UCLouvain

MadNIS

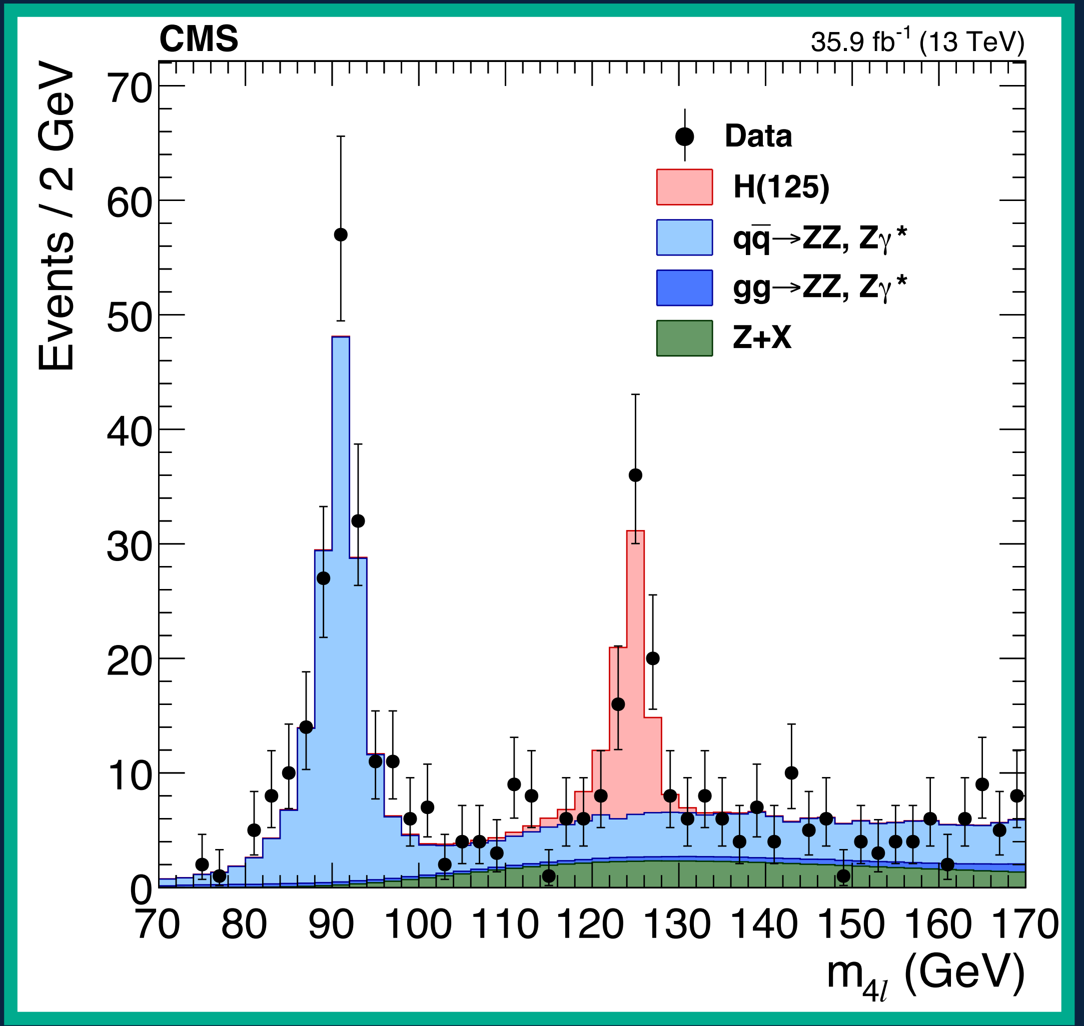
Neural networks for multi-channel integration in MadGraph

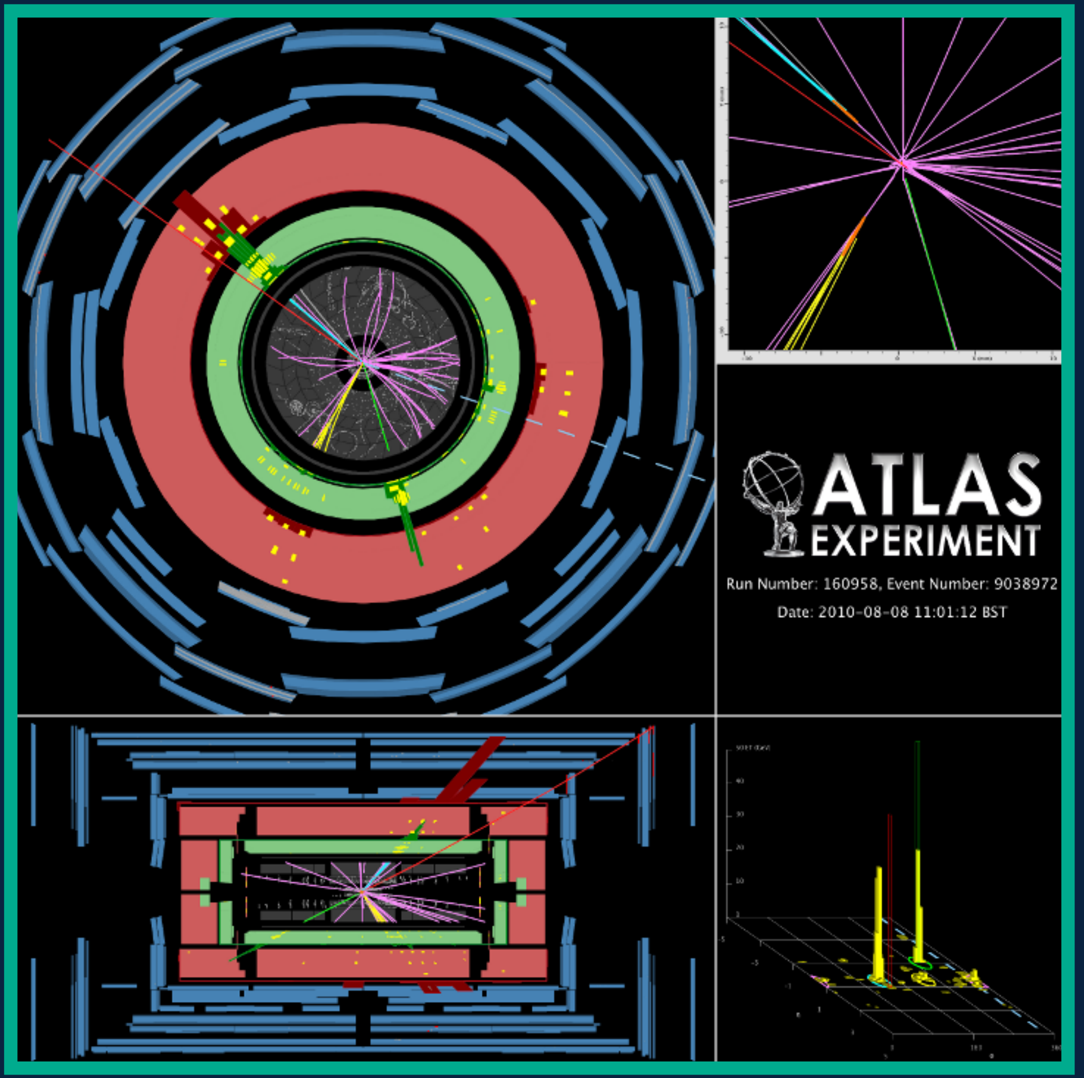
ML4Jets - Rutgers 2022

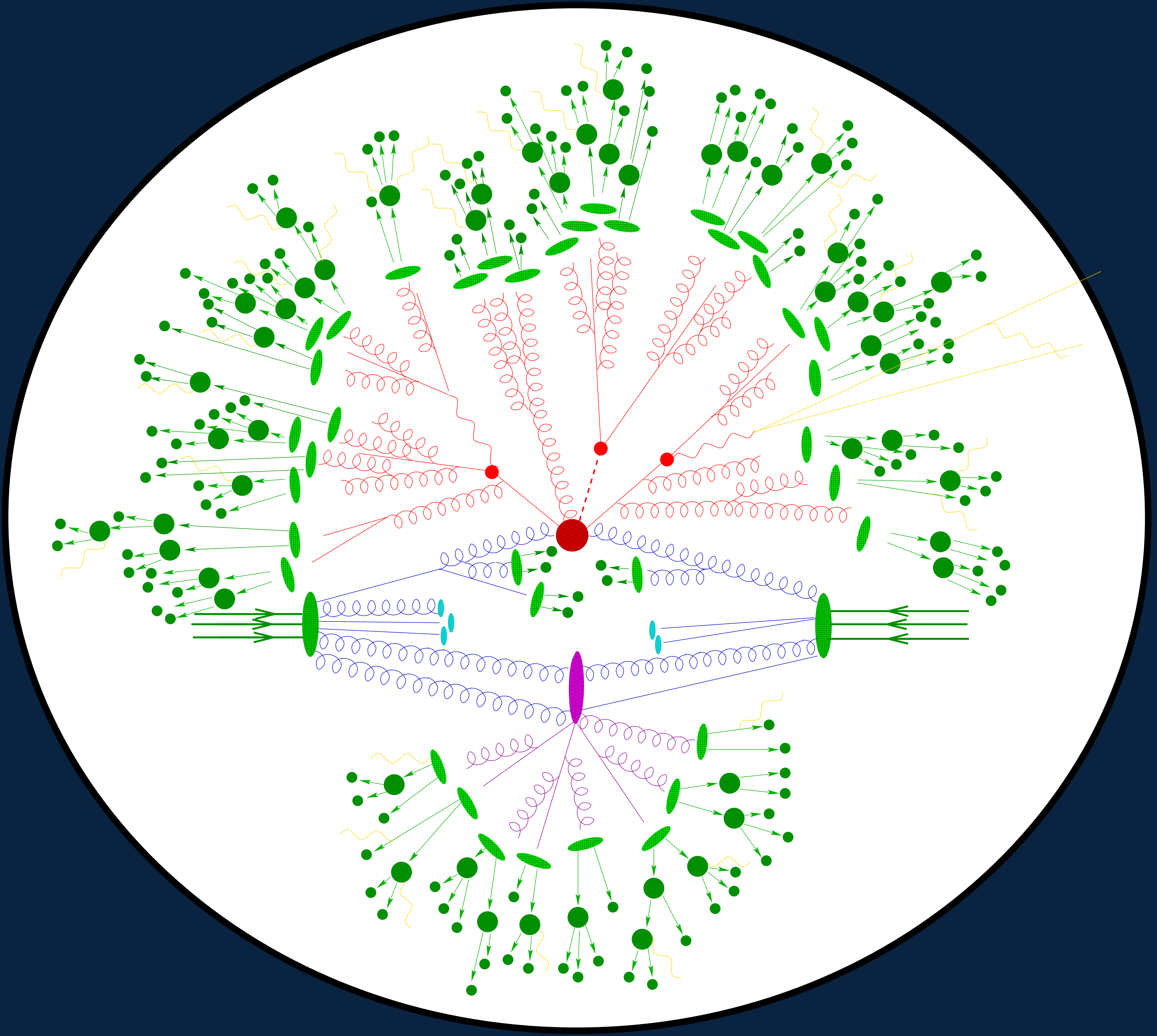
Ramon Winterhalder — UC Louvain

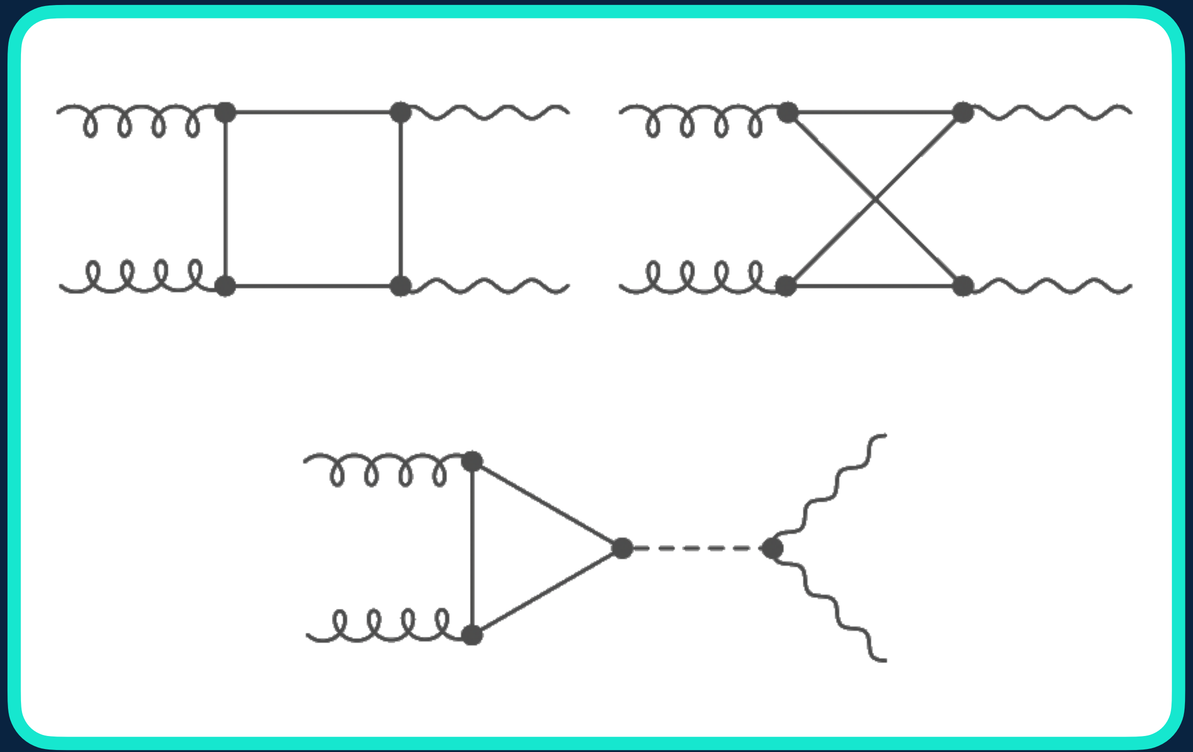
In collaboration with Anja Butter, Theo Heimel, Joshua Isaacson, Claudius Krause, Fabio Maltoni, Olivier Mattelaer and Tilman Plehn

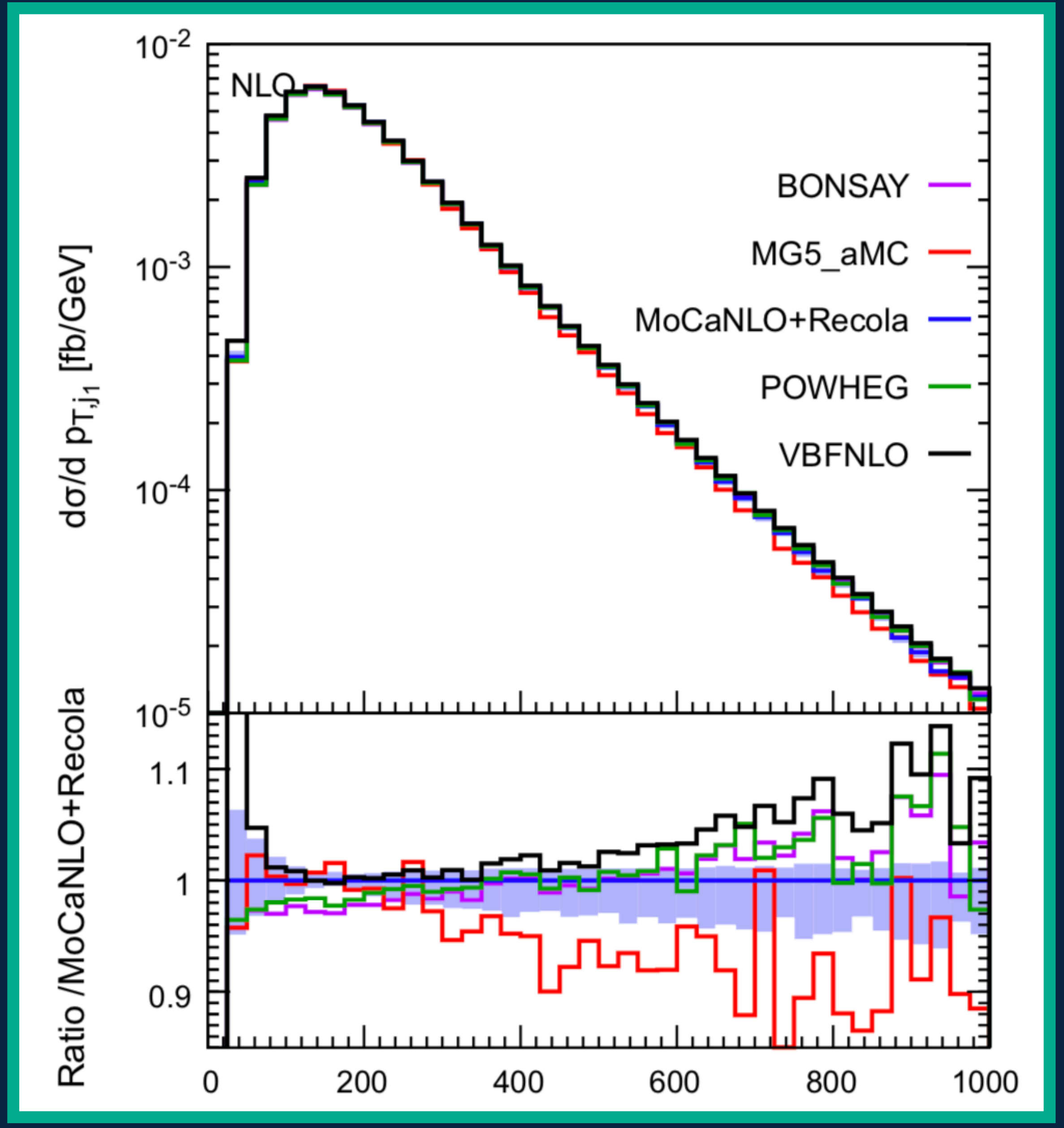
Data analysis in HEP



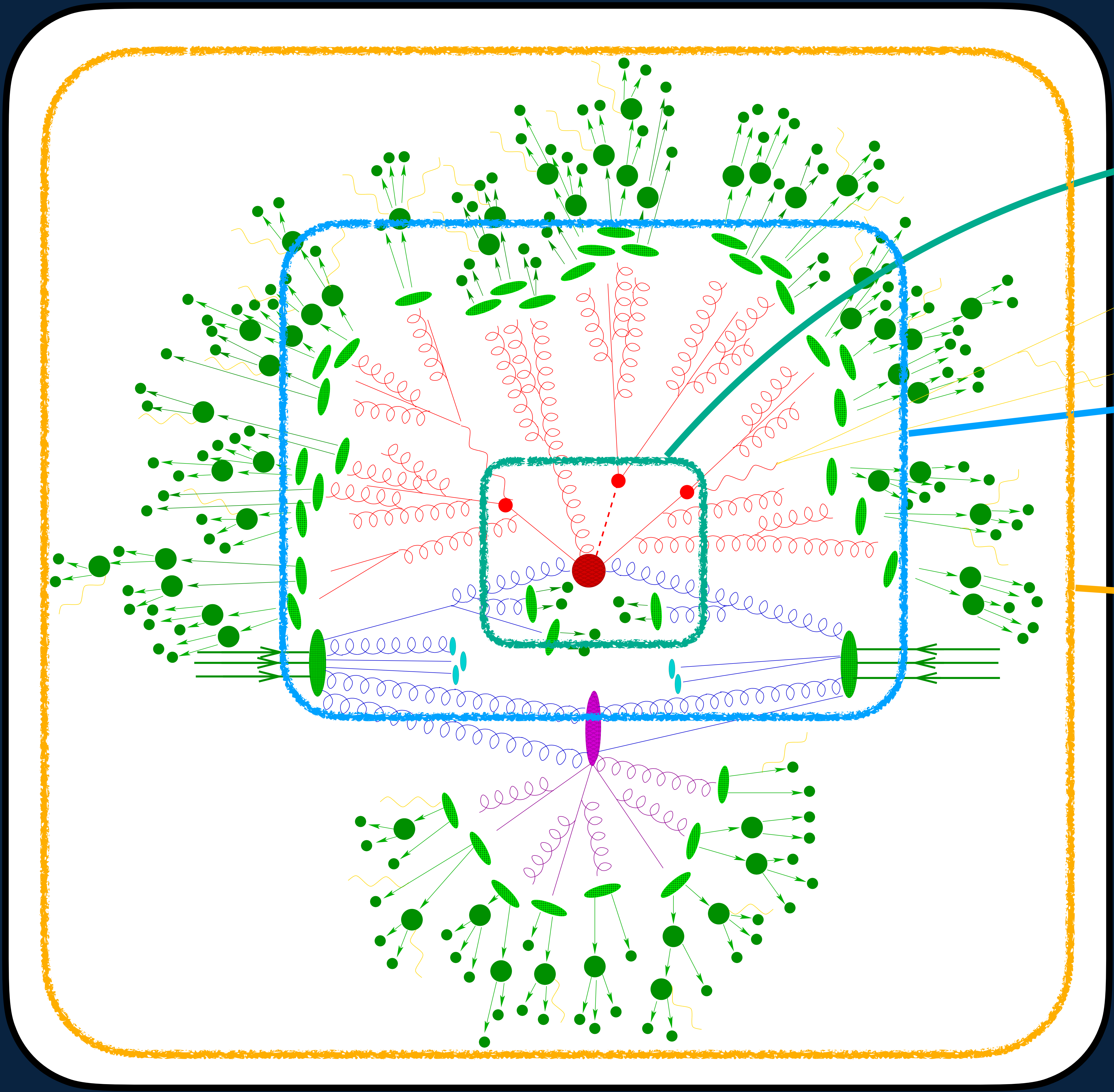








Simulations for the LHC



✦ **Hard Process**

✤ Depends on the Model (SM, SUSY,…)

✤ Perturbative QCD

✦ **Parton Showering**

✤ Universal (QCD)

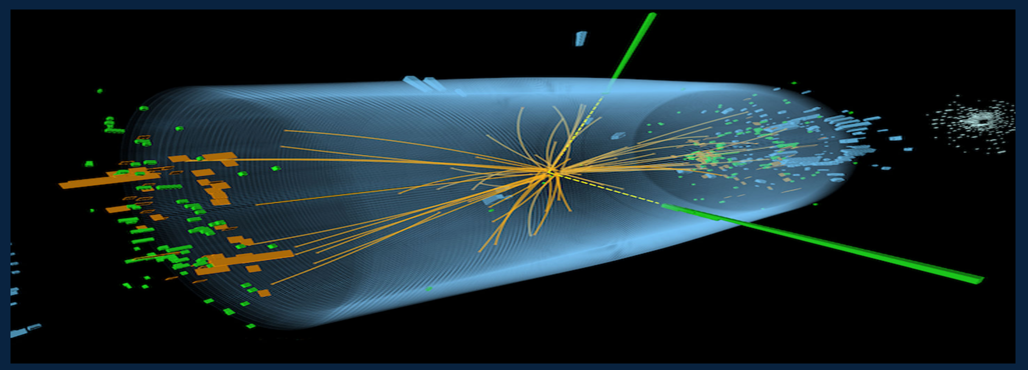
✦ **Hadronization**

✤Model-based, universal

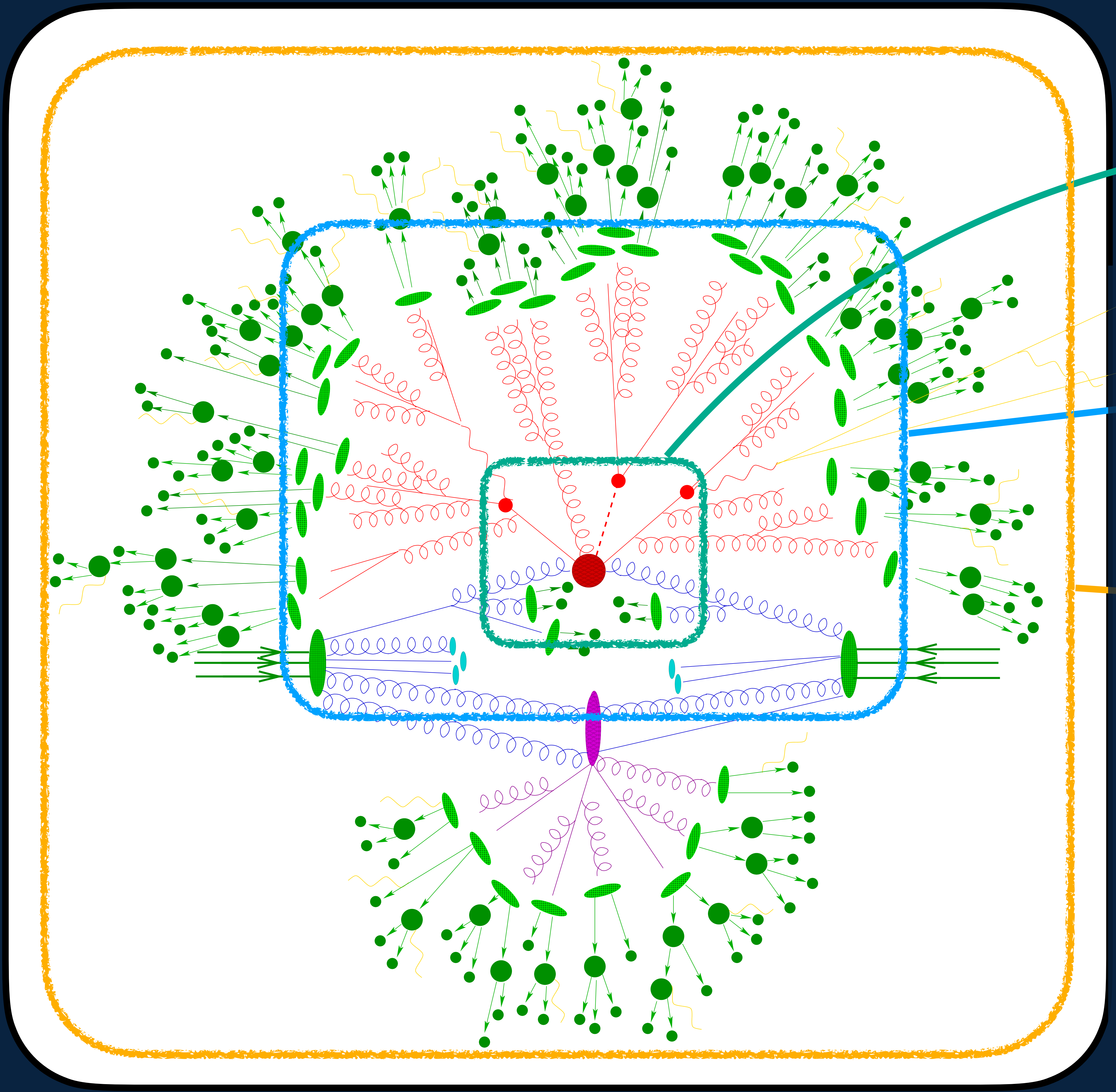
✦ **Underlying Event**

✤Model-based, non-universal

✦ **Detector Simulation**



Simulations for the LHC



✦ **Hard Process**

- ✤ Depends on the Model (SM, SUSY,…)
- ✤ Perturbative QCD

✦ **Parton Showering**

- ✤ Universal (QCD)

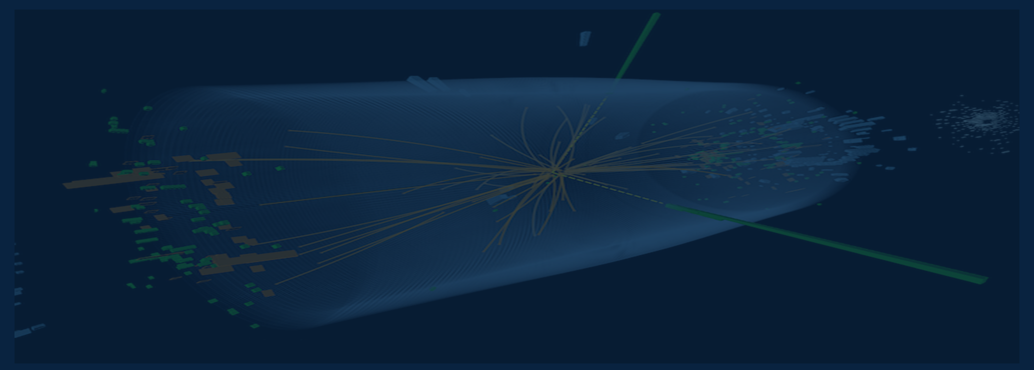
✦ **Hadronization**

- ✤ Model-based, universal

✦ **Underlying Event**

- ✤ Model-based, non-universal

✦ **Detector Simulation**



Theory predictions in HEP

$$M_{\lambda,c,\ldots}(p_a, p_b \,|\, p_1, \ldots, p_n) : \mathbb{M} \to \mathbb{C}$$

Quantum numbers:
spin, colour charge etc.

Kinematics:
Momenta in Minkowski space, masses, etc.

$$\sigma = \frac{1}{\text{flux}} \sum_{a,b} \int \mathrm{d}x_a \mathrm{d}x_b \, f(x_a) f(x_b) \int \mathrm{d}\Phi_n \, \langle \, | M_{\lambda,c,\ldots}(p_a, p_b \,|\, p_1, \ldots, p_n) |^2 \rangle$$

Cross section:
more generally, differential observables*

PDFs:
convolution over all possible initial state configurations

Phase-space integral:
over final state kinematics

Squared amplitude:
summed over final states, averaged over initial states

Theory predictions in HEP

$$M_{\lambda,c,\ldots}(p_a, p_b \,|\, p_1, \ldots, p_n) : \mathbb{M} \to \mathbb{C}$$

Quantum numbers:
spin, colour charge etc.

Kinematics:
Momenta in Minkowski space, masses, etc.

$$\sigma = \frac{1}{\text{flux}} \sum_{a,b} \int \mathrm{d}x_a \mathrm{d}x_b\, f(x_a) f(x_b) \int \mathrm{d}\Phi_n \langle\, |M_{\lambda,c,\ldots}(p_a, p_b \,|\, p_1, \ldots, p_n)|^2 \rangle$$

Cross section:
more generally, differential observables*

PDFs:
convolution over all possible initial state configurations

Phase-space integral:
over final state kinematics

Squared amplitude:
summed over final states, averaged over initial states

Monte Carlo Integration

Standard Monte Carlo integration

$$I = \int_V \mathrm{d}^d x\, f(x) \simeq \frac{1}{N} \sum_{j=1}^{N} f(x_j) = \langle f \rangle_x \qquad \sigma_I \simeq \sqrt{\frac{\langle f^2 \rangle_x - \langle f \rangle_x^2}{N-1}}$$

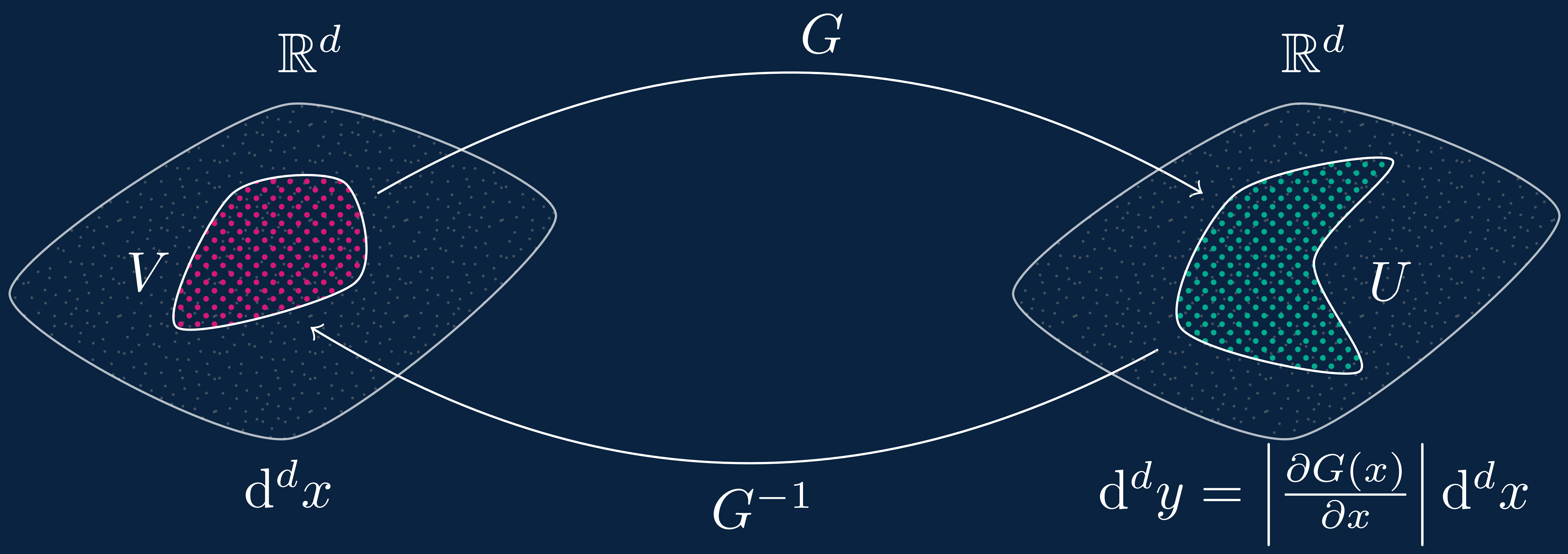
Monte Carlo Integration

Standard Monte Carlo integration

$$I = \int_V \mathrm{d}^d x\, f(x) \simeq \frac{1}{N}\sum_{j=1}^{N} f(x_j) = \langle f \rangle_x \qquad \sigma_I \simeq \sqrt{\frac{\langle f^2 \rangle_x - \langle f \rangle_x^2}{N-1}}$$

Mapping

$$y = G(x) \qquad g(x) = \left|\frac{\partial G(x)}{\partial x}\right|$$



Monte Carlo Integration

Standard Monte Carlo integration

$$I = \int_V \mathrm{d}^d x\, f(x) \simeq \frac{1}{N}\sum_{j=1}^{N} f(x_j) = \langle f \rangle_x \qquad \sigma_I \simeq \sqrt{\frac{\langle f^2 \rangle_x - \langle f \rangle_x^2}{N-1}}$$

Mapping

$$y = G(x) \qquad g(x) = \left| \frac{\partial G(x)}{\partial x} \right|$$

Importance Sampling

$$I = \int_U \mathrm{d}^d y\, \frac{f(x)}{g(x)}\bigg|_{x=G^{-1}(y)} \simeq \langle f/g \rangle_y \quad \sigma_I \simeq \sqrt{\frac{\langle (f/g)^2 \rangle_y - \langle f/g \rangle_y^2}{N-1}}$$

Monte Carlo Integration

Standard Monte Carlo integration

$$I = \int_V \mathrm{d}^d x\, f(x) \simeq \frac{1}{N} \sum_{j=1}^{N} f(x_j) = \langle f \rangle_x \qquad \sigma_I \simeq \sqrt{\frac{\langle f^2 \rangle_x - \langle f \rangle_x^2}{N-1}}$$

Mapping

$$y = G(x) \qquad g(x) = \left| \frac{\partial G(x)}{\partial x} \right|$$

Importance Sampling

$$I = \int_U \mathrm{d}^d y\, \frac{f(x)}{g(x)} \Bigg|_{x=G^{-1}(y)} \simeq \langle f/g \rangle_y \quad \sigma_I \simeq \sqrt{\frac{\langle (f/g)^2 \rangle_y - \langle f/g \rangle_y^2}{N-1}}$$

Wanted

$$\sigma_I \to 0$$

Monte Carlo Integration

Standard Monte Carlo integration

$$I = \int_V \mathrm{d}^d x\, f(x) \simeq \frac{1}{N} \sum_{j=1}^{N} f(x_j) = \langle f \rangle_x \qquad \sigma_I \simeq \sqrt{\frac{\langle f^2 \rangle_x - \langle f \rangle_x^2}{N-1}}$$

Mapping

$$y = G(x) \qquad g(x) = \left| \frac{\partial G(x)}{\partial x} \right|$$

Importance Sampling

$$I = \int_U \mathrm{d}^d y\, \frac{f(x)}{g(x)} \Bigg|_{x=G^{-1}(y)} \simeq \langle f/g \rangle_y \quad \sigma_I \simeq \sqrt{\frac{\langle (f/g)^2 \rangle_y - \langle f/g \rangle_y^2}{N-1}}$$

Wanted

$\sigma_I \to 0$

Needs

$f/g \approx$ const

Monte Carlo Integration

Standard Monte Carlo integration

$$I = \int_V \mathrm{d}^d x\, f(x) \simeq \frac{1}{N}\sum_{j=1}^{N} f(x_j) = \langle f \rangle_x \qquad \sigma_I \simeq \sqrt{\frac{\langle f^2\rangle_x - \langle f \rangle_x^2}{N-1}}$$

Mapping

$$y = G(x) \qquad g(x) = \left|\frac{\partial G(x)}{\partial x}\right|$$

Importance Sampling

$$I = \int_U \mathrm{d}^d y\, \left.\frac{f(x)}{g(x)}\right|_{x=G^{-1}(y)} \simeq \langle f/g \rangle_y \quad \sigma_I \simeq \sqrt{\frac{\langle (f/g)^2\rangle_y - \langle f/g \rangle_y^2}{N-1}}$$

Multi-Channel Integration $\longrightarrow$ $\sum \alpha_i(x) = 1$

$$I = \int_V \mathrm{d}^d x\, f(x) = \sum_i \int_V \mathrm{d}^d x\, \alpha_i(x) f(x) = \sum_i \int_{U_i} \mathrm{d}^d y_i\, \alpha_i(x) \left.\frac{f(x)}{g_i(x)}\right|_{x\equiv x(y_i)}$$

Channel Mappings

$$y_i = G_i(x) \qquad g_i(x) = \left|\frac{\partial G_i(x)}{\partial x}\right|$$

Neural Multi-Channel Monte Carlo

Multi-Channel Importance Sampling

$$I = \sum_i \int_{U_i} \alpha_i(x) \frac{f(x)}{g_i(x)} \mathrm{d}G_i(x)$$

Neural Multi-Channel Monte Carlo

Multi-Channel Importance Sampling

$$I = \sum_i \int_{U_i} \alpha_i(x) \frac{f(x)}{g_i(x)} \mathrm{d}G_i(x)$$

Neural Channel Mappings

$$G_i(x) \to G_i^\theta(x), \quad g_i^\theta(x) = \left| \frac{\partial G_i^\theta(x)}{\partial x} \right|$$

n-dimensional remapping

- tractable Jacobian with **normalizing flow**

Neural Multi-Channel Monte Carlo

Multi-Channel Importance Sampling

$$I = \sum_i \int_{U_i} \alpha_i(x) \frac{f(x)}{g_i(x)} \mathrm{d}G_i(x)$$

Neural Channel Mappings

$$G_i(x) \to G_i^\theta(x), \quad g_i^\theta(x) = \left| \frac{\partial G_i^\theta(x)}{\partial x} \right|$$

n-dimensional remapping

- tractable Jacobian with **normalizing flow**

→ *So far*: improvement factors of ~2-4 are achieved [2001.05486, 2001.05478, 2001.10028, 2112.09145]

Neural Multi-Channel Monte Carlo

Multi-Channel Importance Sampling

$$I = \sum_i \int_{U_i} \alpha_i(x) \frac{f(x)}{g_i(x)} \mathrm{d}G_i(x)$$

Neural Channel Mappings

$$G_i(x) \to G_i^{\theta}(x), \quad g_i^{\theta}(x) = \left| \frac{\partial G_i^{\theta}(x)}{\partial x} \right|$$

n-dimensional remapping

- tractable Jacobian with **normalizing flow**

Possible Improvements?

→ *So far*: improvement factors of ~2-4 are achieved [2001.05486, 2001.05478, 2001.10028, 2112.09145]

Neural Multi-Channel Monte Carlo

Multi-Channel Importance Sampling

$$I = \sum_i \int_{U_i} \alpha_i(x) \frac{f(x)}{g_i(x)} \mathrm{d}G_i(x)$$

Neural Channel Weights

$$\alpha_i(x) \to \alpha_i^{\phi}(x), \quad \sum_i \alpha_i^{\phi}(x) = 1$$

k-dimensional regression

- with boundary condition

Neural Channel Mappings

$$G_i(x) \to G_i^{\theta}(x), \quad g_i^{\theta}(x) = \left| \frac{\partial G_i^{\theta}(x)}{\partial x} \right|$$

n-dimensional remapping

- tractable Jacobian with **normalizing flow**

⟶ *So far*: improvement factors of ~2-4 are achieved [2001.05486, 2001.05478, 2001.10028, 2112.09145]

Neural Multi-Channel Monte Carlo

Multi-Channel Importance Sampling

$$I = \sum_i \int_{U_i} \alpha_i(x) \frac{f(x)}{g(x|i)} \mathrm{d}G(x|i)$$

Neural Channel Weights

$$\alpha_i(x) \rightarrow \alpha_i^{\phi}(x), \quad \sum_i \alpha_i^{\phi}(x) = 1$$

k-dimensional regression

- with boundary condition

Conditional Neural Channel Mappings

$$G_i(x) \rightarrow G^{\theta}(x|i), \quad g^{\theta}(x|i) = \left| \frac{\partial G^{\theta}(x|i)}{\partial x} \right|$$

n-dimensional remapping

- tractable Jacobian with **normalizing flow**
- conditioned on channel

⟶ *So far*: improvement factors of ~2-4 are achieved [2001.05486, 2001.05478, 2001.10028, 2112.09145]

Neural Multi-Channel Monte Carlo

Multi-Channel Importance Sampling

$$I = \sum_i \int_{U_i} \alpha_i(x) \frac{f(x)}{g(x|i)} \mathrm{d}G(x|i)$$

Further Improvements?

Neural Channel Weights

$$\alpha_i(x) \to \alpha_i^\phi(x), \quad \sum_i \alpha_i^\phi(x) = 1$$

k-dimensional regression

- with boundary condition

Conditional Neural Channel Mappings

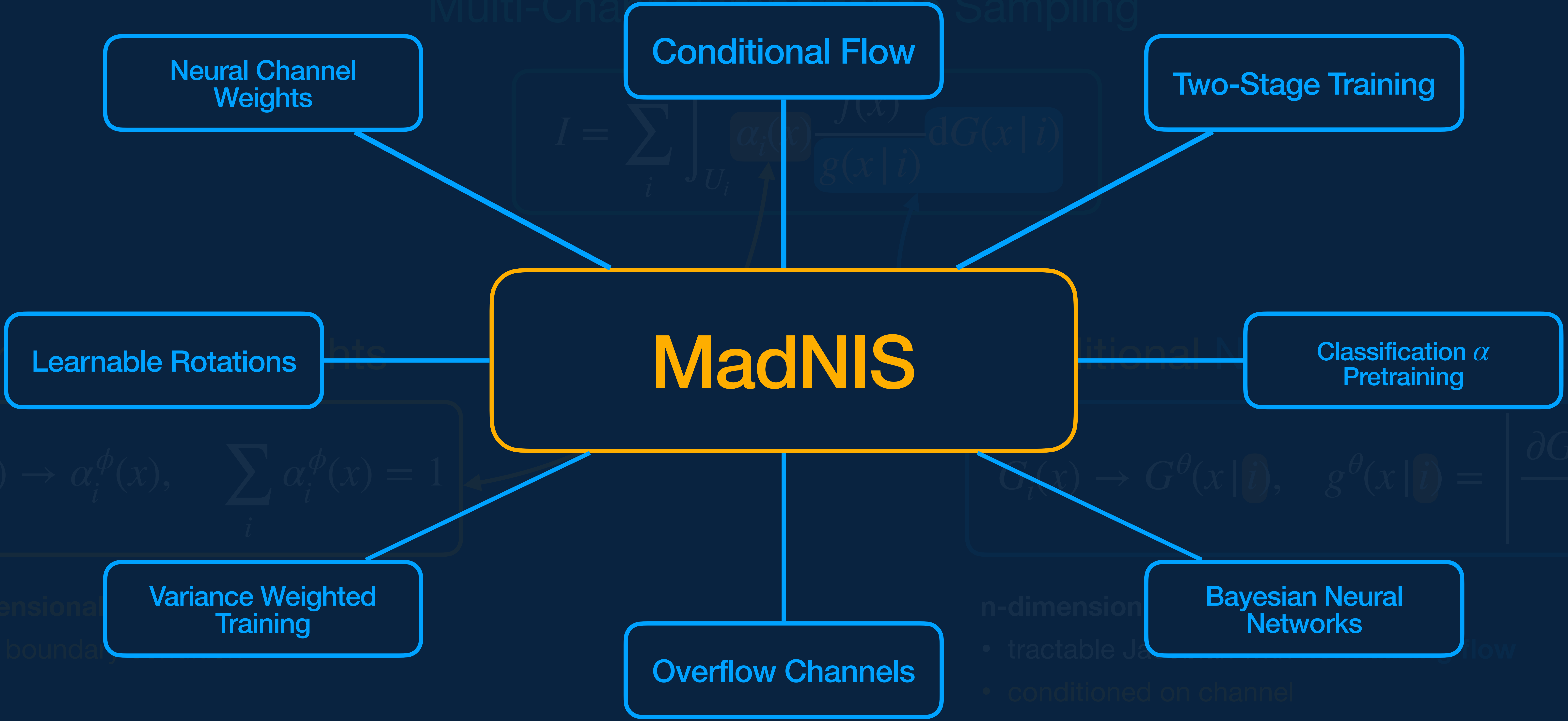
$$G_i(x) \to G^\theta(x|i), \quad g^\theta(x|i) = \left|\frac{\partial G^\theta(x|i)}{\partial x}\right|$$

n-dimensional remapping

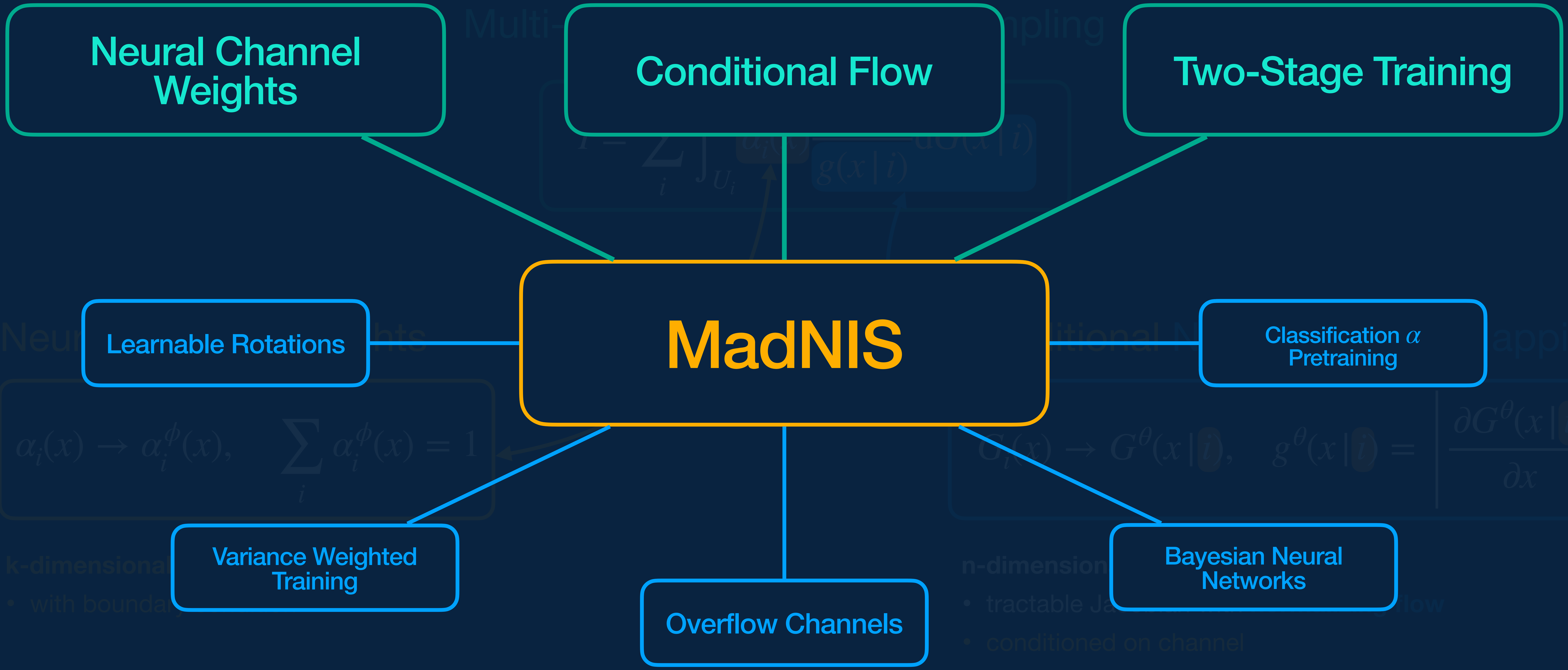
- tractable Jacobian with normalizing flow
- conditioned on channel

→ *So far*: improvement factors of ~2-4 are achieved [2001.05486, 2001.05478, 2001.10028, 2112.09145]

Neural Multi-Channel Monte Carlo



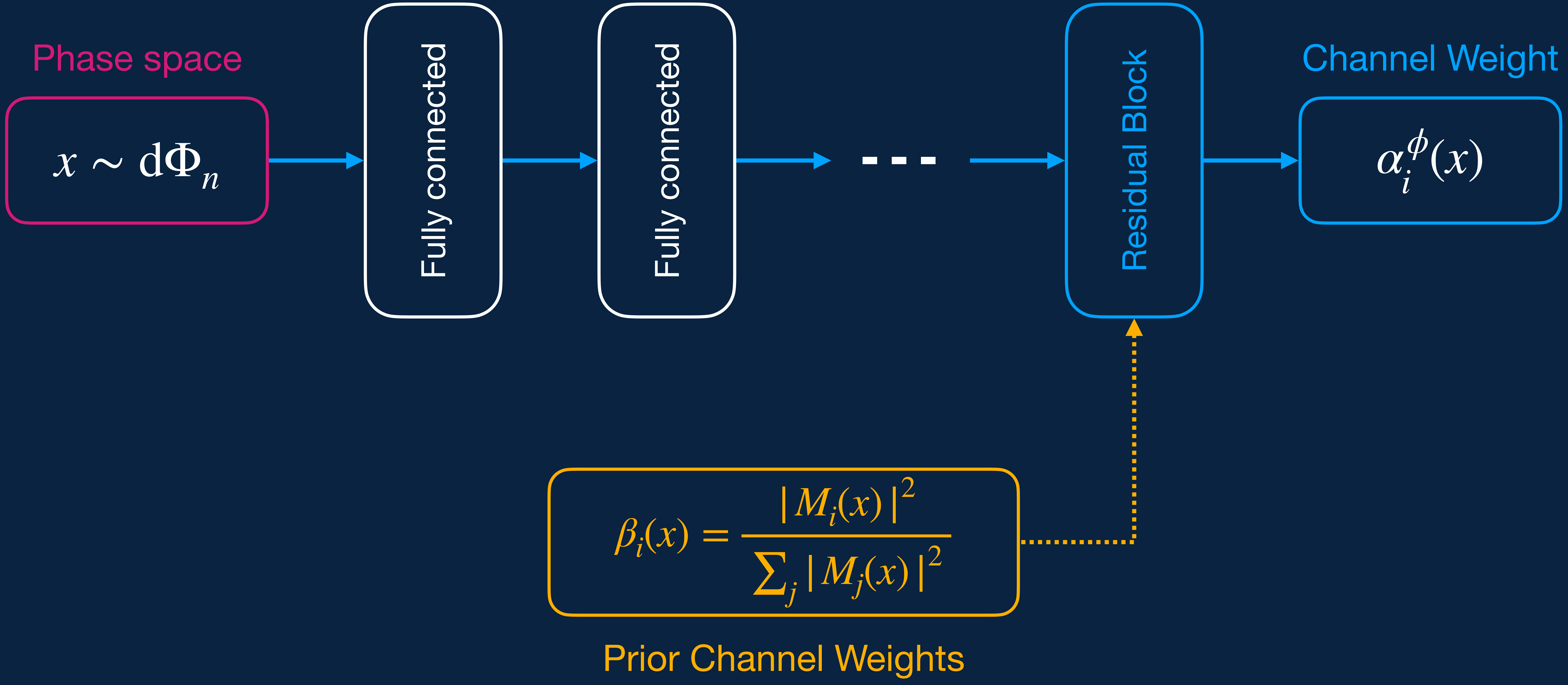
Neural Multi-Channel Monte Carlo



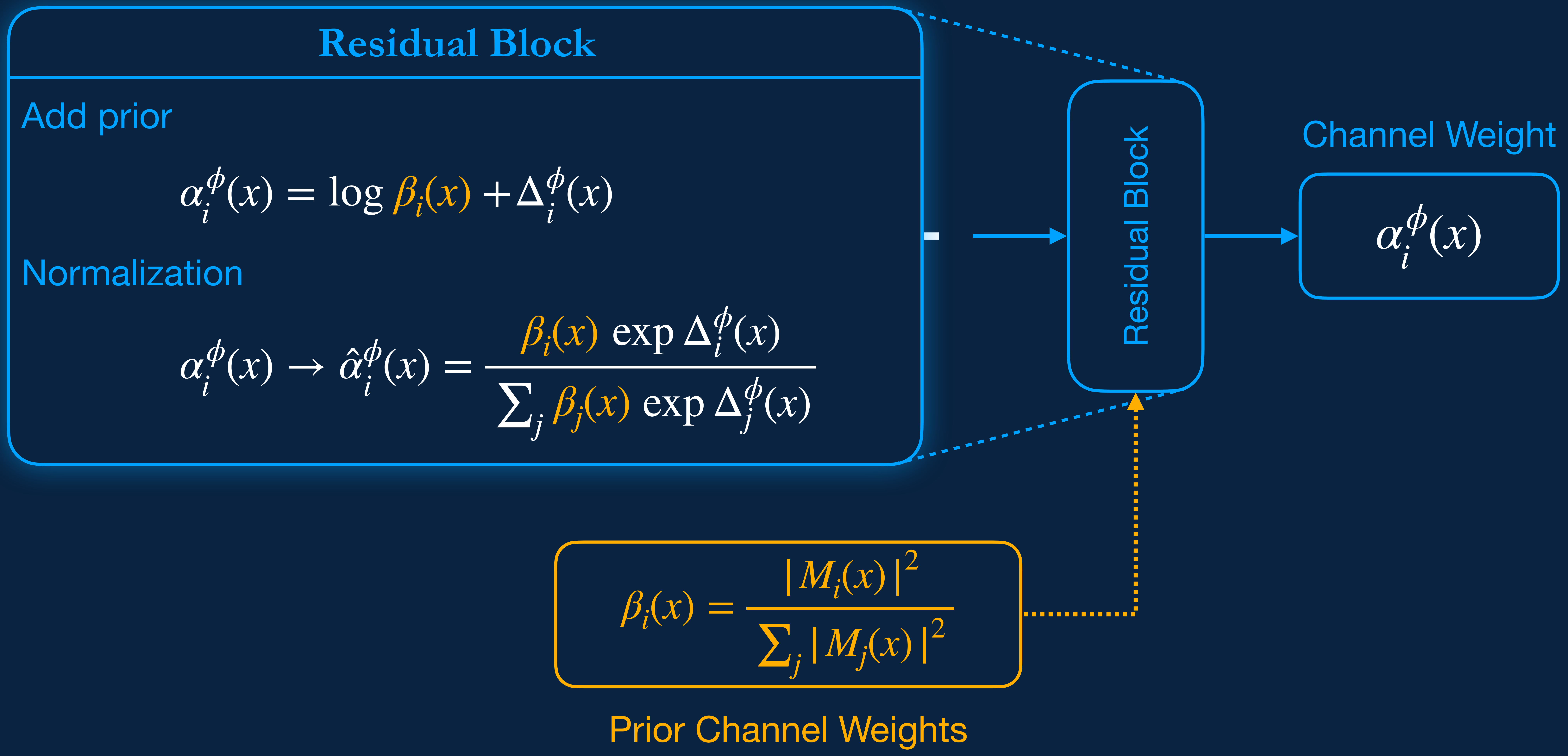
MadNIS

Neural Channel Weights

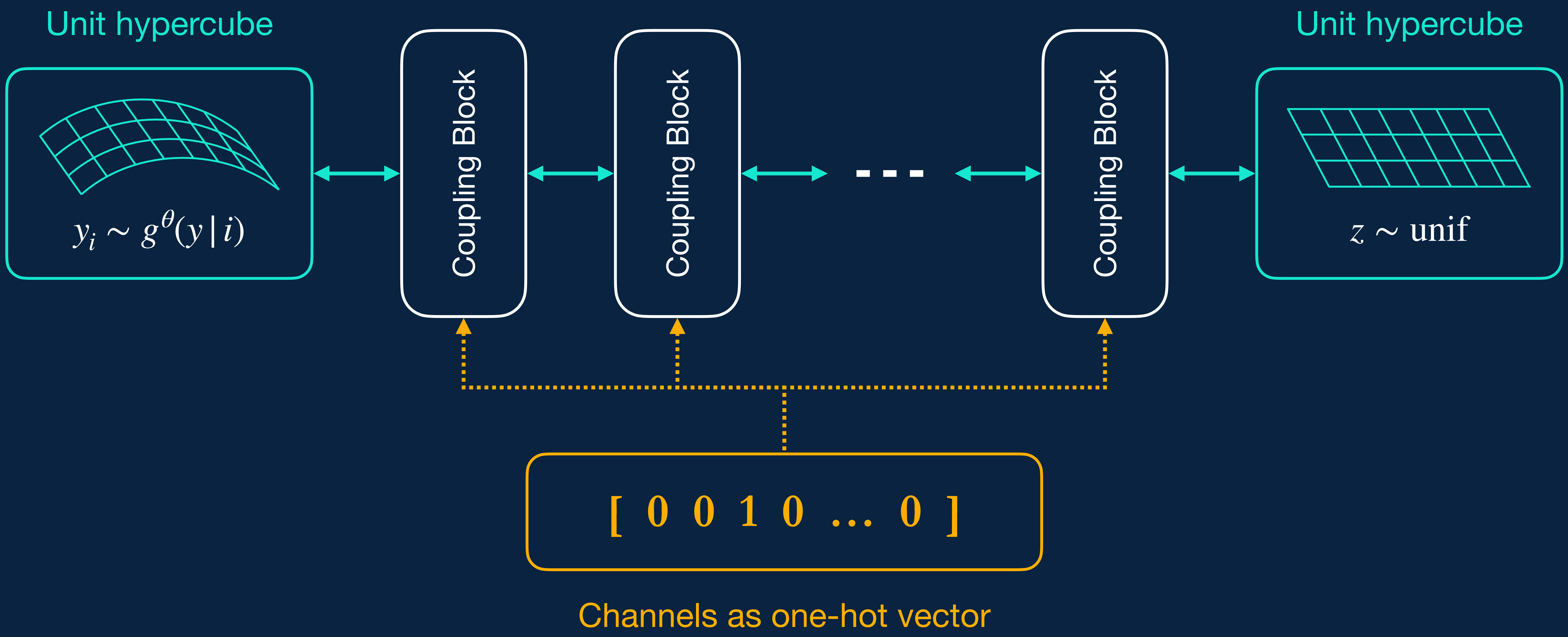
Neural Channel Weights



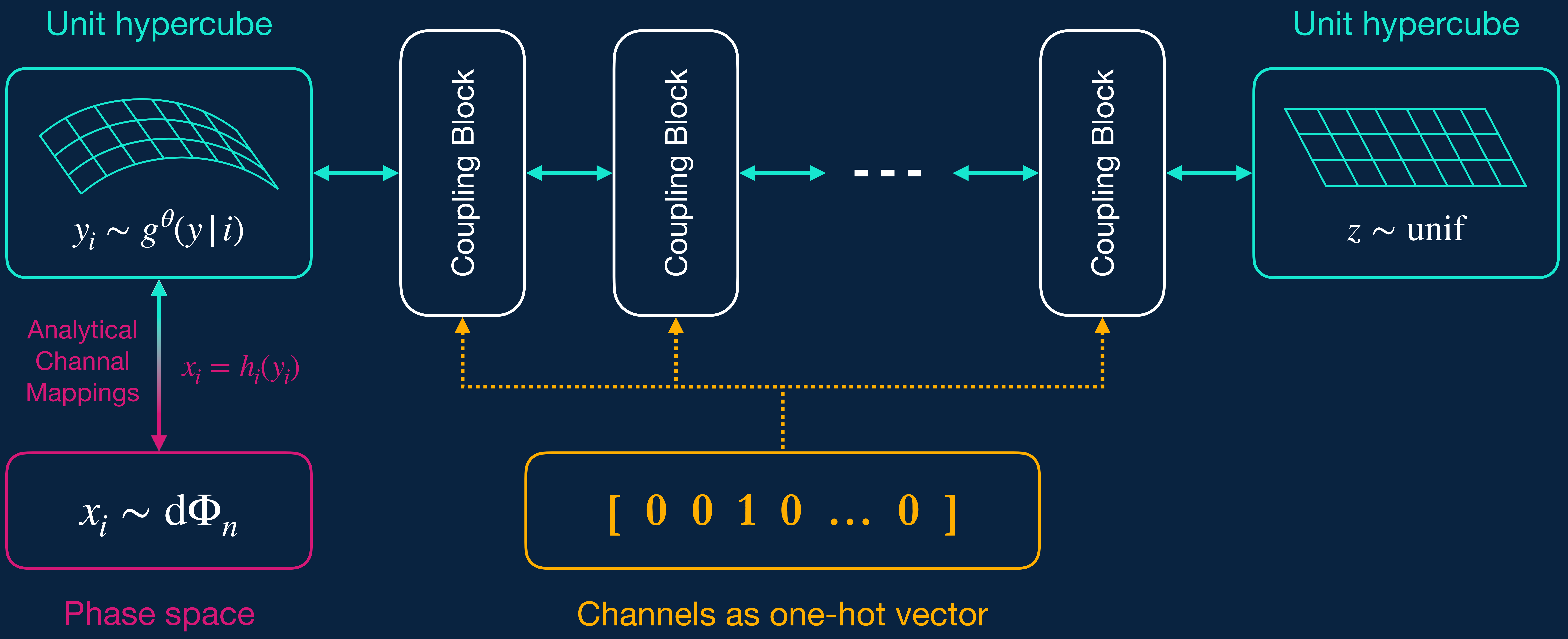
Neural Channel Weights



Conditional Normalizing Flow



Conditional Normalizing Flow



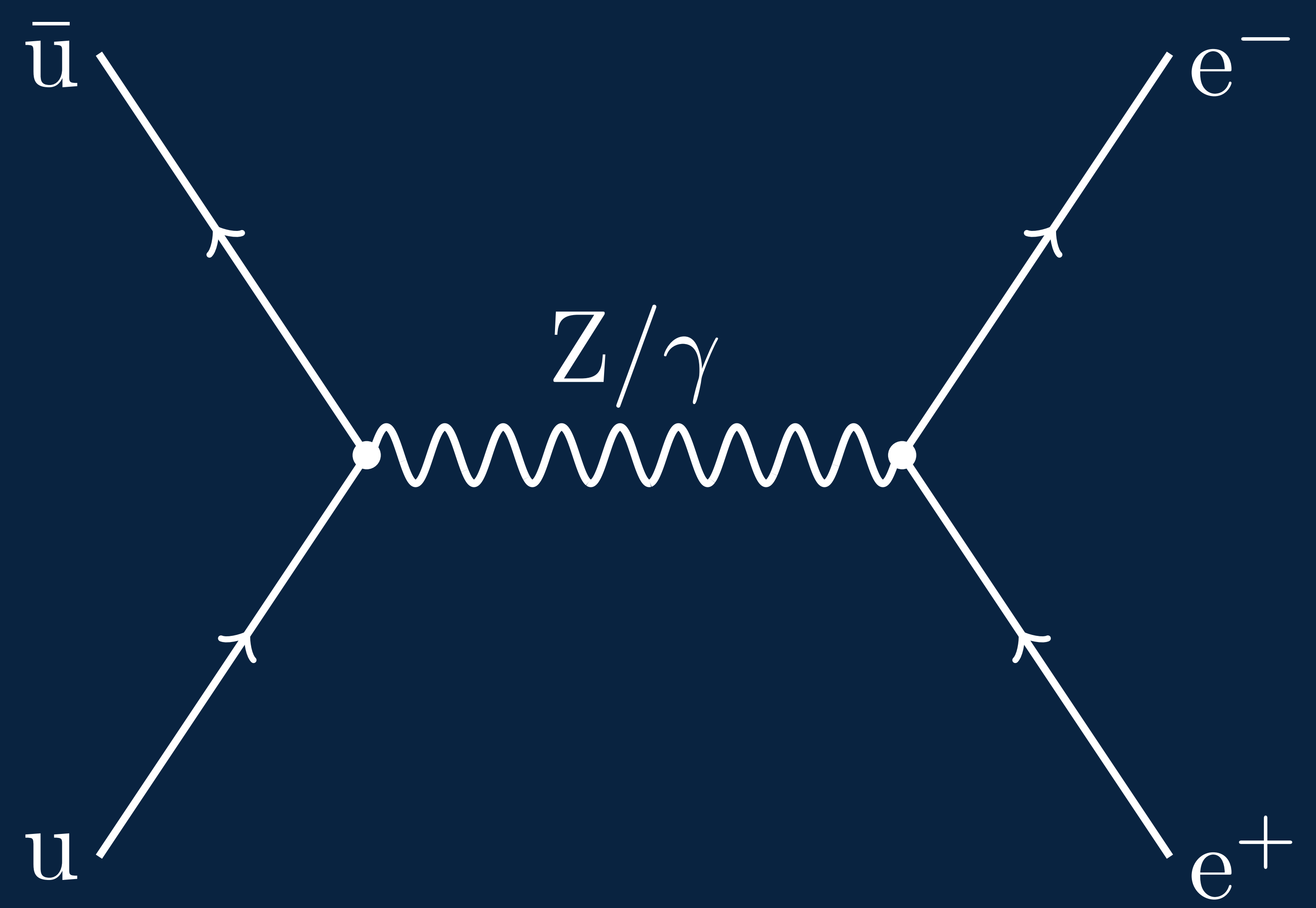
Example

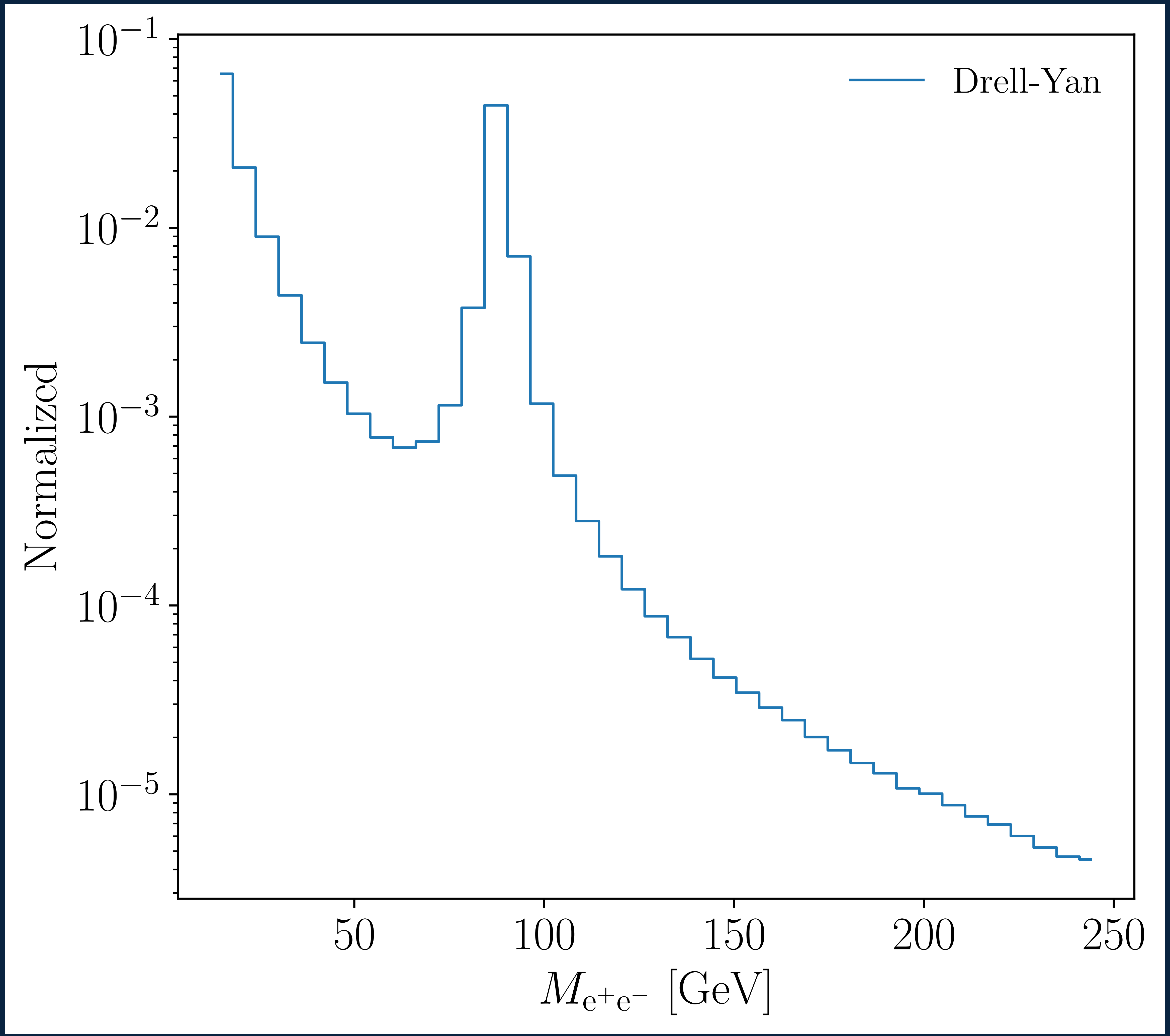
Drell-Yan + BSM

Example: Drell-Yan

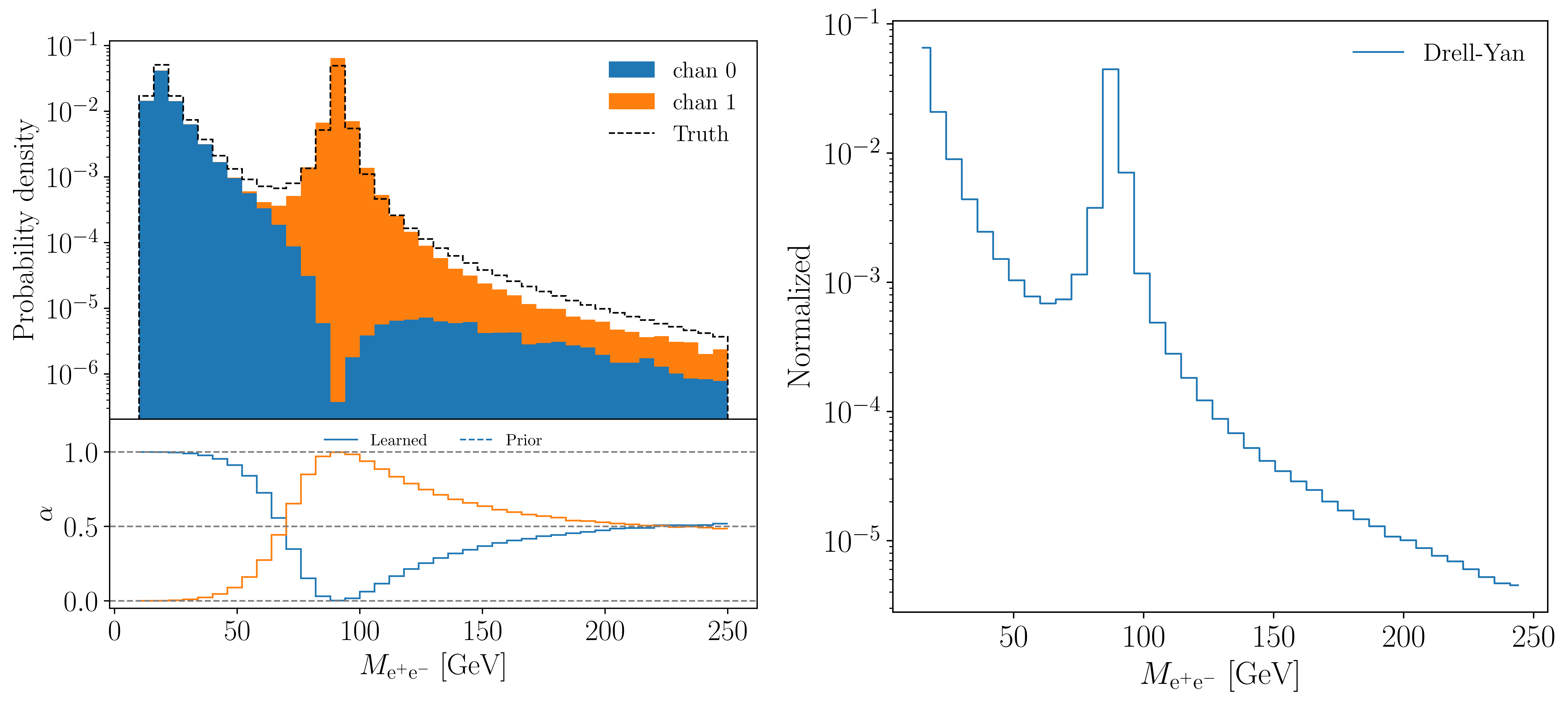
Implementation

- Custom amplitude in TensorFlow2
- Custom PS mappings in TensorFlow2
- PDFs from PDFFlow [2009.06635]

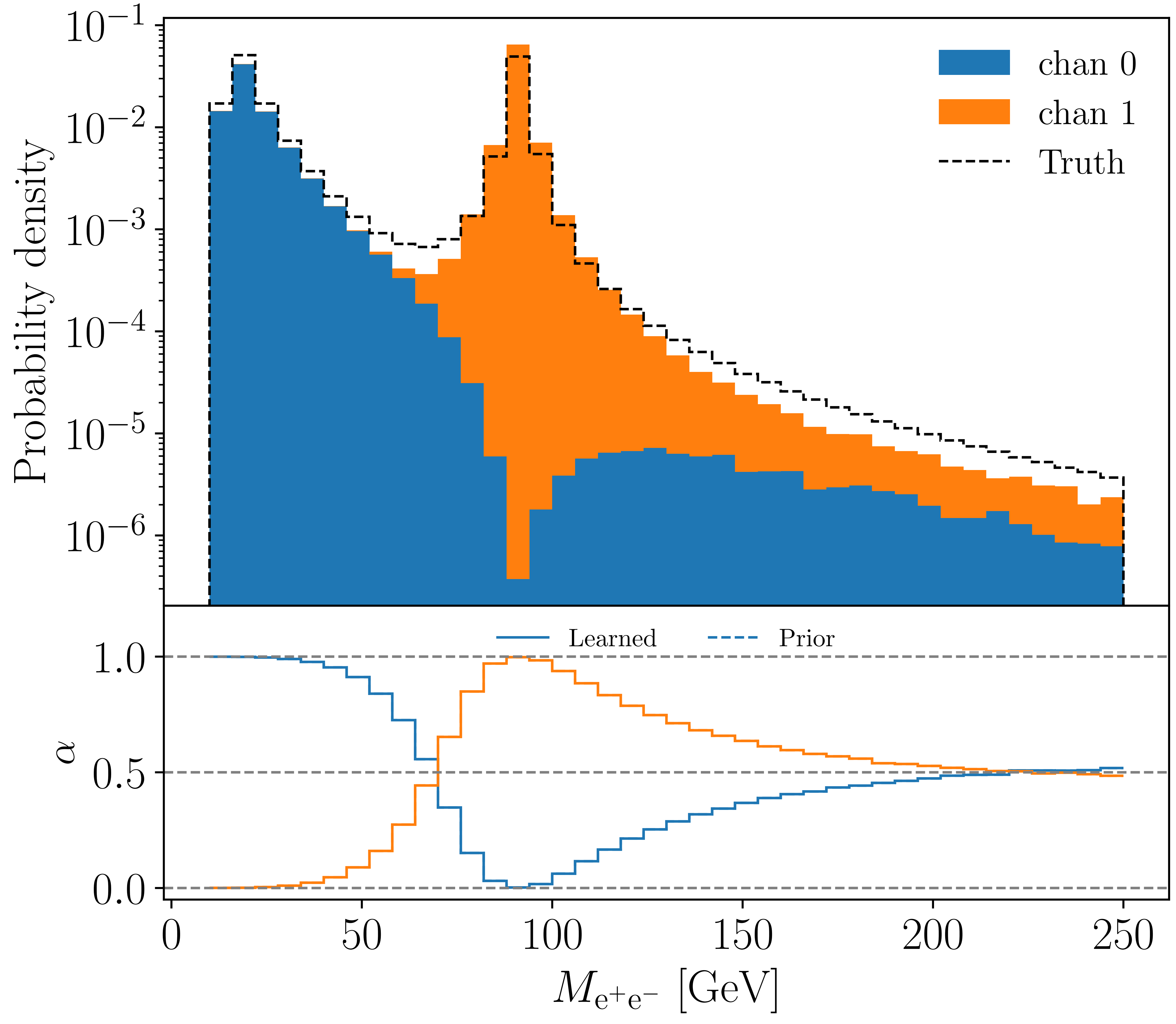


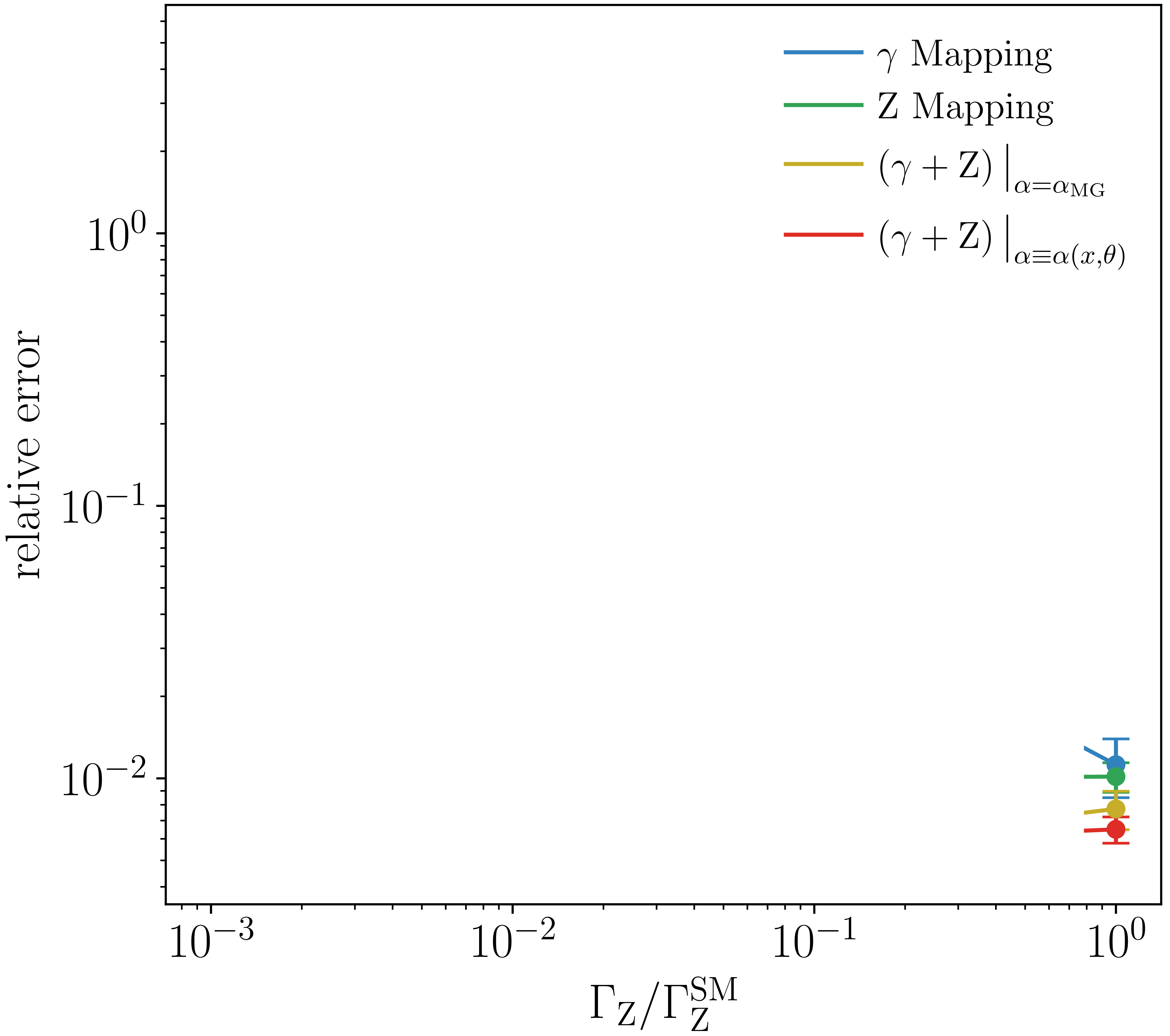


Example: Drell-Yan

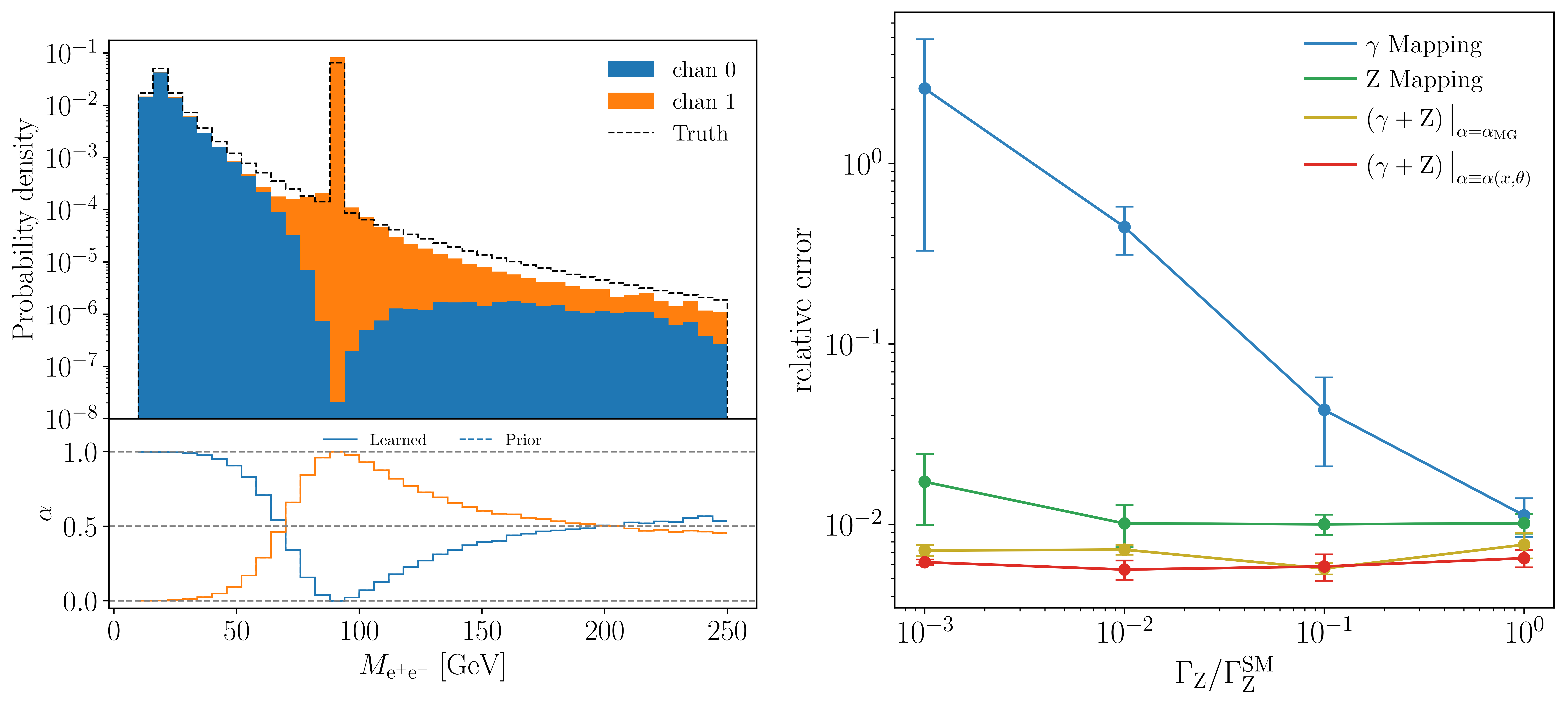


Example: Drell-Yan

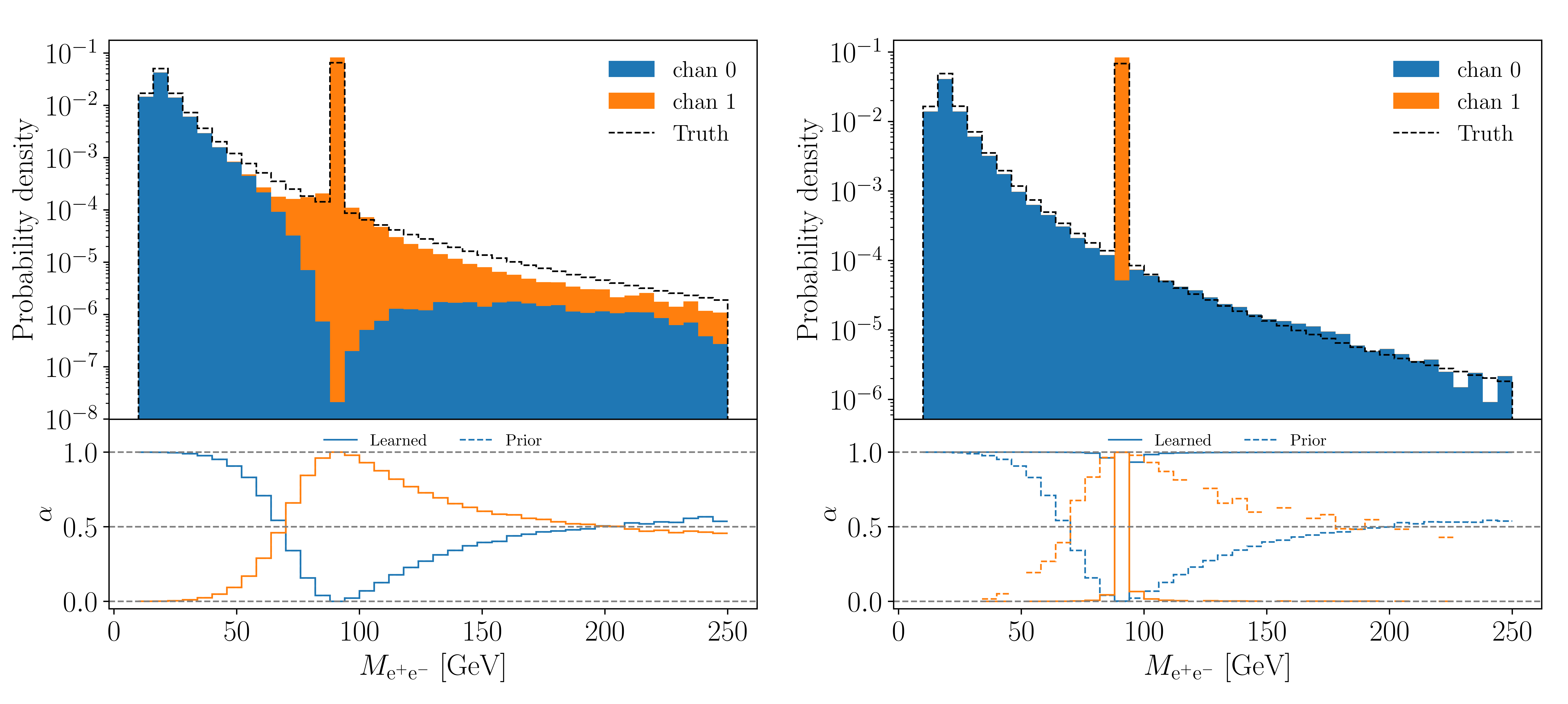




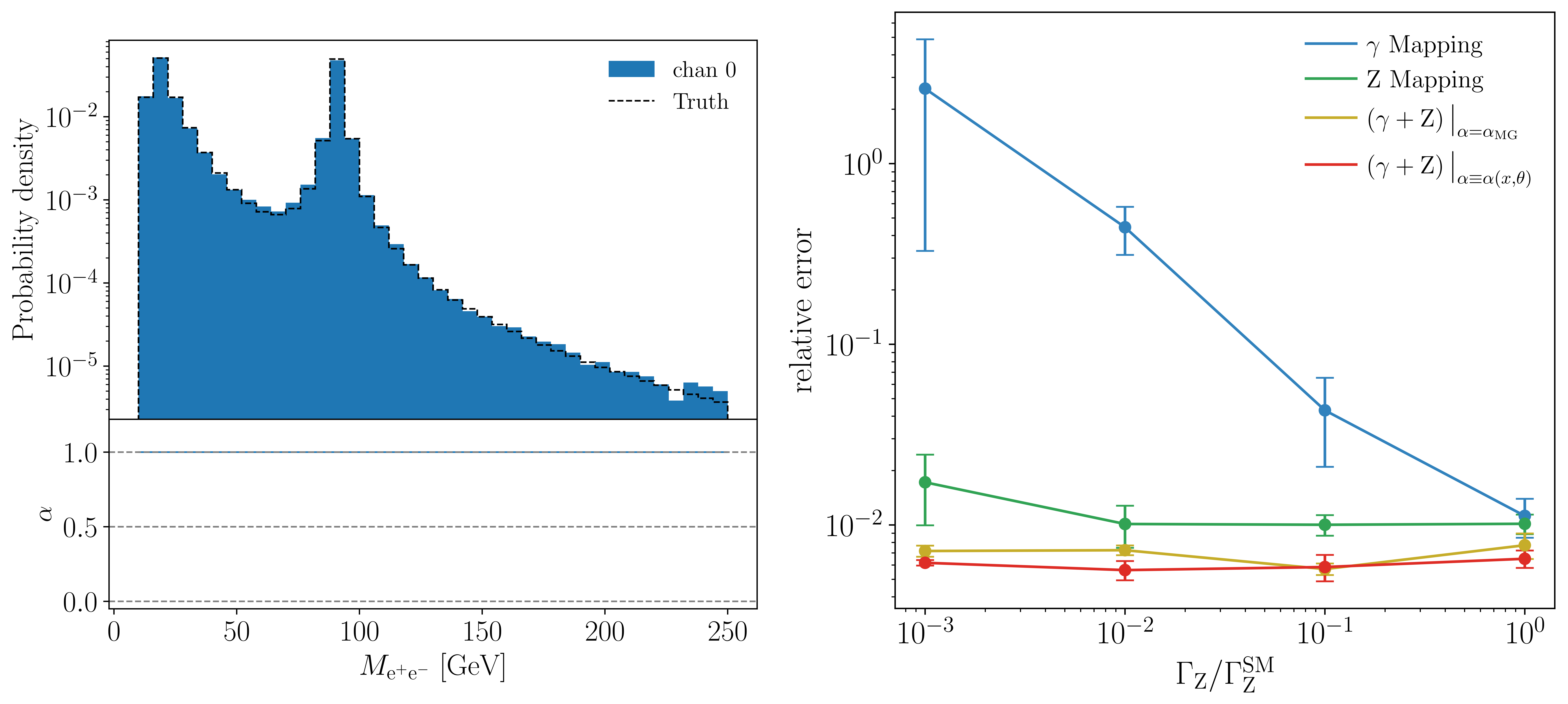
Example: Drell-Yan



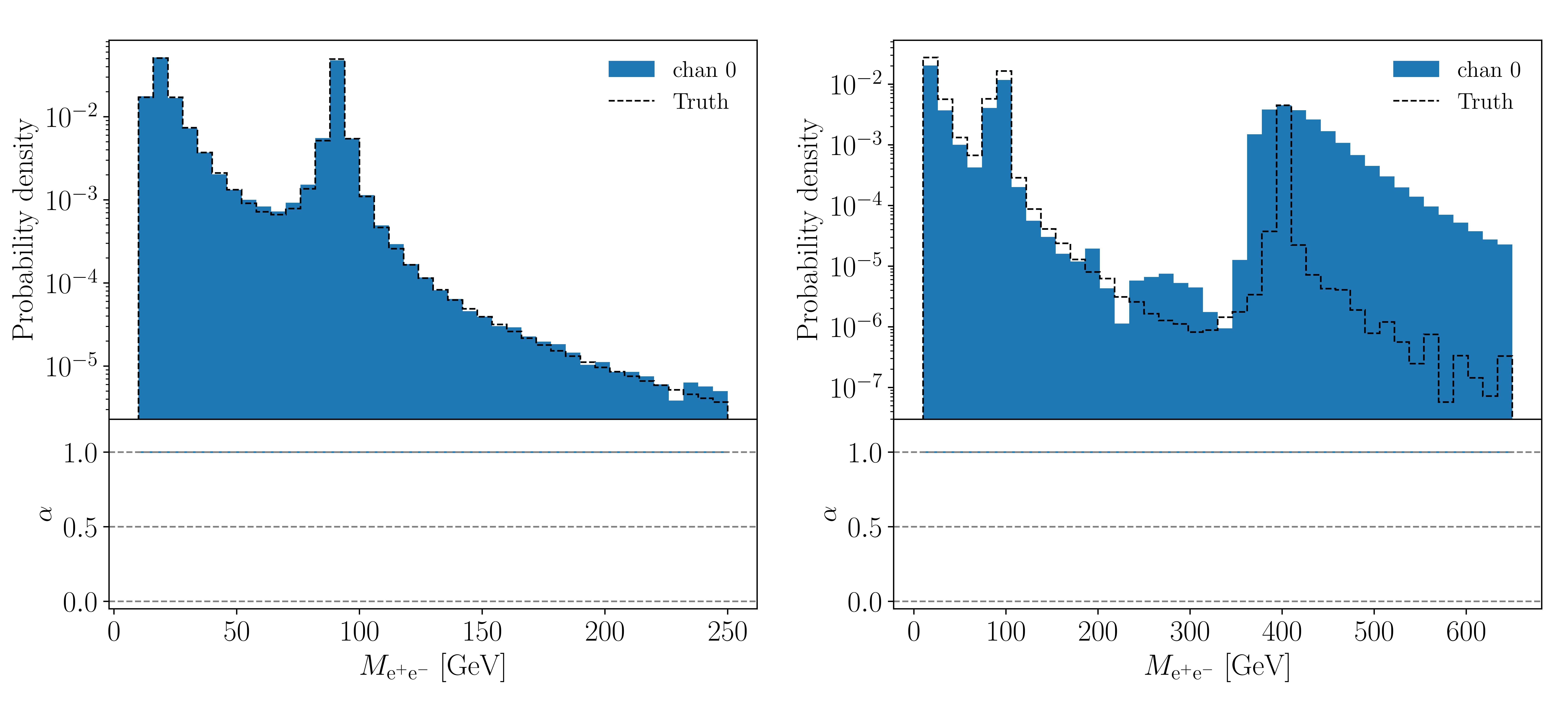
Example: Drell-Yan



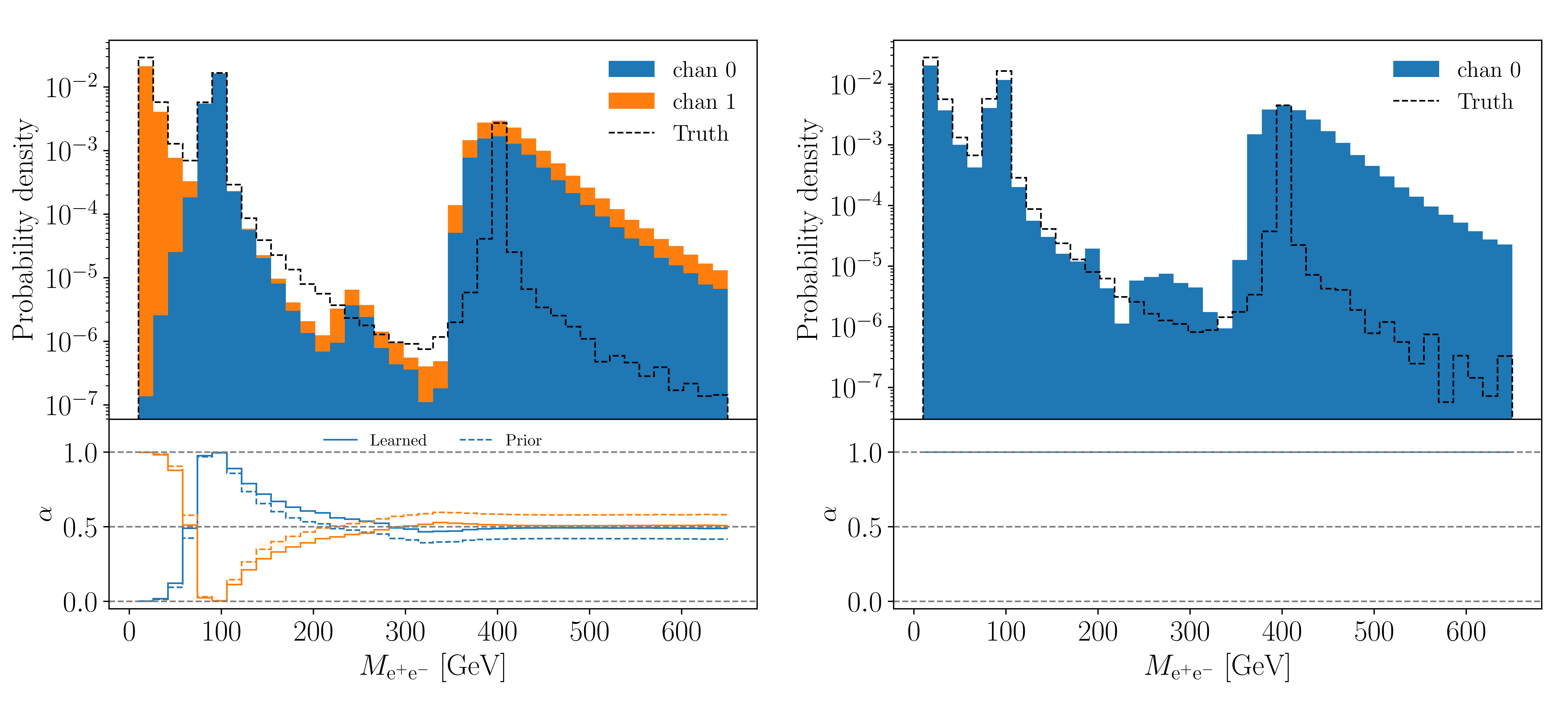
Example: Drell-Yan



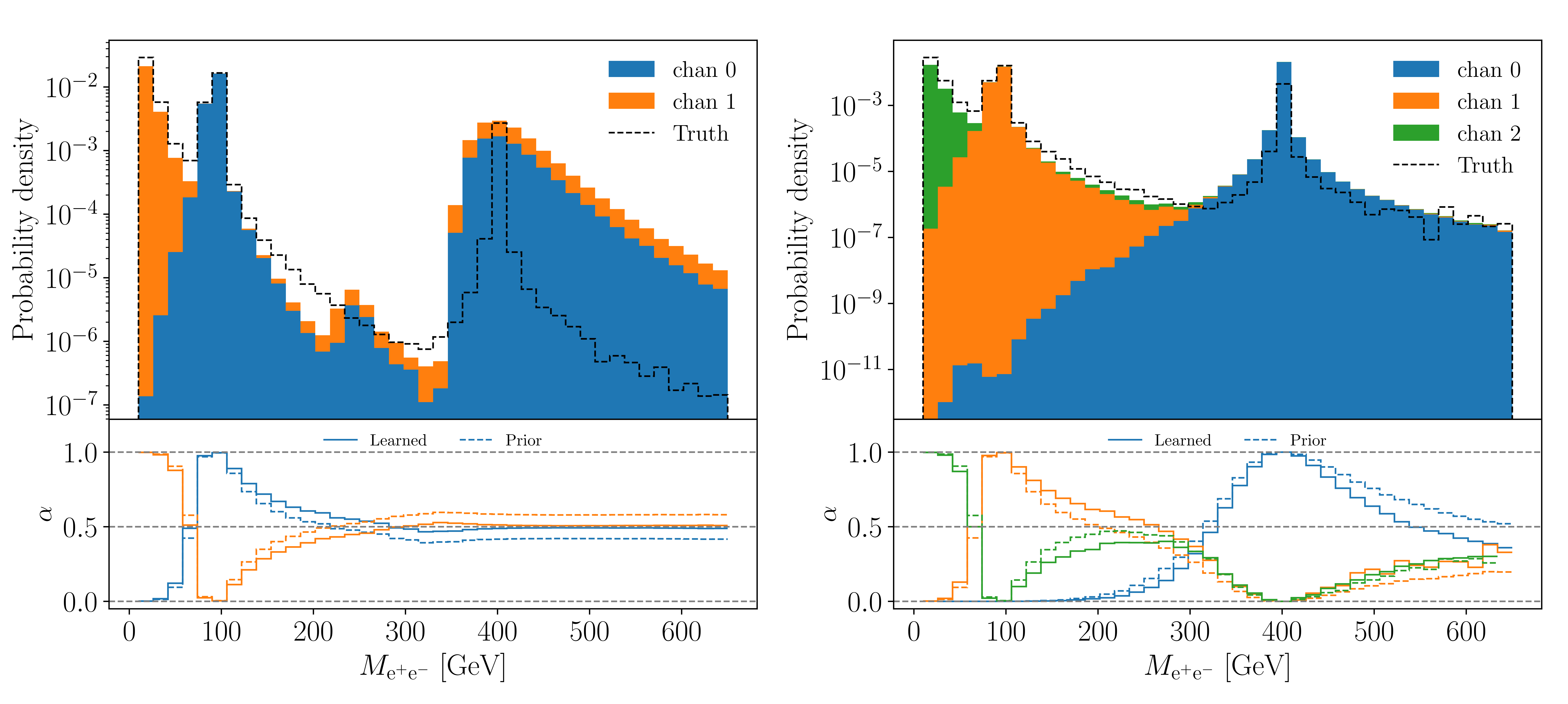
Example: Drell-Yan + Z’



Example: Drell-Yan + Z’

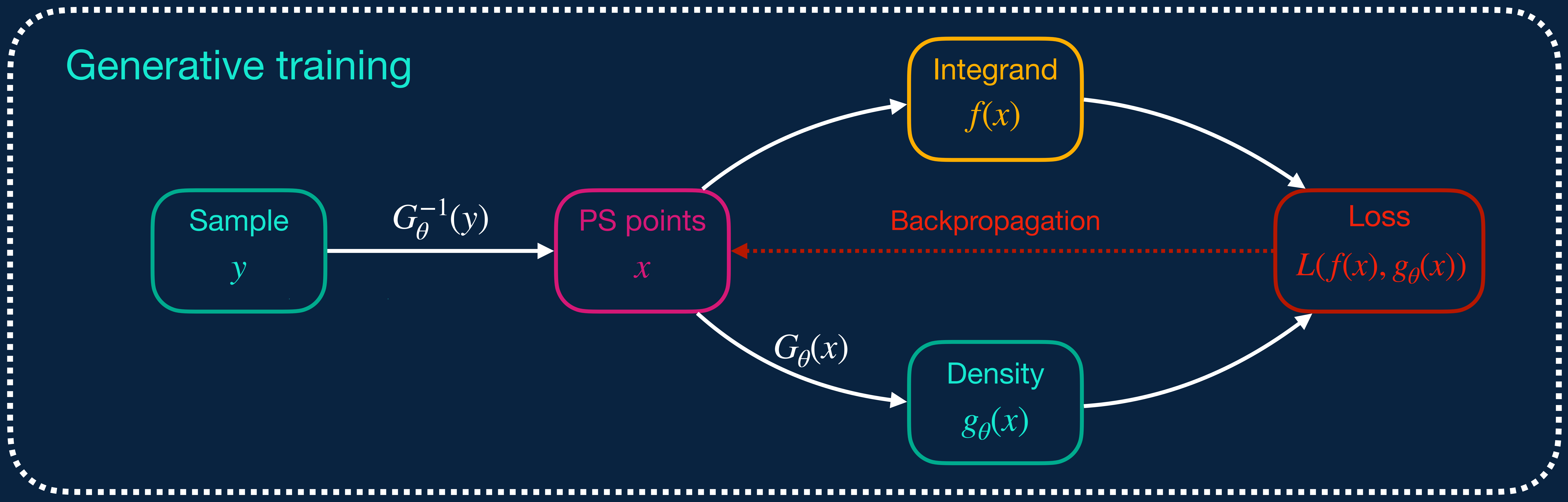


Example: Drell-Yan + Z'

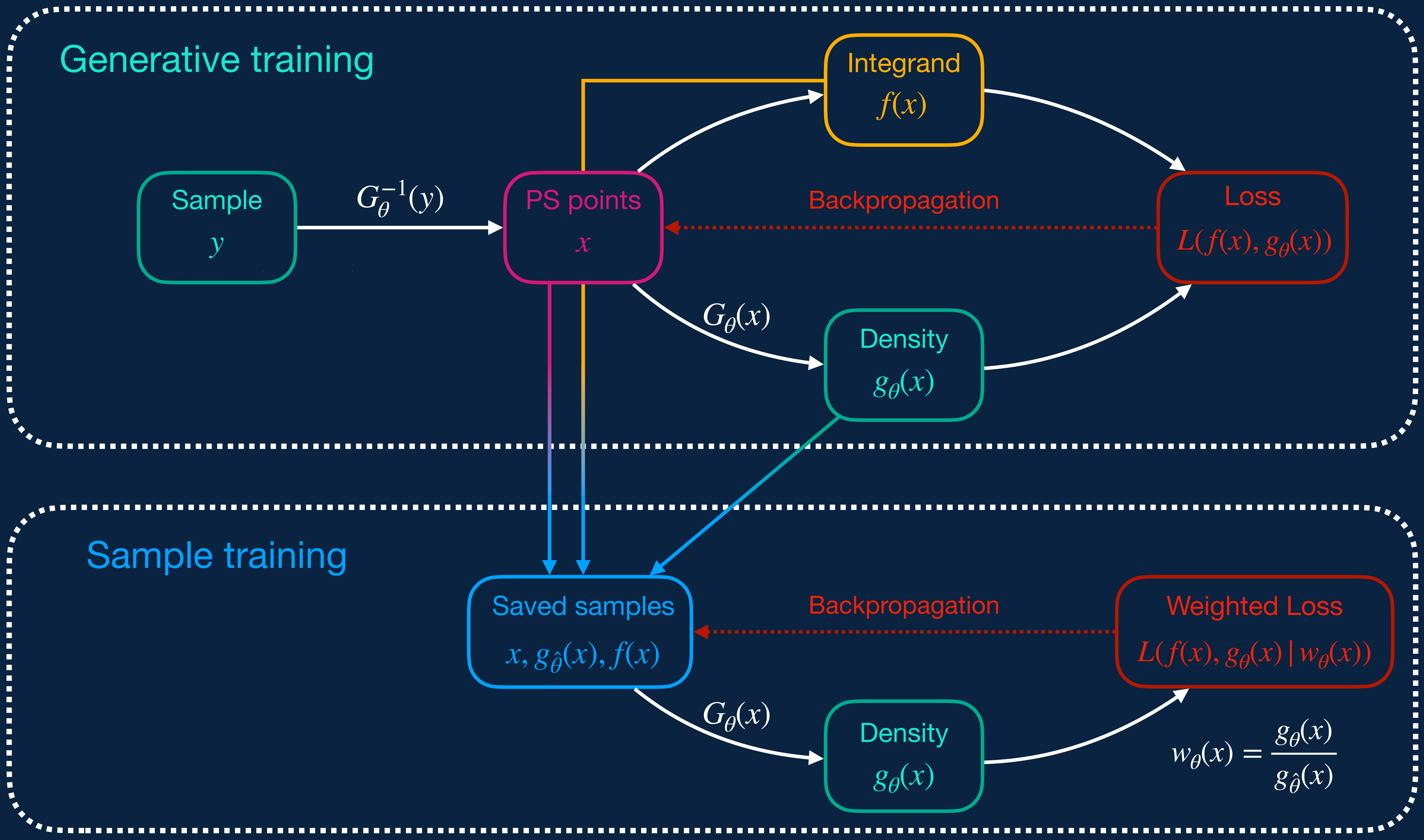


MadNIS

Two-Stage Training

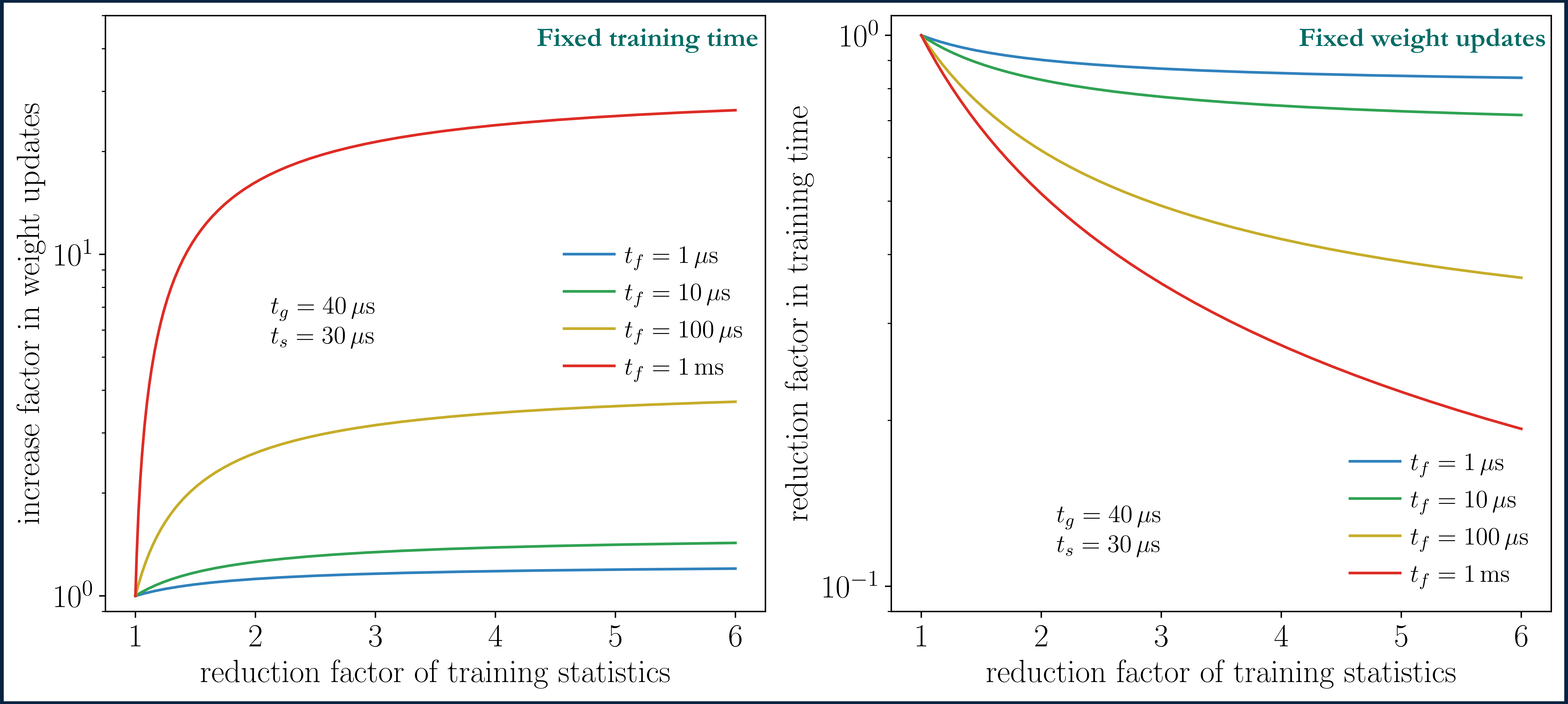


Generative training
Sample
y
$G_\theta^{-1}(y)$
PS points
x
Integrand
$f(x)$
Backpropagation
Loss
$L(f(x), g_\theta(x))$
$G_\theta(x)$
Density
$g_\theta(x)$



Generative training
Sample
y
$G_\theta^{-1}(y)$
PS points
x
Integrand
$f(x)$
Backpropagation
Loss
$L(f(x), g_\theta(x))$
$G_\theta(x)$
Density
$g_\theta(x)$
Sample training
Saved samples
$x, g_{\hat{\theta}}(x), f(x)$
Backpropagation
Weighted Loss
$L(f(x), g_\theta(x) \,|\, w_\theta(x))$
$G_\theta(x)$
Density
$g_\theta(x)$
$w_\theta(x) = \frac{g_\theta(x)}{g_{\hat{\theta}}(x)}$

Two-Stage Training



Summary and Outlook

Summary

- Channel mappings are important
- Multi-channel is more efficient and trained simultanously with the flow
- Conditional flow and 2-stage training reduce computational overhead

Summary and Outlook

Summary

- Channel mappings are important
- Multi-channel is more efficient and trained simultanously with the flow
- Conditional flow and 2-stage training reduce computational overhead

Outlook

- Implementation of MadNIS into MadGraph
- Test performance on real LHC examples: (eg. multi-leg, NLO, complicated cuts, …)
- Make matrix elements run on the GPU and differentiable