UCLouvain

MadNIS

Neural networks for multi-channel integration in MadGraph

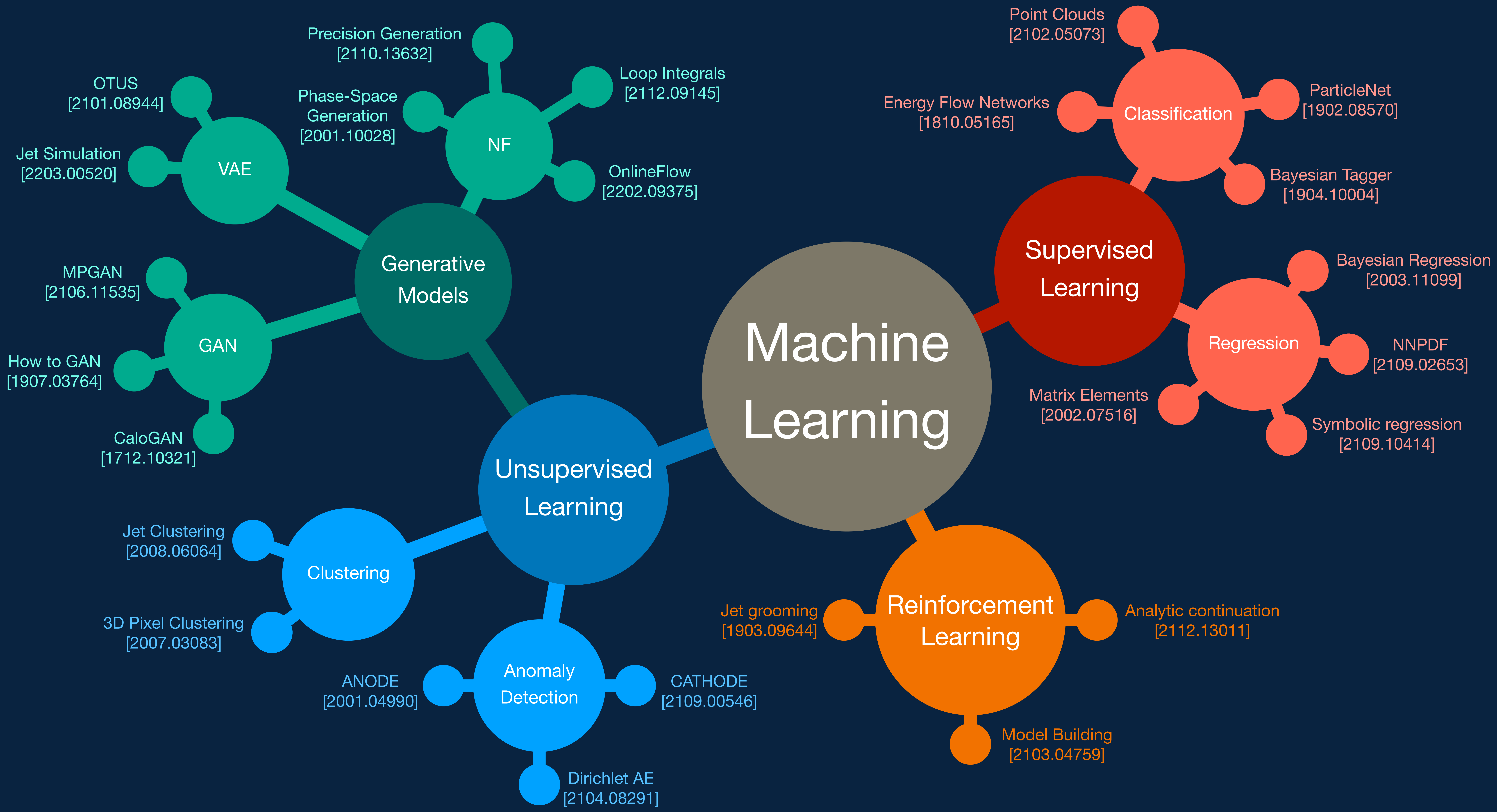
MCnet Meeting - Graz 2022

Ramon Winterhalder — UC Louvain

Why talk about machine learning?

because

- rich toolbox of algorithms to develop expressive and flexible models for science
- fast development of new methods and algorithms in the past years
- promising applications in both theory and experiment
- large interest in HEP community: *IML, ML4Jets, MCnet, workshops,..*



Machine Learning
Supervised Learning
Classification
Point Clouds [2102.05073]
ParticleNet [1902.08570]
Energy Flow Networks [1810.05165]
Bayesian Tagger [1904.10004]
Regression
Bayesian Regression [2003.11099]
NNPDF [2109.02653]
Matrix Elements [2002.07516]
Symbolic regression [2109.10414]
Reinforcement Learning
Jet grooming [1903.09644]
Analytic continuation [2112.13011]
Model Building [2103.04759]
Unsupervised Learning
Clustering
Jet Clustering [2008.06064]
3D Pixel Clustering [2007.03083]
Anomaly Detection
ANODE [2001.04990]
CATHODE [2109.00546]
Dirichlet AE [2104.08291]
Generative Models
VAE
OTUS [2101.08944]
Jet Simulation [2203.00520]
NF
Precision Generation [2110.13632]
Loop Integrals [2112.09145]
Phase-Space Generation [2001.10028]
OnlineFlow [2202.09375]
GAN
MPGAN [2106.11535]
How to GAN [1907.03764]
CaloGAN [1712.10321]

Focus today

Event Generation

Theory predictions in HEP

$$\mathcal{A}_{\lambda,c,\dots}(p_a, p_b | p_1, \dots, p_n) : \mathbb{M} \to \mathbb{C}$$

Quantum numbers:
spin, colour charge etc.

Kinematics:
Momenta in Minkowski space, masses, etc.

$$\sigma = \frac{1}{\text{flux}} \sum_{a.b} \int \mathrm{d}x_a \mathrm{d}x_b\, f(x_a) f(x_b) \int \mathrm{d}\Phi_n \left\langle |\mathcal{A}(p_a, p_b | p_1, \dots, p_n)|^2 \right\rangle$$

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

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

Quantum numbers:
spin, colour charge etc.

Kinematics:
Momenta in Minkowski space, masses, etc.

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

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

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

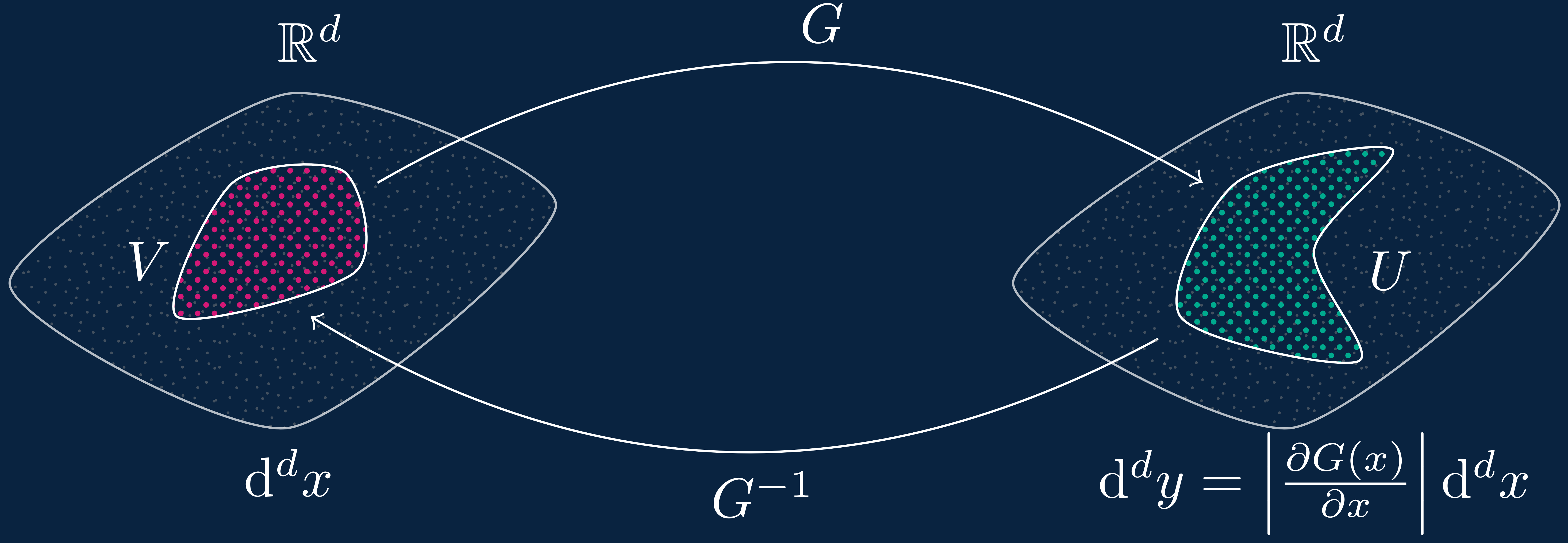
Monte Carlo Integration

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

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

Wanted

$$\sigma_I \to 0$$

Monte Carlo Integration

Standard Monte Carlo integration

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

Importance Sampling

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

Mapping

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

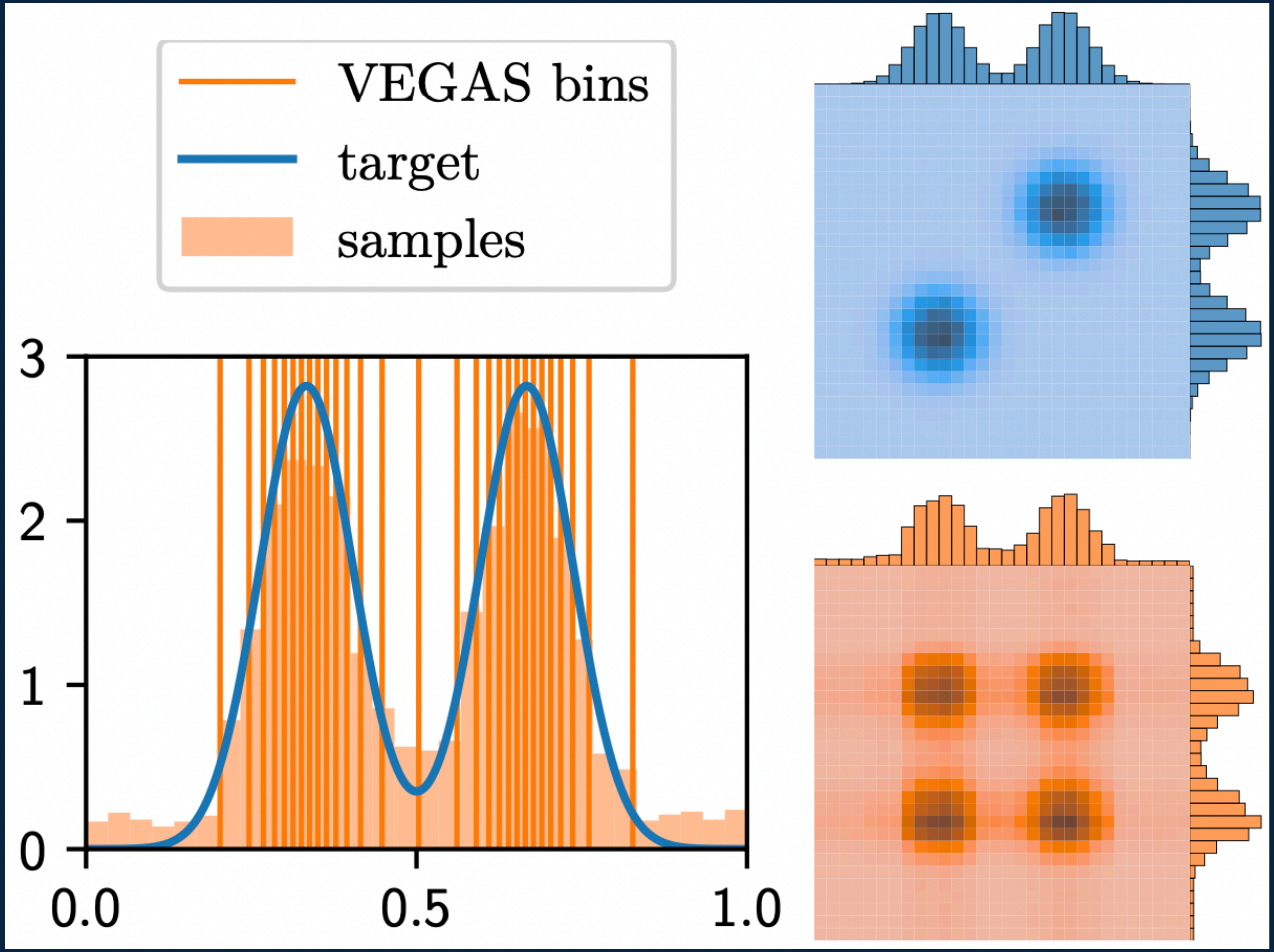
Wanted

$\sigma_I \to 0$

Needs

$f/g \approx$ const

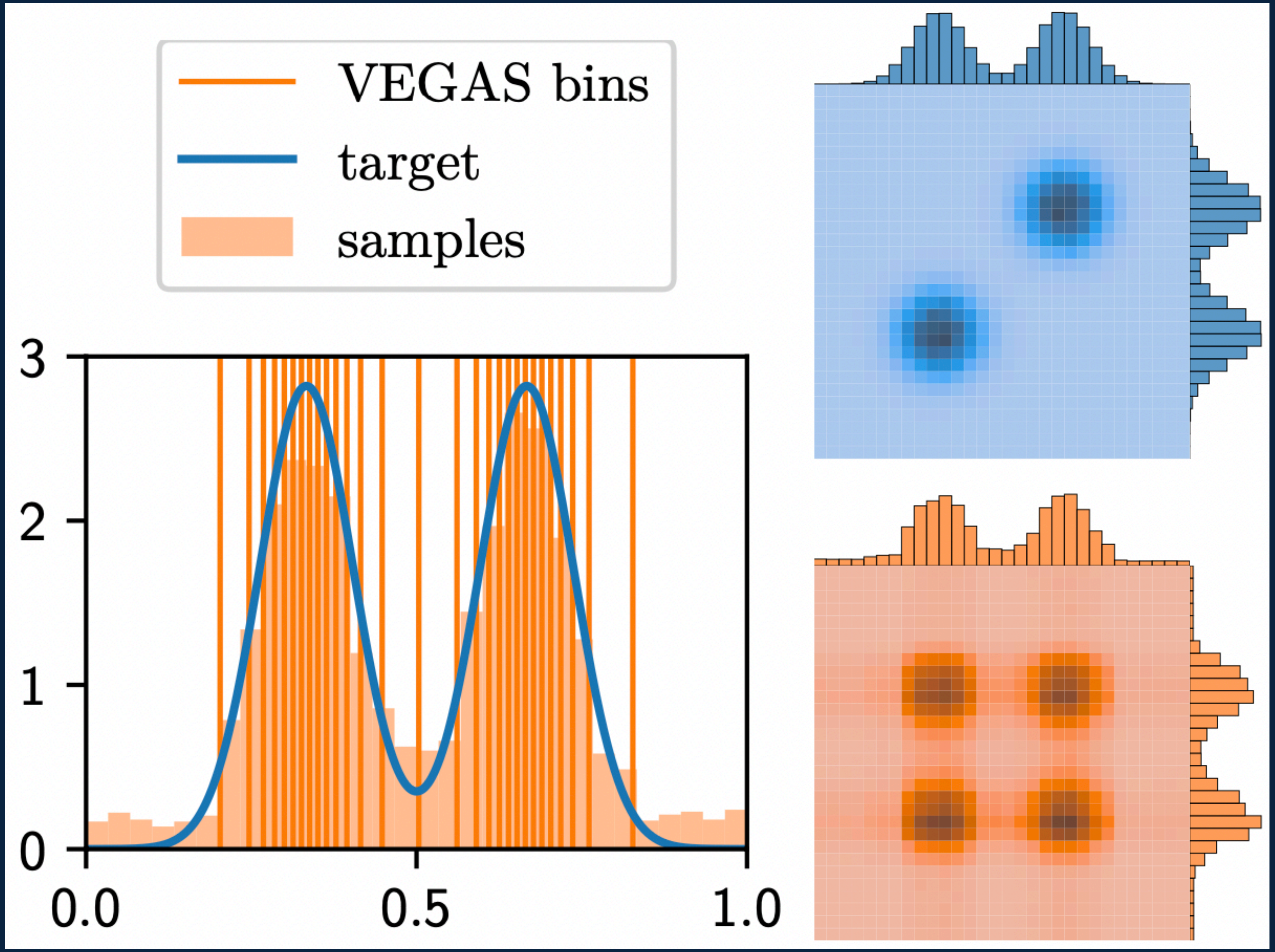
Importance sampling - VEGAS



Why not VEGAS for everything?

- High-dim and rich peaking → slow convergence
- If peaks are not aligned with grid axes → "phantom peaks"

Importance sampling - VEGAS



Why not VEGAS for everything?

- High-dim and rich peaking
  → slow convergence
- If peaks are not aligned with grid axes → "phantom peaks"

Quality score?

- Unweighting efficiency

$$\varepsilon = \frac{\langle w \rangle}{w_{\text{max}}}, \qquad w = \frac{f}{g}$$

Importance sampling - NN

Using a NN

- 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 - NN

Using a NN

- Unbinned and no grids
  → no “phantom peaks”
- Bijectivity not guaranteed
  → training unstable
- Numerical Jacobians
  → slow training and evaluation

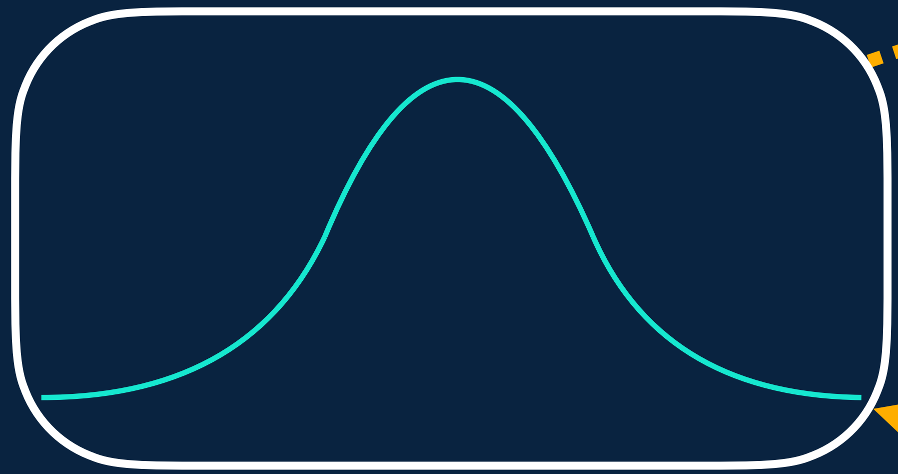
[1707.00028, 1810.11509, 2009.07819]

Using a Flow instead

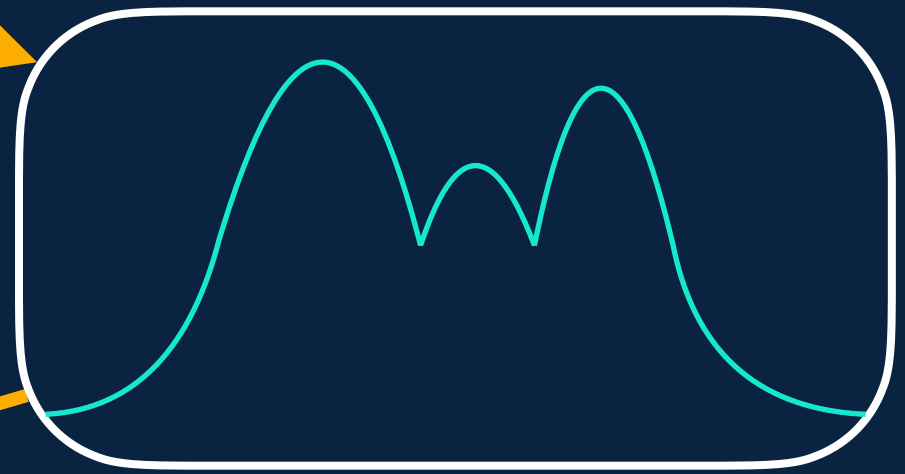
- Invertibility
  → bijective mapping
- tractable Jacobians
  → fast training and evaluation

[2001.05478, 2001.05486, 2001.10028, 2005.12719, 2112.09145]

Normalizing Flow



$$\log p_y(y) = \log p_x(x) + \log \left| \frac{\partial G(x)}{\partial x} \right|$$



Multi-Channel Monte Carlo

Standard Multi-Channel Integration → $g(x) = \sum_i \alpha_i\, g_i(x)$ **, with** $\sum_i \alpha_i = 1$

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

Channel Mappings

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

Multi-Channel Monte Carlo

Standard Multi-Channel Integration → $g(x) = \sum_i \alpha_i \, g_i(x)$, **with** $\sum_i \alpha_i = 1$

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

Channel Mappings

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

MadGraph Multi-Channel Integration → **only use** $\sum_i \beta_i(x) = 1$

Multi-Channel Monte Carlo

Standard Multi-Channel Integration → $g(x) = \sum_i \alpha_i\, g_i(x)$ **, with** $\sum_i \alpha_i = 1$

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

Channel Mappings

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

MadGraph Multi-Channel Integration → **only use** $\sum_i \beta_i(x) = 1$

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

Multi-Channel Monte Carlo

Standard Multi-Channel Integration → $g(x) = \sum_i \alpha_i \, g_i(x)$ **, with** $\sum_i \alpha_i = 1$

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

Channel Mappings

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

MadGraph Multi-Channel Integration → **only use** $\sum_i \beta_i(x) = 1$

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

Connection

$$\beta_i(x) = \alpha_i \frac{g_i(x)}{g(x)}$$

Neural Multi-Channel Monte Carlo

Multi-Channel Importance Sampling

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

Neural Multi-Channel Monte Carlo

Multi-Channel Importance Sampling

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

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

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

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

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

Possible Improvements?

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

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

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

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

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

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

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

Further Improvements?

- Two-stage training
- Variance-weighted training
- Overflow channels
- Bayesian neural networks
- More expressive transformations: FFJORD, CubicSplines, LaSeR,…

[1810.0136] [1906.02145] [2106.00792]

Neural Multi-Channel Monte Carlo

Neural Channel Weights

MadNIS

Two-Stage Training

MadNIS

Neural Channel Weights

Neural Channel Weights

Channel network

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

Neural Channel Weights

Channel network

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

Normalization

$$\alpha_i^{\phi}(x) \to \hat{\alpha}_i^{\phi}(x) = \frac{\exp \alpha_i^{\phi}(x)}{\sum_i \exp \alpha_i^{\phi}(x)}$$

Neural Channel Weights

Channel network

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

Normalization

$$\alpha_i^\phi(x) \to \hat{\alpha}_i^\phi(x) = \frac{\exp \alpha_i^\phi(x)}{\sum_i \exp \alpha_i^\phi(x)}$$

MadGraph Channels

$$\beta_i(x) = \frac{|M_i(x)|^2}{\sum_j |M_j(x)|^2}$$

Neural Channel Weights

Channel network

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

Normalization

$$\alpha_i^\phi(x) \to \hat{\alpha}_i^\phi(x) = \frac{\exp \alpha_i^\phi(x)}{\sum_i \exp \alpha_i^\phi(x)}$$

MadGraph Channels

$$\beta_i(x) = \frac{|M_i(x)|^2}{\sum_j |M_j(x)|^2}$$

Residual channel network

$$\alpha_i^\phi(x) = \log \beta_i(x) + \varphi_i \cdot \Delta_i^\phi(x)$$

Neural Channel Weights

Channel network

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

Normalization

$$\alpha_i^\phi(x) \to \hat{\alpha}_i^\phi(x) = \frac{\exp \alpha_i^\phi(x)}{\sum_i \exp \alpha_i^\phi(x)}$$

MadGraph Channels

$$\beta_i(x) = \frac{|M_i(x)|^2}{\sum_j |M_j(x)|^2}$$

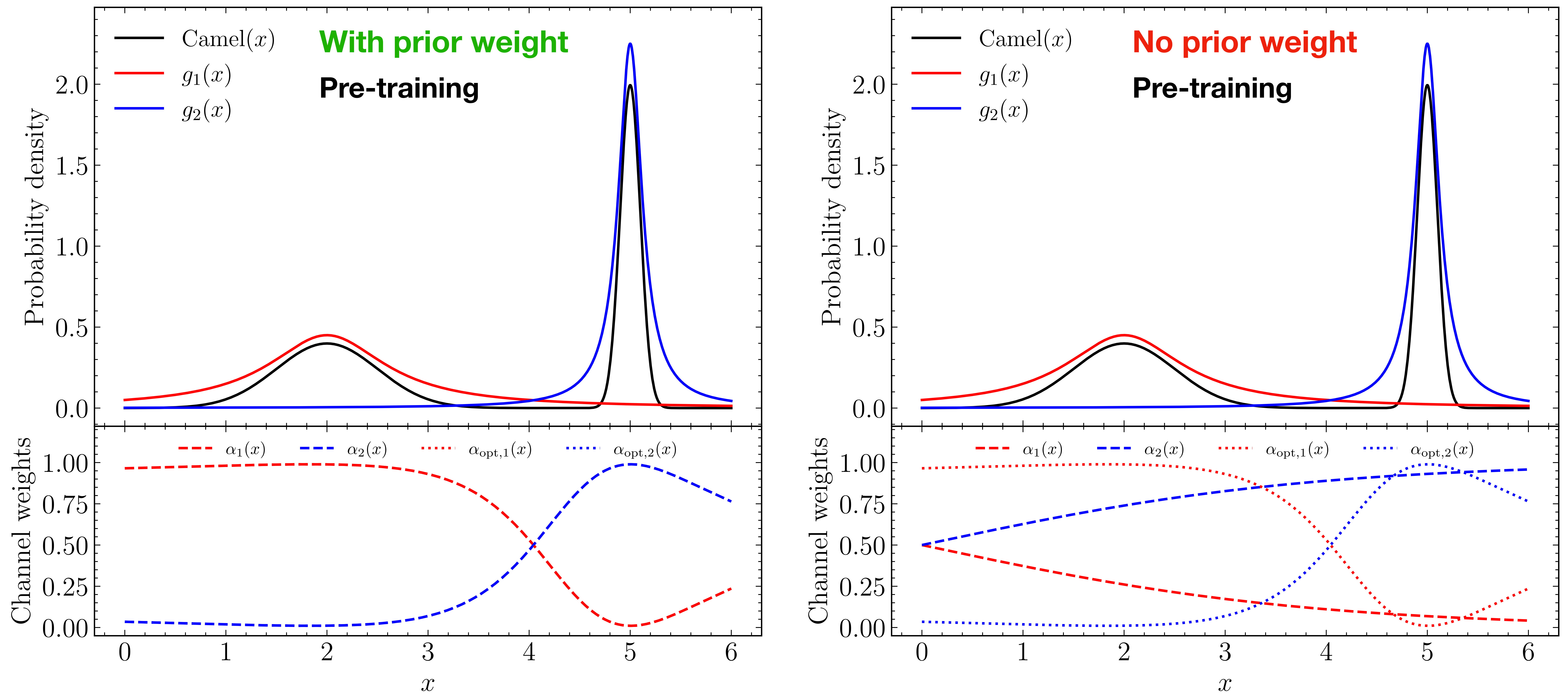
Residual channel network

$$\alpha_i^\phi(x) = \log \beta_i(x) + \varphi_i \cdot \Delta_i^\phi(x)$$

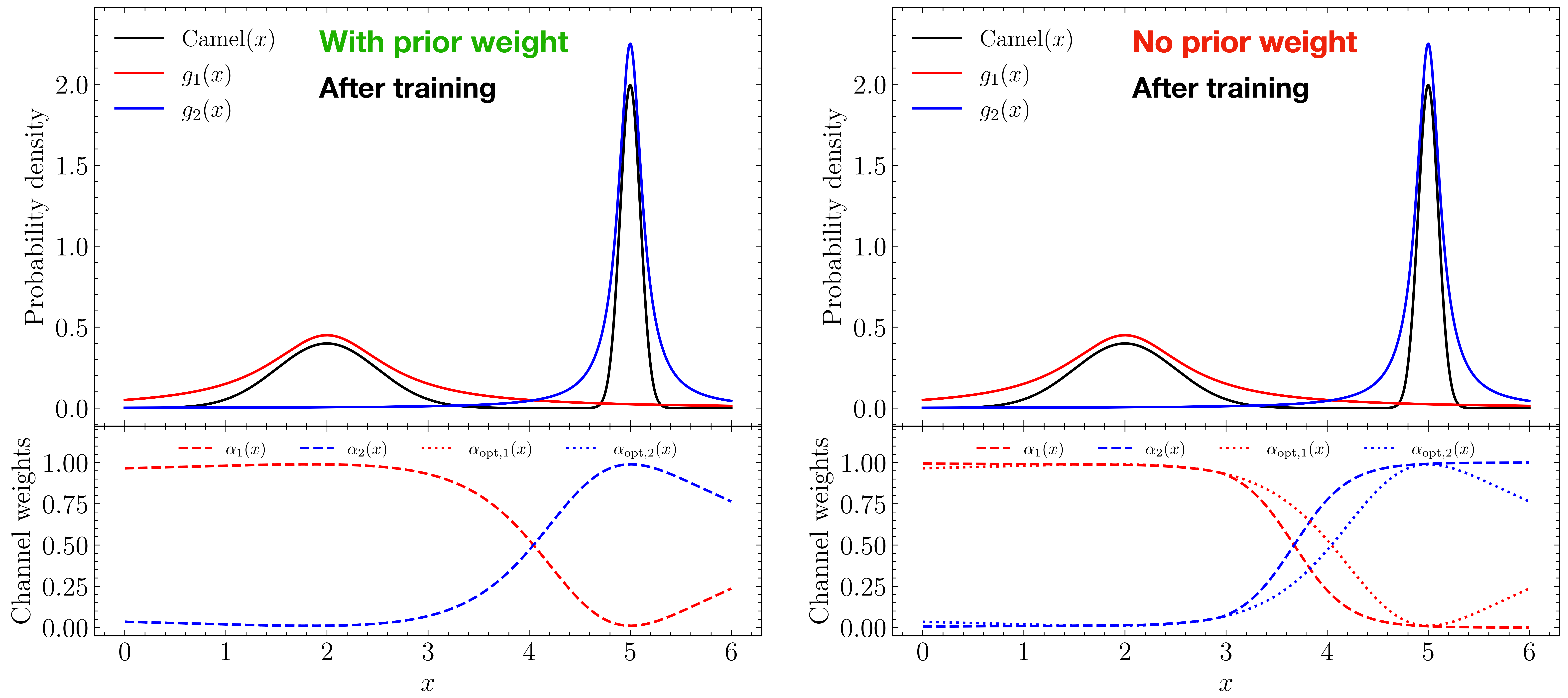
Normalization

$$\alpha_i^\phi(x) \to \hat{\alpha}_i^\phi(x) = \frac{\beta_i(x)\ \exp \varphi_i \cdot \Delta_i^\phi(x)}{\sum_j \beta_j(x)\ \exp \varphi_j \cdot \Delta_j^\phi(x)}$$

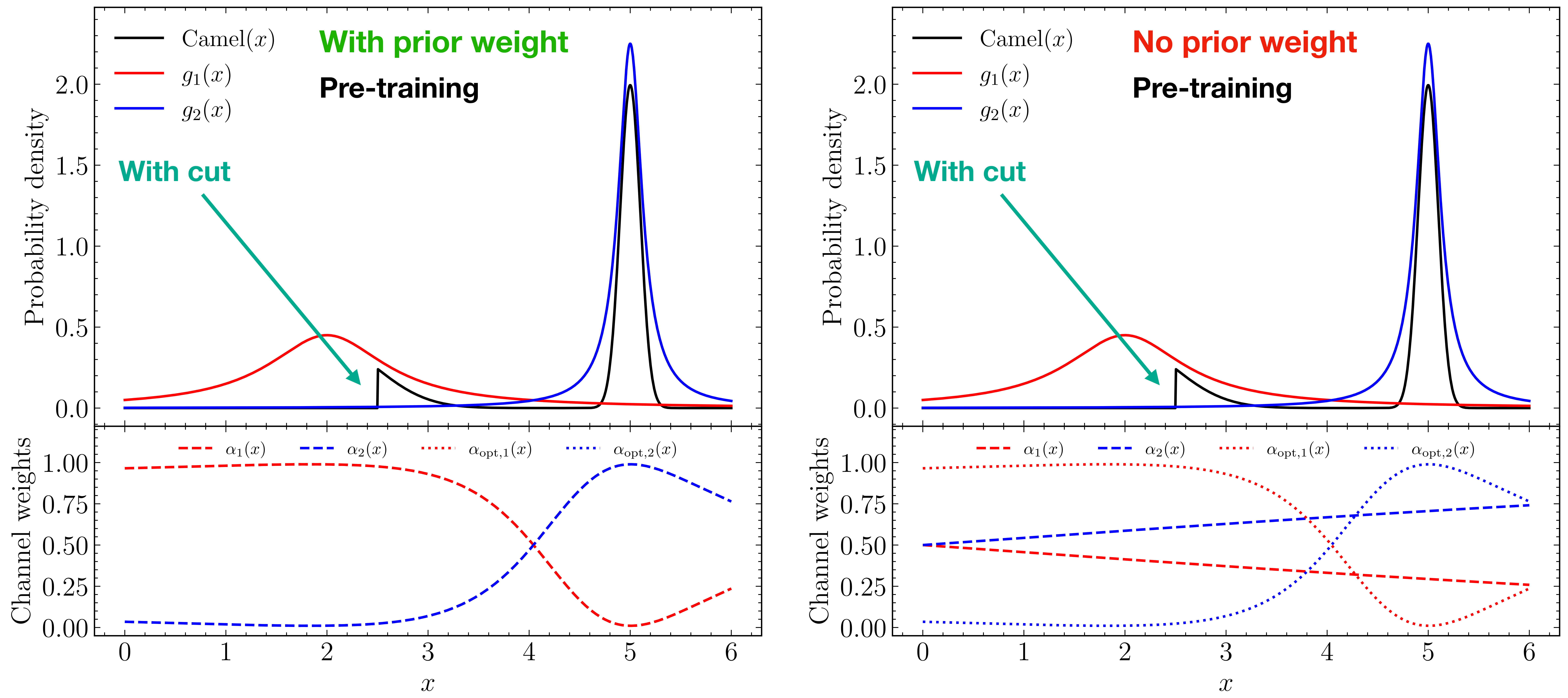
Neural Channel Weights



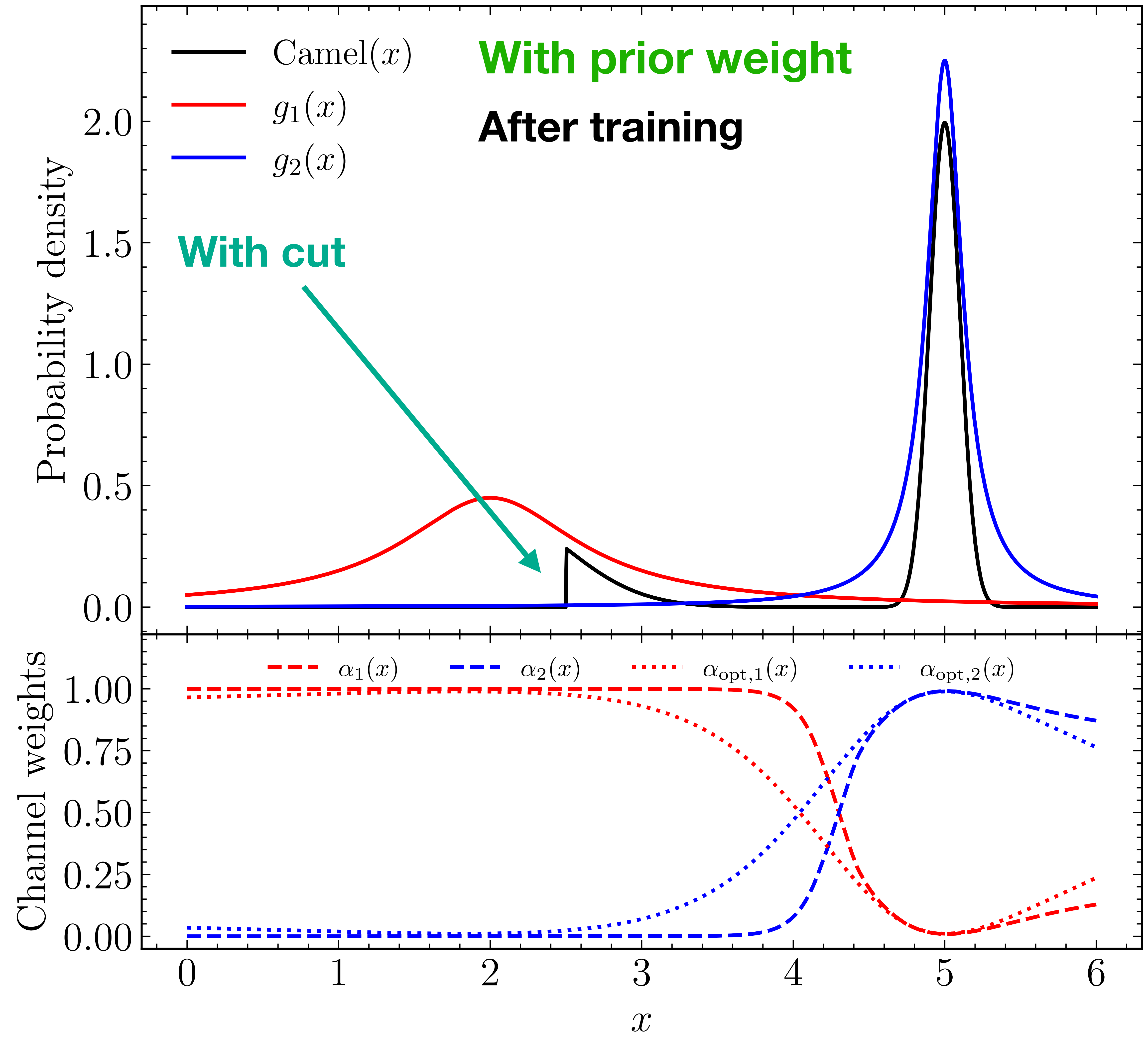
Neural Channel Weights

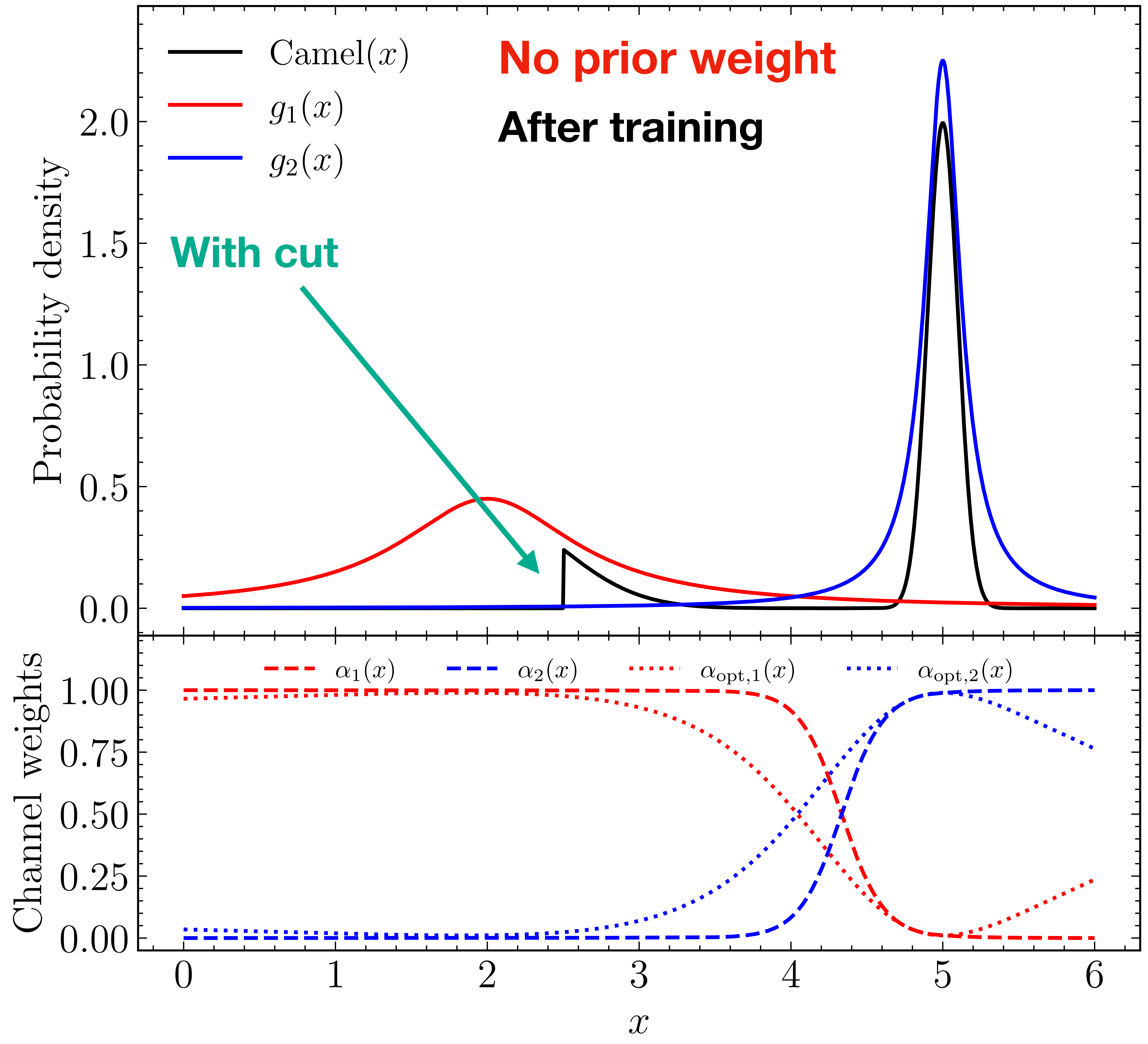


Neural Channel Weights



Neural Channel Weights





MadNIS

Two-Stage Training

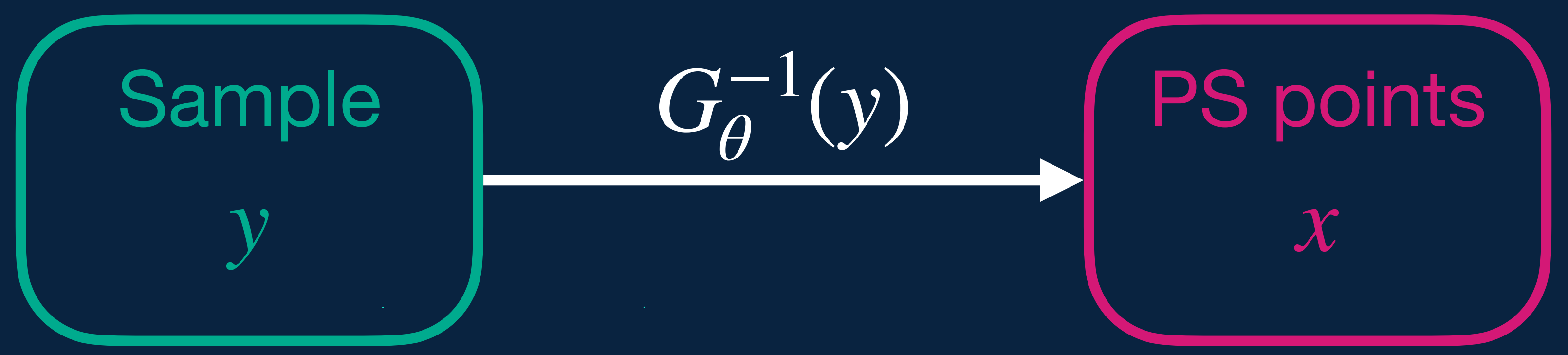
Generative training

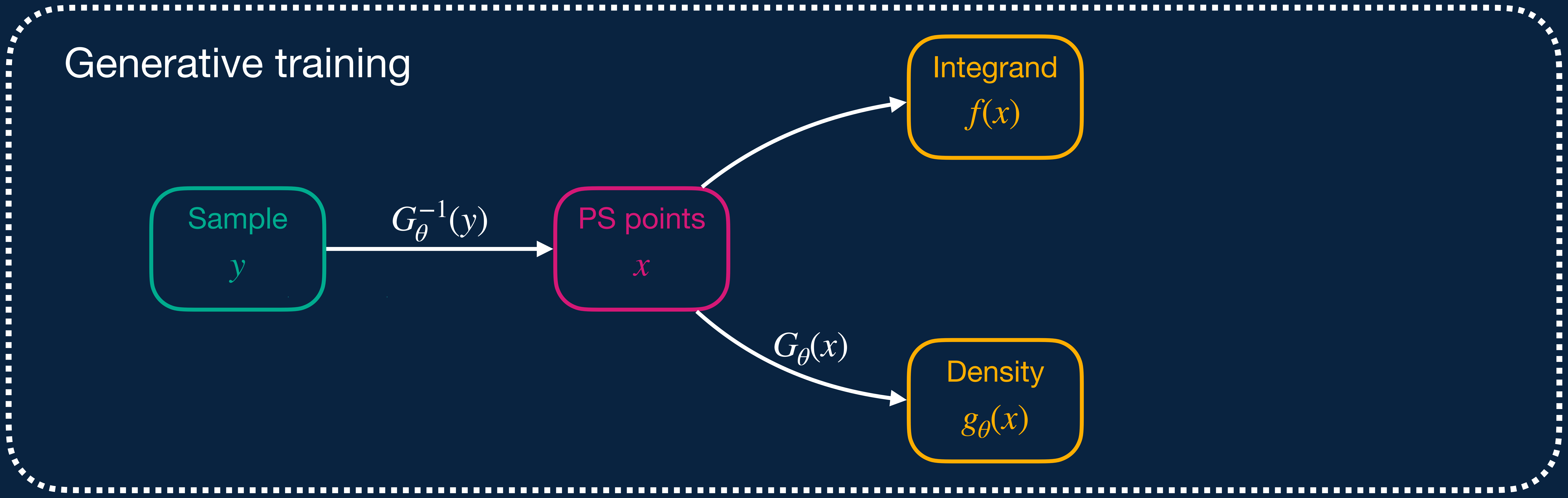
Generative training

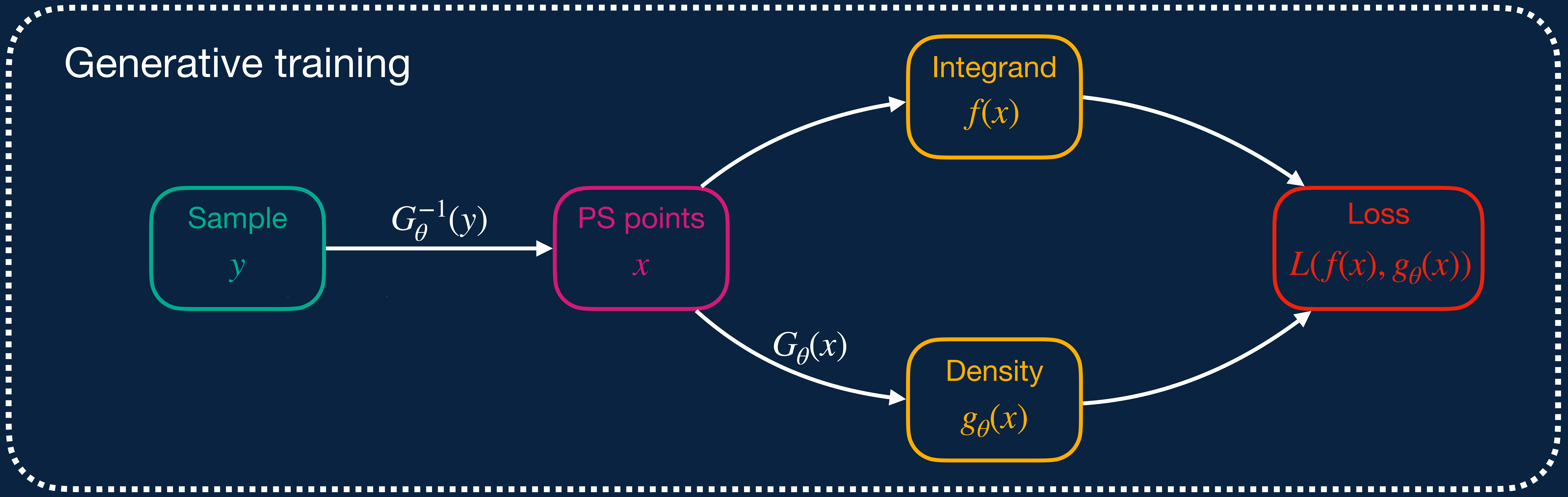
Sample

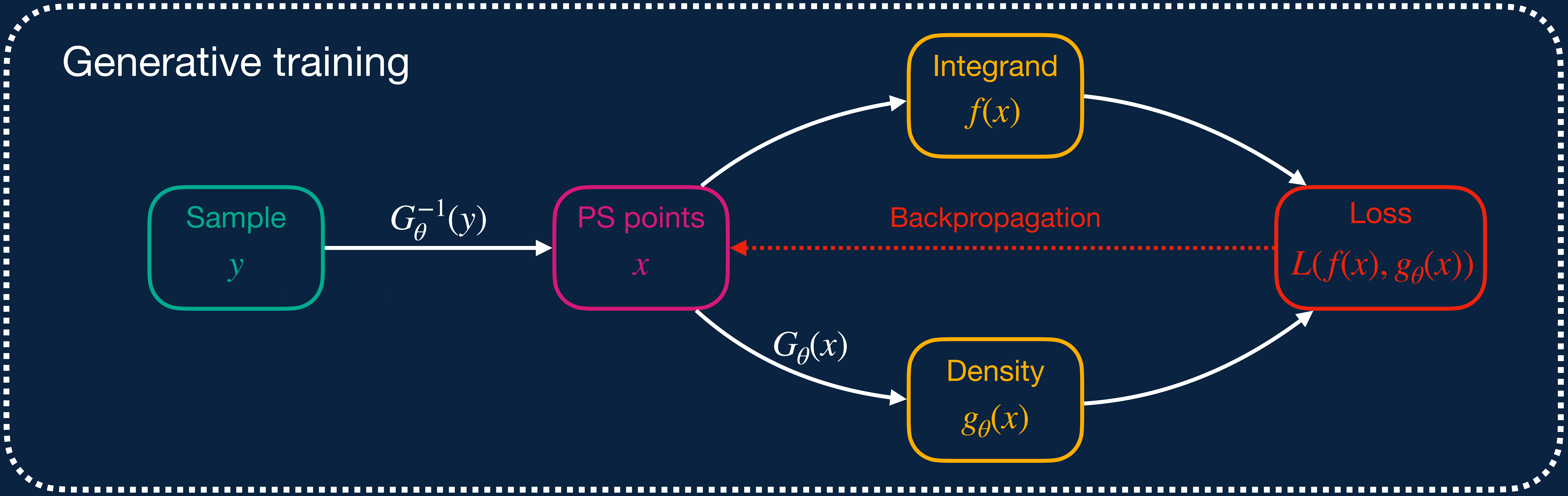
y

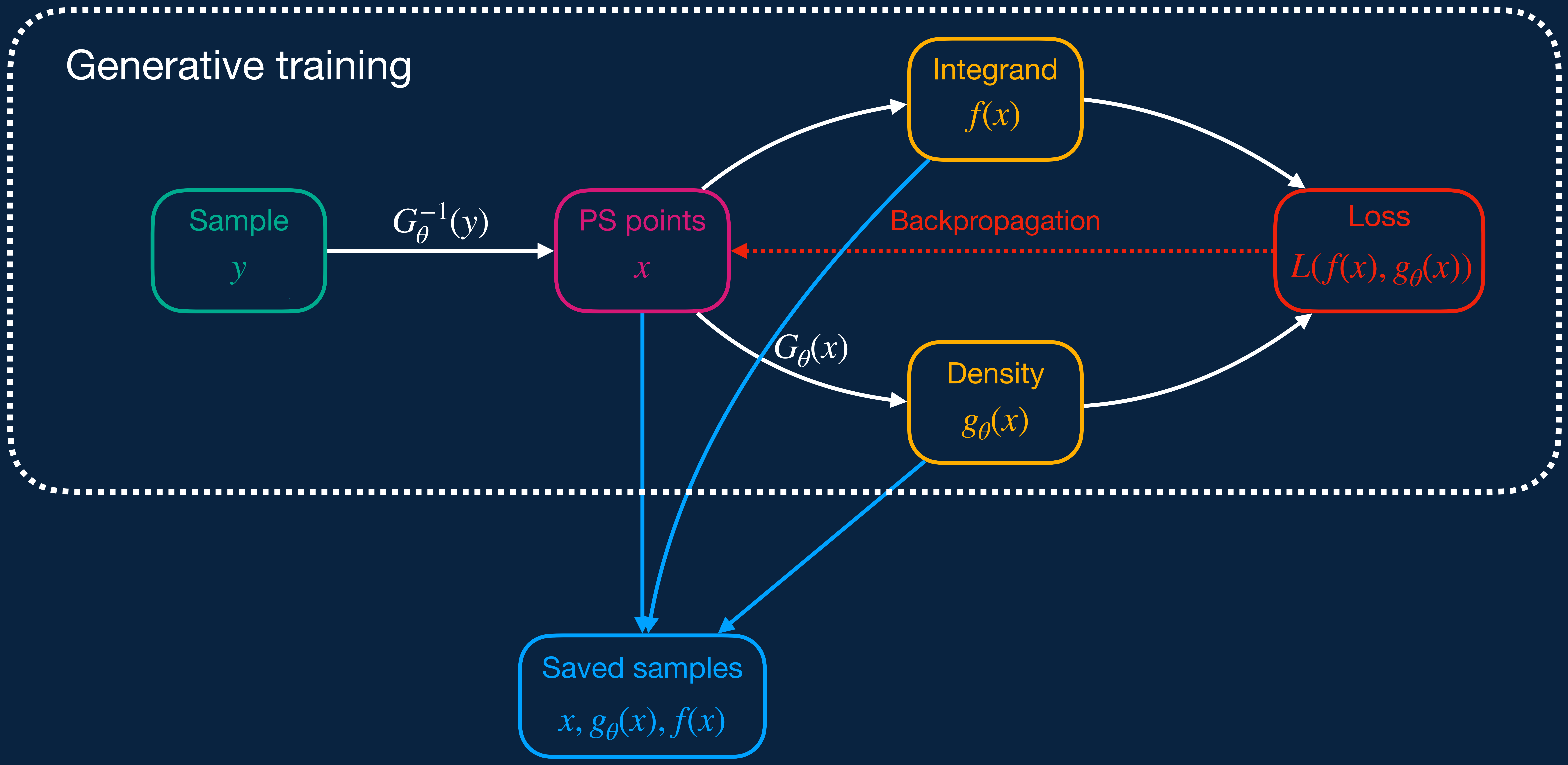
Generative training



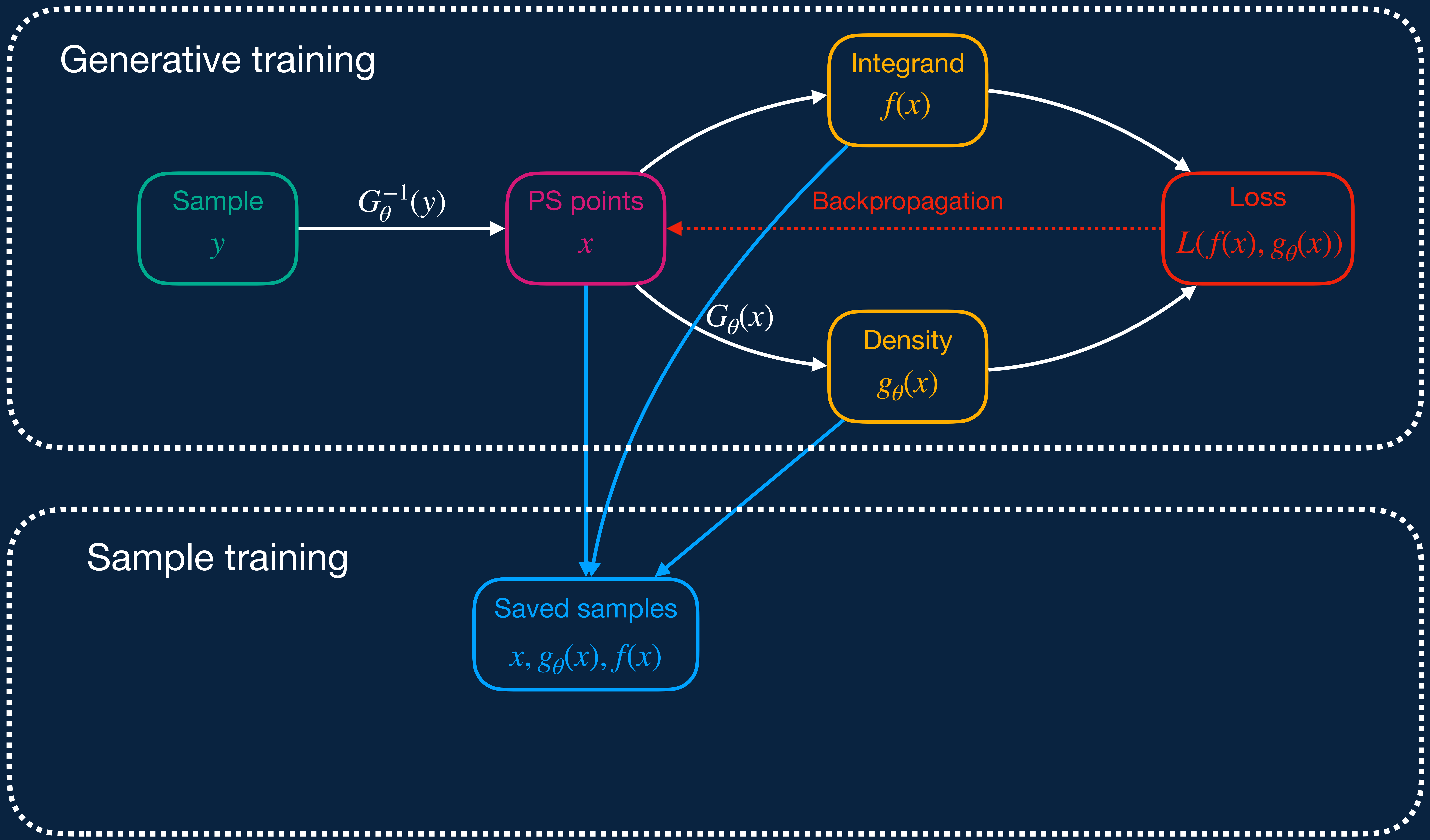

Generative training
Sample
y
$G_\theta^{-1}(y)$
PS points
x
Integrand
$f(x)$
$G_\theta(x)$
Density
$g_\theta(x)$


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

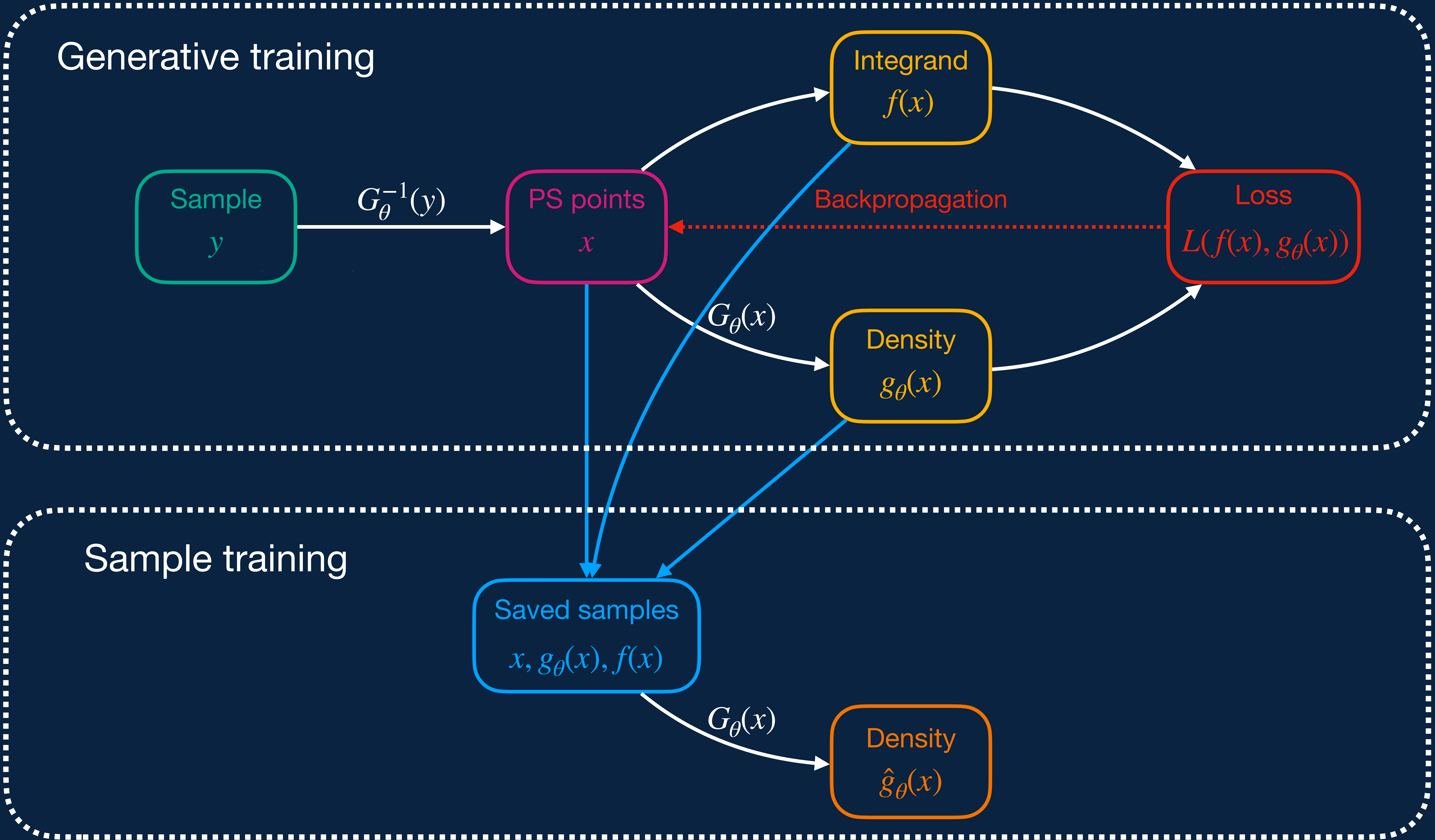

Generative training
Sample
y
$G_\theta^{-1}(y)$
PS points
x
Integrand
$f(x)$
Backpropagation
$G_\theta(x)$
Density
$g_\theta(x)$
Loss
$L(f(x), g_\theta(x))$

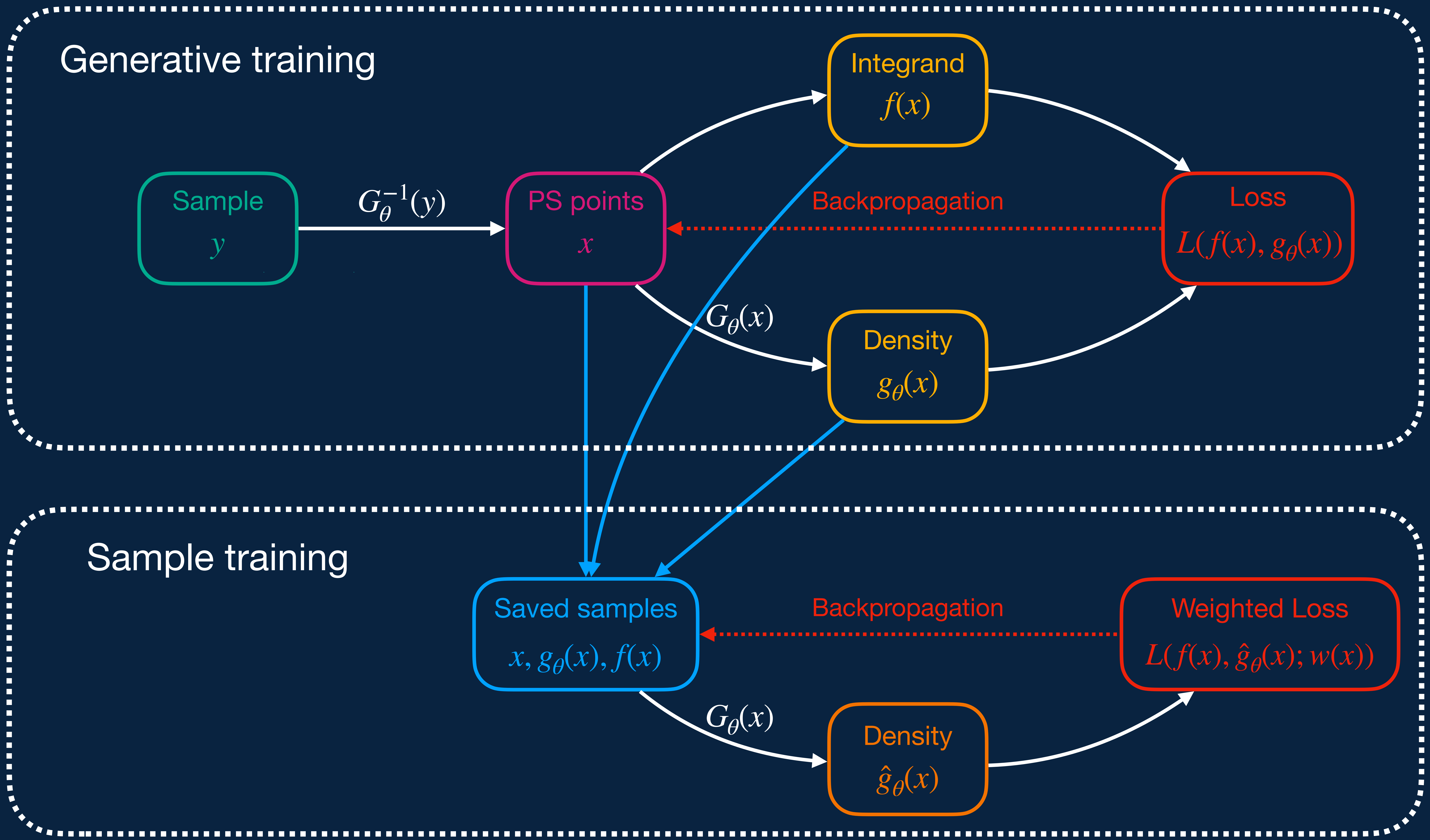


Generative training
Sample
y
$G_\theta^{-1}(y)$
PS points
x
Integrand
$f(x)$
Backpropagation
Loss
$L(f(x), g_\theta(x))$
$G_\theta(x)$
Density
$g_\theta(x)$
Saved samples
$x, g_\theta(x), f(x)$

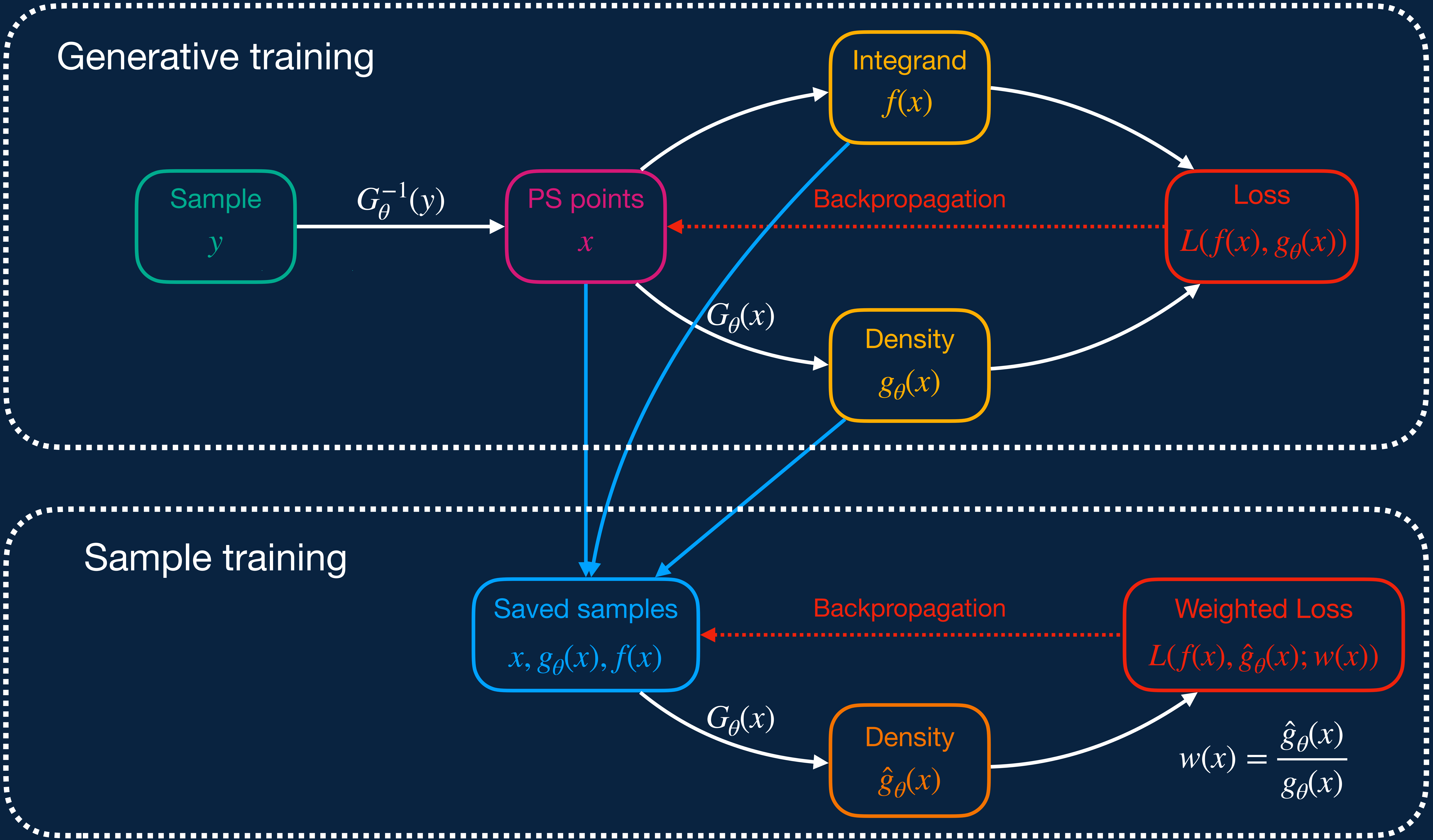


Generative training
Sample
y
$G_\theta^{-1}(y)$
PS points
x
Integrand
$f(x)$
Backpropagation
Loss
$L(f(x), g_\theta(x))$
$G_\theta(x)$
Density
$g_\theta(x)$
Sample training
Saved samples
$x, g_\theta(x), f(x)$


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
$x, g_\theta(x), f(x)$
$G_\theta(x)$
Density
$\hat{g}_\theta(x)$

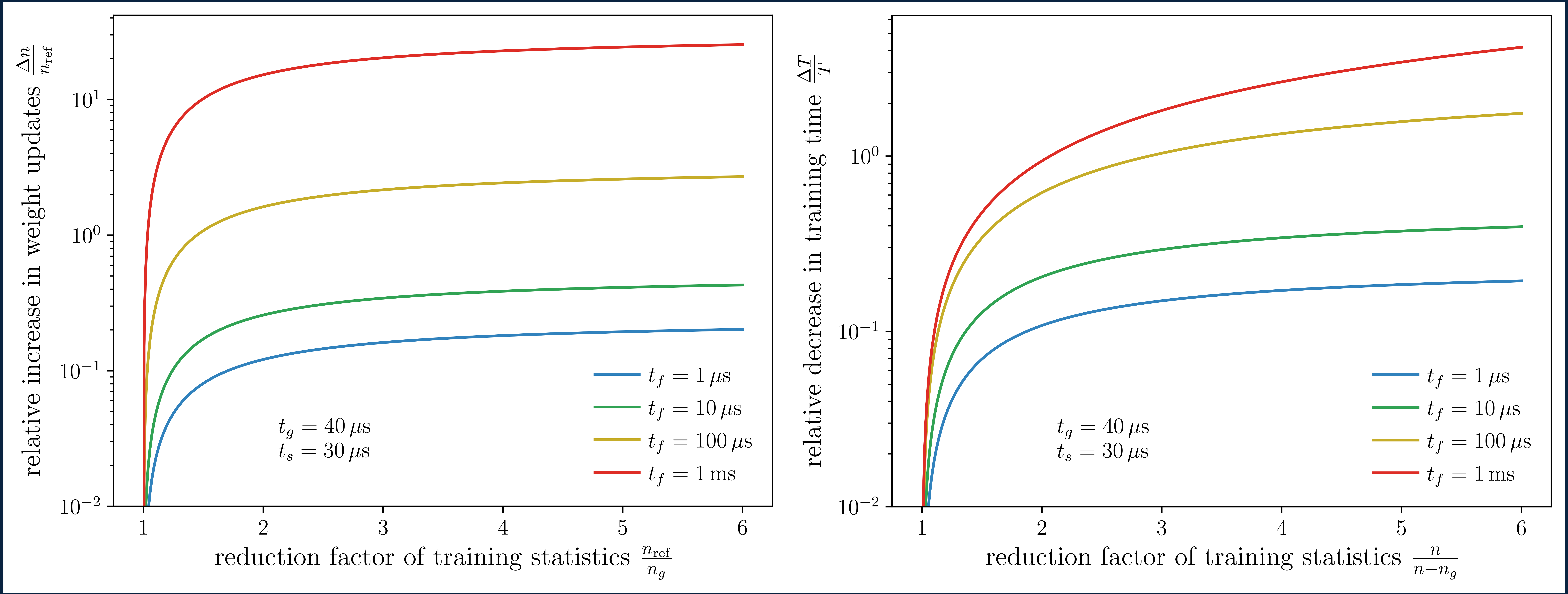


Generative training
Sample
y
$G_\theta^{-1}(y)$
PS points
x
Integrand
$f(x)$
Backpropagation
Loss
$L(f(x), g_\theta(x))$
$G_\theta(x)$
Density
$g_\theta(x)$
Sample training
Saved samples
$x, g_\theta(x), f(x)$
Backpropagation
Weighted Loss
$L(f(x), \hat{g}_\theta(x); w(x))$
$G_\theta(x)$
Density
$\hat{g}_\theta(x)$



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
$x, g_\theta(x), f(x)$
Backpropagation
Weighted Loss
$L(f(x), \hat{g}_\theta(x); w(x))$
$G_\theta(x)$
Density
$\hat{g}_\theta(x)$
$w(x) = \frac{\hat{g}_\theta(x)}{g_\theta(x)}$

Two-Stage Training



Summary and Outlook

Summary

- ML can reduce the error in Monte Carlo Integration
- New developments can improve performance and reduce computational overhead

Summary and Outlook

Summary

- ML can reduce the error in Monte Carlo Integration
- New developments can improve performance and reduce computational overhead
- Overview of ML in HEP: https://iml-wg.github.io/HEPML-LivingReview/

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

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.

Summary and Outlook

Summary

- ML can reduce the error in Monte Carlo Integration
- New developments can improve performance and reduce computational overhead
- Overview of ML in HEP: https://iml-wg.github.io/HEPML-LivingReview/

Outlook

- Test performance of flow on real LHC examples: (eg. multi-leg, NLO, complicated cuts, …)
- Possibly combine with other methods (eg. surrogate unweighting,…)
- Make matrix elements run on the GPU and differentiable

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

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.