Recent Advances in Resonant Anomaly Detection

Ranit Das

Based on arXiv:2410.20537 and arXiv:2312.11629

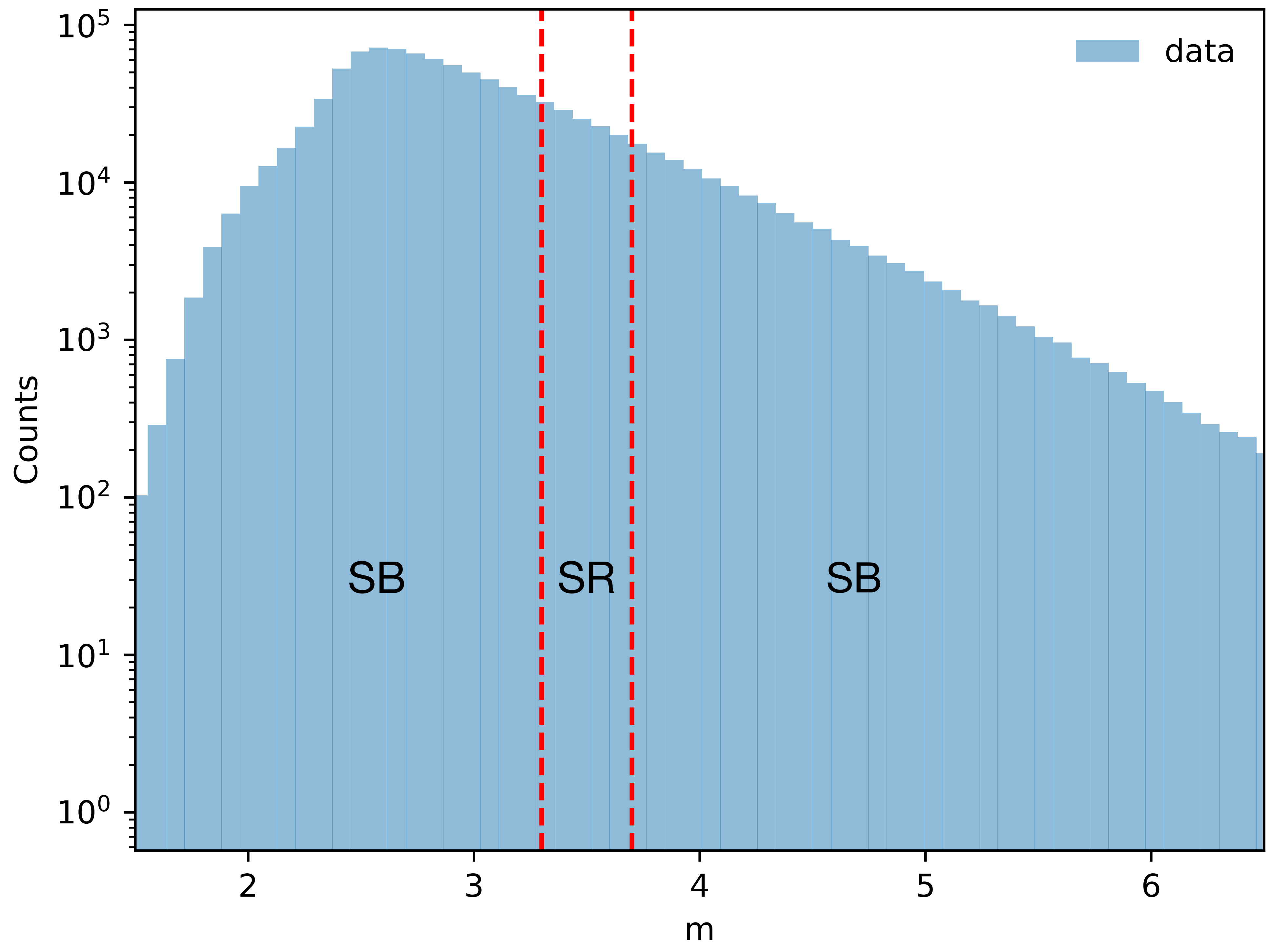
RUTGERS

AD4HEP-2025

06-17-2025

Data Driven Resonant Anomaly Detection with background interpolation

Key Steps:

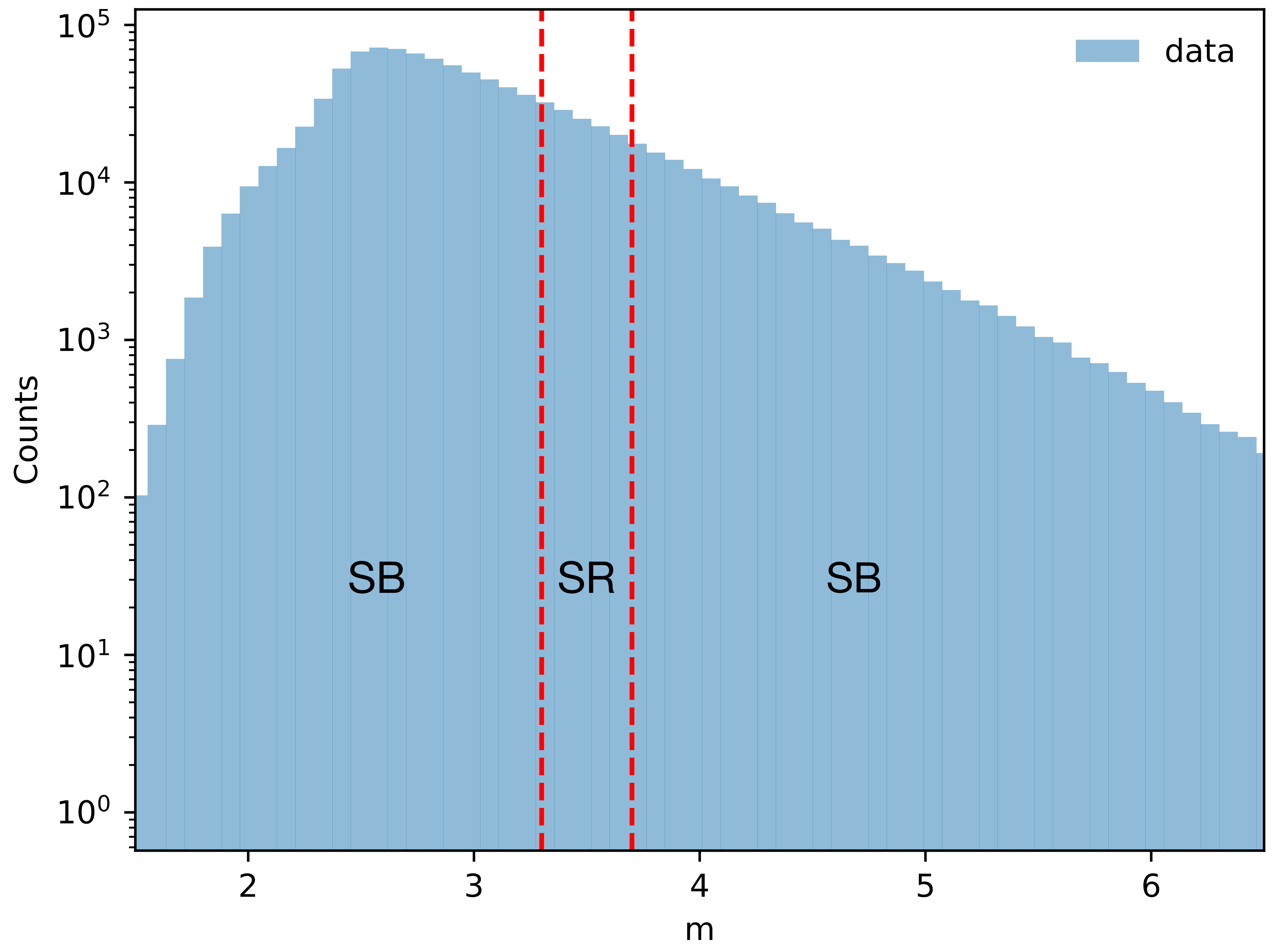


ANODE: arXiv:2001.04990v2
CATHODE: arXiv:2109.00546v3
CURTAINS: arXiv:2203.09470v3

Data Driven Resonant Anomaly Detection with background interpolation

Key Steps:

- Define different Signal Regions(SR) and Side-Band Regions(SB) using a resonant feature ***m***.



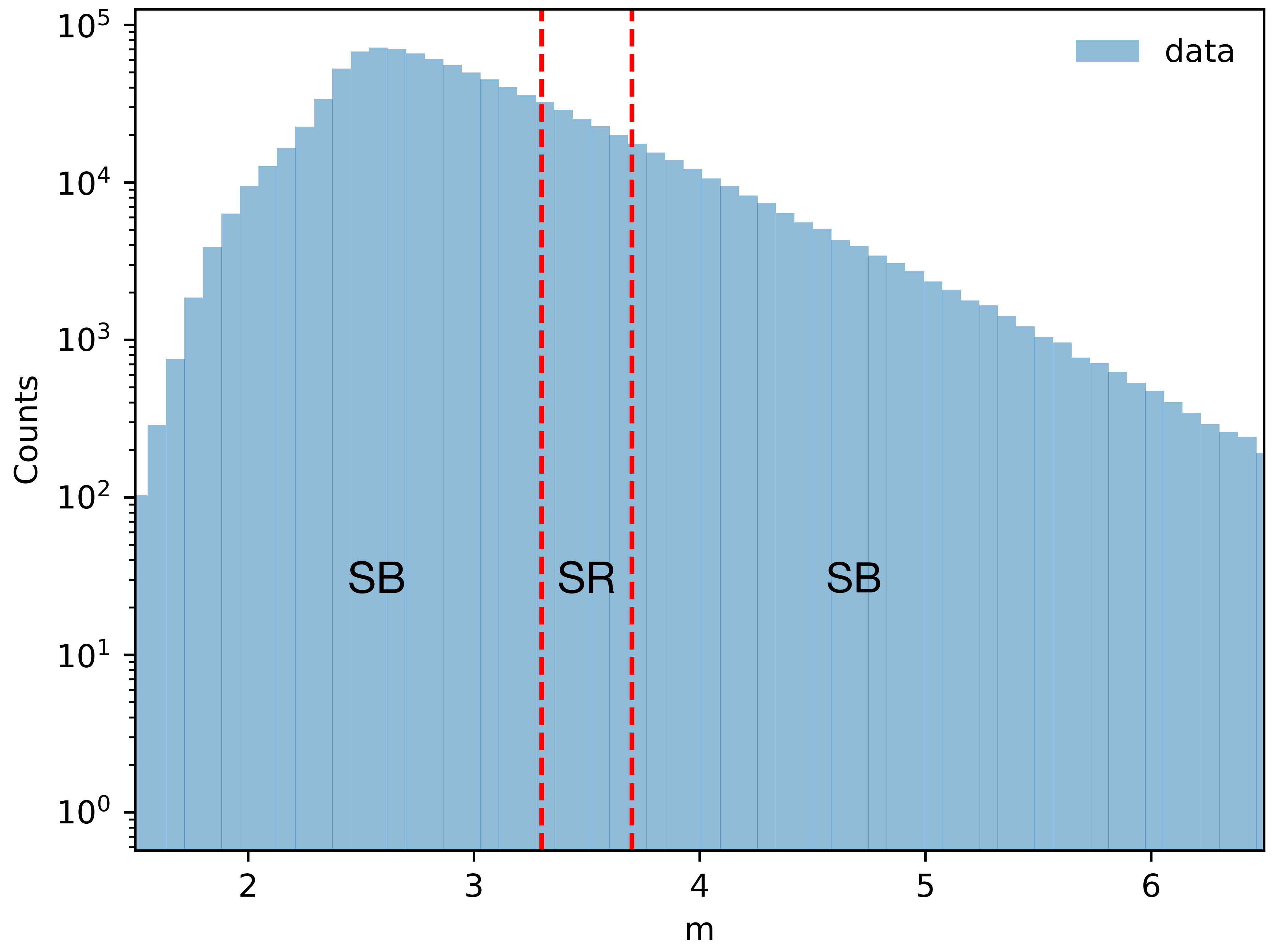
ANODE: arXiv:2001.04990v2
CATHODE: arXiv:2109.00546v3
CURTAINS: arXiv:2203.09470v3

Data Driven Resonant Anomaly Detection with background interpolation

Key Steps:

- Define different Signal Regions(SR) and Side-Band Regions(SB) using a resonant feature ***m***.

- For each SR, generate a background template from SB and interpolated into SR.

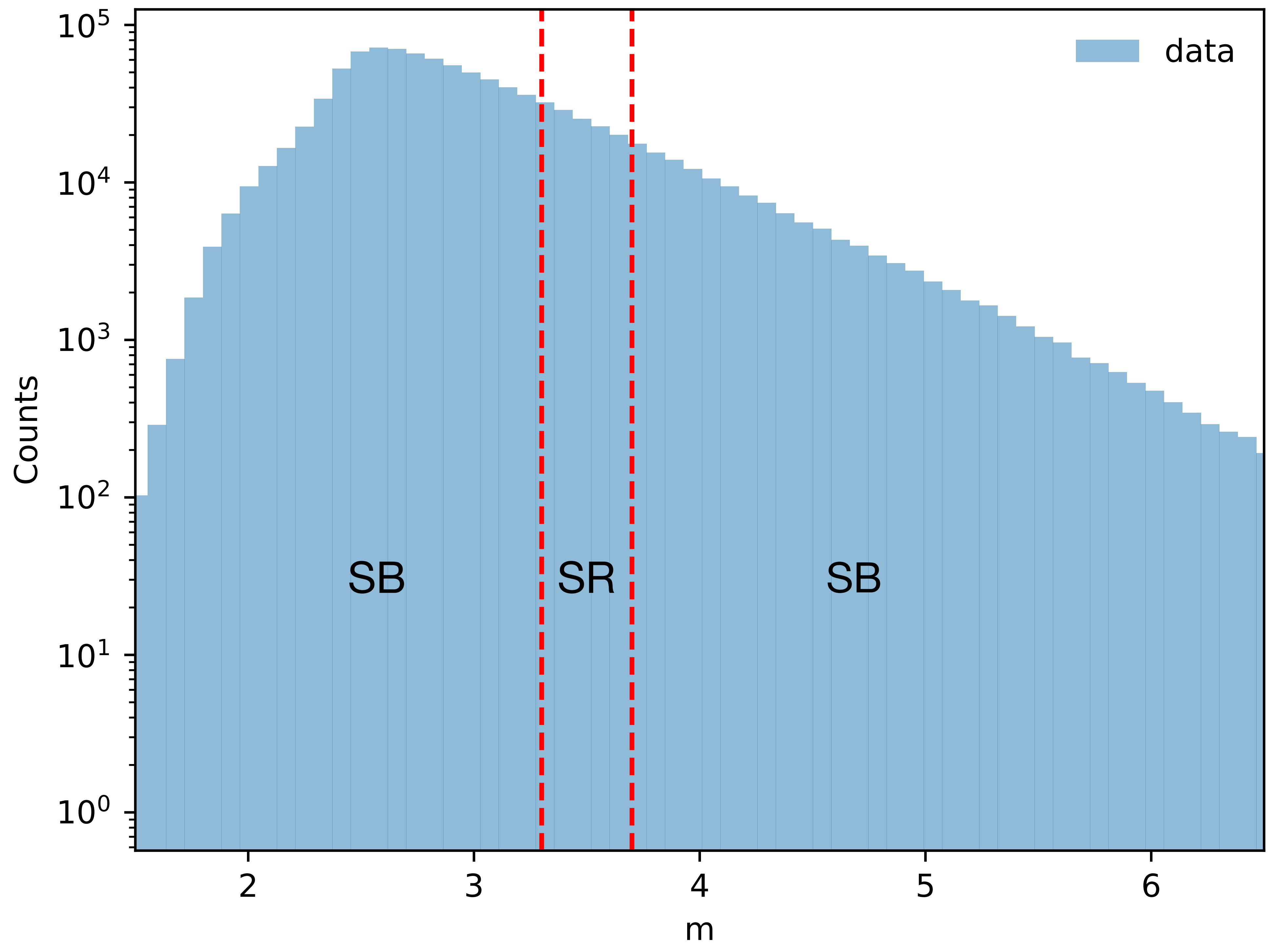


ANODE: arXiv:2001.04990v2
CATHODE: arXiv:2109.00546v3
CURTAINS: arXiv:2203.09470v3

Data Driven Resonant Anomaly Detection with background interpolation

Key Steps:

- Define different Signal Regions(SR) and Side-Band Regions(SB) using a resonant feature ***m***.
- For each SR, generate a background template from SB and interpolated into SR.
- Distinguish between data and background template using classifier (like CATHODE), or density estimators (like ANODE).



ANODE: arXiv:2001.04990v2
CATHODE: arXiv:2109.00546v3
CURTAINS: arXiv:2203.09470v3

Data Driven Resonant Anomaly Detection with background interpolation

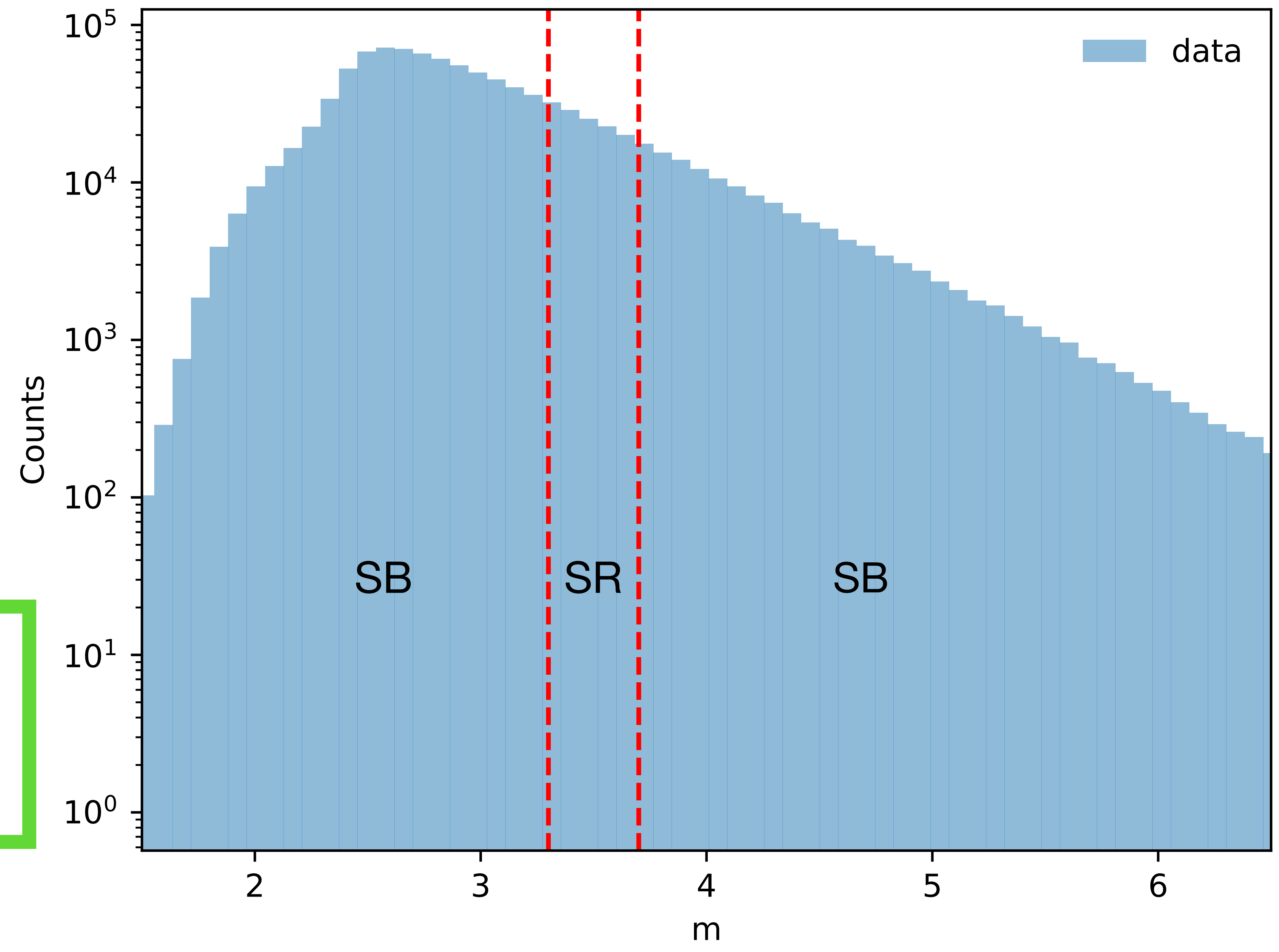
Key Steps:

- Define different Signal Regions(SR) and Side-Band Regions(SB) using a resonant feature ***m***.

SIGMA (second half of my talk)

- For each SR, generate a background template from SB and interpolated into SR.

- Distinguish between data and background template using classifier (like CATHODE), or density estimators (like ANODE).



ANODE: arXiv:2001.04990v2
CATHODE: arXiv:2109.00546v3
CURTAINS: arXiv:2203.09470v3

Data Driven Resonant Anomaly Detection with background interpolation

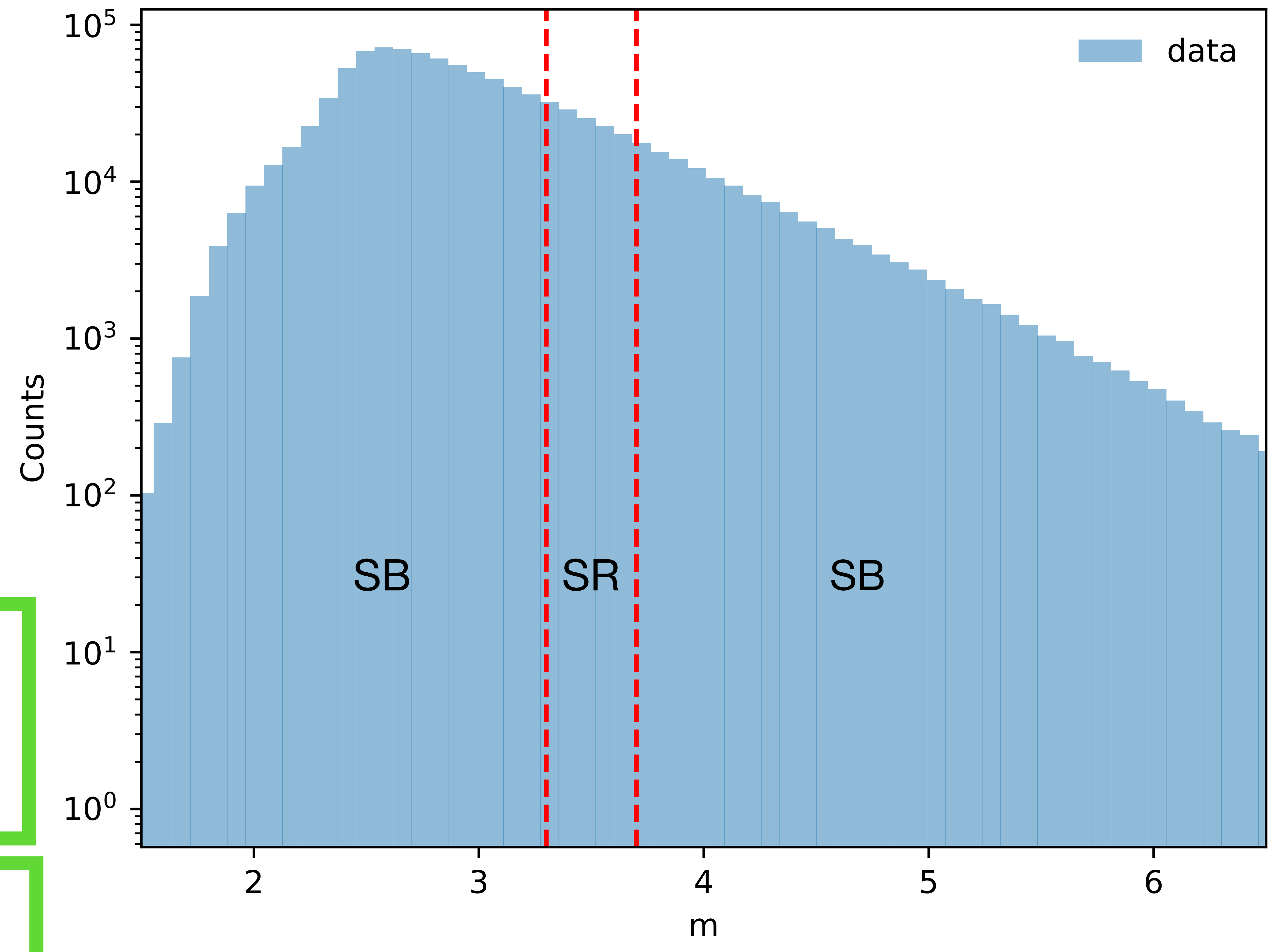
Key Steps:

- Define different Signal Regions(SR) and Side-Band Regions(SB) using a resonant feature ***m***.

SIGMA (second half of my talk)

- For each SR, generate a background template from SB and interpolated into SR.

- Distinguish between data and background template using classifier (like CATHODE), or density estimators (like ANODE).



ANODE: arXiv:2001.04990v2
CATHODE: arXiv:2109.00546v3
CURTAINS: arXiv:2203.09470v3

R-ANODE (first half of my talk!)

Previous methods

Previous methods

Given a background template, construct the NP classifier $R(x)$:

$$R(x) = \frac{P_{data}(x)}{P_B(x)}$$

Previous methods

Given a background template, construct the NP classifier $R(x)$:

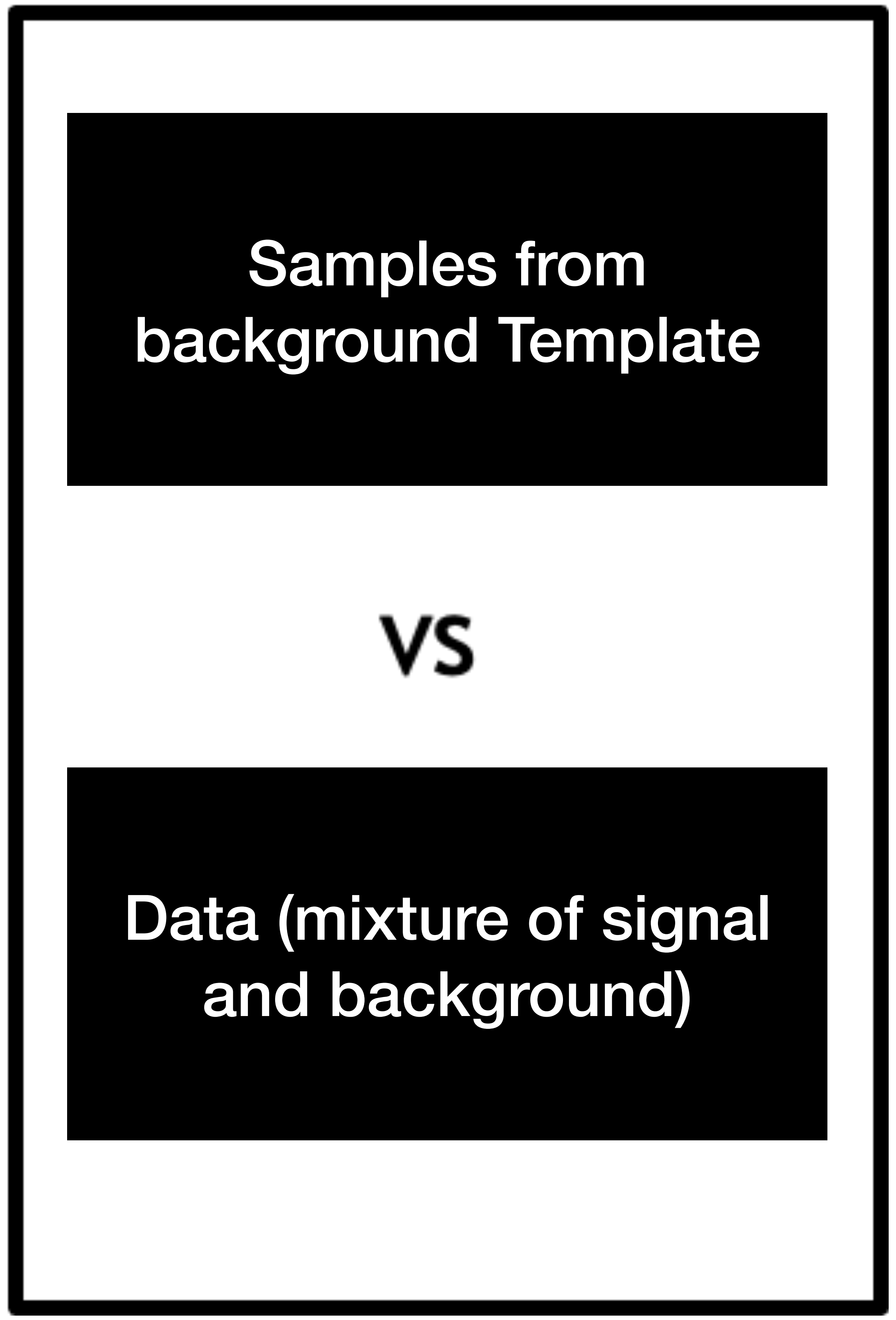
$$R(x) = \frac{P_{data}(x)}{P_B(x)}$$

Classifier based methods:

- CATHODE (arXiv:2109.00546v3)
- CWOLA (arXiv:1902.02634v2)
- CATHODE-BDT (arXiv:2309.13111)

.... etc

Classifier



Previous methods

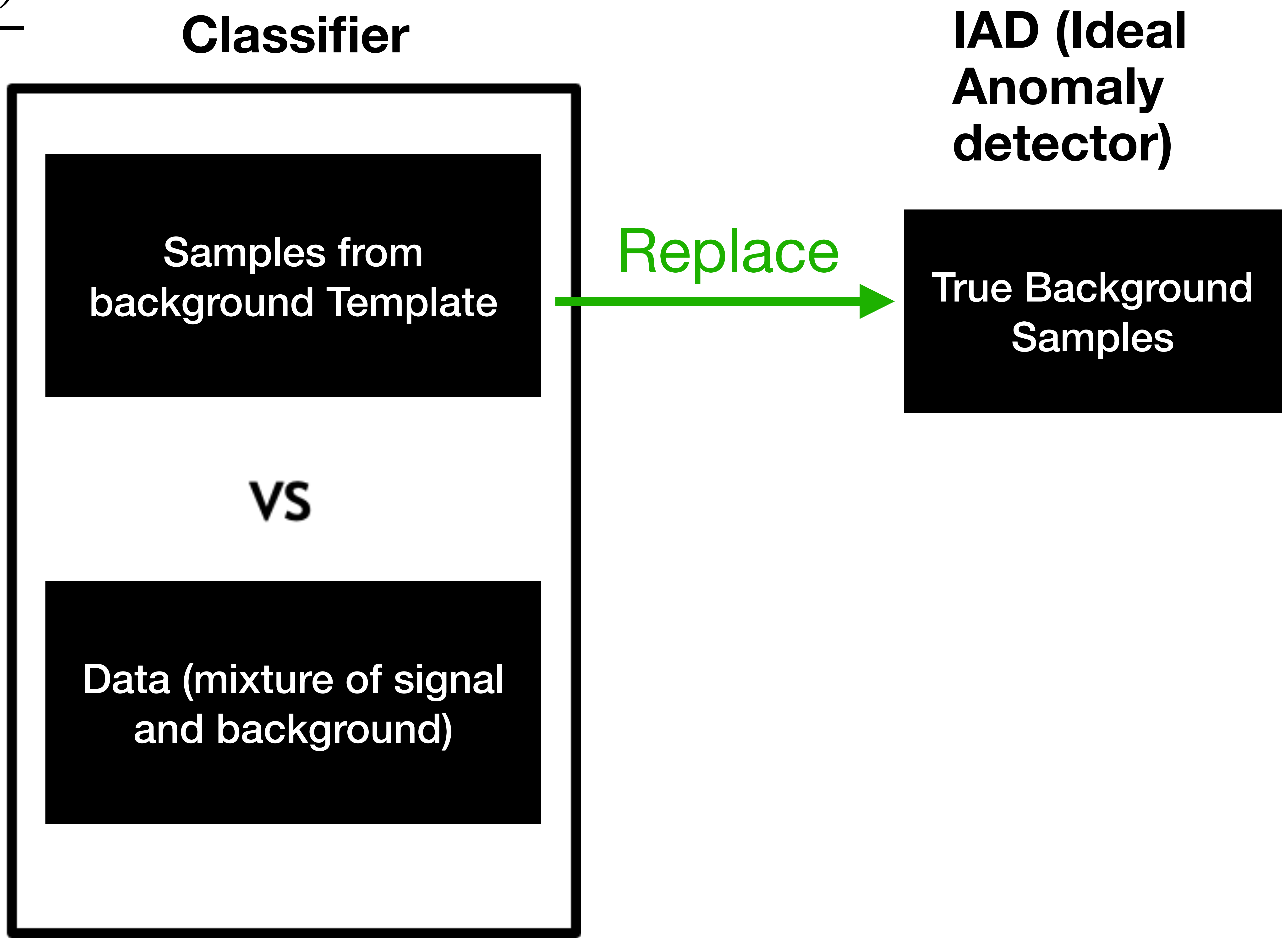
Given a background template, construct the NP classifier $R(x)$:

$$R(x) = \frac{P_{data}(x)}{P_B(x)}$$

Classifier based methods:

- CATHODE (arXiv:2109.00546v3)
- CWOLA (arXiv:1902.02634v2)
- CATHODE-BDT (arXiv:2309.13111)

…….. etc



Previous methods

Given a background template, construct the NP classifier $R(x)$:

$$R(x) = \frac{P_{data}(x)}{P_B(x)}$$

Classifier based methods:

- CATHODE (arXiv:2109.00546v3)
- CWOLA (arXiv:1902.02634v2)
- CATHODE-BDT (arXiv:2309.13111)

.... etc

Classifier

Samples from background Template

VS

Data (mixture of signal and background)

Replace

IAD (Ideal Anomaly detector)

True Background Samples

Upper limit for classifier based approaches !

(Useful for proof of concept studies)

Previous methods

Given a background template, construct the NP classifier $R(x)$:

Density estimation based methods:

- ANODE (arXiv:2001.04990v2)
- R-ANODE (arXiv:2312.11629)

$$R(x) = \frac{\boxed{P_{data}(x)}}{P_B(x)}$$

Estimate using a density estimator!

Previous methods

Given a background template, construct the NP classifier $R(x)$:

Density estimation based methods:

- ANODE (arXiv:2001.04990v2)
- R-ANODE (arXiv:2312.11629)

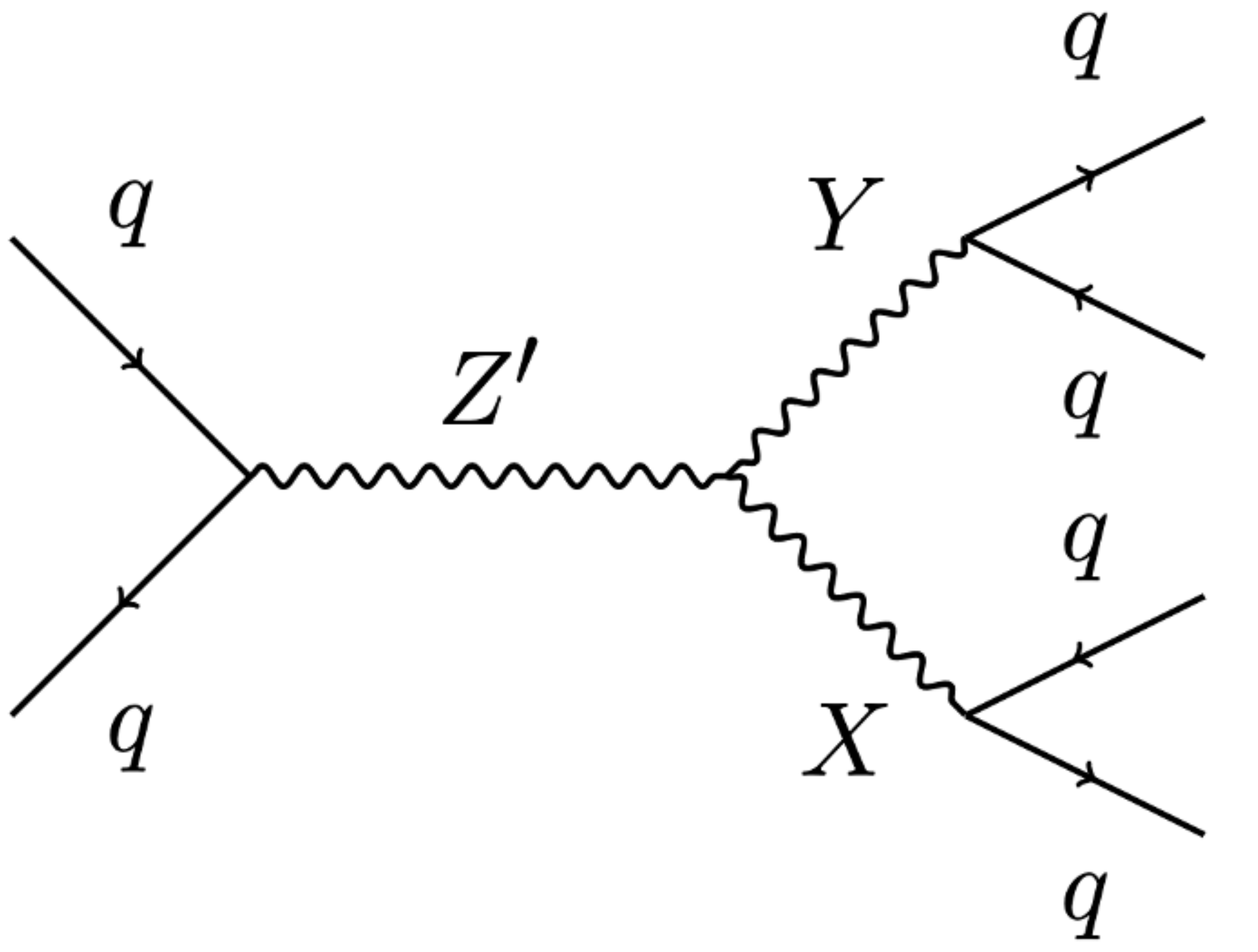
this talk!

$$R(x) = \frac{\boxed{P_{data}(x)}}{P_B(x)}$$

Estimate using a density estimator!

Dataset: LHCO dataset

- Data: 1M QCD di-jet events as background and different amounts of signal events.
- The resonant variable is m_{JJ}, and the features x are $[m_{J_1}, m_{J_2} - m_{J_1}, \tau_{21}^{J_1}, \tau_{21}^{J_2}]$
- The SR : $3.3TeV < m_{JJ} < 3.7TeV$.



The LHC Olympics 2020: A Community Challenge for Anomaly Detection in High Energy Physics : arXiv:2101.08320

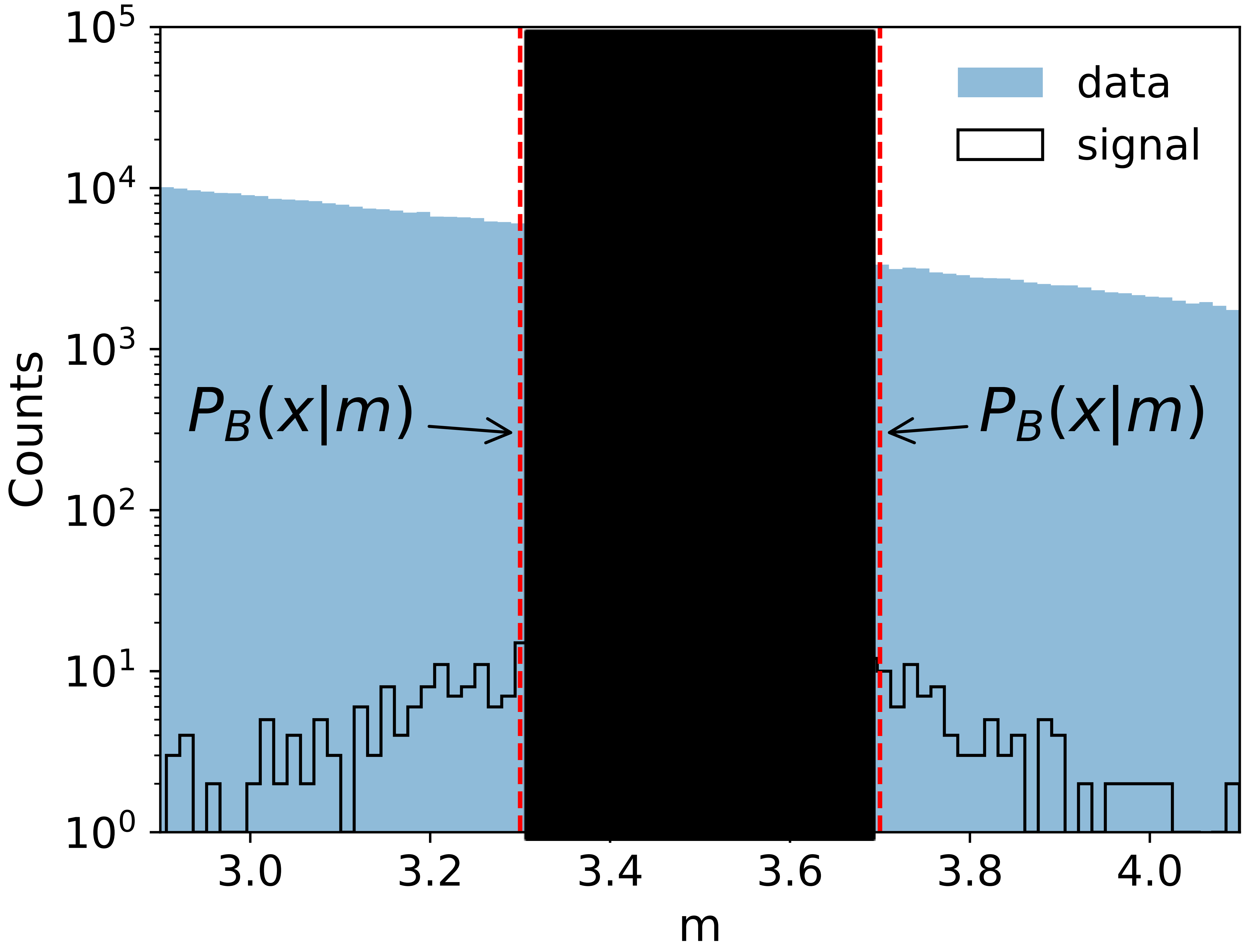
R-ANODE: Residual ANODE

Based on [arXiv:2312.11629](https://arxiv.org/abs/2312.11629)

Ranit Das, Gregor Kasieczka, and David Shih

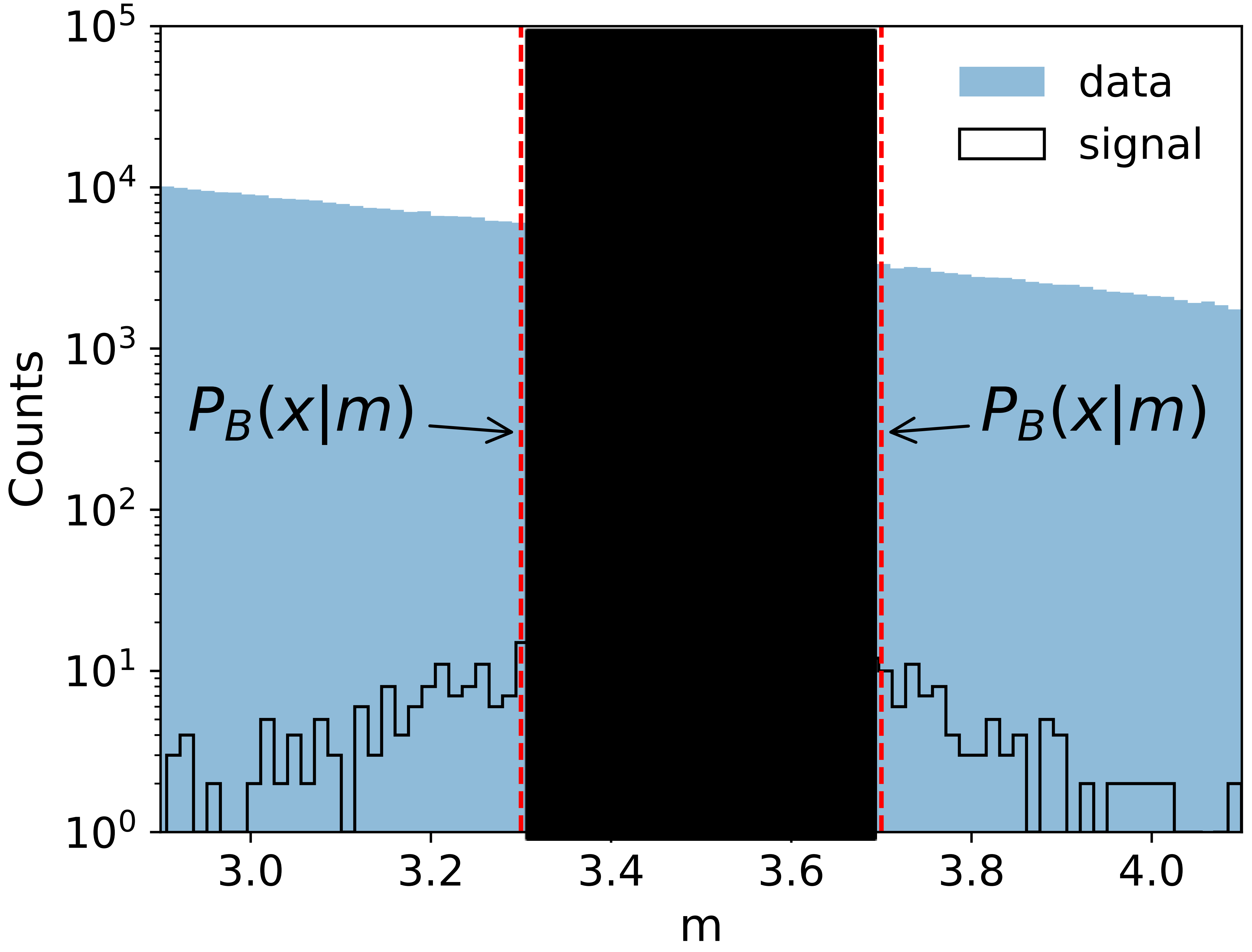
ANODE/CATHODE

Generate background template in SR:



ANODE/CATHODE

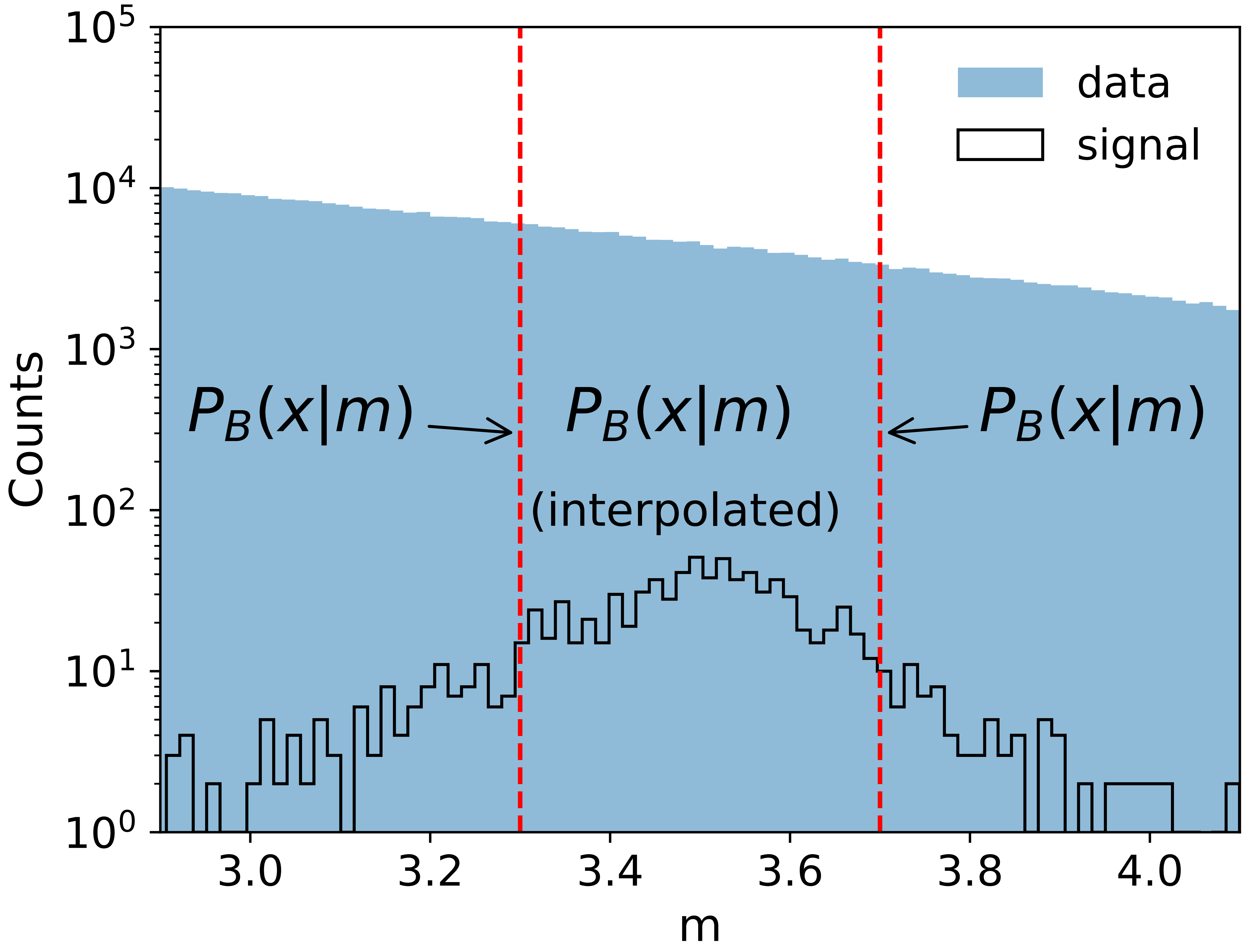
Generate background template in SR:



- A conditional density estimator is trained to learn $P_B(x \,|\, m \in SB)$ in the side-bands (SB).

ANODE/CATHODE

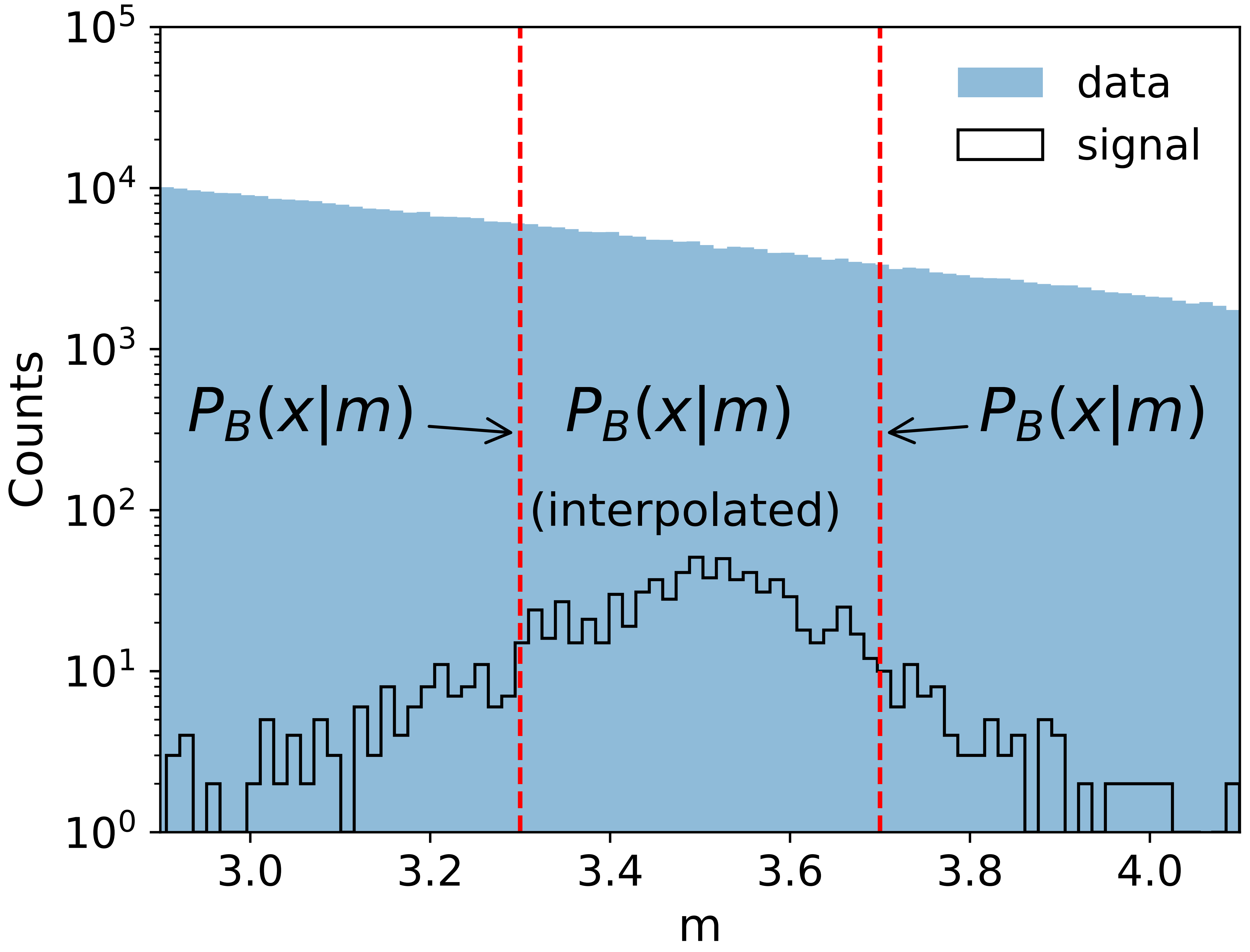
Generate background template in SR:



- A conditional density estimator is trained to learn $P_B(x\,|\,m \in SB)$ in the side-bands (SB).

ANODE/CATHODE

Generate background template in SR:



- A conditional density estimator is trained to learn $P_B(x \mid m \in SB)$ in the side-bands (SB).
- The learned $P_B(x \mid m)$ is used to interpolate into the SR.

R-ANODE

In SR:

$$P_{data}(x, m) = w * P_S(x, m) + (1 - w) * P_B(x, m)$$

R-ANODE

In SR:

$$P_{data}(x, m) = w * P_S(x, m) + (1 - w) * P_B(x, m)$$

(Interpolated Background model held fixed)

R-ANODE

In SR:

$$P_{data}(x, m) = w * P_S(x, m) + (1 - w) * P_B(x, m)$$

(Normalizing flow, learned from data)

(Interpolated Background model held fixed)

R-ANODE

In SR:

$$P_{data}(x,m) = \boxed{w} * P_S(x,m) + (1-w) * P_B(x,m)$$

(Normalizing flow, learned from data)

(Interpolated Background model held fixed)

Learn the signal fraction ***w*** from data

R-ANODE

In SR:

$$P_{data}(x, m) = \boxed{w} * P_S(x, m) + (1 - w) * P_B(x, m)$$

(Normalizing flow, learned from data)

(Interpolated Background model held fixed)

Learn the signal fraction ***w*** from data

R-ANODE (ideal): ***w*** fixed to the true ***w***-value (useful for proof of concept studies)

R-ANODE

In SR:

$$P_{data}(x, m) = \boxed{w} * P_S(x, m) + (1 - w) * P_B(x, m)$$

(Normalizing flow, learned from data)

(Interpolated Background model held fixed)

Learn the signal fraction ***w*** from data

$$R(x) = \frac{\boxed{P_{data}(x)}}{P_B(x)}$$

R-ANODE (ideal): ***w*** fixed to the true ***w***-value (useful for proof of concept studies)

R-ANODE

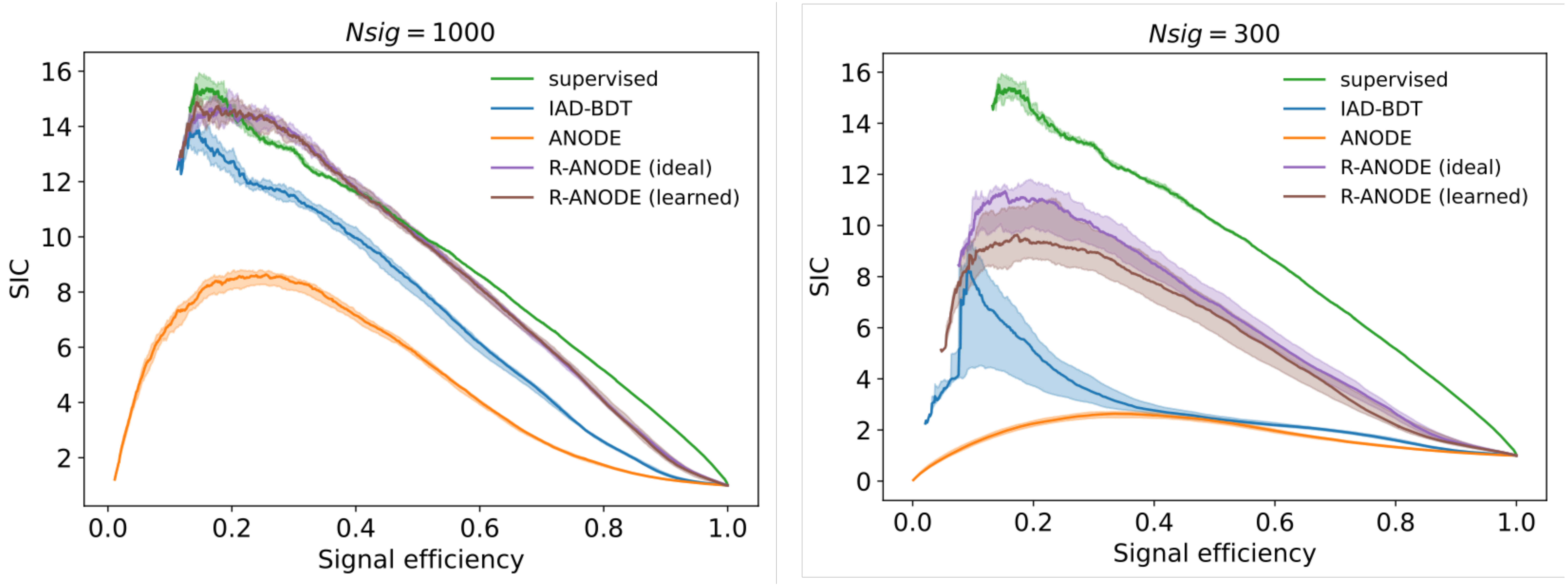
Loss:

$$\boxed{\text{Minimize :} \quad -\log(P_{data}(x,m))}$$

w.r.t parameters of $P_S(x,m)$ and ***w,*** while freezing the parameters of $P_B(x,m)$

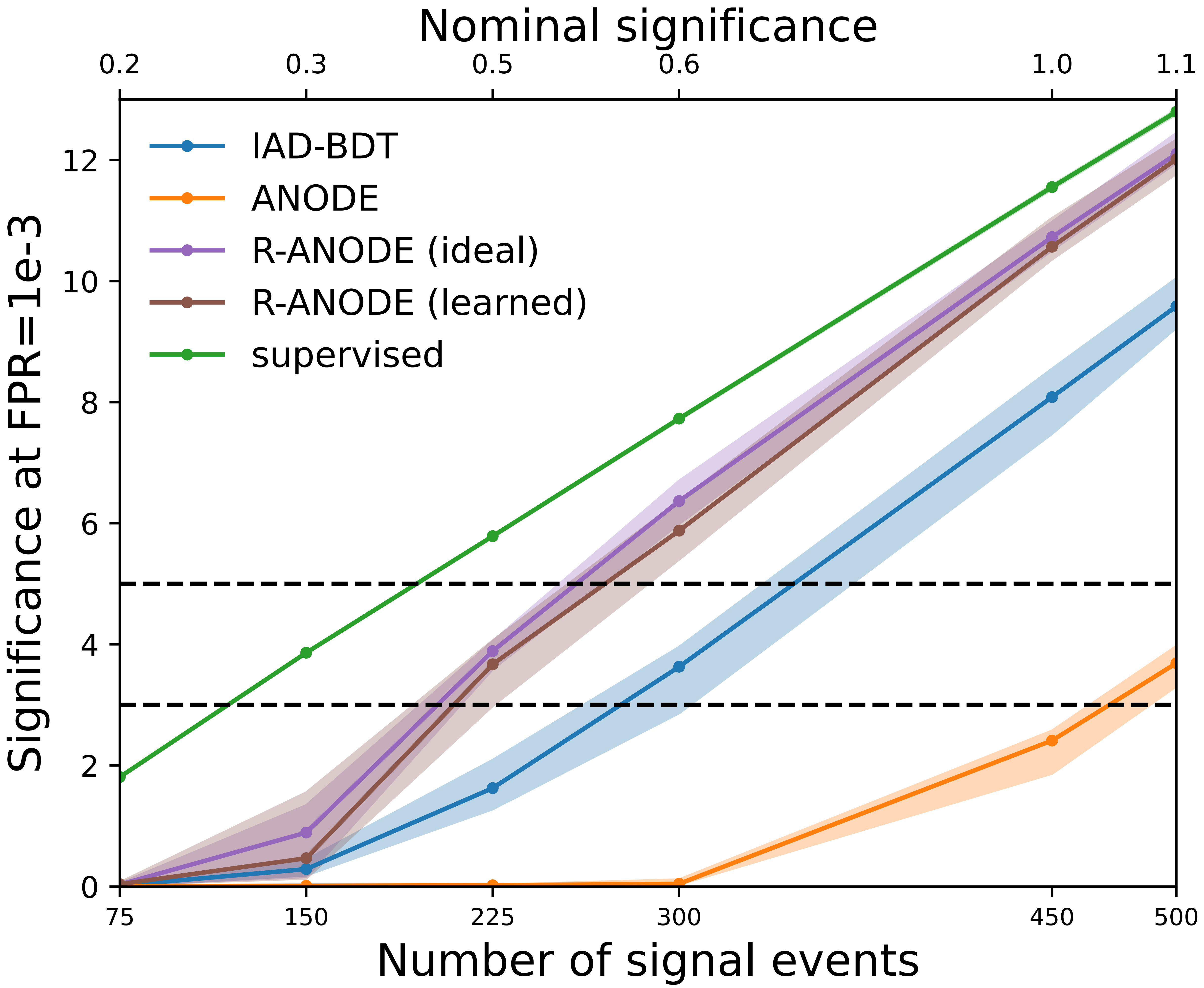
SIC Curves

$$SIC = TPR/\sqrt{FPR}$$



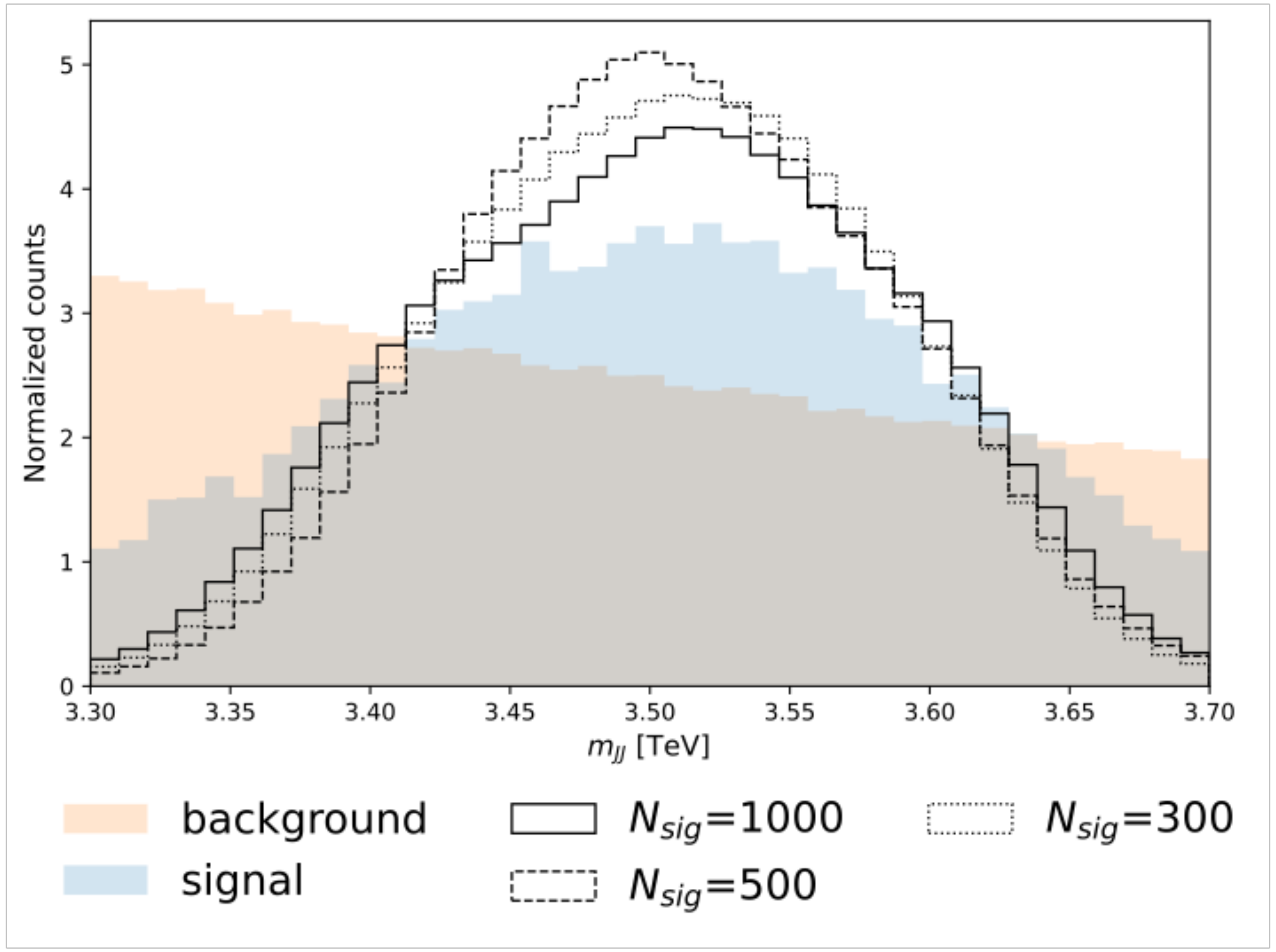
R-ANODE improves ANODE and also gives better SIC Curves than the idealized-AD (IAD) (classifier based approaches!)

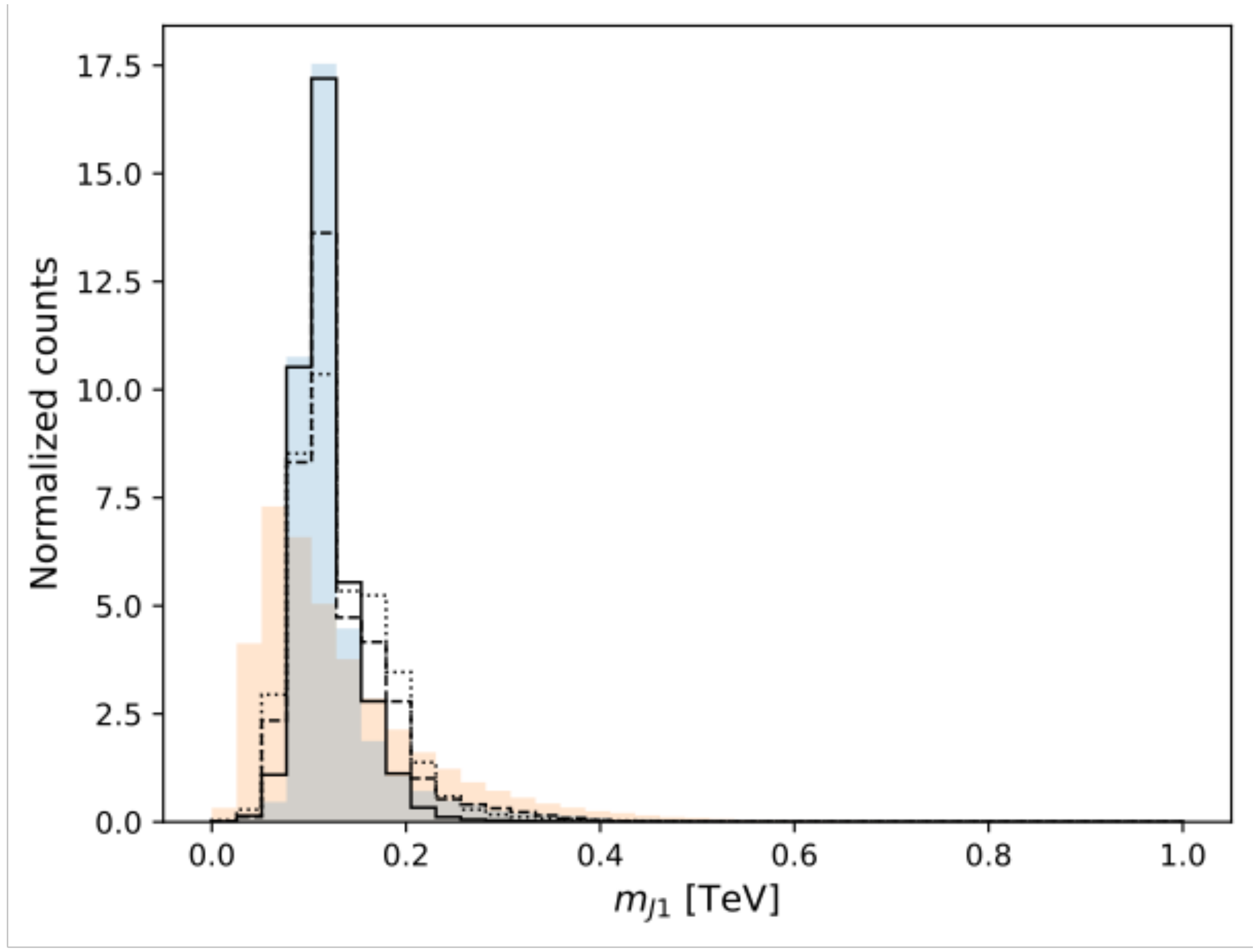
Nsig vs Significance



Significance = $SIC * \frac{S}{\sqrt{B}}$

Samples from $P_S(x, m)$

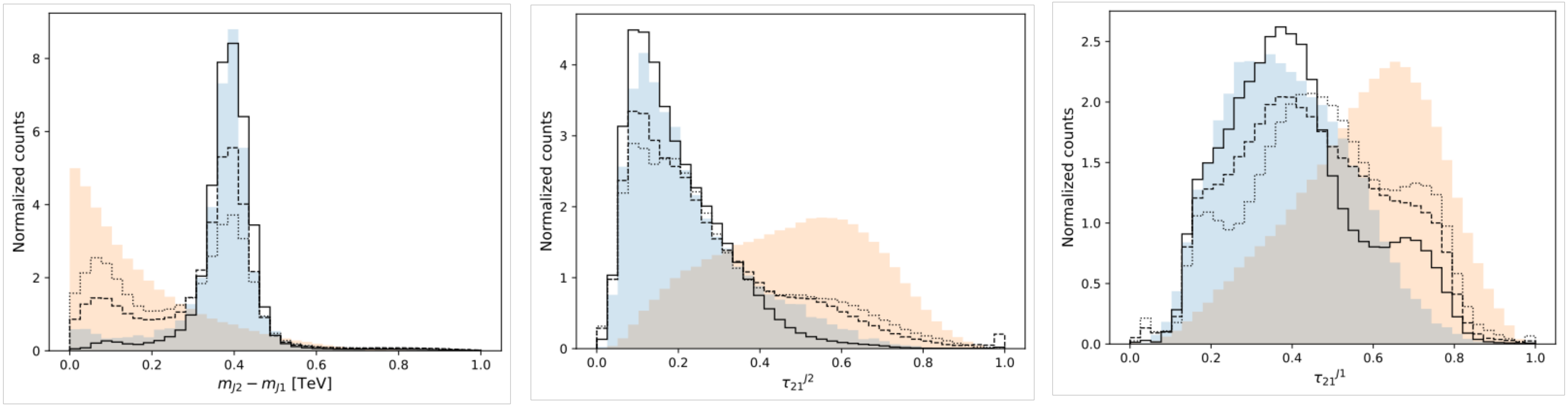




- Directly learning the signal distributions $P_S(x, m)$ leads to a more interpretable method.

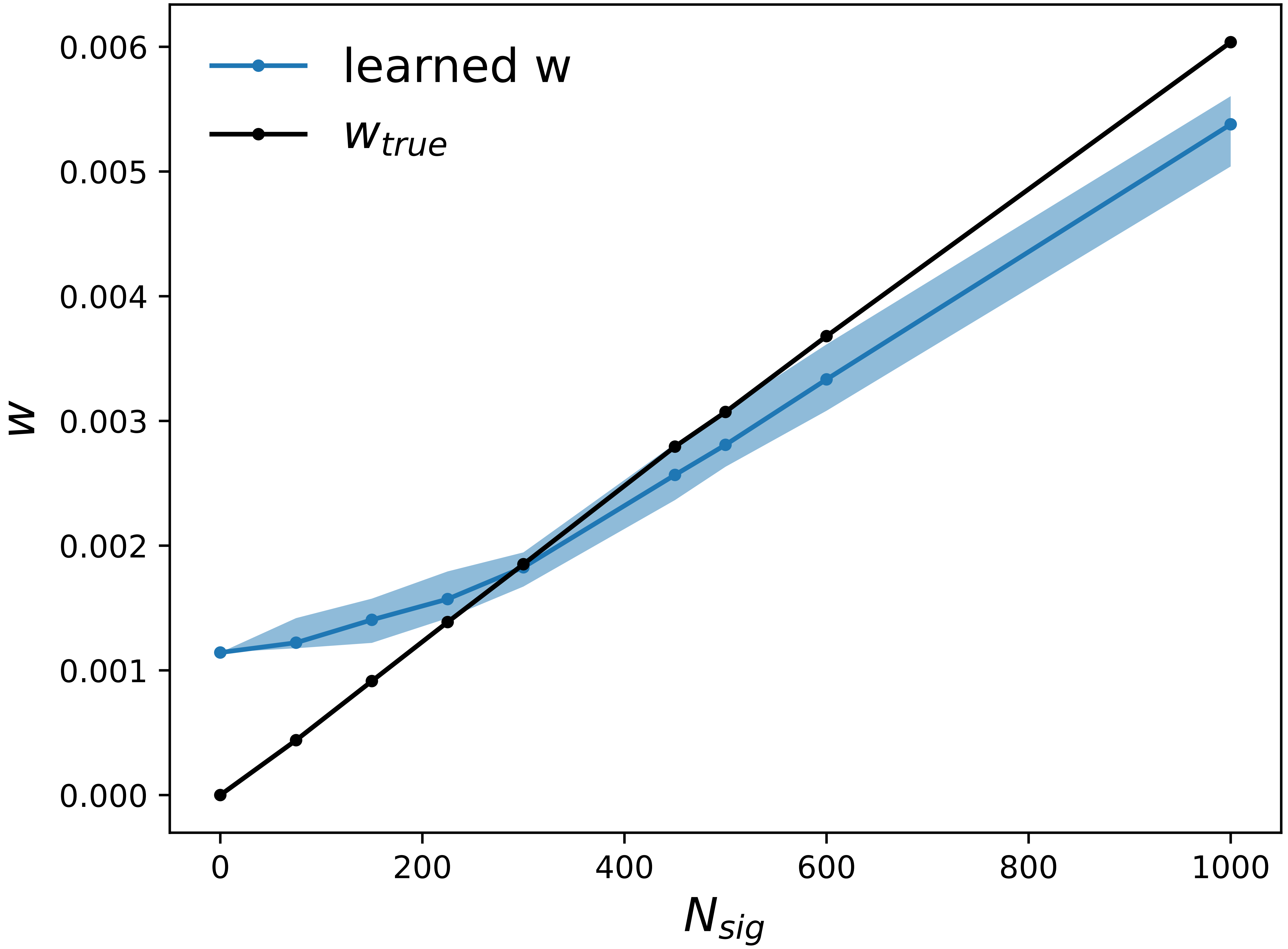
- This could give us information about the signal: eg: mass of subjet, Pronginess of subjet.

Samples from $P_S(x, m)$



- Directly learning the signal distributions $P_S(x, m)$ leads to a more interpretable method.
- This could give us information about the signal: eg: mass of subjet, Pronginess of subjet.

Learned *w*



Learned ***w*** is very close to the true ***w*** values

Summary

Summary

- R-ANODE performs better than classifier-based approaches like CATHODE.

Summary

- R-ANODE performs better than classifier-based approaches like CATHODE.
- Additionally, R-ANODE learns the signal model and the signal fraction directly from data.

Further recent work

Further recent work

09:00	**Foundation Models for AD**	*Vinny Mikuni*	*Nevis Science Center, Columbia University, Nevis Laboratories*	09:00 - 09:18
	AD Interpretation & Phenomenology	*Anna Hallin*	*Nevis Science Center, Columbia University, Nevis Laboratories*	09:18 - 09:36
	Incorporating Physical Priors into Weakly-Supervised Anomaly Detection	*Chi Lung Cheng Cheng*	*Nevis Science Center, Columbia University, Nevis Laboratories*	09:36 - 09:54
10:00	**Surrogate Simulation-based Inference (S2BI)**	*Runze Li*	*Nevis Science Center, Columbia University, Nevis Laboratories*	09:54 - 10:12
	From High Dimensions to Statistical Discovery: A Contrastive Learning Approach to Anomaly Detection	*Gaia Grosso*	*Nevis Science Center, Columbia University, Nevis Laboratories*	10:12 - 10:30

Further recent work

GBI (earlier talk by Runze!)

Confidence intervals on the signal fraction for PAWS and R-ANODE

09:00	**Foundation Models for AD** *Nevis Science Center, Columbia University, Nevis Laboratories*	*Vinny Mikuni* 09:00 - 09:18
	AD Interpretation & Phenomenology *Nevis Science Center, Columbia University, Nevis Laboratories*	*Anna Hallin* 09:18 - 09:36
	Incorporating Physical Priors into Weakly-Supervised Anomaly Detection *Nevis Science Center, Columbia University, Nevis Laboratories*	*Chi Lung Cheng Cheng* 09:36 - 09:54
10:00	**Surrogate Simulation-based Inference (S2BI)** *Nevis Science Center, Columbia University, Nevis Laboratories*	*Runze Li* 09:54 - 10:12
	From High Dimensions to Statistical Discovery: A Contrastive Learning Approach to Anomaly Detection *Nevis Science Center, Columbia University, Nevis Laboratories*	*Gaia Grosso* 10:12 - 10:30

SIGMA: Single Interpolated Generative Model for Anomalies

Based on arXiv:2410.20537

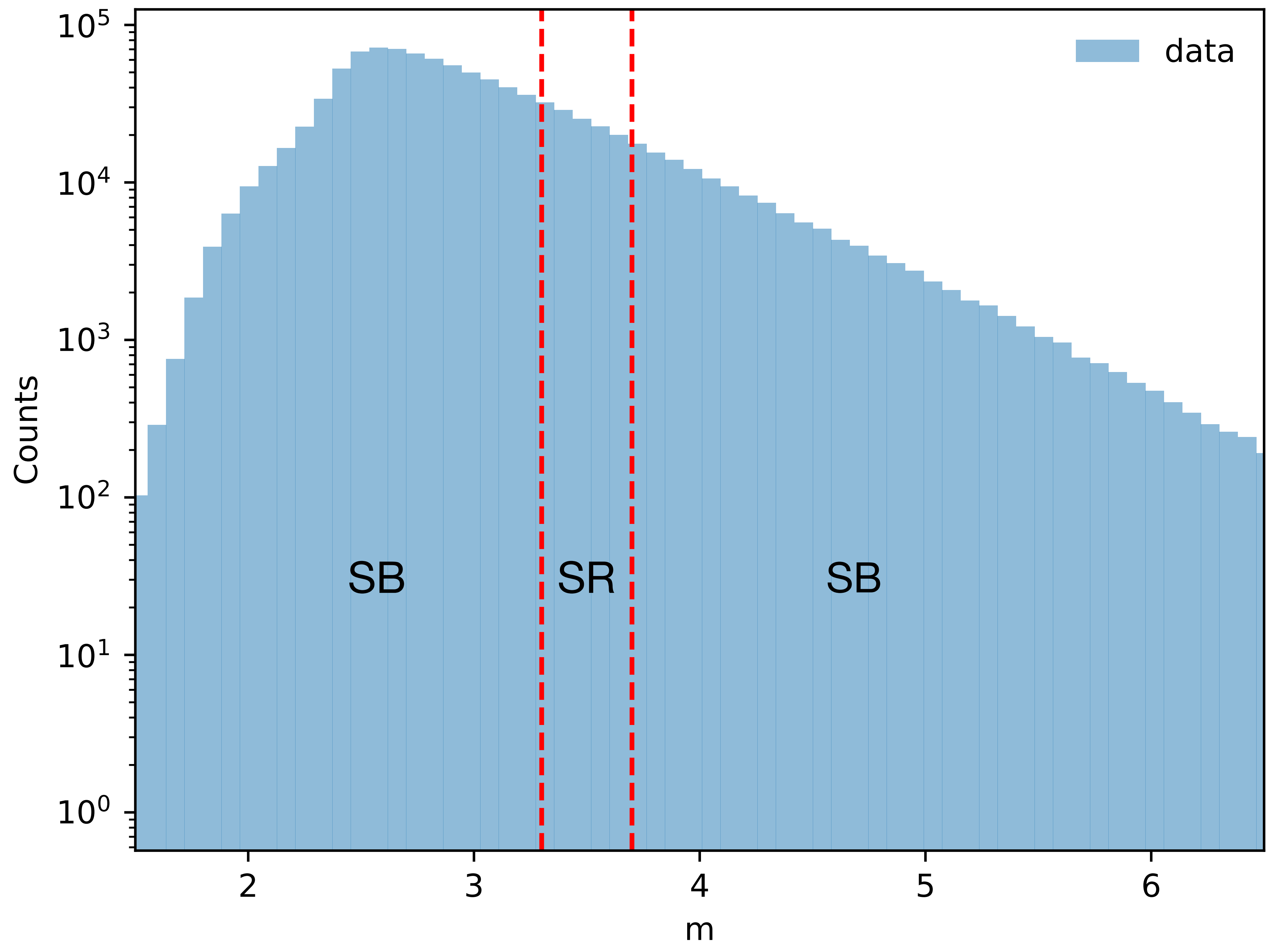
Ranit Das and David Shih

RUTGERS

Data Driven Resonant Anomaly Detection with background interpolation

Key Steps:

- Define different Signal Regions(SR) and Side-Band Regions(SB) using a resonant feature ***m***.
- For each SR, generate a background template from SB and interpolated into SR.
- Distinguish between data and background template using classifier (like CATHODE), or density estimators (like ANODE, R-ANODE).



ANODE: arXiv:2001.04990v2
CATHODE: arXiv:2109.00546v3
CURTAINS: arXiv:2203.09470v3
R-ANODE: arXiv:2312.11629

Data Driven Resonant Anomaly Detection with background interpolation

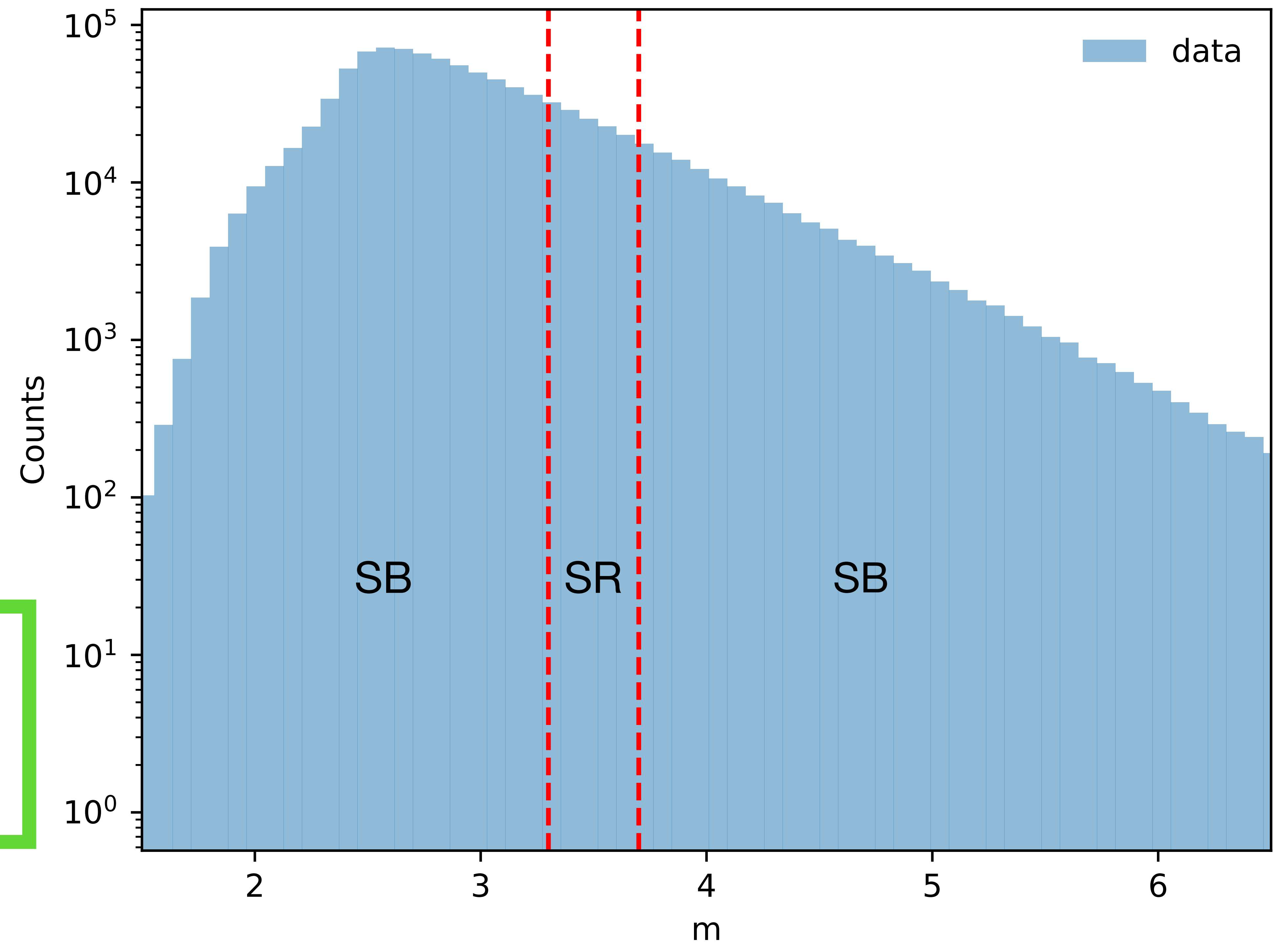
Key Steps:

- Define different Signal Regions(SR) and Side-Band Regions(SB) using a resonant feature ***m***.

SIGMA

- For each SR, generate a background template from SB and interpolated into SR.

- Distinguish between data and background template using classifier (like CATHODE), or density estimators (like ANODE, R-ANODE).



ANODE: arXiv:2001.04990v2
CATHODE: arXiv:2109.00546v3
CURTAINS: arXiv:2203.09470v3
R-ANODE: arXiv:2312.11629

Data Driven Resonant Anomaly Detection with background interpolation

Key Steps:

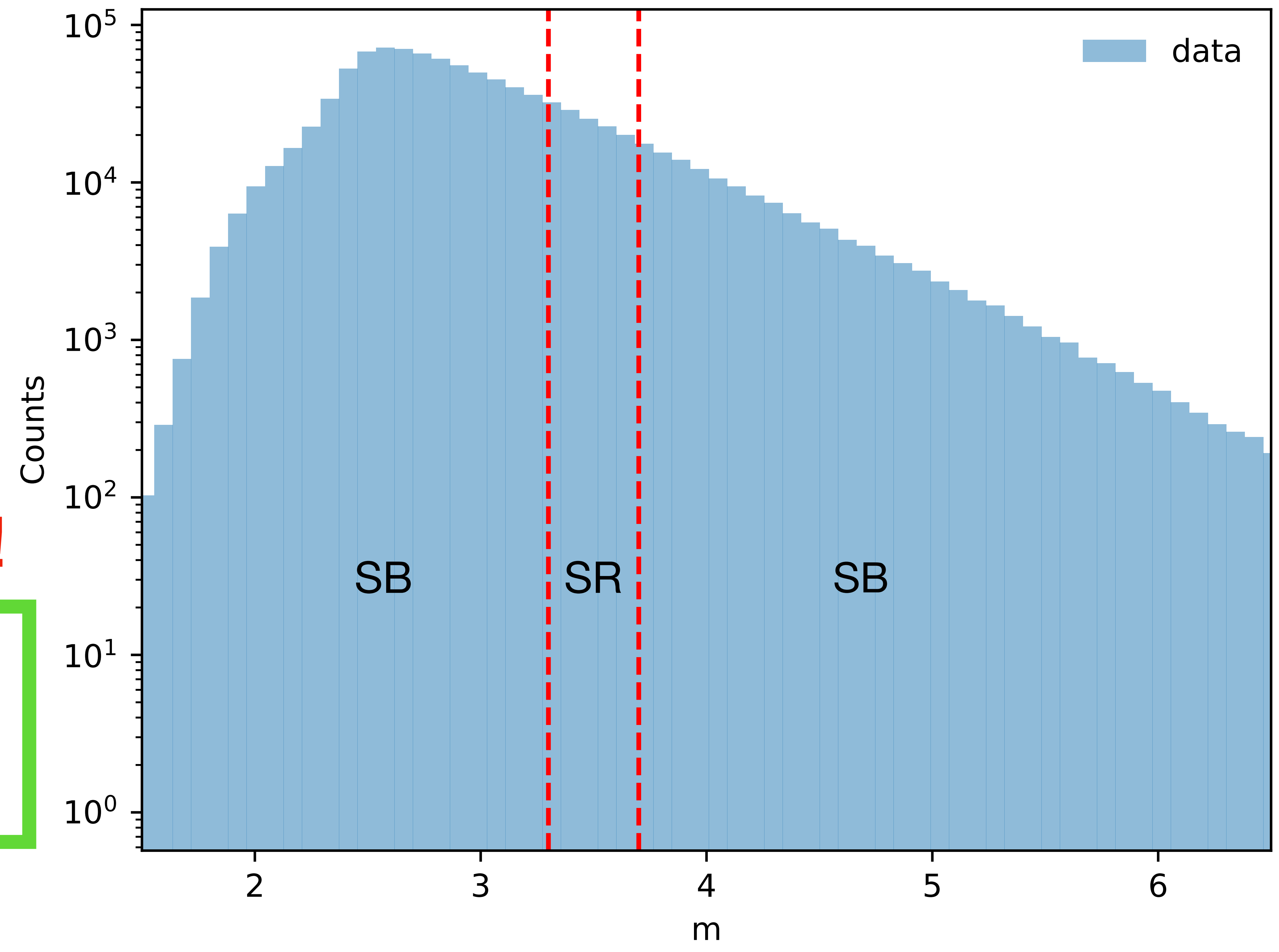
- Define different Signal Regions(SR) and Side-Band Regions(SB) using a resonant feature ***m***.

SIGMA

Problem: Computationally expensive!

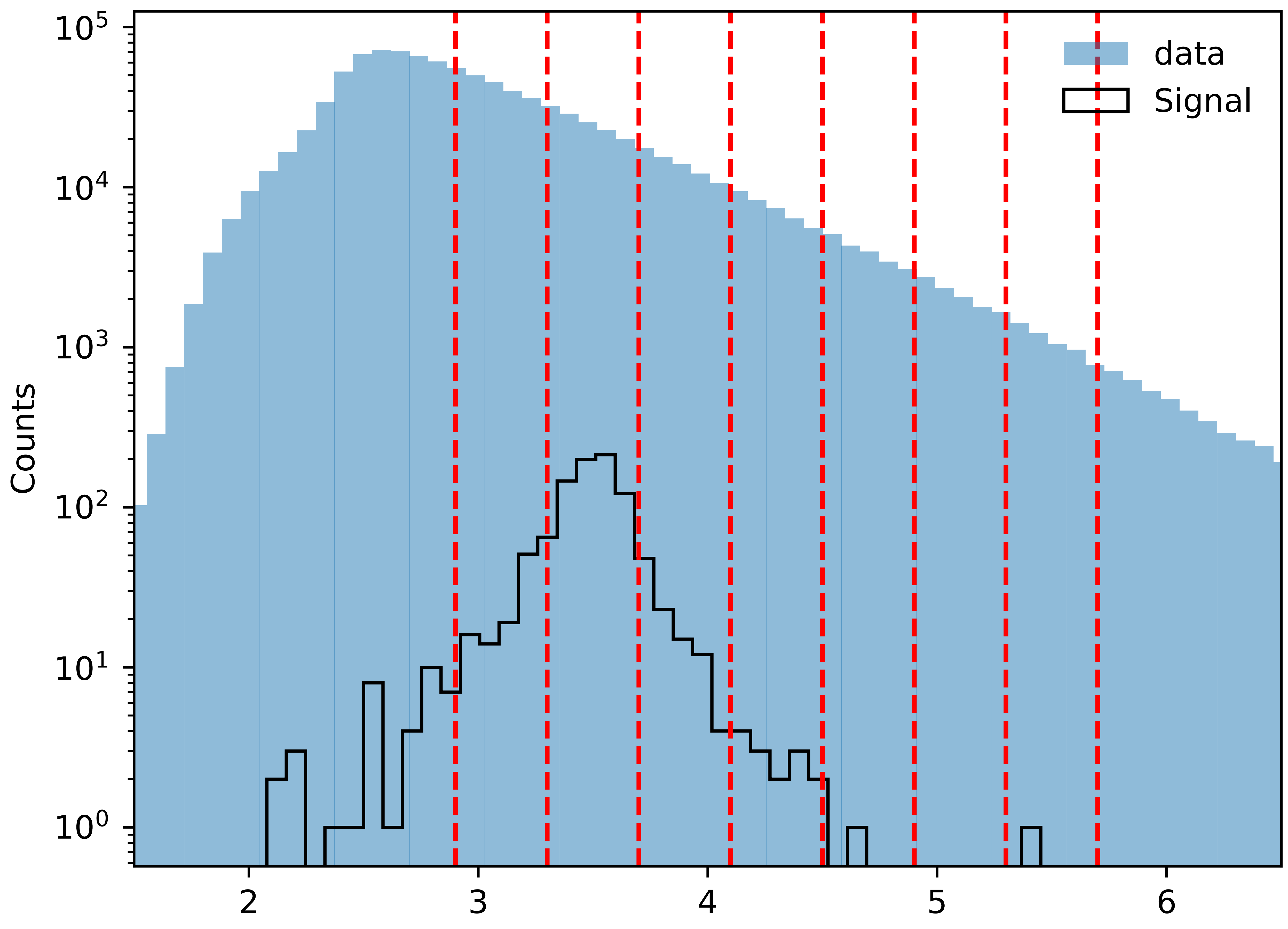
- For each SR, generate a background template from SB and interpolated into SR.

- Distinguish between data and background template using classifier (like CATHODE), or density estimators (like ANODE, R-ANODE).



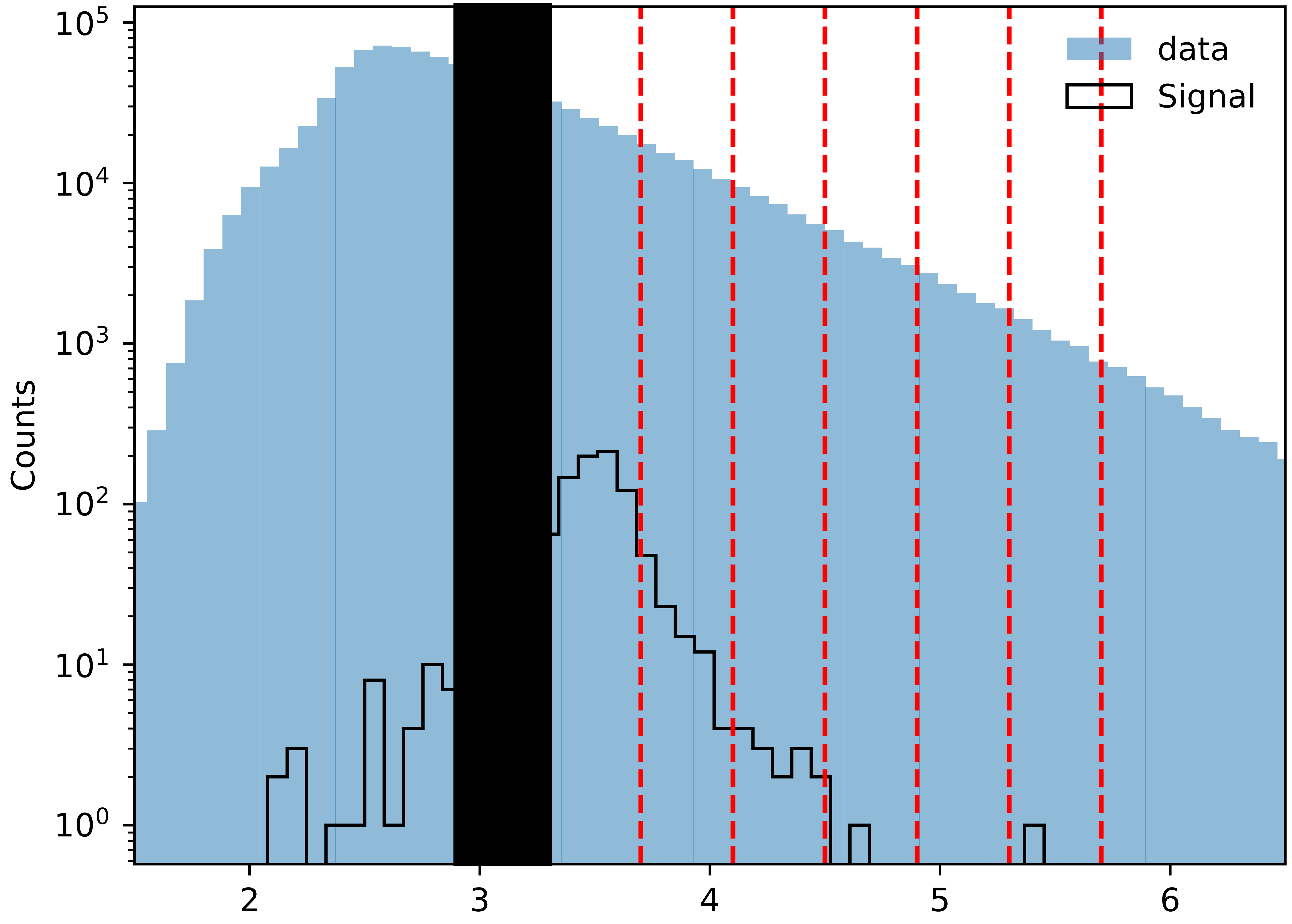
ANODE: arXiv:2001.04990v2
CATHODE: arXiv:2109.00546v3
CURTAINS: arXiv:2203.09470v3
R-ANODE: arXiv:2312.11629

Background Template generation is computationally expensive!



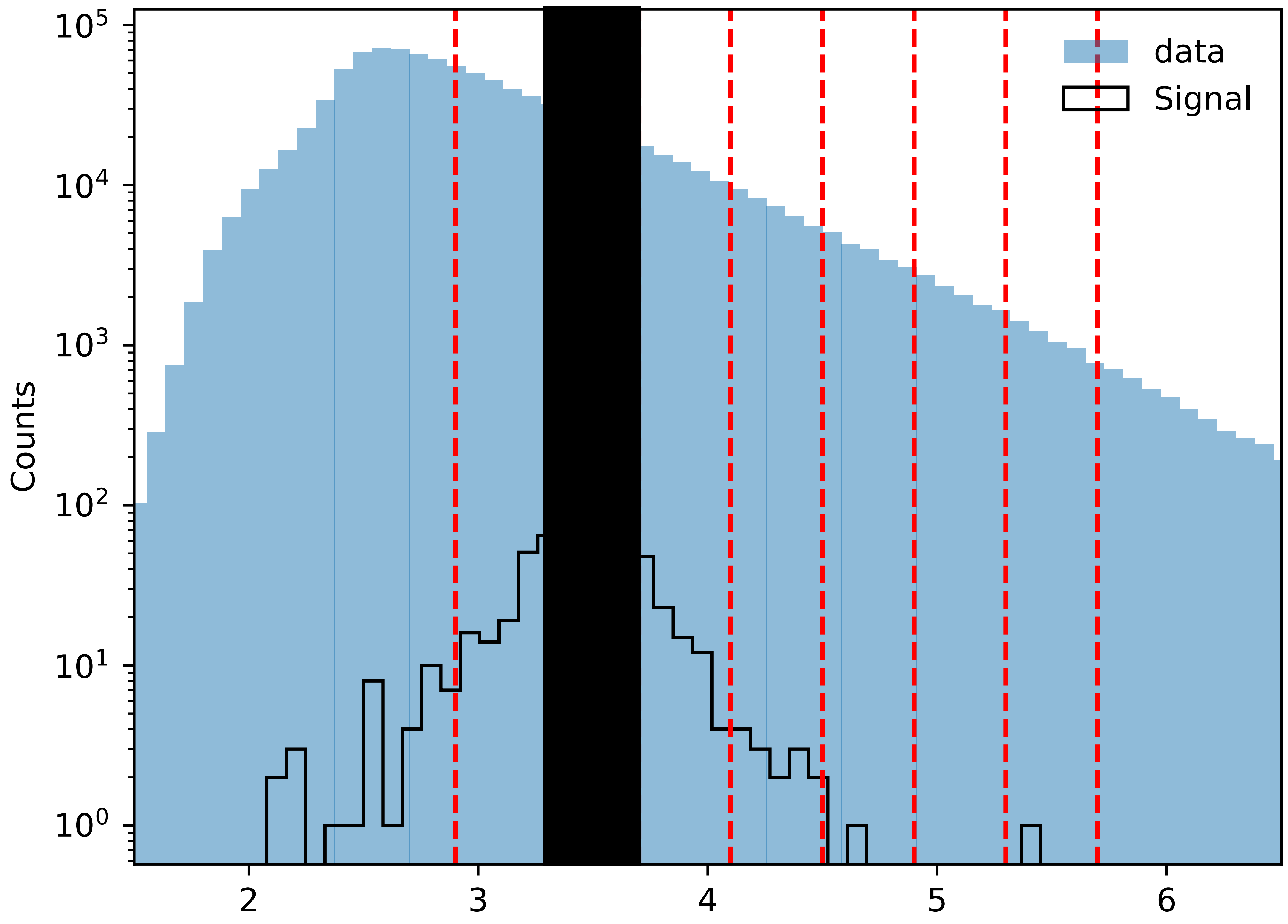
Background Template generation is computationally expensive!

- For each SR, a separate generative model is re-trained on almost the entire data, by masking out that SR.



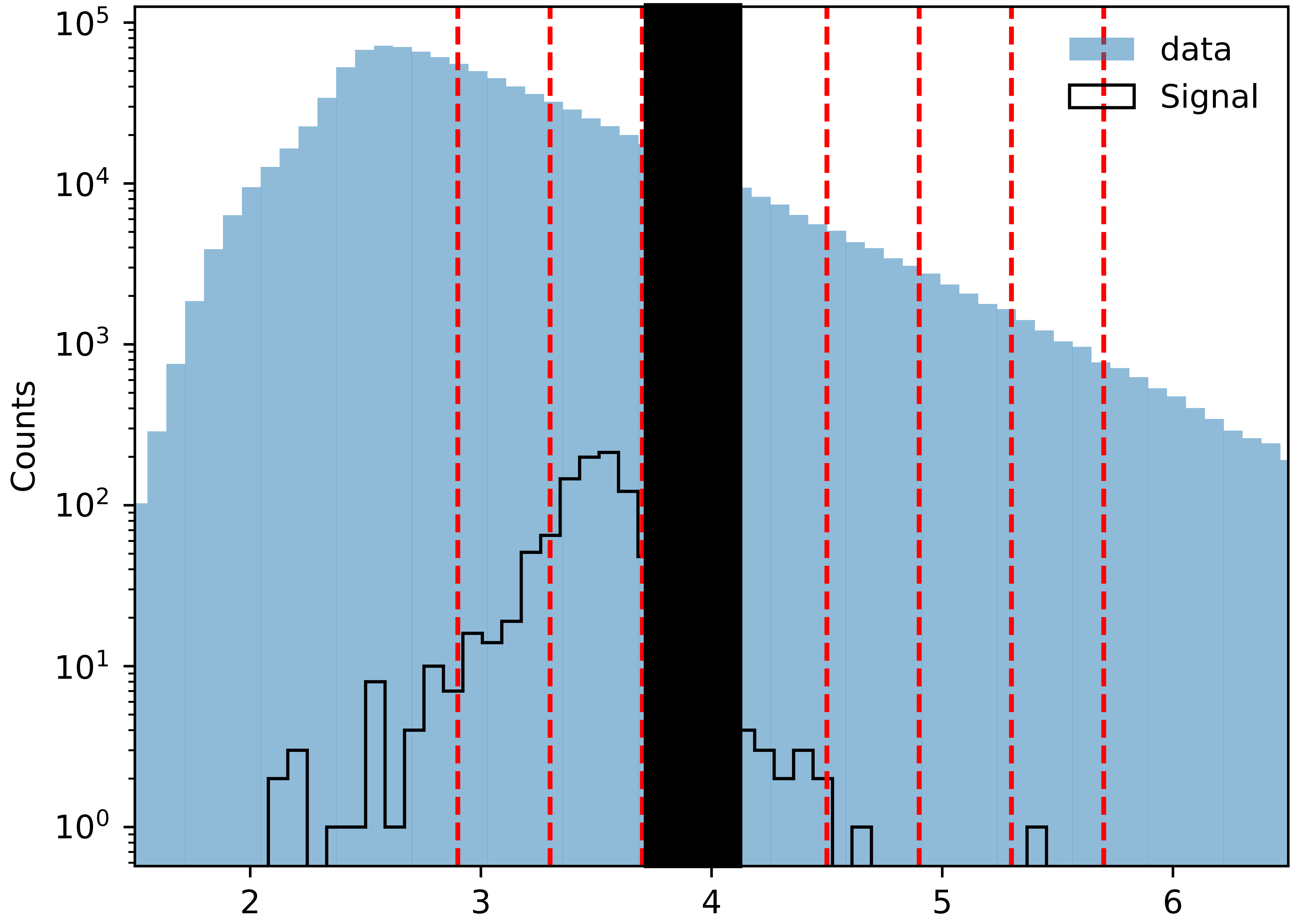
Background Template generation is computationally expensive!

- For each SR, a separate generative model is re-trained on almost the entire data, by masking out that SR.



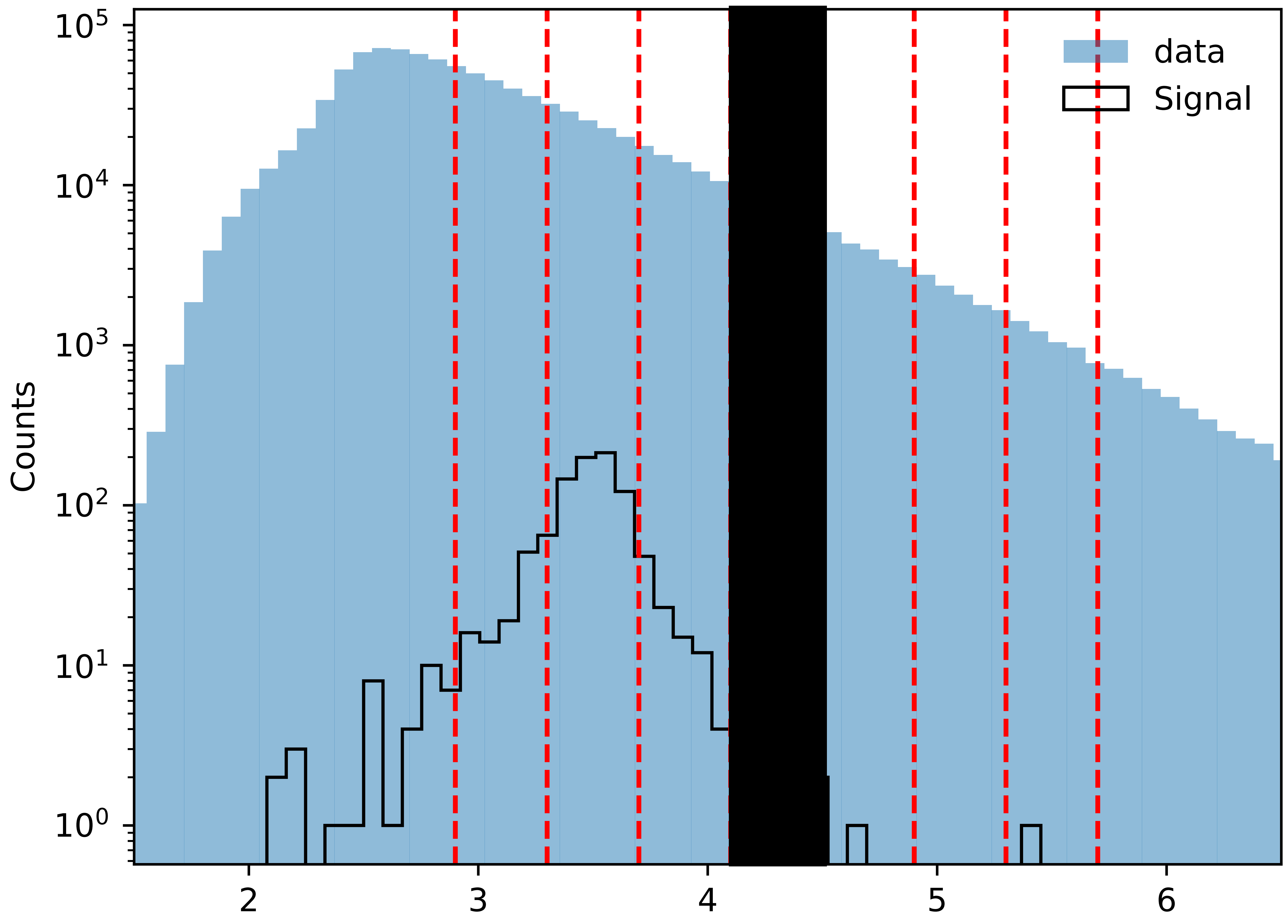
Background Template generation is computationally expensive!

- For each SR, a separate generative model is re-trained on almost the entire data, by masking out that SR.



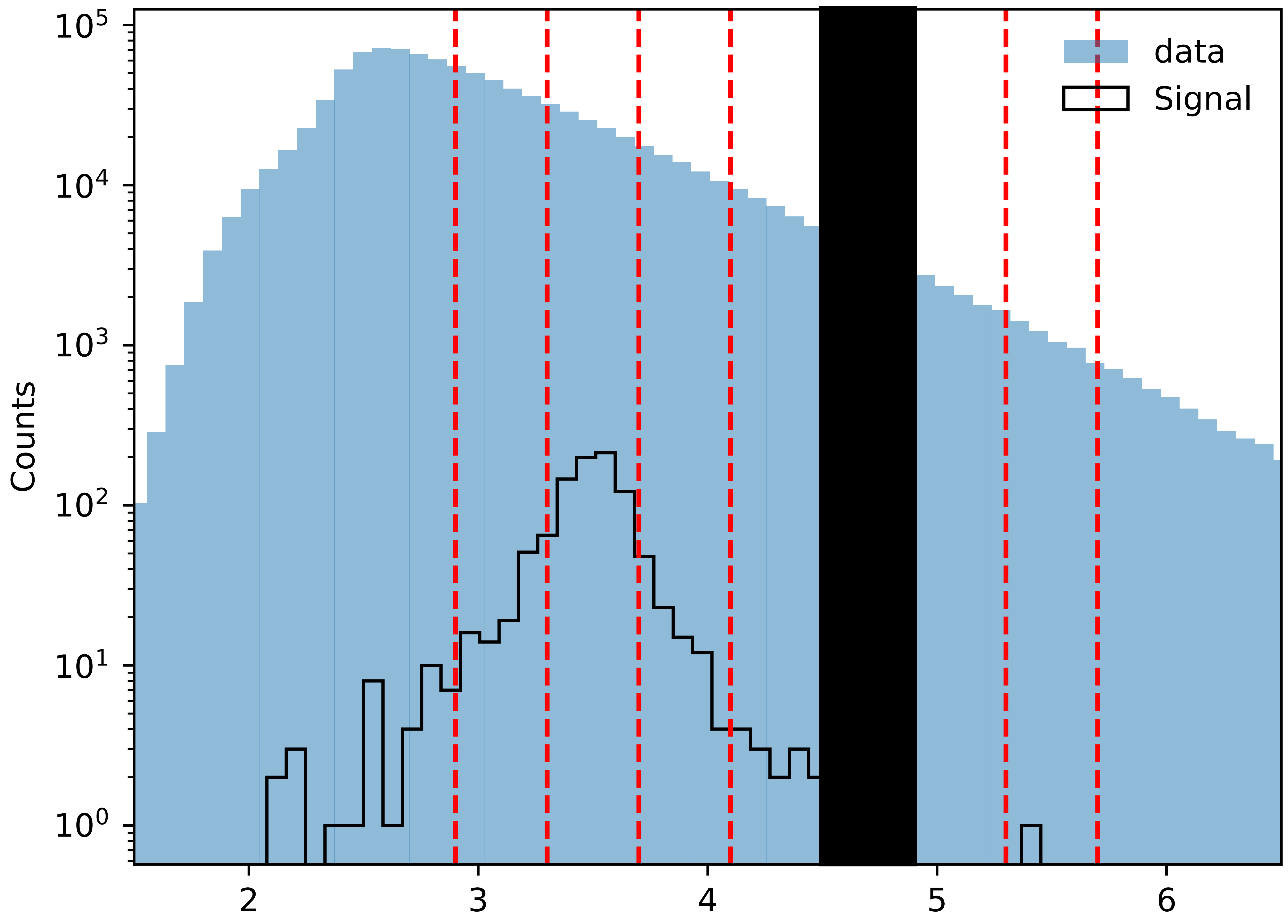
Background Template generation is computationally expensive!

- For each SR, a separate generative model is re-trained on almost the entire data, by masking out that SR.



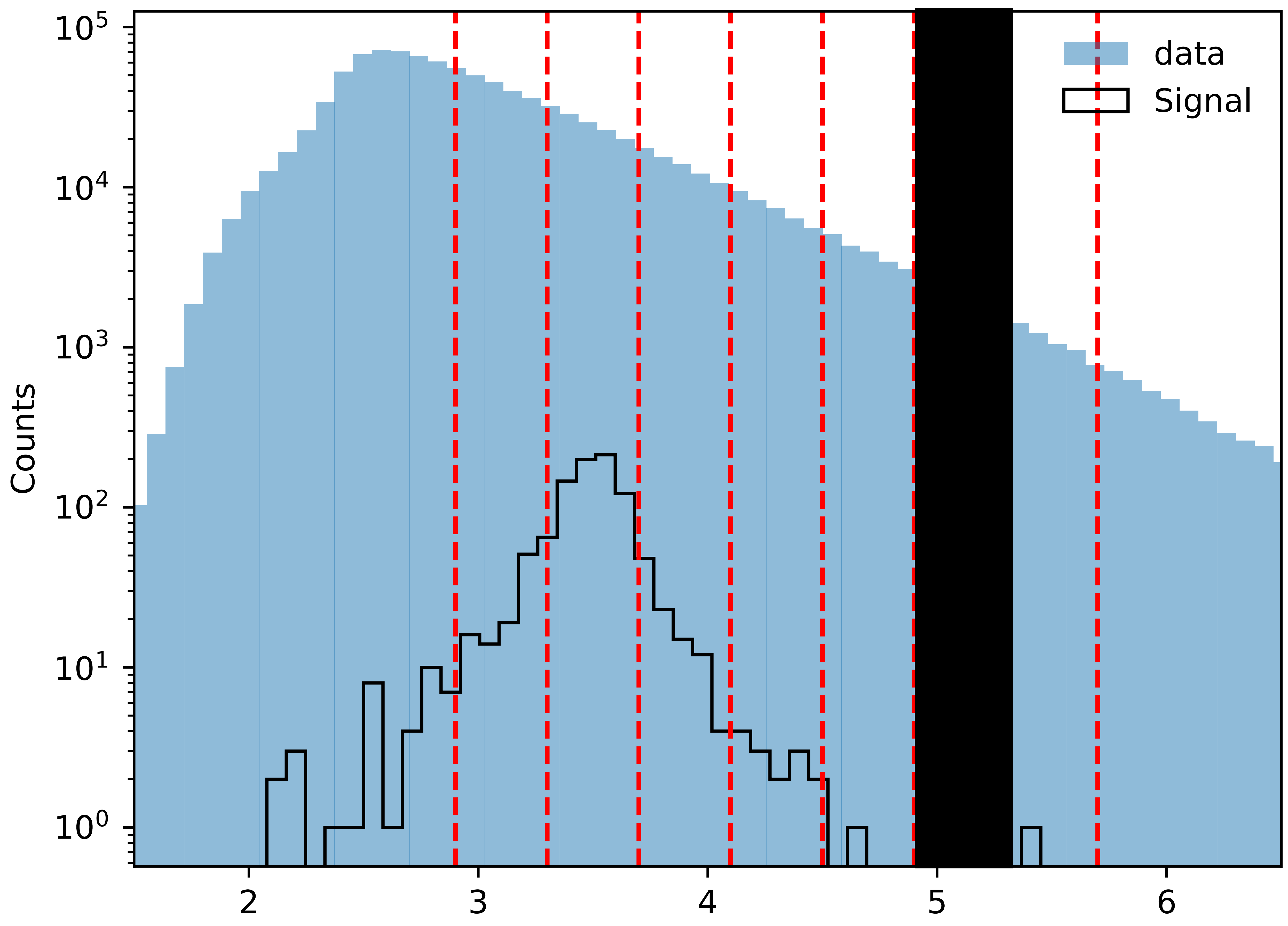
Background Template generation is computationally expensive!

- For each SR, a separate generative model is re-trained on almost the entire data, by masking out that SR.



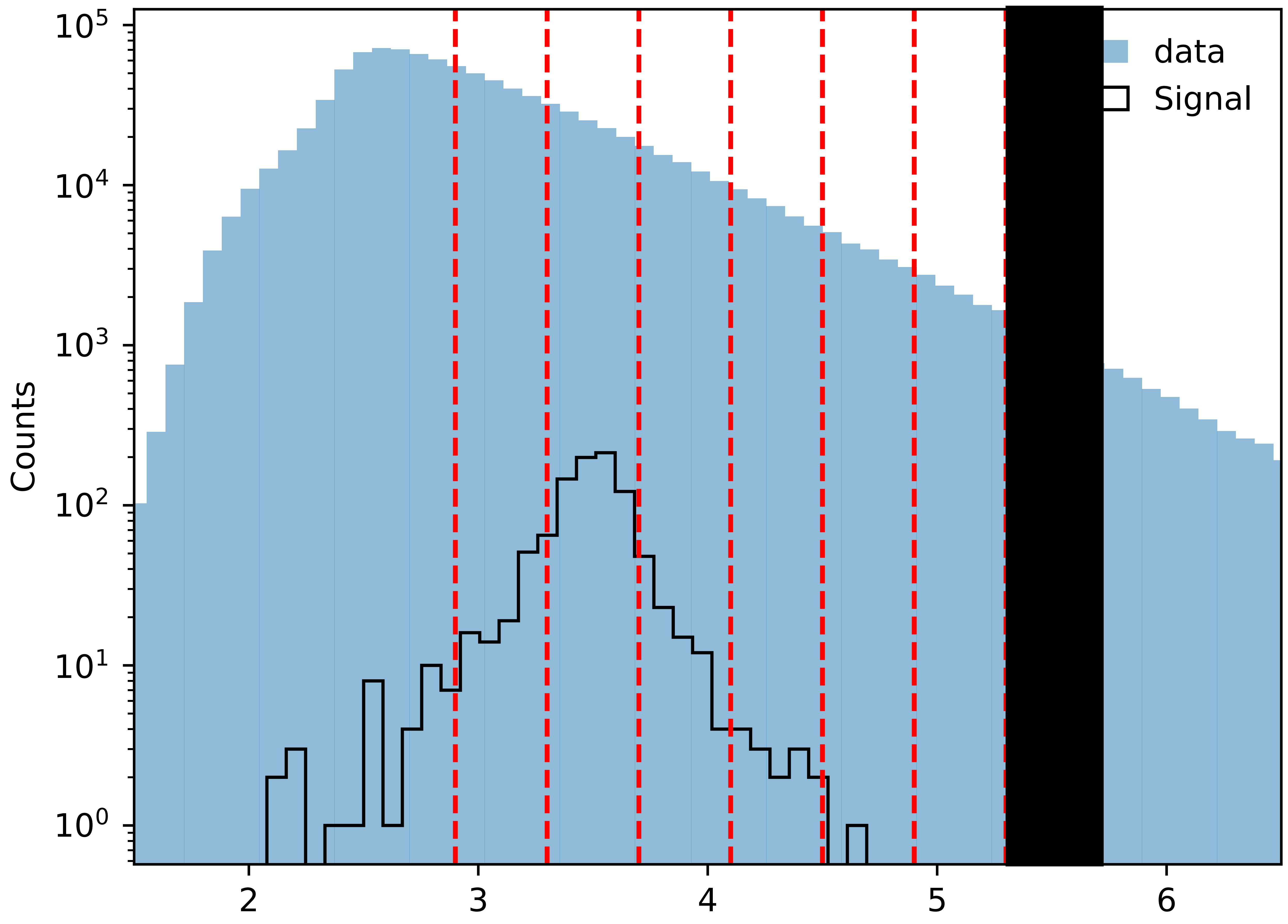
Background Template generation is computationally expensive!

- For each SR, a separate generative model is re-trained on almost the entire data, by masking out that SR.



Background Template generation is computationally expensive!

- For each SR, a separate generative model is re-trained on almost the entire data, by masking out that SR.

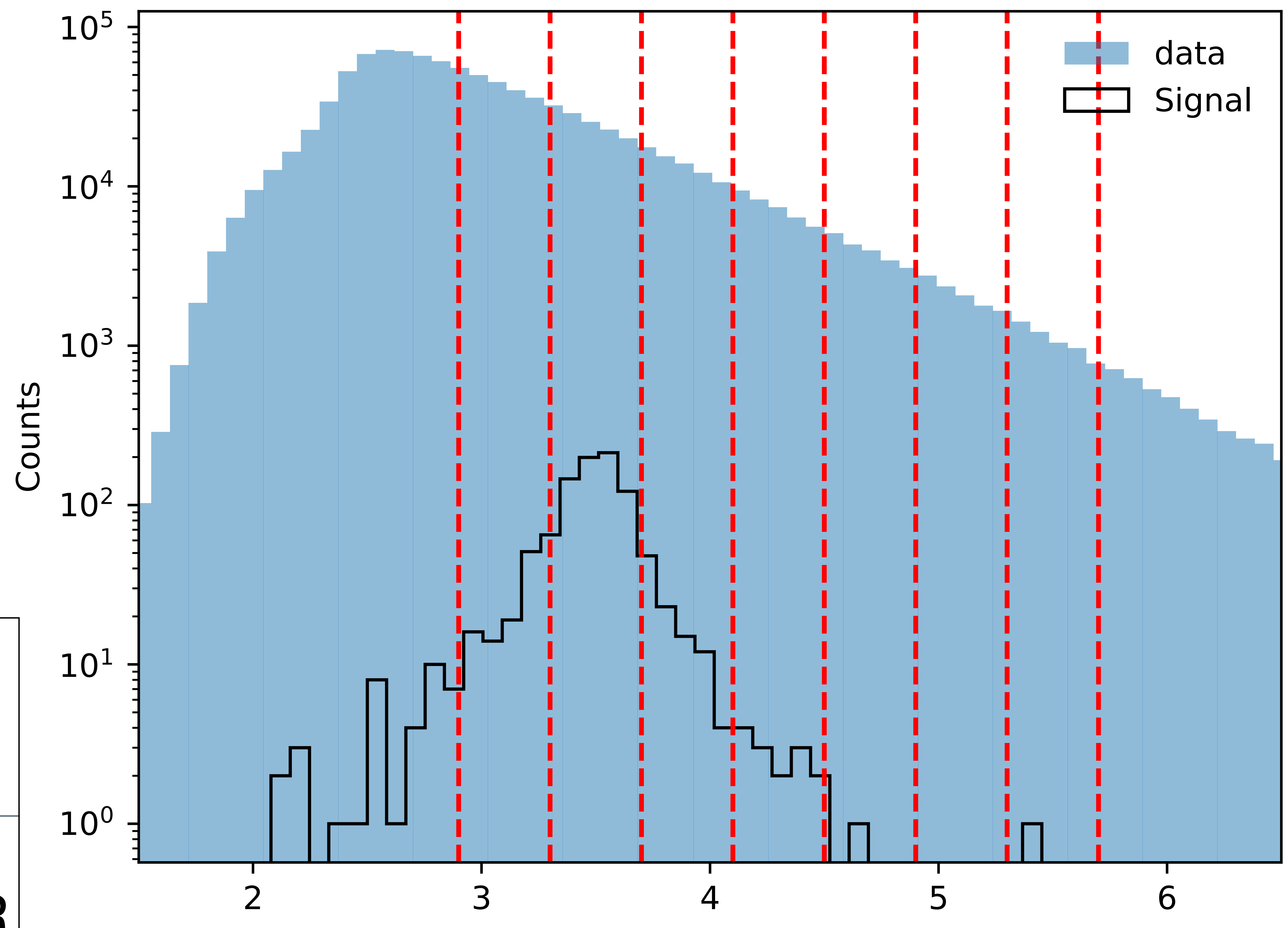


Background Template generation is computationally expensive!

- For each SR, a separate generative model is re-trained on almost the entire data, by masking out that SR.

- This makes the method computationally expensive for datasets with many SRs!

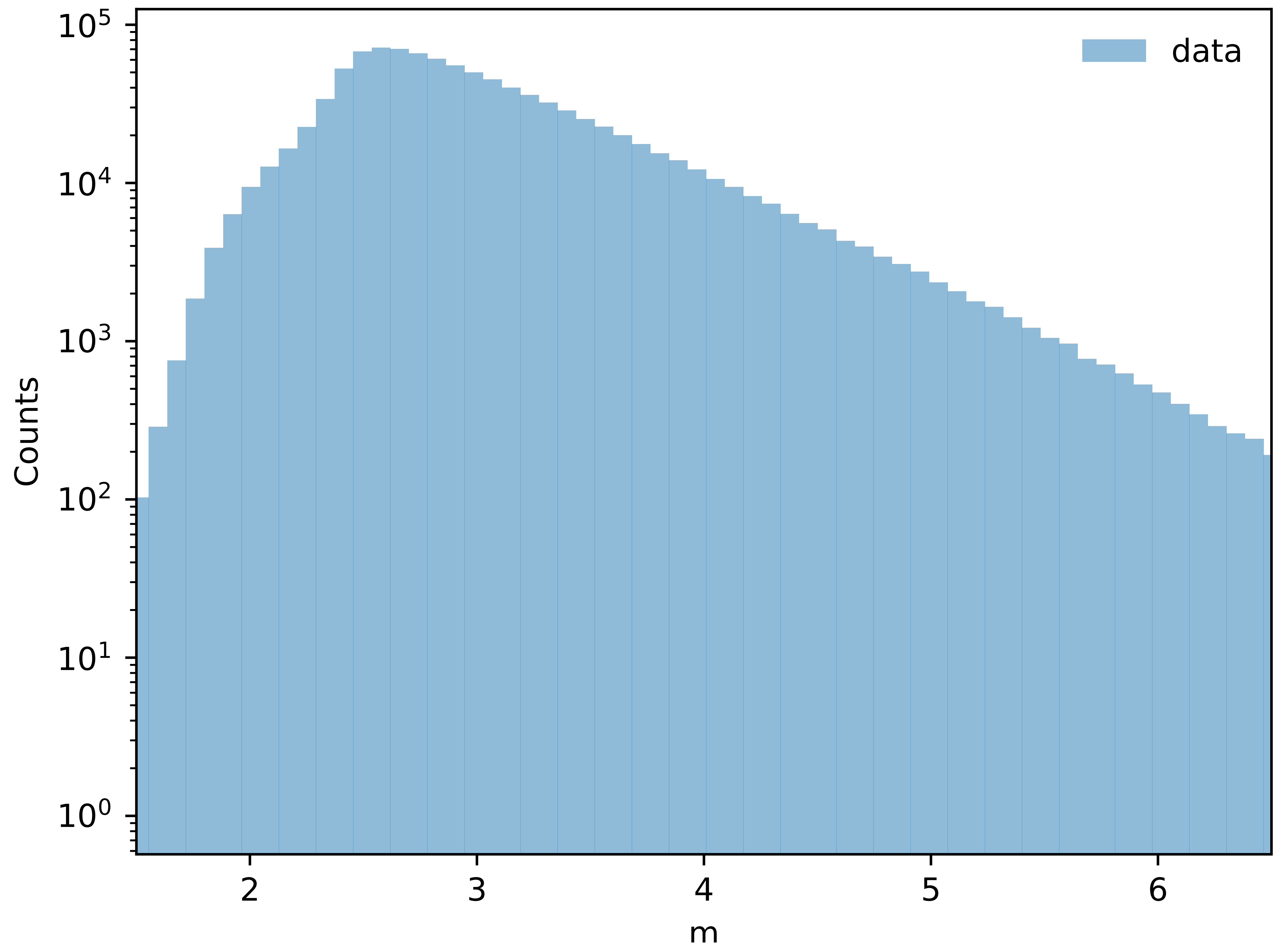
Method	Generative Model	Timing
CATHODE/ ANODE	Normalizing Flows	3 hours per SR



SIGMA: Single Interpolated Generative Model for Anomalies

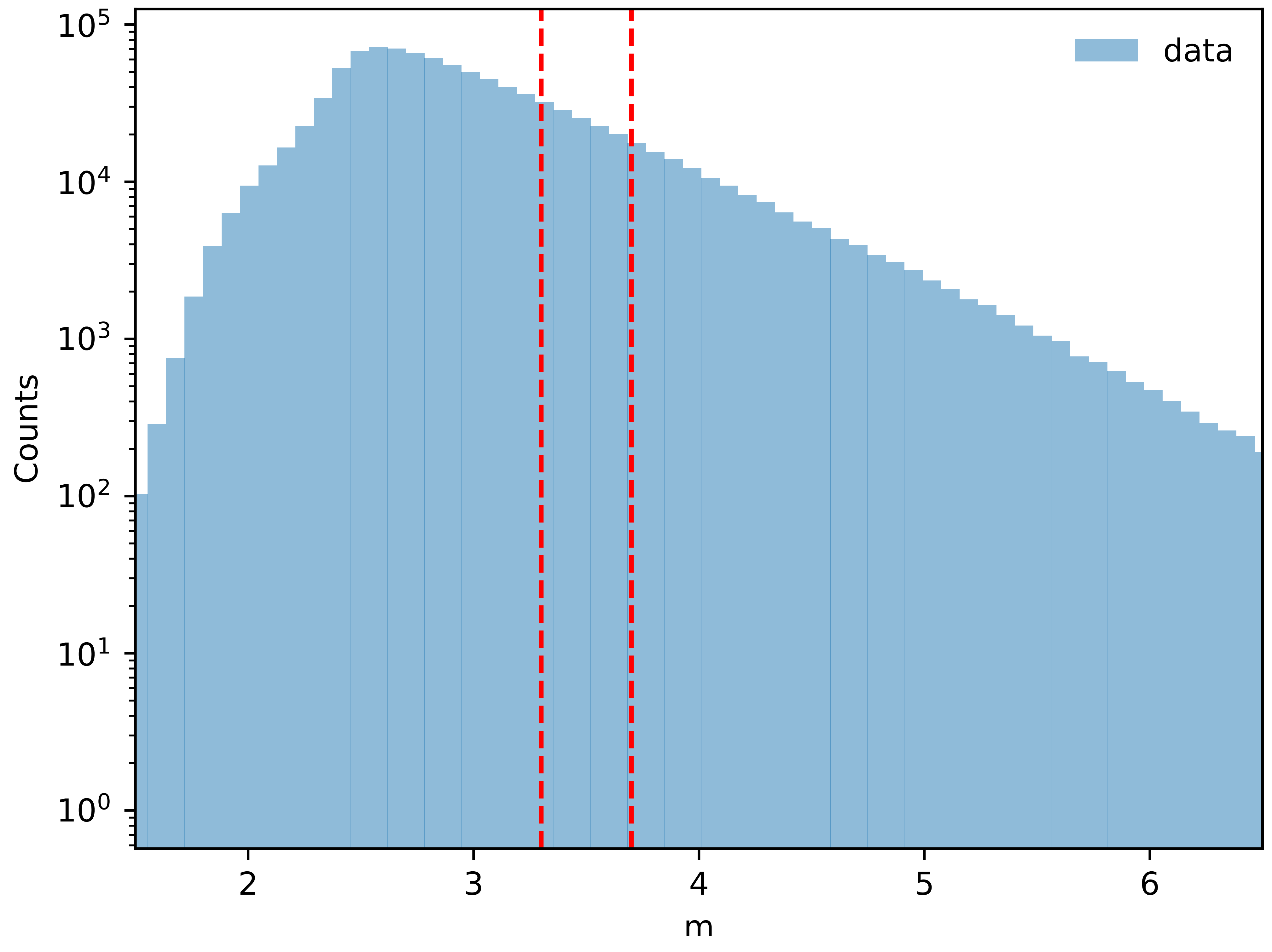
SIGMA: Single Interpolated Generative Model for Anomalies

- We train a single generative model, conditioned on the resonant feature ***m,*** on the entire dataset including signal.



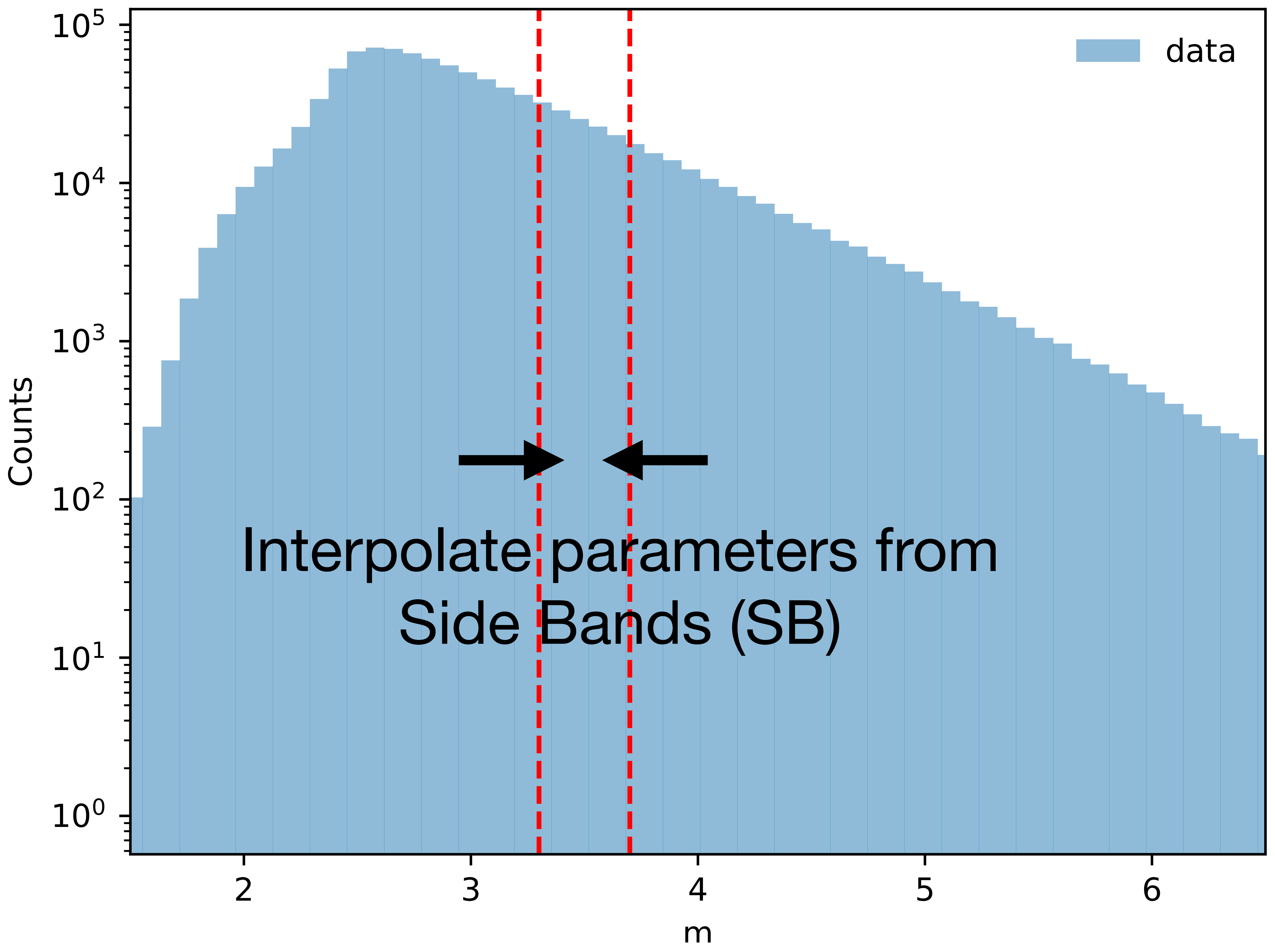
SIGMA: Single Interpolated Generative Model for Anomalies

- We train a single generative model, conditioned on the resonant feature ***m,*** on the entire dataset including signal.



SIGMA: Single Interpolated Generative Model for Anomalies

- We train a single generative model, conditioned on the resonant feature ***m,*** on the entire dataset including signal.
- For each SR, we interpolate the parameters of this model from nearby SB.
- Background template for all SRs are generated from a single trained model (no other training required).



Generative model: Conditional Flow-matching (CFM)

arXiv:2310.00049: EPiC-ly Fast Particle Cloud Generation with Flow-Matching and Diffusion
arXiv:2209.15571: Building Normalizing Flows with Stochastic Interpolants

arXiv:2210.02747: Flow Matching for Generative Modeling
arXiv:2312.00123: Flow Matching Beyond Kinematics: Generating Jets with Particle-ID and Trajectory Displacement Information

Generative model: Conditional Flow-matching (CFM)

Known
Base
Distribution

[arXiv:2310.00049](): EPiC-ly Fast Particle Cloud Generation with Flow-Matching and Diffusion
[arXiv:2209.15571](): Building Normalizing Flows with Stochastic Interpolants
[arXiv:2210.02747:]() Flow Matching for Generative Modeling
[arXiv:2312.00123](): Flow Matching Beyond Kinematics: Generating Jets with Particle-ID and Trajectory Displacement Information

Generative model: Conditional Flow-matching (CFM)

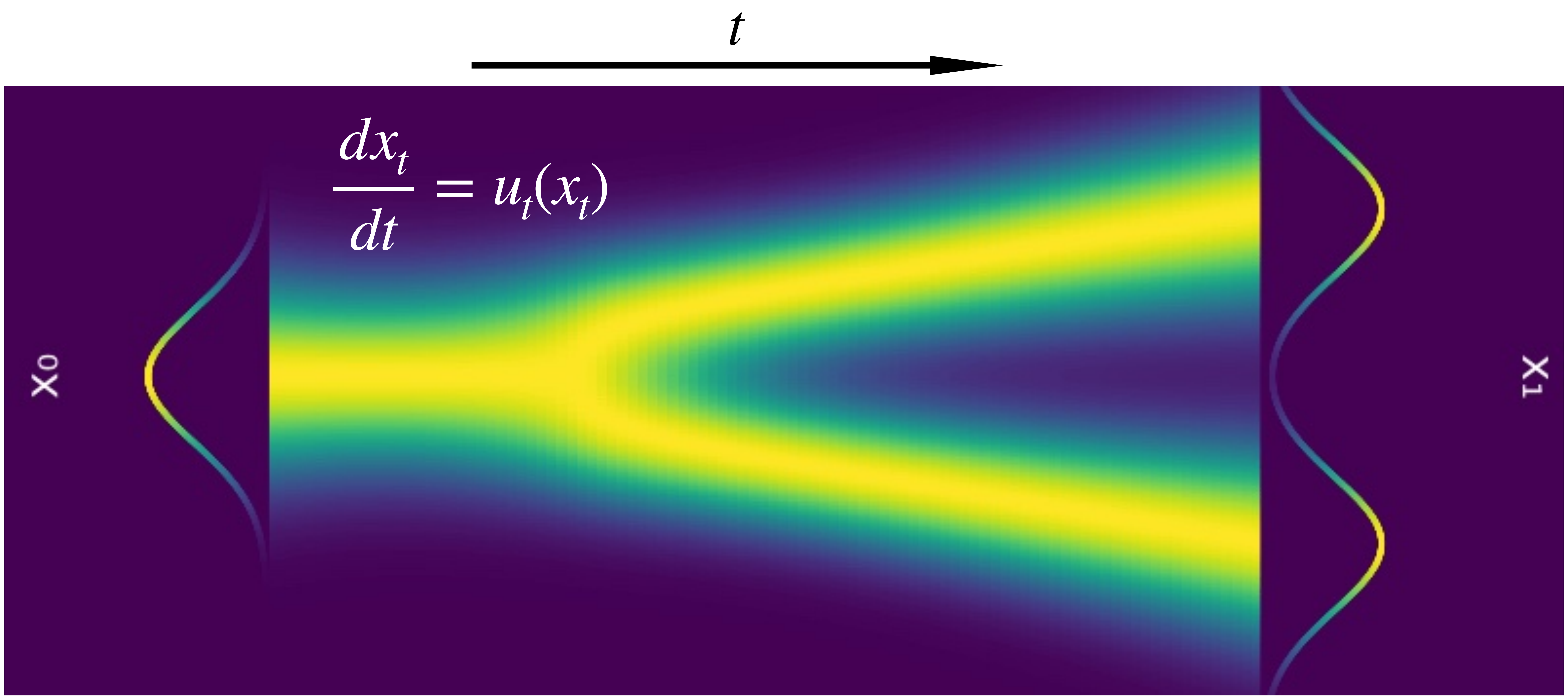
Known
Base
Distribution

Unknown
Data
Distribution

[arXiv:2310.00049](): EPiC-ly Fast Particle Cloud Generation with Flow-Matching and Diffusion
[arXiv:2209.15571](): Building Normalizing Flows with Stochastic Interpolants
[arXiv:2210.02747:]() Flow Matching for Generative Modeling
[arXiv:2312.00123](): Flow Matching Beyond Kinematics: Generating Jets with Particle-ID and Trajectory Displacement Information

Generative model: Conditional Flow-matching (CFM)

Known Base Distribution



Unknown Data Distribution

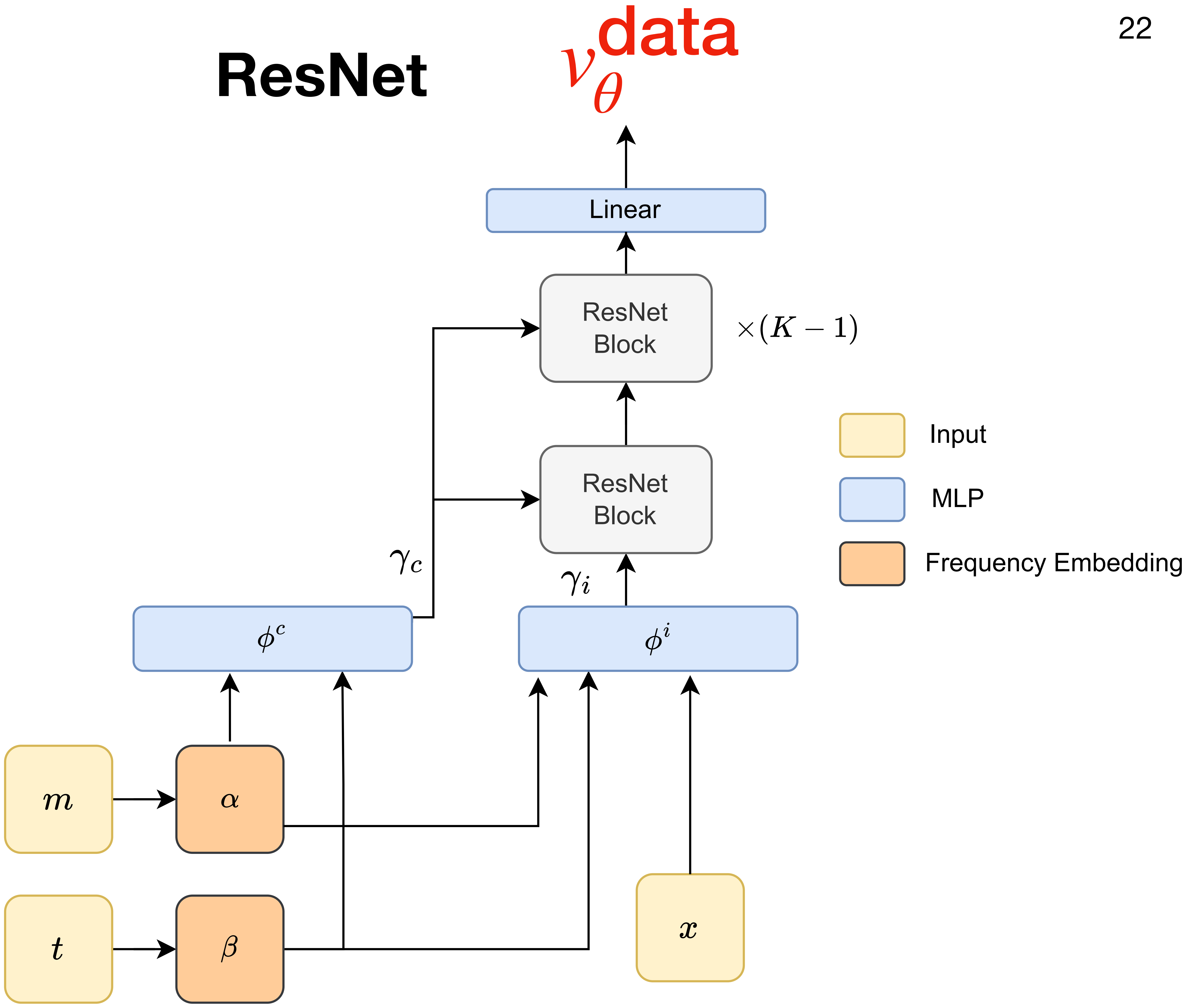
Image from https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html

Trains a neural network $v_\theta(x\,|\,t)$ to regress a conditional vector field $u_t(x\,|\,x_1)$, thereby learning the vector field $u_t(x)$

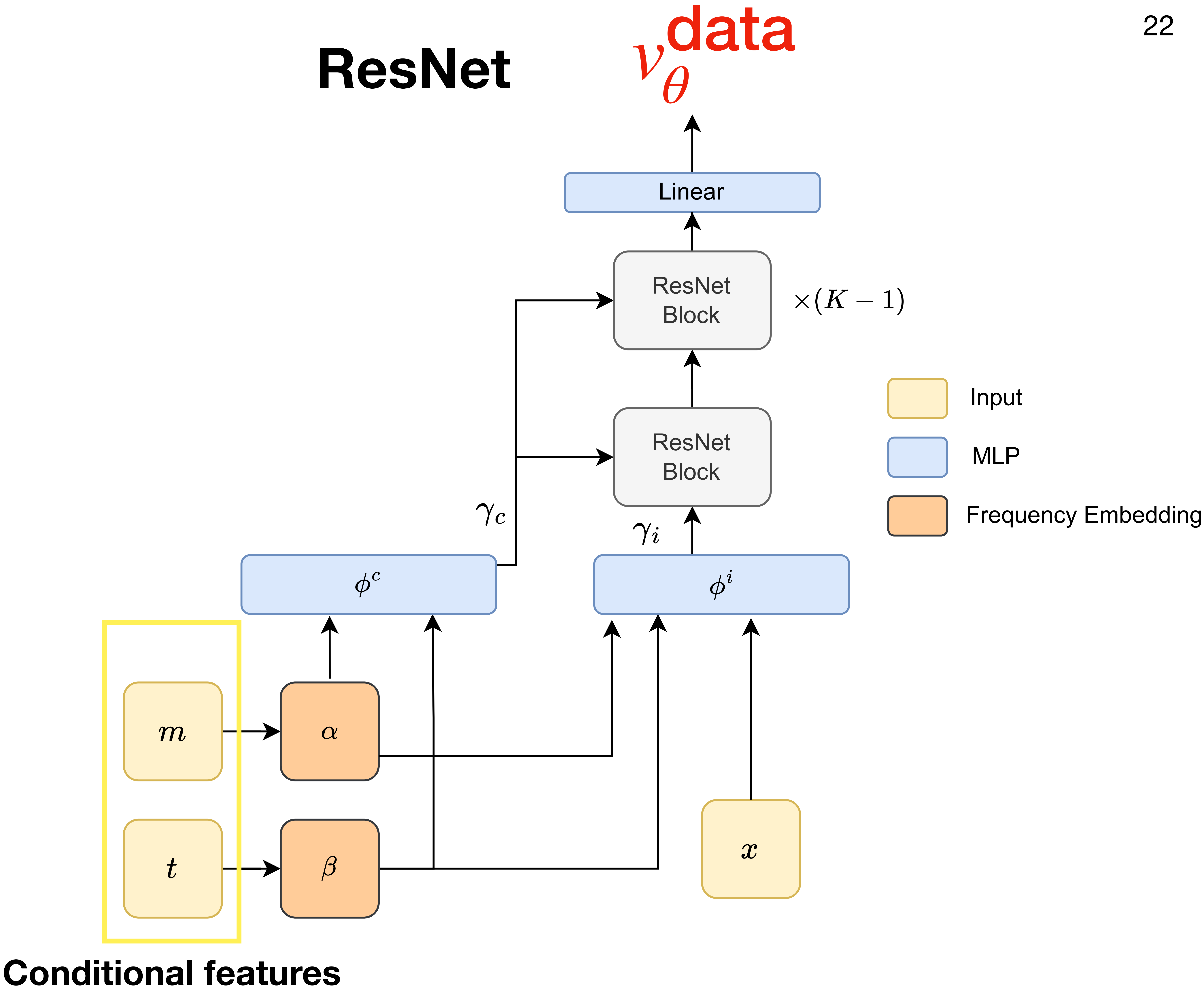
arXiv:2310.00049: EPiC-ly Fast Particle Cloud Generation with Flow-Matching and Diffusion
arXiv:2209.15571: Building Normalizing Flows with Stochastic Interpolants
arXiv:2210.02747: Flow Matching for Generative Modeling
arXiv:2312.00123: Flow Matching Beyond Kinematics: Generating Jets with Particle-ID and Trajectory Displacement Information

Architecture

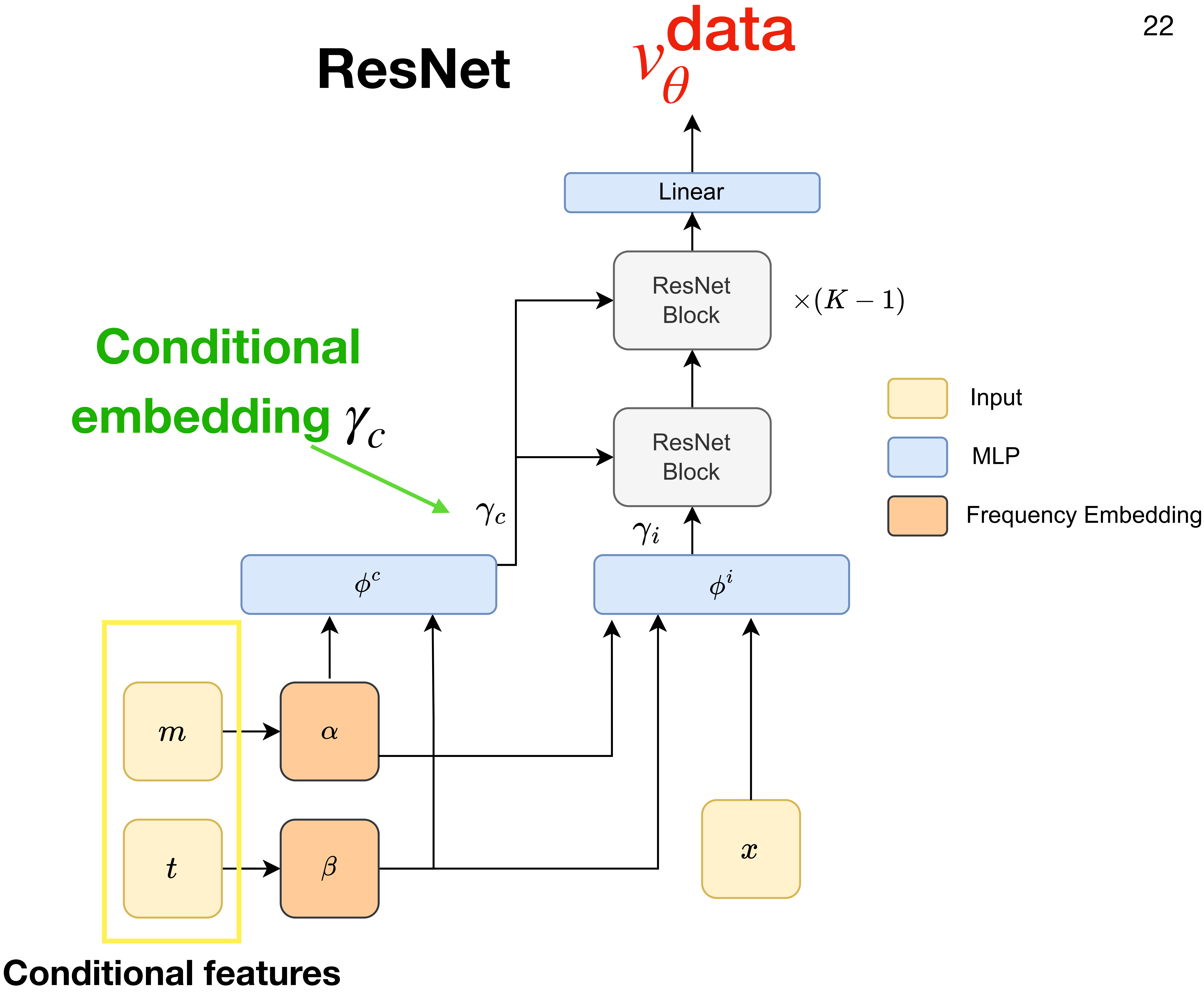
Architecture



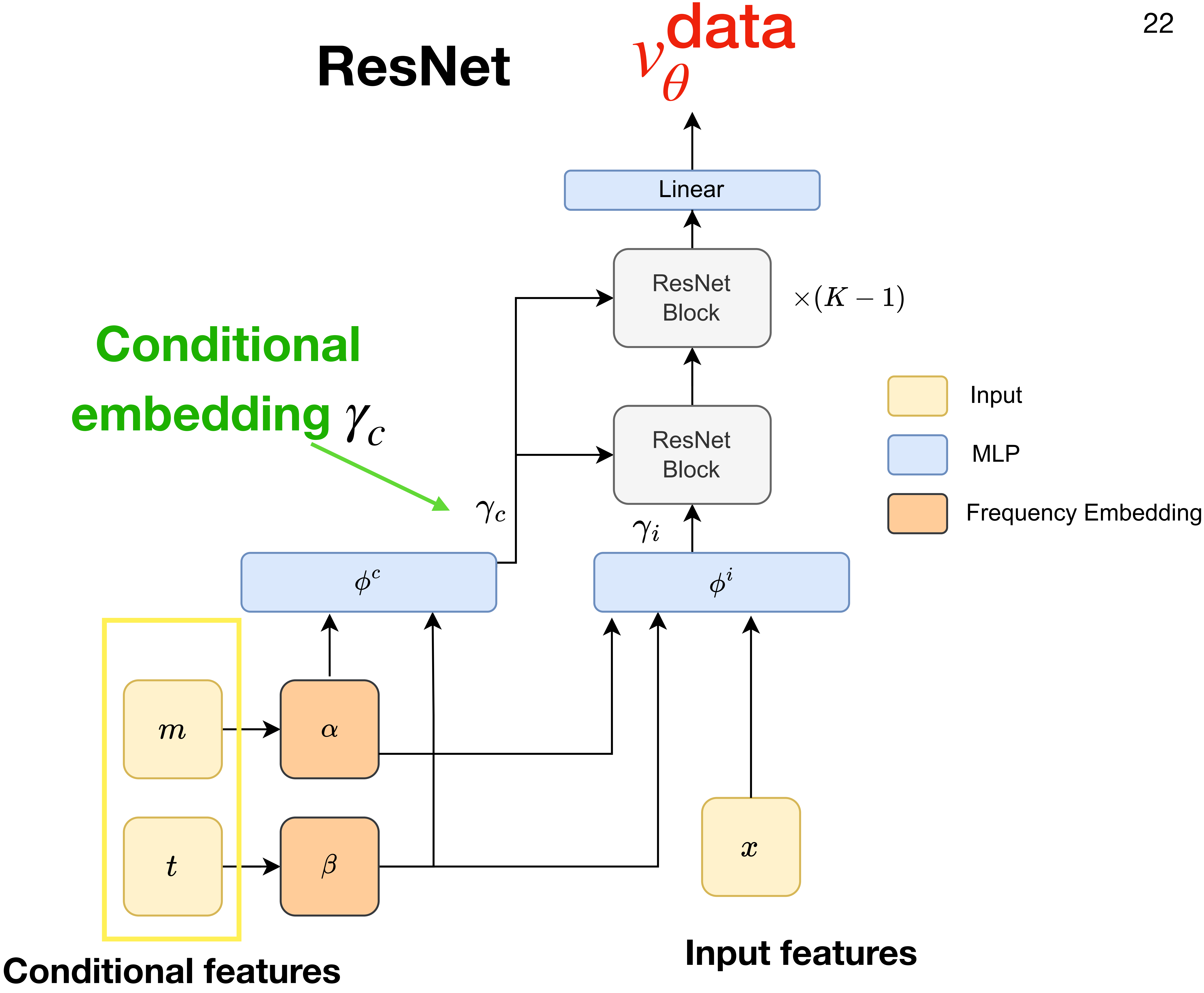
Architecture



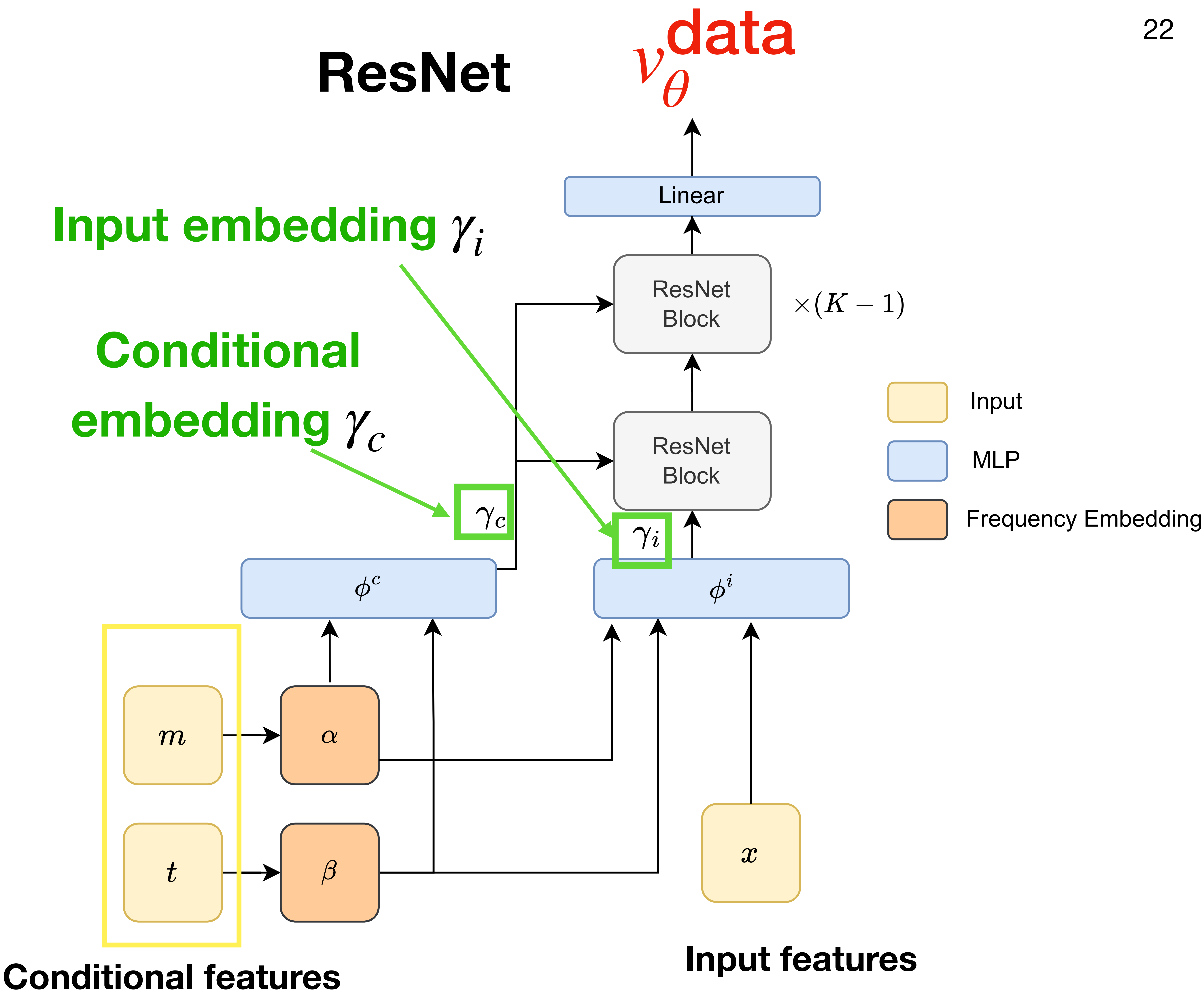
Architecture



Architecture



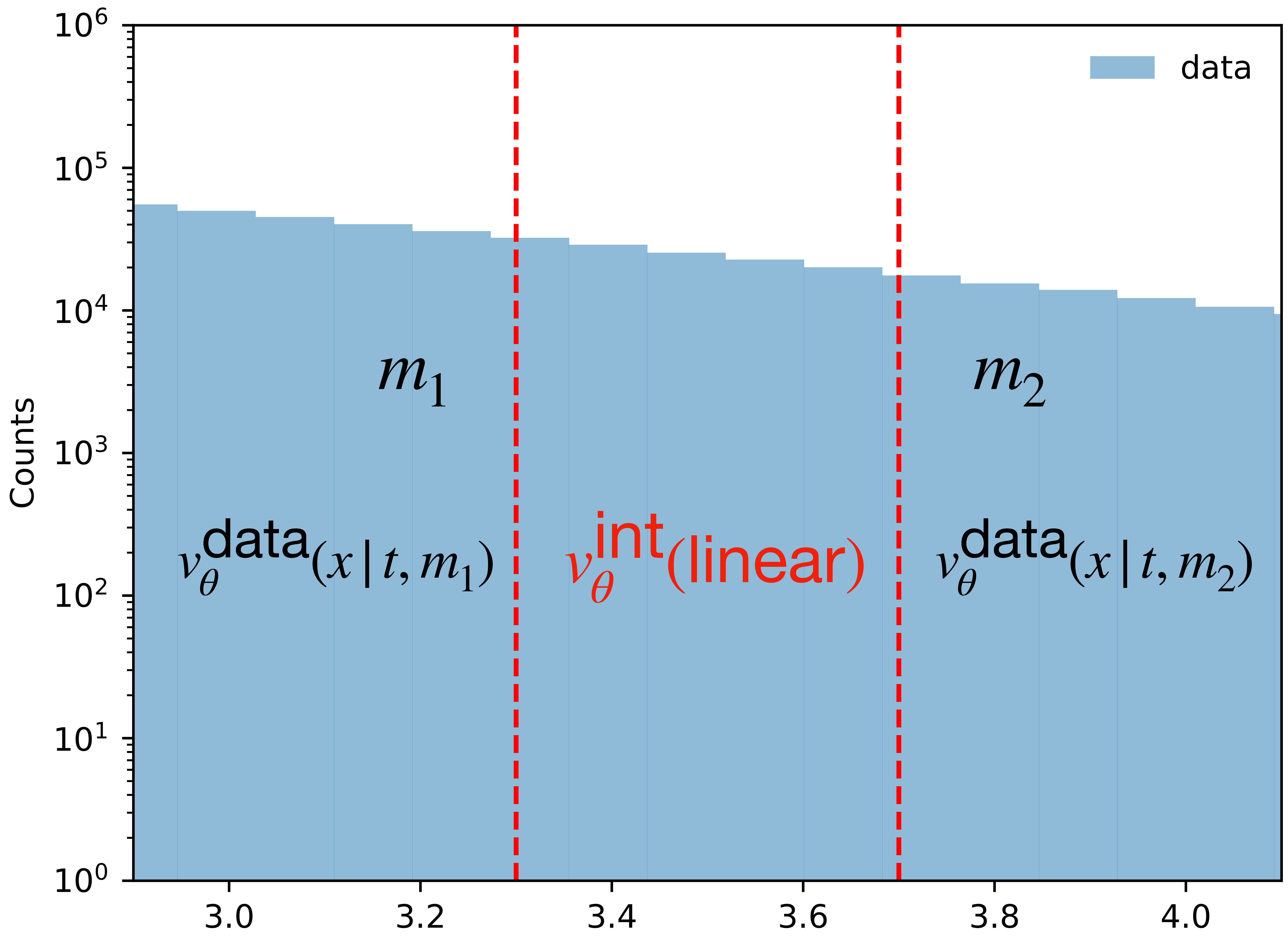
Architecture



Interpolation using SIGMA

- $v_\theta^{\text{int}}(x\,|\,t, m) = \xi * \; v_\theta^{\text{data}}(x\,|\,t, m_1) + (1-\xi) * v_\theta^{\text{data}}(x\,|\,t, m_2)$

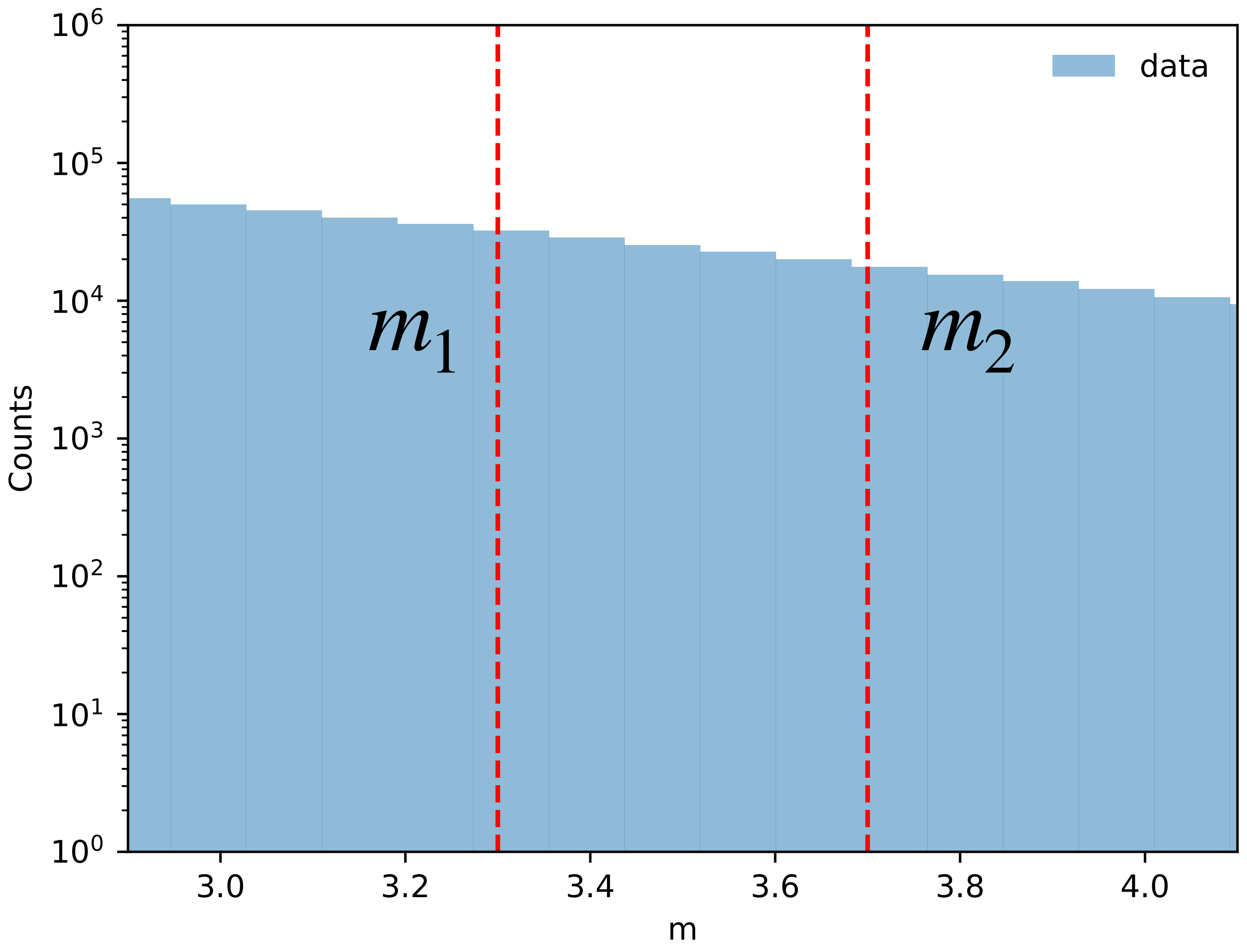
$$\xi = \frac{m - m_2}{m_1 - m_2}$$

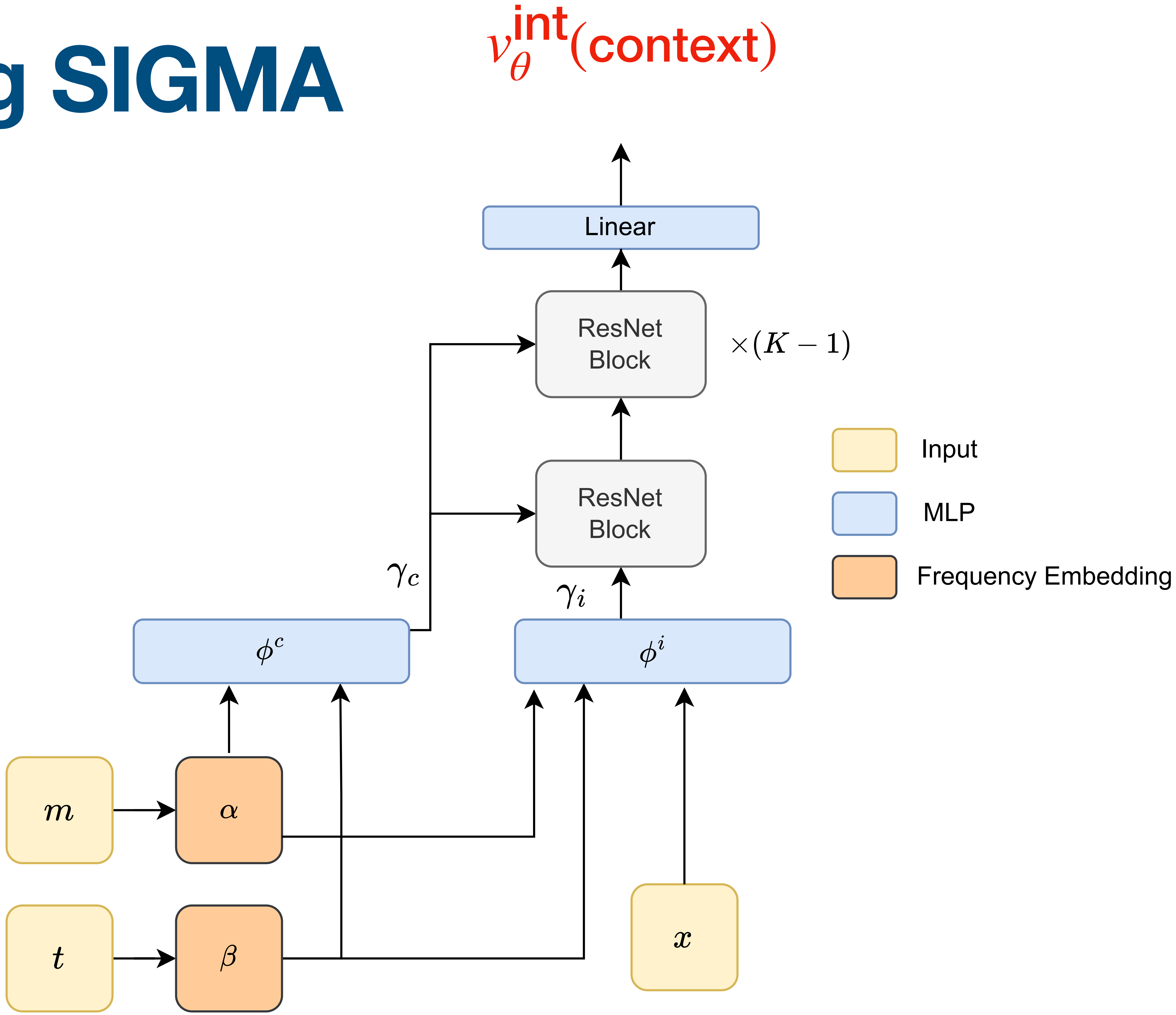


Linearly Interpolate the vector field from the Sidebands!

Interpolation using SIGMA

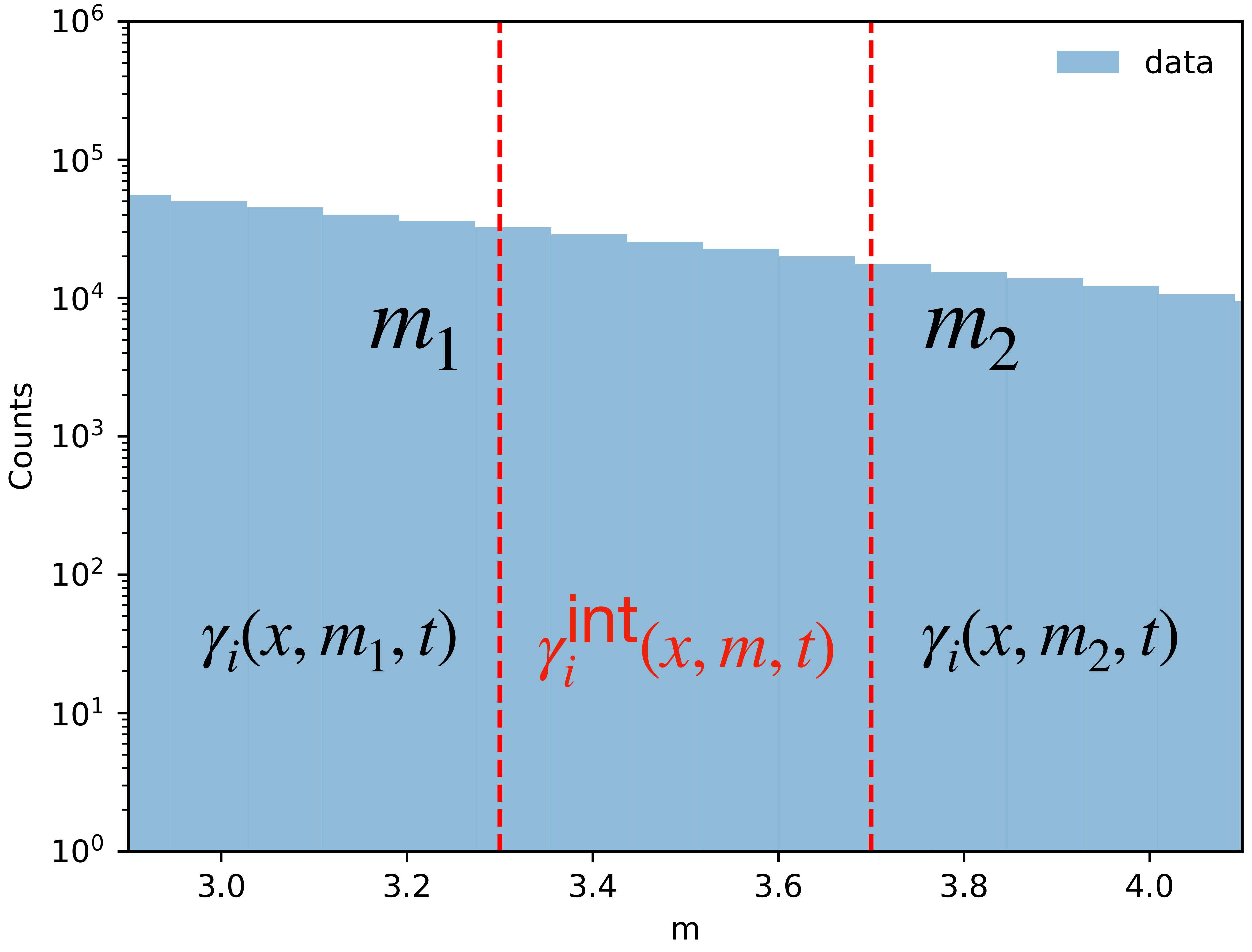
Interpolation using SIGMA

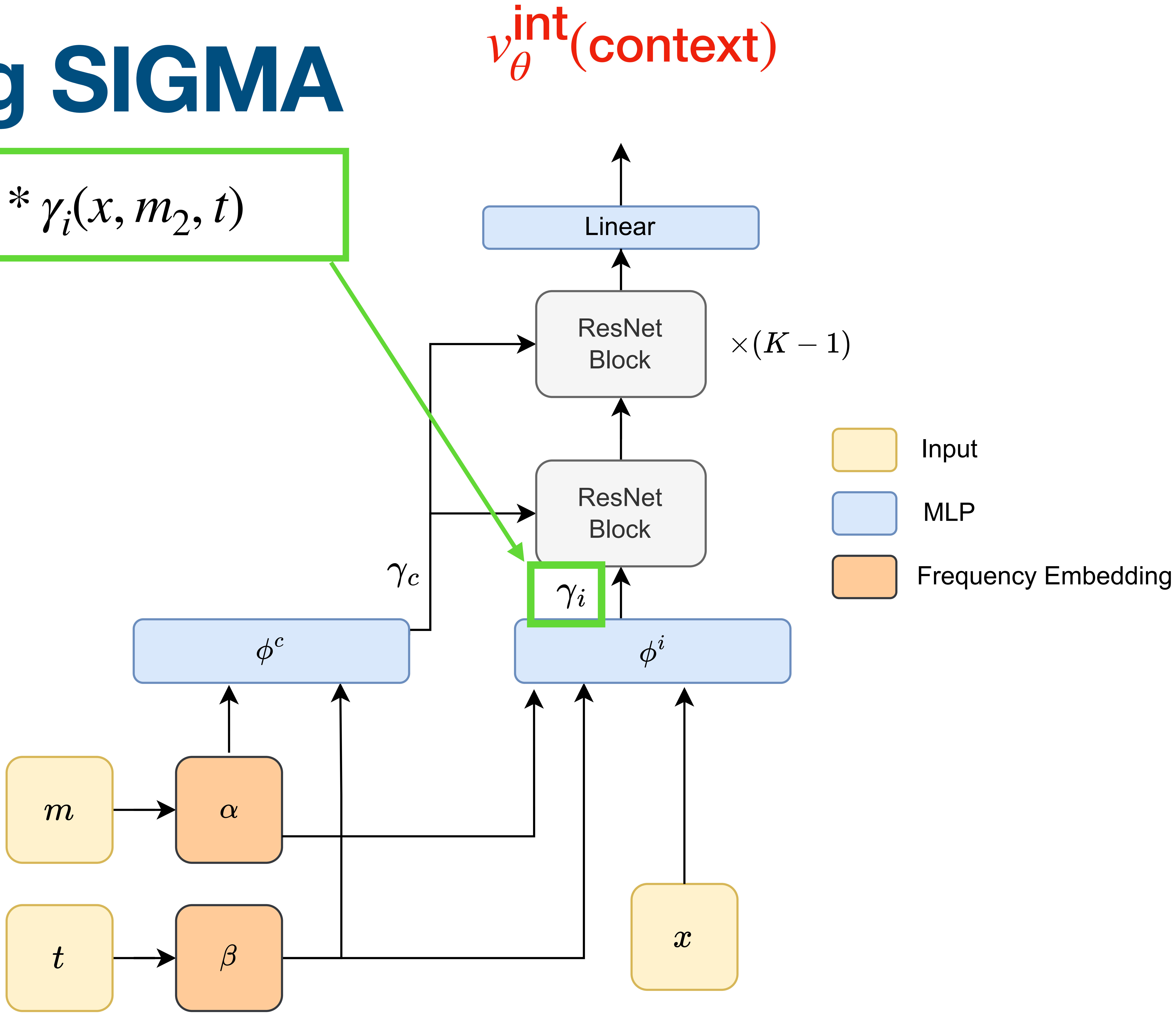




Interpolation using SIGMA

$$\gamma_i^{\text{int}}(x, m, t) = \xi * \gamma_i(x, m_1, t) + (1 - \xi) * \gamma_i(x, m_2, t)$$

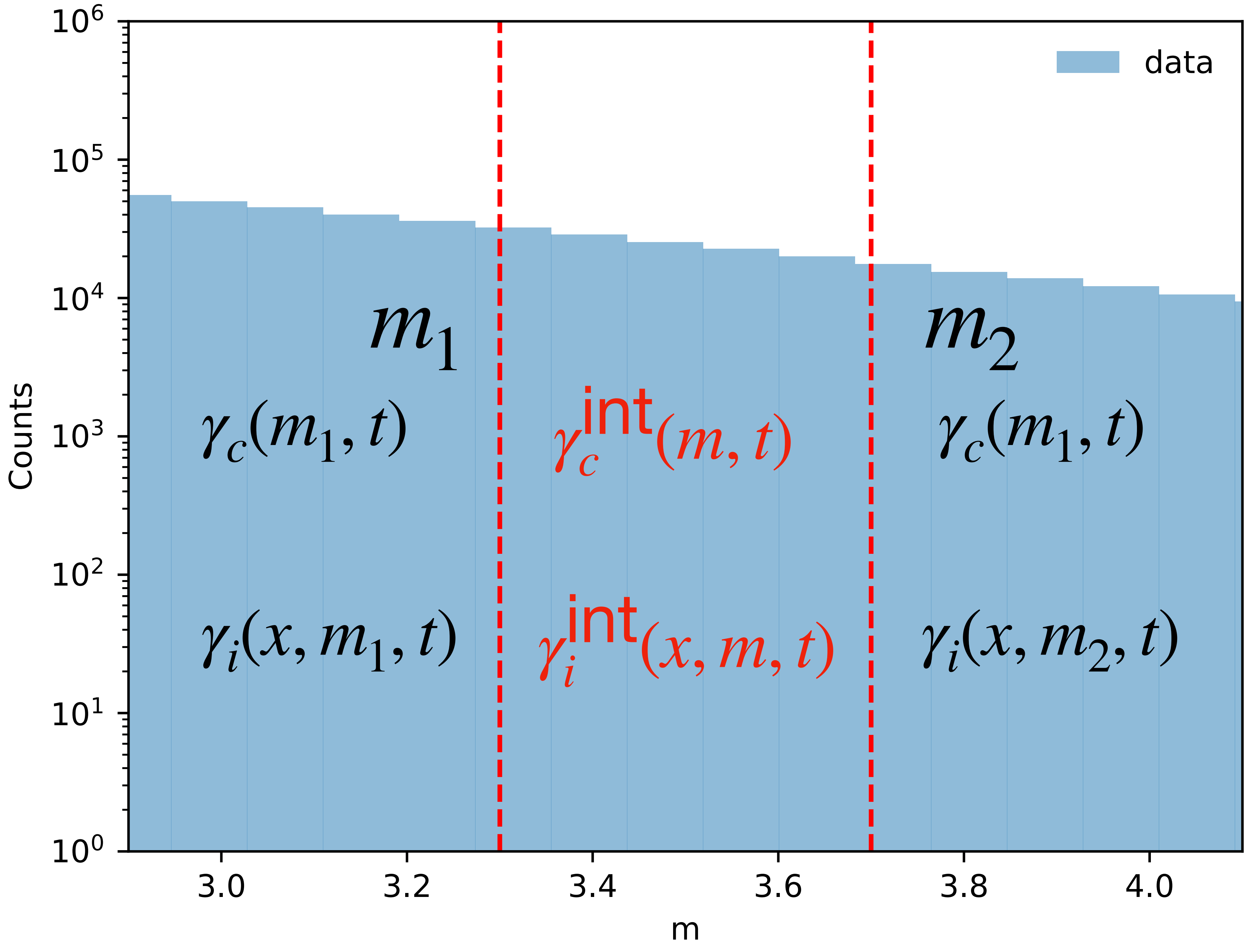


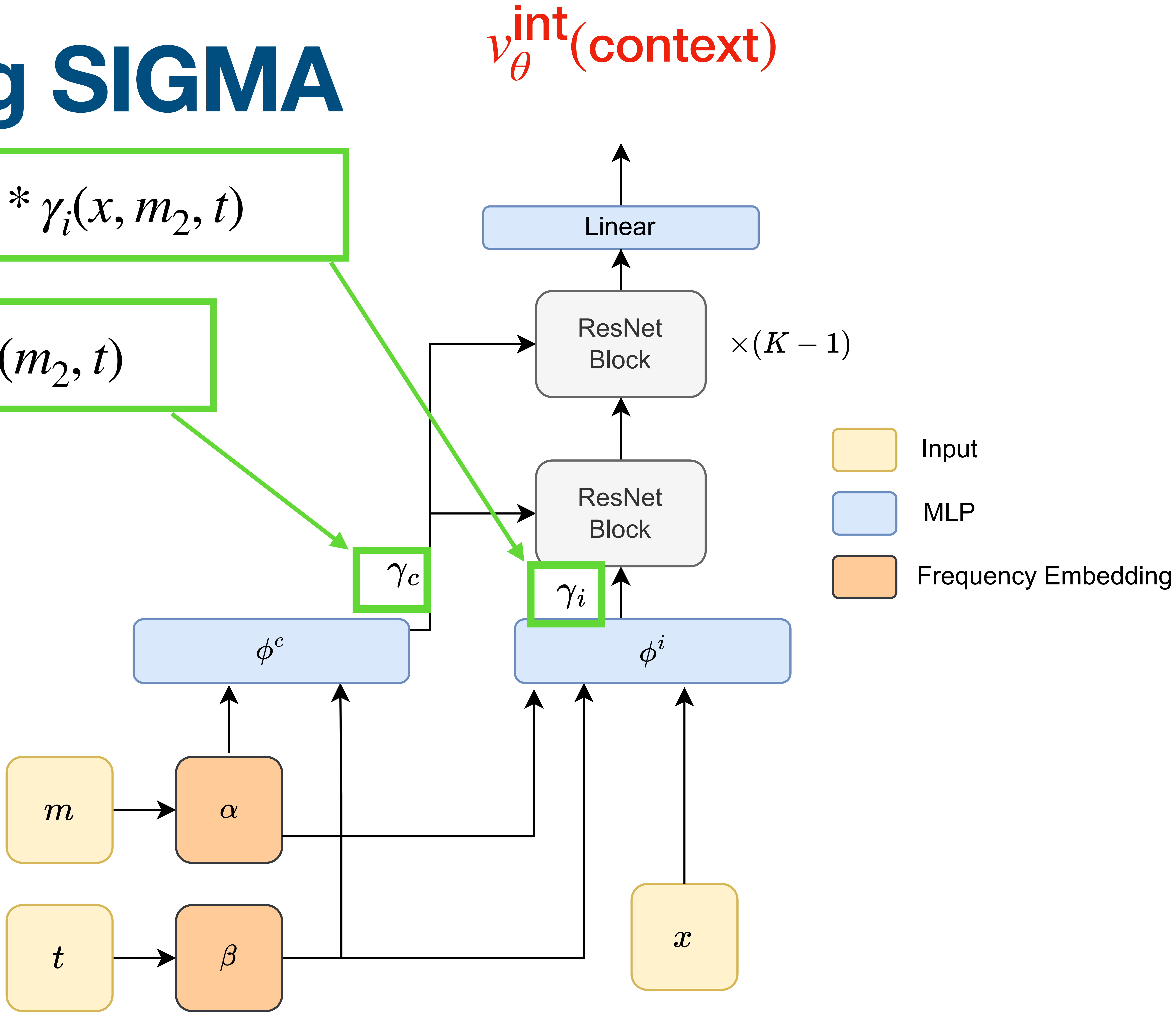


Interpolation using SIGMA

$$\gamma_i^{\text{int}}(x, m, t) = \xi * \gamma_i(x, m_1, t) + (1 - \xi) * \gamma_i(x, m_2, t)$$

$$\gamma_c^{\text{int}}(m, t) = \xi * \gamma_c(m_1, t) + (1 - \xi) * \gamma_c(m_2, t)$$

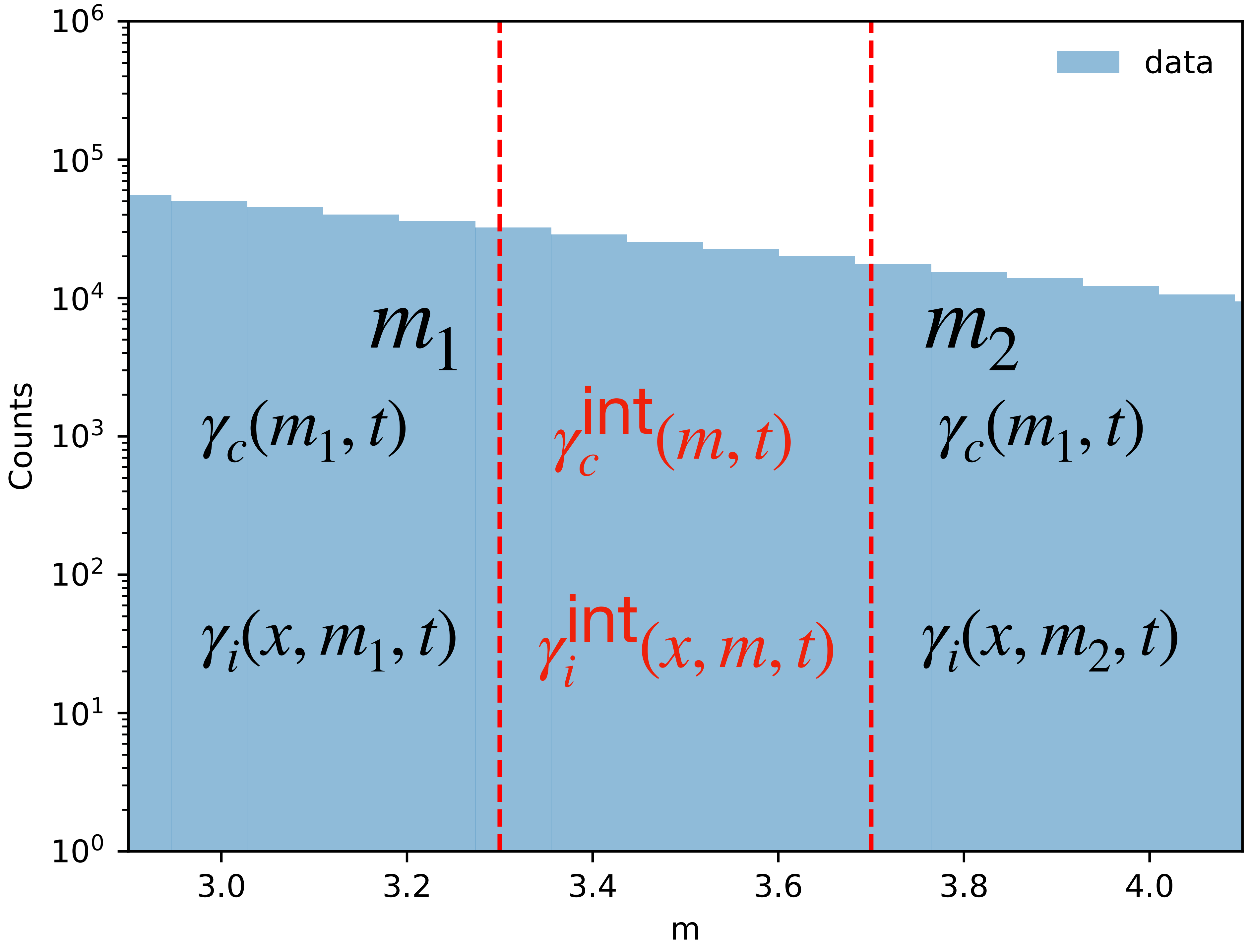


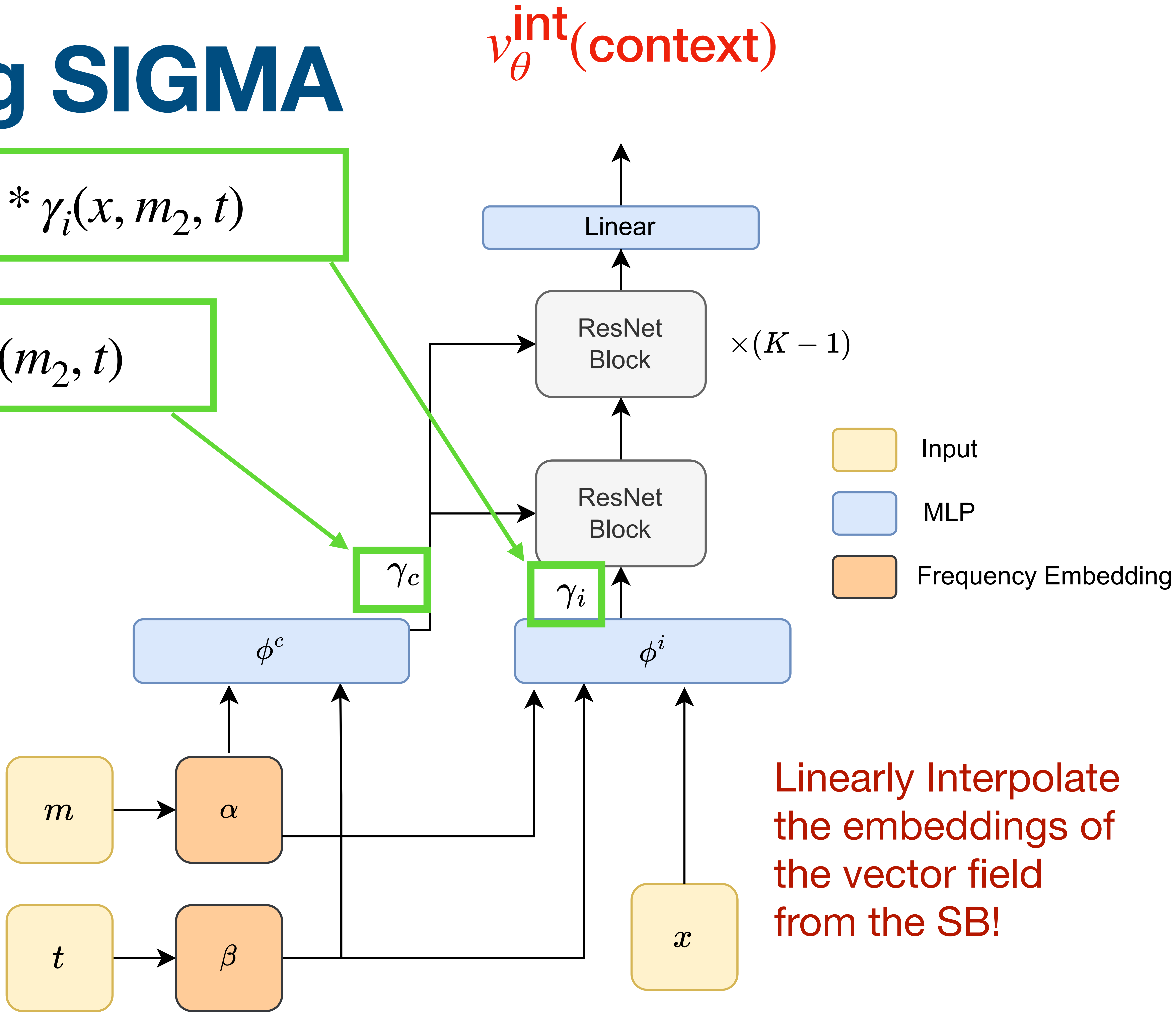


Interpolation using SIGMA

$$\gamma_i^{\text{int}}(x, m, t) = \xi * \gamma_i(x, m_1, t) + (1 - \xi) * \gamma_i(x, m_2, t)$$

$$\gamma_c^{\text{int}}(m, t) = \xi * \gamma_c(m_1, t) + (1 - \xi) * \gamma_c(m_2, t)$$

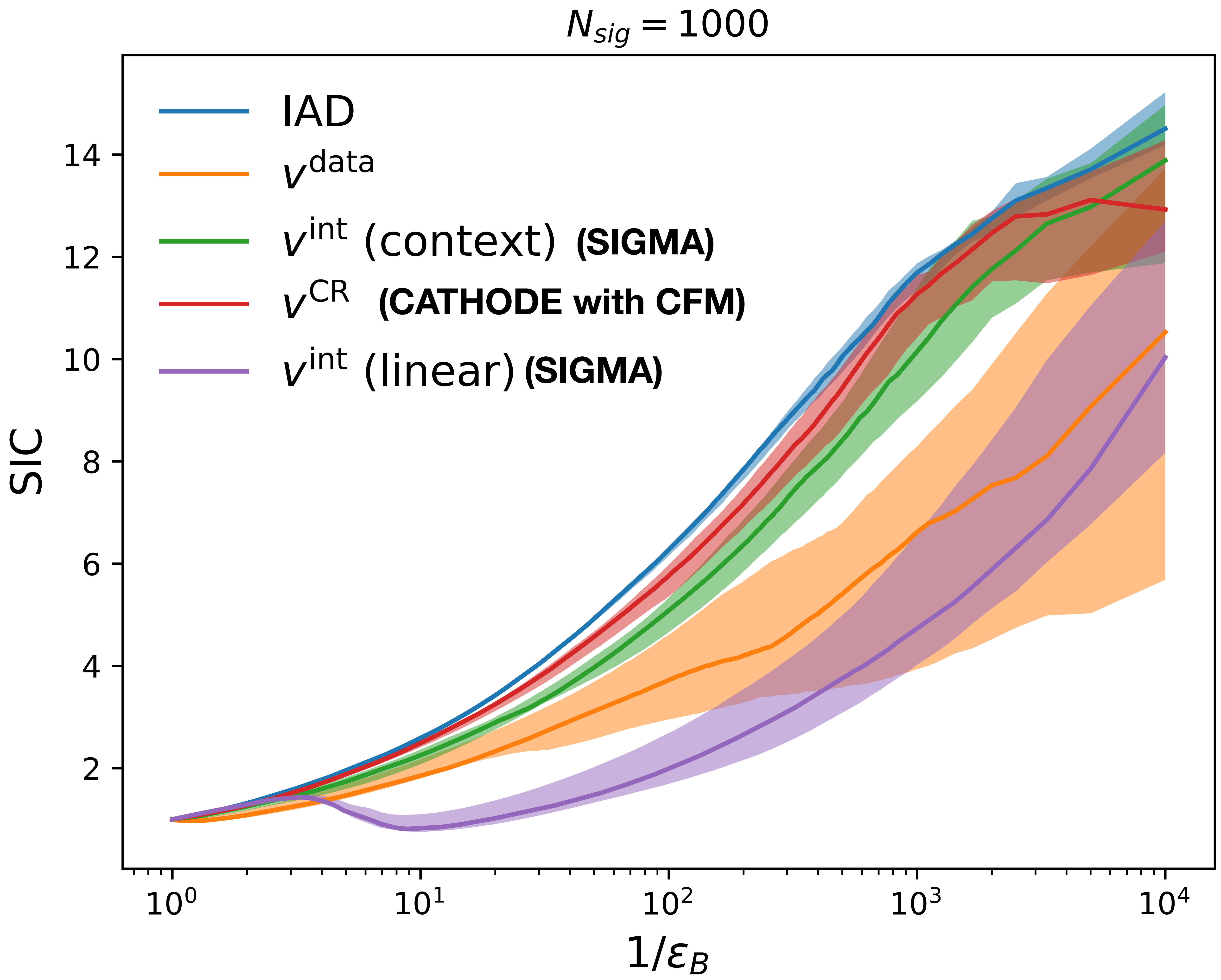




Linearly Interpolate the embeddings of the vector field from the SB!

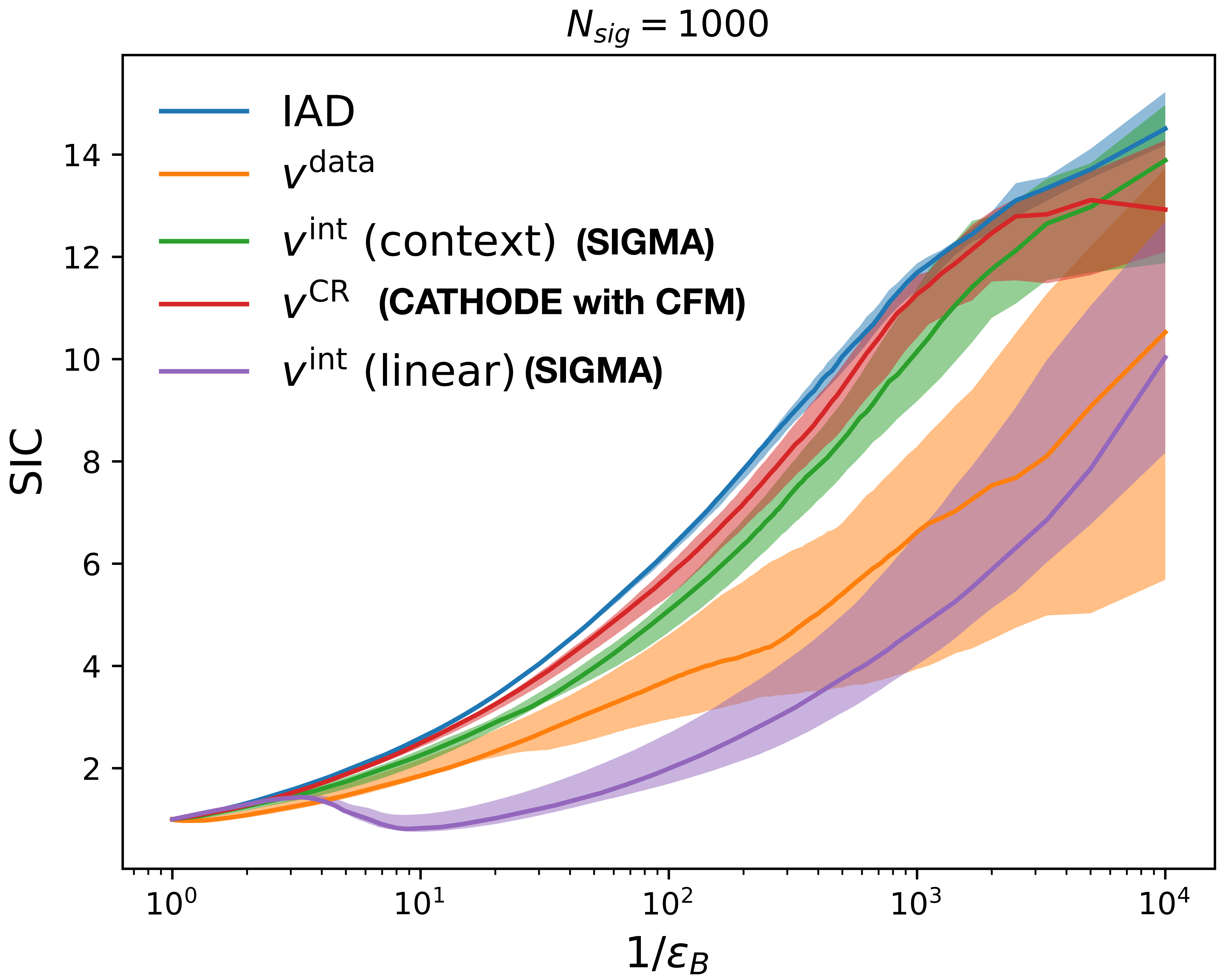
Anomaly detection performance

Anomaly detection performance



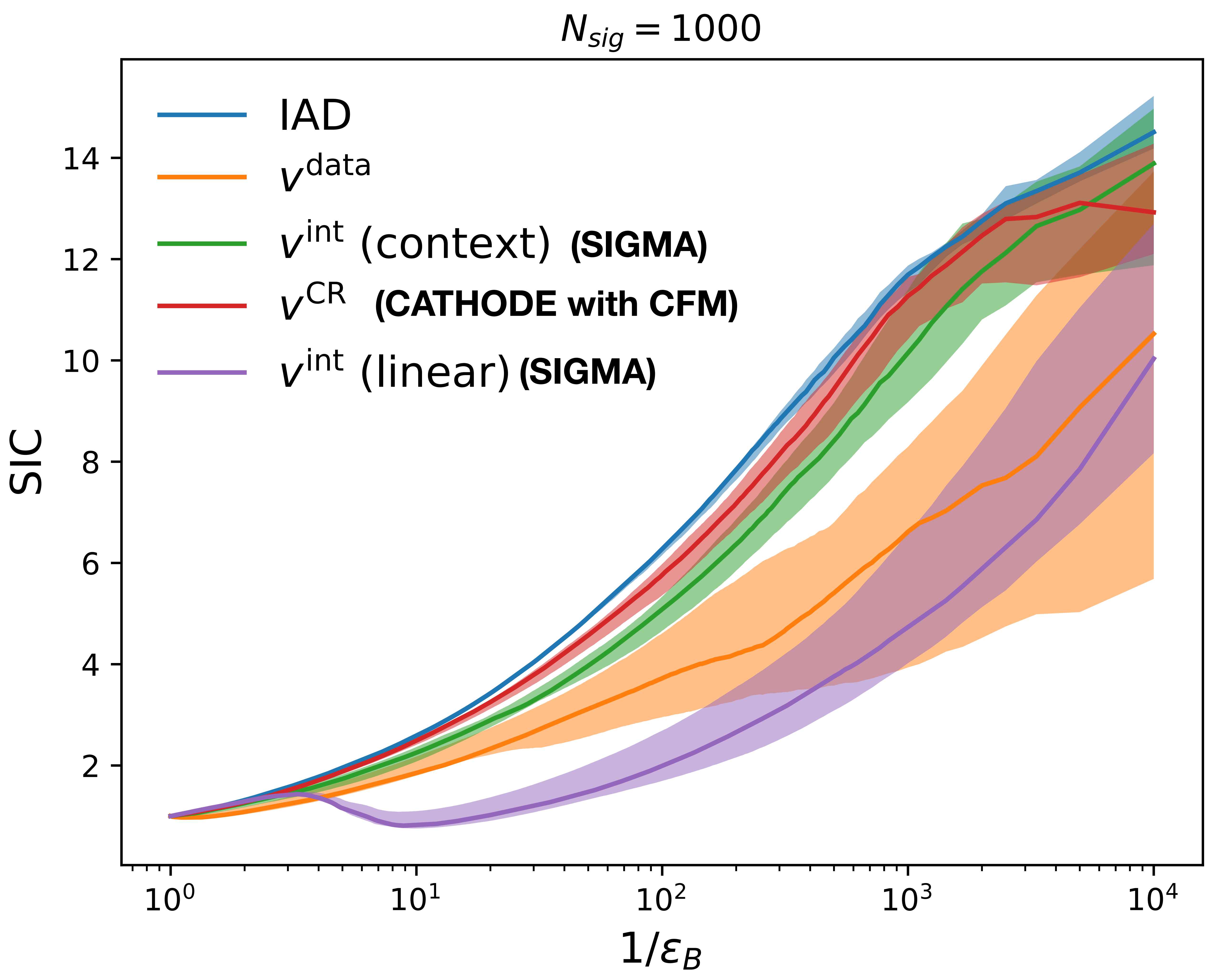
Anomaly detection performance

- v_θ^{data} has worse performance since it learns the signal.



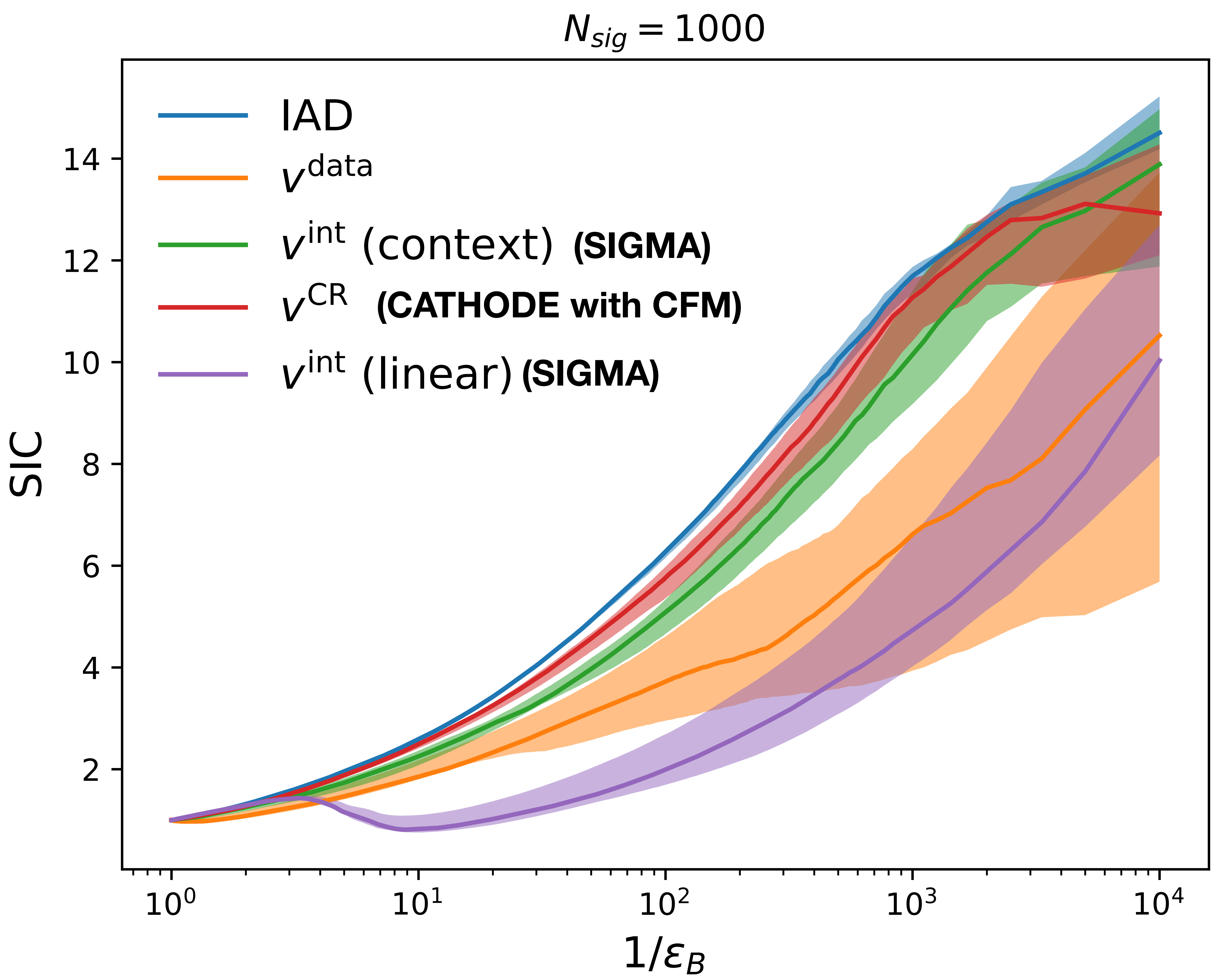
Anomaly detection performance

- v_θ^{data} has worse performance since it learns the signal.
- v_θ^{CR} is slow but has the best performance.



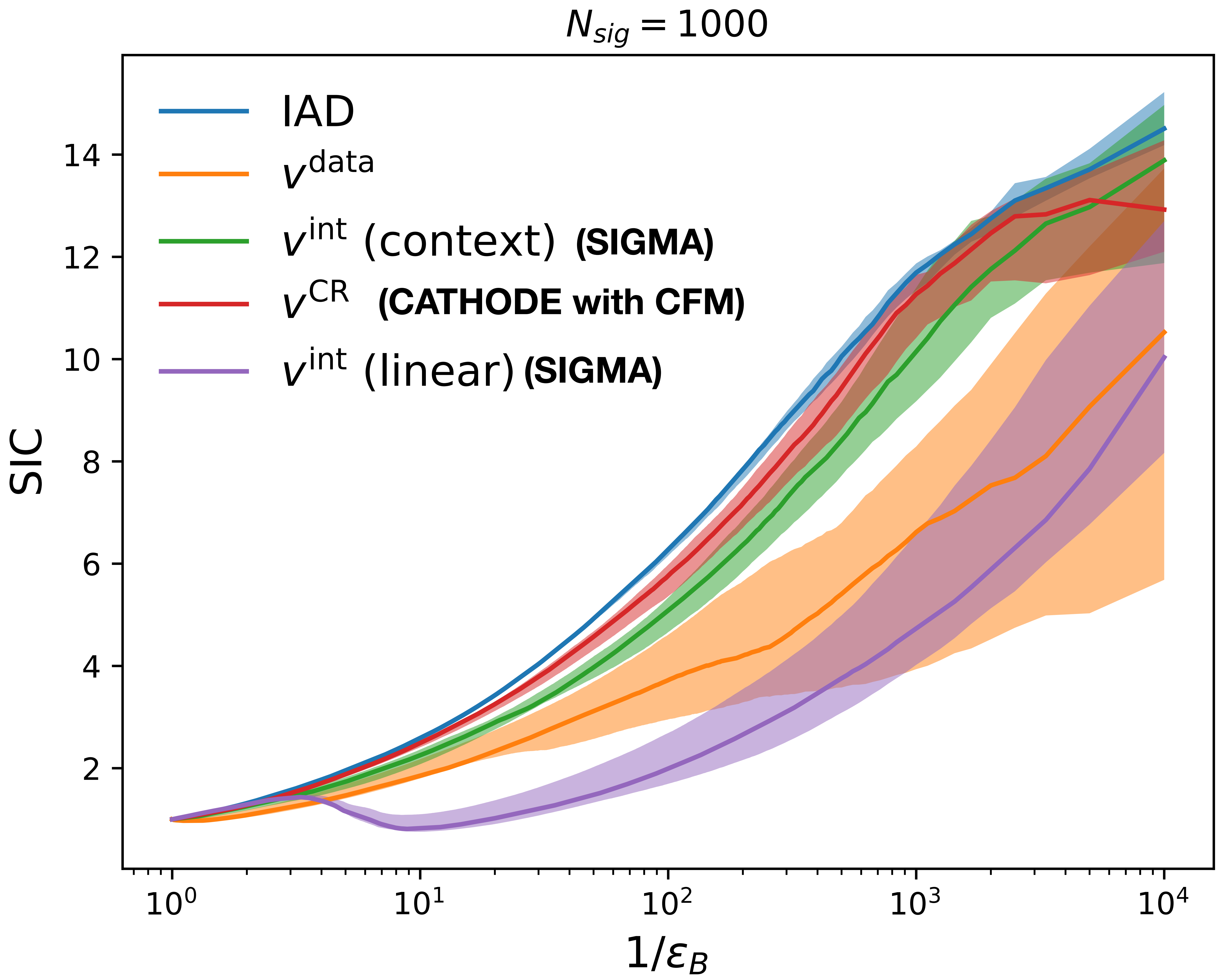
Anomaly detection performance

- v_θ^{data} has worse performance since it learns the signal.
- v_θ^{CR} is slow but has the best performance.
- v_θ^{int}(context) is much faster and has performance similar to v_θ^{CR}.



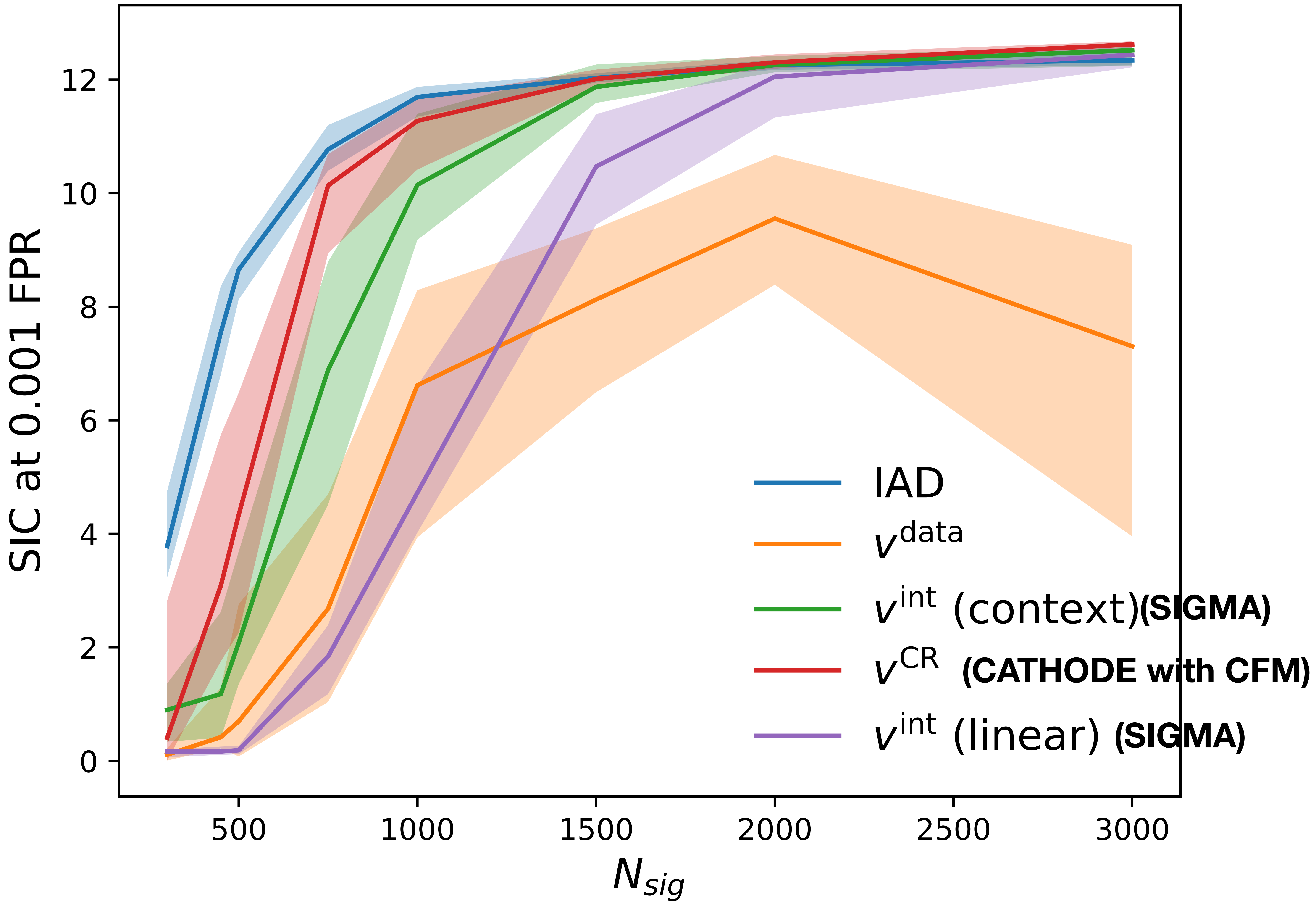
Anomaly detection performance

- v_θ^{data} has worse performance since it learns the signal.
- v_θ^{CR} is slow but has the best performance.
- v_θ^{int}(context) is much faster and has performance similar to v_θ^{CR}.
- v_θ^{int}(context) does better than v_θ^{int}(linear)



Anomaly detection performance

- v_θ^{data} has worse performance since it learns the signal.
- v_θ^{CR} is slow but has the best performance.
- v_θ^{int}(context) is much faster and has performance similar to v_θ^{CR}.
- v_θ^{int}(context) does better than v_θ^{int}(linear)



Timing Comparison

Method	Generative Model	Timing
CATHODE/ANODE	Normalizing Flows	3 hours per SR
CATHODE/ANODE	Flow Matching	30 mins per SR
CURTAINS4F4	Normalizing Flows	3 hours (base model) + 7 mins per SR
RAD-OT	Optimal Transport	10 mins per SR
TRANSIT	No generative model	7 mins per SR
SIGMA (ours)	Flow Matching	30 mins (training) + 30 secs per SR

Summary

Summary

- SIGMA significantly reduces the computational cost relative to previous approaches such as CATHODE, CURTAINS, CURTAINS4F4.

Summary

- SIGMA significantly reduces the computational cost relative to previous approaches such as CATHODE, CURTAINS, CURTAINS4F4.
- Preserves high quality background templates and signal sensitivity

THANK YOU

Backup slides

Timing Comparison

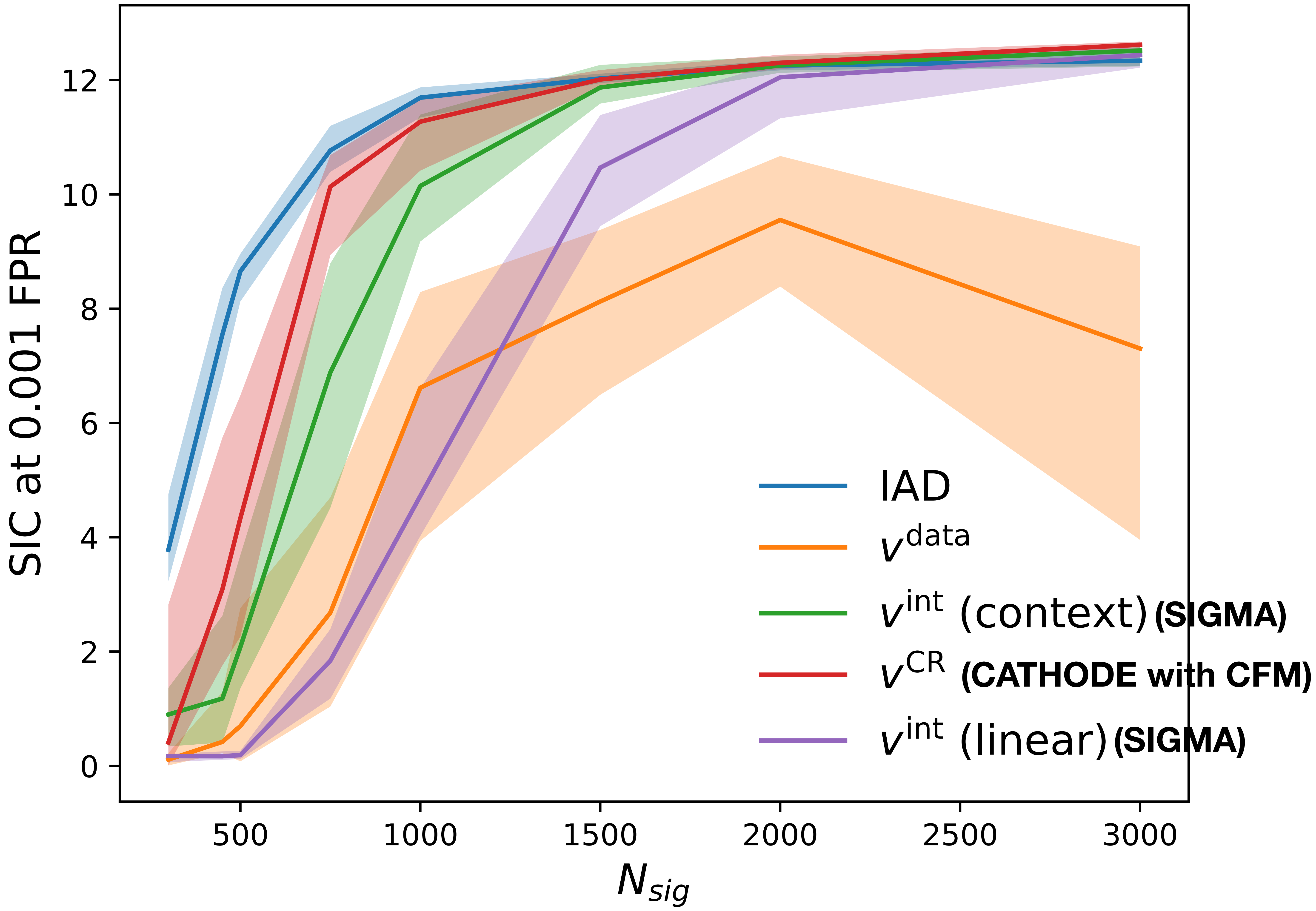
Method	Generative Model	Timing
CATHODE/ANODE	Normalizing Flows	3 hours per SR
CATHODE/ANODE	Flow Matching	30 mins per SR
SIGMA (ours)	Flow Matching	30 mins (training) + 30 secs per SR

How to select best interpolated model?

How to select best interpolated model?

Open question!

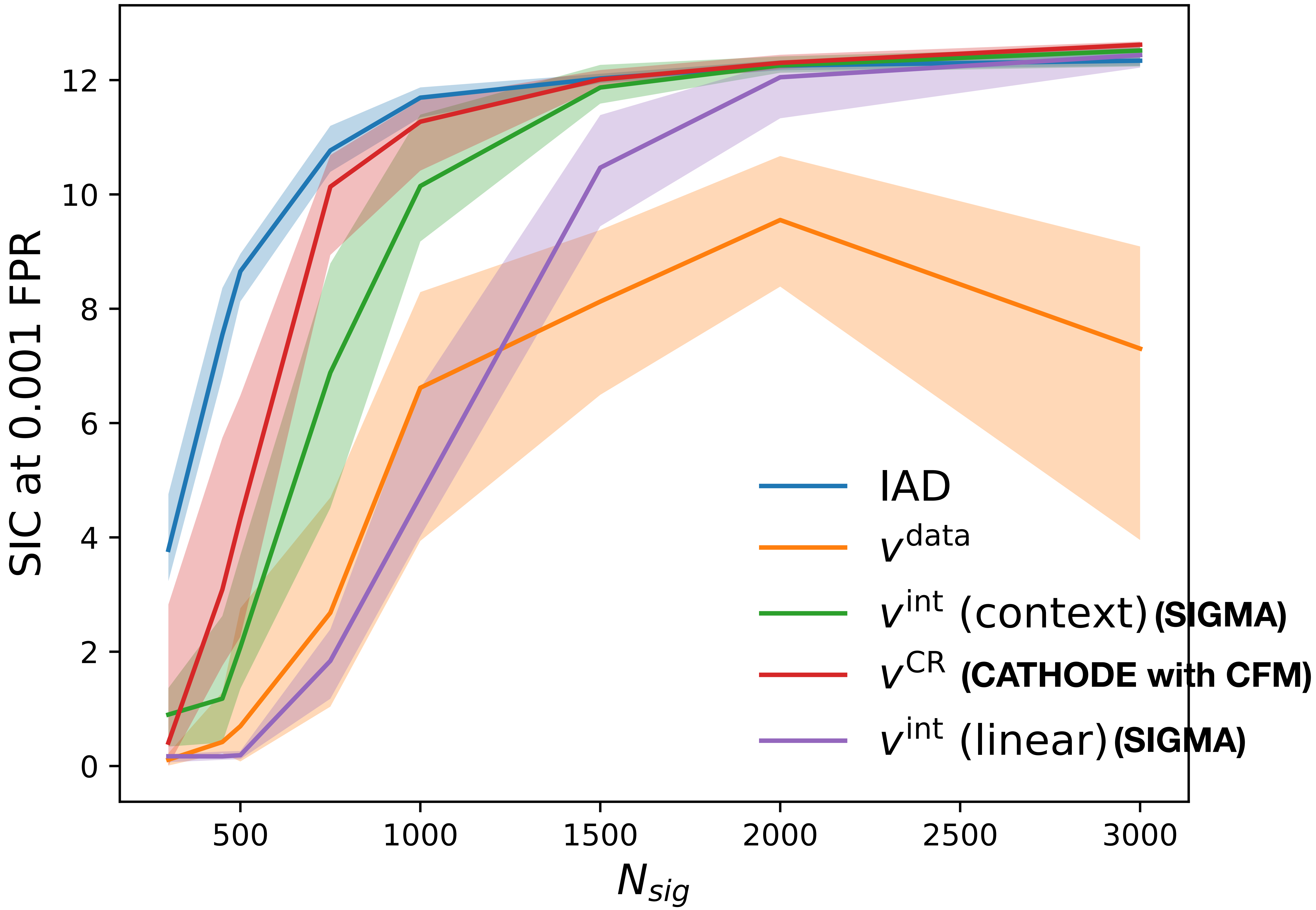
- The SIC is very sensitive to bad background templates.



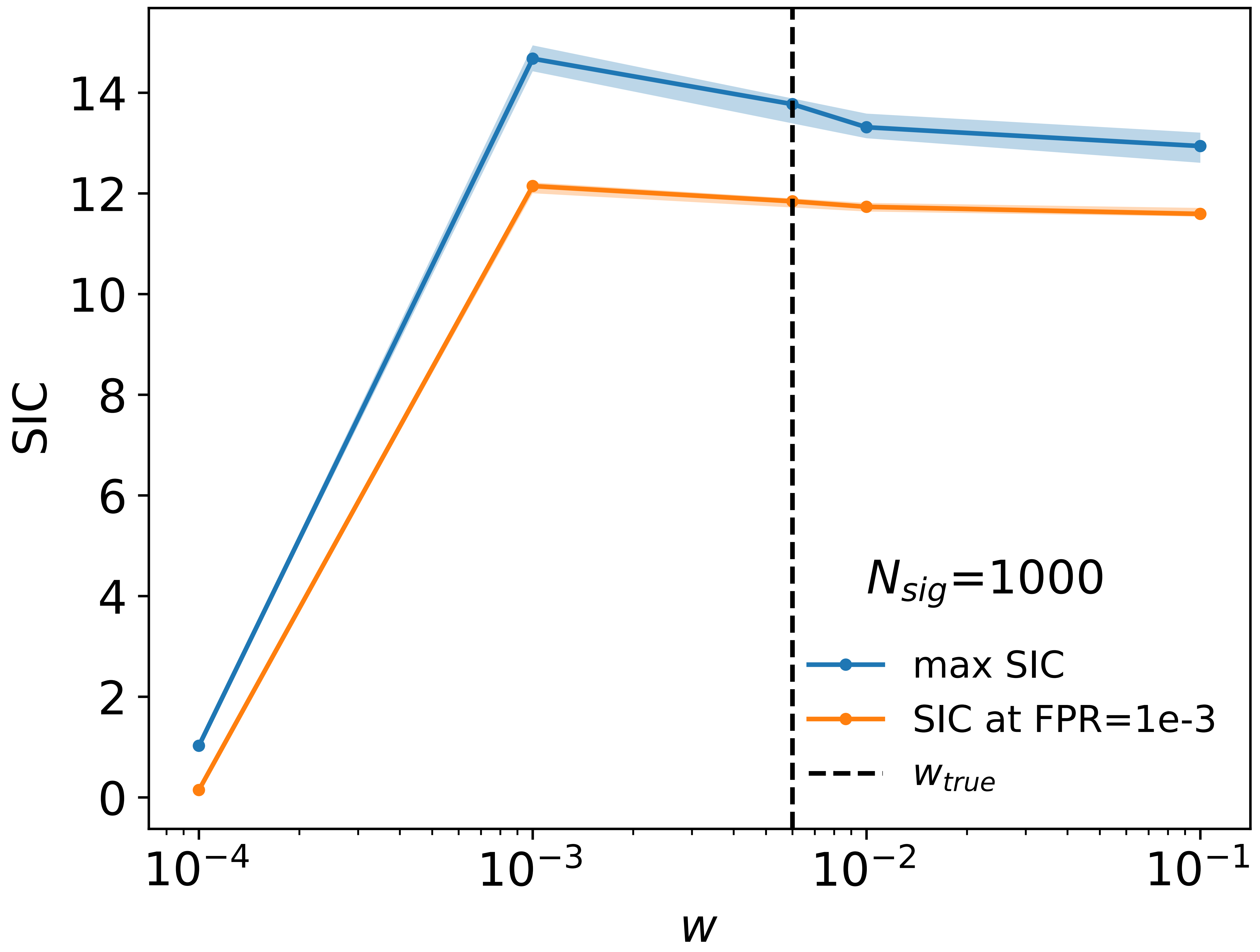
How to select best interpolated model?

Open question!

- The SIC is very sensitive to bad background templates.
- We suggest doing signal injection tests, similar to CMS or ATLAS, or adding artificial gaussian signals to find the best interpolation.



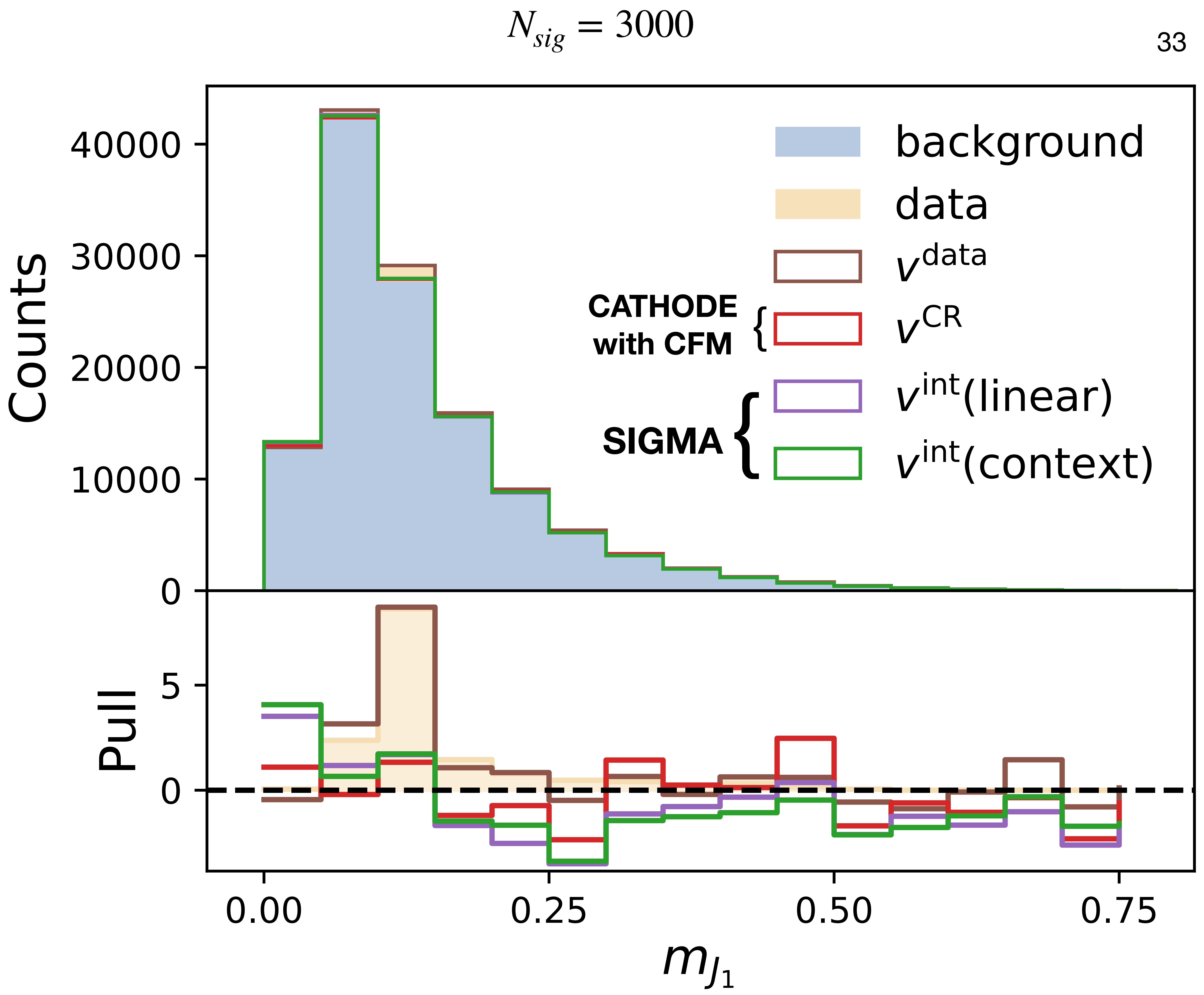
Scanning over *w*



SIC is robust to incorrect choice of ***w***, and could be used to put a lower bound on ***w***

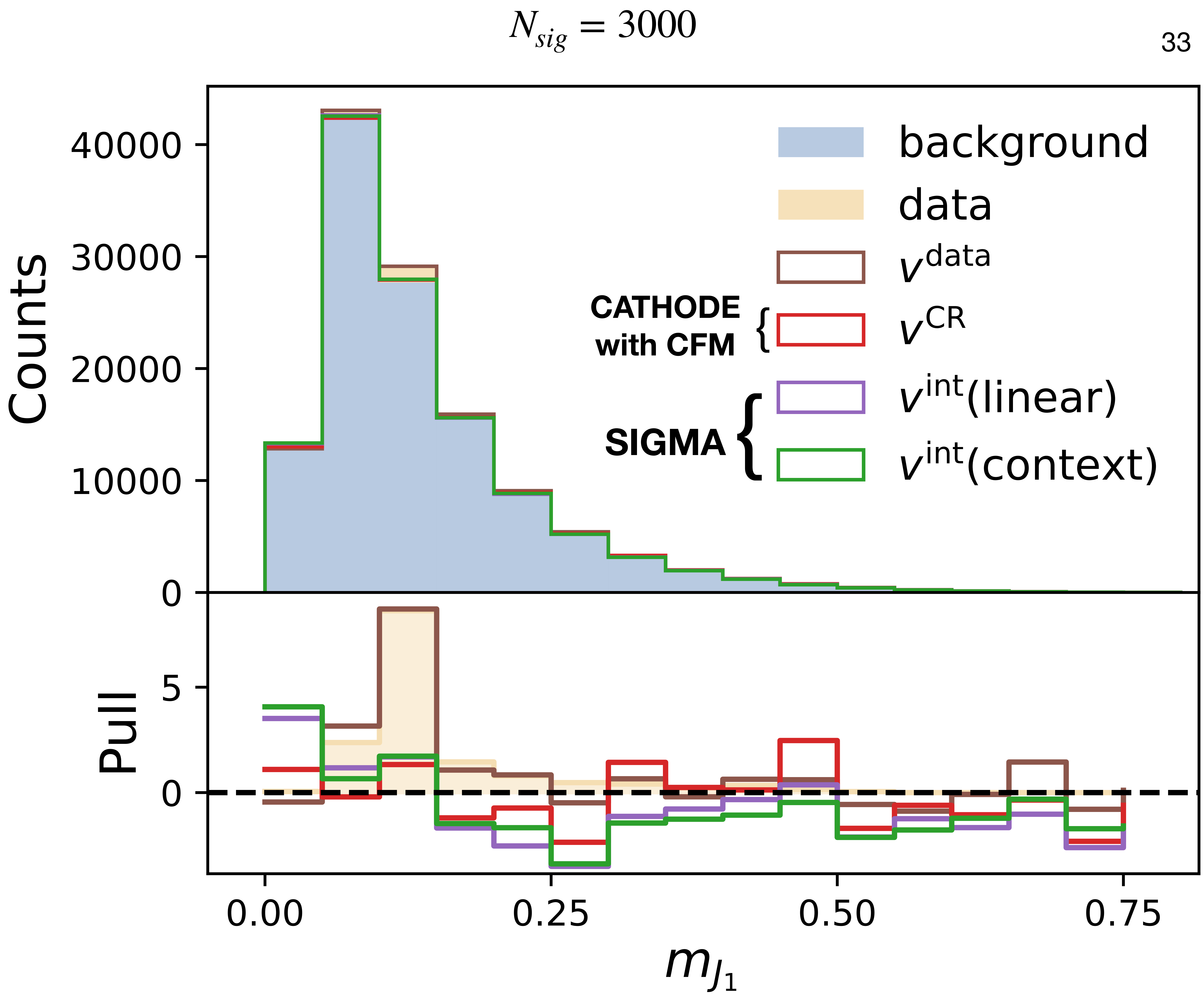
Samples

Samples


$N_{sig} = 3000$
Counts
40000
30000
20000
10000
0
Pull
5
0
0.00
0.25
0.50
0.75
m_{J_1}
background
data
v^{data}
CATHODE with CFM
v^{CR}
SIGMA
v^{int}(linear)
v^{int}(context)

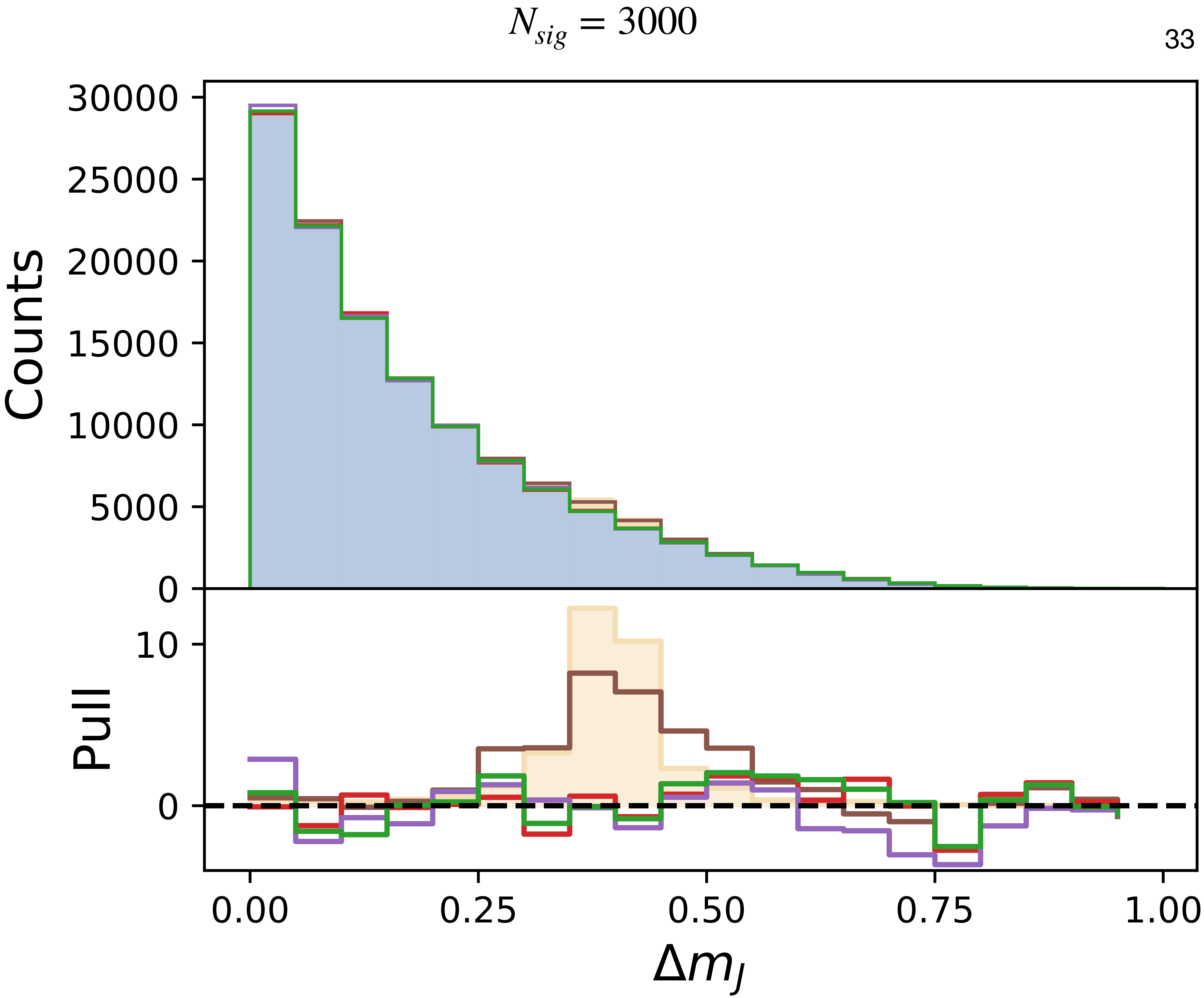
Samples

- The model trained on data, v_θ^{data} learns the signal.
- The previous interpolation method v_θ^{CR} and the new interpolation methods v_θ^{int}(linear) and v_θ^{int} (context) are able to remove the signal



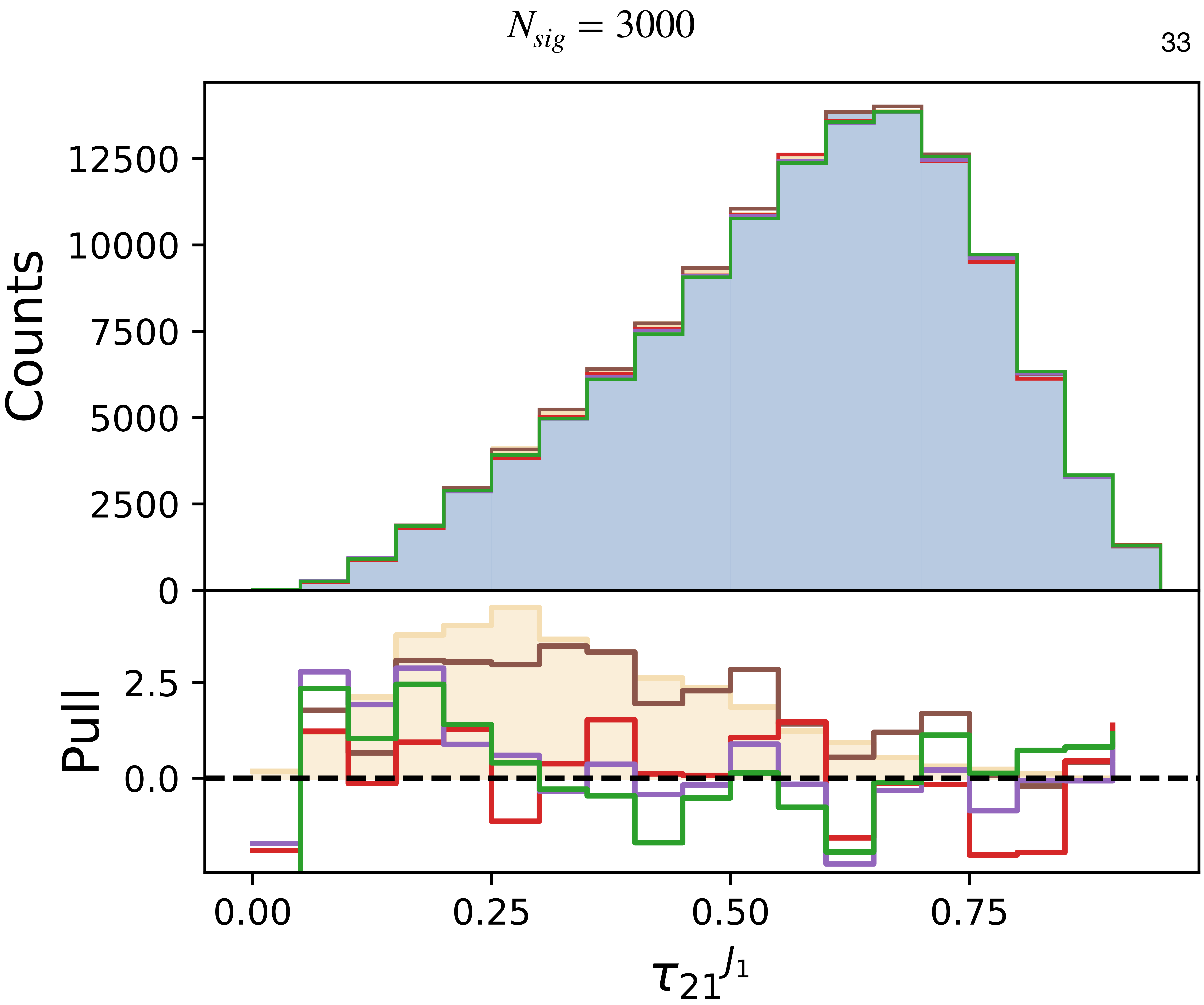
Samples

- The model trained on data, v_θ^{data} learns the signal.
- The previous interpolation method v_θ^{CR} and the new interpolation methods v_θ^{int}(linear) and v_θ^{int} (context) are able to remove the signal



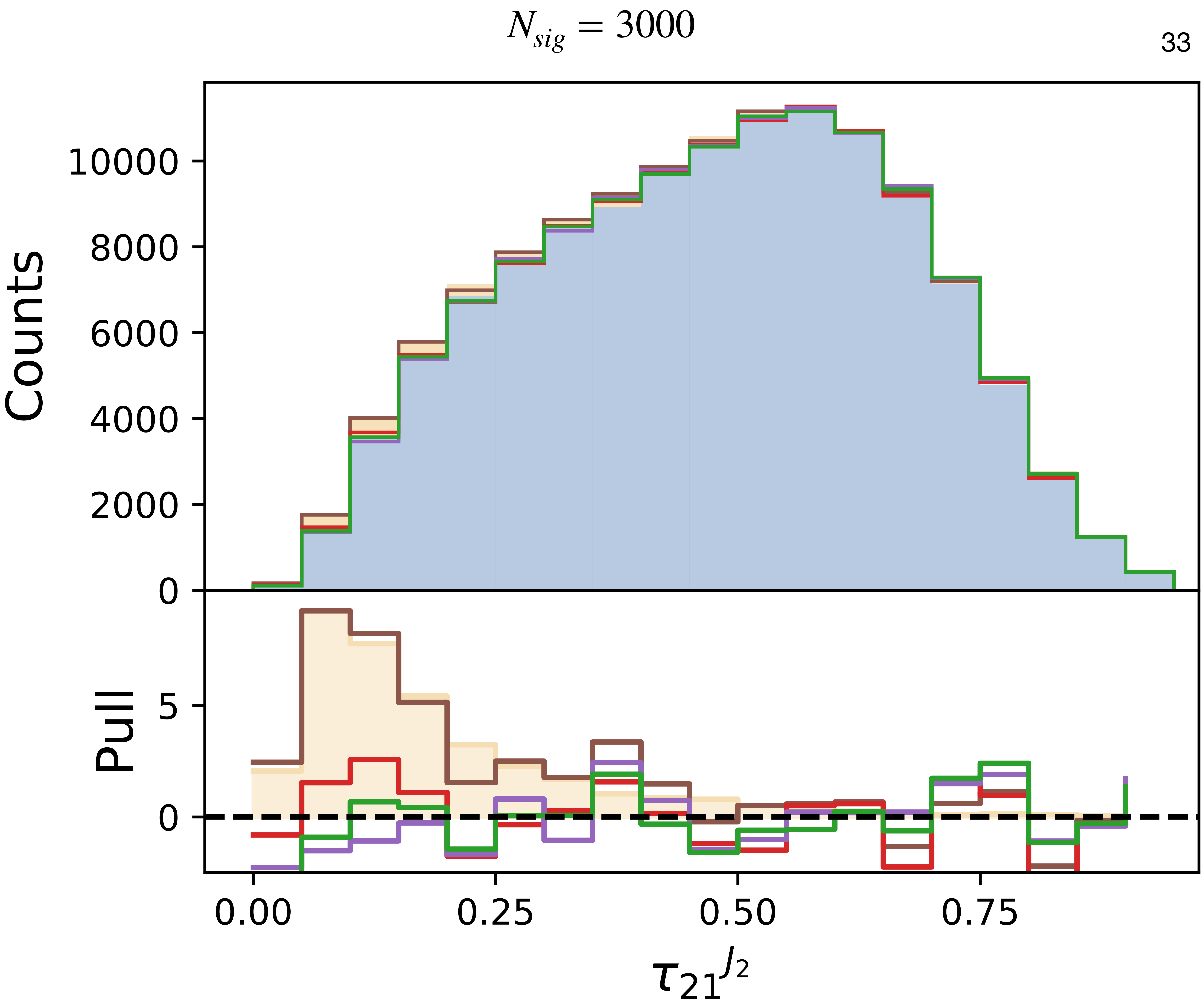
Samples

- The model trained on data, v_θ^{data} learns the signal.
- The previous interpolation method v_θ^{CR} and the new interpolation methods v_θ^{int}(linear) and v_θ^{int} (context) are able to remove the signal



Samples

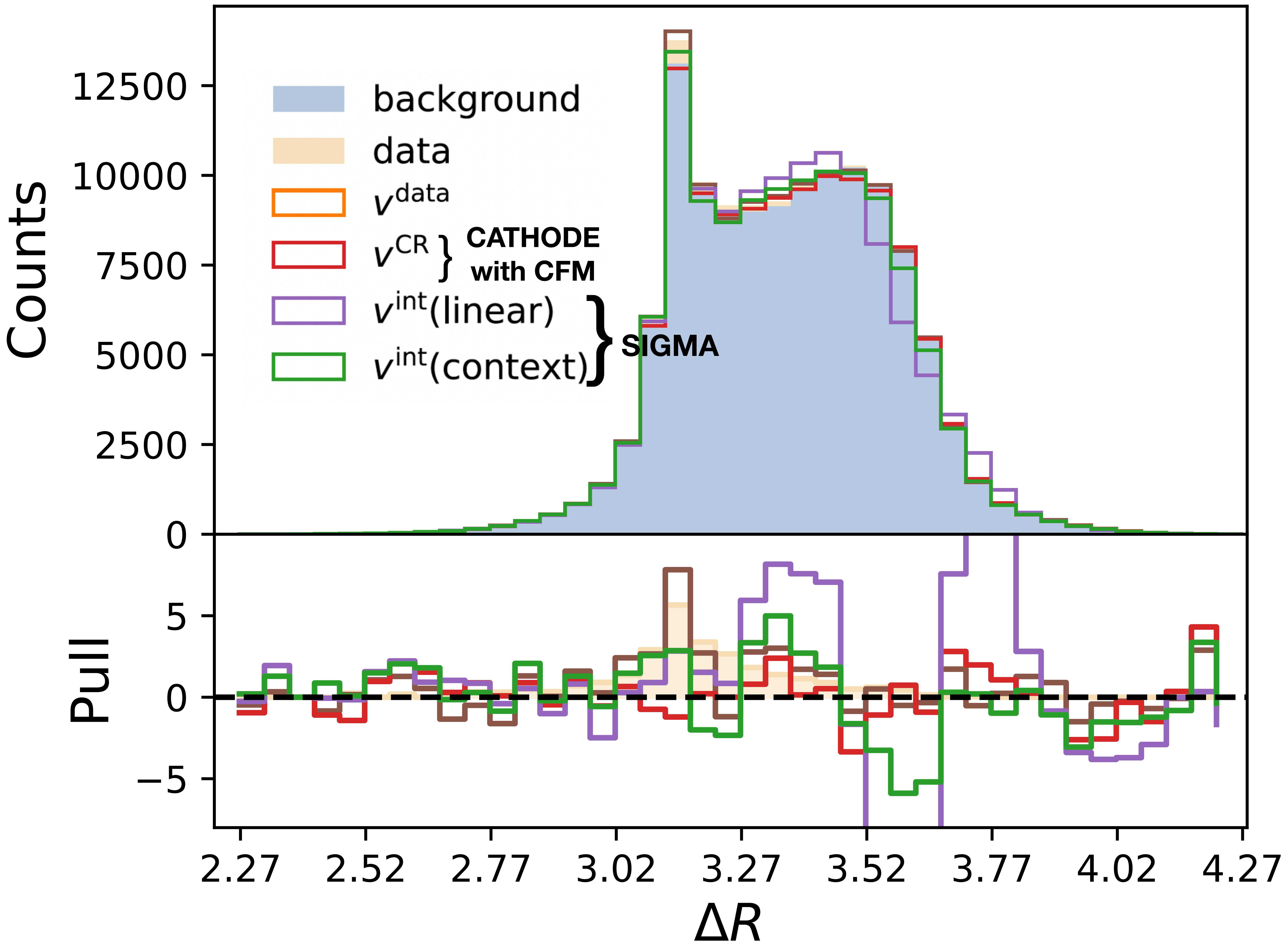
- The model trained on data, v_θ^{data} learns the signal.
- The previous interpolation method v_θ^{CR} and the new interpolation methods v_θ^{int}(linear) and v_θ^{int} (context) are able to remove the signal



Samples

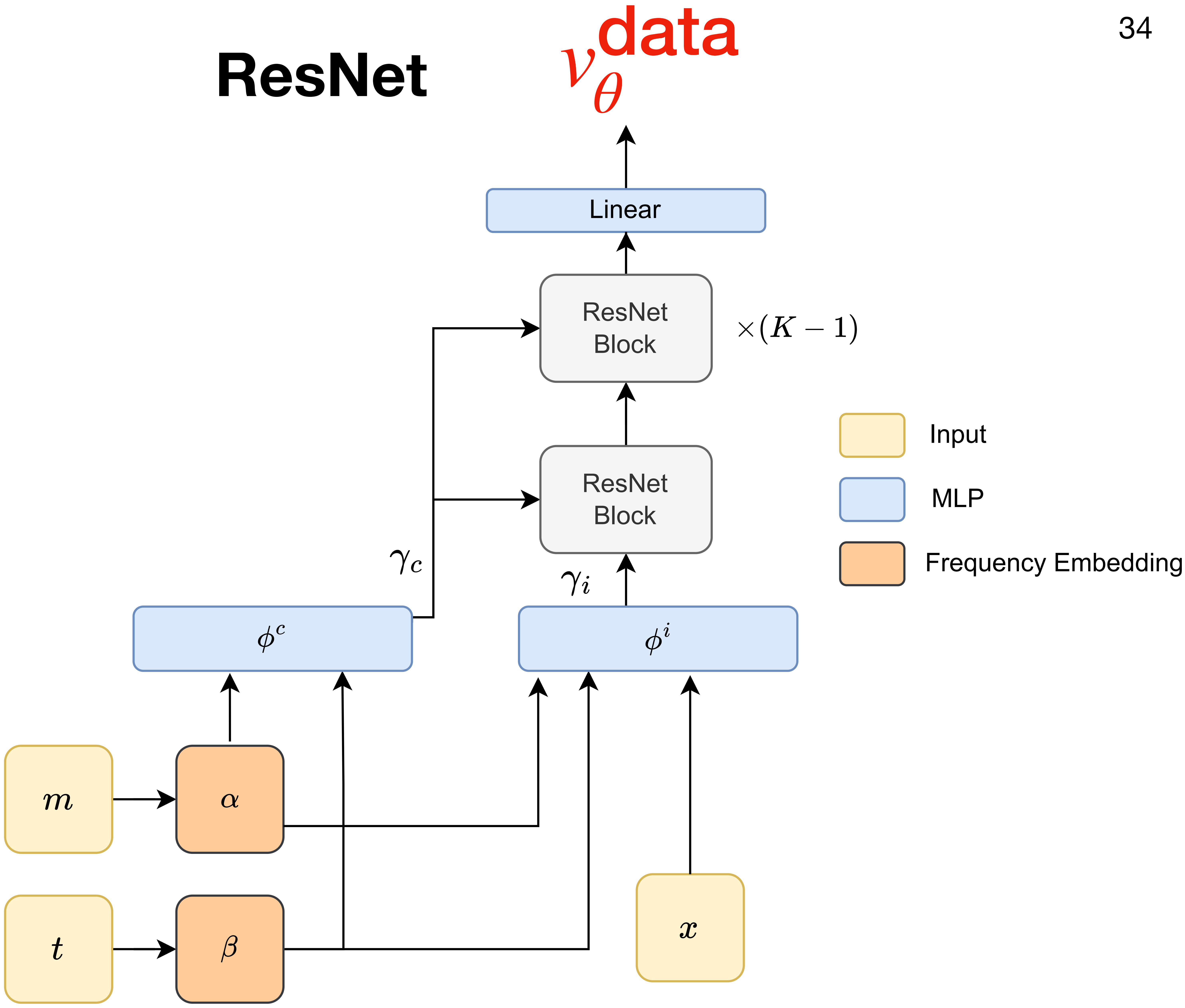
- ΔR is strongly correlated with m.
- v_θ^{int}(context) learns this better than v_θ^{int}(linear).

$N_{sig} = 3000$

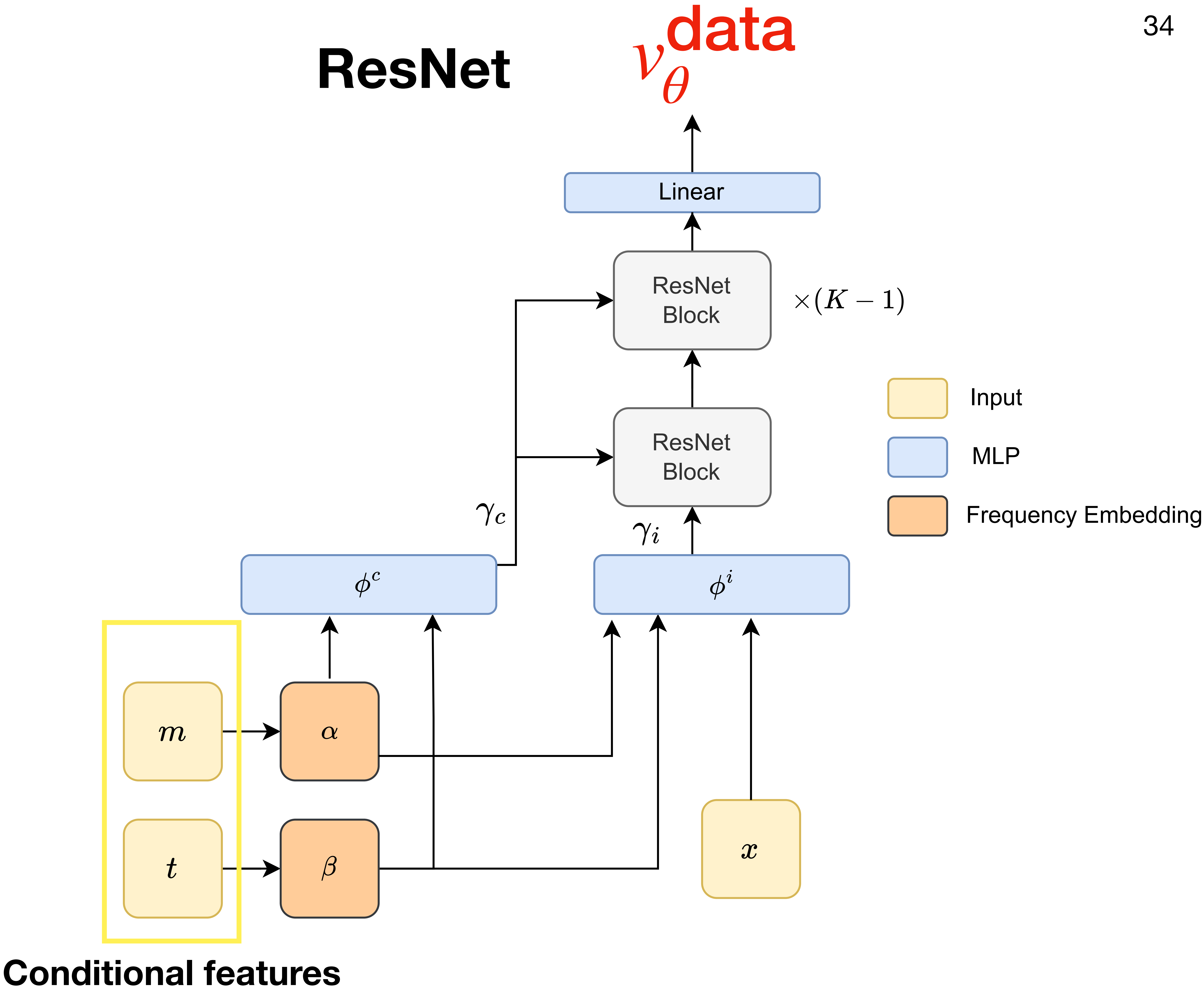


Architecture

Architecture



Architecture

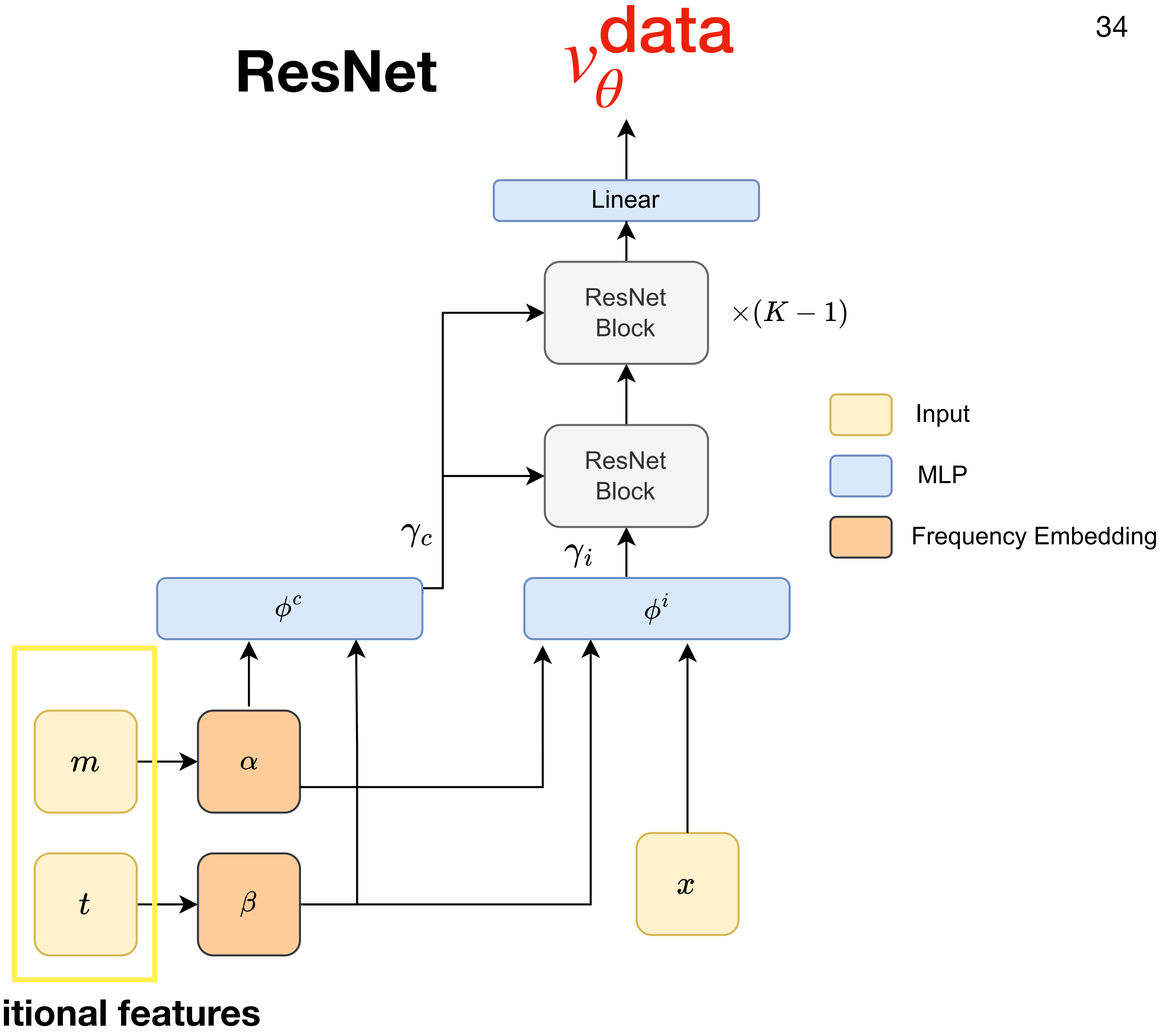


Architecture

To learn the full data distribution optimally, including the more localized, higher frequency modes corresponding to signal, we found that a frequency embedding for ***m*** was beneficial.

$$\alpha(m) = \big(\sin(2^0\pi m), \cos(2^0\pi m), \ldots, \sin(2^{L-1}\pi m), \cos(2^{L-1}\pi m)\big)$$

$$\beta(t) = \big(\sin(\pi t), \cos(\pi t), \ldots, \sin((L'+1)\pi t), \cos((L'+1)\pi t)\big)$$



NERF: arXiv:2003.08934

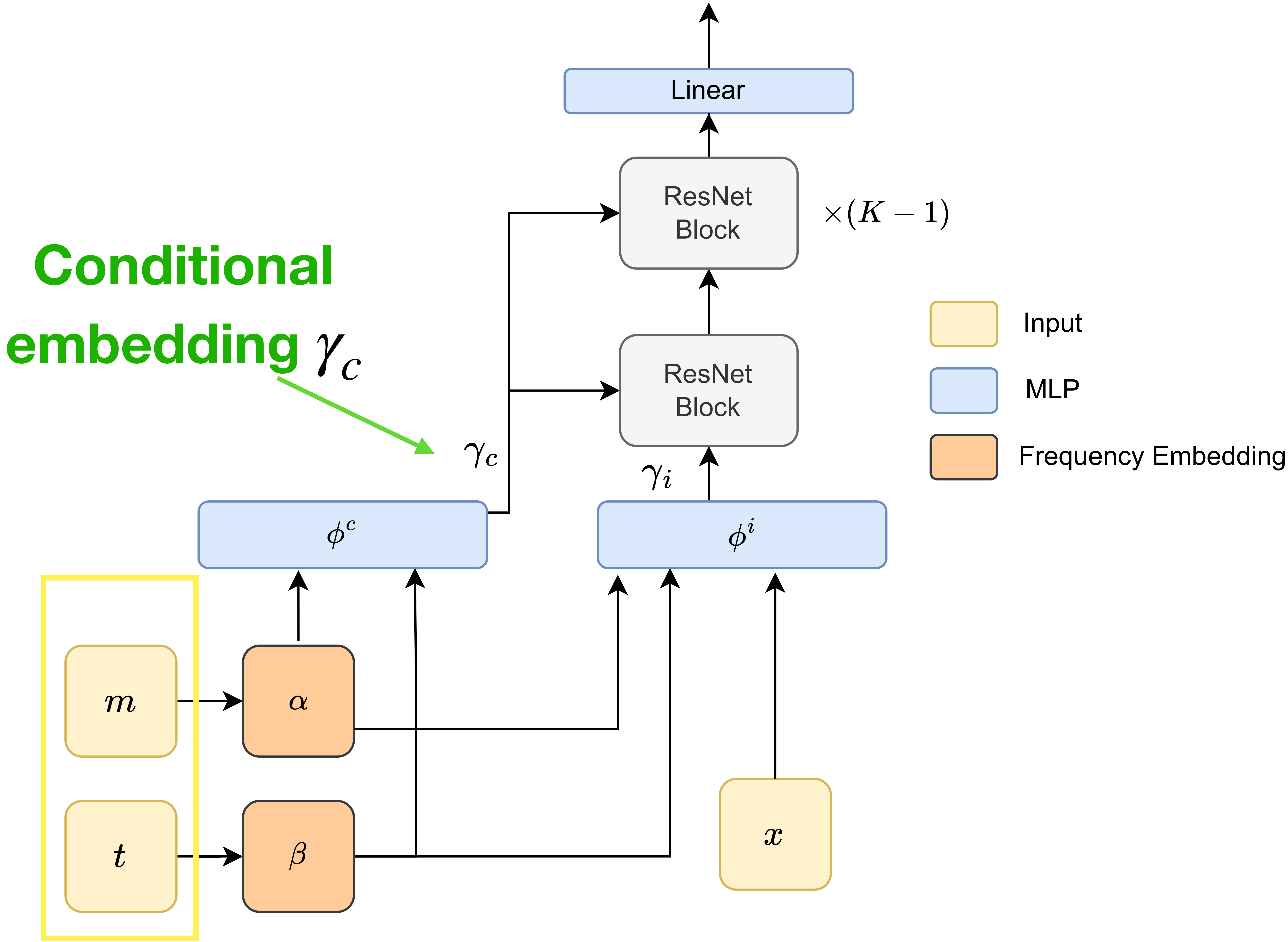
Architecture

To learn the full data distribution optimally, including the more localized, higher frequency modes corresponding to signal, we found that a frequency embedding for ***m*** was beneficial.

$$\alpha(m) = \big(\sin(2^0\pi m), \cos(2^0\pi m), \ldots, \sin(2^{L-1}\pi m), \cos(2^{L-1}\pi m)\big)$$

$$\beta(t) = \big(\sin(\pi t), \cos(\pi t), \ldots, \sin((L'+1)\pi t), \cos((L'+1)\pi t)\big)$$

ResNet v_θ^{data}



Conditional features

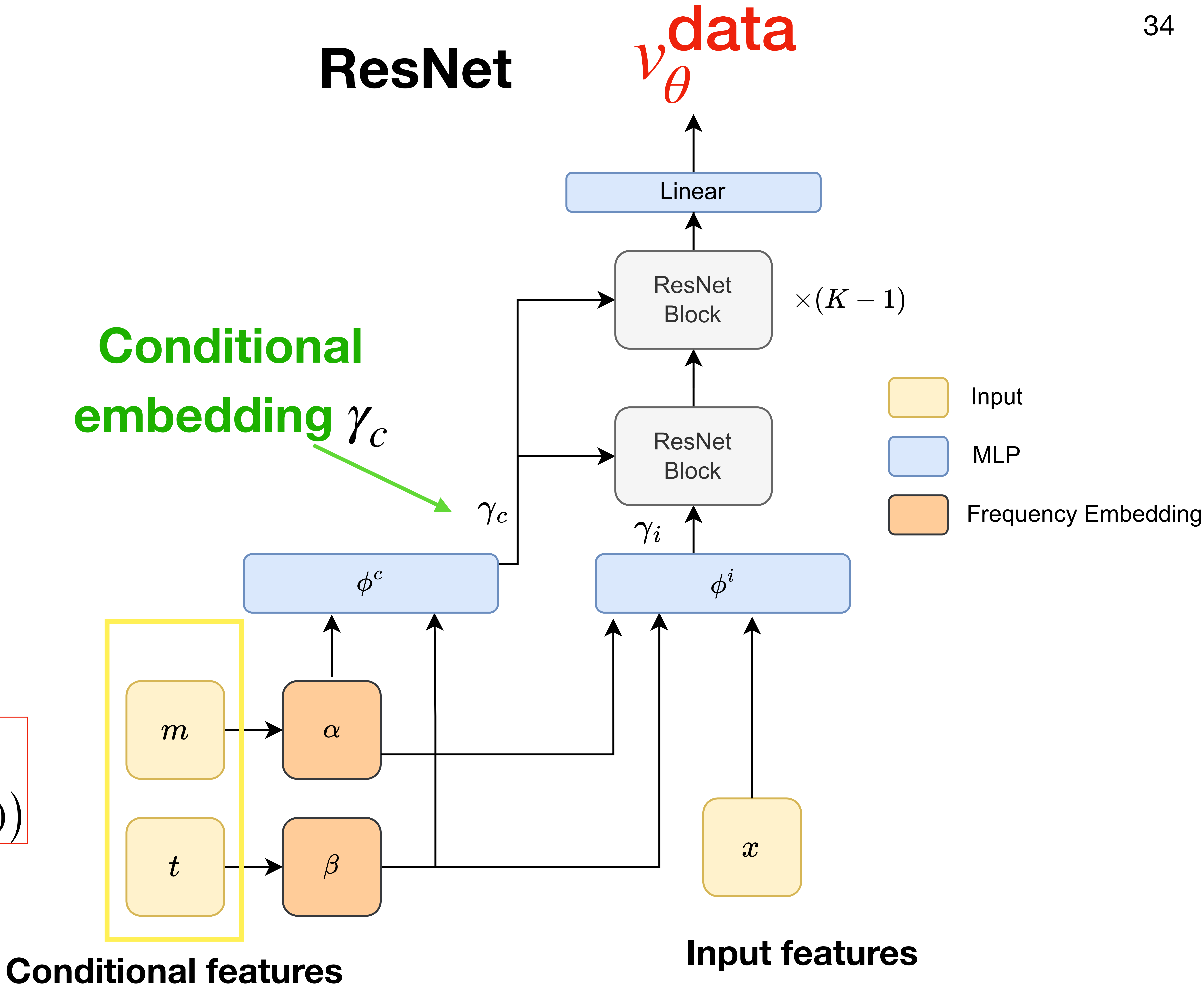
NERF: arXiv:2003.08934

Architecture

To learn the full data distribution optimally, including the more localized, higher frequency modes corresponding to signal, we found that a frequency embedding for ***m*** was beneficial.

$$\alpha(m) = \big(\sin(2^0\pi m), \cos(2^0\pi m), \ldots, \sin(2^{L-1}\pi m), \cos(2^{L-1}\pi m)\big)$$

$$\beta(t) = \big(\sin(\pi t), \cos(\pi t), \ldots, \sin((L'+1)\pi t), \cos((L'+1)\pi t)\big)$$



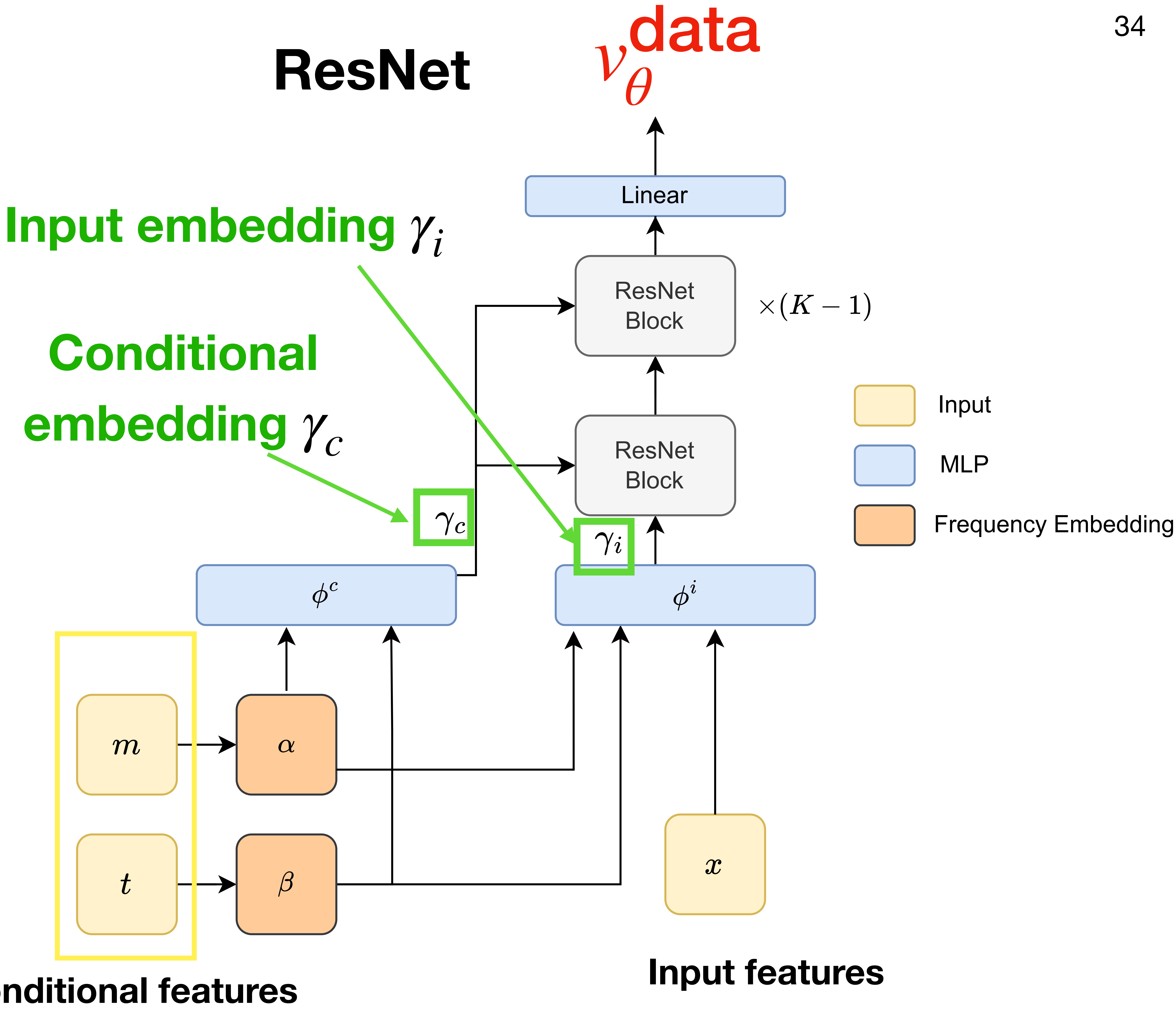
NERF: arXiv:2003.08934

Architecture

To learn the full data distribution optimally, including the more localized, higher frequency modes corresponding to signal, we found that a frequency embedding for ***m*** was beneficial.

$$\alpha(m) = \big(\sin(2^0\pi m), \cos(2^0\pi m), \ldots, \sin(2^{L-1}\pi m), \cos(2^{L-1}\pi m)\big)$$

$$\beta(t) = \big(\sin(\pi t), \cos(\pi t), \ldots, \sin((L'+1)\pi t), \cos((L'+1)\pi t)\big)$$



NERF: [arXiv:2003.08934](https://arxiv.org/abs/2003.08934)