Some Applications of Normalizing Flows to LHC and Gaia

ML at GGI Workshop

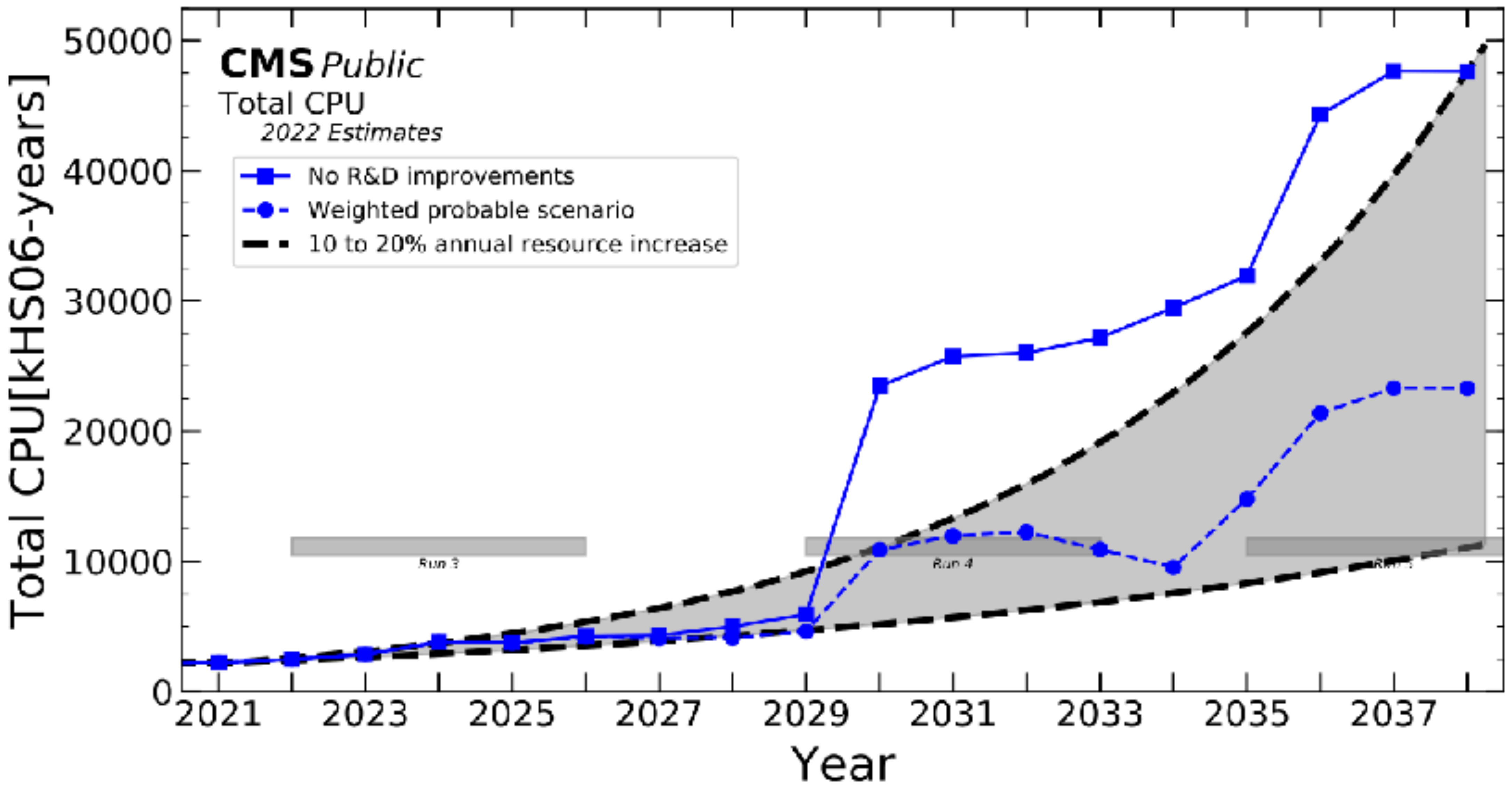
David Shih
September 12, 2022

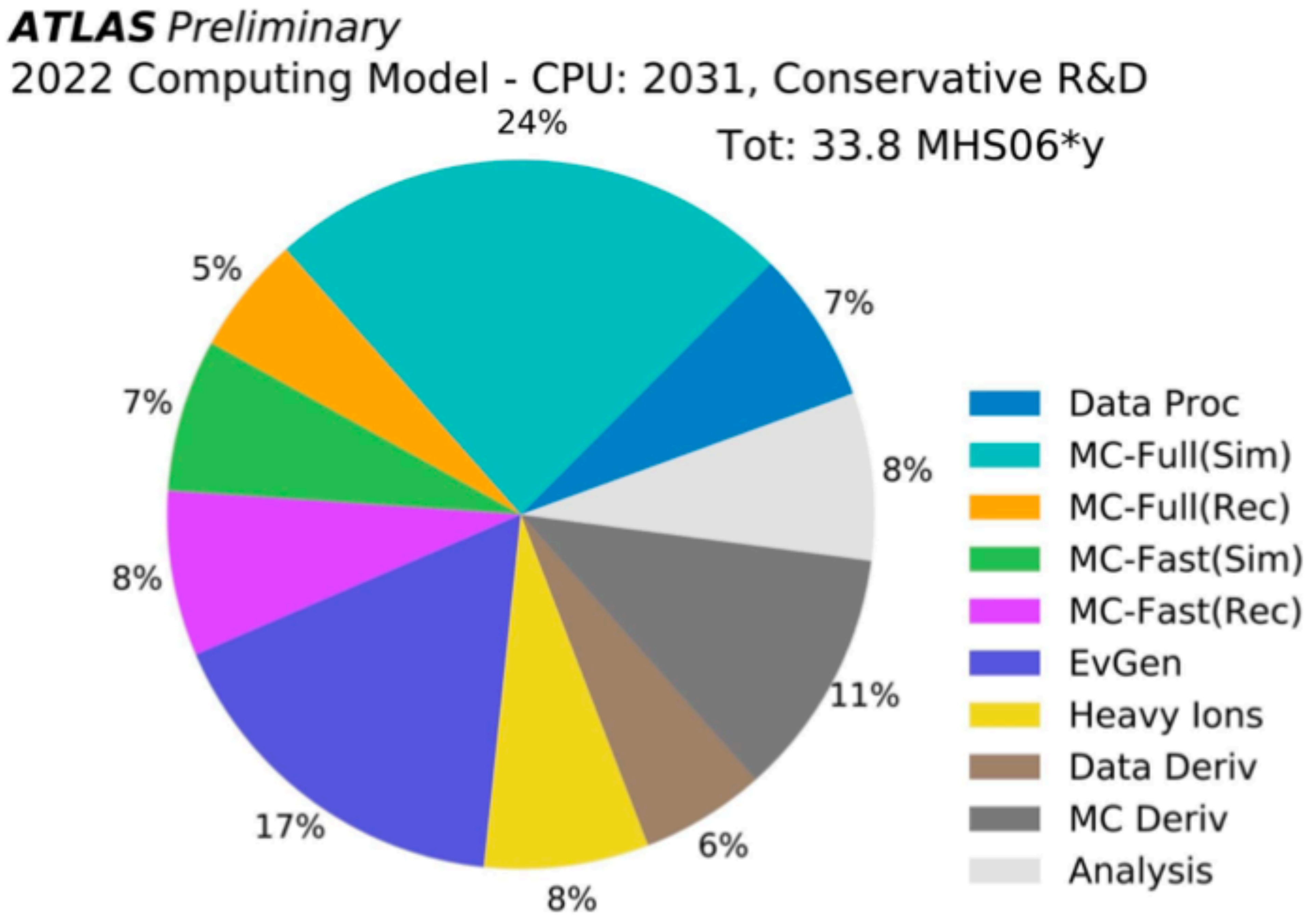
RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY

LHC: Generative Modeling: CaloFlow

Fast ML for Surrogate Modeling

https://twiki.cern.ch/twiki/bin/view/CMSPublic/CMSOfflineComputingResults





CERN-LHCC-2022-005

Detector simulation (GEANT4) and event generation (MG5, Pythia, Herwig, …) are major — and growing — bottlenecks at LHC and other experiments

Fast ML for Surrogate Modeling

GEANT4 → 10^{10} events **SLOW but ACCURATE**

Fast ML for Surrogate Modeling

GEANT4 → 10^{10} events **SLOW but ACCURATE**

GEANT4 → 10^{5} events

Fast ML for Surrogate Modeling

GEANT4 → 10^{10} events **SLOW but ACCURATE**

GEANT4 → 10^{5} events → Surrogate model

(GAN, VAE, Normalizing Flow, …)
Learn underlying distribution of GEANT4 events

Fast ML for Surrogate Modeling

GEANT4 → 10^{10} events **SLOW but ACCURATE**

GEANT4 → 10^5 events → Surrogate model → 10^{10} events

(GAN, VAE, Normalizing Flow, …)
Learn underlying distribution of GEANT4 events

Fast ML for Surrogate Modeling

GEANT4 → 10^{10} events **SLOW but ACCURATE**

GEANT4 → 10^{5} events → Surrogate model → 10^{10} events

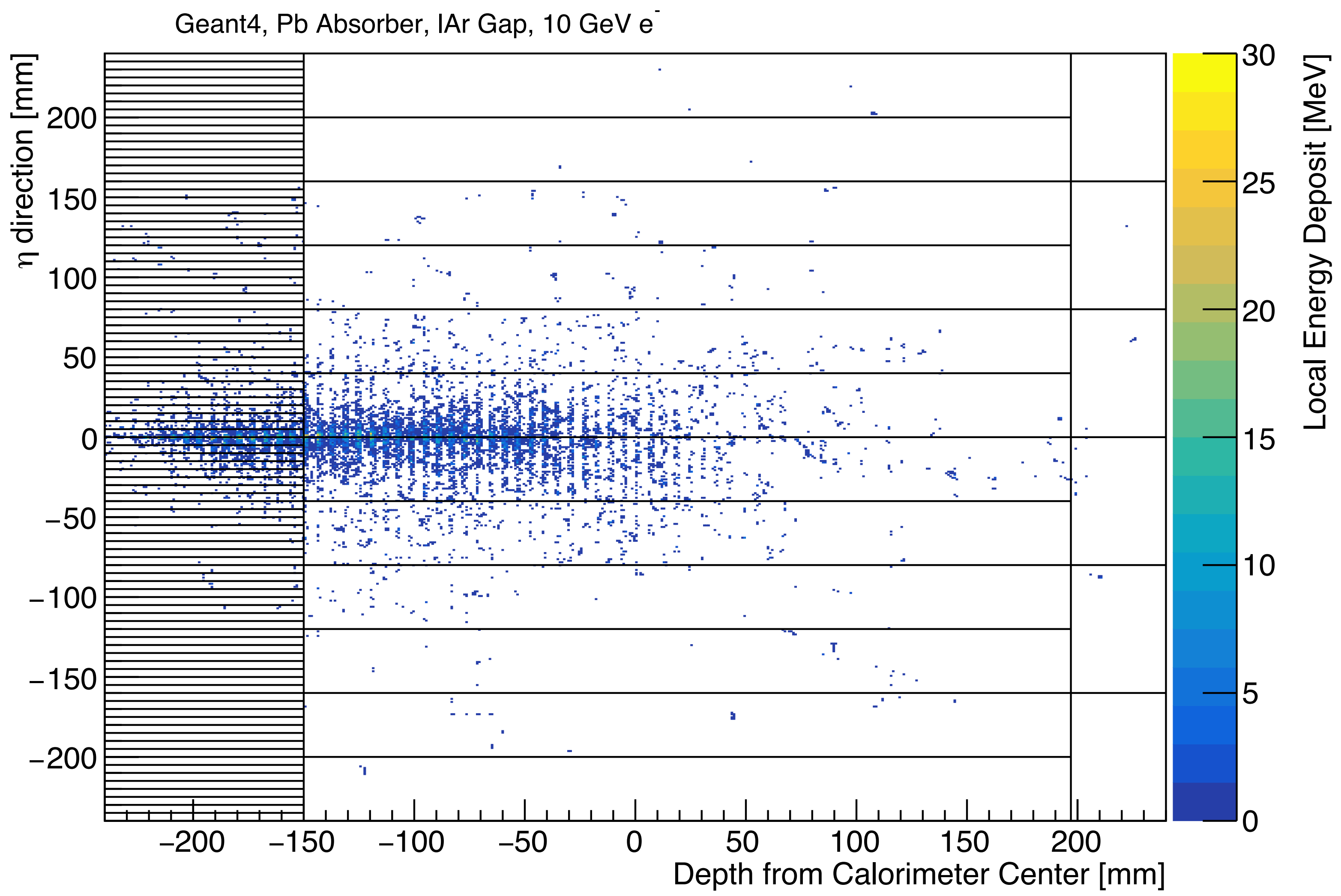
(GAN, VAE, Normalizing Flow, …)
Learn underlying distribution of GEANT4 events

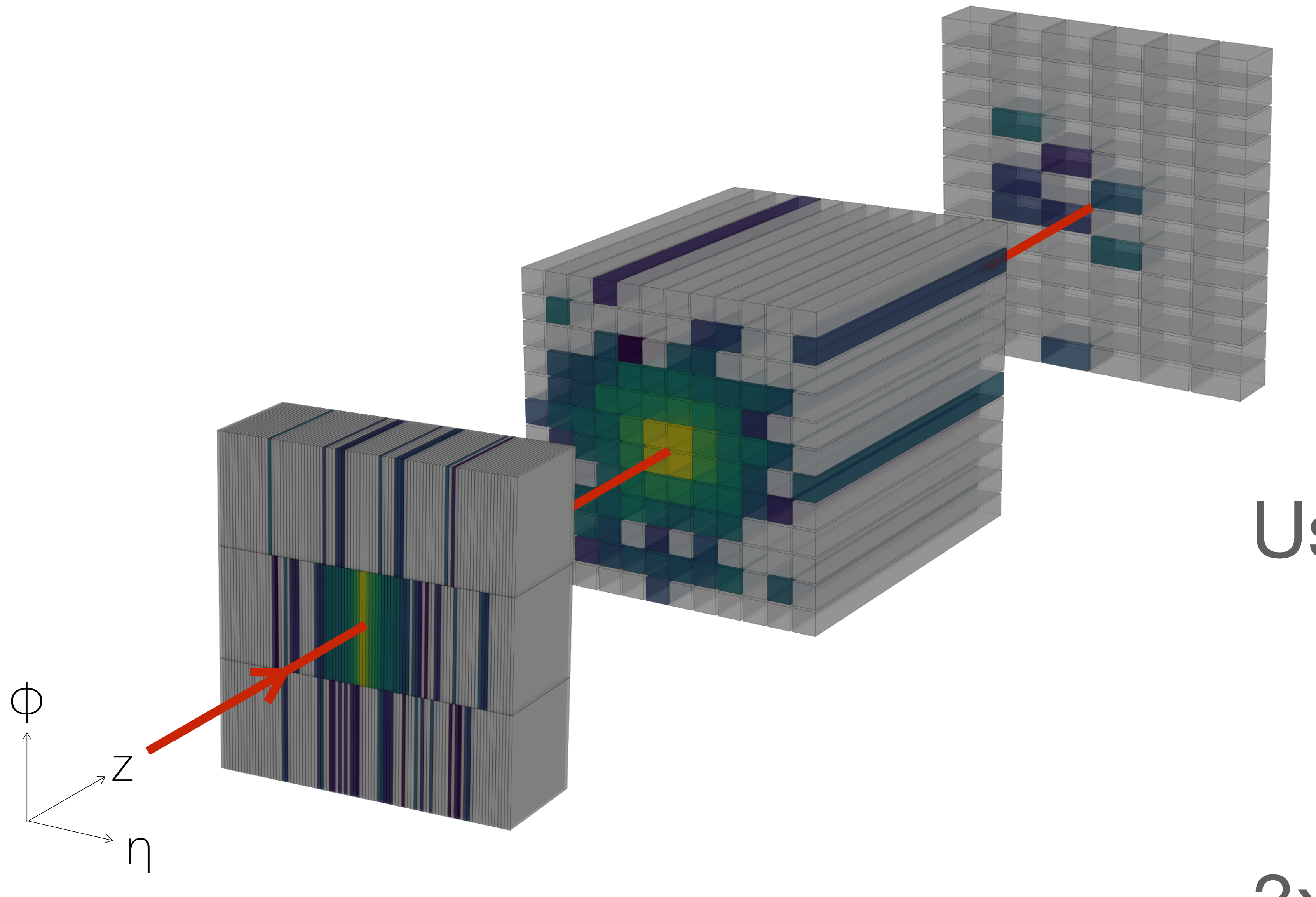
FAST and ACCURATE?

CaloFlow

Krause & DS 2106.05285, 2110.11377

We showed that normalizing flows offer impressive performance gains over previous state of the art (GANs) in surrogate modeling of GEANT4 calorimeter showers.





Used toy calorimeter based on ATLAS ECAL

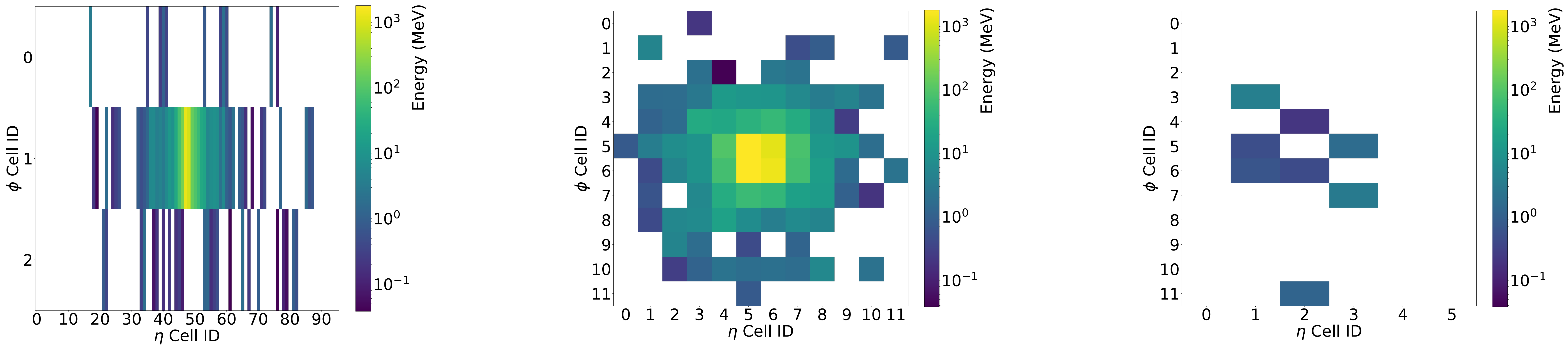
3 layers:
3x96+12x12+12x6=504 voxels

CaloGAN: Paganini, de Oliveira, Nachman [1705.02355, PRL; 1712.10321, PRD]

CaloFlow

Krause & DS 2106.05285, 2110.11377

- e^+, γ, π^+ showers (100k each)
- incident E uniformly sampled 1-100 GeV
- goal: train generative model on the showers and learn $p(\overrightarrow{E}_{voxels} \,|\, E_{inc})$



CaloGAN: Paganini, de Oliveira, Nachman [1705.02355, PRL; 1712.10321, PRD]

CaloFlow I

Krause & DS 2106.05285

MAF with RQS transformations

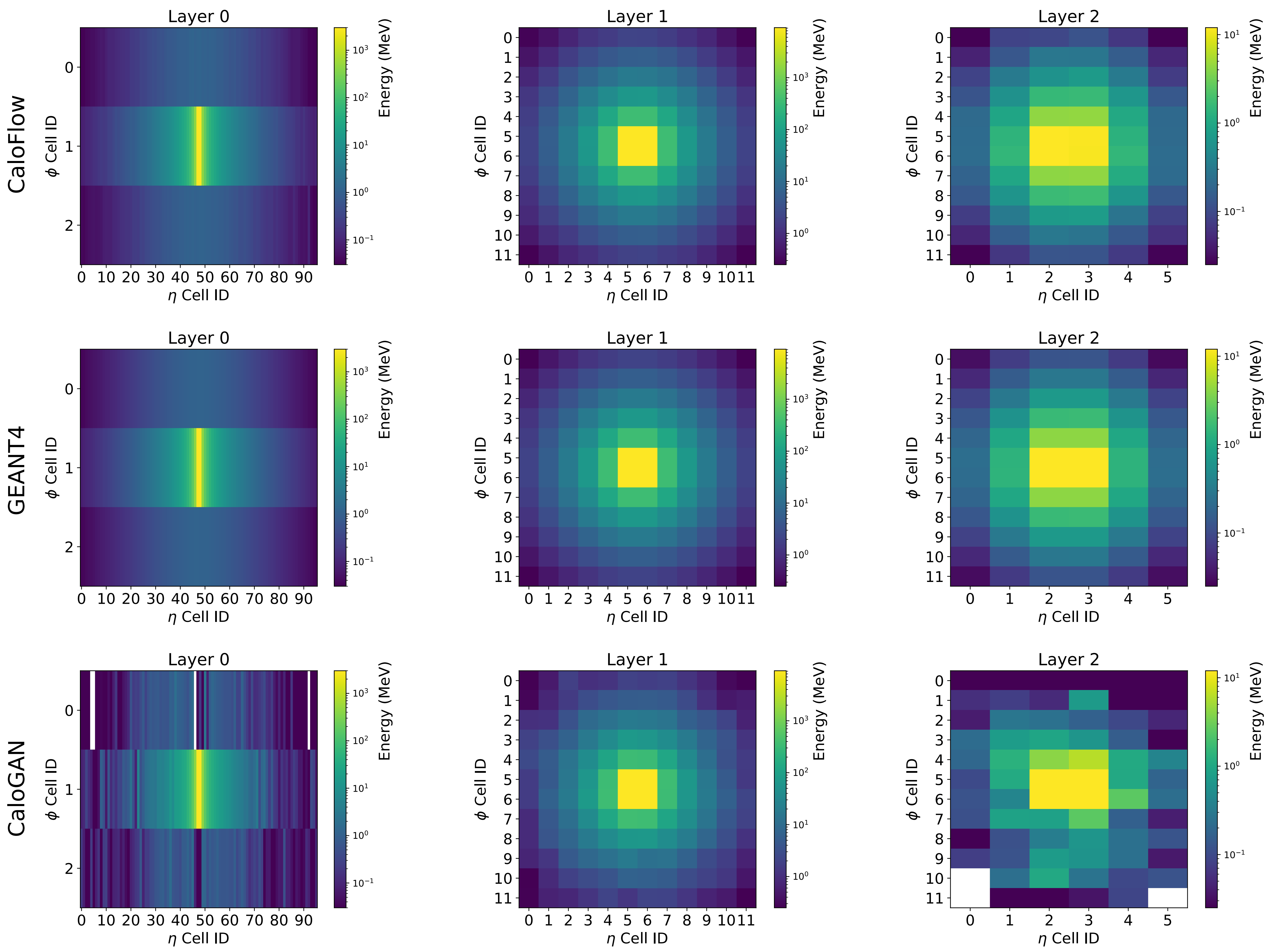
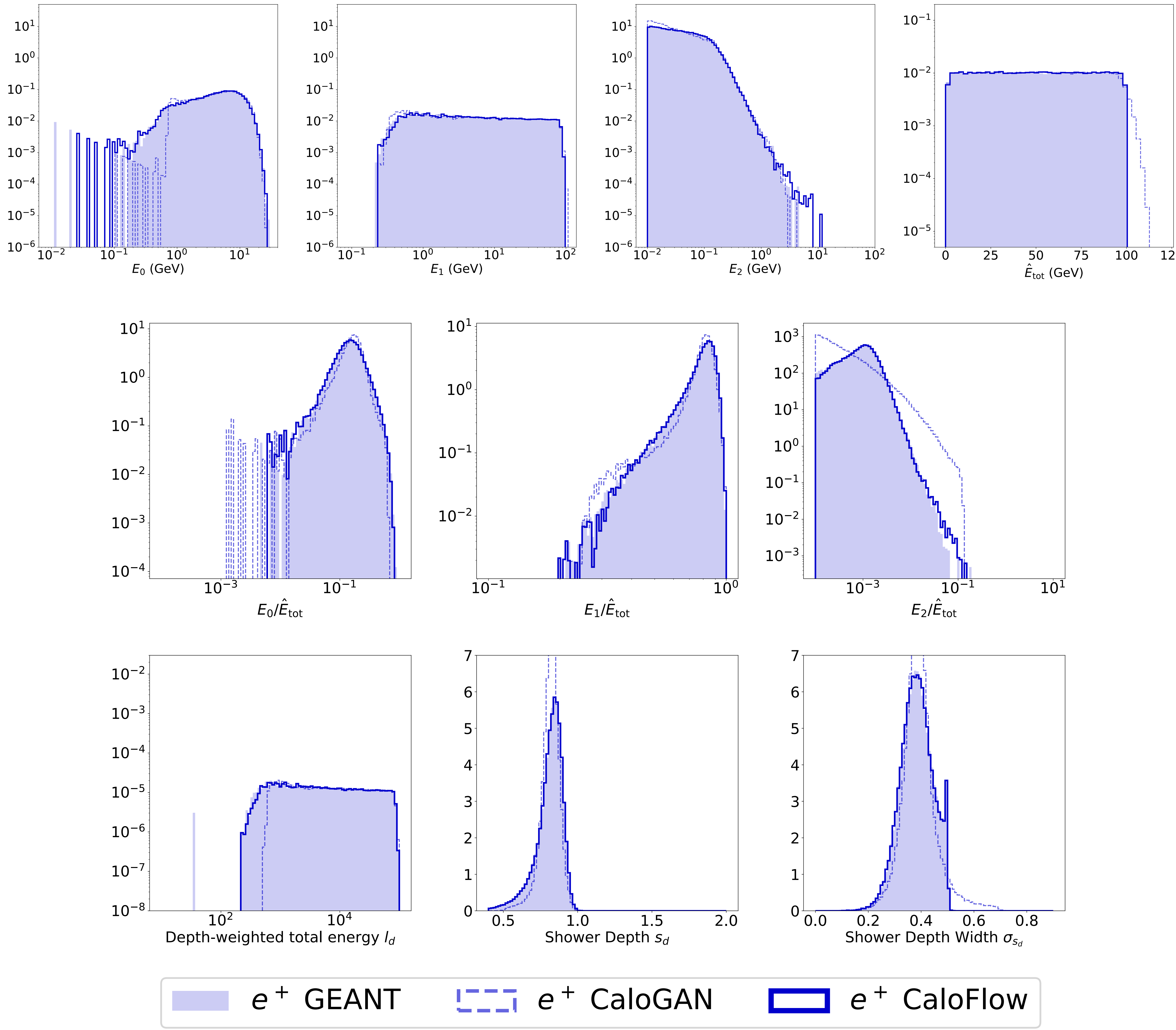


Figure 5. Average shower shapes for e^+. Columns are calorimeter layers 0 to 2, top row shows CaloFlow, center row Geant4, and bottom row CaloGAN



Performance vs GAN much improved!

CaloFlow I

Krause & DS 2106.05285

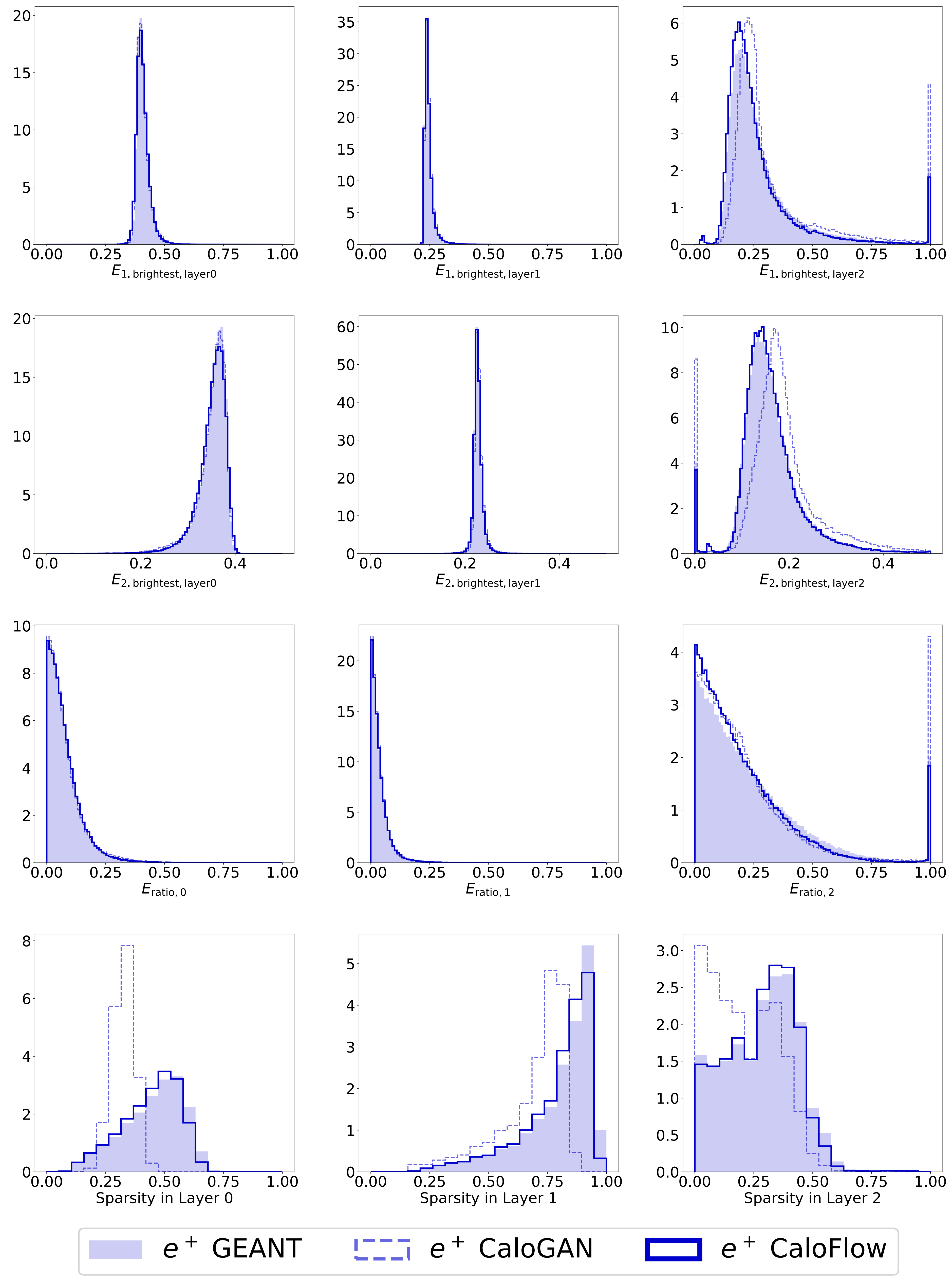
MAF with RQS transformations

AUC	GEANT4 vs. CaloGAN	GEANT4 vs. CaloFlow
e^+	1.000(0)	0.847(8)
γ	1.000(0)	0.660(6)
π^+	1.000(0)	0.632(2)

First to ever pass the “ultimate classifier metric” test:

DNN binary classifier, generated vs reference samples

Perfect generative model $p_{gen}(x) = p_{ref}(x)$ => classifier AUC=0.5



CaloFlow I

Krause & DS 2106.05285

MAF with RQS transformations

Table 4. Generation time of a single calorimeter shower in ms. Times were obtained on a Titan V GPU. Geant4 needs 1772 ms per shower [8].

	CaloGAN		CaloFlow
batch size	batch size requested	100k requested	
10	455	2.2	835
100	45.5	0.3	96.1
1000	4.6	0.08	41.4
5000	1.0	0.07	36.2
10000	0.5	0.07	36.2

Main drawback of MAF: sampling is slow

Accurate, but not fast!

CaloFlow II

IAF with RQS transformations

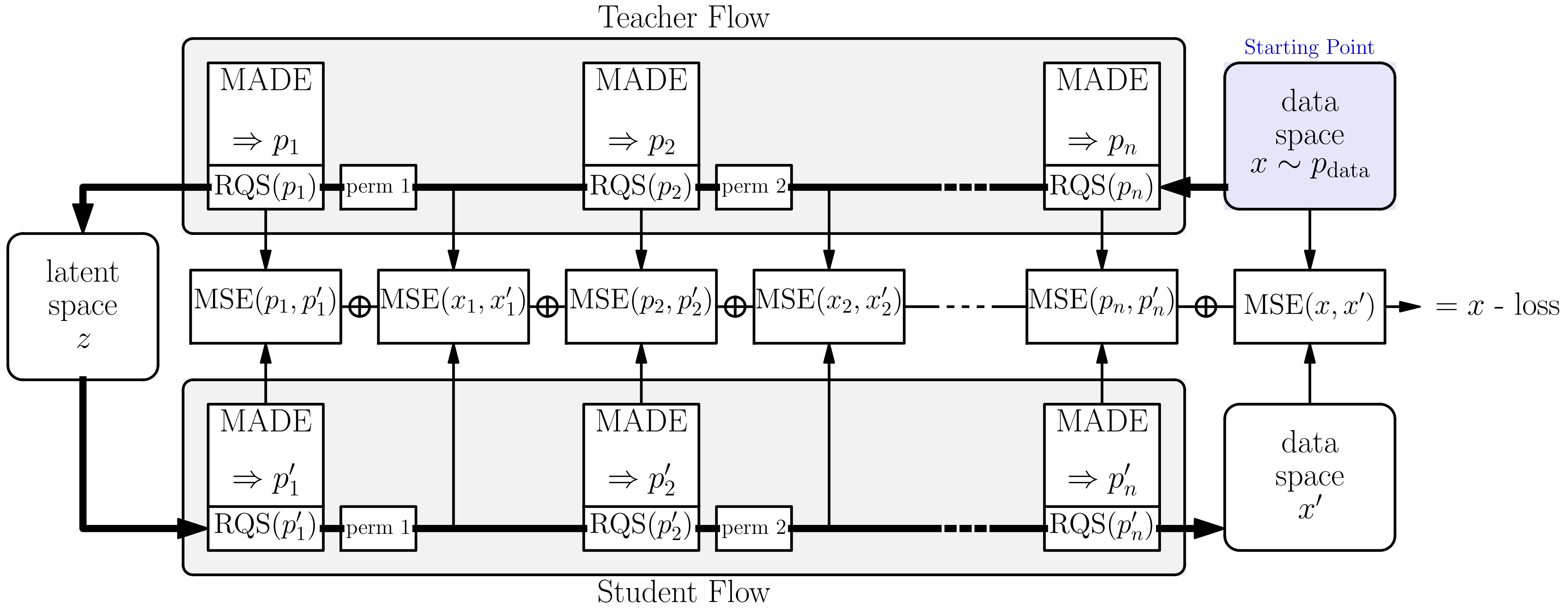
Krause & DS 2110.11377

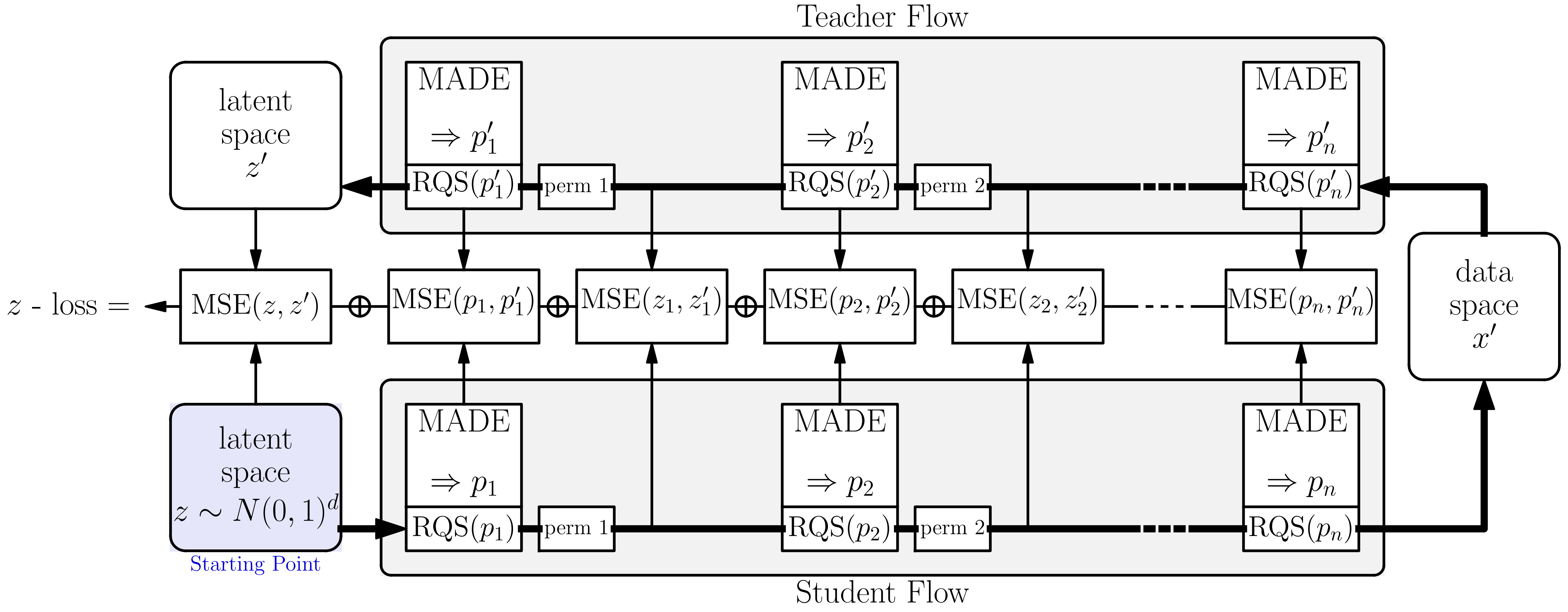
IAF cannot be trained with MLE loss.

Instead we developed a new teacher/student method to train it. Based on “Probability Density Distillation”.
van den Oord et al 1711.10433
Huang et al PMLR 2020

Idea: constrain IAF to be exact inverse of MAF, layer by layer and coupling by coupling

$$L = 0.5\left(\underbrace{\mathrm{MSE}(z,z') + \sum_i \mathrm{MSE}(z^{(i)}, z'^{(i)}) + \sum_i \mathrm{MSE}(p_z^{(i)}, p_z'^{(i)})}_{z\text{-loss}}\right)$$
$$+\,0.5\left(\underbrace{\mathrm{MSE}(x,x') + \sum_i \mathrm{MSE}(x^{(i)}, x'^{(i)}) + \sum_i \mathrm{MSE}(p_x^{(i)}, p_x'^{(i)})}_{x\text{-loss}}\right)$$





Training only involves fast directions of MAF and IAF

CaloFlow II

Krause & DS 2110.11377

IAF with RQS transformations

Similarly impressive accuracy

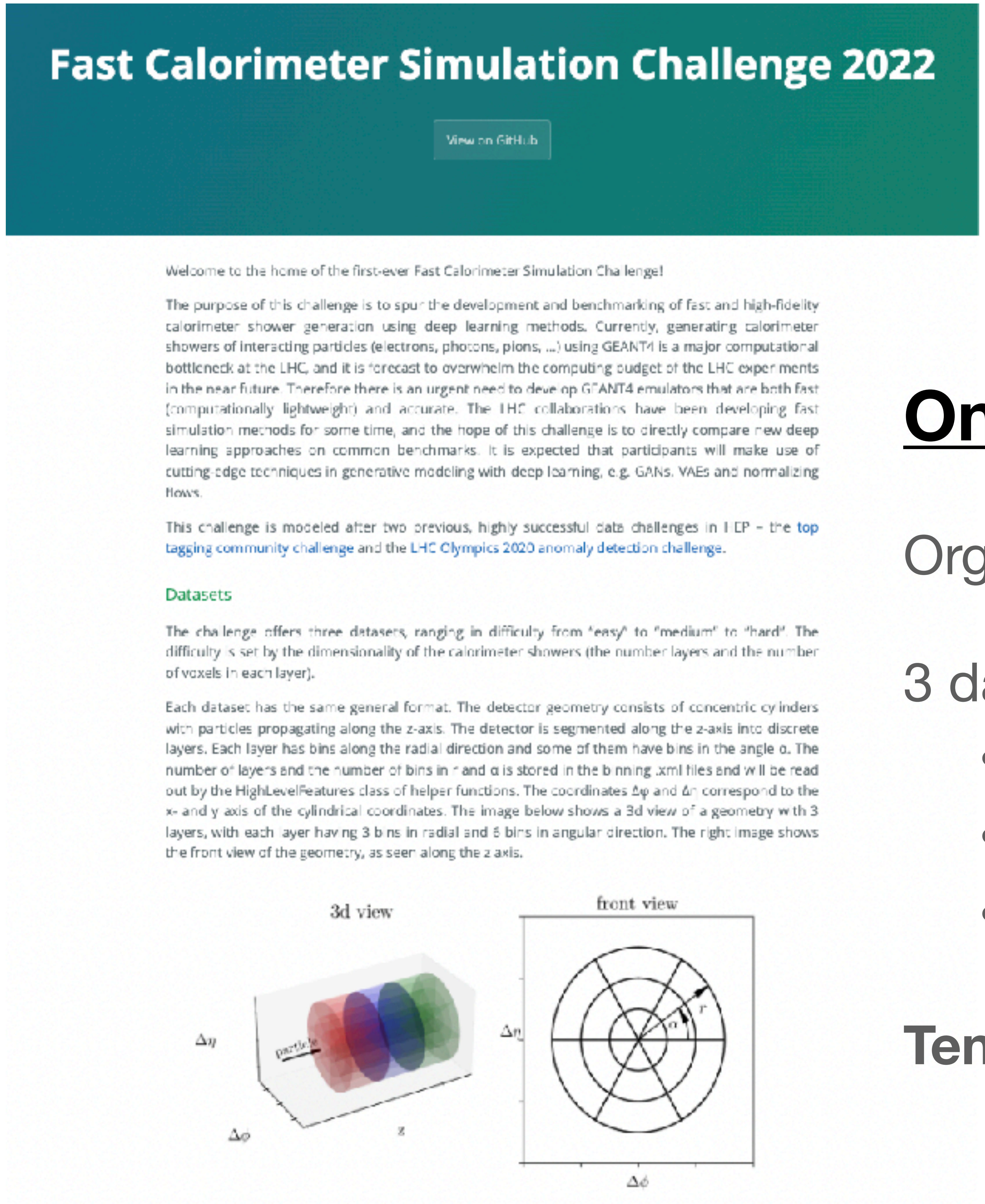
AUC / JSD		DNN: GEANT4 vs. CALOFLOW v2 (student)	DNN: GEANT4 vs. CALOFLOW v1 (teacher) [17]
e^+	unnormalized	0.785(7) / 0.200(10)	0.847(8) / 0.345(12)
	normalized	0.824(5) / 0.255(8)	0.869(2) / 0.376(4)
γ	unnormalized	0.761(14) / 0.167(18)	0.660(6) / 0.067(4)
	normalized	0.761(4) / 0.159(6)	0.794(4) / 0.213(7)
π^+	unnormalized	0.729(2) / 0.144(3)	0.632(2) / 0.048(1)
	normalized	0.807(2) / 0.231(4)	0.751(4) / 0.148(4)

And a factor of 500 faster,
on par with GAN!

Table 4. Training and evaluation times of CALOFLOW and CALOGAN.

	CALOFLOW		CALOGAN		GEANT4
	v1 (teacher) [17]	v2 (student)			
training	22+82 min	+ 480 min	210 min		0 min
generation batch size	time per shower		batch size req.	100k req.	
10	835 ms	5.81 ms	455 ms	2.2 ms	1772 ms
100	96.1 ms	0.60 ms	45.5 ms	0.3 ms	1772 ms
1000	41.4 ms	0.12 ms	4.6 ms	0.08 ms	1772 ms
10000	36.2 ms	**0.08 ms**	0.5 ms	**0.07 ms**	1772 ms

CaloChallenge 2022



Fast Calorimeter Simulation Challenge 2022

View on GitHub

Welcome to the home of the first-ever Fast Calorimeter Simulation Challenge!

The purpose of this challenge is to spur the development and benchmarking of fast and high-fidelity calorimeter shower generation using deep learning methods. Currently, generating calorimeter showers of interacting particles (electrons, photons, pions, ...) using GEANT4 is a major computational bottleneck at the LHC, and it is forecast to overwhelm the computing budget of the LHC experiments in the near future. Therefore there is an urgent need to develop GEANT4 emulators that are both fast (computationally lightweight) and accurate. The LHC collaborations have been developing fast simulation methods for some time, and the hope of this challenge is to directly compare new deep learning approaches on common benchmarks. It is expected that participants will make use of cutting-edge techniques in generative modeling with deep learning, e.g. GANs, VAEs and normalizing flows.

This challenge is modeled after two previous, highly successful data challenges in HEP – the top tagging community challenge and the LHC Olympics 2020 anomaly detection challenge.

Datasets

The challenge offers three datasets, ranging in difficulty from "easy" to "medium" to "hard". The difficulty is set by the dimensionality of the calorimeter showers (the number layers and the number of voxels in each layer).

Each dataset has the same general format. The detector geometry consists of concentric cylinders with particles propagating along the z-axis. The detector is segmented along the z-axis into discrete layers. Each layer has bins along the radial direction and some of them have bins in the angle α. The number of layers and the number of bins in r and α is stored in the binning .xml files and will be read out by the HighLevelFeatures class of helper functions. The coordinates Δφ and Δη correspond to the x- and y axis of the cylindrical coordinates. The image below shows a 3d view of a geometry with 3 layers, with each layer having 3 bins in radial and 6 bins in angular direction. The right image shows the front view of the geometry, as seen along the z axis.

<u>Ongoing data challenge for fast calorimeter simulation</u>

Organizers: Giannelli, Kasieczka, Krause, Nachman, Salamani, **DS**, Zaborowska

3 datasets:

- "easy" — official ATLAS CaloSim (~10^2 voxels)
- "medium" — GEANT4 example detector (~10^3 voxels)
- "hard" — GEANT4 example detector (~10^4 voxels)

Tentative deadline: ML4Jets2022@Rutgers in November

https://calochallenge.github.io/homepage/

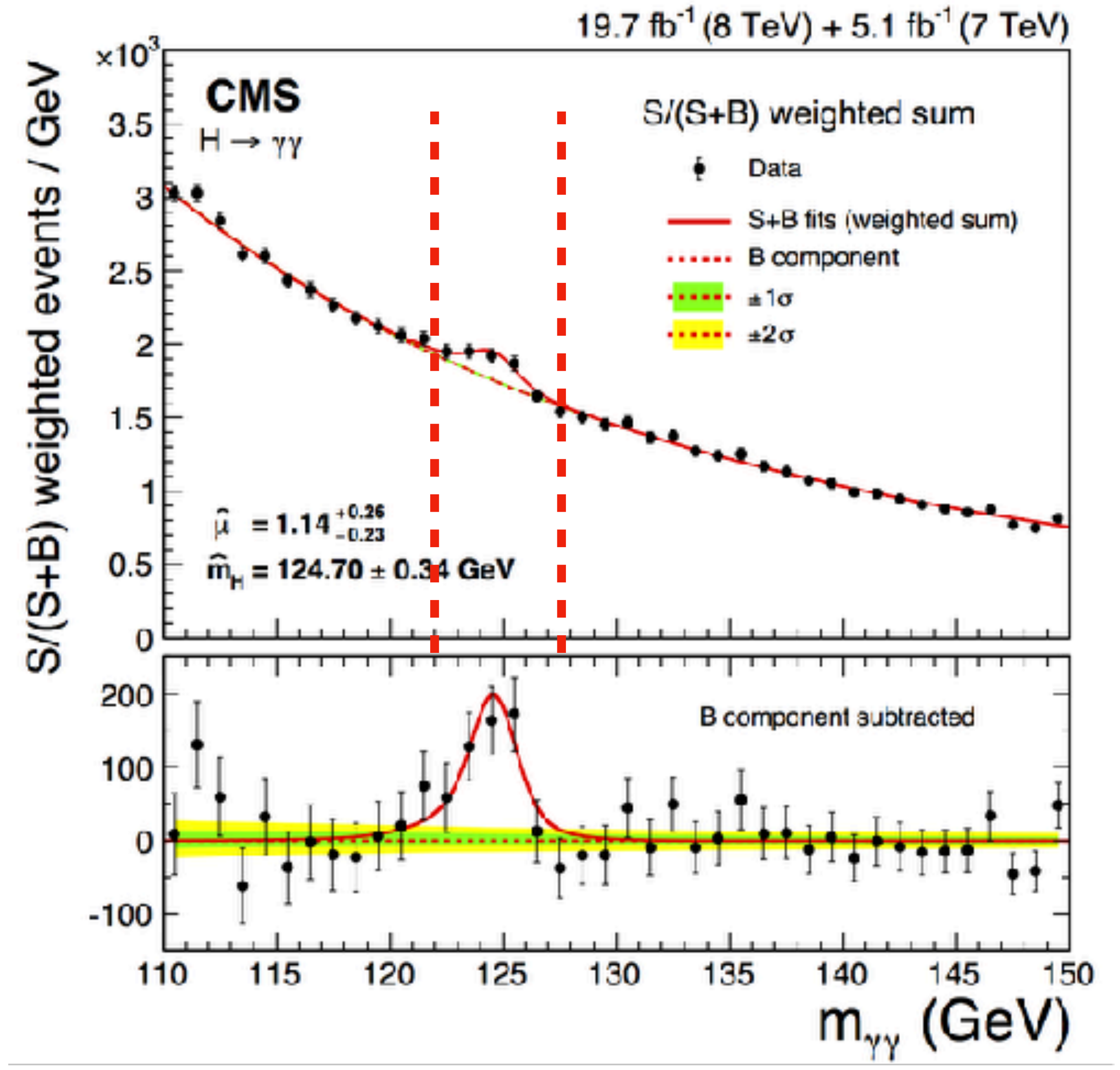
LHC: Anomaly Detection: ANODE and CATHODE

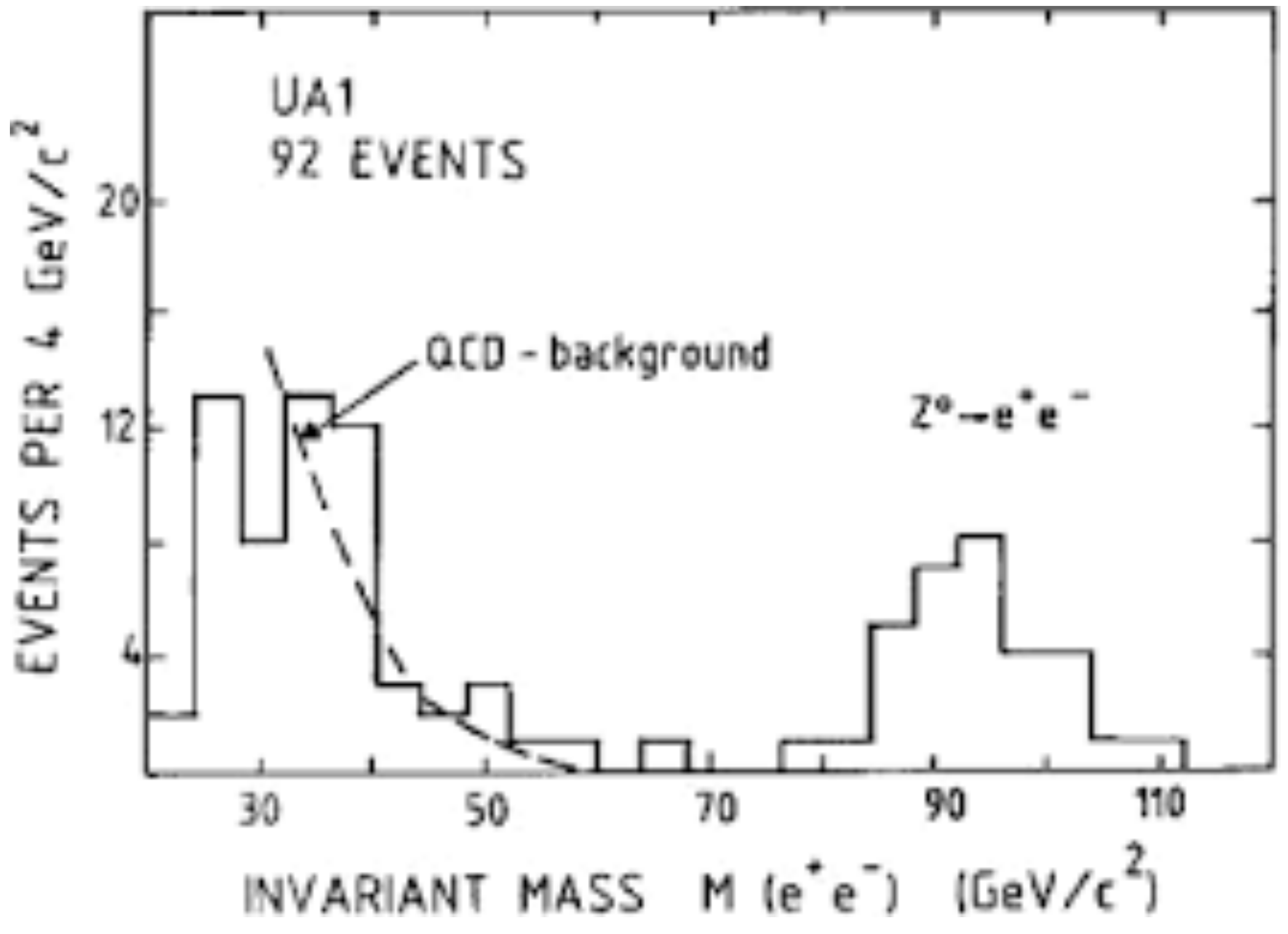
A Classic (Semi-)Model-Agnostic Search

1D Bump Hunt

Idea: assume signal is localized in some feature (usually invariant mass) while background is smooth.

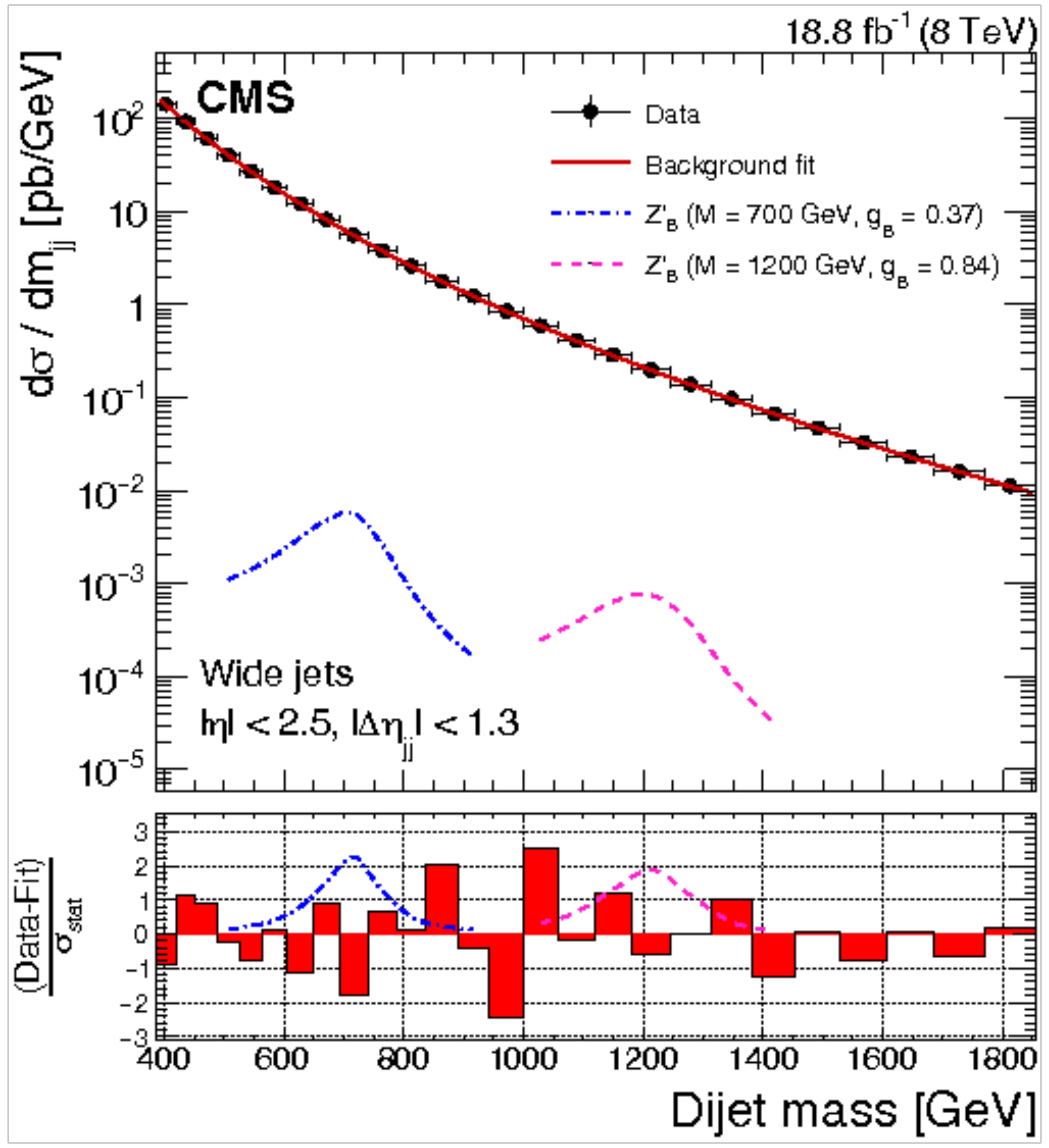
Interpolate from **sidebands** into **signal region** (eg window in invariant mass), search for an excess.





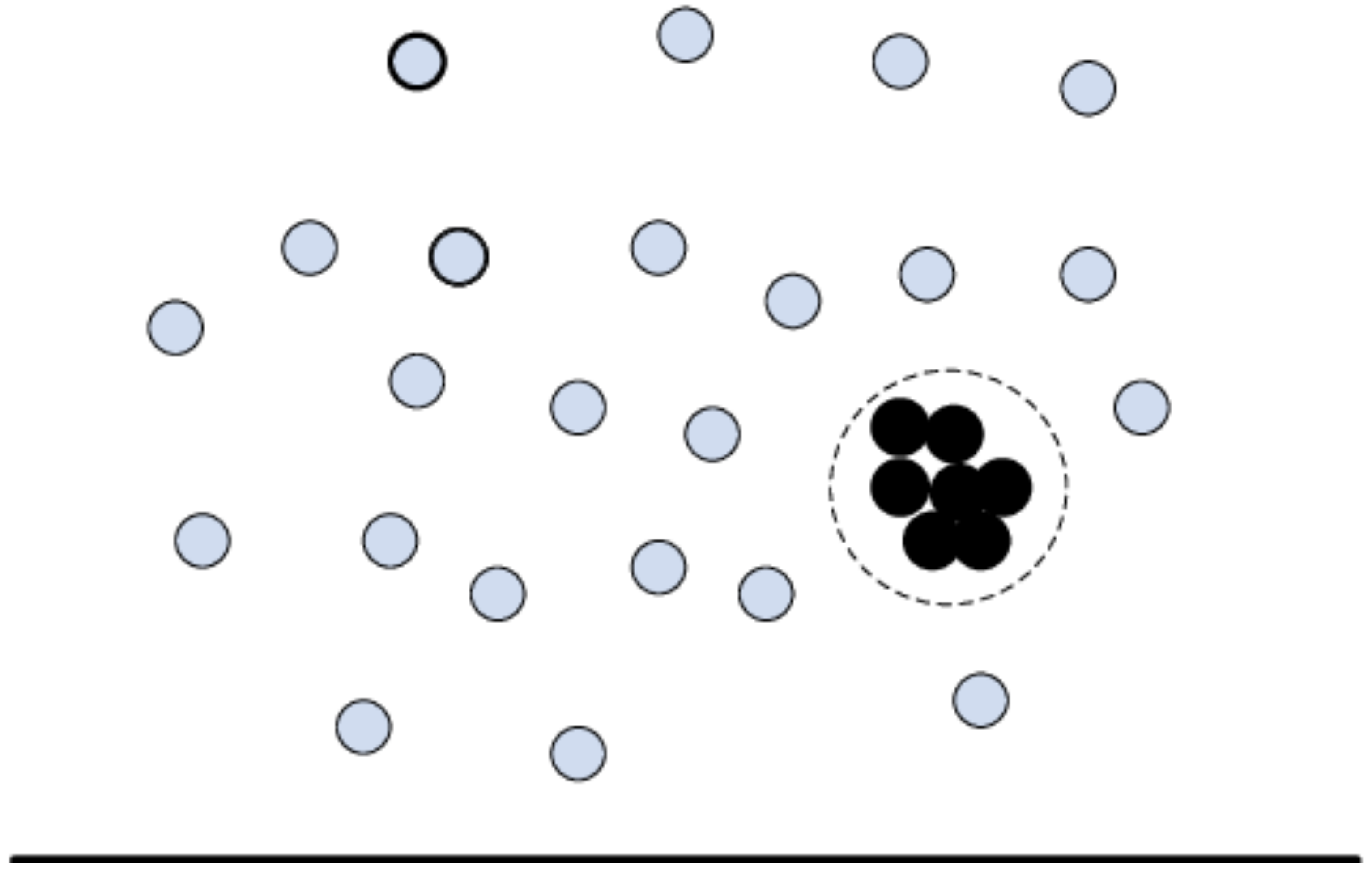
Used in many discoveries!

Enhancing the Bump Hunt



primary resonant feature (m_{JJ})

$\times$



$\vec{x} \in \mathbb{R}^d$

additional features

Q: If the signal is localized in additional features, can we find it in a model-independent way?

A general strategy

<u>Claim:</u> the optimal model-agnostic discriminant would be (Neyman & Pearson)

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

"Idealized Anomaly Detector"

A general strategy

Claim: the optimal model-agnostic discriminant would be (Neyman & Pearson)

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

"Idealized Anomaly Detector"

Proof:

$$p_{data}(x) = \epsilon_{sig} p_{sig}(x) + (1 - \epsilon_{sig}) p_{bg}(x)$$

$$R(x) = (1 - \epsilon_{sig}) + \epsilon_{sig} \frac{p_{sig}(x)}{p_{bg}(x)}$$

A general strategy

Claim: the optimal model-agnostic discriminant would be (Neyman & Pearson)

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

"Idealized Anomaly Detector"

Proof:

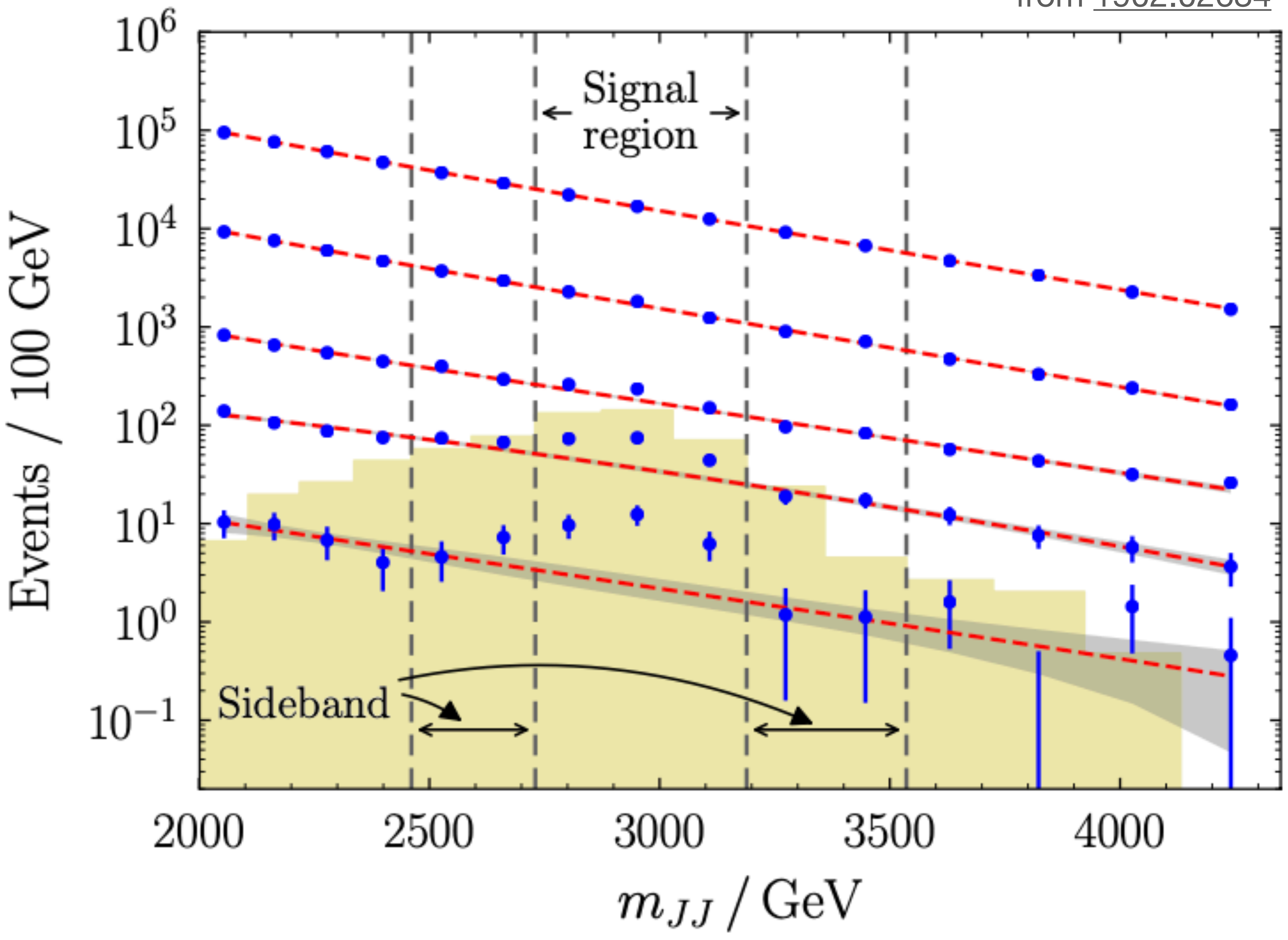
$$p_{data}(x) = \epsilon_{sig} p_{sig}(x) + (1 - \epsilon_{sig}) p_{bg}(x)$$

$$R(x) = (1 - \epsilon_{sig}) + \epsilon_{sig} \frac{p_{sig}(x)}{p_{bg}(x)}$$

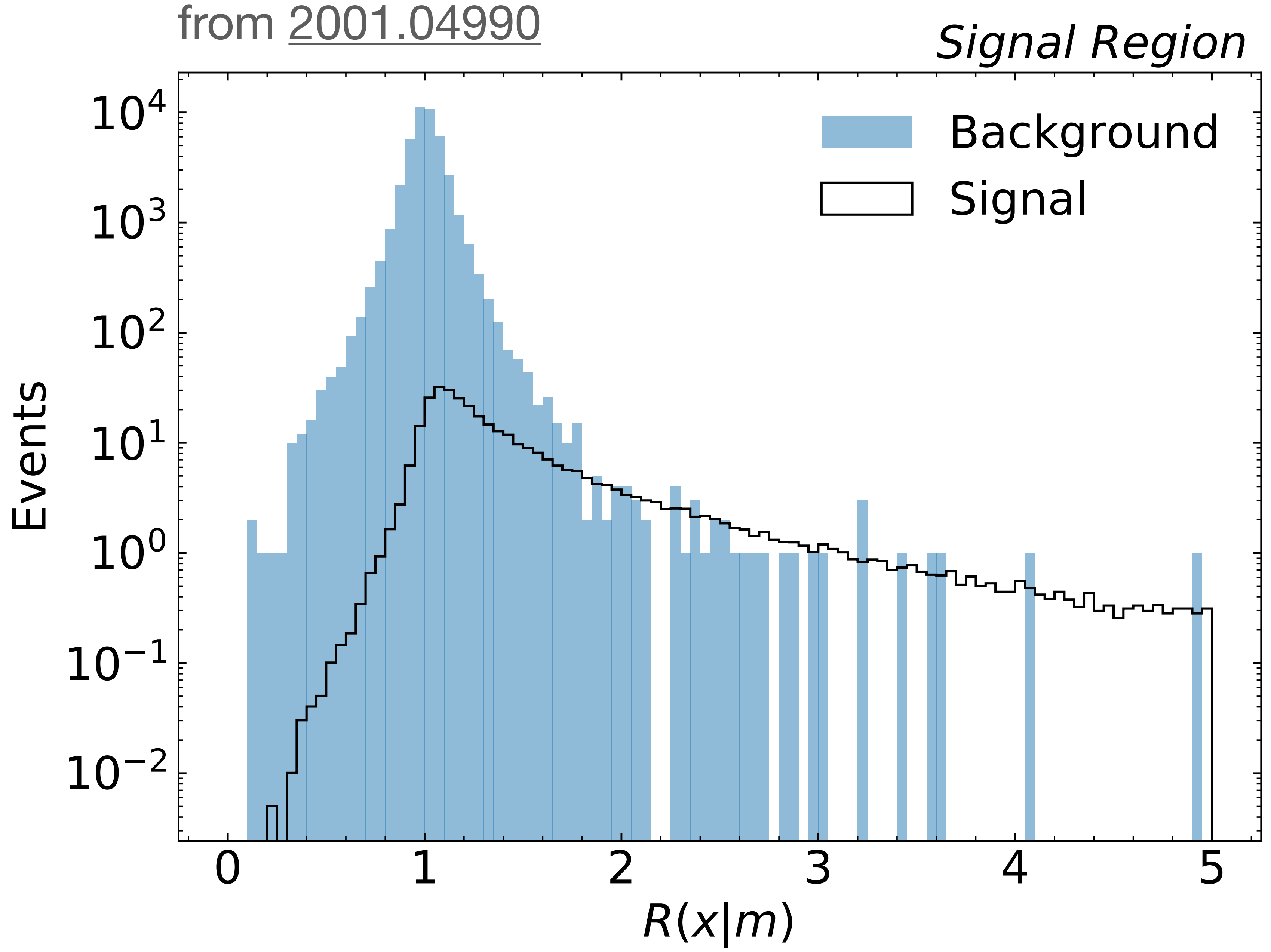
$R(x)$ is monotonic with signal-to-background likelihood ratio ***regardless of unknown, arbitrary signal strength and probability density***

A general strategy

from 1902.02634



from 2001.04990



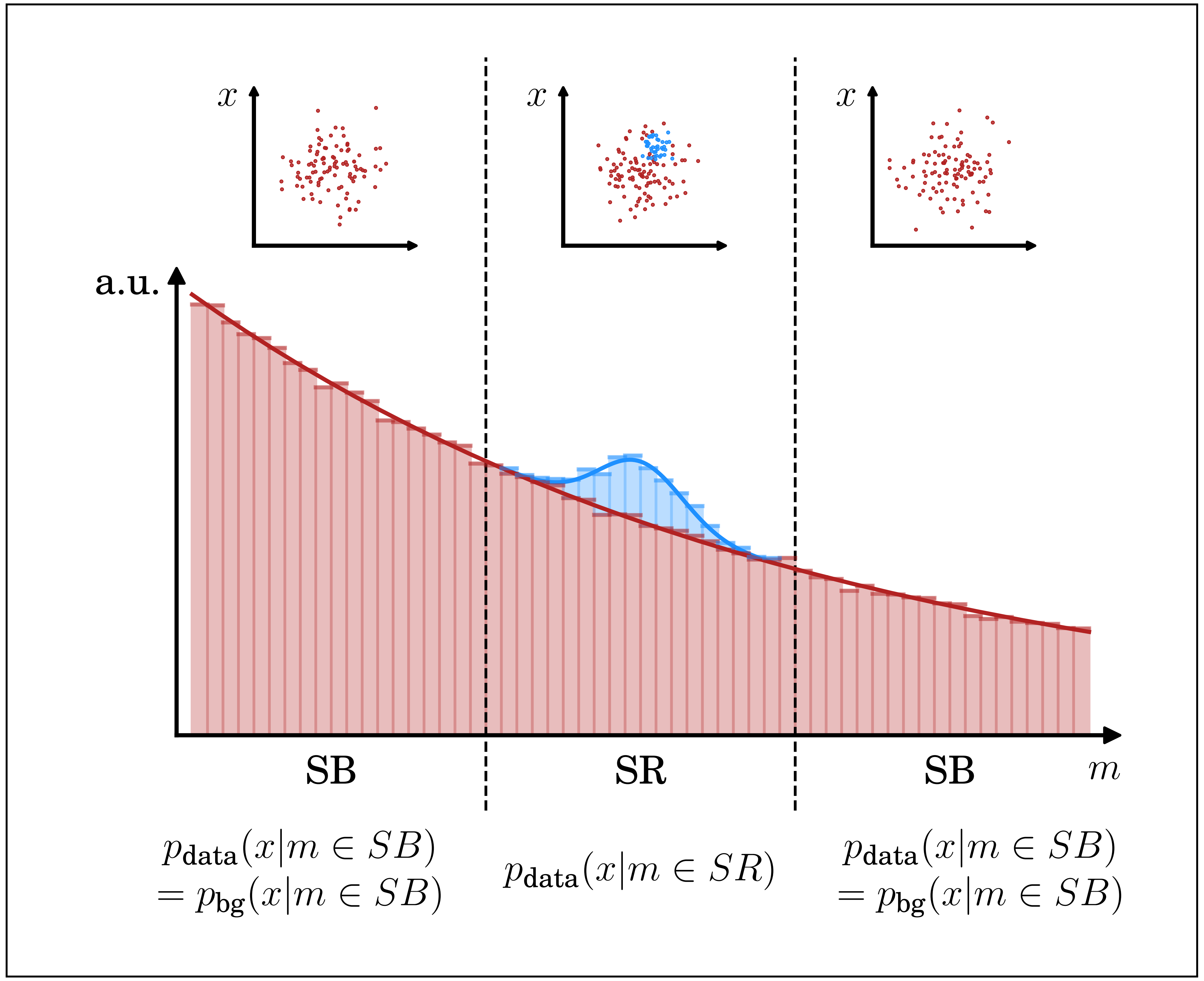
Cutting on $R(x) > R_c$ can greatly enhance the significance of the signal over the background.

Can combine with regular bump hunt in primary resonant feature m_{JJ} for data-driven background estimation.

Idea: data vs sideband classifier

Collins, Howe & Nachman 1805.02664,1902.02634

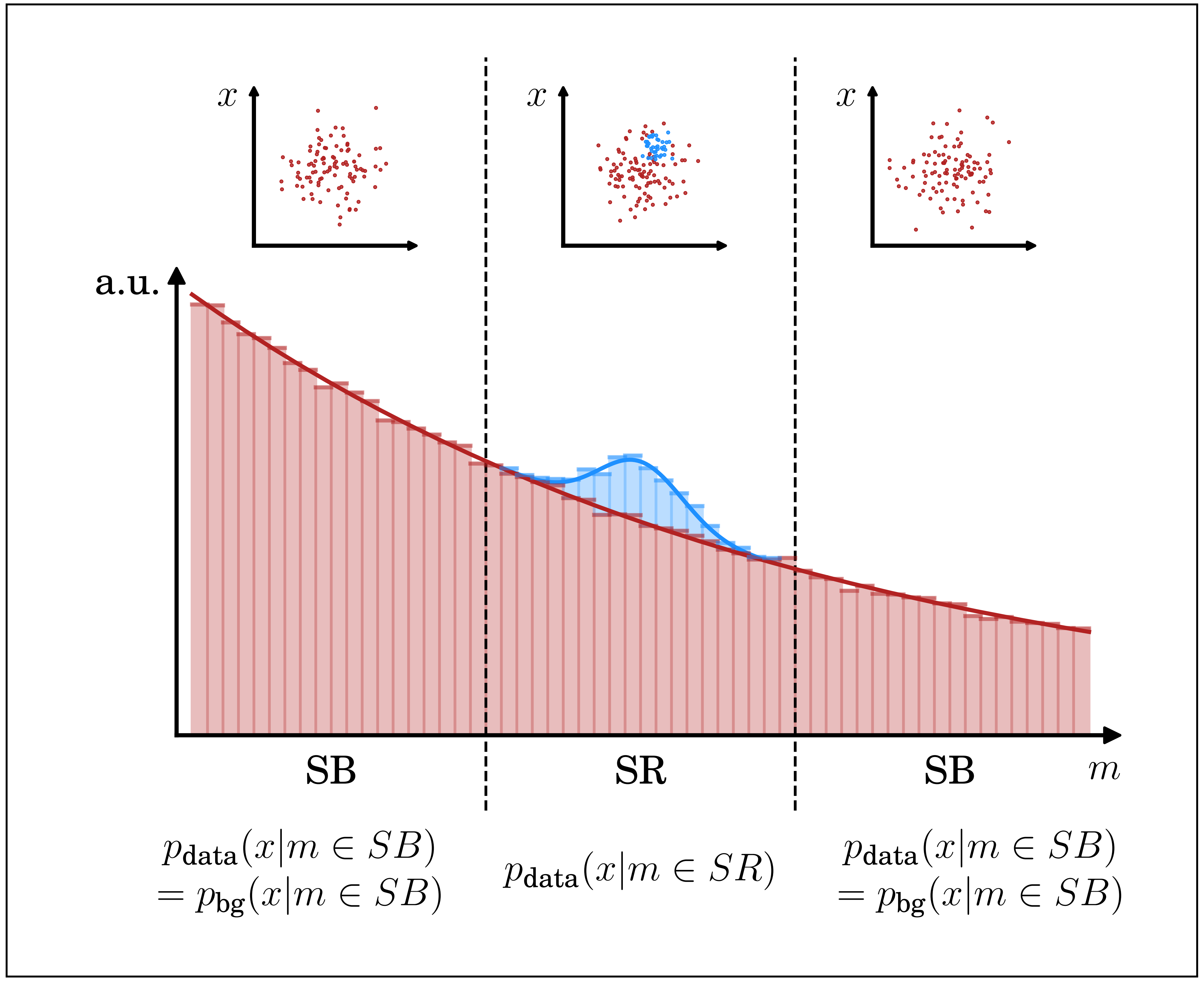
from 2109.00546



Idea: data vs sideband classifier

Collins, Howe & Nachman 1805.02664,1902.02634

from 2109.00546



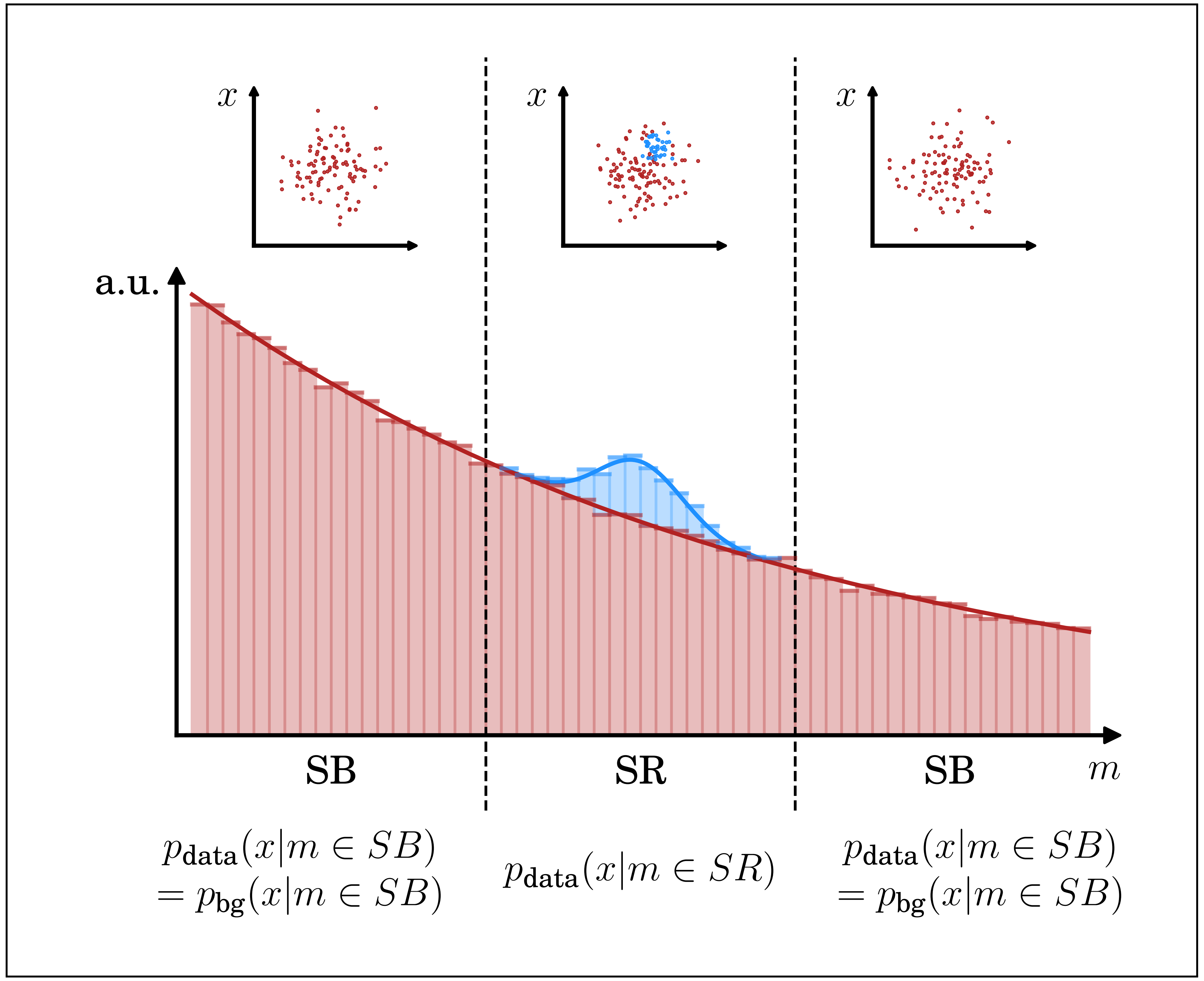
Train a NN classifier on SR vs SB data, learn

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

Idea: data vs sideband classifier

Collins, Howe & Nachman 1805.02664,1902.02634

from 2109.00546



Train a NN classifier on SR vs SB data, learn

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

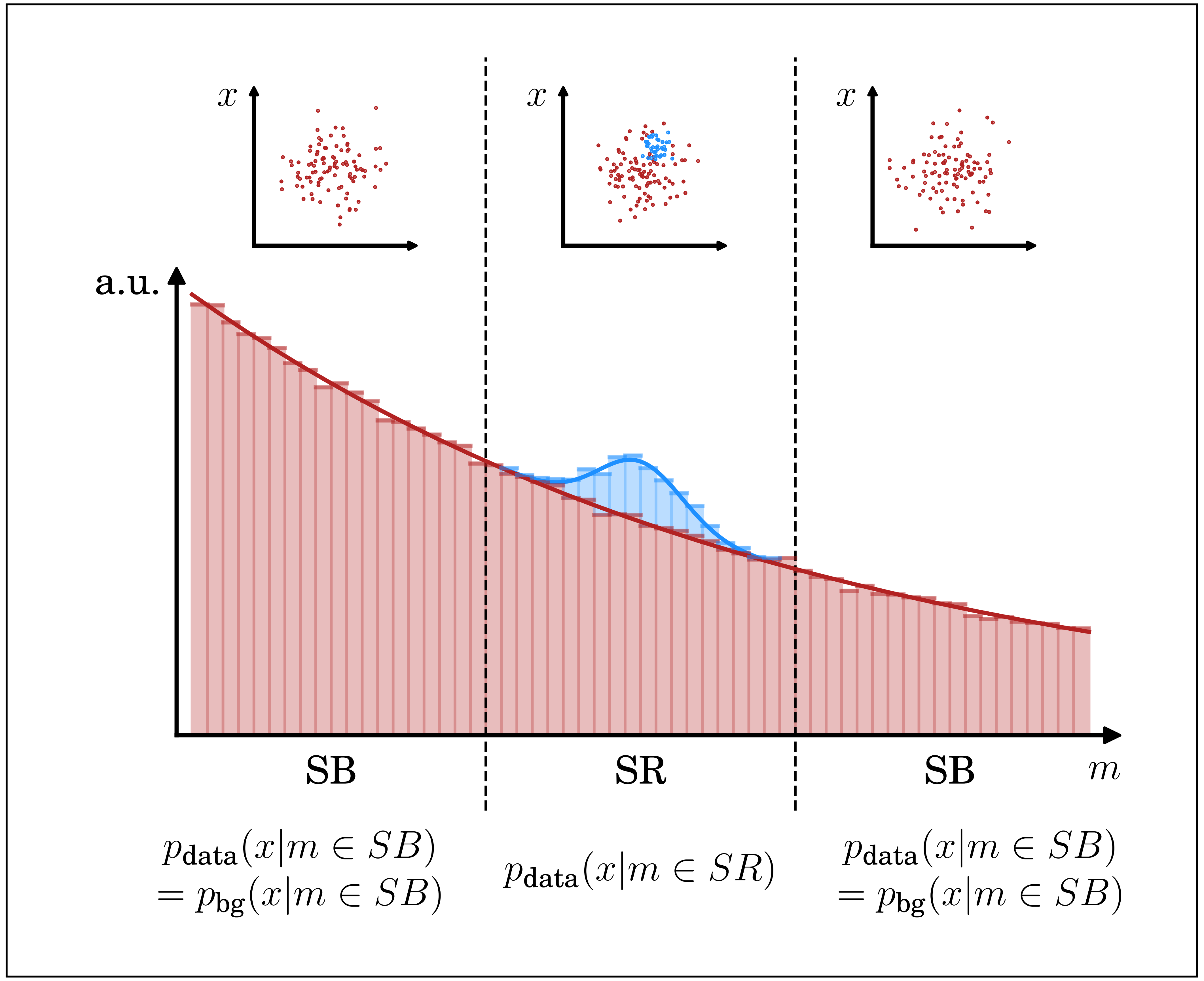
If $p_{bg,SB}(x) = p_{bg,SR}(x)$ [i.e. features x are independent of m in the background] then the classifier gives the desired likelihood ratio.

$$R_{classifier}(x) \to \frac{p_{data,SR}(x)}{p_{bg,SR}(x)}$$

Idea: data vs sideband classifier

Collins, Howe & Nachman 1805.02664,1902.02634

from 2109.00546



Train a NN classifier on SR vs SB data, learn

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

If $p_{bg,SB}(x) = p_{bg,SR}(x)$ [i.e. features x are independent of m in the background] then the classifier gives the desired likelihood ratio.

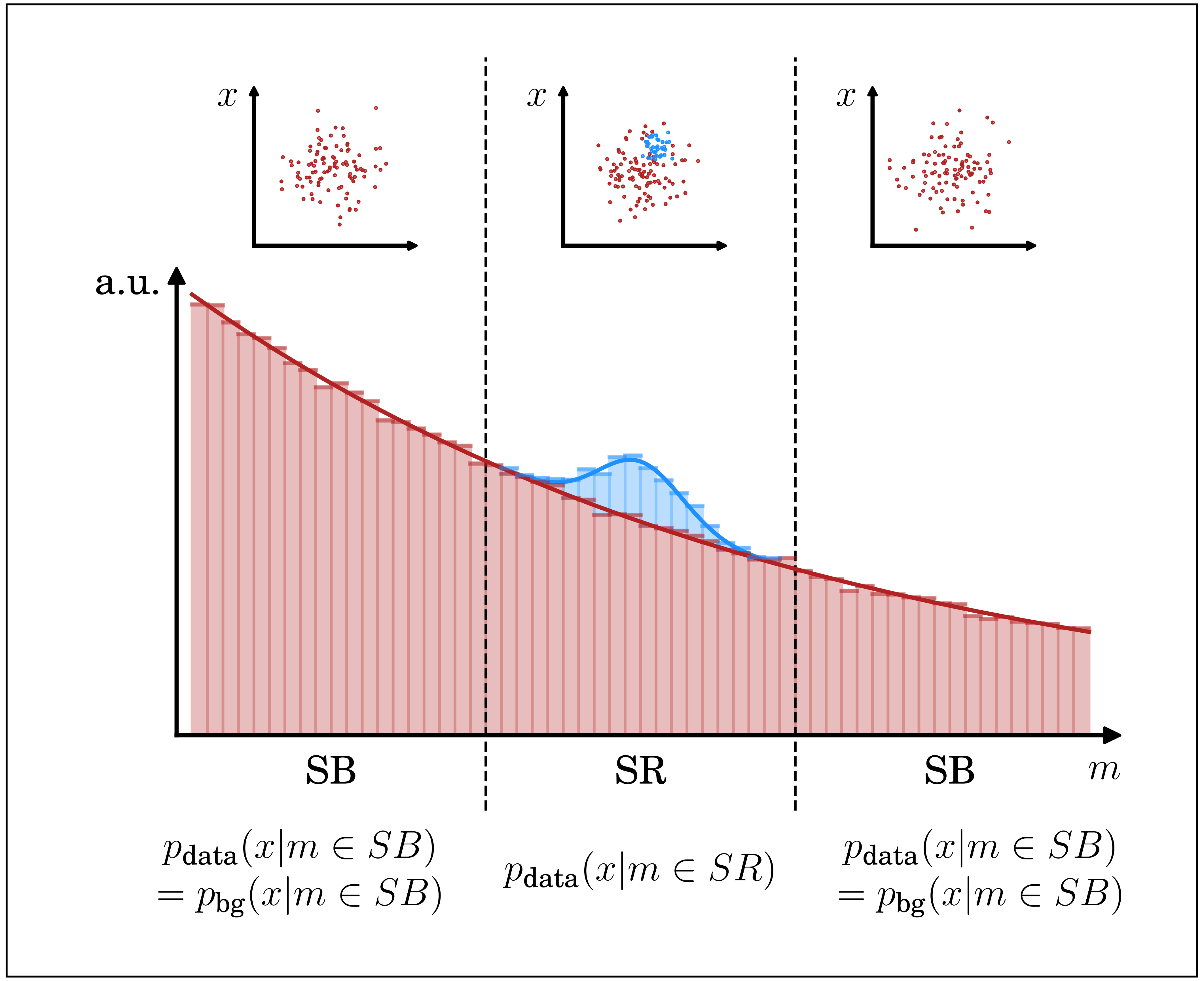
$$R_{classifier}(x) \to \frac{p_{data,SR}(x)}{p_{bg,SR}(x)}$$

"CWoLa Hunting"

Idea: conditional density estimation

Nachman & **DS** 2001.04990

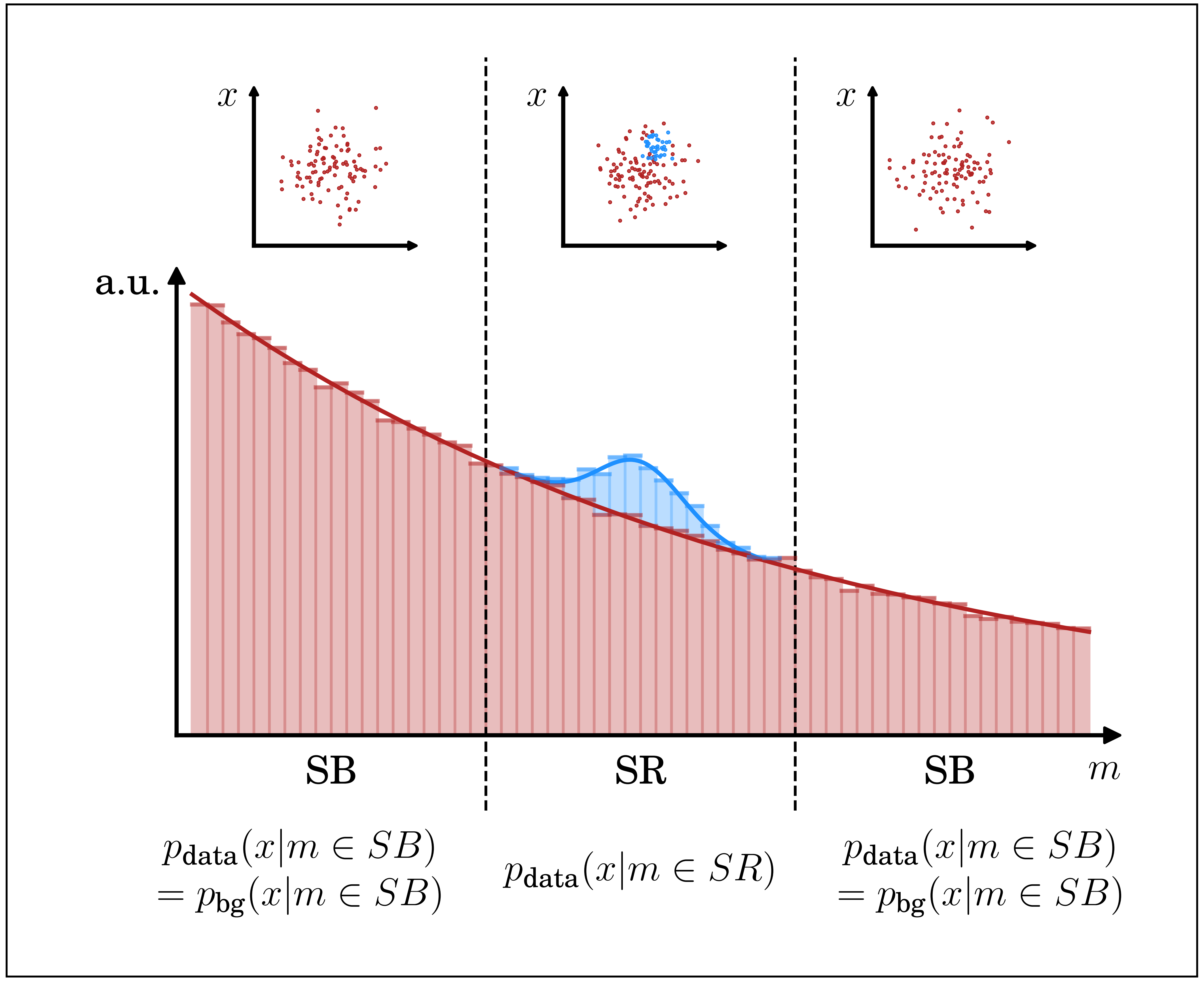
from 2109.00546



Idea: conditional density estimation

Nachman & **DS** 2001.04990

from 2109.00546

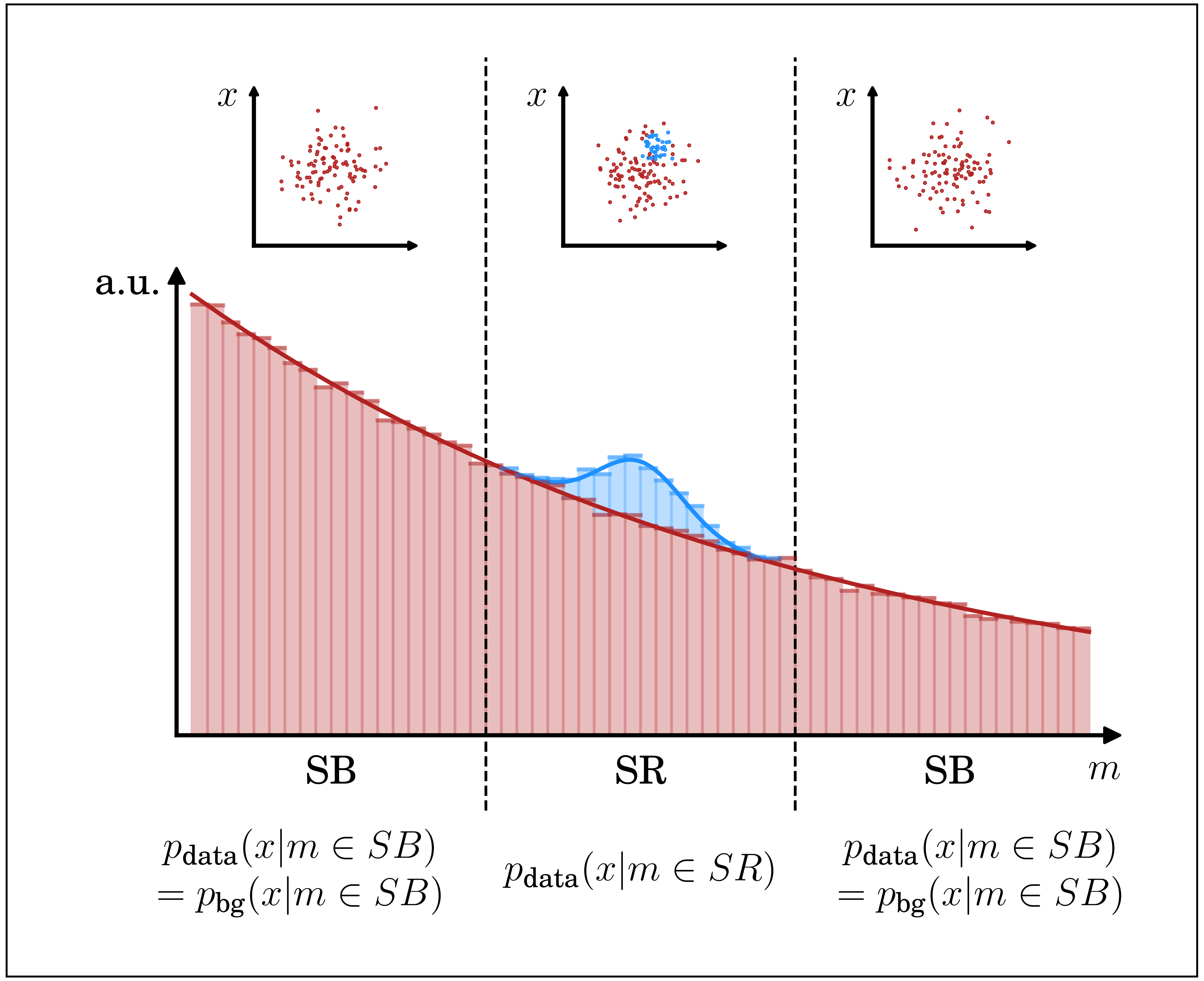


Train two separate ***normalizing flows (MAFs)*** on SR and SB events to learn $p_{data}(x\,|\,m \in SR)$ and $p_{data}(x\,|\,m \in SB) = p_{bg}(x\,|\,m \in SB)$.

Idea: conditional density estimation

Nachman & **DS** 2001.04990

from 2109.00546



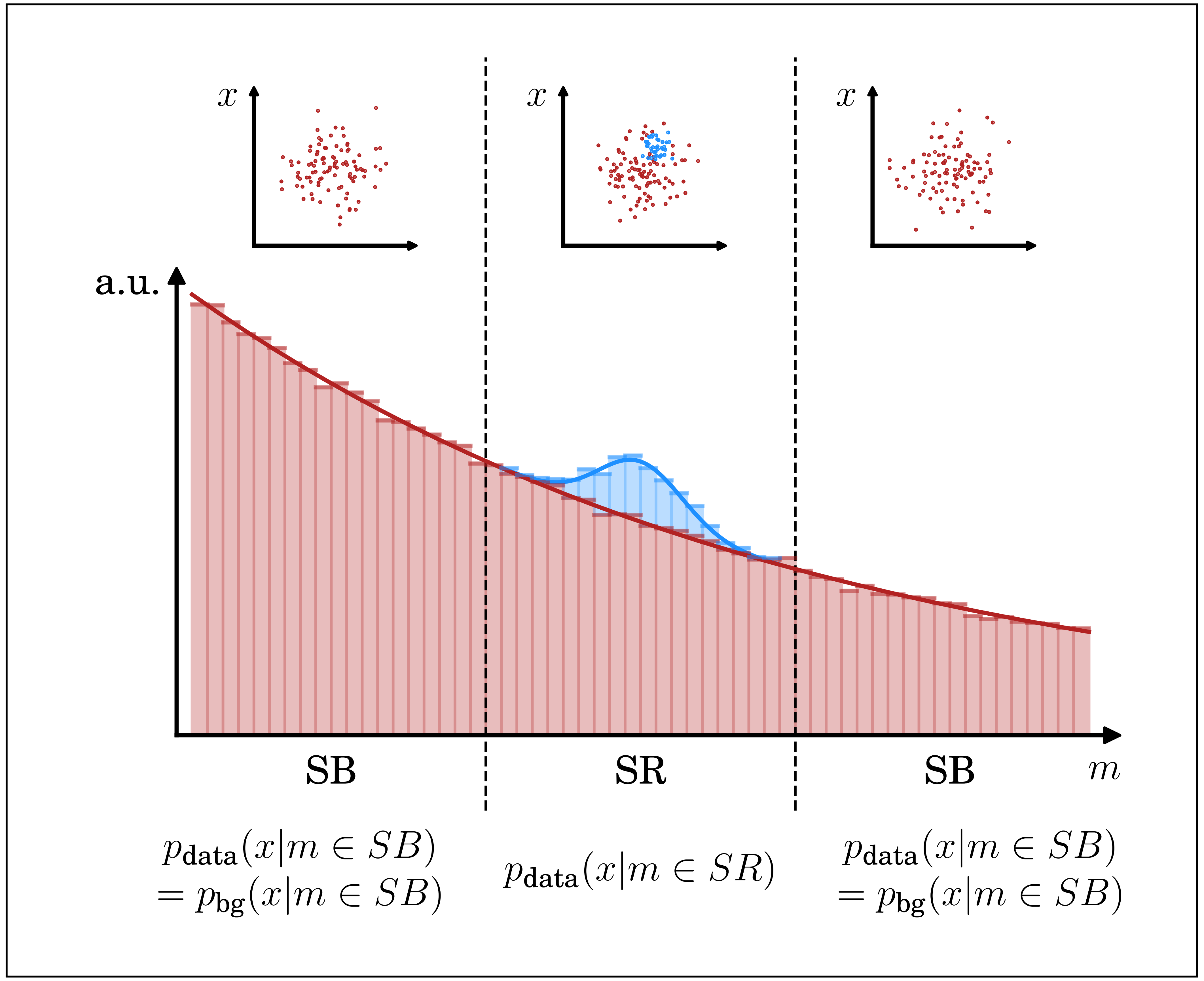
Train two separate ***normalizing flows (MAFs)*** on SR and SB events to learn $p_{data}(x\,|\,m \in SR)$ and $p_{data}(x\,|\,m \in SB) = p_{bg}(x\,|\,m \in SB)$.

The SB MAF automatically interpolates into the SR, giving an estimate of $p_{bg}(x\,|\,m \in SR)$.

Idea: conditional density estimation

Nachman & **DS** 2001.04990

from 2109.00546



Train two separate ***normalizing flows (MAFs)*** on SR and SB events to learn $p_{data}(x\,|\,m \in SR)$ and $p_{data}(x\,|\,m \in SB) = p_{bg}(x\,|\,m \in SB)$.

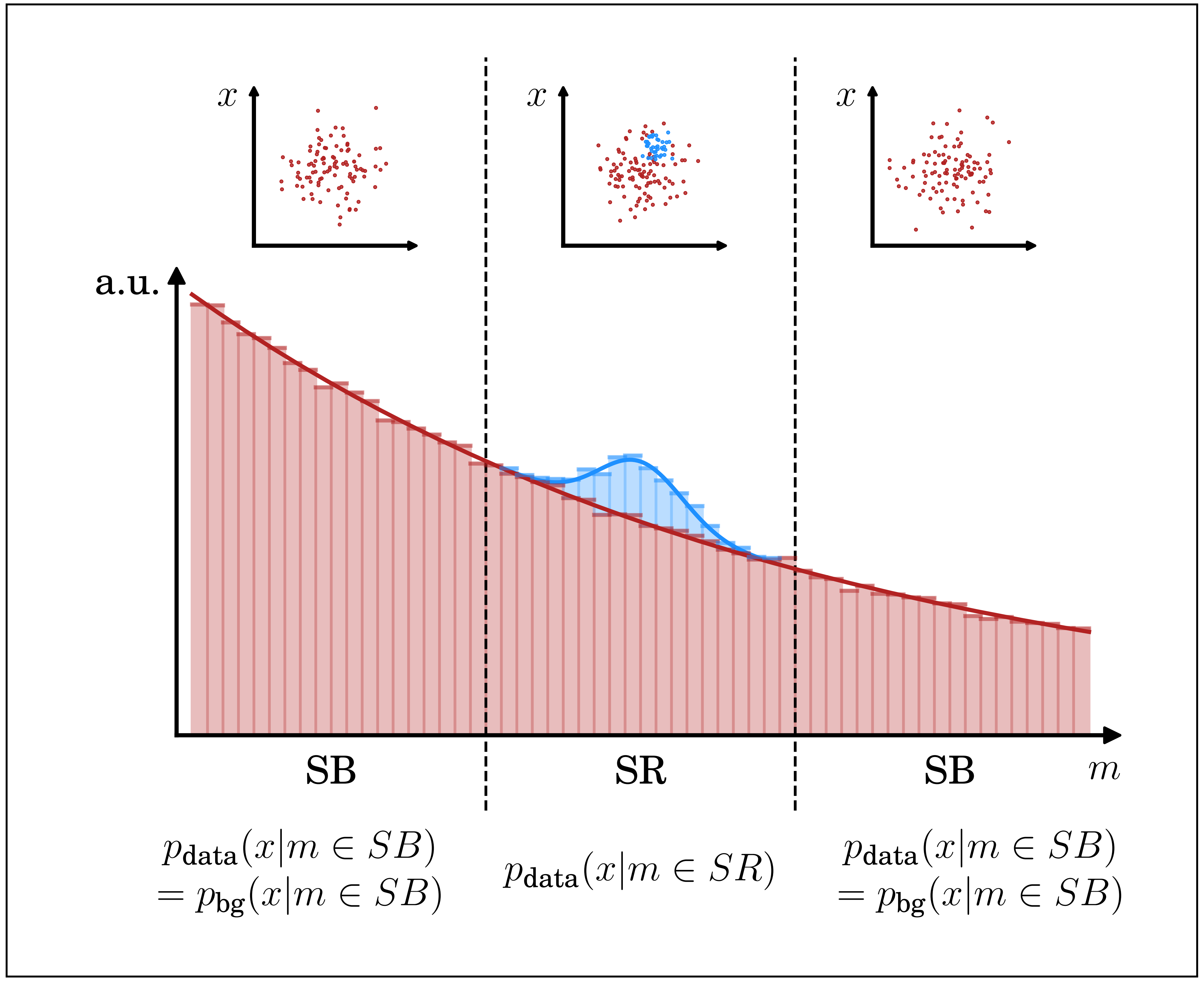
The SB MAF automatically interpolates into the SR, giving an estimate of $p_{bg}(x\,|\,m \in SR)$.

Construct likelihood ratio explicitly.

Idea: conditional density estimation

Nachman & **DS** 2001.04990

from 2109.00546



Train two separate ***normalizing flows (MAFs)*** on SR and SB events to learn $p_{data}(x\,|\,m \in SR)$ and $p_{data}(x\,|\,m \in SB) = p_{bg}(x\,|\,m \in SB)$.

The SB MAF automatically interpolates into the SR, giving an estimate of $p_{bg}(x\,|\,m \in SR)$.

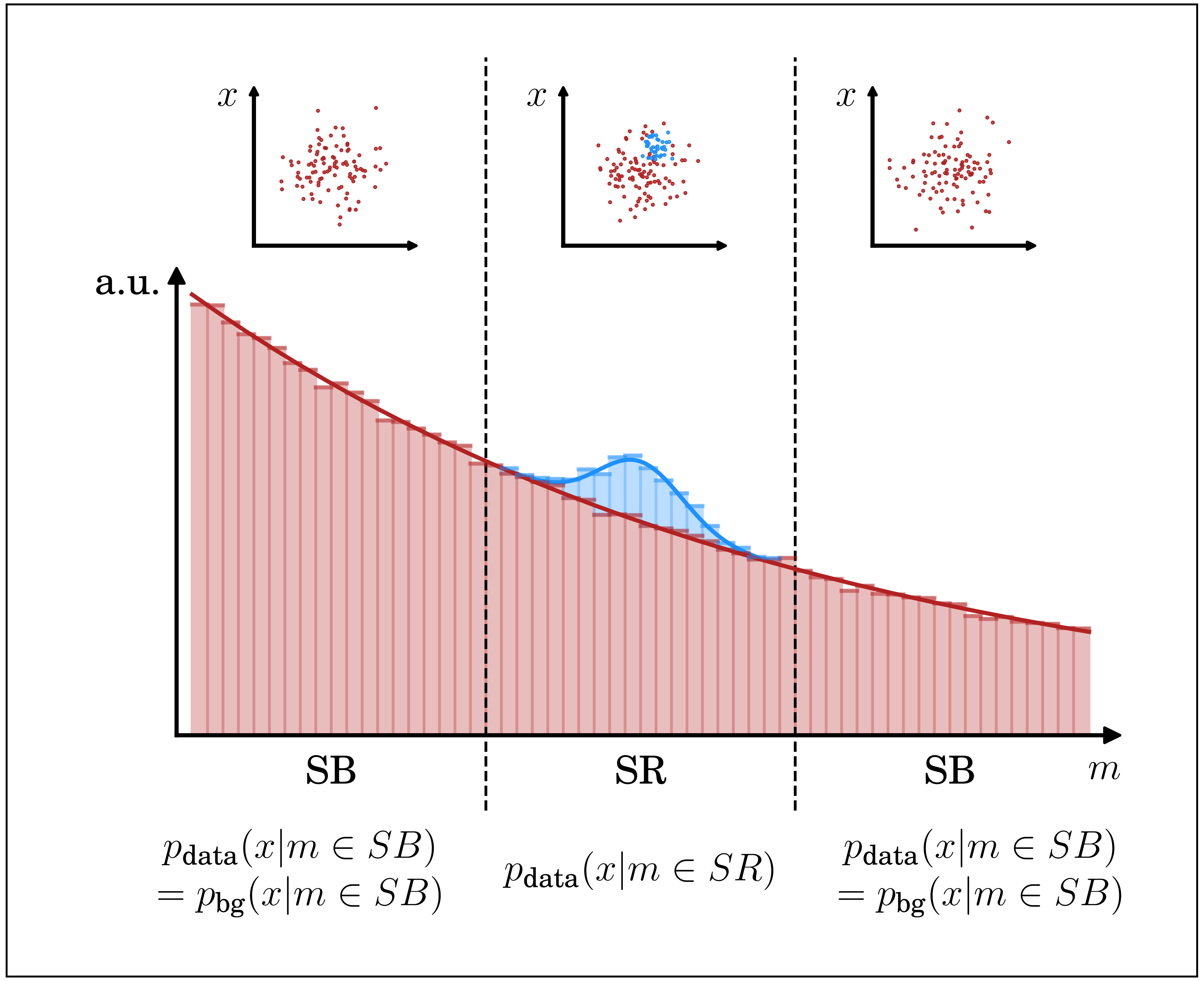
Construct likelihood ratio explicitly.

- Pros: robust against correlations!

Idea: conditional density estimation

Nachman & **DS** 2001.04990

from 2109.00546



Train two separate ***normalizing flows (MAFs)*** on SR and SB events to learn $p_{data}(x\,|\,m \in SR)$ and $p_{data}(x\,|\,m \in SB) = p_{bg}(x\,|\,m \in SB)$.

The SB MAF automatically interpolates into the SR, giving an estimate of $p_{bg}(x\,|\,m \in SR)$.

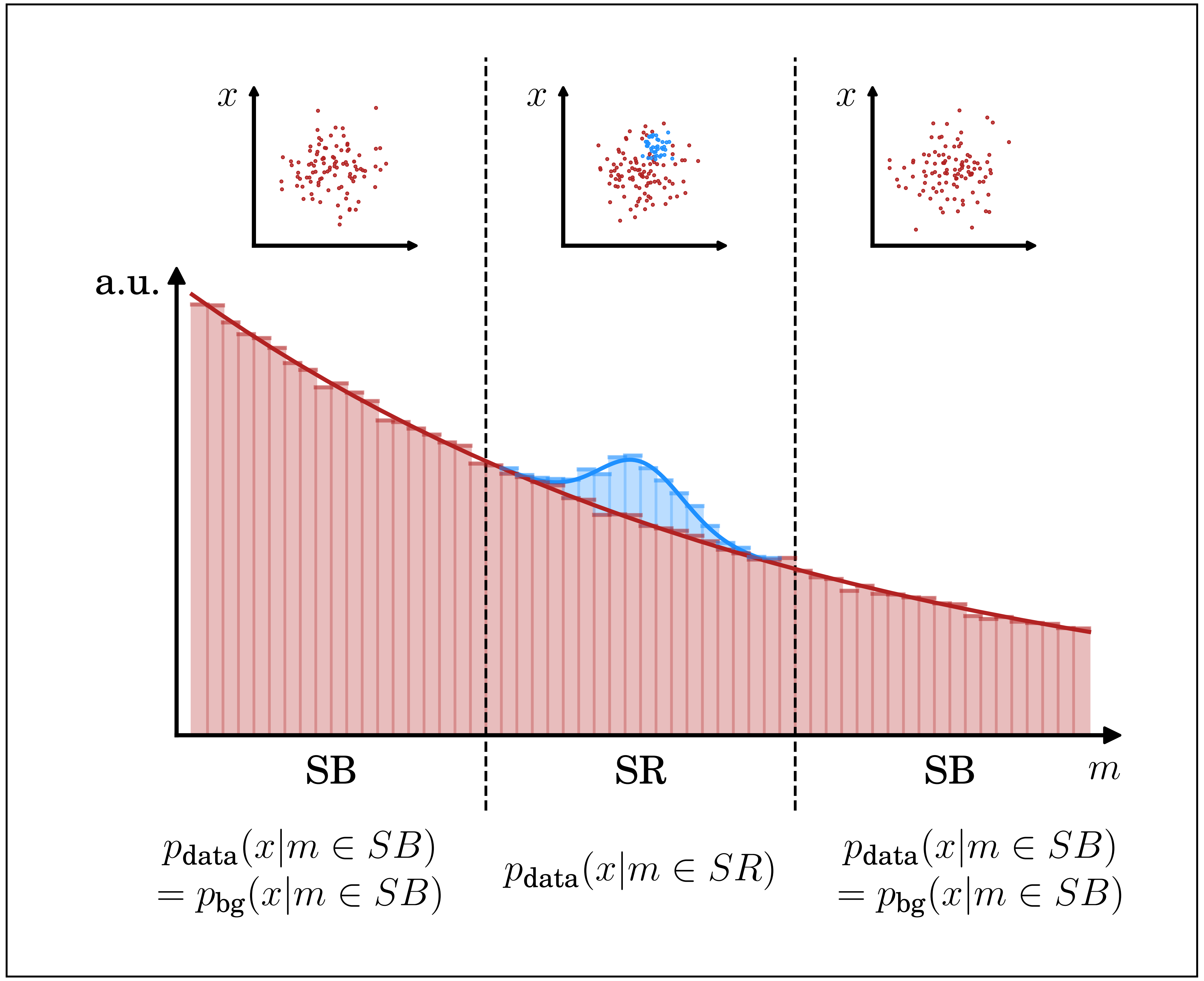
Construct likelihood ratio explicitly.

- Pros: robust against correlations!
- Cons: density estimation much harder than classification

Idea: conditional density estimation

Nachman & **DS** 2001.04990

from 2109.00546



Train two separate ***normalizing flows (MAFs)*** on SR and SB events to learn $p_{data}(x\,|\,m \in SR)$ and $p_{data}(x\,|\,m \in SB) = p_{bg}(x\,|\,m \in SB)$.

The SB MAF automatically interpolates into the SR, giving an estimate of $p_{bg}(x\,|\,m \in SR)$.

Construct likelihood ratio explicitly.

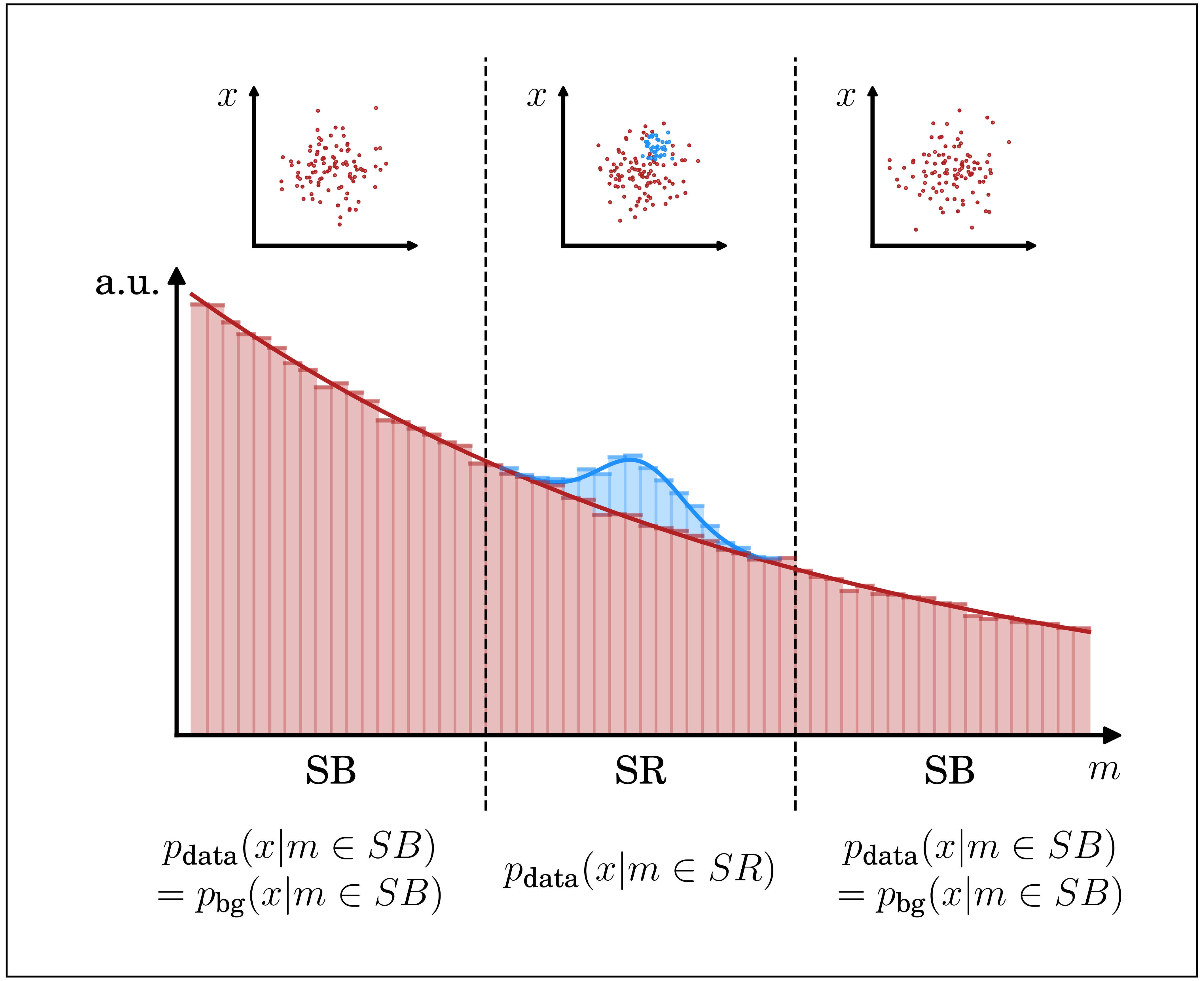
- Pros: robust against correlations!
- Cons: density estimation much harder than classification

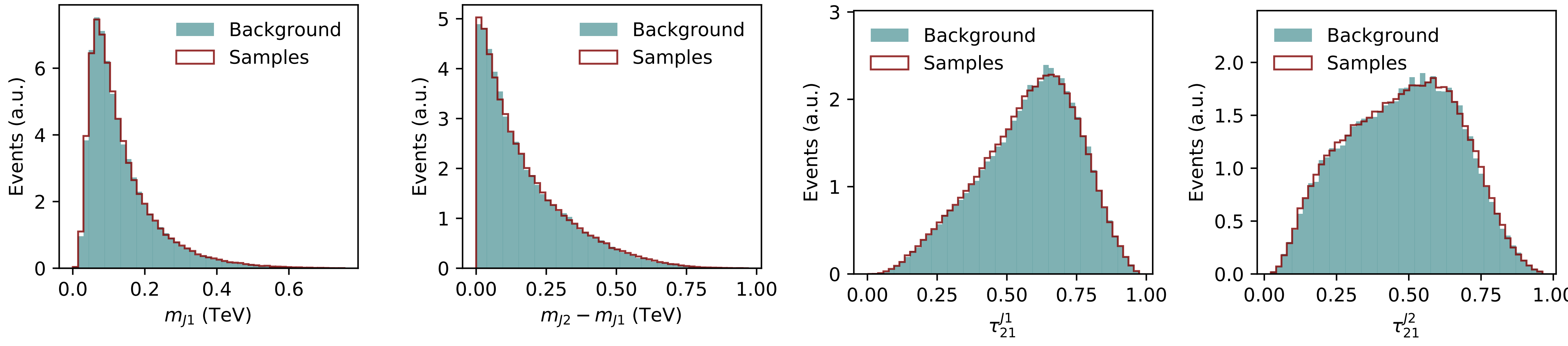
"**ANO**maly detection with **D**ensity **E**stimation **(ANODE)**"

Idea: density estimation+classification

Hallin, Isaacson, Kasieczka, Krause, Nachman, Quadfasel, Schlaffer, **DS** & Sommerhalder 2109.00546

from 2109.00546

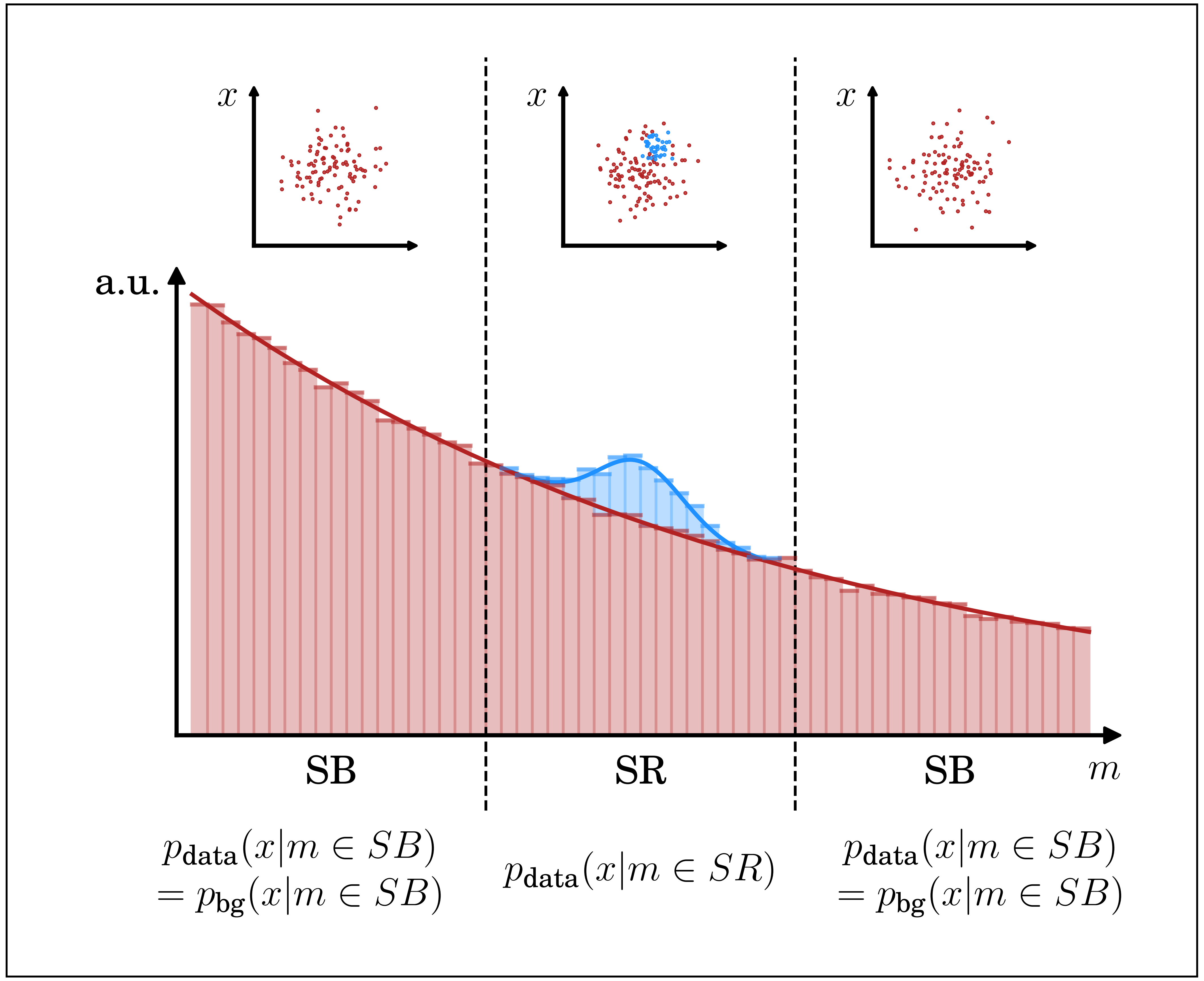


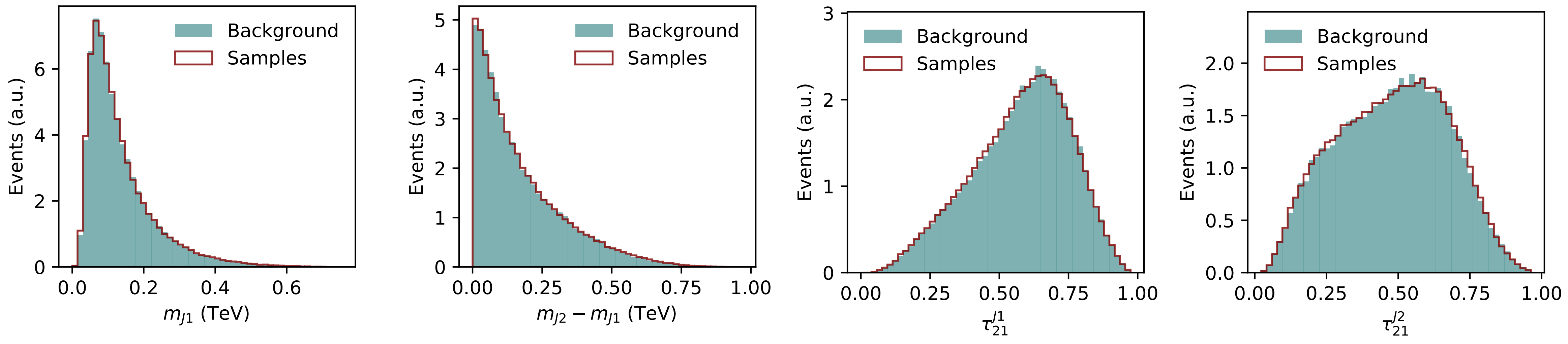


Idea: density estimation+classification

Hallin, Isaacson, Kasieczka, Krause, Nachman, Quadfasel, Schlaffer, **DS** & Sommerhalder 2109.00546

from 2109.00546



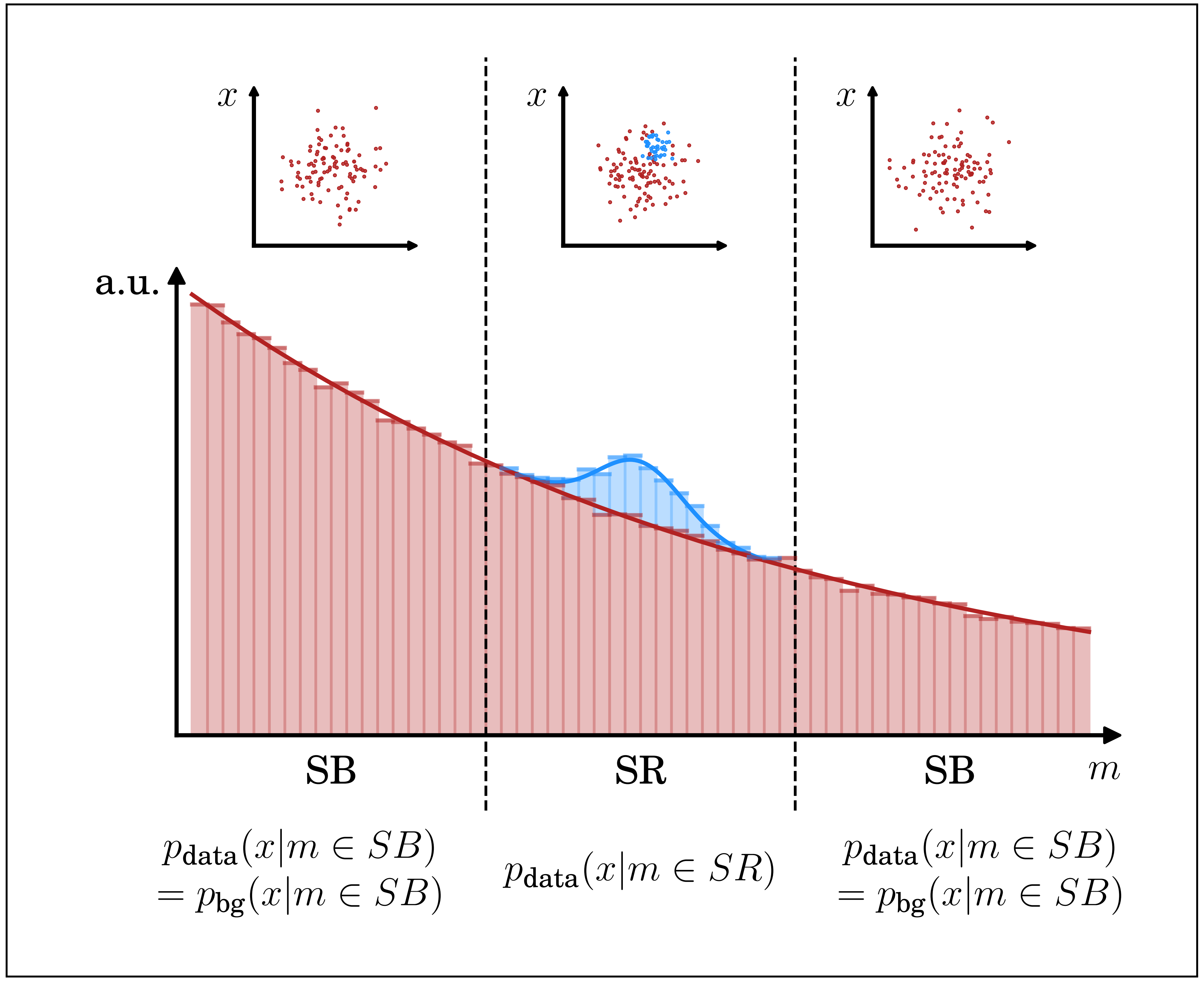


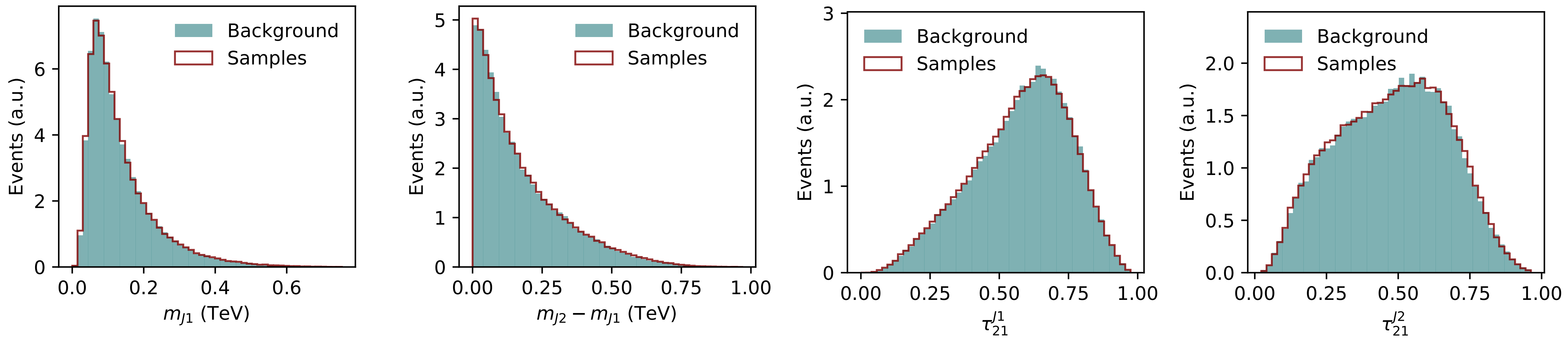
Can also sample from trained density estimator!

Idea: density estimation+classification

Hallin, Isaacson, Kasieczka, Krause, Nachman, Quadfasel, Schlaffer, **DS** & Sommerhalder 2109.00546

from 2109.00546





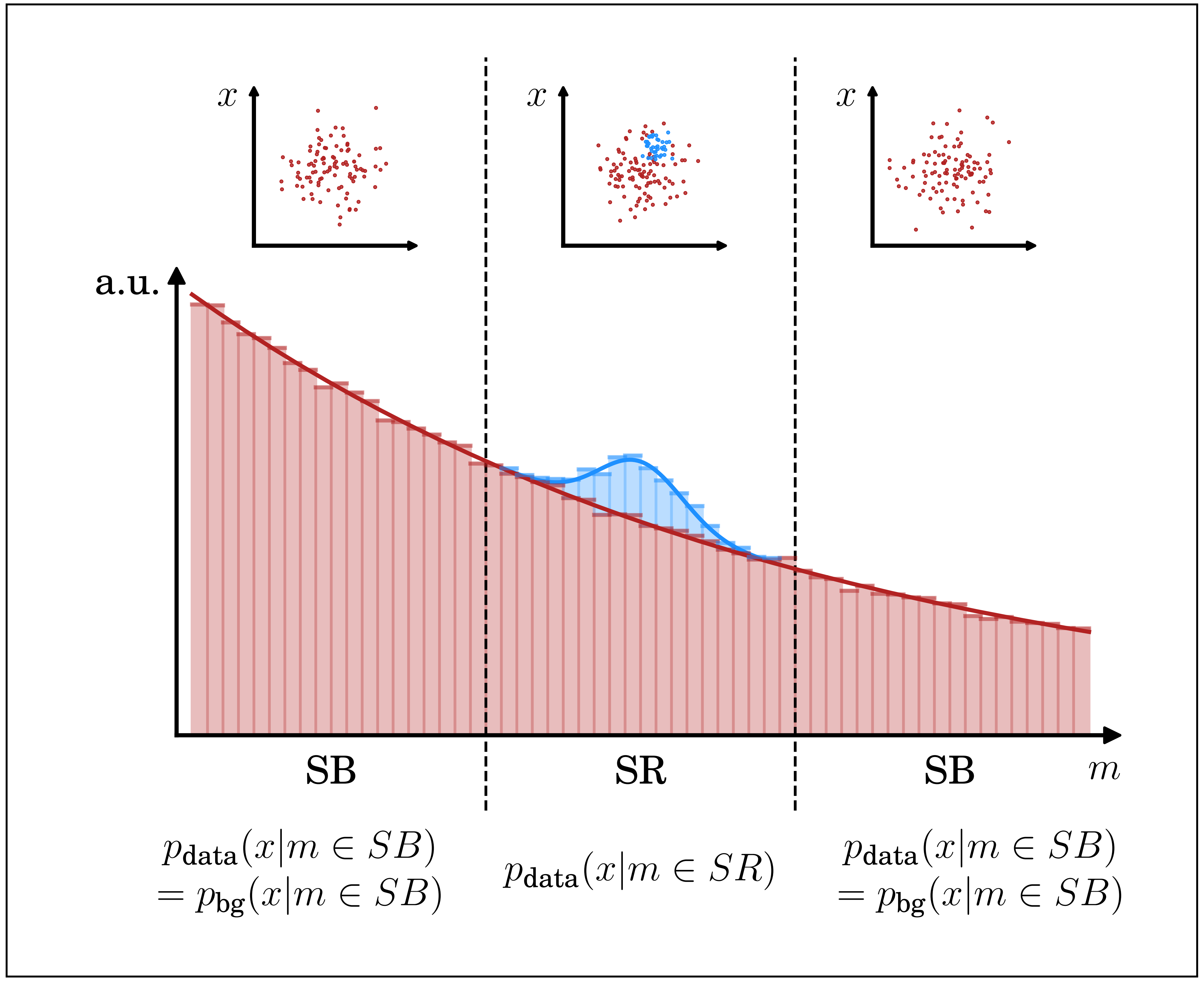
Can also sample from trained density estimator!

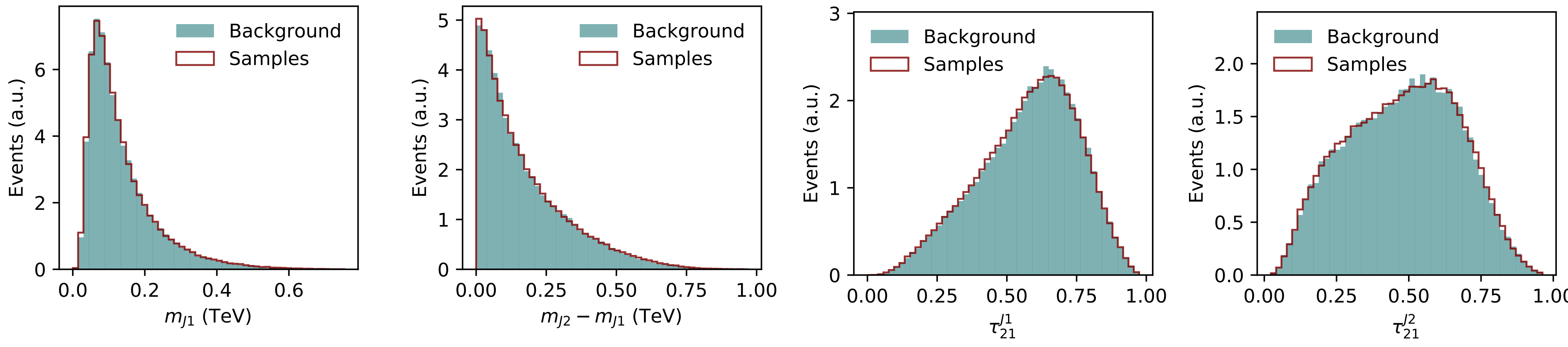
Sample synthetic bg events from the interpolated SB MAF as in ANODE.

Idea: density estimation+classification

Hallin, Isaacson, Kasieczka, Krause, Nachman, Quadfasel, Schlaffer, **DS** & Sommerhalder 2109.00546

from 2109.00546





Can also sample from trained density estimator!

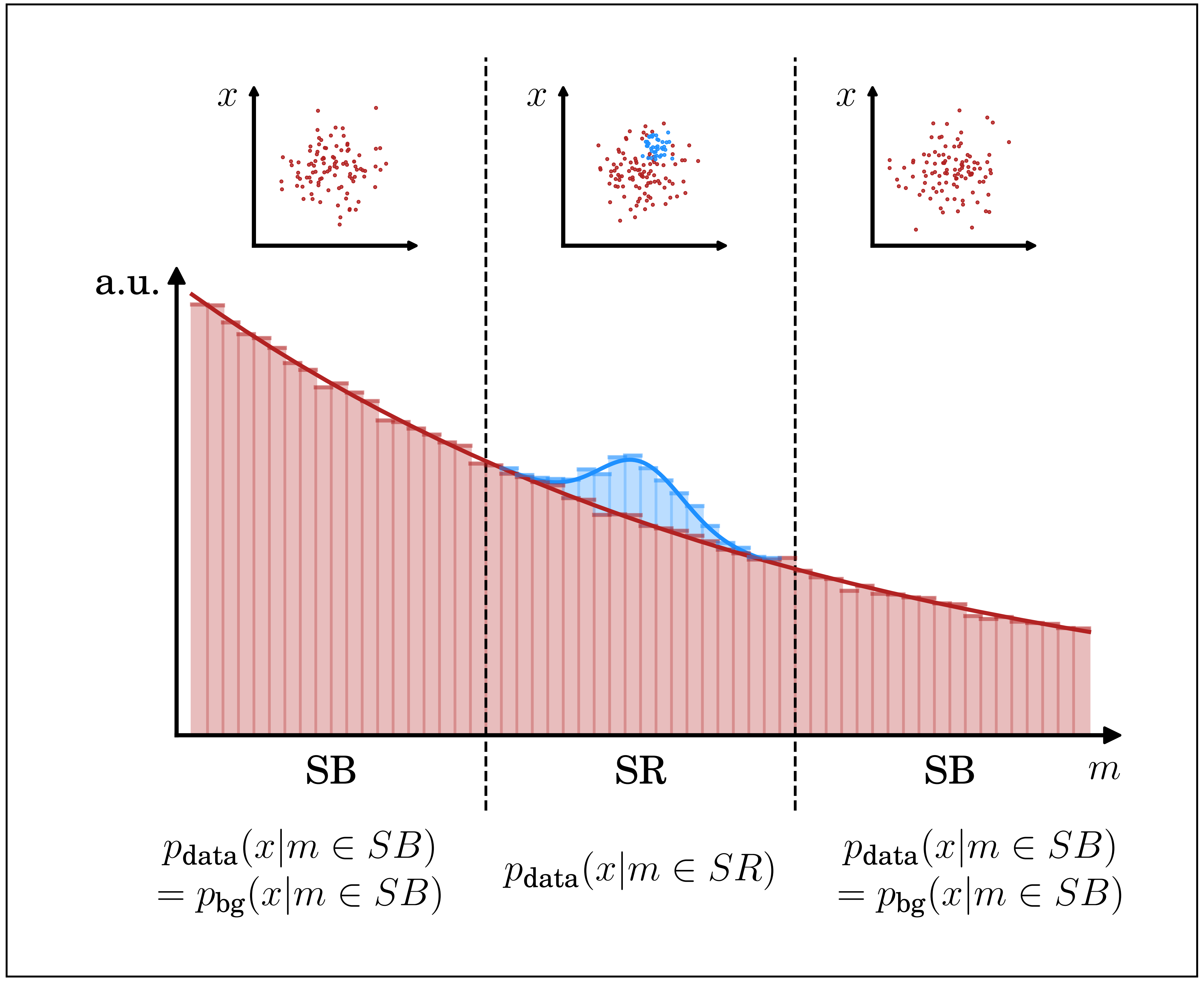
Sample synthetic bg events from the interpolated SB MAF as in ANODE.

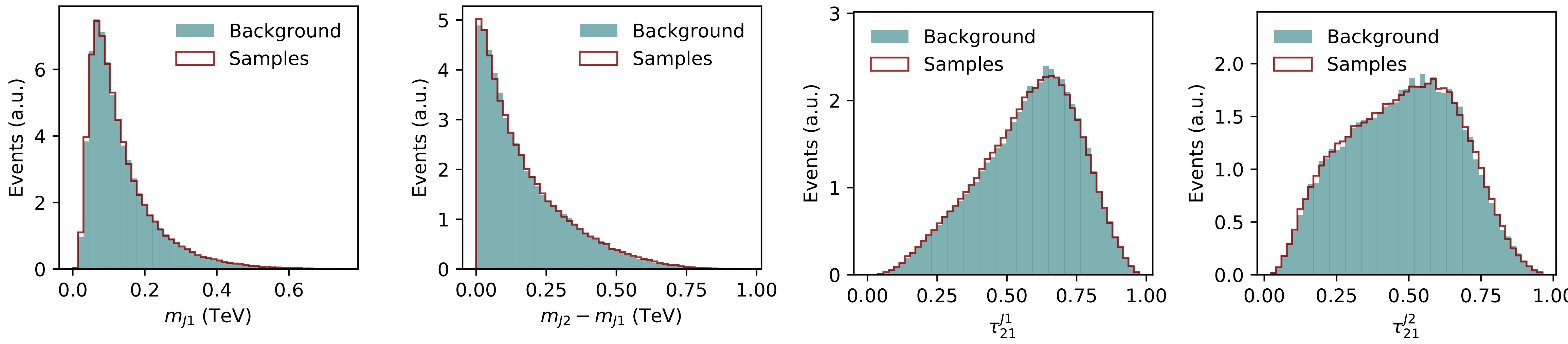
Then learn data-vs-bg classifier as in CWoLa Hunting.

Idea: density estimation+classification

Hallin, Isaacson, Kasieczka, Krause, Nachman, Quadfasel, Schlaffer, **DS** & Sommerhalder 2109.00546

from 2109.00546





Can also sample from trained density estimator!

Sample synthetic bg events from the interpolated SB MAF as in ANODE.

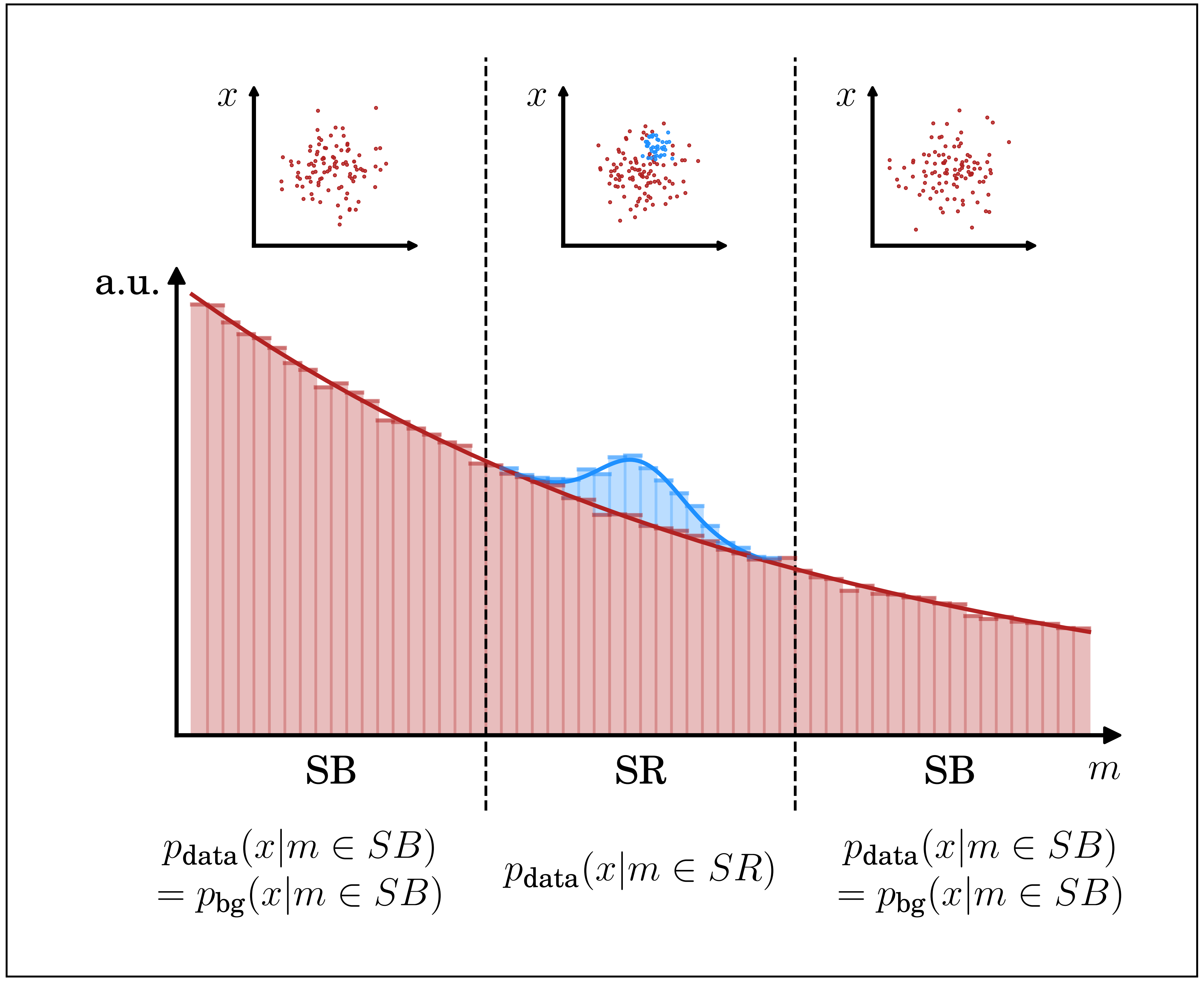
Then learn data-vs-bg classifier as in CWoLa Hunting.

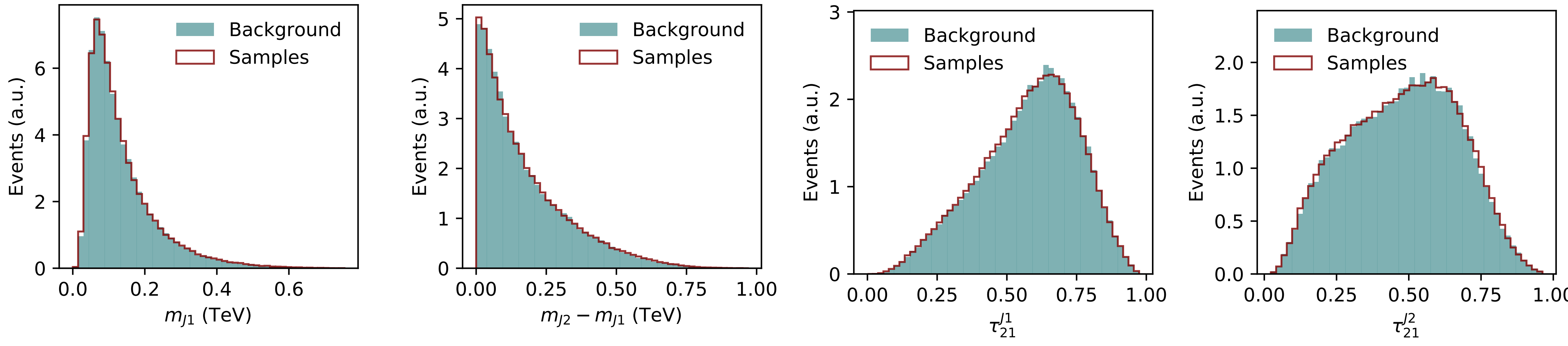
- Pros: Robust against correlations ***and*** don't have to learn separate density estimator for $p_{data}(x \,|\, m \in SR)$

Idea: density estimation+classification

Hallin, Isaacson, Kasieczka, Krause, Nachman, Quadfasel, Schlaffer, **DS** & Sommerhalder 2109.00546

from 2109.00546





Can also sample from trained density estimator!

Sample synthetic bg events from the interpolated SB MAF as in ANODE.

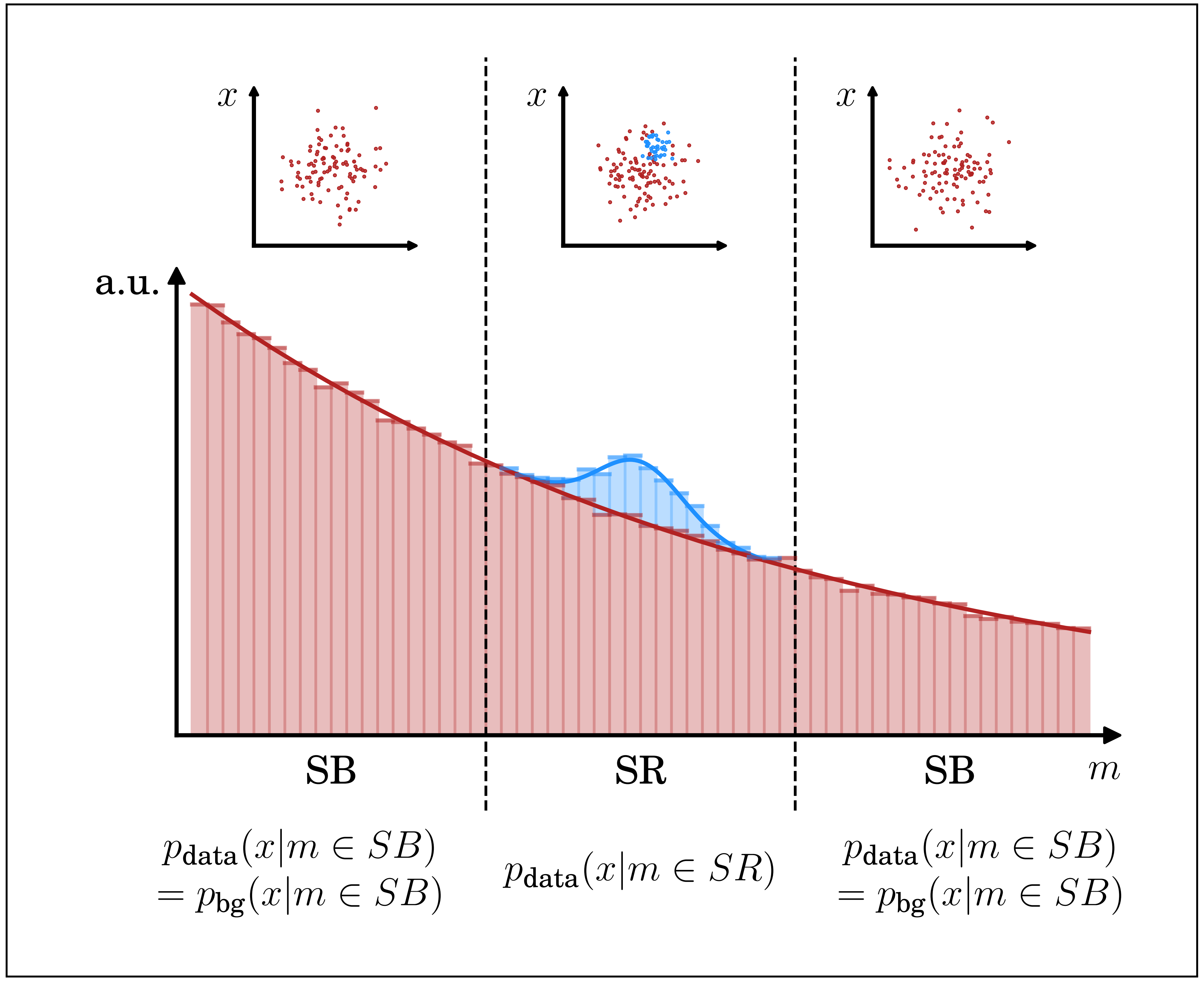
Then learn data-vs-bg classifier as in CWoLa Hunting.

- Pros: Robust against correlations ***and*** don't have to learn separate density estimator for $p_{data}(x \mid m \in SR)$

"Classifying Anomalies THrough Outer Density Estimation (CATHODE)"

Summary of methods

from 2109.00546



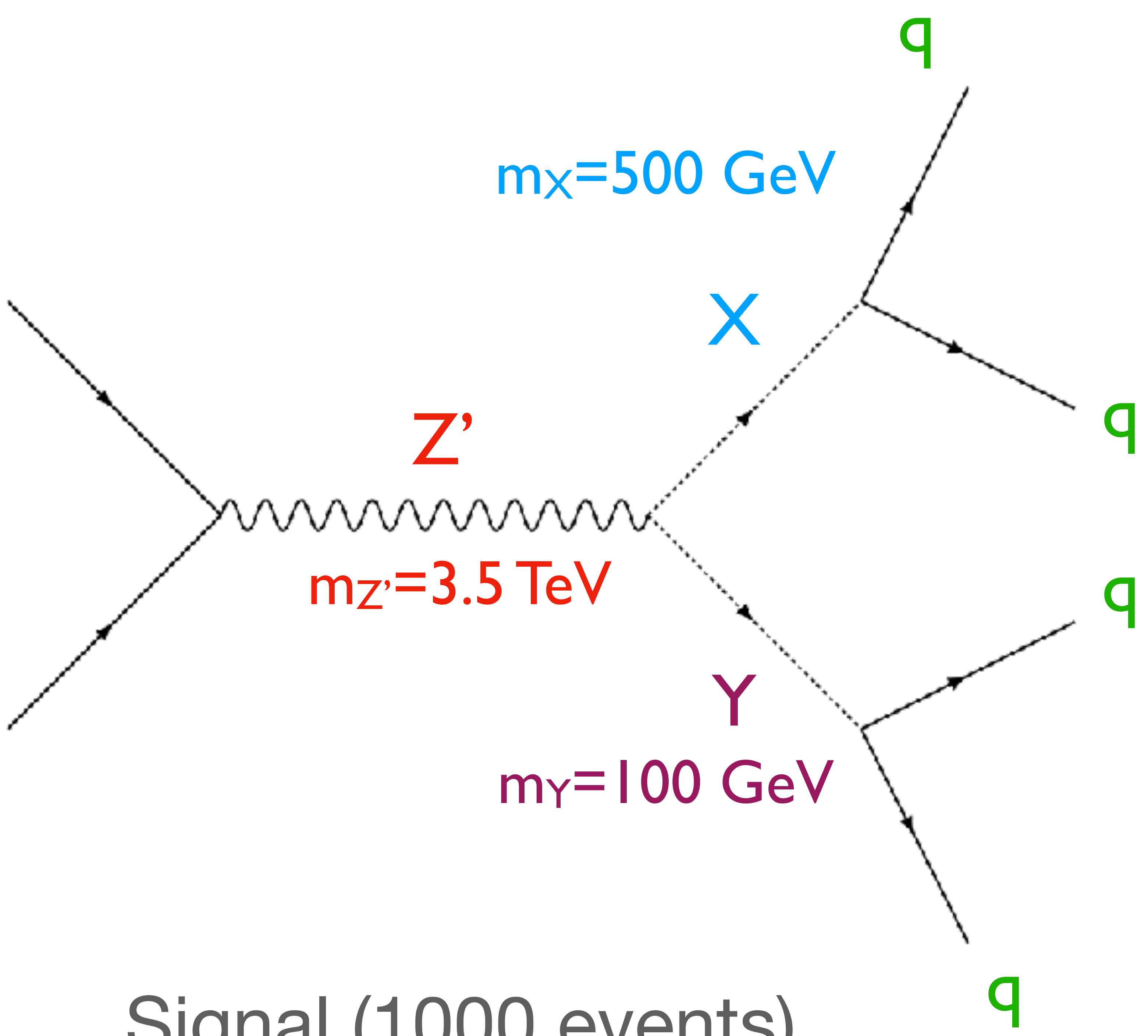
- CWoLa Hunting: **classifier** between SB and SR data
- ANODE: two **conditional density estimators** on SB and SR data; interpolate SB density estimator into SR
- CATHODE: single **conditional density estimator** on SB data; sample interpolated SB density estimator in SR; **classifier** between sampled events and data in SR

Comparison of methods

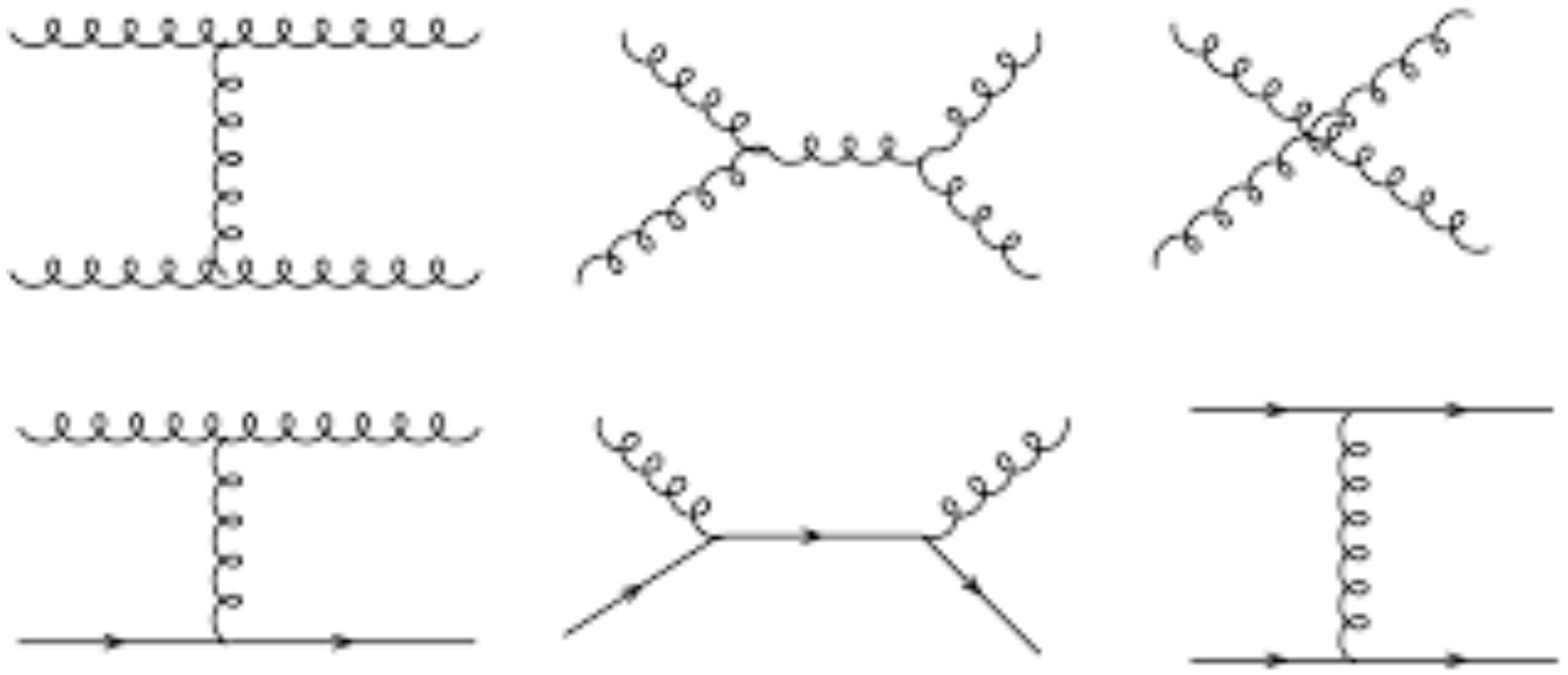
- Simulated with Pythia8 + Delphes
- pT(J1)>1.2 TeV trigger
- 4-vectors of every reconstructed particle in the event

We compared the methods on a common toy dataset:

LHC Olympics 2020 R&D dataset [https://doi.org/10.5281/zenodo.2629072]



Signal (1000 events)

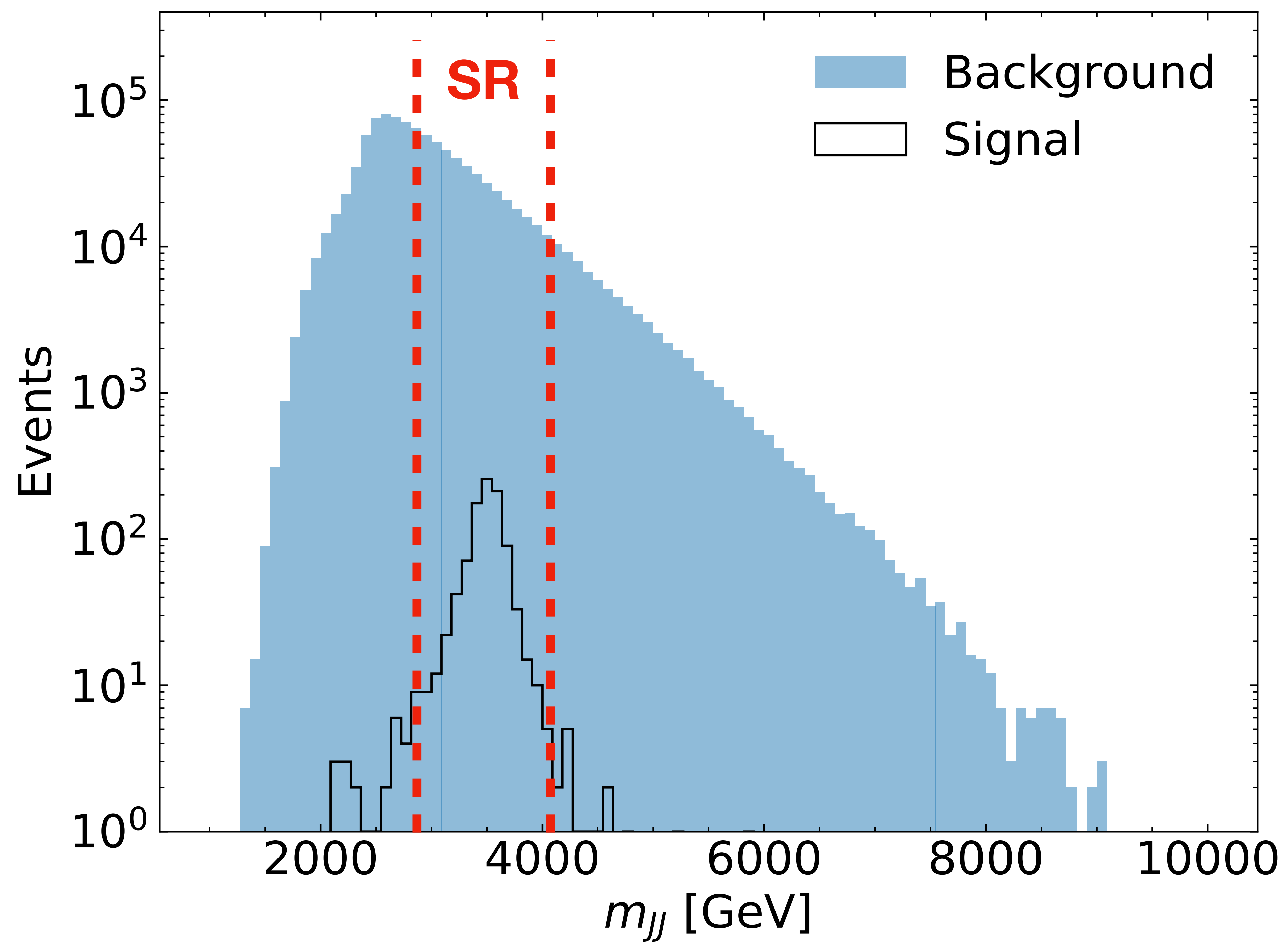


Background: QCD dijets (1M events)

No explicit search at the LHC for this scenario!

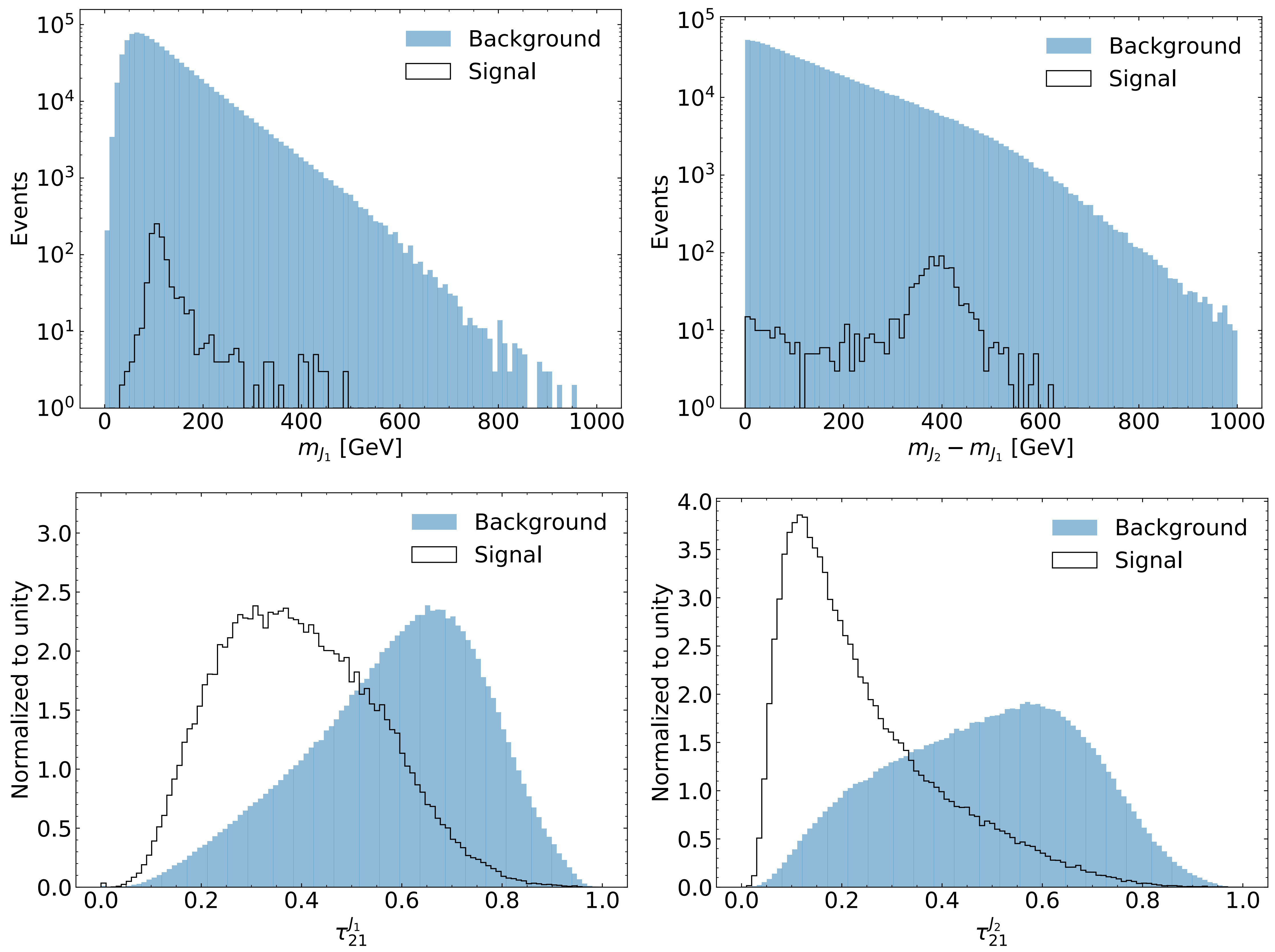
LHCO2020 R&D Dataset

Resonant feature



$m = m_{JJ}$

Additional features



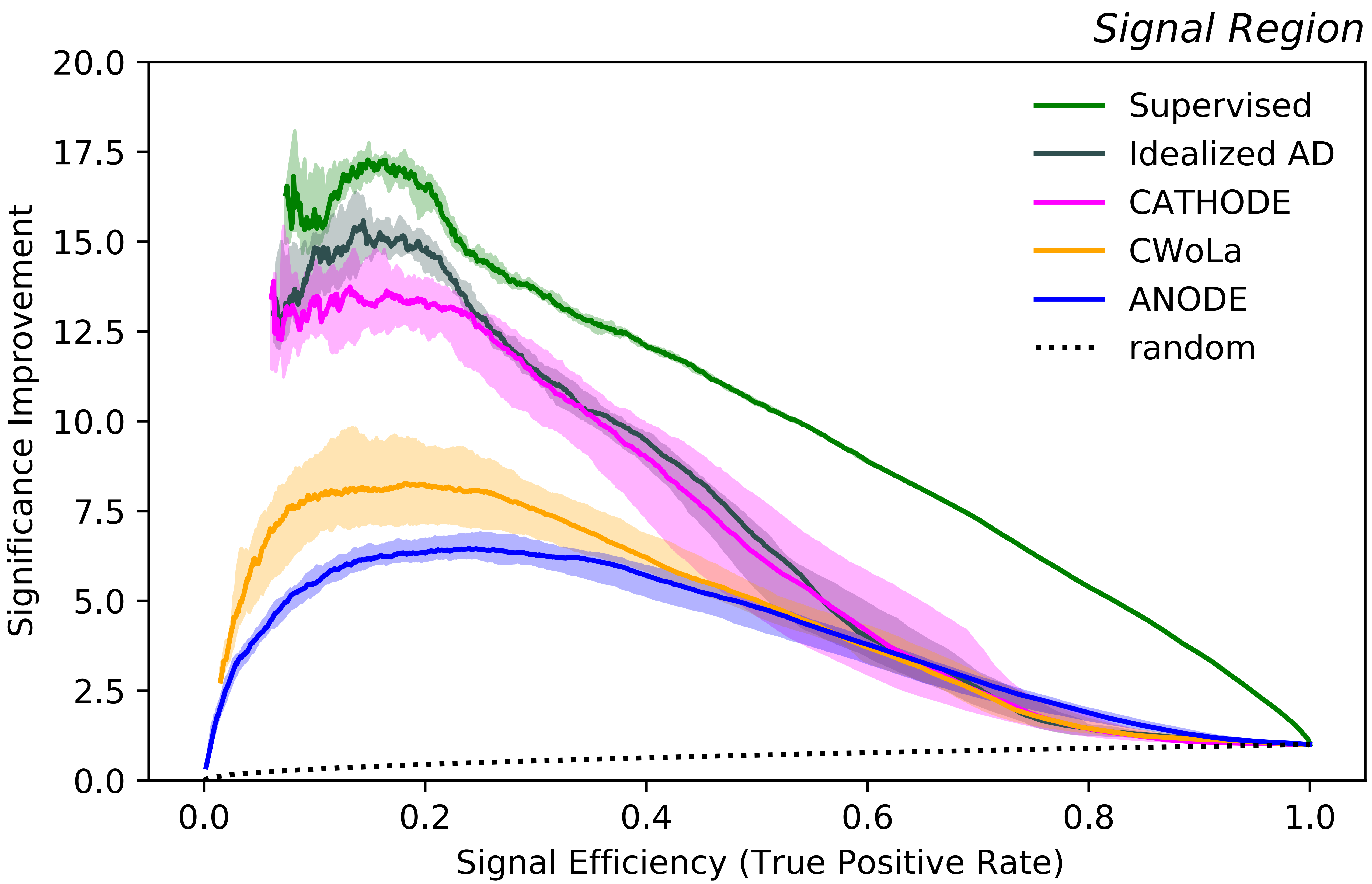
$x = (m_{J_1}, m_{J_2}, \tau_{21}^{J_1}, \tau_{21}^{J_2})$

Benchmark signal strength: S_{SR}=772, B_{SR}=120,000

S_{SR}/B_{SR}~0.6%, $S_{SR}/\sqrt{B_{SR}}$~2.2

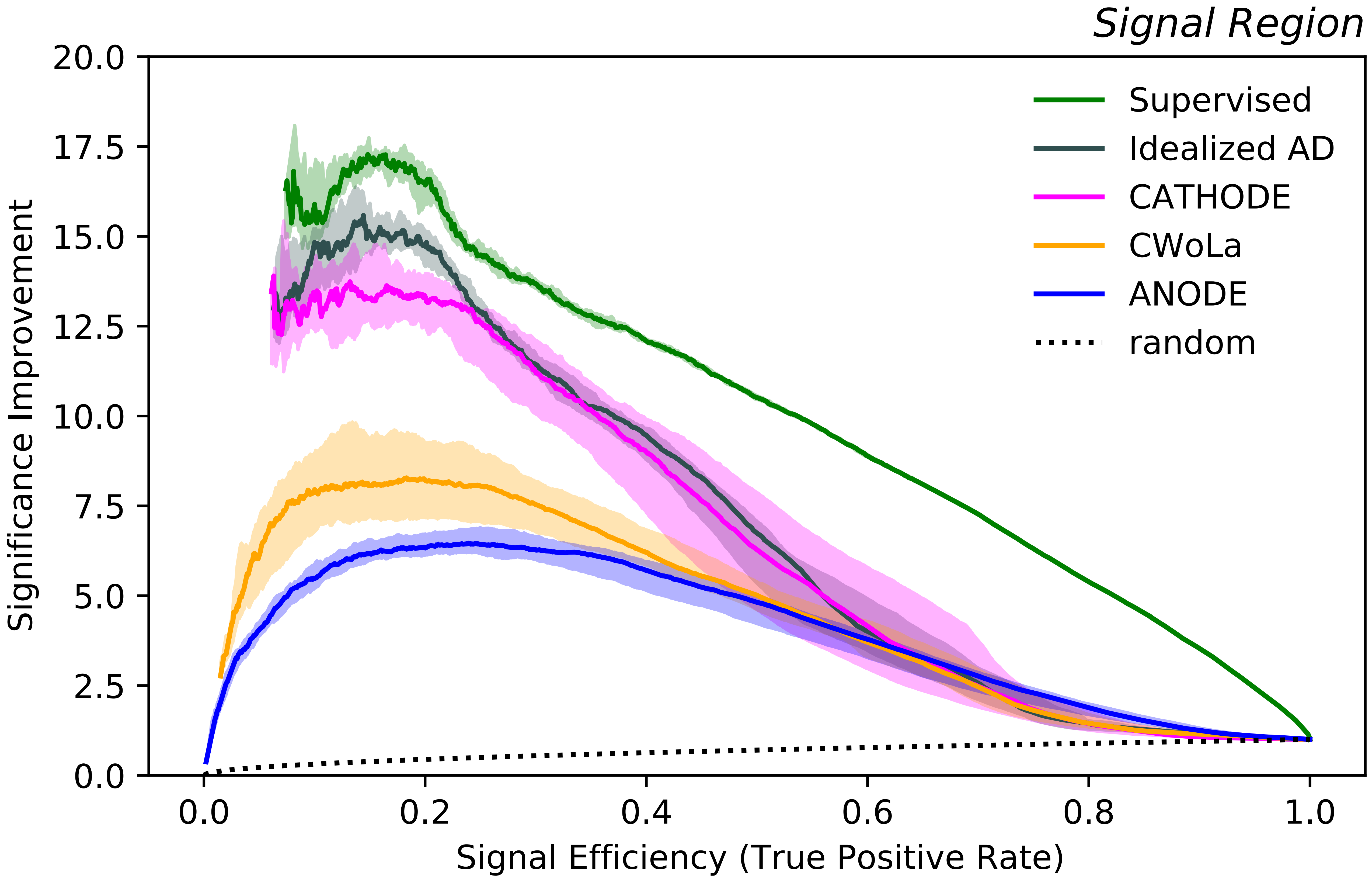
Comparison of methods

Significance improvement characteristic (SIC): $\epsilon_S/\sqrt{\epsilon_B}$



Comparison of methods

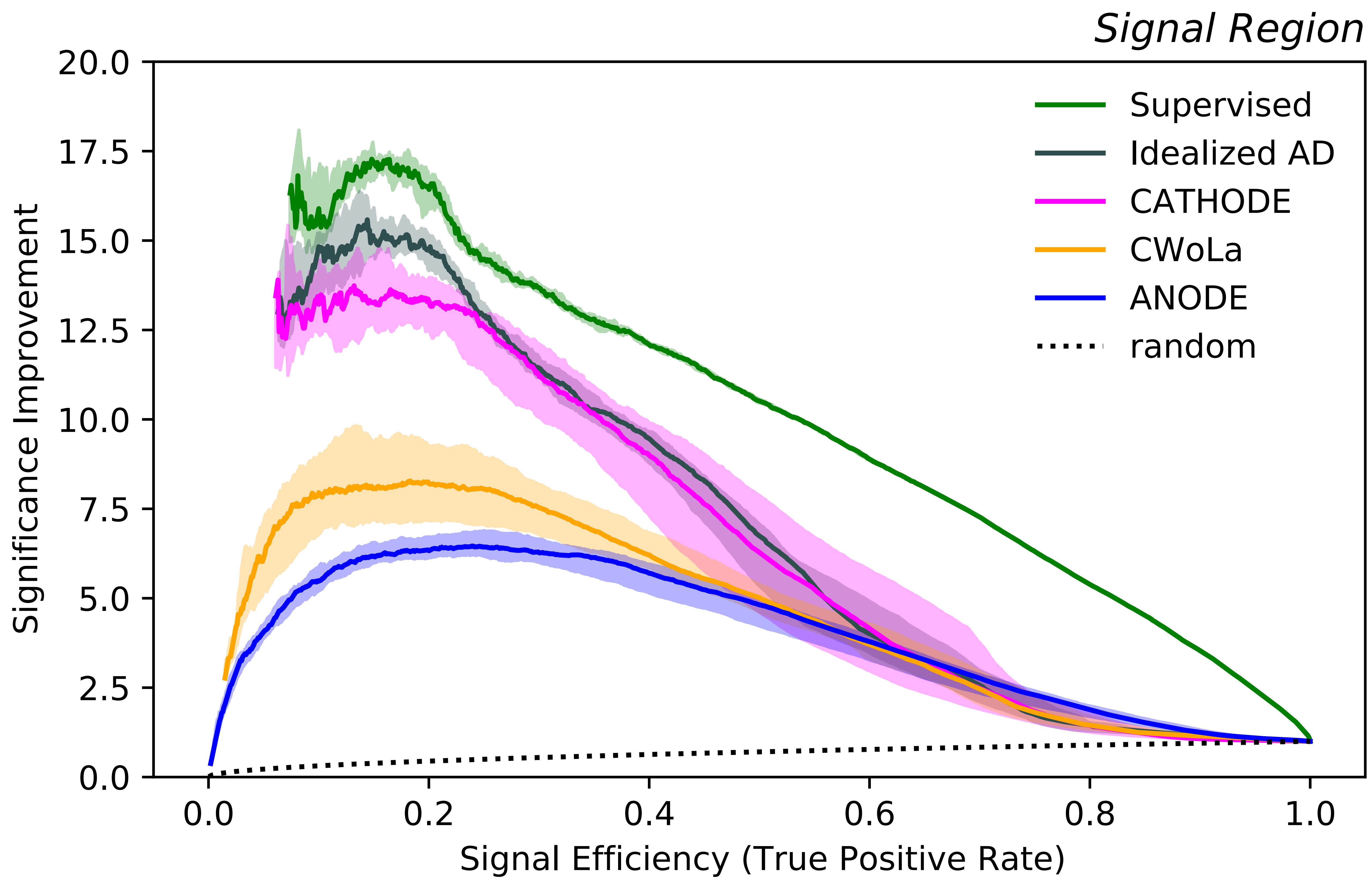
Significance improvement characteristic (SIC): $\epsilon_S/\sqrt{\epsilon_B}$



CATHODE outperforms CWoLa and ANODE and nearly saturates the idealized anomaly detector!

Comparison of methods

Significance improvement characteristic (SIC): $\epsilon_S/\sqrt{\epsilon_B}$



CATHODE outperforms CWoLa and ANODE and nearly saturates the idealized anomaly detector!

Initial significance was ~2.2σ

=> **a ~30σ anomaly could be hiding in the data right now!**

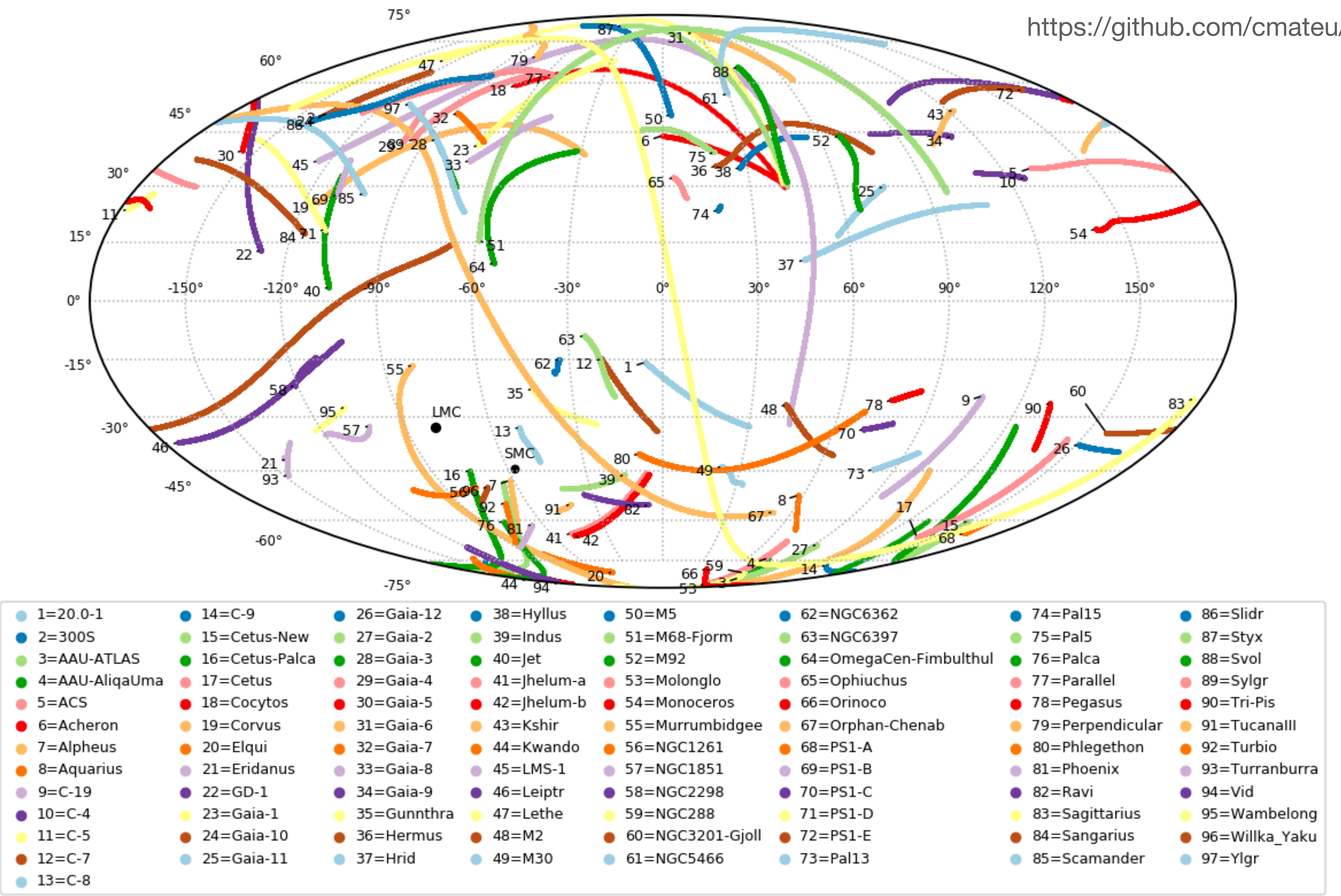
Gaia: Anomaly Detection: Via Machinae

Stellar streams are cold, tidally-stripped remnants of globular clusters and dwarf galaxies, orbiting our galaxy

Stellar streams are cold, tidally-stripped remnants of globular clusters and dwarf galaxies, orbiting our galaxy

Known Stellar Streams (candidates)

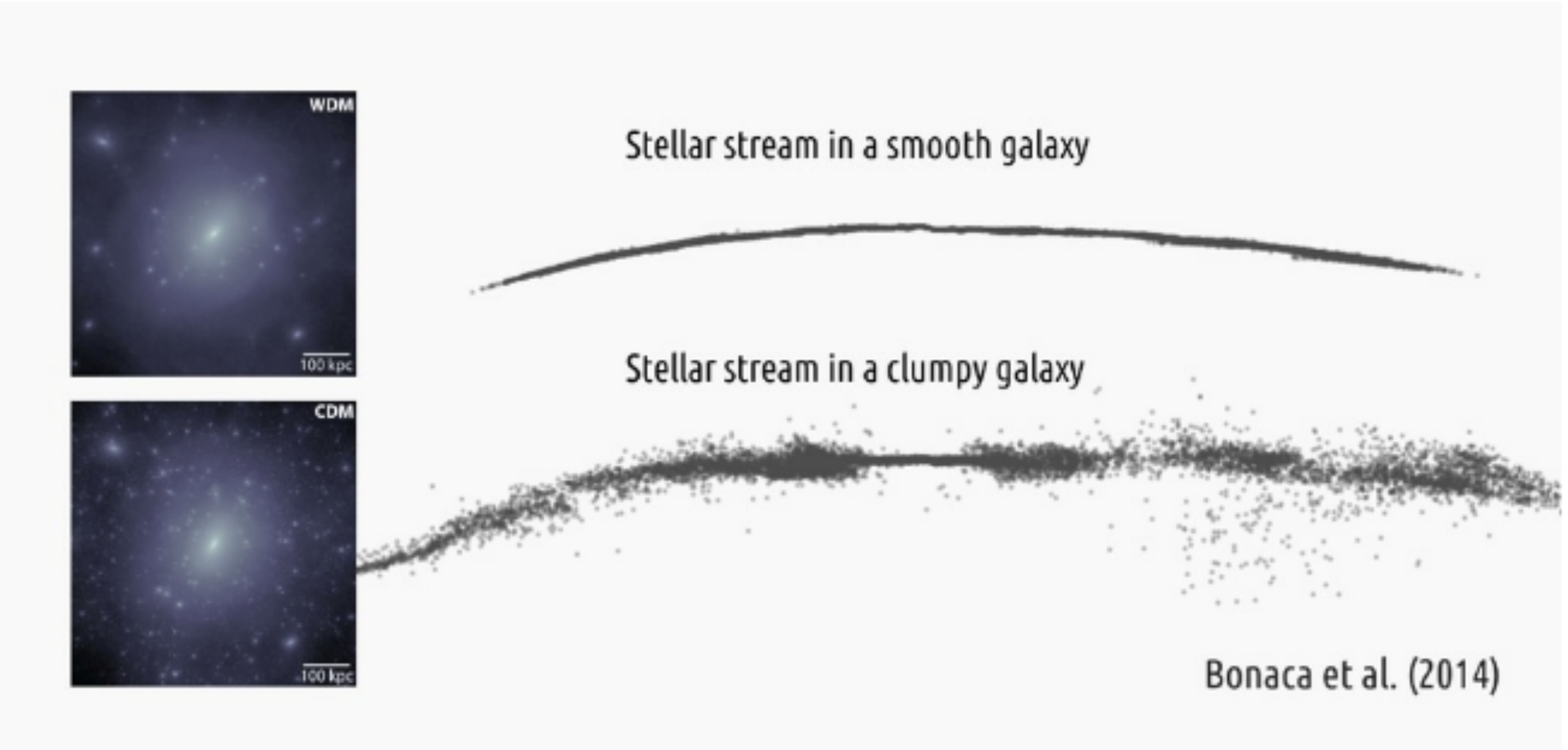
https://github.com/cmateu/galstreams



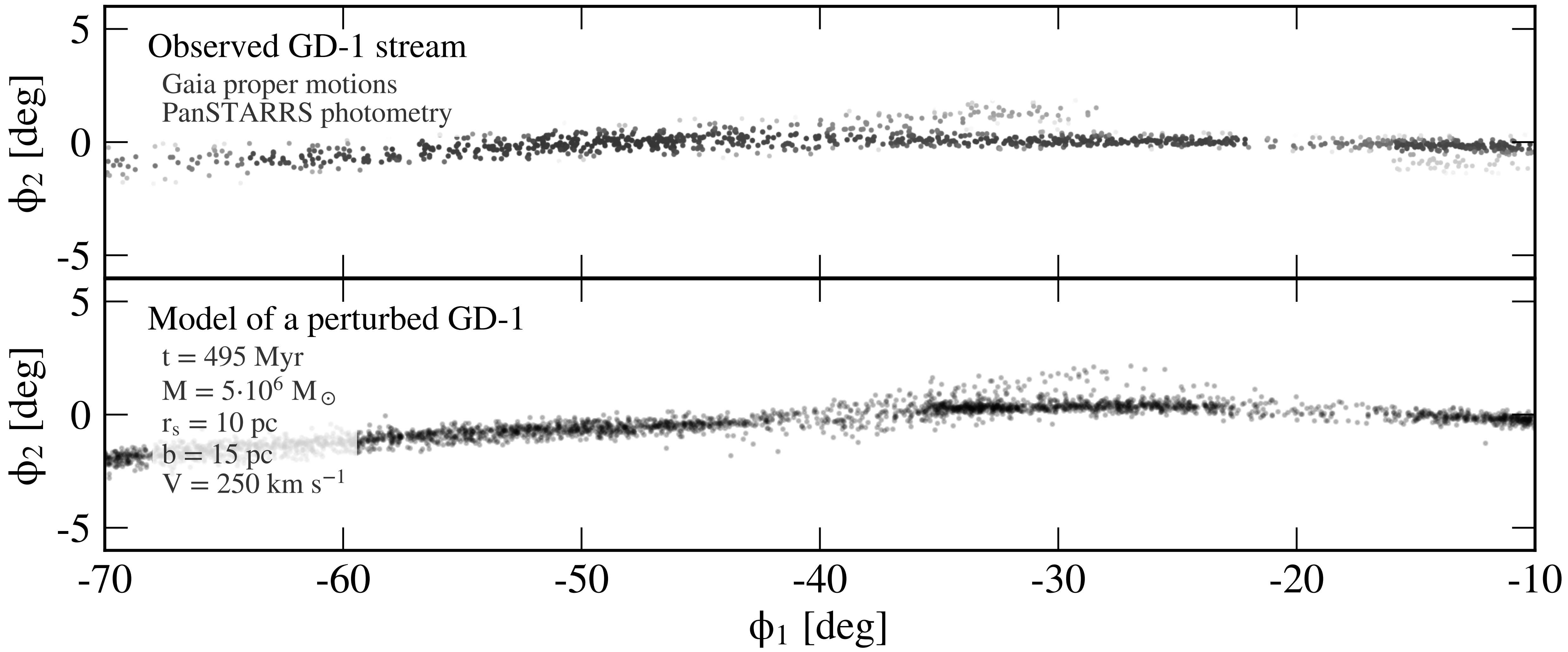
Stellar Streams and Dark Matter

They are very interesting objects of study for astrophysicists and particle physicists.

In particular, they could be unique probes into dark matter substructure.

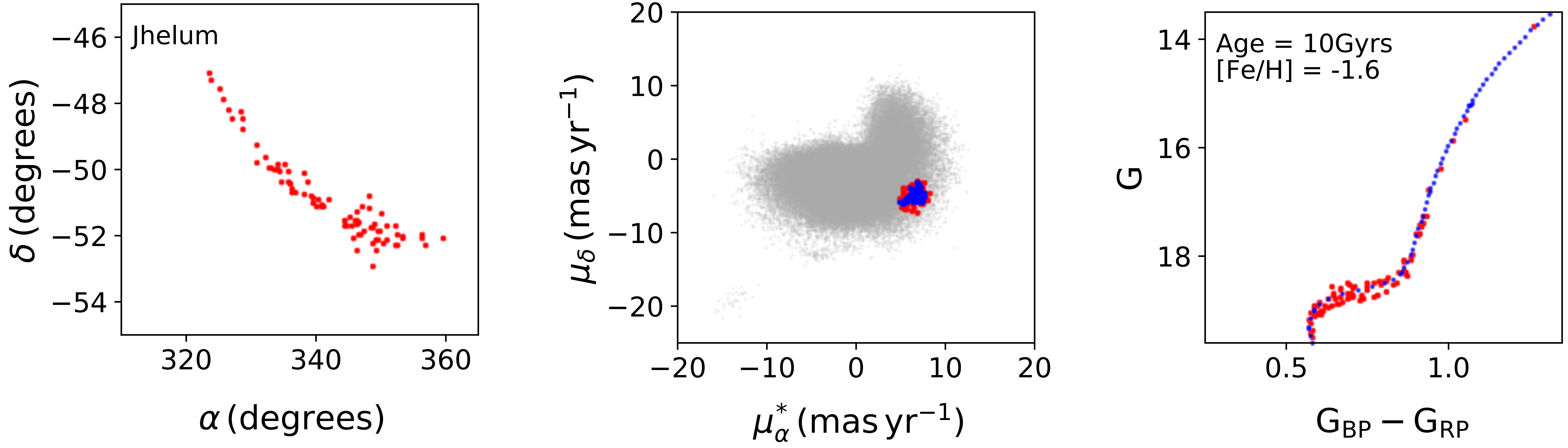


Bonaca et al 2019



Properties of stellar streams

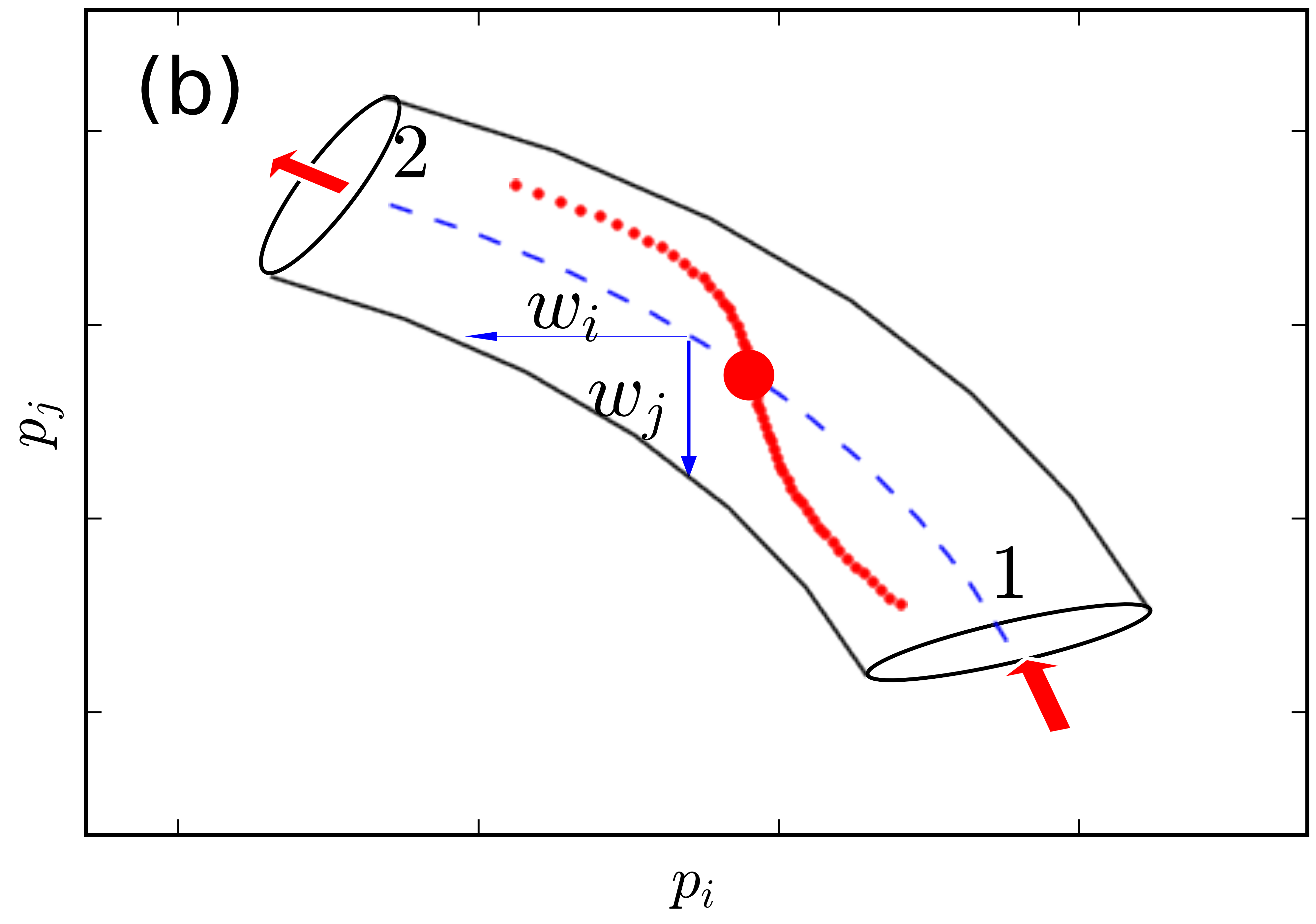
Malhan et al 2018



Streams are local overdensities in position, proper motion, and photometric space.

STREAMFINDER

Malhan & Ibata 2018

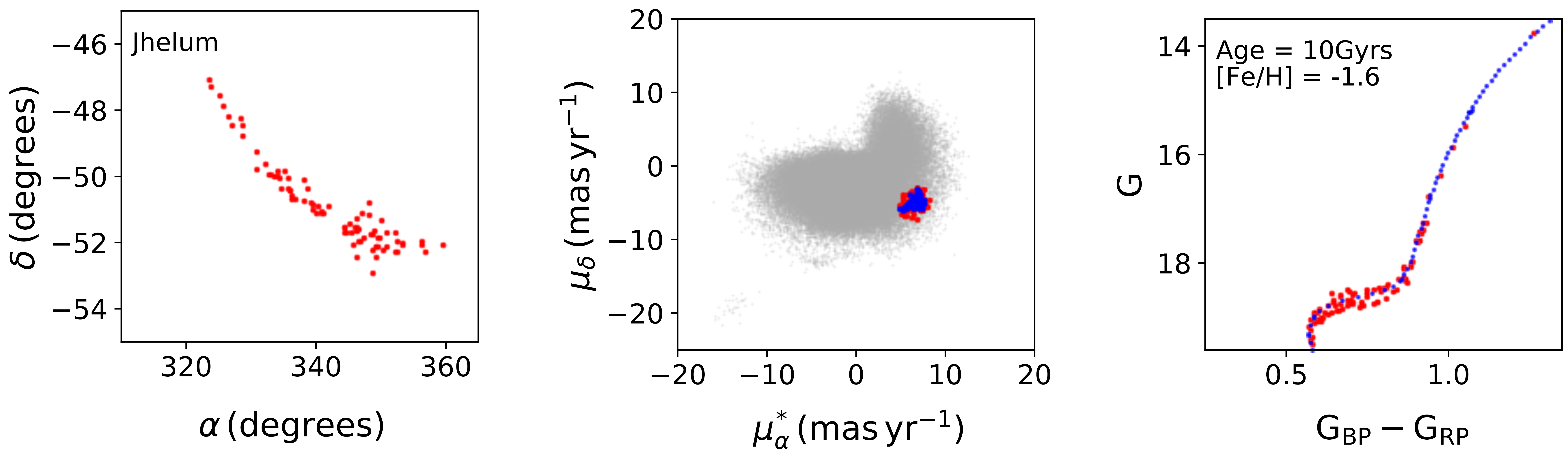


STREAMFINDER is the current state of the art and has found many new stream candidates in the Gaia data.

They assume a particular form of the Galactic potential and isochrones, and integrate trial orbits to find statistically significant groups of stars consistent with a stream.

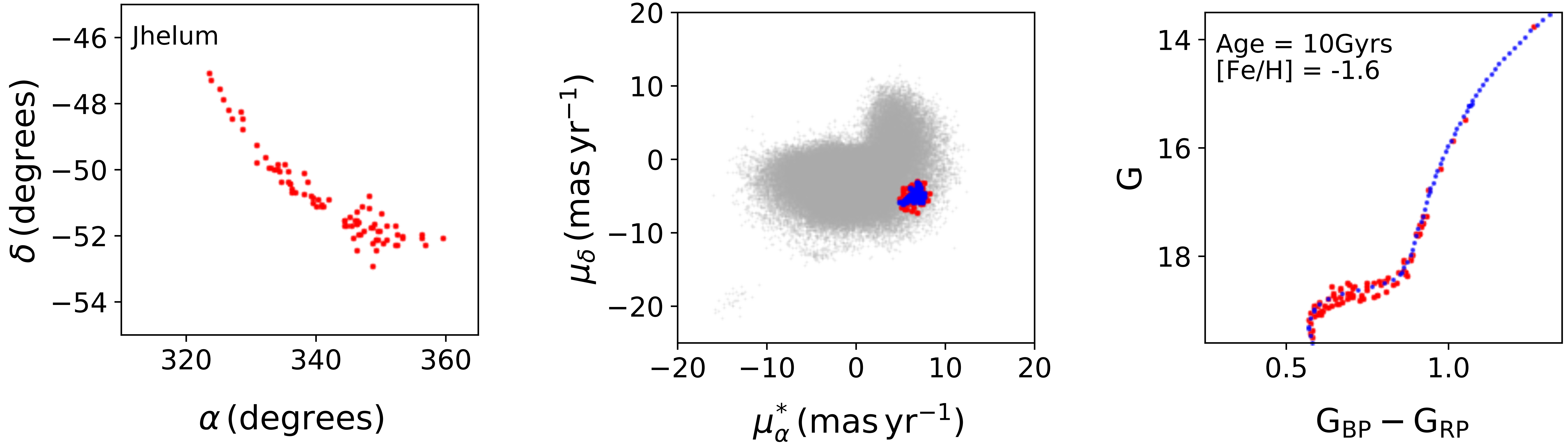
Via Machinae

DS, Buckley, Necib & Tamanas 2104.12789 and 2209.xxxxx

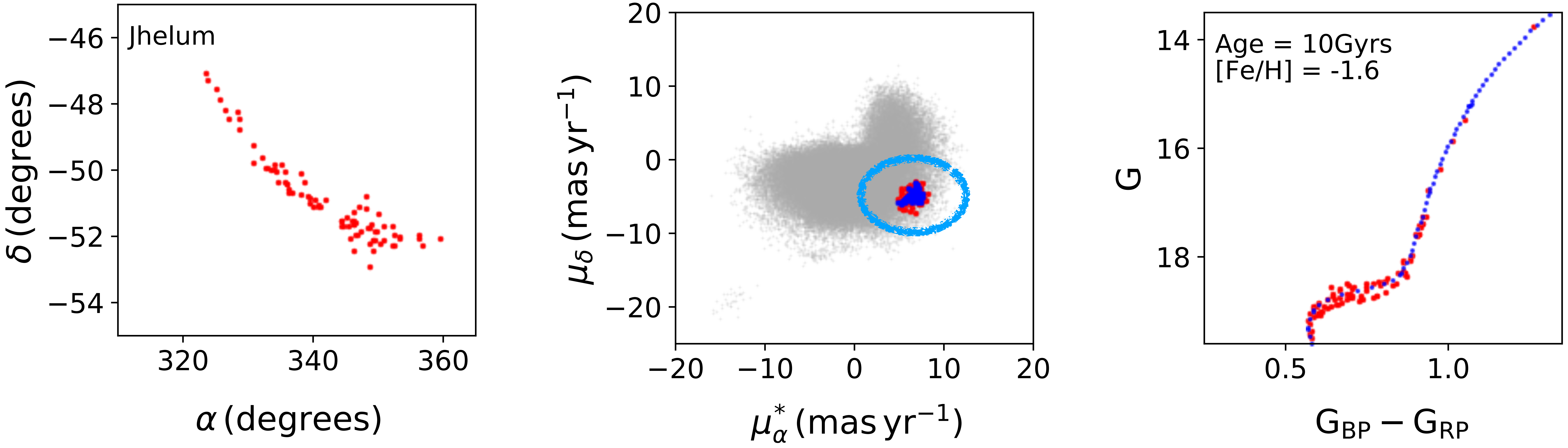


- We were interested if a more model-independent search for stellar streams could be performed, one that didn't assume anything about the form of the Galactic potential or isochrones.
- Streams are local overdensities in multiple features — ideal for **enhanced bump hunt methods**!

Multivariate Bump Hunting…in Space!

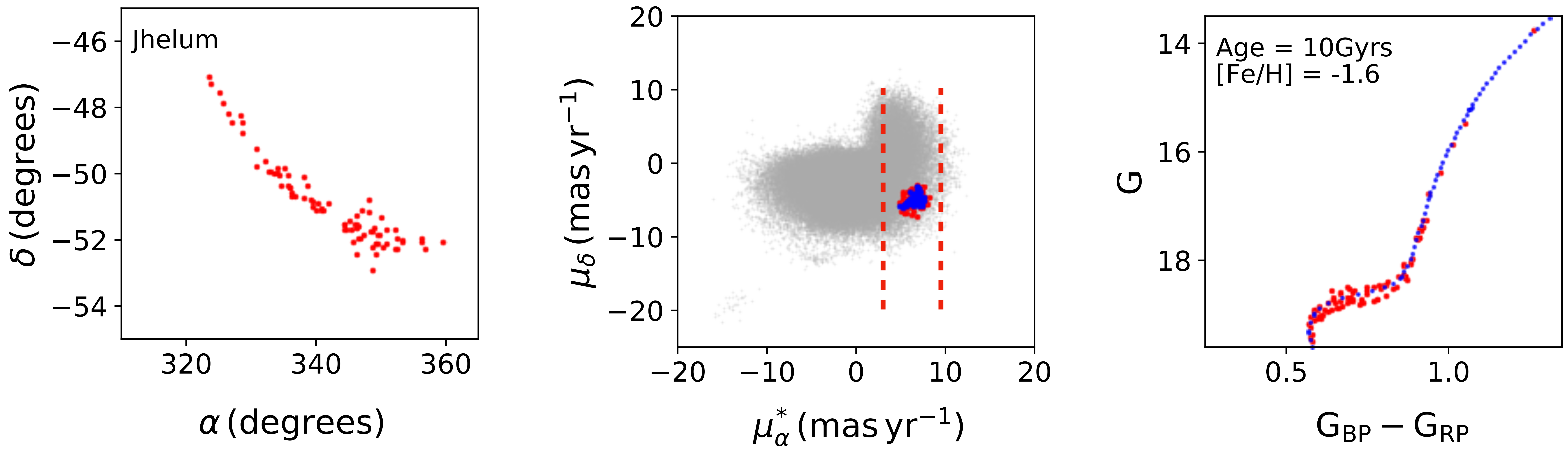


Multivariate Bump Hunting…in Space!



Since they are **cold**, the stars in the stream are fully localized in both proper motions.

Multivariate Bump Hunting…in Space!



Since they are **cold**, the stars in the stream are fully localized in both proper motions.

Can sideband in either one of the proper motions and use enhanced bump hunt methods to look for local overdensities!

Multivariate Bump Hunting…in Space!

- General idea: train enhanced bump hunt method with $m = \mu_\alpha$ (for example) and $x = (\mu_\delta, \alpha, \delta, G, B - R)$
    - ANODE in space:
        - DS, Buckley, Necib & Tamanas (2104.12789 and 2208.xxxxx)
    - CWoLa in space:
        - Thanvantri, Pettee, Nachman, DS, Buckley, Collins … (22xx.xxxxx)
    - CATHODE in space:
        - Hallin, Krause, DS, Buckley, … (22xx.xxxxx)

Multivariate Bump Hunting…in Space!

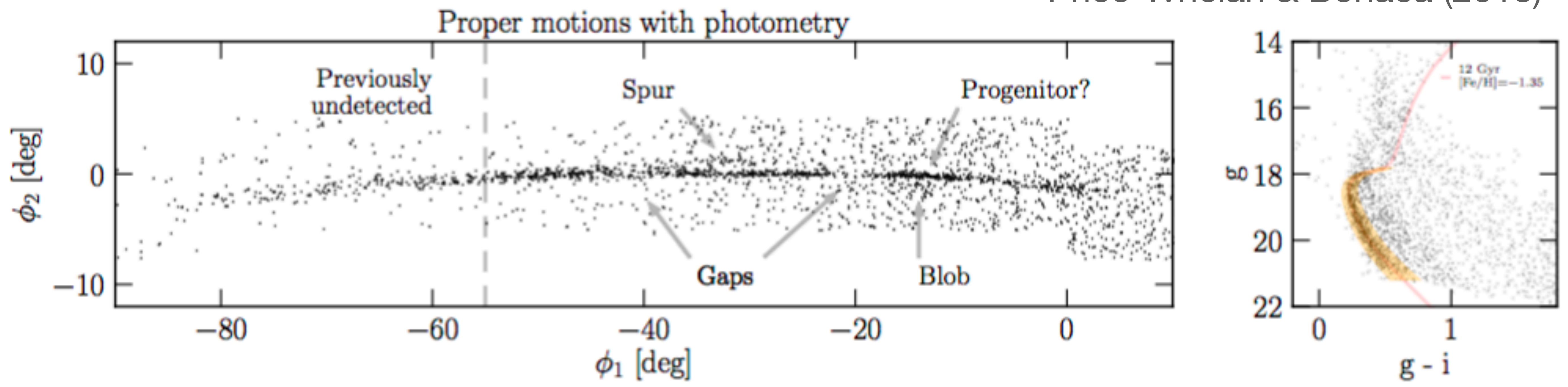
- General idea: train enhanced bump hunt method with $m = \mu_\alpha$ (for example) and $x = (\mu_\delta, \alpha, \delta, G, B - R)$
    - **ANODE in space:**
        - **DS, Buckley, Necib & Tamanas (2104.12789 and 2209.xxxxx)**
    - CWoLa in space:
        - Thanvantri, Pettee, Nachman, DS, Buckley, Collins … (22xx.xxxxx)
    - CATHODE in space:
        - Hallin, Krause, DS, Buckley, … (22xx.xxxxx)

Example: GD-1

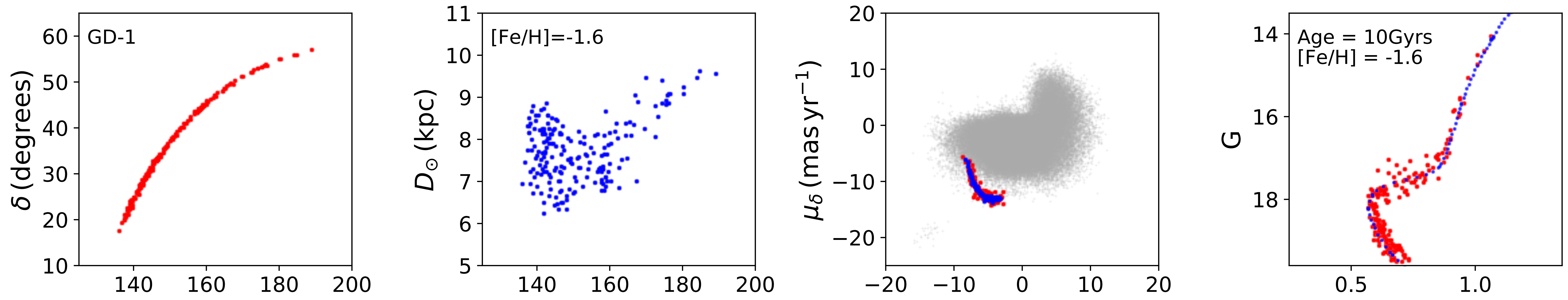
- GD-1 is one of the most prominent and well-studied cold stellar streams

  - ~2000 stars stretching across ~100˚ of the sky
  - Age ~10 Gyr, distance ~8 kpc
  - Discovered by Grillmair & Dionatos in SDSS data (2006)
  - Many subsequent studies with Gaia and other instruments

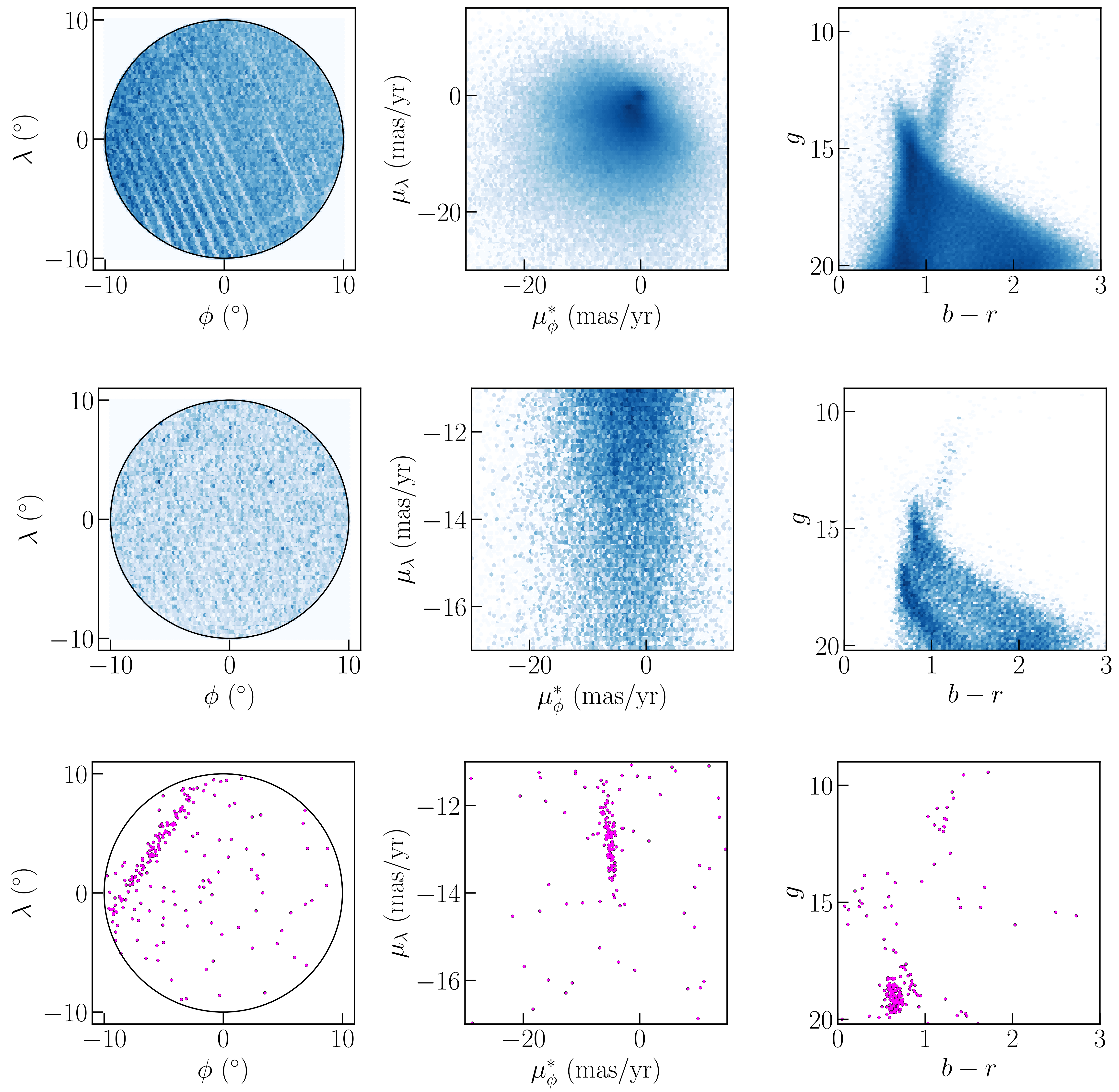
Price-Whelan & Bonaca (2018)



STREAMFINDER, Malhan et al (2018)



GD-1 with ANODE

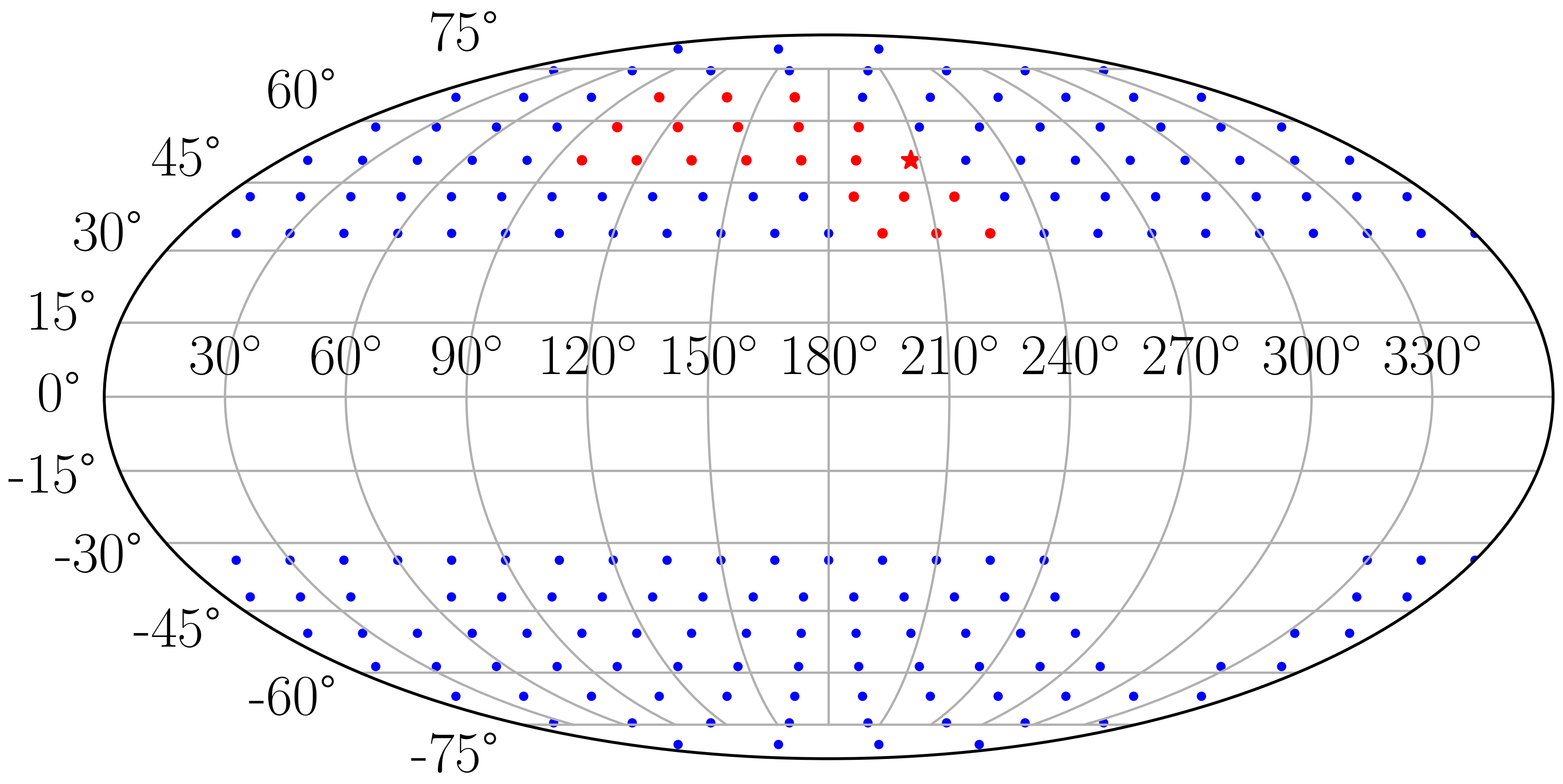


All stars in a patch of the sky containing (part of) GD-1 (ra,dec)=(148.6,24.2)

All stars in a SR containing GD-1 $\mu_\lambda \in [-17, -11]$

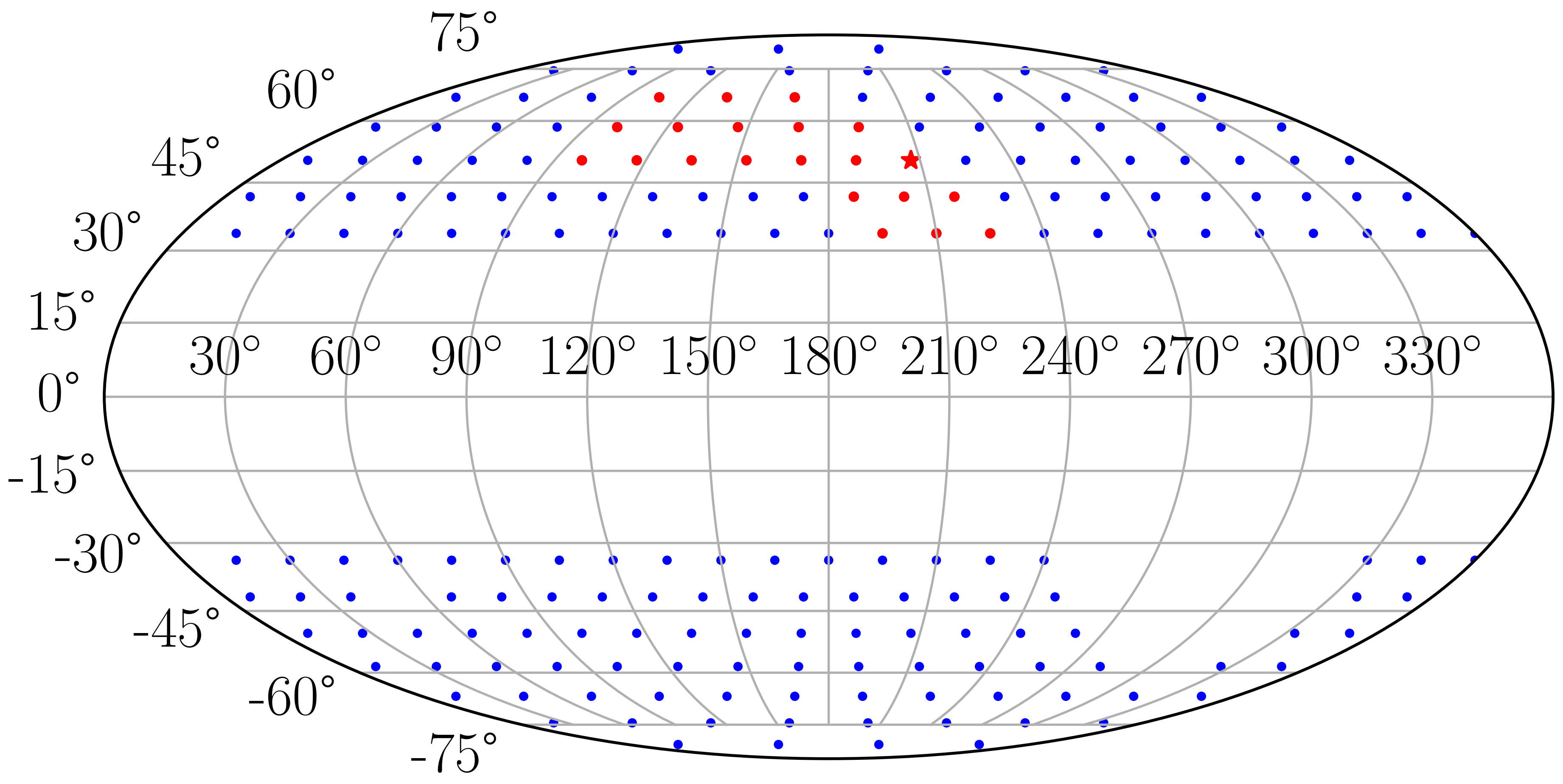
Stars in SR after cut on R(x) obtained from ANODE

ANODE on Gaia data



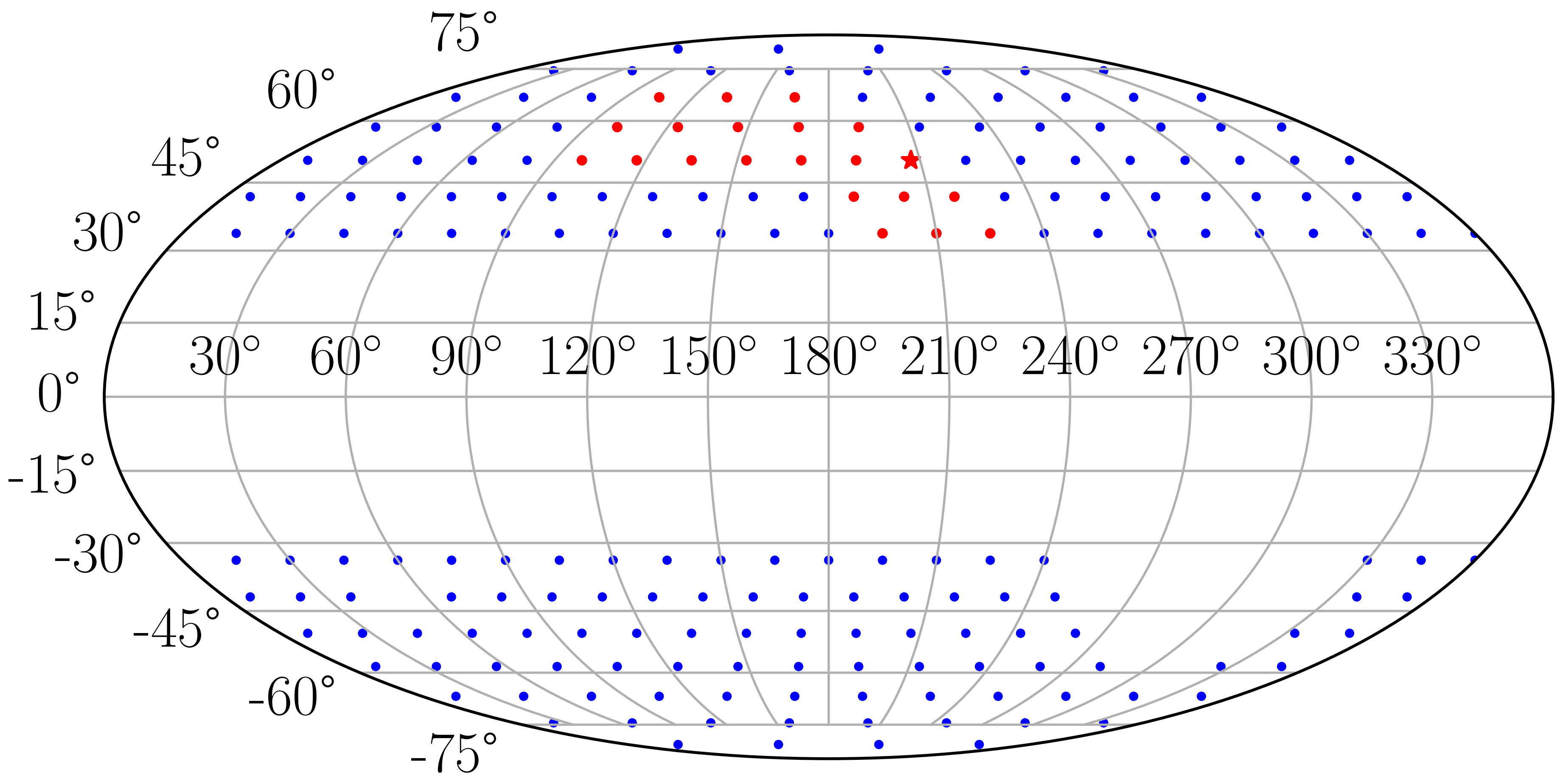
ANODE on Gaia data

- So the method works!

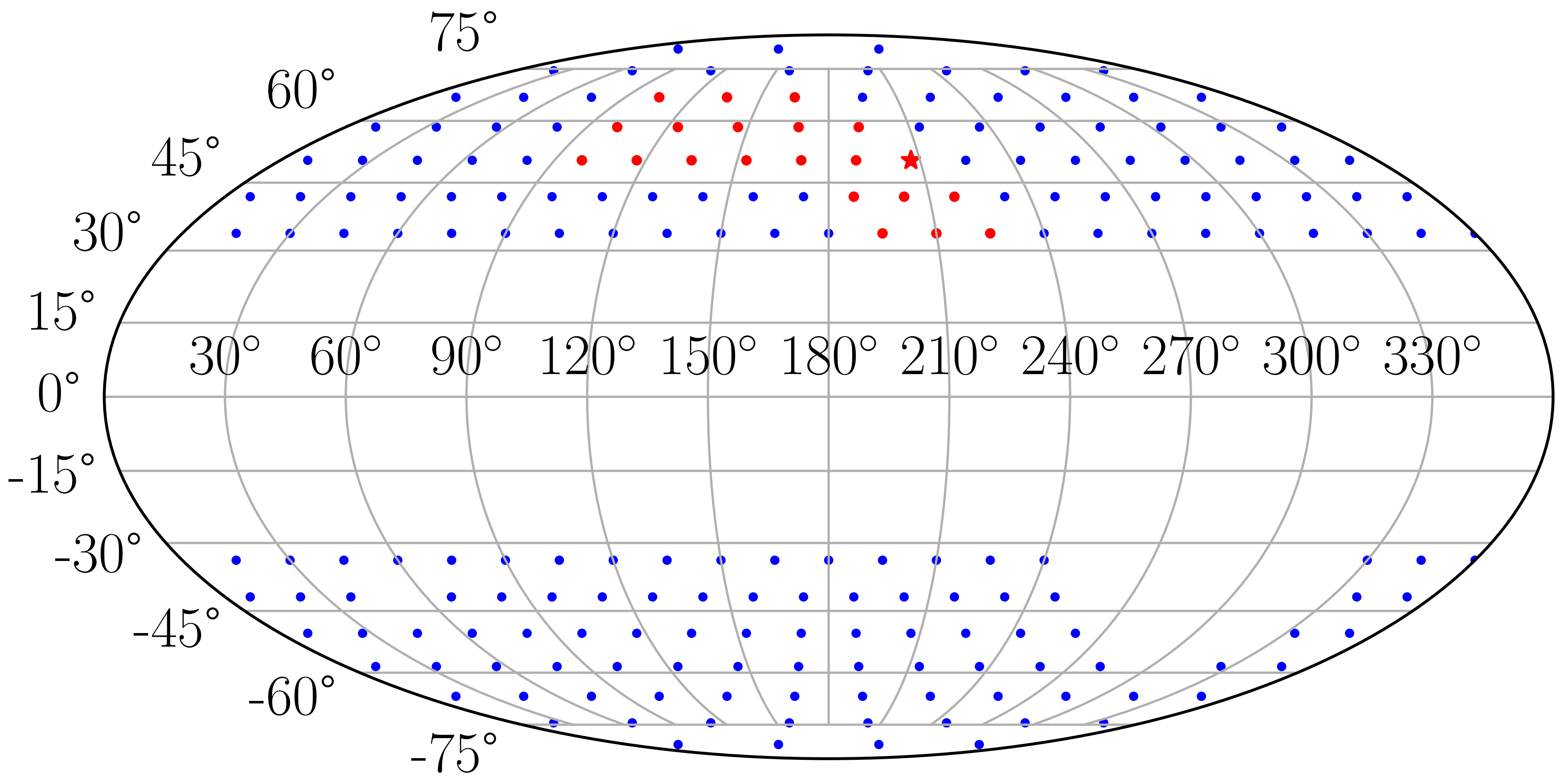


ANODE on Gaia data

- So the method works!
- How do we apply this to the entire sky?

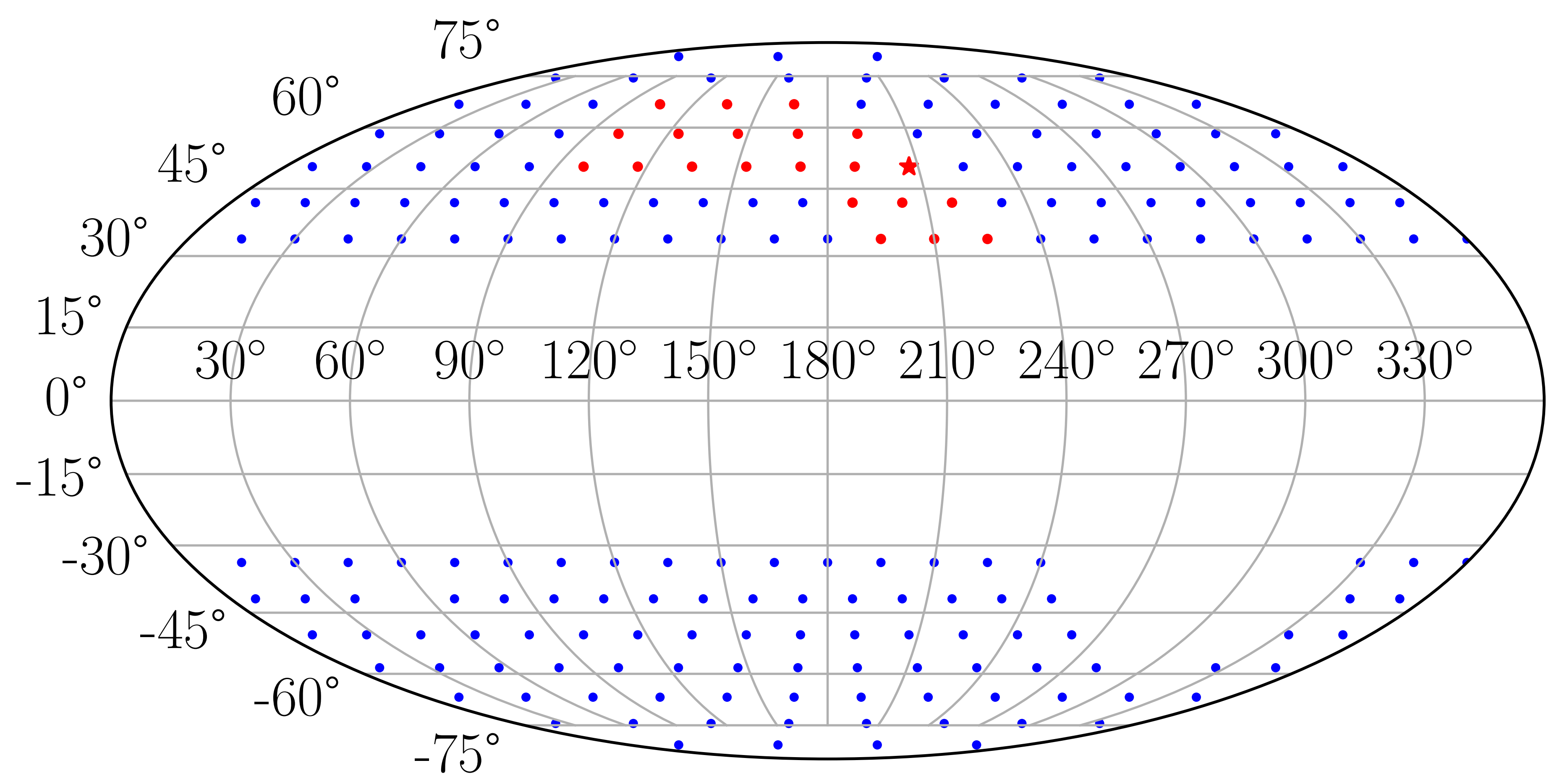


ANODE on Gaia data



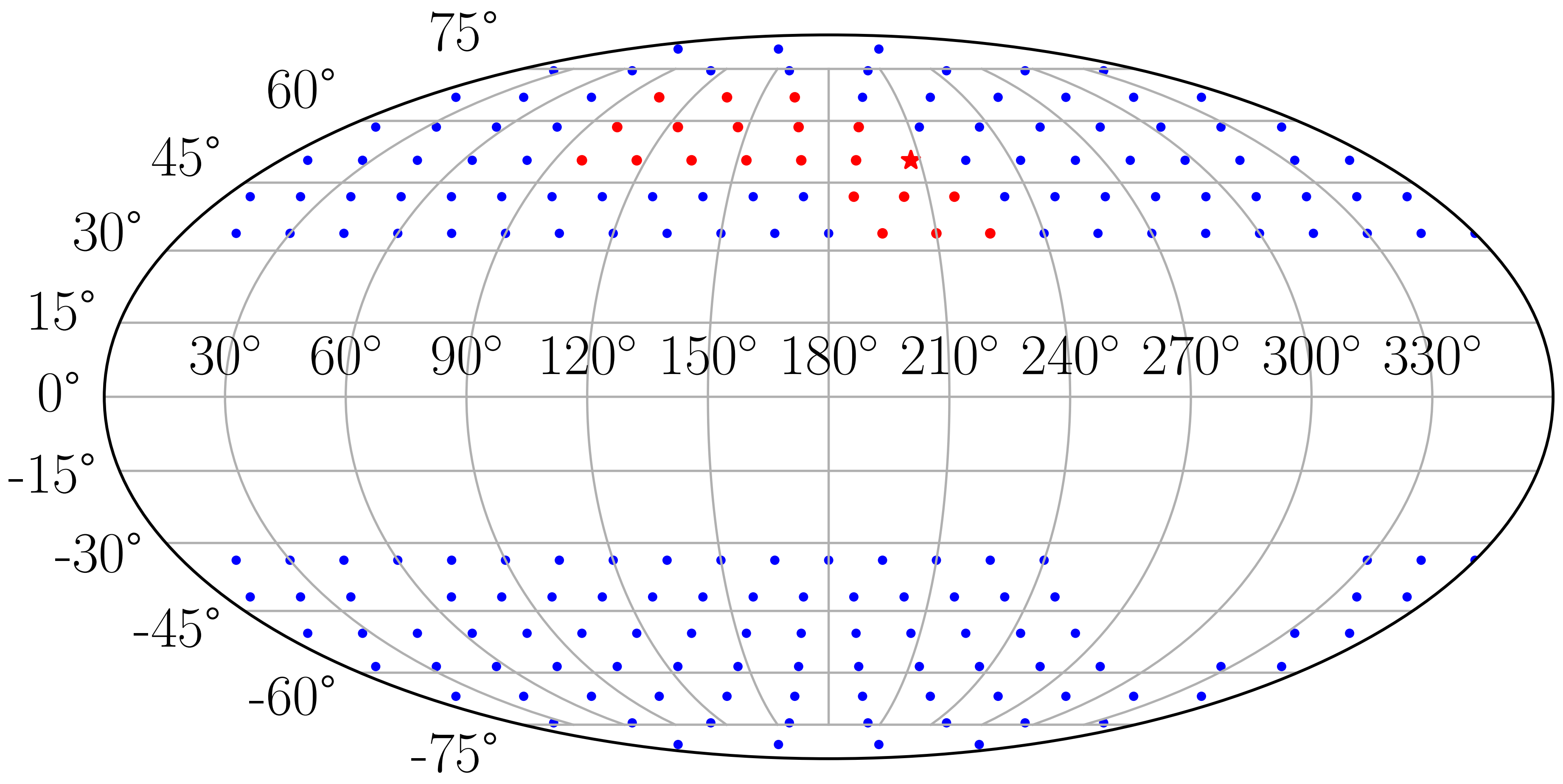
- So the method works!
- How do we apply this to the entire sky?
  - Divide sky up into overlapping 15˚ circular patches

ANODE on Gaia data



- So the method works!
- How do we apply this to the entire sky?
  - Divide sky up into overlapping 15˚ circular patches
  - Exclude patches too close to the disk (too many stars) and overlapping with the LMC and SMC

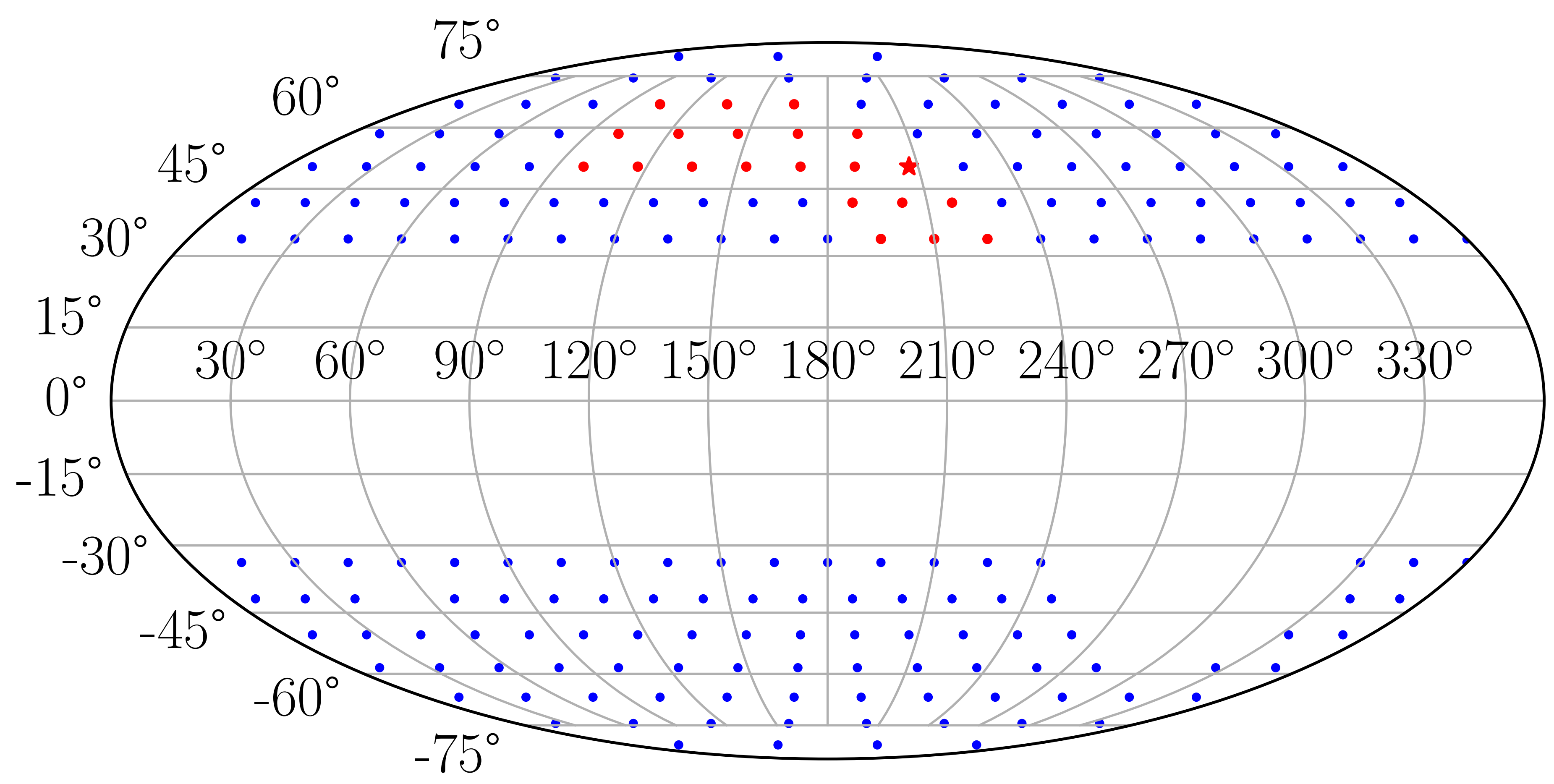
ANODE on Gaia data



- So the method works!
- How do we apply this to the entire sky?
  - Divide sky up into overlapping 15˚ circular patches
  - Exclude patches too close to the disk (too many stars) and overlapping with the LMC and SMC

→ 200 patches

ANODE on Gaia data

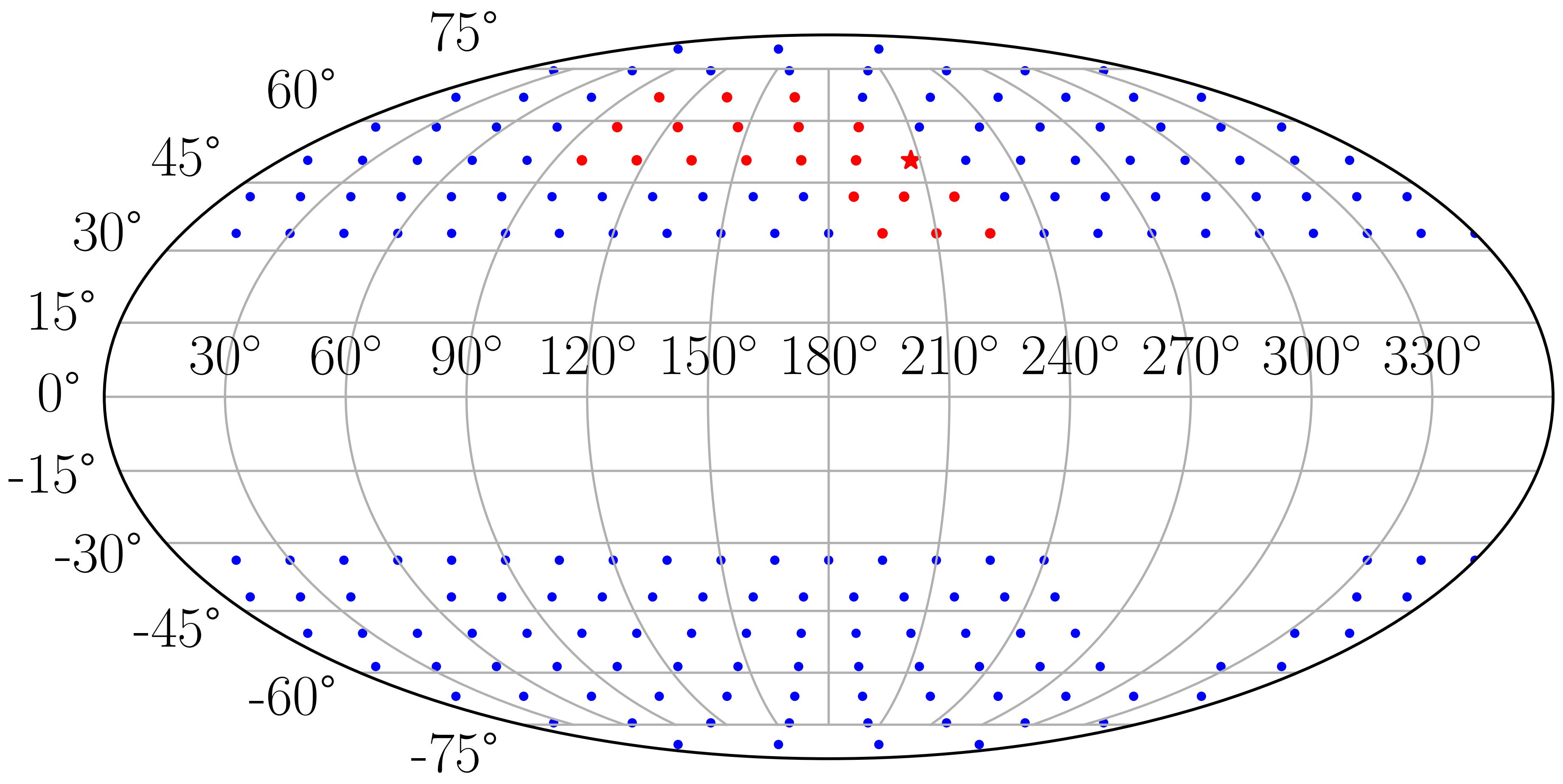


- So the method works!
- How do we apply this to the entire sky?
  - Divide sky up into overlapping 15˚ circular patches
  - Exclude patches too close to the disk (too many stars) and overlapping with the LMC and SMC

200 patches

  - Divide up each patch into overlapping SRs (width 6) in μ_α, train ANODE

ANODE on Gaia data

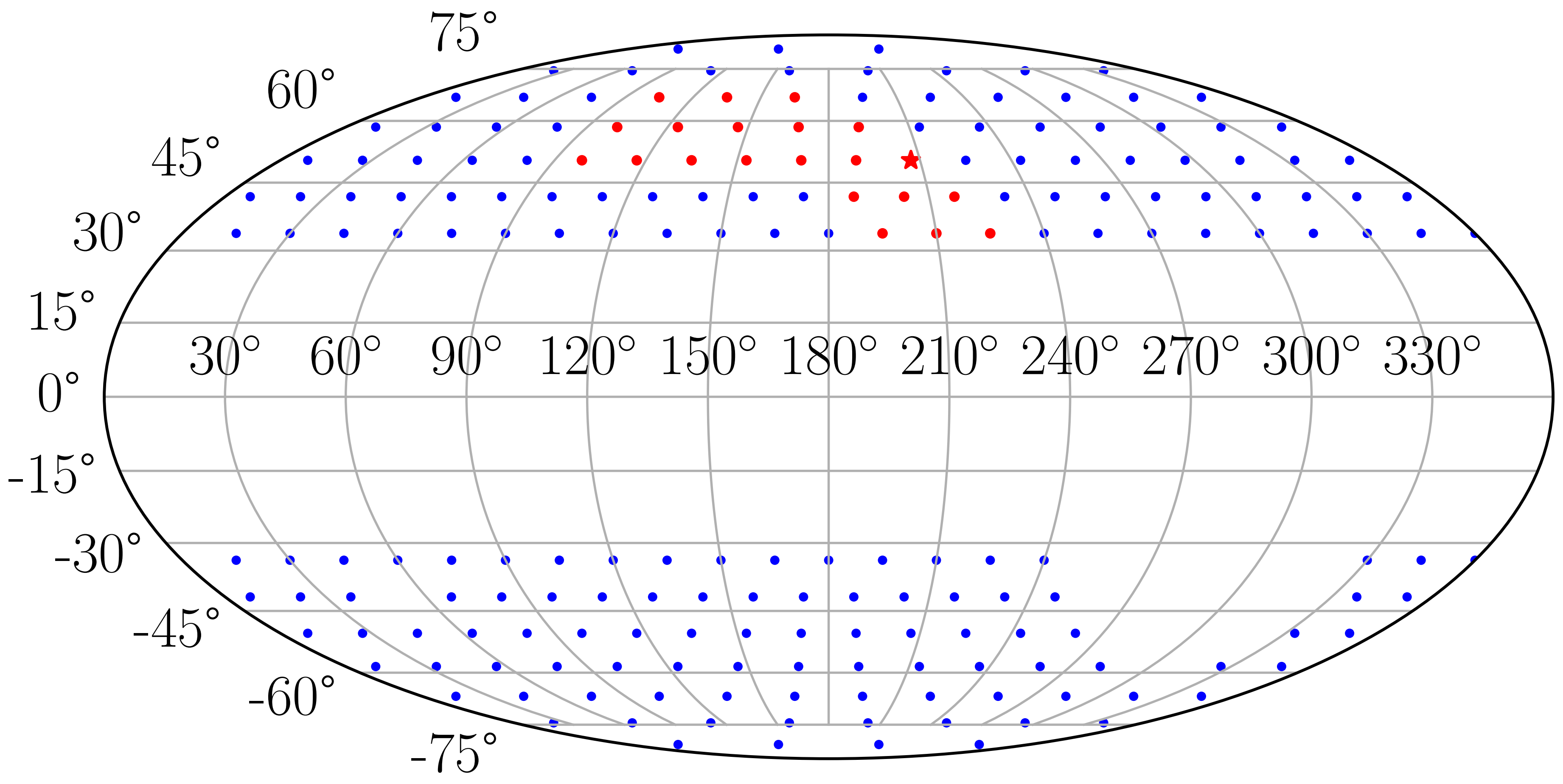


- So the method works!
- How do we apply this to the entire sky?
  - Divide sky up into overlapping 15° circular patches
  - Exclude patches too close to the disk (too many stars) and overlapping with the LMC and SMC

→ 200 patches

  - Divide up each patch into overlapping SRs (width 6) in μ_α, train ANODE → ~30 SRs / patches

ANODE on Gaia data

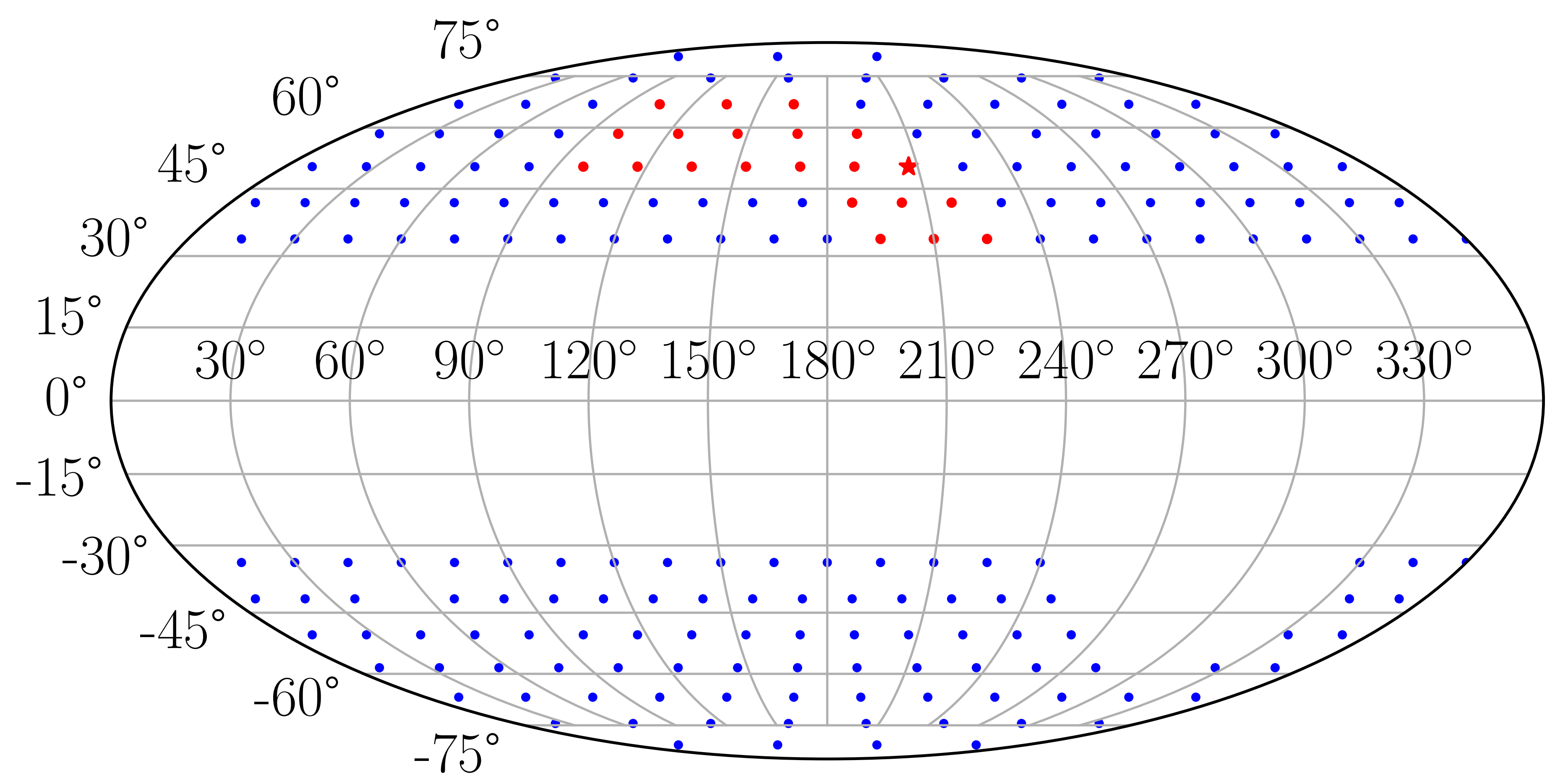


- So the method works!
- How do we apply this to the entire sky?
  - Divide sky up into overlapping 15˚ circular patches
  - Exclude patches too close to the disk (too many stars) and overlapping with the LMC and SMC

→ 200 patches

  - Divide up each patch into overlapping SRs (width 6) in μ_α, train ANODE → ~30 SRs / patches
  - Repeat for SRs in orthogonal proper motion coordinate μ_δ

ANODE on Gaia data

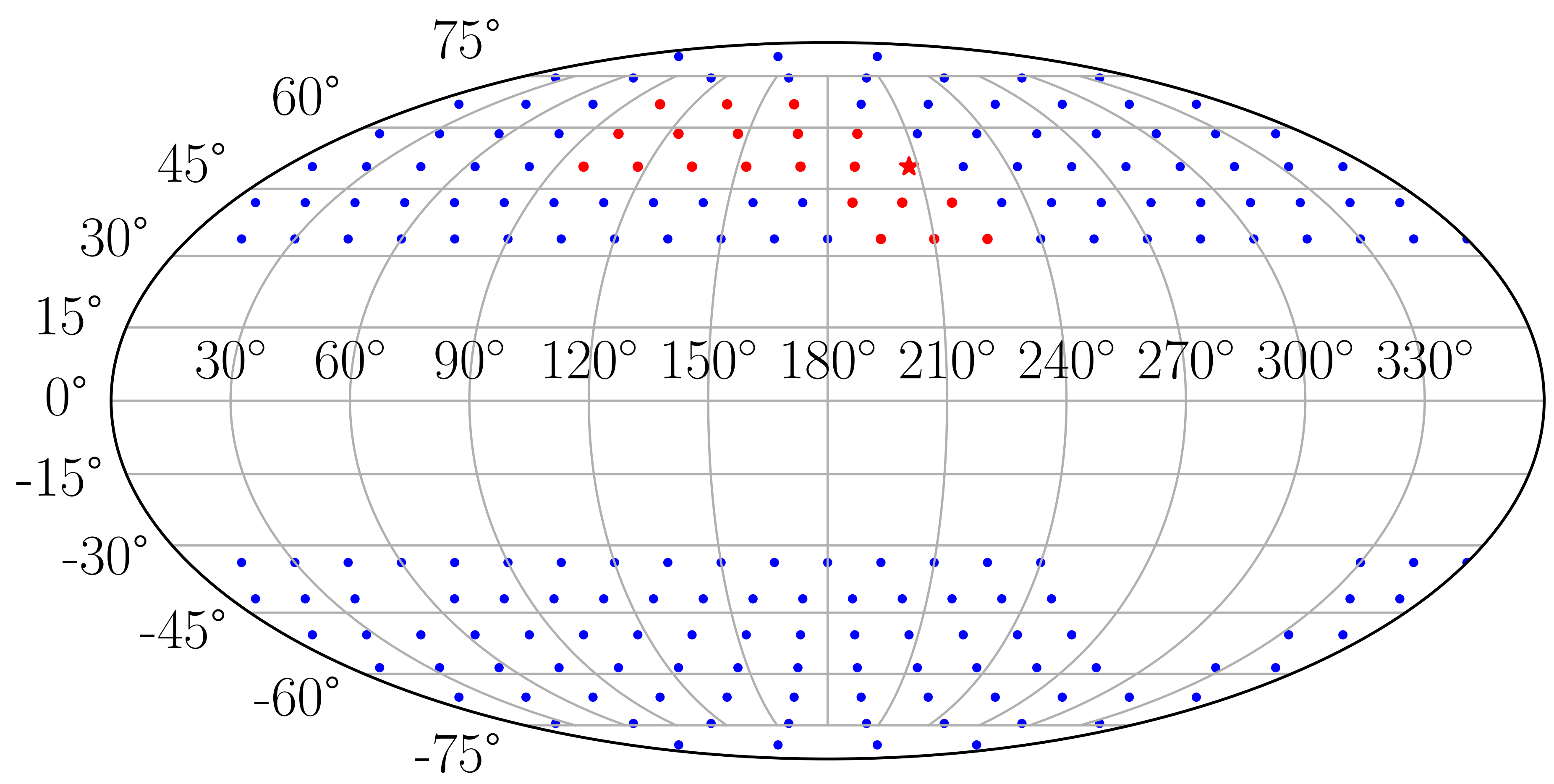


- So the method works!
- How do we apply this to the entire sky?
    - Divide sky up into overlapping 15˚ circular patches
    - Exclude patches too close to the disk (too many stars) and overlapping with the LMC and SMC

→ 200 patches

- Divide up each patch into overlapping SRs (width 6) in μ_α, train ANODE → ~30 SRs / patches
- Repeat for SRs in orthogonal proper motion coordinate μ_δ → x 2

ANODE on Gaia data



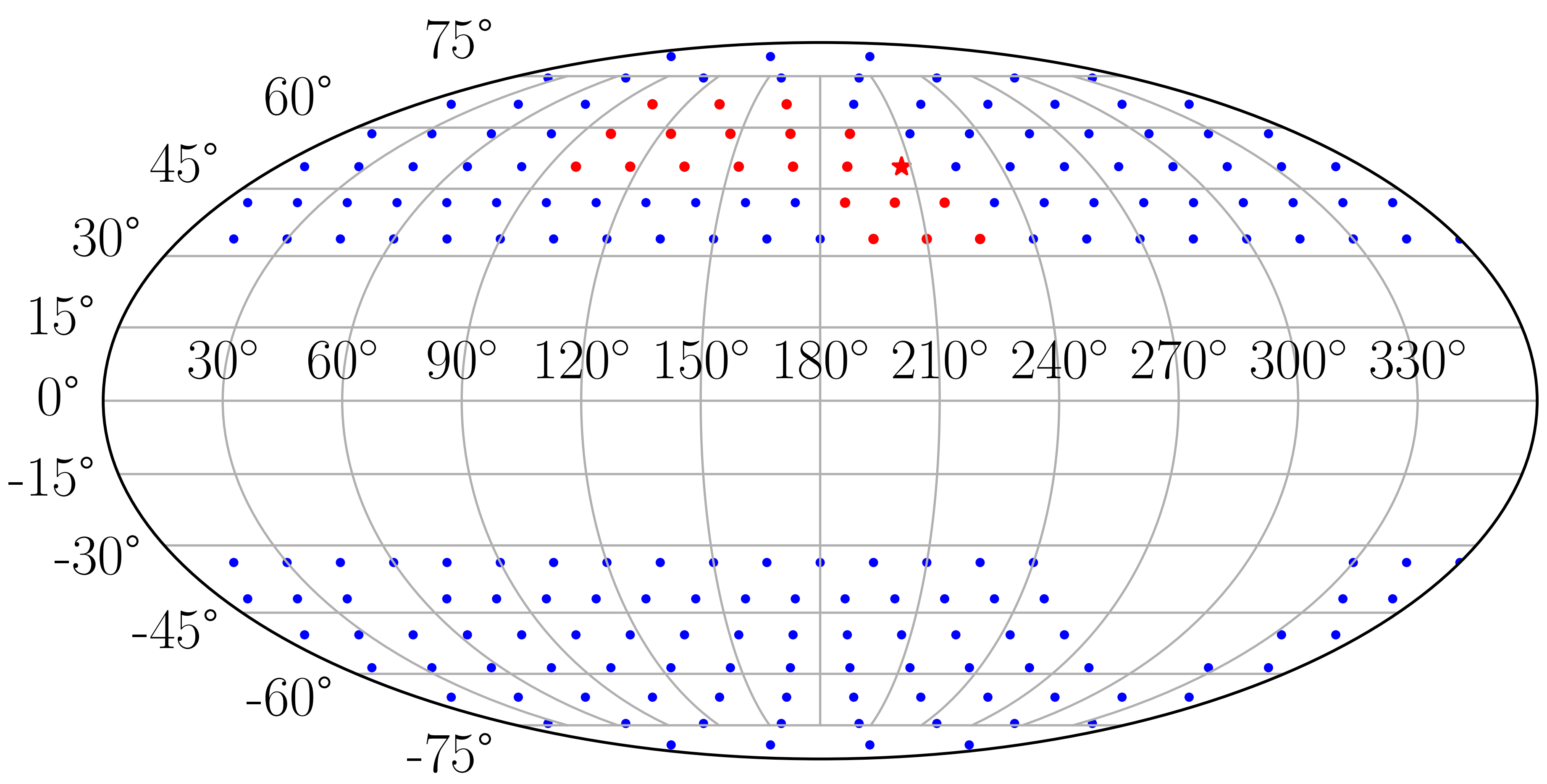
- So the method works!
- How do we apply this to the entire sky?
  - Divide sky up into overlapping 15˚ circular patches
  - Exclude patches too close to the disk (too many stars) and overlapping with the LMC and SMC

→ 200 patches

  - Divide up each patch into overlapping SRs (width 6) in μ_α, train ANODE → ~30 SRs / patches
  - Repeat for SRs in orthogonal proper motion coordinate μ_δ → x 2

Need to train ANODE ~12,000 times to cover the entire sky!!
Each training takes O(10h)

ANODE on Gaia data



- So the method works!
- How do we apply this to the entire sky?
    - Divide sky up into overlapping 15° circular patches
    - Exclude patches too close to the disk (too many stars) and overlapping with the LMC and SMC

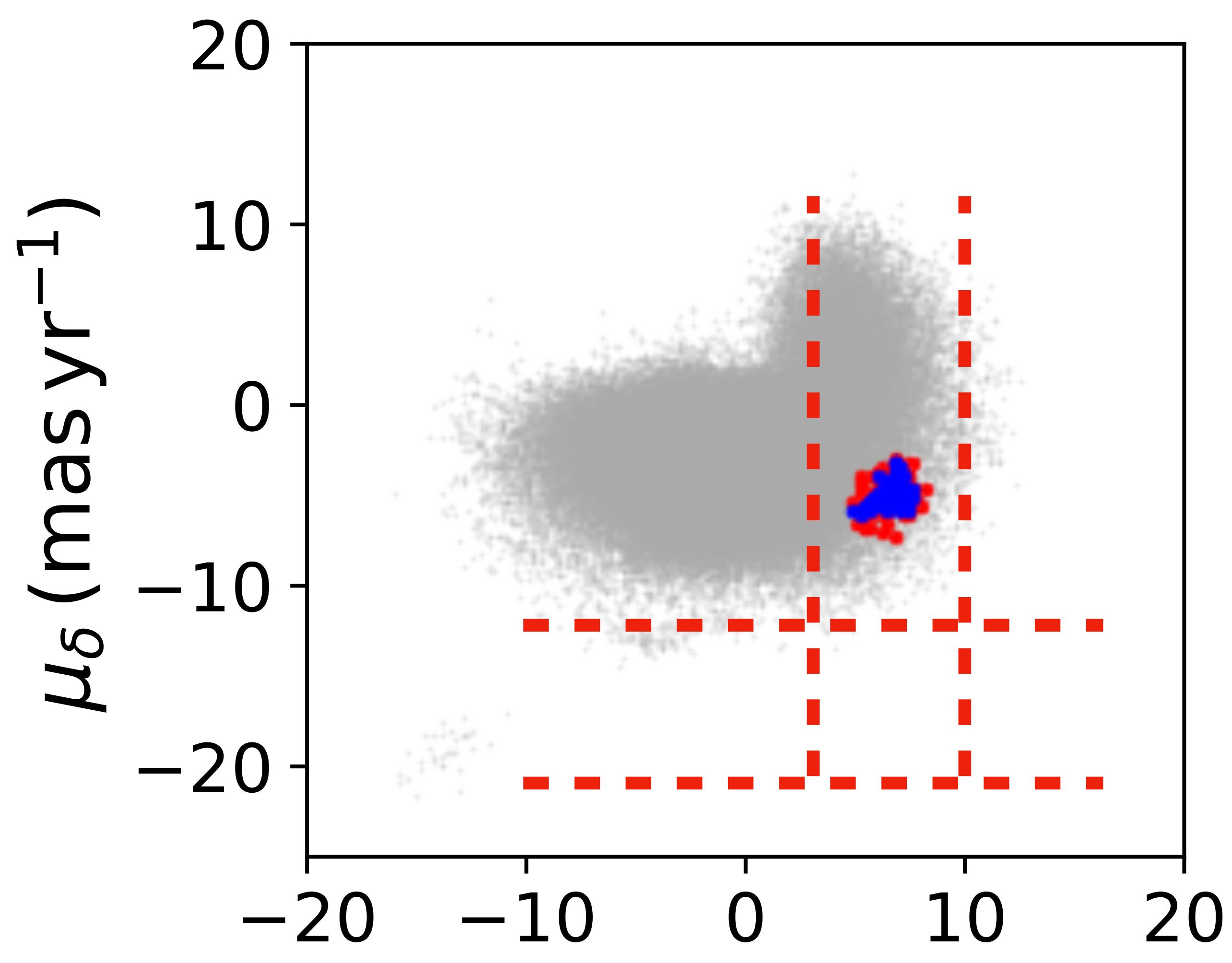
→ 200 patches

- Divide up each patch into overlapping SRs (width 6) in μ_α, train ANODE → ~30 SRs / patches
- Repeat for SRs in orthogonal proper motion coordinate μ_δ → x 2

Need to train ANODE ~12,000 times to cover the entire sky!!
Each training takes O(10h)

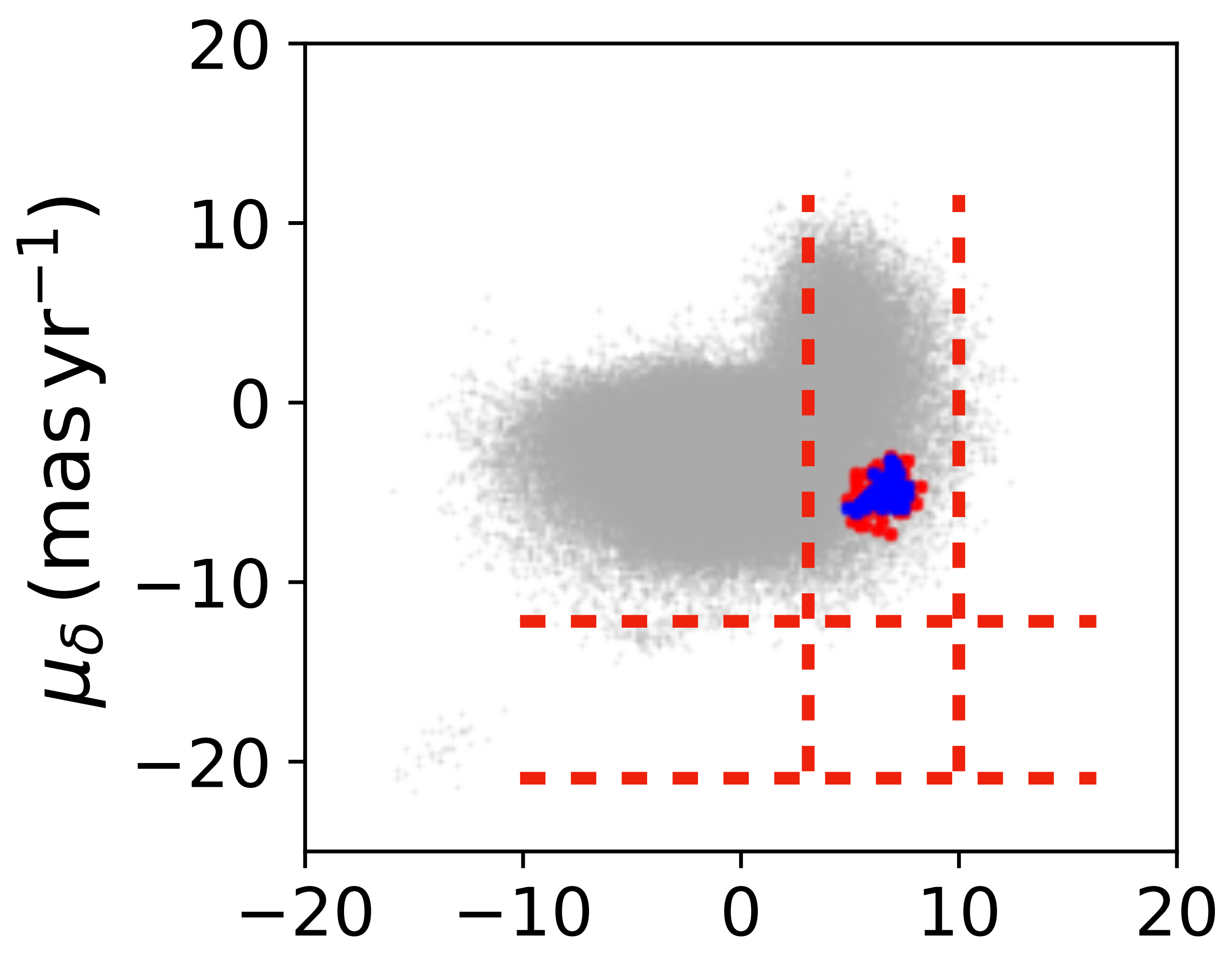
We ran this on the NERSC supercomputer at LBNL

ANODE on Gaia data



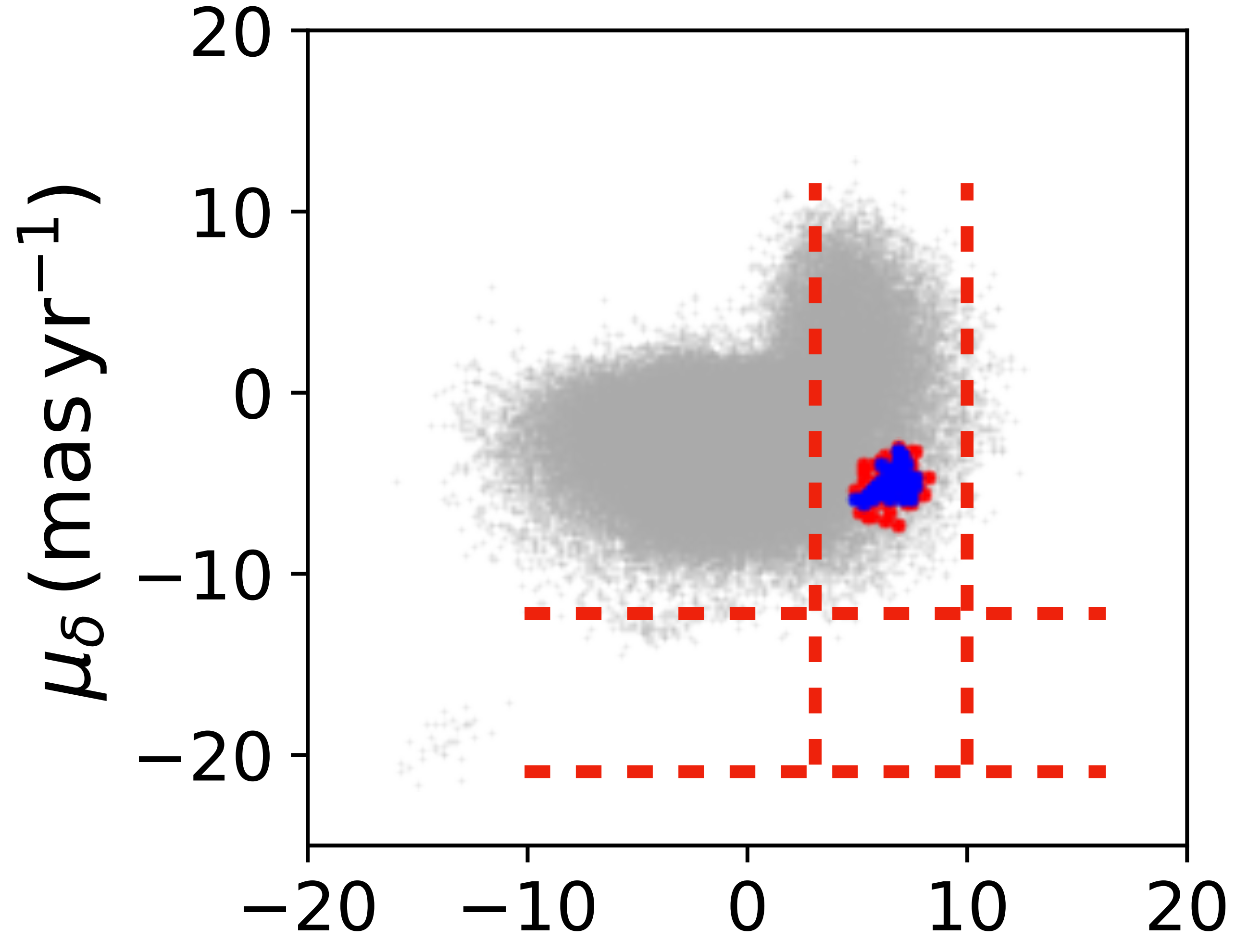
ANODE on Gaia data

- How to set the cut on R(x)?



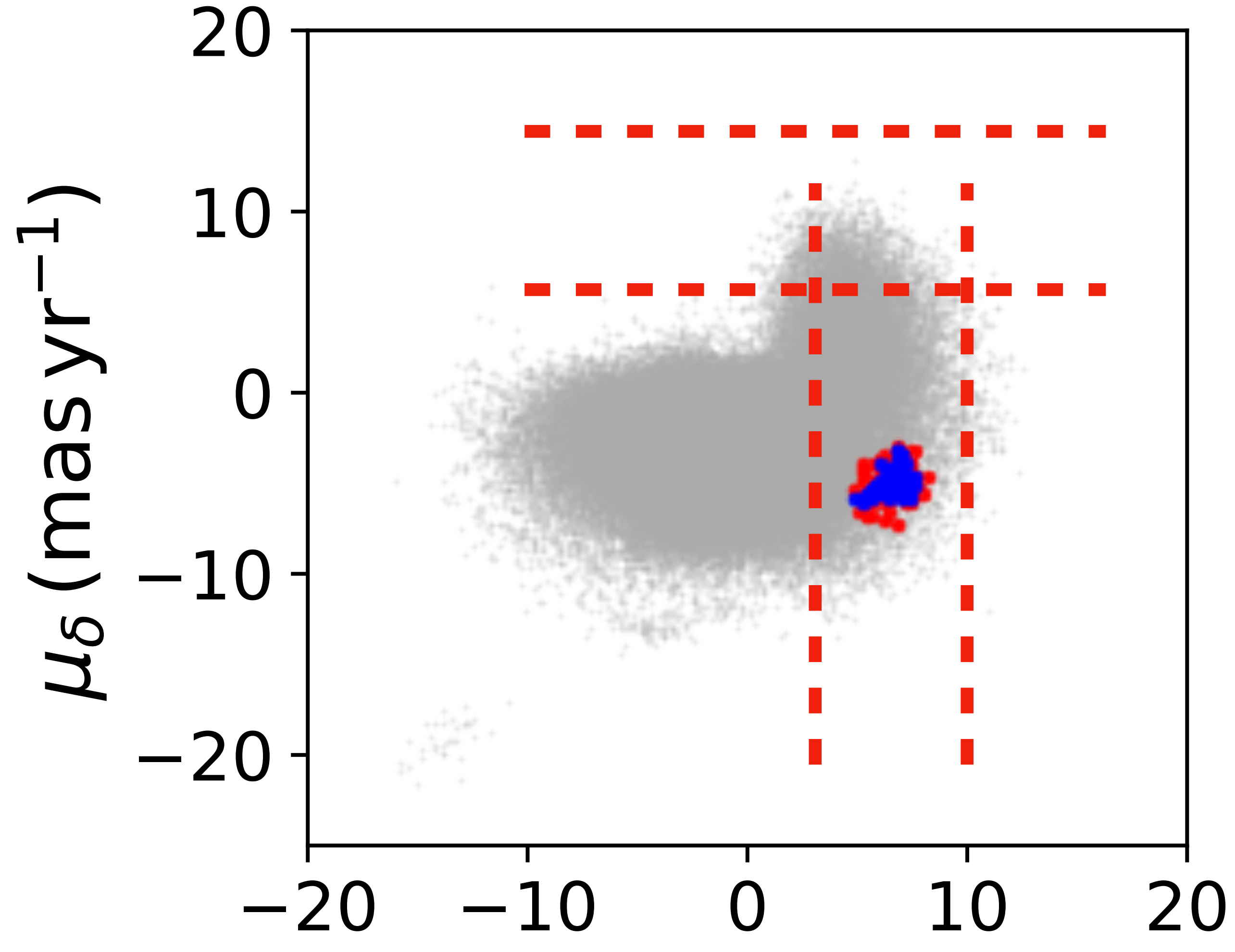
ANODE on Gaia data

- How to set the cut on R(x)?
- We found a defining a single threshold on R(x) across all SRs was insufficient to find other known streams besides GD-1.



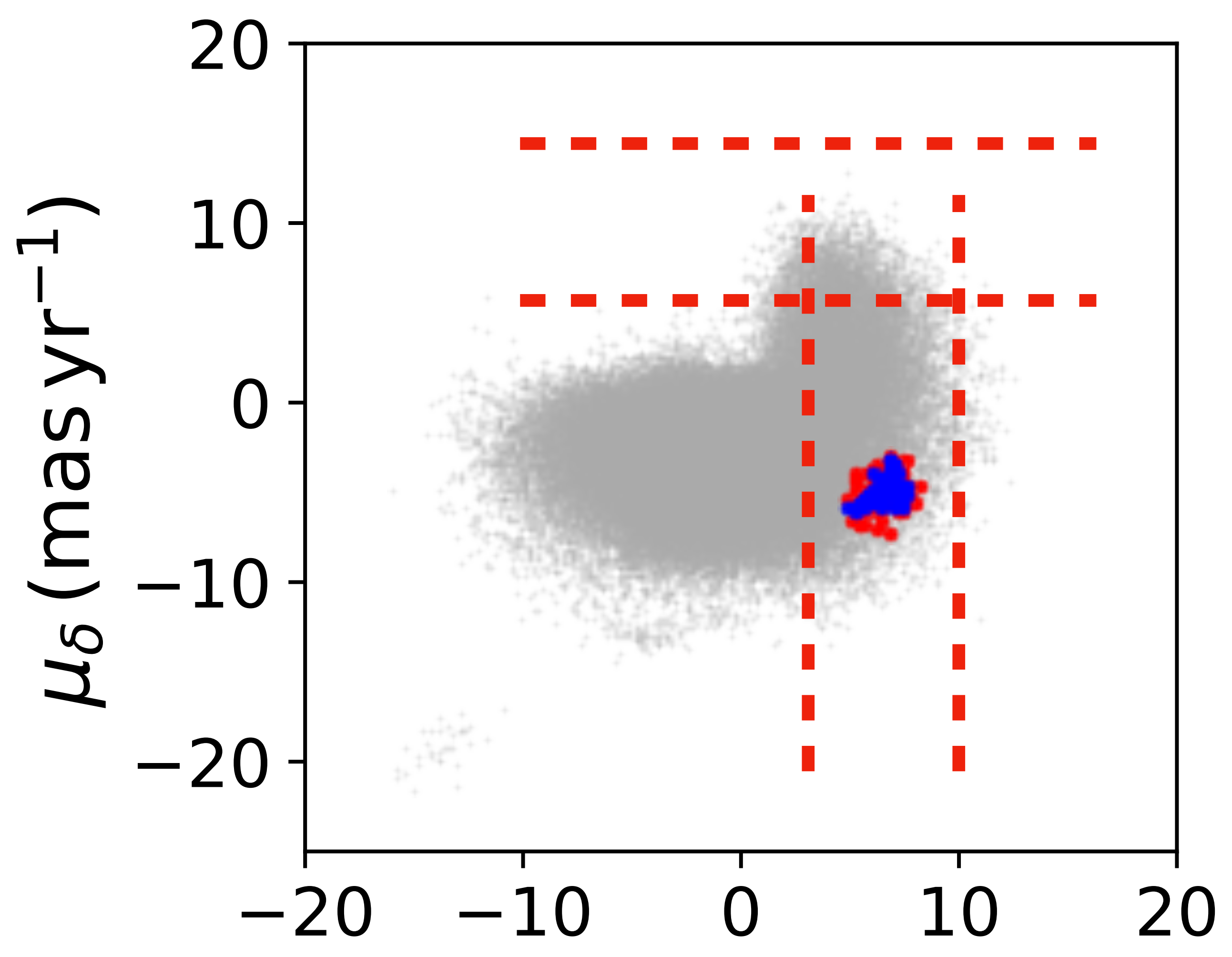
ANODE on Gaia data

- How to set the cut on R(x)?
- We found a defining a single threshold on R(x) across all SRs was insufficient to find other known streams besides GD-1.
- What worked instead was to further subdivide SRs into slices by the orthogonal proper motion => “ROIs”



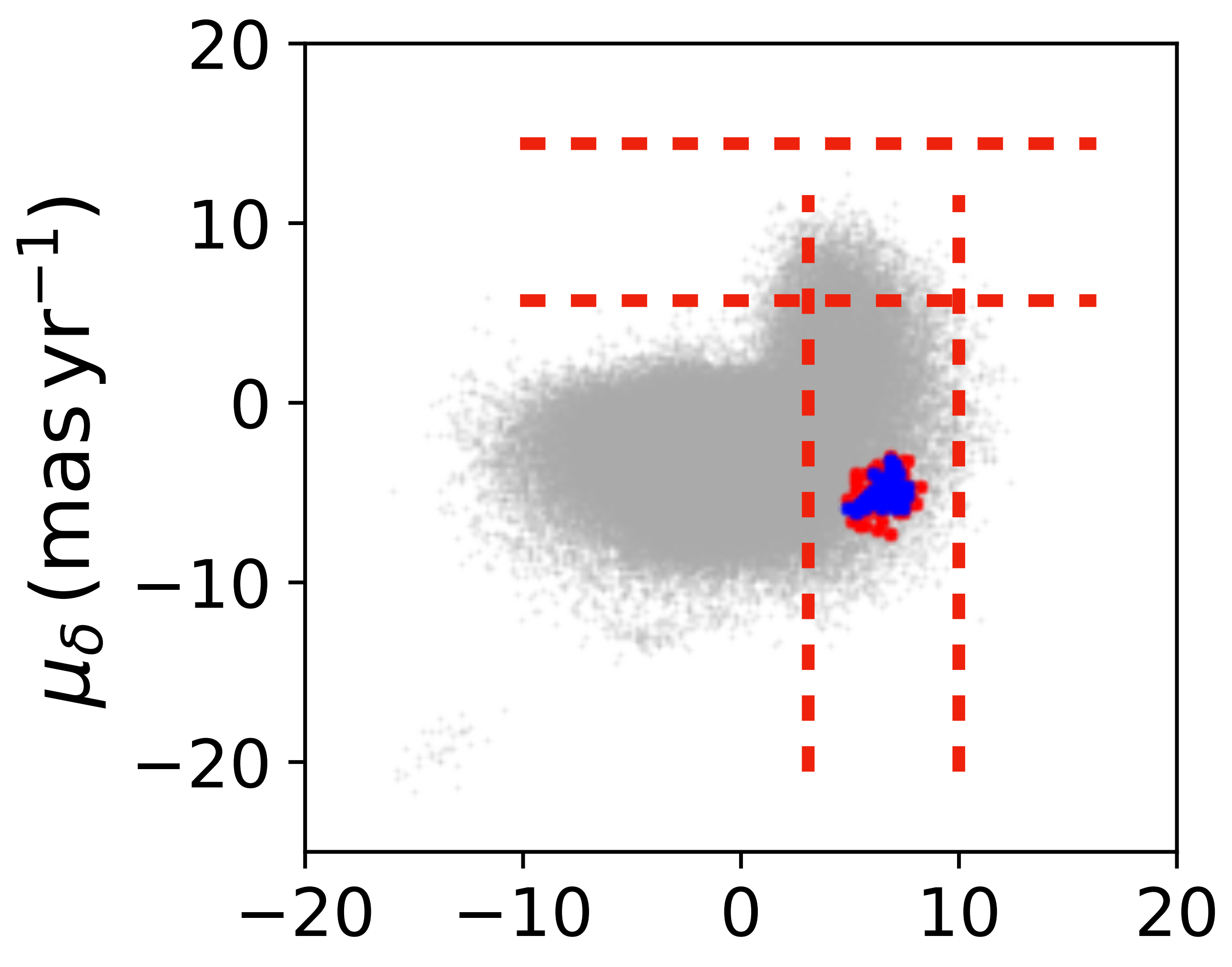
ANODE on Gaia data

- How to set the cut on R(x)?
- We found a defining a single threshold on R(x) across all SRs was insufficient to find other known streams besides GD-1.
- What worked instead was to further subdivide SRs into slices by the orthogonal proper motion => “ROIs”
- In each ROI, take the 100 highest R stars



ANODE on Gaia data

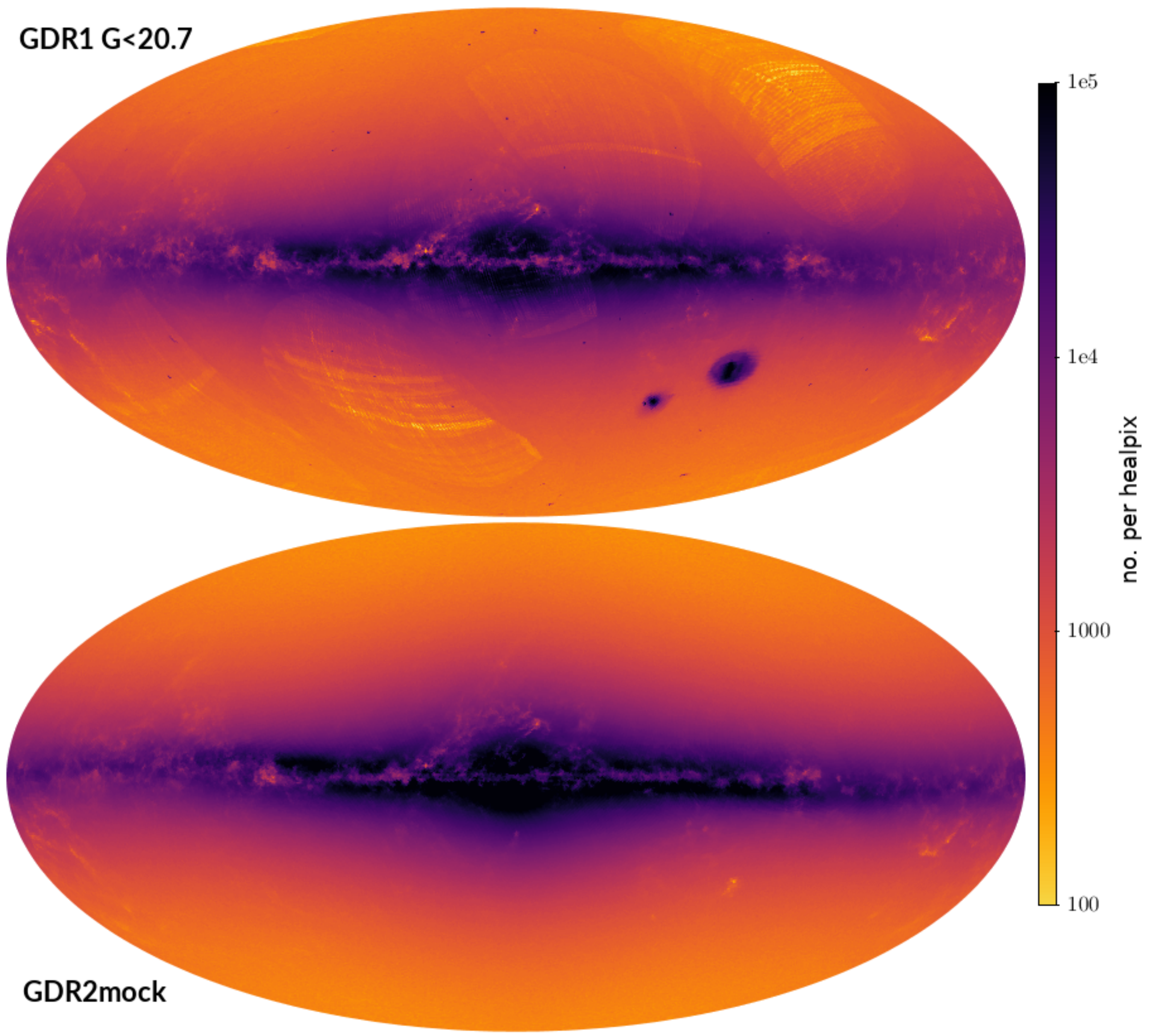
- How to set the cut on R(x)?
- We found a defining a single threshold on R(x) across all SRs was insufficient to find other known streams besides GD-1.
- What worked instead was to further subdivide SRs into slices by the orthogonal proper motion => “ROIs”
- In each ROI, take the 100 highest R stars
- Increases the sensitivity to real streams, but at the cost of a bigger look elsewhere effect.



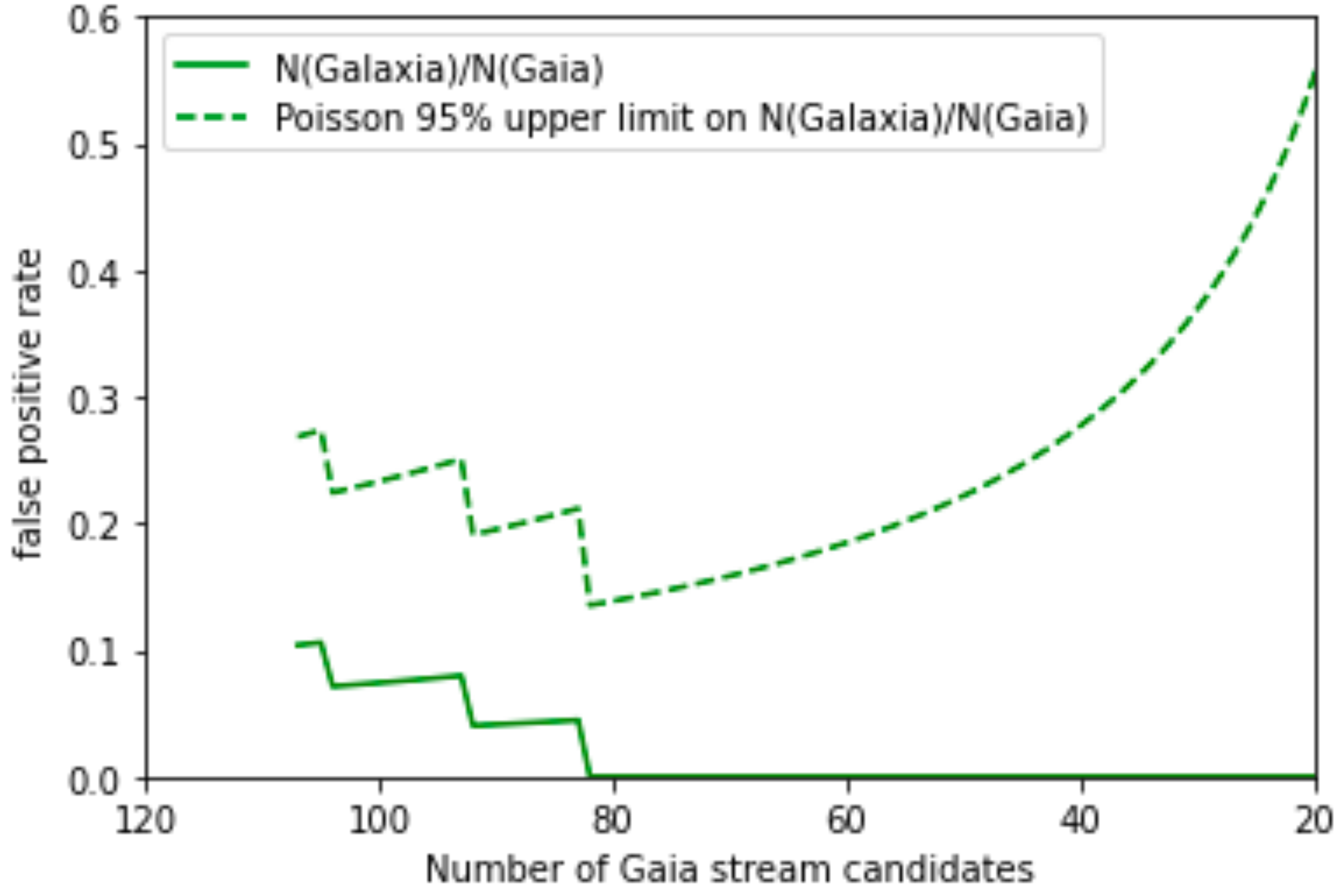
Building streams from fragments

- We end up with ~10^5 ROIs —> need an automated way to scan them for potential streams and a way to cut down on trials factor!
    - Hough transform for line finding => significance
    - Cluster together ROIs from independent runs of ANODE => build stream fragments in each patch and cut down on LEE
    - Cluster together significant stream fragments in different patches to build full stream candidate

Galaxia false positive rate

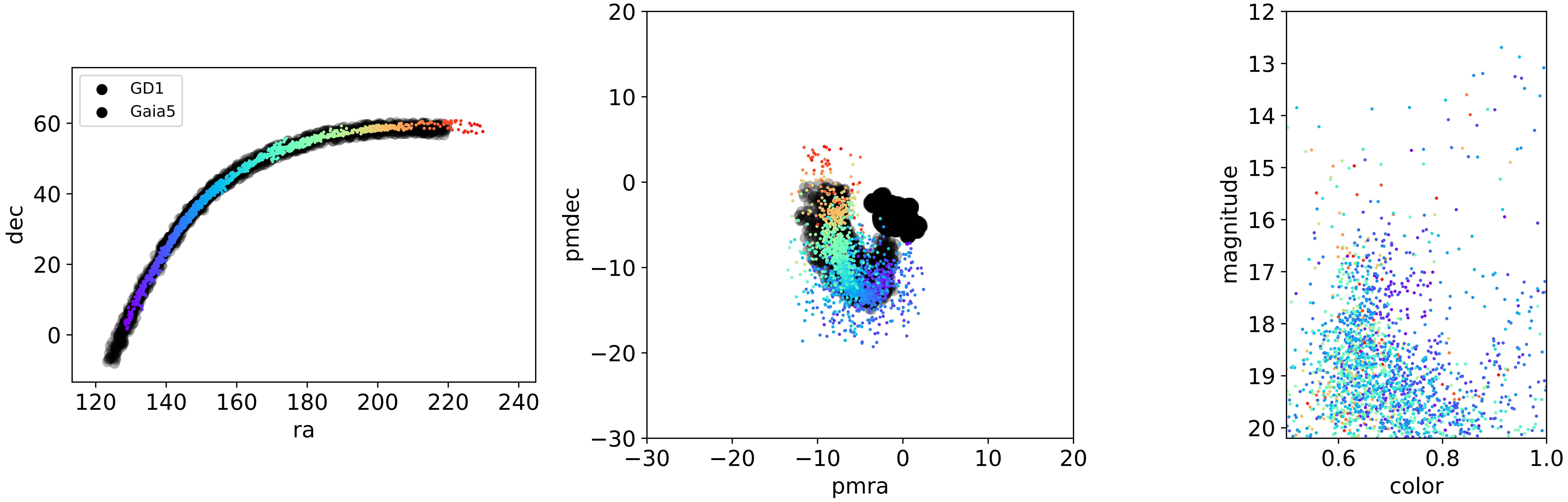


To quantify our false positive rate, we ran our full method on a semi-realistic Gaia mock catalog called Galaxia (Rybizki et al 2018) which does not have stellar streams



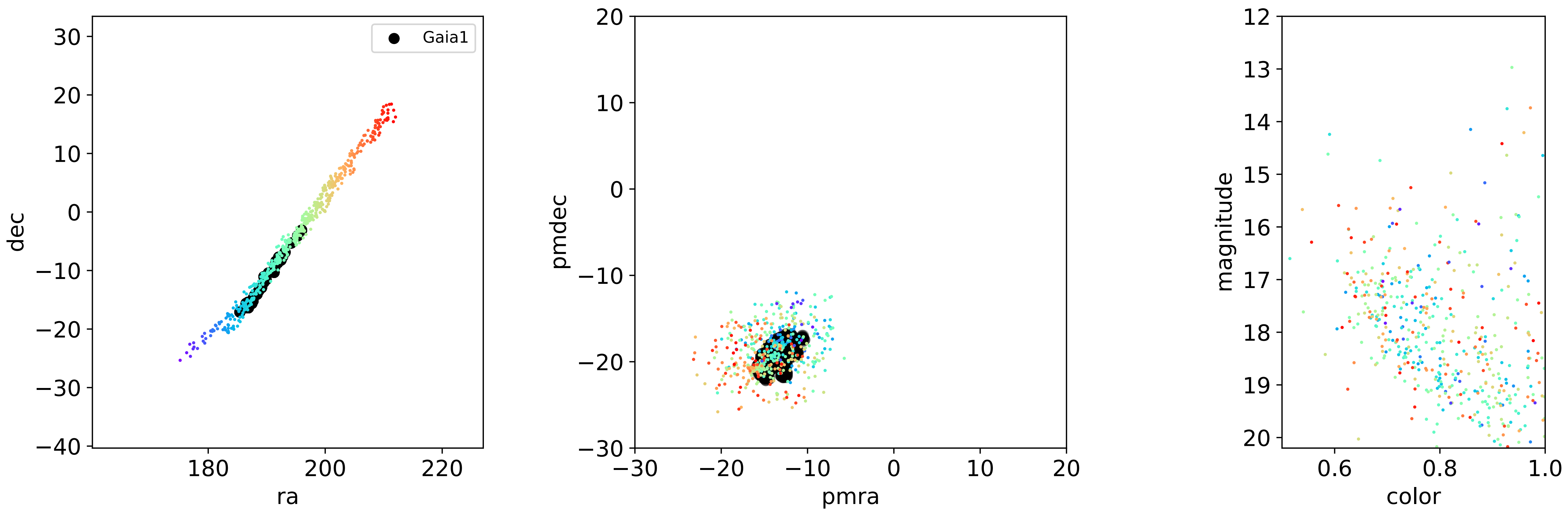
We find 80 stream candidates with a 95% UL on fpr of 13%!

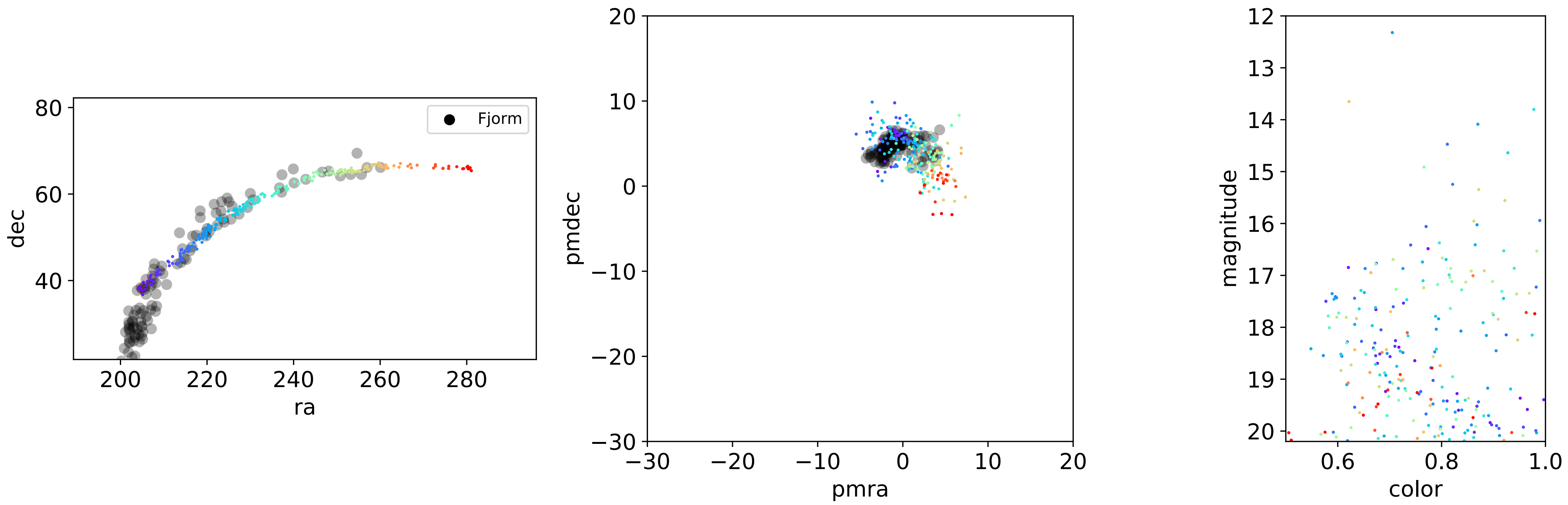
Results: known streams

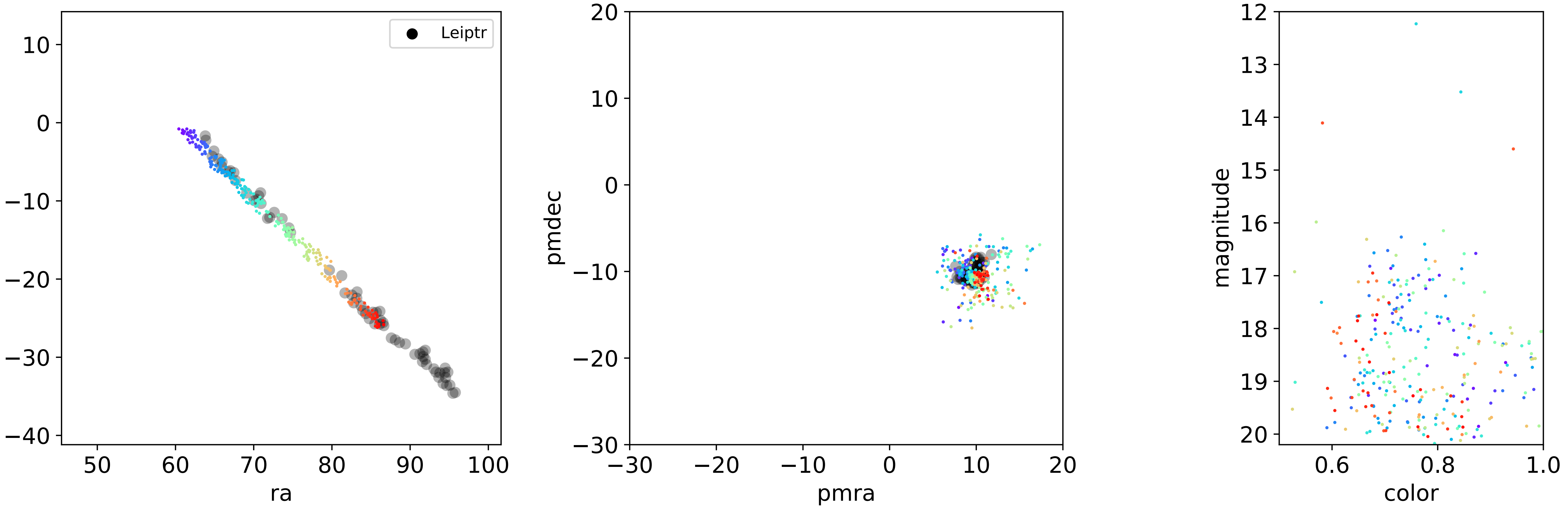


We find essentially all of GD-1, plus possibly a slight extension

Results: known streams

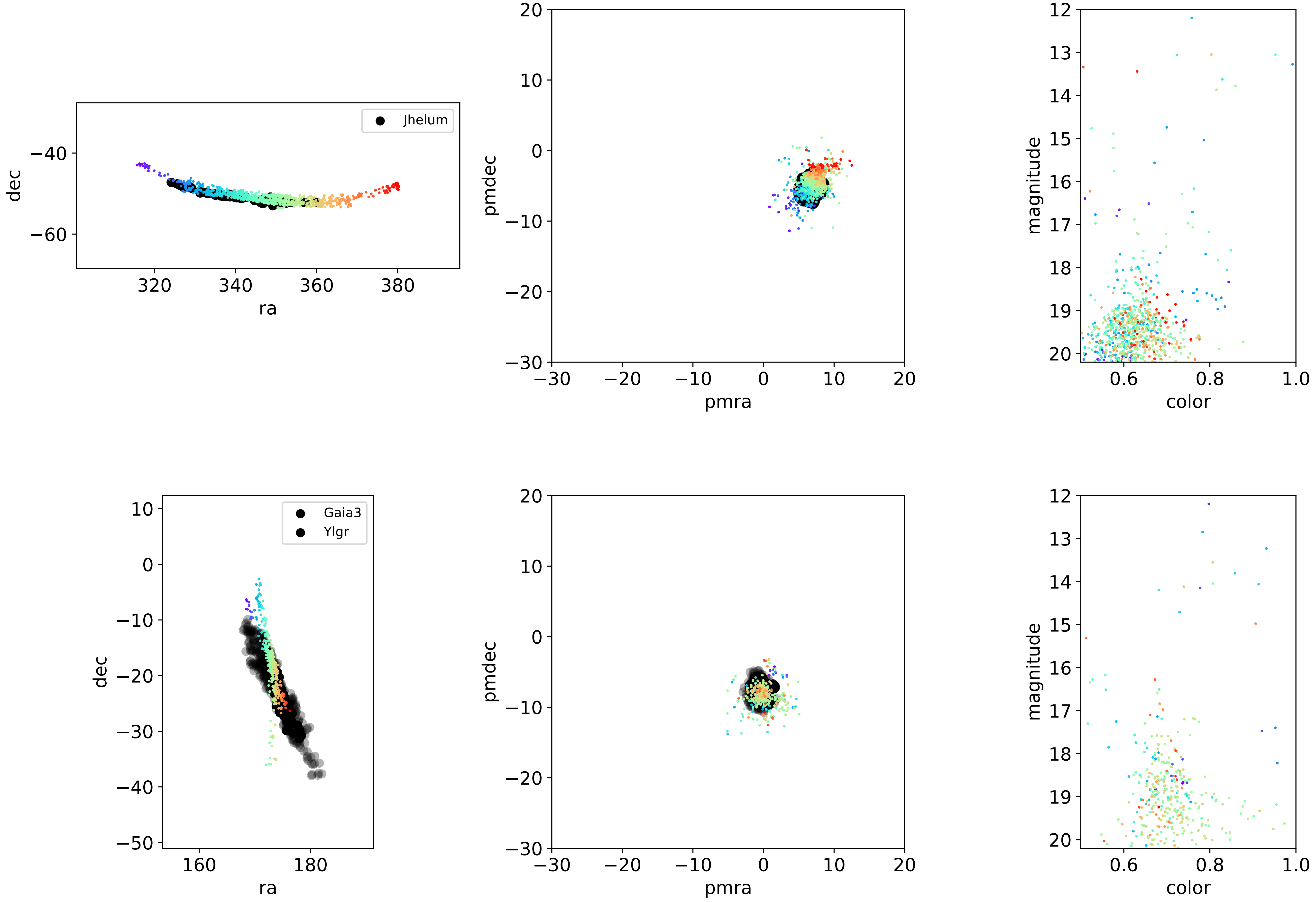






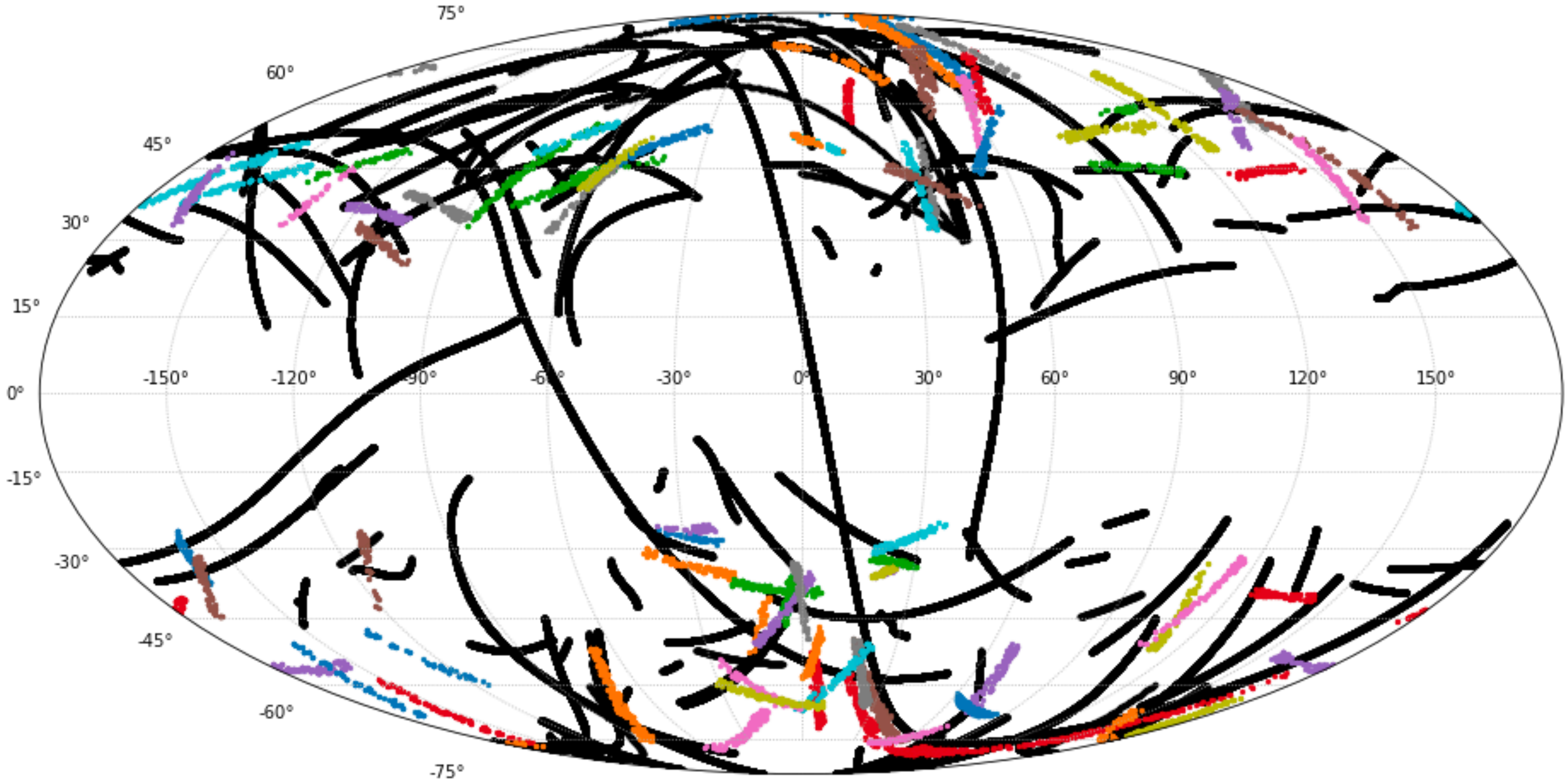
We also find (and potentially extend) many other known streams

Results: known streams



We also find (and potentially extend) many other known streams

Results: new streams



Potentially many new streams (~50-60) discovered!

Gaia: Density Estimation: Measuring Galactic Mass Density

Mass density from phase space density

Buckley, Lim, Putney & **DS** 2205.01129
Green et al 2011.04673, 2205.02244, Naik et al 2112.07657, An et al 2106.05981

- We also realized the same normalizing flows we trained on the sky for ViaMachinae could also have other interesting applications

- Having access to the full 6D phase space density $p(\vec{x}, \vec{v})$ of all the stars in the Galaxy (or at least all the nearby ones) could be very powerful.

- In particular, we can directly infer the mass density $\rho(\vec{x})$ of the Galaxy from knowledge of $p(\vec{x}, \vec{v})$, and from that the mass density $\rho_{DM}(\vec{x})$ of the dark matter.

Mass density from phase space density

Buckley, Lim, Putney & **DS** 2205.01129
Green et al 2011.04673, 2205.02244, Naik et al 2112.07657, An et al 2106.05981

- Liouville theorem: phase space density is conserved
- Stars are well-approximated as collisionless, only interacting through long-ranged gravitational force
- So they must obey the **collisionless Boltzmann equation**:

$$\left[\frac{\partial}{\partial t} + \vec{v} \cdot \frac{\partial}{\partial \vec{x}} + \vec{a}(\vec{x}) \cdot \frac{\partial}{\partial \vec{v}}\right] p(\vec{x}, \vec{v}; t) = 0$$

$$\vec{a}(\vec{x}) = -\frac{d\Phi(\vec{x})}{d\vec{x}}$$ Accelerations

$\Phi(\vec{x})$ is the gravitational potential of the Galaxy (DM+stars+gas…)

Mass density from phase space density

Buckley, Lim, Putney & **DS** 2205.01129
Green et al 2011.04673, 2205.02244, Naik et al 2112.07657, An et al 2106.05981

- If the stars are also in dynamical equilibrium, then $\partial p / \partial t = 0$ and we get:

$$\left[\vec{v} \cdot \frac{\partial}{\partial \vec{x}} + \vec{a}(\vec{x}) \cdot \frac{\partial}{\partial \vec{v}}\right] p(\vec{x}, \vec{v}) = 0$$

- Just from knowledge of $p(\vec{x}, \vec{v})$ and its derivatives we can determine the accelerations $\vec{a} = -\nabla \Phi$

- Taking another derivative gives us the mass density of the Galaxy!

$$4\pi G\rho = \nabla^2 \Phi = \nabla \cdot \vec{a}$$

More on determining the accelerations

Green & Ting 2011.04673, An et al 2106.06981, Naik et al 2112.07657, Buckley, Lim, Putney & **DS** 2205.01129

$$\left[\vec{v} \cdot \frac{\partial}{\partial \vec{x}} + \vec{a}(\vec{x}) \cdot \frac{\partial}{\partial \vec{v}}\right] p(\vec{x}, \vec{v}) = 0$$

- How can we solve for 3 acceleration functions $\vec{a}(\vec{x})$ with just a single equation?
- $\vec{a}(\vec{x})$ doesn't depend on velocity! So this is actually an infinite number of equations for $\vec{a}(\vec{x})$, one for each choice of $\vec{v}$
- We choose to perform least-squares minimization over a sample of velocities to determine best-fit $\vec{a}(\vec{x})$

$$L(\vec{a}(\vec{x})) = \frac{1}{N} \sum_{\alpha=1}^{N} \left(\left[\vec{v}_\alpha \cdot \frac{\partial}{\partial \vec{x}} + \vec{a}(\vec{x}) \cdot \frac{\partial}{\partial \vec{v}}\right] p(\vec{x}, \vec{v}_\alpha) \right)^2$$

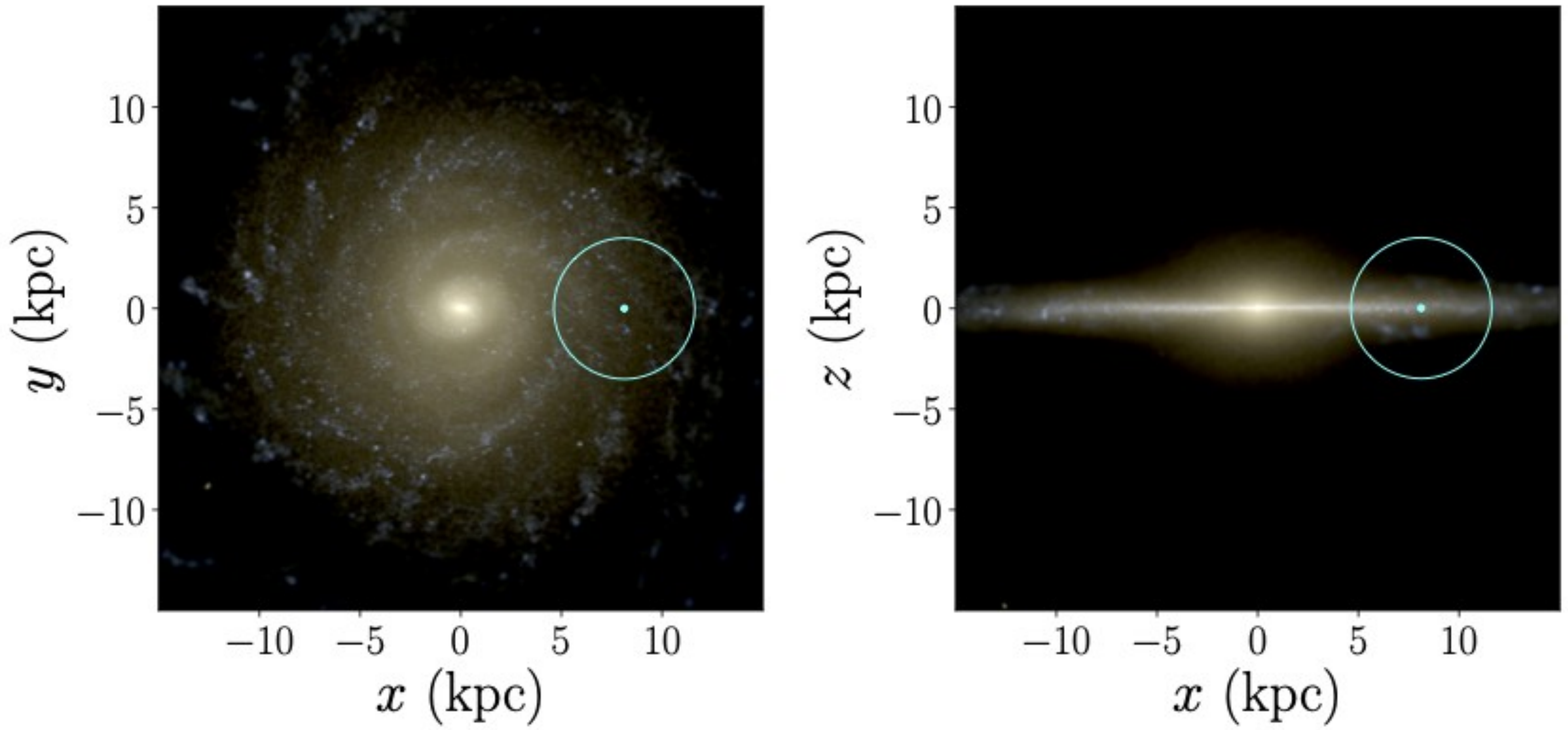
Proof-of-concept

Our work: first to use Nbody+hydro simulation

Buckley, Lim, Putney & **DS** 2205.01129

- Training data: state-of-the-art Nbody+hydro galaxy simulation from "Nbody shop" collaboration [https://b2share.eudat.eu/records/c9f232d8ac804785aad35004177a704e]

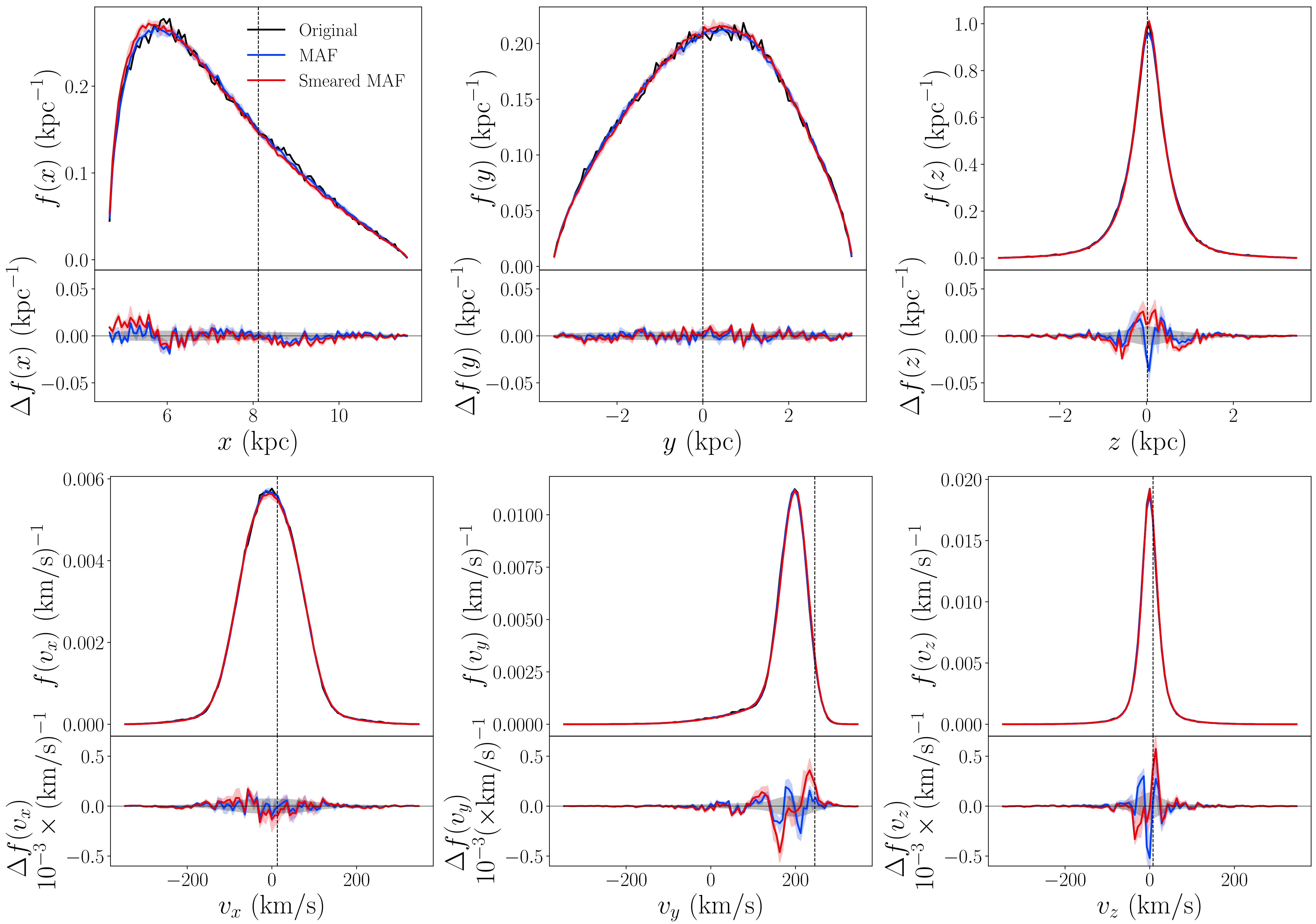
- Milky Way like Galaxy h277



- number of ~~stars~~ star particles
  153,174 (<< size of Gaia 6D dataset)
- observer's location
  [8.122, 0., 0.0208] kpc
- observing radius = 3.5 kpc
- simulation resolution: 0.173 kpc
- Using only kinematic information:
  position and velocity

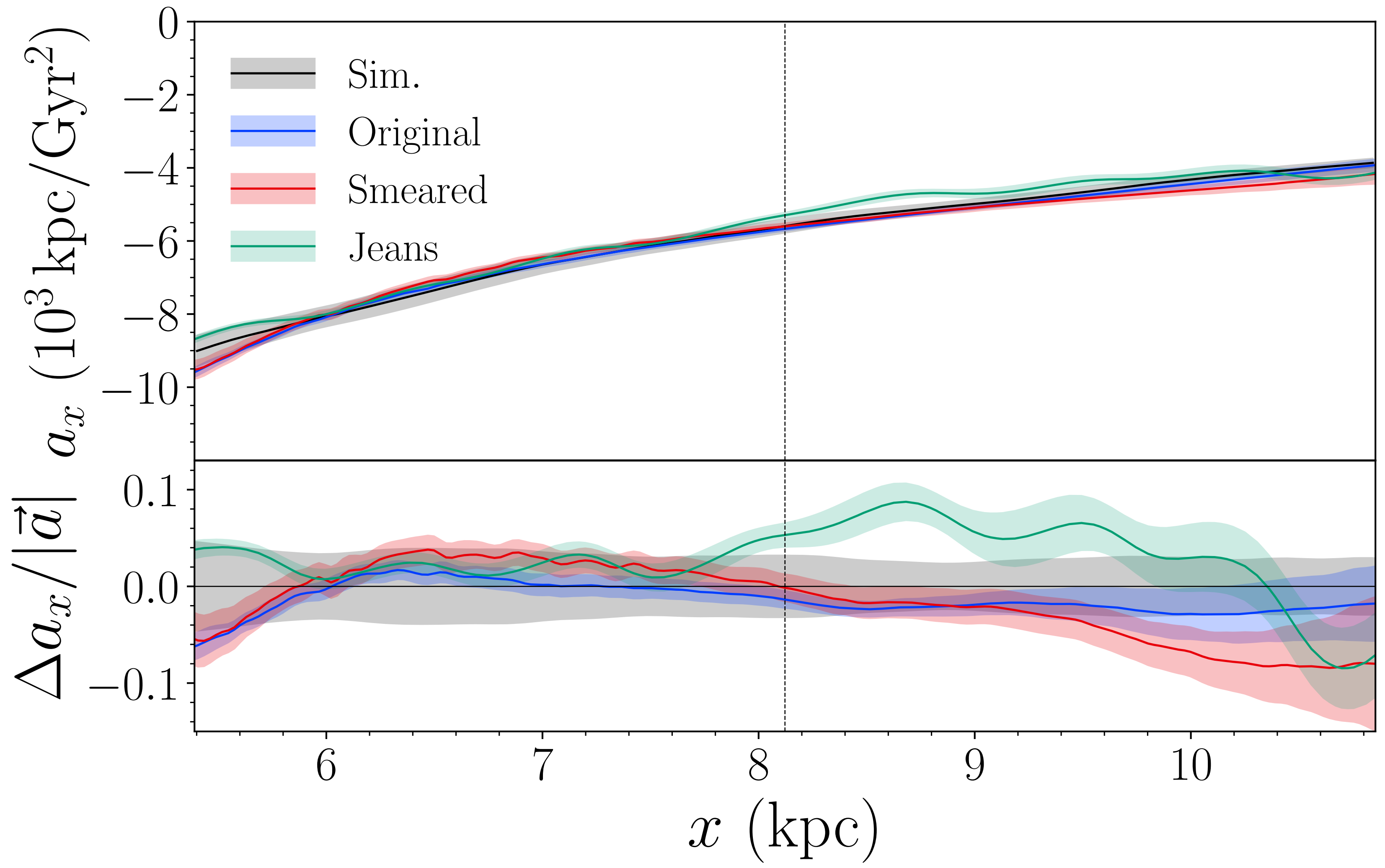
Results: density estimation

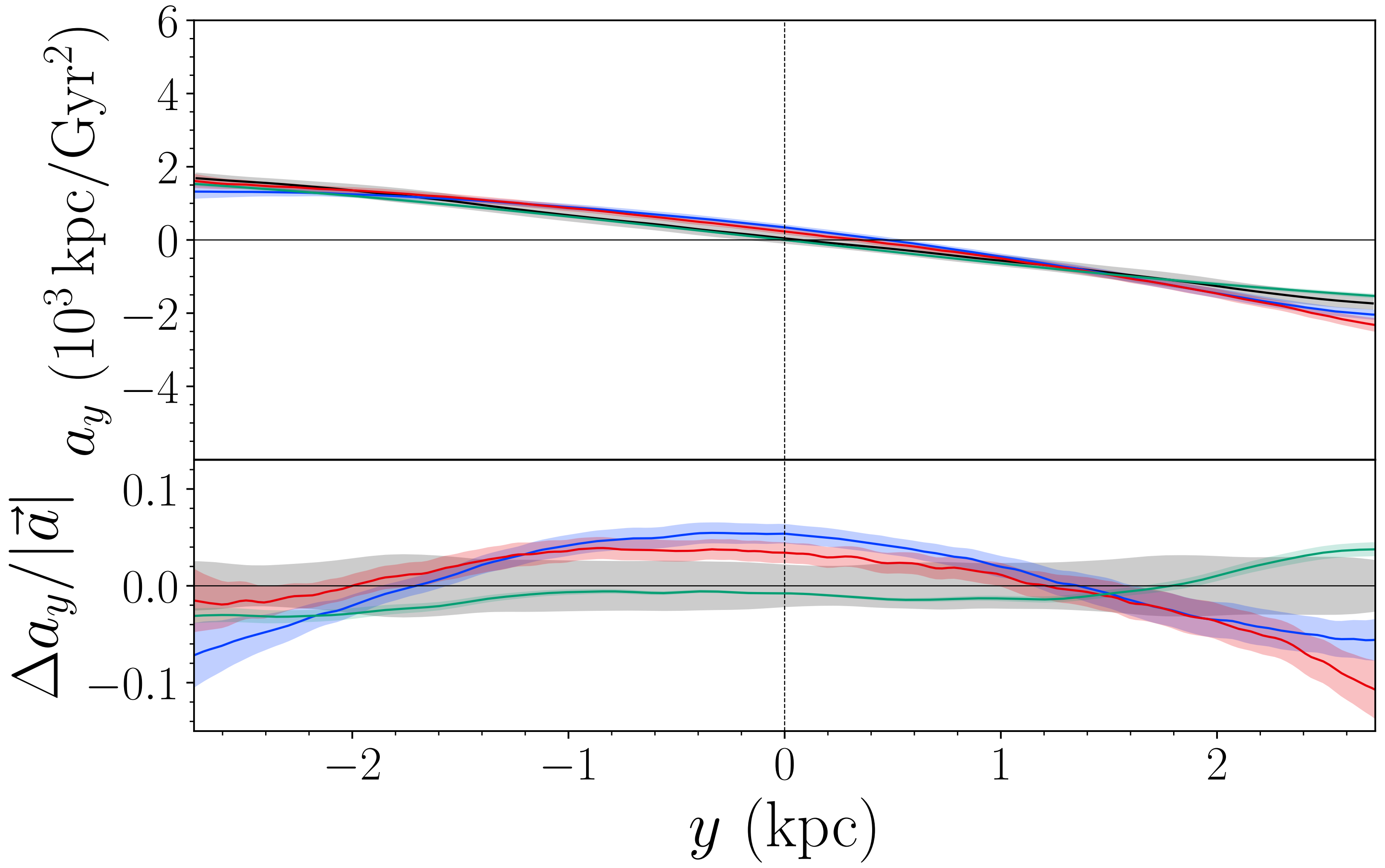
Buckley, Lim, Putney & **DS** 2205.01129

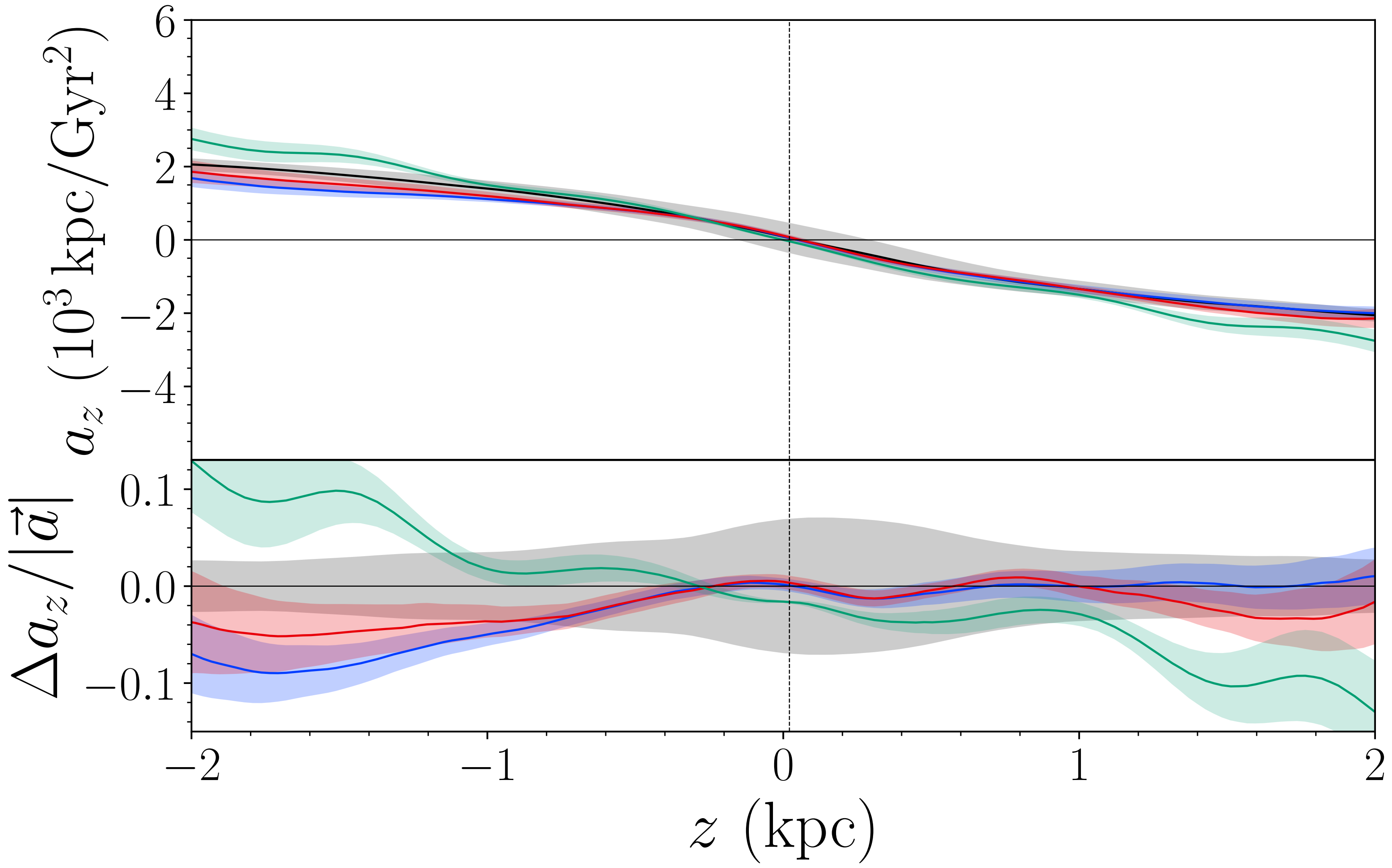


Results: accelerations

Buckley, Lim, Putney & **DS** 2205.01129







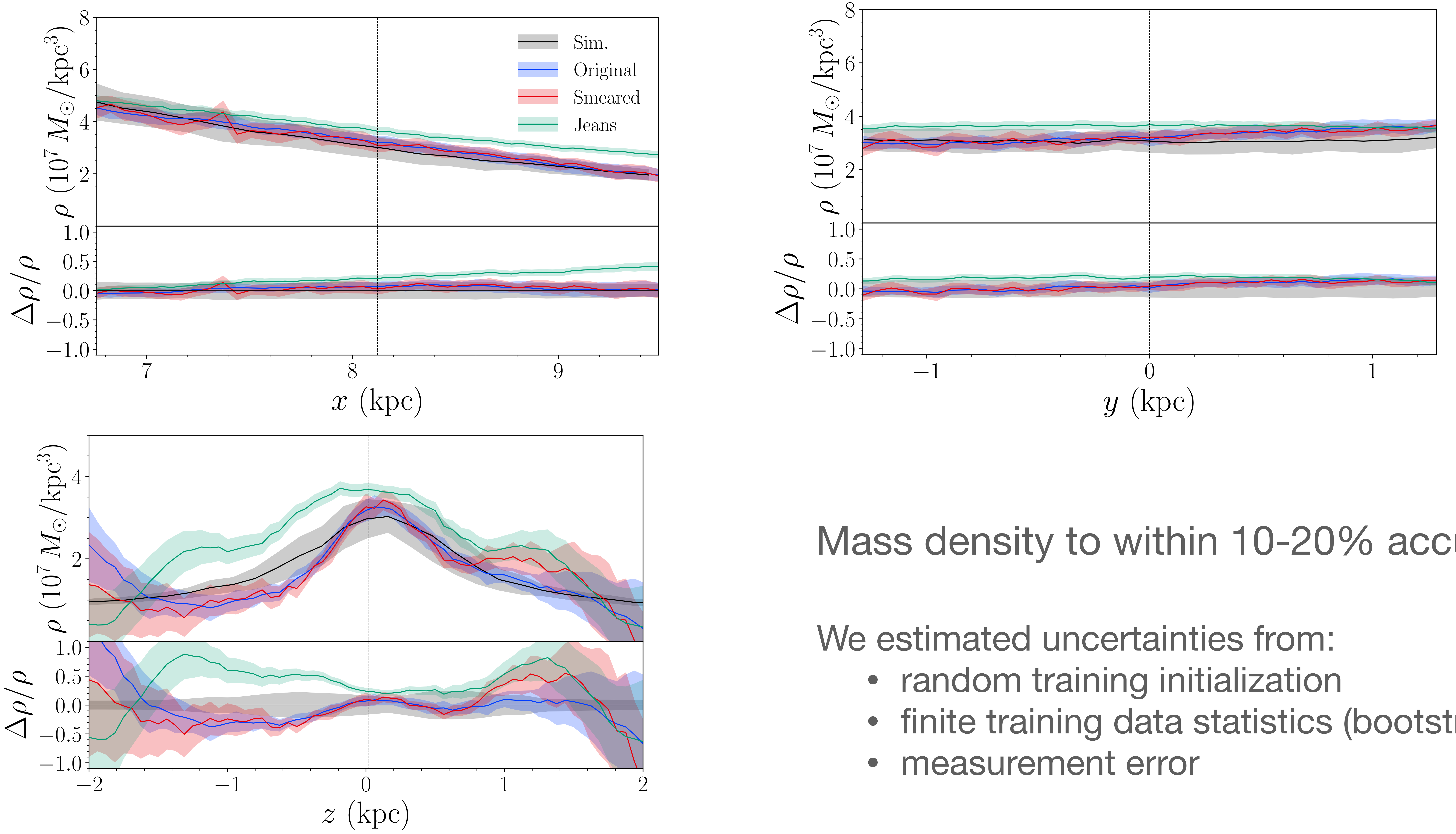
Accelerations to within 5% accuracy!

We estimated uncertainties from:

- random training initialization
- finite training data statistics (bootstrap)
- measurement error

Results: mass density

Buckley, Lim, Putney & **DS** 2205.01129



Mass density to within 10-20% accuracy!

We estimated uncertainties from:

- random training initialization
- finite training data statistics (bootstrap)
- measurement error

Comparison to existing measurements

- Existing measurements typically use Jean's equation (second moment of Boltzmann equation)

$$\overline{v_i v_j}\frac{\partial \nu(\vec{x})}{\partial x_i} + \nu(\vec{x})\frac{\partial \overline{v_i v_j}}{\partial x_i} = -\nu(\vec{x})\frac{\partial \Phi}{\partial x_j}$$

- Assume axisymmetry, reflection symmetry…
- Bin data and perform parametric fits to extract $\Phi(\overrightarrow{x})$
- Results can seem precise but might not be accurate (biased)

From de Salas & Widmark 2012.11477