SIGMA: Single Interpolated Generative Model for Anomalies

Based on arXiv:2410.20537

Ranit Das and David Shih

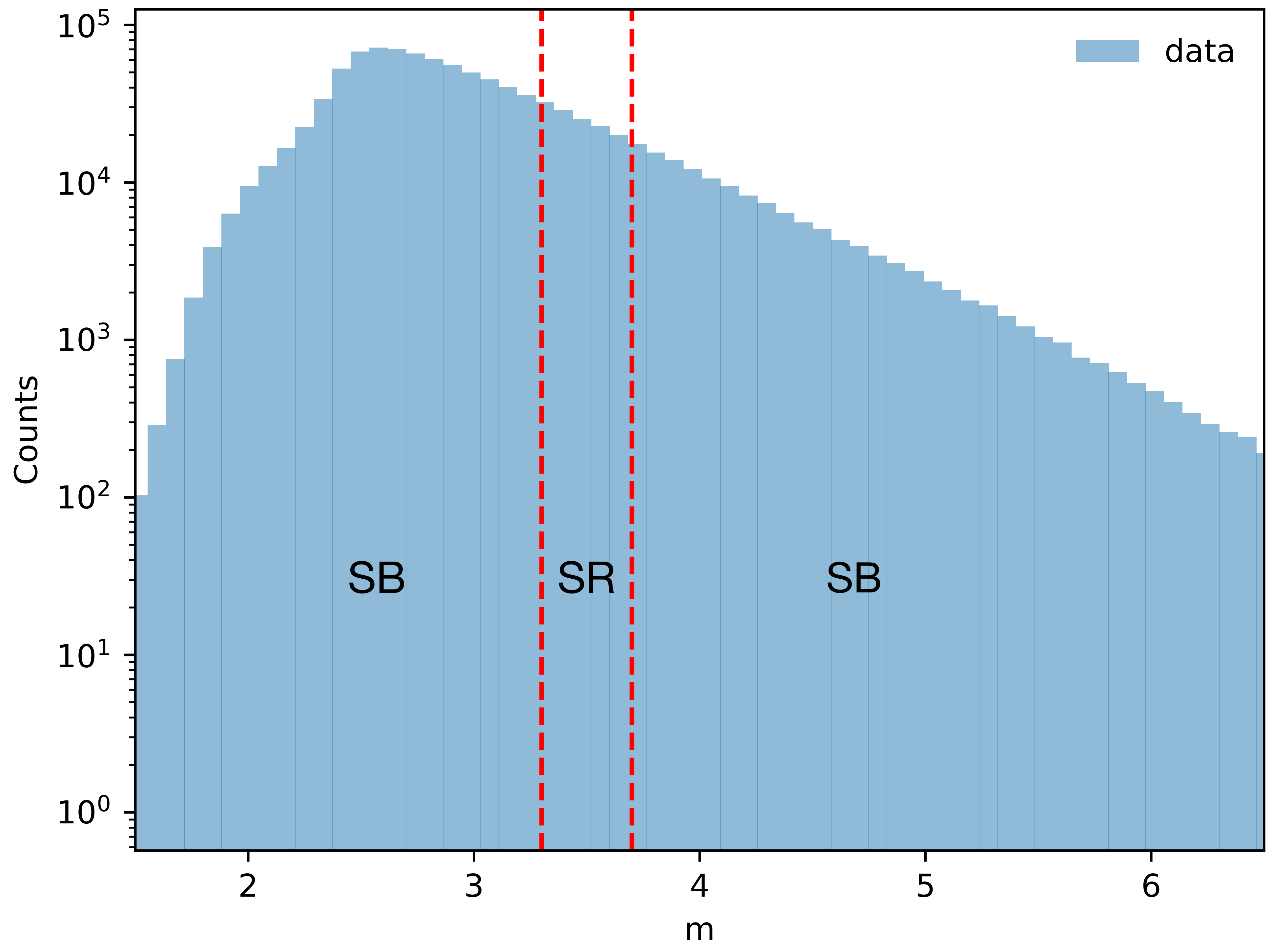
RUTGERS

ML4Jets2024

07-11-2024

Data Driven Resonant Anomaly Detection with background interpolation

Key Steps:

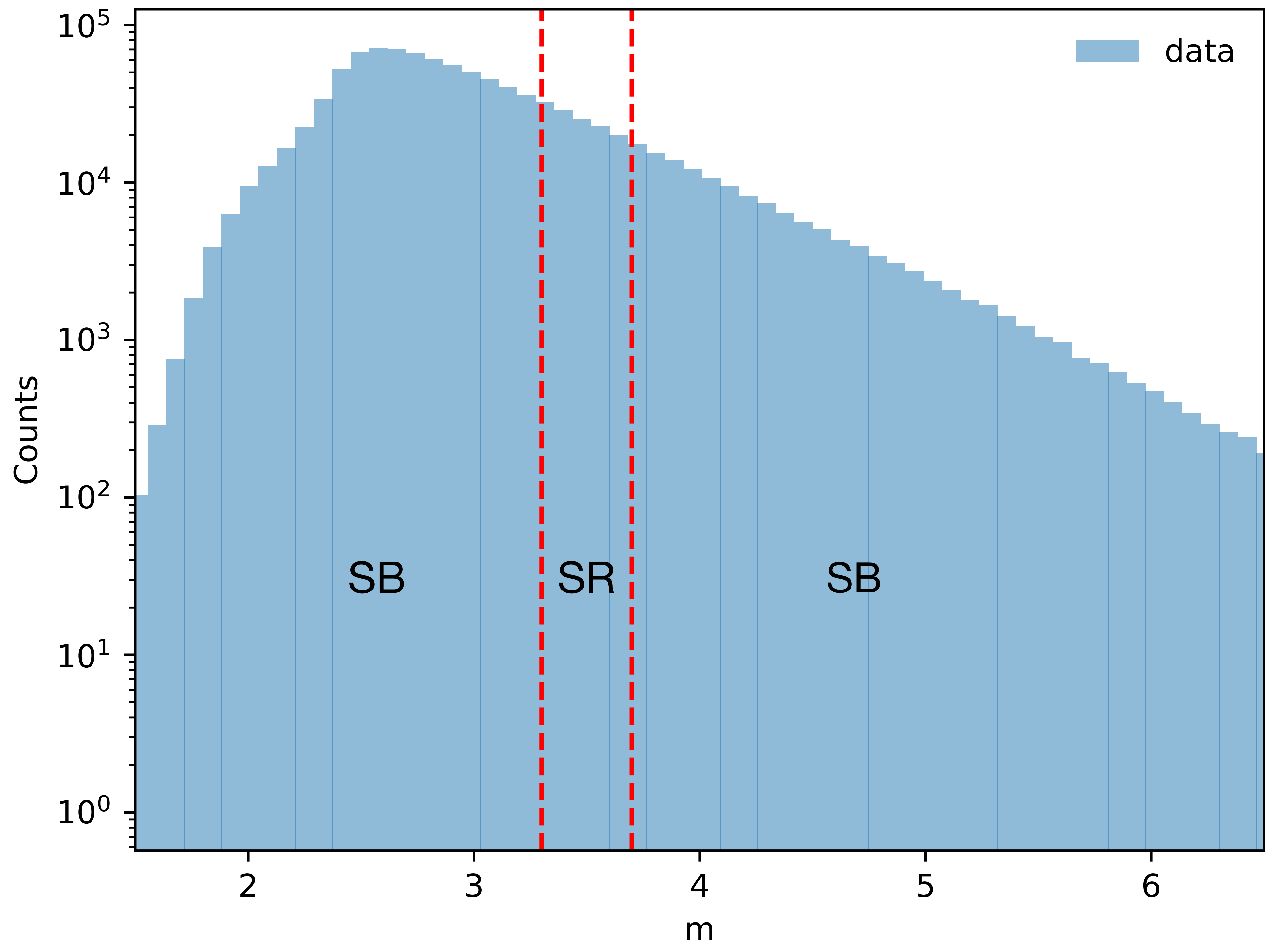


ANODE: arXiv:2001.04990v2
CATHODE: arXiv:2109.00546v3
CURTAINS: arXiv:2203.09470v3
R-ANODE: arXiv:2312.11629

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***.



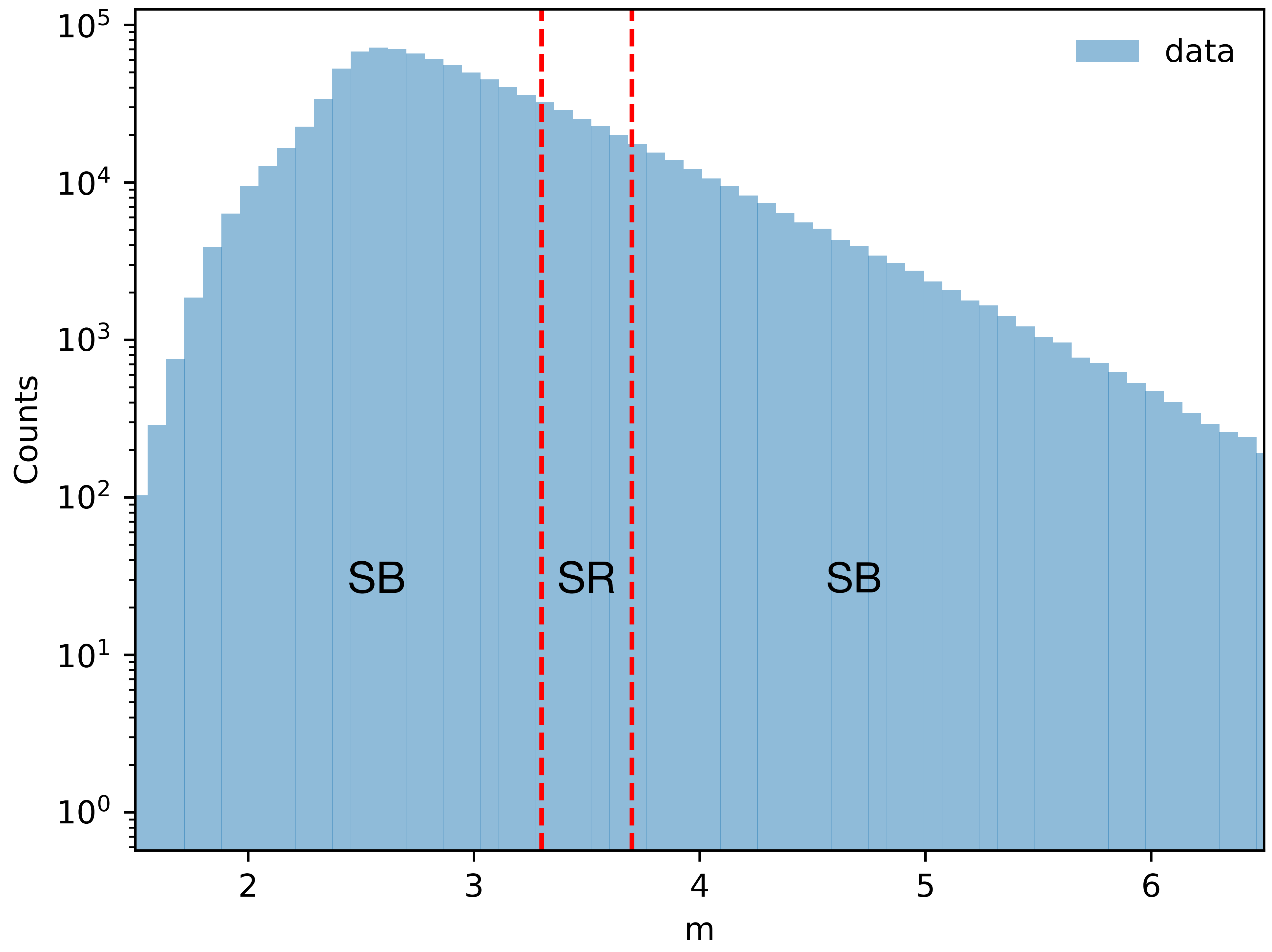
ANODE: arXiv:2001.04990v2
CATHODE: arXiv:2109.00546v3
CURTAINS: arXiv:2203.09470v3
R-ANODE: arXiv:2312.11629

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***.

- 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
R-ANODE: arXiv:2312.11629

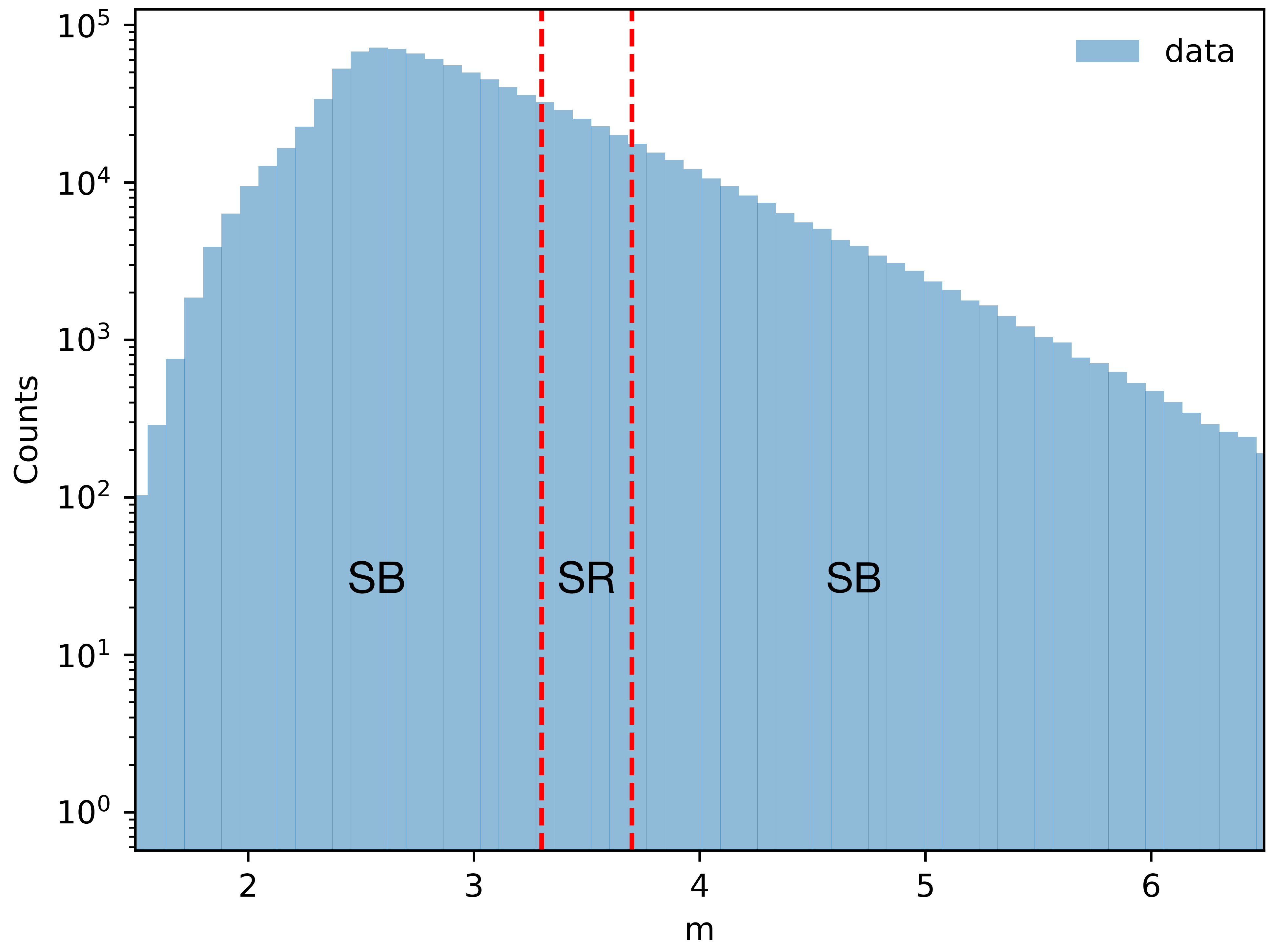
Data Driven Resonant Anomaly Detection with background interpolation

Key Steps:

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

- 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

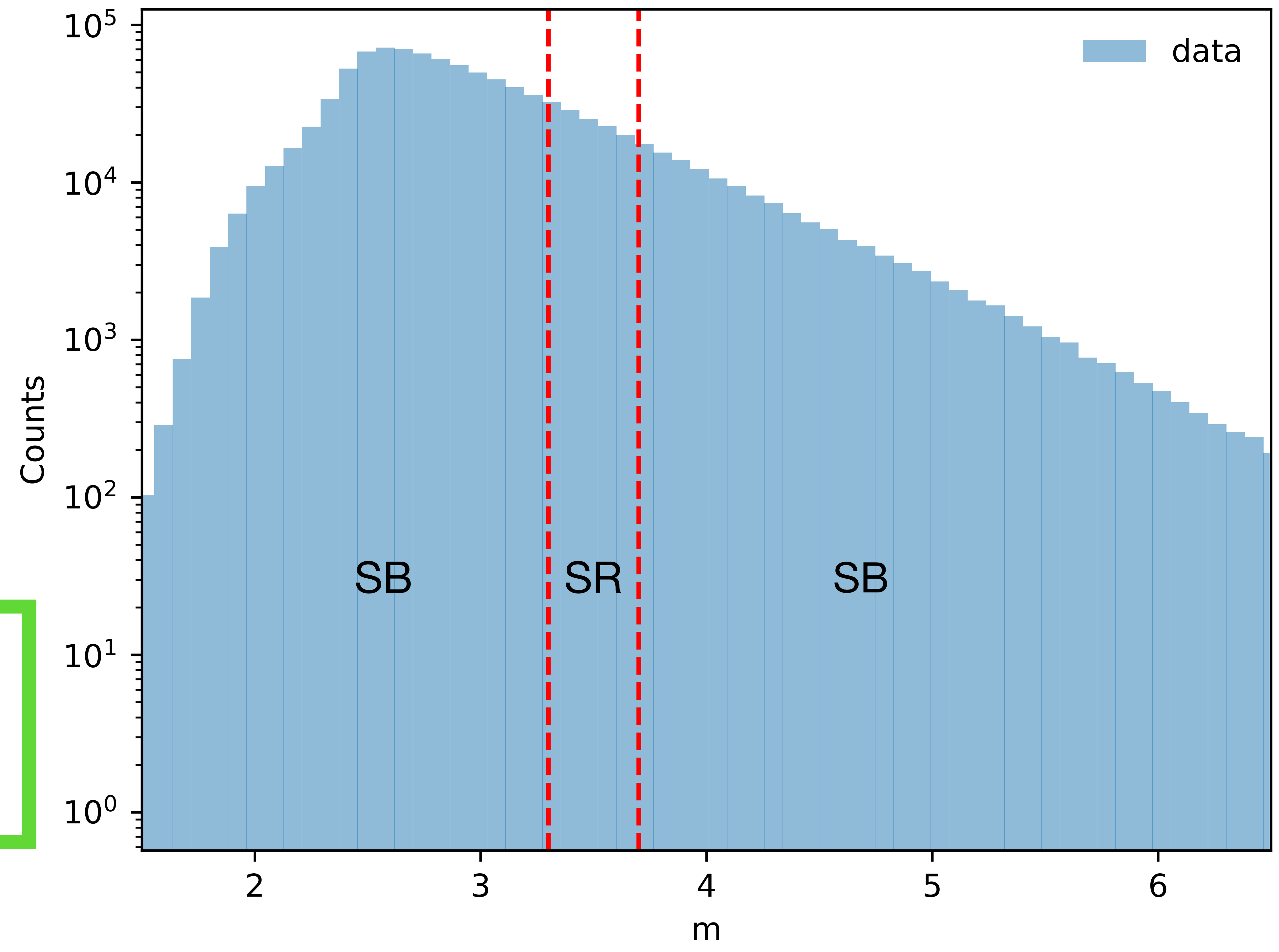
Key Steps:

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

This 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, 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:

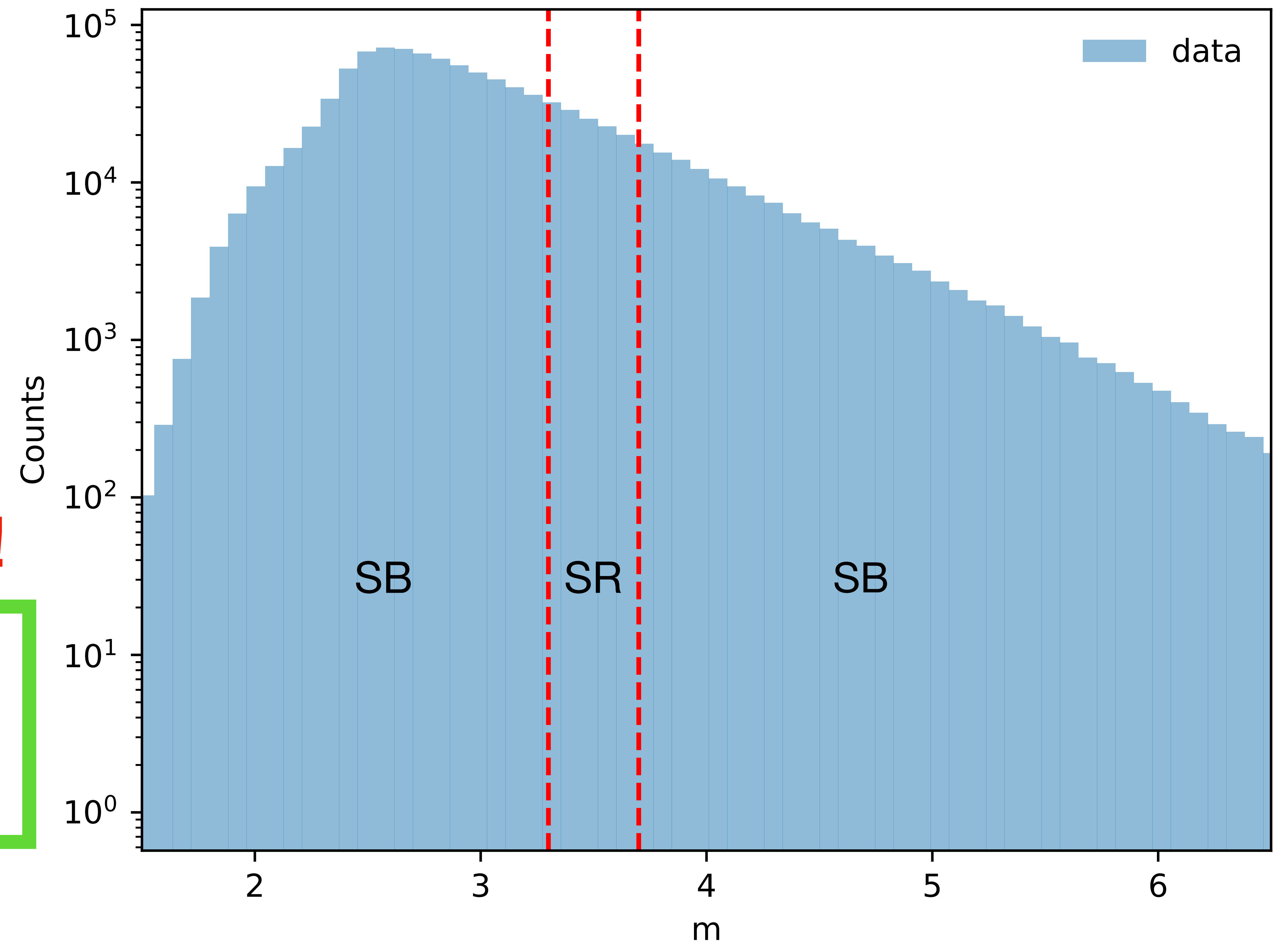
- Define different Signal Regions(SR) and Side-Band Regions(SB) using a resonant feature ***m***.

This talk!

Problem: Computationally expensive!

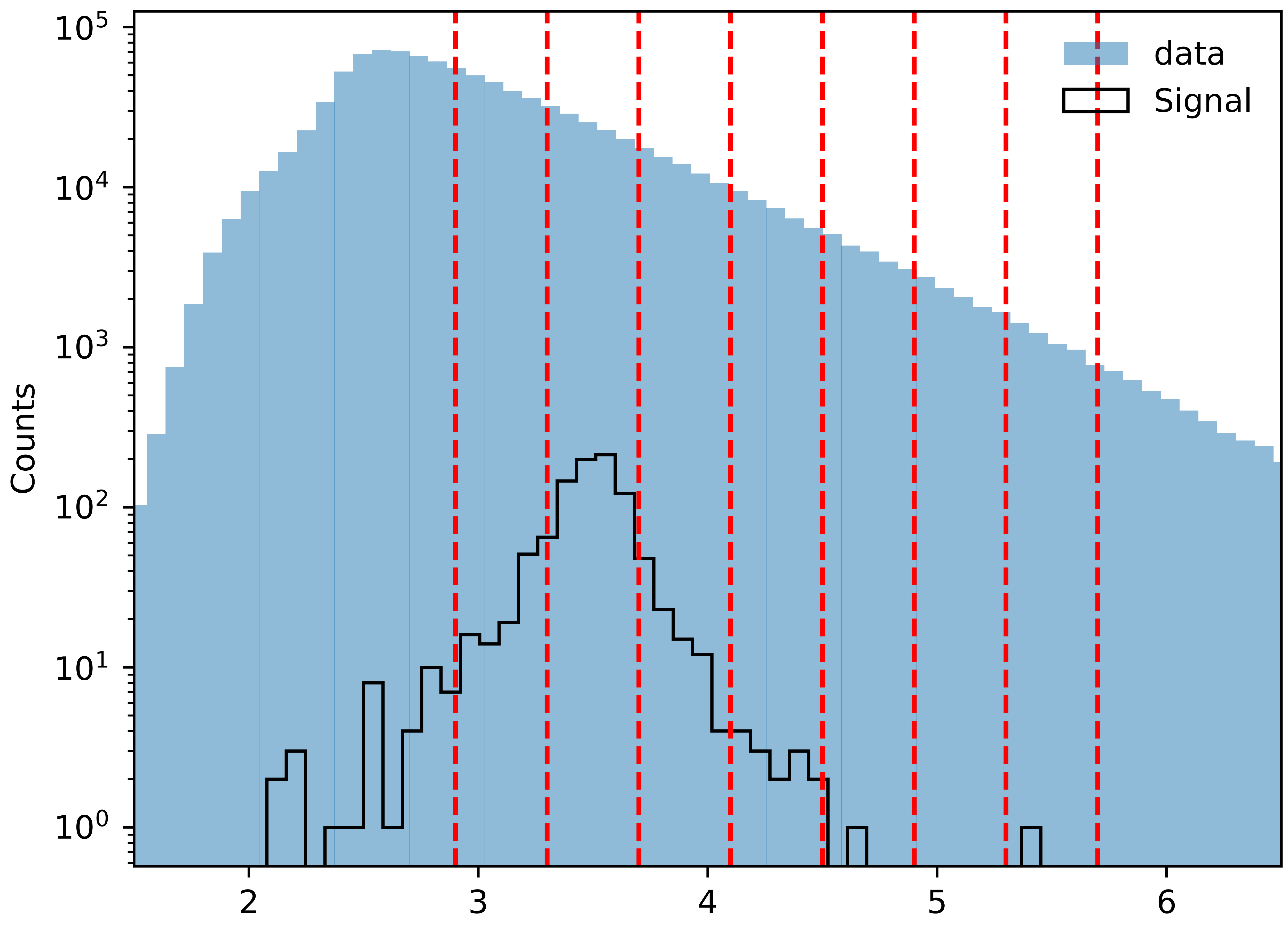
- For each SR, generate a background template from SB and interpolated into SR.

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



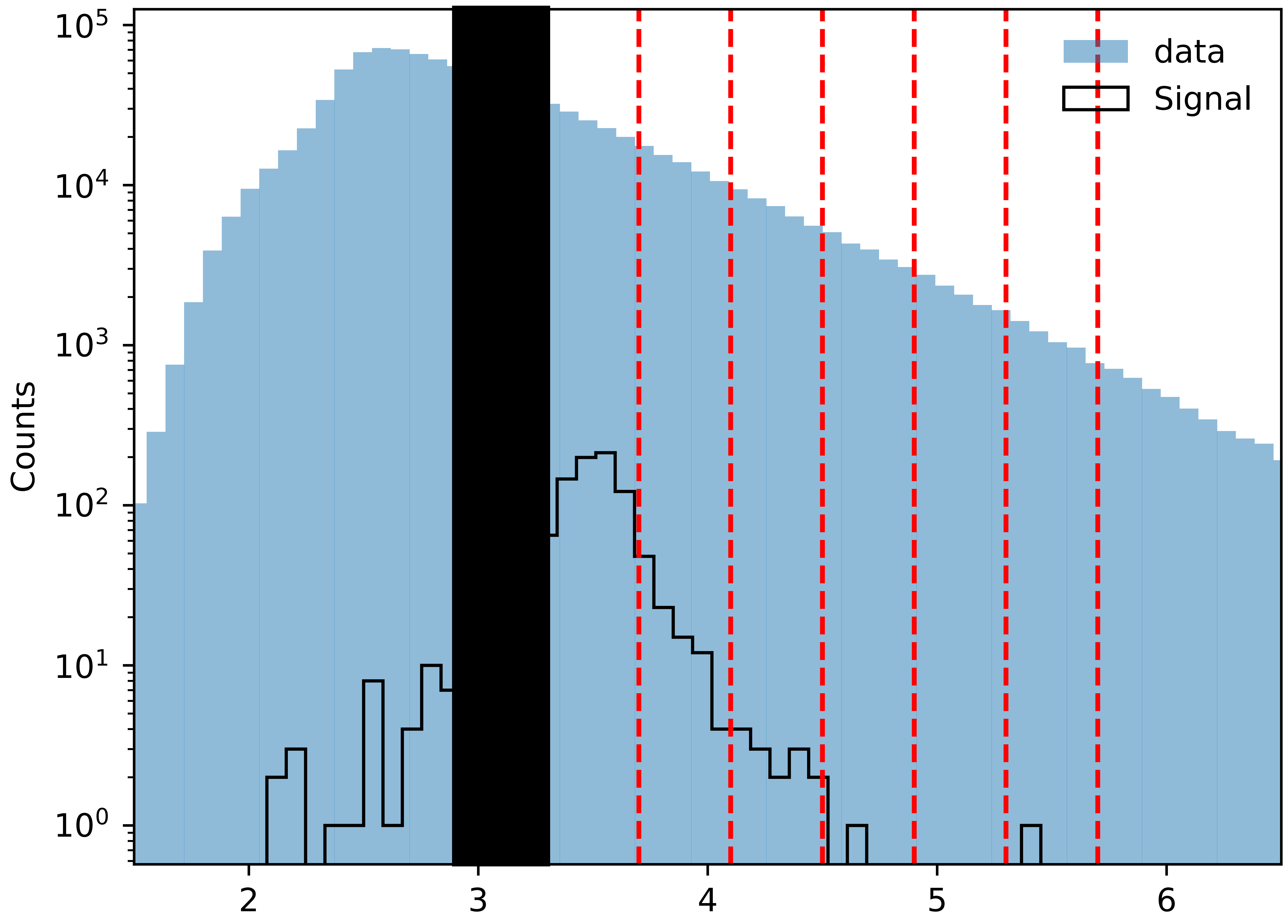
ANODE: arXiv:2001.04990v2
CATHODE: arXiv:2109.00546v3
CURTAINS: arXiv:2203.09470v3
R-ANODE: arXiv:2312.11629

Background Template generation is computationally expensive!



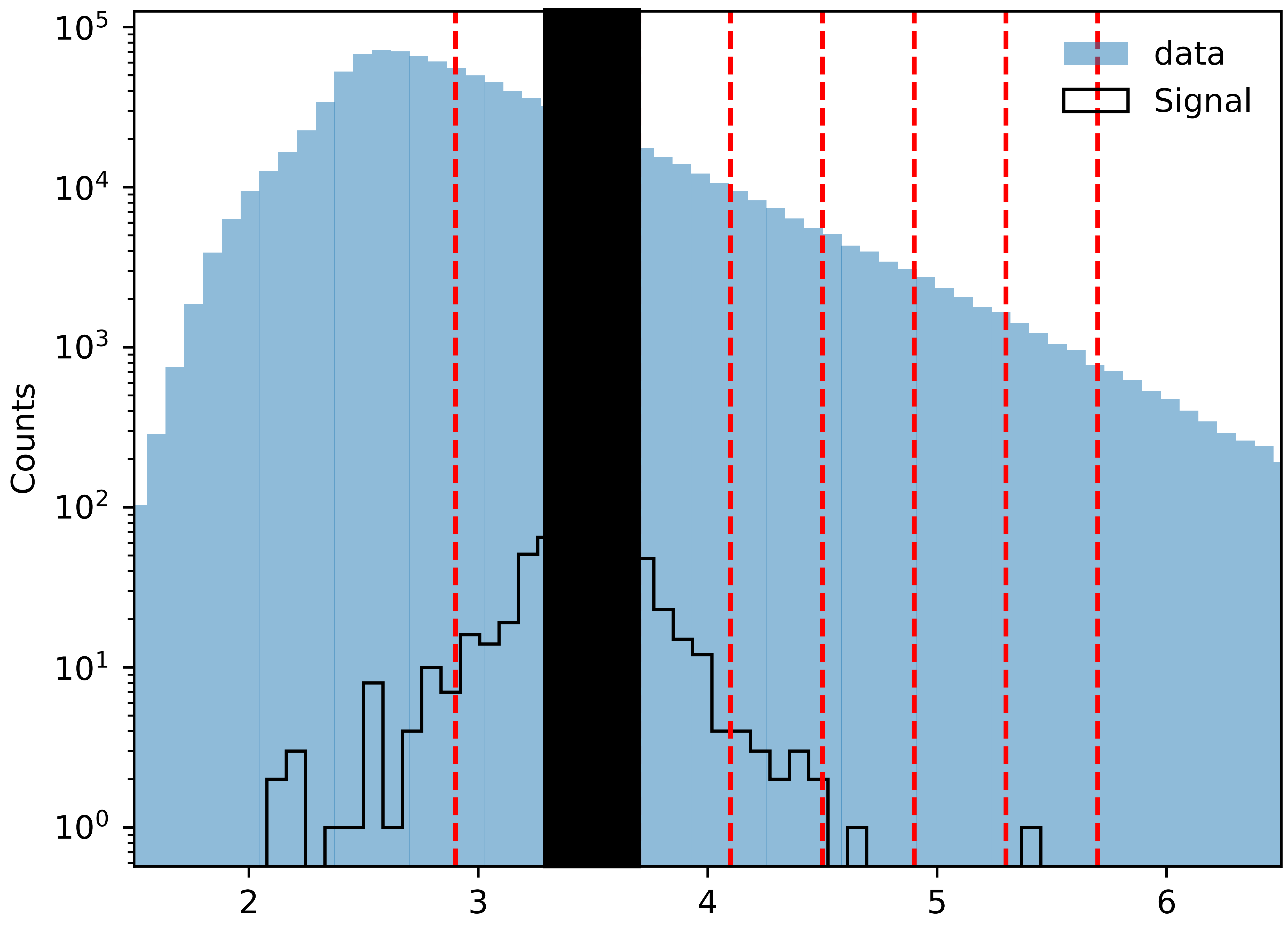
Background Template generation is computationally expensive!

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



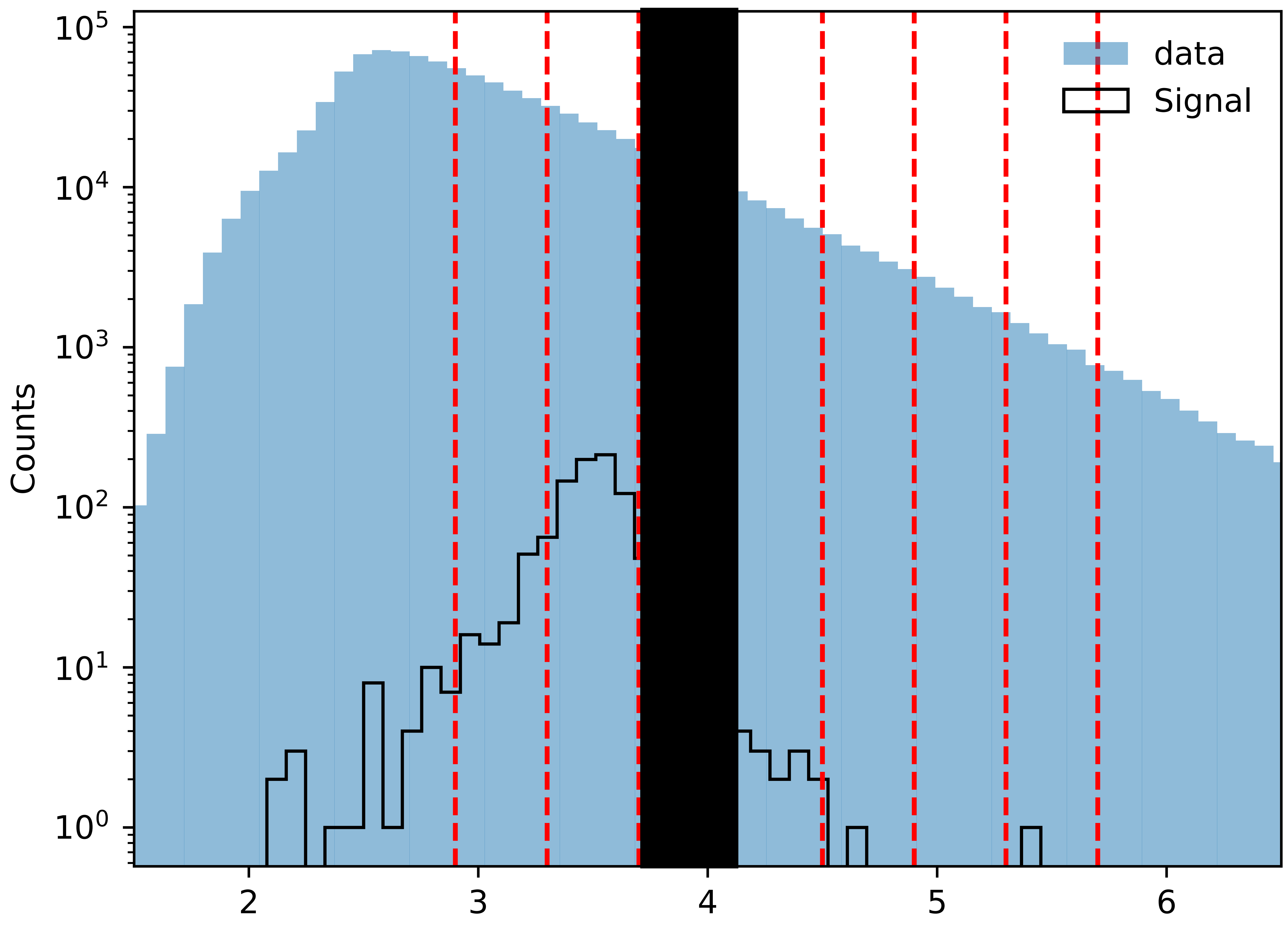
Background Template generation is computationally expensive!

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



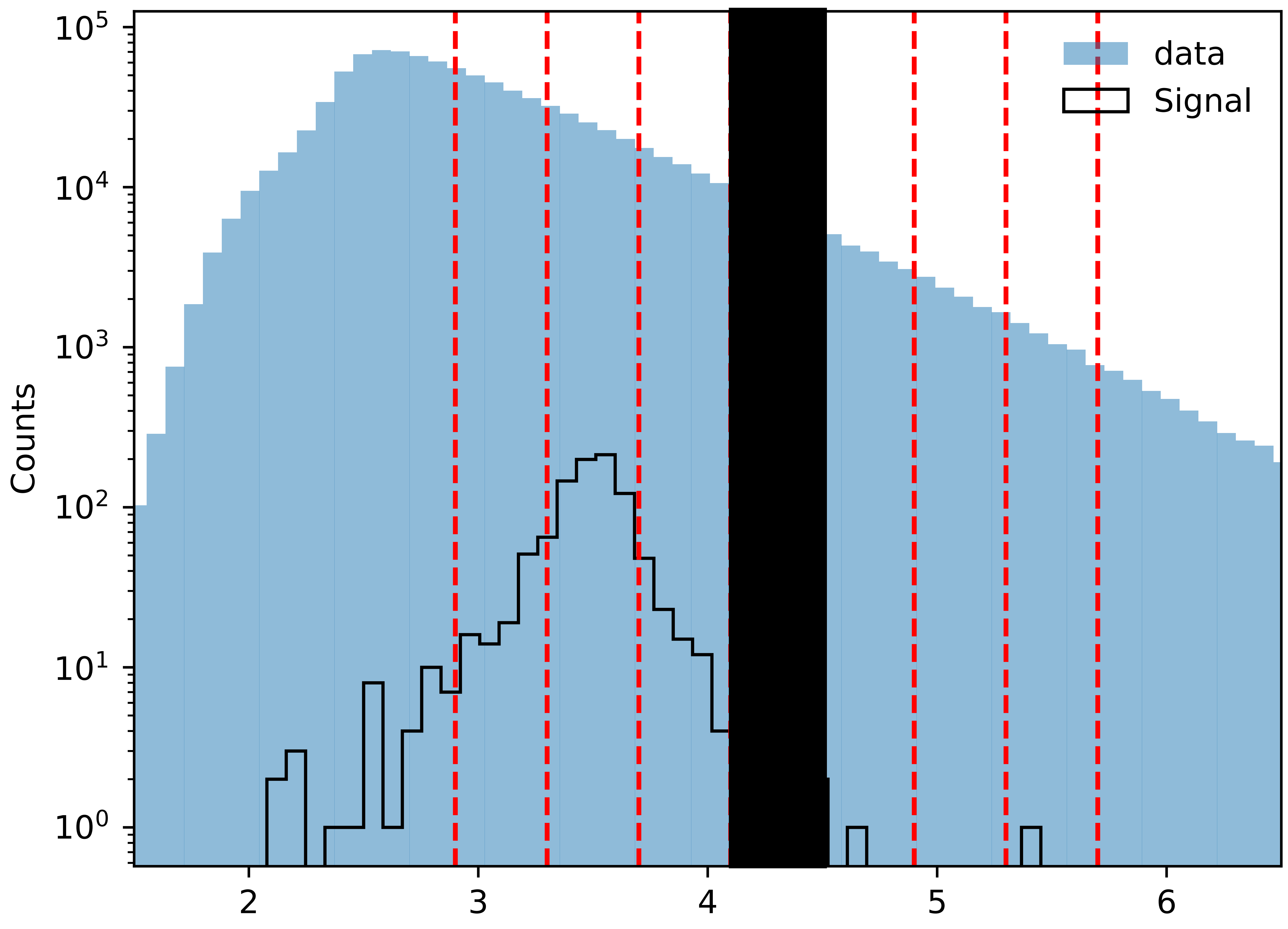
Background Template generation is computationally expensive!

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



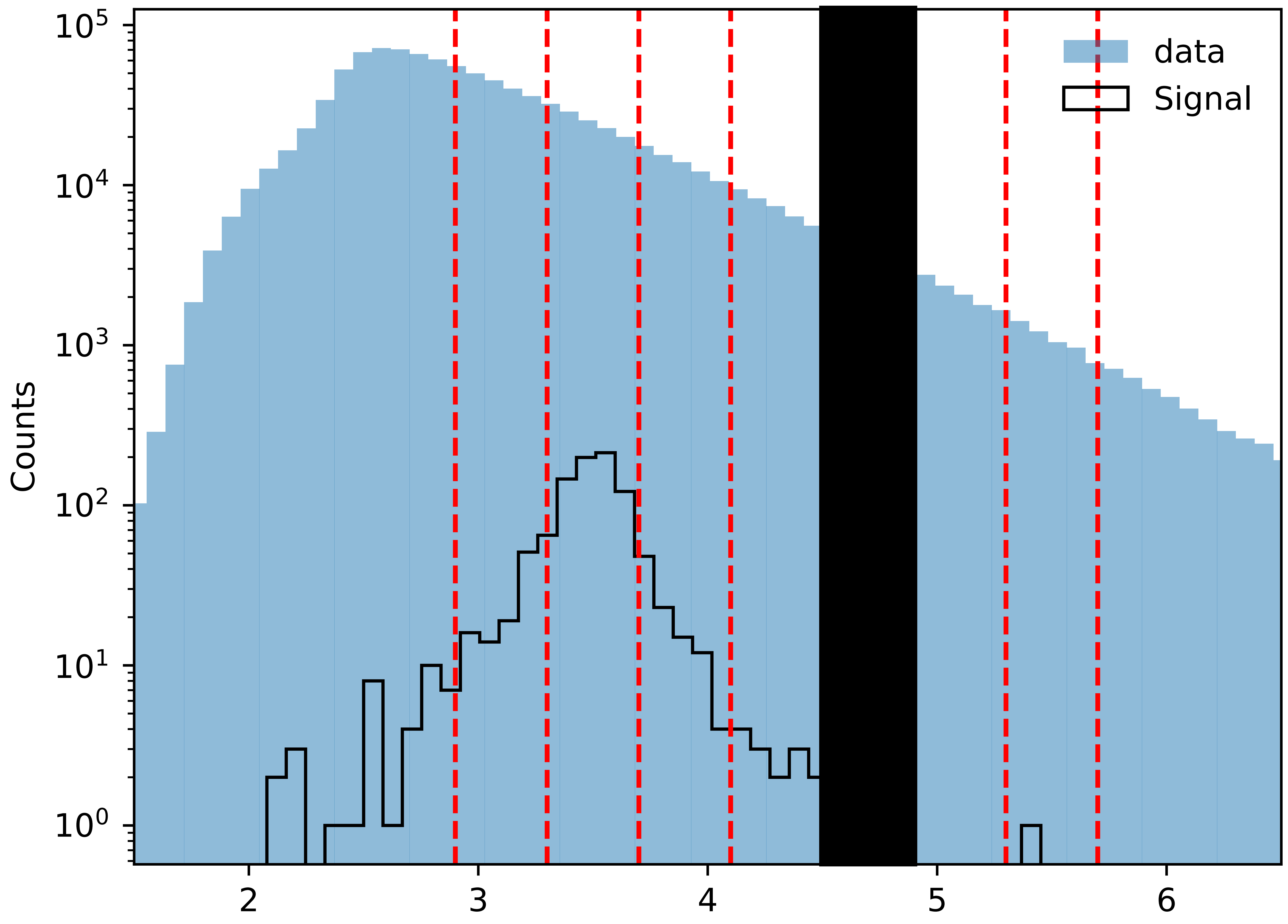
Background Template generation is computationally expensive!

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



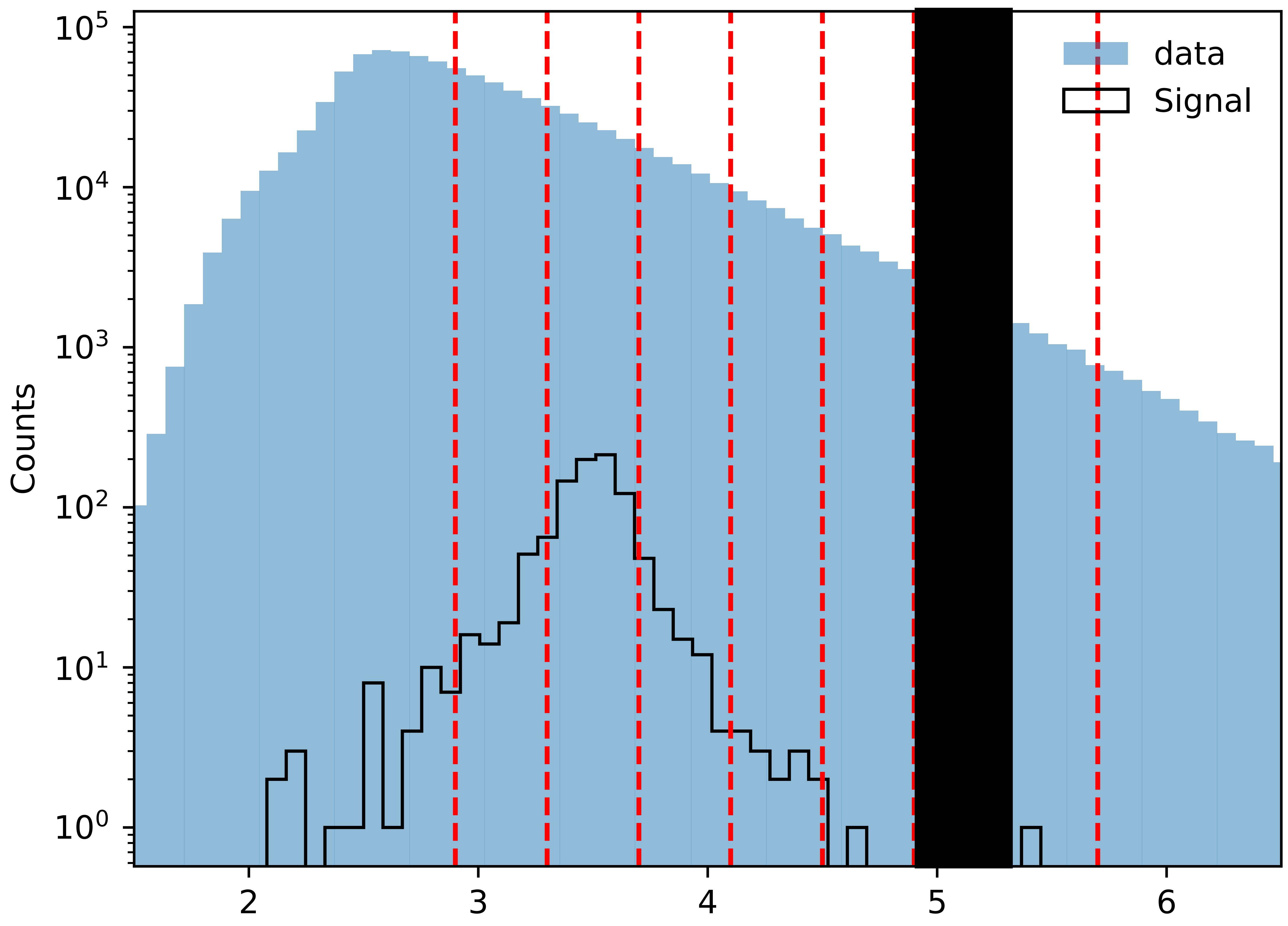
Background Template generation is computationally expensive!

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



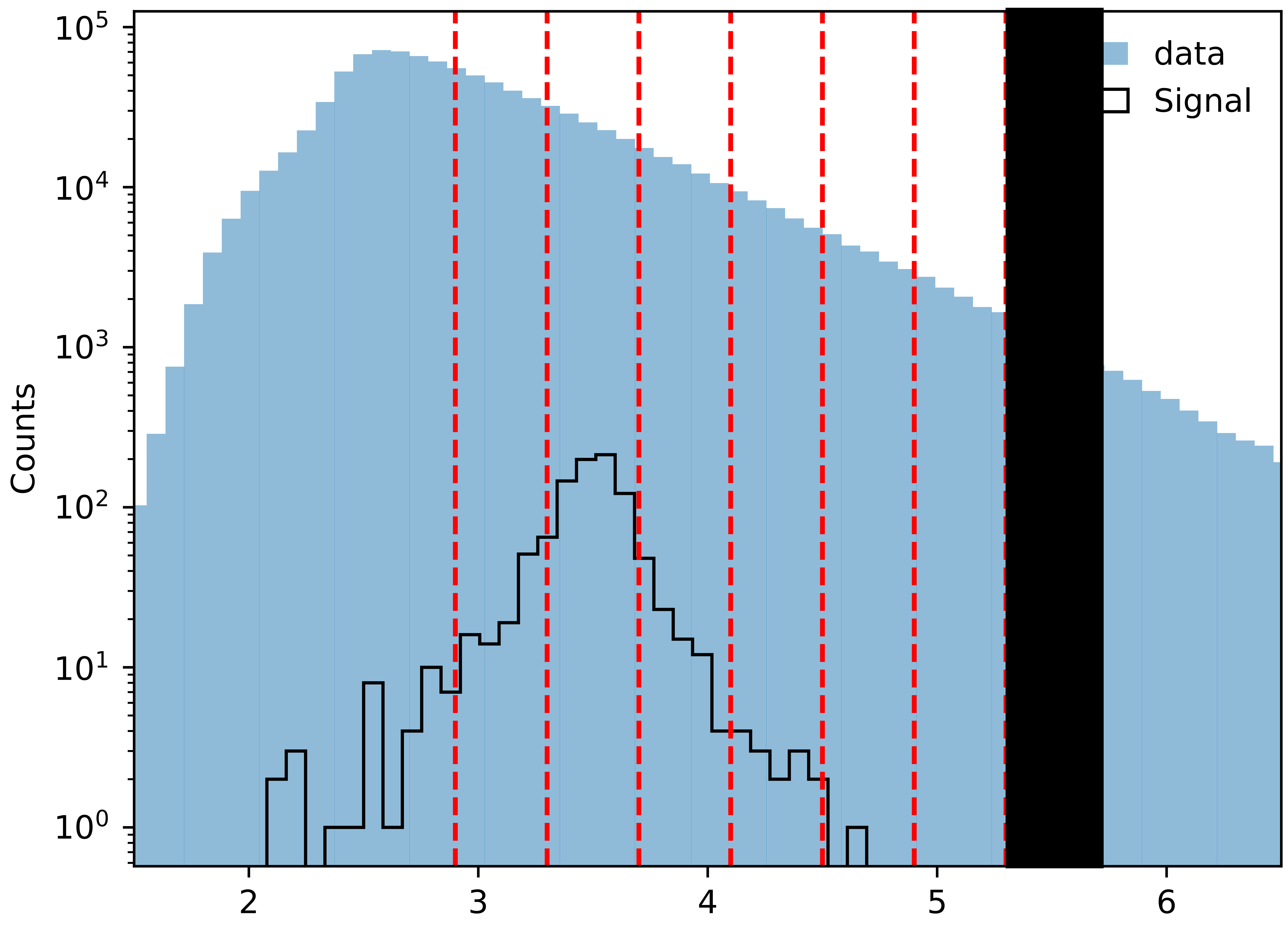
Background Template generation is computationally expensive!

- 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.

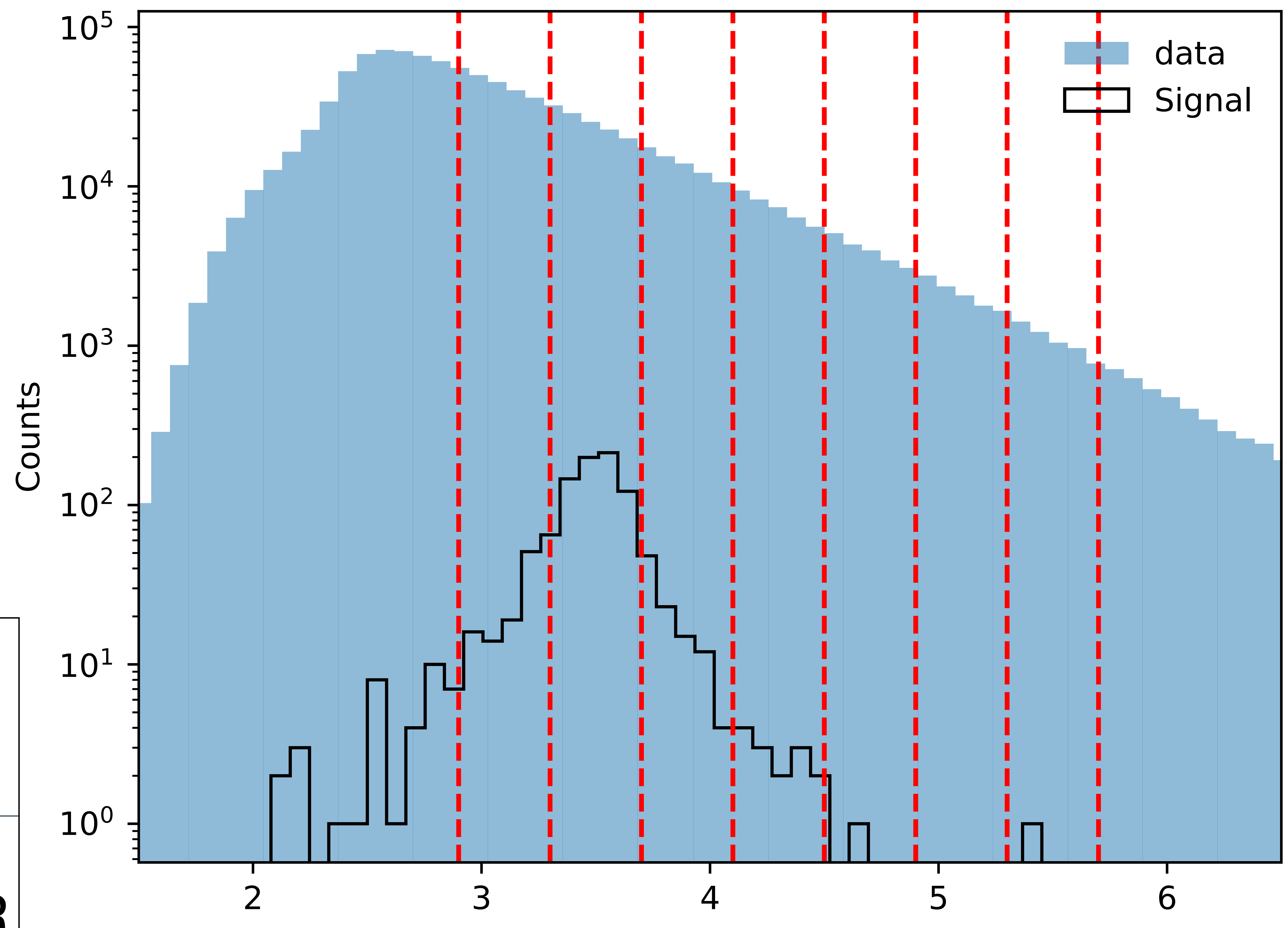


Background Template generation is computationally expensive!

- 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



Previous methods for faster template generation

Previous methods for faster template generation

- CURTAINs4F4 trains a base model on entire dataset. For each SR a lighter model is trained on shorter sidebands. (See arXiv:2305.04646)

Previous methods for faster template generation

- CURTAINs4F4 trains a base model on entire dataset. For each SR a lighter model is trained on shorter sidebands. (See arXiv:2305.04646)

- RAD-OT just uses Optimal Transport instead of a generative model for each SR. (See arXiv:2407.19818).

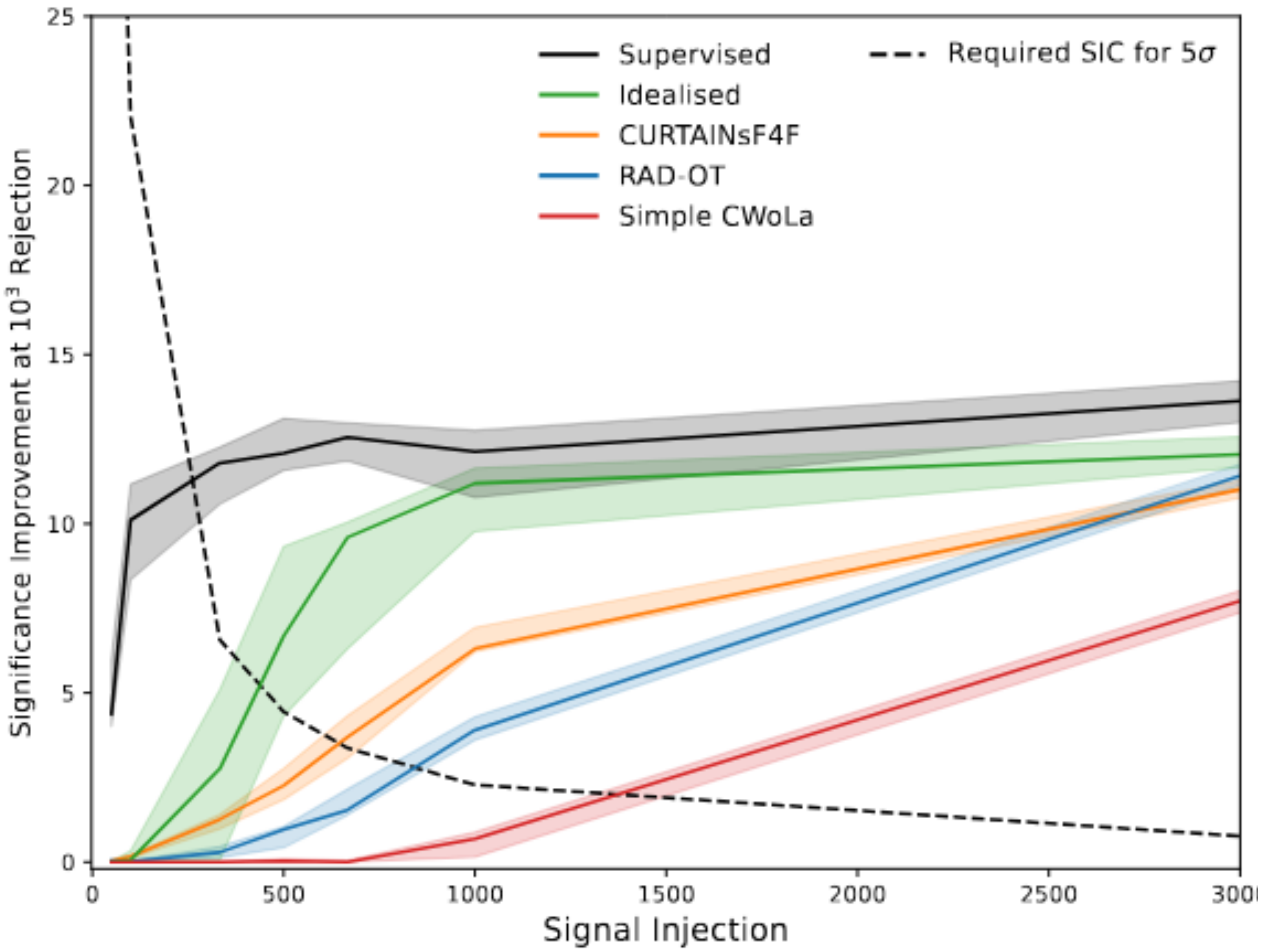
Previous methods for faster template generation

- CURTAINs4F4 trains a base model on entire dataset. For each SR a lighter model is trained on shorter sidebands. (See arXiv:2305.04646)
- RAD-OT just uses Optimal Transport instead of a generative model for each SR. (See arXiv:2407.19818).

Method	Generative Model	Timing
CATHODE/ANODE	Normalizing Flows	3 hours per SR
CURTAINS4F4	Normalizing Flows	3 hours (base model) + 7 mins per SR
RAD-OT	Optimal Transport	10 mins per SR

Previous methods for faster template generation

- CURTAINs4F4 trains a base model on entire dataset. For each SR a lighter model is trained on shorter sidebands. (See [arXiv:2305.04646](https://arxiv.org/abs/2305.04646))
- RAD-OT just uses Optimal Transport instead of a generative model for each SR. (See [arXiv:2407.19818](https://arxiv.org/abs/2407.19818)).
- RAD-OT is fast, but compromises in signal sensitivity.

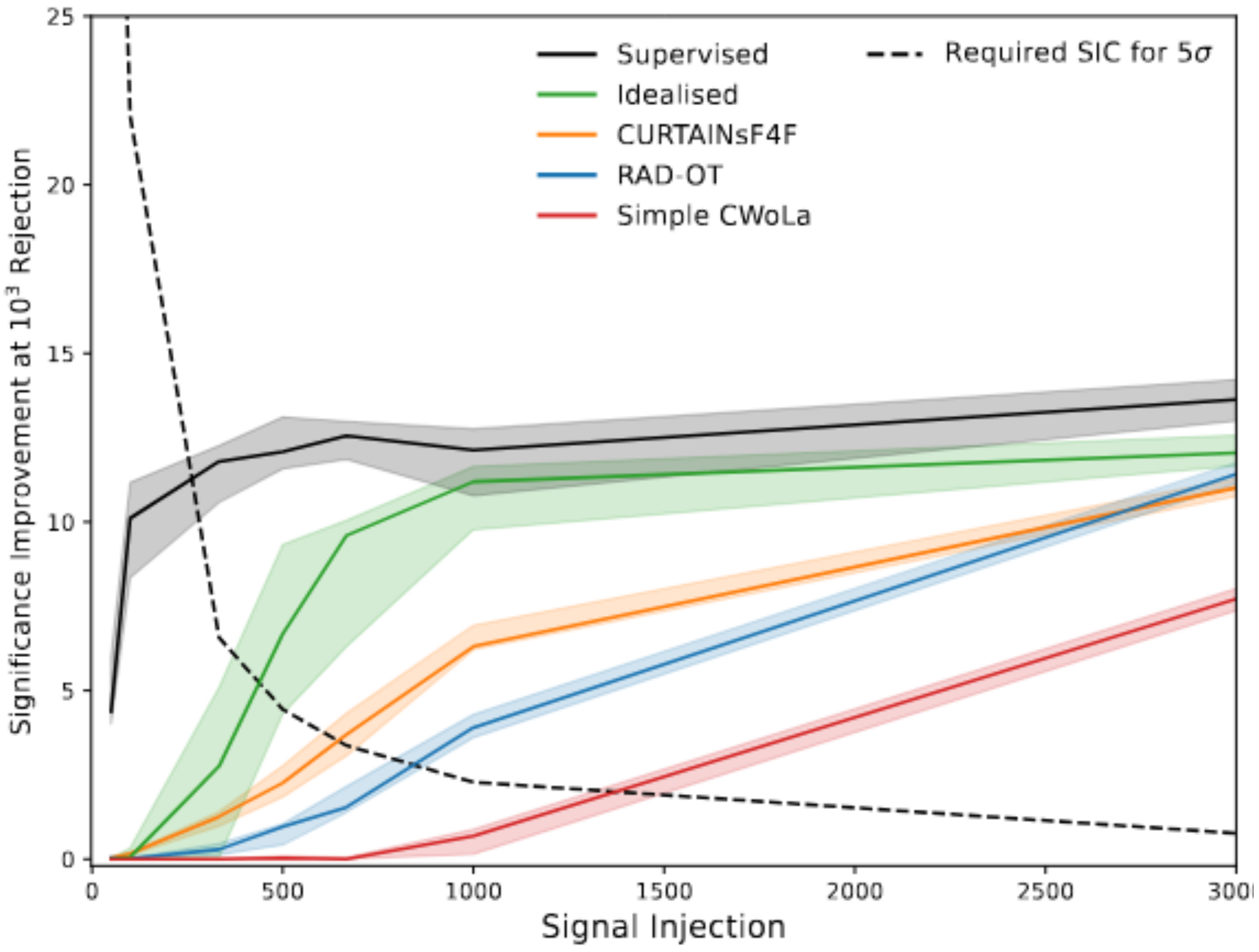


Previous methods for faster template generation

- CURTAINs4F4 trains a base model on entire dataset. For each SR a lighter model is trained on shorter sidebands. (See arXiv:2305.04646)

- RAD-OT just uses Optimal Transport instead of a generative model for each SR. (See arXiv:2407.19818).

- RAD-OT is fast, but compromises in signal sensitivity.

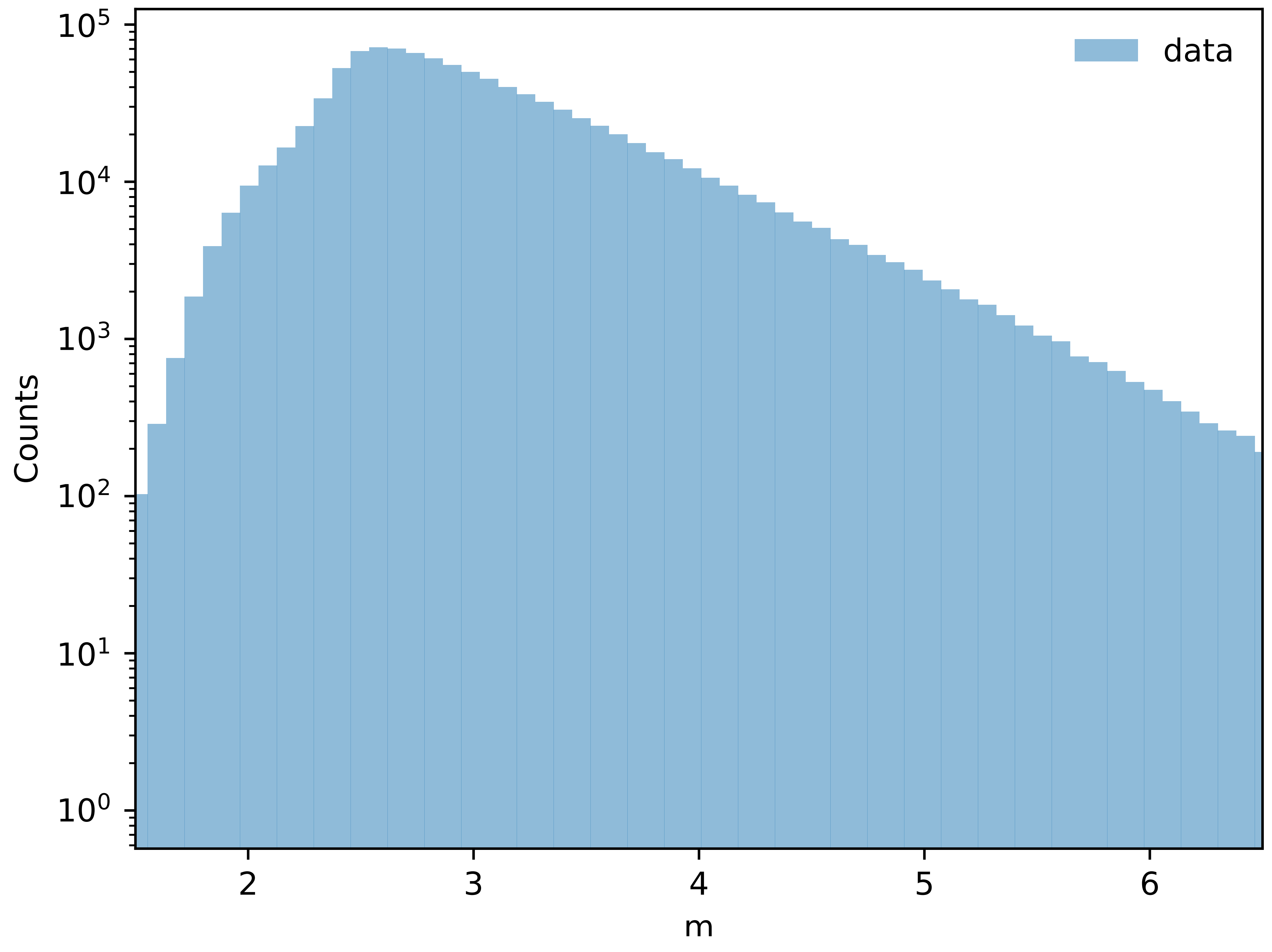


TRANSIT: A new method (next talk by Ivan)

SIGMA: Single Interpolated Generative Model for Anomalies

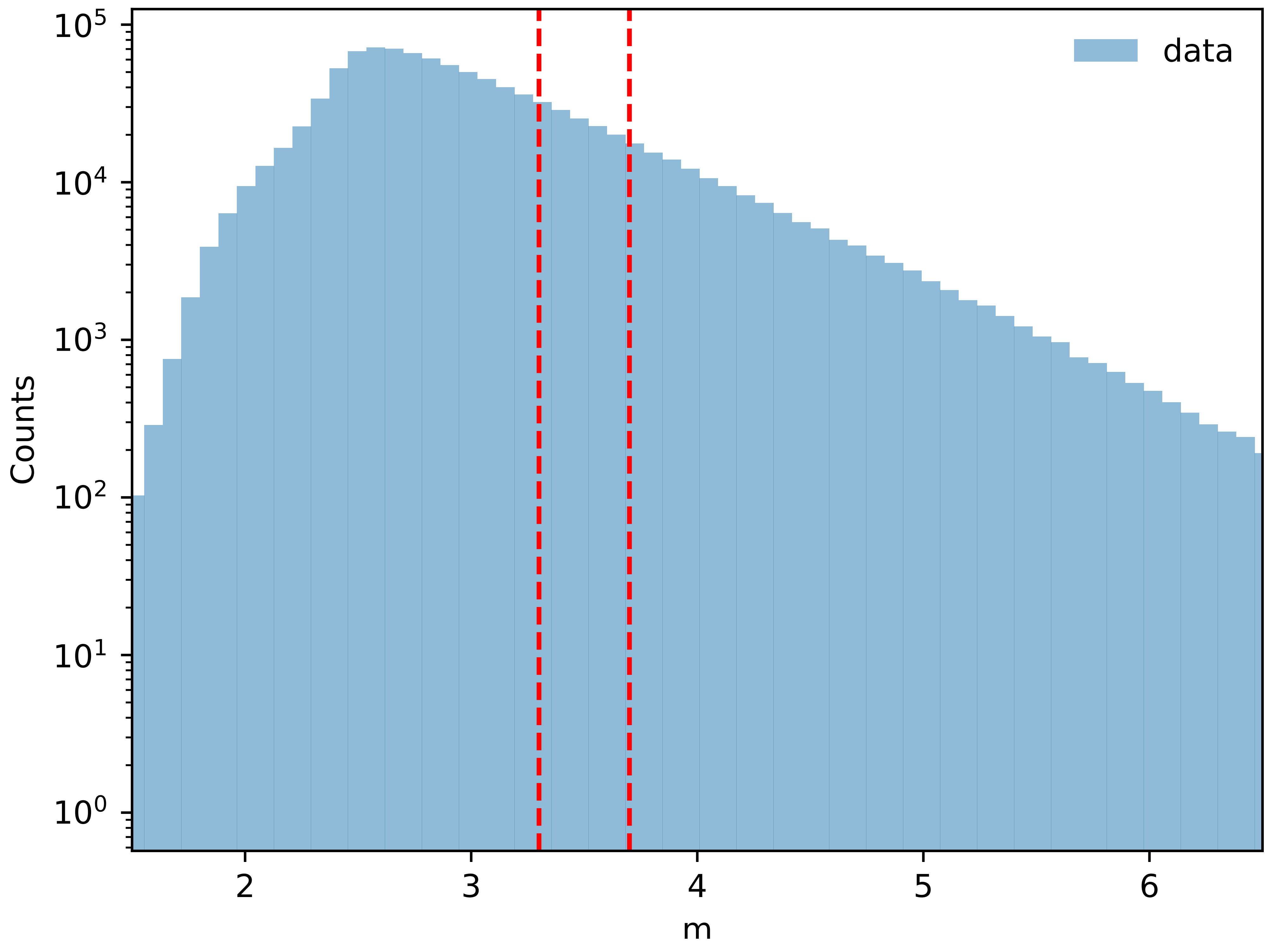
SIGMA: Single Interpolated Generative Model for Anomalies

- 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.

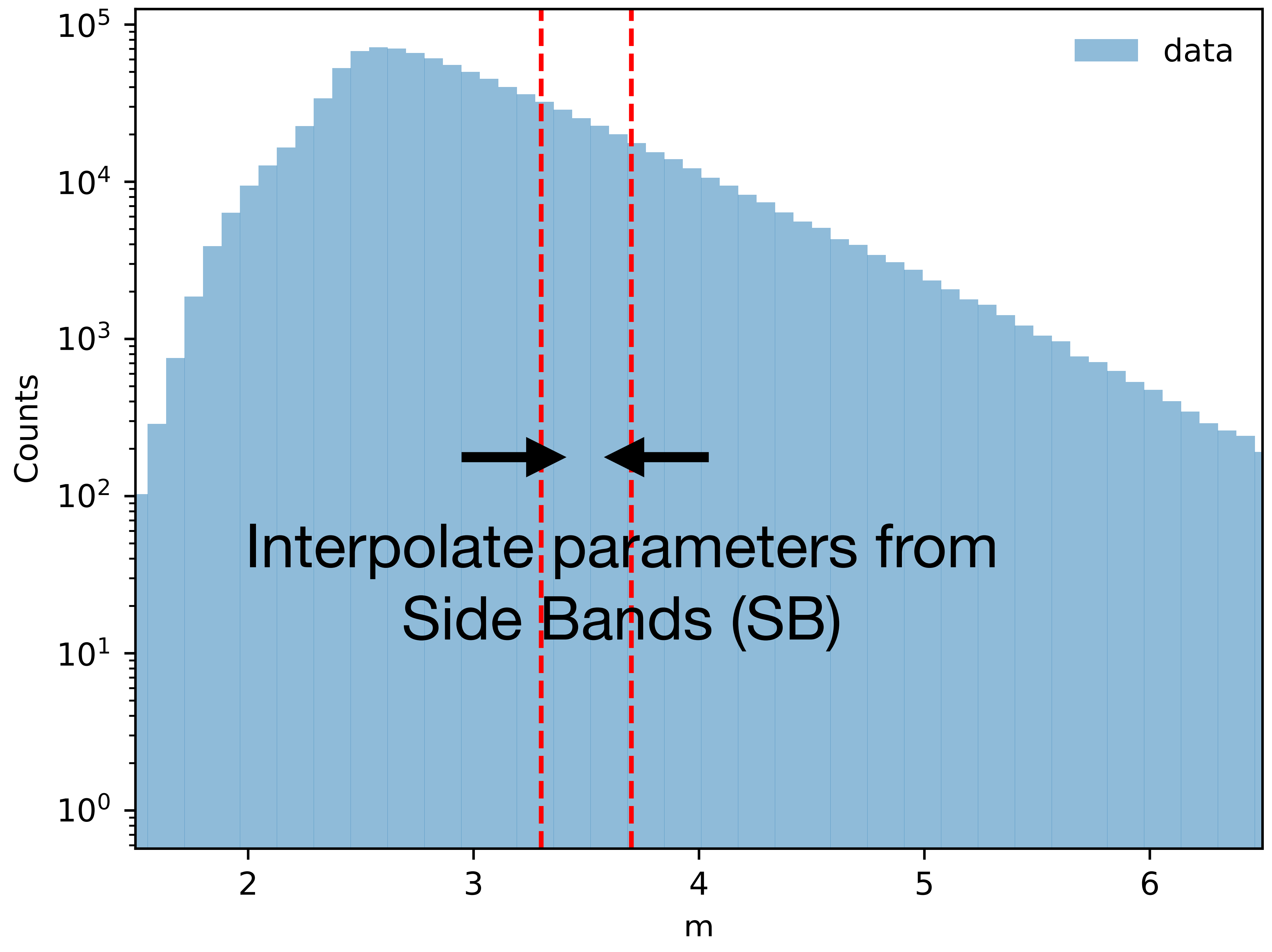


SIGMA: Single Interpolated Generative Model for Anomalies

- 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)

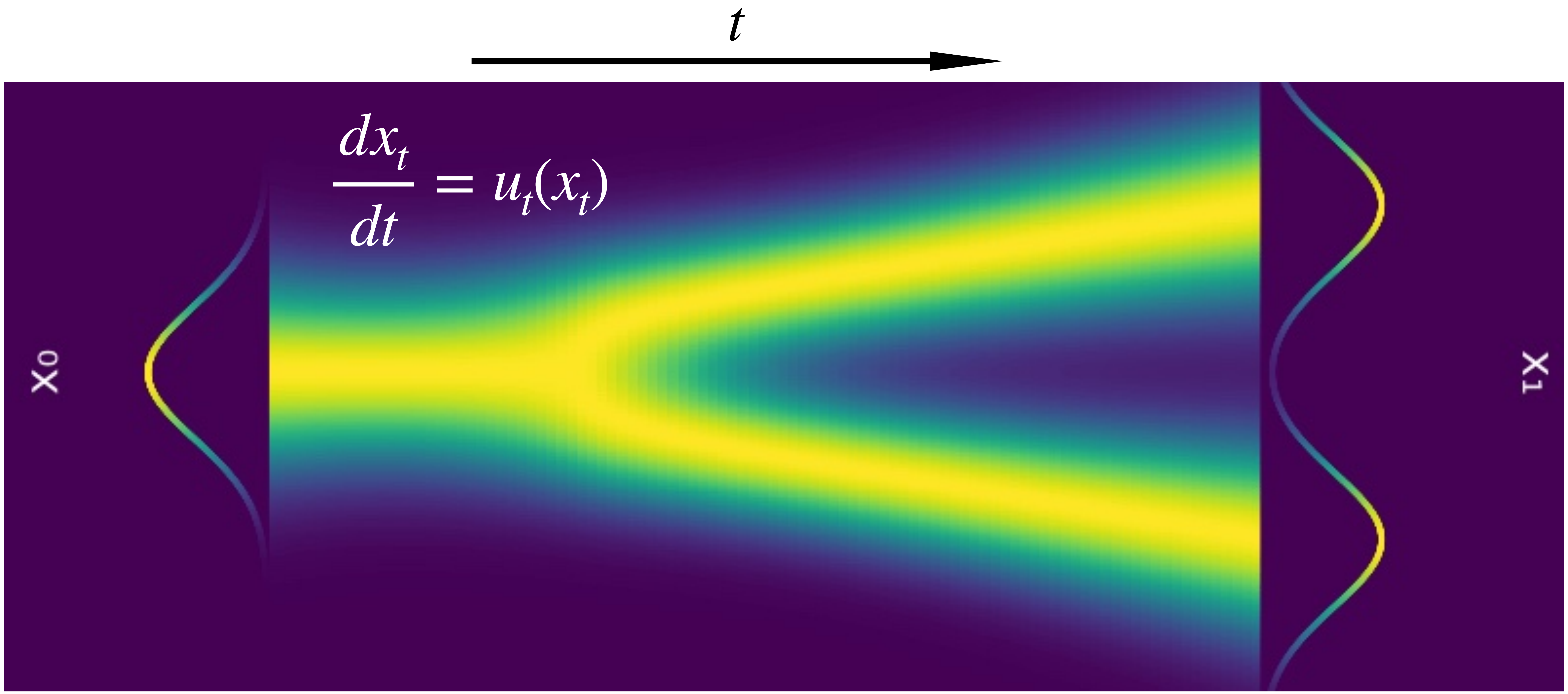
Known
Base
Distribution

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

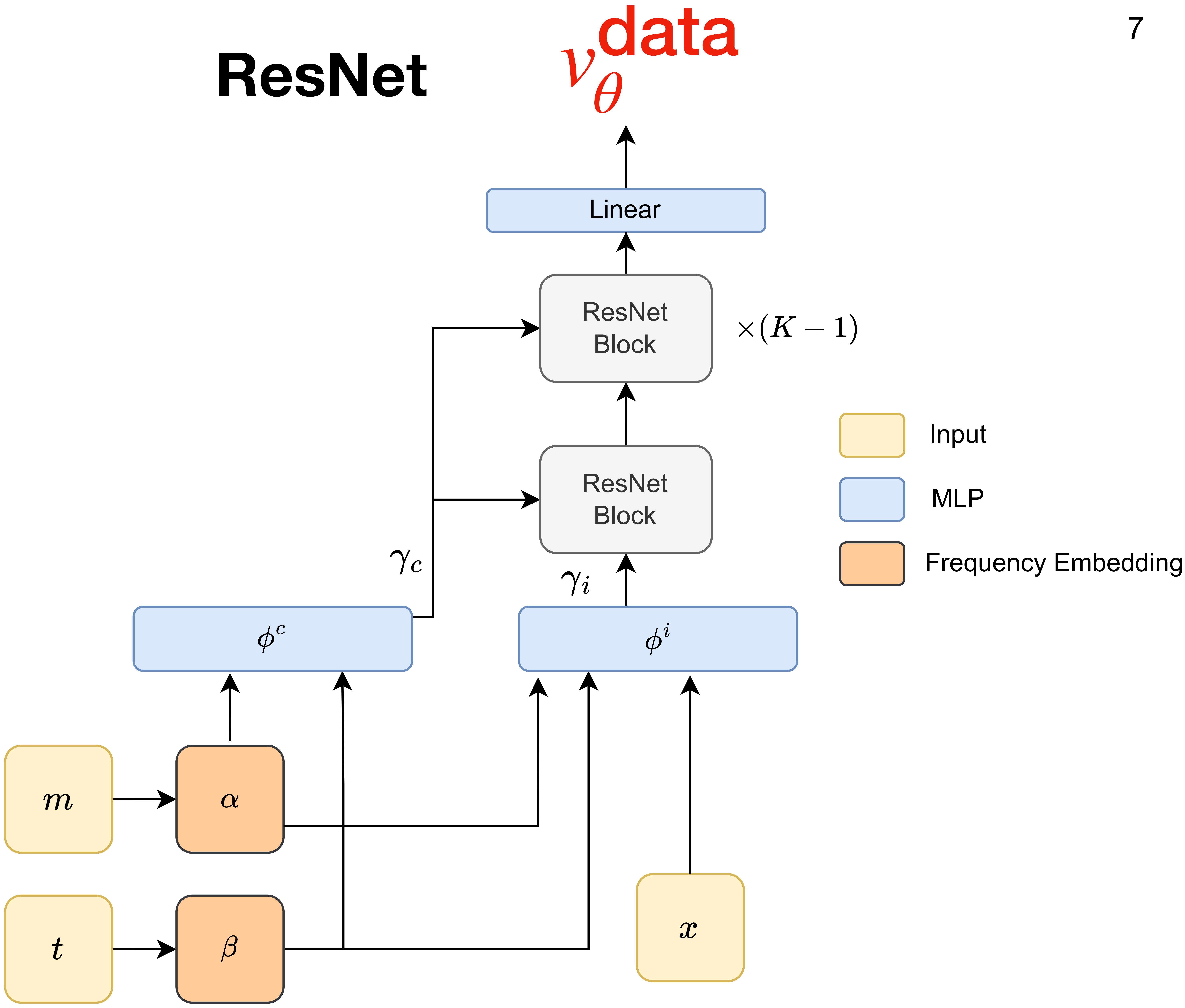
Image from https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html

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

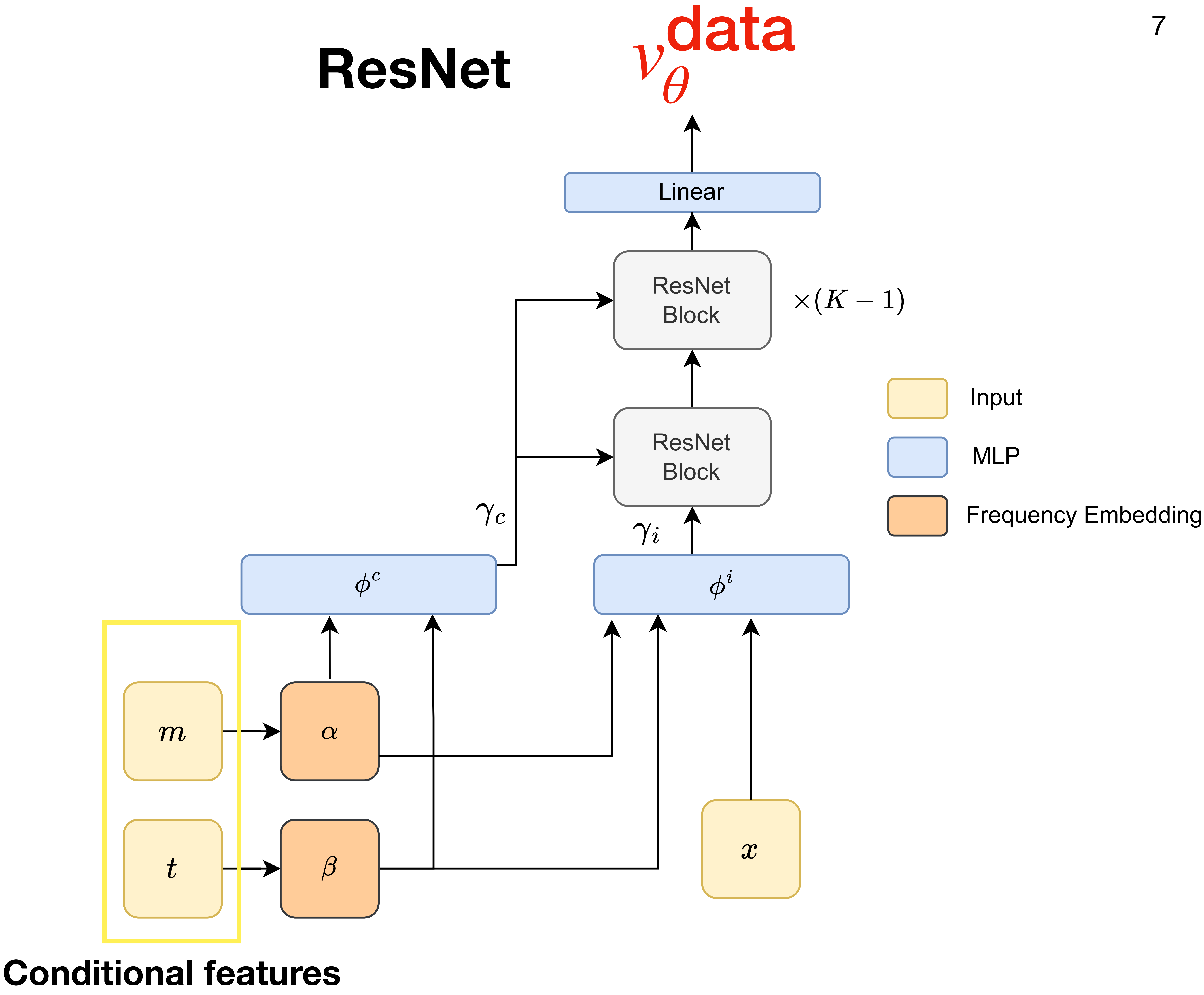
arXiv:2310.00049: EPiC-ly Fast Particle Cloud Generation with Flow-Matching and Diffusion
arXiv:2209.15571: Building Normalizing Flows with Stochastic Interpolants
arXiv:2210.02747: Flow Matching for Generative Modeling
arXiv:2312.00123: Flow Matching Beyond Kinematics: Generating Jets with Particle-ID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

Architecture

Architecture



Architecture



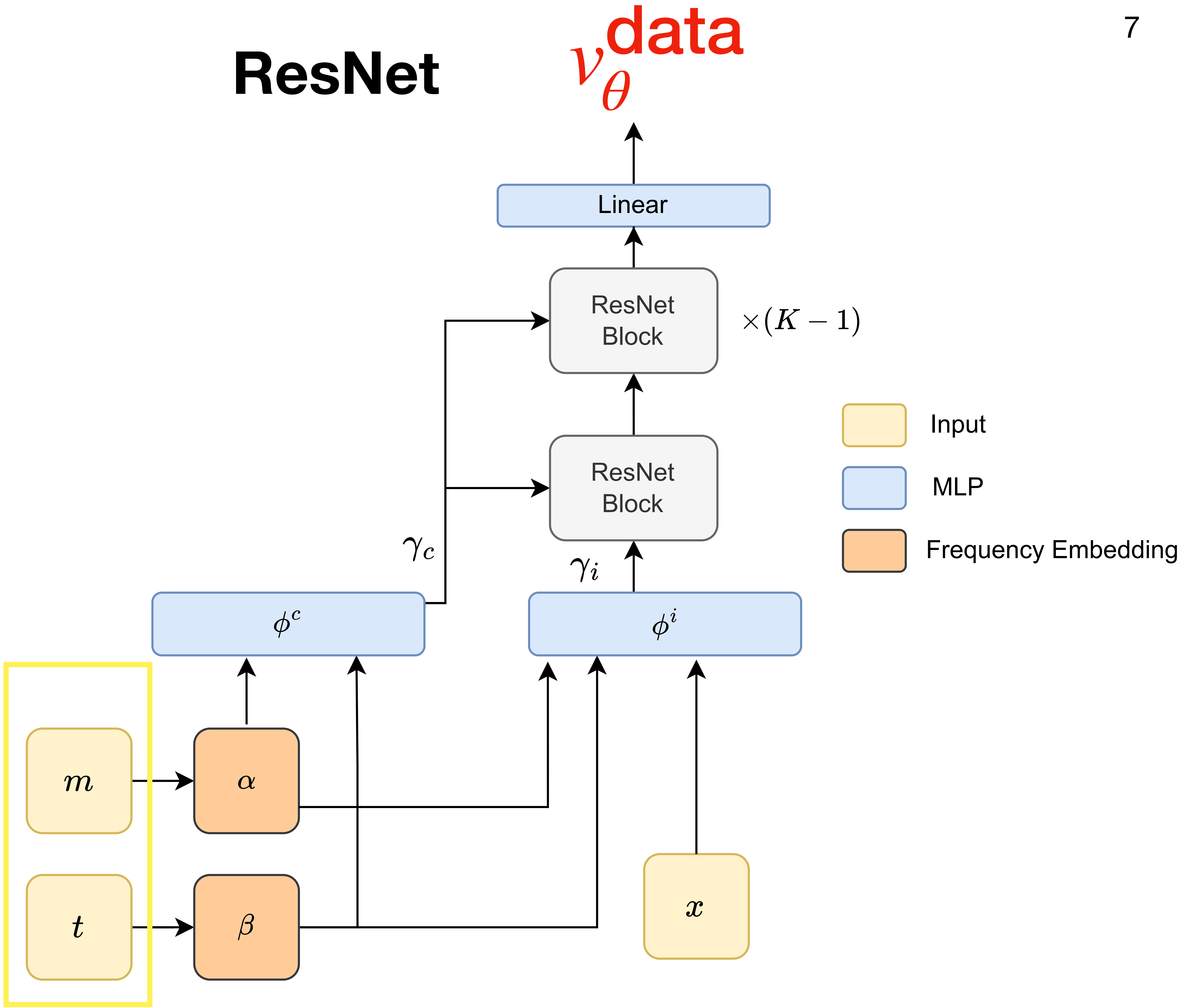
Architecture

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



Conditional features

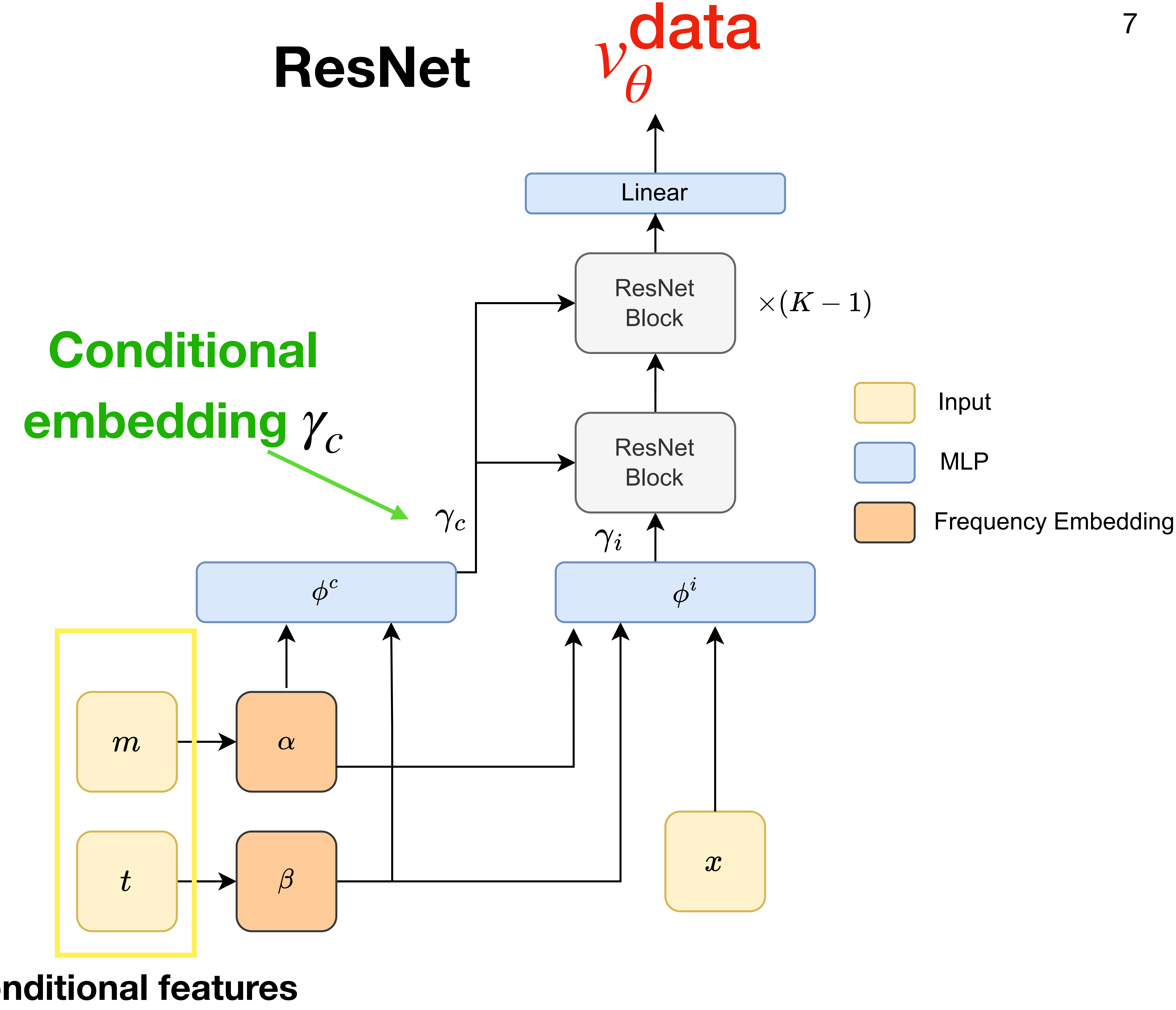
NERF: arXiv:2003.08934

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.

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

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



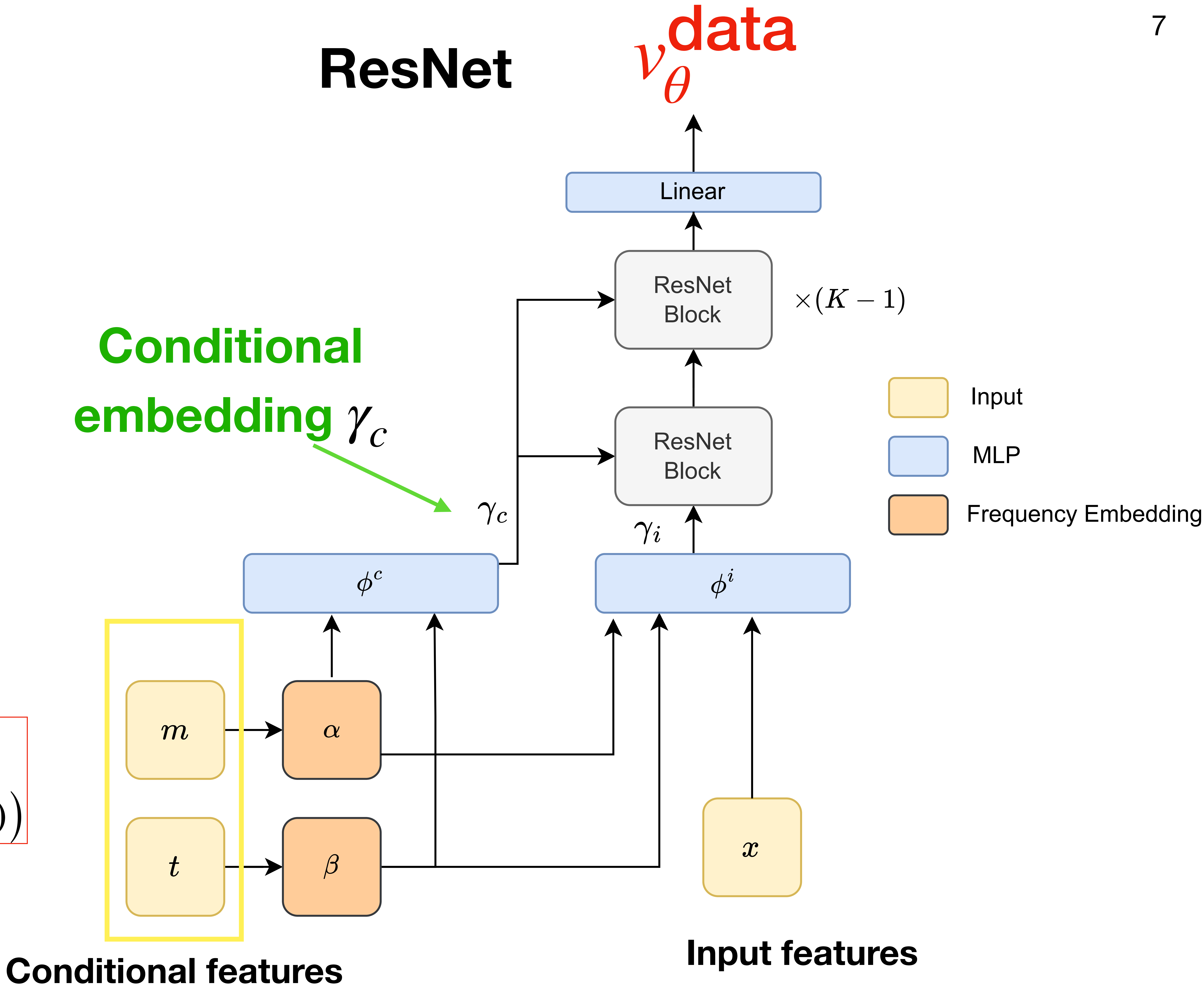
NERF: arXiv:2003.08934

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.

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

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



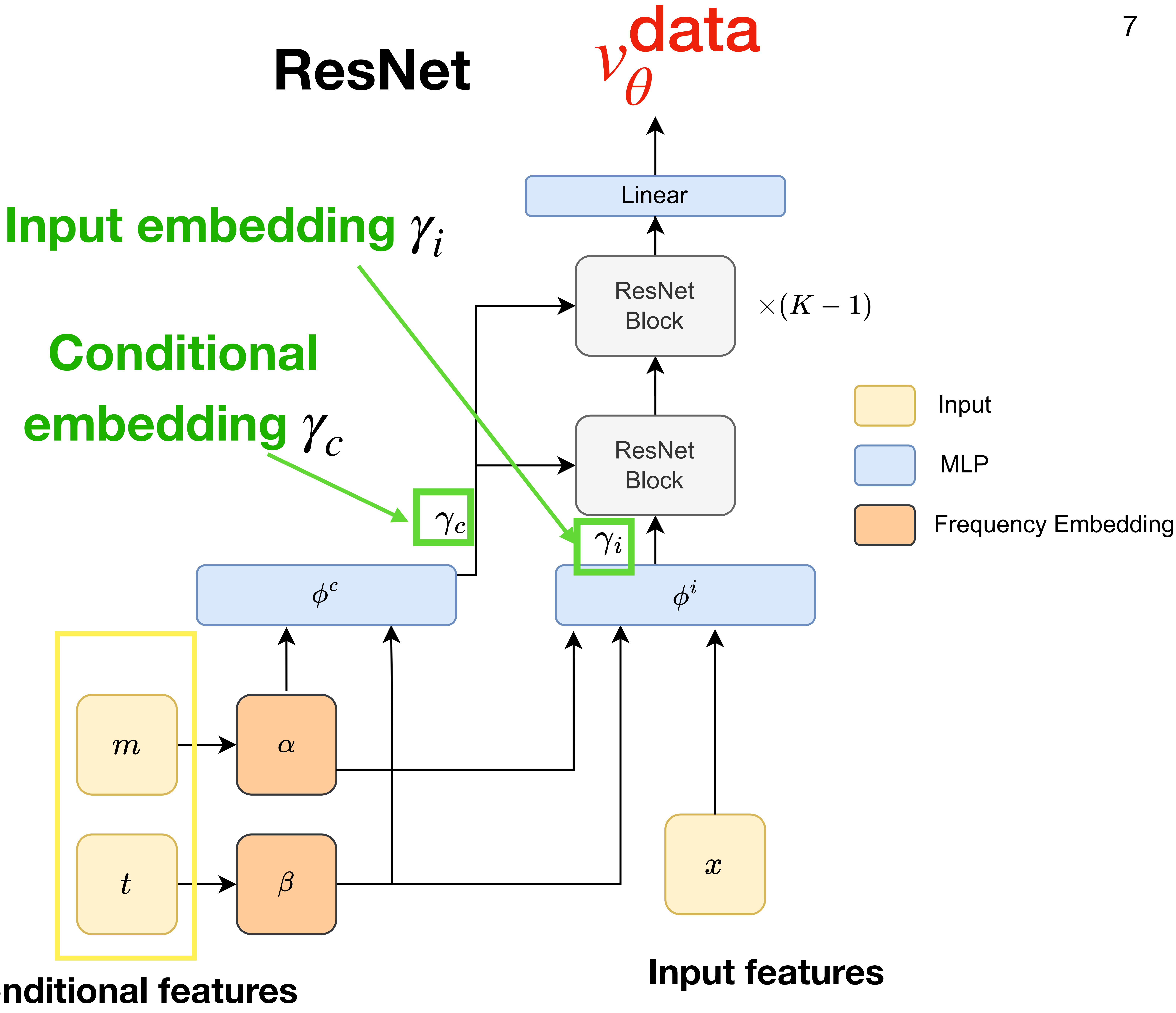
NERF: arXiv:2003.08934

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.

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

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

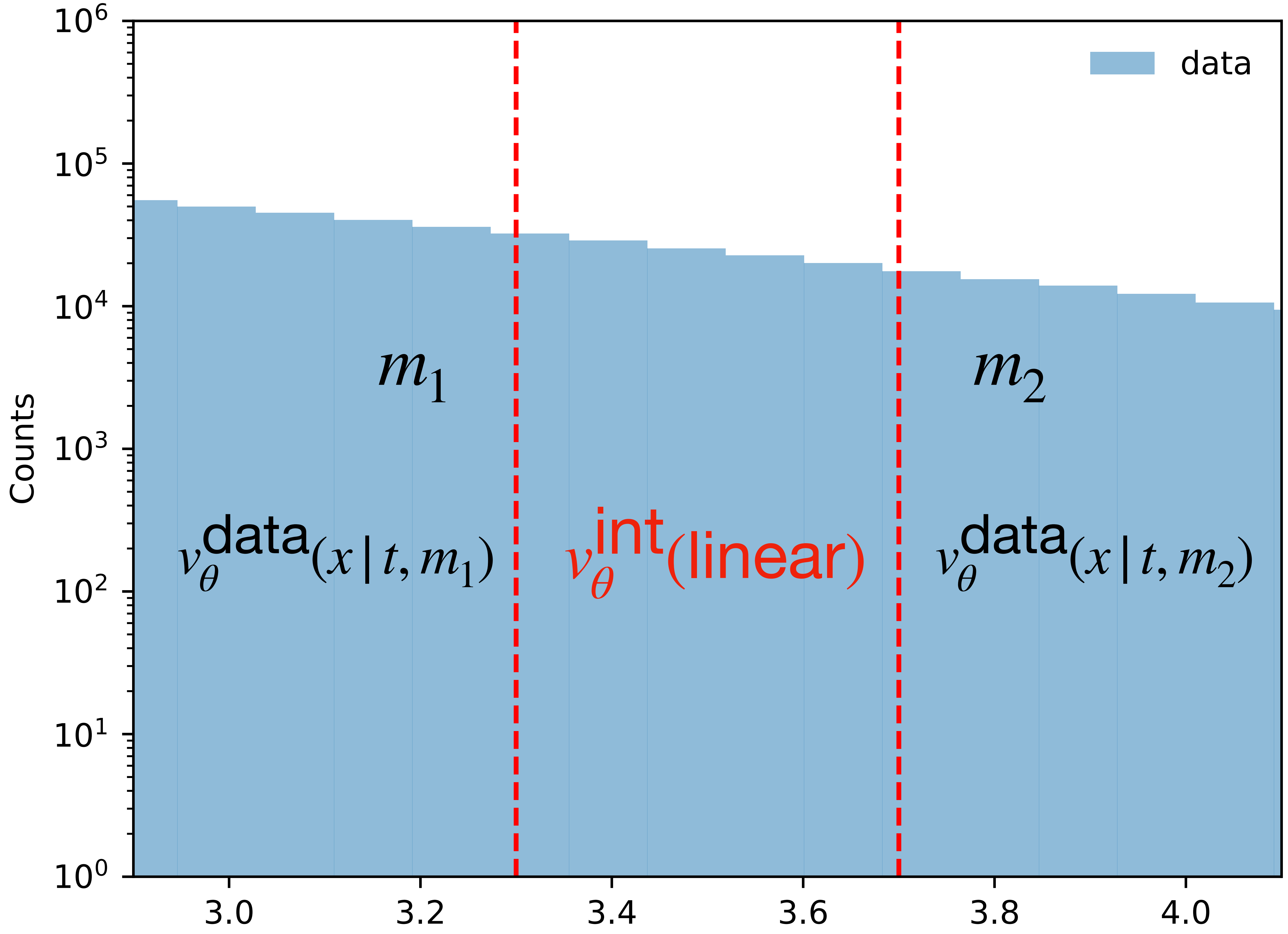


NERF: arXiv:2003.08934

Interpolation using SIGMA

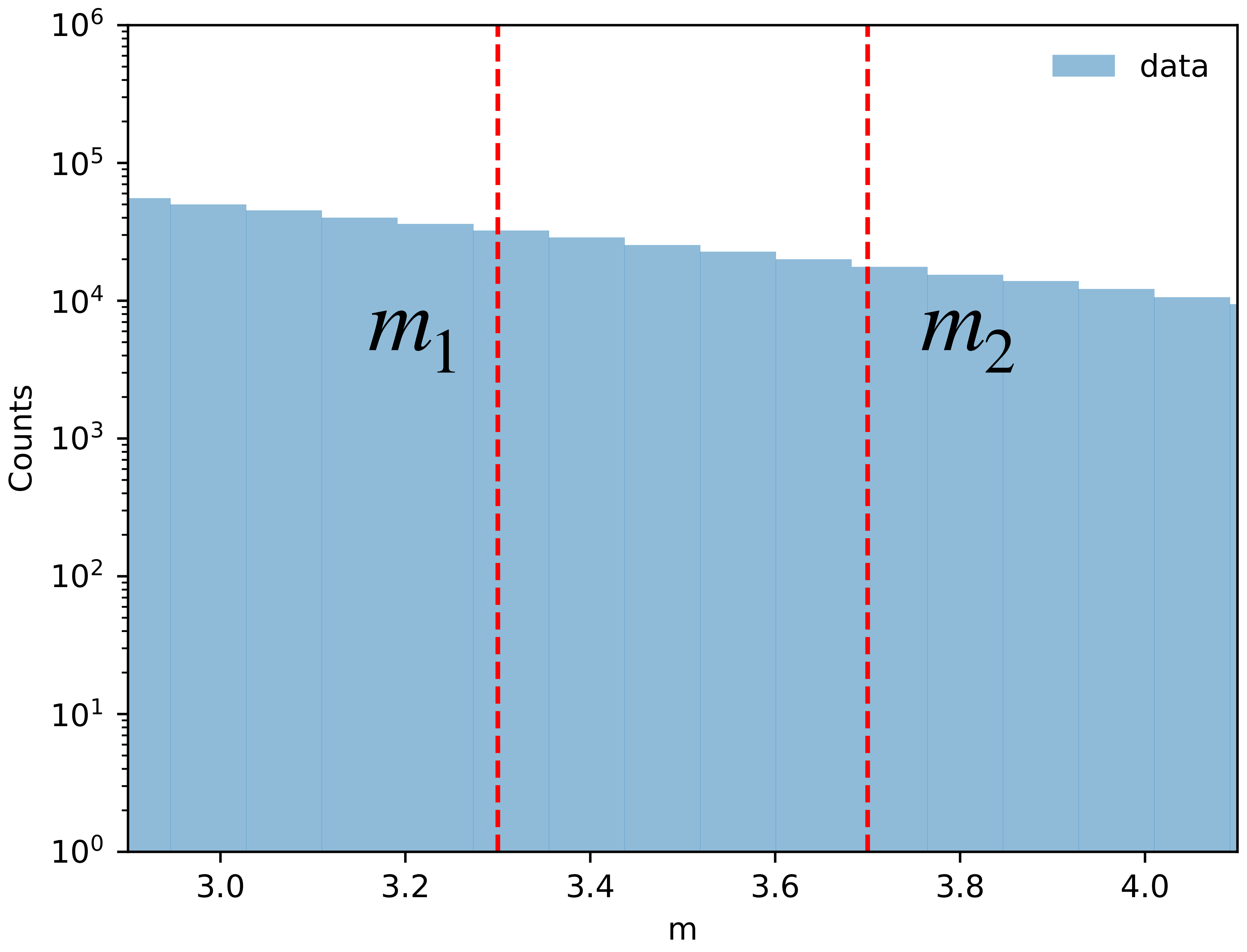
- $v_\theta^{\text{int}}(x\,|\,t, m) = \xi * \; v_\theta^{\text{data}}(x\,|\,t, m_1) + (1-\xi) * v_\theta^{\text{data}}(x\,|\,t, m_2)$

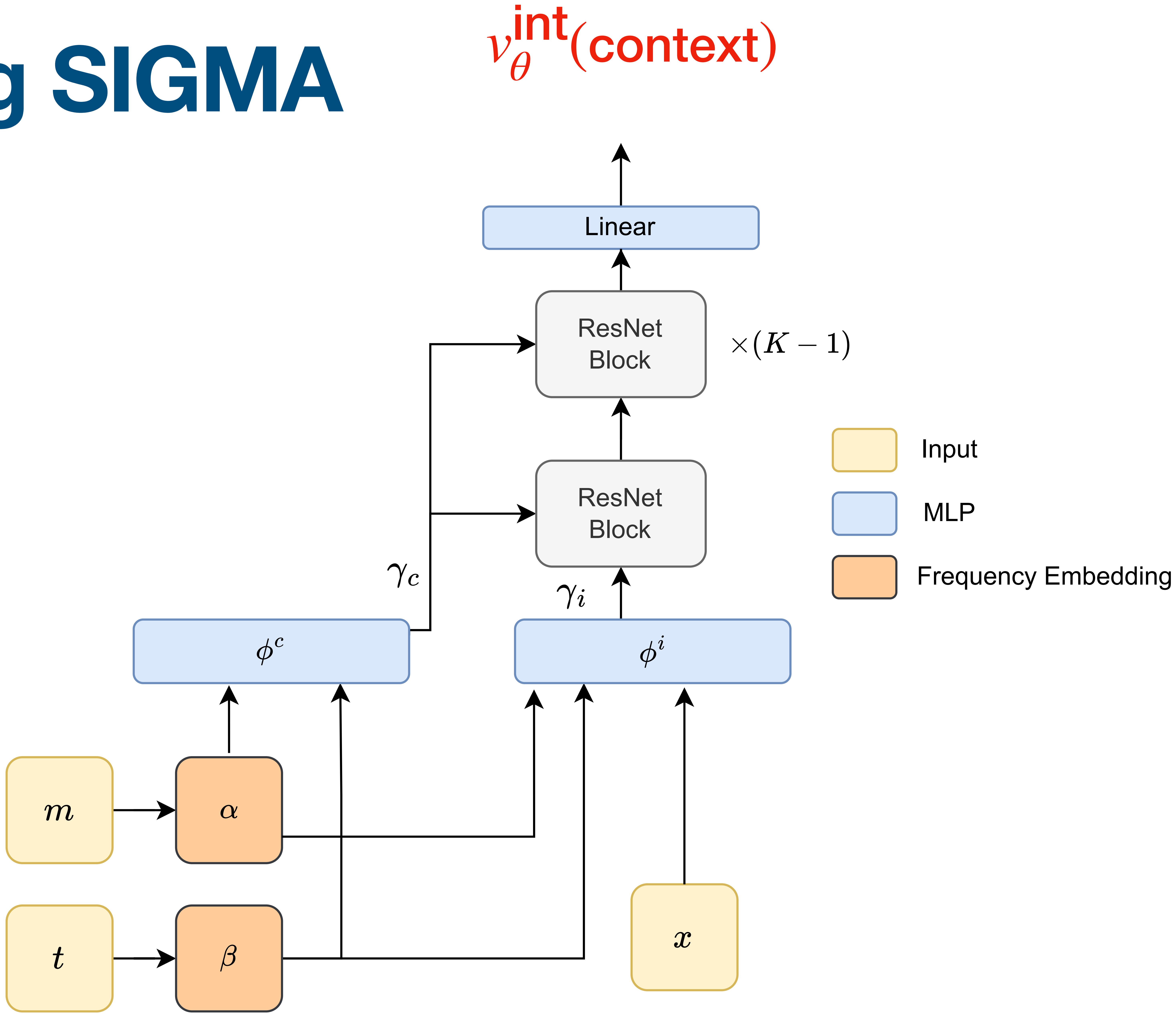
$$\xi = \frac{m - m_2}{m_1 - m_2}$$



Interpolation using SIGMA

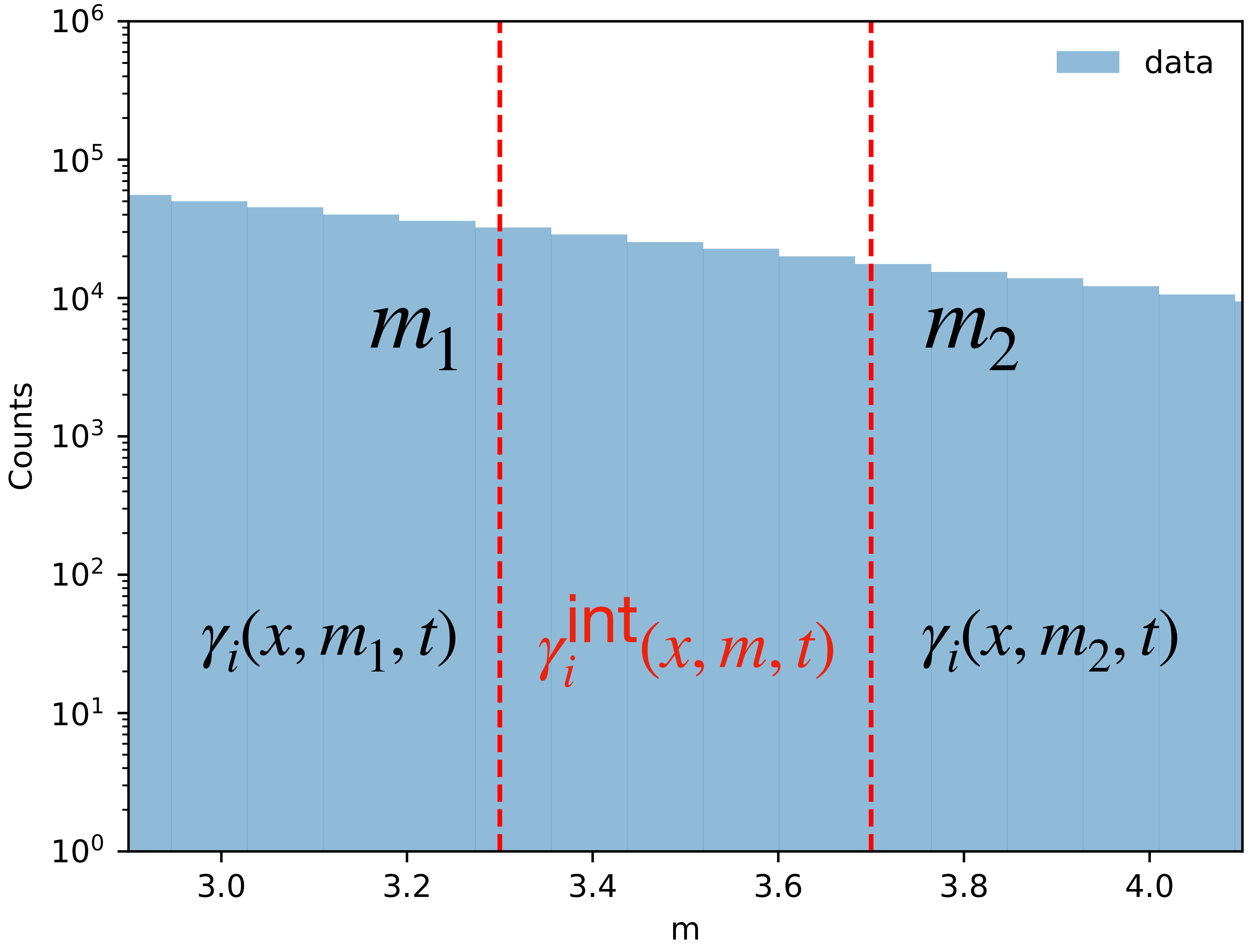
Interpolation using SIGMA

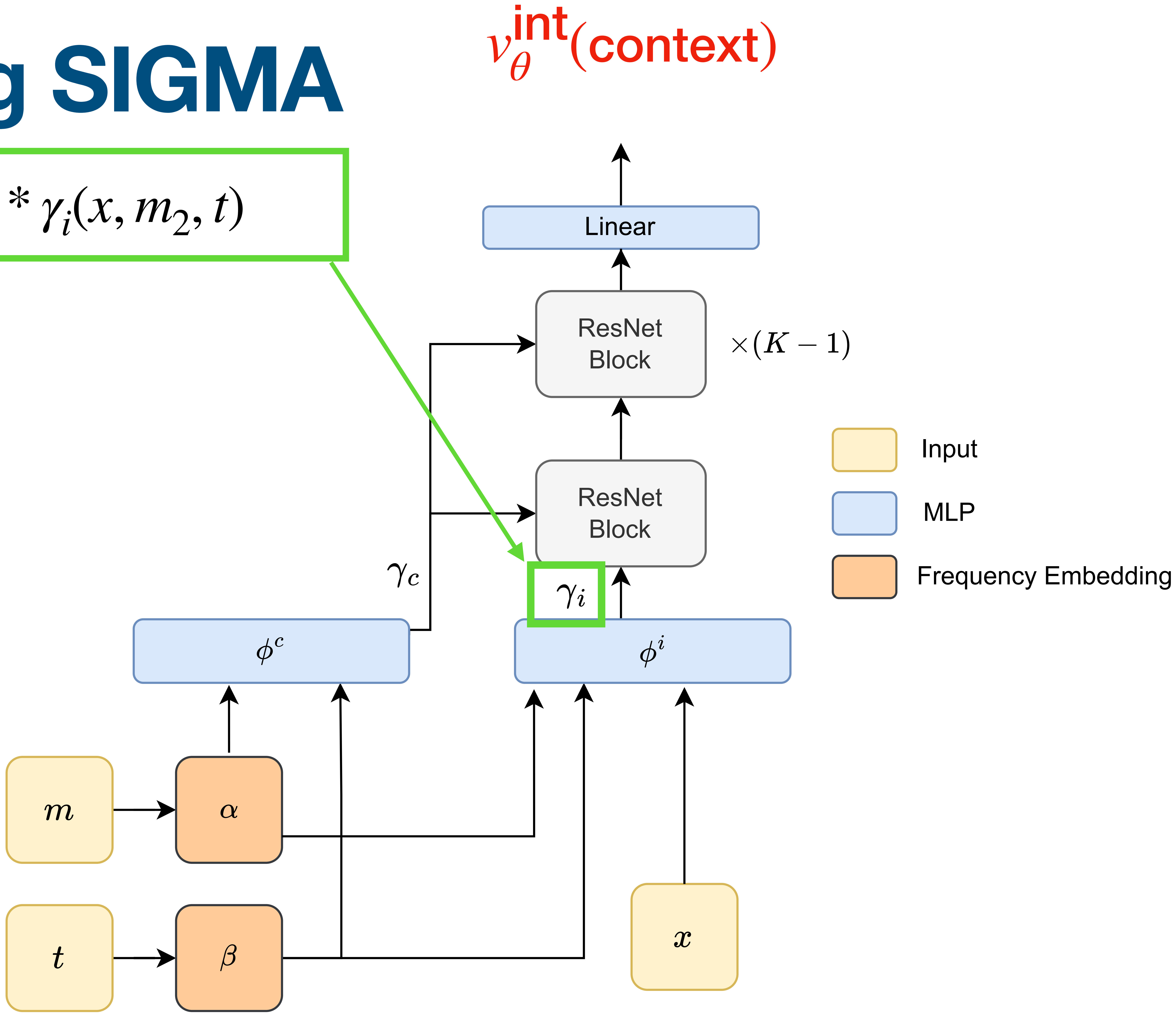




Interpolation using SIGMA

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

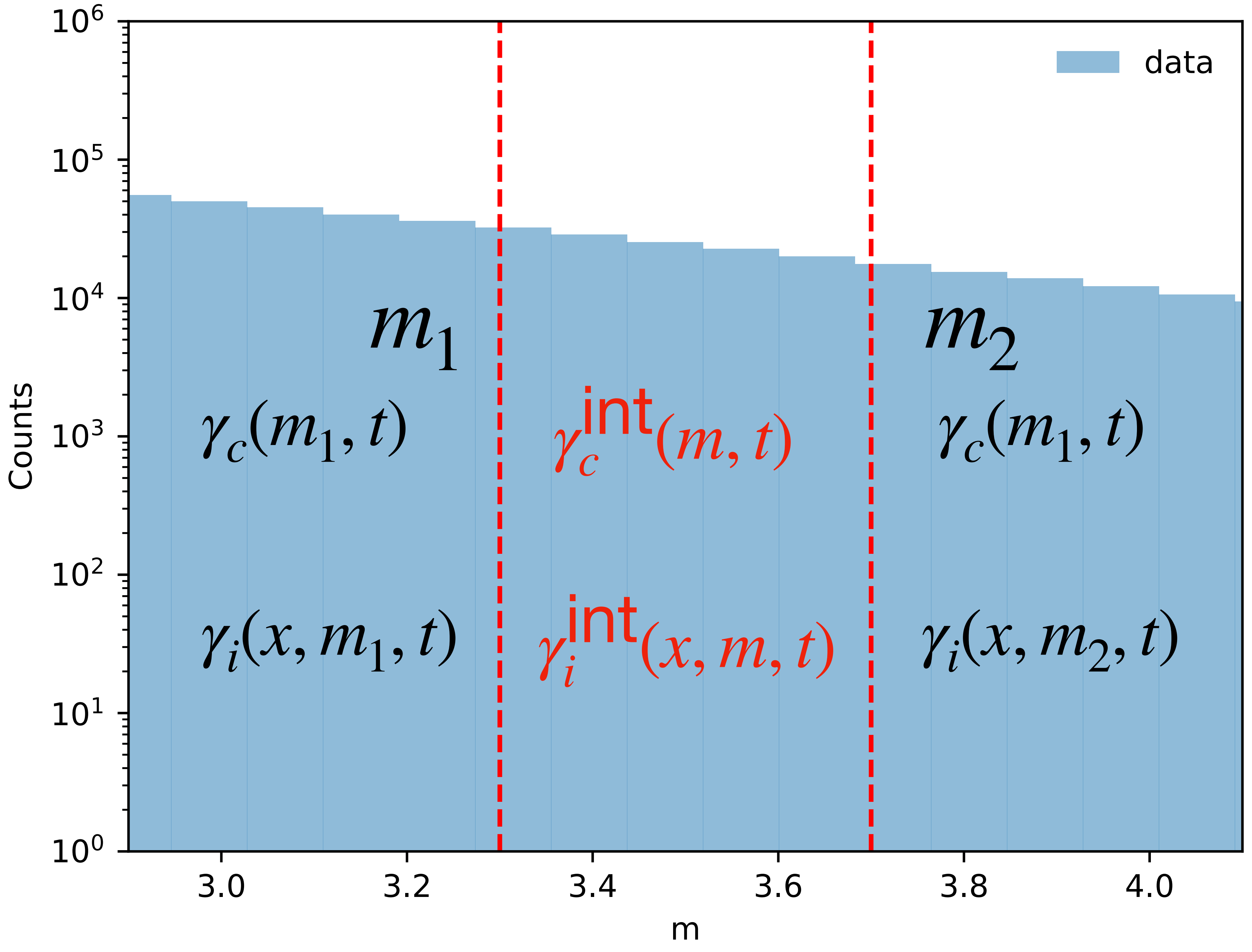


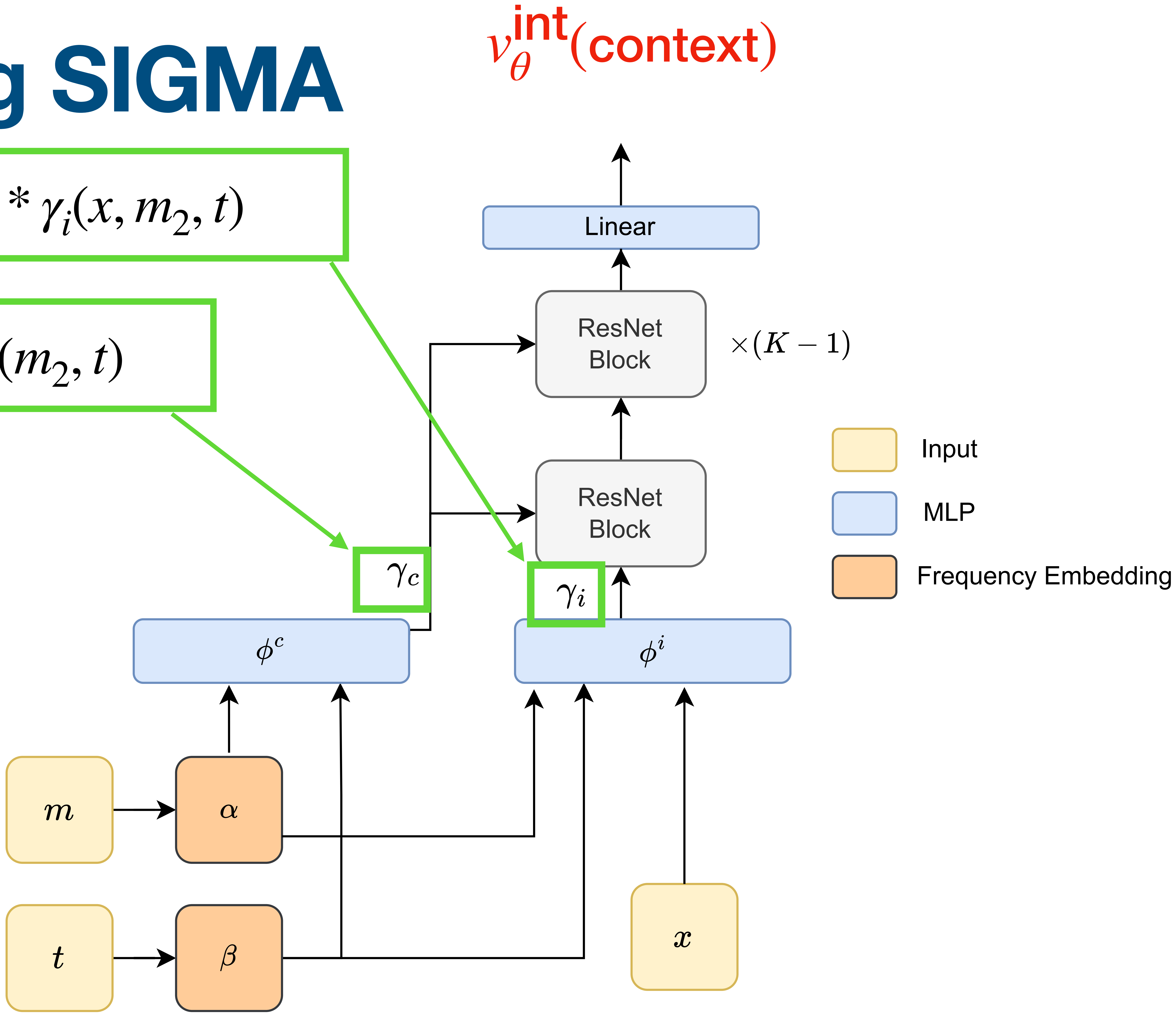


Interpolation using SIGMA

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

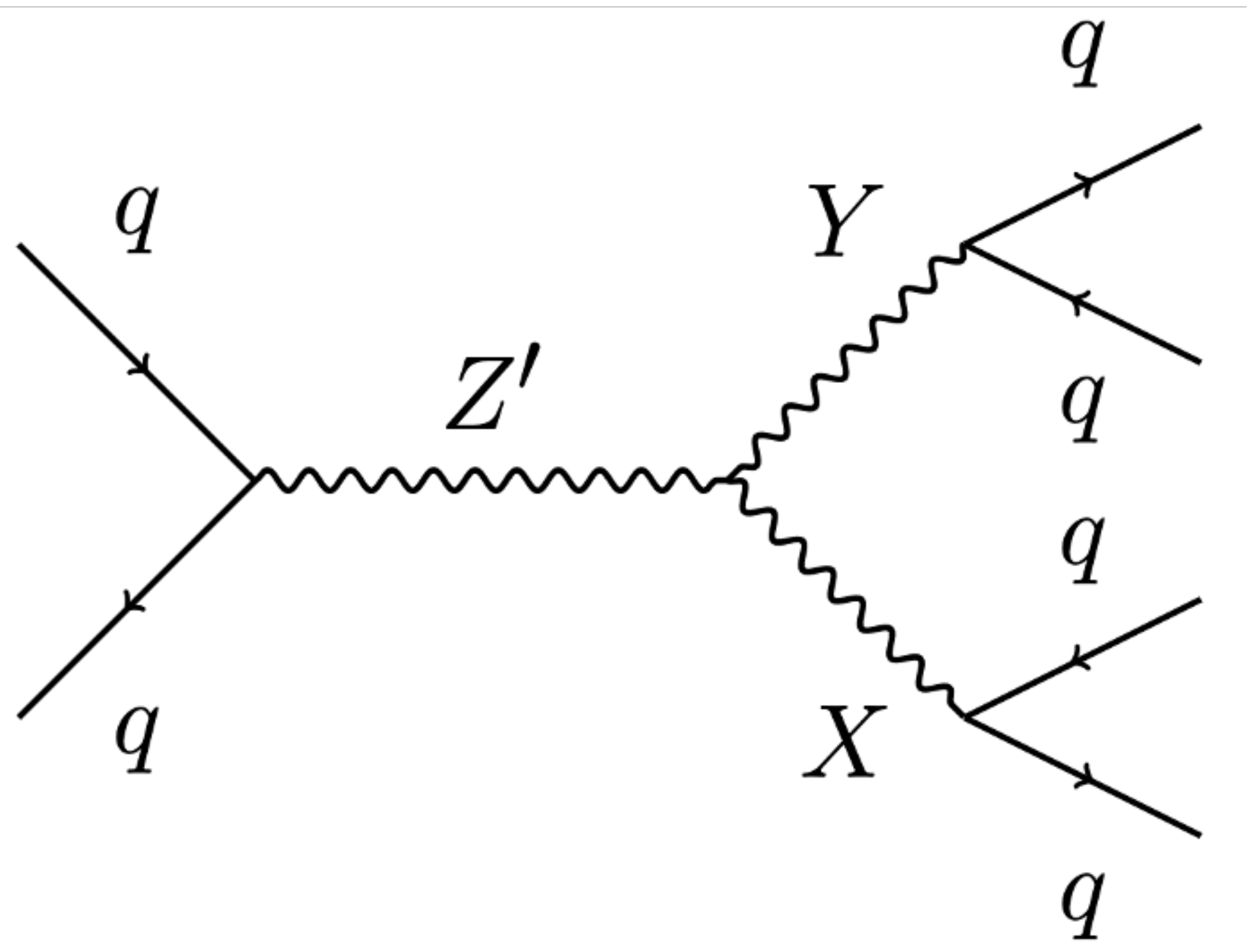
$$\gamma_c^{\text{int}}(m, t) = \xi * \gamma_c(m_1, t) + (1 - \xi) * \gamma_c(m_2, t)$$





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 $[m_{J_1}, m_{J_2} - m_{J_1}, \tau_{21}^{J_1}, \tau_{21}^{J_2}, \Delta R]$
- The SR : $3.3TeV < m_{JJ} < 3.7TeV$.

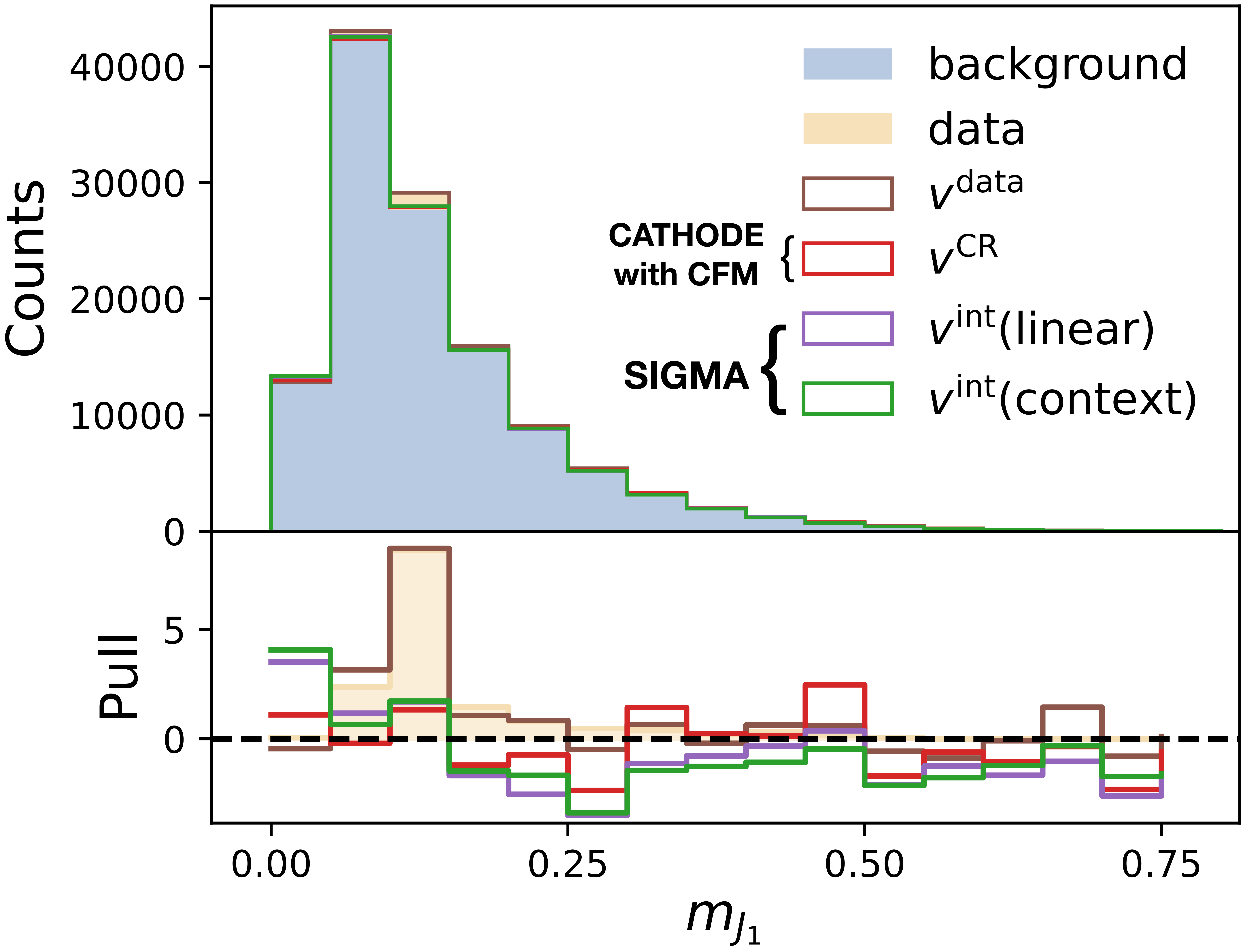


The LHC Olympics 2020: A Community Challenge for Anomaly Detection in High Energy Physics : arXiv:2101.08320

Samples

Samples

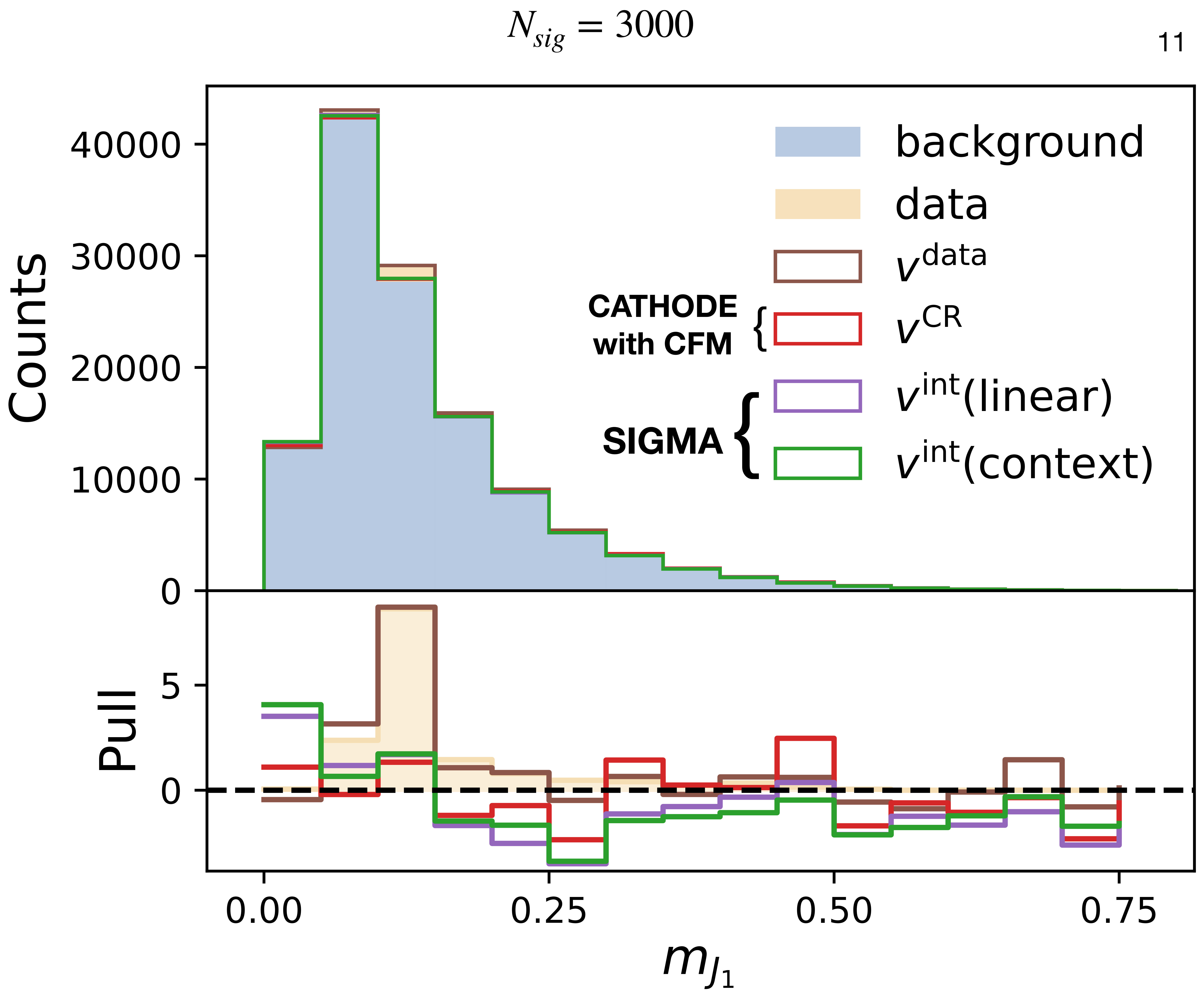
$N_{sig} = 3000$



Samples

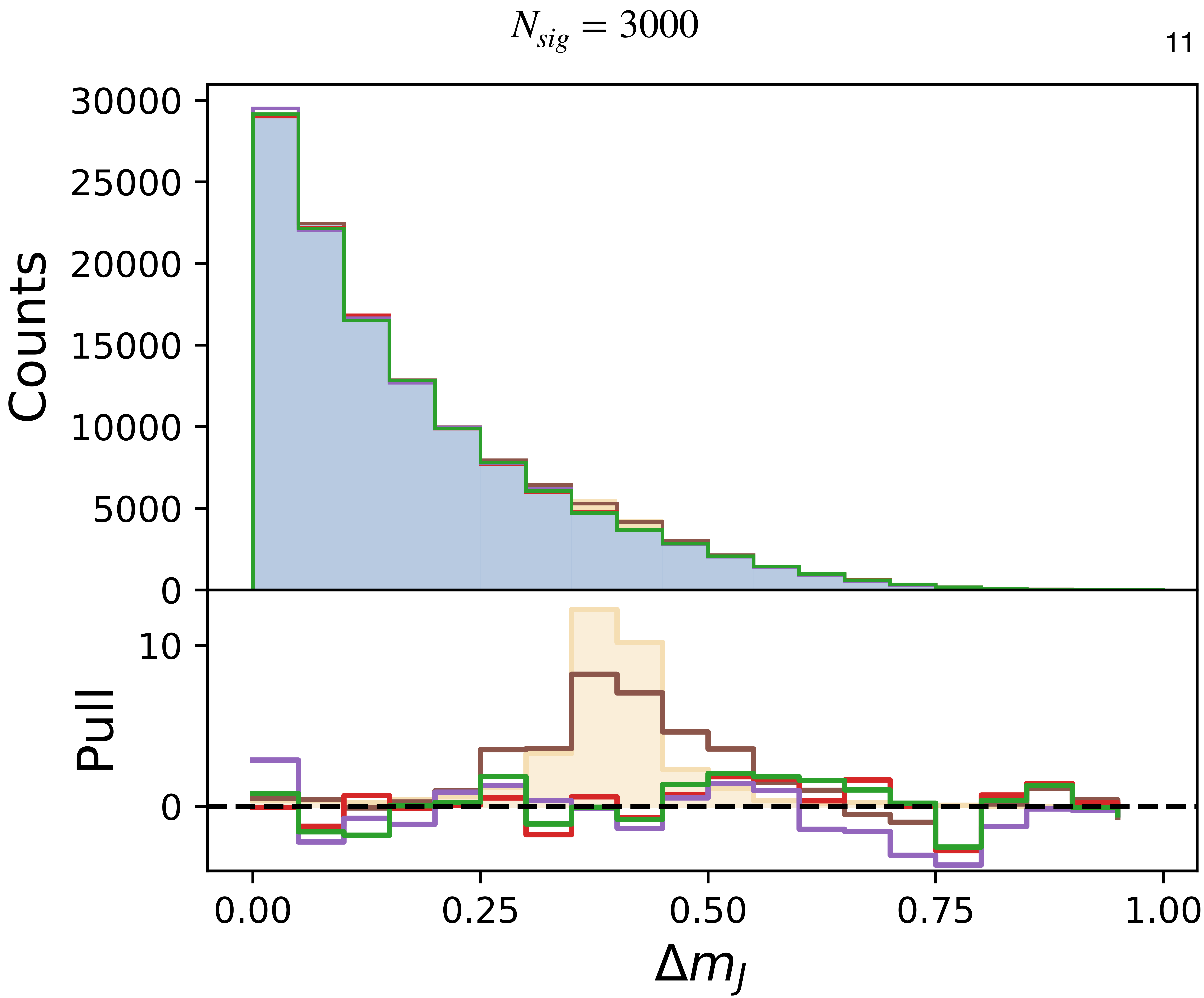
- The model trained on data, v_θ^{data} learns the signal.

- 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

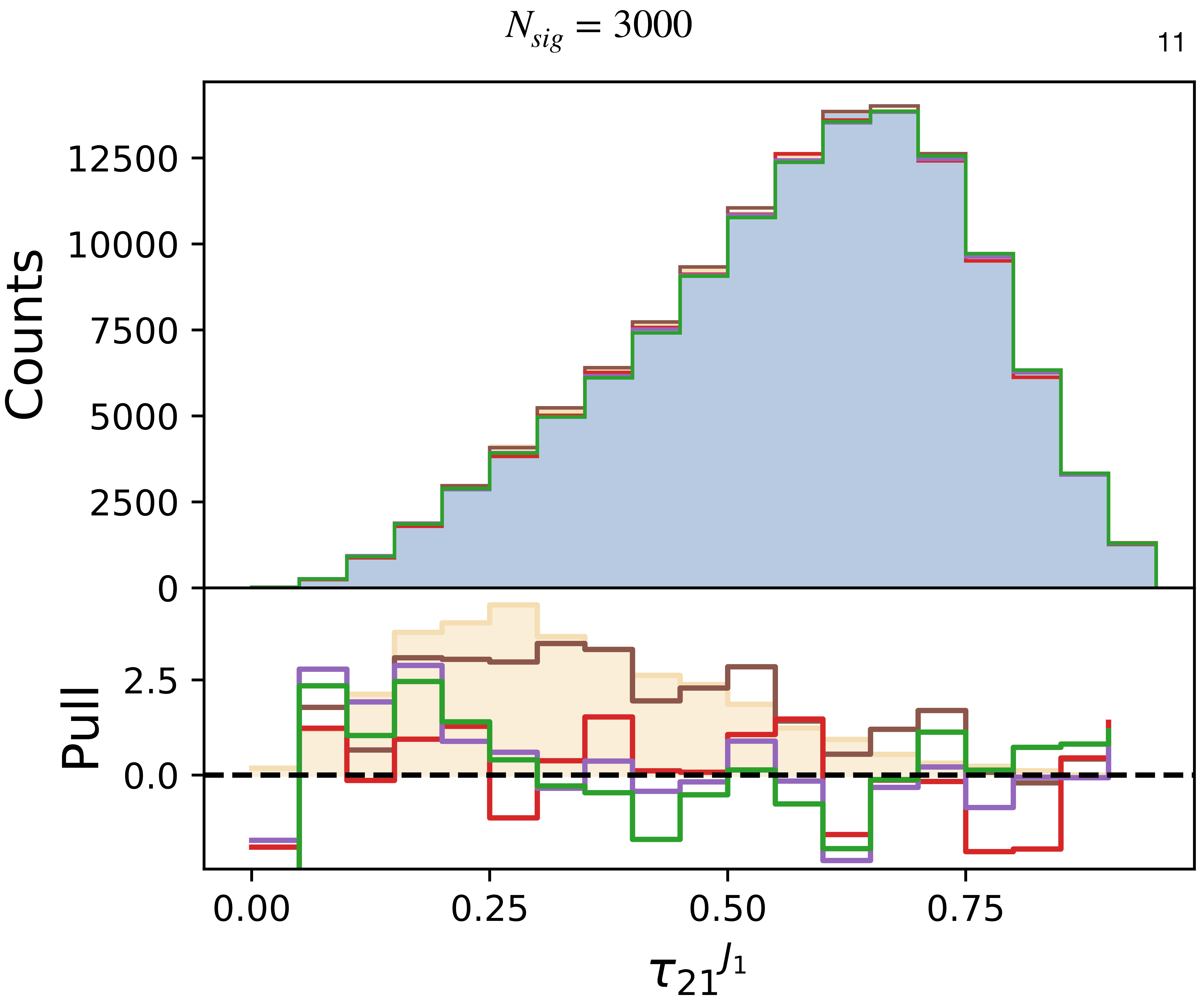
- The model trained on data, v_θ^{data} learns the signal.
- The previous interpolation method v_θ^{CR} and the new interpolation methods v_θ^{int}(linear) and v_θ^{int} (context) are able to remove the signal



Samples

- The model trained on data, v_θ^{data} learns the signal.

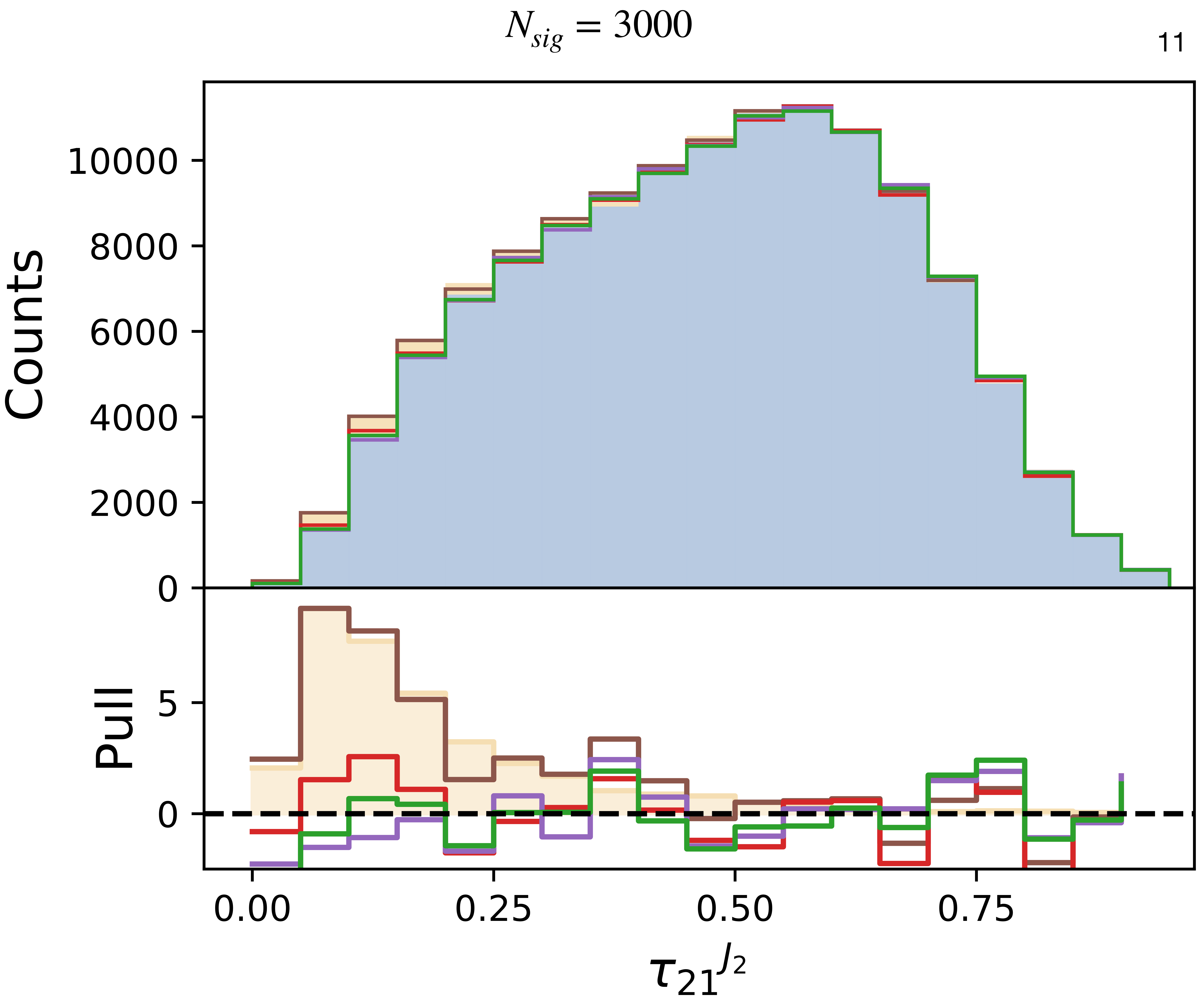
- The previous interpolation method v_θ^{CR} and the new interpolation methods v_θ^{int}(linear) and v_θ^{int} (context) are able to remove the signal



Samples

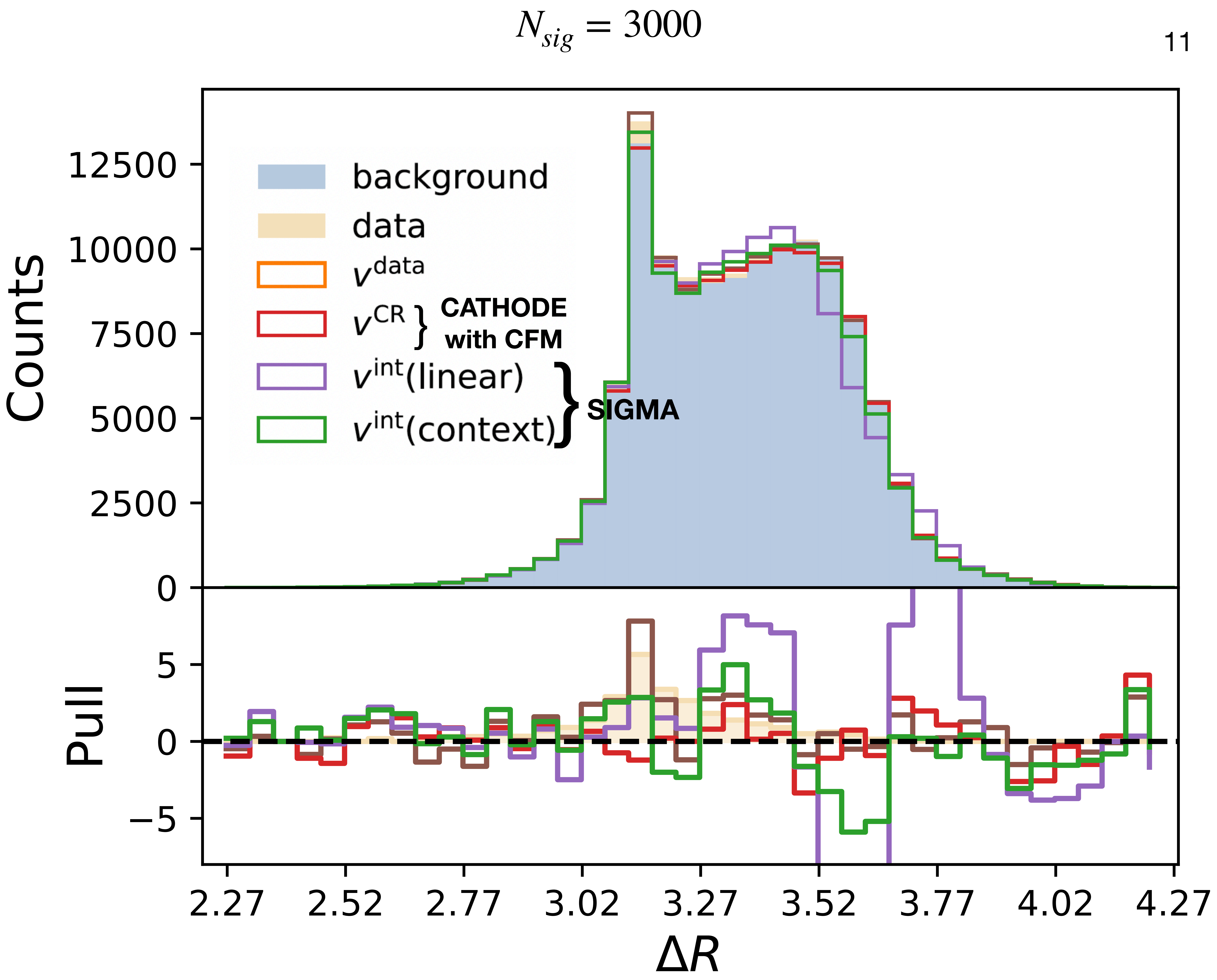
- The model trained on data, v_θ^{data} learns the signal.

- 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



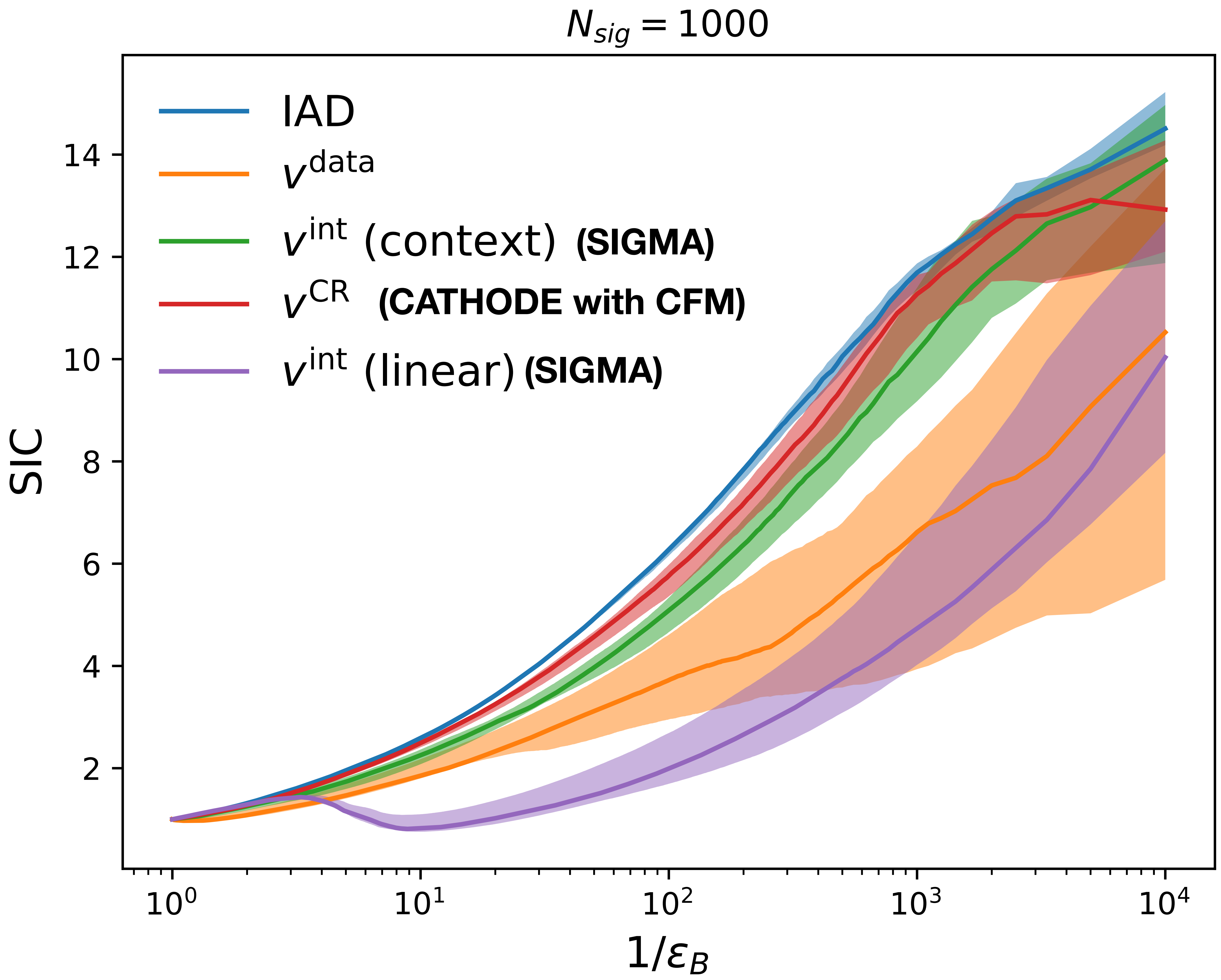
Samples

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



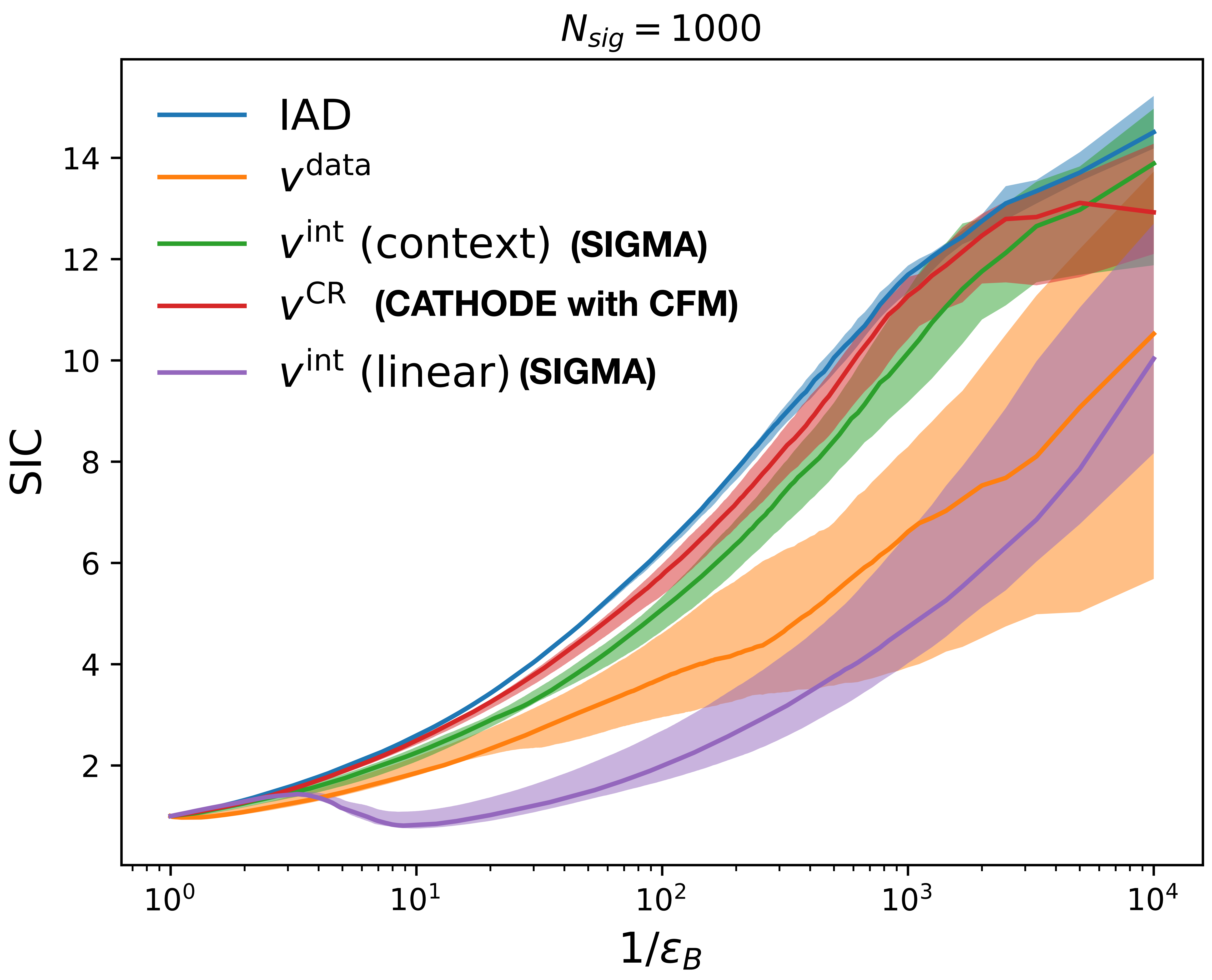
Anomaly detection performance

Anomaly detection performance



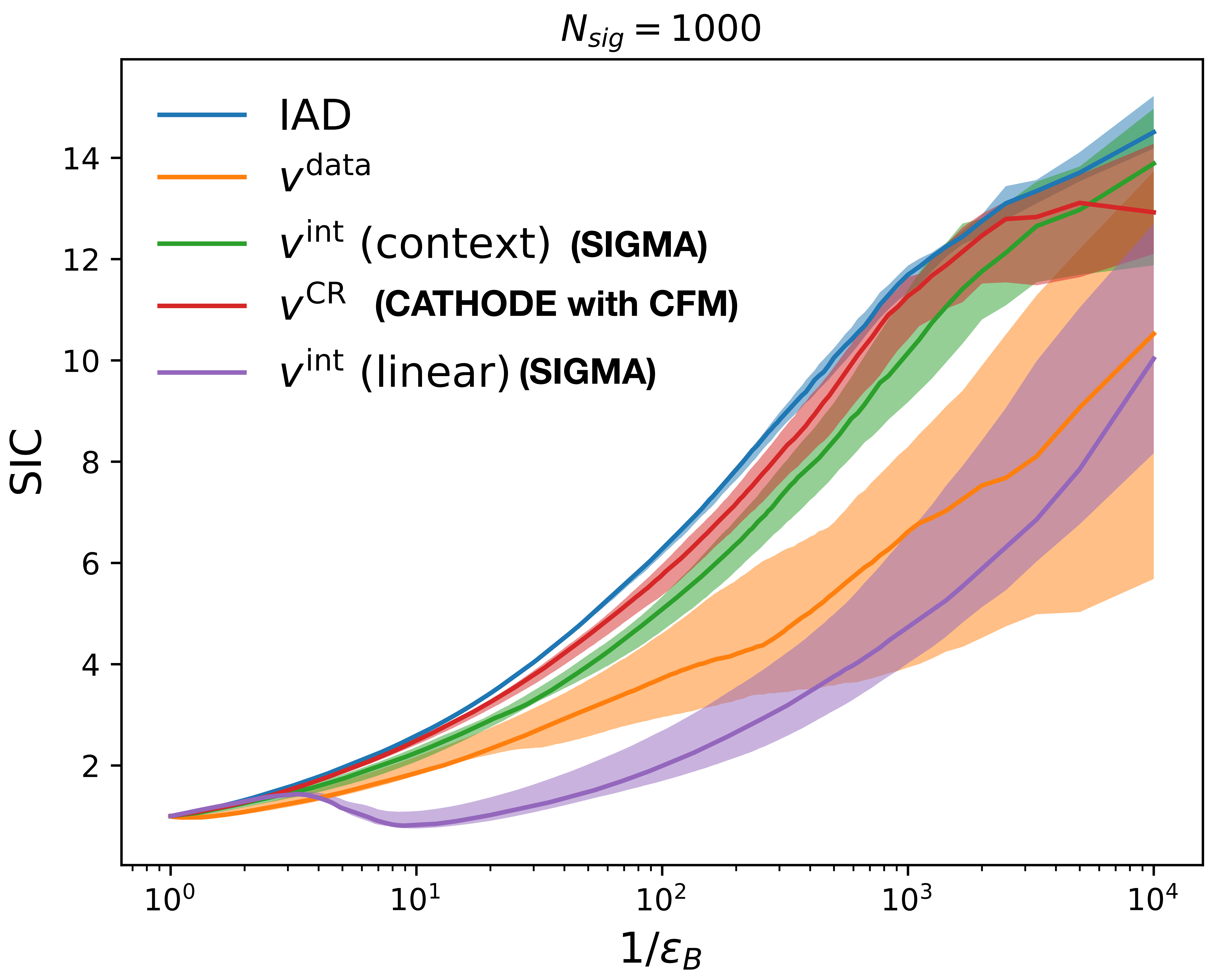
Anomaly detection performance

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



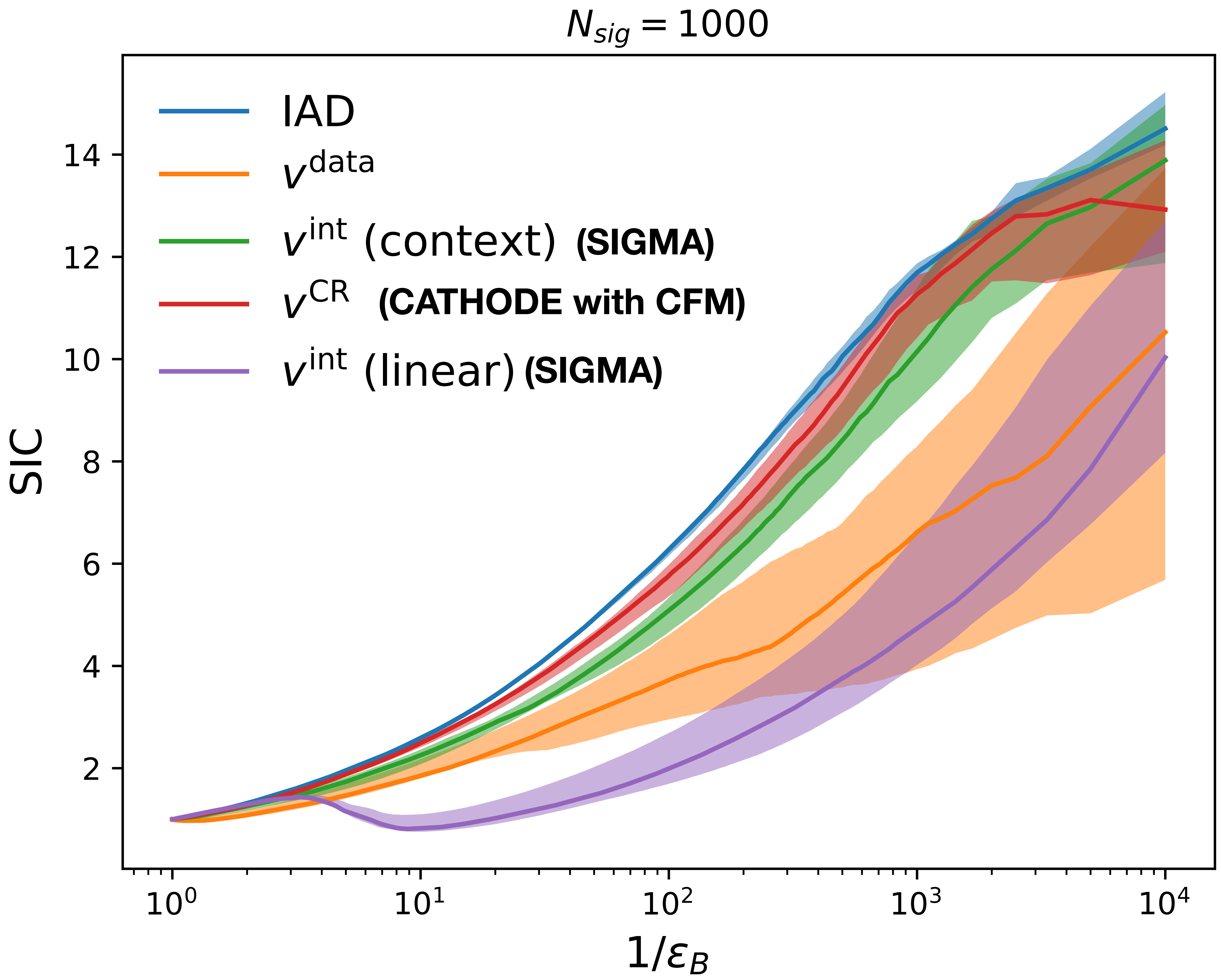
Anomaly detection performance

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



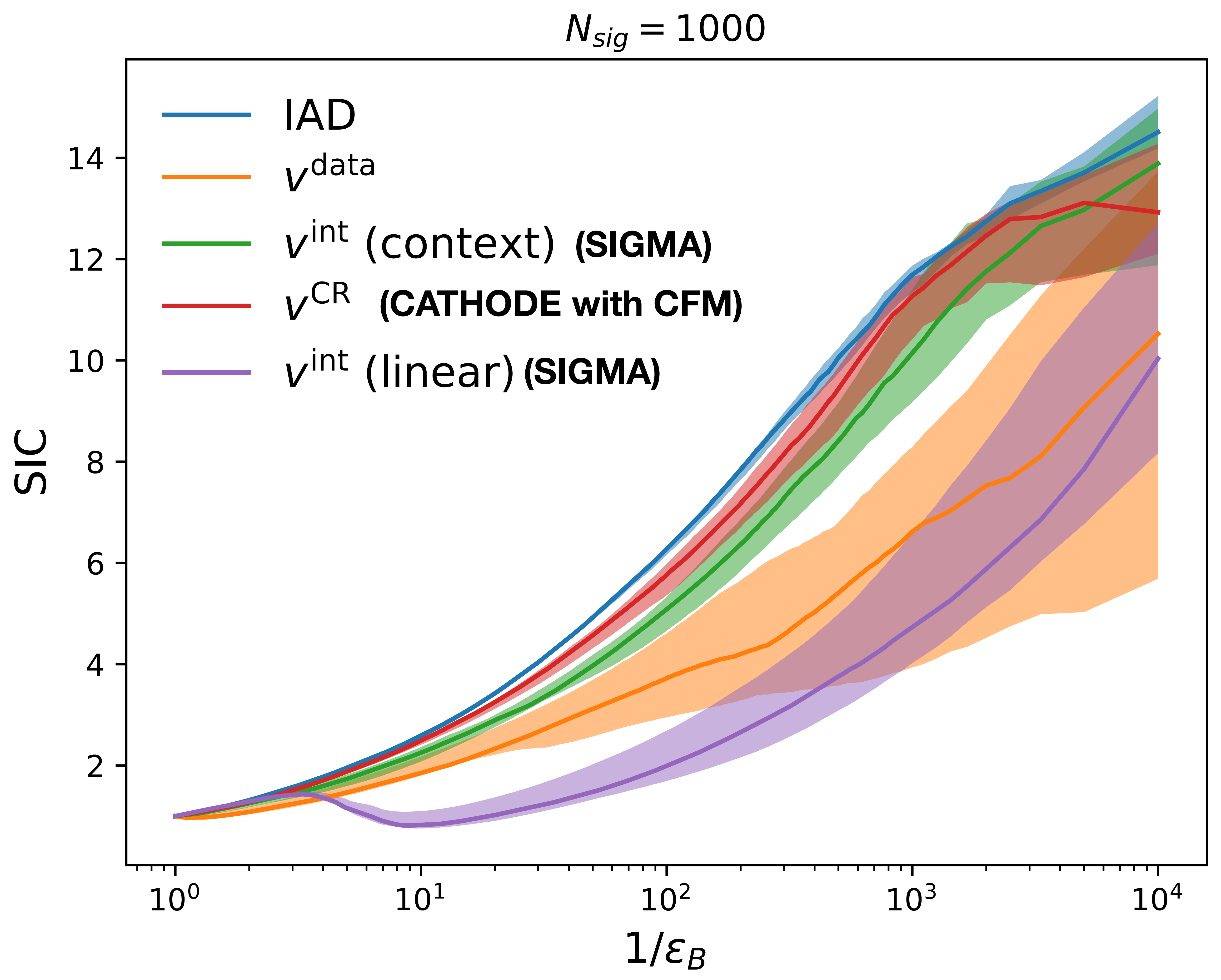
Anomaly detection performance

- 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}.



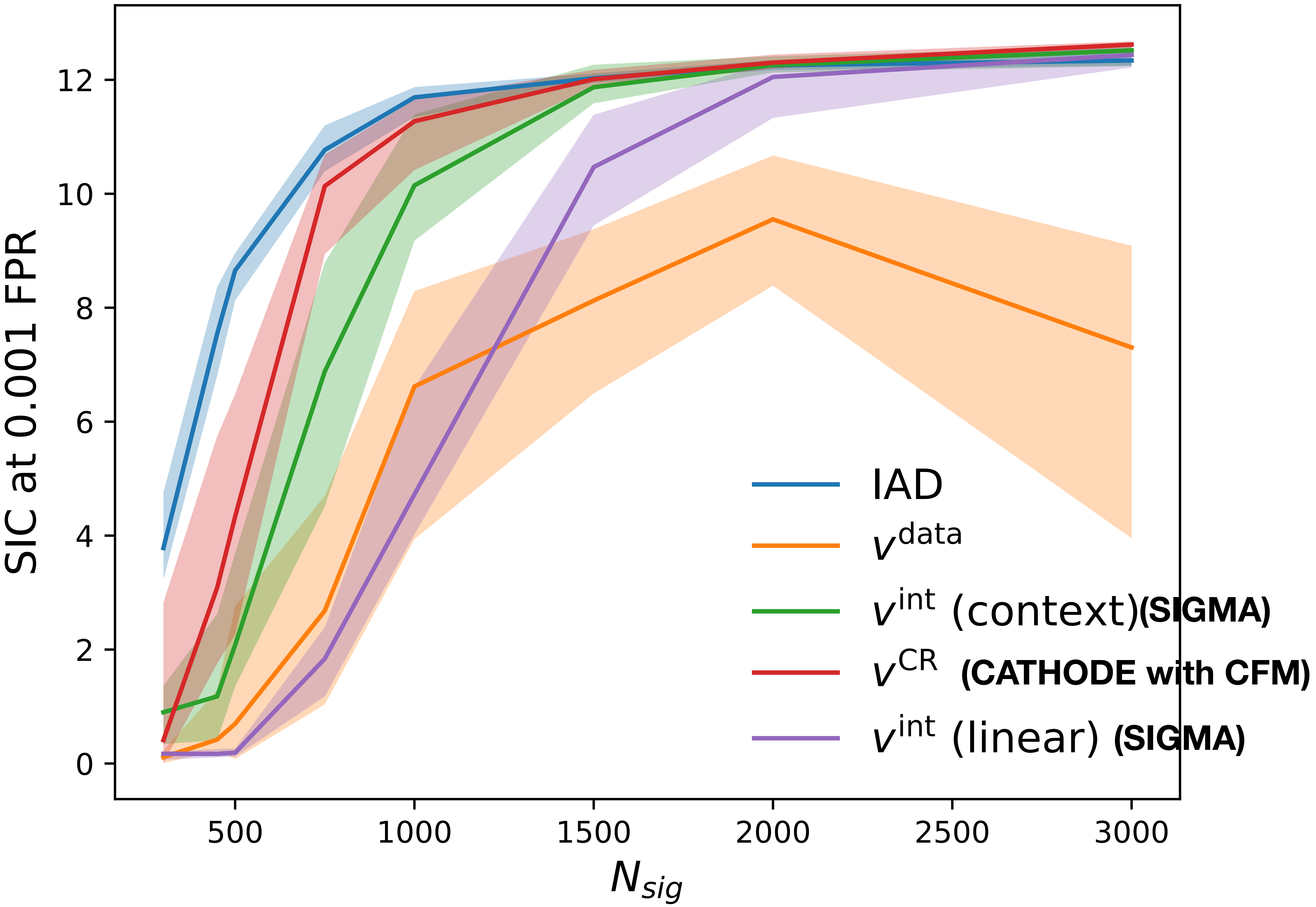
Anomaly detection performance

- 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

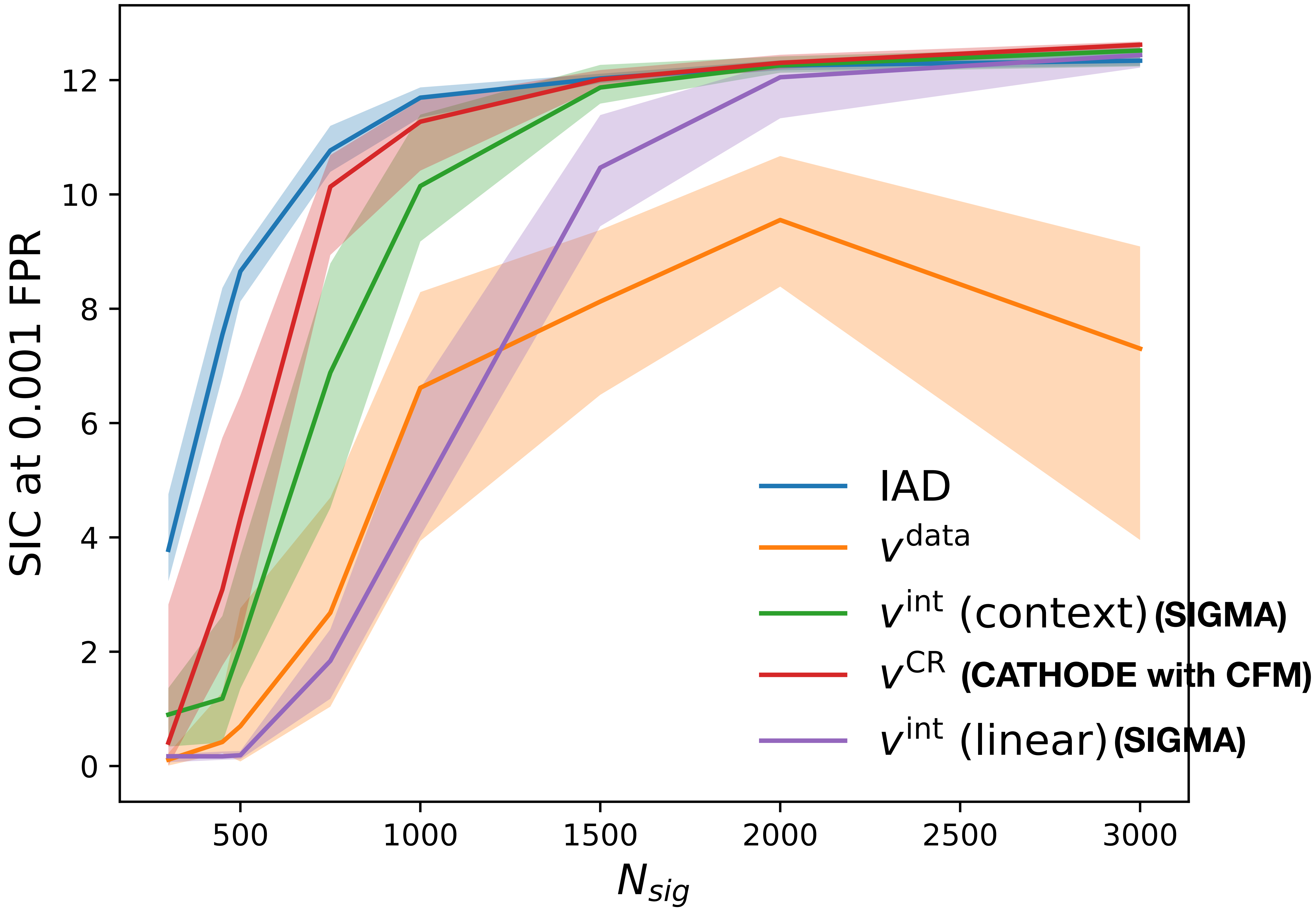
Method	Generative Model	Timing
CATHODE/ANODE	Normalizing Flows	3 hours per SR
CATHODE/ANODE	Flow Matching	30 mins per SR
CURTAINS4F4	Normalizing Flows	3 hours (base model) + 7 mins per SR
RAD-OT	Optimal Transport	10 mins per SR
SIGMA (ours)	Flow Matching	30 mins (training) + 30 secs per SR

How to select best interpolated model?

How to select best interpolated model?

Open question!

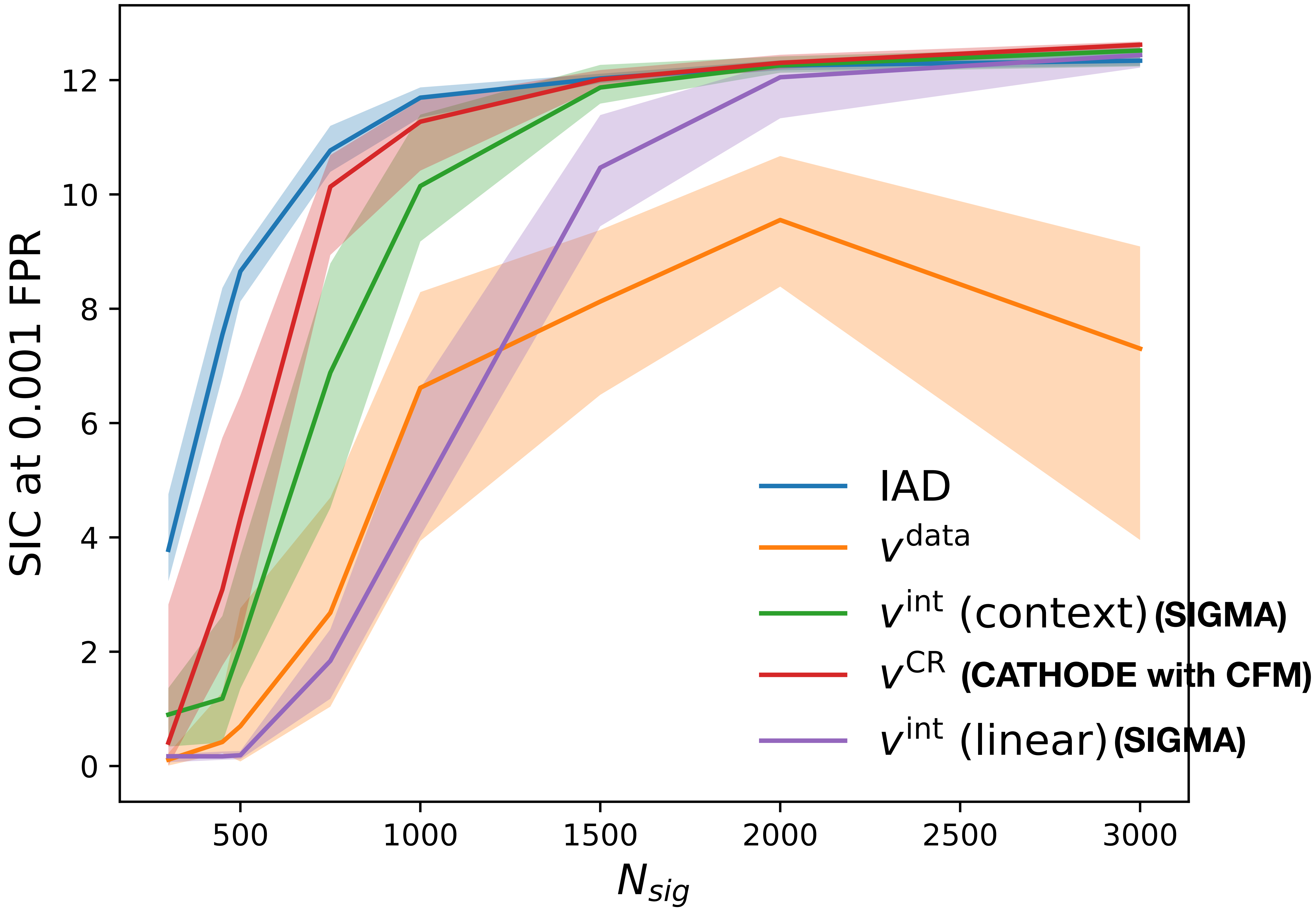
- The SIC is very sensitive to bad background templates.



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.



Conclusions

Conclusions

- SIGMA re-uses a single generative model trained on all of the data, with a subsequent interpolation of its parameters from SB into SR.

Conclusions

- SIGMA re-uses a single generative model trained on all of the data, with a subsequent interpolation of its parameters from SB into SR.
- Reduces the computational cost of SIGMA significantly relative to previous approaches such as ANODE/CATHODE, while preserving the high quality of their background templates and signal sensitivity.

Conclusions

- SIGMA re-uses a single generative model trained on all of the data, with a subsequent interpolation of its parameters from SB into SR.
- Reduces the computational cost of SIGMA significantly relative to previous approaches such as ANODE/CATHODE, while preserving the high quality of their background templates and signal sensitivity.
- Given that there was still a small performance gap between the previous, expensive interpolation method (masking out the CR) and SIGMA, on could explore further possible improvements:

Conclusions

- SIGMA re-uses a single generative model trained on all of the data, with a subsequent interpolation of its parameters from SB into SR.
- Reduces the computational cost of SIGMA significantly relative to previous approaches such as ANODE/CATHODE, while preserving the high quality of their background templates and signal sensitivity.
- Given that there was still a small performance gap between the previous, expensive interpolation method (masking out the CR) and SIGMA, on could explore further possible improvements:
    - Using diffusion models instead of flow-matching.

Conclusions

- SIGMA re-uses a single generative model trained on all of the data, with a subsequent interpolation of its parameters from SB into SR.
- Reduces the computational cost of SIGMA significantly relative to previous approaches such as ANODE/CATHODE, while preserving the high quality of their background templates and signal sensitivity.
- Given that there was still a small performance gap between the previous, expensive interpolation method (masking out the CR) and SIGMA, on could explore further possible improvements:
    - Using diffusion models instead of flow-matching.
    - Performing some kind of non-linear interpolation using more than two mass points in the control region.

THANK YOU