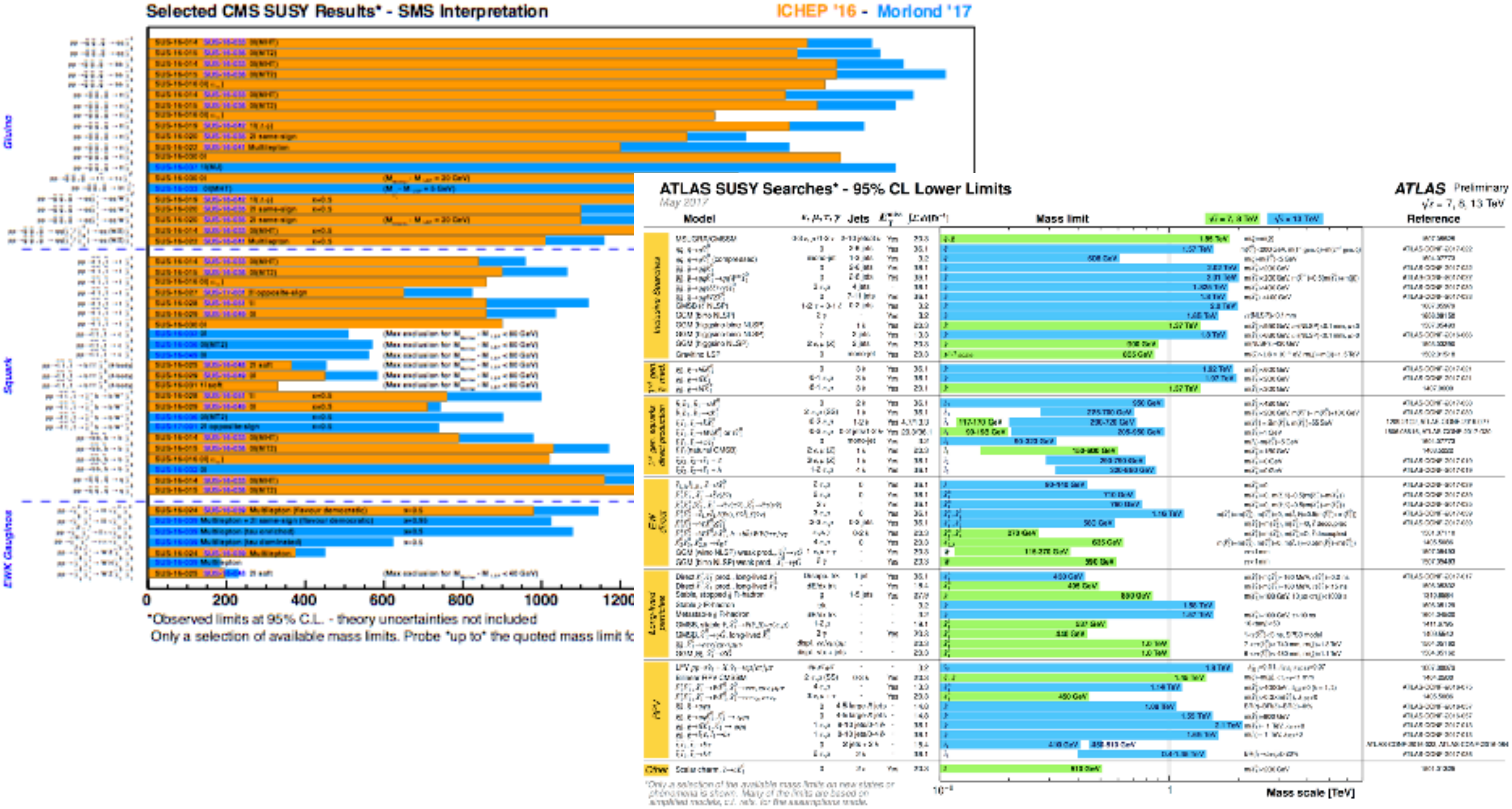
Some Applications of Deep Learning at the LHC

David Shih
NHETC, Rutgers University

NCTS Annual Theory Meeting 2018

Based on Farina, Nakai & DS 1808.08992
and Macaluso & DS 1803.00107

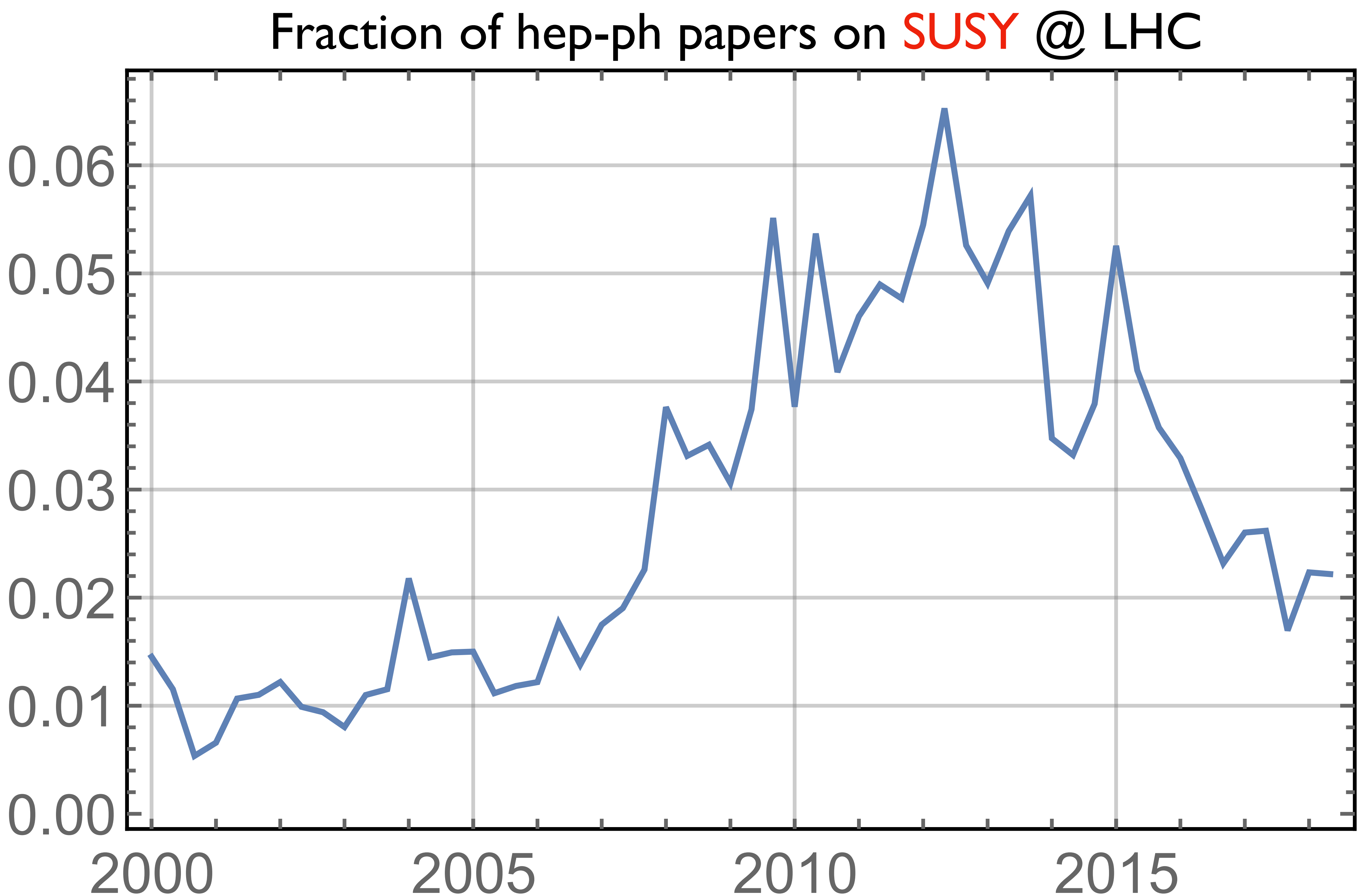


The LHC has searched for new physics in many places.

So far, there has been no evidence of anything beyond the SM.

Many well-motivated models (SUSY, composite higgs, dark matter, …) have not turned up as expected.

People are losing interest in “well-motivated models”…



We need new ideas!

Can we search for new physics in the data without knowing what we're looking for?

Can we find the unexpected?

Can we find a needle in a haystack, without knowing what needles are?

Sounds hopelessly difficult…

Maybe deep learning can help!

Deep learning at LHC

Recently there has been a lot of interest in applications of deep learning to the LHC.

- classification (eg quark/gluon tagging, boosted resonance tagging)
- pile-up removal
- event generation
- triggering
- anomaly detection
- …..

There have been some very impressive successes, especially for classification!

Beginning to be adopted by the LHC collaborations!

Deep learning at LHC

Recently there has been a lot of interest in applications of deep learning to the LHC.

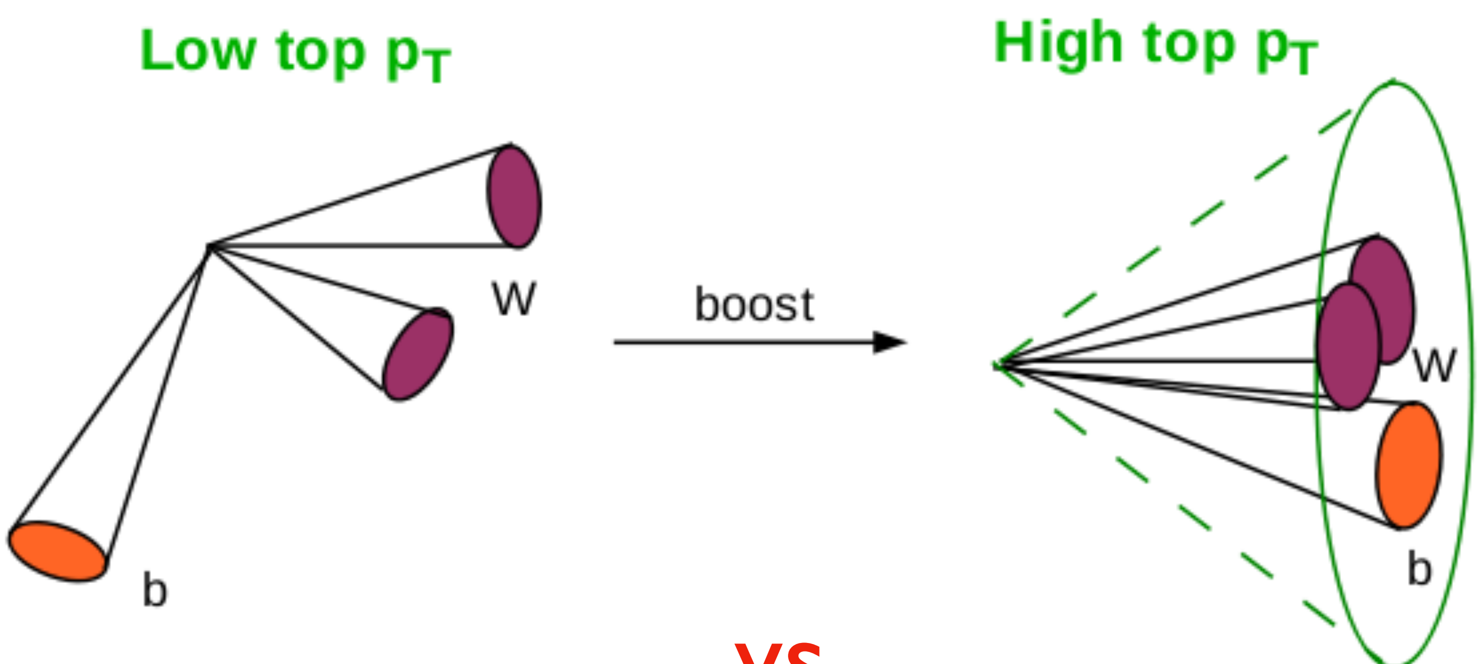
- classification (eg quark/gluon tagging, boosted resonance tagging)
- pile-up removal
- event generation
- triggering
- anomaly detection
- …..

This talk

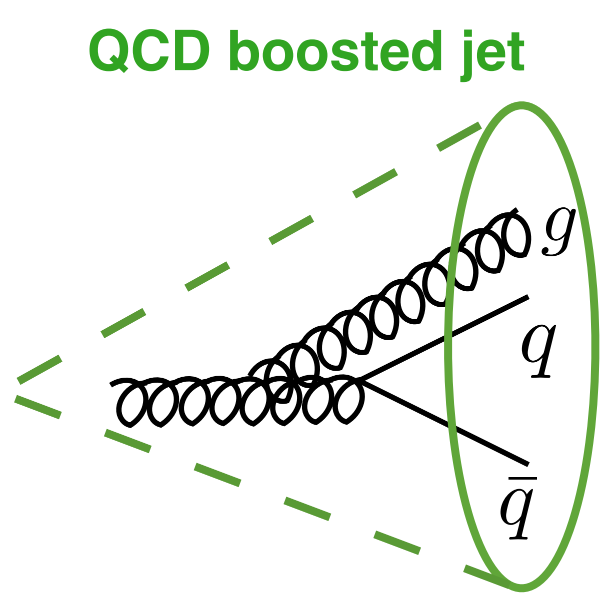
There have been some very impressive successes, especially for classification!

Beginning to be adopted by the LHC collaborations!

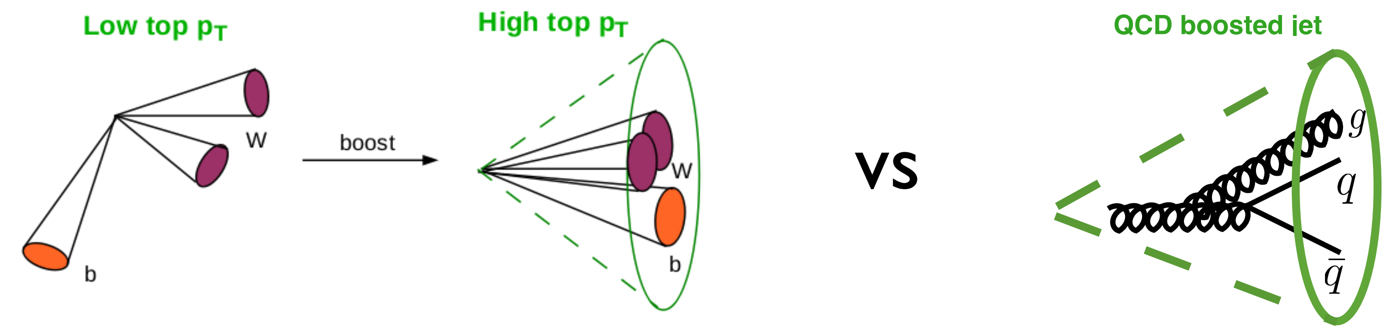
A benchmark application: boosted resonance tagging



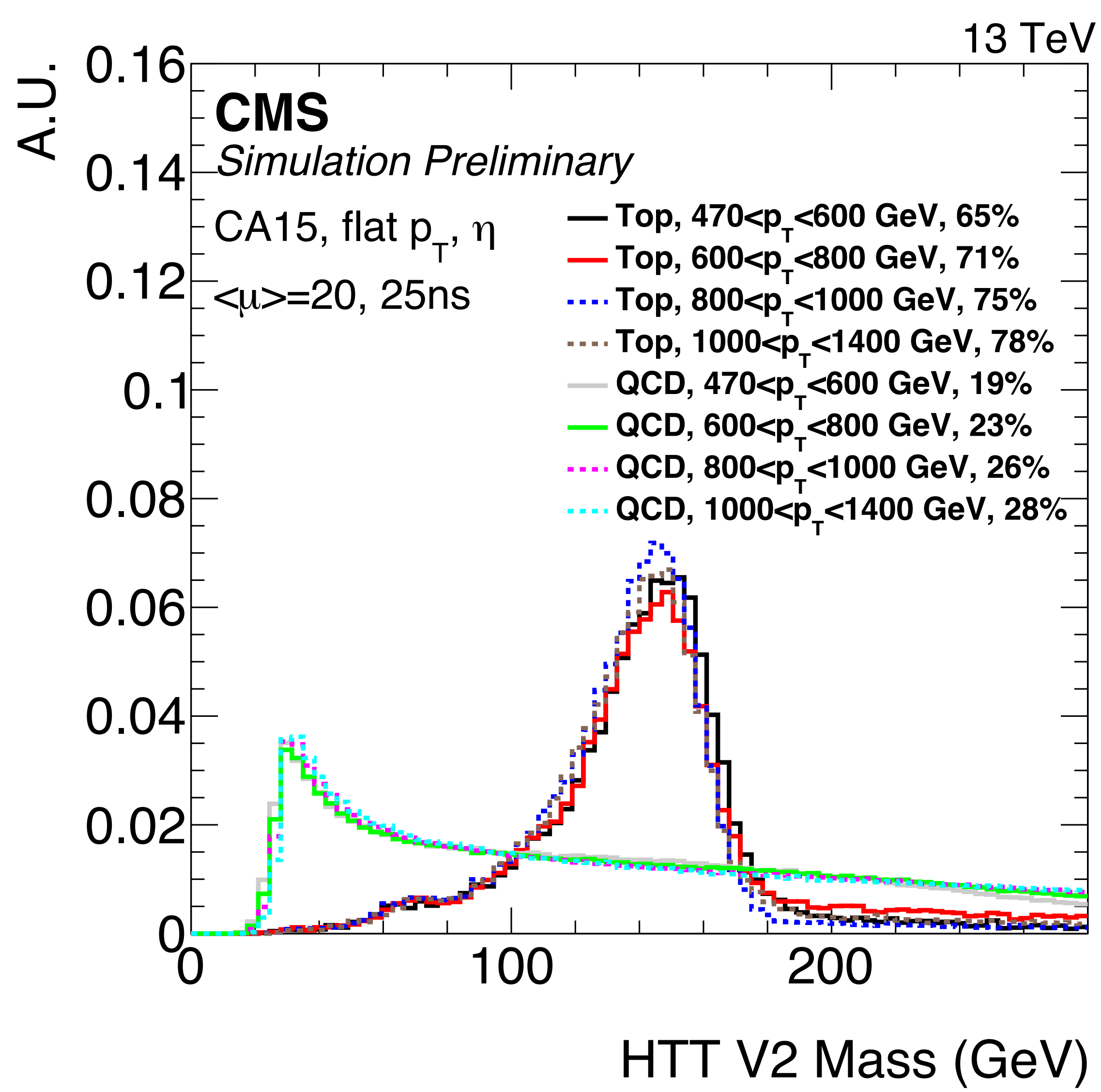
VS.

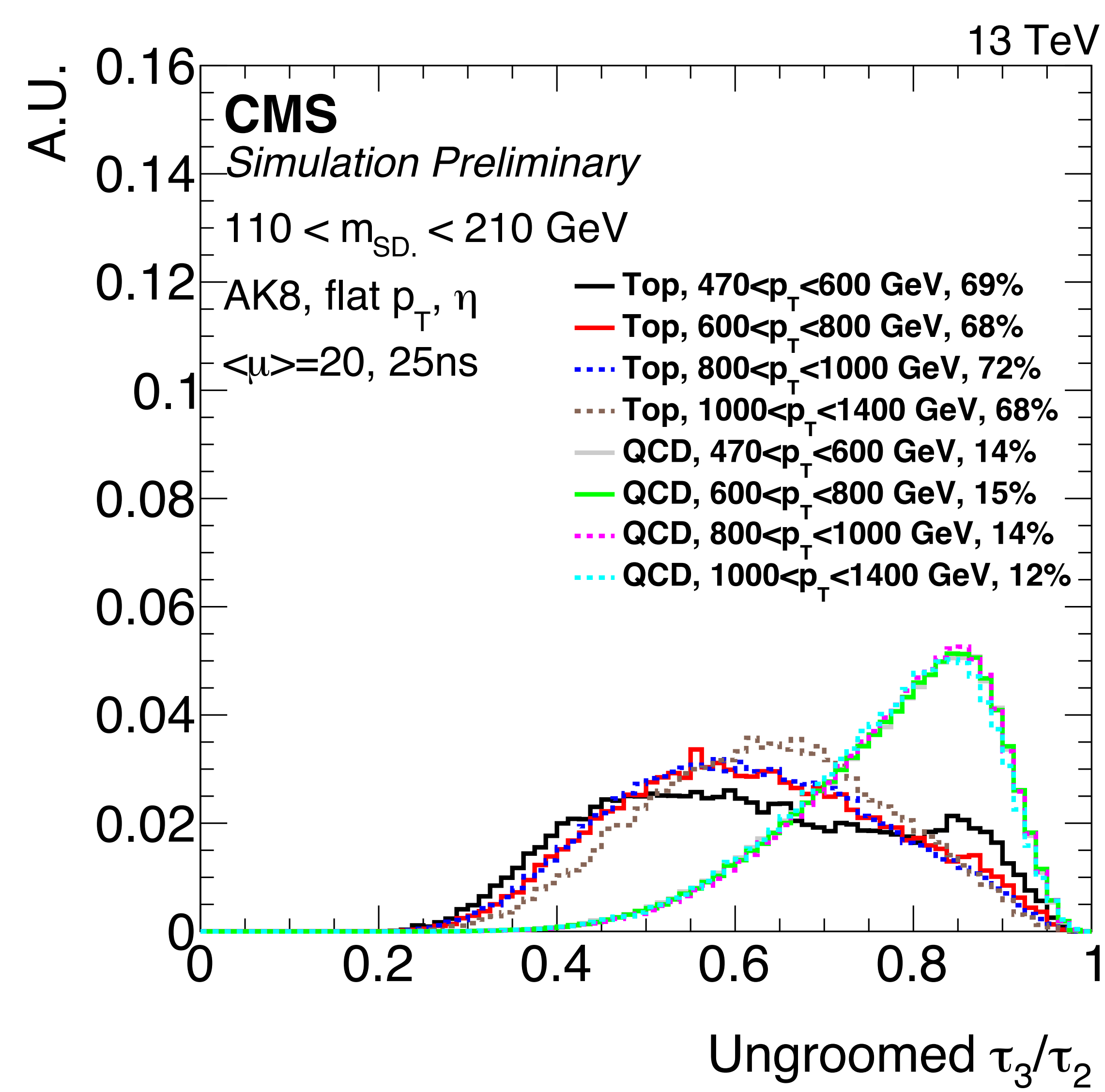


How to differentiate between these two types of jets??



Some obvious ideas:



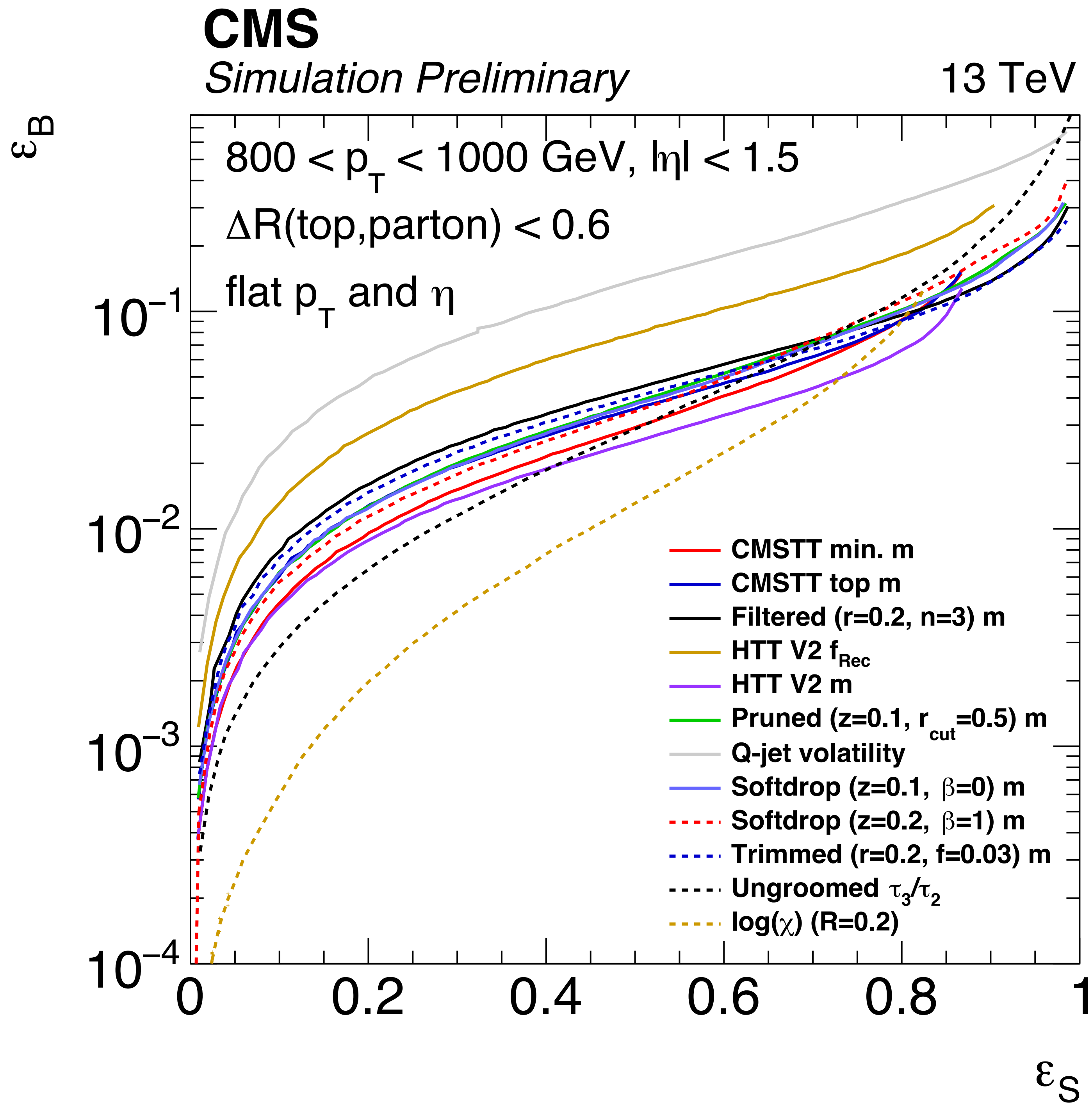


jet mass (m_{top} vs 0)

jet substructure (3 vs 1)

State of the art with cuts on kinematic quantities:

QCD jet mistag rate



“ROC curve”

top tagging efficiency

Deep learning can do much better!

Plan of the talk

1. Introduction to Deep Learning

   Convolutional Neural Networks

2. Jet Images

   Example: Top Tagging with CNNs

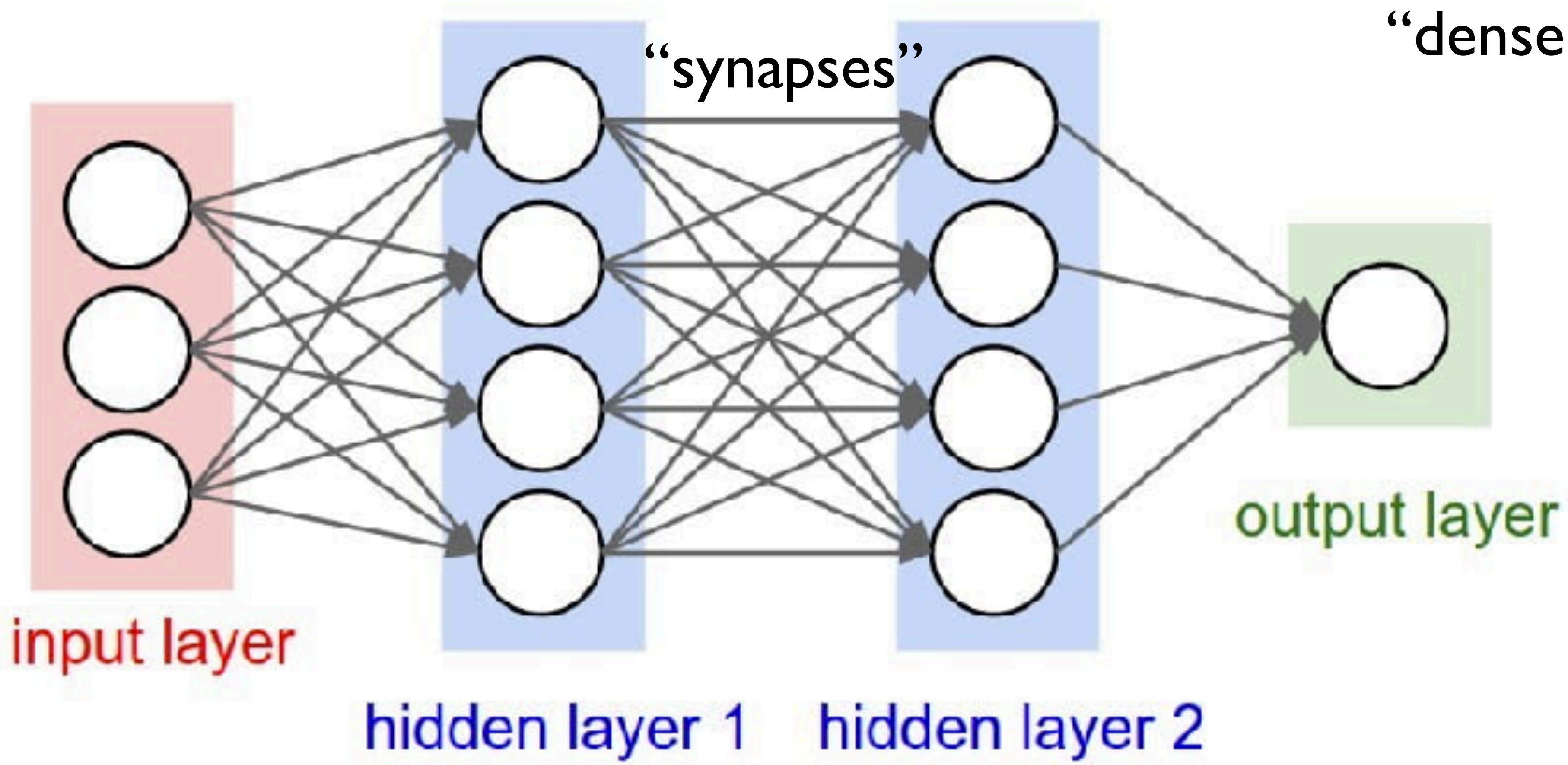
3. Deep Autoencoders for Anomaly Detection at the LHC

What is deep learning?

Deep learning refers to a powerful new class of neural networks with many hidden layers.

“neurons”

“fully connected” or
“dense” NN

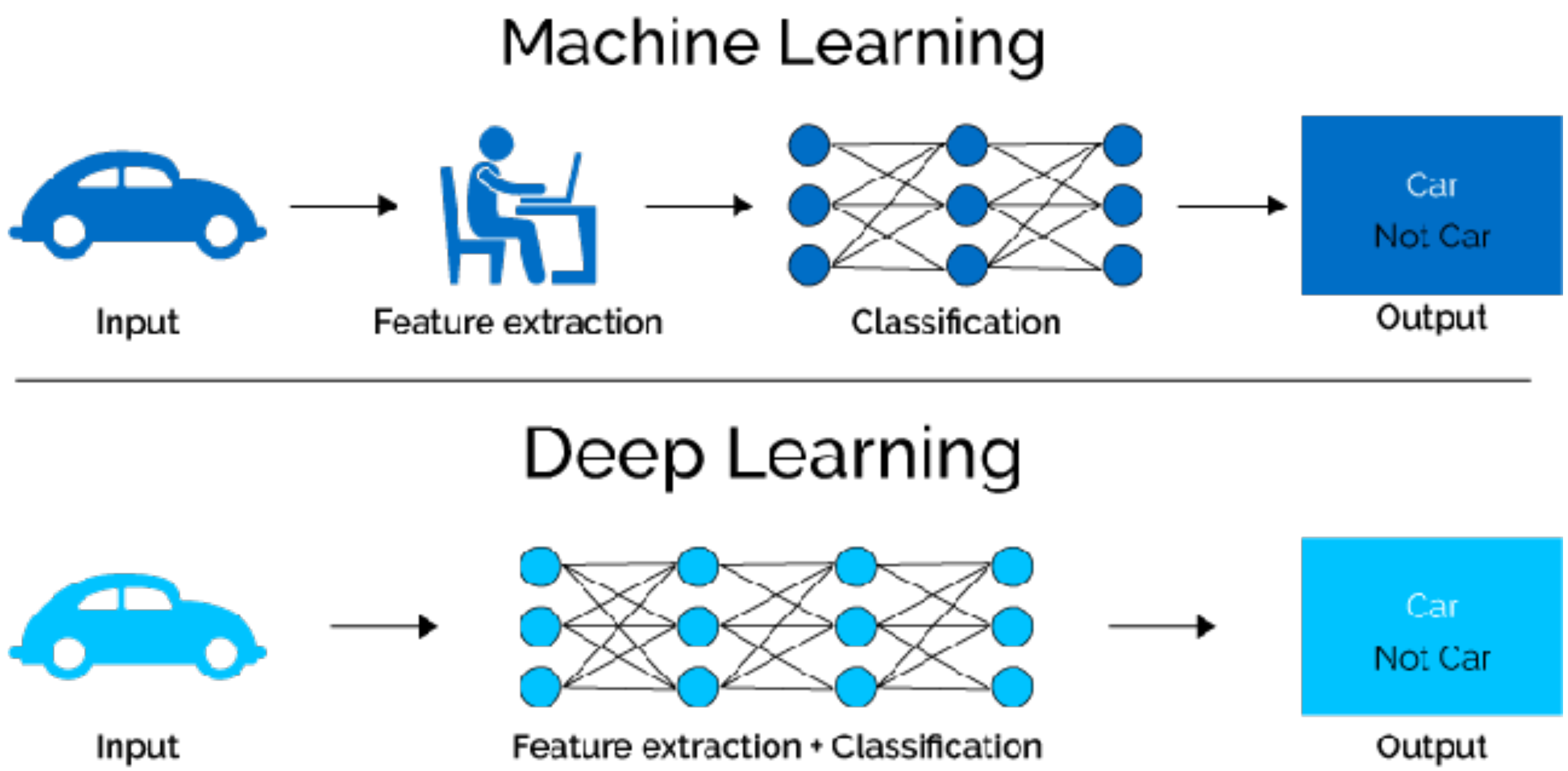


$$x'_{i+1} = a(w_i x_i + b_i)$$

What is deep learning?

The many hidden layers enable the deep NN to learn more abstract concepts (such as "car" and "not car"), starting from raw inputs (e.g. images).

Deep NNs automate the process of "feature engineering".

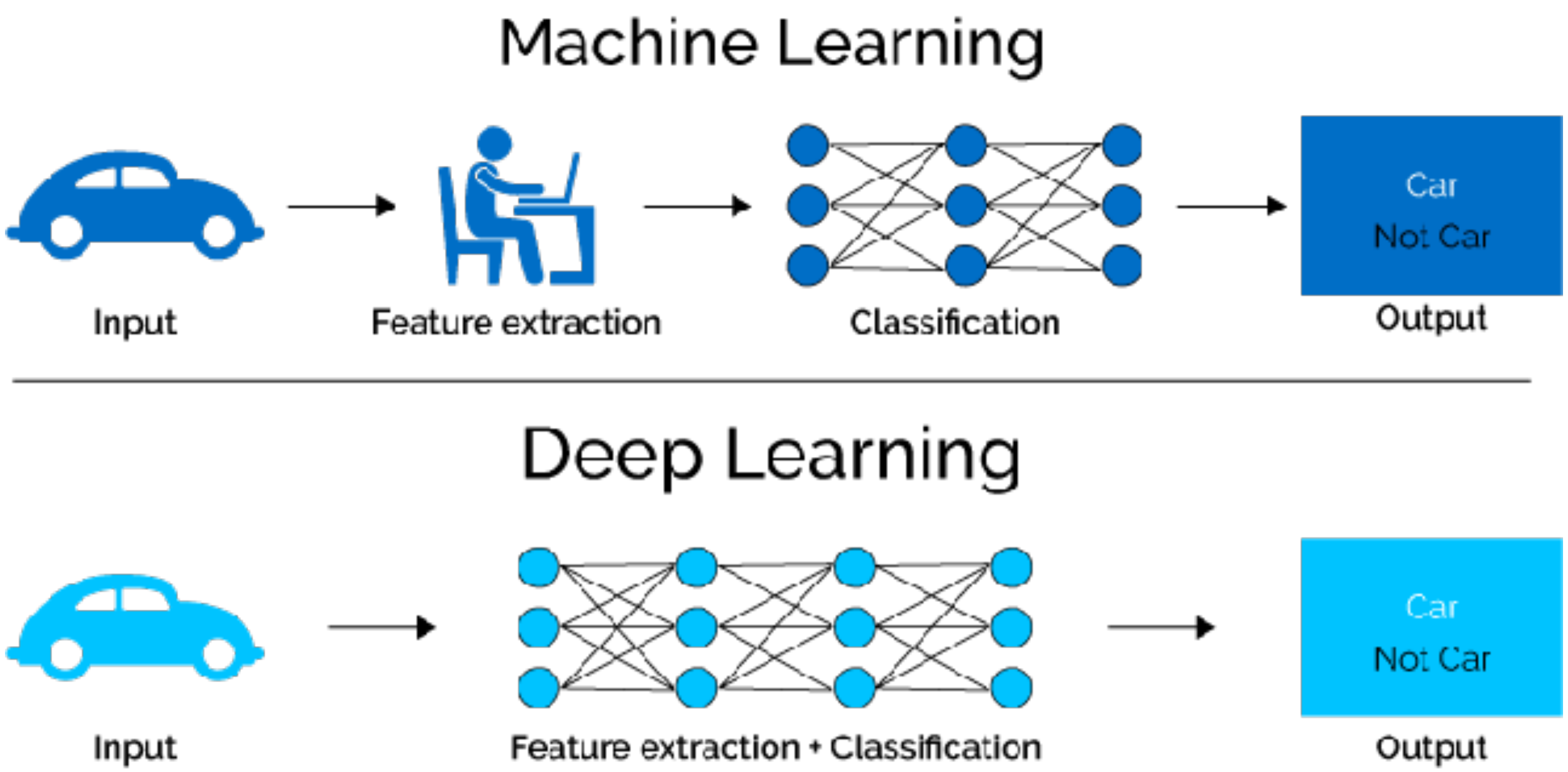


From towardsdatascience.com

What is deep learning?

The many hidden layers enable the deep NN to learn more abstract concepts (such as "car" and "not car"), starting from raw inputs (e.g. images).

Deep NNs automate the process of "feature engineering".



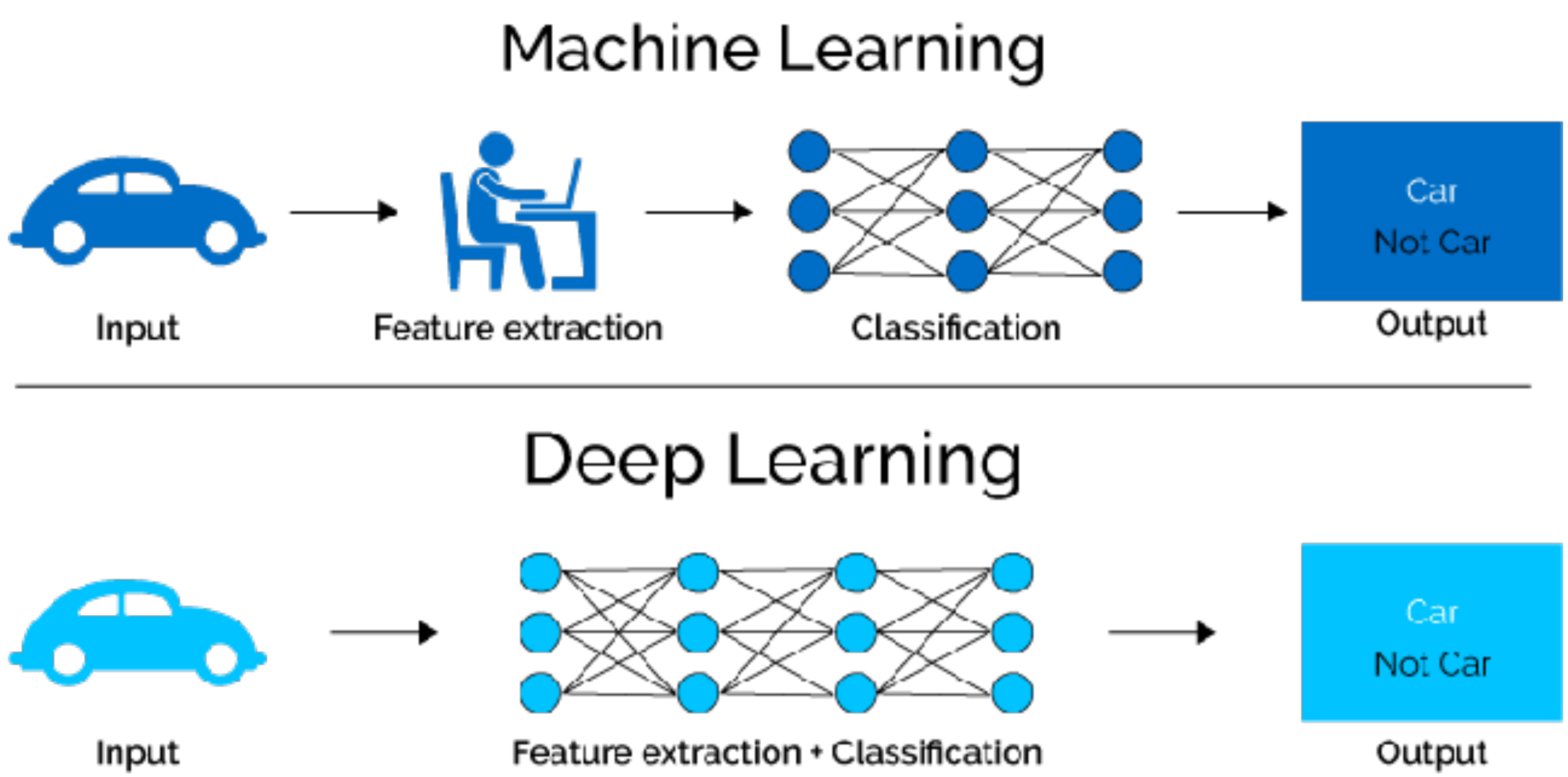
From towardsdatascience.com

Physics Example:

What is deep learning?

The many hidden layers enable the deep NN to learn more abstract concepts (such as "car" and "not car"), starting from raw inputs (e.g. images).

Deep NNs automate the process of "feature engineering".



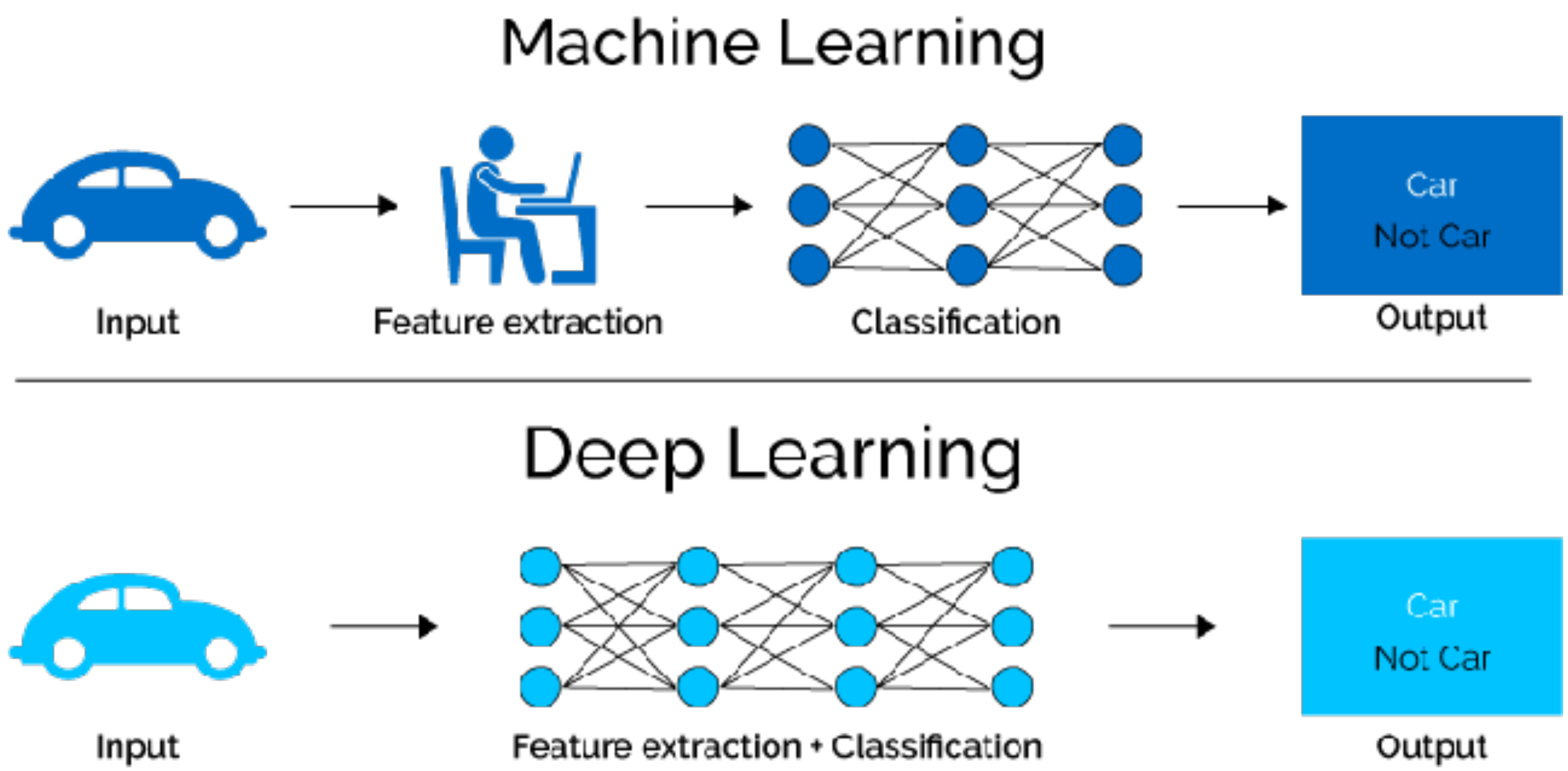
From towardsdatascience.com

Physics Example: Jets

What is deep learning?

The many hidden layers enable the deep NN to learn more abstract concepts (such as "car" and "not car"), starting from raw inputs (e.g. images).

Deep NNs automate the process of "feature engineering".



From towardsdatascience.com

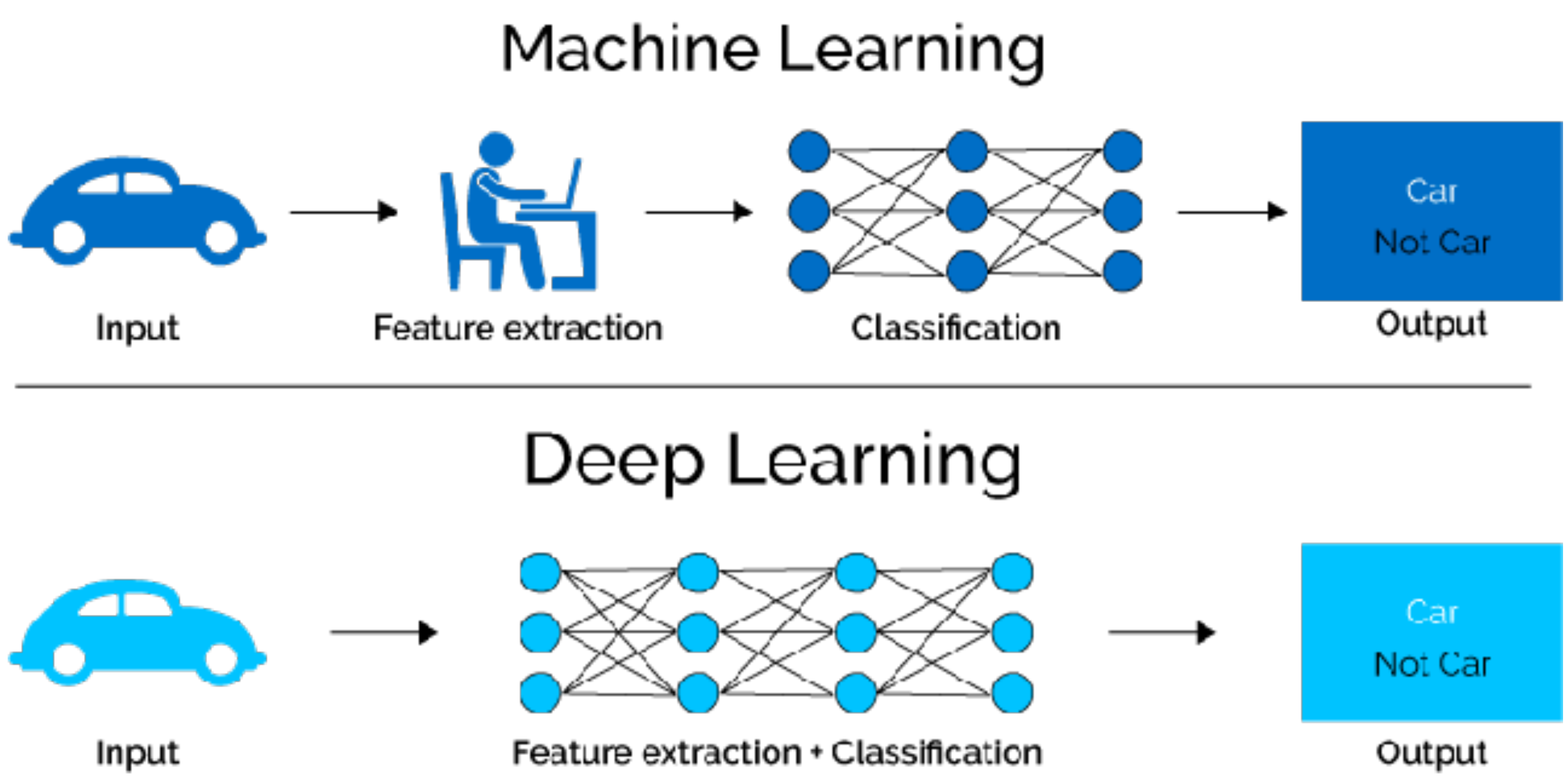
Physics Example:

Jets →

What is deep learning?

The many hidden layers enable the deep NN to learn more abstract concepts (such as "car" and "not car"), starting from raw inputs (e.g. images).

Deep NNs automate the process of "feature engineering".



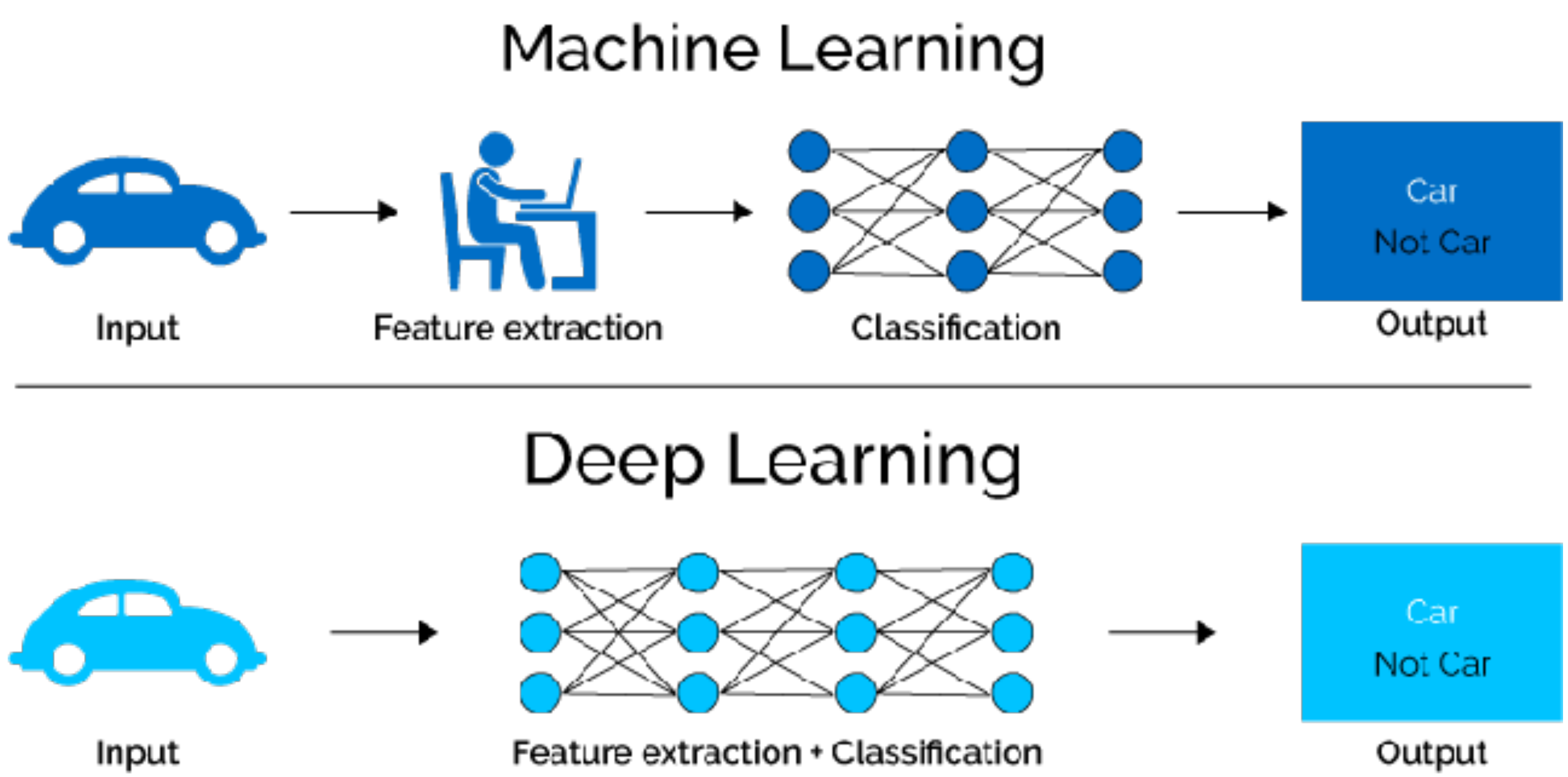
From towardsdatascience.com

Physics Example: Jets $\longrightarrow$ $m_{inv}, \tau_{21}, \tau_{32}, \ldots$

What is deep learning?

The many hidden layers enable the deep NN to learn more abstract concepts (such as "car" and "not car"), starting from raw inputs (e.g. images).

Deep NNs automate the process of "feature engineering".



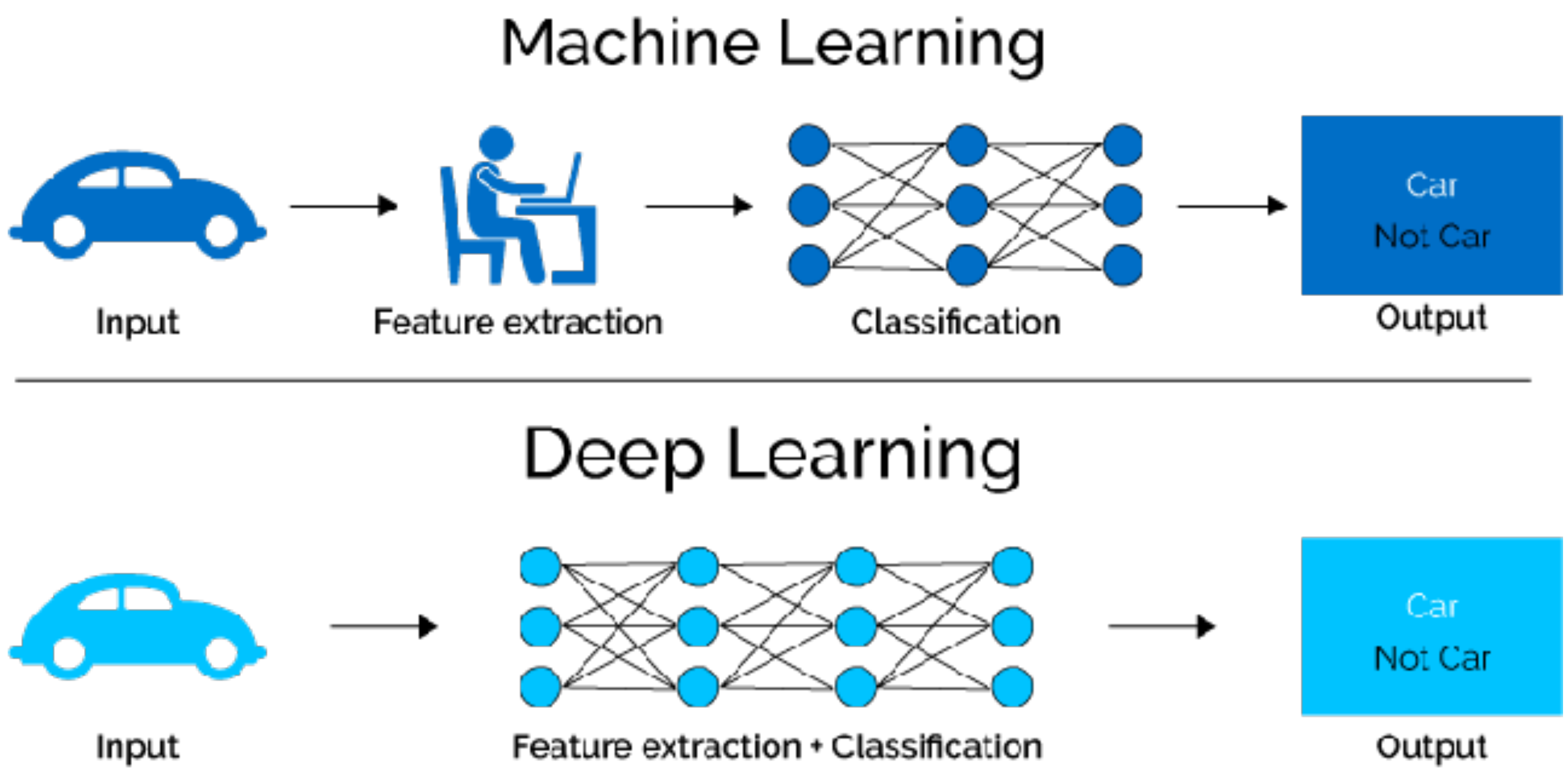
From towardsdatascience.com

Physics Example: Jets $\longrightarrow$ m_{inv}, τ_{21}, τ_{32}, … $\longrightarrow$

What is deep learning?

The many hidden layers enable the deep NN to learn more abstract concepts (such as "car" and "not car"), starting from raw inputs (e.g. images).

Deep NNs automate the process of "feature engineering".



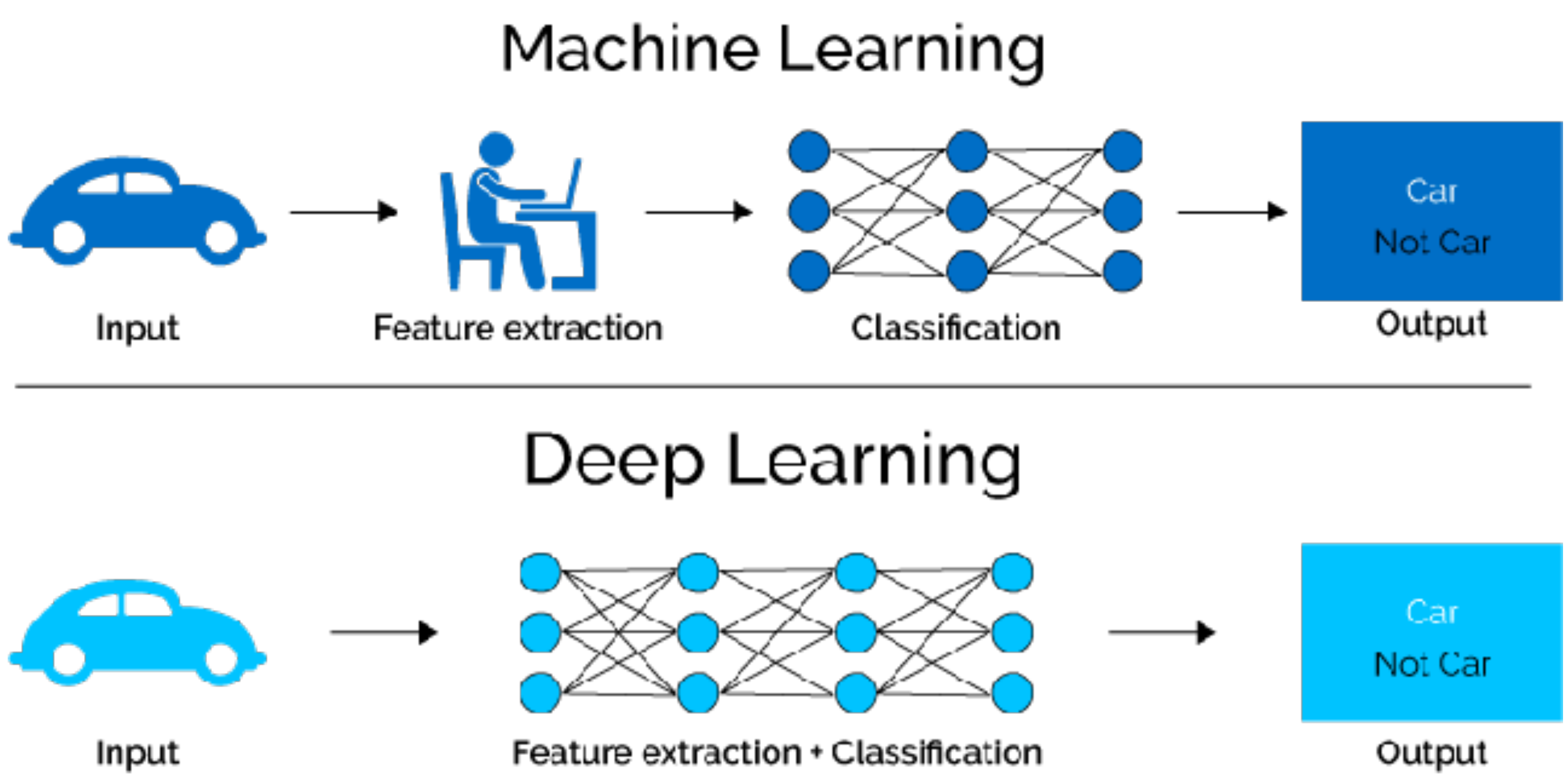
From towardsdatascience.com

Physics Example: Jets $\longrightarrow$ $m_{inv}, \tau_{21}, \tau_{32}, \ldots$ $\longrightarrow$ Cuts

What is deep learning?

The many hidden layers enable the deep NN to learn more abstract concepts (such as "car" and "not car"), starting from raw inputs (e.g. images).

Deep NNs automate the process of "feature engineering".



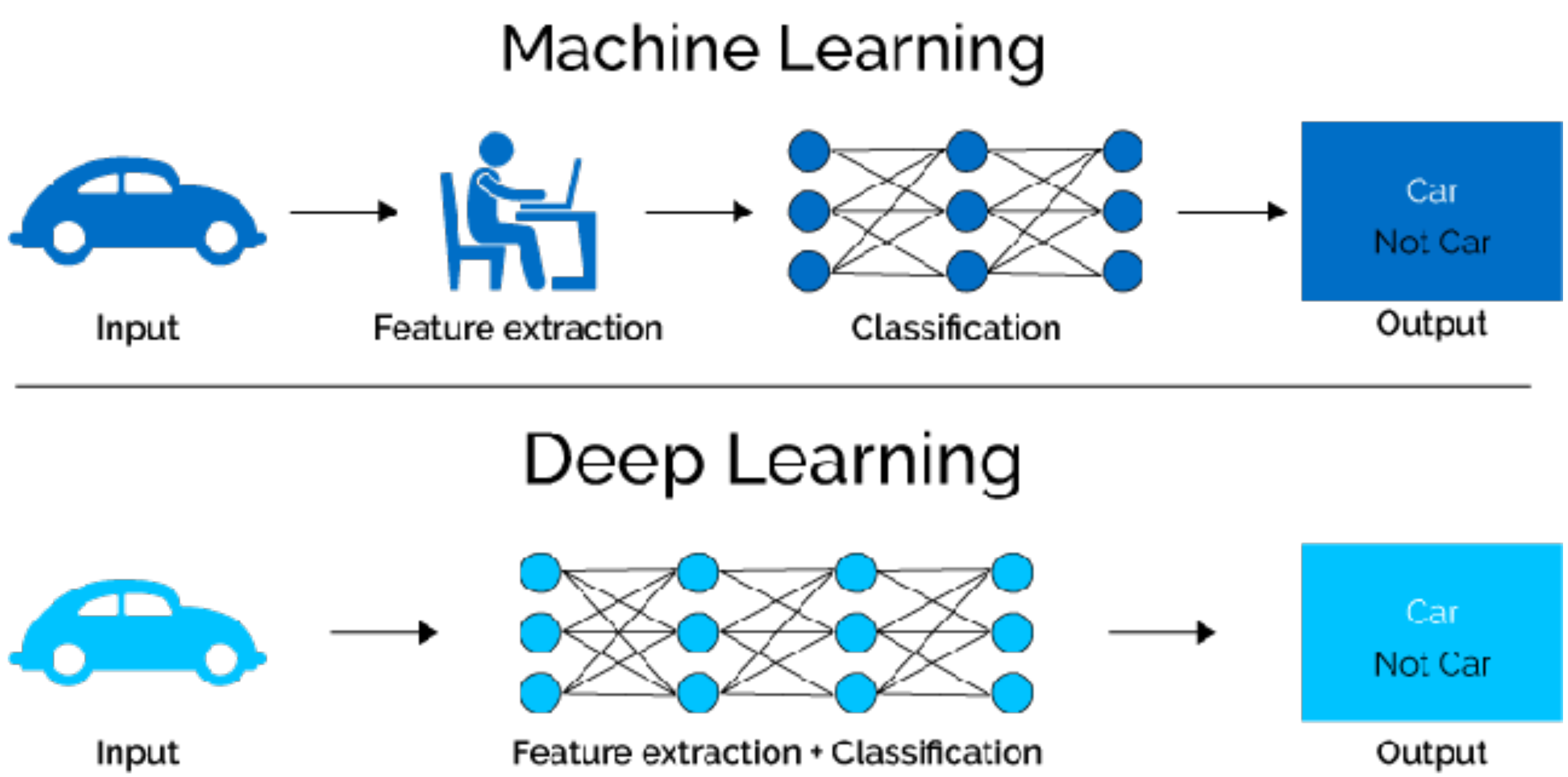
From towardsdatascience.com

Physics Example: Jets → m_{inv}, τ_{21}, τ_{32}, … → Cuts →

What is deep learning?

The many hidden layers enable the deep NN to learn more abstract concepts (such as "car" and "not car"), starting from raw inputs (e.g. images).

Deep NNs automate the process of "feature engineering".



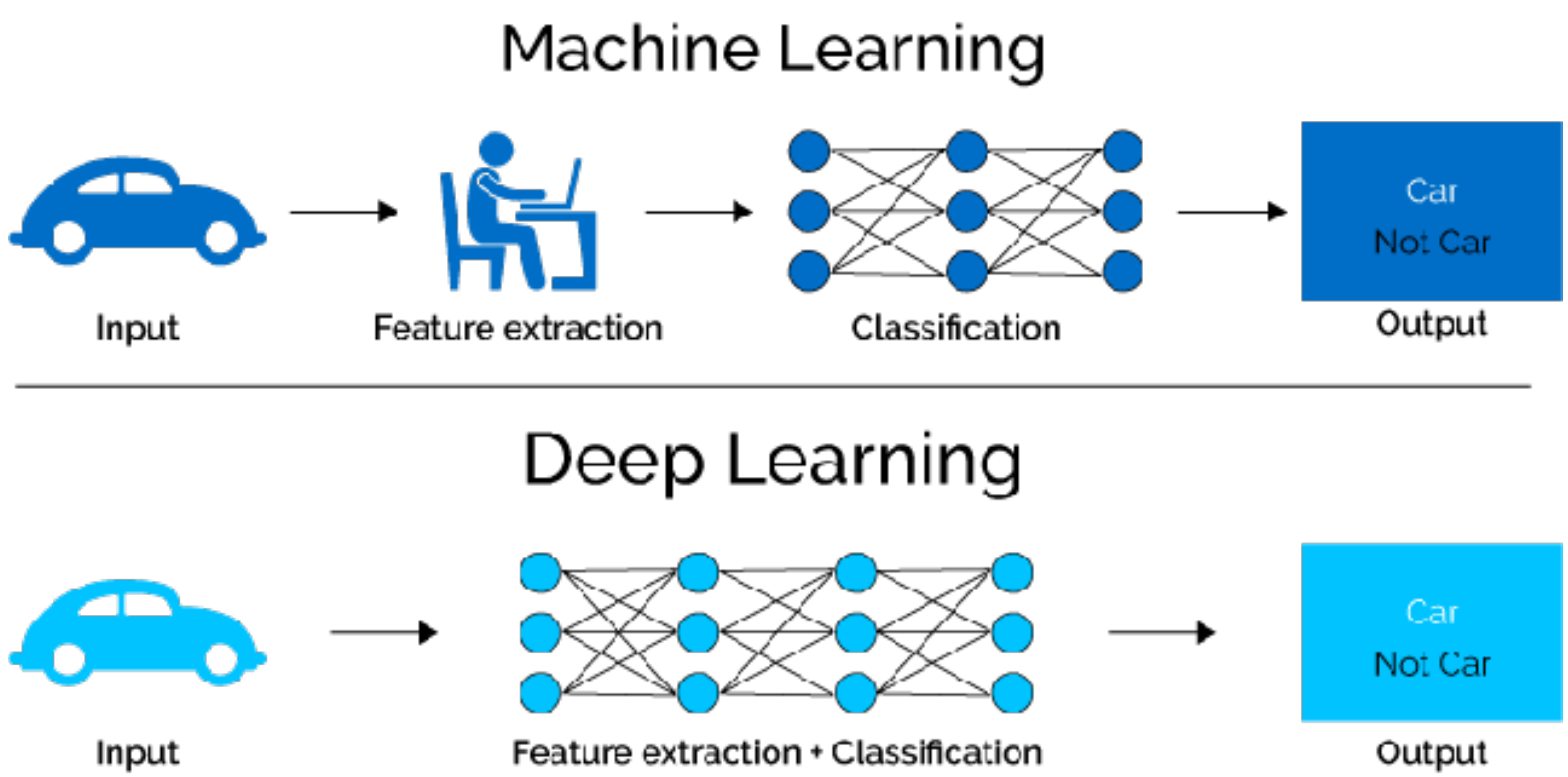
From towardsdatascience.com

Physics Example: Jets $\longrightarrow$ m_{inv}, τ_{21}, τ_{32}, ... $\longrightarrow$ Cuts $\longrightarrow$ Top or QCD

What is deep learning?

The many hidden layers enable the deep NN to learn more abstract concepts (such as "car" and "not car"), starting from raw inputs (e.g. images).

Deep NNs automate the process of "feature engineering".



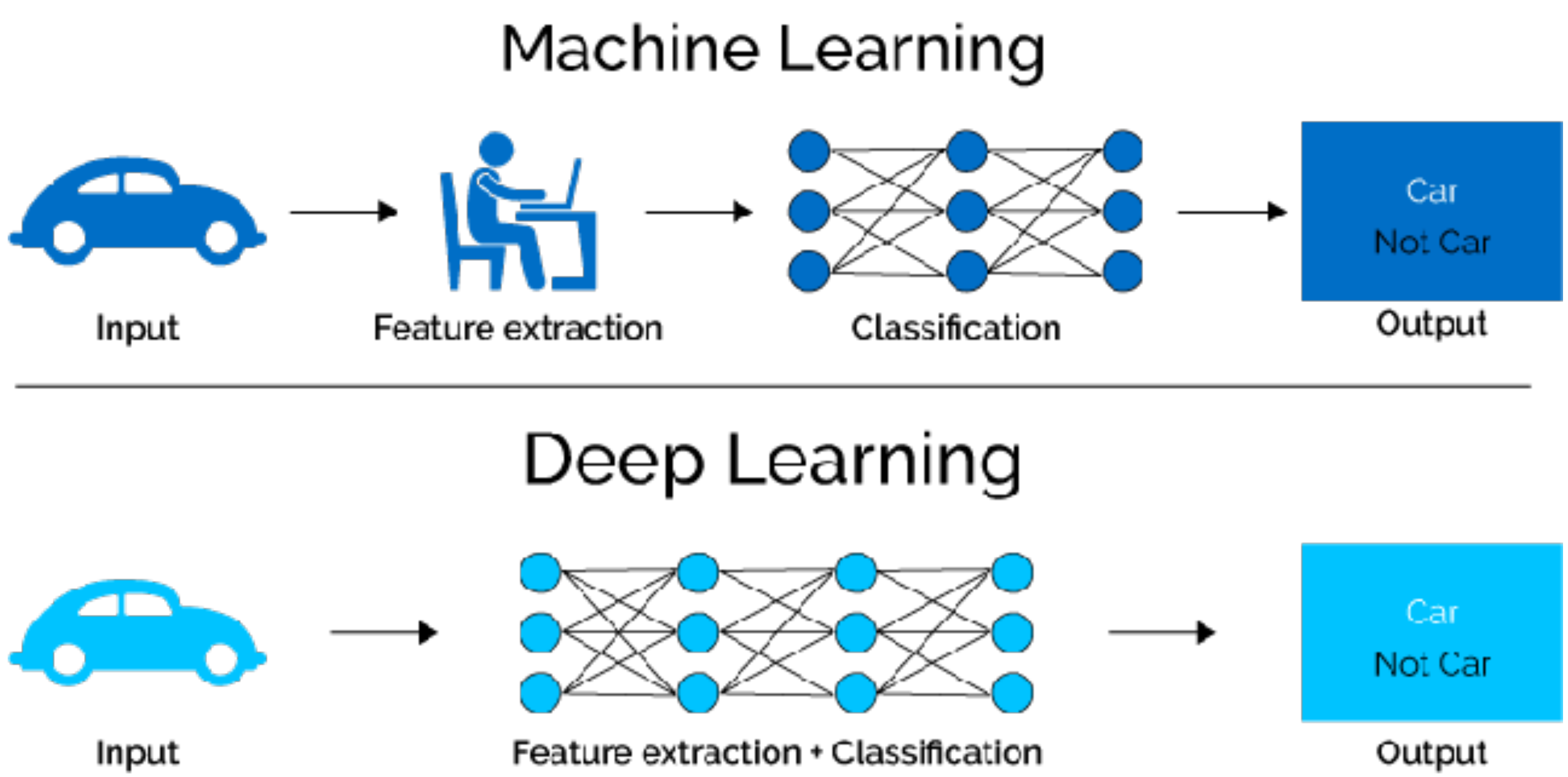
From towardsdatascience.com

Physics Example: Jets $\longrightarrow$ $m_{inv}, \tau_{21}, \tau_{32}, \ldots$ $\longrightarrow$ BDT $\longrightarrow$ Top or QCD

What is deep learning?

The many hidden layers enable the deep NN to learn more abstract concepts (such as "car" and "not car"), starting from raw inputs (e.g. images).

Deep NNs automate the process of "feature engineering".

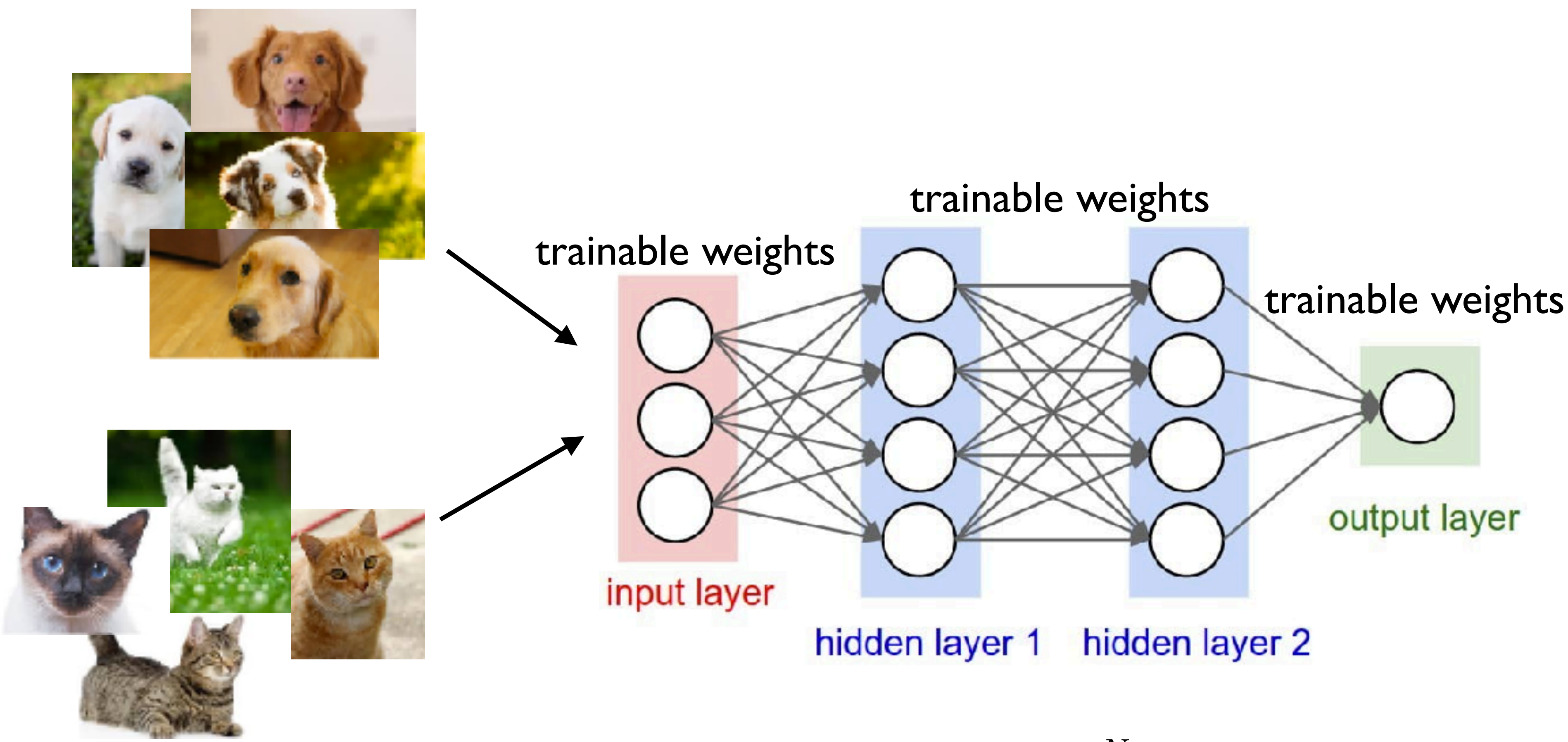


From towardsdatascience.com

Physics Example: Jets → Deep learning algorithm → Top or QCD

Training the deep neural network

Neural networks need to be "trained" on a set of examples. Goal of training is to minimize a "loss function" that quantifies performance of NN.



$NN(x_i)$=prob(cat)

y_i = truth value
= 0 or 1

x_i

"Mean squared error" $L = \sum_{i=1}^{N} (NN(x_i; w) - y_i)^2$

"Binary cross entropy" $L = \sum_{i=1}^{N} \Big(y_i \log(NN(x_i; w)) + (1 - y_i) \log(1 - NN(x_i; w)) \Big)$

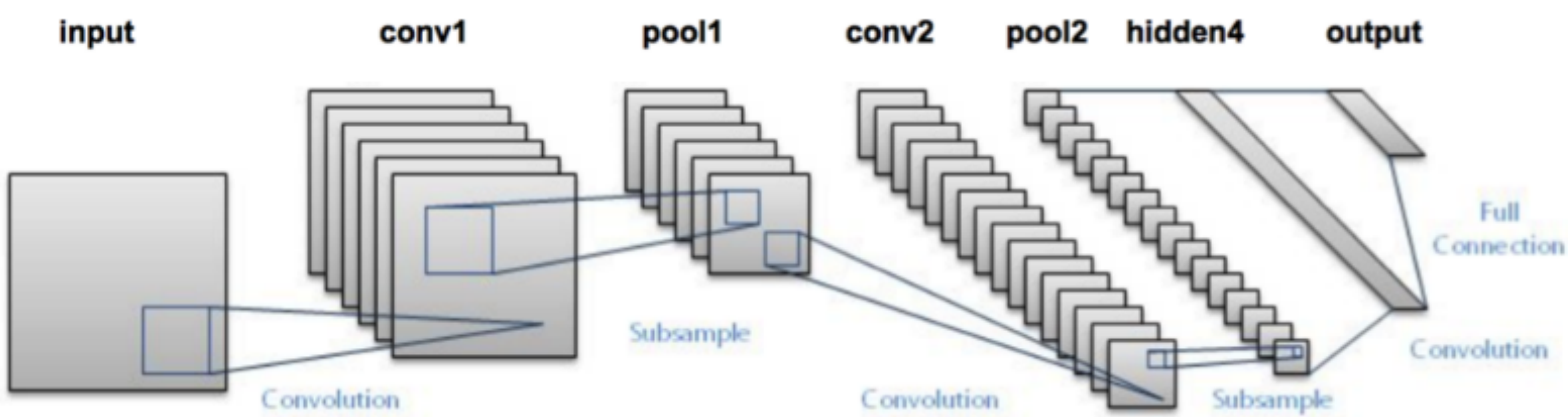
Successes of deep learning

Many stunning real-world successes in recent years…

- Image recognition
- Self-driving cars
- Amazon Go
- Speech and text recognition
- Autocomplete/Autocorrect
- Digital assistants (Siri/Alexa/Google Home/…)
- AlphaGo
- Chess
- …

Deep Learning for Images: Computer Vision

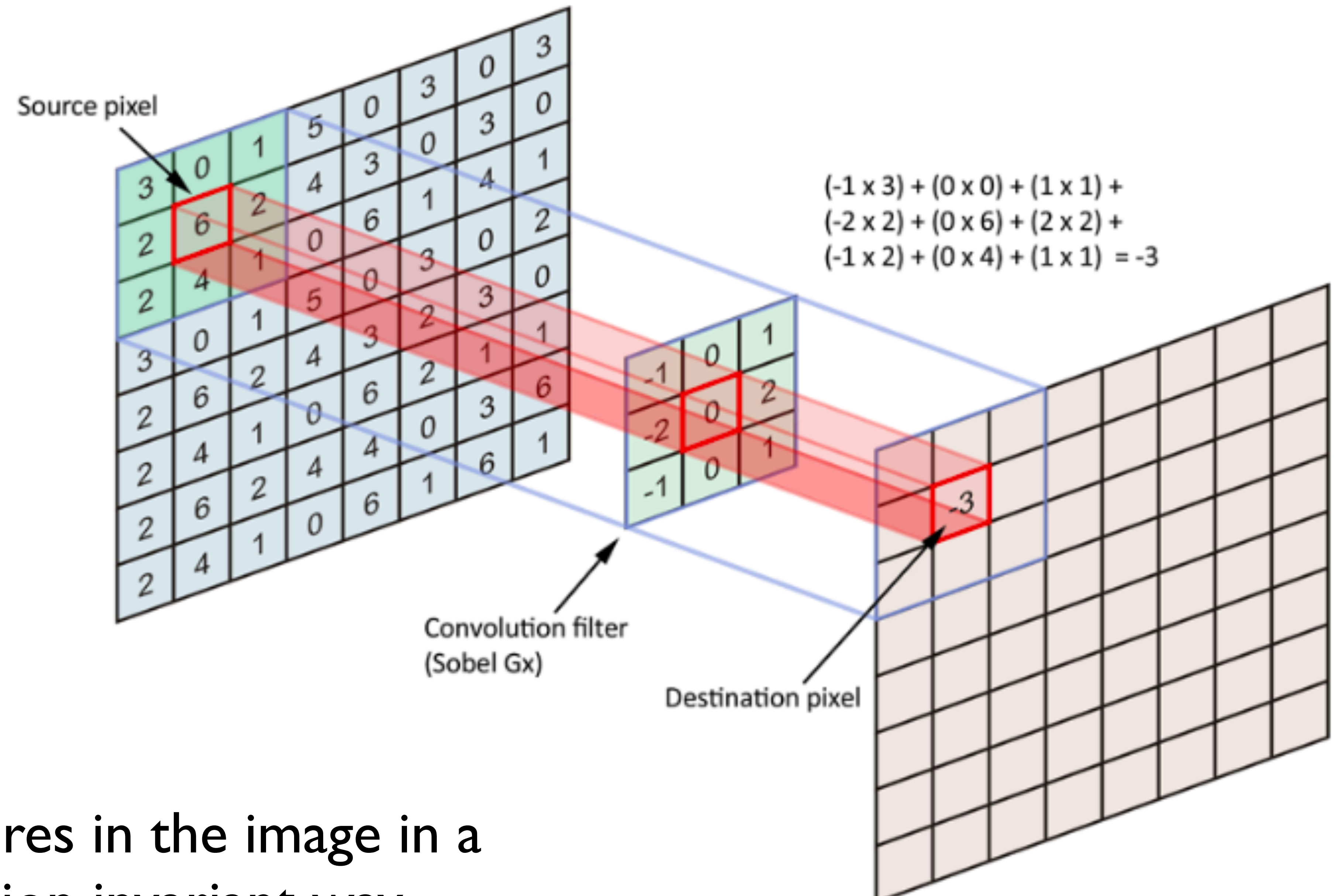
In 1998, the first modern convolutional neural network (CNN) was invented. (LeCun, Bottou, Bengio, Haffner)



"LeNet-5"

Achieved 99% accuracy on database of handwritten digits (MNIST)

Convolutional Layer

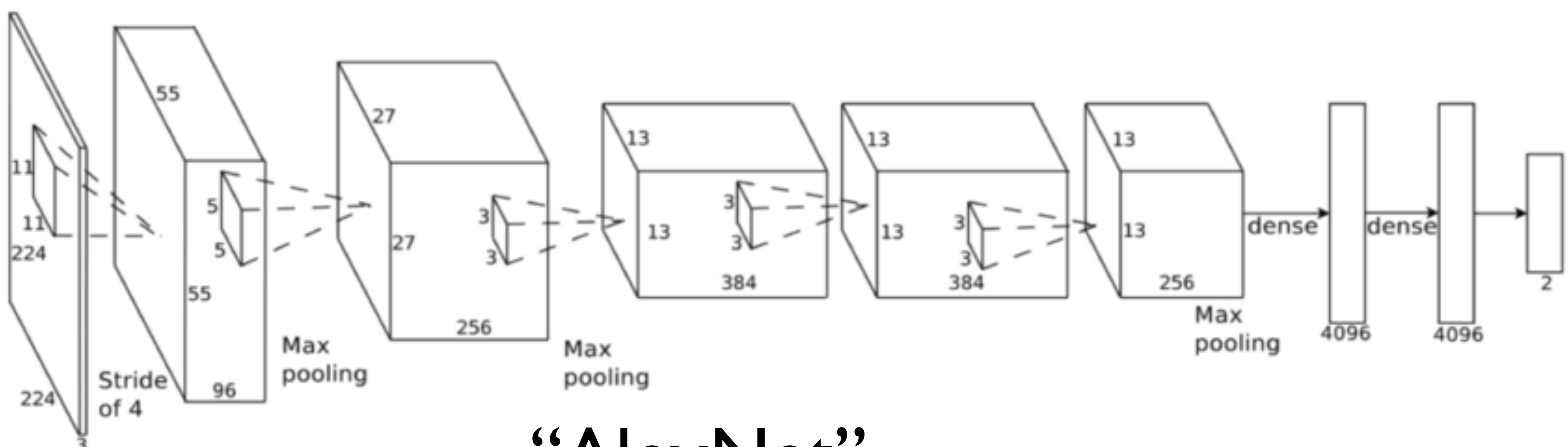


Finds features in the image in a translation invariant way

"feature map"

Deep Learning for Images: Computer Vision

In 2012, a much more powerful CNN won the "ImageNet" image classification competition by a huge margin. This dramatic breakthrough inaugurated the modern revolution in deep learning. (Krizhevsky, Sutskever, Hinton)



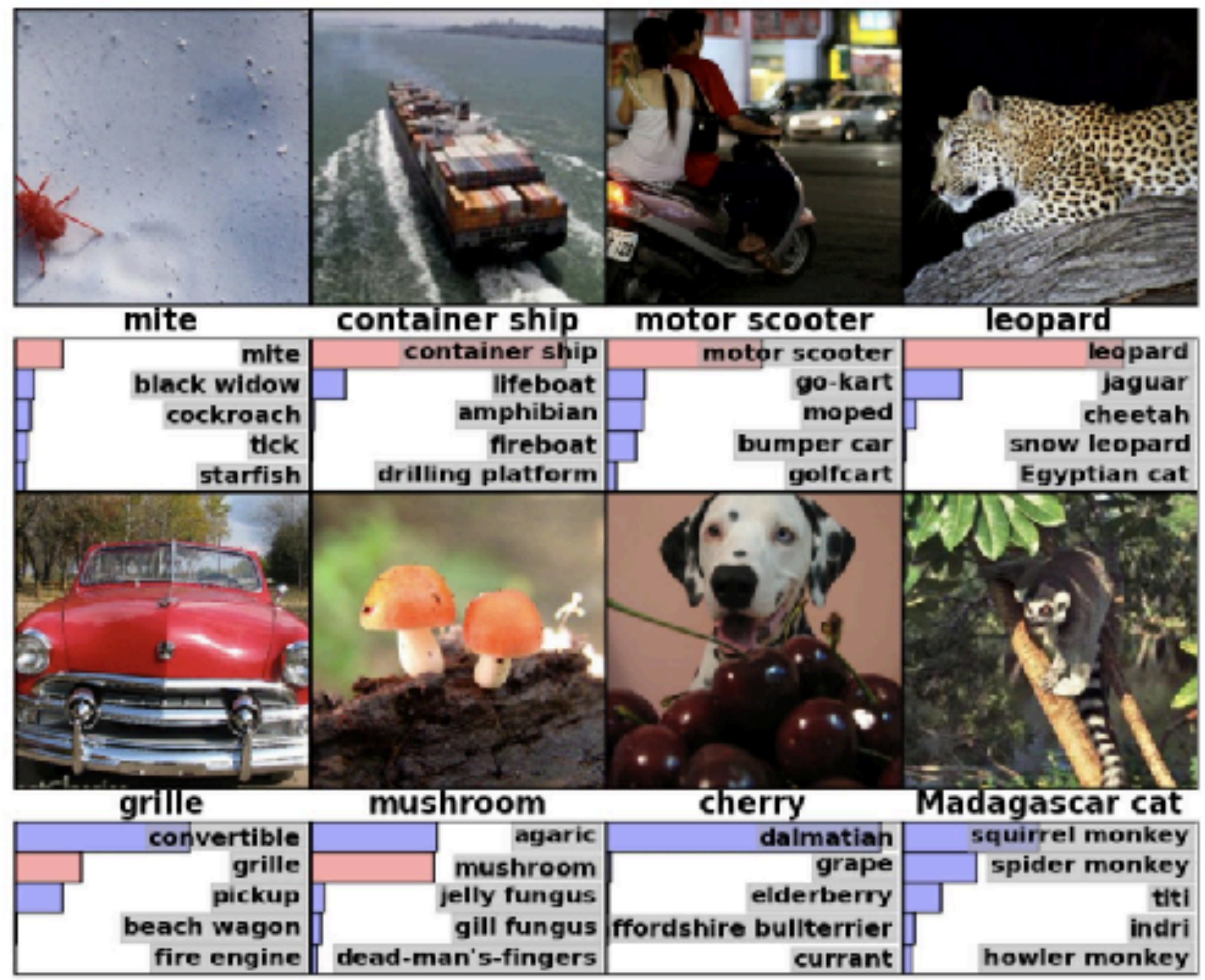
"AlexNet"

# of parameters 1000 x LeNet (60M). Required training on a GPU.

ImageNet Challenge

IMAGENET

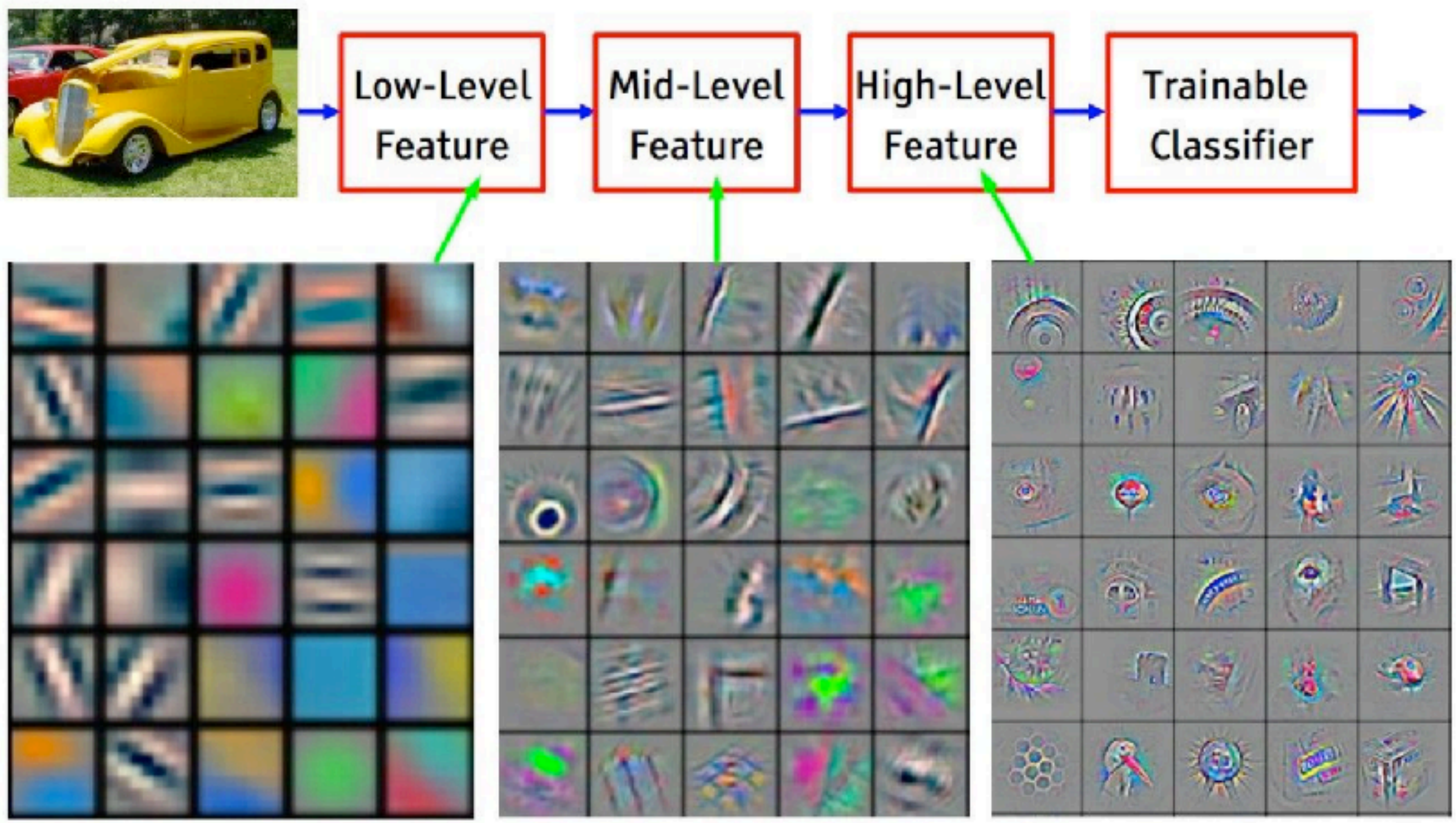
- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.



AlexNet achieved a "top-5" error rate of 15% (next best was 25%).

Many improvements since AlexNet breakthrough.
Current world-best around ~2% (better than humans!)

Automated feature engineering



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Jet Images

Cogan et al 1407.5675

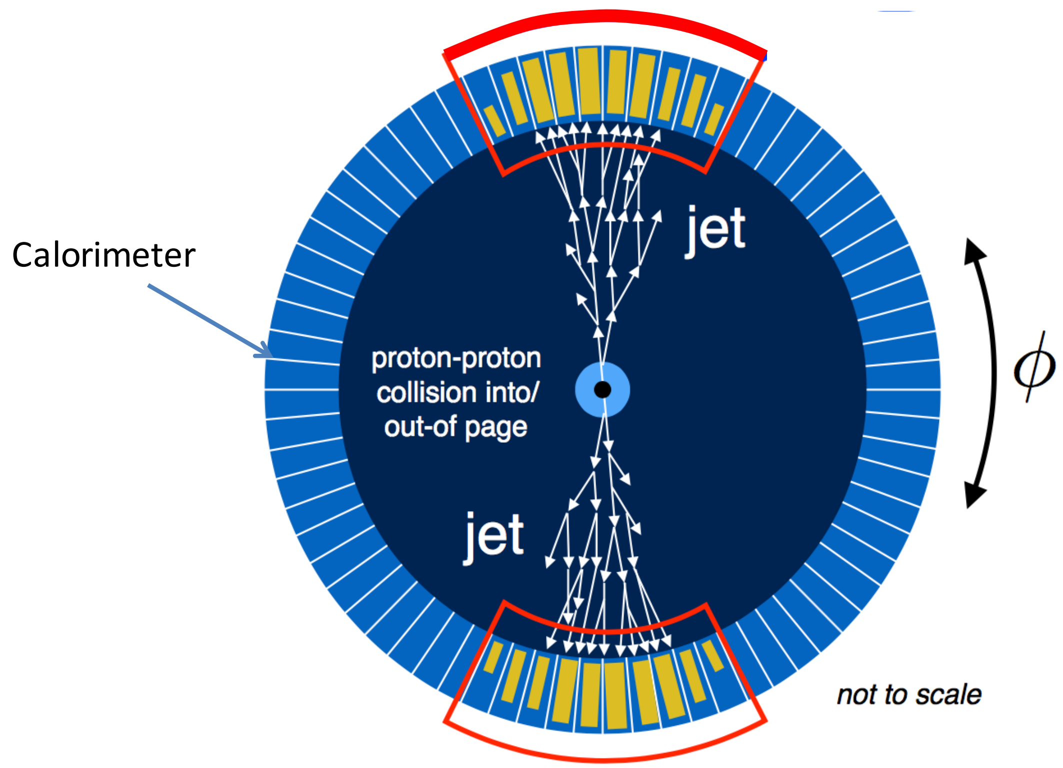
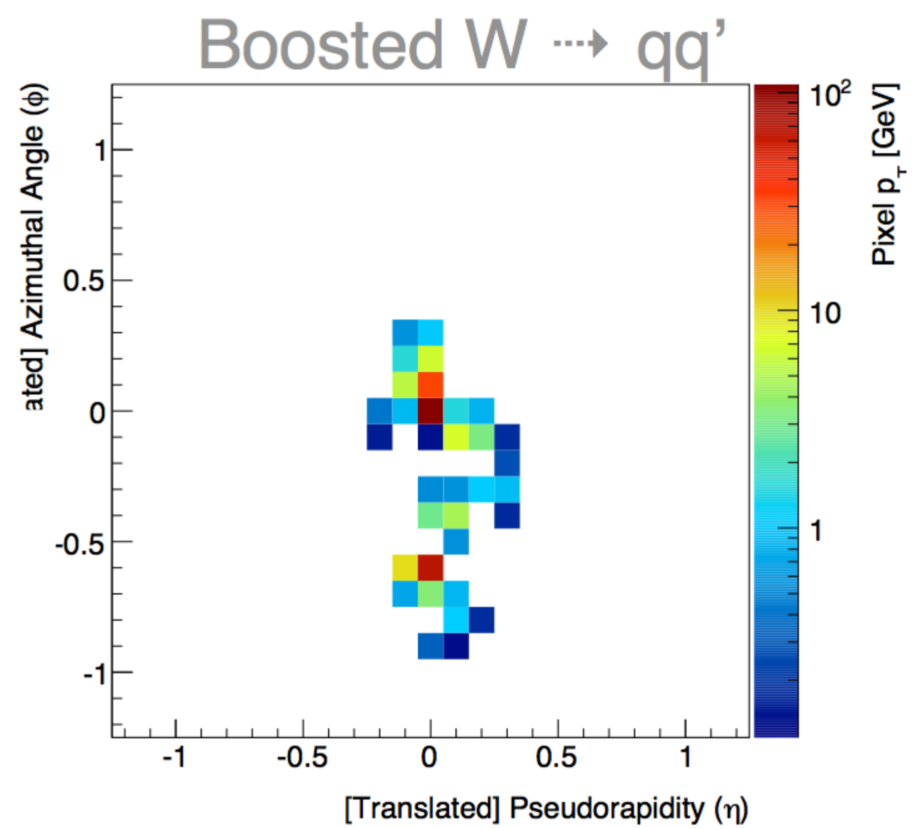


Figure credit:
B. Nachman



Can think of a jet as an image in eta and phi, with

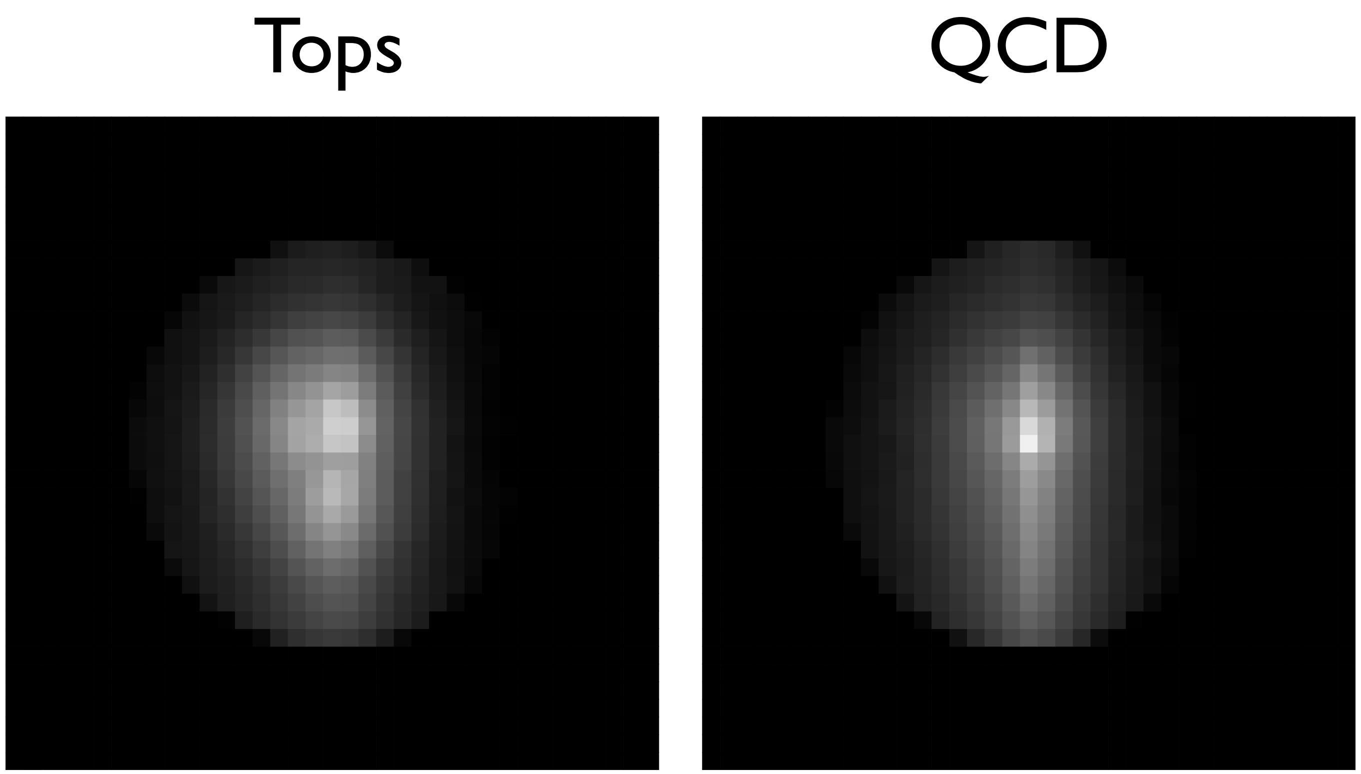
- Pixelation provided by calorimeter towers
- Pixel intensity = pT recorded by each tower

Should be able to apply "off-the-shelf" CNN technology to classify and analyze jets at the LHC! de Oliveira et al 1511.05190

Example: Top Tagging with CNNs

Macaluso & DS 1803.00107

	CMS
Jet sample	13 TeV $p_T \in (800, 900)$ GeV, $\lvert\eta\rvert < 1$ PYTHIA 8 and DELPHES particle-flow match: $\Delta R(t, j) < 0.6$ merge: $\Delta R(t, q) < 0.6$ 1.2M + 1.2M
Image	37×37 $\Delta\eta = \Delta\phi = 3.2$
Colors	$(p_T^{neutral}, p_T^{track}, N_{track}, N_{muon})$

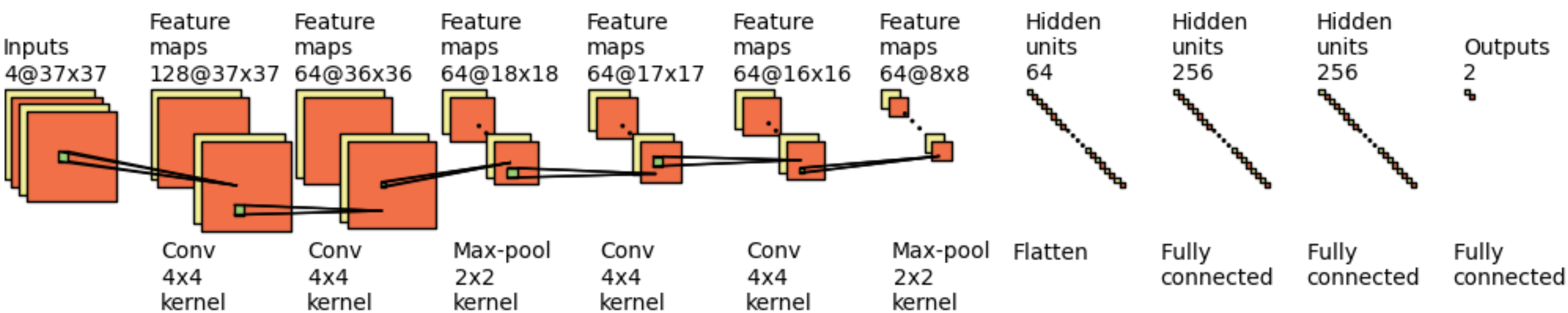


Building on previous "DeepTop" tagger of Kasieczka et al 1701.08784

Other approaches also promising (dense NNs, recursive NNs, recurrent NNs, LSTMs, …)

Example: Top Tagging with CNNs

Macaluso & DS 1803.00107



AdaDelta

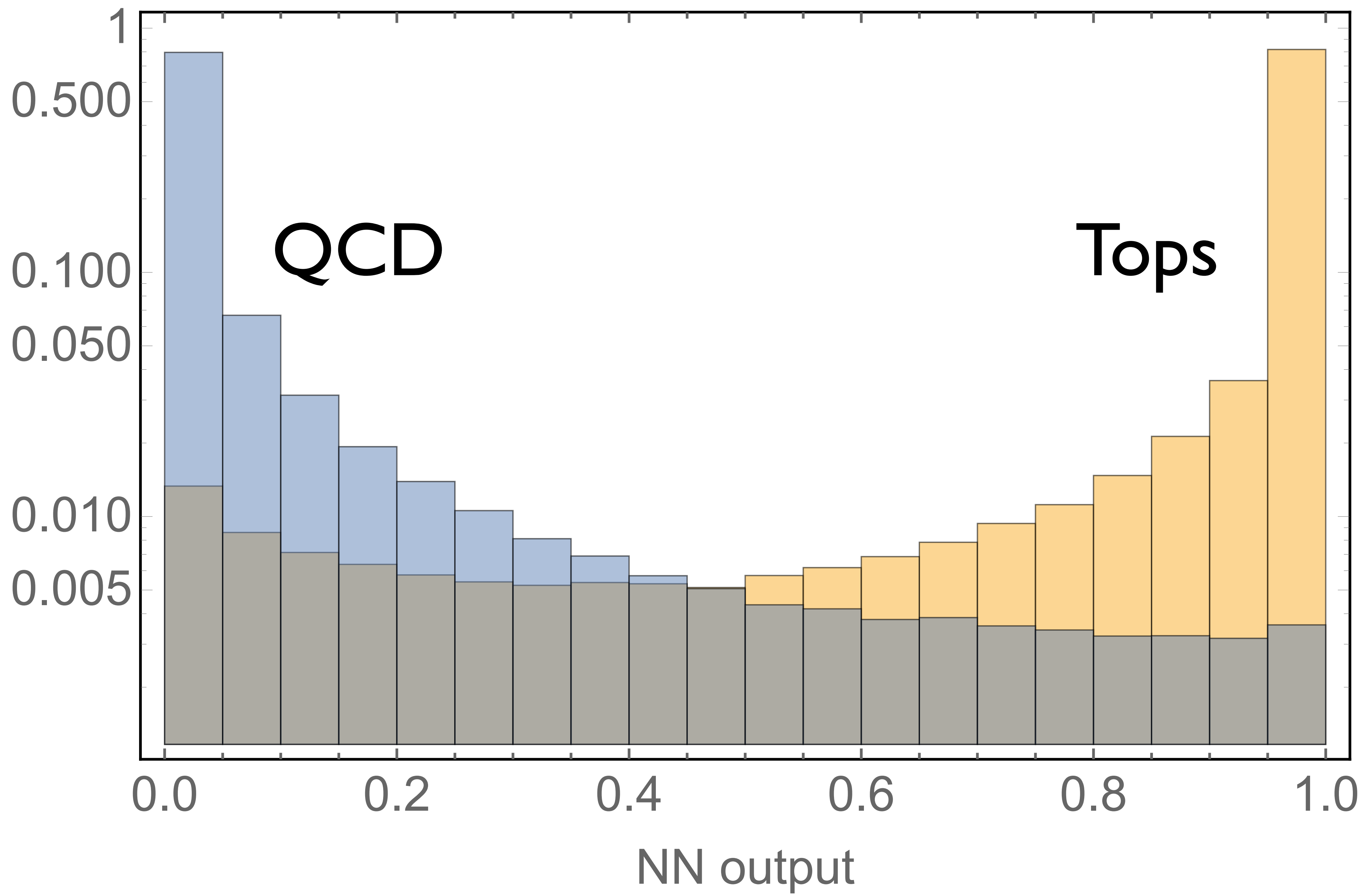
$\eta = 0.3$ with annealing schedule

minibatch size=128

cross entropy loss

Example: Top Tagging with CNNs

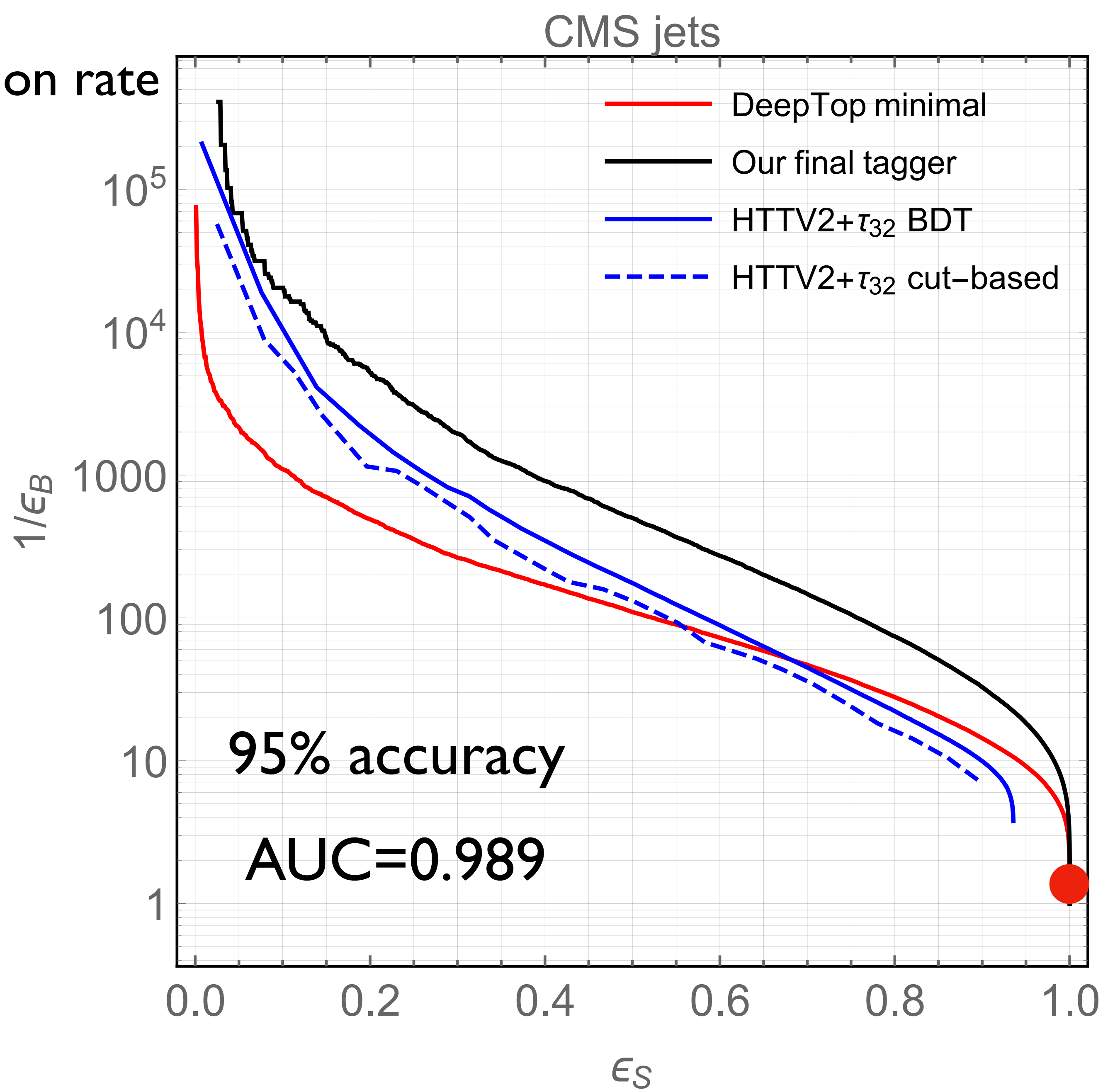
Macaluso & DS 1803.00107



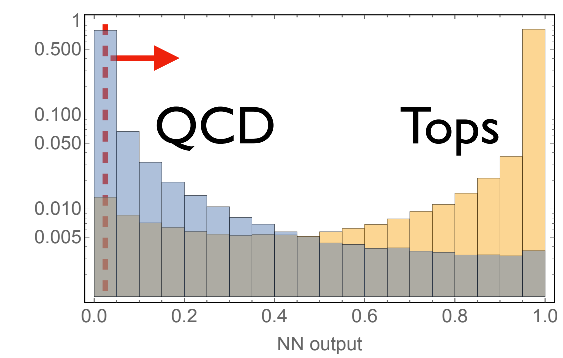
Example: Top Tagging with CNNs

Macaluso & DS 1803.00107

QCD rejection rate



Top tagging efficiency

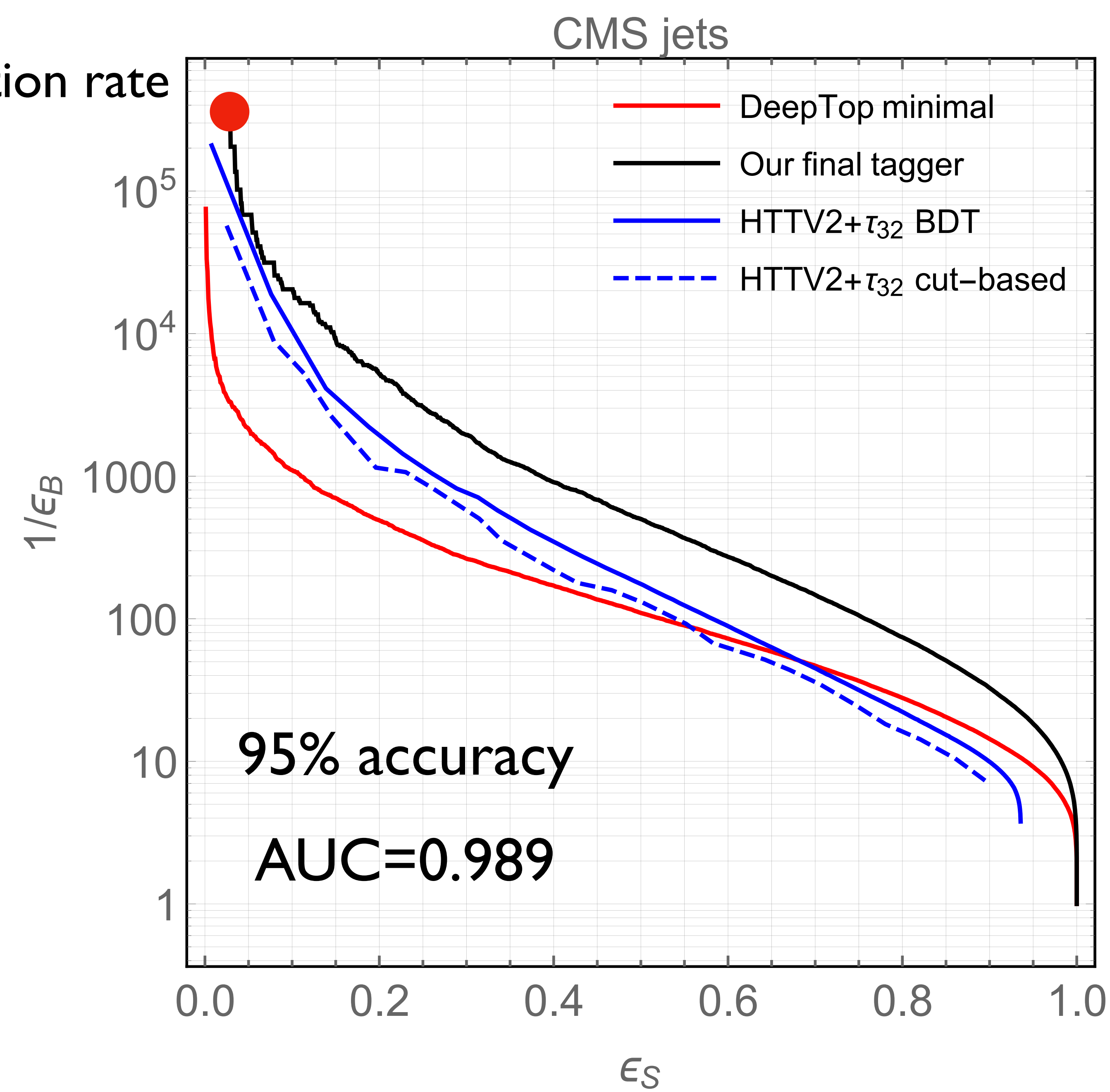


Can achieve factor of ~3 improvement over cut-based approaches and BDTs!

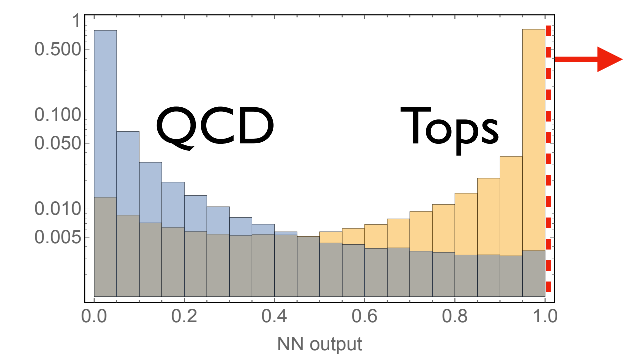
Example: Top Tagging with CNNs

Macaluso & DS 1803.00107

QCD rejection rate



Top tagging efficiency



Can achieve factor of ~3 improvement over cut-based approaches and BDTs!

Supervised vs Unsupervised ML

Top tagging is a prime example of "supervised machine learning" — training with labeled datasets.

Supervised learning is great if you know what you're looking for.

But we are interested in searching for the unexpected.

If data has a small, **unknown** signal in it, can we train a NN to find it?

We need "unsupervised learning": training on unlabeled datasets.

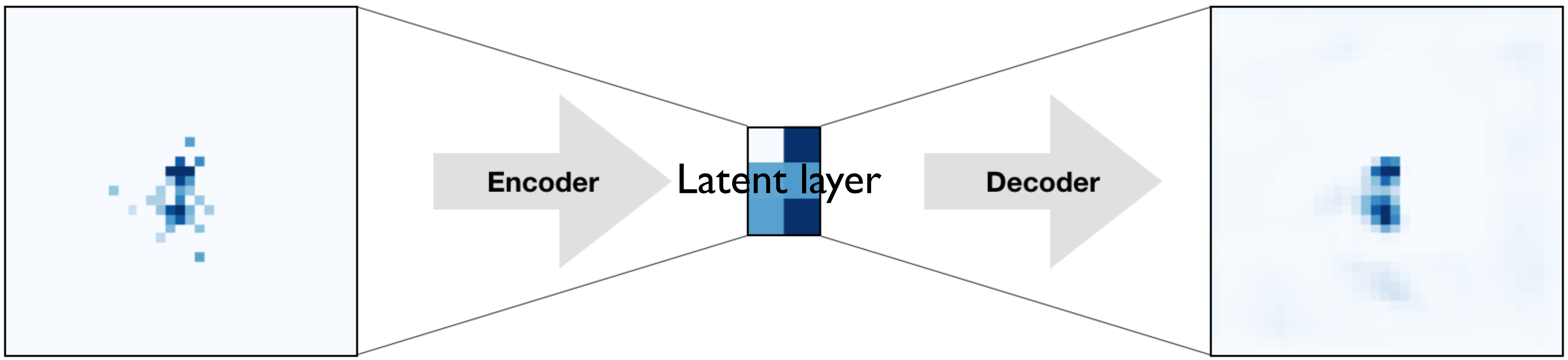
Supervised vs Unsupervised ML

Supervised Learning	Unsupervised Learning
Training on labeled data	Training on unlabeled data
Need separate training set	Train directly on entire input dataset
Used for prediction	Used for analysis
Classification, regression	Clustering, density estimation, dimensionality reduction

A promising idea for anomaly detection:

Autoencoders

Heimel et al 1808.08979;   Farina, Nakai & DS 1808.08992



An autoencoder maps an input into a "latent representation" and then attempts to reconstruct the original input.

The encoding is lossy ("information bottleneck"), so the decoding cannot be perfect.

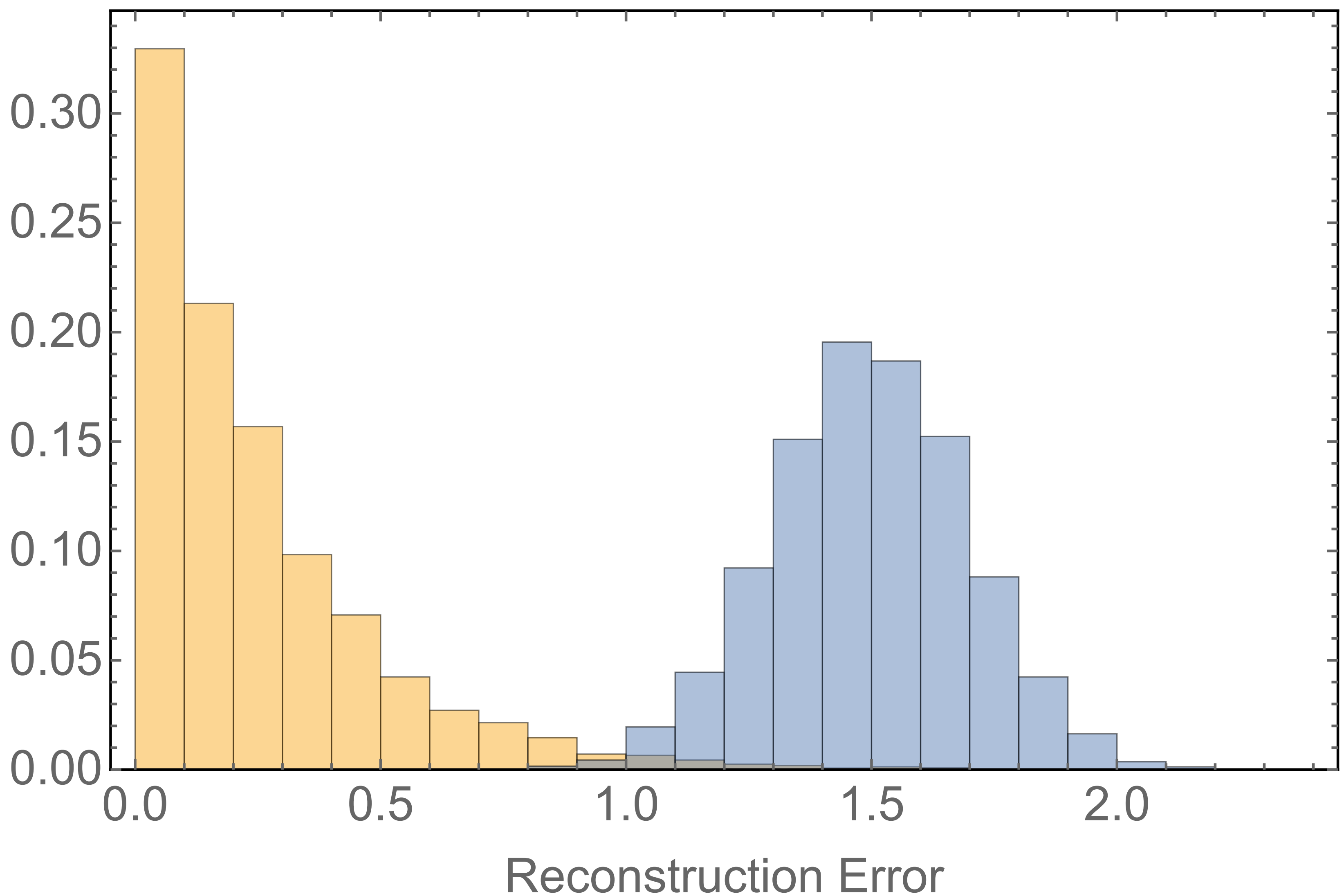
Some previous approaches:
Aguilar-Saavedra et al, "A generic anti-QCD jet tagger" 1709.01087
Collins et al, "CWoLa Hunting" 1805.02664
Hajer et al "Novelty Detection Meets Collider Physics" 1807.10261

Autoencoders

Heimel et al 1808.08979; Farina, Nakai & DS 1808.08992

Quantify AE performance using reconstruction error:

$$L = \frac{1}{N}\sum_{i=1}^{N}(x_i^{in} - x_i^{out})^2$$

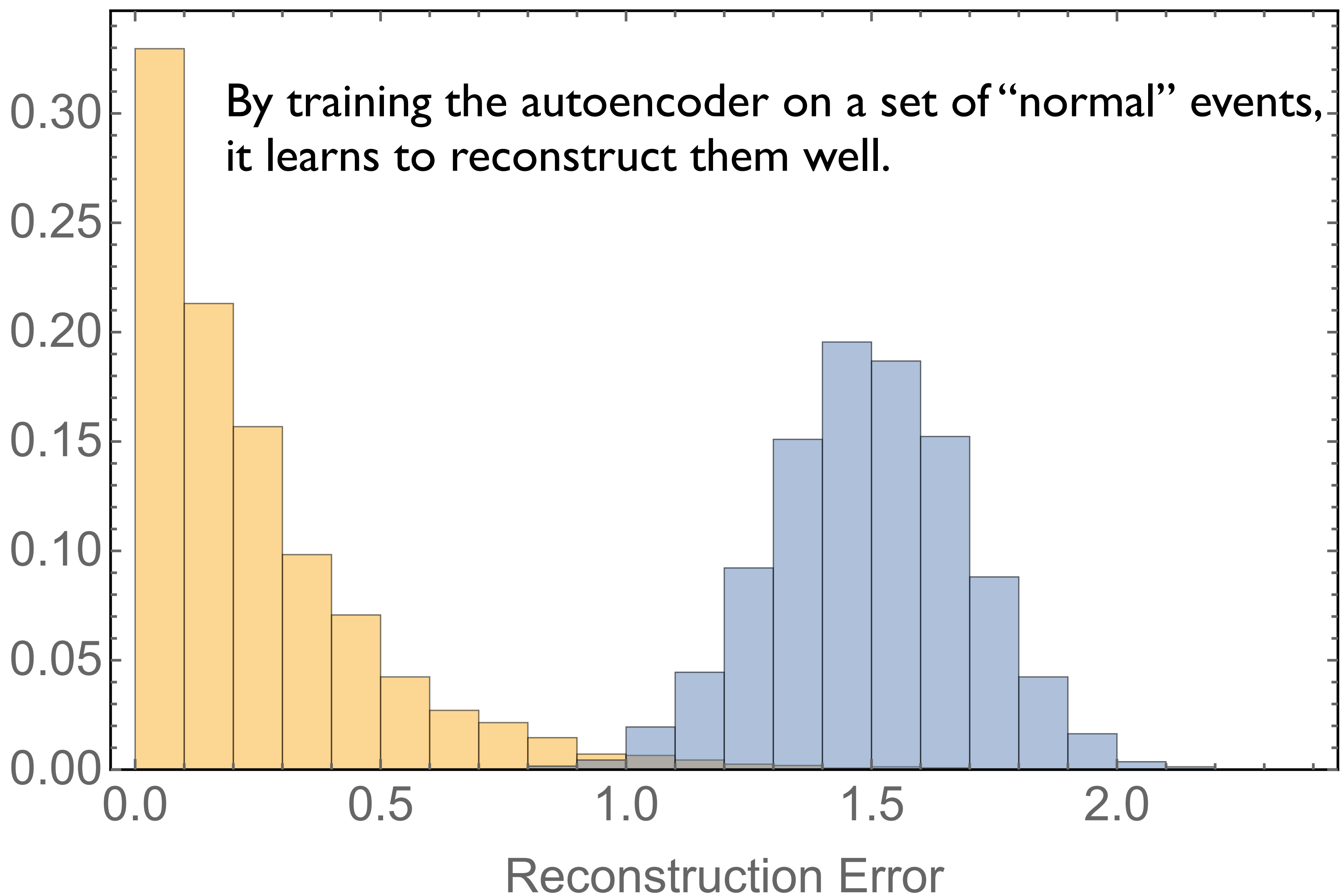


Autoencoders

Heimel et al 1808.08979;   Farina, Nakai & DS 1808.08992

Quantify AE performance using reconstruction error:

$$L = \frac{1}{N}\sum_{i=1}^{N}(x_i^{in} - x_i^{out})^2$$

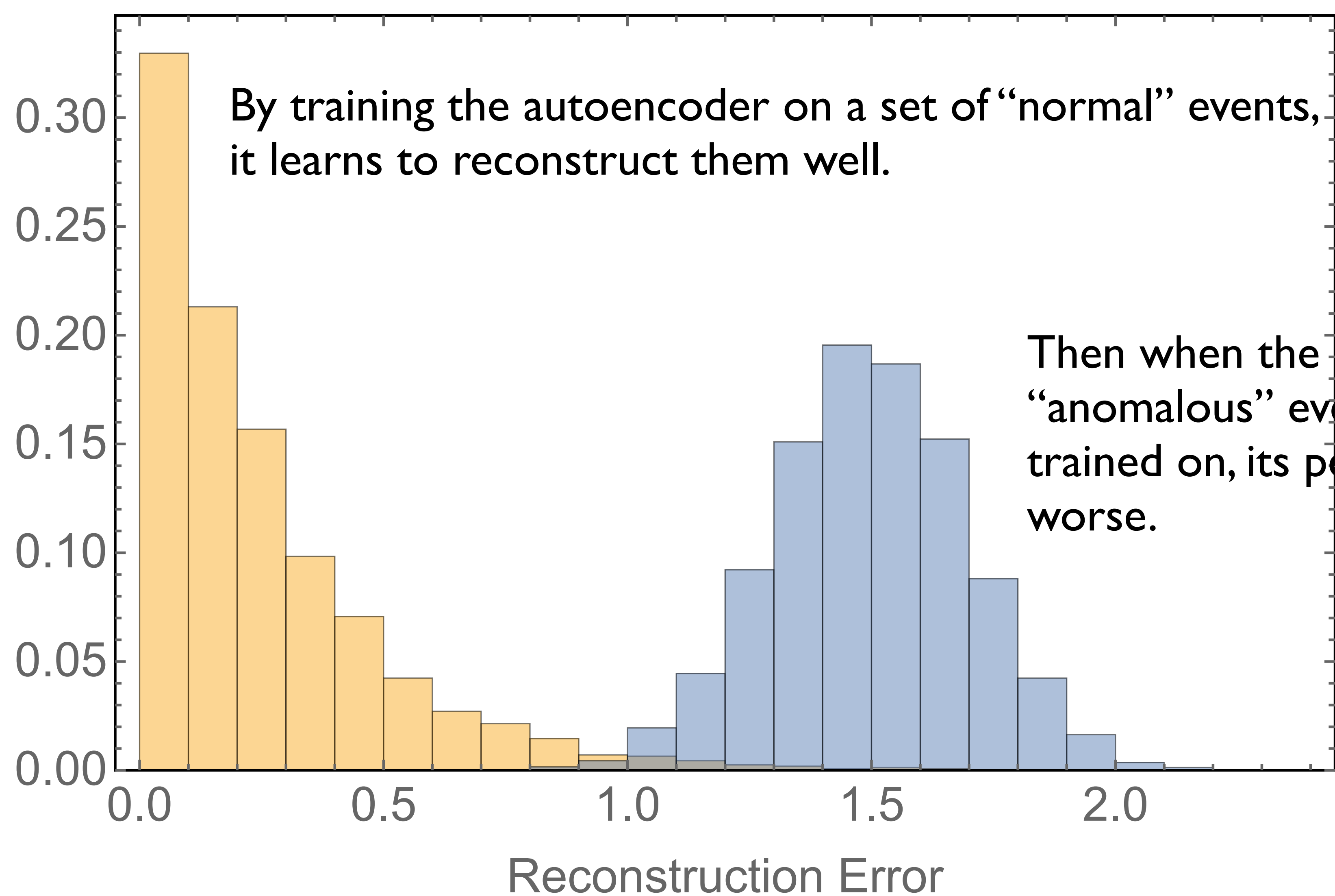


Autoencoders

Heimel et al 1808.08979;   Farina, Nakai & DS 1808.08992

Quantify AE performance using reconstruction error:

$$L = \frac{1}{N}\sum_{i=1}^{N}(x_i^{in} - x_i^{out})^2$$



By training the autoencoder on a set of "normal" events, it learns to reconstruct them well.

Then when the autoencoder encounters "anomalous" events that it was not trained on, its performance should be worse.

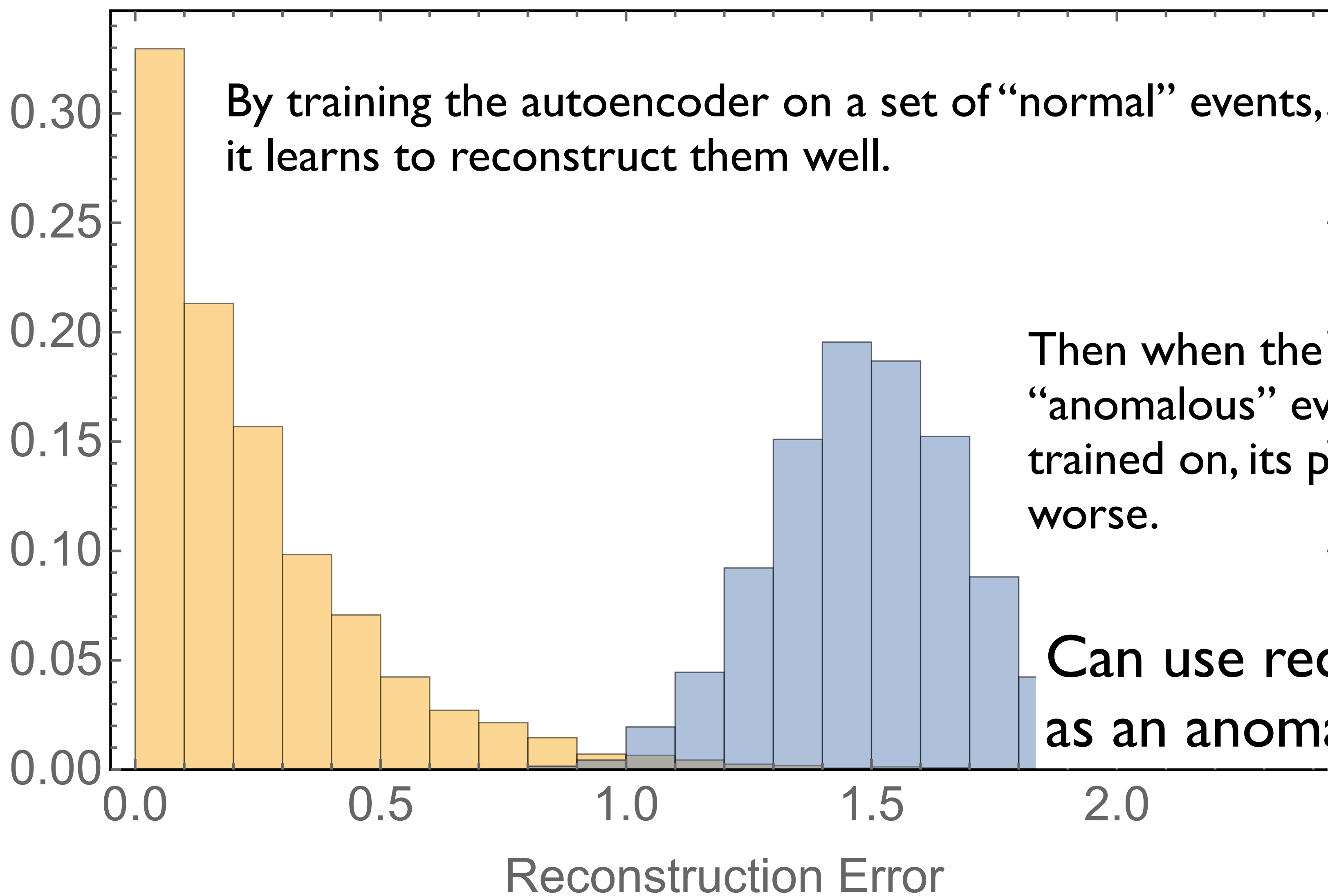
Autoencoders

Heimel et al 1808.08979;   Farina, Nakai & DS 1808.08992

Quantify AE performance using reconstruction error:

$$L = \frac{1}{N}\sum_{i=1}^{N}(x_i^{in} - x_i^{out})^2$$

By training the autoencoder on a set of "normal" events, it learns to reconstruct them well.



Then when the autoencoder encounters "anomalous" events that it was not trained on, its performance should be worse.

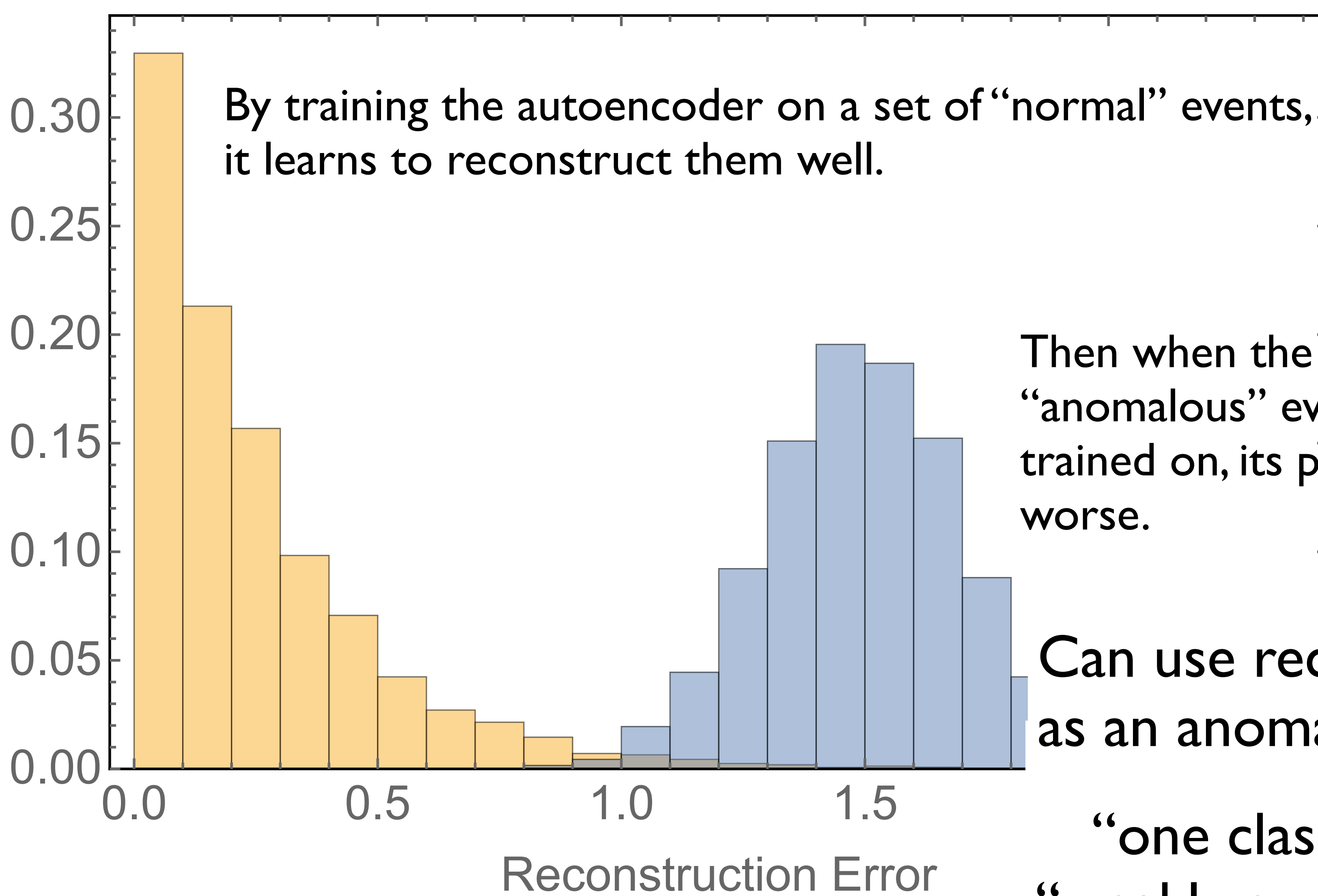
Can use reconstruction error as an anomaly threshold!

Autoencoders

Heimel et al 1808.08979;   Farina, Nakai & DS 1808.08992

Quantify AE performance using reconstruction error:

$$L = \frac{1}{N}\sum_{i=1}^{N}(x_i^{in} - x_i^{out})^2$$



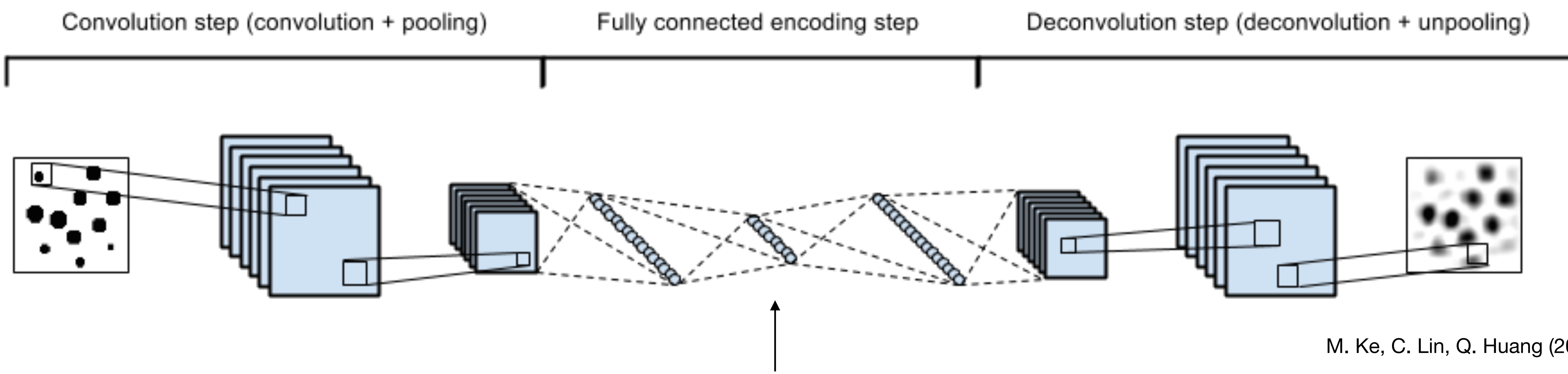
By training the autoencoder on a set of "normal" events, it learns to reconstruct them well.

Then when the autoencoder encounters "anomalous" events that it was not trained on, its performance should be worse.

Can use reconstruction error as an anomaly threshold!

"one class classification"
"weakly-supervised learning"

Autoencoder architecture



M. Ke, C. Lin, Q. Huang (2017)

Encoder **Latent space** **Decoder**

128C3-MP2-128C3-MP2-128C3-32N-6N-32N-12800N-128C3-US2-128C3-US2-1C3

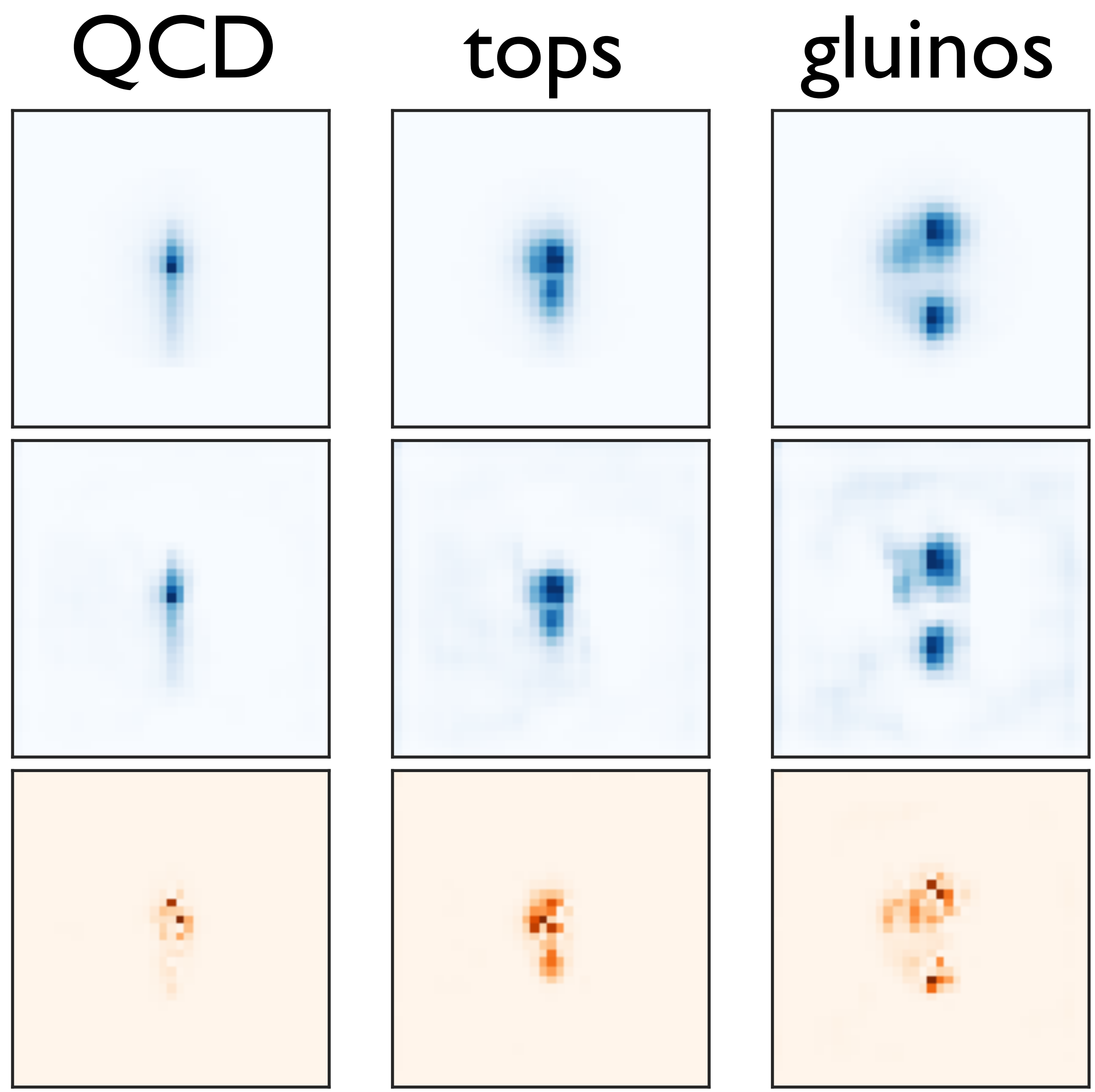
Sample definitions

Same jet specifications as for top tagging study.

We took QCD jets as background, and considered tops and 400 GeV gluinos as signals.

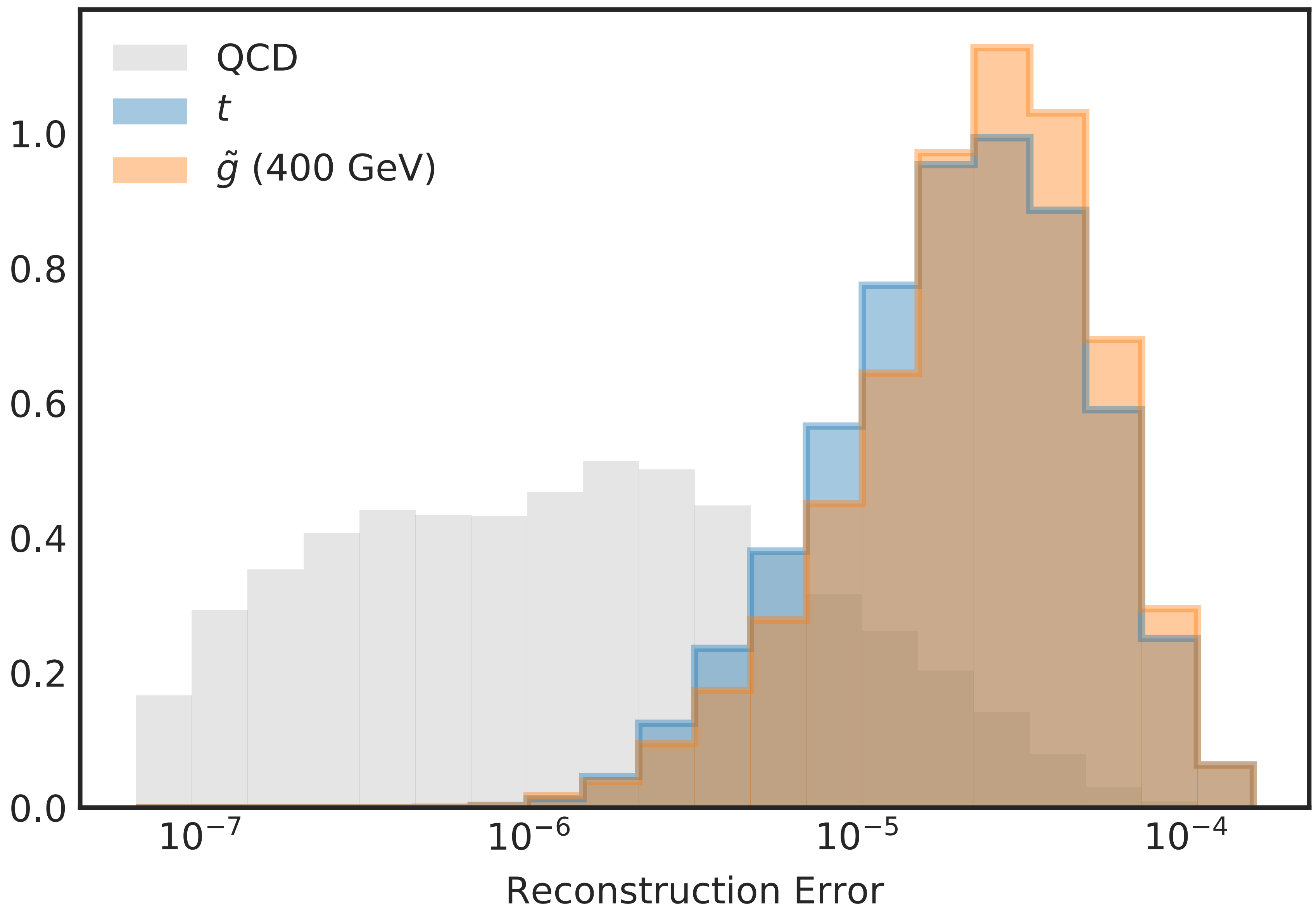
	CMS
Jet sample	13 TeV $p_T \in (800, 900)$ GeV, $\vert\eta\vert < 1$ Pythia 8 and Delphes particle-flow match: $\Delta R(t, j) < 0.6$ merge: $\Delta R(t, q) < 0.6$ 1.2M + 1.2M
Image	37×37 $\Delta\eta = \Delta\phi = 3.2$
Colors	$(p_T^{neutral}, p_T^{track}, N_{track}, N_{muon})$

Performance should be worse on "anomalous" events that autoencoder was not trained on.



The algorithm works when trained on QCD backgrounds!

Can use reconstruction error as an anomaly threshold.

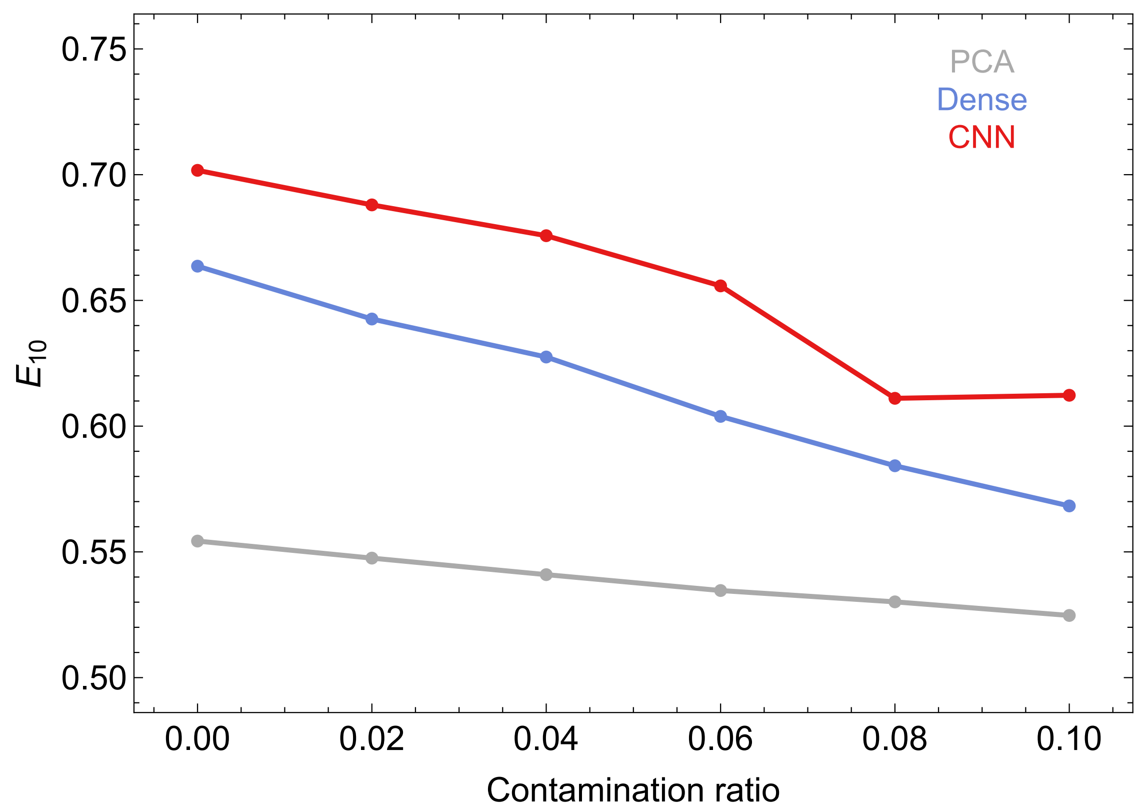


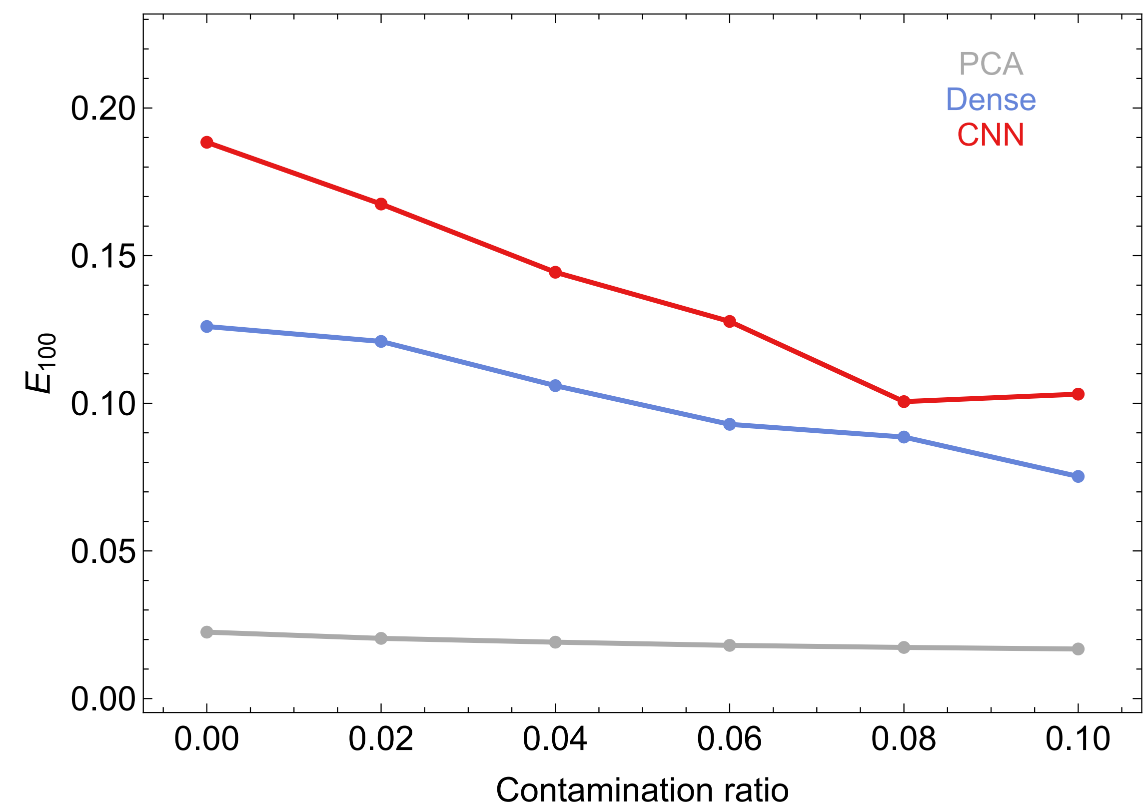
The algorithm works when trained on QCD backgrounds!

Fully unsupervised learning

Train on sample of QCD background "contaminated" with a small fraction of signal.

Representative of actual data.





(E_x = signal efficiency at bg rejection = x)

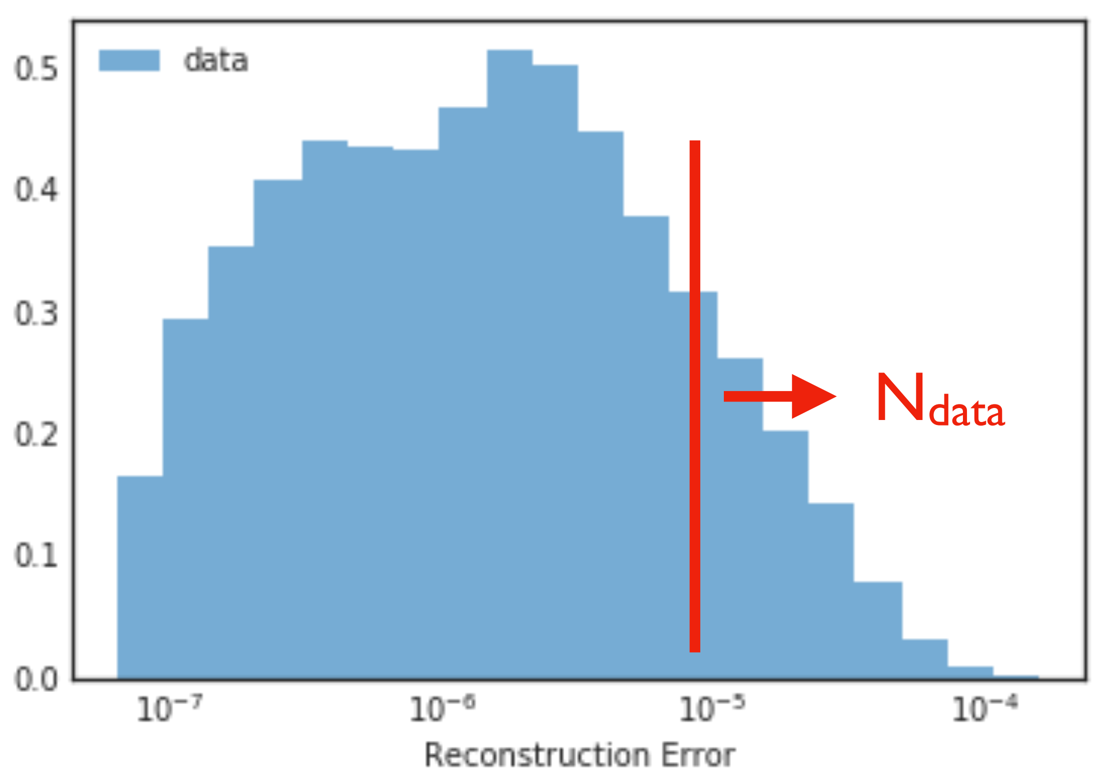
Performance of AE surprisingly robust even up to 10% contamination!

Discovering new physics with an autoencoder

How would one actually discover new physics with an autoencoder?

Need some way of estimating the background. Want it to be data-driven — cannot rely on simulations.

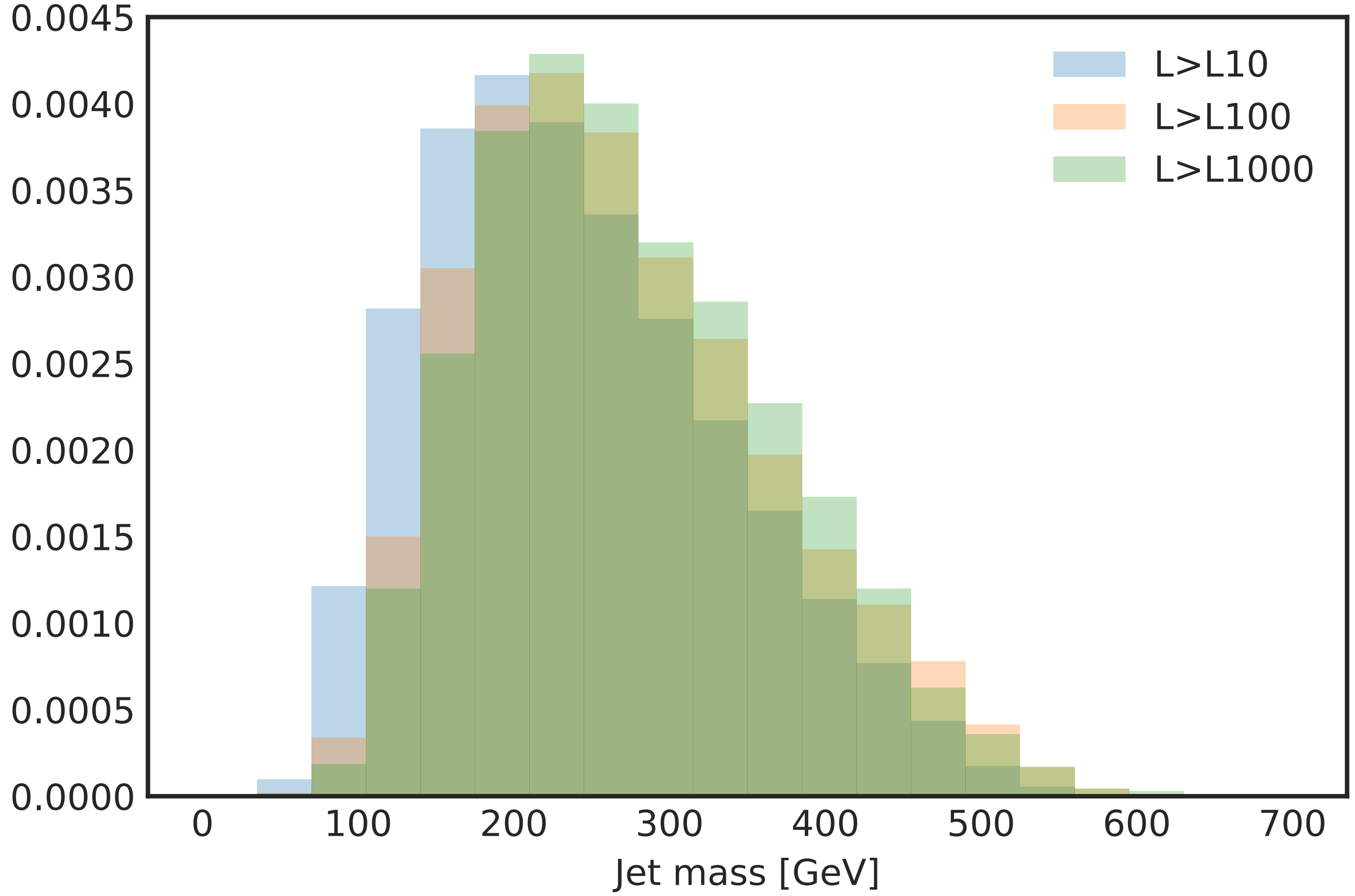
So simply counting number of events above a threshold in reconstruction error is not enough.



N_{bg}??
How do we know if we have an excess?

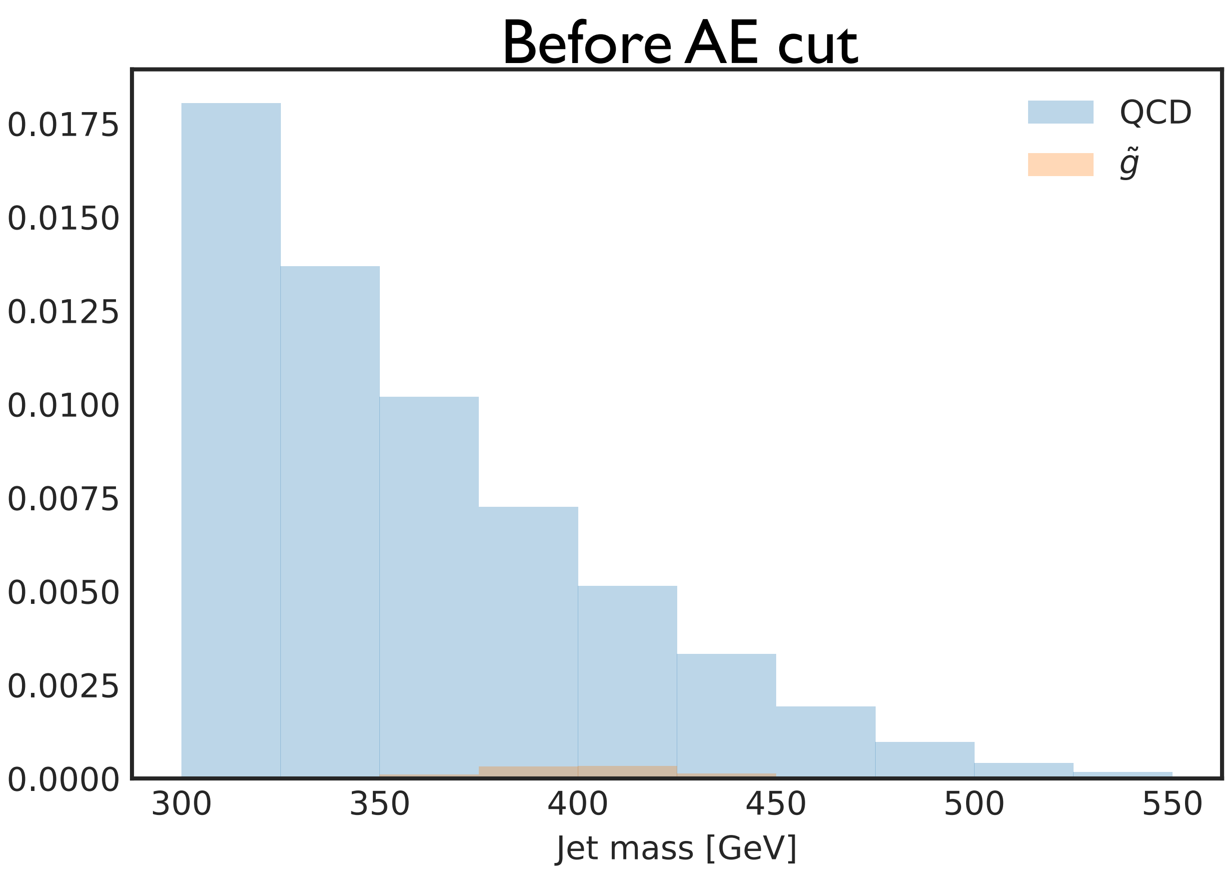
Bump hunt with deep autoencoder

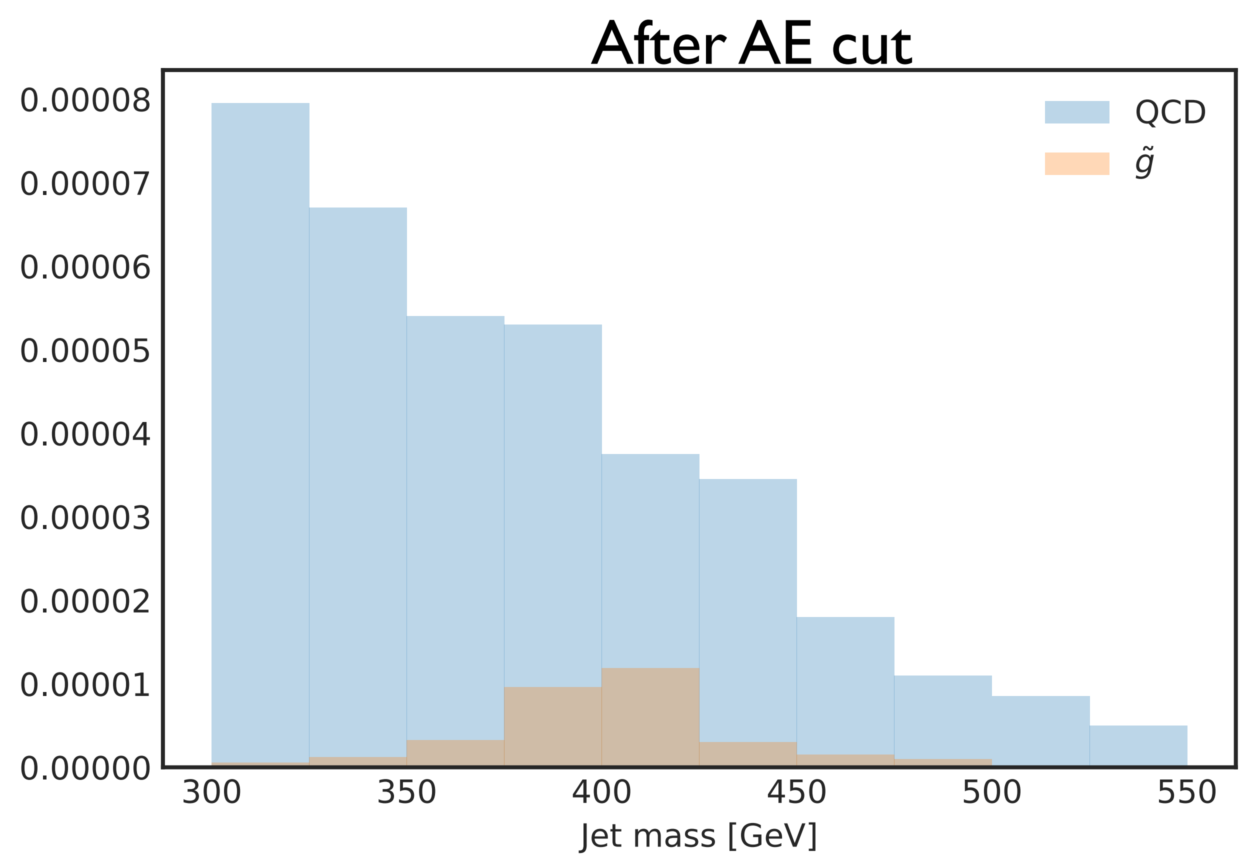
One idea: combine the autoencoder with a bump hunt in jet mass.
Estimate backgrounds using sideband method in jet mass distribution.



Only works if the jet mass distribution is stable against cuts on the reconstruction error!

Bump hunt with deep autoencoder





Train directly on data that contains 400 GeV gluinos.

Use the AE to clean away "boring" QCD jets.

Enhance the bump hunt (improve S/B) by a lot!

Could really discover new physics this way!

Conclusions

Deep learning has revolutionized the field of artificial intelligence and has given birth to a number of stunning real-world applications.

The revolution is coming to high-energy physics.

In this talk, we gave an overview of deep learning and computer vision. We described two applications to HEP:

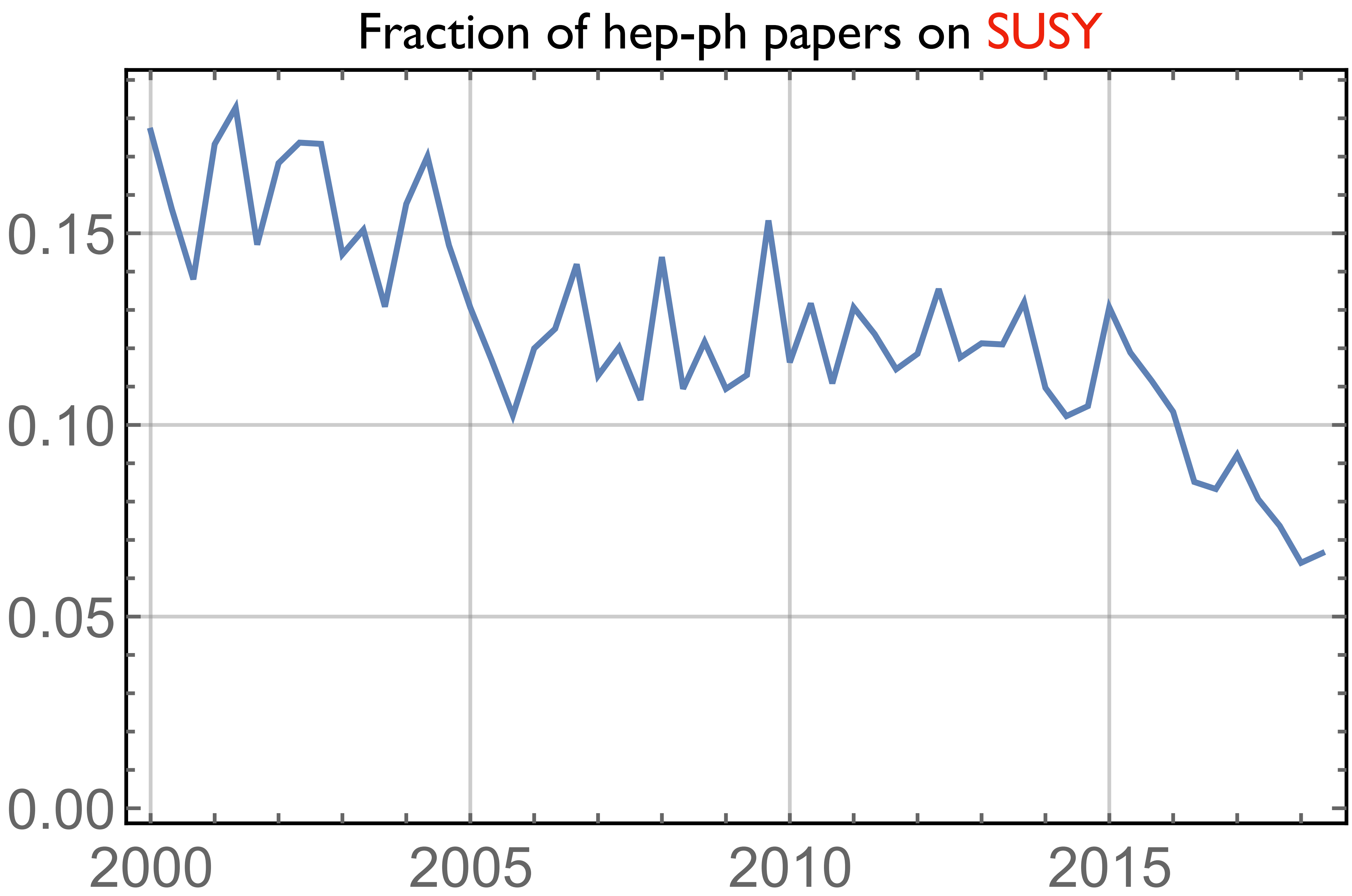
- Top tagging with CNNs (supervised learning)
- Deep autoencoders for anomaly detection (unsupervised learning)

Don't expect unsupervised learning to give better performance than supervised learning — things are always better when you know what you're looking for.

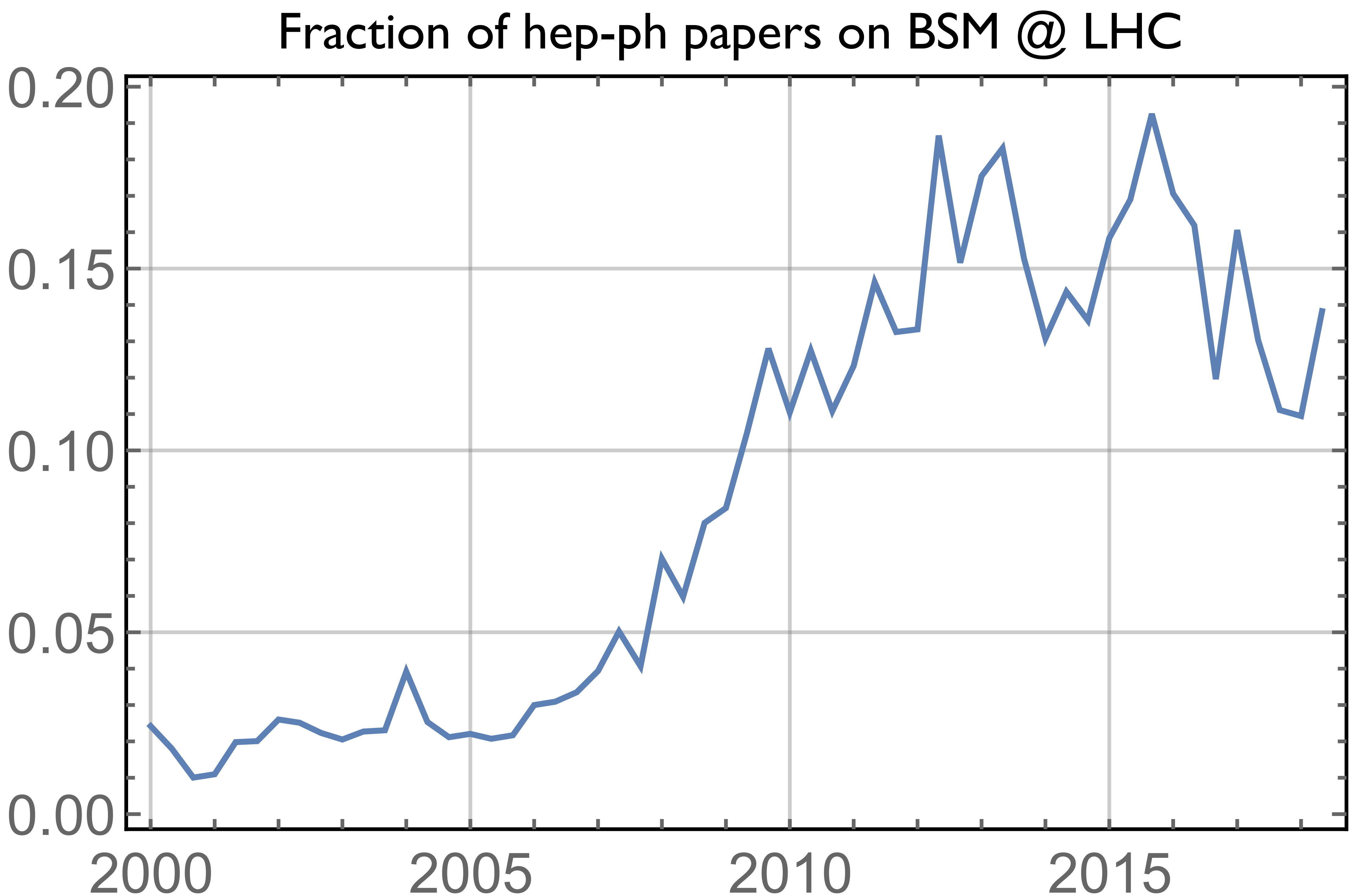
But unsupervised learning gives us the hope of discovering something new and unexpected! We need more ideas like this!

Backup material

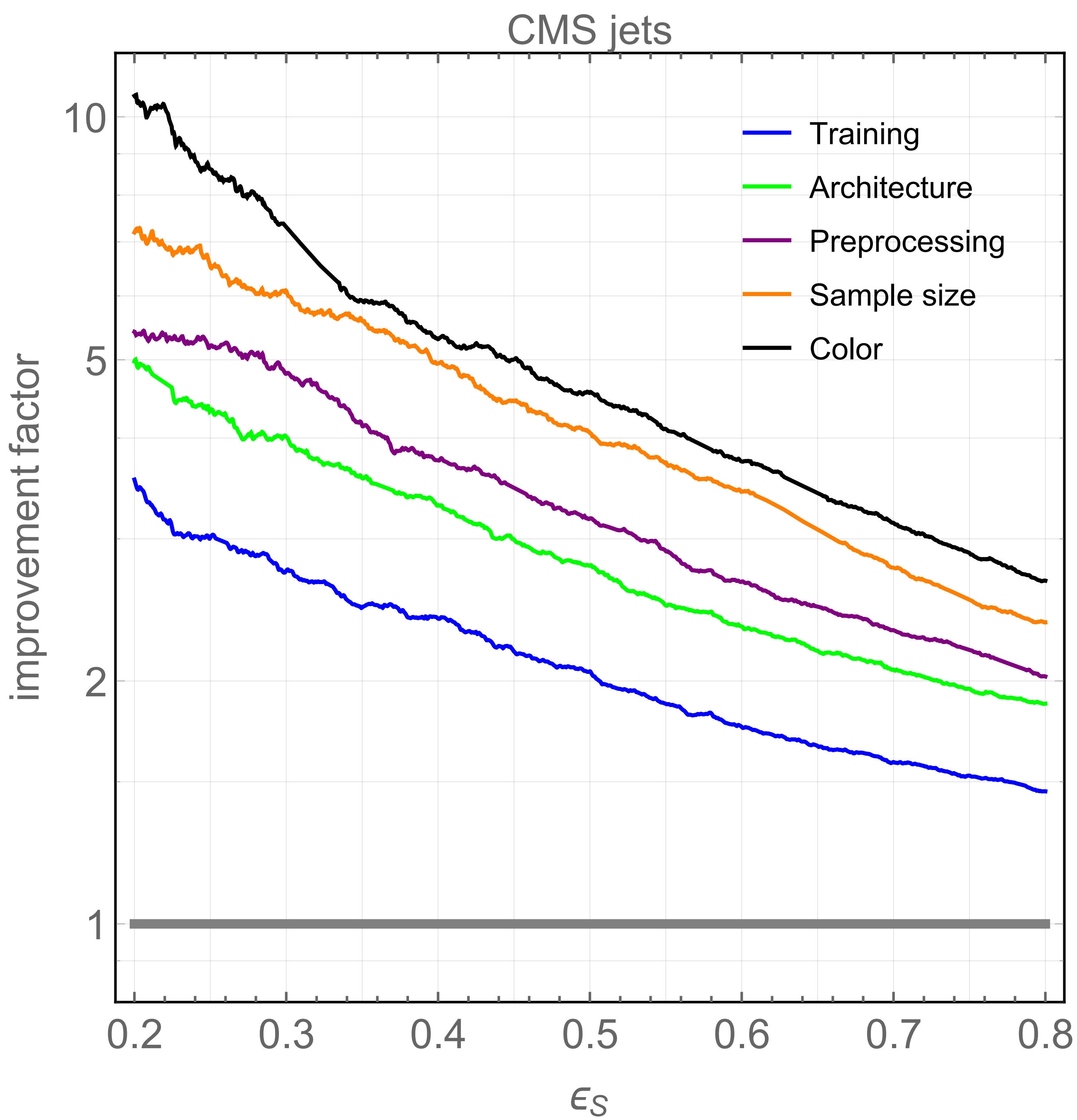
Theorists are losing interest…



Theorists are losing interest…



CNN Top Tagger Details

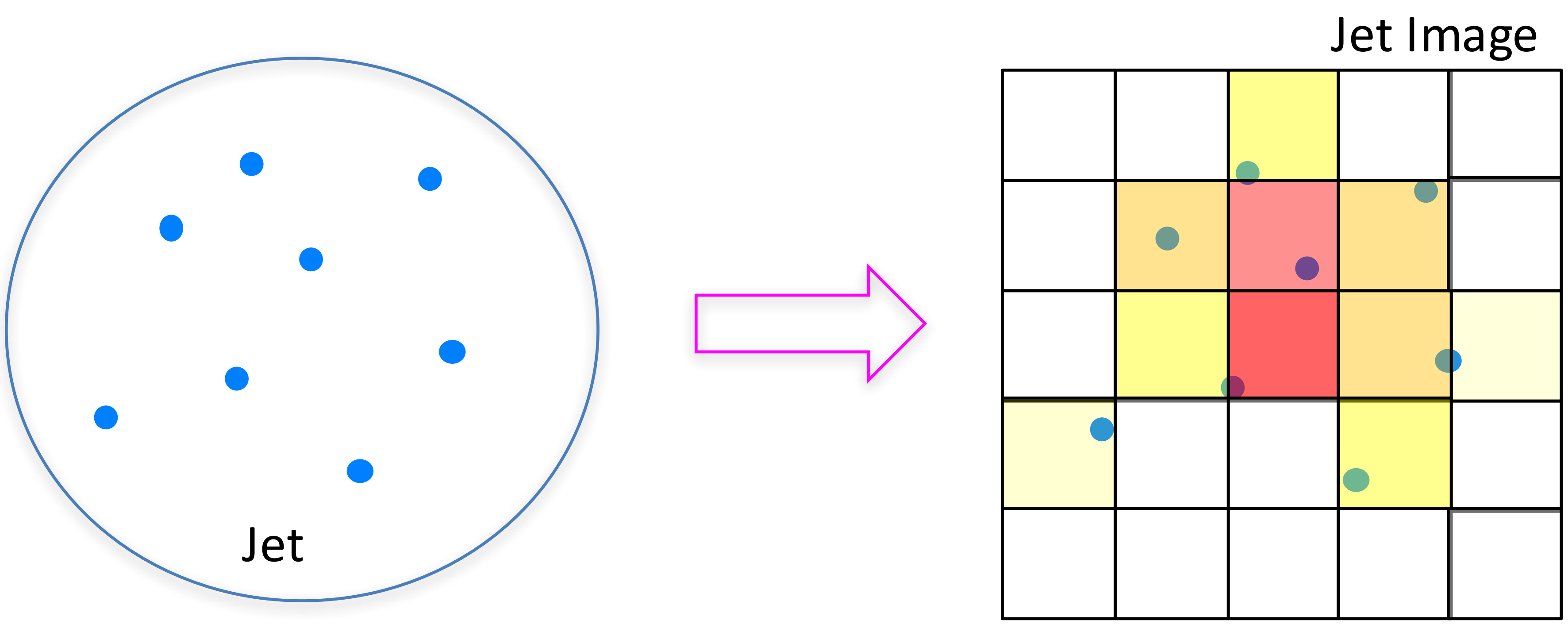


CNN Top Tagger Details

	DeepTop minimal	Our final tagger
Training	SGD $\eta = 0.003$ minibatch size=1000 MSE loss	AdaDelta $\eta = 0.3$ with annealing schedule minibatch size=128 cross entropy loss
CNN architecture	8C4-8C4-MP2-8C4-8C4- 64N-64N-64N	128C4-64C4-MP2-64C4-64C4-MP2- 64N-256N-256N
Preprocessing	pixelate$\rightarrow$center $\rightarrow$ normalize	center$\rightarrow$rotate$\rightarrow$flip $\rightarrow$ normalize$\rightarrow$pixelate
Sample size	150k+150k	1.2M+1.2M
Color	$p_T^{calo} = p_T^{neutral} + p_T^{track}$	$(p_T^{neutral},\, p_T^{track},\, N_{track},\, N_{muon})$

	t	$\tilde{g}$
PCA	0.51 / 0.04	0.98 / 0.36
Dense	0.66 / 0.13	0.90 / 0.39
CNN	0.70 / 0.19	0.77 / 0.23

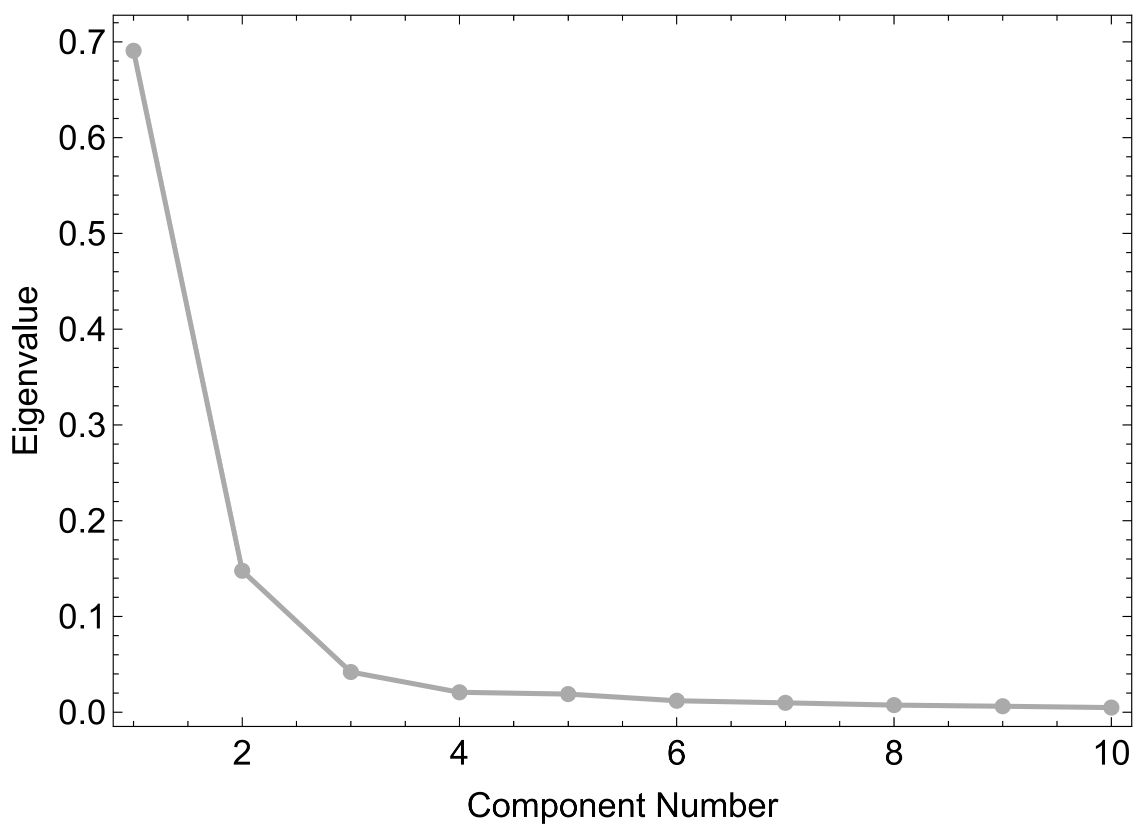
Jets as images

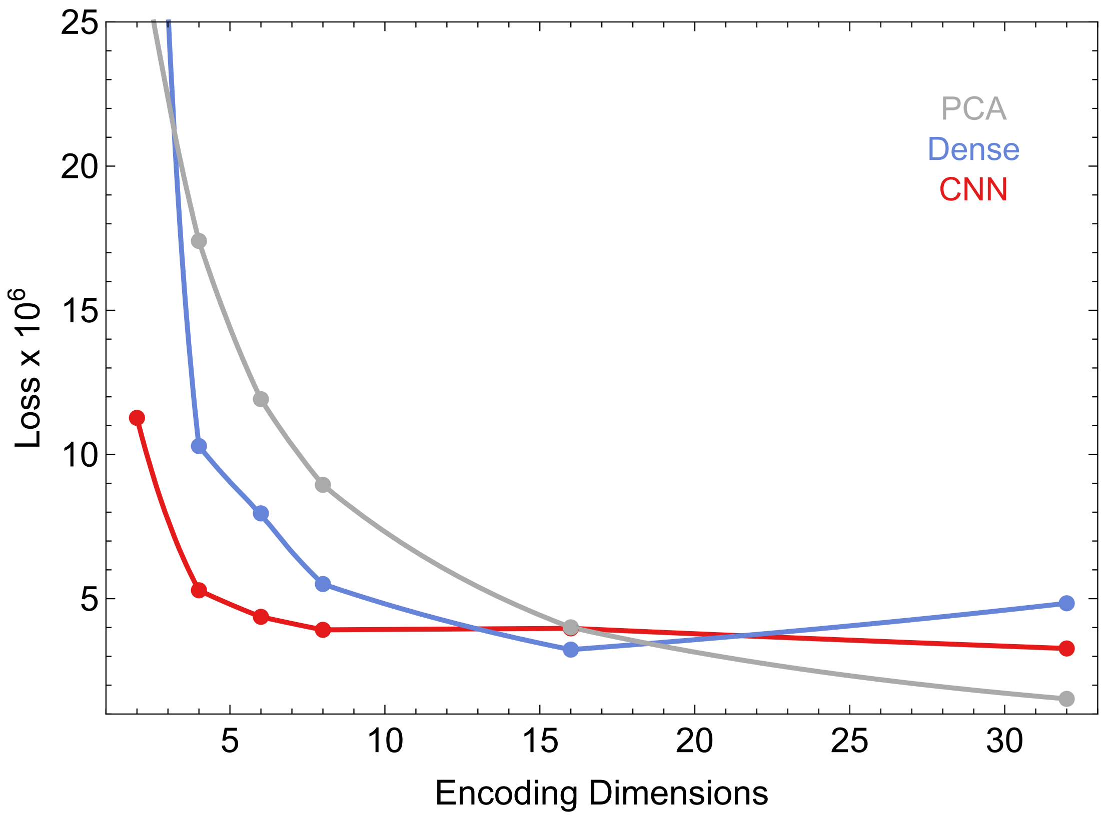


Choosing the latent dimension

d too large → autoencoder becomes identity transform
d too small → autoencoder cannot learn all the features

Should choose the latent dimension in an unsupervised manner
(ie without optimizing on a specific signal)



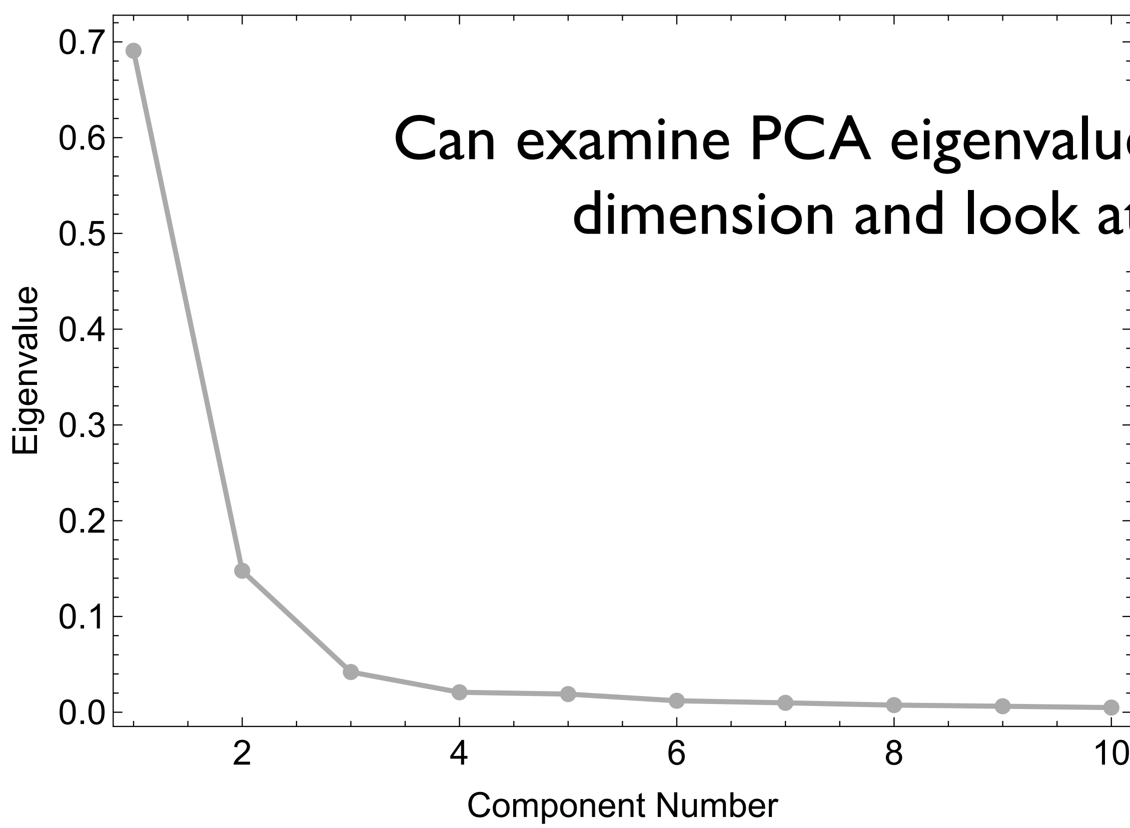


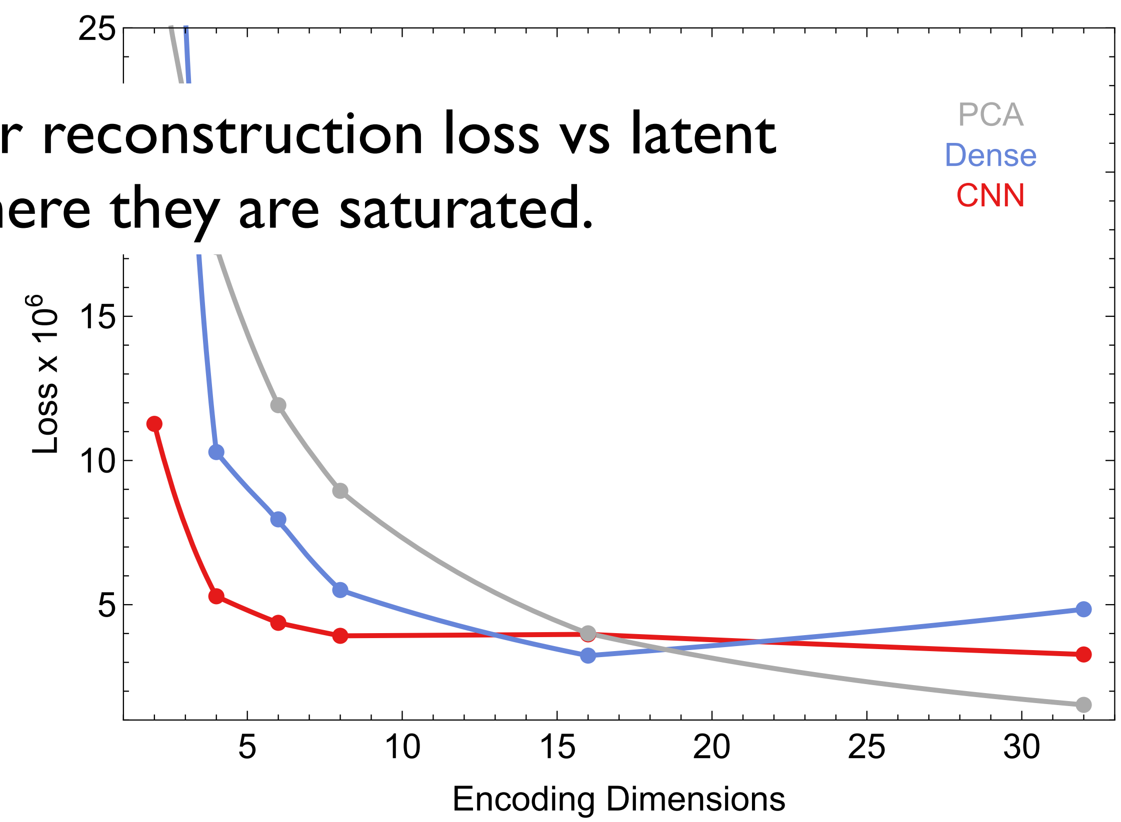
Choosing the latent dimension

d too large → autoencoder becomes identity transform
d too small → autoencoder cannot learn all the features

Should choose the latent dimension in an unsupervised manner
(ie without optimizing on a specific signal)

Can examine PCA eigenvalues or reconstruction loss vs latent dimension and look at where they are saturated.



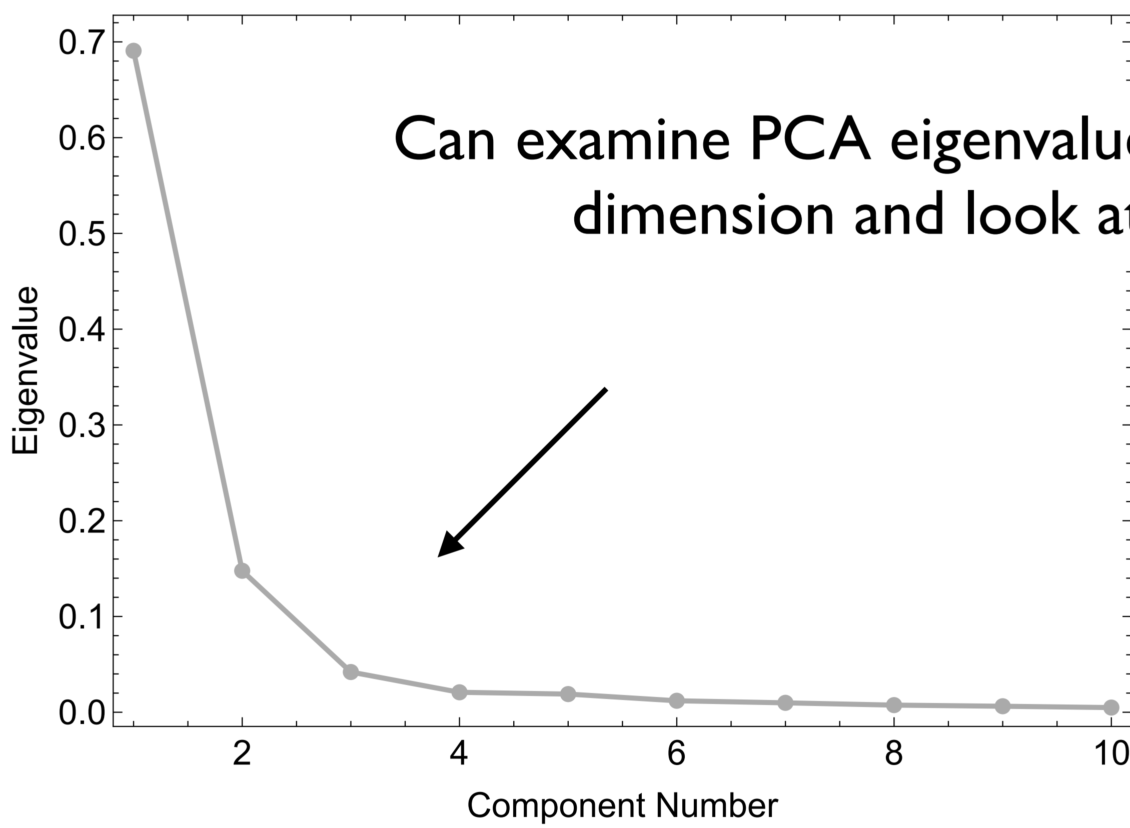


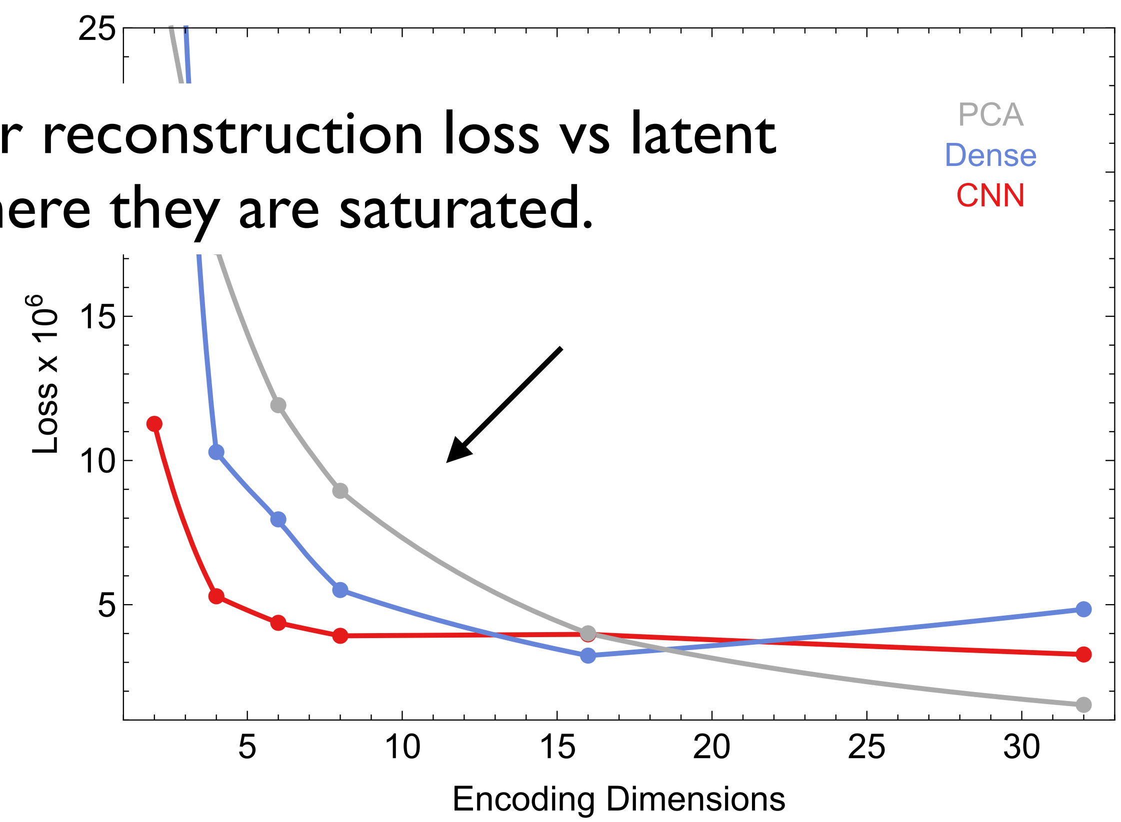
Choosing the latent dimension

d too large → autoencoder becomes identity transform
d too small → autoencoder cannot learn all the features

Should choose the latent dimension in an unsupervised manner
(ie without optimizing on a specific signal)

Can examine PCA eigenvalues or reconstruction loss vs latent dimension and look at where they are saturated.



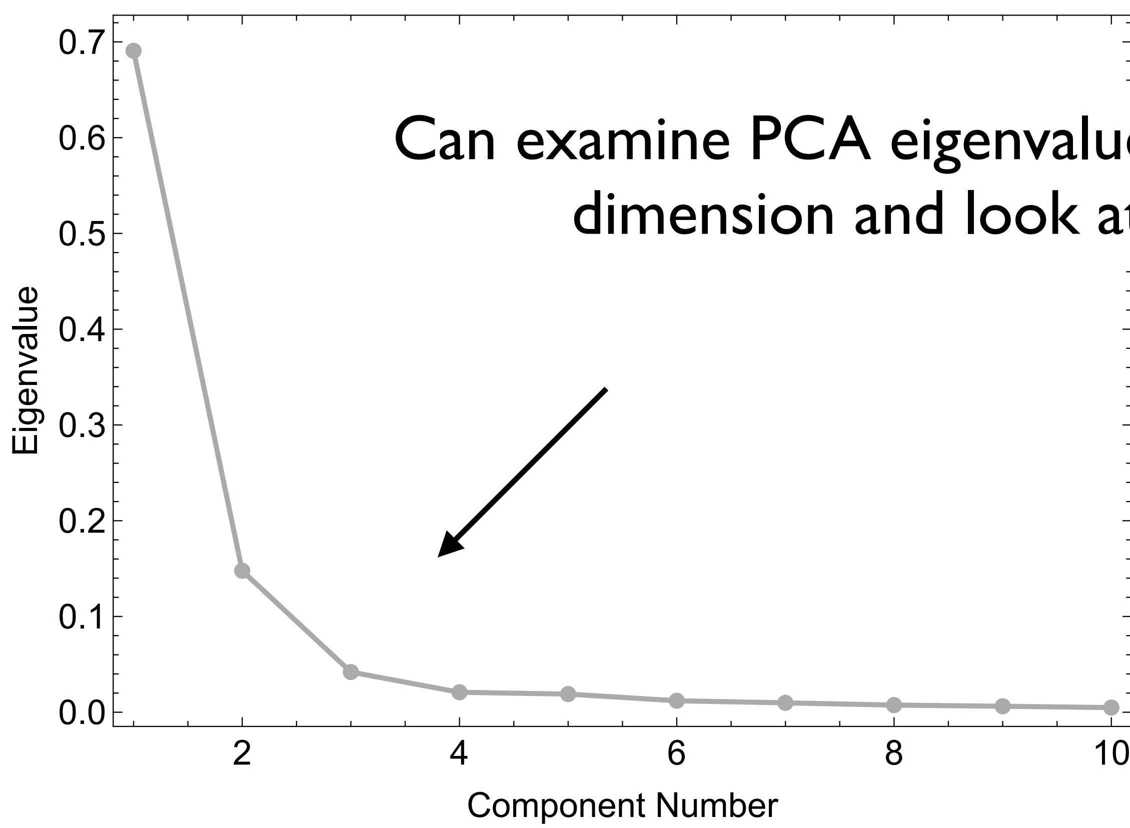


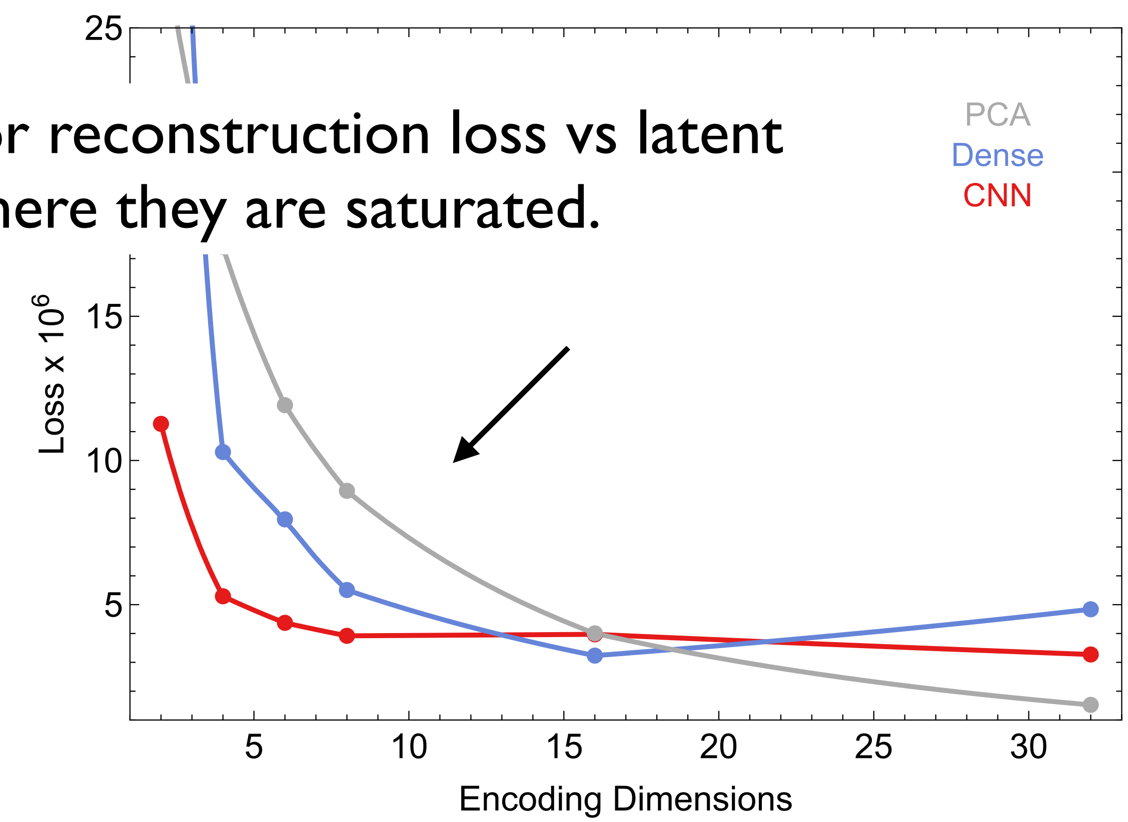
Choosing the latent dimension

d too large → autoencoder becomes identity transform
d too small → autoencoder cannot learn all the features

Should choose the latent dimension in an unsupervised manner
(ie without optimizing on a specific signal)

Can examine PCA eigenvalues or reconstruction loss vs latent dimension and look at where they are saturated.





We chose d=6

Autoencoder architectures

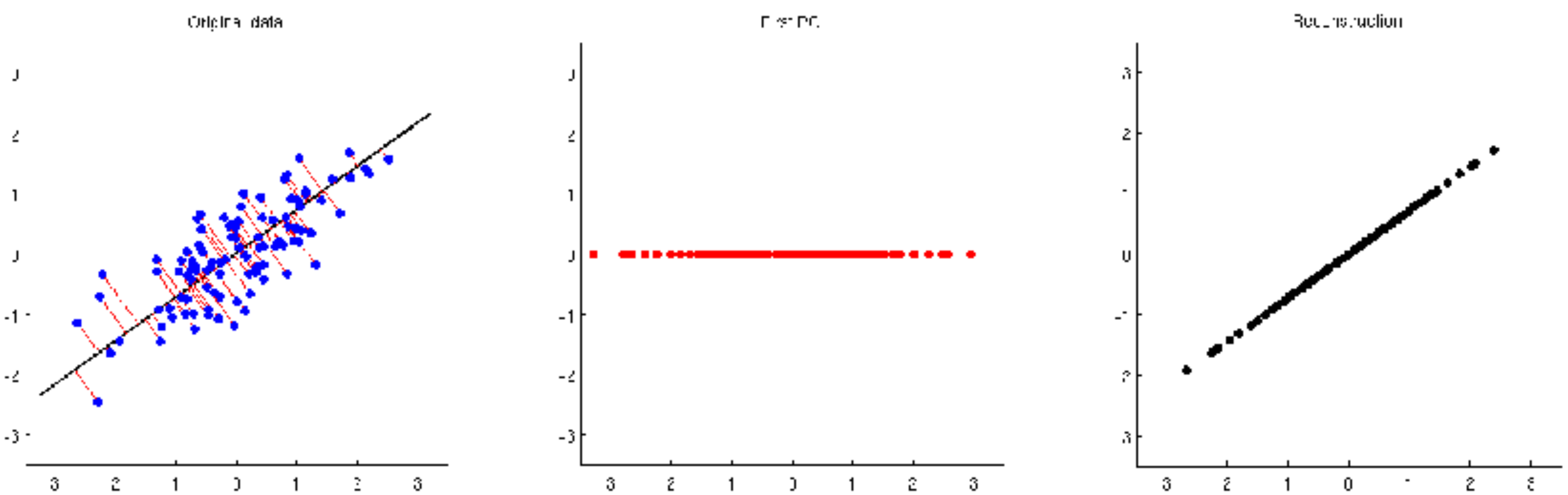
We considered three autoencoder architectures (many more are possible):

- Principal Component Analysis (PCA)
- Dense NN
- Convolutional NN

Autoencoder architectures

We considered three autoencoder architectures (many more are possible):

- Principal Component Analysis (PCA)



Project onto the first d PCA eigenvectors $$z = \mathcal{P}_d x_{in}$$

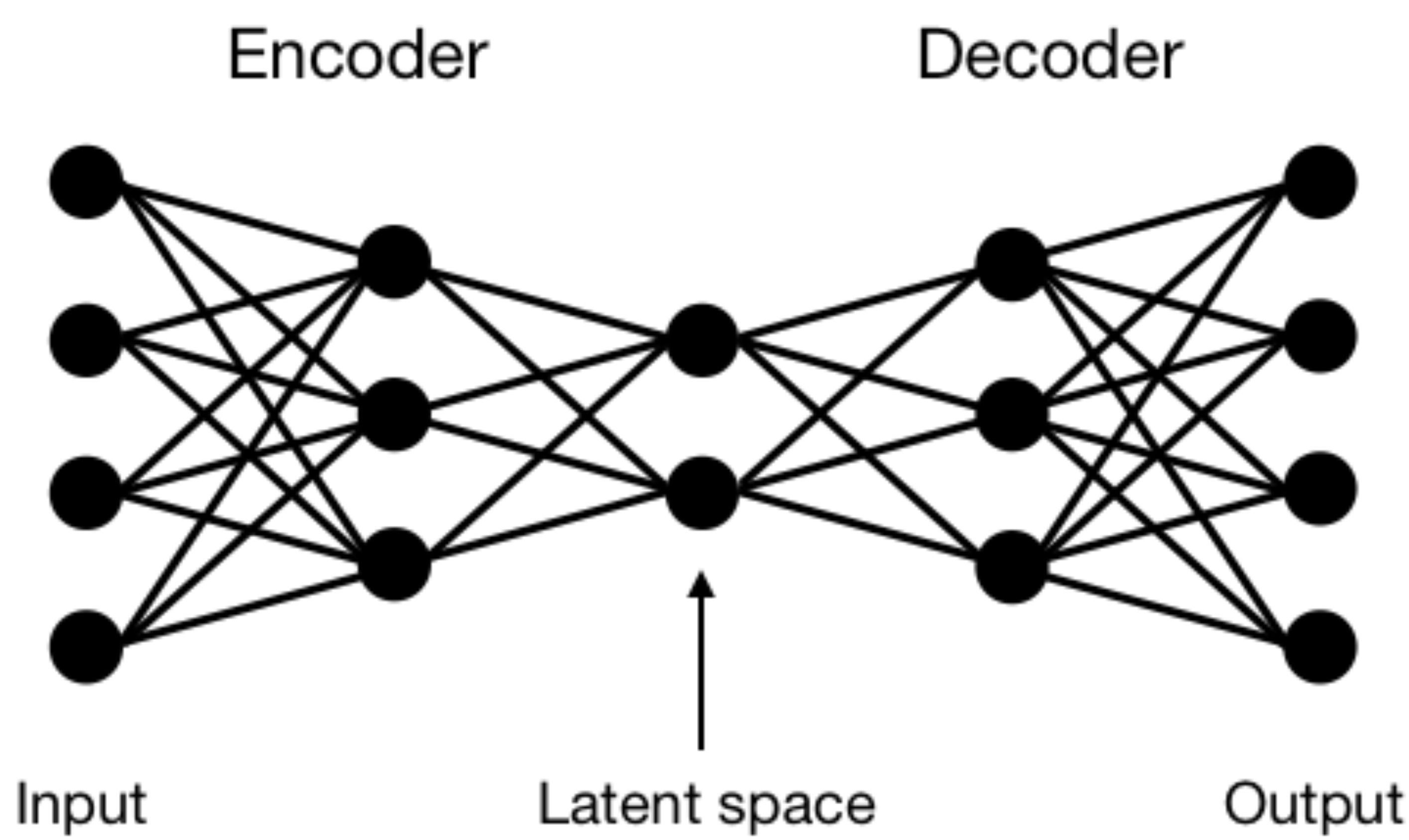
Inverse transform to reconstruct original input $$x_{out} = \mathcal{P}_d^T z = \mathcal{P}_d^T \mathcal{P}_d x_{in}$$

Autoencoder architectures

We considered three autoencoder architectures (many more are possible):

- Principal Component Analysis (PCA)
- Dense NN
- Convoluti

Flatten image into column vector

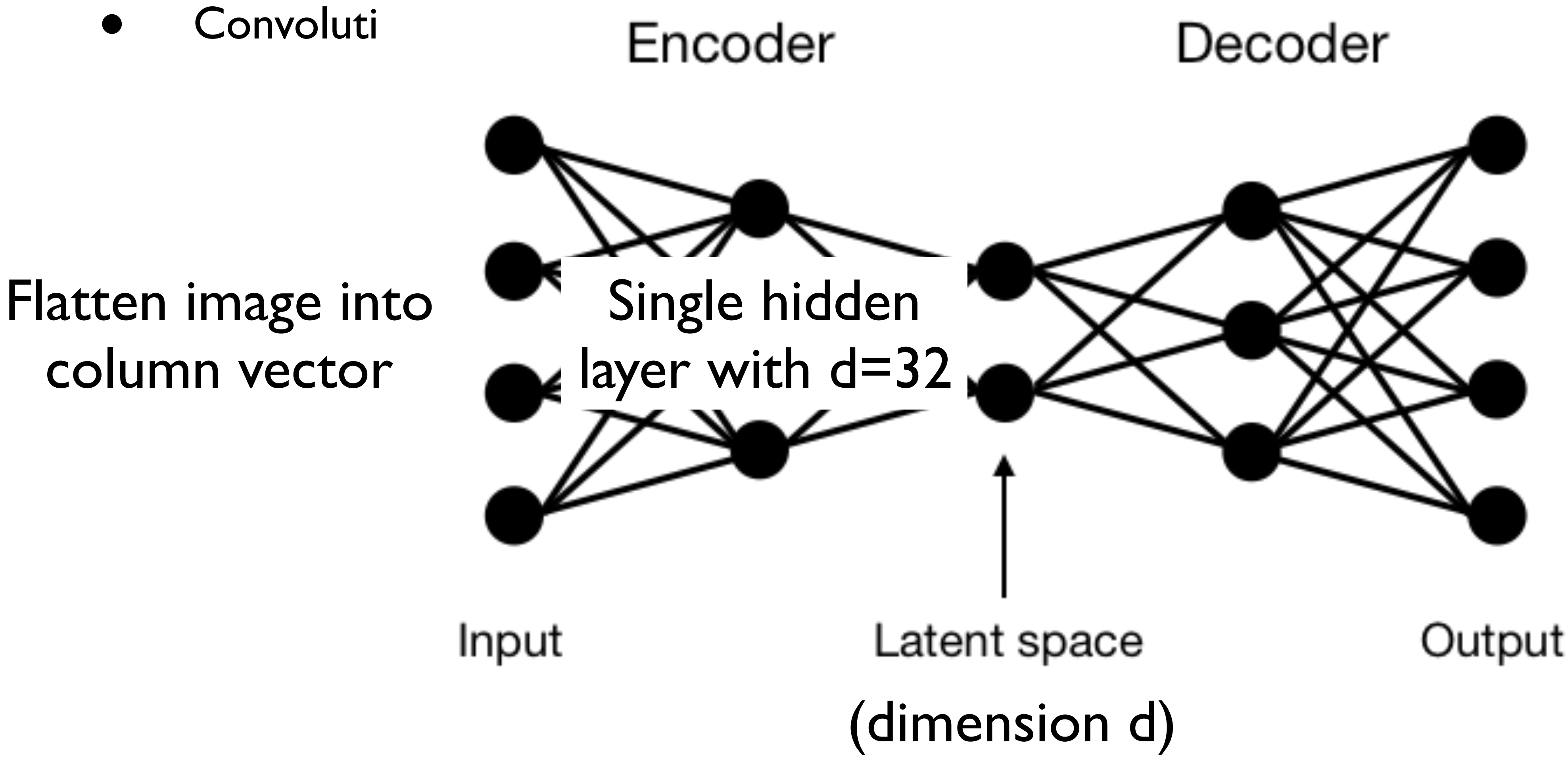


(dimension d)

Autoencoder architectures

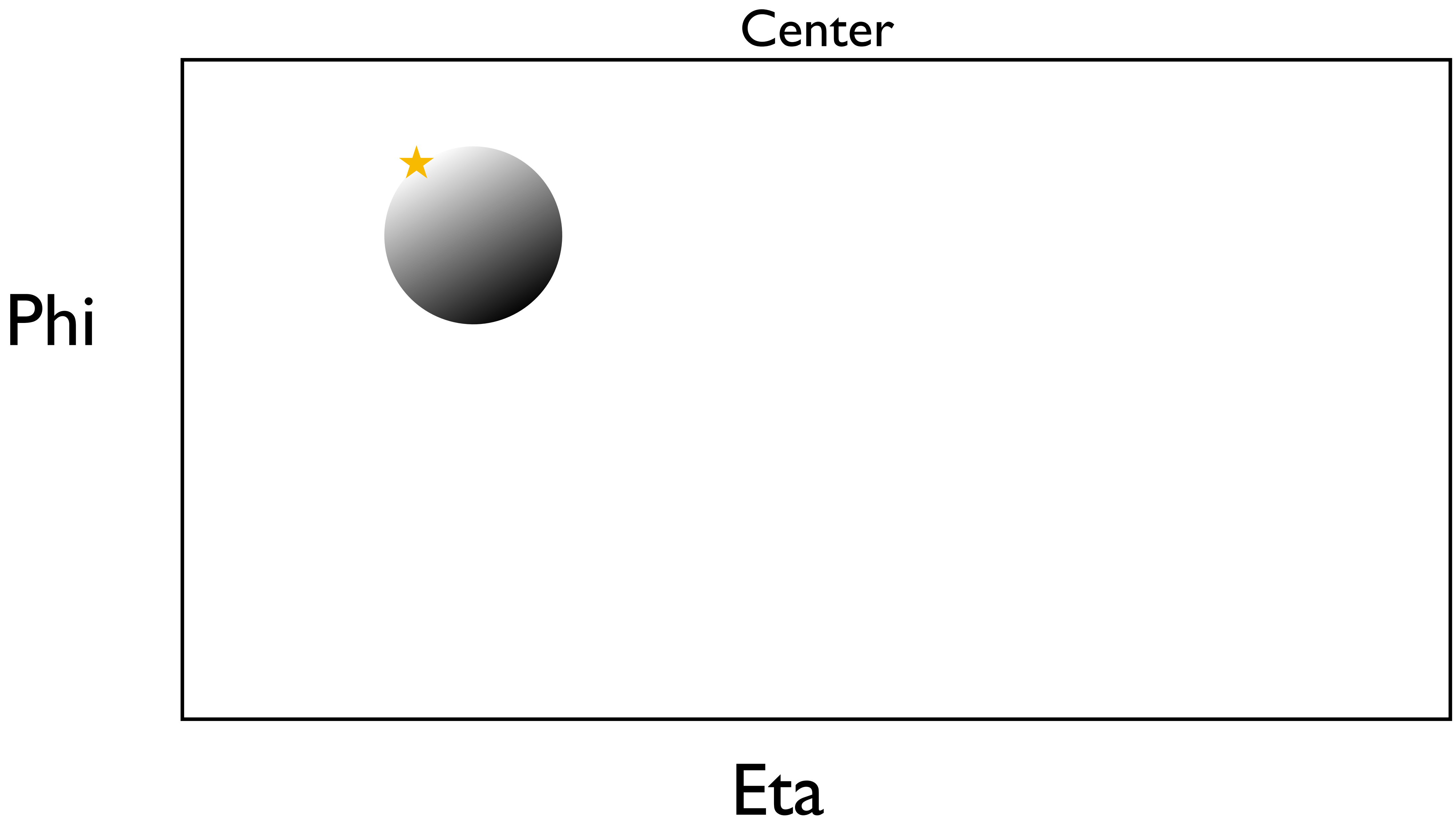
We considered three autoencoder architectures (many more are possible):

- Principal Component Analysis (PCA)
- Dense NN
- Convoluti



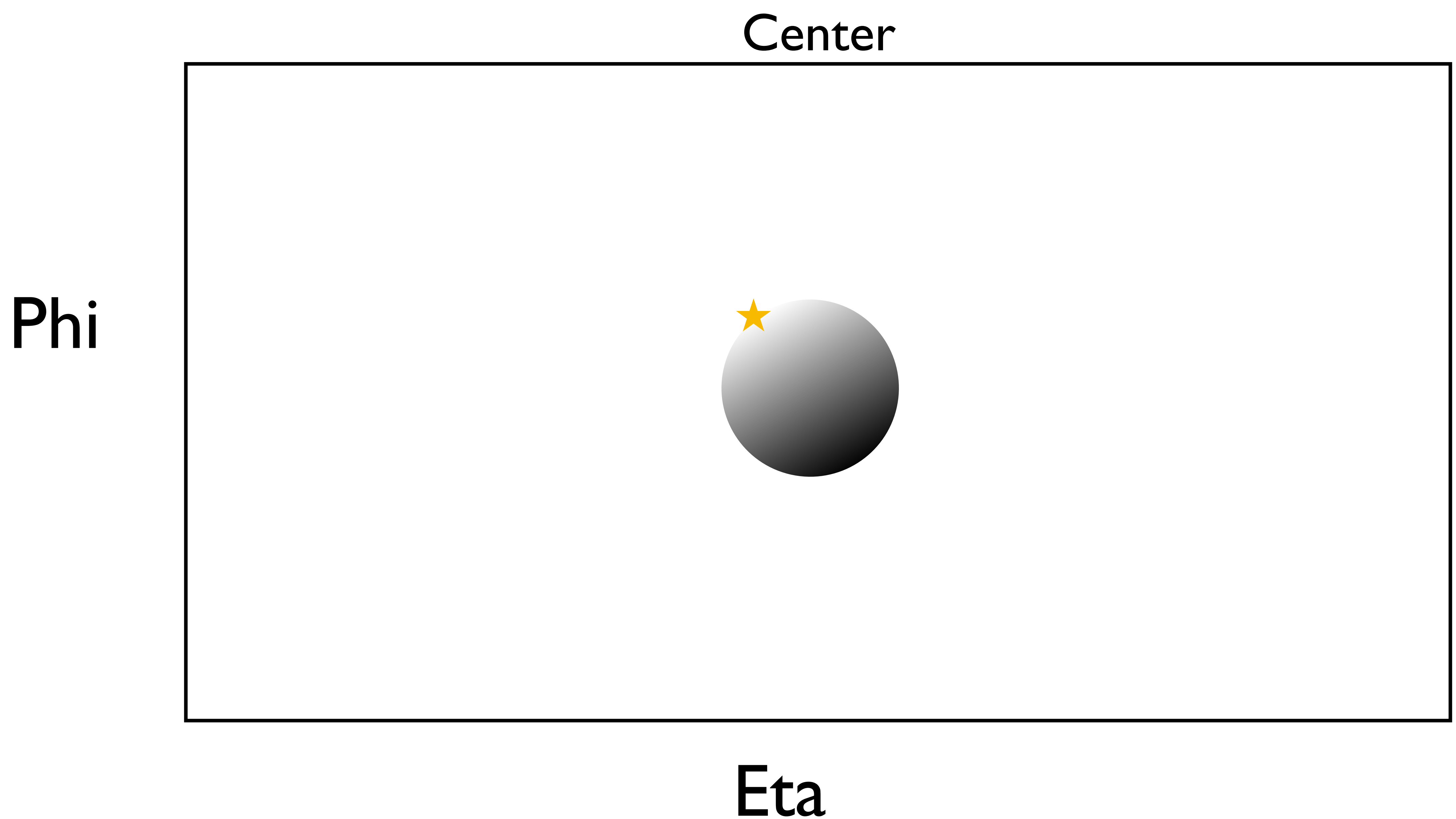
Jet Images — Preprocessing

For training the neural network, it is very useful to uniformize the jet images as much as possible, consistently with the physics.



Jet Images — Preprocessing

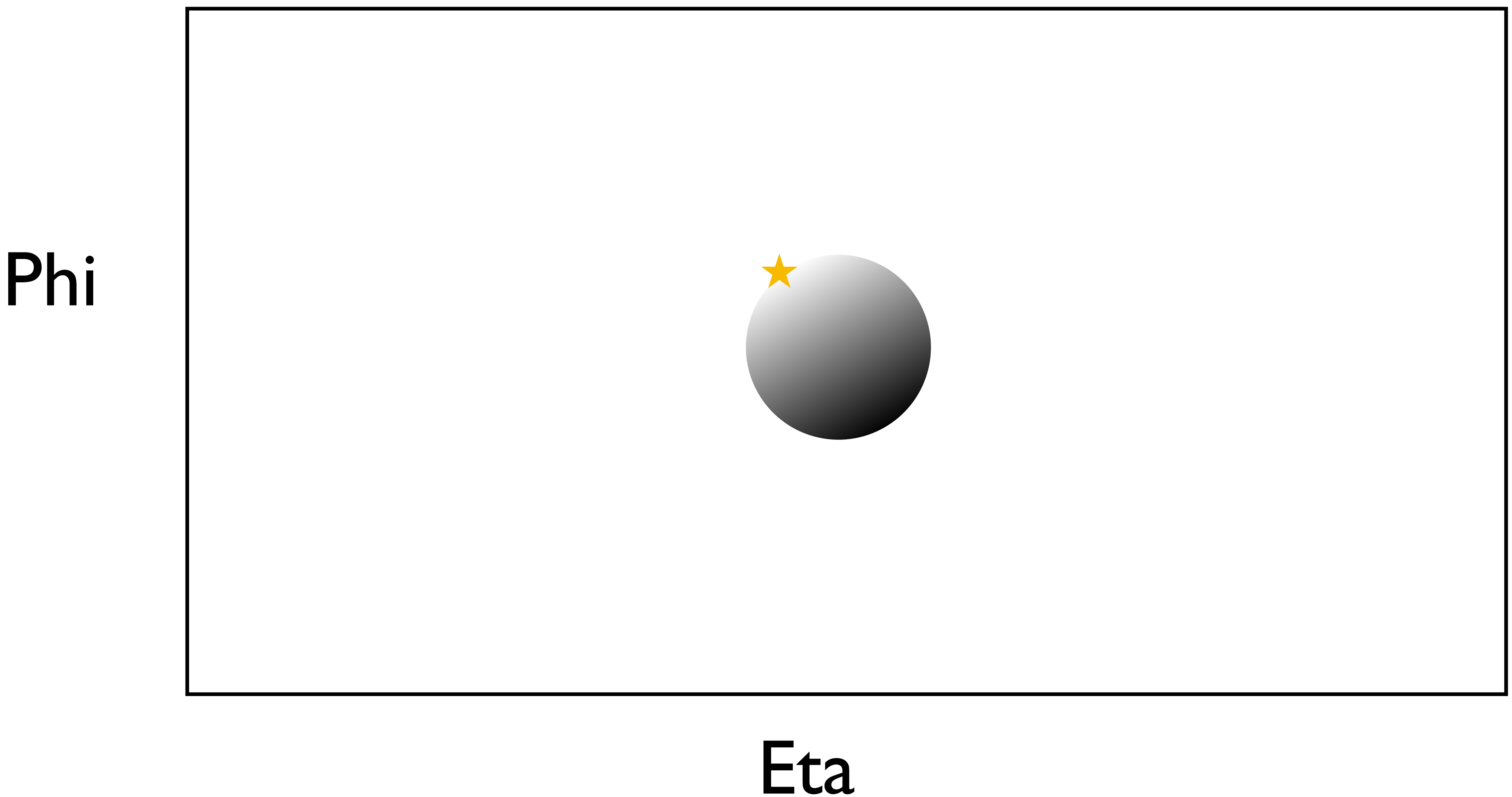
For training the neural network, it is very useful to uniformize the jet images as much as possible, consistently with the physics.



Jet Images — Preprocessing

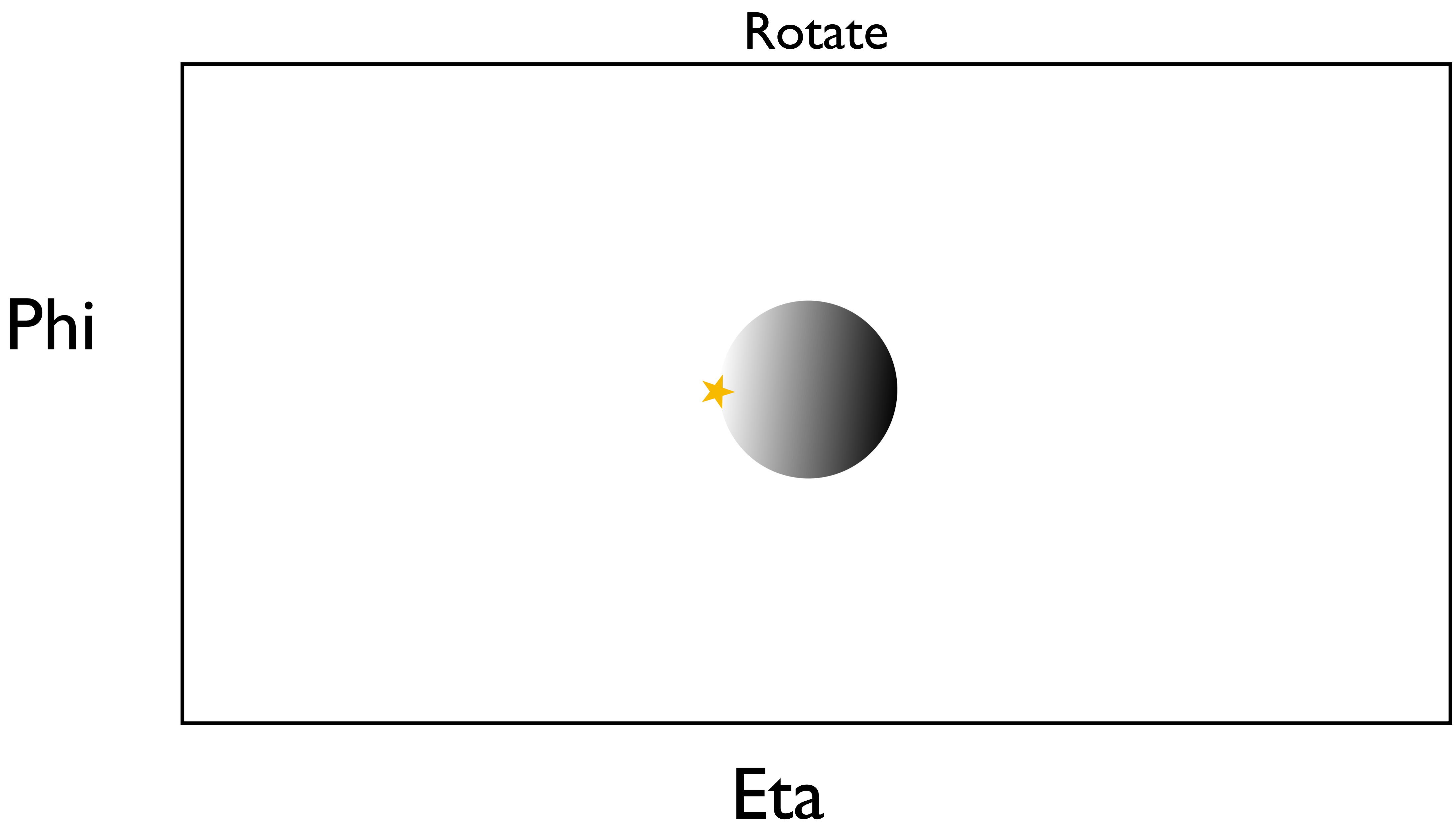
For training the neural network, it is very useful to uniformize the jet images as much as possible, consistently with the physics.

Rotate



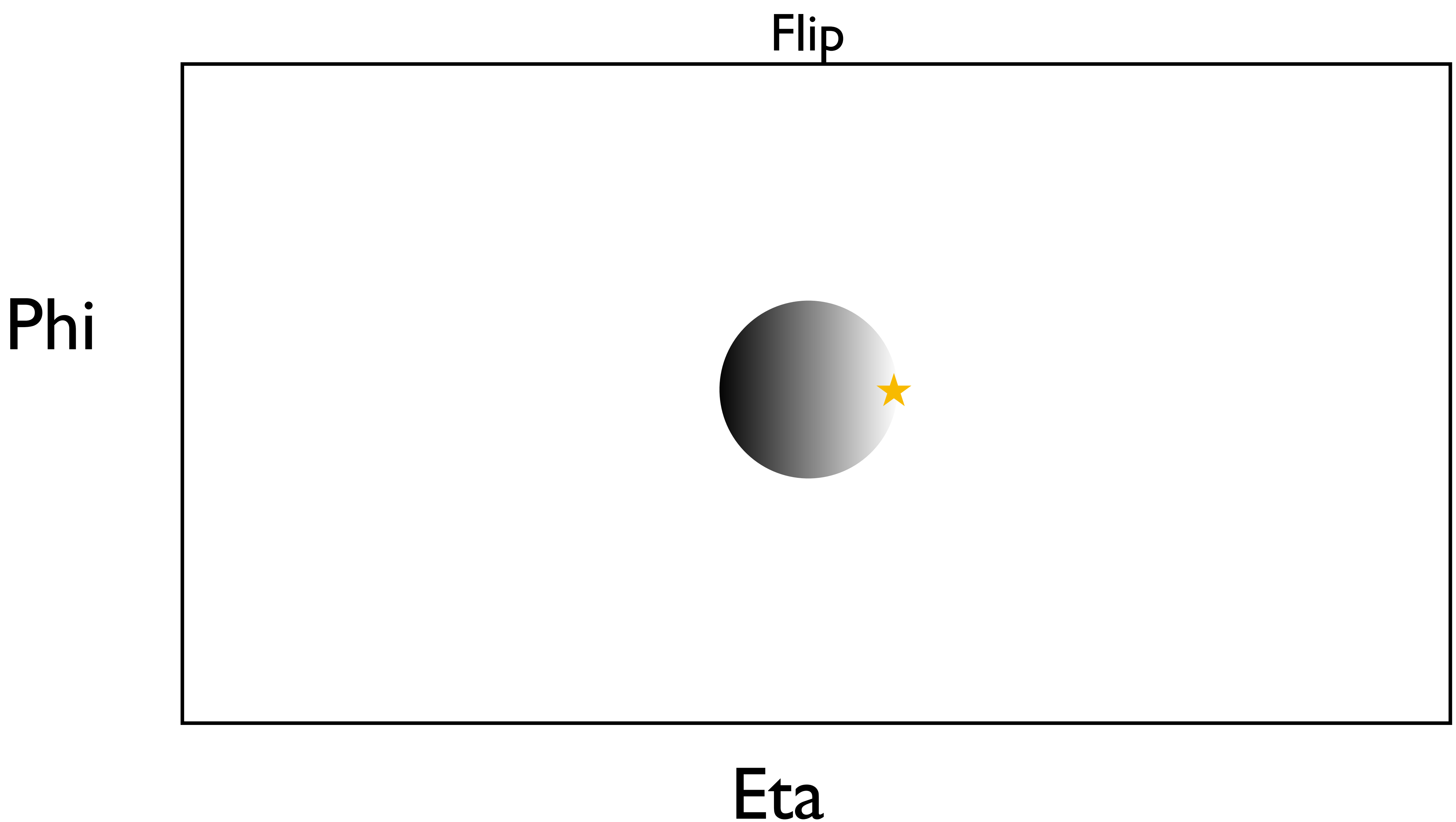
Jet Images — Preprocessing

For training the neural network, it is very useful to uniformize the jet images as much as possible, consistently with the physics.



Jet Images — Preprocessing

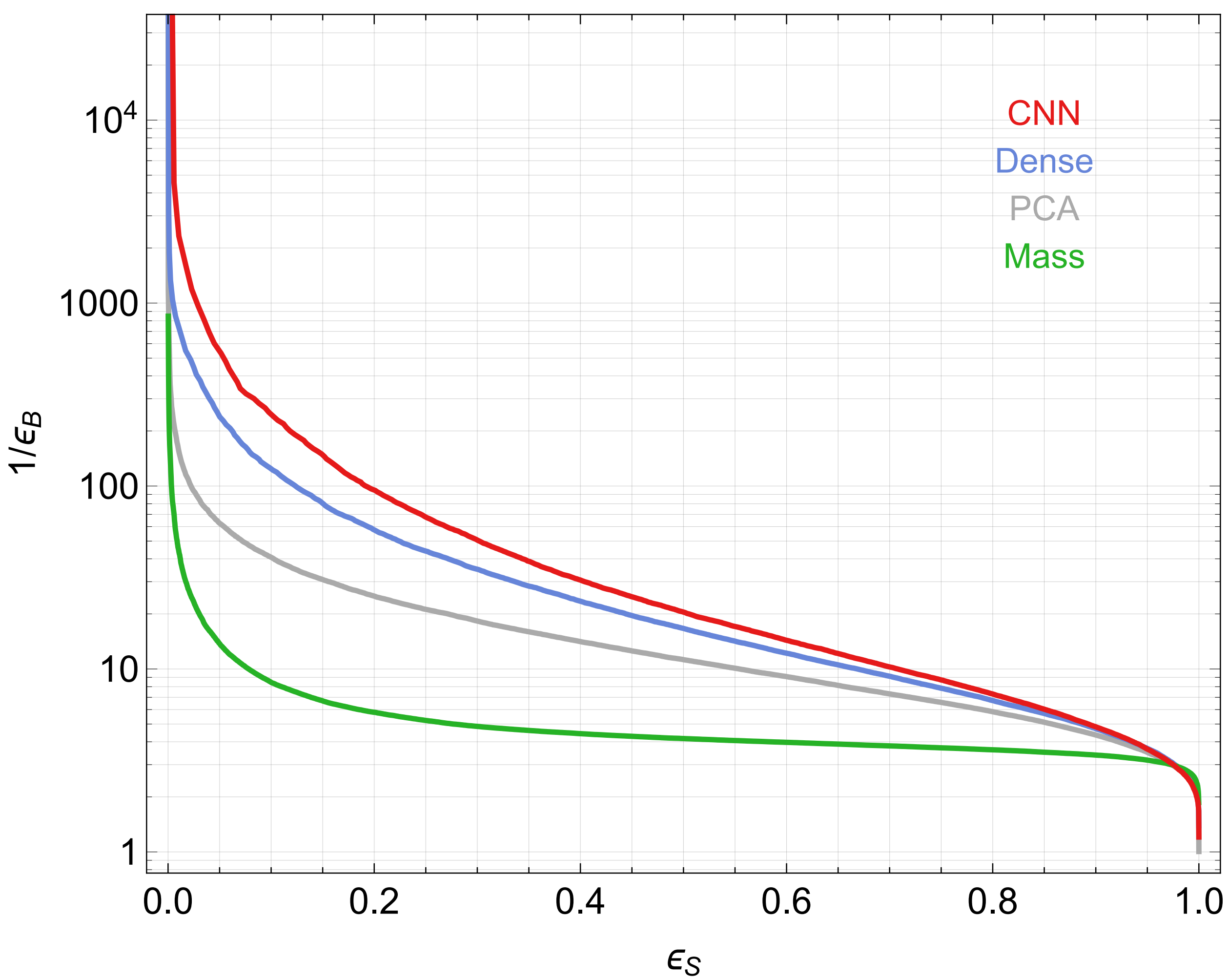
For training the neural network, it is very useful to uniformize the jet images as much as possible, consistently with the physics.



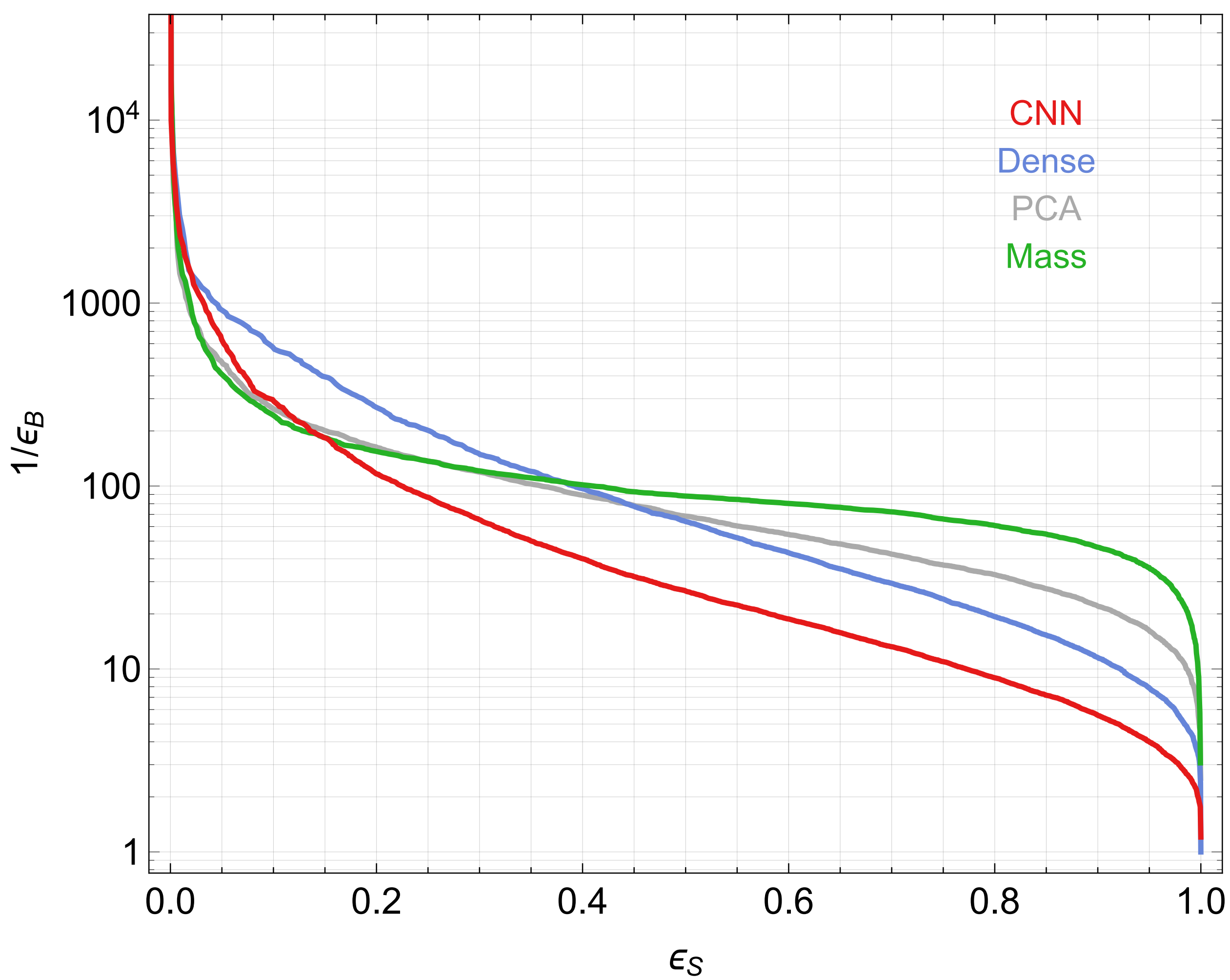
Comparison vs jet mass

How do our fancy autoencoders compare against a simpler anomaly detection method: jet mass bump hunt?

Tops



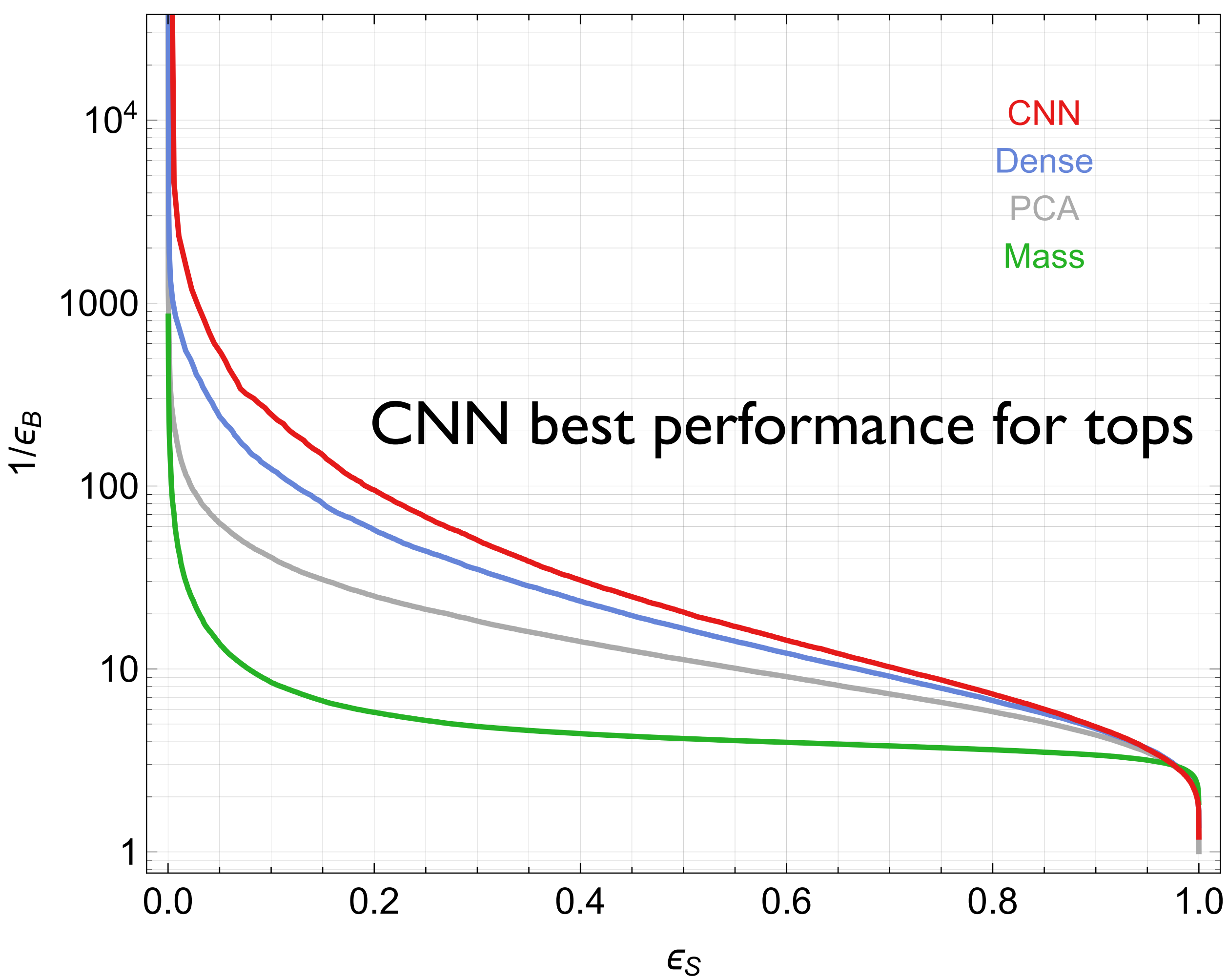
400 GeV gluinos



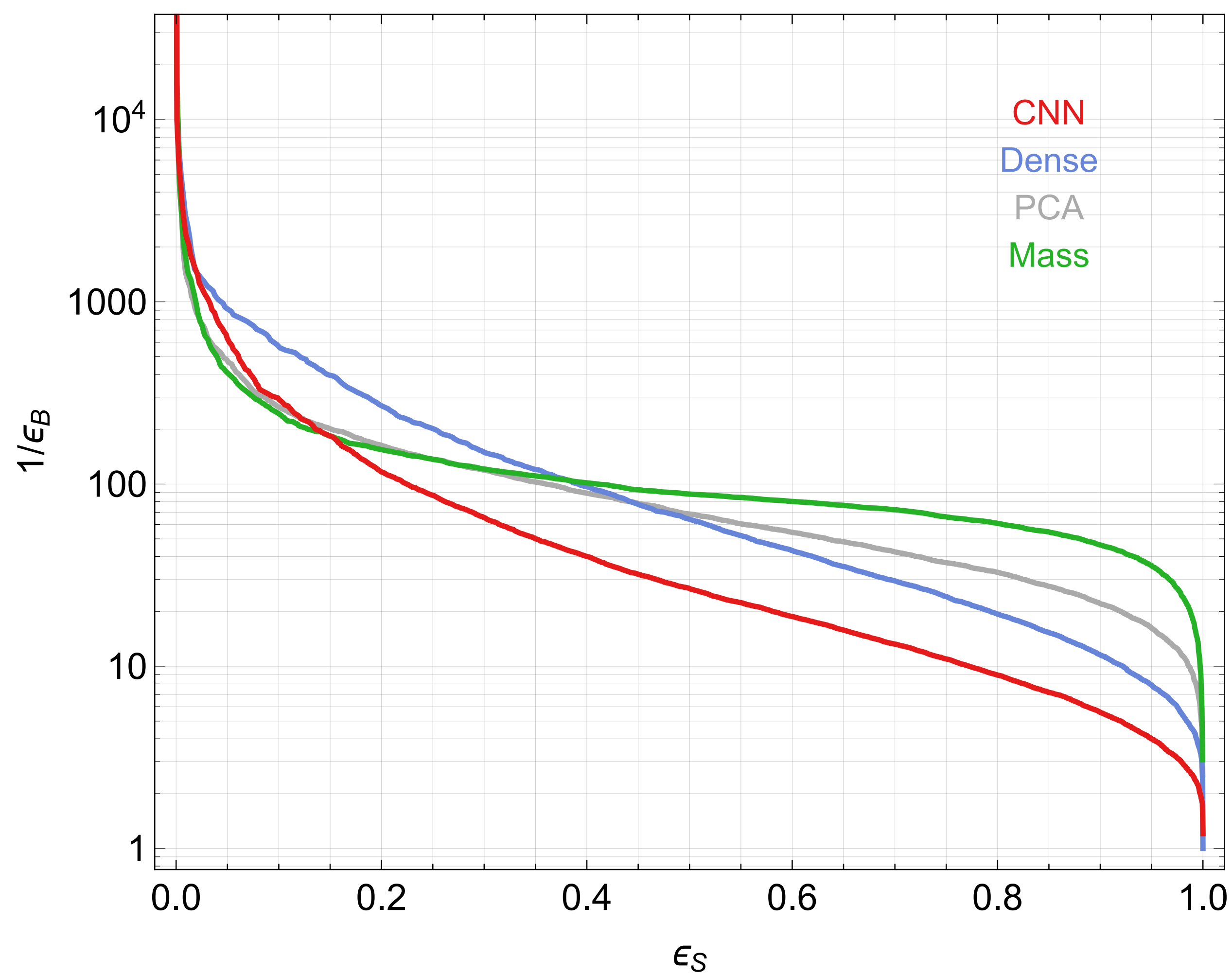
Comparison vs jet mass

How do our fancy autoencoders compare against a simpler anomaly detection method: jet mass bump hunt?

Tops

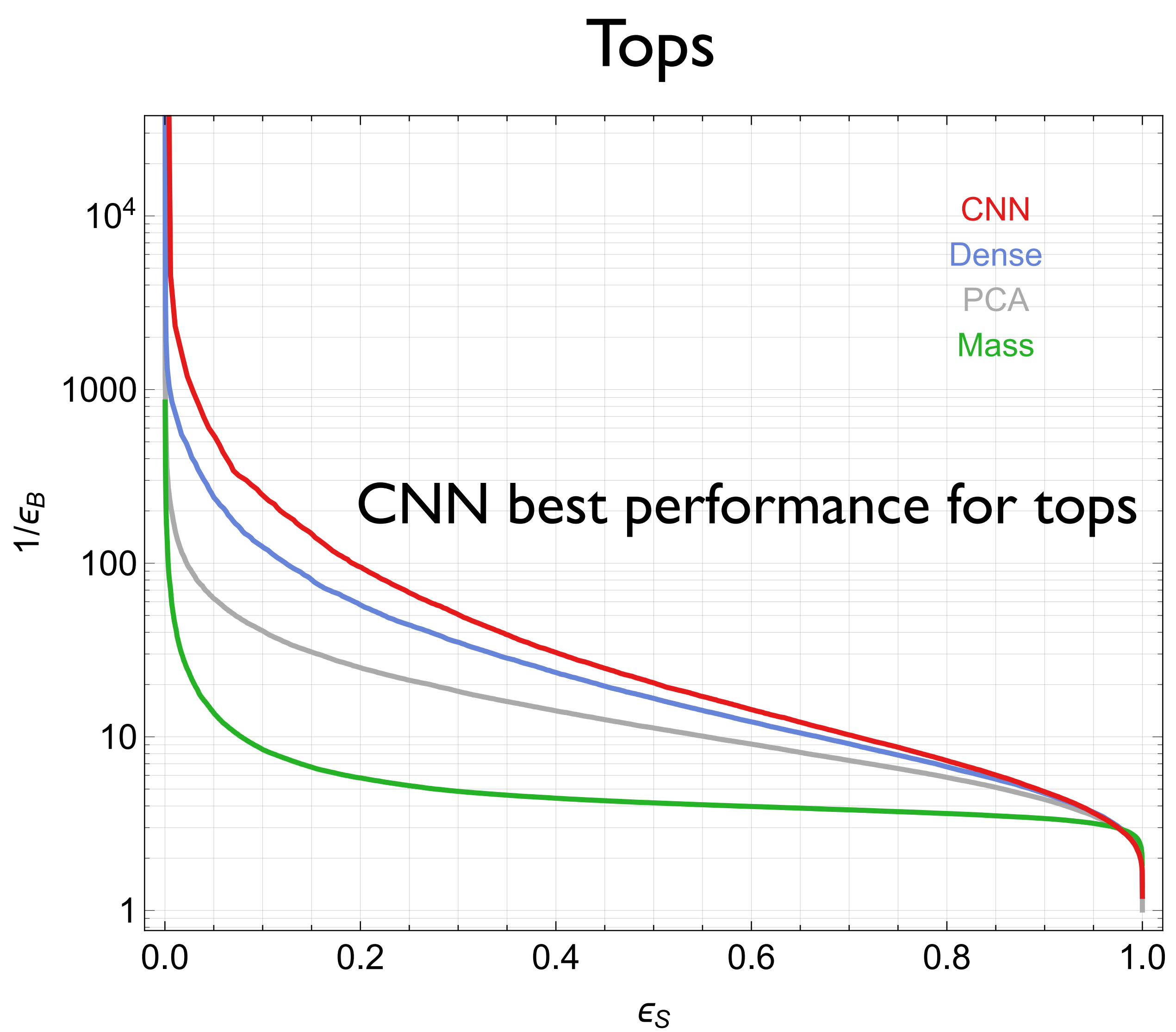


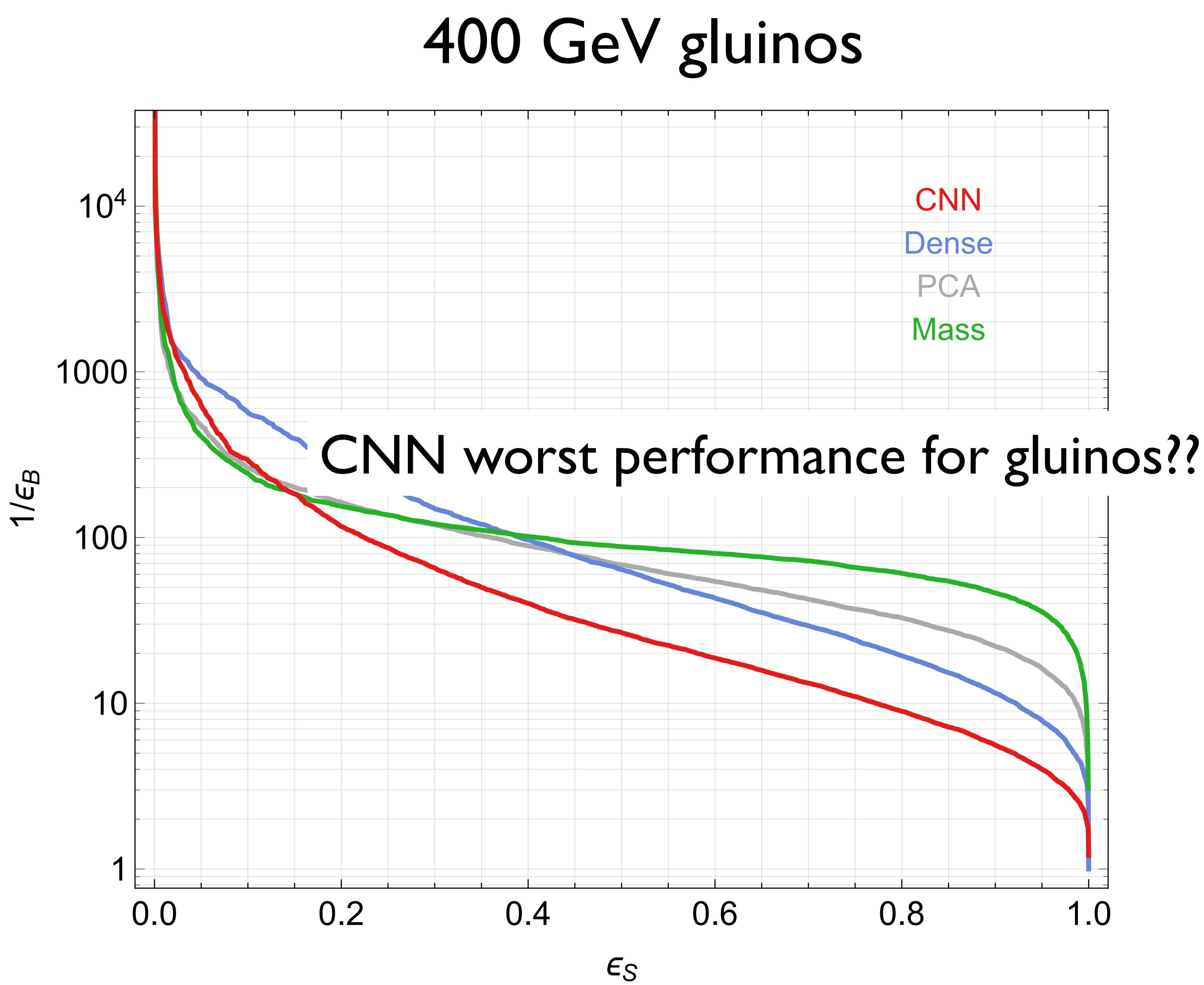
400 GeV gluinos



Comparison vs jet mass

How do our fancy autoencoders compare against a simpler anomaly detection method: jet mass bump hunt?

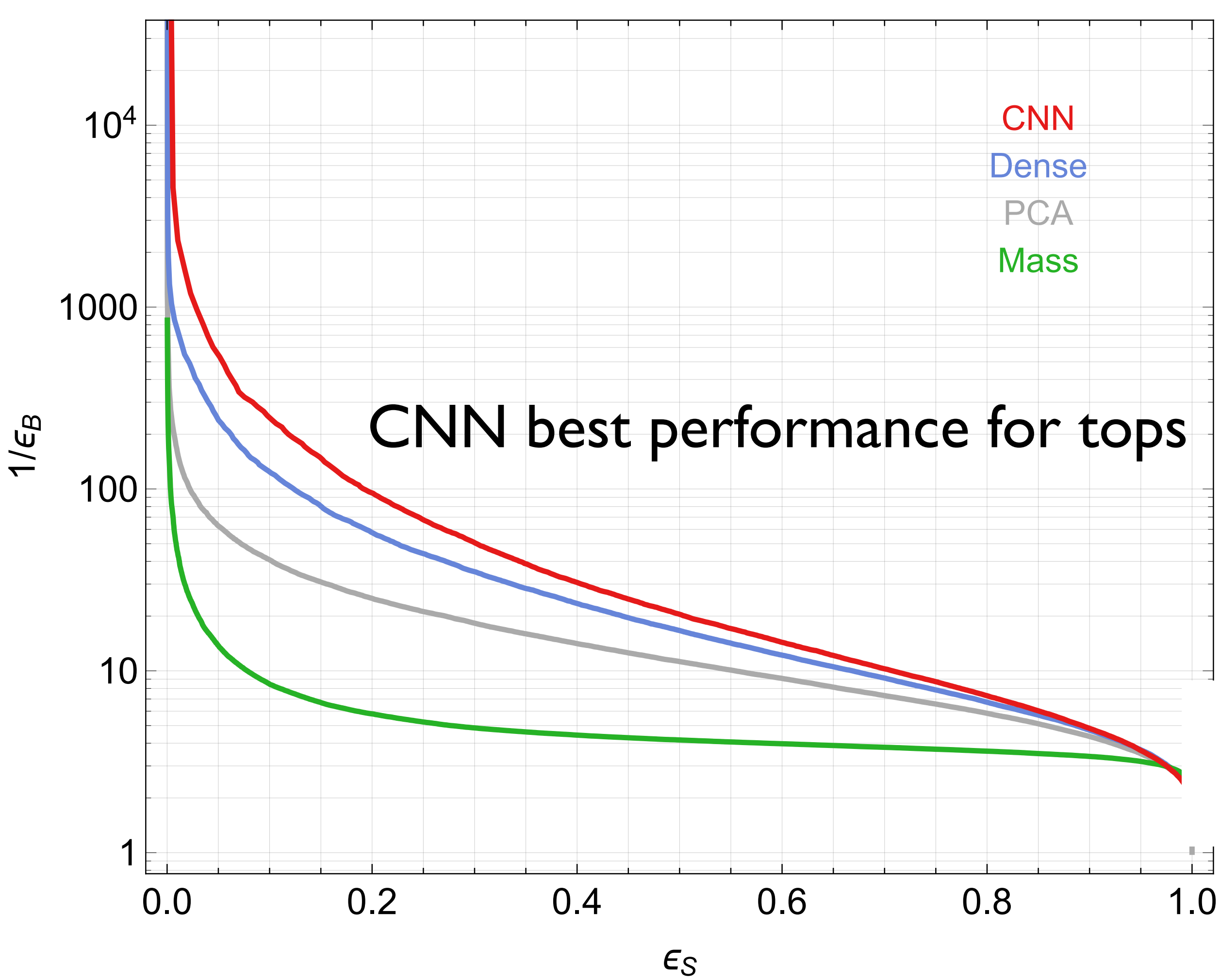




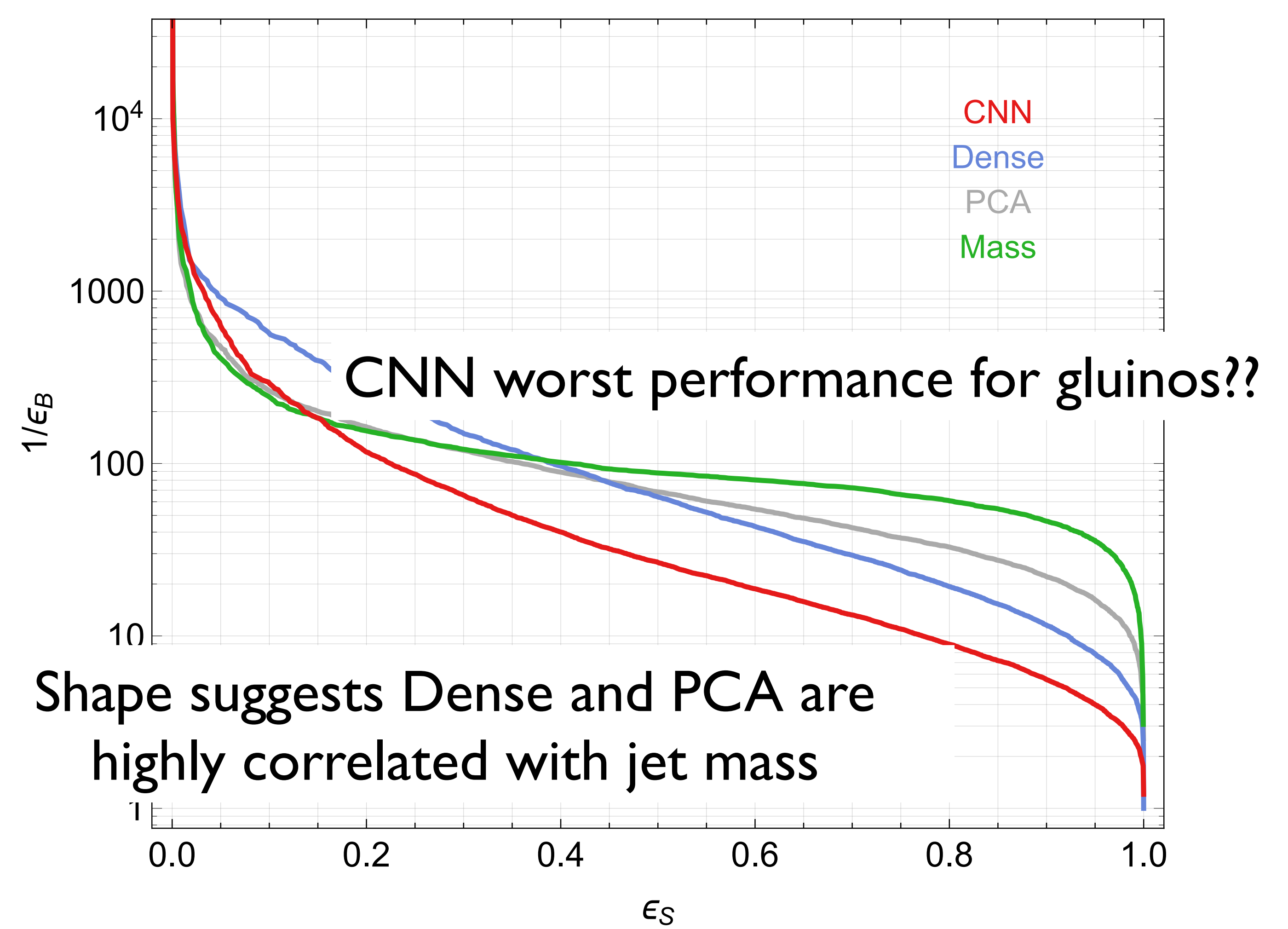
Comparison vs jet mass

How do our fancy autoencoders compare against a simpler anomaly detection method: jet mass bump hunt?

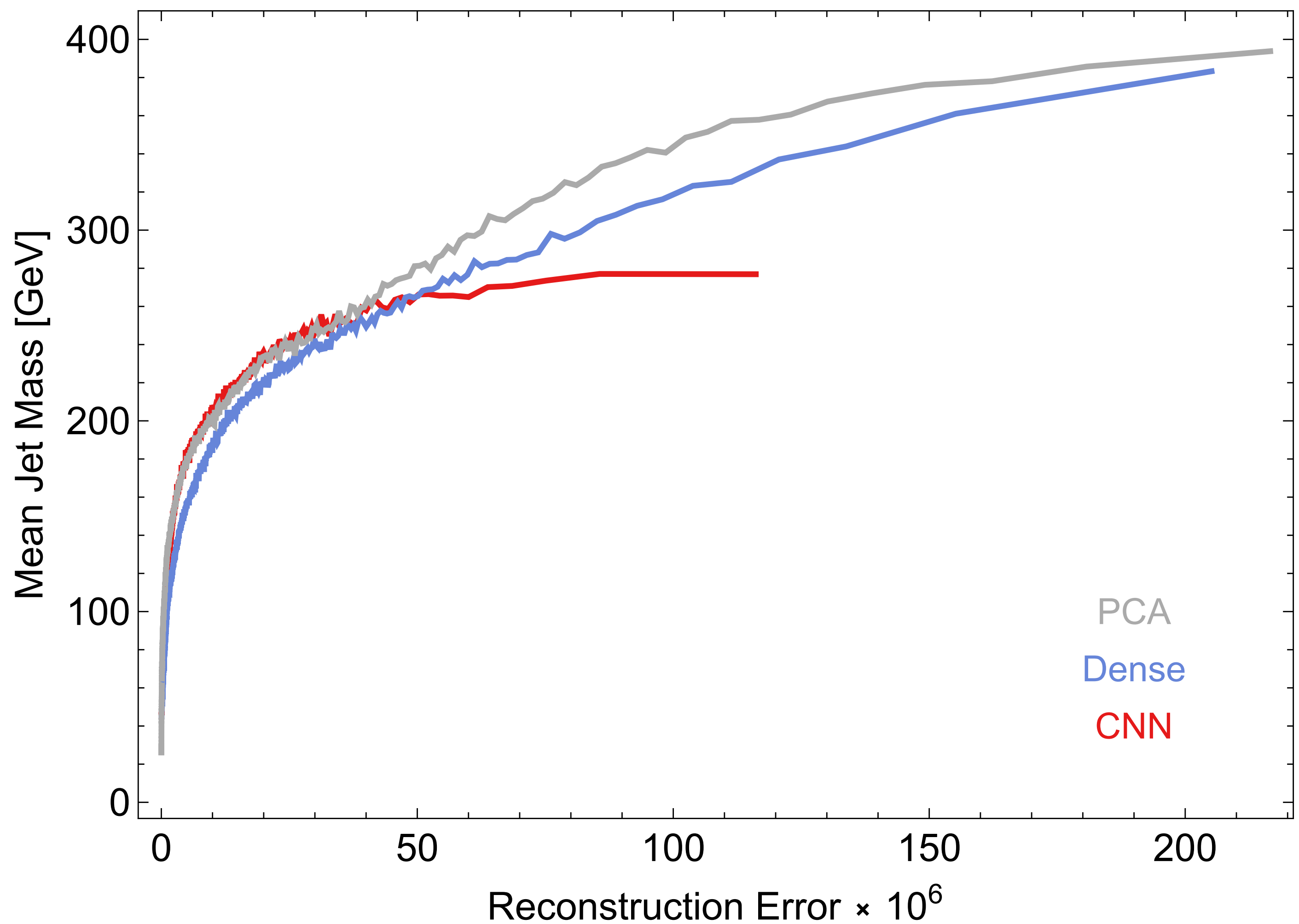
Tops



400 GeV gluinos



Correlation with jet mass



Indeed, this is confirmed by looking at mean jet mass in bins of reconstruction error for the QCD background.

CNN is no longer correlated with jet mass for m≳250 GeV

Robustness with other Monte Carlo