Anomaly Detection in Collider Physics via Factorized Observables

Raymond Wynne

Based on work with Eric Metodiev and Jesse Thaler

Preprint and code coming soon!

ML4Jets 2023 - 07 November 2023

Anomaly Detection in Collider Physics via Factorized Observables

How to Become a FORCE Wielder

Raymond Wynne

Based on work with Eric Metodiev and Jesse Thaler

Preprint and code coming soon!

ML4Jets 2023 - 07 November 2023

Introduction

Introduction

Current set of anomaly detection methods come with specific assumptions

Introduction

Current set of anomaly detection methods come with specific assumptions

Examples:

Introduction

Current set of anomaly detection methods come with specific assumptions

Examples:

- Comparison between data and simulation [1709.01087, 1806.02350, 1807.06038]

Introduction

Current set of anomaly detection methods come with specific assumptions

Examples:

- Comparison between data and simulation [1709.01087, 1806.02350, 1807.06038]
- New physics appears as resonance [1805.02664, 1902.02634, 2009.02205]

Introduction

Current set of anomaly detection methods come with specific assumptions

Examples:

- Comparison between data and simulation [1709.01087, 1806.02350, 1807.06038]
- New physics appears as resonance [1805.02664, 1902.02634, 2009.02205]
- Autoencoder operationally defines anomaly [1808.08979, 1903.02032, 2010.05531]

Introduction

Current set of anomaly detection methods come with specific assumptions

Examples:

- Comparison between data and simulation [1709.01087, 1806.02350, 1807.06038]
- New physics appears as resonance [1805.02664, 1902.02634, 2009.02205]
- Autoencoder operationally defines anomaly [1808.08979, 1903.02032, 2010.05531]

Introduce new method with different assumption:

Introduction

Current set of anomaly detection methods come with specific assumptions

Examples:

- Comparison between data and simulation [1709.01087, 1806.02350, 1807.06038]
- New physics appears as resonance [1805.02664, 1902.02634, 2009.02205]
- Autoencoder operationally defines anomaly [1808.08979, 1903.02032, 2010.05531]

Introduce new method with different assumption: **factorization** [1708.02949, 1802.00008, 2002.12376]

What is factorization?

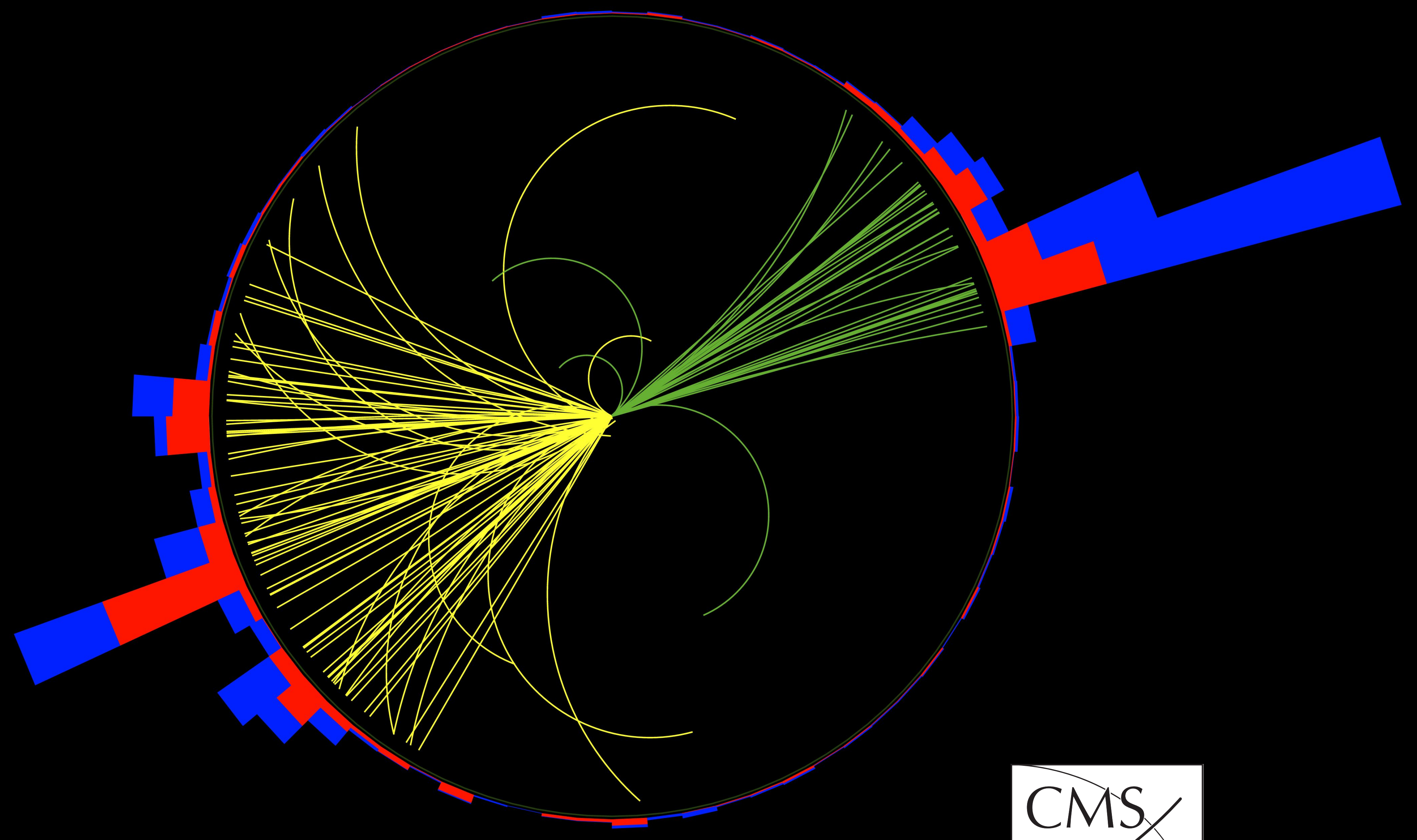
What is factorization?

Factorization: physics at different energy scales approximately independent

What is factorization?

Factorization: physics at different energy scales approximately independent

Canonical example: ***jets***

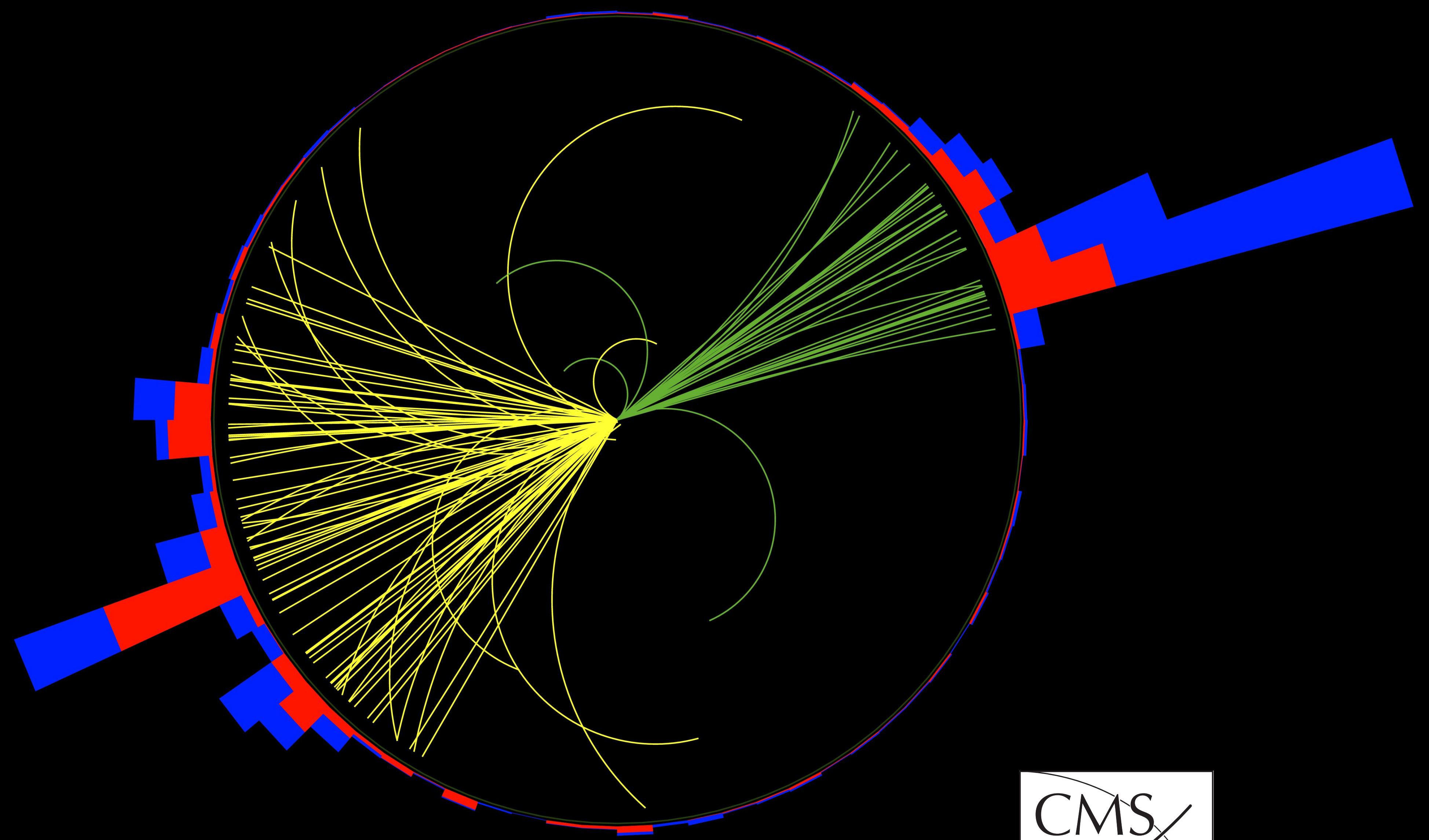


What is factorization?

Factorization: physics at different energy scales approximately independent

Given $p_T, y, \mathcal{O}$, this gives

Canonical example: ***jets***



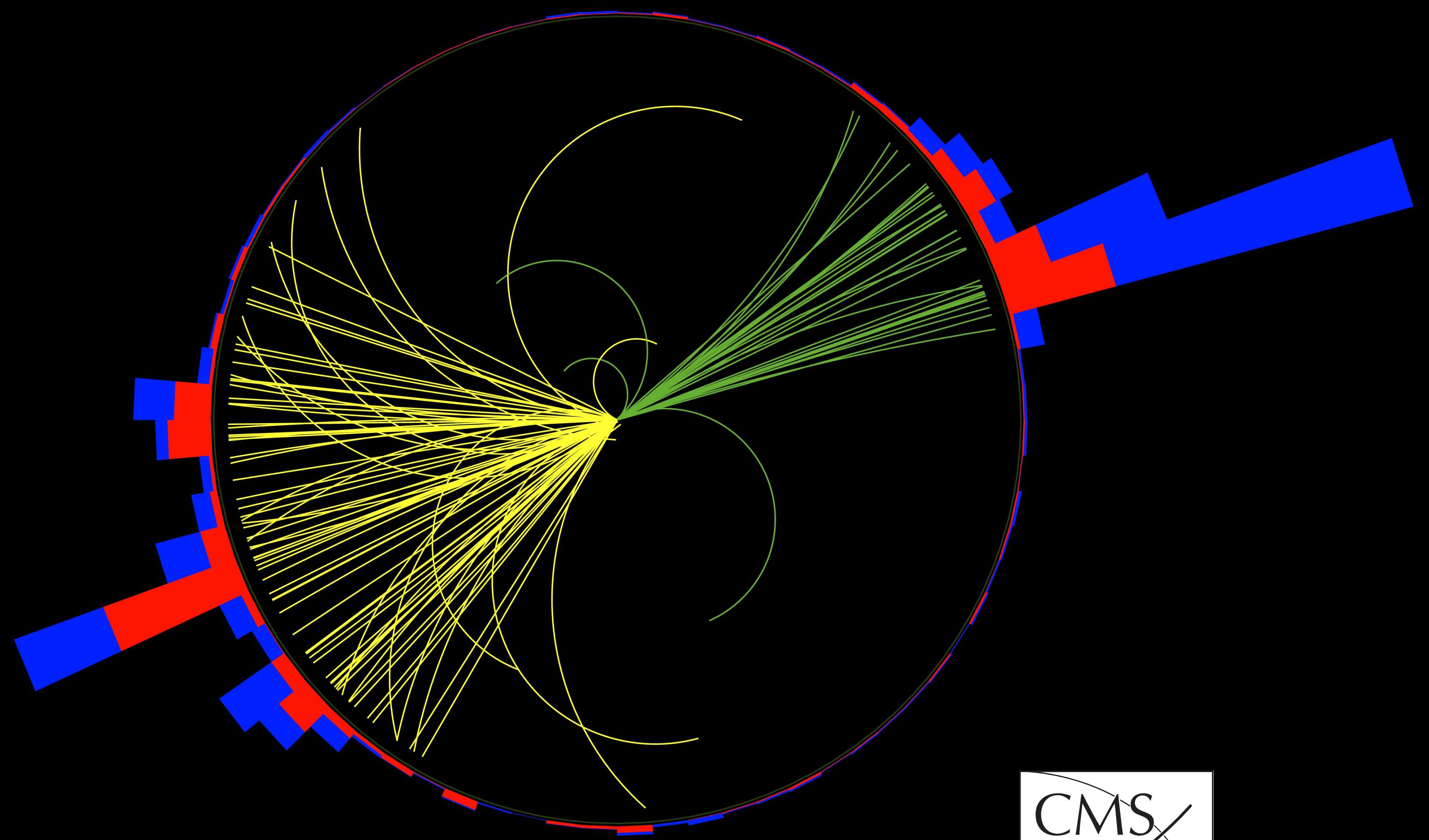
What is factorization?

Factorization: physics at different energy scales approximately independent

Given $p_T, y, \mathcal{O}$, this gives

$$p(p_T, y, \mathcal{O}) \approx \sum_i f_i \, p_i(p_T, y) \, p_i(\mathcal{O} \,|\, p_T)$$

Canonical example: ***jets***



What is factorization?

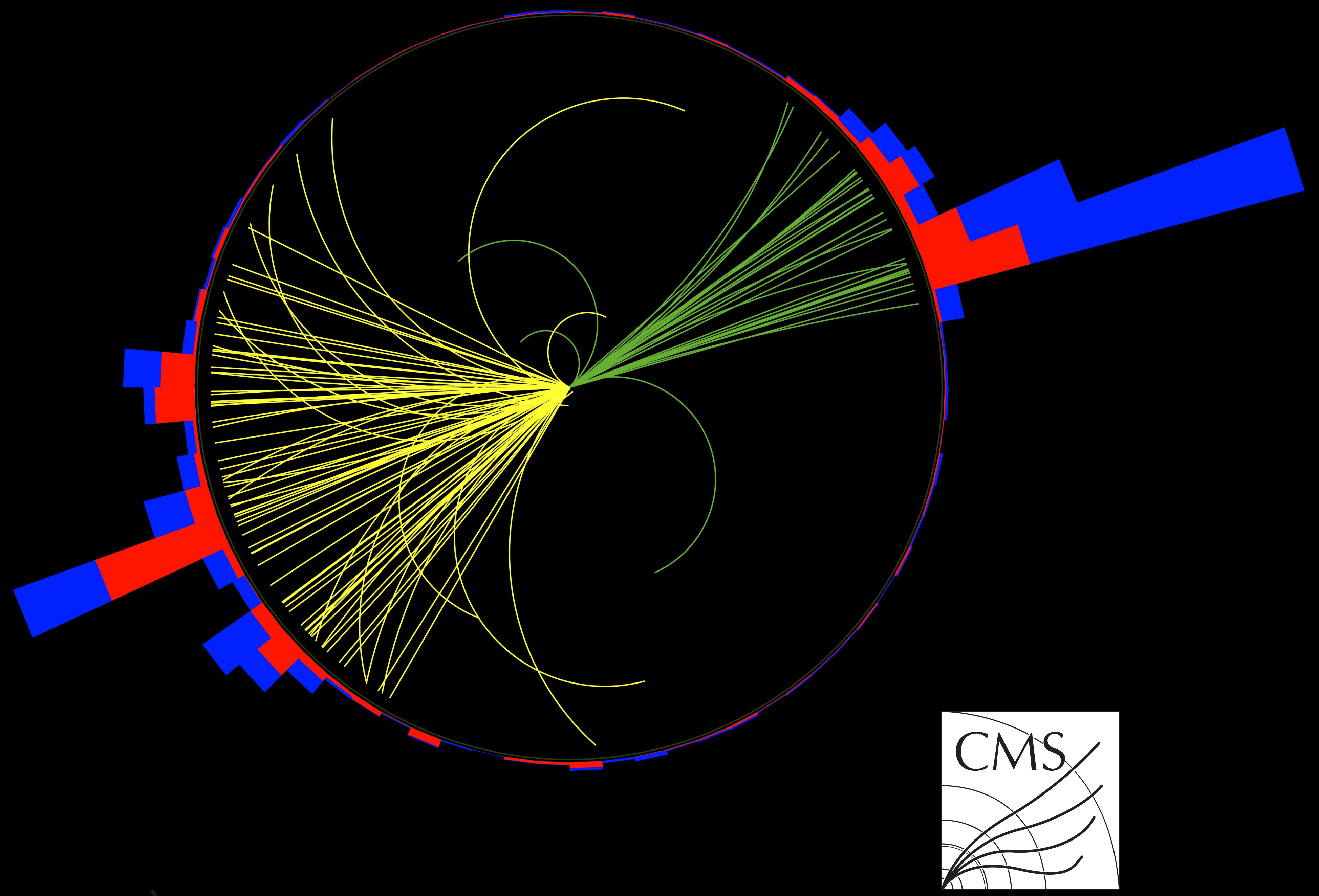
Factorization: physics at different energy scales approximately independent

Given $p_T, y, \mathcal{O}$, this gives

$$p(p_T, y, \mathcal{O}) \approx \sum_i f_i\, p_i(p_T, y)\, p_i(\mathcal{O} \,|\, p_T)$$

Marginalizing out y and taking a scale-/boost-invariant $\mathcal{O}$, this gives

Canonical example: ***jets***



What is factorization?

Factorization: physics at different energy scales approximately independent

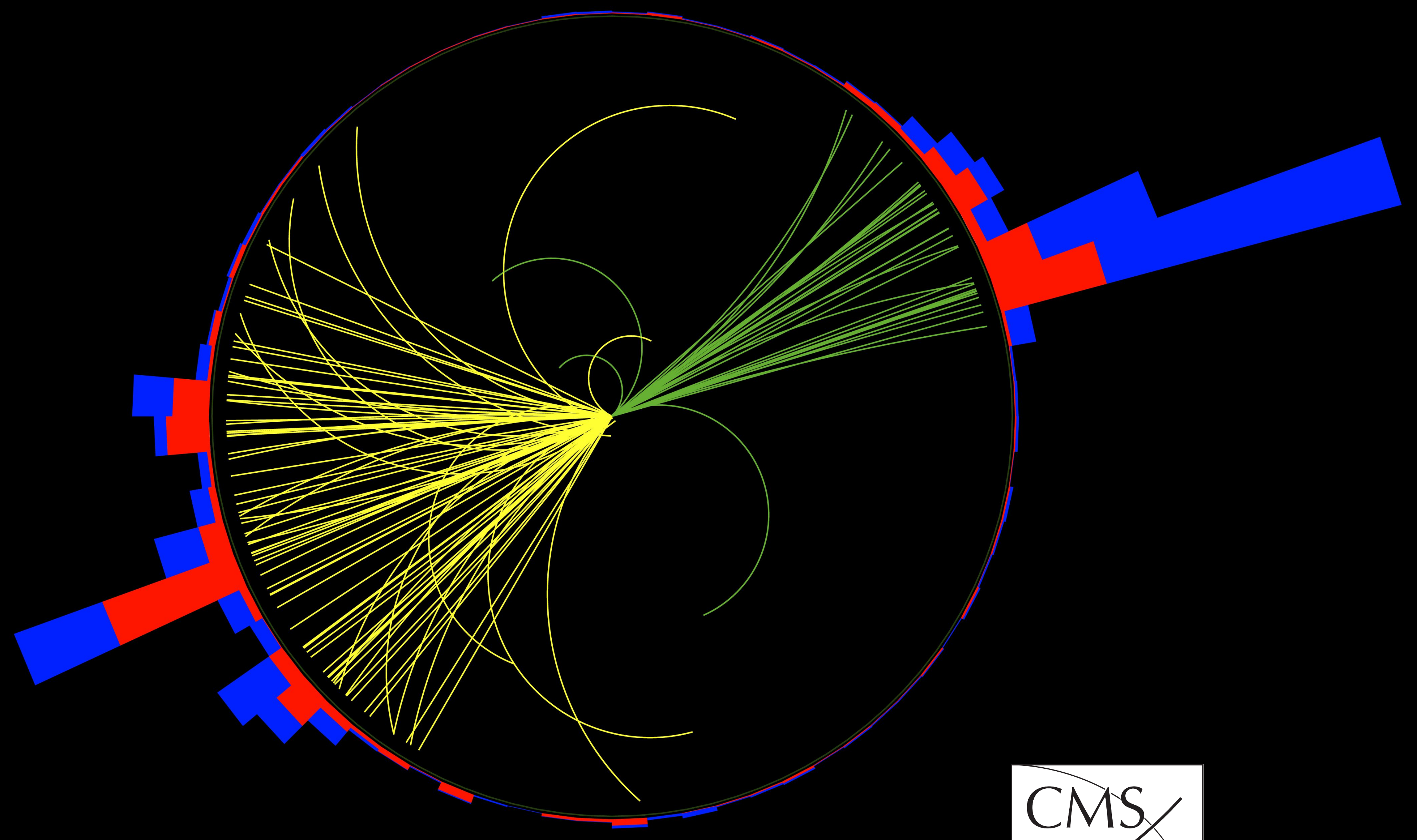
Given $p_T, y, \mathcal{O}$, this gives

$$p(p_T, y, \mathcal{O}) \approx \sum_i f_i \, p_i(p_T, y) \, p_i(\mathcal{O} \,|\, p_T)$$

Marginalizing out y and taking a scale-/boost-invariant $\mathcal{O}$, this gives

$$p(p_T, \mathcal{O}) \approx \sum_i f_i \, p_i(p_T) \, p_i(\mathcal{O})$$

Canonical example: ***jets***



What is factorization?

Factorization: physics at different energy scales approximately independent

Given $p_T, y, \mathcal{O}$, this gives

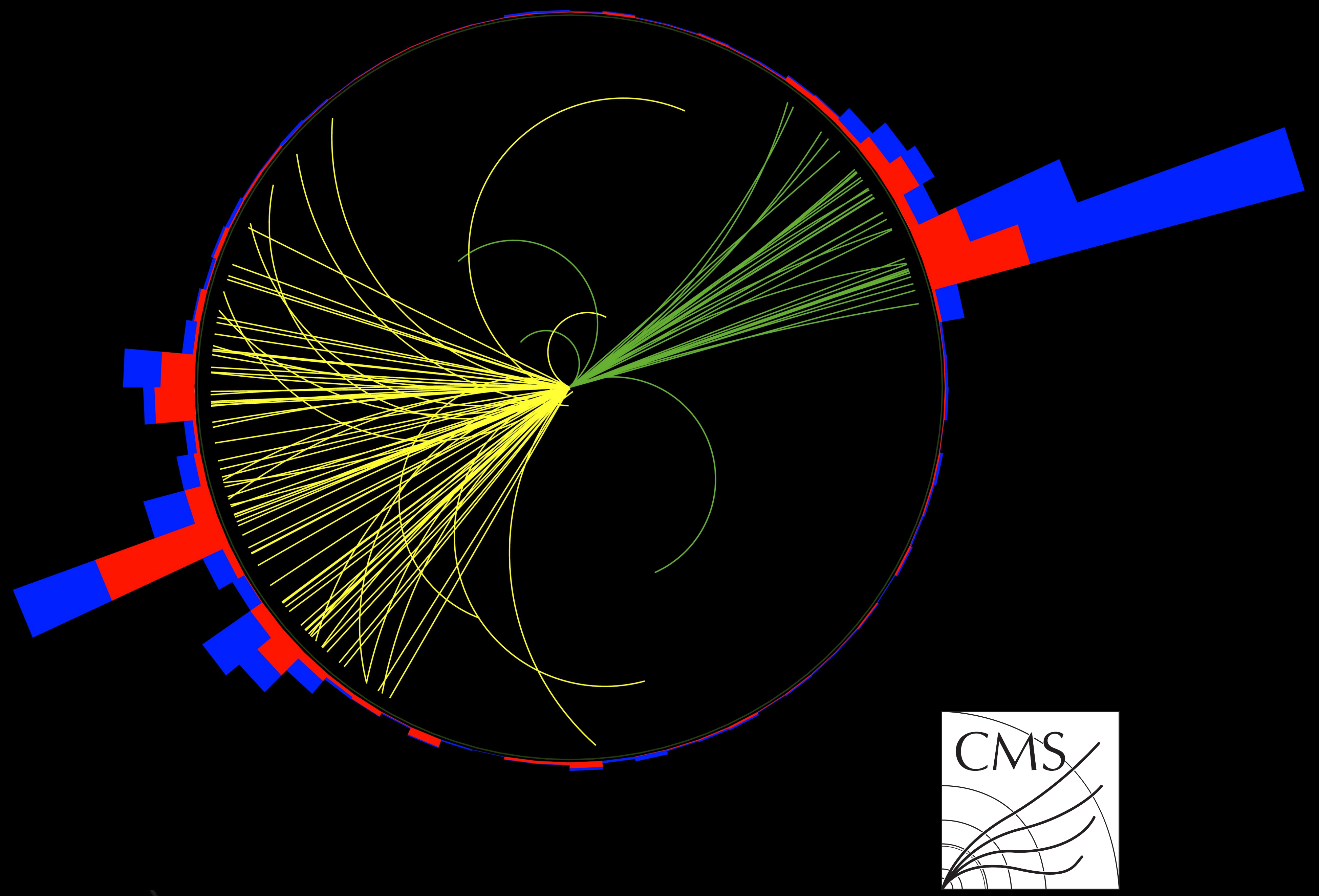
$$p(p_T, y, \mathcal{O}) \approx \sum_i f_i \, p_i(p_T, y) \, p_i(\mathcal{O} \,|\, p_T)$$

Marginalizing out y and taking a scale-/boost-invariant $\mathcal{O}$, this gives

$$p(p_T, \mathcal{O}) \approx \sum_i f_i \, p_i(p_T) \, p_i(\mathcal{O})$$

Ex $\mathcal{O}$:

Canonical example: ***jets***



What is factorization?

Factorization: physics at different energy scales approximately independent

Given $p_T, y, \mathcal{O}$, this gives

$$p(p_T, y, \mathcal{O}) \approx \sum_i f_i \, p_i(p_T, y) \, p_i(\mathcal{O} \,|\, p_T)$$

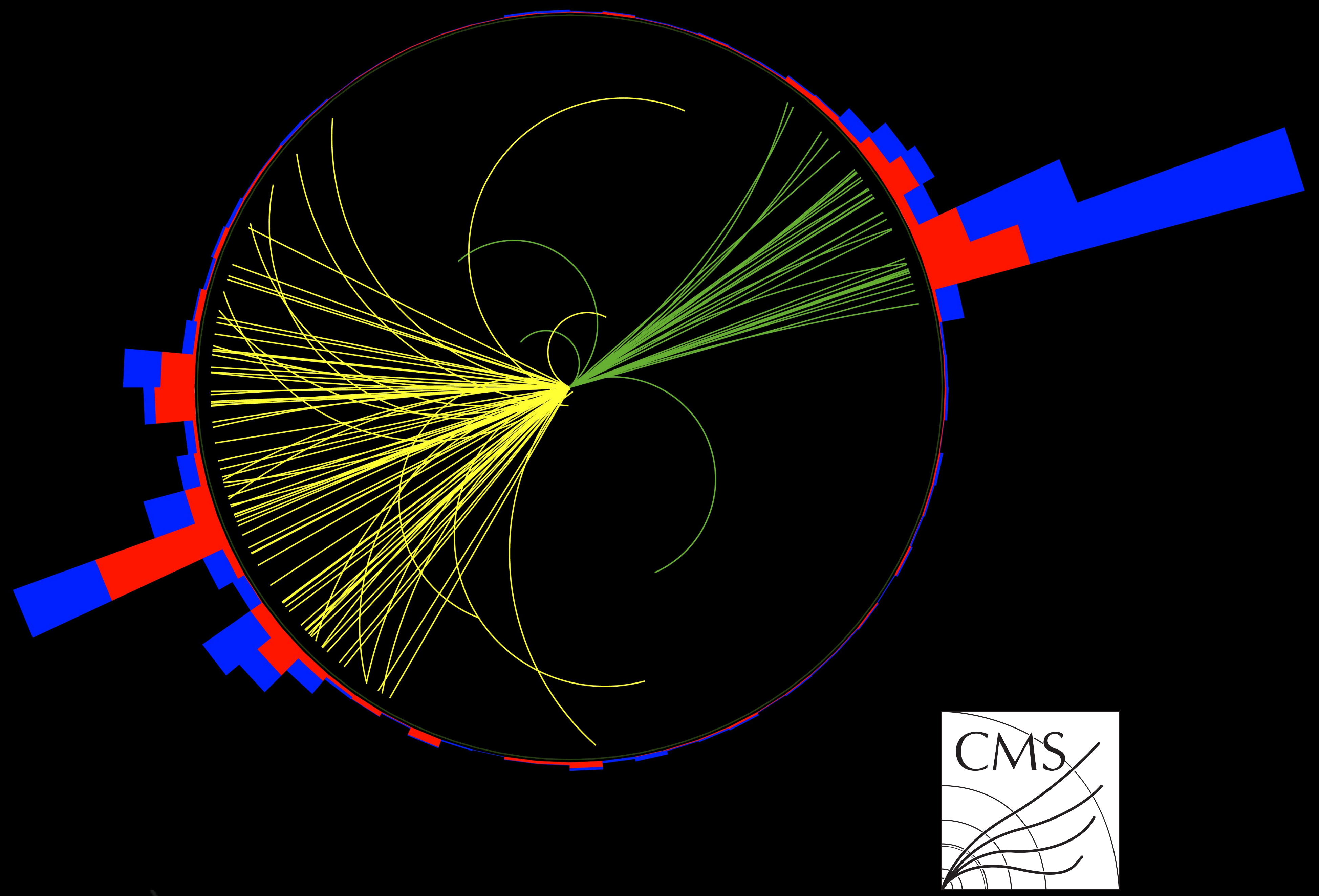
Marginalizing out y and taking a scale-/boost-invariant $\mathcal{O}$, this gives

$$p(p_T, \mathcal{O}) \approx \sum_i f_i \, p_i(p_T) \, p_i(\mathcal{O})$$

Ex $\mathcal{O}$: N – subjettiness ratios,

[1108.2701]

Canonical example: ***jets***



What is factorization?

Factorization: physics at different energy scales approximately independent

Given $p_T, y, \mathcal{O}$, this gives

$$p(p_T, y, \mathcal{O}) \approx \sum_i f_i\, p_i(p_T, y)\, p_i(\mathcal{O} \,|\, p_T)$$

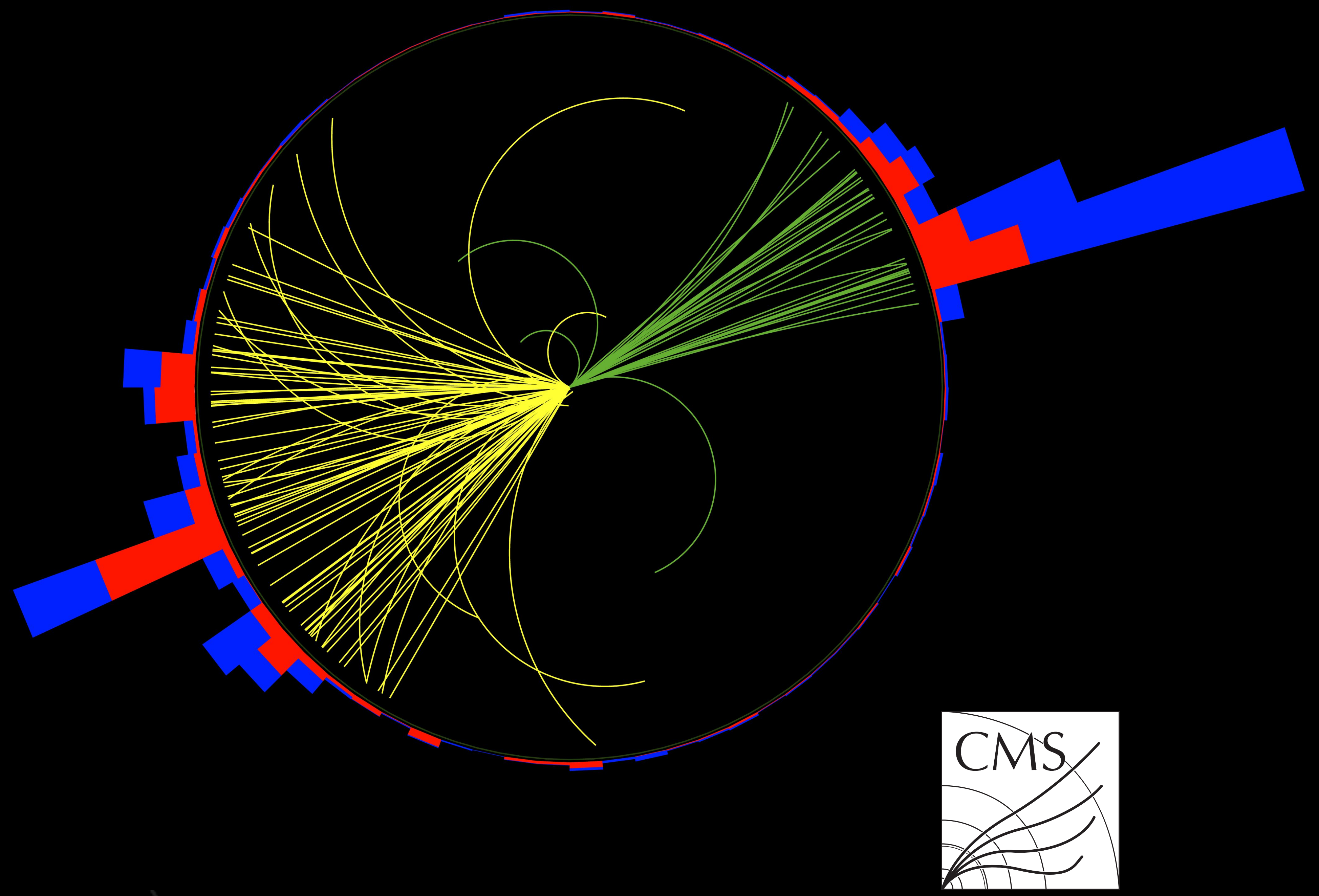
Marginalizing out y and taking a scale-/boost-invariant $\mathcal{O}$, this gives

$$p(p_T, \mathcal{O}) \approx \sum_i f_i\, p_i(p_T)\, p_i(\mathcal{O})$$

Ex $\mathcal{O}$: $N-$subjettiness ratios, D_2,

[1108.2701] [1401.4458]

Canonical example: ***jets***



What is factorization?

Factorization: physics at different energy scales approximately independent

Given $p_T, y, \mathcal{O}$, this gives

$$p(p_T, y, \mathcal{O}) \approx \sum_i f_i \, p_i(p_T, y) \, p_i(\mathcal{O} \,|\, p_T)$$

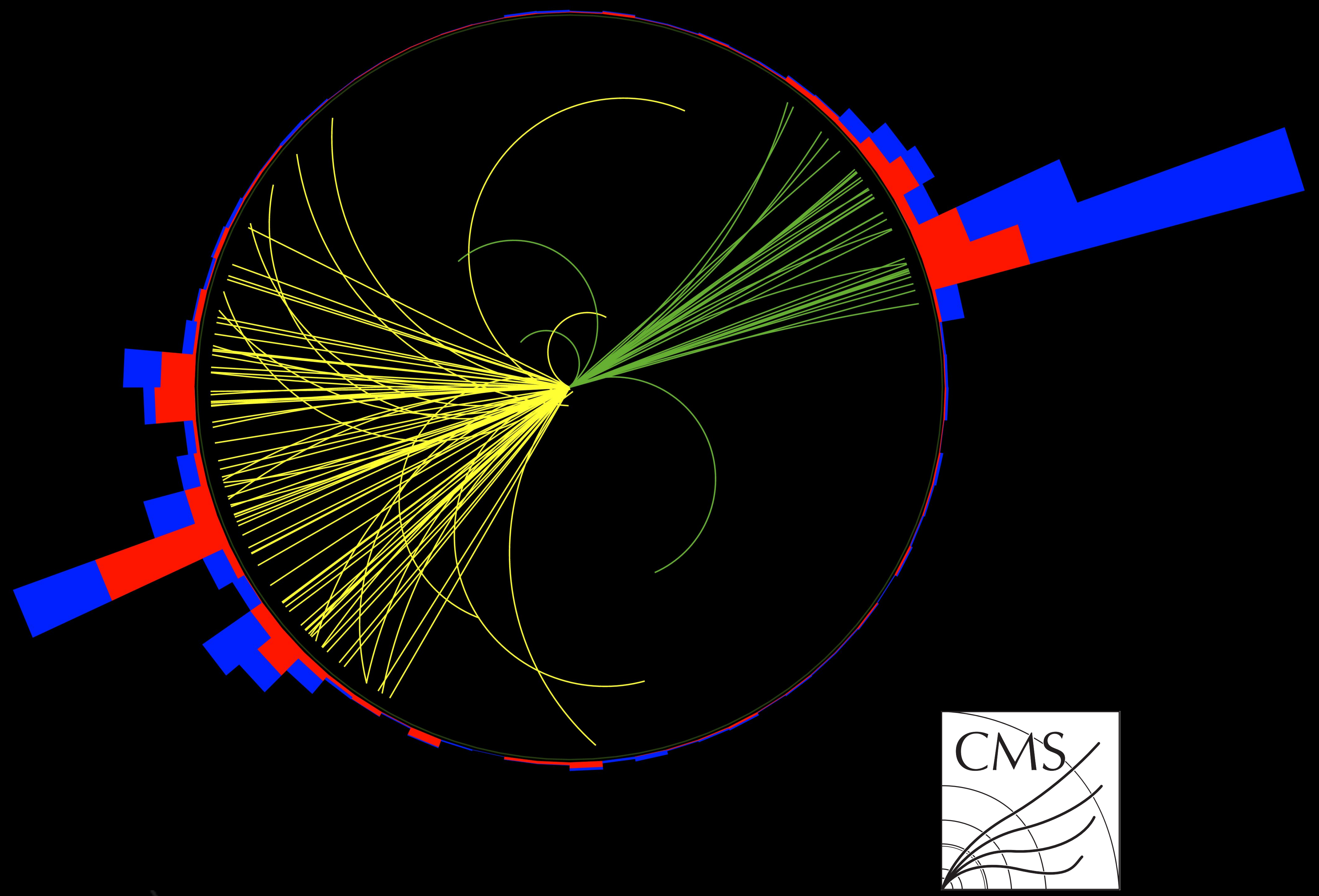
Marginalizing out y and taking a scale-/boost-invariant $\mathcal{O}$, this gives

$$p(p_T, \mathcal{O}) \approx \sum_i f_i \, p_i(p_T) \, p_i(\mathcal{O})$$

Ex $\mathcal{O}$: $N-$ subjettiness ratios, D_2, D_3,

[1108.2701] [1401.4458] [1507.03018]

Canonical example: ***jets***



What is factorization?

Factorization: physics at different energy scales approximately independent

Given $p_T, y, \mathcal{O}$, this gives

$$p(p_T, y, \mathcal{O}) \approx \sum_i f_i \, p_i(p_T, y) \, p_i(\mathcal{O} \,|\, p_T)$$

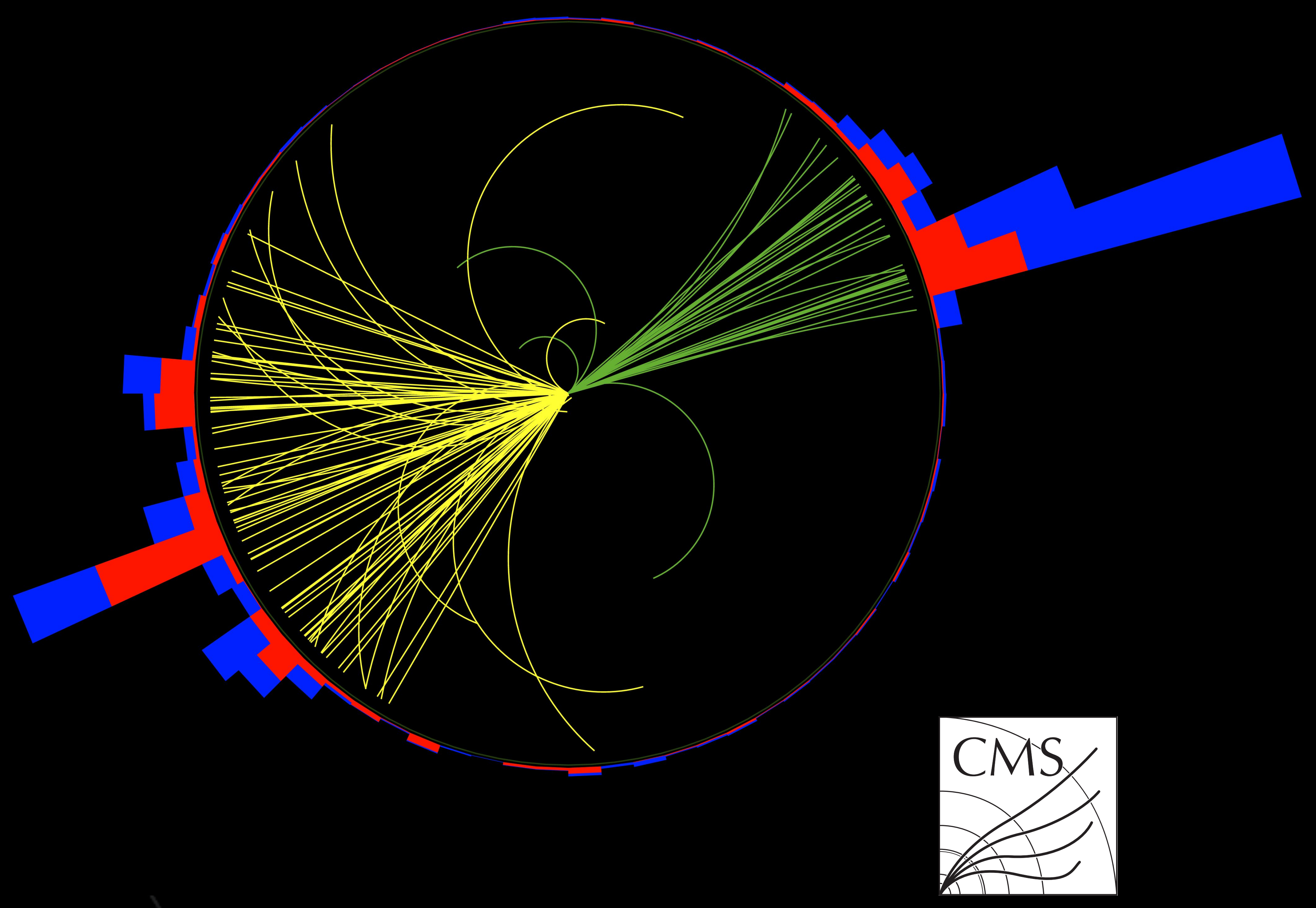
Marginalizing out y and taking a scale-/boost-invariant $\mathcal{O}$, this gives

$$p(p_T, \mathcal{O}) \approx \sum_i f_i \, p_i(p_T) \, p_i(\mathcal{O})$$

Ex $\mathcal{O}$: N – subjettiness ratios, D_2, D_3, N_i

[1108.2701] [1401.4458] [1507.03018] [1609.07483]

Canonical example: ***jets***



Signal versus Background Factorization

Signal versus Background Factorization

Assume we have a distribution of the form

Signal versus Background Factorization

Assume we have a distribution of the form

$$p(p_T, \mathcal{O}) = f_S \; p_S(p_T)\, p_S(\mathcal{O}) + f_B \, p_B(p_T)\, p_B(\mathcal{O})$$

Signal versus Background Factorization

Assume we have a distribution of the form

$$p(p_T, \mathcal{O}) = f_S \; p_S(p_T)\, p_S(\mathcal{O}) + f_B\, p_B(p_T)\, p_B(\mathcal{O})$$

Calculate the conditional expectation

Signal versus Background Factorization

Assume we have a distribution of the form

$$p(p_T, \mathcal{O}) = f_S \; p_S(p_T)\, p_S(\mathcal{O}) + f_B \, p_B(p_T)\, p_B(\mathcal{O})$$

Calculate the conditional expectation

$$\mathbb{E}[p_T \,|\, \mathcal{O}] = \langle p_T \rangle_B + f_S \frac{(\langle p_T \rangle_S - \langle p_T \rangle_B) L_{S/B}(\mathcal{O})}{1 - f_S + f_S \, L_{S/B}(\mathcal{O})}$$

Signal versus Background Factorization

Assume we have a distribution of the form

$$p(p_T, \mathcal{O}) = f_S\, p_S(p_T)\, p_S(\mathcal{O}) + f_B\, p_B(p_T)\, p_B(\mathcal{O})$$

Calculate the conditional expectation

$$\mathbb{E}[p_T \,|\, \mathcal{O}] = \langle p_T \rangle_B + f_S \frac{(\langle p_T \rangle_S - \langle p_T \rangle_B) L_{S/B}(\mathcal{O})}{1 - f_S + f_S\, L_{S/B}(\mathcal{O})} = \mathcal{M}\left(L_{S/B}(\mathcal{O}) \right)$$

Monotonically related to $L_{S/B}(\mathcal{O})$, the optimal classifier!

Signal versus Background Factorization

Assume we have a distribution of the form

$$p(p_T, \mathcal{O}) = f_S \; p_S(p_T)\, p_S(\mathcal{O}) + f_B\, p_B(p_T)\, p_B(\mathcal{O})$$

Calculate the conditional expectation

$$\mathbb{E}[p_T \,|\, \mathcal{O}] = \langle p_T \rangle_B + f_S \frac{(\langle p_T \rangle_S - \langle p_T \rangle_B) L_{S/B}(\mathcal{O})}{1 - f_S + f_S\, L_{S/B}(\mathcal{O})} = \mathcal{M}(\, L_{S/B}(\mathcal{O})\,)$$

Monotonically related to $L_{S/B}(\mathcal{O})$, the optimal classifier!

For $f_S = 0$, we get a random classifier

How do we extract $\mathbb{E}[p_T \,|\, \mathcal{O}]$?

How do we extract $\mathbb{E}[p_T \,|\, \mathcal{O}]$?

$\mathbb{E}[p_T \,|\, \mathcal{O}]$ minimizes the mean-squared error loss!

How do we extract $\mathbb{E}[p_T | \mathcal{O}]$?

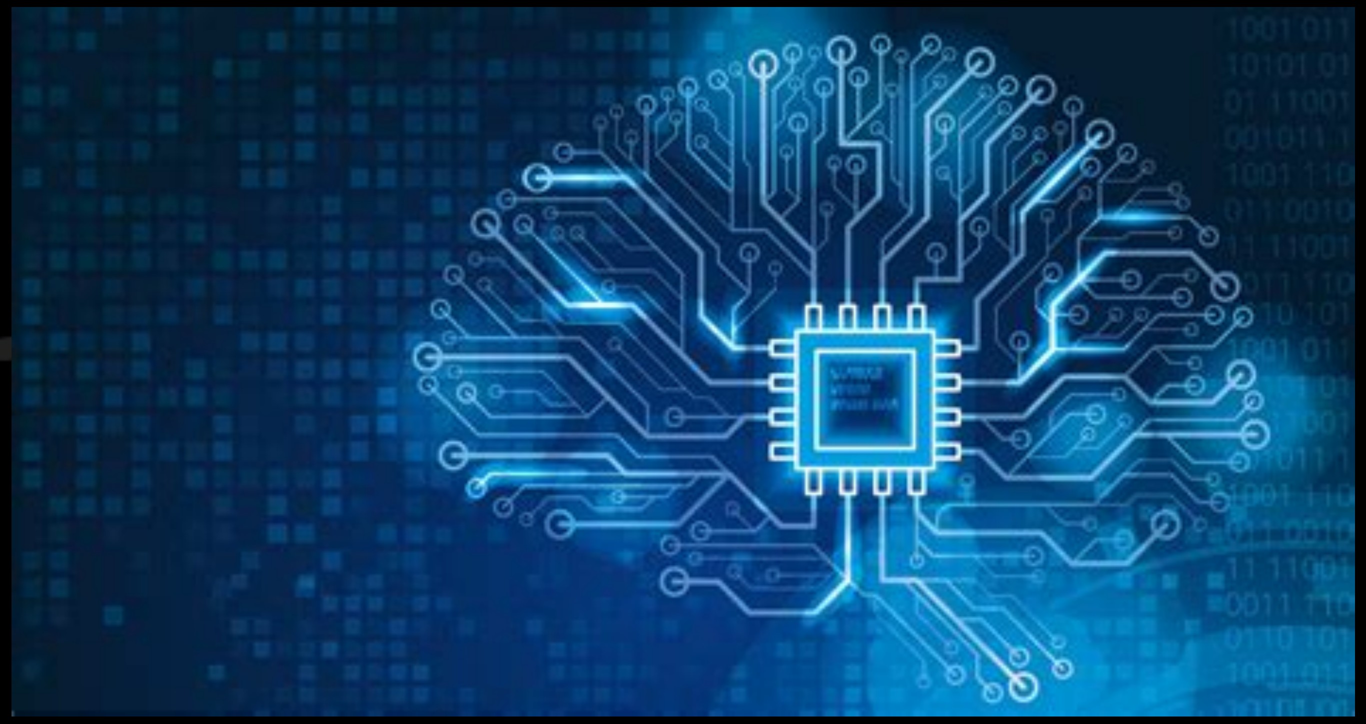
$\mathbb{E}[p_T | \mathcal{O}]$ minimizes the mean-squared error loss!

Enter machine learning!

How do we extract $\mathbb{E}[p_T | \mathcal{O}]$?

$\mathbb{E}[p_T | \mathcal{O}]$ minimizes the mean-squared error loss!

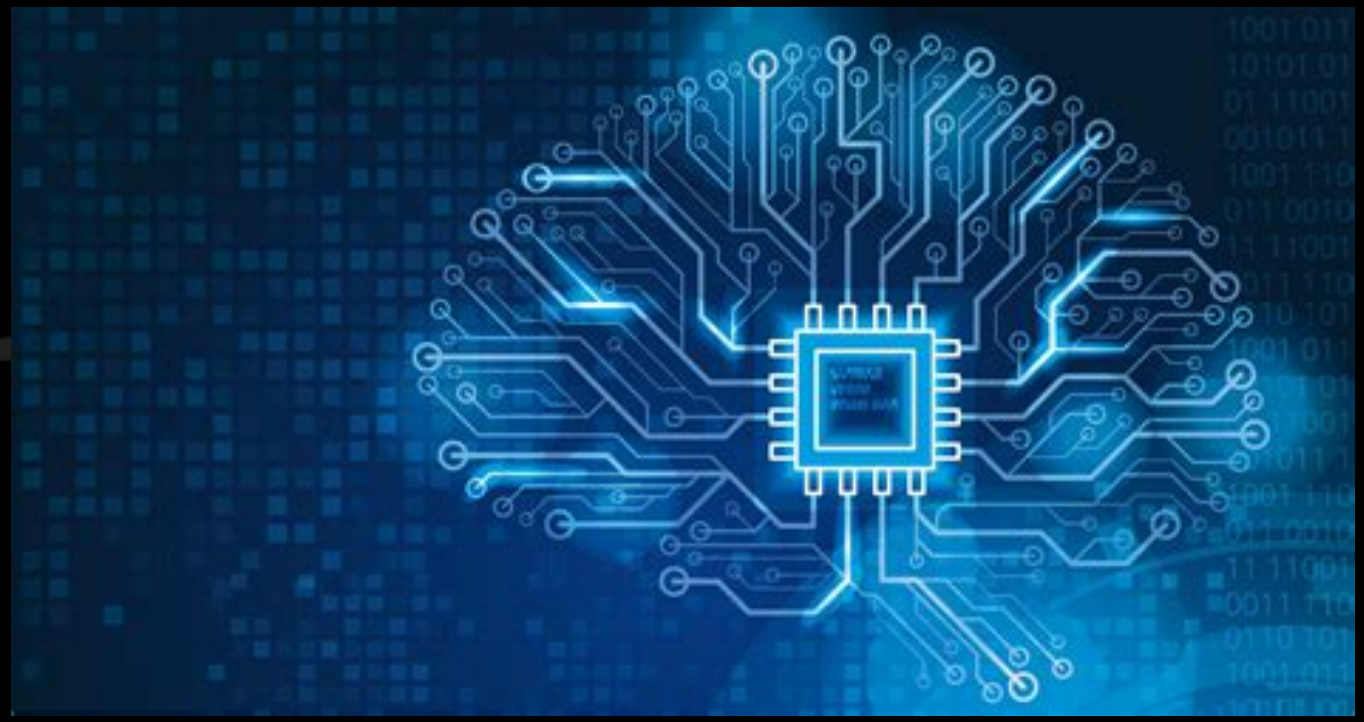
Enter machine learning!



How do we extract $\mathbb{E}[p_T | \mathcal{O}]$?

$\mathbb{E}[p_T | \mathcal{O}]$ minimizes the mean-squared error loss!

Enter machine learning!

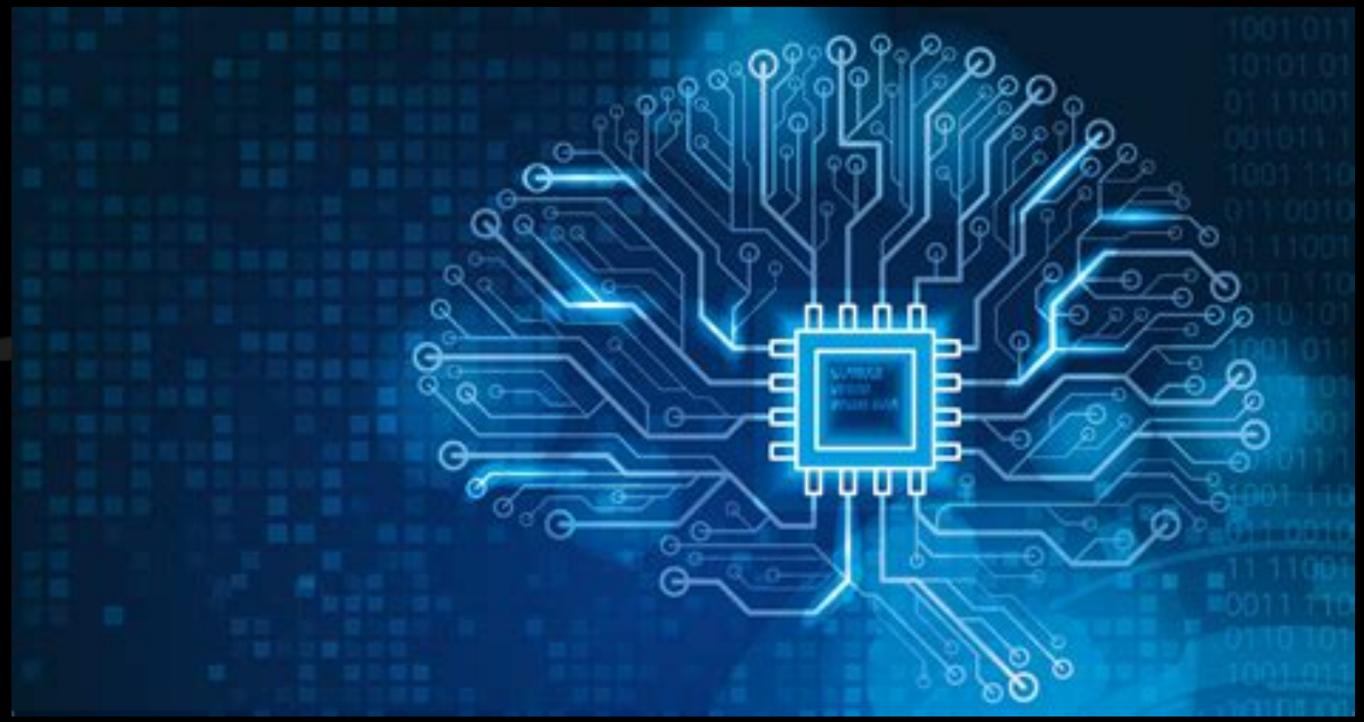


… really just need →

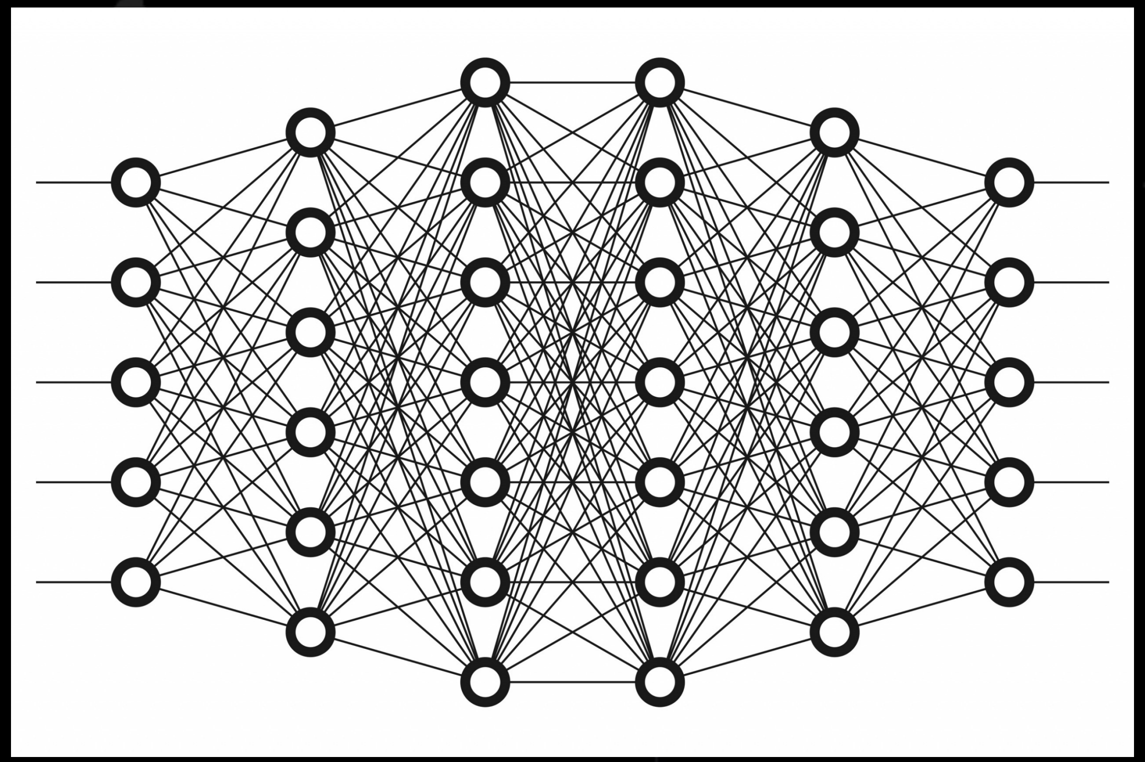
How do we extract $\mathbb{E}[p_T \,|\, \mathcal{O}]$?

$\mathbb{E}[p_T \,|\, \mathcal{O}]$ minimizes the mean-squared error loss!

Enter machine learning!



… really just need →



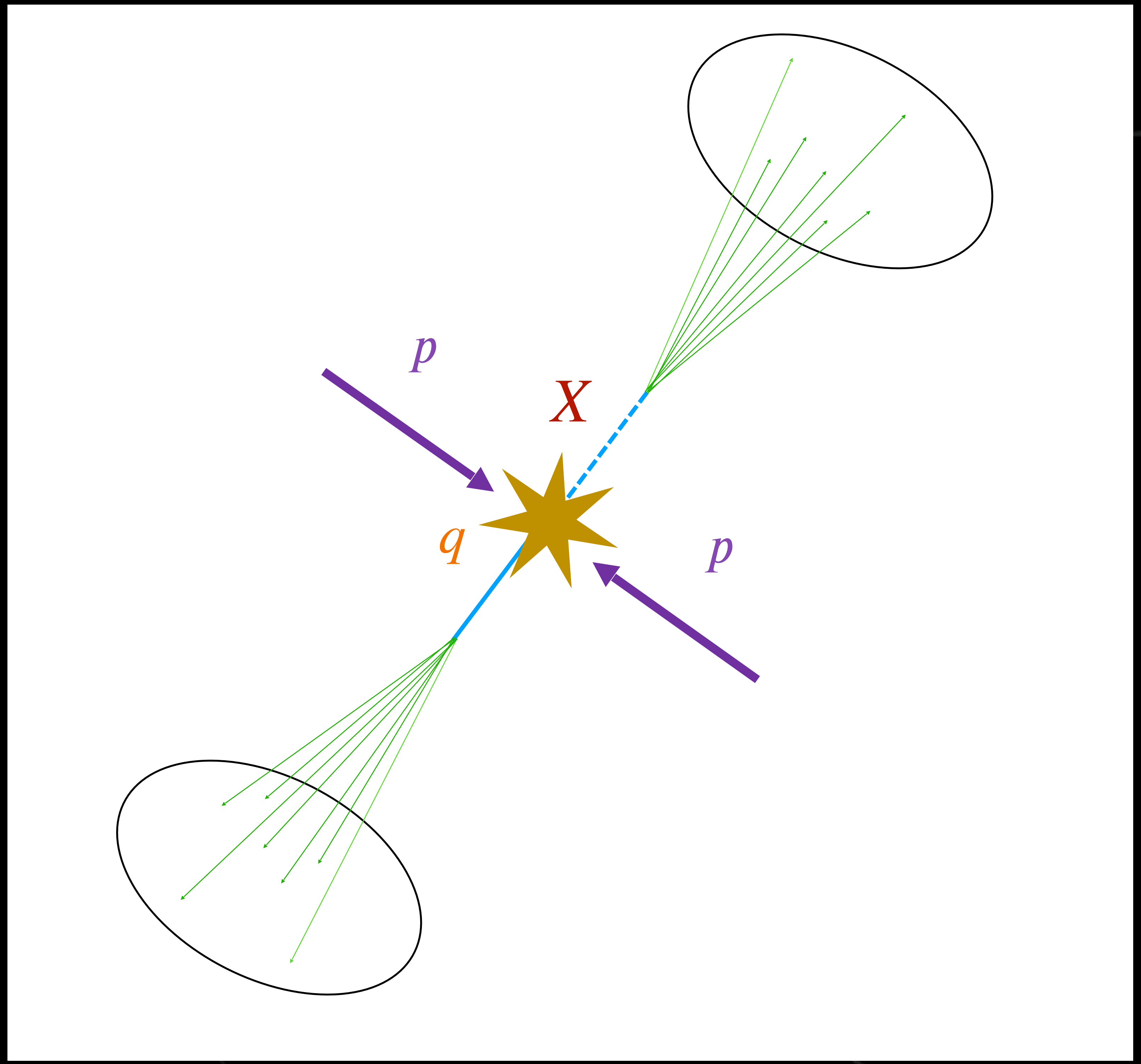
The FORCE Method

Factorized Observables for Regressing Conditional Expectation

The FORCE Method

Factorized Observables for Regressing Conditional Expectation

Define approximately factorized objects (e.g.~jets)

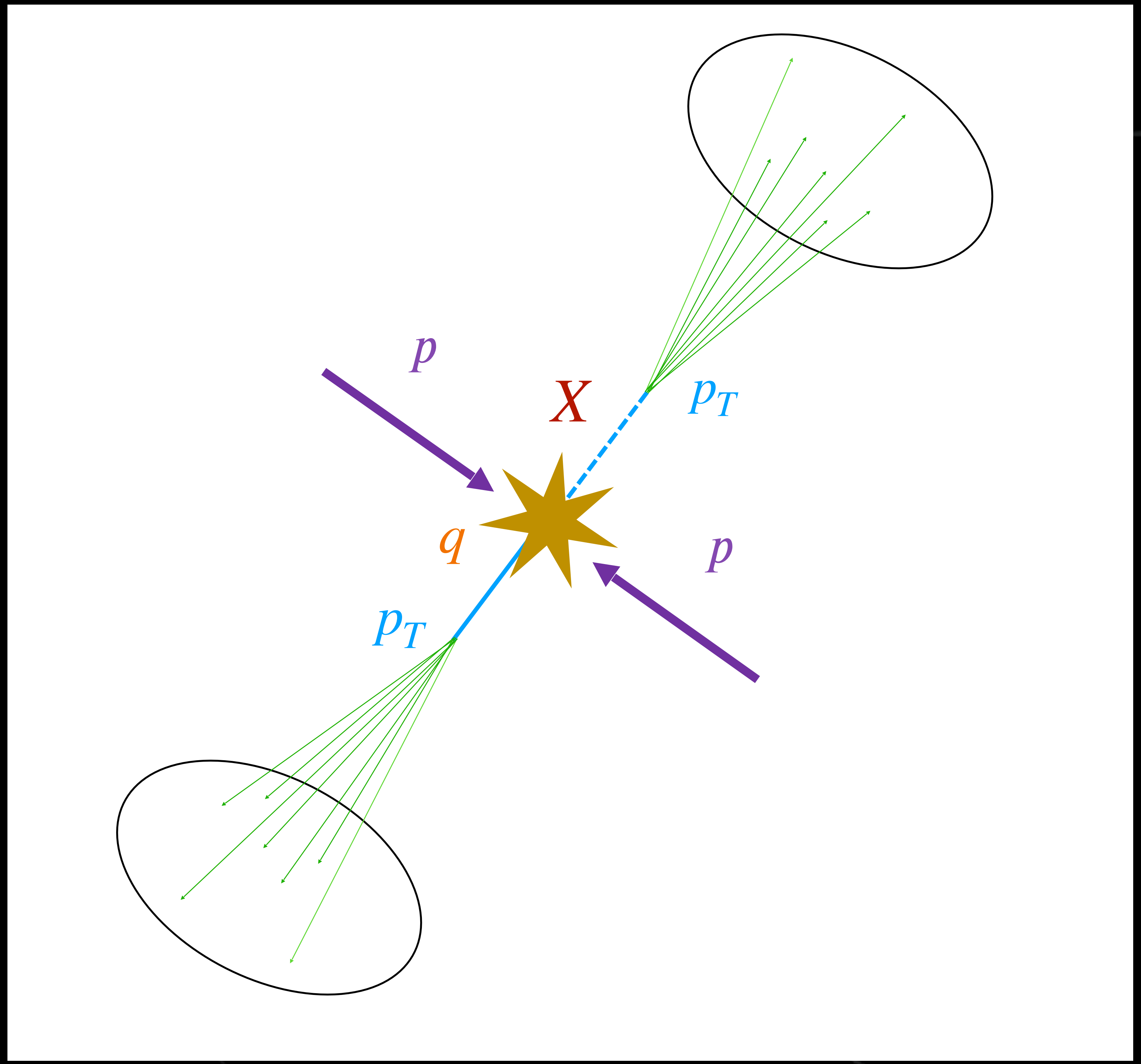


The FORCE Method

Factorized **O**bservables for **R**egressing **C**onditional **E**xpectation

Define approximately factorized objects (e.g.~jets)

- kinematics p_T

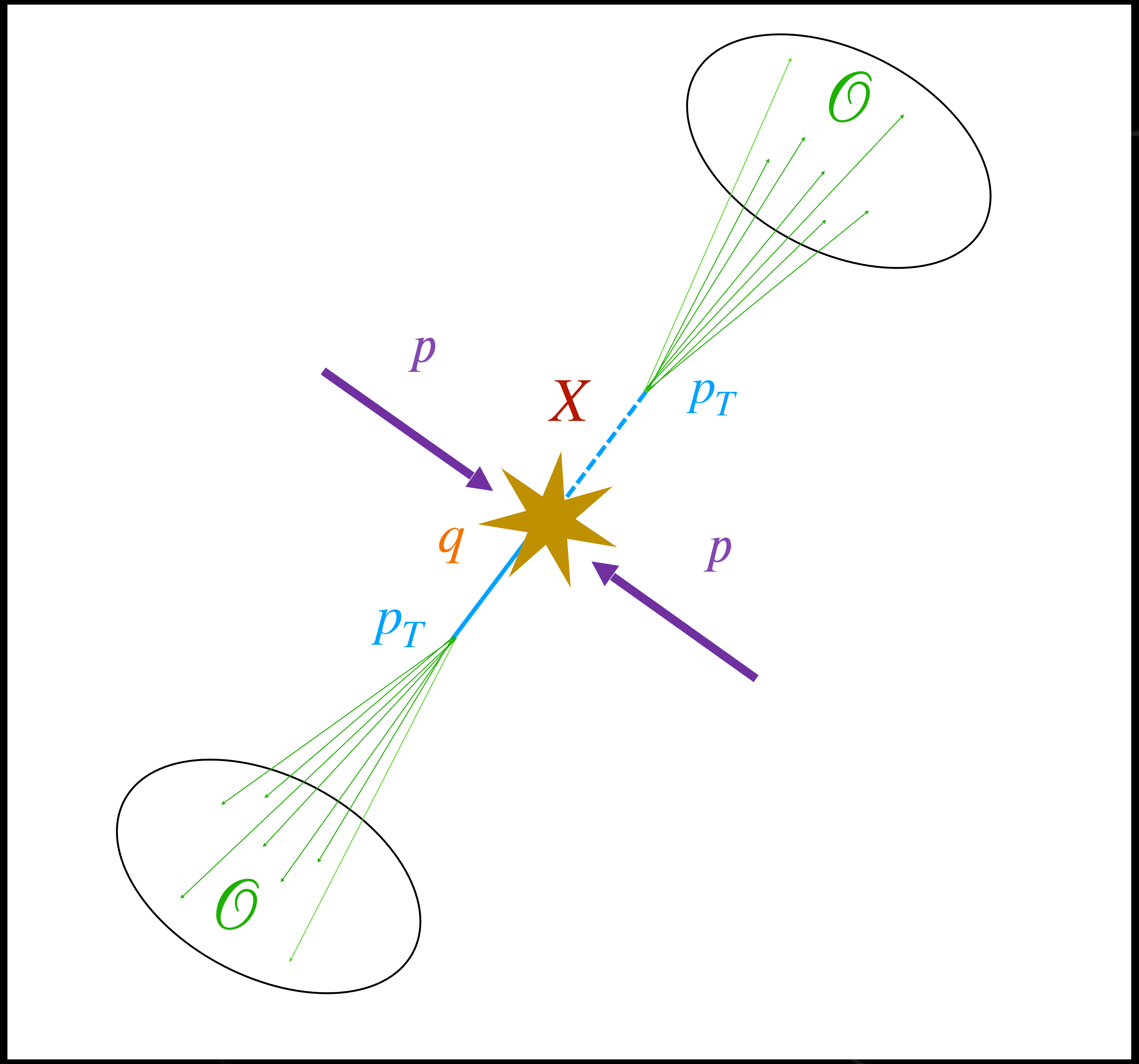


The FORCE Method

Factorized **O**bservables for **R**egressing **C**onditional **E**xpectation

Define approximately factorized objects (e.g.~jets)

- kinematics p_T
- scale-/boost-invariant substructure $\mathcal{O}$



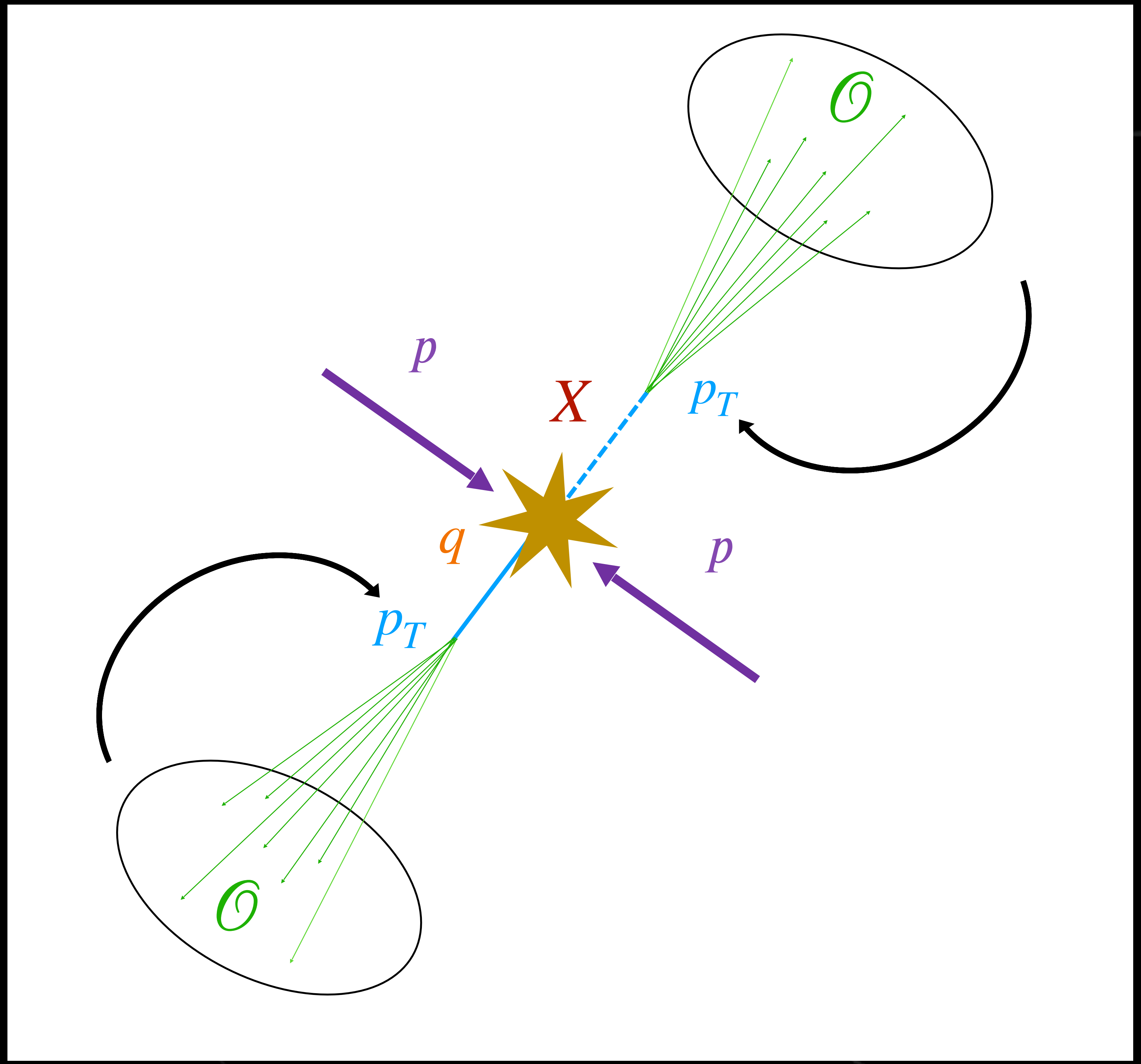
The FORCE Method

Factorized **O**bservables for **R**egressing **C**onditional **E**xpectation

Define approximately factorized objects (e.g.~jets)

- kinematics p_T
- scale-/boost-invariant substructure $\mathcal{O}$

Train a machine-learning model to predict p_T from $\mathcal{O}$



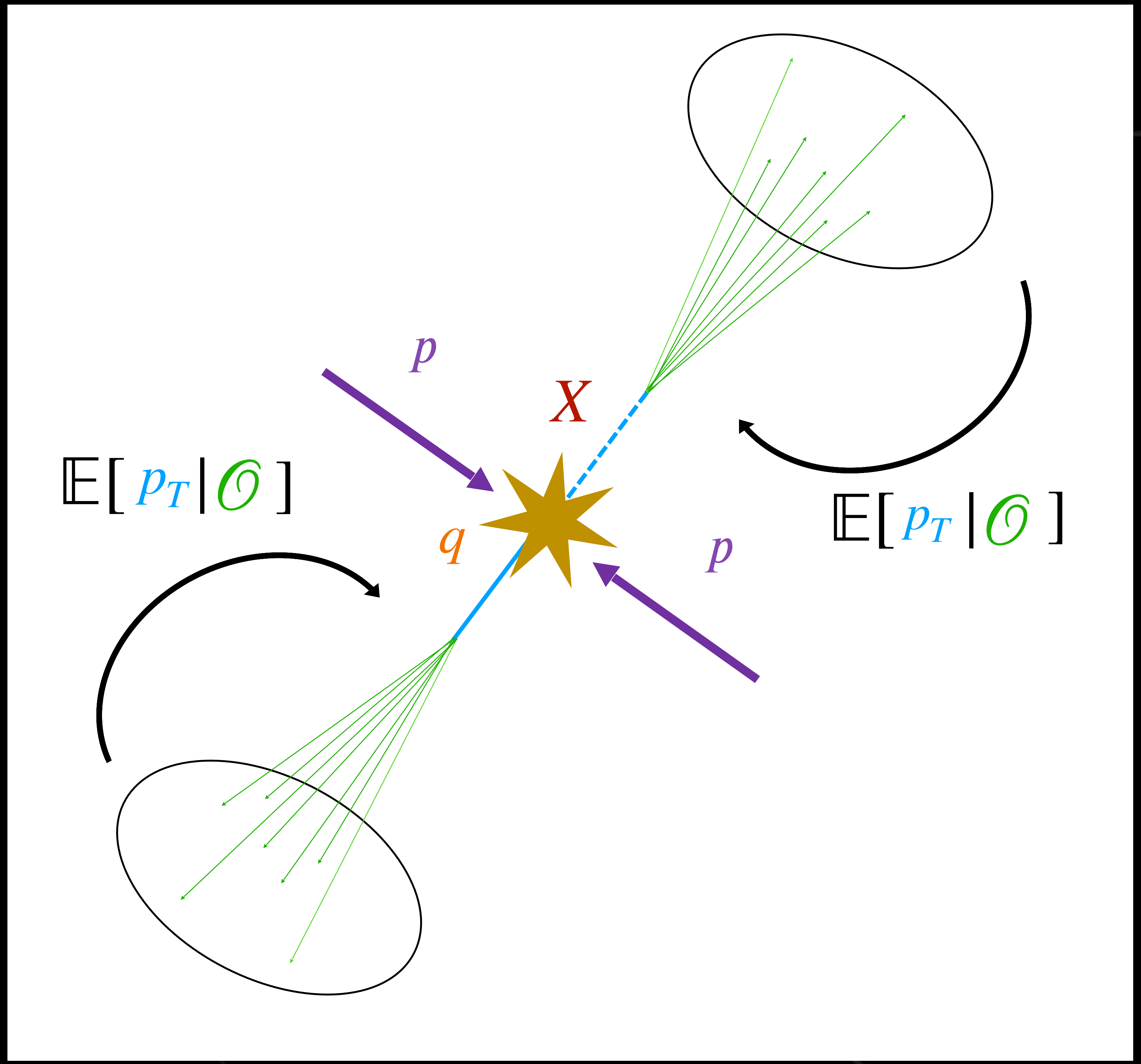
The FORCE Method

Factorized **O**bservables for **R**egressing **C**onditional **E**xpectation

Define approximately factorized objects (e.g.~jets)

- kinematics p_T
- scale-/boost-invariant substructure $\mathcal{O}$

Train a machine-learning model to predict p_T from $\mathcal{O}$ with the mean-squared error loss



The FORCE Method

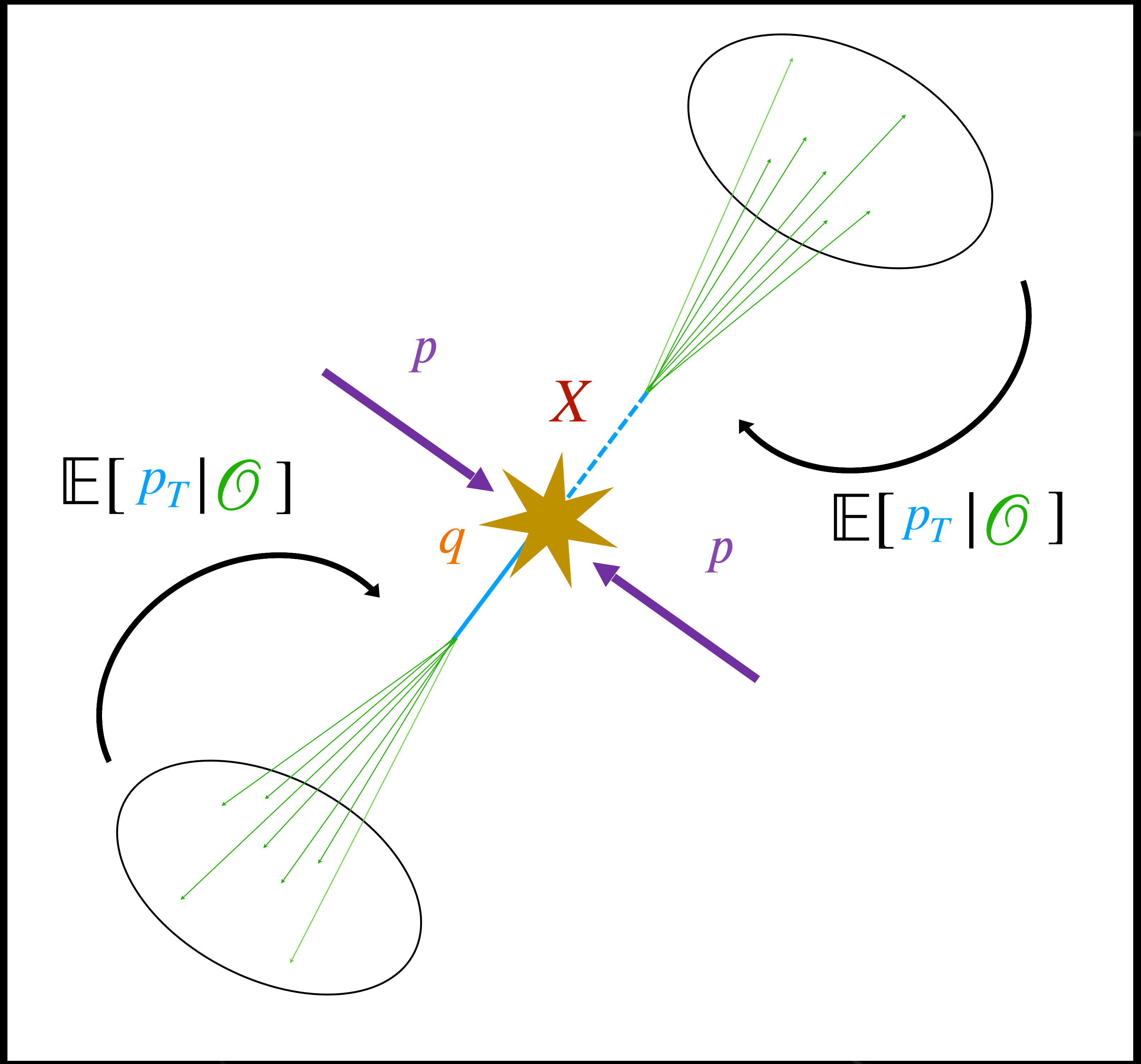
Factorized **O**bservables for **R**egressing **C**onditional **E**xpectation

Define approximately factorized objects (e.g.~jets)

- kinematics p_T
- scale-/boost-invariant substructure $\mathcal{O}$

Train a machine-learning model to predict p_T from $\mathcal{O}$ with the mean-squared error loss

Classify anomalous objects via cutting on the model output



The FORCE Method

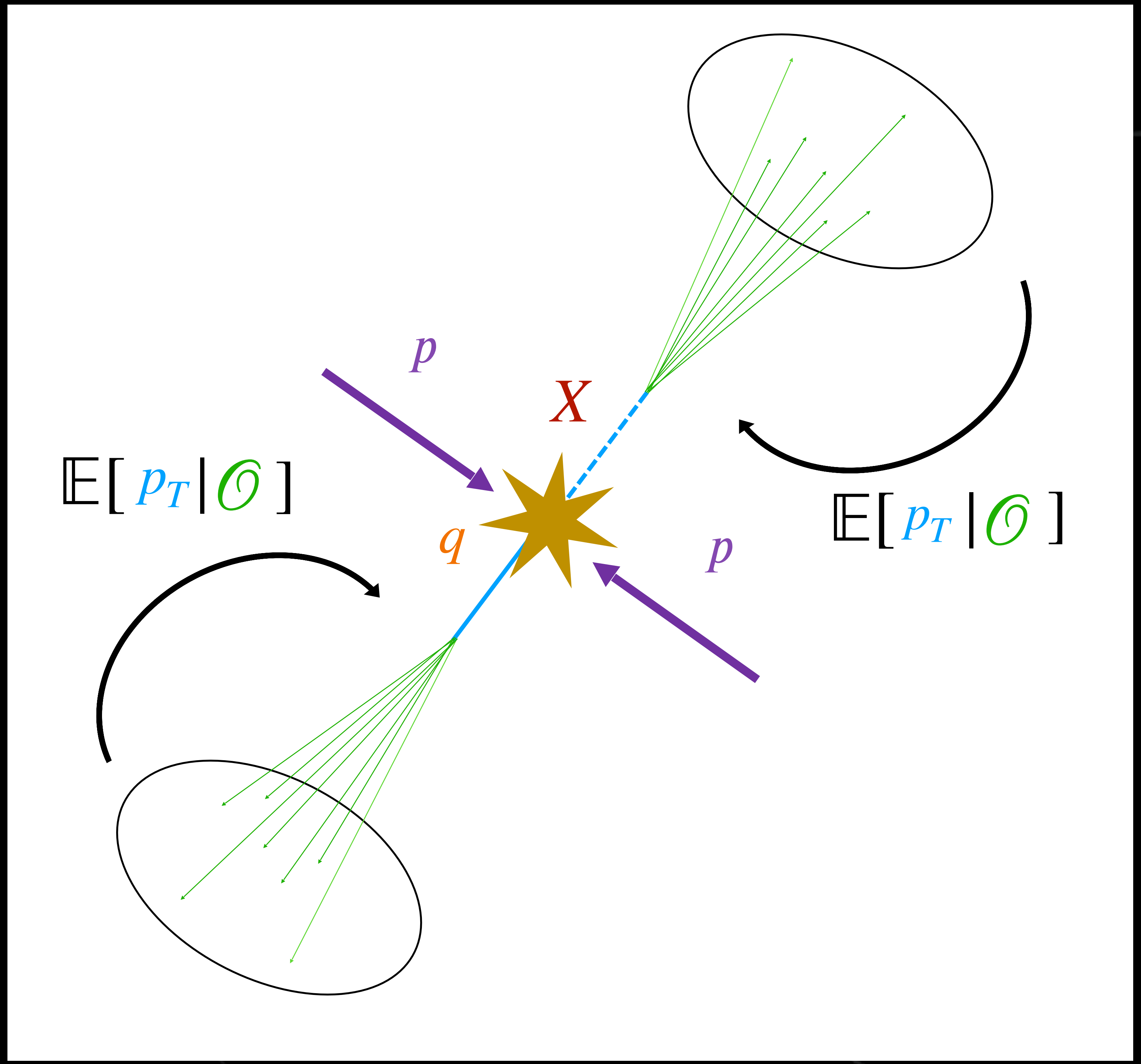
Factorized **O**bservables for **R**egressing **C**onditional **E**xpectation

Define approximately factorized objects (e.g.~jets)

- kinematics p_T
- scale-/boost-invariant substructure $\mathcal{O}$

Train a machine-learning model to predict p_T from $\mathcal{O}$ with the mean-squared error loss

Classify anomalous objects via cutting on the model output



Find anomalies by predicting kinematics from substructure

Use the FORCE Luke

Demonstrations of FORCE:

Use the FORCE Luke

Demonstrations of FORCE:

Use the FORCE Luke

Demonstrations of FORCE:

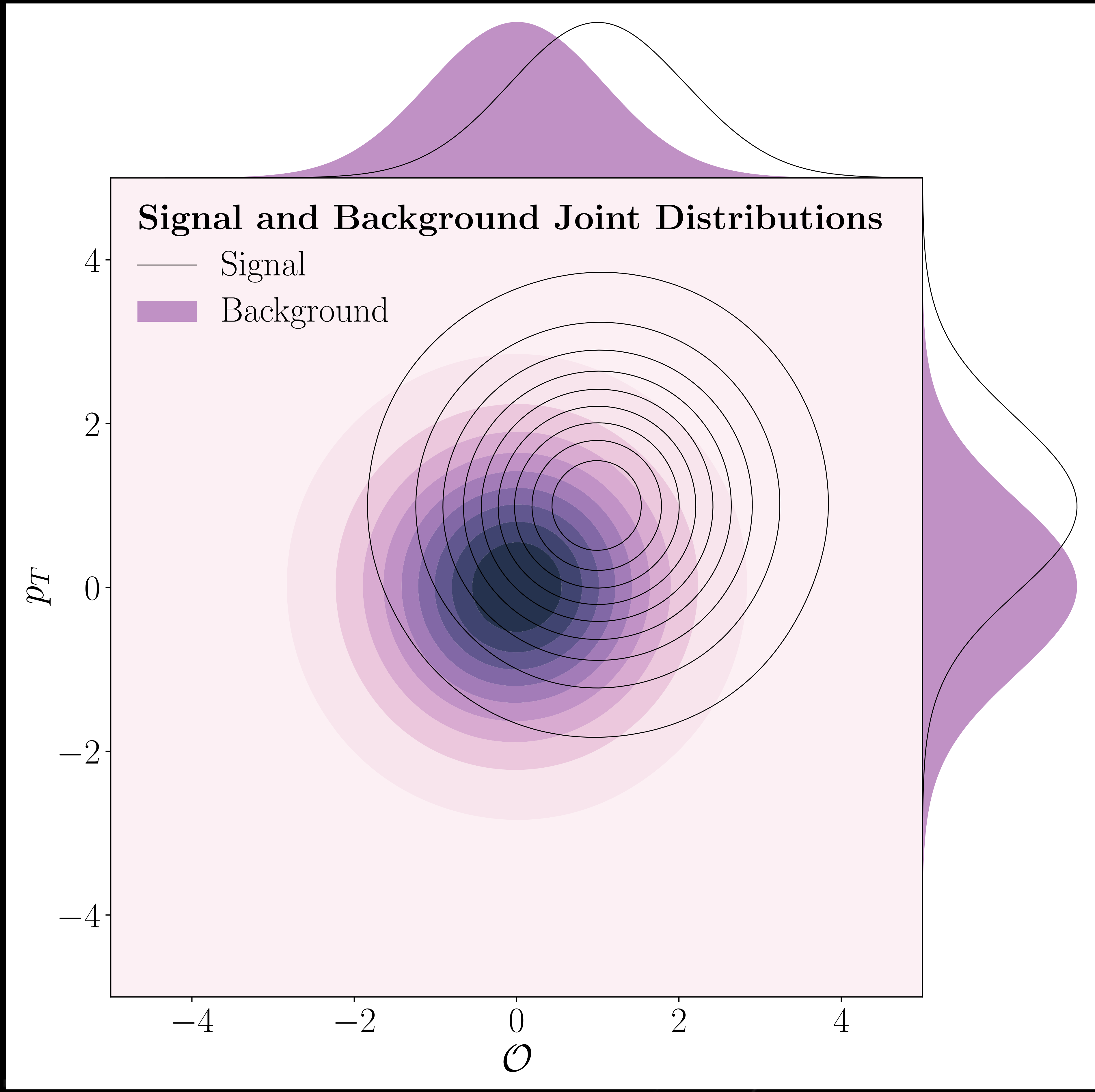
1. Toy Gaussian Dataset

Use the FORCE Luke

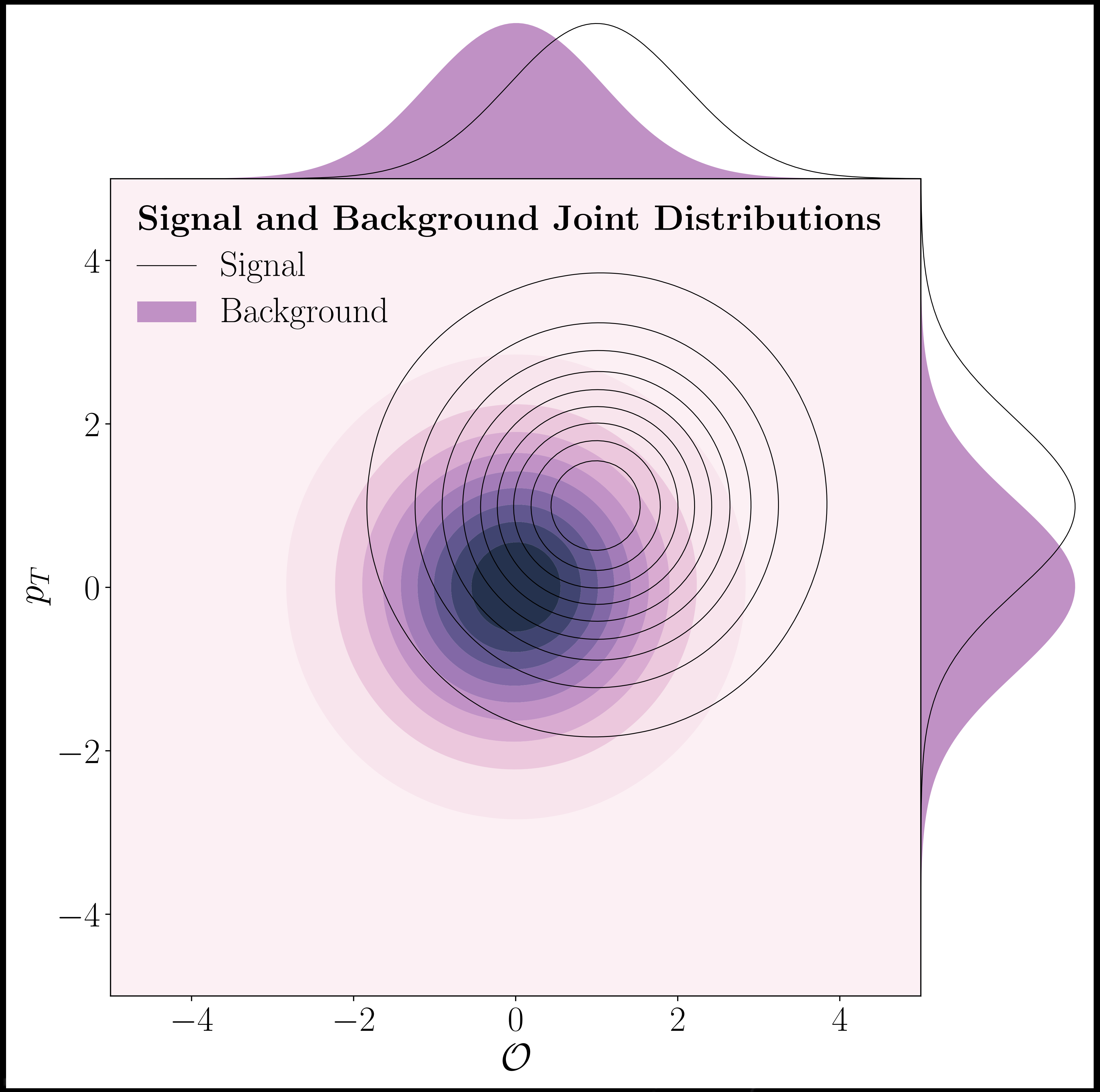
Demonstrations of FORCE:

1. Toy Gaussian Dataset
2. LHC Olympics R&D Dataset

Gaussian Simulation

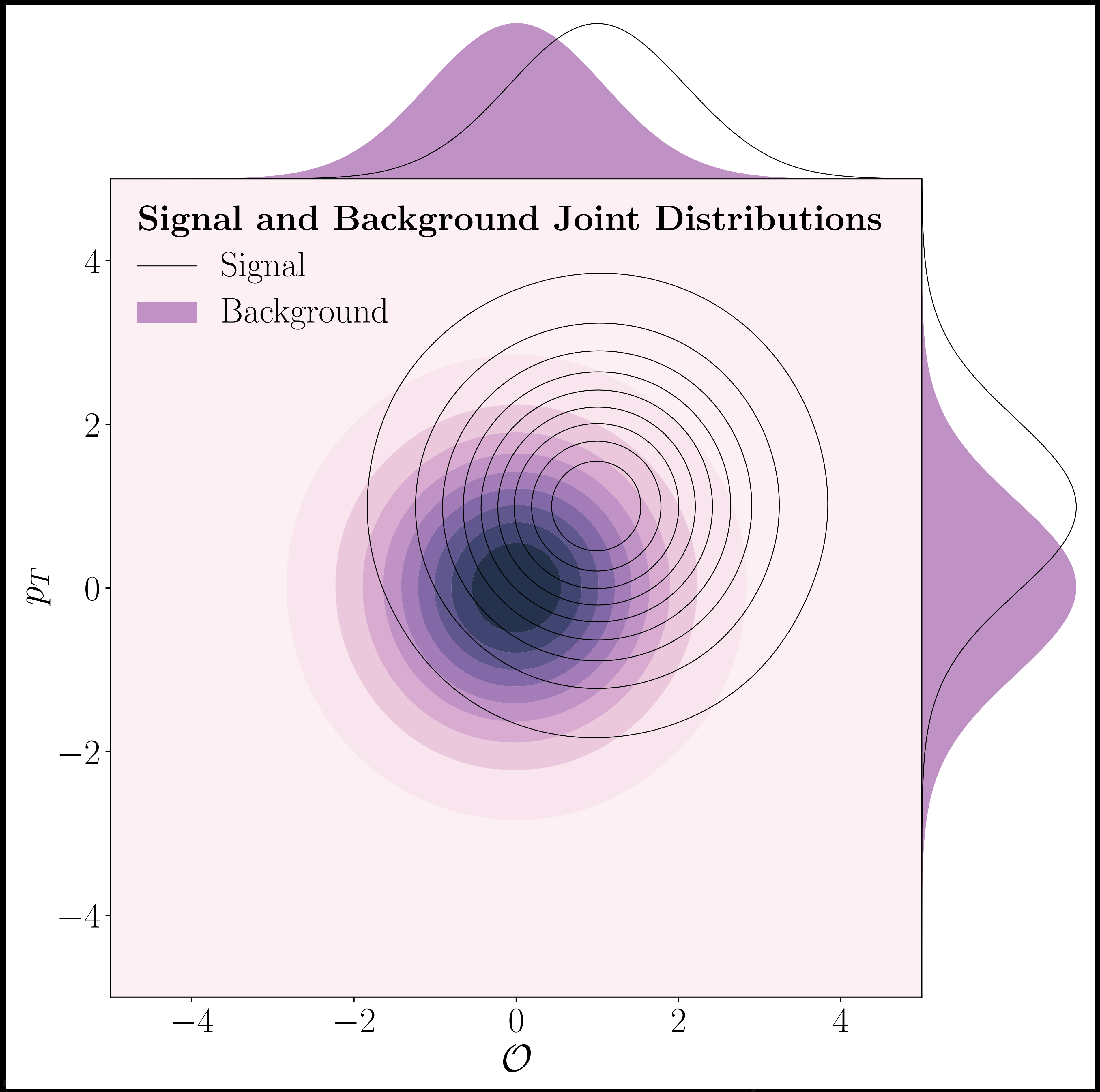


Gaussian Simulation



1 kinematic variable, 1 substructure variable

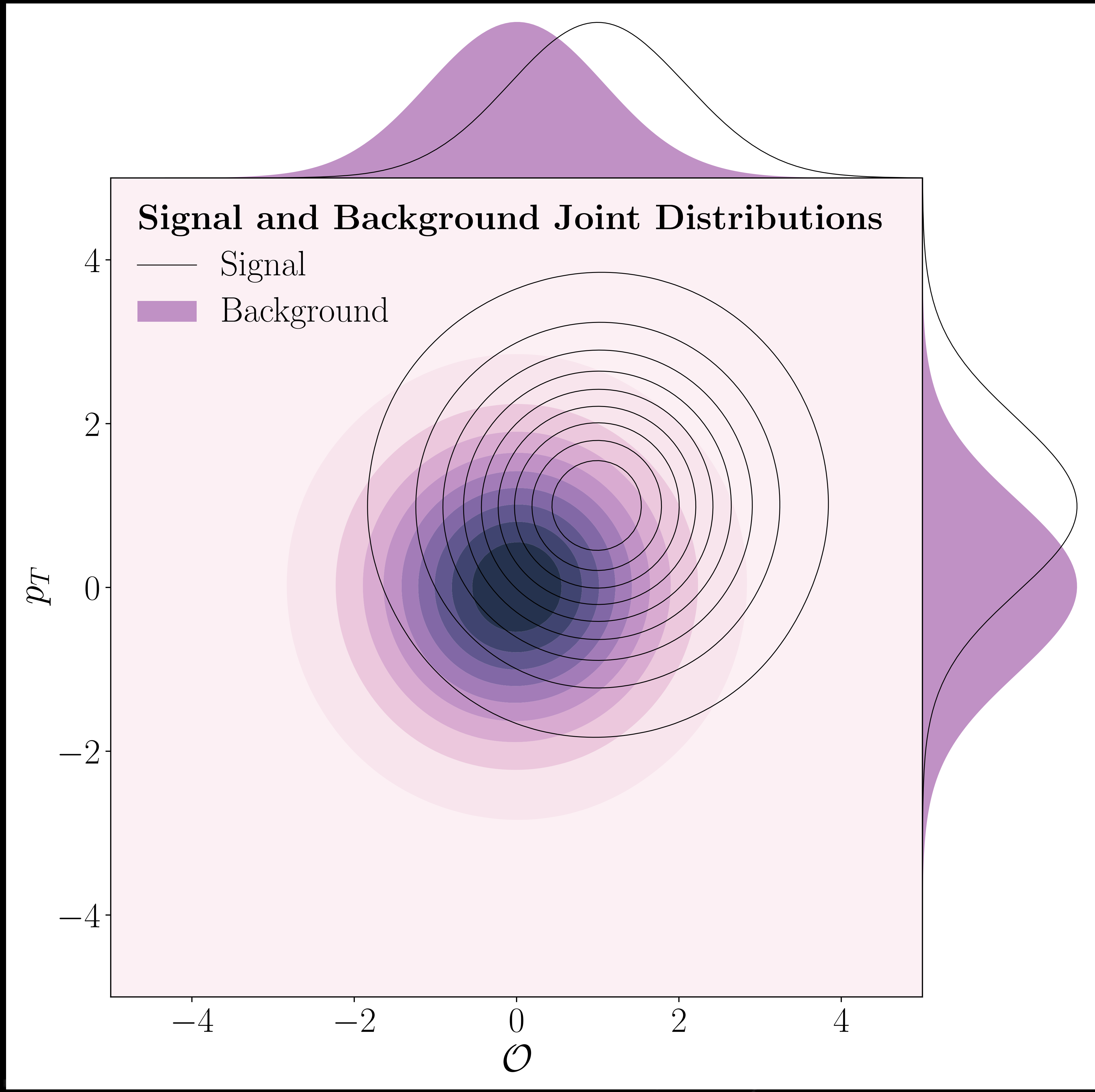
Gaussian Simulation



1 kinematic variable, 1 substructure variable

1 million draws from background model

Gaussian Simulation



1 kinematic variable, 1 substructure variable

1 million draws from background model

$f_S \cdot$ 1 million draws from signal model

ROC-AUC for Gaussian Simulation

ROC-AUC for Gaussian Simulation

- Fully connected network
- 3 layers of 100 nodes
- ReLU Activation

ROC-AUC for Gaussian Simulation

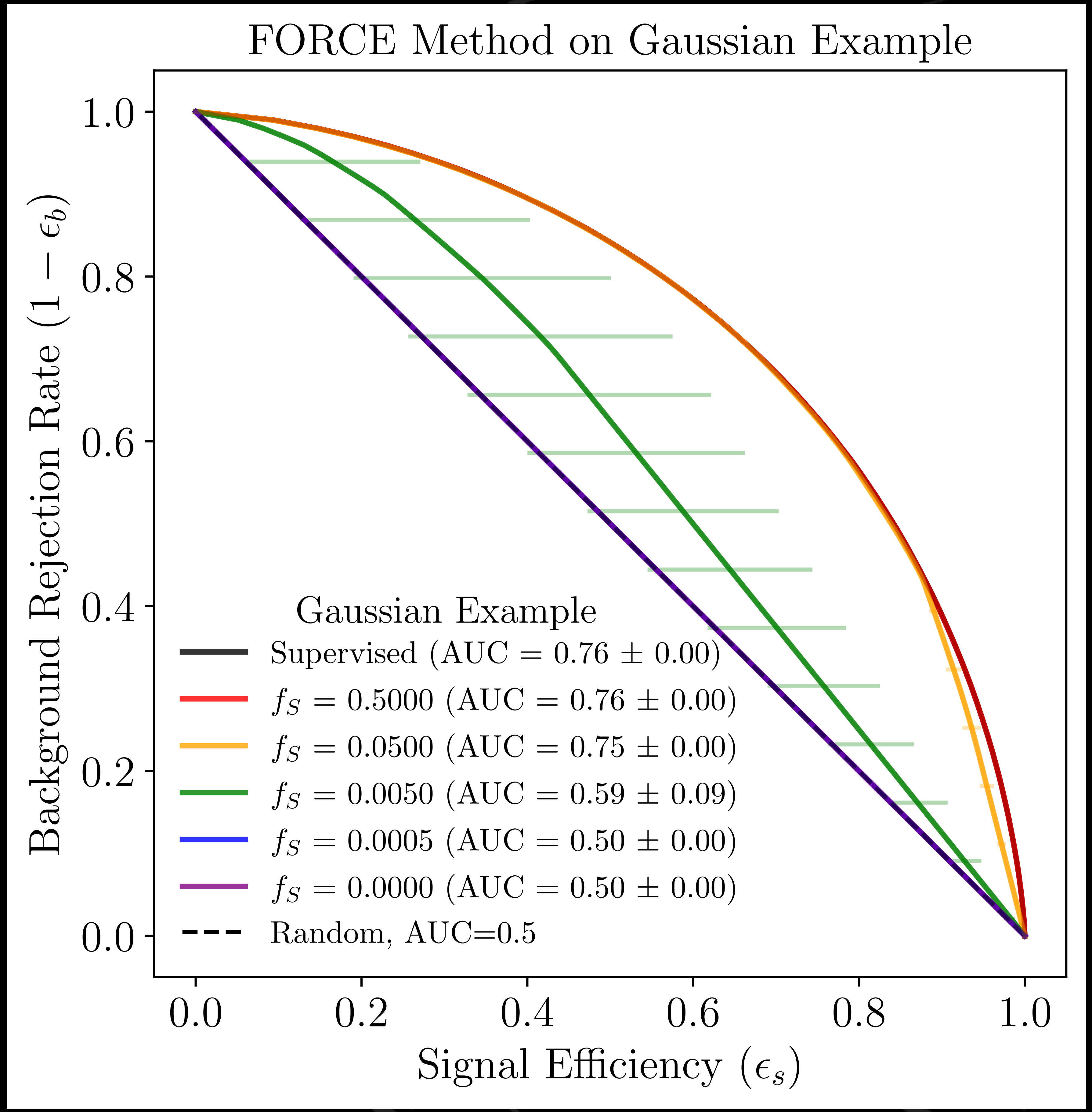
- Fully connected network
- 3 layers of 100 nodes
- ReLU Activation

$\Longrightarrow$

ROC-AUC for Gaussian Simulation

- Fully connected network
- 3 layers of 100 nodes
- ReLU Activation

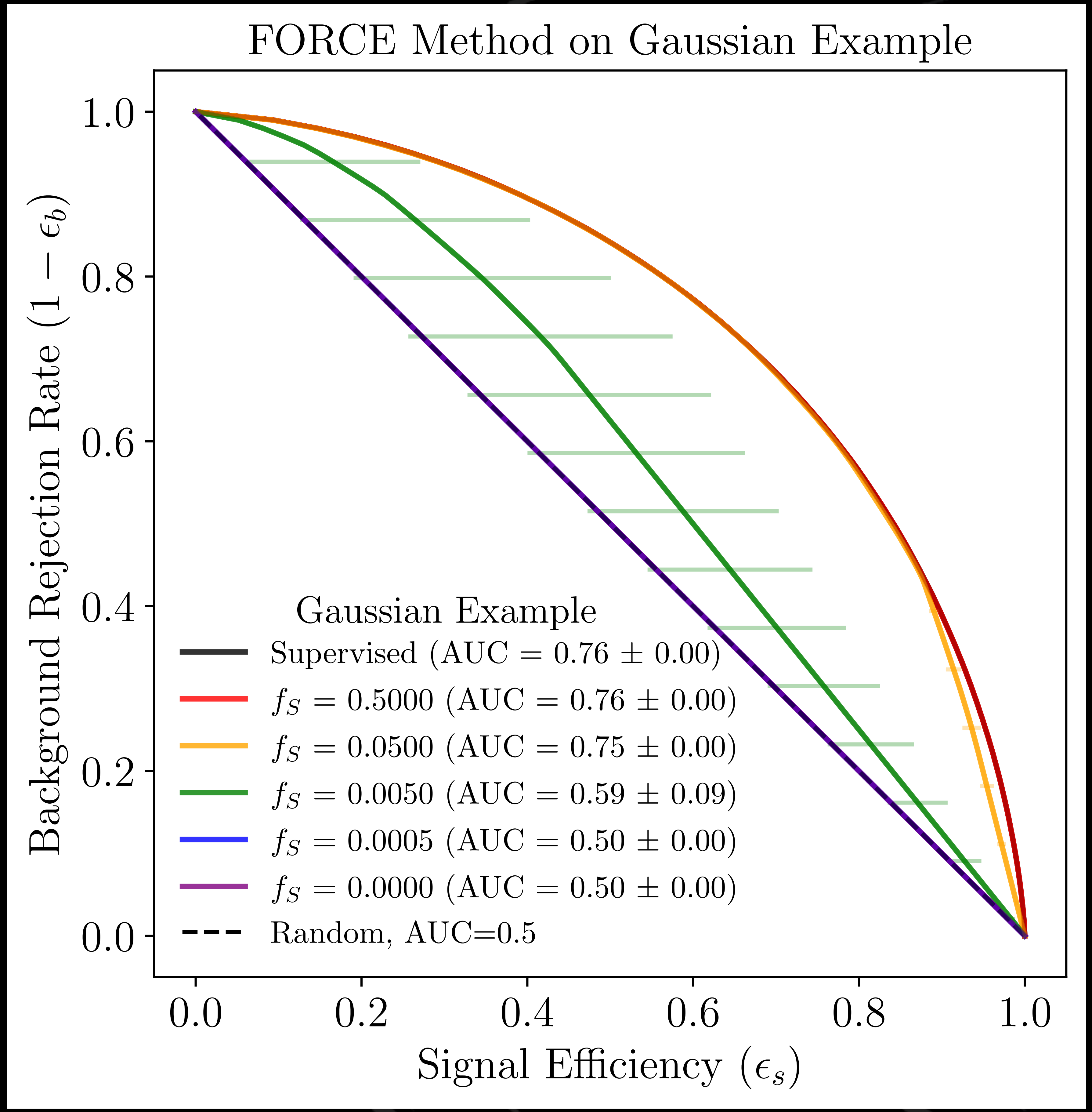
⟹



ROC-AUC for Gaussian Simulation

- Fully connected network
- 3 layers of 100 nodes
- ReLU Activation

⟹

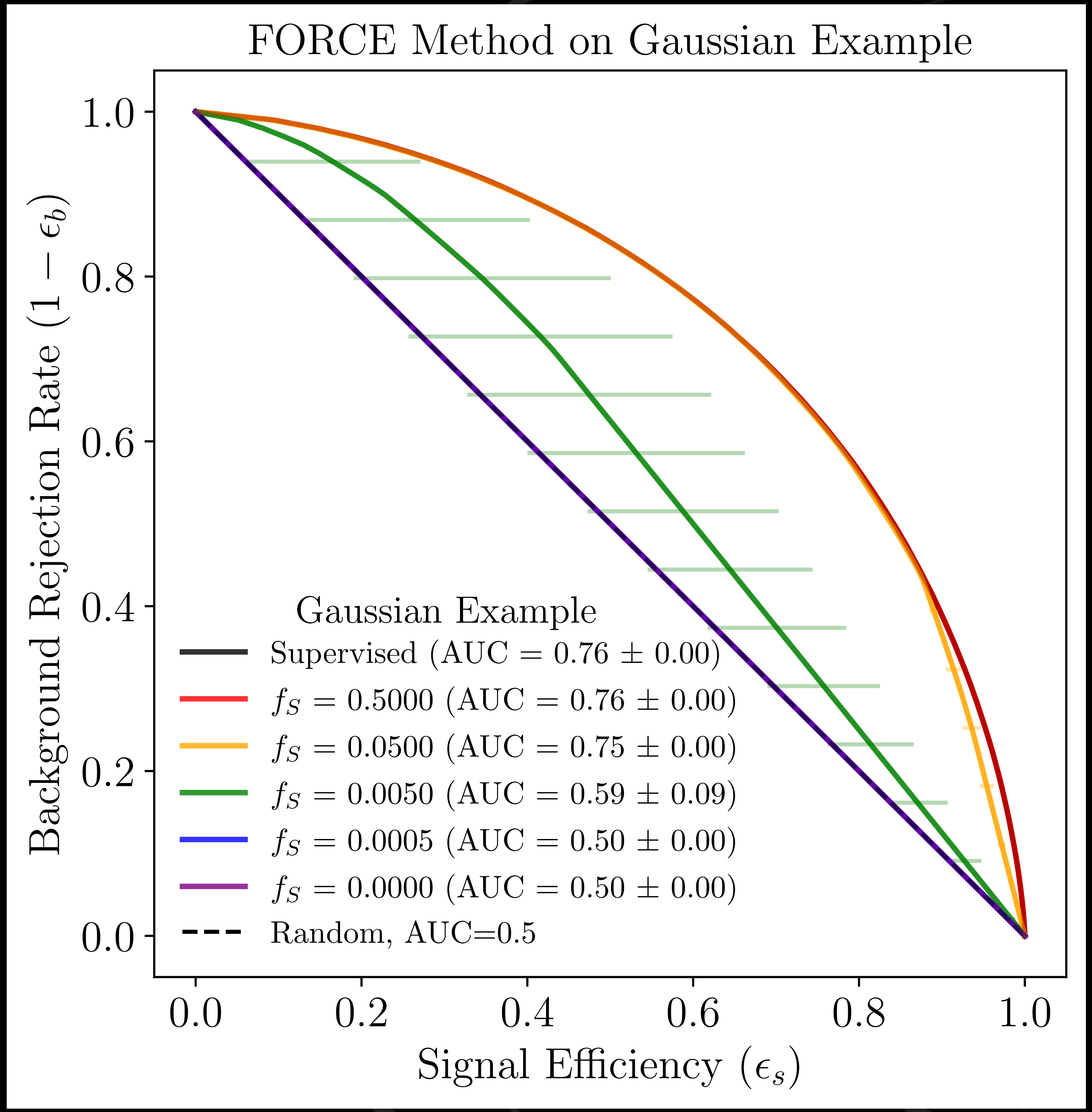


Optimal performance in high signal limit

ROC-AUC for Gaussian Simulation

- Fully connected network
- 3 layers of 100 nodes
- ReLU Activation

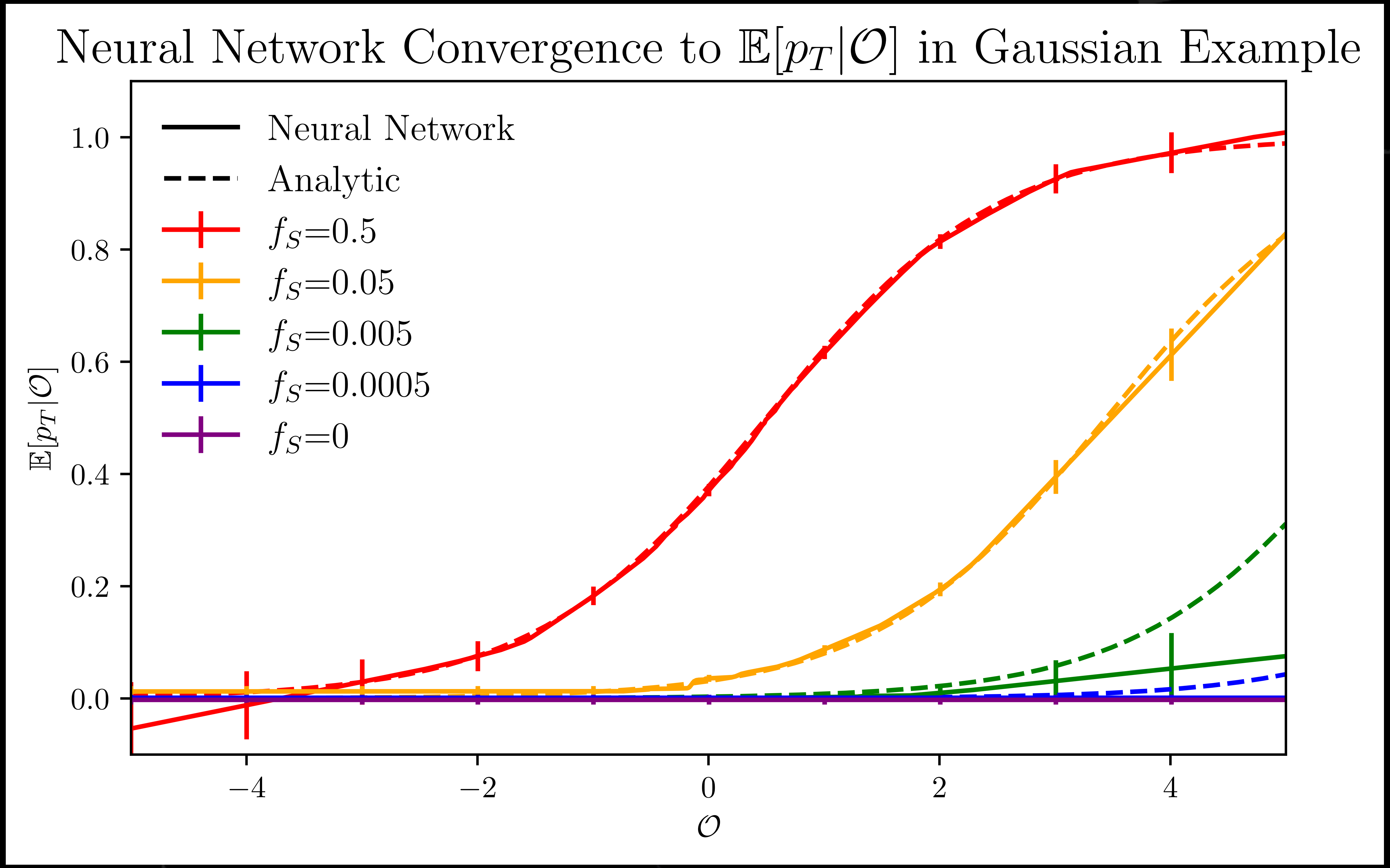
$\Longrightarrow$



Optimal performance in high signal limit

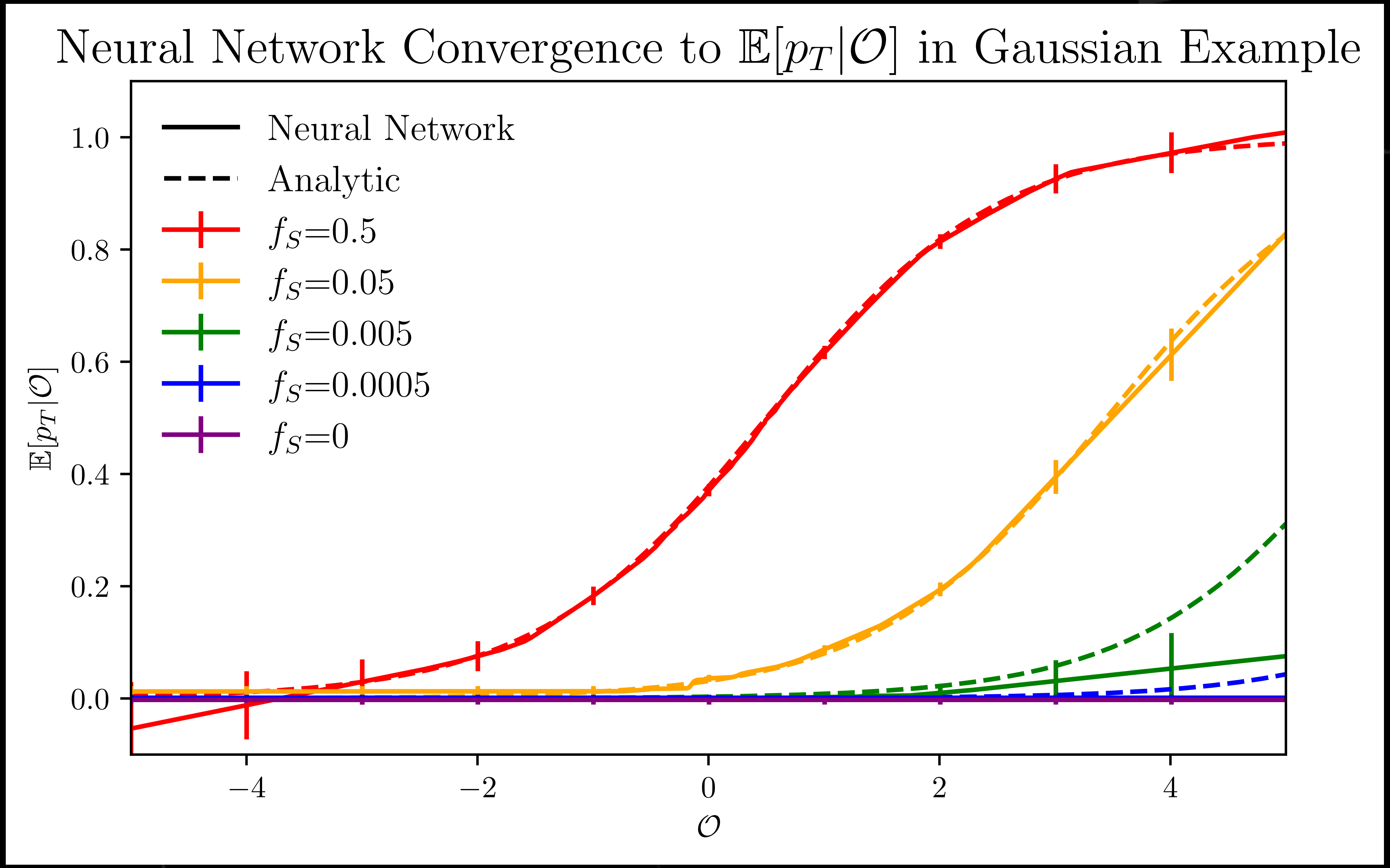
Random classifier in low signal limit

Convergence for Gaussian Simulation



Convergence for Gaussian Simulation

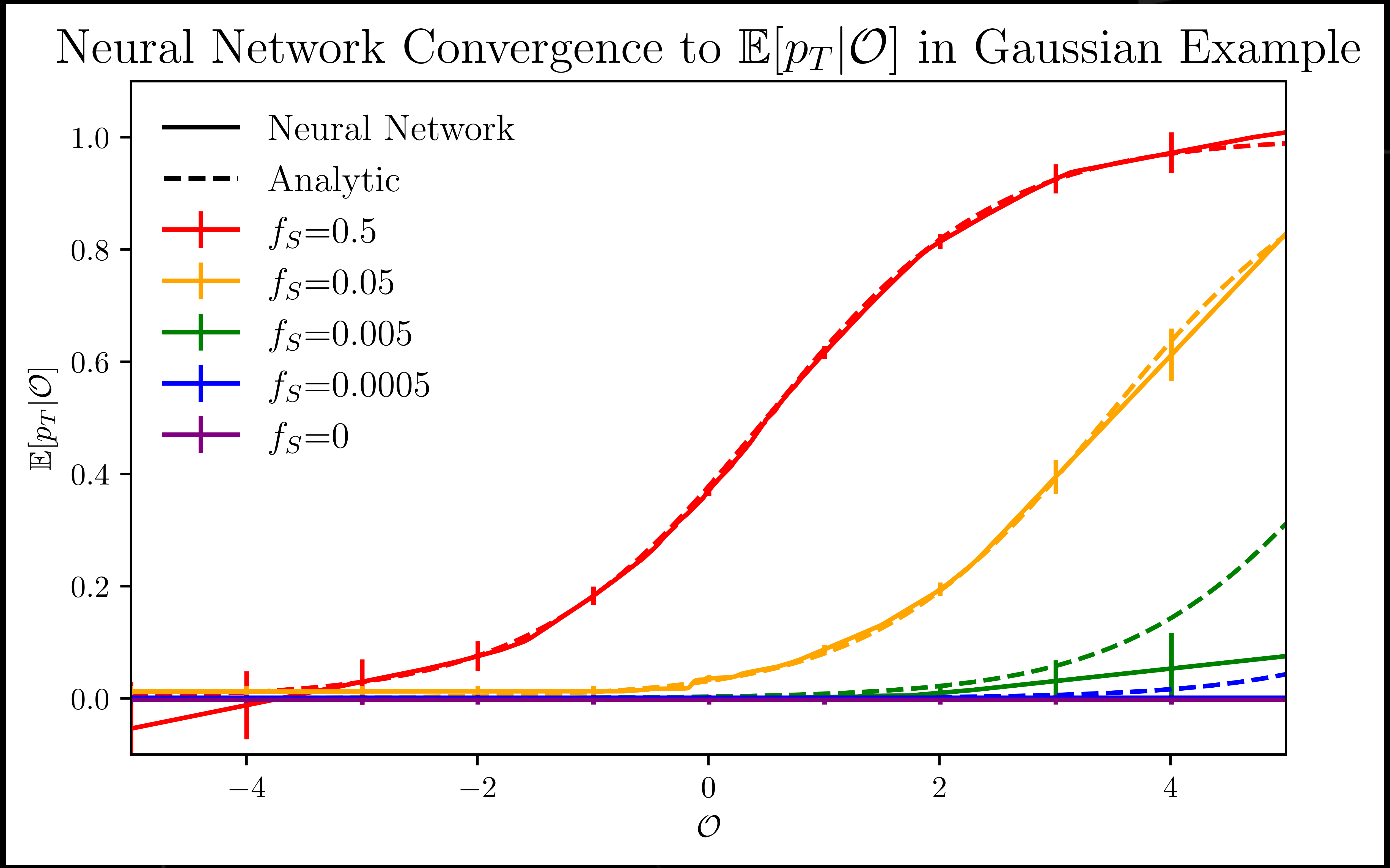
Since we constructed from Gaussians, can compute $\mathbb{E}[p_T \,|\, \mathcal{O}]$ analytically



Convergence for Gaussian Simulation

Since we constructed from Gaussians, can compute $\mathbb{E}[p_T \,|\, \mathcal{O}]$ analytically

Find solid convergence in large signal limit, with decaying performance as signal fraction decreases



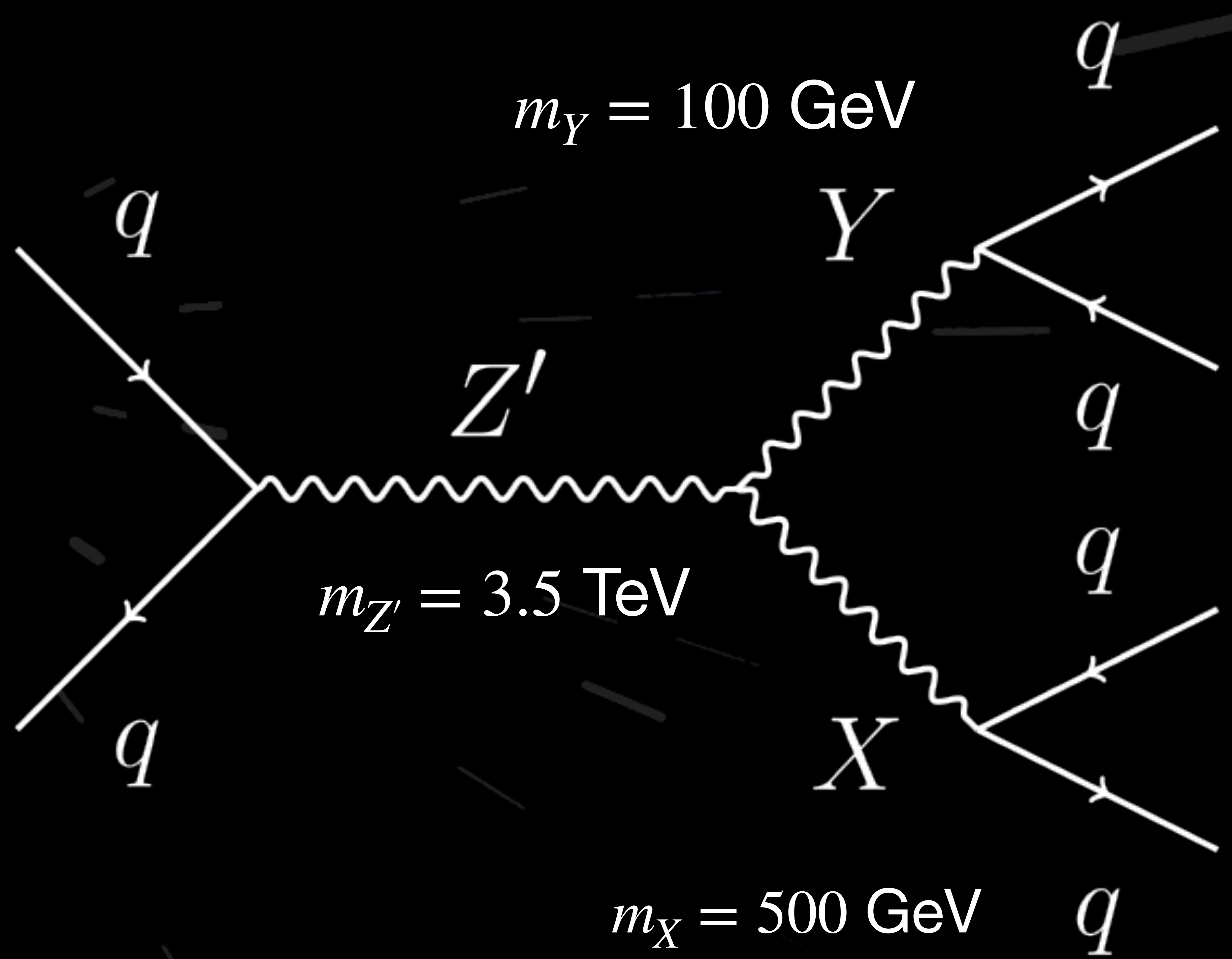
LHC Olympics

R&D Dataset

LHC Olympics

R&D Dataset

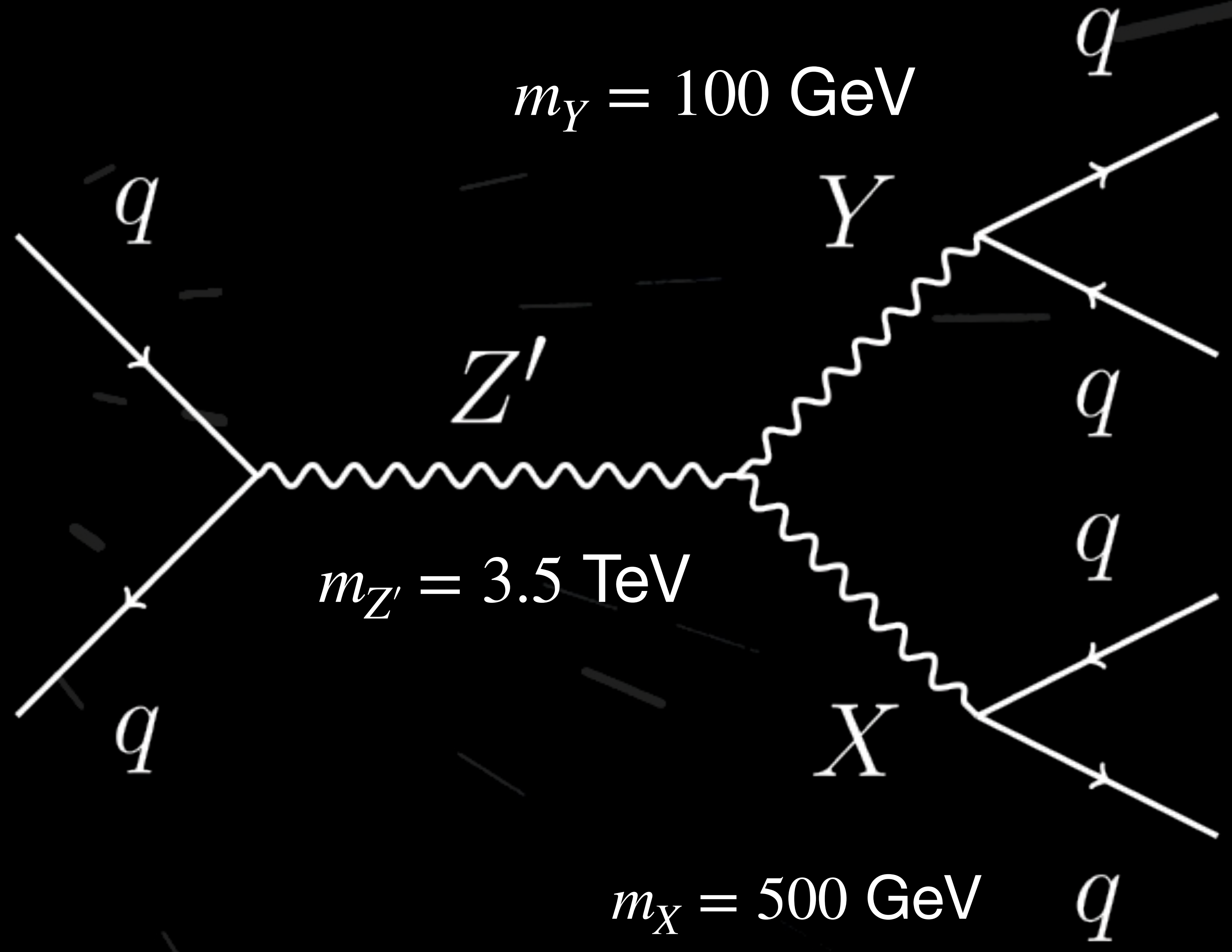
[2101.08320]



LHC Olympics

R&D Dataset

[2101.08320]

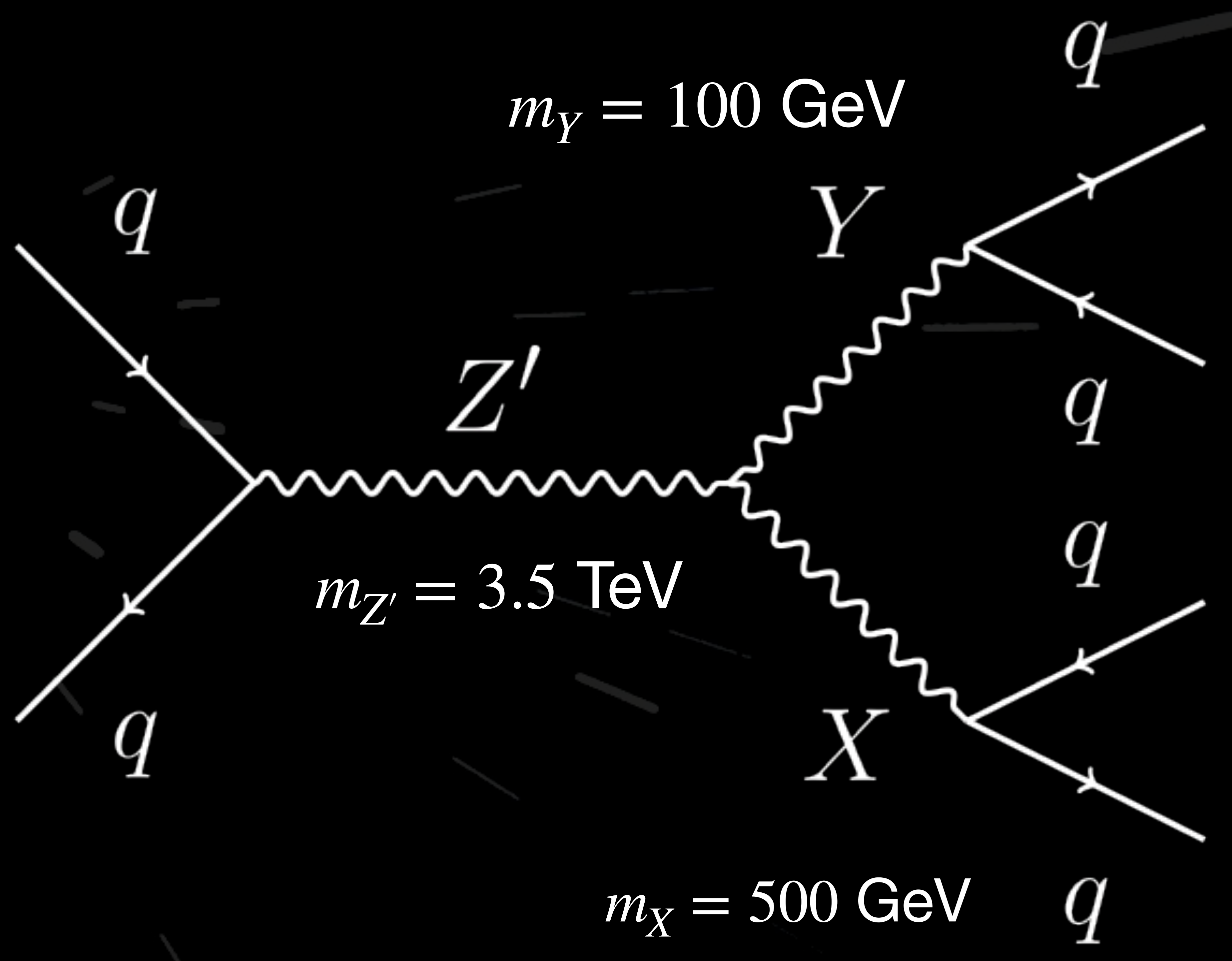


While hypothesis is resonance, FORCE doesn't use resonance info

LHC Olympics

R&D Dataset

[2101.08320]



While hypothesis is resonance, FORCE doesn't use resonance info

$\Longrightarrow$ Can combine FORCE with other bump hunting algorithms

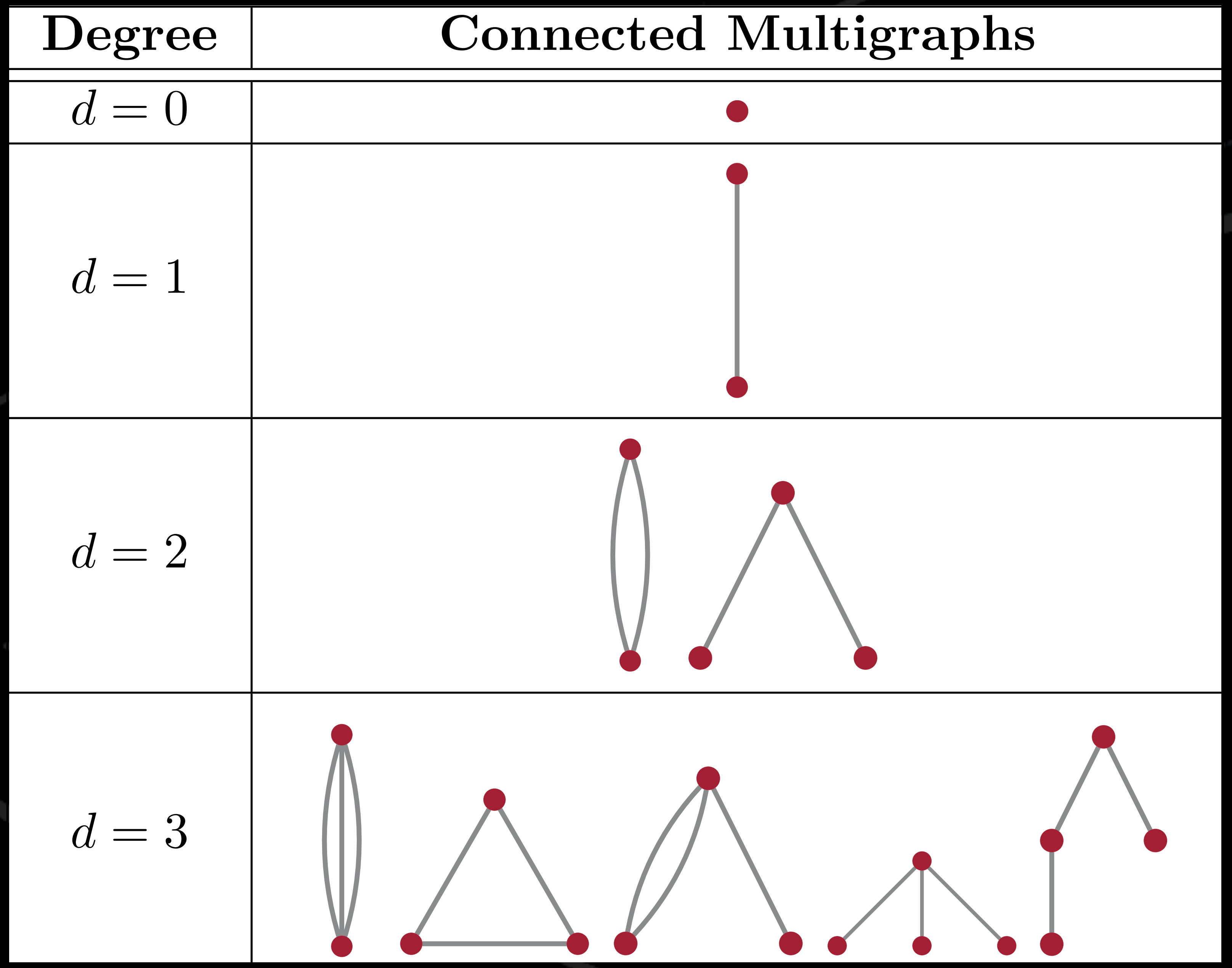
What’s our scale-/boost-invariant substructure observables?

What’s our scale-/boost-invariant substructure observables?

Energy flow polynomials - systematic expansion in energy and angle

What’s our scale-/boost-invariant substructure observables?

Energy flow polynomials - systematic expansion in energy and angle

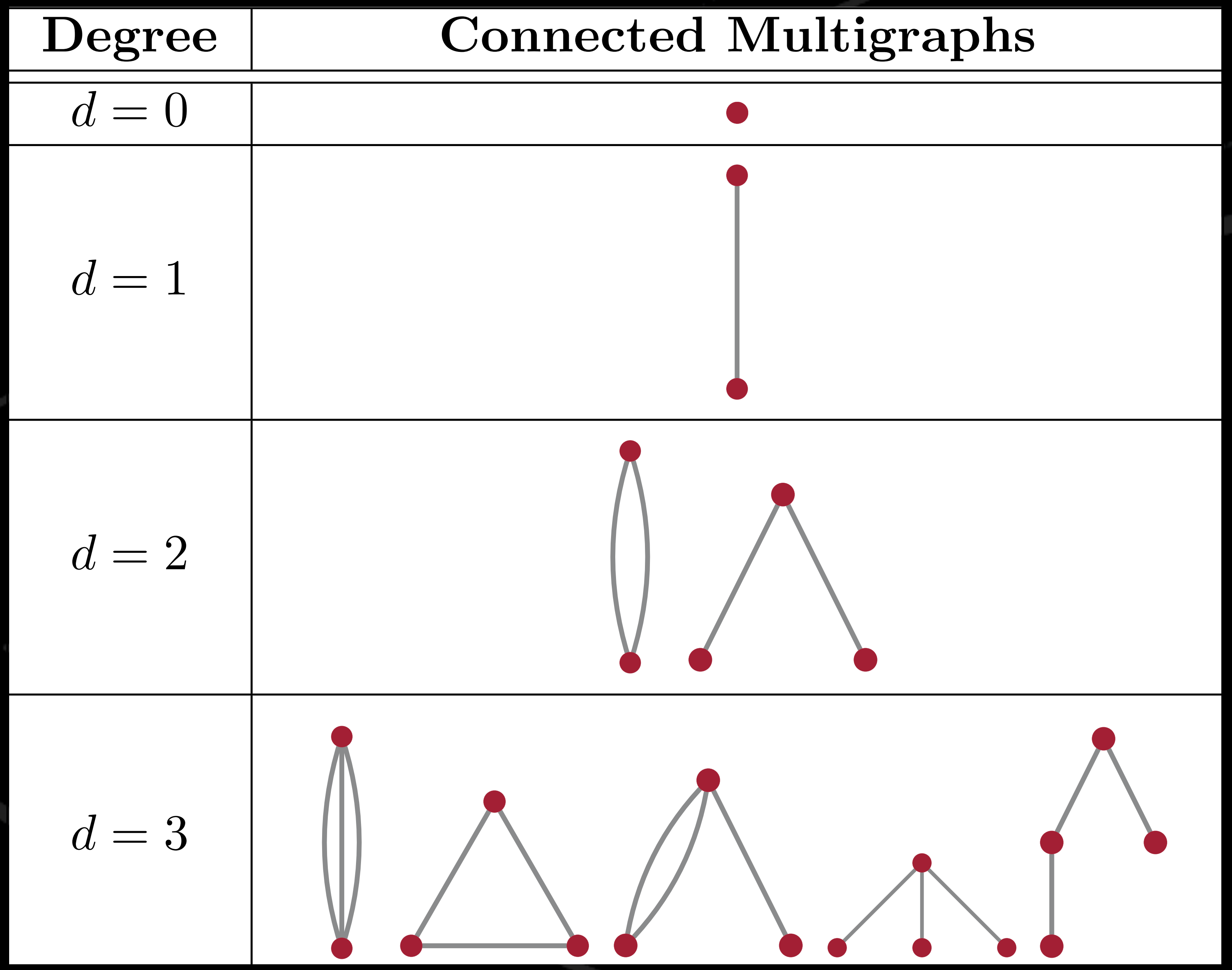


[1712.07124]

What's our scale-/boost-invariant substructure observables?

Energy flow polynomials - systematic expansion in energy and angle

Take EFPs $d \leq 3 \implies$ 13 features



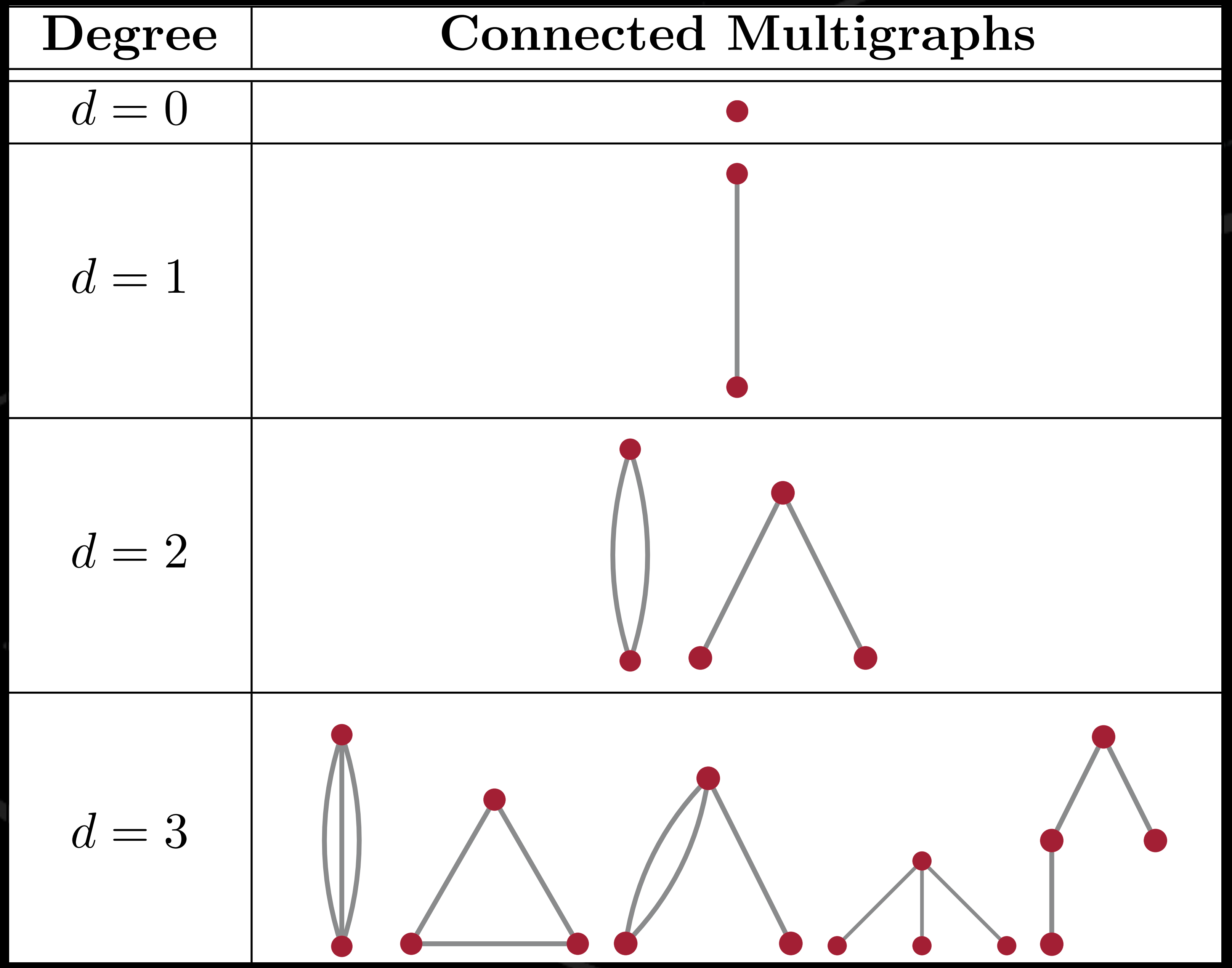
[1712.07124]

What's our scale-/boost-invariant substructure observables?

Energy flow polynomials - systematic expansion in energy and angle

Take EFPs $d \leq 3 \implies$ 13 features

Transverse boosts scale



[1712.07124]

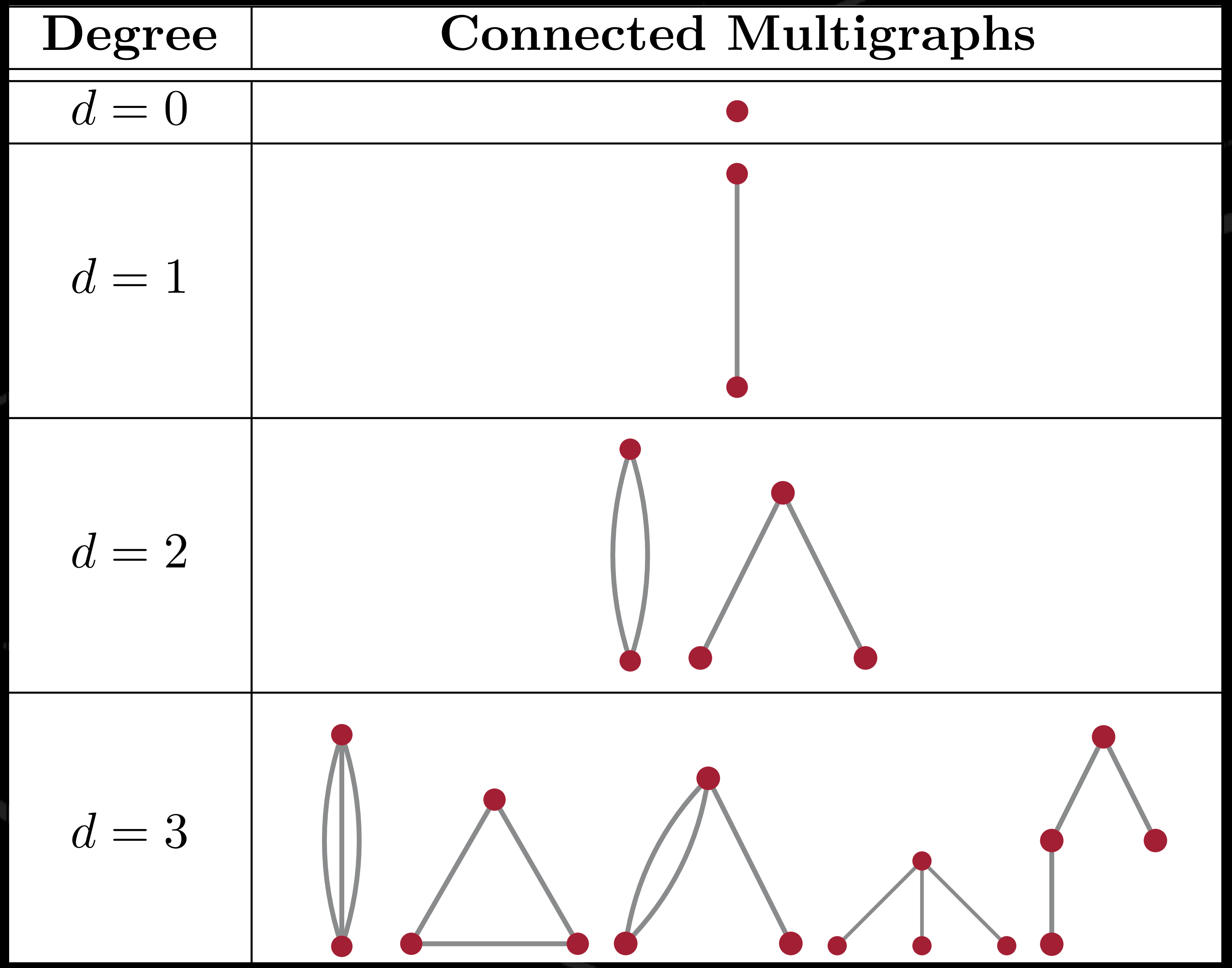
What’s our scale-/boost-invariant substructure observables?

Energy flow polynomials - systematic expansion in energy and angle

Take EFPs $d \leq 3 \implies$ 13 features

Transverse boosts scale

Energy: γ



[1712.07124]

What’s our scale-/boost-invariant substructure observables?

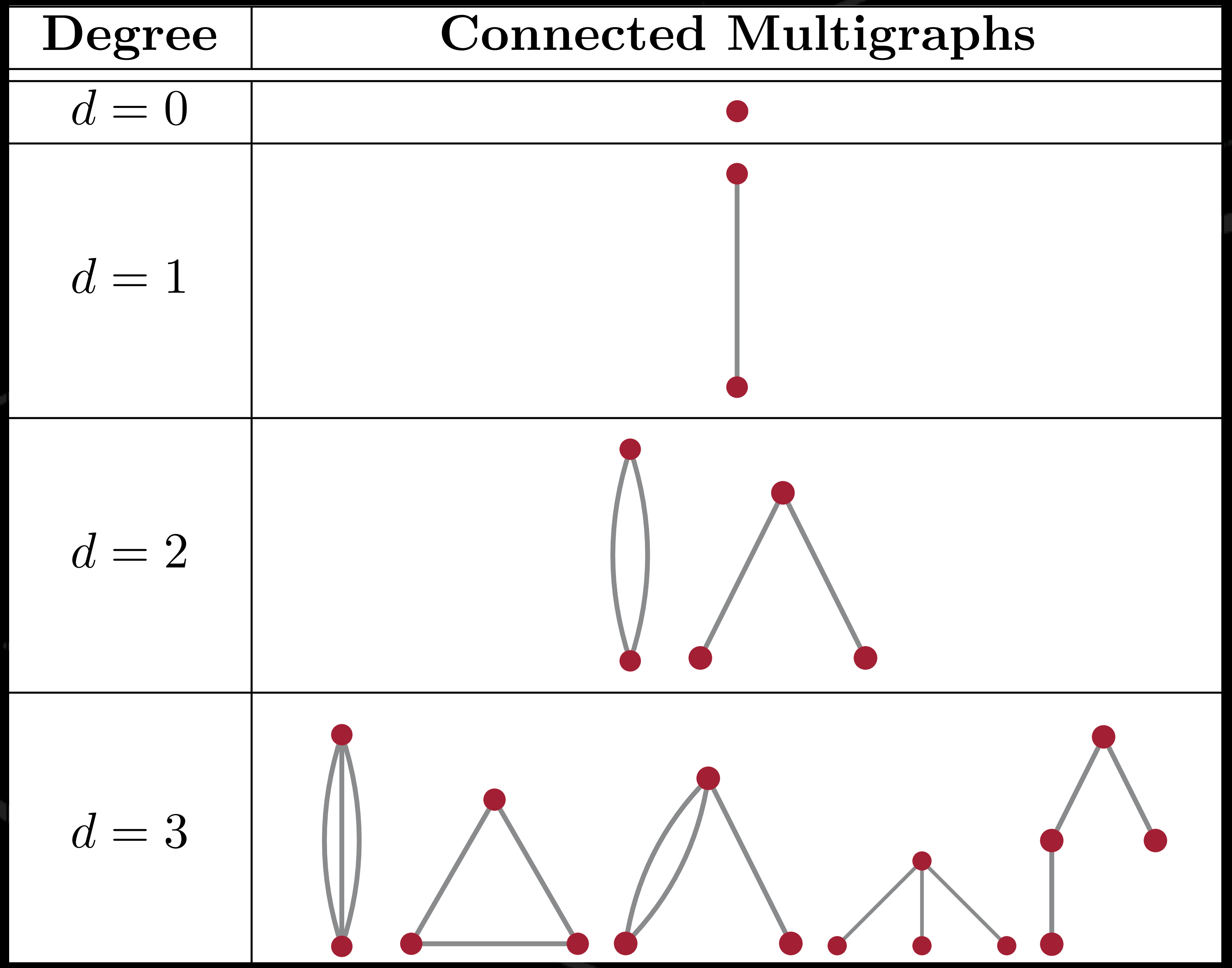
Energy flow polynomials - systematic expansion in energy and angle

Take EFPs $d \leq 3 \implies$ 13 features

Transverse boosts scale

Energy: γ

Angle: $1/\gamma$



[1712.07124]

What's our scale-/boost-invariant substructure observables?

Energy flow polynomials - systematic expansion in energy and angle

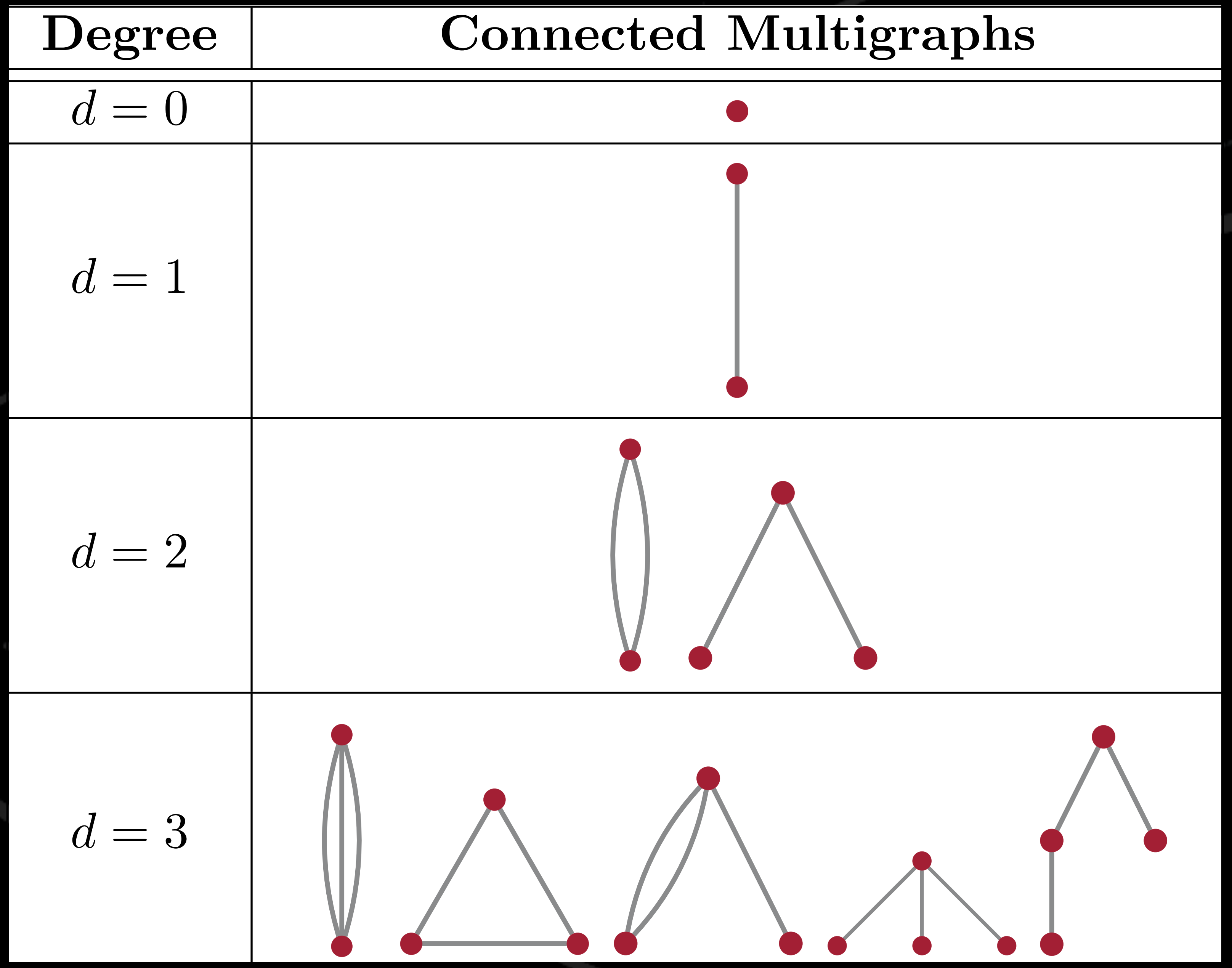
Take EFPs $d \leq 3 \implies$ 13 features

Transverse boosts scale

Energy: γ

Angle: $1/\gamma$

Introduce normalized EFPs



[1712.07124]

What's our scale-/boost-invariant substructure observables?

Energy flow polynomials - systematic expansion in energy and angle

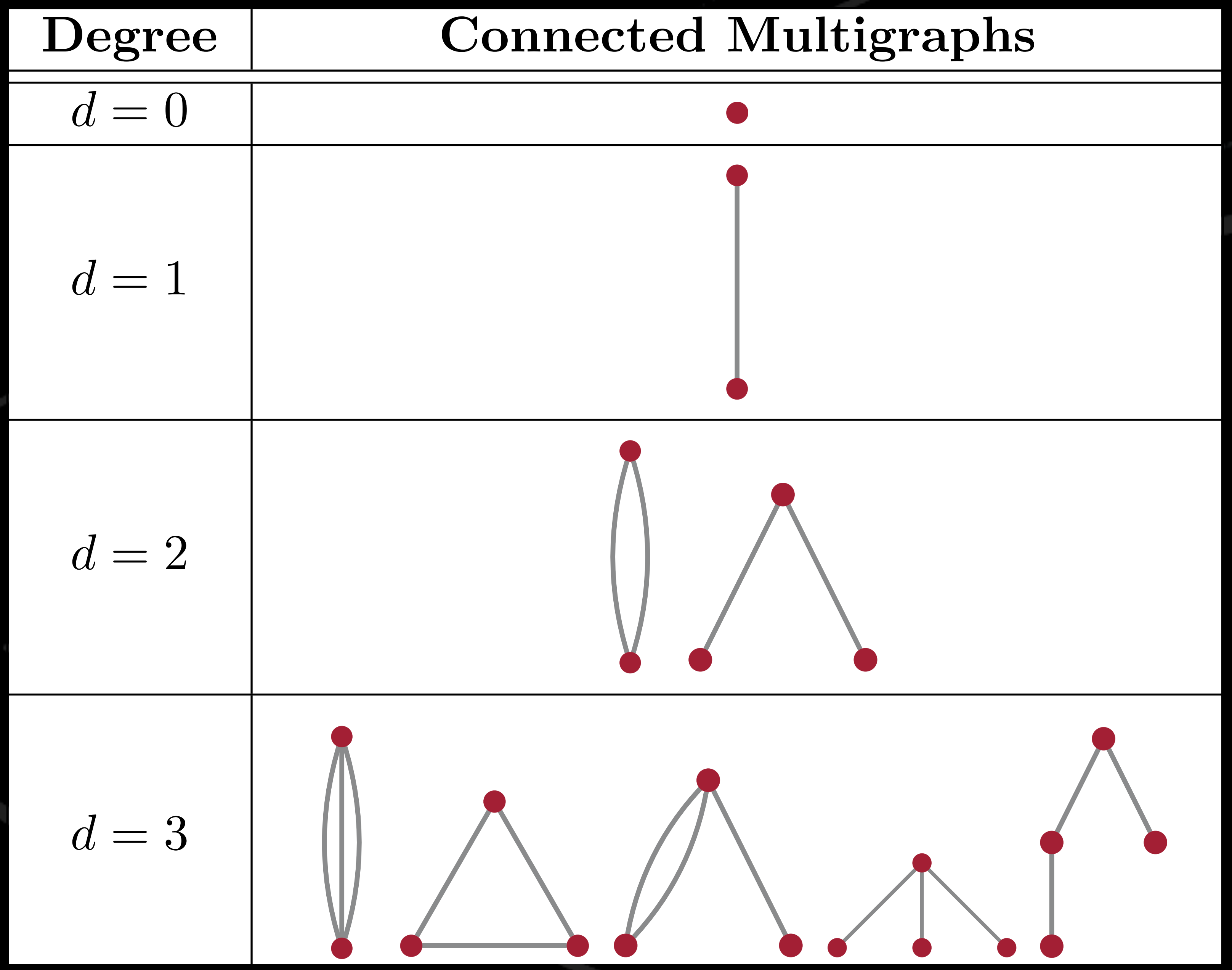
Take EFPs $d \leq 3 \implies$ 13 features

Transverse boosts scale

Energy: γ

Angle: $1/\gamma$

Introduce normalized EFPs $\implies$



[1712.07124]

What's our scale-/boost-invariant substructure observables?

Energy flow polynomials - systematic expansion in energy and angle

Take EFPs $d \leq 3 \implies$ 13 features

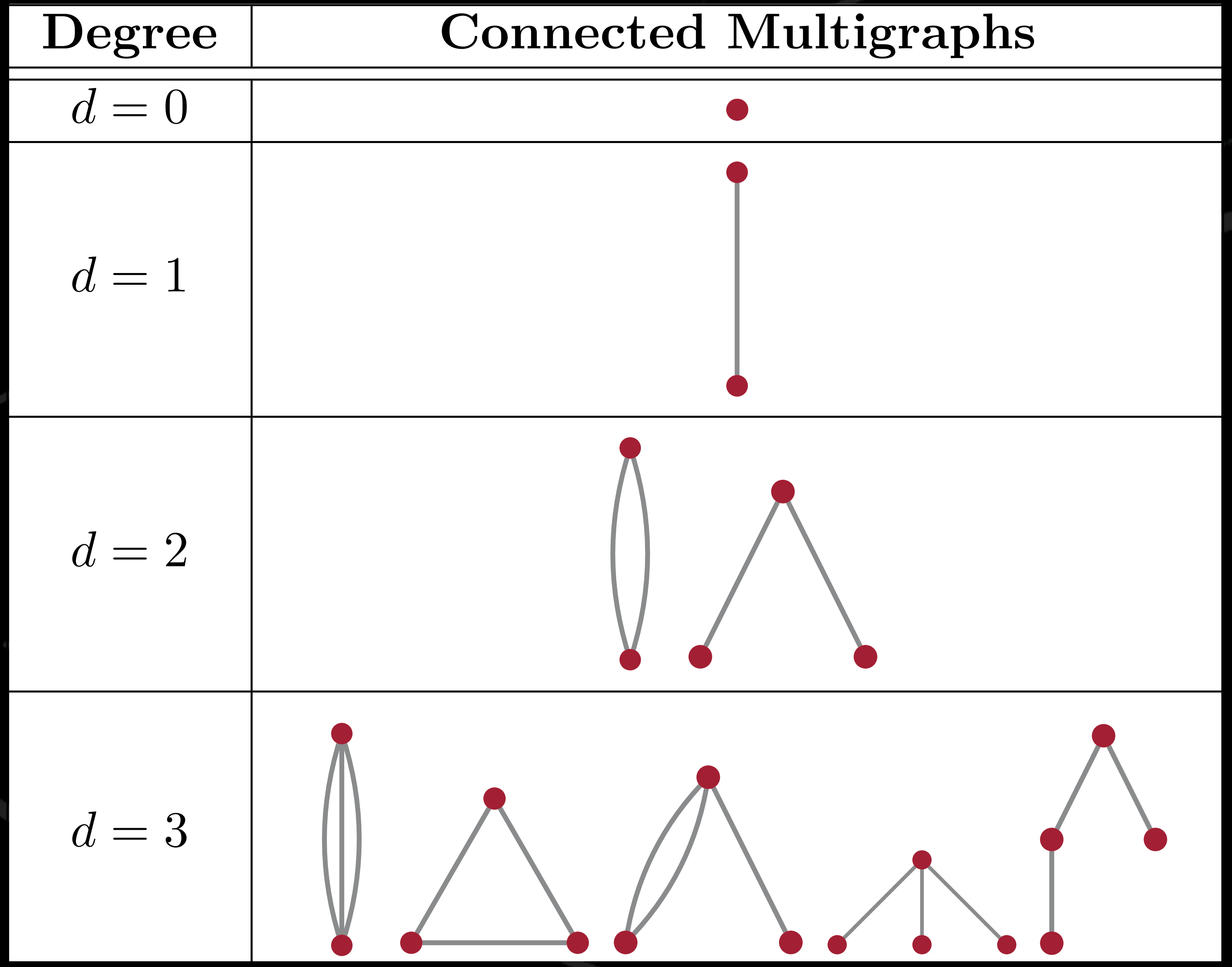
Transverse boosts scale

Energy: γ

Angle: $1/\gamma$

Introduce normalized EFPs $\implies$

$$\text{EFP} \rightarrow \frac{\text{EFP}}{\left(\sum_{i=1}^{M} p_{T,i}\right)^{N-2d} \left(\sum_{i=1}^{M} \sum_{j=1}^{M} p_{T,i} p_{T,j} \theta_{ij}\right)^{d}}$$



[1712.07124]

What's our scale-/boost-invariant substructure observables?

Energy flow polynomials - systematic expansion in energy and angle

Take EFPs $d \leq 3 \implies$ 13 features

Transverse boosts scale

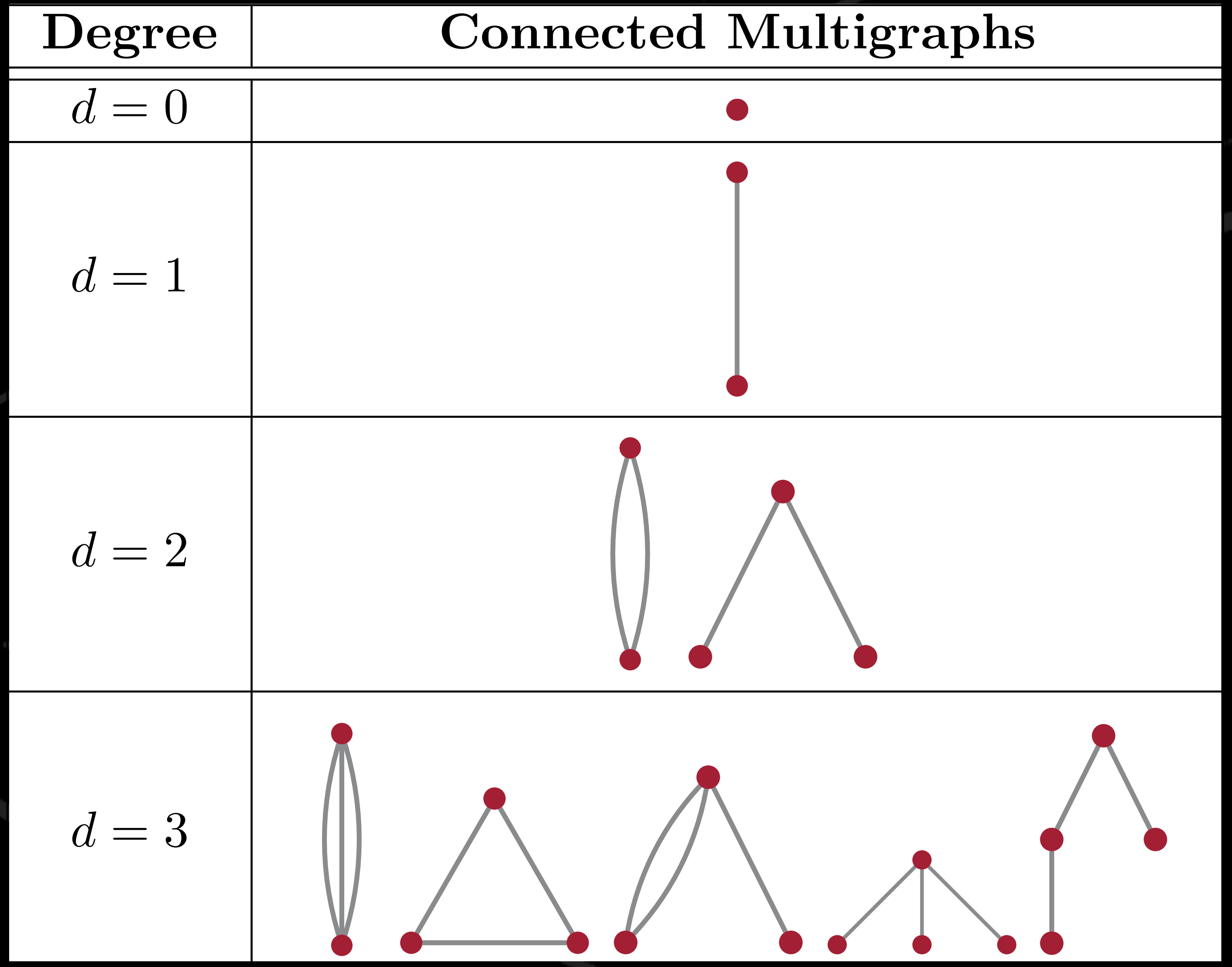
Energy: γ

Angle: $1/\gamma$

Introduce normalized EFPs $\implies$

$$\text{EFP} \to \frac{\text{EFP}}{\left(\sum_{i=1}^{M} p_{T,i}\right)^{N-2d} \left(\sum_{i=1}^{M}\sum_{j=1}^{M} p_{T,i}p_{T,j}\theta_{ij}\right)^{d}}$$

$13 \to 8$ independent observables



[1712.07124]

ROC-AUC for LHCO

ROC-AUC for LHCO

- Fully connected network
- 3 layers of 100 nodes
- ReLU Activation
- Dropout and L2 Regularization
- Mean across 10 models

ROC-AUC for LHCO

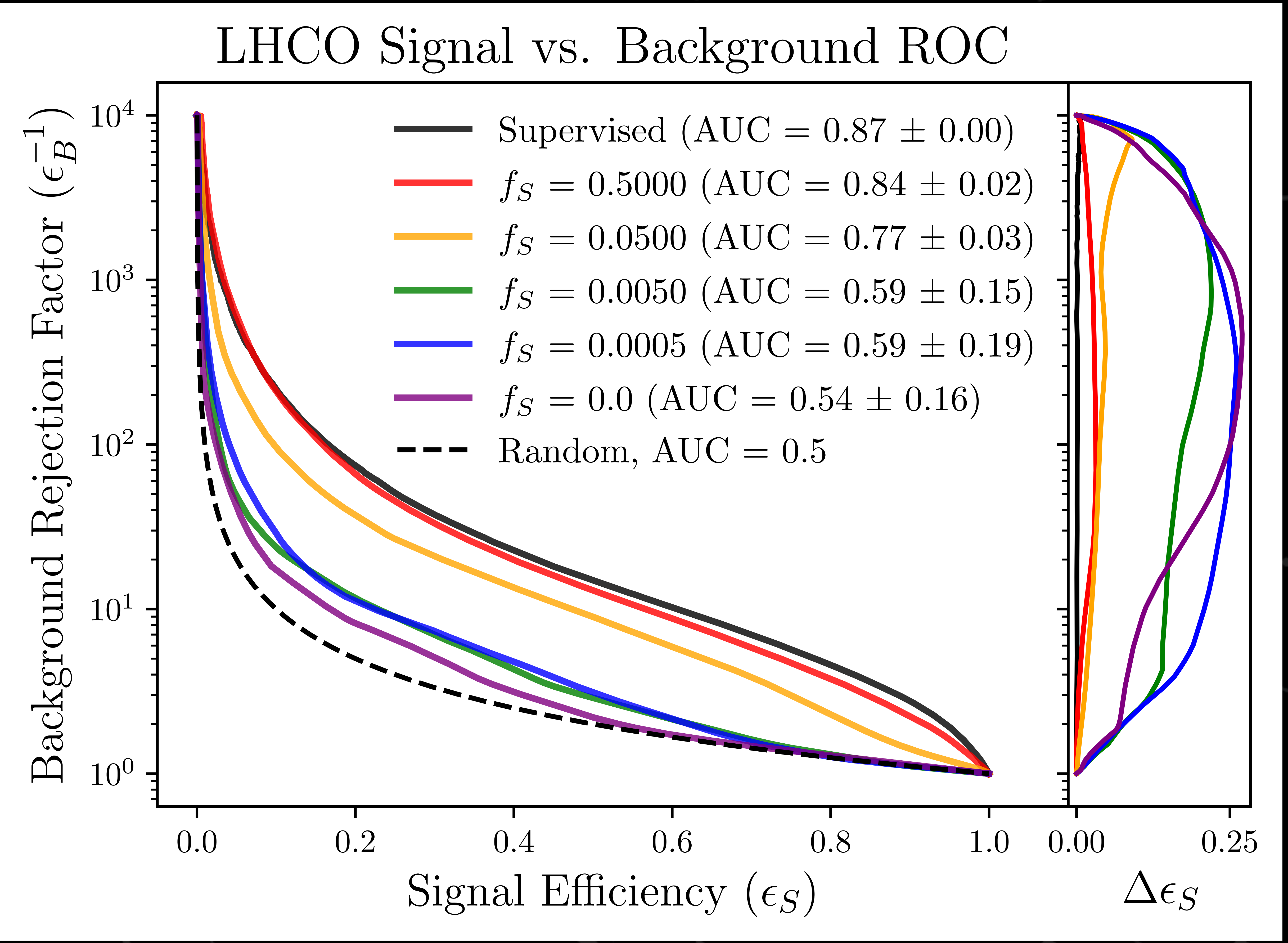
- Fully connected network
- 3 layers of 100 nodes
- ReLU Activation
- Dropout and L2 Regularization
- Mean across 10 models

$\Longrightarrow$

ROC-AUC for LHCO

- Fully connected network
- 3 layers of 100 nodes
- ReLU Activation
- Dropout and L2 Regularization
- Mean across 10 models

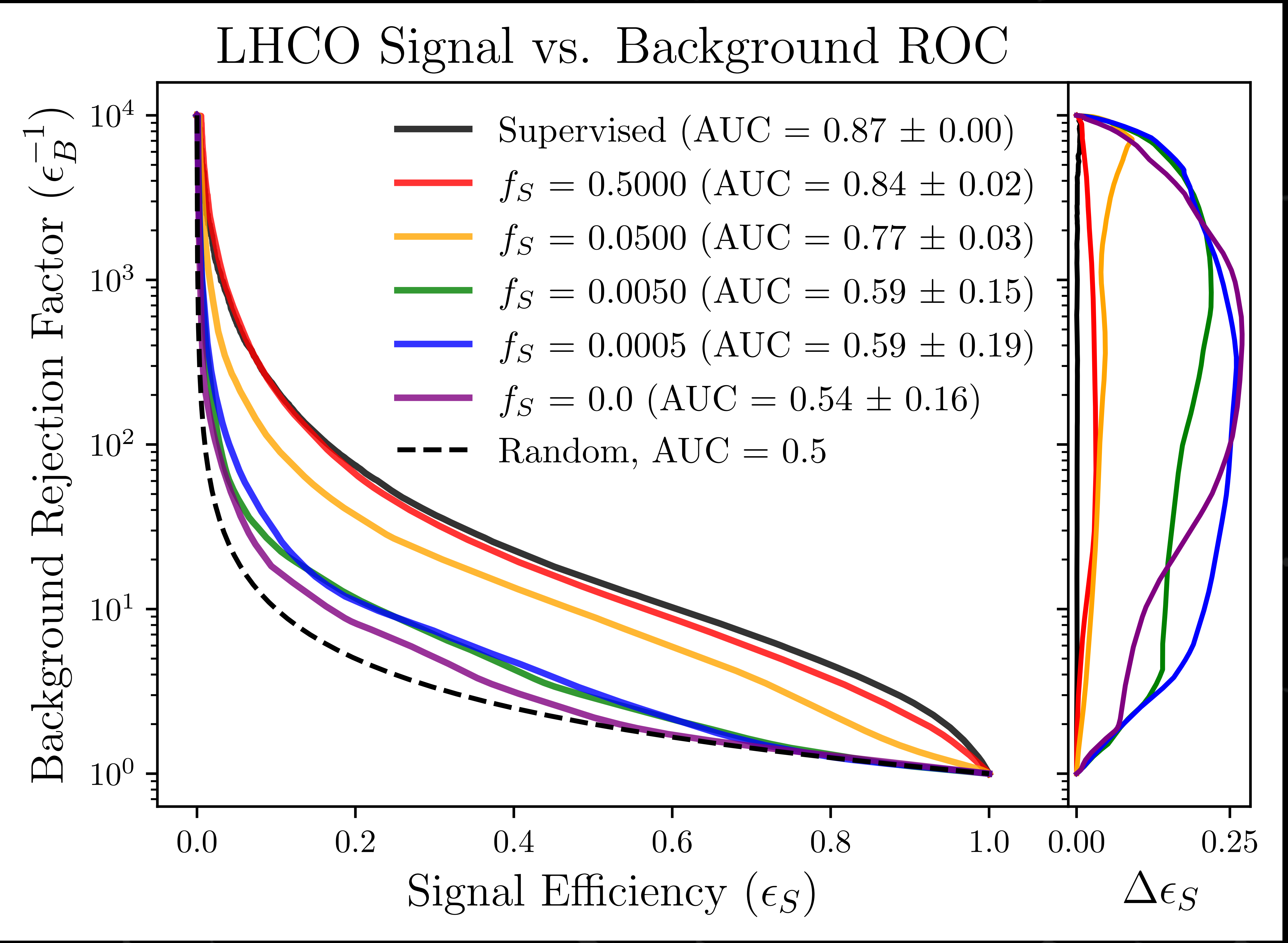
$\Longrightarrow$



ROC-AUC for LHCO

- Fully connected network
- 3 layers of 100 nodes
- ReLU Activation
- Dropout and L2 Regularization
- Mean across 10 models

$\Longrightarrow$

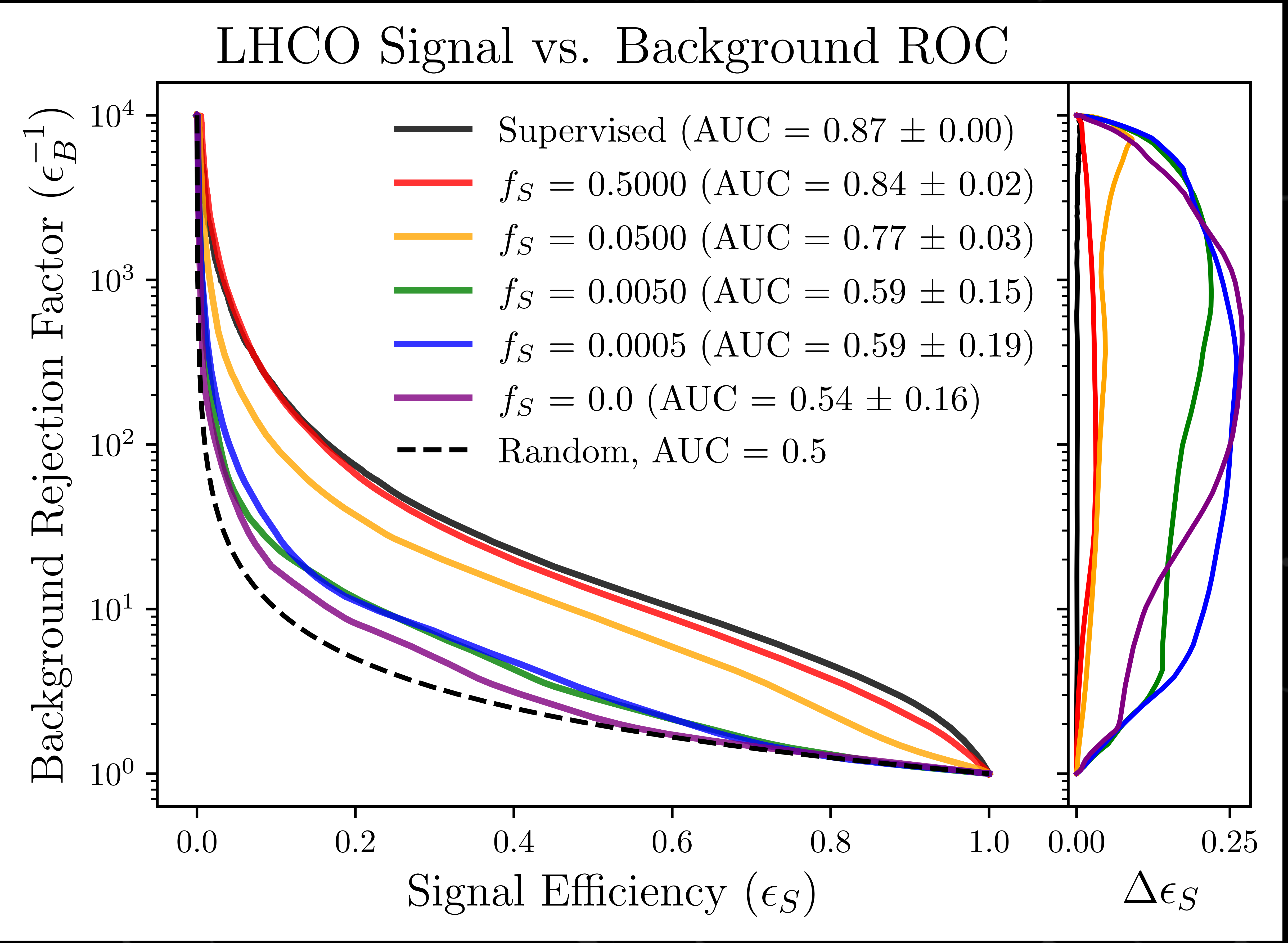


Near optimal in high signal limit

ROC-AUC for LHCO

- Fully connected network
- 3 layers of 100 nodes
- ReLU Activation
- Dropout and L2 Regularization
- Mean across 10 models

$\Longrightarrow$



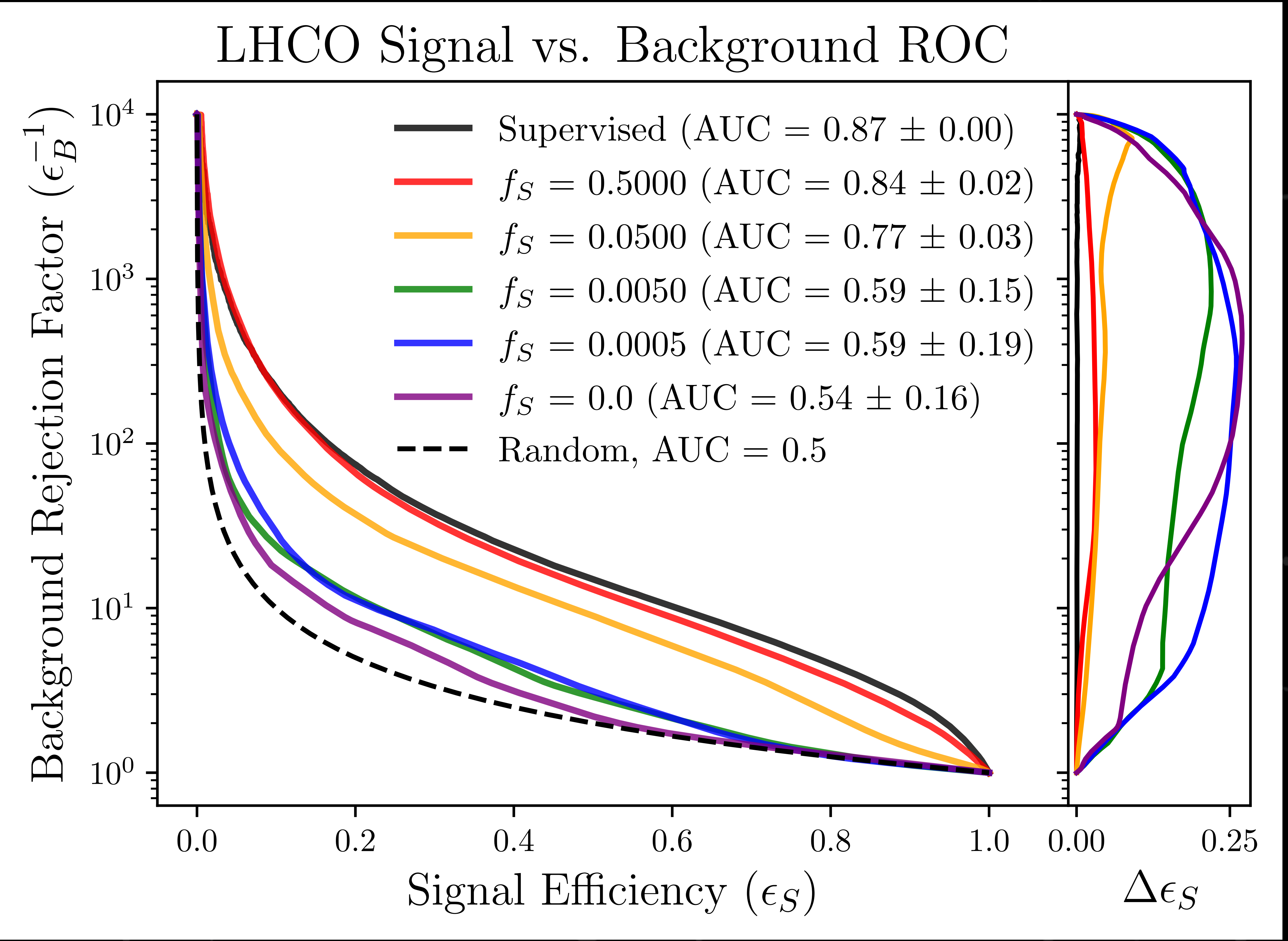
Near optimal in high signal limit

Non-trivial discrimination power for $f_S = 0$

ROC-AUC for LHCO

- Fully connected network
- 3 layers of 100 nodes
- ReLU Activation
- Dropout and L2 Regularization
- Mean across 10 models

$\Longrightarrow$



Near optimal in high signal limit

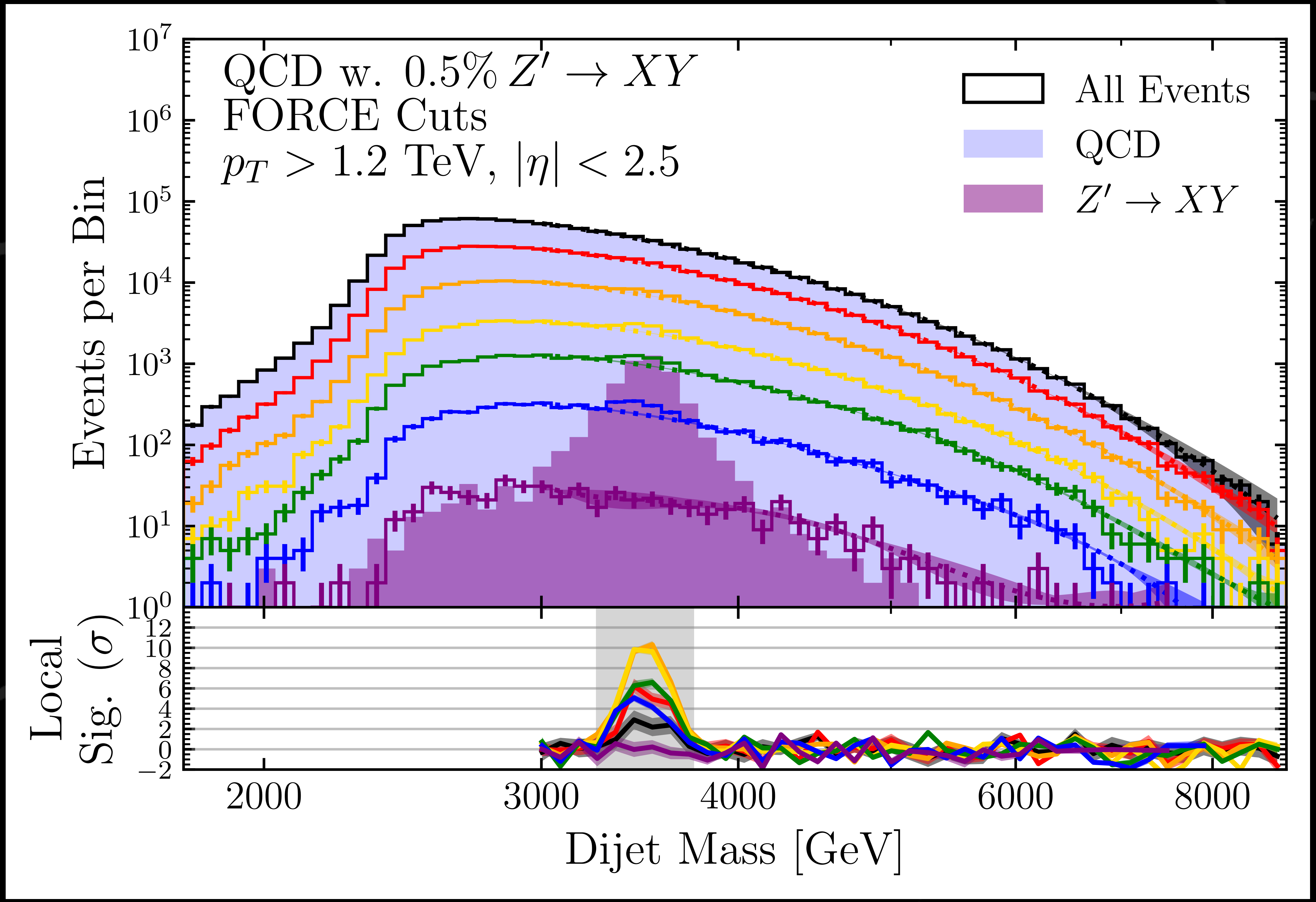
Non-trivial discrimination power for $f_S = 0$

More on that later

Let's go on a bump hunt!

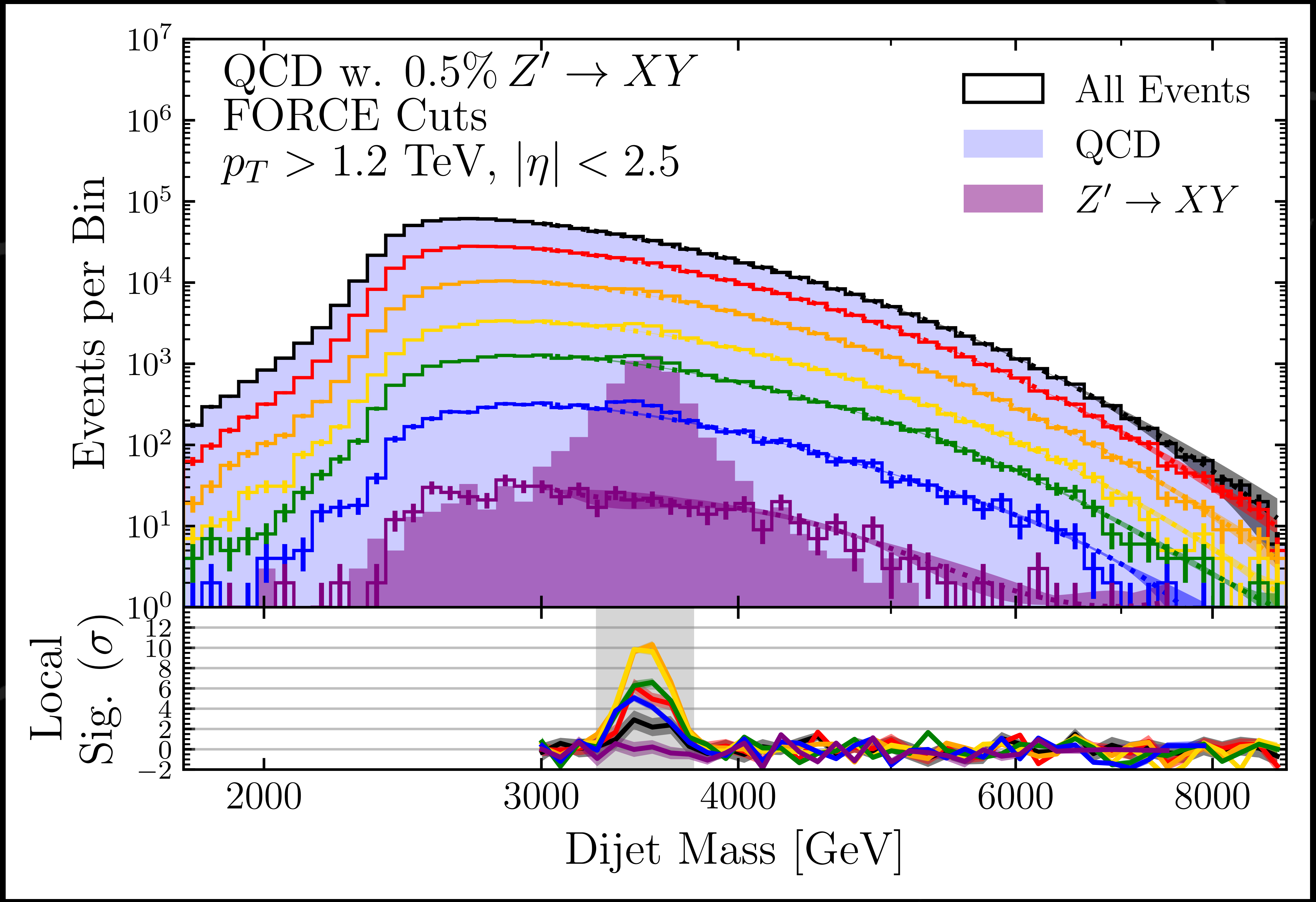
Let's go on a bump hunt!

We find the Z'

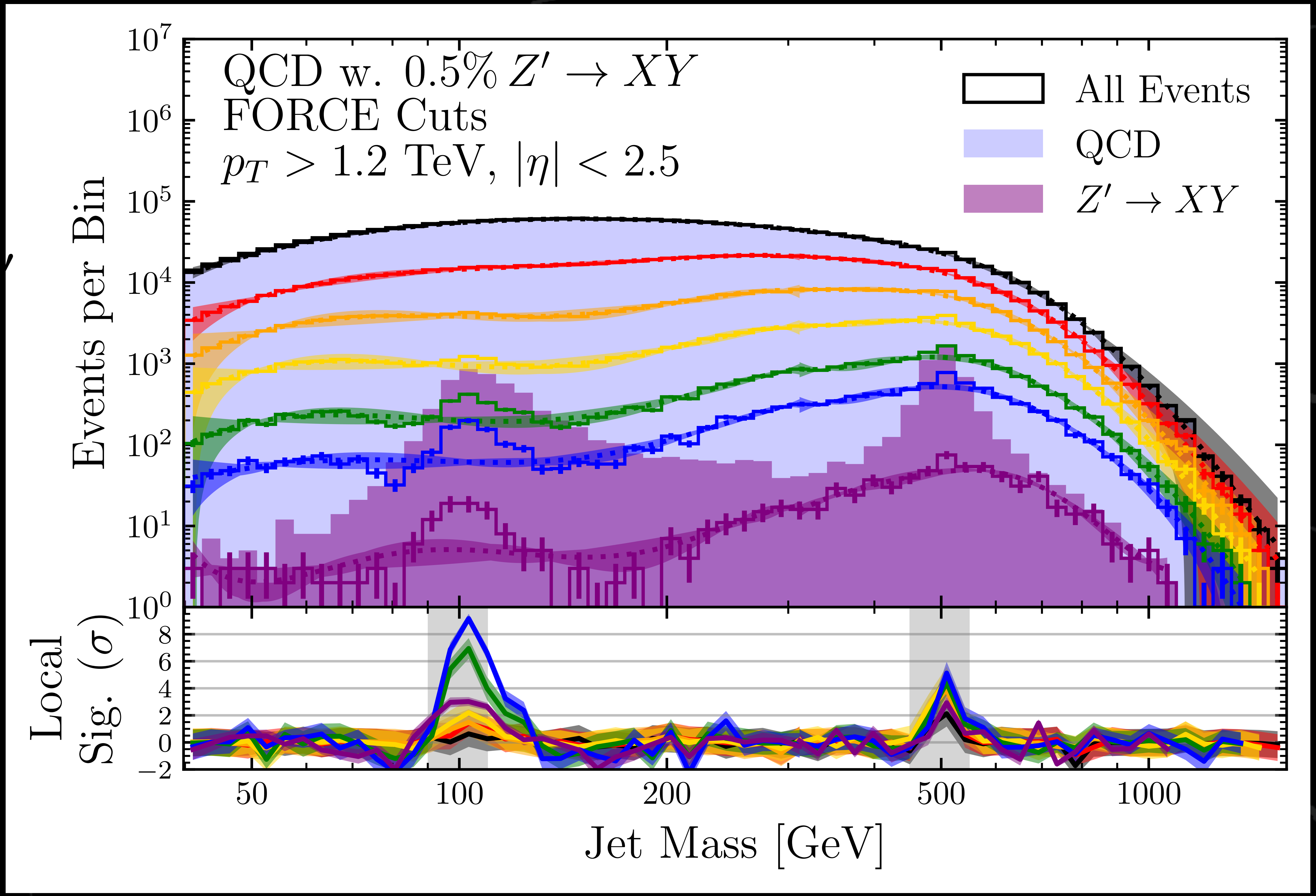


Let's go on a bump hunt!

We find the Z'



and the X and Y!



FORCE-ing Factorization with Shuffling

FORCE-ing Factorization with Shuffling

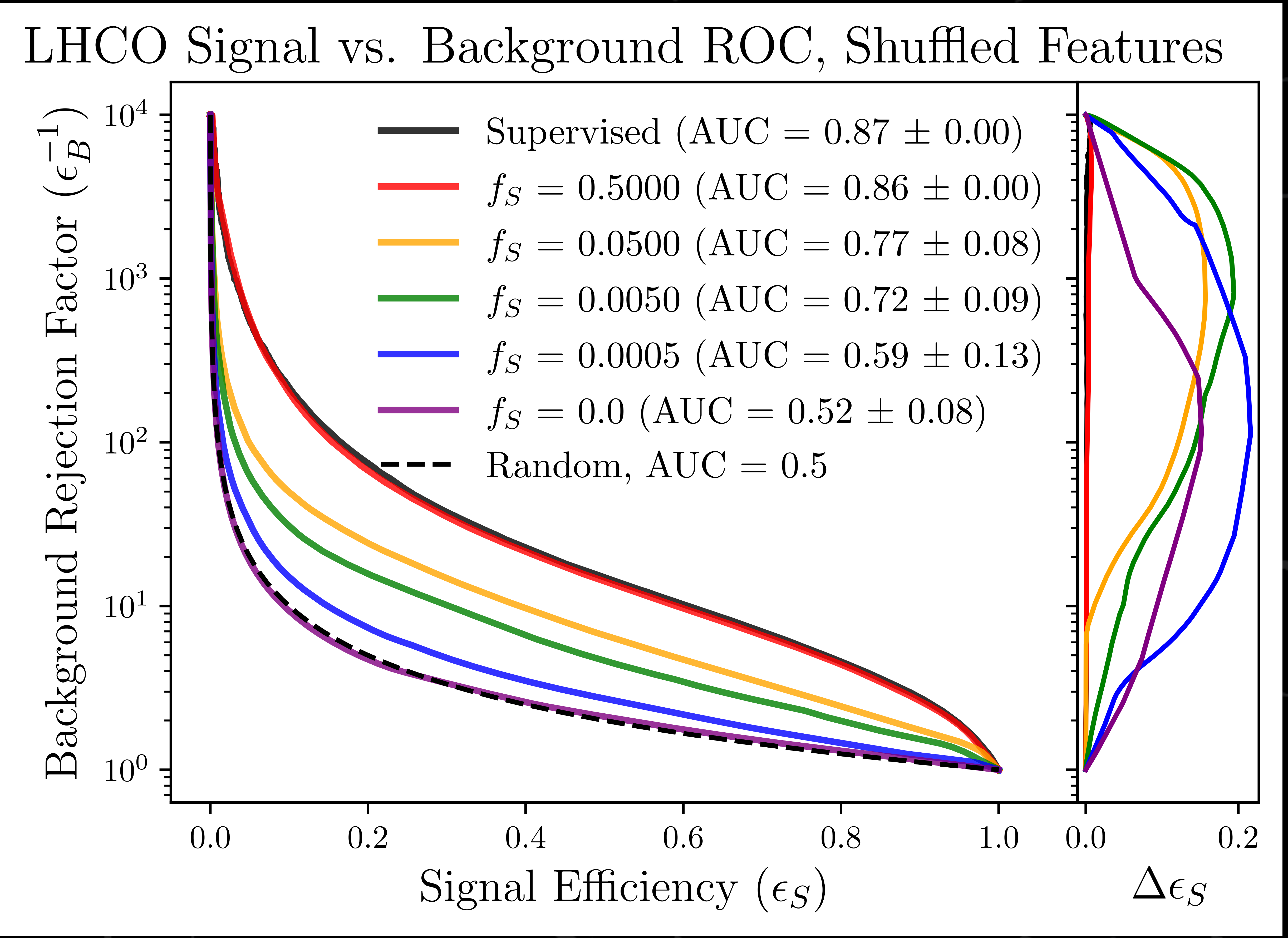
Explicitly construct factorized distribution by separately shuffling signal and background

FORCE-ing Factorization with Shuffling

Explicitly construct factorized distribution by separately shuffling signal and background $\Longrightarrow$

FORCE-ing Factorization with Shuffling

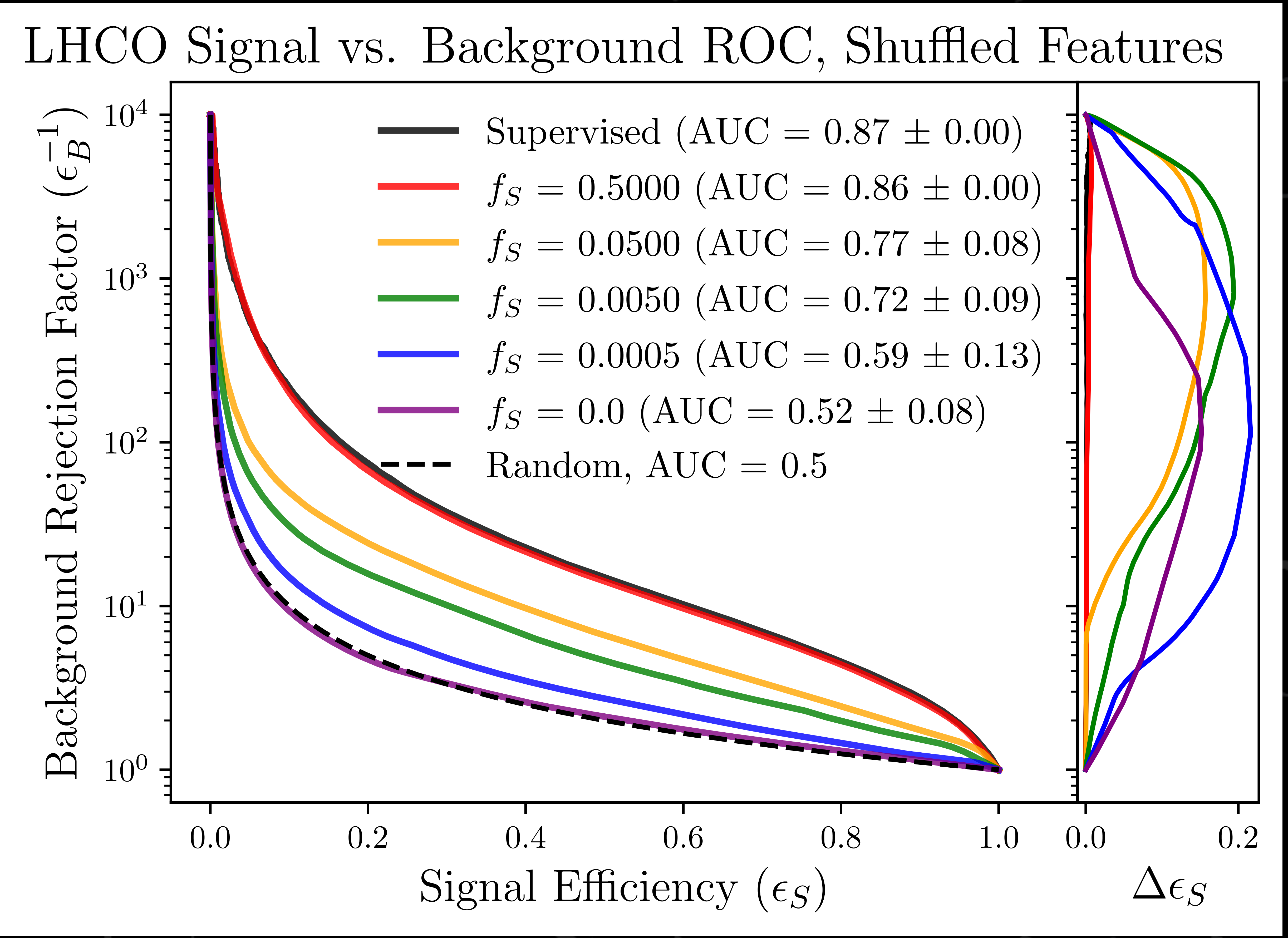
Explicitly construct factorized distribution by separately shuffling signal and background ⟹



FORCE-ing Factorization with Shuffling

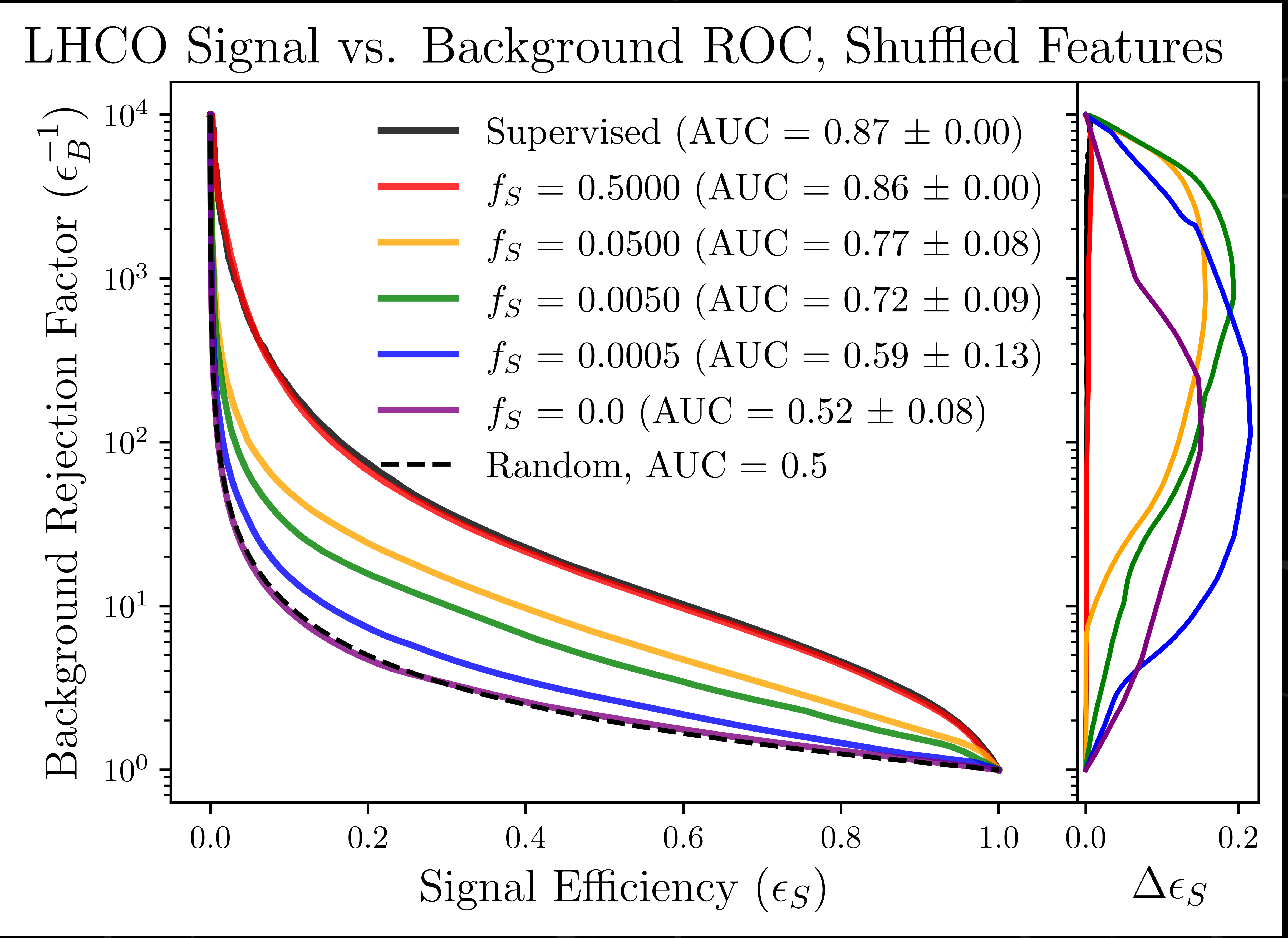
Explicitly construct factorized distribution by separately shuffling signal and background ⟹

Optimal performance in high signal limit



FORCE-ing Factorization with Shuffling

Explicitly construct factorized distribution by separately shuffling signal and background $\implies$

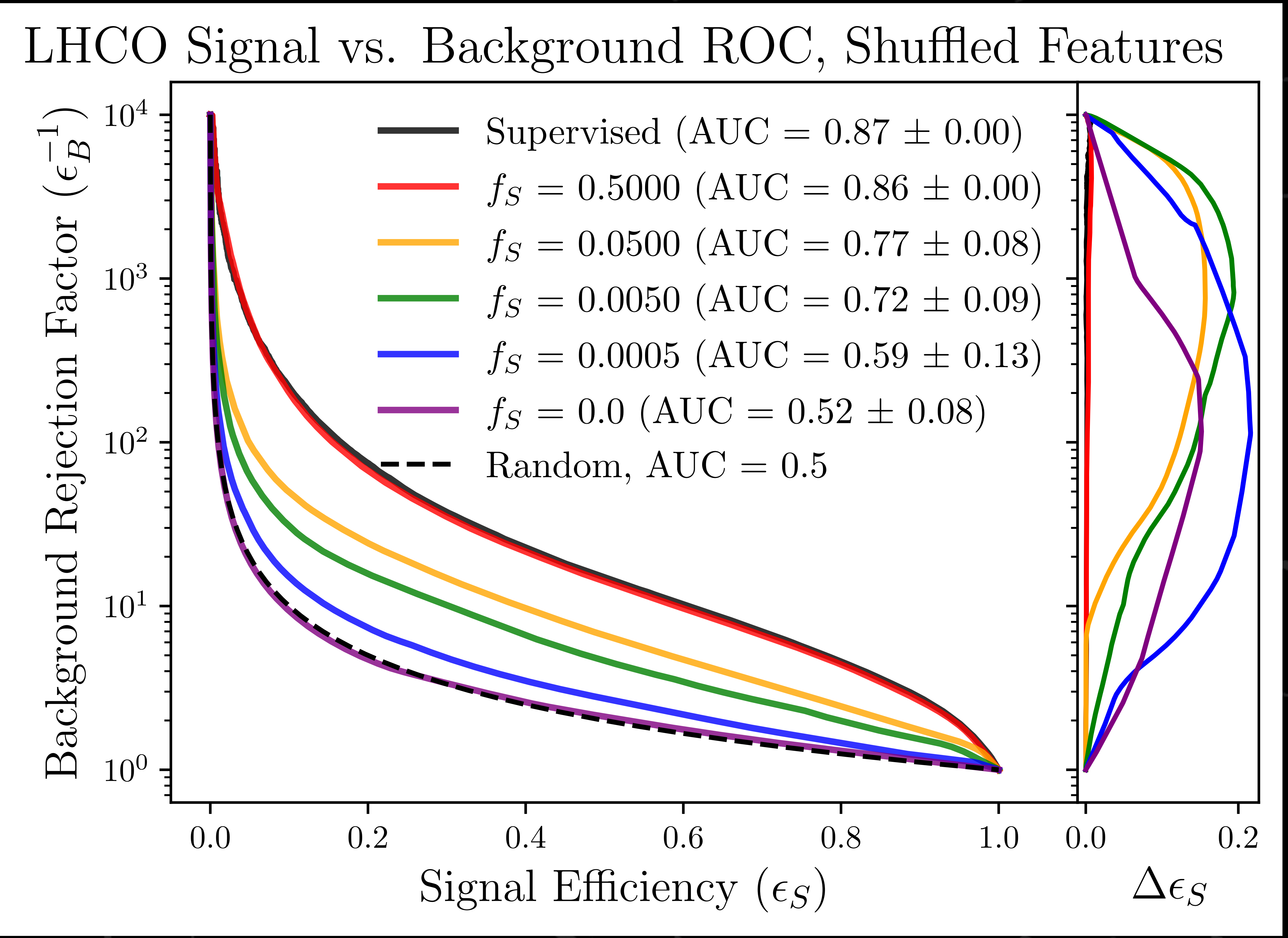


Optimal performance in high signal limit

Smooth decay of statistical power

FORCE-ing Factorization with Shuffling

Explicitly construct factorized distribution by separately shuffling signal and background $\Longrightarrow$

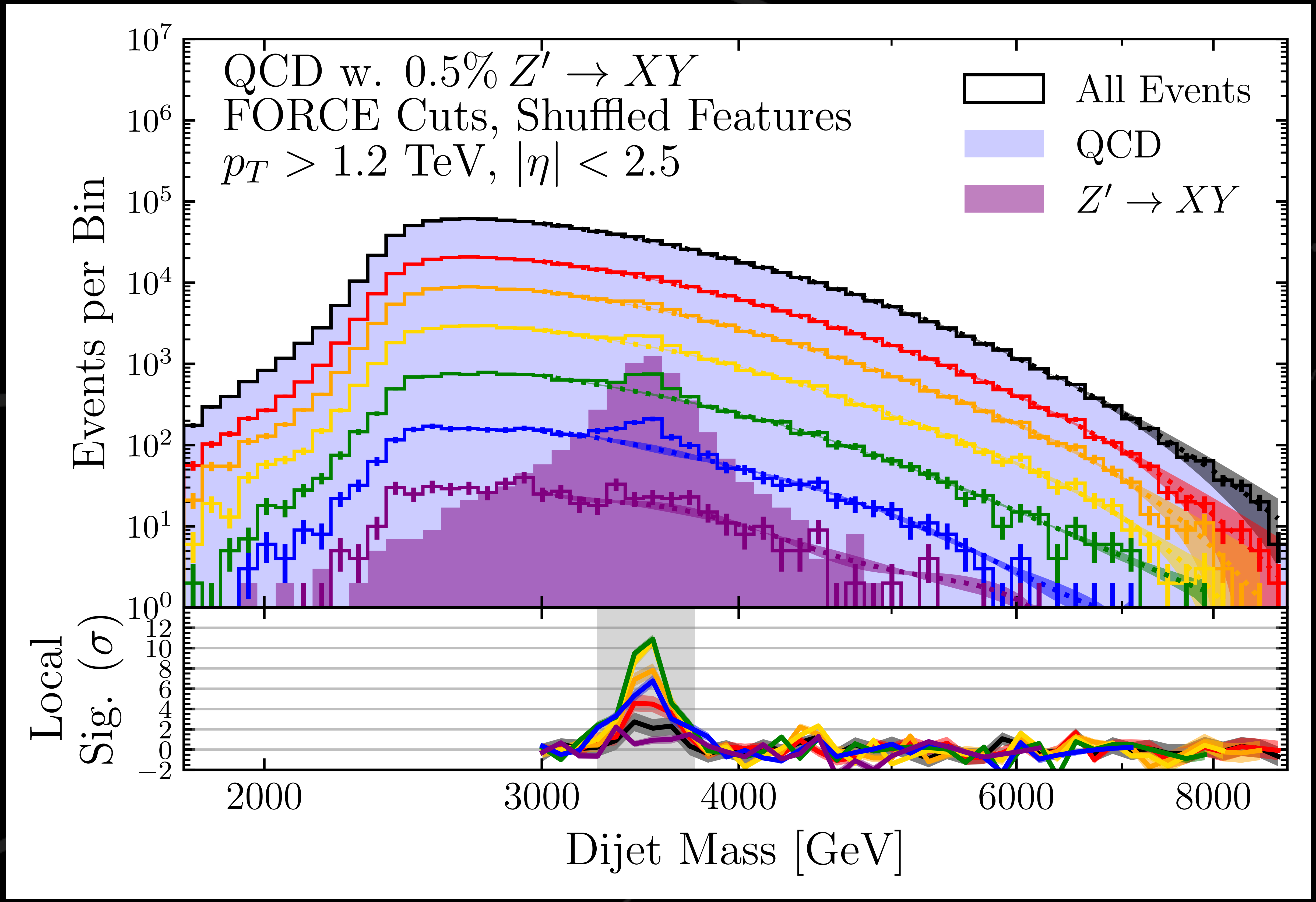


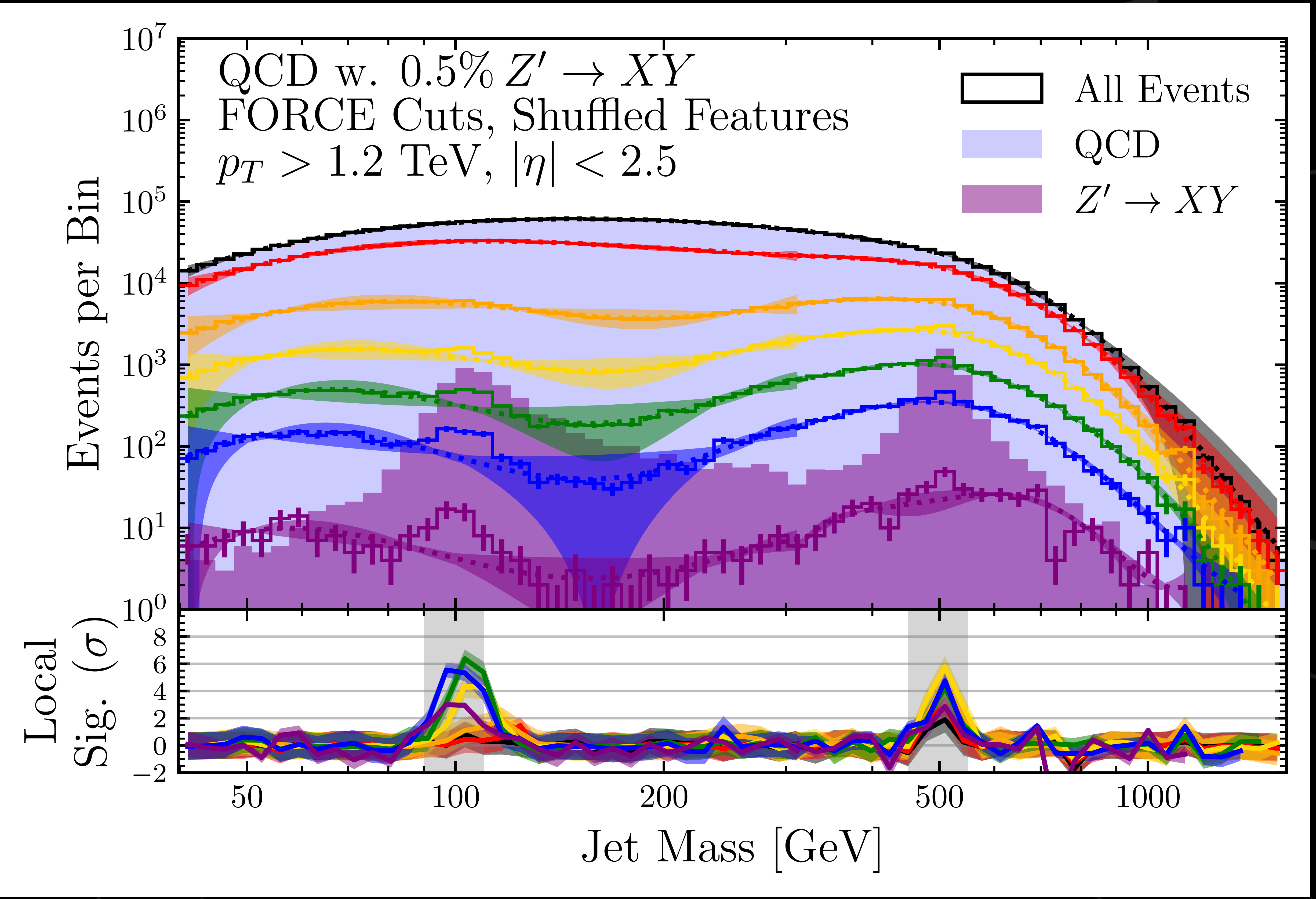
Optimal performance in high signal limit

Smooth decay of statistical power

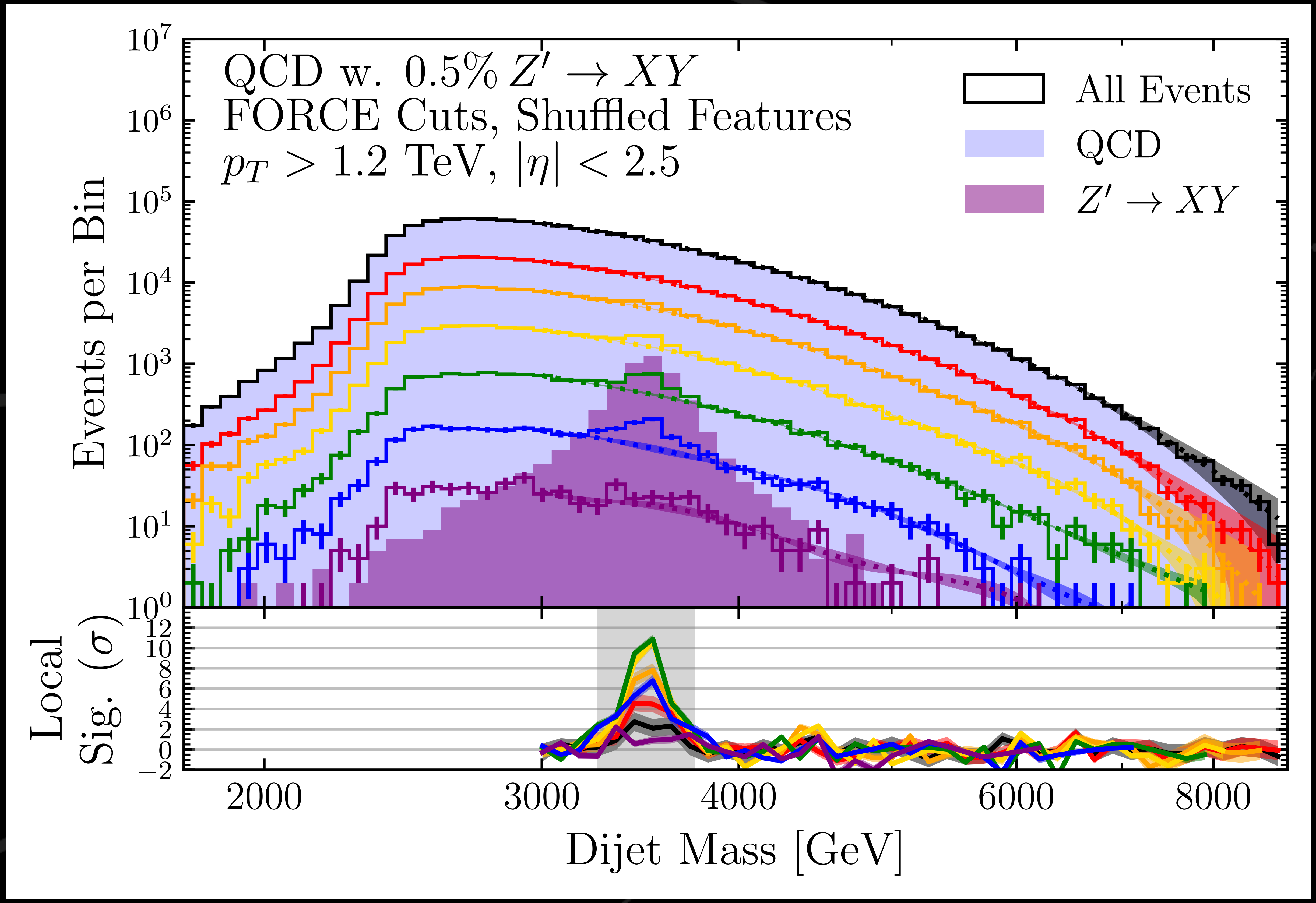
Random classifier in low signal limit

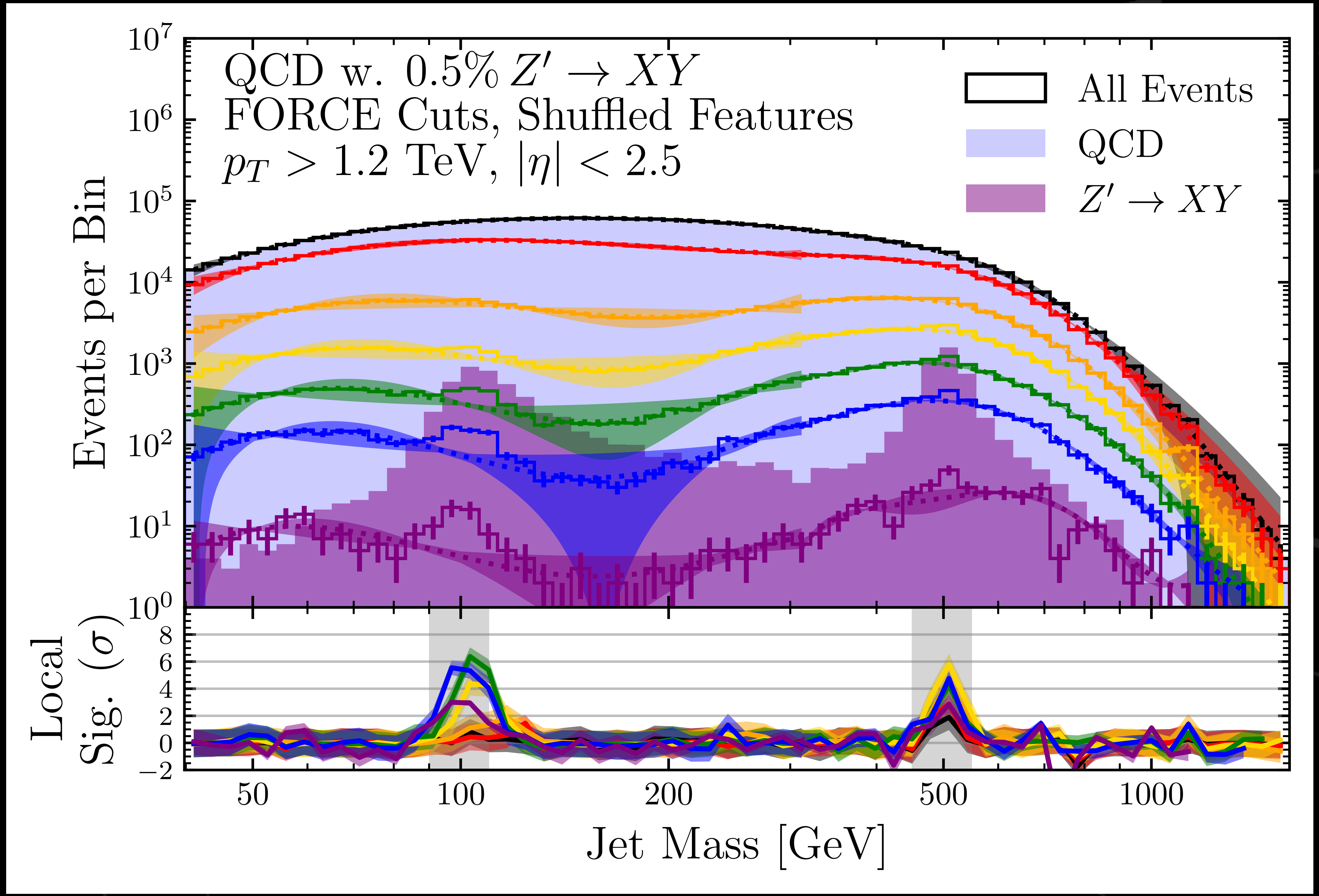
How does shuffling affect the bump hunt?





How does shuffling affect the bump hunt?





We see comparable results to non-shuffled features, motivating original feature set

Come to the dark side: Future Work

Come to the dark side: Future Work

Come to the dark side: Future Work

Integrate with existing anomaly detection methods

Come to the dark side: Future Work

Integrate with existing anomaly detection methods

Generalize to more than 1 kinematic feature and more than 2 event categories

Come to the dark side: Future Work

Integrate with existing anomaly detection methods

Generalize to more than 1 kinematic feature and more than 2 event categories

Make method more sensitive to small signal fractions

Come to the dark side: Future Work

Integrate with existing anomaly detection methods

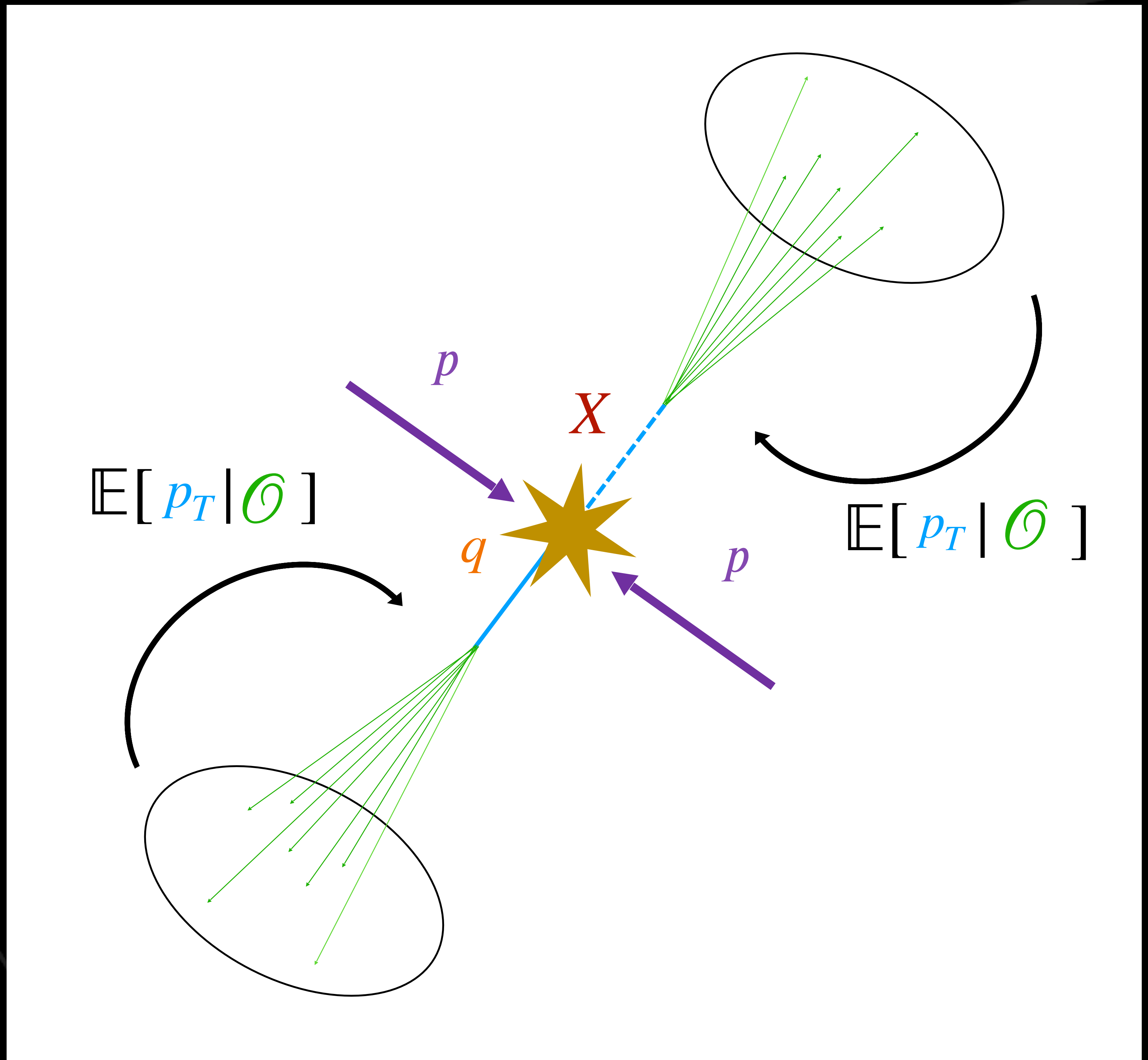
Generalize to more than 1 kinematic feature and more than 2 event categories

Make method more sensitive to small signal fractions

Interrogate conditional expectation to recover f_S, $\langle p_T \rangle_B$, $\langle p_T \rangle_S$, and $L_{S/B}(\mathcal{O})$

Conclusion

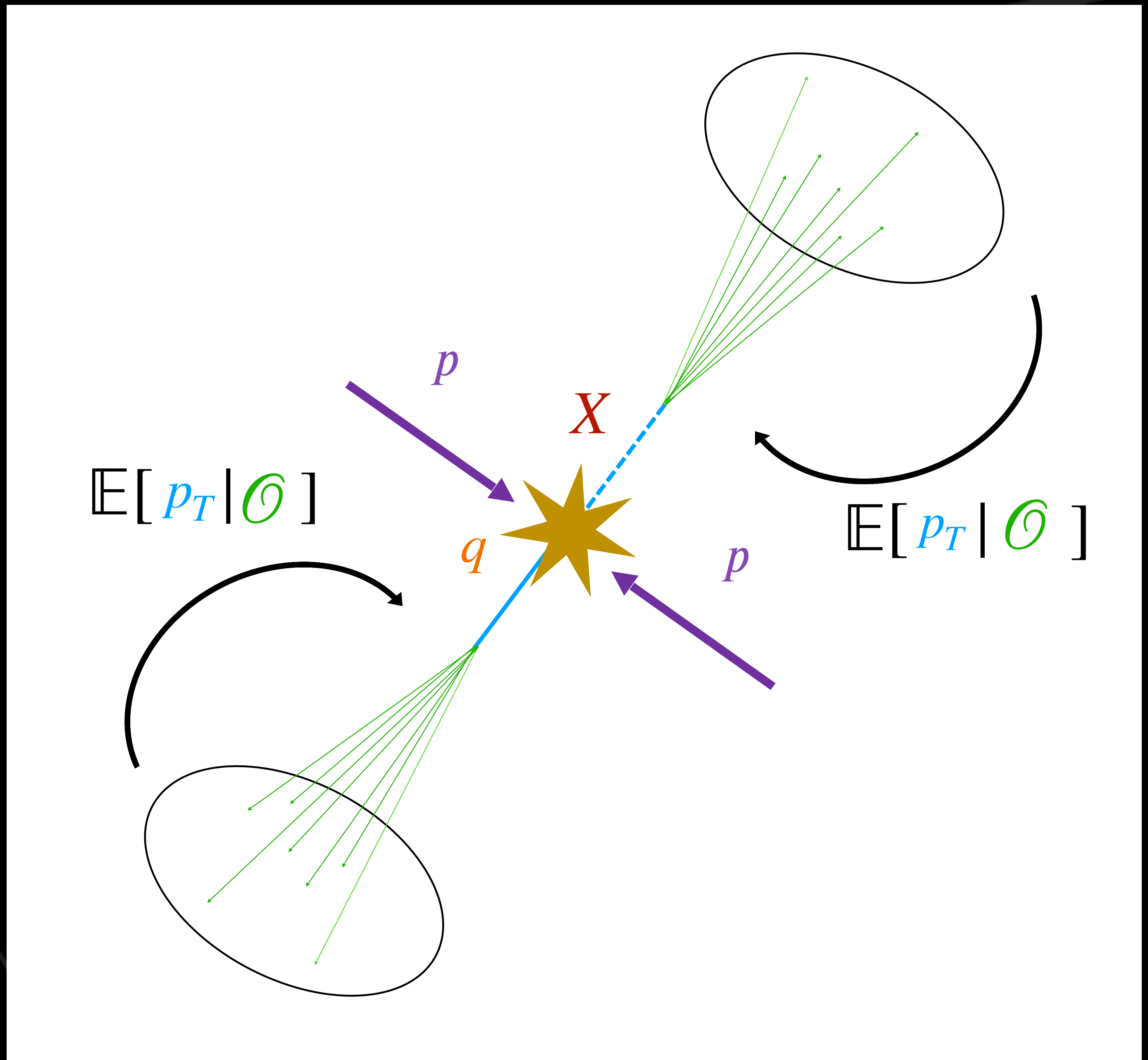
Key Takeaways:



Conclusion

Key Takeaways:

Train ML model to predict kinematics from substructure $\Longrightarrow$ powerful classifier

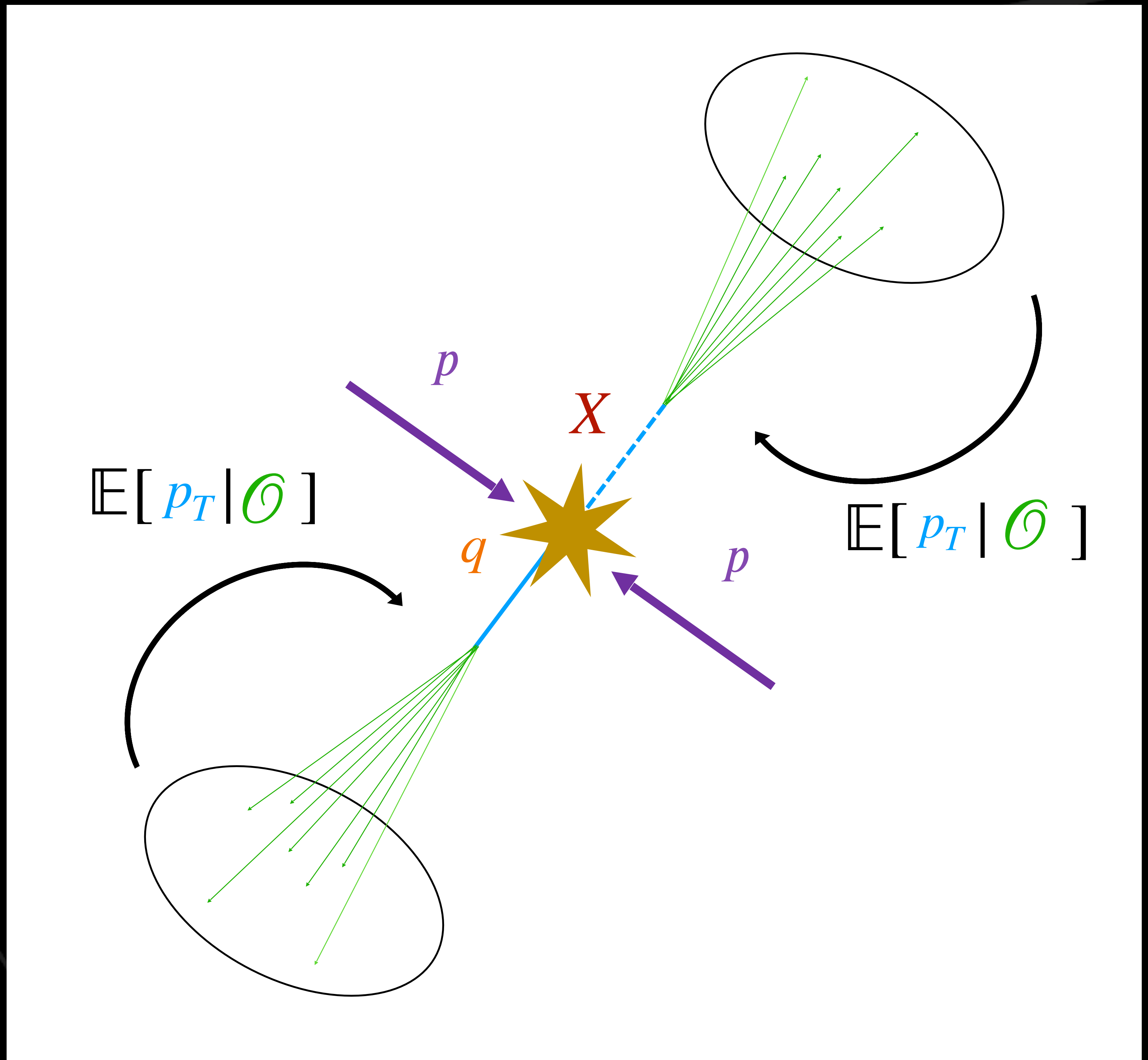


Conclusion

Key Takeaways:

Train ML model to predict kinematics from substructure $\Longrightarrow$ powerful classifier

Shift discussion from specific models to factorized structure



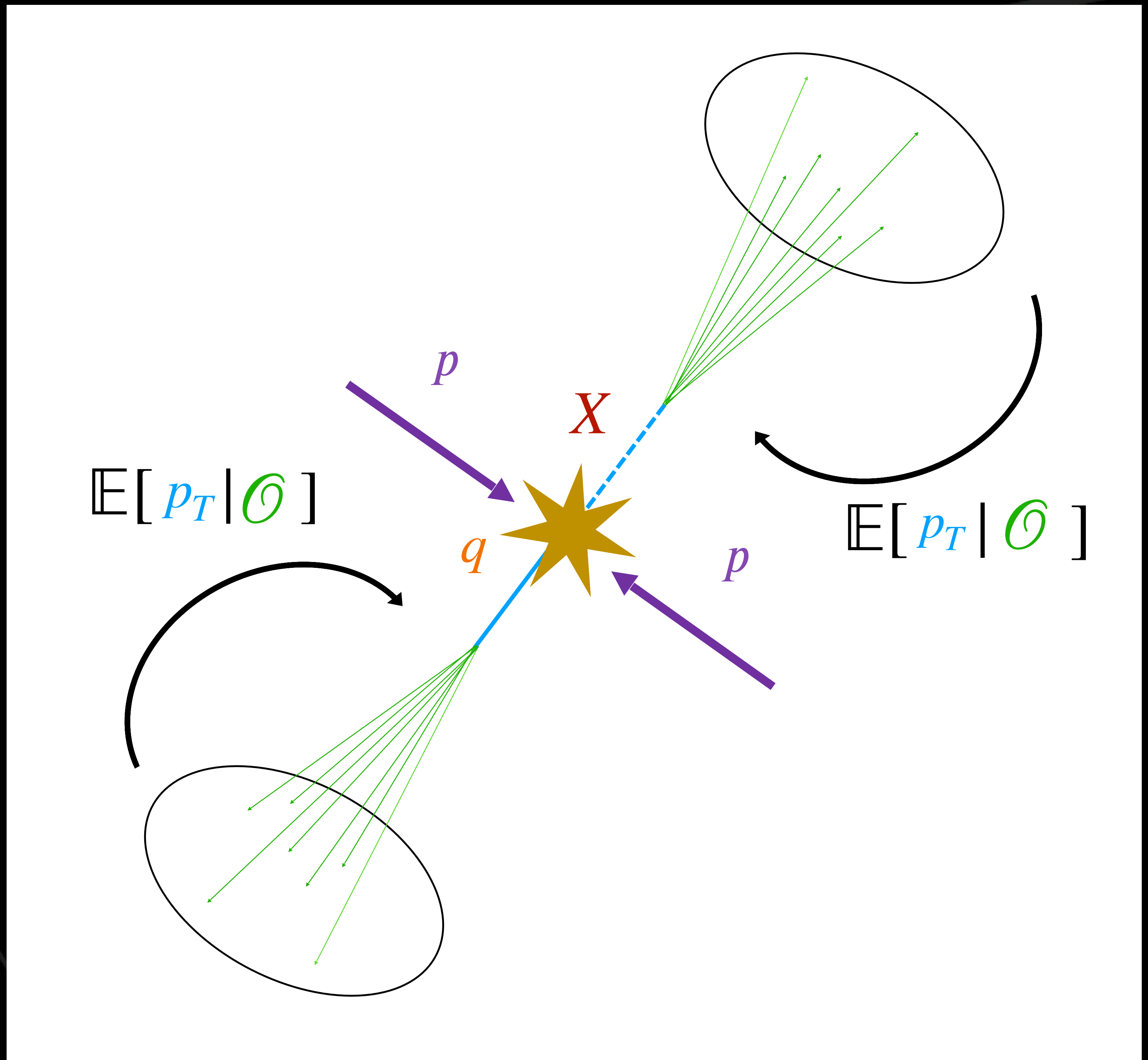
Conclusion

Key Takeaways:

Train ML model to predict kinematics from substructure $\Longrightarrow$ powerful classifier

Shift discussion from specific models to factorized structure

Focused on jets, but works for any factorized objects



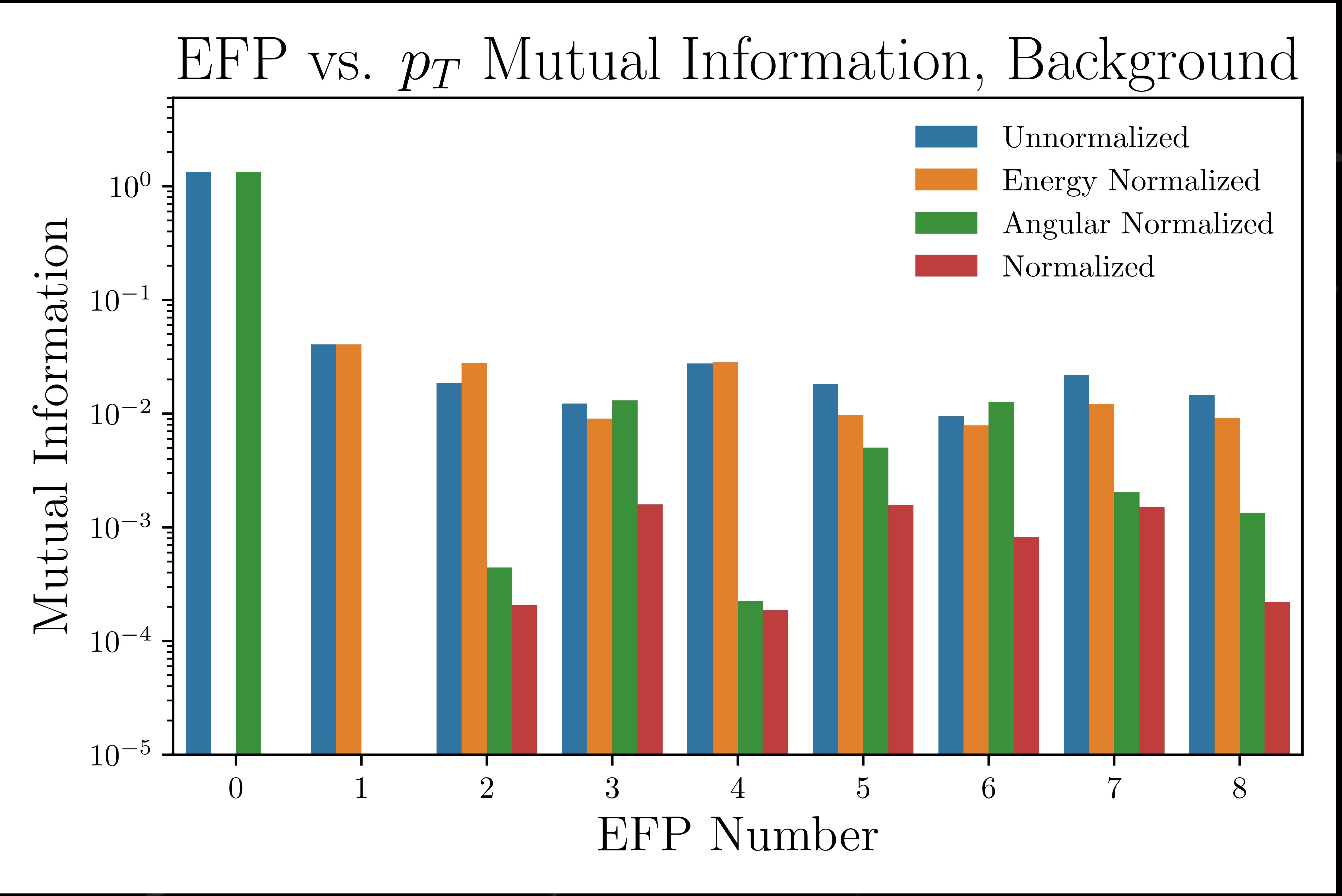
Thank you!

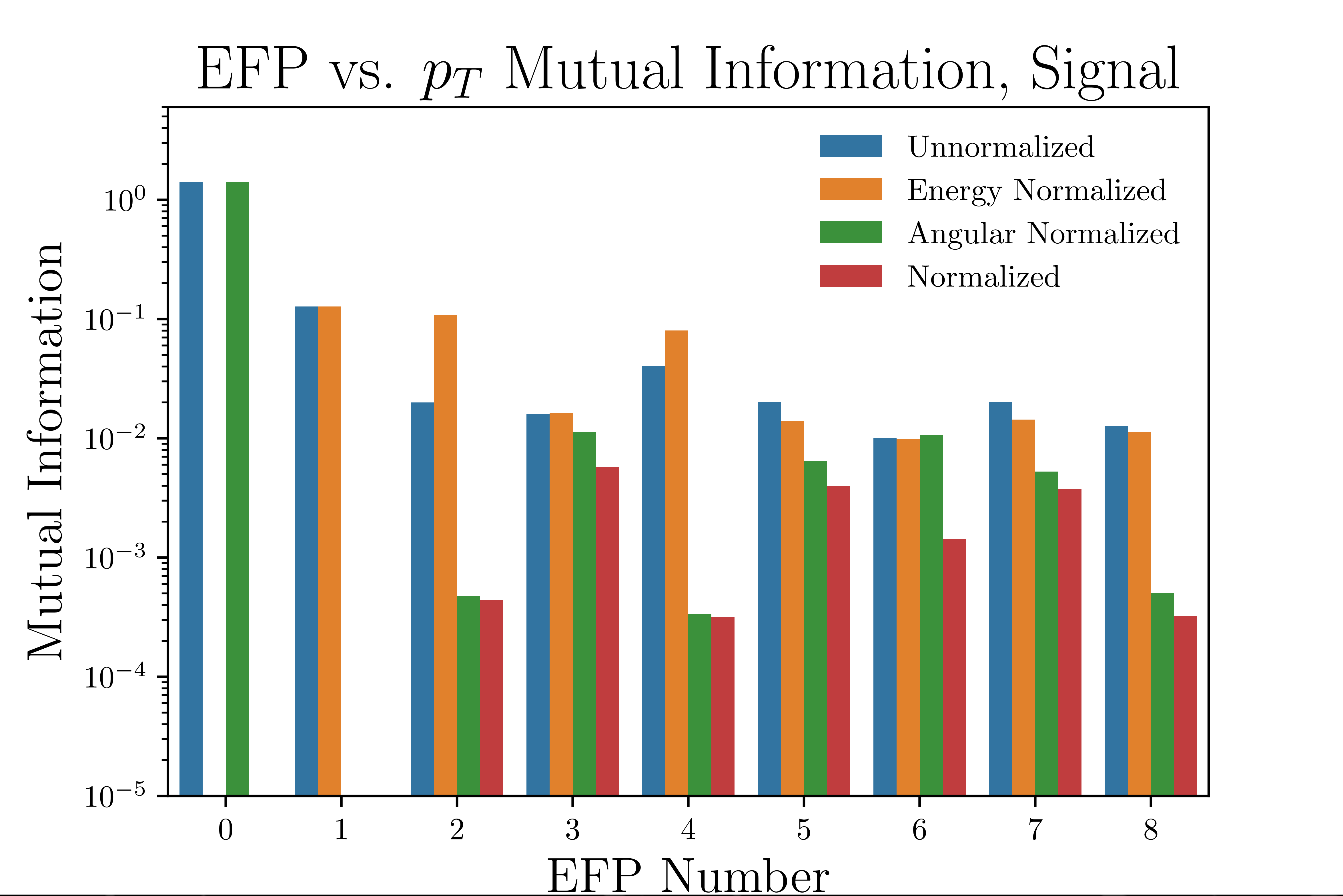
Thank you!

Backup Slides

Interrogating the Normalization

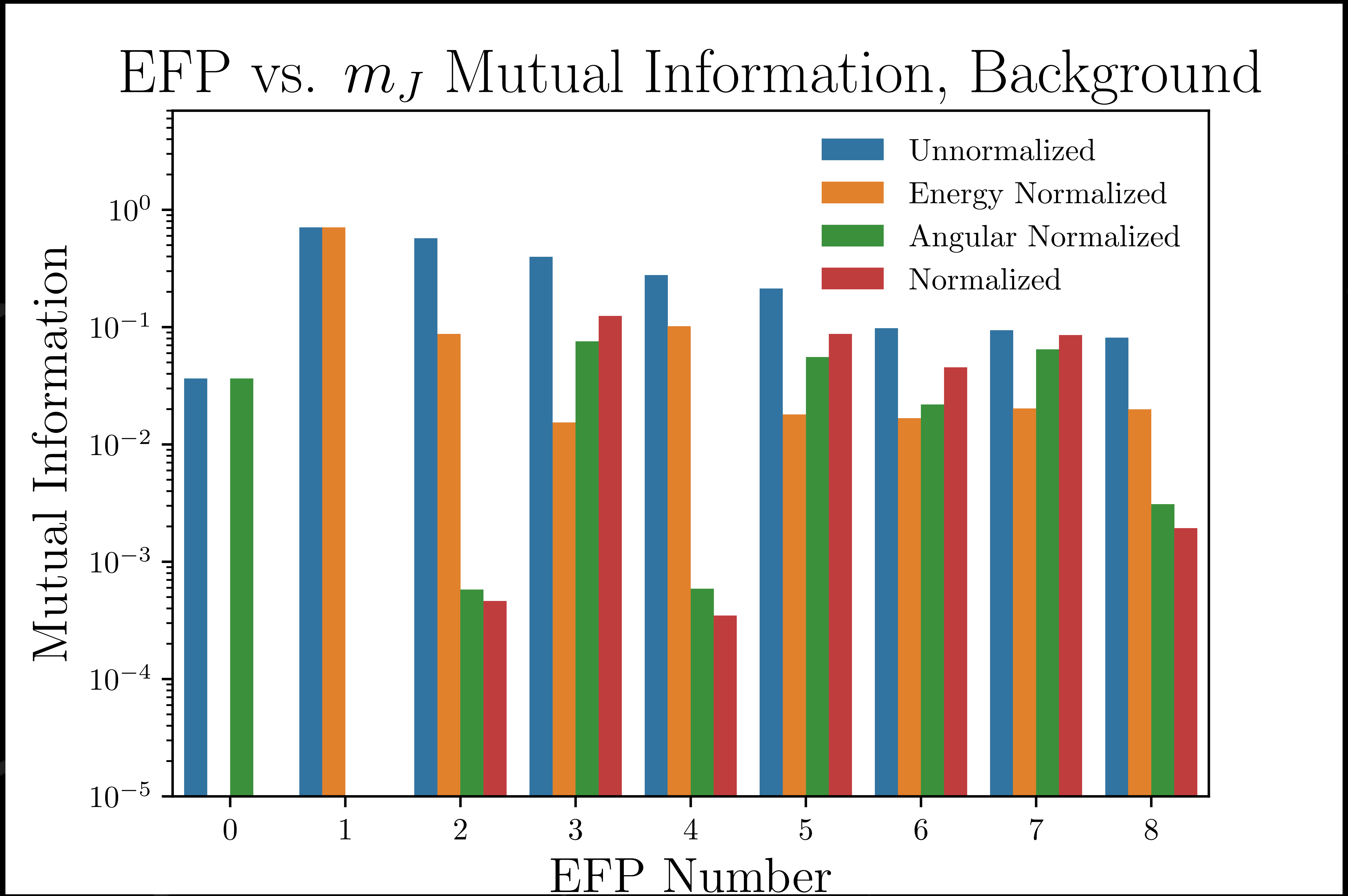
EFPs vs p_T Mutual Information

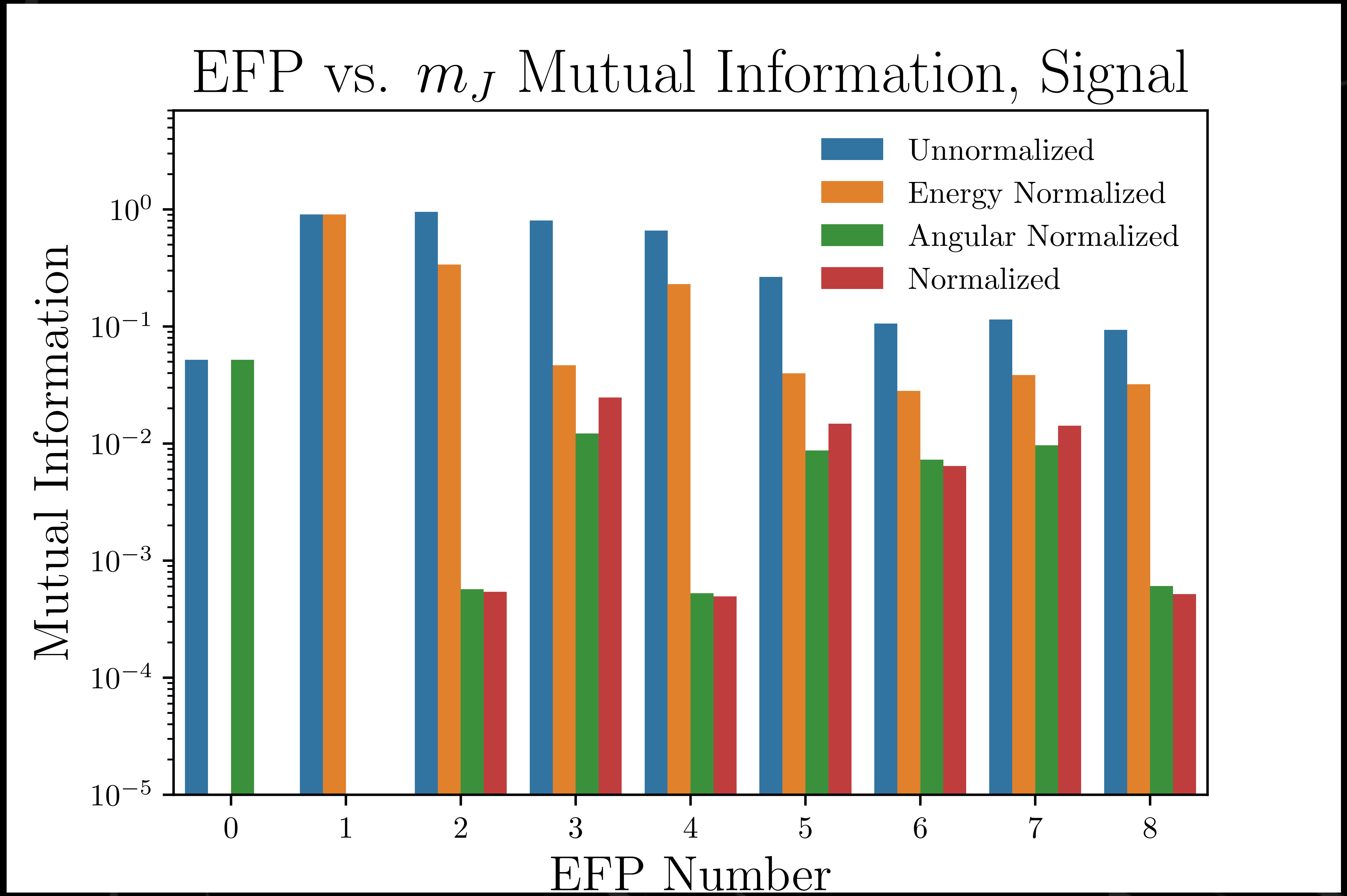




Interrogating the Normalization

EFPs vs m_J Mutual Information





Bump Hunt w/ $f_S = 0$