HEPSim2Real*

Creating background templates with normalizing flows

**Still working on a suitable acronym…*

Radha Mastandrea

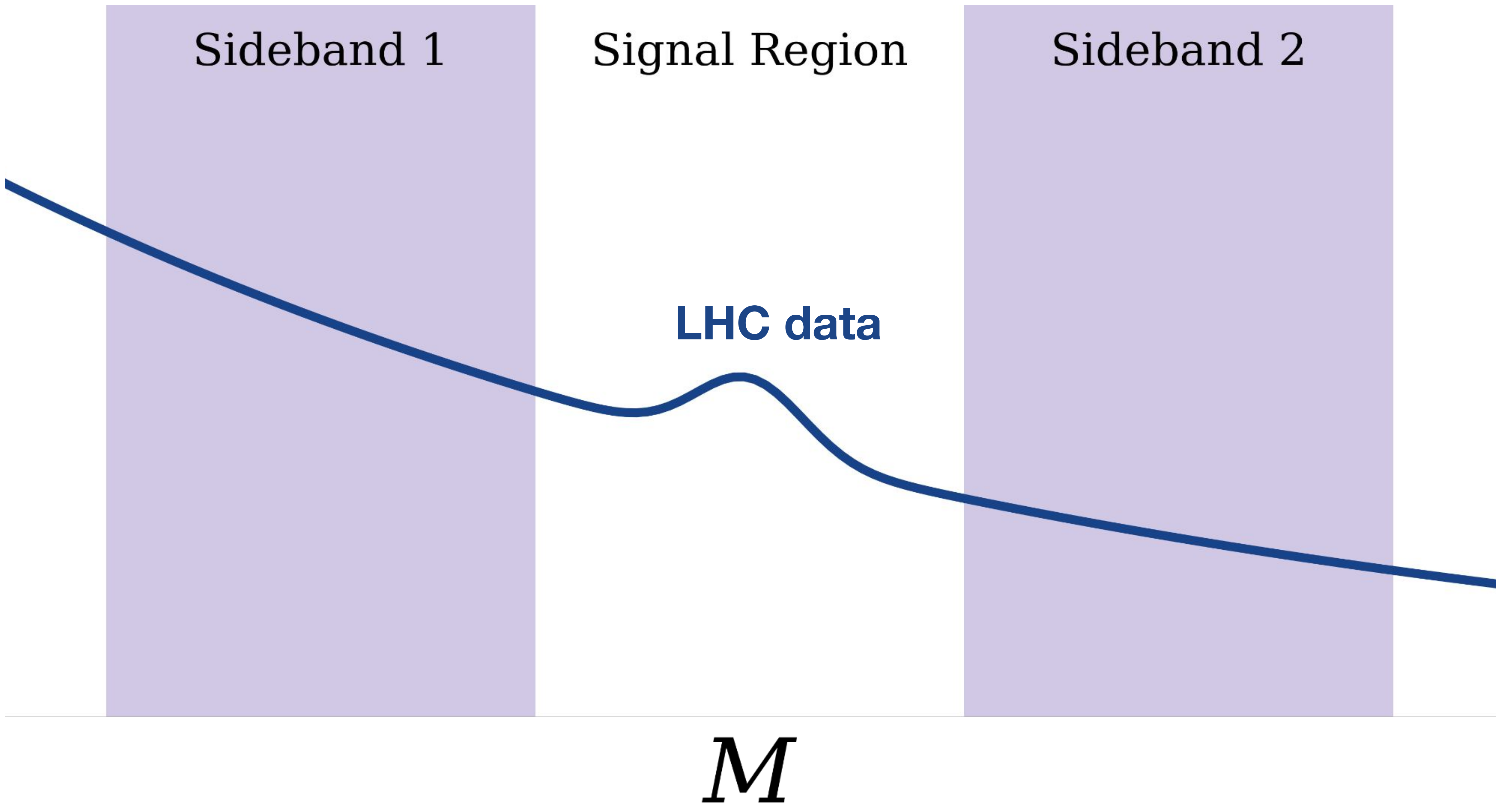
In collaboration with Tobias Golling, Samuel Klein, and Ben Nachman

ML4Jets

11/03/2022

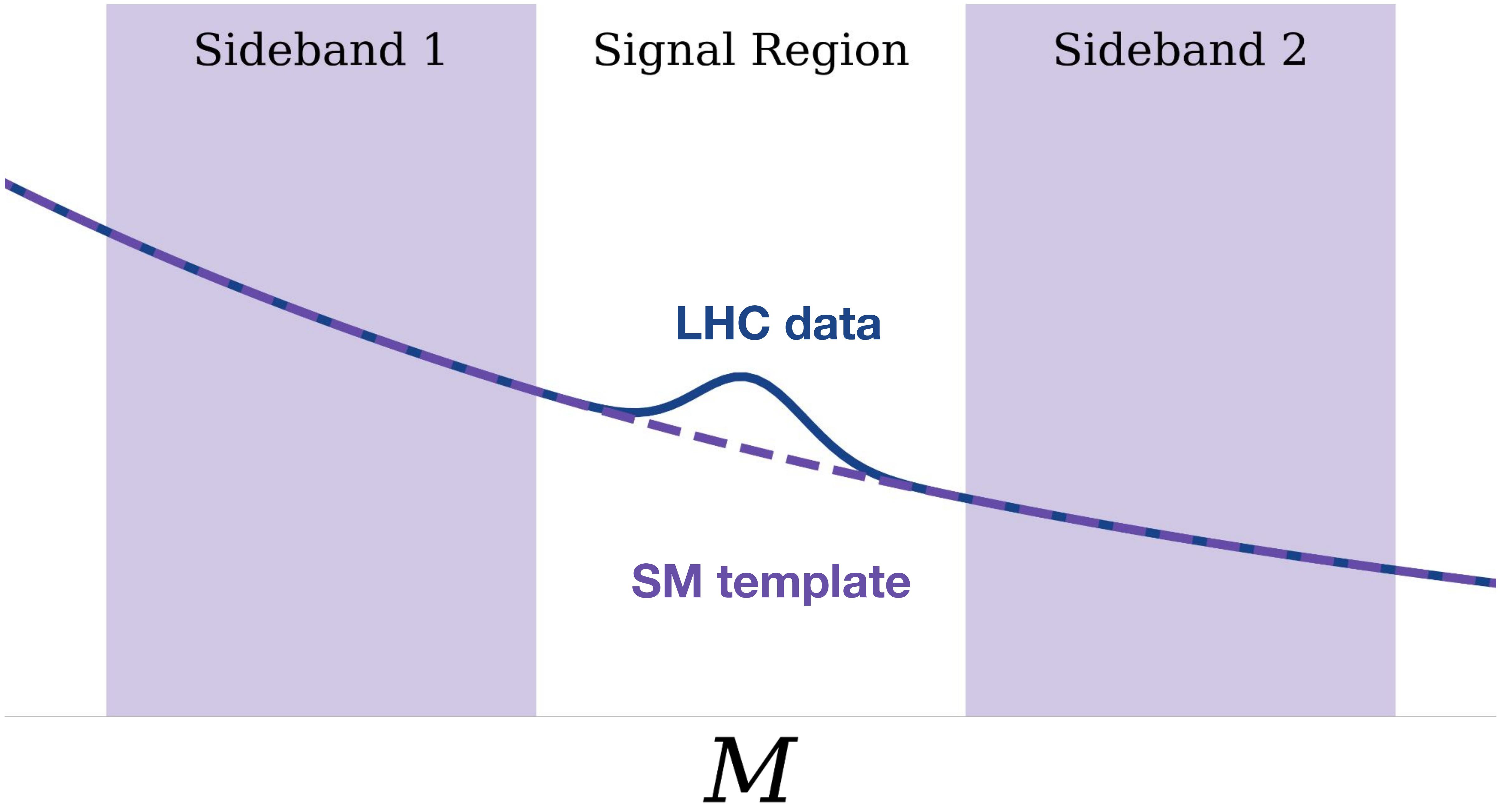
The goal

Create an accurate **SM background template** for resonant anomaly detection



The goal

Create an accurate **SM background template**
for resonant anomaly detection



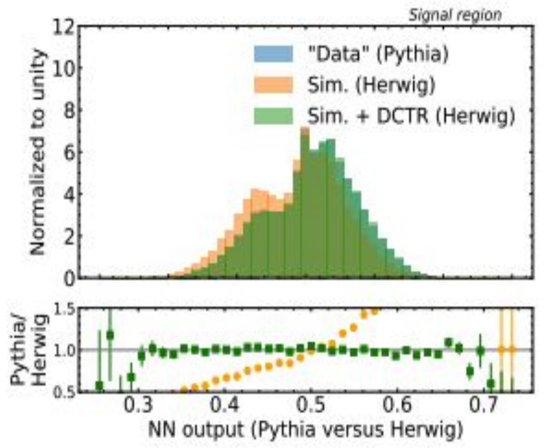
Previous attempts to model SR background

<table>
<tr><th></th><th>Learn from simulation</th><th>Learn from data (SB)</th></tr>
<tr><td>Modeling the Likelihood (Ratio)</td><td>

SALAD

2001.05001

Andreassen, Nachman, Shih

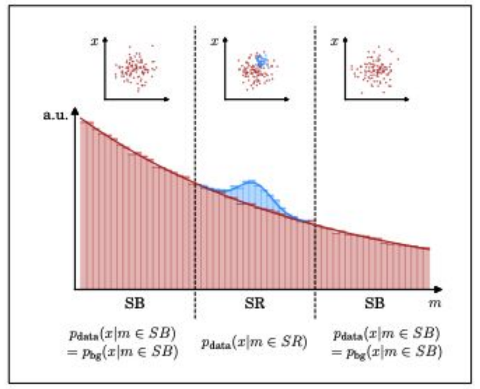


</td><td>

CATHODE

2109.00546

Hallin, Isaacson, Kasieczka et al.



</td></tr>
<tr><td>Morphing the Features</td><td>

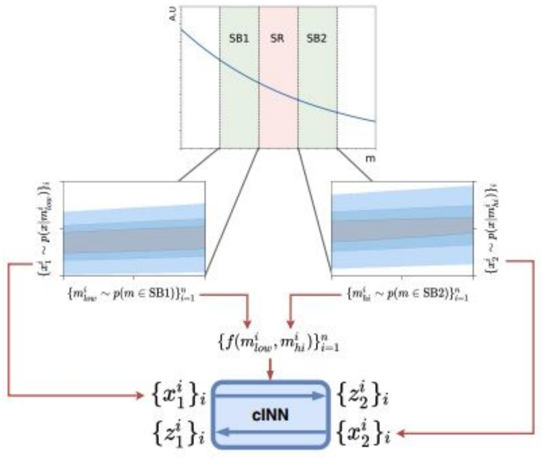
(This presentation)

</td><td>

CURTAINS

2203.09470

Raine, Klein, Sengupta et al.

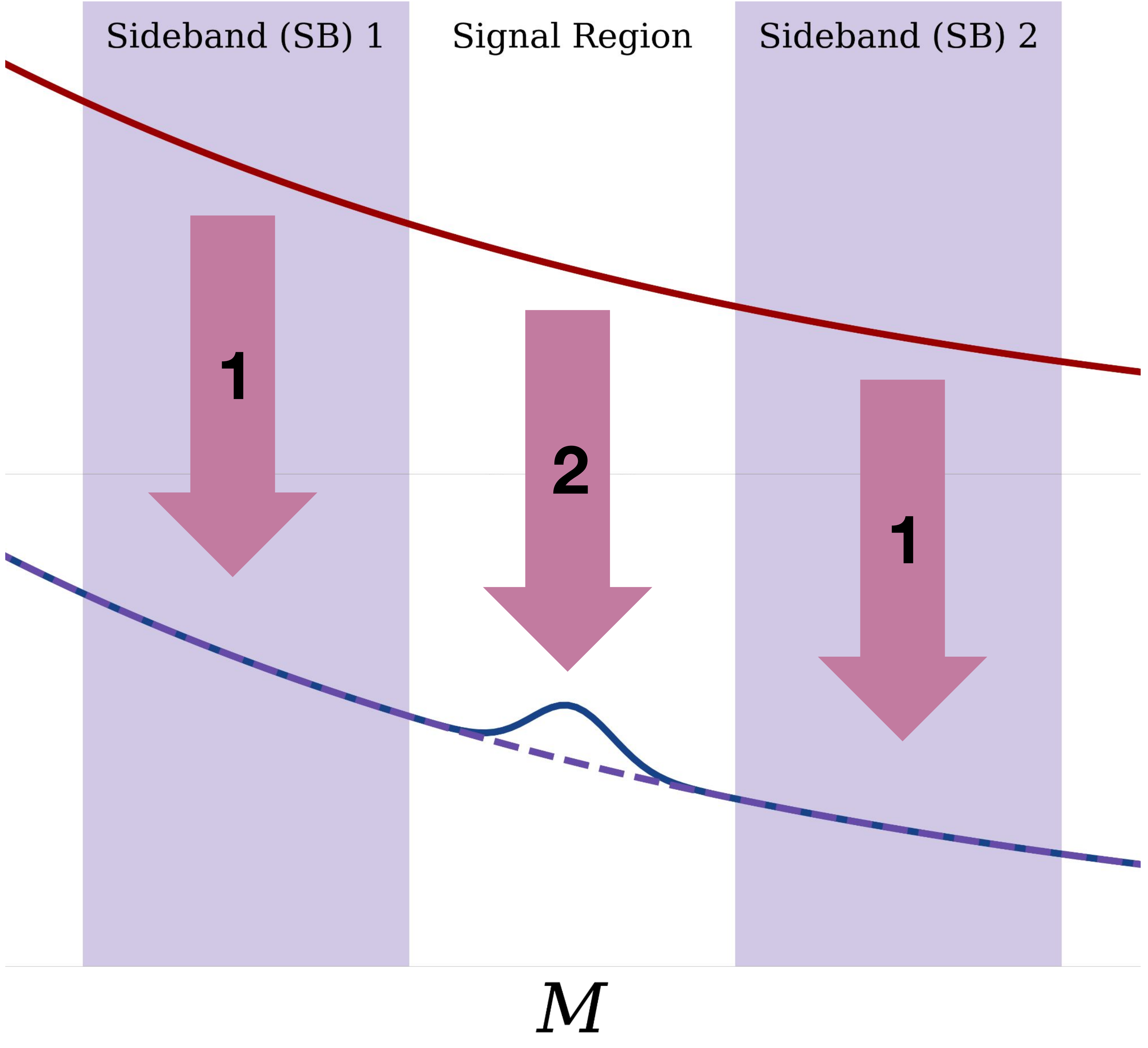


</td></tr>
</table>

Our approach: HEPSim2Real

(1) Use normalizing flows to learn a map from **SM simulation** to **data** in SB

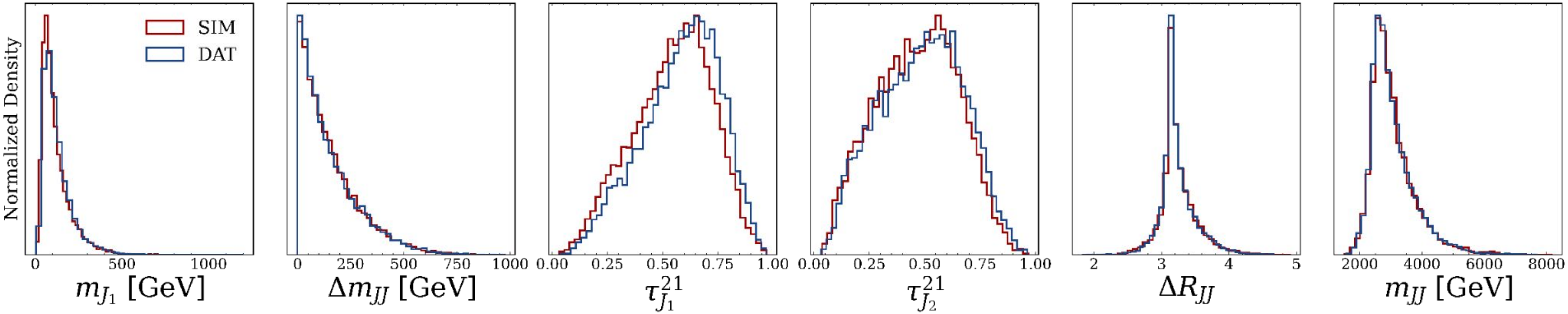
(2) Apply this **map** to **simulation SR** to construct a **background template for SM**



…then search for resonant anomalies by comparing the **template** to **SR data**

The dataset: LHCO Herwig and Pythia

- The LHC Olympics dataset (on Zenodo) consists of 1mil background QCD dijet events and 100k signal dijet events.
    - Herwig → “simulation”; Pythia → “data”



$$\begin{aligned} \text{SB1: } & m_{JJ} \in [2900, 3300] \text{ GeV} \\ \text{SR : } & m_{JJ} \in [3300, 3700] \text{ GeV} \\ \text{SB2: } & m_{JJ} \in [3700, 4100] \text{ GeV} \end{aligned}$$

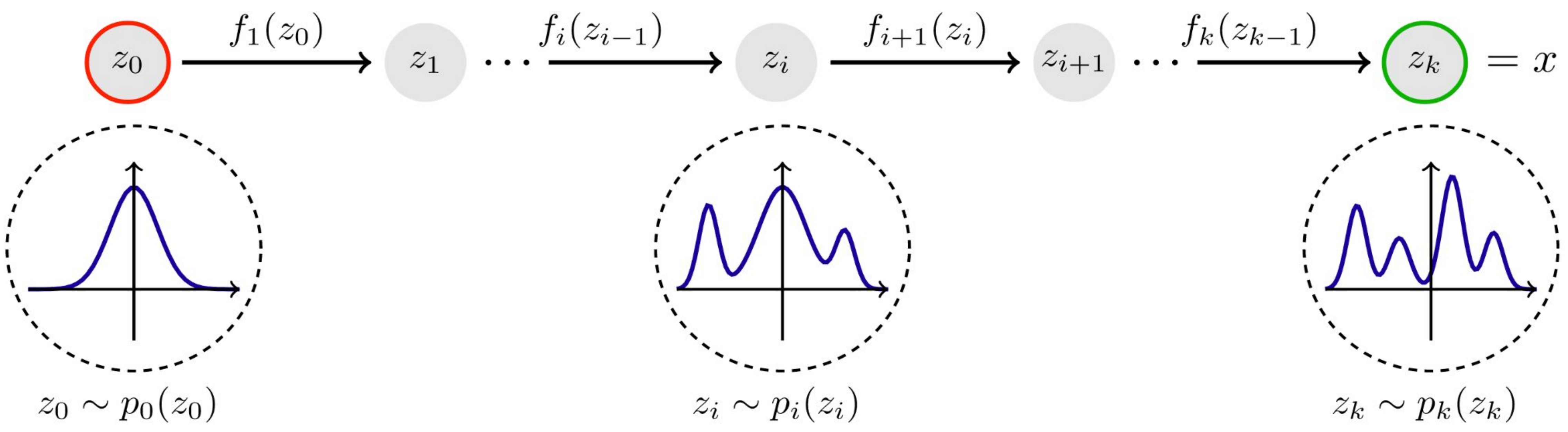
Computational procedure

Datasets

- Train the flow on 280k (each) simulation and data events in the SB
- Train classifiers on 120k (each) transformed simulation and data events in the SR
- Test classifiers on 20k signal, 20k background events

Training

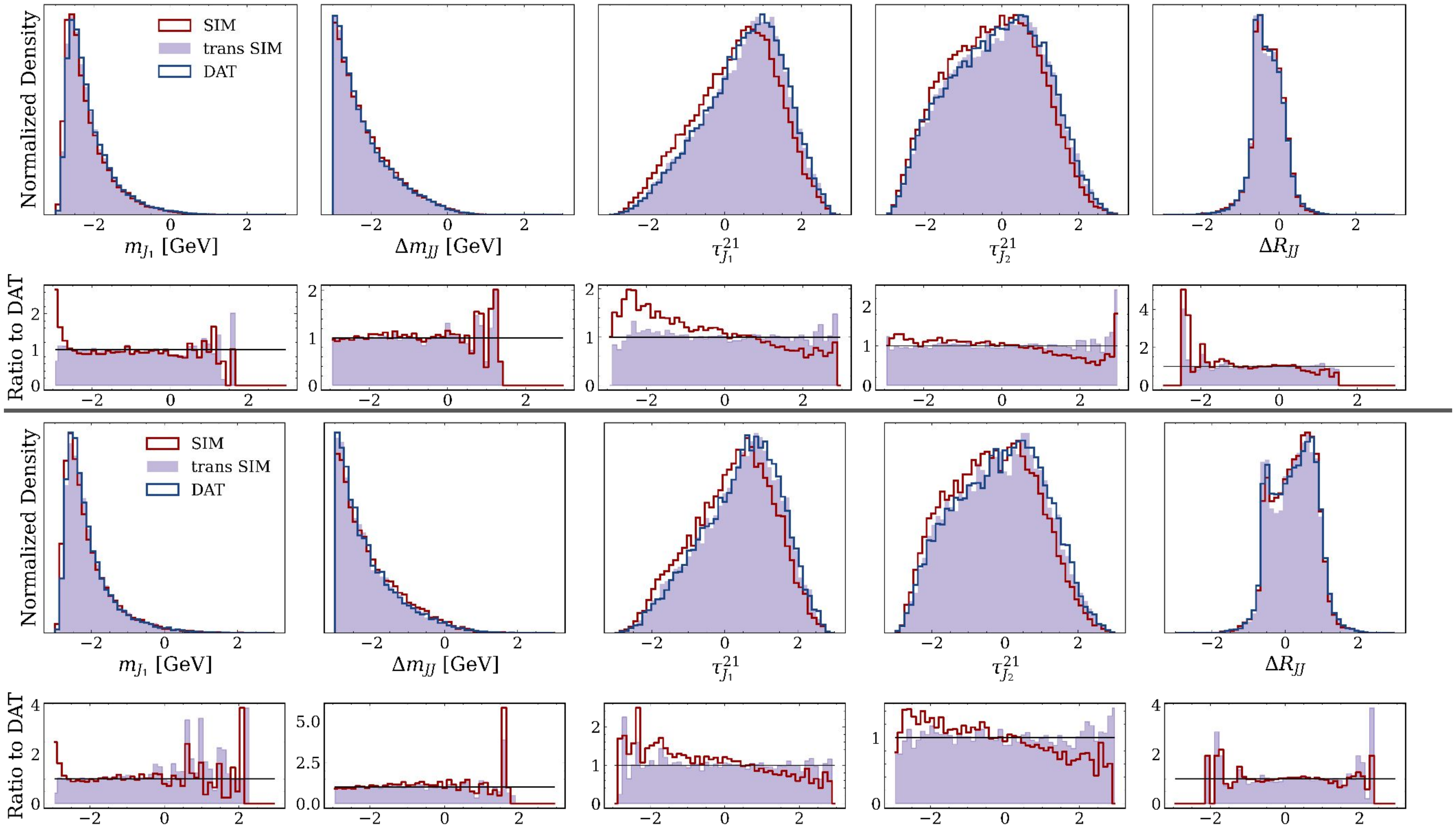
- Train a coupling normalizing flow for 100 epochs, LR 5e-3, BS 256
- Detailed flow architecture given in the backup slides



https://flowtorch.ai/users/

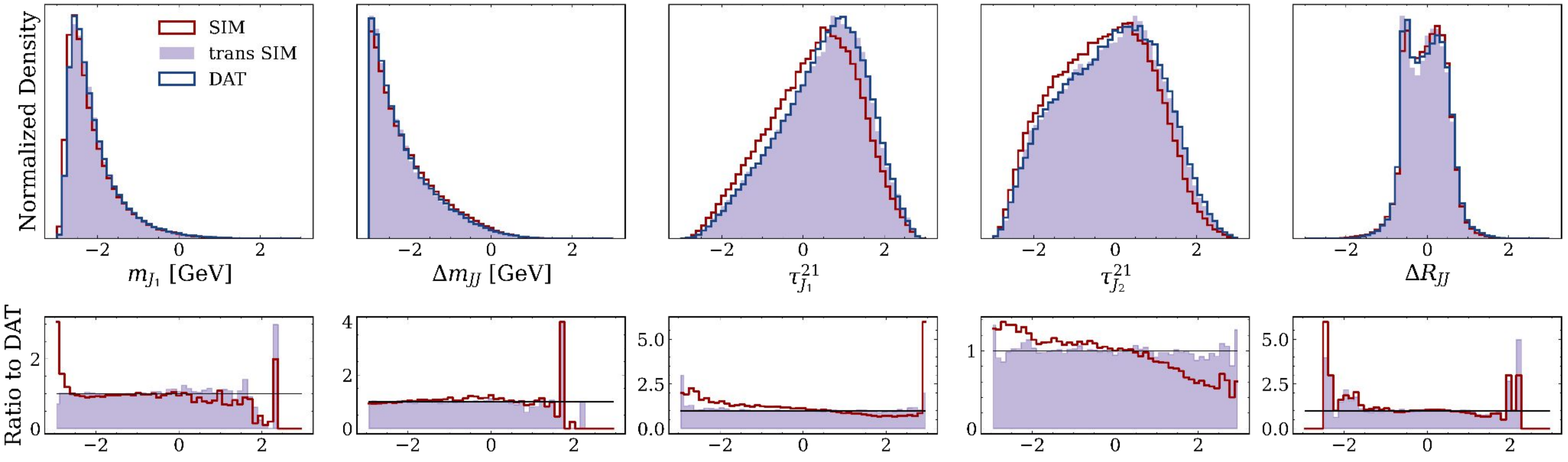
Does the flow learn the optimal transport mapping? Details in the backups!

The flow effectively learns to map simulation to data in the SB…



Goal: align **trans SIM** with **DAT**

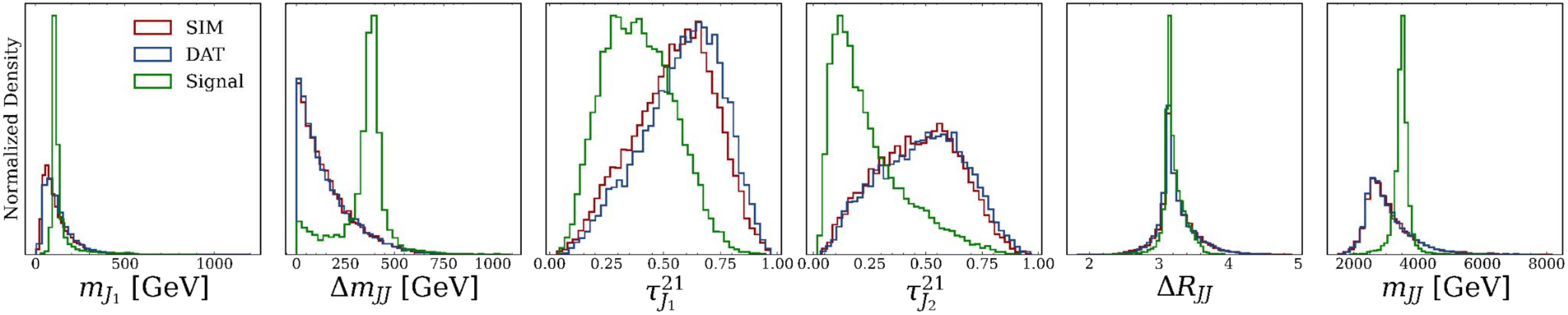
…and the flow performs well when interpolated into the SR



Signal injection studies

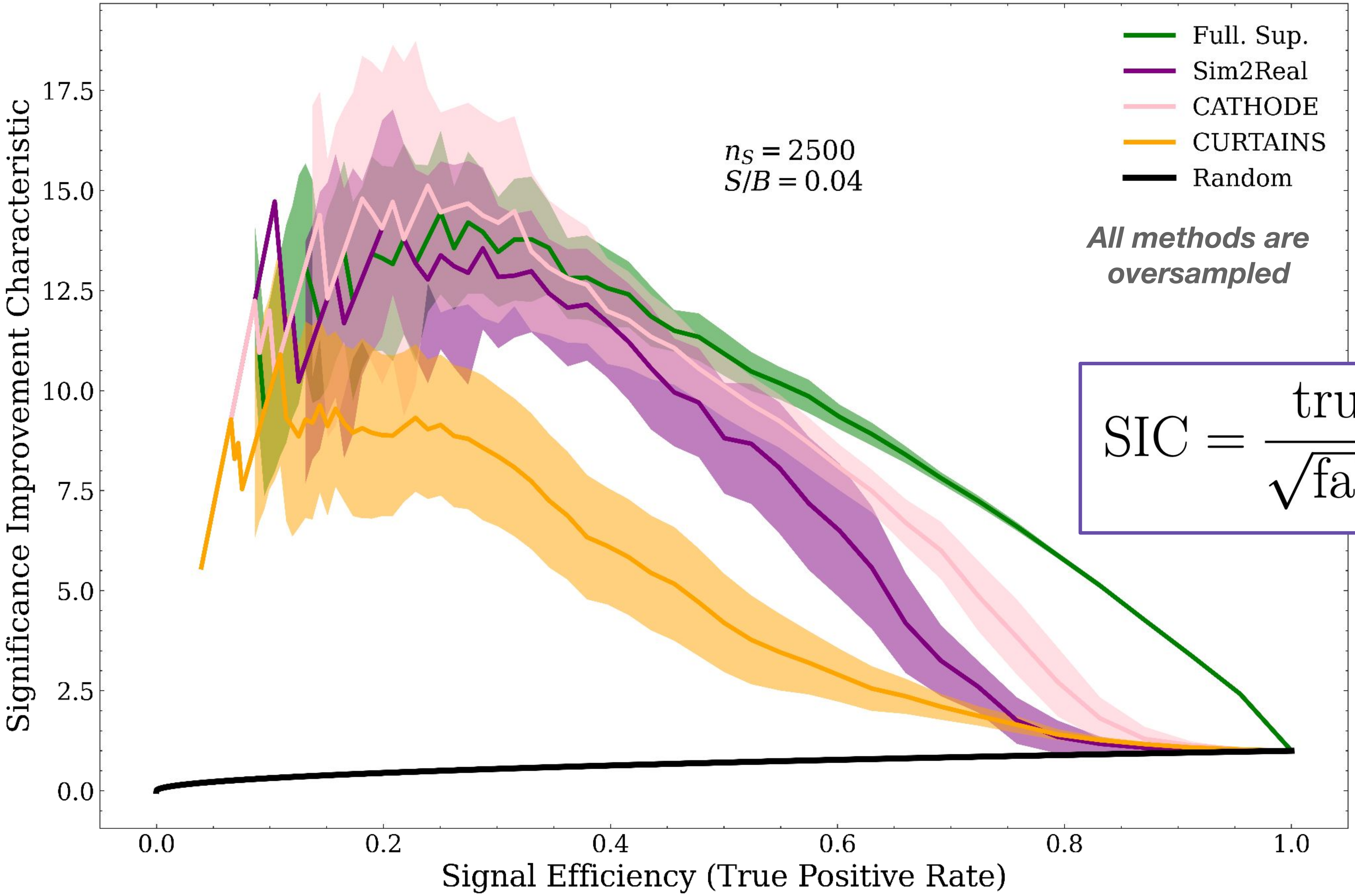
Signal injection procedure

- Inject a known number of signal events into the “data” (Pythia) dataset
    - Signal comes from $Z' \to X(\to qq)Y(\to qq)$, with a new resonance Z' at 3.5 TeV
    - ~ 20% of the events are injected into the SB



- Rerun the flow training procedure, and compare the results with those from the CATHODE and CURTAINS procedures

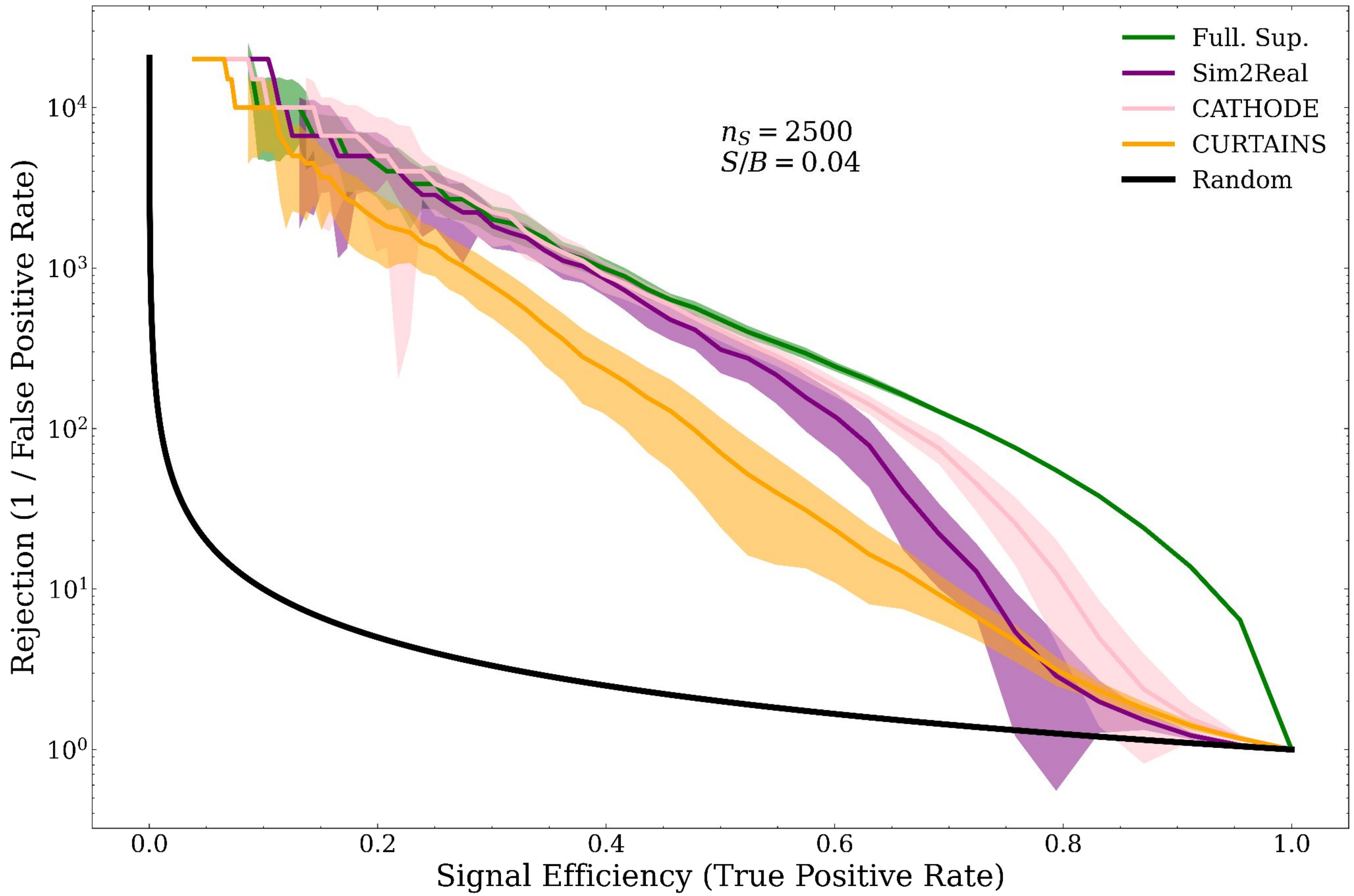
Summary plot: Significance Improvement Characteristic



$$\mathrm{SIC} = \frac{\text{true positive rate}}{\sqrt{\text{false positive rate}}}$$

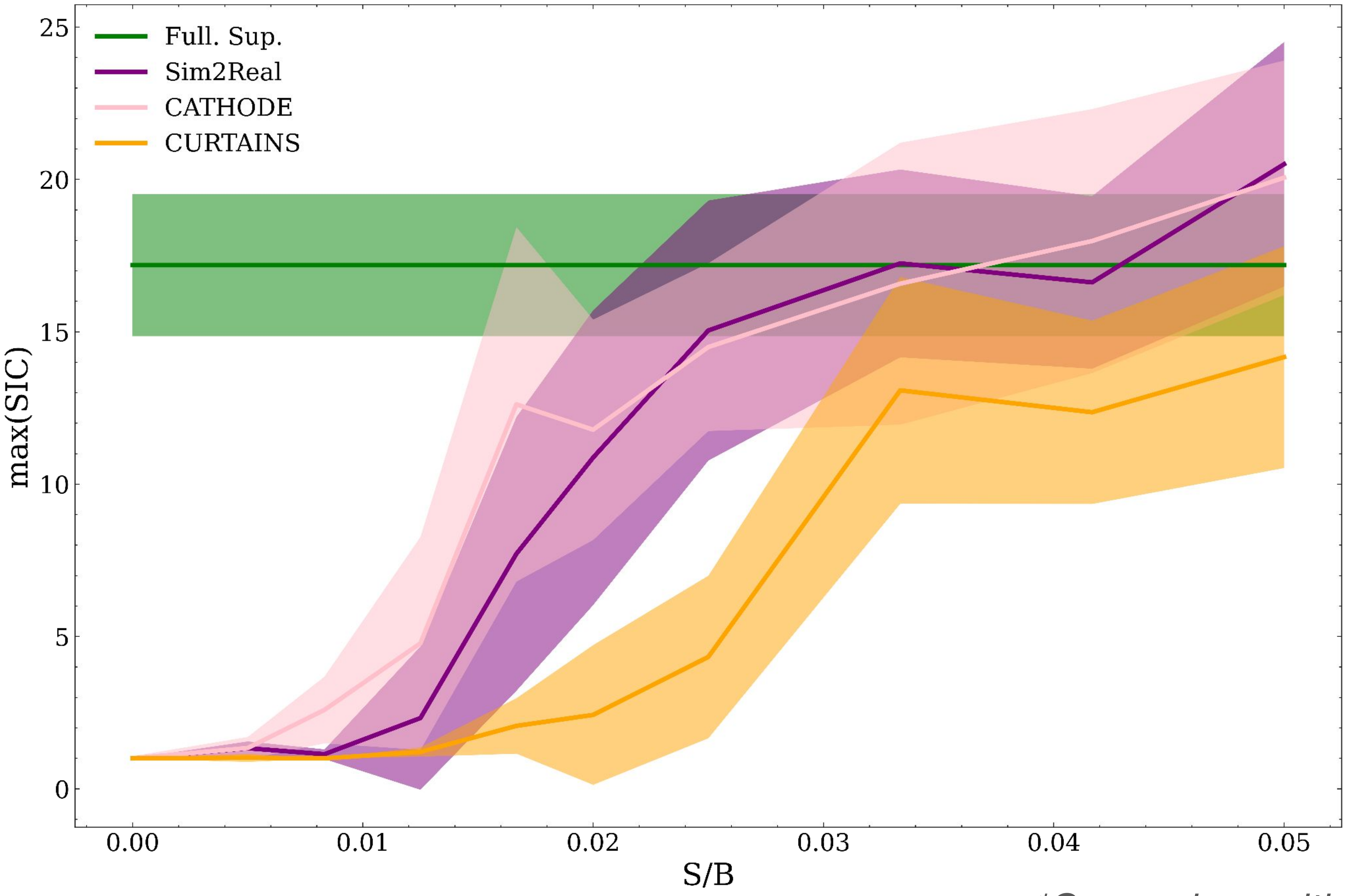
**Comparison with SALAD forthcoming…*

Summary plot: Rejection



**Comparison with SALAD forthcoming…*

Summary plot: Maximum significance improvement



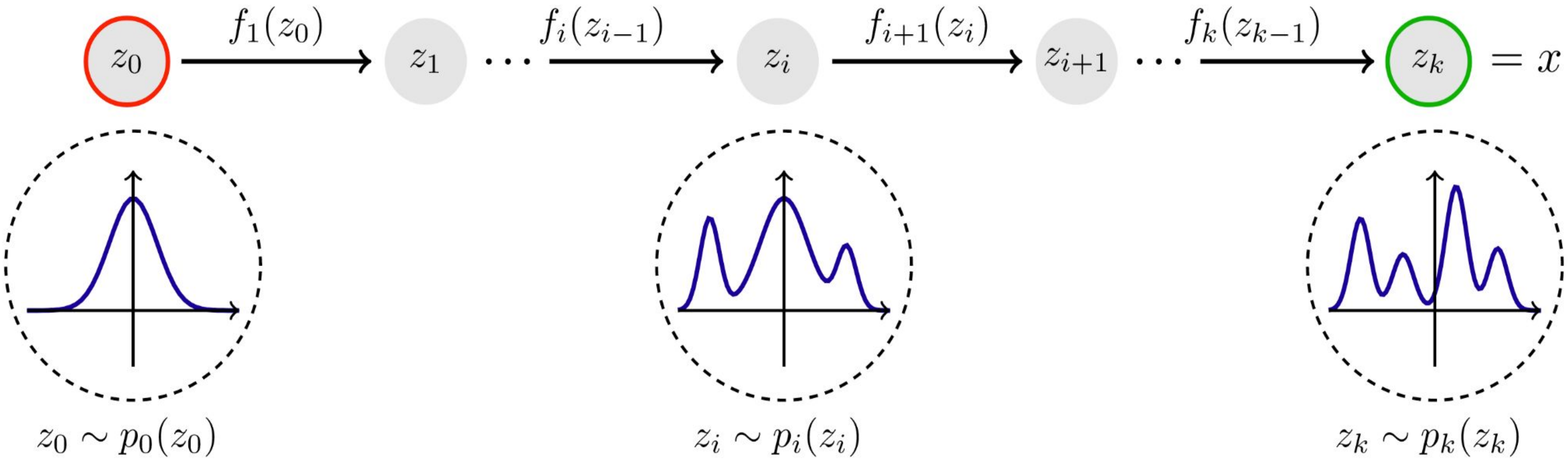
**Comparison with SALAD forthcoming…*

Conclusions and outlook

- Sim2Real is a simulation-augmented method to construct **faithful SM background templates** for resonant AD
  - Simulation gives us a more **informative prior**
  - Feature morphing works well for **low-density regions** of phase space
- Sim2Real, CATHODE, CURTAINS, and SALAD can be treated as a set of **complementary techniques** for a wide range of datasets and resonances

Backups

Normalizing flows learn mappings between probability densities



https://flowtorch.ai/users/

$$\mathcal{L} = \log p(z) + \sum_{i=0}^{k} \log J_i$$

Event band numbers breakdown

Band	GeV Bounds	HERWIG ("Simulation")	PYTHIA ("Data")
SB1	(2900, 3300)	210767	212115
SR	(3300, 3700)	121978	121339
SB2	(3700, 4100)	68609	66646
SB1 + SB2	–	279376	278761

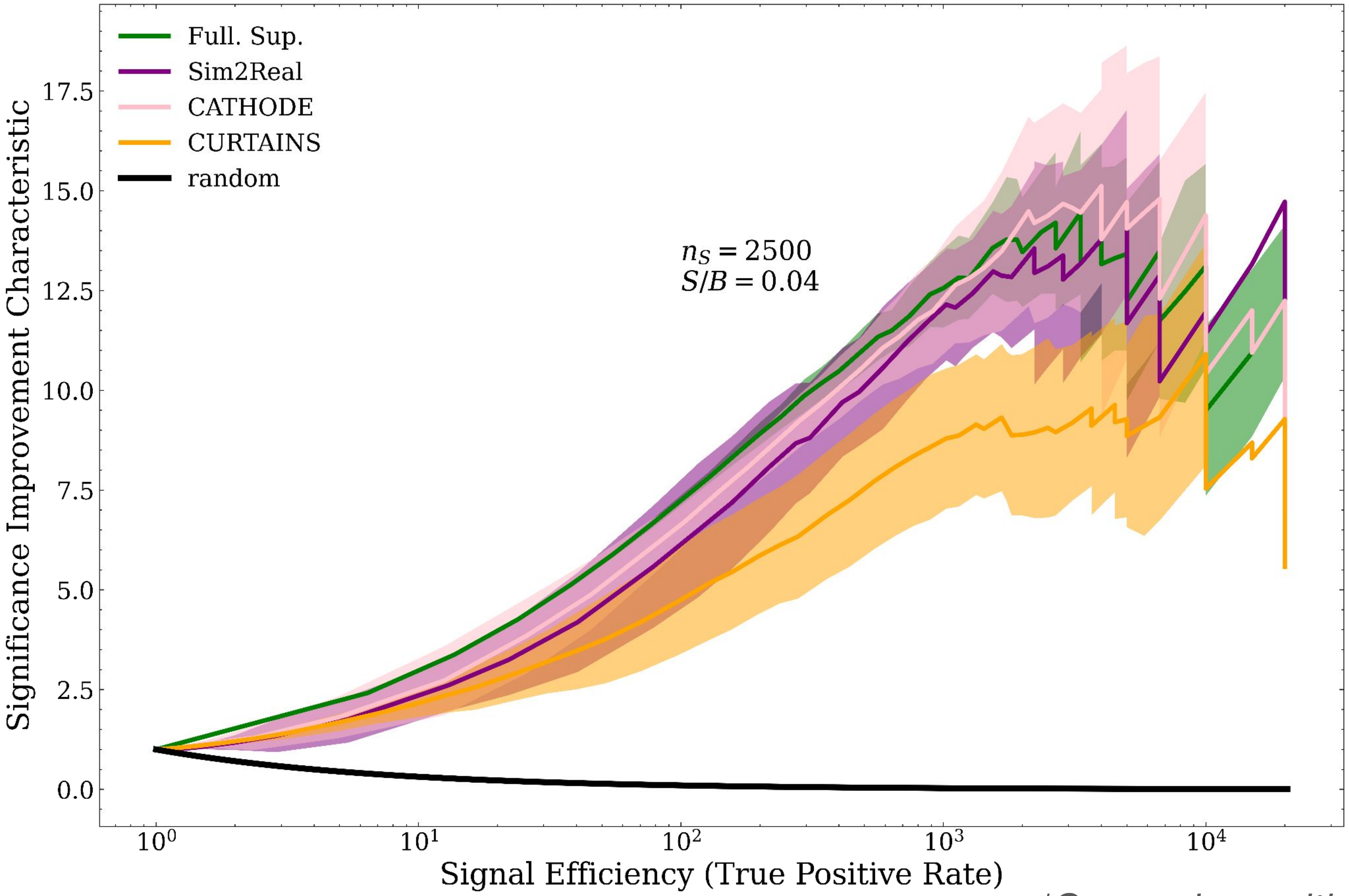
Flow architecture and hyperparameters

The **Base density** flow learns the probability density of simulation, and the **Transport** flow learns to transport between simulation and data densities

Parameter	"Base density" flow	"Transport" flow
Flow type	Autoregressive	Coupling
Spline	Piecewise RQ	Piecewise RQ
Num. MADE blocks	15	8
Num. layers	1	2
Num. hidden features	128	16
Epochs	100	100
Batch size	128	256
Learning rate	1e-4	5e-4
Weight Decay	1e-4	1e-5

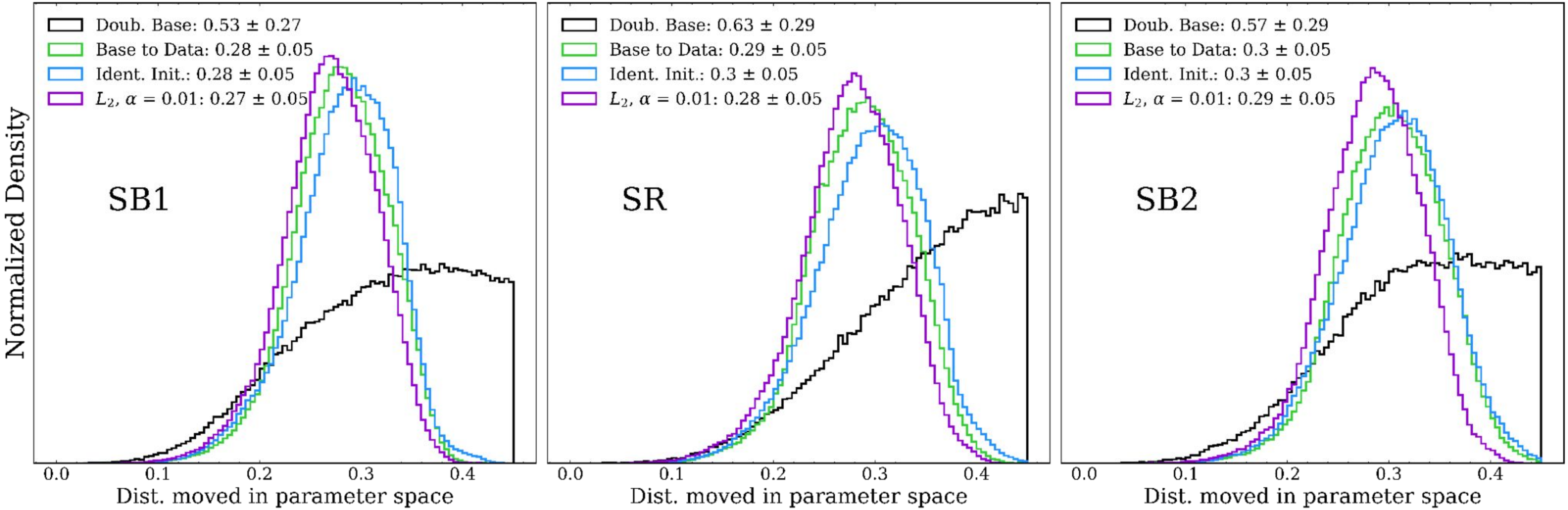
All flows are implemented with the `nflows` package in `Pytorch`. Training is optimized with `AdamW`, and the learning rate is cosine-annealed. The model from the epoch with the lowest validation loss is used for evaluation.

Summary plot: SIC vs. Rejection



**Comparison with SALAD forthcoming…*

Optimal transport: distance traveled in feature space



**Formal results to be presented at the Machine Learning and the Physical Sciences workshop at NeurIPS 2022*

Optimal transport: SIM → DAT transformer performance

Band	Double Base	Base to Data	Identity Init.	L_2 ($\alpha = 10^{-2}$)
OB1	0.630 ± 0.024	0.511 ± 0.003	0.508 ± 0.004	0.507 ± 0.002
SB1	0.501 ± 0.000	0.502 ± 0.001	0.501 ± 0.000	0.502 ± 0.001
SR	0.553 ± 0.011	0.503 ± 0.001	0.503 ± 0.001	0.503 ± 0.000
SB2	0.501 ± 0.000	0.503 ± 0.001	0.503 ± 0.001	0.502 ± 0.001
OB2	0.594 ± 0.030	0.506 ± 0.002	0.507 ± 0.004	0.507 ± 0.003

**Formal results to be presented at the Machine Learning and the Physical Sciences workshop at NeurIPS 2022*