

Physicists Learning From Machines Learning

Smart but Interpretable Neural Networks for Physics at the LHC

TAYLOR FAUCETT

FEBRUARY 1, 2021

Defining The Problem

We Have a Lot of Data

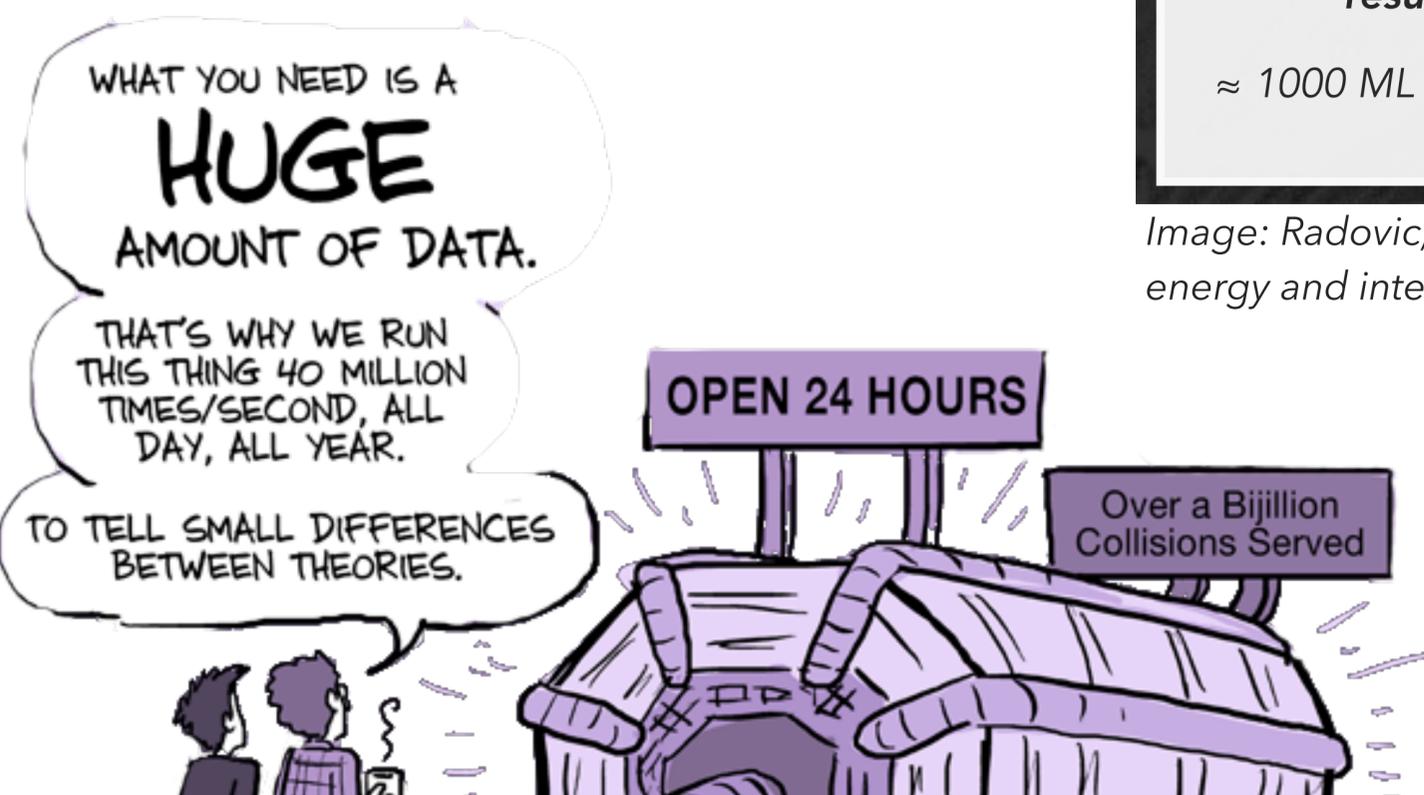
LHC Run 2 across 4 experiments generates, on average, 25 GB/s of data.

- High Energy Uses ML in
 - Object classification
 - Triggering
 - Event reconstruction
 - Event selection
 - Data pre-processing
 - Measurement uncertainties
 - Detector Design

Better machine learning means better results, better simulations.

≈ 1000 ML papers published to ArXiv just this month

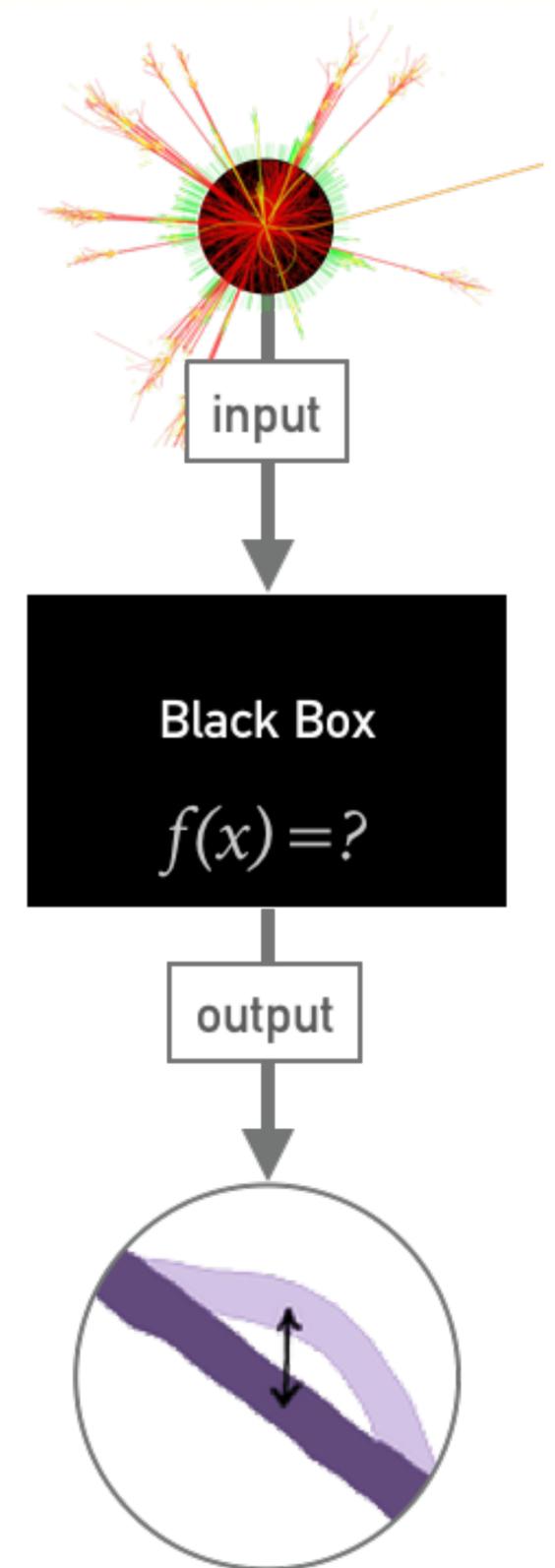
Image: Radovic, A., Williams, M., Rousseau, D., Nature, M. K., 2018. (n.d.). Machine learning at the energy and intensity frontiers of particle physics



The Curse of ML Performance! The Black Box Problem

- Modern ML tools tend to be Black Box solutions.
 - Neural Networks (NN)
 - Deep Neural Networks (DNN)
 - Boosted Decision/Regression Trees (BDT/BRT)
 - Support Vector Machines (SVM)
 - Generative Adversarial Networks (GAN)
 - Autoencoders
- Black Box methods can't tell you what they've learned.
- A Black Box is a "mysterious function" that maps data to predictions.

So what? Why is that a problem?

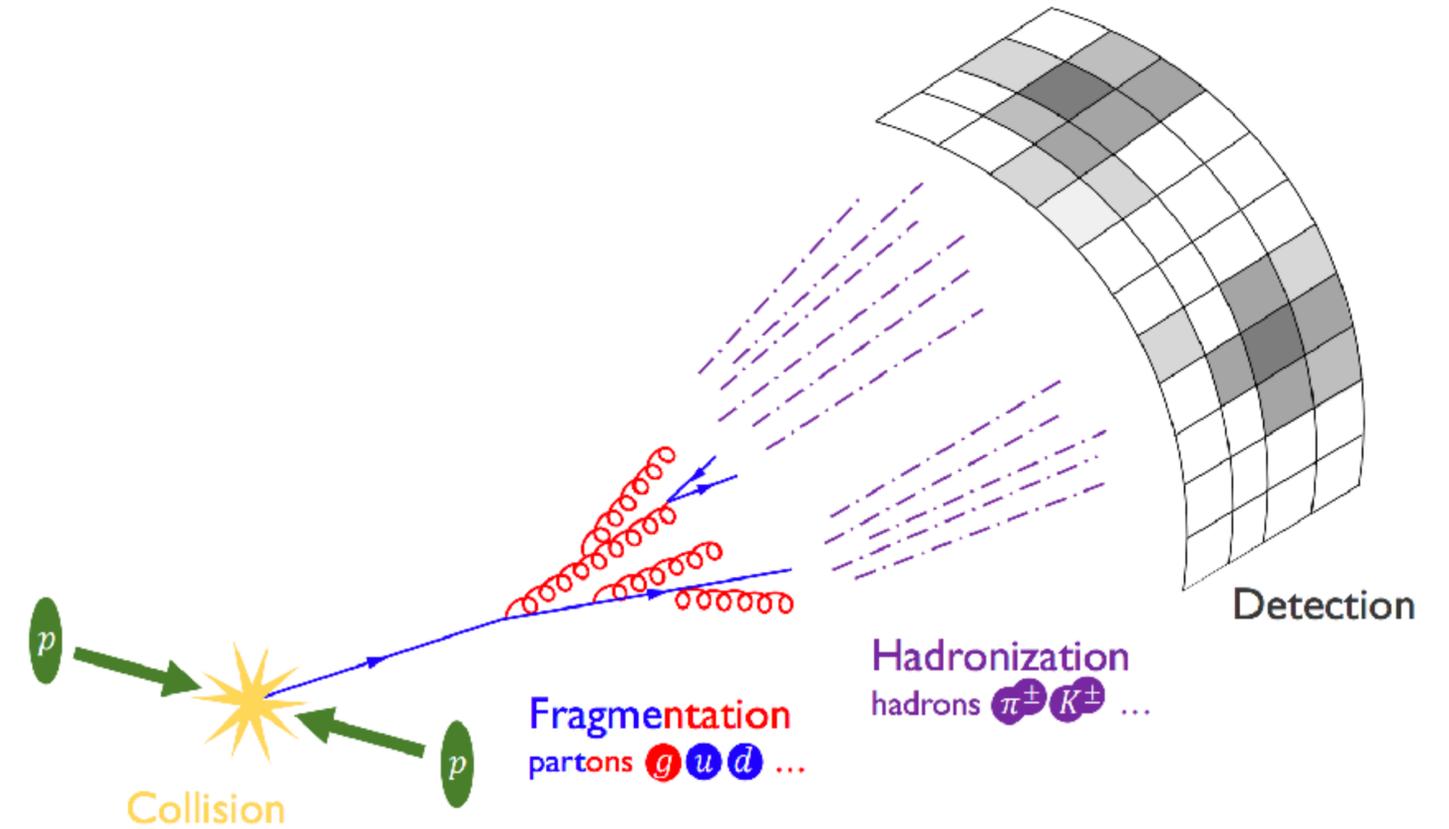


Real World Example: Jet Classification With Black Box Learning

Image: <https://www.ericmetodiev.com/post/jetformation/>

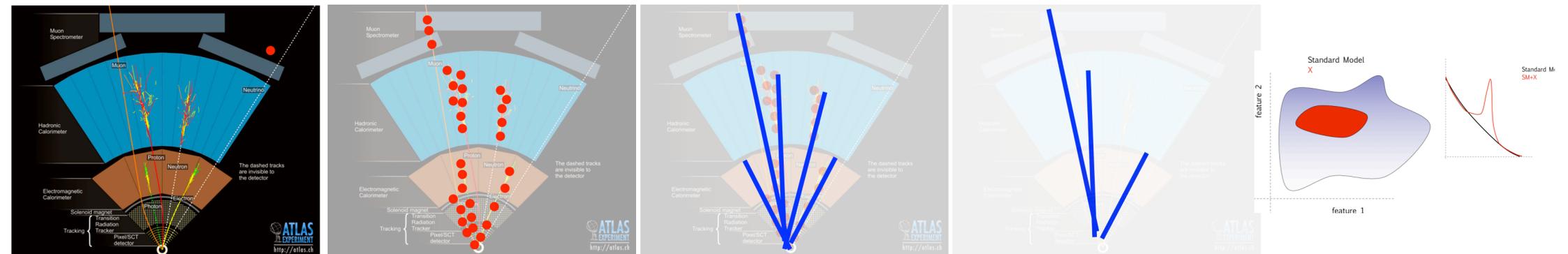
Consider the task of identifying jets

- Jets: collimated groups of stable hadrons from hadronizing quarks/gluons
- Jets are detected as energy depositions in a calorimeter.



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

Detector Measurements can exist in many different states of "dimensionality reduction"



Comparing ML With Low Level and High Level Jet Data

2 popular versions of jet data:

1. Low Level (high-dimensional) : Jet Images
2. High Level (low-dimensional): Jet Substructure (JSS)

**Which works better ?
HL or LL ?
Blind NN vs Physicists!**

High Level (HL) - Jet Substructure

Physics motivated variables which encode information into simple 1-D variables

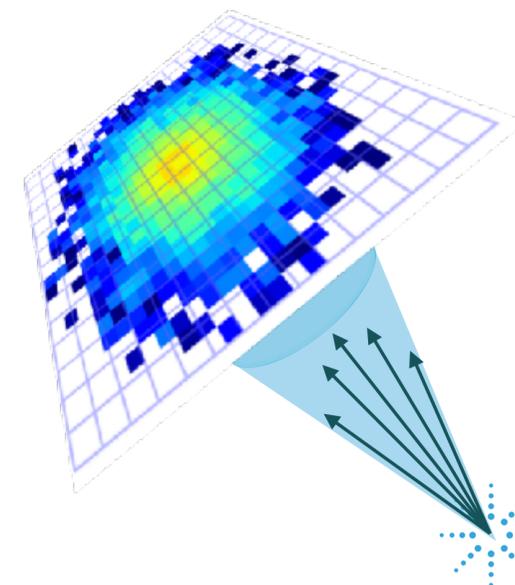
Mass
$$M_{jet} = \frac{1}{2} \sum_a^N \sum_b^N z_a z_b \left(\frac{2p_a^\mu p_b^\mu}{E_a E_b} \right) \quad z_i = \frac{P_{T,i}}{\sum_i P_{T,i}}$$

N-subjetiness
$$\tau_N = \sum_k p_T \min(\theta_{ab}, \theta_{bc}, \dots, \theta_{N,k})$$

Sub-structure examples

Low Level (LL) - Jet Images

Transverse energy (E_T) in an (η, ϕ) calorimeter grid. Treat it as an "image" and learn with a Convolutional Neural Network (CNN)

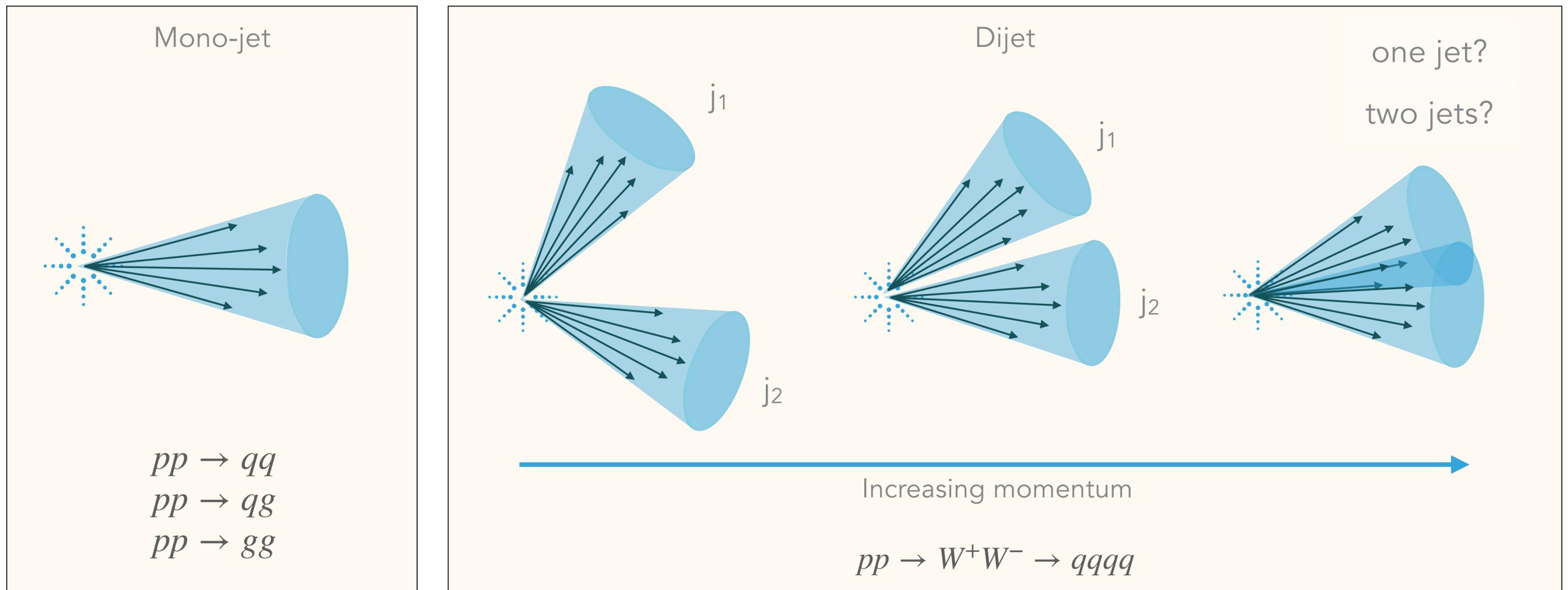


- E_T = Transverse Energy
- Position (η, ϕ)
- $\eta = -\ln(\tan(\theta/2))$

Test Case for Jet Classification: Binary Classifier of Mono-Jet vs Boosted Di-Jet

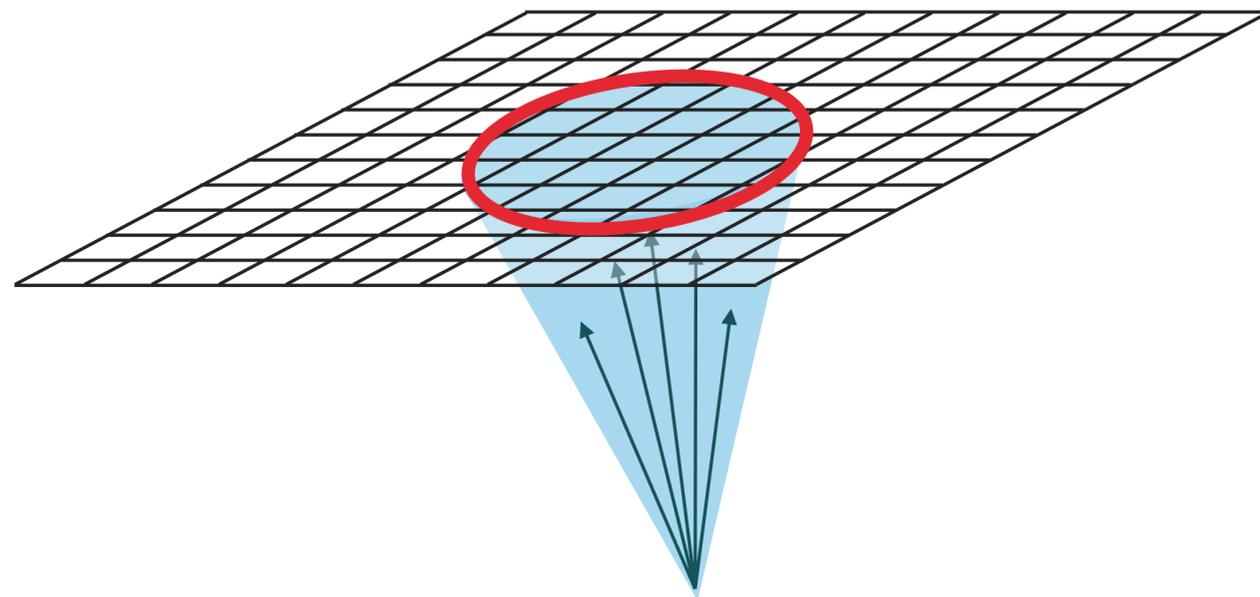
Boosted W bosons ($W \rightarrow qq'$) create highly collimated di-jets.

Can we teach a CNN to identify boosted di-jets from QCD background mono-jets (qq , qg , gg)?

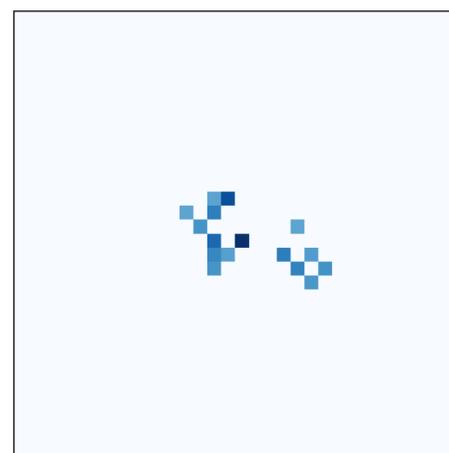
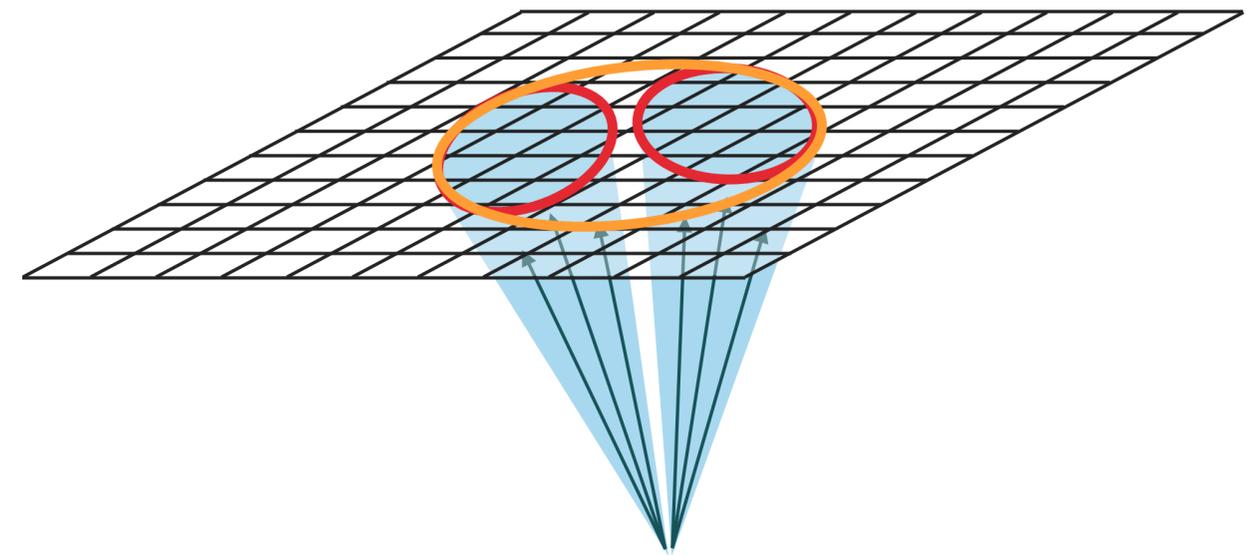


Binary Classifier: Di-Jet vs Mono-Jet

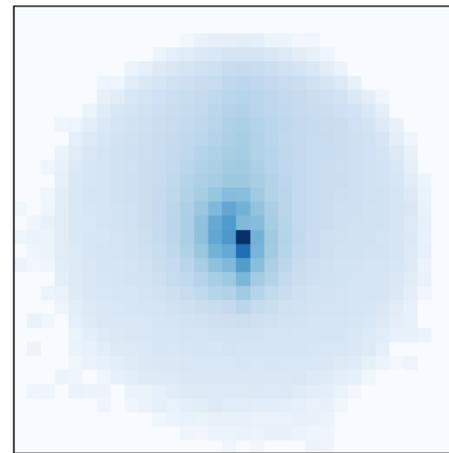
QCD Jet (q, g)



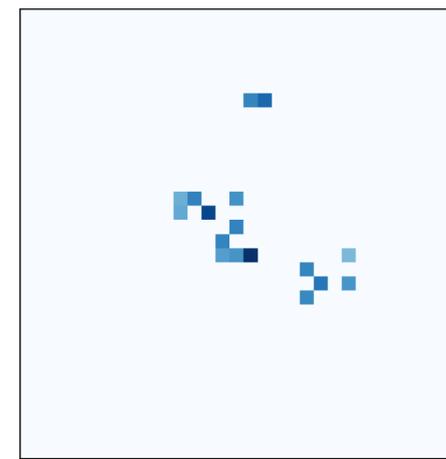
W jet



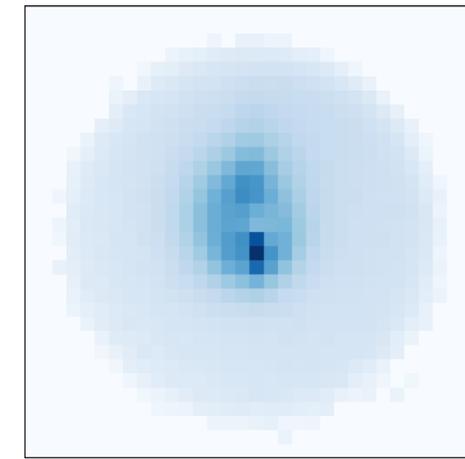
1 Event



Average of all events



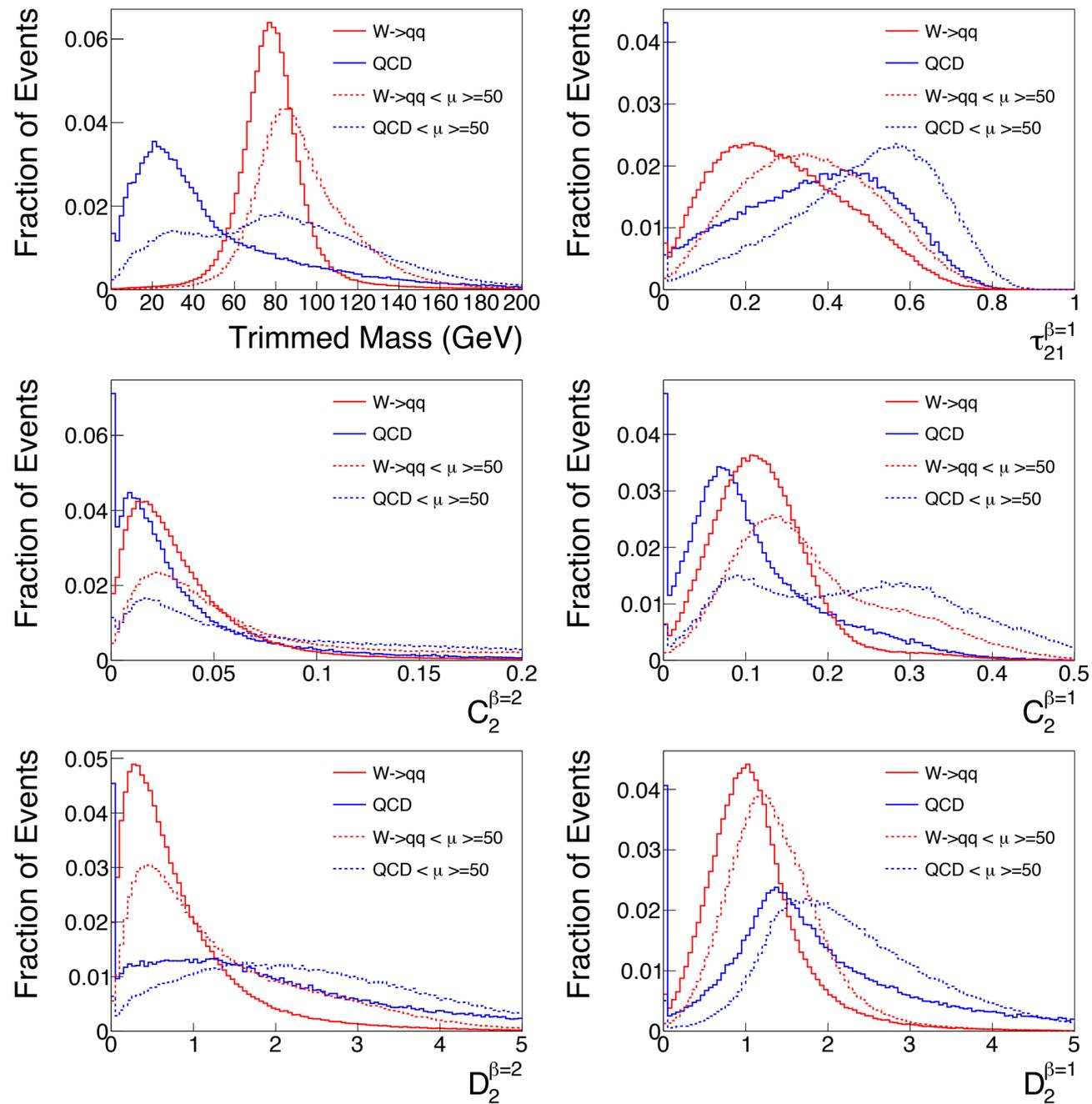
1 Event



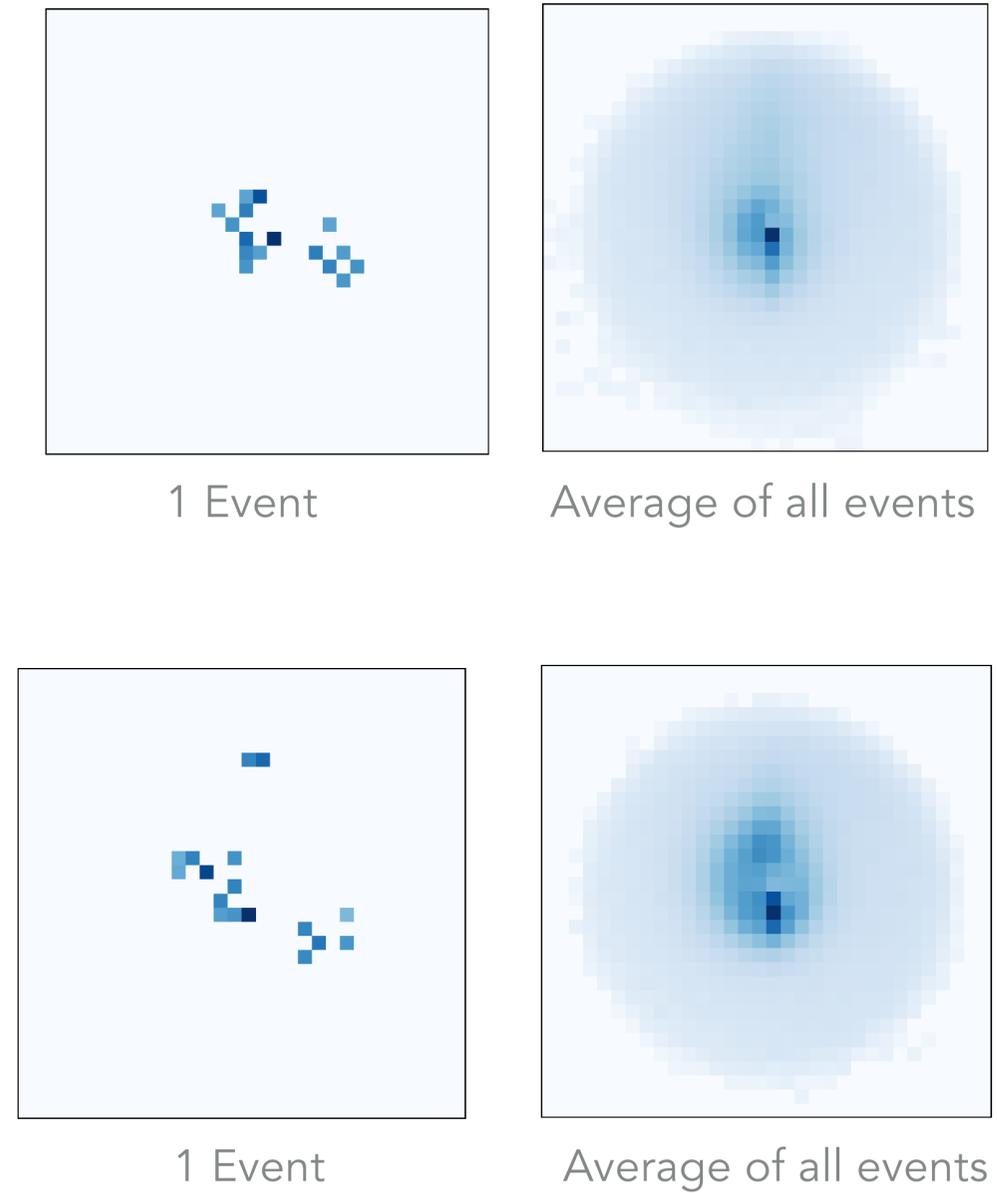
Average of all events

HL Vs LL

6 HL Variables



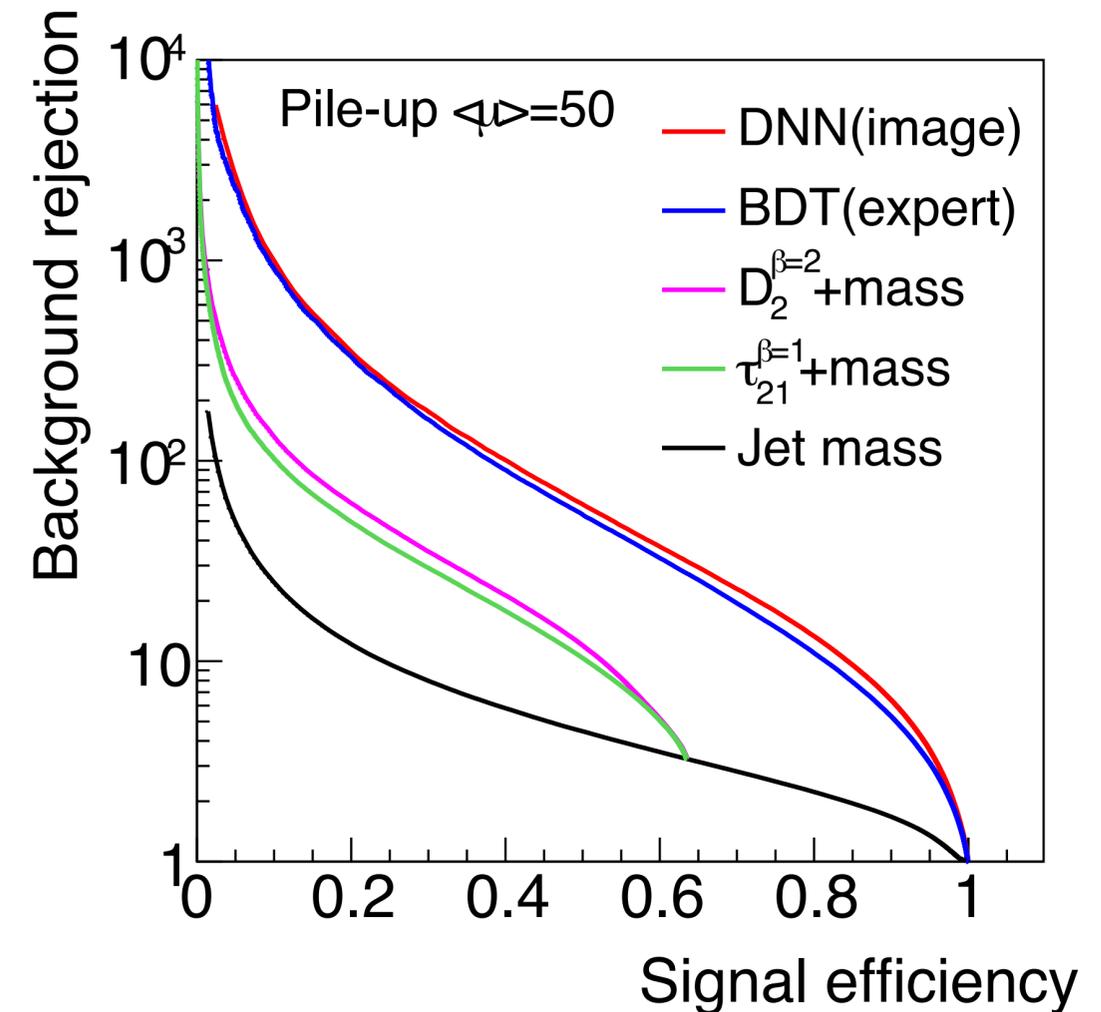
LL Jet Images



A Perfect Test Case

- Baldi et al. find a CNN on jet images performs better than Jet Substructure
 - **Jet Images (red line):** AUC = $95.30\% \pm 0.02\%$
 - **JSS (blue line):** AUC = $95.00\% \pm 0.02\%$
- Where is that extra information coming from?
- Why don't our standard Jet Substructure observables contain this information?
- Is it real physics that we don't know about yet?

We've used a black box, so now what? _(ツ)_/



Baldi, P., Bauer, K., Eng, C., Sadowski, P., & Whiteson, D. (2016, March 30). Jet Substructure Classification in High-Energy Physics with Deep Neural Networks. *arXiv.org*. <http://doi.org/10.1103/PhysRevD.93.094034>

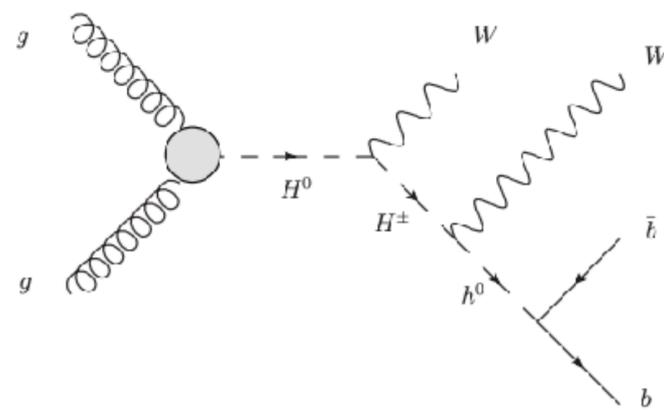
That's Not the Only Problem

Black Box models trained on LL data:

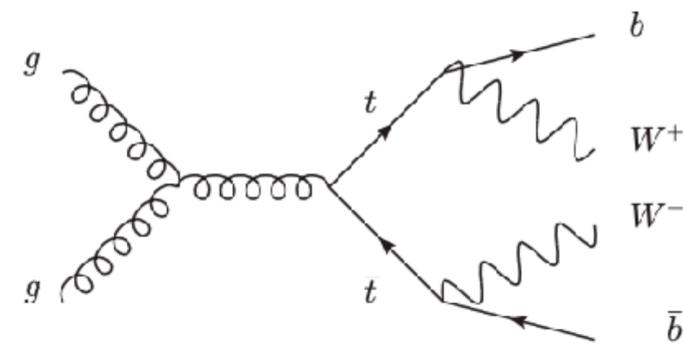
- Can't be validated as using real/physical information vs "improvements" through some quirks of processing, sampling, architecture. HL observables can be individually inspected.
- Can't measure systematic uncertainties. HL observables can be individually studied and calibrated.
- Are storage, memory and training intensive.
- Can't improve or contribute new insights into the physics of the problem being studied.

More Examples of an Information Gap in HEP (HL Vs LL)

Higgs Classification



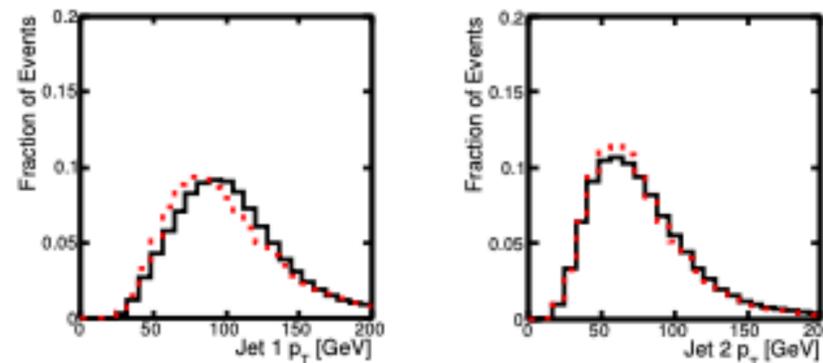
(a)



(b)

Low-Level

Jet pT of constituents, MET & Jet b-tags

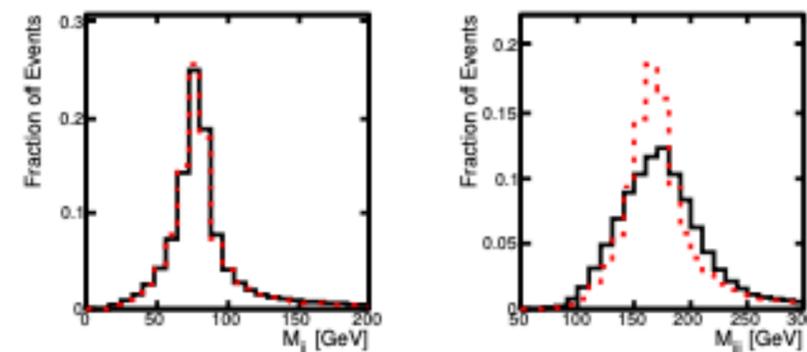


(a)

(b)

High-Level

Invariant mass: $M(WWbb)$, $M(Wbb)$, etc



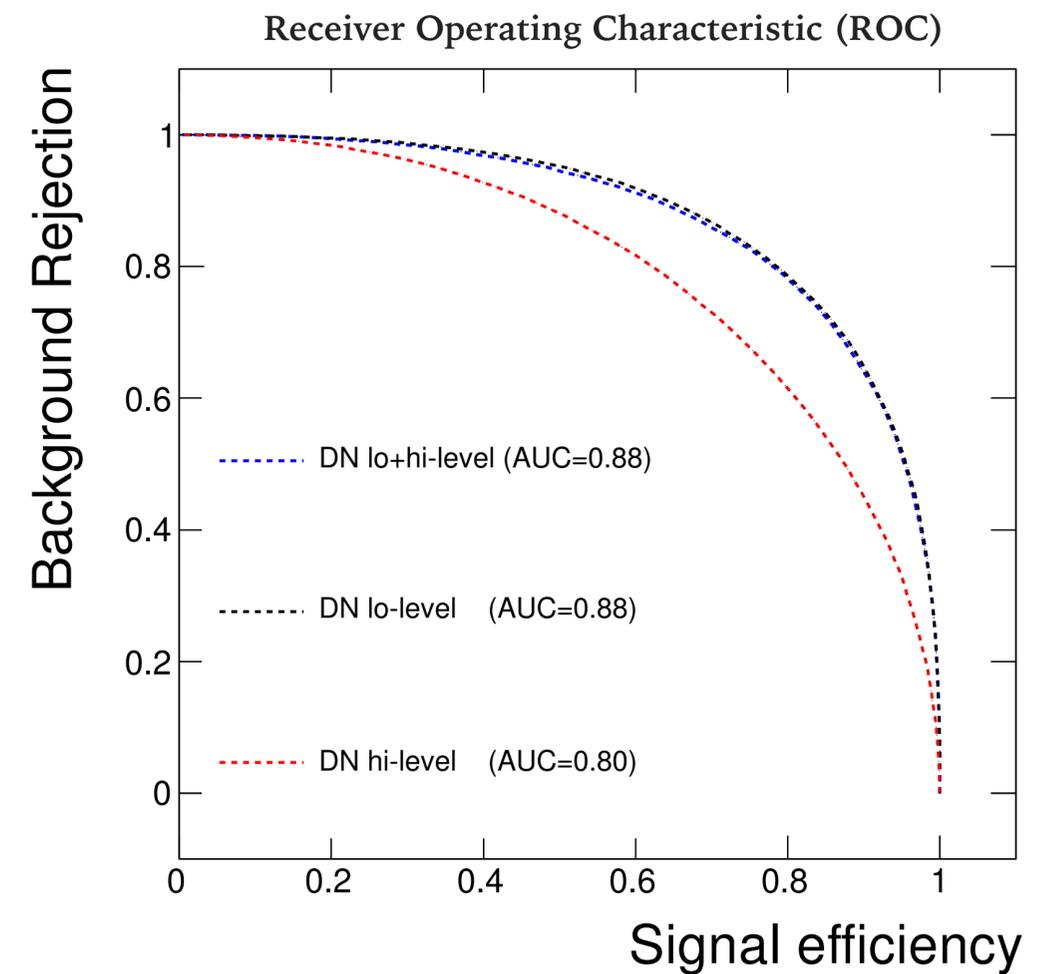
(a)

(b)

Performance

Low-Level: AUC=0.88

High-Level: AUC=0.80



<https://arxiv.org/abs/1402.4735> - Searching for Exotic Particles in High-Energy Physics with Deep Learning

More Examples of an Information Gap in HEP (HL Vs LL)

Electron Identification

$pp \rightarrow Z' \rightarrow e^+e^-$ vs QCD Jet Background

Low-Level

High-Level

Performance

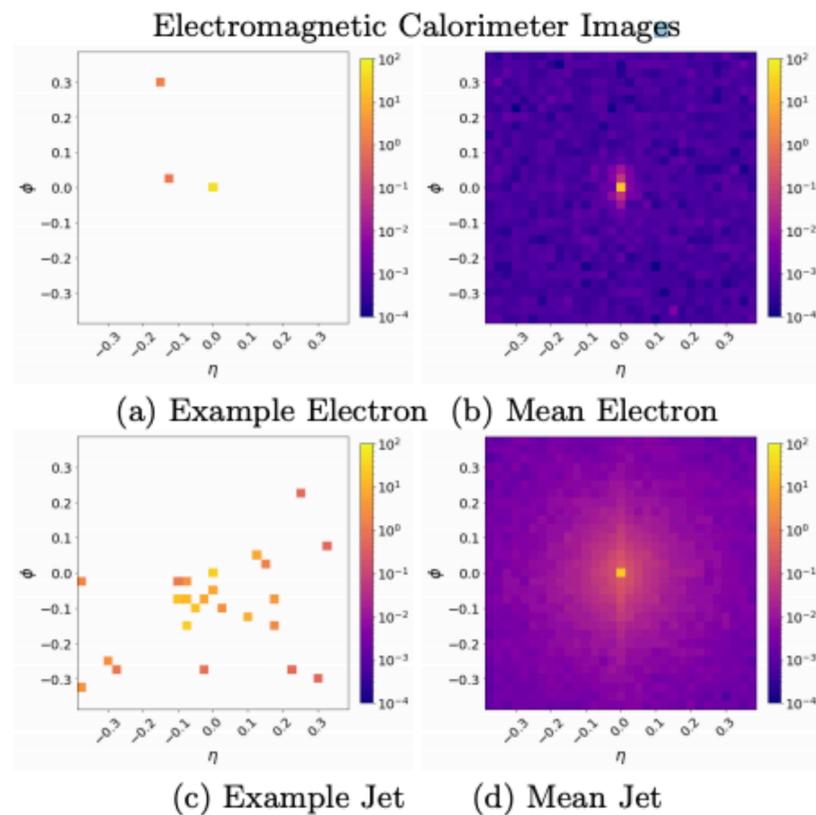


FIG. 2: Images in the electromagnetic calorimeter for signal electrons (top) and background jets (bottom). On the left are individual examples, on the right are mean images. See Fig. 3 for corresponding hadronic calorimeter images.

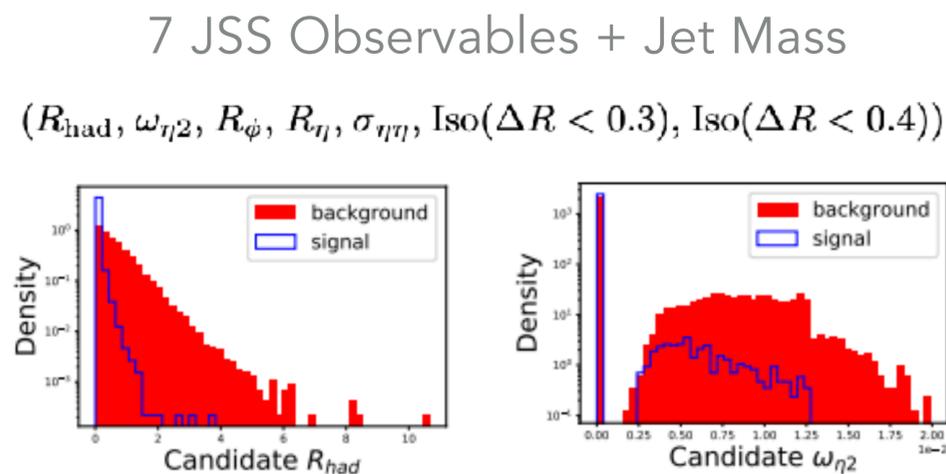
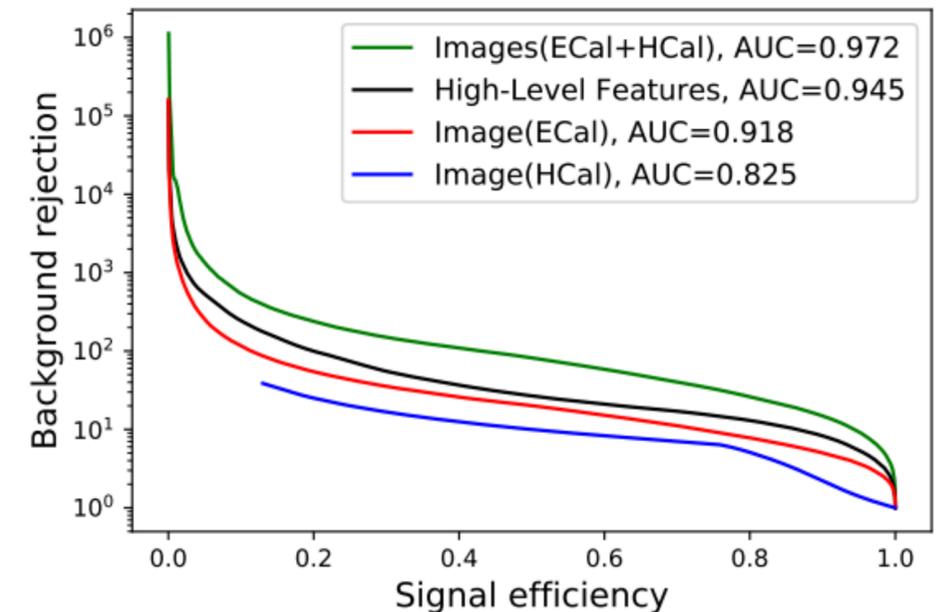


FIG. 4: Distribution of signal electron (red) and background jets (blue) for seven existing typically-used high-level features, as well as for mass.

Low-Level: $AUC=0.972 \pm 0.001$
 High-Level: $AUC=0.956 \pm 0.001$



<https://arxiv.org/abs/2011.01984> - Learning to Identify Electrons

Solving The Problem

Solving the Black Box Problem

We've established a problem with Black Box solutions. How do we fix it?

3 Approaches to dealing with the problem:

1. Only use interpretable ML.

- Not Ideal. Loss in performance AND you lose the opportunity to learn new physics.

2. Modify Black Box to be interpretable

- Disadvantages: Captures "most" information but not all.

- Complex to design & modify.

- Example: Microsoft Research - Intelligible Machine Learning Models for HealthCare

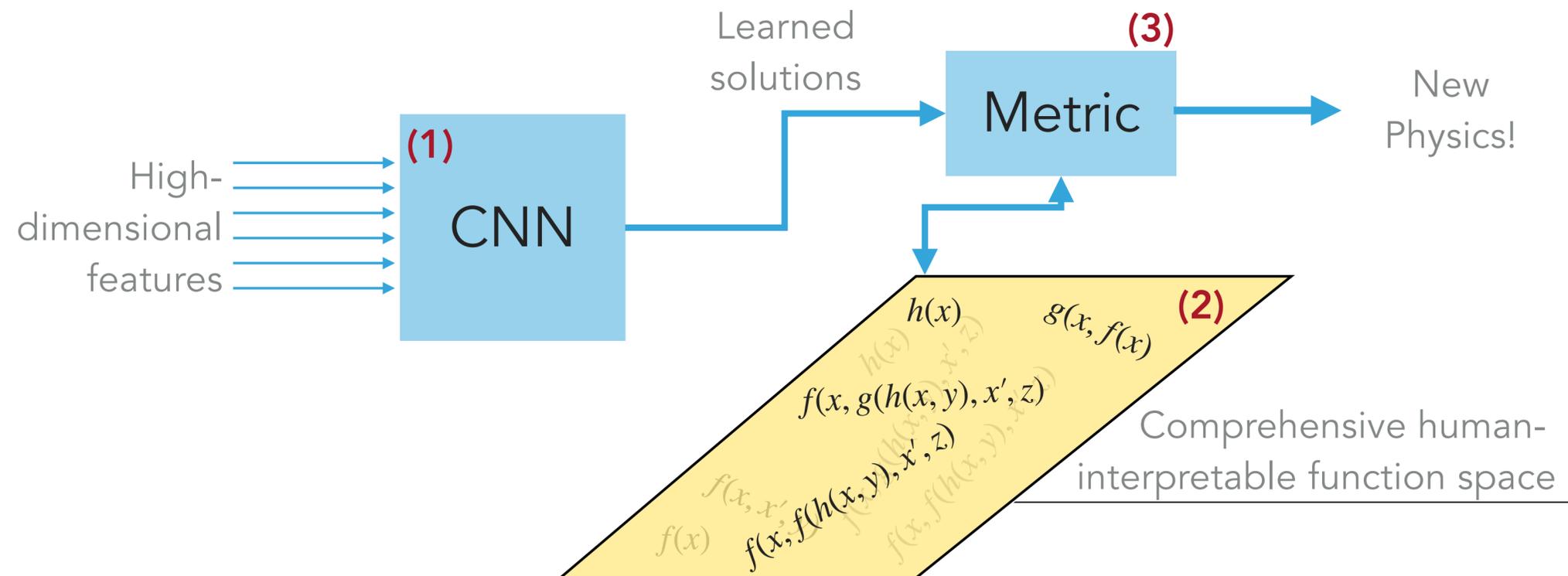
3. [Our Solution] -> Use the Black Box and map it's solution to a human readable space

- Take full advantage of your models performance

- Use any ML method you want. DNN, SVM, etc.

Mapping ML to a Human Readable Space

- What do we need to map learned information to a human readable space?
 - (1) A LL solution that performs better than the HL
 - ✓ We have this from our CNN on jet images.
 - (2) A "human readable" space of HL variables.
 - (3) A metric for mapping the LL solution into those HL variables.



Piece 2 - the Engineered Space of Human-Interpretable Variables

Energy Flow Polynomials (EFP):
Complete linear basis set for jet substructure

The set of EFPs is defined as all isomorphic graphs, with p_T and position (θ) as defined below

Graph components

Node/Vertex:		$= \sum_a^N z_a$
Edges:		$= \theta_{ab}$
Multiple Edges		$= (\theta_{ab})^2$

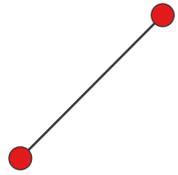
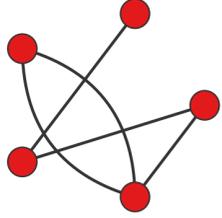
$z_a = p_{T,a}^\kappa$
 $\theta_{ab} = (\Delta\eta_{ab}^2 + \Delta\varphi_{ab}^2)^{\beta/2}$

For every set of graphs, we can also modify 2 parameters (κ, β)

$$z_i = \frac{p_{T,i}^\kappa}{\sum_i p_{T,i}}$$

$$\theta_{ij} = (\Delta y_{ij}^2 + \Delta_{ij}^2)^{\beta/2}$$

Examples

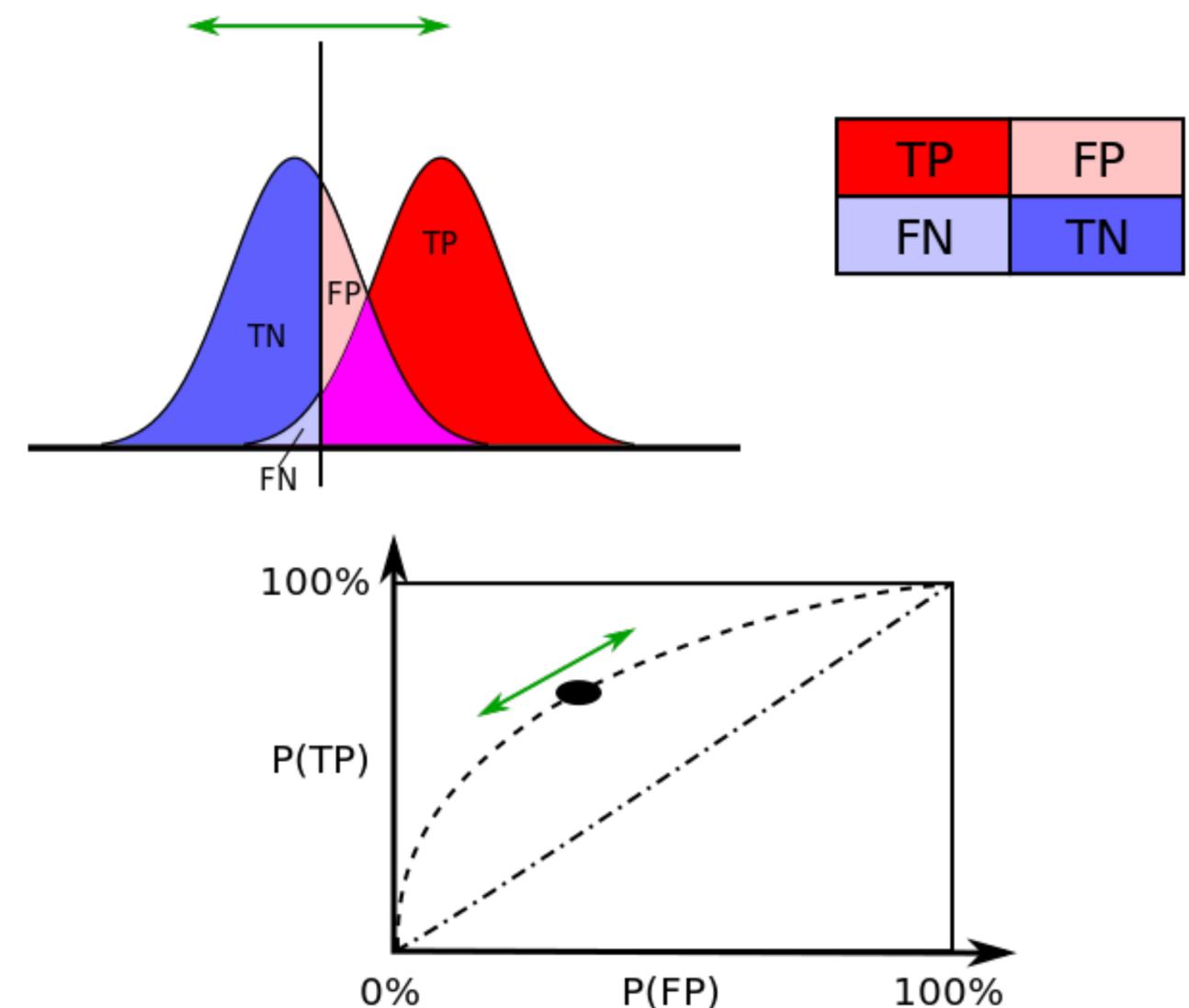
	$= \sum_a^N \sum_b^N z_a z_b \theta_{ab}$
	$= \sum_a^N \sum_b^N \sum_c^N \sum_d^N \sum_e^N z_a z_b z_c z_d z_e \theta_{ac}^2 \theta_{bd} \theta_{be} \theta_{cd}$

Komiske, P. T., Metodiev, E. M., & Thaler, J. (2017, December 19). Energy flow polynomials: A complete linear basis for jet substructure.

Piece 3 - ROC Is the Right Way To Approach It, but We Need To Modify It.

- We don't want to use ROC/AUC.
- ROC describes training performance, not decision making similarity.
- ROC compares True Positive Rates at different thresholds of False Positive Rates.
- But it's the right idea! We want:
 - For 2 neural networks, at different thresholds of NN output, what is the relative similarity of the decision surfaces of each?

ROC/AUC is analogous to our process, but
Replacing TP/FP comparisons with decision boundaries



Piece 3 - Average Decision Ordering (ADO)

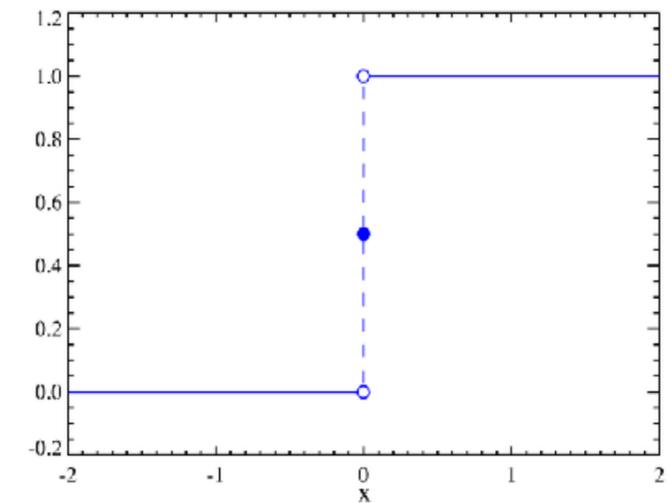
We want an equivalent to ROC for 2 discriminating functions $[f(x)$ and $g(x)]$. Classification decision of two functions at different thresholds.

Step 1: Decision Ordering

For points from signal (1) and background (0) (called x and x'), we compare how each NN (i.e. function) maps those in their final prediction relative to one another.

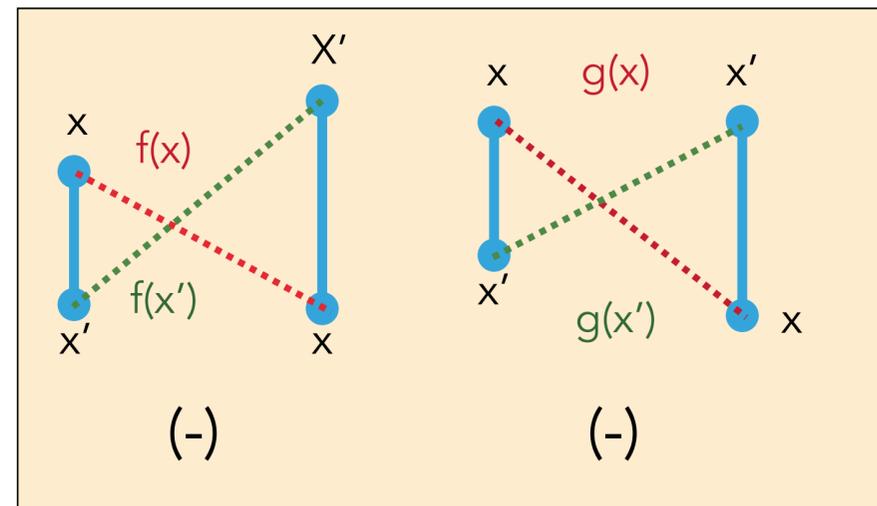
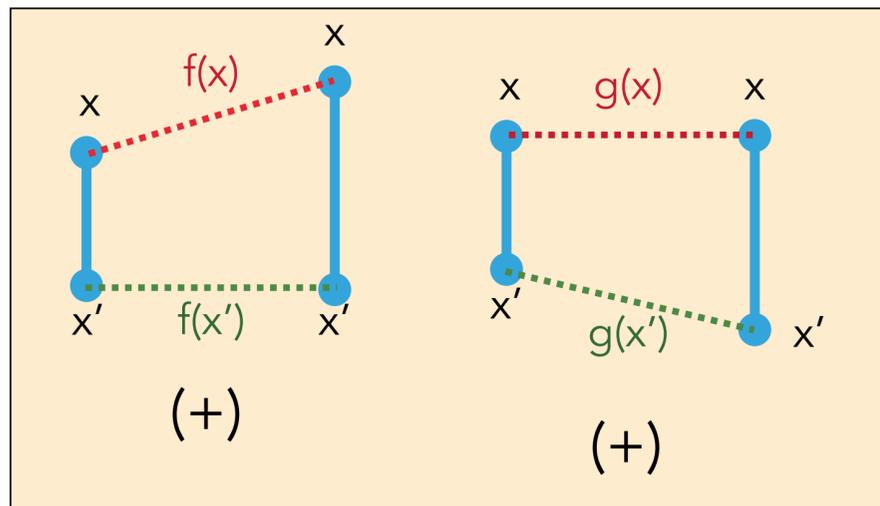
$$DO(x, x') = H \left[(f(x) - f(x')) \cdot (g(x) - g(x')) \right]$$

Heaviside(x)

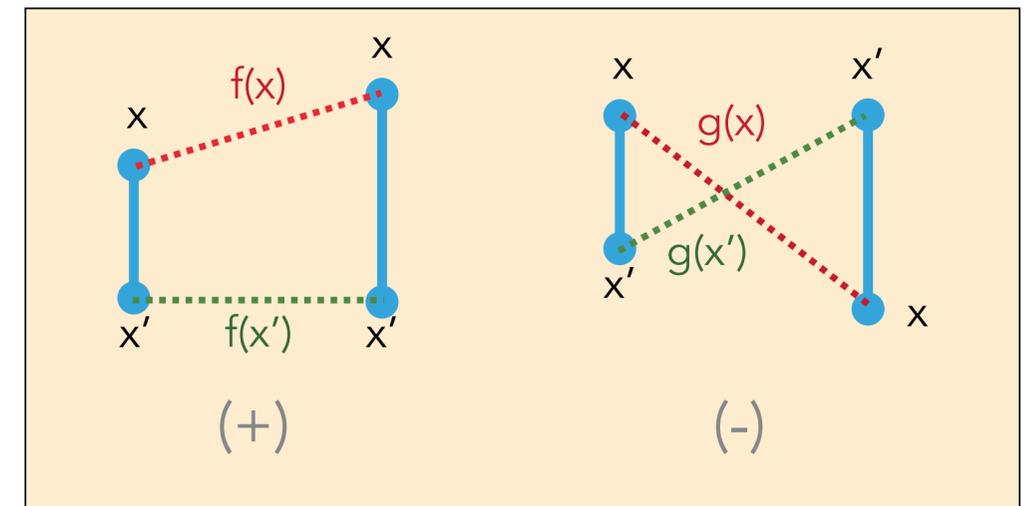


Similar Orderings $\rightarrow +1$
 Dissimilar Orderings $\rightarrow 0$

Similar Orderings



Dissimilar Orderings



Piece 3 - Average Decision Ordering (ADO)

Step 2: Average

Sum over all combinations of signal/
background decision orderings

$$ADO' = \sum DO(x, x')$$

Step 3

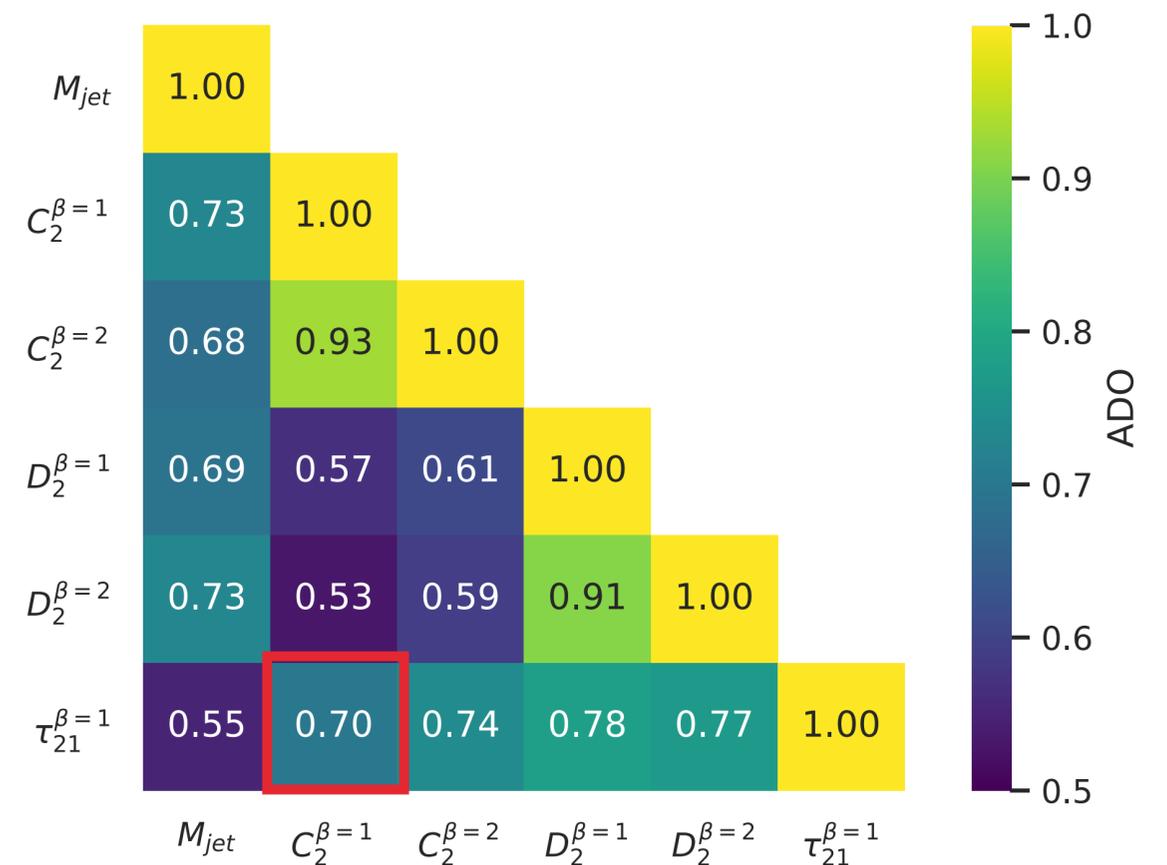
Invert averages less than 0.5 (The
decision function can be inverted to
make the opposite/correct decisions)

$$ADO = 1 - ADO'$$

- Similar performance (AUC) is not equivalent to similar decision making.
 - Example: N-subjettiness (τ_{21}) and C2B1 have the same AUC but significantly different decisions are made with those features ($ADO[\tau_{21}, C2B1]=0.70$)
- Traditional HL variables are not an "orthogonal" to one another. They share mutual information.

Observable	AUC
M_{jet}	0.898 ± 0.004
$C_2^{\beta=1}$	0.660 ± 0.006
$C_2^{\beta=2}$	0.604 ± 0.007
$D_2^{\beta=1}$	0.790 ± 0.005
$D_2^{\beta=2}$	0.807 ± 0.005
$\tau_2^{\beta=1}$	0.662 ± 0.006

AUC of 6HL Observables



ADO similarity of pairs of HL Observables

How To Find New Jet Substructure - Guided Iteration by ADO

- 3 Components

1. **CNN(LL): Low Level Model**
2. **DNN(HL): Initial High Level Observables (i.e. M_{jet} , n -subjettiness, etc)**
3. **EFP: Candidate EFPs**

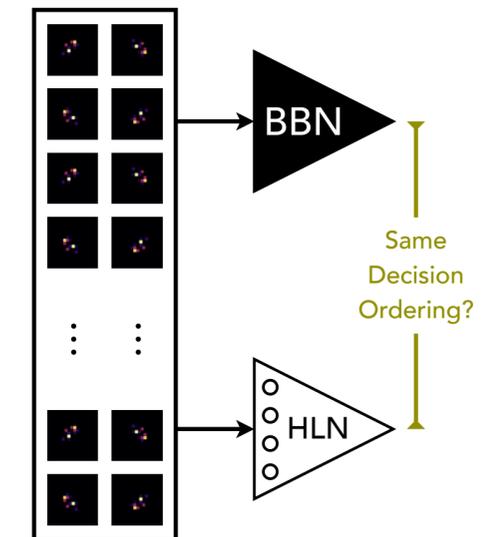
- Steps in the algorithm:

1. (pass 0): Train a NN on initial HL variables **DNN(HL)**.
2. Generate a randomized set of signal/background pairs from the data.
3. Isolate the "Differently Ordered" subset of sig/bkg pairs.
 - i.e. what pairs of events do **DNN(HL)** and **CNN(LL)** make different decisions?
4. From the "Differently Ordered" subset, find **EFP** with the maximum ADO with the **CNN(LL)**
 1. i.e. $EFP_i = \text{MAX}[\text{ADO}[EFP_i, \text{CNN(LL)}]]$.
5. Include this EFP_i into your HL Observables and return to step 1

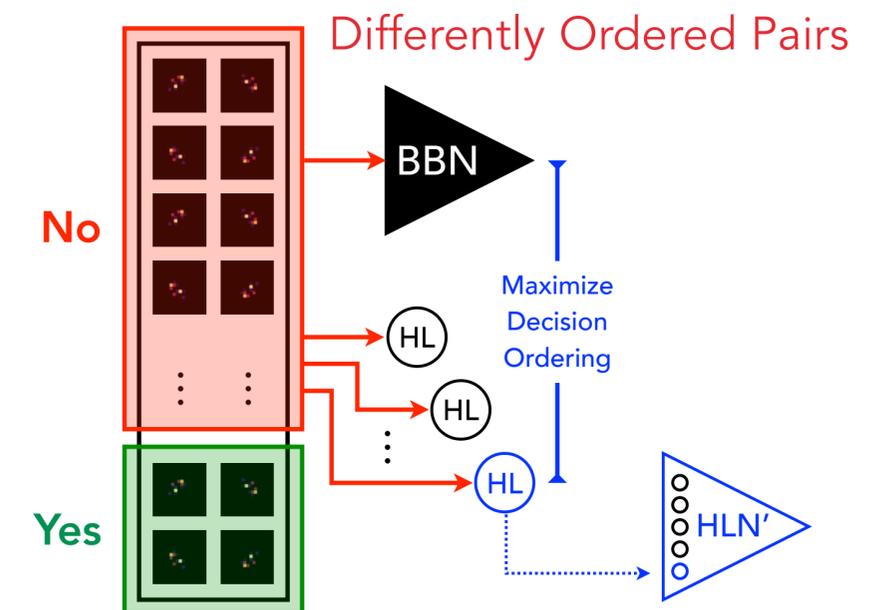
DNN(HL) becomes **DNN(HL, EFP_i)**

Step 1

Signal/Background Pairs

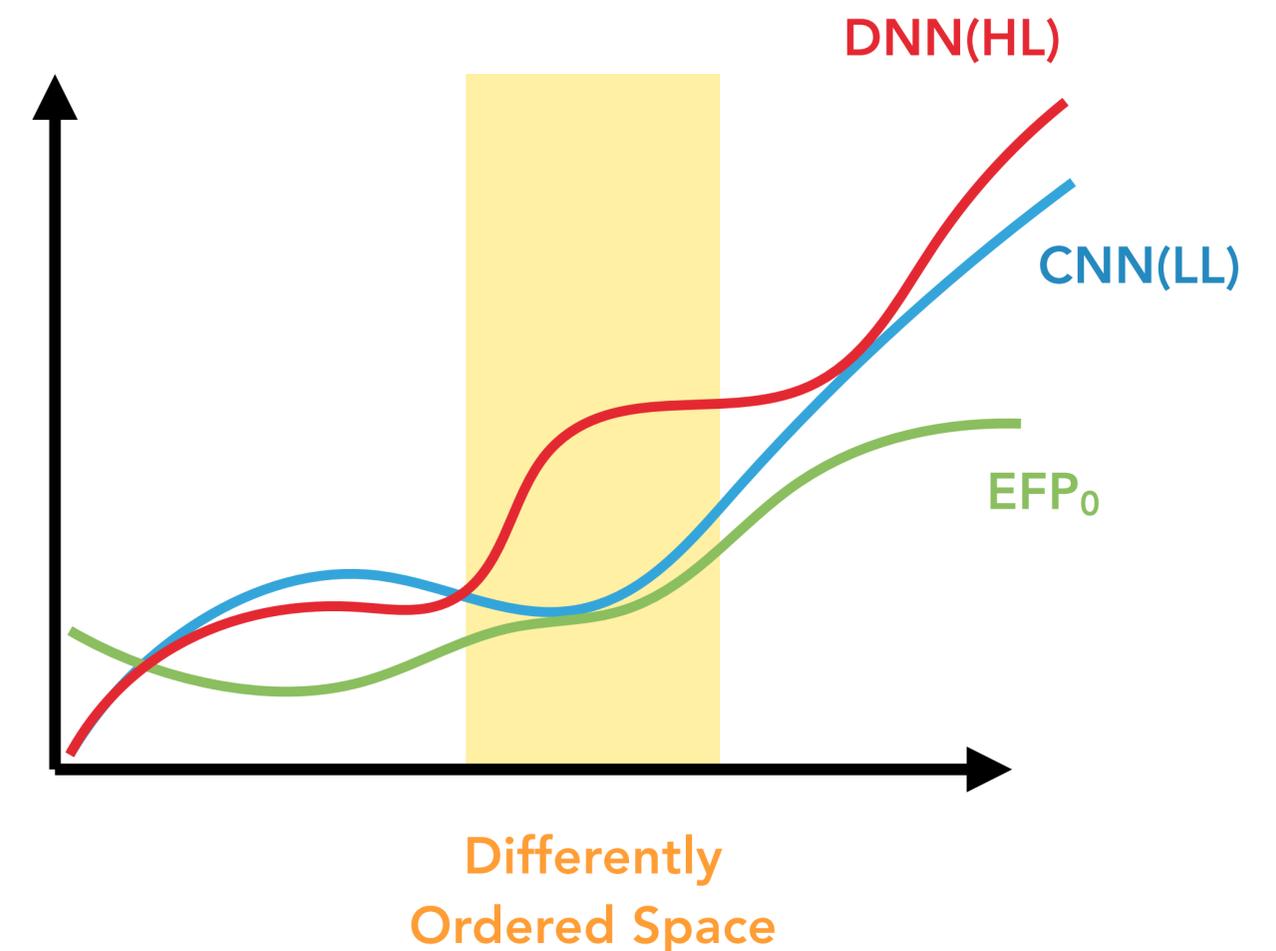


Step 3, 4, 5



What's the Logic?

- We want to iteratively build a set of useful observables.
- Each pass isolates the subspace where HL and LL are making different choices.
- We are finding the EFP most suitable to fill the gap
- Subsequent passes, DNN(HL) includes this EFP. The next EFP should be looking to fill a "new" gap in information.
- Repeat this process until they no longer disagree!

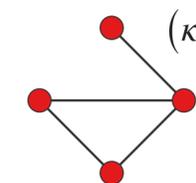


Guided Iteration - A "Supplemental Search"

Which EFP did we pick?

How many extra EFPs do we need to supplement our existing HL variables?

Just 1!



$$(\kappa=2, \beta=1/2) = \sum_{a,b,c,d=1}^N z_a^2 z_b^2 z_c^2 z_d^2 \sqrt{\theta_{ab}\theta_{bc}\theta_{ac}\theta_{ad}}$$

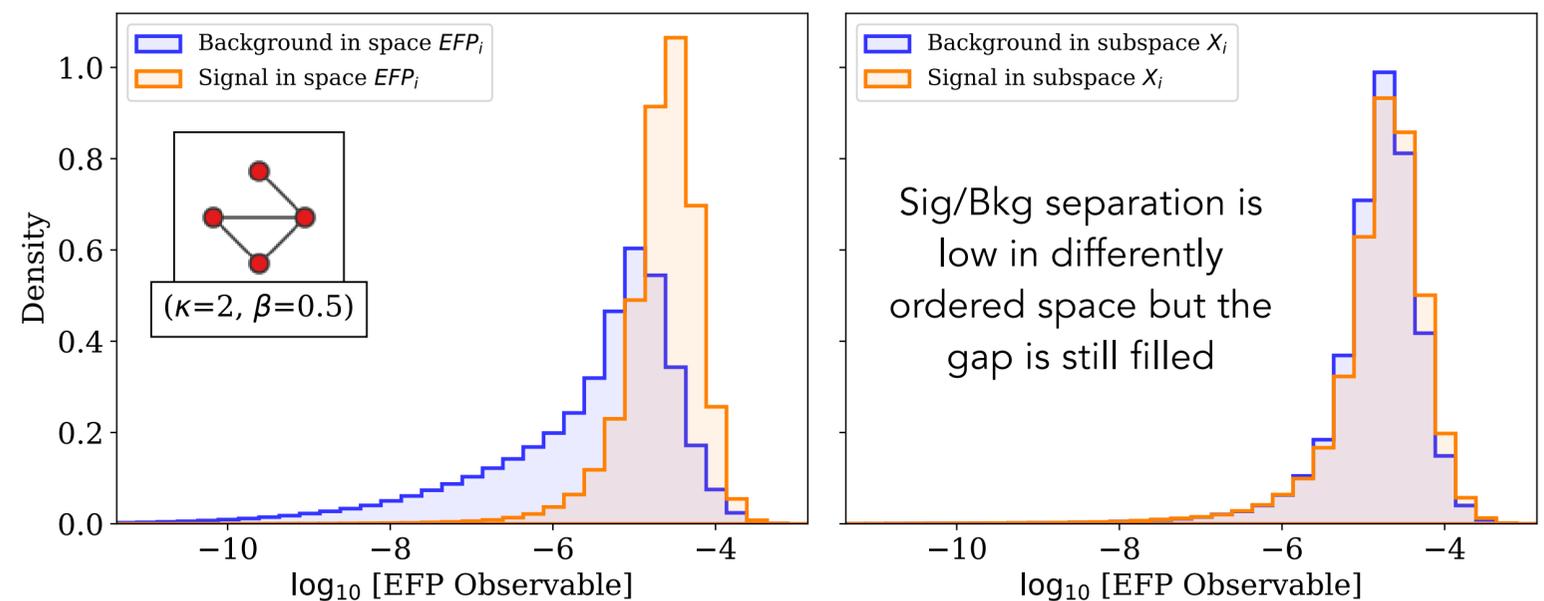
Noteworthy details

- EFP is not Infrared-safe ($k \neq 1$)
- $\beta=1/2$ is probing small-angle behaviour
- Chromatic #3 graph (probing 2-prong substructure)

Observable	AUC	ADO[CNN, Obs.]
M_{jet}	0.898 ± 0.004	0.807
$C_2^{\beta=1}$	0.660 ± 0.006	0.584
$C_2^{\beta=2}$	0.604 ± 0.007	0.548
$D_2^{\beta=1}$	0.790 ± 0.005	0.743
$D_2^{\beta=2}$	0.807 ± 0.005	0.762
$\tau_2^{\beta=1}$	0.662 ± 0.006	0.600
Existing HL 6HL	0.9504 ± 0.0002	0.971
CNN	0.9531 ± 0.0002	1.000
Existing HL +1 extra EFP 7HL _{black-box}	0.9528 ± 0.0003	0.971

AUC are equal but ADO < 1.

Equal performance but not equivalent decision makers.



Guided Iteration - A "Black Box" Approach

Alternate Approach: Start with the fewest HL observables possible (Jet Mass and Jet p_T) and ask the process to choose ALL of our EFPs from scratch.

Iteration (n)	EFP	κ	β	Chrom #	ADO[EFP, CNN] $_{x_{n-1}}$	AUC[EFP]	AUC[HLN $_n$]
0	$M_{\text{jet}} + p_T$	—	—	—	—	—	0.9119
1		2	$\frac{1}{2}$	2	0.8144	0.8190	0.9382
2		0	2	2	0.6377	0.8106	0.9458
3		0	—	1	0.5460	0.6737	0.9476
4		1	$\frac{1}{2}$	2	0.5274	0.8464	0.9487
5		-1	—	1	0.5450	0.5882	0.9504
6		1	$\frac{1}{2}$	4	0.5382	0.7678	0.9523
7		-1	$\frac{1}{2}$	2	0.5561	0.5957	0.9528

DNN(HL)

performance

improves on every

pass

We can match

performance of the LL

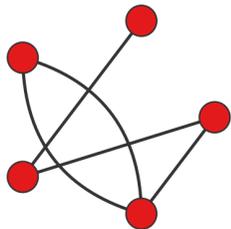
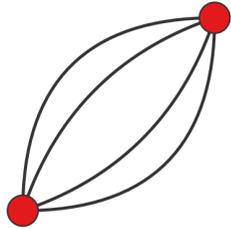
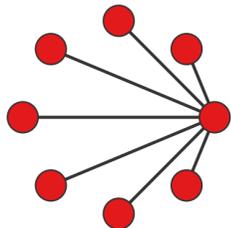
network with purely

"learned" observables

EFPs with poor individual performance contribute in a mixed dataset

Learning From the EFPs

Black Box Guided EFP Selections (1-4)

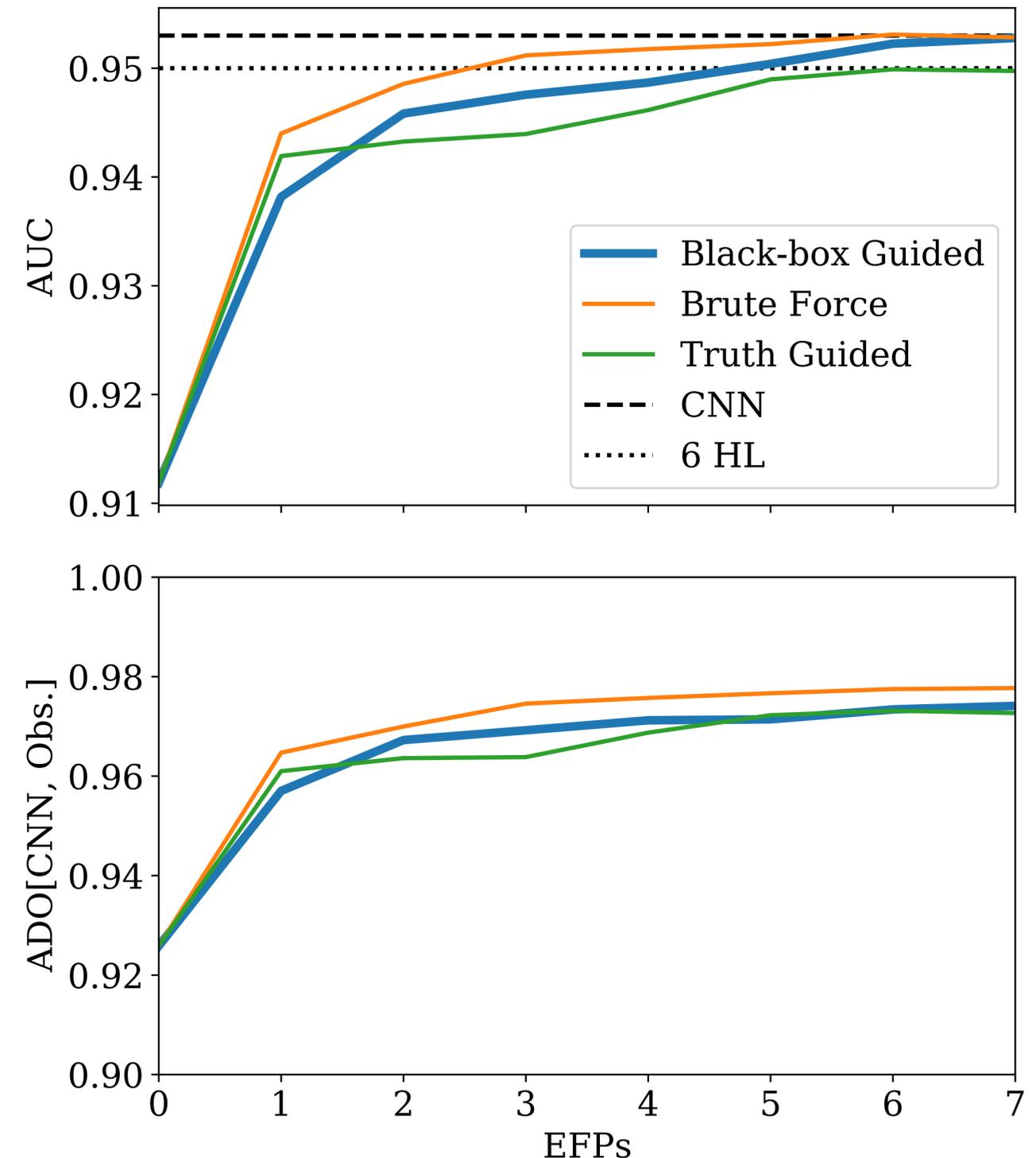
Pass	Graph	(κ, β)	Interpretation
1		$(2, 1/2)$	<ul style="list-style-type: none"> - First EFP is $\kappa=2$. This matches the Supplemental Search (IRC Unsafe) - Chromatic #2 Mass (same as Jet Mass) which probes 1-prong substructure - 5-point Correlator (different from Jet Mass, a 2-point correlator) - $\beta=1/2$ (unlike mass with $\beta=2$) probes small-angle radiation
2		$(0, 2)$	<ul style="list-style-type: none"> - Second EFP is $\kappa=0$. Also IRC Unsafe. - Also Chromatic #2 - Reinforces importance of 1-prong substructure. - $(\kappa=0, \beta=2)$ probes soft, wide-angle radiation.
3		$(0, -)$	<ul style="list-style-type: none"> - Equivalent to constituent multiplicity. This is an existing substructure observable that we simply weren't using in HL - Reinforces controlling composition of quark/gluon is important for W tagging
4		$(1, 1/2)$	<ul style="list-style-type: none"> - First IRC Safe information - Another Chromatic #2 graph (still probing 1-prong substructure) - Small angle radiation

Additional Observations

- 5 out of 7 EFPs (including the earliest choices) are IRC unsafe.
- 1-prong substructure is a strong predictor despite the fact the signal is 2-prong
- First EFP not probing 1-prong substructure is pass 6 ($c=4$)
- Although matching the CNN(LL) and Supplemental Search, the EFP choices are VERY different and "unconventional"

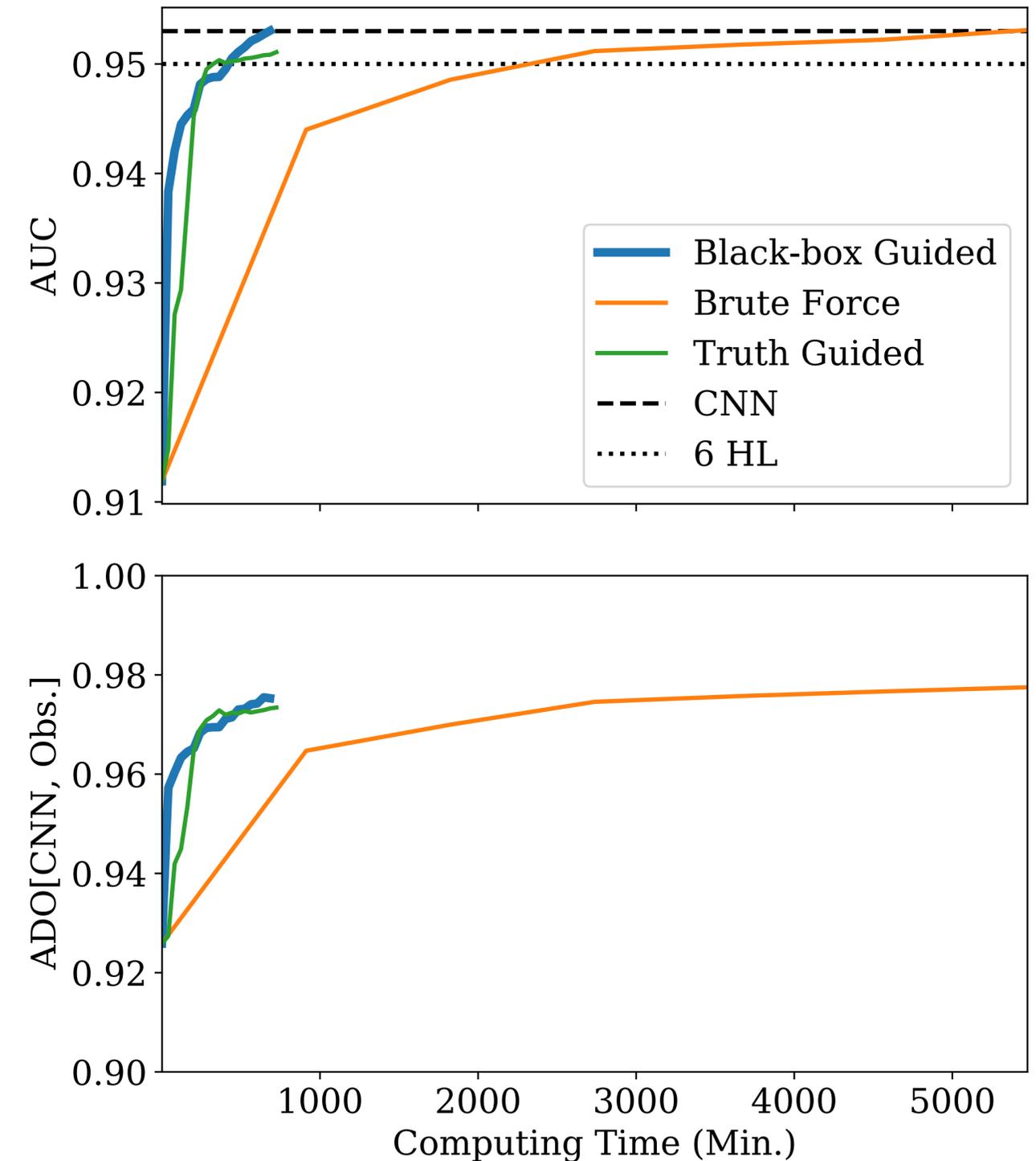
Guided Iteration - A "Black Box" Approach

- Some reasonable questions:
 - "If the information exists in the EFPs, why not just try combinations of them?"
 - "Why choose EFPs based on the CNN? Why not let ground truth (i.e. labels on your data) be your guide?"
- We compare these other methods:
 1. **Black-box Guided:** Guided Iteration by ADO with CNN(LL) as the benchmark. (our approach)
 2. **Brute Force:** Try every combination of EFP in a DNN. (Very inefficient)
 3. **Truth Guided:** Guided Iteration by ADO with ground truth as the benchmark. (Note: this is equivalent to selecting EFPs based on AUC performance)



Guided Iteration - Results

- Truth-Guided can't recover the LL performance.
 - Truth Guided attempts to optimize to truth labels even when that information isn't accessible in the data
 - The CNN is optimizing explicitly for performance and reducing/expanding the space for us. We are trying to piggy-back on that problem solving
- Brute Force gets good performance (as you would expect just trying them all) but is extremely inefficient.
 - 1 EFP selection from Brute Force takes longer than the entire guided search.
 - Hyperparameter optimization is common. Full brute force must be re-run for every change to the NN.



Overview/Discussion

- As ML methods increase in complexity, they become more opaque.
- Opaque Black Box learning shouldn't be viewed as having "learned" the data in a meaningful sense. The results can't be validated or understood. We can't inspect the Black Box. We can't measure systematic uncertainties of the Black Box.
- Where possible, use interpretable models that match opaque solutions.
- When equivalent performance is not possible with interpretable models, a Black Box can tell you how much performance is possible but an interpretable approach (like Guided Iteration by ADO) should be used to try and recover that information.
- Answers we don't understand aren't really "answers" at all.

Questions ?