Ephemeral Learning: Augmenting Triggers with online trained normalising flows

Anja Butter, **Sascha Diefenbacher**, Gregor Kasieczka, Benjamin Nachman, Tilman Plehn, David Shih, Ramon Winterhalder

sascha.daniel.diefenbacher@uni-hamburg.de

CERN IML 12.05.2022

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Introduction

- LHC data rate is incredibly high
  - At 40 MHz ~40 Tb/s

Introduction

- LHC data rate is incredibly high
  - At 40 MHz ~40 Tb/s
  - Cargo container filled with hard drives (100 Pb)

Introduction

- LHC data rate is incredibly high
  - At 40 MHz ~40 Tb/s
  - Cargo container filled with hard drives (100 Pb) … every 41 minutes

Introduction

- LHC data rate is incredibly high
  - At 40 MHz ~40 Tb/s
  - Cargo container filled with hard drives (100 Pb) … every 41 minutes
  - 35 Containers per day

Introduction

- We can not save everything
  - Reduce data rate: Triggers, save only the interesting events
    storage ~ trigger-rate x luminosity

Introduction

- We can not save everything
  - Reduce data rate: Triggers, save only the interesting events
    storage ~ trigger-rate x luminosity
  - Reduce per event size: Save only high level observables
    storage ~ (reduced size/full size) x luminosity

Introduction

- We can not save everything
  - Reduce data rate: Triggers, save only the interesting events
    storage ~ trigger-rate x luminosity
  - Reduce per event size: Save only high level observables
    storage ~ (reduced size/full size) x luminosity
  - Save only histograms of observables
    storage ~ number of bins, lose non-specified correlations

Introduction

- We can not save everything
  - Reduce data rate: Triggers, save only the interesting events
    storage ~ trigger-rate x luminosity
  - Reduce per event size: Save only high level observables
    storage ~ (reduced size/full size) x luminosity
  - Save only histograms of observables
    storage ~ number of bins, lose non-specified correlations
  - Encode data in generative model
    storage ~ network weights

Introduction

LHC data

Trained model

Online Training

This is quite ambitious

Online Training

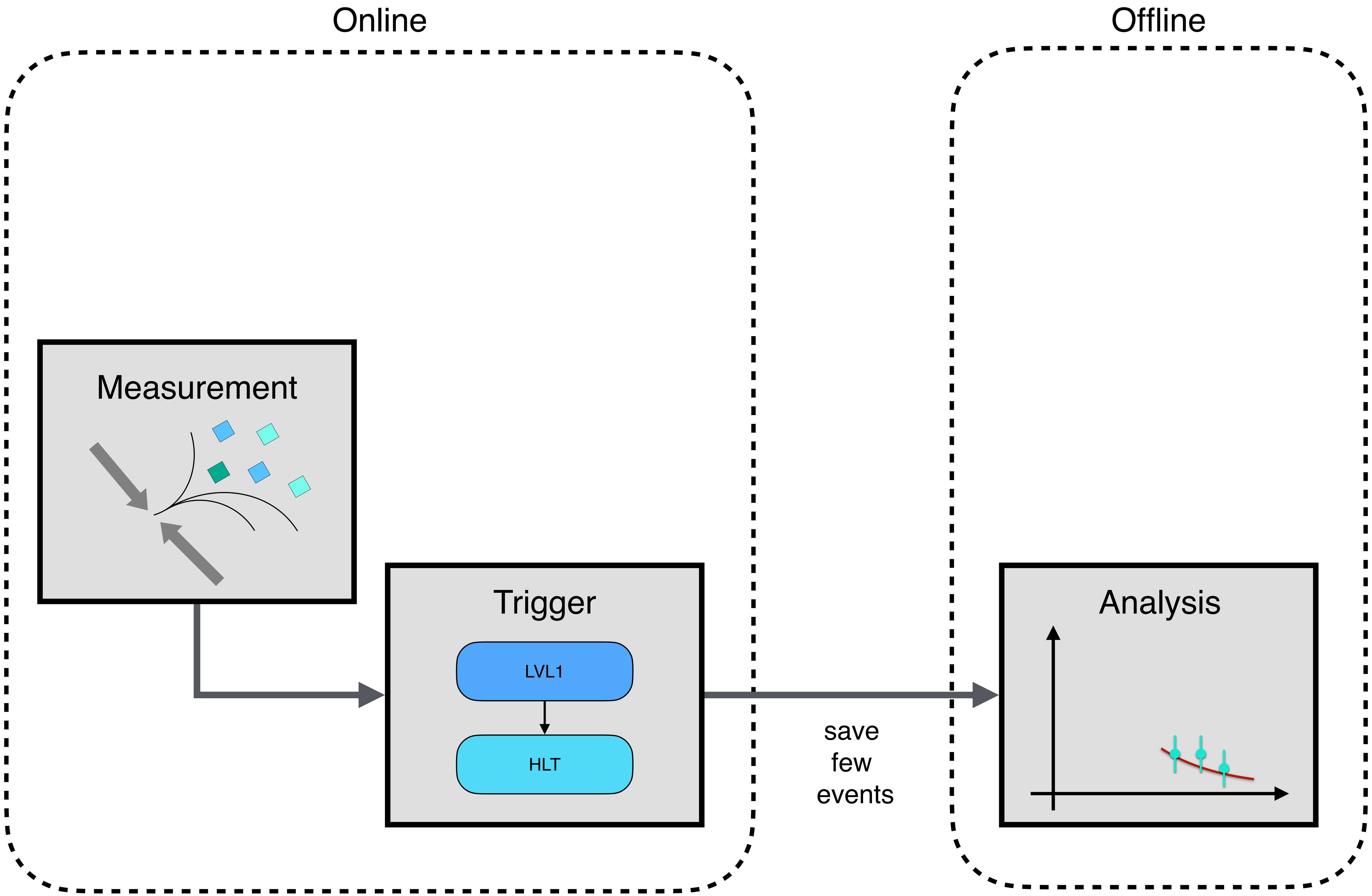
This is quite ambitious

- Lets start smaller

Online Training

This is quite ambitious

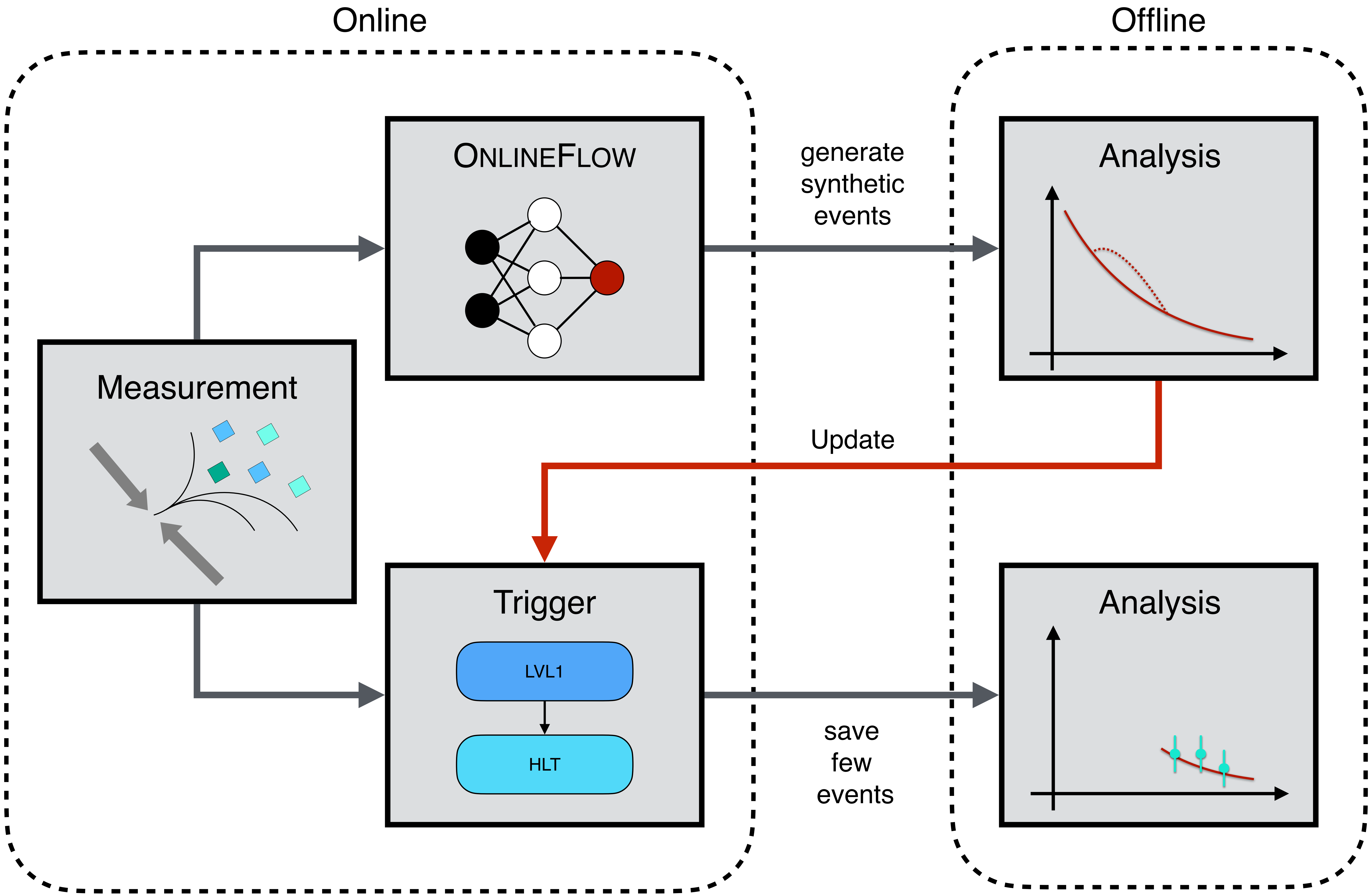
- Lets start smaller
- Online model as additional scouting tool
- Investigate regions ignored by triggers



Online Training

This is quite ambitious

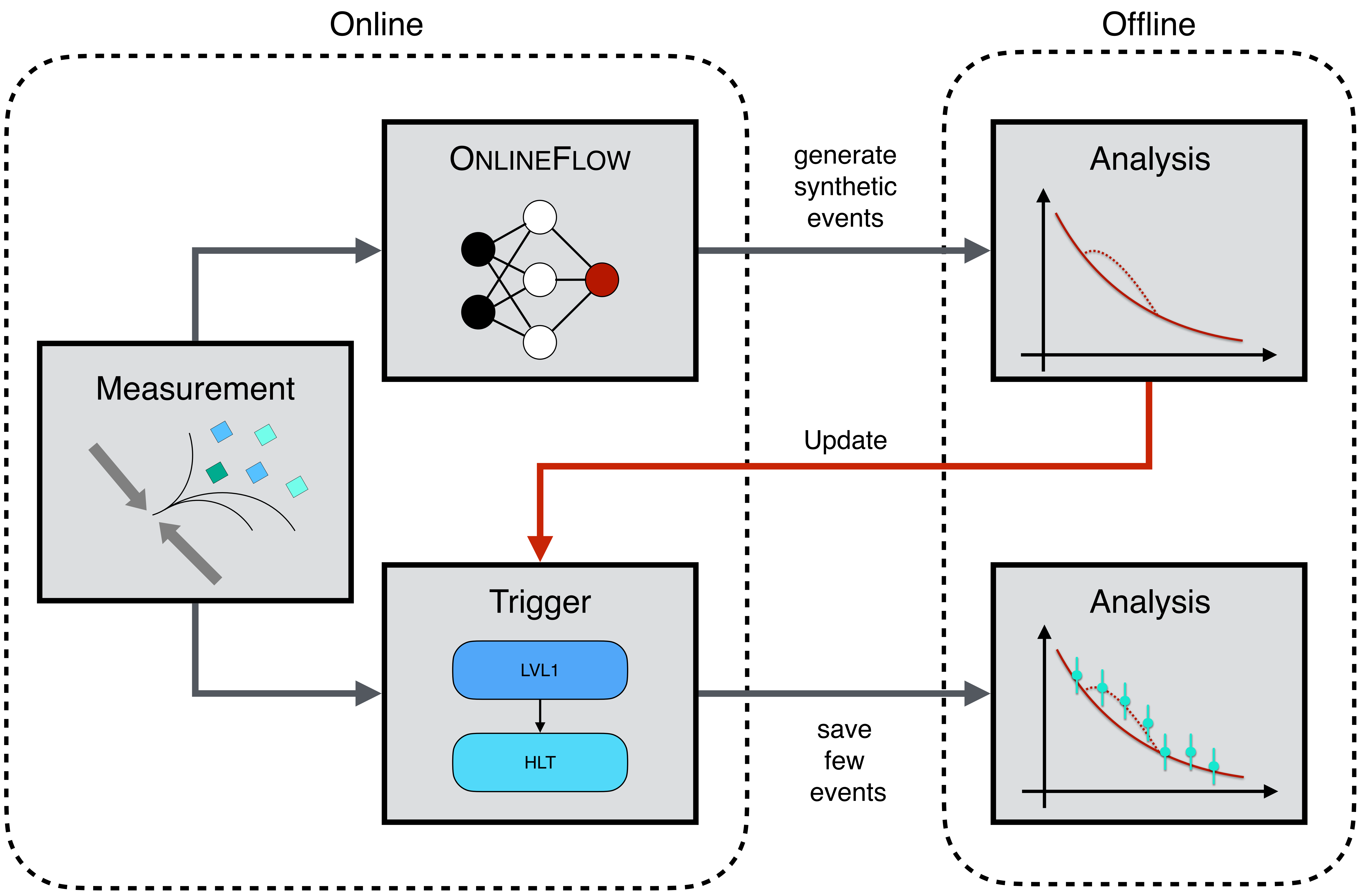
- Lets start smaller
- Online model as additional scouting tool
- Investigate regions ignored by triggers



Online Training

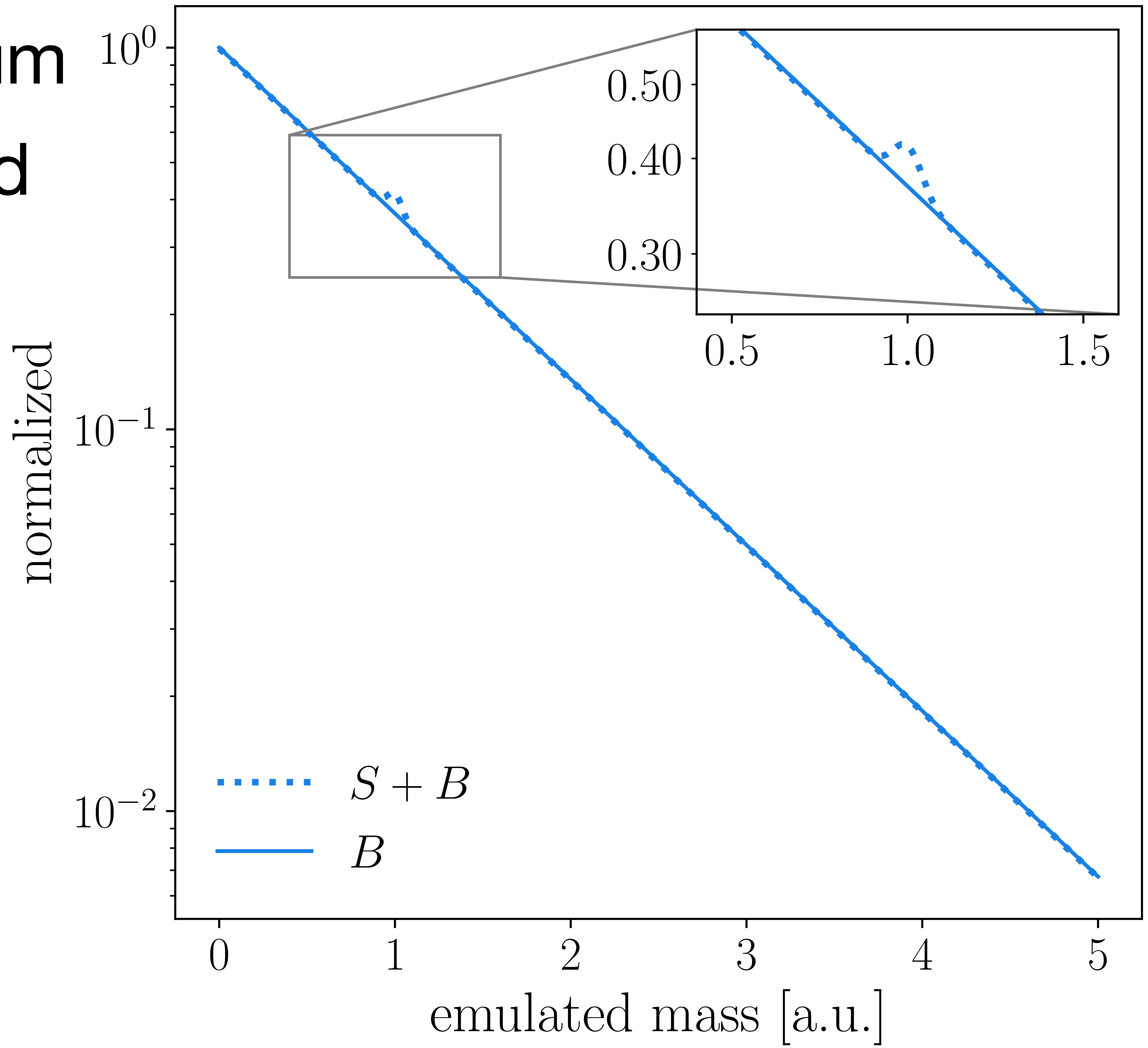
This is quite ambitious

- Lets start smaller
- Online model as additional scouting tool
- Investigate regions ignored by triggers



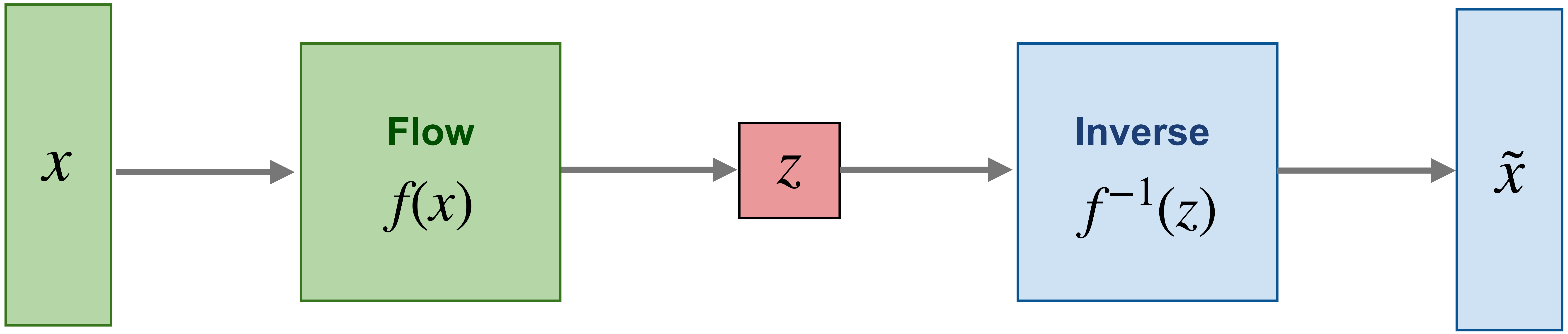
Proof of Concept

- 1 dimension toy mass spectrum
- Exponential falling background
- Gaussian peak signal
- Train generative model



Generative Model

Normalising Flow

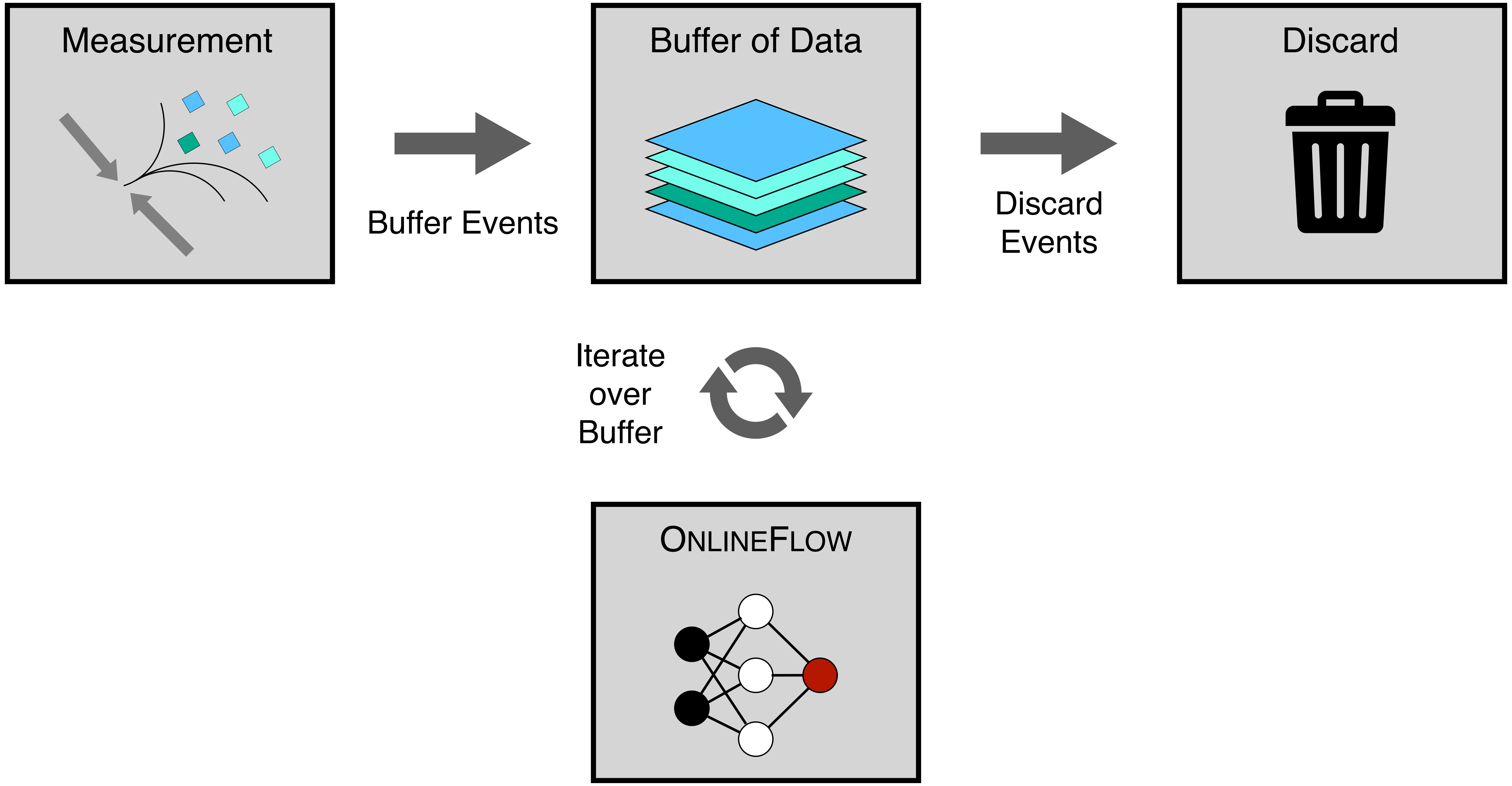


- Train invertible transformation to map data to latent space
- Use inverse to generate new data from new latent samples

Online Training

Troubles with online training:

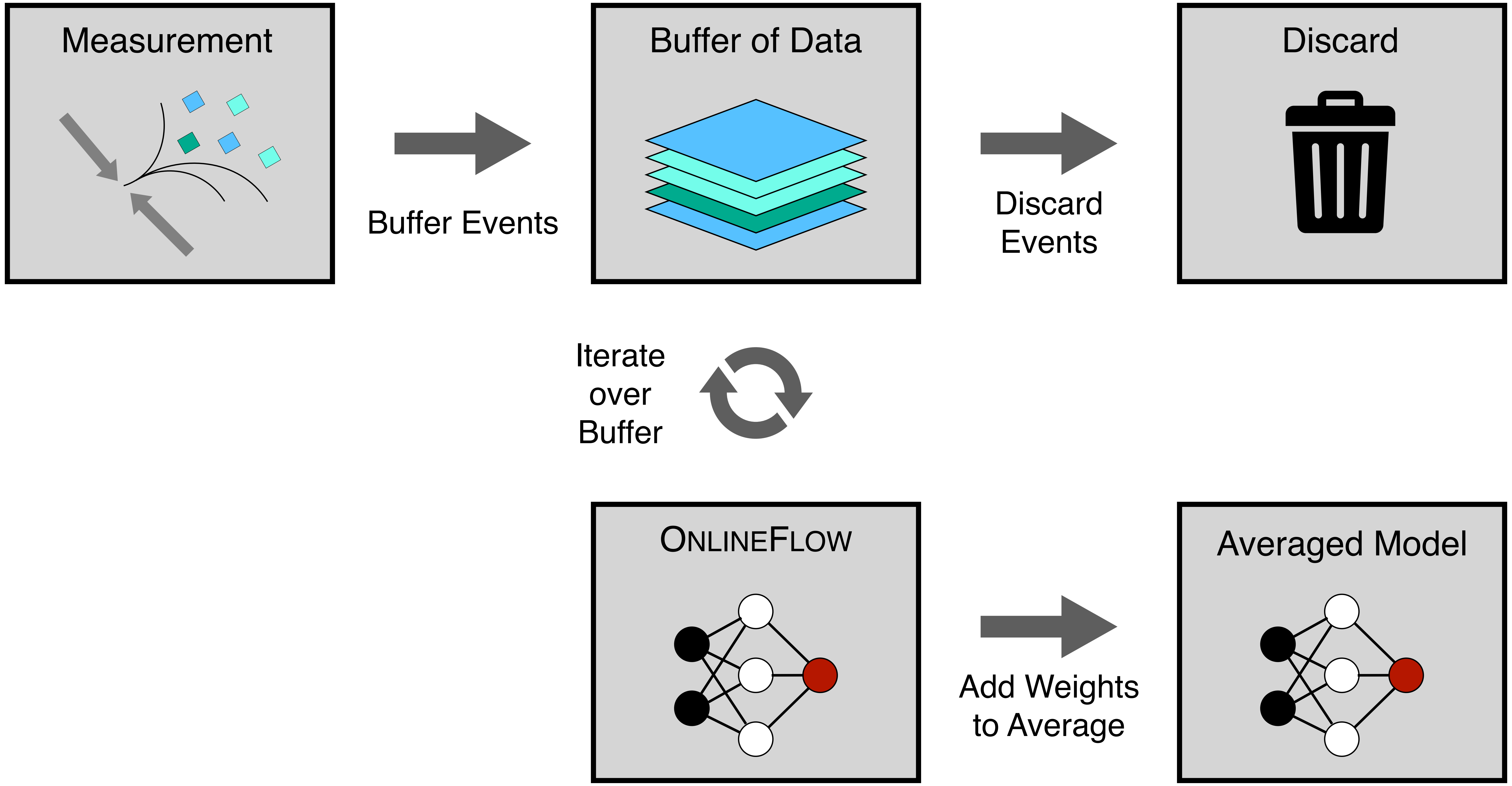
- Models designed for parallel training on batches
- Points should be seen more than once



Online Training

Model will be biased towards more recent buffers

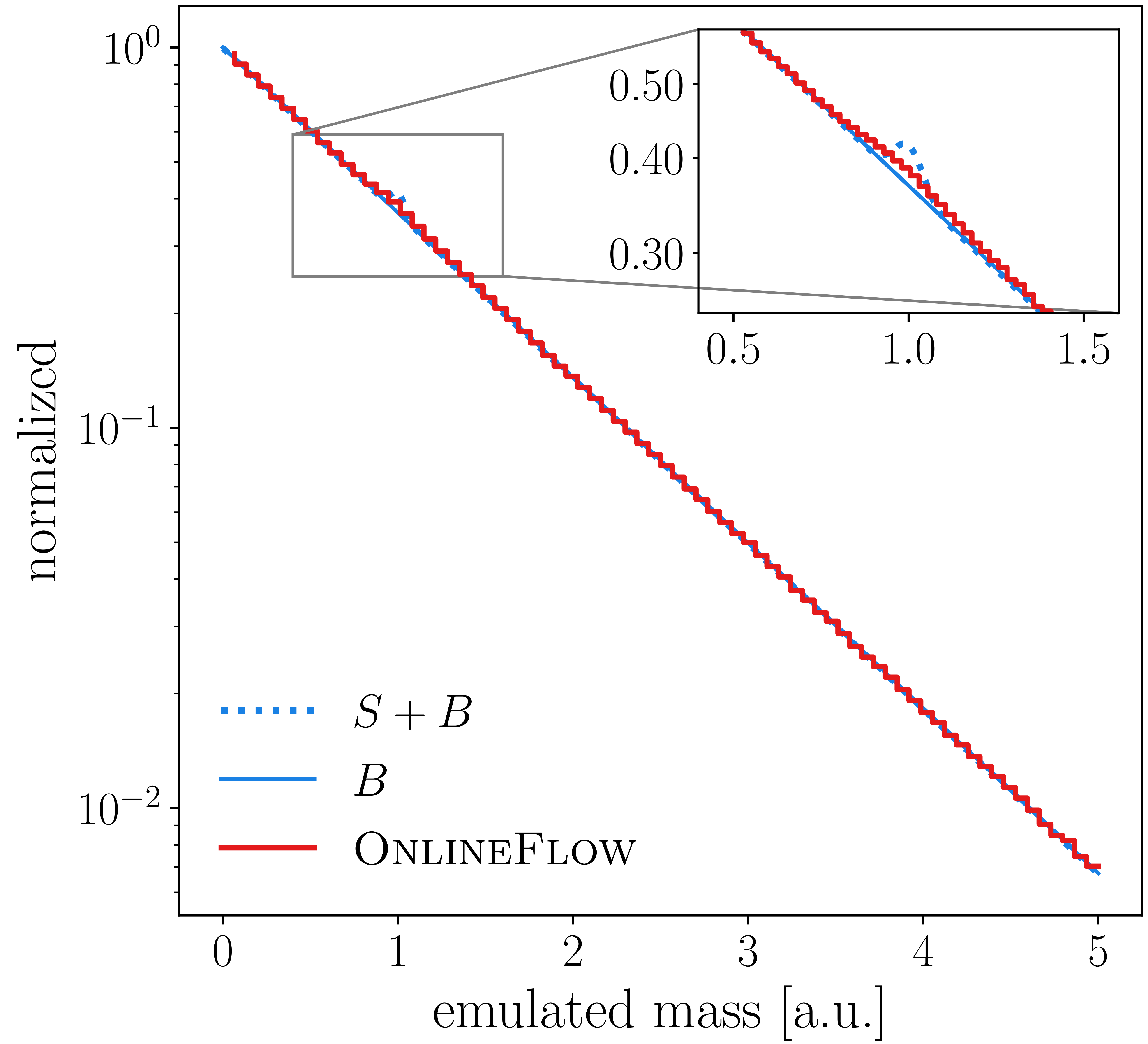
- Stochastic Weight Averaging (SWA)
- Keep running average over model during training



Proof of Concept

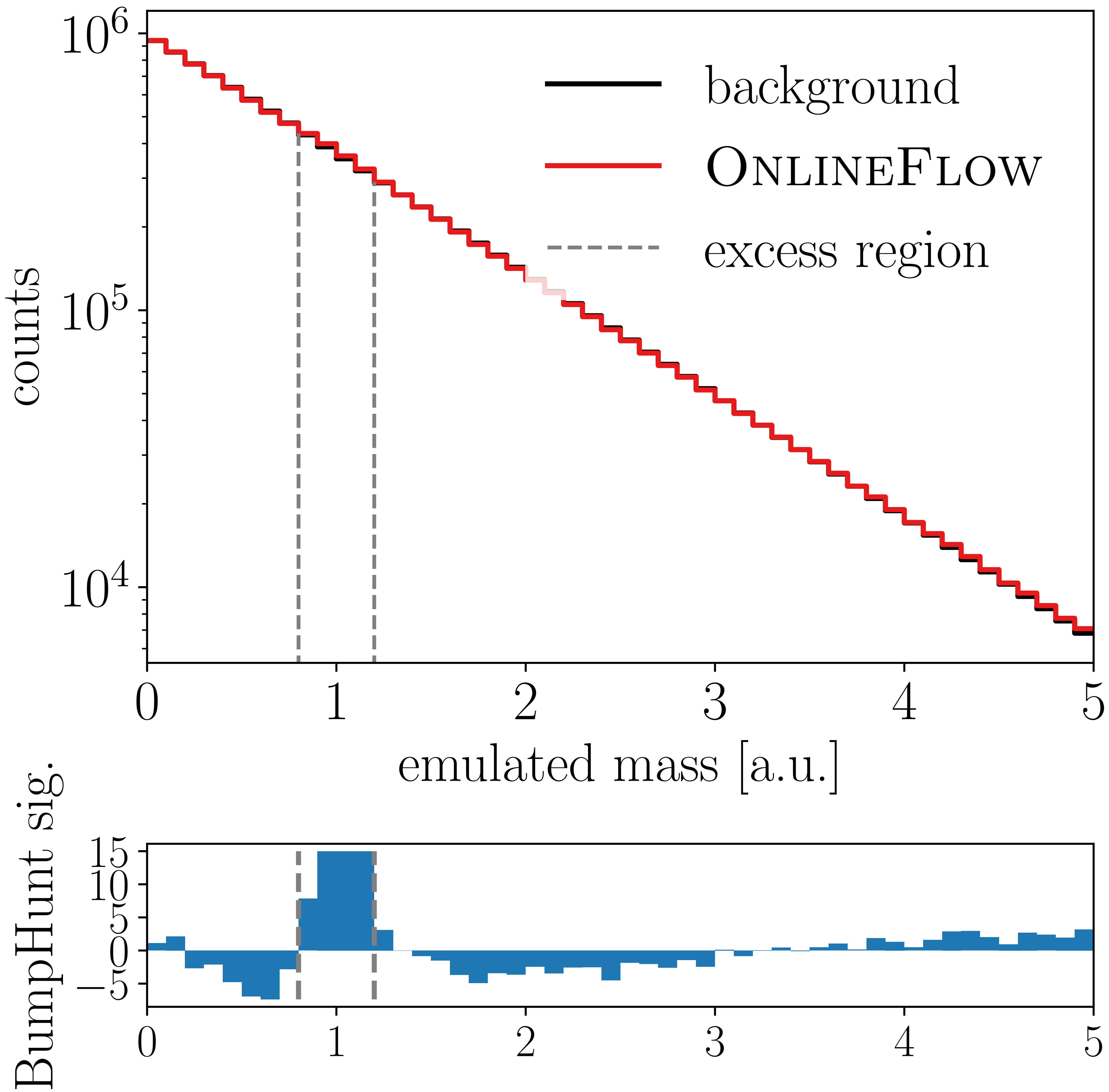
- Train flow on training data
- Sample new events from flow
- Signal not modeled perfectly
- Shaping is visible
- Extract signal from flow?

➡ Run Bumphunt on flow



Proof of Concept

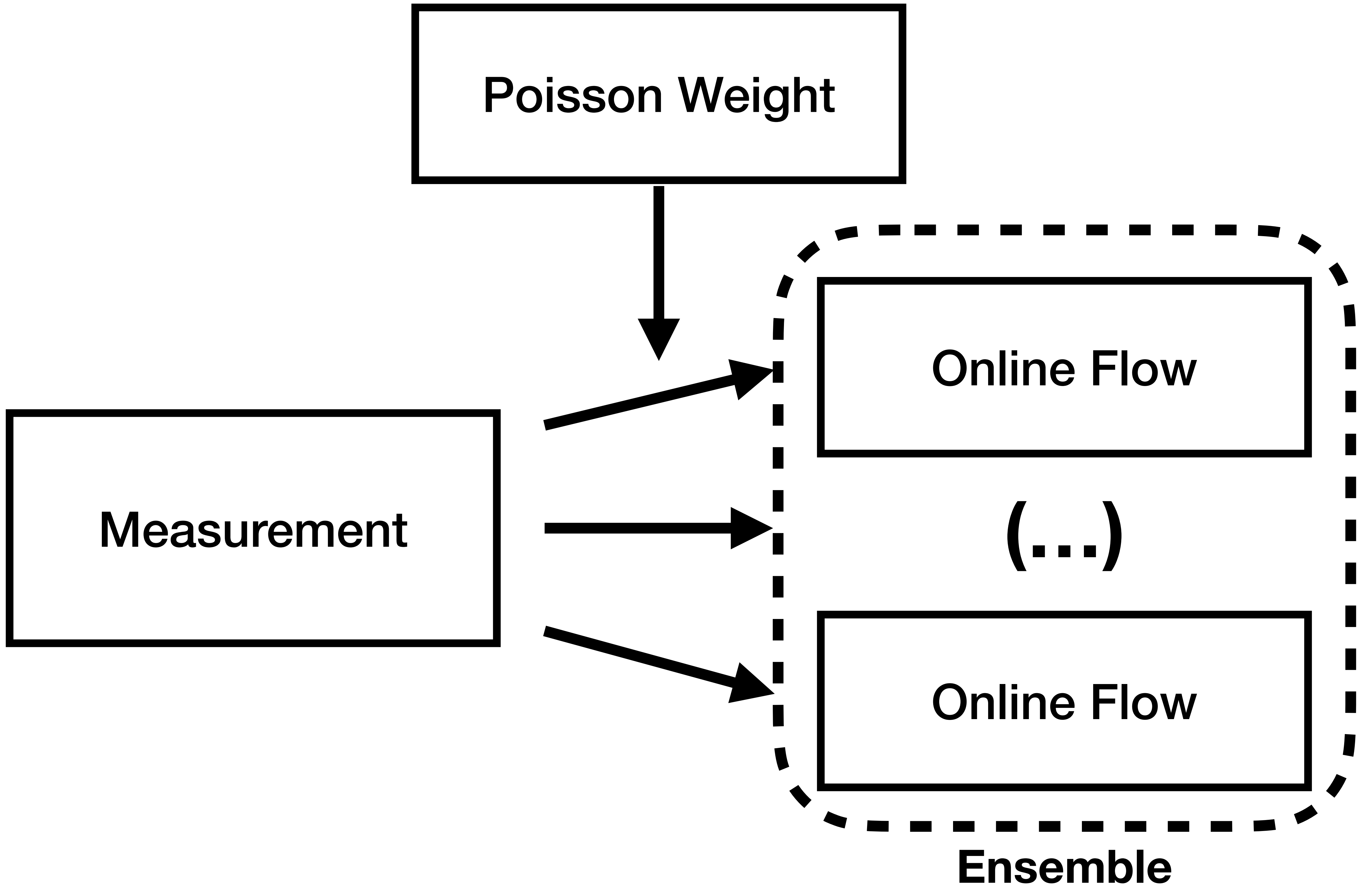
- Analyse data
- Fit exponential function to data to get background model
- Run bump hunt (pybumphunter)
- What is our significance?
- No Poisson uncertainty
- Cannot use $\frac{S}{\sqrt{B}}$



Louis Vaslin, Julien Donini, **pyBumpHunter,** https://pypi.org/project/pyBumpHunter/

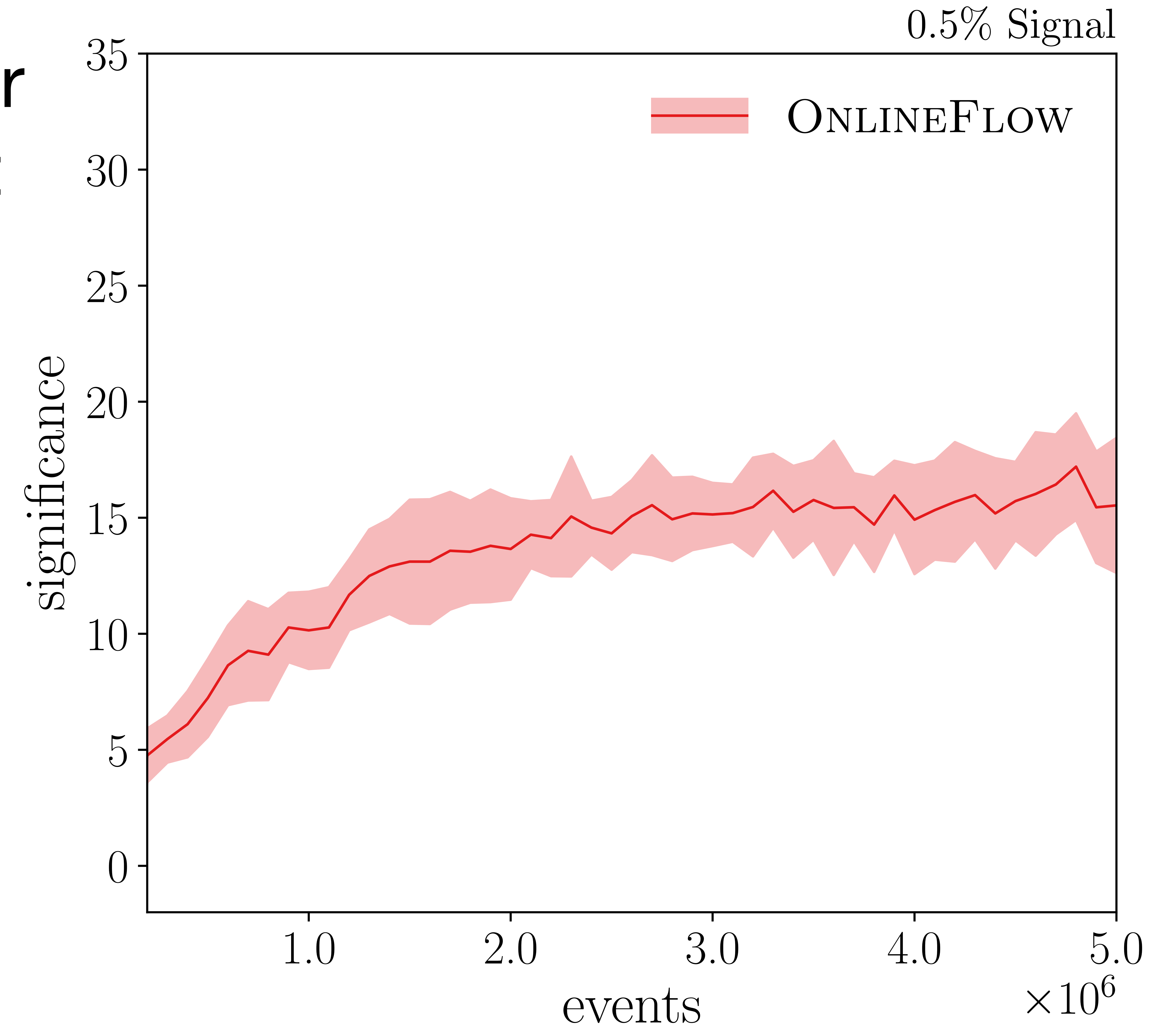
Proof of Concept

- Train **bootstraped ensemble of Flows**
  - Pass resampled version of dataset to each flow
  - Emulate online with Poisson weight for each event
- Compare flows within ensemble
  - Estimate uncertainty from standard deviation of ensemble predictions



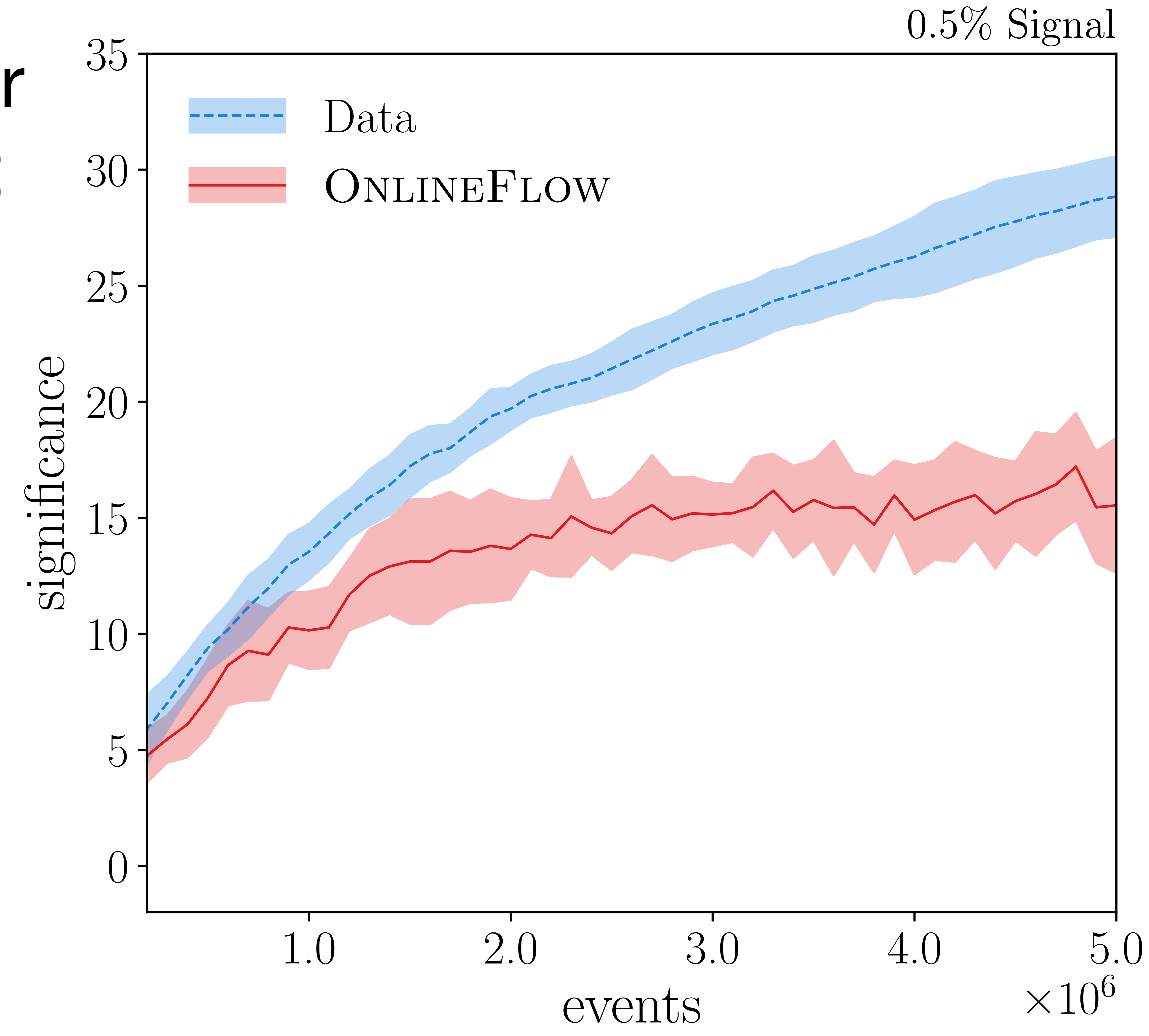
Proof of Concept

- Plot as function of the number of events to see development



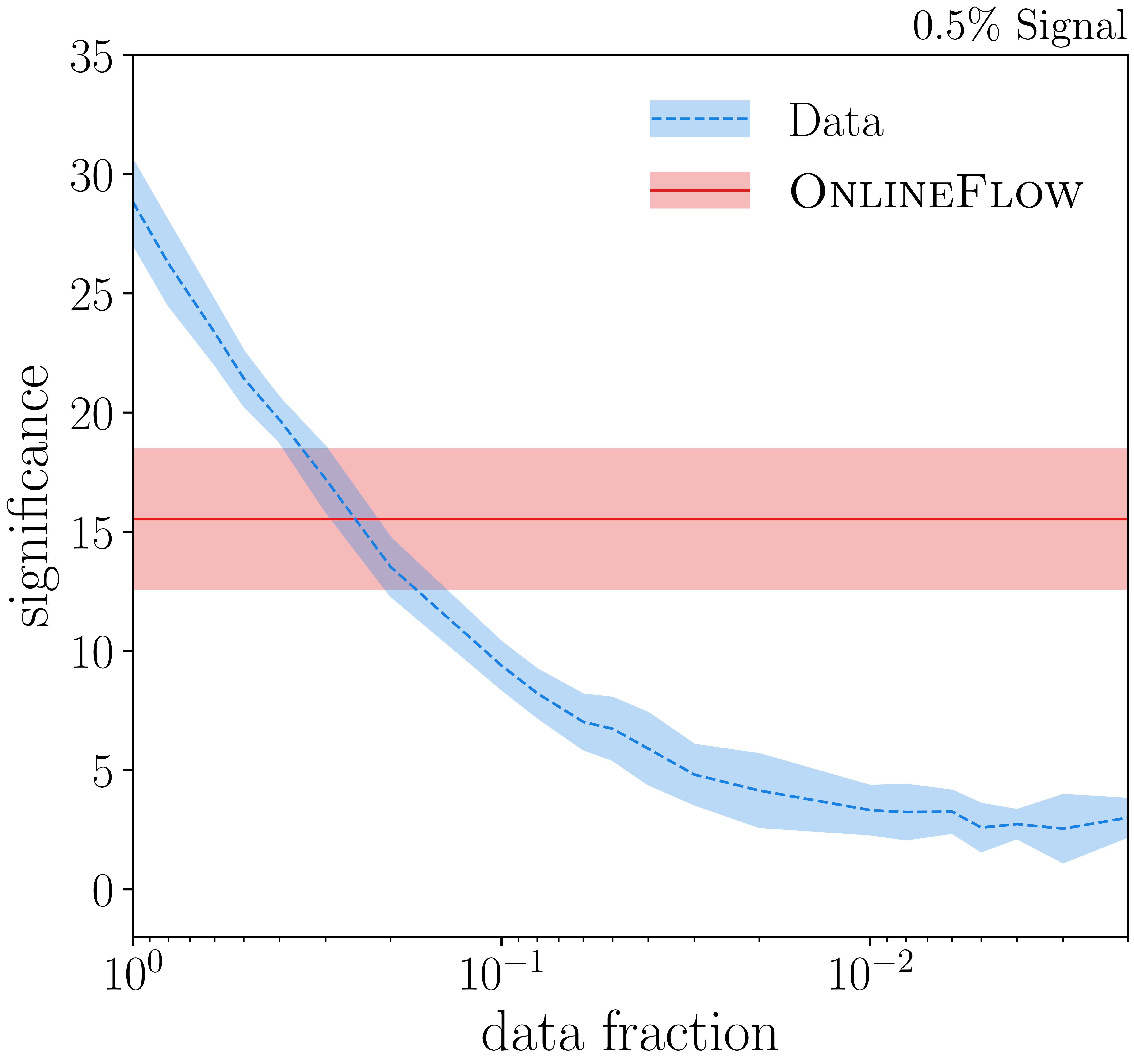
Proof of Concept

- Plot as function of the number of events to see development
- Run same analysis in the training data itself
- Compare significance
- Flow similar shape as data
- Reaches high significance



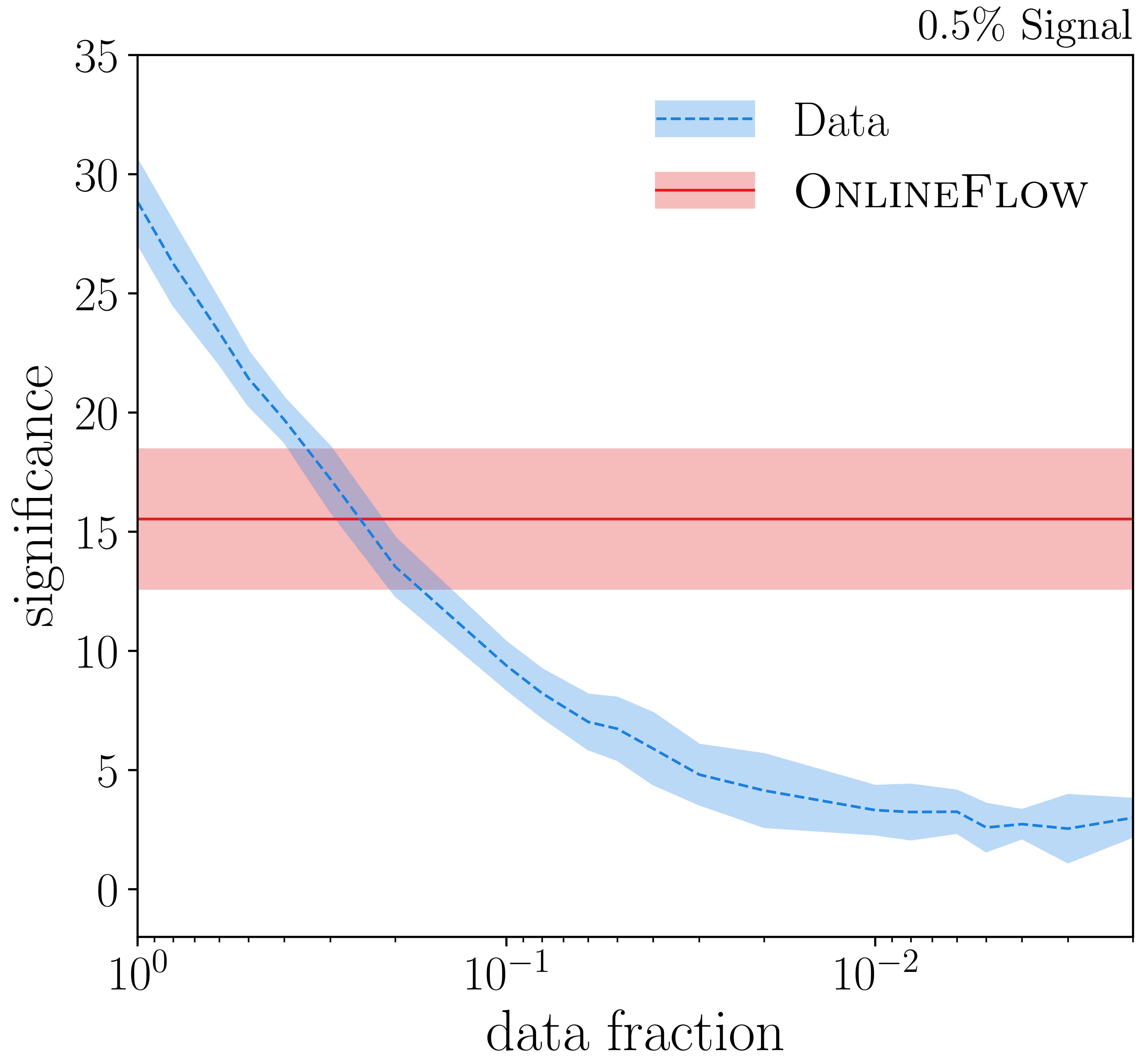
Proof of Concept

- Compare to fractions of data
- Crossover at 25%
- OnlineFlow significance equivalent to 25% of full data



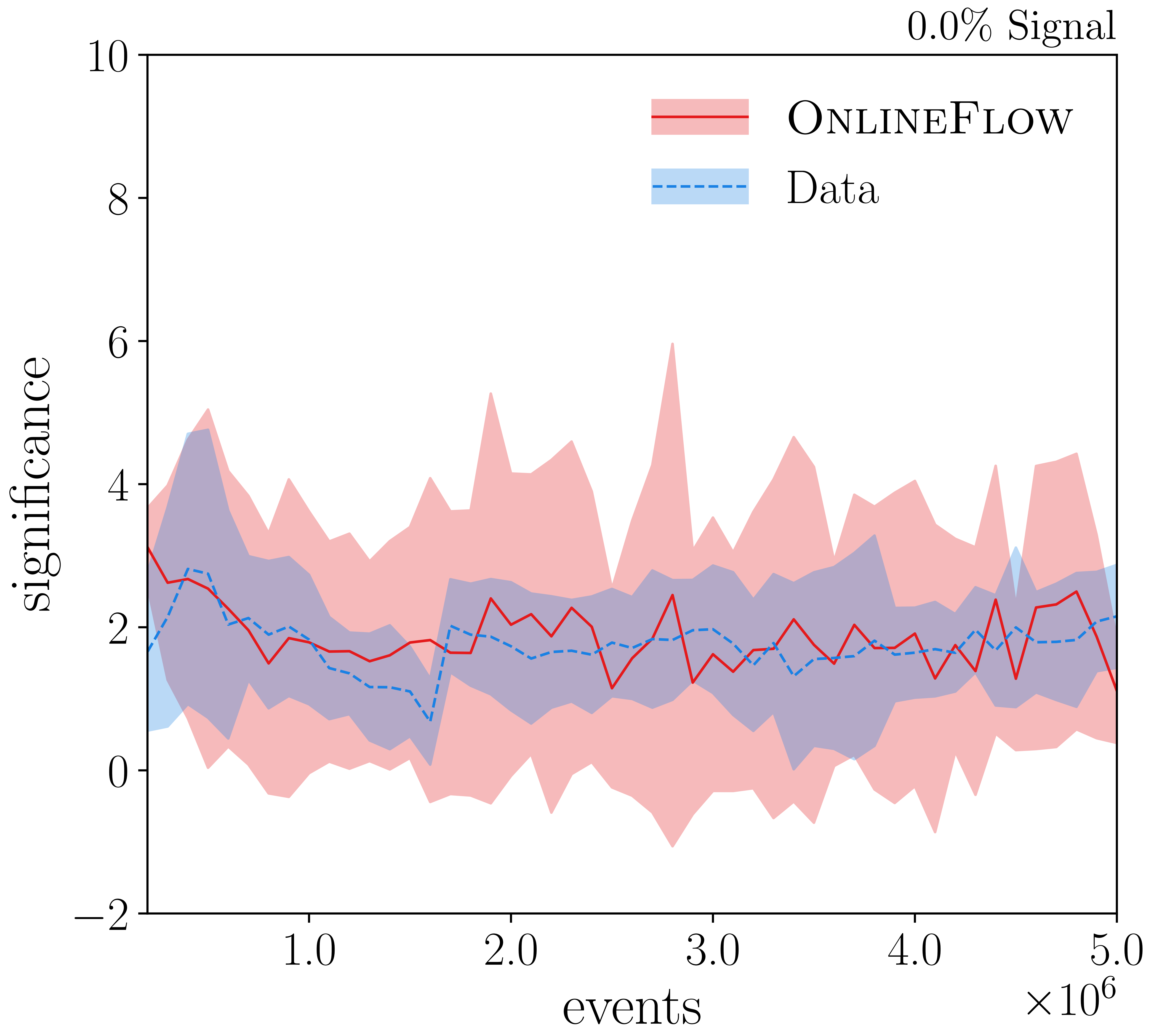
Proof of Concept

- False positives?



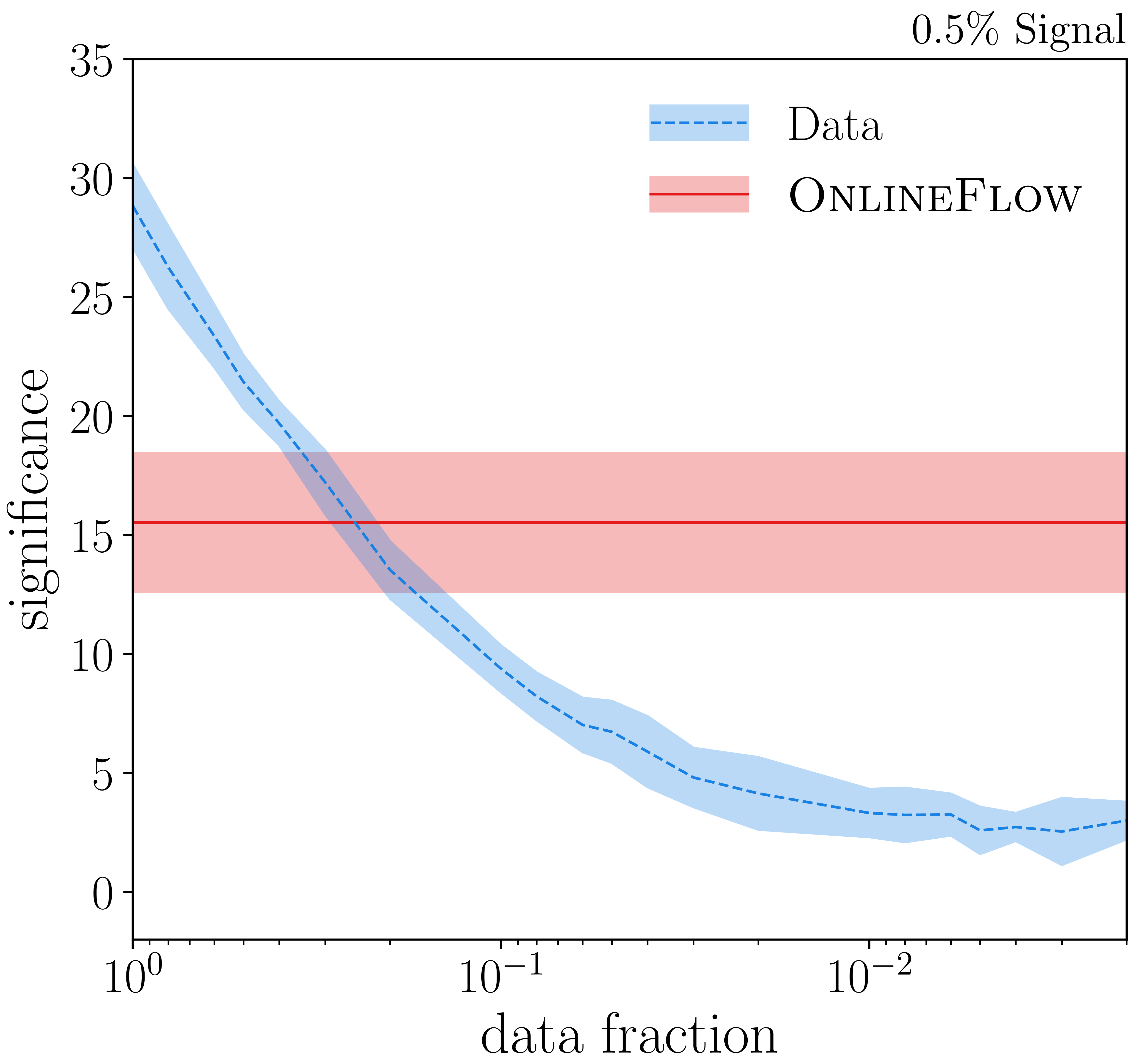
Proof of Concept

- False positives?
- Run same training on only background without signal
- Negligible significance in direct search
- Flow significance nearly identical



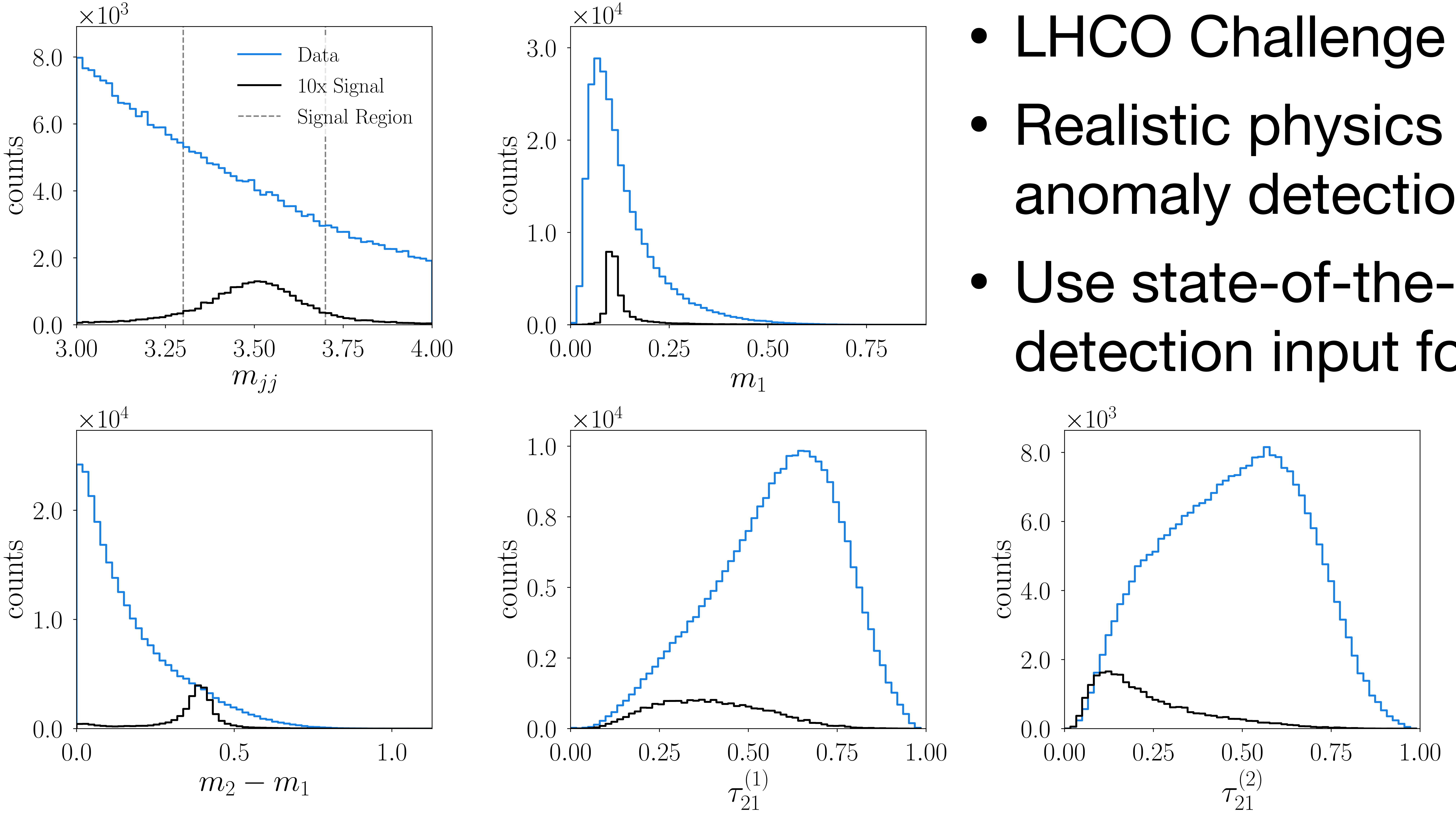
Conclusion

- Tested OnlineFlow approach on 1D proof-of-concept data
  - Promising results
- Check for Full paper check: https://arxiv.org/abs/2202.09375
  - Higher dimensional case on physics dataset



Thank You

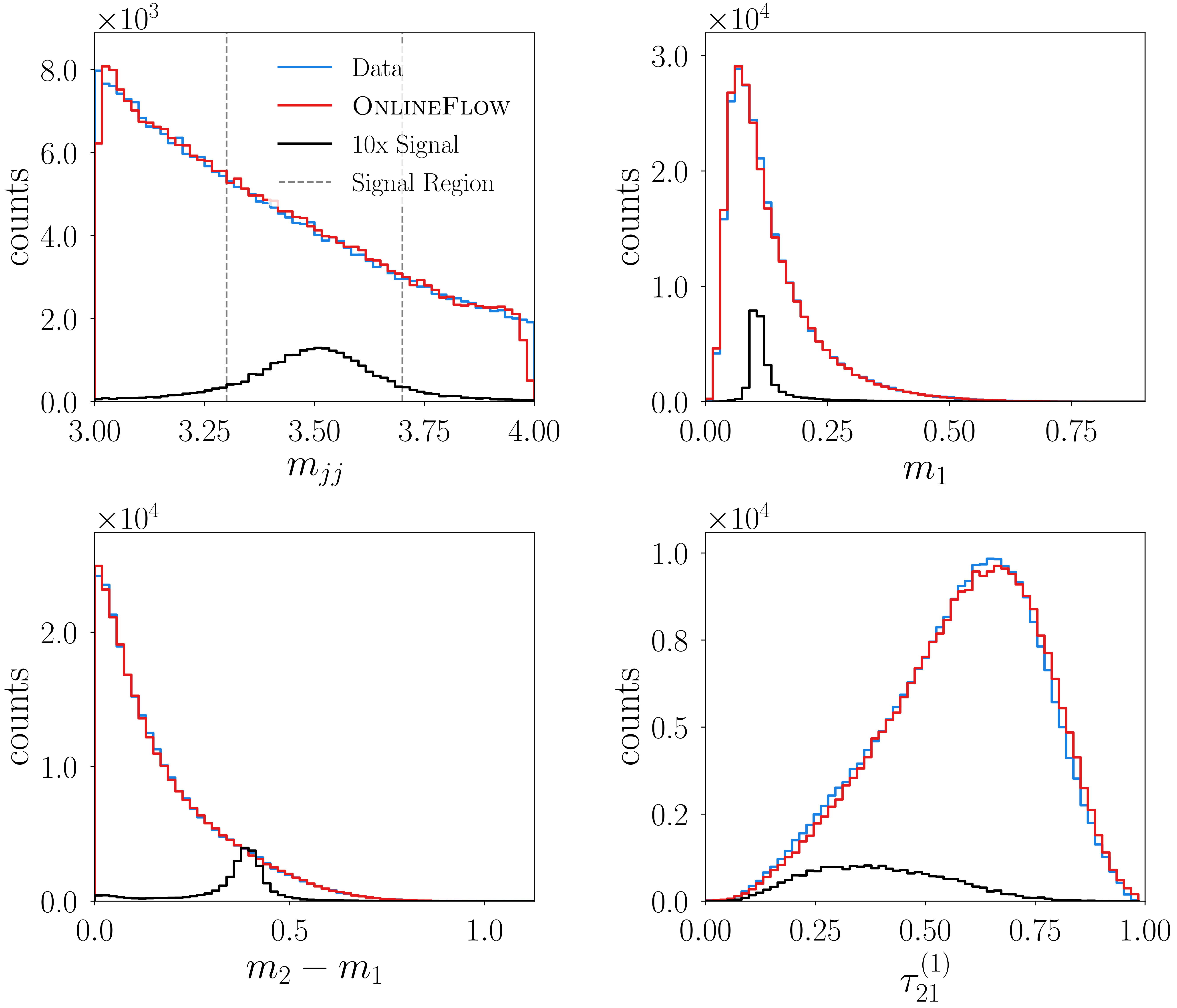
Higher Dimensional Case

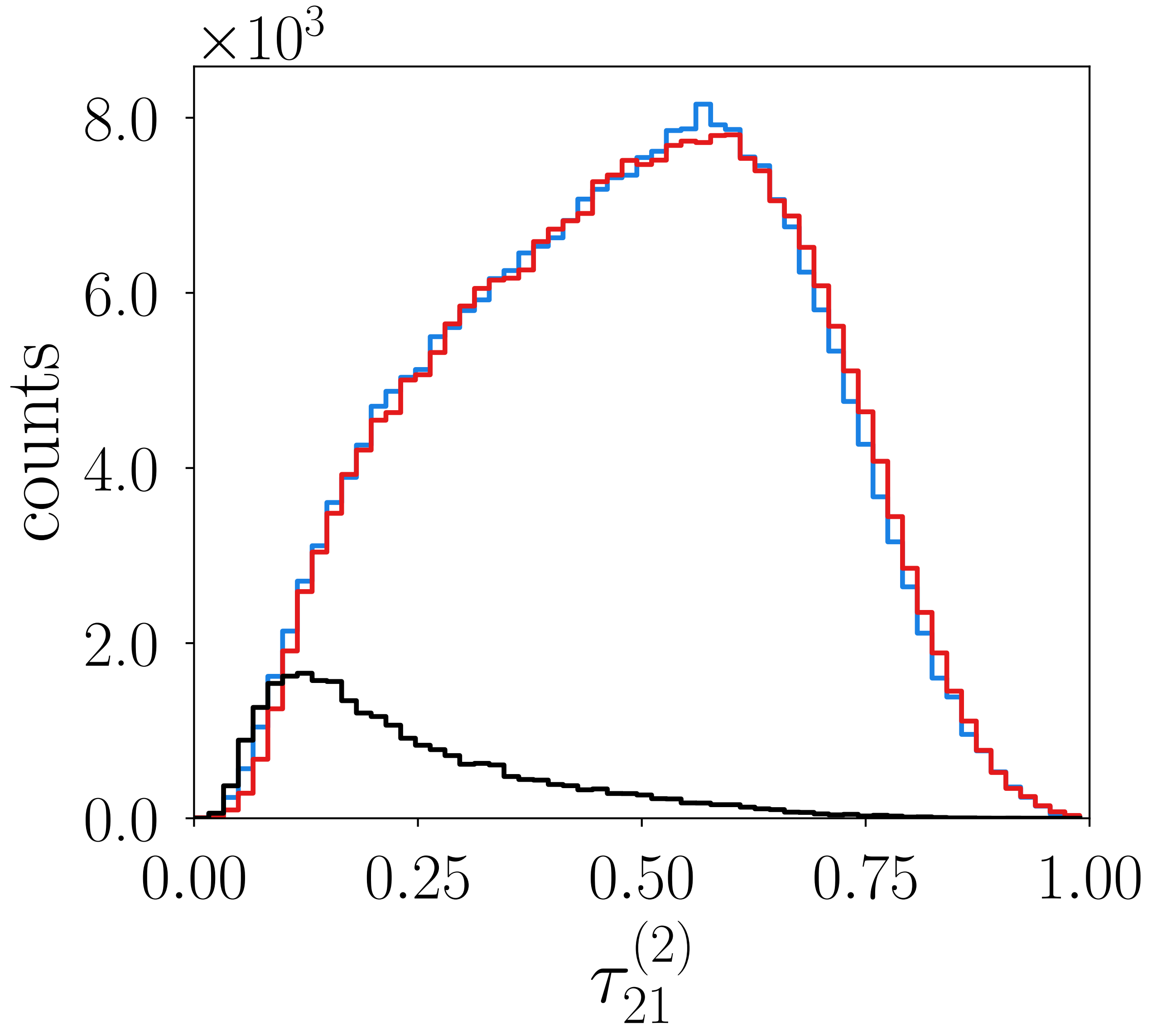


- LHCO Challenge dataset
- Realistic physics case for anomaly detection
- Use state-of-the-art anomaly detection input format

Kasieczka et al.: **The LHC Olympics 2020: A Community Challenge for Anomaly Detection in High Energy Physics** kl 2101.08320

Higher Dimensional Case

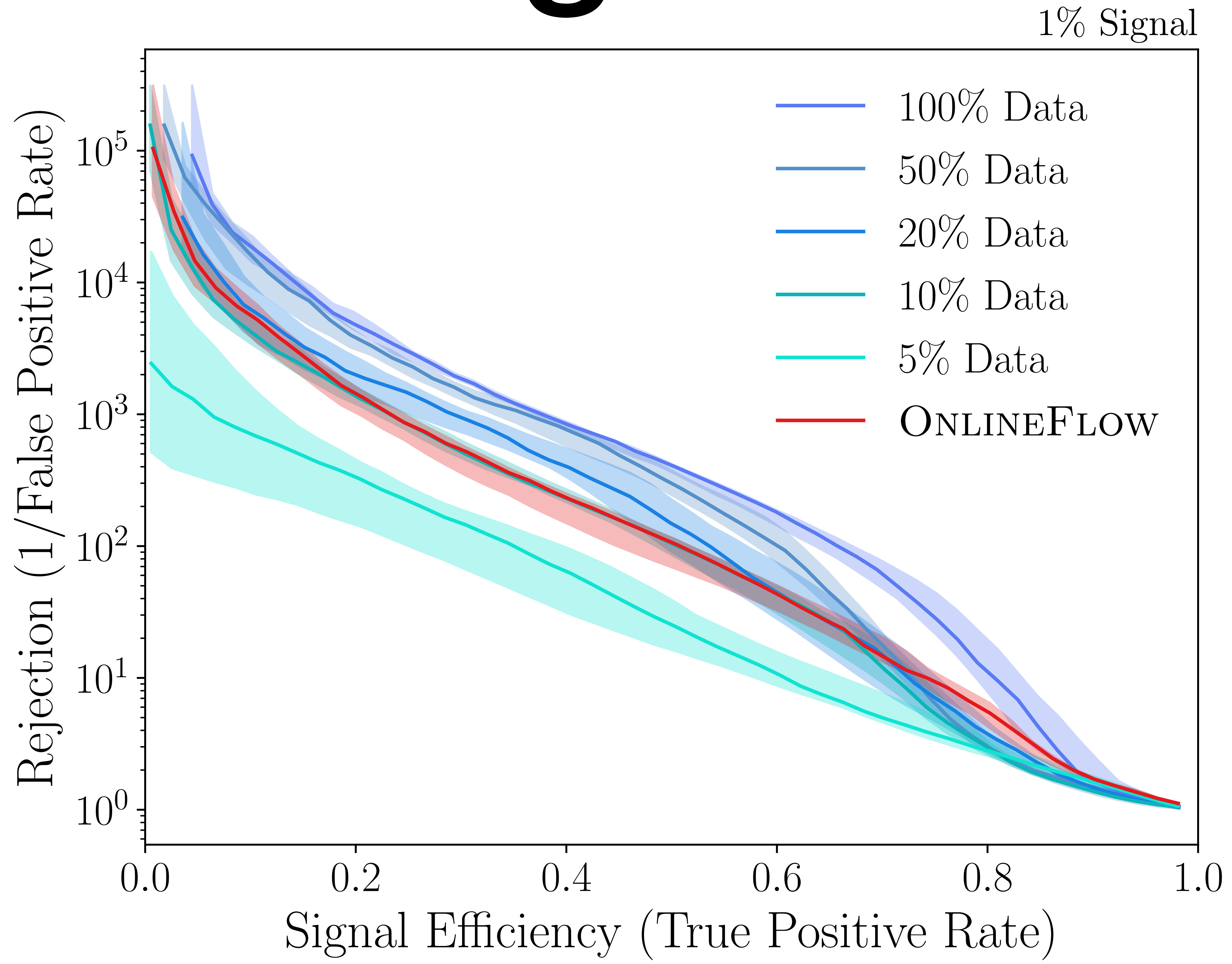


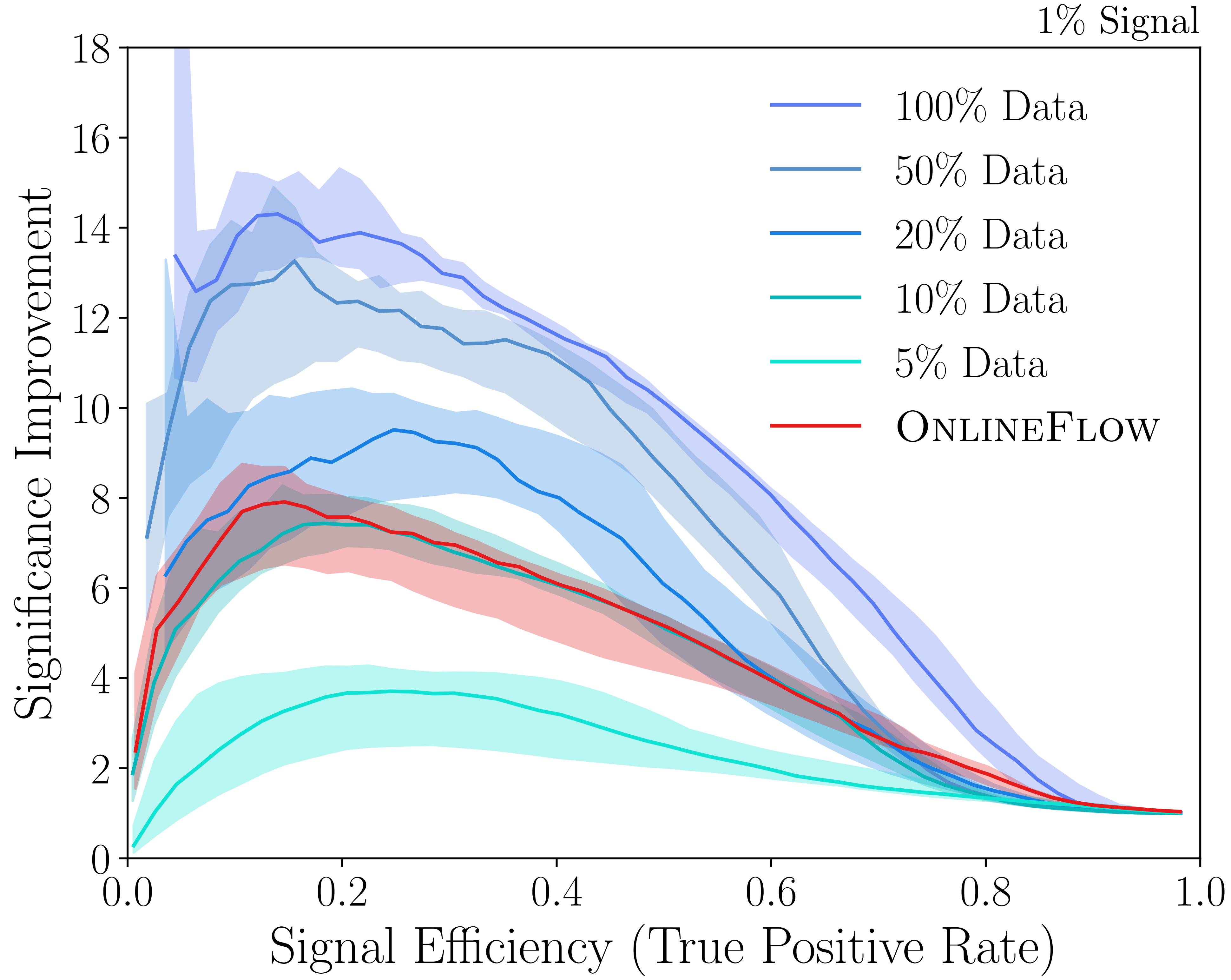


- Train flow on LHCO data
- Run anomaly detection setup on flow data
- Compare to running on training data itself

Kasieczka et al.: **The LHC Olympics 2020: A Community Challenge for Anomaly Detection in High Energy Physics** kl
2101.08320

Higher Dimensional Case

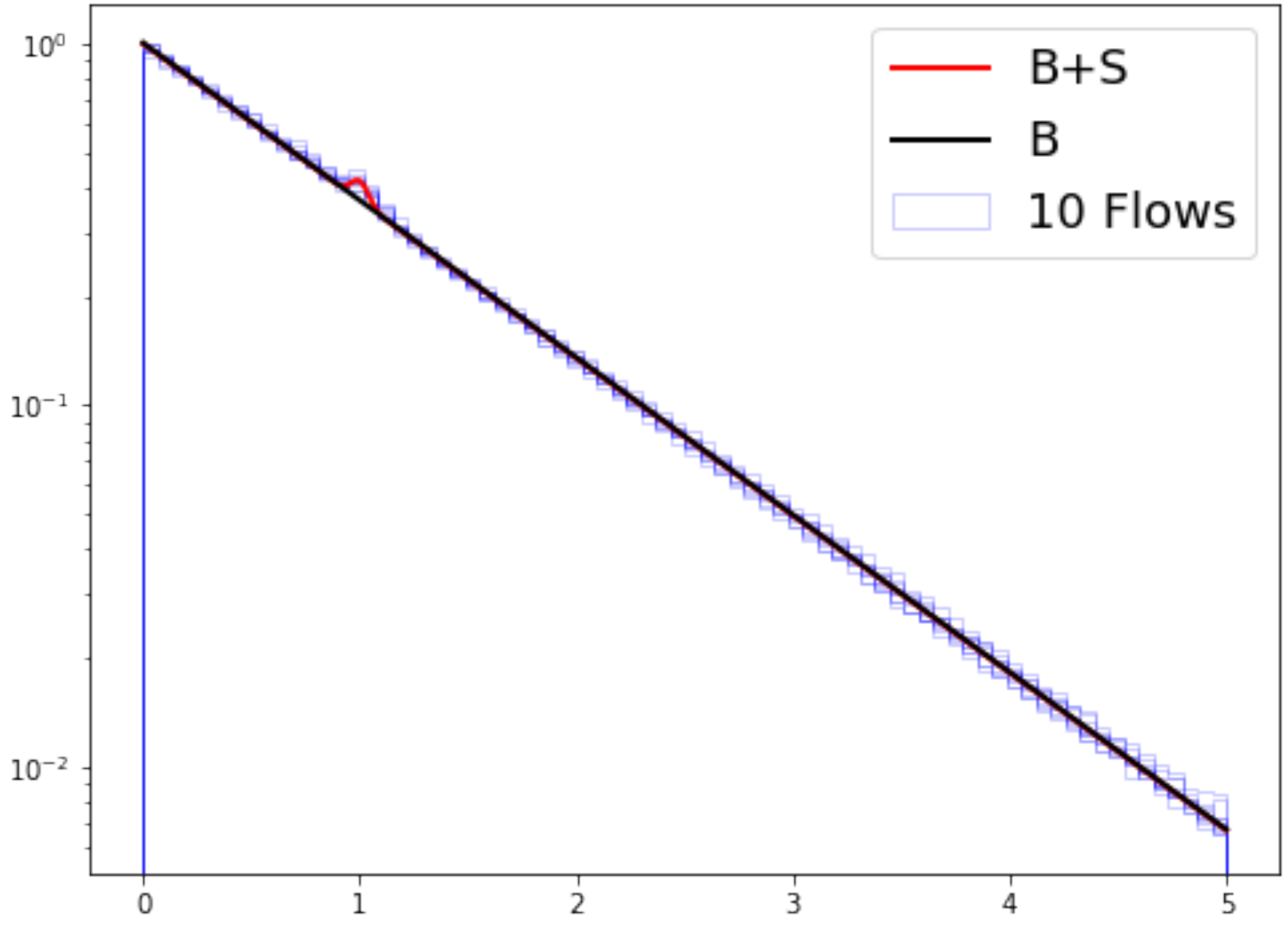




- ROC and SIC curves, larger Rejection/Improvement is better
- Flow data performs about as well as 10% of the training data
- Still gain benefit if perscale factor is 10 or larger

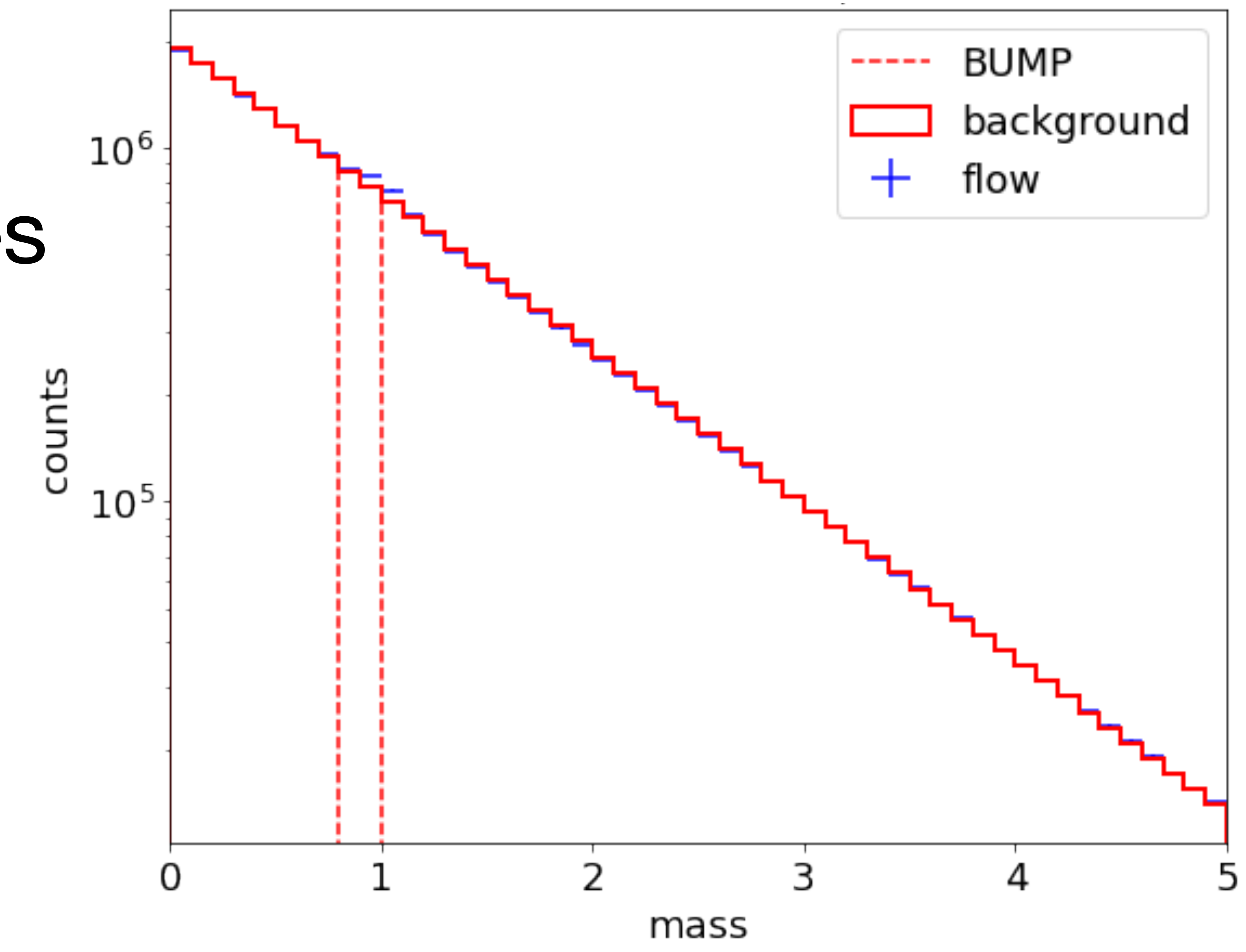
Backup

- Train **ensemble** of Flows
  - Same underlying distribution
  - Independant data points
- Lets us estimate systematic uncertainty
- Allows for easy parallelised training



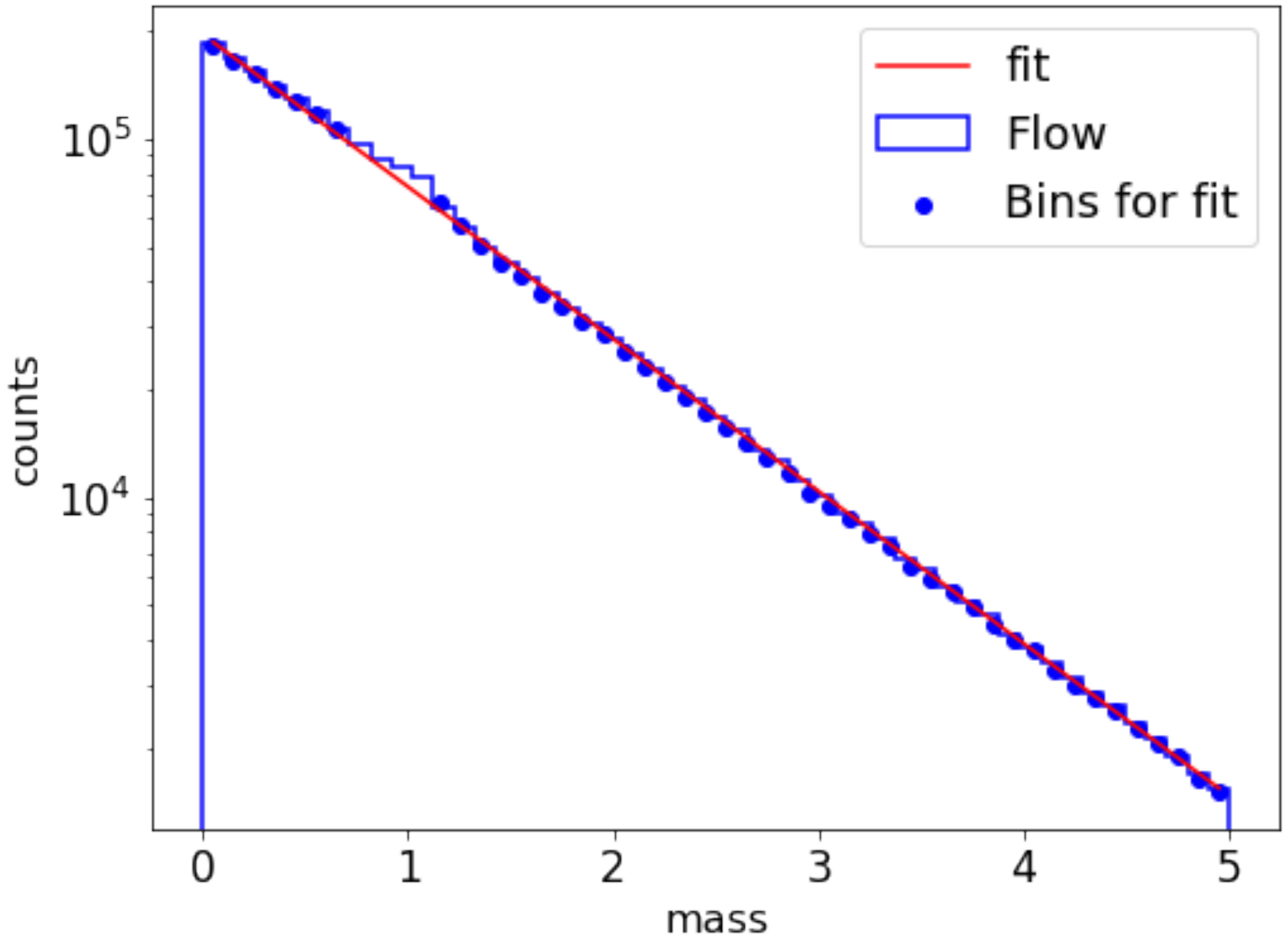
Backup

- Uncertainty estimate:
- Step 1:
  - Look at **combined** flow samples
  - Run bump-hunt
  - Get upper and lower bound for signal region



Backup

- Step 2:
  - Look at each **individual** flow F_i out of N flows in ensemble
  - Generate samples from F_i
  - Fit background model on samples excluding signal region from step 1
  - Get B_i and $(S+B)_i$ in signal region



Backup

- Step 3:
  - Calculate $B = \sum_i^N B_i$ and $\delta B = \mathrm{std}(B_i)\sqrt{N}$
  - Calculate $(S + B)$ and $\delta(S + B)$ equivalently
  - From this get S and δS
  - Significance: $\dfrac{S}{\sqrt{(\delta S)^2 + B}}$

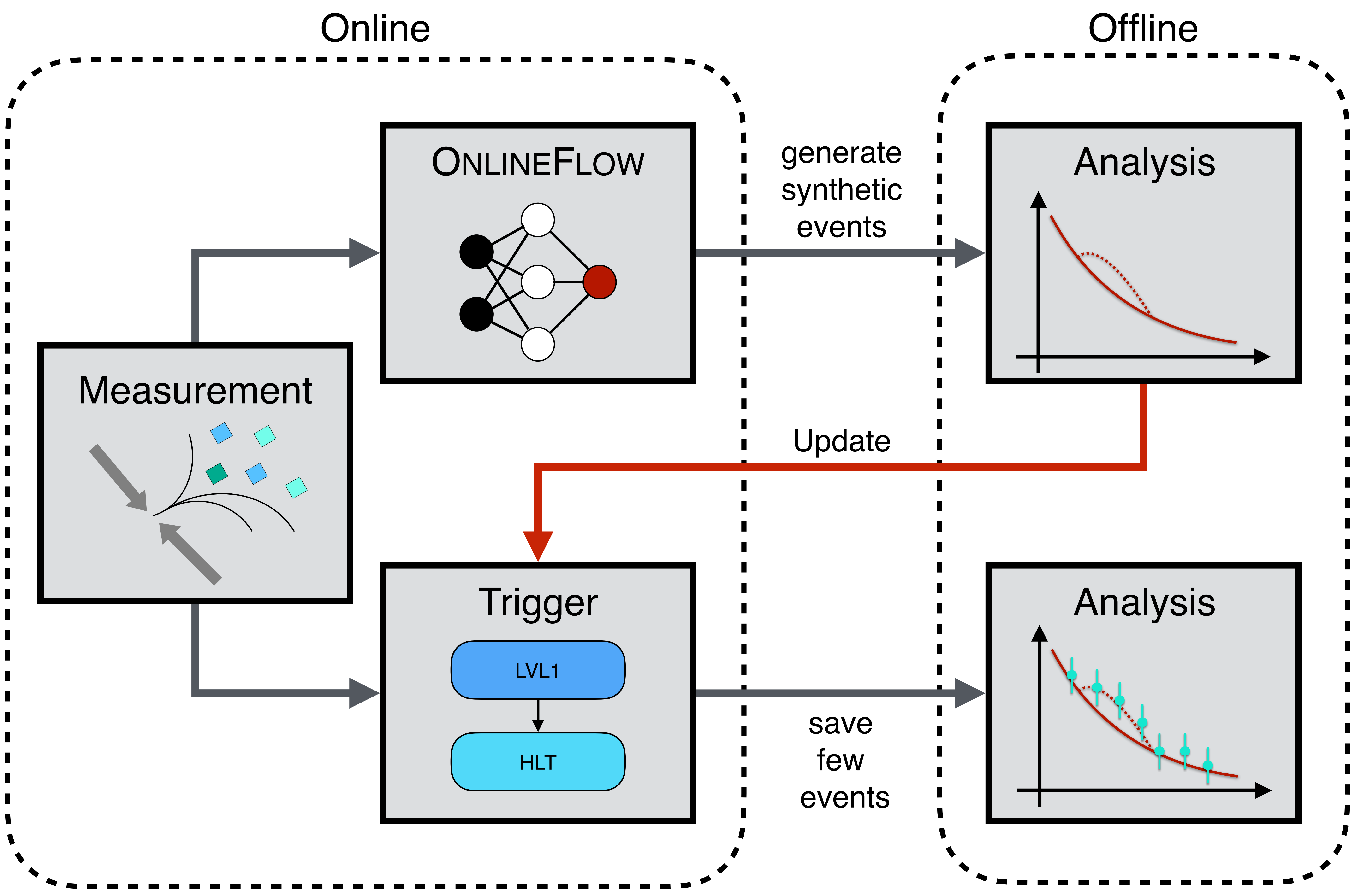
Systematic error via ensemble

Statistical error

Prescale Case

Some event classes:

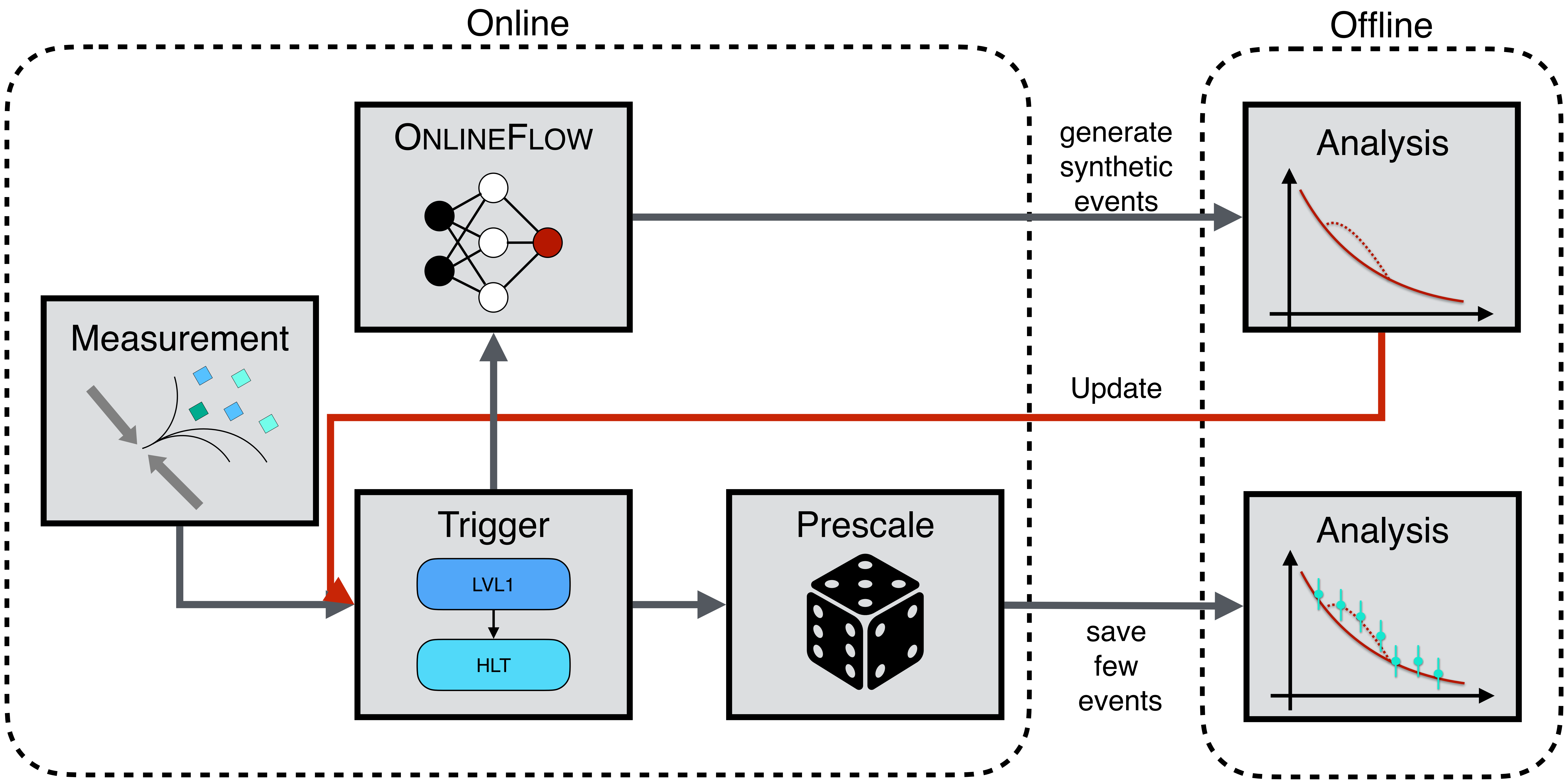
- Too high rates even after trigger



Prescale Case

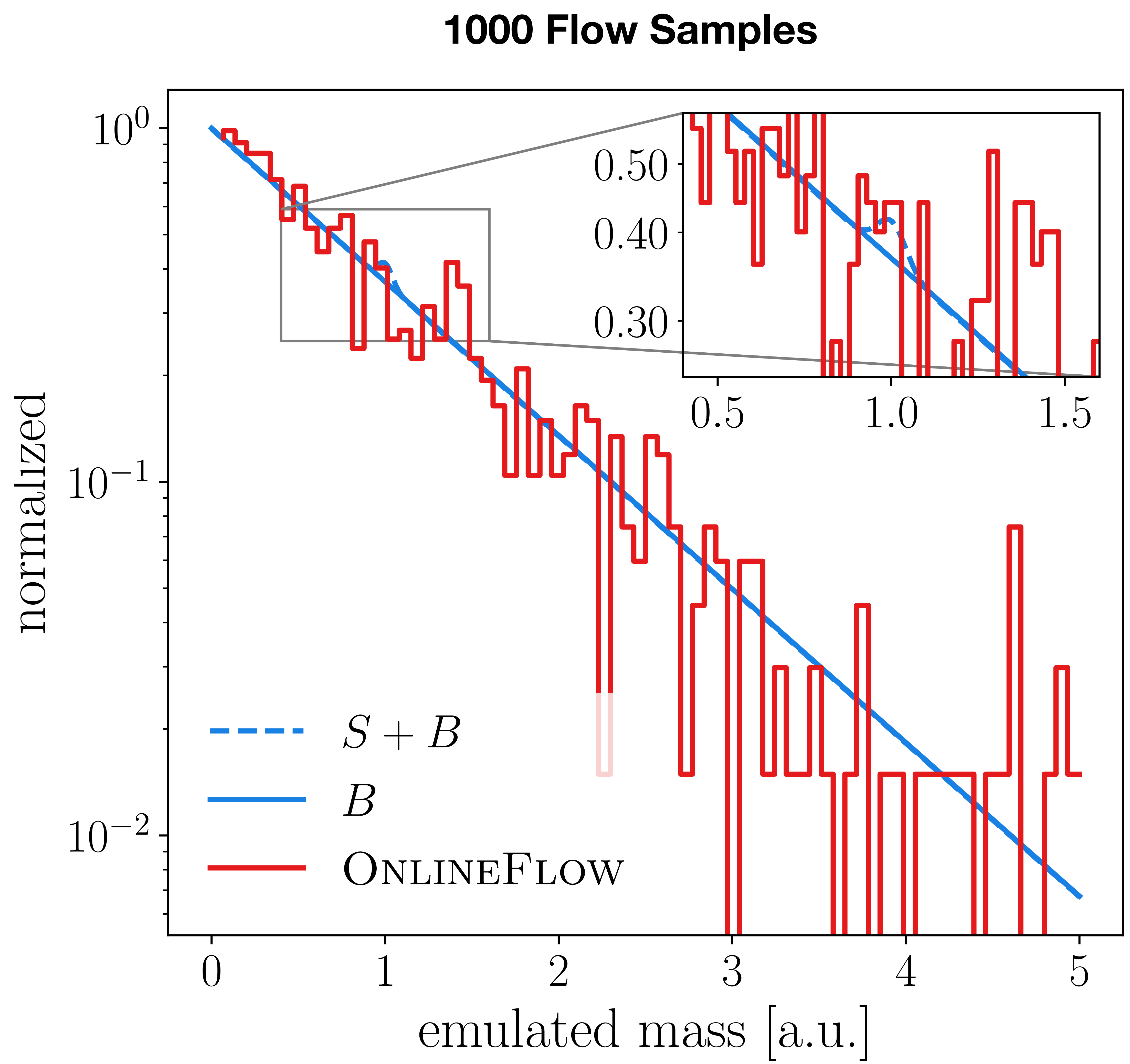
Some event classes:

- Too high rates even after trigger
- Apply percale
- Randomly select events to save
- 1/prescale chance to keep event



Proof of Concept

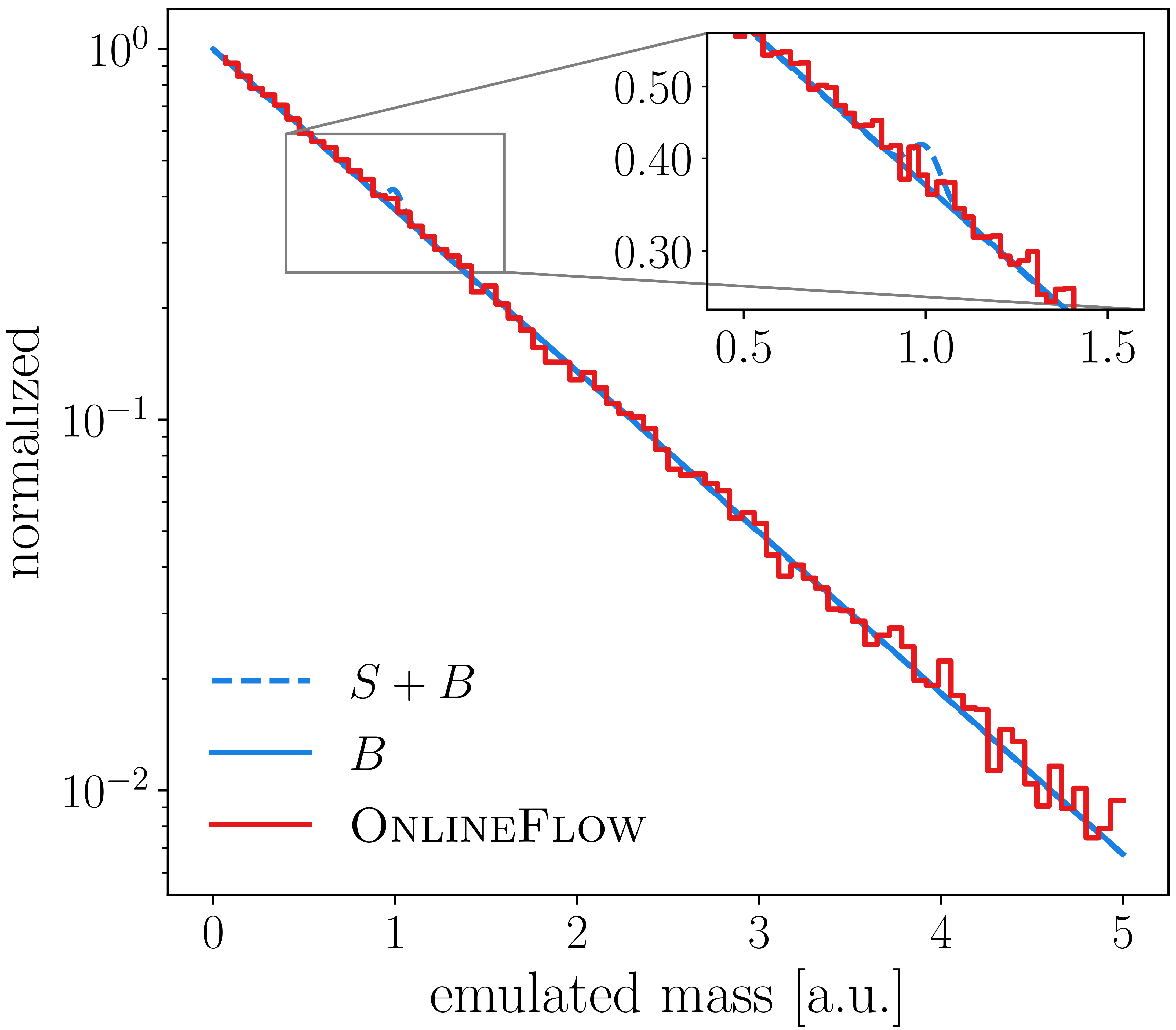
- What is our significance?
  - Classic assumption $\frac{S}{\sqrt{B}}$
  - Generative models break this
    - Fluctuations if few points



Proof of Concept

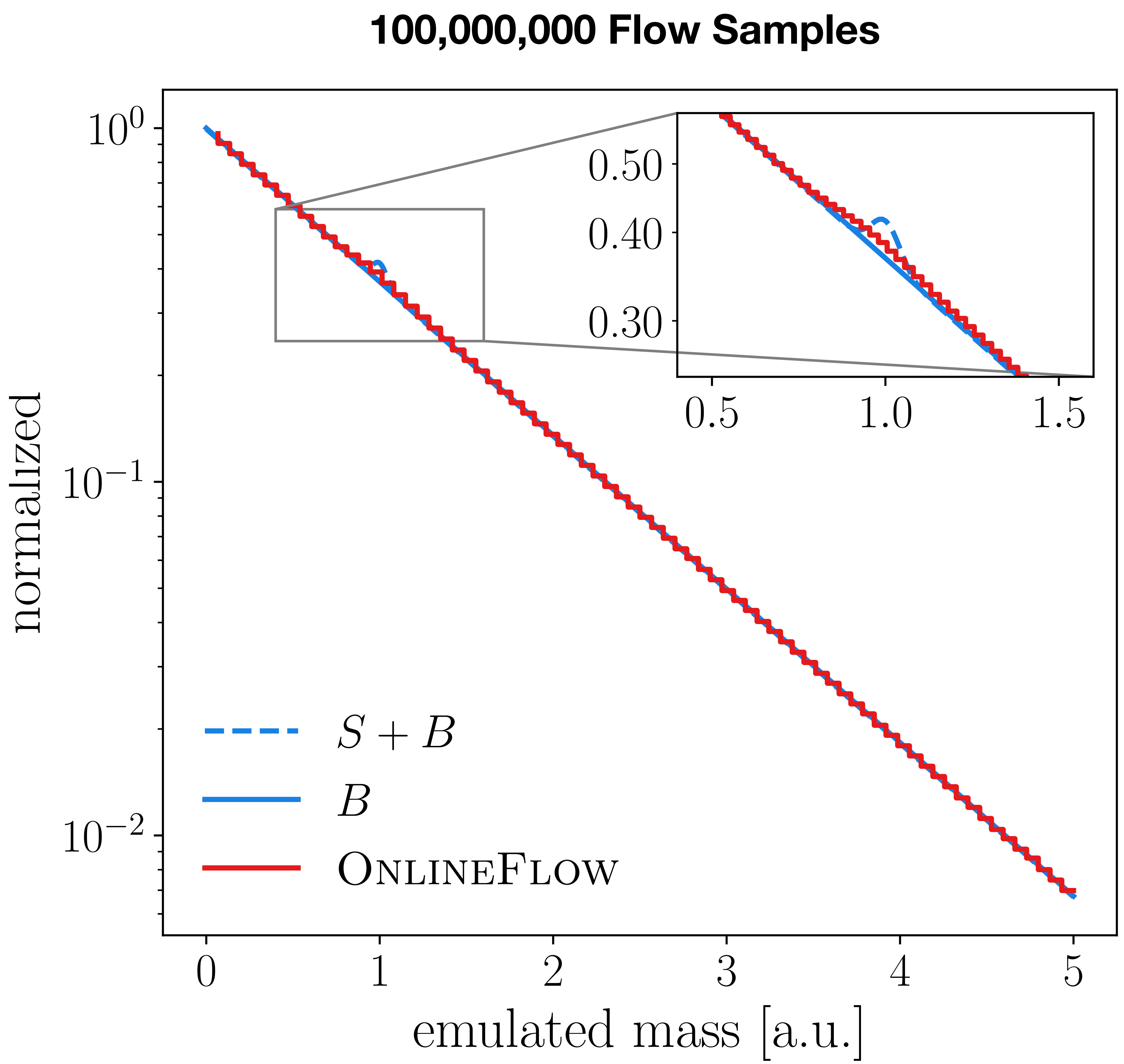
- What is our significance?
  - Classic assumption $\frac{S}{\sqrt{B}}$
  - Generative models break this
    - Fluctuations if few points
    - More points reduce this

100,000 Flow Samples



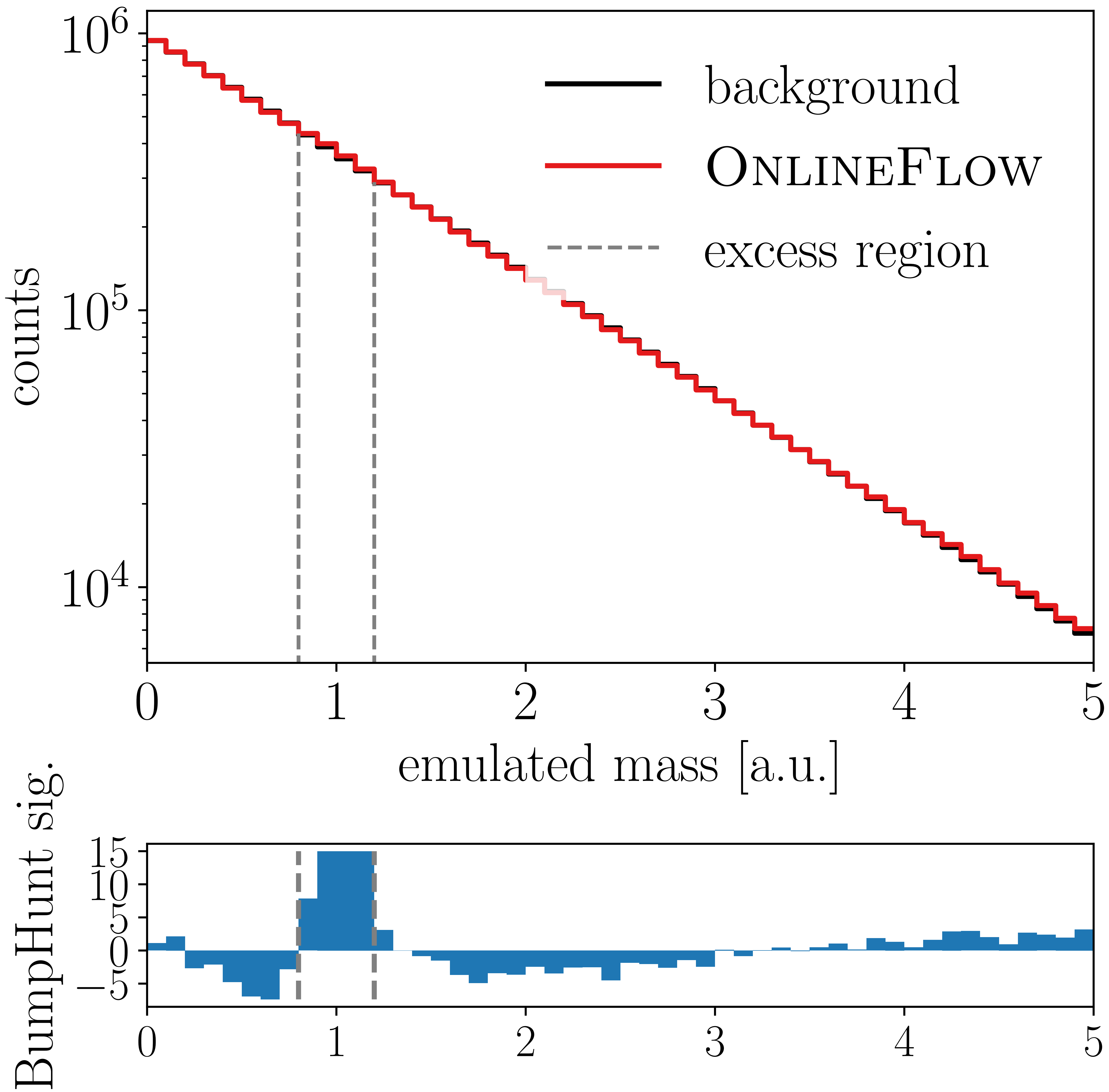
Proof of Concept

- What is our significance?
  - Classic assumption $\frac{S}{\sqrt{B}}$
  - Generative models break this
    - Fluctuations if few points
    - More points reduce this



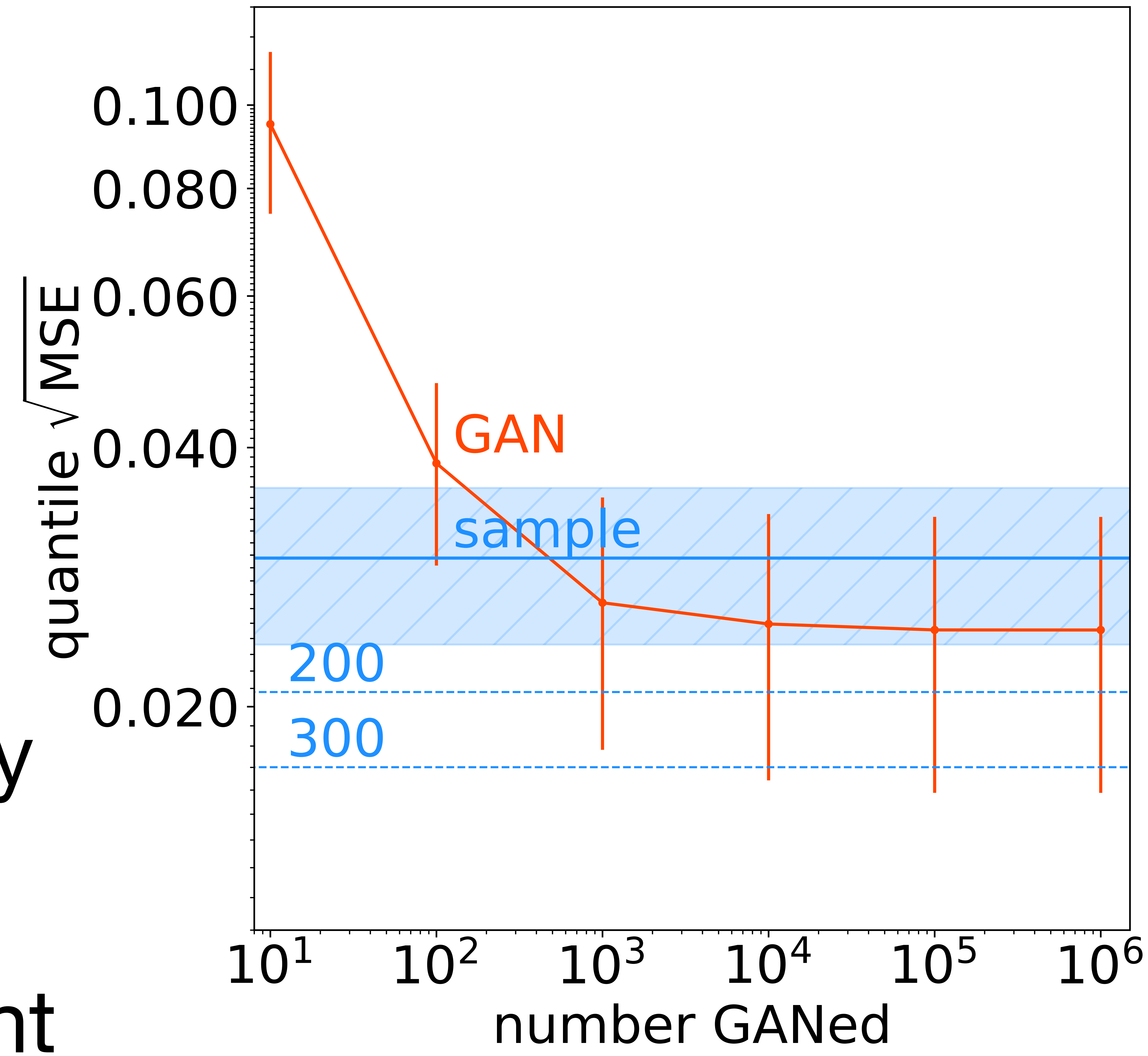
Proof of Concept

- What is our significance?
  - Classic assumption $\frac{S}{\sqrt{B}}$
  - Generative models break this



Proof of Concept

- What is our significance?
  - Classic assumption $\frac{S}{\sqrt{B}}$
  - Generative models break this
    - Fluctuations if few points
    - More points reduce this
    - More samples $\neq$ less uncertainty
    - No Poisson uncertainty
    - Systematic uncertainty dominant



Butter et al.: **Amplifying Statistics using Generative Models:** NeurIPS ML4PS 2020, 2008.06545