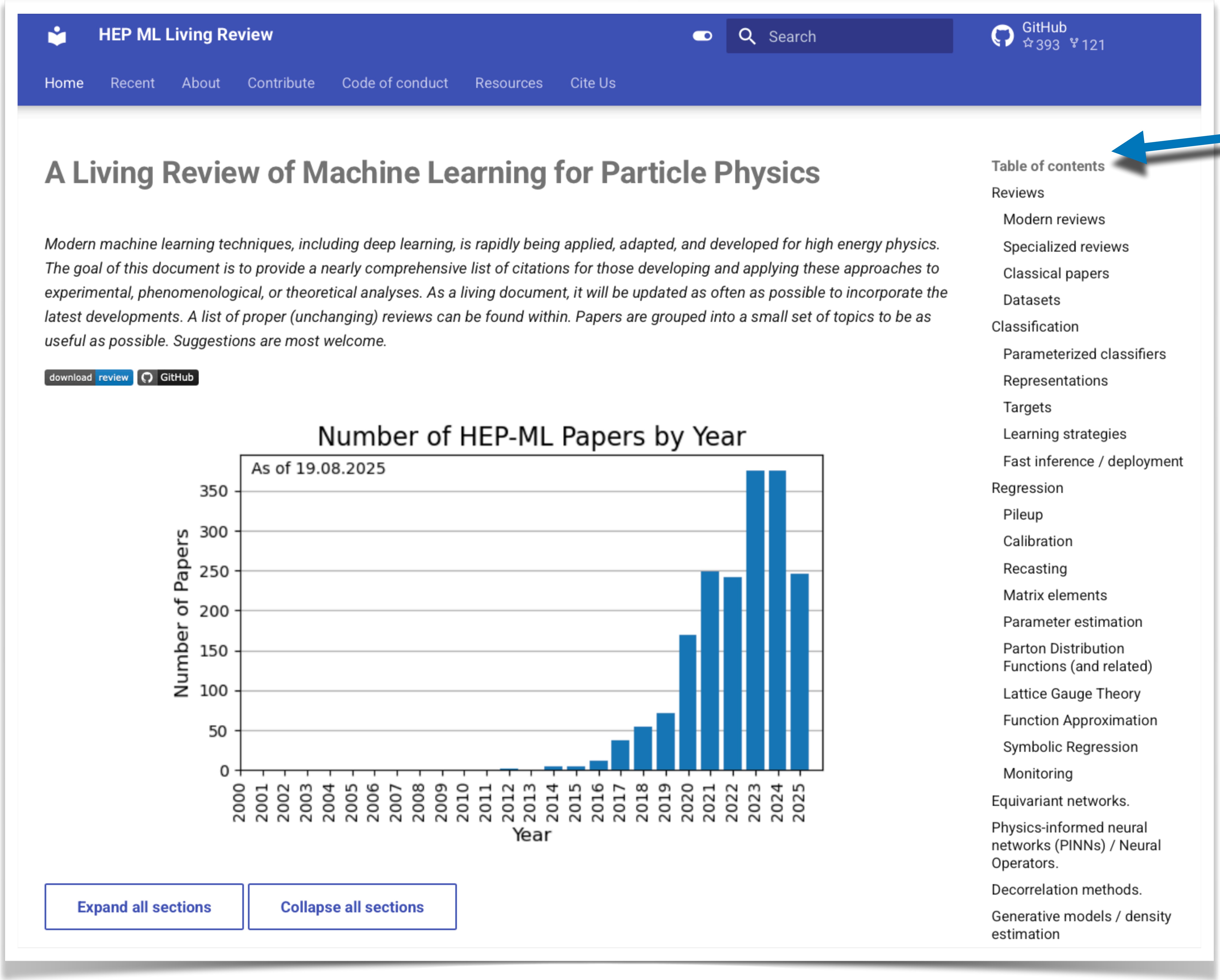
Deep Generative Models

UNIVERSITÀ DEGLI STUDI DI MILANO
INFN MILANO

IWR Summer School | Heidelberg

Ramon Winterhalder

HEP ML Living Review


HEP ML Living Review
Search
GitHub ☆393 ⑂121

Home Recent About Contribute Code of conduct Resources Cite Us

A Living Review of Machine Learning for Particle Physics

Modern machine learning techniques, including deep learning, is rapidly being applied, adapted, and developed for high energy physics. The goal of this document is to provide a nearly comprehensive list of citations for those developing and applying these approaches to experimental, phenomenological, or theoretical analyses. As a living document, it will be updated as often as possible to incorporate the latest developments. A list of proper (unchanging) reviews can be found within. Papers are grouped into a small set of topics to be as useful as possible. Suggestions are most welcome.

download review | GitHub

Expand all sections | Collapse all sections

Table of contents

- Check LivingReview for many **ML4HEP applications**

HEPML-Review

Plan of attack

1. Basics of DGMs and Normalizing Flows

2. GANs, VAE and SurVAE

3. Diffusion Models

Blackboard Session I

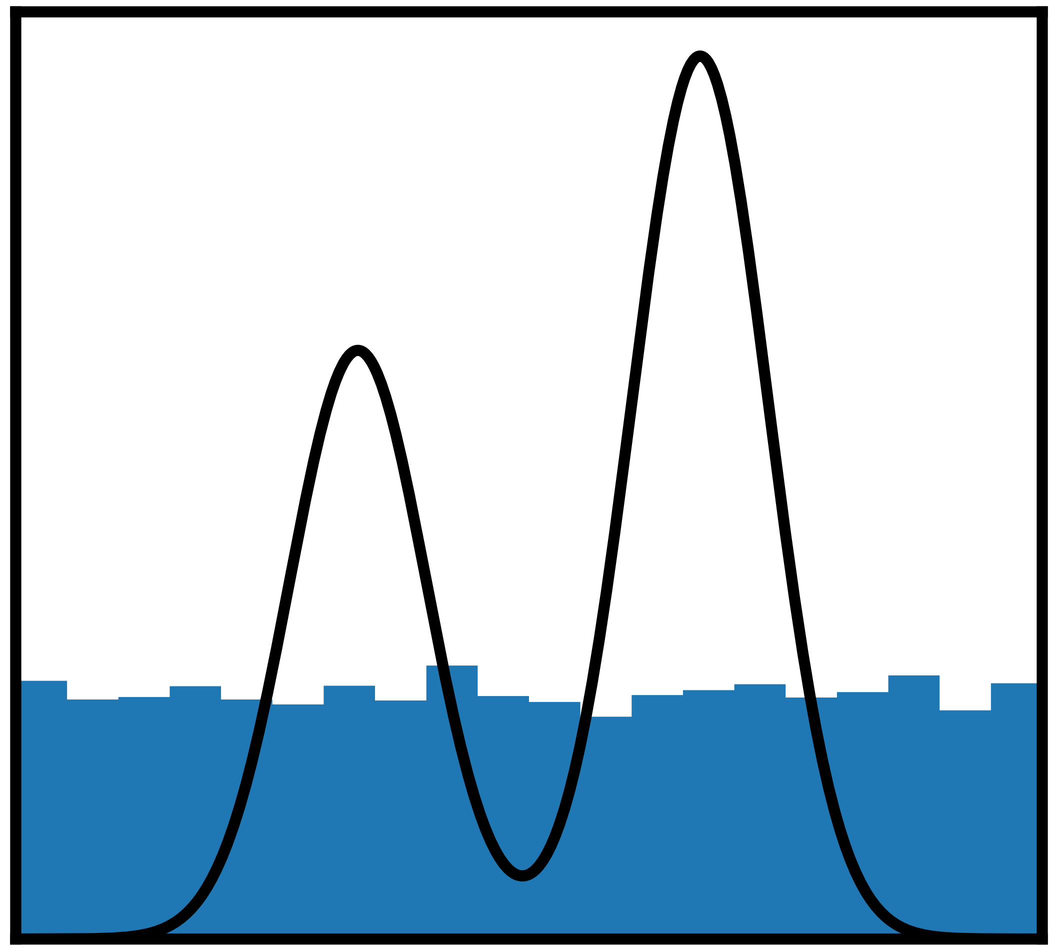
Normalizing Flows

Neural Importance Sampling

Monte Carlo integration

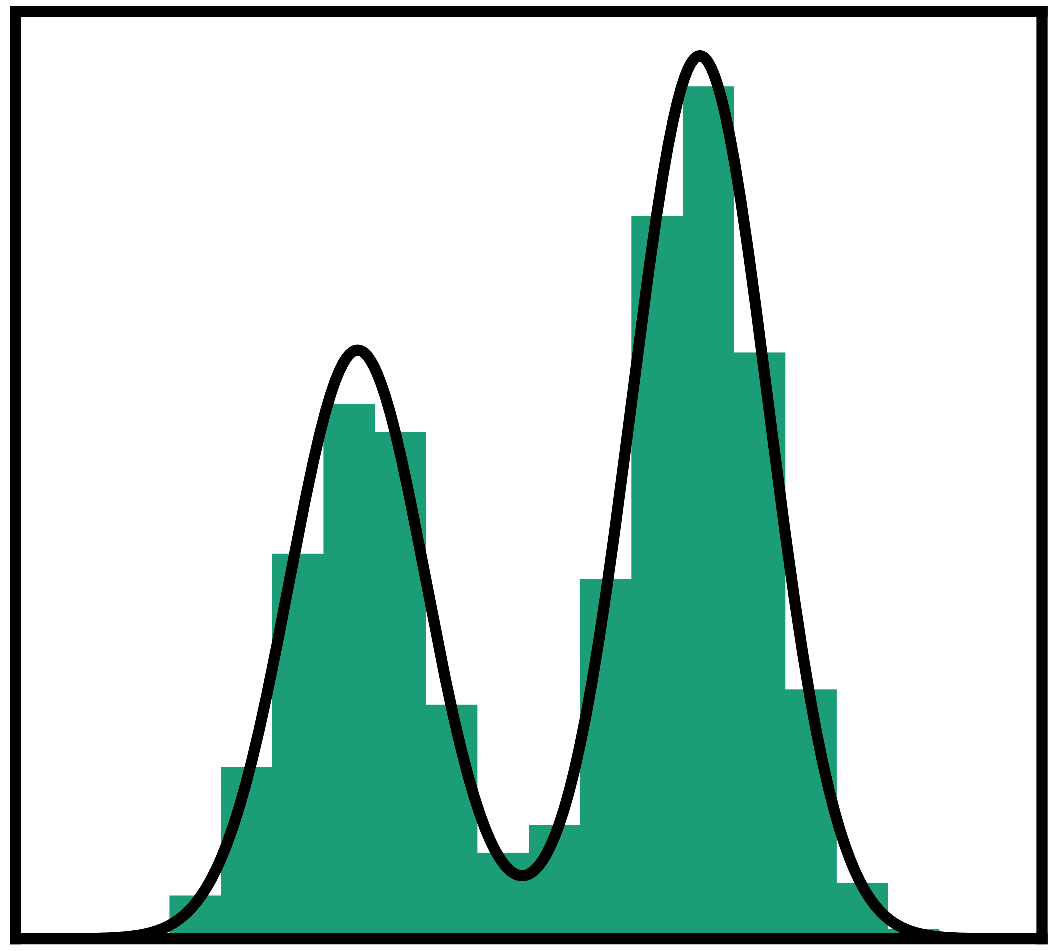
Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \mid p_1, \ldots, p_n)|^2 \,\right\rangle$$



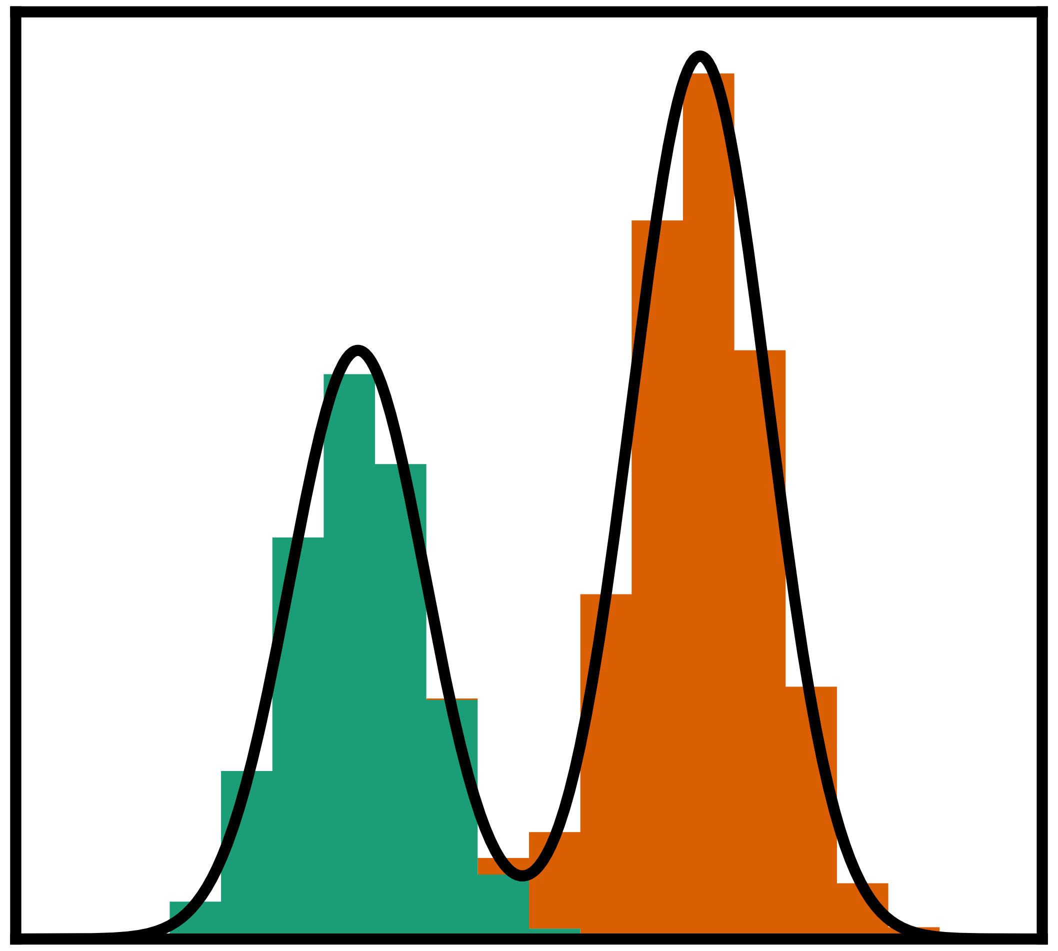
Flat sampling:
inefficient

$$I = \langle f(x) \rangle_{x \sim \text{unif}}$$



Importance sampling:
find p close to f

$$I = \left\langle \frac{f(x)}{p(x)} \right\rangle_{x \sim p(x)}$$



Multi-channel:
one map for each channel

$$I = \sum_i \left\langle \alpha_i(x) \frac{f(x)}{p_i(x)} \right\rangle_{x \sim p_i(x)}$$

Event generation in MadGraph

Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \mid p_1, \ldots, p_n)|^2 \,\right\rangle$$

$$I = \sum_i \left\langle \alpha_i(x) \frac{f(x)}{p_i(x)} \right\rangle_{x \sim p_i(x)}$$

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

Channel weights

MadGraph: $\alpha_i^{\text{MG}}(x) \sim |M_i|^2$

Channel mappings

MadGraph: use amplitude structure, …

Analytic mappings + refine with VEGAS (factorized, histogram based importance sampling)

Event generation in MadNIS

Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \,|\, p_1, \ldots, p_n)|^2 \,\right\rangle$$

MadNIS

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

$$I = \sum_i \left\langle \alpha_i(x)\, \frac{f(x)}{p_i^{\omega}(x)} \right\rangle_{x \sim p_i^{\omega}(x)}$$

Channel weights

MadGraph: $\alpha_i^{\text{MG}}(x) \sim |M_i|^2$

Learned channel mappings

MadGraph: use amplitude structure, …

Analytic mappings + ~~refine with VEGAS~~

refine with NF

Event generation in MadNIS

Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b \, f(x_a) f(x_b) \, d\Phi_n \left\langle \, |M_{\lambda,c,\ldots}(p_a, p_b \mid p_1, \ldots, p_n)|^2 \, \right\rangle$$

MadNIS

$$I = \sum_i \left\langle \alpha_i^{\xi}(x) \frac{f(x)}{p_i^{\omega}(x)} \right\rangle_{x \sim p_i^{\omega}(x)}$$

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $d\sigma / dx$

Learned channel weights

MadGraph: $\alpha_i^{\mathrm{MG}}(x) \sim |M_i|^2$

$\alpha_i(x) \to \alpha_i^{\xi}(x) = \alpha_i^{\mathrm{MG}}(x) \cdot K_i^{\xi}(x)$

parametrize with **NN**

Learned channel mappings

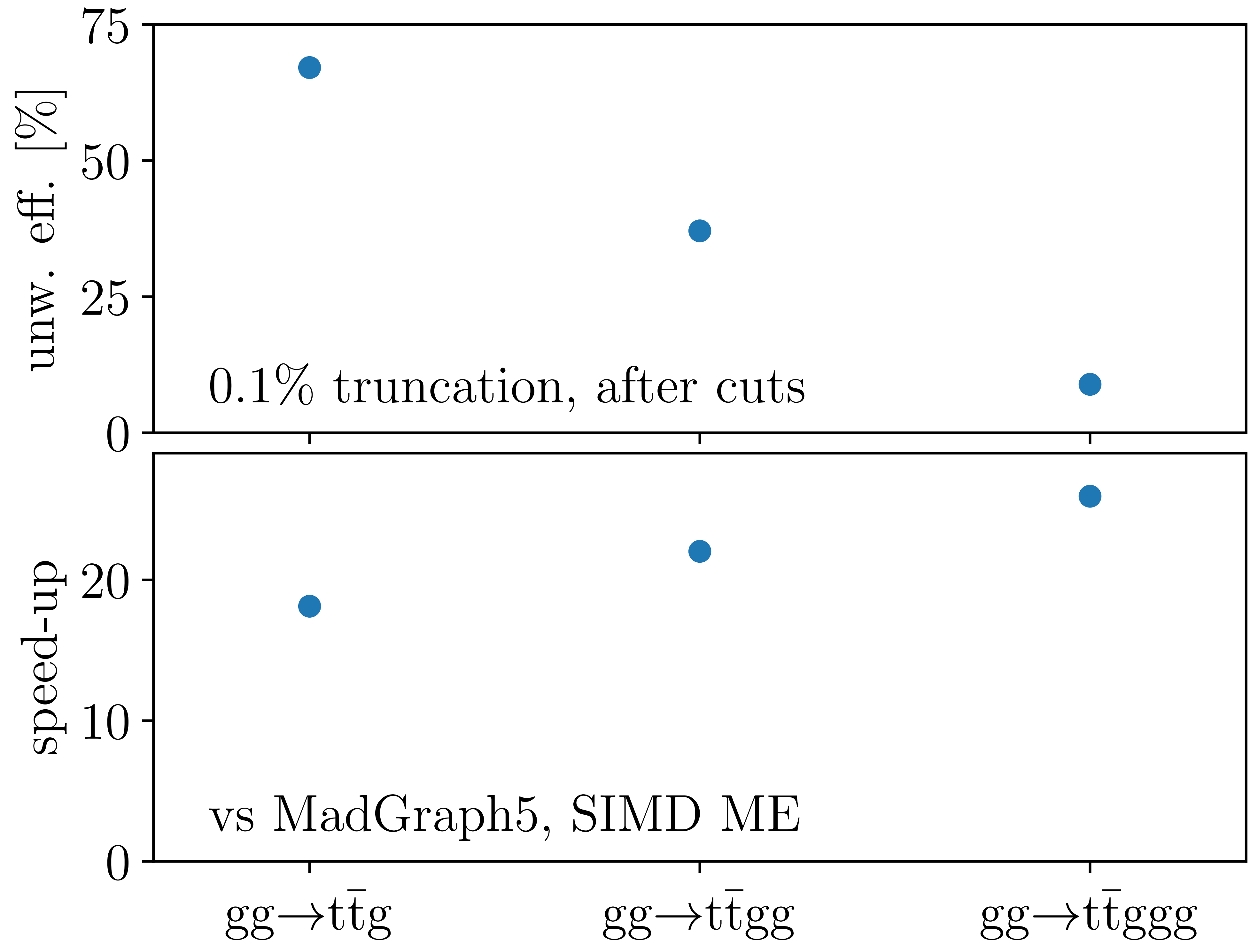
MadGraph: use amplitude structure, …

Analytic mappings + ~~refine with VEGAS~~

refine with **NF**

Unweighting performance

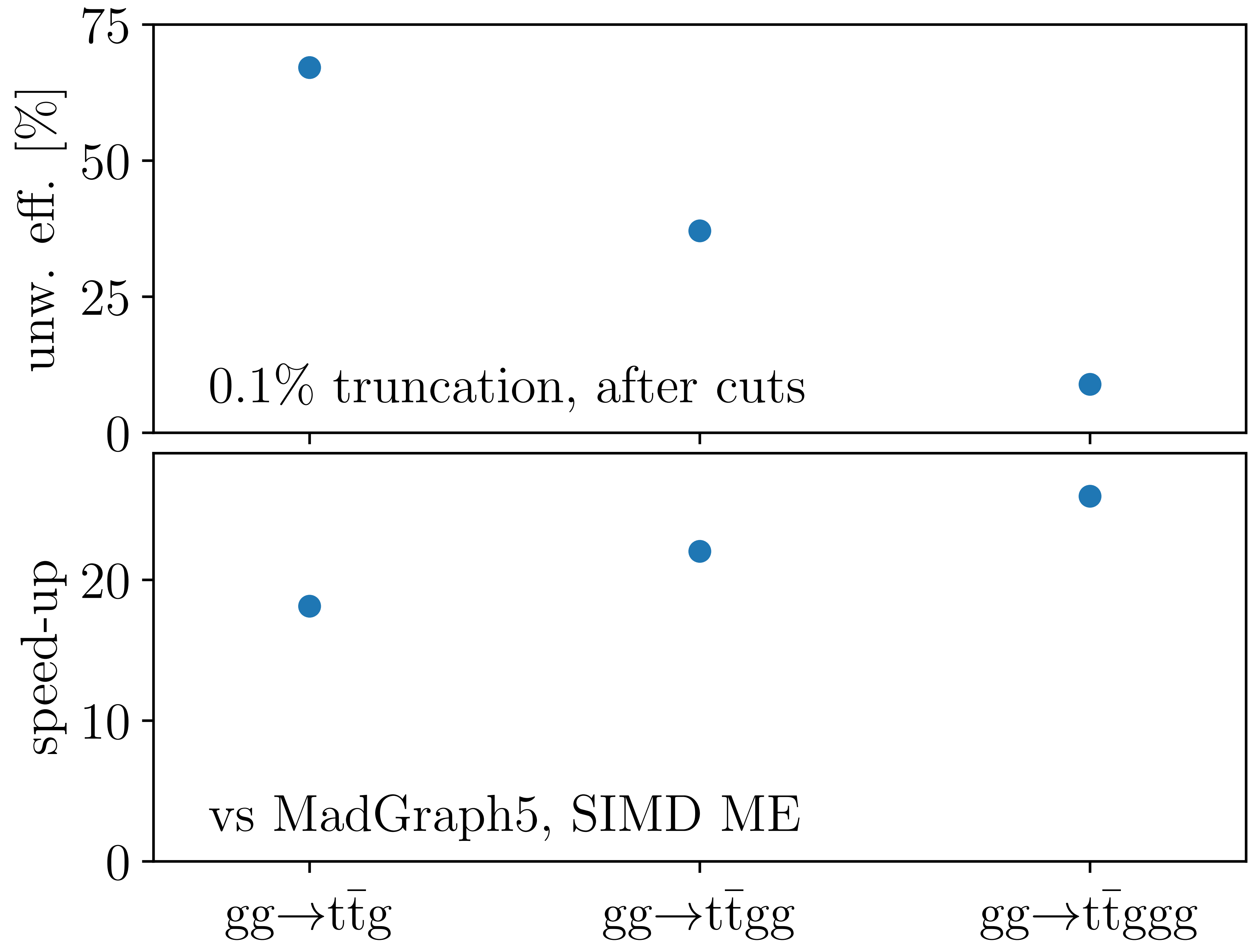
Unweighting efficiency



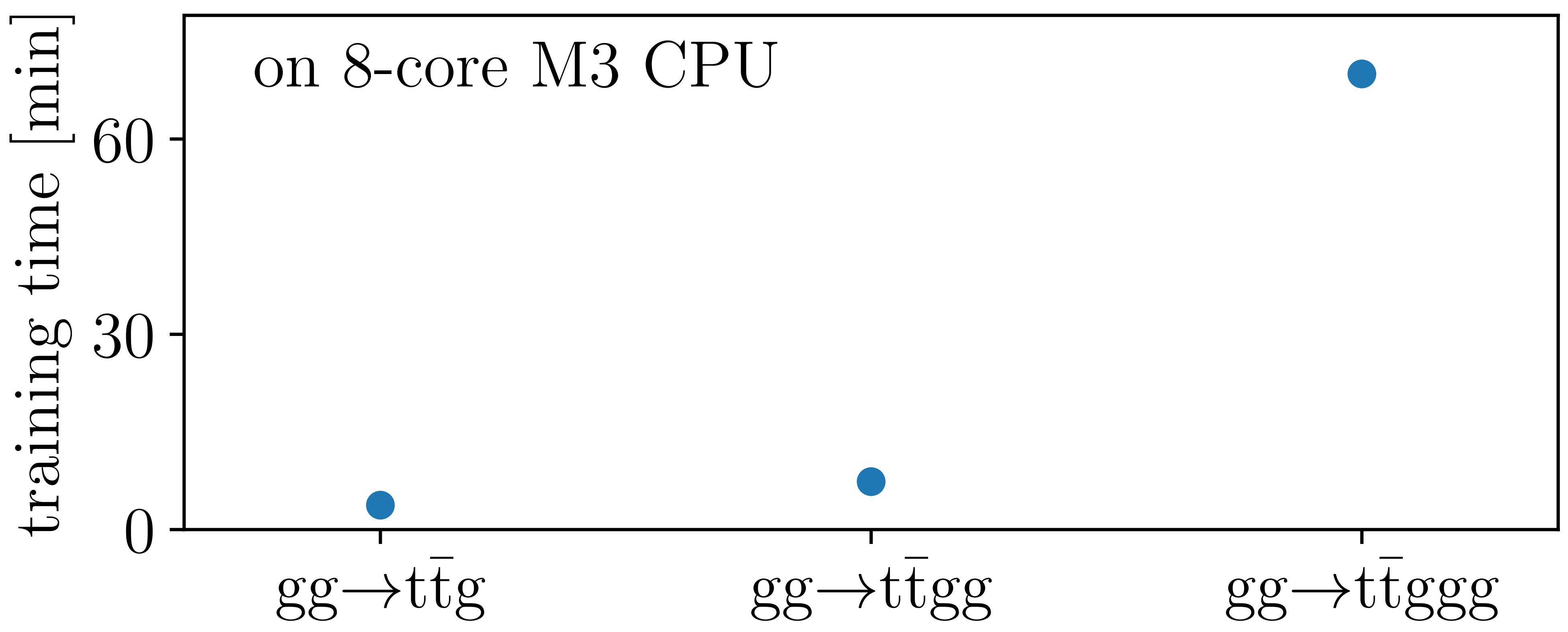
(preliminary)

Unweighting performance

Unweighting efficiency



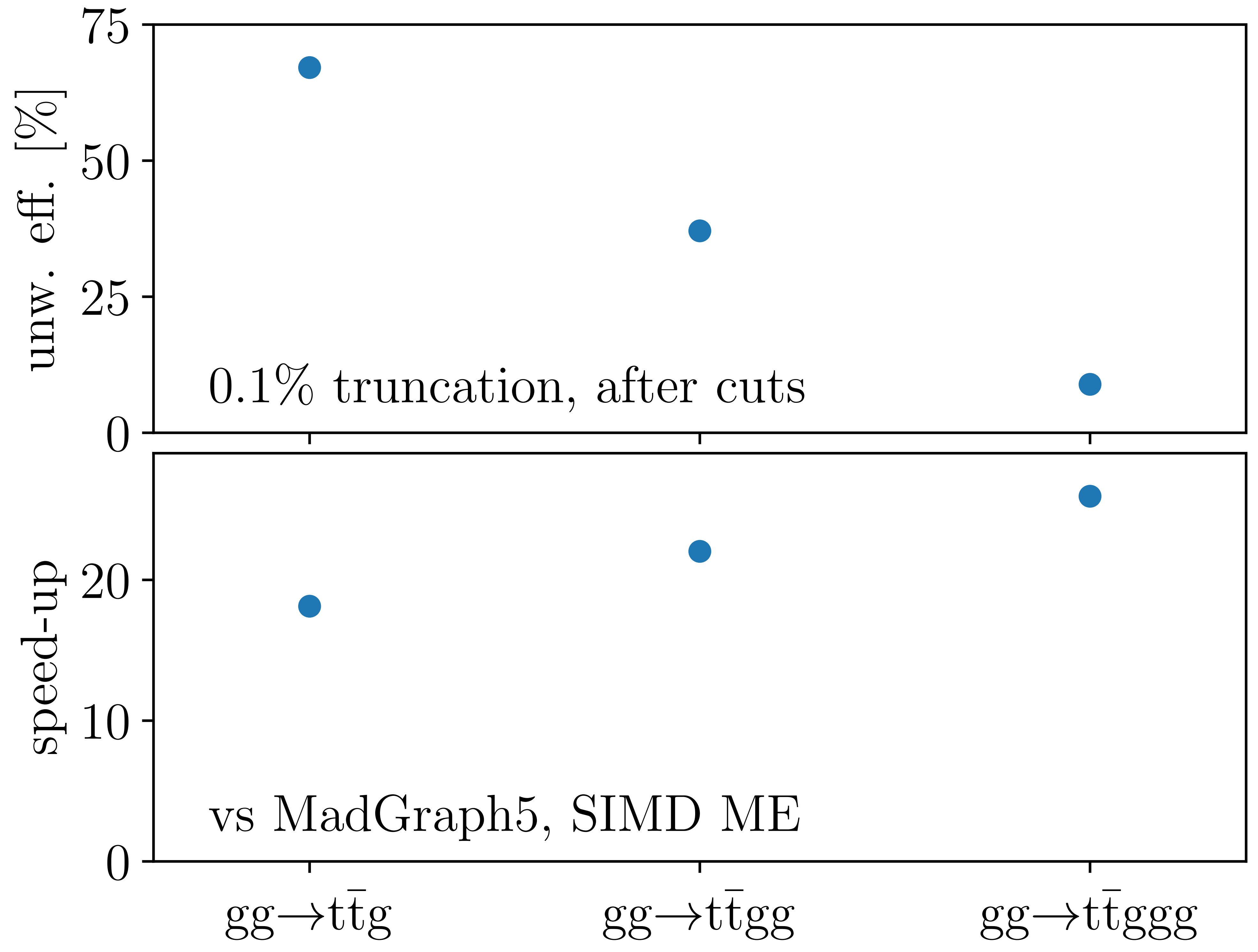
Training time



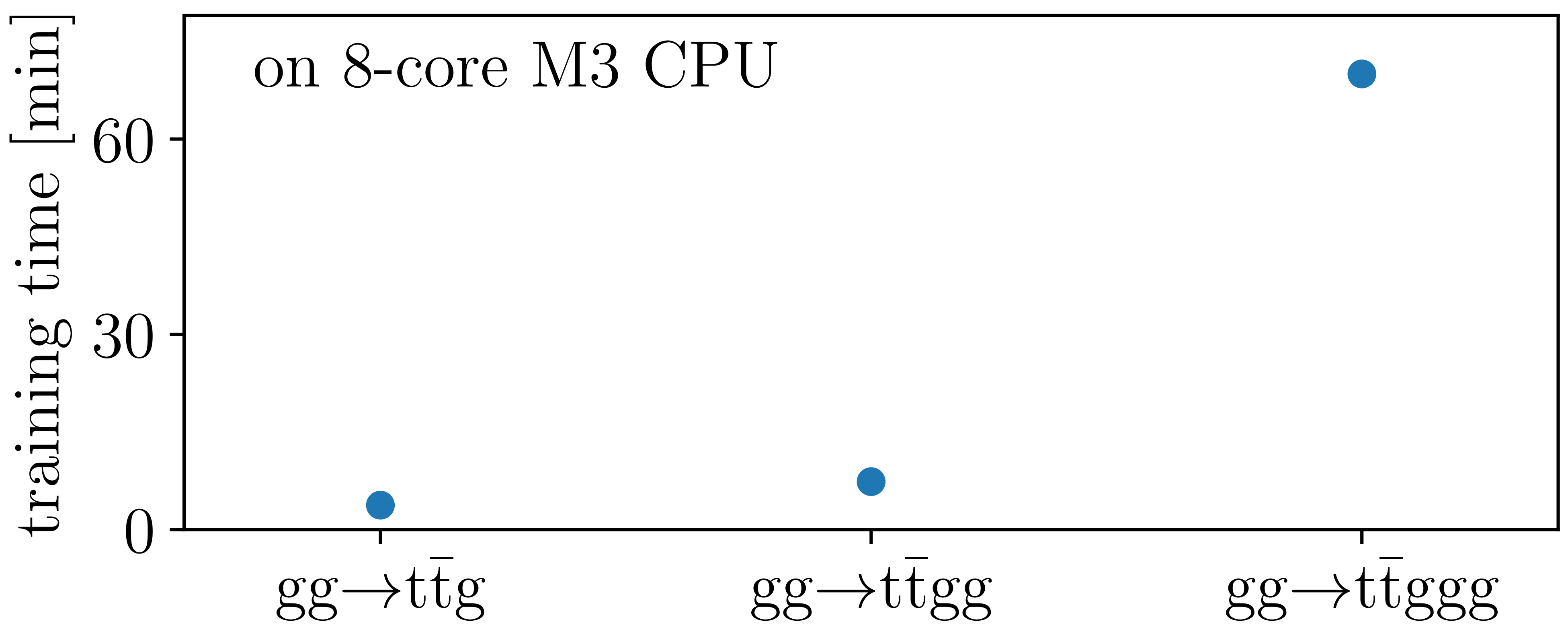
(preliminary)

Unweighting performance

Unweighting efficiency



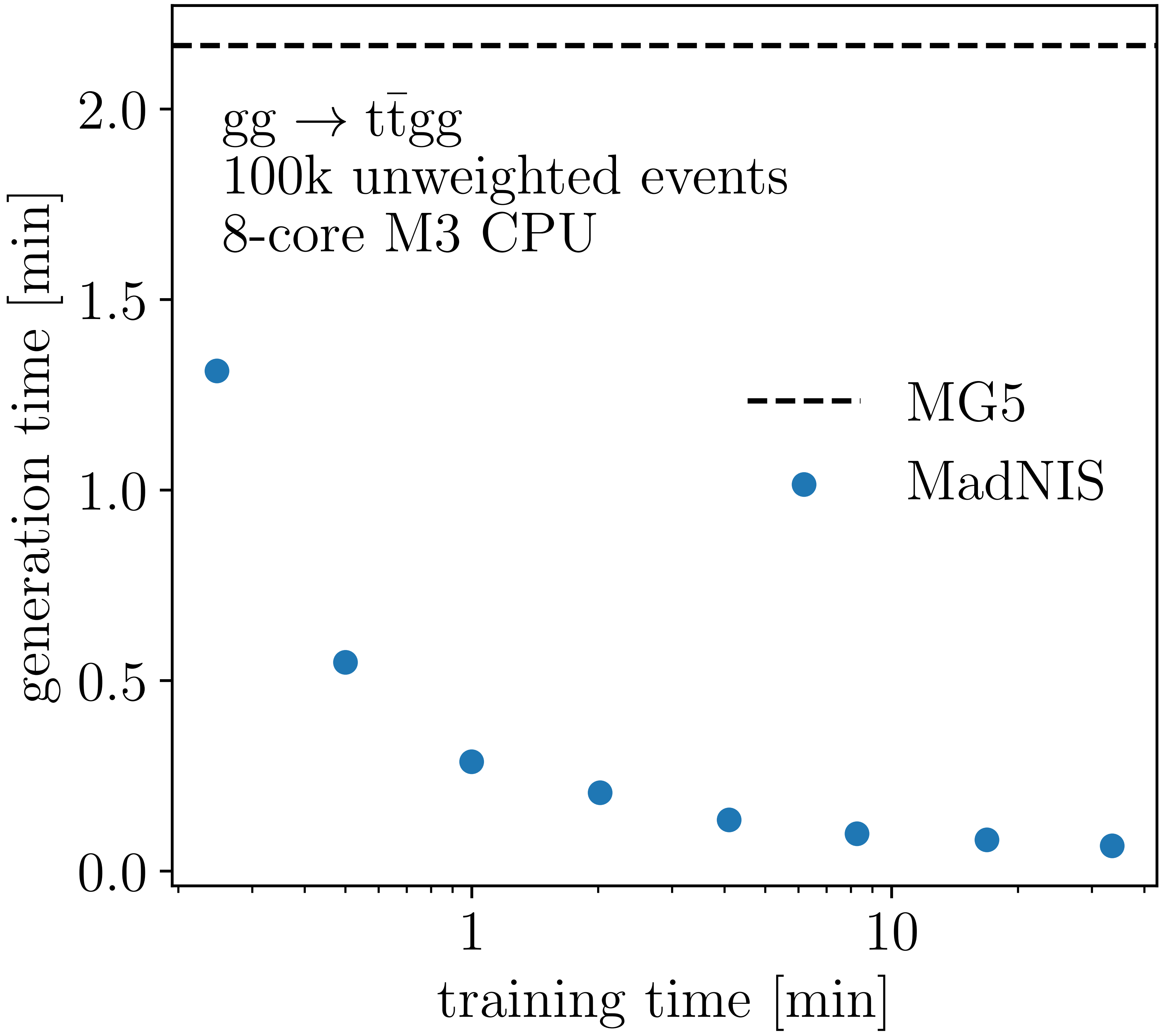
Training time

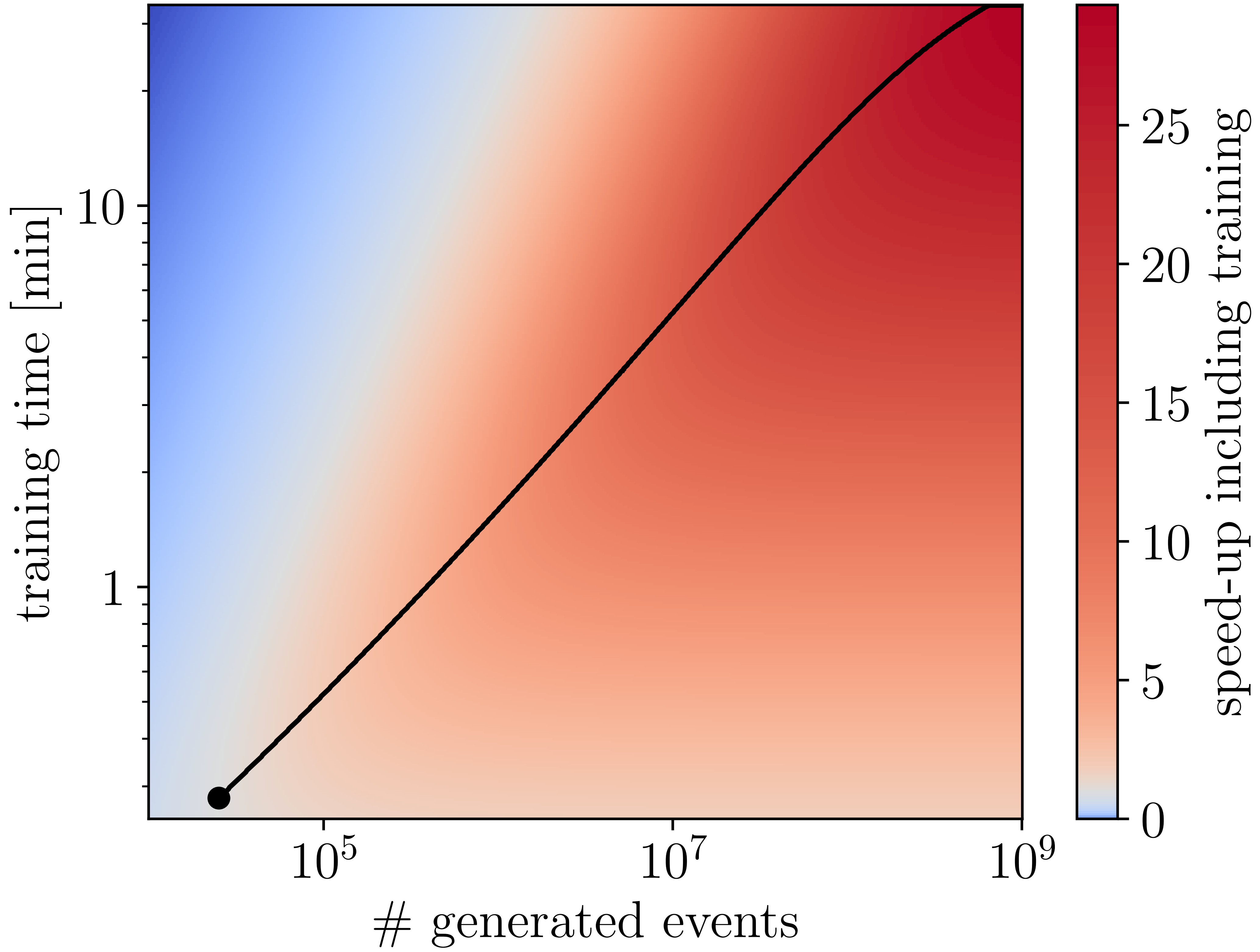


→ Does it still pay off?

(preliminary)

Training time and amortization



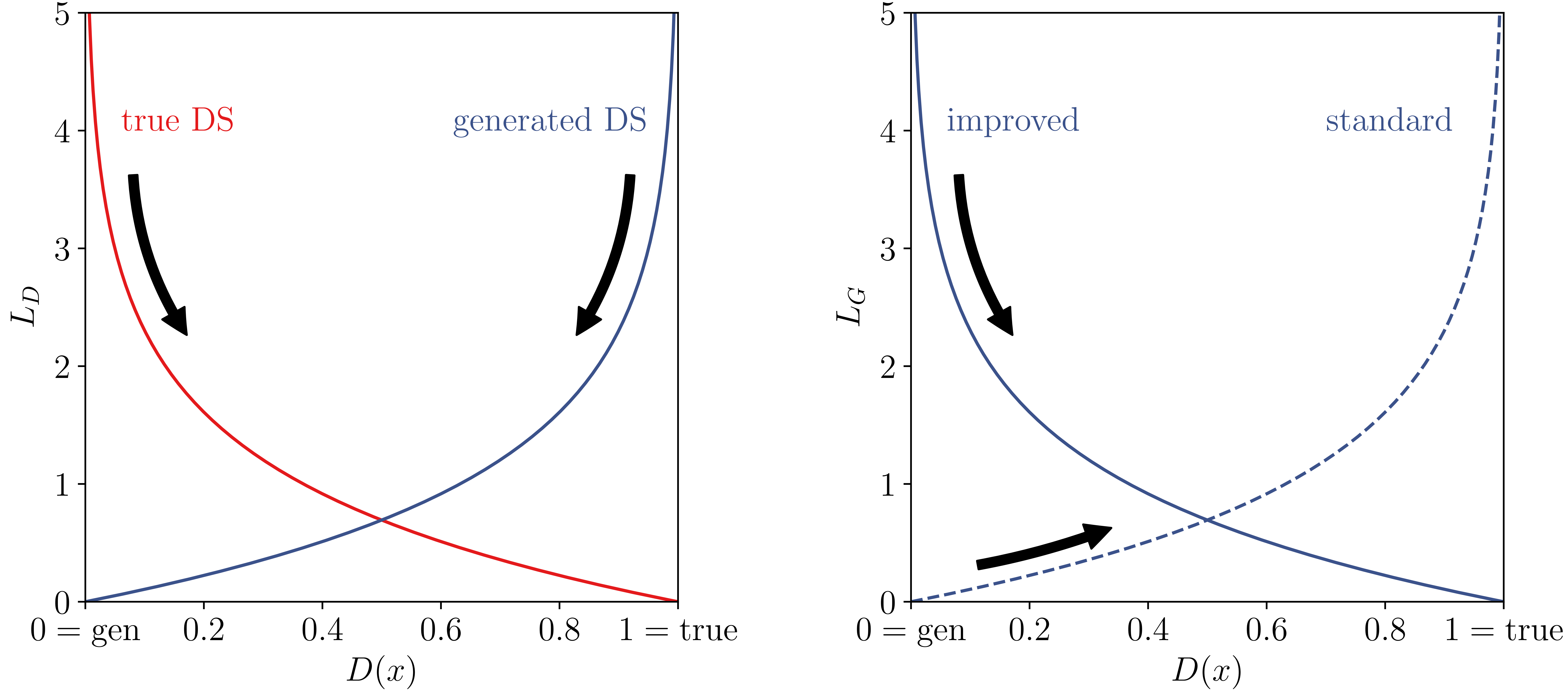


(preliminary)

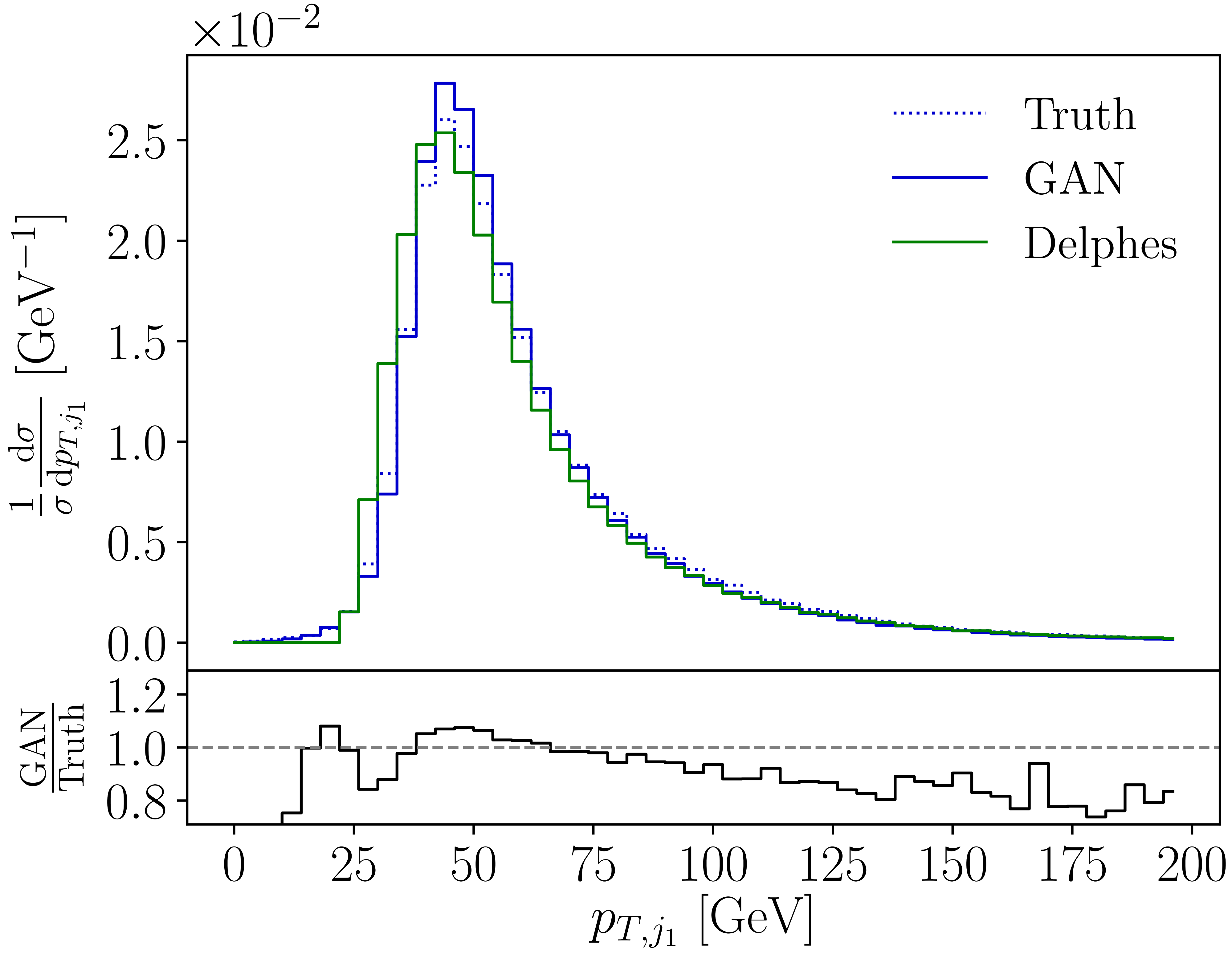
MadNIS is faster starting at 10k events!

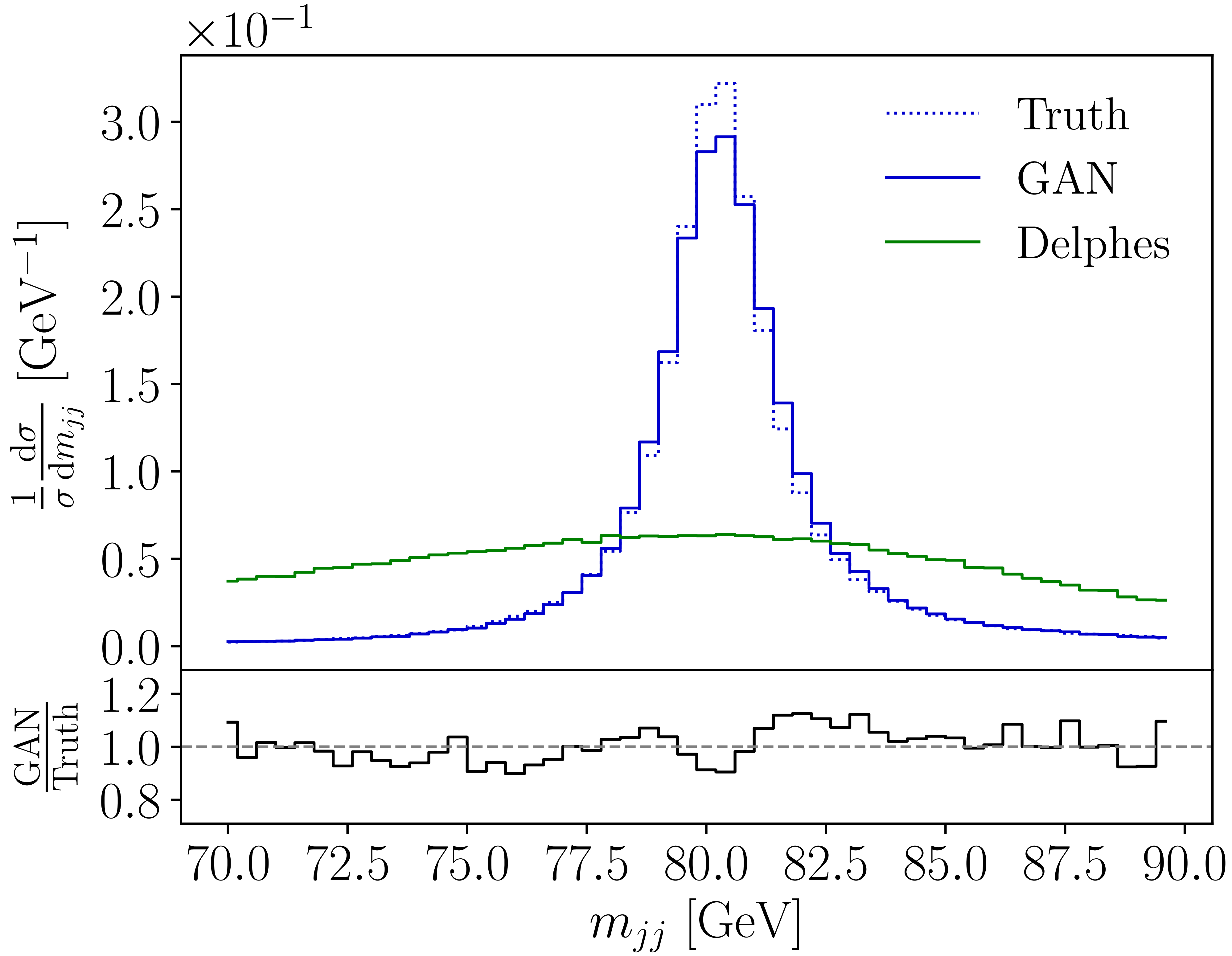
Blackboard Session II

GAN Loss

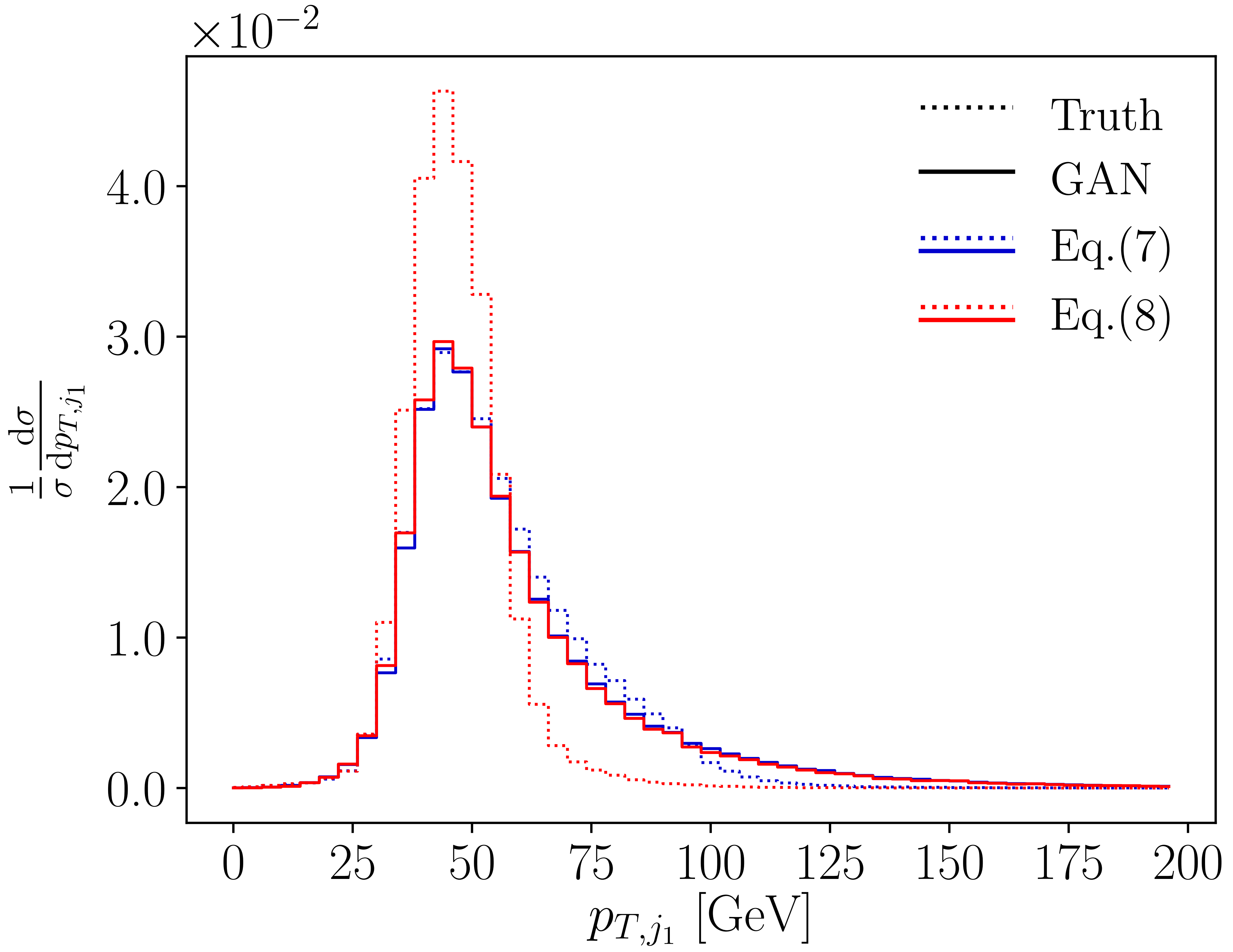

5
4
3
2
1
0
L_D
true DS
generated DS
0 = gen
0.2
0.4
0.6
0.8
1 = true
$D(x)$
5
4
3
2
1
0
L_G
improved
standard
0 = gen
0.2
0.4
0.6
0.8
1 = true
$D(x)$

GAN Unfolding





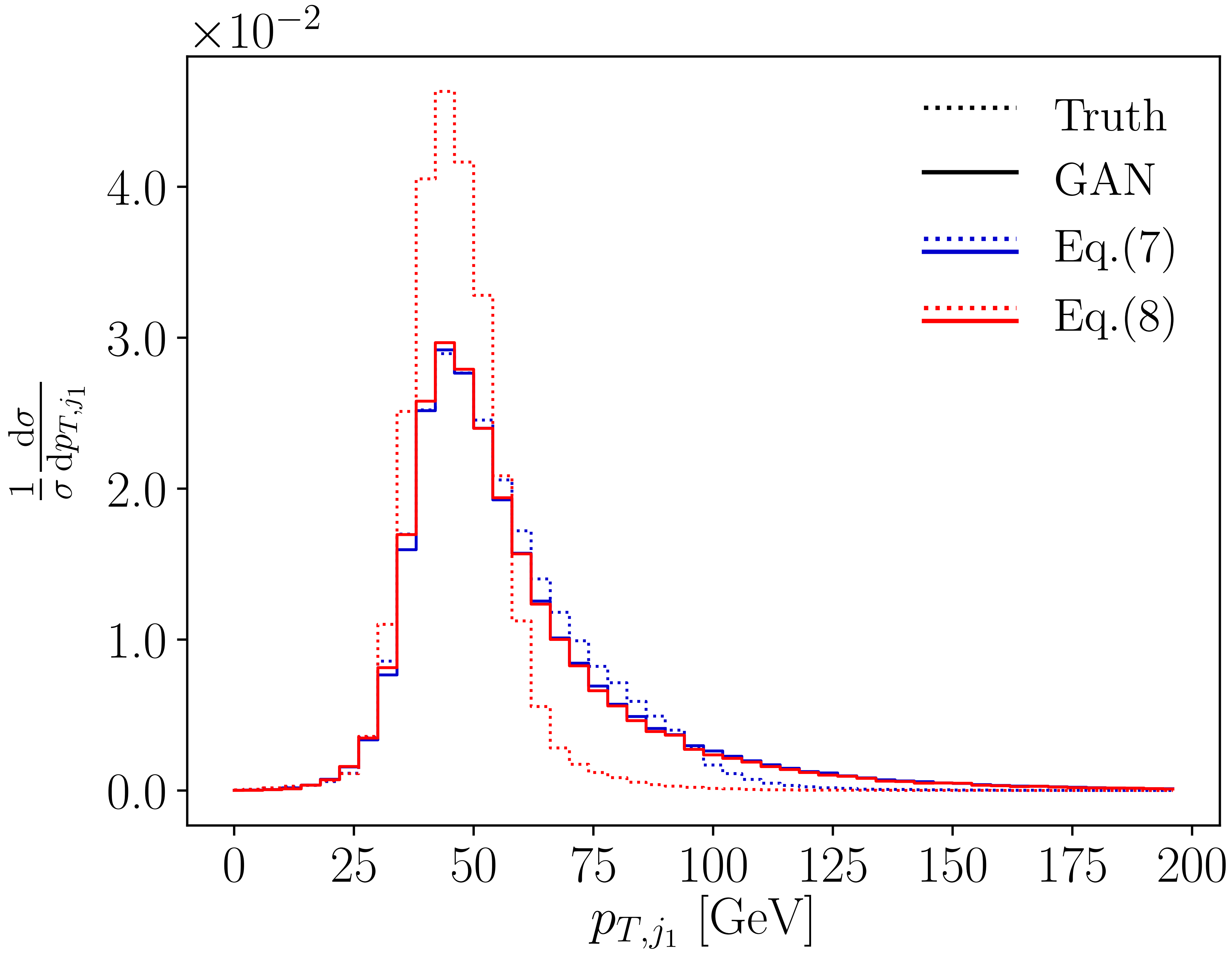
GAN Unfolding

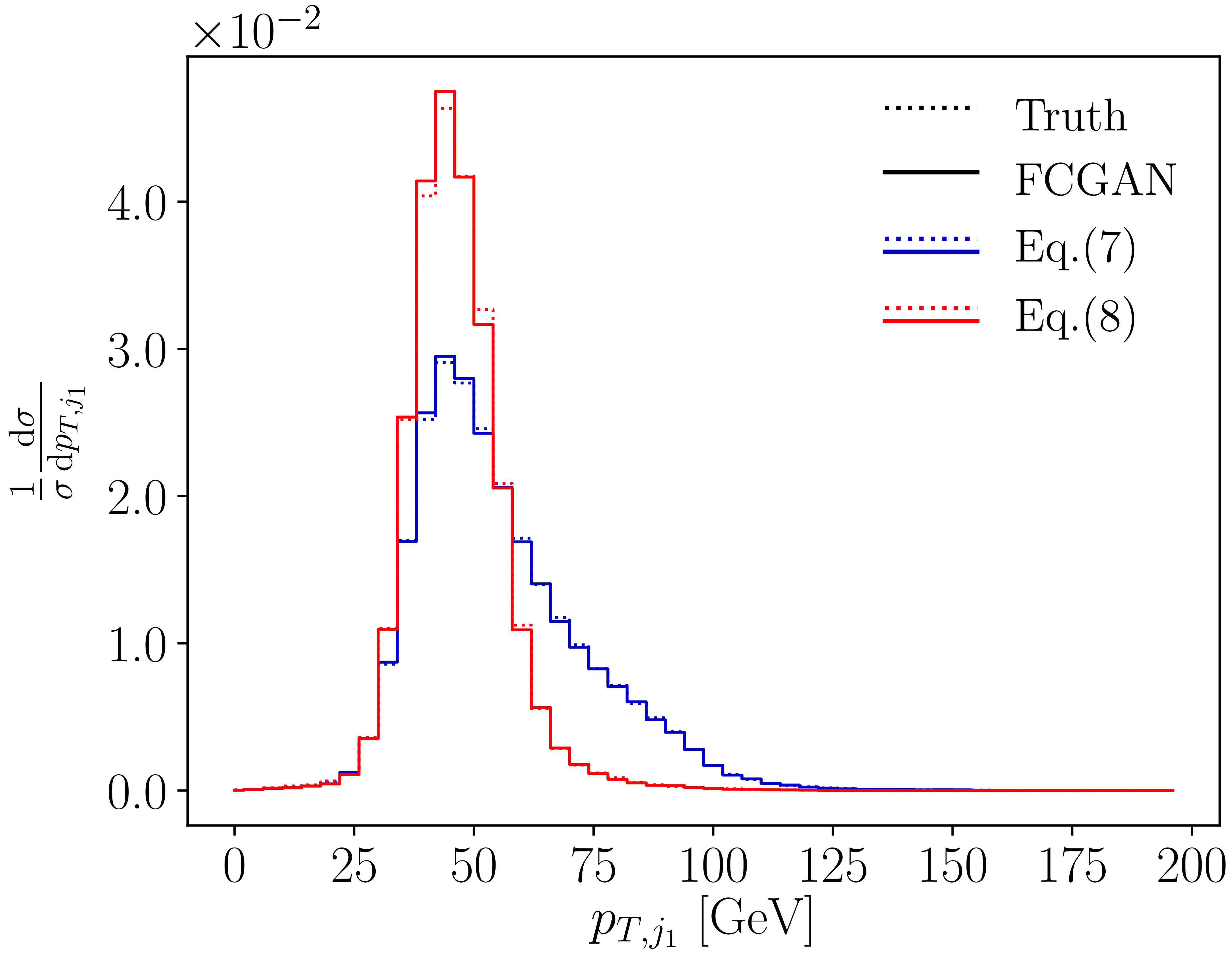


Cut I:
$30\,\text{GeV} < p_{T,j_1} < 100\,\text{GeV}$

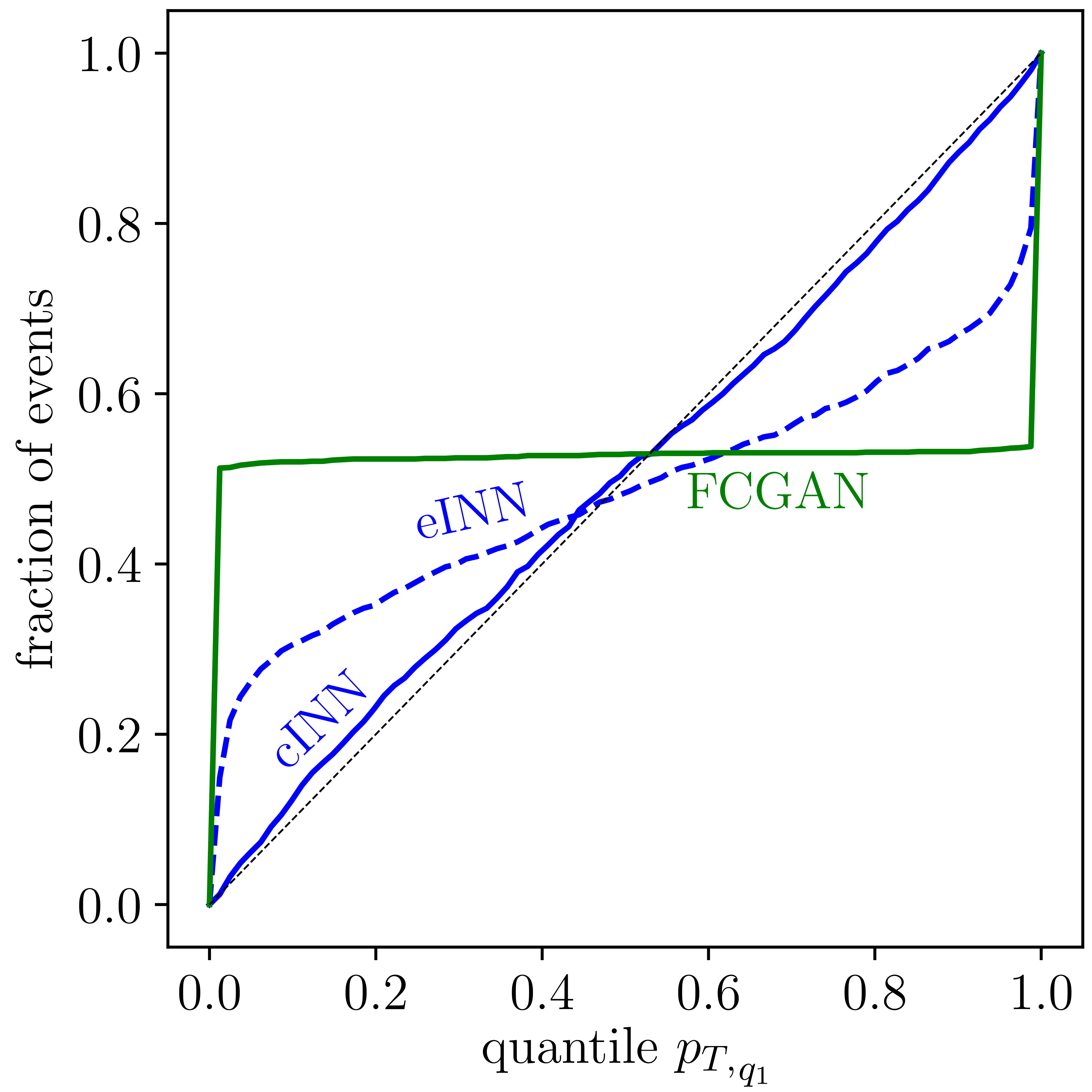
Cut II:
$30\,\text{GeV} < p_{T,j_1} < 50\,\text{GeV}$
$30\,\text{GeV} < p_{T,j_2} < 60\,\text{GeV}$

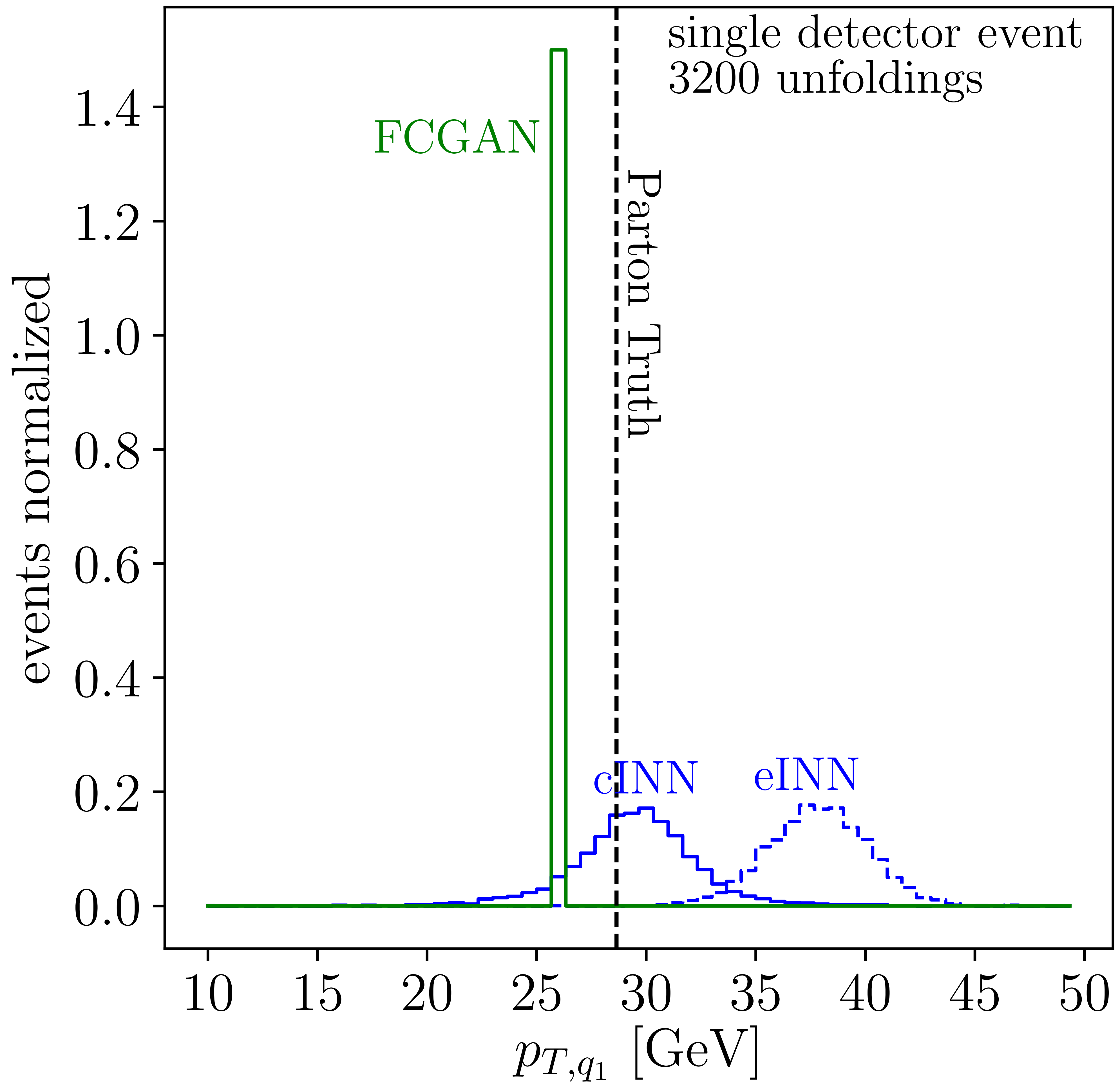
FCGAN Unfolding





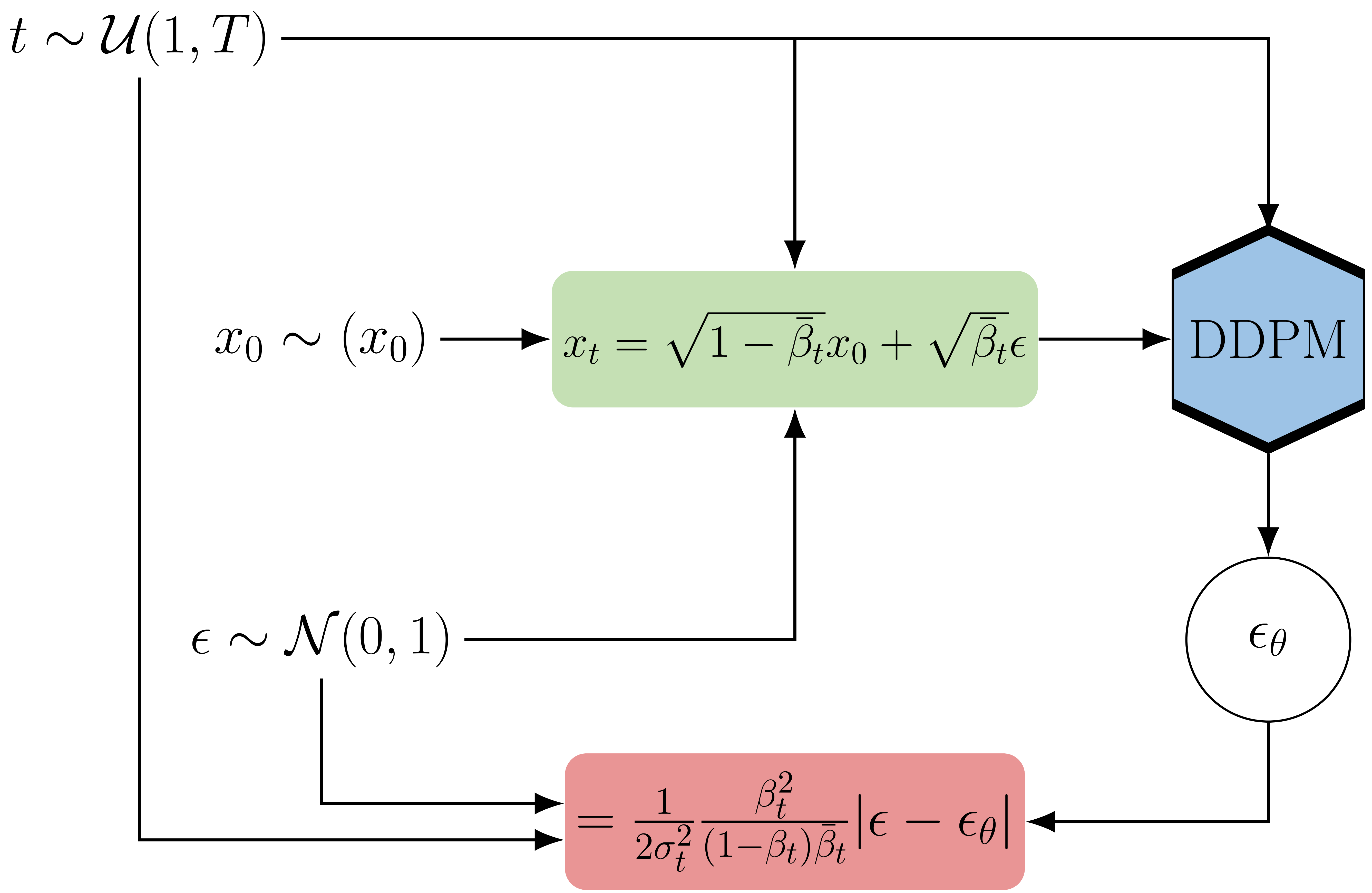
cINN Unfolding





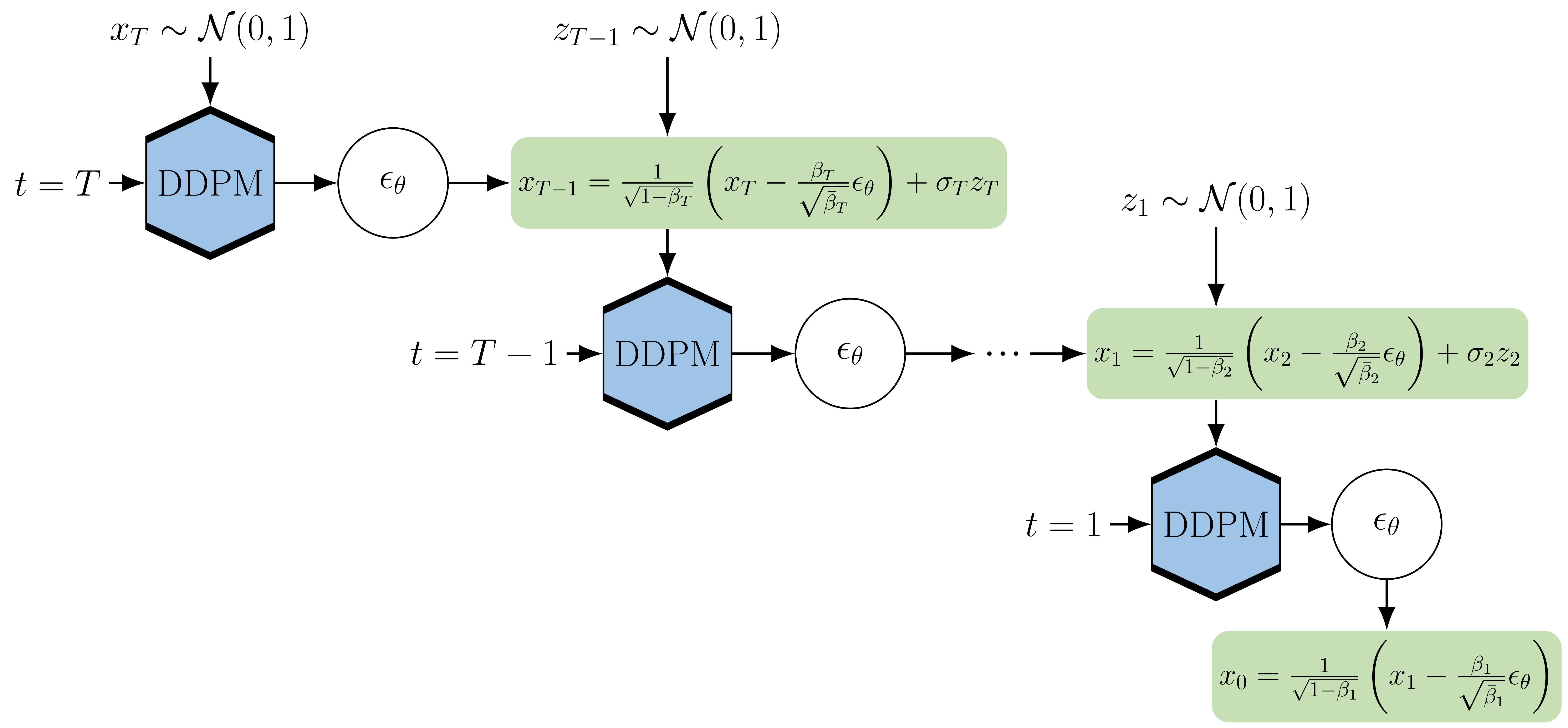
Blackboard Session III

DDPM Training



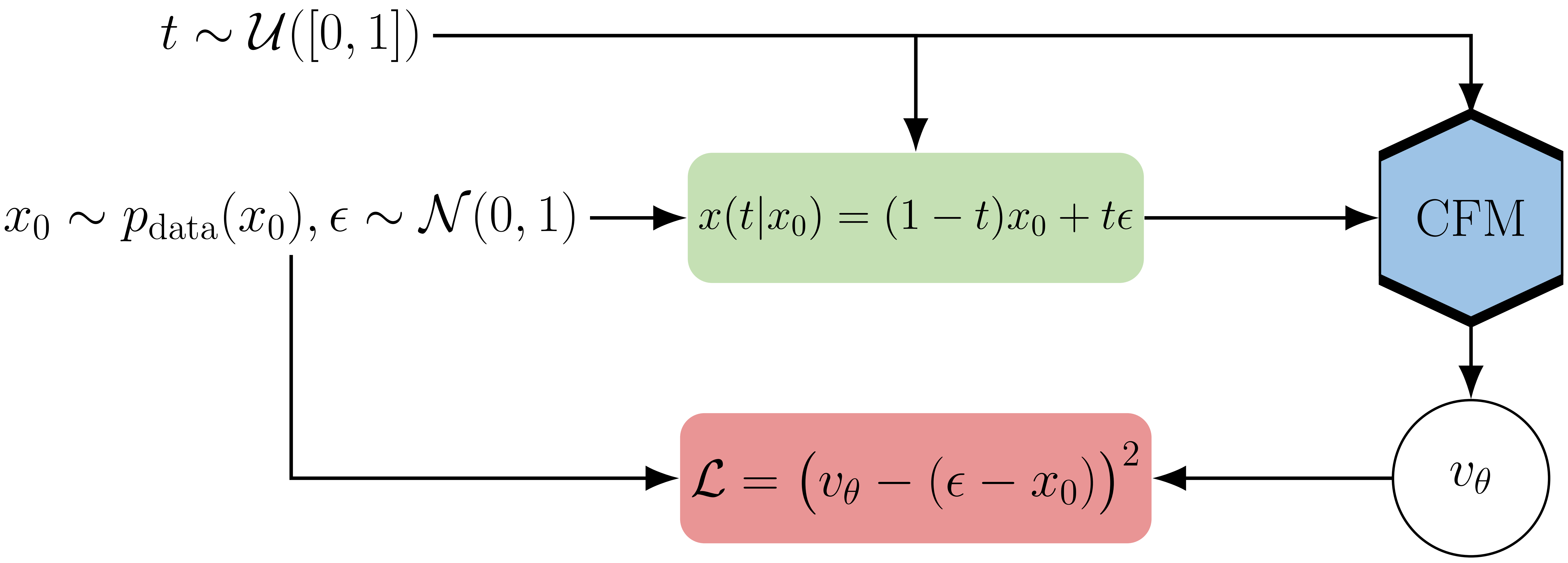
[2305.10475]

DDPM Sampling



[2305.10475]

CFM Training and Sampling



Solve ODE numerically for sampling

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = v_\theta(x(t), t)$$

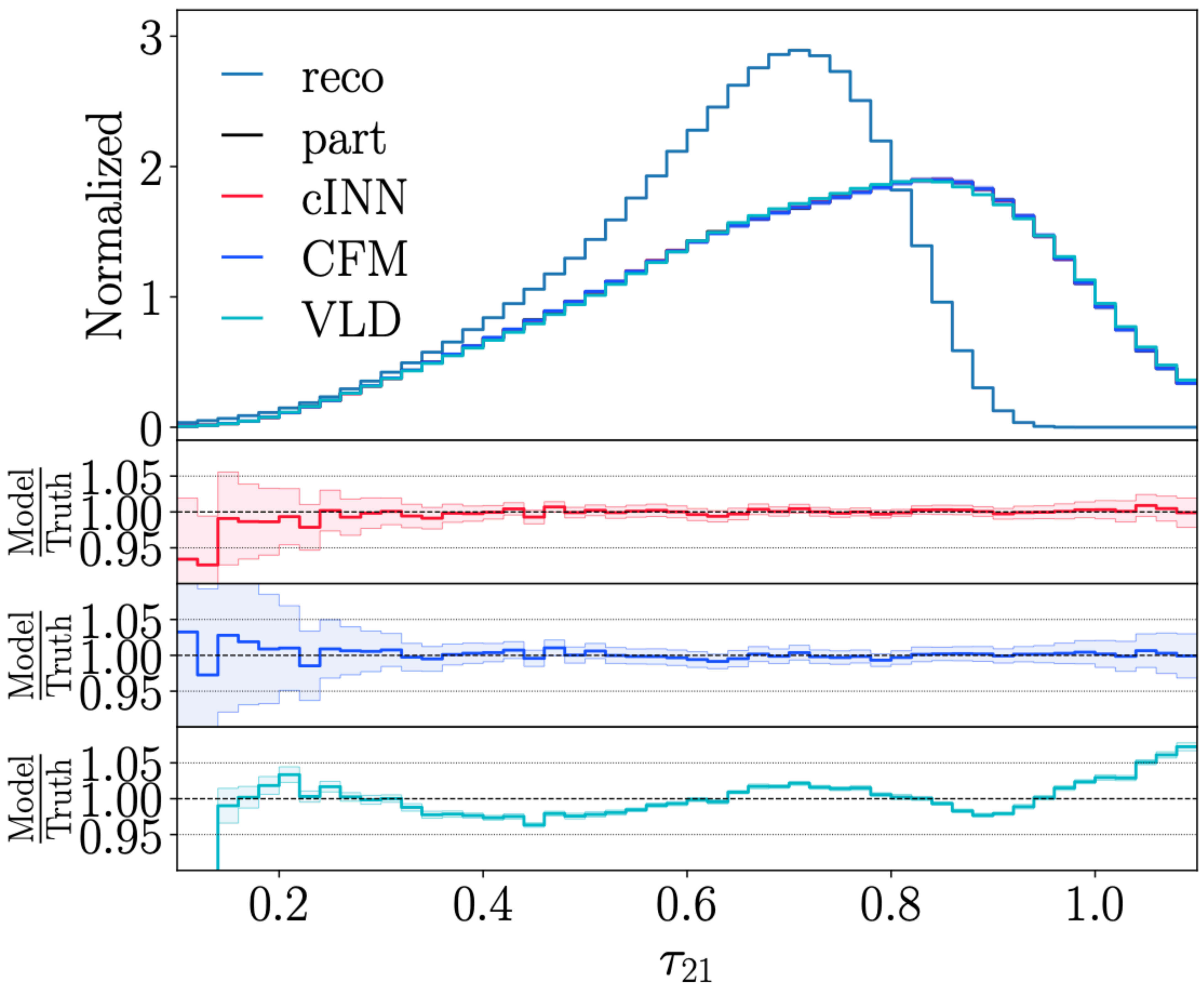
[2305.10475]

Diffusion Models

Unfolding

Unfolding with Diffusion

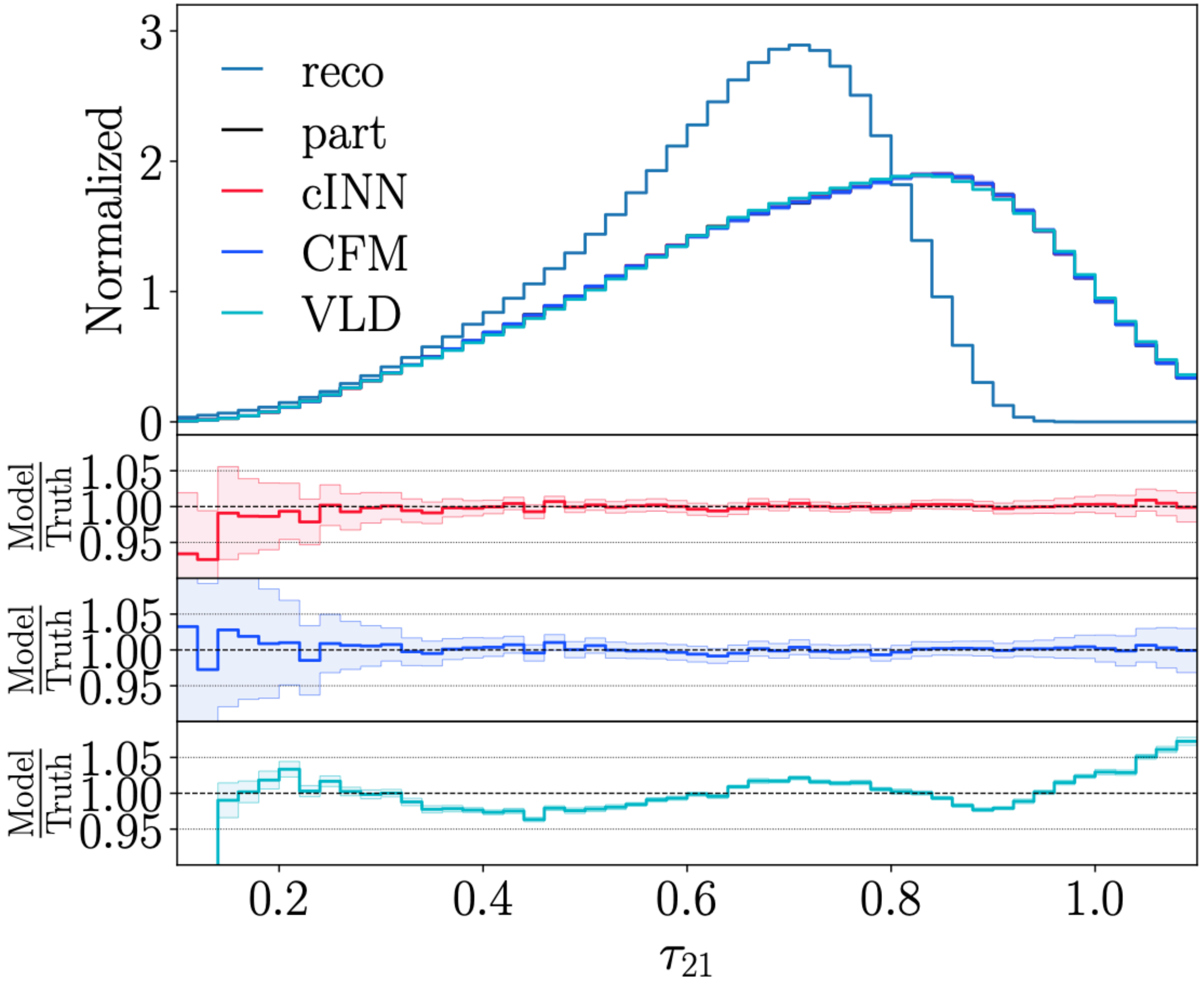
Z+jets (simple)



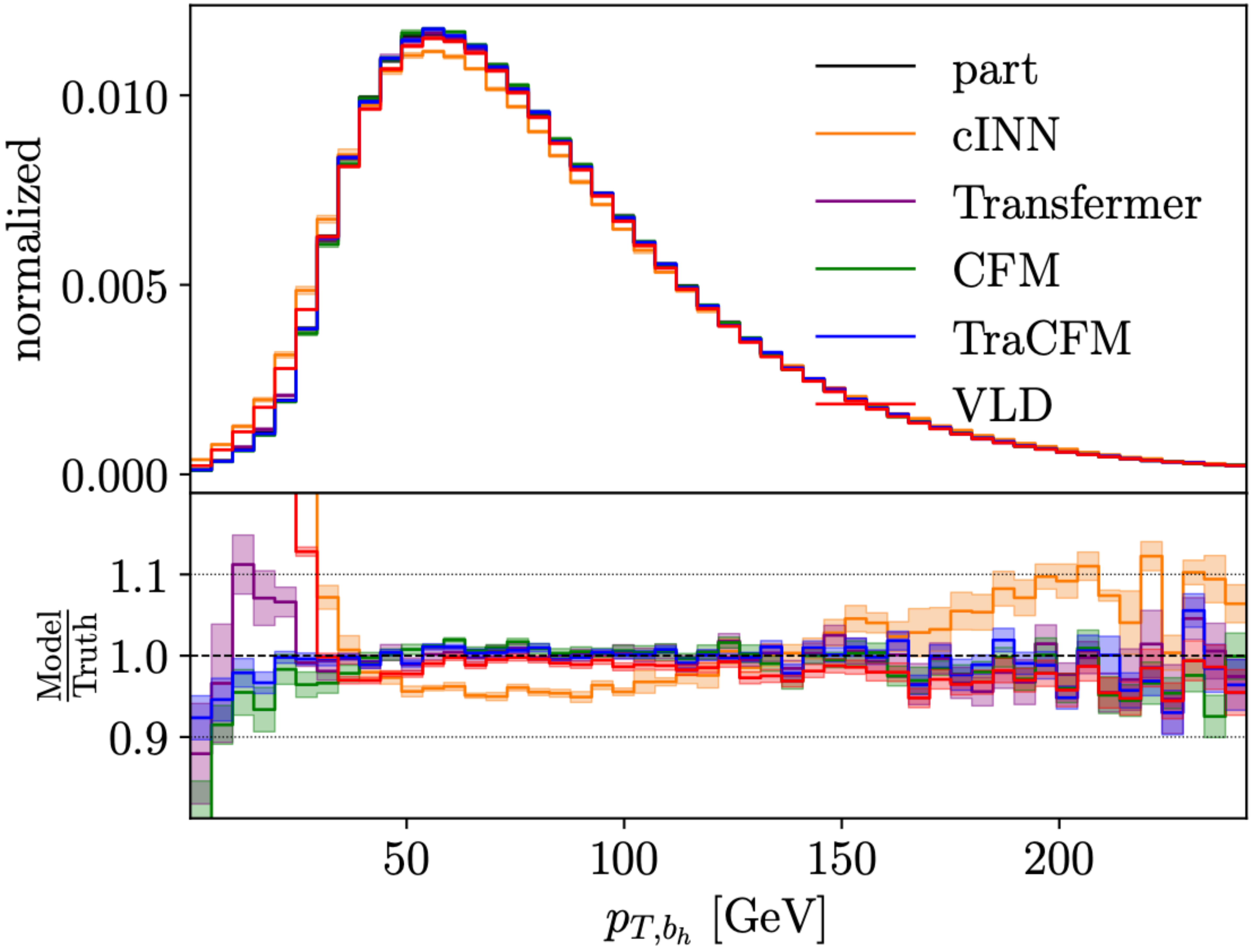
[2404.18807]

Unfolding with Diffusion

Z+jets (simple)



Top-pair with decays (hard)



[2404.18807]

Open Discussion