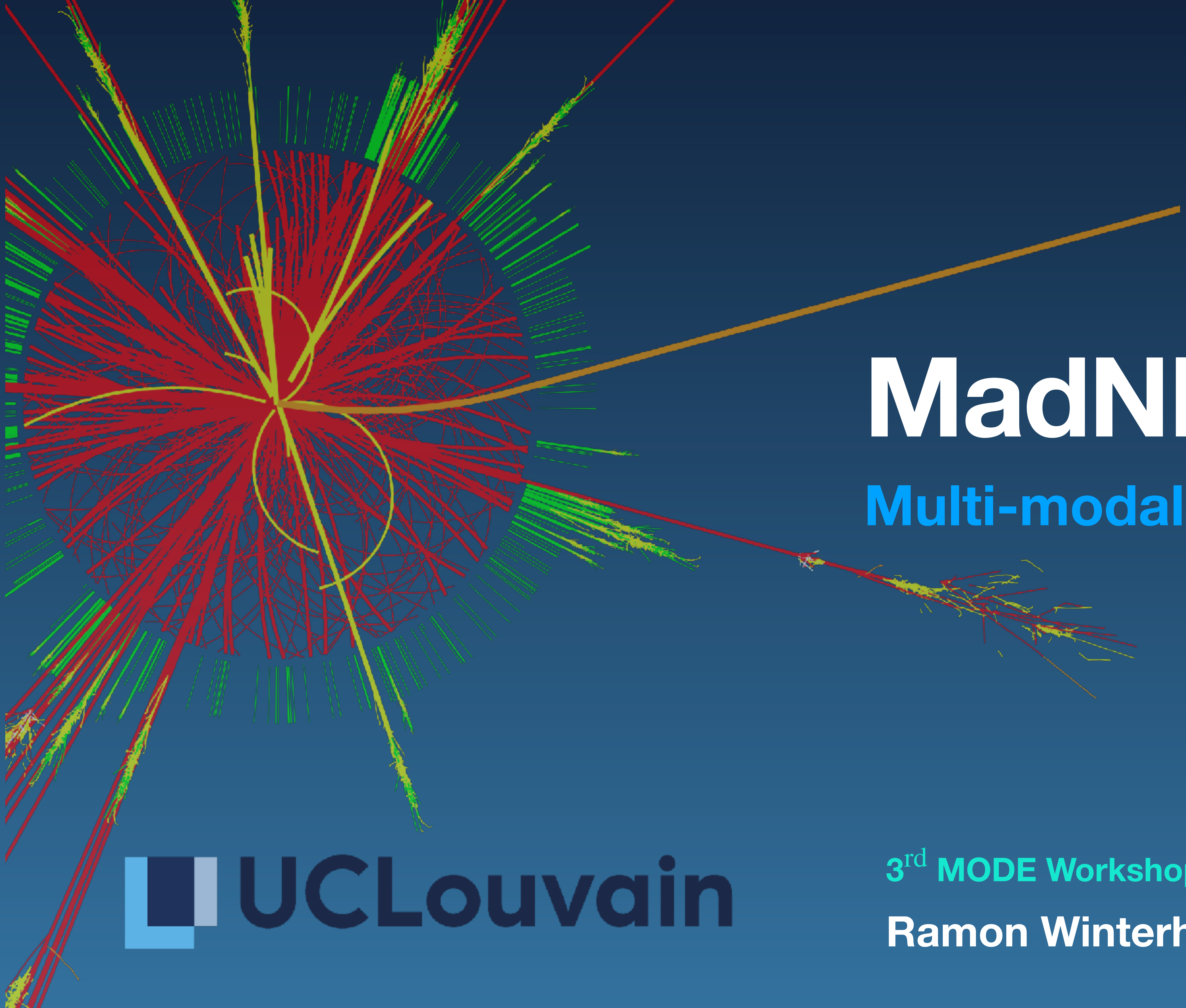


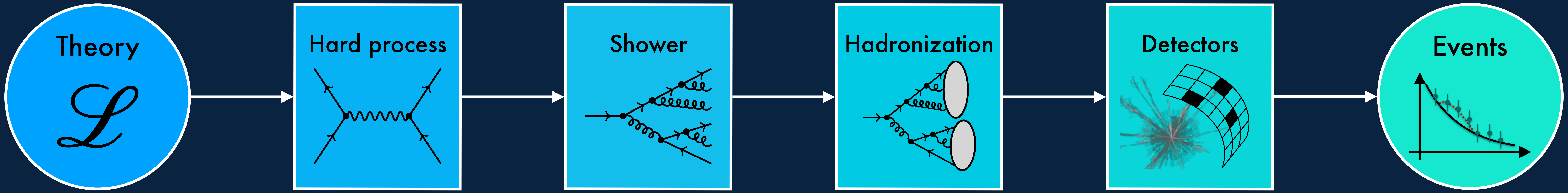
MadNIS and ELSA

Multi-modal event generation

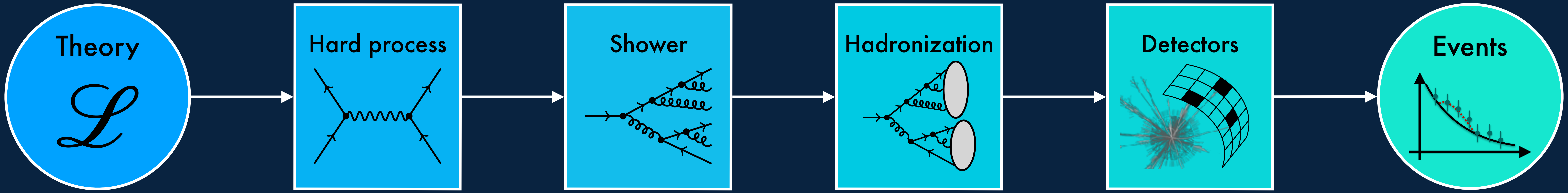
3rd MODE Workshop

Ramon Winterhalder — UCLouvain

LHC simulation chain



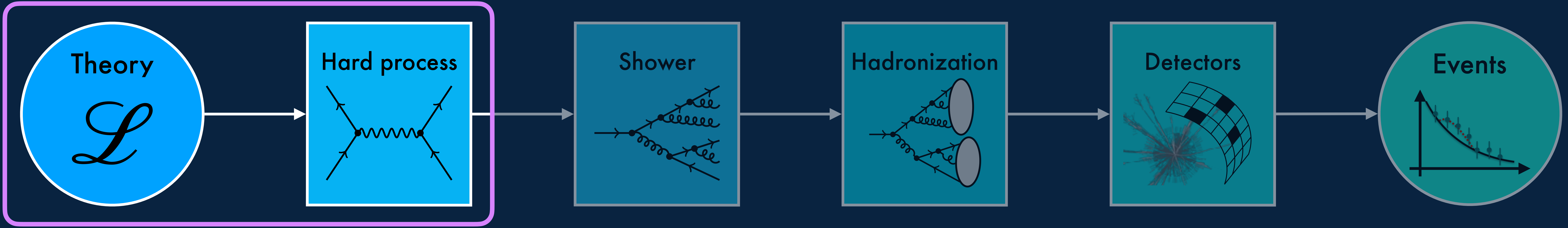
LHC simulation chain



How can we prevent MC event generation from becoming a bottleneck in future LHC runs?

LHC simulation chain

How can we prevent MC event generation from becoming a bottleneck in future LHC runs?



Differential cross section known from QFT

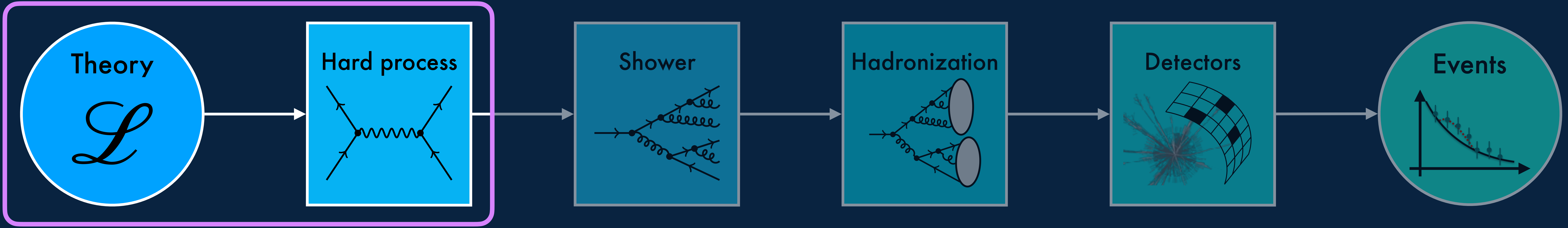
$$d\sigma \sim \mathrm{pdf}(x) \cdot |\mathcal{M}(x)|^2 \cdot d\Phi$$

Total cross section

$$\sigma = \int_\Phi d\sigma$$

LHC simulation chain

How can we prevent MC event generation from becoming a bottleneck in future LHC runs?



Differential cross section known from QFT

$$d\sigma \sim \text{pdf}(x) \cdot |\mathscr{M}(x)|^2 \cdot d\Phi$$

Total cross section

$$\sigma = \int_\Phi d\sigma$$

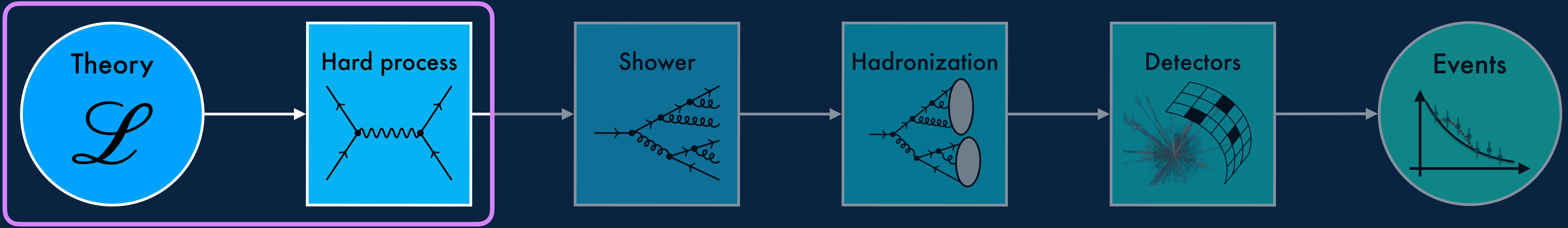
Monte Carlo integration and sampling from differential cross section

↓

accelerate with deep generative models

LHC simulation chain

How can we prevent MC event generation from becoming a bottleneck in future LHC runs?



Differential cross section known from QFT

$$d\sigma \sim \mathrm{pdf}(x) \cdot |\mathscr{M}(x)|^2 \cdot d\Phi$$

Total cross section

$$\sigma = \int_\Phi d\sigma$$

Monte Carlo integration and sampling from differential cross section

↓

accelerate with deep generative models

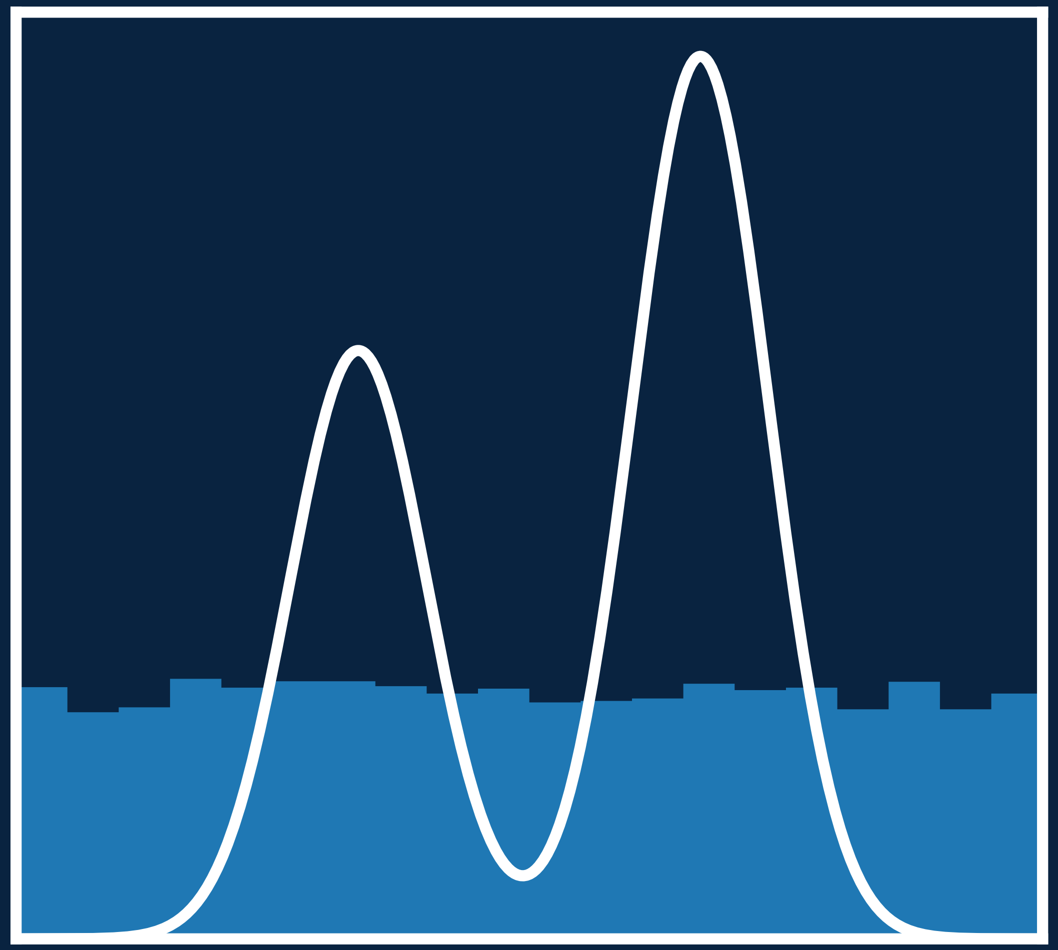
Exact sampling ensured by known likelihood

↓

better model = faster sampling

Monte Carlo integration

$$I = \int \mathrm{d}x\, f(x)$$

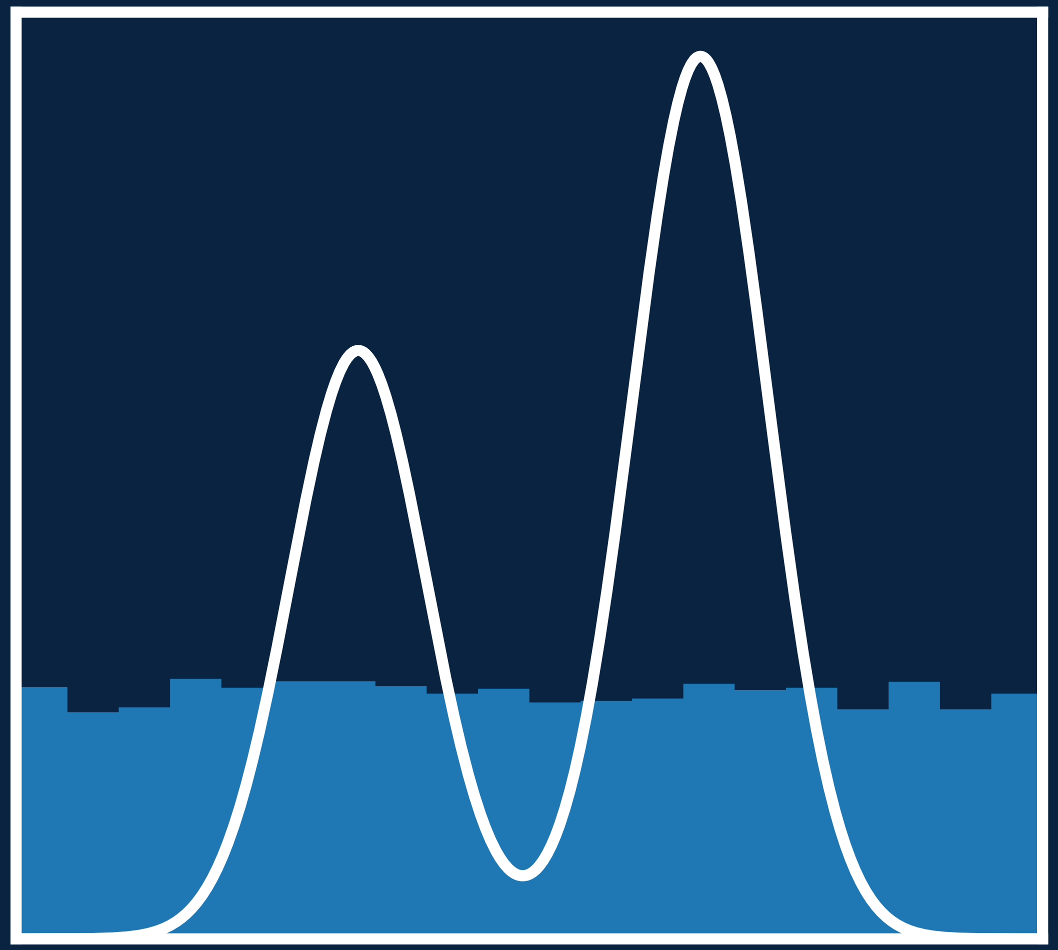


Flat sampling:
inefficient

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

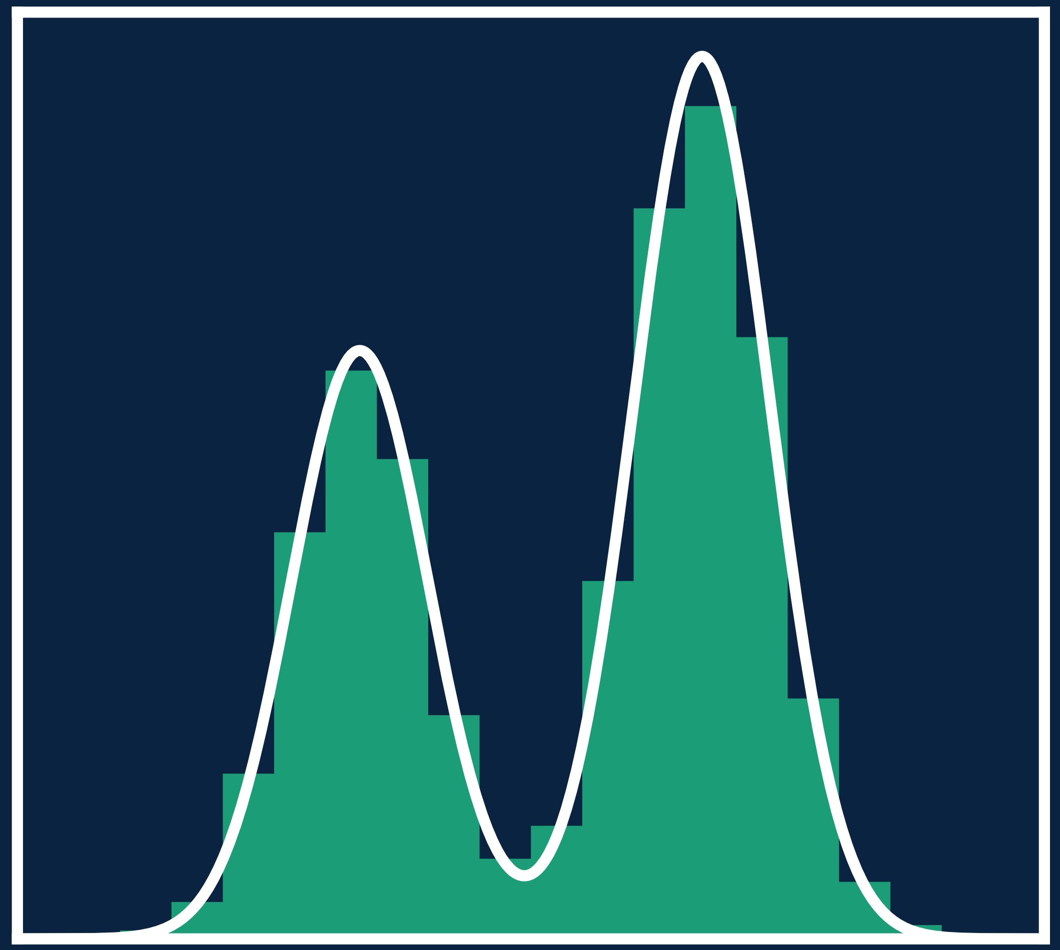
Monte Carlo integration

$$I = \int \mathrm{d}x\, f(x)$$



Flat sampling:
inefficient

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



Importance sampling:
find g close to f

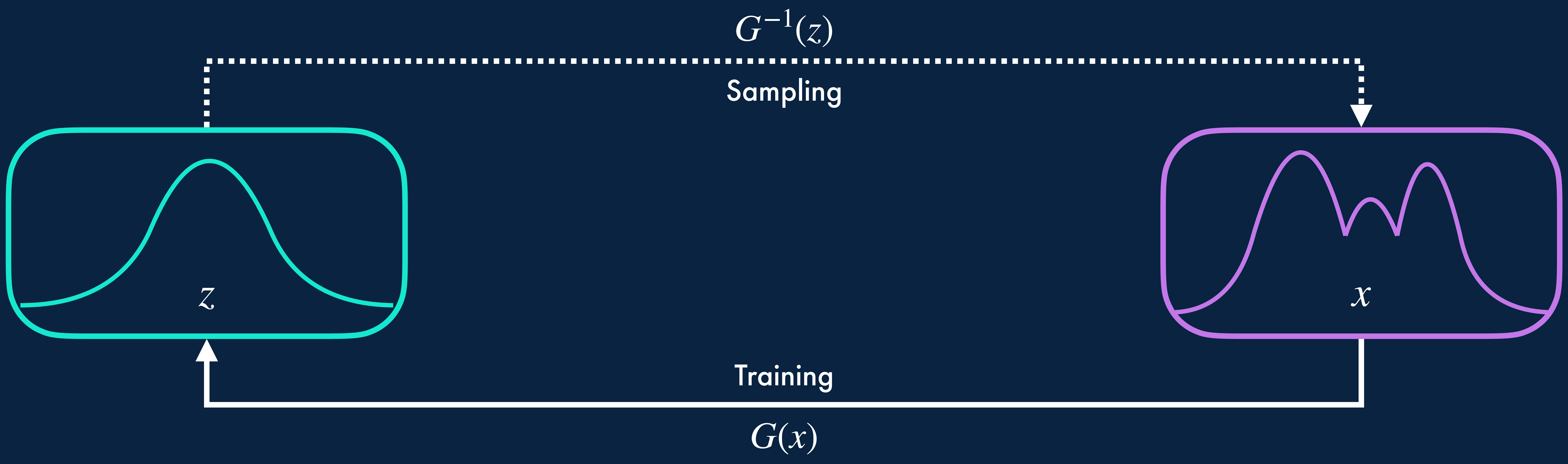
$$I = \left\langle \frac{f(x)}{g(x)} \right\rangle_{x \sim g(x)}$$

Normalizing Flow

Chain of **invertible**, **learnable** transformations with **exact likelihood** from change of variables formula

$$\log p_\theta(x) = \log p_Z(G_\theta(x)) + \log \left| \frac{\partial G_\theta(x)}{\partial x} \right|$$

[2001.05478, 2001.05486, 2001.10028,2005.12719, 2112.09145]



Are there problems with flows?

Topological obstruction

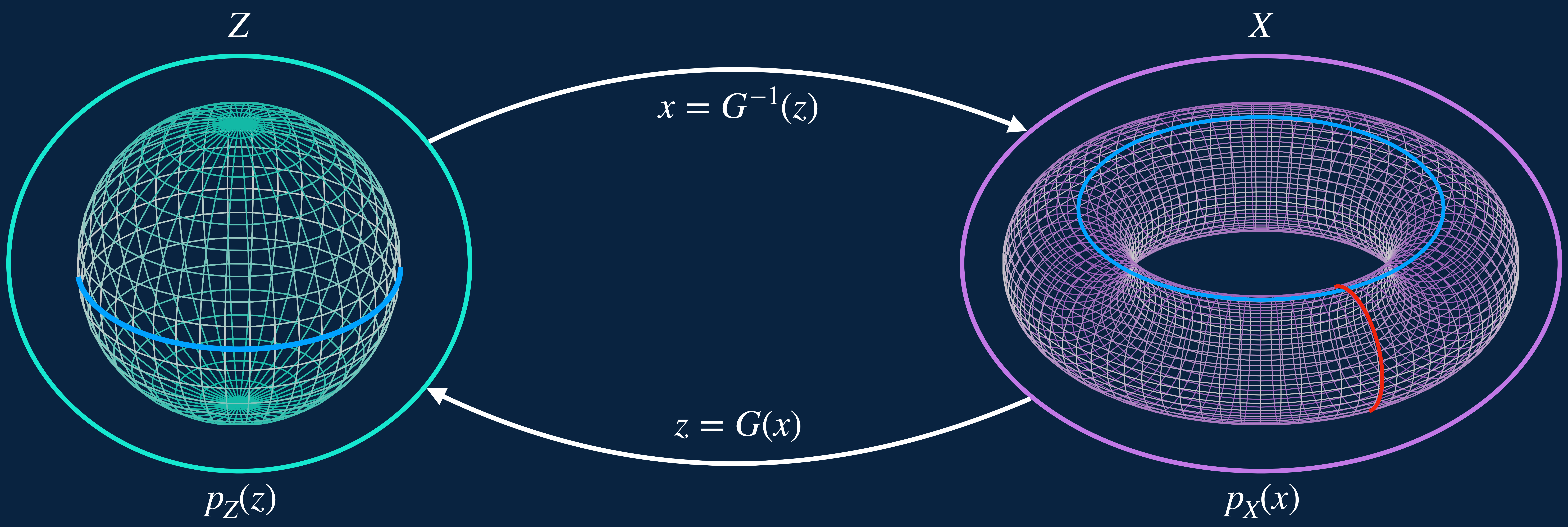
Lemma

Any **bijective mapping** $G(z)$ is a homeomorphism and preserves the topological structure of the input space. (Younes (2010), Dupon et al. [1904.01681])

Topological obstruction

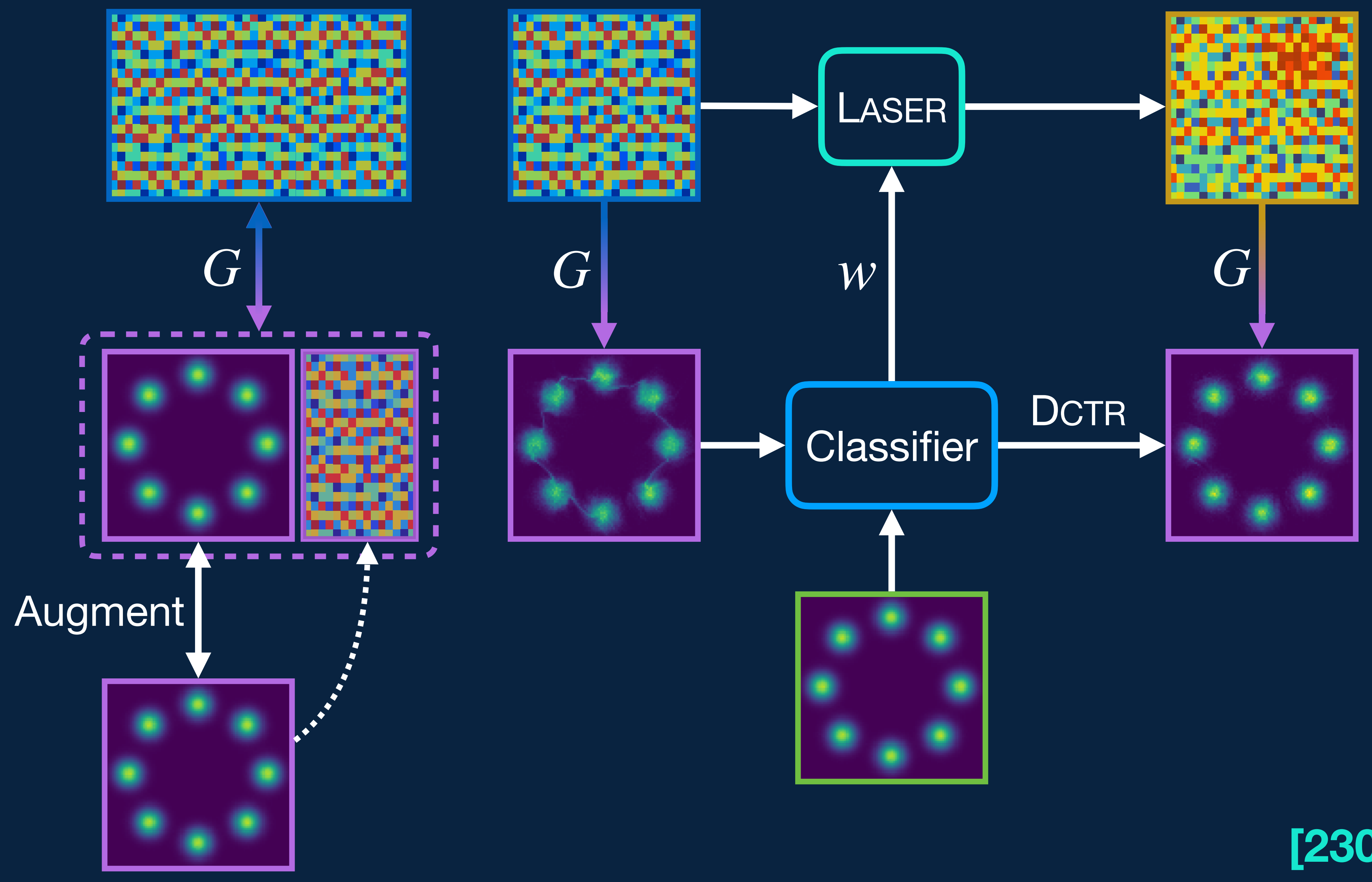
Lemma

Any **bijective mapping** $G(z)$ is a homeomorphism and preserves the topological structure of the input space. (Younes (2010), Dupon et al. [1904.01681])



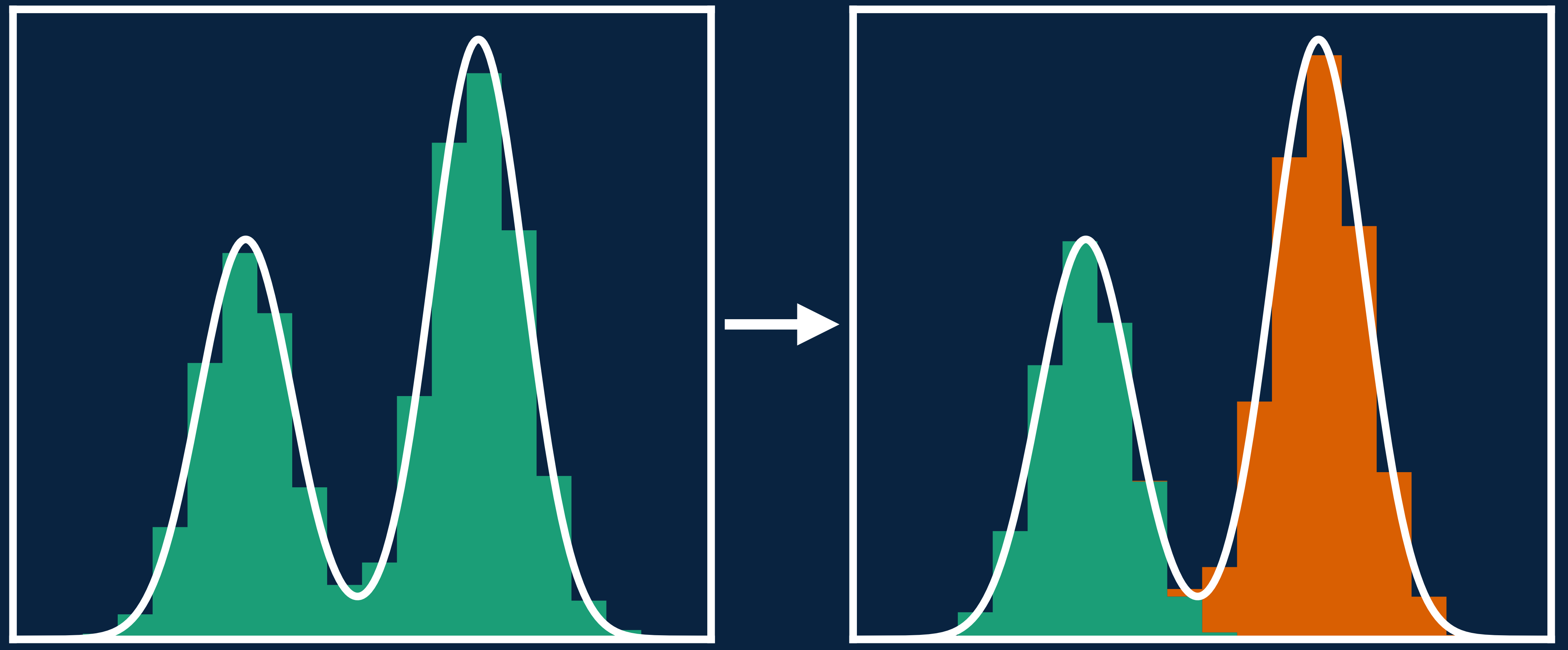
Avoiding the bottleneck

ELSA



[2305.07696]

MADNIS



Neural Multi-channel:
one map for each mode

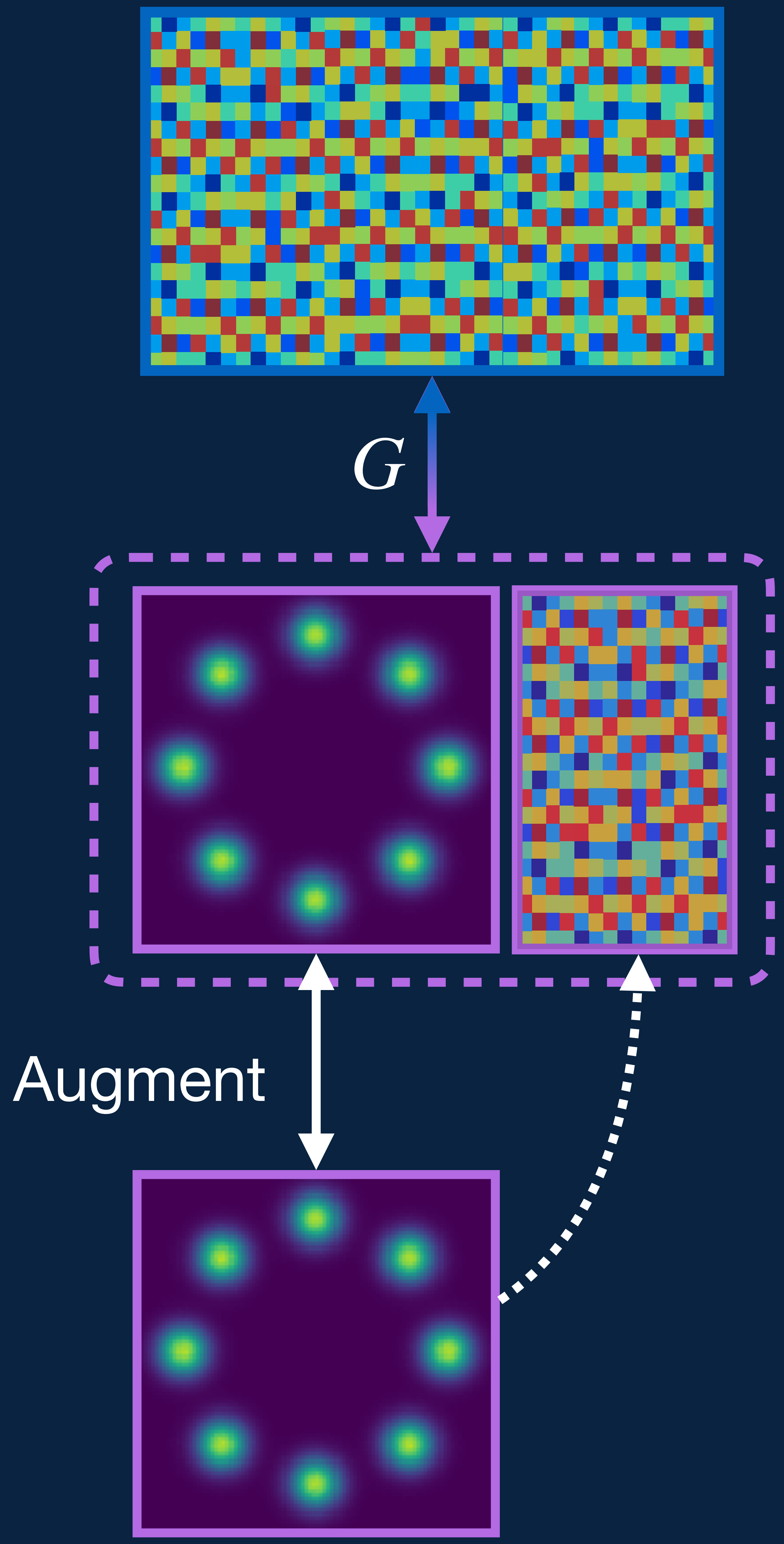
$$I = \sum_i \left\langle \alpha_i(x,\phi) \frac{f(x)}{g_i(x,\theta)} \right\rangle_{x \sim g_i(x,\theta)}$$

[2212.06172]

ELSA

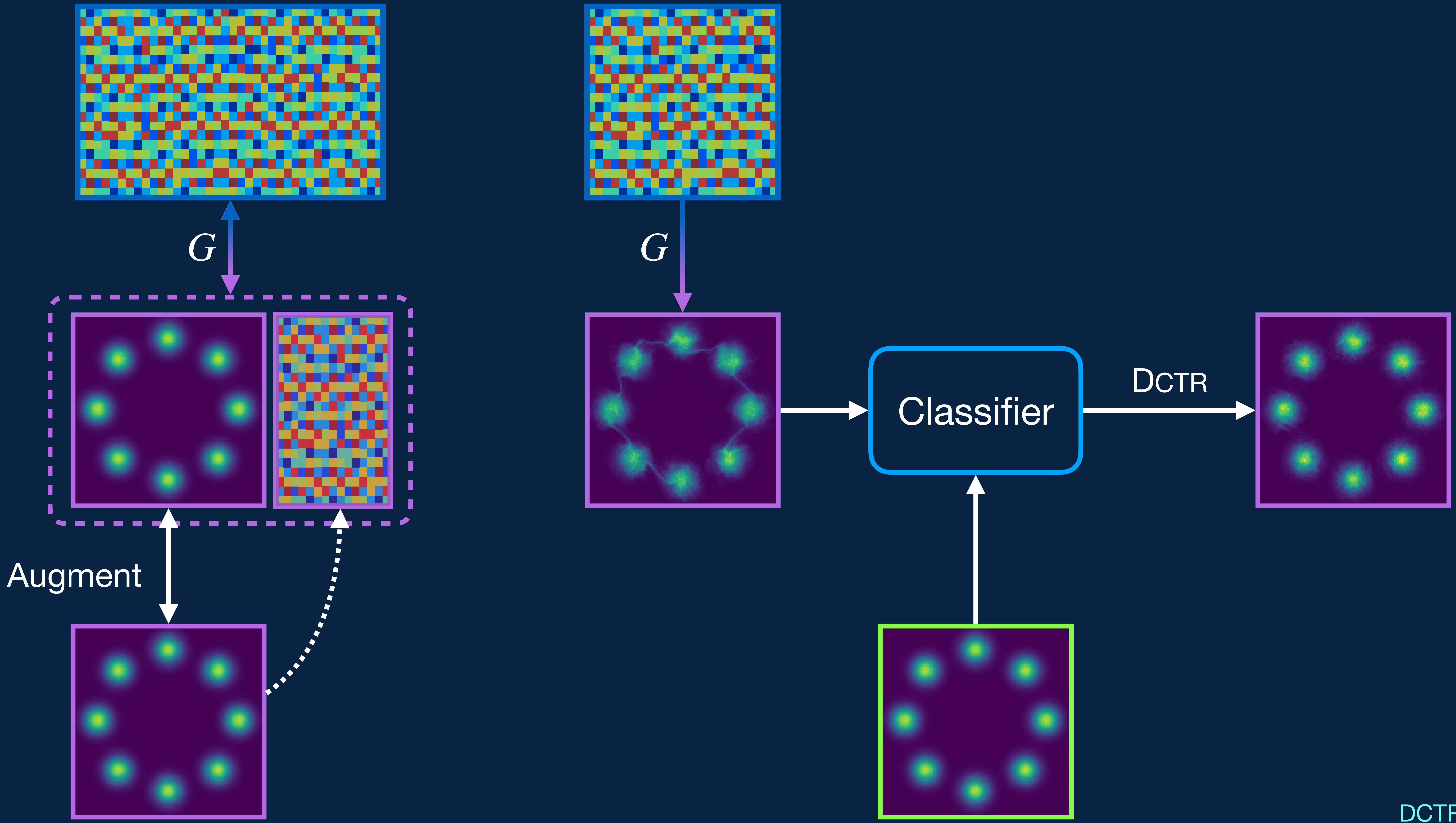
Enhanced Latent Spaces

ELSA — Basic functionalities



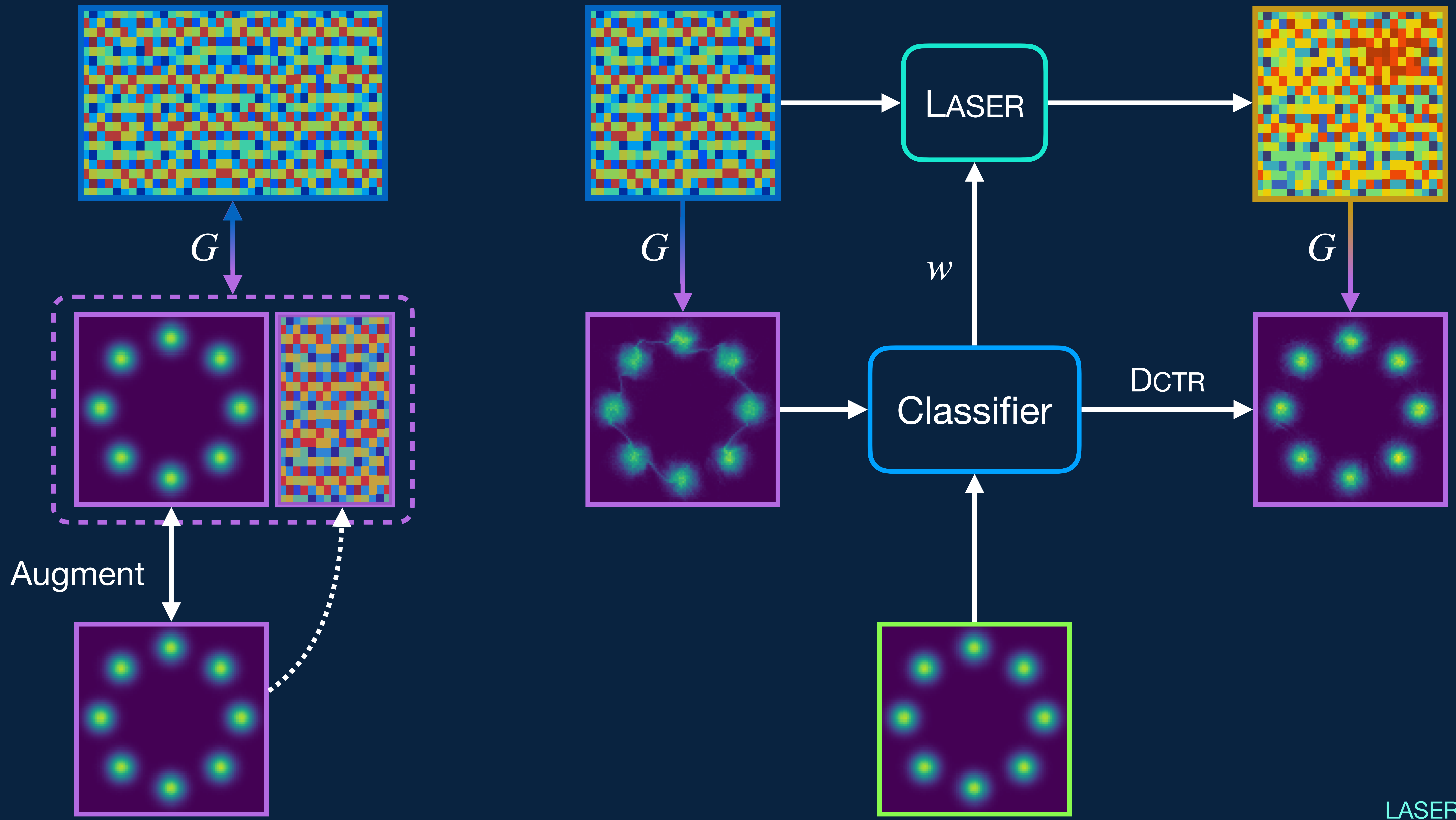
SurVAE [2007.02731], AugFlows [2002.07101, 2002.09741]

ELSA — Basic functionalities



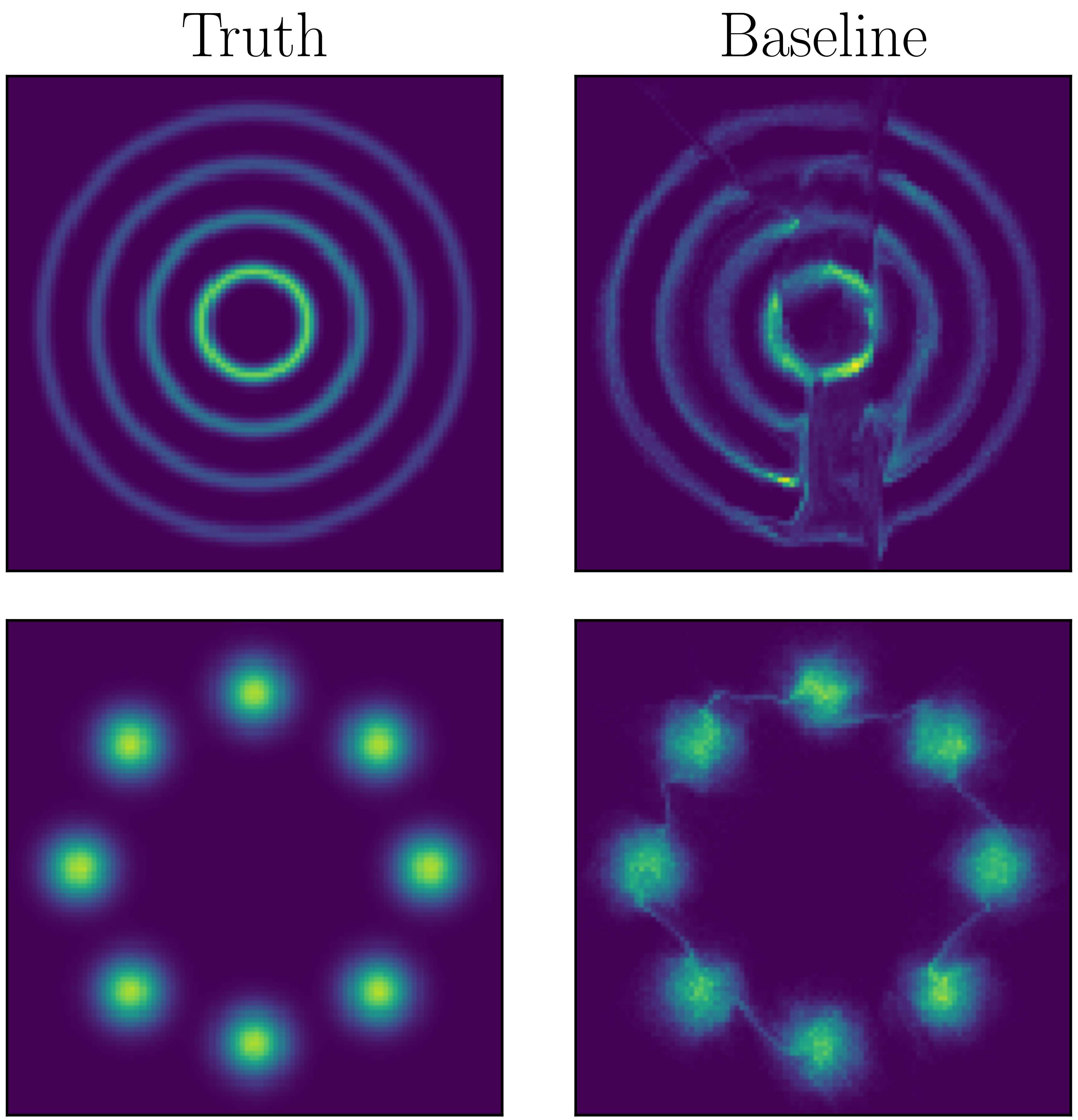
DCTR [1907.08209]

ELSA — Basic functionalities

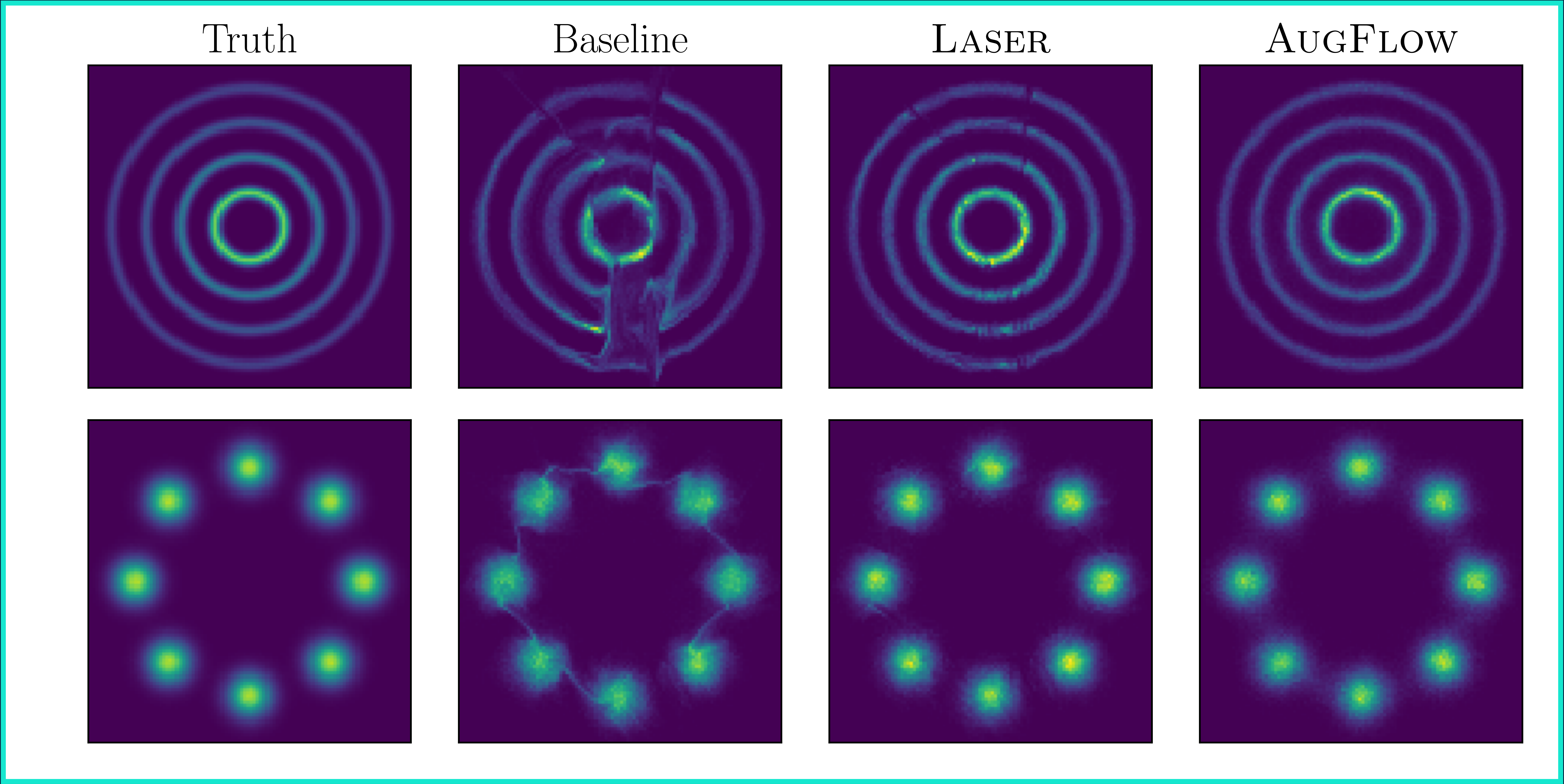


LASER [2106.00792]

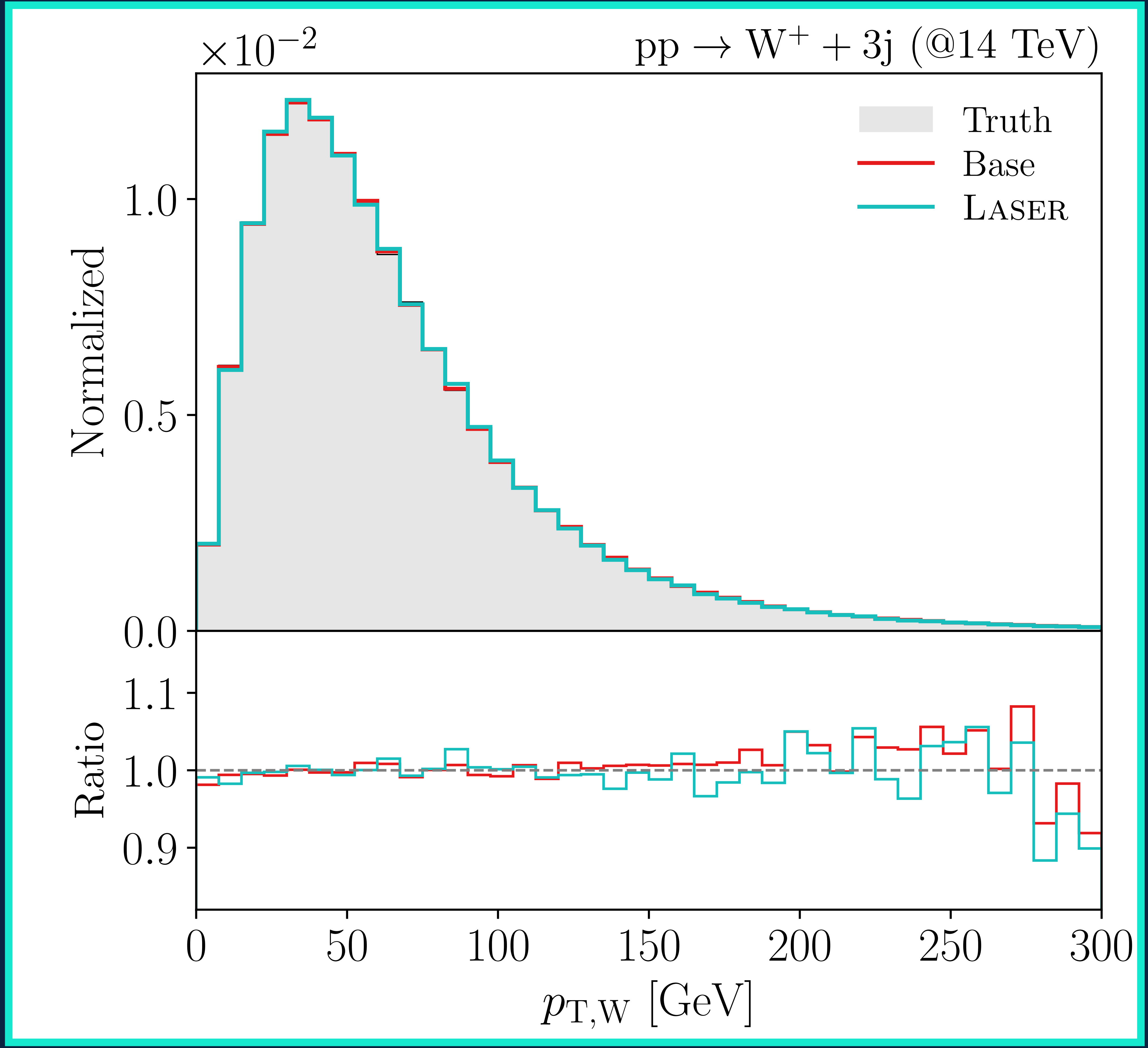
Toy examples



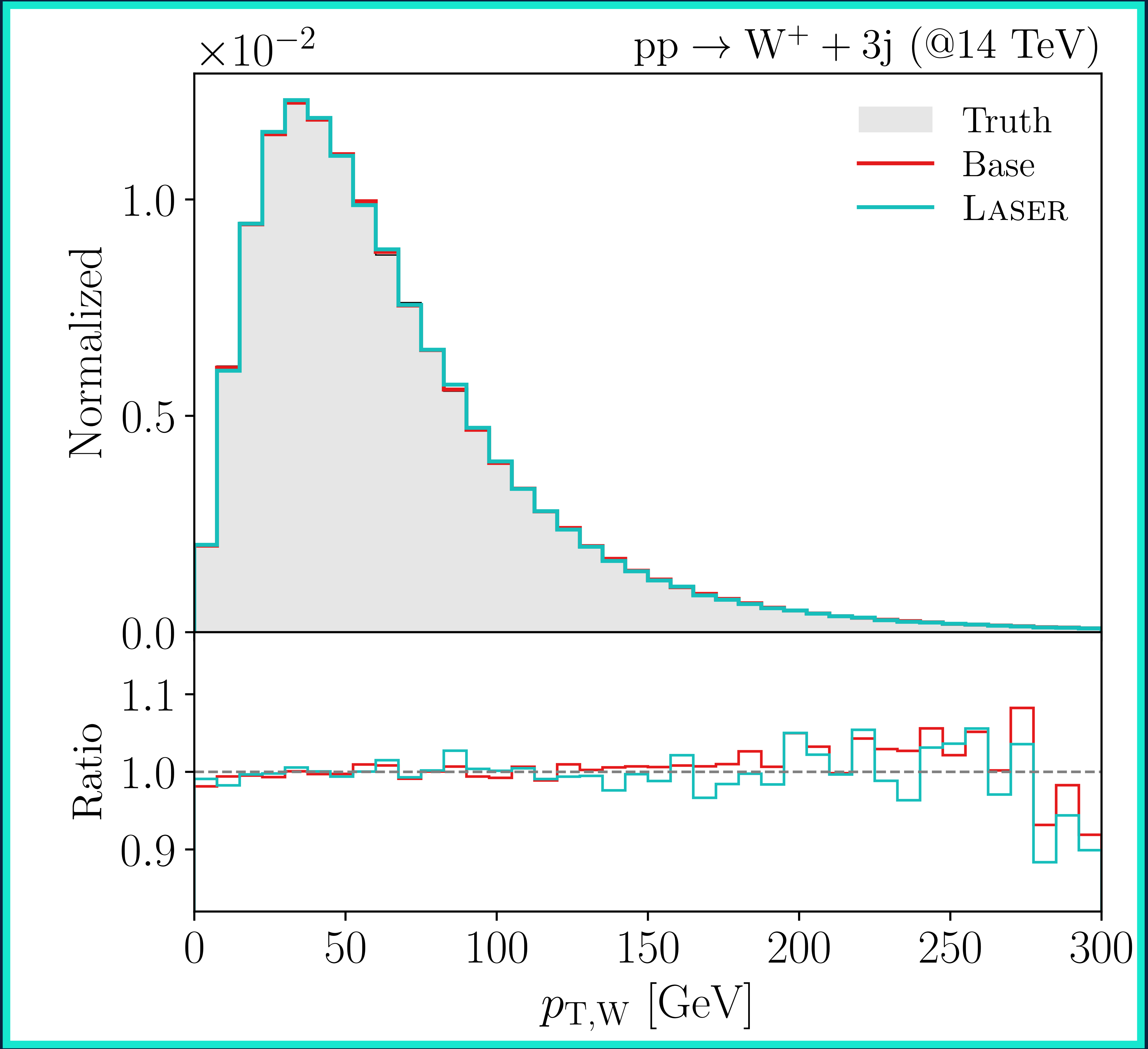
Toy examples

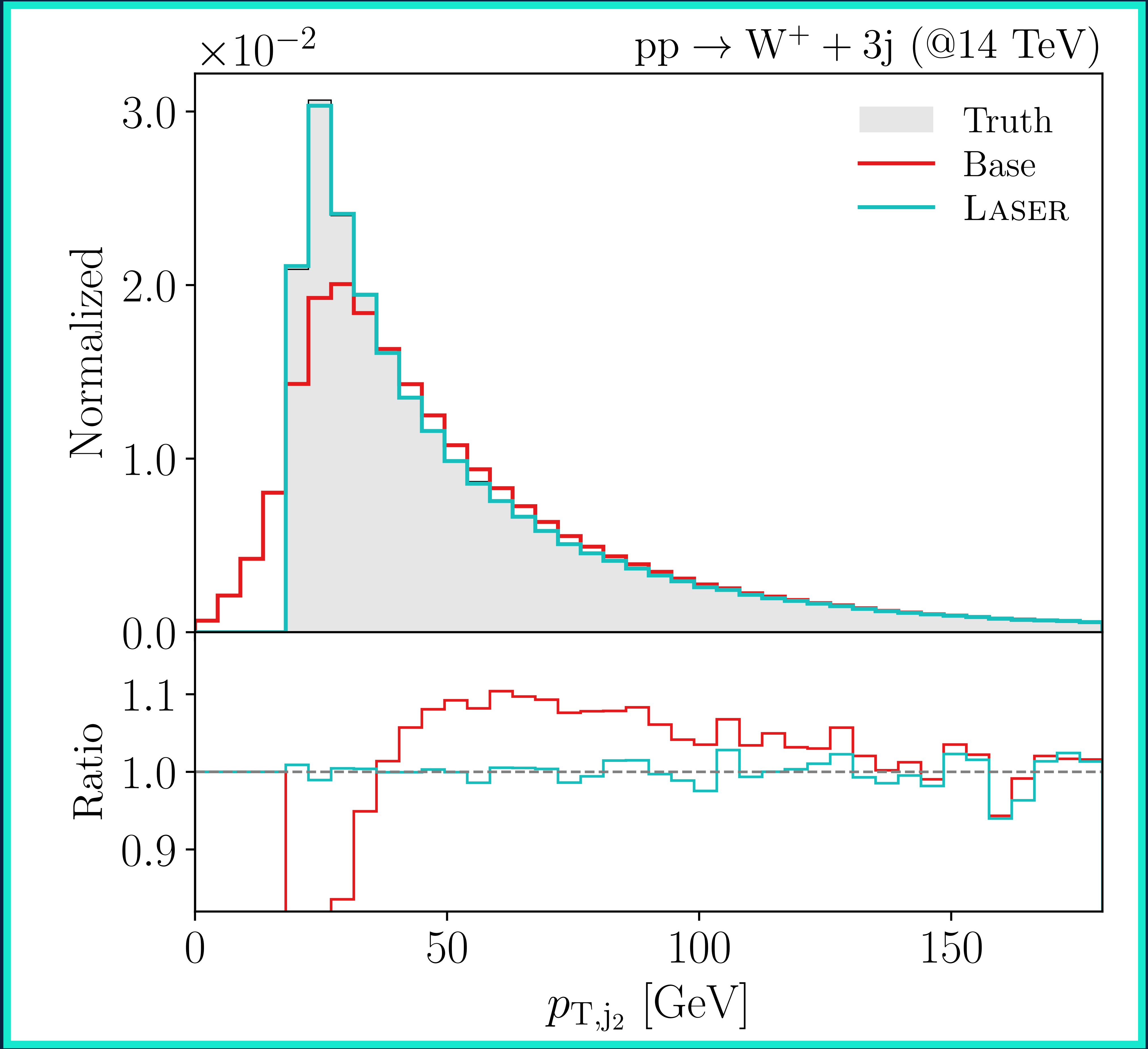


LHC example — W + 3 jets

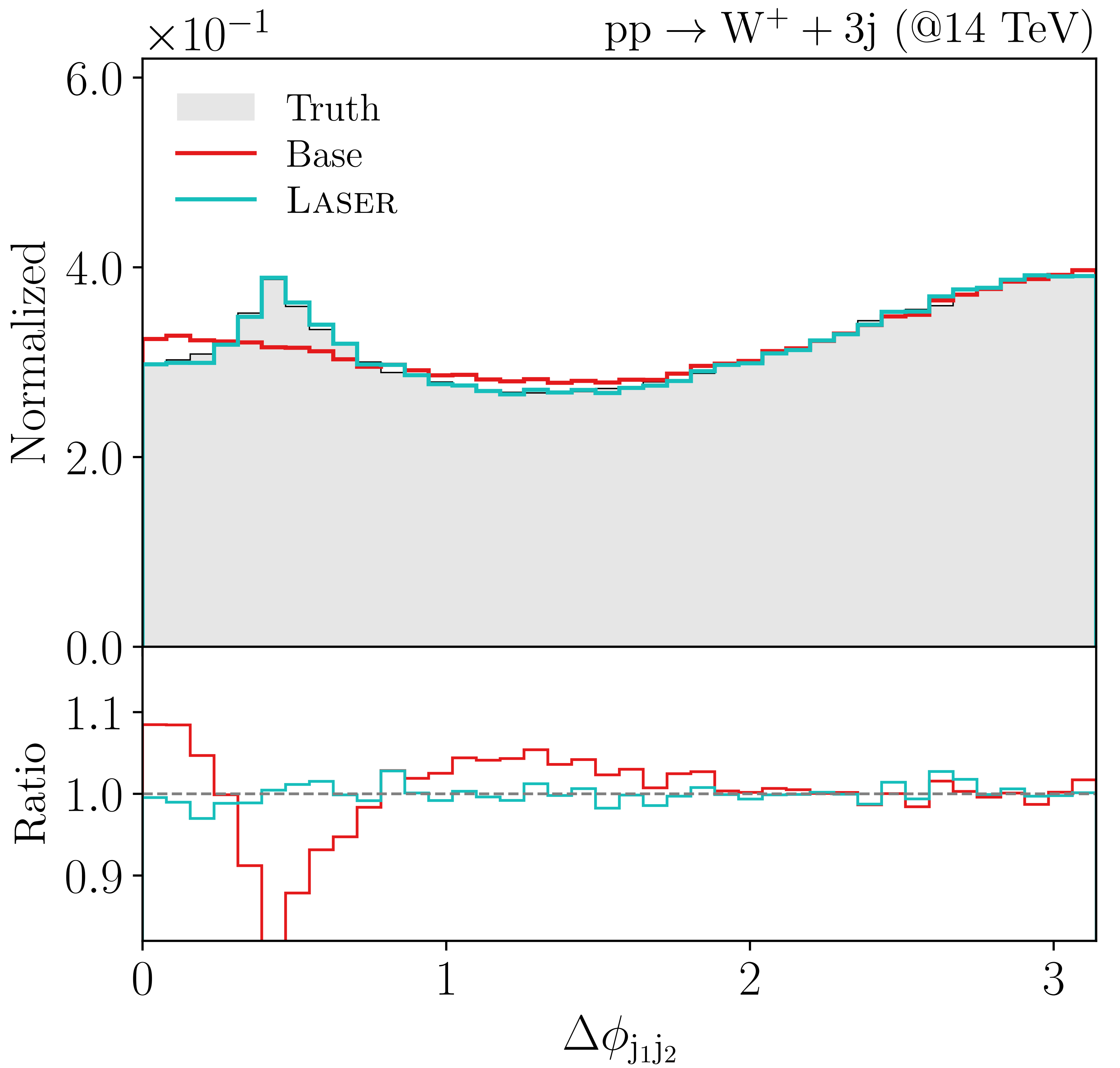


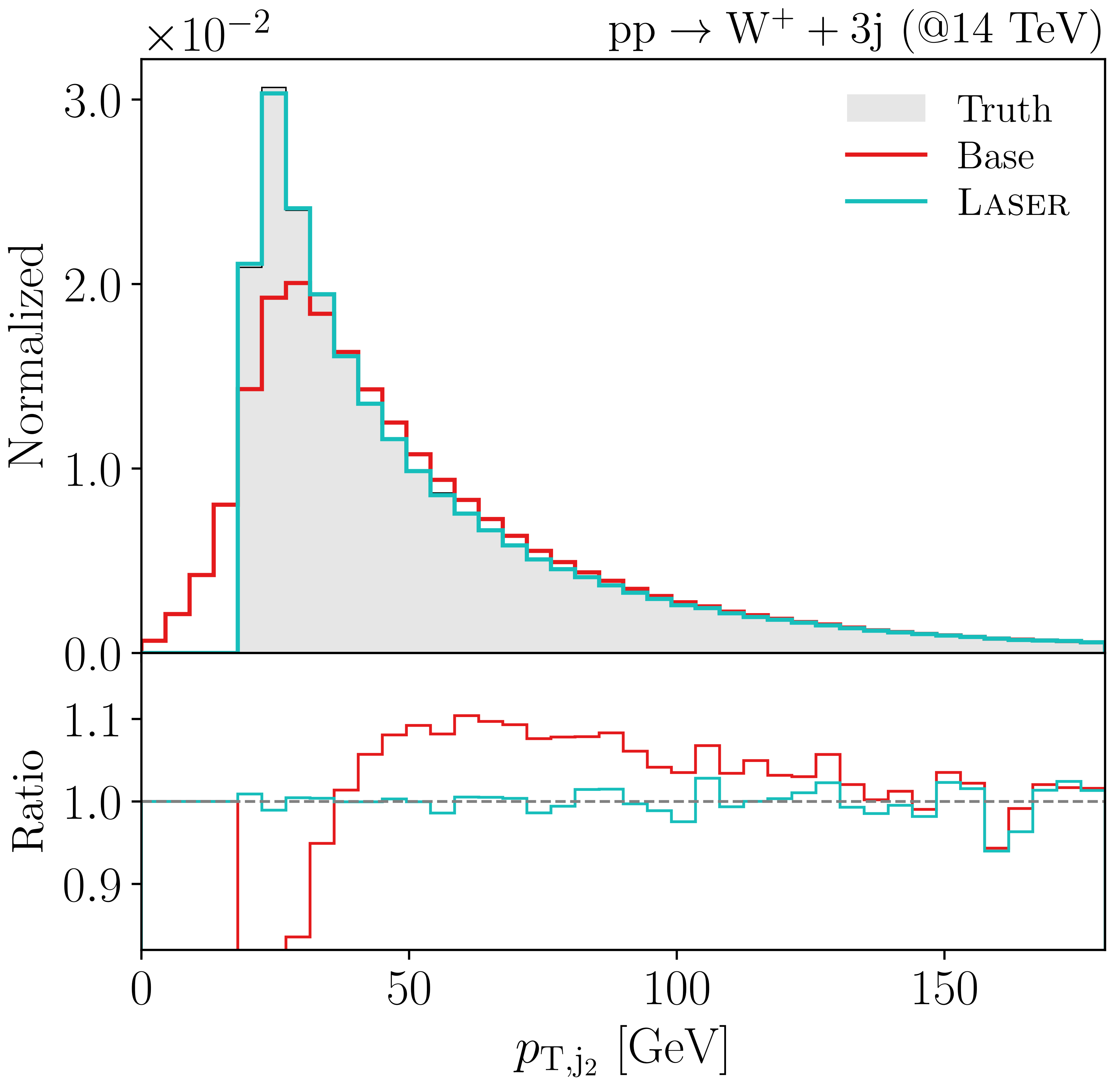
LHC example — W + 3 jets





LHC example — W + 3 jets





MadNIS

Neural Importance Sampling

MadNIS — Basic functionality

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

MadNIS — Basic functionality

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

Use physics knowledge to construct channel and mappings

MadNIS — Basic functionality

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

Use physics knowledge to construct channel and mappings

Normalizing flow to refine channel mappings

Fully connected network to refine channel weights

MadNIS — Basic functionality

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

$\downarrow$

Use physics knowledge to construct channel and mappings

$\downarrow$ $\downarrow$

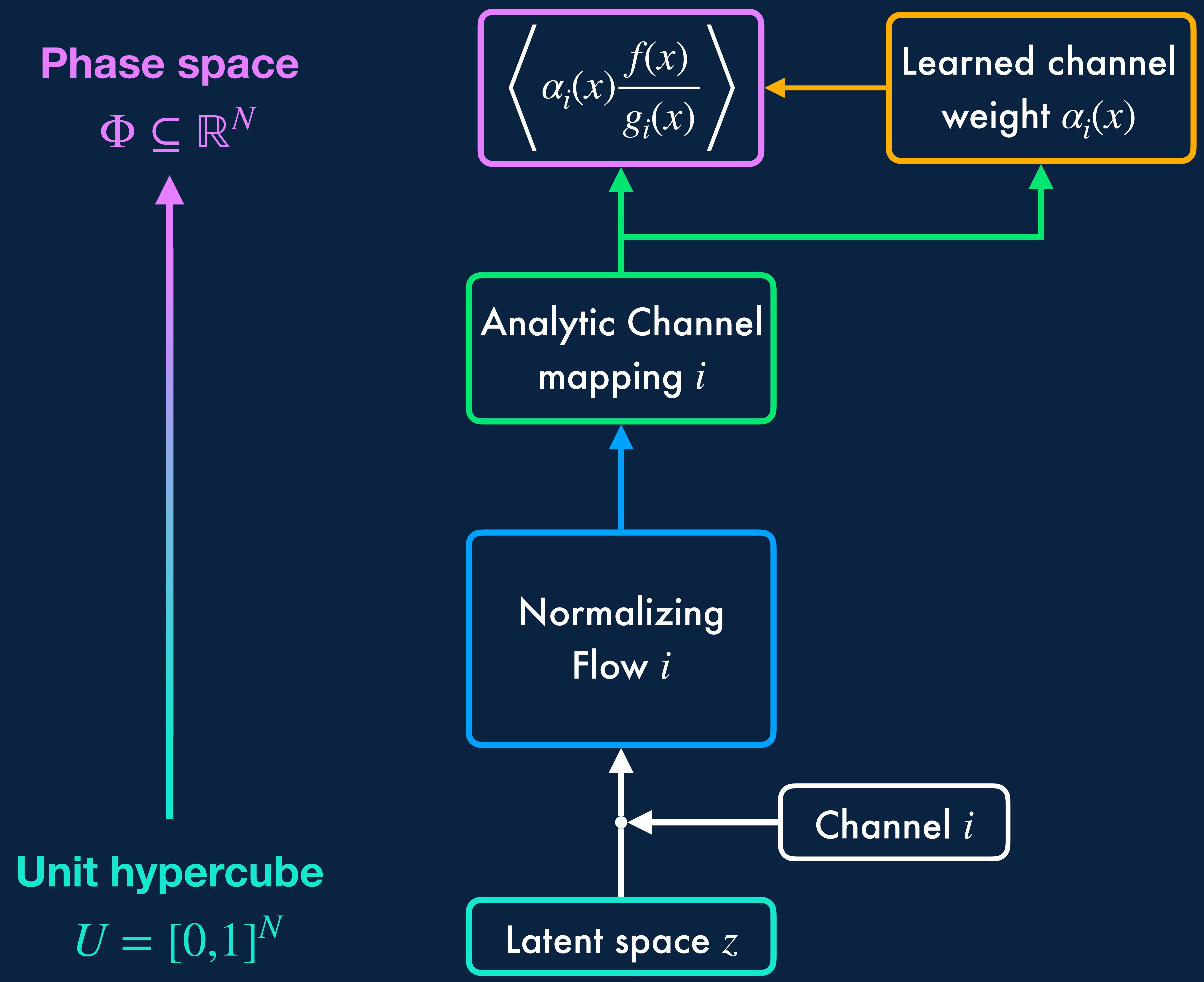
Normalizing flow to refine channel mappings

Fully connected network to refine channel weights

$\downarrow$ $\downarrow$

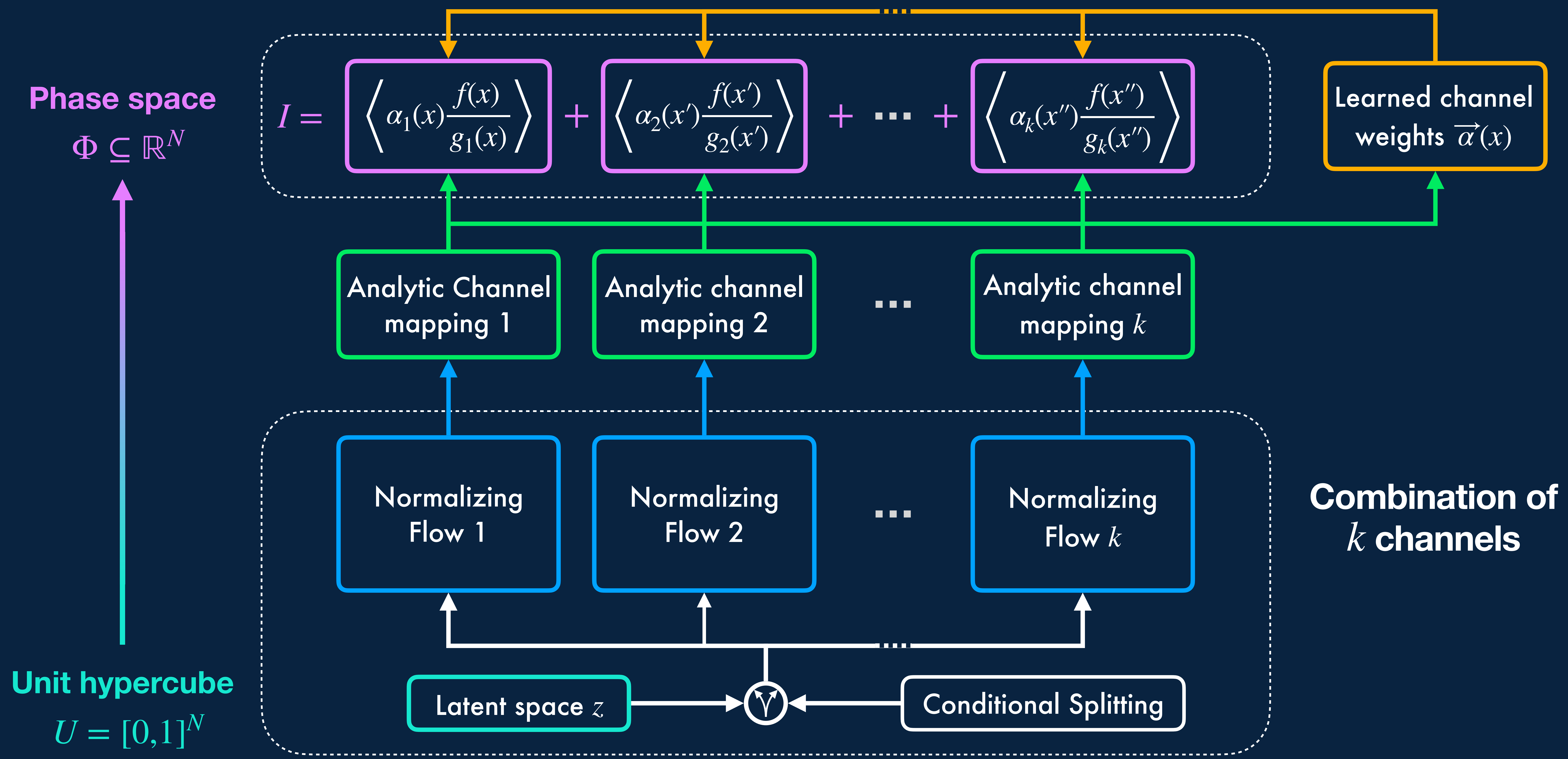
Update simultanously with variance as loss function

MadNIS — Basic functionality



Single channel i

MadNIS — Basic functionality



MadNIS — Overview

MADNIS

Basic functionality

- Neural Channel Weights
- Normalizing Flow
- MadGraph matrix elements
- MadEvent channel mappings

Improved multi-channeling

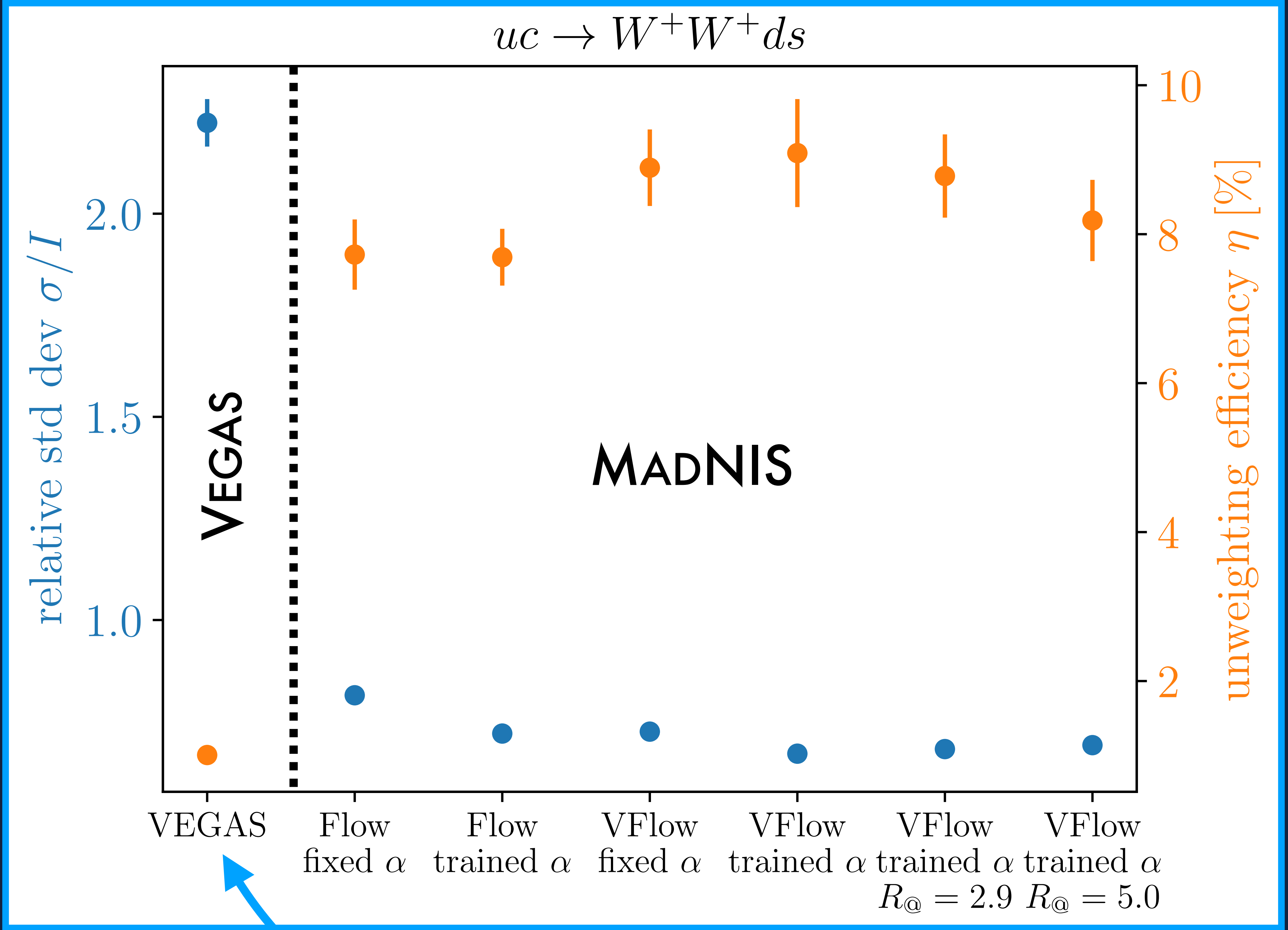
- Conditional flows
- Overflow Channels
- Symmetries between channels
- Stratified Sampling/ Training

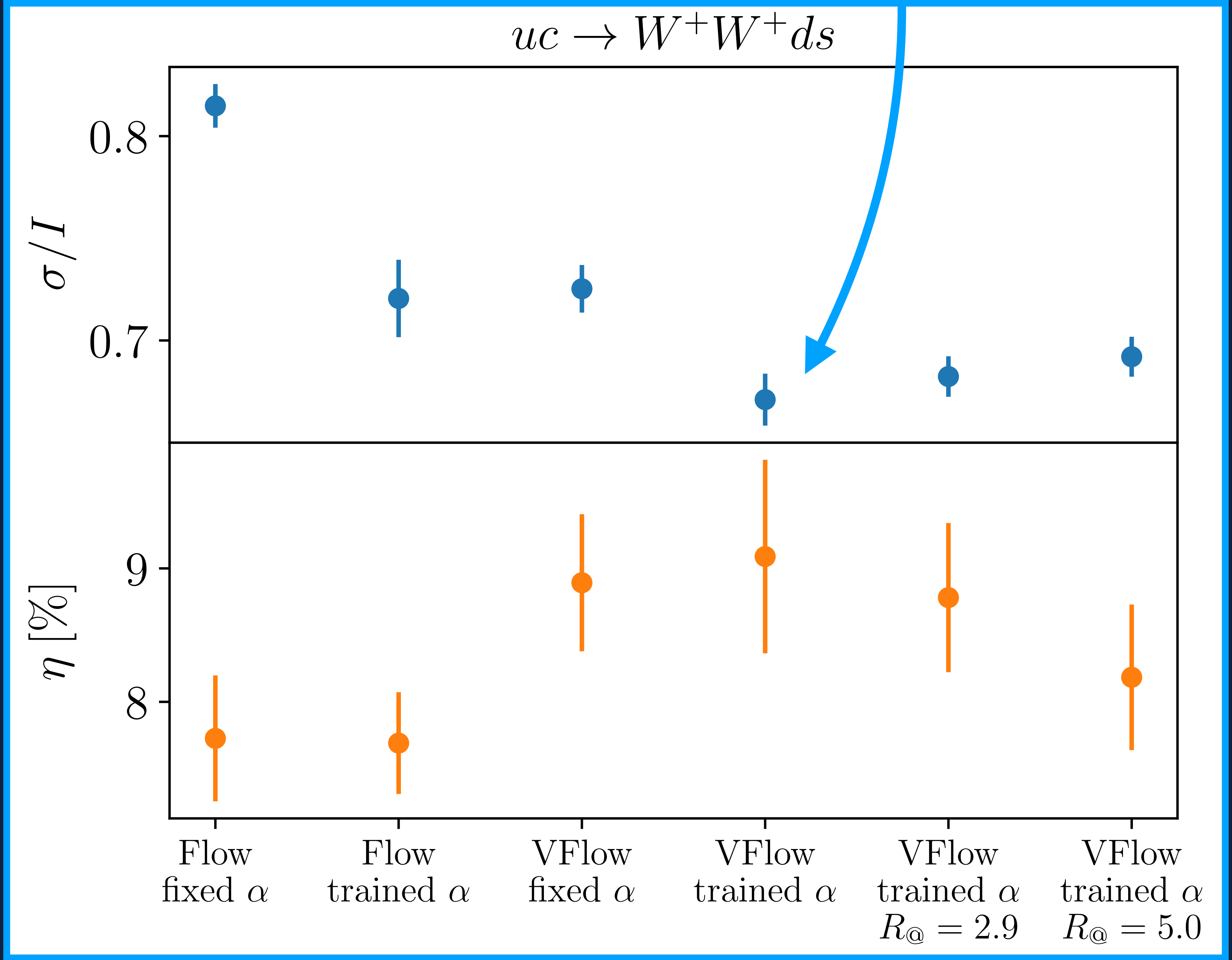
Improved training

- VEGAS Initialization
- Buffered Training
- Trainable Rotations

LHC example I — VBS

Significant improvement
from trained channel weights



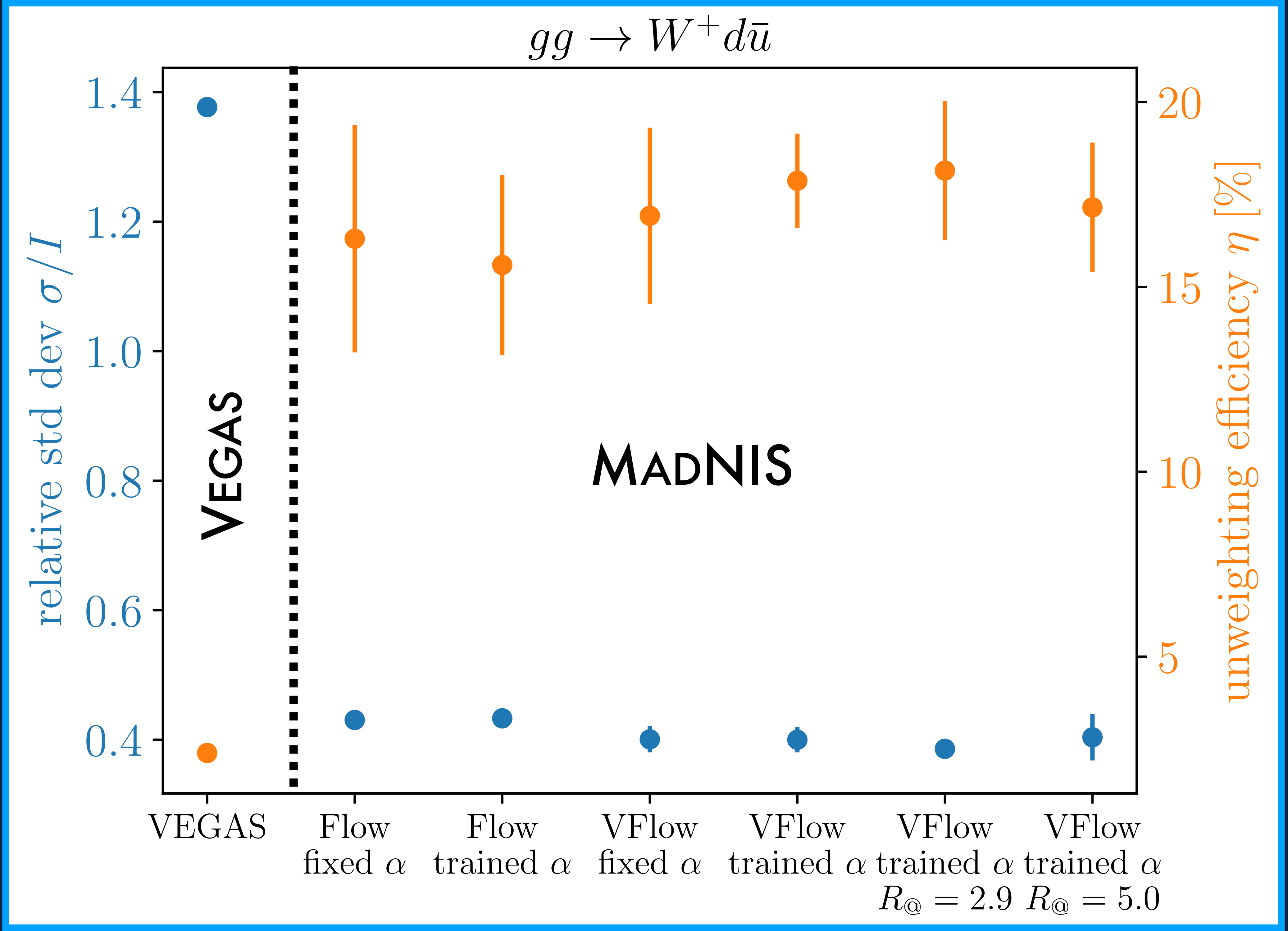


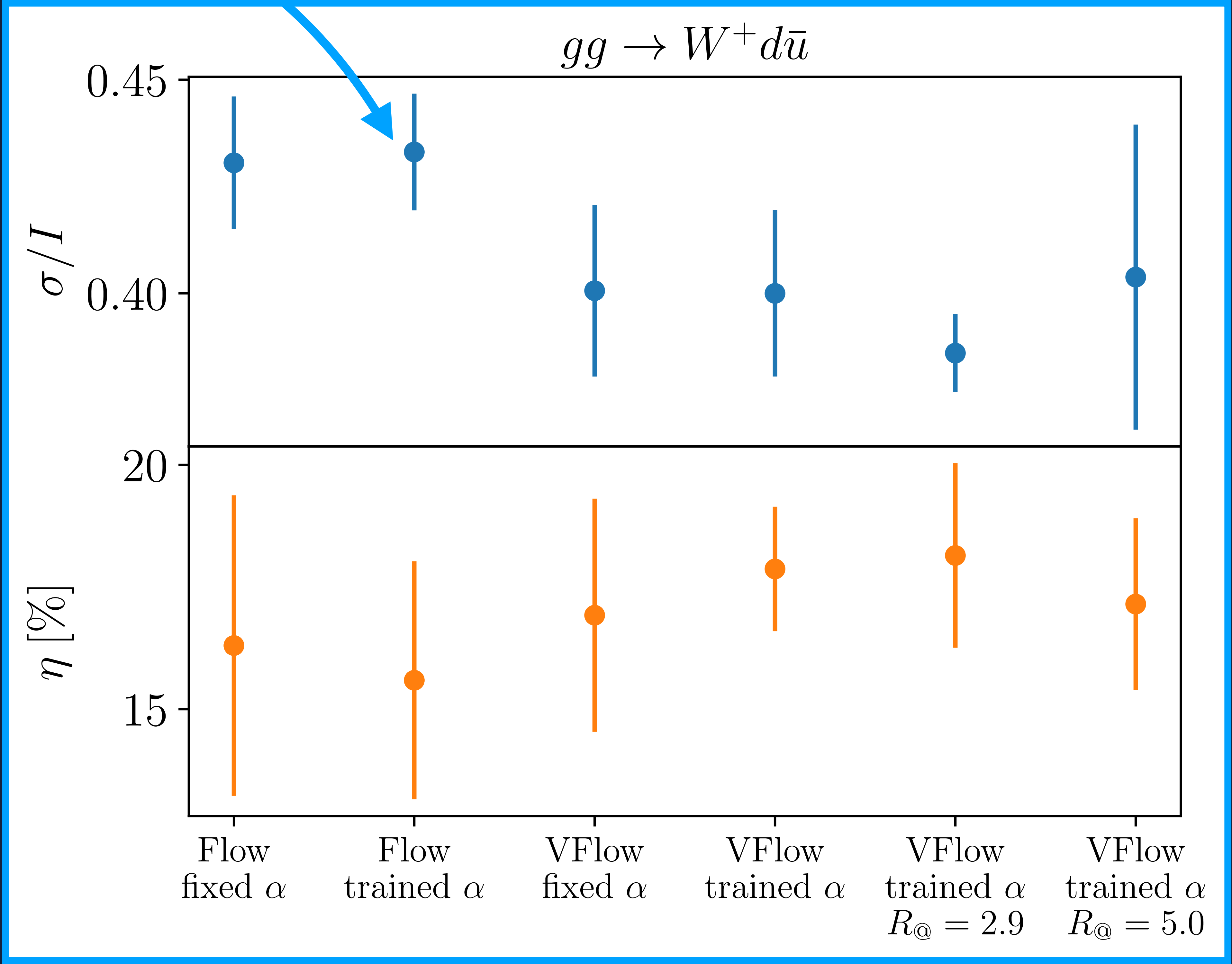
(preliminary)

Unweighting efficiency improved
up to factor ~10 compared to VEGAS

LHC example II — W + 2 jets

Process has small interference terms
→ no significant improvement from trained channel weights





(preliminary)

Otherwise similar to results for VBS

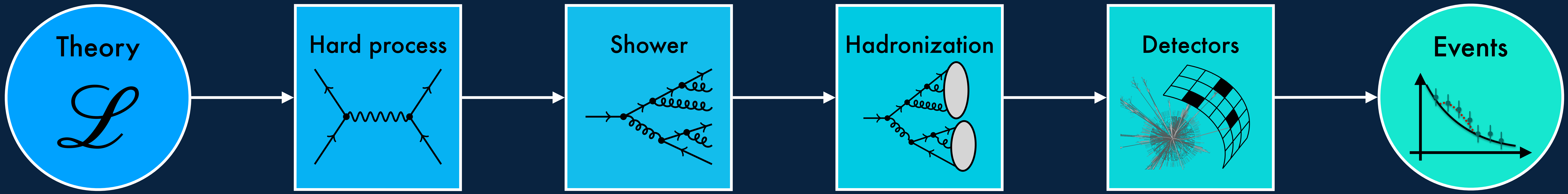
Summary and Outlook

Take-home message

- **Fast** and **precise** predictions with ML-based simulations
- Normalizing flows provide statistically **well-defined likelihoods** for inference
- Account for **multi-modal distributions** with **modfied latent spaces** or **multi-channel flow**

Future exercises

- **Full integration** of **MadNIS** into standard tools → **MadGraph,….**
- Make everything run on the **GPU and differentiable** (MadJax - Heinrich et al. [2203.00057])



Summary and Outlook

SciPost — SciPost Phys. 14, 079 (2023)

Machine learning and LHC event generation

Anja Butter[1,2], Tilman Plehn[1], Steffen Schumann[3], Simon Badger[4], Sascha Caron[5,6] Kyle Cranmer[7,8], Francesco Armando Di Bello[9], Etienne Dreyer[10], Stefano Forte[11], Sanmay Ganguly[12], Dorival Gonçalves[13], Eilam Gross[10], Theo Heimel[1], Gudrun Heinrich[14], Lukas Heinrich[15], Alexander Held[16], Stefan Höche[17], Jessica N. Howard[18], Philip Ilten[19], Joshua Isaacson[17], Timo Janßen[3], Stephen Jones[20], Marumi Kado[9,21], Michael Kagan[22], Gregor Kasieczka[23], Felix Kling[24], Sabine Kraml[25], Claudius Krause[26], Frank Krauss[20], Kevin Kröninger[27], Rahool Kumar Barman[13], Michel Luchmann[1], Vitaly Magerya[14], Daniel Maitre[20], Bogdan Malaescu[2], Fabio Maltoni[28,29], Till Martini[30], Olivier Mattelaer[28], Benjamin Nachman[31,32], Sebastian Pitz[1], Juan Rojo[6,33], Matthew Schwartz[34], David Shih[25], Frank Siegert[35], Roy Stegeman[11], Bob Stienen[5], Jesse Thaler[36], Rob Verheyen[37], Daniel Whiteson[18], Ramon Winterhalder[28], and Jure Zupan[19]

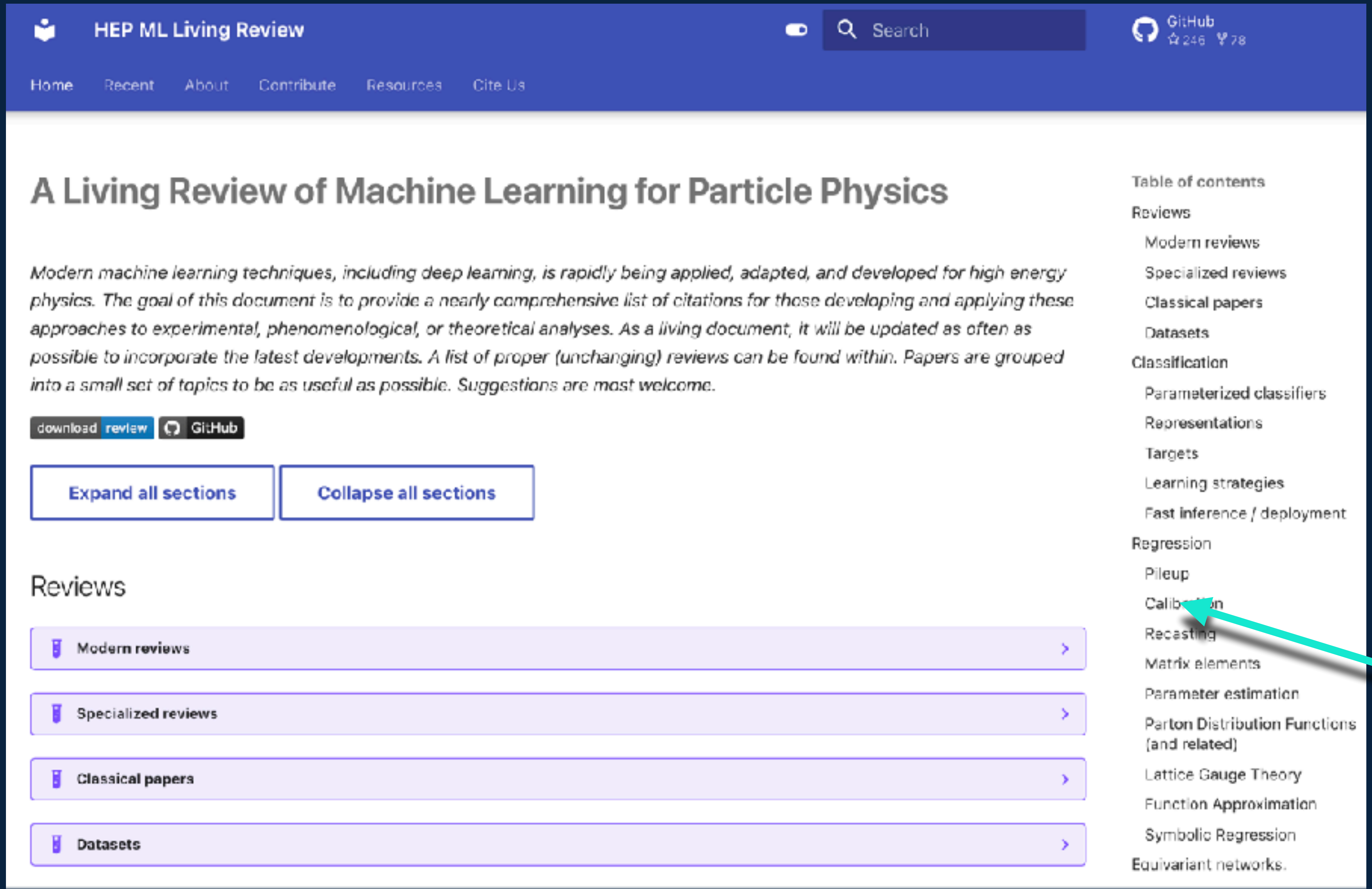
Abstract

First-principle simulations are at the heart of the high-energy physics research program. They link the vast data output of multi-purpose detectors with fundamental theory predictions and interpretation. This review illustrates a wide range of applications of modern machine learning to event generation and simulation-based inference, including conceptional developments driven by the specific requirements of particle physics. New ideas and tools developed at the interface of particle physics and machine learning will improve the speed and precision of forward simulations, handle the complexity of collision data, and enhance inference as an inverse simulation problem.

Future exercises

- **Full integration** of **MadNIS** into standard tools → **MadGraph,….**
- Make everything run on the **GPU and differentiable** (MadJax - Heinrich et al. [2203.00057])
- More details in our **Snowmass report**

Summary and Outlook



Future exercises

- **Full integration** of **MadNIS** into standard tools → **MadGraph,….**
- Make everything run on the **GPU and differentiable** (MadJax - Heinrich et al. [2203.00057])
- More details in our **Snowmass report**
- Stay tuned for many other **ML4HEP applications**