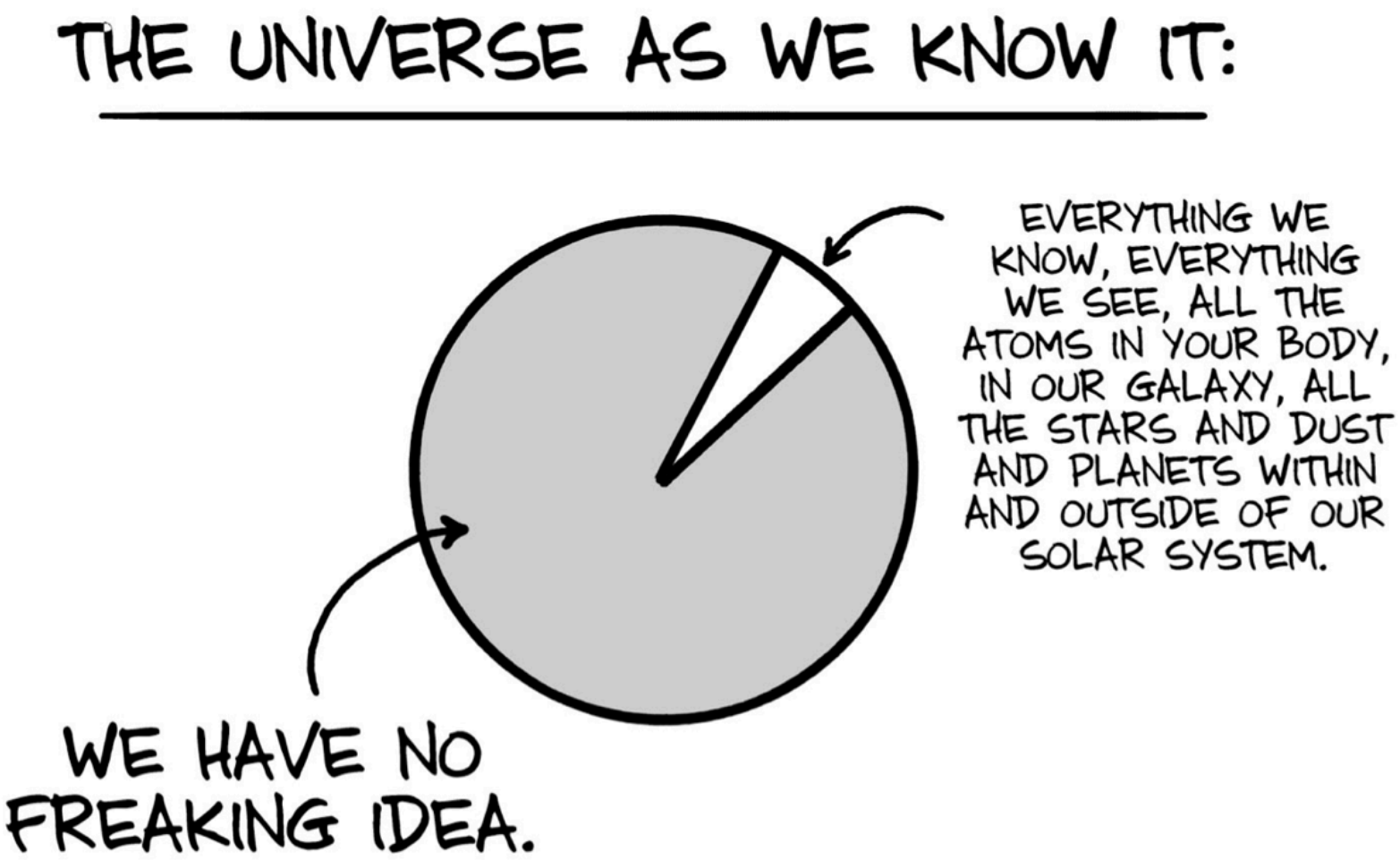
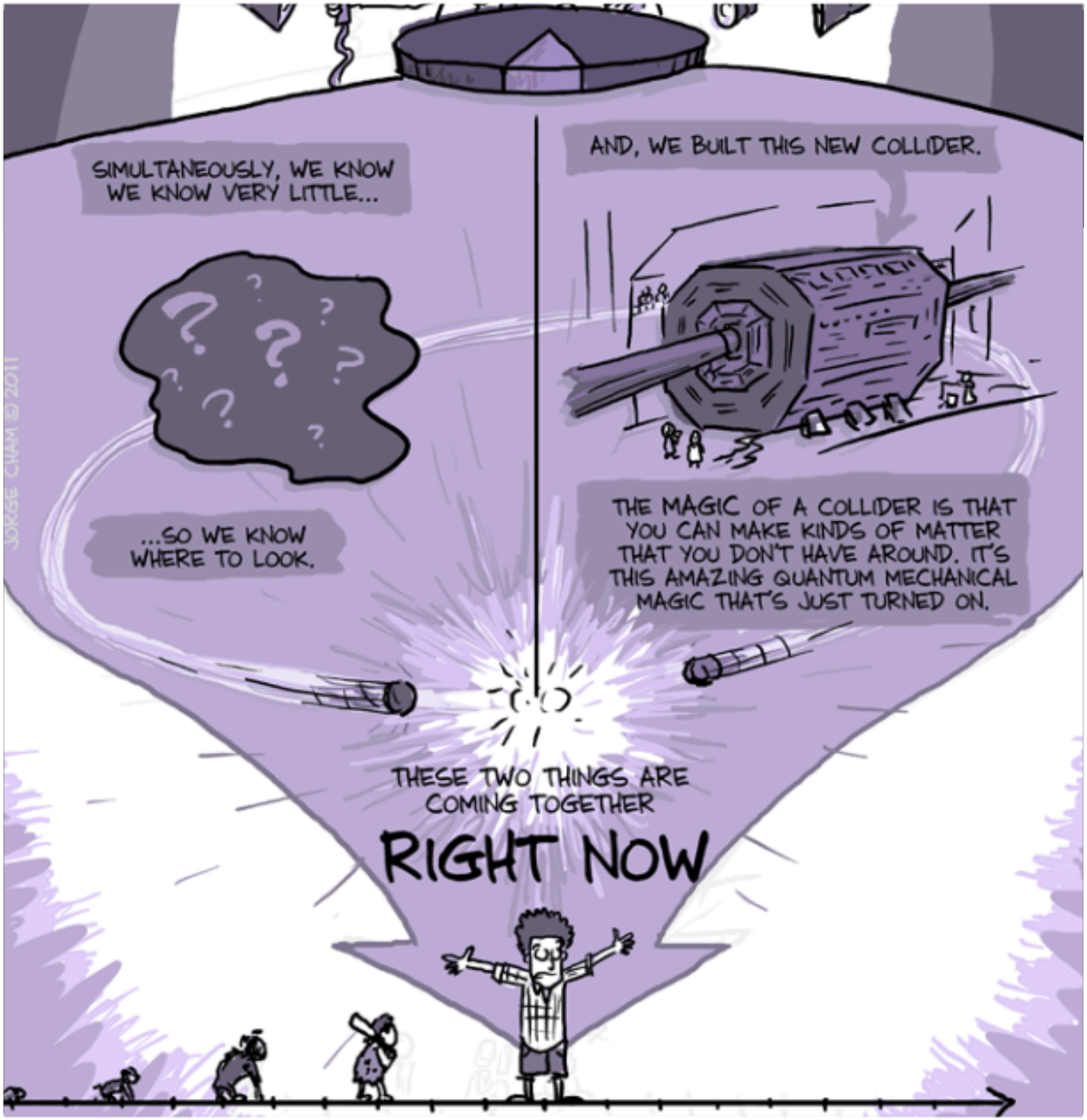
Open ML problems in High Energy Physics

Daniel Whiteson, UC Irvine

NIPS2015, ALEPH workshop

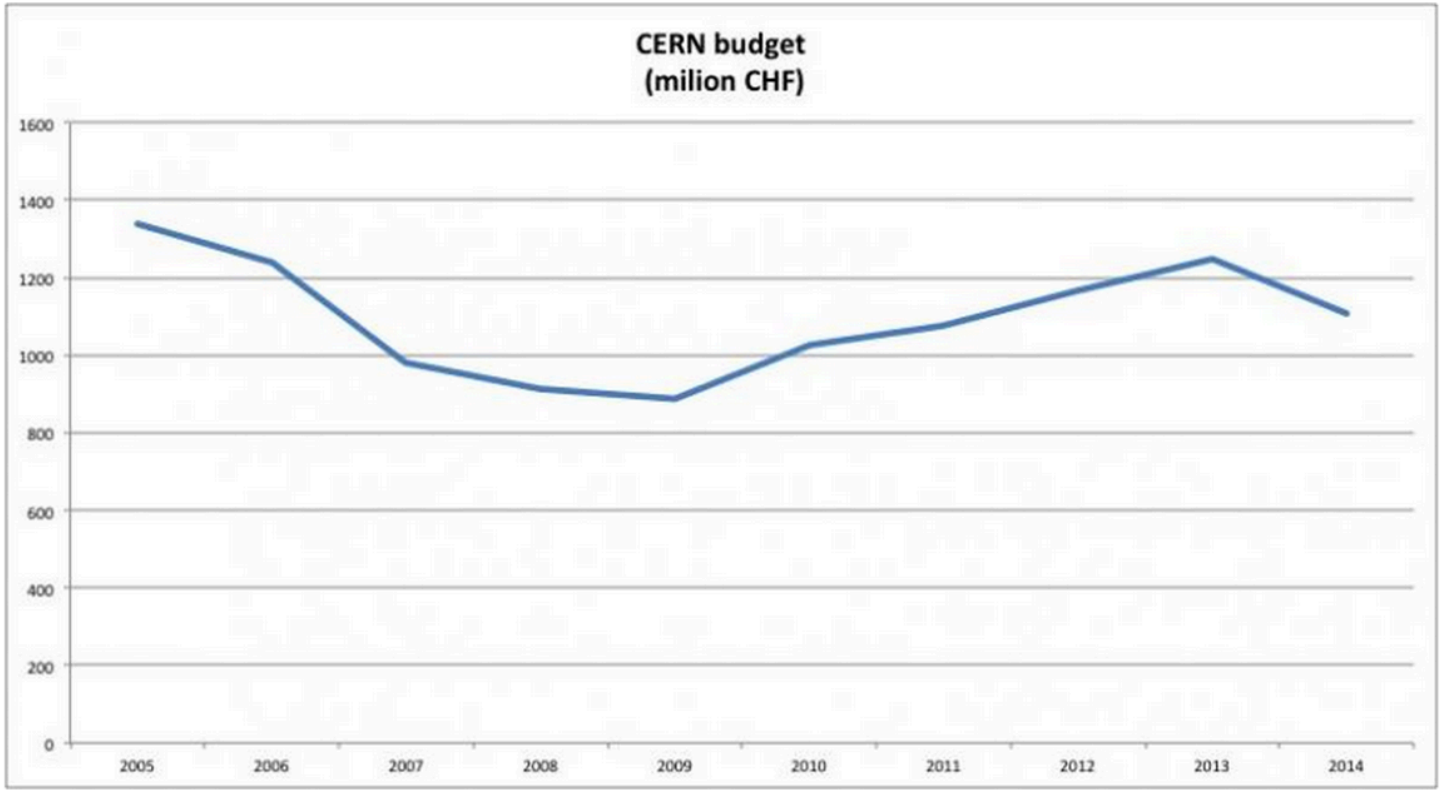
Motivation


THE UNIVERSE AS WE KNOW IT:
EVERYTHING WE KNOW, EVERYTHING WE SEE, ALL THE ATOMS IN YOUR BODY, IN OUR GALAXY, ALL THE STARS AND DUST AND PLANETS WITHIN AND OUTSIDE OF OUR SOLAR SYSTEM.
WE HAVE NO FREAKING IDEA.


SIMULTANEOUSLY, WE KNOW WE KNOW VERY LITTLE...
...SO WE KNOW WHERE TO LOOK.
AND, WE BUILT THIS NEW COLLIDER.
THE MAGIC OF A COLLIDER IS THAT YOU CAN MAKE KINDS OF MATTER THAT YOU DON'T HAVE AROUND. IT'S THIS AMAZING QUANTUM MECHANICAL MAGIC THAT'S JUST TURNED ON.
THESE TWO THINGS ARE COMING TOGETHER
RIGHT NOW
JORGE CHAM © 2011

Why do we need ML?

Budget overview



Outline

Defining the problem

Dimensionality reduction

Current approaches

Deep(er) networks for low(er)-level data

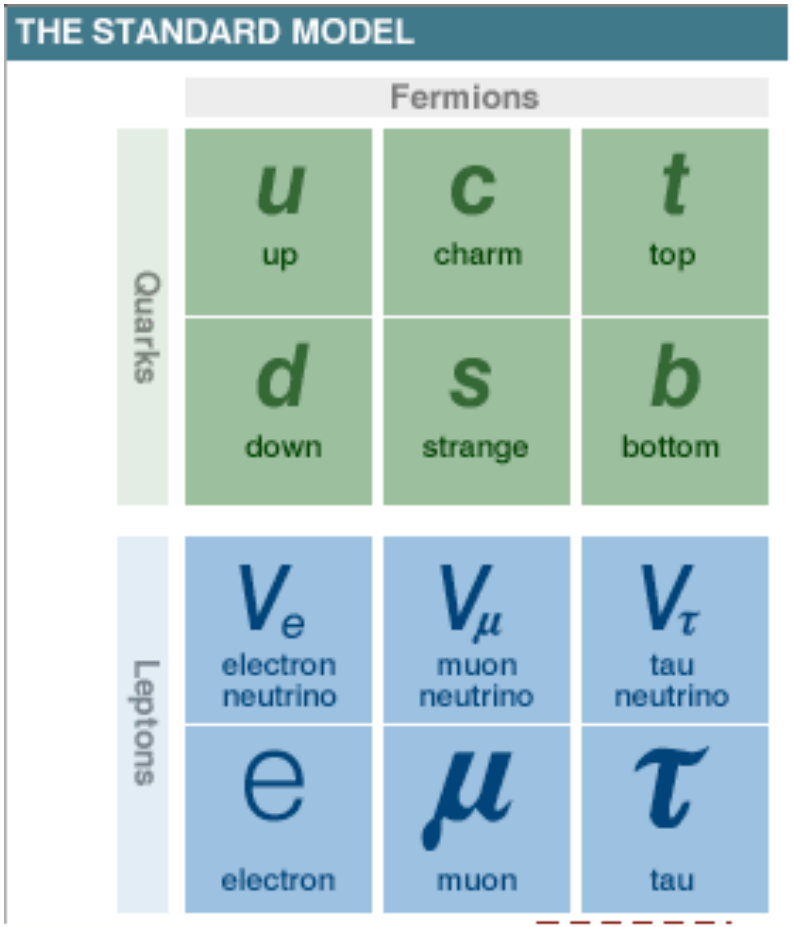
Dimensionality increase

Parameterized networks for sets of problems

Hypothesis testing

To search for a new particle, we compare the predictions of two hypotheses:

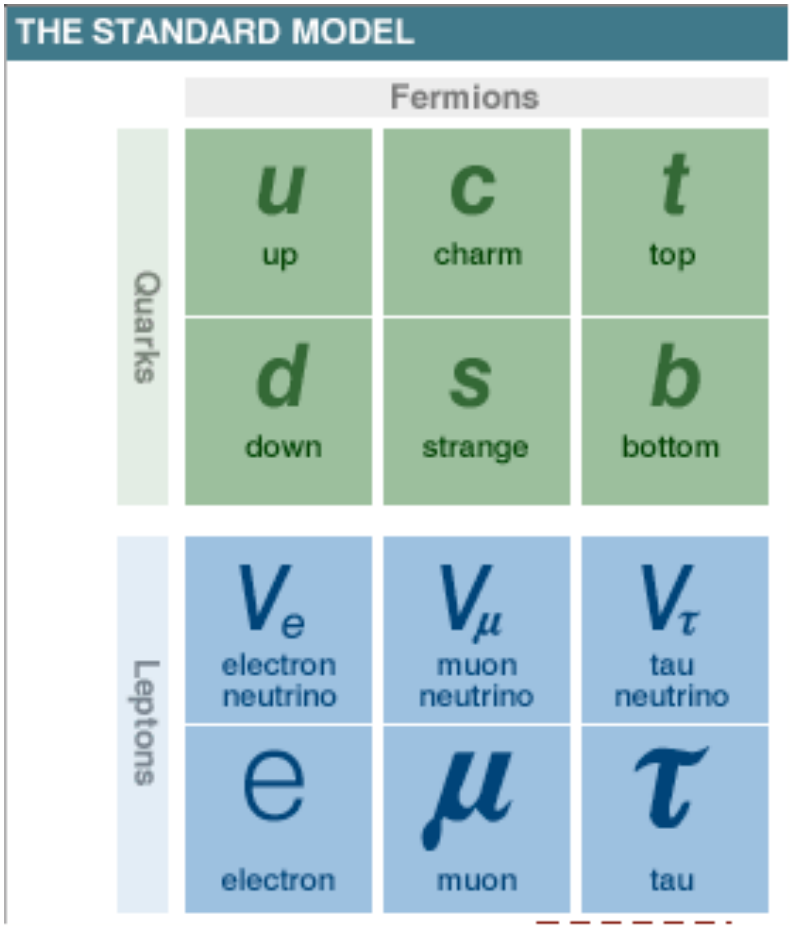
1.



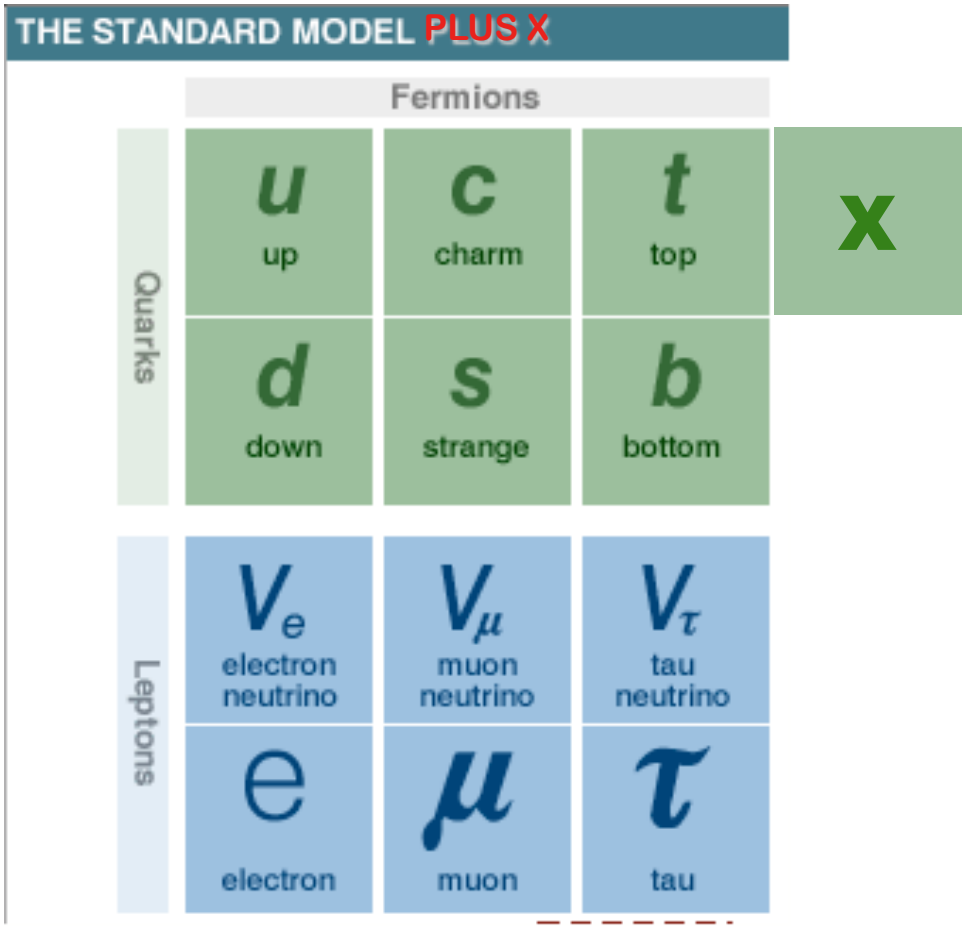
Hypothesis testing

To search for a new particle, we compare the predictions of two hypotheses:

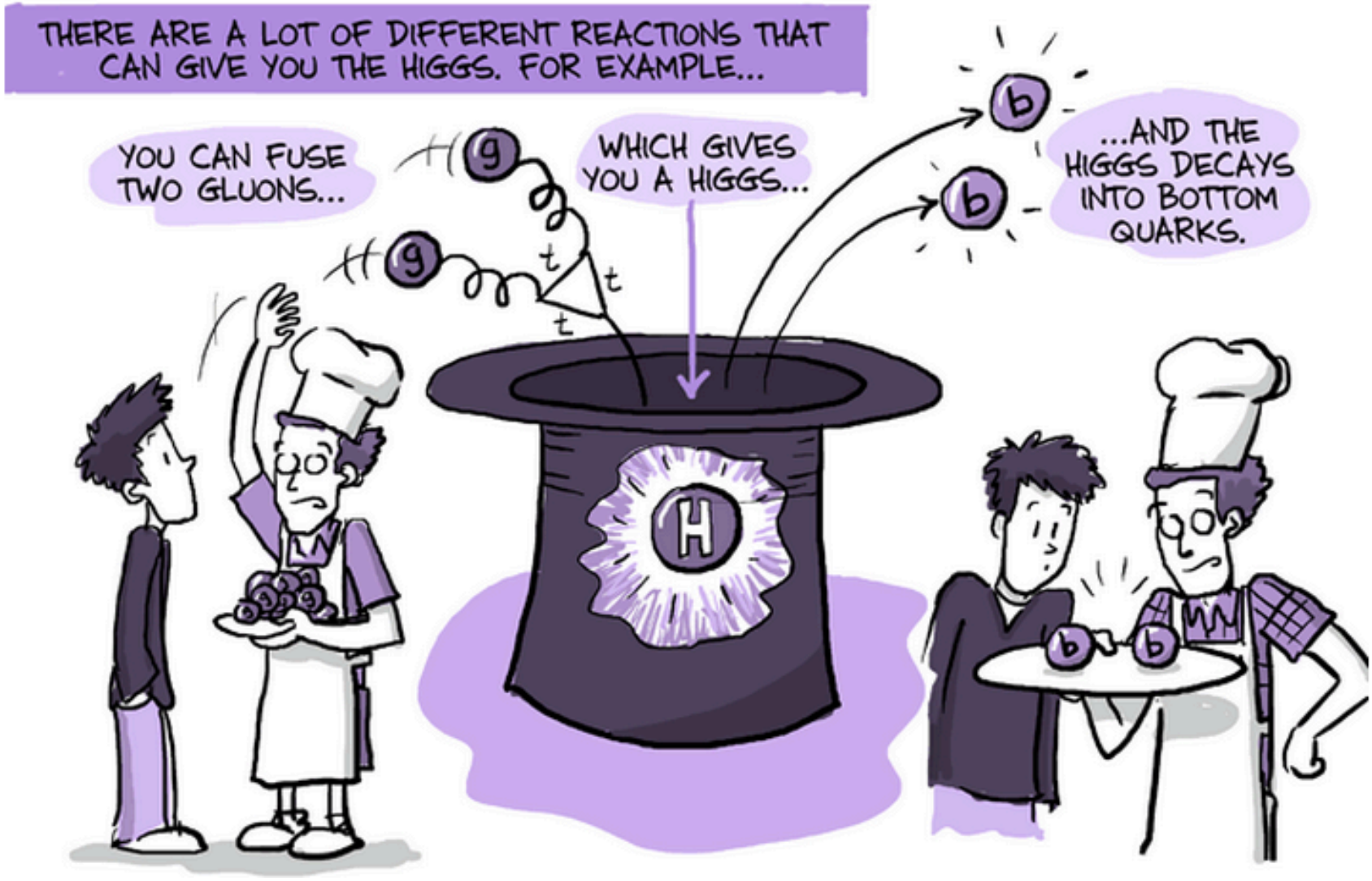
1.



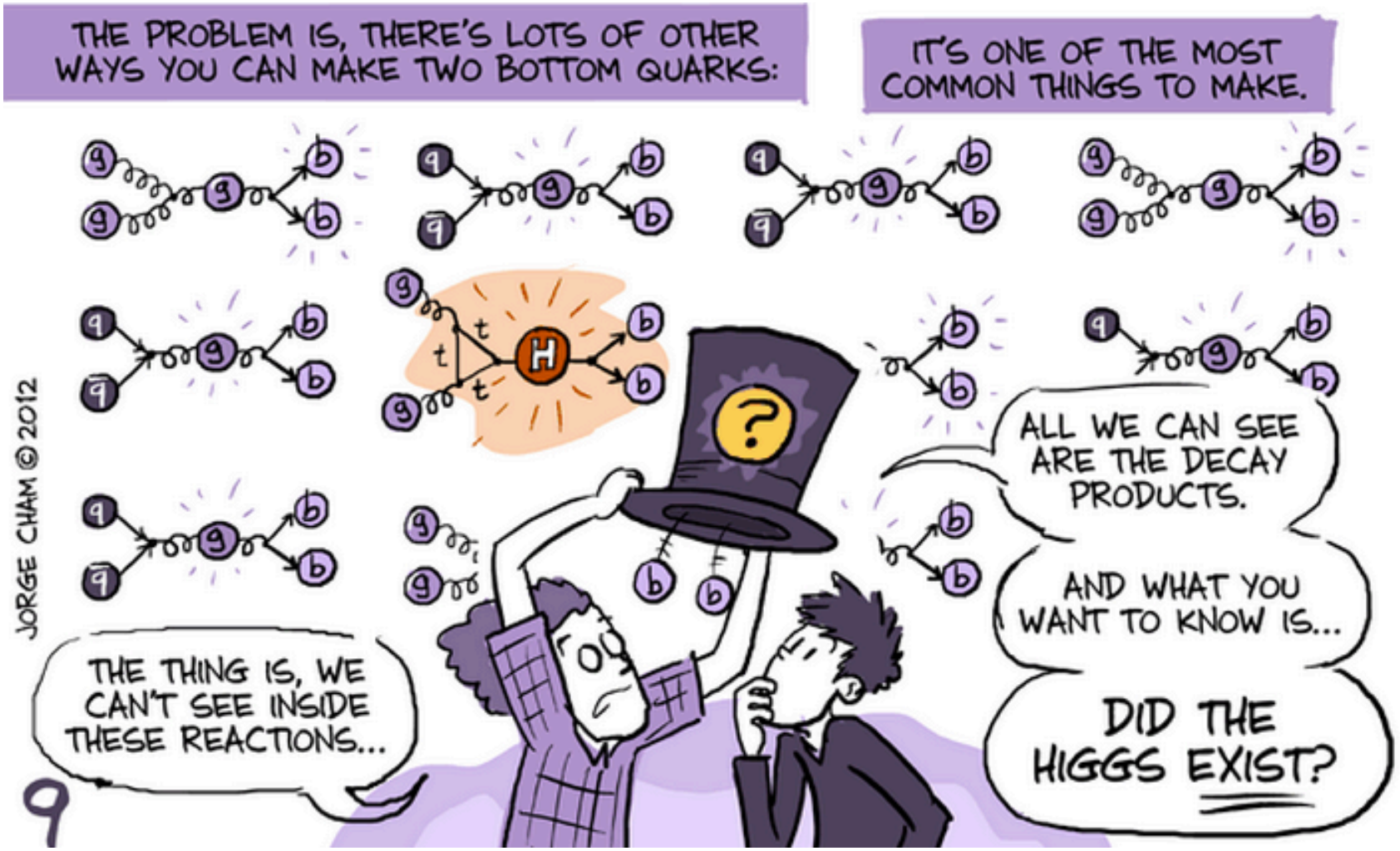
2.



Making a new particle

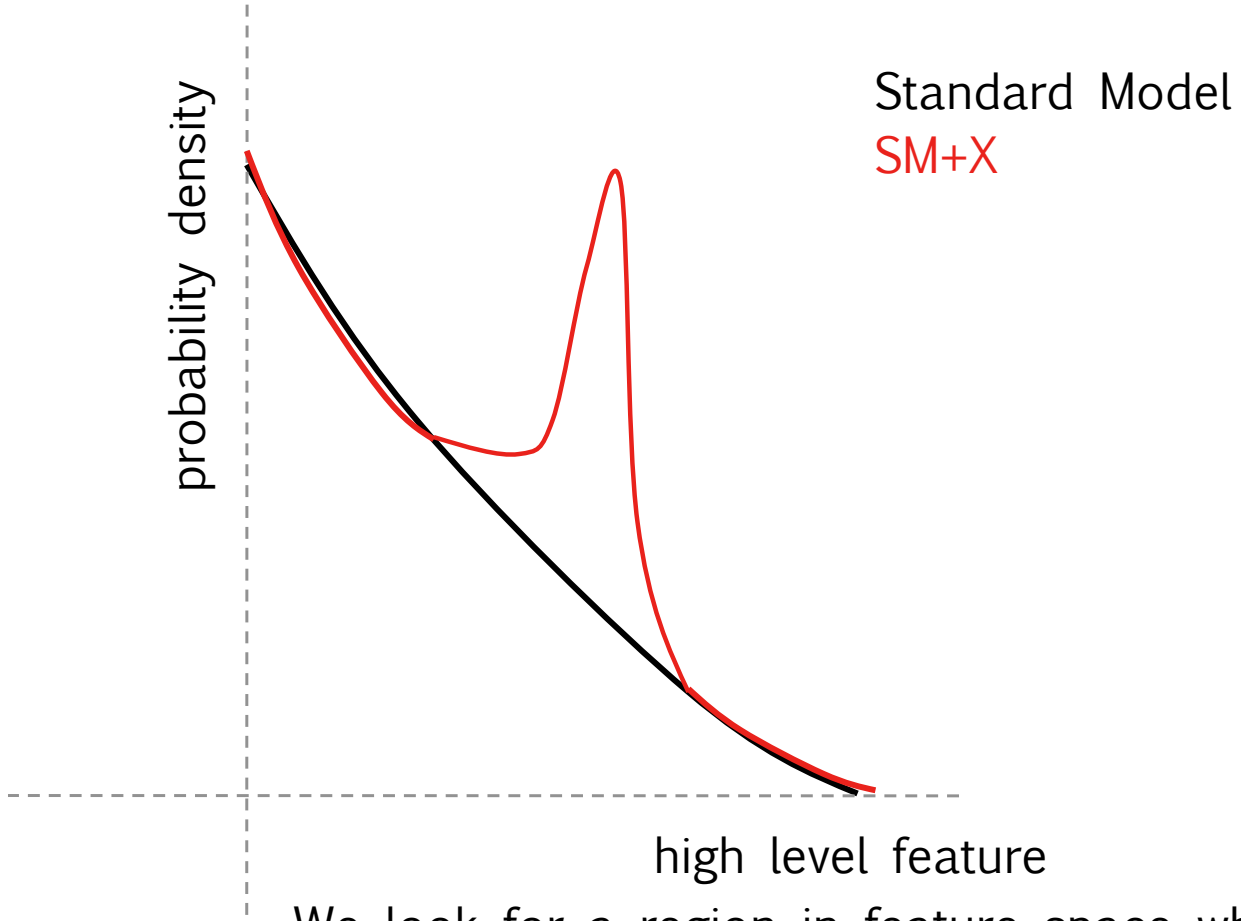


Backgrounds



Hypothesis Testing

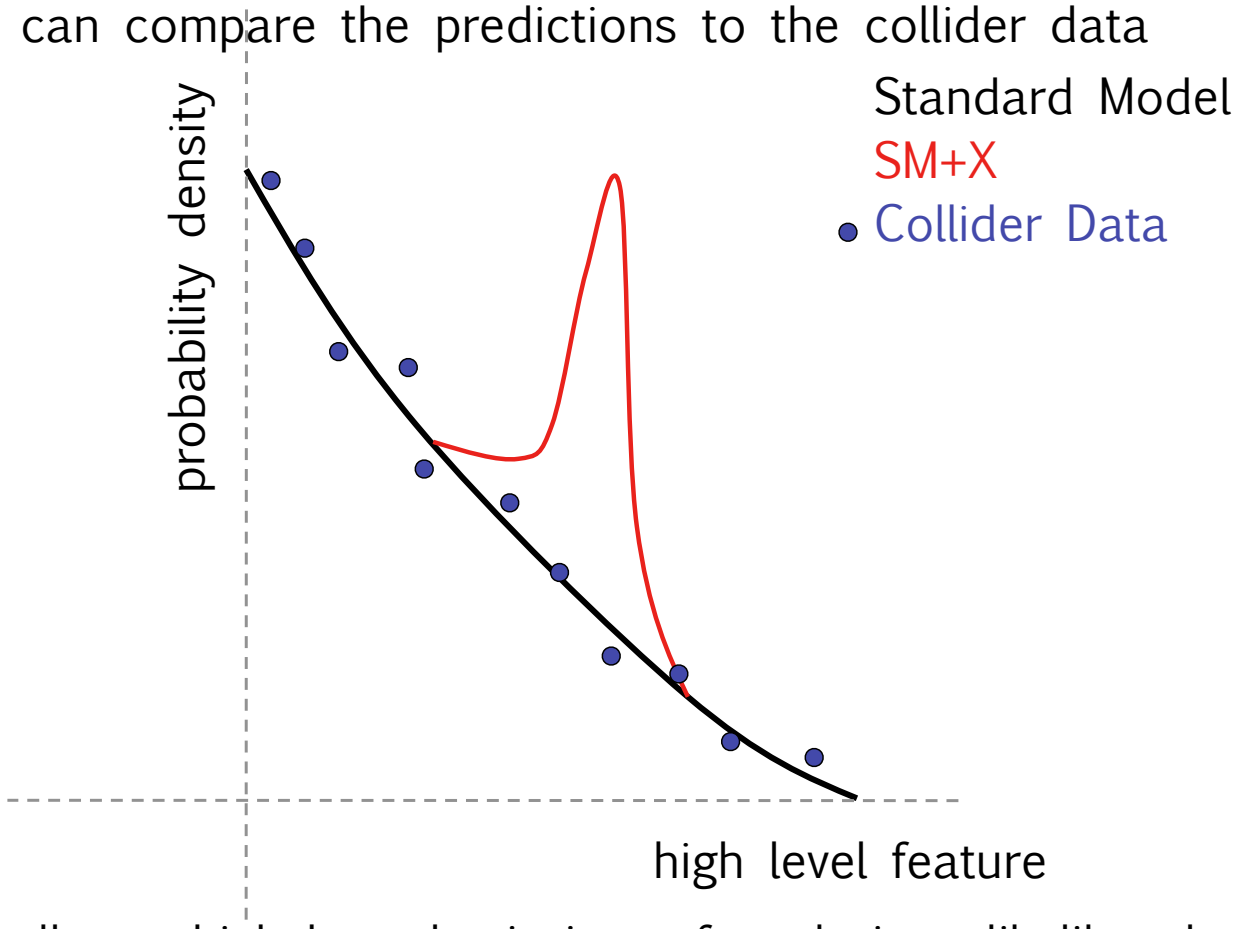
Which are boiled down to a few (10-50) high level features



We look for a region in feature space where
the two hypotheses have large differences in their predictions.

Hypothesis Testing

We can compare the predictions to the collider data



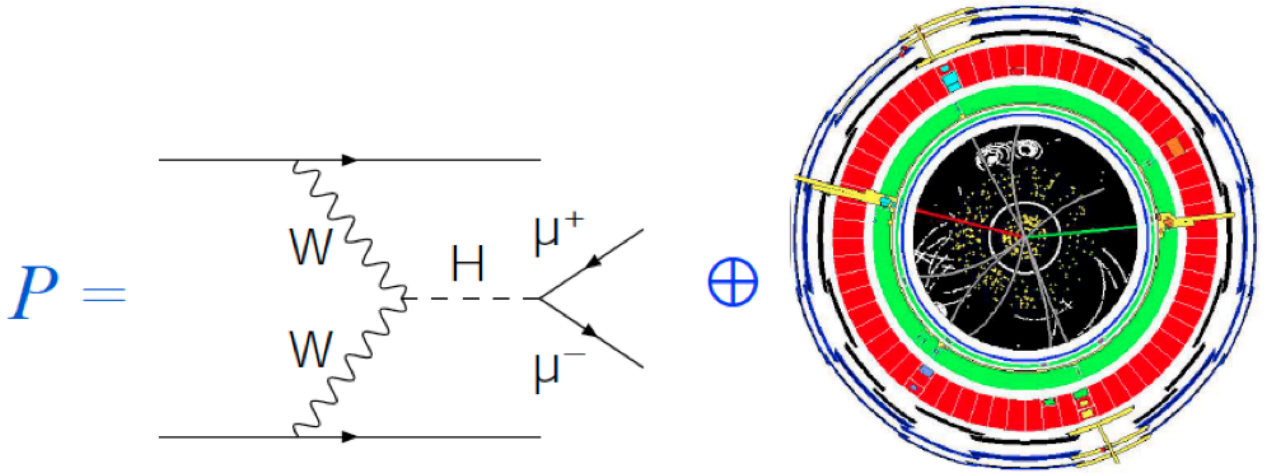
Which can tell us which hypothesis is preferred via a likelihood ratio:

$$\frac{L_{SM+X}}{L_{SM}} = \frac{P(\text{data} \mid \text{SM+X})}{P(\text{data} \mid \text{SM})}$$

P(data | SM)

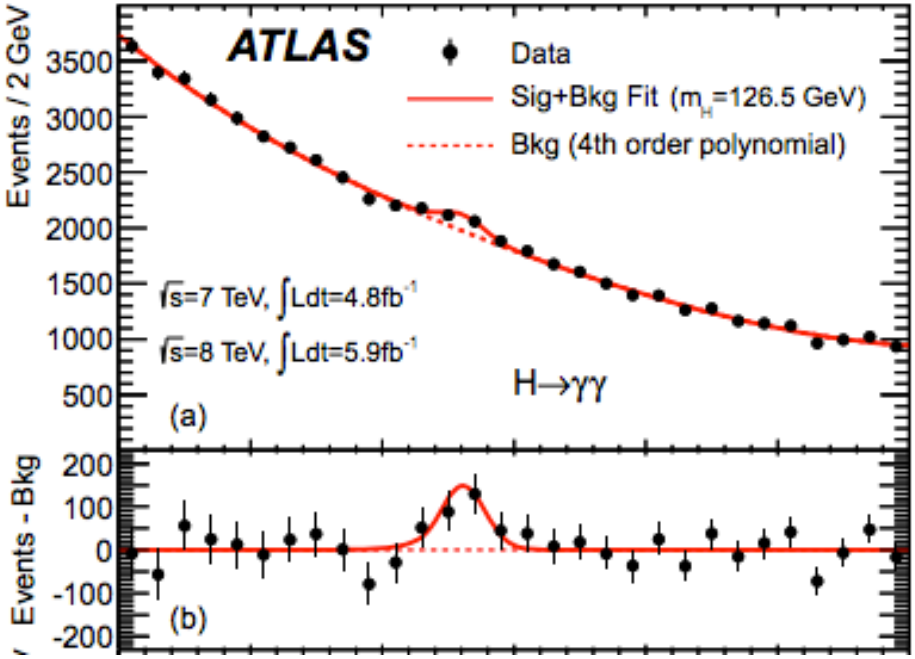
We don't know how to calculate

P(data | SM)

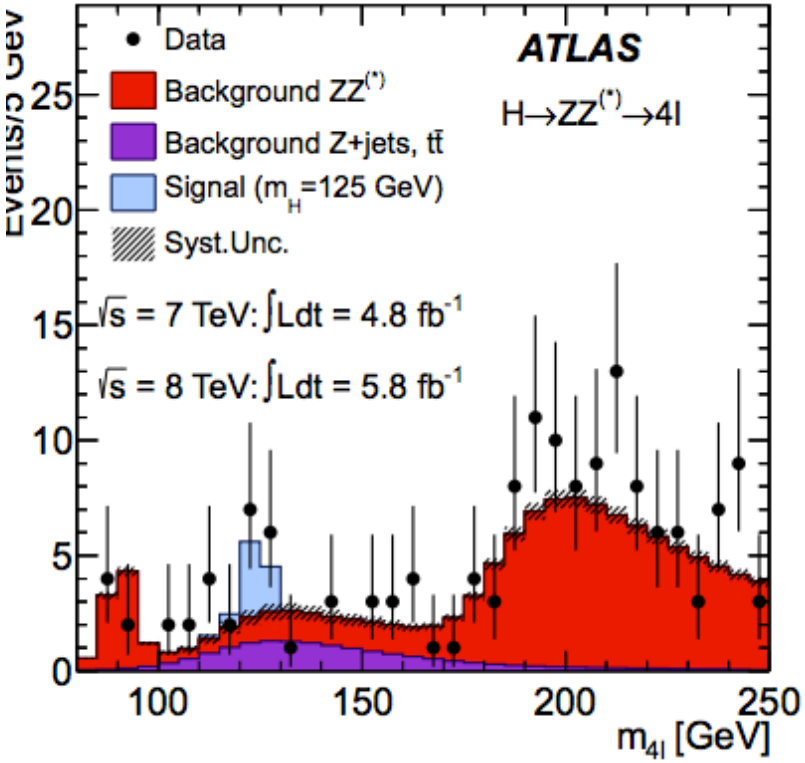


ATLAS

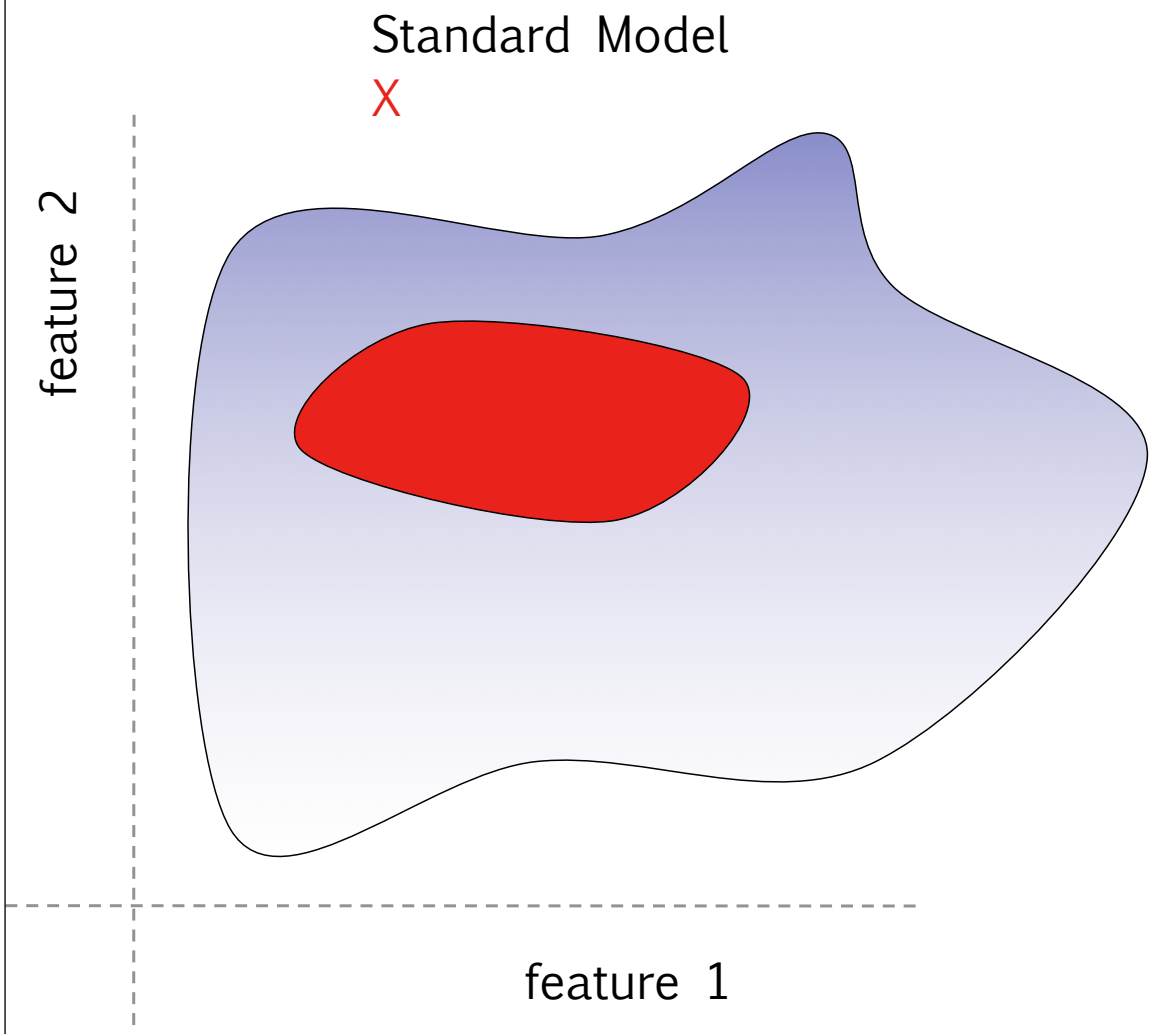
Two photons



Two Zs to 4 leptons



Simulation



generate simulated collisions for both hypotheses

Use histograms to define

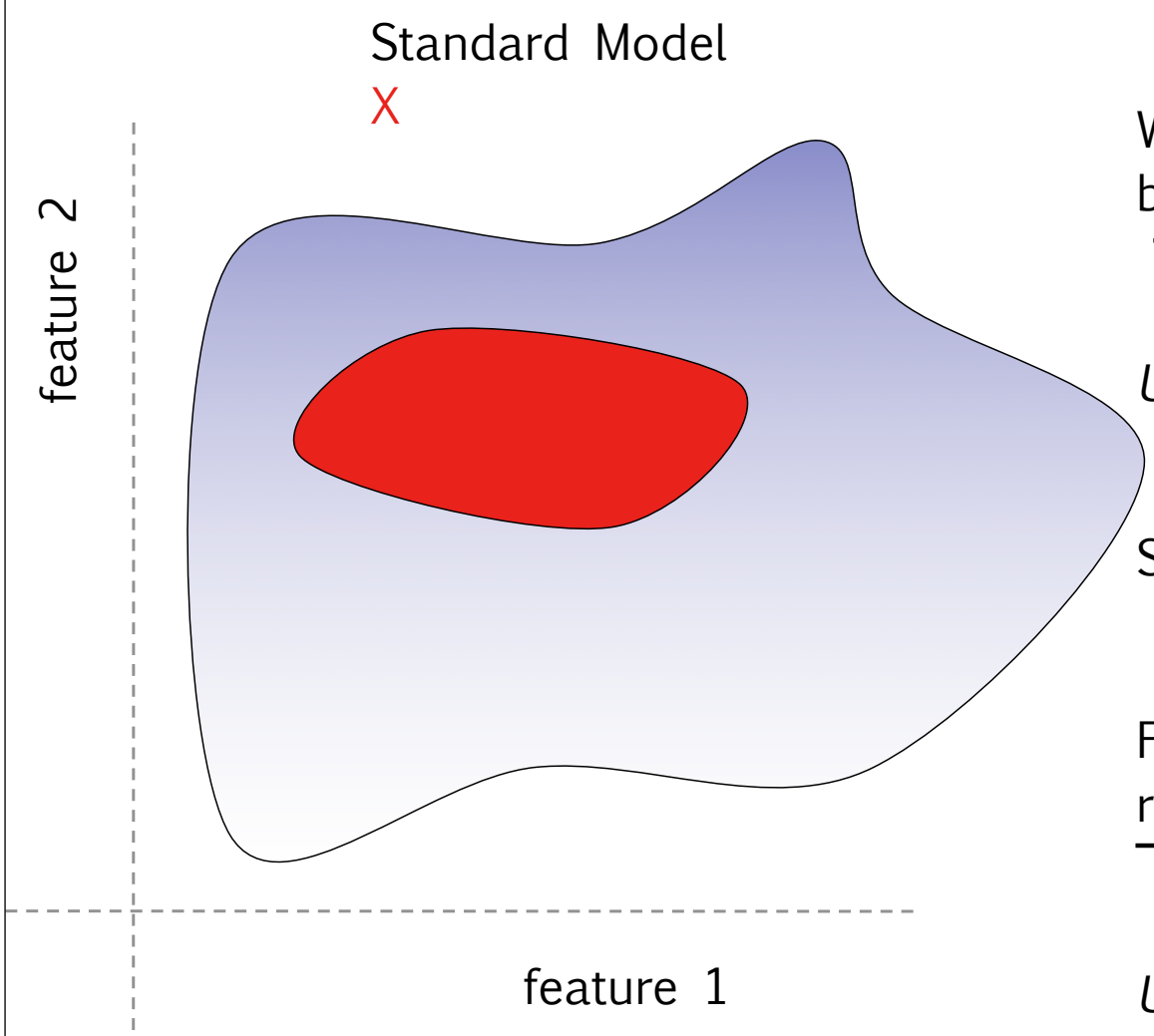
$$P(\text{data} \mid \text{SM+X}),$$

and

$$P(\text{data} \mid \text{SM})$$

(We've also tried kernel-based probability density estimates).

Dimensionality



Want to harness all differences between hypotheses.

⇩

Use as many features as possible.

Simulation is slow
(~1 - 10 min/collision)

Filling a *d*-dimensional histogram requires ~ 100^d simulated collisions!

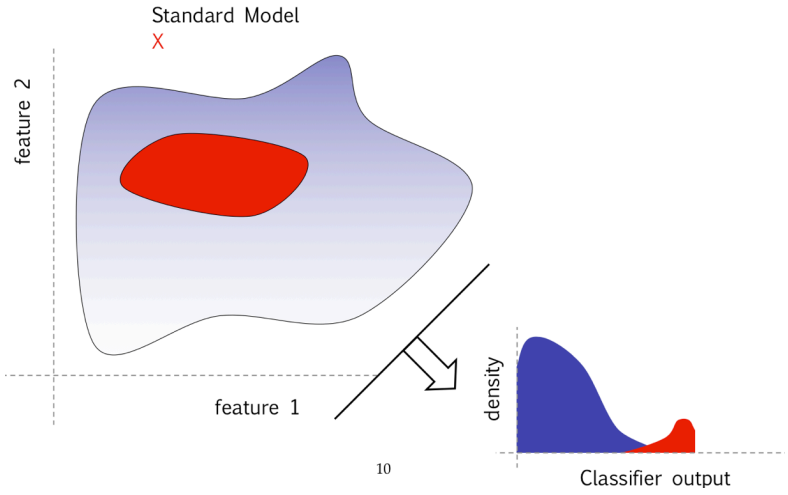
⇩

Use as few features as possible.

Function

Find a function

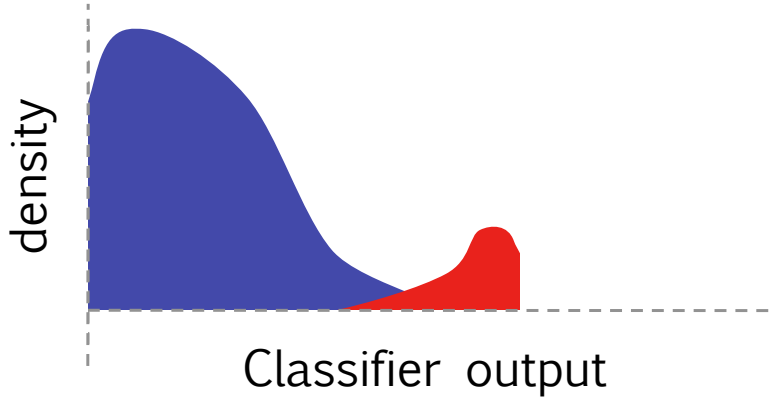
$$f(\mathcal{X}^N \rightarrow \mathcal{R}^1)$$



Neural networks can learn these shapes in high-dim and summarize in a 1D output

Dimensional Reduction

This dimensional reduction can be very helpful.



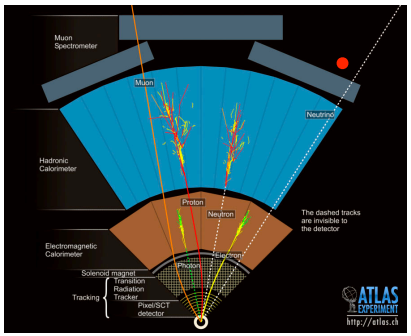
Summarize the differences between the hypotheses

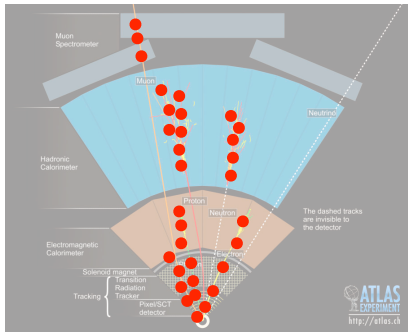
$$\frac{L_{SM+X}}{L_{SM}} = \frac{P(\text{data} \mid SM+X)}{P(\text{data} \mid SM)}$$

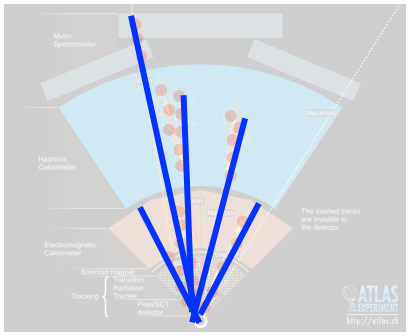
And require a histogram in only one dimension

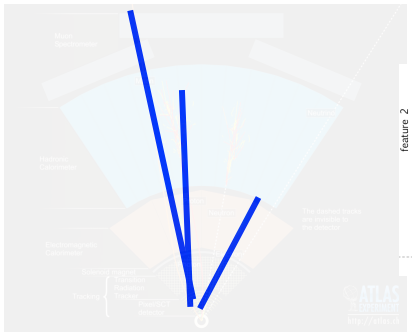
Dimensionality

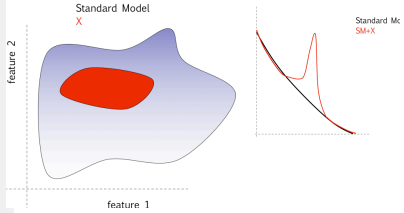
Raw	Sparsified	Reco	Select	Physics	Ana
1e7	1e4	100-ish*	50	10	1











ish*: dimensionality is variable here

So what?

What's wrong with this picture?

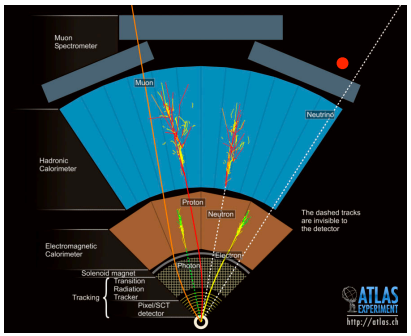
Nothing dramatic

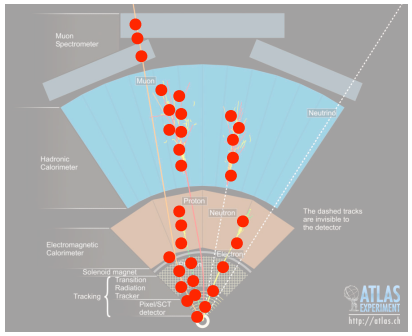
Each step is done using domain knowledge and decades of experience.

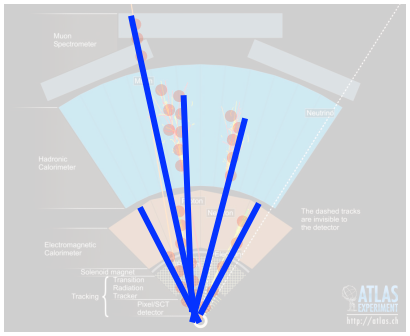
But is there information lost? Certainly.

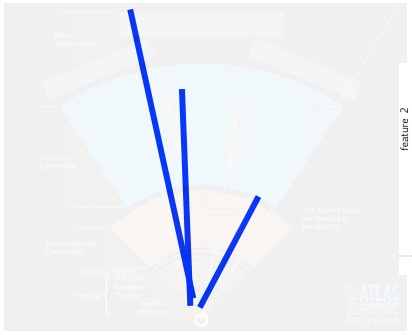
Dimensionality

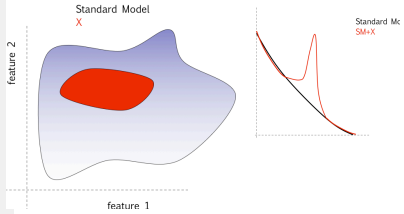
Raw	Sparsified	Reco	Select	Physics	Ana
1e7	1e4	100-ish*	50	10	1





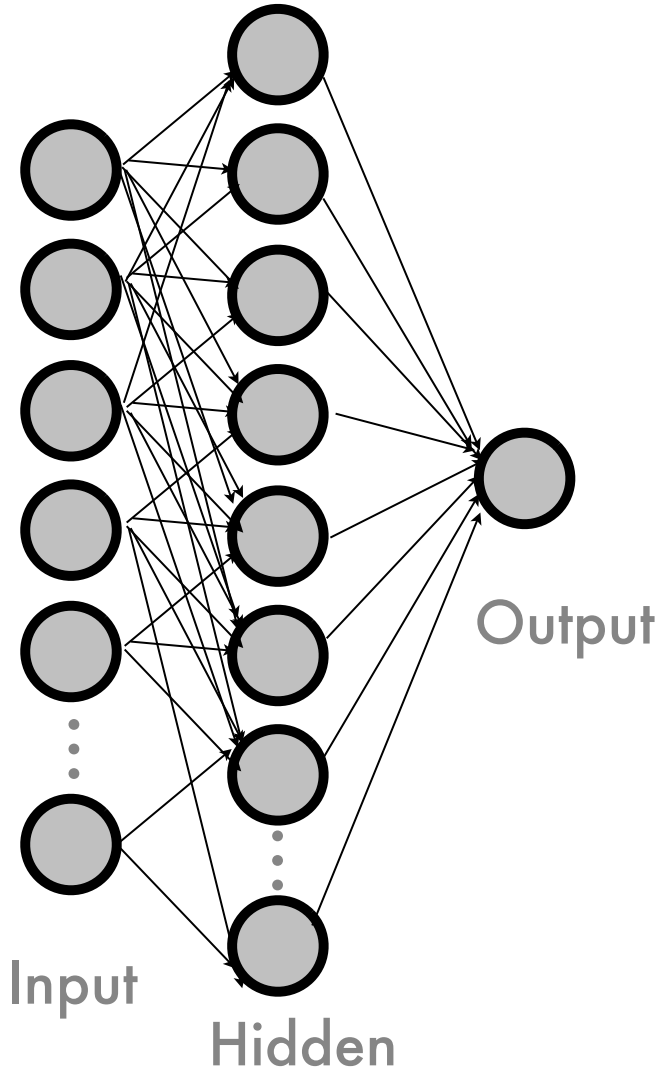






Neural Networks

Essentially a functional fit with many parameters



Function

Each neuron's output is a function of the weighted sum of inputs.

Goal

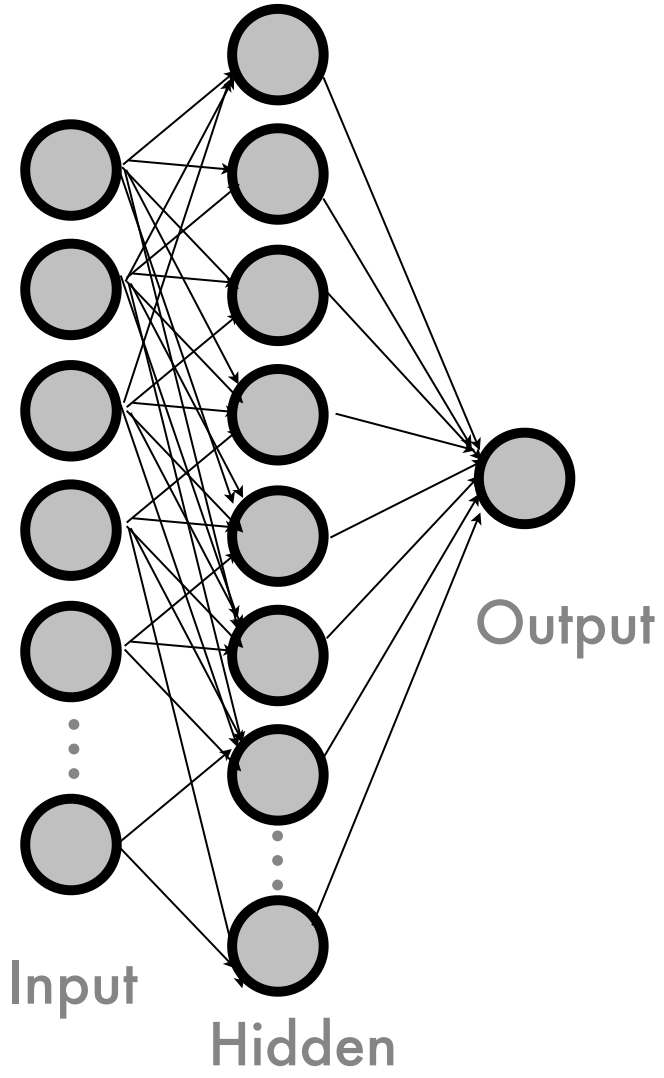
find set of weights which give most useful function

Learning

give examples, back-propagate error to adjust weights

Neural Networks

Essentially a functional fit with many parameters



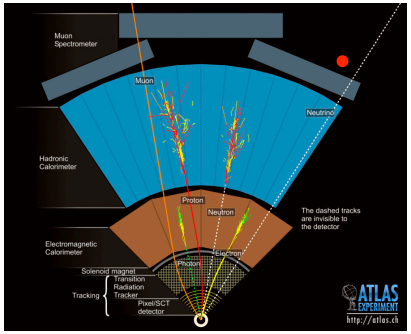
Problem:
Networks with > 1 layer are very difficult to train.

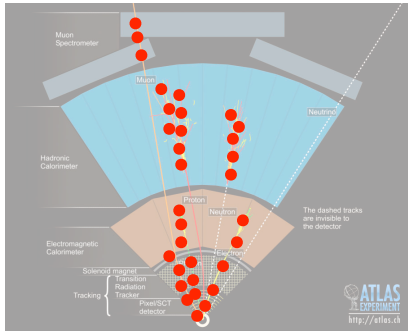
Consequence:
Networks are not good at learning non-linear functions.
(like invariant masses!)

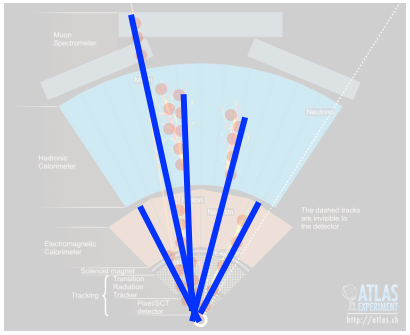
In short:
Can't just throw 4-vectors at NN.

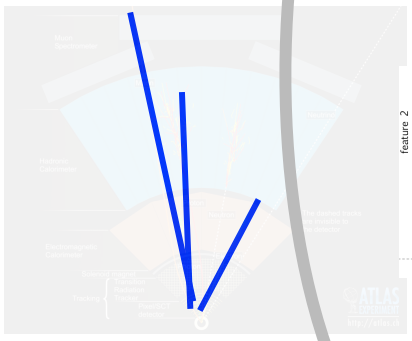
Summary

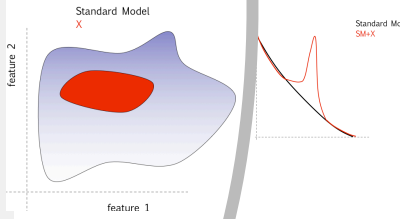
Raw	Sparsified	Reco	Select	Physics	Ana
1e7	1e4	100-ish*	50	10	1











Why we have this step

Search for Input

Can't just use reco or selected objects

Can't give it too many inputs

Painstaking search through input feature space.

ATLAS-CONF-2013-108

Variable	VBF			Boosted		
	$\tau_{\text{lep}}\tau_{\text{lep}}$	$\tau_{\text{lep}}\tau_{\text{had}}$	$\tau_{\text{had}}\tau_{\text{had}}$	$\tau_{\text{lep}}\tau_{\text{lep}}$	$\tau_{\text{lep}}\tau_{\text{had}}$	$\tau_{\text{had}}\tau_{\text{had}}$
$m_{\tau\tau}^{\text{MMC}}$	•	•	•	•	•	•
$\Delta R(\tau,\tau)$	•	•	•		•	•
$\Delta\eta(j_1, j_2)$	•	•	•			
m_{j_1,j_2}	•	•	•			
$\eta_{j_1} \times \eta_{j_2}$		•	•			
$p_{\text{T}}^{\text{Total}}$		•	•			
sum p_{T}					•	•
$p_{\text{T}}(\tau_1)/p_{\text{T}}(\tau_2)$					•	•
$E_{\text{T}}^{\text{miss}}\phi$ centrality		•	•	•	•	•
$x_{\tau 1}$ and $x_{\tau 2}$						•
$m_{\tau\tau,j_1}$				•		
m_{ℓ_1,ℓ_2}				•		
$\Delta\phi_{\ell_1,\ell_2}$				•		
sphericity				•		
$p_{\text{T}}^{\ell_1}$				•		
$p_{\text{T}}^{j_1}$				•		
$E_{\text{T}}^{\text{miss}}/p_{\text{T}}^{\ell_2}$				•		
m_{T}		•			•	
$\min(\Delta\eta_{\ell_1\ell_2,\text{jets}})$	•					
j_3 η centrality	•					
$\ell_1 \times \ell_2$ η centrality	•					
ℓ η centrality		•				
$\tau_{1,2}$ η centrality			•			

Table 3: Discriminating variables used for each channel and category. The filled circles identify which variables are used in each decay mode. Note that variables such as $\Delta R(\tau,\tau)$ are defined either between the two leptons, between the lepton and τ_{had}, or between the two τ_{had} candidates, depending on the decay mode.

Search for Input

Can't just use reco or selected objects

Can't give many inputs

Painstaking se through input feature space.

ATLAS-CONF-2013-108

Show Glazers talk on feature selection

		Boosted		
		$\tau_{\text{lep}}\tau_{\text{lep}}$	$\tau_{\text{lep}}\tau_{\text{had}}$	$\tau_{\text{had}}\tau_{\text{had}}$
		•	•	•
			•	•
			•	•
			•	•
		•	•	•
				•
		•		
		•		
		•		
	•		•	
$\min(\Delta\eta_{\ell_1,\ell_2,\text{jets}})$	•			
j_3 η centrality	•			
$\ell_1 \times \ell_2$ η centrality	•			
ℓ η centrality		•		
$\tau_{1,2}$ η centrality			•	

Table 3: Discriminating variables used for each channel and category. The filled circles identify which variables are used in each decay mode. Note that variables such as $\Delta R(\tau,\tau)$ are defined either between the two leptons, between the lepton and τ_{had}, or between the two τ_{had} candidates, depending on the decay mode.

Search for Input

Can't just use reco or selected objects

Can't give many inpu

Painstaking search through input feature space.

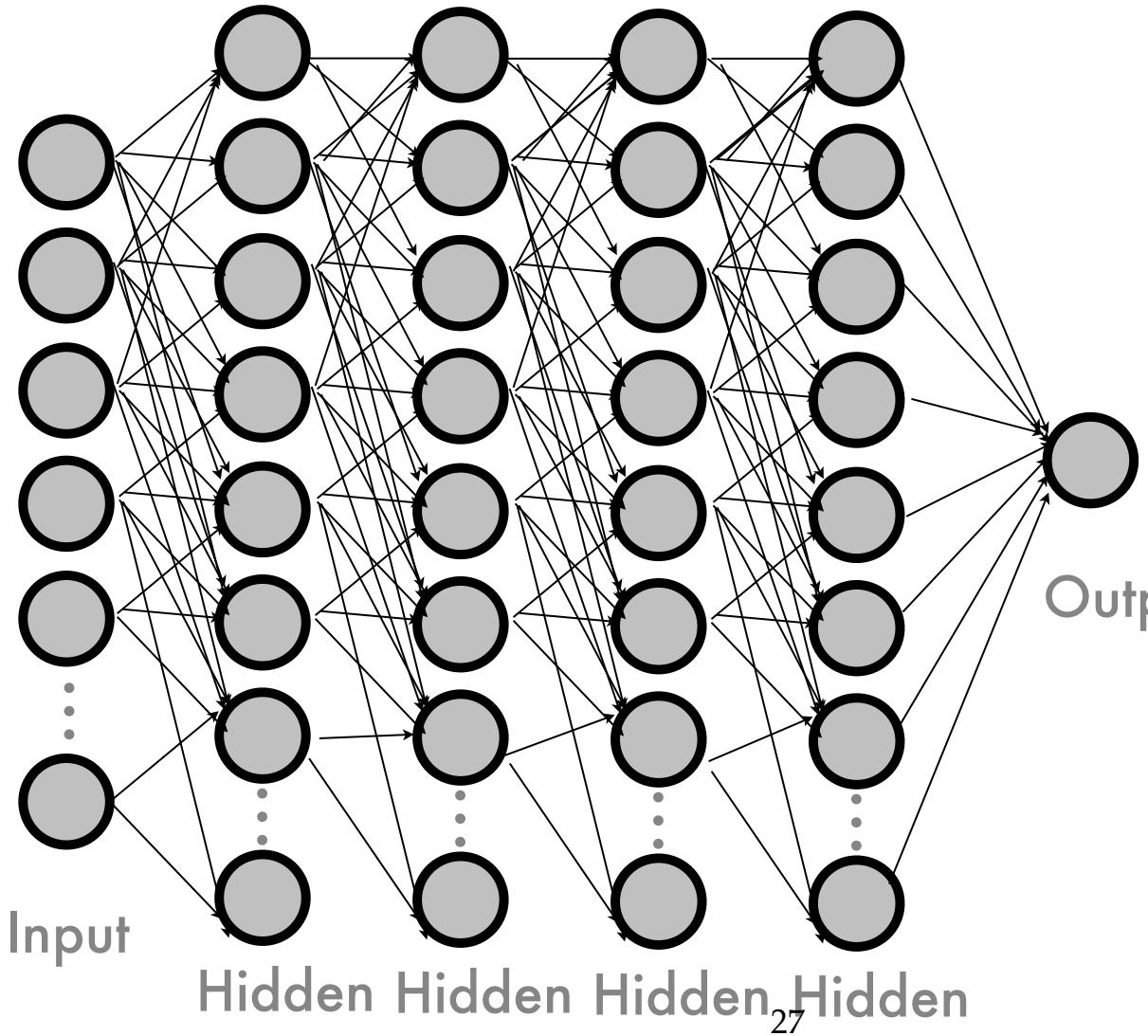
ATLAS-CONF-2013-108

Also true for BDTs, SVNs, etc

Variable	VBF		Boosted		
	$\tau_{\text{lep}}\tau_{\text{lep}}$	$\tau_{\text{had}}\tau_{\text{had}}$	$\tau_{\text{lep}}\tau_{\text{lep}}$	$\tau_{\text{lep}}\tau_{\text{had}}$	$\tau_{\text{had}}\tau_{\text{had}}$
$m_{\tau\tau}^{\text{MMC}}$			•	•	•
$\Delta R(\tau_{\text{lep}}, \tau_{\text{had}})$				•	•
$\Delta R(\tau_{\text{had}}, \tau_{\text{had}})$					
$\Delta\eta(\tau_{\text{had}}, \tau_{\text{had}})$					
$\Delta\phi(\tau_{\text{had}}, \tau_{\text{had}})$					
$\Delta\phi(\tau_{\text{had}}, \tau_{\text{had}})$					
$\Delta\phi(\ell, \tau_{\text{had}})$				•	•
$\Delta\phi(\ell, \tau_{\text{had}})$				•	•
$\Delta\phi(\ell, \tau_{\text{had}})$				•	•
$\Delta\phi(\ell, \tau_{\text{had}})$					•
$\Delta\phi(\ell, \tau_{\text{had}})$			•		
$\Delta\phi(\ell, \tau_{\text{had}})$			•		
$\Delta\phi(\ell, \tau_{\text{had}})$			•		
$\Delta\phi(\ell, \tau_{\text{had}})$			•		
$p_{\text{T}}^{j_1}$			•		
$E_{\text{T}}^{\text{miss}}/p_{\text{T}}^{\ell_2}$			•		
m_{T}	•			•	
$\min(\Delta\eta_{\ell_1,\ell_2,\text{jets}})$	•				
j_3 η centrality	•				
$\ell_1 \times \ell_2$ η centrality	•				
ℓ η centrality		•			
$\tau_{1,2}$ η centrality		•			

Table 3: Discriminating variables used for each channel and category. The filled circles identify which variables are used in each decay mode. Note that variables such as $\Delta R(\tau, \tau)$ are defined either between the two leptons, between the lepton and τ_{had}, or between the two τ_{had} candidates, depending on the decay mode.

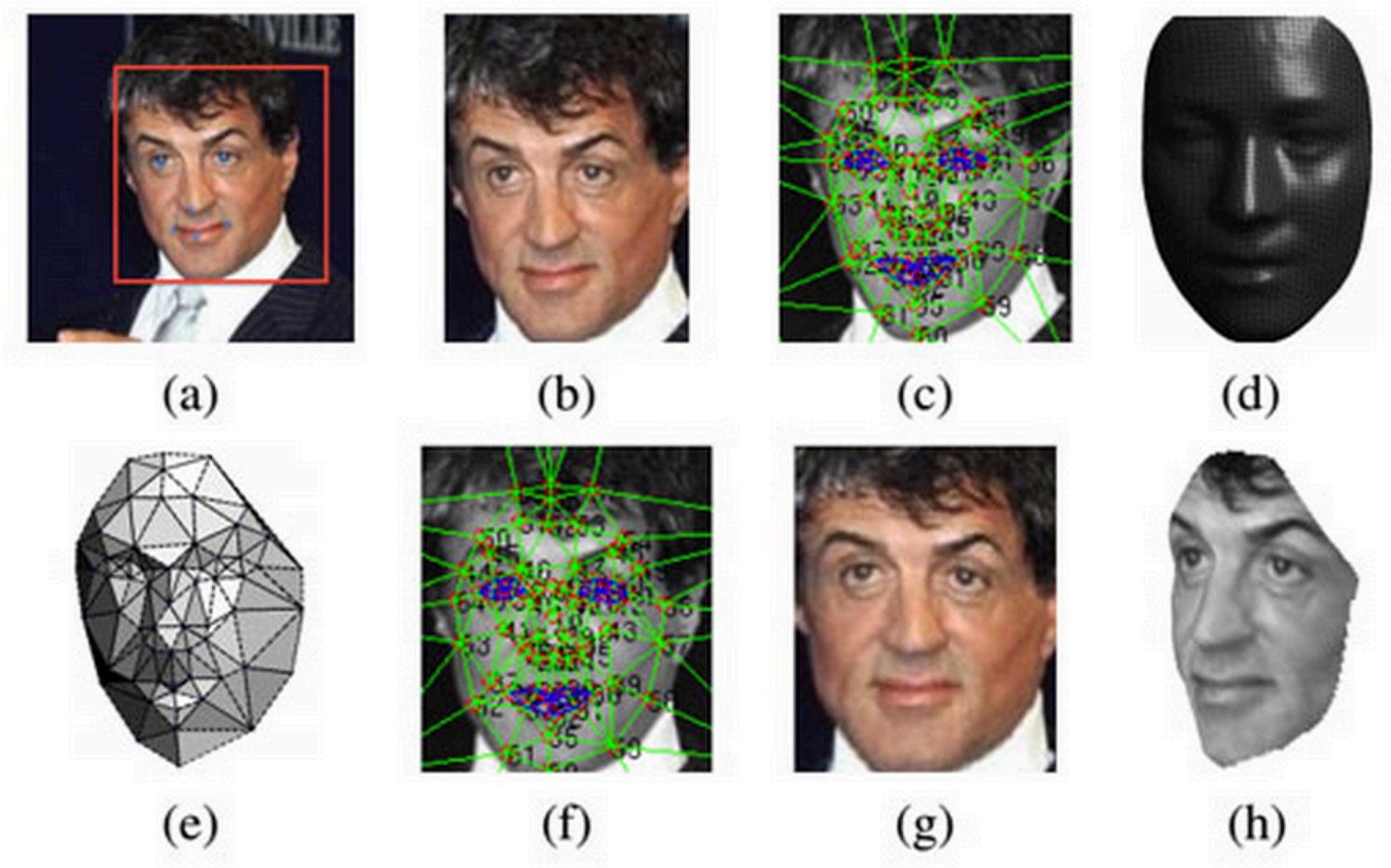
Deep networks



New tools let us train deep networks.

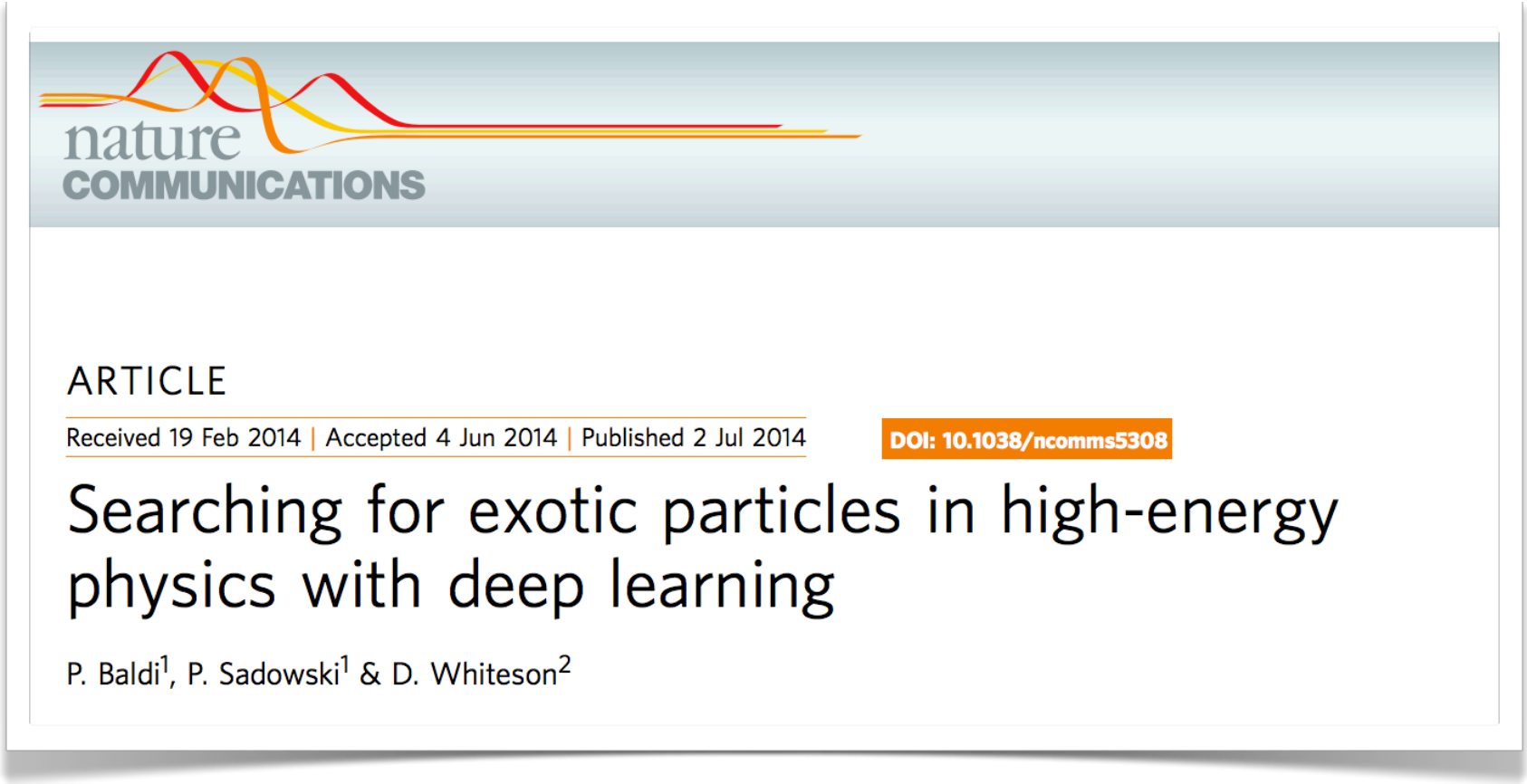
How well do they work?

Real world applications



Head turn: DeepFace uses a 3-D model to rotate faces, virtually, so that they face the camera. Image (a) shows the original image, and (g) shows the final, corrected version.

Paper



nature COMMUNICATIONS

ARTICLE

Received 19 Feb 2014 | Accepted 4 Jun 2014 | Published 2 Jul 2014

DOI: 10.1038/ncomms5308

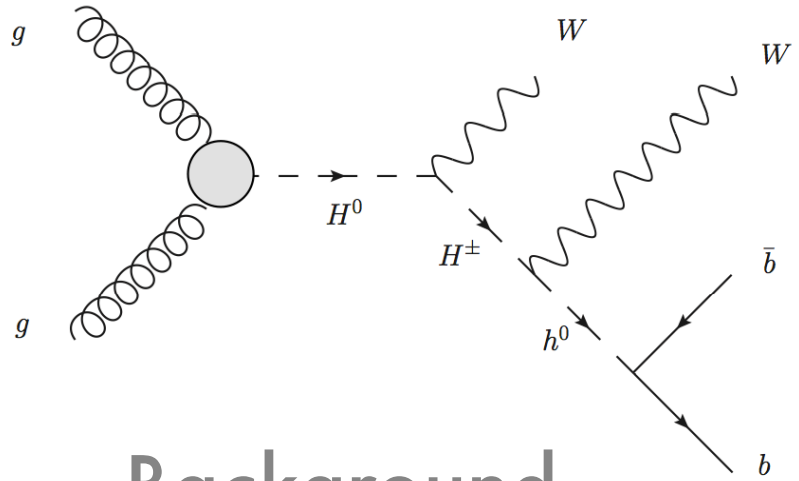
Searching for exotic particles in high-energy physics with deep learning

P. Baldi[1], P. Sadowski[1] & D. Whiteson[2]

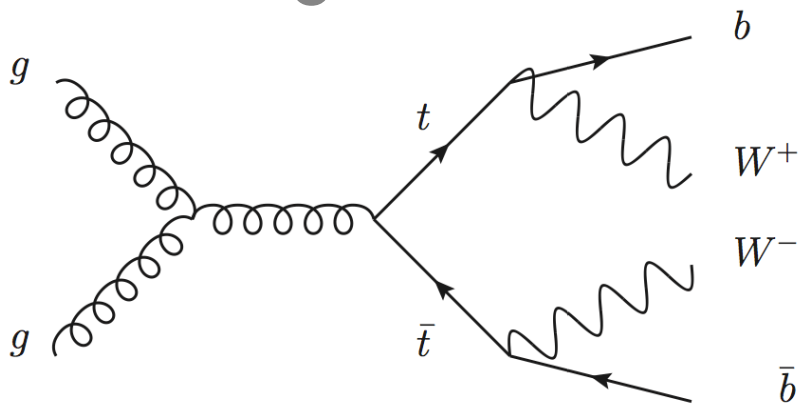
arXiv: 1402.4735

Benchmark problem

Signal



Background



Can deep networks automatically discover useful variables?

4-vector inputs

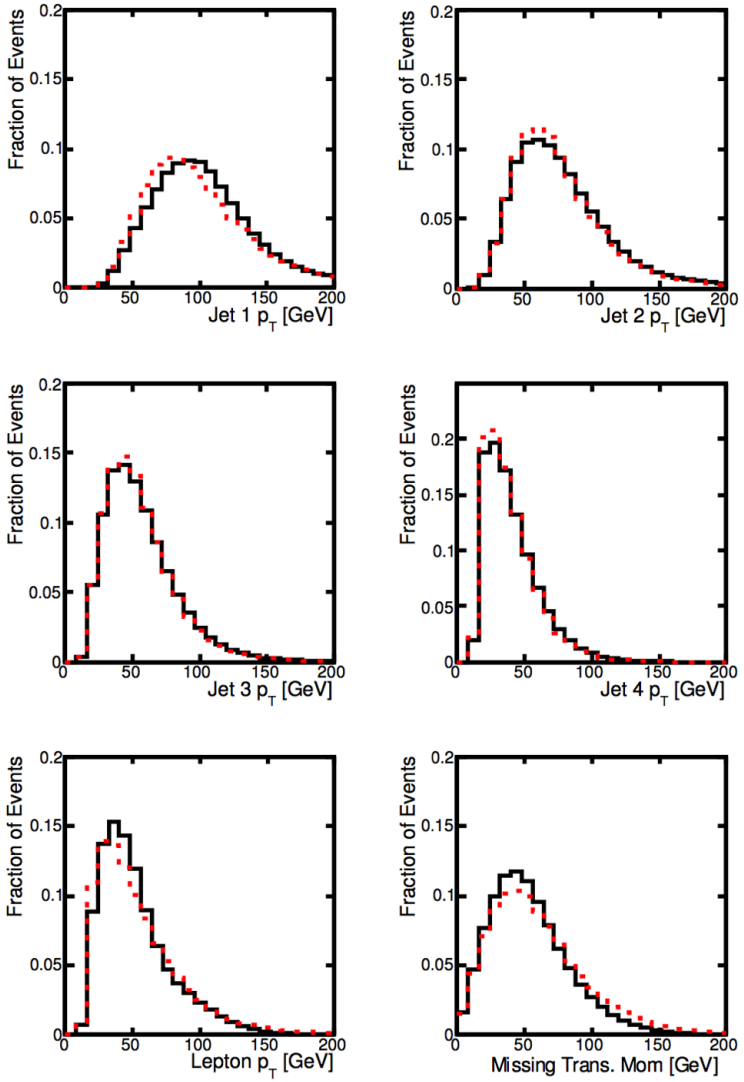
<u>21 Low-level vars</u>

jet+lepton mom. (3x5)

missing ET (2)

jet btags (4)

Not much separation visible in 1D projections



4-vector inputs

7 High-level vars

m(WWbb)

m(Wbb)

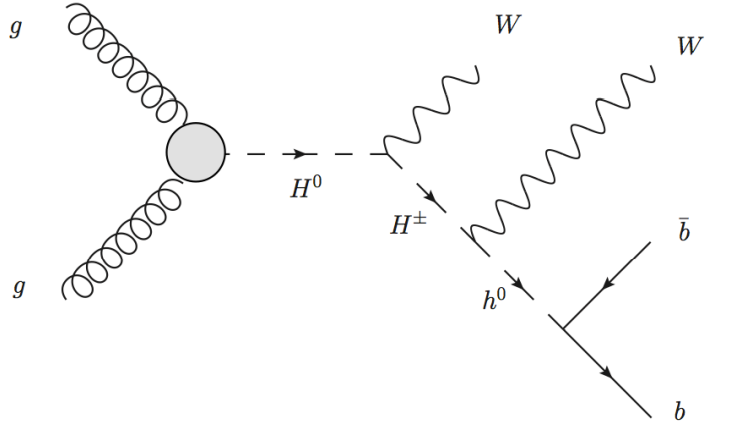
m(bb)

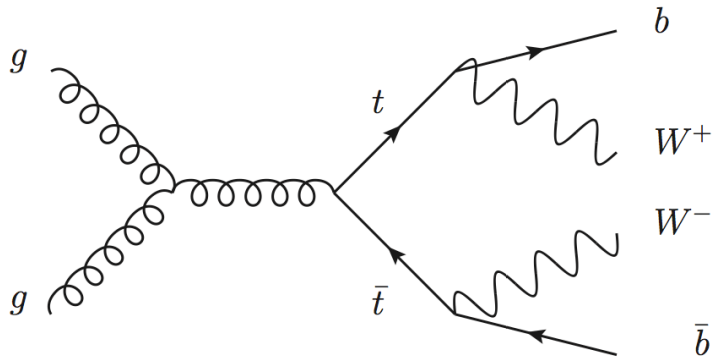
m(bjj)

m(jj)

m(lv)

m(blv)





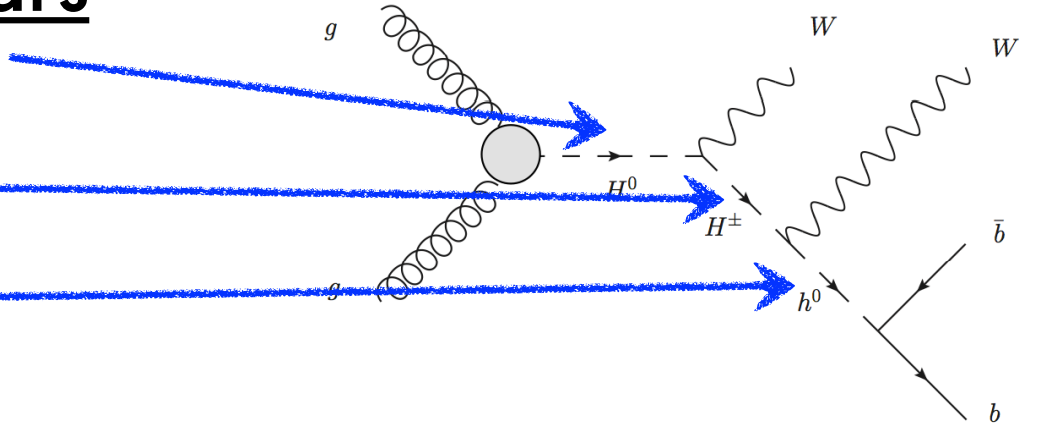
4-vector inputs

7 High-level vars

m(WWbb)

m(Wbb)

m(bb)

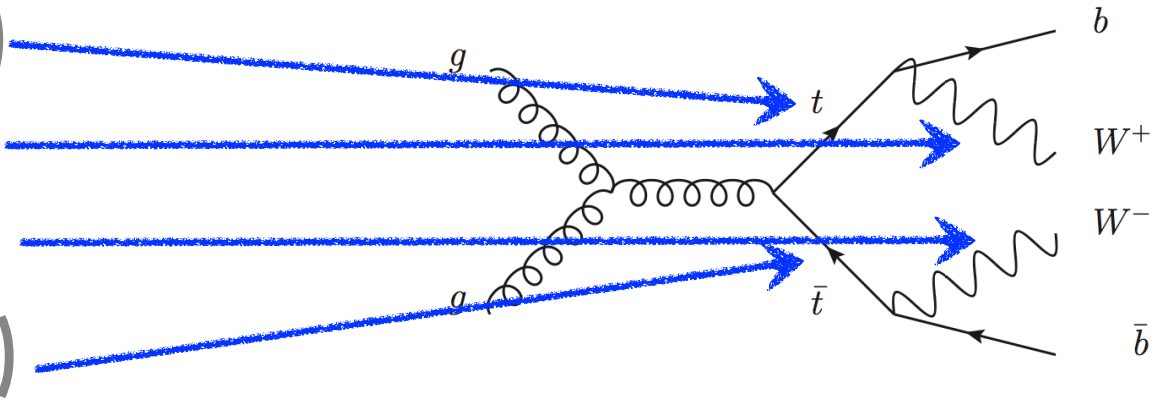


m(bjj)

m(jj)

m(lv)

m(blv)



4-vector inputs

7 High-level vars

m(WWbb)

m(Wbb)

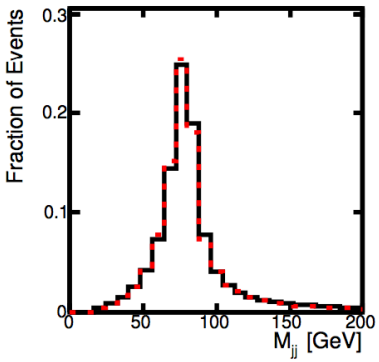
m(bb)

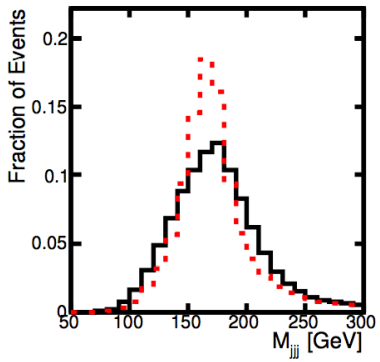
m(bjj)

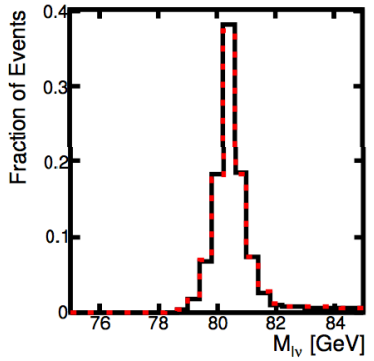
m(jj)

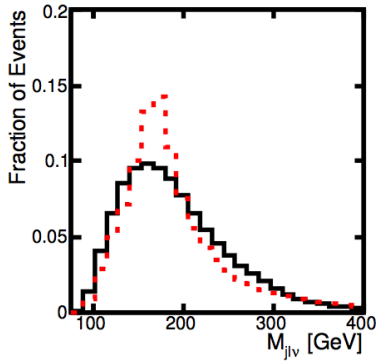
m(lv)

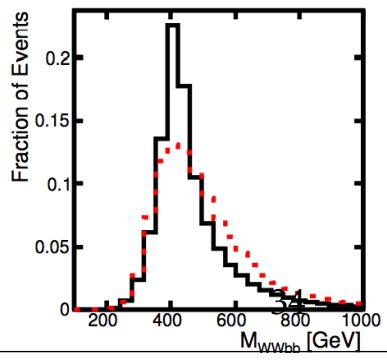
m(blv)

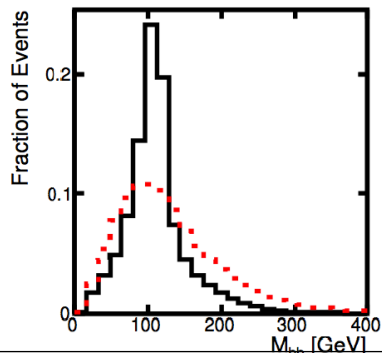


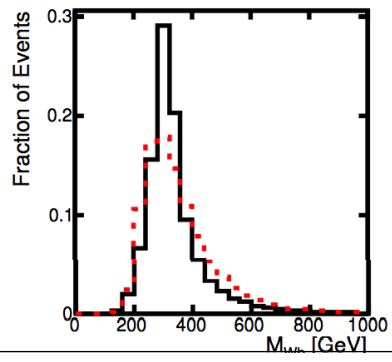


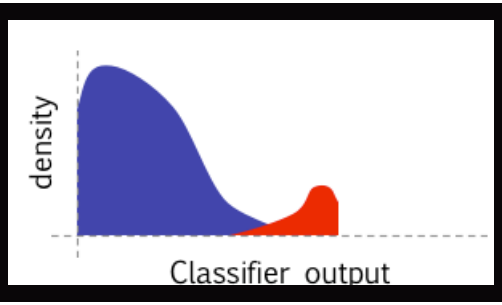




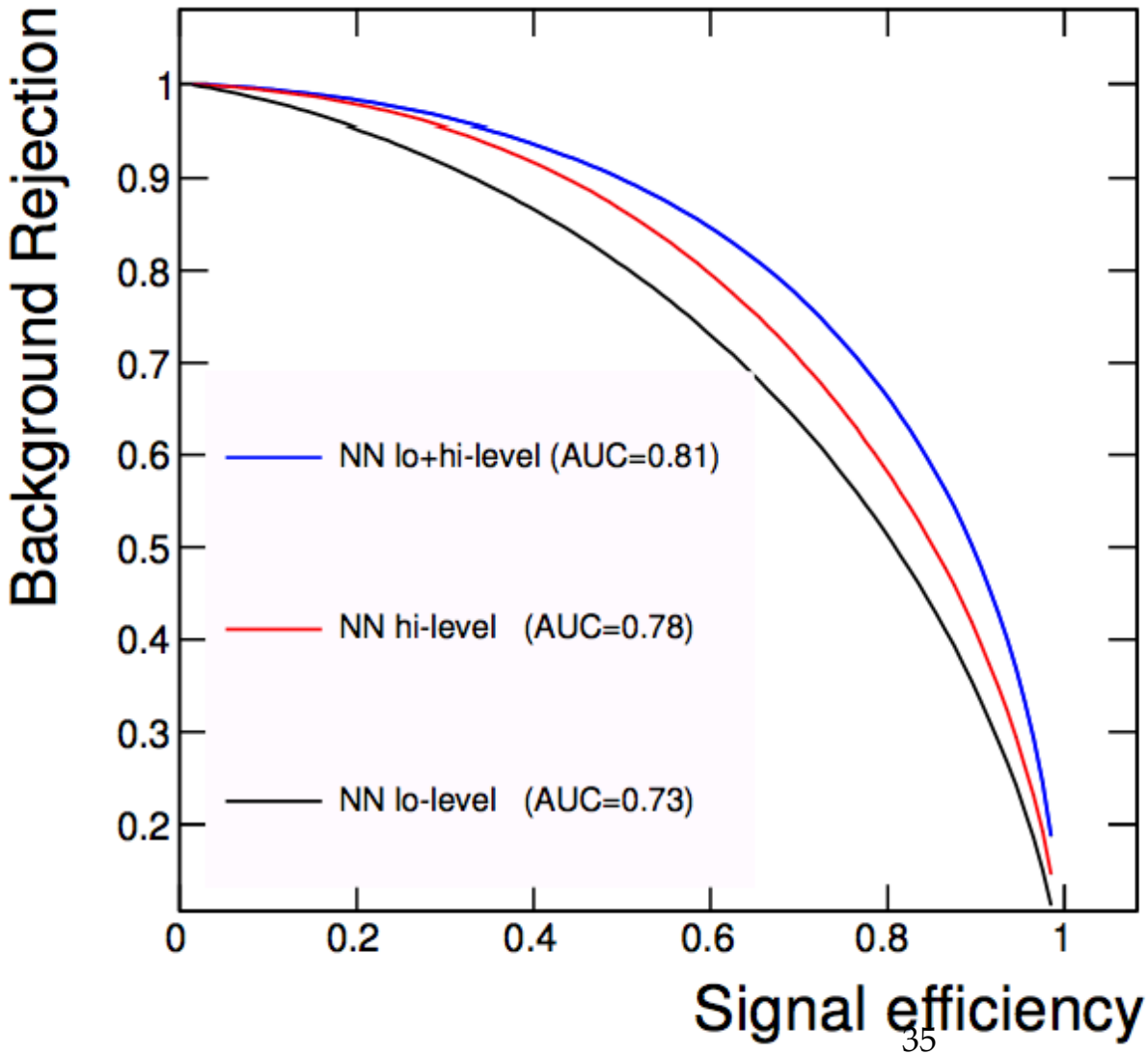








Standard NNs



Results

Adding hi-level boosts performance

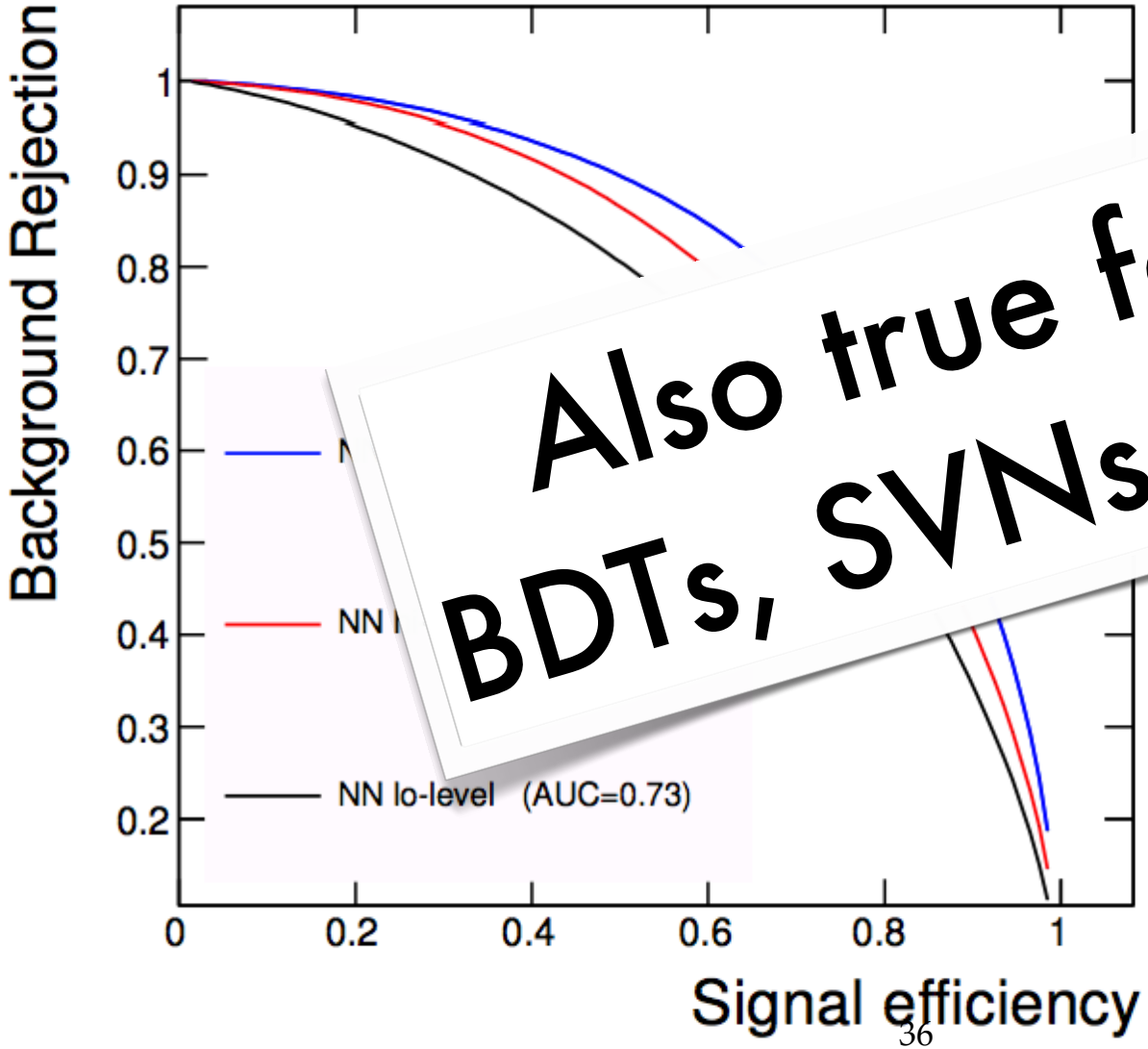
Better: lo+hi-level.

Conclude:

NN can't find hi-level vars.

Hi-level vars do not have all info

Standard NNs



Also true for BDTs, SVNs, etc

Results

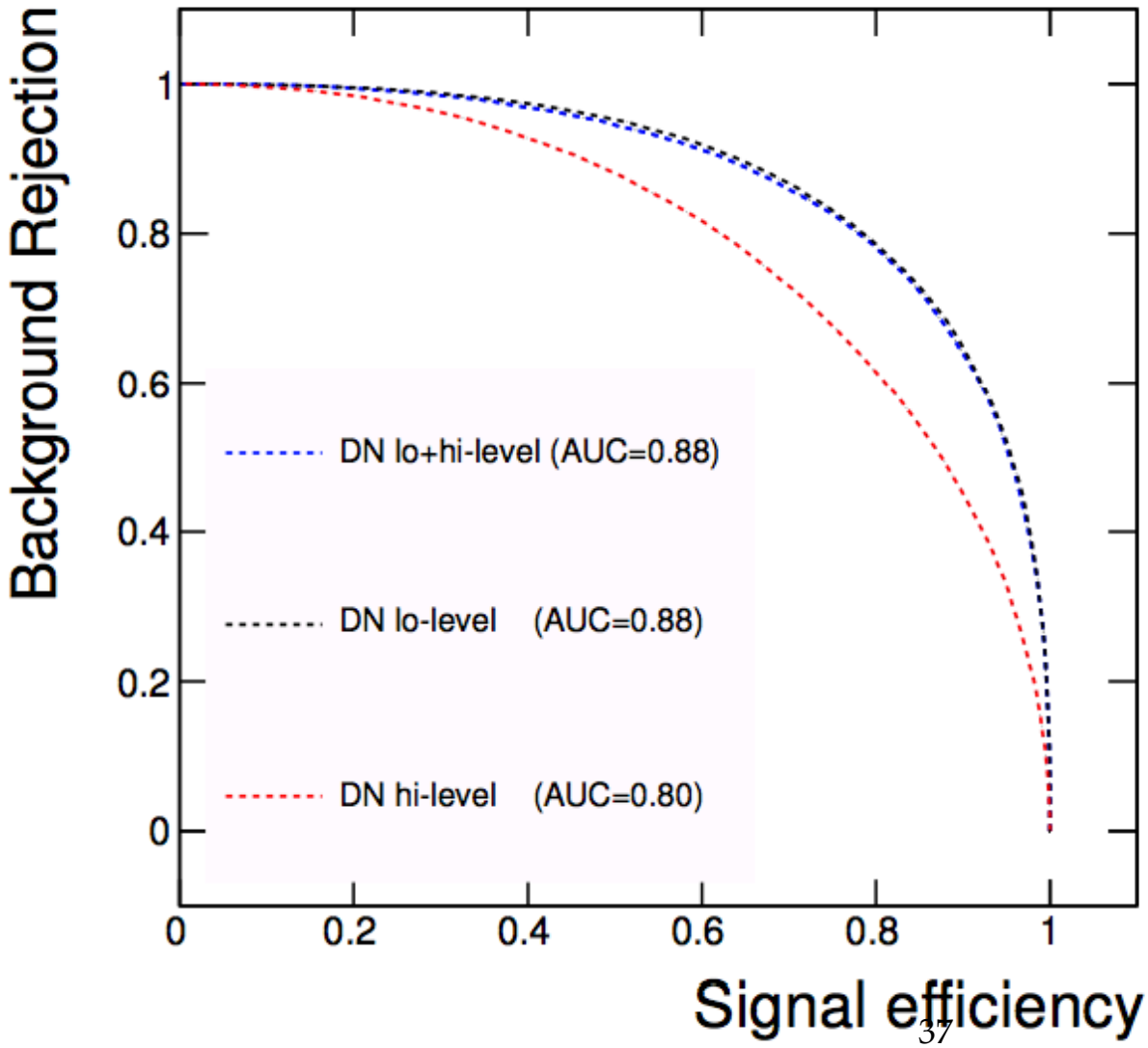
g hi-level
performance
lo+hi-level.

Conclude:

NN can't find
hi-level vars.

Hi-level vars
do not have all info

Deep Networks



Results

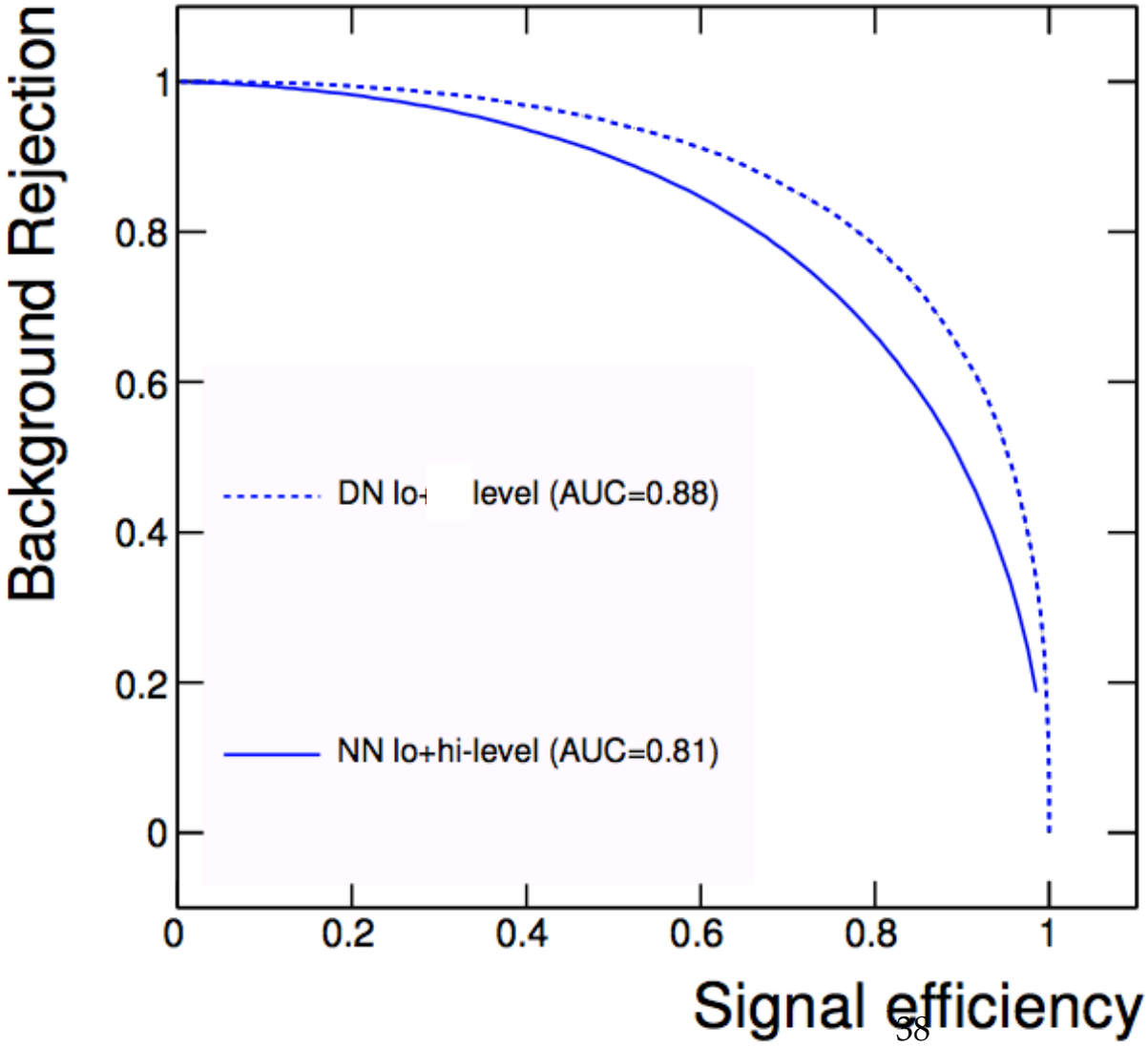
Lo+hi = lo.

Conclude:

DN can find
hi-level vars.

Hi-level vars
do not have all info
are unnecessary

Deep Networks



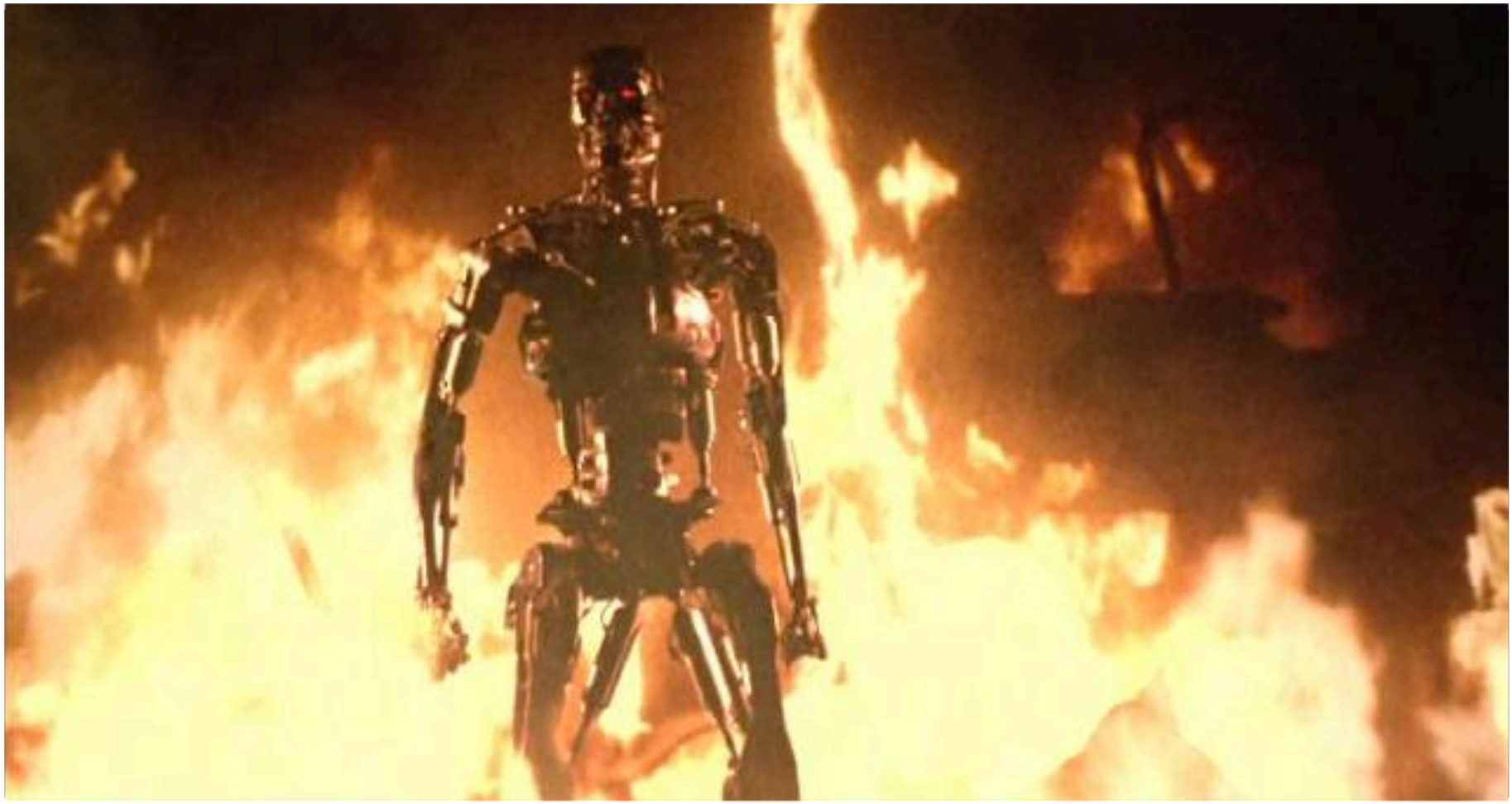
Results

DN > NN

Conclude:

DN does better than human assisted NN

The AIs win



Results

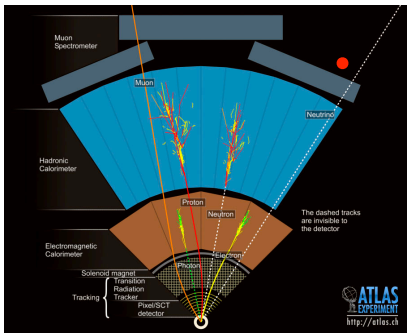
Identified example benchmark where traditional NNs fail to discover all discrimination power.

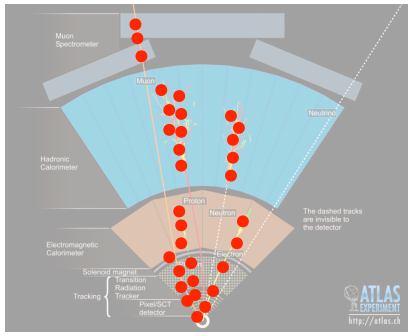
Adding human insight helps traditional NNs.

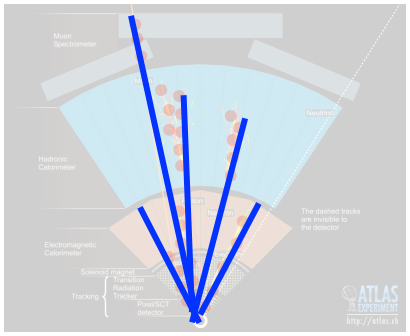
Deep networks succeed without human insight.
Outperform human-boosted traditional NNs.

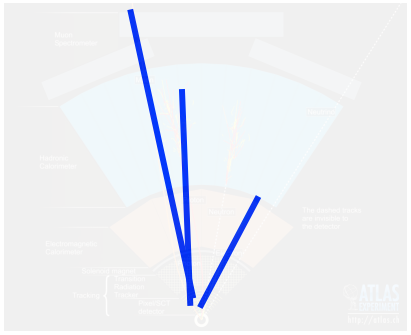
Dimensionality

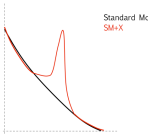
Raw	Sparsified	Reco	Select	Physics	Ana
1e7	1e4	100-ish*	50	10	1





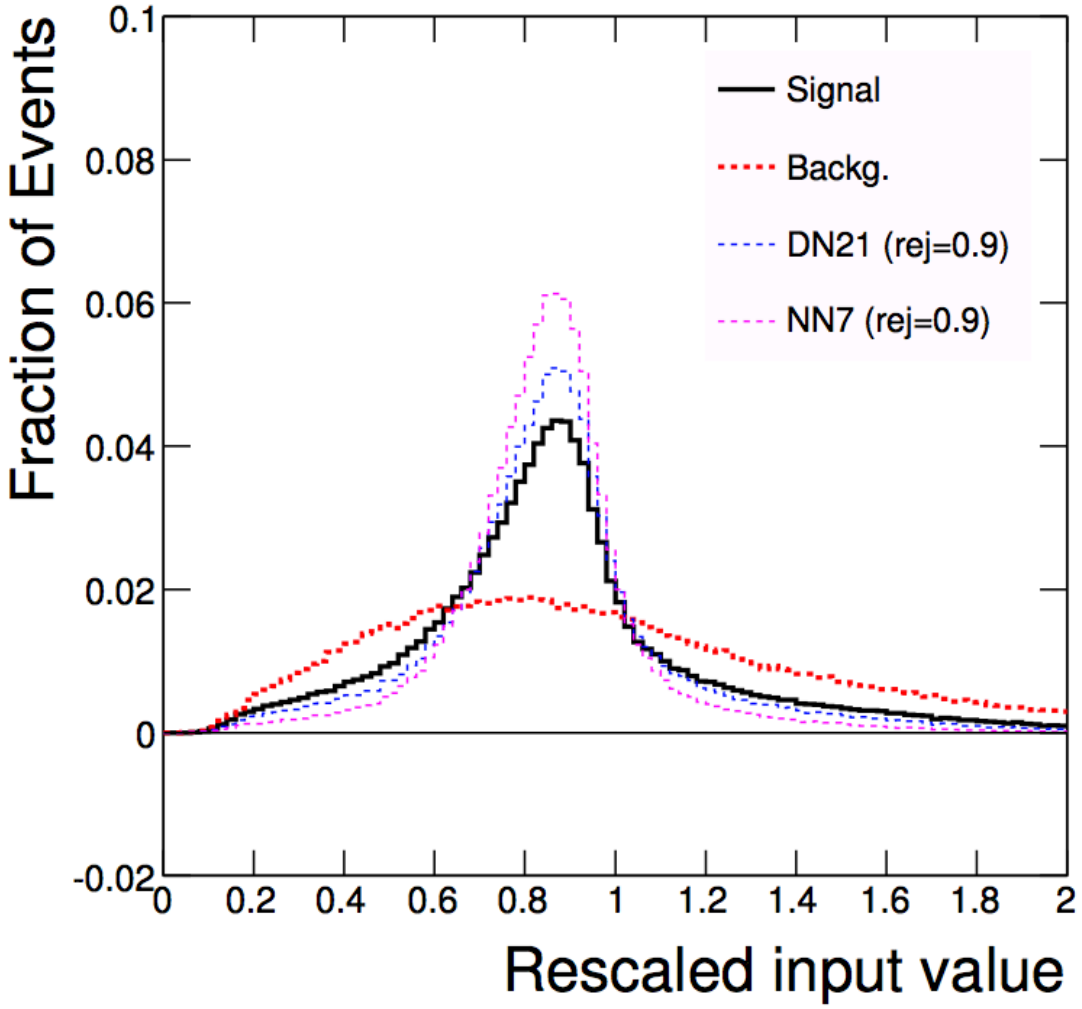






Why?

DN not as reliant on signal features. Cuts into background space.



Depth

Supplementary Table 3: **Study of network size and depth.** Comparison of shallow networks with different numbers of hidden units (single hidden layer), and deep networks with varying hidden layers in terms of the Area Under the ROC Curve (AUC) for the HIGGS benchmark. The deep networks have 300 units in each hidden layer.

	AUC		
Technique	Low-level	High-level	Complete
NN 300-hidden	0.733	0.777	0.816
NN 1000-hidden	0.788	0.783	0.841
NN 2000-hidden	0.787	0.788	0.842
NN 10000-hidden	0.790	0.789	0.841
DN 3 layers	0.836	0.791	0.850
DN 4 layers	0.868	0.797	0.872
DN 5 layers	0.880	0.800	0.885
DN 6 layers	0.888	0.799	0.893

Depth

Supplementary Table 3: **Study of network size and depth.** Comparison of shallow networks with different numbers of hidden units (single hidden layer), and deep networks with varying hidden layers in terms of the Area Under the ROC Curve (AUC) for the HIGGS benchmark. The deep networks have 300 units in each hidden layer.

	AUC		
Technique	Low-level	High-level	Complete
NN 300-hidden	0.733	0.777	0.816
NN 1000-hidden	0.788	0.783	0.841
NN 2000-hidden	0.787	0.788	0.842
NN 10000-hidden	0.790	0.789	0.841
DN 3 layers	0.836	0.791	0.850
DN 4 layers	0.868	0.797	0.872
DN 5 layers	0.880	0.800	0.885
DN 6 layers	0.888	0.799	0.893

Diminishing gains after 4-5 layers

Depth

Supplementary Table 3: **Study of network size and depth.** Comparison of shallow networks with different numbers of hidden units (single hidden layer), and deep networks with varying hidden layers in terms of the Area Under the ROC Curve (AUC) for the HIGGS benchmark. The deep networks have 300 units in each hidden layer.

	AUC		
Technique	Low-level	High-level	Complete
NN 300-hidden	0.733	0.777	0.816
NN 1000-hidden	0.788	0.783	0.841
NN 2000-hidden	0.787	0.788	0.842
NN 10000-hidden	0.790	0.789	0.841
DN 3 layers	0.836	0.791	0.850
DN 4 layers	0.868	0.797	0.872
DN 5 layers	0.880	0.800	0.885
DN 6 layers	0.888	0.799	0.893

Deep and thin > shallow and wide

Can we trust it?

Black-box summary of N-dim space into 1-D

- no physics intuition available to critique
- possible reliance on poorly modeled corners
- impossible to by-hand validate N-dim corr.

Can we trust it?

Black-box summary of N-dim space into 1-D

- no physics intuition available to critique
- possible reliance on poorly modeled corners
- impossible to by-hand validate N-dim corr.

Problem is the same for shallow and deep nets

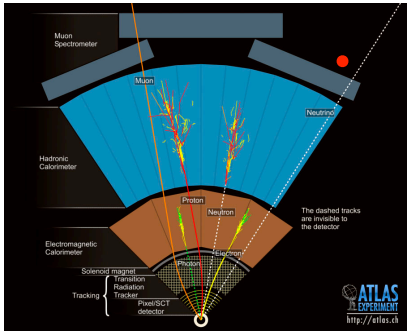
- only difference is non-linearity of function

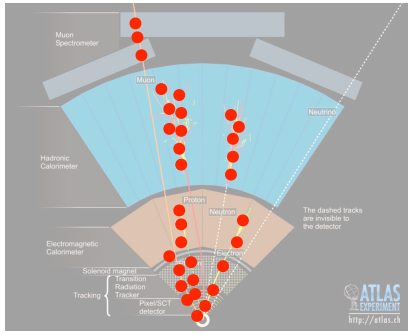
(1) validate inputs in 1D

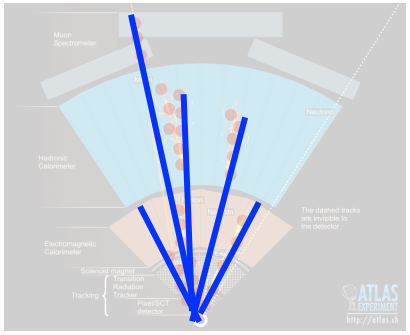
(2) validate outputs in data with bg control regions

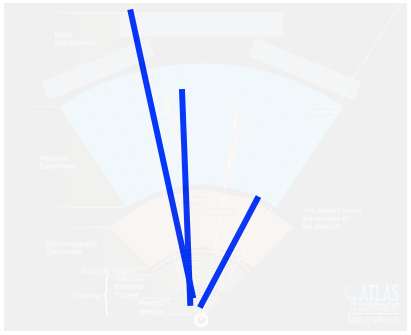
What is possible?

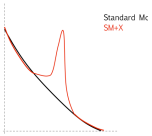
Raw	Sparsified	Reco	Select	Ana
1e7	1e3	100	50	1





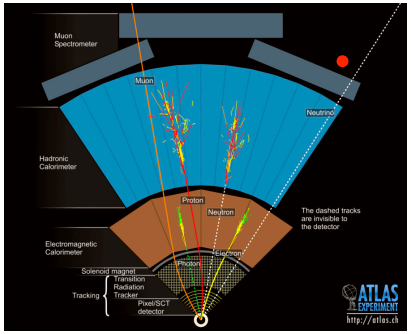


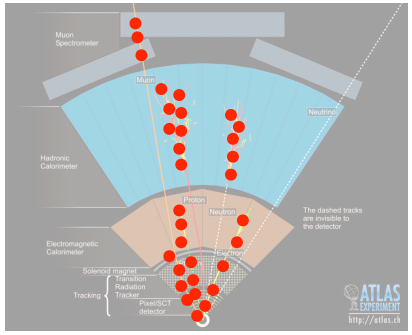


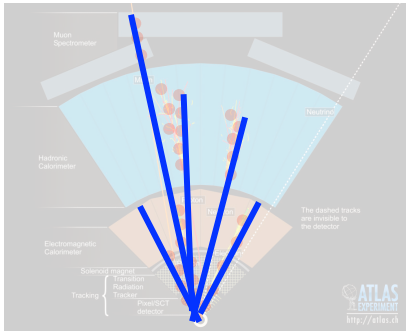


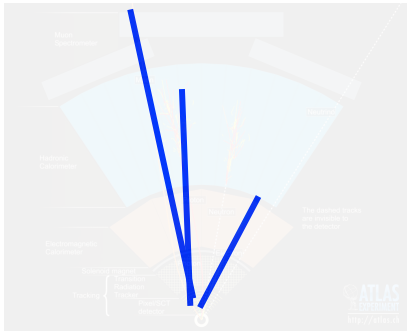
What is possible?

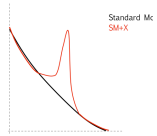
Raw	Sparsified	Reco	Select	Ana
1e7	1e3	100	50	1







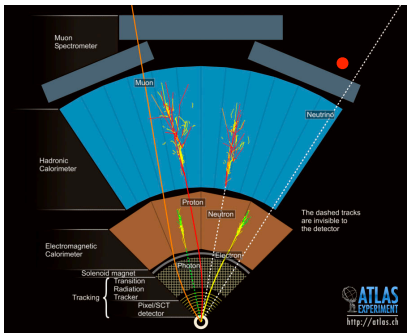


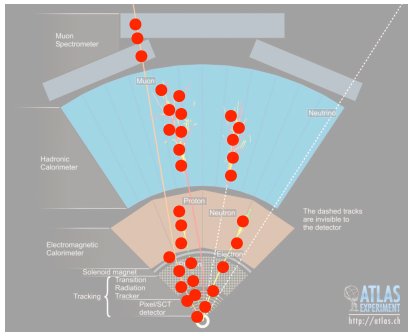


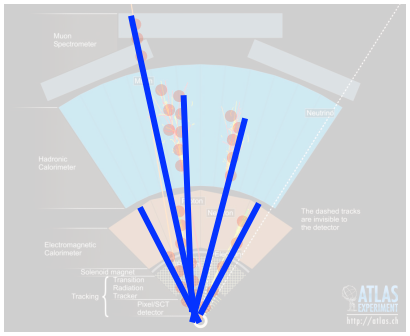
Skip more steps with ML?

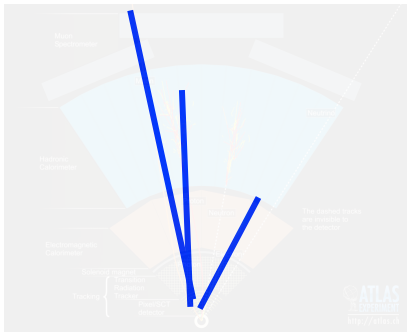
Or this?

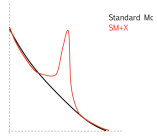
Raw	Sparsified	Reco	Select	Ana
1e7	1e3	100	50	1

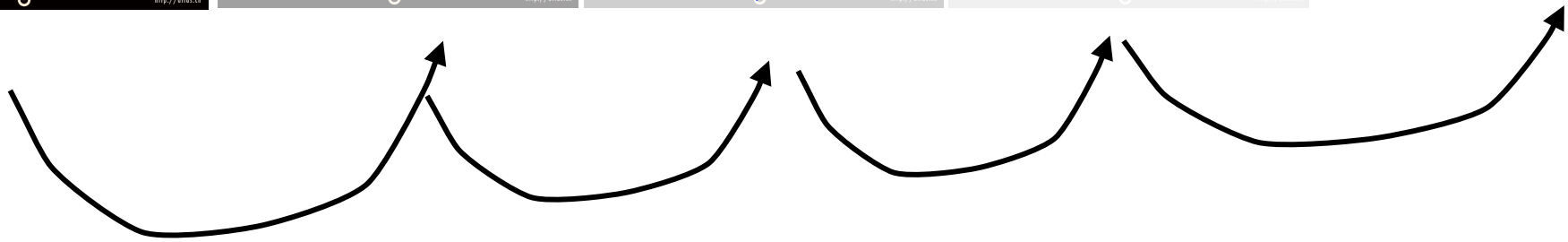












Improve each step with ML?

Outline

Defining the problem

Dimensionality reduction

Current approaches

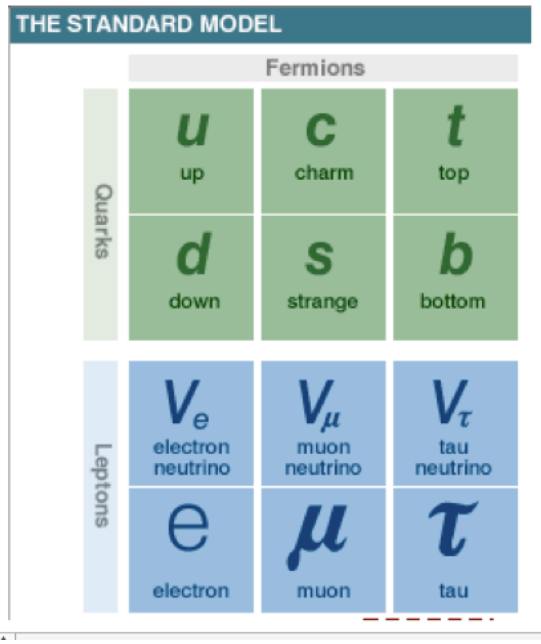
Deep(er) networks for low(er)-level data

Dimensionality increase

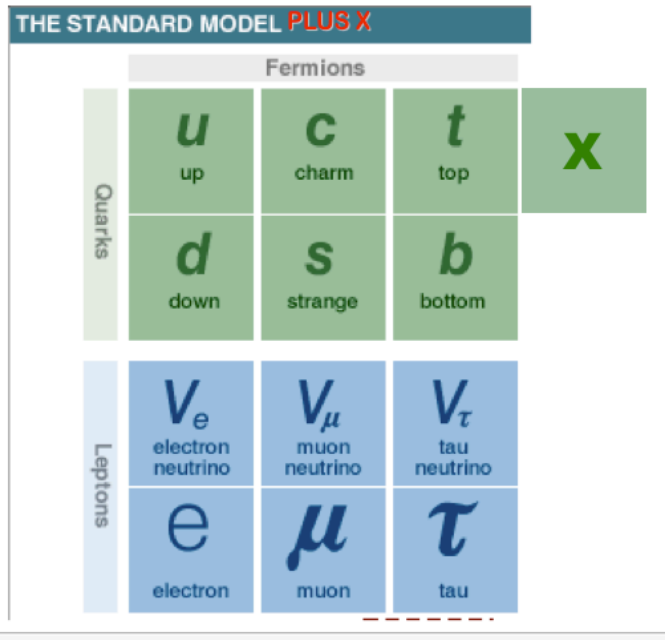
Parameterized networks for sets of problems

Larger Scope

1.

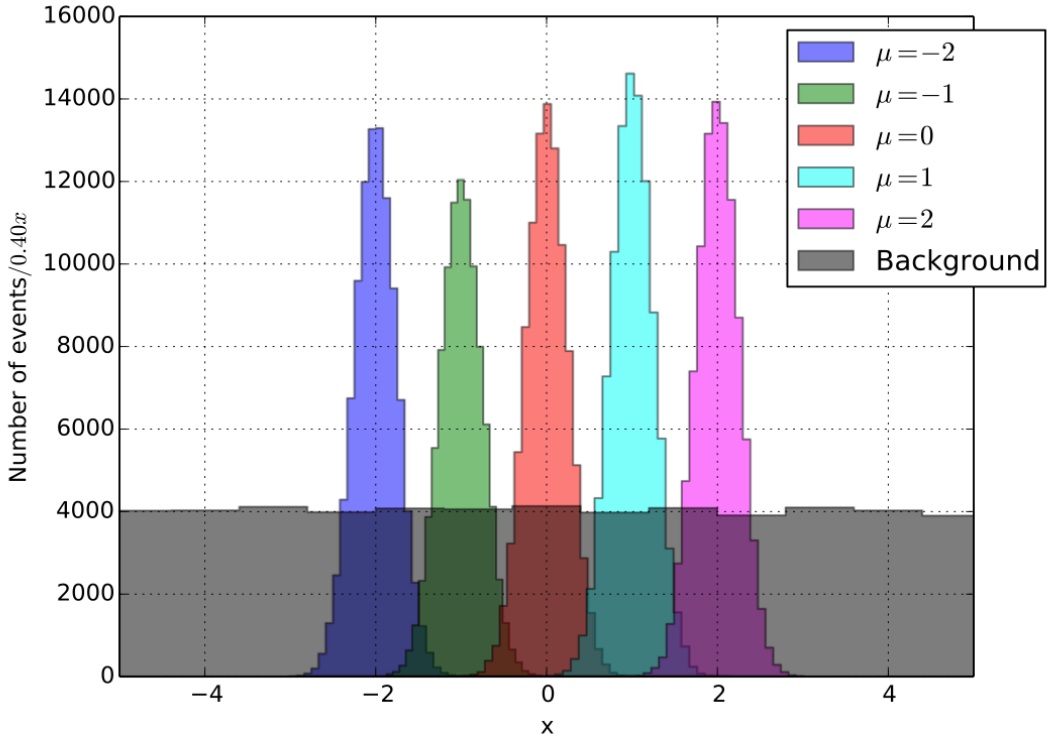


2.



What if you don't know the mass of X?

Related problems



You might have a set of closely related problems

Current approaches

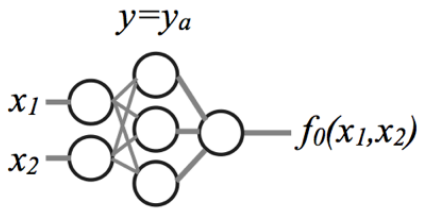
1. Develop unique solution at each point
   - Pros: optimality
   - Cons: cannot interpolate

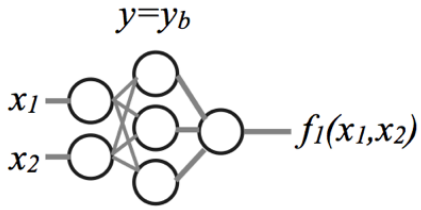
2. Develop single solution for all points
   - Pros: can interpolate
   - Cons: not optimal anywhere

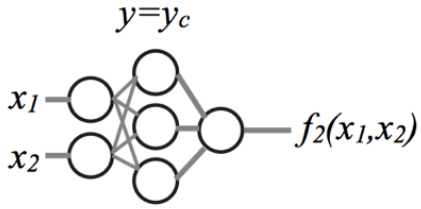
3. Develop single solution at one point, use everywhere
   - Pros: can interpolate, some optimality
   - Cons: not optimal everywhere

New approach

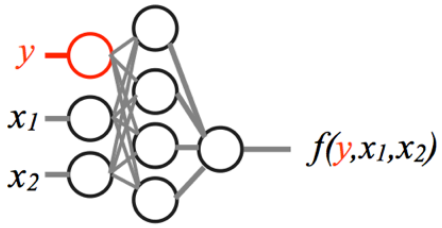
Replace **set** of solutions....



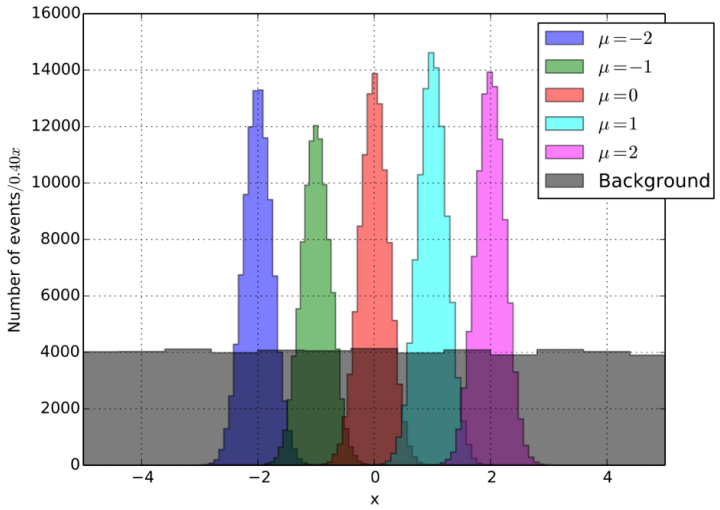


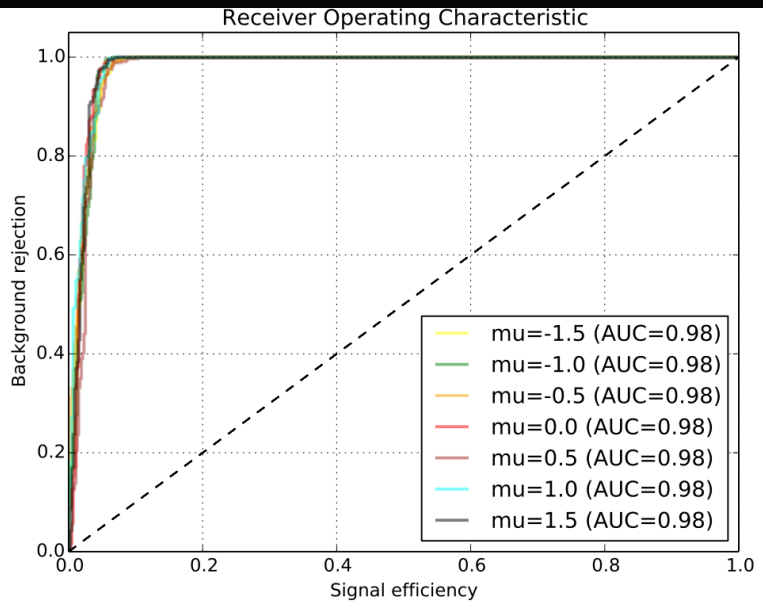


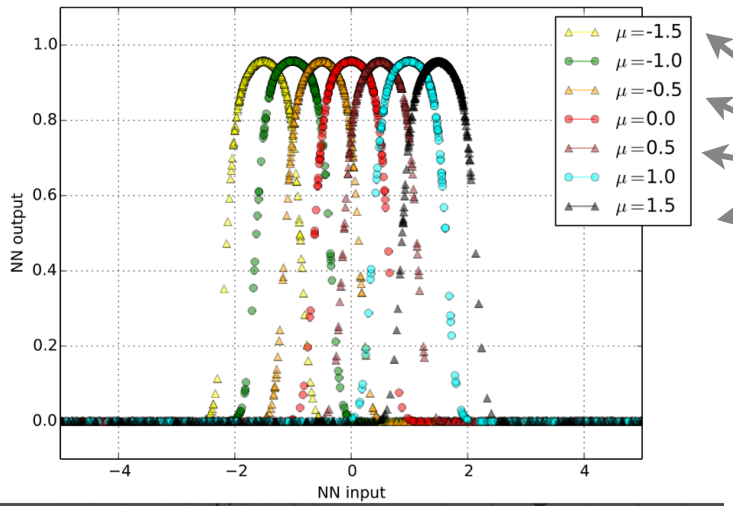
.....with single **parameterized** solution



Toy example 1



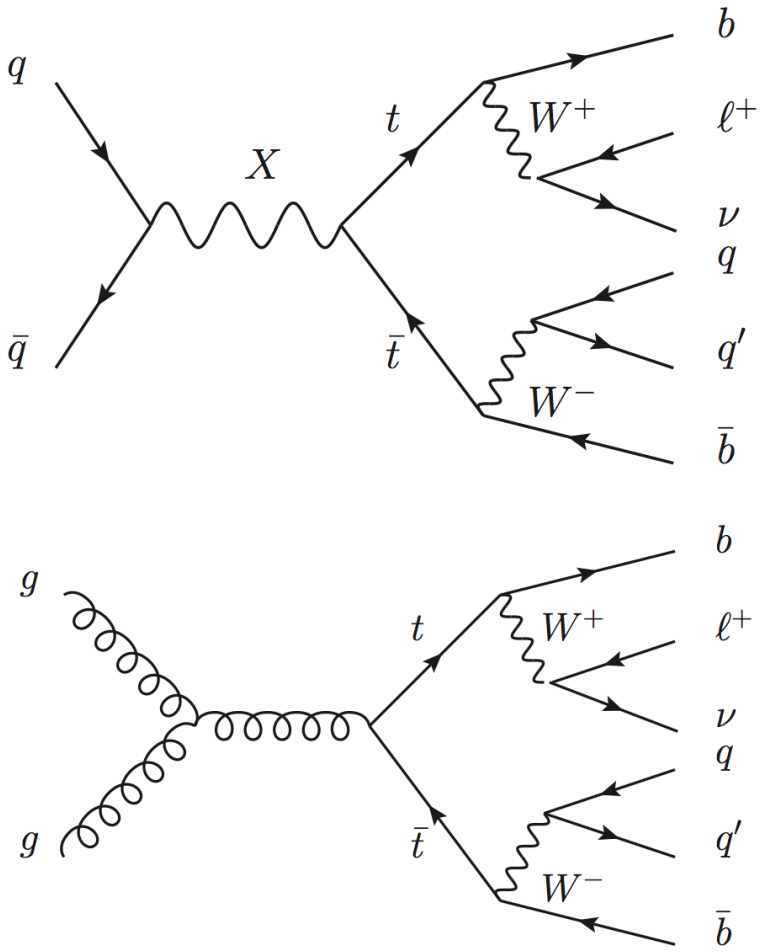


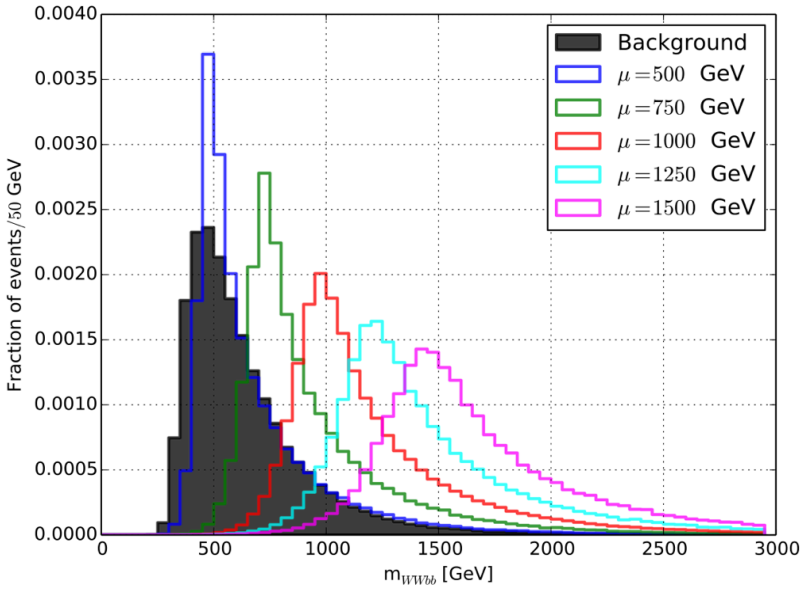


NN generalized solution.

Good performance at values of parameter with no training!

Half-toy example



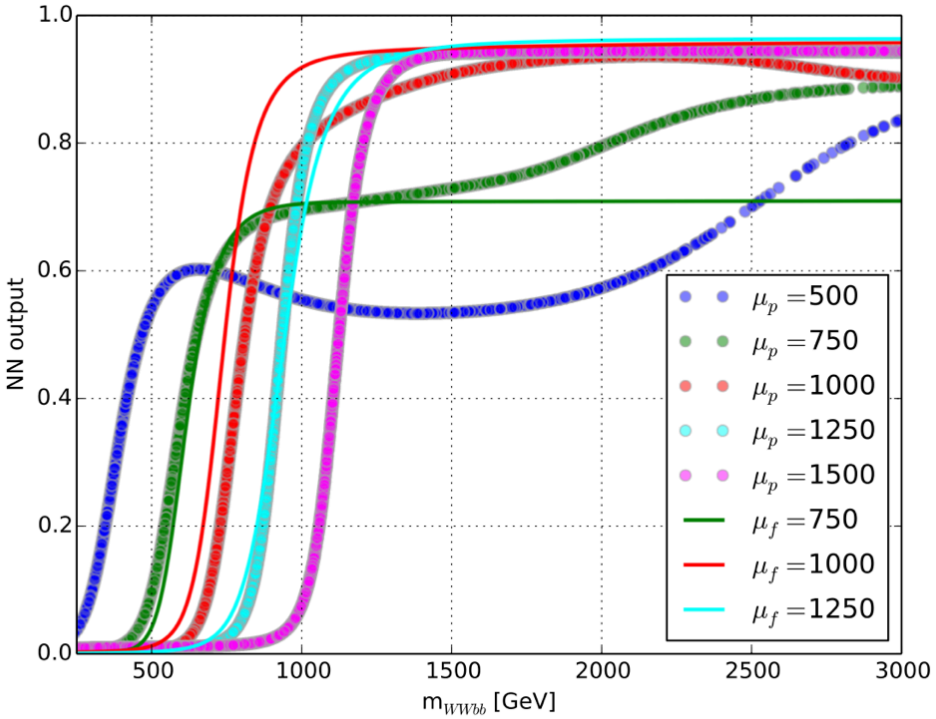


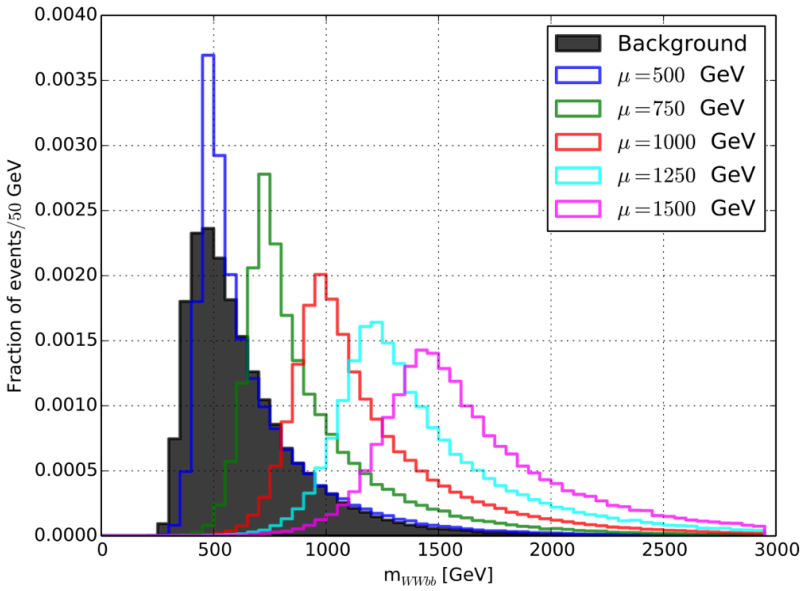
Half-toy example

Compare NN response

Single network trained at mu

Param network not trained at mu



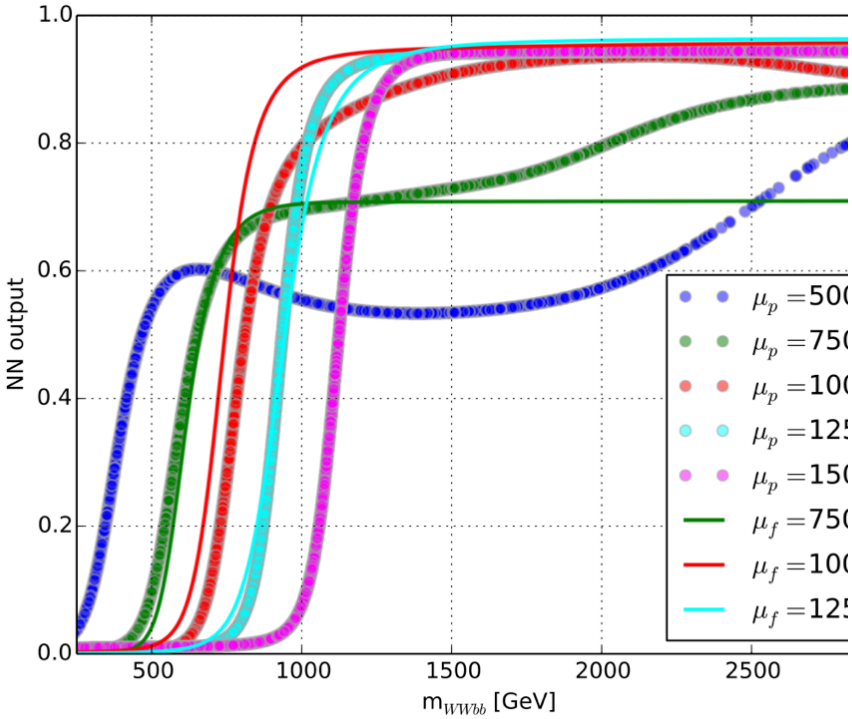


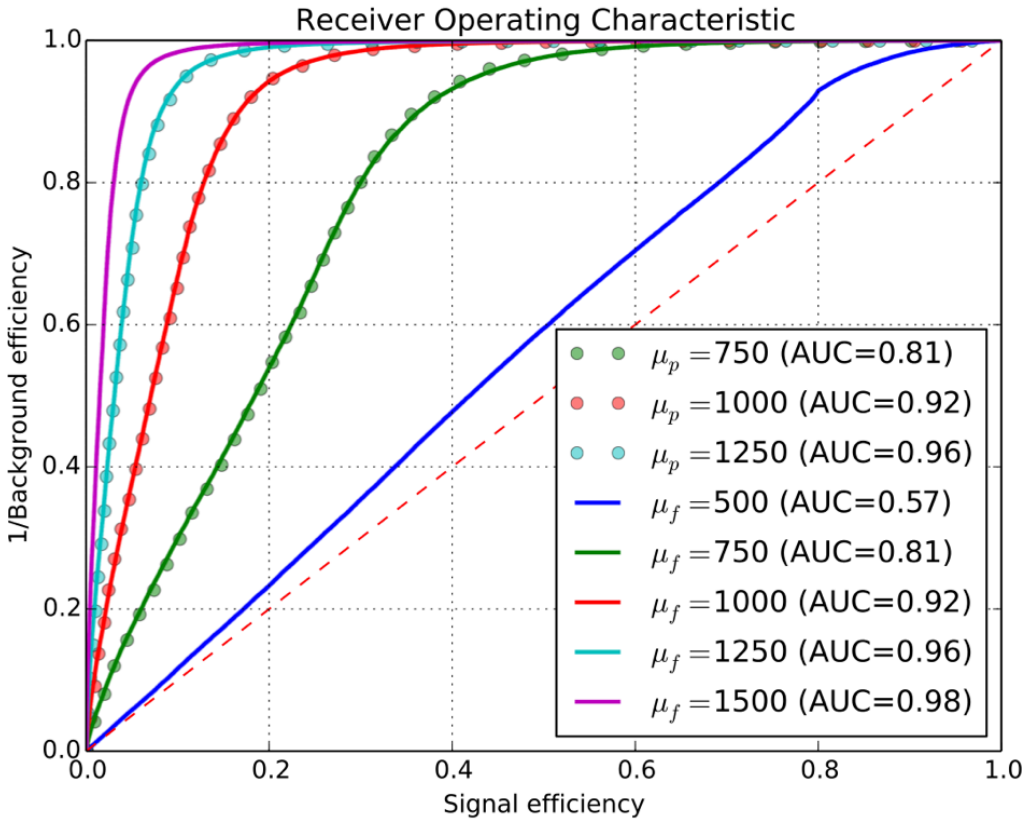
Half-toy example

Compare NN discrimination

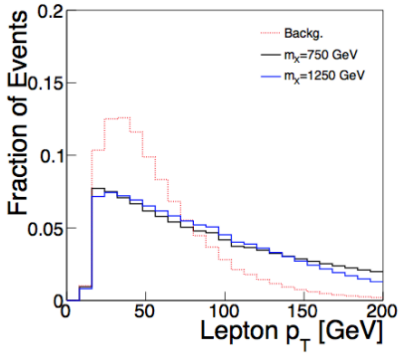
Single network trained at mu

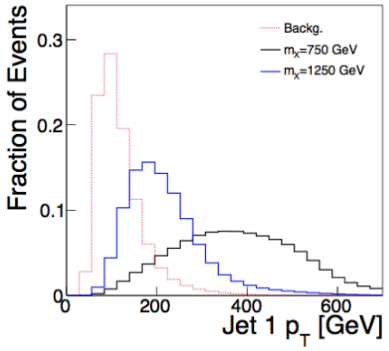
Param network not trained at mu

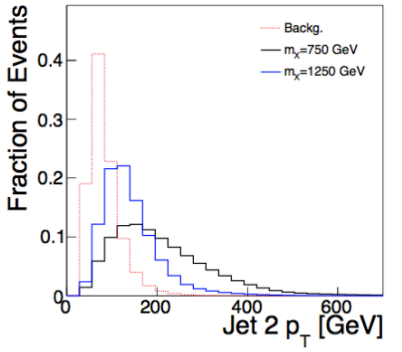


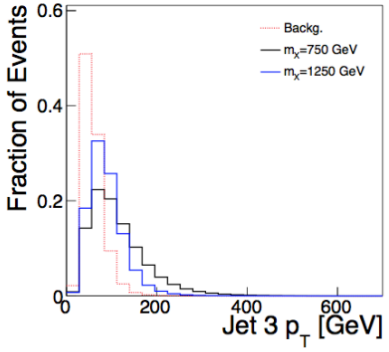


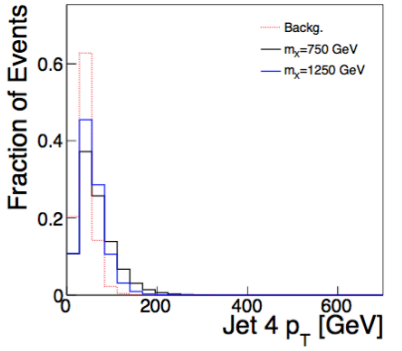
Full example

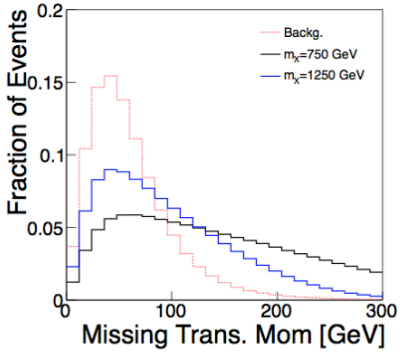




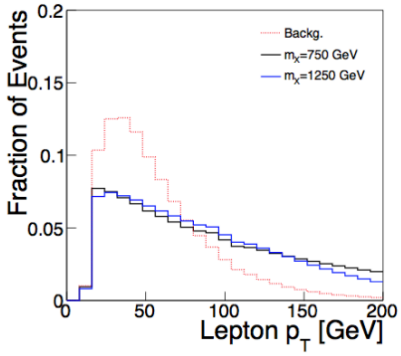


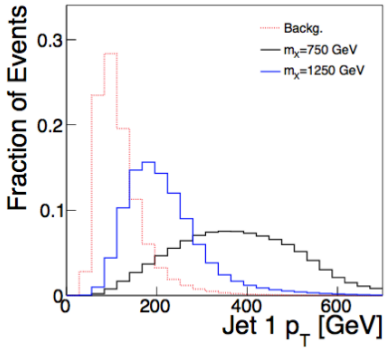


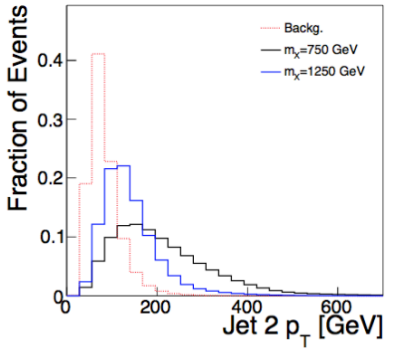


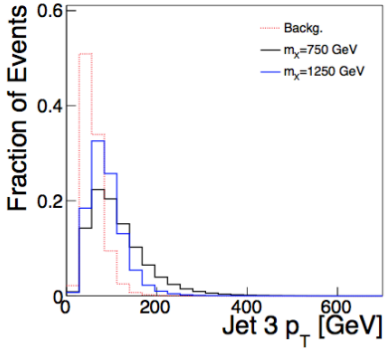


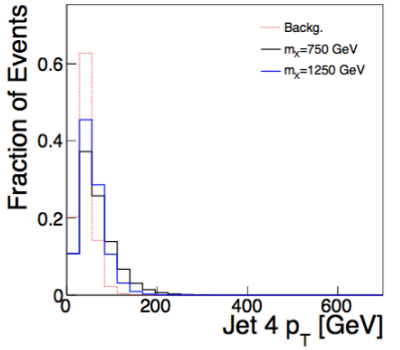
Full example

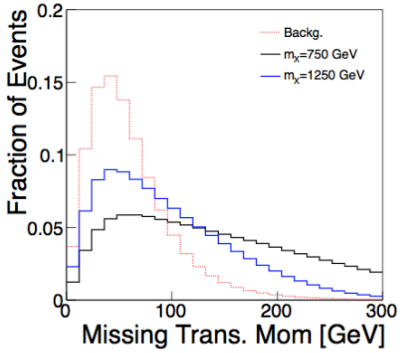


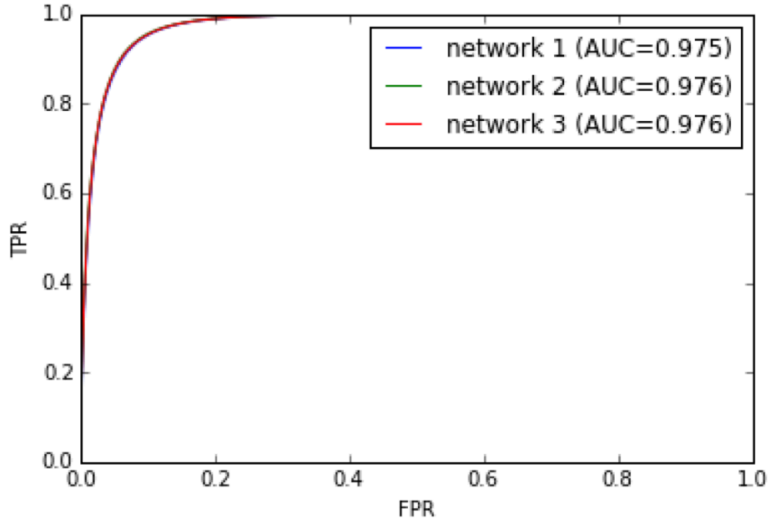












N1: param network
trained at m=500,750,1250,1500

N2: fixed network
trained at m=1000

N3: param network
trained at m=500,750,1000,1250,1500

Discussion

We have an intractable high-dimensional p problem of calculating:

$$\frac{L_{SM+X}}{L_{SM}} = \frac{P(\text{data} \mid SM+X)}{P(\text{data} \mid SM)}$$

Human tricks to reduce dimensionality are non-optimal.

Gains possible at many levels by replacing brains with AI.