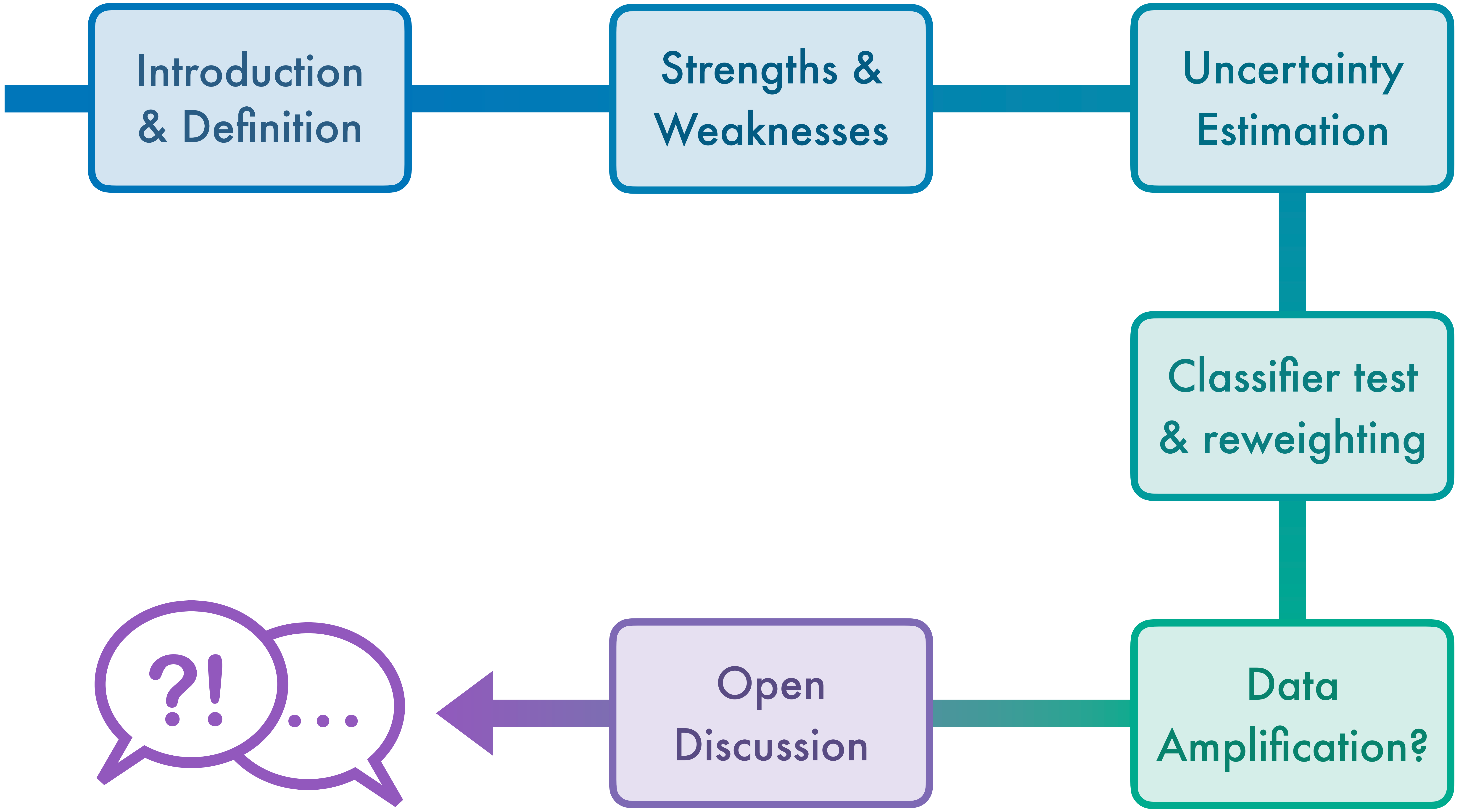
Statistics of generative models for the LHC

UNIVERSITÀ
DEGLI STUDI
DI MILANO

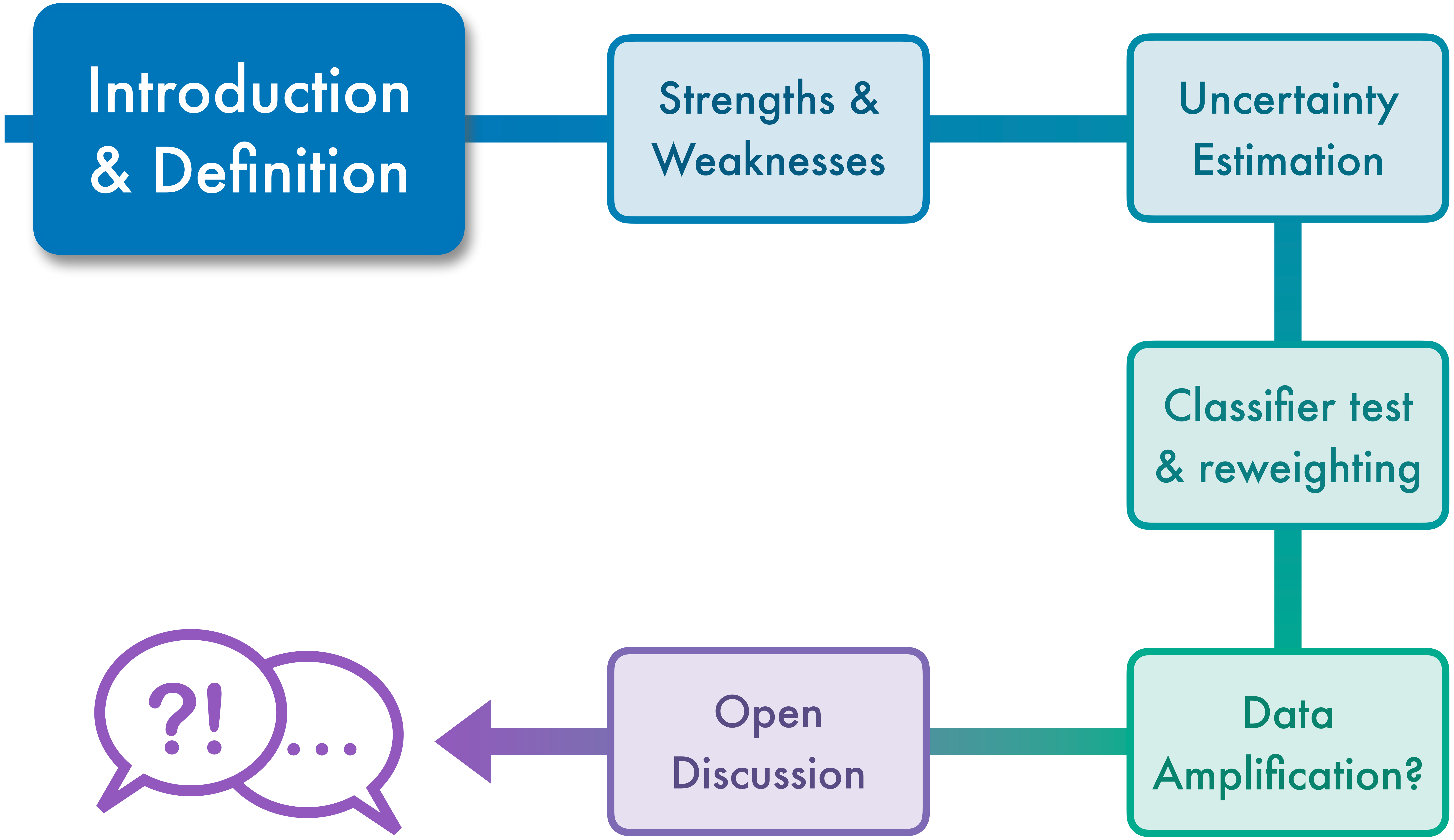
PHYSTAT — Statistics meets ML | Imperial College London

Ramon Winterhalder — UniMi

Deep Generative Models



Introduction & Definition



LHC analysis (oversimplified)

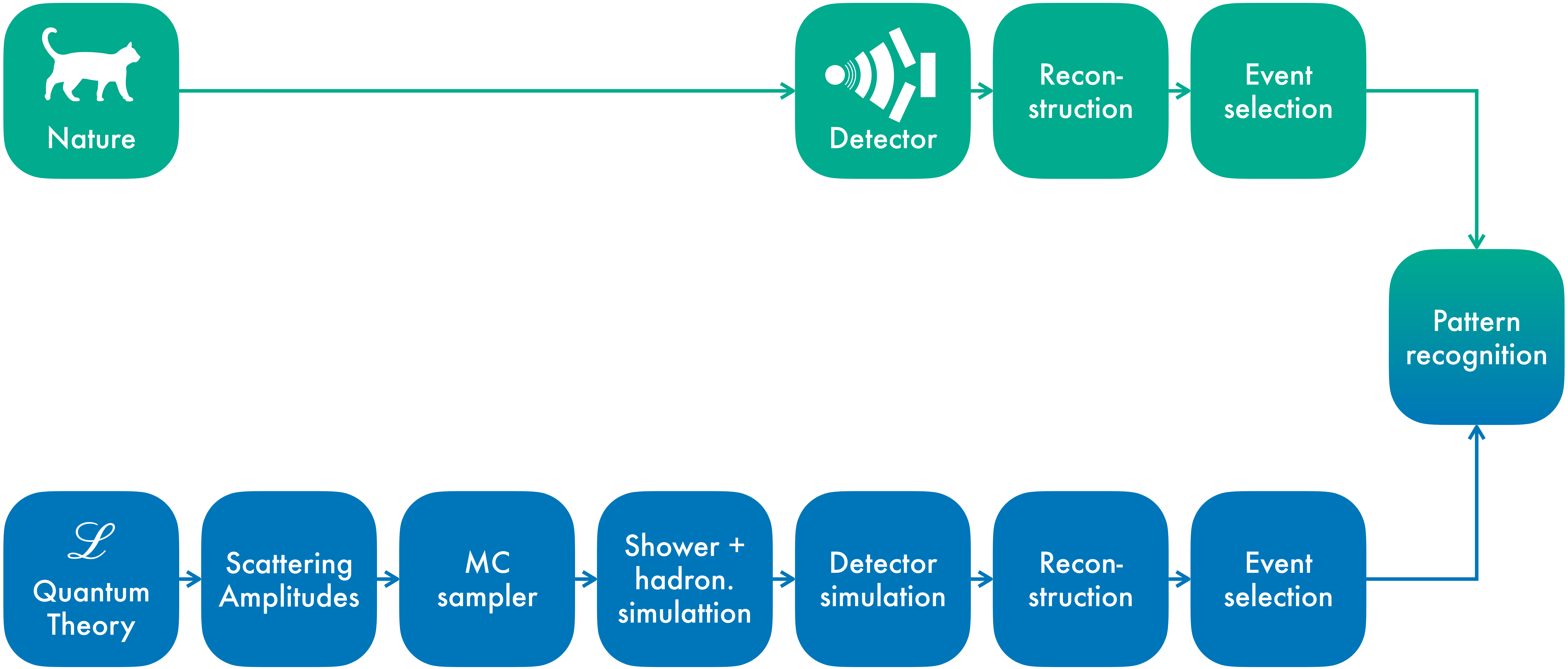


Figure adopted from [2405.14806]

LHC analysis (oversimplified)

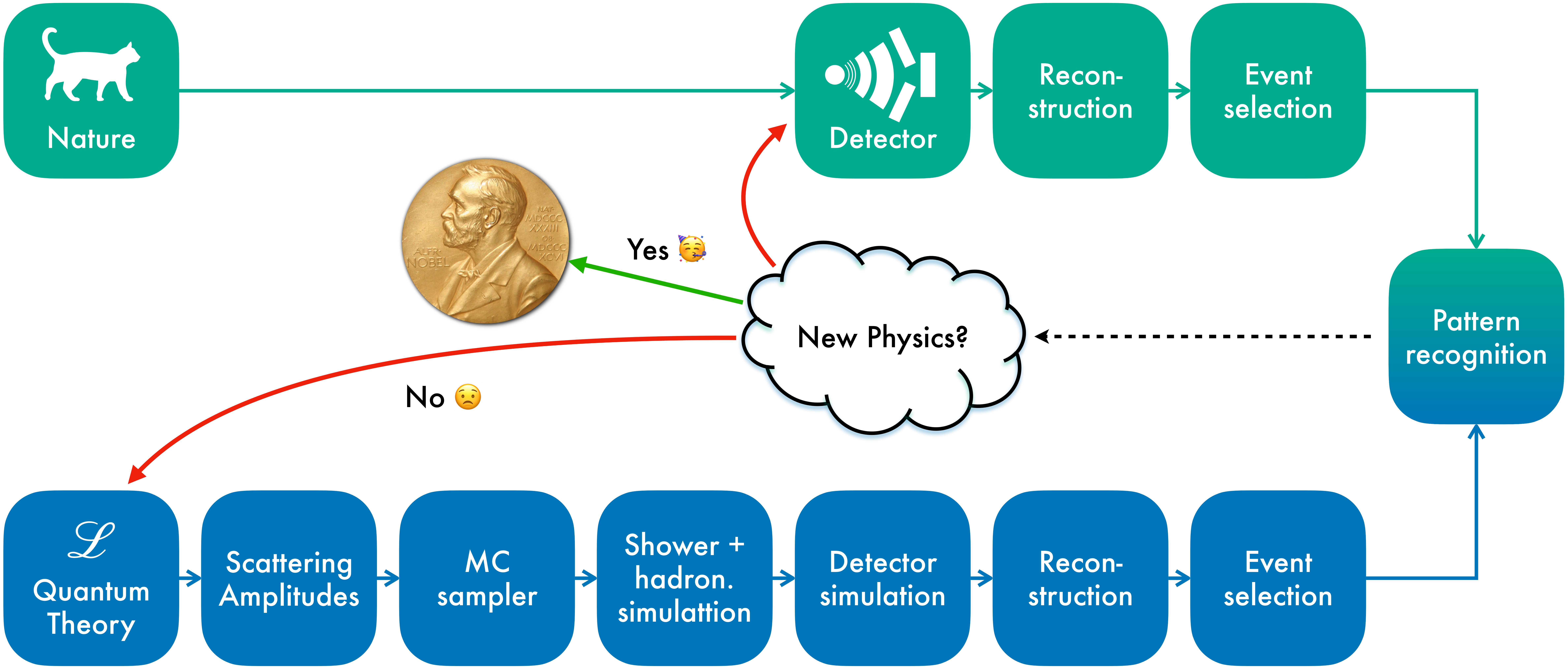


Figure adopted from [2405.14806]

LHC analysis + DGMs

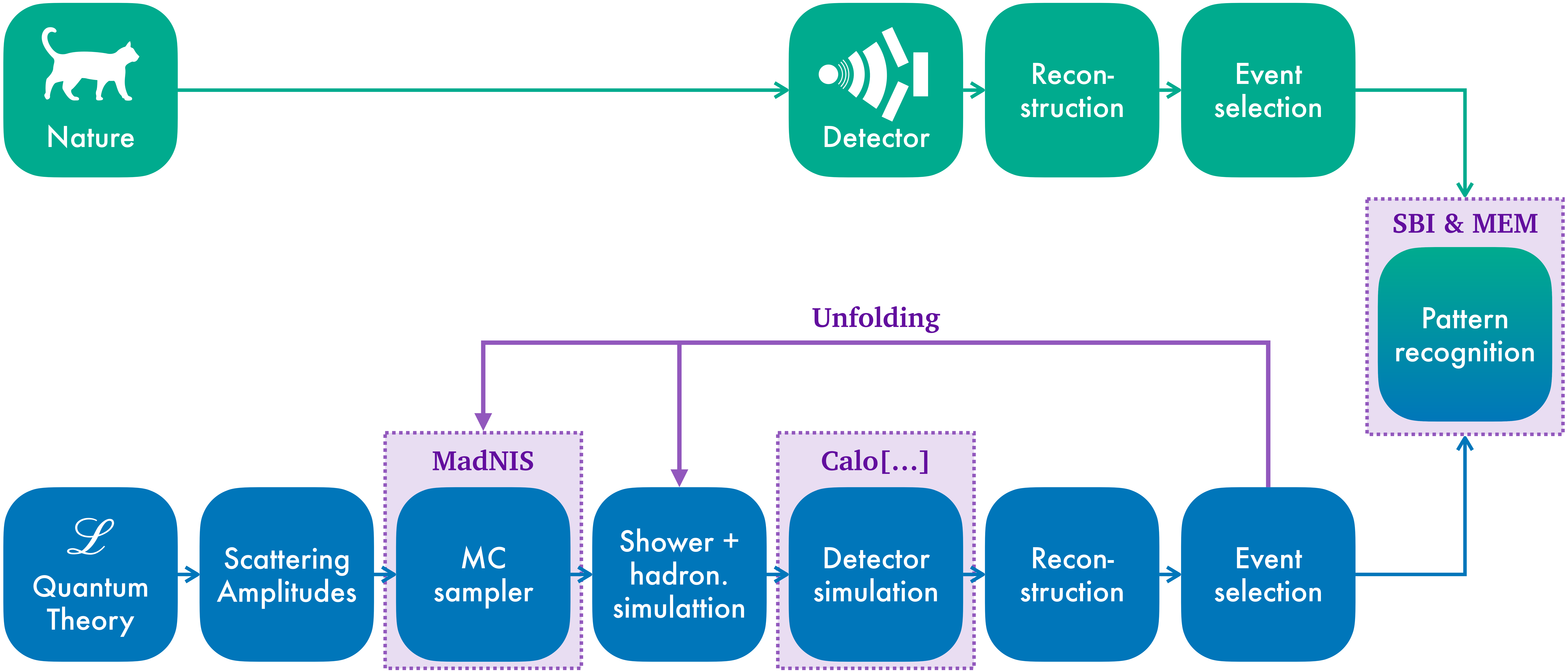
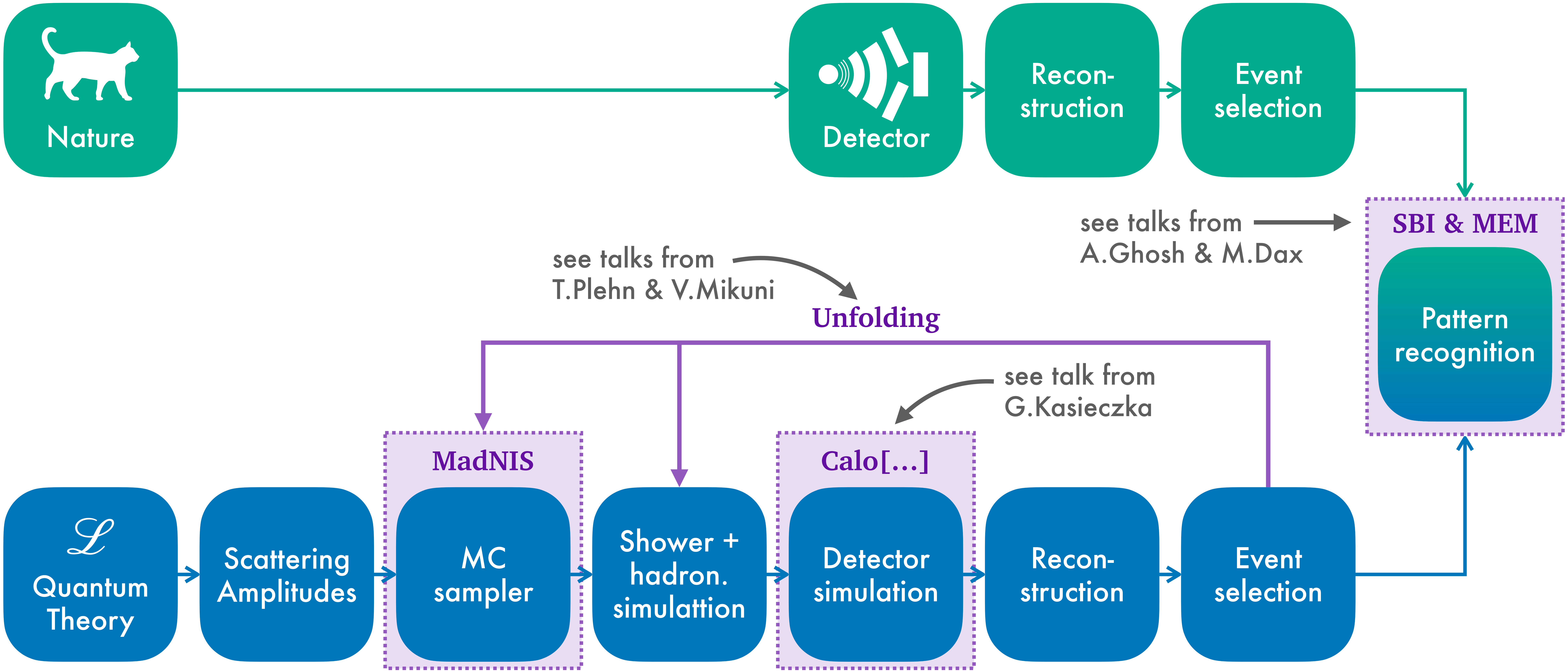


Figure adopted from [2405.14806]

LHC analysis + DGMs



see talk from J. Spinner → Figure adopted from [2405.14806]

LHC analysis + DGMs

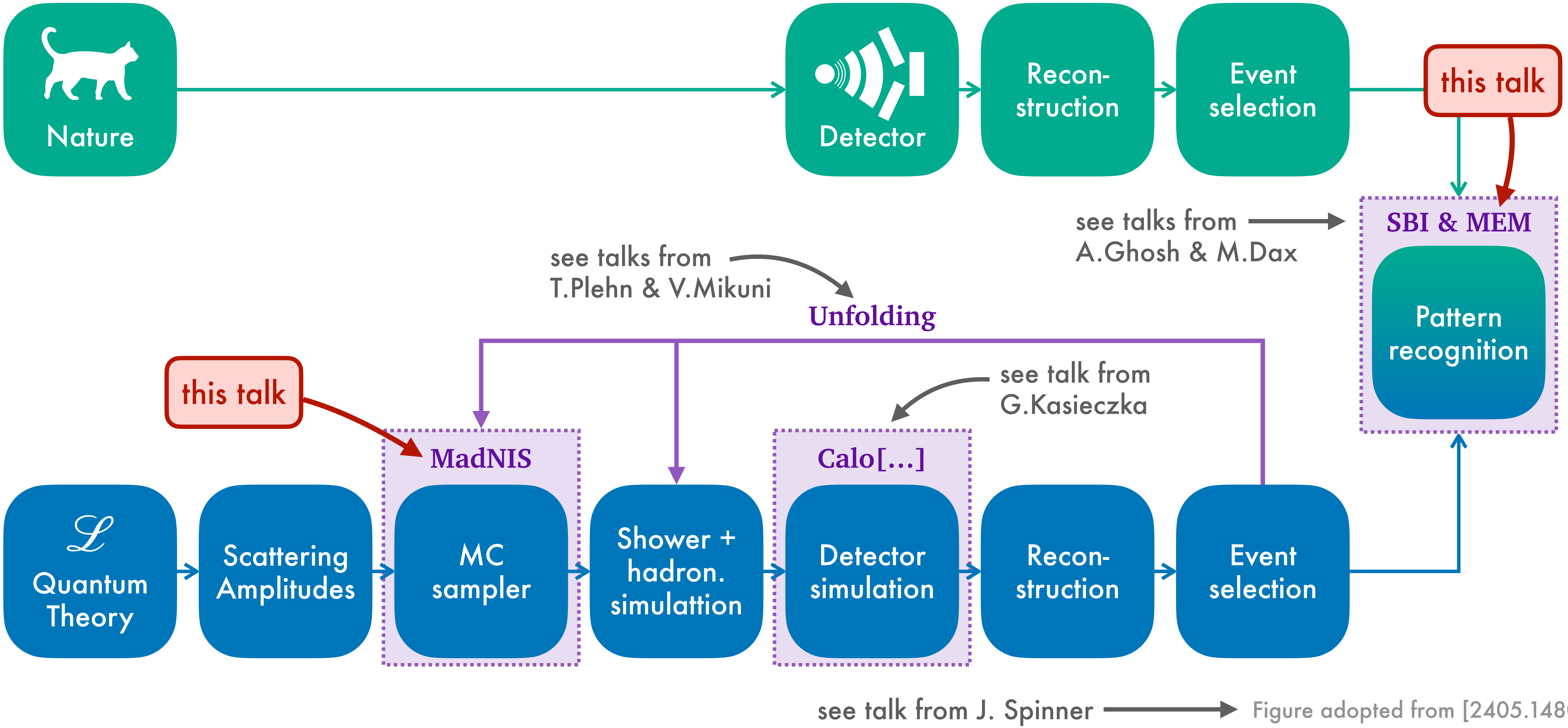
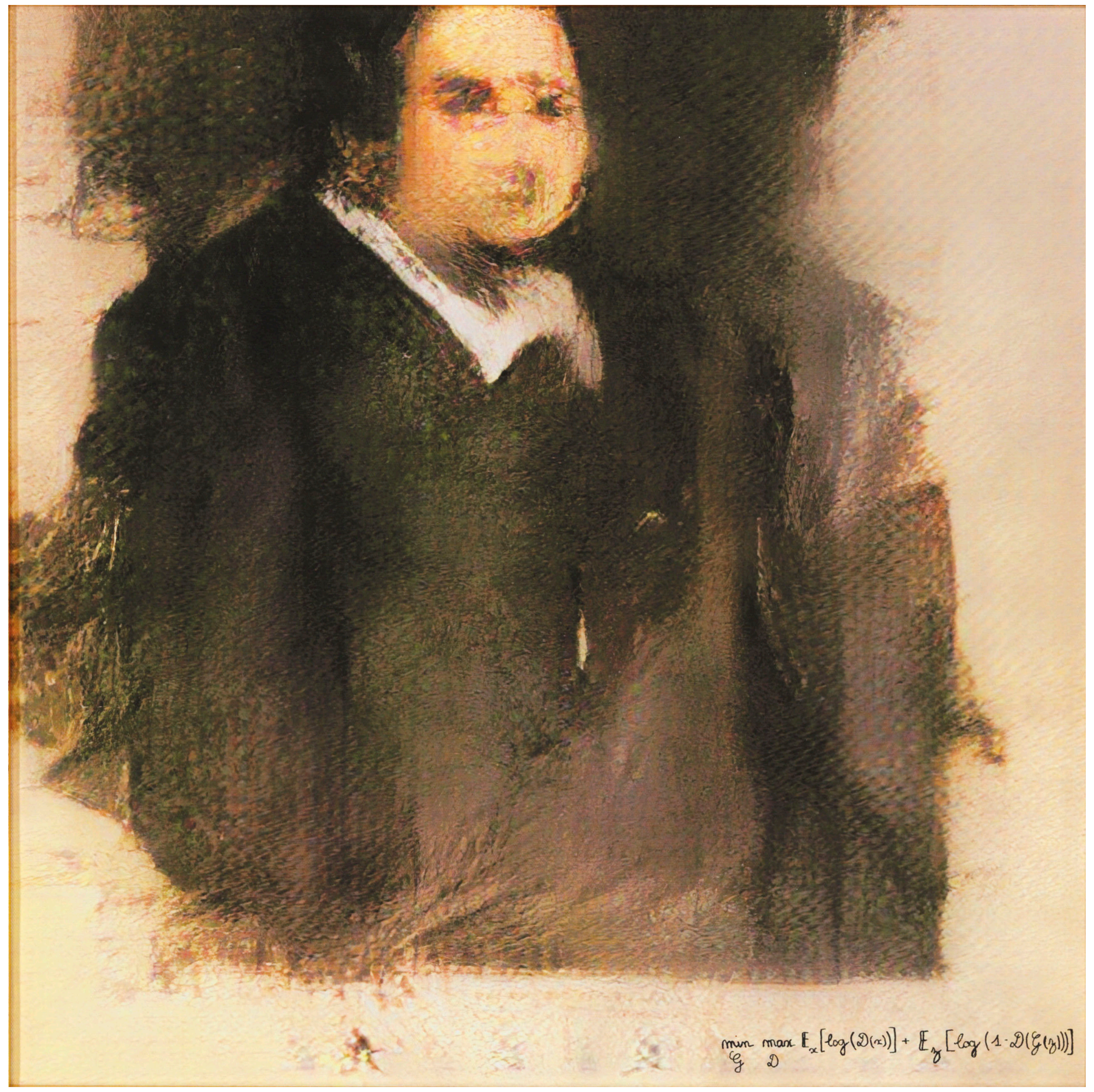


Figure adopted from [2405.14806]

Deep Generative Models

GAN



GAN Art (2018)
→ sold for $432,500

Diffusion Models

State-of-the-art
image generation

Transformer

ChatGPT

State-of-the-art
text generation

What is a Generative Model?

We have: $p_{\text{truth}} \equiv p_{\text{data}}(x)$ → We want to generate new samples $x \sim p_\omega(x) \simeq p_{\text{data}}(x)$

The distribution p_{truth} is usually given as:

- **explicit** as function (e.g. $\mathrm{d}\sigma \propto$ differential cross-section)
- **implicit** via a set of training data $\{x\} \sim p_{\text{data}}(x)$

In **particle physics:**

- Event generation
- Calorimeter simulation
- Unfolding
- MEM (transfer function)

What is a Generative Model?

We have: $p_{\text{truth}} \equiv p_{\text{data}}(x)$ → We want to generate new samples $x \sim p_\omega(x) \simeq p_{\text{data}}(x)$

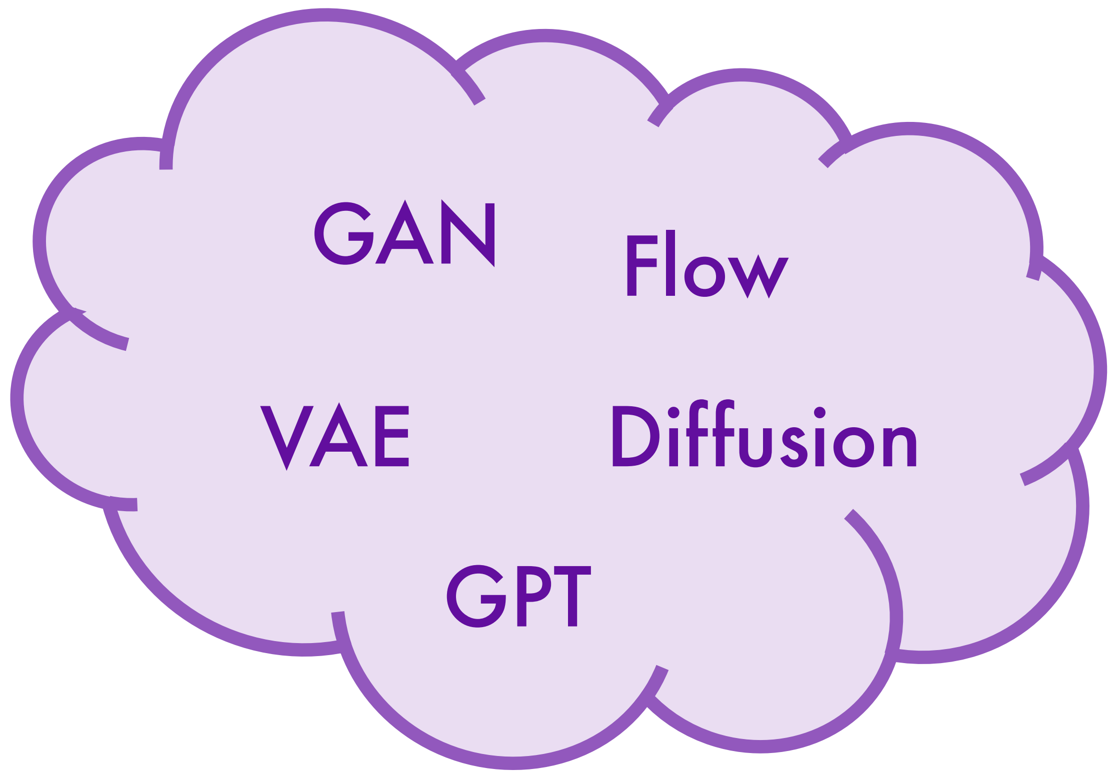
The distribution p_{truth} is usually given as:

- **explicit** as function (e.g. $\mathrm{d}\sigma \propto$ differential cross-section)
- **implicit** via a set of training data $\{x\} \sim p_{\text{data}}(x)$

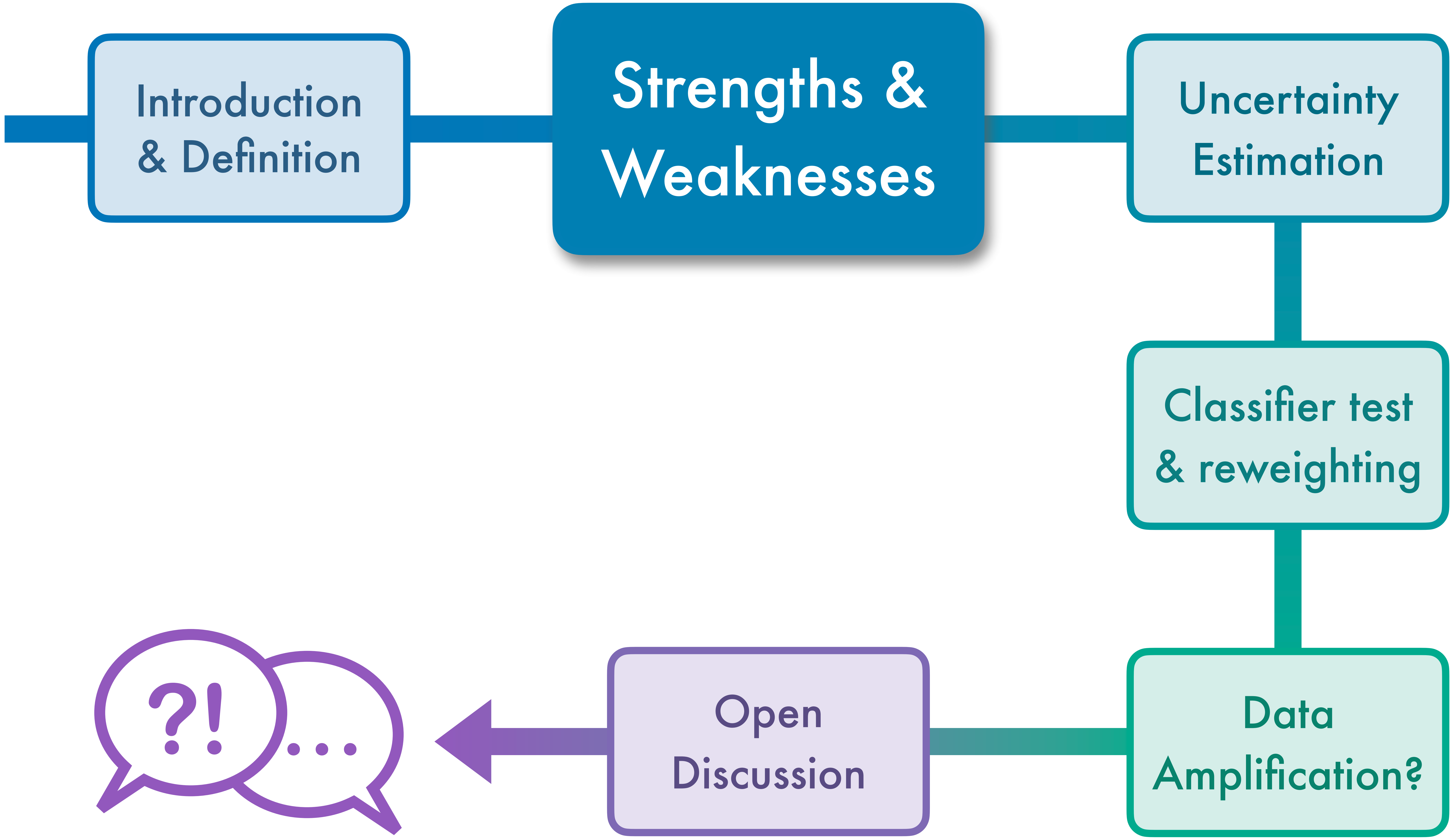
In **particle physics:**

- Event generation
- Calorimeter simulation
- Unfolding
- MEM (transfer function)

→ **Multiple types** of generative models



Strengths & Weaknesses of DGMs



Deep generative models

Variational Autoencoder

- β-VAE
- Hierarchical VAE
- VQ-VAE

Diffusion Model

- Diffusion Probabilistic Model
- Score-matching Model
- Conditional Flow Matching

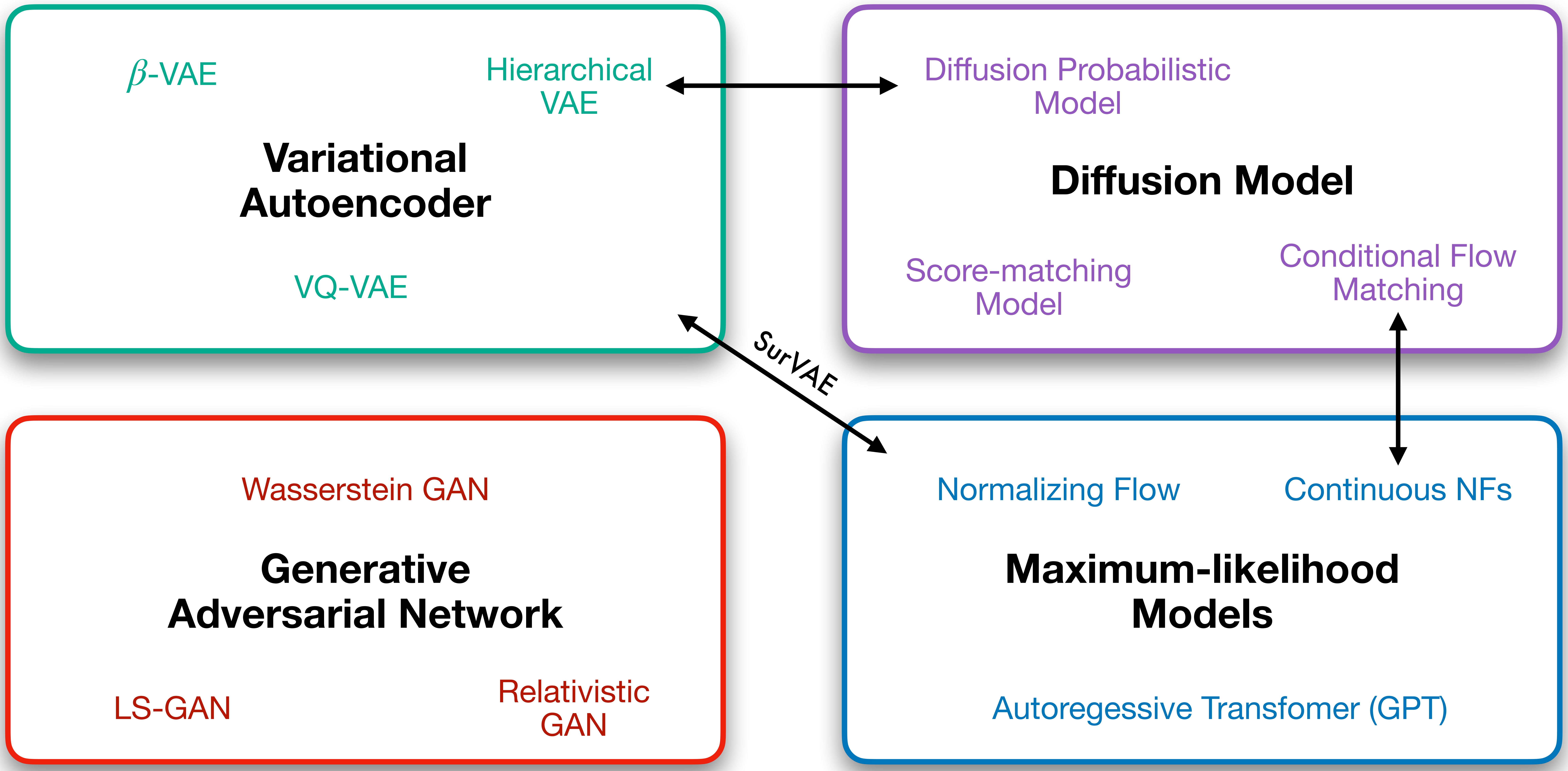
Generative Adversarial Network

- Wasserstein GAN
- LS-GAN
- Relativistic GAN

Maximum-likelihood Models

- Normalizing Flow
- Continuous NFs
- Autoregessive Transfomer (GPT)

Deep generative models



Deep generative models

β-VAE

Hierarchical VAE

Variational Autoencoder

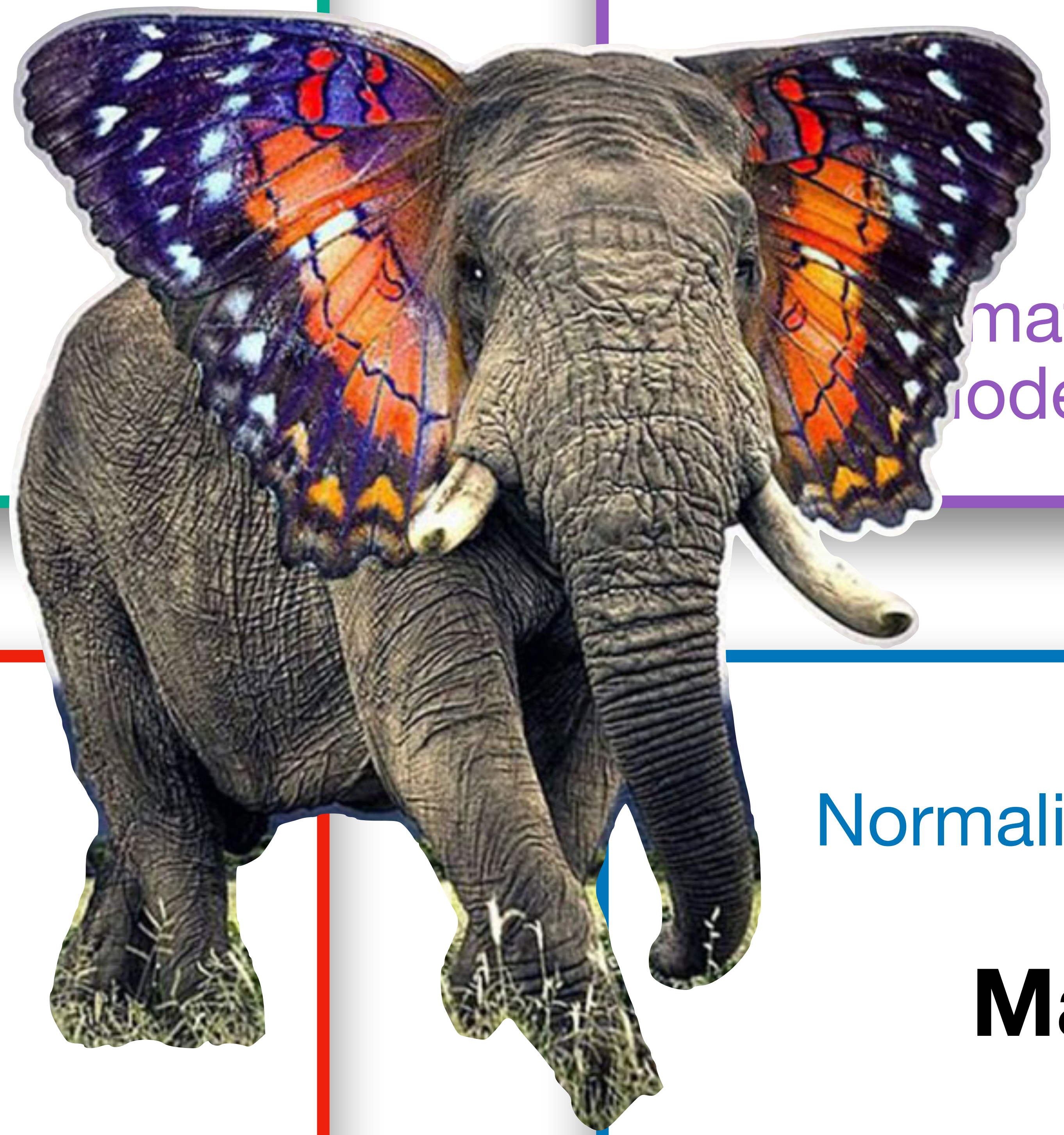
VQ-VAE

Diffusion Probabilistic Model

Diffusion Model

matching
odel

Conditional Flow Matching



Wasserstein GAN

Generative Adversarial Network

LS-GAN

Relativistic GAN

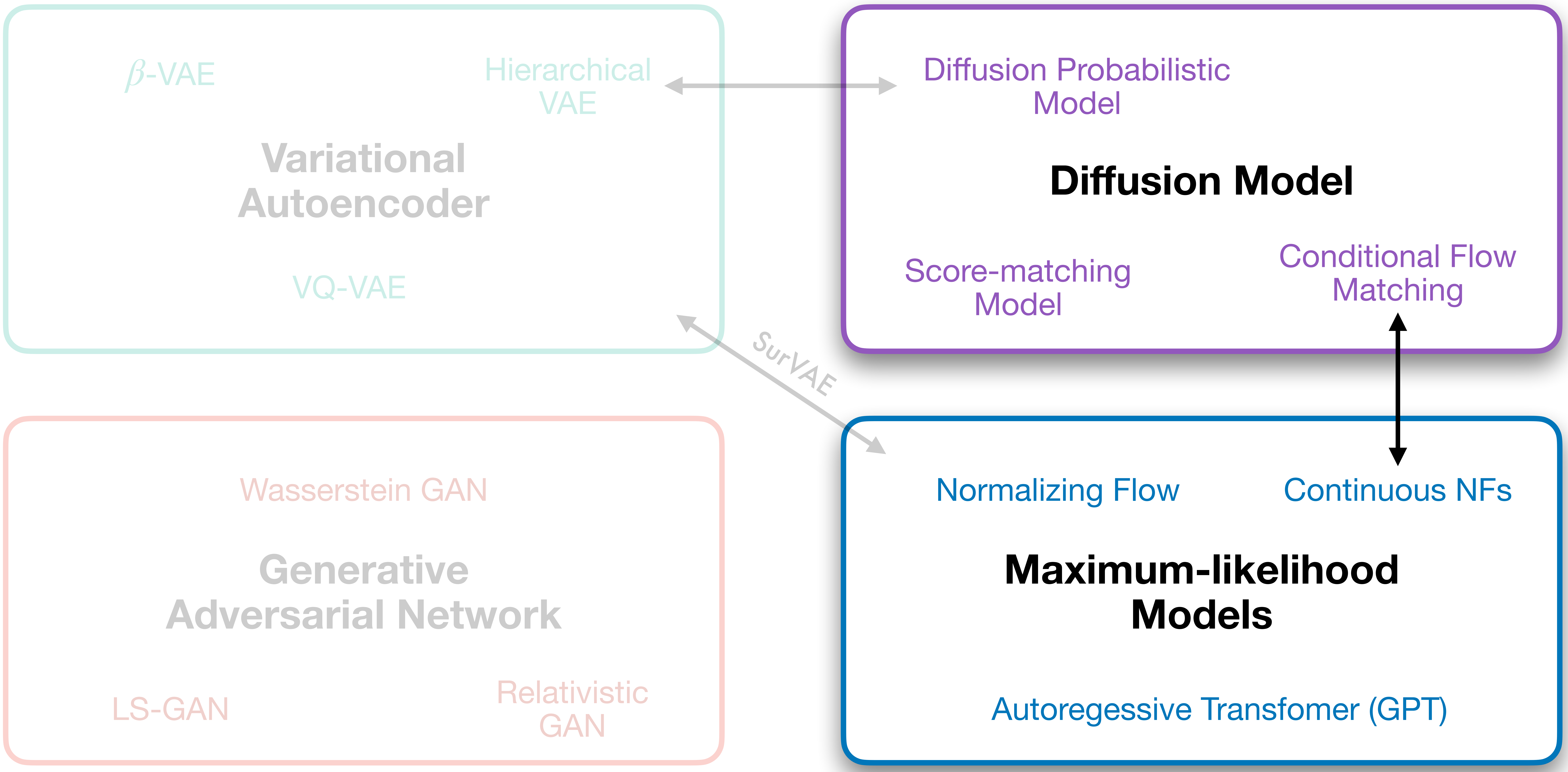
Normalizing Flow

Continuous NFs

Maximum-likelihood Models

Autoregessive Transfomer (GPT)

Deep generative models



Deep generative models

- ⊕ State-of-the-art in precision
- ⊕ Fast and stable training
- ⊖ Slow likelihood estimation

Application dependent

- ⊕ Fast training and evaluation
- ⊕ Tractable and fast likelihoods
- ⊖ Reduced flexibility and expressivity

Diffusion Model

Diffusion Probabilistic Model

Score-matching Model

Conditional Flow Matching

Maximum-likelihood Models

Normalizing Flow

Continuous NFs

Autoregessive Transfomer (GPT)

LHC analysis + DGMs

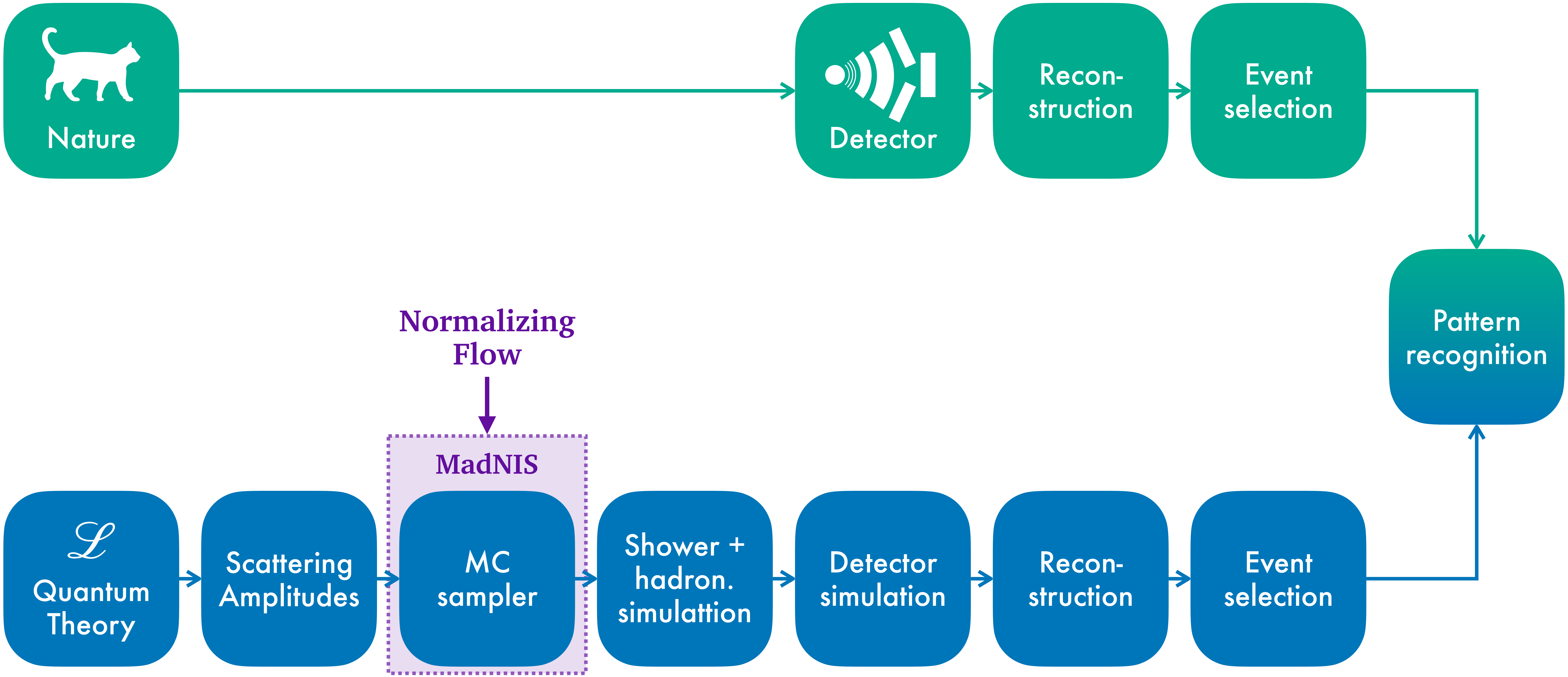
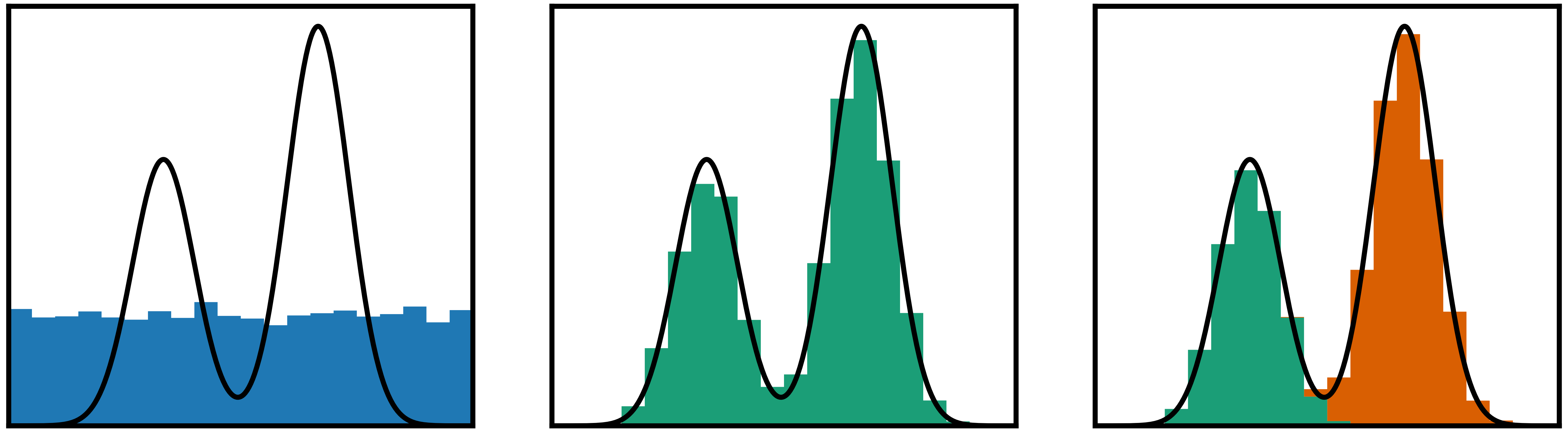


Figure adopted from [2405.14806]

MadNIS — Neural importance sampling

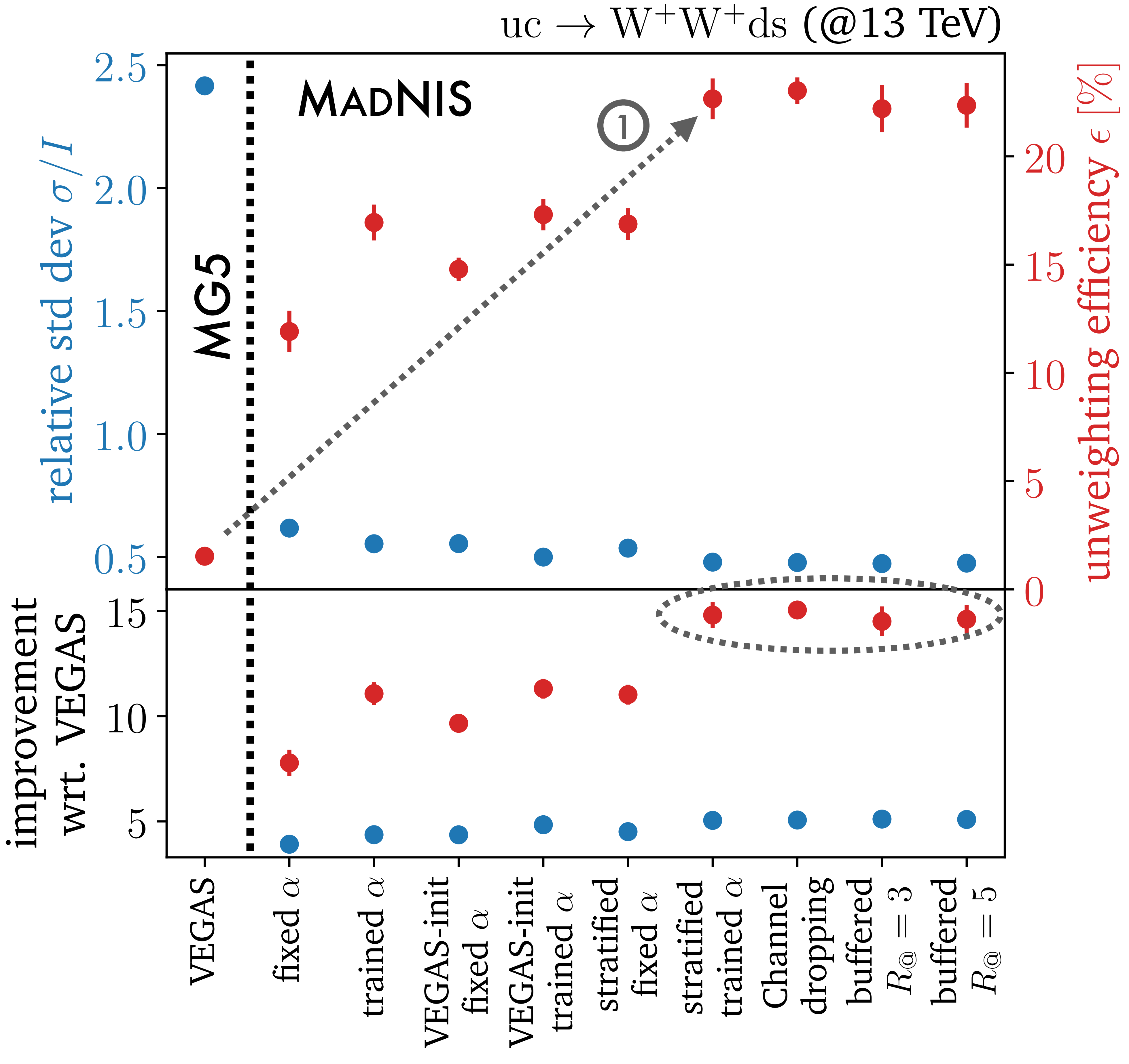


Flat sampling — Importance Sampling — Multi-channel

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

Parametrize with **NN**

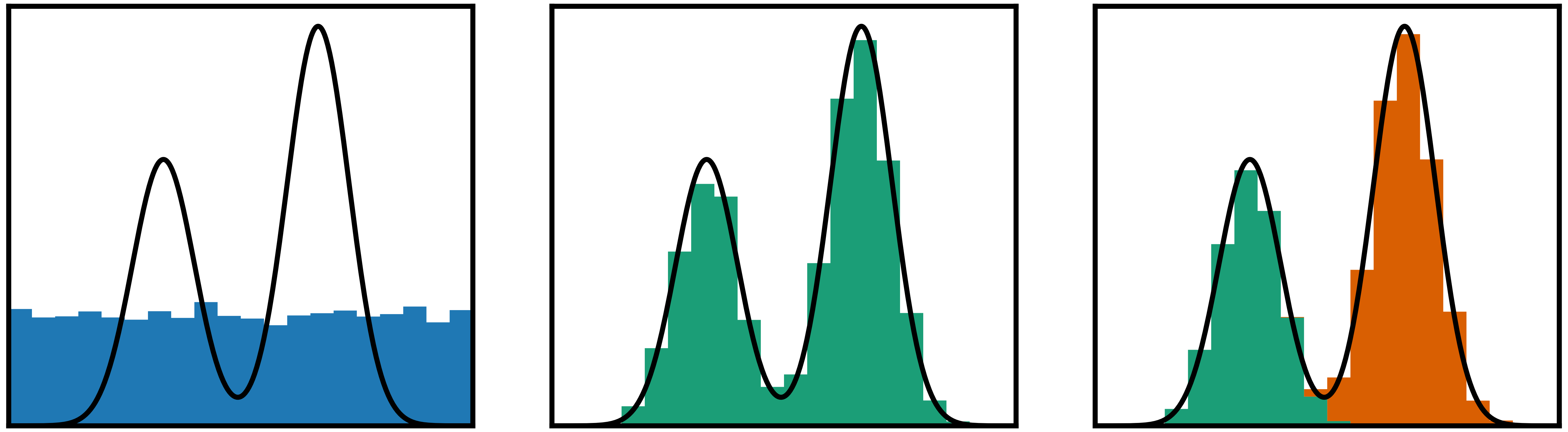
Parametrize with **Normalizing Flow**



① excellent results with all features

[2212.06172, 2311.01548, 2408.01486]

MadNIS — Neural importance sampling

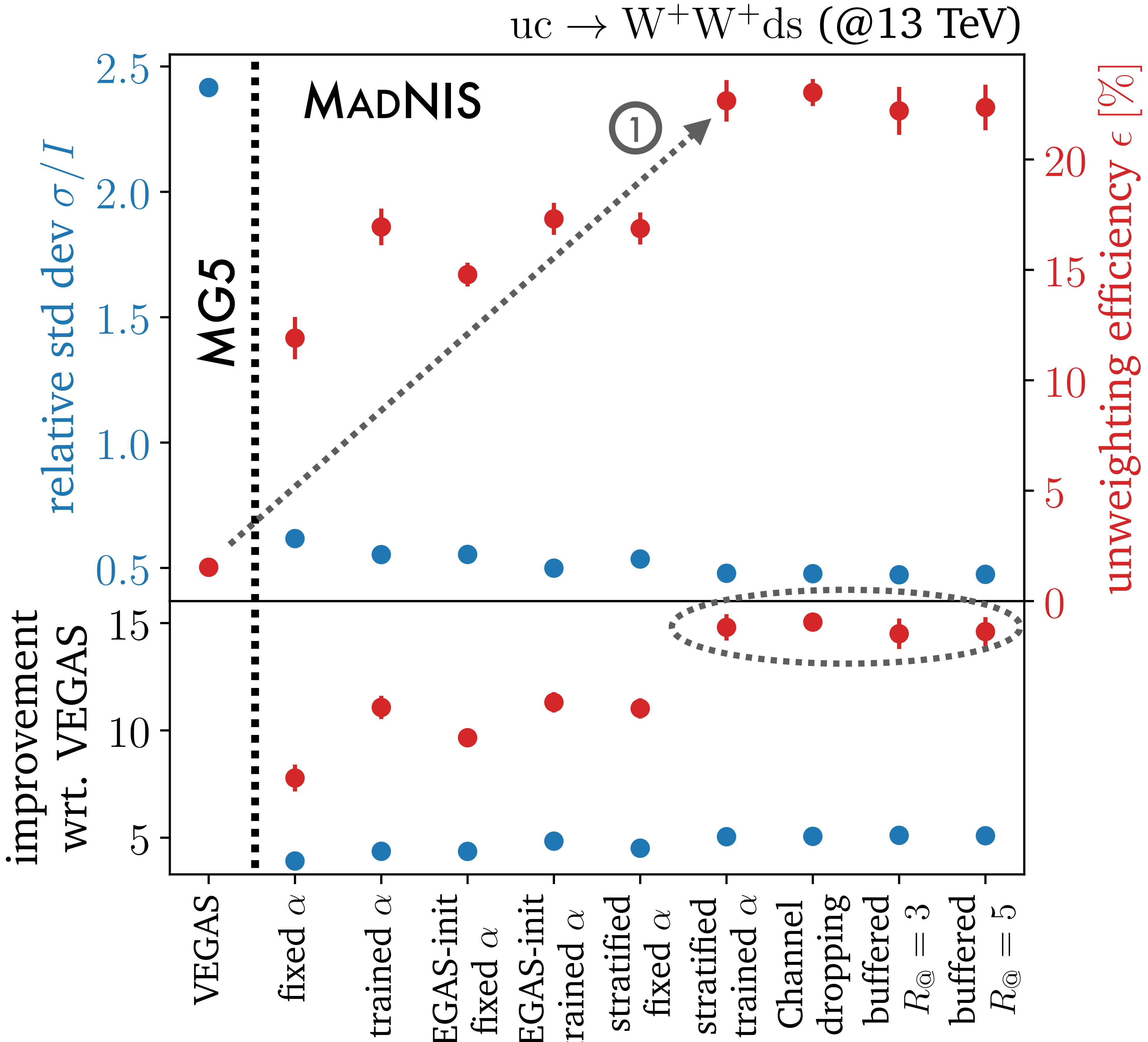


Flat sampling | Importance Sampling | Multi-channel

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

Parametrize with **NN** (for $\alpha_i(x)$)

Parametrize with **Normalizing Flow** (for $g_i(x)$)



① excellent results with all features

[2212.06172, 2311.01548, 2408.01486] ← **Fully differentiable version available**

LHC analysis + DGMs

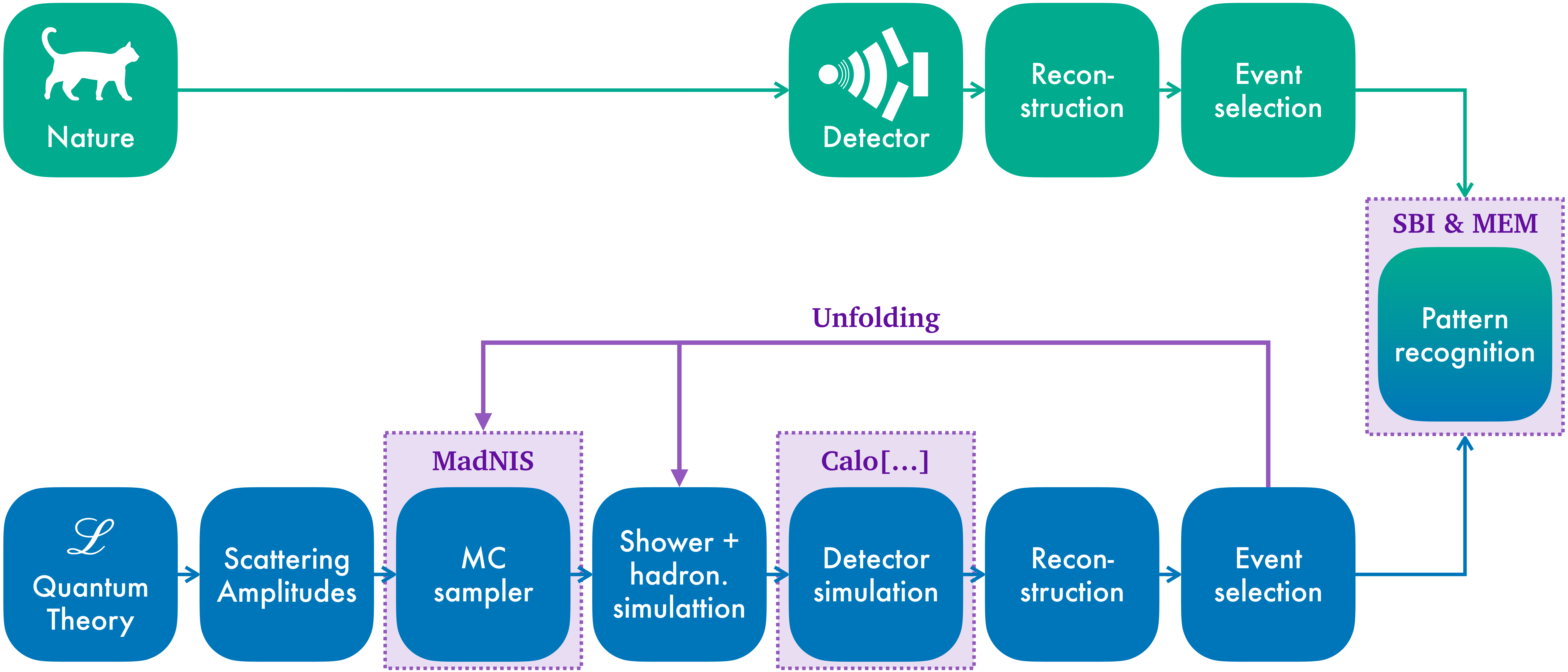


Figure adopted from [2405.14806]

LHC analysis + DGMs

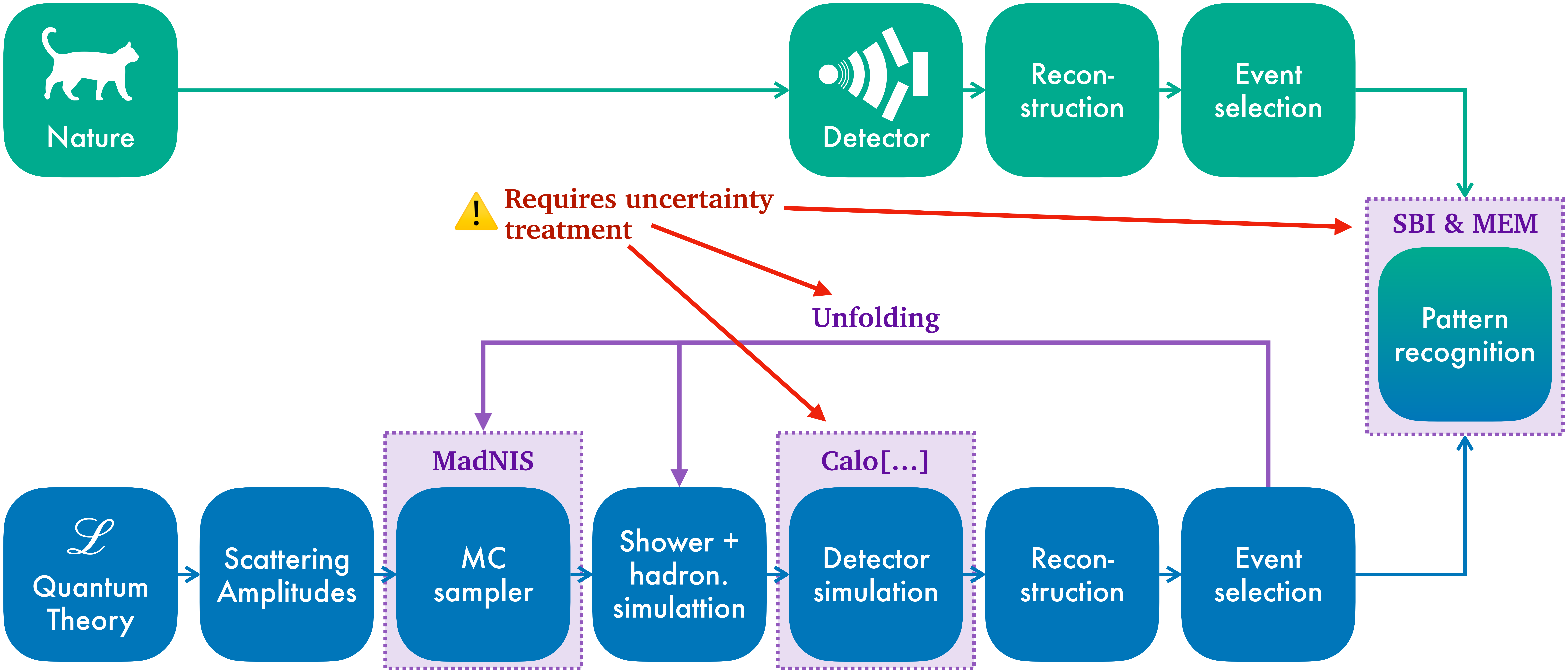
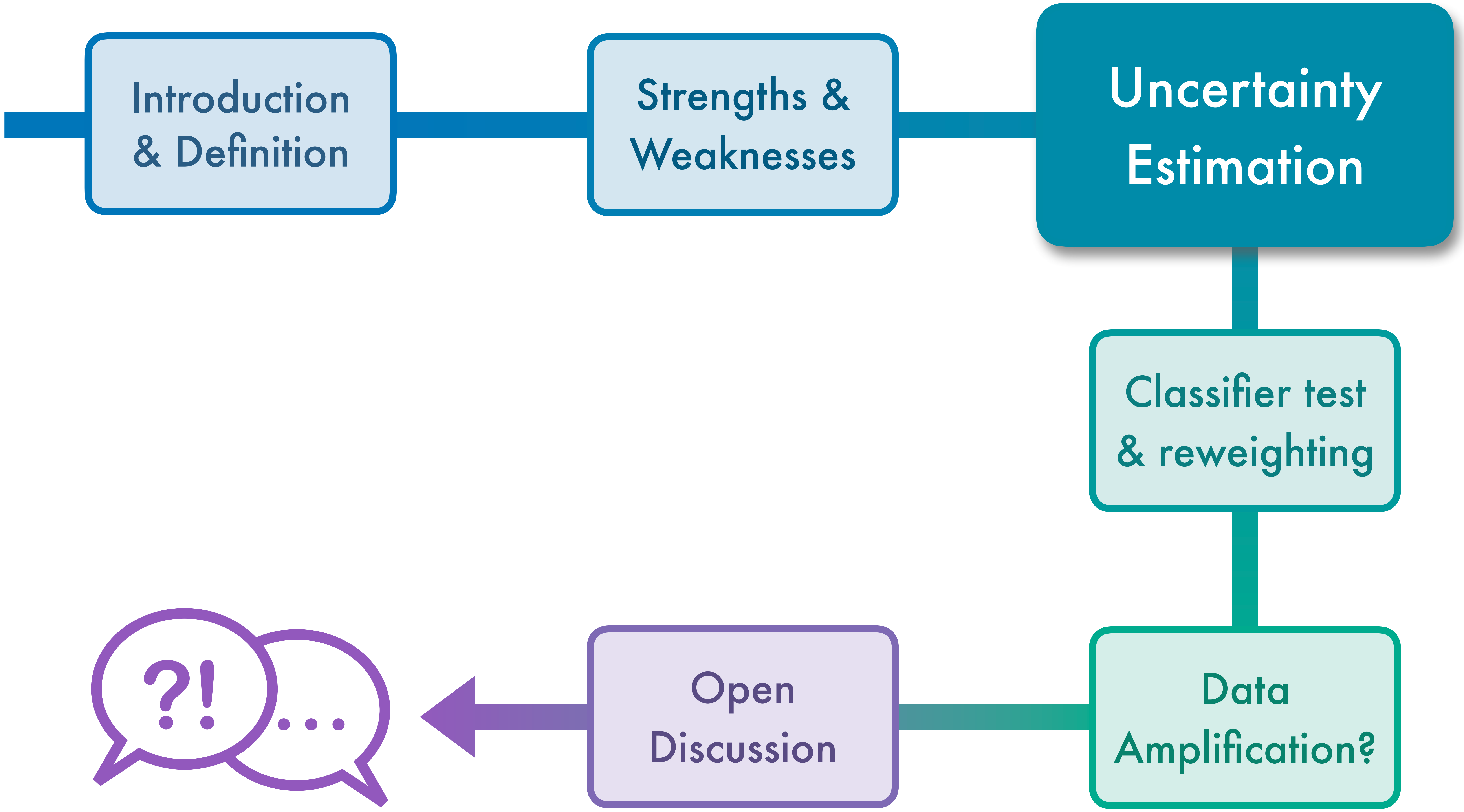


Figure adopted from [2405.14806]

Uncertainty Estimation



What we have *vs* what we want

So far: NNs are deterministic → **fixed** input x → **fixed** output y

However, in **particle physics:**

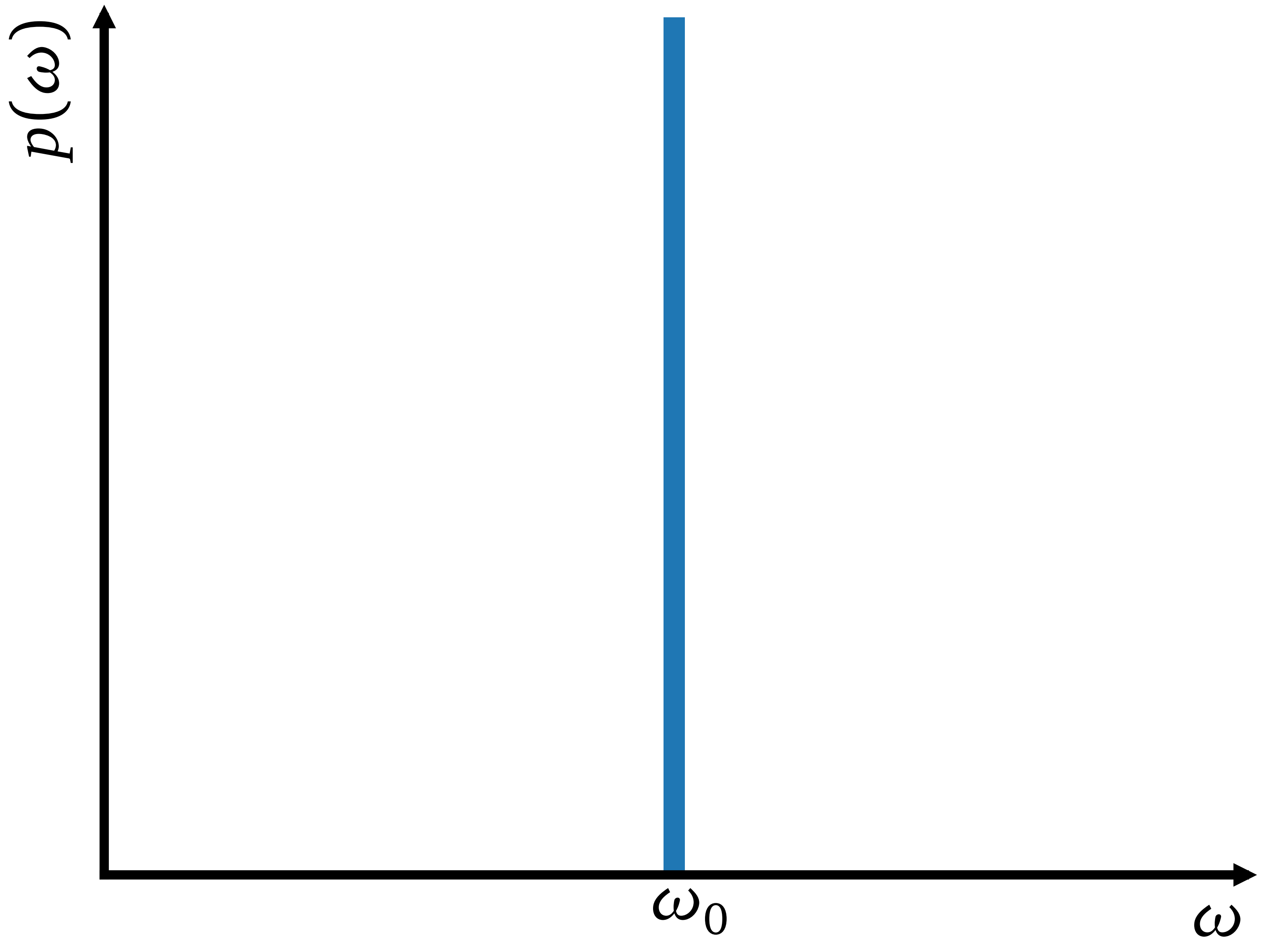
→ we are not only interested in **results**, but also in **errorbars**

We need: **fixed** input x → **probabilistic** output $\langle y \rangle \pm \sigma_y$ → $y \sim p(y|x)$

How do we achieve this?

Bayesian neural networks

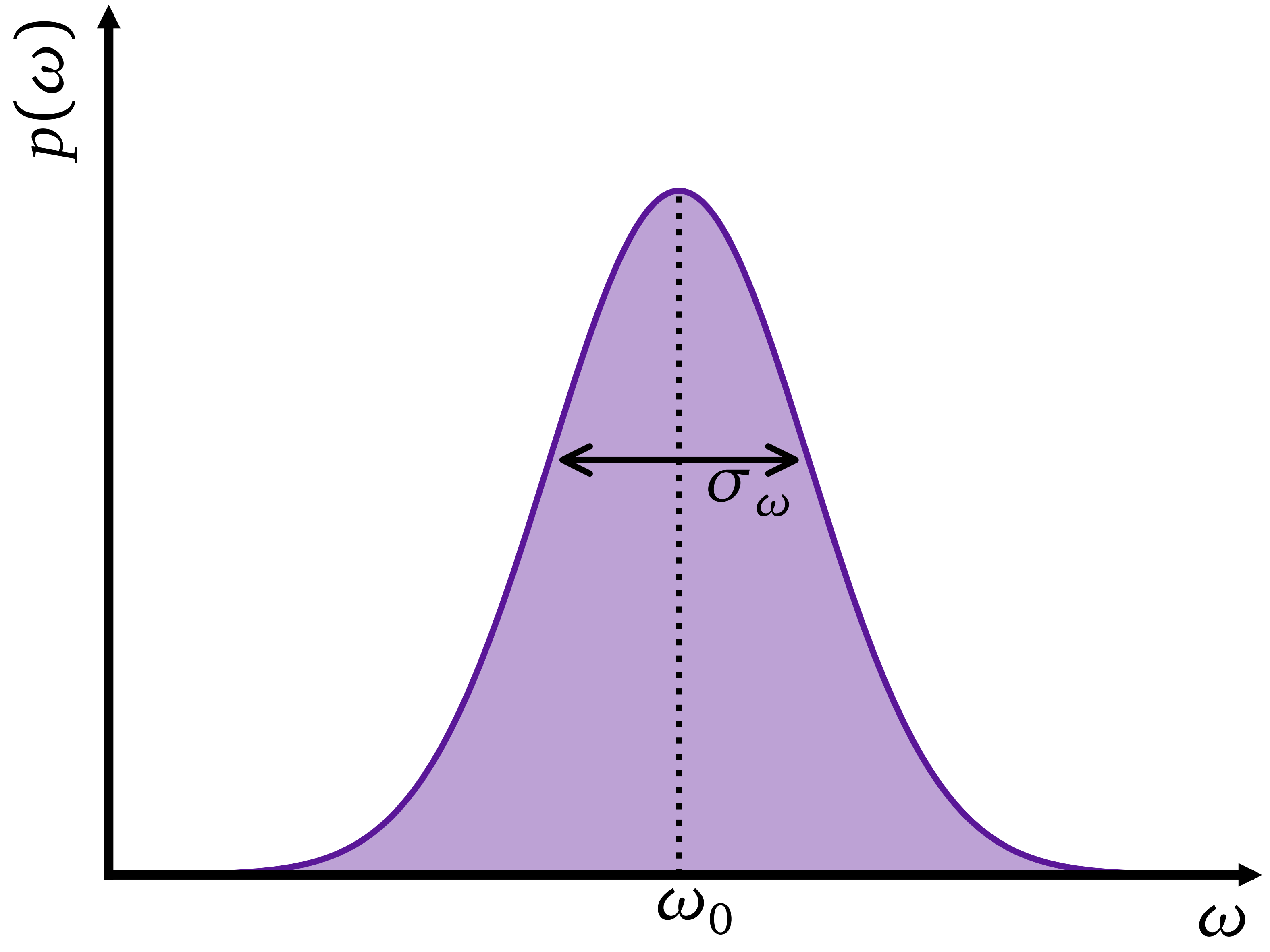
Neural network



Network weights are
deterministic

$$\omega = \omega_0$$

Bayesian Neural network



Network weights are
drawn from distribution

$$\omega \sim p(\omega)$$

BNN loss function

- Predictive distribution in **DGMs**: $p(y\,|\,x) = \int \mathrm{d}\omega\, p(y\,|\,\omega, x)\, p(\omega\,|\,x)$ ← Average over posterior

  DGM output (points to $p(y\,|\,\omega, x)$); Intractable ☹ (points to $p(\omega\,|\,x)$)

- Approximate posterior: $p(\omega\,|\,x) \simeq q_\alpha(\omega\,|\,x) \approx q_\alpha(\omega)$ → e.g. Gaussian with $\alpha = (\mu, \sigma)$

BNN loss function

- Predictive distribution in **DGMs**: $p(y|x) = \int d\omega\, p(y|\omega, x)\, p(\omega|x)$ ← Average over posterior
  - DGM output: $p(y|\omega, x)$
  - Intractable ☹: $p(\omega|x)$

- Approximate posterior: $p(\omega|x) \simeq q_\alpha(\omega|x) \approx q_\alpha(\omega)$ → e.g. Gaussian with $\alpha = (\mu, \sigma)$

- Variational inference: find α that minimizes $\mathrm{KL}(q_\alpha(x), p(\omega|x))$ ← Bayes' Theorem $p(\omega|x) = \frac{p(x|\omega)\, p(\omega)}{p(x)}$

- BNN loss function

$$\mathcal{L}_{\mathrm{BNN}} = -\underbrace{\langle \log p(x|\omega) \rangle_{q_\alpha(\omega)}}_{1} + \underbrace{\mathrm{KL}(q_\alpha(\omega), p(\omega))}_{2}$$

1. Neg log-likelihood averaged over q_α
2. q_α shoud not deviate too much from prior!

LHC analysis + DGMs

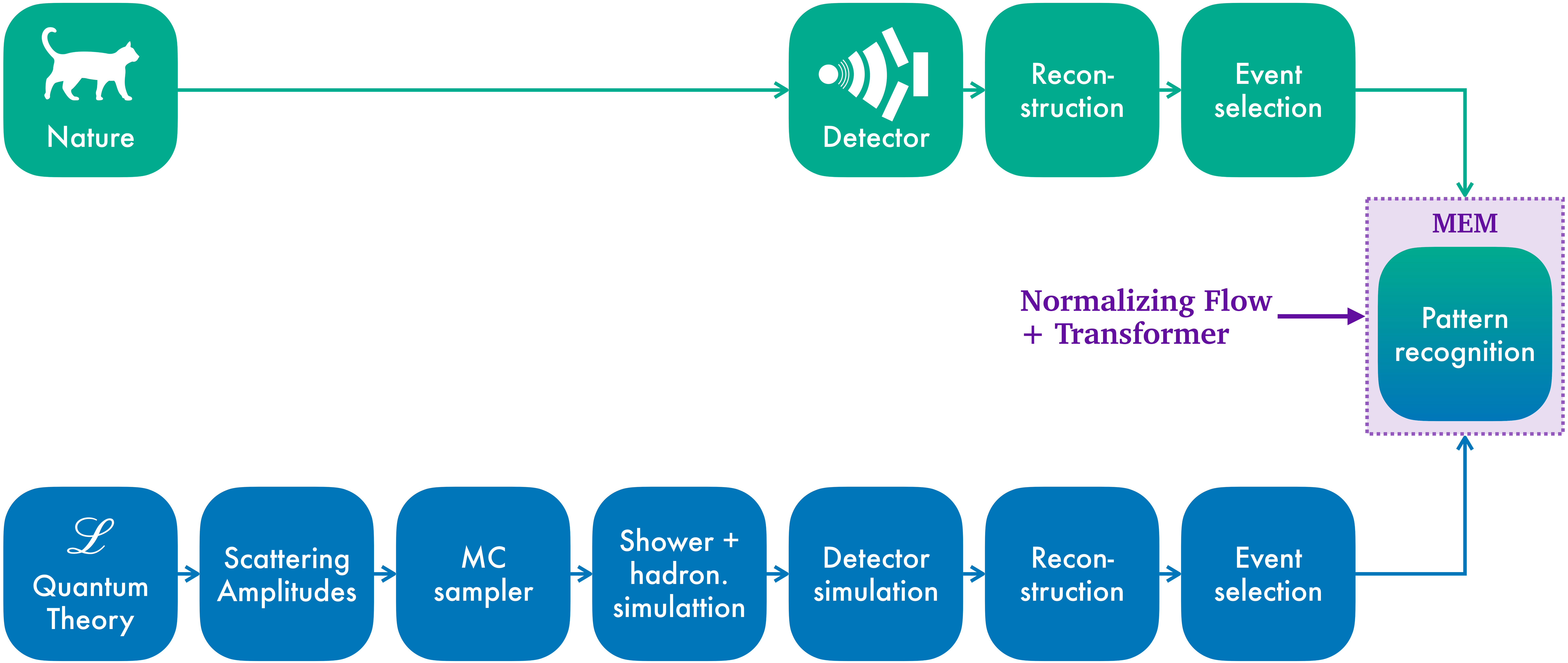
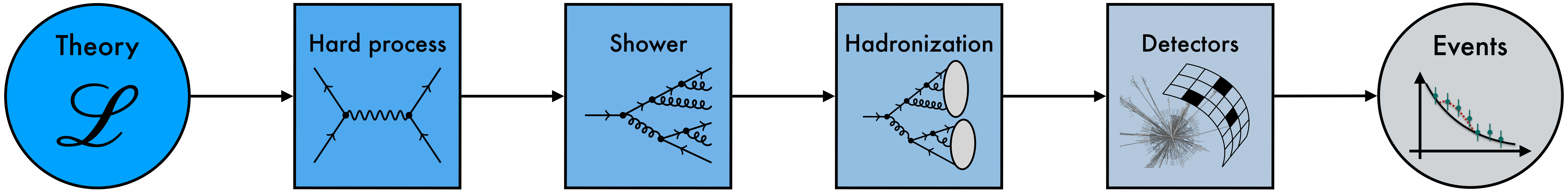


Figure adopted from [2405.14806]

Inference — Matrix element method



Theory parameter θ

Known from theory

Likelihood intractable
→ learn with neural network

Reconstructed momenta x_{reco}

$$p(x_{\text{reco}}|\theta) = \int \mathrm{d}x_{\text{hard}}\; p(x_{\text{hard}}|\theta)\; p(x_{\text{reco}}|x_{\text{hard}})\; \epsilon(x_{\text{hard}})$$

Efficient MC integration

importance sampling:
Normalizing Flow

$$x_{\text{hard}} \sim p(x_{\text{hard}}|x_{\text{reco}}, \theta)$$

Theory knowledge

diff. cross-section

$$\frac{1}{\sigma(\alpha)} \frac{\mathrm{d}\sigma(\alpha)}{\mathrm{d}x_{\text{hard}}}$$

Transfer function

density estimation:
Normalizing Flow
+ Transformer

Acceptance function

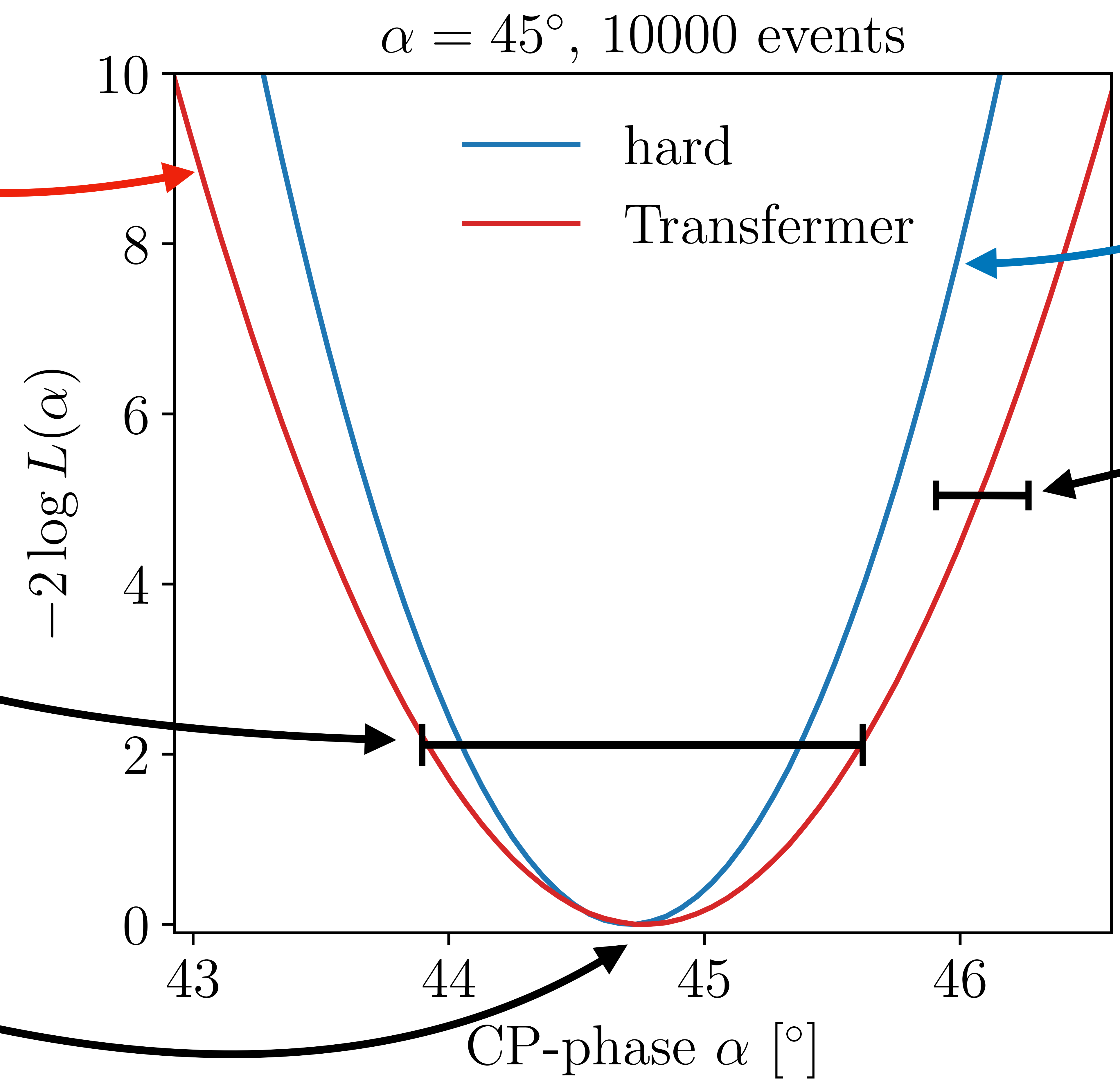
learn with simple
classifier network

Inference — Matrix element method

result from
our method

statistical
uncertainty

minimum:
measured value

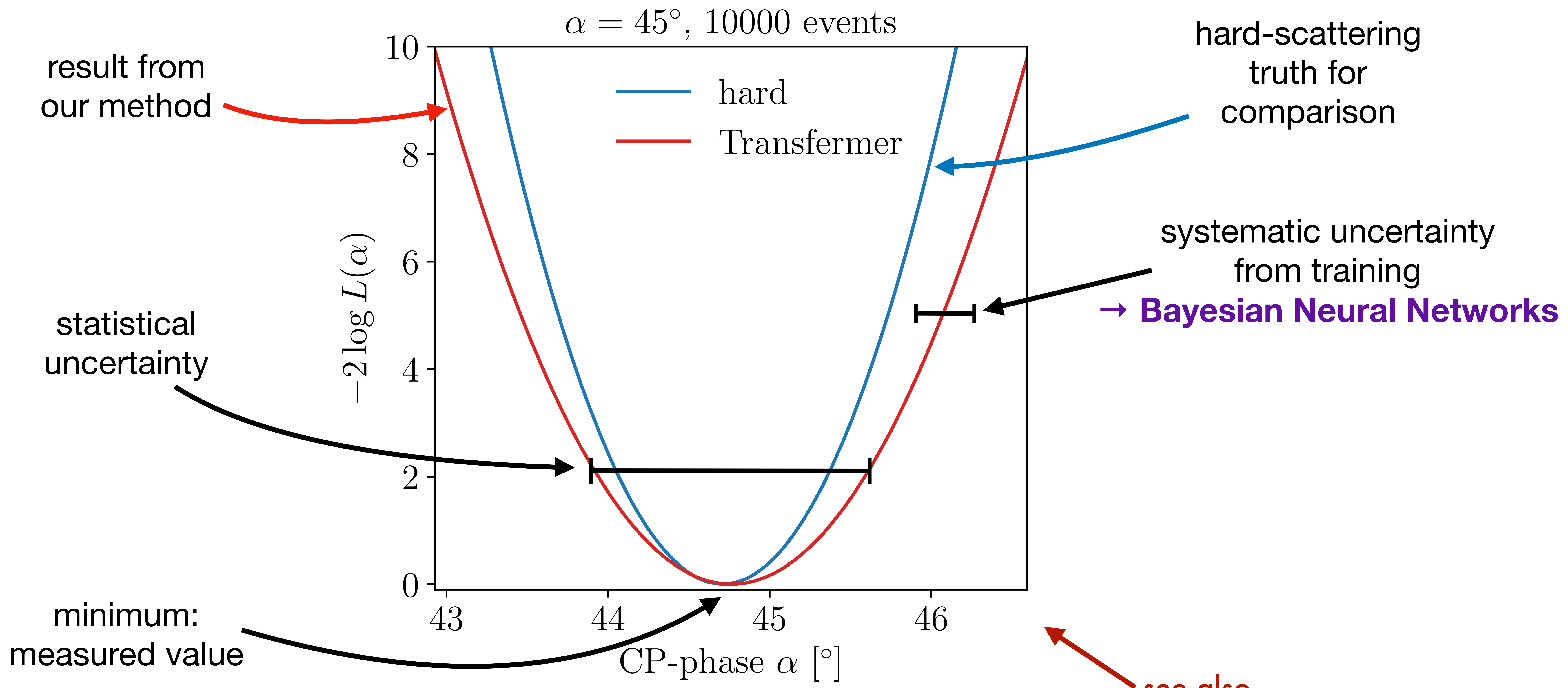


hard-scattering
truth for
comparison

systematic uncertainty
from training
→ **Bayesian Neural Networks**

[2210.00019, 2310.07752]

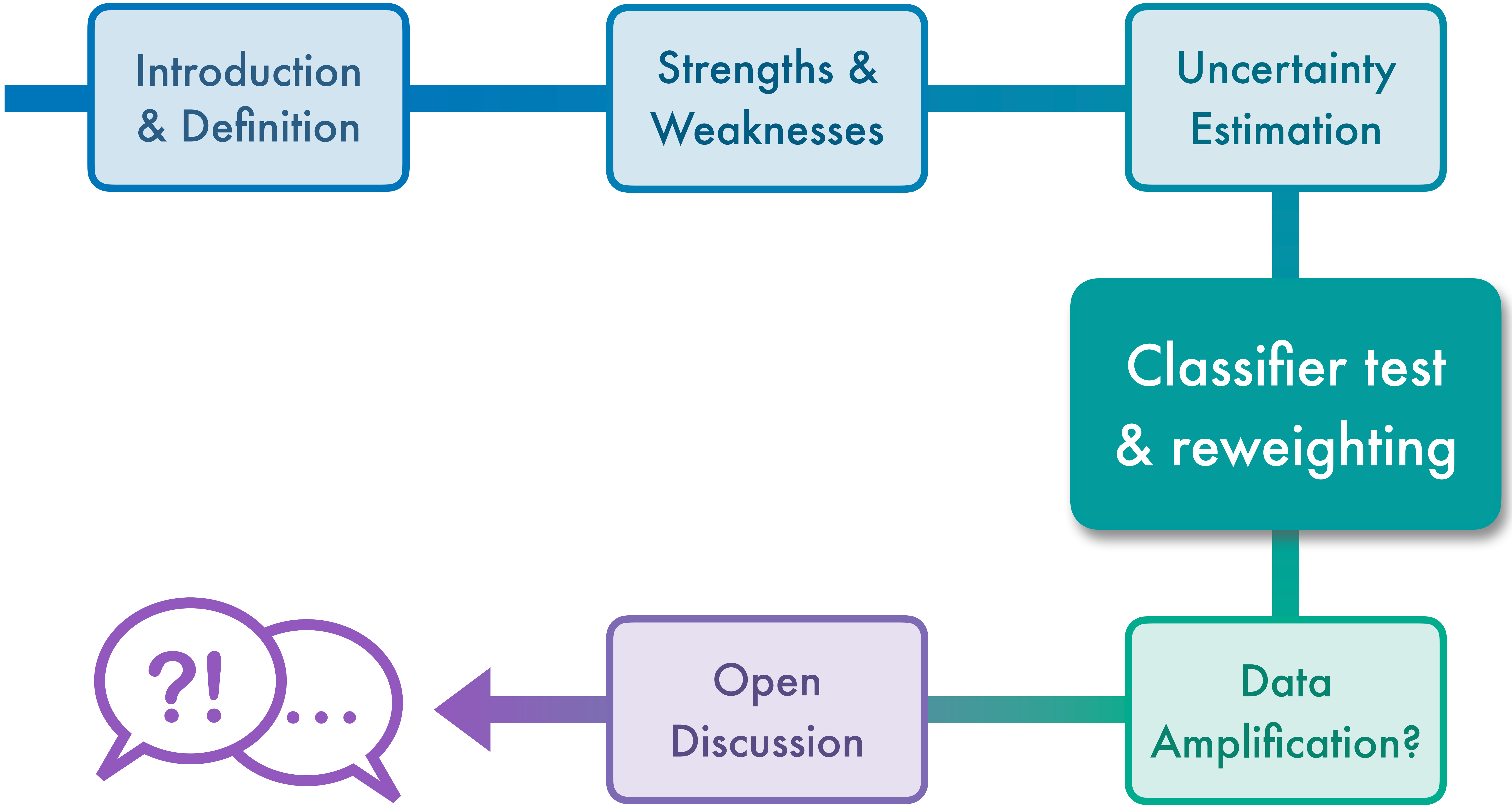
Inference — Matrix element method



see also
our poster (link)

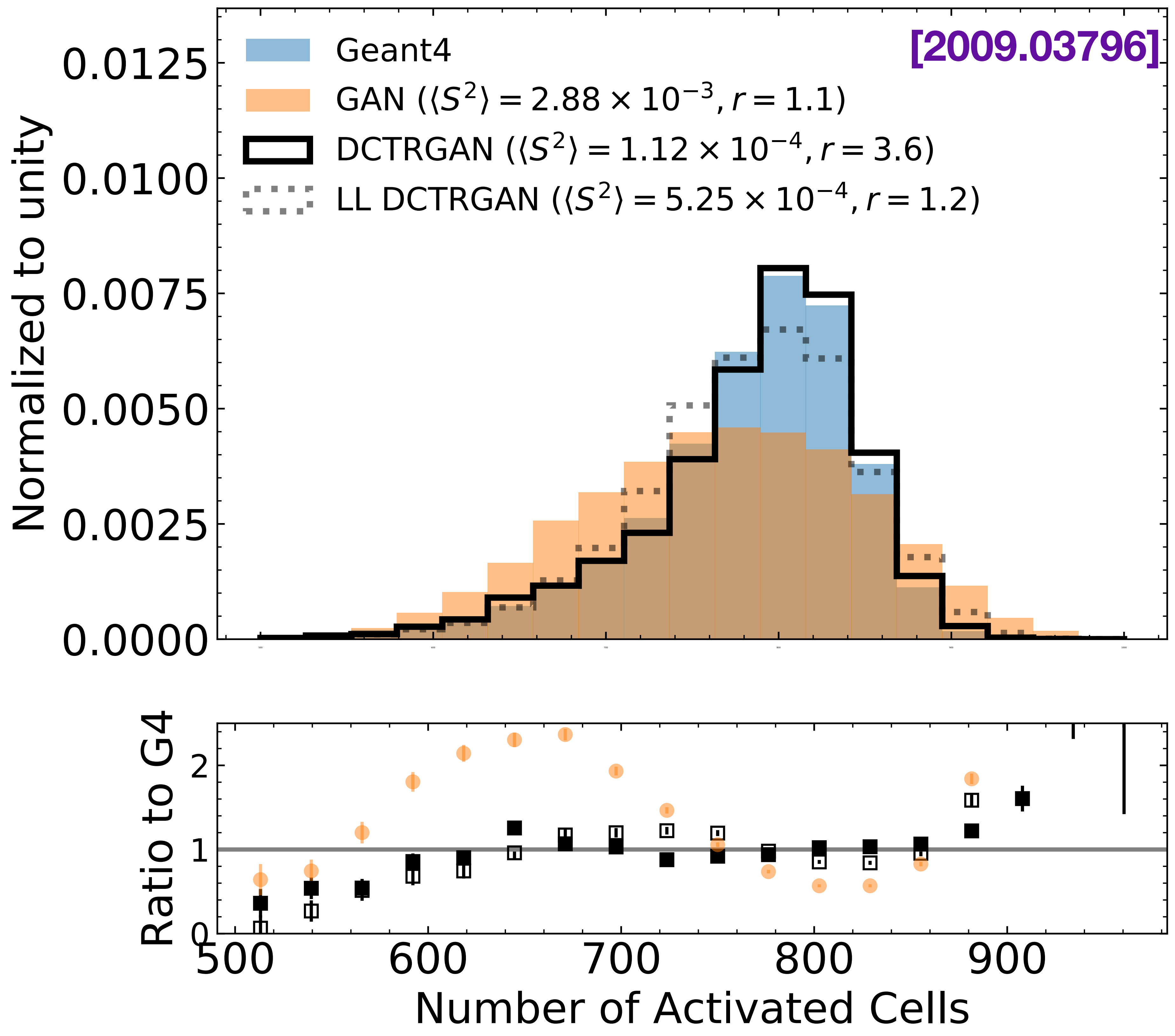
[2210.00019, 2310.07752]

Classifier test & reweighting



Reweighting and refinement

DCTRGAN



⊕ Additional classifier improves precision

DctrGAN [1907.08209, 2009.03796],
Laser [2106.00792, 2305.07696]

Classifier

If we have samples from **data** and **DGM…**

…an **optimal classifier** yields

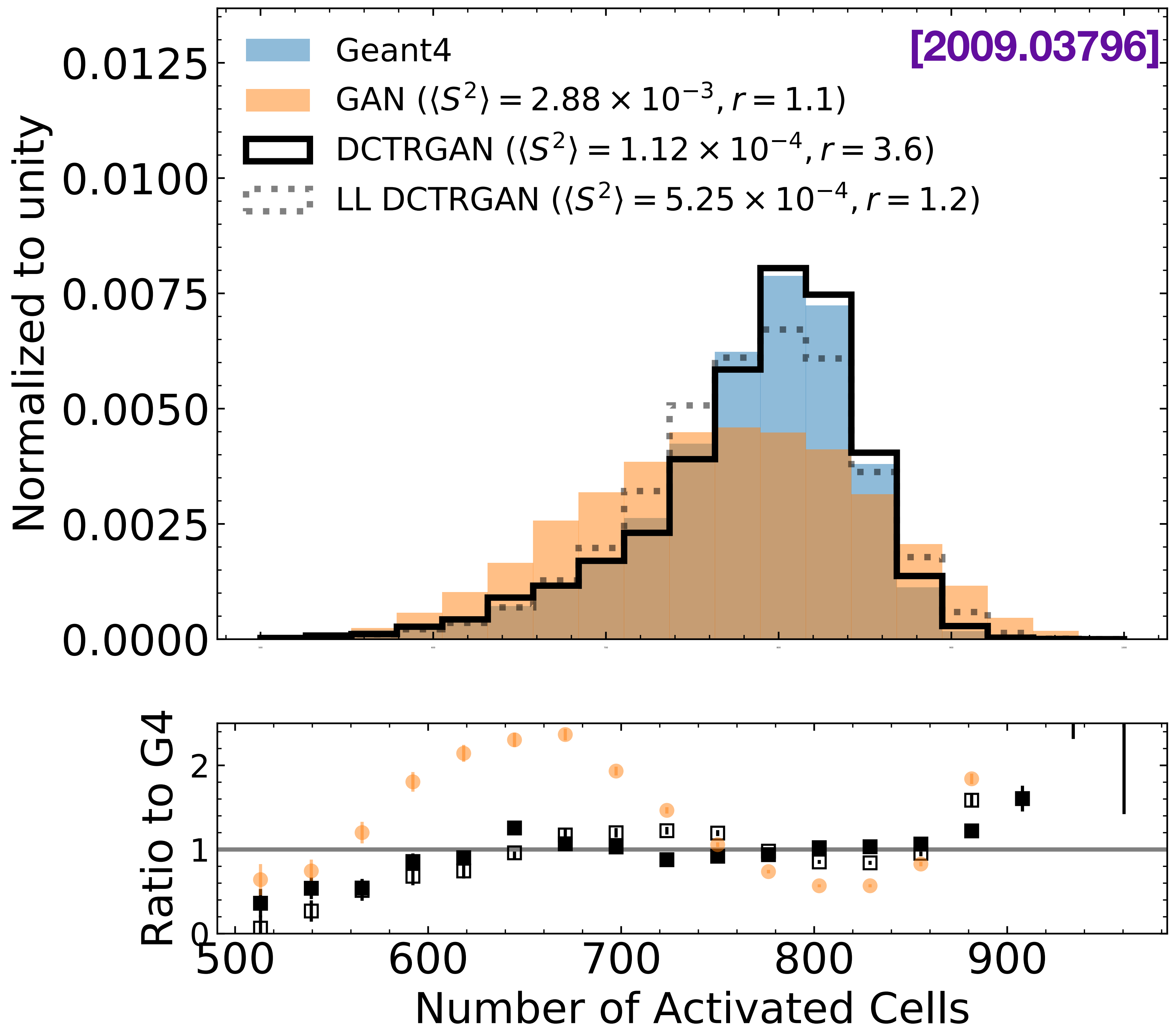
$$f(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}$$

and defines weight

$$w(x) = \frac{p_{\text{data}}(x)}{p_G(x)} = \frac{f(x)}{1 - f(x)}$$

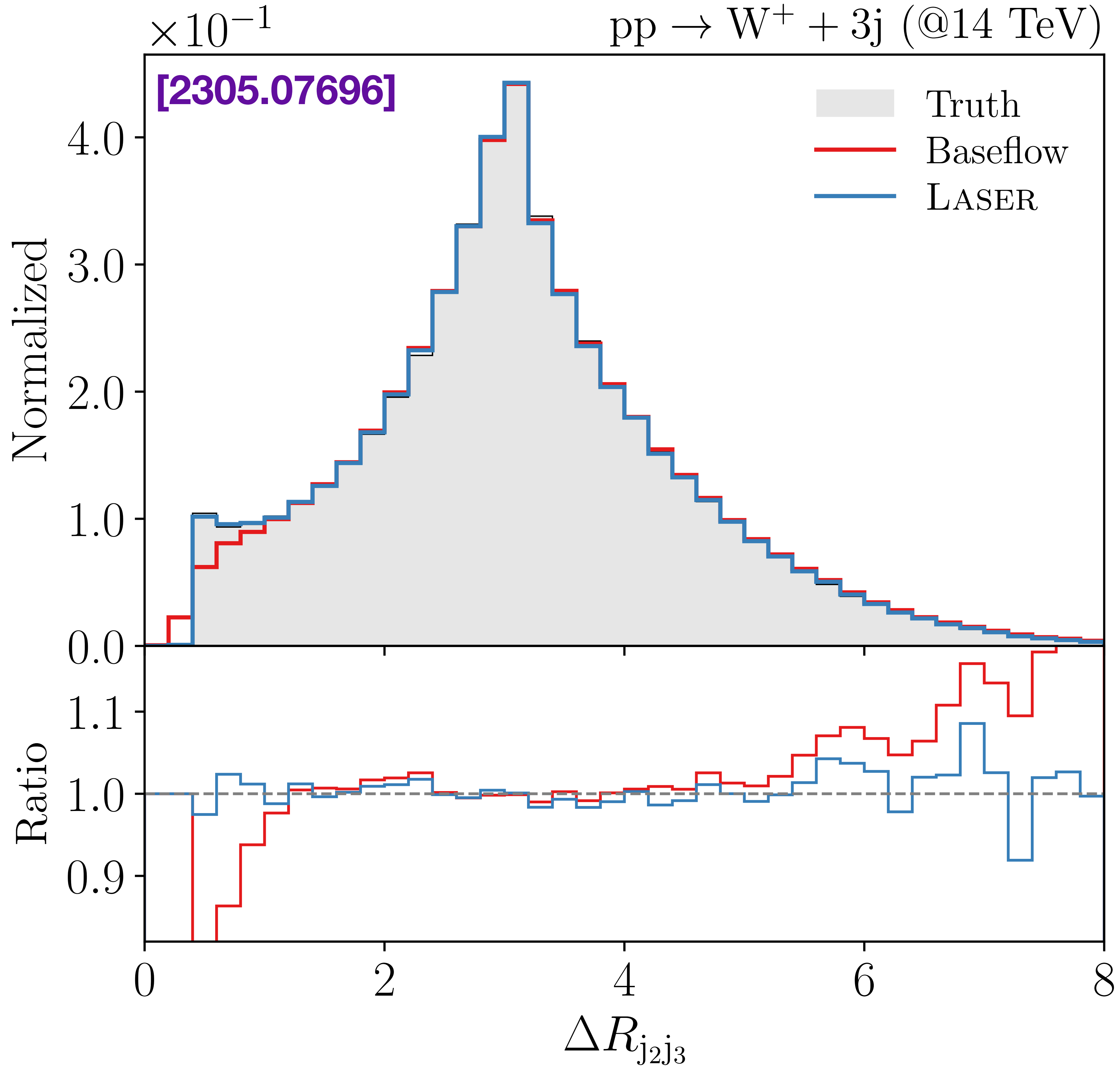
Reweighting and refinement

DCTRGAN



pull weights to latent space

LASER

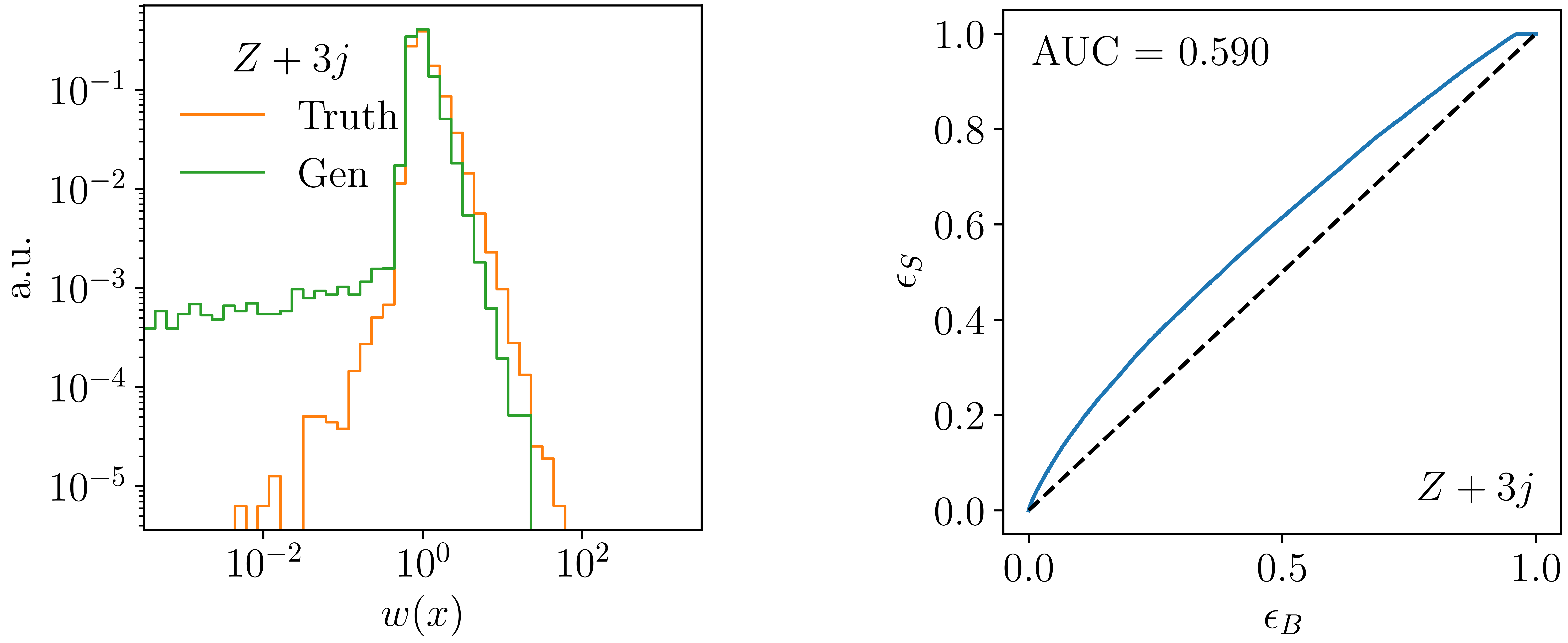


⊕ Additional classifier improves precision

⊕ Laser gives unweighted events

DctrGAN [1907.08209, 2009.03796],
Laser [2106.00792, 2305.07696]

Classifier test

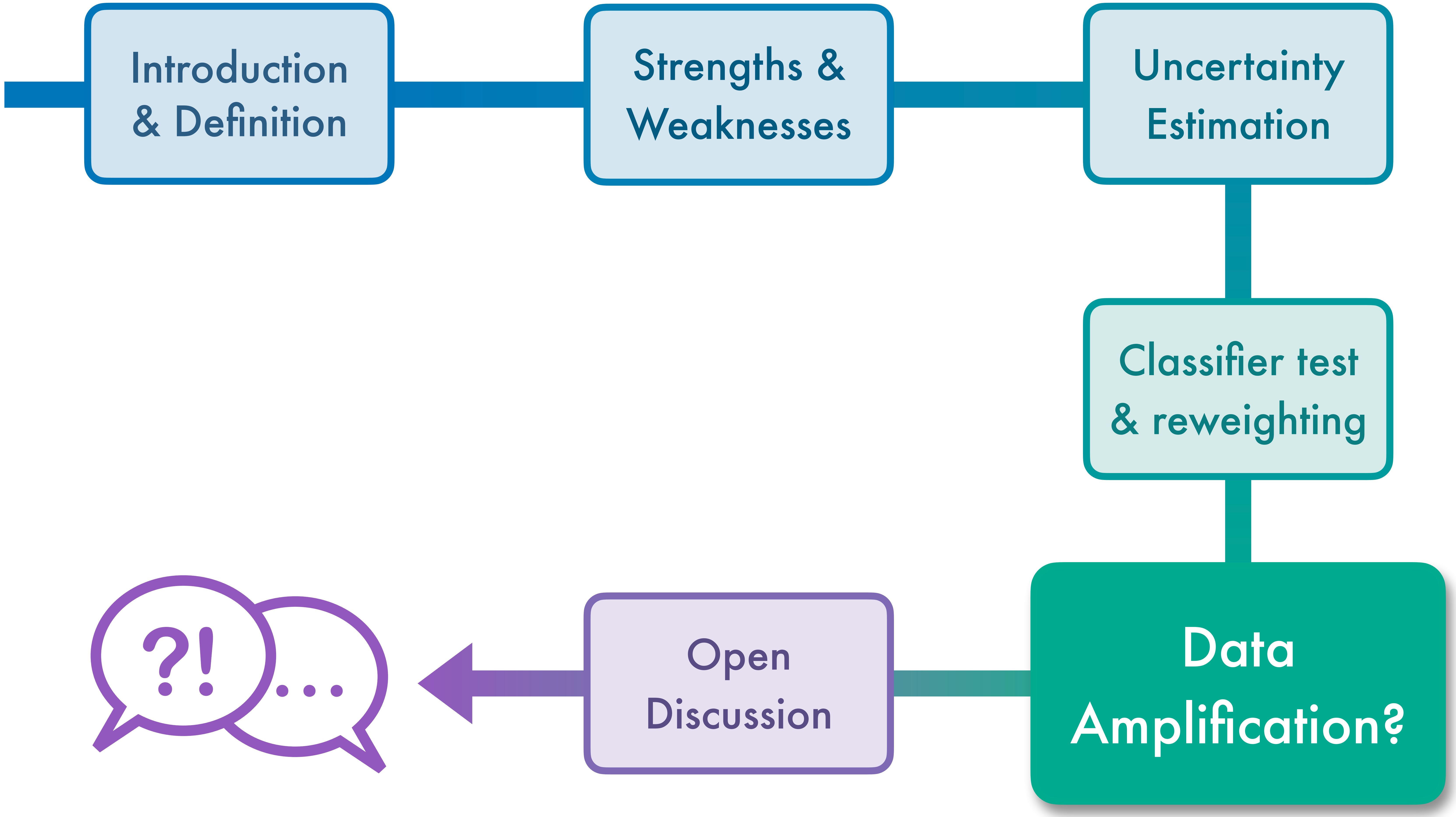


→ classifier detects **subtle difference** between densities

→ yields a **valuable test metric**

[2305.16774]

Data Amplification

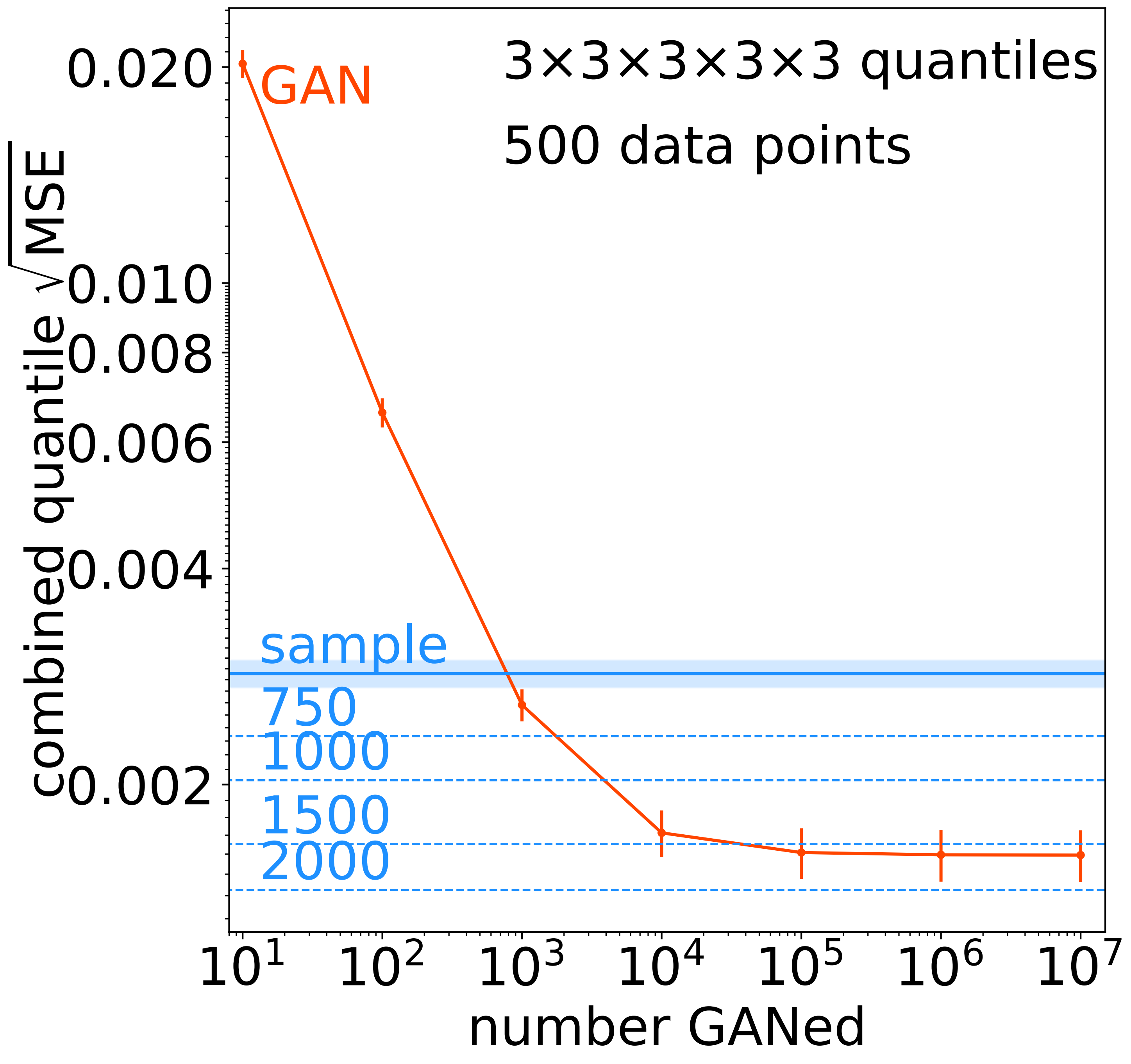


Discussion Topic

Given N training samples, how many more events $N_g > N$ can the DGM generate that can effectively increase the statistics?

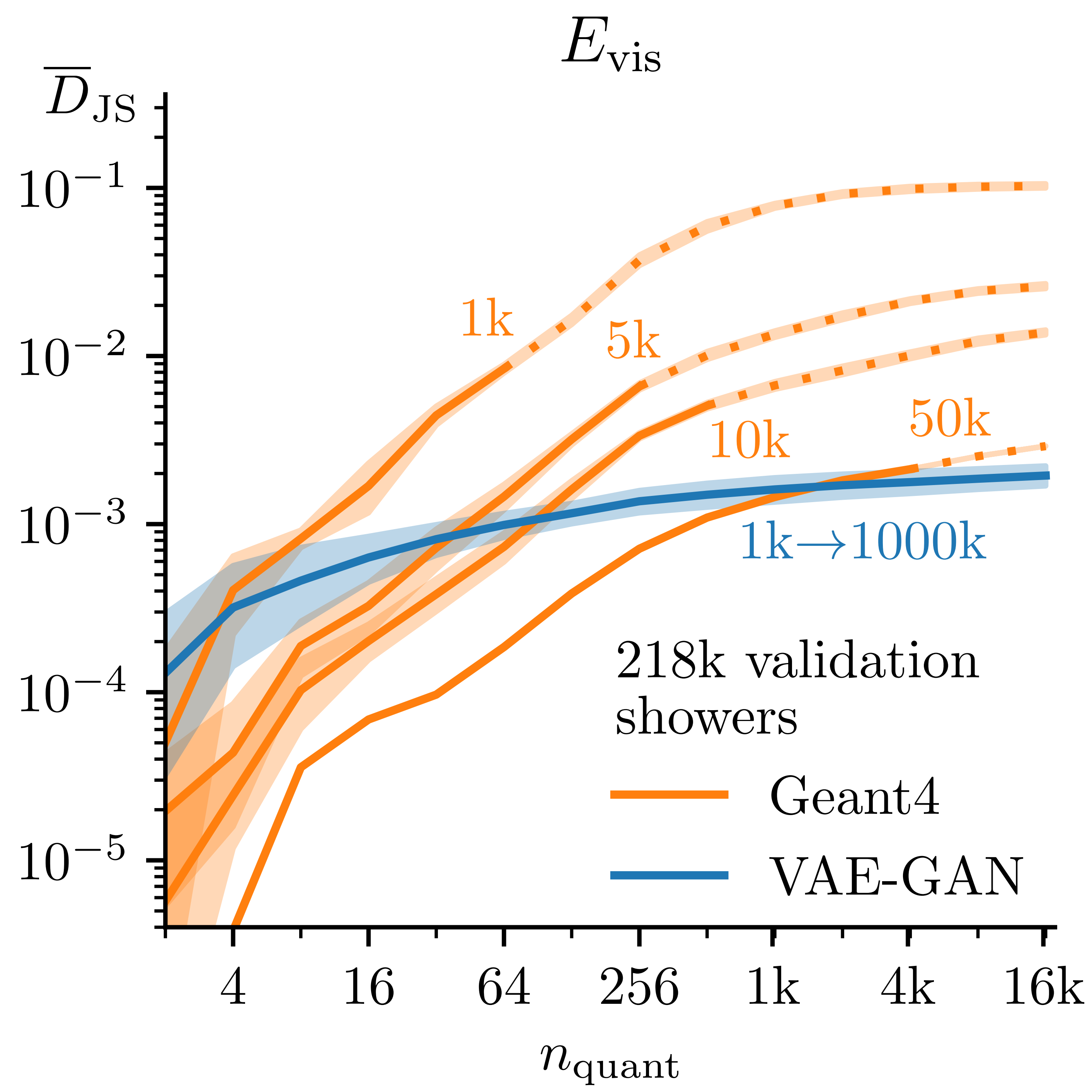
GANplification

5D Gaussian



[2008.06545]

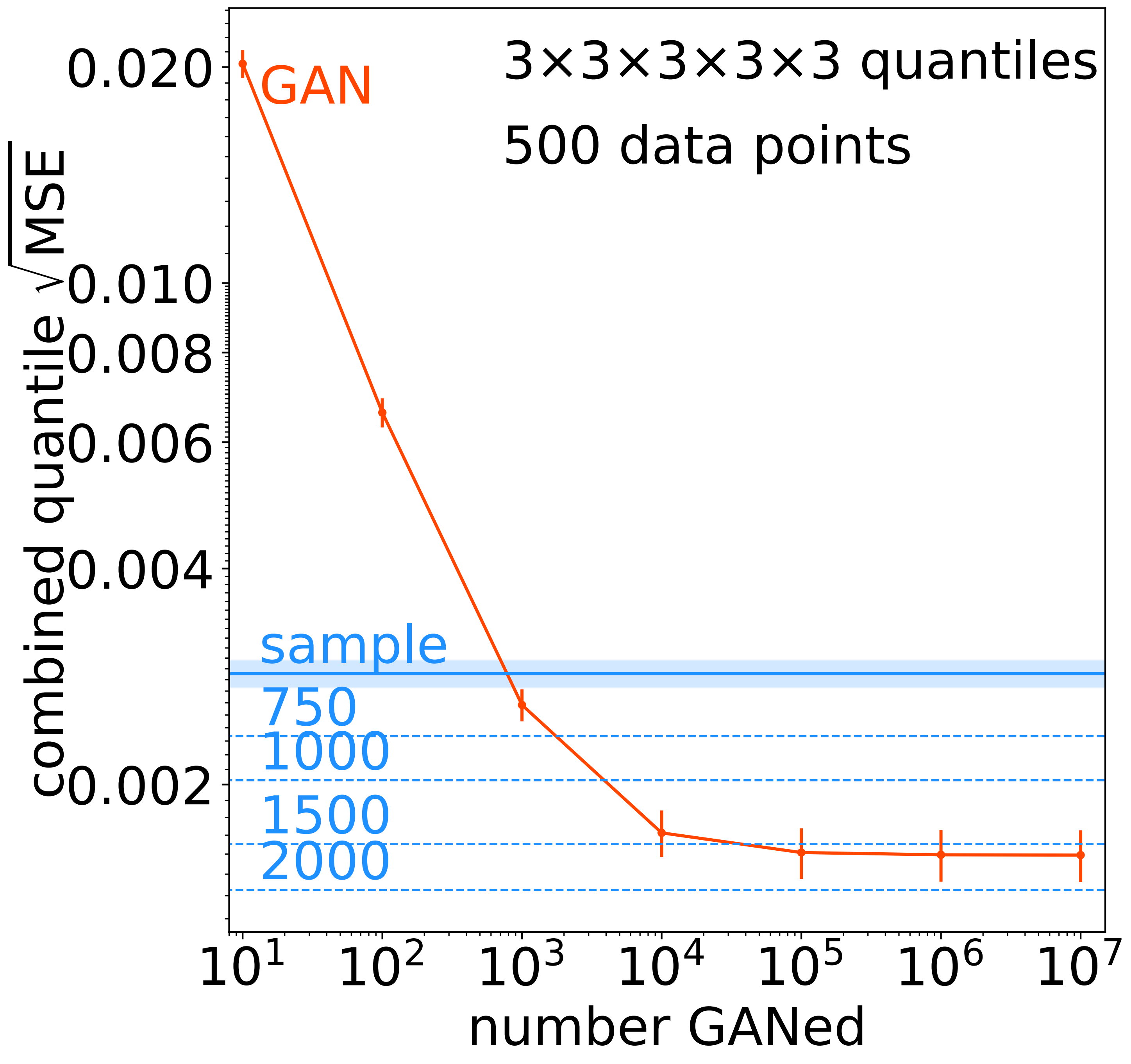
Calorimeter Simulation



[2202.07352]

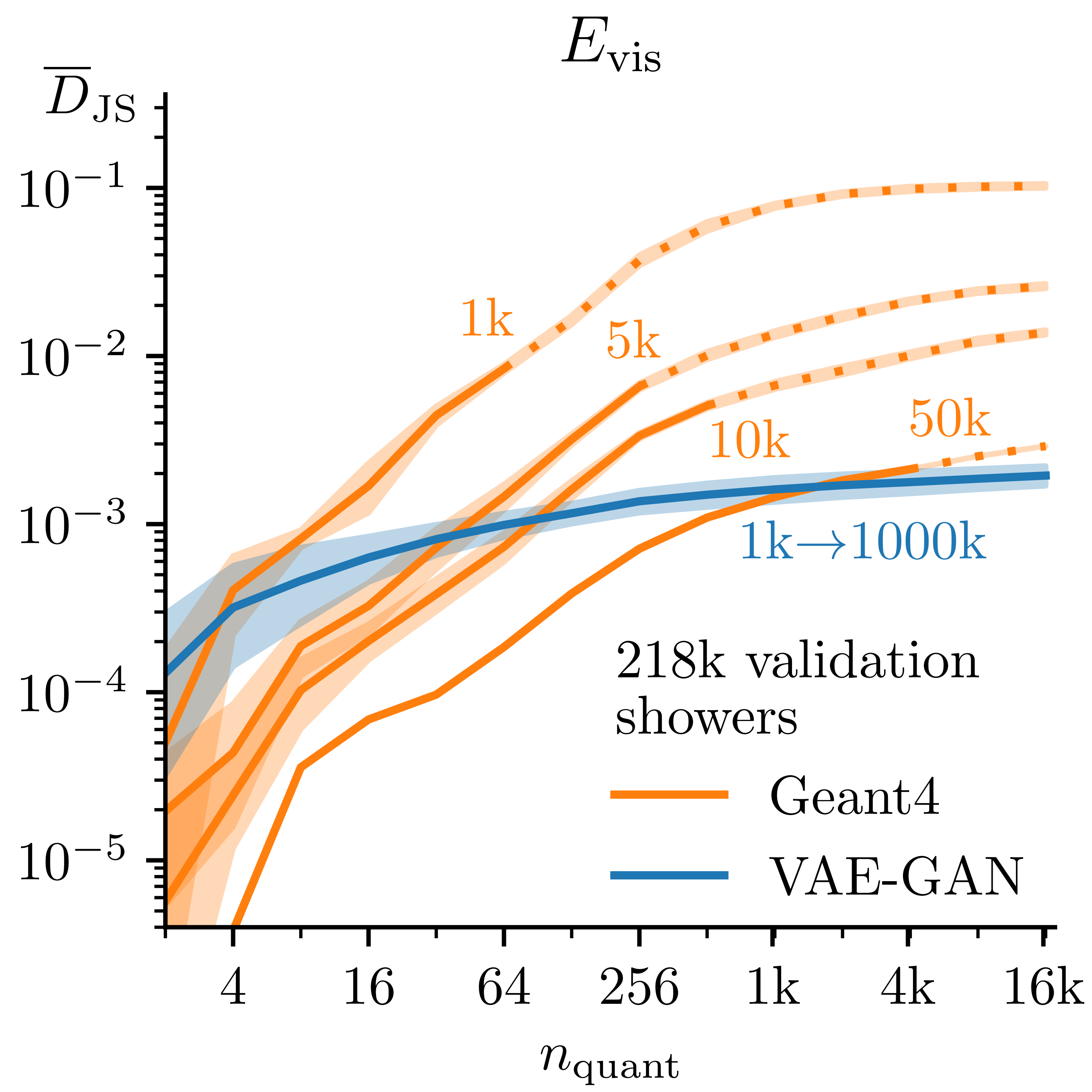
GANplification

5D Gaussian



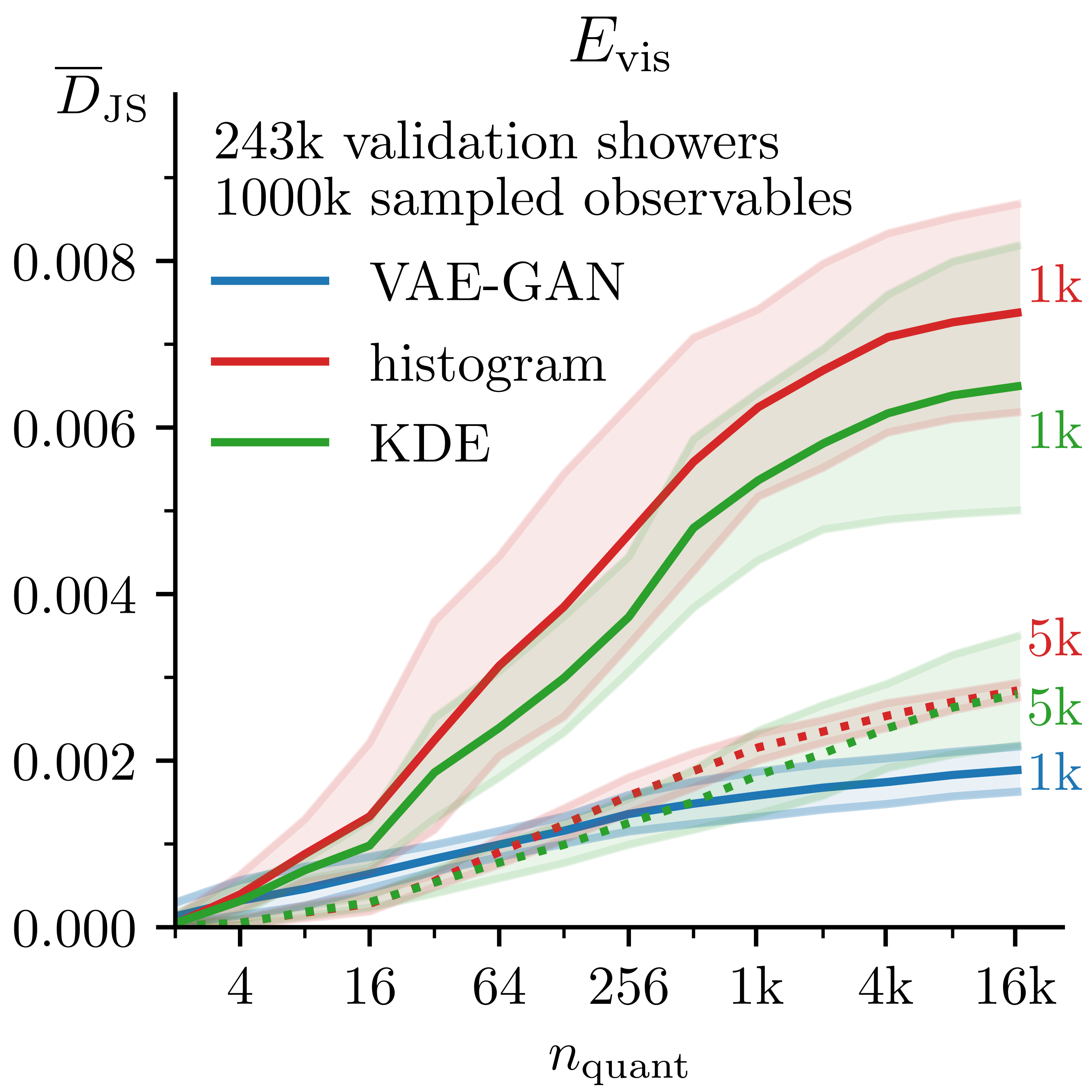
[2008.06545]

Calorimeter Simulation



[2202.07352]

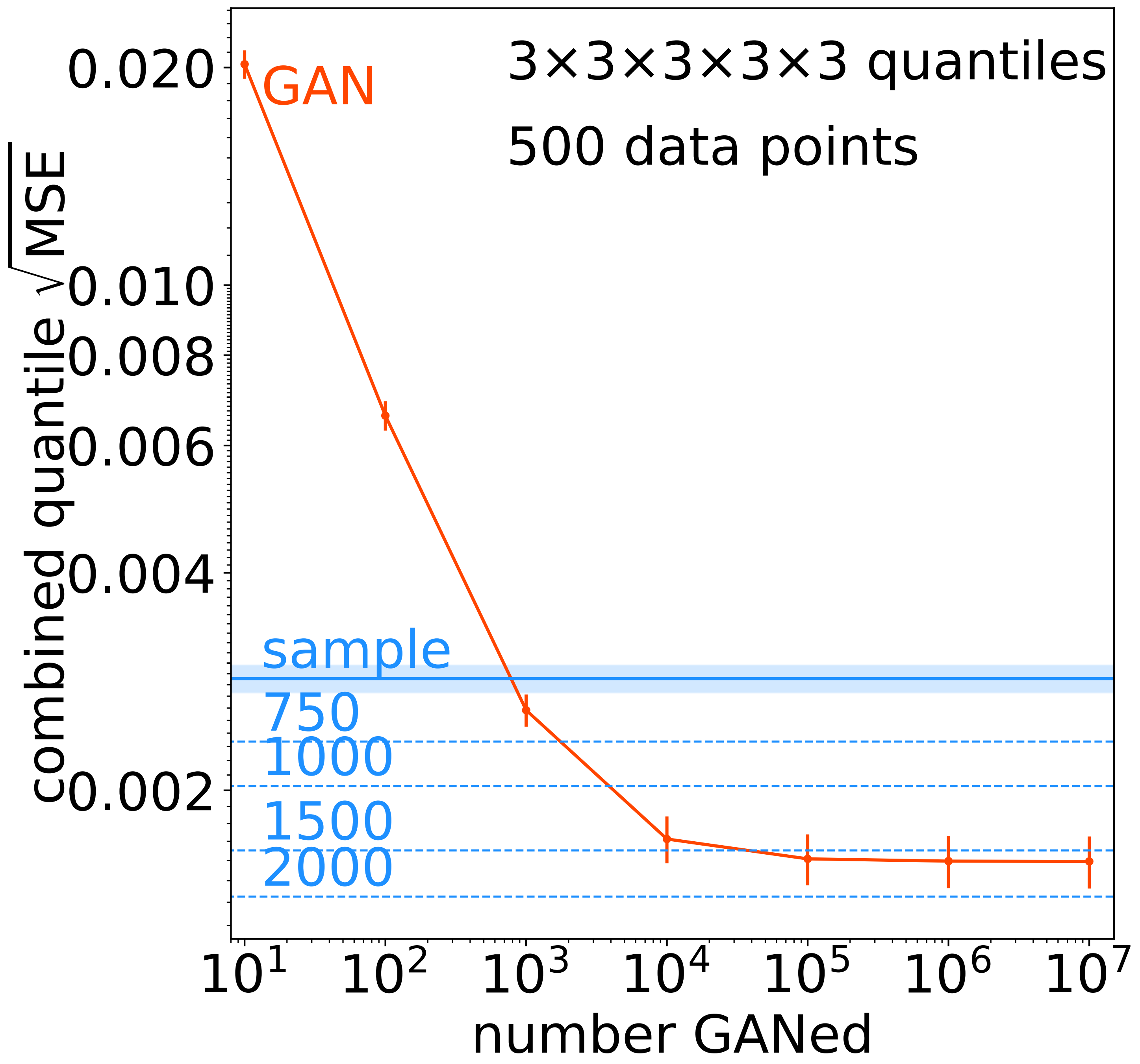
Method Comparison



[2202.07352]

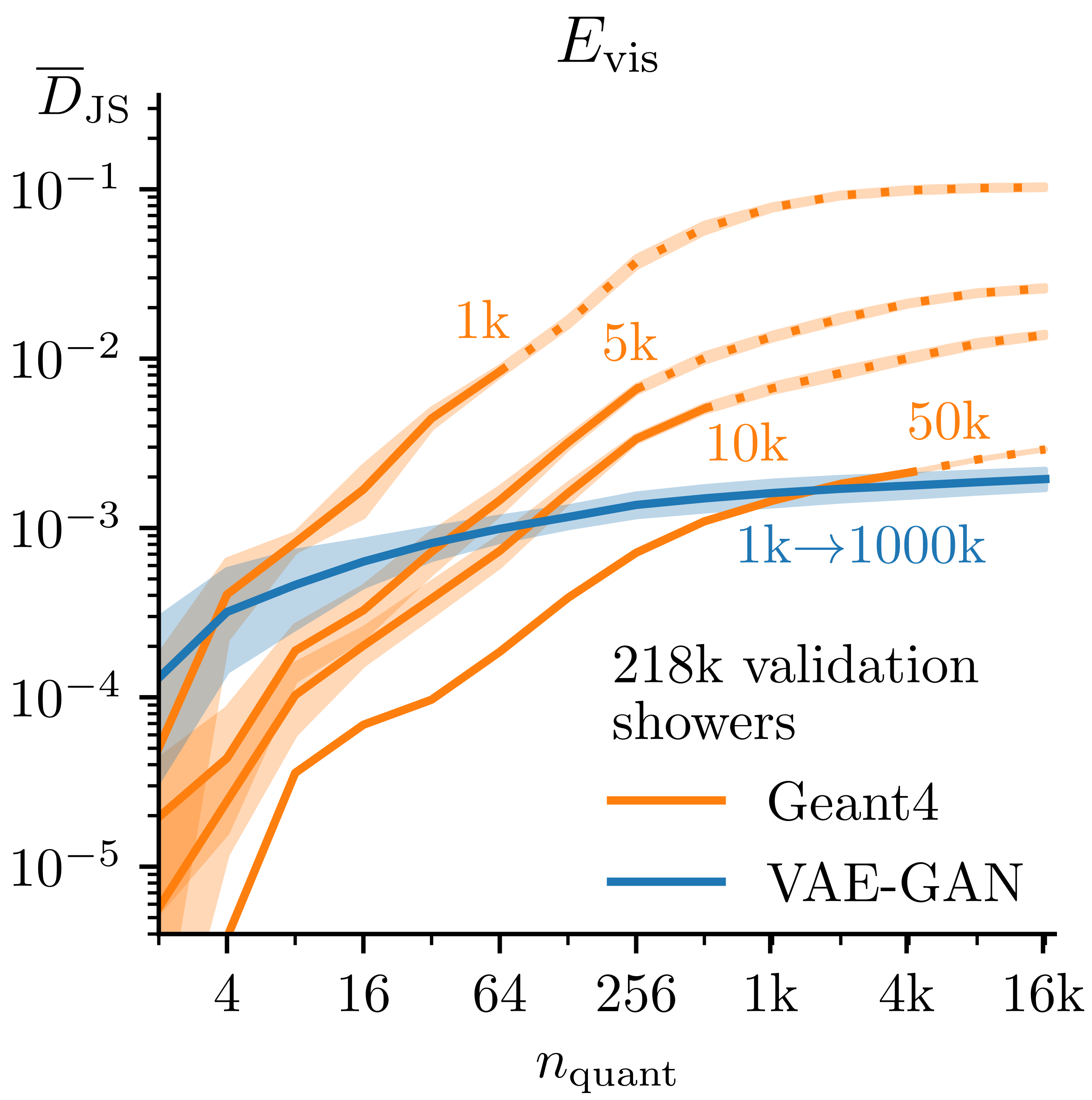
GANplification

5D Gaussian



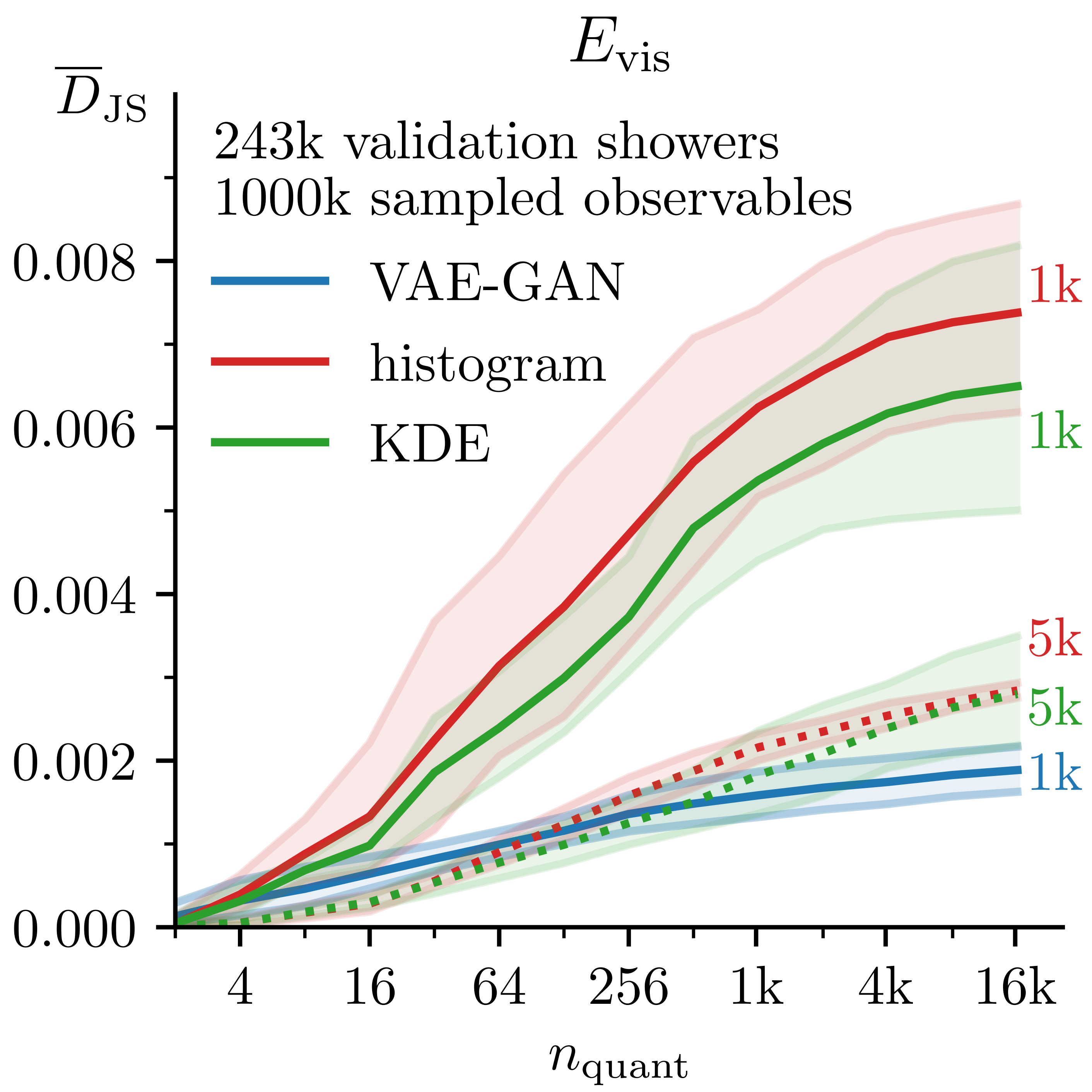
[2008.06545]

Calorimeter Simulation



[2202.07352]

Method Comparison

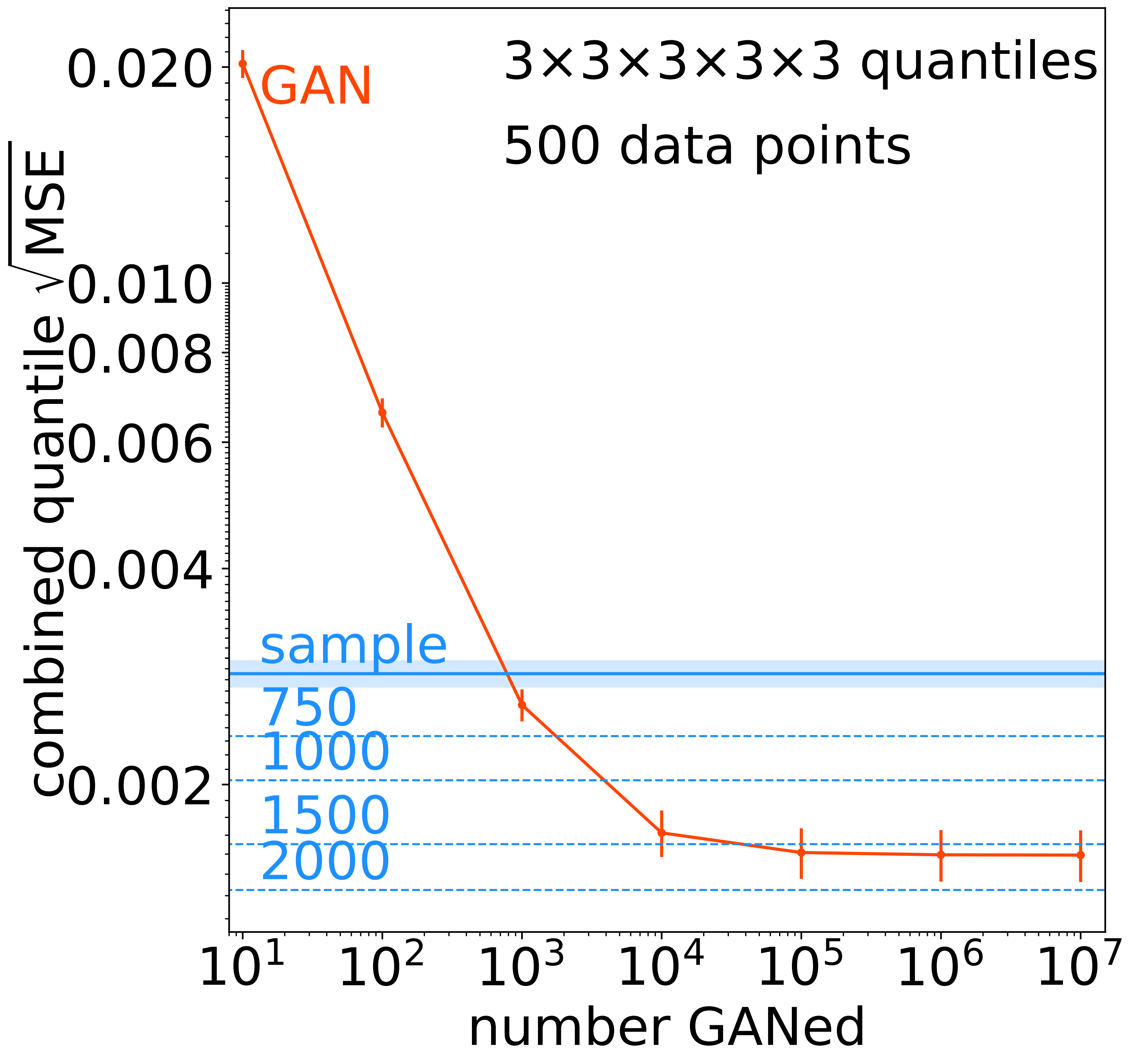


[2202.07352]

→ works for **different DGMs**

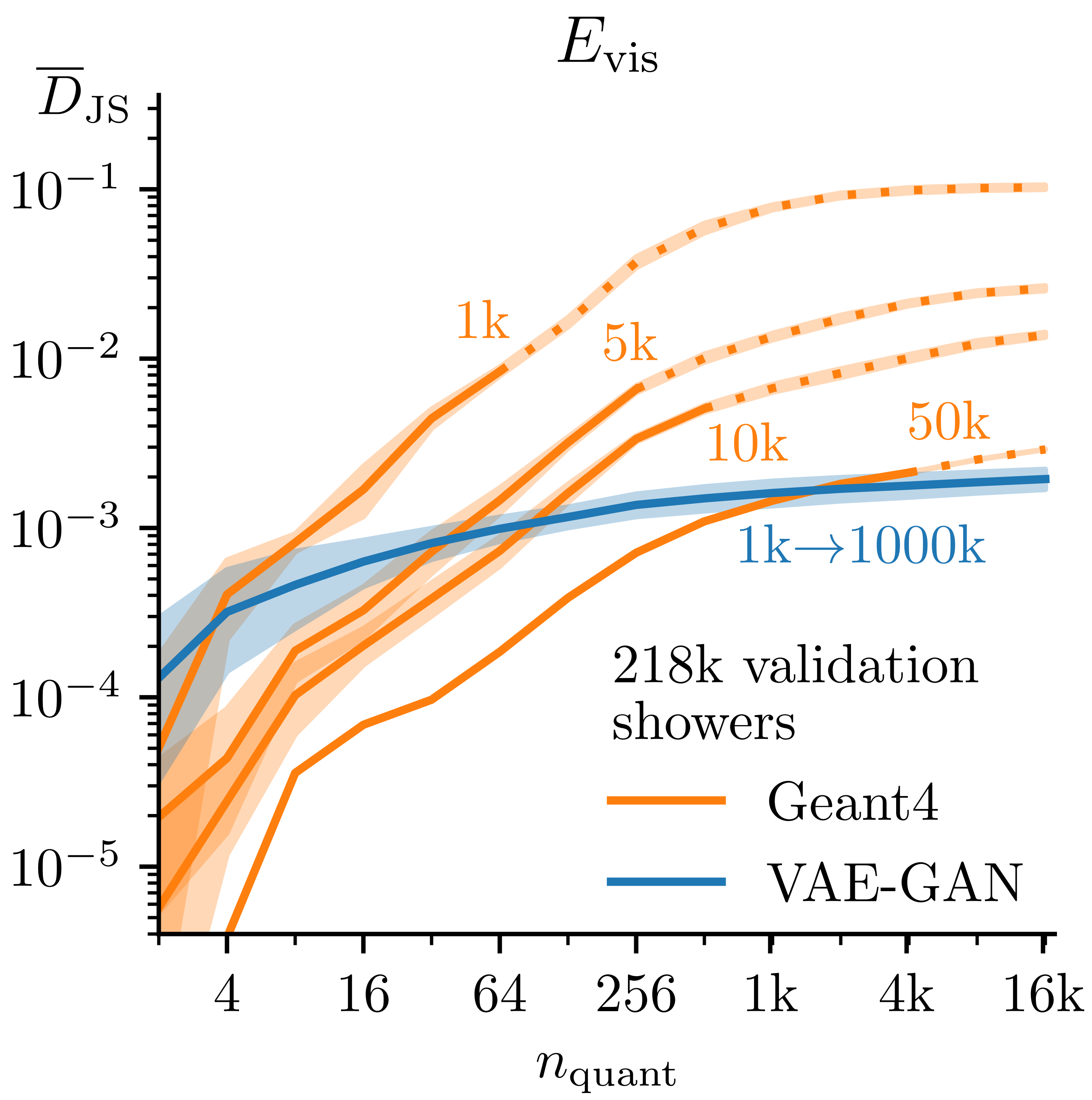
GANplification

5D Gaussian



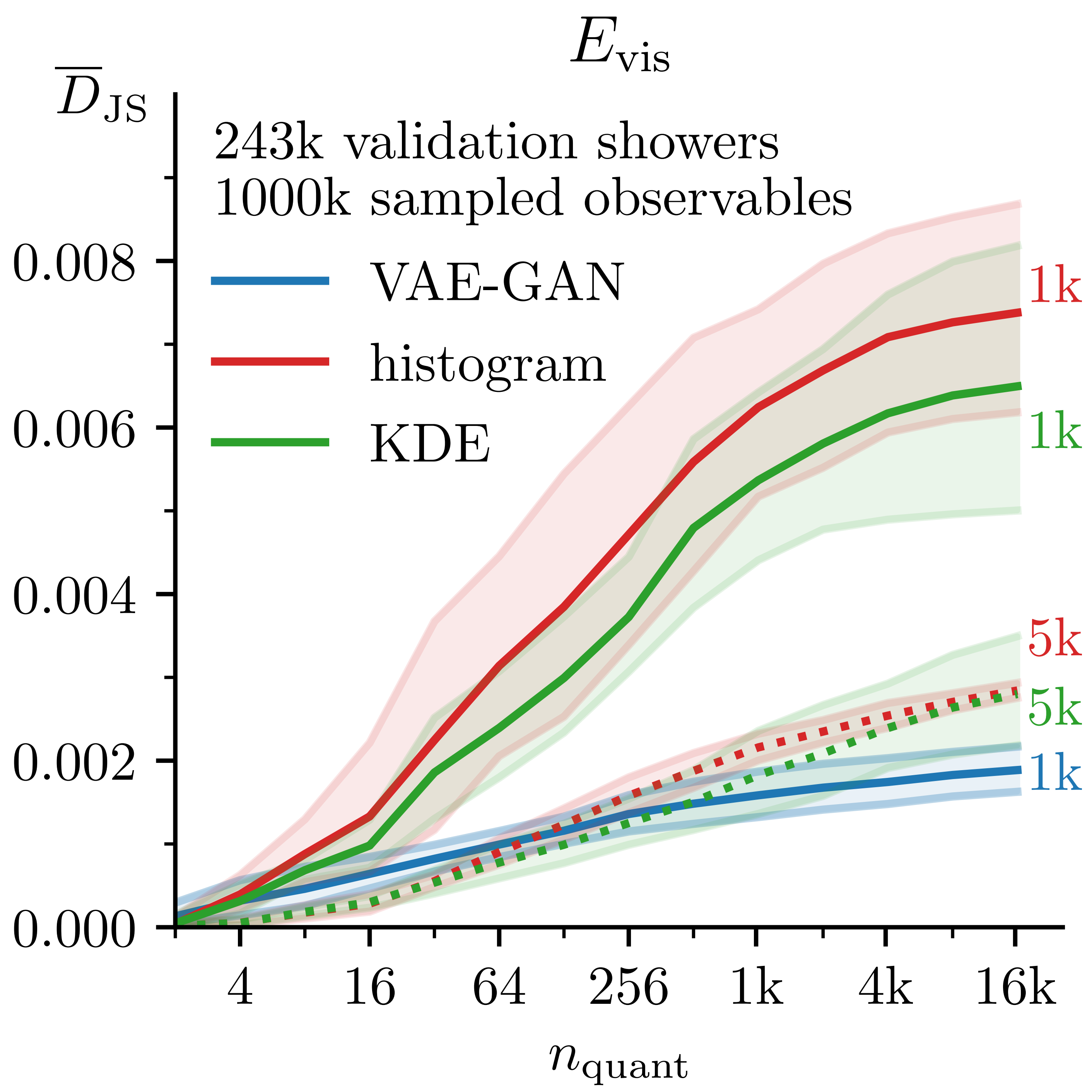
[2008.06545]

Calorimeter Simulation



[2202.07352]

Method Comparison

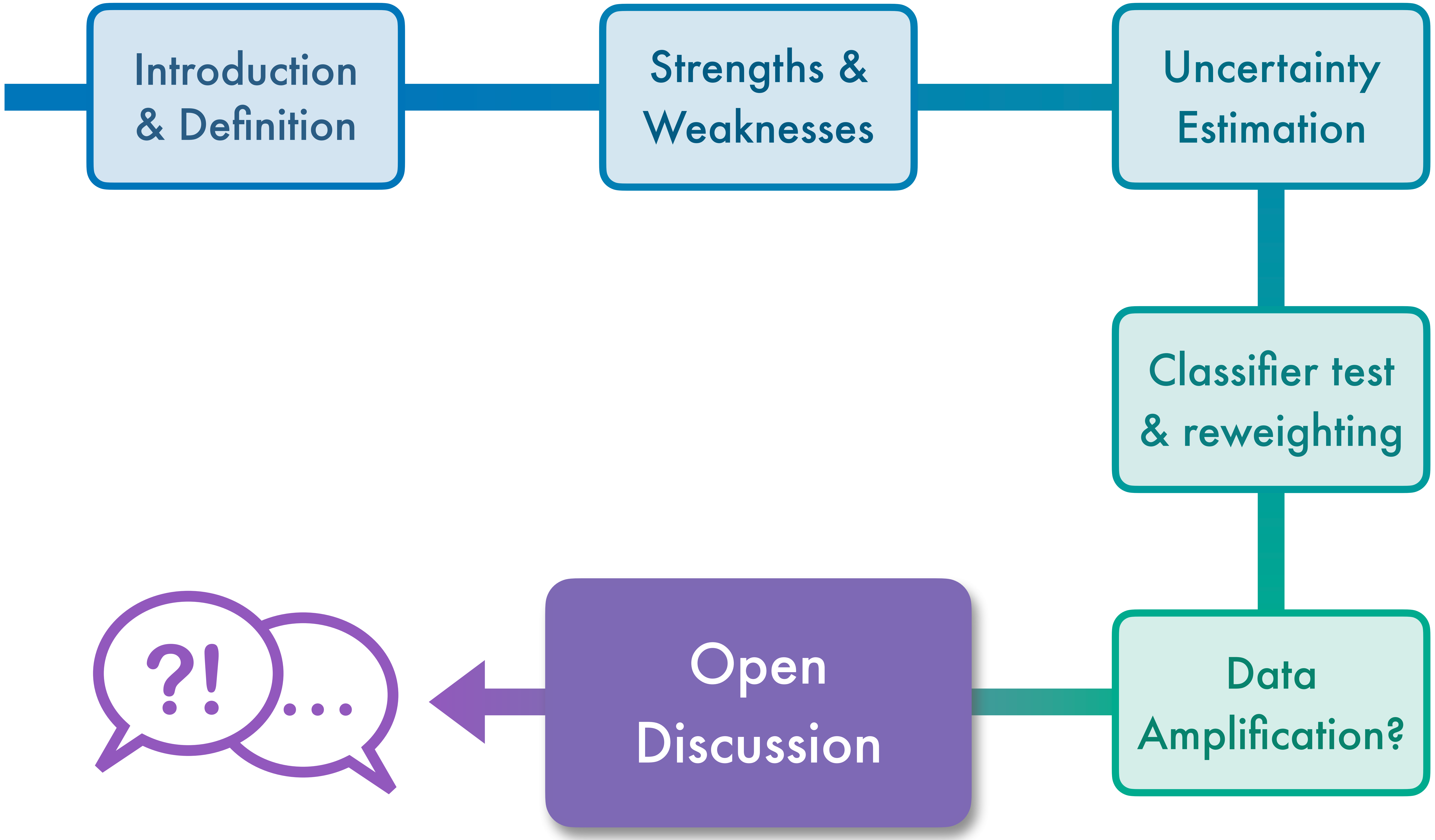


[2202.07352]

→ works for **different DGMs**

→ More details and link to BNNs, **see talk from G.Kasieczka and [2408.00838]**

Deep Generative Models



Open Discussion