UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

High Multiplicity with JetGPT

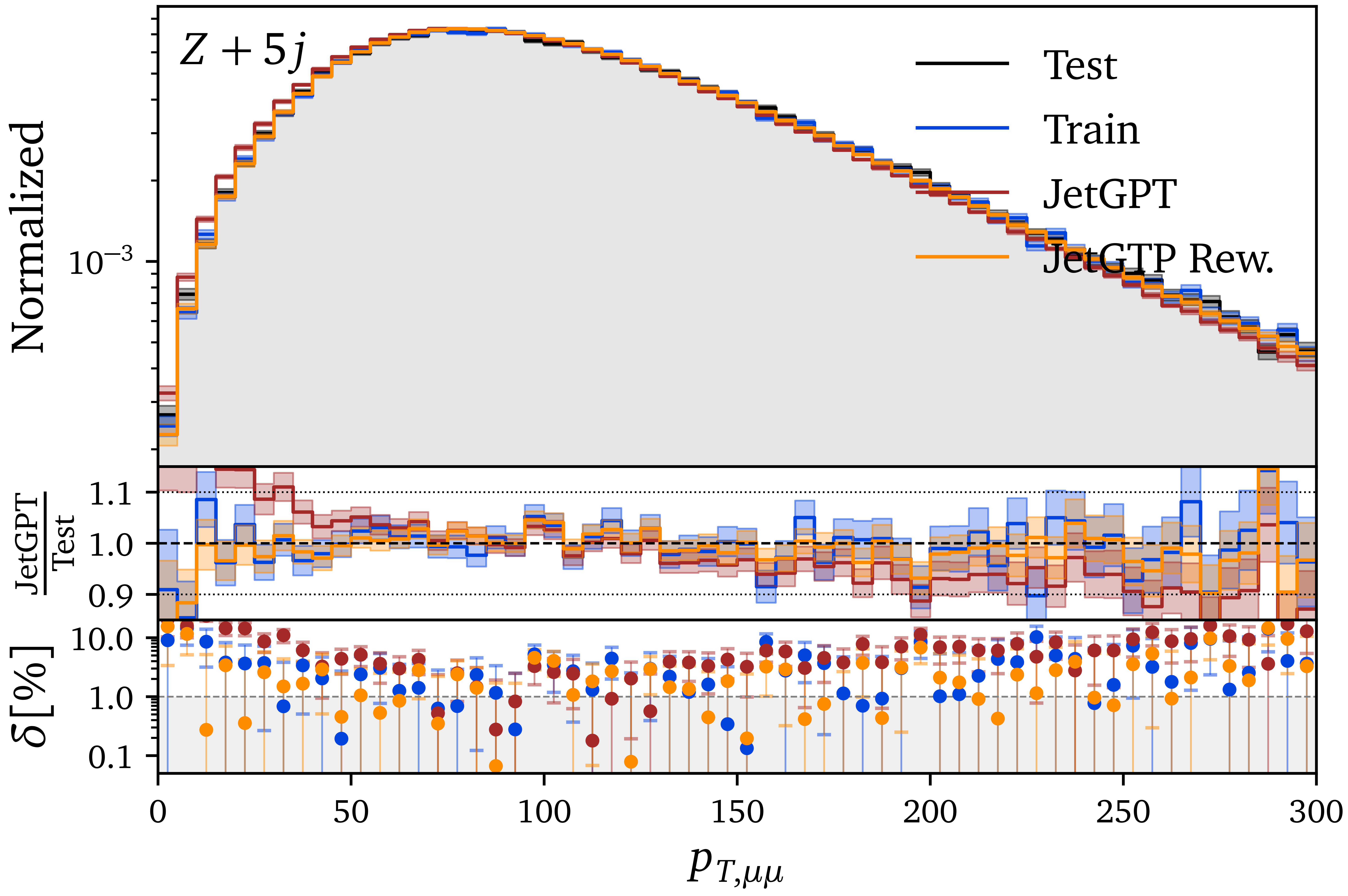
LHC Event Generation with Autoregressive Transformers

Jonas Spinner

Based on work in collaboration with:
Anja Butter, Nathanael Ediger, Nathan Hütsch,
Maeve Madigan, Sofia Palacios and Tilman Plehn

2305.10475

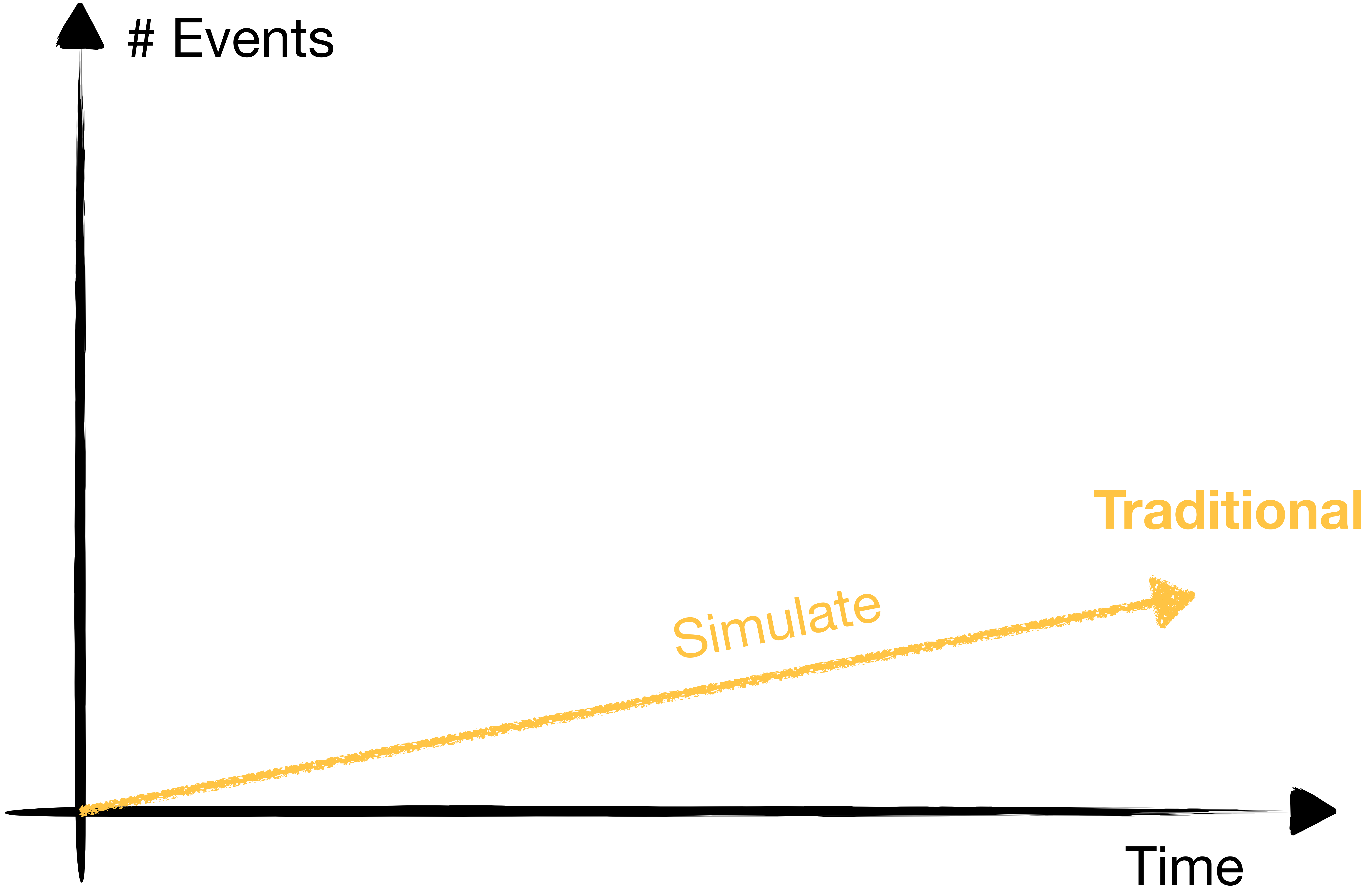
GlühWien 2023



$Z+5j$
Test
Train
JetGPT
JetGTP Rew.
Normalized
10^{-3}
$\frac{\text{JetGPT}}{\text{Test}}$
1.1
1.0
0.9
$\delta[\%]$
10.0
1.0
0.1
0
50
100
150
200
250
300
$p_{T,\mu\mu}$

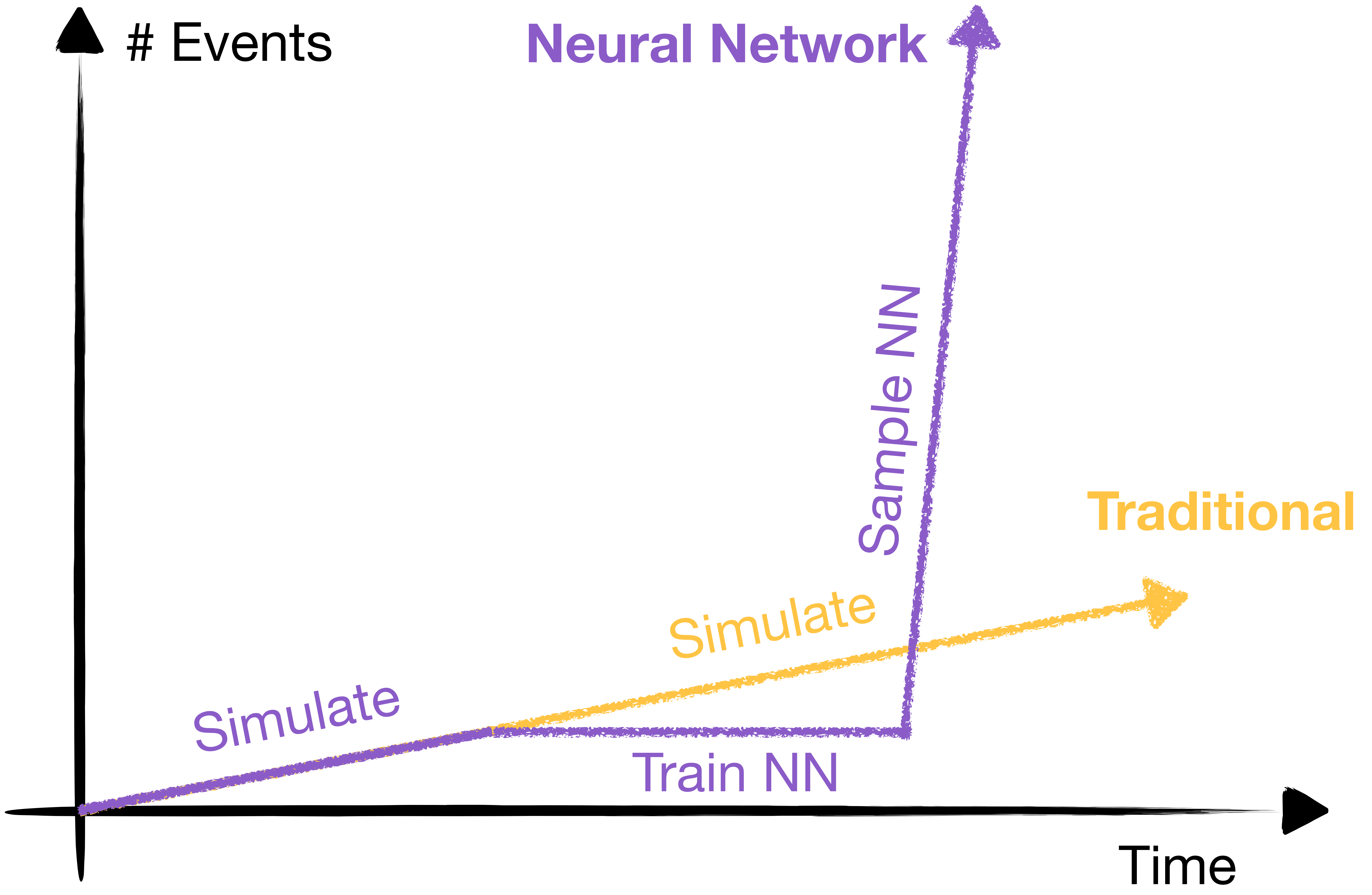
Motivation

End-to-End-Generation with Neural Networks



Motivation

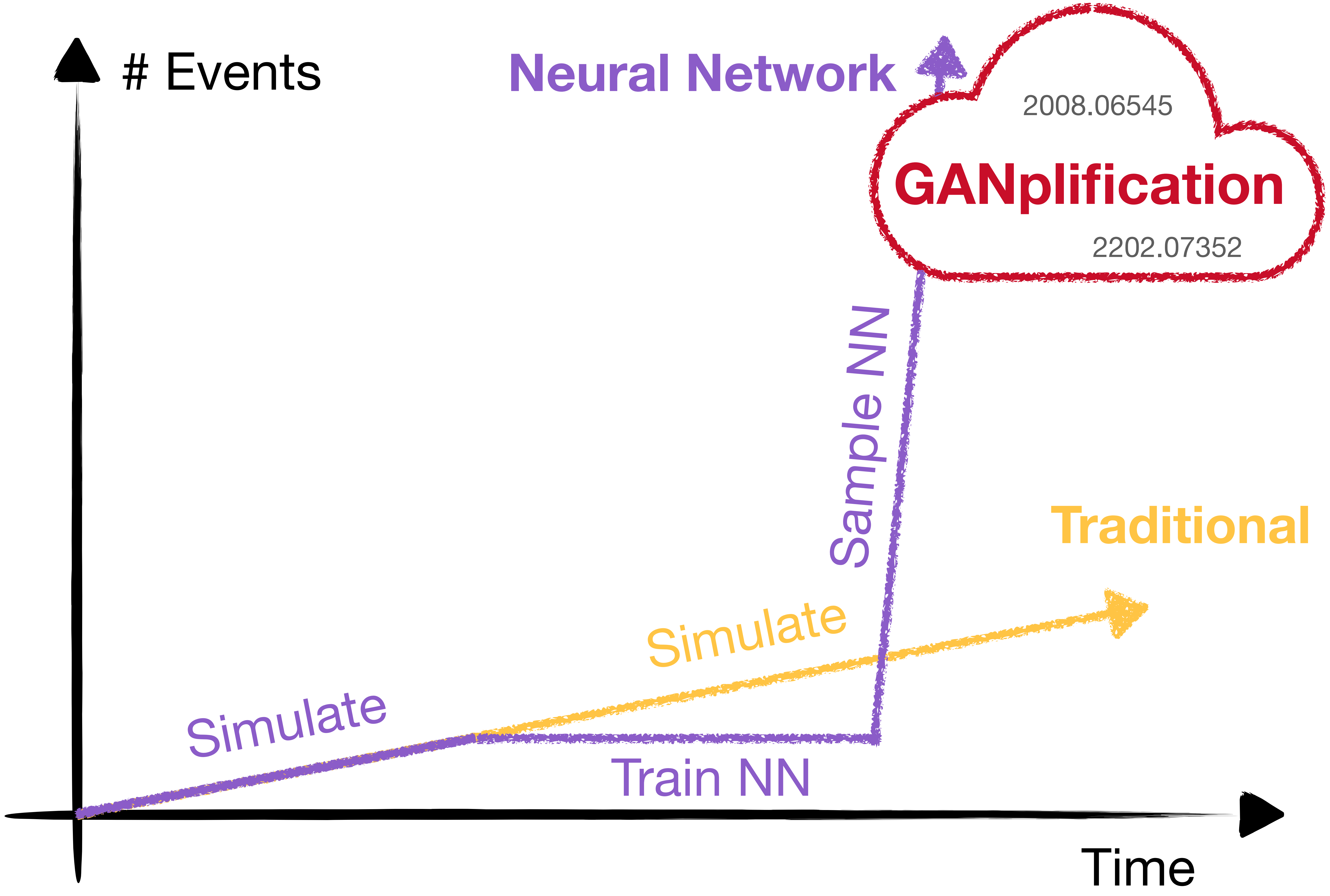
End-to-End-Generation with Neural Networks



- **Faster** when many events are required
- NNs are a more **efficient** encoding of distributions
- NNs **scale better** towards complex processes

Motivation

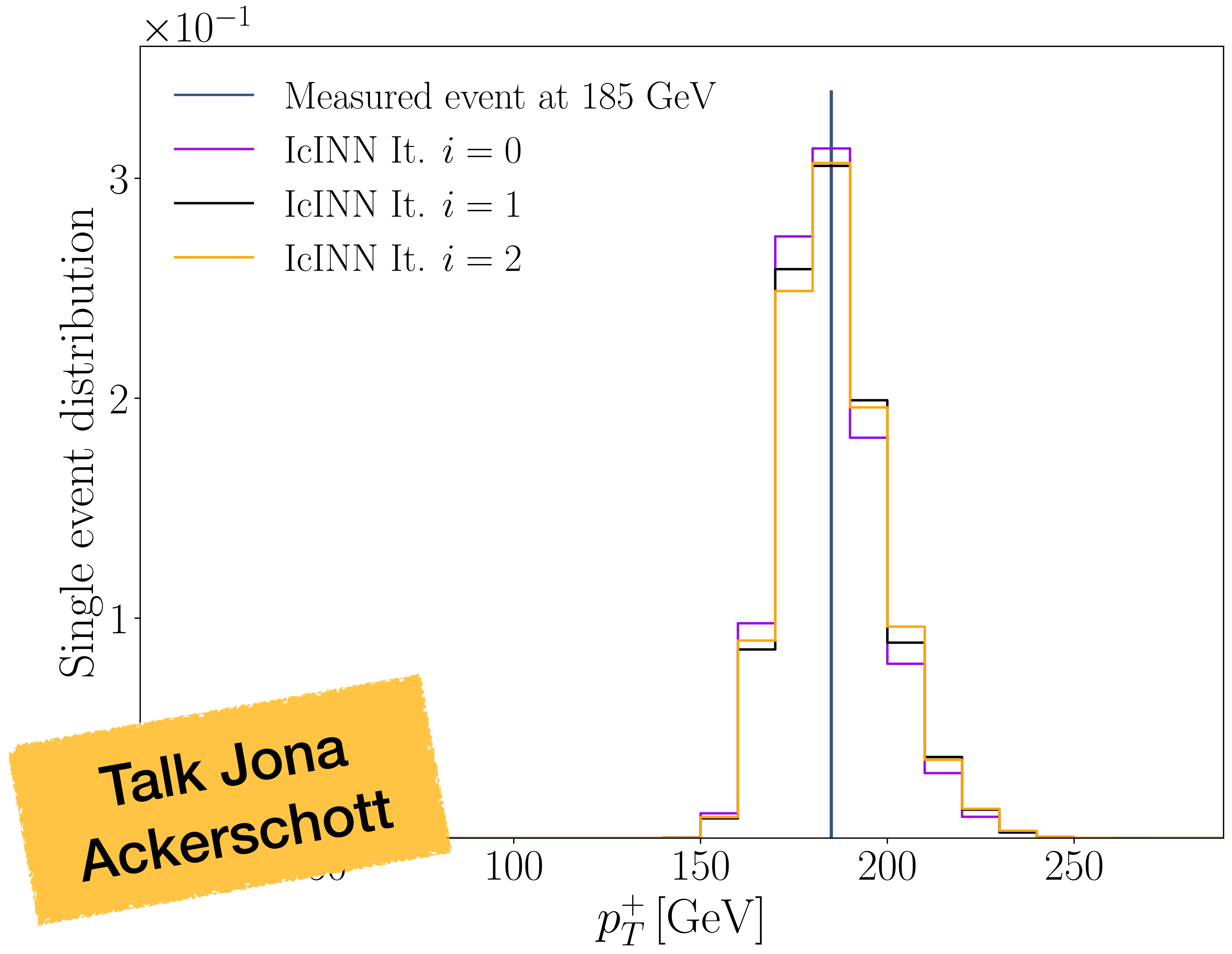
End-to-End-Generation with Neural Networks



- **Faster** when many events are required
- NNs are a more **efficient** encoding of distributions
- NNs **scale better** towards complex processes

Motivation

Inference with Conditional Generative Neural Networks



2006.06685

2308.00027

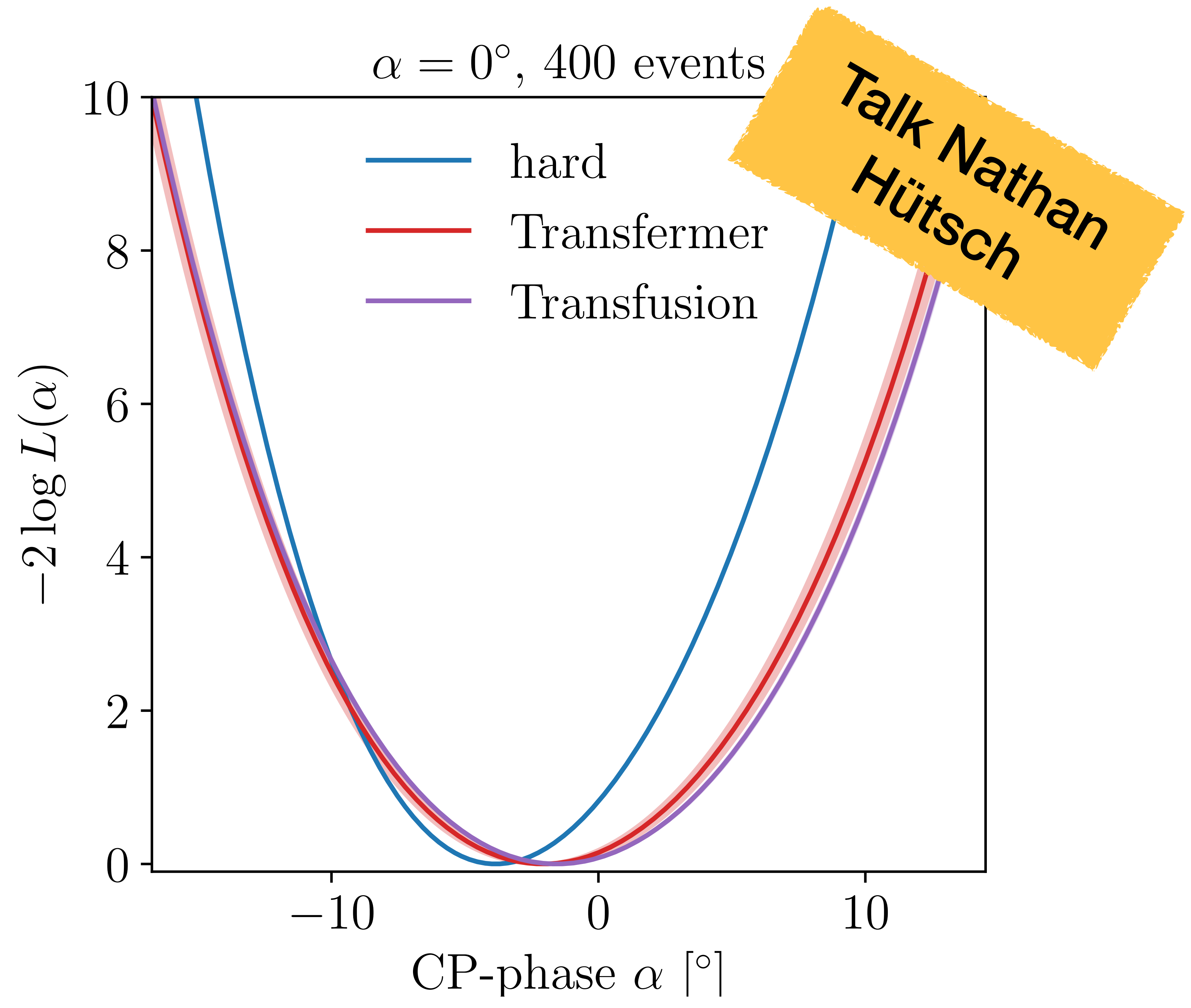
2305.10399

Generative Unfolding

1912.00477

2212.08674

2308.12351



2210.00019

2310.07752

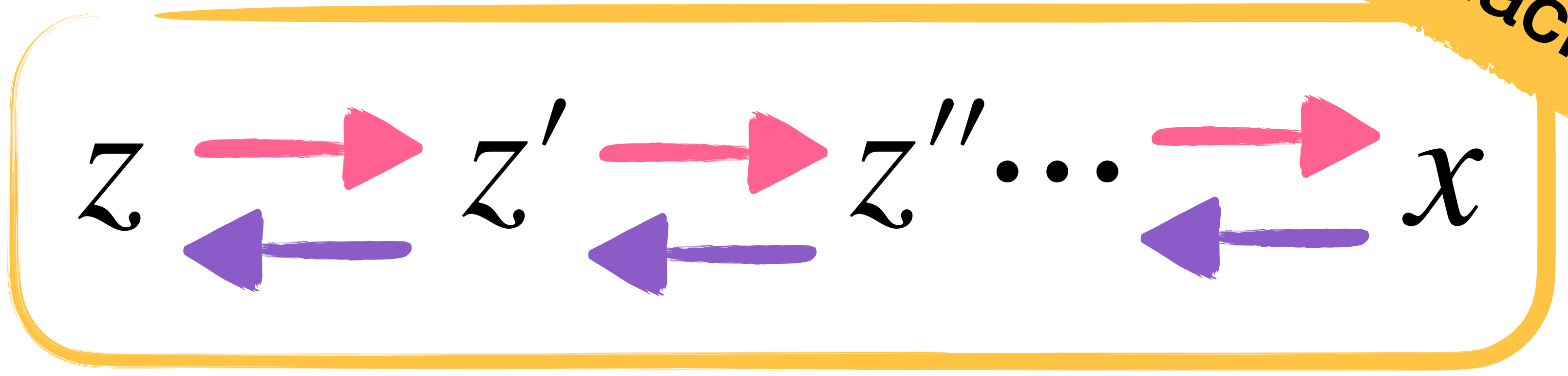
Generative Matrix Element Method

Motivation

Generative Neural Networks

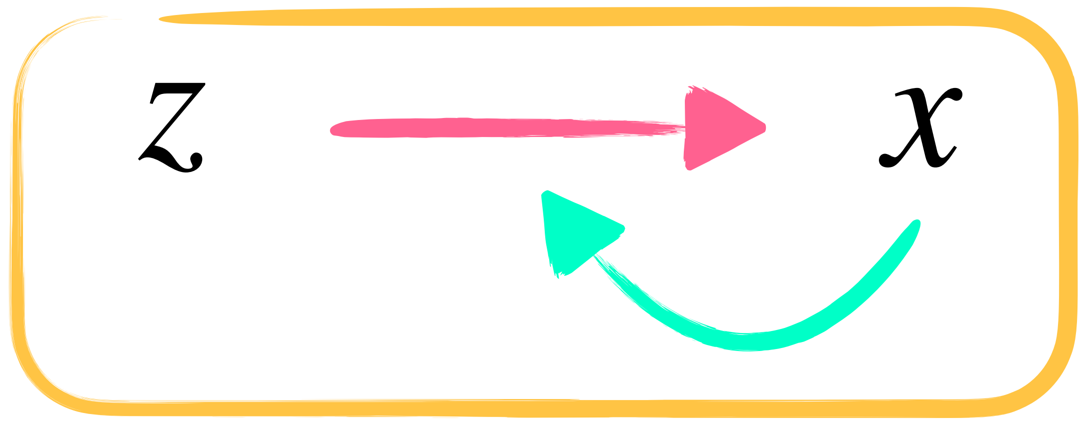
Diffusion Models

Talk Sofia Palacios

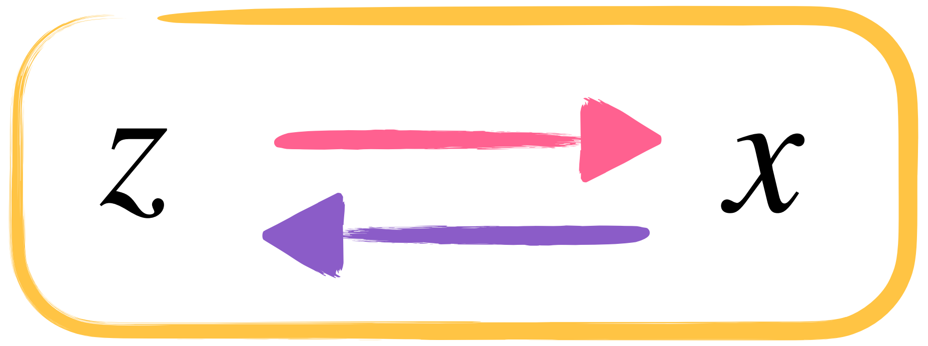


The ***Precise***

GANs

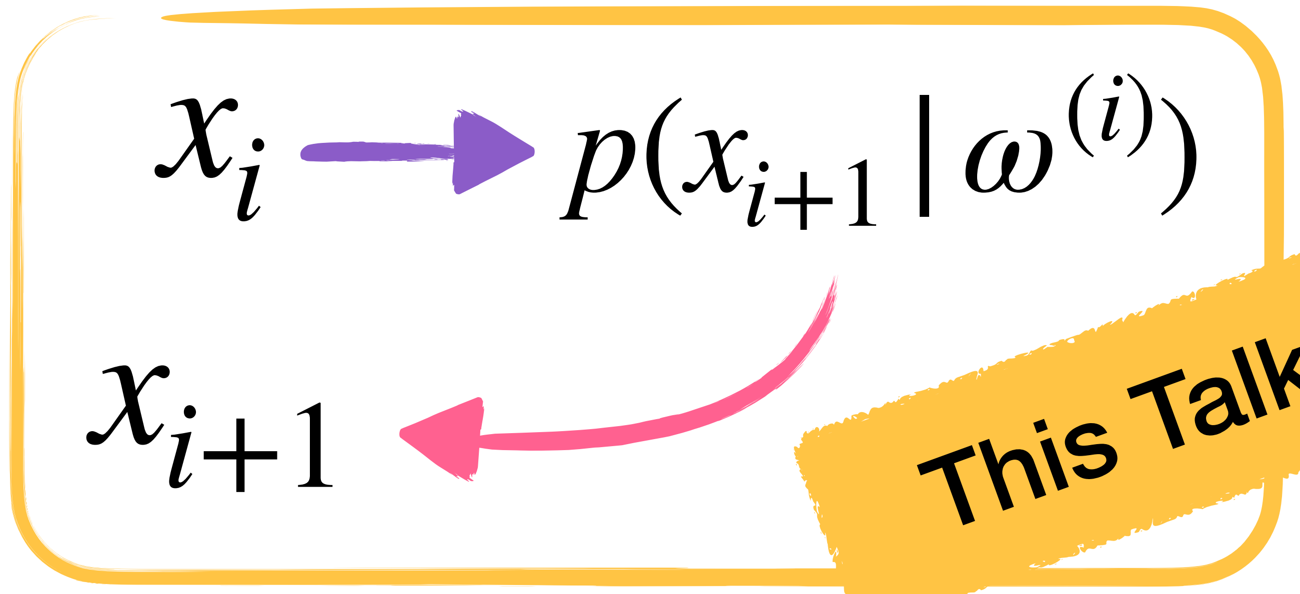


Normalizing Flows



The ***Fast***

Autoregressive Transformers



This Talk

The ***Flexible***

x **Phase Space**

z **Latent Space**

Sampling

Density Estimation

Classifier

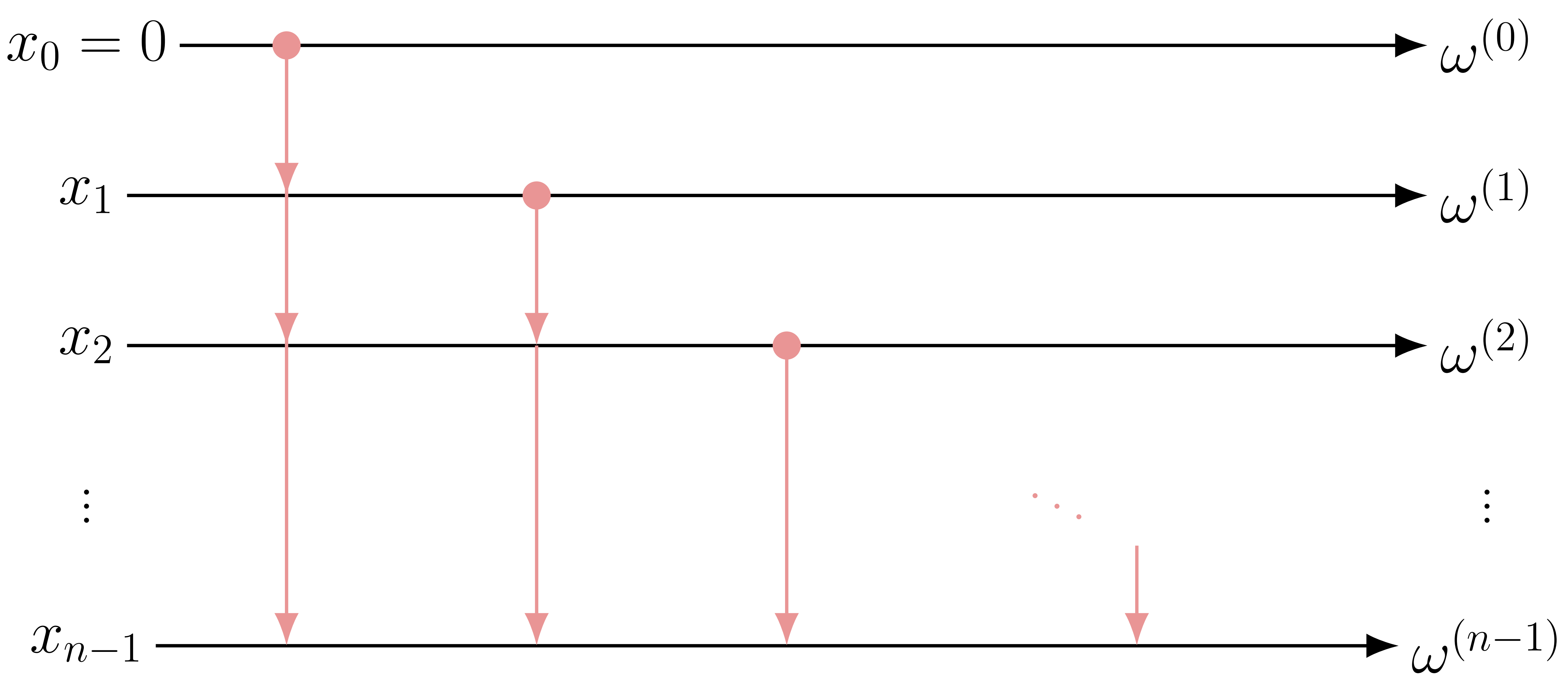
Autoregressive Transformers

Autoregressive Transformer

Generating Events

Autoregression

$$
\begin{aligned}
p(x_1, x_2 \cdots x_n) &= p(x_1) \quad p(x_2 \,|\, x_1) \quad \cdots \quad p(x_n \,|\, x_1 \cdots x_{n-1}) \\
&= p(x_1 \,|\, \omega^{(0)}) \quad p(x_2 \,|\, \omega^{(1)}) \quad \cdots \quad p(x_n \,|\, \omega^{(n-1)})
\end{aligned}
$$



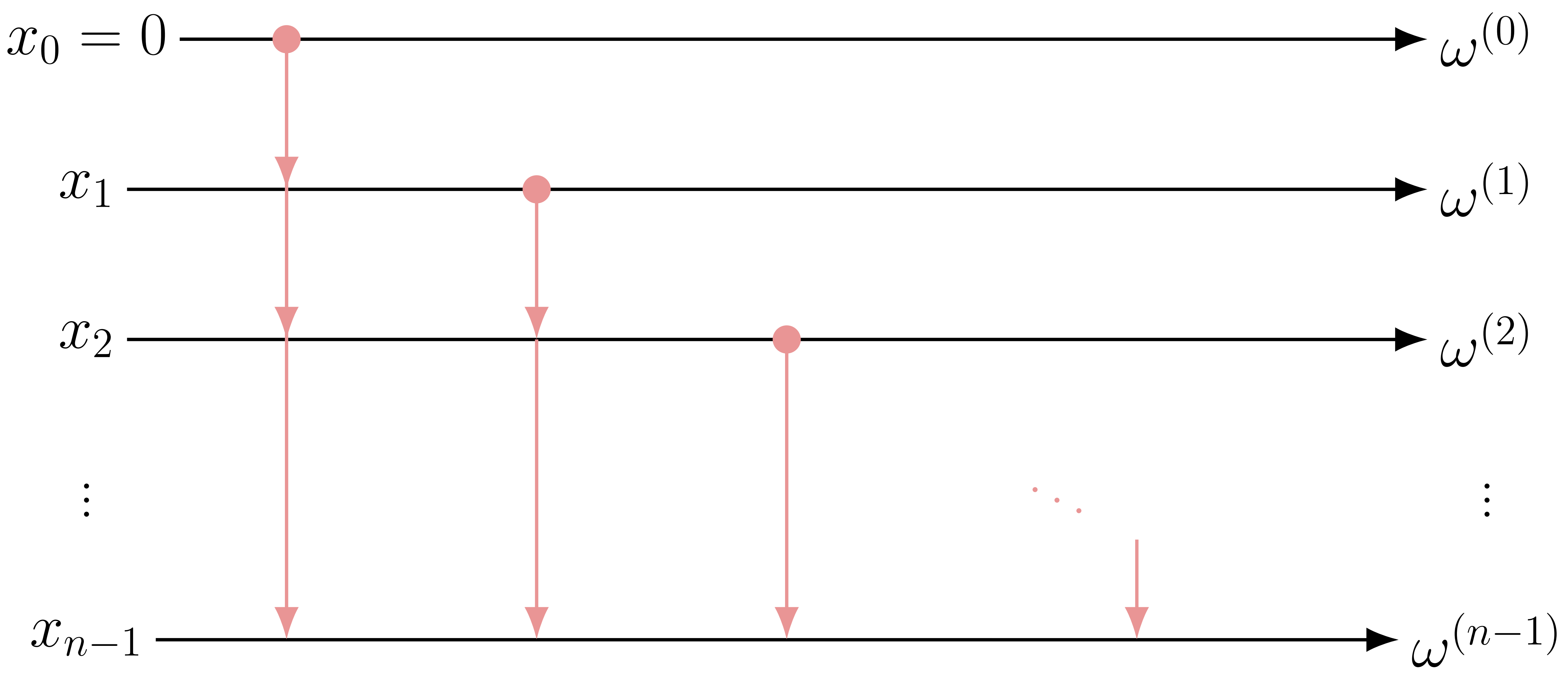
Autoregressive Transformer

Generating Events

Autoregression

$$p(x_1, x_2 \cdots x_n) = p(x_1) \qquad p(x_2 | x_1) \qquad \cdots \qquad p(x_n | x_1 \cdots x_{n-1})$$

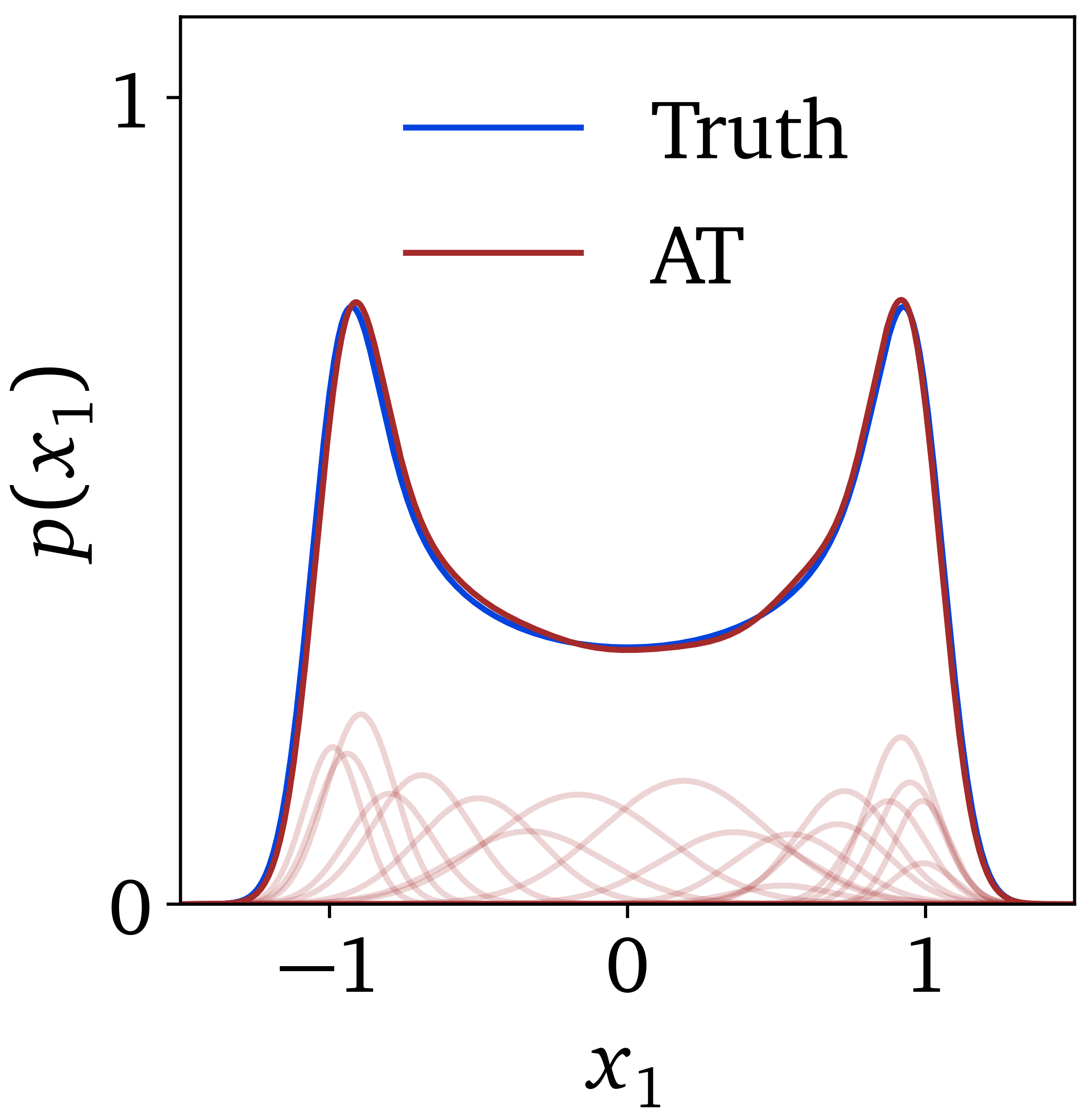
$$= p(x_1 | \omega^{(0)}) \qquad p(x_2 | \omega^{(1)}) \qquad \cdots \qquad p(x_n | \omega^{(n-1)})$$



Gaussian Mixture Model

$$\omega^{(i)} = \{w_j^{(i)}, \mu_j^{(i)}, \sigma_j^{(i)}\}$$

$$p(x_{i+1} | \omega^{(i)}) = \sum_j w_j^{(i)} \mathcal{N}(x_{i+1}; \mu_j^{(i)}, \sigma_j^{(i)})$$



$p(x_1, x_2)$
x_2
x_1
$p(x_1)$
Truth
AT
$p(x_2|x_1 = -1.2)$
x_2

$p(x_1, x_2)$
x_2
x_1
$p(x_1)$
Truth
AT
$p(x_2|x_1 = -0.9)$
Truth
AT
x_2

$p(x_1, x_2)$
x_2
x_1
Truth
AT
$p(x_1)$
x_1
$p(x_2|x_1 = -0.0)$
Truth
AT
x_2

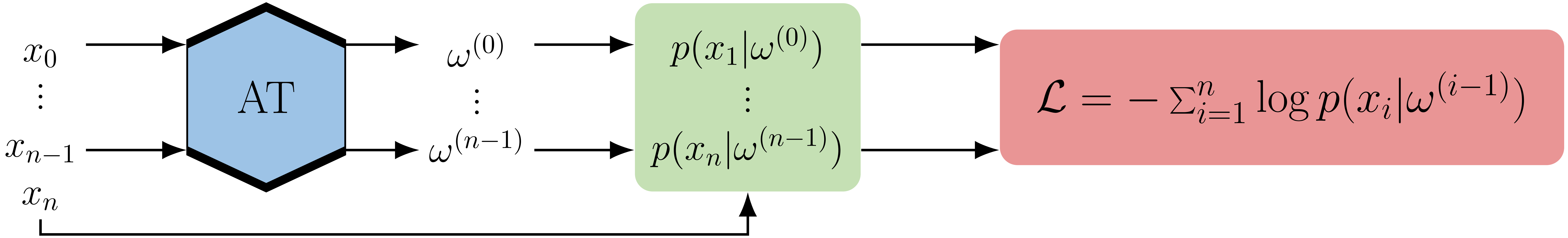
$p(x_1, x_2)$
x_2
x_1
Truth
AT
$p(x_1)$
x_1
$p(x_2|x_1 = 0.6)$
Truth
AT
x_2

$p(x_1, x_2)$
x_2
x_1
$p(x_1)$
Truth
AT
$p(x_2|x_1 = 1.0)$
x_2


$p(x_1, x_2)$
x_2
x_1
$p(x_1)$
Truth
AT
$p(x_2|x_1 = 1.4)$
x_2

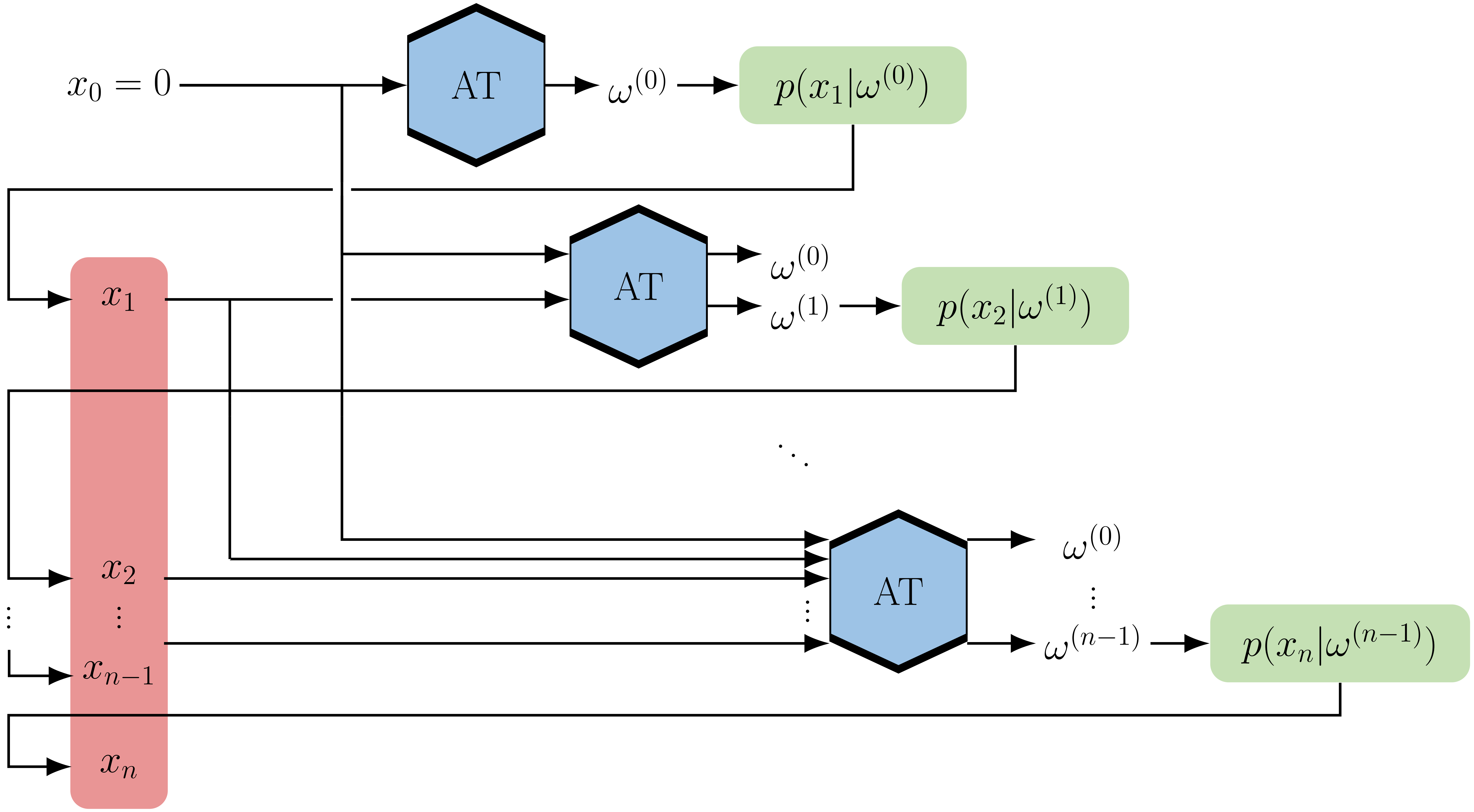
Autoregressive Transformer

(Fast) Density Estimation



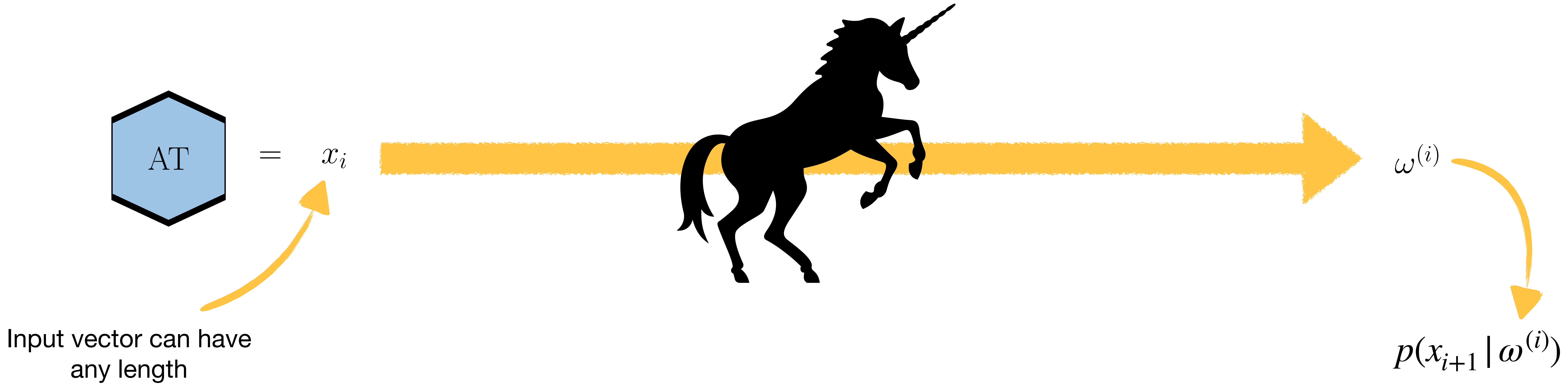
Autoregressive Transformer

(Slow) Sampling



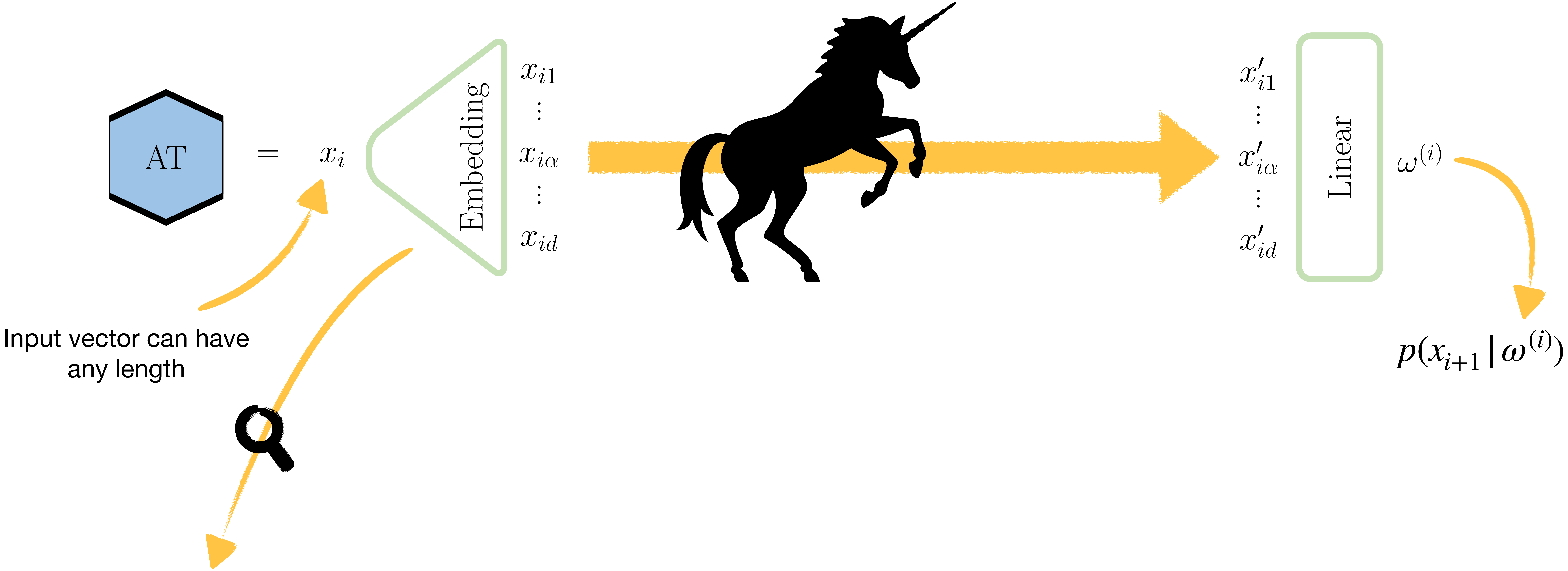
Autoregressive Transformer

Transformer Architecture 1/3



Autoregressive Transformer

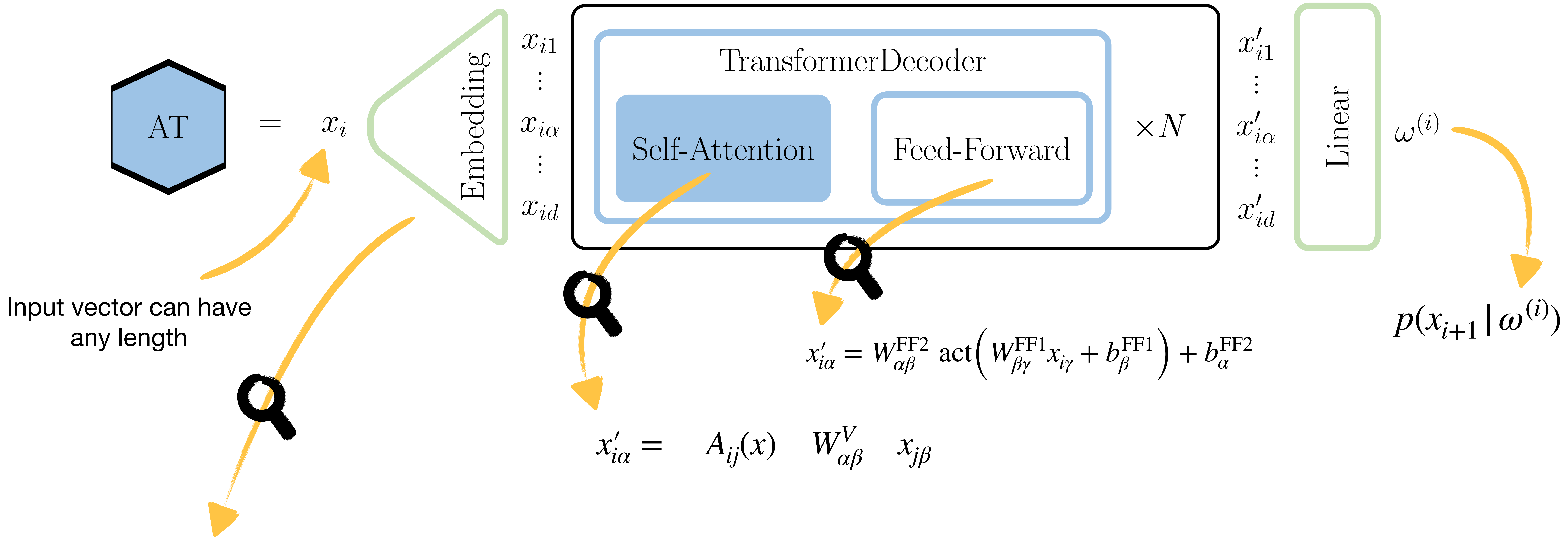
Transformer Architecture 2/3



$$x_{i\alpha} = W^{\mathrm{TE}}_{\alpha} x_i + W^{\mathrm{PE}}_{\alpha\beta}\ \mathrm{OneHot}_{\beta}(i)$$

Autoregressive Transformer

Transformer Architecture 3/3



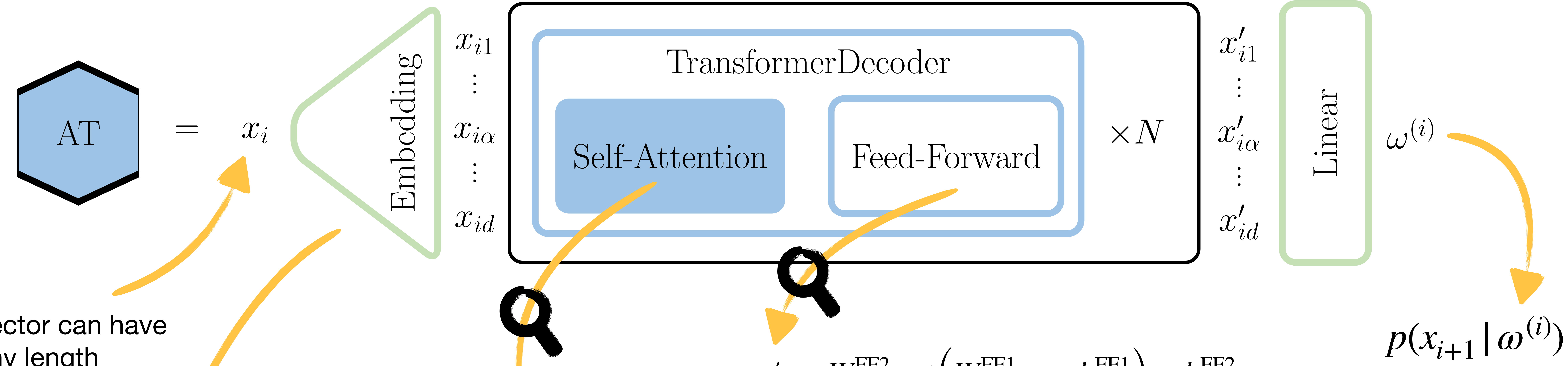
$$x'_{i\alpha} = W^{\mathrm{FF2}}_{\alpha\beta}\ \mathrm{act}\Big(W^{\mathrm{FF1}}_{\beta\gamma} x_{i\gamma} + b^{\mathrm{FF1}}_{\beta}\Big) + b^{\mathrm{FF2}}_{\alpha}$$

$$x'_{i\alpha} = \quad A_{ij}(x) \quad W^{V}_{\alpha\beta} \quad x_{j\beta}$$

$$x_{i\alpha} = W^{\mathrm{TE}}_{\alpha} x_i + W^{\mathrm{PE}}_{\alpha\beta}\ \mathrm{OneHot}_{\beta}(i)$$

Autoregressive Transformer

Transformer Architecture 3/3



Input vector can have any length

$$x'_{i\alpha} = W^{\mathrm{FF2}}_{\alpha\beta}\ \mathrm{act}\left(W^{\mathrm{FF1}}_{\beta\gamma} x_{i\gamma} + b^{\mathrm{FF1}}_{\beta}\right) + b^{\mathrm{FF2}}_{\alpha}$$

$$x'_{i\alpha} = \quad A_{ij}(x) \quad W^{V}_{\alpha\beta} \quad x_{j\beta}$$

$$x_{i\alpha} = W^{\mathrm{TE}}_{\alpha} x_i + W^{\mathrm{PE}}_{\alpha\beta}\ \mathrm{OneHot}_{\beta}(i)$$

$$A_{ij}(x) = \mathrm{Softmax}_j\left(\frac{x_{i\gamma} W^{Q}_{\delta\gamma} W^{K}_{\delta\sigma} x_{j\sigma}}{\sqrt{d}}\right)$$

*neglecting Dropout, Layer Normalisation, residual connections, Multi-Headed Attention

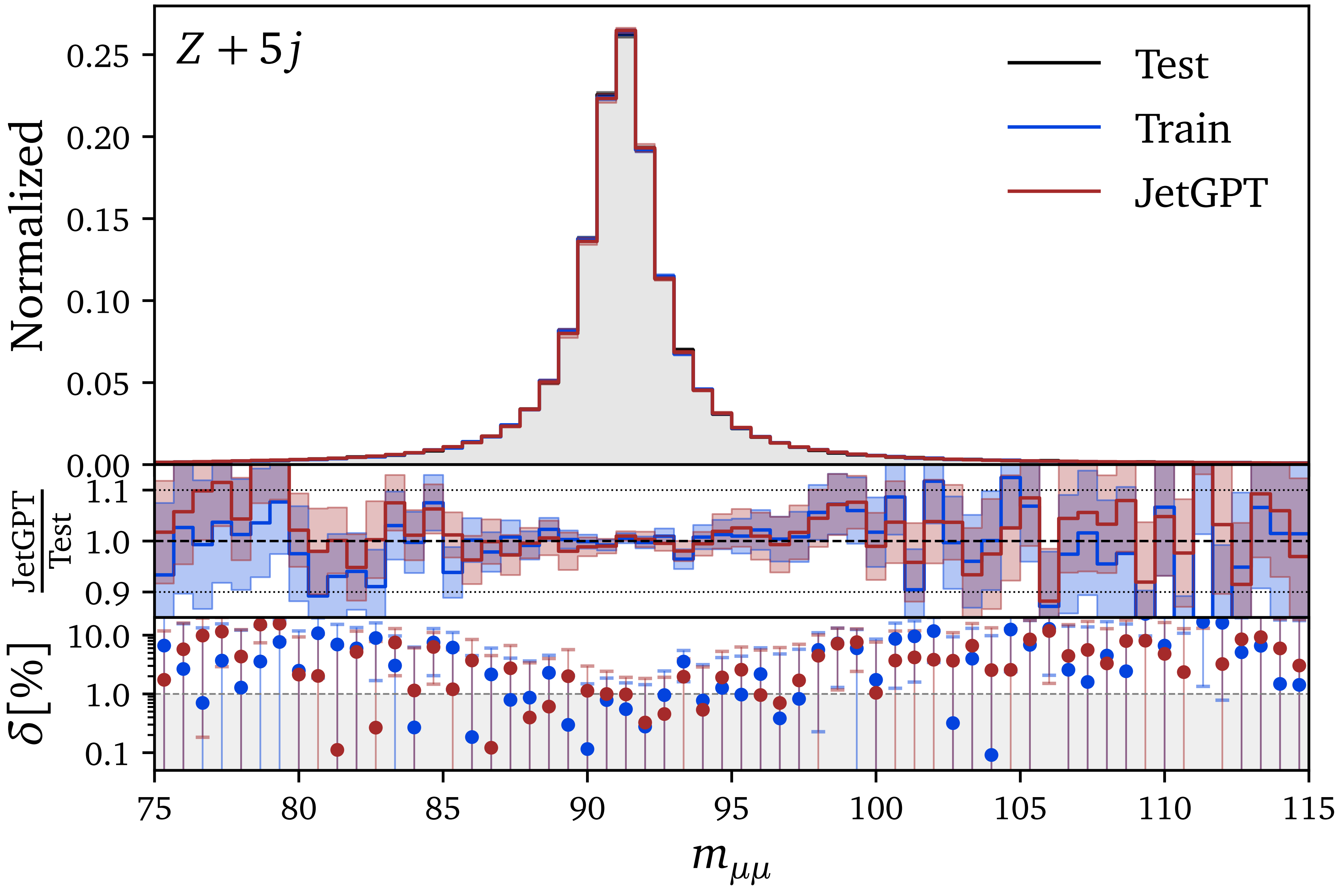
Generating LHC Events

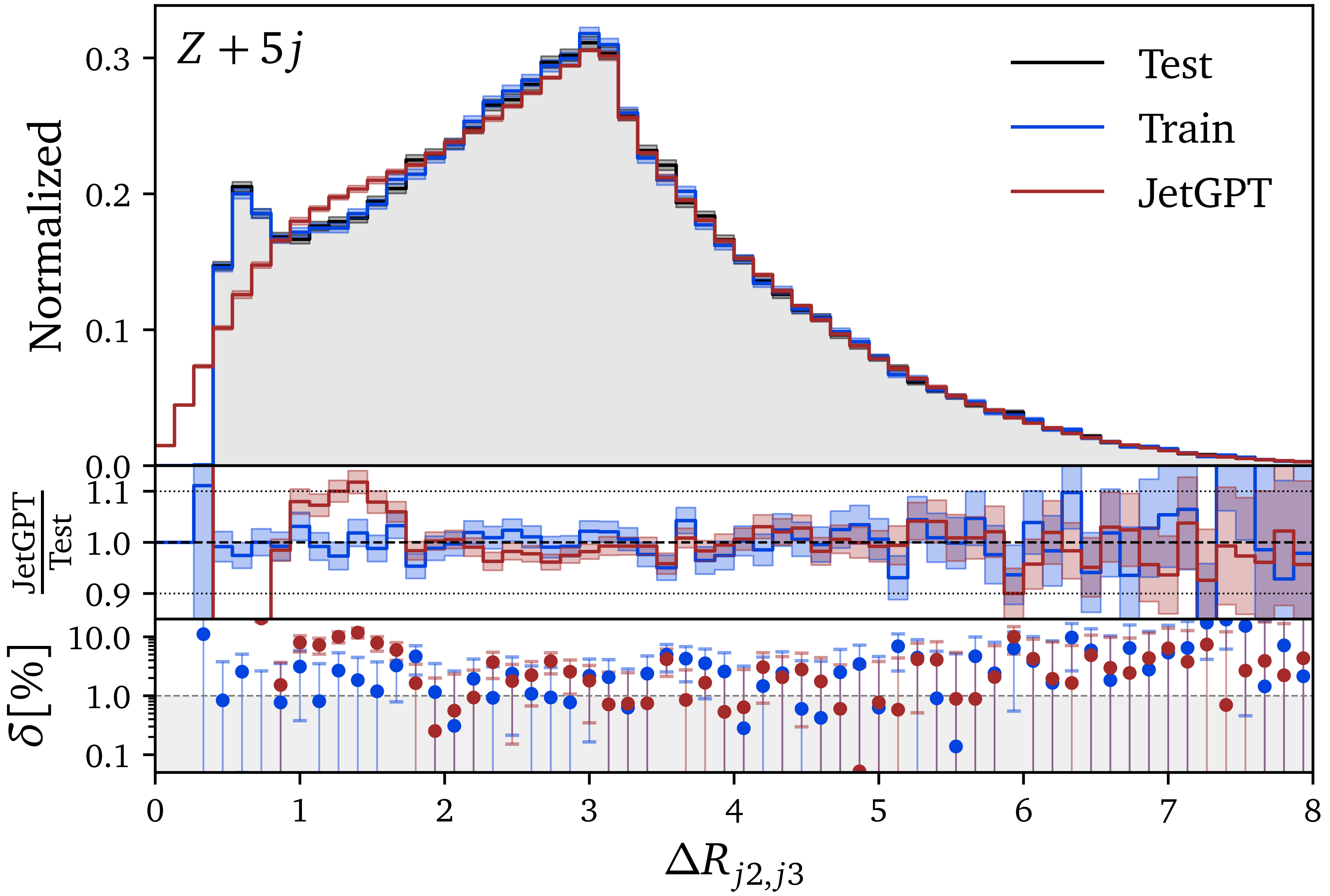
Generating LHC Events

Dataset: $Z(\mu\mu) + \text{jets}$

- MadGraph + Pythia
- Events with 3-5 jets (3j: 5M, 4j: 1M, 5j: 200k)
- Autoregressive Ordering:

$$\left\{ m_{\mu\mu}, \phi, p_T, \eta, m_j \right\}$$

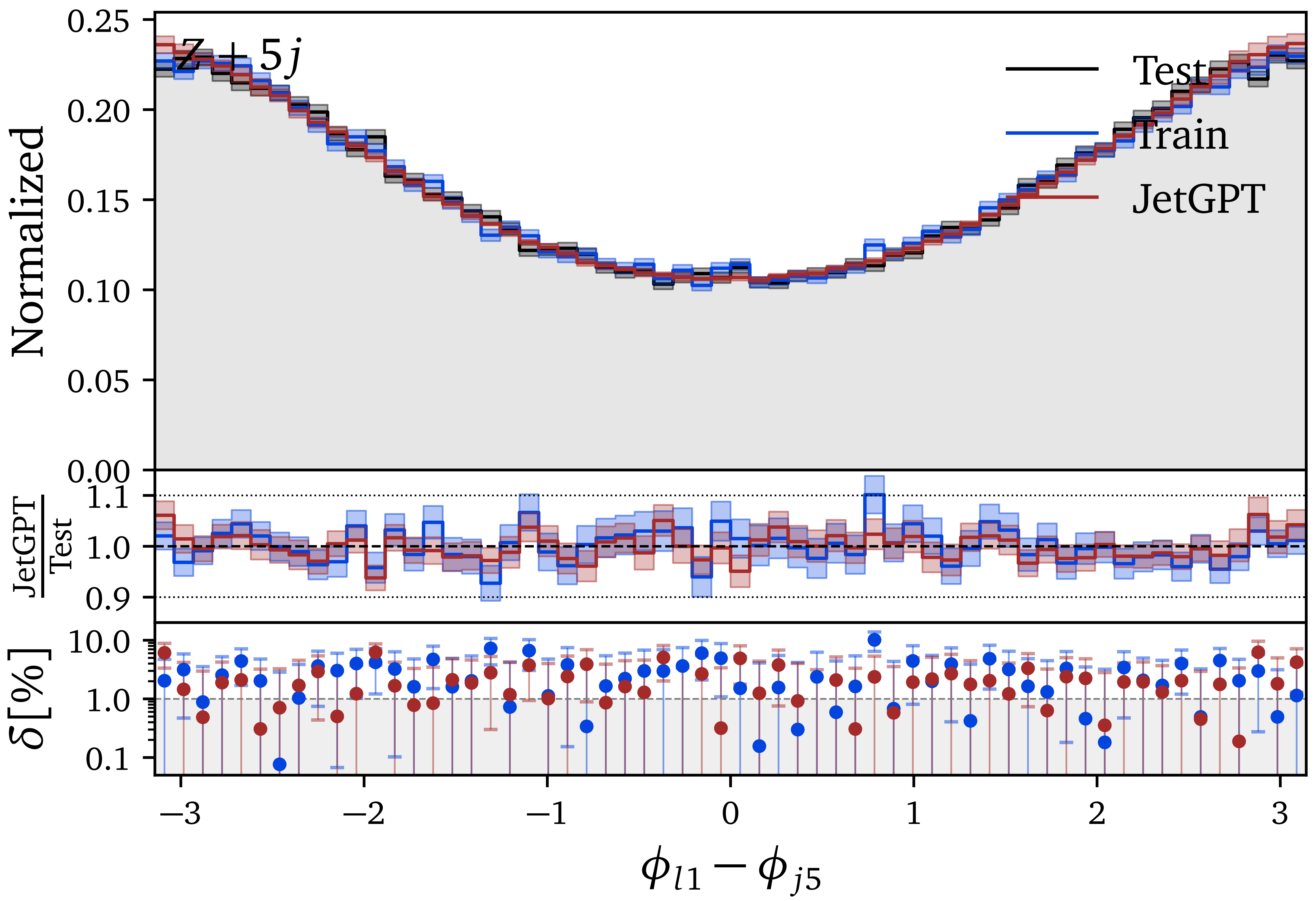


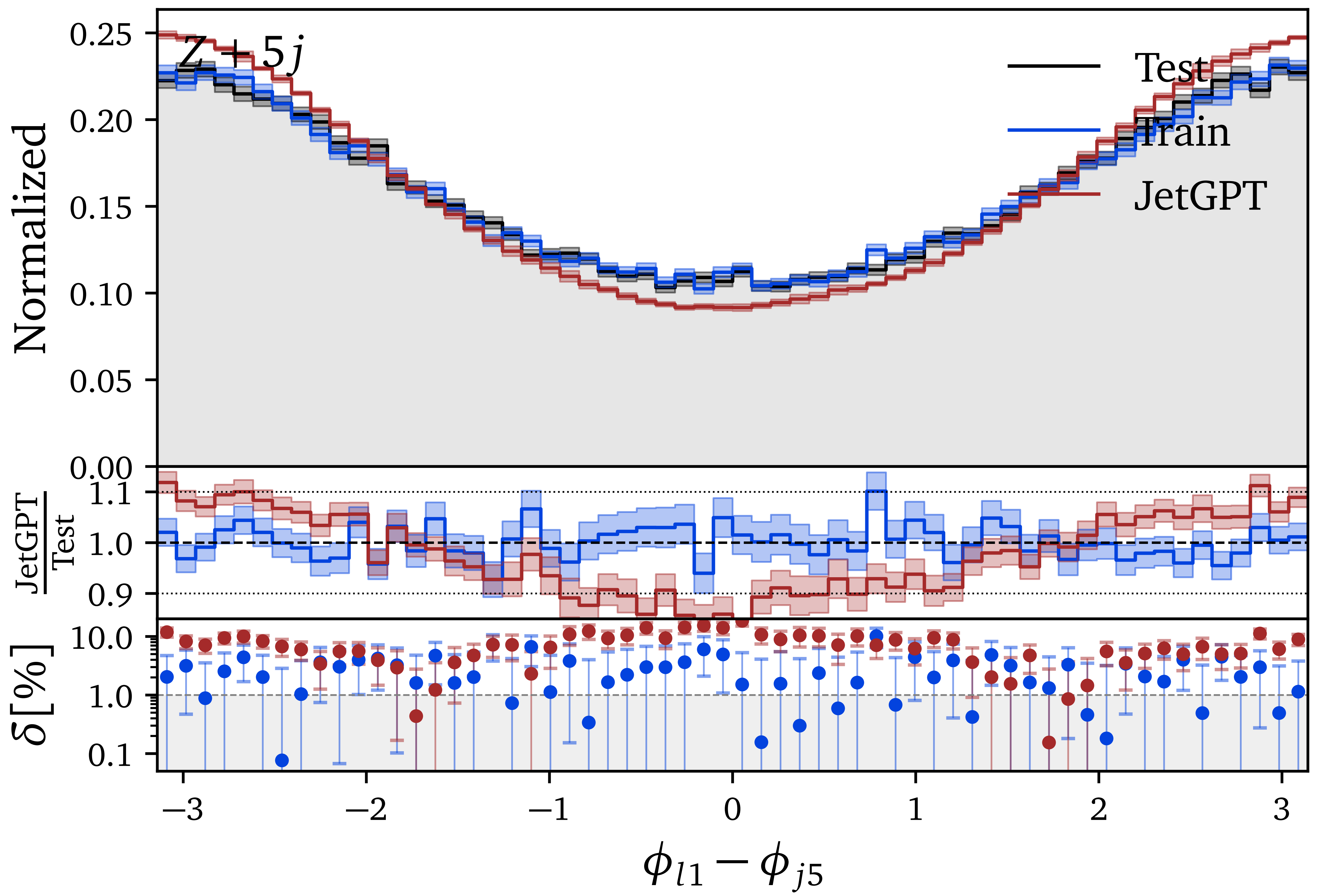


Generating LHC Events

Autoregressive Ordering

$$\left\{ m_{\mu\mu}, \boxed{\phi}, p_T, \eta, m_j \right\} \quad \text{flip ordering} \quad \left\{ m_j, \eta, p_T, \boxed{\phi}, m_{\mu\mu} \right\}$$

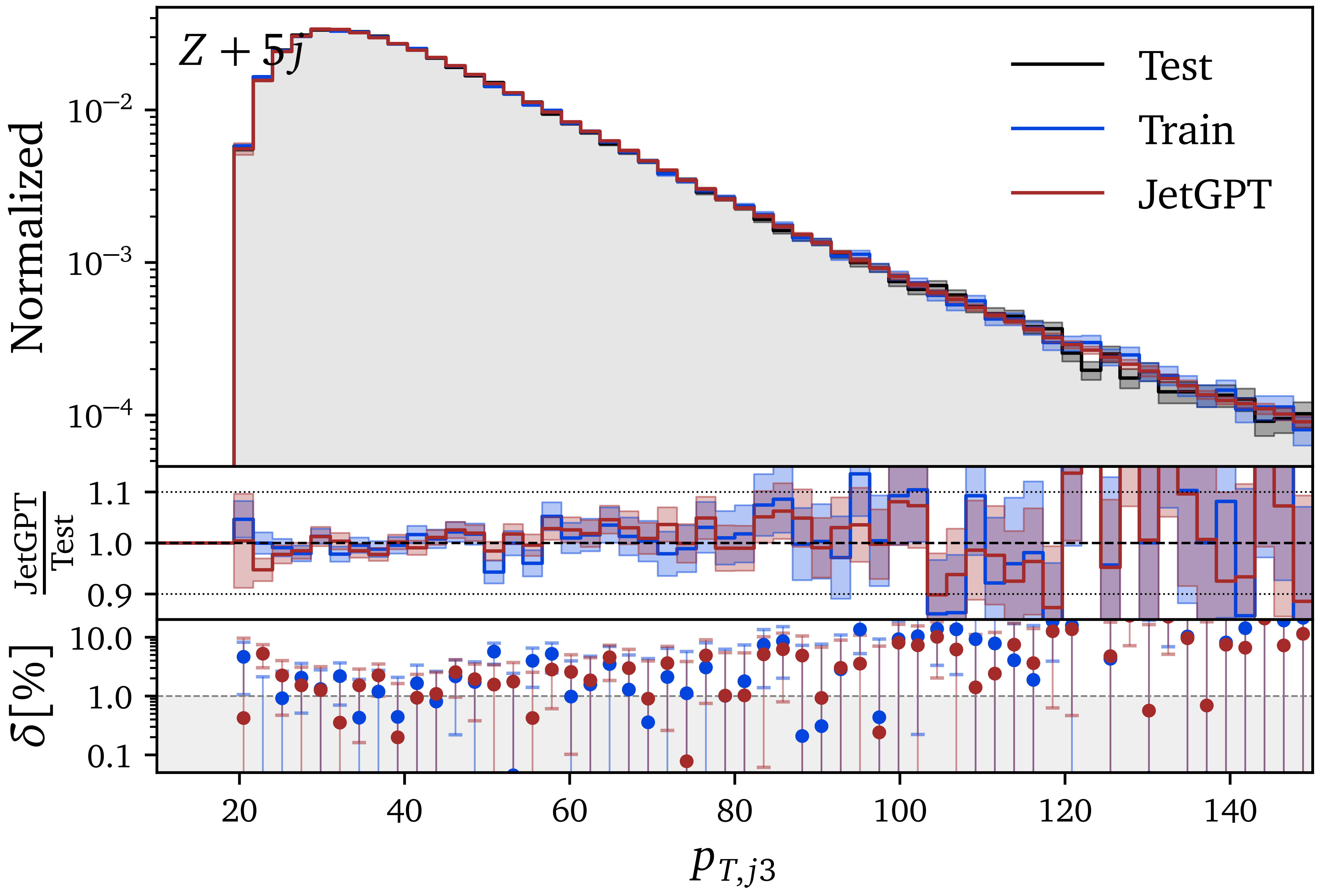


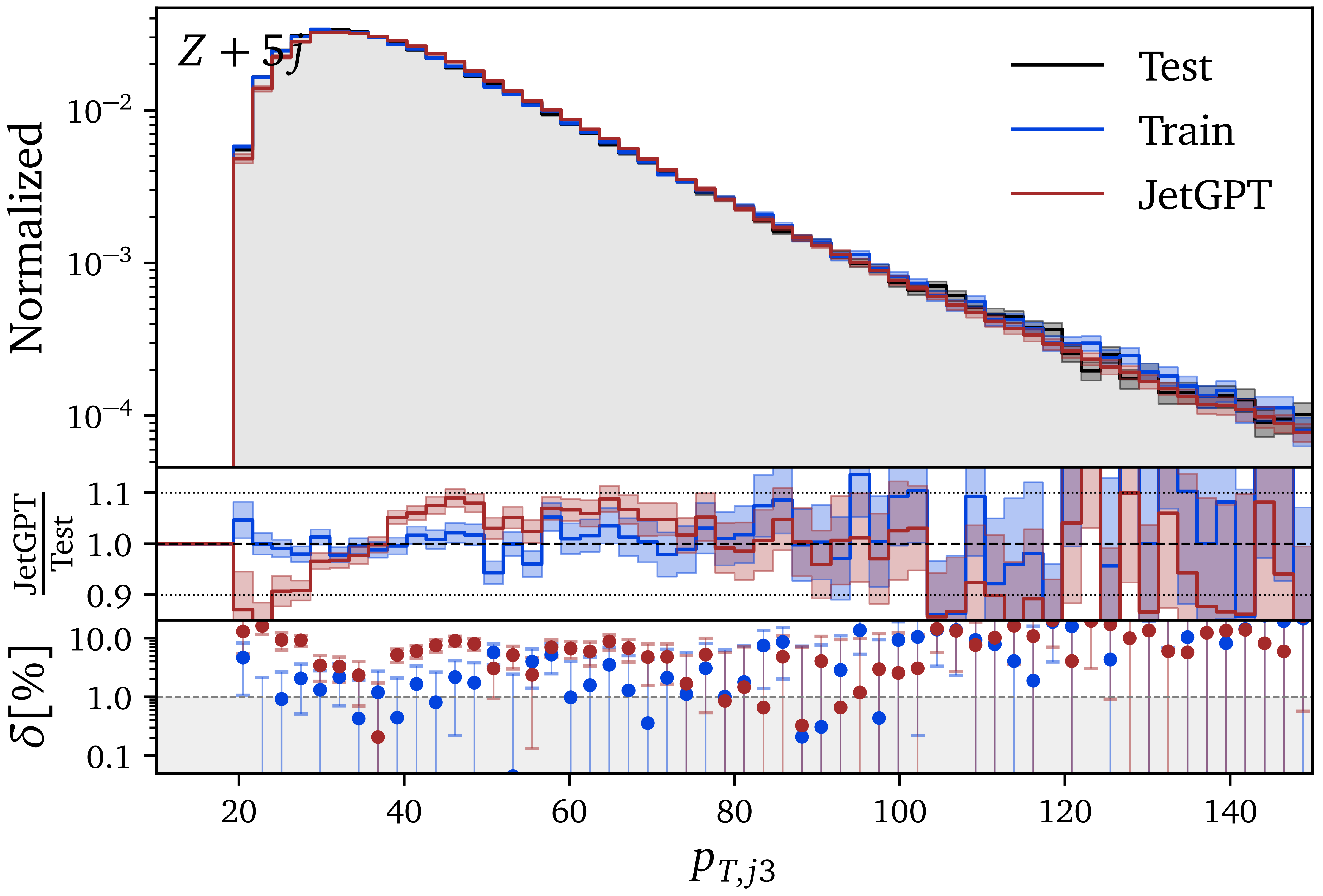


Generating LHC Events

Autoregressive Ordering

$$\left\{ m_{\mu\mu}, \phi, p_T, \eta, m_j \right\} \quad \text{flip ordering} \quad \left\{ m_j, \eta, p_T, \phi, m_{\mu\mu} \right\}$$

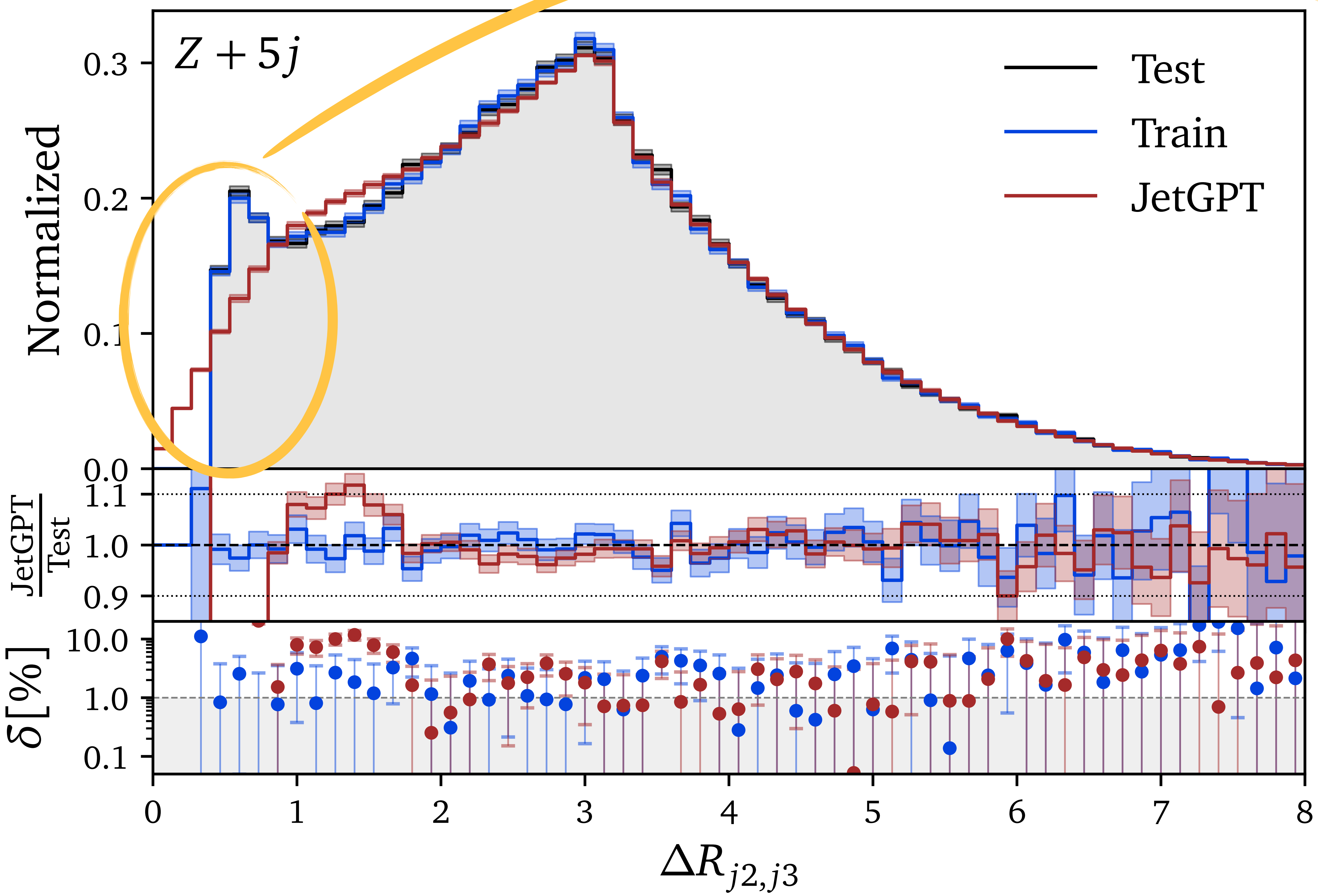




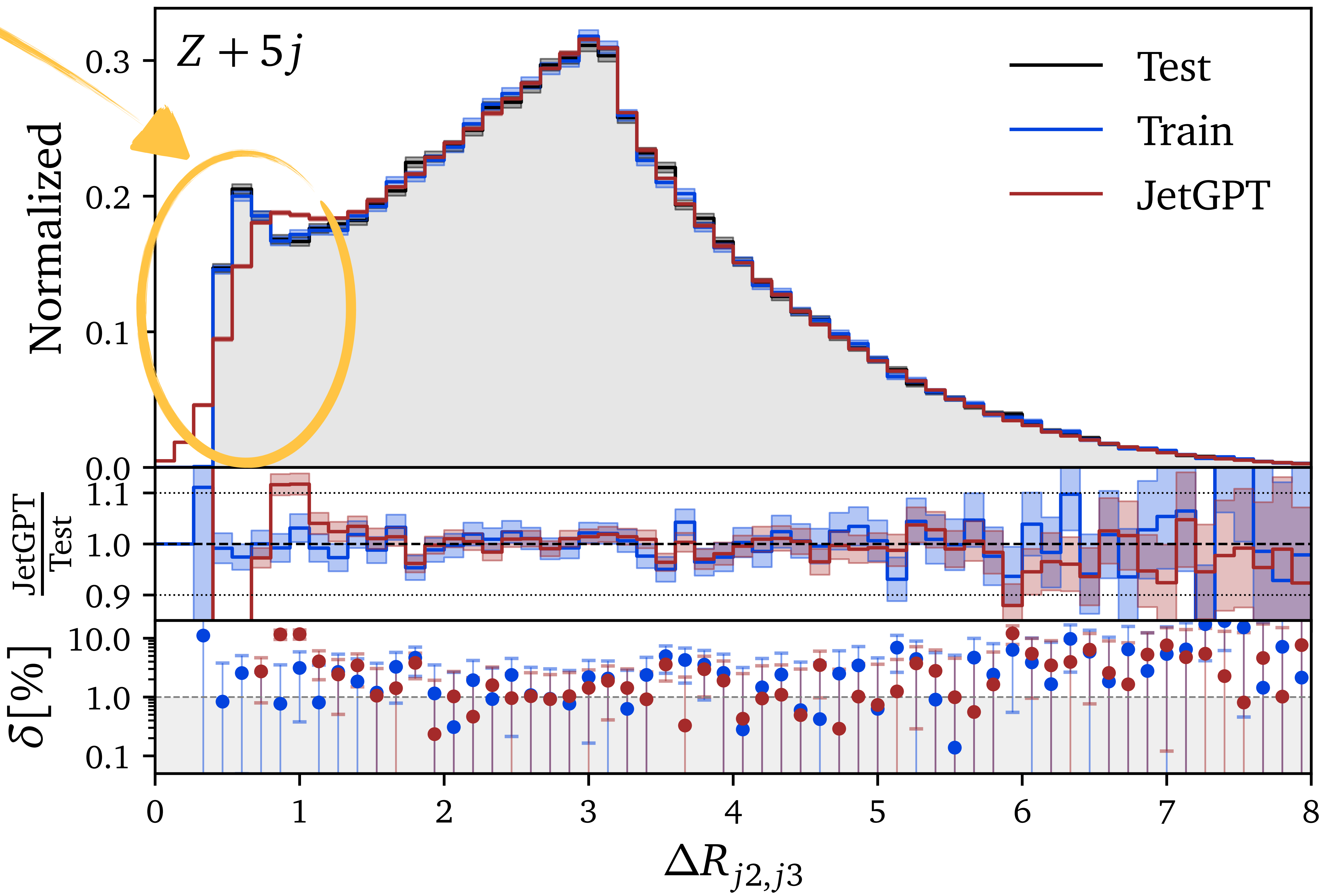
Generating LHC Events

Joint Training

Training on 5j only



Joint Training on 3j-5j



Generating LHC Events

Joint Training

Training on 5j only | **Joint Training on 3j-5j**

Classifier Reweighting

Classifier Reweighting

Likelihood Ratio Trick

$$\mathscr{L}_{\text{BCE}} = -\left\langle \log D(x) \right\rangle_{x\sim p_{\text{data}}} - \left\langle \log(1 - D(x)) \right\rangle_{x\sim p_{\text{model}}}$$

$$= -\int dx\ p_{\text{data}} \log D - \int dx\ p_{\text{model}} \log(1 - D)$$

Minimisation

$$0 = \frac{\delta \mathscr{L}_{\text{BCE}}}{\delta D} = -\frac{p_{\text{data}}}{D} + \frac{p_{\text{model}}}{1 - D}$$

$$\boxed{\frac{p_{\text{data}}}{p_{\text{model}}} = \frac{D}{1 - D}}$$

Classification

$$w(x) = \frac{p_{\text{data}}(x)}{p_{\text{model}}(x)}$$

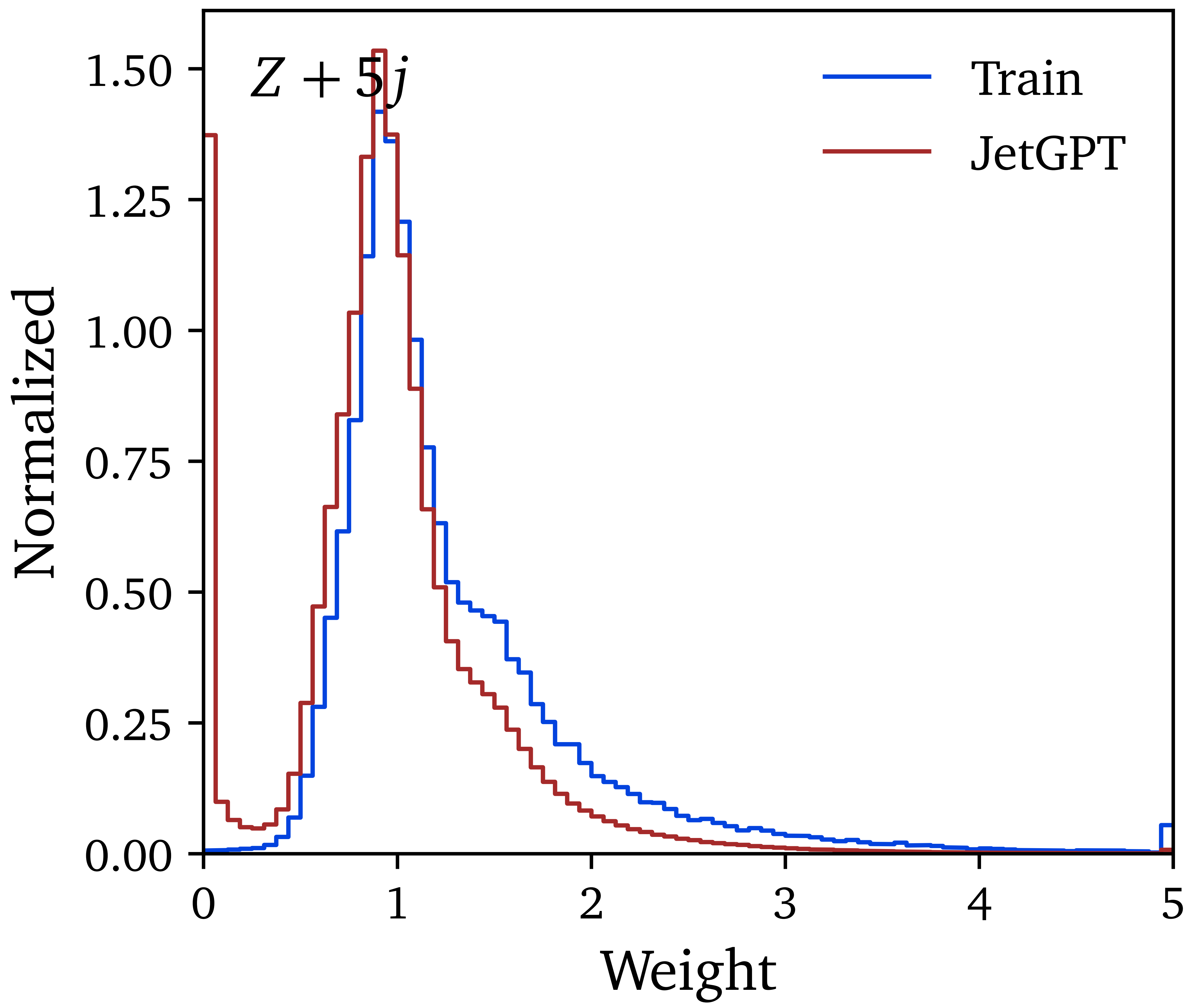
Reweighting

$$p_{\text{data}} = p_{\text{model}} \times \frac{p_{\text{data}}}{p_{\text{model}}}$$

Classifier Reweighting

Track the limitations

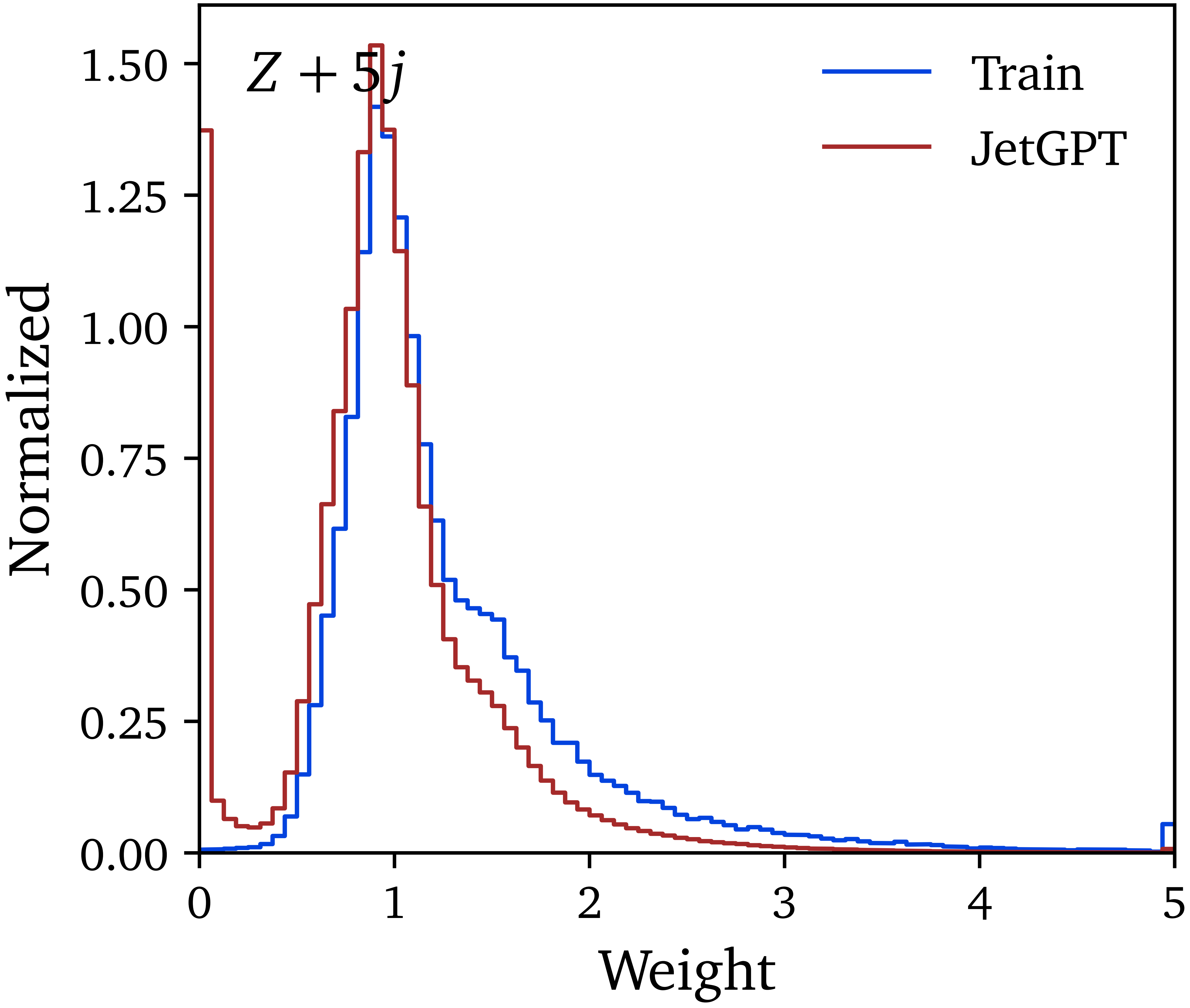
$$w(x) = \frac{p_{\text{data}}(x)}{p_{\text{model}}(x)}$$

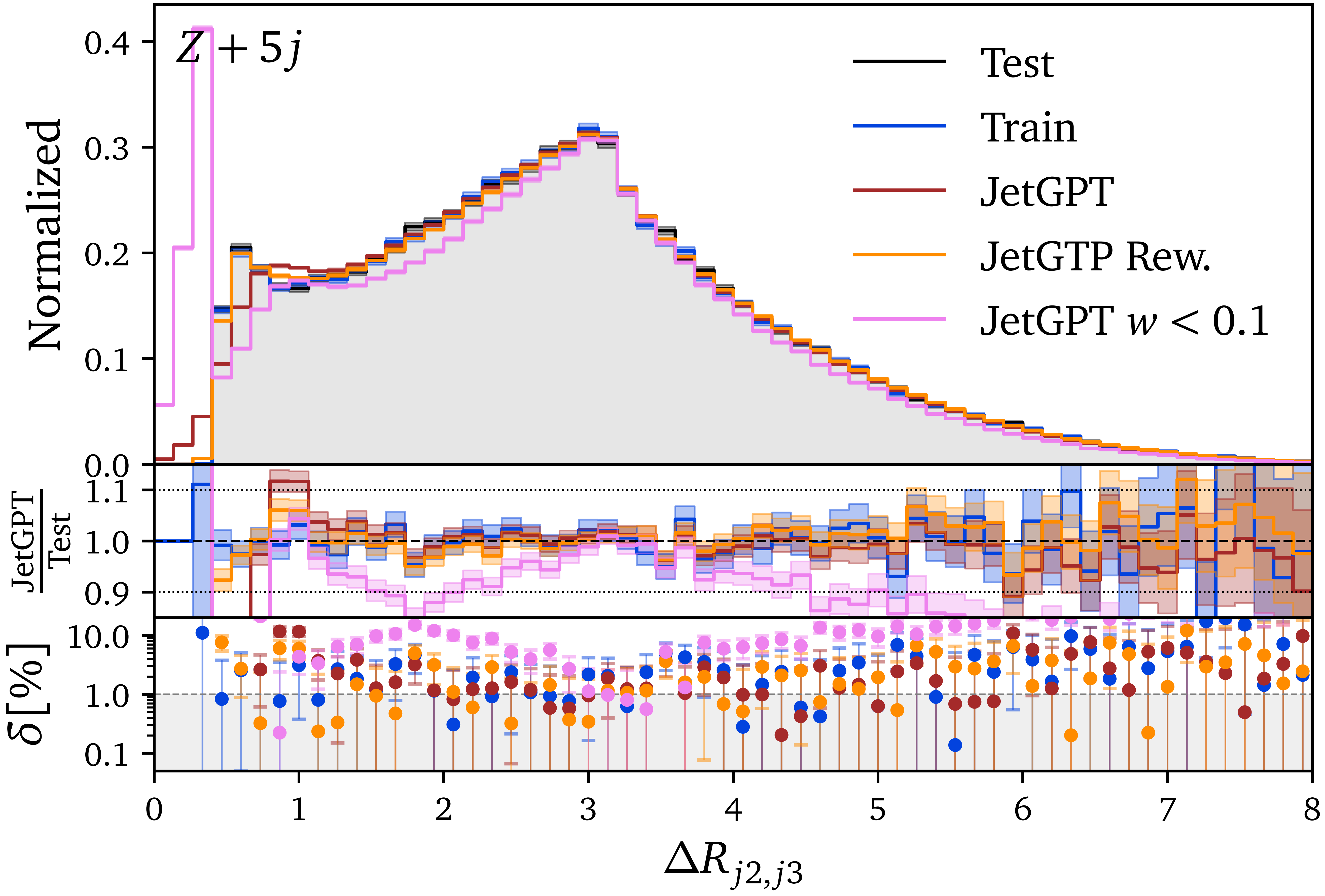


Classifier Reweighting

Track the limitations

$$w(x) = \frac{p_{\text{data}}(x)}{p_{\text{model}}(x)}$$

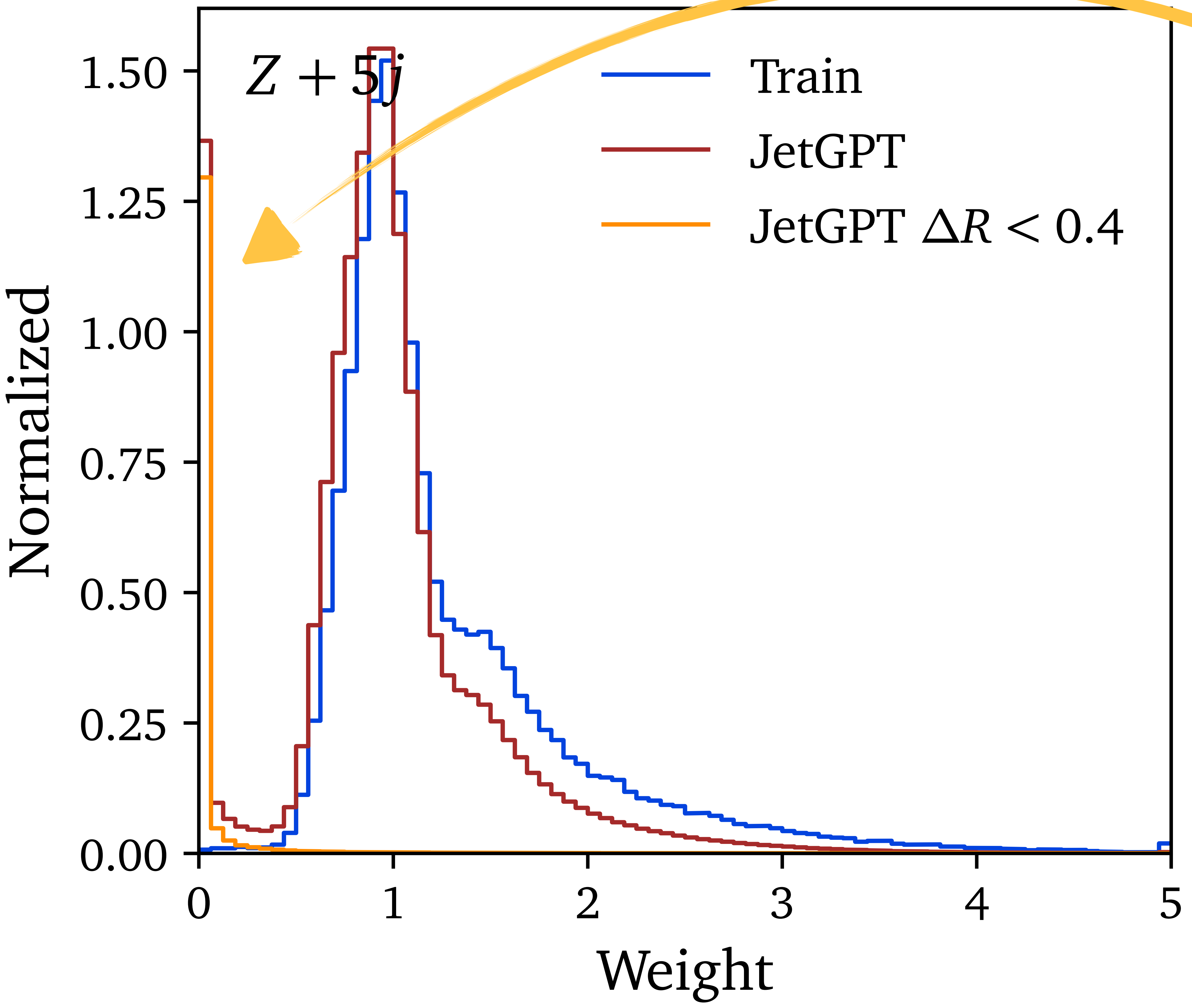


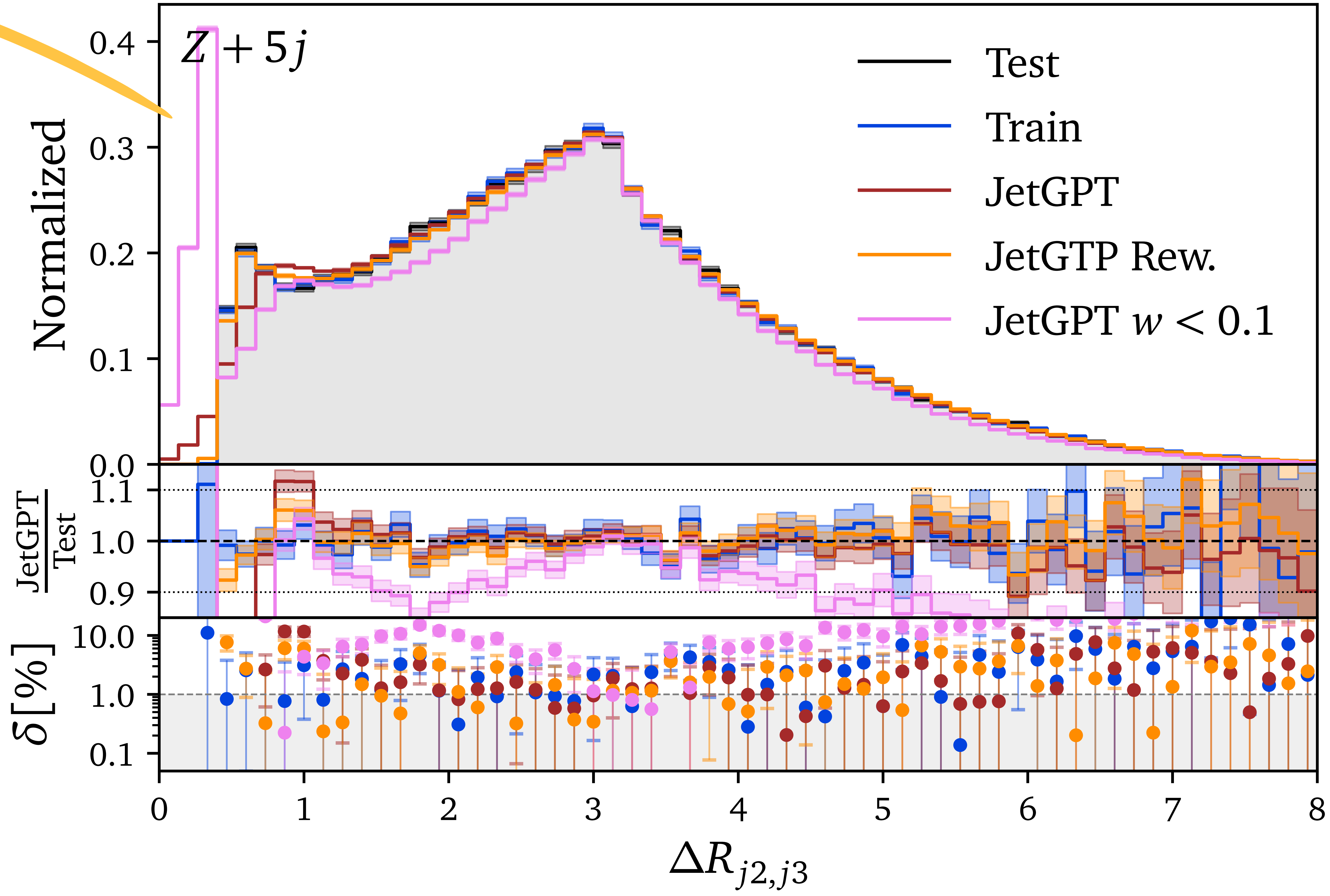


Classifier Reweighting

Track the limitations

$$w(x) = \frac{p_{\text{data}}(x)}{p_{\text{model}}(x)}$$

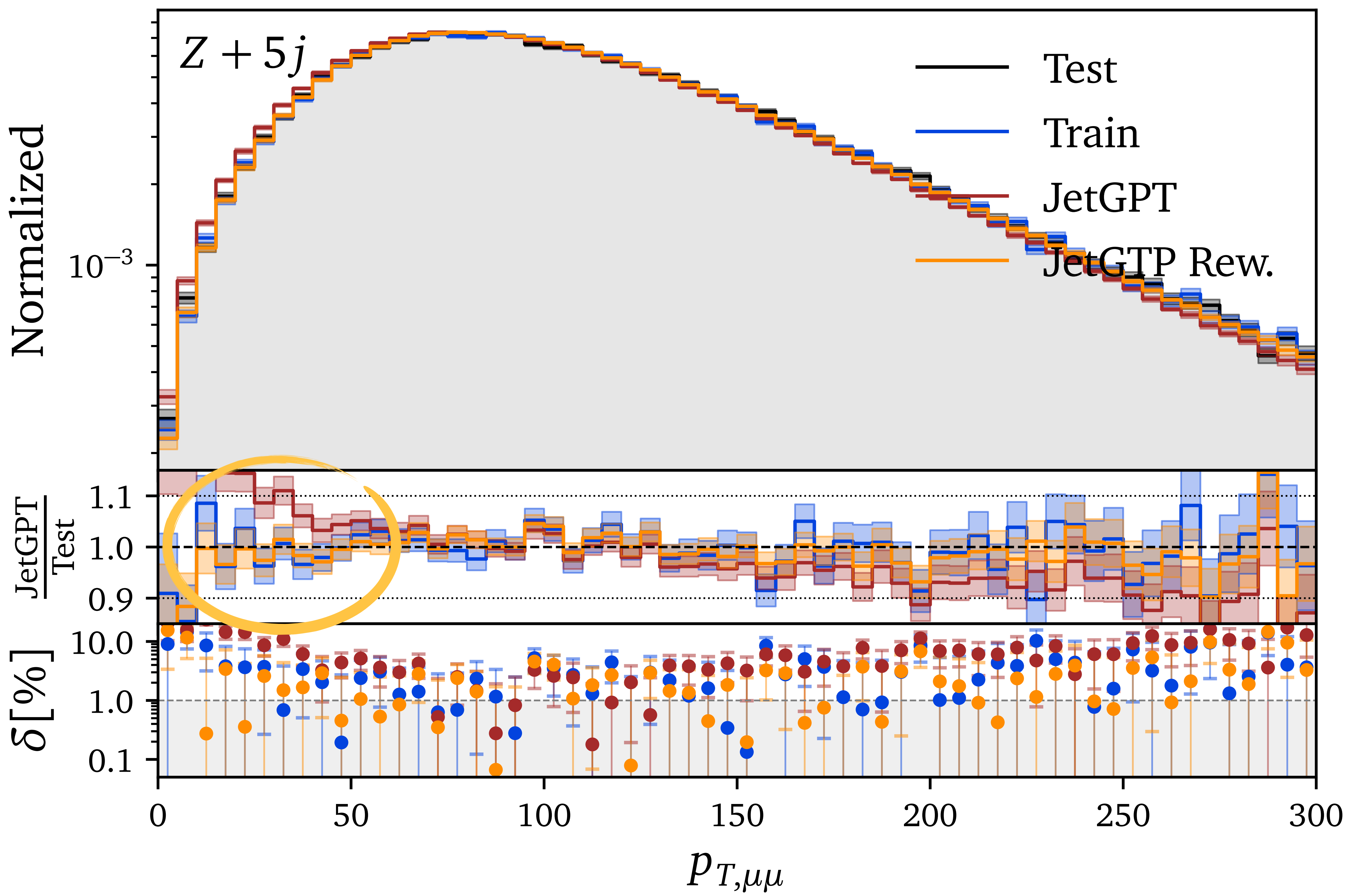


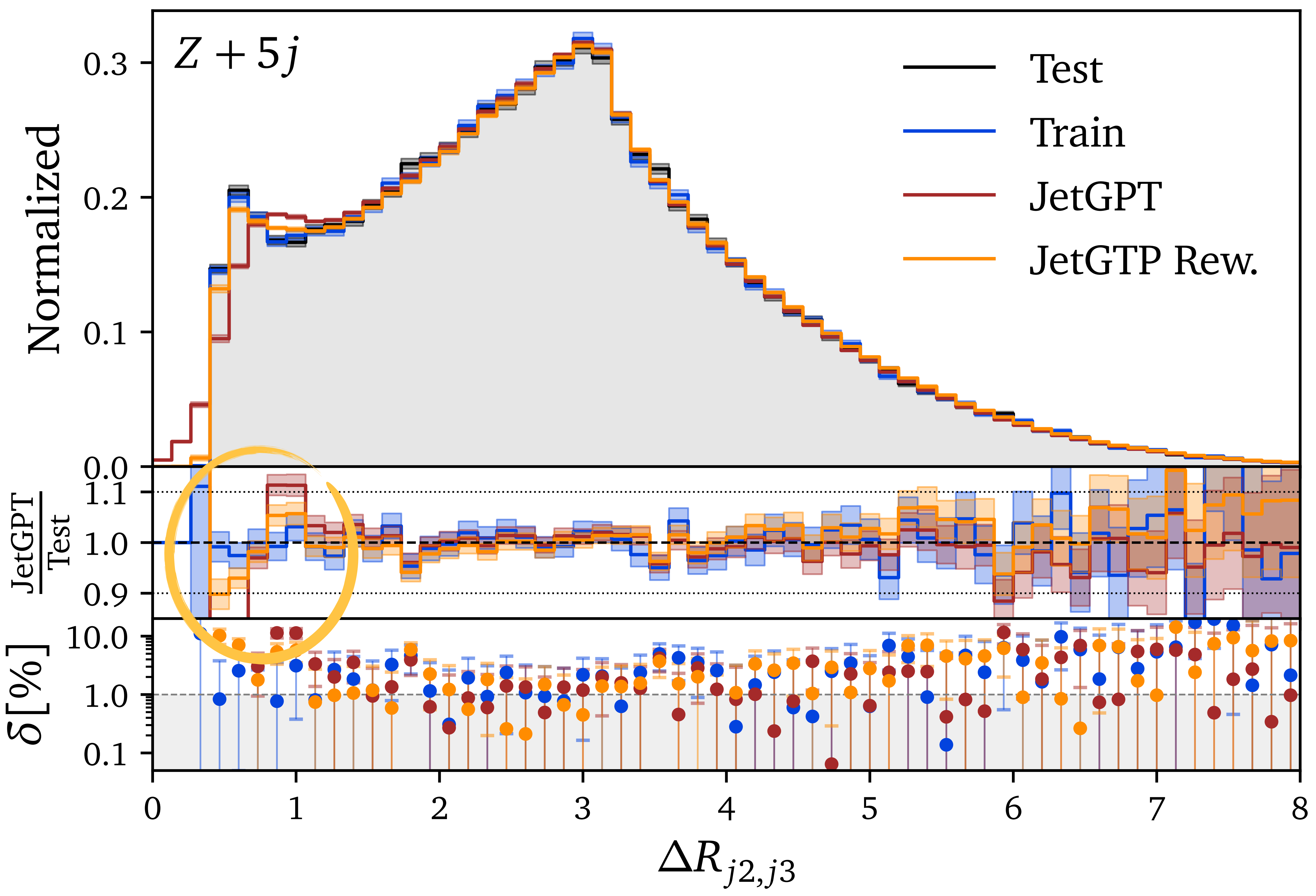


Classifier Reweighting

Overcome the limitations

$$p_{\text{data}} = p_{\text{model}} \times \frac{p_{\text{data}}}{p_{\text{model}}}$$





Conclusions

- Neural Networks can generate LHC events with **percent-level** accuracy
- **Autoregressive ordering** as powerful handle to provide implicit bias
- Transformers can be **trained jointly** on high-multiplicity datasets
- Neural Classifiers can **find and reweight** remaining discrepancies

Backup

Generating LHC Events

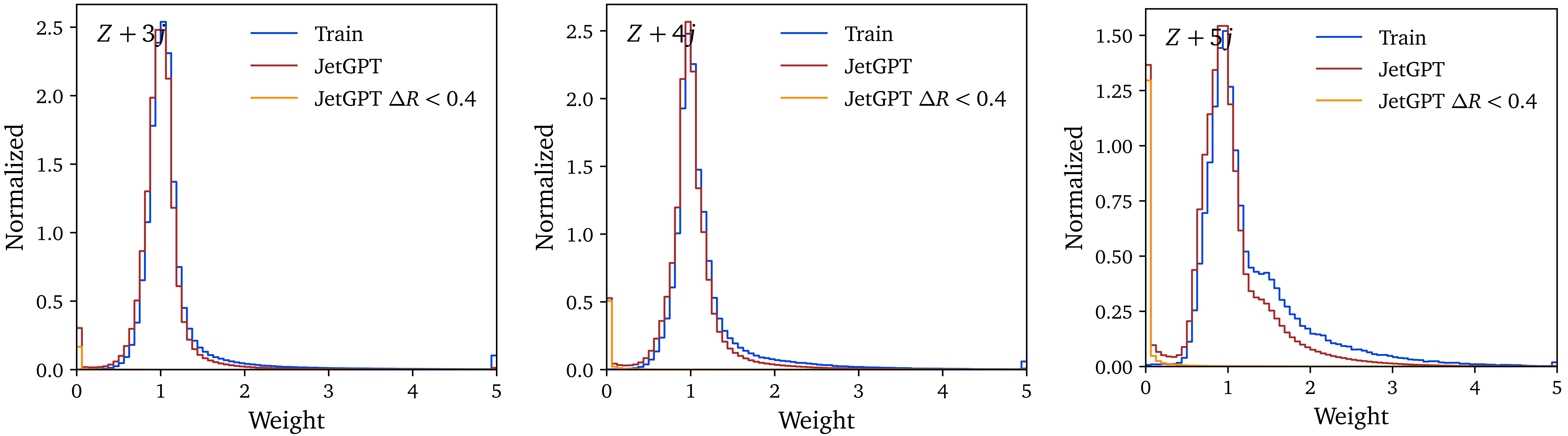
Preprocessing

$$p^\mu \Rightarrow \begin{matrix} p_T \\ \phi \\ \eta \\ m \end{matrix} \Rightarrow \begin{matrix} \log(p_T - p_{T,\min}) \\ \operatorname{arctanh}\frac{\phi}{\pi} \\ \eta \\ \log m \end{matrix} = y_i \Rightarrow x_i = \frac{y_i - \bar{y}_i}{\sigma_{y_i}}$$

Classifier Reweighting

Track the limitations

$$w(x) = \frac{p_{\text{data}}(x)}{p_{\text{model}}(x)}$$



$Z+3j$
$Z+5j$
Test
Train
JetGPT
JetGTP Rew.
Normalized
$\frac{\text{JetGPT}}{\text{Test}}$
$\delta[\%]$
$\Delta R_{j2,j3}$
$\eta_{j2}-\eta_{j3}$
$p_{T,\mu\mu}$
$\phi_{j2}-\phi_{j3}$
$p_{T,l1}$
$p_{T,l2}$
$\Delta R_{j4,j5}$