# Uncovering latent (top) jet substructure

Jernej F. Kamenik
mostly based on 1904.04200, 2005.12319, 2009.xxxxx
with D. A. Faroughy & B. M. Dillon, M. Szewc,
A. Smolkovic, B. Bortolato, A. Matevc

**Institut**
**”Jožef Stefan”**
**Ljubljana, Slovenija**

Univerza v Ljubljani
Fakulteta za matematiko in fiziko

Zoom
17/09/2020

# Introduction

Recent explosion of ML tools in particle physics

⇒ Common goal to classify among different (un)known physical processes ⟺ to learn/approximate likelihoods

Crucial to understand physics learned by the machine

⇒ Helps to understand systematics & validate assumptions (i.e. MC, control region dependence)

Top jets well defined (and understood?) test case

# Supervised ML
# a.k.a. Universal Function Approximation

---

*Input:* representation of model $p(x)$, finite number of examples $\{x_i\}$ sampled/computed/generated from $p$

*Output:* mapping $f(x \to z)$ minimizing a loss function $\mathcal{L}(f, \{x_i\})$

see e.g. B. Nachman, 1909.03081

*Common example:* model of two distributions $p_s(x)\,,\ s = 0\,,\ 1$ with loss function $\mathcal{L} = -[s_i \log(z_i) + (1 - s_i) \log(1 - z_i)]$

(cross-entropy)

$\Rightarrow$ $f$ will approximate $f(x) \sim \dfrac{p_0(x)/p_1(x)}{1 + p_0(x)/p_1(x)}$ (likelihood ratio)

What is the physics contained in $f(x)$? How is it sensitive to biases & systematics of $\{x_i\}$?

*What has the machine learned?*

# Bayesian generative models

$f(x)$ can be thought of as mapping (compression) from target space $X \ni x$ to latent space $Z \ni z$, i.e. between distributions $p(x) \nrightarrow p(z|x)$

$\Rightarrow$ 'Inverse' of $f$ : $f^{-1}(z \rightarrow x)$ can generate artificial data $\{x'_i\}$ in $X$ by sampling from latent space $Z$

$\Rightarrow$ Can learn $f$, $f^{-1}$ by maximizing $p(\{x_i\}|f, f^{-1}, p')$

(Bayes theorem)

Model assumptions encoded as priors $p'(z)$ in latent space $Z$ - possibility of unsupervised ML

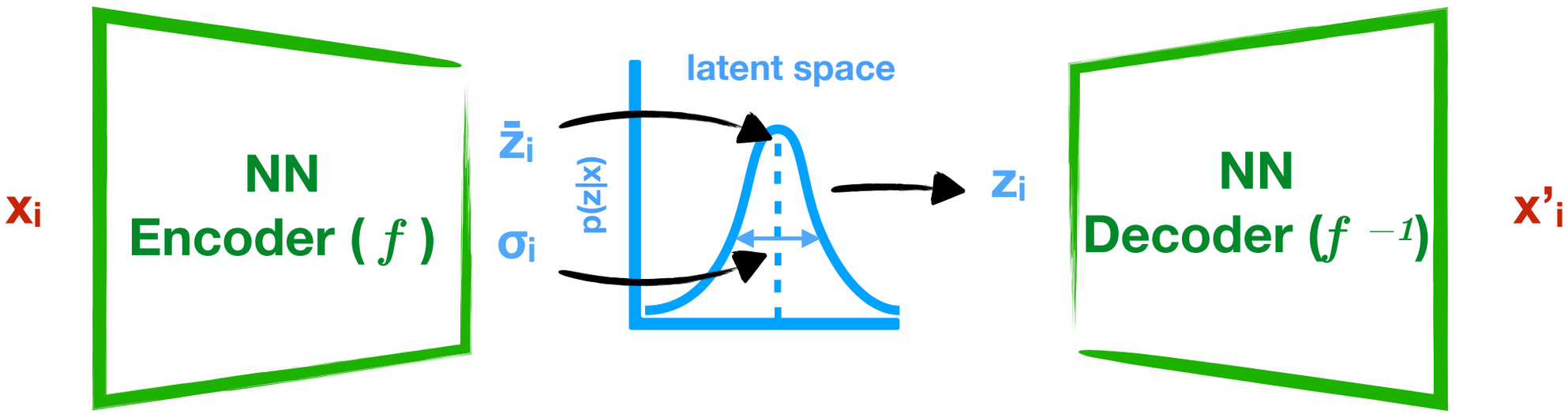
Can infer on physics in $f$ through latent space representation (and use it for classification)

# 1st Example

Classification in latent space with Variational Autoencoders

# VAE architecture

D. P Kingma, M. Welling, 1312.6114




Encoder: mapping $\{x_i\}$ into $p(z|x)$ in latent space

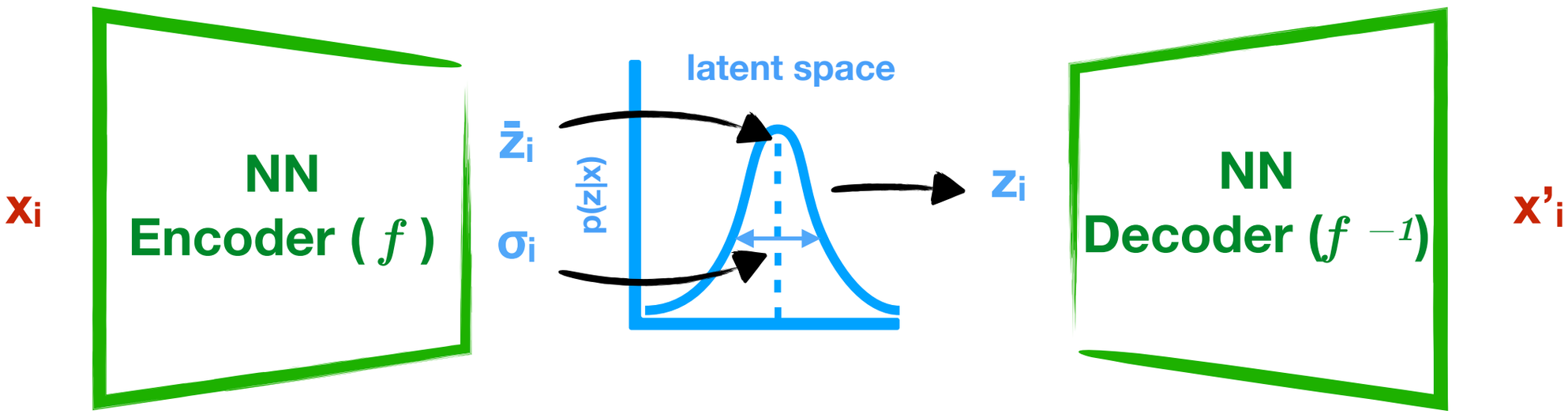
Latent space distribution: Gaussian($\overline{z}_i, \sigma_i$) with normal prior $p'$

Decoder: generative model (likelihood approximator)

Classifier: posterior distribution $p(z|x)$ *in latent space*

# VAE architecture

D. P Kingma, M. Welling, 1312.6114




Loss function $\mathcal{L} = R \cdot \mathrm{MSE}(x_i - x'_i) + D_{KL}(p(z|x)||p'(z)) + r \cdot \log \sigma_i$

$R \gg 1$ avoids over-regularization from KL (~ β-VAE)

Higgins et al., ICLR 2017

reconstruction loss of Decoder

Kullback-Leibler divergence with respect to prior, ensures clustering in Z

r ~ 1 regulates component collapse in Encoder

see e.g. Lucas et al., ICLR 2019

# VAE generative model of (global) jet observables

see also Cheng et al., 2007.01850

⇒ Define several global jet observables, e.g. jet mass ($m_j$), N-subjettiness variables ($\tau_N$), …

J. Thaler & K. Van Tilburg, 1011.2268

⇒ Train VAE on ensemble of jets

→ $p(z|x)$ (latent representation of data)

---

⇒ Scan over the latent space

⇒ Pass these values through the decoder

→ $p(x|z)$ (generative model)

\+ Cut on latent variable ($z$) can also be used to define a classifier

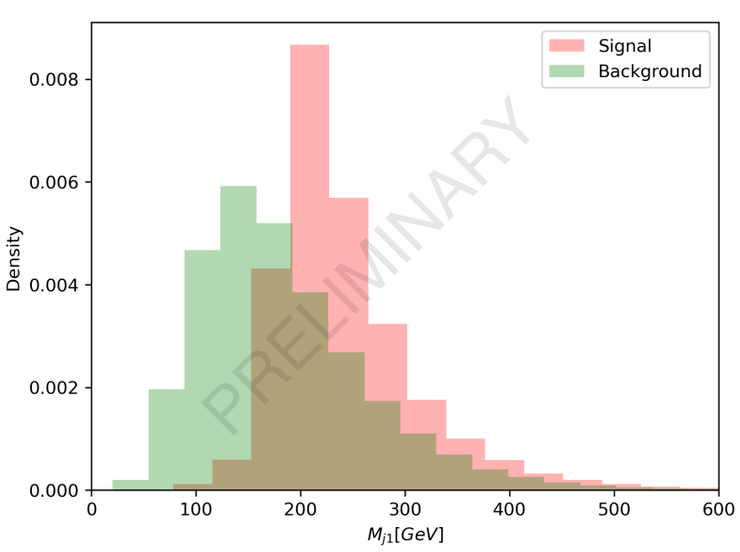
# Example VAE model: top jets

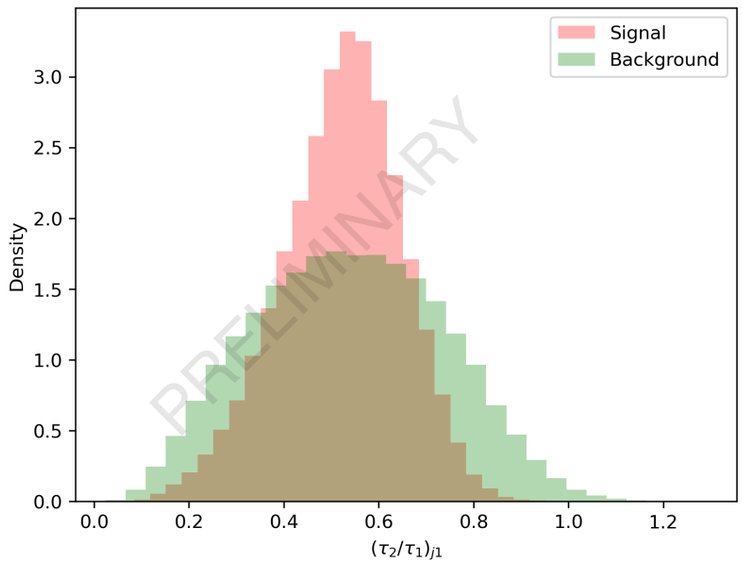
1D latent dimension VAE trained on mixed sample (S+B) using $m_j, (\tau_2/\tau_1)_j, (\tau_3/\tau_2)_j$

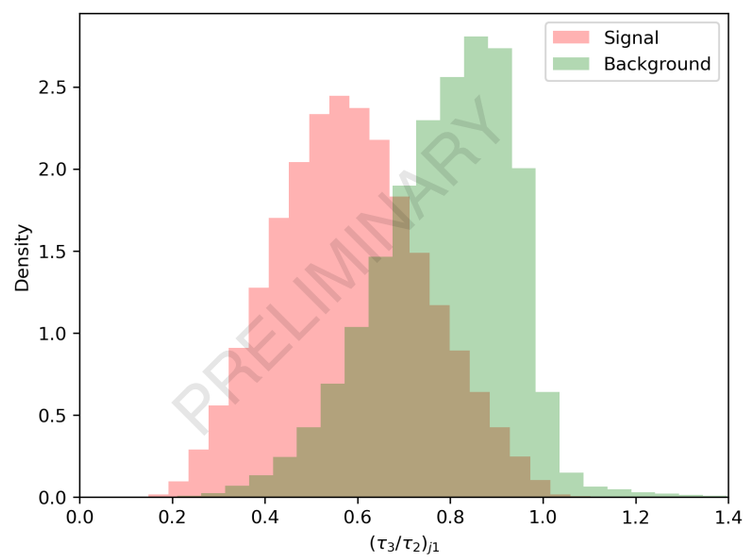
*B*: QCD (light quark & gluon) dijets

*S*: $pp \to t\bar{t} \to W^+W^-b\bar{b}, \quad S/B = 1$

⇒ Jet observable reconstruction: Encoder Input $\{x_i\}$



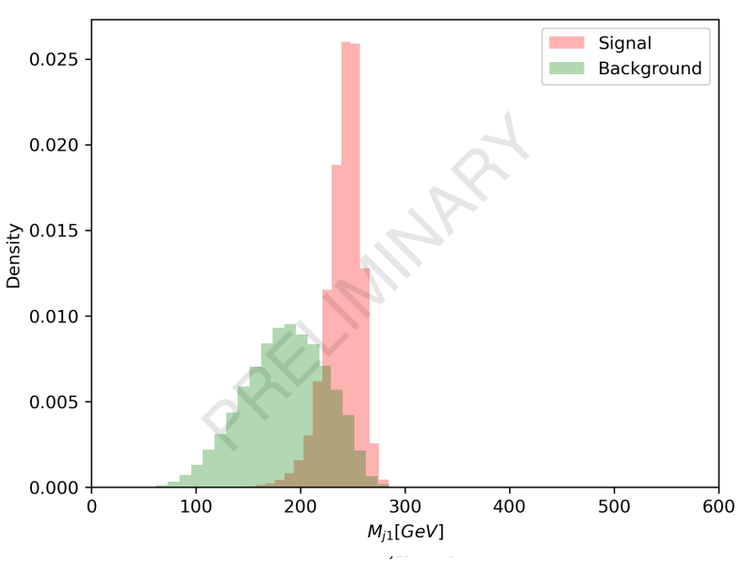
# Example VAE model: top jets

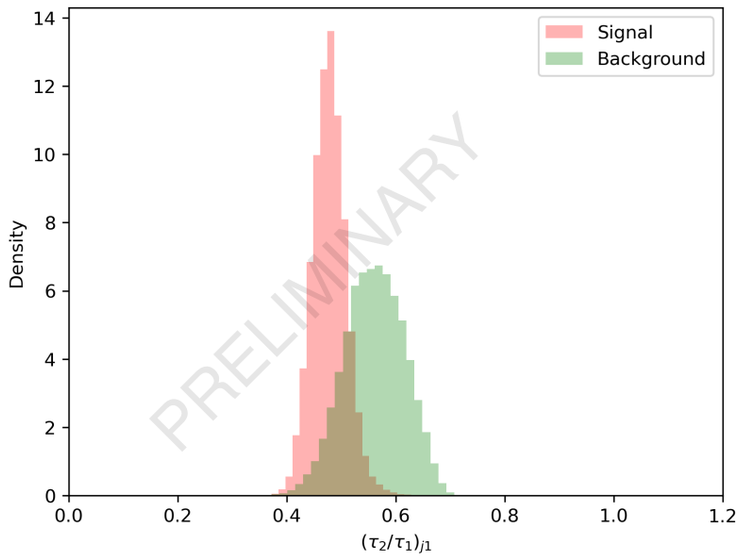
1D latent dimension VAE trained on mixed sample (S+B) using $m_j, (\tau_2/\tau_1)_j, (\tau_3/\tau_2)_j$

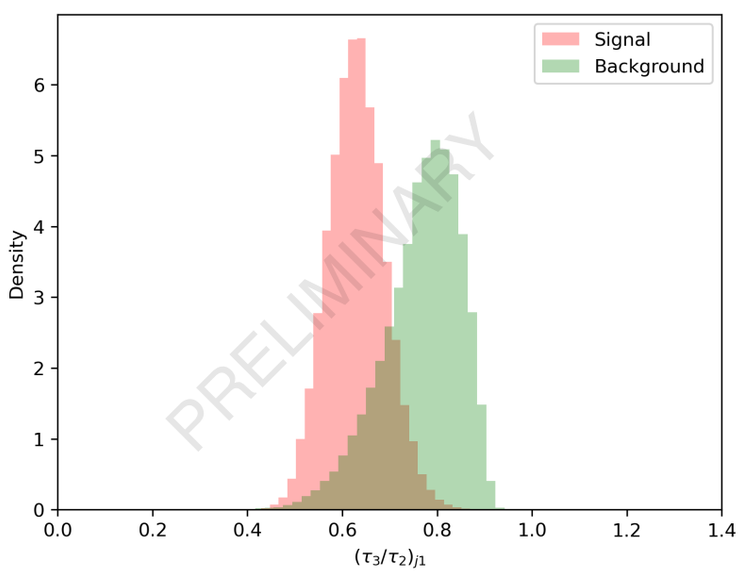
*B*: QCD (light quark & gluon) dijets

*S*: $pp \to t\bar{t} \to W^+W^-b\bar{b}, \quad S/B = 1$

⇒ Jet observable reconstruction: Decoder Output $\{x'_i\}$




(Model with 2 dense layers of 100 nodes for encoder & decoder, with AdaDelta optimization, SeLu activation, trained for 100 epochs)

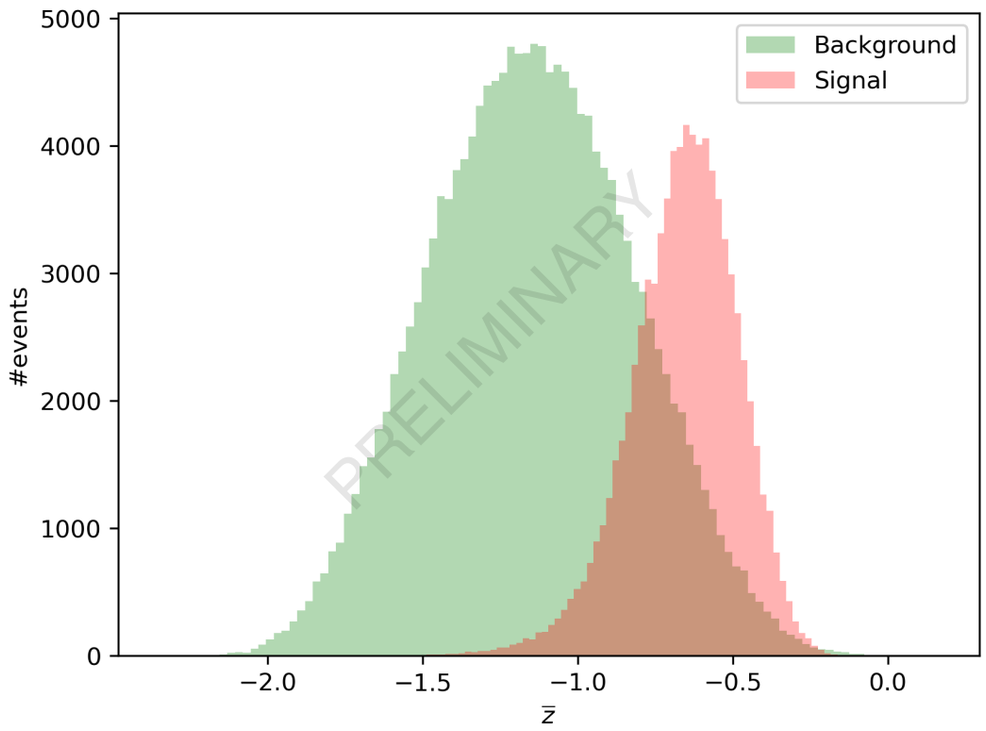
# Example VAE model: top jets

1D latent dimension VAE trained on mixed sample (S+B) using $m_j, (\tau_2/\tau_1)_j, (\tau_3/\tau_2)_j$

*B*: QCD (light quark & gluon) dijets

*S*: $pp \to t\bar{t} \to W^+W^-b\bar{b}, \quad S/B = 1$

⇒ Signal & background clustering in latent space



# Example VAE model: top jets

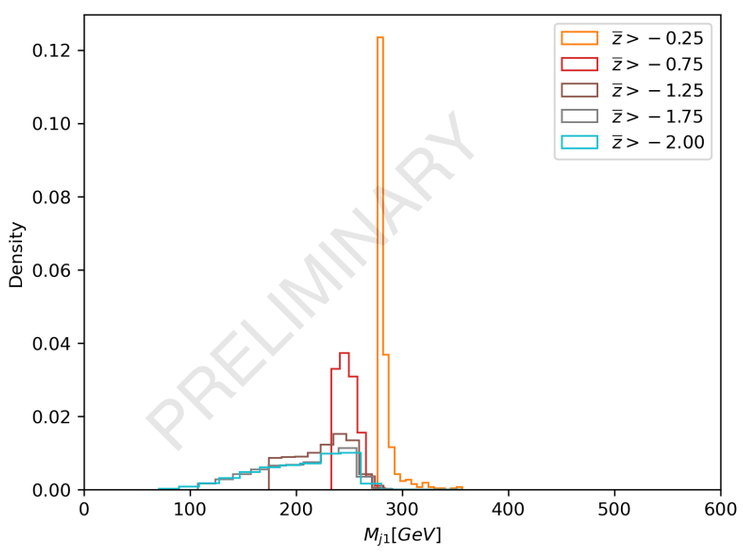
1D latent dimension VAE trained on mixed sample (S+B) using $m_j, (\tau_2/\tau_1)_j, (\tau_3/\tau_2)_j$

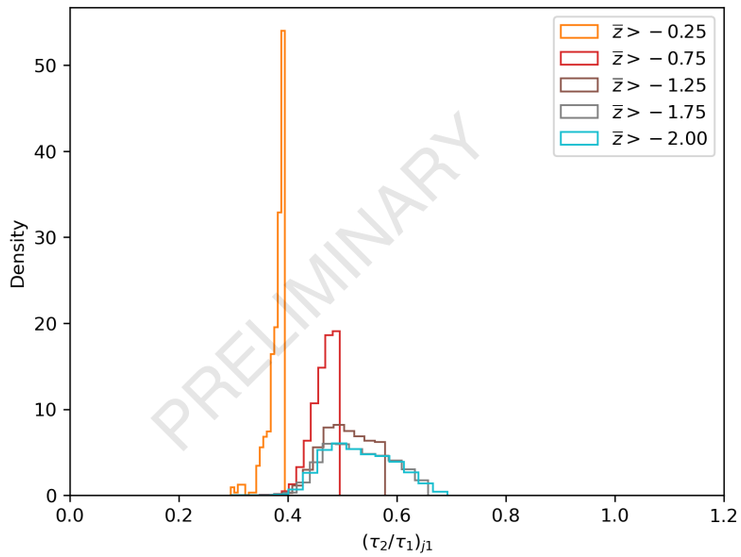
*B*: QCD (light quark & gluon) dijets

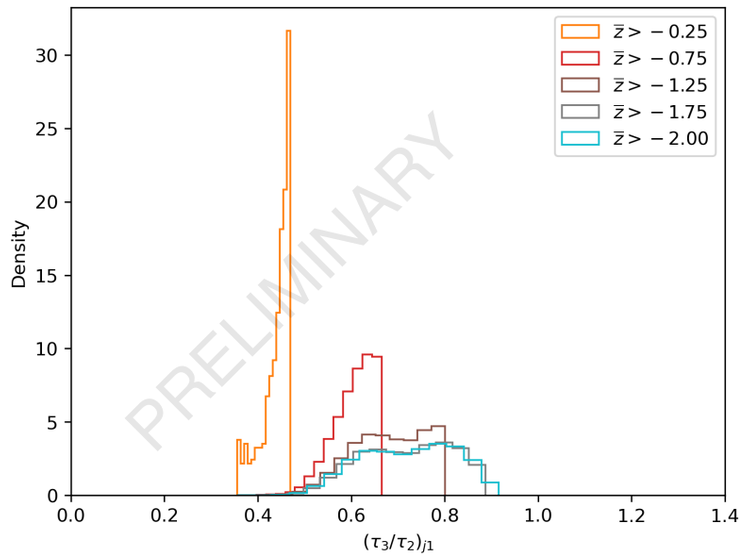
*S*: $pp \to t\bar{t} \to W^+W^-b\bar{b}, \quad S/B = 1$

⇒ Signal & background clustering in latent space

⇒ Physics content of classifier in latent space



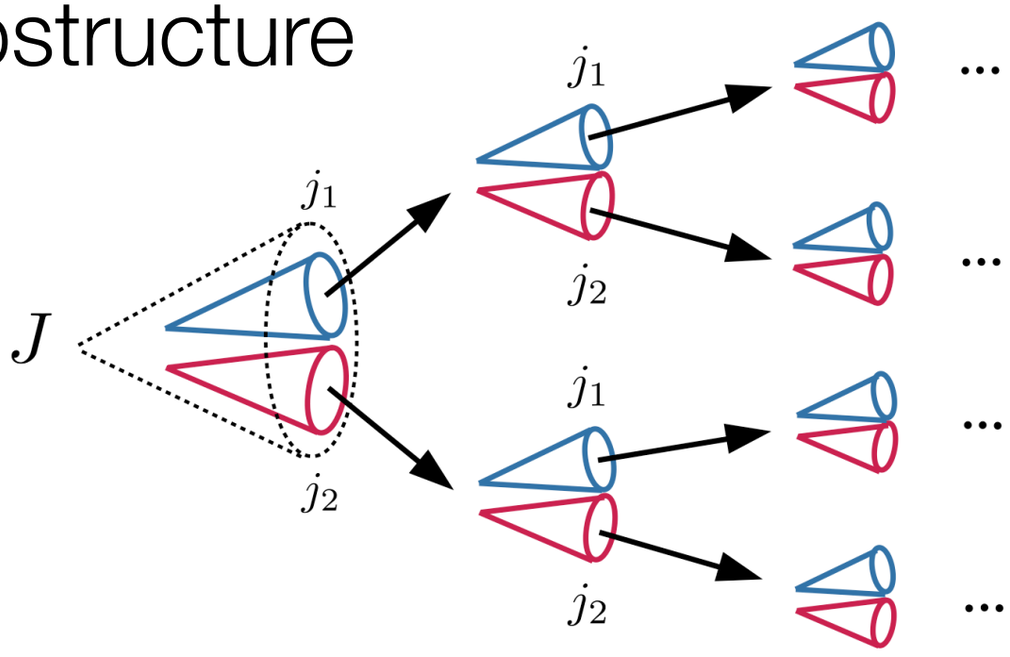



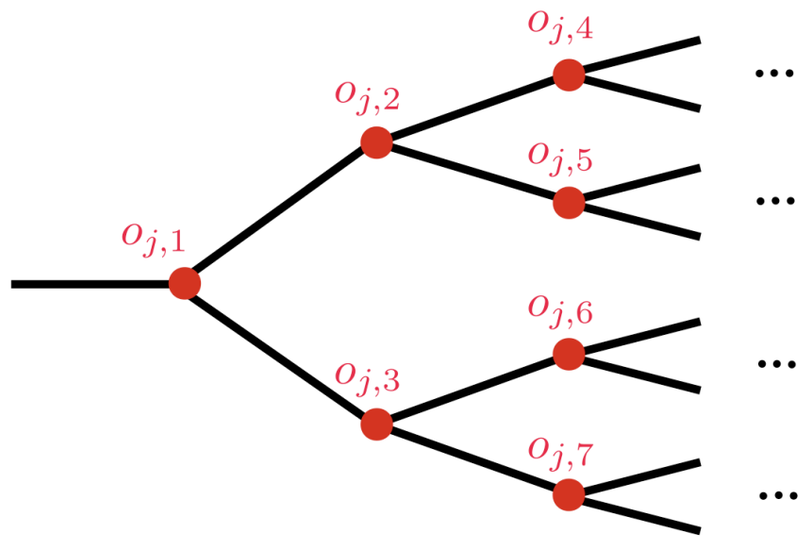
# 2nd Example

Inferring Jet Substructure with Latent Dirichlet Allocation

# Jet substructure representation (beyond global o's)

Represent jets as lists of binned measurements of jet substructure




Each clustering node defines own 'Lund plane' kinematics

⇒ observables $x \sim o_{j,i} =$ bins in space spanned by

Dreyer, Salam & Soyez, 1807.04758

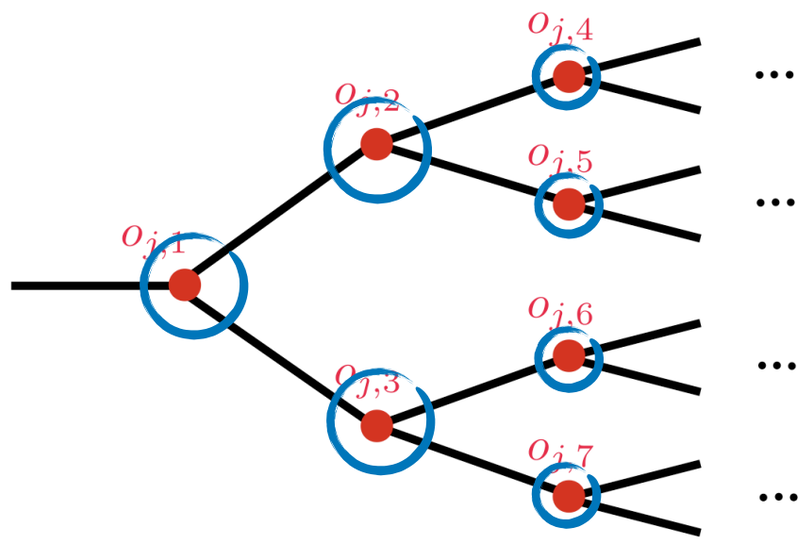
$$\Delta \equiv \Delta R_{ij}, \qquad k_t \equiv p_{tj\Delta}, \qquad m^2 \equiv (p_i + p_j)^2$$

$$z \equiv \frac{p_{tj}}{p_{ti} + p_{tj}}, \qquad \kappa \equiv z\Delta, \qquad \psi \equiv \tan^{-1} \frac{y_j - y_i}{\phi_j - \phi_i}$$

# Simplified generative model of jet (observables)

Assumptions:

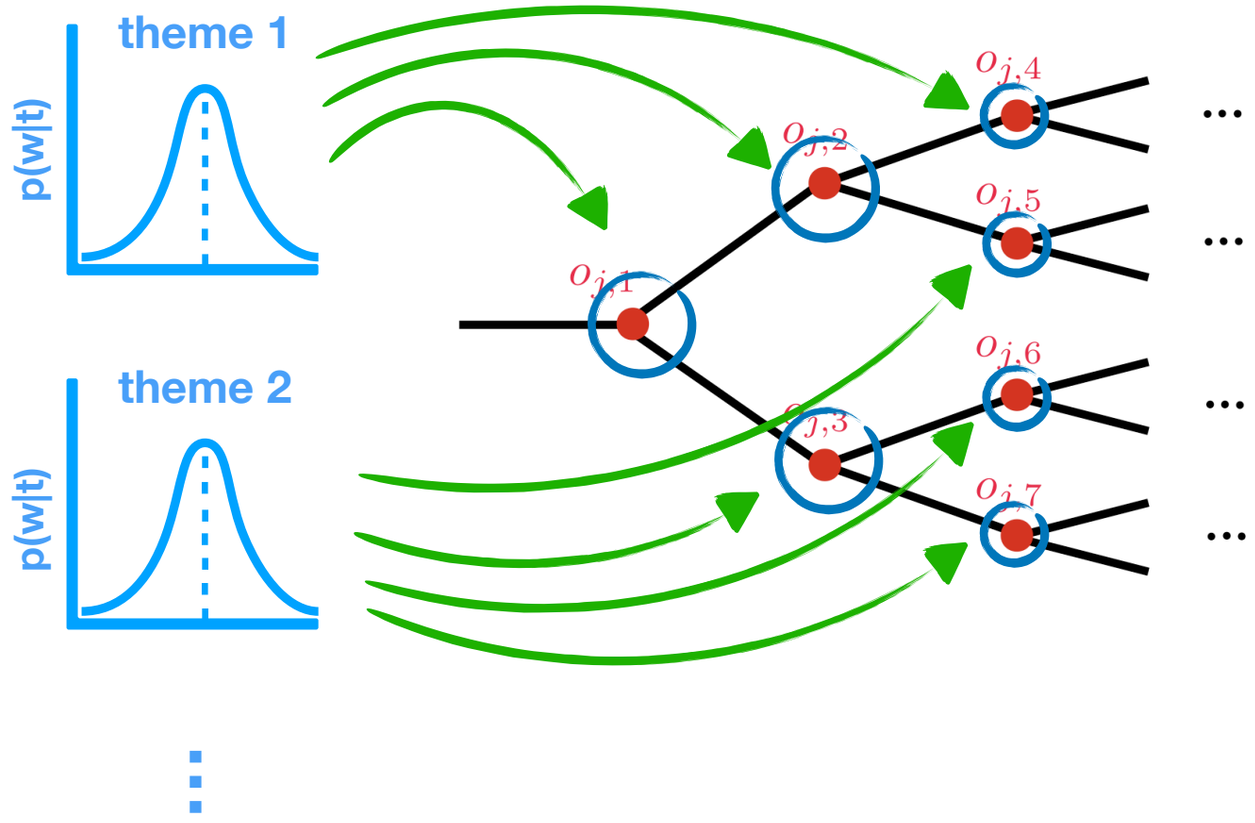
- most useful jet information contained in node observables



# Simplified generative model of jet (observables)

Assumptions:

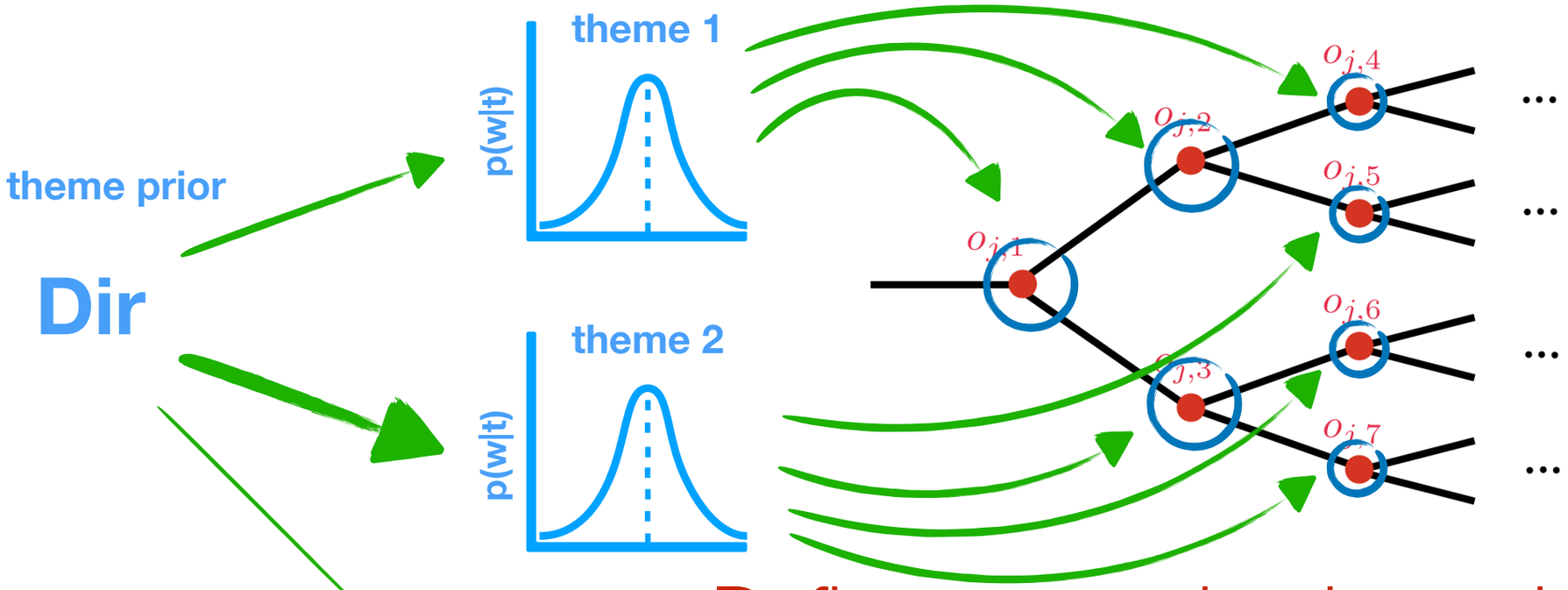
- most useful jet information contained in node observables
- their values are generated by sampling from several underlying ‘latent’ distributions (e.g. QCD splitting, particle decay,…) - *themes*



# Simplified generative model of jet (observables)

Assumptions:

- most useful jet information contained in node observables
- their values are generated by sampling from several underlying ‘latent’ distributions (e.g. QCD splitting, particle decay,…) - *themes*



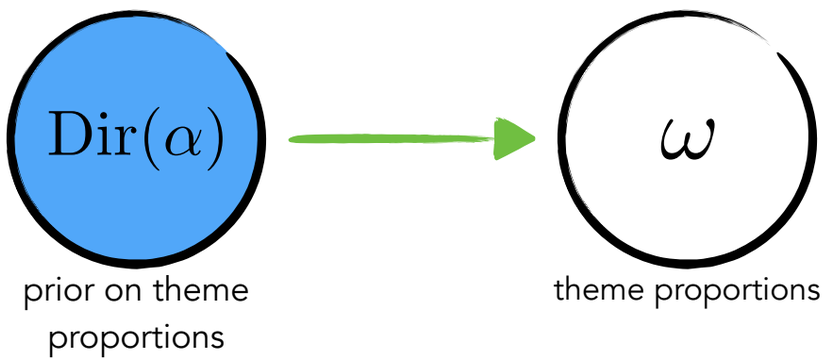

Defines s.c. mixed membership model

Example: Latent Dirichlet Allocation

Blei, Ng, & Jordan (2002)

# Latent Dirichlet Allocation for jet observables

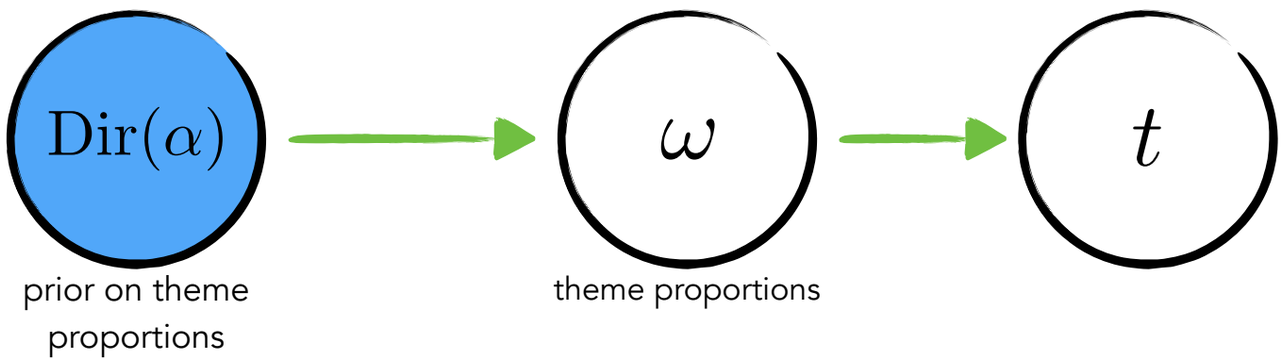
Construct the generative model for jets with *K* themes




Step 1: sample proportions for each theme, a K-dimensional multinomial

# Latent Dirichlet Allocation for jet observables

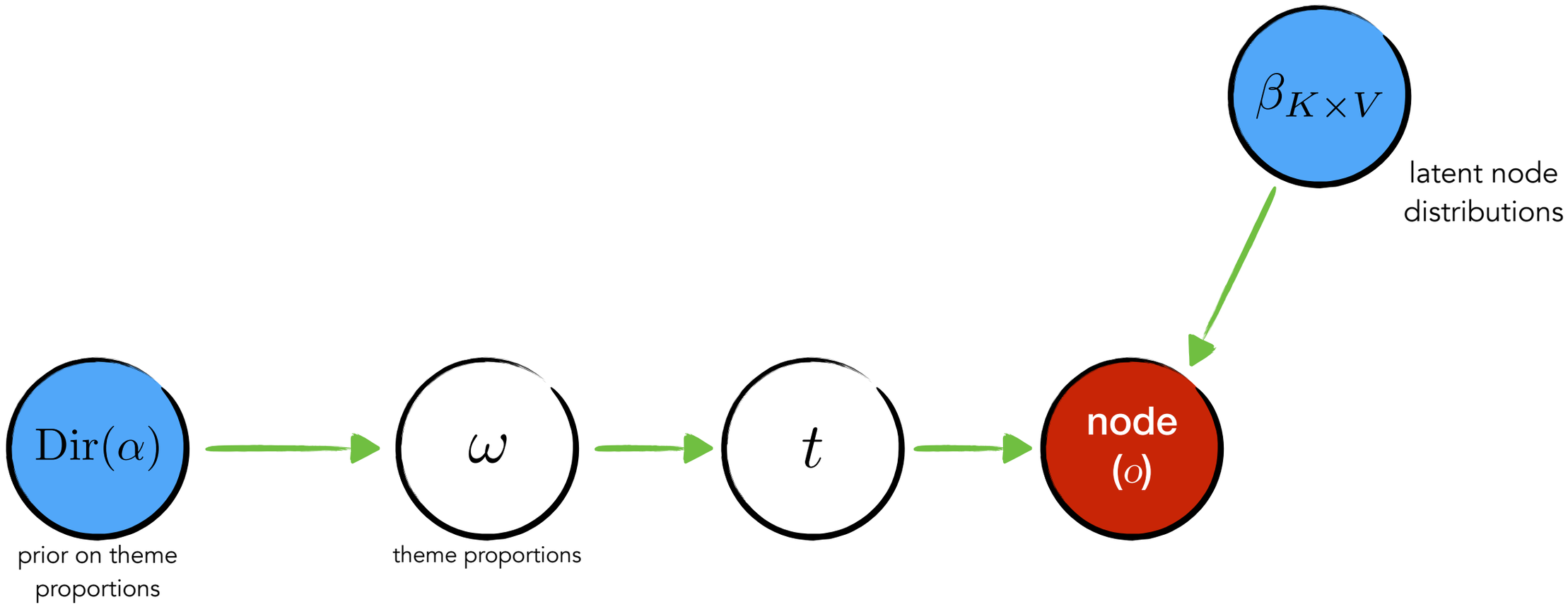
Construct the generative model for jets with K themes




Step 2: sample a single theme from the multinomial

# Latent Dirichlet Allocation for jet observables

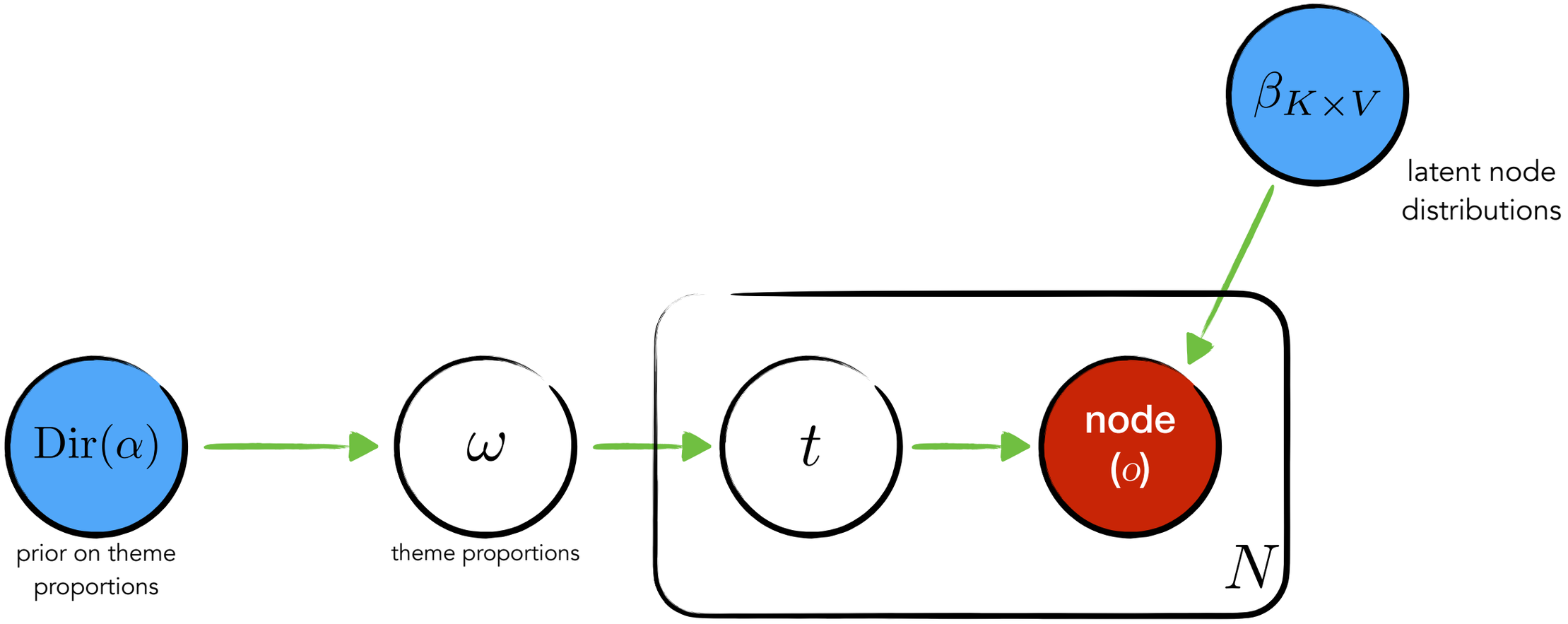
Construct the generative model for jets with K themes




Step 3: sample a node from the appropriate theme distribution

# Latent Dirichlet Allocation for jet observables

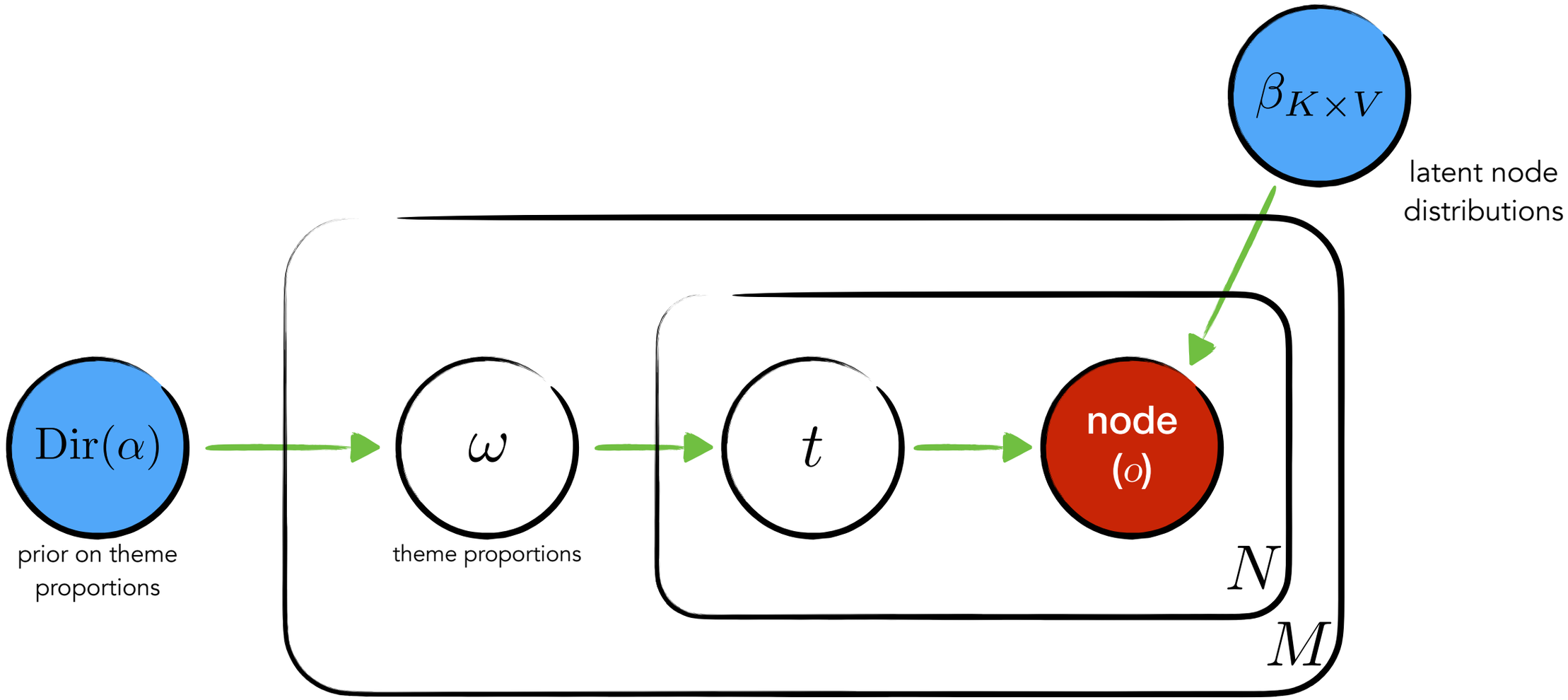
Construct the generative model for jets with K themes




- repeat this for each of the *N* nodes in the jet

# Latent Dirichlet Allocation for jet observables

## Construct the generative model for jets with K themes




- repeat this for each of the *N* nodes in the jet
- repeat again for each of the *M* jets you want to generate

# Latent Dirichlet Allocation for jet observables

Construct the generative model for jets with *K* themes

Define probability to generate a set of node observables ($o_{i,j}$)

$$p(\text{jet}|\alpha,\beta) = \int_w p(w|\alpha) \prod_{o\in\text{jet}} \left( \sum_t p(t|w) p(o|t,\beta) \right)$$

Solve for latent theme distributions ($\beta$) using Bayes theorem & approximate inference

$$\beta^{\text{MLE}}_{K\times V} = \underset{\beta}{\text{argmax}} \;\; \log \left( \prod_{i=1}^{M} p\left(\text{jet}_i|\alpha,\beta\right) \right)$$

Originally constructed for study of genotypes & text topics

Papadimitriou, Raghavan, Tamaki & Vempala (1998)
Hofmann (1999)
Blei, Ng, & Jordan (2002)

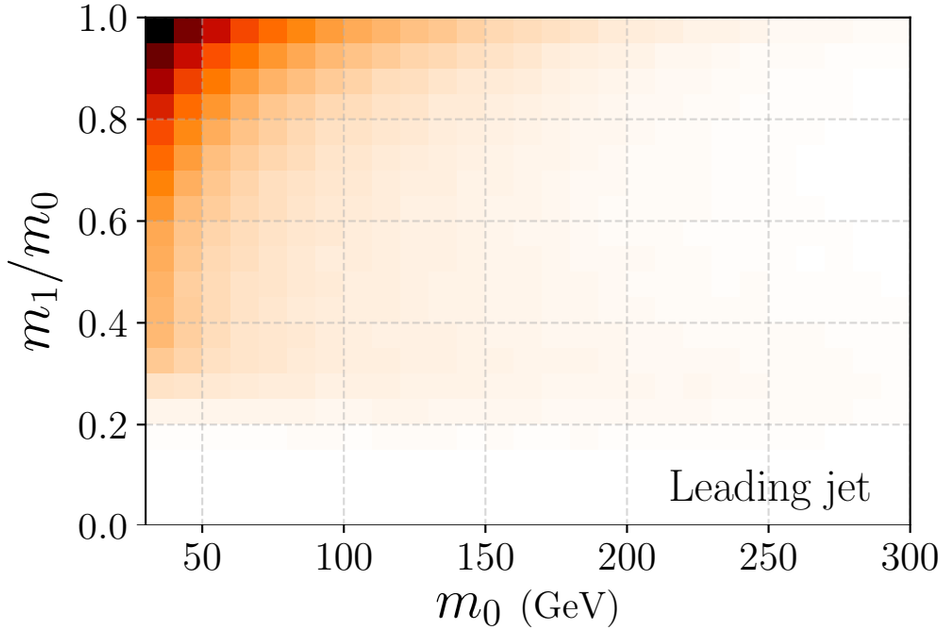
# Example LDA model: top jets

2-theme LDA trained on mixed sample (S+B) using jet mass

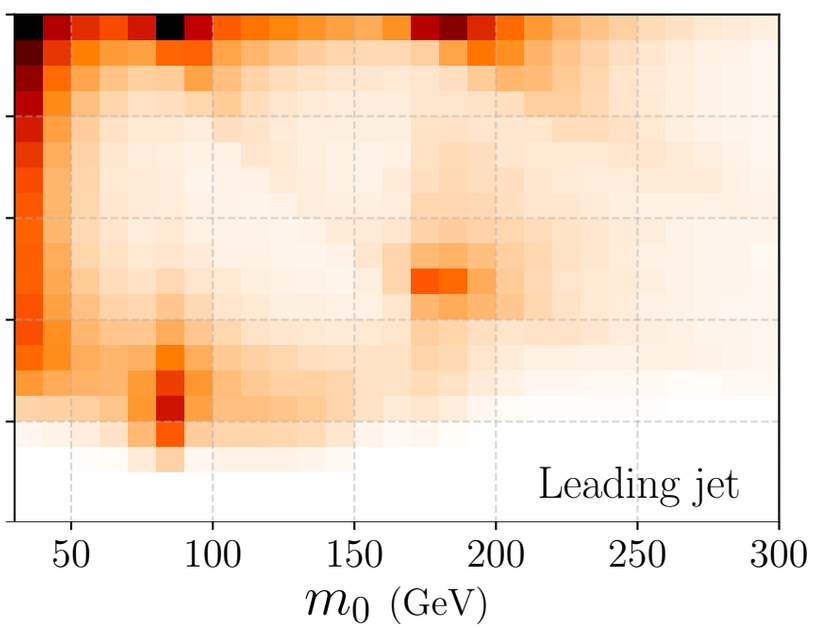
*B*: QCD (light quark & gluon) dijets

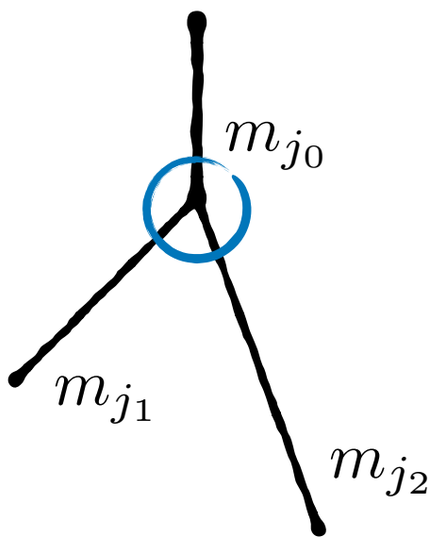
*S*: $pp \to t\bar{t} \to W^+W^-b\bar{b}, \quad S/B = 1$

Background truth




Signal truth



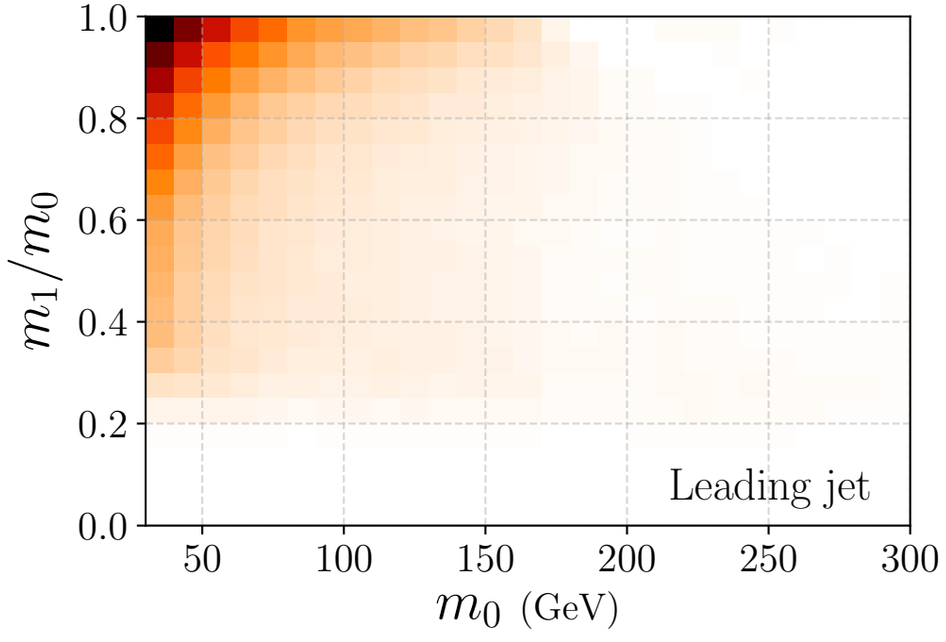
# Example LDA models: top jets

2-theme LDA trained on mixed sample (S+B) using jet mass

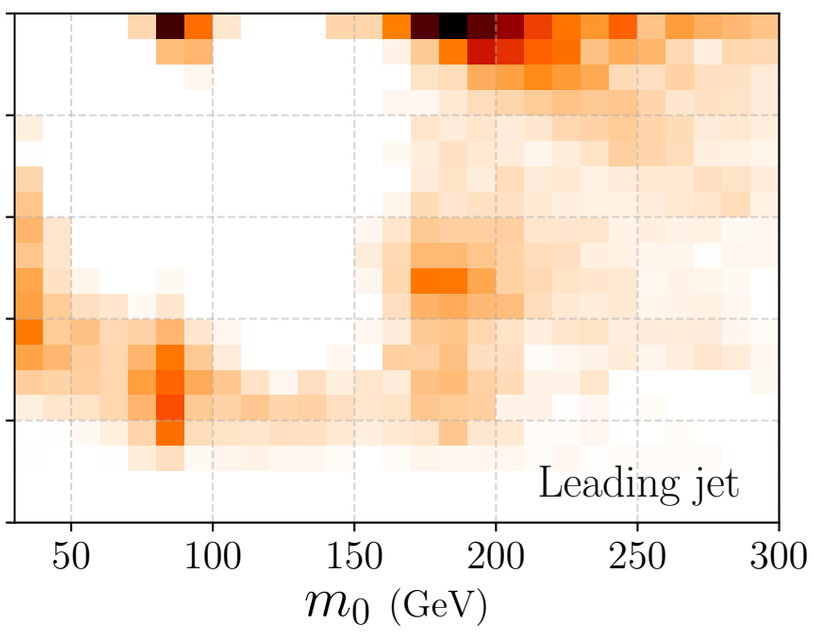
*B*: QCD (light quark & gluon) dijets

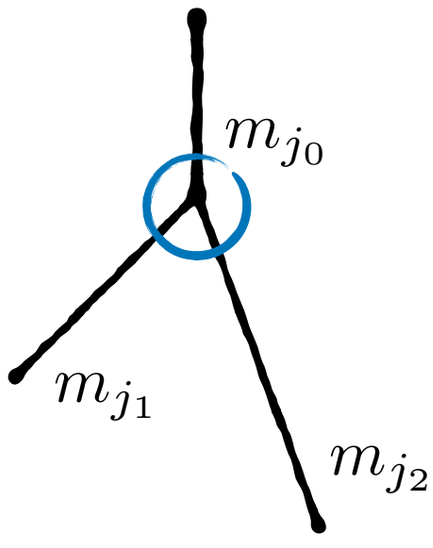
*S*: $pp \to t\bar{t} \to W^+W^-b\bar{b}, \quad S/B = 1$

Theme 1




Theme 2




(Model selected via marginalization over Dirichlet priors using model perplexity as measure.)

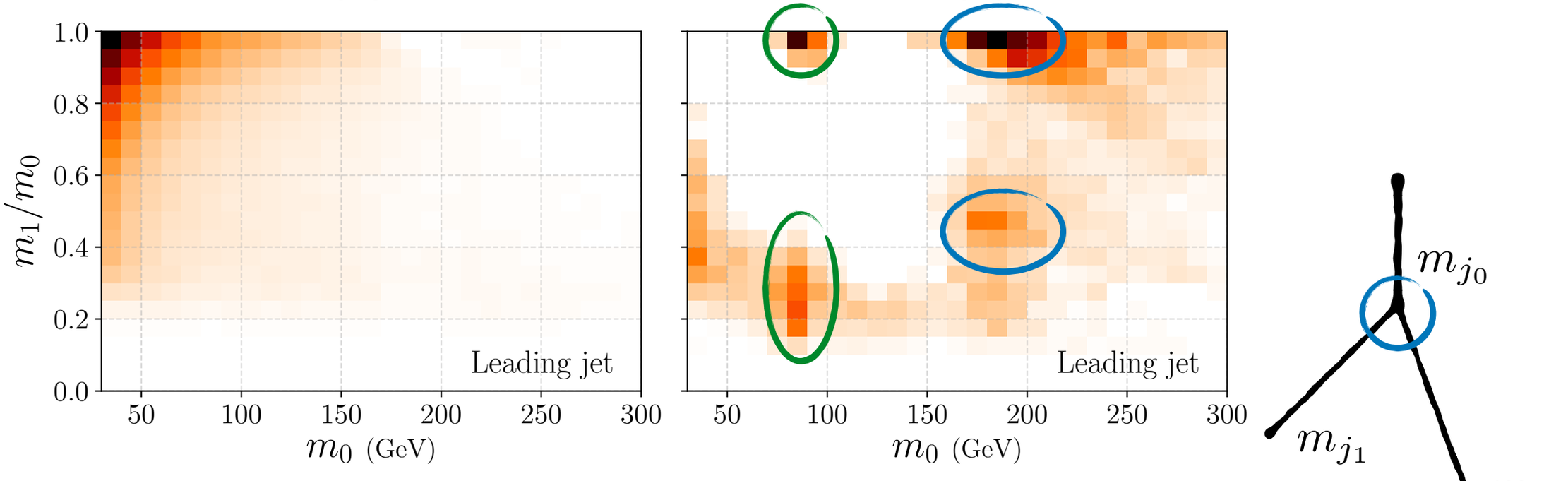
# Example LDA models: top jets

2-theme LDA trained on mixed sample (S+B) using jet mass

*B*: QCD (light quark & gluon) dijets

*S*: $pp \to t\bar{t} \to W^+W^- b\bar{b}, \quad S/B = 1$

Theme 1 Theme 2



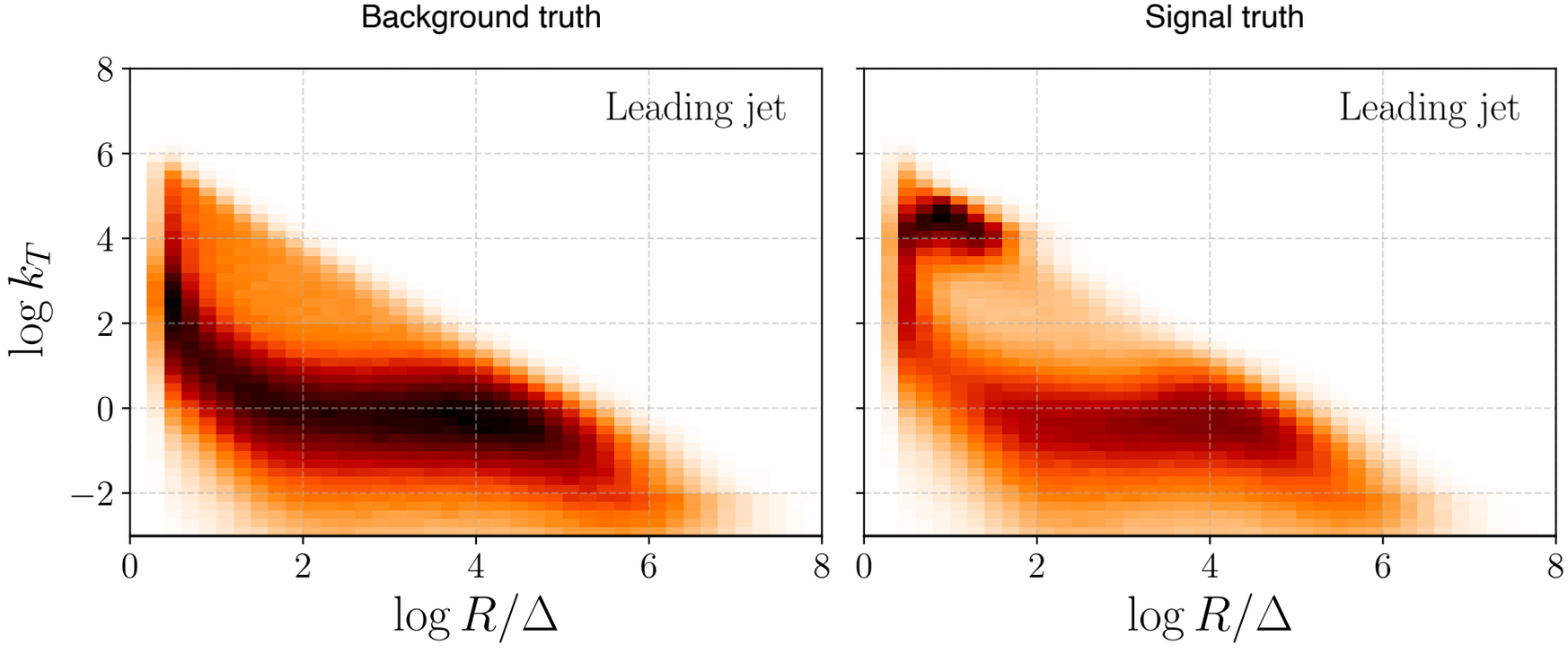

Accurate characterisation of two admixtures (S,B)!

# Example LDA models: top jets

2-theme LDA trained on mixed sample (S+B) in Lund plane

*B*: QCD (light quark & gluon) dijets

*S*: $pp \to t\bar{t} \to W^+W^-b\bar{b}, \quad S/B = 1$

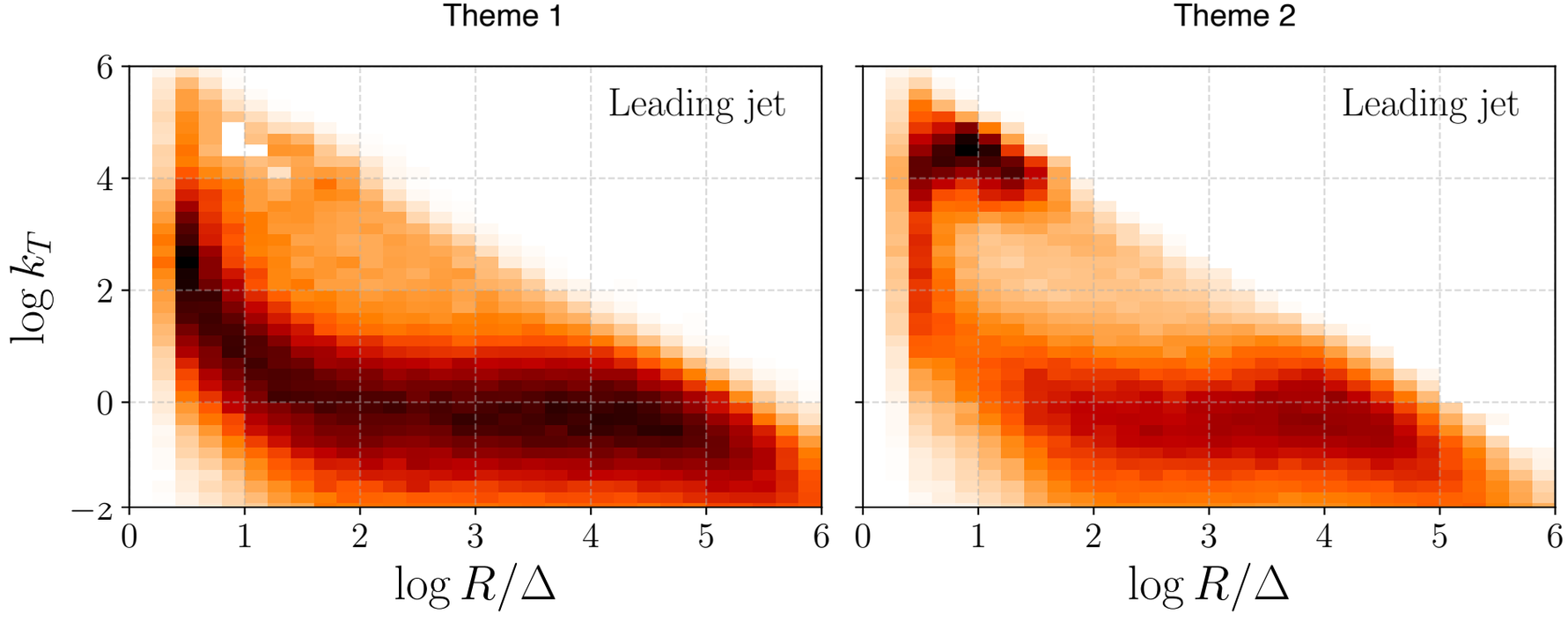


# Example LDA models: top jets

2-theme LDA trained on mixed sample (S+B) in Lund plane

*B*: QCD (light quark & gluon) dijets

*S*: $pp \to t\bar{t} \to W^+W^-b\bar{b}, \quad S/B = 1$




Latent (model) representation directly in terms of physical observables!

# Conclusions

In ML approaches to particle physics event classification imperative to understand *What has the machine learned?*

*Latent representations* of *generative models* promising tool

Presented two top-jet based examples:

LDA : continuous mixture of finite No. of latent representations directly in target space of observables

VAE : continuum of representations in latent space, can be projected (decoded) back to space of observables

# Conclusions

In ML approaches to particle physics event classification imperative to understand *What has the machine learned?*

*Latent representations* of *generative models* promising tool

Presented two top-jet based examples (LDA, VAE)

⇒ Both methods allow for unsupervised classification of jets/events based directly on their latent representations.

⇒ Can be generalized to other observables (for VAE also low level), higher dimensional latent spaces…

⇒ Work even for asymmetric S/B mixtures (anomaly detect.) & directly on data (no sidebands, control regions)

# Supplements

# What exactly is the Dirichlet Distribution

Multivariate equivalent of Beta distribution
(e.g. dice factory vs. coin factory)

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}$$

$\alpha_i$ determines prior - mean shape and sparsity

Dirichlet is defined over (k-1) simplex (k non-negative arguments which sum to one)

Dirichlet is conjugate prior to multinomial distribution - posterior is also Dirichlet

In jet LDA, themes are V-dimentional Dirichlet; theme proportions are K-dimensional Dirichlet

# What exactly is the Dirichlet Distribution

Multivariate equivalent of Beta distribution
(e.g. dice factory vs. coin factory)

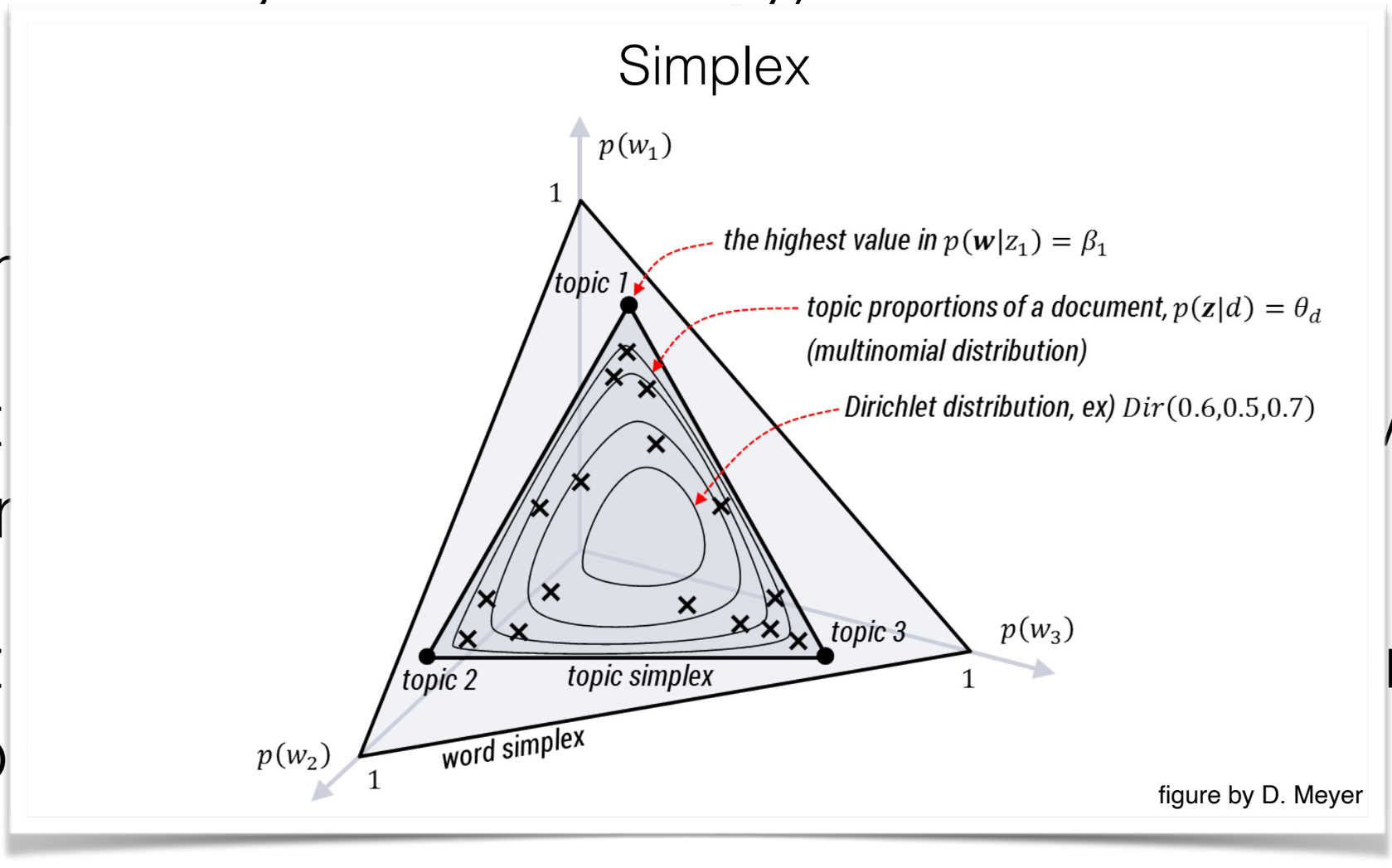


figure by D. Meyer

$\alpha_i$ deterr

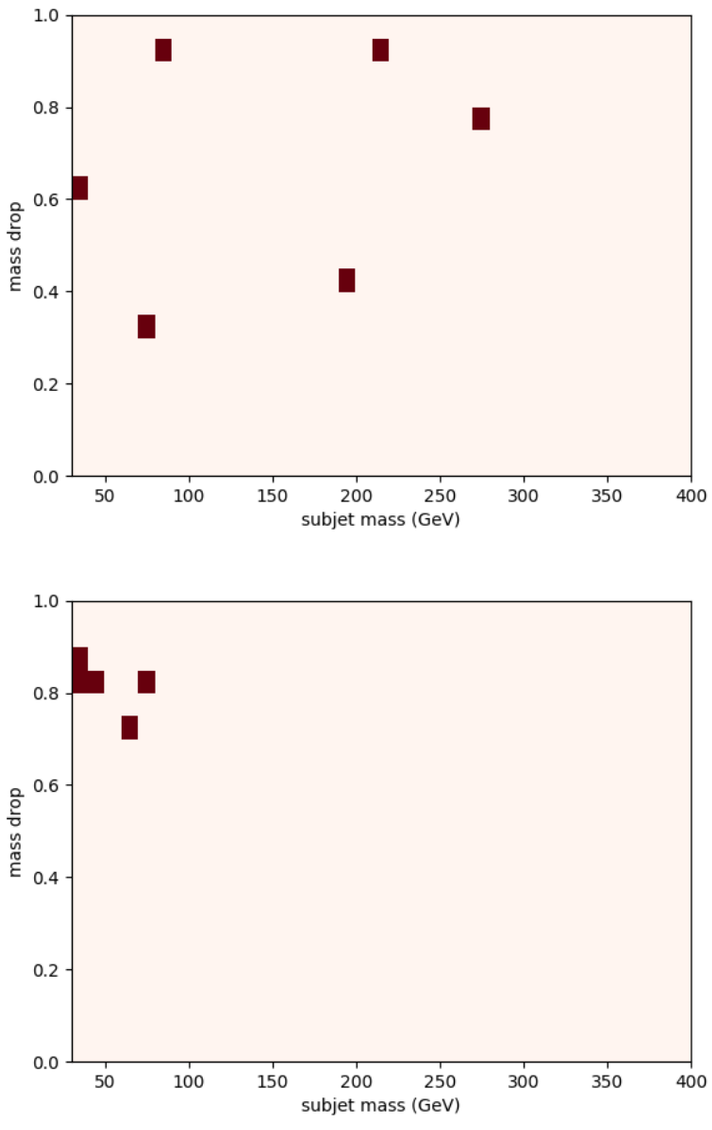
Dirichlet ve
argumer

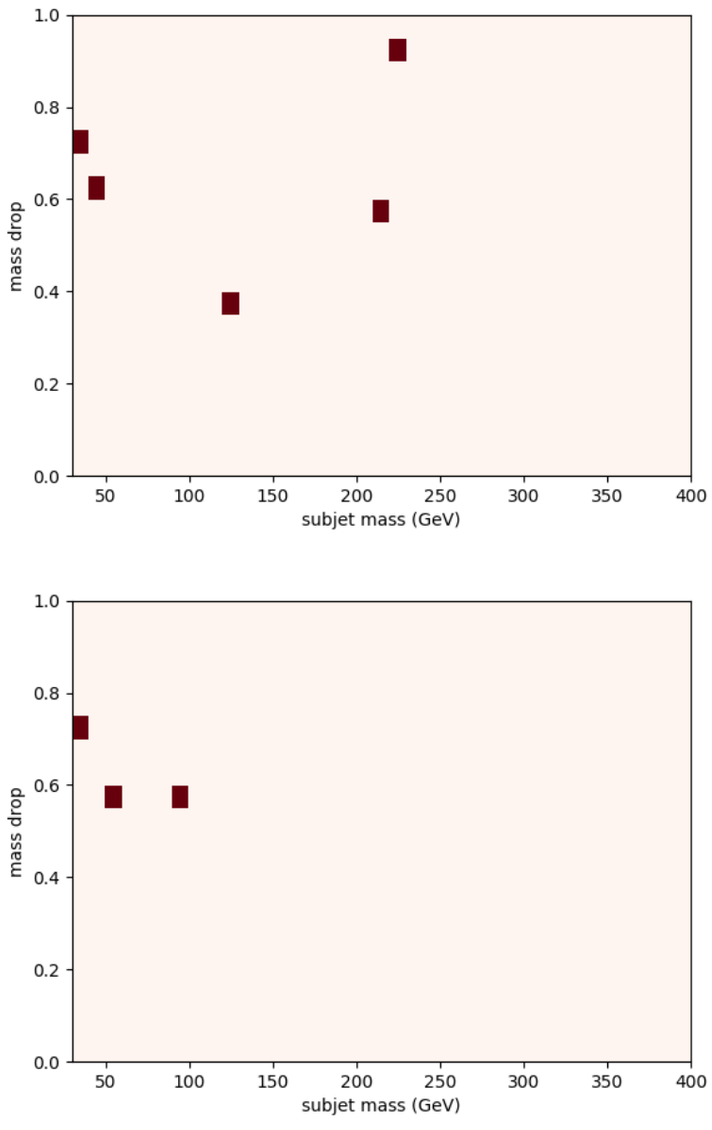
Dirichlet n -
posterio

In jet LDA, themes are V-dimentional Dirichlet; theme proportions are K-dimensional Dirichlet

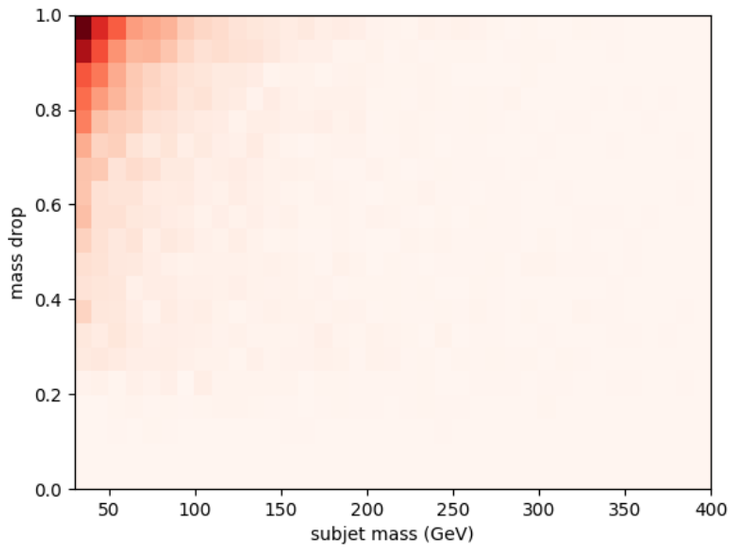
# QCD jets

## Examples:



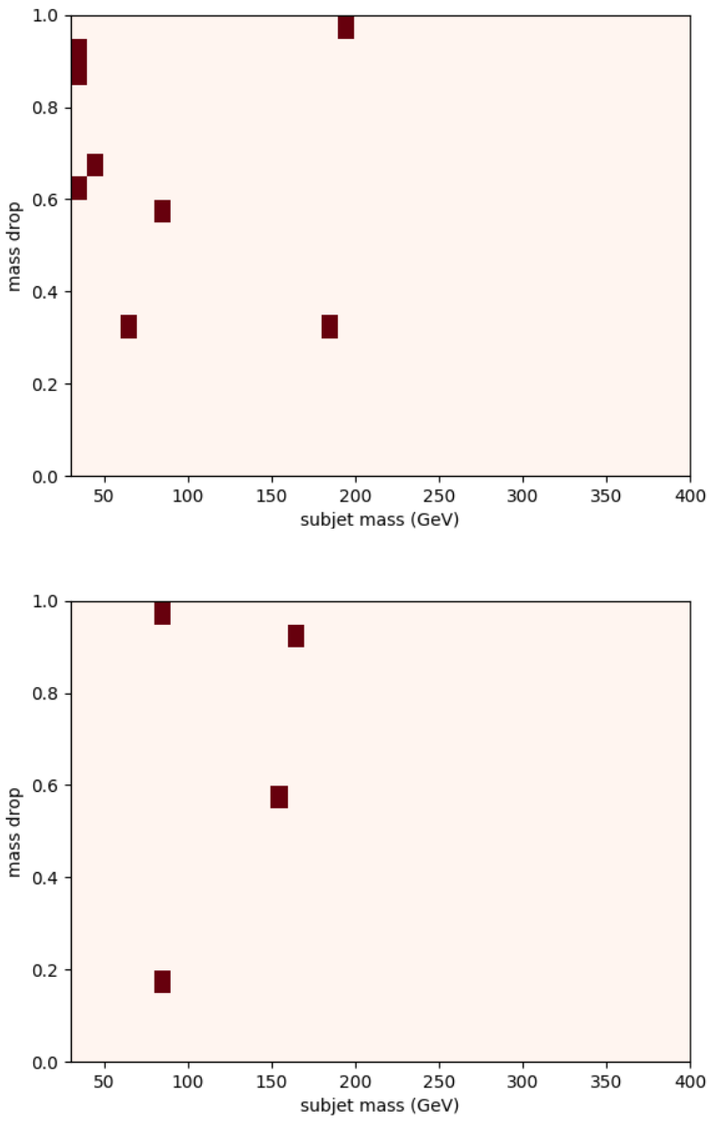



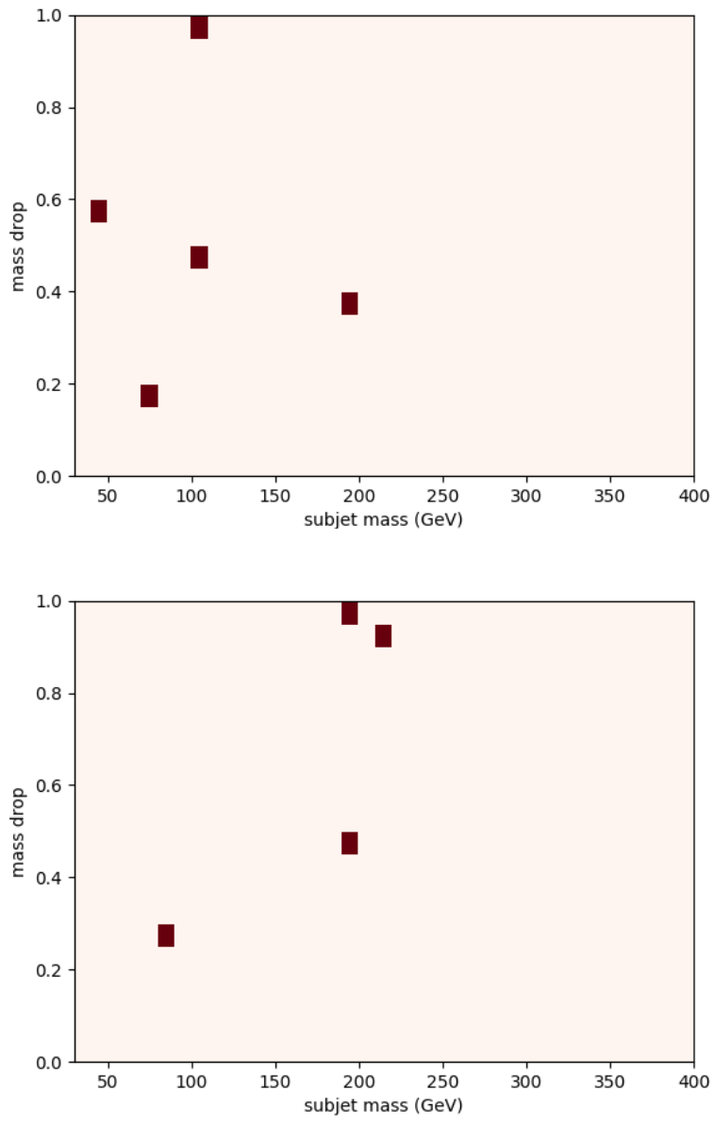

## Average over many jets



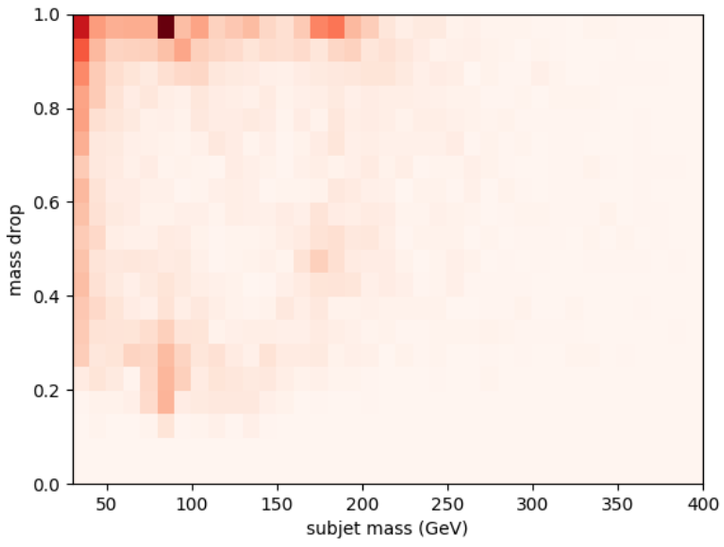
# top jets

## Examples:




## Average over many jets

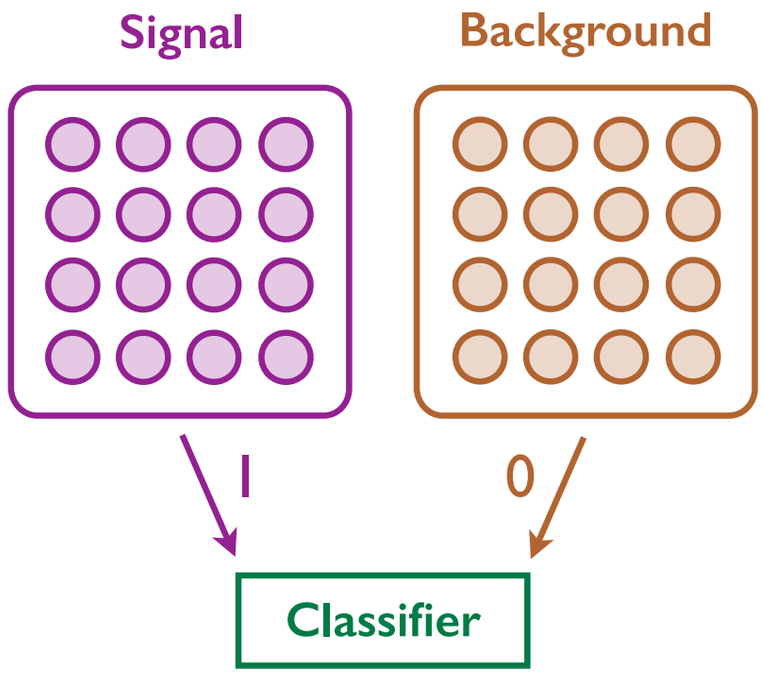


# Jet classification: basics

x list of observables useful for distinguishing S from B

$p_S(x)$ and $p_B(x)$ - probability distributions of x for S and B

classifier h(x) close to 1 for S and close to 0 for B - to be learned by minimizing loss function (e.g mean-square)



# Jet classification: basics

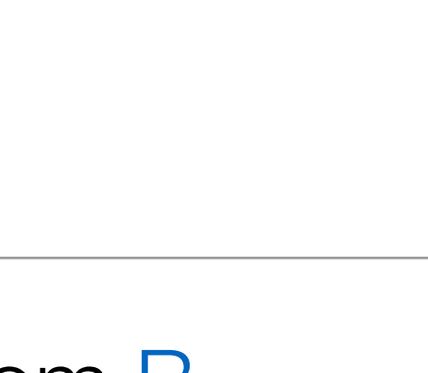
x list of observables useful for distinguishing S from B

$p_S(x)$ and $p_B(x)$ - probability distributions of x for S and B

classifier h(x) close to 1 for S and close to 0 for B - to be learned by minimizing loss function (e.g mean-square)

receiver operating characteristic (ROC) curve

$$\epsilon_S = \int \mathrm{d}\vec{x}\, p_S(\vec{x})\, \Theta(h(\vec{x}) - c)$$
$$\epsilon_B = \int \mathrm{d}\vec{x}\, p_B(\vec{x})\, \Theta(h(\vec{x}) - c)$$




Neyman-Pearson lemma: $h_{\mathrm{optimal}}(\vec{x}) = p_S(\vec{x})/p_B(\vec{x})$ (likelihood ratio)

If x - low dimensional, can use histograms directly, otherwise use supervised ML (BDTs, NNs, …)

# Jet classification: basics

x list of c

$p_S(x)$ anc

classifier

learned b

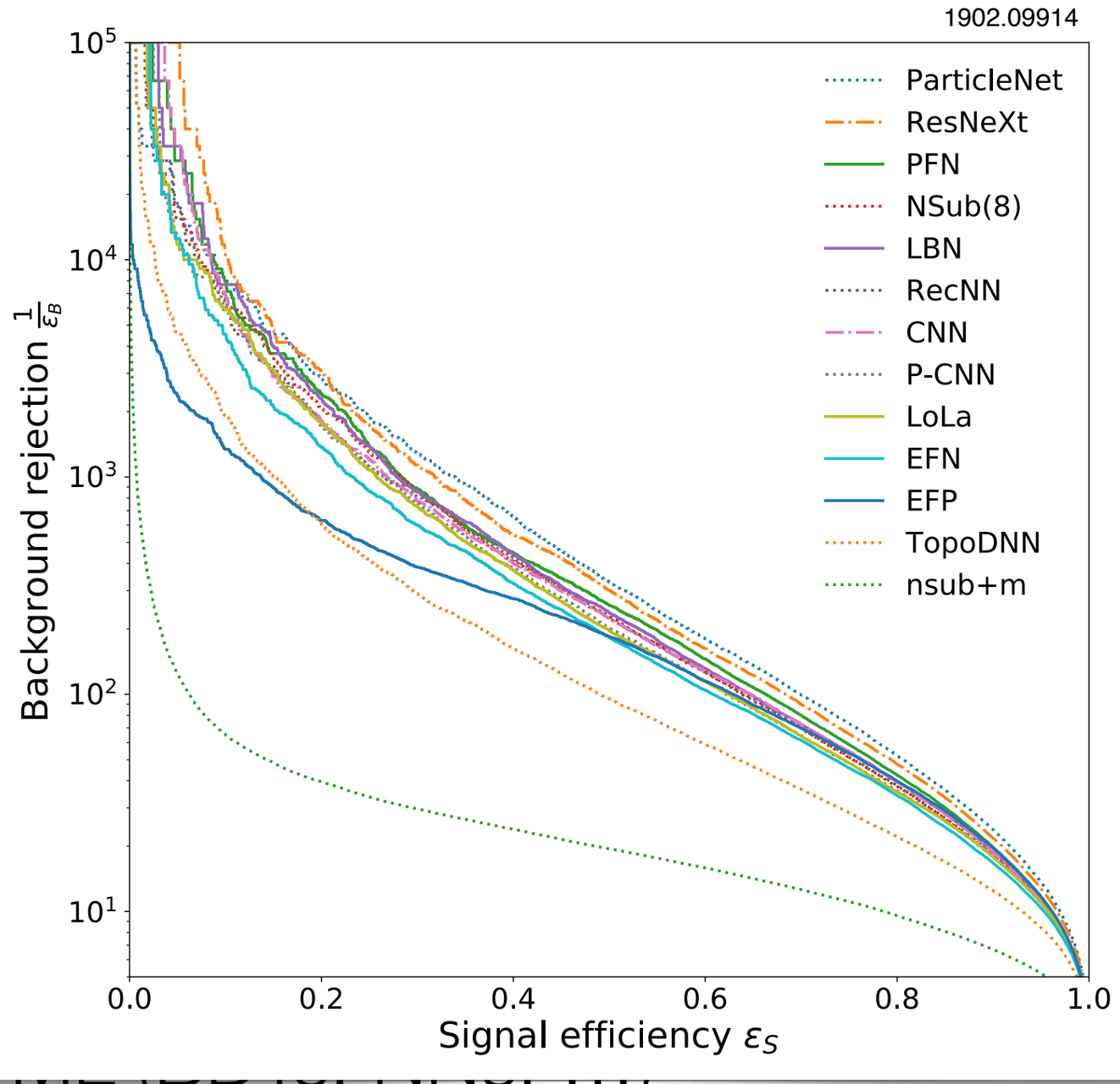
receiver

Neyman-

If x - low

otherwis

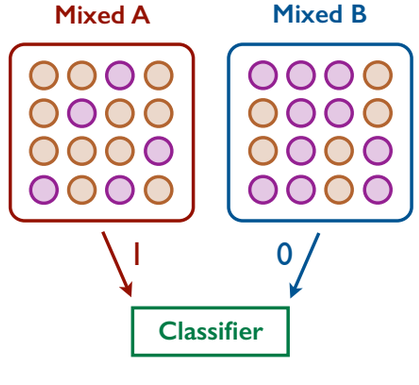
Example: QCD j vs. t classification (top-tagging)



# Jet classification: mixed samples

Classification from mixed samples: pure samples not available in real data

$$p_{M_1}(\vec{x}) = f_1\, p_S(\vec{x}) + (1 - f_1)\, p_B(\vec{x}),$$
$$p_{M_2}(\vec{x}) = f_2\, p_S(\vec{x}) + (1 - f_2)\, p_B(\vec{x}),$$




1.) Assume $f_1$, $f_2$ known (e.g. from MC), then simply

$$h^{M_1/M_2}_{\text{optimal}}(\vec{x}) = p_{M_1}(\vec{x})/p_{M_2}(\vec{x})$$

2.) Assume only $f_1 > f_2$ then use monotonicity of

$$\frac{p_{M_1}}{p_{M_2}} = \frac{f_1\, p_S + (1 - f_1)\, p_B}{f_2\, p_S + (1 - f_2)\, p_B}$$

(Classification Without Labels)

Metodiev, Nachman & Thaler, 1708.02949

Can be used directly on latent distributions!