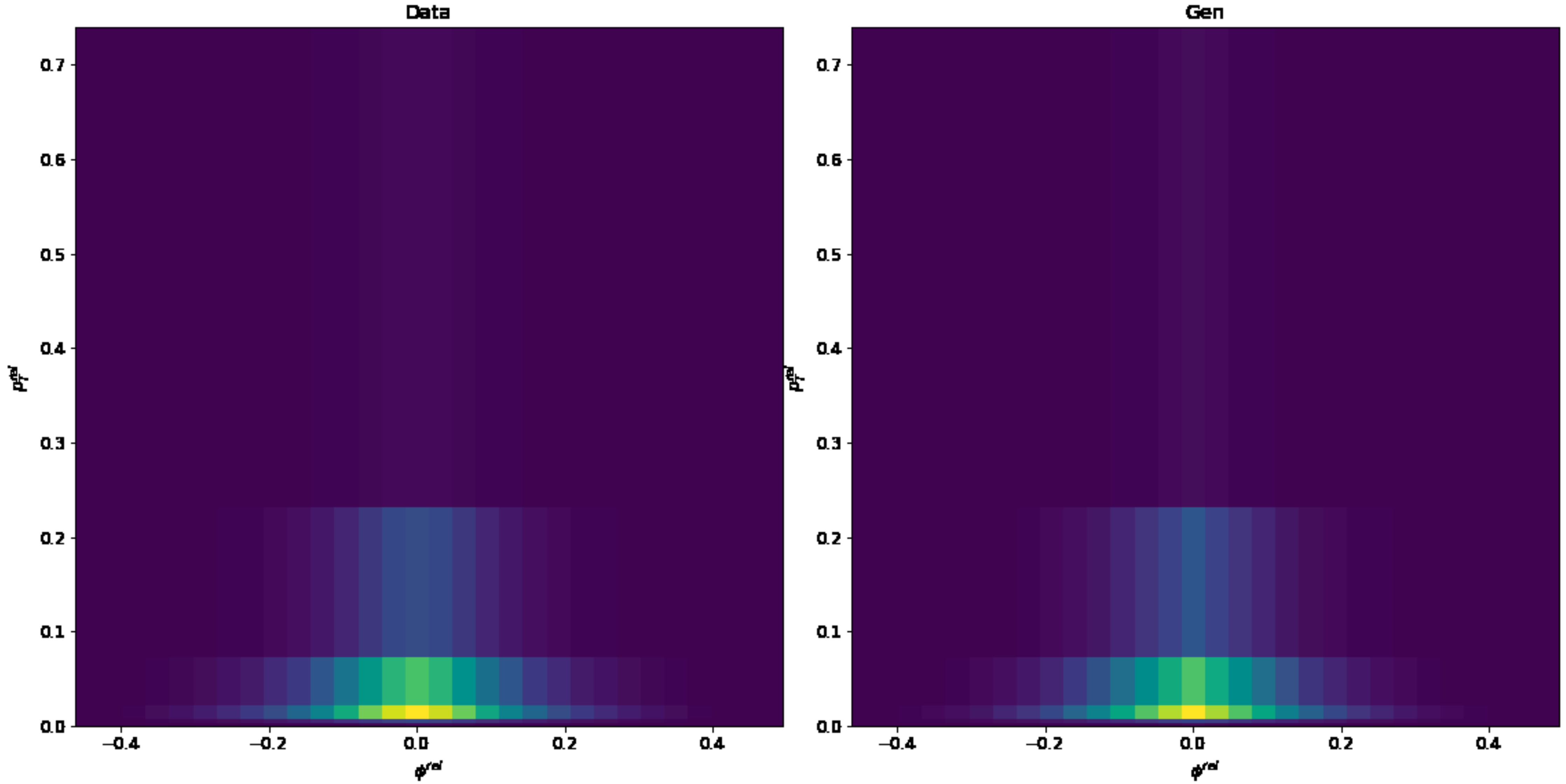
2D Histograms

$\eta^{rel}\phi^{rel}$



2D Histograms

$\phi^{rel}\, p_T^{rel}$



2D Histograms

$\eta^{rel}\, p_T^{rel}$