Multitask, Every Region, All at Once :

Fine-Tunable Region-Conditioned SBI for Collider Analysis

Yong Sheng Koay[a] , Thorsten Glüsenkamp[b], Stefano Moretti[a,c] , Prashant Singh[d] , Harri Waltari[a]

a Department of Physics & Astronomy, Uppsala University, Box 516, 75120 Uppsala, Sweden
b Department of Physics, Stockholm University, SE-10691 Stockholm, Sweden
c School of Physics & Astronomy, University of Southampton, Southampton SO17 1BJ, UK
d Department of Information Technology, Uppsala University, Box 516, 75120 Uppsala, Sweden

UPPSALA UNIVERSITET

Collider Analysis Workflow with ML (or an over-simplification of it)

Collider Analysis Workflow with ML

I. Choose a BSM model → New Particles → New Signals

Collider Analysis Workflow with ML

I. Fixed a BSM model

II. Choose the Signal Process → Relevant BSM parameters

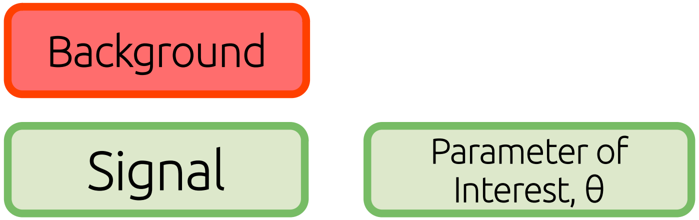
Signal

Parameter of Interest, θ

Collider Analysis Workflow with ML

I. Fixed a BSM model

II. Fixed the Signal Process → Relevant BSM parameters → Fixed the Final state → Fixed the SM background process

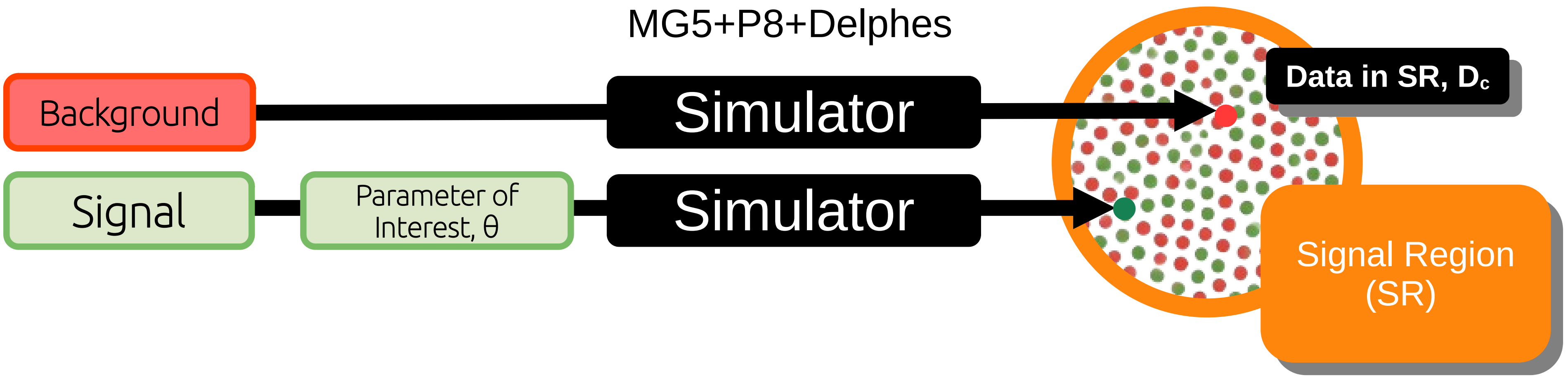


Collider Analysis Workflow with ML

I. Fixed a BSM model
II. Fixed the processes
III. Build the relevant Signal Regions(SR) → Based on physics + human intuition / ML

Collider Analysis Workflow with ML

I. Fixed a BSM model
II. Fixed the processes
III. Fixed the Signal Regions (SR)
→ Populate it with simulated events



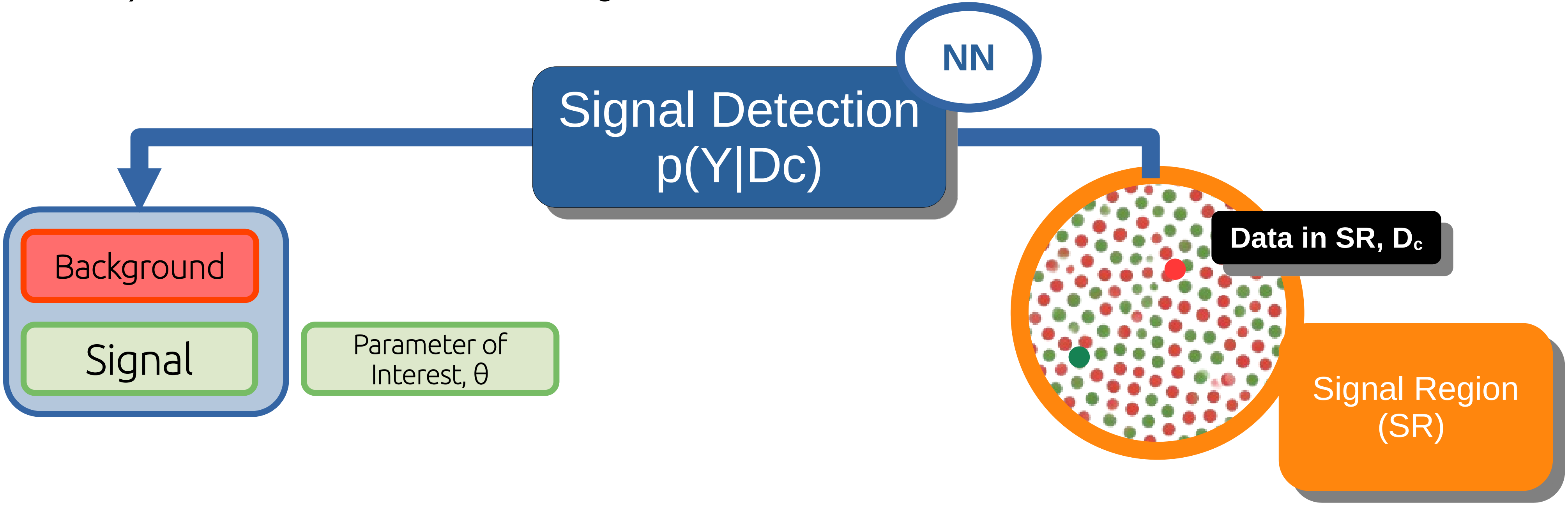
Collider Analysis Workflow with ML

I. Fixed a BSM model

II. Fixed the processes

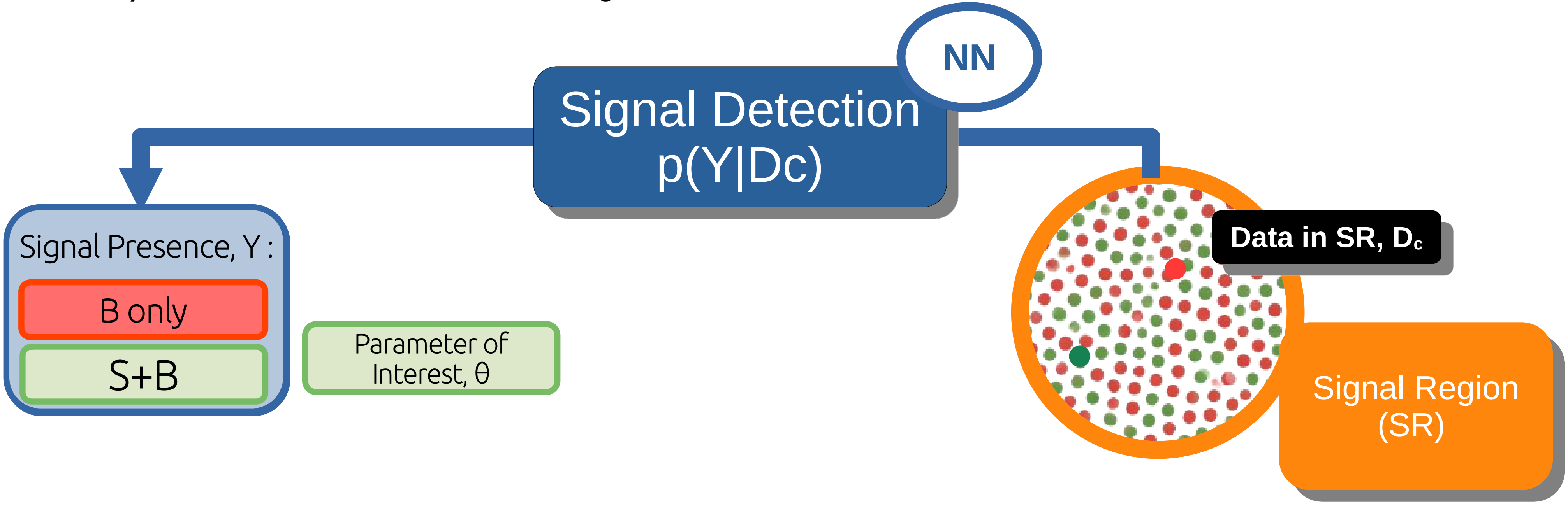
III. Fixed the Signal Regions (SR)

  a) Train NN that detects the signal



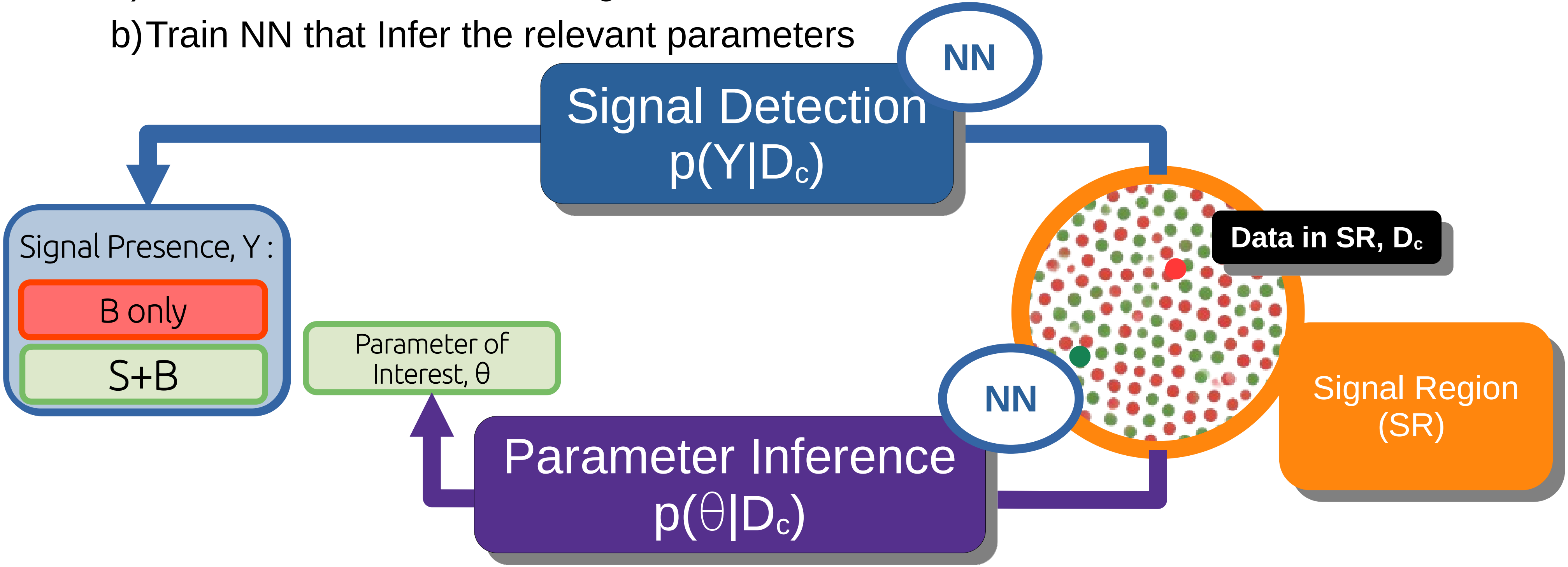
Collider Analysis Workflow with ML

I. Fixed a BSM model
II. Fixed the processes
III. Fixed the Signal Regions (SR)
   a) Train NN that detects the signal



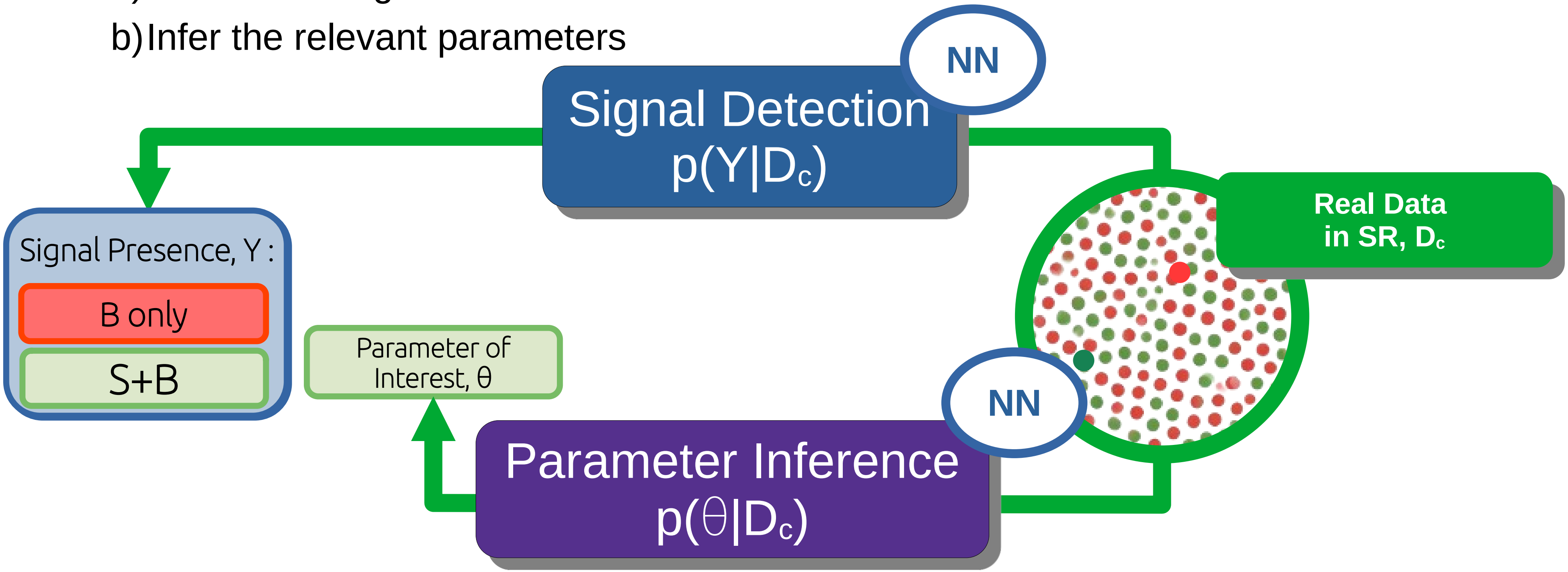
Collider Analysis Workflow with ML

I. Fixed a BSM model
II. Fixed the processes
III. Fixed the Signal Regions (SR)
   a) Train NN that detects the signal
   b) Train NN that Infer the relevant parameters



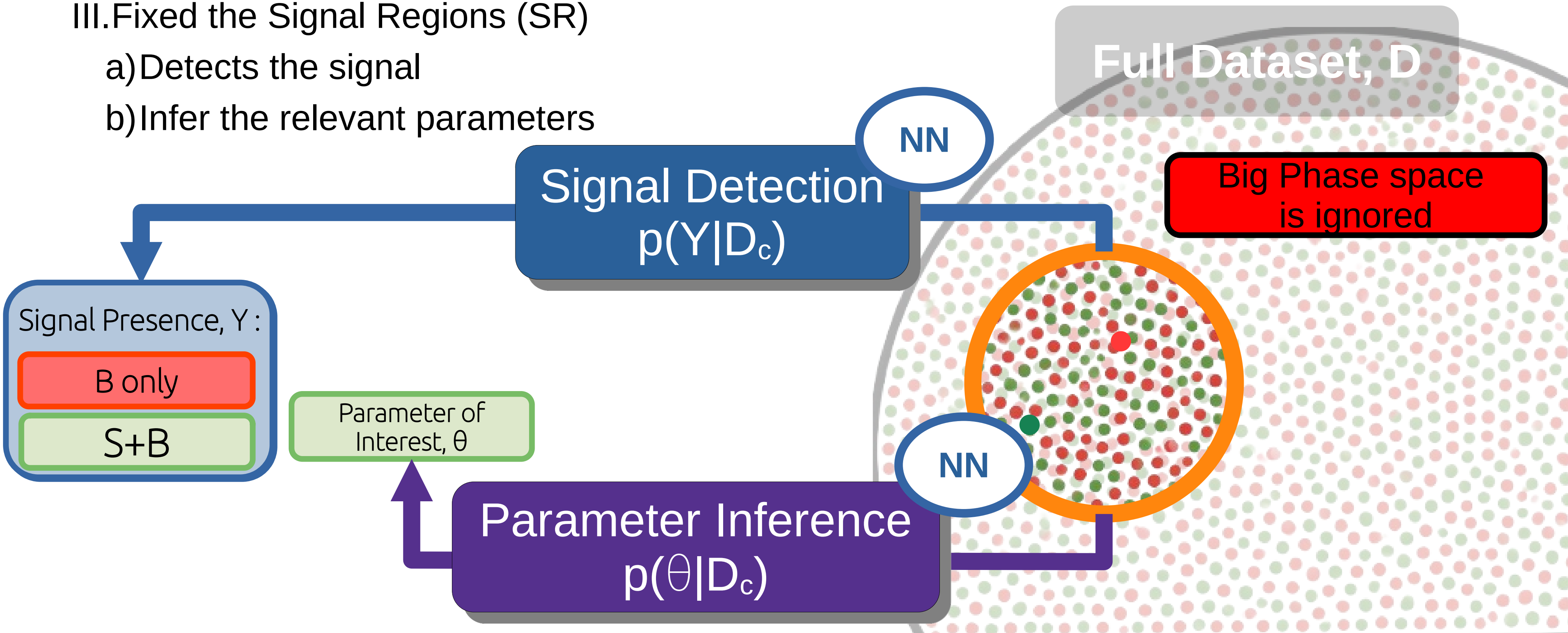
Collider Analysis Workflow with ML

I. Fixed a BSM model
II. Fixed the processes
III. Fixed the Signal Regions (SR)
   a) Detects the signal
   b) Infer the relevant parameters



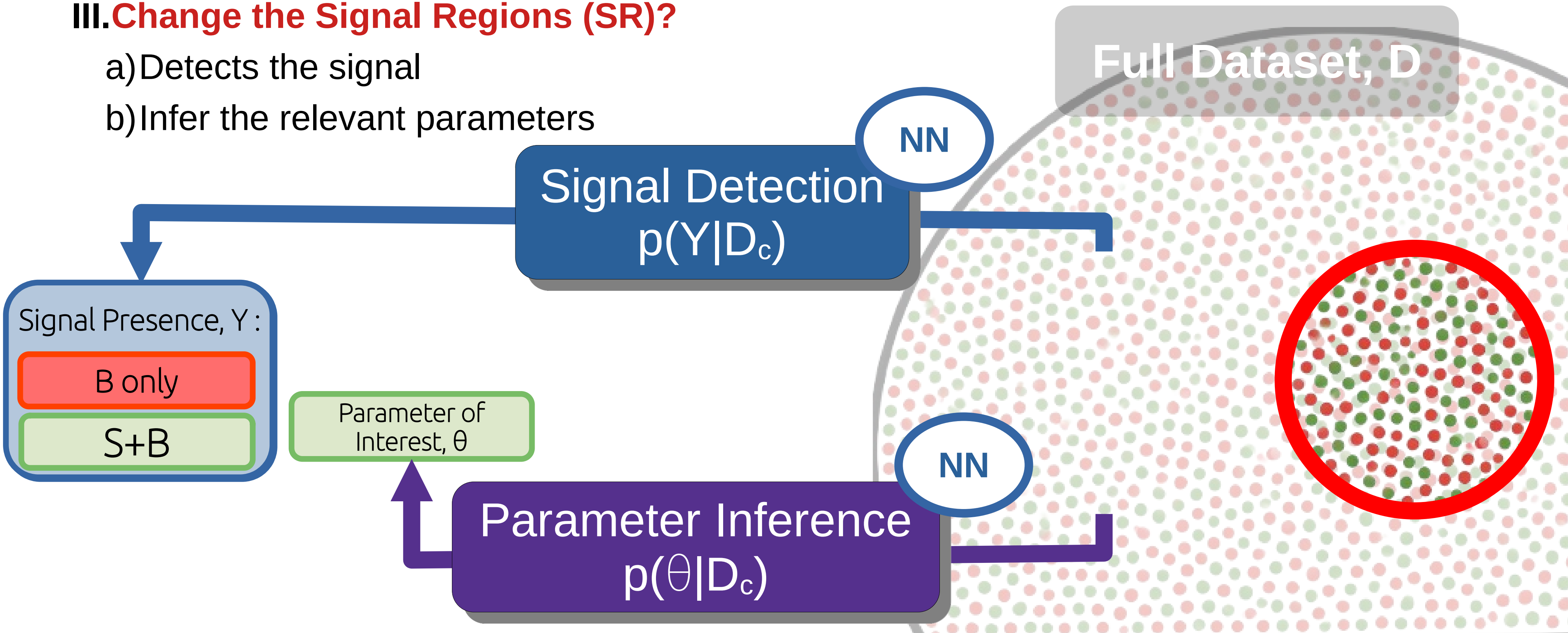
Collider Analysis Workflow with ML

I. Fixed a BSM model
II. Fixed the processes
III. Fixed the Signal Regions (SR)
   a) Detects the signal
   b) Infer the relevant parameters



Collider Analysis Workflow with ML

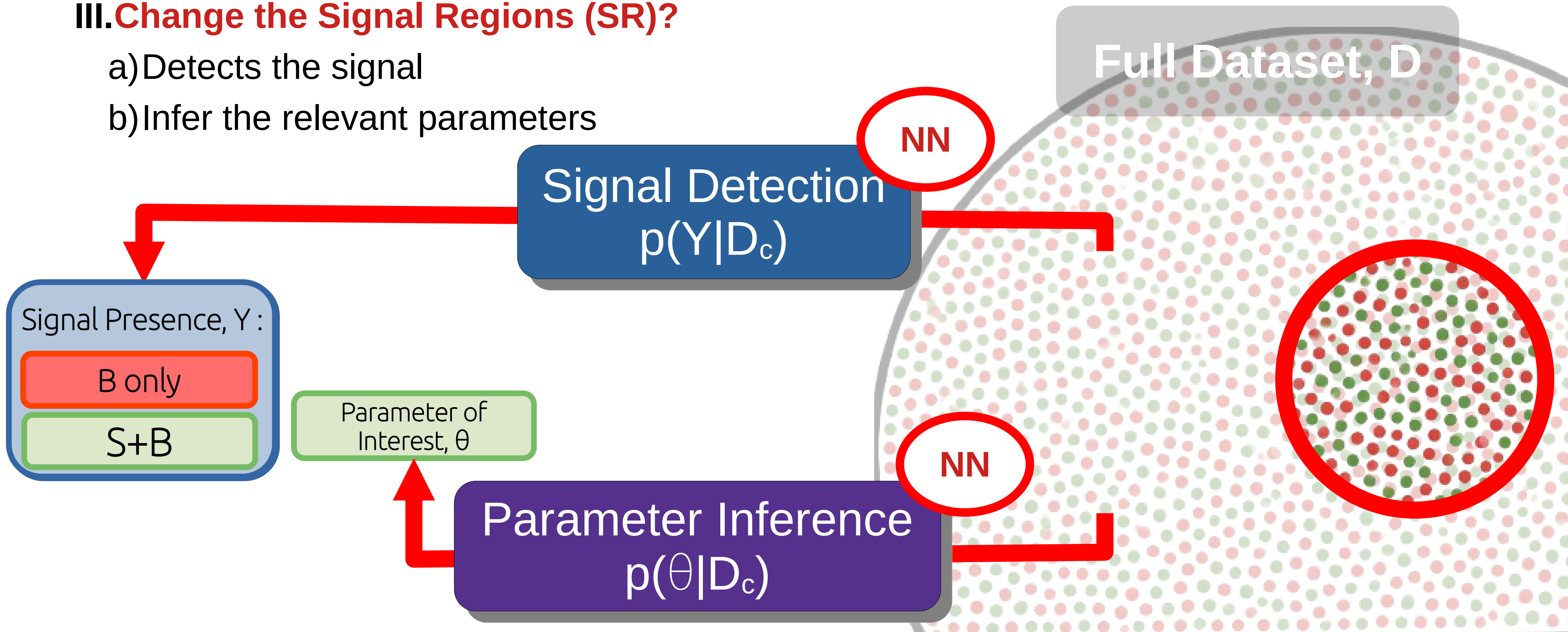
I. Fixed a BSM model
II. Fixed the processes
III. Change the Signal Regions (SR)?
  a) Detects the signal
  b) Infer the relevant parameters



Collider Analysis Workflow with

In some cases:
Networks trained on fixed signal regions
Changing means retrain from scratch!

I. Fixed a BSM model
II. Fixed the processes
III. Change the Signal Regions (SR)?
   a) Detects the signal
   b) Infer the relevant parameters

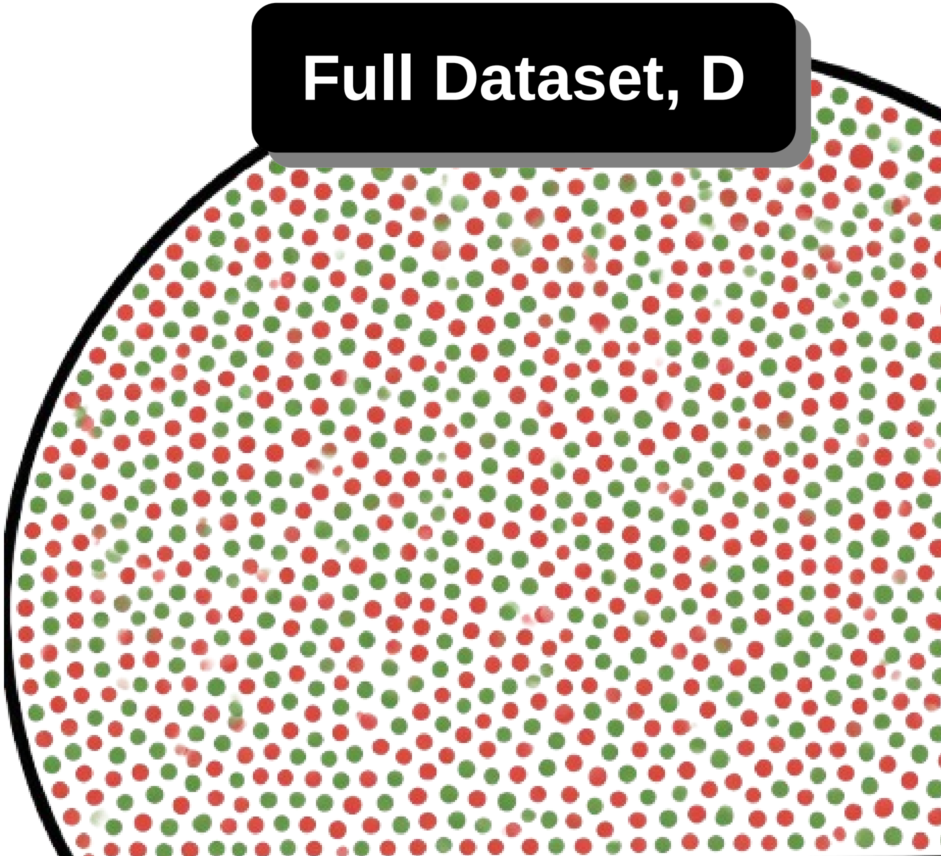


Collider Analysis Workflow with ML

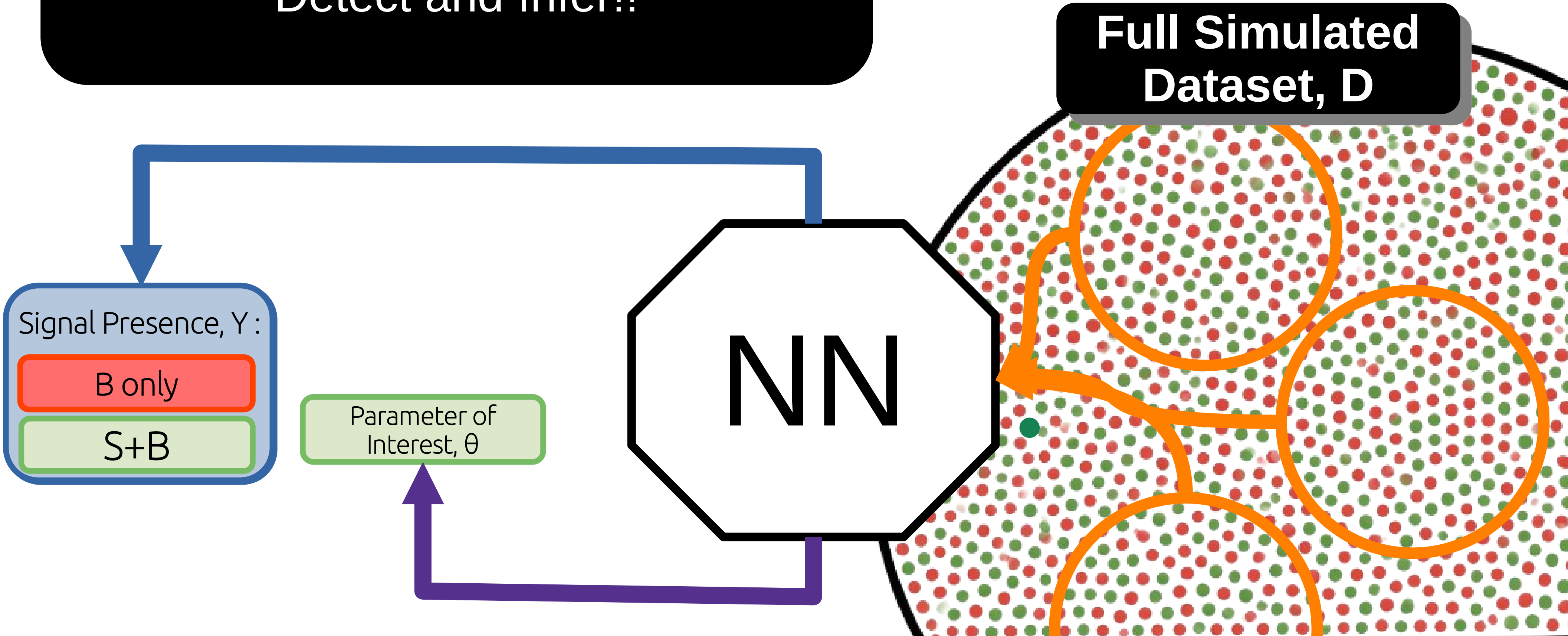
Why are we focusing on
only One or few SRs
when training our models?

Why are we training
different models
for different tasks?

All the physics and statistics are in the same dataset!

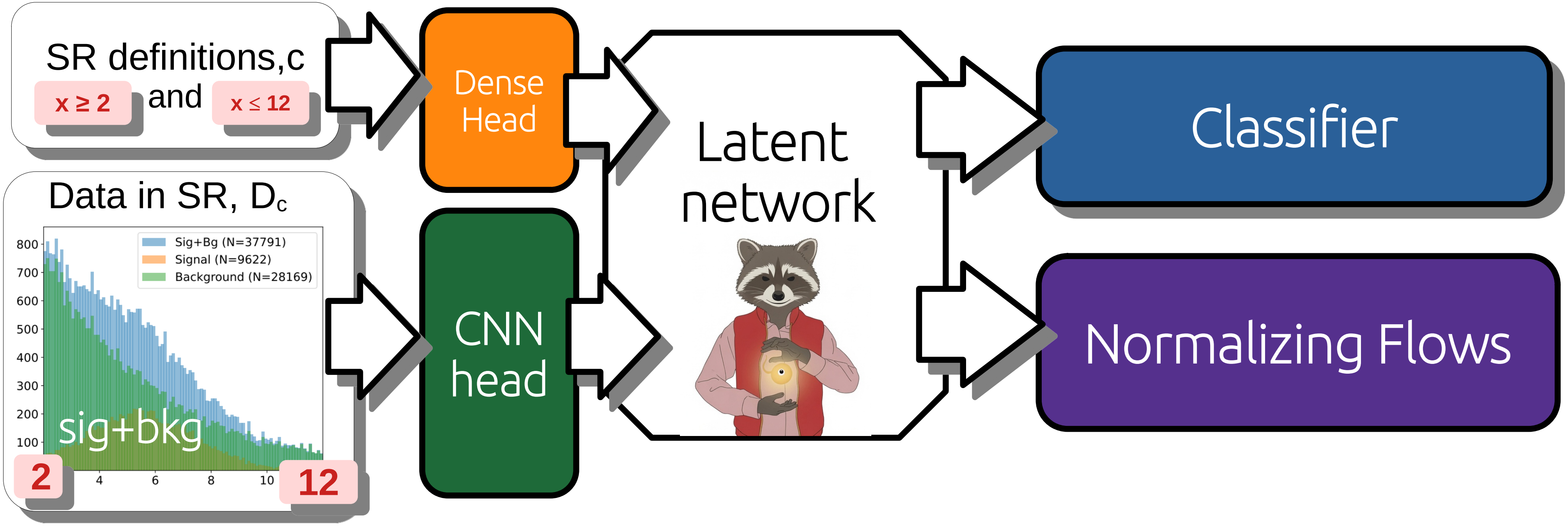


Goal:
Amortize over Signal Regions
Detect and Infer!!



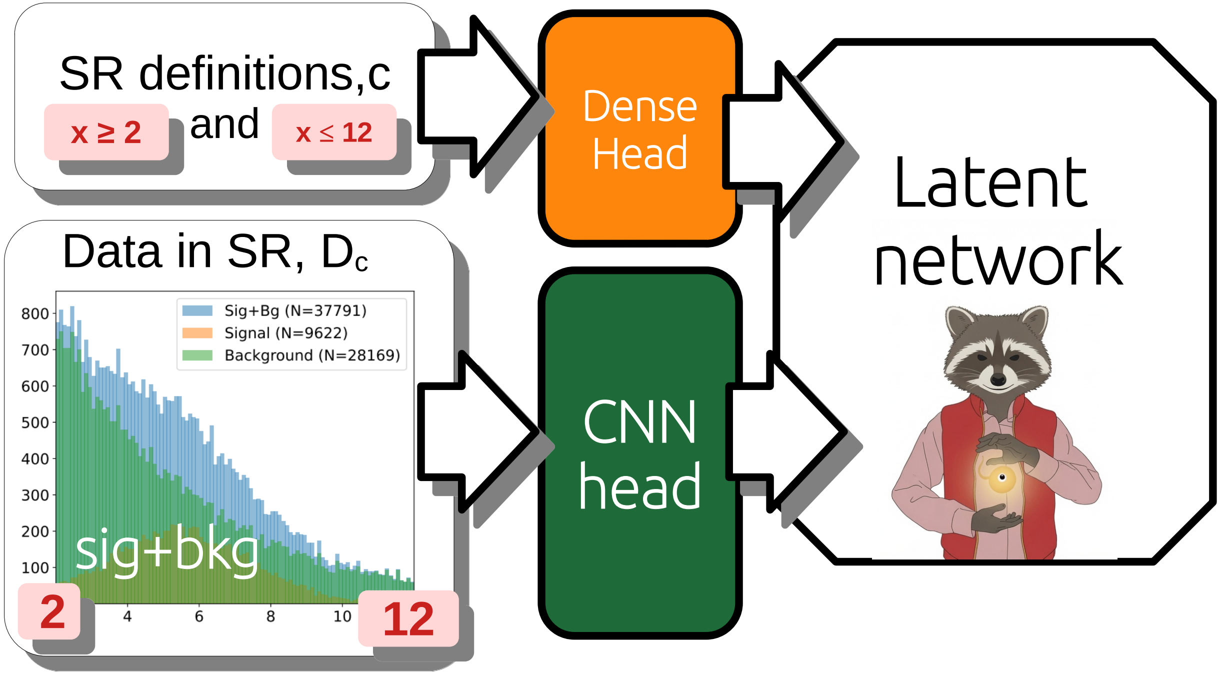
RACNN-Classiflows

Region Aware Convolutional Neural Network with Classifier and Flows



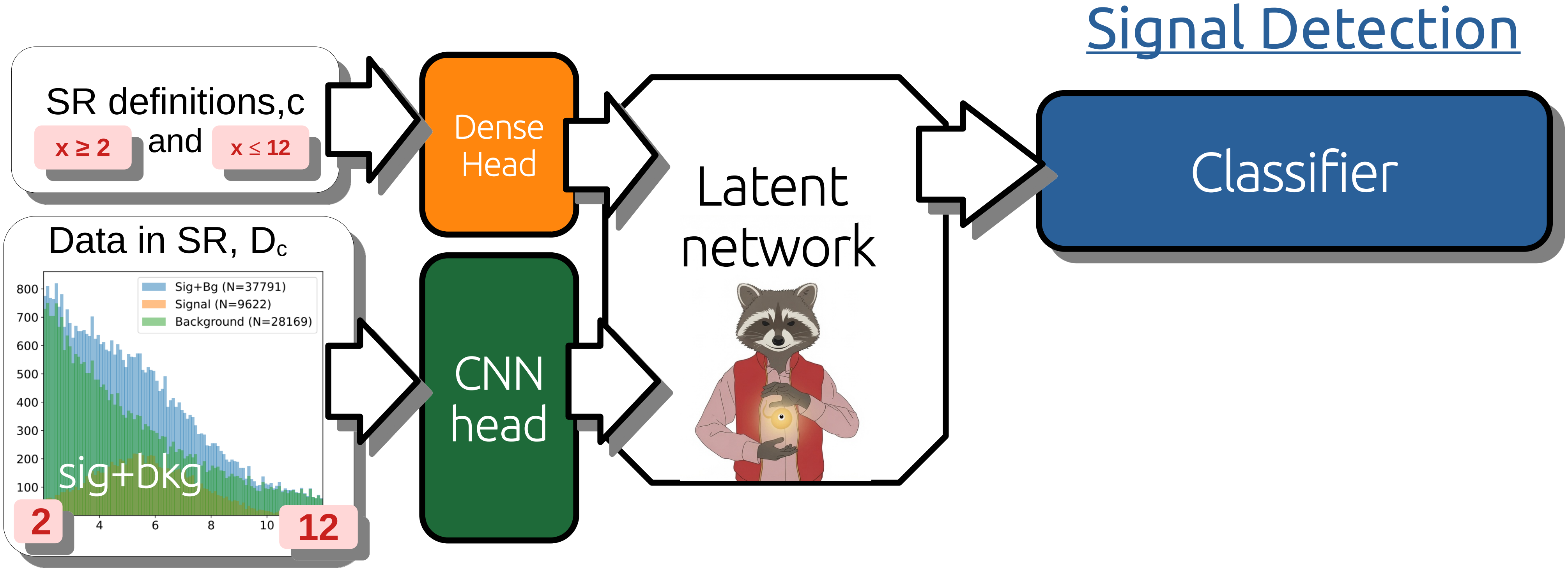
RACNN-Classiflows

Region Aware Convolutional Neural Network with Classifier and Flows



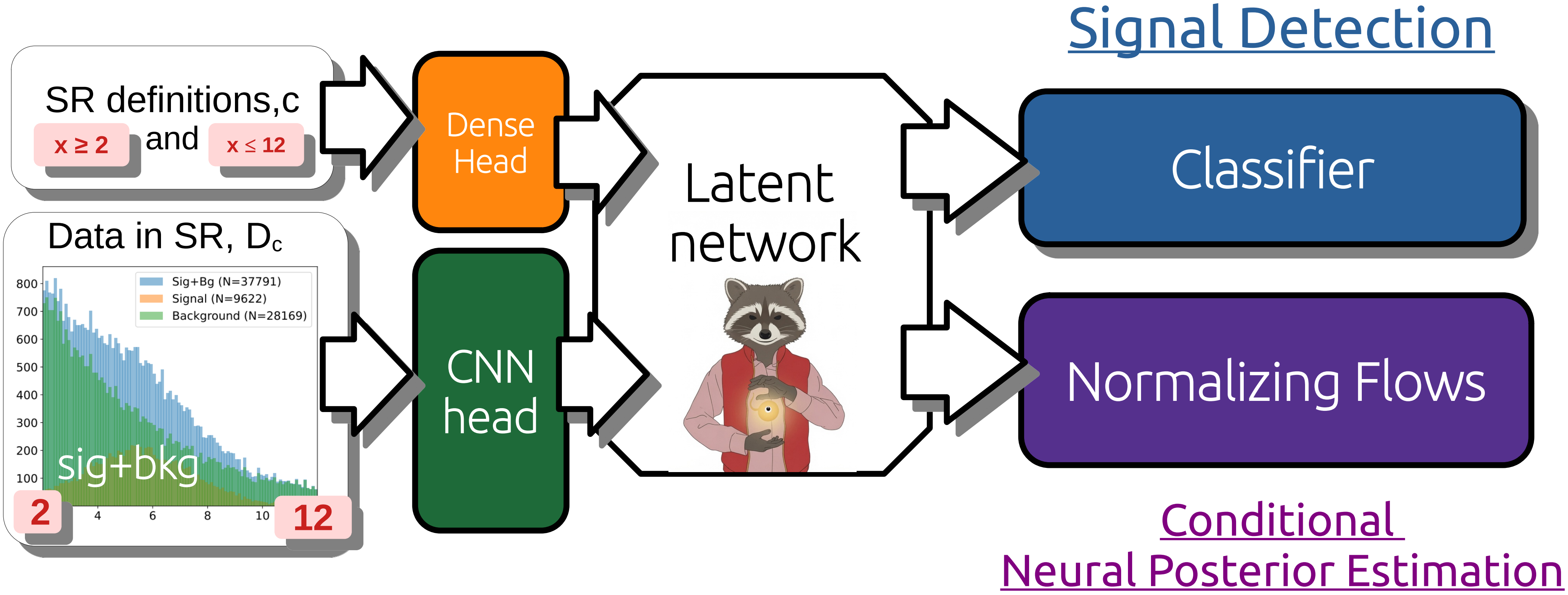
RACNN-Classiflows

Region Aware Convolutional Neural Network with Classifier and Flows



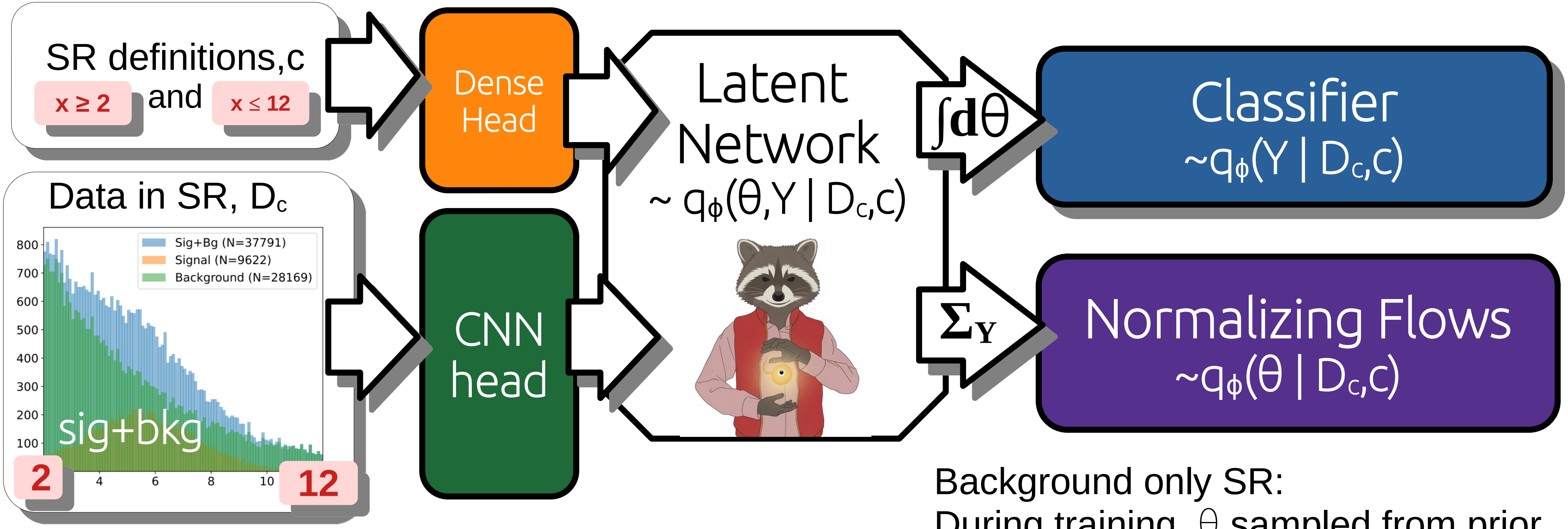
RACNN-Classiflows

Region Aware Convolutional Neural Network with Classifier and Flows



RACNN-Classiflows

Region Aware Convolutional Neural Network with Classifier and Flows

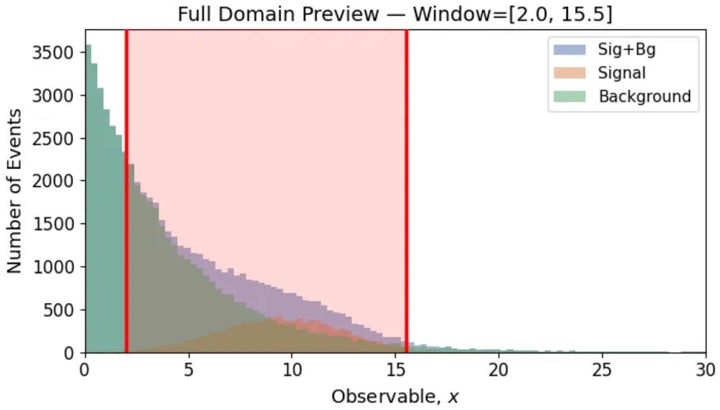


Background only SR:
During training, θ sampled from prior

Loss = NLL + BCE

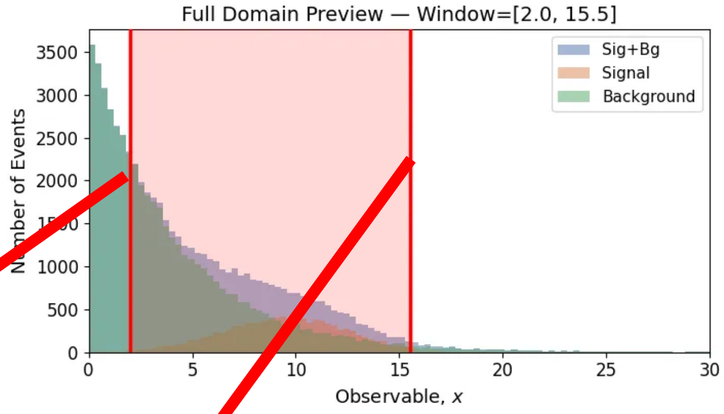
Toy Example:

- **Find position and width of bump**
- **Train:**
  - **~450 θ**
  - **~150 SR per θ**

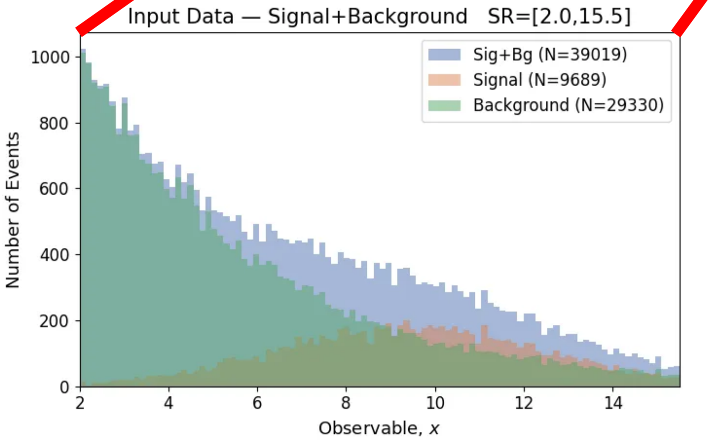


Toy Example:

- **Find position and width of bump**
- **Train:**
  - **~450 θ**
  - **~150 SR per θ**



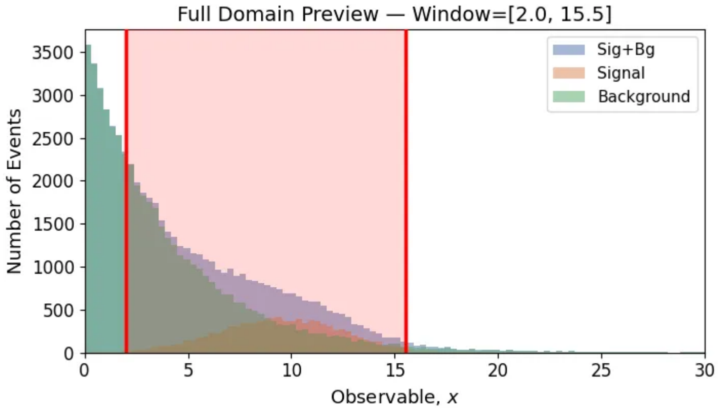
X ≥ 2 , X ≤ 15.5



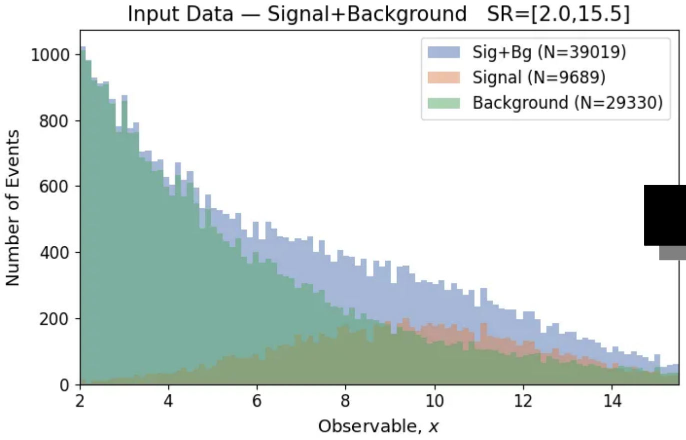
Significance = 54.1

Toy Example:

- **Find position and width of bump**
- **Train:**
  - **~450 θ**
  - **~150 SR per θ**



$X \geq 2$, $X \leq 15.5$

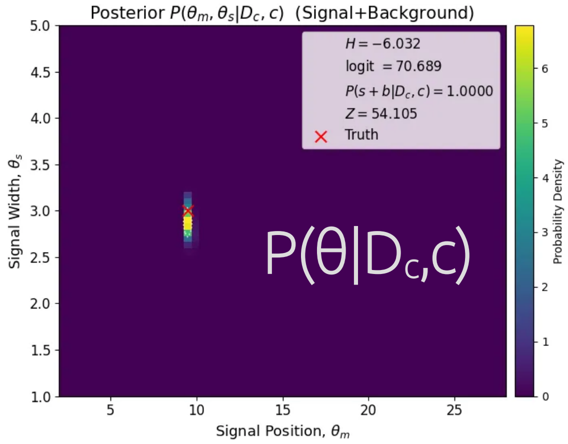


Significance = 54.1

RACNN-Classiflows

$P(s+b|D_c,c) = 1.0000$

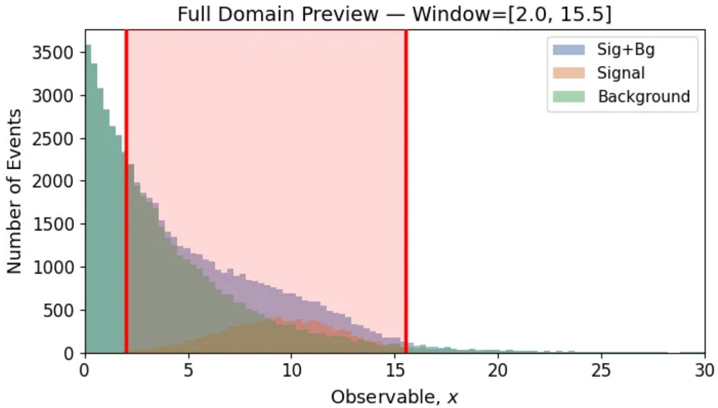
Logits = 70.689



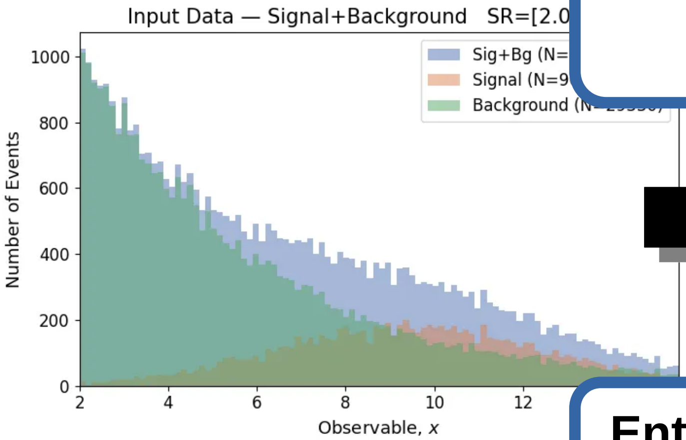
Entropy, H[q] = -6.032

Toy Example:

- **Find position and width of bump**
- **Train:**
  - **~450 θ**
  - **~150 SR per θ**

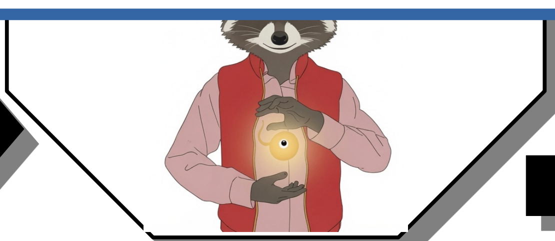


X ≥ 2 , X ≤ 15.5



Significance = 54.1

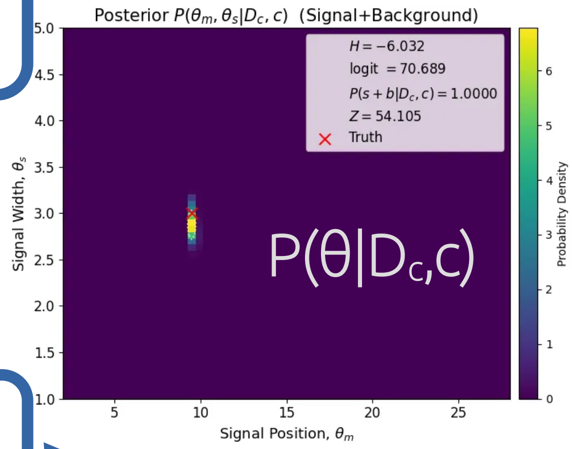
RACNN-



Logits ~ tells you how likely the signal is present

$P(s+b|D_c,c) = 1.0000$

Logits = 70.689



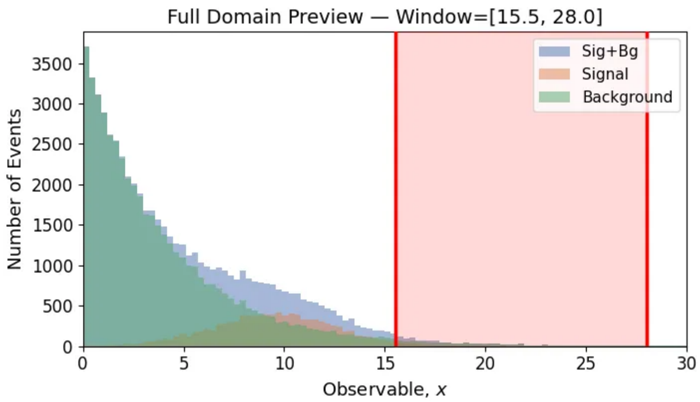
$P(\theta|D_c,c)$

Entropy ~ Uncertainty of the model on the parameters

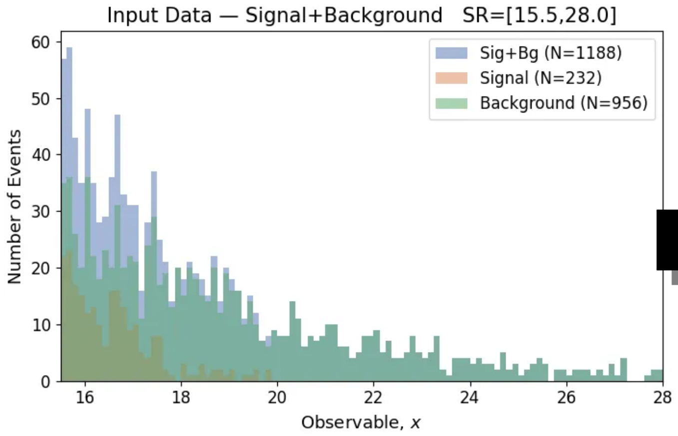
Entropy, H[q] = -6.032

Example Runs:

Different SR,
same θ



$X \geq 15.5$, $X \leq 28$

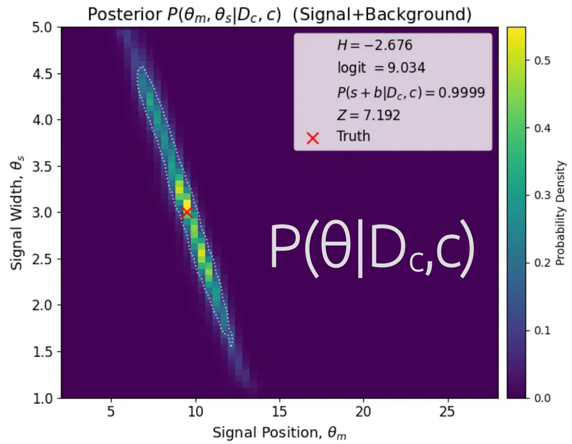


Significance = 7.192

RACNN-
Classiflows

$P(s+b|D_c,c) = 0.9999$

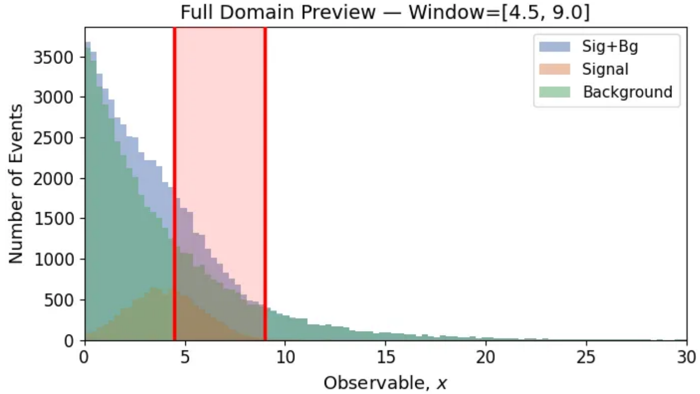
Logits = 9.034



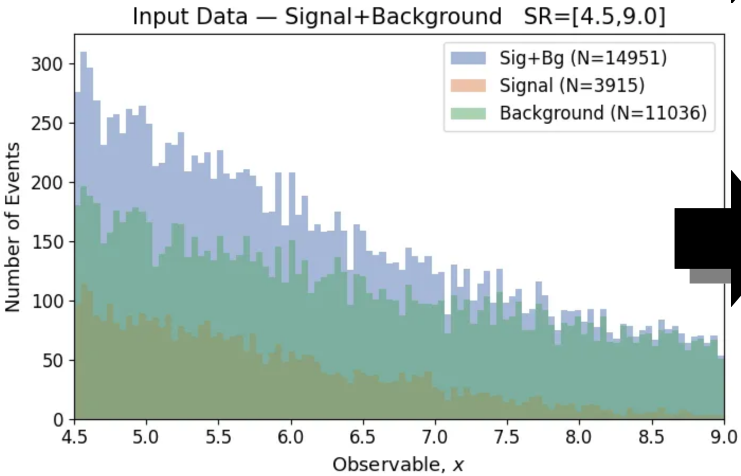
Entropy, H[q] = -2.676

Example Runs:

Different θ,
Different SR



$X \geq 14.5$, $X \leq 28$

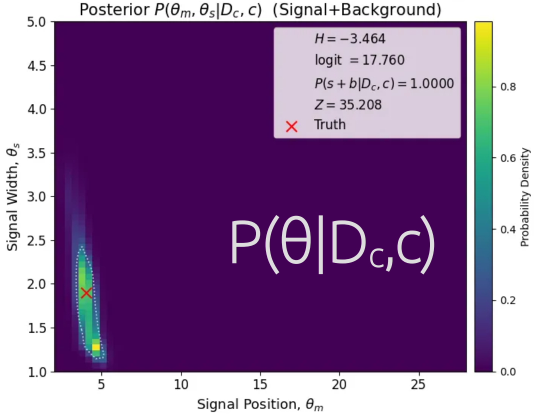


Significance = 35.2

RACNN-Classiflows

$P(s+b|D_c,c) = 1.000$

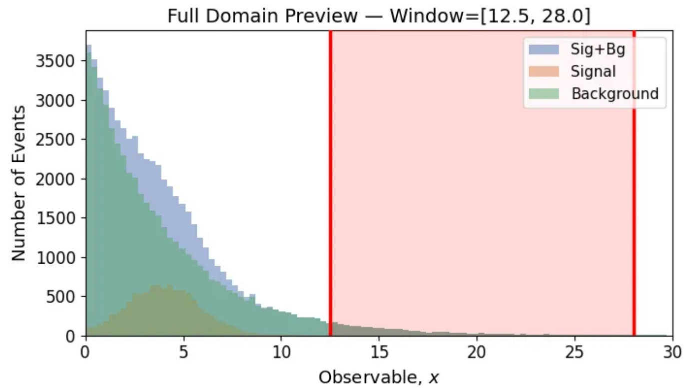
Logits = 17.76



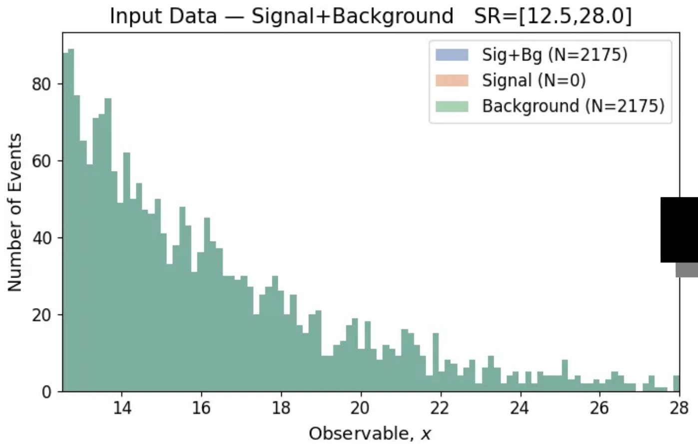
Entropy, H[q] = -3.464

Example Runs:

Background only SR:
During training,
θ sampled from prior



X ≥ 12.5 , X ≤ 28

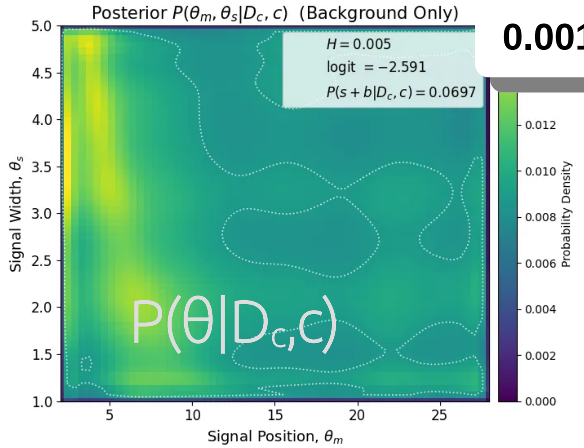


Significance = 0

RACNN-Classiflows

$P(s+b|D_c,c) = 0.0697$

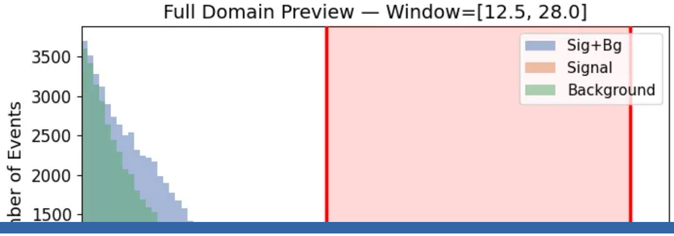
Logits = -2.591



0.0016

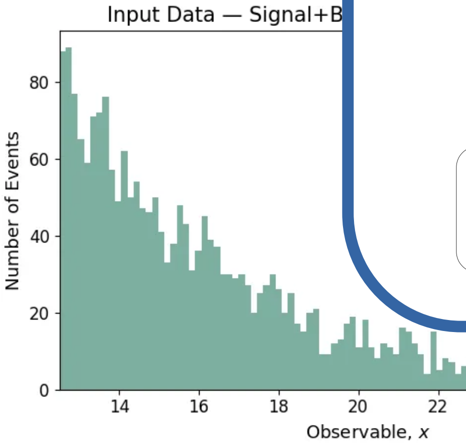
Entropy, H[q] = 0.005

Example Runs:



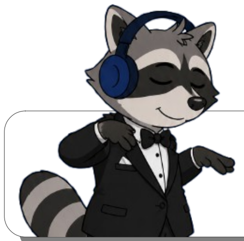
Background only SR:
During training,
θ sampled fro

X ≥ 12



Significance = 0

Try
RACNN-Classiflows
yourself!

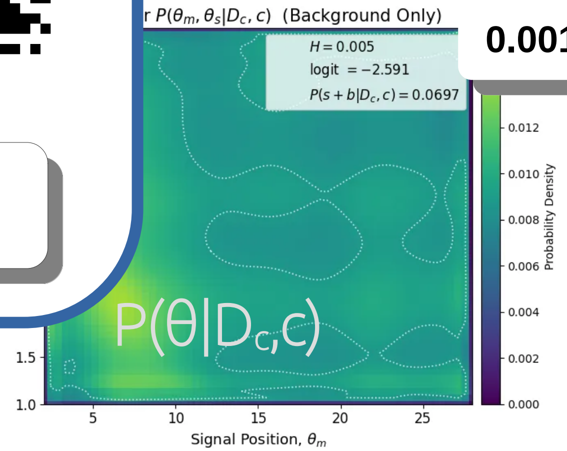


kys-sheng-racnn-bumphunt-demo.hf.space

$b|D_c,c) = 0.0697$

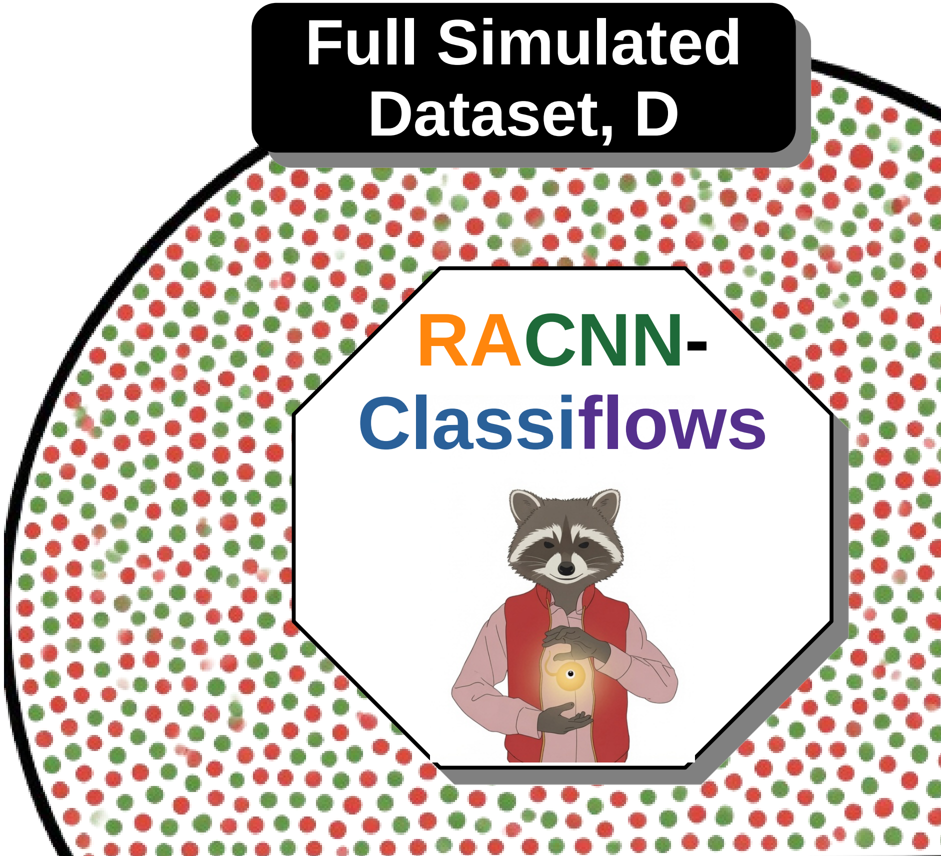
Logits = -2.591

0.0016

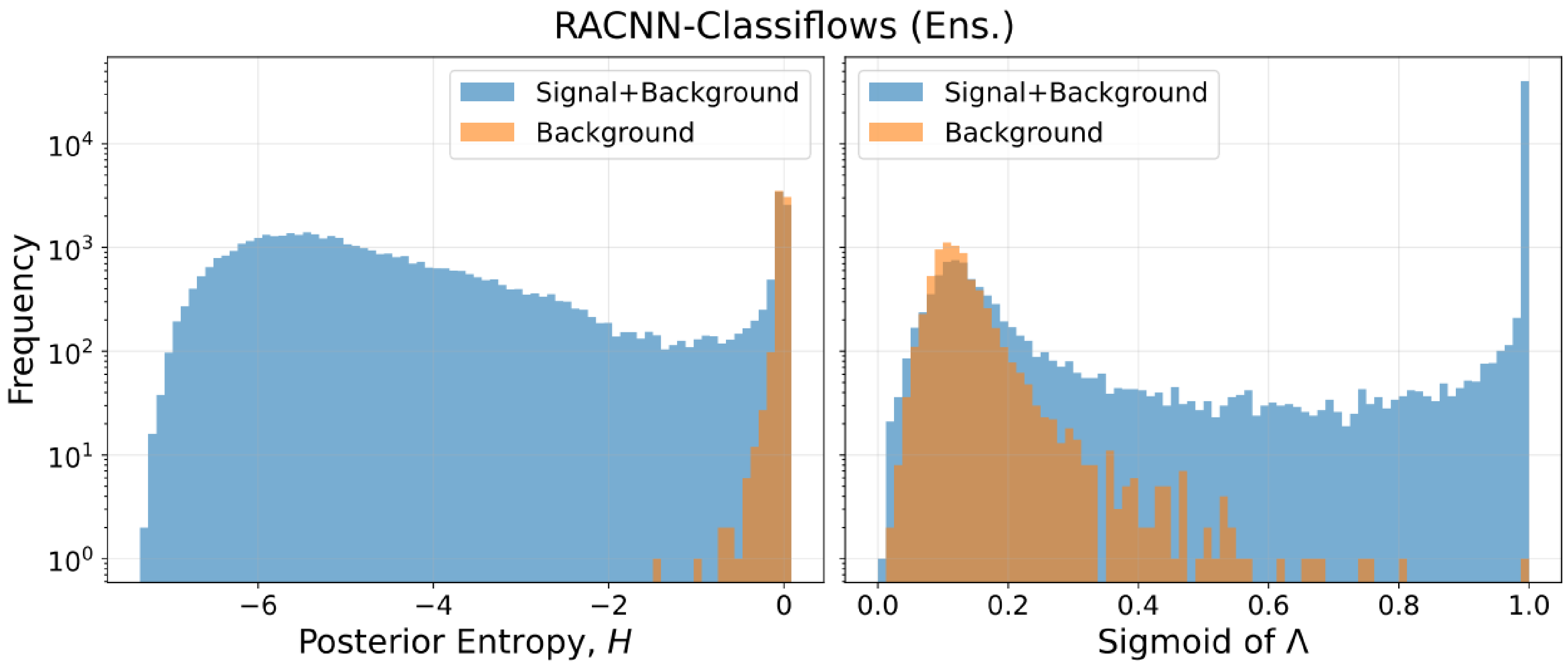


Entropy, H[q] = 0.005

Great,
detect and infer
from any region
any more we can
get out of it?

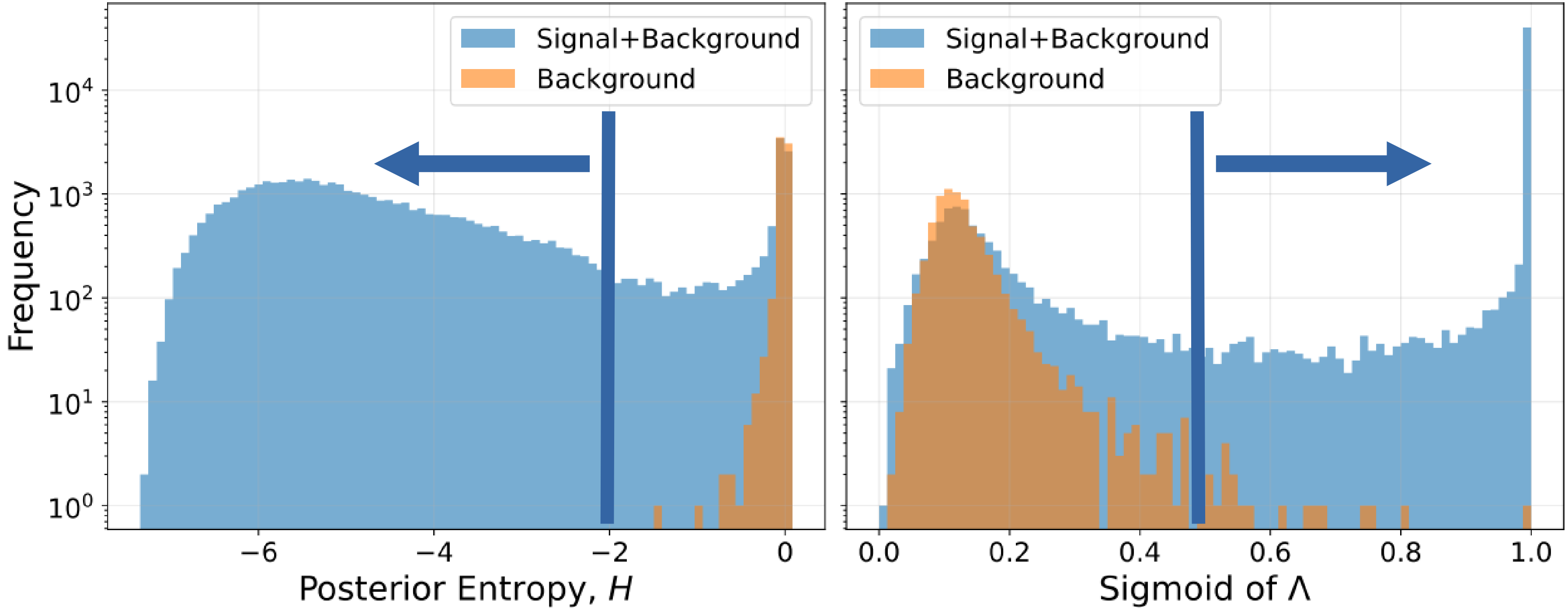


If we look at Entropy and Logits by class...



P(s+b|Dc,c)

Posterior Entropy looks a lot like P(s+b|Dc,c)
Flow models can do signal detection too!



P(s+b|Dc,c)

Signal Detection

Model	AUC_H	AUC_C	Model	AUC_H	AUC_C
RACNN-ClassiFlow ens.	0.942	0.946	CNN-ClassiFlow ens.	0.901	0.917
RACNN-ClassiFlow	0.939	0.944	CNN-ClassiFlow	0.901	0.916
RACNN-Flow ens.	**0.944**	—	CNN-Flow ens.	0.900	—
RACNN-Flow	0.939	—	CNN-Flow	0.893	—
RACNN-Class ens.	—	**0.954**	CNN-Class ens.	—	0.921
RACNN-Class	—	0.952	CNN-Class	—	0.919

Signal Detection

Model	AUC_H	AUC_C	Model	AUC_H	AUC_C
RACNN-ClassiFlow ens.	0.942	0.946	CNN-ClassiFlow ens.	0.901	0.917
RACNN-ClassiFlow	0.939	0.944	CNN-ClassiFlow	0.901	0.916
RACNN-Flow ens.	**0.944**	—	CNN-Flow ens.	0.900	—
RACNN-Flow	0.939	—	CNN-Flow	0.893	—
RACNN-Class ens.	—	**0.954**	CNN-Class ens.	—	0.921
RACNN-Class	—	0.952	CNN-Class	—	0.919

- Flows / Posterior Estimation → Signal Detection!
-

Signal Detection

Model	AUC_H	AUC_C	Model	AUC_H	AUC_C
RACNN-ClassiFlow ens.	0.942	0.946	CNN-ClassiFlow ens.	0.901	0.917
RACNN-ClassiFlow	0.939	0.944	CNN-ClassiFlow	0.901	0.916
RACNN-Flow ens.	**0.944**	—	CNN-Flow ens.	0.900	—
RACNN-Flow	0.939	—	CNN-Flow	0.893	—
RACNN-Class ens.	—	**0.954**	CNN-Class ens.	—	0.921
RACNN-Class	—	0.952	CNN-Class	—	0.919

- Flows / Posterior Estimation → Signal Detection!
- Region Conditioning better than without!

Conditional Neural Posterior Estimation

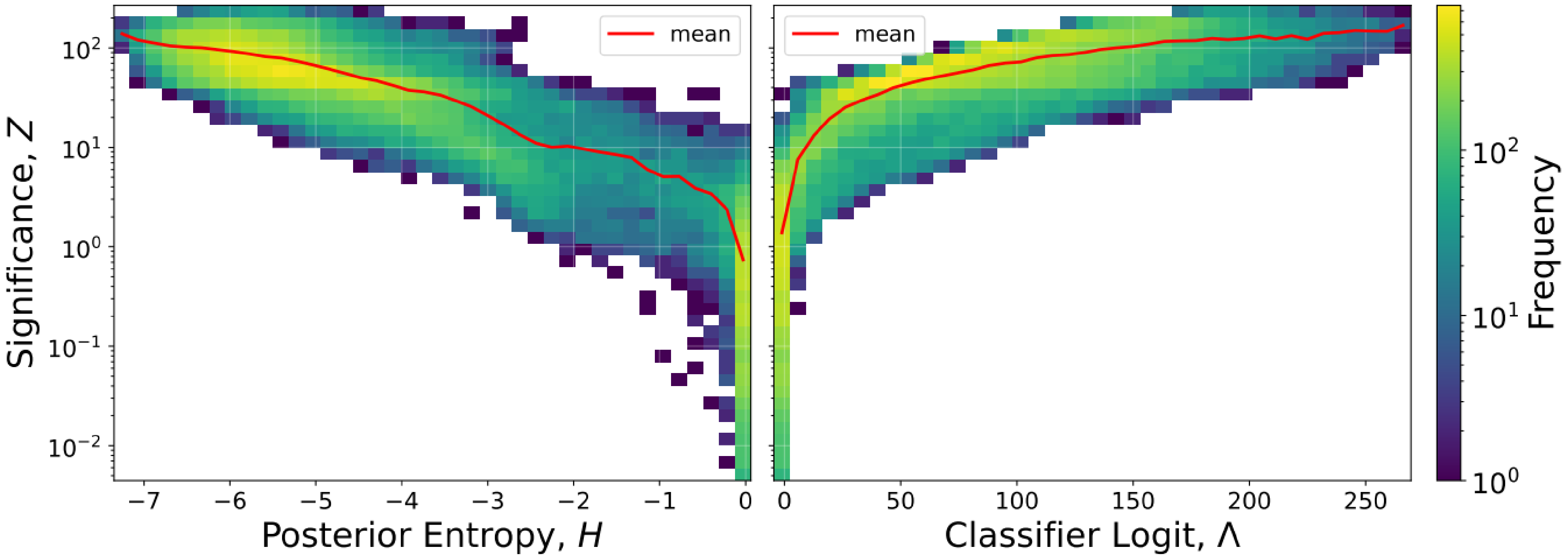
Model	Mean NLL			
	All	$\hat{H} \leq -1$	$\hat{H} \leq -4$	$\hat{H} = \hat{H}^{\theta}_{\text{min}}$
RACNN-ClassiFlow ens.	**-4.602**	**-5.398**	**-6.029**	**-6.462**
RACNN-ClassiFlow	-4.527	-5.333	-5.921	-6.073
RACNN-Flow ens.	-4.490	-5.335	-6.002	-6.433
RACNN-Flow	-4.405	-5.160	-5.662	-5.804
CNN-ClassiFlow ens.	-2.996	-4.067	-5.291	-5.666
CNN-ClassiFlow	-2.737	-3.729	-4.928	-4.896
CNN-Flow ens.	-2.986	-4.093	-5.306	-5.601
CNN-Flow	-2.987	-4.107	-5.232	-5.537

Conditional Neural Posterior Estimation

Model	Mean NLL			
	All	$\hat{H} \leq -1$	$\hat{H} \leq -4$	$\hat{H} = \hat{H}^{\theta}_{\mathrm{min}}$
RACNN-ClassiFlow ens.	**-4.602**	**-5.398**	**-6.029**	**-6.462**
RACNN-ClassiFlow	-4.527	-5.333	-5.921	-6.073
RACNN-Flow ens.	-4.490	-5.335	-6.002	-6.433
RACNN-Flow	-4.405	-5.160	-5.662	-5.804
CNN-ClassiFlow ens.	-2.996	-4.067	-5.291	-5.666
CNN-ClassiFlow	-2.737	-3.729	-4.928	-4.896
CNN-Flow ens.	-2.986	-4.093	-5.306	-5.601
CNN-Flow	-2.987	-4.107	-5.232	-5.537

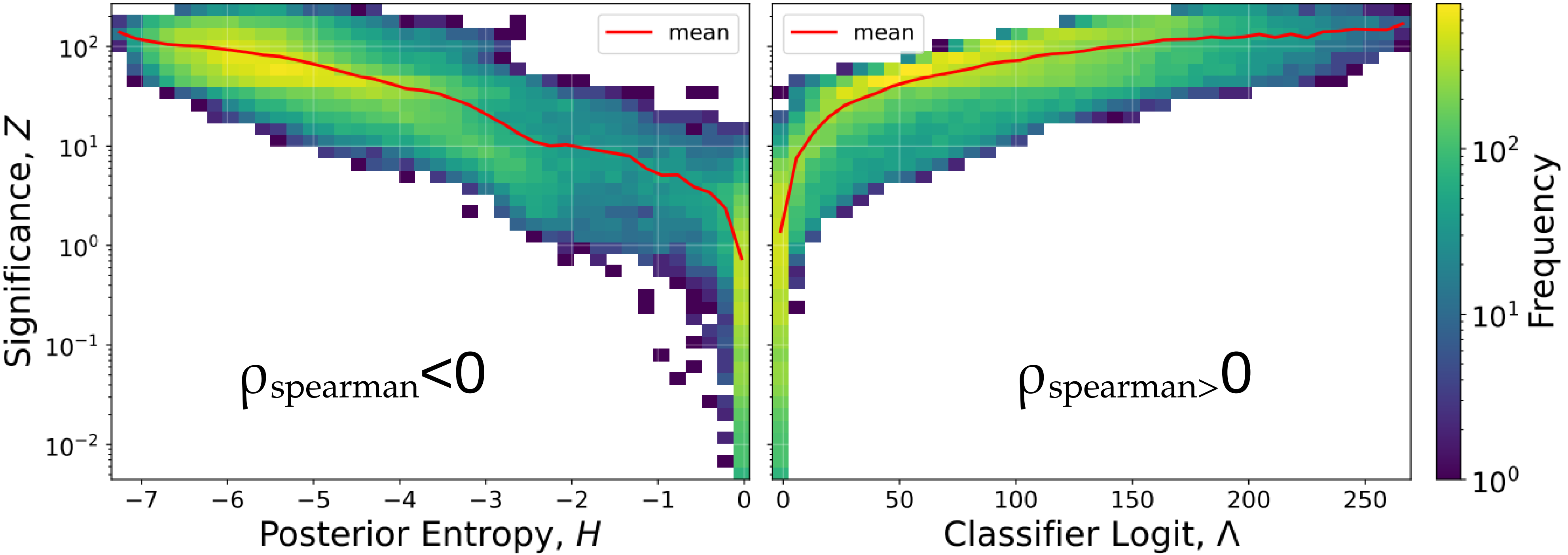
- One can use Entropy threshold to cut away bad SRs

If we look at their relation with significance...

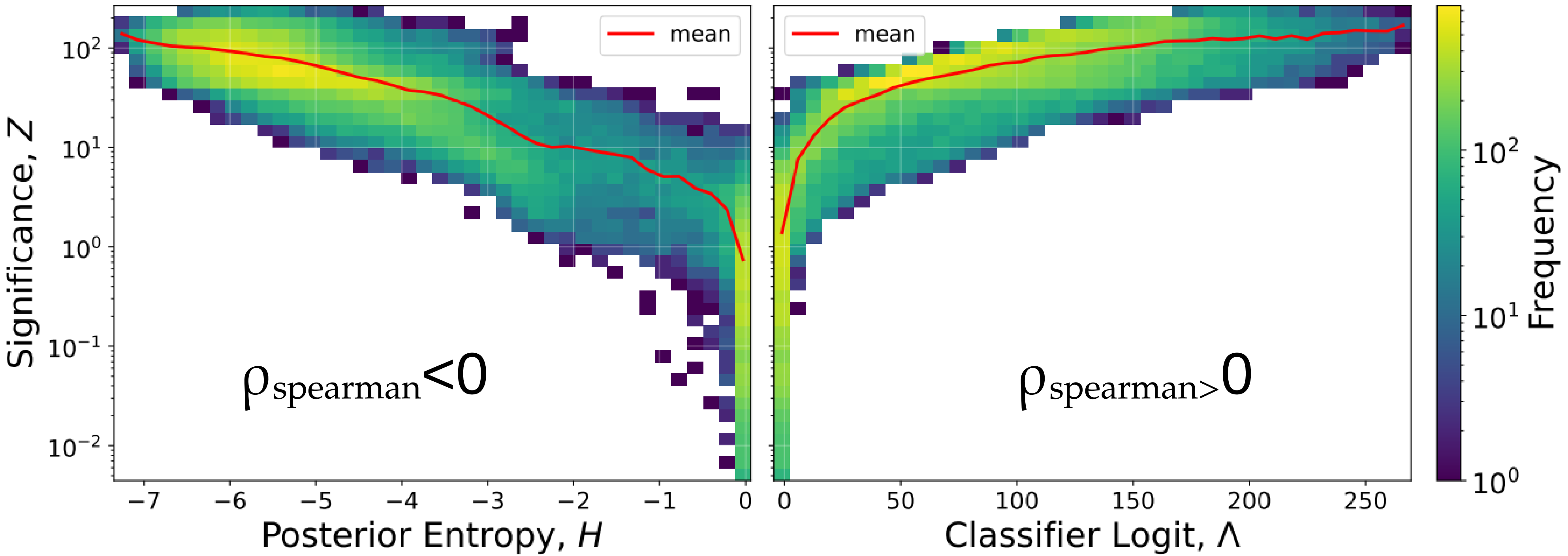


Lower Entropy, Higher Significance

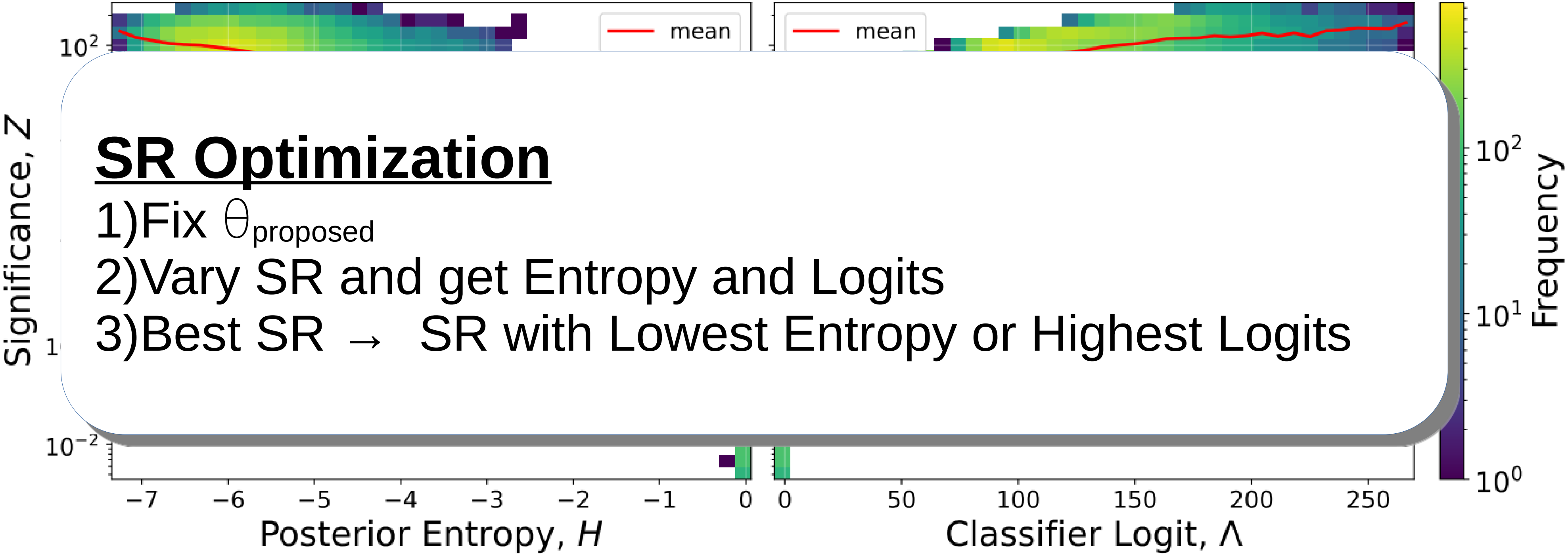
Higher Logits, Higher Significance



Entropy and Logit act as proxy metric for Region Ranking!



Entropy and Logit act as proxy metric for Region Ranking!



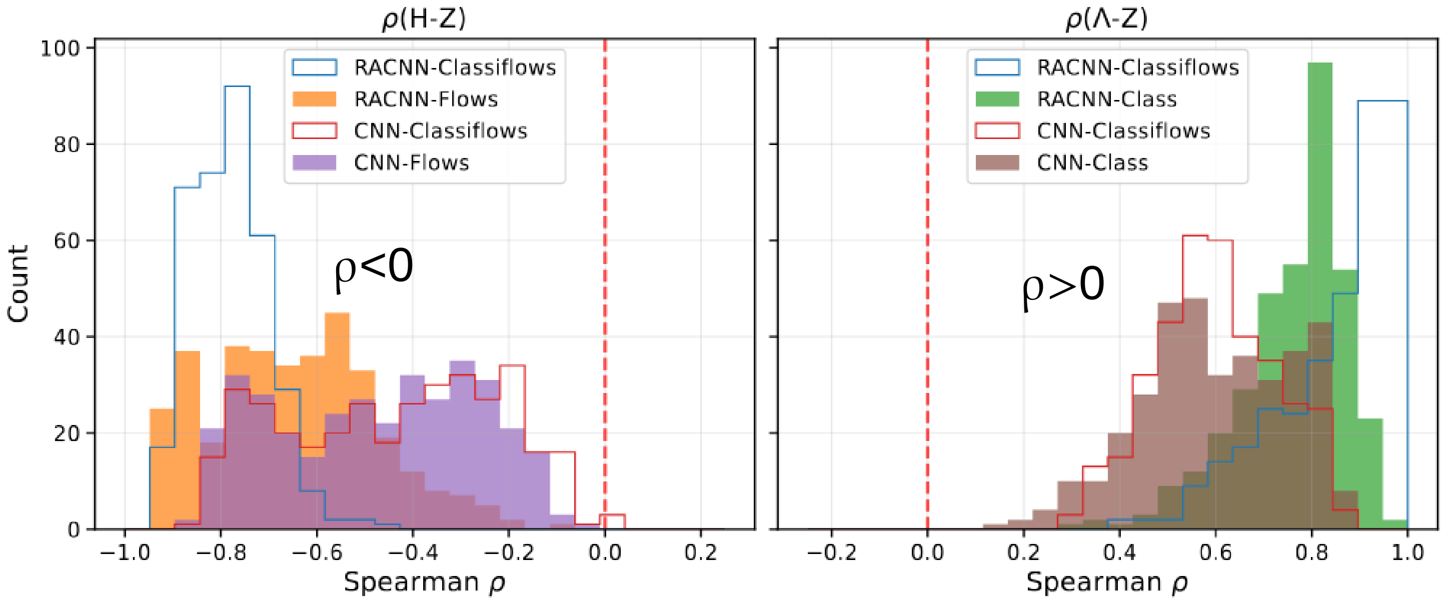
SR Optimization

1)Fix $\theta_{proposed}$

2)Vary SR and get Entropy and Logits

3)Best SR → SR with Lowest Entropy or Highest Logits

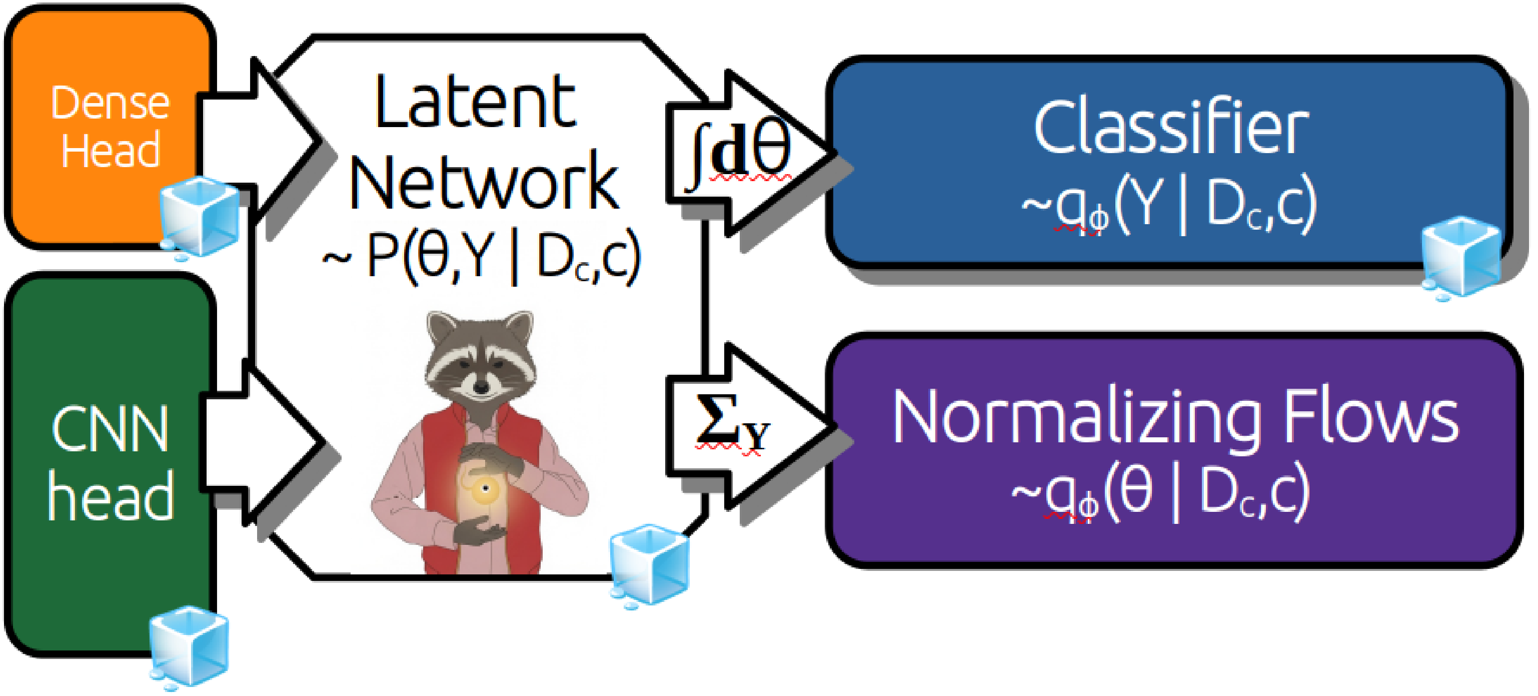
Only works, if the relations holds for all $\theta_{proposed}$ and It does!



Spearman Correlation → Spread but mostly right signs!

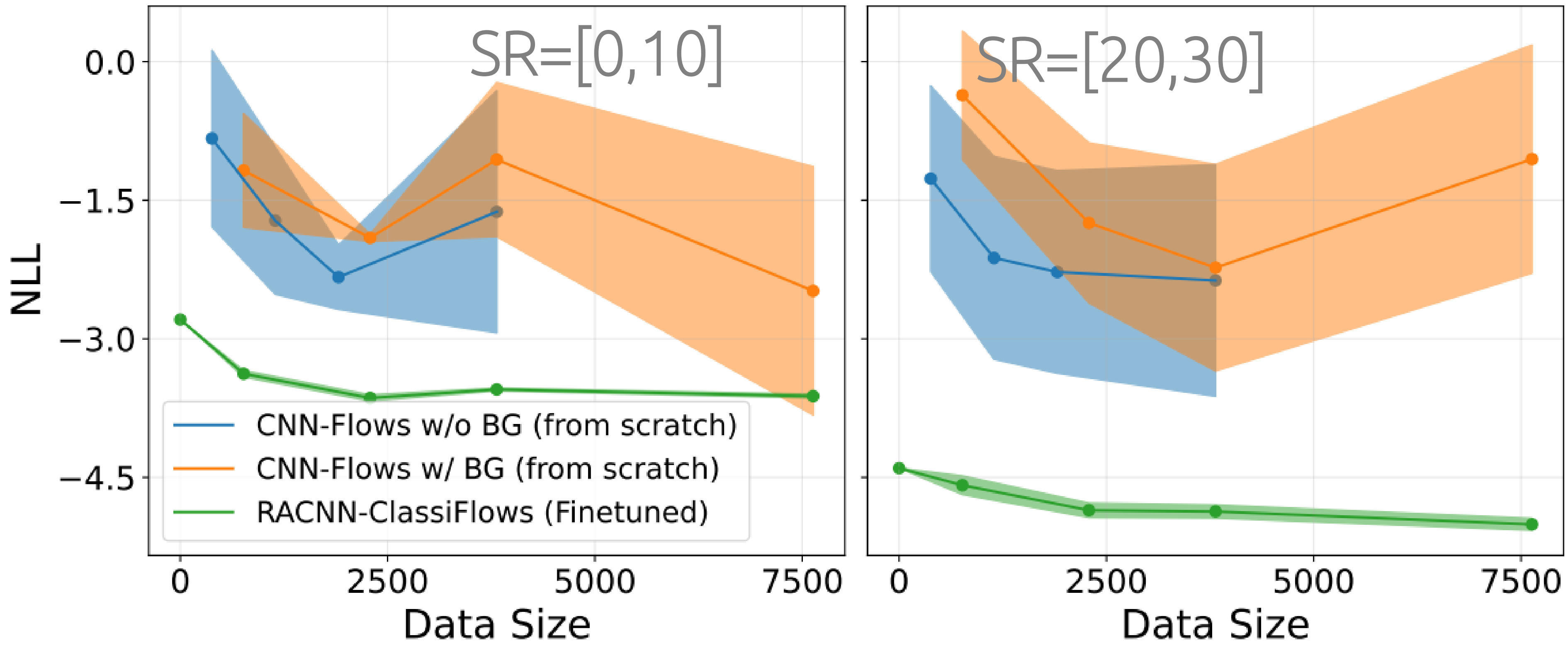
Fine-tuning

You can finetune with more data once SR fixed!



Fine-tuning

Better than from scratch!



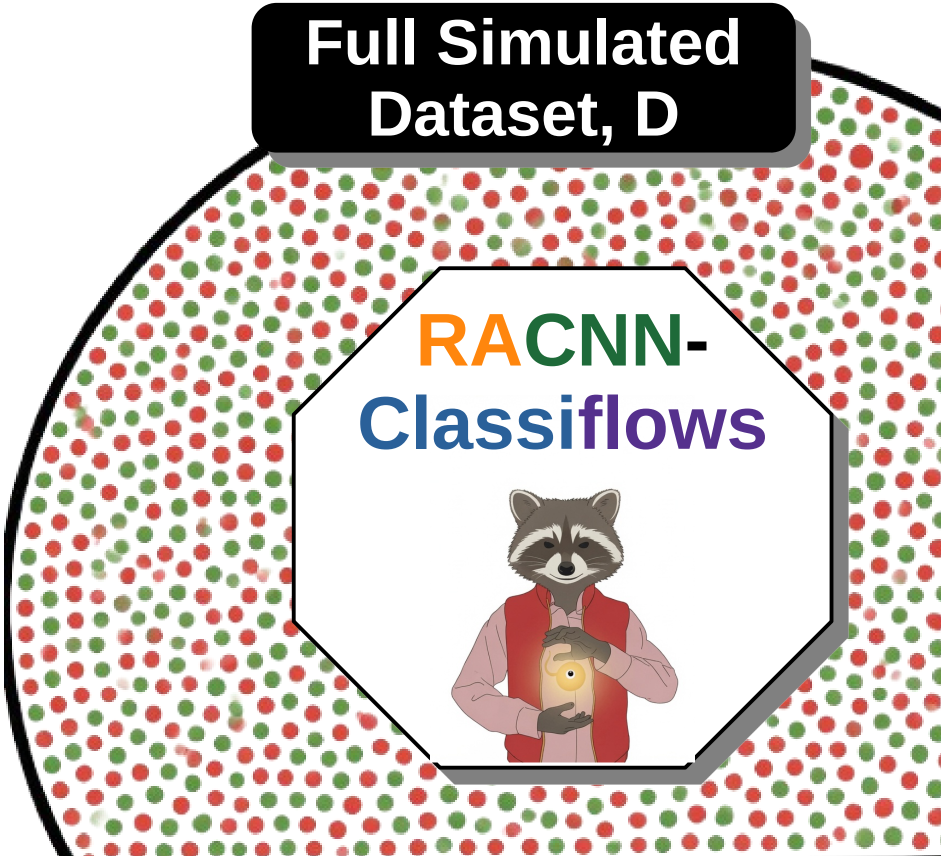
Multitask, Every Region, All at once!

Conditioned on:
- Region Information

Trained on :
- Detection and Inference

Extra/”Emergent” Capabilities:
- Flows do Detection too
- Region Optimization becomes possible!

Finetuning can improve further!



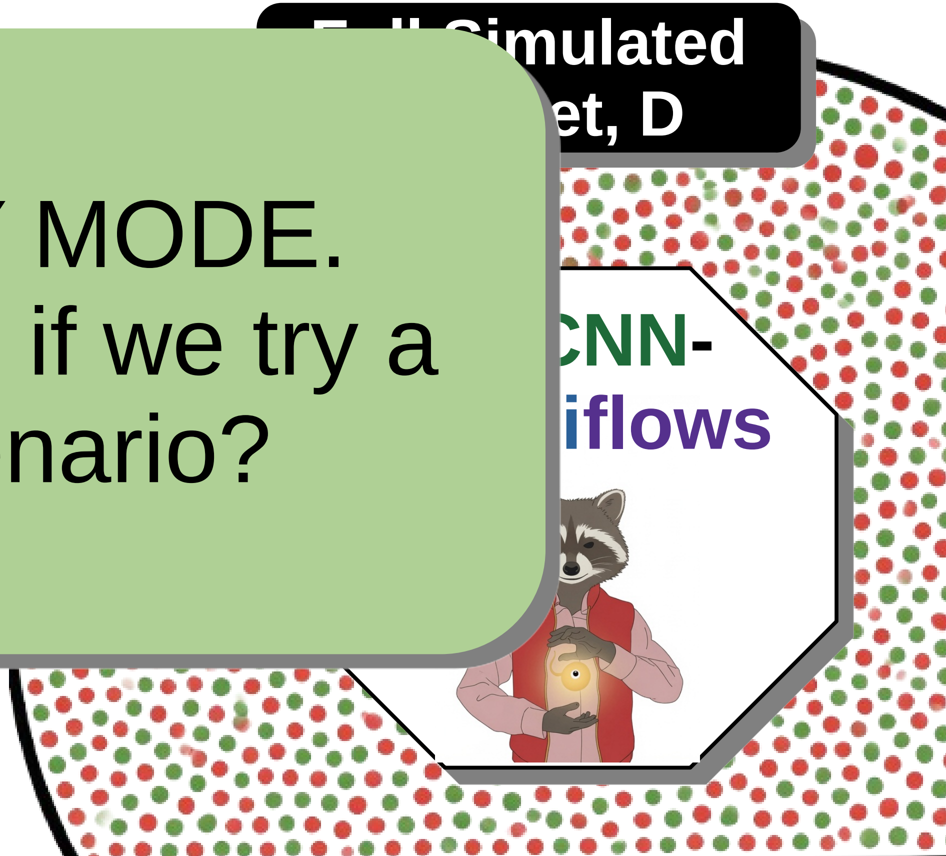
Multitask, Every Region, All at once!

Conditioned on:
• Region Inf

Trained on
• Detection

Extra/"Eme
• Flows do
• Region C
possible!

So far, EASY MODE.
What happens if we try a harder scenario?



Finetuning can improve further!

Stress-test with Dark Matter Search

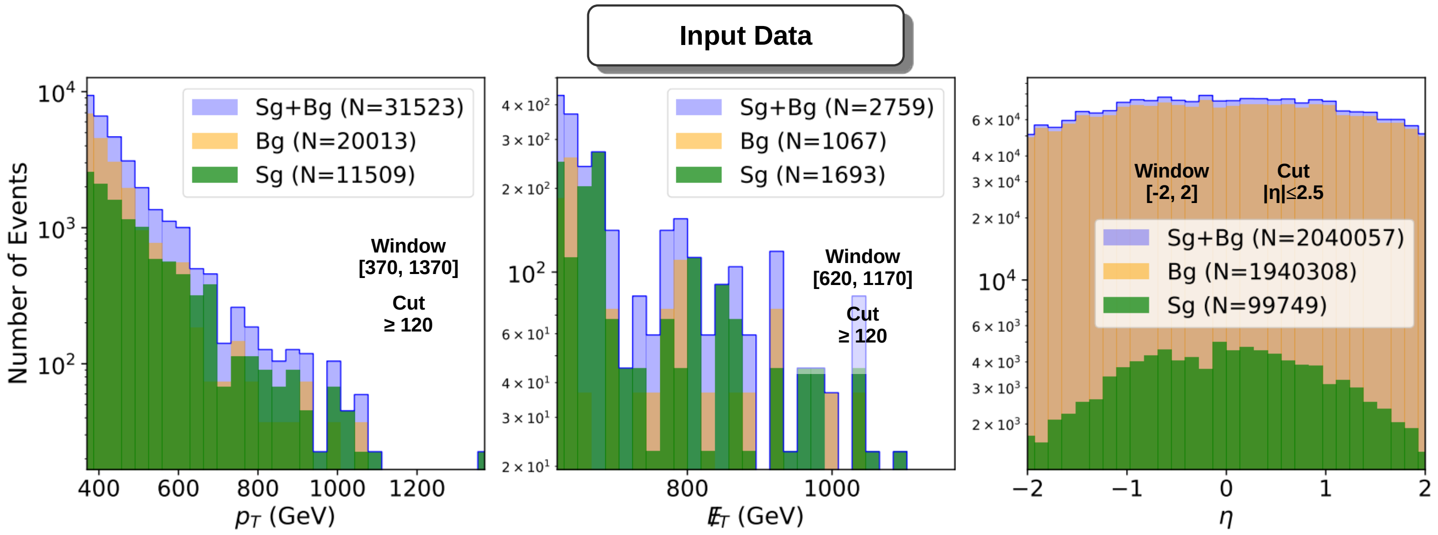
Simplified s-channel Dark Matter Model
Complex Scalar DM with Vector Mediator

Monojet ($pp \rightarrow j+X+X$)

Infer :
Dark Matter Mass, $\mathbf{m_{DM}}$
Dm-Mediator Coupling, $\mathbf{g}$

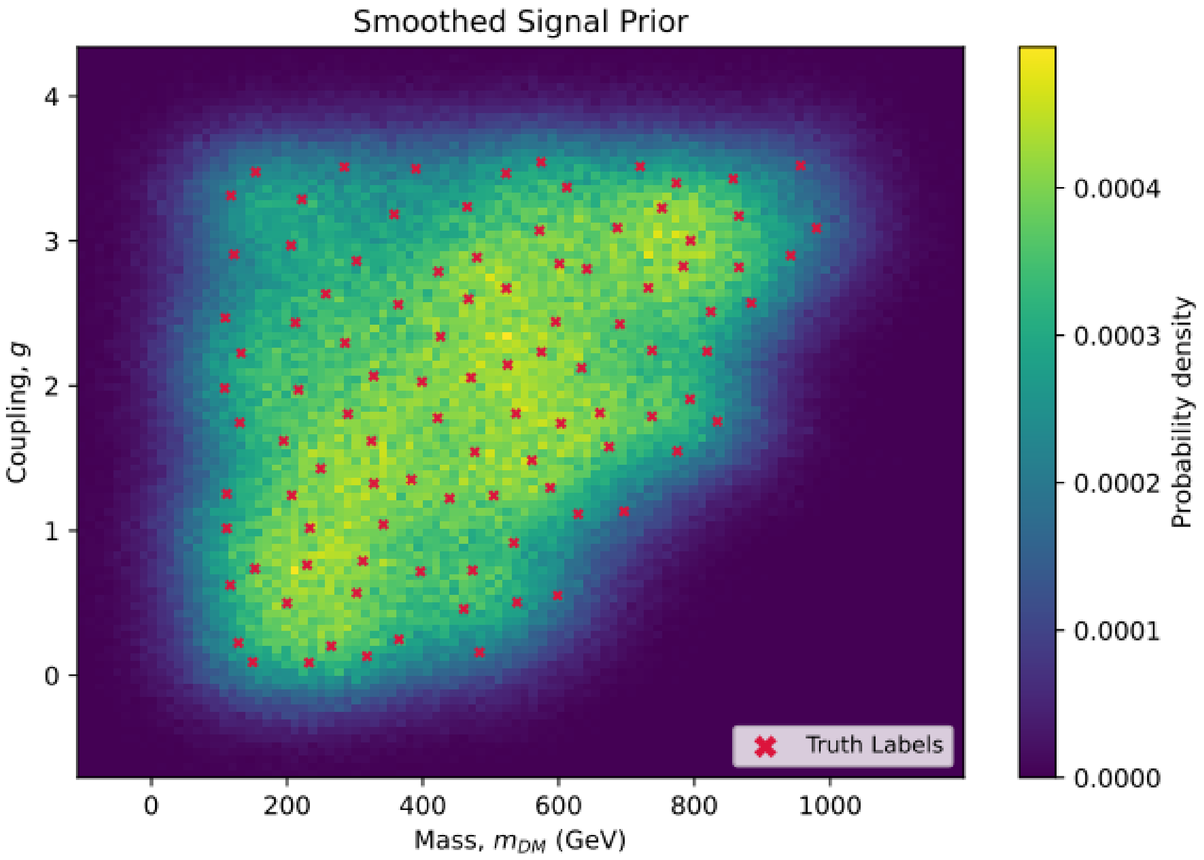
How is it harder?

- Non-resonant distributions
- 3 observables:
  - 8 cut variables
  - Many Combinations
  - O(10^4) SR per θ



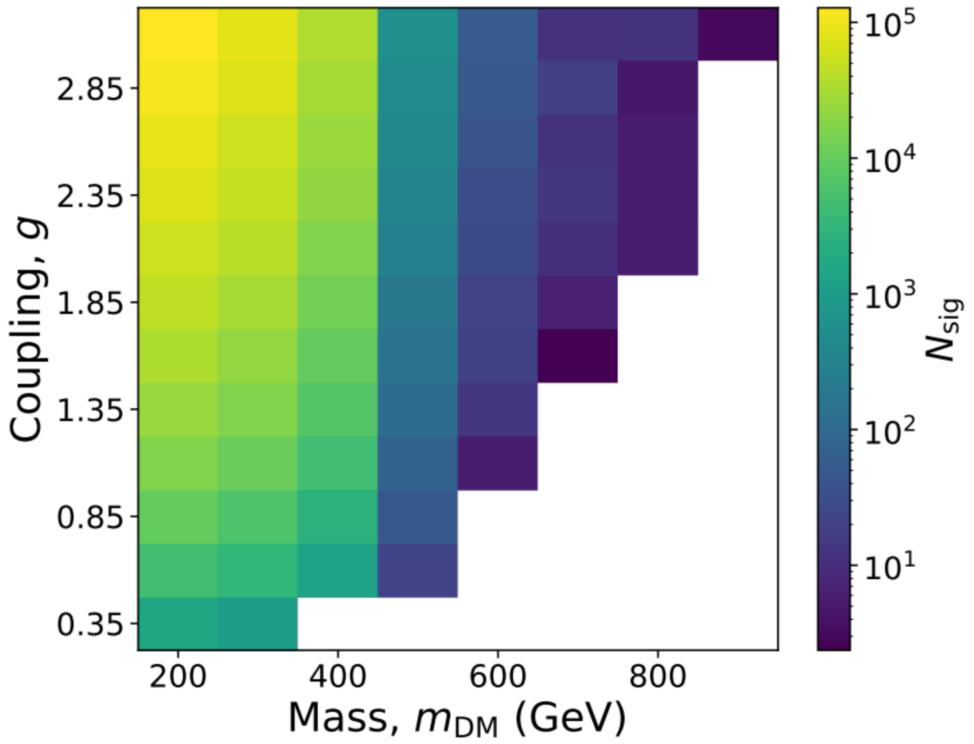
How is it harder?

- Non-resonant distributions
- 3 observables:
  - 8 cut variables
  - Many Combinations
  - $O(10^4)$ SR per θ
- Fixed simulation budget
  - $O(10^2)$ θ
  - Need Gaussian Smoothing



How is it harder?

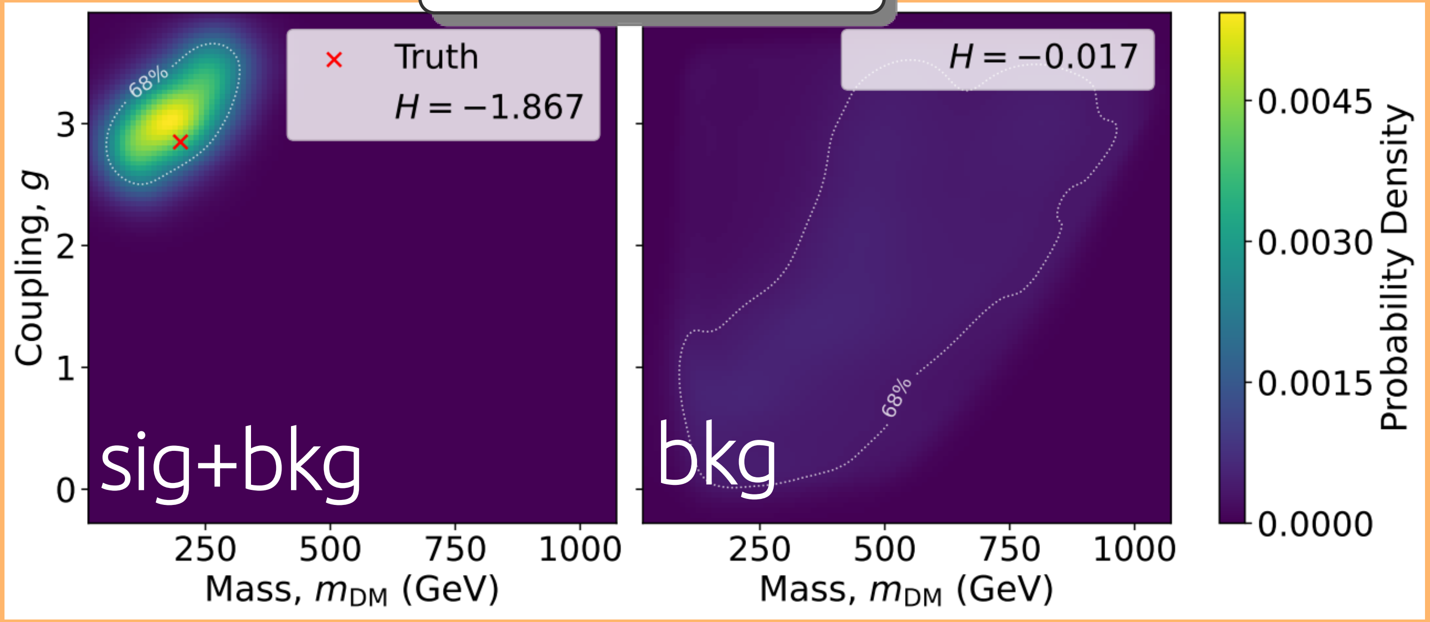
- Non-resonant distributions
- 3 observables:
  - 8 cut variables
  - Many Combinations
  - $O(10^4)$ SR per θ
- Fixed simulation budget
  - $O(10^2)$ θ
  - Need Gaussian Smoothing
- Signal yields : $1\sim10^5$ events



- Non-r
- 3 obs
  - 8 c
  - Ma
  - O(1
- Fixed
  - O(1
  - Ne
- Signa

Does it work?

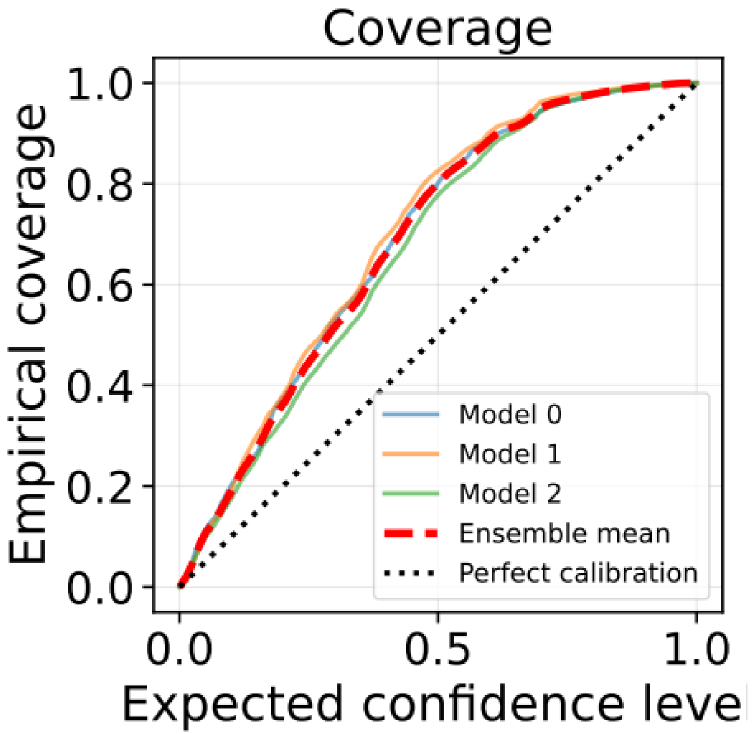
Posterior



it WORK-**ish**

Only in certain regimes with some quirks!

Too Few Training θs, Too Many Signal Regions



Conservative posterior

Gaussian Smoothing

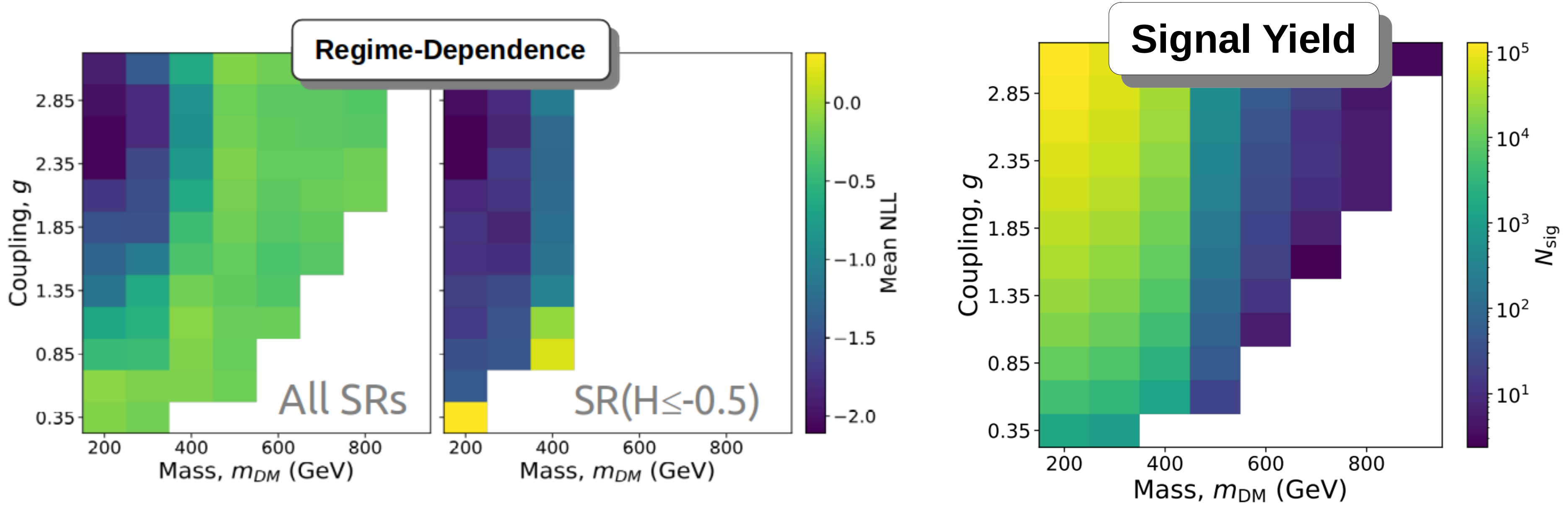
Introduced new epistemic uncertainties

Future :

Fine-tuning is now necessary

Wide Signal-Yield Range ($N_{sig} \rightarrow 1\sim10^5$ events)

→ Regime-dependent behaviour



Future : Improve data-processing network
(CNN not best approach)

Many SRs but limited cut diversity

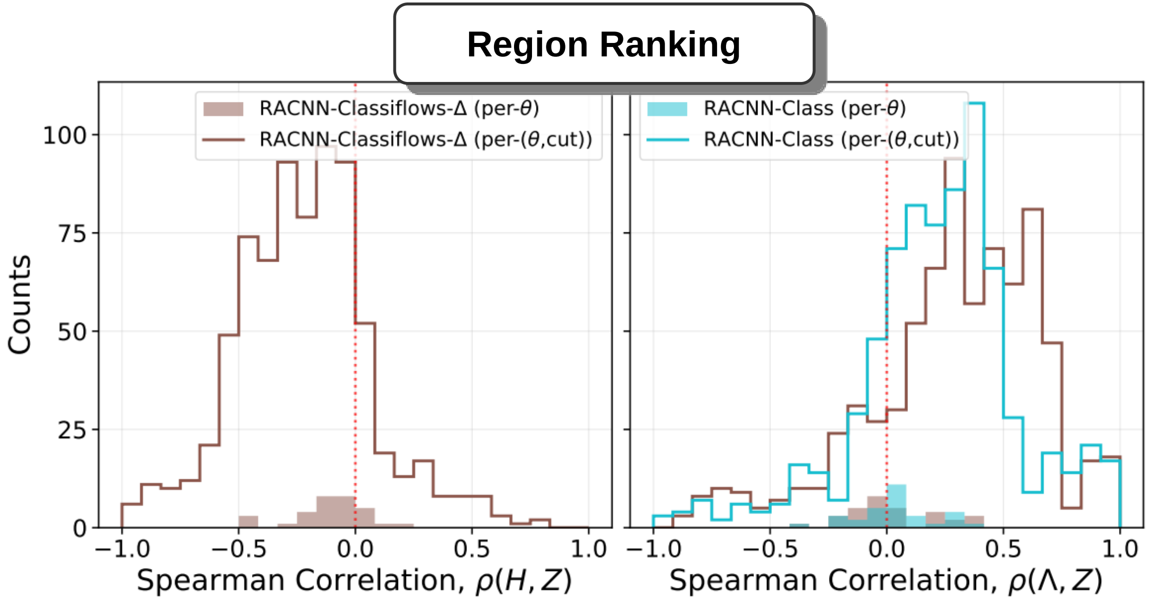
Problem : Fixed allow SR cut choices, for easy preprocessing

→ **Unreliable region ranking**

unless evaluated per-(θ,hard cut)

Implementation Problem

Future : On-the-fly SR generation



CNPE

Model	Mean NLL			
	All	$\hat{H} \leq -0.5$	$\hat{H} \leq -1$	$\hat{H} = \hat{H}^{\theta}_{\text{min}}$
RACNN-ClassiFlow-$\mathscr{L}$ ens.	**-0.870**	**-1.801**	**-1.889**	-0.531
RACNN-ClassiFlow-Δ ens.	-0.624	-1.689	-1.795	**-0.719**
RACNN-Flow-$\mathscr{L}$ ens.	-0.094	—	—	-0.095
RACNN-Flow-Δ ens.	-0.088	—	—	-0.087
CNN-ClassiFlow-$\mathscr{L}$ ens.	-0.695	-1.705	-1.834	-0.203
CNN-ClassiFlow-Δ ens.	-0.442	-1.543	-1.782	-0.416
CNN-Flow-$\mathscr{L}$ ens.	-0.103	—	—	-0.101
CNN-Flow-Δ ens.	-0.074	—	—	-0.073

Signal Detection

Model	AUC_C	AUC_H	Model	AUC_C	AUC_H
RACNN-Classifier ens.	**0.807**	—	CNN-Classifier ens.	0.722	—
RACNN-ClassiFlow-$\mathscr{L}$ ens.	0.746	**0.703**	CNN-ClassiFlow-$\mathscr{L}$ ens.	0.731	0.672
RACNN-ClassiFlow-Δ ens.	0.722	0.650	CNN-ClassiFlow-Δ ens.	0.693	0.607
RACNN-Flow-$\mathscr{L}$ ens.	—	0.500	CNN-Flow-$\mathscr{L}$ ens.	—	0.500
RACNN-Flow-Δ ens.	—	0.500	CNN-Flow-Δ ens.	—	0.500

Flow only model cant learn at all!

Classifier regularizes the classiflow's training

CNPE

Model	Mean NLL All	$\hat{H} \leq -0.5$	$\hat{H} \leq -1$	$\hat{H} = \hat{H}^{\theta}_{\min}$
RACNN-ClassiFlow-$\mathscr{L}$ ens.	**-0.870**	**-1.801**	**-1.889**	-0.531
RACNN-ClassiFlow-Δ ens.	-0.624	-1.689	-1.795	**-0.719**
RACNN-Flow-$\mathscr{L}$ ens.	-0.094	—	—	-0.095
RACNN-Flow-Δ ens.	-0.088	—	—	-0.087
CNN-ClassiFlow-$\mathscr{L}$ ens.	-0.695	-1.705	-1.834	-0.203
CNN-ClassiFlow-Δ ens.	-0.442	-1.543	-1.782	-0.416
CNN-Flow-$\mathscr{L}$ ens.	-0.103	—	—	-0.101
CNN-Flow-Δ ens.	-0.074	—	—	-0.073

Signal Detection

Model	AUC_C	AUC_H	Model	AUC_C	AUC_H
RACNN-Classifier ens.	**0.807**	—	CNN-Classifier ens.	0.722	—
RACNN-ClassiFlow-$\mathscr{L}$ ens.	0.746	**0.703**	CNN-ClassiFlow-$\mathscr{L}$ ens.	0.731	0.672
RACNN-ClassiFlow-Δ ens.	0.722	0.650	CNN-ClassiFlow-Δ ens.	0.693	0.607
RACNN-Flow-$\mathscr{L}$ ens.	—	0.500	CNN-Flow-$\mathscr{L}$ ens.	—	0.500
RACNN-Flow-Δ ens.	—	0.500	CNN-Flow-Δ ens.	—	0.500

O(100K) parameters

Towards an FM at analysis level?

I. Fixed a BSM model

II. Fixed a final state signatures

III. **Build** Many Signal Regions (SR)

   I. **Detect** signals from Many SR

   II. **Infer** parameters from Many SR

   III. **Finetune-able** to Many SR

A
Foundation
Model
at the
Analysis Level
For
an
Analysis

THE Foundation Model at the Analysis Level

I. **Any** BSM model →SMEFT? →Parameterized AD?

II. **Any** final state signatures →Minimal Set Signature ?

III. Build **Any** Signal Regions (SR)

  I. Detect signals from **Any** SR

  II. Infer parameters from **Any** SR

  III. Finetune-able to **Any** SR

→ Include Systematic Uncertainties
→ Better Architecture (RAT-Classiflows...)
→ On-the-fly SR building

Thanks!!

kys-sheng-racnn-bumphunt-demo.hf.space

Try RACNN-Classiflows yourself!

Backup

BumpHunt NLL

Model	Mean NLL					ρ_p
	All	$\hat{H} \leq -1$	$\hat{H} \leq -4$	$\hat{H} = \hat{H}^{\theta}_{\text{min}}$	$\text{NLL} = \text{NLL}^{\theta}_{\text{min}}$	
RACNN-ClassiFlow ens.	**-4.602**	**-5.398**	**-6.029**	**-6.462**	-8.083	**0.975**
RACNN-ClassiFlow	-4.527	-5.333	-5.921	-6.073	**-8.494**	0.933
RACNN-Flow ens.	-4.490	-5.335	-6.002	-6.433	-7.730	0.971
RACNN-Flow	-4.405	-5.160	-5.662	-5.804	-8.205	0.914
CNN-ClassiFlow ens.	-2.996	-4.067	-5.291	-5.666	-7.110	0.960
CNN-ClassiFlow	-2.737	-3.729	-4.928	-4.896	-8.260	0.893
CNN-Flow ens.	-2.986	-4.093	-5.306	-5.601	-7.064	0.954
CNN-Flow	-2.987	-4.107	-5.232	-5.537	-7.691	0.919

Table 2: Mean test NLL in the bump-hunt study. *All*: unconditional mean over all (θ, SR) test pairs. $\hat{H} \leq -1$, $\hat{H} \leq -4$: mean over pairs satisfying the entropy cut. $\hat{H} = \hat{H}^{\theta}_{\text{min}}$: for each θ, pick the SR with lowest $\hat{H}$, then average over θ. $\text{NLL} = \text{NLL}^{\theta}_{\text{min}}$: same but selecting by lowest NLL. Bold marks the best per column; ρ_p: Pearson correlation between per-example NLL and $\hat{H}$; “ens.”: ensemble.

BumpHunt AUC

Model	AUC_H	AUC_C	Model	AUC_H	AUC_C
RACNN-ClassiFlow ens.	0.942	0.946	CNN-ClassiFlow ens.	0.901	0.917
RACNN-ClassiFlow	0.939	0.944	CNN-ClassiFlow	0.901	0.916
RACNN-Flow ens.	0.944	—	CNN-Flow ens.	0.900	—
RACNN-Flow	0.939	—	CNN-Flow	0.893	—
RACNN-Class ens.	—	0.954	CNN-Class ens.	—	0.921
RACNN-Class	—	0.952	CNN-Class	—	0.919

Table 3: ROC AUC for signal detection in the bump-hunt study. AUC_H: entropy used as test statistic. AUC_C: classifier output used as test statistic. "—": model does not have the corresponding head. "ens.": ensemble.

Dark Matter

Model	Mean NLL				Mean NLL by regime	
	All	$\hat{H} \leq -0.5$	$\hat{H} \leq -1$	$\hat{H} = \hat{H}^{\theta}_{\min}$	HYR	LYR
RACNN-ClassiFlow-$\mathscr{L}$ ens.	**-0.870**	**-1.801**	**-1.889**	-0.531	**-1.525**	**-0.277**
RACNN-ClassiFlow-Δ ens.	-0.624	-1.689	-1.795	**-0.719**	-1.051	-0.237
RACNN-Flow-$\mathscr{L}$ ens.	-0.094	—	—	-0.095	-0.077	-0.109
RACNN-Flow-Δ ens.	-0.088	—	—	-0.087	-0.081	-0.095
CNN-ClassiFlow-$\mathscr{L}$ ens.	-0.695	-1.705	-1.834	-0.203	-1.224	-0.216
CNN-ClassiFlow-Δ ens.	-0.442	-1.543	-1.782	-0.416	-0.712	-0.198
CNN-Flow-$\mathscr{L}$ ens.	-0.103	—	—	-0.101	-0.096	-0.109
CNN-Flow-Δ ens.	-0.074	—	—	-0.073	-0.053	-0.092

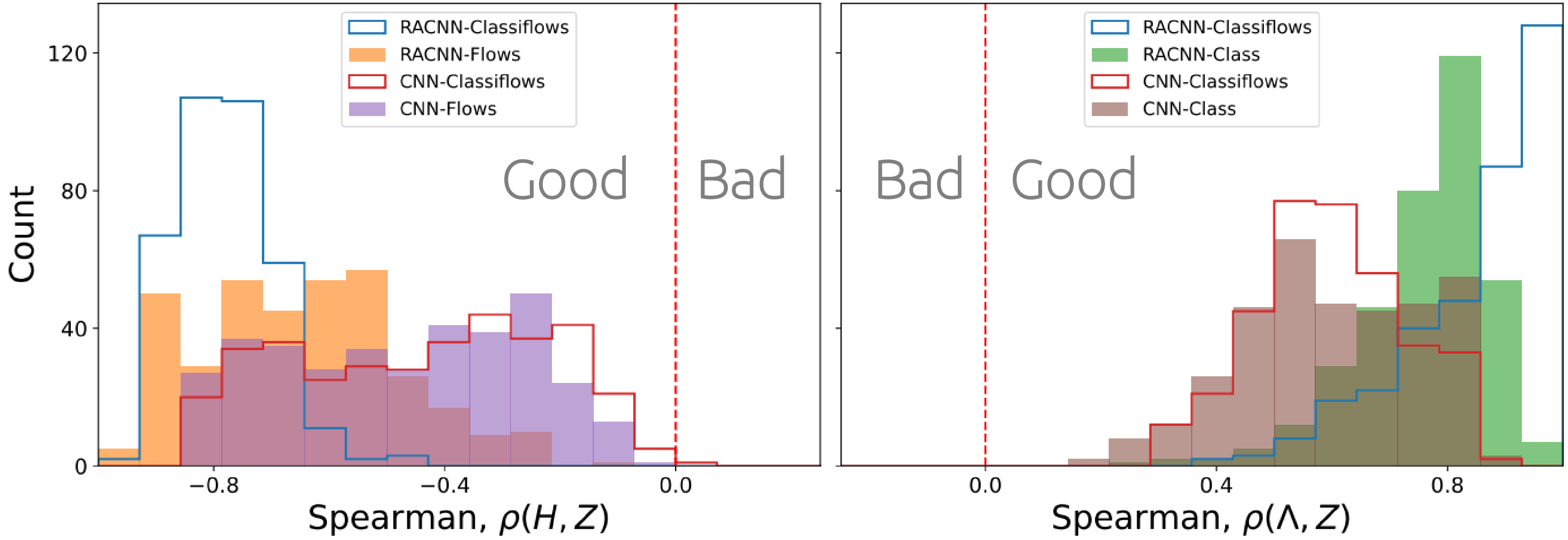
Table 5: Mean test NLL in the mono-jet study. Column definitions follow Table 2; HYR/LYR splits are over θ with $\mu_s \gtrsim 3000$ and the remainder, respectively. Bold marks the best per column. Flow-only models produce no examples below $\hat{H} = -0.5$.

Dark Matter

Model	AUC_C	AUC_H	Model	AUC_C	AUC_H
RACNN-Classifier ens.	0.807	—	CNN-Classifier ens.	0.722	—
RACNN-ClassiFlow-$\mathscr{L}$ ens.	0.746	0.703	CNN-ClassiFlow-$\mathscr{L}$ ens.	0.731	0.672
RACNN-ClassiFlow-Δ ens.	0.722	0.650	CNN-ClassiFlow-Δ ens.	0.693	0.607
RACNN-Flow-$\mathscr{L}$ ens.	—	0.500	CNN-Flow-$\mathscr{L}$ ens.	—	0.500
RACNN-Flow-Δ ens.	—	0.500	CNN-Flow-Δ ens.	—	0.500

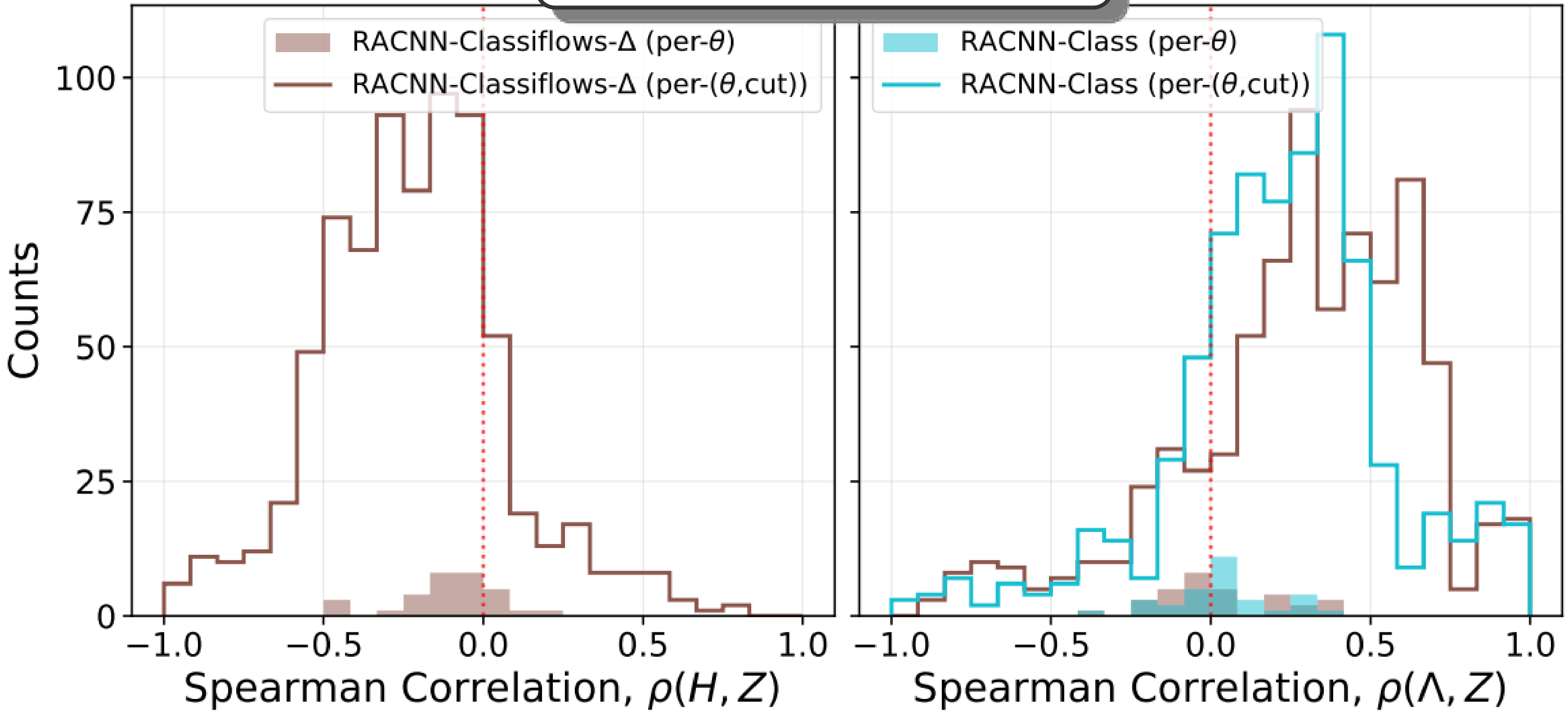
Table 7: ROC AUC for signal detection in the mono-jet study. AUC_C: classifier output. AUC_H: entropy as test statistic. "—": model lacks the corresponding head. "ens.": ensemble.

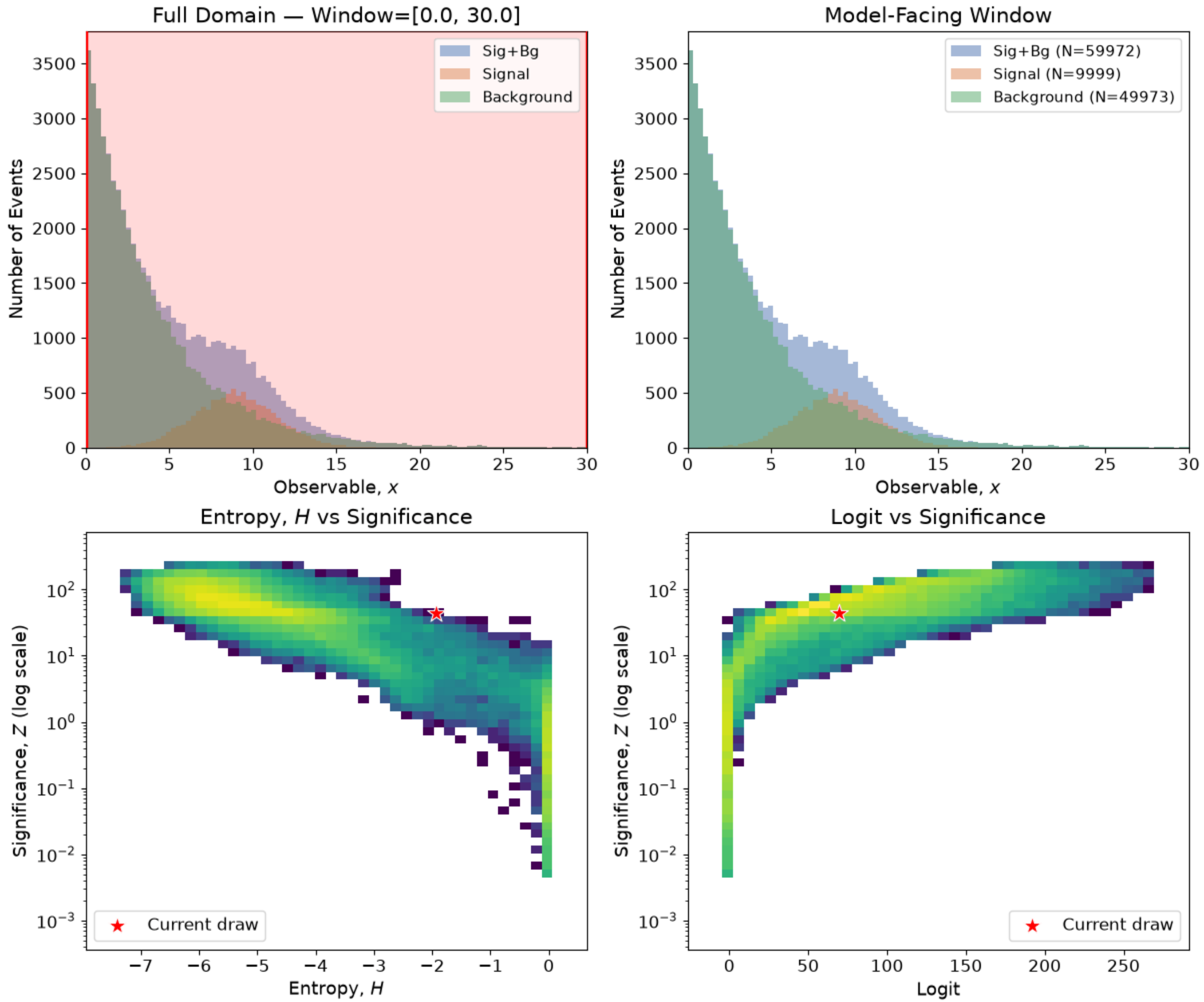
Spearman Correlation ρ for different $\theta_{proposed}$ shows ranking performance



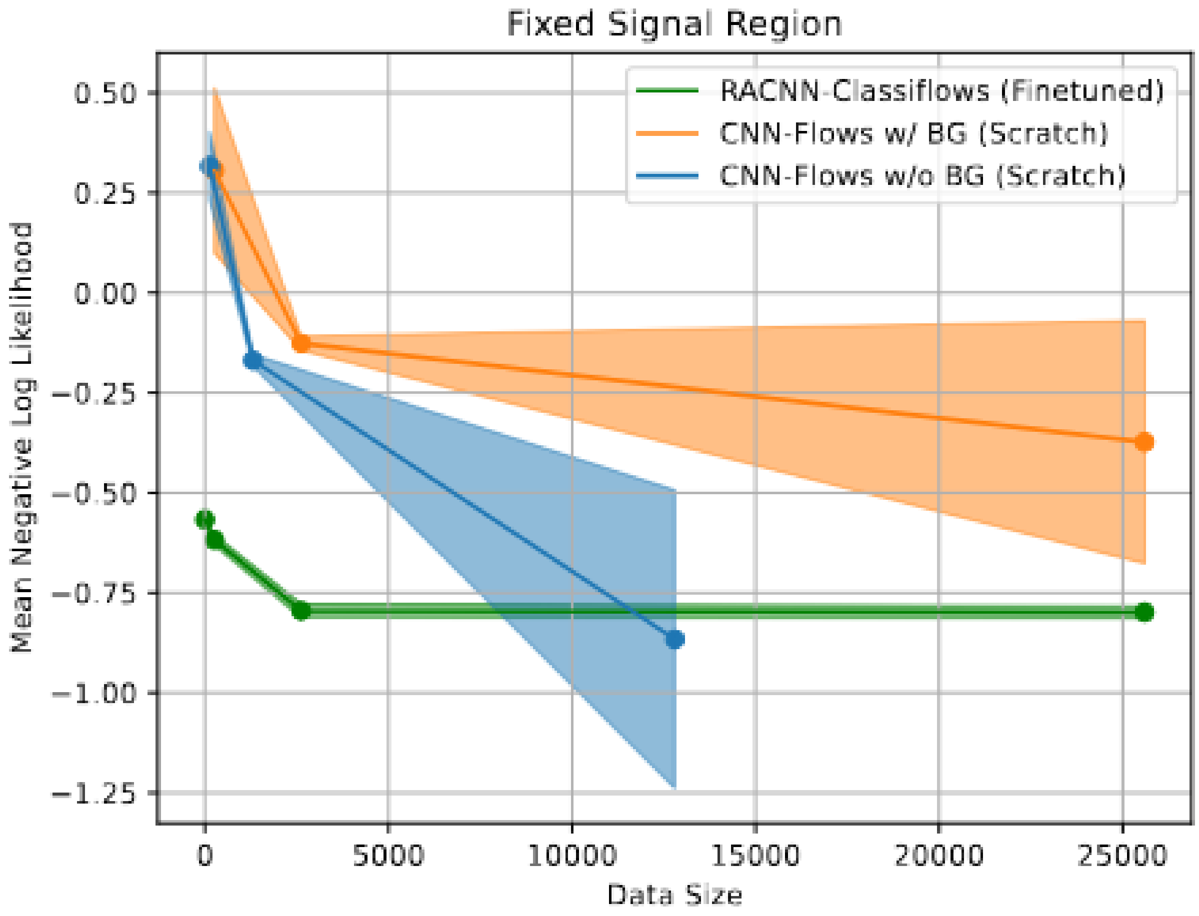
Spearman Correlation ρ for different $\theta_{proposed}$ shows ranking performance

Region Ranking



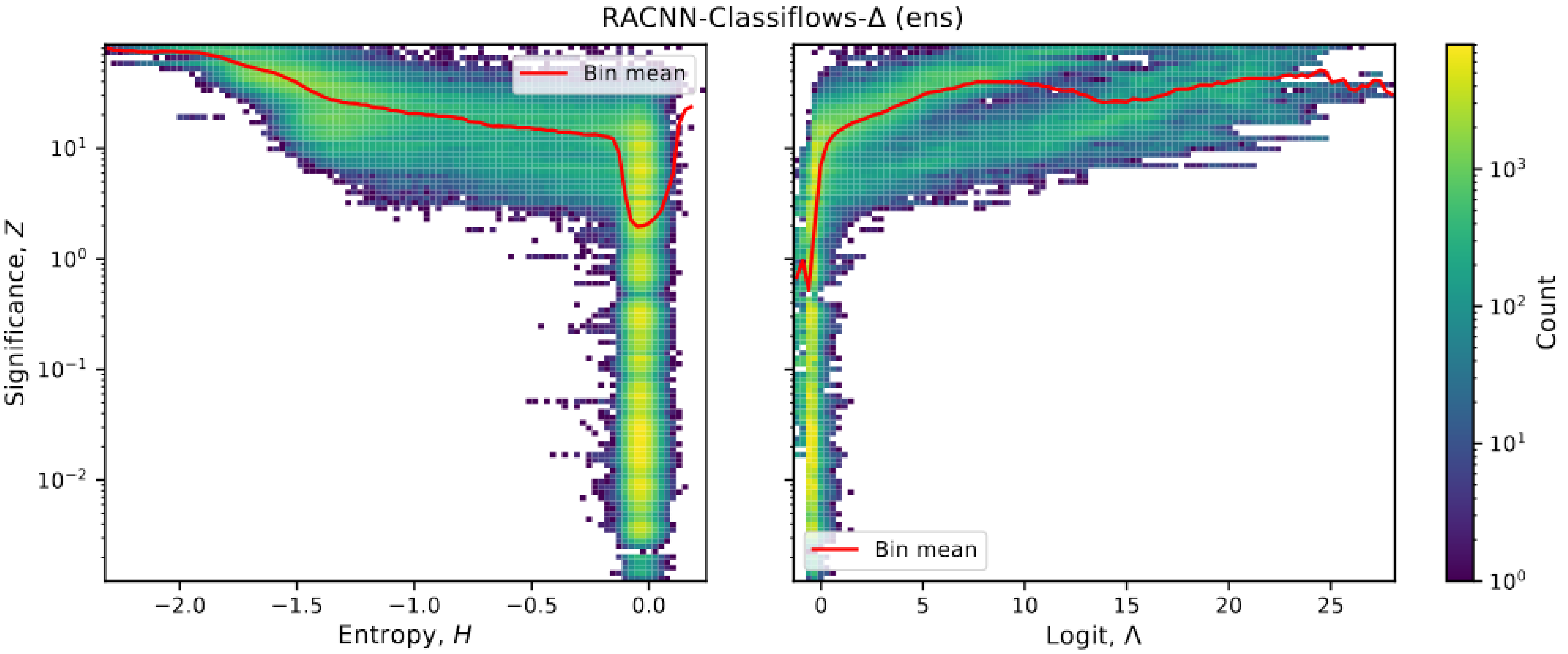

Full Domain — Window=[0.0, 30.0]
Sig+Bg
Signal
Background
Number of Events
Observable, x
Model-Facing Window
Sig+Bg (N=59972)
Signal (N=9999)
Background (N=49973)
Number of Events
Observable, x
Entropy, H vs Significance
Significance, Z (log scale)
Current draw
Entropy, H
Logit vs Significance
Significance, Z (log scale)
Current draw
Logit

Dark Matter



Fixed Signal Region
RACNN-Classiflows (Finetuned)
CNN-Flows w/ BG (Scratch)
CNN-Flows w/o BG (Scratch)
Mean Negative Log Likelihood
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
0
5000
10000
15000
20000
25000
Data Size

Dark Matter


RACNN-Classiflows-Δ (ens)
Bin mean
Significance, Z
10^{1}
10^{0}
10^{-1}
10^{-2}
−2.0
−1.5
−1.0
−0.5
0.0
Entropy, H
Bin mean
0
5
10
15
20
25
Logit, Λ
10^{3}
10^{2}
10^{1}
10^{0}
Count

Fine-tuning

Infer when background only, weird? Finetune away!

- Tried it, but too easily overtrained
- More reliable when training with backgrounf only scenarios currently

Classifier can be Neural Likelihood Ratio Estimation?

- Finetune it to be one
- Too inefficient if you have large θ space.

Finetuning to different signal model with same signature?

- Possible too