August 20, 2019
25th International Summer Institute
@ Sandpine, Gangneung, Korea

Tagging boosted weak gauge bosons with deep learning

Cheng-Wei Chiang
National Taiwan University

Yu-Chen Janice Chen, CWC, Giovanna Cottin, David Shih, 1908.08256 [hep-ph]

Outline

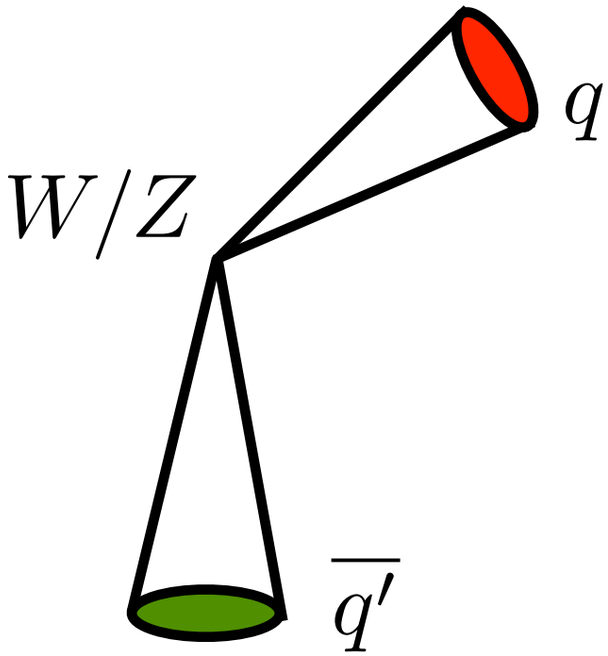
- Motivations
- Deep learning and existing jet taggers
- Our taggers
- Performance
- Summary

MOTIVATIONS

Motivations

- Weak boson scatterings at high energy provide a direct probe of the EWSB mechanism.
- New physics particles, such as Z', W',or heavy Higgs, often decay to weak bosons.
- Such weak bosons are generally highly boosted and, when decaying hadronically, form one collimated jet.

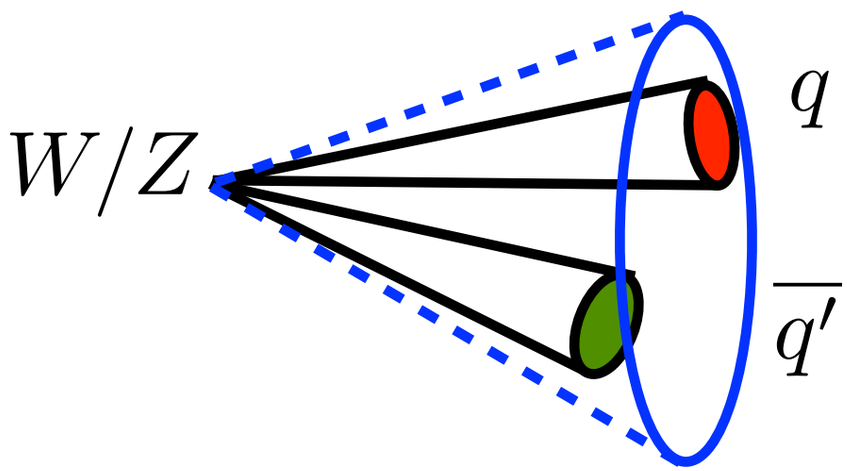
low momentum



two resolvable jets

boost

high momentum

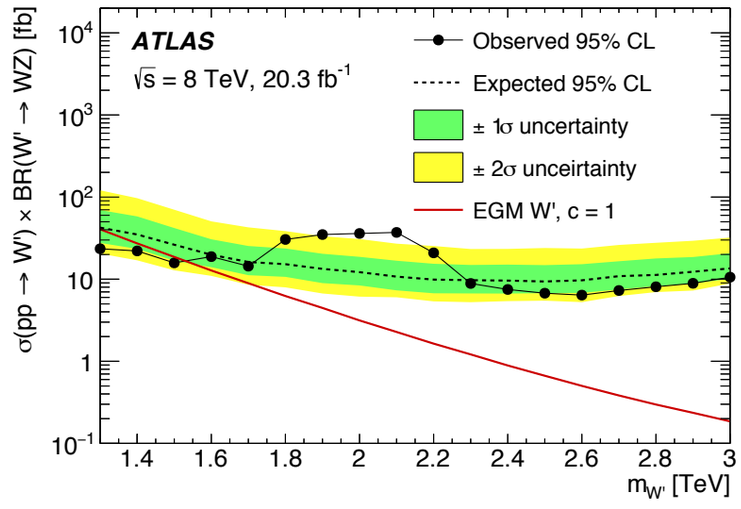


one boosted fat jet

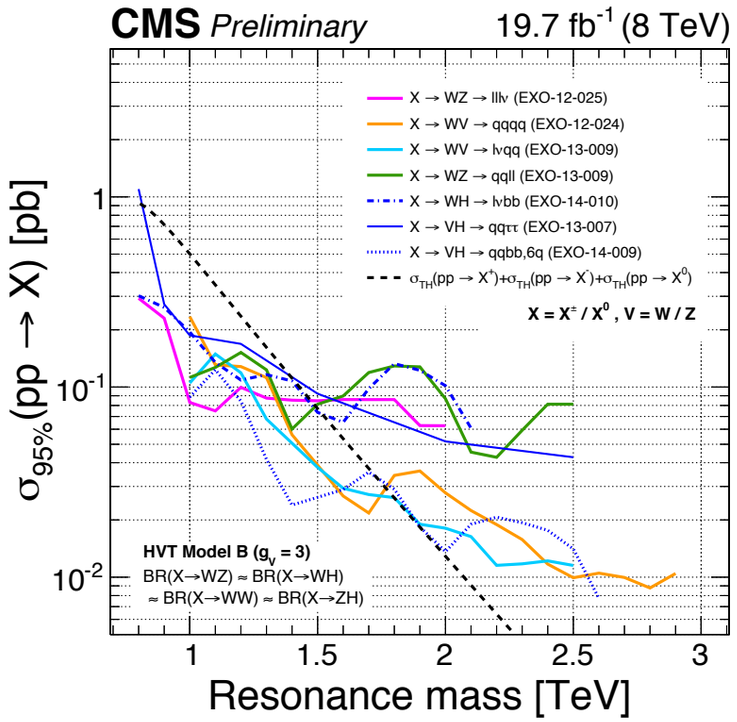
Diboson Resonance Search

- Using jet mass and jet substructure properties, ATLAS searched for high-mass diboson resonances in the mass range of 1.3 to 3.0 TeV using the invariant mass distribution of dijets, each of which tagged as a hadronically decaying boosted W or Z boson.
- 2-TeV resonances in the WZ, WW and ZZ channels at 3.4σ, 2.6σ and 2.9σ, respectively, were suspected.
- Similar analyses and results by CMS were also reported.
- No charge information was used.

ATLAS 2015



CMS 2015



Doubly Charged Higgs

- The only possible interactions of doubly-charged Higgs with SM particles allowed by the symmetries are:

proportional to g and v_Δ

$$g v_\Delta \Delta^{++} W^- W^- + \text{h.c.}$$

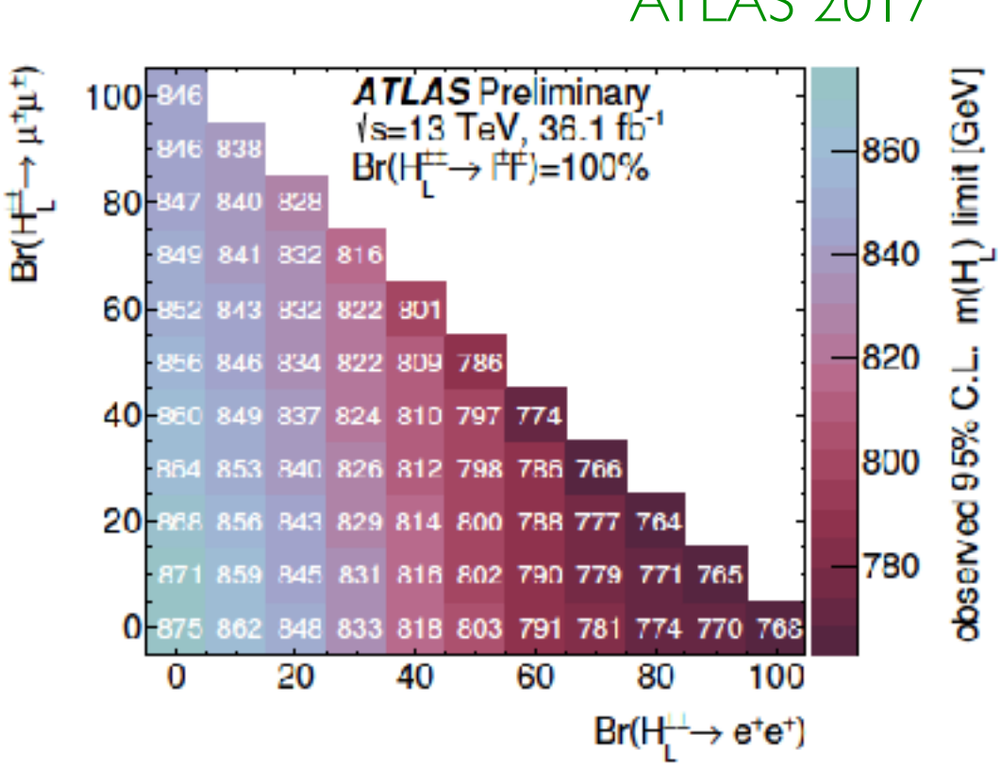
Like-sign final states

$$y_{ij} \Delta^{++} \ell_i^- \ell_j^- + \text{h.c.}$$

LNV

generally LFV, related to neutrino mass and mixing data

ATLAS 2017



A general lower bound of ~800 GeV @ NTU

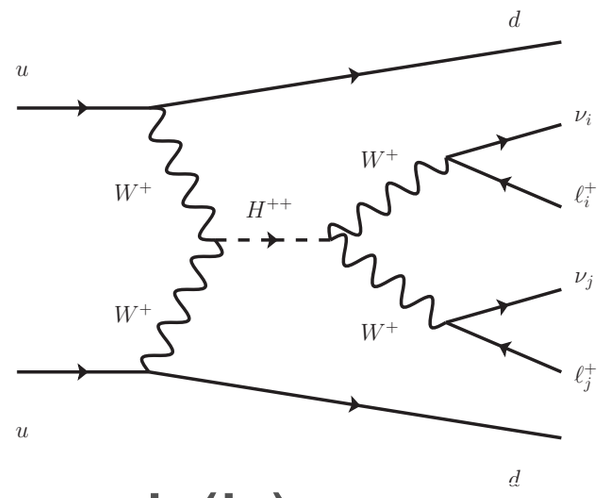
- Lots of experimental efforts in the scenario with the latter type of interaction being dominant (smaller triplet VEV, thus larger Yukawas).

➠ time for the large v_Δ scenario

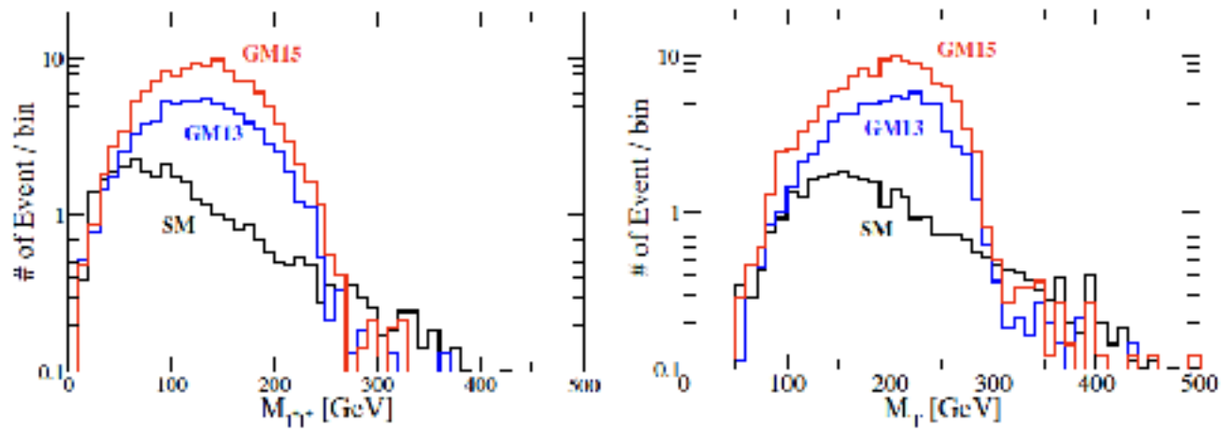
Golden Channel

- From differential distributions, particularly (a) $m_{\ell\ell'}$ and (b) cluster transverse mass

$$m_T^2 \equiv \left[\sqrt{m^2(\ell\ell) + p_T^2(\ell\ell)} + |\not{\mathbf{p}}_T|\right]^2 - \left[\mathbf{p}_T(\ell\ell) + \not{\mathbf{p}}_T\right]^2$$



one can observe (a) a bump ending at $m_{H^{++}}$ and (b) a Jacobian-like peak edged at $m_{H^{++}}$.



CWC, Kuo, Yagyu 2013

- Small BR's for leptonic modes, involving missing energy
  - ➠ what about hadronic/semi-leptonic mode (larger BR's)?
  - ➠ reliable to determine charge?

DEEP LEARNING AND EXISTING JET TAGGERS

Machine Learning

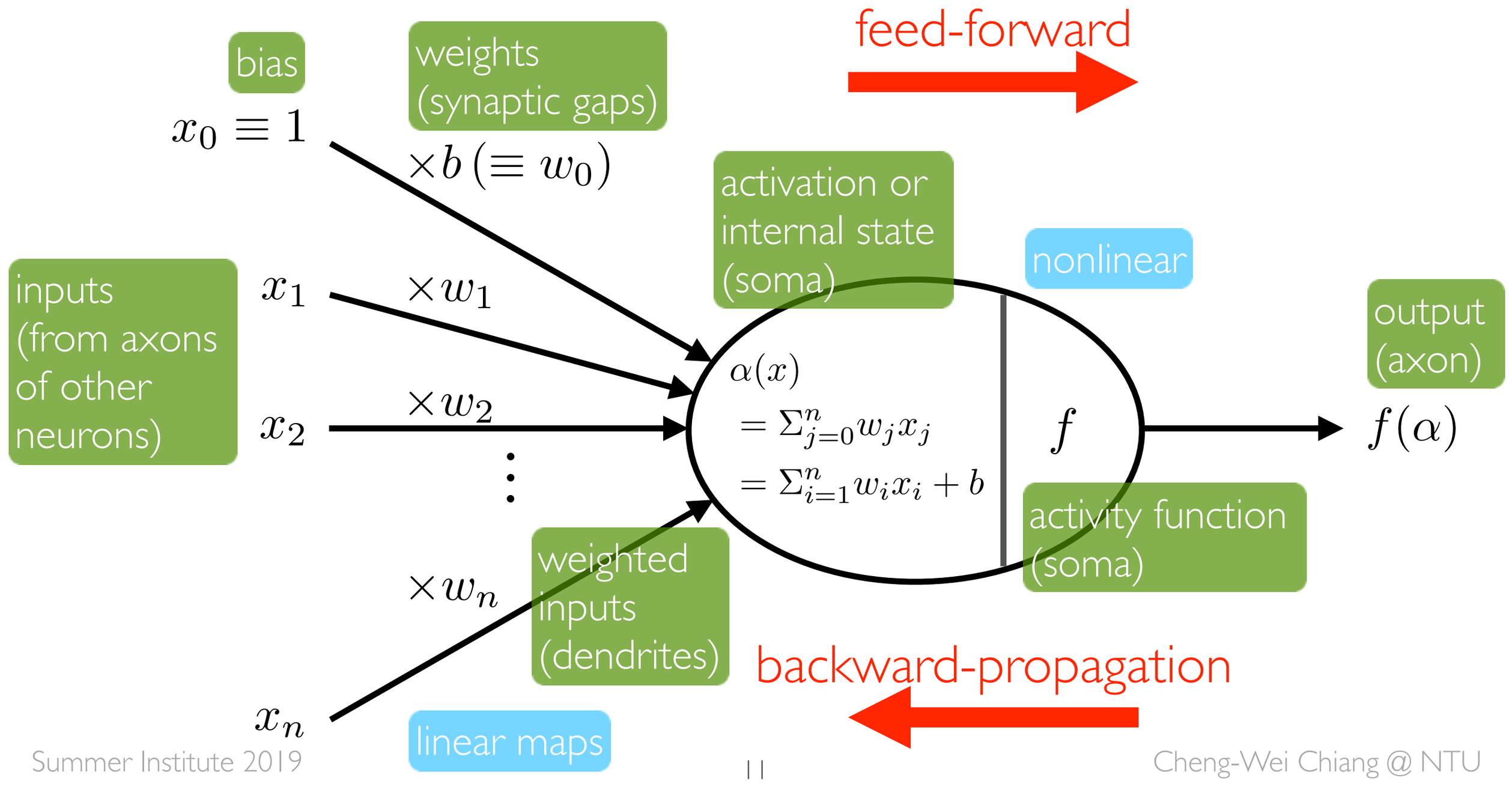
- ML is the tool used for large-scale data processing and is well suited for complex datasets with huge numbers of variables and features (patterns and regularities), especially for deep learning neural networks (NNs).
- The Universal Theorem: any function can be approximated by a neural network with at least one hidden layer.
- For a long time, given this theorem and the difficulty in complex networks, people have restricted themselves to shallow networks with only one hidden layer.
- Recently, people have realized that deeper, more complex networks with many hidden layers can understand higher levels of abstraction than shallow layers.

Resurgence of NN

- NNs had become popular and then forgotten at least twice before.
- They have resurged in the last decade partly due to:
  - having faster computers, with the use of GPUs versus the traditional use of CPUs,
  - better algorithms and neural nets design, and
  - increasingly large datasets.

Artificial Neuron

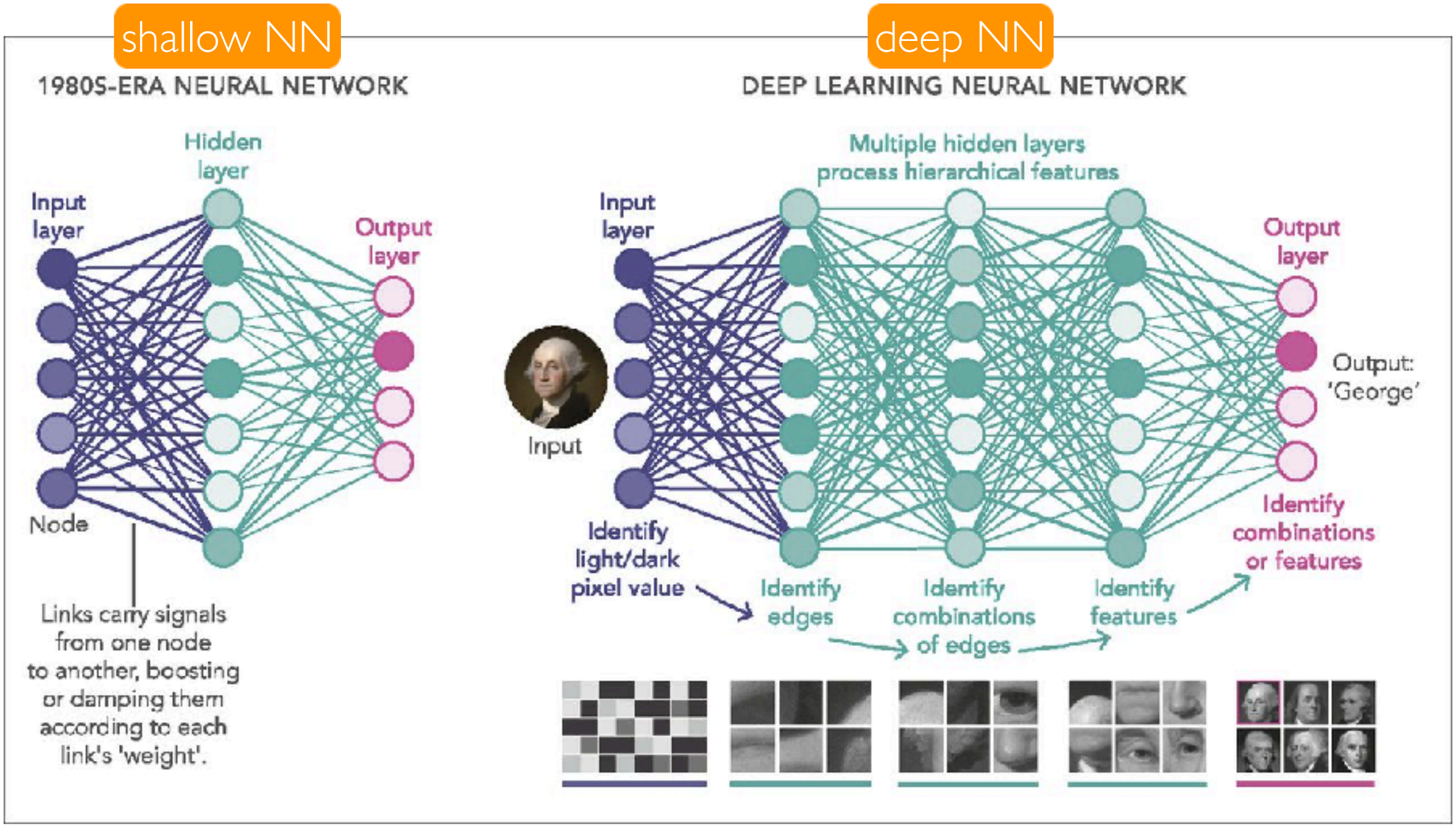
- Different types of artificial neurons are modeled using different types of activity functions.



Deep Neural Network

shallow NN

deep NN



Waldrop 2019

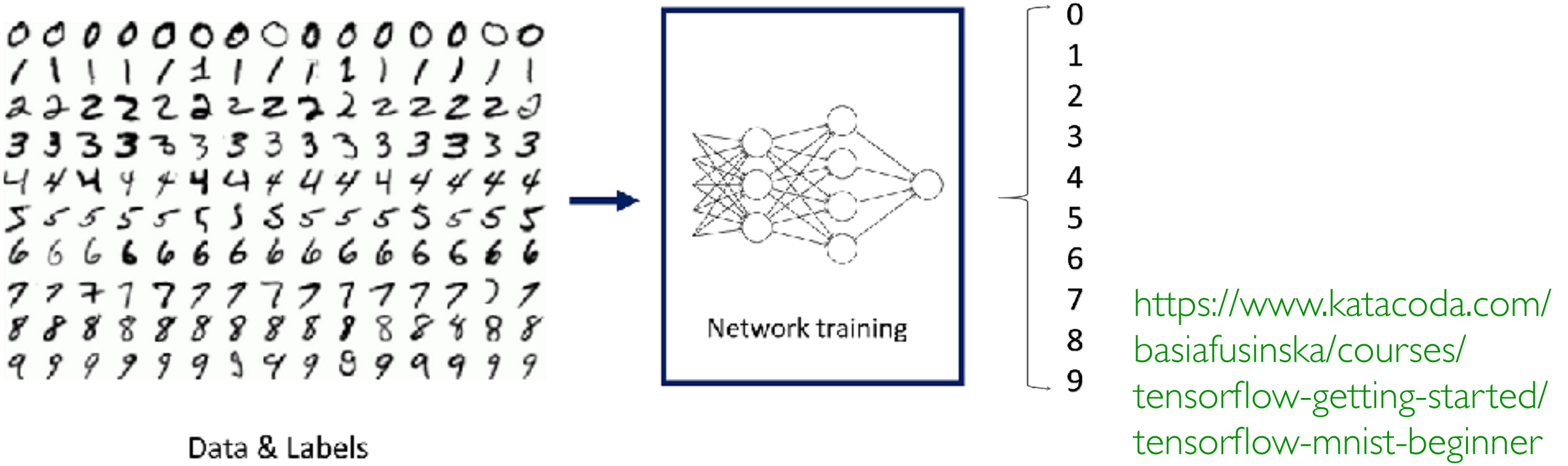
Common NN Types

- Dense neutral network (DNN): a network with standard fully-connected feed-forward layers that take flattened vectors as the input, prototypical for most tasks; sometimes also called multi-layer perceptron (MLP).
- Recurrent neural network (RNN): a network that deals with sequences of variable length by defining a recurrence relation over these sequences, suitable for natural language processing and speech recognition tasks.
- Convolutional neural network (CNN): a network with special layers that filter data, suitable for computer vision.
  ➠ ideal for jet image recognition task in collider physics

*Some evidence shows that neurons in CNNs are organized in a way similar to biological cells in the visual cortex of the human brain.

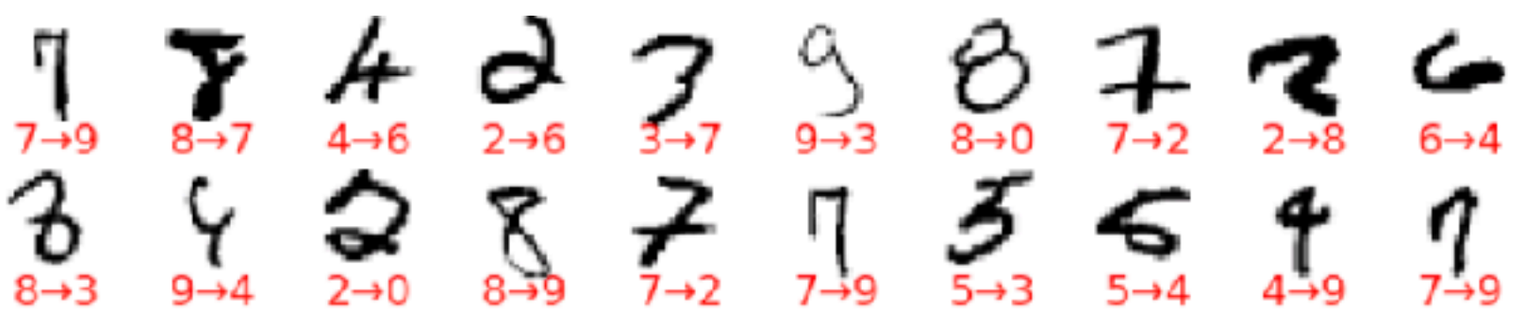
Script Digits Recognition

- One of the most classic example of CNN is recognizing hand-written digits (with 60,000 training images and 10,000 testing images, and each image being normalized to 28×28 pixels and have 256 grey levels).



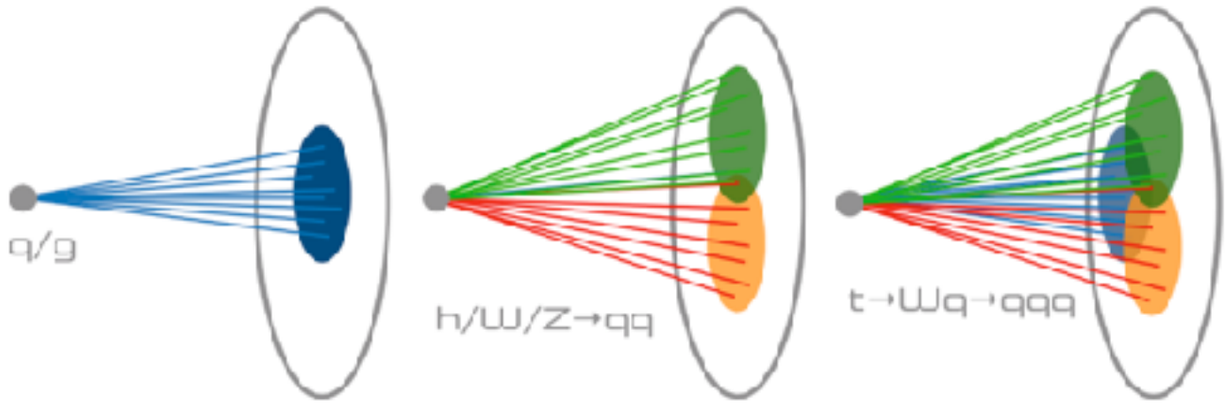
https://www.katacoda.com/basiafusinska/courses/tensorflow-getting-started/tensorflow-mnist-beginner

passing test samples to NN gives an accuracy of ~99%, with some mistakes from time to time:



Jet Tagging

- In past decade or so, lots of efforts have been spent on classifying jets using jet substructure, according to the distribution of energy within jets.



Moreno et al 2019

- In addition to usual QCD jets (light quarks, b-quark, and gluons), the large collision energy of LHC produces new classes of jets with collimated prongs, derived from boosted W, Z, t-quark, or Higgs boson.
- More recently, jet tagging has become one of the deep ML exercises in particle physics.

de Oliveira et al 2016
Larkoski, Moult, Nachman 2017

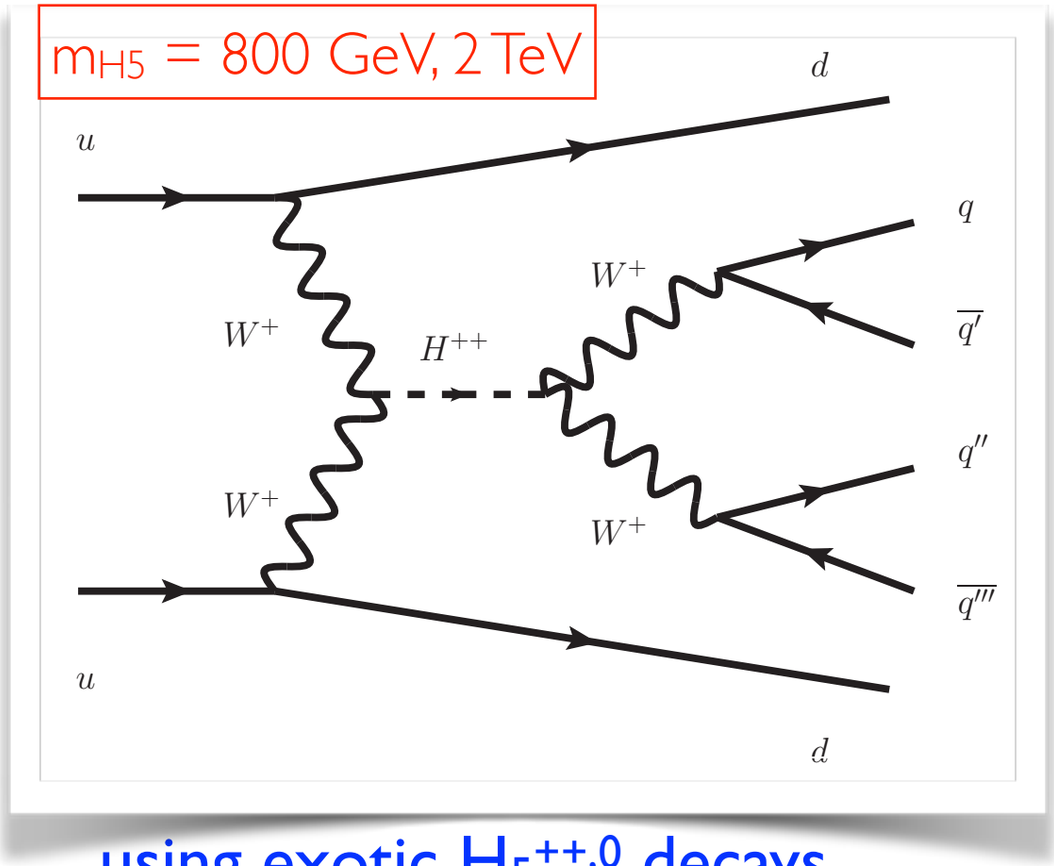
Existing Jet Classifiers

- Jet flavor (light or heavy origin) tagging — Guest et al 2016
- Top tagging — Pearkes, Fedorko, Lister, Gay 2017; Egan, Fedorko, Lister, Pearkes, Gay 2017; Kasieczka, Plehn, Russell, Schell 2017; Butter, Kasieczka, Plehn, Russell 2018; Macaluso, Shih 2018; Butter et al 2019
- Quark/gluon tagging — Komiske, Metodiev, Schwartz 2017; Butter, Kasieczka, Plehn, Russell 2018; Macaluso, Shih 2018; Fraser, Schwartz 2018
- Boosted Z-jet tagging (from QCD-jets) — Larkoski, Salam, Thaler 2013; Larkoski, Moult, Neill 2016
- Boosted W-jet tagging (from QCD-jets) — Cui, Han, Schwartz 2011

OUR TAGGERS

Sample Preparation

- Physical process:



using exotic $H_5^{++,0}$ decays
in Georgi-Machacek model
@ 13-TeV LHC

- Simulations:

parton-level processes
➠ MG5 aMC@NLOv2.6.1

showering and hadronization
➠ PYTHIA 8.2.19

detector simulation
➠ DELPHES 3.4.1 w/ CMS card

jet reconstruction
➠ FastJet 3.1.3

Jet Samples

- Jet selection:

m_{H5} = 800 GeV

Jet sample	$p_T \in (350, 450)$ GeV, $\lvert\eta\rvert \leq 1$ jets with anti-k_T and $R = 0.7$ V-V merging : $\Delta R(V_1, V_2) < 0.6$ V-jet matching : $\Delta R(V, j) < 0.1$

- Sample sizes:

Jet sample size		
	Training set	Testing set
W^+	188k	38k
W^-	198k	40k
Z	175k	35k

90% : true training set
10% : validation set

Higher-Level Inputs

- Traditional analyses make use of higher-level observables:

Jet invariant mass

$$\mathcal{M}_J^2 = \left(\sum_{i\in J} E_i\right)^2 - \left(\sum_{i\in J} \mathbf{p}_i\right)^2$$

Jet charge

$$\mathcal{Q}_\kappa = \frac{1}{p_{T,J}^\kappa} \sum_{i\in J} q_i \times (p_T^i)^\kappa$$

where J denotes a jet, i runs over jet constituents (tracks) with $p_T > 500$ MeV, q_i is the integer charge of i in units of proton charge, and κ is a free parameter.

- Q_κ is computed in this p_T-weighted scheme in the hope of minimizing mis-measurements from low-p_T particles.

Higher-Level Inputs

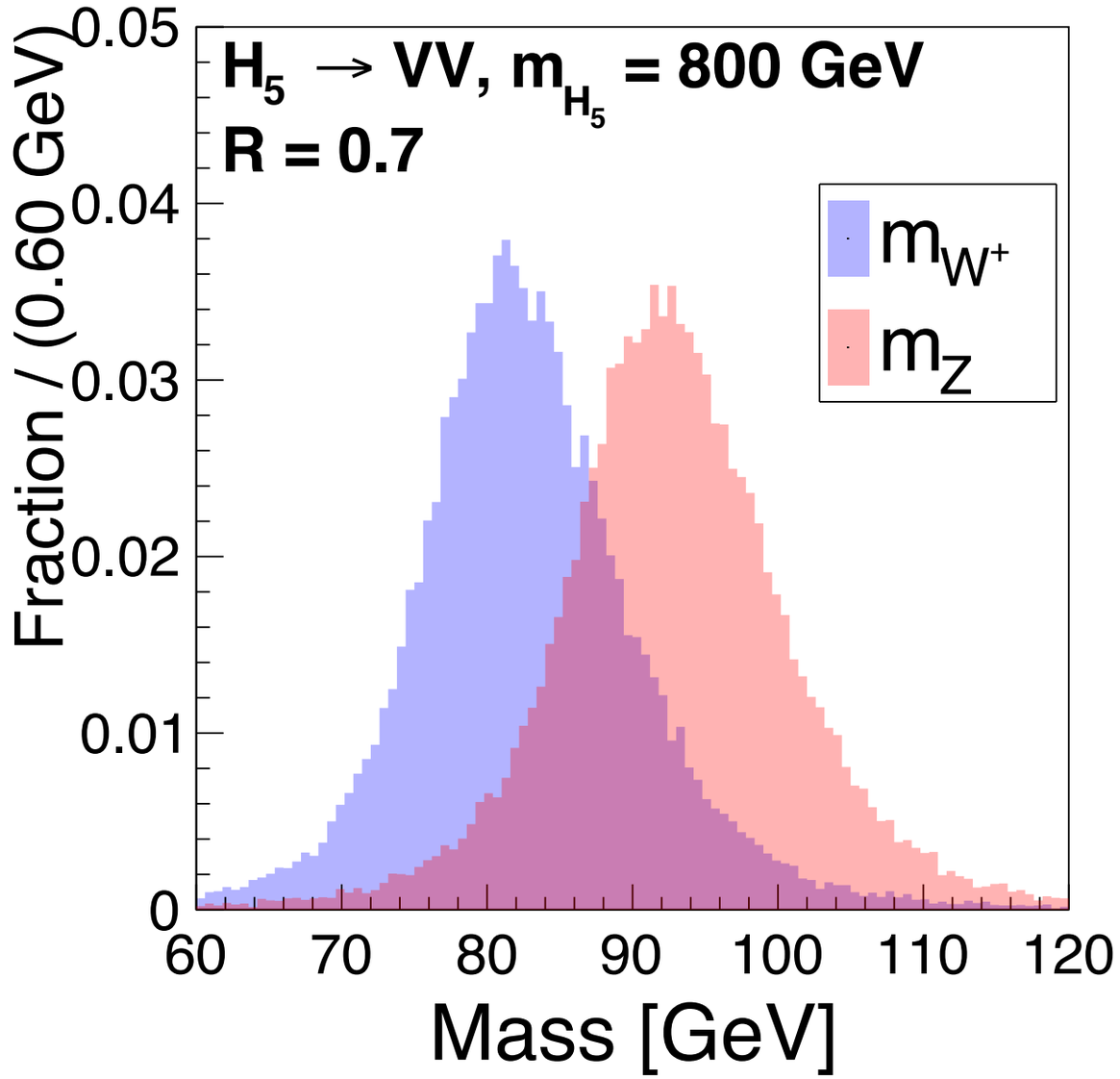
- Traditional analyses make use of higher-level observables:

Jet invariant mass

$$\mathcal{M}_J^2 = \left(\sum_{i\in J} E_i\right)^2 - \left(\sum_{i\in J} \mathbf{p}_i\right)^2$$

- The broader widths in the mass distribution originate from a combination of showering, hadronization, jet clustering and detector effects.

➠ no clear boundary



Higher-Level Inputs

- Traditional analyses make use of higher-level observables:

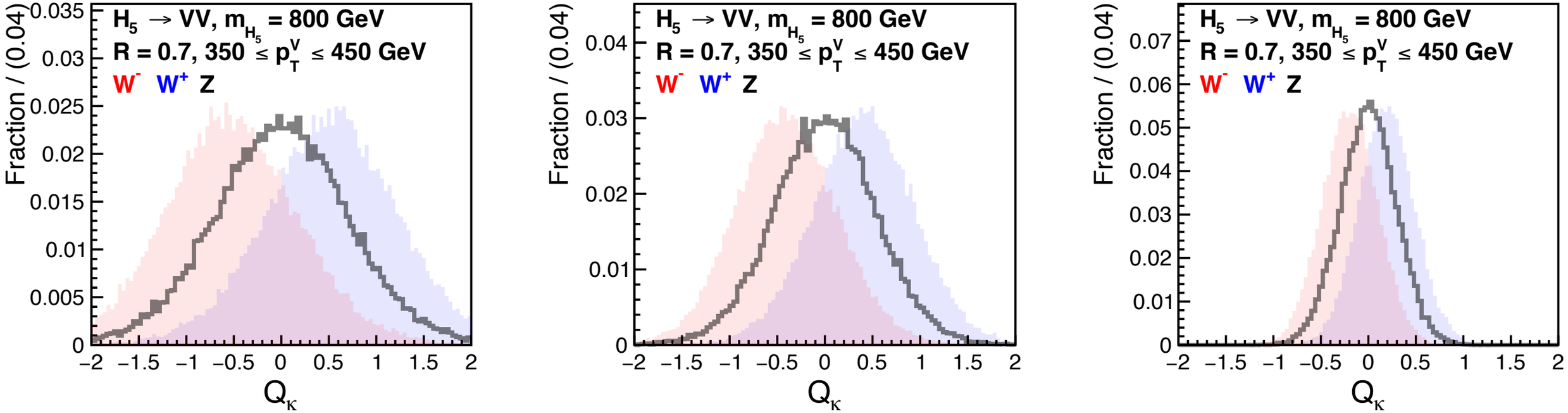
Jet charge

$$Q_\kappa = \frac{1}{p_{T,J}^\kappa} \sum_{i \in J} q_i \times (p_T^i)^\kappa$$

p_T-weighted scheme:
- $\kappa = 0$ ➠ equal-weight
- $\kappa = 1$ ➠ proportional to p_T

Field, Feynman 1978

jet charge (κ = 0.2) | jet charge (κ = 0.3) | jet charge (κ = 0.6)



- The separation is not well because of the choices of weight factor κ, jet cone size R, etc.

Reference Taggers

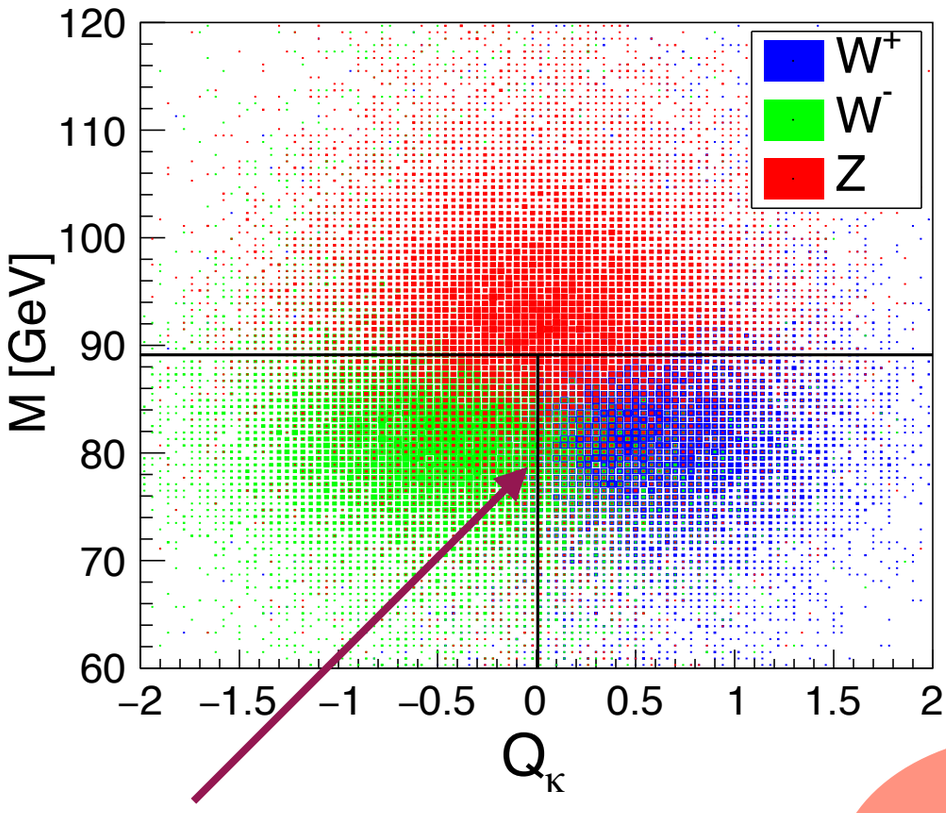
- Cut-based tagger
  impose simple 2D rectangular cuts in the (M,Q_κ) plane, and optimize the overall accuracy
- single-κ boosted decision tree (BDT) tagger
  choose a specific κ, implement with `sklearn` package, and assume default parameters
- multi-κ boosted decision tree (BDT) tagger
  same as above, but allowing κ = 0.2, 0.3, or 0.4

- All use high-level inputs (M,Q_κ).
- single-κ BDT, when taking the optimal κ value, generally has a comparable performance as the multi-κ BDT.

Reference Taggers

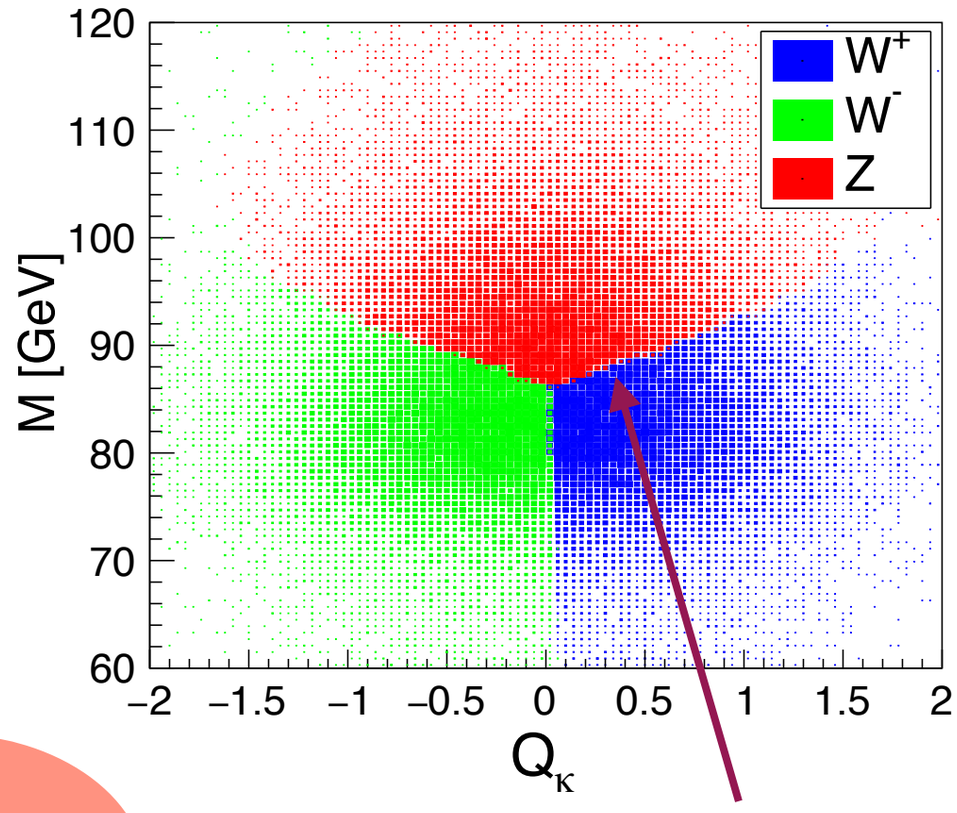
- For the ternary ($W^+/W^-/Z$) classification task, the reference taggers can be visualized as follows:

cut-based tagger

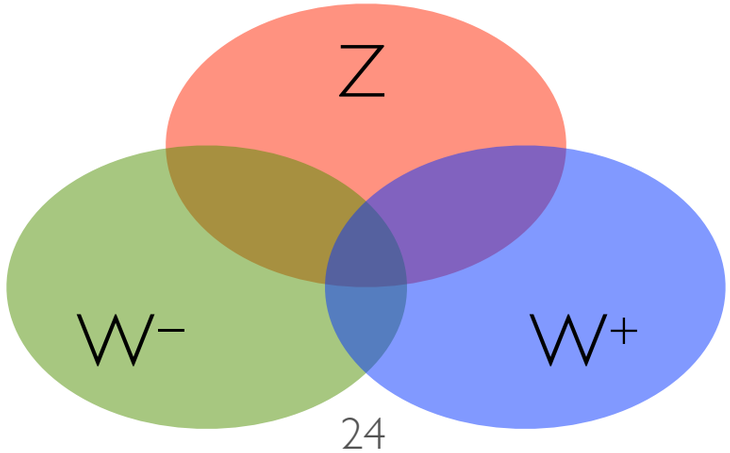


rectangular cuts

single-κ BDT (κ=0.3)



Y-shaped cuts

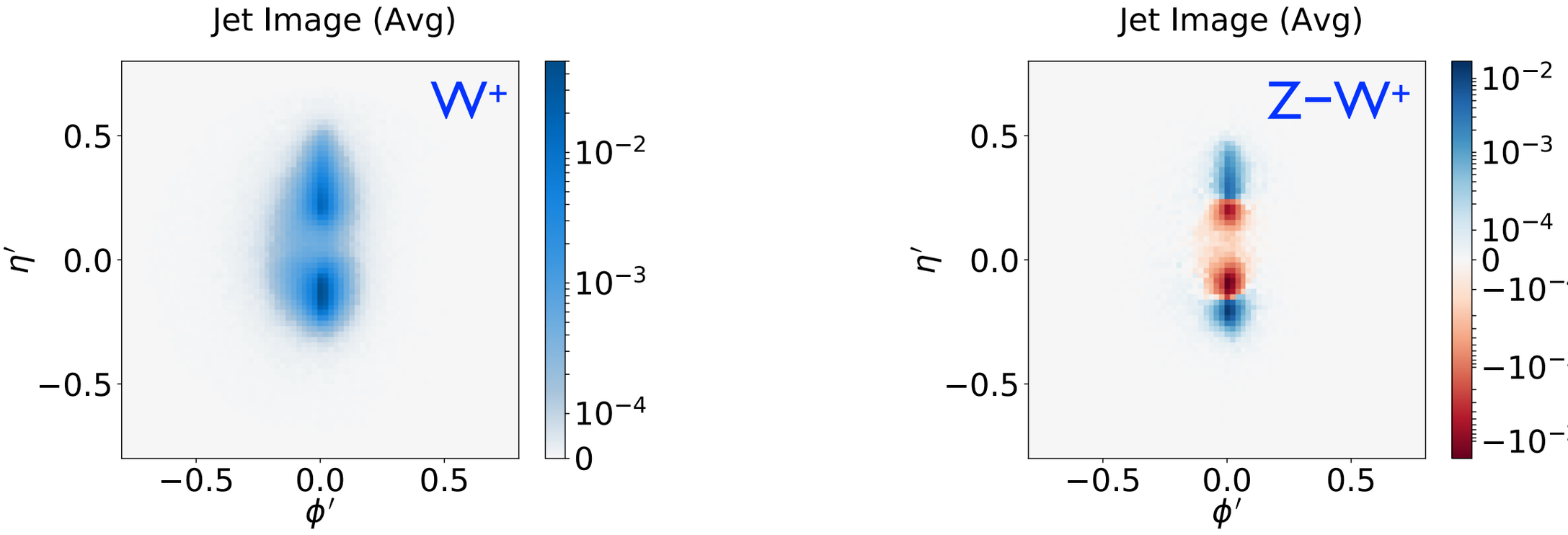


Jet Images and Channels

- Deep learning based taggers studied in our work are based on jet images, utilizing lower-level inputs and processed by CNNs.
- Jet images are made from jets reconstructed in a box of $\Delta\eta = \Delta\phi = 1.6$ (central region) with 75×75 pixels.
  ➠ a resolution consistent with that of the CMS ECal
- The input variables or channels are Q_K and p_T per pixel.
  ➠ now the sum $\sum_{i \in J}$ is done within each pixel

Lower-Level Inputs

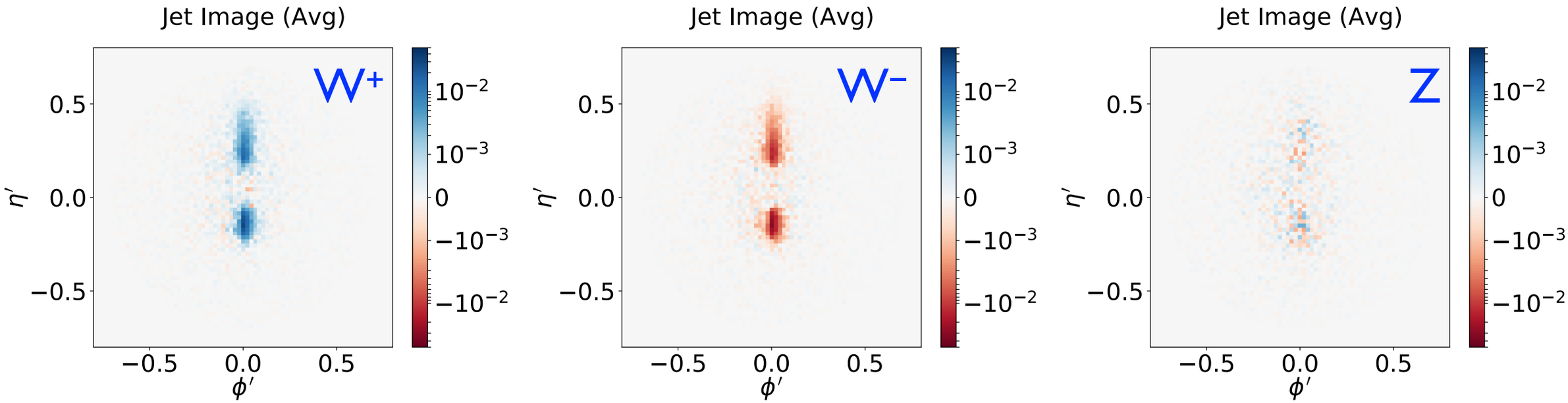
- Preprocess each image, involving centralization, rotation and flipping (➠ jet with larger p_T is in first quadrant).
- p_T channel:



- Z average jet images have a wider distribution in ΔR than W jets, as expected from its larger invariant mass.

Lower-Level Inputs

- Preprocess each image, involving centralization, rotation and flipping (➠ jet with larger p_T is in first quadrant).
- Q_κ channel:



- The average Z jet charge image is close to zero as the constituent charges in different events tend to cancel out.

A Typical CNN

Skansi 2018

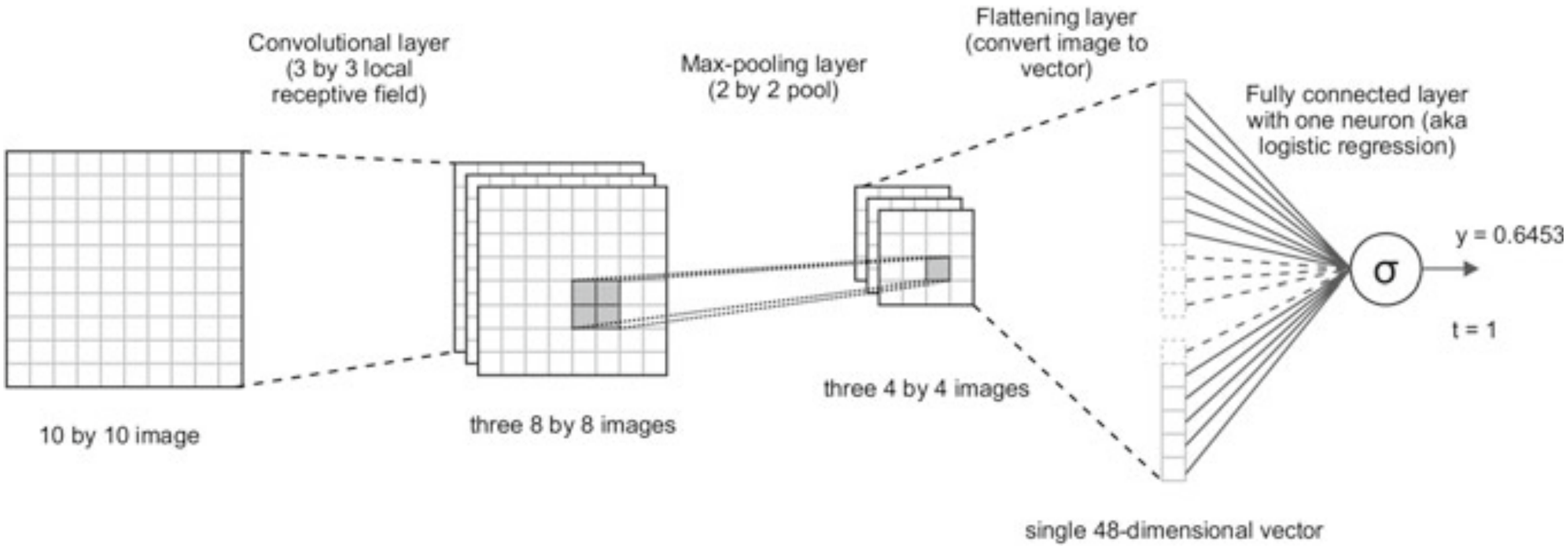


Fig. 6.3 A convolutional neural network with a convolutional layer, a max-pooling layer, a flattening layer and a fully connected layer with one neuron

Our CNN Taggers

	CNN	CNN^2	
Image	(75×75) pixels within $(\lvert\eta\rvert \leq 0.8, \lvert\phi\rvert \leq 0.8)$		
Channels	p_T, $\mathcal{Q}_\kappa$	p_T	$\mathcal{Q}_\kappa$
Architecture	BN-32C6-MP2-128C4-MP2-256C6-MP2-512N-512N	BN-32C3-32C3-MP2-64C3-MP2-64C3-MP2-64C3-64C3-128C5-256C5-256N-256N	BN-32C3-32C3-MP2-64C4-64C4-MP2-256C6-MP2-256N
Settings	`Relu` Activation, `Padding=same`, *Dropout* $= 0.5$, l2 `Regularizer` $= 0.01$		
Preprocessing	*Centralization, Rotation, Flipping*		
Training	Adam Optimizer, `Minibatchsize`=512, *Cross entropy* loss		

using Keras library with TensorFlow backend

Our CNN Taggers

- a deeper Q_κ network tends to overfit W^+/W^-
- a deeper p_T network helps identifying Z

	CNN	CNN^2	
Image	(75×75) pixels within $(\lvert\eta\rvert \leq 0.8, \lvert\phi\rvert \leq 0.8)$		
Channels	p_T, $\mathcal{Q}_\kappa$	p_T	$\mathcal{Q}_\kappa$
Architecture	BN-32C6-MP2-128C4-MP2-256C6-MP2-512N-512N	BN-32C3-32C3-MP2-64C3-MP2-64C3-MP2-64C3-128C5-256C5-256N-256N	BN-32C3-32C3-MP2-64C4-64C4-MP2-256C6-MP2-256N
Settings	`Relu` Activation, `Padding=same`, *Dropout* $= 0.5$, l2 `Regularizer` $= 0.01$		
Preprocessing	*Centralization, Rotation, Flipping*		
Training	Adam Optimizer, `Minibatchsize`$=512$, *Cross entropy* loss		

using Keras library with TensorFlow backend

Our CNN Taggers

- a deeper Q_κ network tends to overfit W^+/W^-
- a deeper p_T network helps identifying Z

	CNN	CNN^2	
Image	(75×75) pixels within $(\lvert\eta\rvert \leq 0.8, \lvert\phi\rvert \leq 0.8)$		
Channels	p_T, $\mathcal{Q}_\kappa$	p_T	$\mathcal{Q}_\kappa$
Architecture	BN-32C6-MP2-128C4-MP2-256C6-MP2-512N-512N	BN-32C3-32C3-MP2-64C3-MP2-64C3-MP2-64C3-64C3-128C5-256C5-256N-256N	BN-32C3-32C3-MP2-64C4-64C4-MP2-256C6-MP2-256N
Settings	`Relu` Activation, `Padding=same`, *Dropout* $= 0.5$, l2 `Regularizer` $= 0.01$		
Preprocessing	*Centralization, Rotation, Flipping*		
Training	Adam Optimizer, `Minibatchsize`$=512$, *Cross entropy* loss		

activated to enable
a deeper network

using Keras library with TensorFlow backend

Our CNN Taggers

- a deeper Q_κ network tends to overfit W^+/W^-
- a deeper p_T network helps identifying Z

	CNN	CNN^2	
Image	(75×75) pixels within $(\lvert\eta\rvert \leq 0.8, \lvert\phi\rvert \leq 0.8)$		
Channels	p_T, $\mathcal{Q}_\kappa$	p_T	$\mathcal{Q}_\kappa$
Architecture	BN-32C6-MP2-128C4-MP2-256C6-MP2-512N-512N	BN-32C3-32C3-MP2-64C3-MP2-64C3-MP2-64C3-128C5-256C5-256N-256N	BN-32C3-32C3-MP2-64C4-64C4-MP2-256C6-MP2-256N
Settings	`Relu` Activation, `Padding=same`, *Dropout* $= 0.5$, l2 `Regularizer` $= 0.01$		
Preprocessing	*Centralization, Rotation, Flipping*		
Training	Adam Optimizer, `Minibatchsize`$=512$, *Cross entropy* loss		

activated to enable a deeper network

set to prevent overfitting

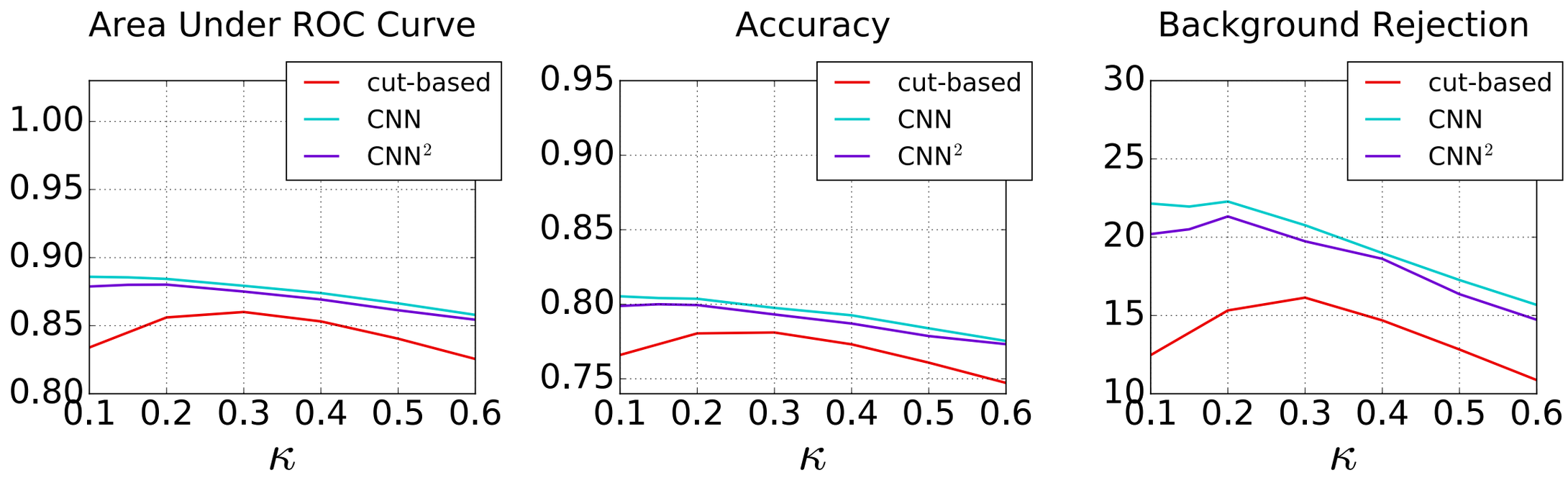
using Keras library with TensorFlow backend

PERFORMANCE OF OUR TAGGERS

- binary W^- vs W^+
- binary Z vs W^+
- ternary W^- / W^+ / Z

W⁻/W⁺ Classification

- Only charge Q_K distribution is useful.



- Slightly qualitatively different κ dependence for cut-based taggers, while similar between CNNs.
- CNN slightly better than CNN².
- CNNs have a smaller optimal κ.

W⁻/W⁺ Classification

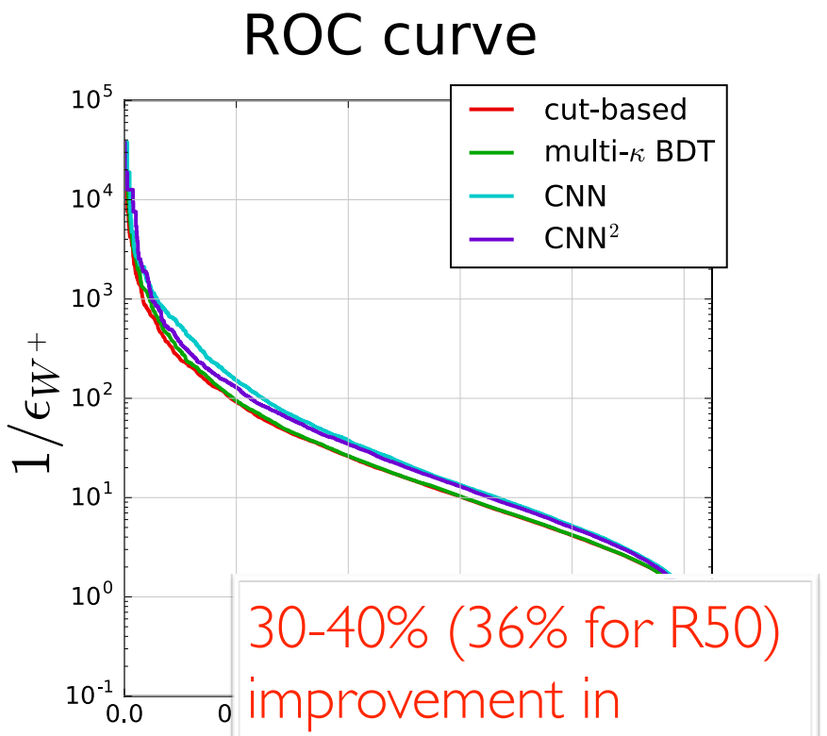
- Performance metrics for all taggers, except for the single-κ BDT, which is the same as the cut-based one.

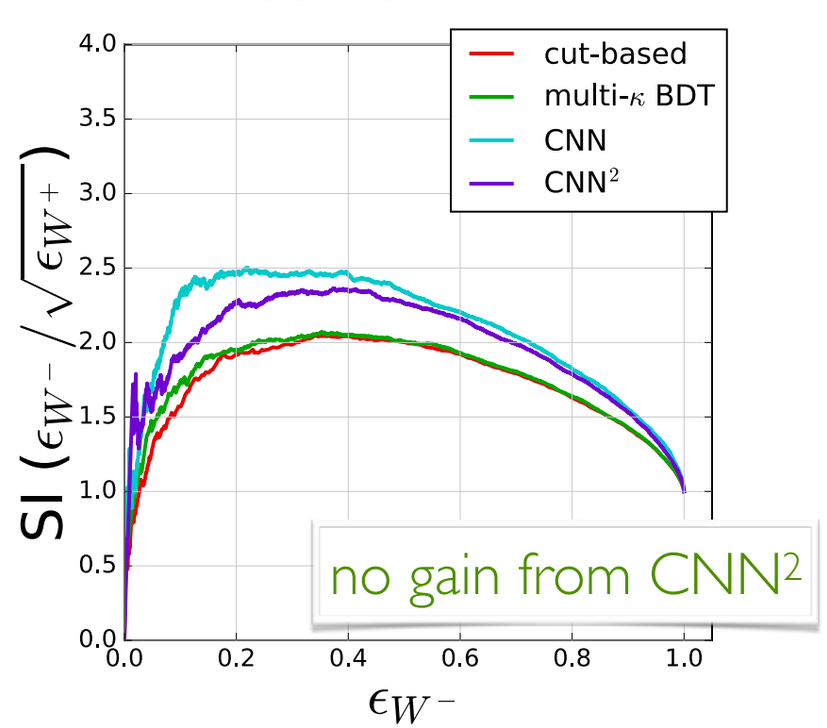
Area under ROC curve

	R50	AUC	ACC
cut-based	16.1372	0.8600	0.7811
multi-κ BDT	16.0960	0.8615	0.7820
CNN	21.9559	0.8855	0.8042
CNN^2	20.5057	0.8800	0.8000

background rejection rate at a 50% signal efficiency working point, $(1/\varepsilon_b)|_{\varepsilon_s=50\%}$.

best accuracy





Comparison

- Though differing in details, our performance gain from BDT to CNN is comparable to Fraser and Schwartz in their down/up quark jet discrimination (1-TeV benchmark).

tagger	AUC	mistag rate	ACC
BDT	0.8602	0.0633	0.7811
CNN	0.8855	0.0438	0.8042
CNN^2	0.8800	0.0497	0.8000

$$\frac{1/0.0438}{1/0.0633} \simeq 1.45$$

Network	1000 GeV Up Quark Efficiency	1000 GeV AUC
RecNN	0.049	0.876
CNN	0.048	0.879
RNN	0.054	0.874
Residual CNN	0.053	0.877
κ and λ BDT	0.068	0.859
Trainable κ NN	0.080	0.841
Jet Charge	0.090	0.832

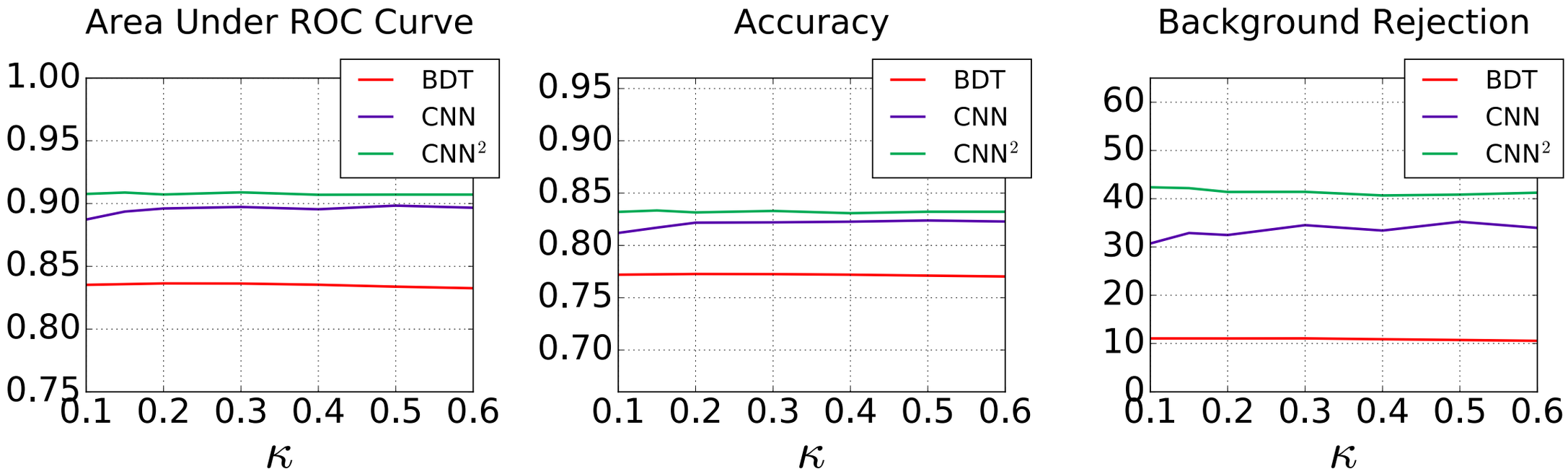
$$\frac{1/0.048}{1/0.068} \simeq 1.42$$

gain in background rejection rate

Fraser, Schwartz 2018

Z/W+ Classification

- Now the signal (Z) differs from the background (W^+) also in constituent p_T distribution.



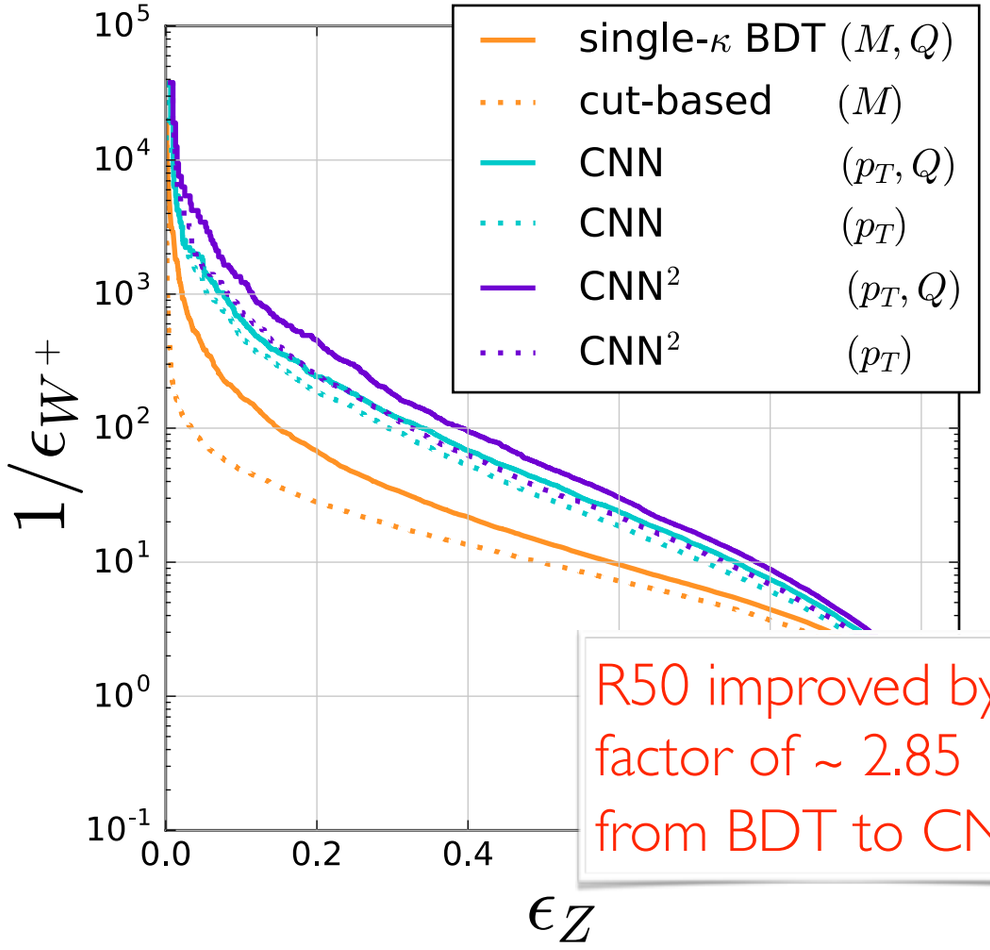
- Little κ dependence for all.
- CNN^2 slightly better than CNN.
- Use same optimal κ = 0.15 for consistency.

Z/W+ Classification

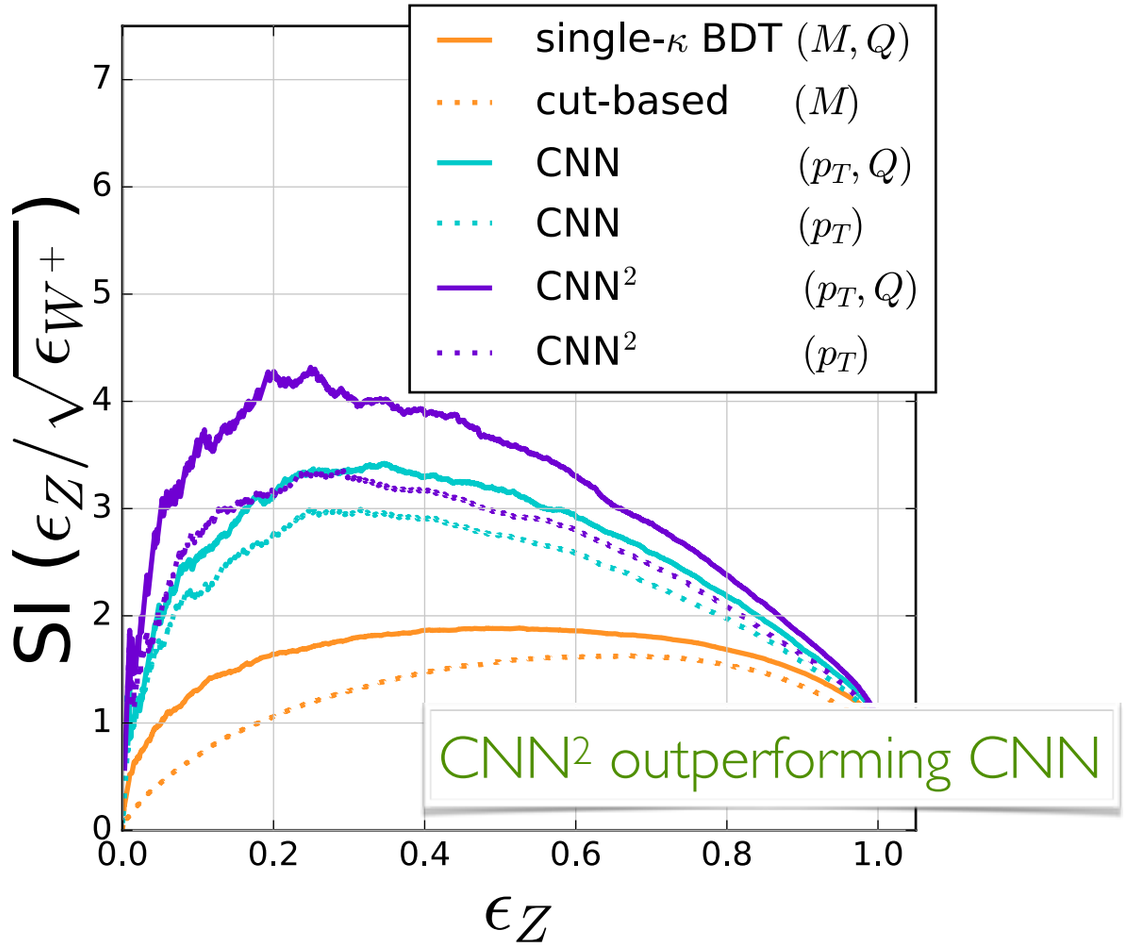
In a wide range of working points, our CNN taggers enjoy a ~30% gain in the background rejection rate by incorporating Q_κ.

	R50	AUC	ACC
cut-based	9.9590	0.8118	0.7705
single-κ BDT	14.1638	0.8608	0.7875
multi-κ BDT	14.2383	0.8611	0.7880
CNN	40.4205	0.9091	0.8345
CNN^2	52.6028	0.9206	0.8452

ROC curve



SIC curve



$W^+/W^-/Z$ Classification

- We summarize and compare the performance of the ternary taggers according to two metrics:
  (a) their overall accuracy

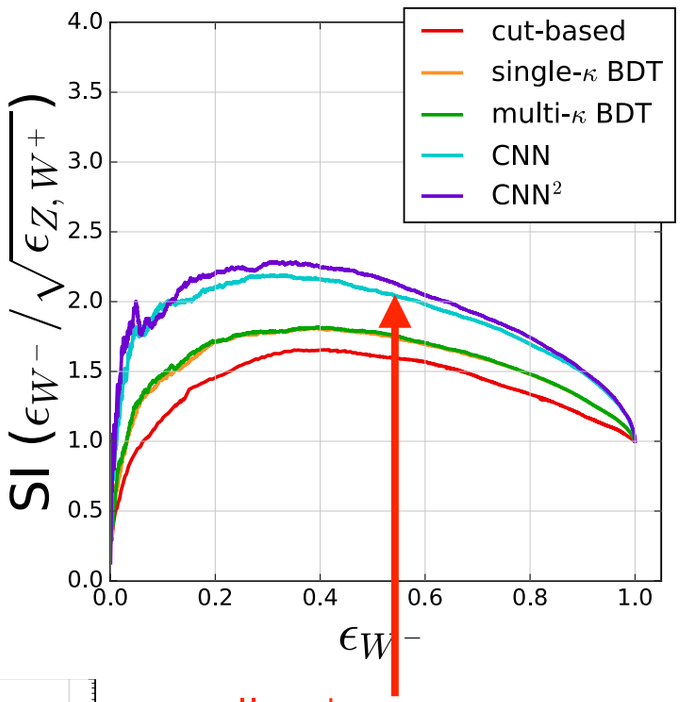
$$\frac{\text{number of correct predictions}}{\text{total number of instances}}$$

  and
  (b) a "one-against-all" metric
  one class as "signal" ⬌ all the rest as "background"

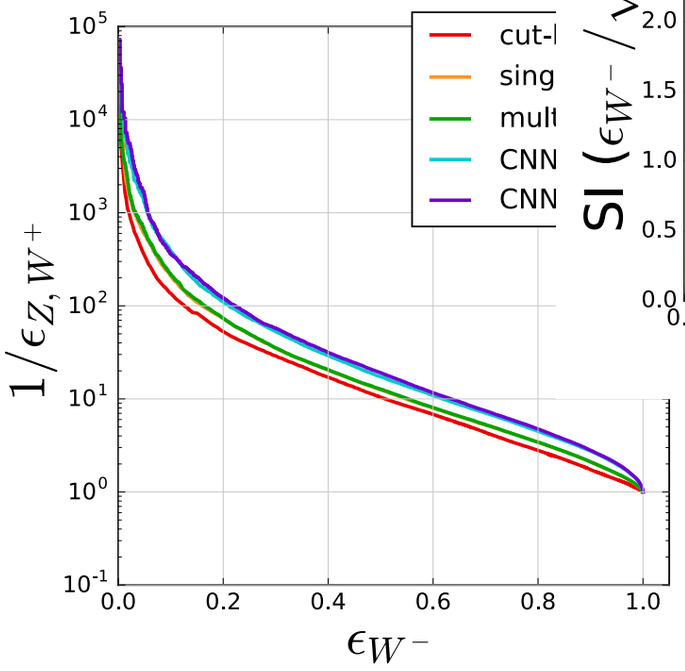
W^- or Z versus the Rest

W^- as signal

SIC curve



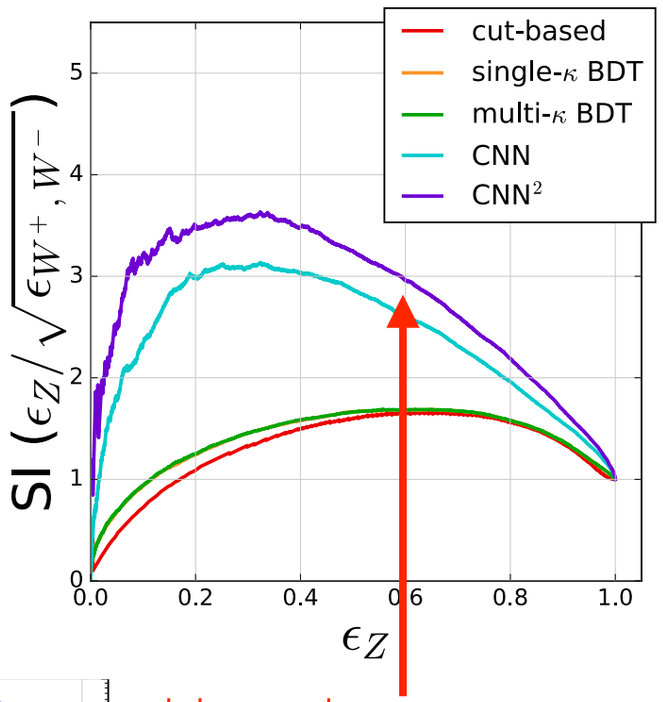
ROC curve



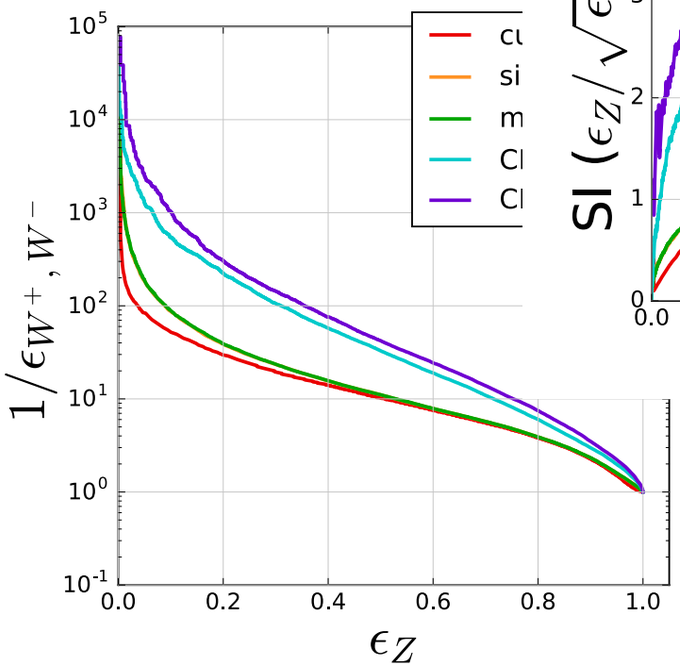
smaller improvement from CNN to CNN²

Z as signal

SIC curve



ROC curve



bigger improvement from CNN to CNN²

	overall	signal: W^-			signal: Z		
	ACC	R50	AUC	ACC	R50	AUC	ACC
cut-based	0.6581	8.0262	0.7893	0.7643	10.0882	0.8233	0.7839
single-κ BDT	0.6667	12.5230	0.8339	0.7576	11.0726	0.8363	0.7725
multi-κ BDT	0.6675	12.7115	0.8348	0.7579	11.0678	0.8366	0.7726
CNN	0.7197	17.3403	0.8715	0.7890	32.8981	0.8936	0.8170
CNN²	0.7318	19.0907	0.8764	0.7950	42.1927	0.9088	0.8334

From Ternary to Binary

- Our ternary taggers should be able to fully recover the binary taggers after an appropriate "projection."
- Suppose the ternary NN output class probability is denoted by $P_i(x)$, where x is a data point and i = 1, …, N is the class label, then the projection to binary classification between class i and class j is:

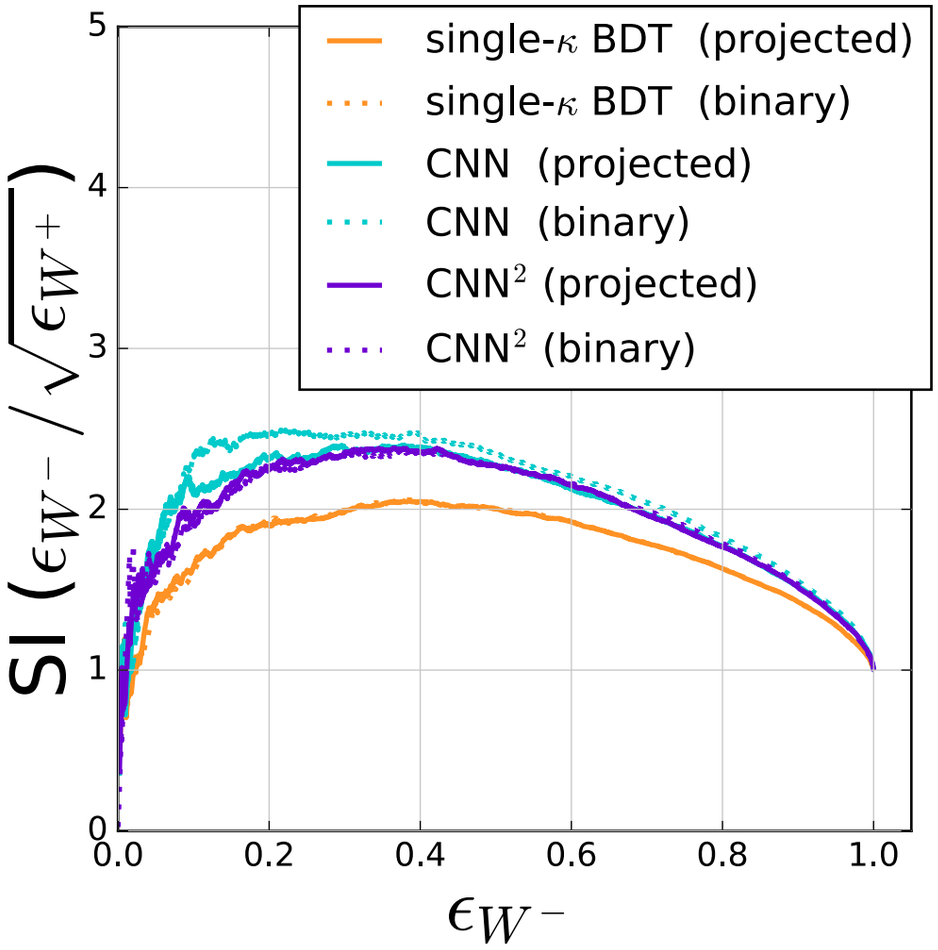
$$P_i^{i/j}(x) = \frac{P_i(x)}{P_i(x) + P_j(x)}$$

cf. Monty Hall problem

From Ternary to Binary

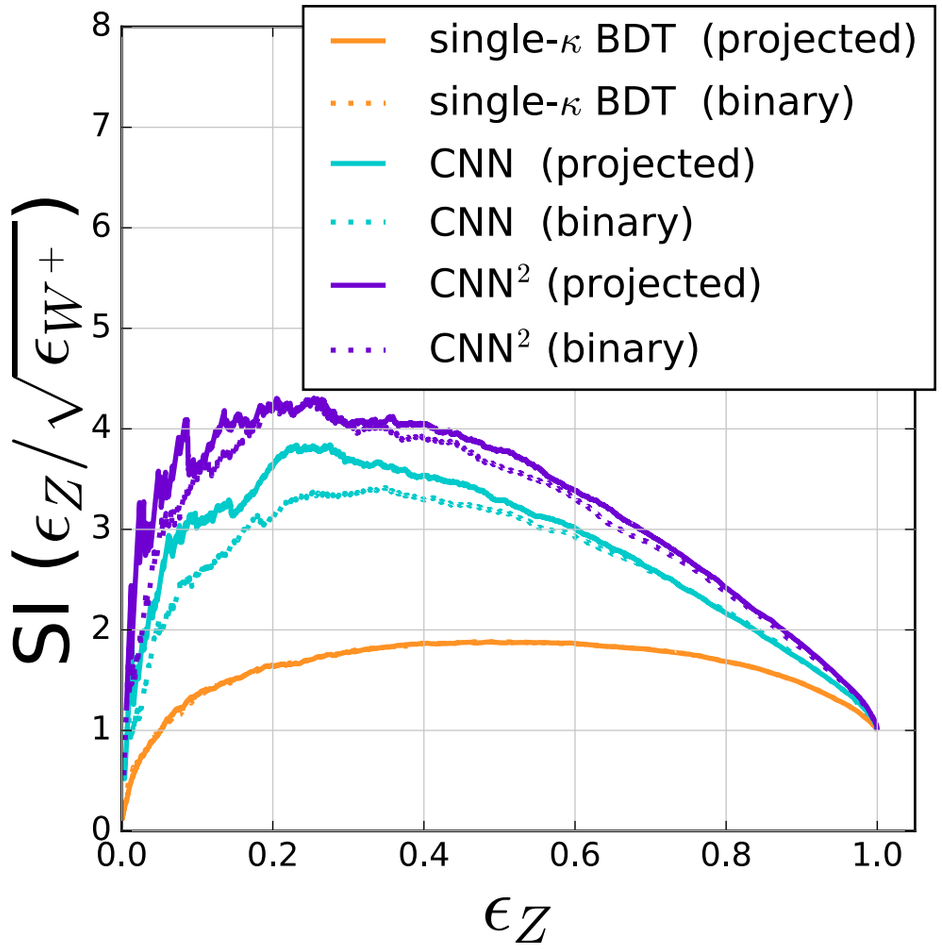
W^- vs W^+

SIC curve



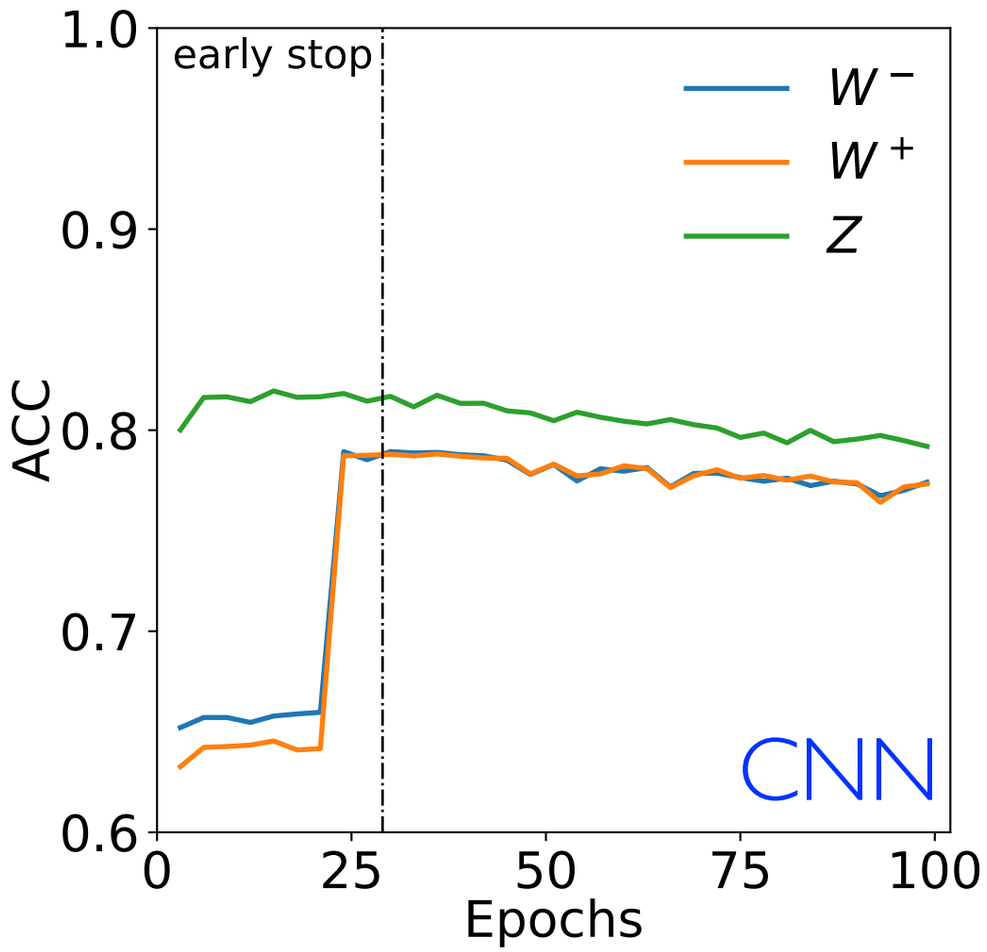
Z vs W^+

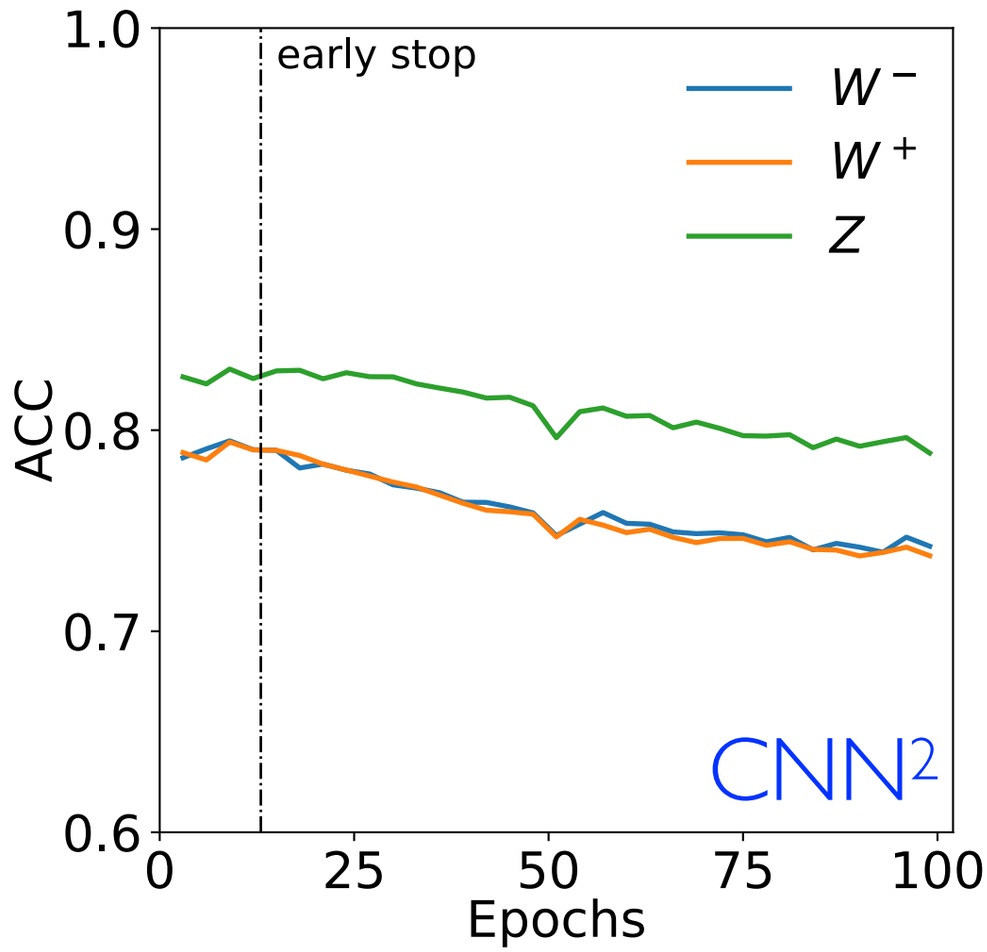
SIC curve



Phase Transition in DL

- A "phase transition" in the CNN architecture for $W^{\pm}$ samples around 25th epoch during training, but not CNN².





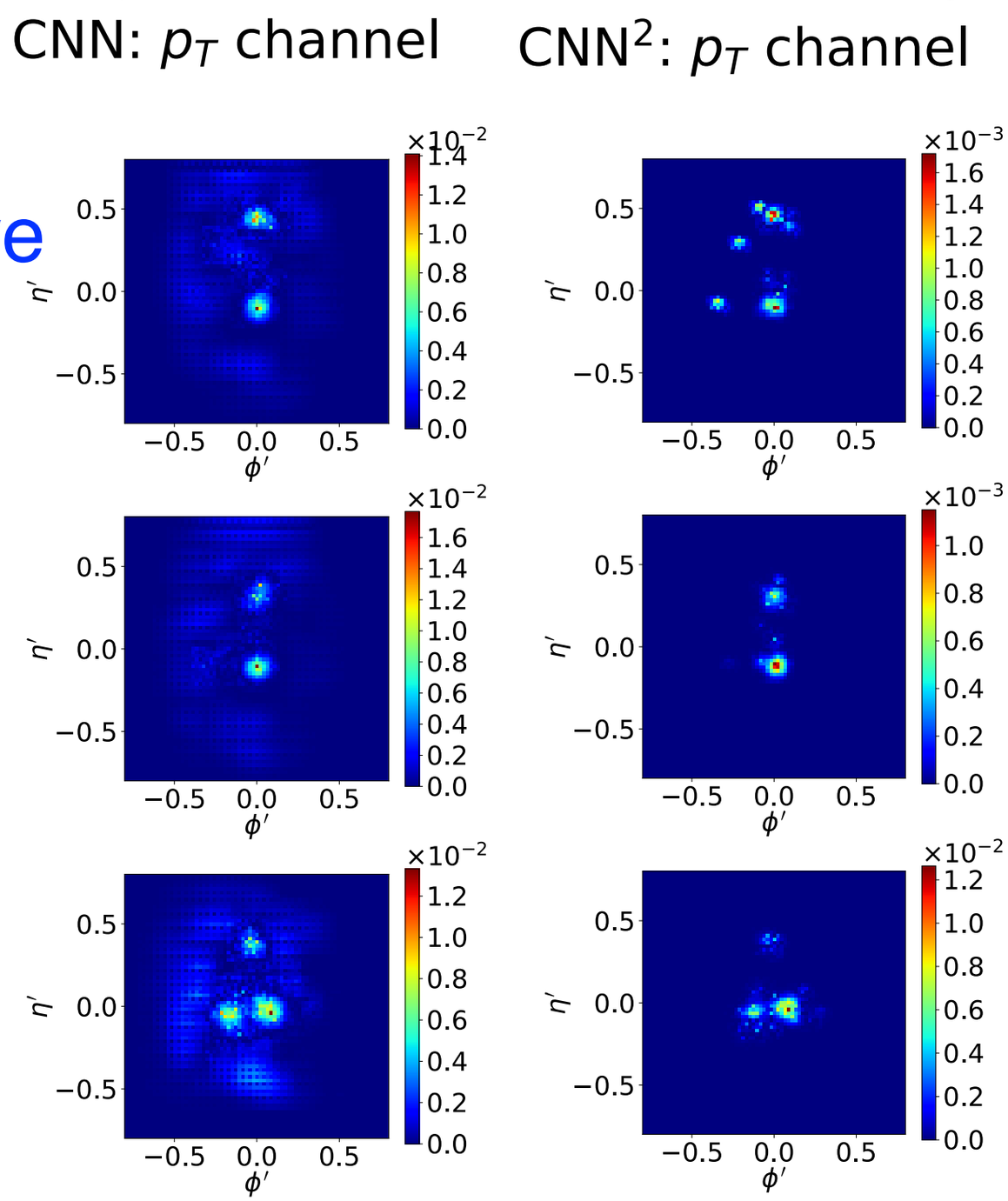
Saliency Maps

- The saliency map is a way to visualize how the machine learns, with the class saliency defined as pixel-wise derivative of the class probability $P_i(x)$:

$$w_i = \left.\frac{\partial P_i(x)}{\partial x}\right|_{x_0}$$

Simonyan, Vedaldi, Zisserman 2013

saliency maps for three W^- samples



Summary

- We apply modern deep learning techniques to build better taggers of boosted, hadronically-decaying W/Z bosons.
- Going beyond previous works, we incorporate jet charge information to discriminate between the charged W bosons, and between W and Z bosons.
- We construct binary and ternary CNN taggers, taking BDT and cut-based taggers for comparison, and see significant gains in classification accuracy and background rejection.
- We propose a novel/better composite CNN^2 architecture (better with Z classification), with different depths for p_T and Q_κ channels.
- Our taggers will enhance SM measurements and NP searches that are sensitive to electric charges of weak bosons.
- Improvement? Find a network that can mix info of p_T and Q_κ and learn optimal combination of them and fix κ.

Thank You!