Modern machine learning methods in HEP

Lecture II — Generative models for the LHC

UCLouvain

4th Baltic School of High-Energy Physics

Ramon Winterhalder — UCLouvain

Plan of attack

Wednesday

1. Introduction to Machine Learning

- Basic concepts of machine learning
- Classification and Regression
- Example: **Top Tagging, MadMiner**

Lecture I (90min)

Today

2. Generative Models for the LHC

- Normalizing flows
- CWoLA and Anomaly detection
- Examples: **MadNIS, CWoLA-Hunting,…**

Lecture II (90min)

Plan of attack

Wednesday

1. Introduction to Machine Learning

- Basic concepts of machine learning
- Classification and Regression
- Example: **Top Tagging, MadMiner**

Lecture I (90min)

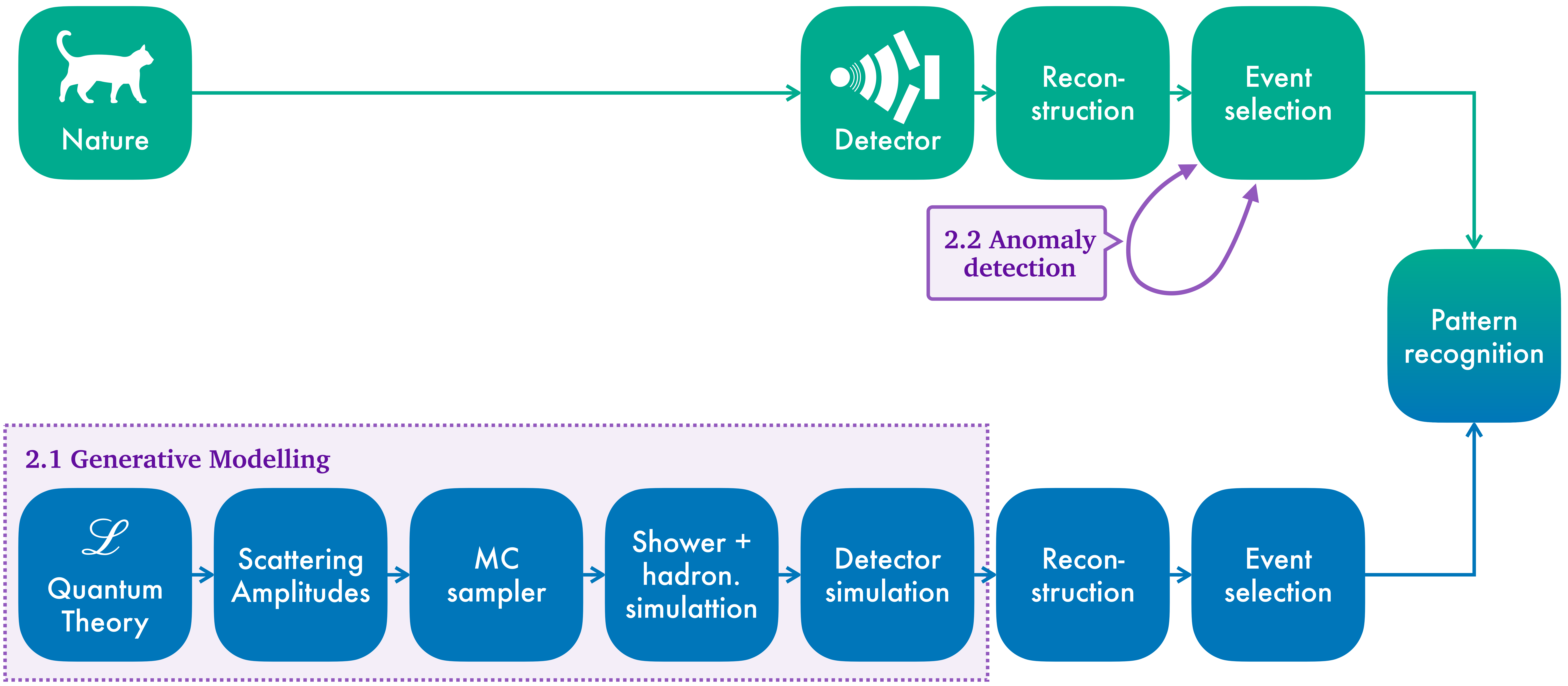
Today

2. Generative Models for the LHC

- Normalizing flows
- CWoLA and Anomaly detection
- Examples: **MadNIS, CWoLA-Hunting,…**

Lecture II (90min)

Reminder — LHC analysis + ML



Part II

Generative Models for the LHC

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?

What is a Generative Model?

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

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 $\{x\} \sim p_{\text{data}}(x)$

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
- Anomaly detection
- 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 $\{x\} \sim p_{\text{data}}(x)$

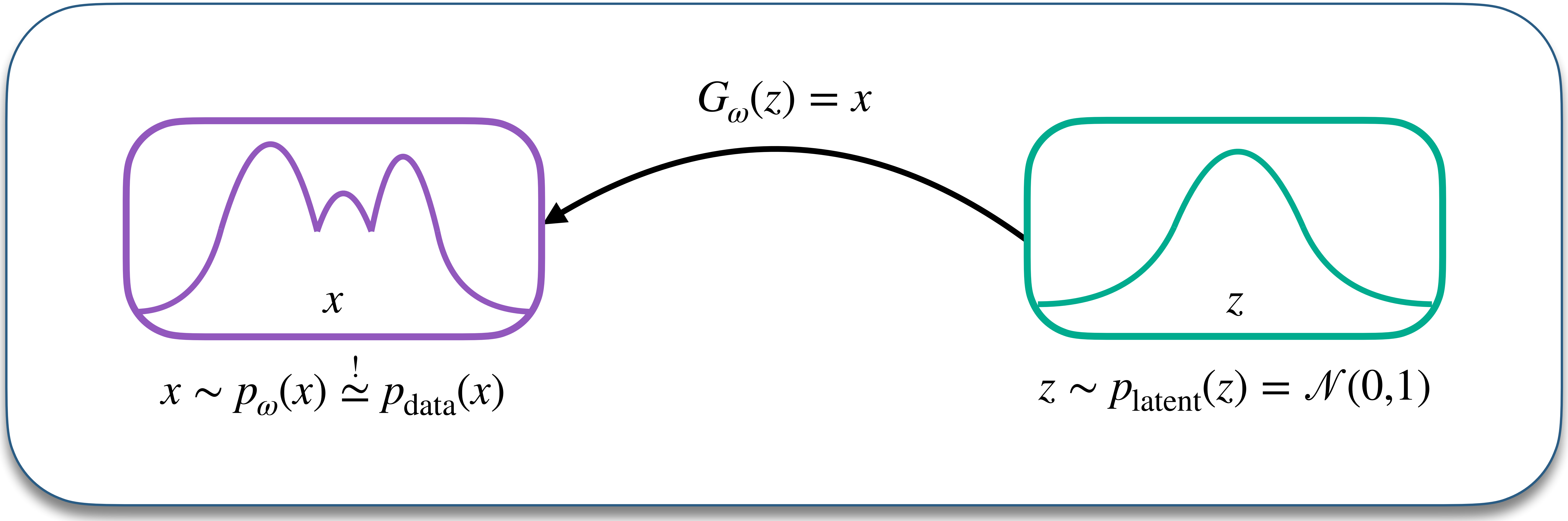
→ this is a stochastic (random) process (RNG)

→ needs "random" input

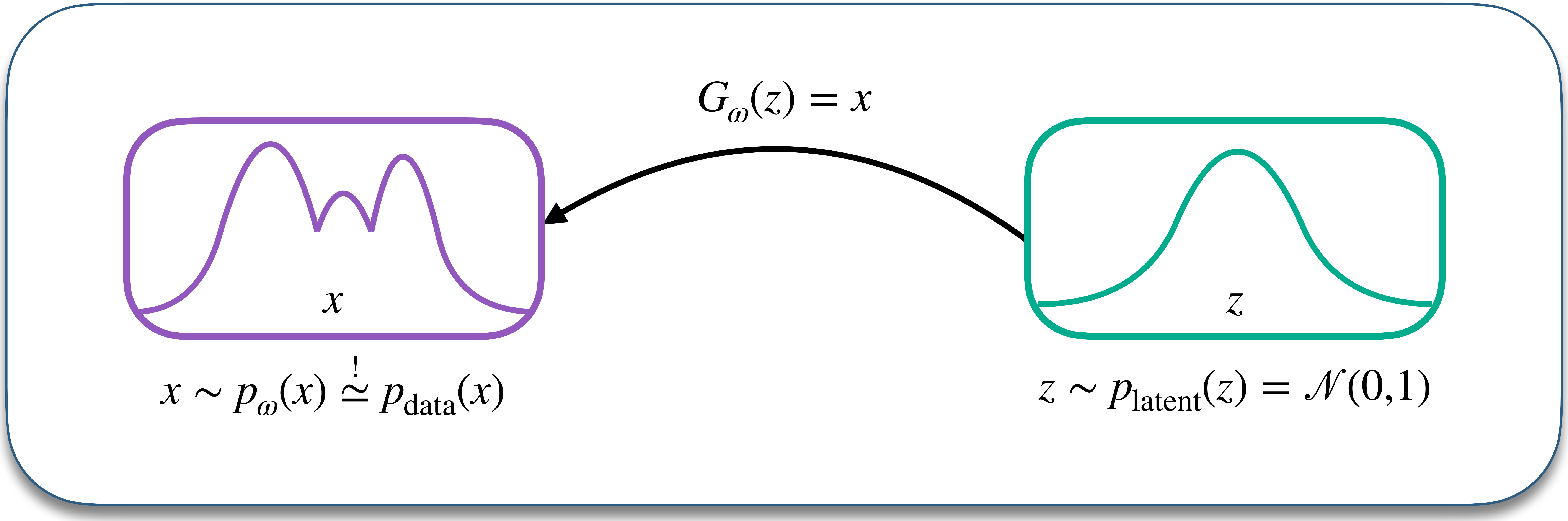
In **particle physics:**

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

What is a Generative Model?

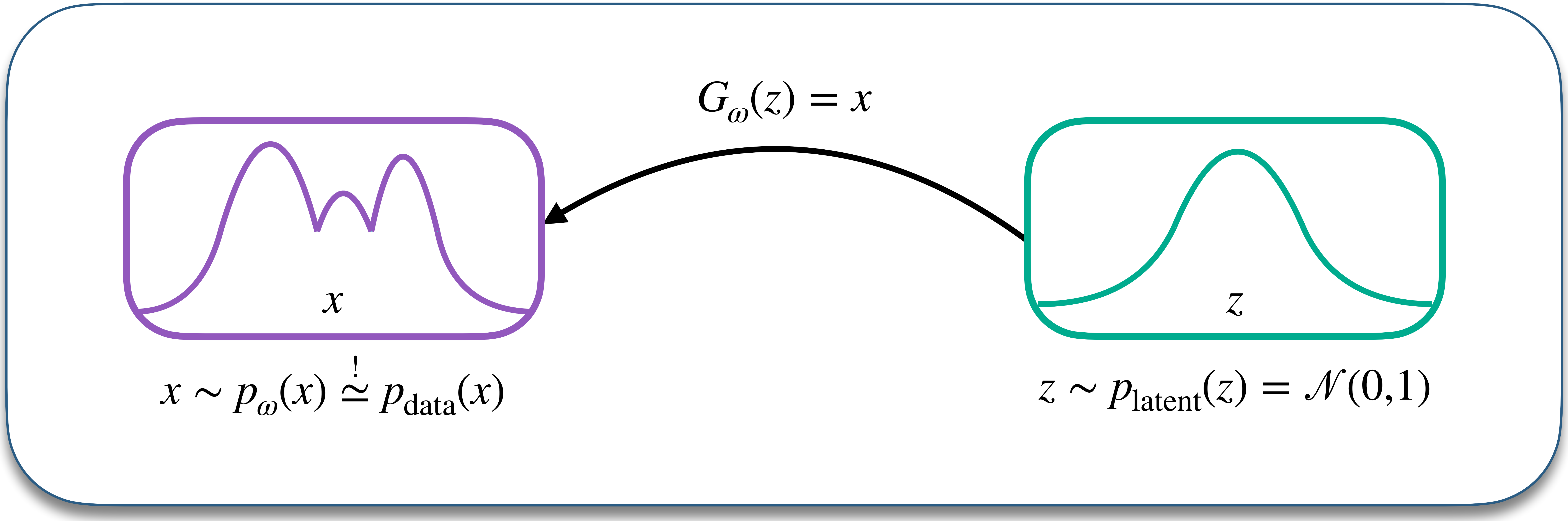


What is a Generative Model?



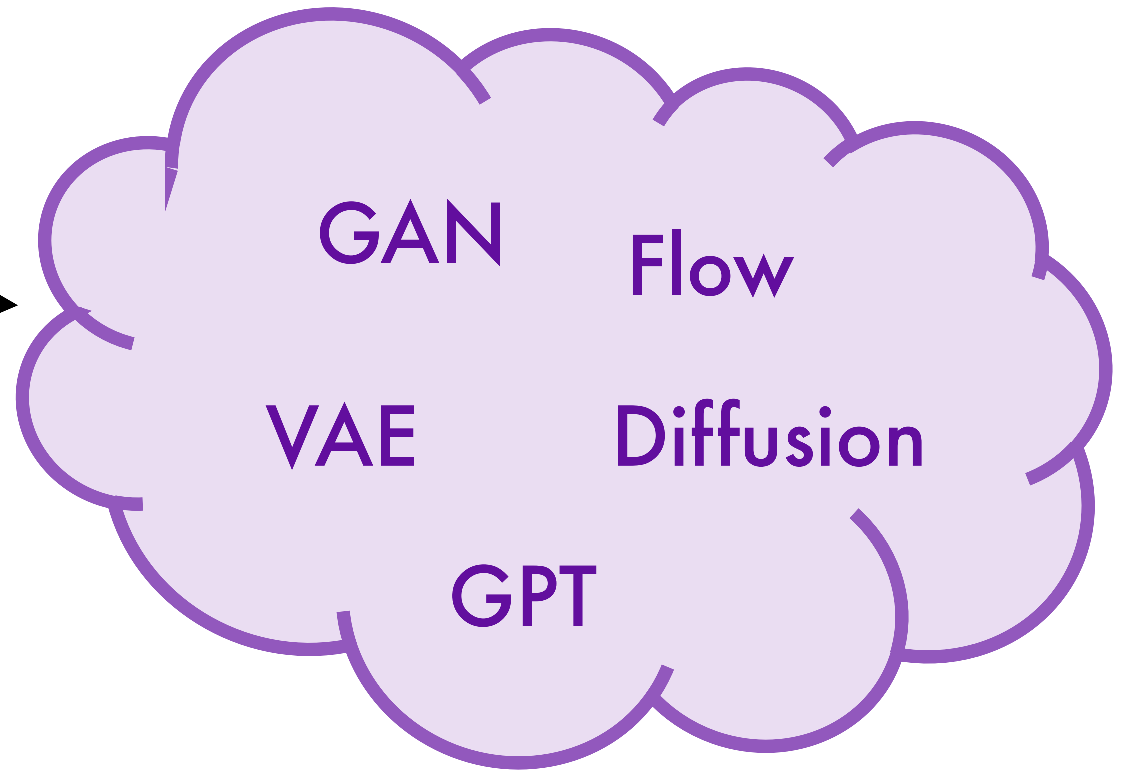
→ How to **construct** and **train** $G_\omega(z)$?

What is a Generative Model?



→ How to **construct** and **train** $G_\omega(z)$?

→ **Multiple types** of generative models



Types of deep generative models

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

Deep generative models

Variational Autoencoder

- β-VAE
- Hierarchical VAE
- VQ-VAE

↔

Diffusion Model

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

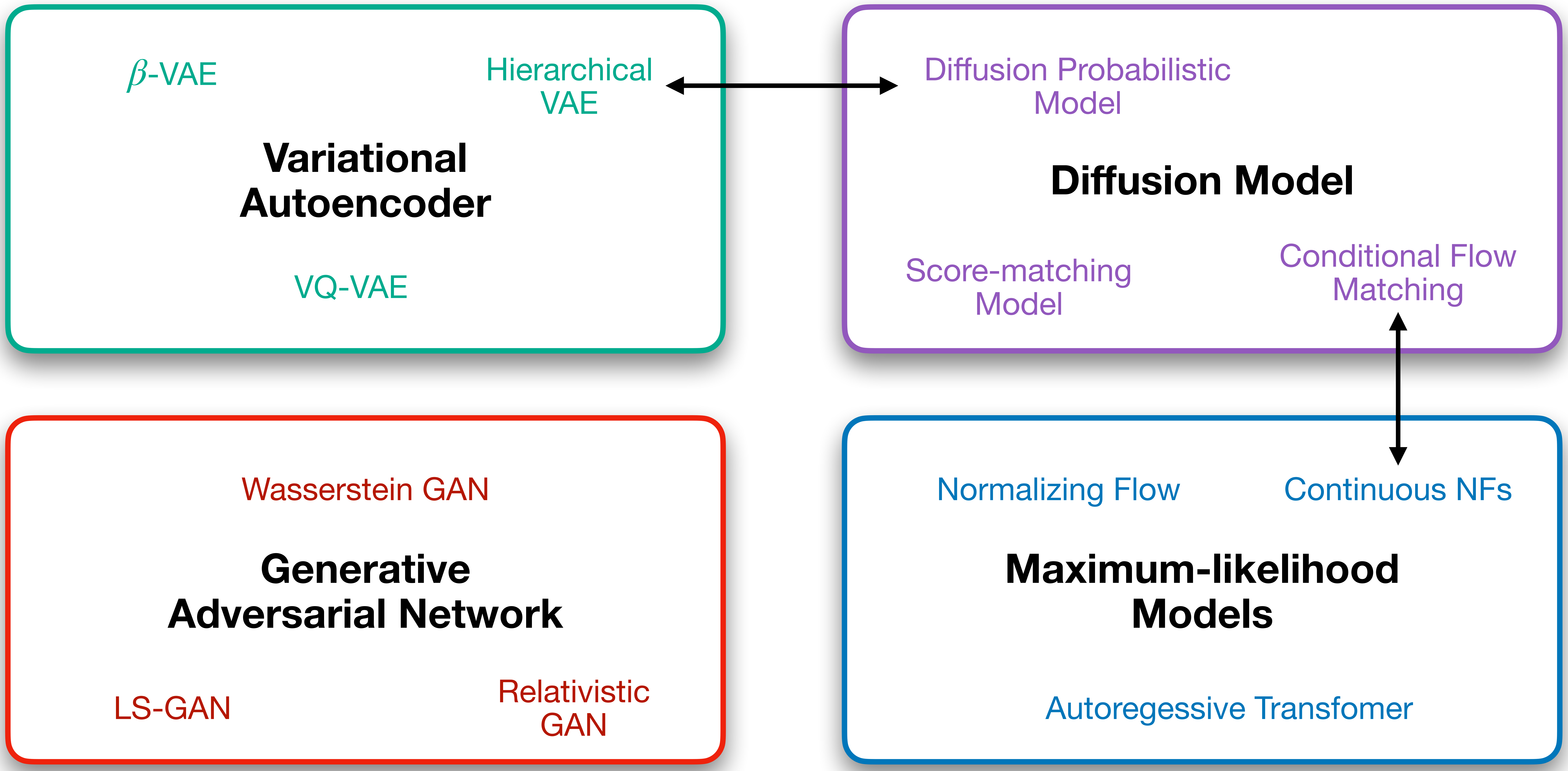
Generative Adversarial Network

- Wasserstein GAN
- LS-GAN
- Relativistic GAN

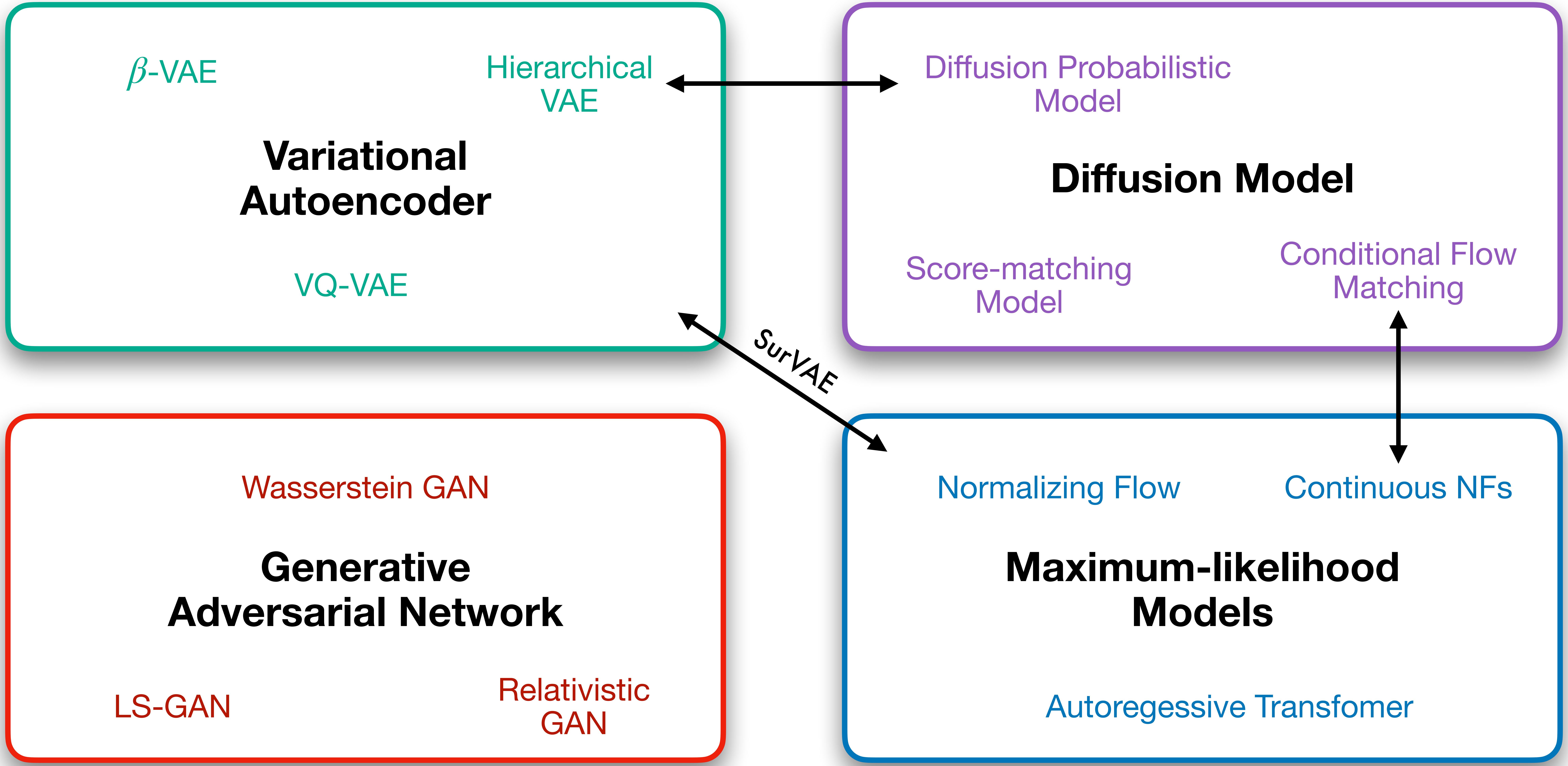
Maximum-likelihood Models

- Normalizing Flow
- Continuous NFs
- Autoregessive Transfomer

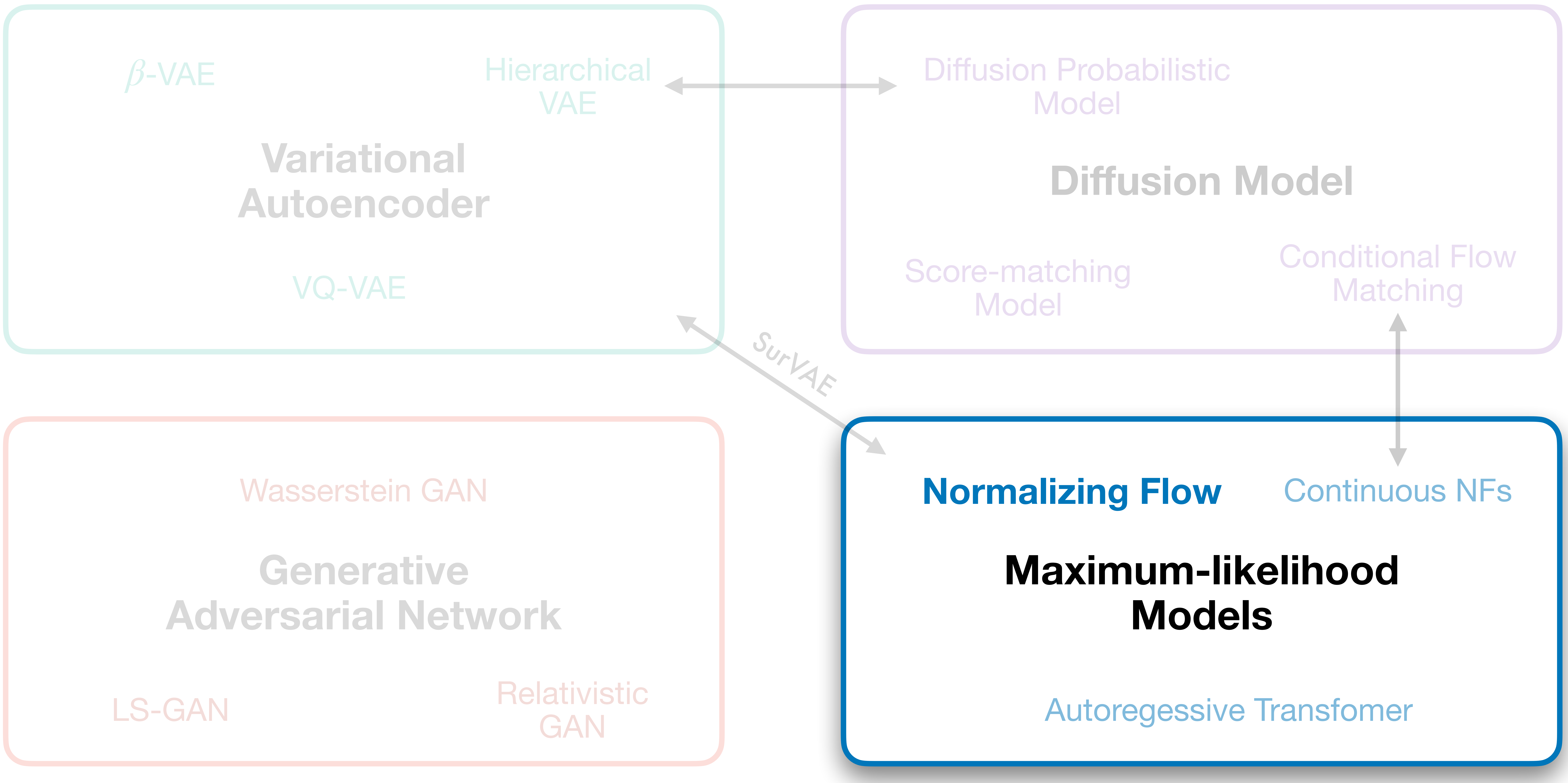
Deep generative models



Deep generative models

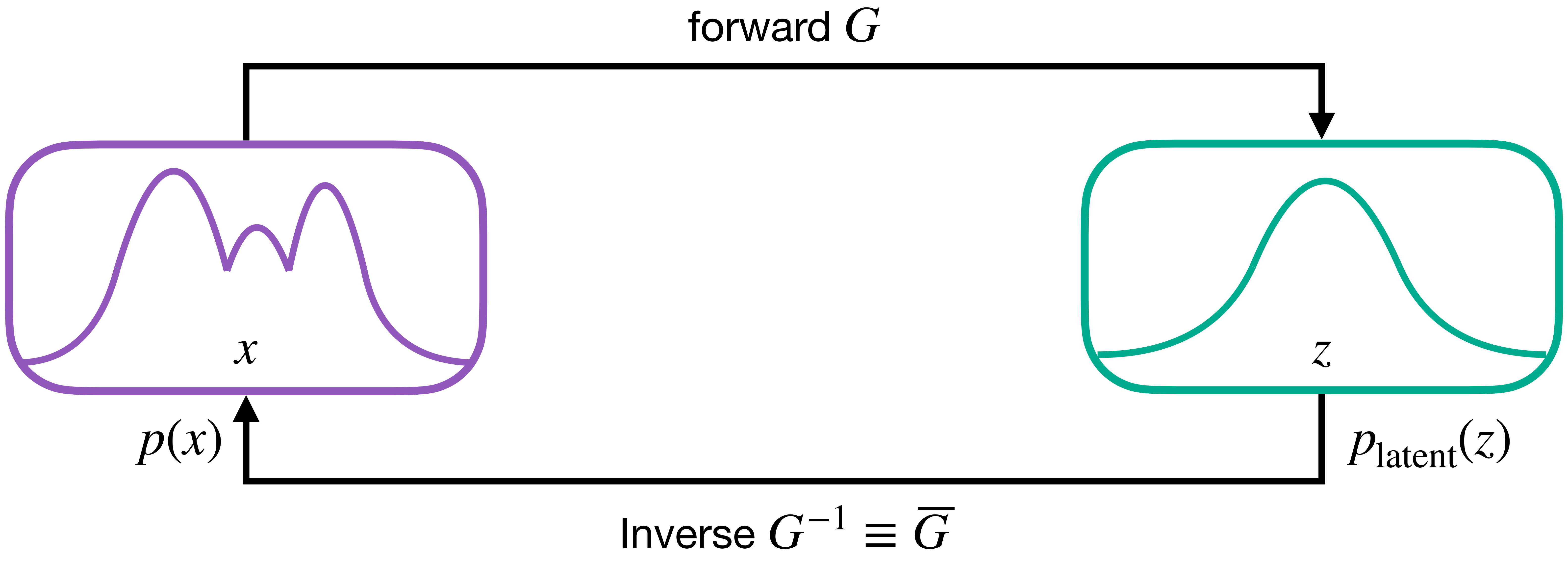


Deep generative models



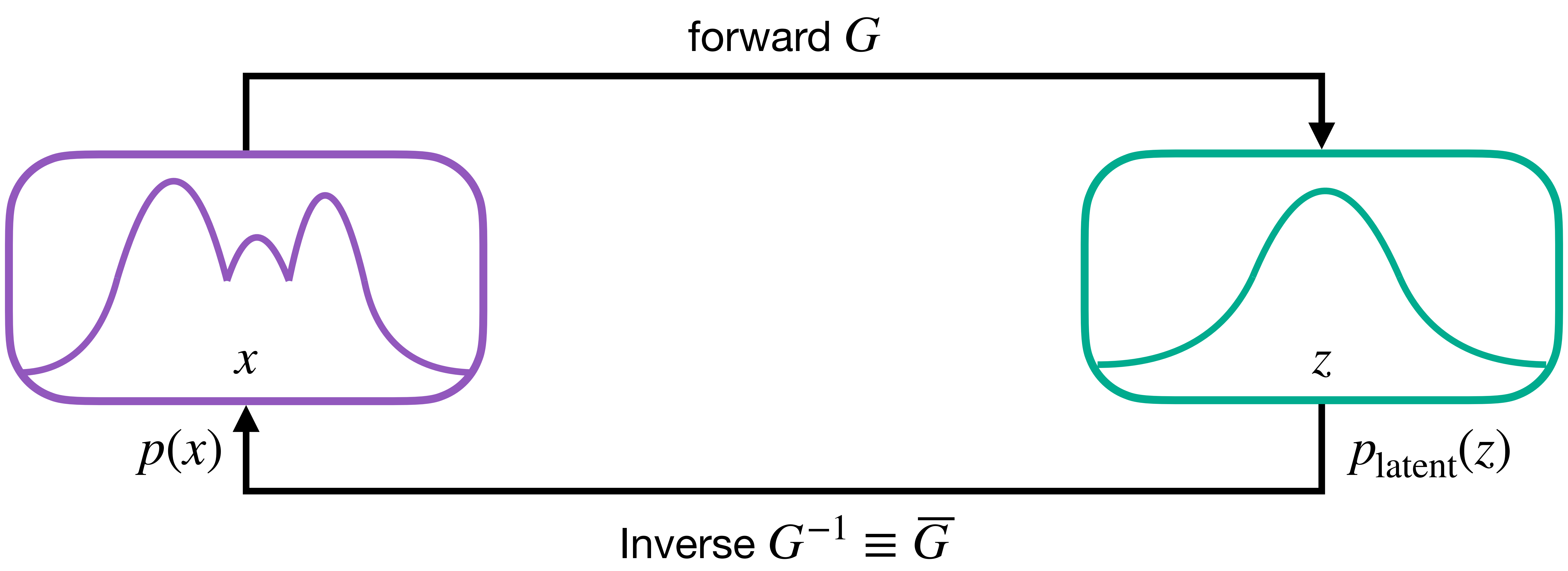
What is a normalizing flow?

Normalizing flow — Basics



[1505.05770, 1908.09257]

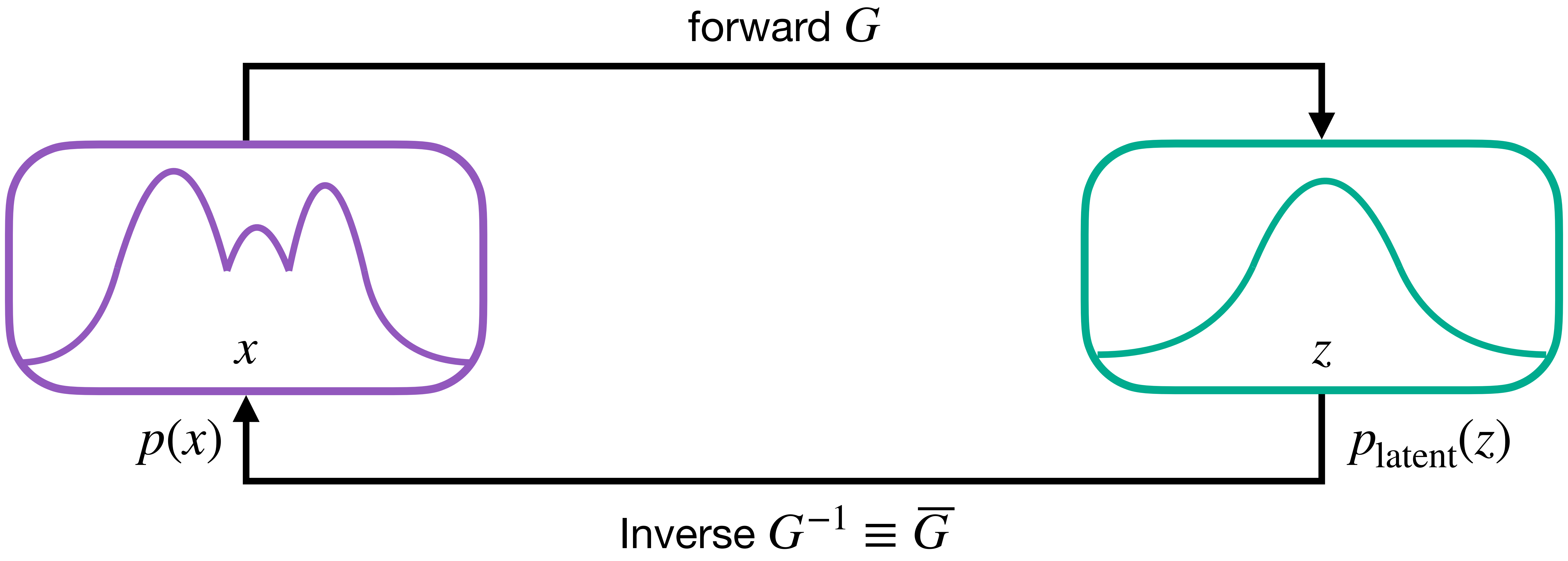
Normalizing flow — Basics



Conservation of probability: $p(x)\,\mathrm{d}x = p_{\text{latent}}(z)\,\mathrm{d}z$ with $z = G_\omega(x) \quad x = \overline{G}_\omega(z)$

[1505.05770, 1908.09257]

Normalizing flow — Basics

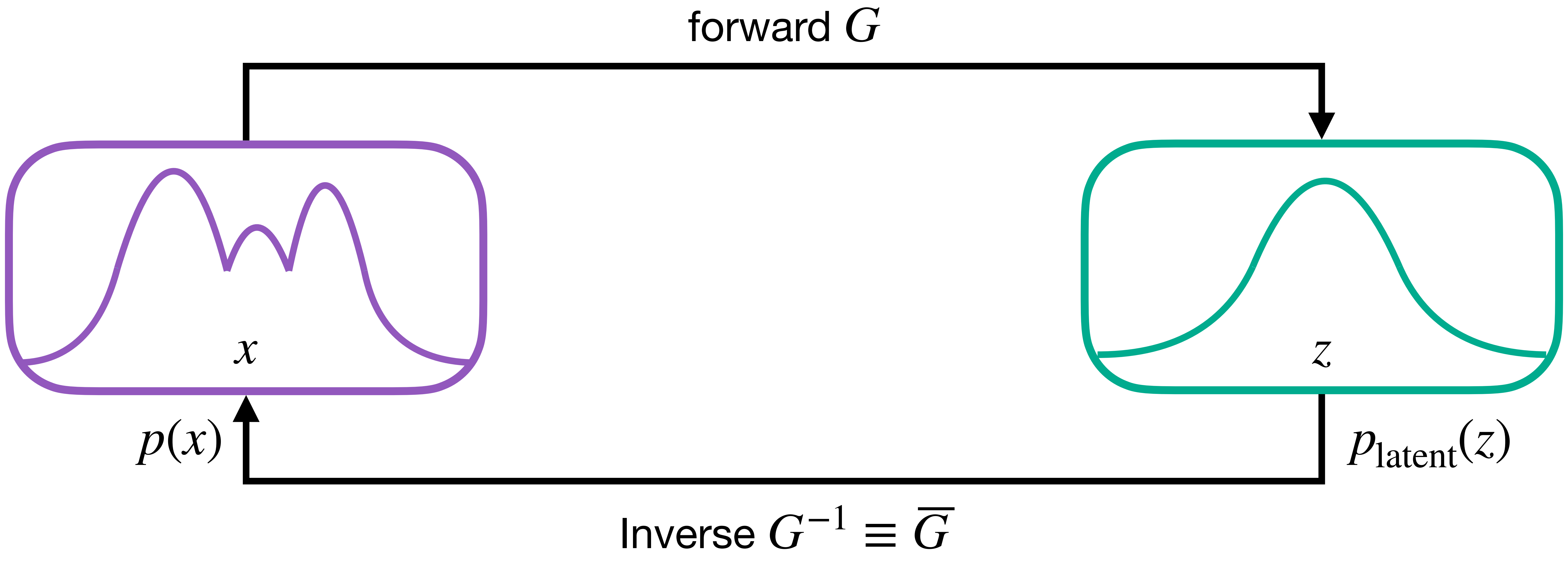


Conservation of probability: $p(x)\,\mathrm{d}x = p_{\text{latent}}(z)\,\mathrm{d}z$ with $z = G_\omega(x) \quad x = \overline{G}_\omega(z)$

Change-of-variables formula: $$p_\omega(x) = p_{\text{latent}}(z = G_\omega(x)) \cdot \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

[1505.05770, 1908.09257]

Normalizing flow — Basics

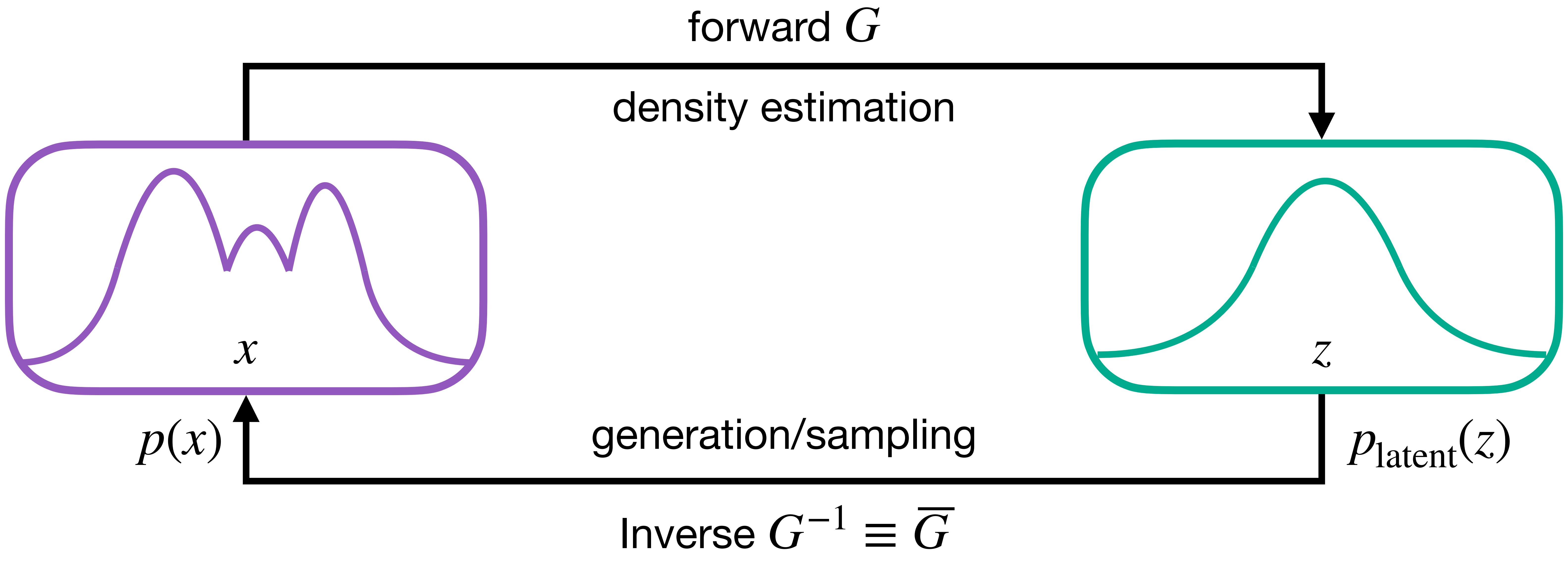


Conservation of probability: $p(x)\,\mathrm{d}x = p_{\text{latent}}(z)\,\mathrm{d}z$ with $z = G_\omega(x) \quad x = \overline{G}_\omega(z)$

Change-of-variables formula: $\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$

[1505.05770, 1908.09257]

Normalizing flow — Basics



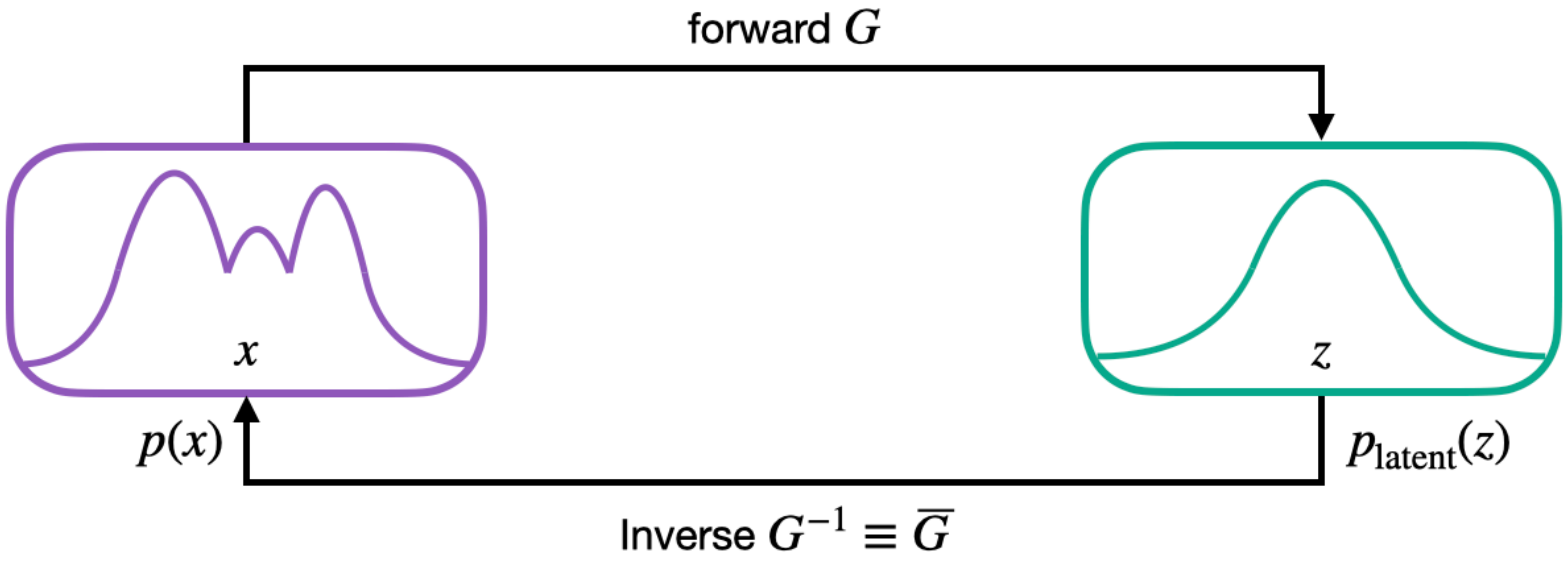
Conservation of probability: $p(x)\,\mathrm{d}x = p_{\text{latent}}(z)\,\mathrm{d}z$ with $z = G_\omega(x) \quad x = \overline{G}_\omega(z)$

Change-of-variables formula: $\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$

[1505.05770, 1908.09257]

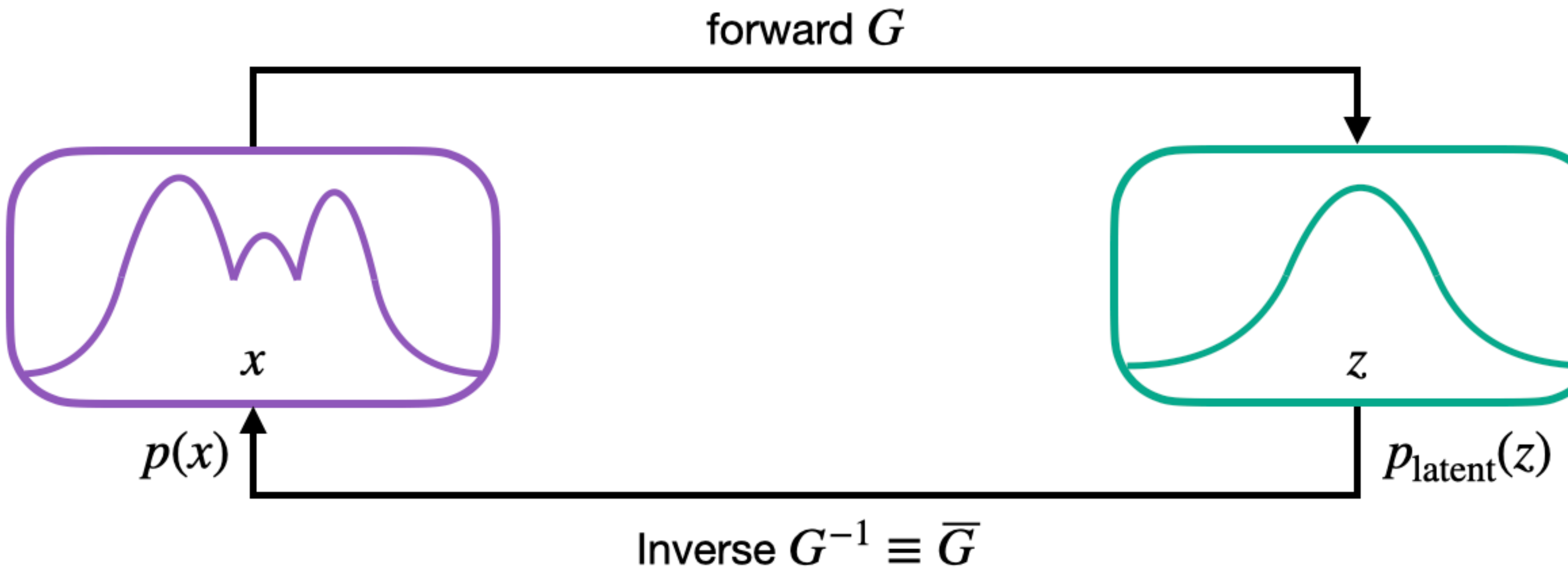
How to train it?

Normalizing flow — Training



$$\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

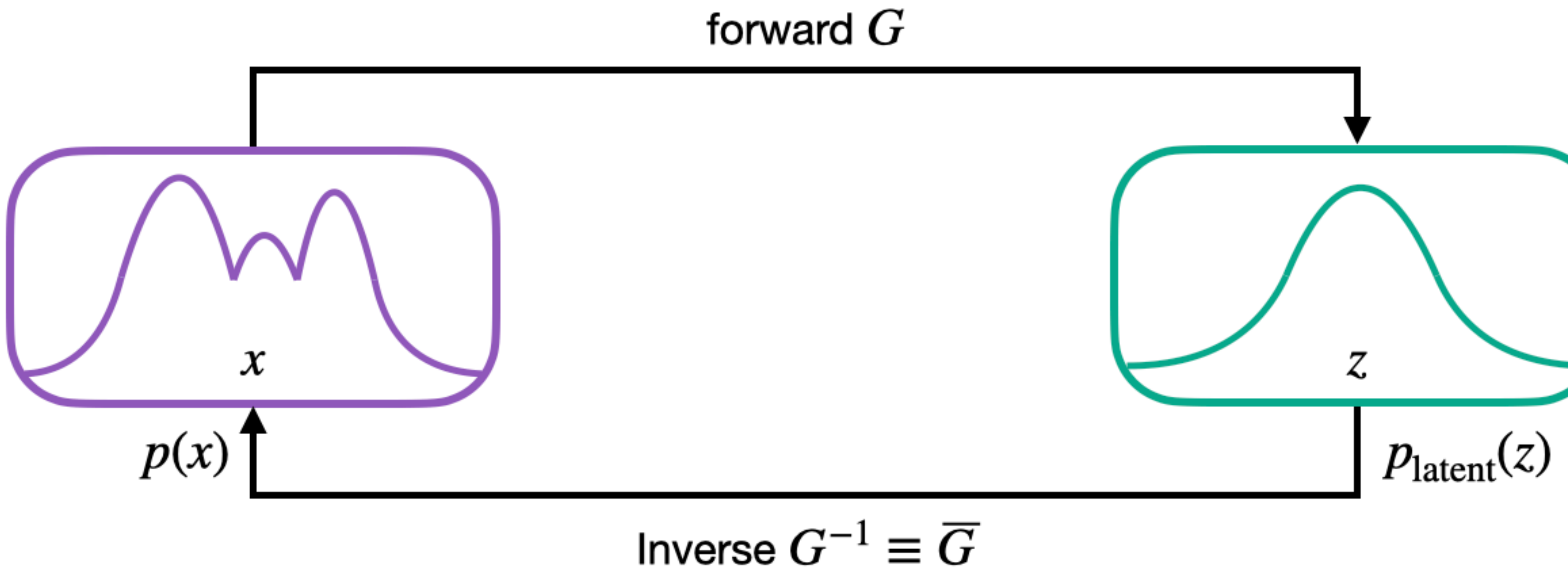
Normalizing flow — Training



$$\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

→ Match $p_\omega(x)$ with $p_{\text{data}}(x)$

Normalizing flow — Training



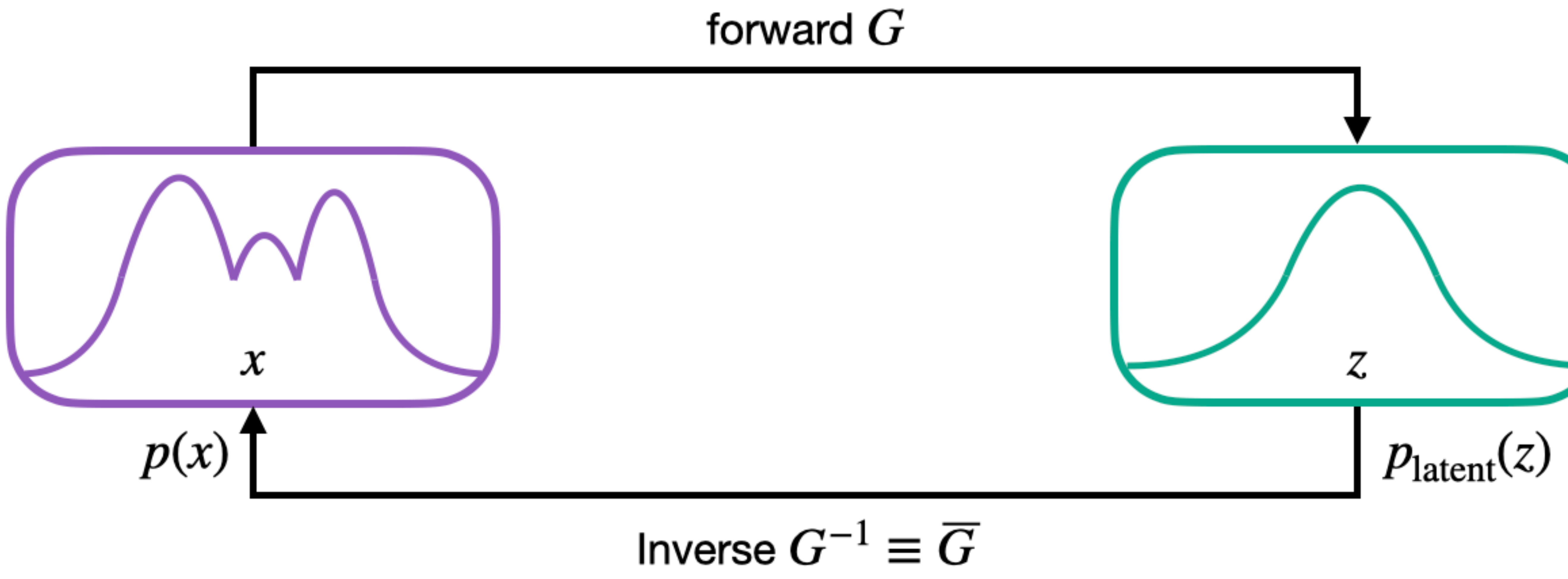
$$\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

→ Match $p_\omega(x)$ with $p_{\text{data}}(x)$

Kullback-Leibler divergence:

$$\begin{aligned} \mathrm{KL}(p_{\text{data}}(x) \,|\, p_\omega(x)) &= \int \mathrm{d}x \, p_{\text{data}}(x) \, \log \frac{p_{\text{data}}(x)}{p_\omega(x)} \\ &= -\int \mathrm{d}x \, p_{\text{data}}(x) \, \log p_\omega(x) + \int \mathrm{d}x \, p_{\text{data}}(x) \, \log p_{\text{data}}(x) \end{aligned}$$

Normalizing flow — Training



$$\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

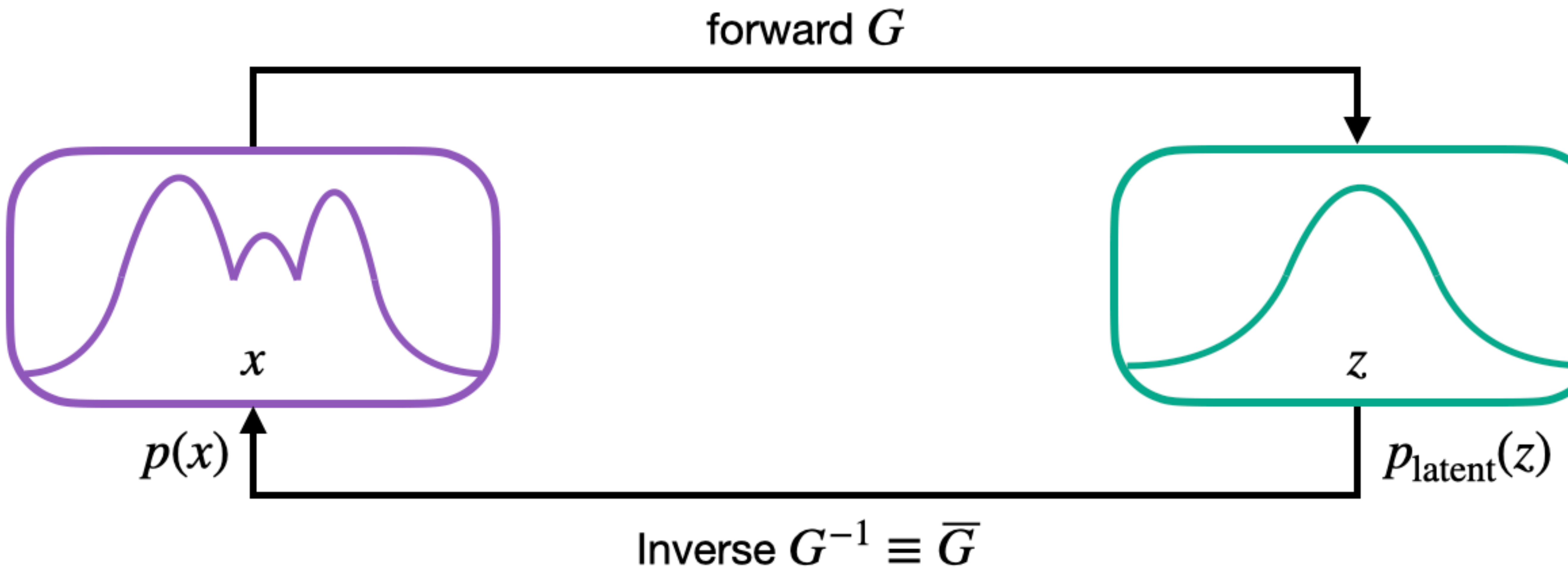
→ Match $p_\omega(x)$ with $p_{\text{data}}(x)$

Kullback-Leibler divergence:

$$\begin{aligned} \mathrm{KL}(p_{\text{data}}(x) \,|\, p_\omega(x)) &= \int \mathrm{d}x\, p_{\text{data}}(x)\, \log \frac{p_{\text{data}}(x)}{p_\omega(x)} \\ &= -\int \mathrm{d}x\, p_{\text{data}}(x)\, \log p_\omega(x) + \boxed{\int \mathrm{d}x\, p_{\text{data}}(x)\, \log p_{\text{data}}(x)} \end{aligned}$$

No ω dependence

Normalizing flow — Training



$$\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

→ Match $p_\omega(x)$ with $p_{\text{data}}(x)$

Kullback-Leibler divergence:

$$\begin{aligned}\mathrm{KL}(p_{\text{data}}(x) \,|\, p_\omega(x)) &= \int \mathrm{d}x\, p_{\text{data}}(x) \log \frac{p_{\text{data}}(x)}{p_\omega(x)} \\ &= -\int \mathrm{d}x\, p_{\text{data}}(x) \log p_\omega(x) + \int \mathrm{d}x\, p_{\text{data}}(x) \log p_{\text{data}}(x)\end{aligned}$$

No ω dependence

Negative log-likelihood loss:

$$\mathscr{L}_{\text{NLL}} = -\int \mathrm{d}x\, p_{\text{data}}(x) \log p_\omega(x) = \langle -\log p_\omega(x) \rangle_{x \sim p_{\text{data}}}$$

Tractable Jacobian?

$$\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

→ Requires tractable Jacobian!

Tractable Jacobian?

$$\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

→ Requires tractable Jacobian!

In general: $g_\omega(x) = \left| \frac{\partial G_\omega(x)}{\partial x} \right|$ is $d \times d$ matrix

→ Scales with $\mathcal{O}(d^3)$ ☹

Tractable Jacobian?

$$\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

→ Requires tractable Jacobian!

In general: $g_\omega(x) = \left| \frac{\partial G_\omega(x)}{\partial x} \right|$ is $d \times d$ matrix → Scales with $\mathcal{O}(d^3)$ ☹

Solution: **Autoregressive transformations** $z = \begin{pmatrix} z_1 \\ \vdots \\ z_d \end{pmatrix}$ $x = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$

Tractable Jacobian?

$$\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

→ Requires tractable Jacobian!

In general: $g_\omega(x) = \left| \frac{\partial G_\omega(x)}{\partial x} \right|$ is $d \times d$ matrix → Scales with $\mathcal{O}(d^3)$ ☹

Solution: **Autoregressive transformations** $z = \begin{pmatrix} z_1 \\ \vdots \\ z_d \end{pmatrix}$ $x = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$

$$\begin{aligned} z_1 &\equiv z_1(x_1) \\ z_2 &\equiv z_2(x_1, x_2) \\ &\vdots \\ z_d &\equiv z_d(x_1, x_2, \ldots, x_d) \end{aligned}$$

Tractable Jacobian?

$$\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

→ Requires tractable Jacobian!

In general: $g_\omega(x) = \left| \frac{\partial G_\omega(x)}{\partial x} \right|$ is $d \times d$ matrix → Scales with $\mathcal{O}(d^3)$ ☹

Solution: **Autoregressive transformations** $z = \begin{pmatrix} z_1 \\ \vdots \\ z_d \end{pmatrix}$ $x = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$

$$\begin{aligned} z_1 &\equiv z_1(x_1) \\ z_2 &\equiv z_2(x_1, x_2) \\ &\vdots \\ z_d &\equiv z_d(x_1, x_2, \ldots, x_d) \end{aligned}$$

→

$$J_{ij}(x) = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_2}{\partial x_1} & \cdots & \frac{\partial z_d}{\partial x_1} \\ 0 & \frac{\partial z_2}{\partial x_2} & \cdots & \frac{\partial z_d}{\partial x_1} \\ \vdots & \ddots & & \vdots \\ 0 & \ldots & 0 & \frac{\partial z_d}{\partial x_d} \end{pmatrix}$$

Tractable Jacobian?

$$\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$ → Requires tractable Jacobian!

In general: $g_\omega(x) = \left| \frac{\partial G_\omega(x)}{\partial x} \right|$ is $d \times d$ matrix → Scales with $\mathcal{O}(d^3)$ ☹

Solution: **Autoregressive transformations** $z = \begin{pmatrix} z_1 \\ \vdots \\ z_d \end{pmatrix}$ $x = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$

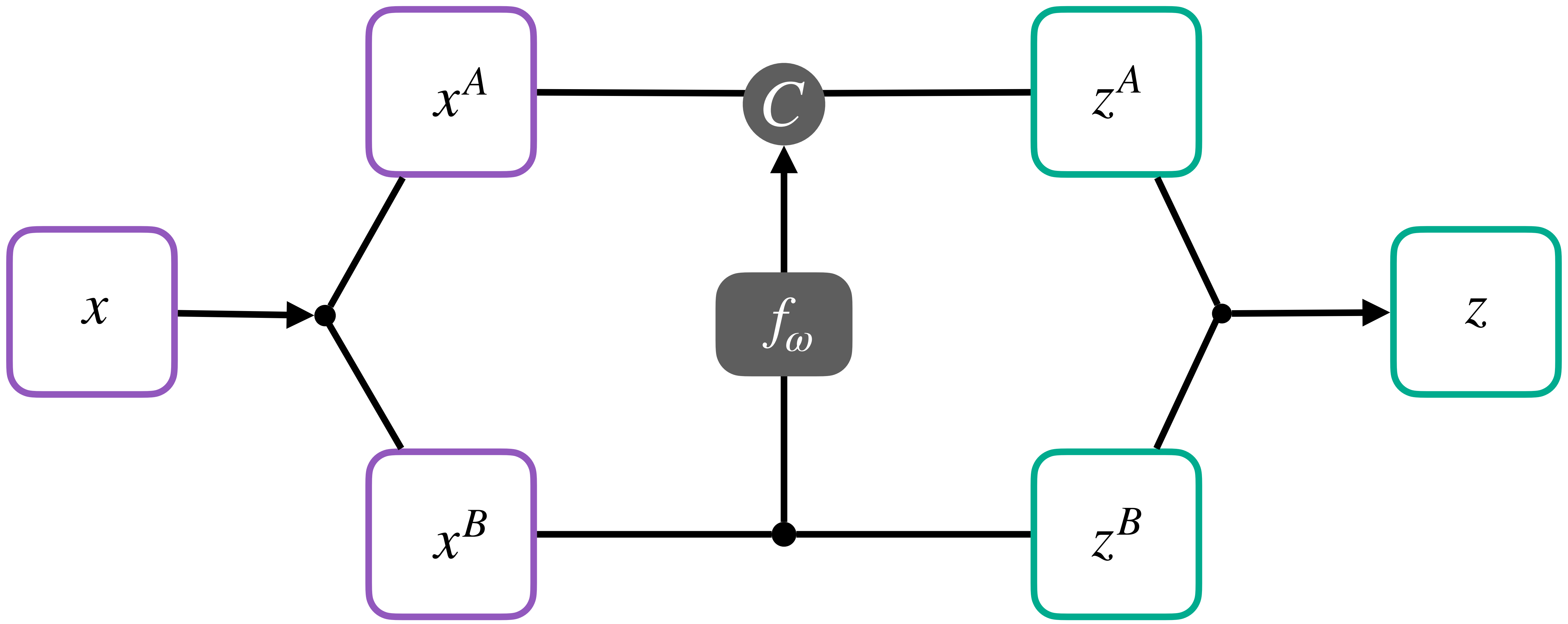
$$\begin{aligned} z_1 &\equiv z_1(x_1) \\ z_2 &\equiv z_2(x_1, x_2) \\ &\vdots \\ z_d &\equiv z_d(x_1, x_2, \ldots, x_d) \end{aligned}$$

→

$$J_{ij}(x) = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_2}{\partial x_1} & \cdots & \frac{\partial z_d}{\partial x_1} \\ 0 & \frac{\partial z_2}{\partial x_2} & \cdots & \frac{\partial z_d}{\partial x_1} \\ \vdots & \ddots & & \vdots \\ 0 & \ldots & 0 & \frac{\partial z_d}{\partial x_d} \end{pmatrix}$$

→ $\det J = \prod_i J_{ii} \sim \mathcal{O}(d)$ ☺

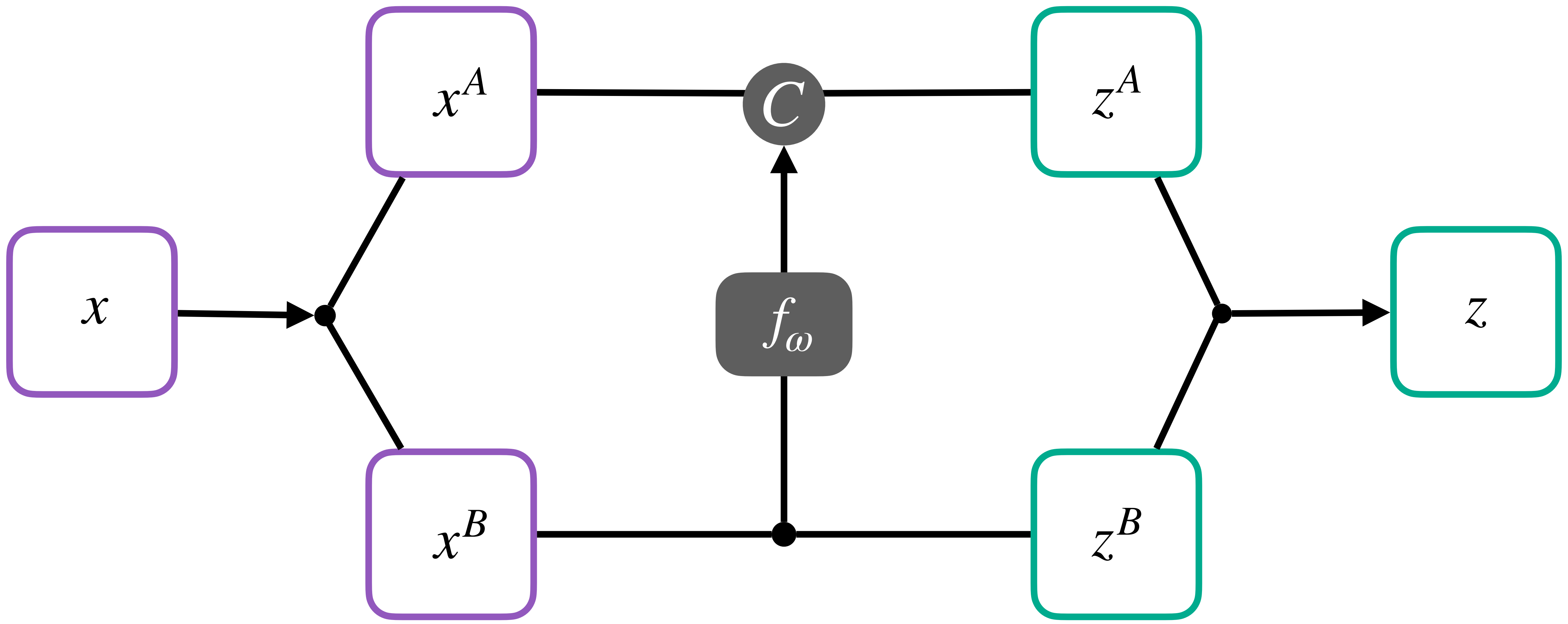
Coupling block



Forward pass:

$$z^A = C(x^A; f_\omega(x^B))$$
$$z^B = x^B$$

Coupling block

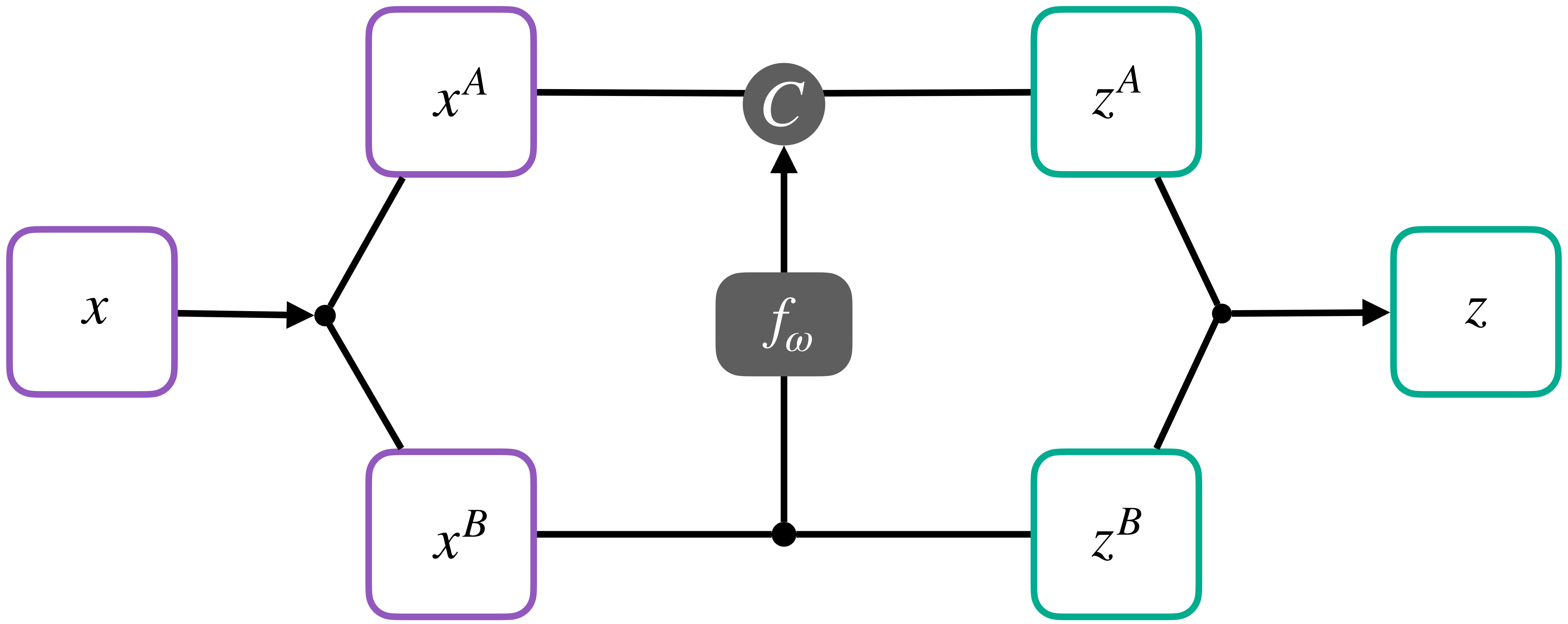


Forward pass:

$$z^A = C(x^A; f_\omega(x^B))$$
$$z^B = x^B$$

$$J_{ij}(x) = \begin{pmatrix} \frac{\partial C}{\partial x^A} & \frac{\partial C}{\partial f_\omega}\frac{\partial f_\omega}{\partial x^B} \\ 0 & I_m \end{pmatrix}$$

Coupling block



Forward pass:

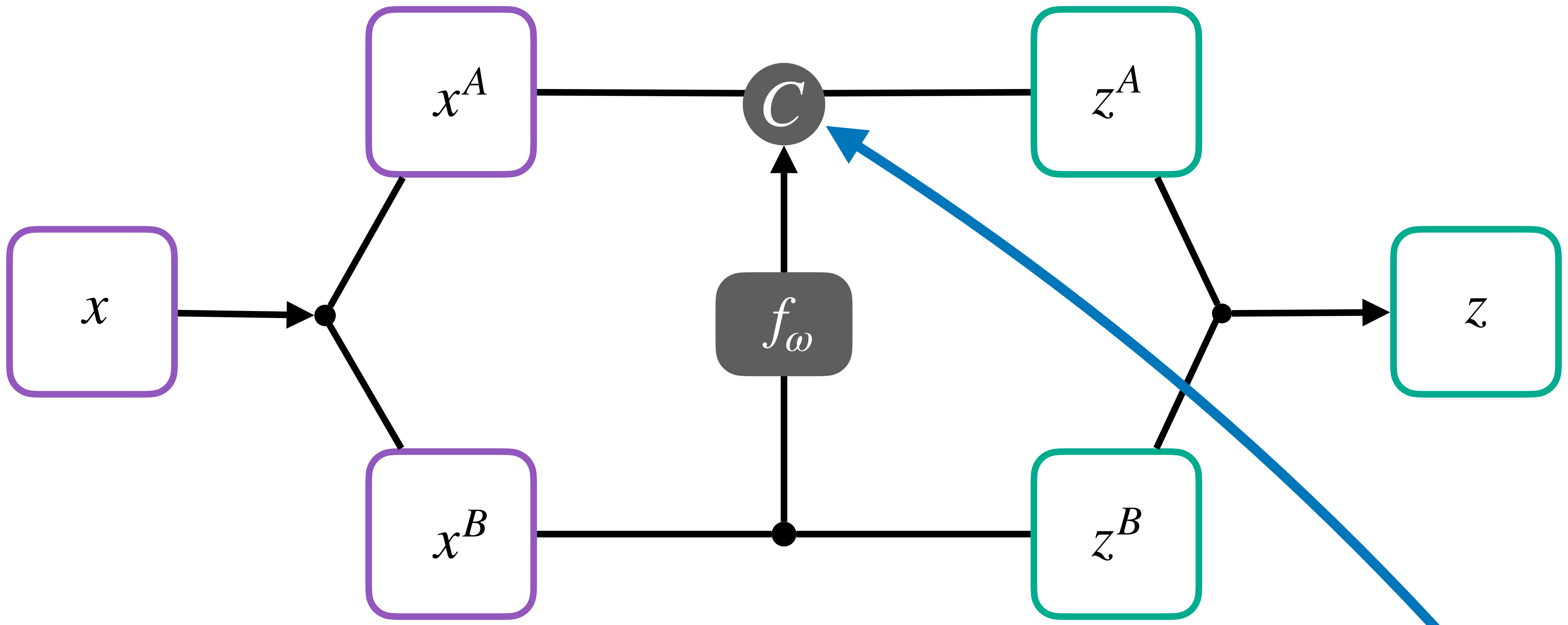
$$z^A = C(x^A; f_\omega(x^B))$$
$$z^B = x^B$$

Inverse pass:

$$x^A = C^{-1}(z^A; f_\omega(z^B))$$
$$x^B = z^B$$

$$J_{ij}(x) = \begin{pmatrix} \frac{\partial C}{\partial x^A} & \frac{\partial C}{\partial f_\omega}\frac{\partial f_\omega}{\partial x^B} \\ 0 & I_m \end{pmatrix}$$

Coupling block



Forward pass:

$$z^A = C(x^A; f_\omega(x^B))$$
$$z^B = x^B$$

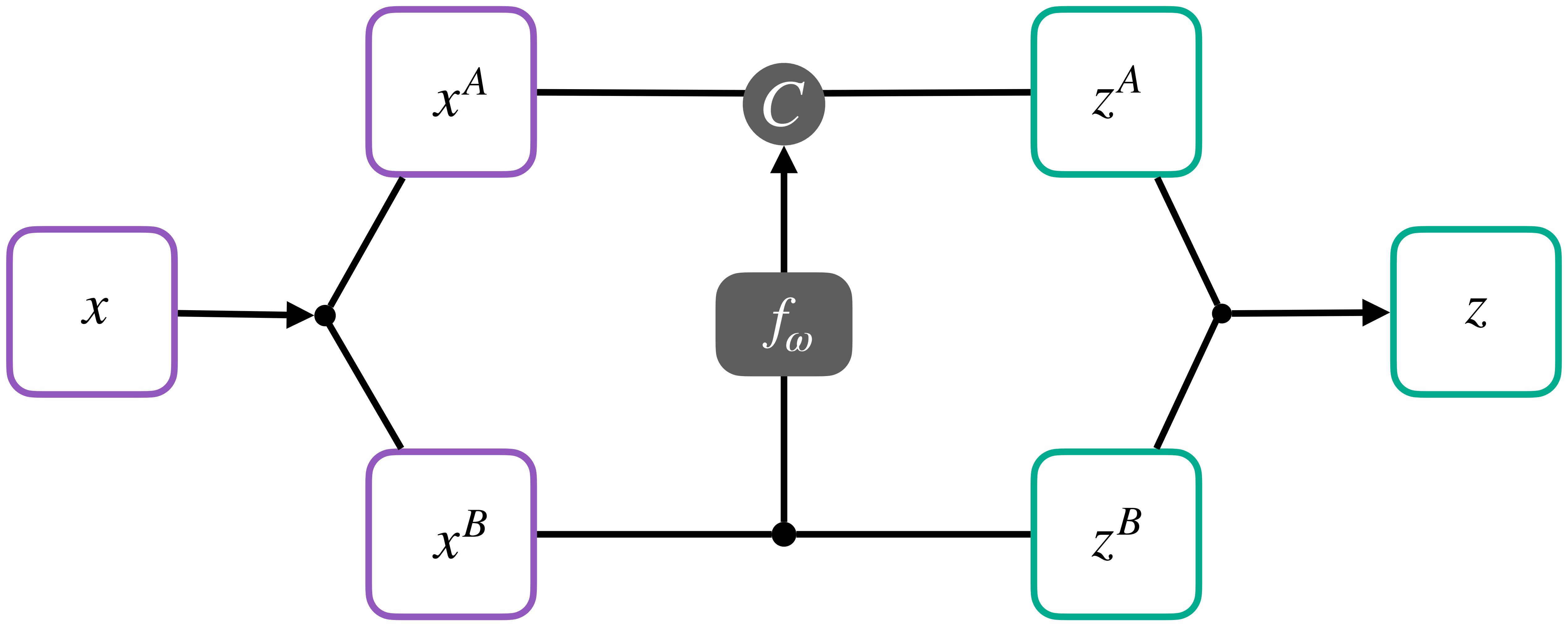
Inverse pass:

$$x^A = C^{-1}(z^A; f_\omega(z^B))$$
$$x^B = z^B$$

$$J_{ij}(x) = \begin{pmatrix} \frac{\partial C}{\partial x^A} & \frac{\partial C}{\partial f_\omega}\frac{\partial f_\omega}{\partial x^B} \\ 0 & I_m \end{pmatrix}$$

What is the function C?

Coupling block



Forward pass:

$$z^A = C(x^A; f_\omega(x^B))$$
$$z^B = x^B$$

Inverse pass:

$$x^A = C^{-1}(z^A; f_\omega(z^B))$$
$$x^B = z^B$$

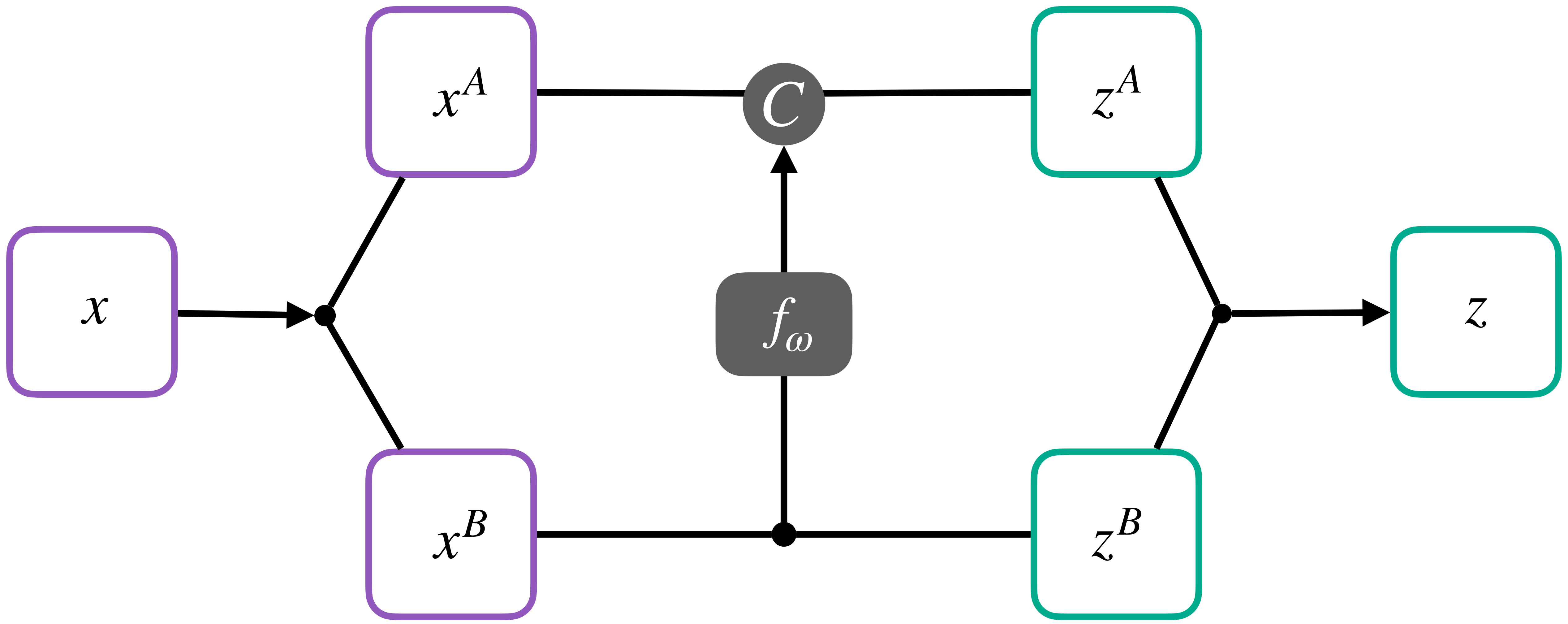
$$J_{ij}(x) = \begin{pmatrix} \frac{\partial C}{\partial x^A} & \frac{\partial C}{\partial f_\omega}\frac{\partial f_\omega}{\partial x^B} \\ 0 & I_m \end{pmatrix}$$

Affine [1605.08803]

$$C^A = \alpha_\omega(x^B) \cdot x^A + \mu_\omega(x^B)$$

parametrized by NN

Coupling block



Forward pass:

$$z^A = C(x^A; f_\omega(x^B))$$
$$z^B = x^B$$

Inverse pass:

$$x^A = C^{-1}(z^A; f_\omega(z^B))$$
$$x^B = z^B$$

$$J_{ij}(x) = \begin{pmatrix} \frac{\partial C}{\partial x^A} & \frac{\partial C}{\partial f_\omega} \frac{\partial f_\omega}{\partial x^B} \\ 0 & I_m \end{pmatrix}$$

Affine [1605.08803]	$C^A = \alpha_\omega(x^B) \cdot x^A + \mu_\omega(x^B)$
Quadratic [1808.03856]	$C = a_\omega x^2 + b_\omega x + c_\omega$
Rational quadratic [1906.04032]	$C = \dfrac{a_\omega x^2 + b_\omega x + c_\omega}{d_\omega x^2 + e_\omega x + f_\omega}$


Machine Learning
Supervised Learning
Classification
Point Clouds [2102.05073]
PELICAN [2211.00454]
Energy Flow Networks [1810.05165]
Landscape of Top tagger [1902.09914]
Bayesian Tagger [1904.10004]
Regression
MadMiner [1907.10621,…]
NNPDF [2109.02653]
Matrix Elements [2206.14831]
Symbolic regression [2109.10414]
Reinforcement Learning
Flavor structure [2304.14176]
Simplifying Polylogs [2206.04115]
String vacua & landscape [1903.11616, 2111.11466]
Model Building [2103.04759]
Analytic continuation [2112.13011]
Unsupervised Learning
Clustering
Jet Clustering [2008.06064]
3D Pixel Clustering [2007.03083]
Anomaly Detection
Normalized AE [2206.14225]
CATHODE [2109.00546]
CWoLA Hunting [1902.02634]
(R-)ANODE [2001.04990,…]
Generative Models
NF
Precision Generation [2110.13632]
MEM [2210.00019]
MADNIS [2212.06172,..]
CaloFlow I-IV [2106.05285,…]
ELSA [2305.07696]
VAE
OTUS [2101.08944]
Jet Simulation [2203.00520]
Diffusion Models
PC-JeDi [2303.05376]
FPCD [2304.01266]
DDPM & CFM [2305.10475]
GAN
EPiC-GAN [2301.08128]
How to GAN [1907.03764]
CaloGAN [1712.10321]
Transformer
JetGPT [2305.10475]

Machine Learning
Generative Models
VAE
OTUS [2101.08944]
Jet Simulation [2203.00520]
NF
Precision Generation [2110.13632]
MEM [2210.00019]
MadNIS [2212.06172,..]
CaloFlow I-IV [2106.05285,…]
ELSA [2305.07696]
Diffusion Models
PC-JeDi [2303.05376]
FPCD [2304.01266]
DDPM & CFM [2305.10475]
GAN
EPiC-GAN [2301.08128]
How to GAN [1907.03764]
CaloGAN [1712.10321]
Transformer
JetGPT [2305.10475]
Unsupervised Learning
Clustering
Jet Clustering [2008.06064]
3D Pixel Clustering [2007.03083]
Anomaly Detection
Normalized AE [2206.14225]
CATHODE [2109.00546]
CWoLA Hunting [1902.02634]
(R-)ANODE [2001.04990,…]
Supervised Learning
Classification
Point Clouds [2102.05073]
PELICAN [2211.00454]
Energy Flow Networks [1810.05165]
Landscape of Top tagger [1902.09914]
Bayesian Tagger [1904.10004]
Regression
MadMiner [1907.10621,…]
NNPDF [2109.02653]
Matrix Elements [2206.14831]
Symbolic regression [2109.10414]
Reinforcement Learning
Flavor structure [2304.14176]
Simplifying Polylogs [2206.04115]
String vacua & landscape [1903.11616, 2111.11466]
Model Building [2103.04759]
Analytic continuation [2112.13011]

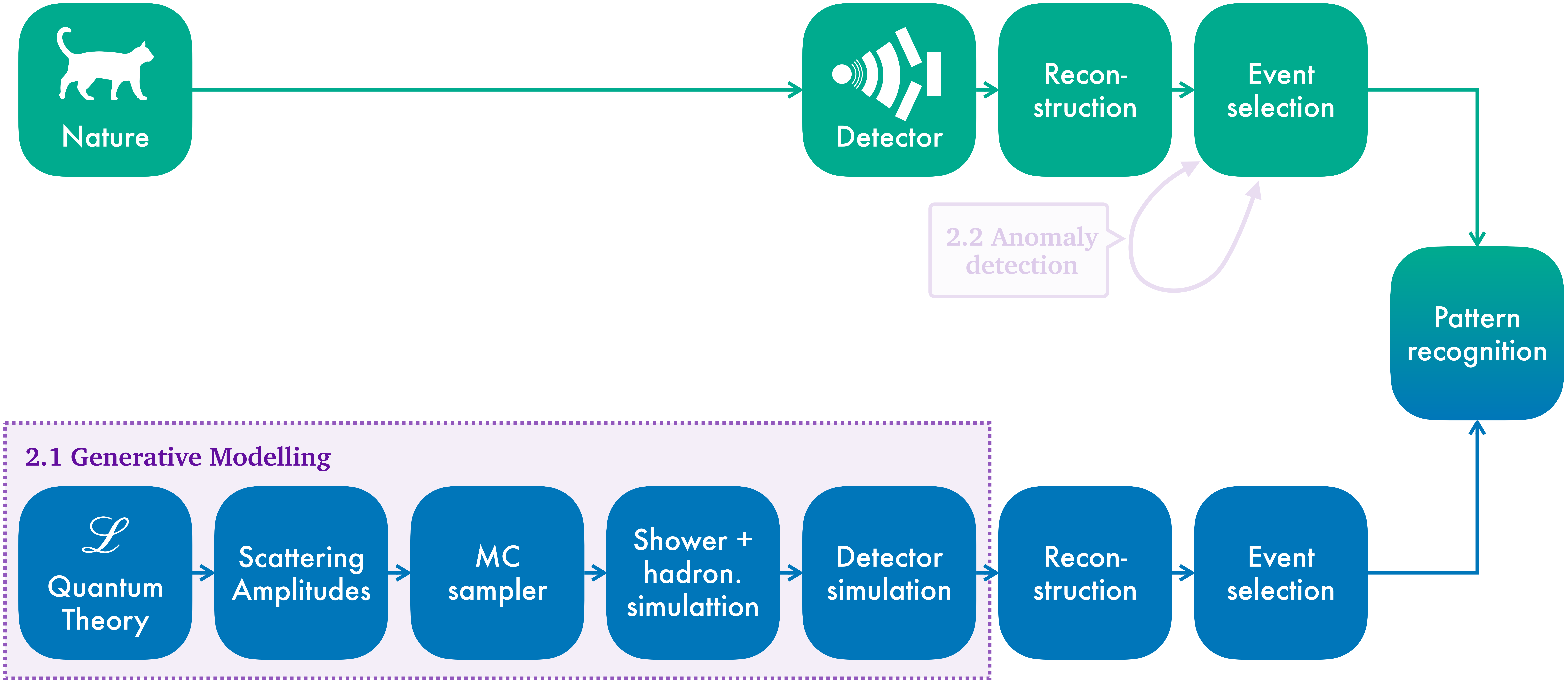
Example I

Neural importance sampling with MadNIS

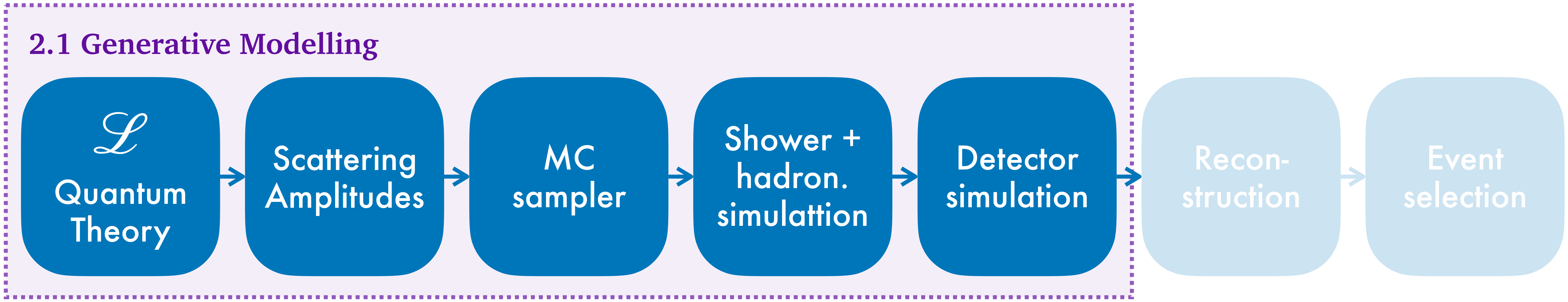
Heimel, Huetsch, Maltoni, Mattelaer, Plehn, RW [2311.01548]

Heimel, RW, Butter, Isaacson, Krause, Maltoni, Mattelaer, Plehn [2212.06172]

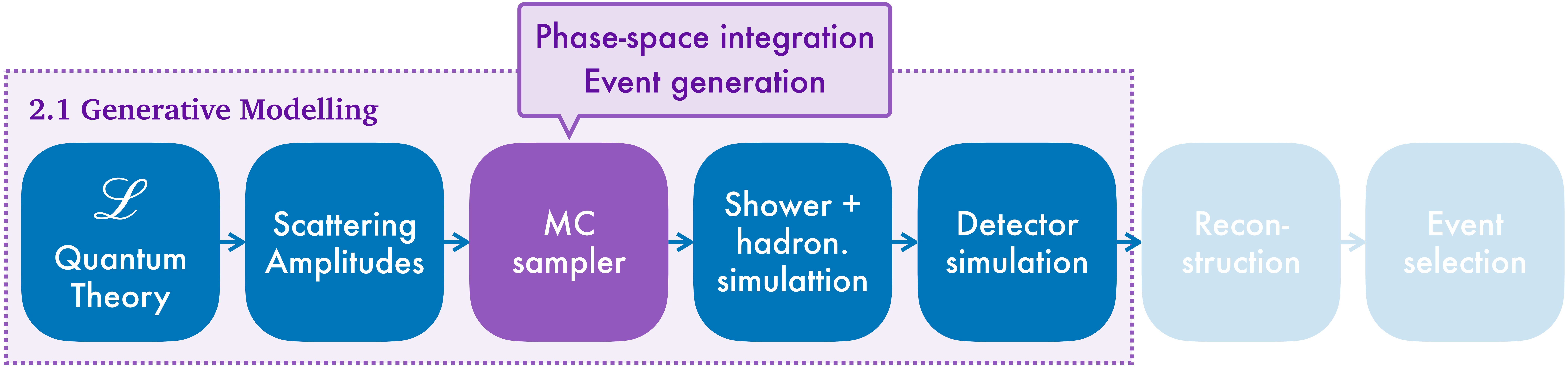
Reminder — LHC analysis + ML



LHC simulation chain



LHC simulation chain



Importance sampling

BDT [1707.00028], NN [1810.11509, 2009.07819]
NF [2001.05486, 2001.05478, 2001.10028, 2005.12719, 2112.09145, 2212.06172, 2311.01548]
Chili [2302.10449]

Monte Carlo integration

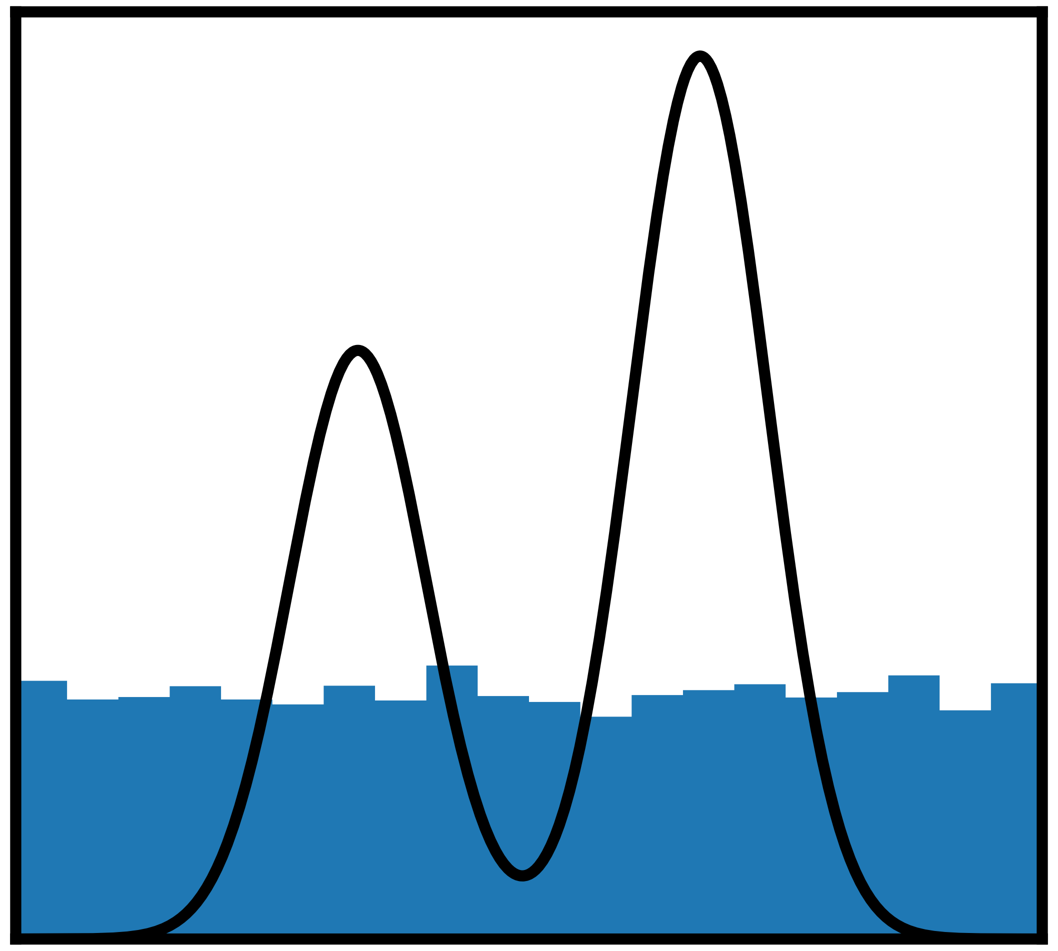
Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \mid p_1, \ldots, p_n)|^2 \,\right\rangle$$

Monte Carlo integration

Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \mid p_1, \ldots, p_n)|^2 \,\right\rangle$$



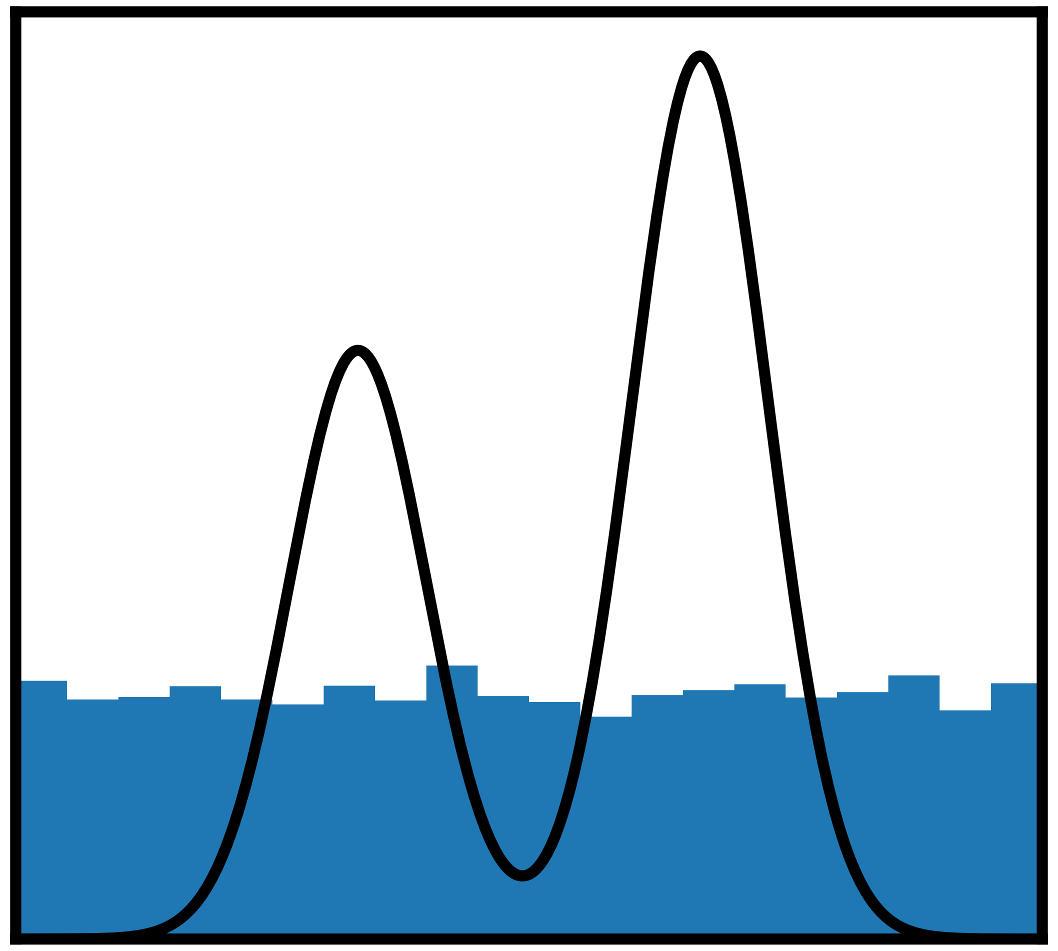
Flat sampling:
inefficient

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

Monte Carlo integration

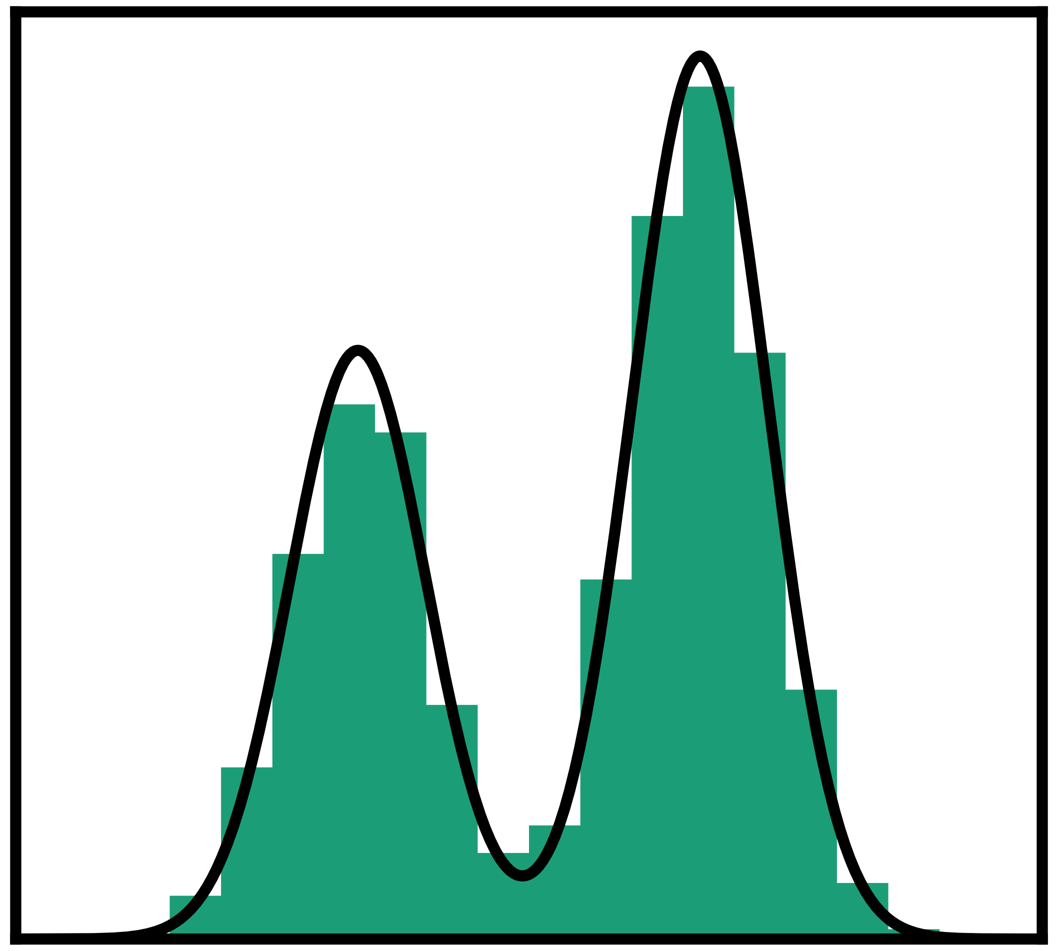
Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \mid p_1, \ldots, p_n)|^2 \,\right\rangle$$



Flat sampling:
inefficient

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



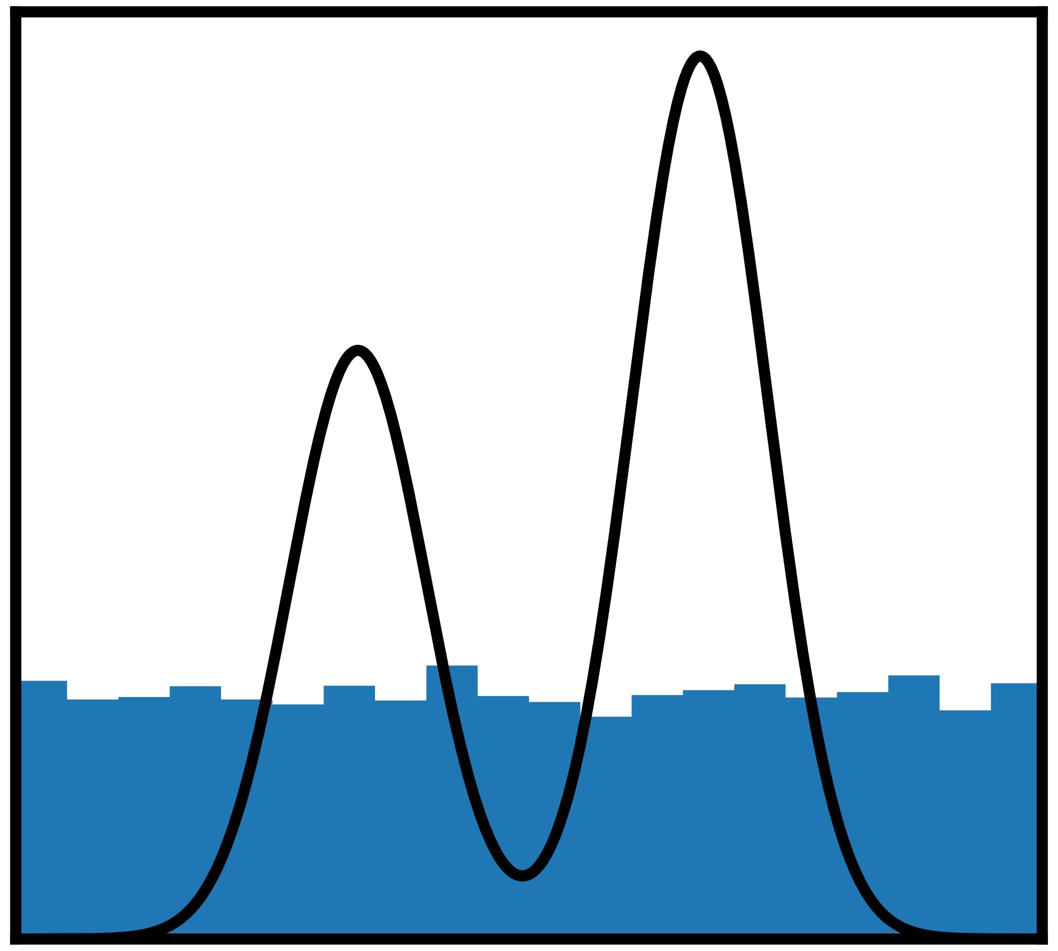
Importance sampling:
find p close to f

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

Monte Carlo integration

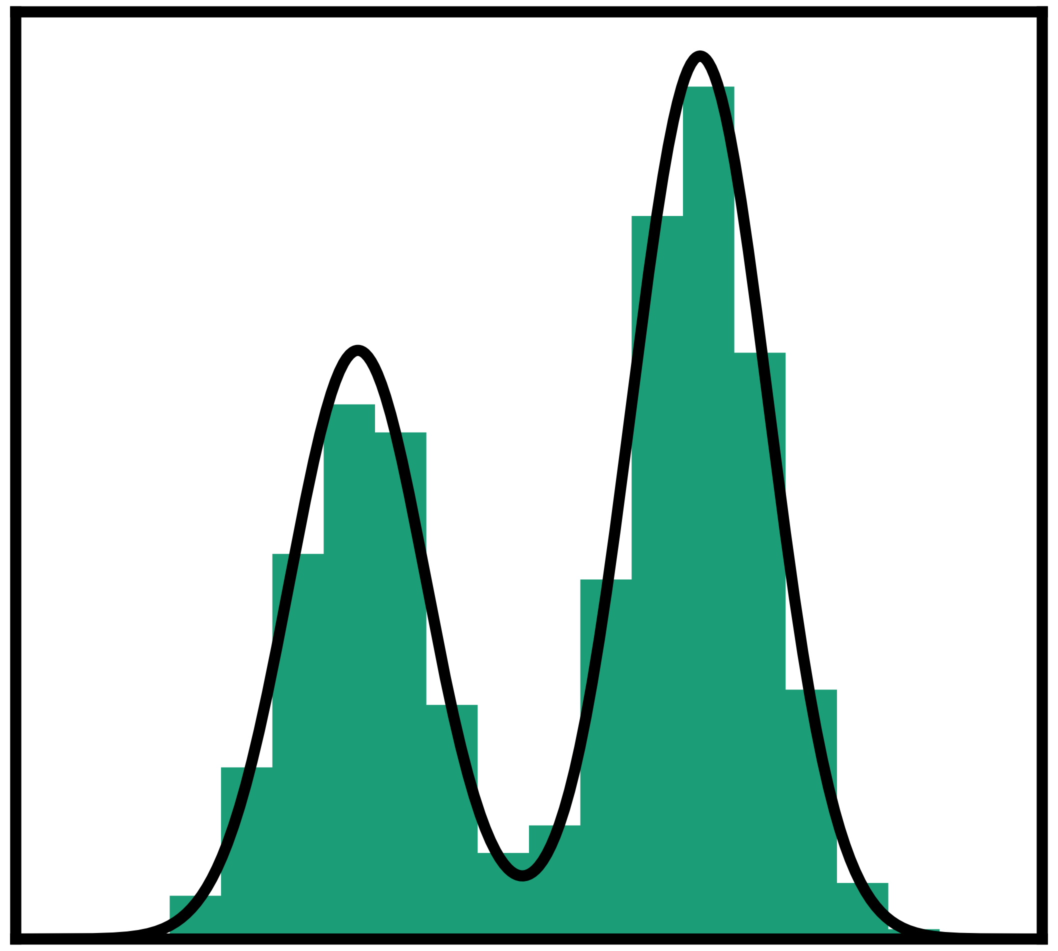
Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \mid p_1, \ldots, p_n)|^2 \,\right\rangle$$



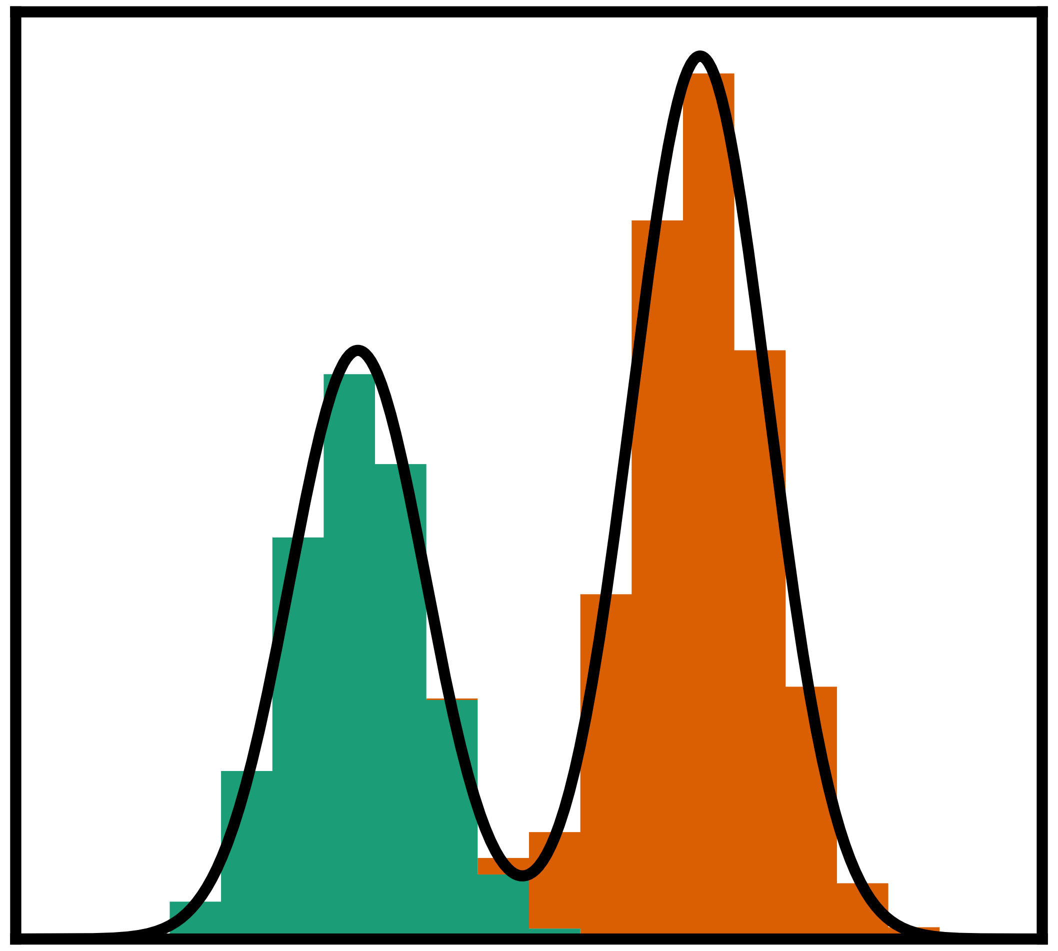
Flat sampling:
inefficient

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



Importance sampling:
find p close to f

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



Multi-channel:
one map for each channel

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

Event generation

Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \mid p_1, \ldots, p_n)|^2 \,\right\rangle$$

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

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

Channel weights

MadGraph: $\alpha_i^{\text{MG}}(x) \sim |M_i|^2$

Channel mappings

MadGraph: use amplitude structure, …

Analytic mappings + refine with **VEGAS**

(factorized, histogram based importance sampling)

Event generation + MadNIS

Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b \, f(x_a) f(x_b) \, d\Phi_n \left\langle \, |M_{\lambda,c,\ldots}(p_a, p_b \,|\, p_1, \ldots, p_n)|^2 \, \right\rangle$$

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $d\sigma / dx$

$$I = \sum_i \left\langle \alpha_i(x) \, \frac{f(x)}{p_i^{\omega}(x)} \right\rangle_{x \sim p_i^{\omega}(x)}$$

Channel weights

MadGraph: $\alpha_i^{\text{MG}}(x) \sim |M_i|^2$

Learned channel mappings

MadGraph: use amplitude structure, …

Analytic mappings + ~~refine with VEGAS~~

refine with NF

Event generation + MadNIS

Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \,|\, p_1, \ldots, p_n)|^2 \,\right\rangle$$

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

$$I = \sum_i \left\langle \alpha_i^{\xi}(x) \frac{f(x)}{p_i^{\omega}(x)} \right\rangle_{x \sim p_i^{\omega}(x)}$$

Learned Channel weights

MadGraph: $\alpha_i^{\mathrm{MG}}(x) \sim |M_i|^2$

$$\alpha_i(x) \to \alpha_i^{\xi}(x) = \alpha_i^{\mathrm{MG}}(x) \cdot K_i^{\xi}(x)$$

parametrize with **NN**

Learned channel mappings

MadGraph: use amplitude structure, …

Analytic mappings + ~~refine with VEGAS~~

refine with **NF**

MadNIS — Overview

MadNIS

Basic functionality

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

Improved multi-channeling

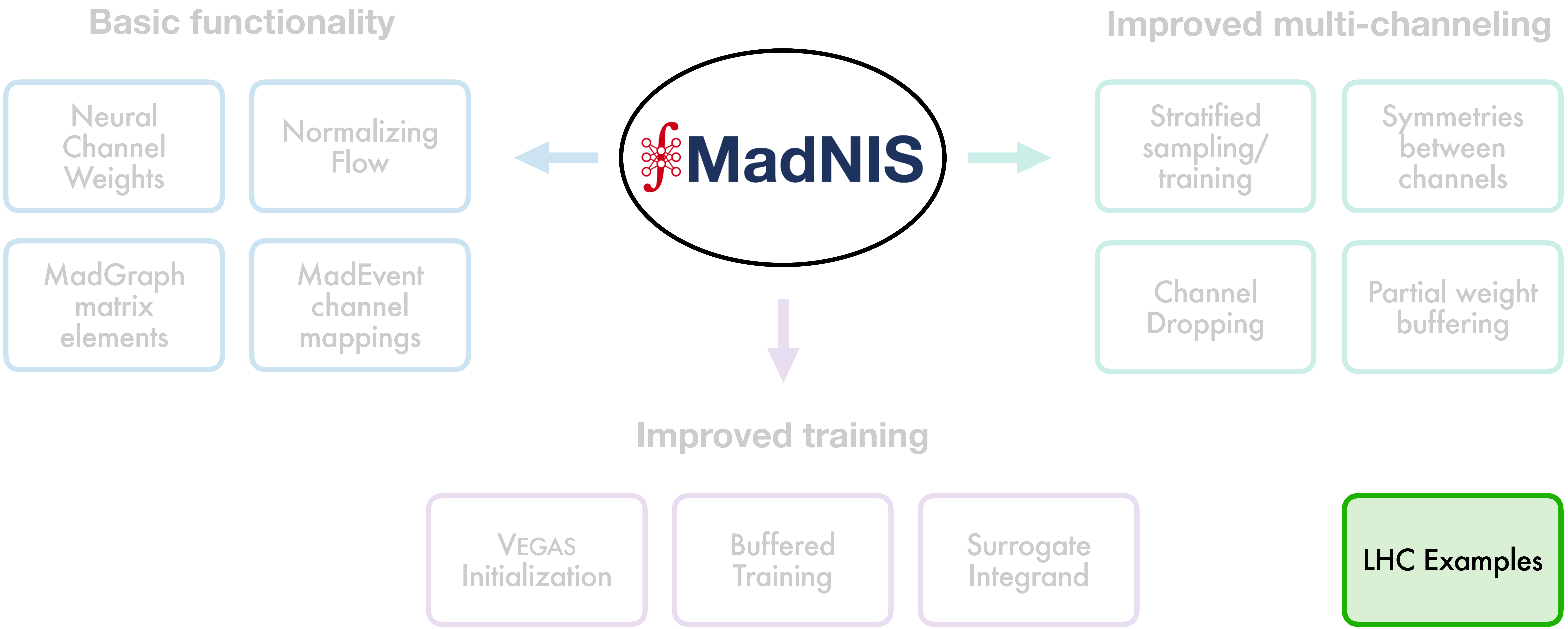
- Stratified sampling/ training
- Symmetries between channels
- Channel Dropping
- Partial weight buffering

Improved training

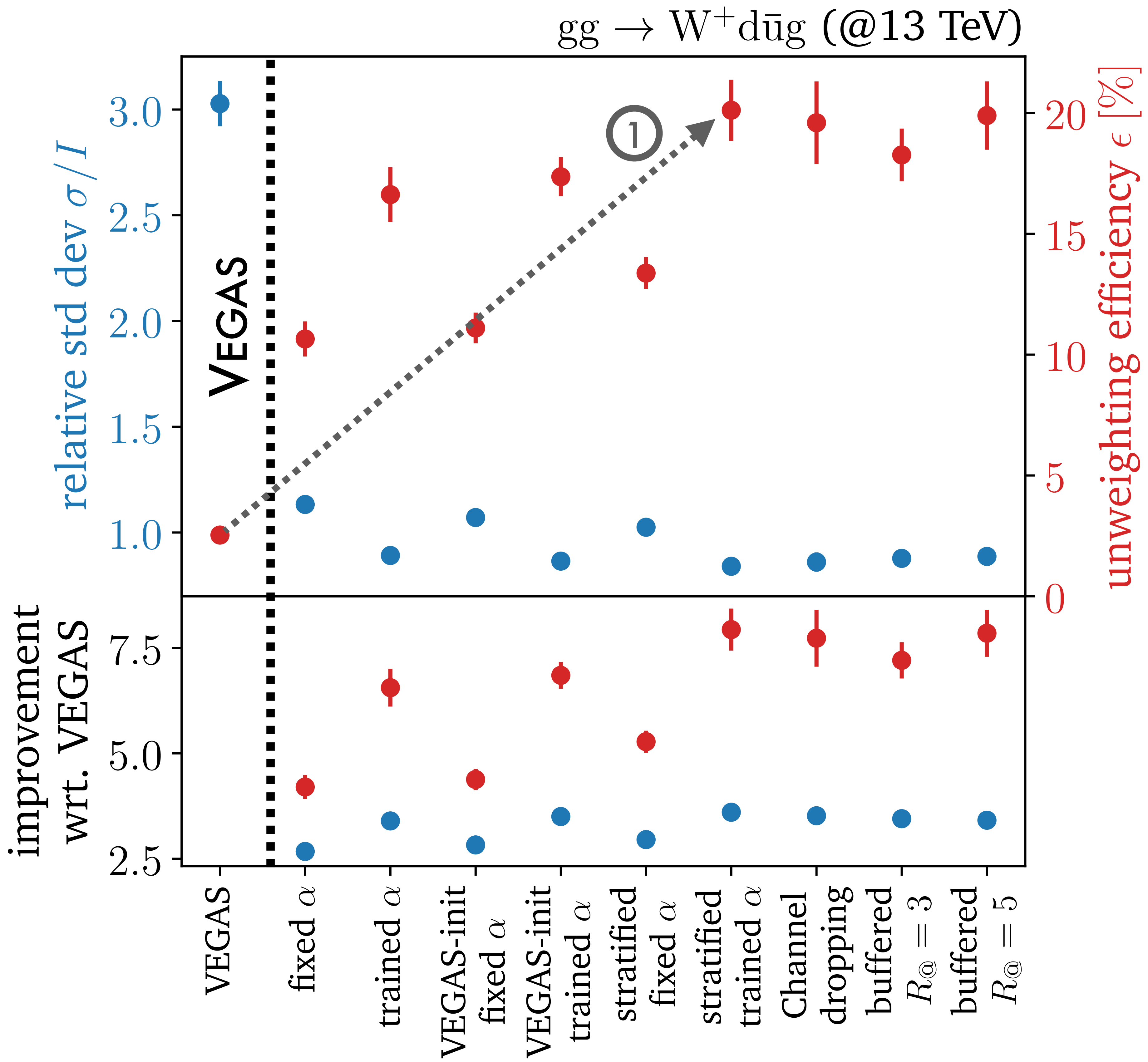
- VEGAS Initialization
- Buffered Training
- Surrogate Integrand

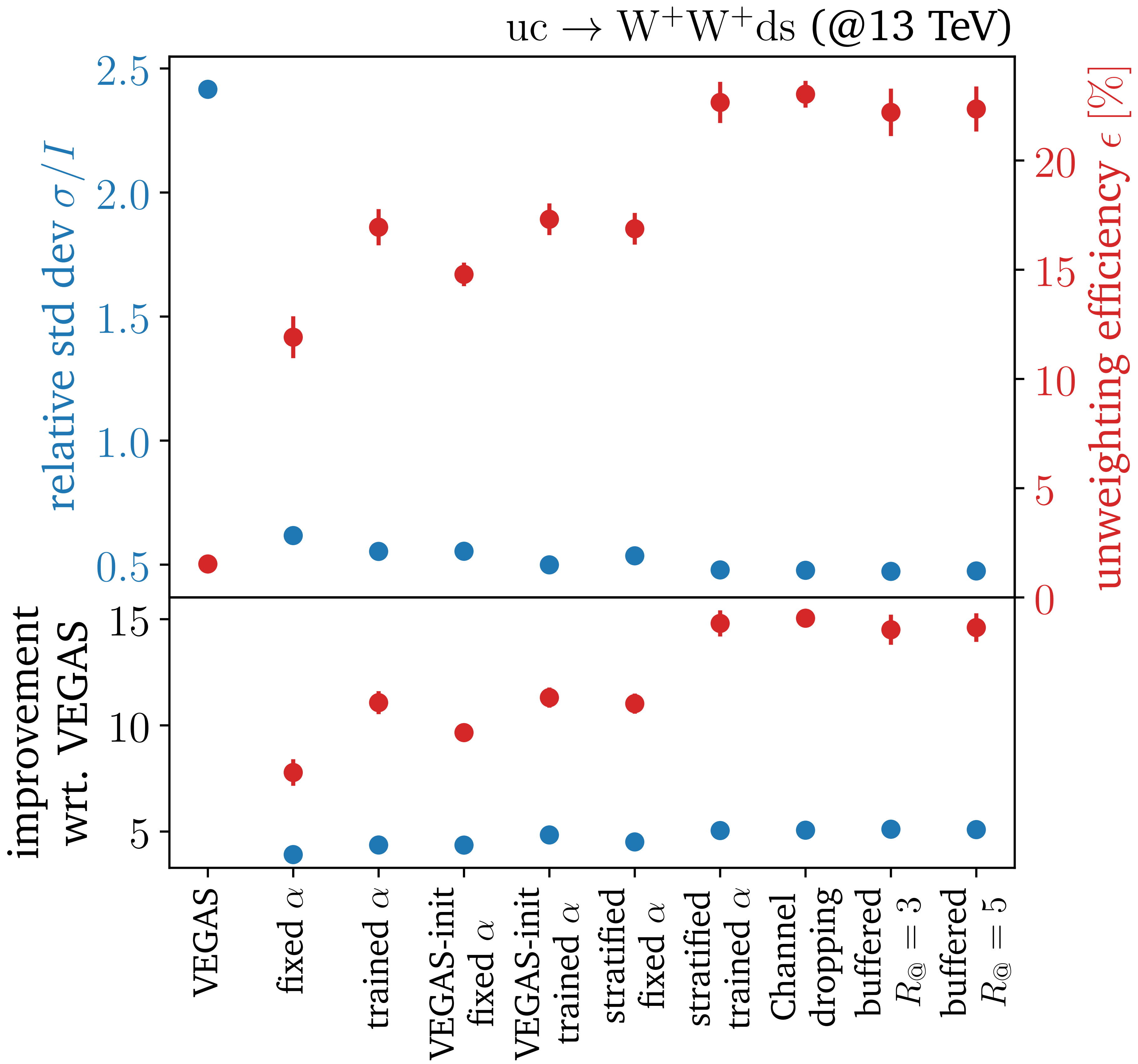
LHC Examples

MadNIS — Overview



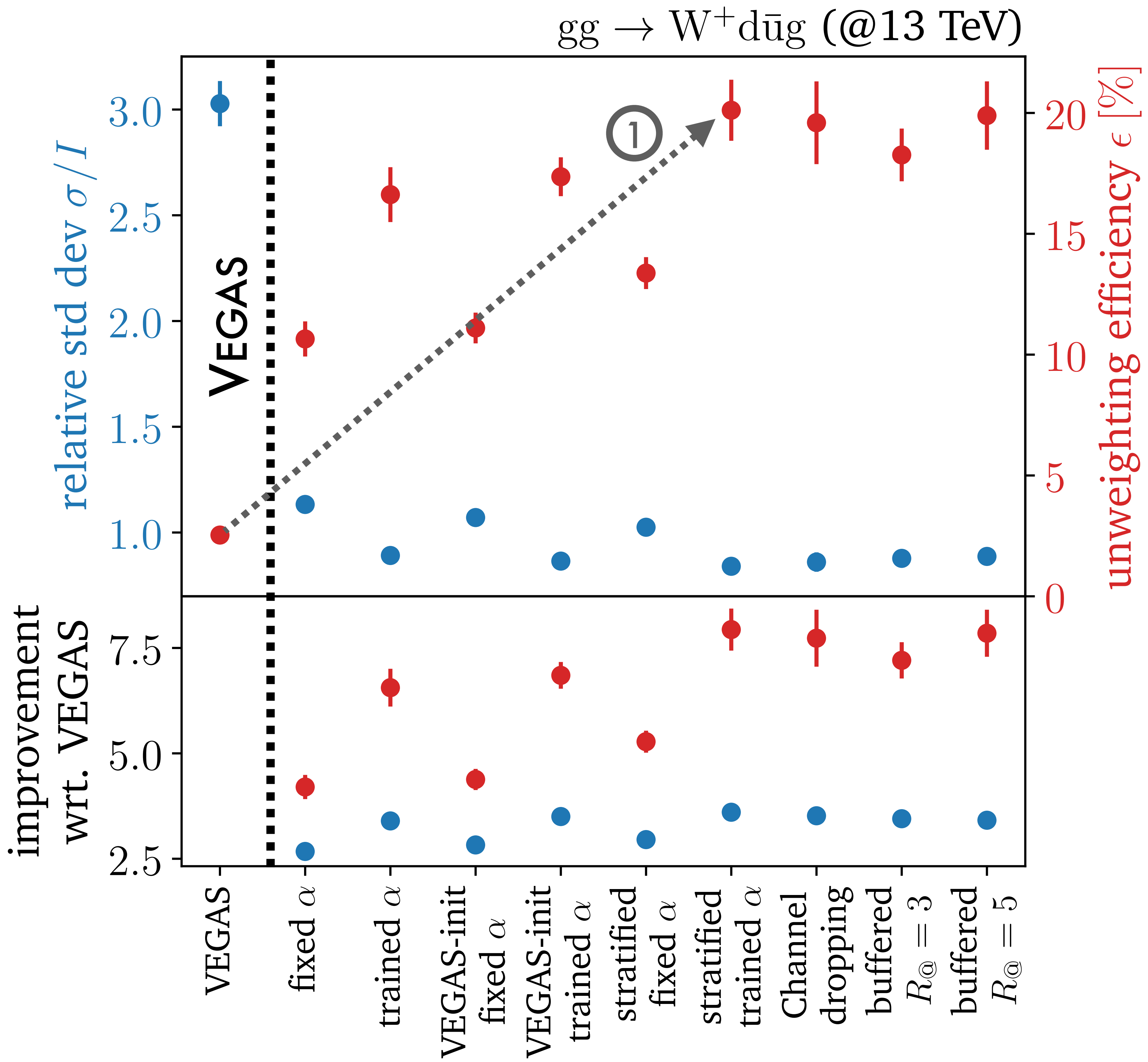
LHC processes

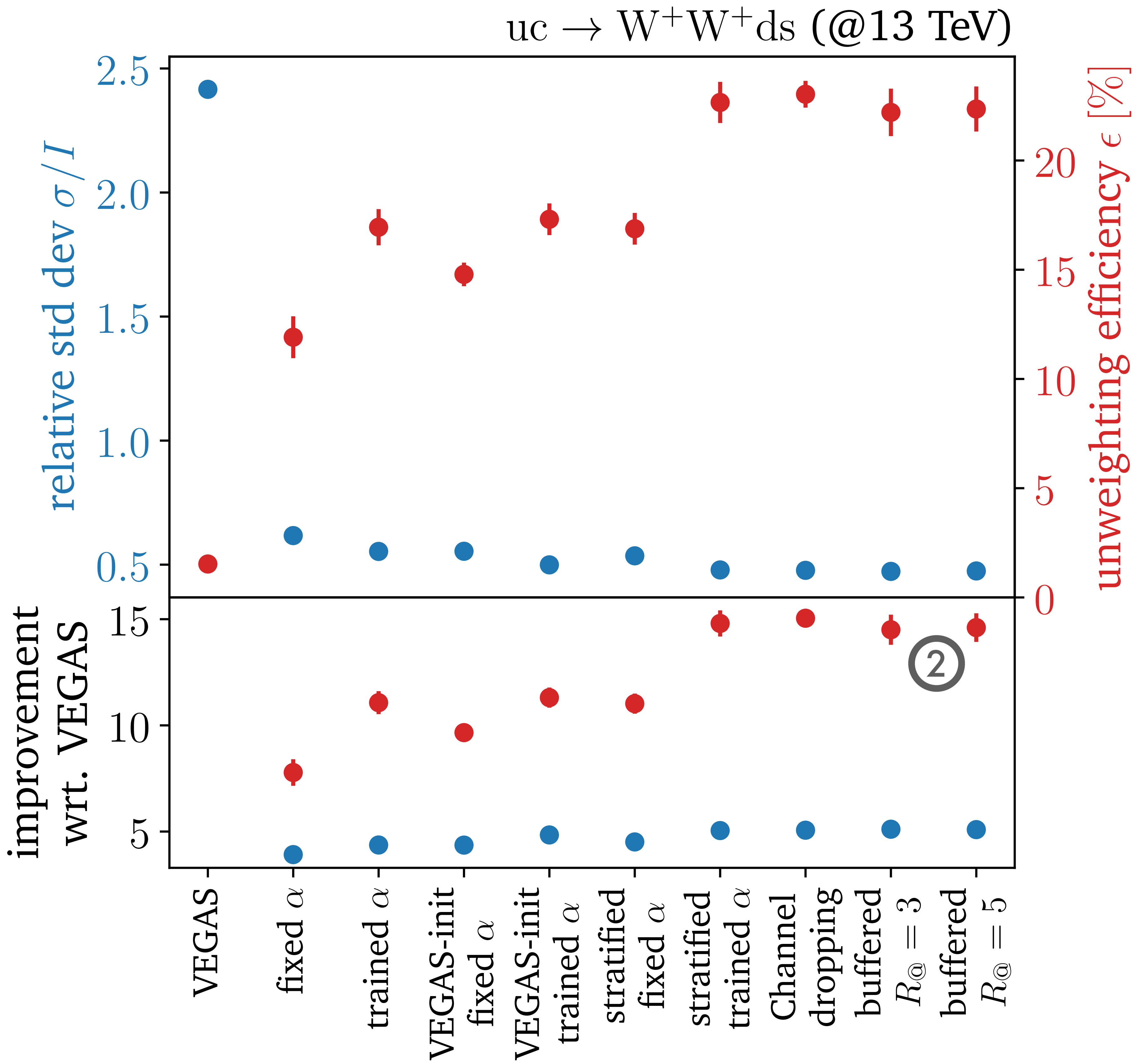




1. excellent results with all improvements

LHC processes





1. excellent results with all improvements
2. Larger improvements for processes with large interference terms


Machine Learning
Supervised Learning
Classification
Point Clouds [2102.05073]
PELICAN [2211.00454]
Energy Flow Networks [1810.05165]
Landscape of Top tagger [1902.09914]
Bayesian Tagger [1904.10004]
Regression
MadMiner [1907.10621,…]
NNPDF [2109.02653]
Matrix Elements [2206.14831]
Symbolic regression [2109.10414]
Reinforcement Learning
Flavor structure [2304.14176]
Simplifying Polylogs [2206.04115]
String vacua & landscape [1903.11616, 2111.11466]
Model Building [2103.04759]
Analytic continuation [2112.13011]
Unsupervised Learning
Clustering
Jet Clustering [2008.06064]
3D Pixel Clustering [2007.03083]
Anomaly Detection
Normalized AE [2206.14225]
CATHODE [2109.00546]
CWoLA Hunting [1902.02634]
(R-)ANODE [2001.04990,…]
Generative Models
NF
Precision Generation [2110.13632]
MEM [2210.00019]
MADNIS [2212.06172,..]
CaloFlow I-IV [2106.05285,…]
ELSA [2305.07696]
VAE
OTUS [2101.08944]
Jet Simulation [2203.00520]
Diffusion Models
PC-JeDi [2303.05376]
FPCD [2304.01266]
DDPM & CFM [2305.10475]
GAN
EPiC-GAN [2301.08128]
How to GAN [1907.03764]
CaloGAN [1712.10321]
Transformer
JetGPT [2305.10475]


Machine Learning
Supervised Learning
Classification
Point Clouds [2102.05073]
PELICAN [2211.00454]
Energy Flow Networks [1810.05165]
Landscape of Top tagger [1902.09914]
Bayesian Tagger [1904.10004]
Regression
MadMiner [1907.10621,…]
NNPDF [2109.02653]
Matrix Elements [2206.14831]
Symbolic regression [2109.10414]
Reinforcement Learning
Flavor structure [2304.14176]
Simplifying Polylogs [2206.04115]
String vacua & landscape [1903.11616, 2111.11466]
Model Building [2103.04759]
Analytic continuation [2112.13011]
Unsupervised Learning
Generative Models
VAE
OTUS [2101.08944]
Jet Simulation [2203.00520]
NF
Precision Generation [2110.13632]
MEM [2210.00019]
MADNIS [2212.06172,..]
CaloFlow I-IV [2106.05285,…]
ELSA [2305.07696]
Diffusion Models
PC-JeDi [2303.05376]
FPCD [2304.01266]
DDPM & CFM [2305.10475]
GAN
EPiC-GAN [2301.08128]
How to GAN [1907.03764]
CaloGAN [1712.10321]
Transformer
JetGPT [2305.10475]
Clustering
Jet Clustering [2008.06064]
3D Pixel Clustering [2007.03083]
Anomaly Detection
Normalized AE [2206.14225]
CATHODE [2109.00546]
CWoLA Hunting [1902.02634]
(R-)ANODE [2001.04990,…]

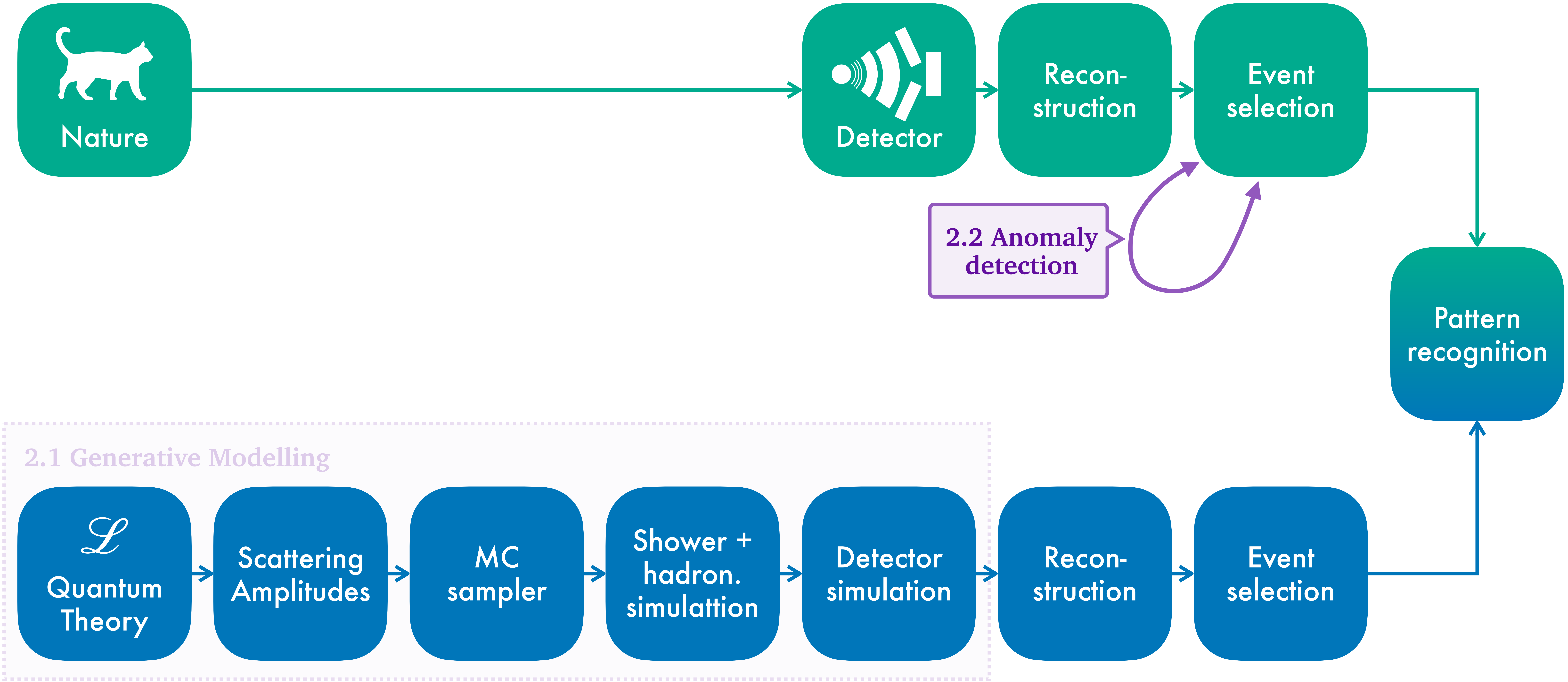
Break & Questions

5 Minutes

Part III

Anomaly Detection

Reminder — LHC analysis + ML



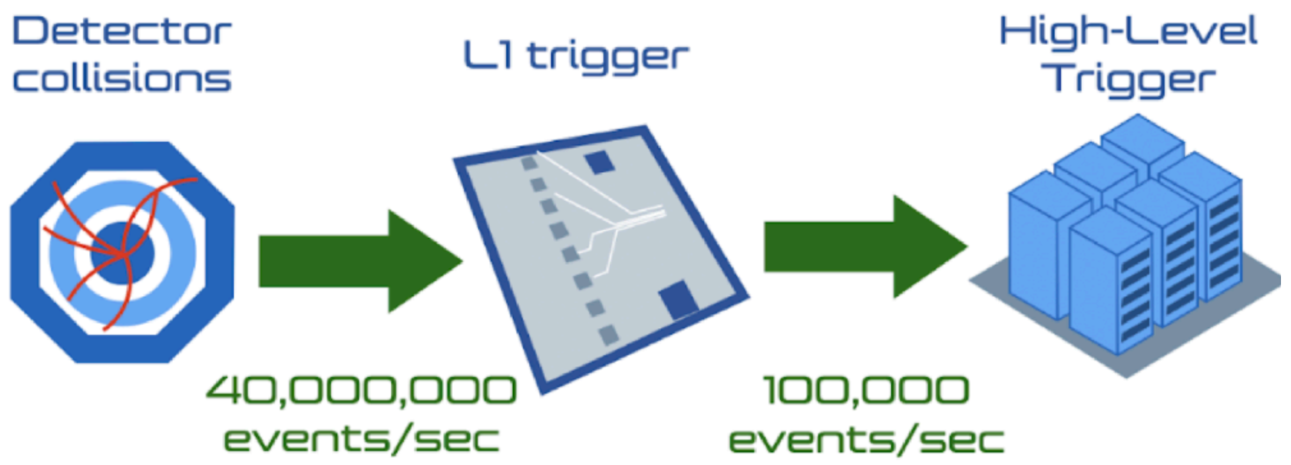
Community interest in AD

LHC Olympics

[Kasieczka et al: 2107.02821, 2101.08320]

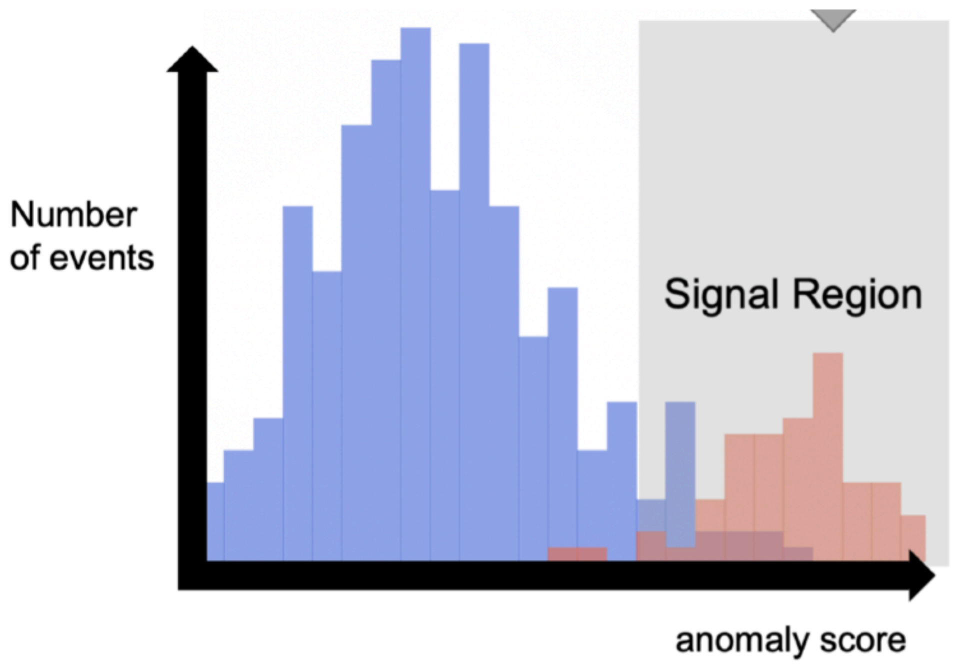
ADC2021

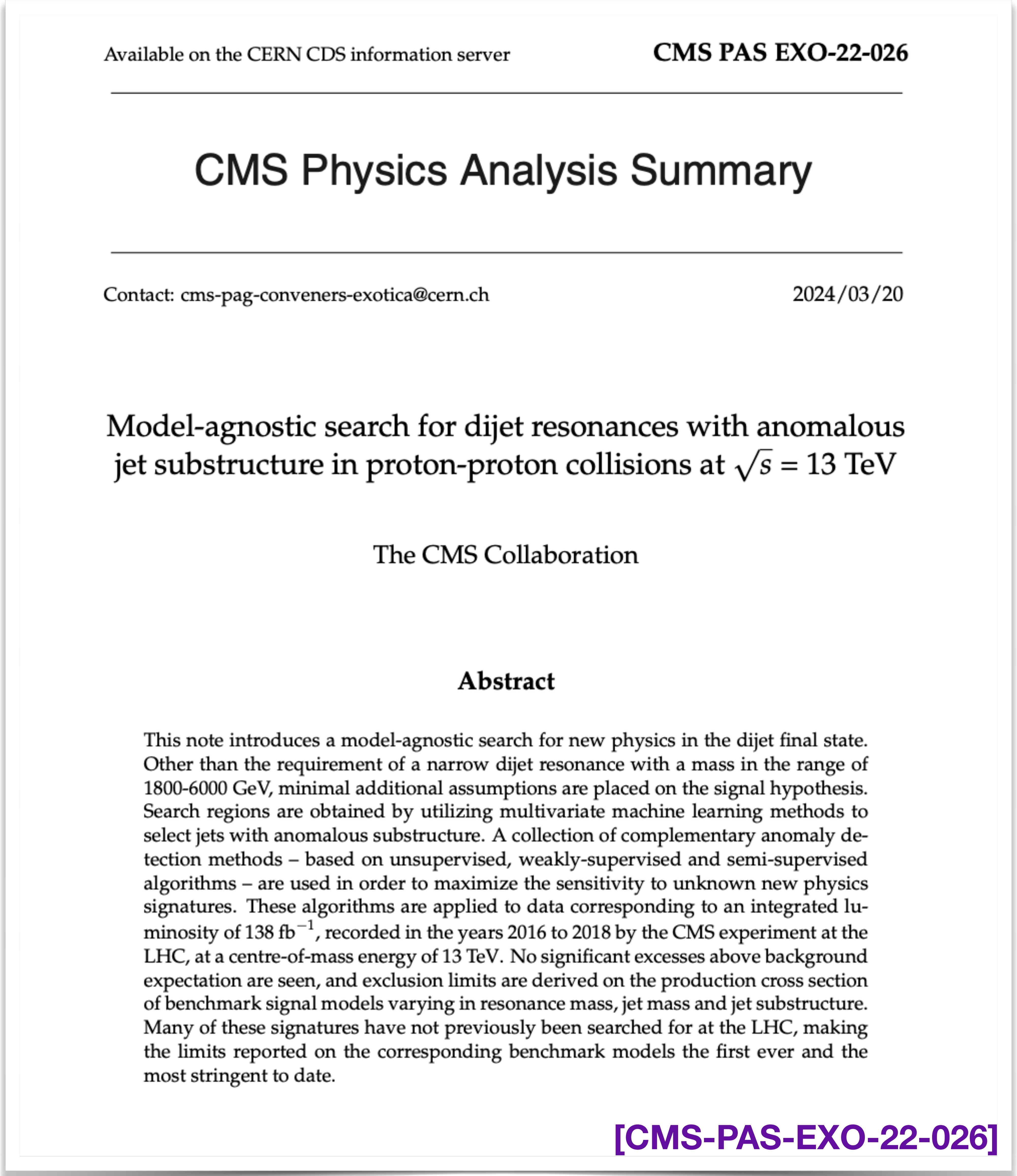
[Govorkova et al: 2107.02157]



Dark Machines

[Ostdiek et al: 2105.14027]





Available on the CERN CDS information server — **CMS PAS EXO-22-026**

CMS Physics Analysis Summary

Contact: cms-pag-conveners-exotica@cern.ch — 2024/03/20

Model-agnostic search for dijet resonances with anomalous jet substructure in proton-proton collisions at $\sqrt{s} = 13$ TeV

The CMS Collaboration

Abstract

This note introduces a model-agnostic search for new physics in the dijet final state. Other than the requirement of a narrow dijet resonance with a mass in the range of 1800-6000 GeV, minimal additional assumptions are placed on the signal hypothesis. Search regions are obtained by utilizing multivariate machine learning methods to select jets with anomalous substructure. A collection of complementary anomaly detection methods – based on unsupervised, weakly-supervised and semi-supervised algorithms – are used in order to maximize the sensitivity to unknown new physics signatures. These algorithms are applied to data corresponding to an integrated luminosity of 138 fb^{-1}, recorded in the years 2016 to 2018 by the CMS experiment at the LHC, at a centre-of-mass energy of 13 TeV. No significant excesses above background expectation are seen, and exclusion limits are derived on the production cross section of benchmark signal models varying in resonance mass, jet mass and jet substructure. Many of these signatures have not previously been searched for at the LHC, making the limits reported on the corresponding benchmark models the first ever and the most stringent to date.

[CMS-PAS-EXO-22-026]

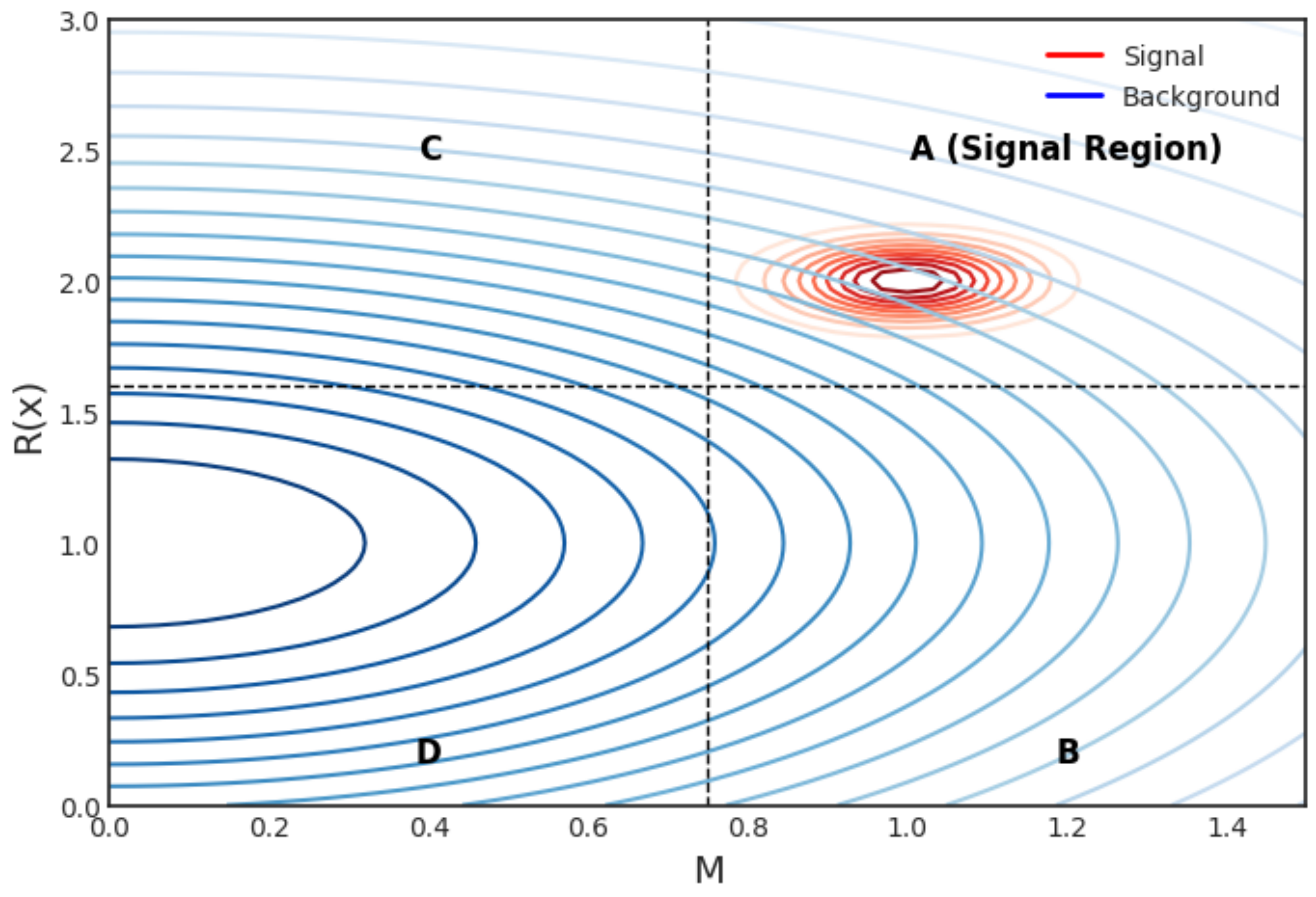
What is anomaly detection?

Two Types of Anomaly Detection

Outlier Detection
(non-resonant)

- Searching for unique and unexpected events
- In HEP, this (might) appear in the tails of dist.

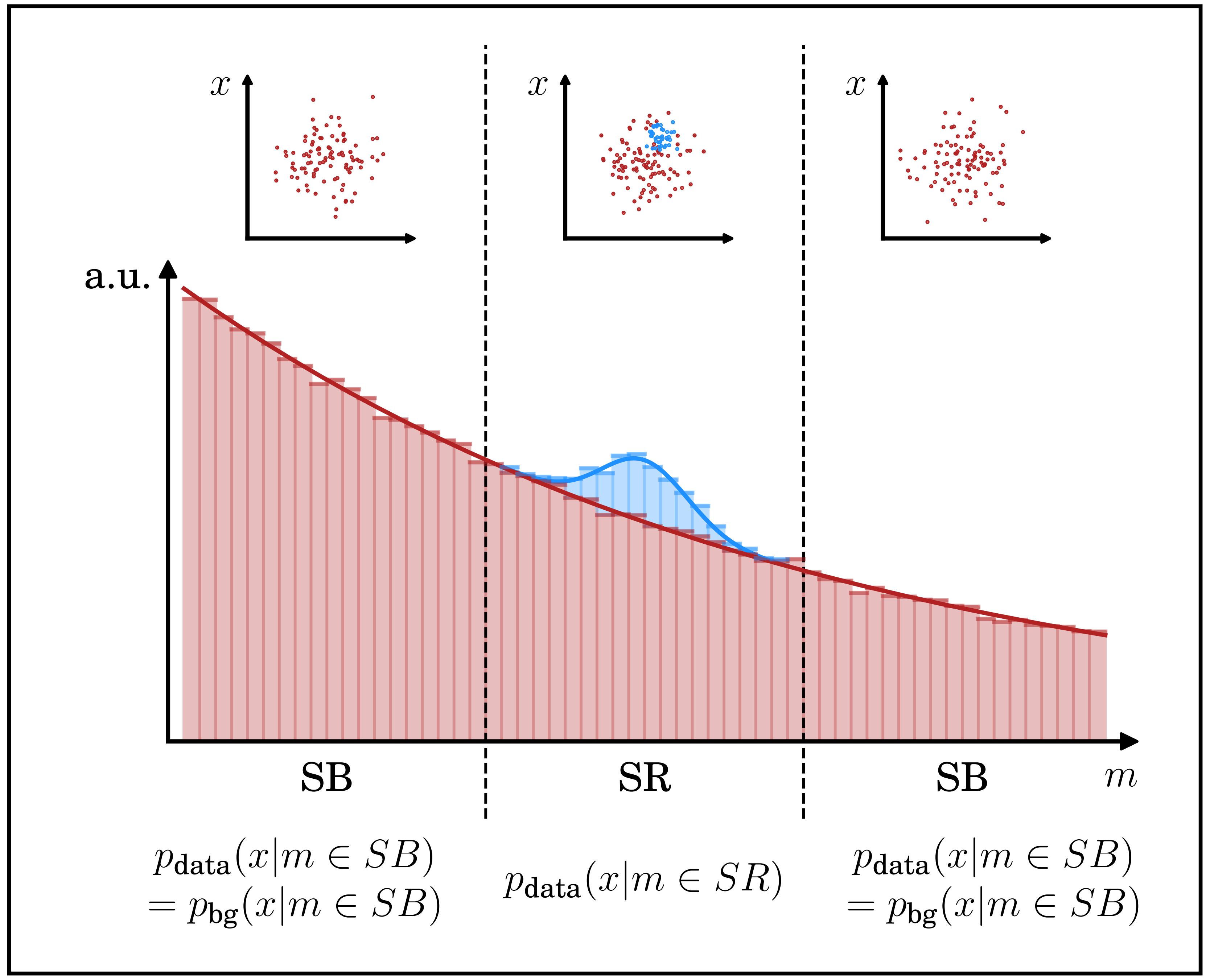
[2404.07258]



Overdensities
(resonant)

- Analagous to traditional bump hunt

[2109.00546]



Two Types of Anomaly Detection

Outlier Detection
(non-resonant)

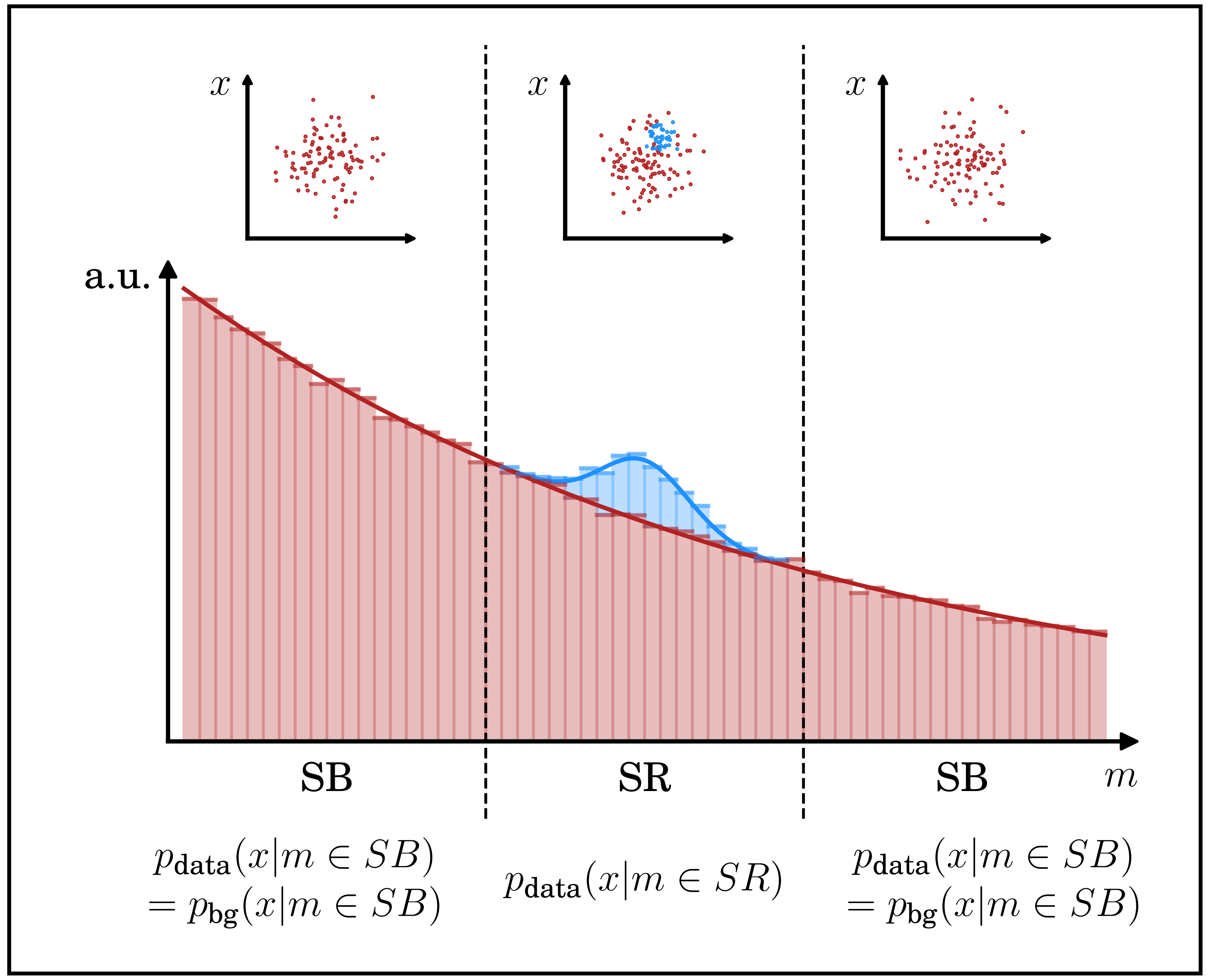
- Searching for unique and unexpected events
- In HEP, this (might) appear in the tails of dist.

[2404.07258]

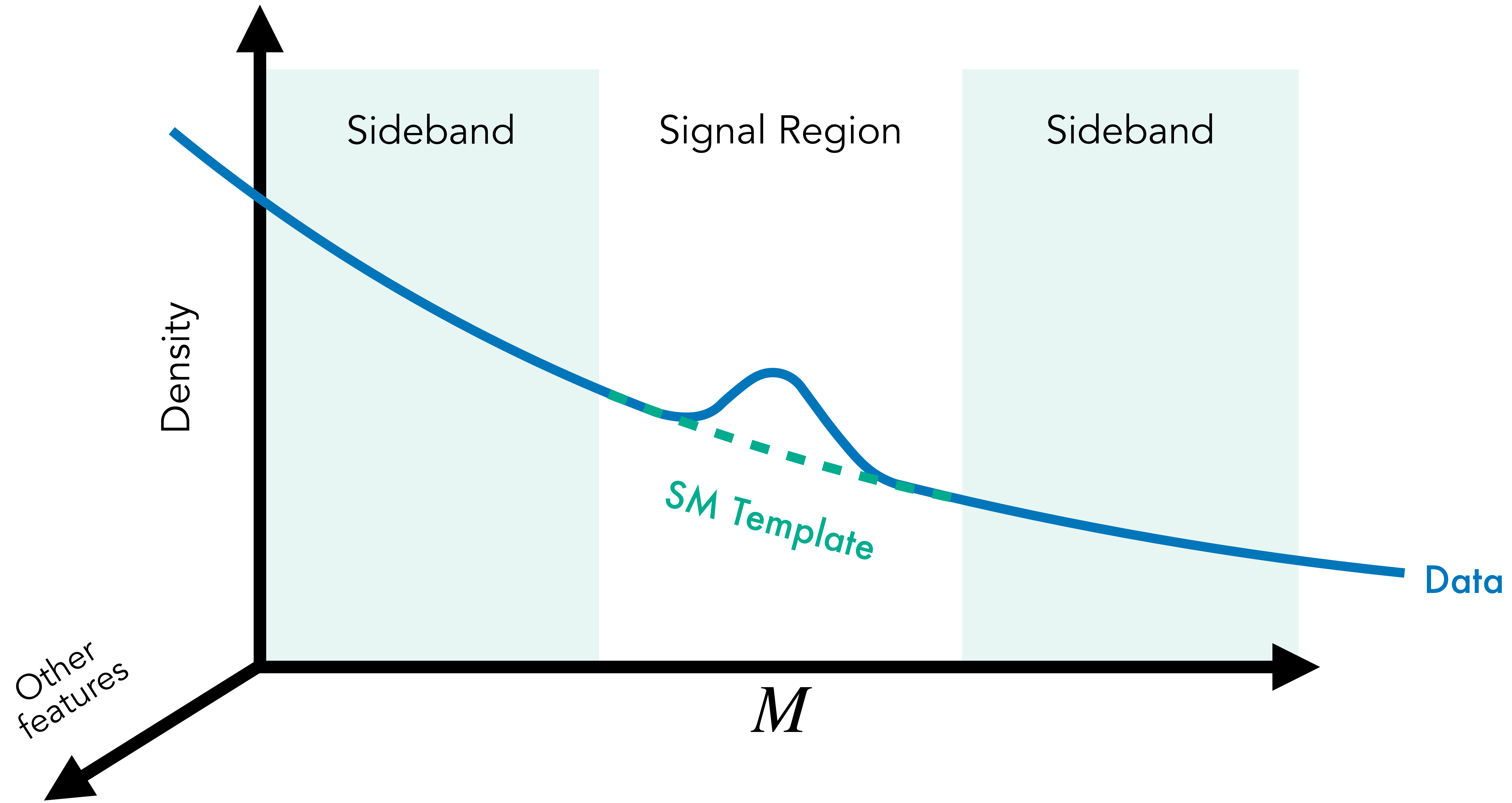
Overdensities
(resonant)

- Analagous to traditional bump hunt

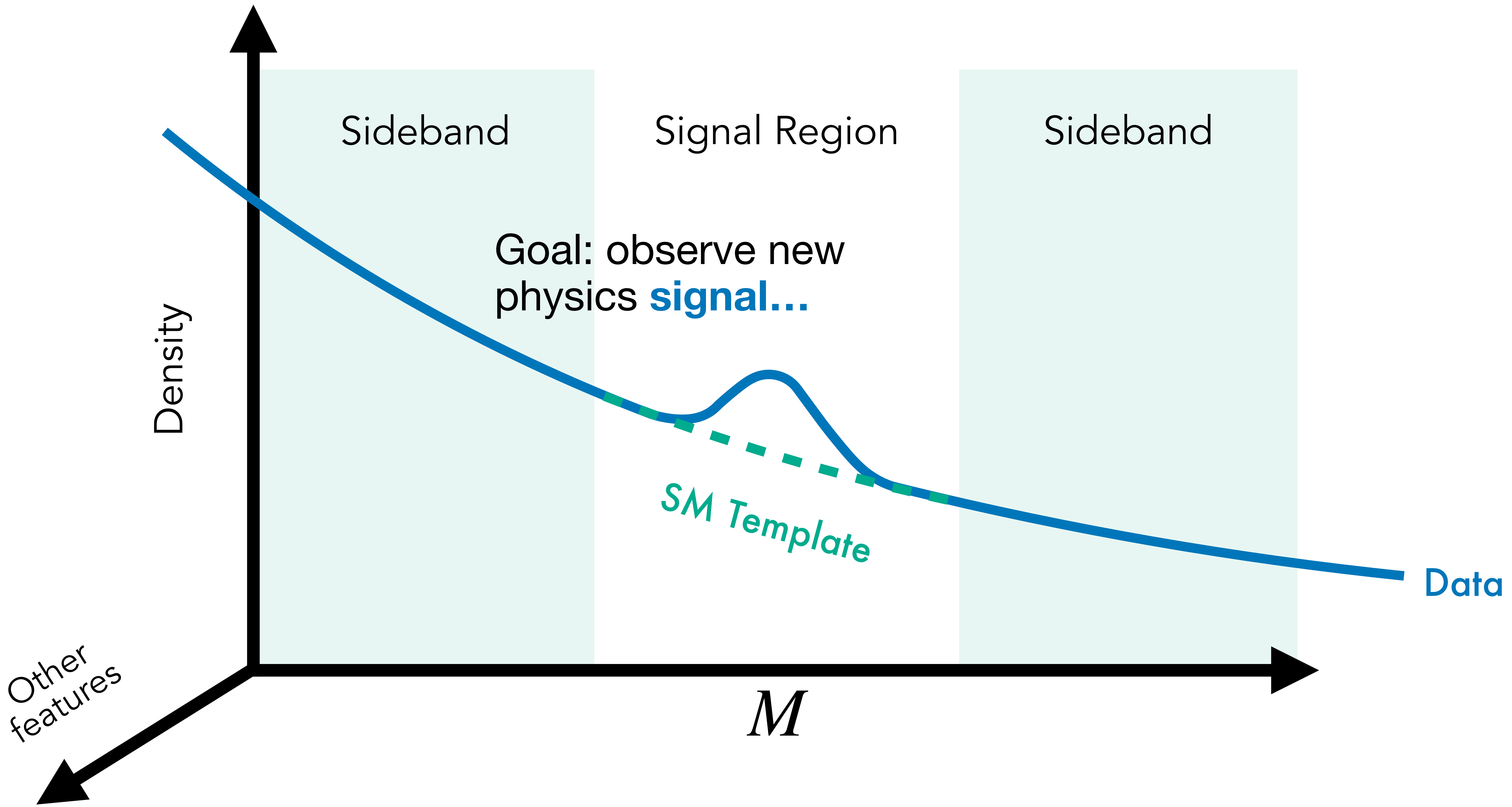
[2109.00546]



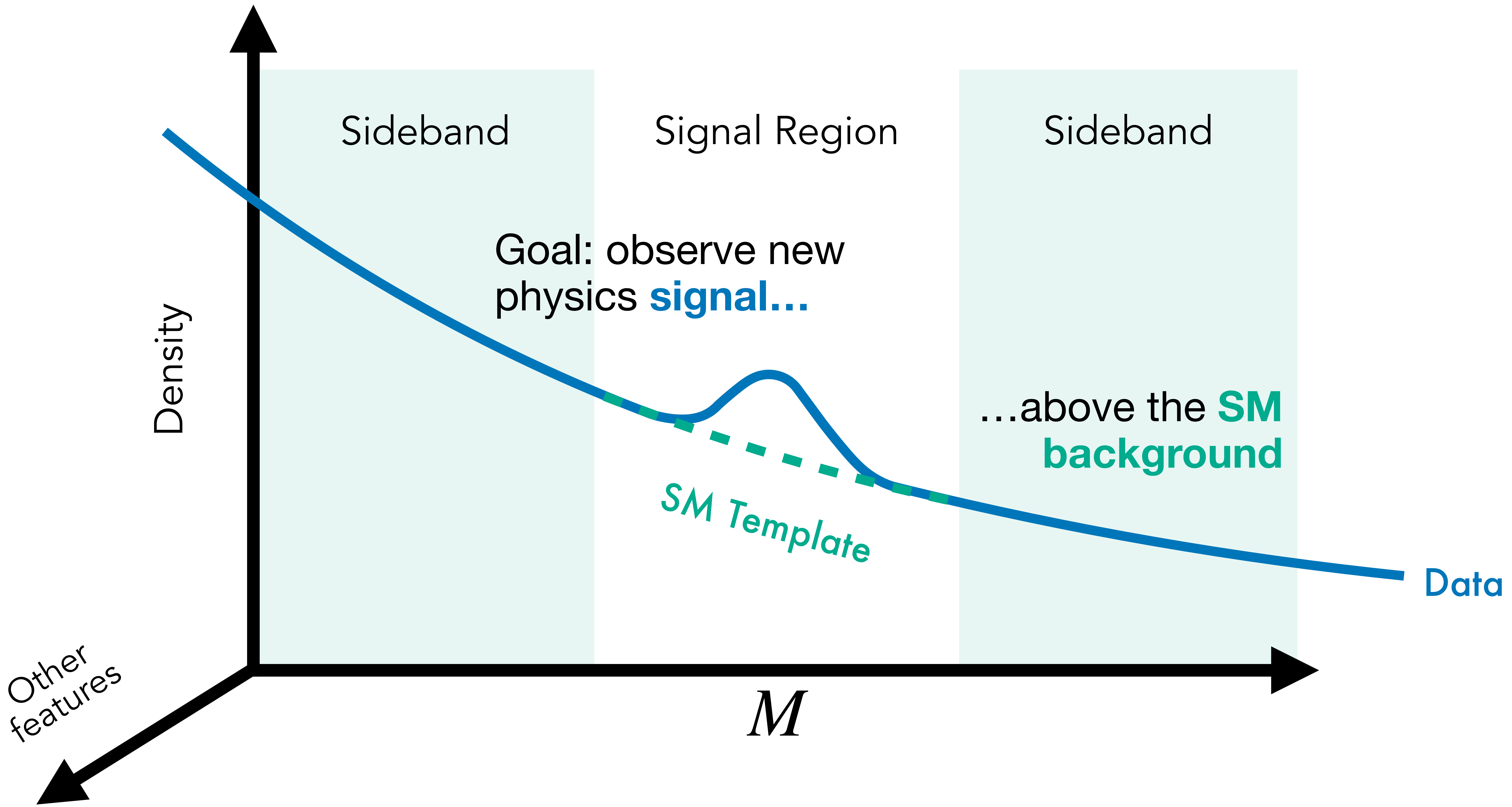
Resonant AD as a search strategy



Resonant AD as a search strategy



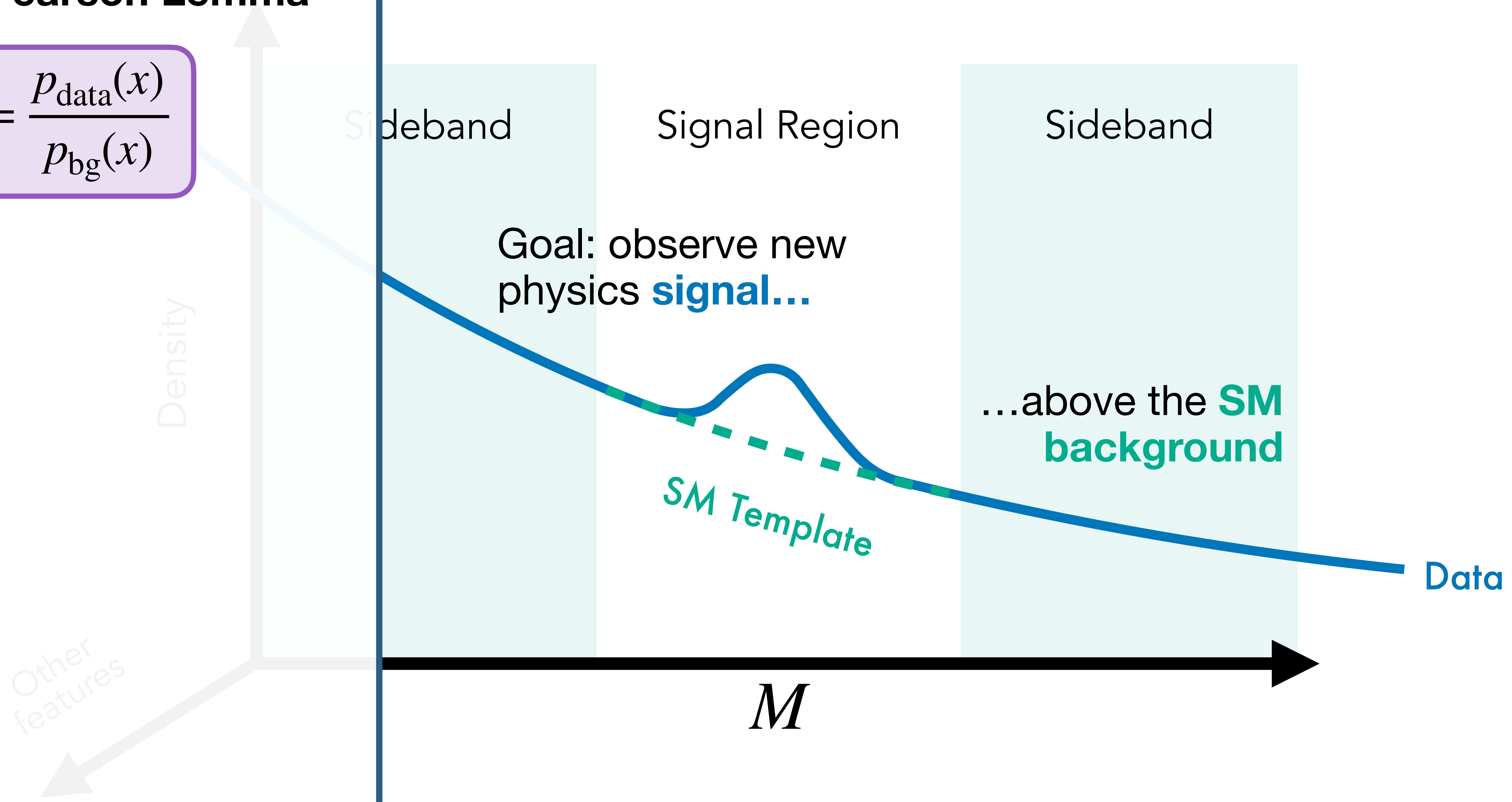
Resonant AD as a search strategy



Resonant AD as a search strategy

Neyman-Pearson Lemma

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

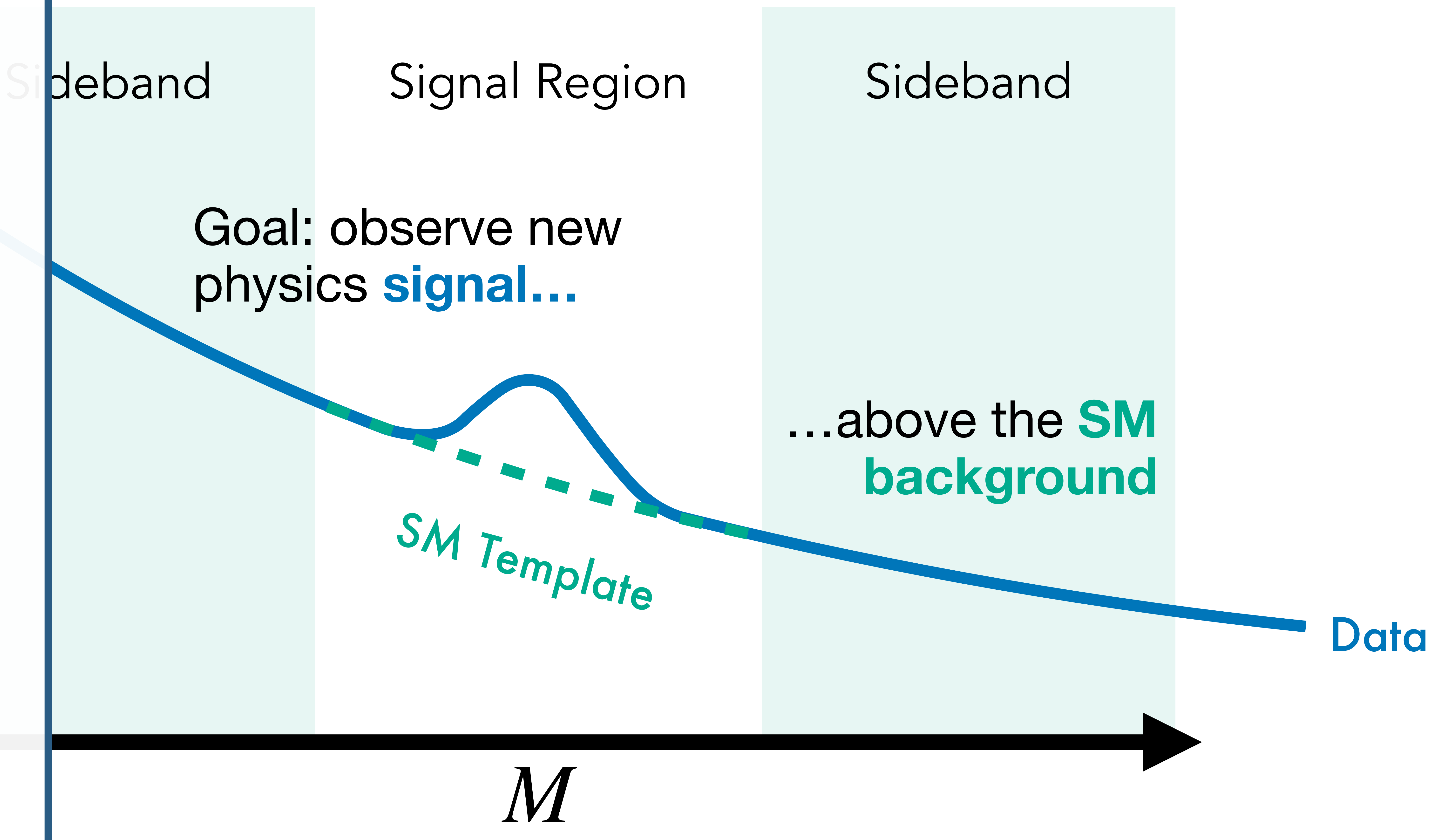


Resonant AD as a search strategy

Neyman-Pearson Lemma

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

Optimal hypothesis test



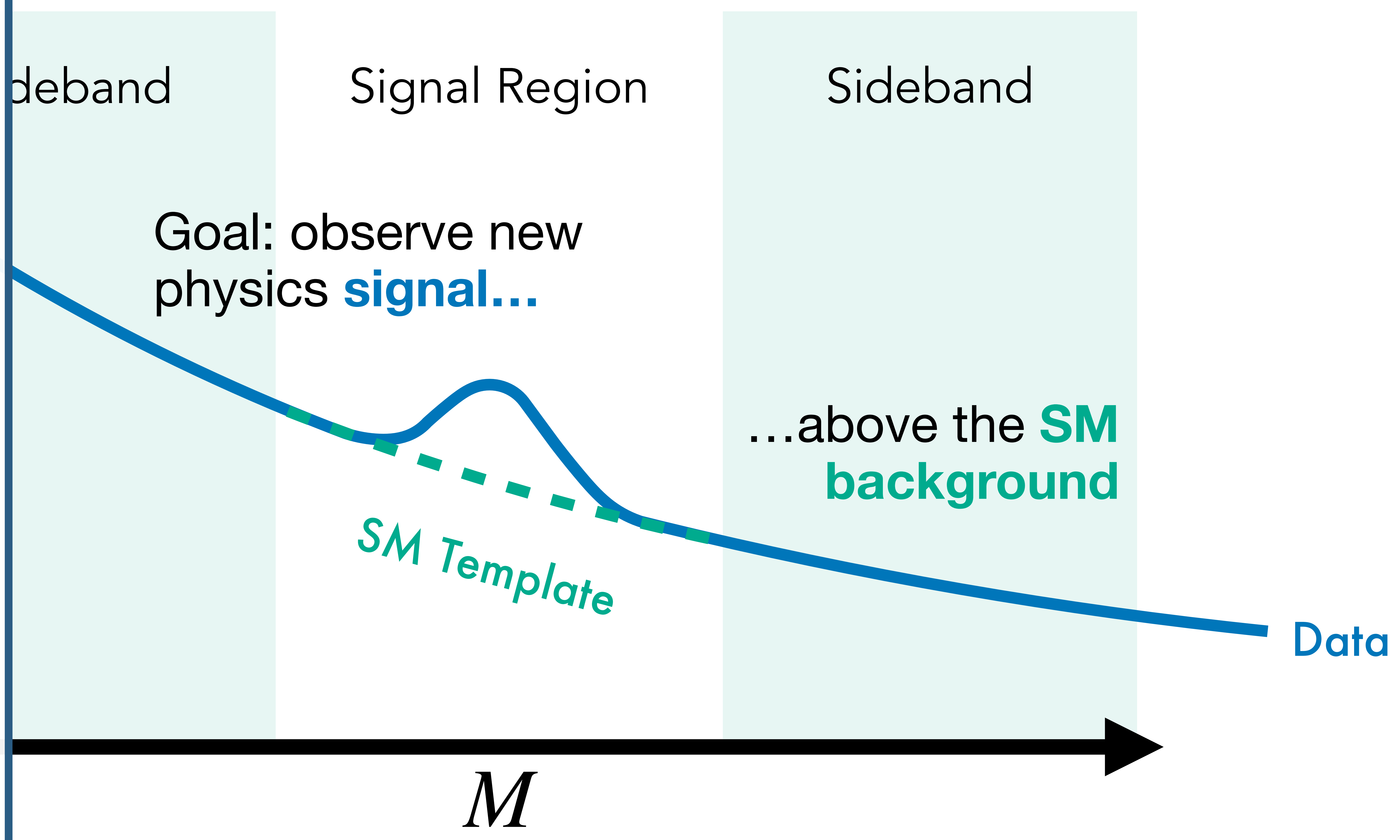
Resonant AD as a search strategy

Neyman-Pearson Lemma

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

Optimal hypothesis test

❖ Idealized anomaly detector (IAD)



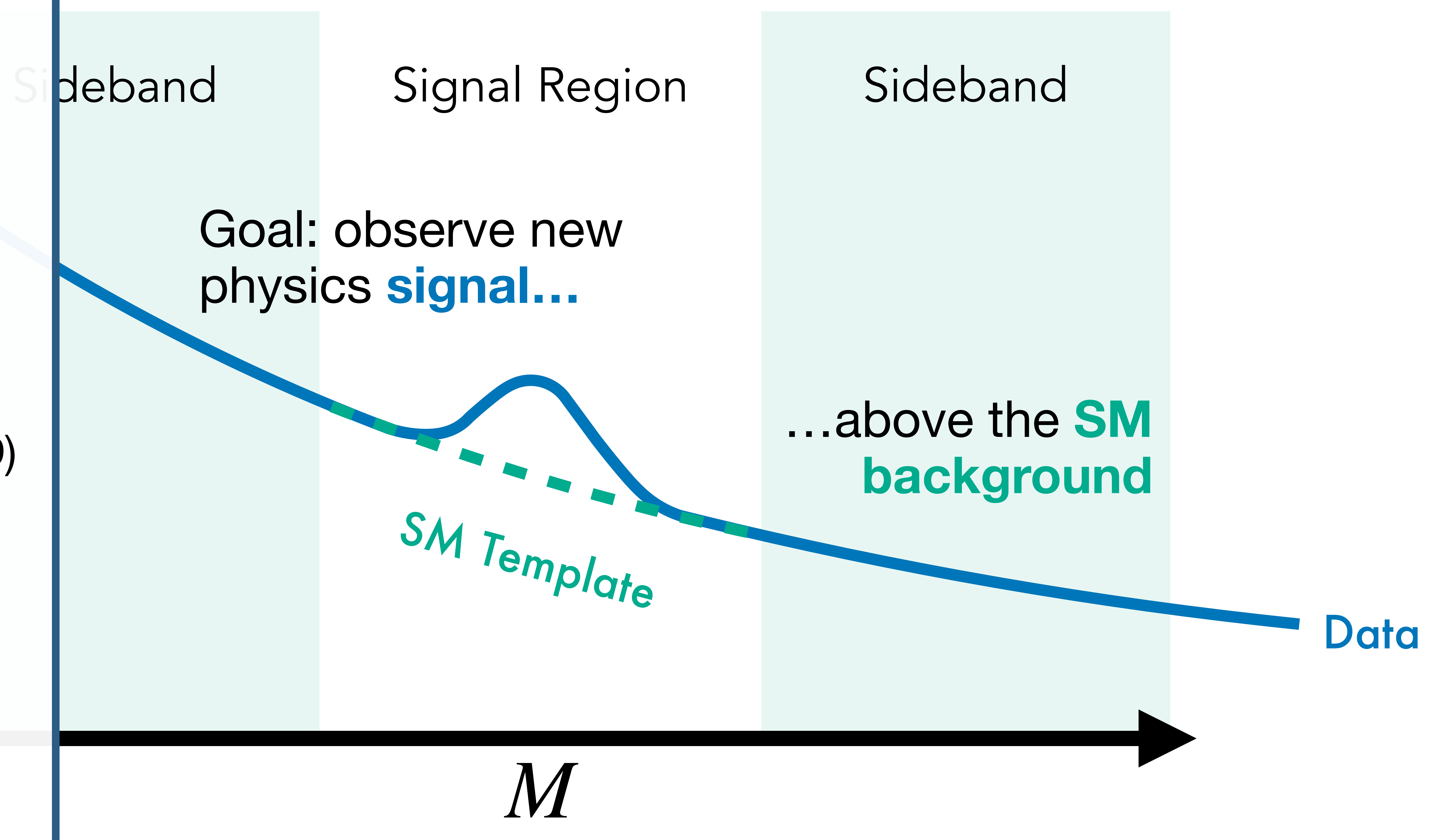
Resonant AD as a search strategy

Neyman-Pearson Lemma

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

Optimal hypothesis test

- Idealized anomaly detector (IAD)
- Best you can do **if…**
  …you know p_{data} and p_{bg}



Resonant AD as a search strategy

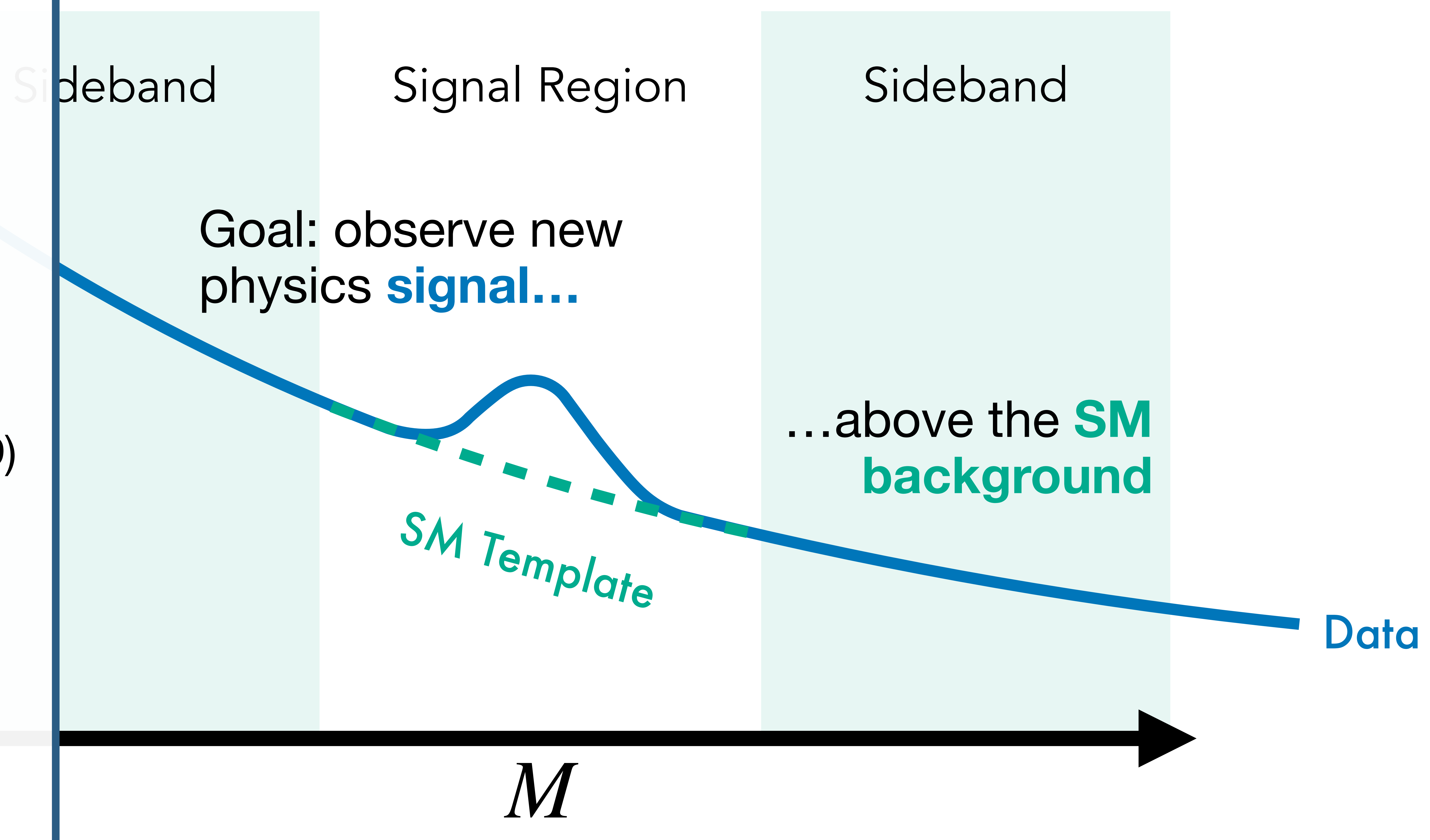
Neyman-Pearson Lemma

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

Optimal hypothesis test

- Idealized anomaly detector (IAD)
- Best you can do **if…**
  …you know p_{data} and p_{bg}

ML



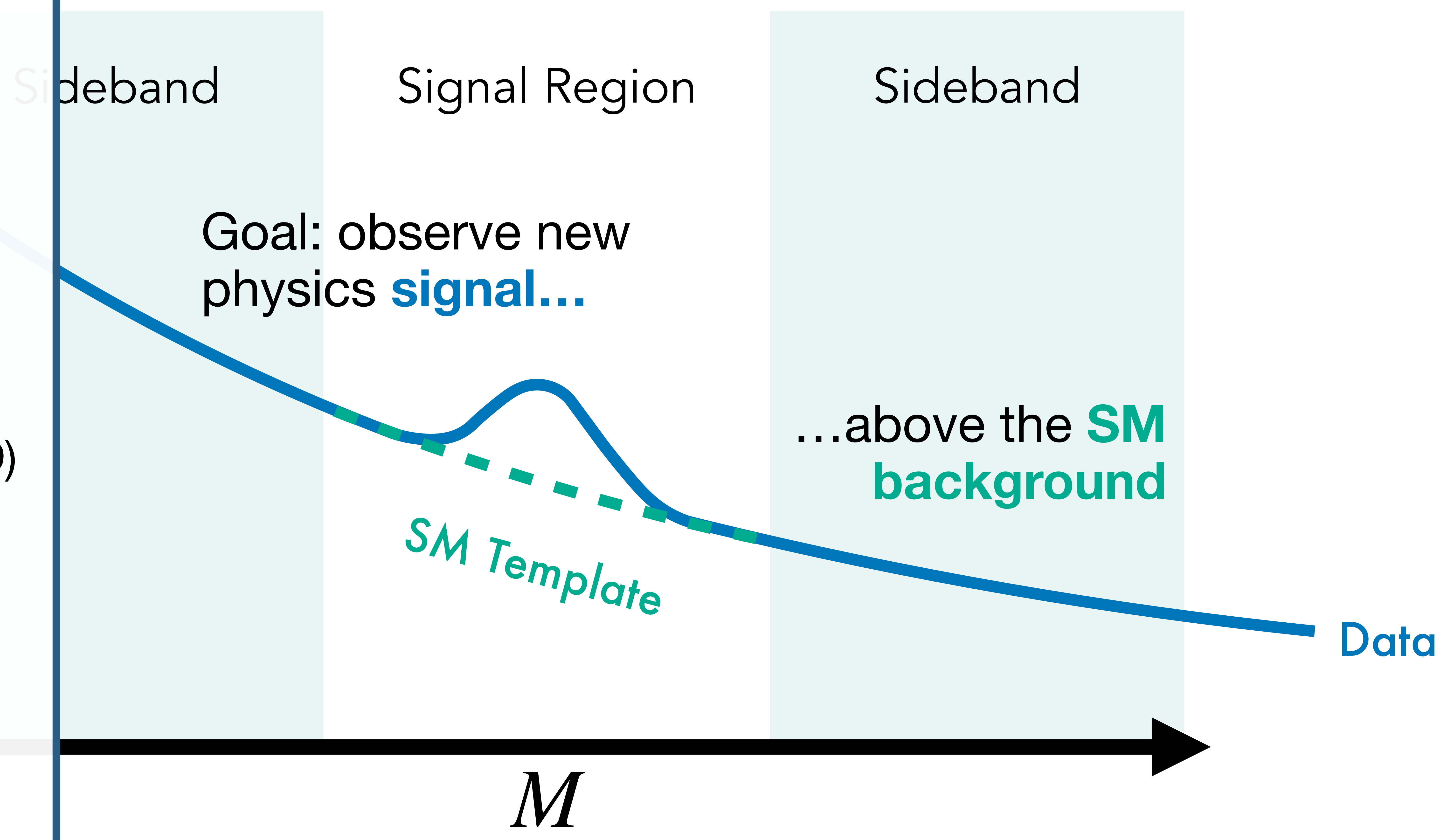
Resonant AD as a search strategy

Neyman-Pearson Lemma

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

Optimal hypothesis test

- Idealized anomaly detector (IAD)
- Best you can do **if…**
  …you know p_{data} and p_{bg} (ML)
- Use R as cut discriminant
  → $R > R_c$



How to get the optimal test statistic?

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

How to get the optimal test statistic?

Classifier

If we have samples from
data and **SM background…**

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

How to get the optimal test statistic?

Classifier

If we have samples from **data** and **SM background…**

…an **optimal classifier** yields

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

←

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

How to get the optimal test statistic?

Classifier

If we have samples from **data** and **SM background…**

…an **optimal classifier** yields

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

- Get $x \sim p_{\text{data}}$ and $x \sim p_{\text{bg}}$ from **MC simulations**

←

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

How to get the optimal test statistic?

Classifier

If we have samples from **data** and **SM background…**

…an **optimal classifier** yields

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

- Get $x \sim p_{\text{data}}$ and $x \sim p_{\text{bg}}$ from **MC simulations**
- Estimate samples from **data**:

$$x \sim p_{\text{data}}(x \,|\, \text{SR})$$

$$x \sim p_{\text{data}}(x \,|\, \text{SB}) \approx p_{\text{bg}}(x)$$

←

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

How to get the optimal test statistic?

Classifier

If we have samples from
data and **SM background…**

…an **optimal classifier** yields

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

- Get $x \sim p_{\text{data}}$ and $x \sim p_{\text{bg}}$ from **MC simulations**
- Estimate samples from **data**:

$$x \sim p_{\text{data}}(x \,|\, \text{SR})$$

$$x \sim p_{\text{data}}(x \,|\, \text{SB}) \approx p_{\text{bg}}(x)$$

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

Density estimator

Instead of learning the
likelihood ratio directly…

How to get the optimal test statistic?

Classifier

If we have samples from **data** and **SM background…**

…an **optimal classifier** yields

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

- Get $x \sim p_{\text{data}}$ and $x \sim p_{\text{bg}}$ from **MC simulations**
- Estimate samples from **data**:

$$x \sim p_{\text{data}}(x \,|\, \text{SR})$$

$$x \sim p_{\text{data}}(x \,|\, \text{SB}) \approx p_{\text{bg}}(x)$$

$$R = \frac{p_{\text{data}}(x)}{p_{\text{bg}}(x)}$$

Density estimator

Instead of learning the **likelihood ratio** directly…

…use a **density estimator** to learn

$$p_\omega(x \,|\, \text{SR}) \simeq p_{\text{data}}(x \,|\, \text{SR})$$

$$p_\omega(x \,|\, \text{SB}) \simeq p_{\text{bg}}(x)$$

How to get the optimal test statistic?

Classifier

If we have samples from **data** and **SM background…**

…an **optimal classifier** yields

$$f(x) = \frac{p_{\rm data}(x)}{p_{\rm data}(x) + p_{\rm bg}(x)}$$

- Get $x \sim p_{\rm data}$ and $x \sim p_{\rm bg}$ from **MC simulations**
- Estimate samples from **data**:

$$x \sim p_{\rm data}(x\,|\,{\rm SR})$$

$$x \sim p_{\rm data}(x\,|\,{\rm SB}) \approx p_{\rm bg}(x)$$

$$R = \frac{p_{\rm data}(x)}{p_{\rm bg}(x)}$$

Density estimator

Instead of learning the **likelihood ratio** directly…

…use a **density estimator** to learn

$$p_\omega(x\,|\,{\rm SR}) \simeq p_{\rm data}(x\,|\,{\rm SR})$$

$$p_\omega(x\,|\,{\rm SB}) \simeq p_{\rm bg}(x)$$

- Then **calculate** R directly from the individual likelihoods

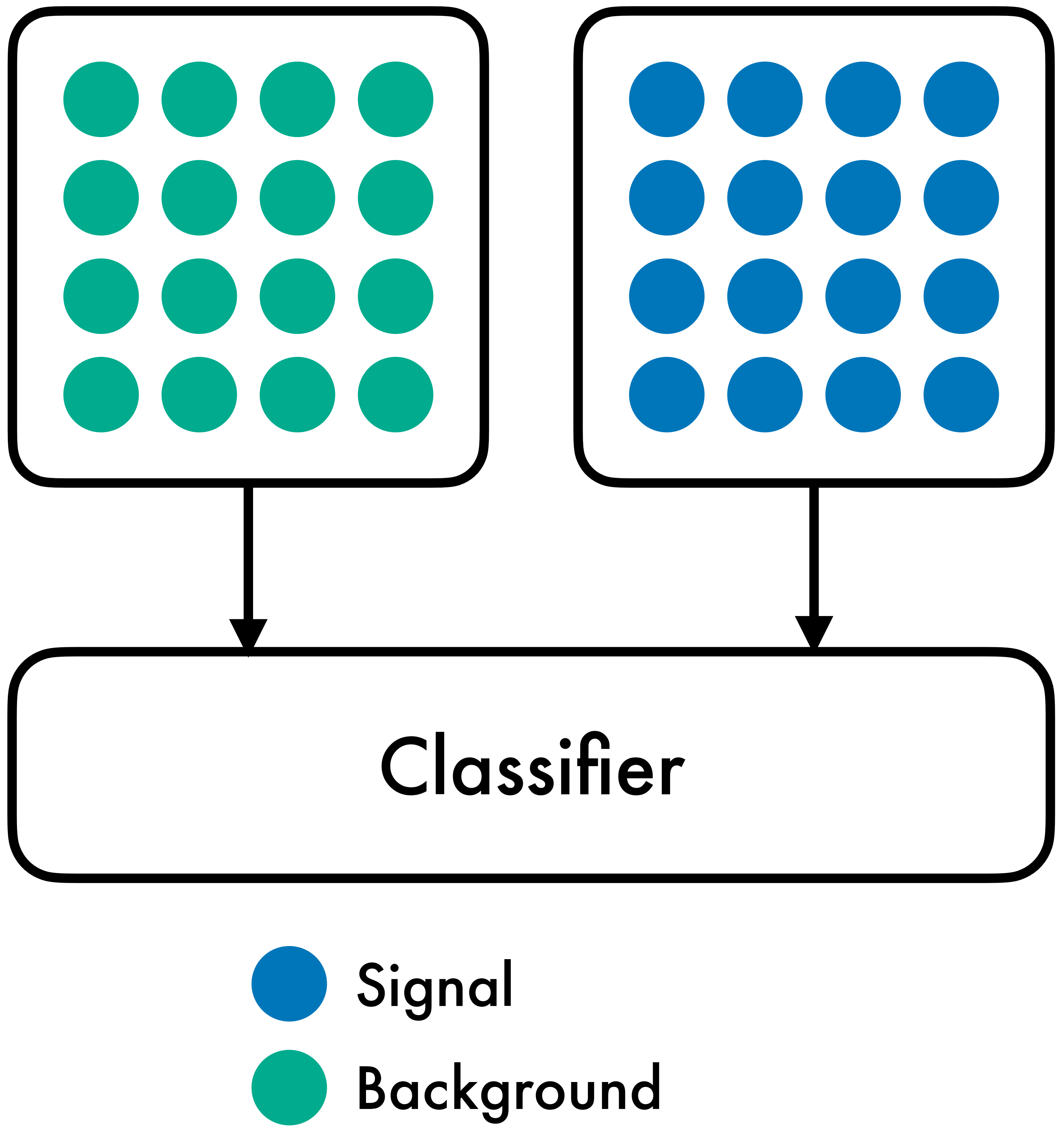
Example I

CWoLa Hunting

Metodiev, Nachman, Thaler [1708.02949]
Collins, Howe, Nachman [1805.02664]

Reminder — Classification Problem

Goal: learn the signal to background ratio

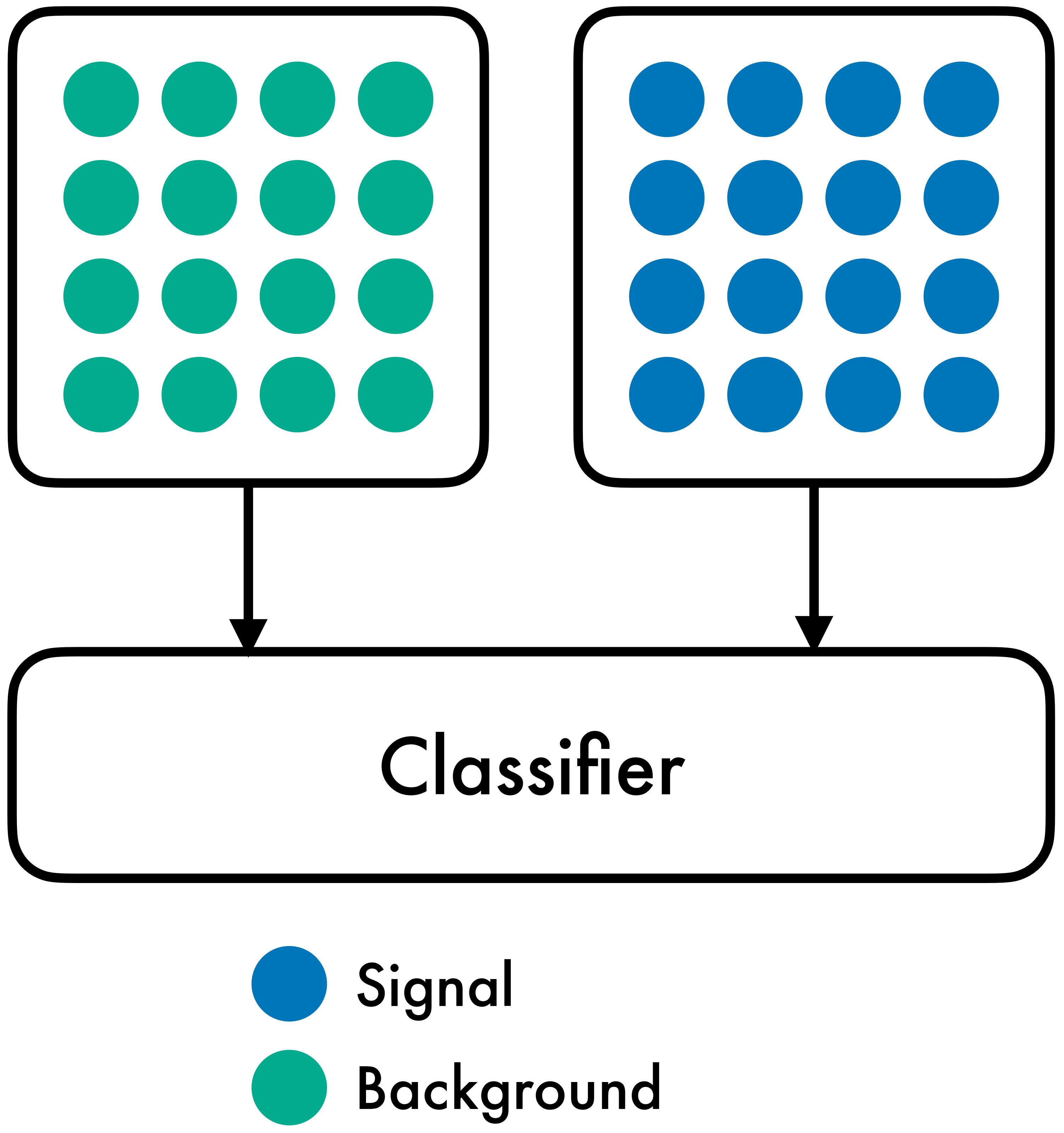


Reminder — Classification Problem

Goal: learn the signal to background ratio

An optimal classifier yields the likelihood ratio

$$R_{\text{optimal}} = \frac{f(x)}{1 - f(x)} = \frac{p_{\text{sig}}(x)}{p_{\text{bg}}(x)}$$



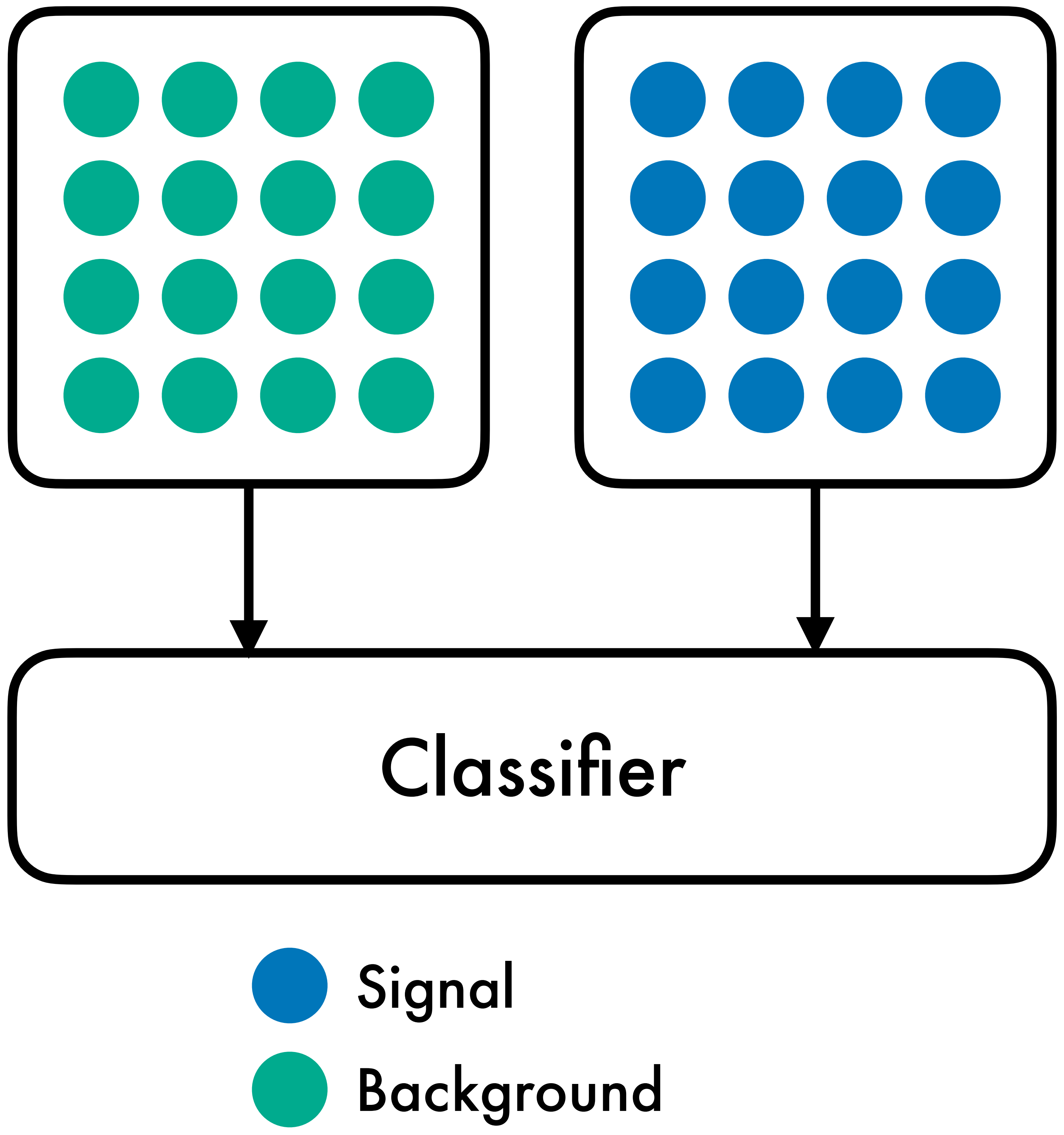
Reminder — Classification Problem

Goal: learn the signal to background ratio

An optimal classifier yields the likelihood ratio

$$R_{\text{optimal}} = \frac{f(x)}{1 - f(x)} = \frac{p_{\text{sig}}(x)}{p_{\text{bg}}(x)}$$

⊕ Can be approximated with a **supervised classifier (ML)**



Reminder — Classification Problem

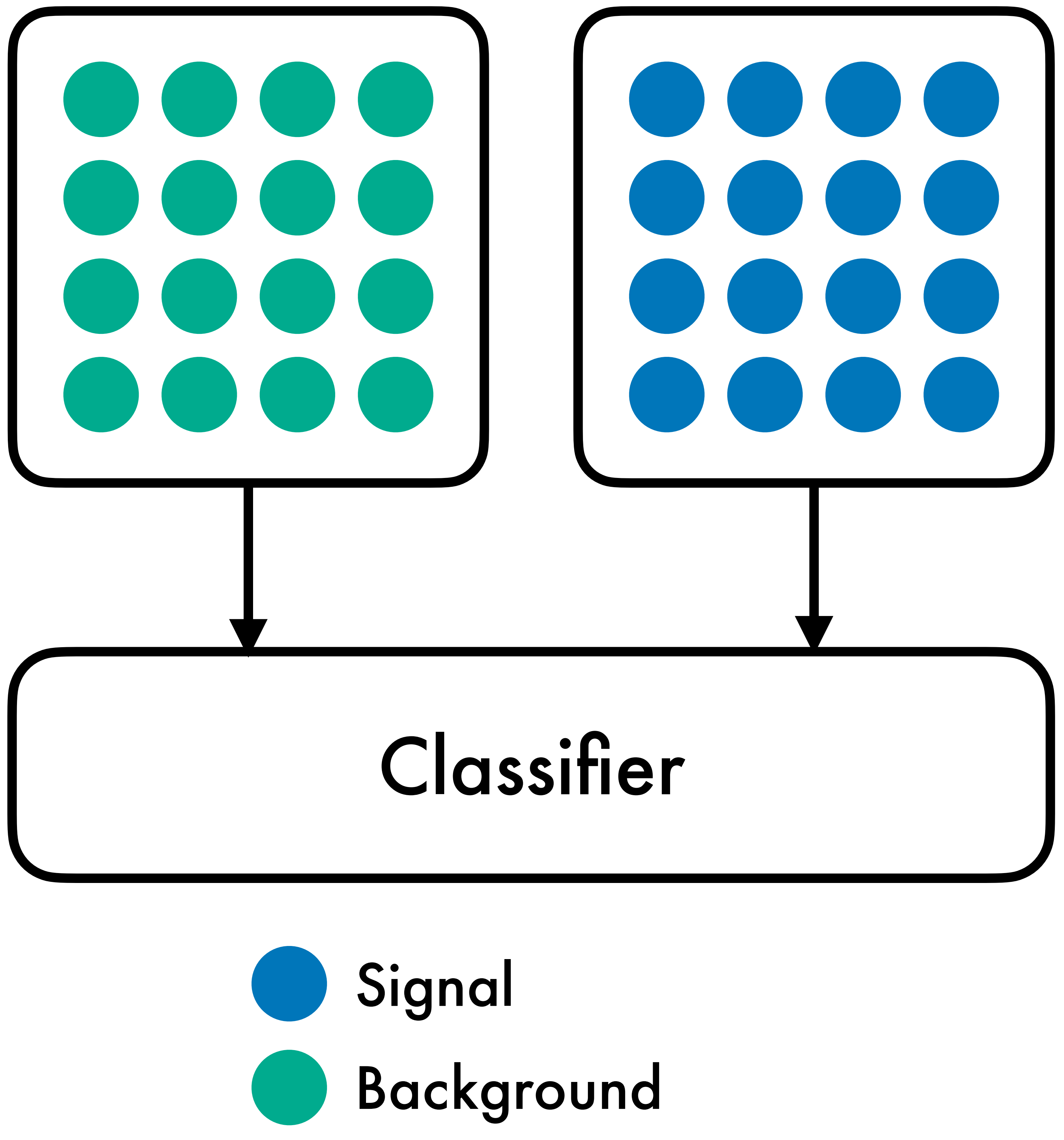
Goal: learn the signal to background ratio

An optimal classifier yields the likelihood ratio

$$R_{\text{optimal}} = \frac{f(x)}{1 - f(x)} = \frac{p_{\text{sig}}(x)}{p_{\text{bg}}(x)}$$

⊕ Can be approximated with a **supervised classifier (ML)**

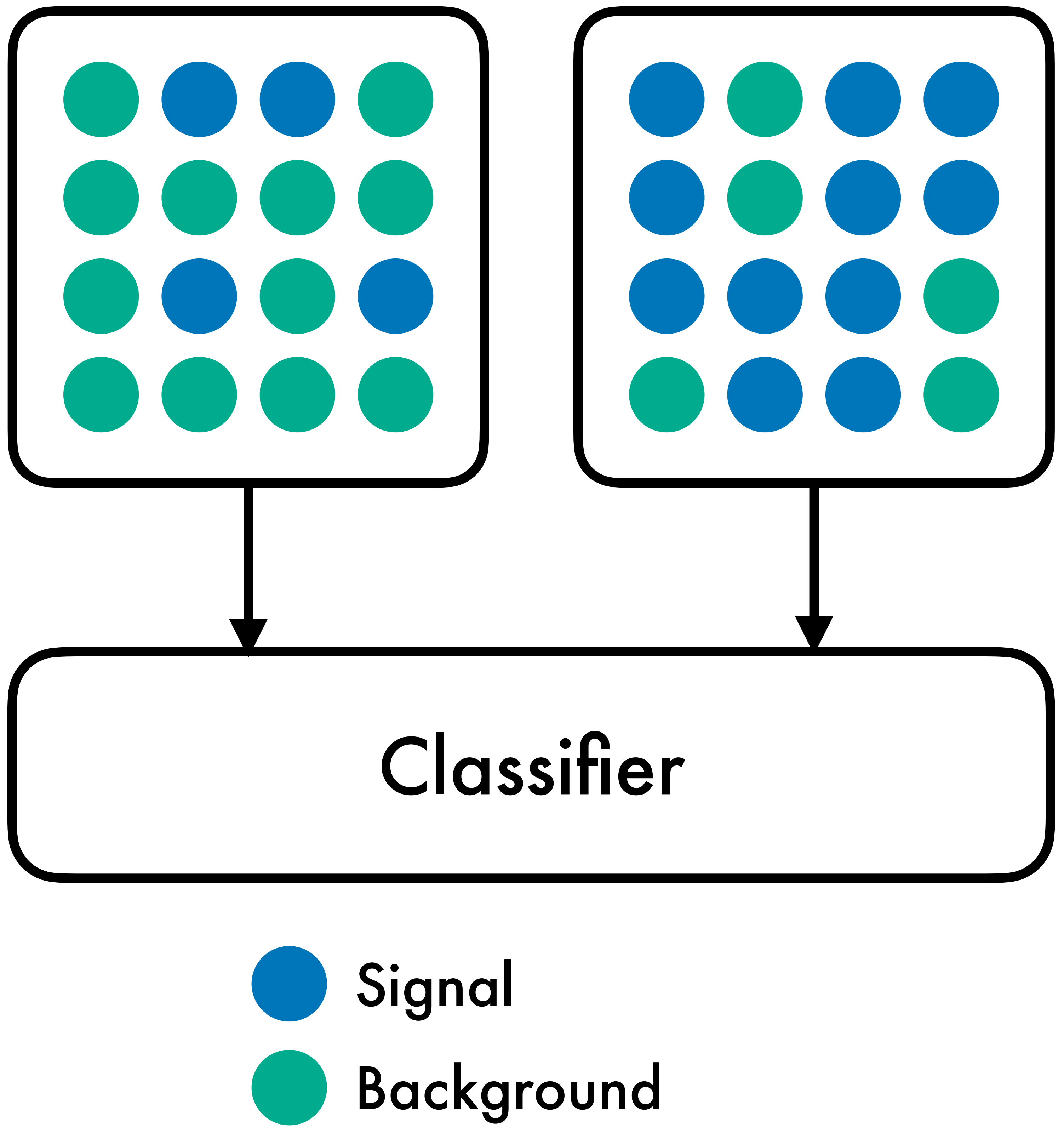
⊖ Labels **are not available** in experimental data



Classification without labels (CWoLa)

Two **mixed** **datasets** with signal fractions w_i

$$p_i(x) = w_i\, p_{\mathrm{sig}}(x) + (1 - w_i)\, p_{\mathrm{bg}}(x)$$



Metodiev, Nachman, Thaler [1708.02949]

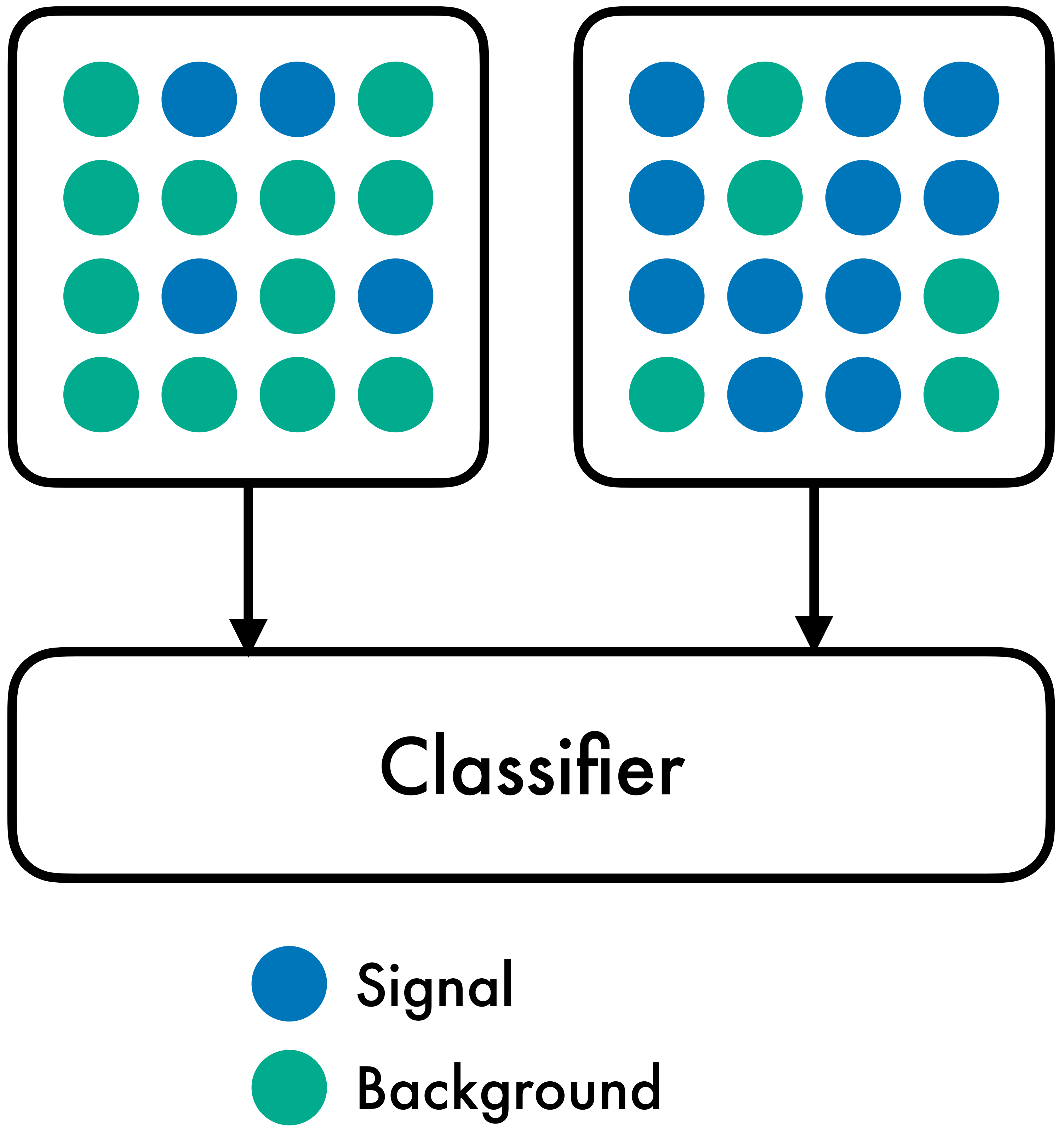
Classification without labels (CWoLa)

Two **mixed datasets** with signal fractions w_i

$$p_i(x) = w_i \, p_{\text{sig}}(x) + (1 - w_i) \, p_{\text{bg}}(x)$$

Classifier gives likelihood ratio

$$R_{\text{mixed}} = \frac{w_1 \, R_{\text{optimal}}(x) + (1 - w_1)}{w_2 \, R_{\text{optimal}}(x) + (1 - w_2)}$$



Metodiev, Nachman, Thaler [1708.02949]

Classification without labels (CWoLa)

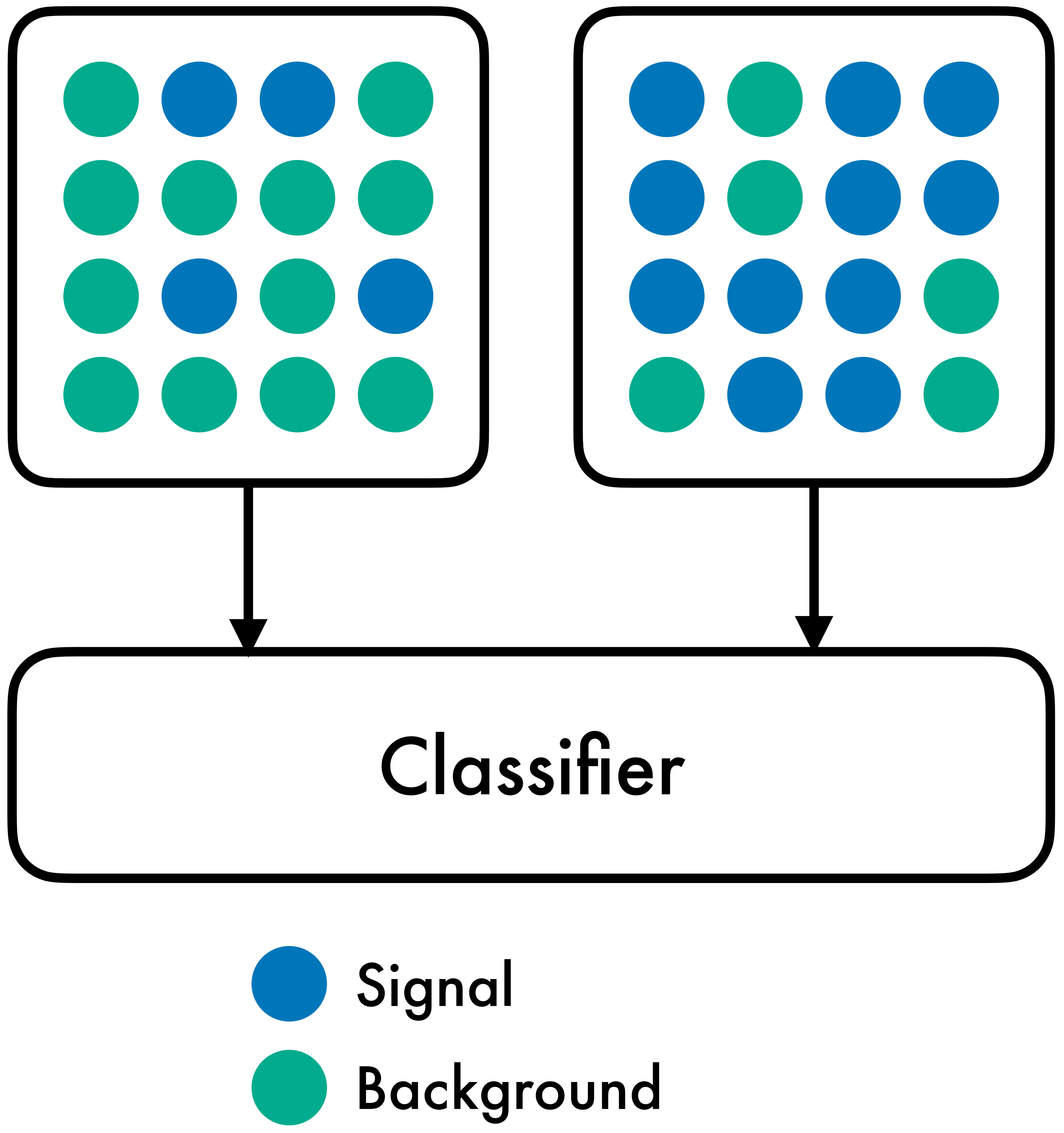
Two **mixed datasets** with signal fractions w_i

$$p_i(x) = w_i\, p_{\text{sig}}(x) + (1 - w_i)\, p_{\text{bg}}(x)$$

Classifier gives likelihood ratio

$$R_{\text{mixed}} = \frac{w_1\, R_{\text{optimal}}(x) + (1 - w_1)}{w_2\, R_{\text{optimal}}(x) + (1 - w_2)}$$

⊕ Monotonic function
→ optimal on mixed = optimal on pure sample



Metodiev, Nachman, Thaler [1708.02949]

Classification without labels (CWoLa)

Two **mixed datasets** with signal fractions w_i

$$p_i(x) = w_i\, p_{\text{sig}}(x) + (1 - w_i)\, p_{\text{bg}}(x)$$

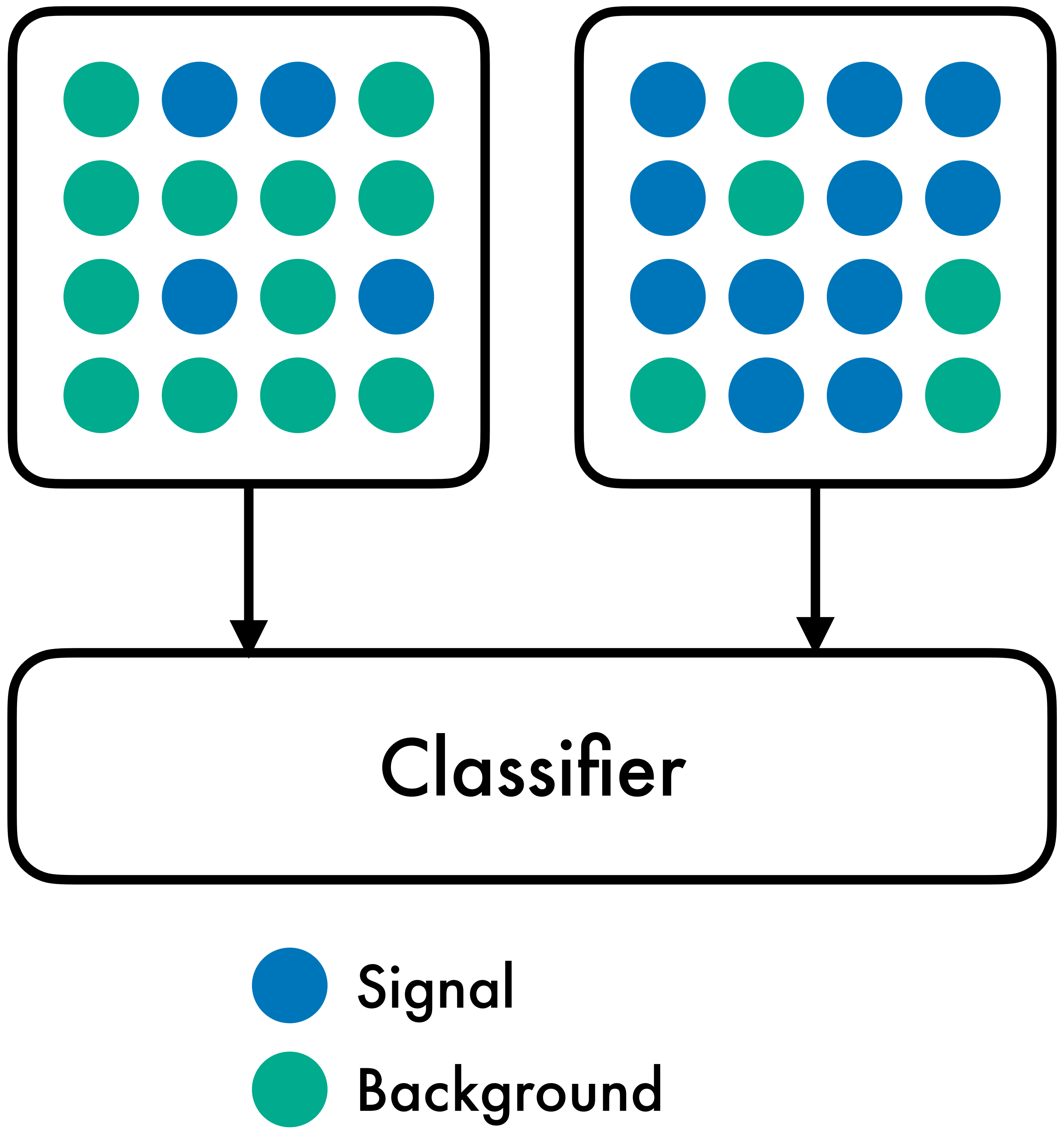
Classifier gives likelihood ratio

$$R_{\text{mixed}} = \frac{w_1\, R_{\text{optimal}}(x) + (1 - w_1)}{w_2\, R_{\text{optimal}}(x) + (1 - w_2)}$$

⊕ Monotonic function

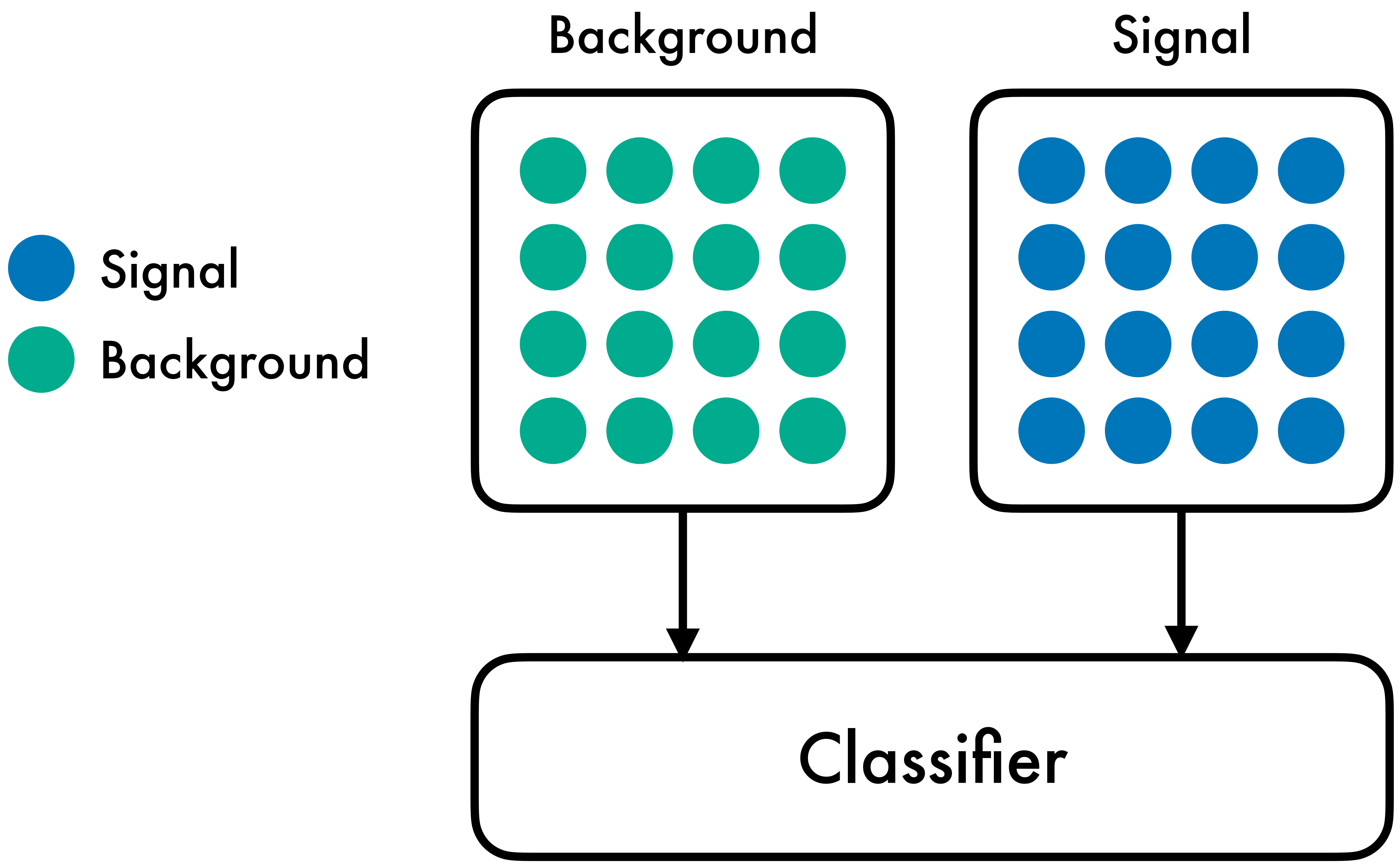
→ optimal on mixed = optimal on pure sample

→ Basis of **weak supervised classification**

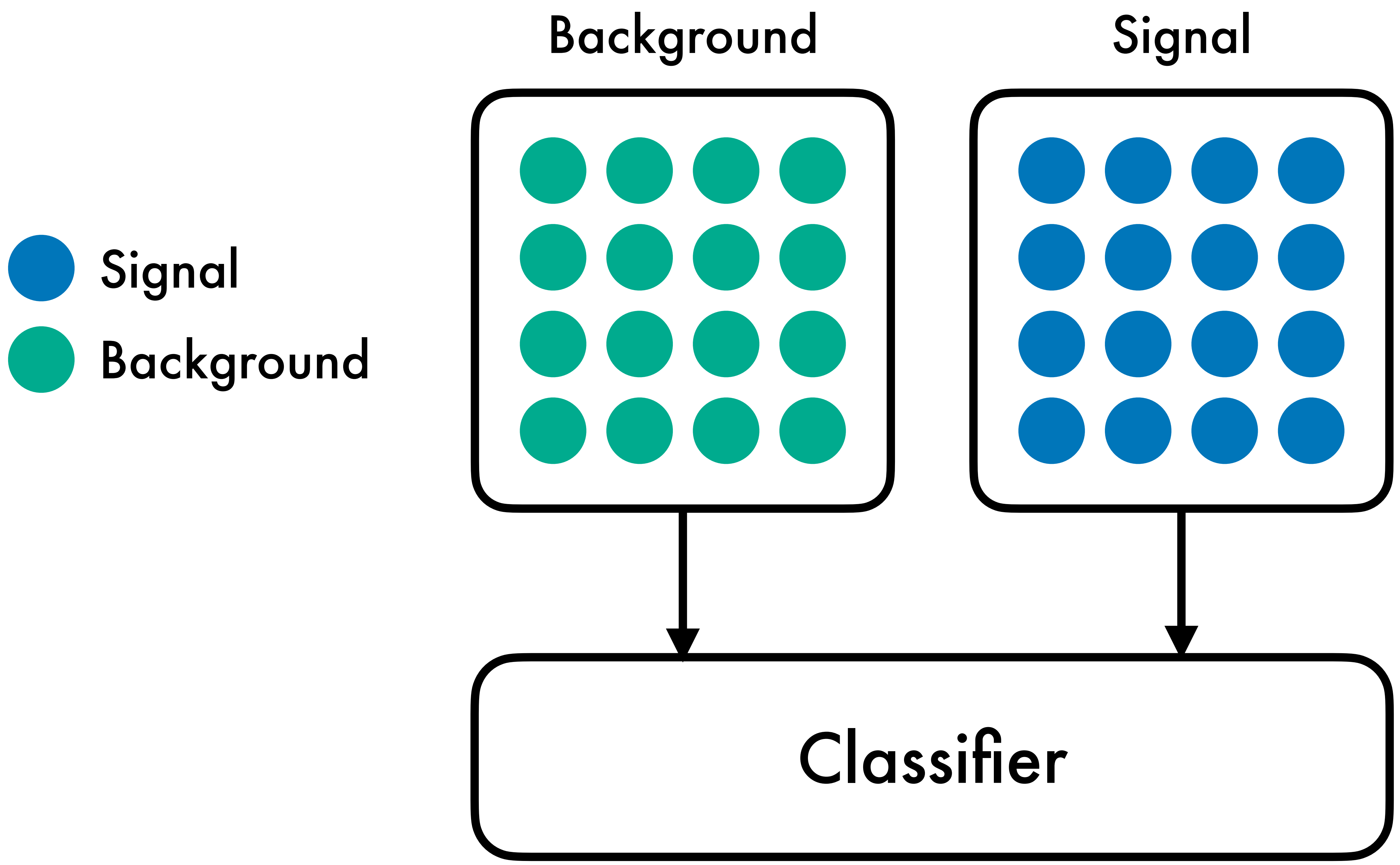


Metodiev, Nachman, Thaler [1708.02949]

Supervised versus IAD

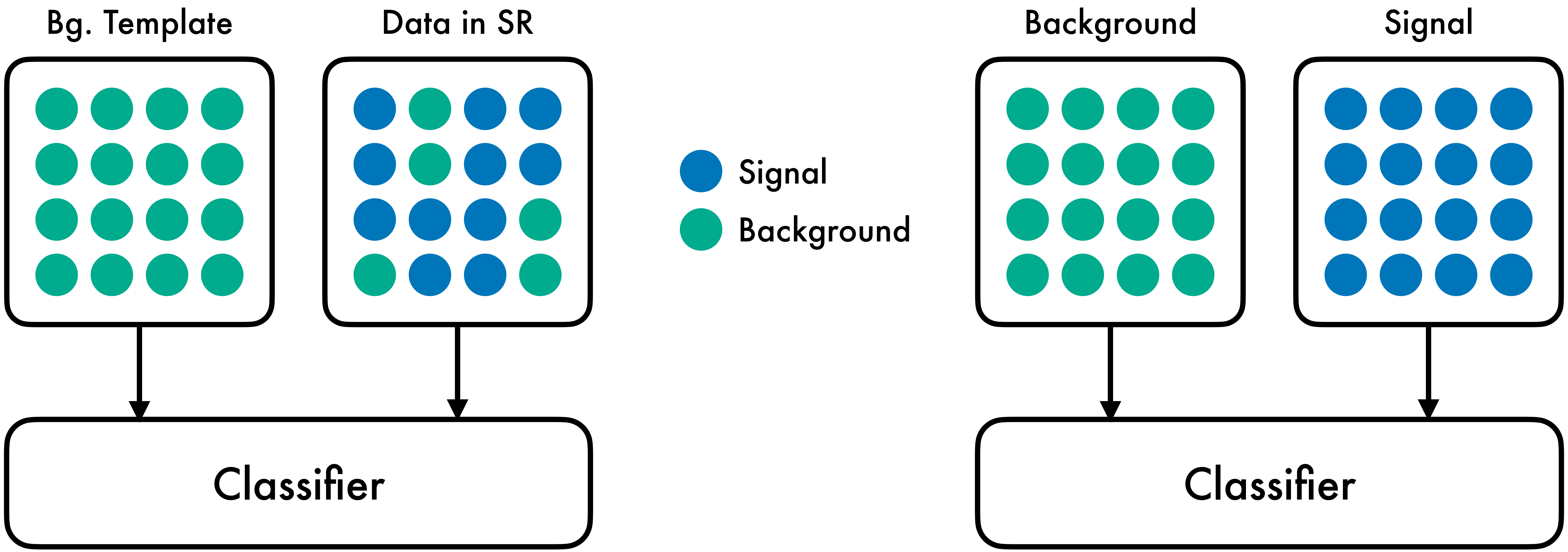


Supervised versus IAD



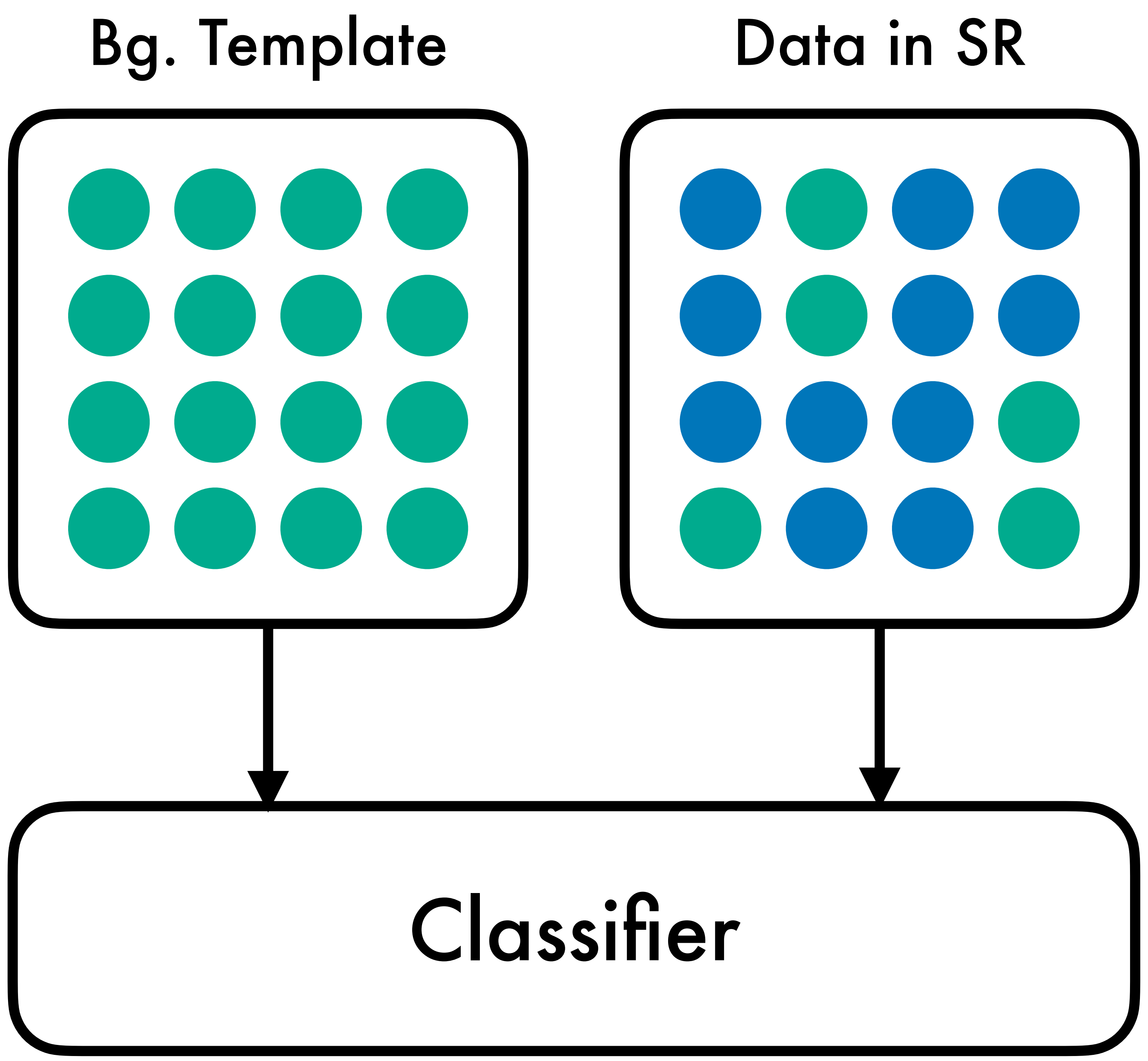
$$R_{\text{supervised}} = \frac{p_{\text{sig}}(x)}{p_{\text{bg}}(x)}$$

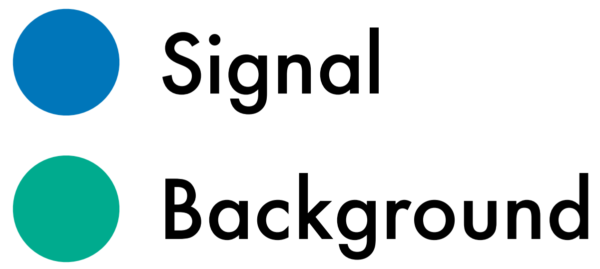
Supervised versus IAD

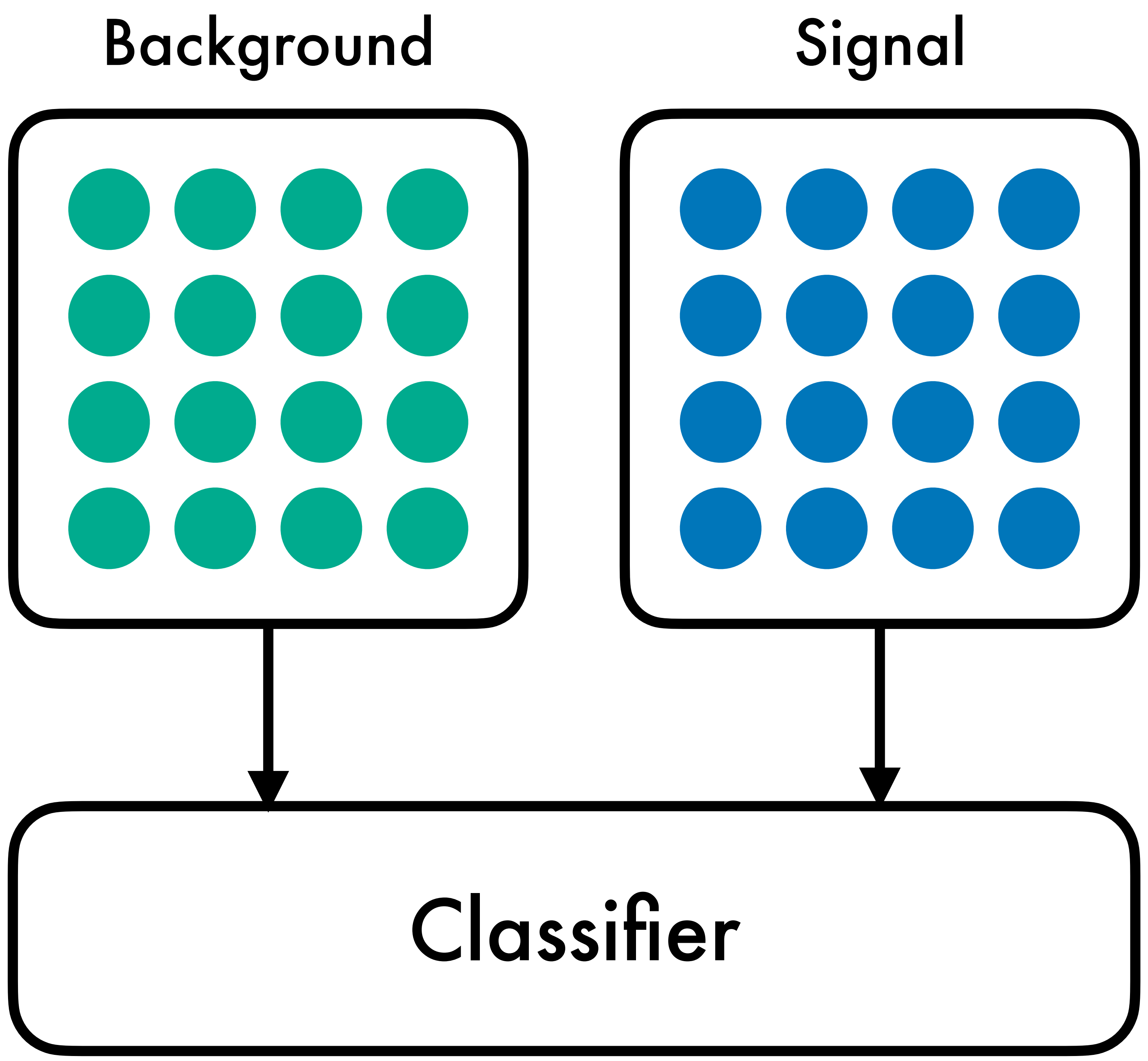


$$R_{\text{supervised}} = \frac{p_{\text{sig}}(x)}{p_{\text{bg}}(x)}$$

Supervised versus IAD



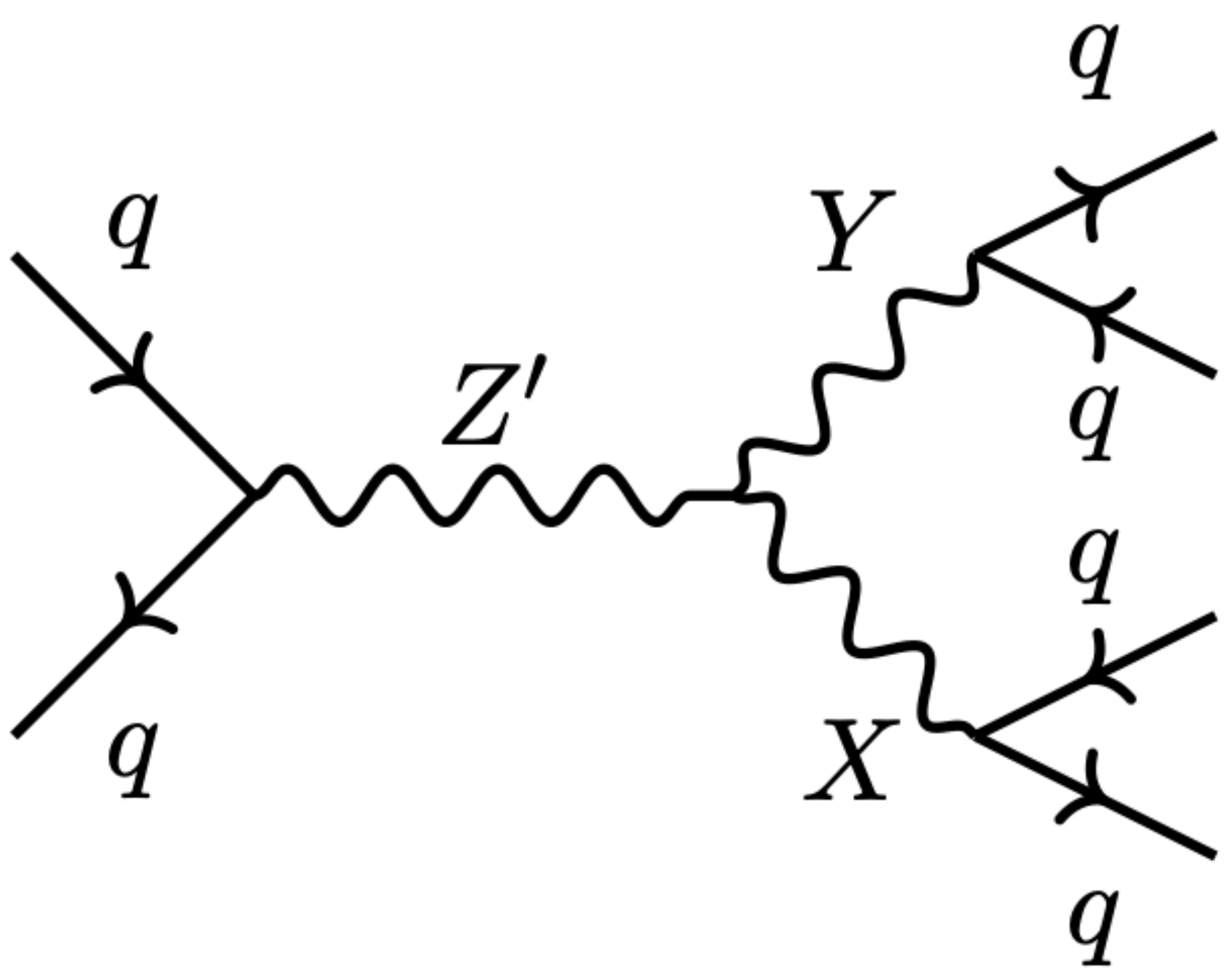




$$R_{\mathrm{IAD}} = \frac{p_{\mathrm{data}}(x)}{p_{\mathrm{bg}}(x)} = \epsilon\, R_{\mathrm{supervised}} + (1 - \epsilon)$$

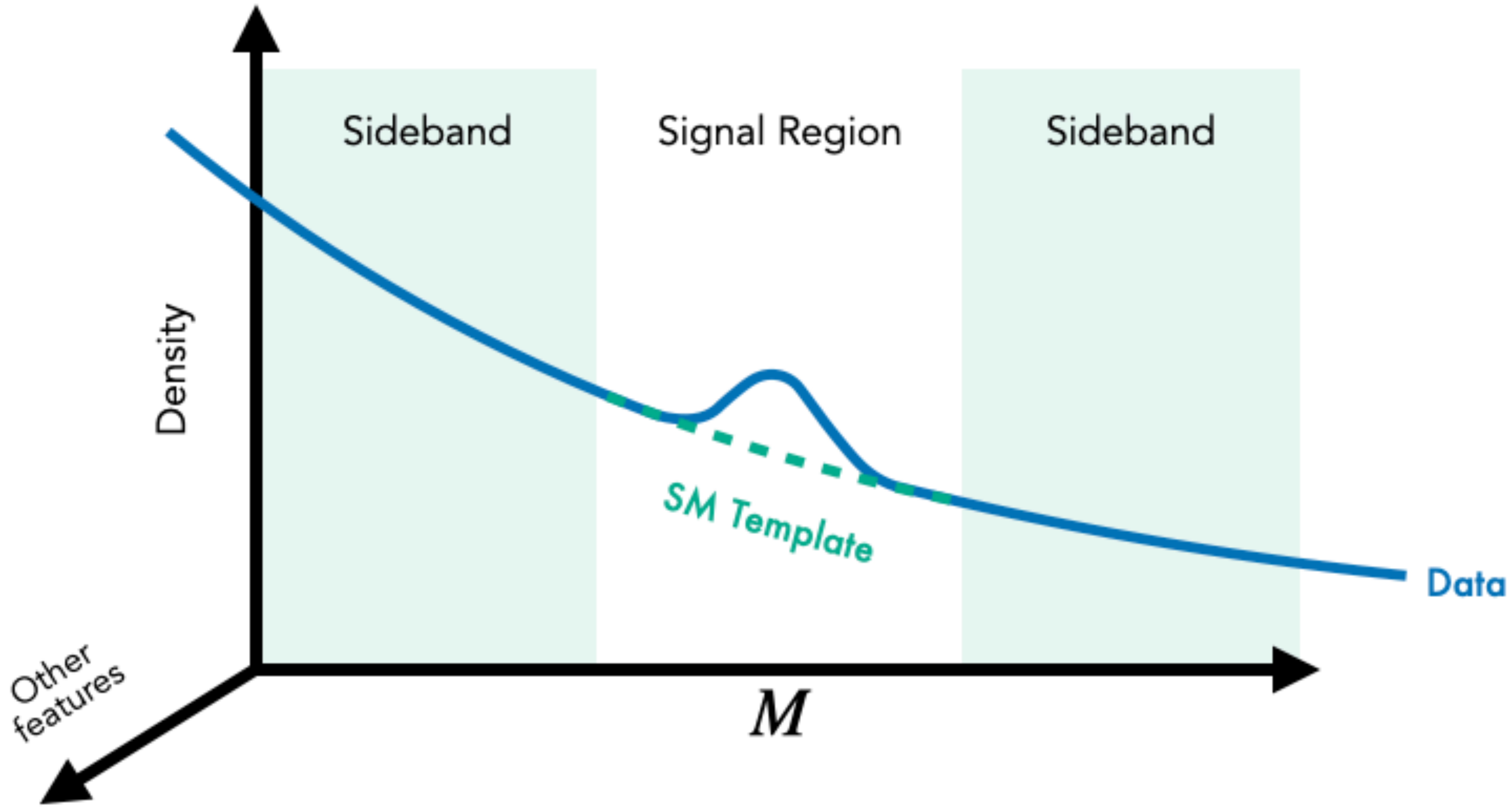
$$R_{\mathrm{supervised}} = \frac{p_{\mathrm{sig}}(x)}{p_{\mathrm{bg}}(x)}$$

CWoLa Hunting



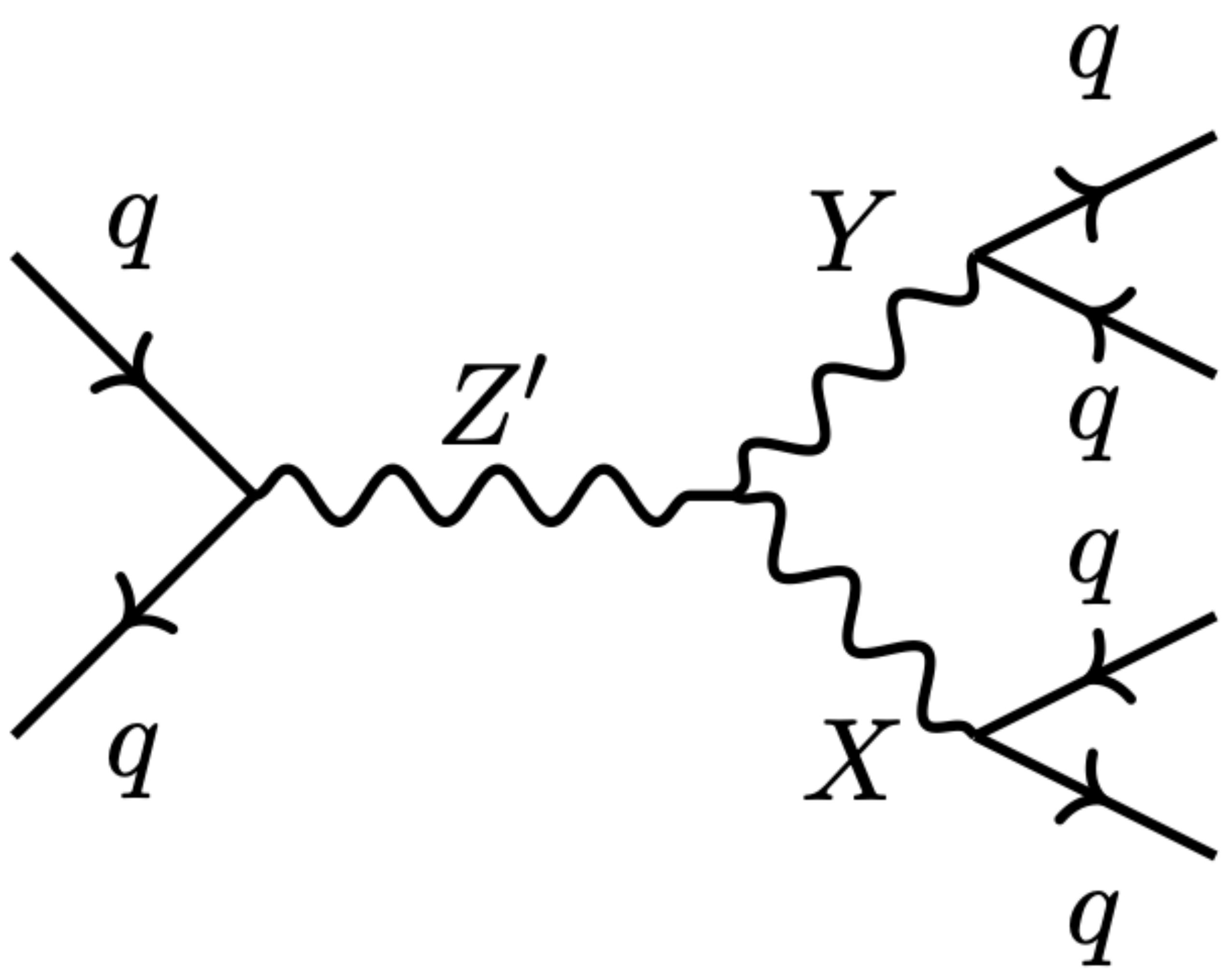
LHC Olympics

[Kasieczka et al: 2107.02821, 2101.08320]



[1902.02634]

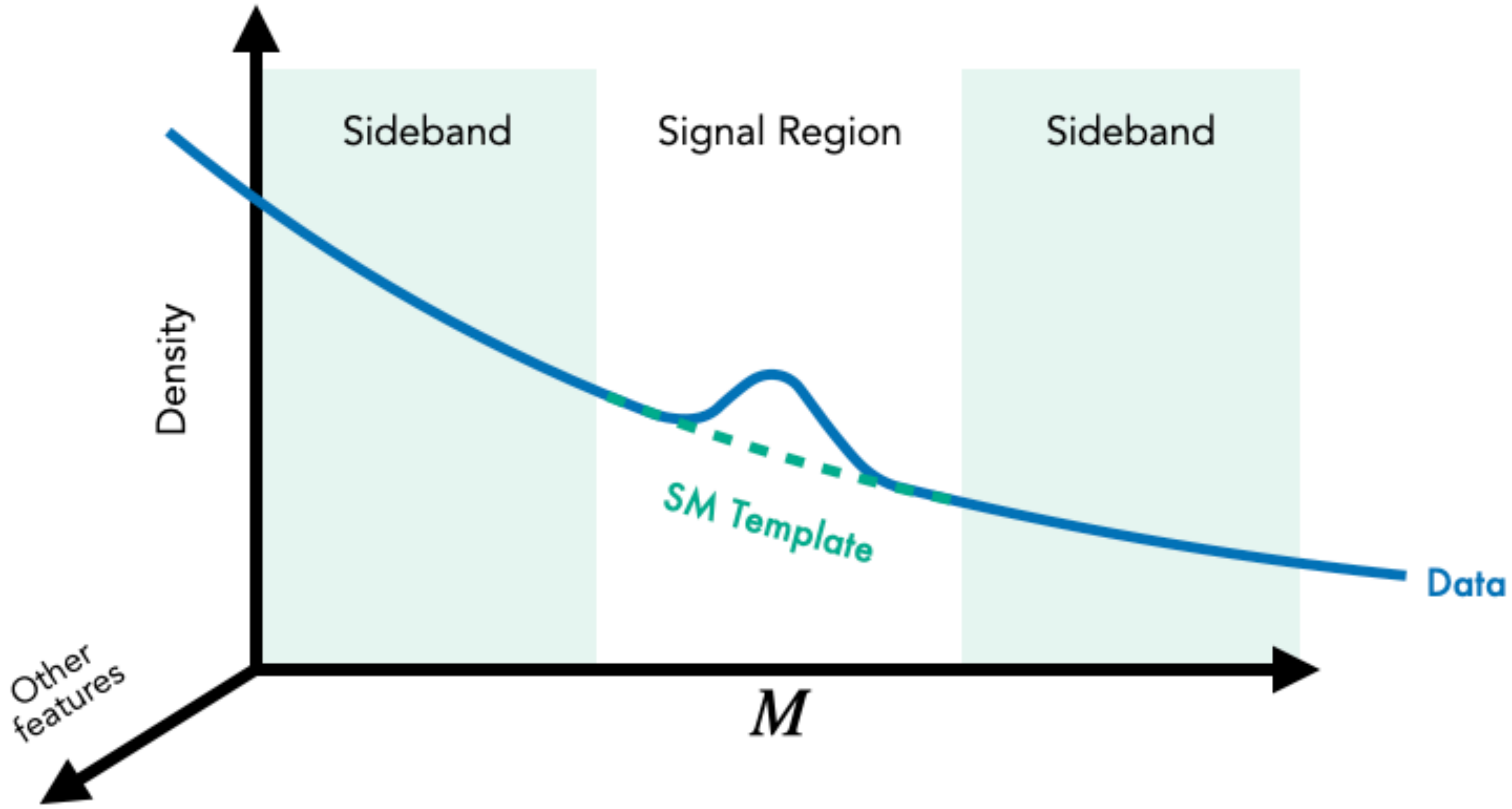
CWoLa Hunting



Resonant observable $m_{jj} = m_{Z'} > m_X, m_Y$

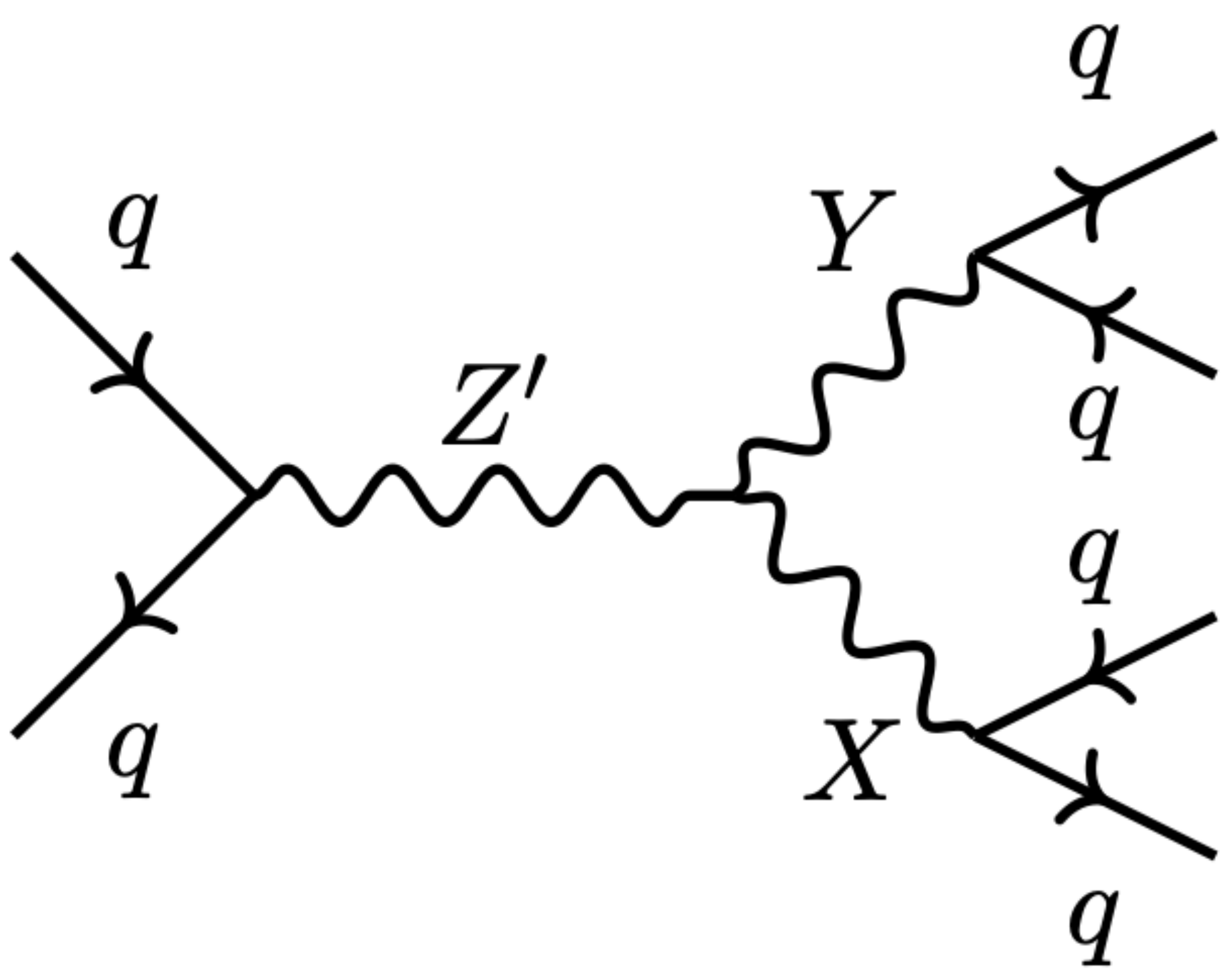
LHC Olympics

[Kasieczka et al: 2107.02821, 2101.08320]



[1902.02634]

CWoLa Hunting

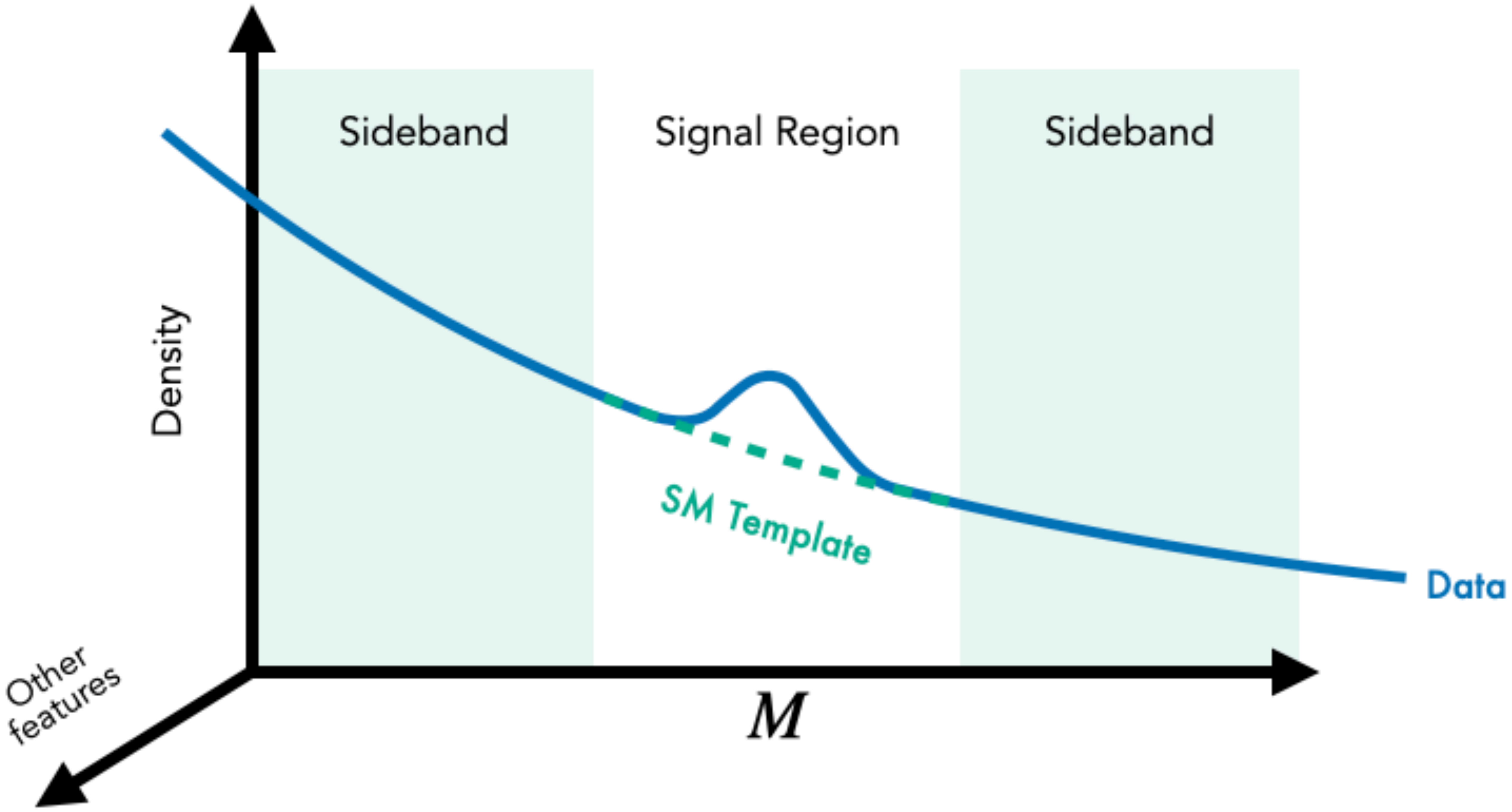


Resonant observable $m_{jj} = m_{Z'} > m_X, m_Y$

Other features $x = \{m_X, m_Y, \Delta m_j, \tau_{21}^{(1)}, \tau_{21}^{(2)}\}$

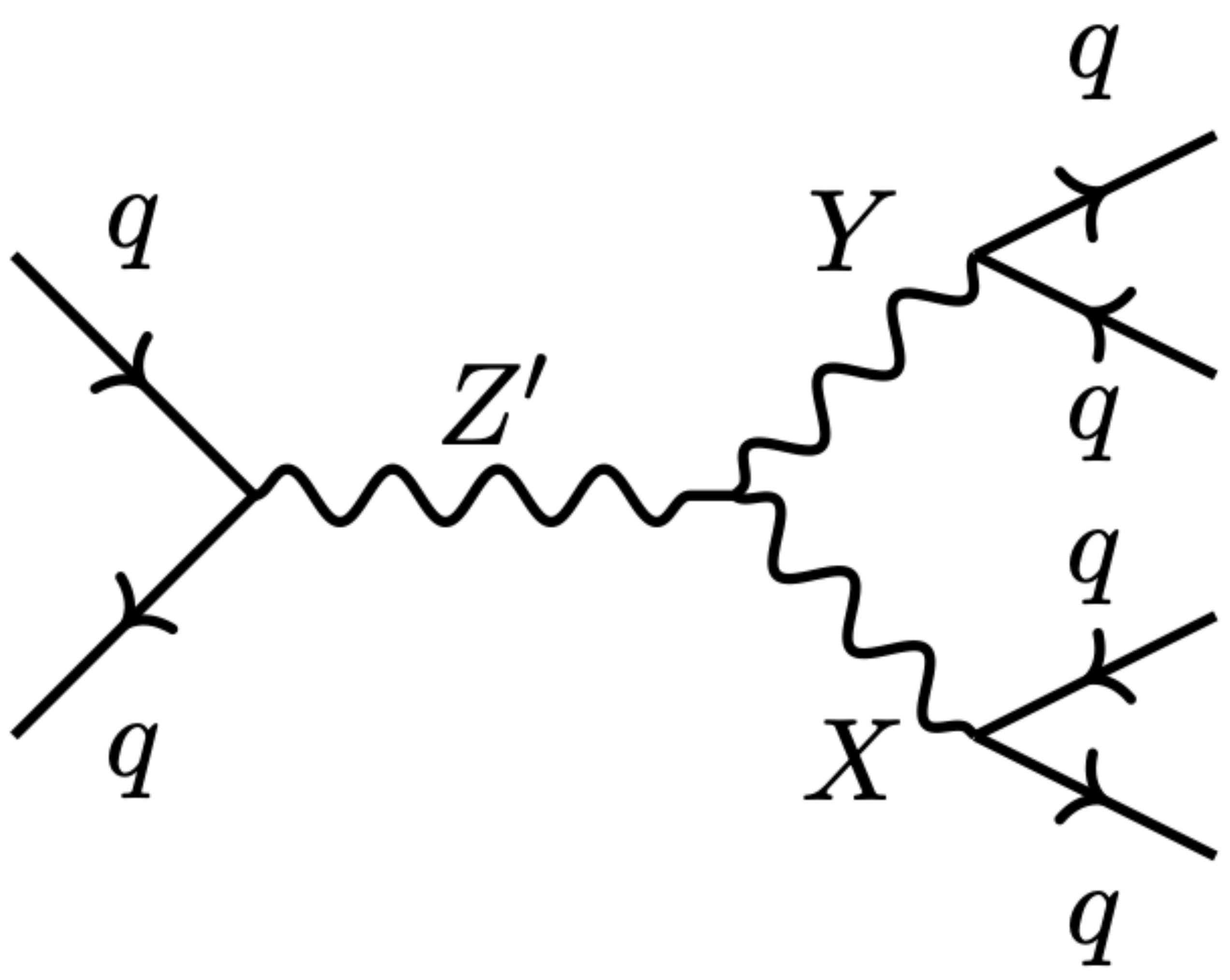
LHC Olympics

[Kasieczka et al: 2107.02821, 2101.08320]



[1902.02634]

CWoLa Hunting



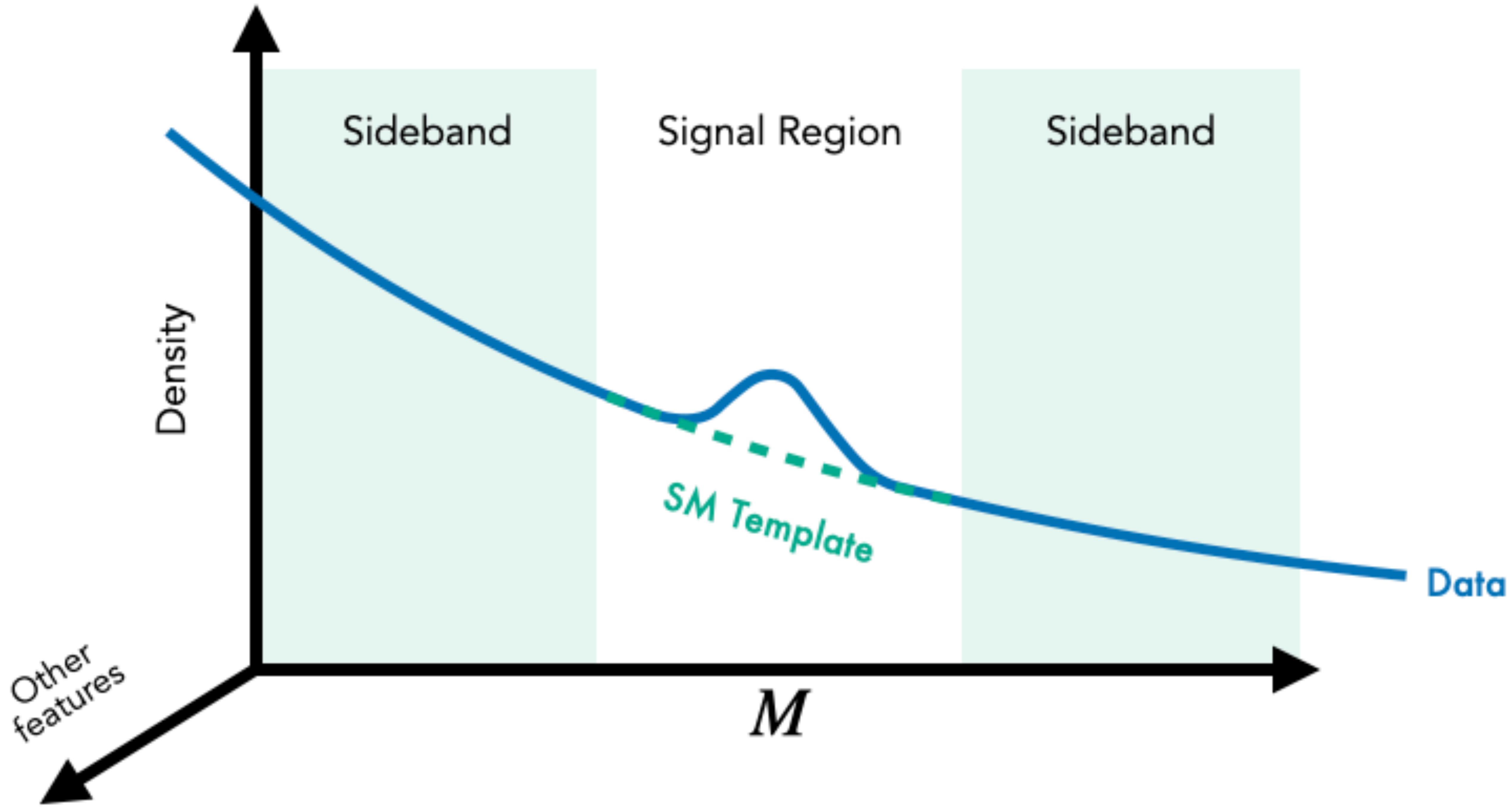
Resonant observable $m_{jj} = m_{Z'} > m_X, m_Y$

Other features $x = \{m_X, m_Y, \Delta m_j, \tau_{21}^{(1)}, \tau_{21}^{(2)}\}$

$$p_{\text{bg}}(x \,|\, m_{jj} \in \text{SR}) \approx p_{\text{bg}}(x \,|\, m_{jj} \in \text{SB}) \approx p_{\text{bg}}(x)$$

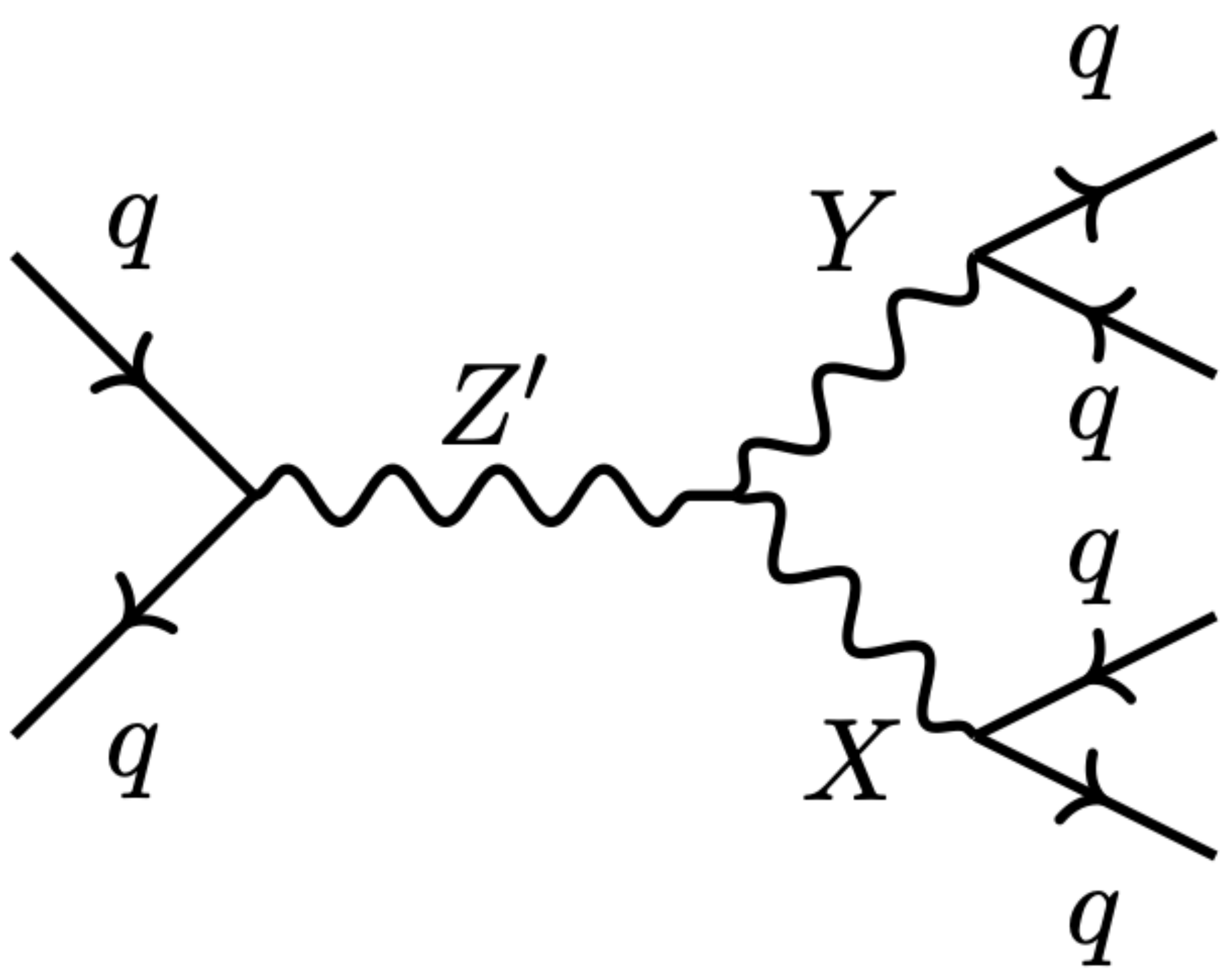
LHC Olympics

[Kasieczka et al: 2107.02821, 2101.08320]



[1902.02634]

CWoLa Hunting



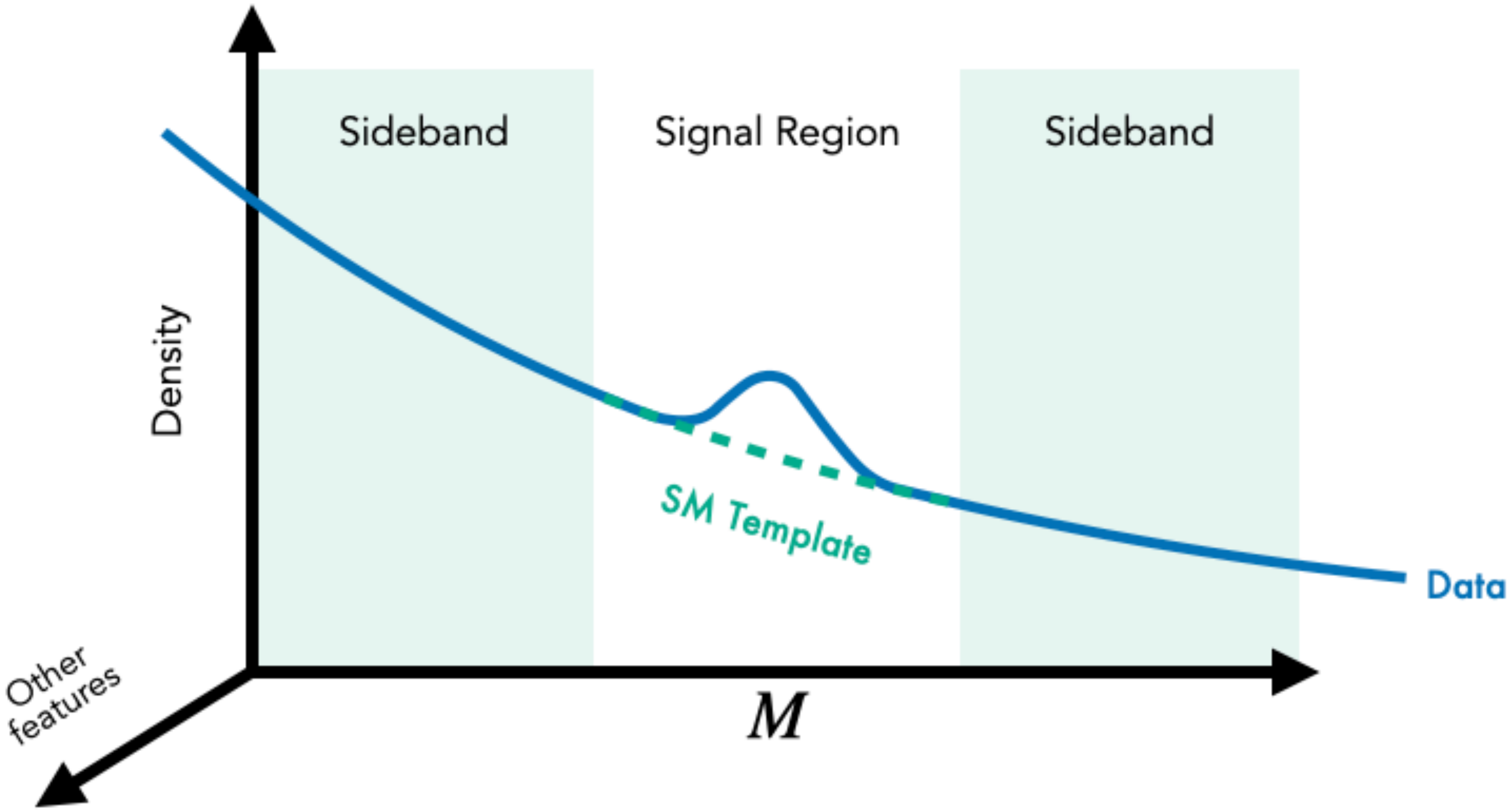
Resonant observable $m_{jj} = m_{Z'} > m_X, m_Y$

Other features $x = \{m_X, m_Y, \Delta m_j, \tau_{21}^{(1)}, \tau_{21}^{(2)}\}$

$$p_{\rm bg}(x \,|\, m_{jj} \in {\rm SR}) \approx p_{\rm bg}(x \,|\, m_{jj} \in {\rm SB}) \approx p_{\rm bg}(x)$$

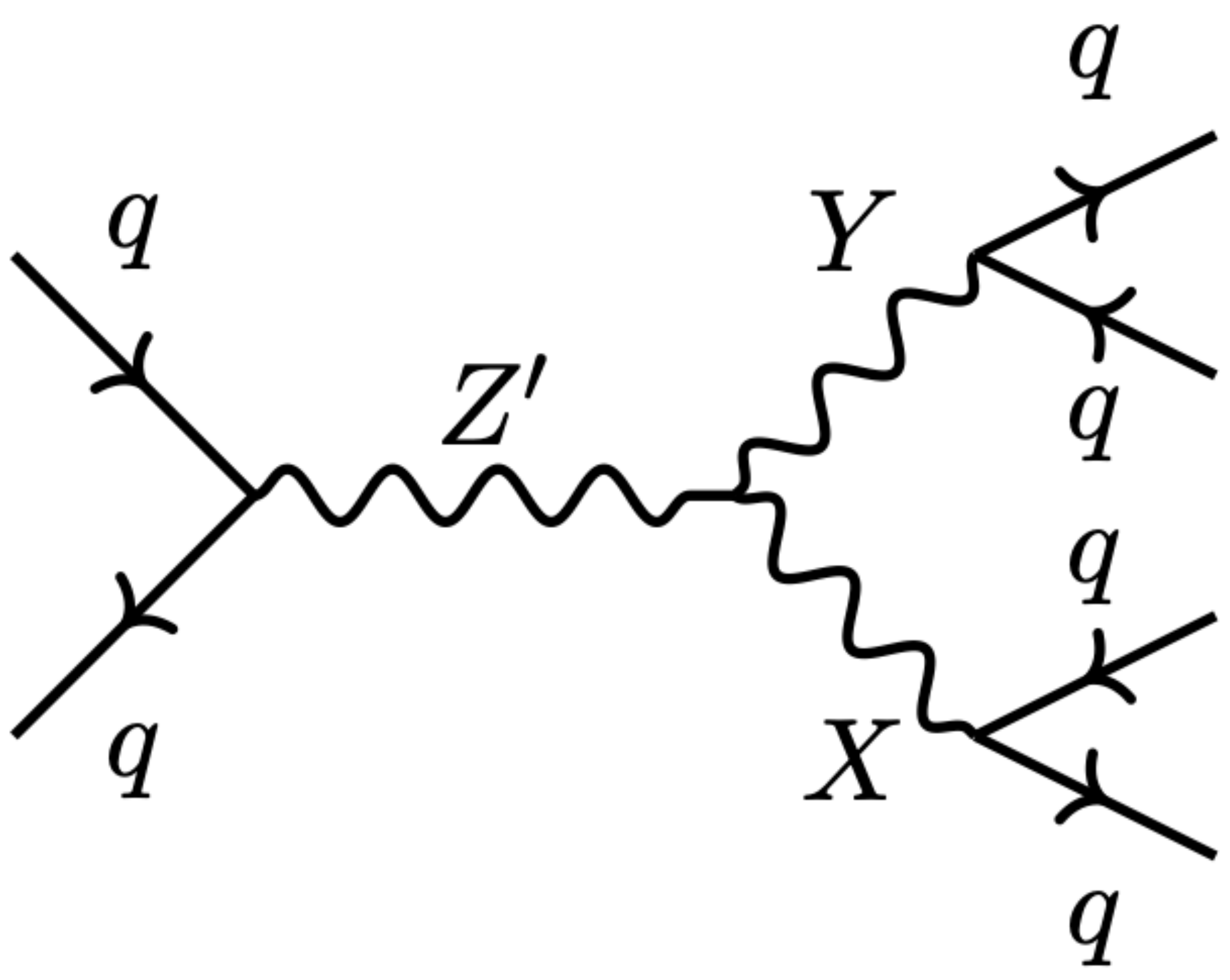
CWoLa Likelihood estimate

$$R_{\rm CWoLa} = \frac{p_{\rm data}(x \,|\, {\rm SR})}{p_{\rm bg}(x \,|\, {\rm SB})}$$



[1902.02634]

CWoLa Hunting



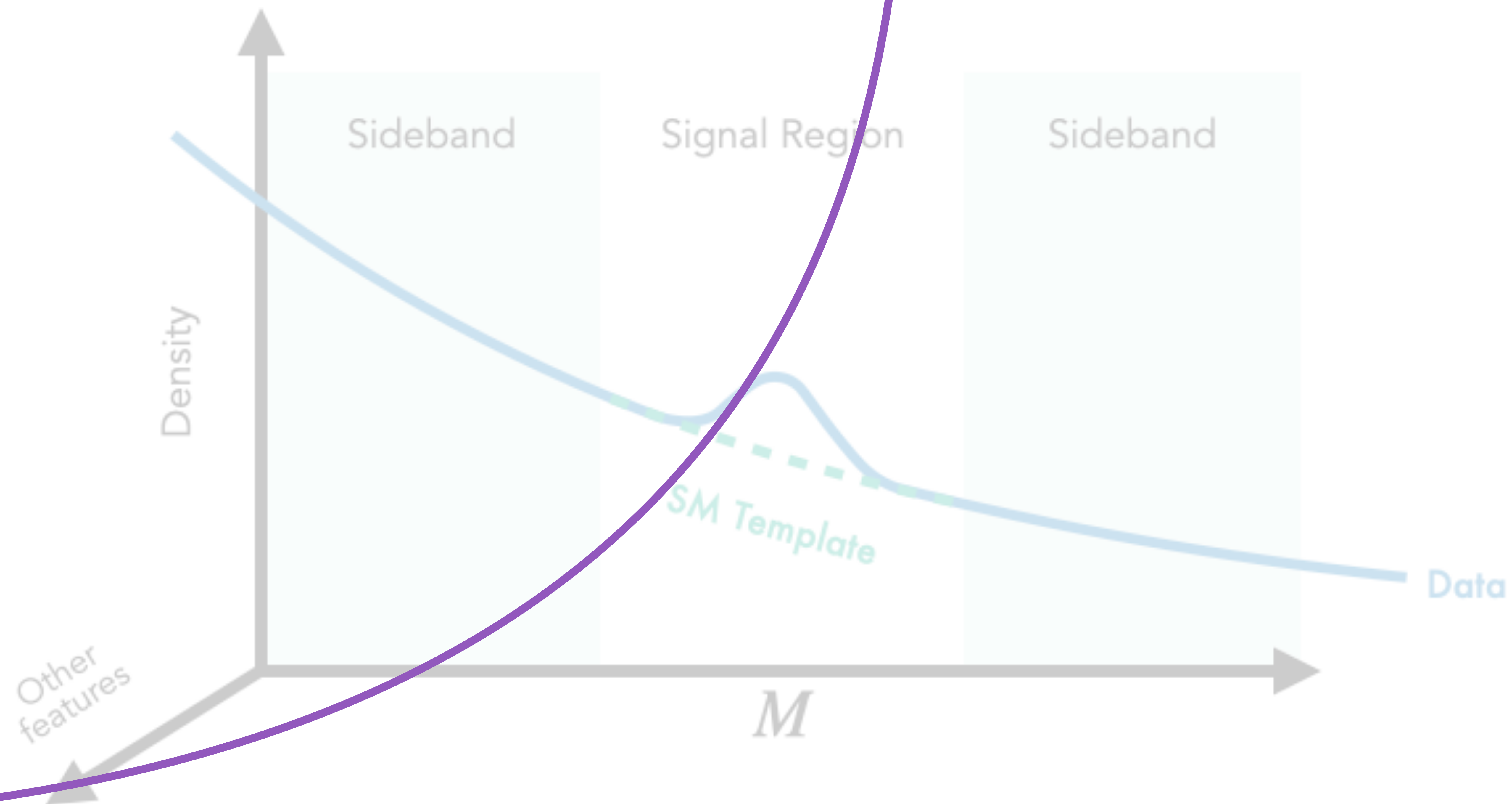
Resonant observable $m_{jj} = m_{Z'} > m_X, m_Y$

Other features $x = \{m_X, m_Y, \Delta m_j, \tau_{21}^{(1)}, \tau_{21}^{(2)}\}$

$$p_{\mathrm{bg}}(x \,|\, m_{jj} \in \mathrm{SR}) \approx p_{\mathrm{bg}}(x \,|\, m_{jj} \in \mathrm{SB}) \approx p_{\mathrm{bg}}(x)$$

CWoLa Likelihood estimate

$$R_{\mathrm{CWoLa}} = \frac{p_{\mathrm{data}}(x \,|\, \mathrm{SR})}{p_{\mathrm{bg}}(x \,|\, \mathrm{SB})} \approx \frac{p_{\mathrm{data}}(x \,|\, \mathrm{SR})}{p_{\mathrm{bg}}(x \,|\, \mathrm{SR})}$$



[1902.02634]

Can we do better?

Example II

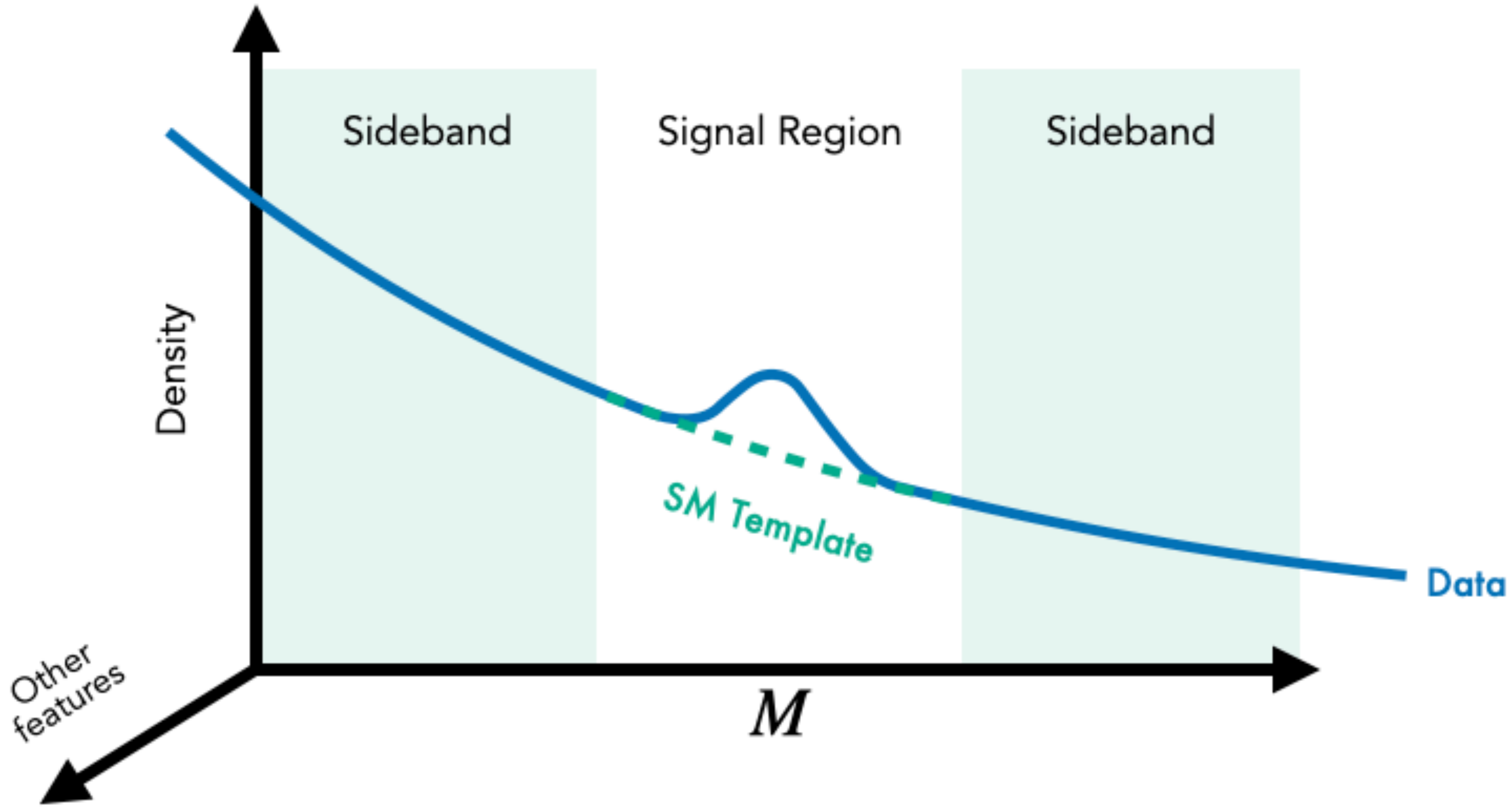
ANOmaly detection with Density Estimation (ANODE)

Nachman, Shih [2001.04990]

ANODE

CWoLa Likelihood estimate

$$R_{\text{CWoLa}} = \frac{p_{\text{data}}(x\,|\,\text{SR})}{p_{\text{bg}}(x\,|\,\text{SB})}$$



[2001.04990]

ANODE

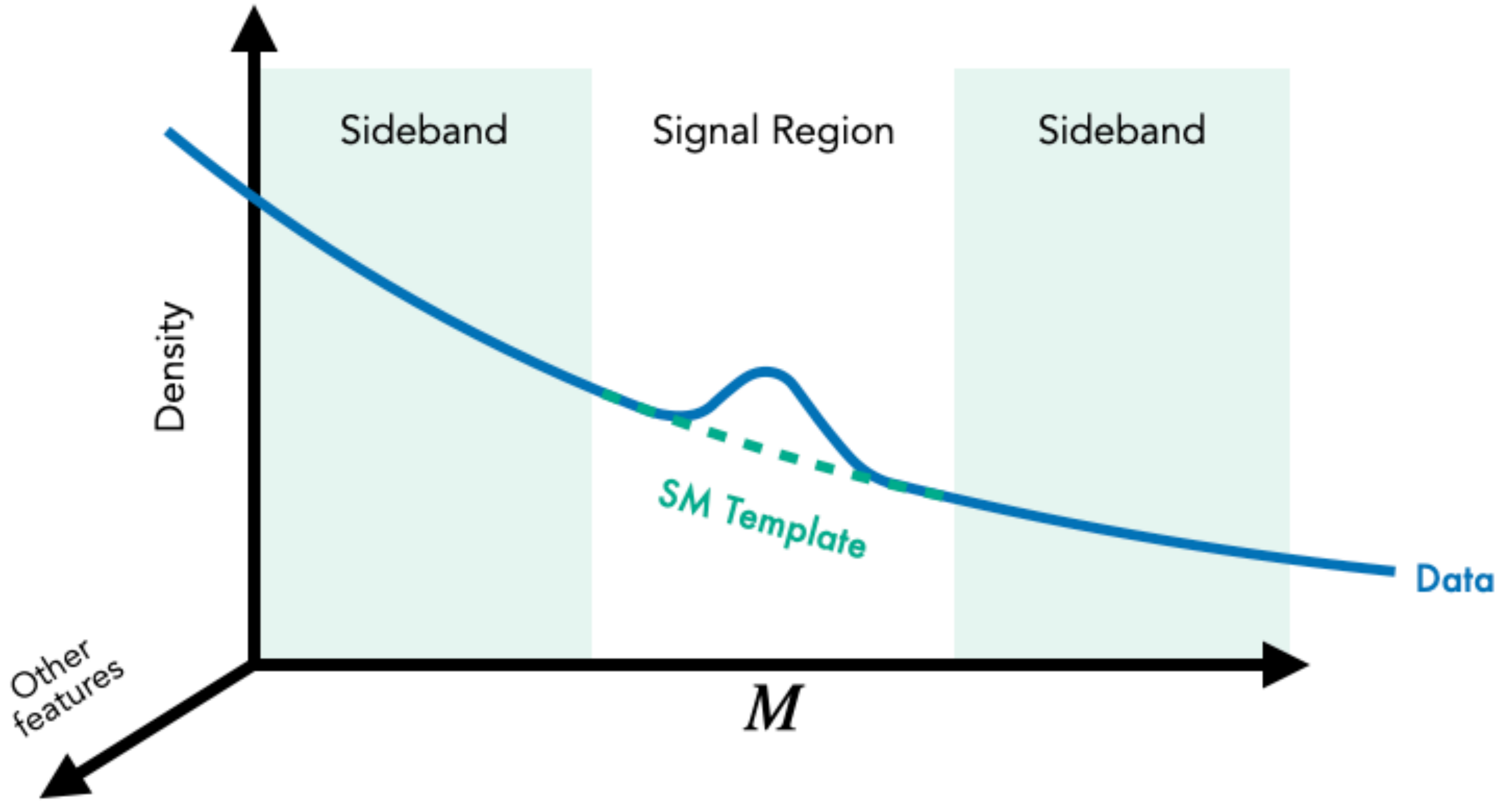
CWoLa Likelihood estimate

$$R_{\text{CWoLa}} = \frac{p_{\text{data}}(x \,|\, \text{SR})}{p_{\text{bg}}(x \,|\, \text{SB})}$$

The ANODE method

$$p_{\omega_0}(x \,|\, m) \simeq p_{\text{bg}}(x \,|\, m)$$

$$p_{\omega_1}(x \,|\, m) \simeq p_{\text{data}}(x \,|\, m)$$



[2001.04990]

ANODE

CWoLa Likelihood estimate

$$R_{\text{CWoLa}} = \frac{p_{\text{data}}(x\,|\,\text{SR})}{p_{\text{bg}}(x\,|\,\text{SB})}$$

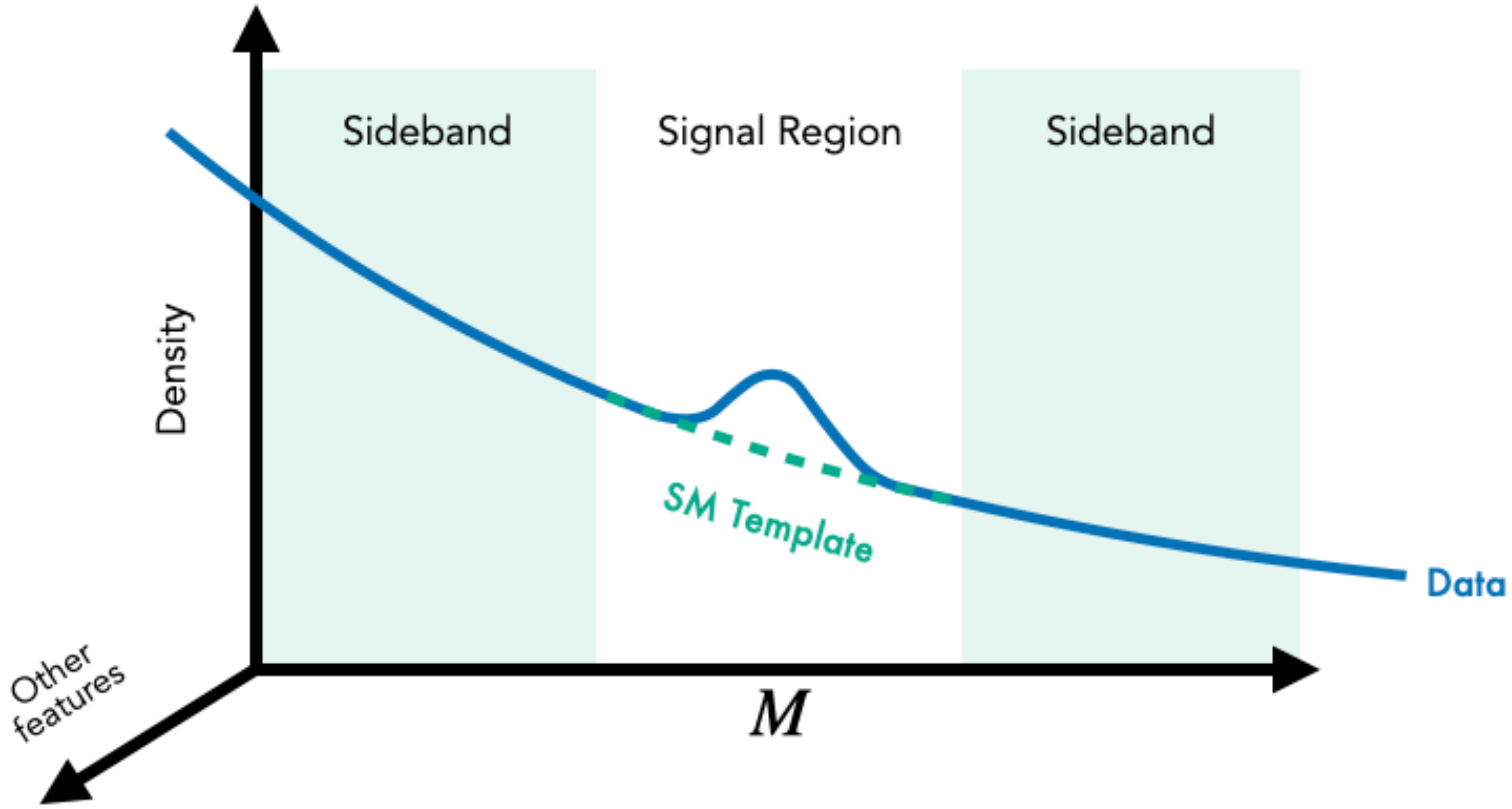
The ANODE method

NF

$$p_{\omega_0}(x\,|\,m) \simeq p_{\text{bg}}(x\,|\,m)$$

$$p_{\omega_1}(x\,|\,m) \simeq p_{\text{data}}(x\,|\,m)$$

NF



[2001.04990]

ANODE

CWoLa Likelihood estimate

$$R_{\text{CWoLa}} = \frac{p_{\text{data}}(x\,|\,\text{SR})}{p_{\text{bg}}(x\,|\,\text{SB})}$$

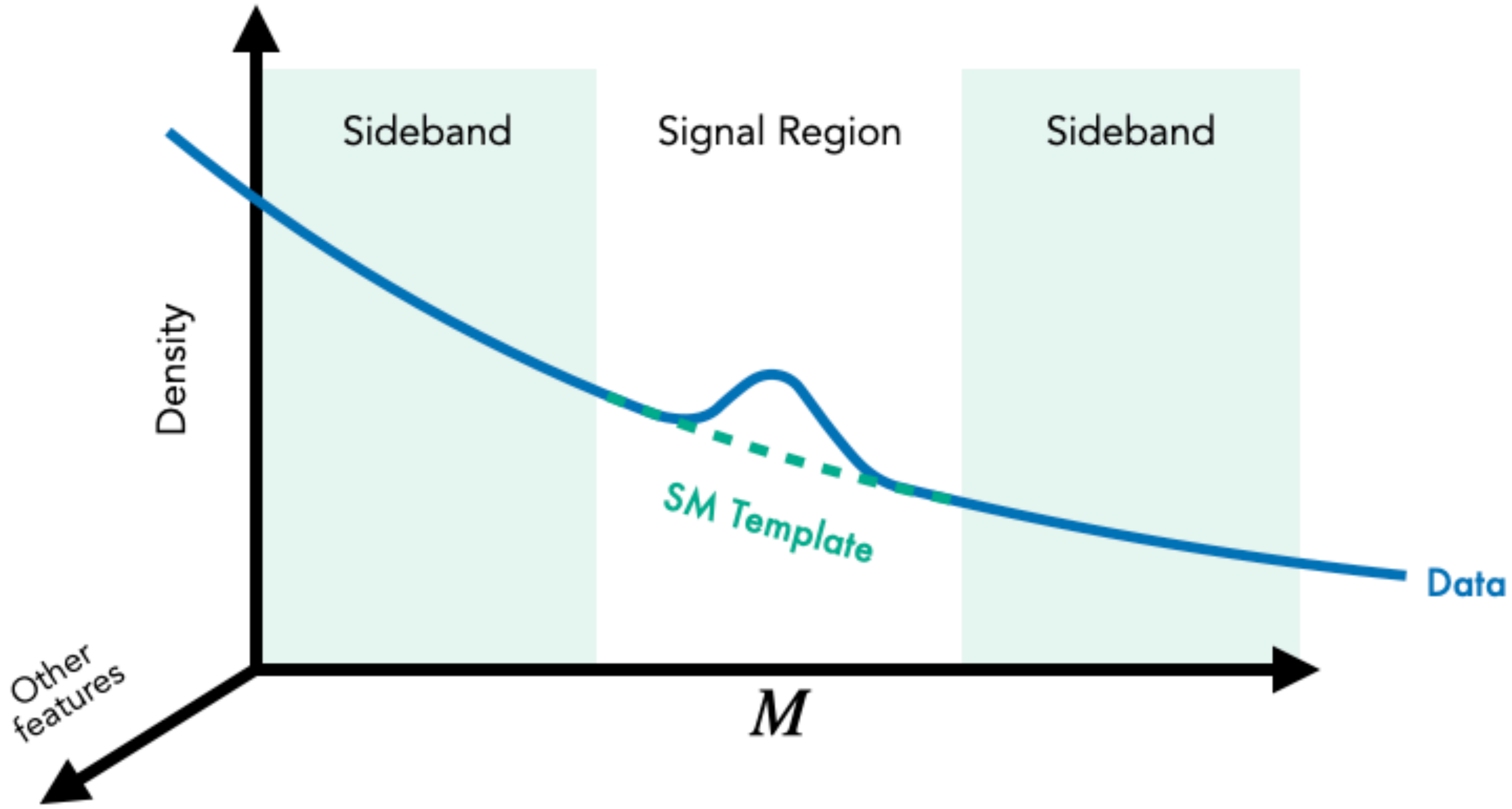
The ANODE method

NF

$p_{\omega_0}(x\,|\,m) \simeq p_{\text{bg}}(x\,|\,m)$ Trained in $m \in \text{SB}$

$p_{\omega_1}(x\,|\,m) \simeq p_{\text{data}}(x\,|\,m)$ Trained in $m \in \text{SR}$

NF



[2001.04990]

ANODE

CWoLa Likelihood estimate

$$R_{\text{CWoLa}} = \frac{p_{\text{data}}(x\,|\,\text{SR})}{p_{\text{bg}}(x\,|\,\text{SB})}$$

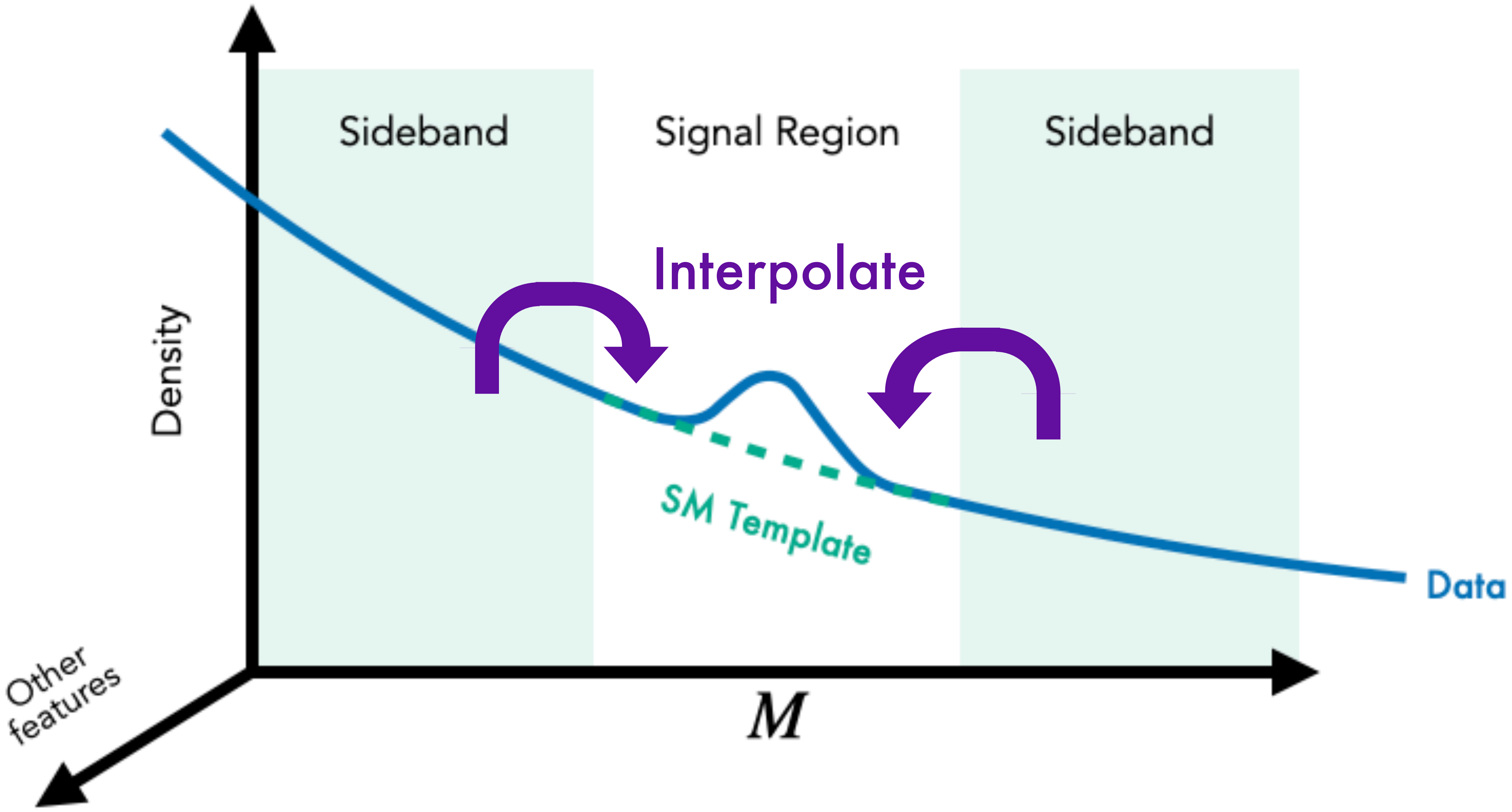
The ANODE method

NF

$p_{\omega_0}(x\,|\,m) \simeq p_{\text{bg}}(x\,|\,m)$ Trained in $m \in \text{SB}$

$p_{\omega_1}(x\,|\,m) \simeq p_{\text{data}}(x\,|\,m)$ Trained in $m \in \text{SR}$

NF



[2001.04990]

ANODE

CWoLa Likelihood estimate

$$R_{\text{CWoLa}} = \frac{p_{\text{data}}(x\,|\,\text{SR})}{p_{\text{bg}}(x\,|\,\text{SB})}$$

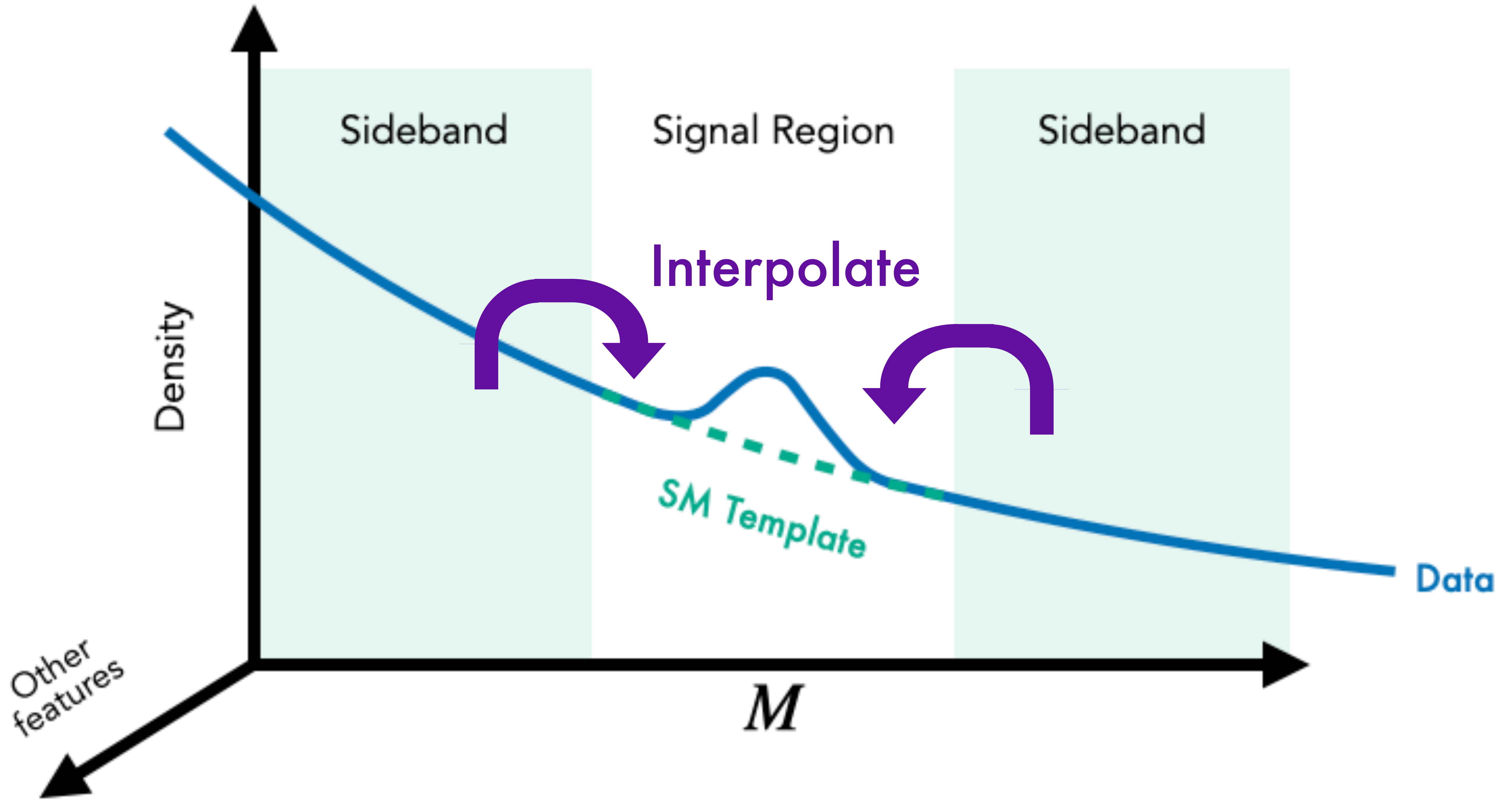
The ANODE method

NF

$p_{\omega_0}(x\,|\,m) \simeq p_{\text{bg}}(x\,|\,m)$ Trained in $m \in \text{SB}$

$p_{\omega_1}(x\,|\,m) \simeq p_{\text{data}}(x\,|\,m)$ Trained in $m \in \text{SR}$

NF



[2001.04990]

ANODE

CWoLa Likelihood estimate

$$R_{\text{CWoLa}} = \frac{p_{\text{data}}(x\,|\,\text{SR})}{p_{\text{bg}}(x\,|\,\text{SR})}$$

ANODE Likelihood estimate

$$R_{\text{ANODE}} = \frac{p_{\omega_1}(x\,|\,\text{SR})}{p_{\omega_0}(x\,|\,\text{SR})} \simeq \frac{p_{\text{data}}(x\,|\,\text{SR})}{p_{\text{bg}}(x\,|\,\text{SR})}$$

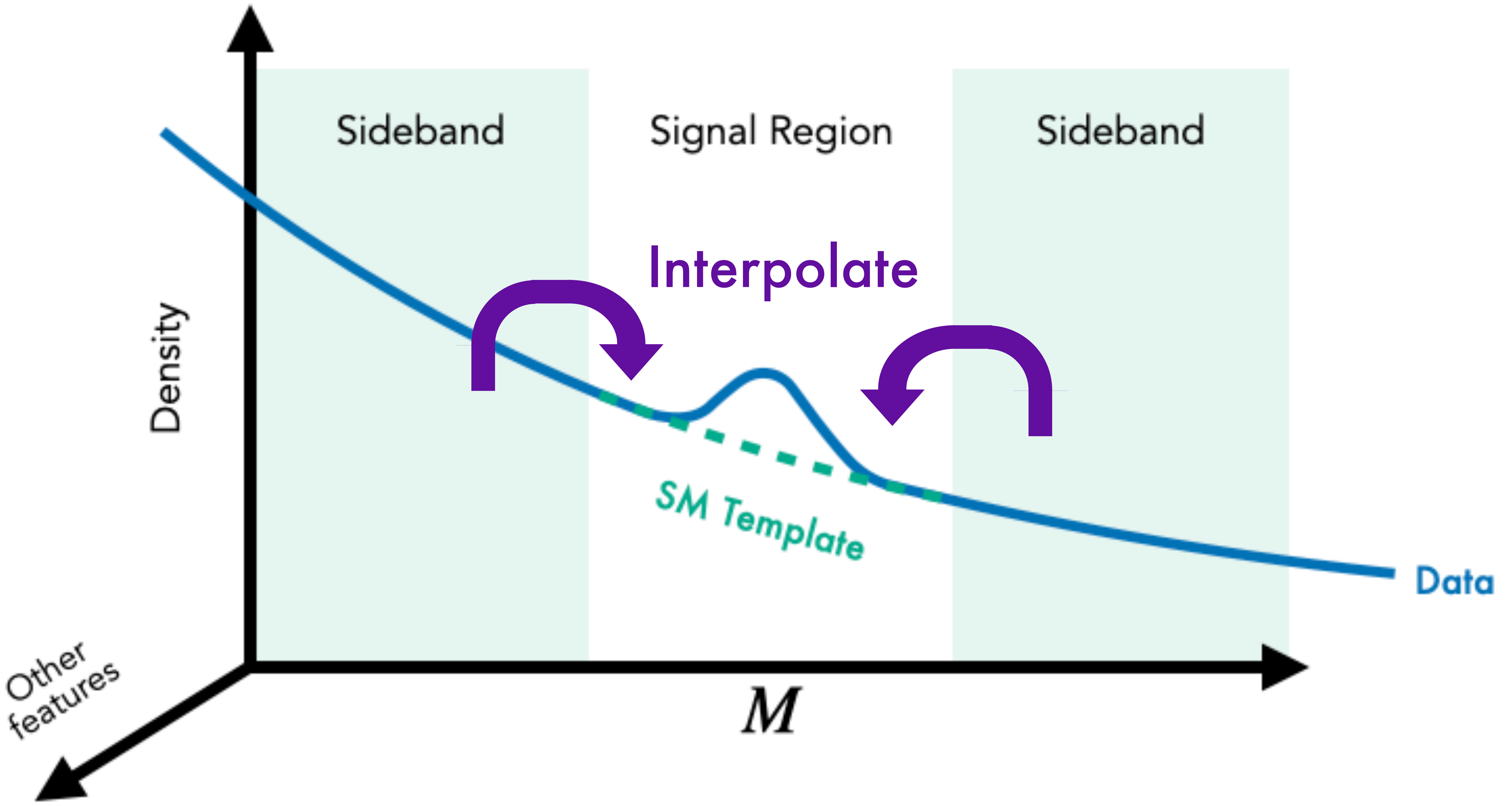
The ANODE method

NF

$p_{\omega_0}(x\,|\,m) \simeq p_{\text{bg}}(x\,|\,m)$ Trained in $m \in \text{SB}$

$p_{\omega_1}(x\,|\,m) \simeq p_{\text{data}}(x\,|\,m)$ Trained in $m \in \text{SR}$

NF



[2001.04990]

Are we already happy?

CWoLA versus ANODE

CWoLa Likelihood estimate

$$R_{\text{CWoLa}} = \frac{p_{\text{data}}(x \,|\, \text{SR})}{p_{\text{bg}}(x \,|\, \text{SB})}$$

Pros and cons:

[1902.02634]

ANODE Likelihood estimate

$$R_{\text{ANODE}} = \frac{p_{\omega_1}(x \,|\, \text{SR})}{p_{\omega_0}(x \,|\, \text{SR})}$$

Pros and cons:

[2001.04990]

CWoLA versus ANODE

CWoLa Likelihood estimate

$$R_{\mathrm{CWoLa}} = \frac{p_{\mathrm{data}}(x\,|\,\mathrm{SR})}{p_{\mathrm{bg}}(x\,|\,\mathrm{SB})}$$

Pros and cons:

⊕ Classification is easy and precise

[1902.02634]

ANODE Likelihood estimate

$$R_{\mathrm{ANODE}} = \frac{p_{\omega_1}(x\,|\,\mathrm{SR})}{p_{\omega_0}(x\,|\,\mathrm{SR})}$$

Pros and cons:

[2001.04990]

CWoLA versus ANODE

CWoLa Likelihood estimate

$$R_{\mathrm{CWoLa}} = \frac{p_{\mathrm{data}}(x\,|\,\mathrm{SR})}{p_{\mathrm{bg}}(x\,|\,\mathrm{SB})}$$

Pros and cons:

⊕ Classification is easy and precise

⊖ Sensitive to correlations between m_{jj} and other features x

[1902.02634]

ANODE Likelihood estimate

$$R_{\mathrm{ANODE}} = \frac{p_{\omega_1}(x\,|\,\mathrm{SR})}{p_{\omega_0}(x\,|\,\mathrm{SR})}$$

Pros and cons:

[2001.04990]

CWoLA versus ANODE

CWoLa Likelihood estimate

$$R_{\text{CWoLa}} = \frac{p_{\text{data}}(x \,|\, \text{SR})}{p_{\text{bg}}(x \,|\, \text{SB})}$$

Pros and cons:

⊕ Classification is easy and precise

⊖ Sensitive to correlations between m_{jj} and other features x

[1902.02634]

ANODE Likelihood estimate

$$R_{\text{ANODE}} = \frac{p_{\omega_1}(x \,|\, \text{SR})}{p_{\omega_0}(x \,|\, \text{SR})}$$

Pros and cons:

⊕ Robust against correlations

[2001.04990]

CWoLA versus ANODE

CWoLa Likelihood estimate

$$R_{\mathrm{CWoLa}} = \frac{p_{\mathrm{data}}(x\,|\,\mathrm{SR})}{p_{\mathrm{bg}}(x\,|\,\mathrm{SB})}$$

Pros and cons:

- ⊕ Classification is easy and precise
- ⊖ Sensitive to correlations between m_{jj} and other features x

[1902.02634]

ANODE Likelihood estimate

$$R_{\mathrm{ANODE}} = \frac{p_{\omega_1}(x\,|\,\mathrm{SR})}{p_{\omega_0}(x\,|\,\mathrm{SR})}$$

Pros and cons:

- ⊕ Robust against correlations
- ⊖ Less powerful and sensitive than classification

[2001.04990]

Can we get the best of both worlds?

Example III

Classifying Anomalies THrough Outer Density Estimation (CATHODE)

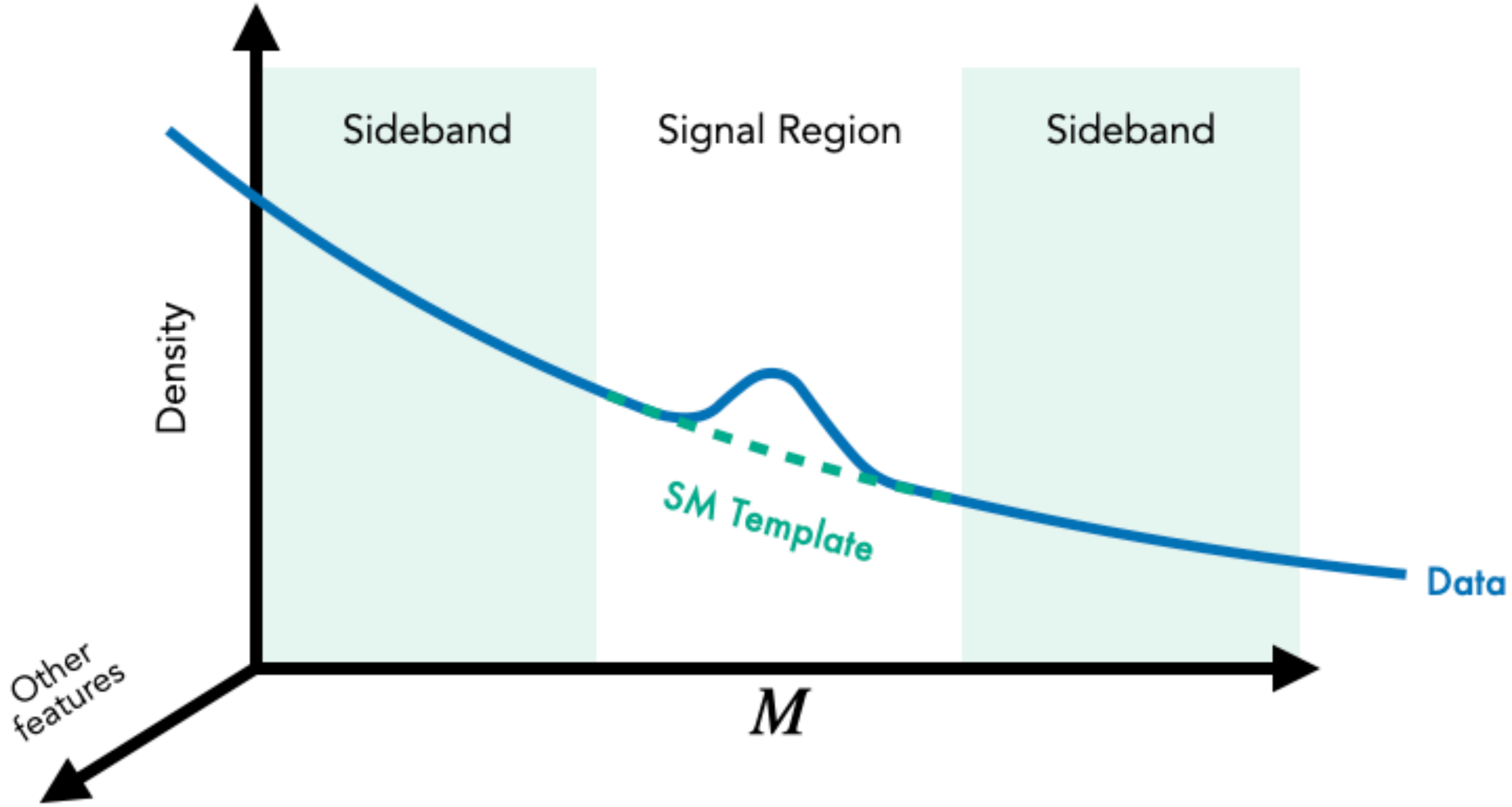
Hallin, Isaacson, Kasieczka, Krause, Nachman, Quadfasel, Schlaffer, Shih, Sommerhalder [2109.00546]

Best of both worlds — CATHODE

The CATHODE method

$p_{\omega_0}(x\,|\,m) \simeq p_{\mathrm{bg}}(x\,|\,m)$ Trained in $m \in \mathrm{SB}$

~~$p_{\omega_1}(x\,|\,m) \simeq p_{\mathrm{data}}(x\,|\,m)$~~



[2109.00546]

Best of both worlds — CATHODE

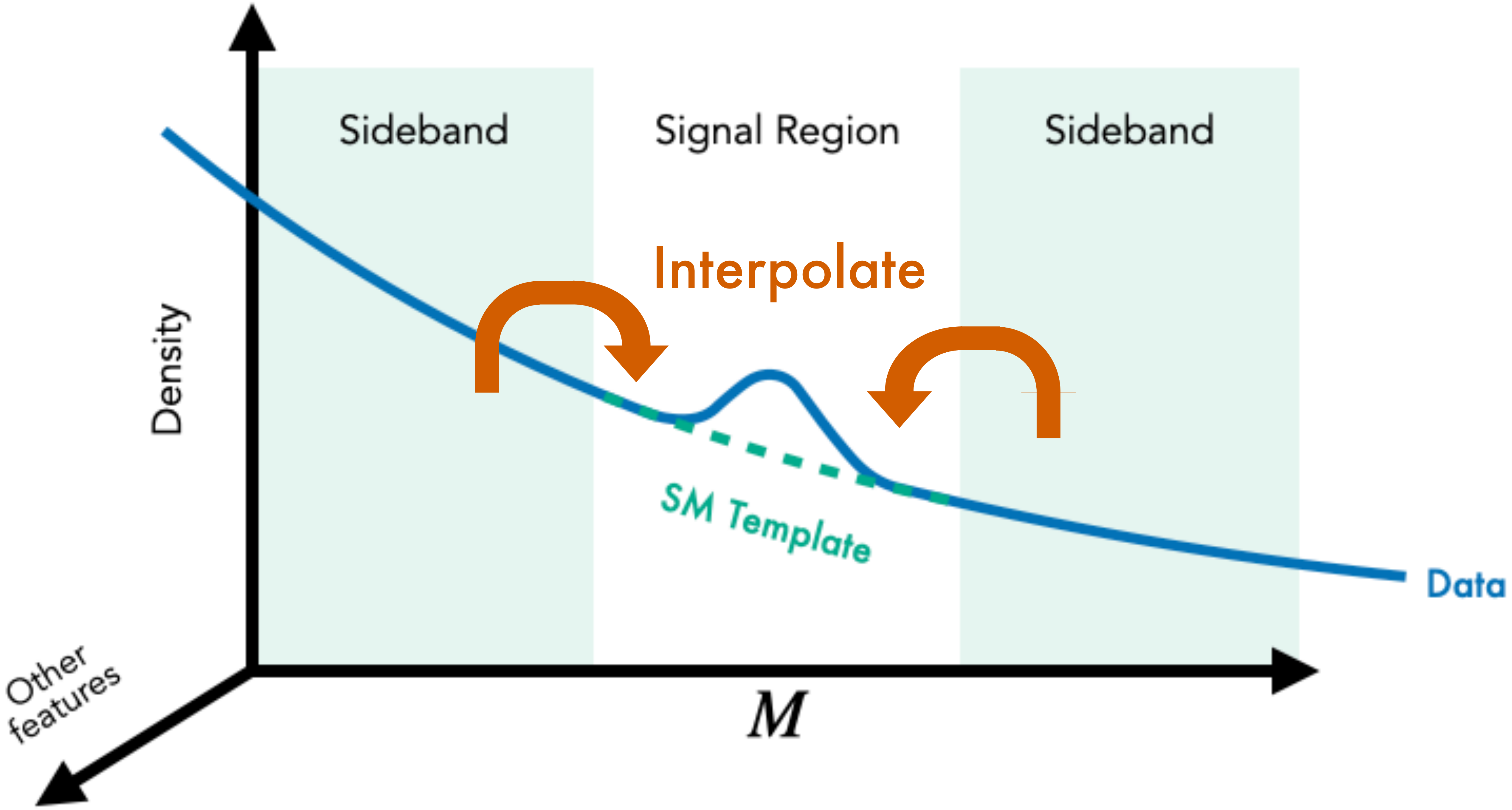
The CATHODE method

$p_{\omega_0}(x|m) \simeq p_{\text{bg}}(x|m)$ Trained in $m \in \text{SB}$

~~$p_{\omega_1}(x|m) \simeq p_{\text{data}}(x|m)$~~

1. Interpolate **SM background template** to SR and sample:

$$\hat{x}_{\text{bg}} \sim p_{\omega_0}(x|m \in \text{SR}) \simeq p_{\text{bg}}(x|\text{SR})$$



[2109.00546]

Best of both worlds — CATHODE

The CATHODE method

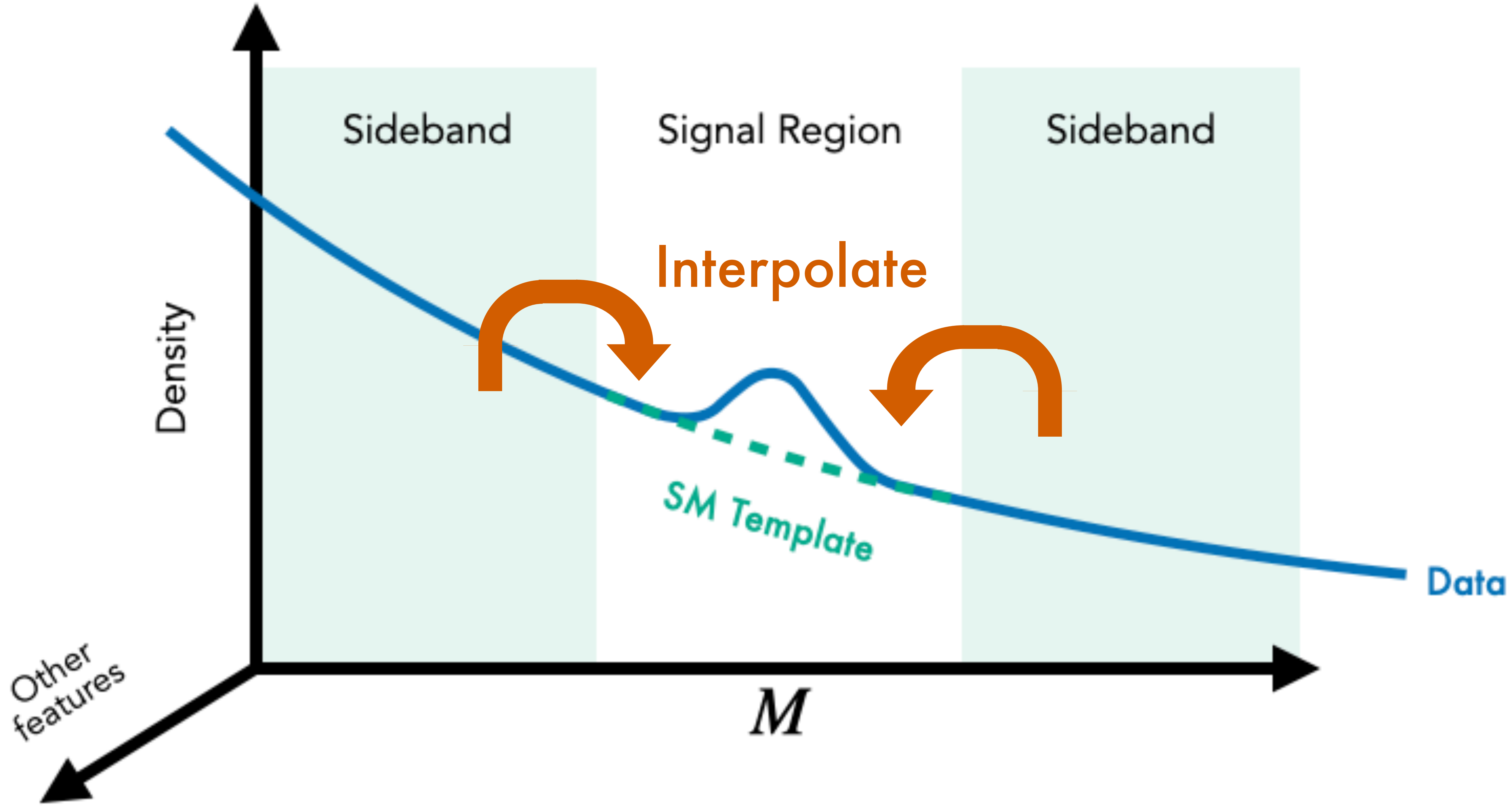
$p_{\omega_0}(x\,|\,m) \simeq p_{\text{bg}}(x\,|\,m)$ Trained in $m \in \text{SB}$

~~$p_{\omega_1}(x\,|\,m) \simeq p_{\text{data}}(x\,|\,m)$~~

1. Interpolate **SM background template** to SR and sample:

   $$\hat{x}_{\text{bg}} \sim p_{\omega_0}(x\,|\,m \in \text{SR}) \simeq p_{\text{bg}}(x\,|\,\text{SR})$$

2. Then **train classifier** between $\hat{x}_{\text{bg}}$ and $x \sim p_{\text{data}}(x\,|\,\text{SR})$ as in **CWoLA**



[2109.00546]

Best of both worlds — CATHODE

The CATHODE method

$$p_{\omega_0}(x\,|\,m) \simeq p_{\rm bg}(x\,|\,m)$$ Trained in $m \in \mathrm{SB}$

~~$p_{\omega_1}(x\,|\,m) \simeq p_{\rm data}(x\,|\,m)$~~

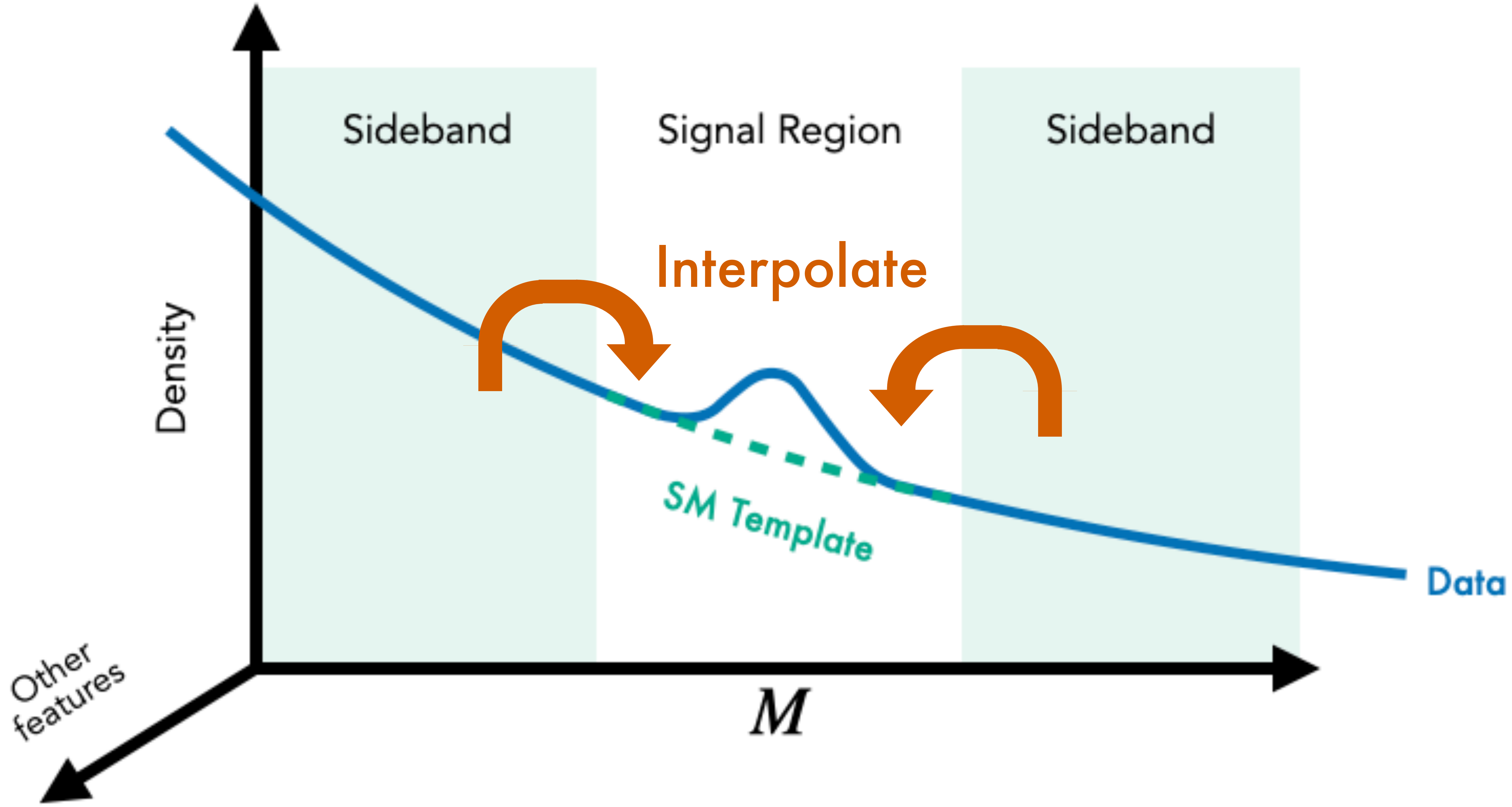
1. Interpolate **SM background template** to SR and sample:

$$\hat{x}_{\rm bg} \sim p_{\omega_0}(x\,|\,m \in \mathrm{SR}) \simeq p_{\rm bg}(x\,|\,\mathrm{SR})$$

2. Then **train classifier** between $\hat{x}_{\rm bg}$ and $x \sim p_{\rm data}(x\,|\,\mathrm{SR})$ as in **CWoLA**

CATHODE Likelihood estimate

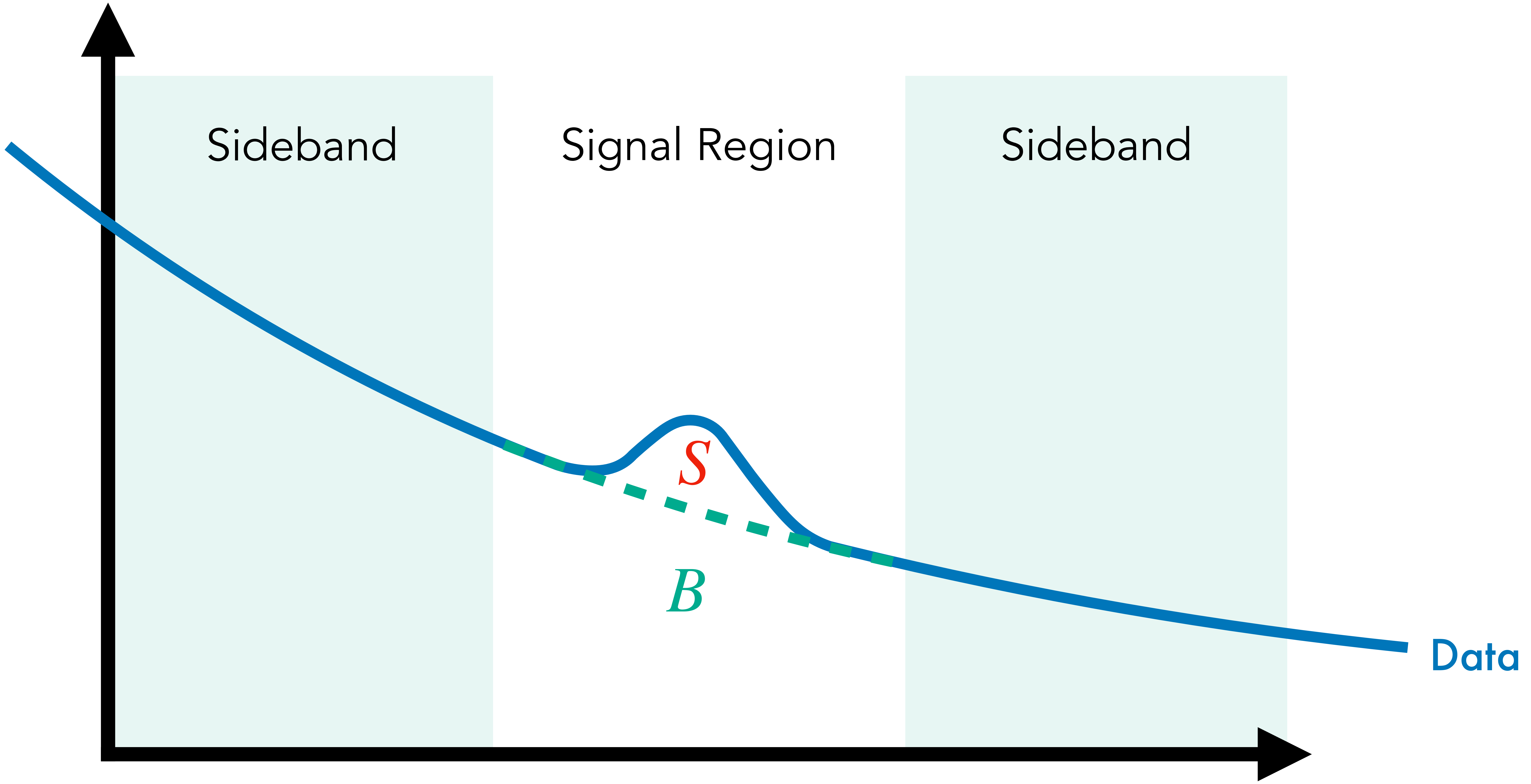
$$R_{\rm CATHODE} = \frac{p_{\rm data}(x\,|\,\mathrm{SR})}{p_{\omega_0}(x\,|\,\mathrm{SR})} \simeq \frac{p_{\rm data}(x\,|\,\mathrm{SR})}{p_{\rm bg}(x\,|\,\mathrm{SR})}$$



[2109.00546]

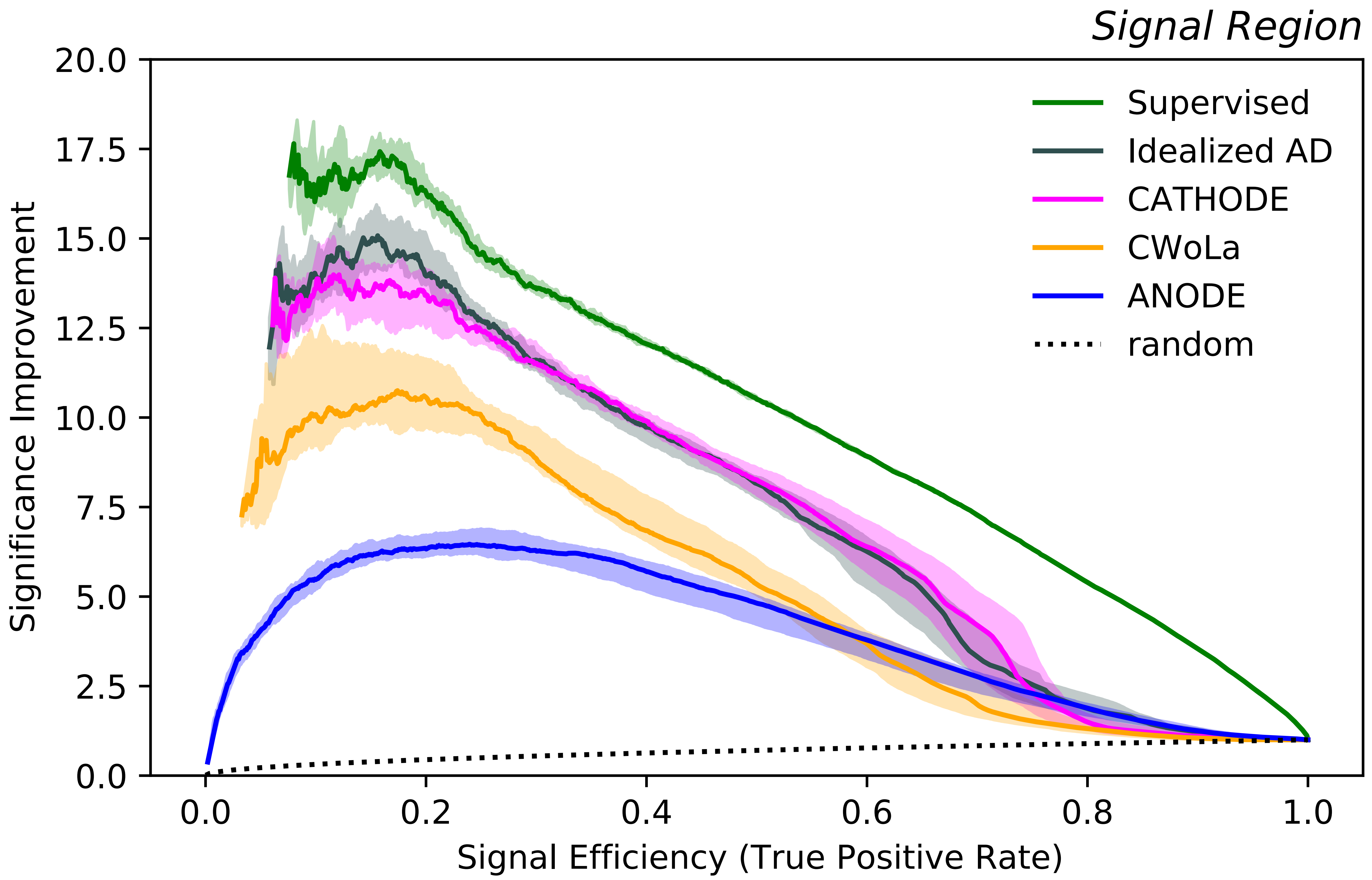
How do they compare?

How to quantify improvement?

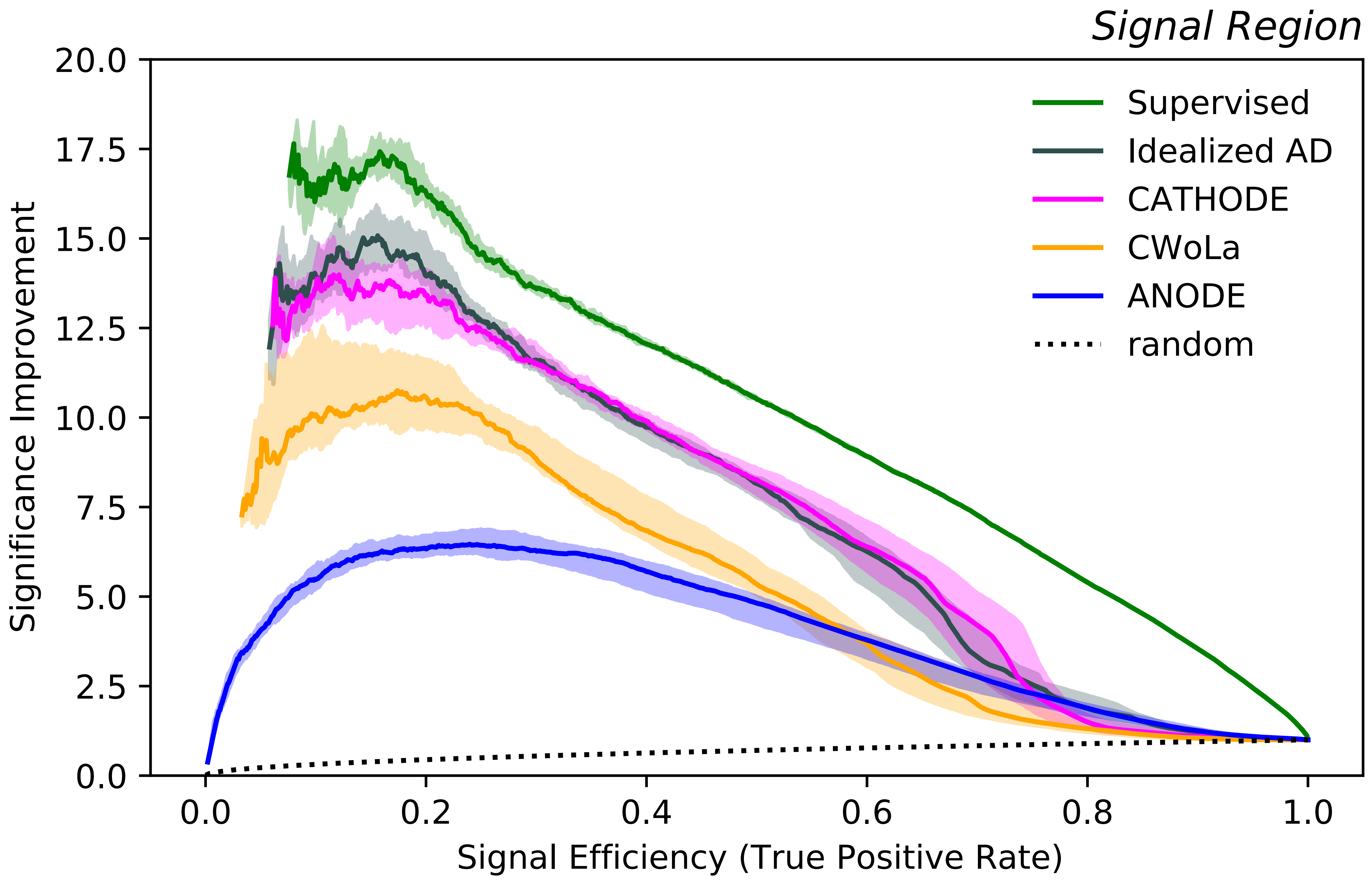


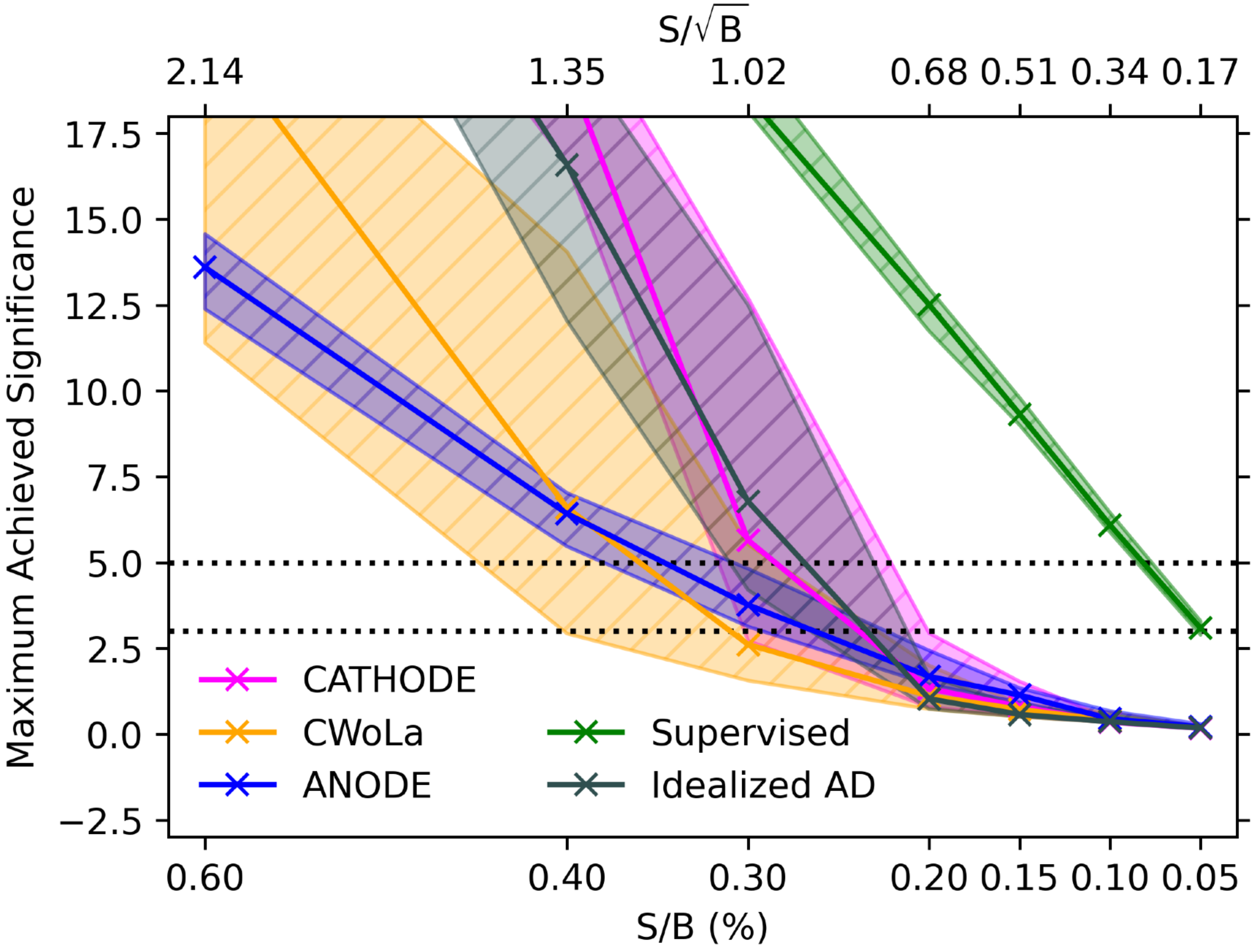
Statistical significance: $\frac{S}{\sqrt{B}}$ $\xrightarrow{\text{AD}}$ $\frac{S \cdot \epsilon_S}{\sqrt{B \cdot \epsilon_B}} = \frac{S}{\sqrt{B}} \cdot \frac{\epsilon_S}{\sqrt{\epsilon_B}}$ ← Improvement factor

Results — Comparison



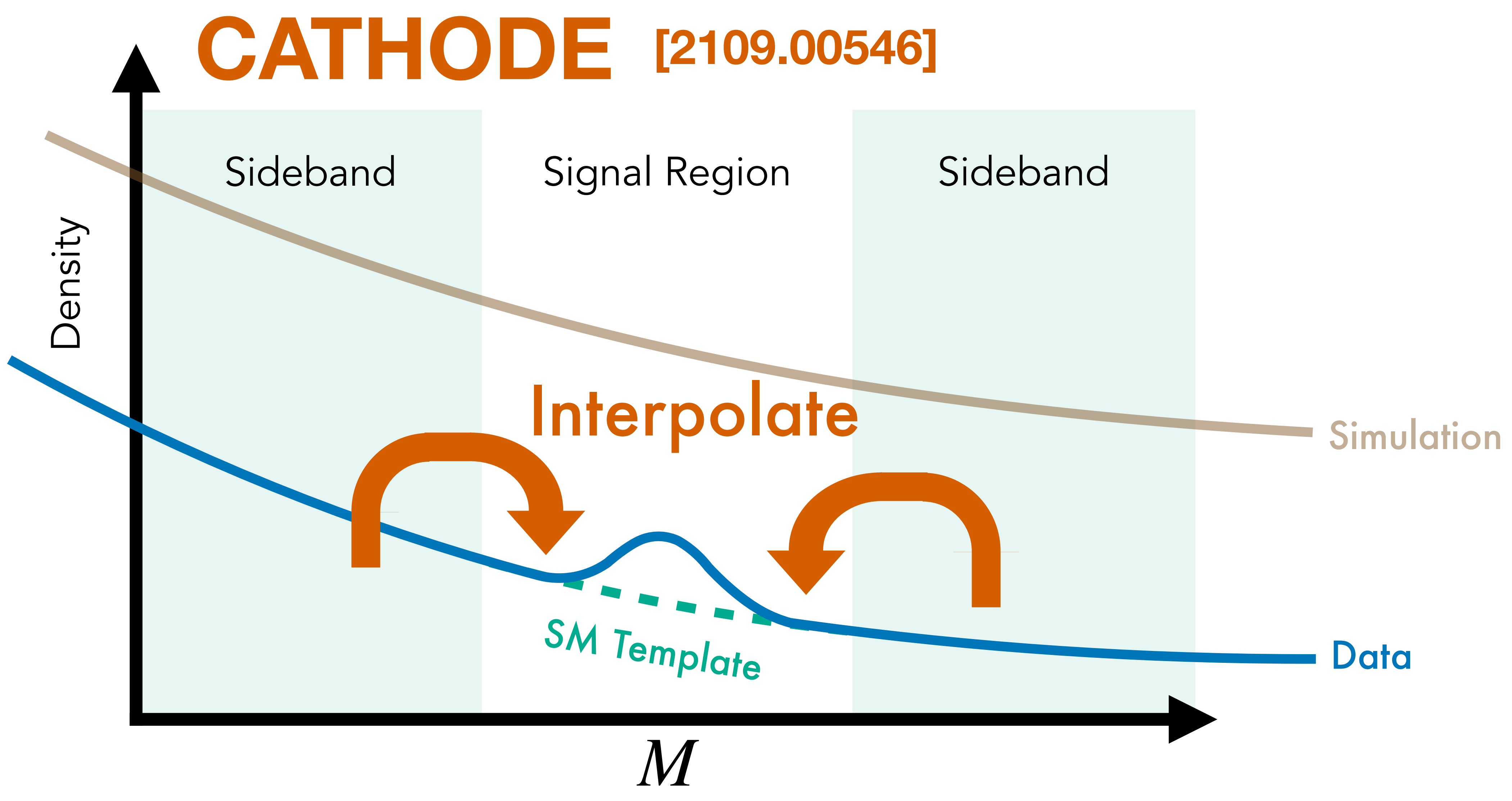
Results — Comparison



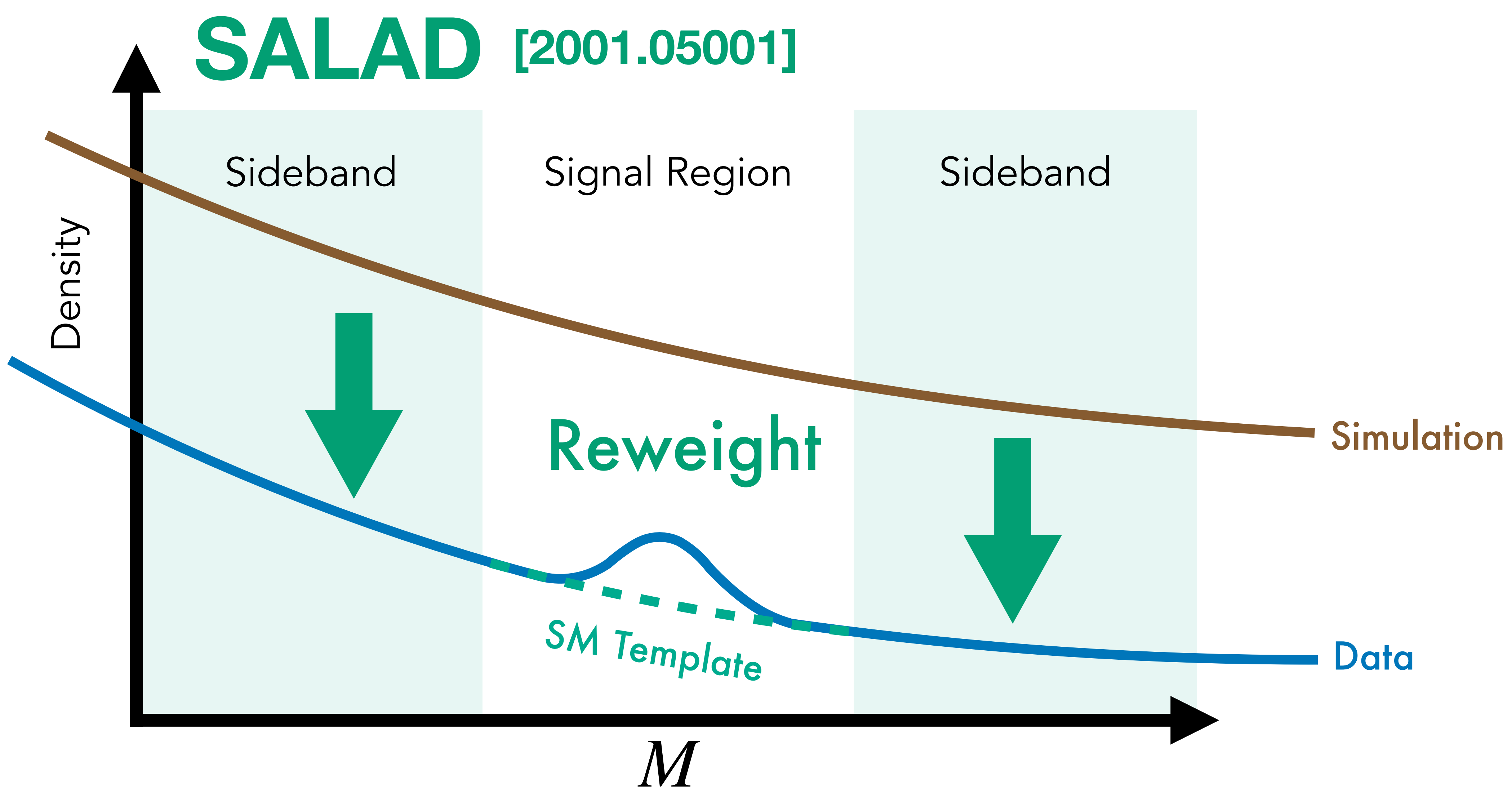


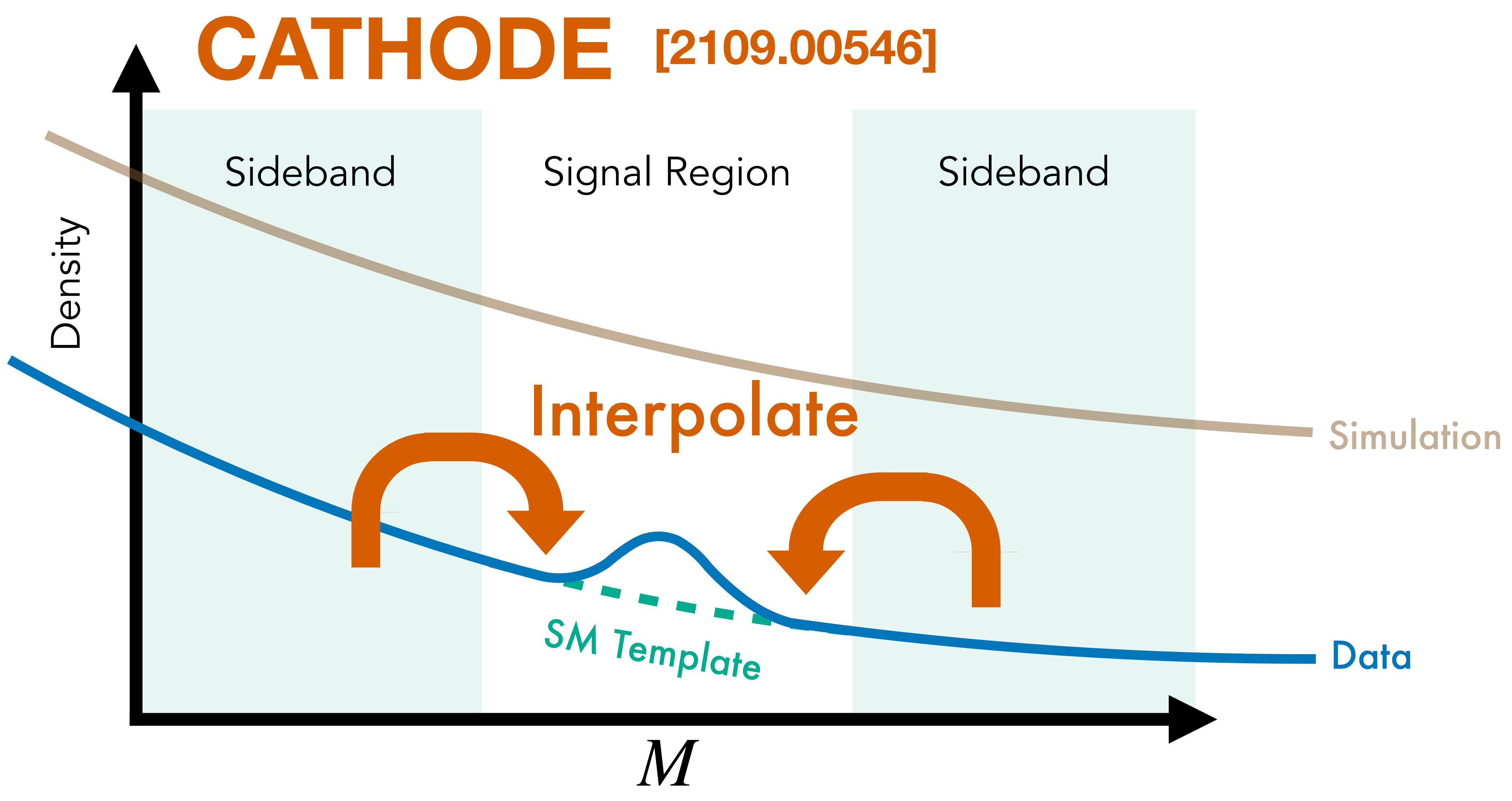
Are there other ways?

ML techniques to construct SM template



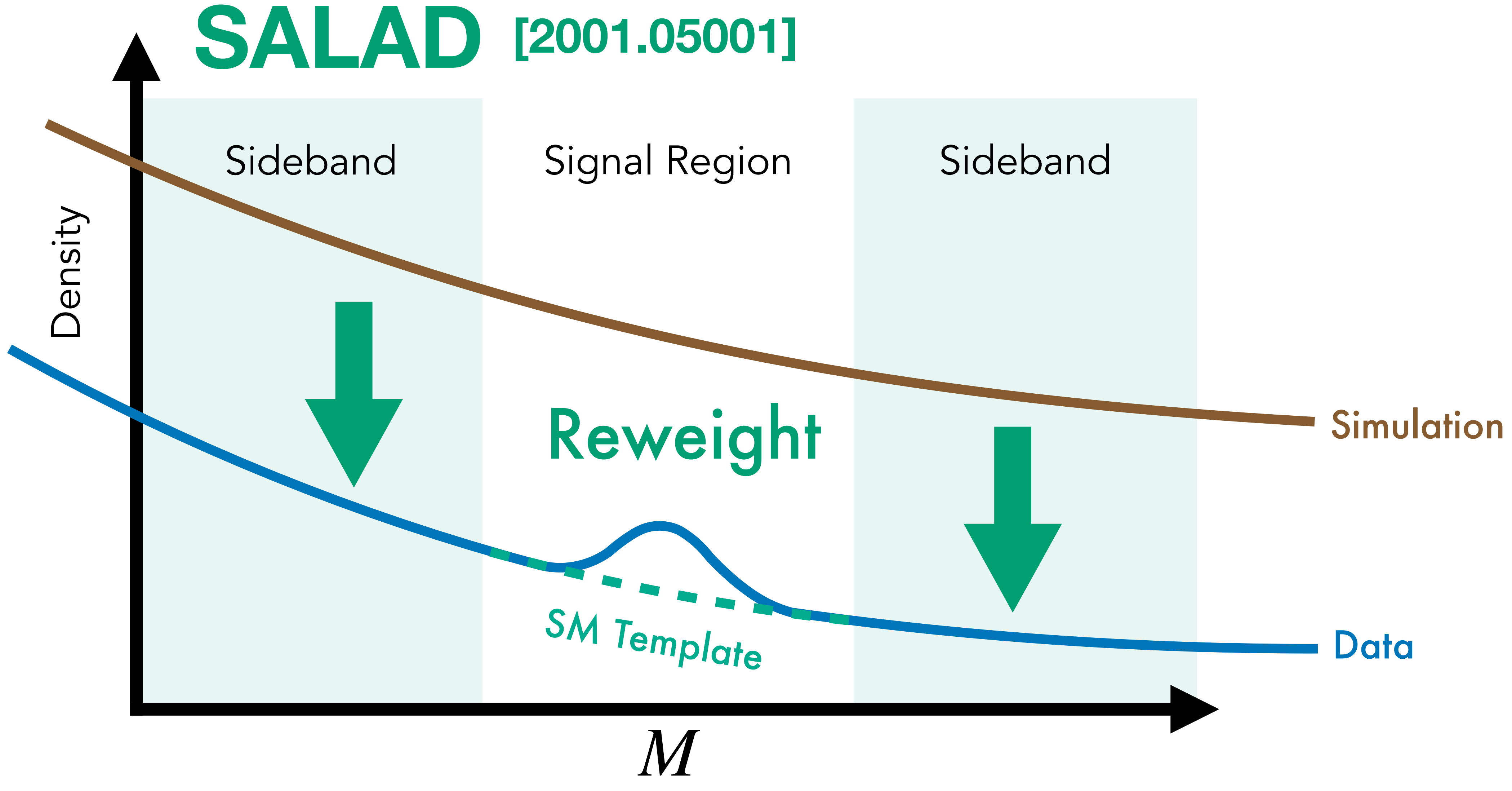
ML techniques to construct SM template



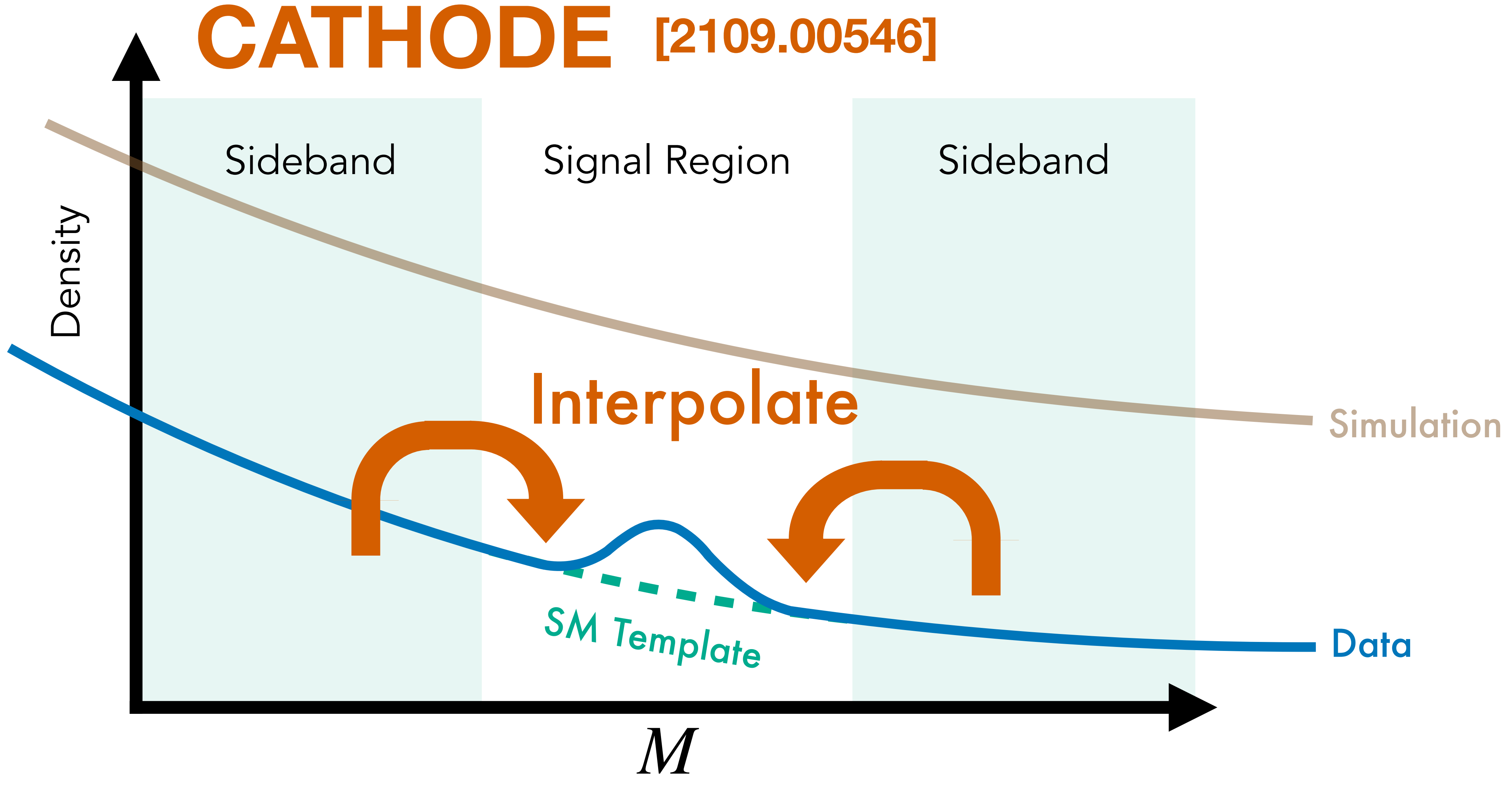


ML techniques to construct SM template

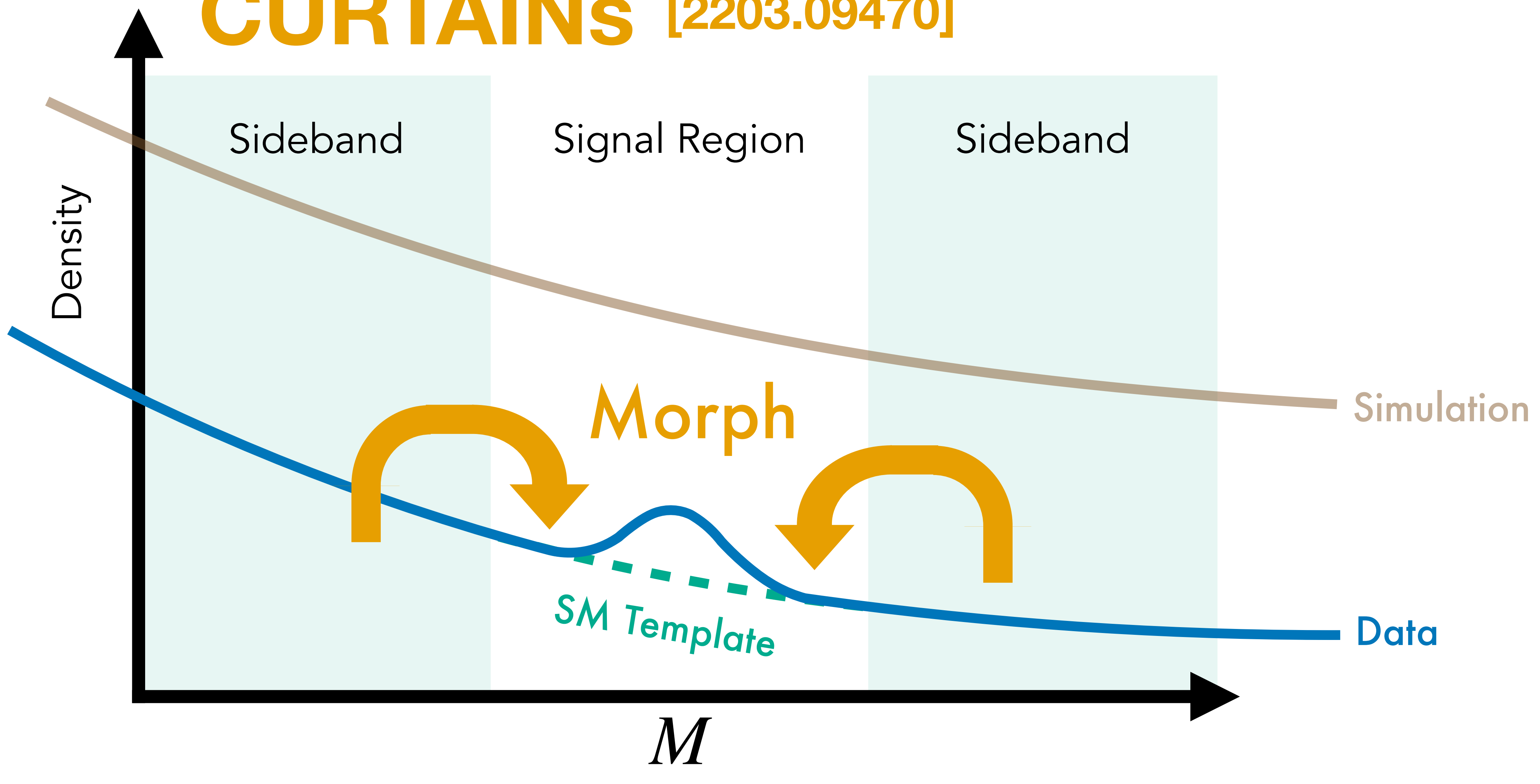
SALAD [2001.05001]



CATHODE [2109.00546]

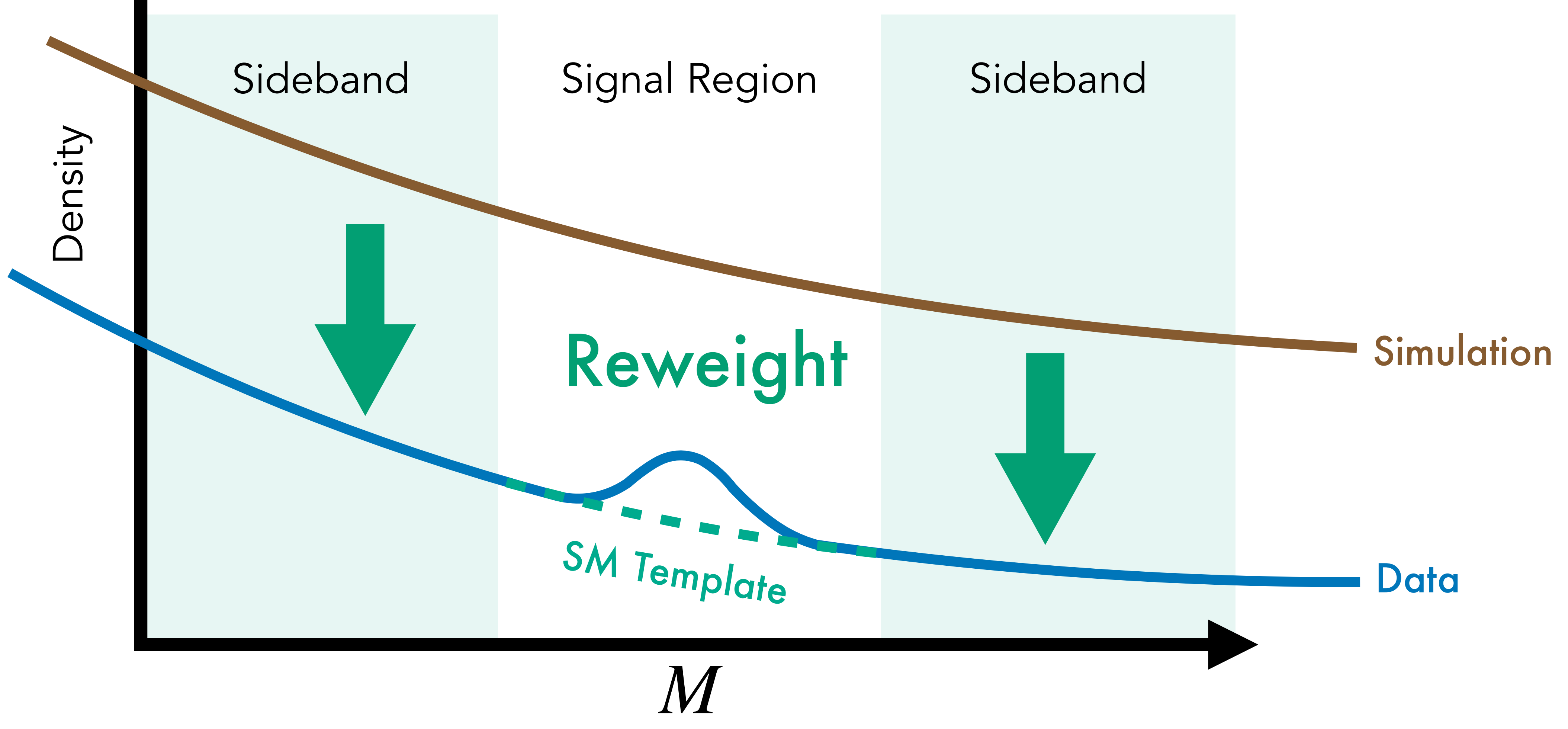


CURTAINs [2203.09470]

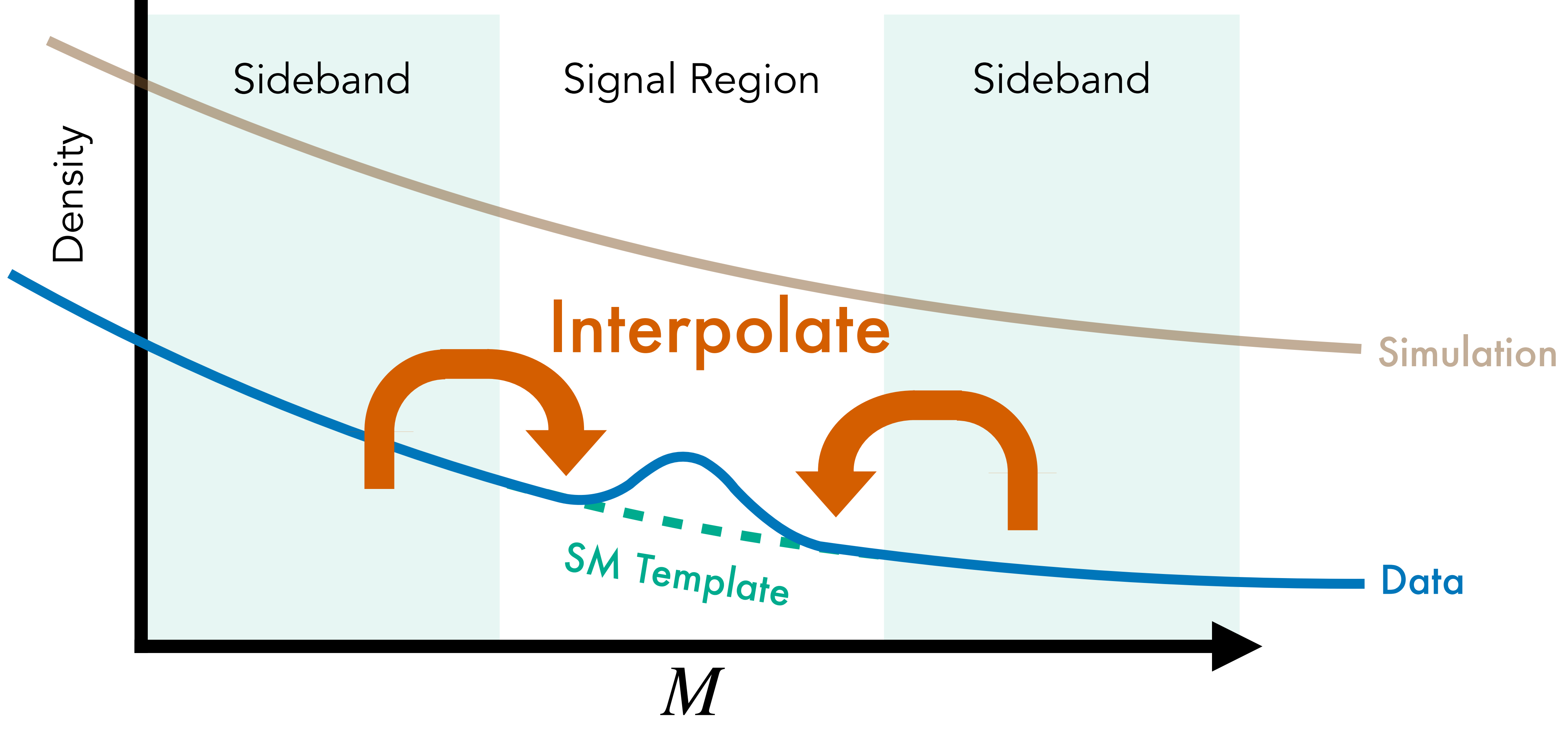


ML techniques to construct SM template

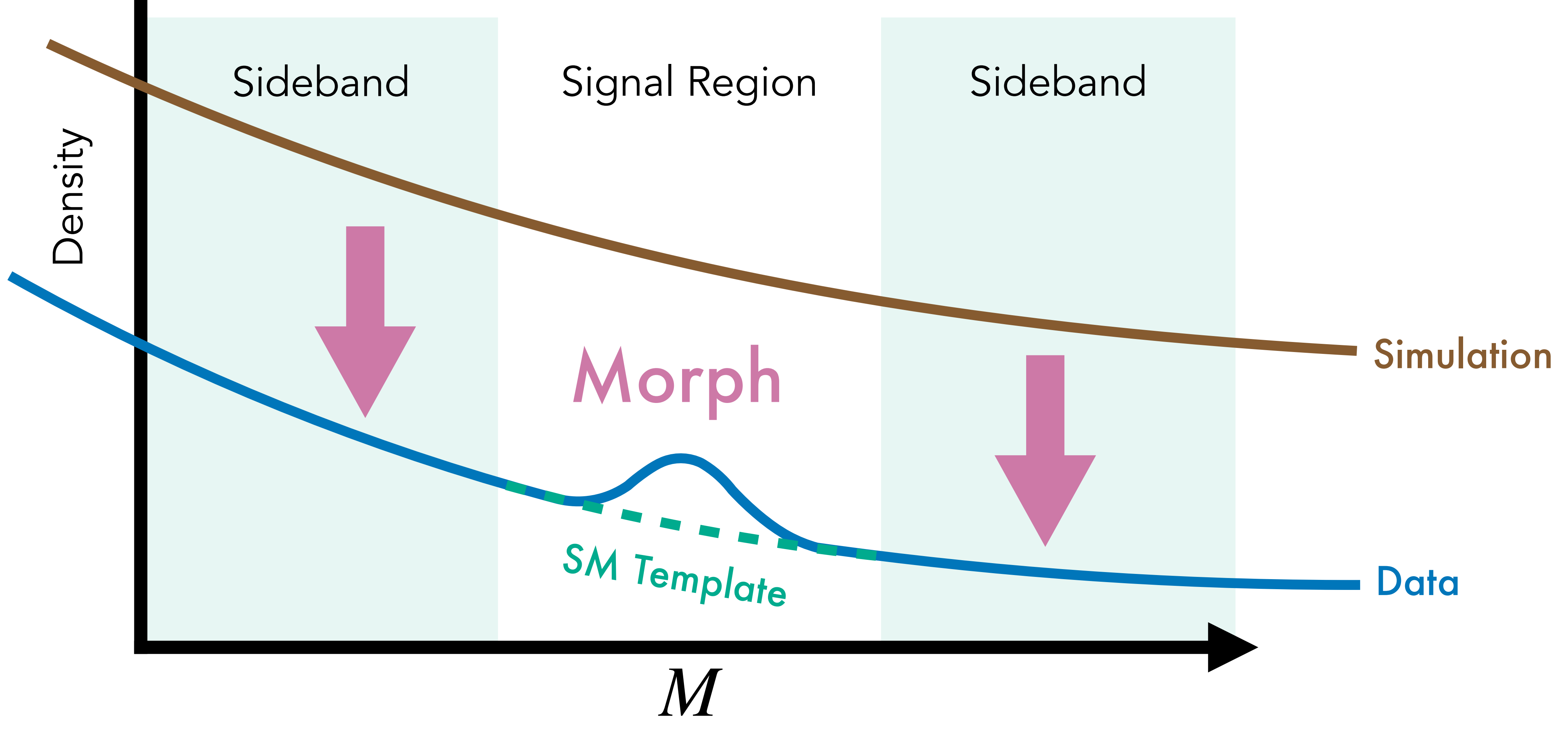
SALAD [2001.05001]



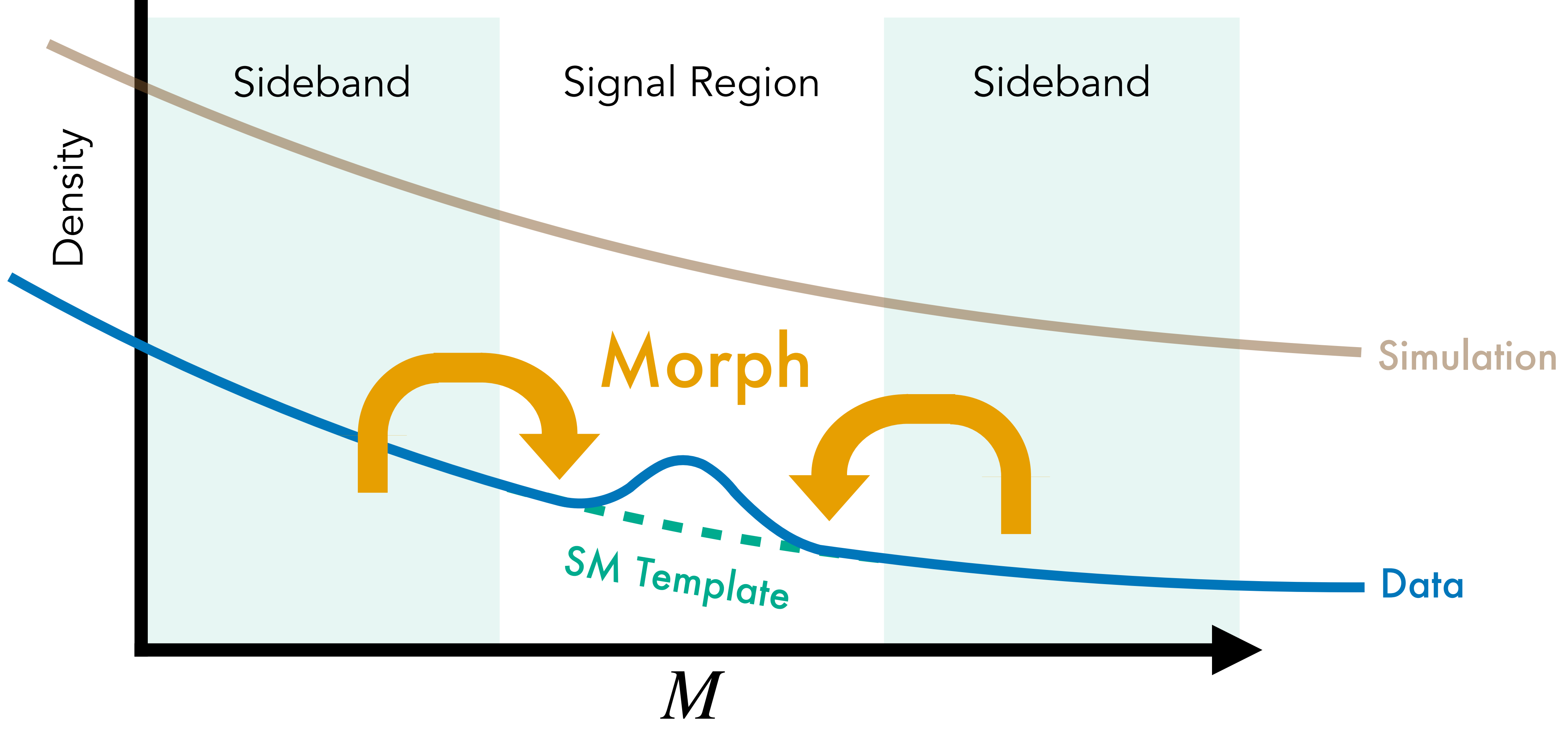
CATHODE [2109.00546]



FETA [2212.11285]



CURTAINs [2203.09470]



ML techniques to construct SM template

The Interplay of Machine Learning–based Resonant Anomaly Detection Methods

Tobias Golling,[a] Gregor Kasieczka,[b] Claudius Krause,[c] Radha Mastandrea,[d,e] Benjamin Nachman,[e,f] John Andrew Raine,[a] Debajyoti Sengupta,[a] David Shih,[g] and Manuel Sommerhalder[b]

[a] *Département de physique nucléaire et corpusculaire, Université de Genève, 1211 Genève, Switzerland*
[b] *Institut für Experimentalphysik, Universität Hamburg, 22761 Hamburg, Germany*
[c] *Institut für Theoretische Physik, Universität Heidelberg, 69120 Heidelberg, Germany*
[d] *Department of Physics, University of California, Berkeley, CA 94720, USA*
[e] *Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[f] *Berkeley Institute for Data Science, University of California, Berkeley, CA 94720, USA*
[g] *NHETC, Dept. of Physics and Astronomy, Rutgers University, Piscataway, NJ 08854, USA*

E-mail: tobias.golling@unige.ch, gregor.kasieczka@uni-hamburg.de, claudius.krause@thphys.uni-heidelberg.de, rmastand@berkeley.edu, bpnachman@lbl.gov, john.raine@unige.ch, debajyoti.sengupta@unige.ch, shih@physics.rutgers.edu, manuel.sommerhalder@uni-hamburg.de

Abstract: Machine learning–based anomaly detection (AD) methods are promising tools for extending the coverage of searches for physics beyond the Standard Model (BSM). One class of AD methods that has received significant attention is resonant anomaly detection, where the BSM physics is assumed to be localized in at least one known variable. While there have been many methods proposed to identify such a BSM signal that make use of simulated or detected data in different ways, there has not yet been a study of the methods' complementarity. To this end, we address two questions. First, in the absence of any signal, do different methods pick the same events as signal-like? If not, then we can significantly reduce the false-positive rate by comparing different methods on the same dataset. Second, if there is a signal, are different methods fully correlated? Even if their maximum performance is the same, since we do not know how much signal is present, it may be beneficial to combine approaches. Using the Large Hadron Collider (LHC) Olympics dataset, we provide quantitative answers to these questions. We find that there are significant gains possible by combining multiple methods, which will strengthen the search program at the LHC and beyond.

[2307.11157]

Summary and Outlook

Take-home messages

- ML beneficial in **every step** of the **simulation** and **analysis chain**
- We find both **proof-of-concepts** as well as established use cases (→ **AD, MadNIS,…**)
- Interesting **interplay** between **physics** and **ML**

Summary and Outlook

Take-home messages

- ML beneficial in **every step** of the **simulation** and **analysis chain**
- We find both **proof-of-concepts** as well as established use cases (→ **AD, MadNIS,…**)
- Interesting **interplay** between **physics** and **ML**

  → Physics provides **~infinite data** for ML

Summary and Outlook

Take-home messages

- ML beneficial in **every step** of the **simulation** and **analysis chain**
- We find both **proof-of-concepts** as well as established use cases (→ **AD, MadNIS,…**)
- Interesting **interplay** between **physics** and **ML**

  → Physics provides **~infinite data** for ML

  → Physics requirements (**precision, symmertries,…**) **different** than industry applications

Summary and Outlook

Take-home messages

- ML beneficial in **every step** of the **simulation** and **analysis chain**
- We find both **proof-of-concepts** as well as established use cases (→ **AD, MadNIS,…**)
- Interesting **interplay** between **physics** and **ML**

  → Physics provides **~infinite data** for ML

  → Physics requirements (**precision, symmertries,…**) **different** than industry applications

Future exercises

- Full integration of ML-based methods into standard tools → **Taggers, MadGraph,….**
- Make everything run on **GPUs** and make it **differentiable**
- Foster deeper collaboration between **theory**, **experiment**, and **ML** community