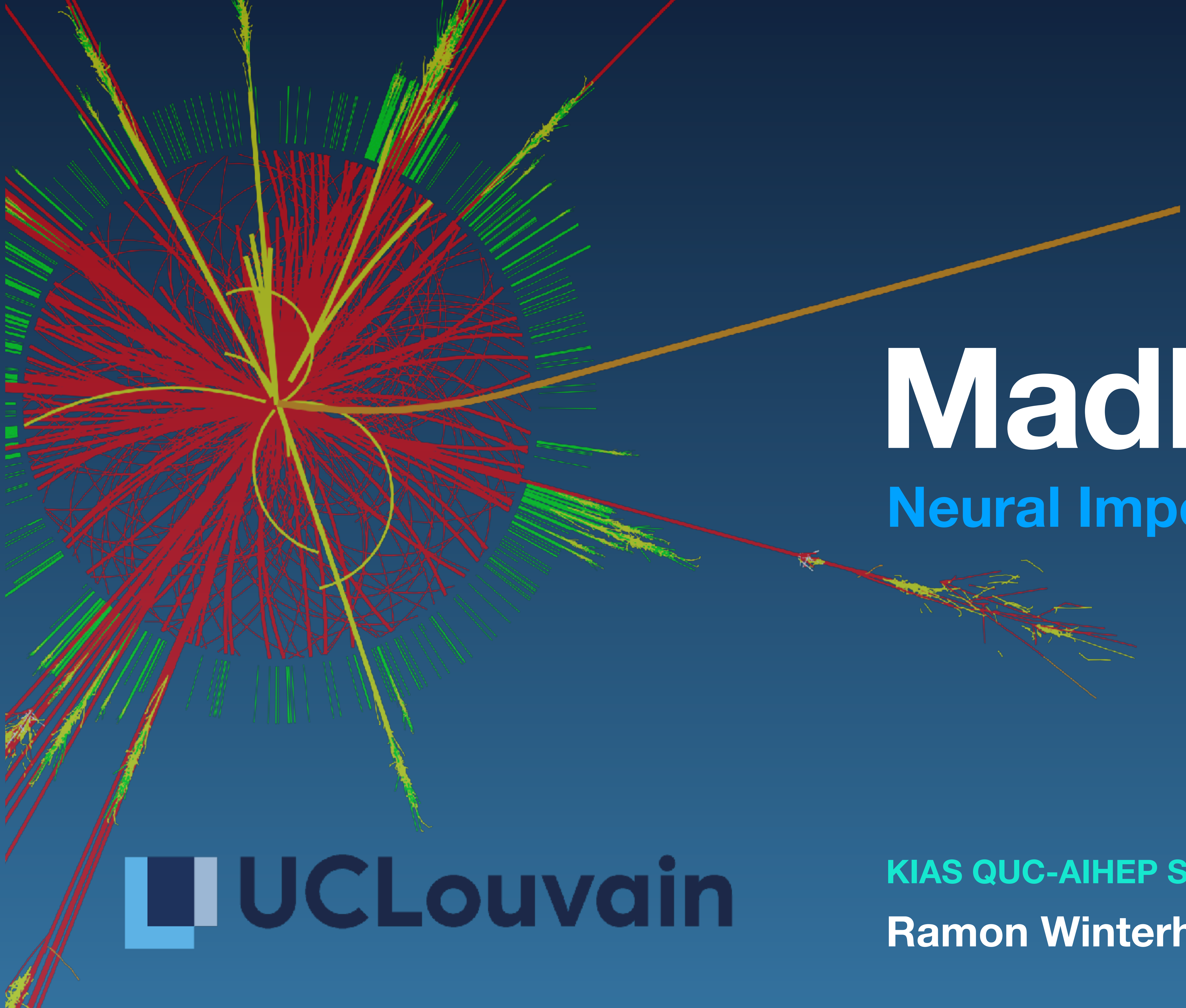


MadNIS

Neural Importance Sampling

KIAS QUC-AIHEP Seminar - Seoul 2023

Ramon Winterhalder — UC Louvain

Plan of attack

1. Machine learning for particle physics?

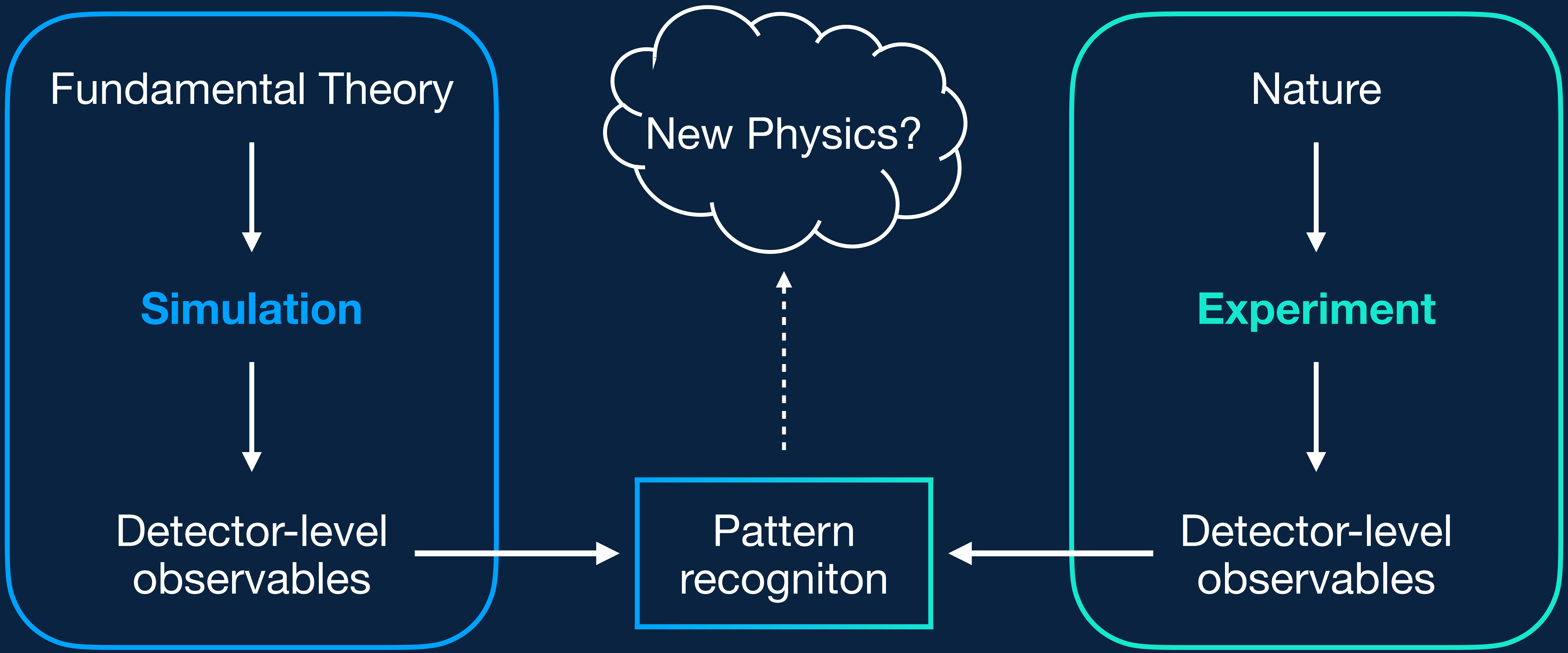
2. MadNIS — Basic functionality

3. MadNIS — Additional features

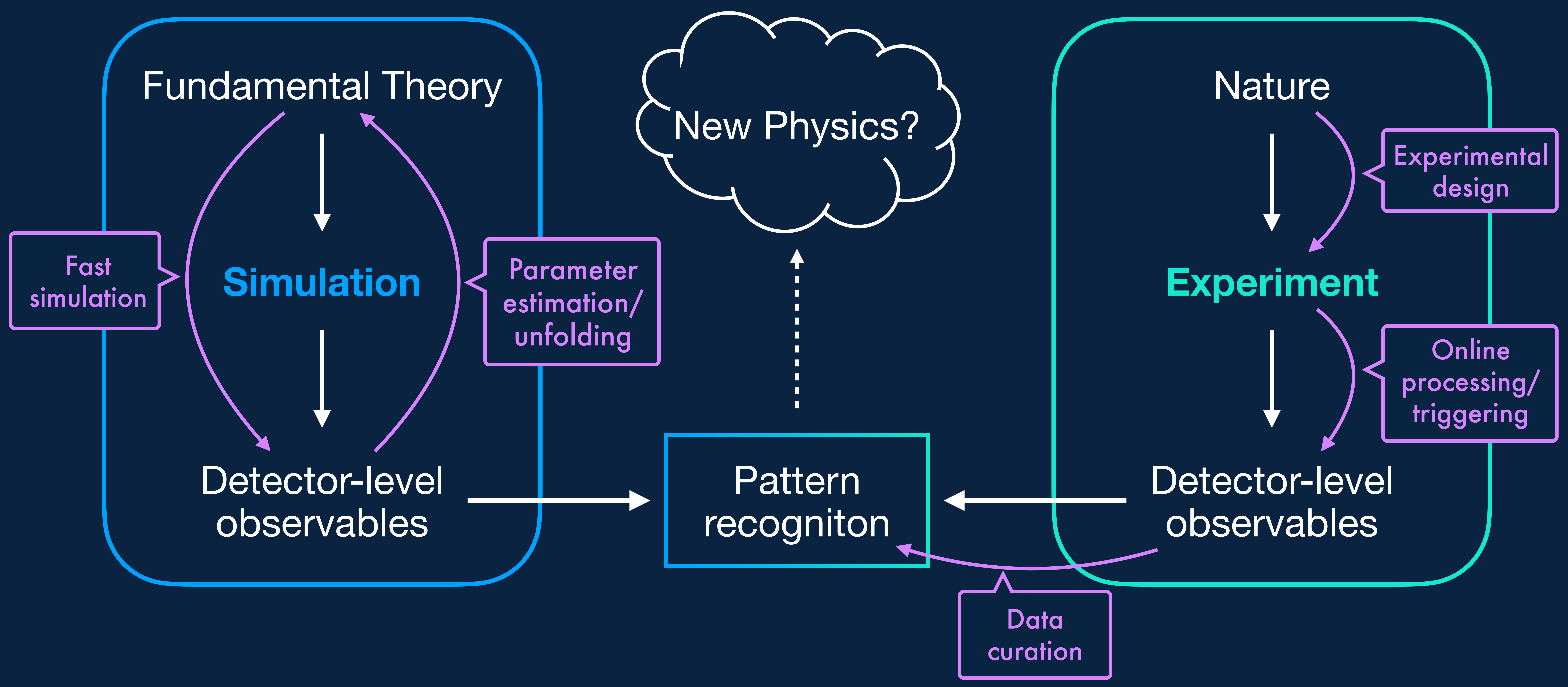
4. Summary and discussion

How can ML help in particle physics?

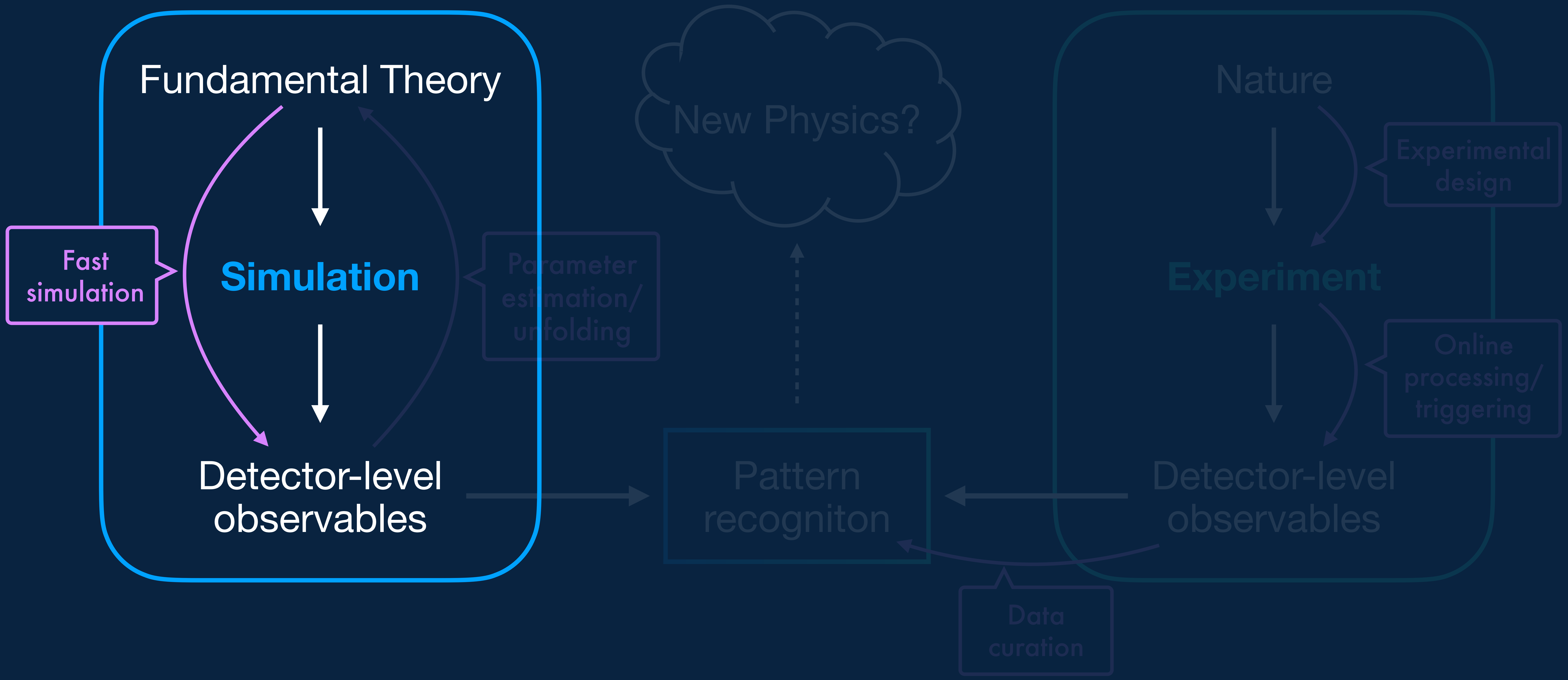
LHC analysis (oversimplified)



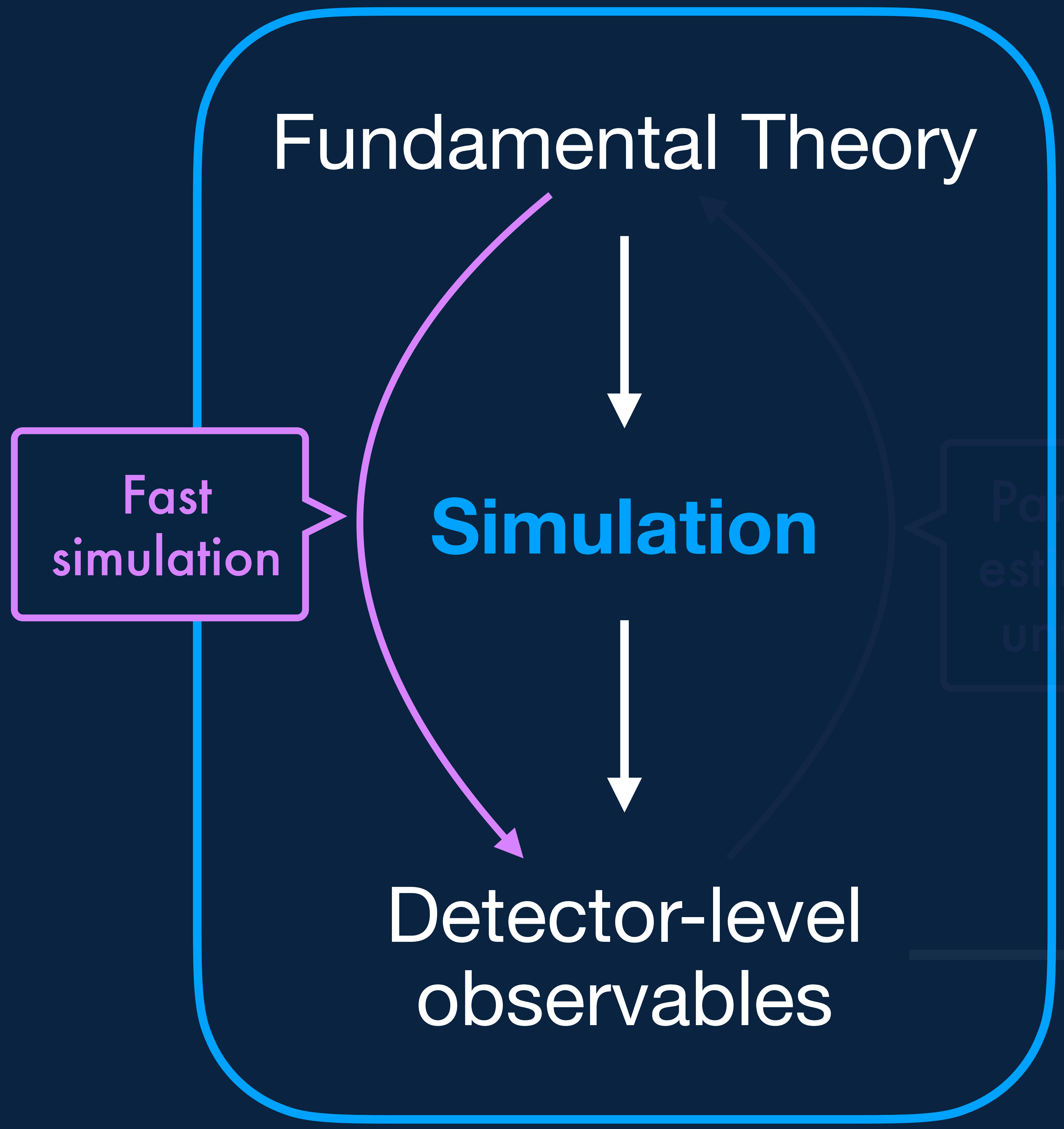
LHC analysis + ML



LHC analysis + ML

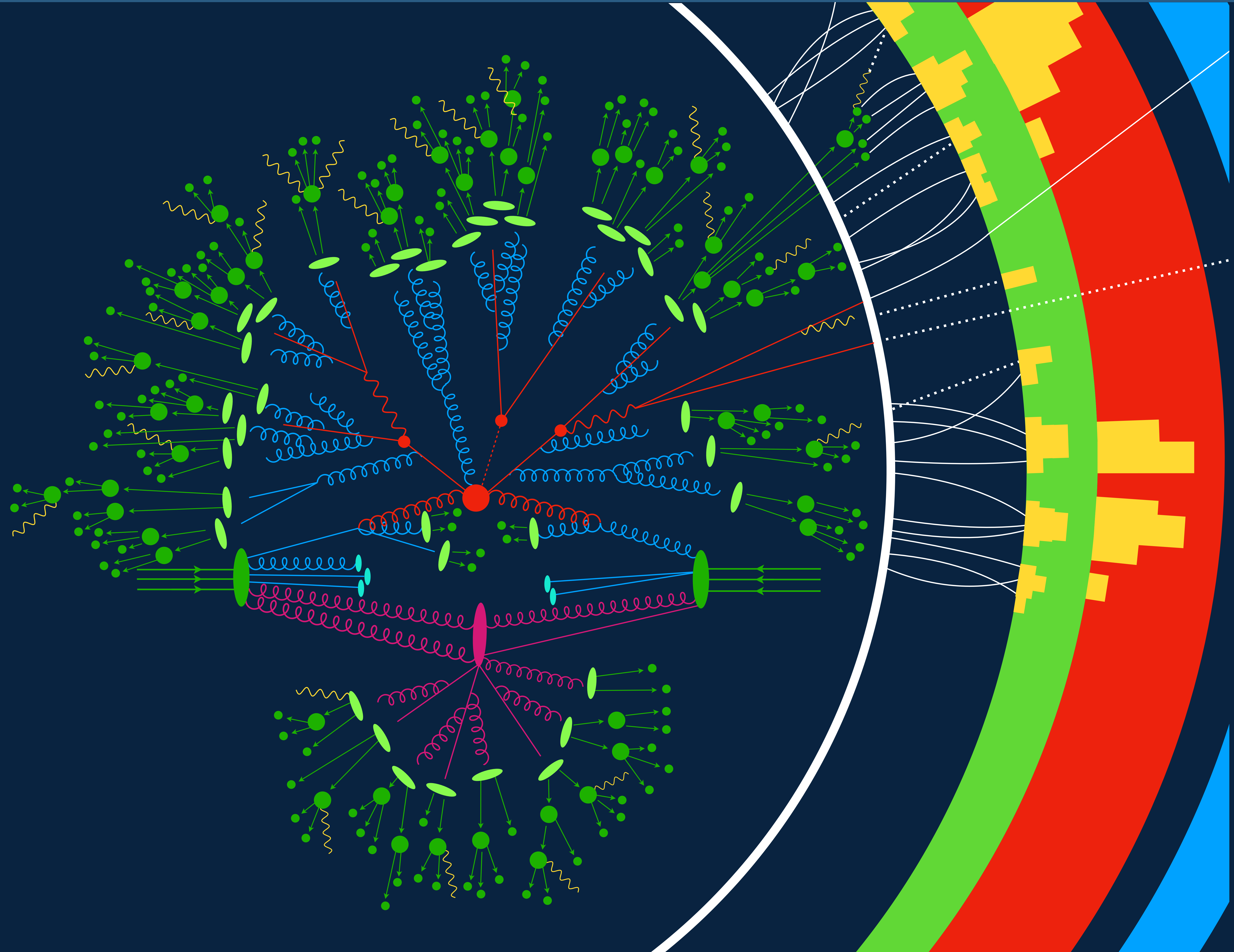


LHC analysis + ML



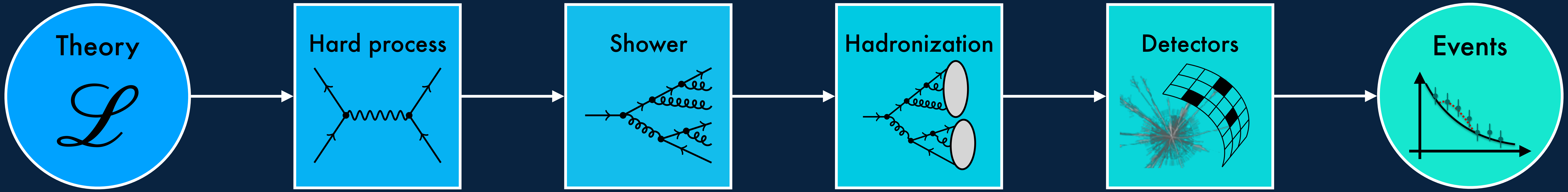
How to simulate LHC events?

How to simulate LHC events

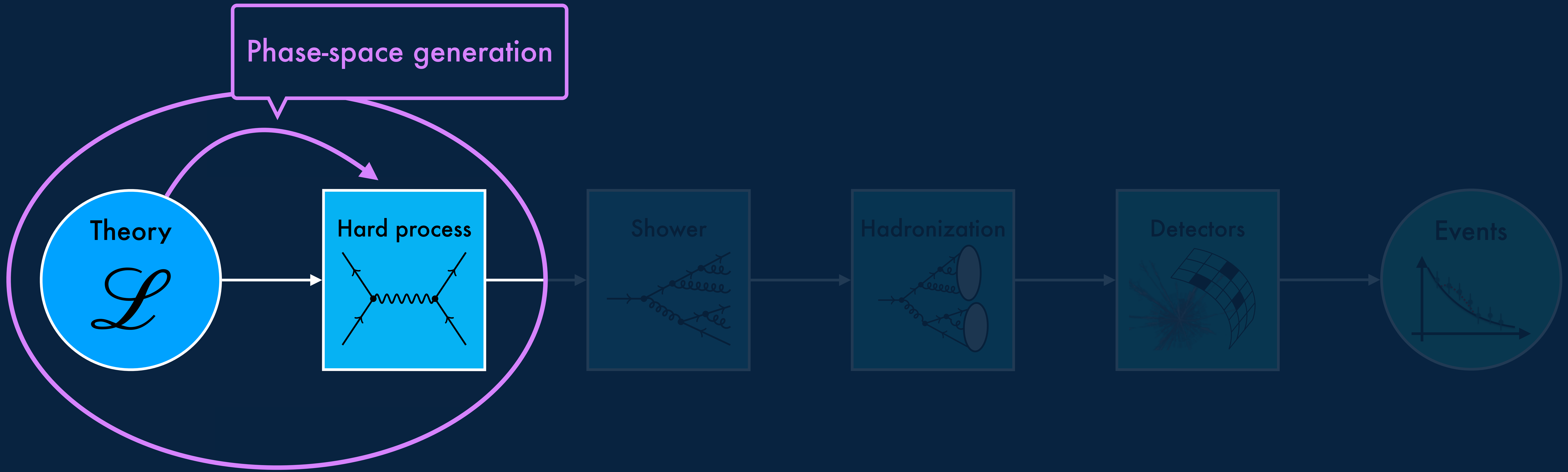


How to simulate LHC events

ML improved simulations

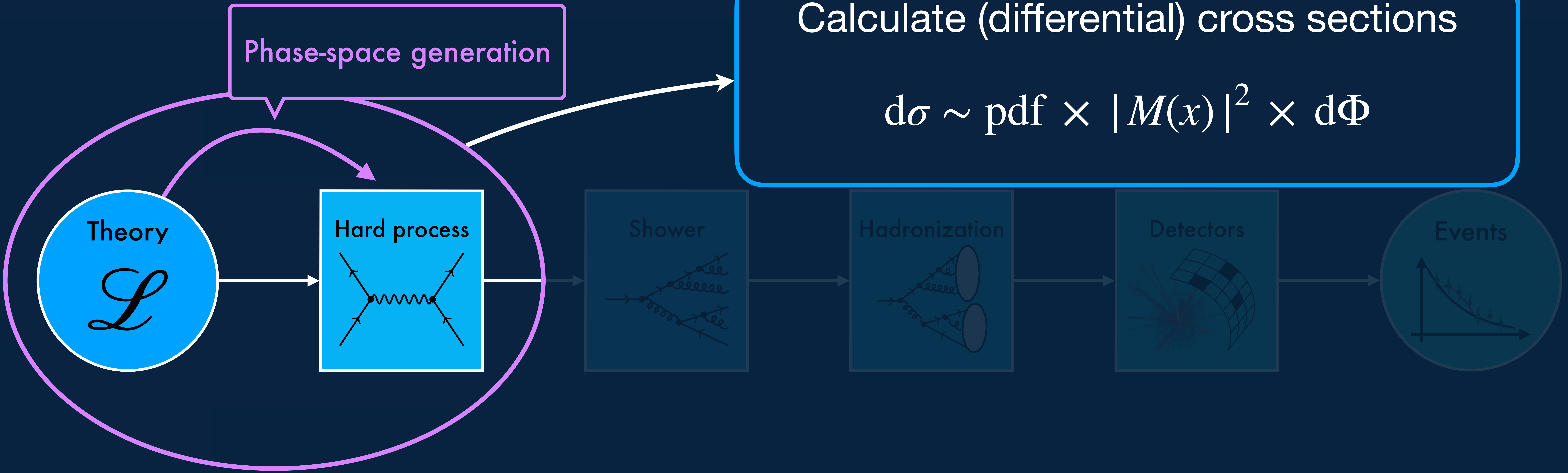


ML improved simulations



BDT [1707.00028, …], NN [1810.11509, 2009.07819, …]
NF [2001.05486, 2001.05478, 2001.10028, 2005.12719, 2112.09145, 2212.06172, …]

ML improved simulations

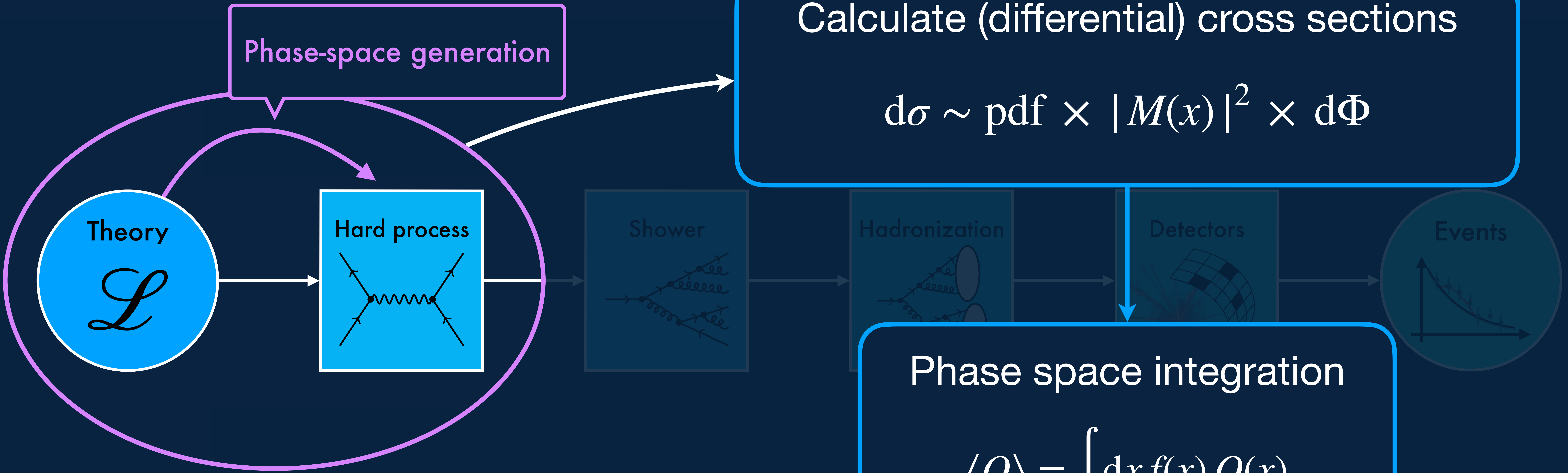


Calculate (differential) cross sections

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

BDT [1707.00028, …], NN [1810.11509, 2009.07819, …]
NF [2001.05486, 2001.05478, 2001.10028, 2005.12719, 2112.09145, 2212.06172, …]

ML improved simulations



Calculate (differential) cross sections

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

Phase space integration

$$\langle O \rangle = \int dx\, f(x)\, O(x)$$

BDT [1707.00028, …], NN [1810.11509, 2009.07819, …]
NF [2001.05486, 2001.05478, 2001.10028, 2005.12719, 2112.09145, 2212.06172, …]

Are there bottlenecks?

Are there bottlenecks?

Yes! Because

- ⊖ Analytic integration **not feasible**: PDFs, cuts, jet algorithm, complex amplitudes, …
- ⊖ Another problem is the **high-dimensionality** of the integrand
- ⊖ Standard numerical methods **scale badly**: error $\sim N^{-2/D} \dots N^{-4/D}$

Are there bottlenecks?

Yes! Because

⊖ Analytic integration **not feasible**: PDFs, cuts, jet algorithm, complex amplitudes, …

⊖ Another problem is the **high-dimensionality** of the integrand

⊖ Standard numerical methods **scale badly**: error $\sim N^{-2/D} \dots N^{-4/D}$

→ Use **Monte Carlo integration** instead: error $\sim N^{-1/2}$

Are there bottlenecks?

Yes! Because

- ⊖ Analytic integration **not feasible**: PDFs, cuts, jet algorithm, complex amplitudes, …
- ⊖ Another problem is the **high-dimensionality** of the integrand
- ⊖ Standard numerical methods **scale badly**: error $\sim N^{-2/D} \dots N^{-4/D}$

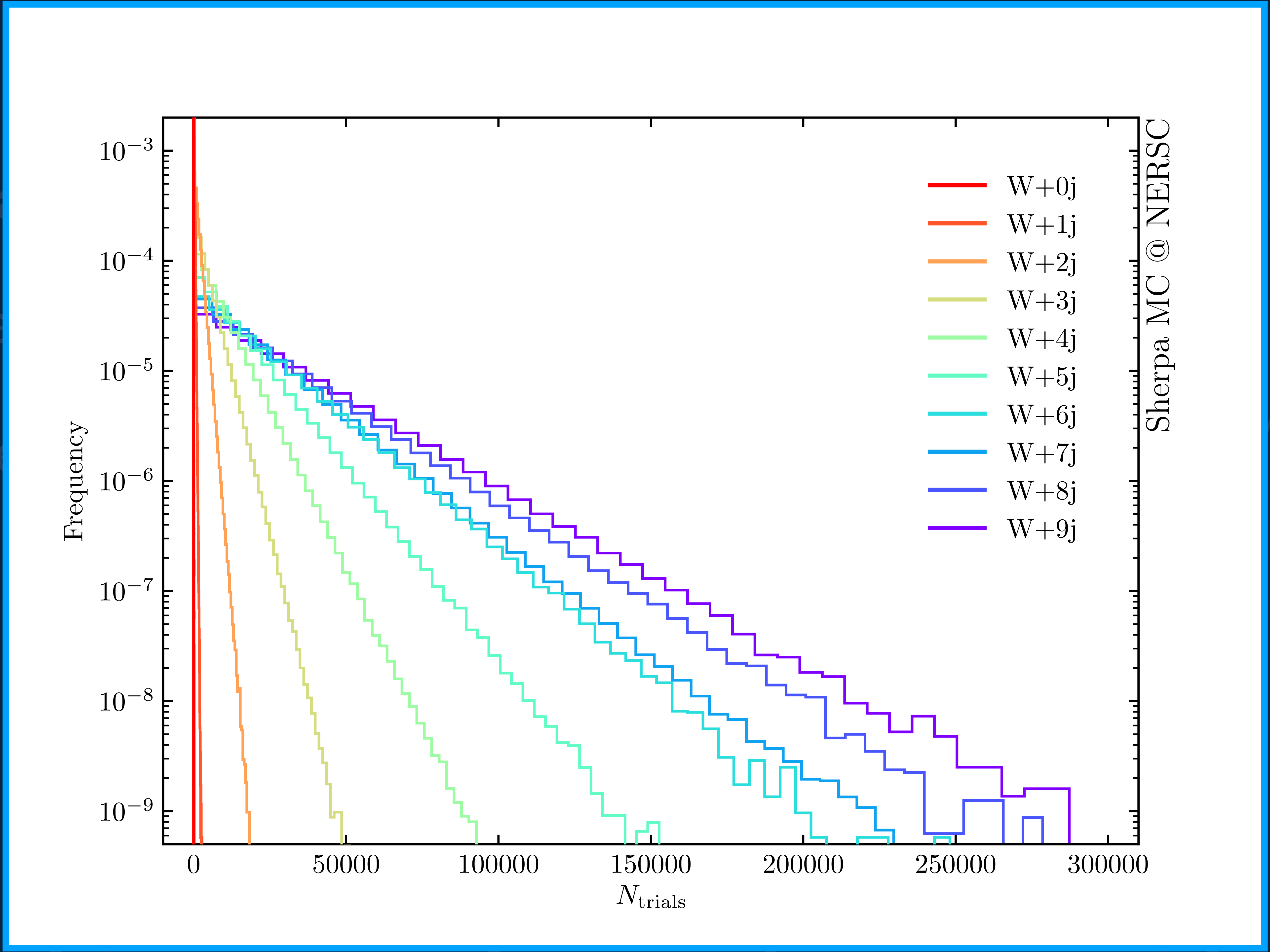
→ Use **Monte Carlo integration** instead: error $\sim N^{-1/2}$

⚠ **Efficiency still a problem!** ⚠

Are there bottlenecks?

- Analytic integra… …lex amplitudes, …
- Another proble…
- Standard num…
- Use Monte Ca…

Höche et al. [1905.05120]

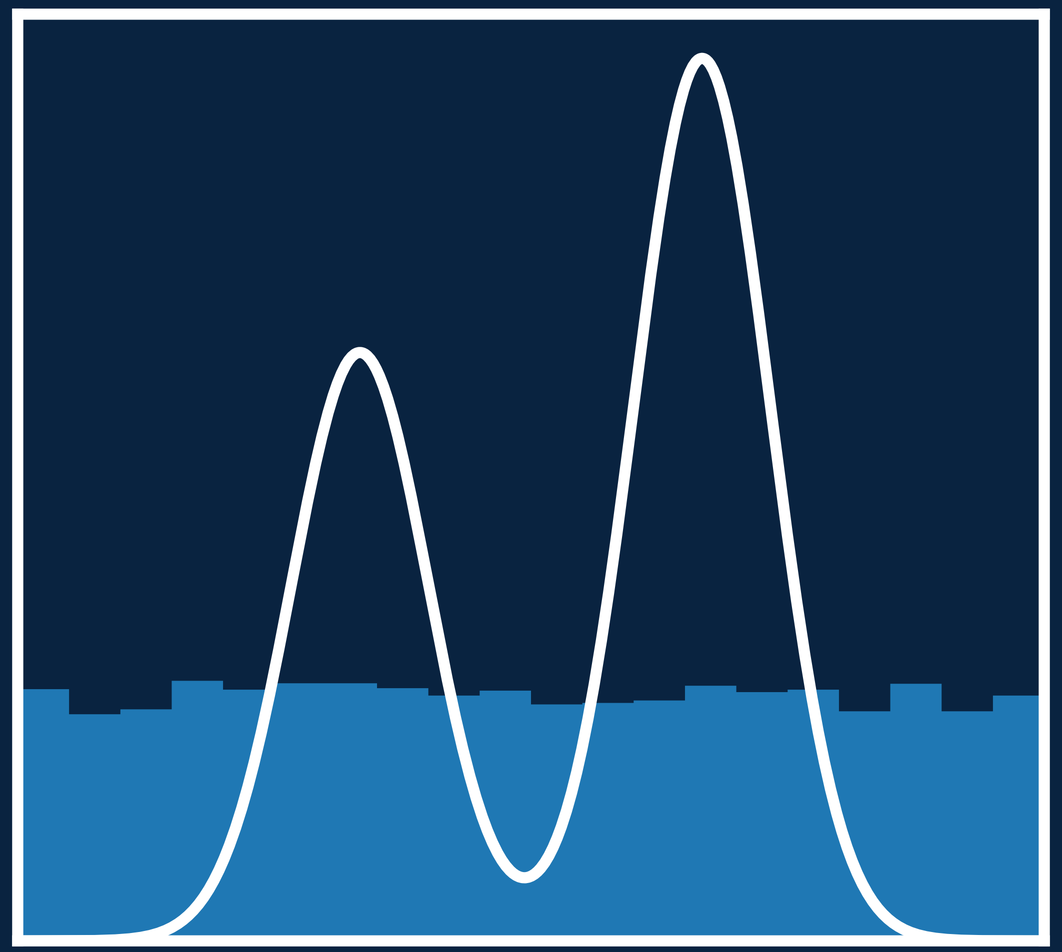


Monte Carlo integration

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

Monte Carlo integration

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

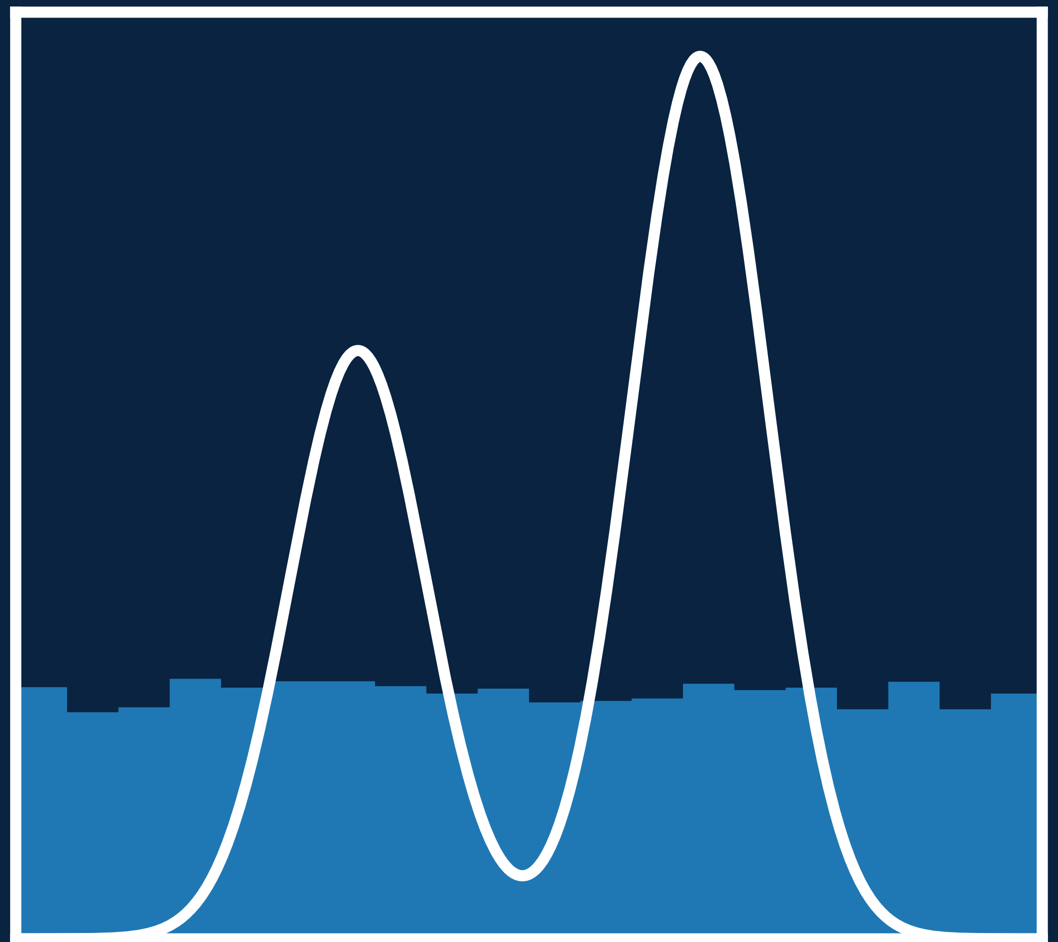


Flat sampling:
inefficient

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

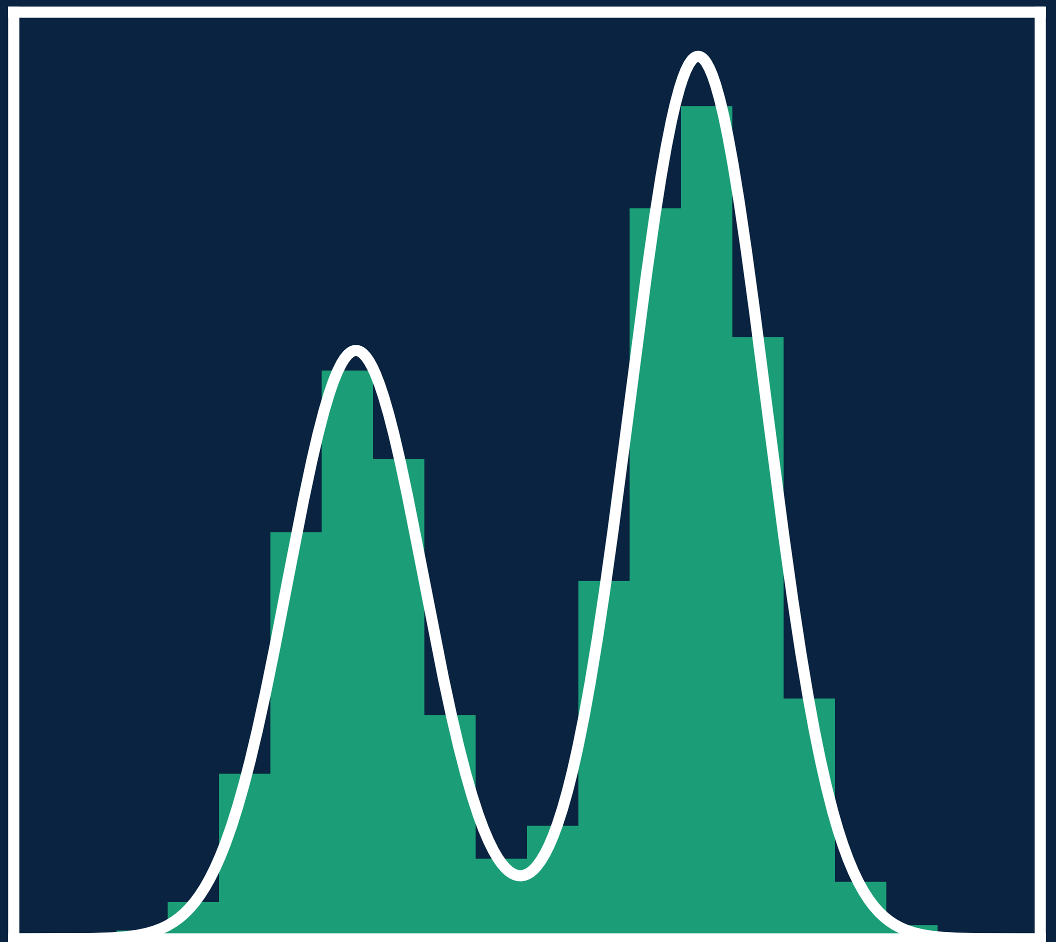
Monte Carlo integration

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



Flat sampling:
inefficient

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

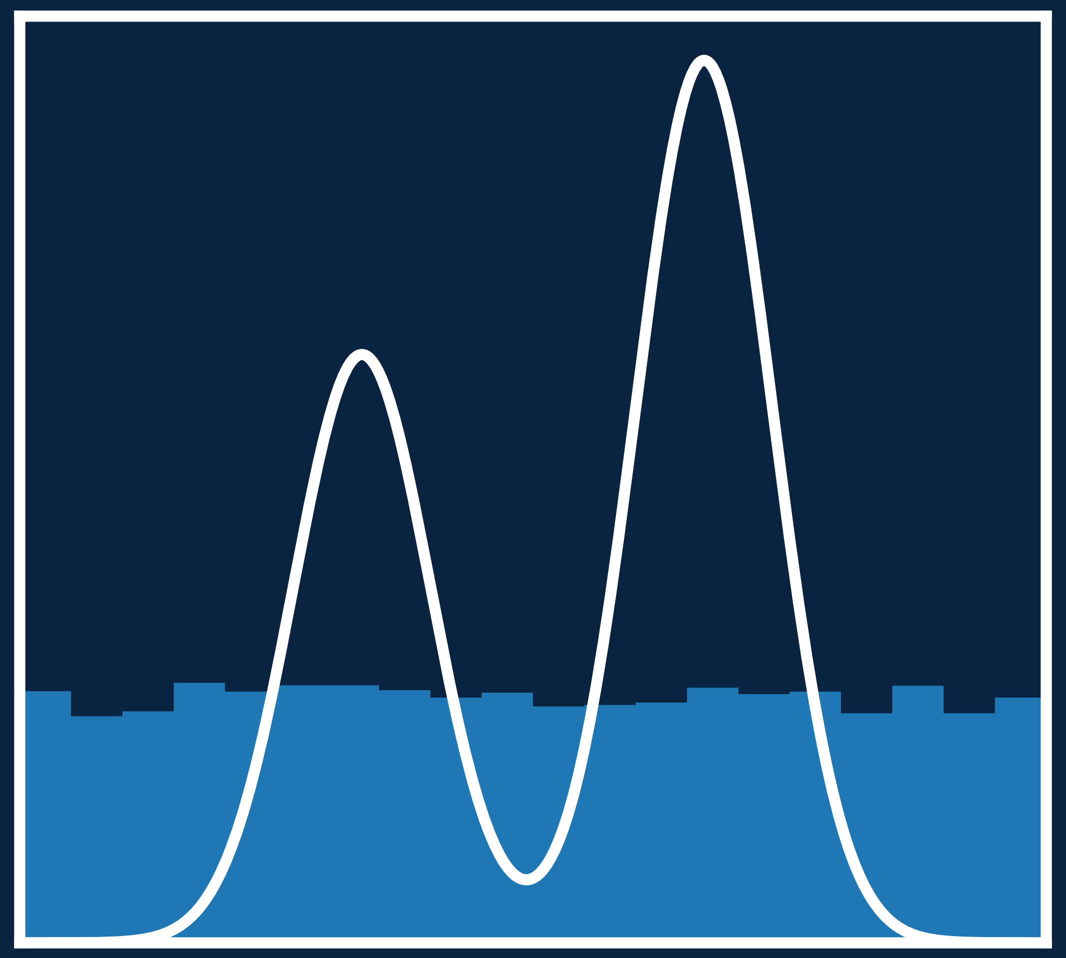


Importance sampling:
find g close to f

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

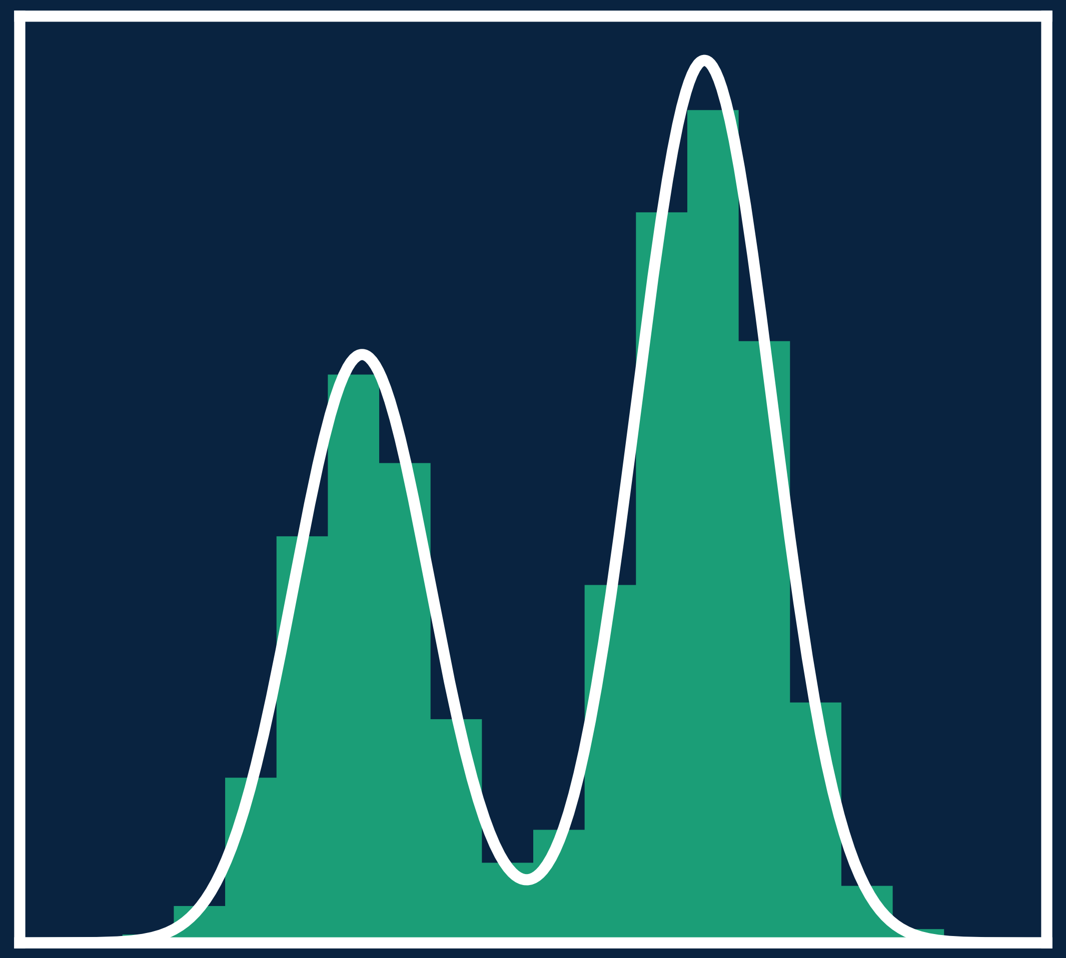
Monte Carlo integration

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



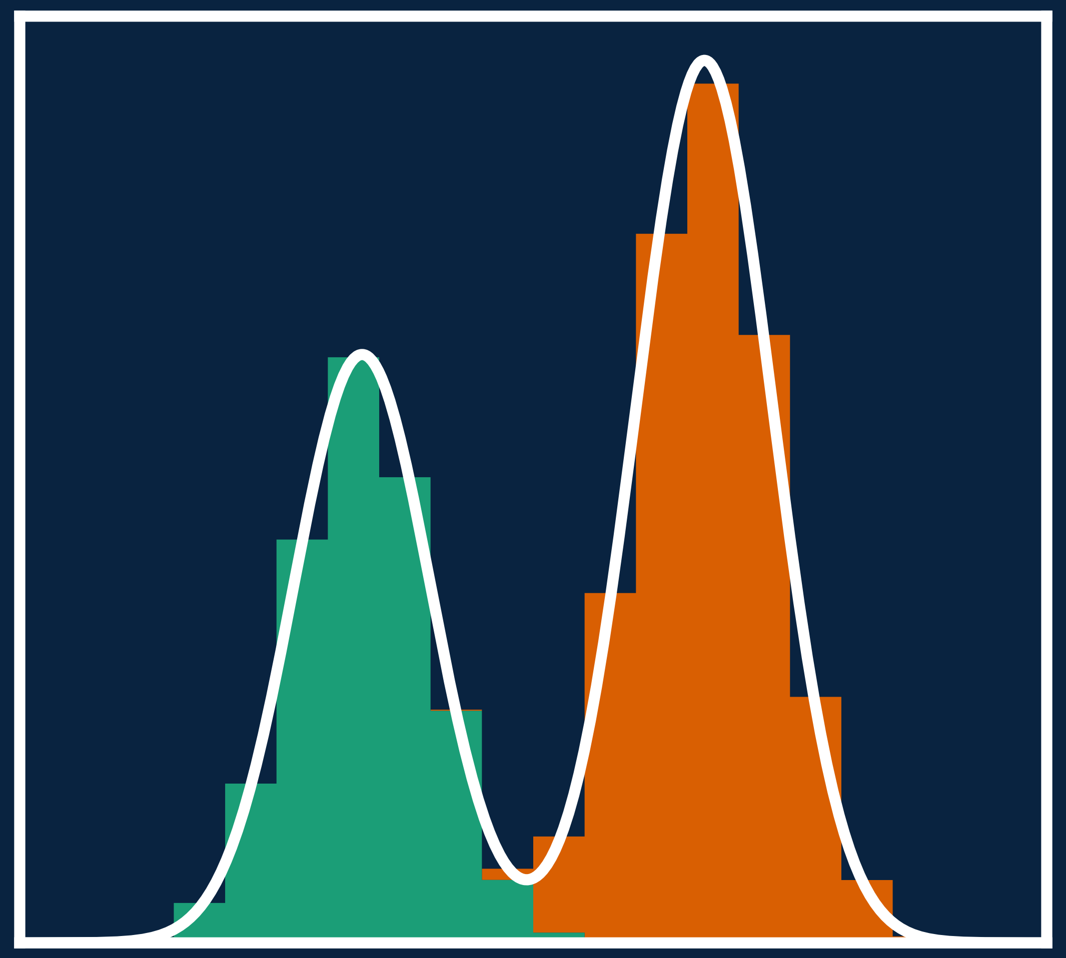
Flat sampling:
inefficient

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



Importance sampling:
find g close to f

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



Multi-channel:
one map for each channel

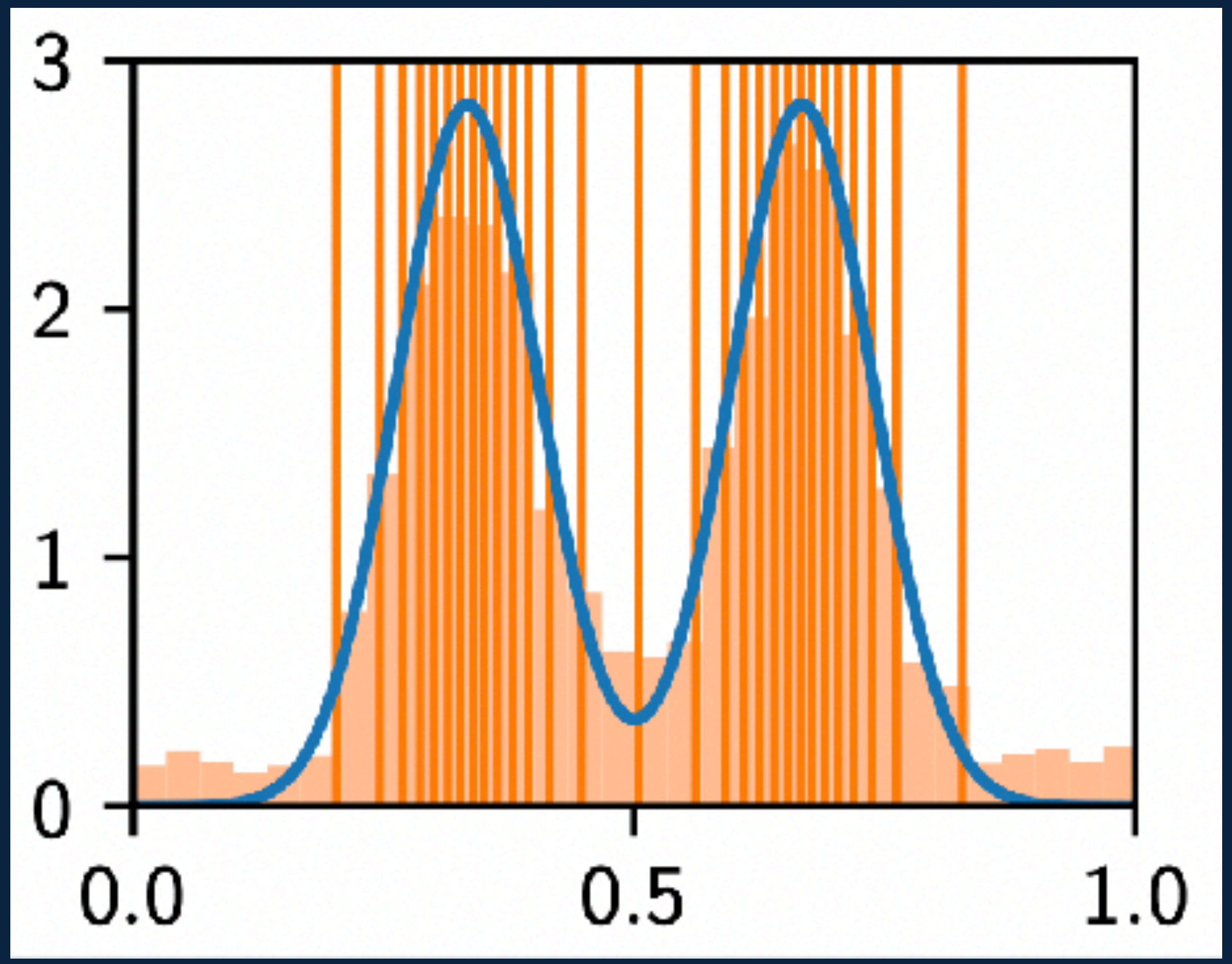
$$I = \sum_i \left\langle \alpha_i(x) \frac{f(x)}{g_i(x)} \right\rangle_{x \sim g_i(x)}$$

Importance sampling — VEGAS

Factorize probability

$$p(x) = p(x_1)\cdots p(x_n)$$

Fit bins with equal probability
and varying width

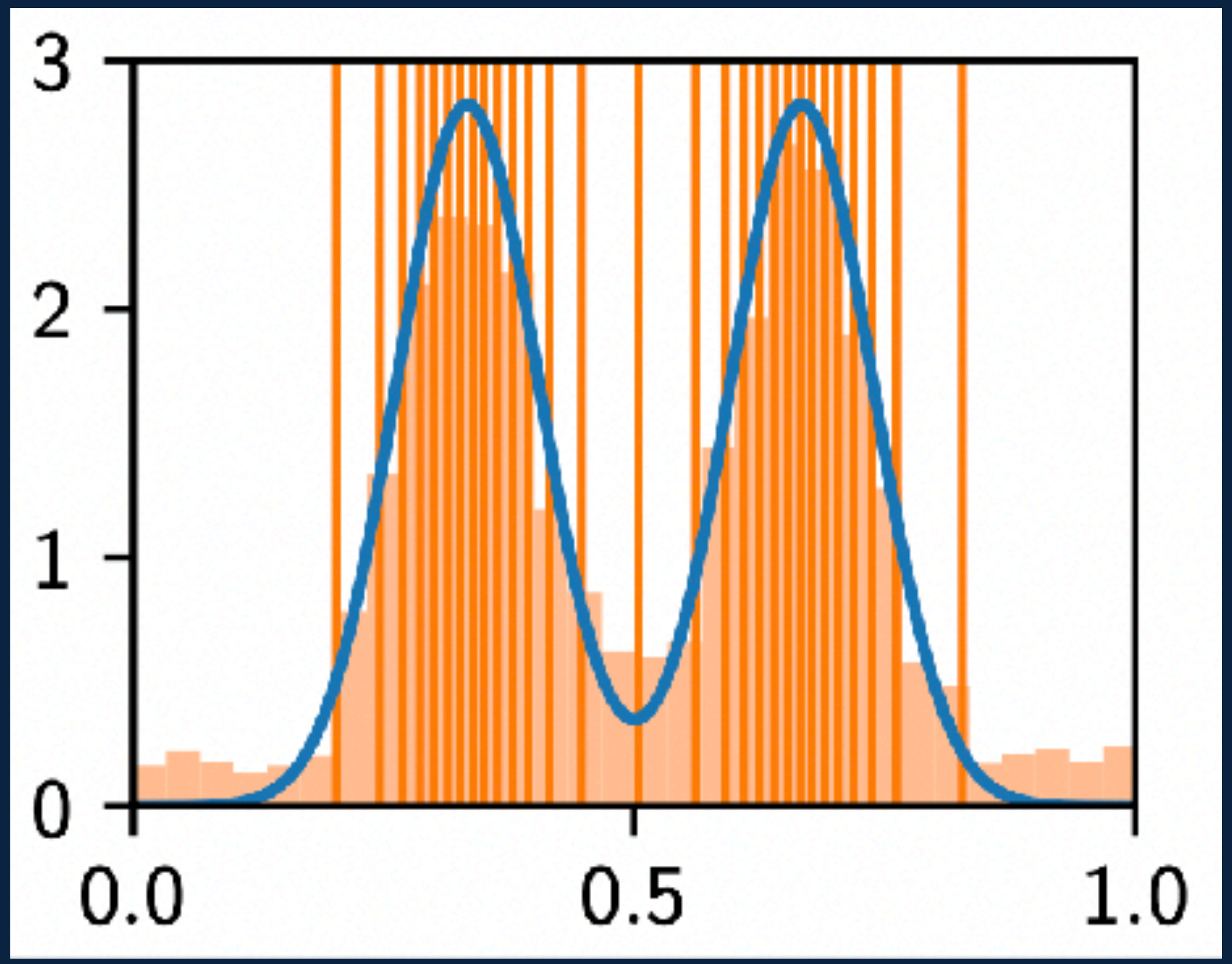


Importance sampling — VEGAS

Factorize probability

$$p(x) = p(x_1)\cdots p(x_n)$$

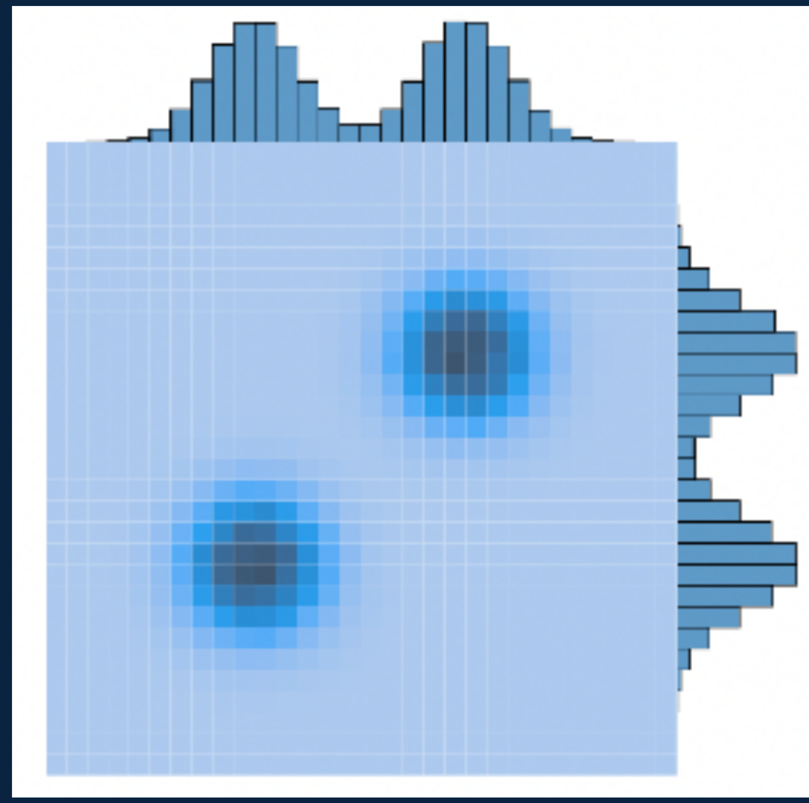
Fit bins with equal probability and varying width

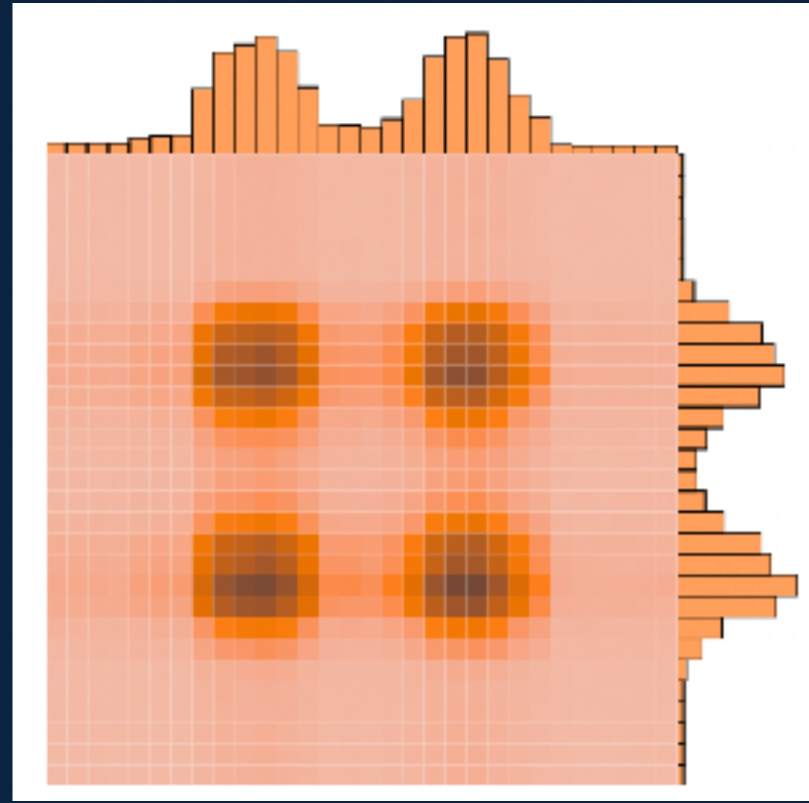


⊕ Computationally cheap

⊖ High-dim and rich peaking functions
→ **slow convergence**

⊖ Peaks not aligned with grid axes
→ **phantom peaks**





Importance sampling — NN

Using a Neural Network

⊕ Unbinned and no grids

→ **no "phantom peaks"**

⊖ Bijectivity not guaranteed

→ **training unstable**

⊖ Numerical Jacobians

→ **slow training and evaluation**

[1707.00028, 1810.11509, 2009.07819]

Importance sampling — Flow

Using a Neural Network

- ⊕ Unbinned and no grids
  → **no "phantom peaks"**
- ⊖ Bijectivity not guaranteed
  → **training unstable**
- ⊖ Numerical Jacobians
  → **slow training and evaluation**

[1707.00028, 1810.11509, 2009.07819]

→

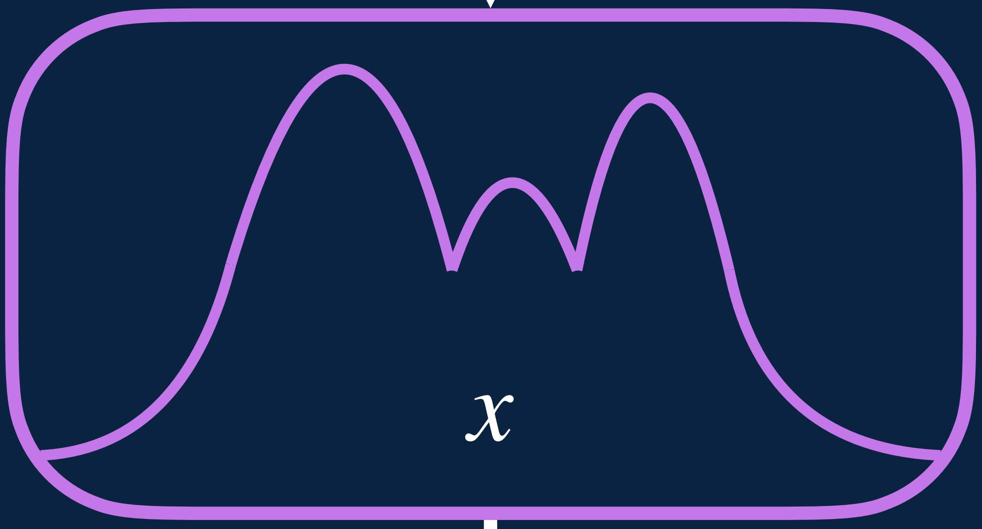
Using a Normalizing Flow

- ⊕ Invertibility
  → **bijective mapping**
- ⊕ tractable Jacobians
  → **fast training and evaluation**

[2001.05478, 2001.05486, 2001.10028,2005.12719, 2112.09145]

Sampling

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



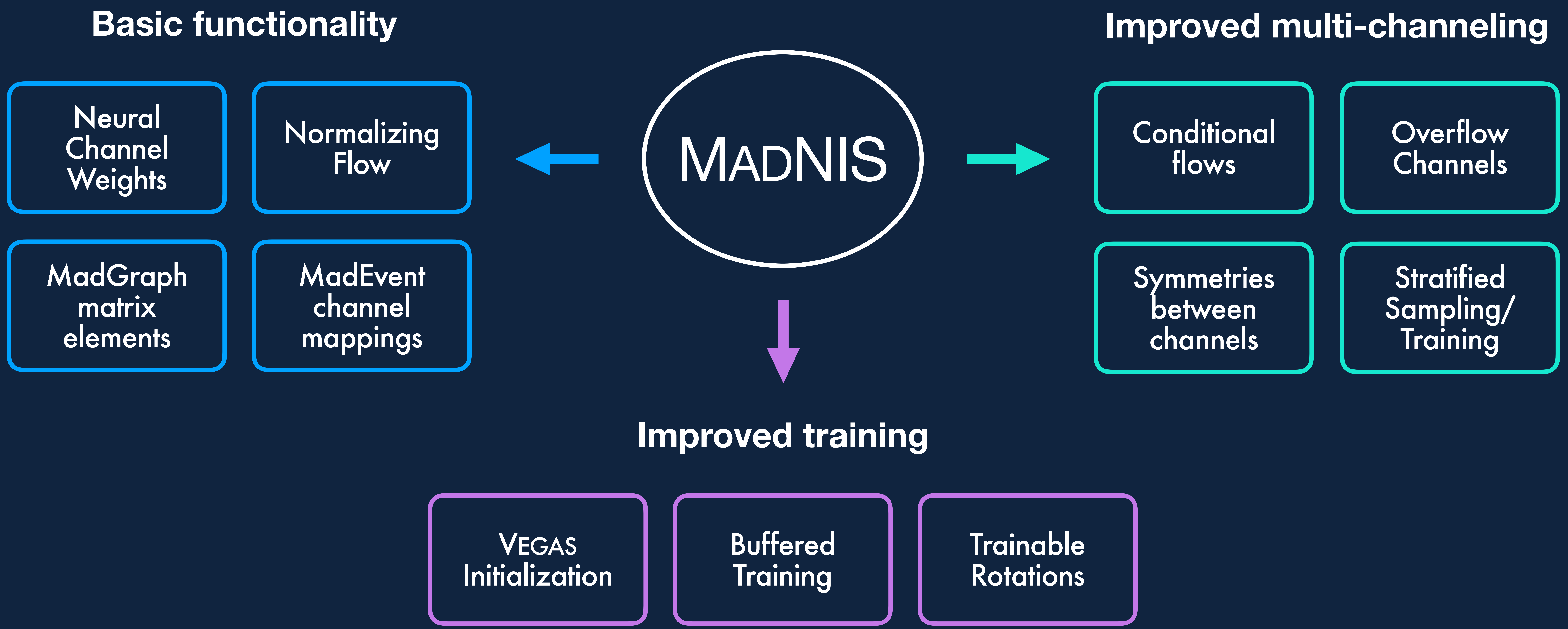
Training

MadNIS

Neural Importance Sampling

[2212.06172]

MadNIS — Overview



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

MadNIS

Basic Functionality

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

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

$\downarrow$

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

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

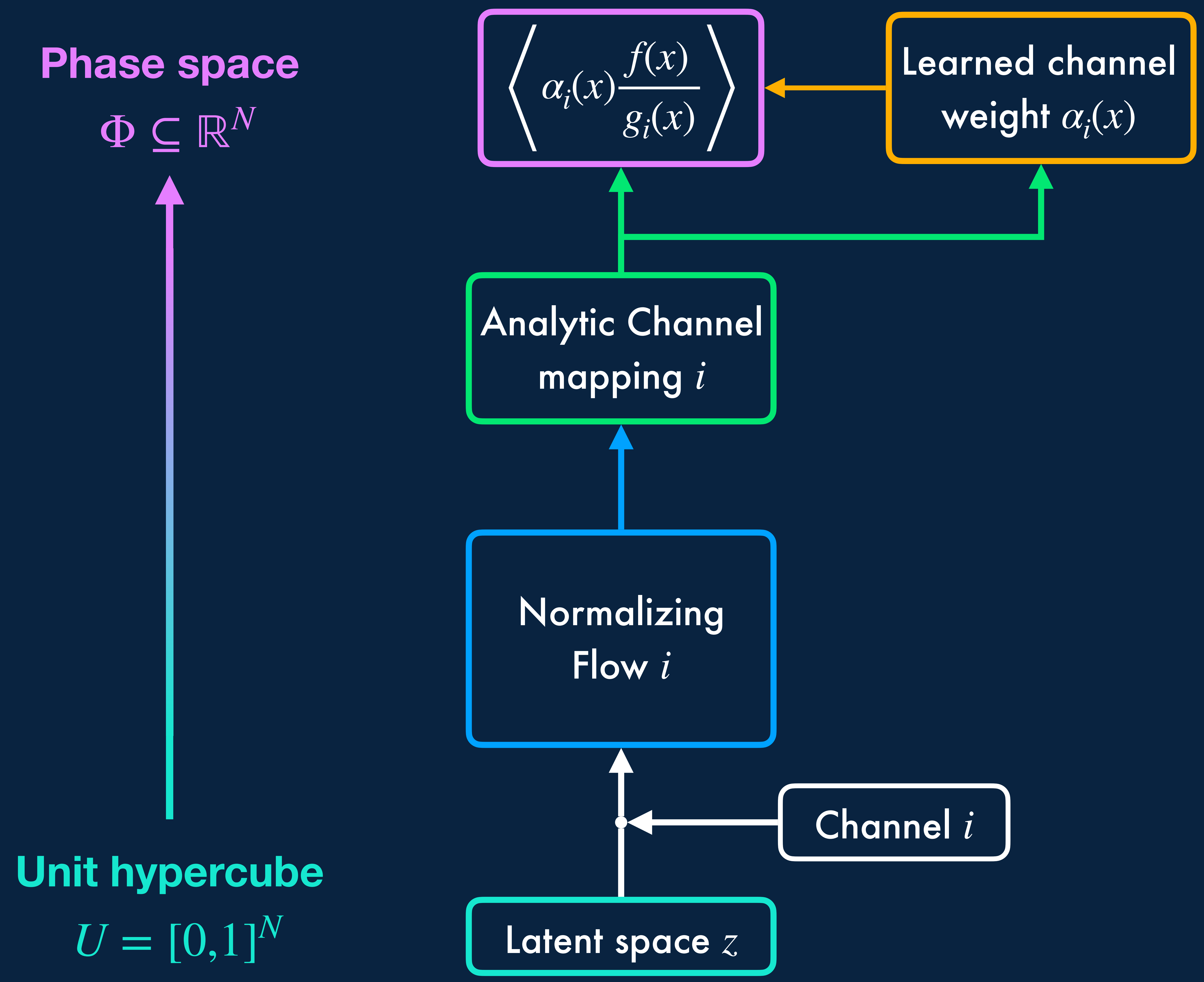
Use physics knowledge to construct channel and mappings

Normalizing flow to refine channel mappings

Fully connected network to refine channel weights

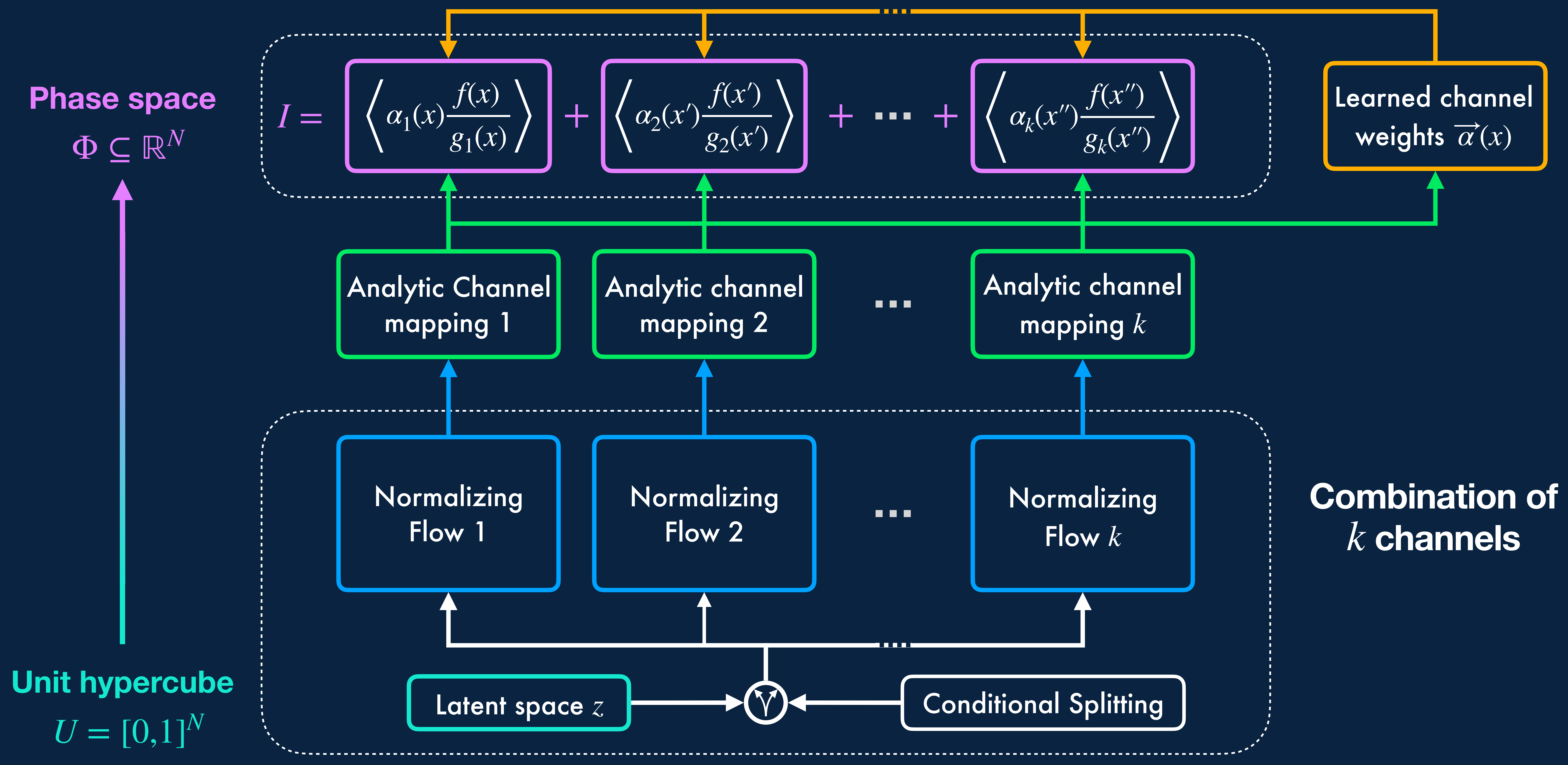
Update simultanously with variance as loss function

MadNIS — Basic functionality

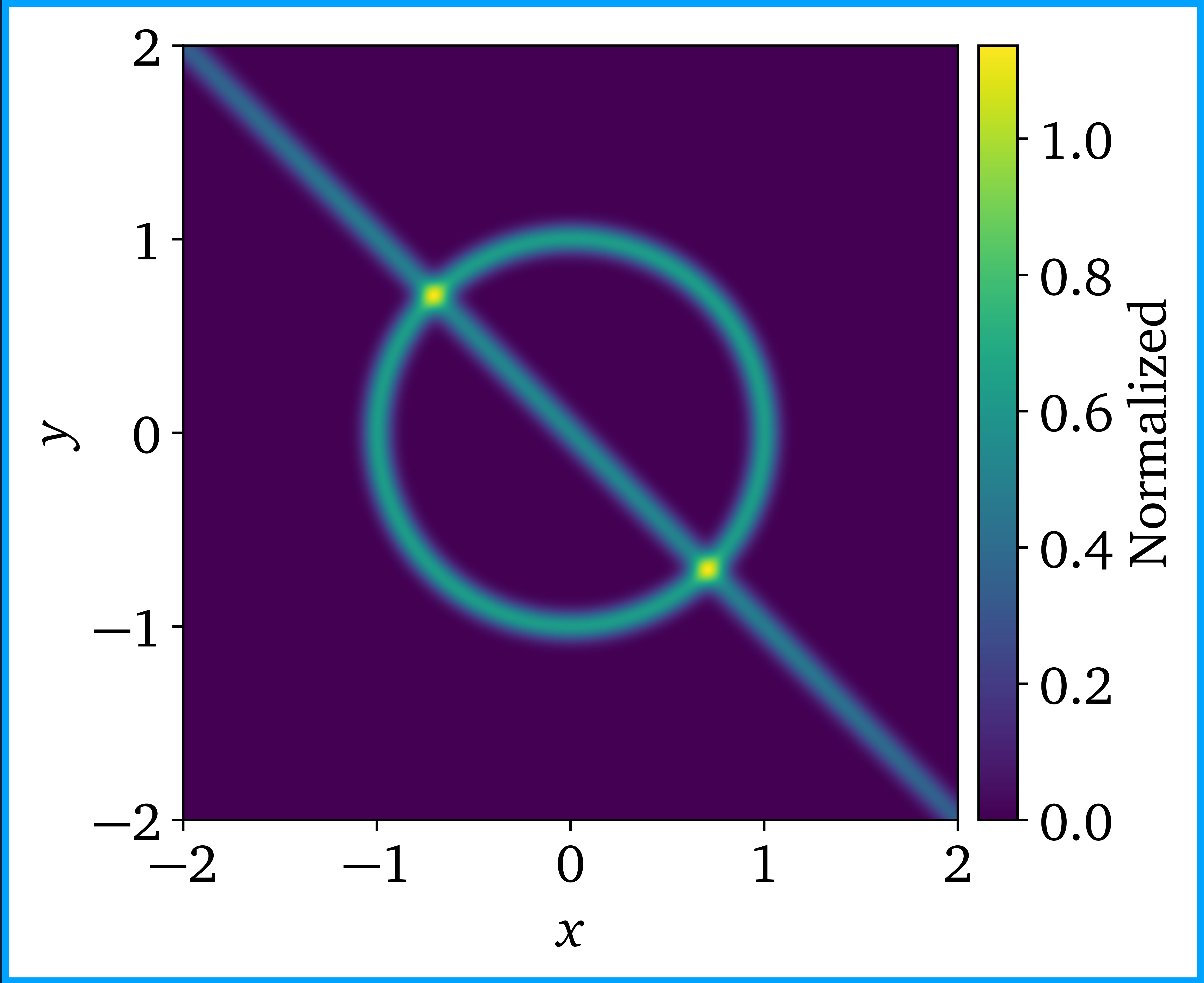


Single channel i

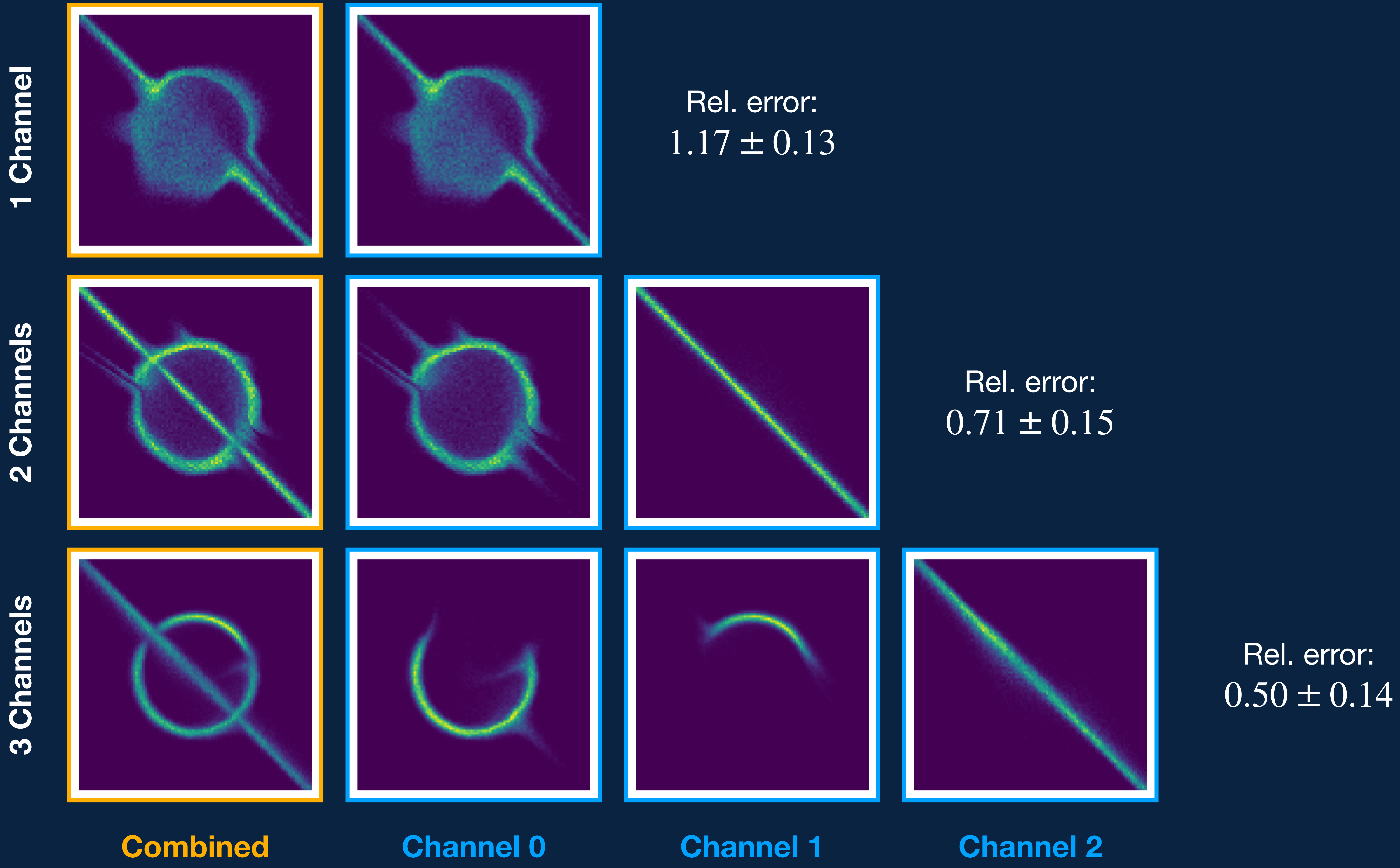
MadNIS — Basic functionality



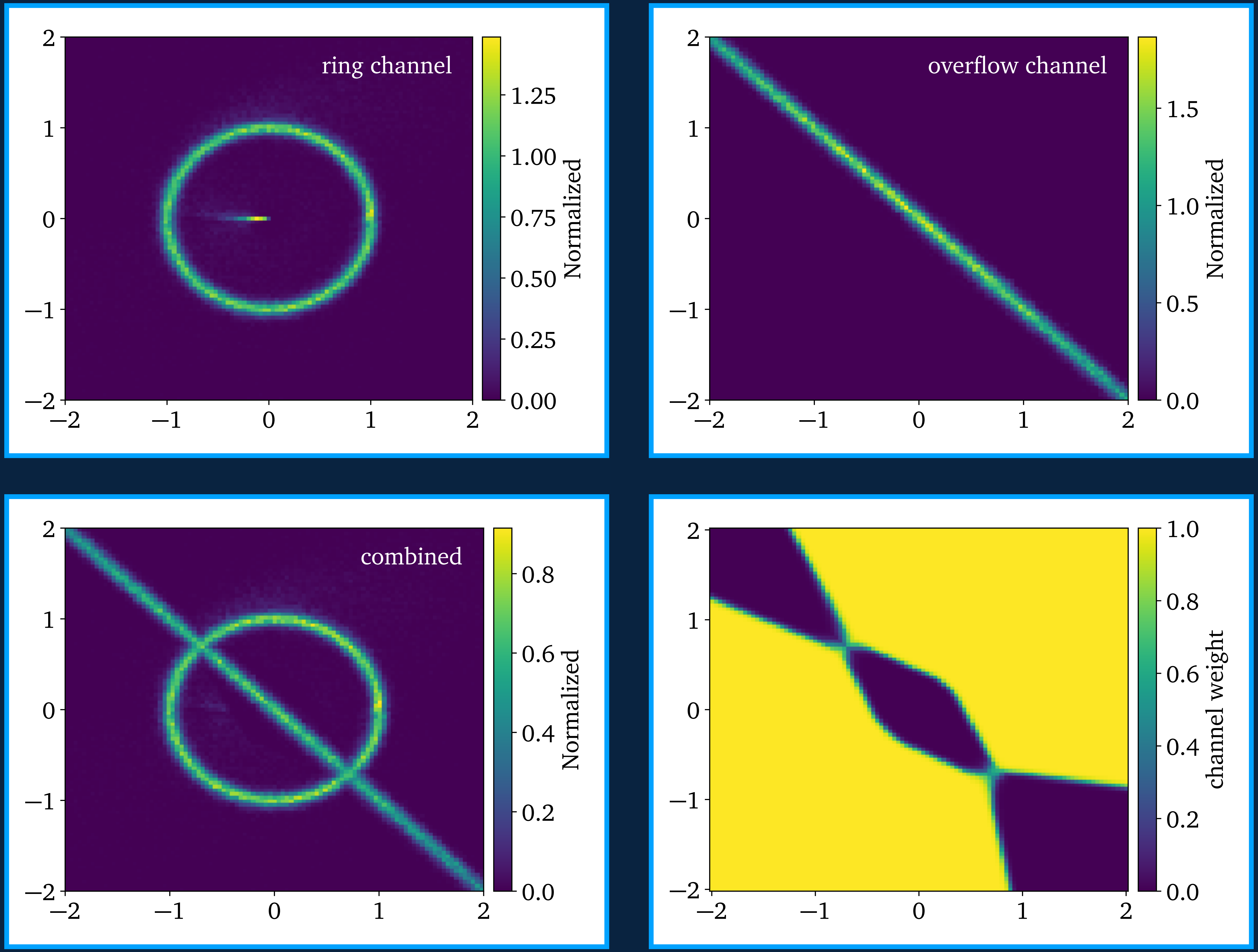
Toy example — Crossed ring



Toy example — Crossed ring



Toy example — Crossed ring



Rel. error:
0.37 ± 0.05

MadNIS

Additional Features

MadNIS — Overview

MadNIS

Basic functionality

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

Improved multi-channeling

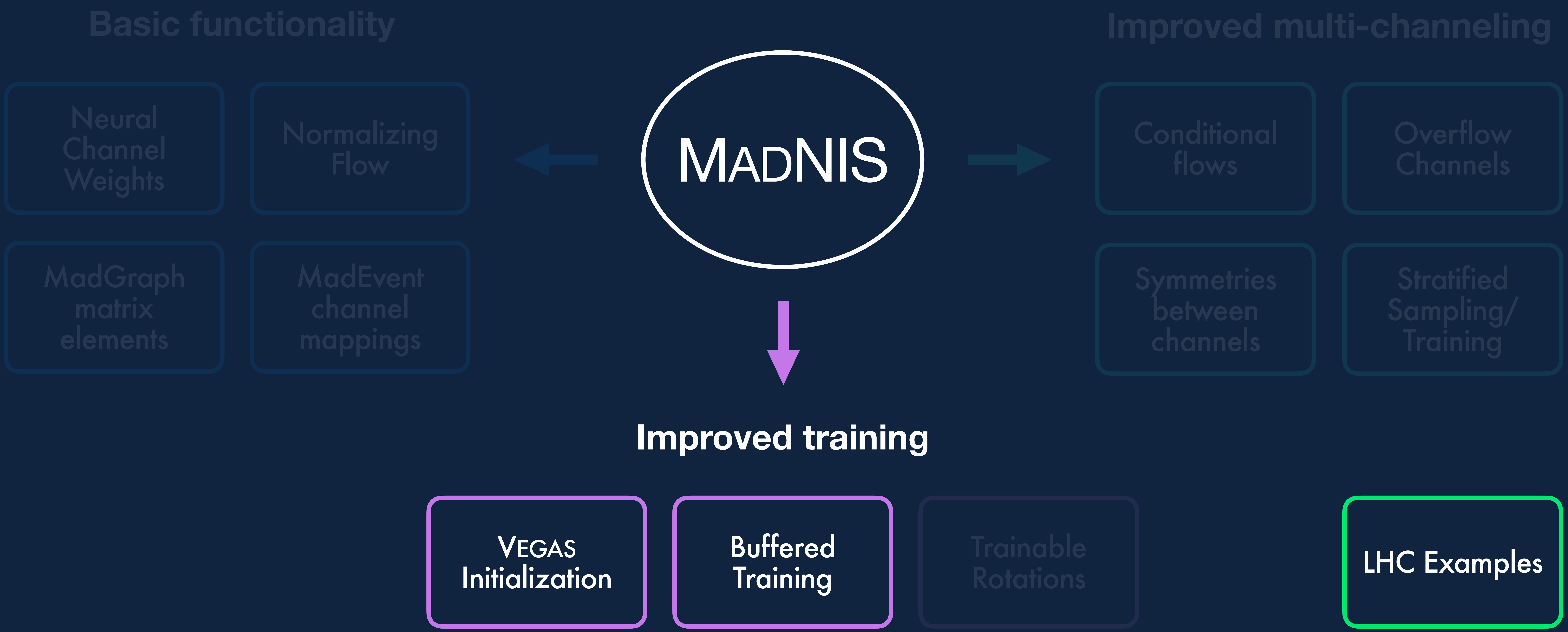
- Conditional flows
- Overflow Channels
- Symmetries between channels
- Stratified Sampling/ Training

Improved training

- VEGAS Initialization
- Buffered Training
- Trainable Rotations

LHC Examples

MadNIS — Overview



VEGAS initialization

Basic functionality

Neural Channel Weights

Normalizing Flow

MadGraph matrix elements

MadEvent channel mappings

MADNIS

Improved multi-channeling

Conditional flows

Overflow Channels

Symmetries between channels

Stratified Sampling/ Training

Improved training

VEGAS Initialization

Buffered Training

Trainable Rotations

LHC Examples

VEGAS initialization

	VEGAS	Flow
Training	Fast	Slow
Correlations	No	Yes

Combine advantages:

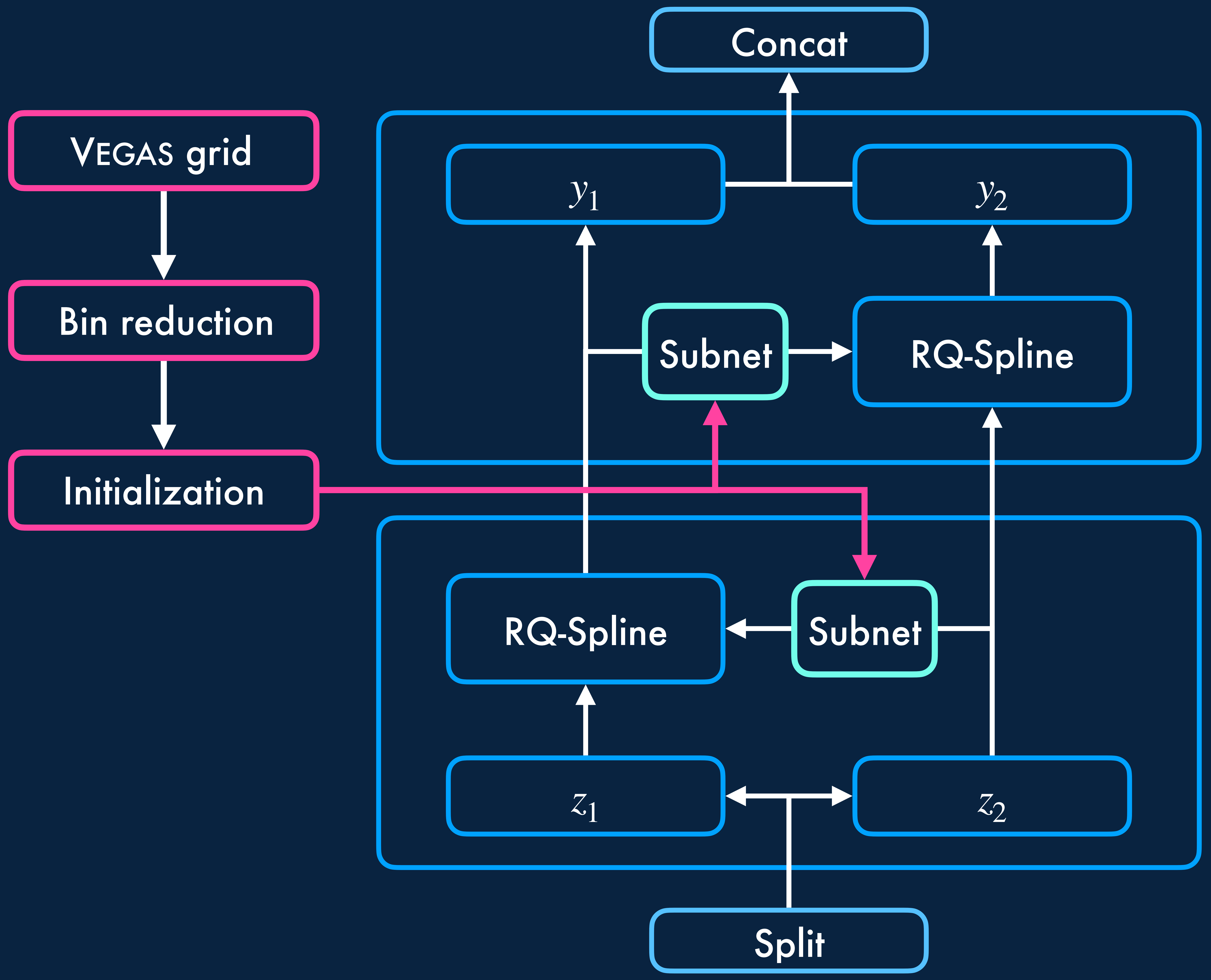
Pre-trained VEGAS grid as starting point for flow training

VEGAS initialization

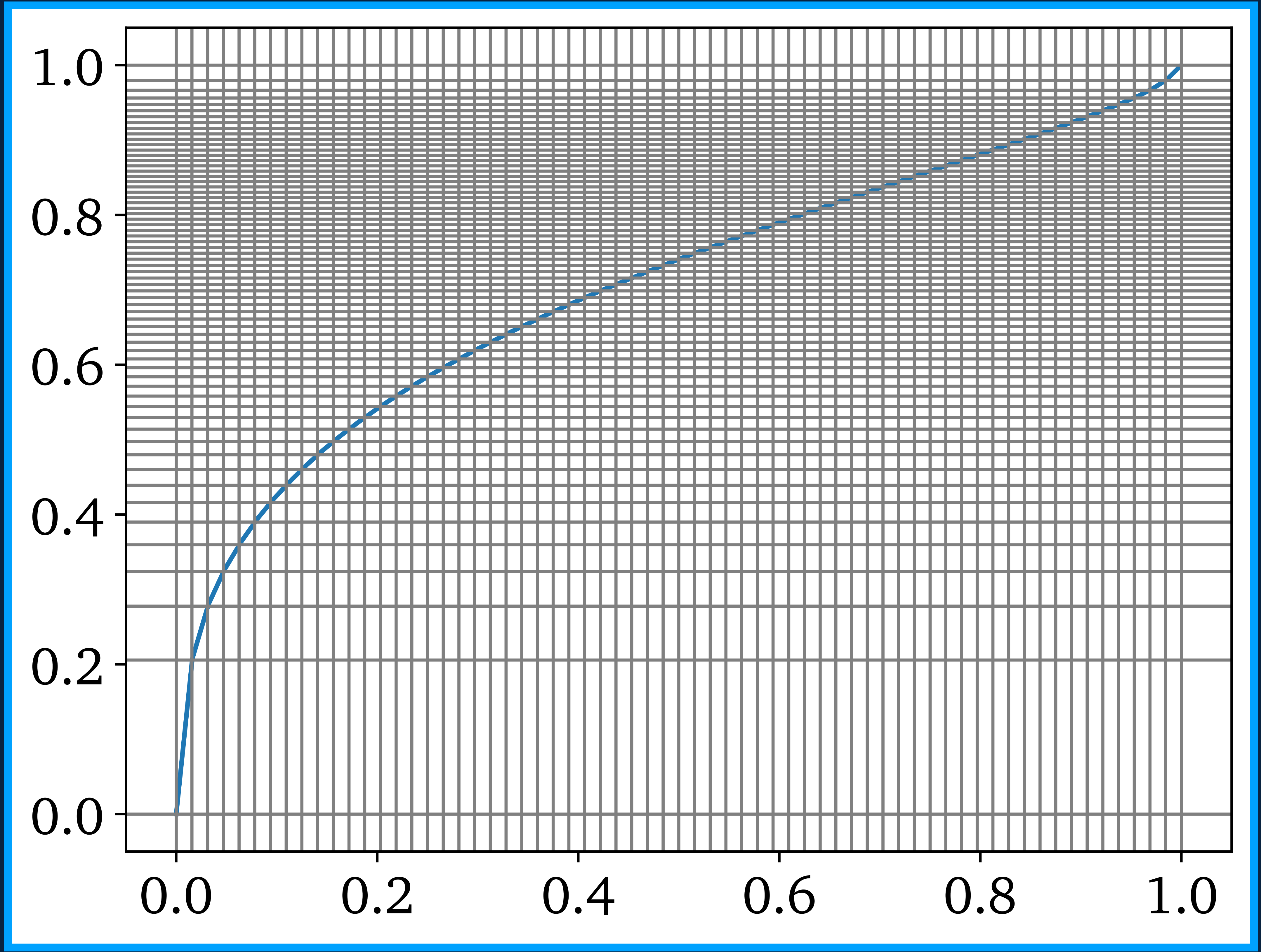
	VEGAS	Flow
Training	Fast	Slow
Correlations	No	Yes

Combine advantages:

Pre-trained VEGAS grid as starting point for flow training

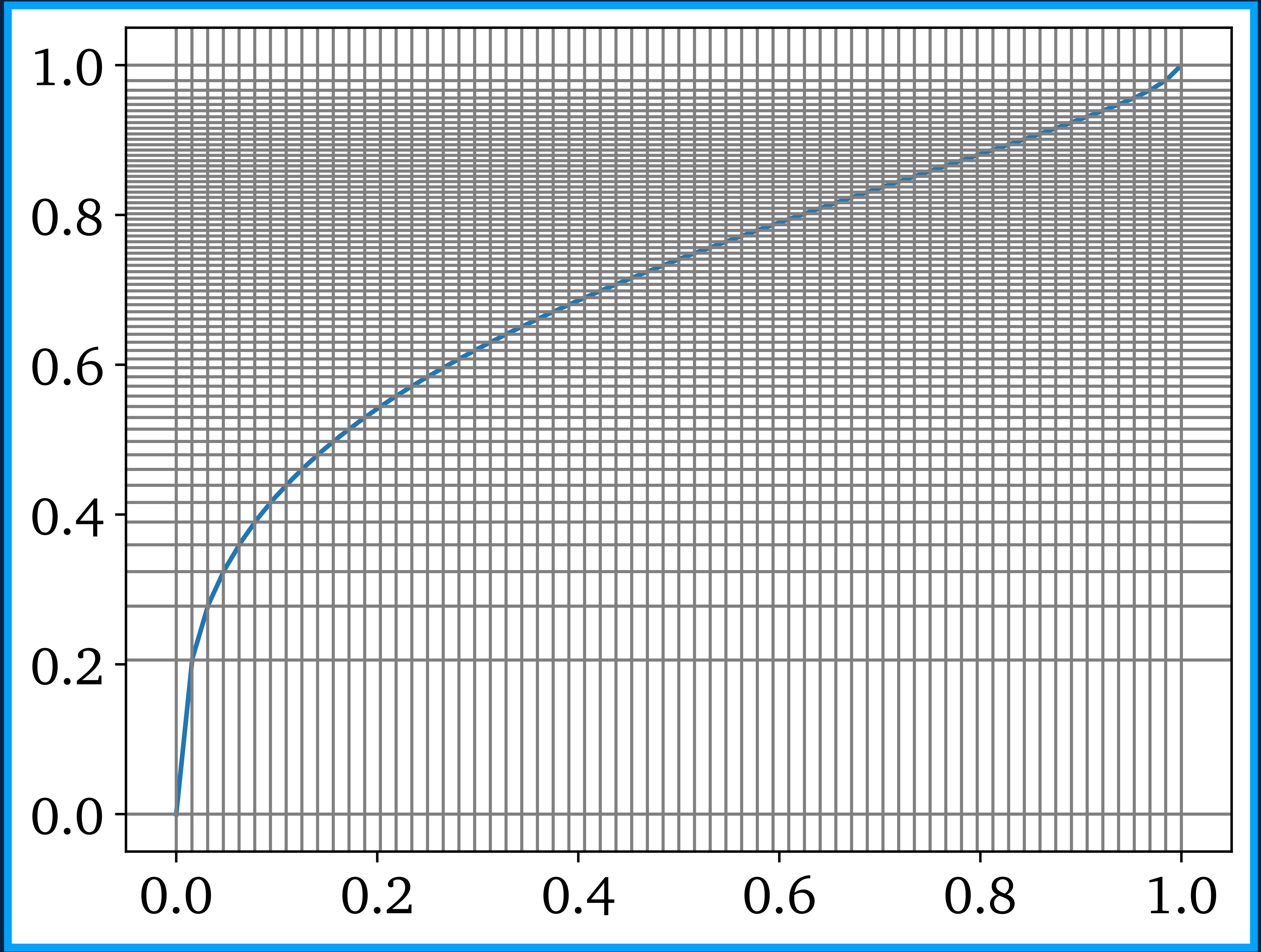


Bin reduction

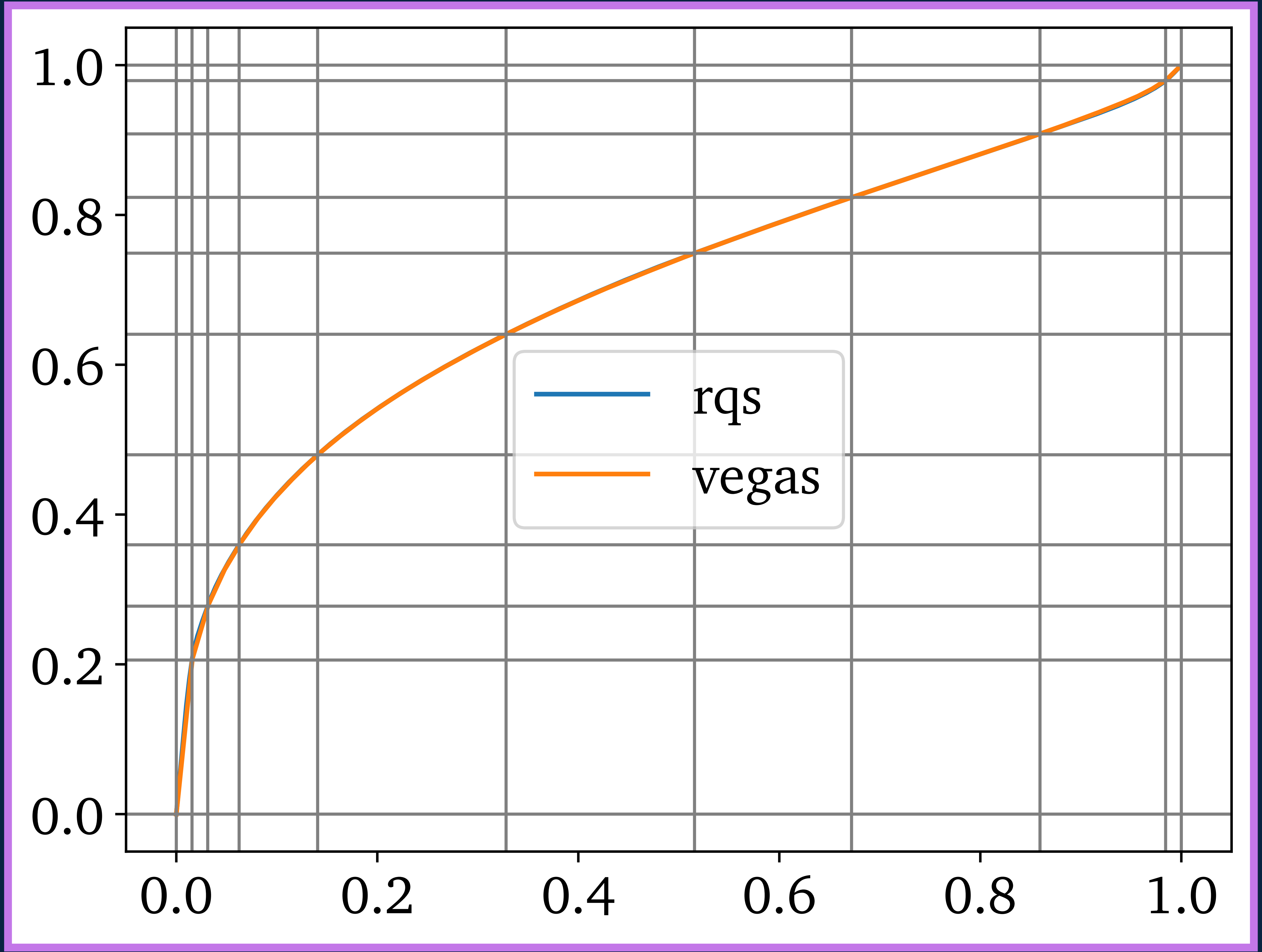


64 VEGAS bins

Bin reduction

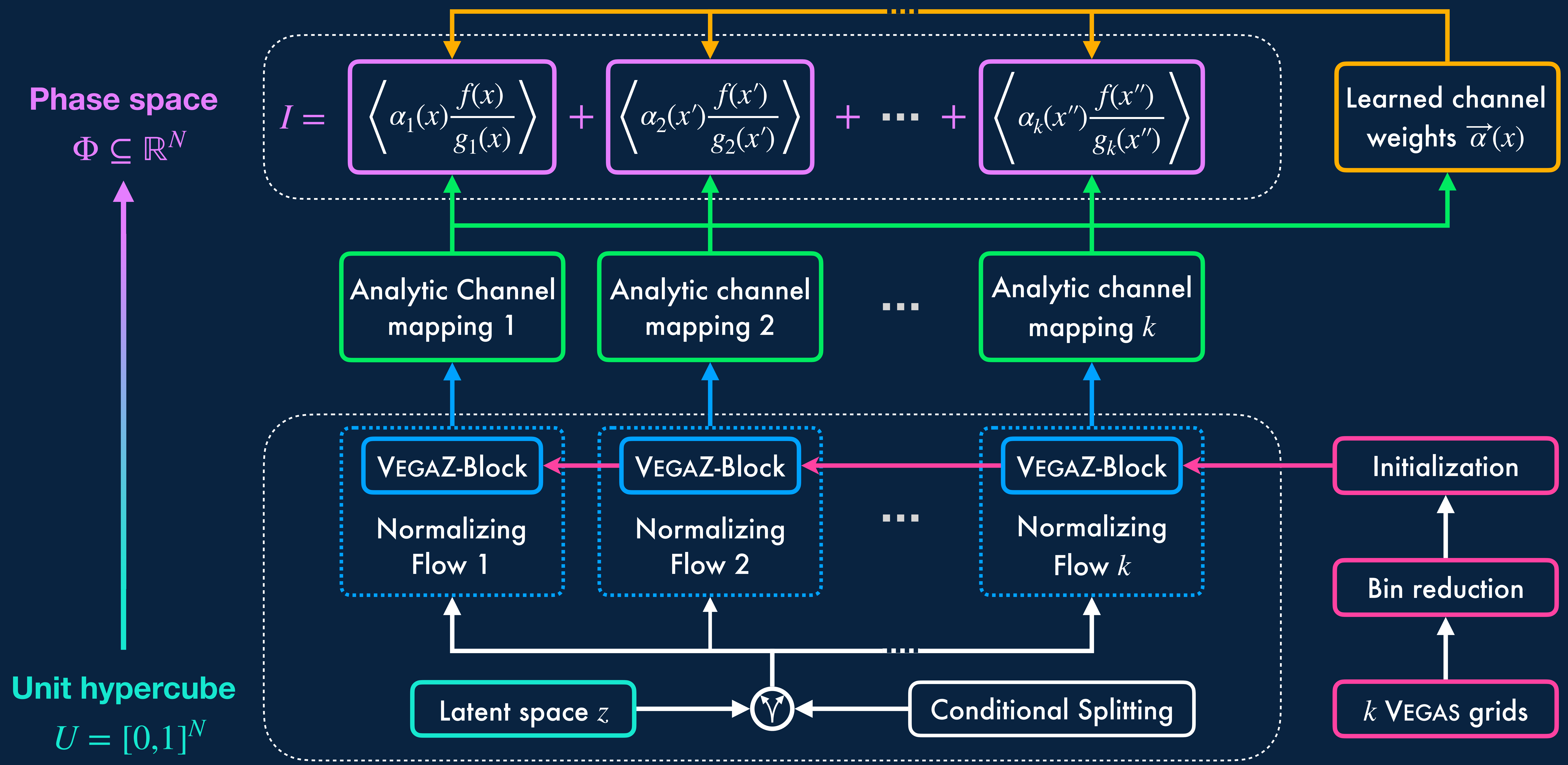


64 VEGAS bins



10 RQS bins

MadNIS — VEGAZ-Block



MadNIS — Overview

Basic functionality

Neural Channel Weights

Normalizing Flow

MadGraph matrix elements

MadEvent channel mappings

MADNIS

Improved multi-channeling

Conditional flows

Overflow Channels

Symmetries between channels

Stratified Sampling/ Training

Improved training

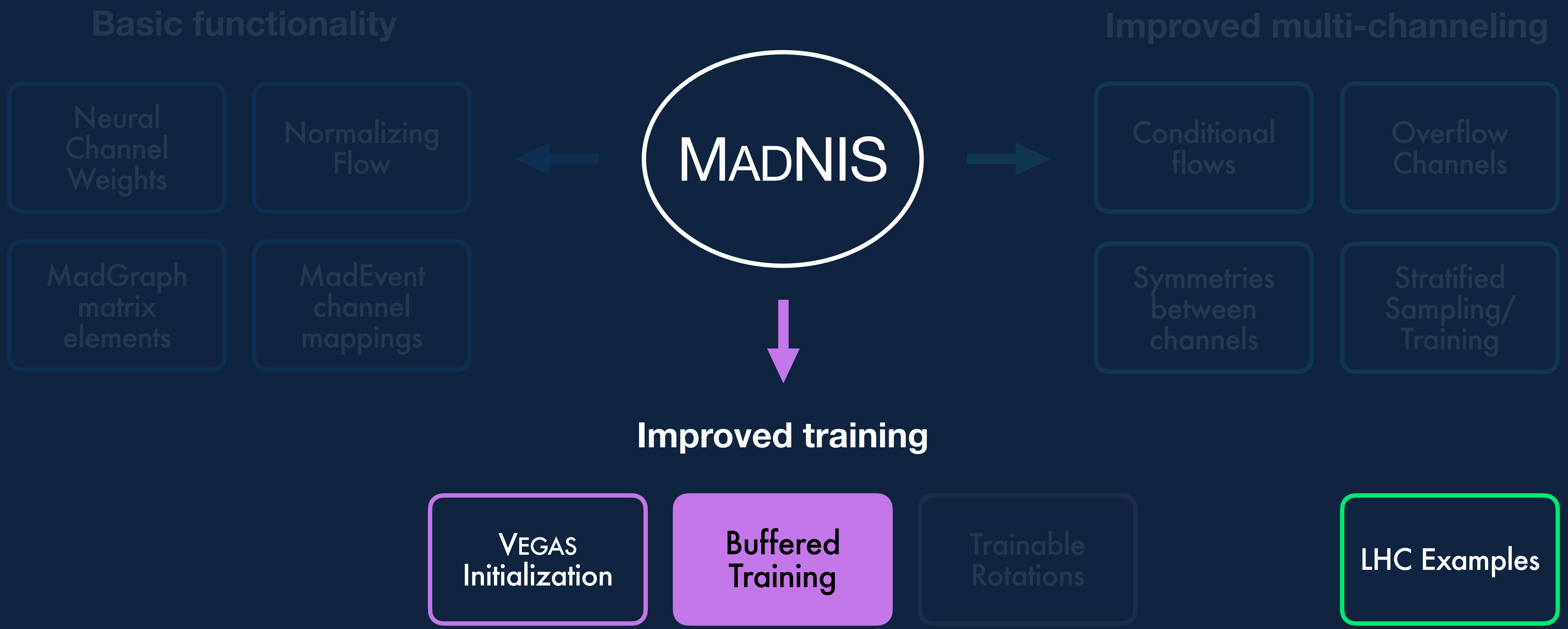
VEGAS Initialization

Buffered Training

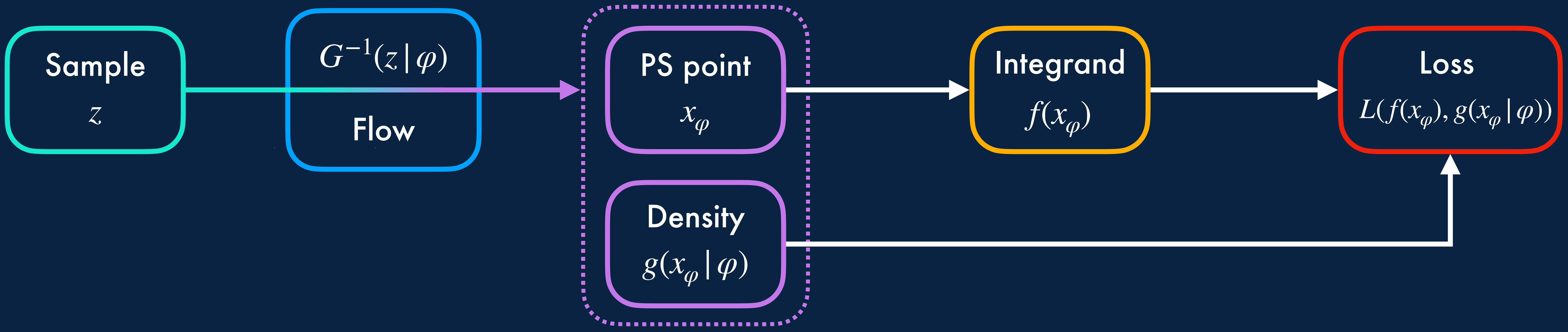
Trainable Rotations

LHC Examples

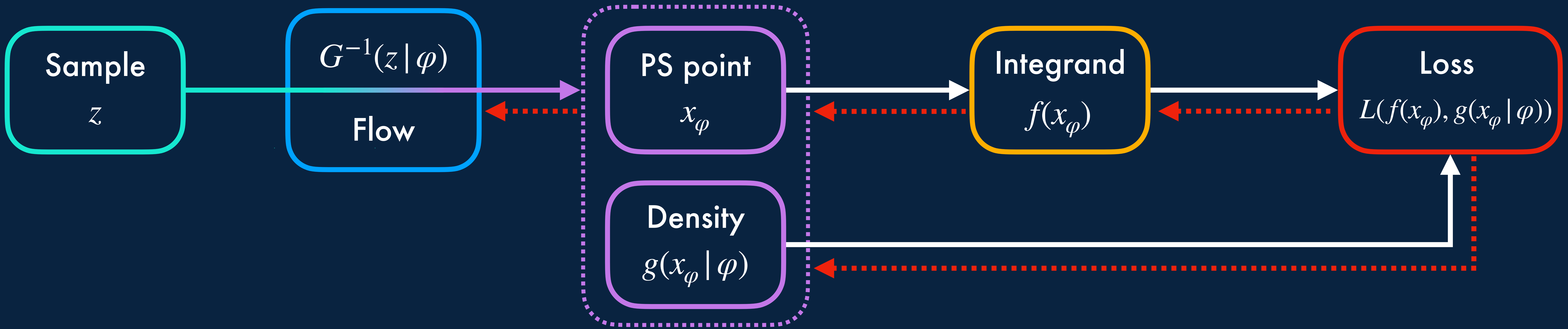
Buffered training

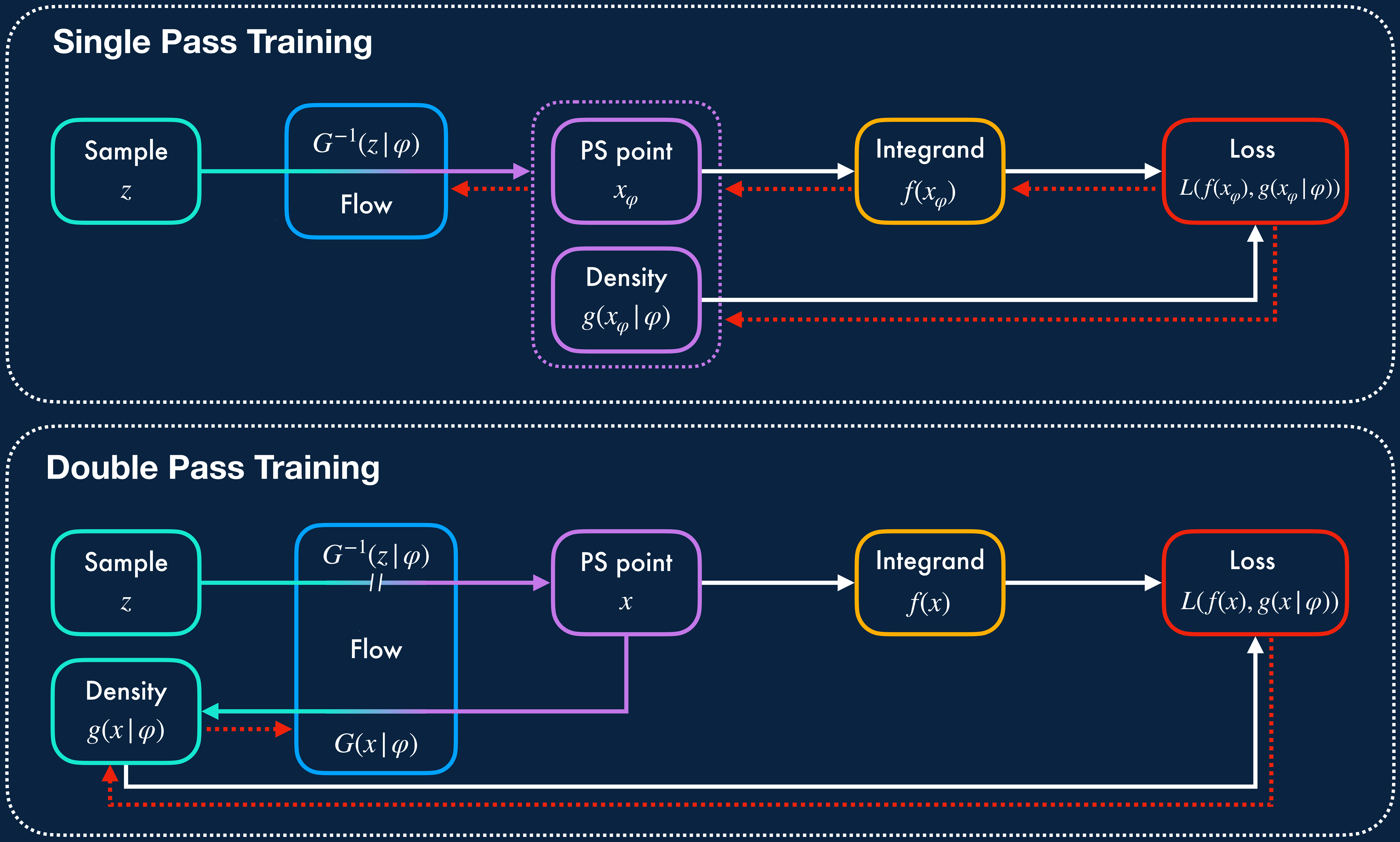


Single Pass Training



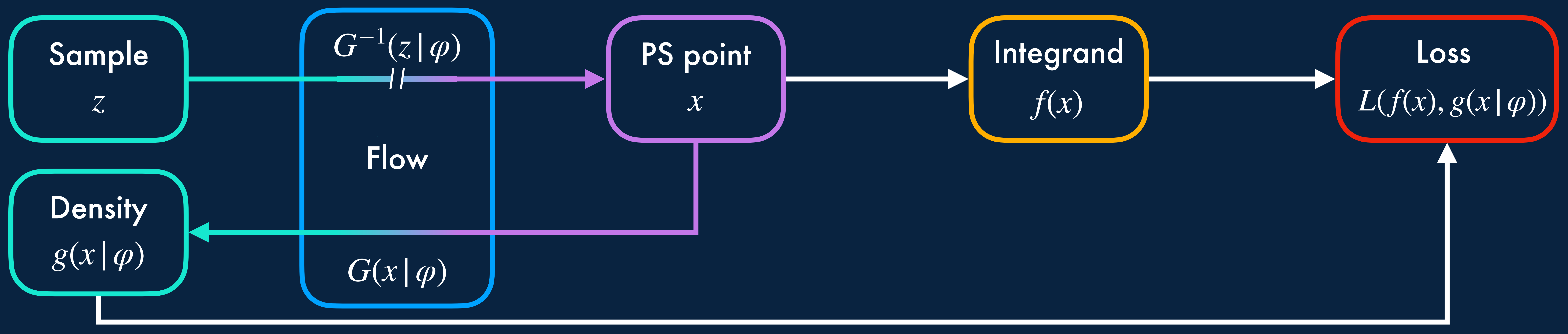
Single Pass Training

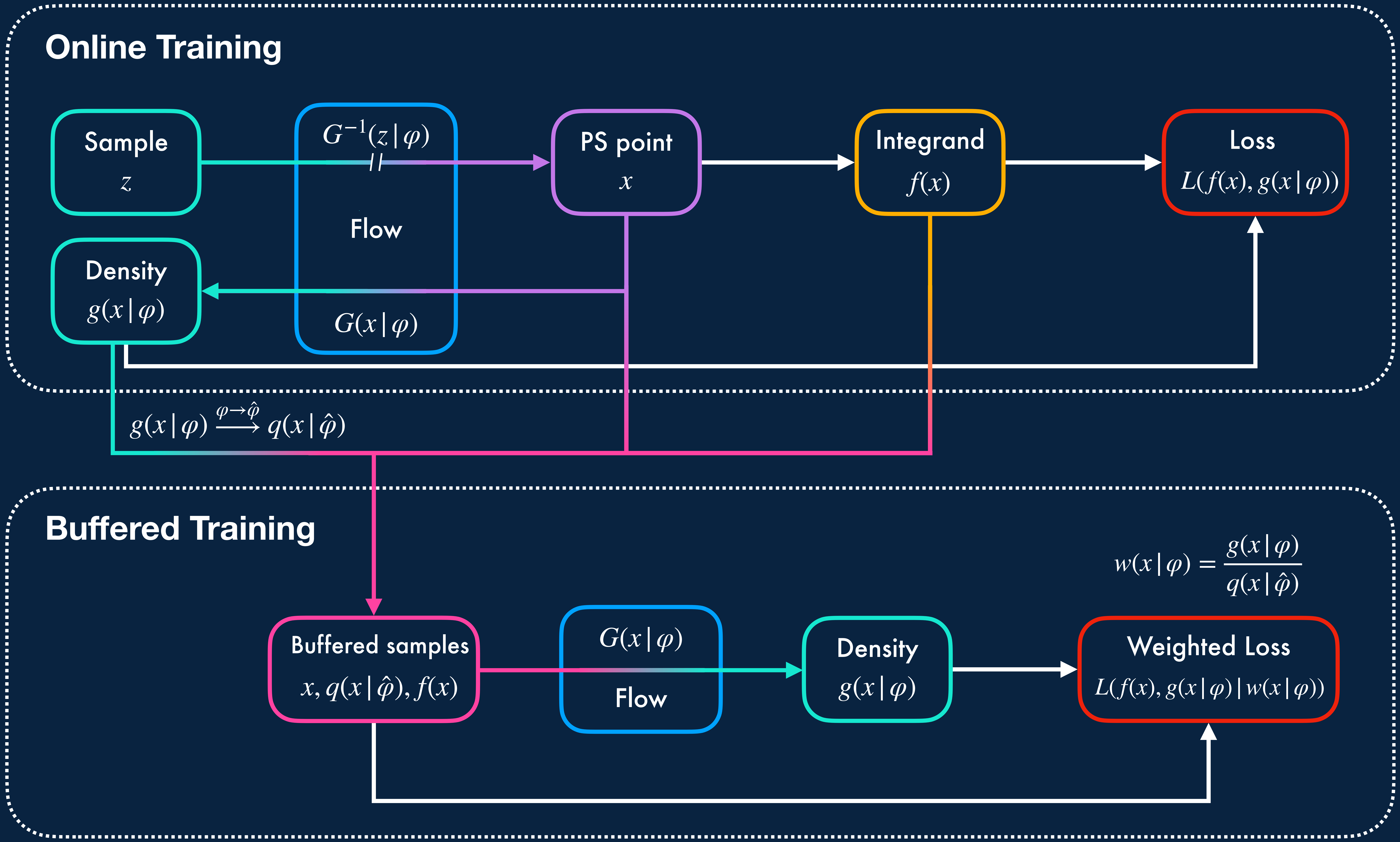




Single Pass Training
Sample
z
$G^{-1}(z|\varphi)$
Flow
PS point
x_φ
Density
$g(x_\varphi|\varphi)$
Integrand
$f(x_\varphi)$
Loss
$L(f(x_\varphi), g(x_\varphi|\varphi))$
Double Pass Training
Sample
z
Density
$g(x|\varphi)$
$G^{-1}(z|\varphi)$
Flow
$G(x|\varphi)$
PS point
x
Integrand
$f(x)$
Loss
$L(f(x), g(x|\varphi))$

Online Training





Online Training
Sample
z
$G^{-1}(z|\varphi)$
Flow
$G(x|\varphi)$
PS point
x
Integrand
$f(x)$
Loss
$L(f(x), g(x|\varphi))$
Density
$g(x|\varphi)$
$g(x|\varphi) \xrightarrow{\varphi \to \hat{\varphi}} q(x|\hat{\varphi})$
Buffered Training
$w(x|\varphi) = \frac{g(x|\varphi)}{q(x|\hat{\varphi})}$
Buffered samples
$x, q(x|\hat{\varphi}), f(x)$
$G(x|\varphi)$
Flow
Density
$g(x|\varphi)$
Weighted Loss
$L(f(x), g(x|\varphi)|w(x|\varphi))$

Buffered training

Training algorithm

generate new samples, train on them, save samples

↓

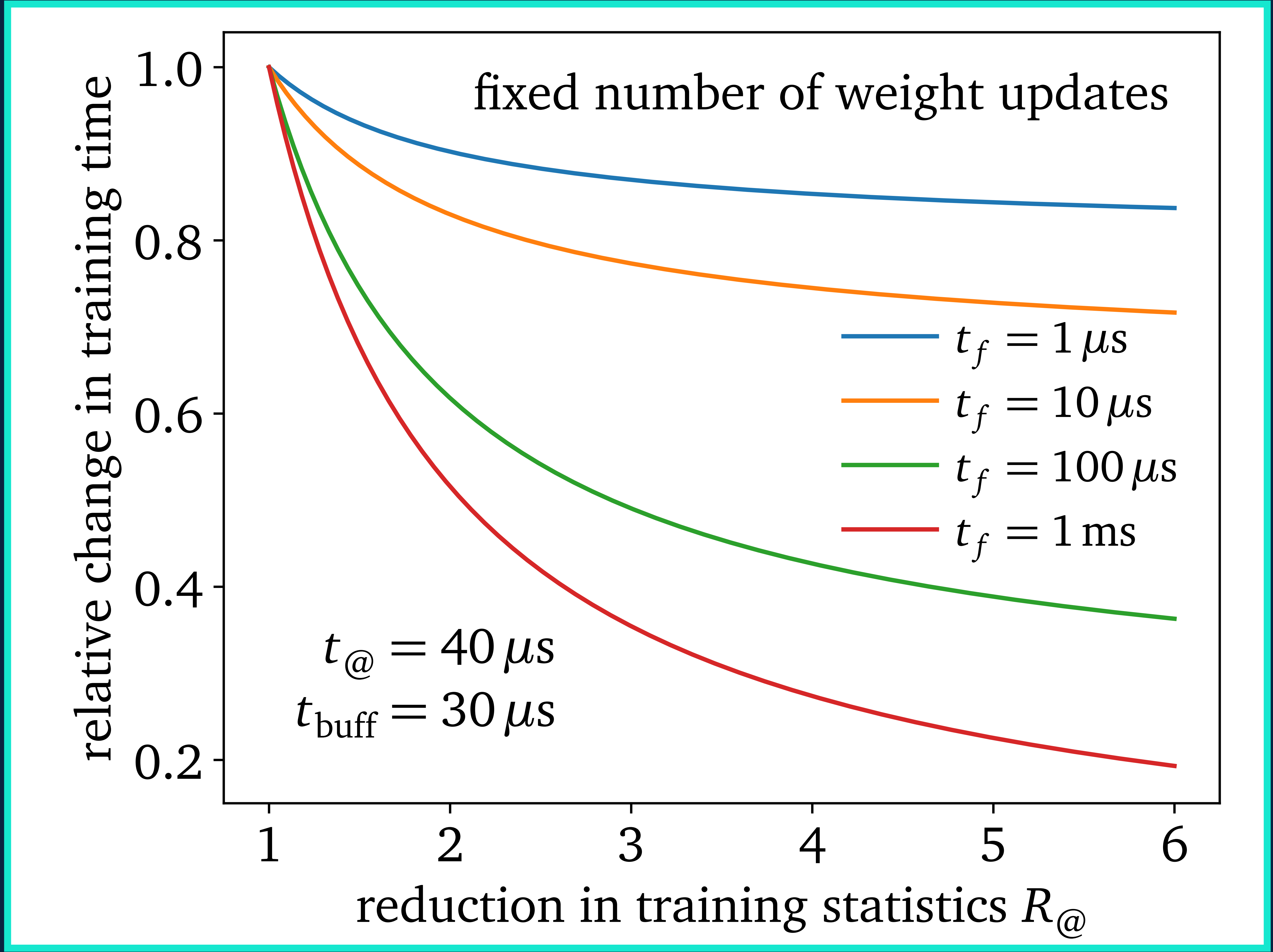
train on saved samples n times

↓

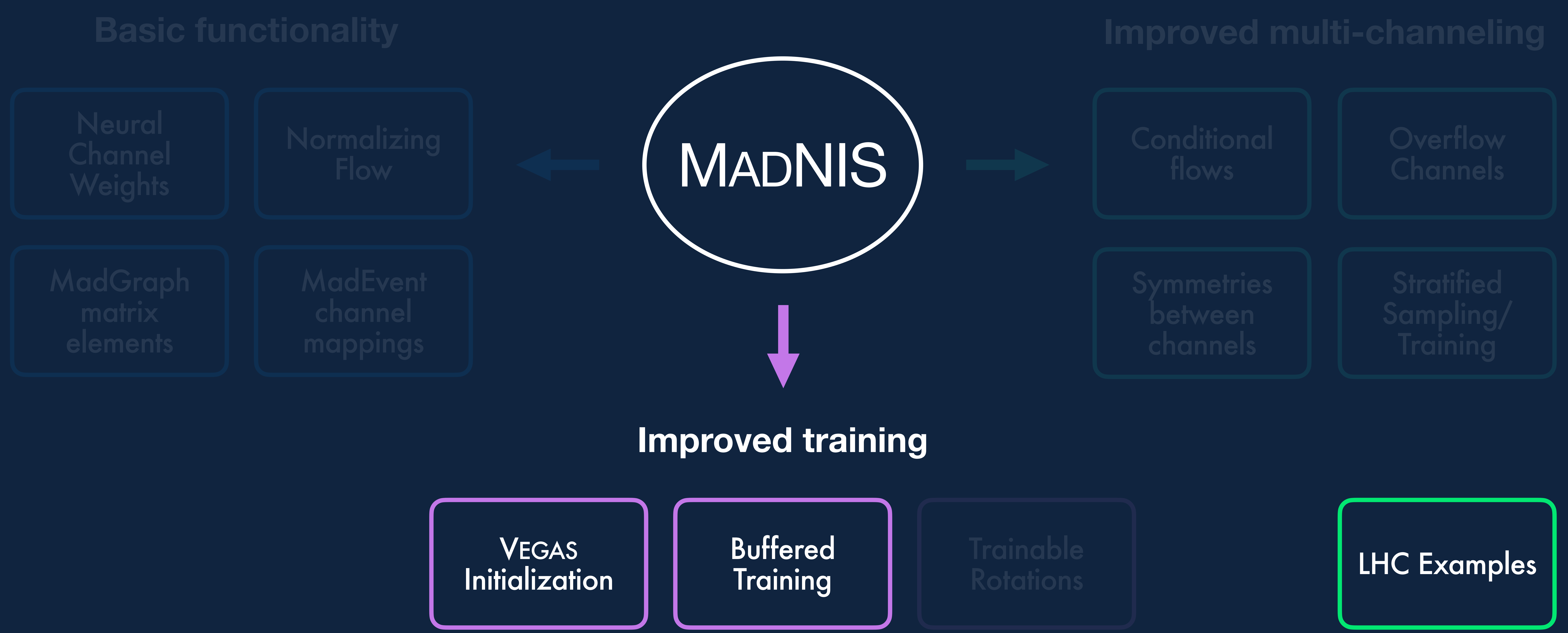
repeat

Reduction in training statistics by

$$R_@ = n + 1$$



MadNIS — Overview



LHC examples

Basic functionality

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

MADNIS

Improved multi-channeling

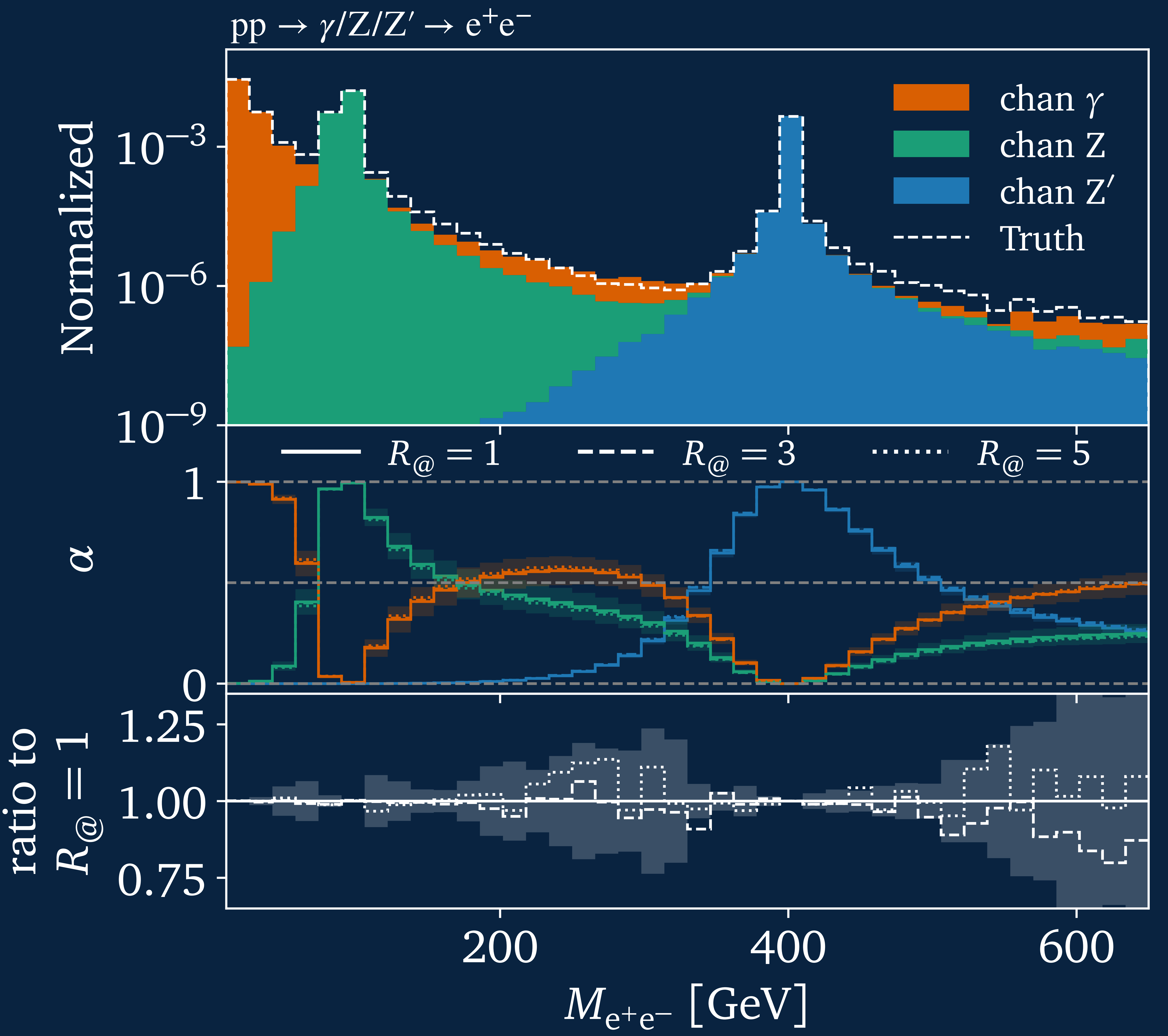
- Conditional flows
- Overflow Channels
- Symmetries between channels
- Stratified Sampling/ Training

Improved training

- VEGAS Initialization
- Buffered Training
- Trainable Rotations

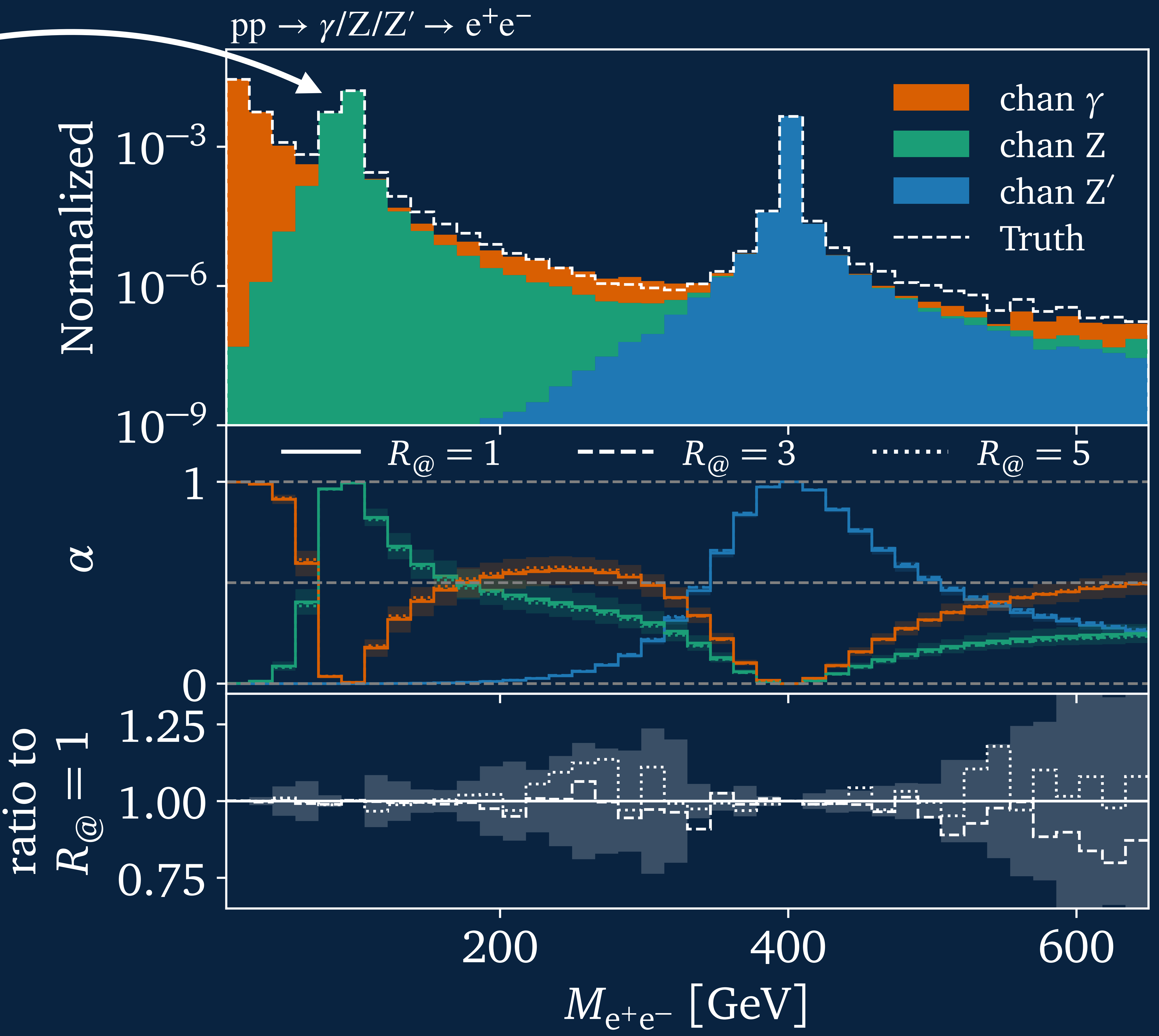
LHC Examples

LHC example I — Drell-Yan



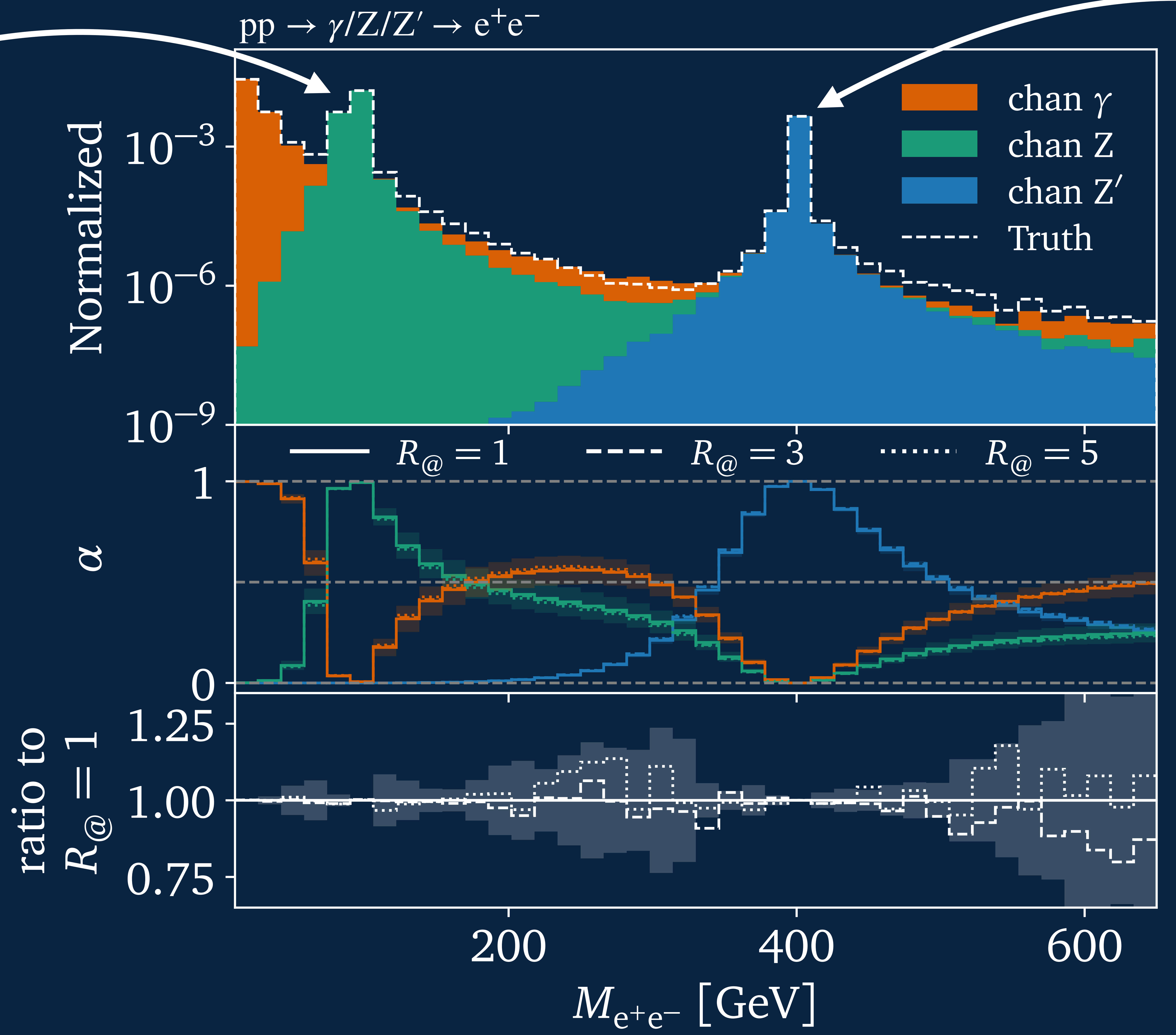
LHC example I — Drell-Yan

Learned distribution matches truth



LHC example I — Drell-Yan

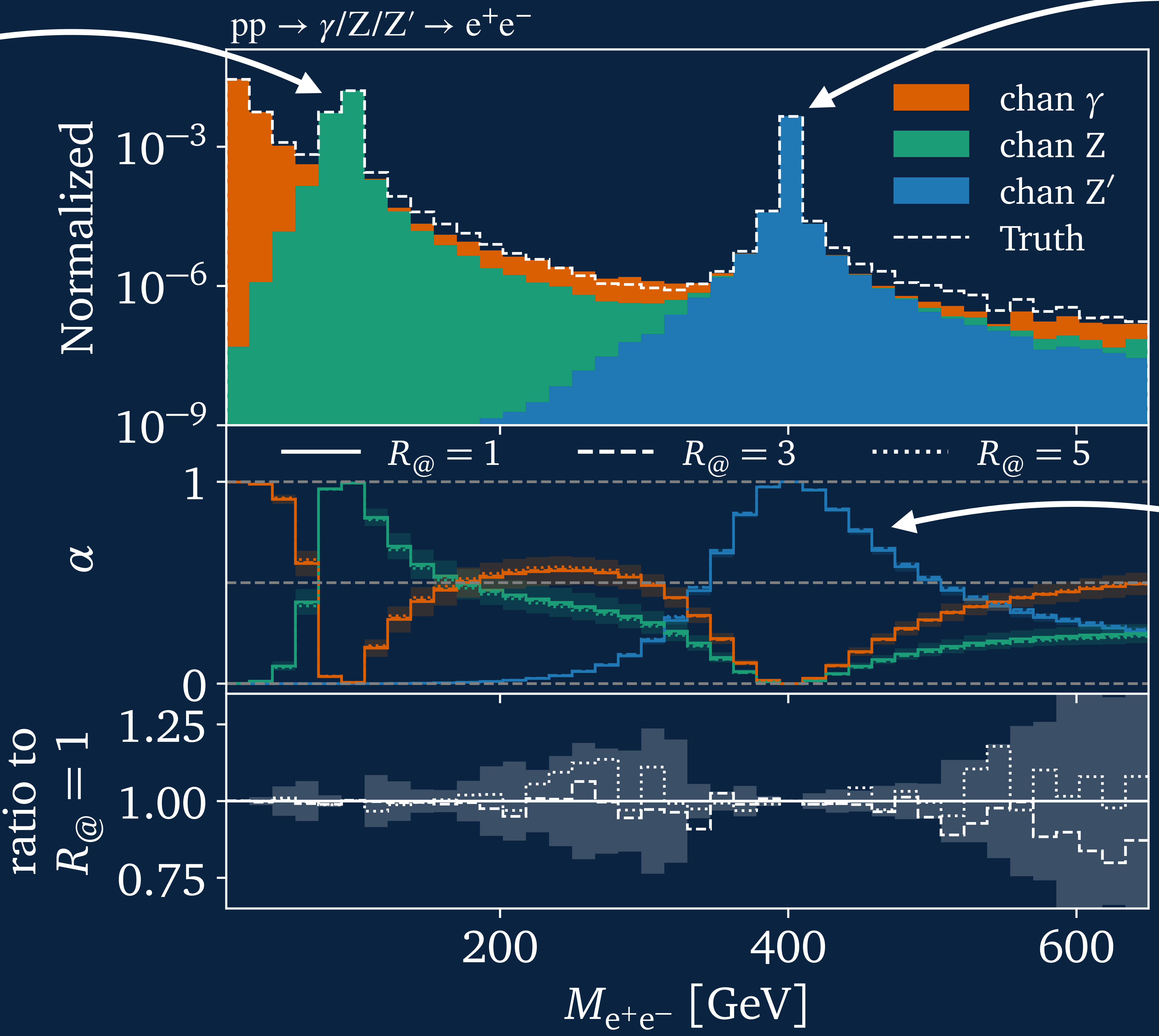
Learned distribution matches truth



Peaks mapped out by different channels

LHC example I — Drell-Yan

Learned distribution matches truth



Peaks mapped out by different channels

Channel weights learned by network

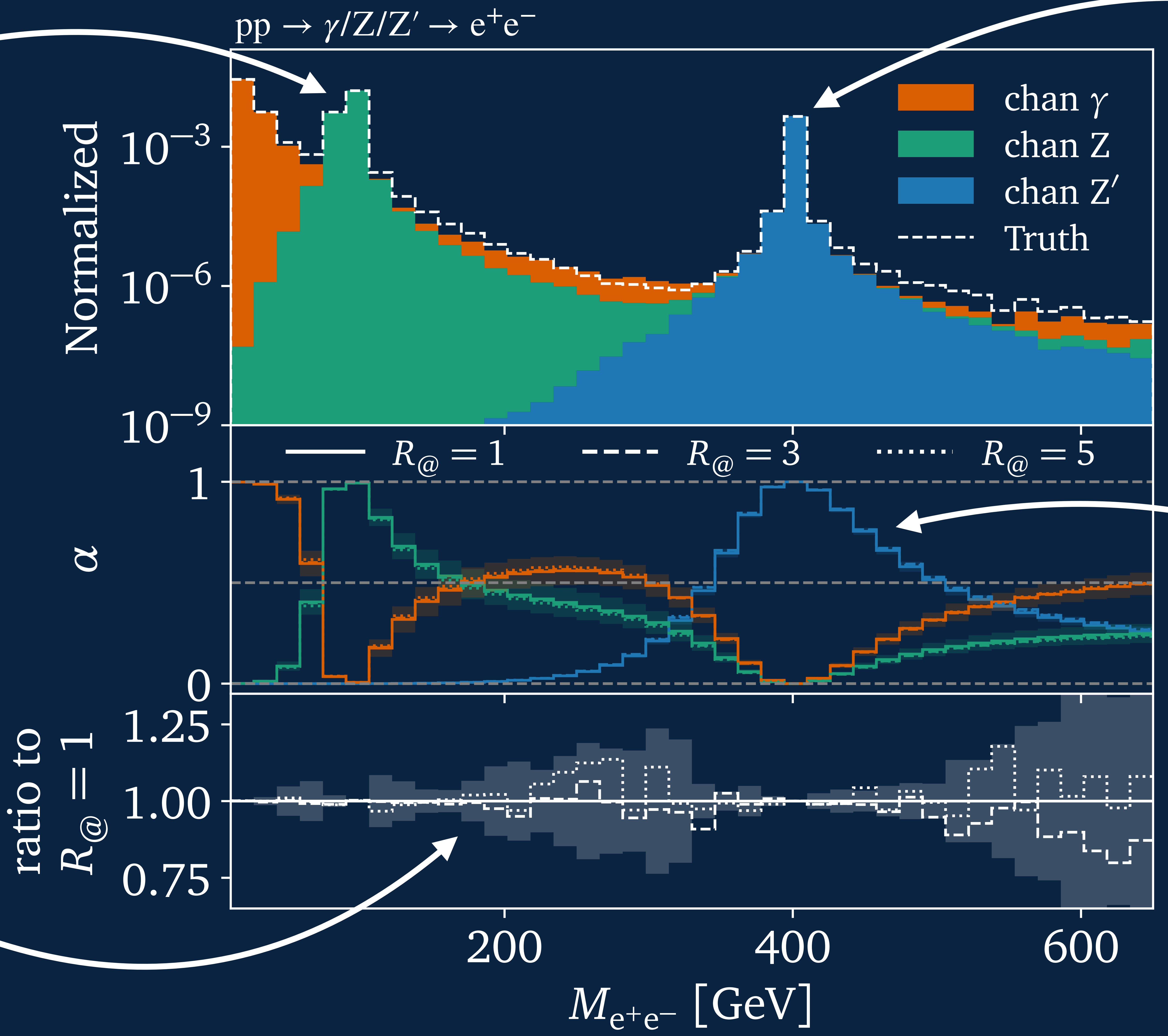
LHC example I — Drell-Yan

Learned distribution matches truth

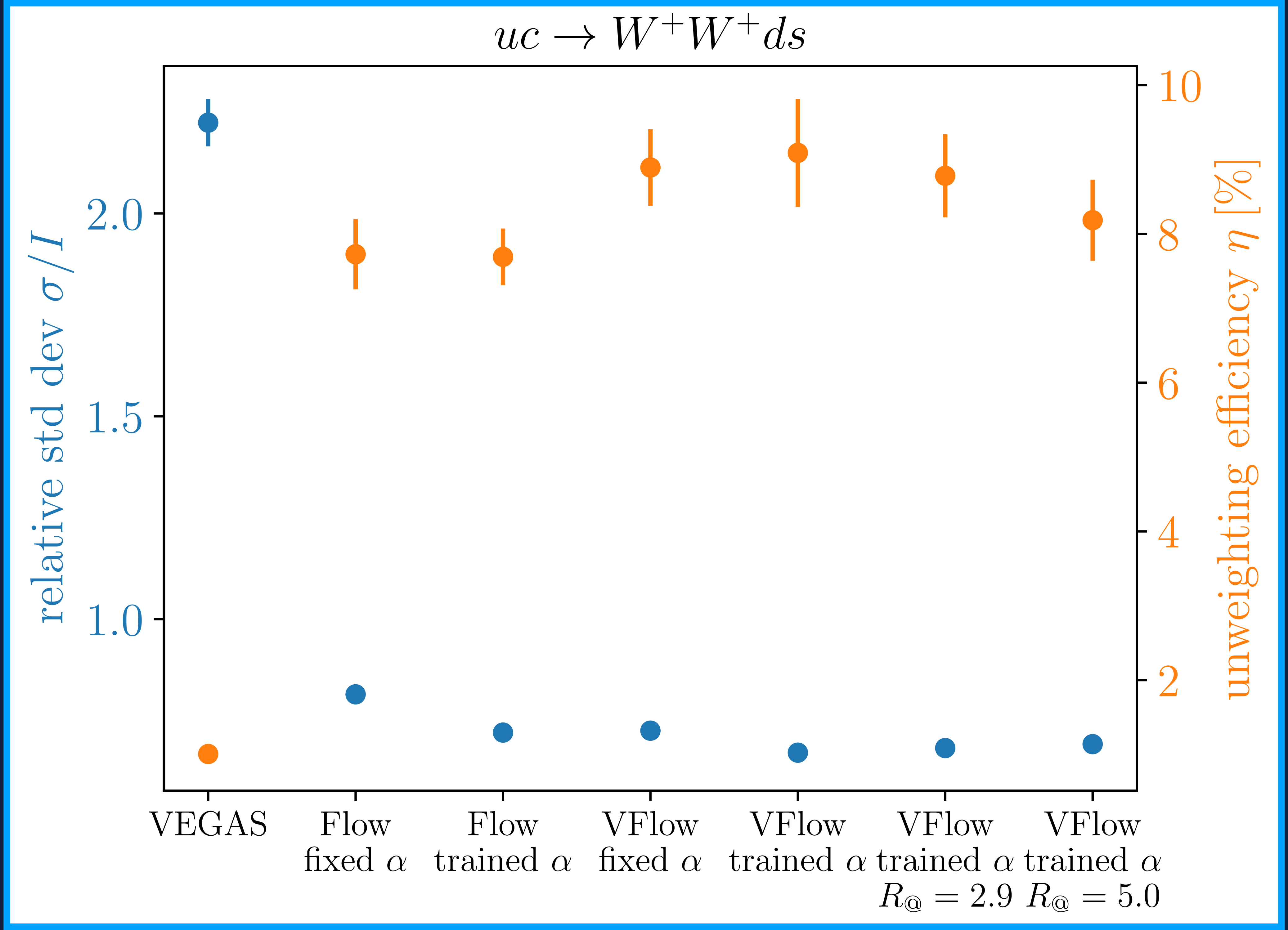
Peaks mapped out by different channels

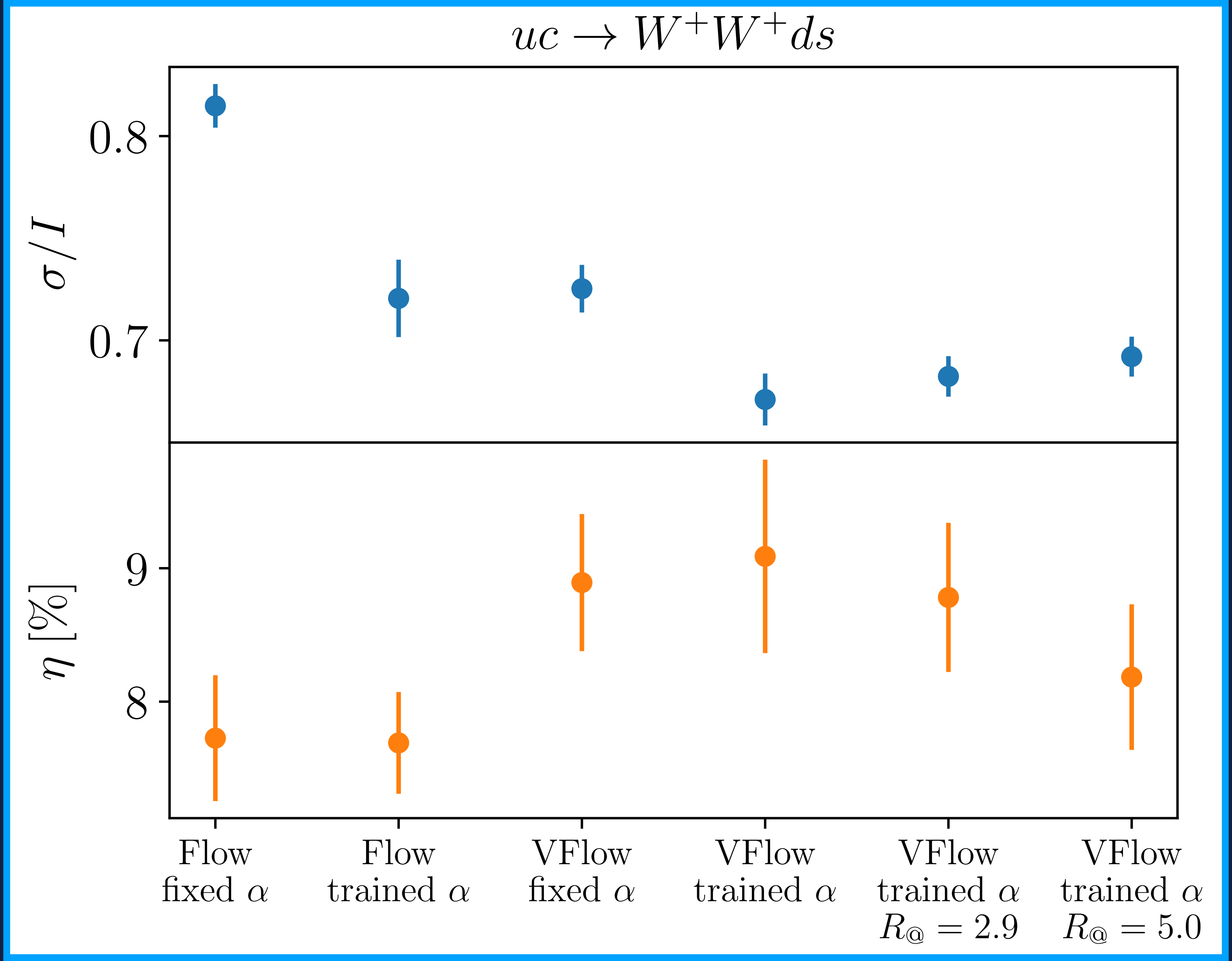
Channel weights learned by network

Use samples multiple times to make training faster



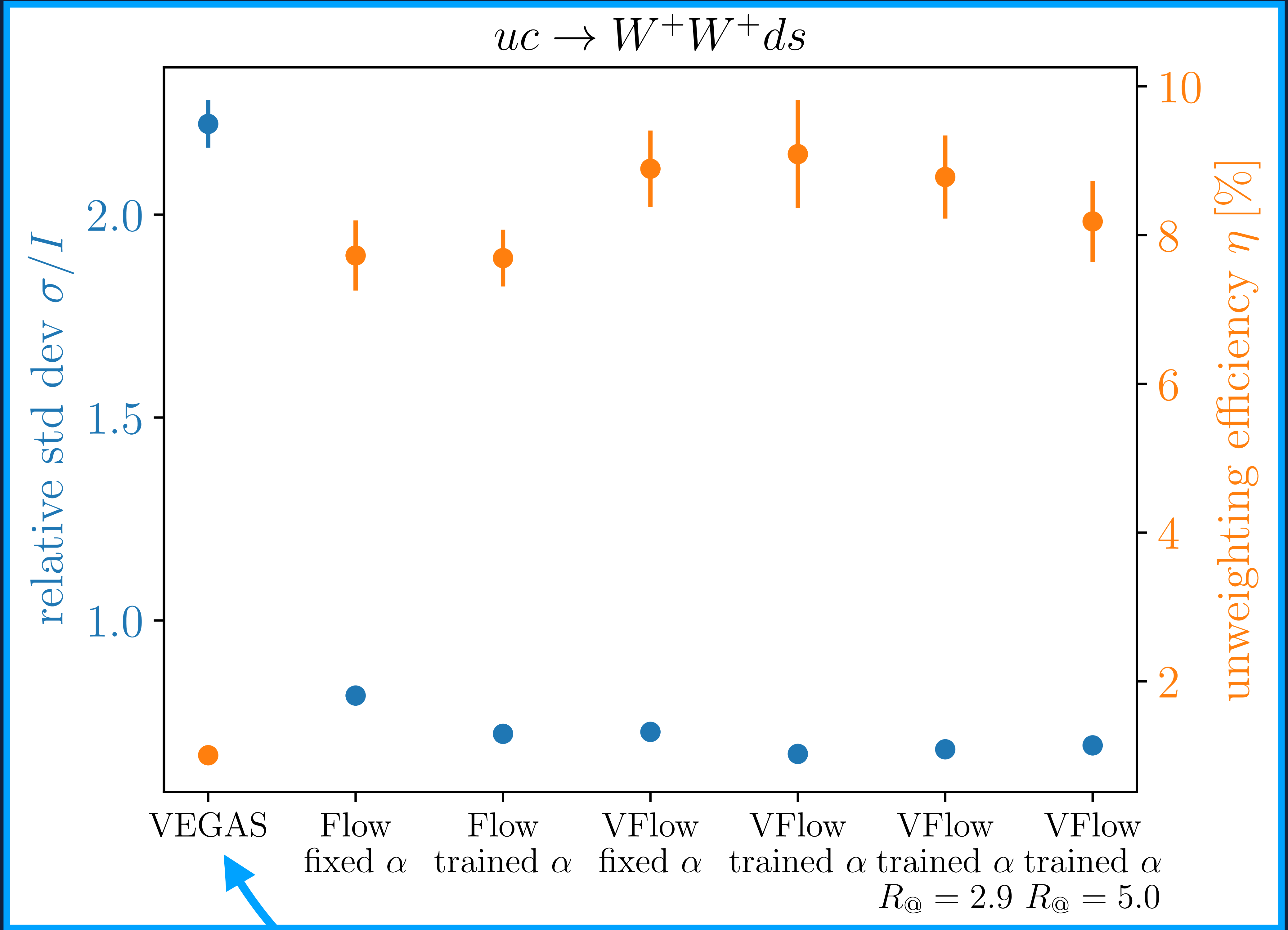
LHC example II — VBS

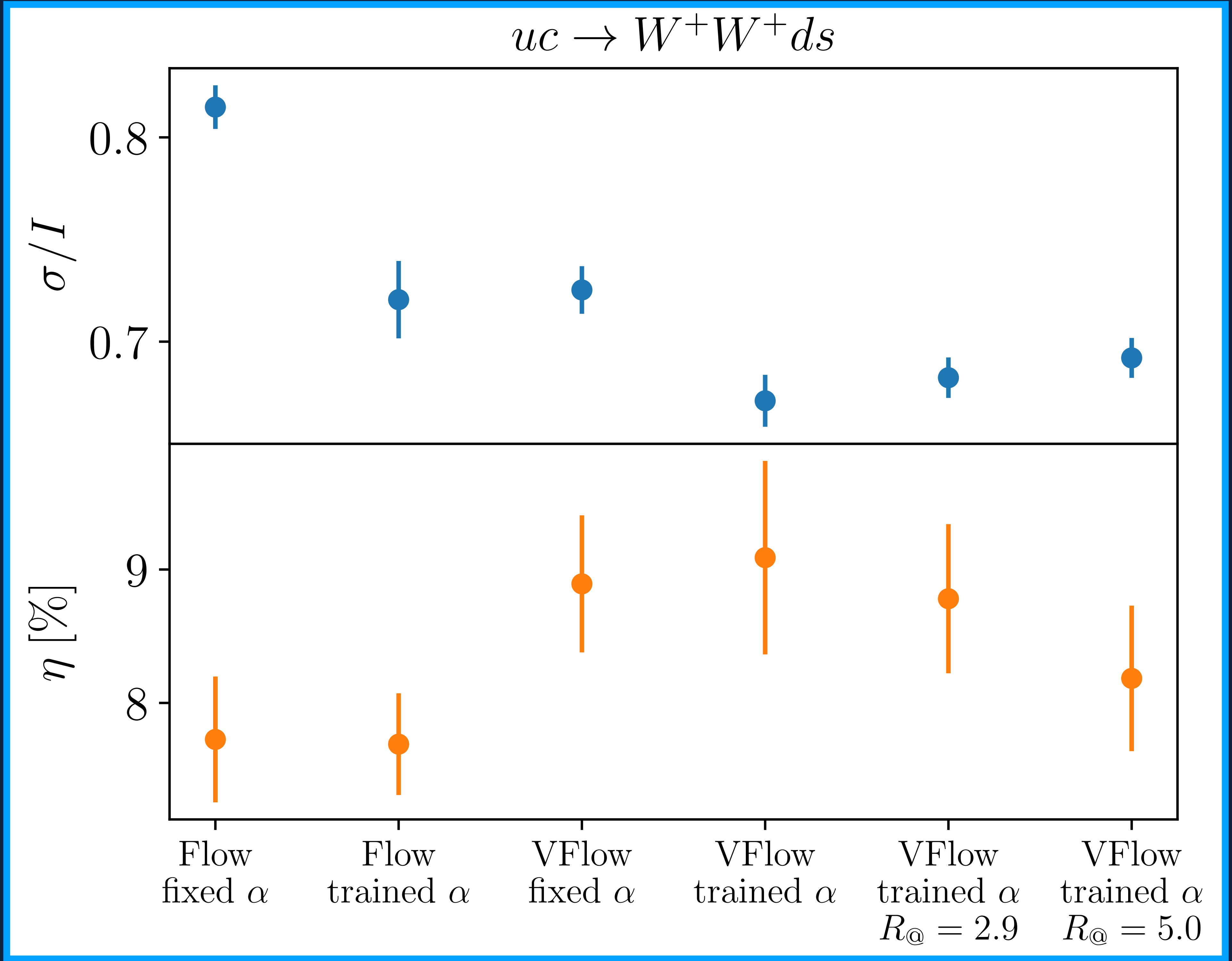




(preliminary)

LHC example II — VBS

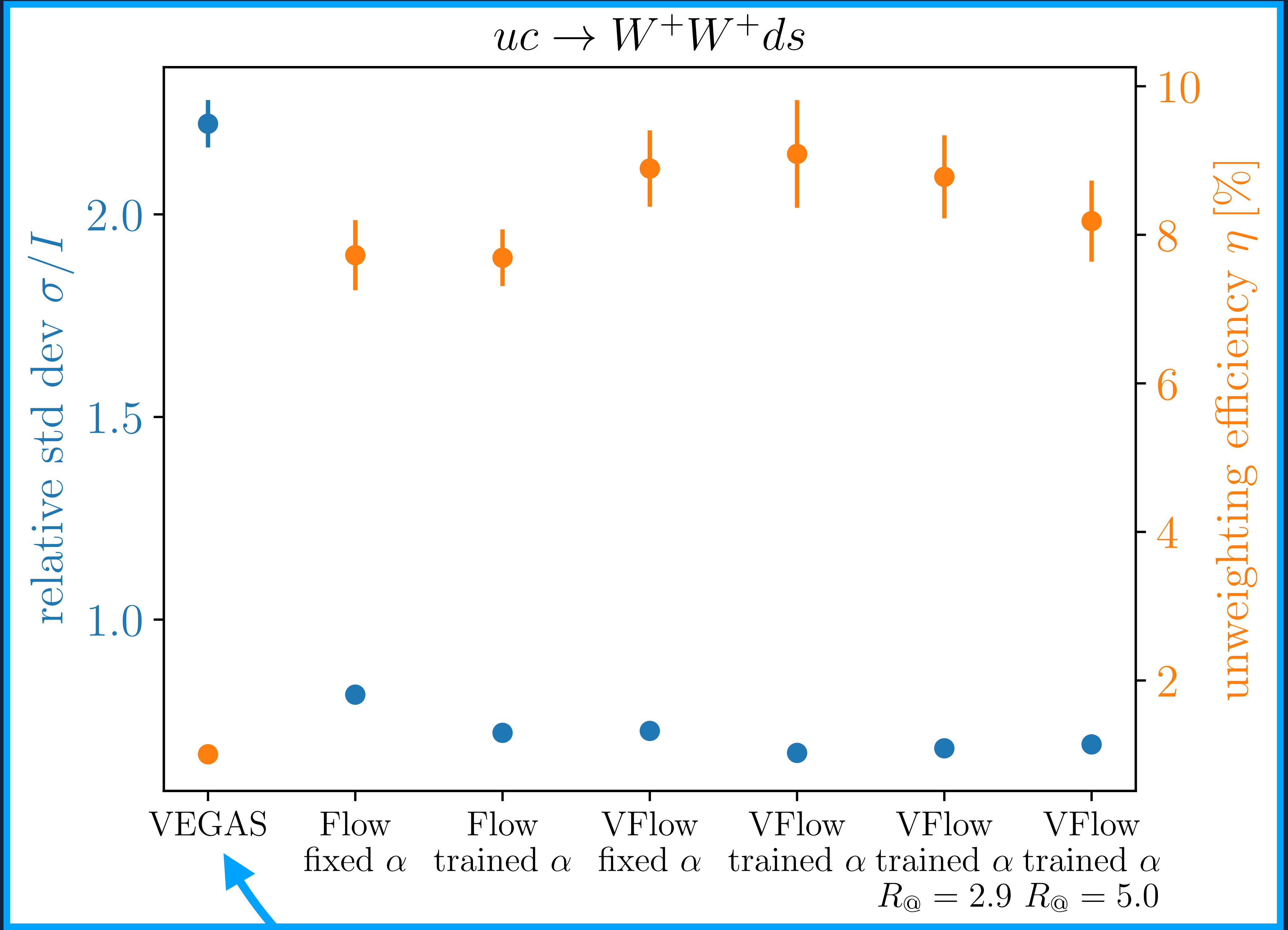


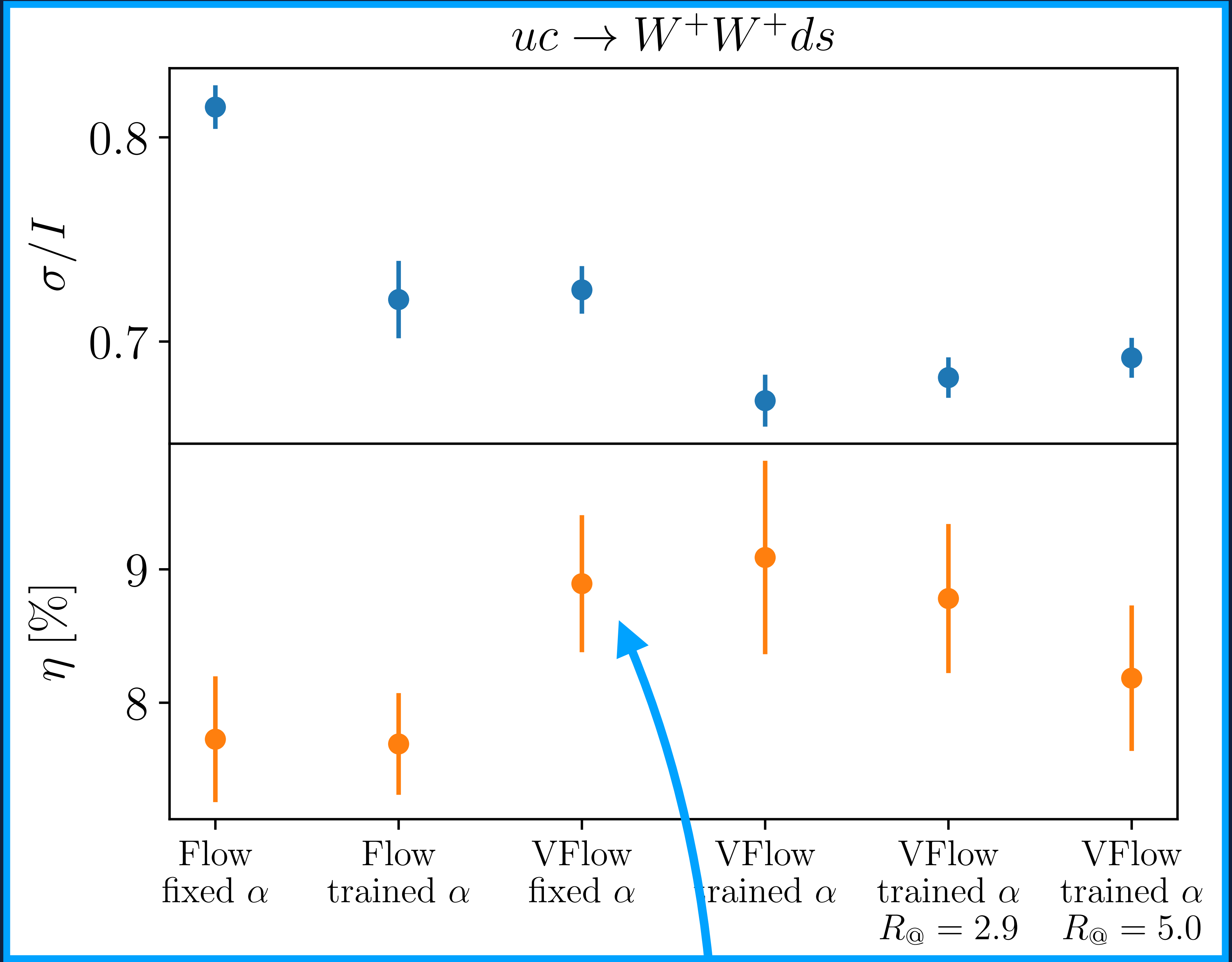


(preliminary)

Unweighting efficiency improved up to factor ~9 compared to VEGAS

LHC example II — VBS





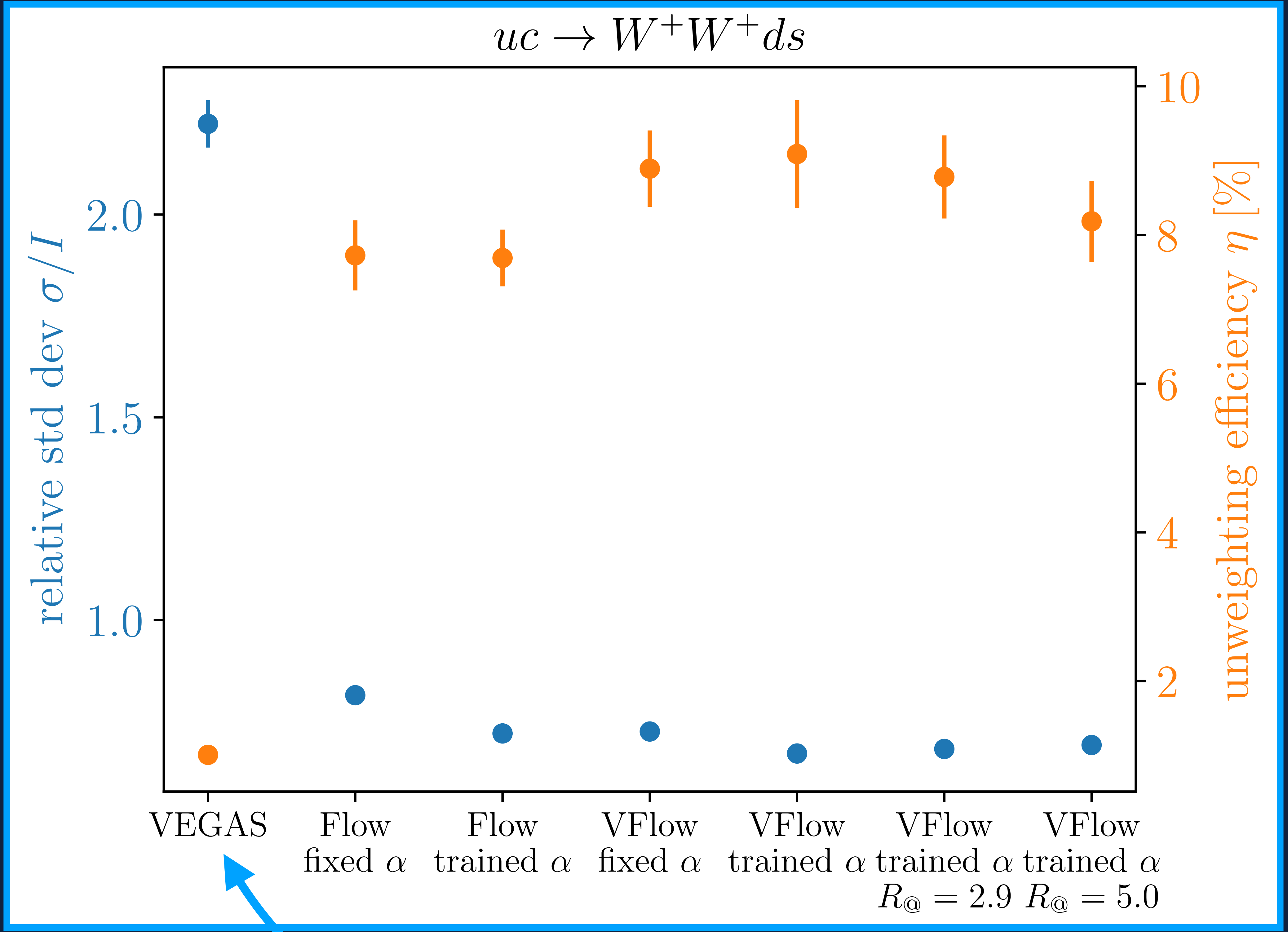
(preliminary)

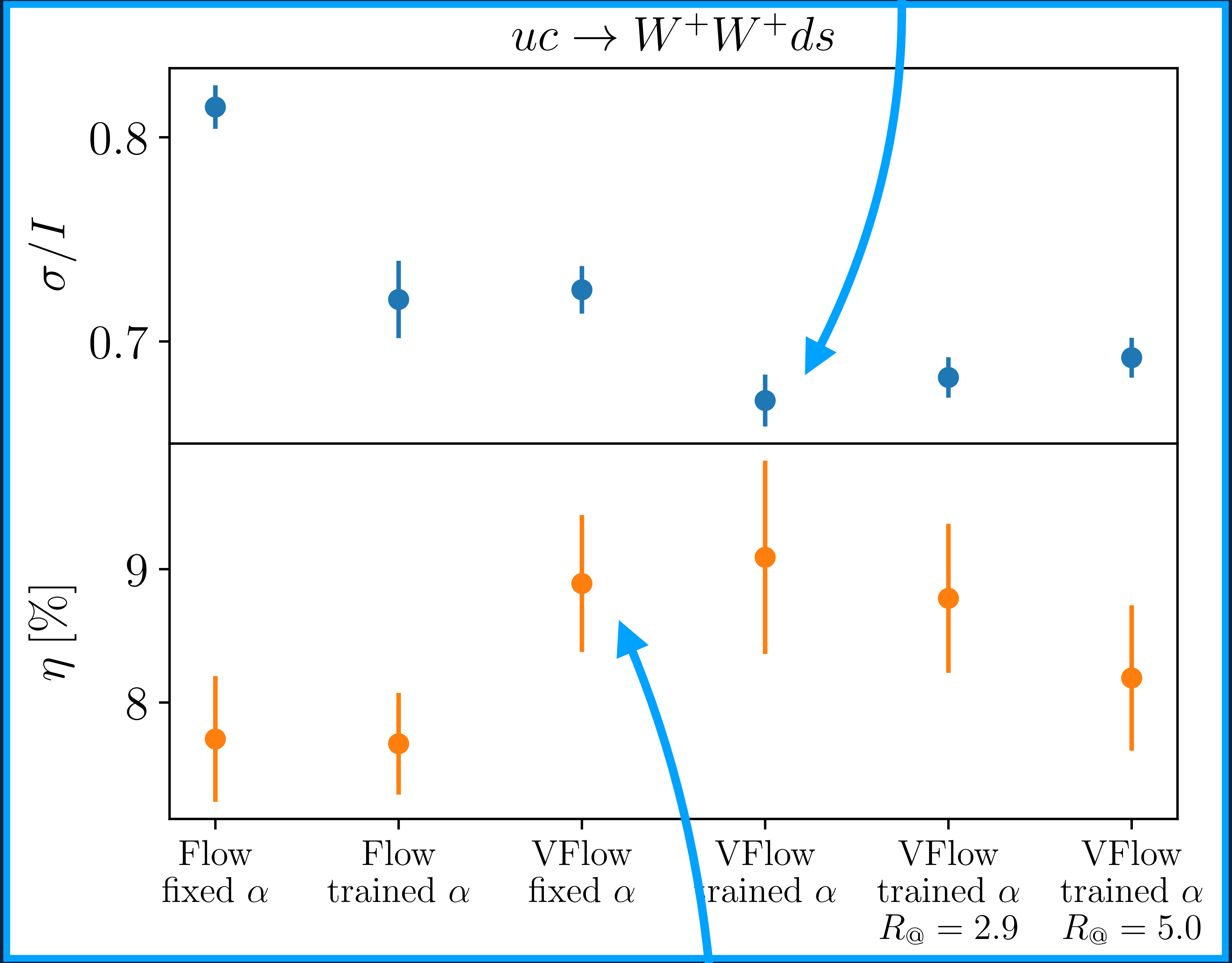
Unweighting efficiency improved up to factor ~9 compared to VEGAS

Big improvement from VEGAS initialization

LHC example II — VBS

Significant improvement from trained channel weights





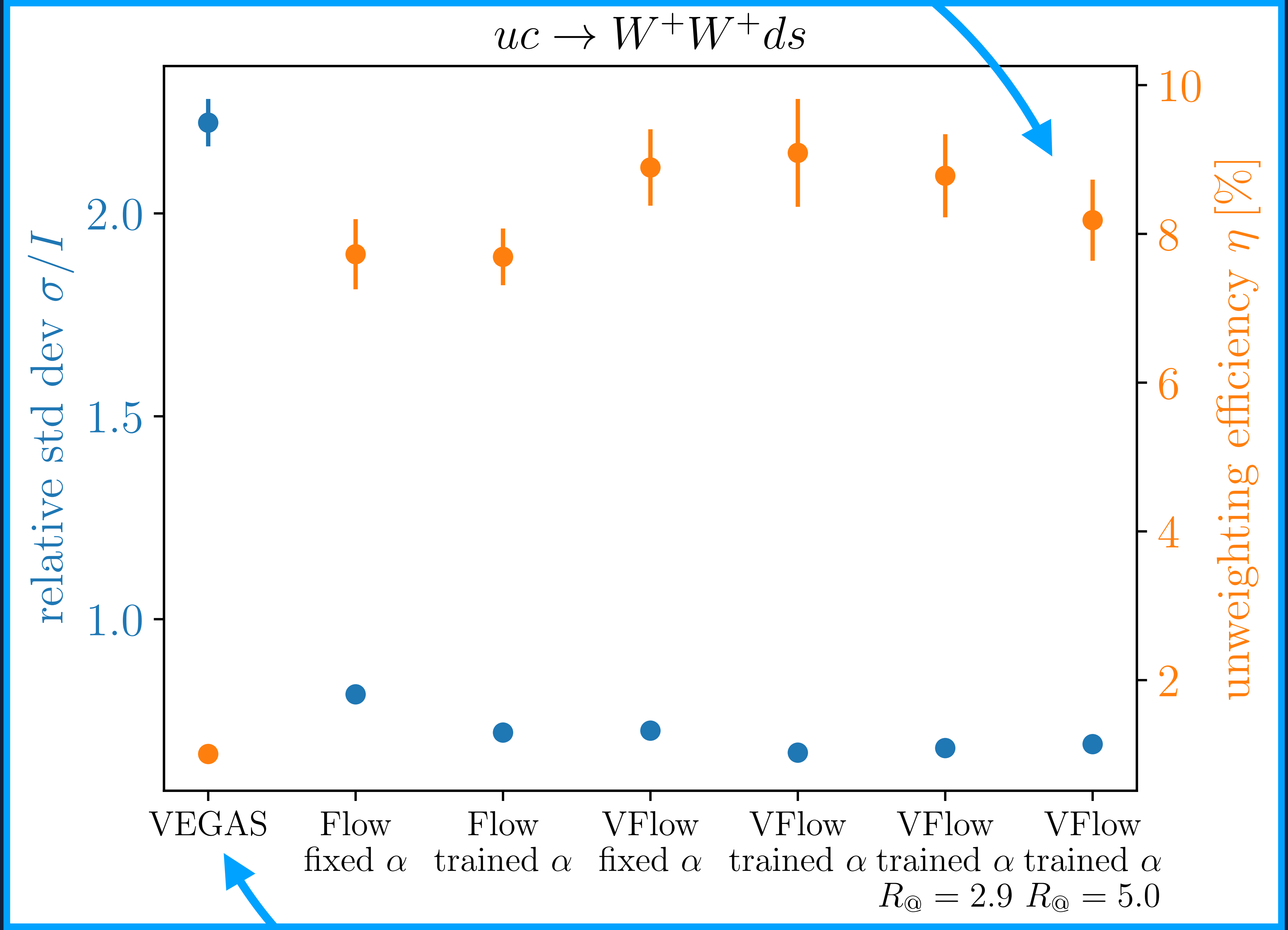
(preliminary)

Unweighting efficiency improved up to factor ~9 compared to VEGAS

Big improvement from VEGAS initialization

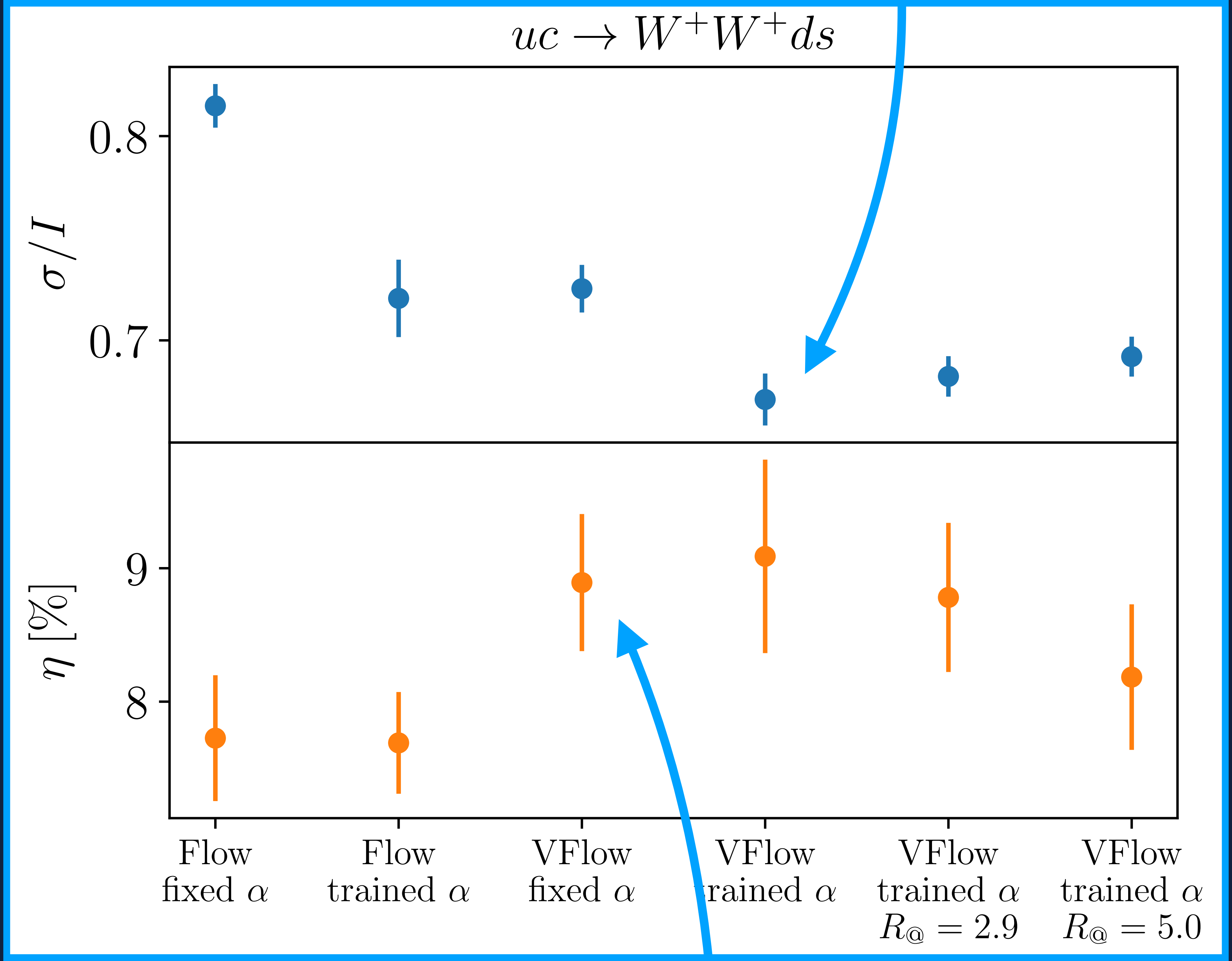
LHC example II — VBS

Buffered training: small effect on performance, much faster training



Unweighting efficiency improved up to factor ~9 compared to VEGAS

Significant improvement from trained channel weights

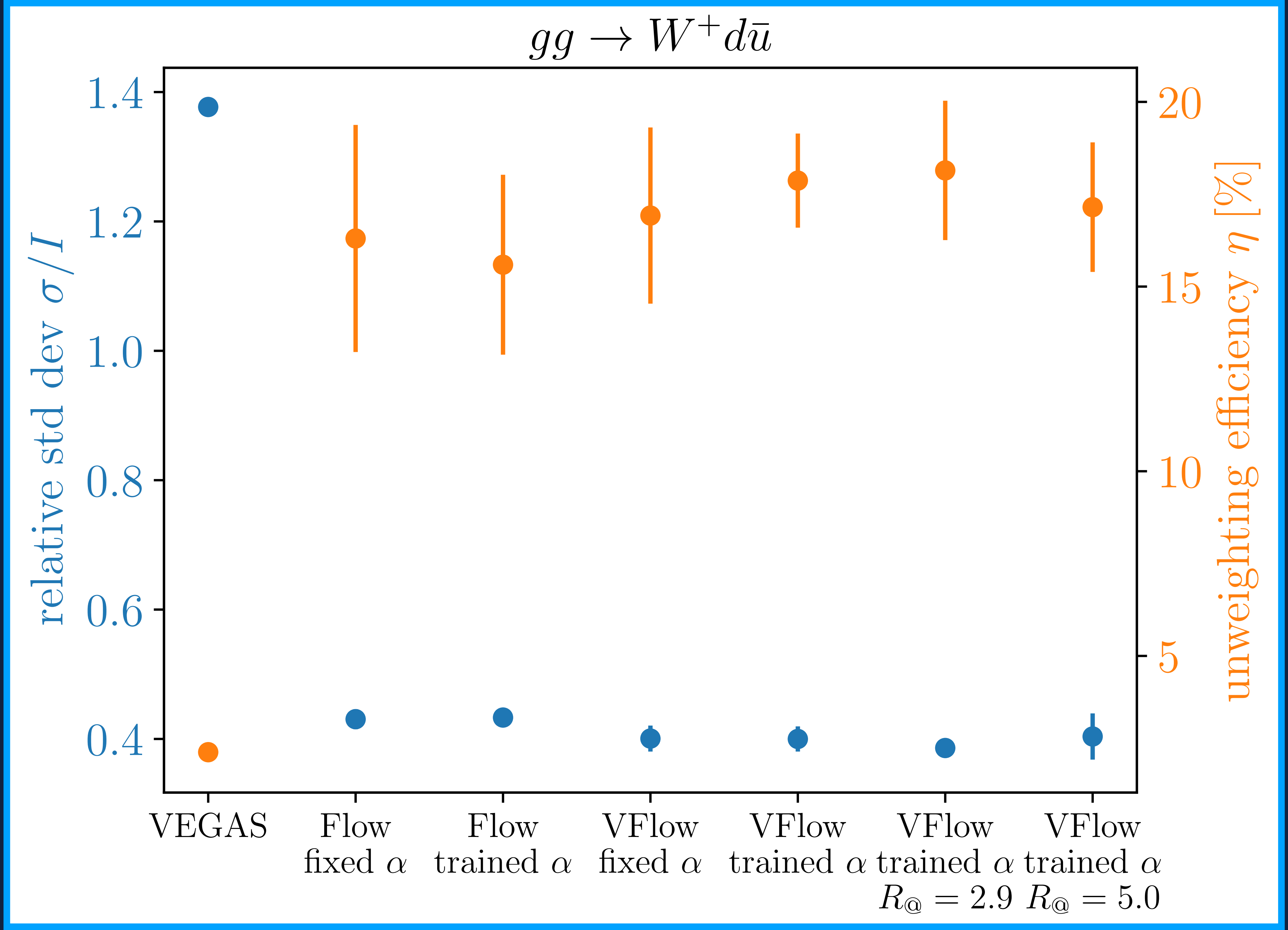


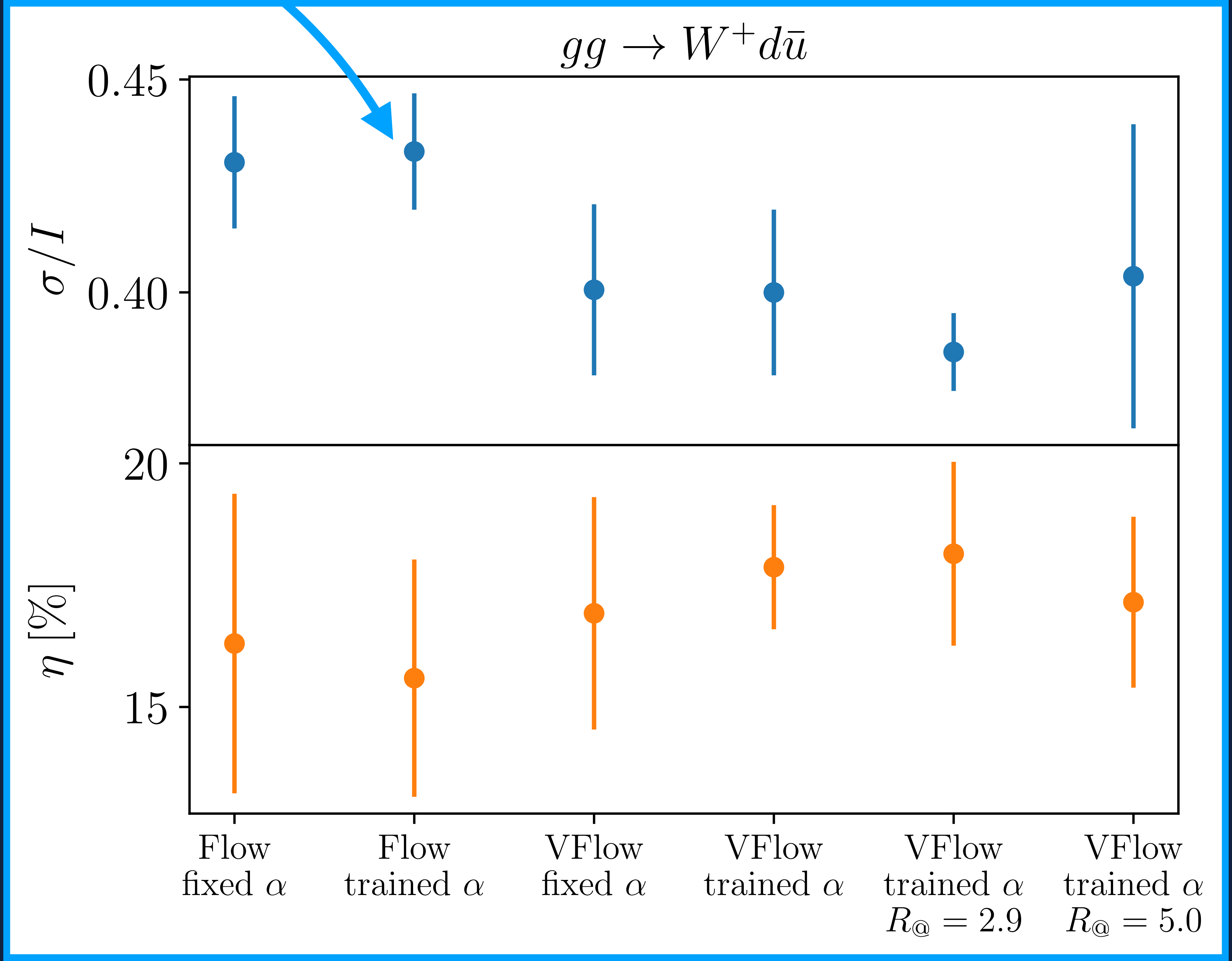
Big improvement from VEGAS initialization

(preliminary)

LHC example III — 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

Summary

- MadNIS **outperforms** current sampling methods
- Multi-channel is **more efficient** when **trained simultanously** with the flow
- Vegas initialization **improves performance**

Outlook

- Full integration of **MadNIS** into **MadGraph**
- Test performance on **real LHC examples:** (eg. multi-leg, NLO, complicated cuts, …)
- Make everything run on the **GPU and differentiable** [MadJax 2203.00057]

MadNIS

Summary and outlook

HEPML-LivingReview

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

The purpose of this note is to collect references for modern machine learning as applied to particle physics. A minimal number of categories is chosen in order to be as useful as possible. Note that papers may be referenced in more than one category. The fact that a paper is listed in this document does not endorse or validate its content - that is for the community (and for peer-review) to decide. Furthermore, the classification here is a best attempt and may have flaws - please let us know if (a) we have missed a paper you think should be included, (b) a paper has been misclassified, or (c) a citation for a paper is not correct or if the journal information is now available. In order to be as useful as possible, this document will continue to evolve so please check back before you write your next paper. If you find this review helpful, please consider citing it using \cite{hepmllivingreview} in HEPML.bib.

- Reviews
  - Modern reviews
    - Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning [DOI]
    - Deep Learning and its Application to LHC Physics [DOI]
    - Machine Learning in High Energy Physics Community White Paper [DOI]
    - Machine learning at the energy and intensity frontiers of particle physics
    - Machine learning and the physical sciences [DOI]
    - Machine and Deep Learning Applications in Particle Physics [DOI]
    - Modern Machine Learning and Particle Physics

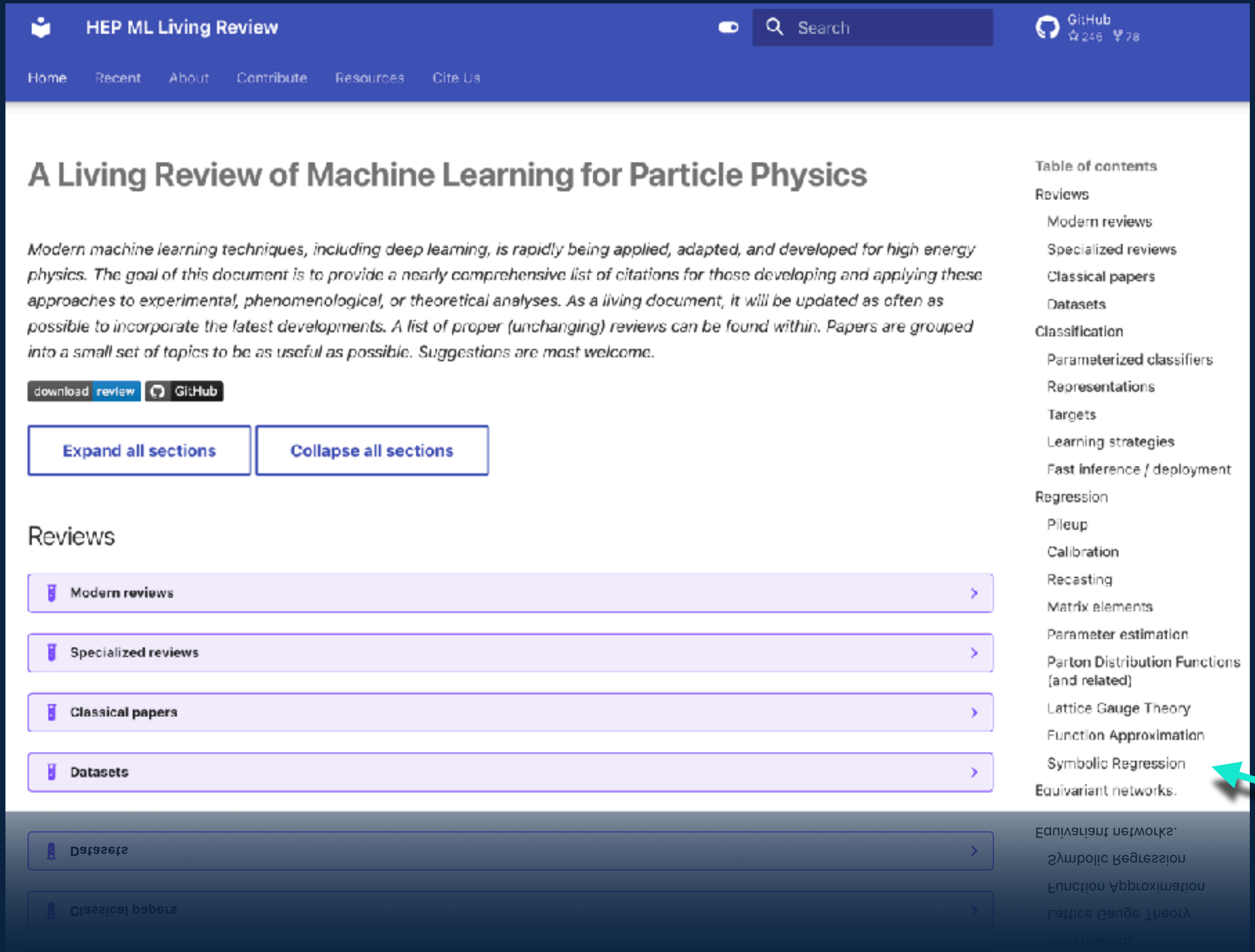
OUTDATED

Outlook

- Full integration of **MadNIS** into **MadGraph**
- Test performance on **real LHC examples:** (eg. multi-leg, NLO, complicated cuts, …)
- Make everything run on the **GPU and differentiable** [MadJax 2203.00057]
- Stay tuned for many other **ML4HEP applications**

HEPML

Summary and outlook



Got a facelift recently!

Outlook

- Full integration of **MadNIS** into **MadGraph**
- Test performance on **real LHC examples:** (eg. multi-leg, NLO, complicated cuts, …)
- Make everything run on the **GPU and differentiable** [MadJax 2203.00057]
- Stay tuned for many other **ML4HEP applications**

HEPML