Differentiable binning optimisation in multidimensional phase space

Johannes Erdmann, Nitish Kumar Kasaraguppe, Florian Mausolf
III. Physikalisches Institut A, RWTH Aachen University

Differential analysis blueprint workshop
05 March, 2026

Research Training Group Physics of the Heaviest Particles at the LHC | RWTH AACHEN UNIVERSITY

Physics Institute III A | RWTH AACHEN UNIVERSITY

Overview

- Typical HEP analysis workflow deals with many collision events
  → Only a small subset contributes significantly to the measurement sensitivity
  → We categorise events to optimise the sensitivity to a signal

- **How to categorise for optimal signal significance?**
  - Often chosen manually
    → Popular choice is to apply *argmax* projections of multi-class scores and bin equidistantly in 1D

- We present a method to directly optimise signal significance in multi-dimensional discriminants using **differentiable approach**
  → Learning to bin (arXiv:2601.07756)

- Algorithms shared as Python package, lightweight plugin for analyses: gato-hep (**G**radient-based c**AT**egorization **O**ptimizer) available on PyPI, GitHub
  → Built on TensorFlow's automatic differentiation

gato-hep

TensorFlow

Setup

- Toy events are sampled from multivariate normal distribution

$$\mathcal{N}(\mu_i, \Sigma)$$

→ Each process has specific means μ_i and a common covariance matrix

Process	Scenario 1 $\mu_x^{(1)}$	$\mu_y^{(1)}$	$\mu_z^{(1)}$	Scenario 2 $\mu_x^{(2)}$	$\mu_y^{(2)}$	$\mu_z^{(2)}$	σ [pb]
Signal 1	0.4	−0.4	1.0	0.9	−0.9	1.0	0.5
Signal 2	−0.4	0.4	1.0	−0.9	0.9	1.0	0.1
Background 1	0.0	0.0	−0.5	0.2	0.2	0.5	100.0
Background 2	−0.2	0.0	0.0	–	–	–	80.0
Background 3	0.1	0.4	−0.3	–	–	–	50.0
Background 4	−0.1	0.6	−0.2	–	–	–	20.0
Background 5	−0.2	0.1	−0.1	–	–	–	10.0

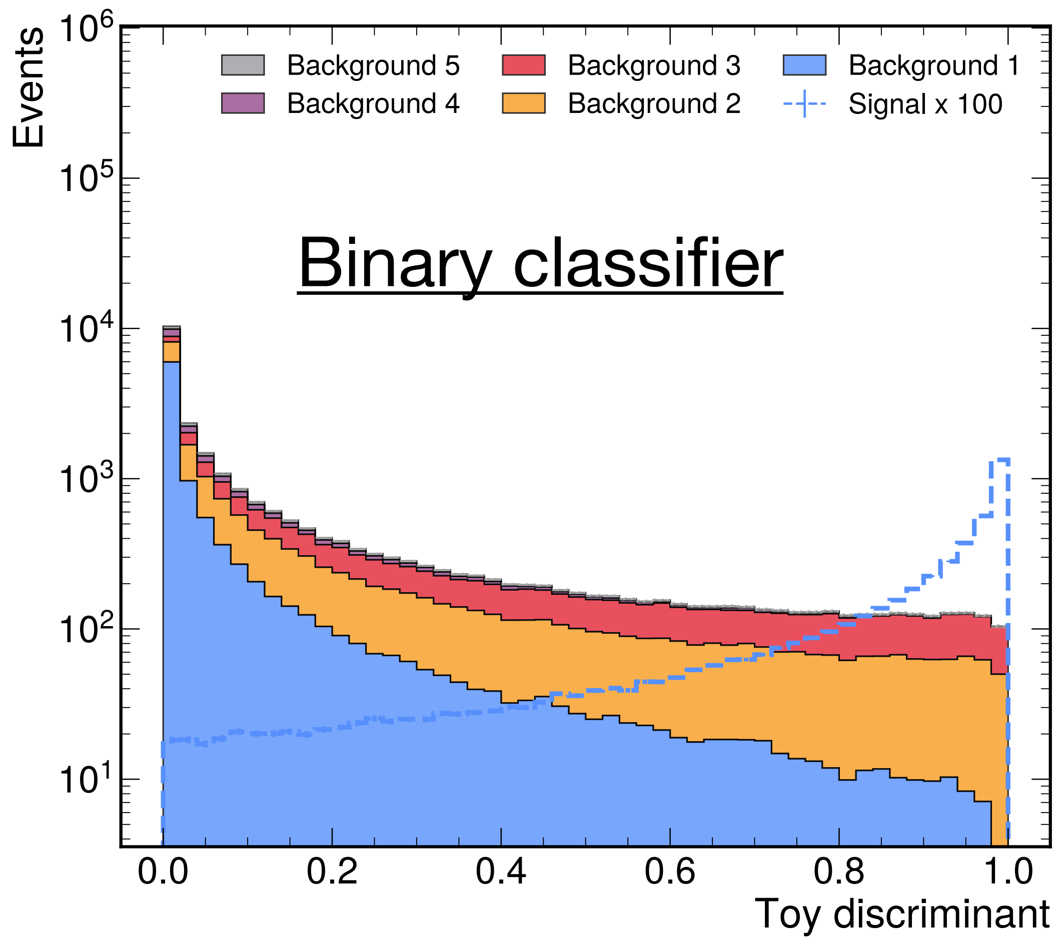
$$\Sigma = \mathrm{diag}(1,1,1) + 0.2\,(\mathbf{1} - \mathrm{diag}(1,1,1))$$

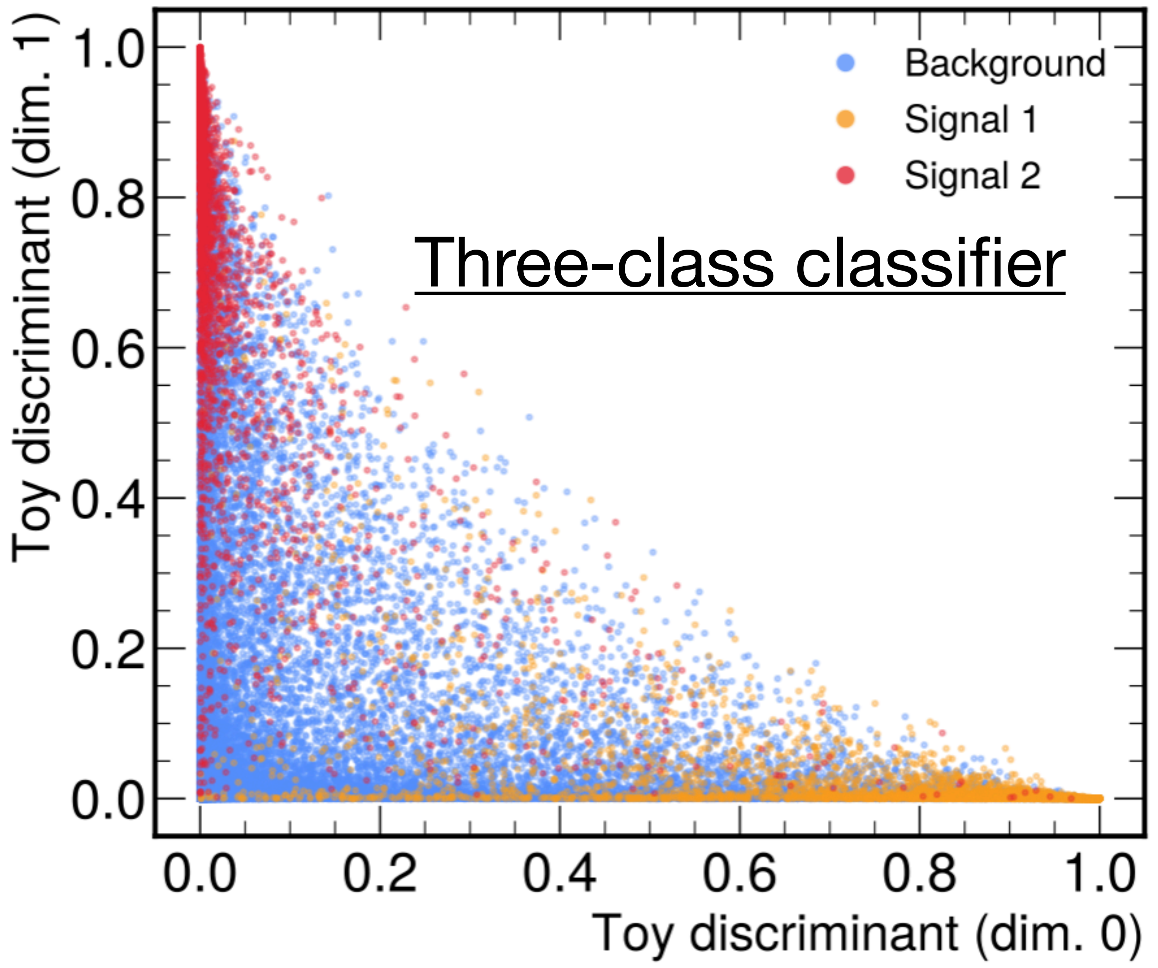
- Two scenarios of toy dataset:
  → Scenario 1: significant overlap between signals and background
  → Scenario 2: separation between signals and background nearly perfect

- One-dimensional case: consider **one signal + combined background**
  → Build signal-to-background likelihood ratio and normalise to $[0,1]$

- Three-dimensional case: consider **two signal + combined background**
  → Build softmax-like scores with three outputs summing to 1

- We optimise the bin boundaries to maximise the asymptotic significance significance: $Z_A = \sqrt{2\left[(S+B)\ln\left(1+\frac{S}{B}\right) - S\right]}$ (G. Cowan et al, 1007.1727)





Method: multi-dimensional binning

- In multi-dimensional categorisation problems (e.g. multi-classifier output or multiple 1D observables), more flexible solutions can be obtained using a **Gaussian Mixture Model (GMM)**
  → Cuts are not restricted to be rectangular

  - Define N **Gaussians** in d-dimensional space, one per bin k:

$$\ell_k(x) = \mathcal{N}(x \mid \boldsymbol{\mu}_k, \Sigma_k).$$

```
class gatohep.models.gato_gmm_model(n_cats, dim, temperature=1.0,
mean_norm='softmax', mean_range=(0.0, 1.0), name='gato_gmm_model')
```

  - The binning is performed such that an event is assigned to the component with the largest score $s_k(x)$

$$s_k(x) = \log \ell_k(x) + \log \pi_k.$$

  → Learnable mean: μ_k
  → Learnable covariance: Σ_k
  → Learnable mixture weight: π_k

Optimisation and inference

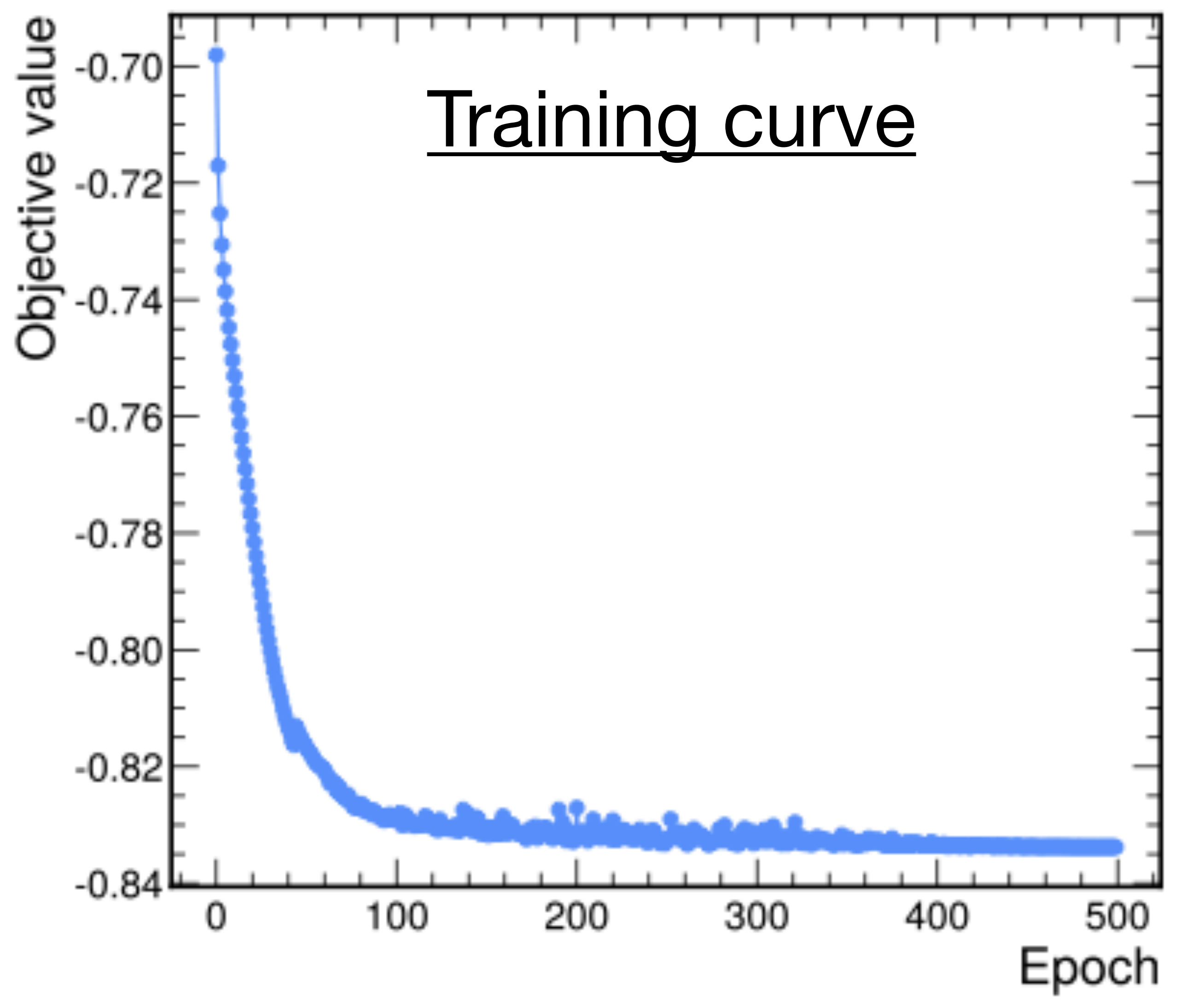
- **Optimisation**: the key is differentiable approximation
  - Obtained by replacing discrete maximisation with temperature scaled softmax of $s_k(x)$

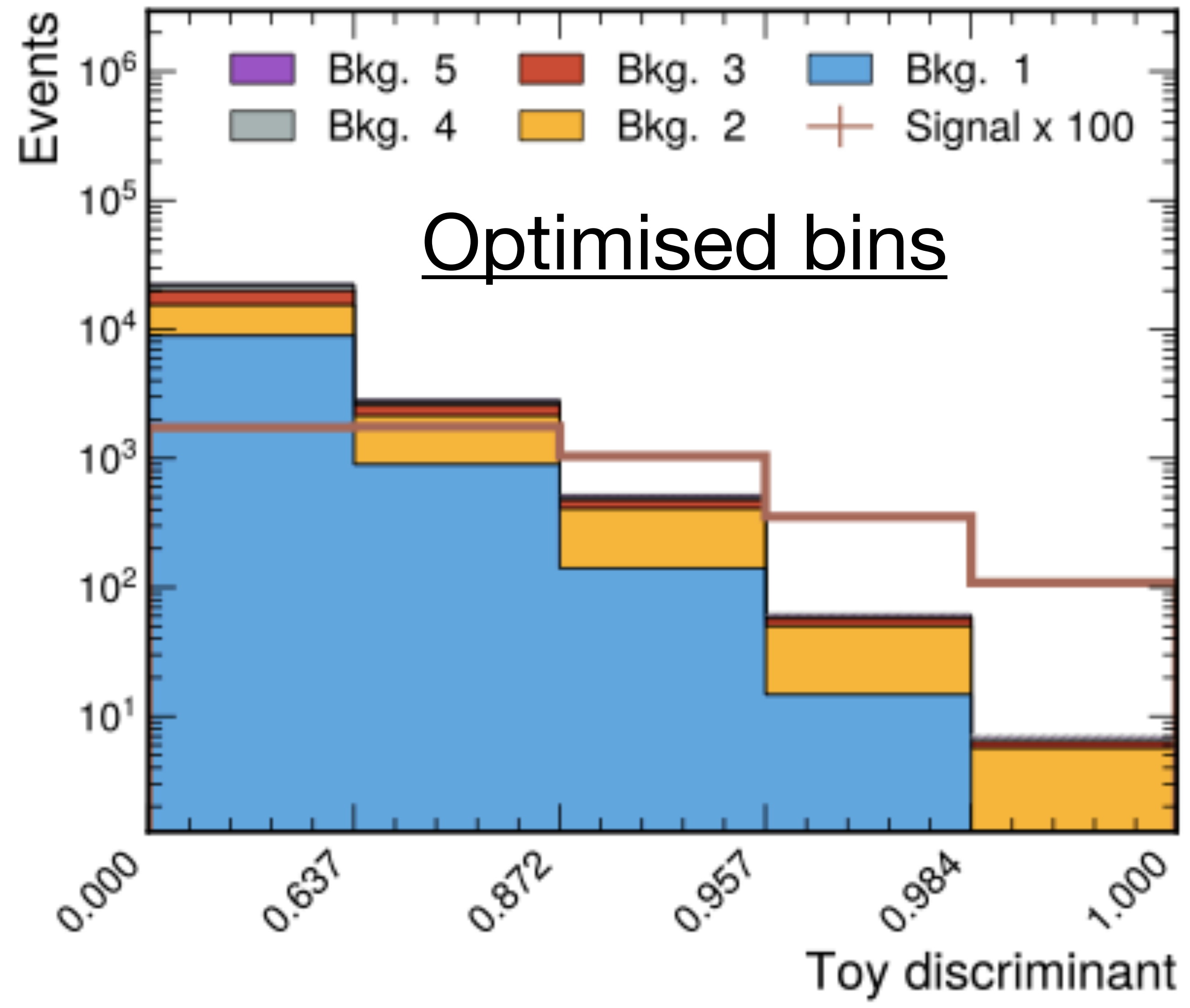
$$\gamma_k(x) = \mathrm{softmax}_k\left(\frac{s_k(x)}{T}\right)$$

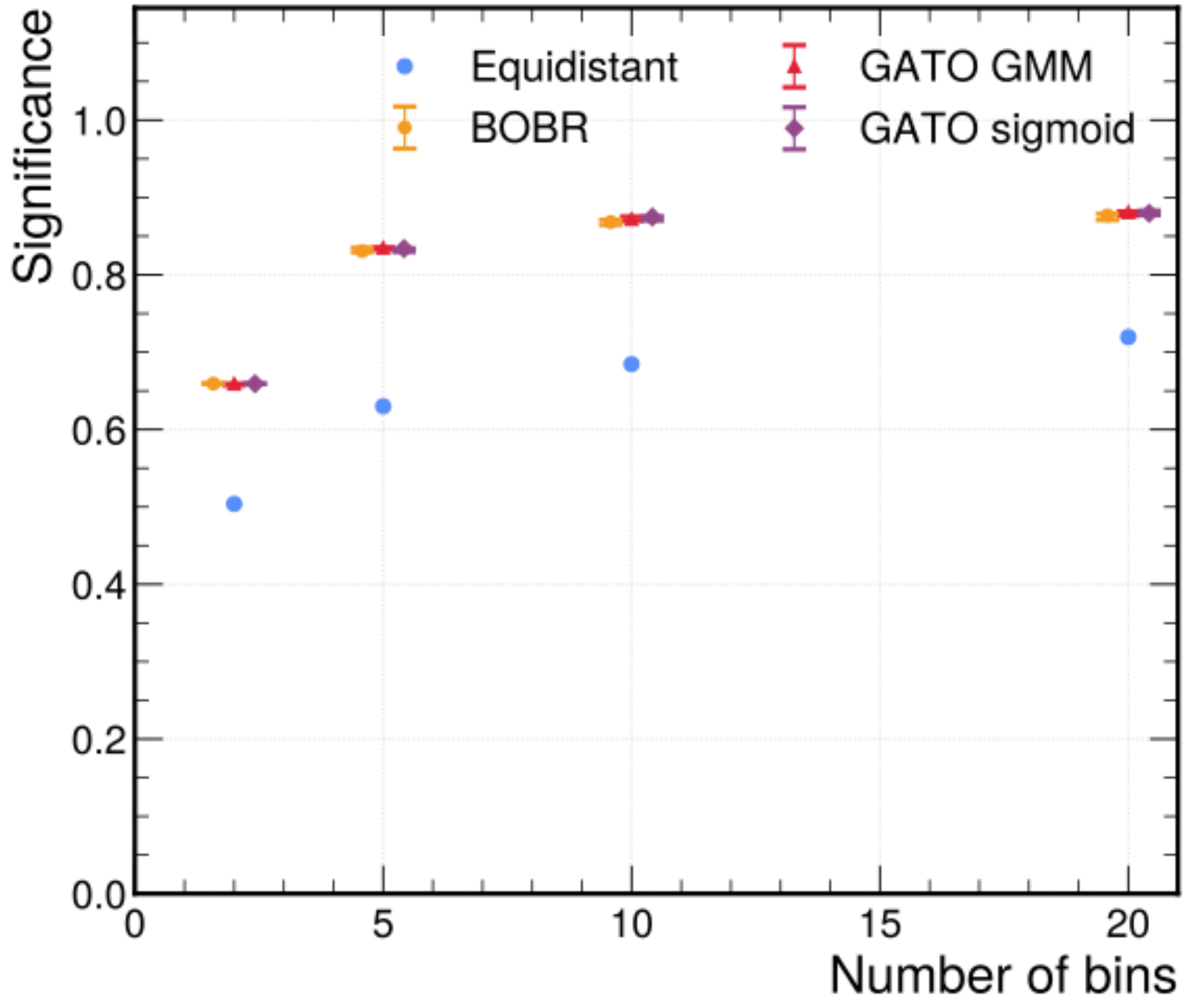
    $\rightarrow$ Each bin carries a fraction of each event

  - Temperature T controls level of approximation: $T \rightarrow 0$ gives hard assignment
- **Inference**:
  - Event is assigned to the component with the largest score $s_k(x)$

Results: one-dimensional case





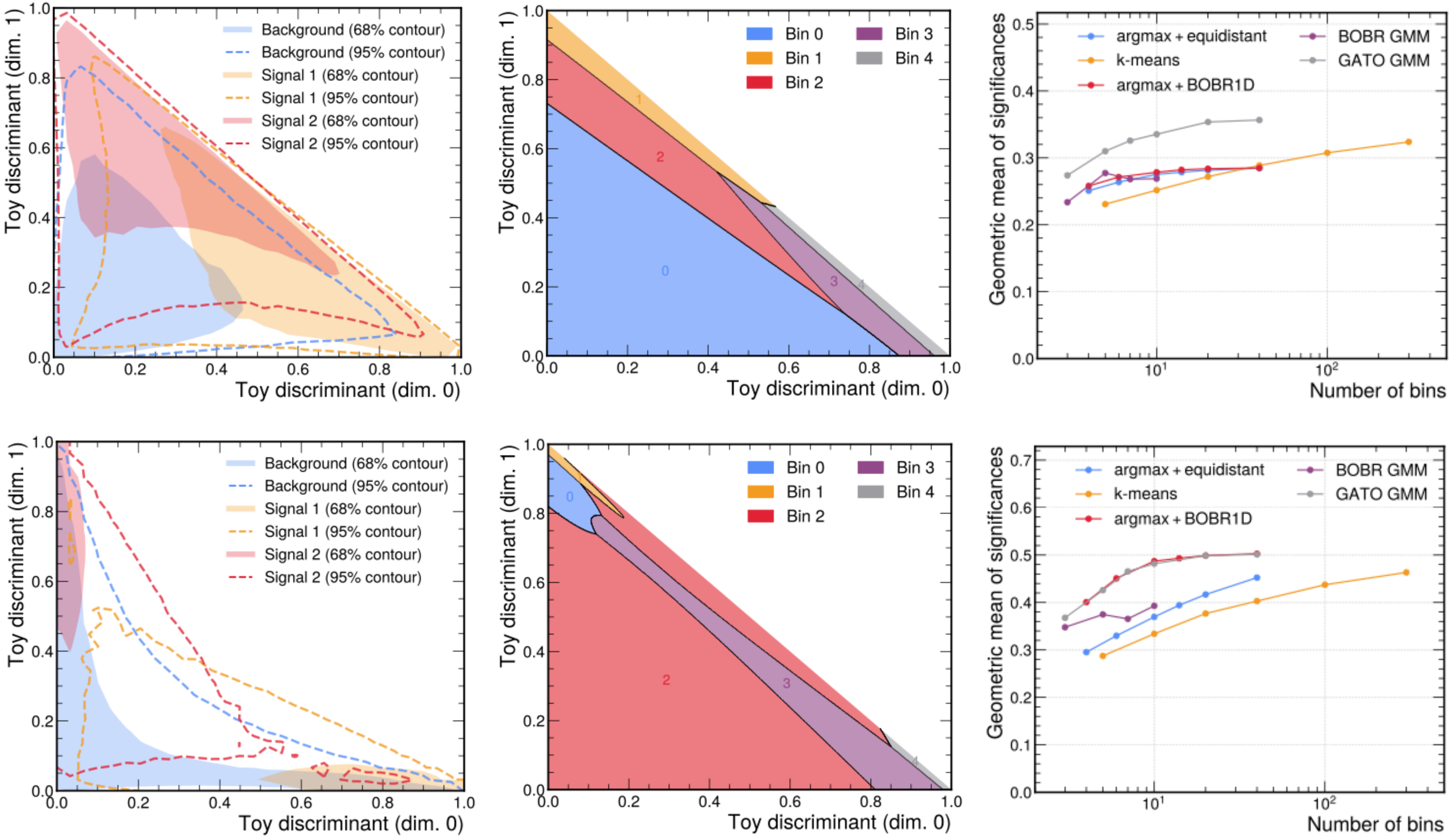


Gradient based method with GMM binning strategy performs significantly better compared to equidistant binning

***B**ayesian **O**ptimization of **B**in bounda**R**ies (**BOBR**) also available as a package: bobr-hep*

Results: three dimensional case

Results on two scenarios of toy datasets



Penalties during optimisation

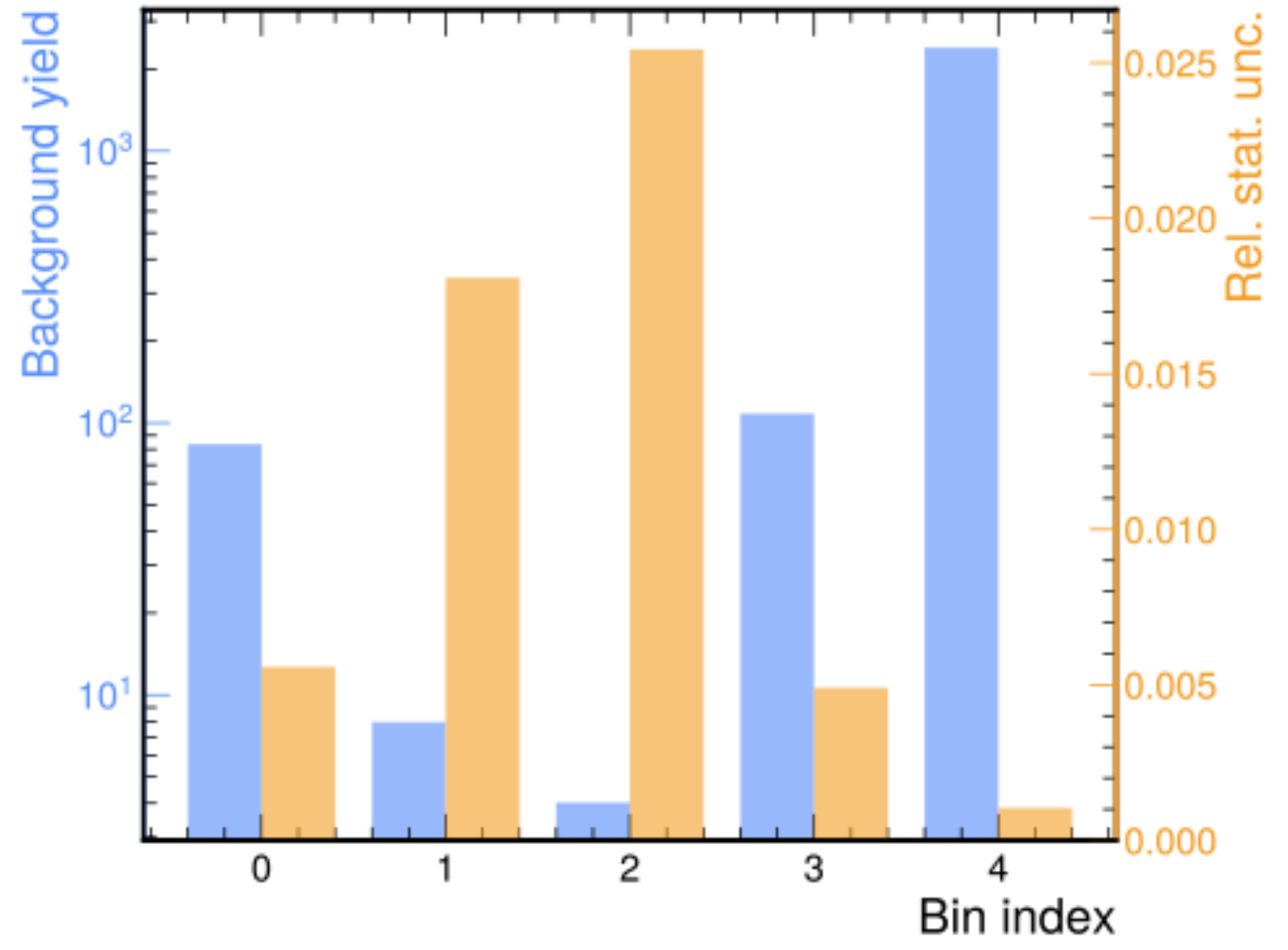
- We use penalties to discourage bins with zero background yield or large statistical uncertainty
  - Low background yield penalty:

$$P_{\text{low}}(B) = \sum_{k=1}^{K} \left[\max(0,\, B_{\min} - B_k)\right]^2$$

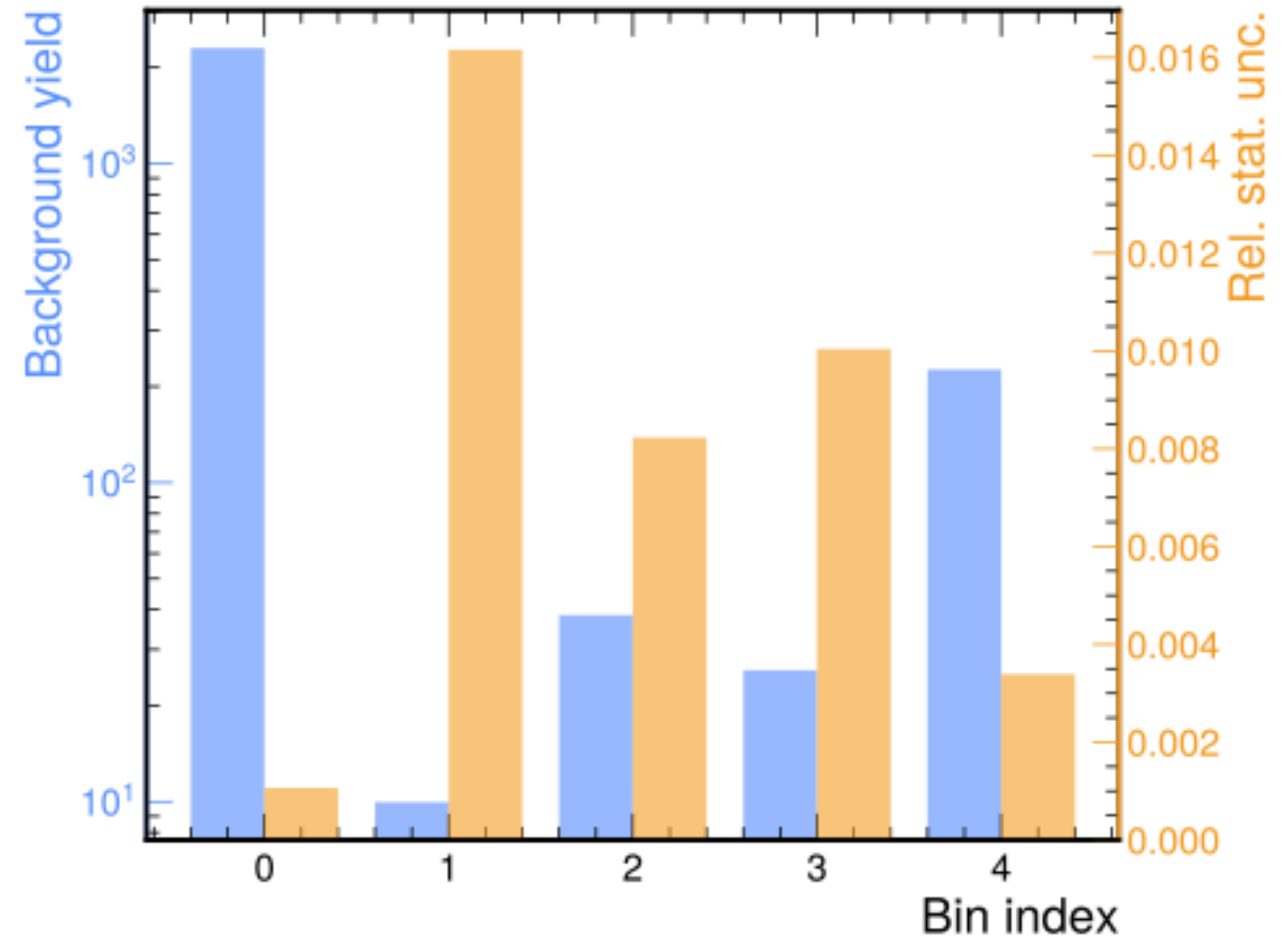
  - Large statistical uncertainty penalty:

$$P_{\text{unc}}(B) = \sum_{k=1}^{K} \left[\max(0,\, \sigma_{\text{rel},k} - r)\right]^2$$

Low background yield penalty of one event

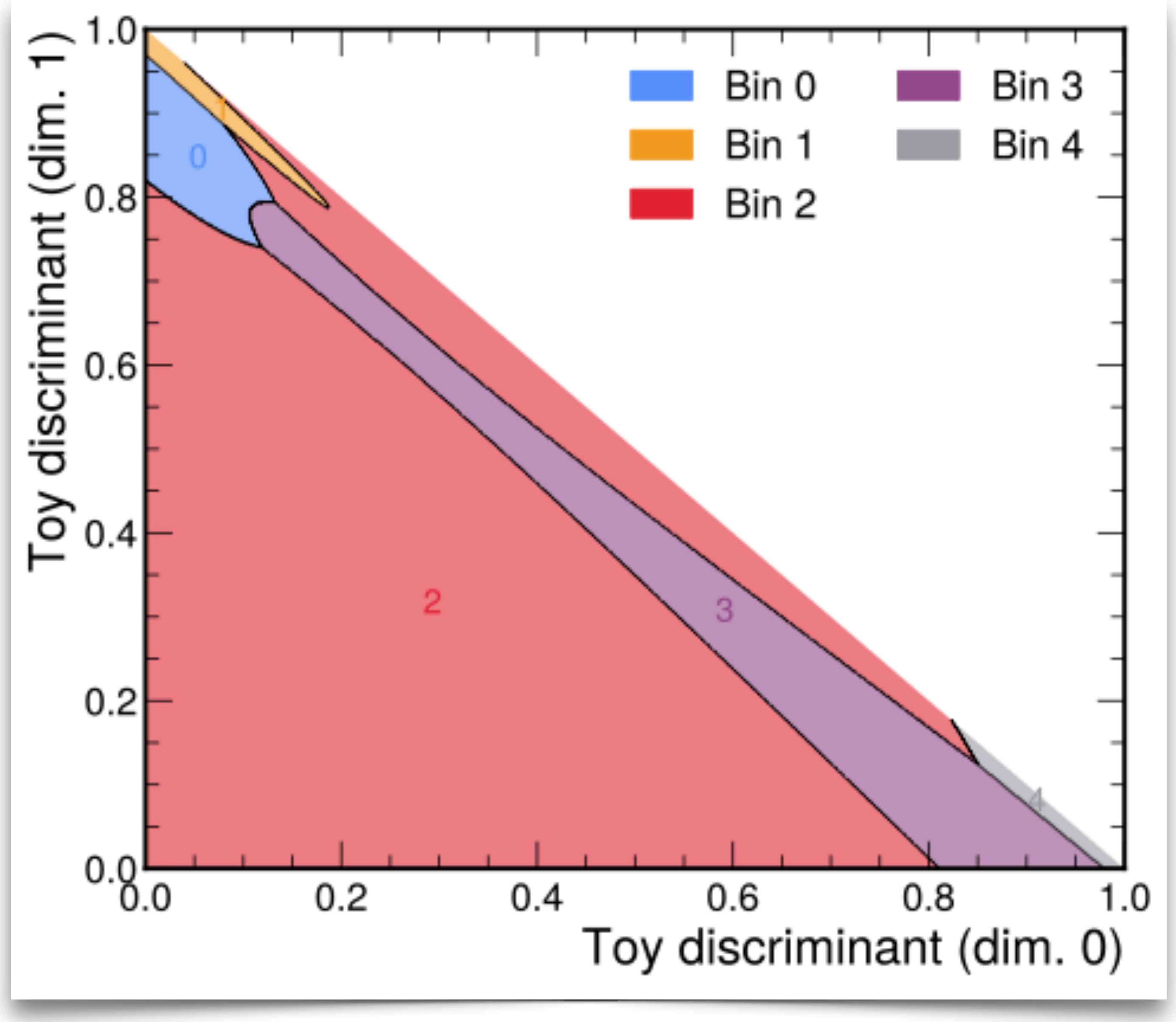


Low background yield penalty of ten events



Summary

- Novel approach for multidimensional binning optimisation based on Gaussian Mixture Models: Learning to bin (arXiv:2601.07756)
  → Learnable binning with flexible geometry in multidimensional phase space via differentiable formulation
  → **Can also be used as a piece in fully differentiable analysis pipeline**



- We presented optimisation for signal significance
  → Objective could also be any function of signal and background yields

- gato-hep shared as Python package, lightweight plugin for analyses
  → JAX-based implementation is also planned

gato-hep

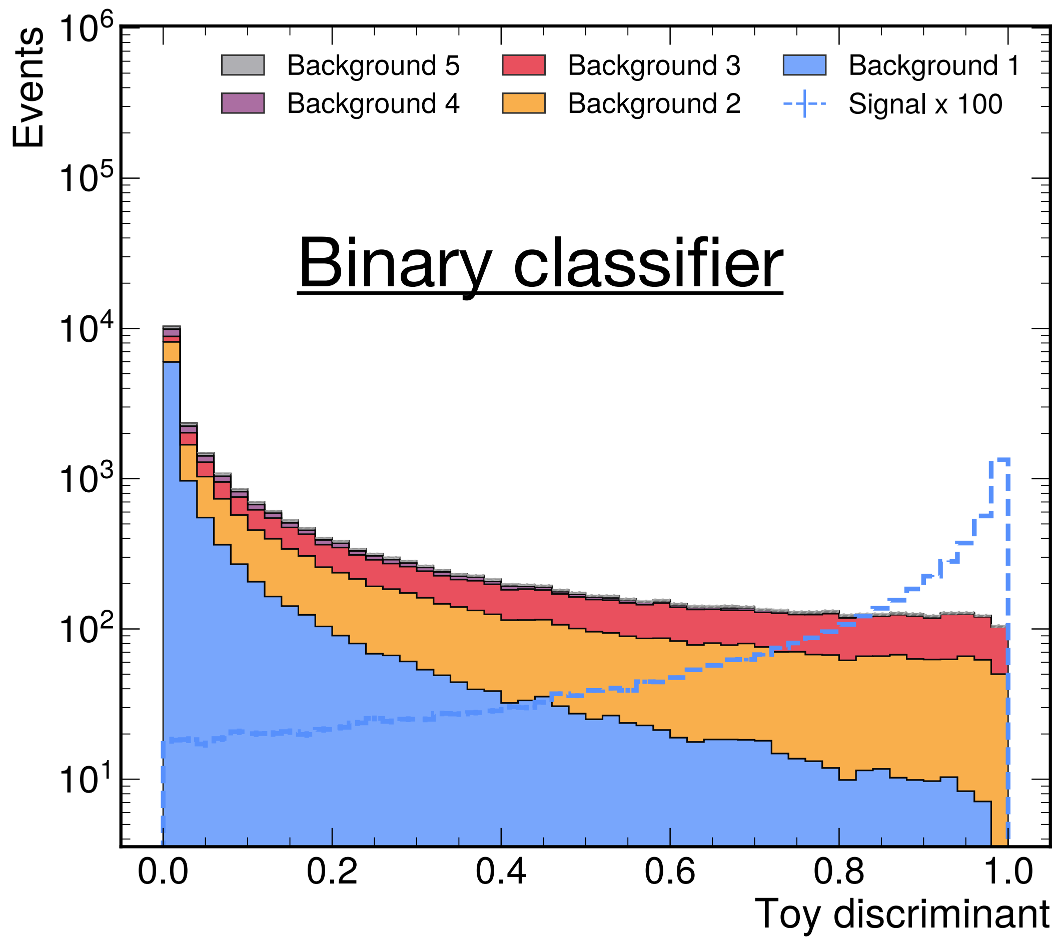
Backup Slides

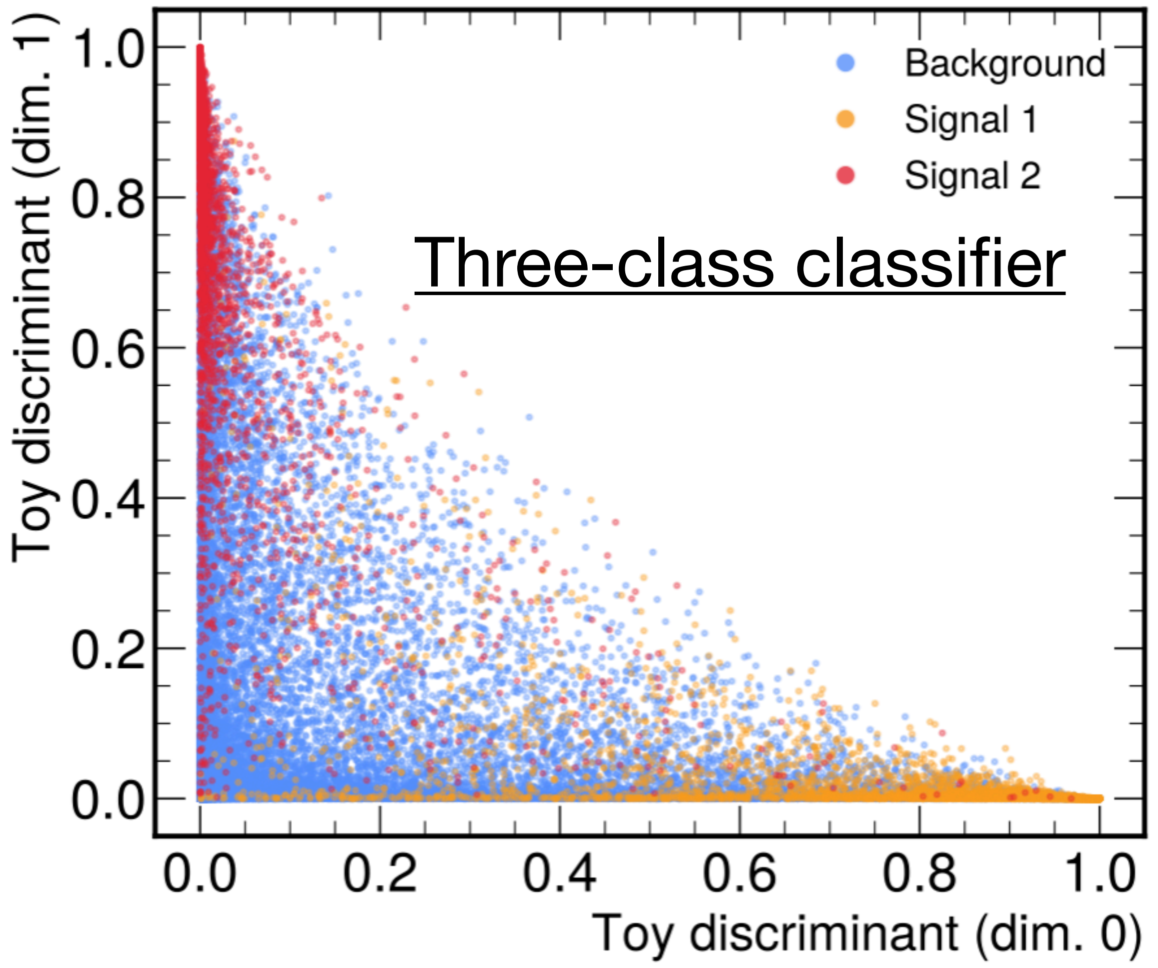
Setup

- Each physics process i (signals and backgrounds) is represented as three-dimensional feature vector
- Toy events are sampled from multivariate normal distribution
$$\mathcal{N}(\mu_i, \Sigma)$$
→ Each process has specific means μ_i and a common covariance matrix
- The total background likelihood for an events is:
$$p_{\text{bkg.}}(X) = \sum_{i \in \{\text{bkg. 1},\ldots,\text{bkg. 5}\}} \frac{\sigma_i}{\sum_{j \in \{\text{bkg. 1},\ldots,\text{bkg. 5}\}} \sigma_j} \mathcal{N}(X \mid \mu_i, \Sigma)$$
- For 1D:
$$L(X) = \frac{p_{\text{sig. 1}}(X)}{p_{\text{bkg.}}(X)}, \quad D(X) = \frac{L(X)}{1 + L(X)} \in [0,1]$$
- For 3D:
$$S_k(X) = \frac{p_k(X)}{p_{\text{sig. 1}}(X) + p_{\text{sig. 2}}(X) + p_{\text{bkg.}}(X)},$$

	Scenario 1			Scenario 2			
Process	$\mu_x^{(1)}$	$\mu_y^{(1)}$	$\mu_z^{(1)}$	$\mu_x^{(2)}$	$\mu_y^{(2)}$	$\mu_z^{(2)}$	σ [pb]
Signal 1	0.4	−0.4	1.0	0.9	−0.9	1.0	0.5
Signal 2	−0.4	0.4	1.0	−0.9	0.9	1.0	0.1
Background 1	0.0	0.0	−0.5	0.2	0.2	0.5	100.0
Background 2	−0.2	0.0	0.0	–	–	–	80.0
Background 3	0.1	0.4	−0.3	–	–	–	50.0
Background 4	−0.1	0.6	−0.2	–	–	–	20.0
Background 5	−0.2	0.1	−0.1	–	–	–	10.0

$$\Sigma = \text{diag}(1,1,1) + 0.2\,(\mathbf{1} - \text{diag}(1,1,1))$$



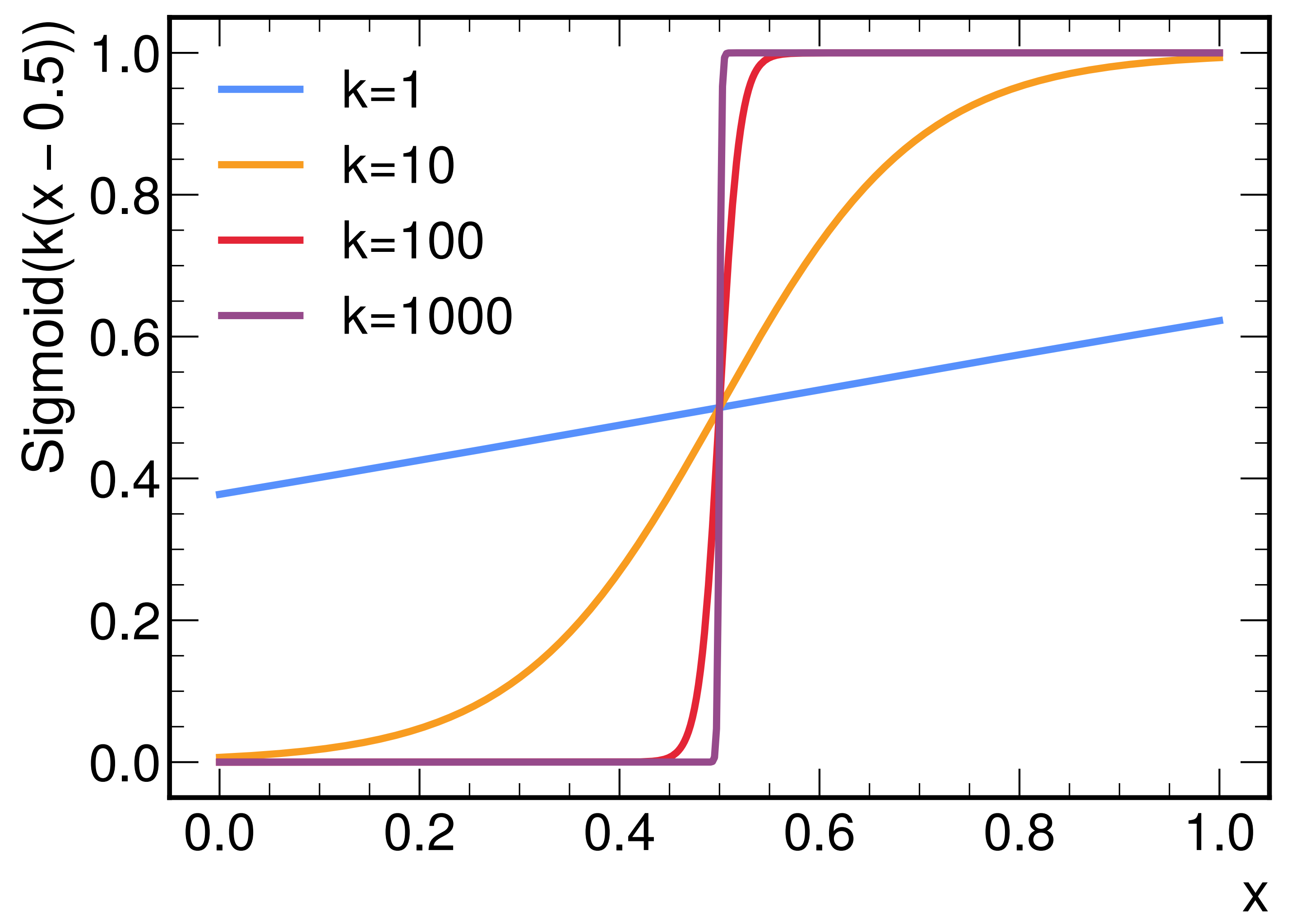


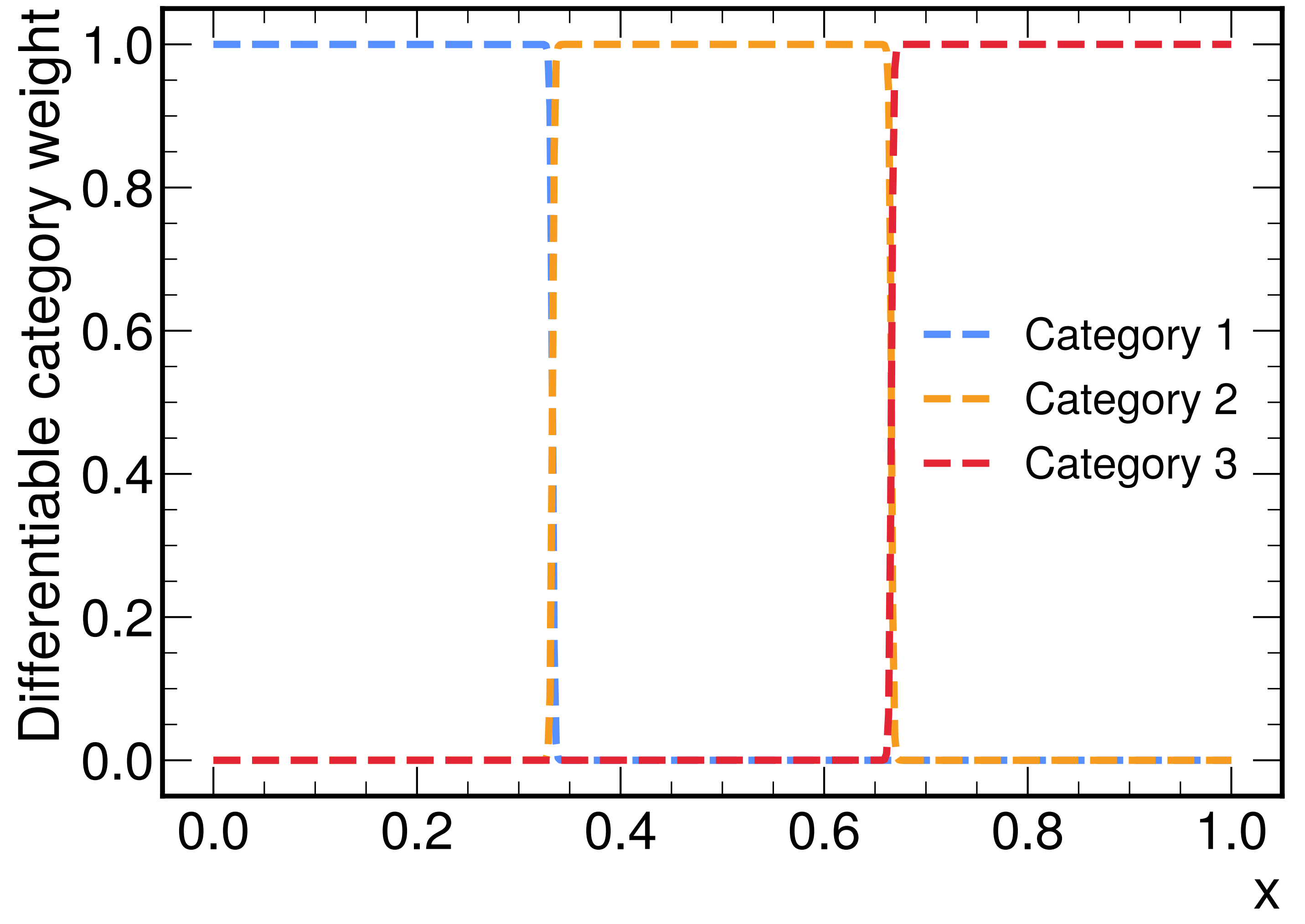
Method: one-dimensional case

- Straightforward to replace non-differentiable cuts with sigmoids
- Multiple sigmoid functions can be combined to calculate weight for each bin
  → Event with observable x and N bins (boundaries $e_{1,\ldots,N-1}$), the category membership weights are defined as:
  $$w_i(x) = \sigma\big(k(x - e_{i-1})\big) - \sigma\big(k(x - e_i)\big)$$

```
class gatohep.models.gato_sigmoid_model(variables_config, *, global_steepness=5.0, name='gato_sigmoid_model')
```

- RMSprop is used for optimising the trainable parameters





Method: multi-dimensional binning

- In multi-dimensional categorisation problems (e.g. multi-classifier output or multiple 1D observables), more flexible solutions can be obtained using a **Gaussian Mixture Model (GMM)**
  → Cuts are not restricted to be rectangular

  - Define N **Gaussians** in d-dimensional space, one per bin k:

$$\ell_k(x) = \mathcal{N}\big(x \mid \boldsymbol{\mu}_k, \Sigma_k\big).$$

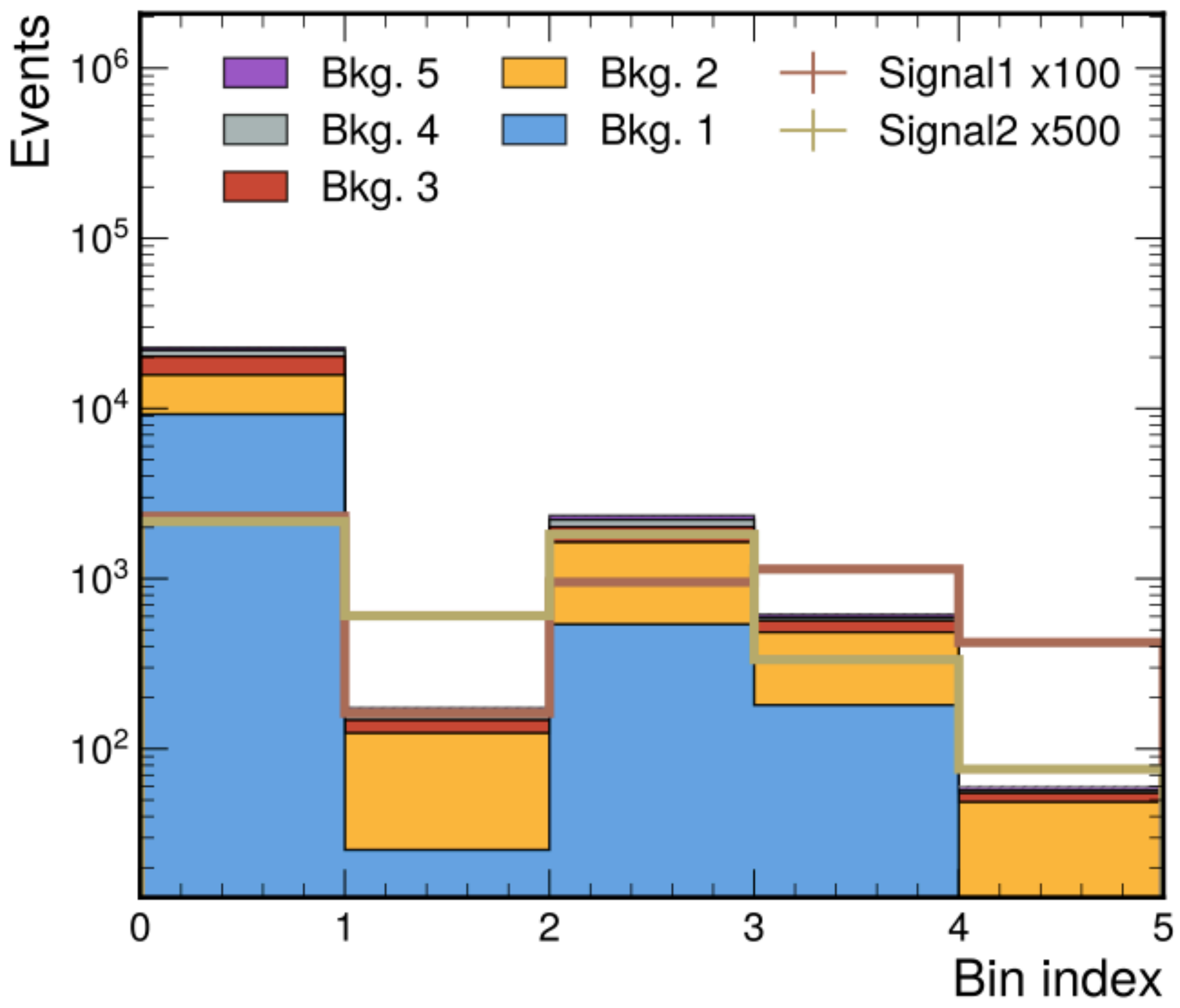
```
class gatohep.models.gato_gmm_model(n_cats, dim, temperature=1.0, mean_norm='softmax', mean_range=(0.0, 1.0), name='gato_gmm_model')
```

  - The binning is performed such that an event is assigned to the component with the largest score $s_k(x)$

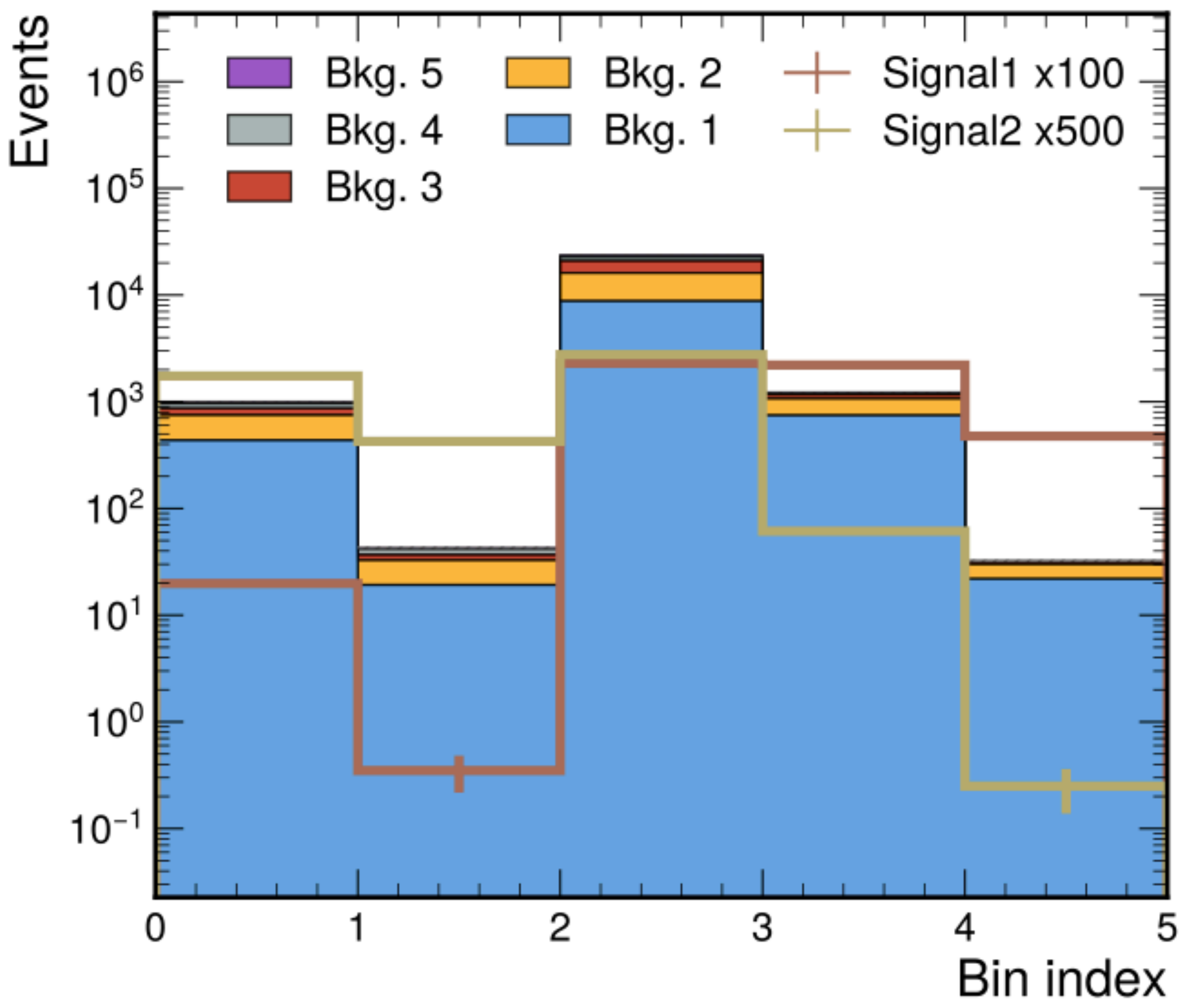
$$s_k(x) = \log \ell_k(x) + \log \pi_k.$$

    → Learnable mean: parameters restricted to user's data range using activation functions
    → Learnable covariance: Cholesky factorisation $\Sigma_i = L_i L_i^T$, where L_i is a lower-triangular matrix with positive diagonal, obtained using exponential of unconstrained learnable parameters
    → Learnable mixture weight: π_k

Optimised bins for three dimensional case



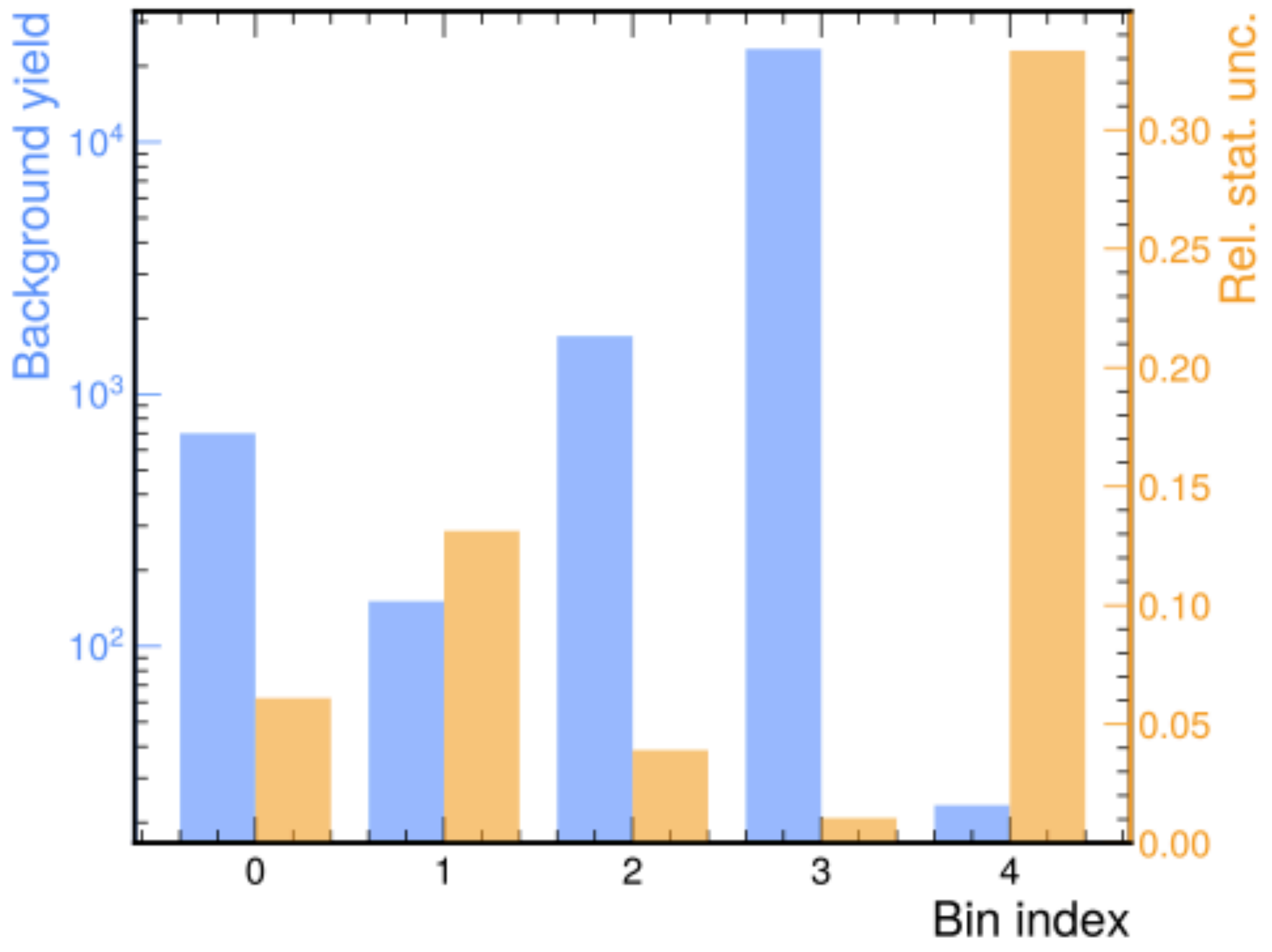
(e) First scenario



(f) Second scenario

Penalty on relative background uncertainty

Relative background uncertainty of 50%



Relative background uncertainty of 10%