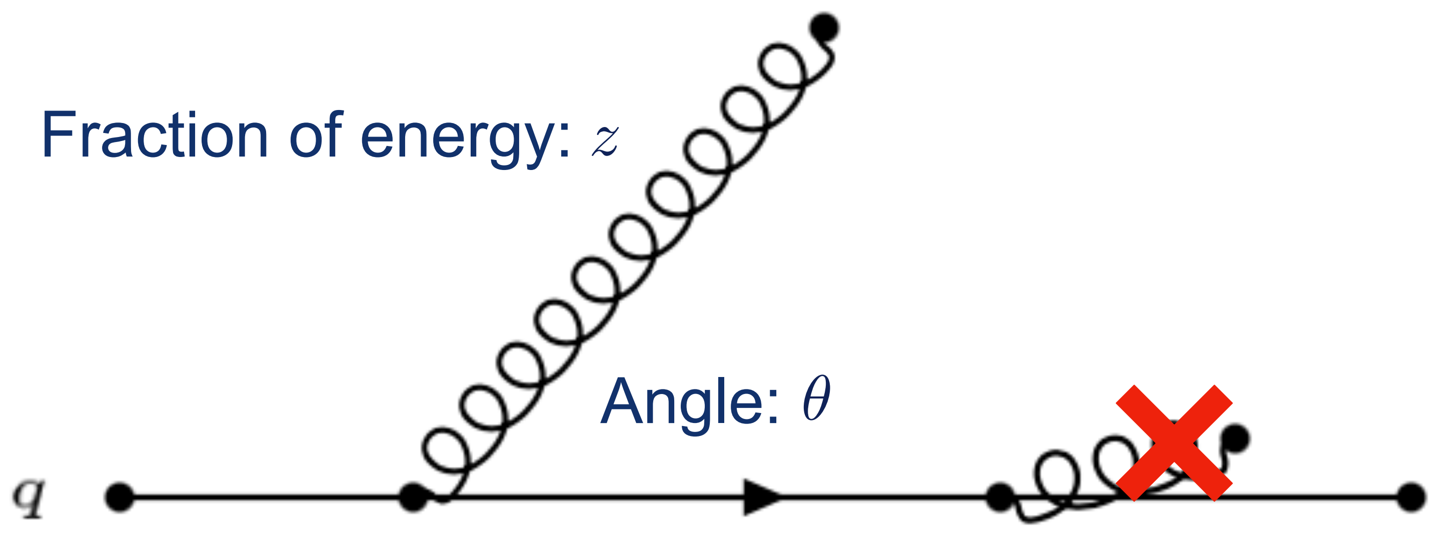
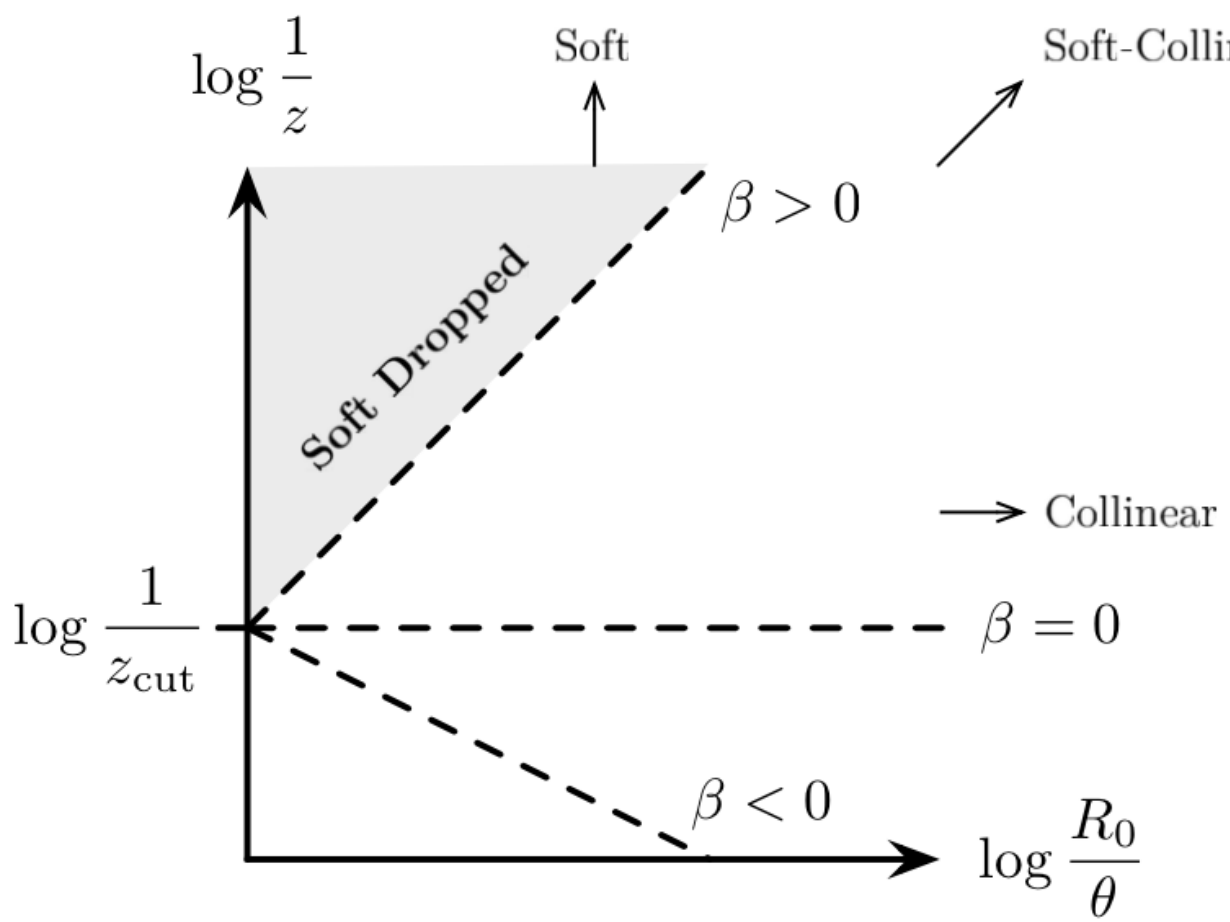
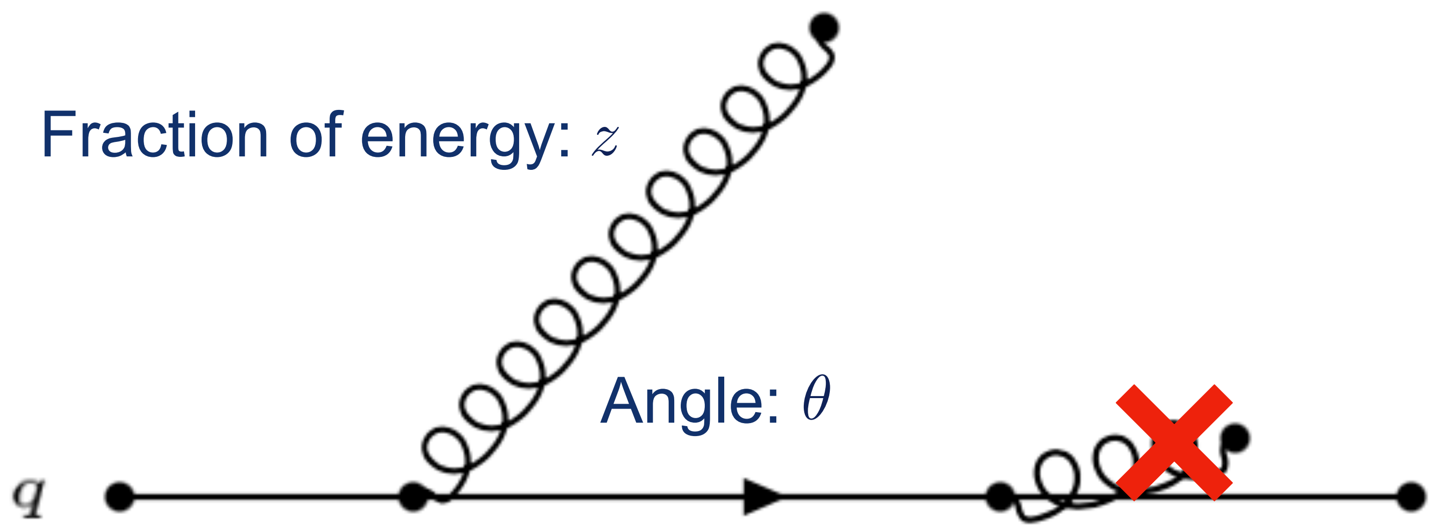
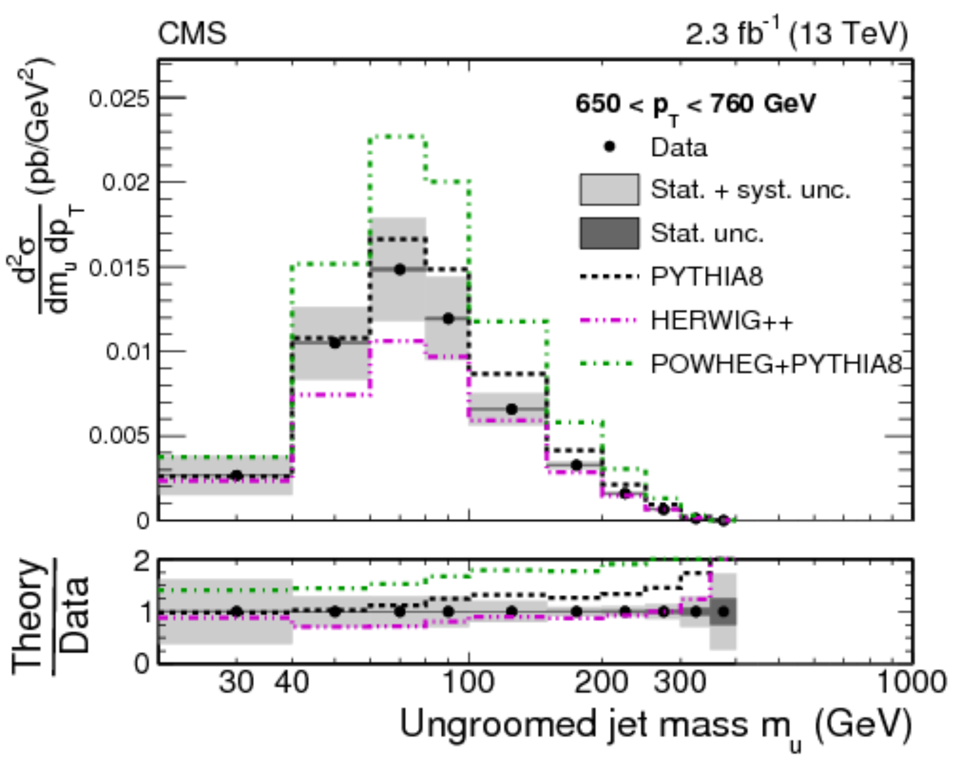
Grooming soft radiation

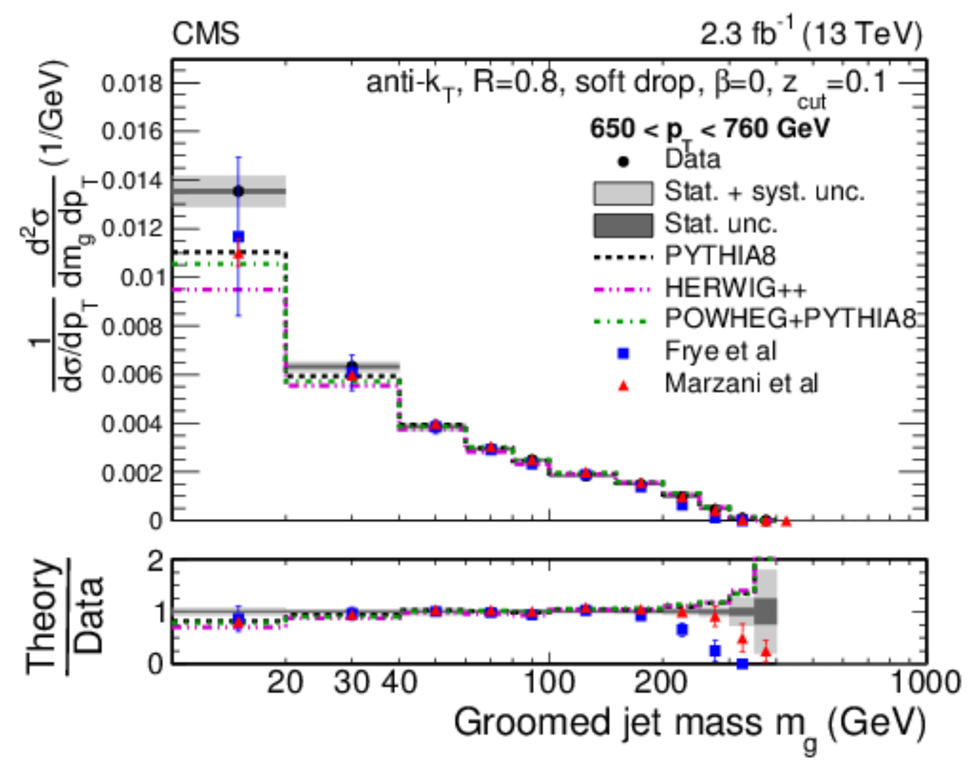




Grooming soft radiation







Larkoski, Marzani, Soyez, Thaler, JHEP 1309 (2013) 029,

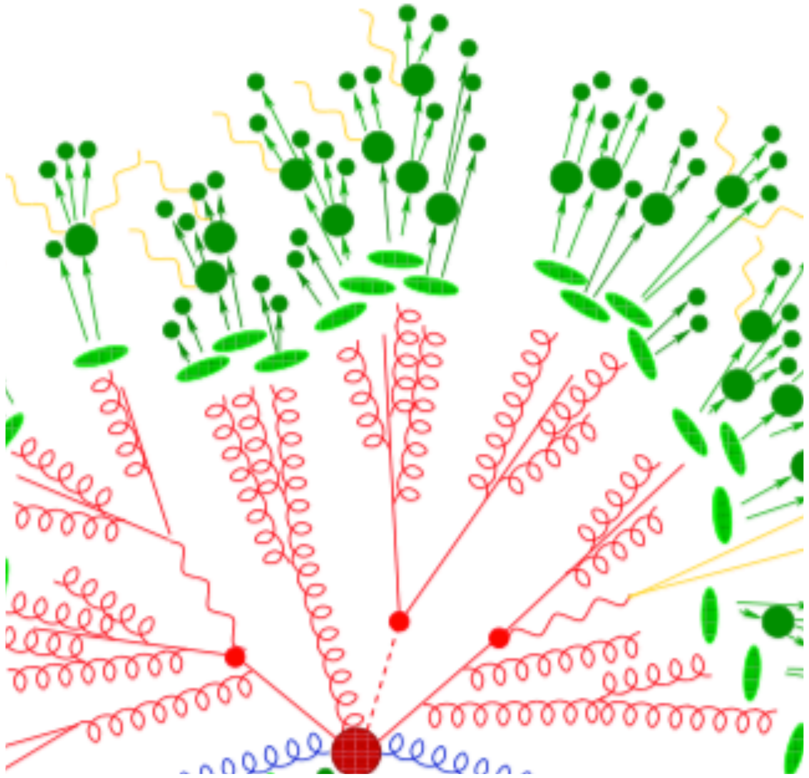
M. Dasgupta, A. Fregoso, S. Marzani, and G. P. Salam, JHEP 1309 (2013) 029,

CMS, *JHEP* 11 (2018) 113

Clustering

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$

$$d_{iB} = k_{ti}^{2p} \,,$$

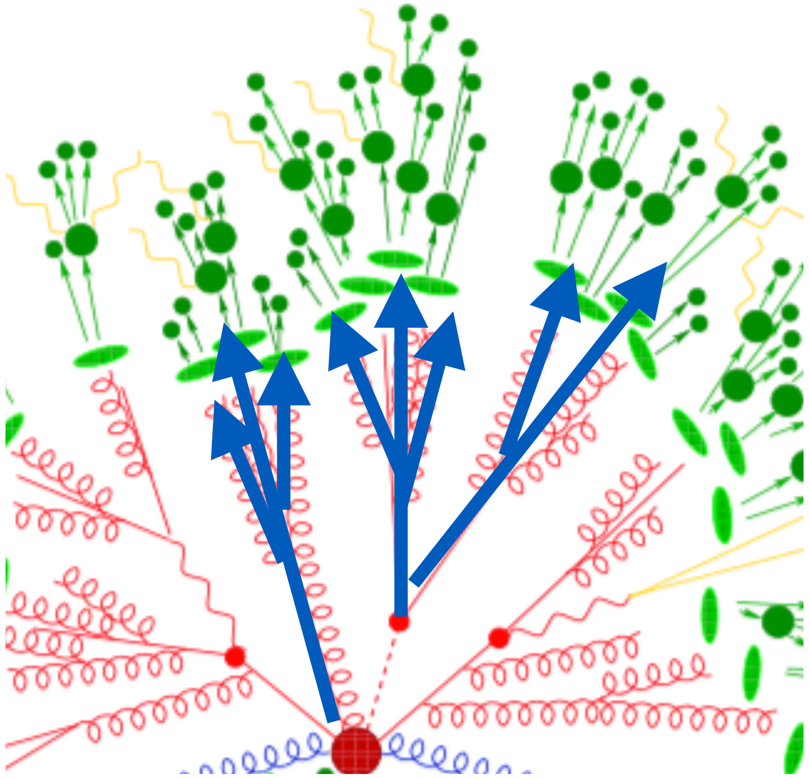


[hep-ph]
Cacciari, Salam, Soyez

Clustering

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$

$$d_{iB} = k_{ti}^{2p} ,$$



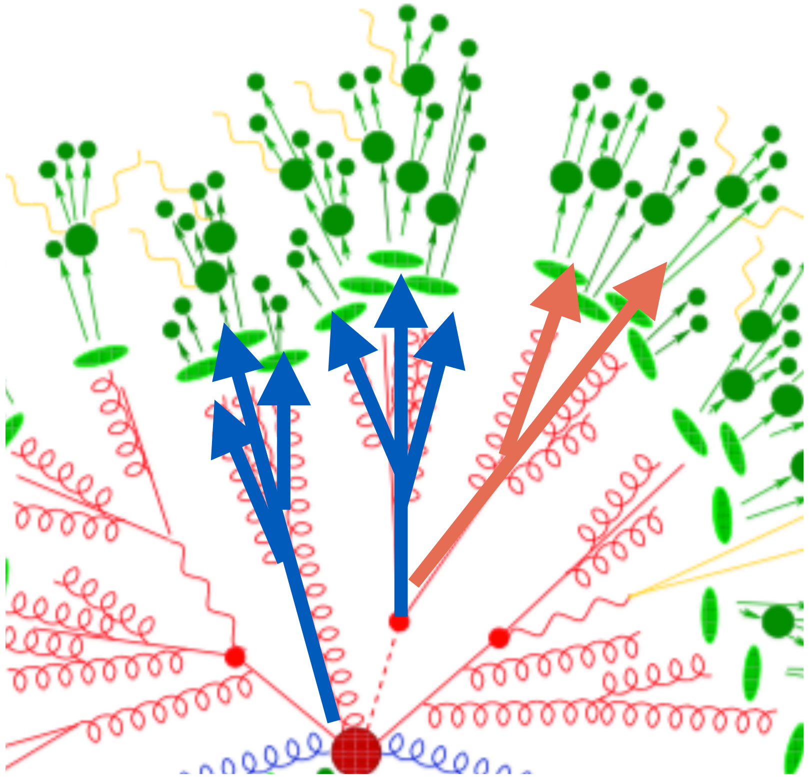
Start with full list of particles

[hep-ph]
Cacciari, Salam, Soyez

Clustering

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p})\frac{\Delta_{ij}^2}{R^2}$$

$$d_{iB} = k_{ti}^{2p}\,,$$



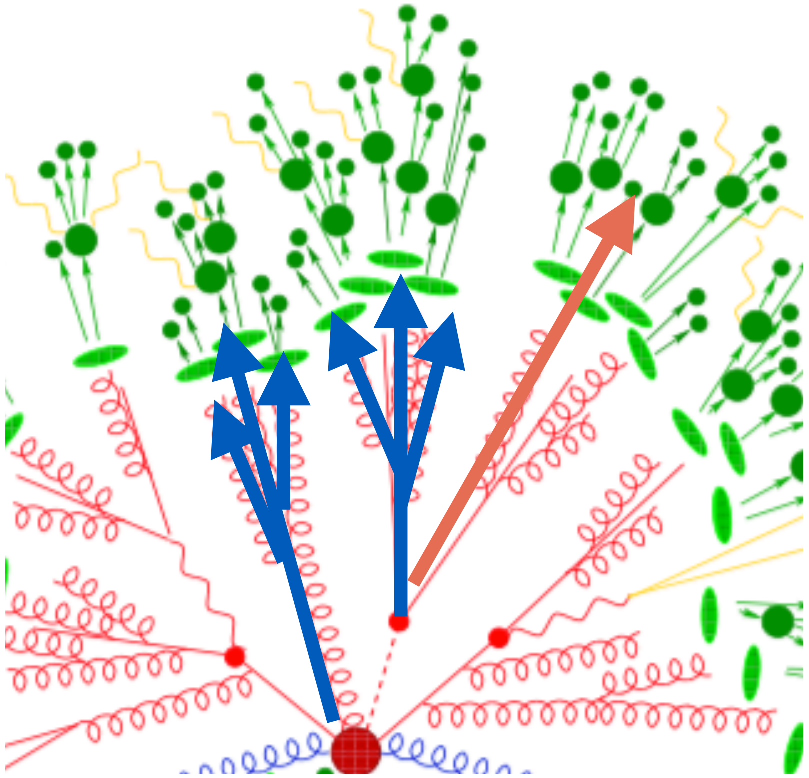
Iterate…

[hep-ph]
Cacciari, Salam, Soyez

Clustering

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p})\frac{\Delta_{ij}^2}{R^2}$$

$$d_{iB} = k_{ti}^{2p},$$

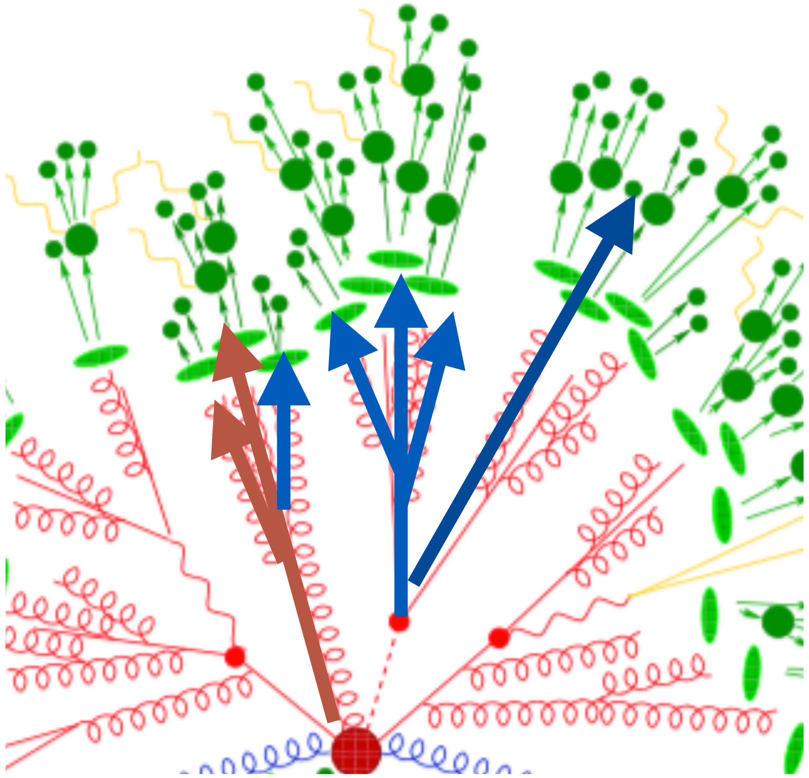


Iterate…

[hep-ph]
Cacciari, Salam, Soyez

Clustering

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$
$$d_{iB} = k_{ti}^{2p} \,,$$



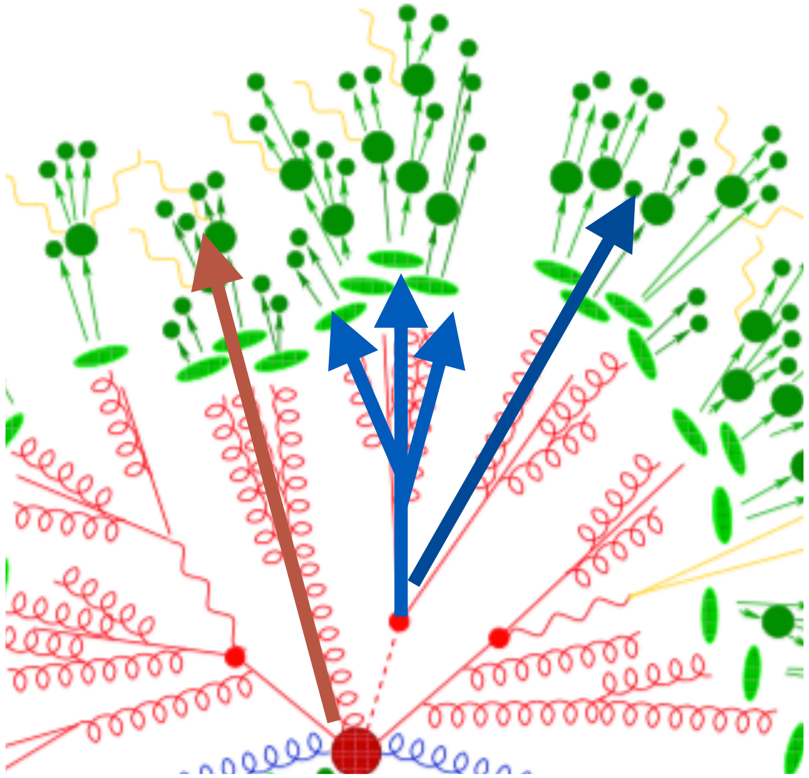
Iterate…

[hep-ph]
Cacciari, Salam, Soyez

Clustering

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$

$$d_{iB} = k_{ti}^{2p},$$



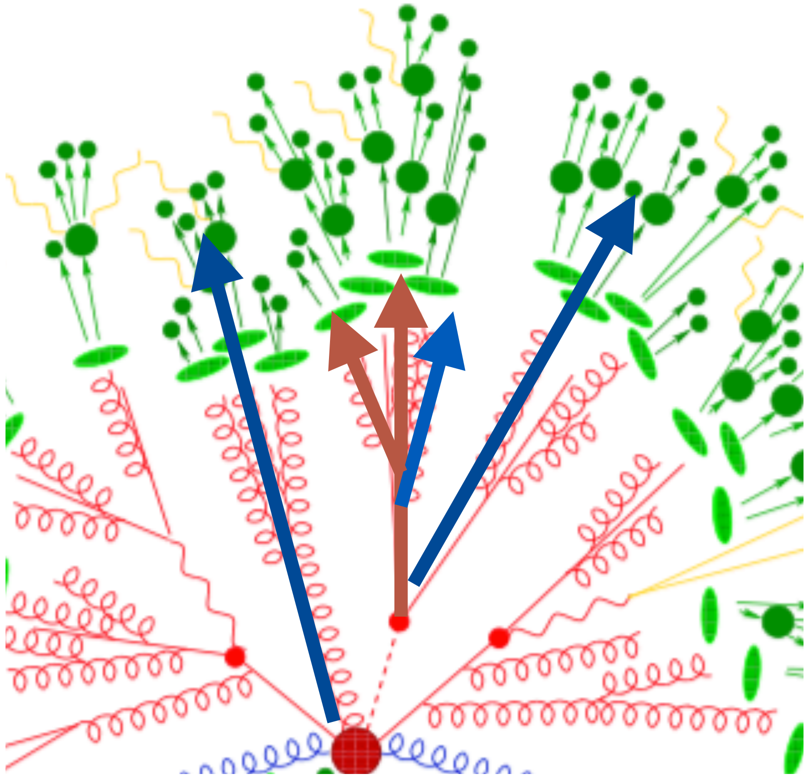
Iterate…

[hep-ph]
Cacciari, Salam, Soyez

Clustering

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$

$$d_{iB} = k_{ti}^{2p} \, ,$$



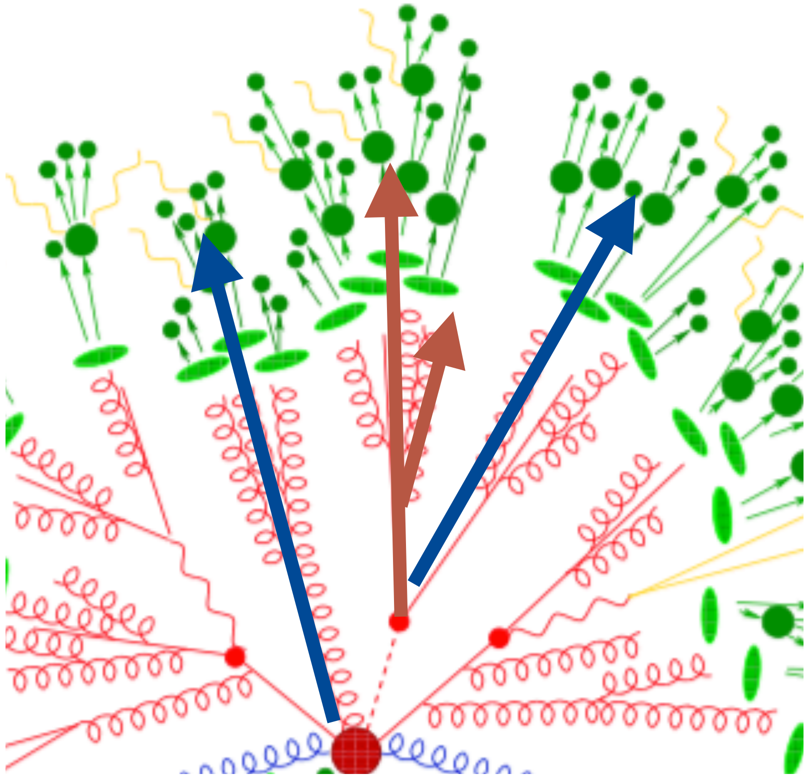
Iterate…

[hep-ph]
Cacciari, Salam, Soyez

Clustering

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p})\frac{\Delta_{ij}^2}{R^2}$$

$$d_{iB} = k_{ti}^{2p}\ ,$$



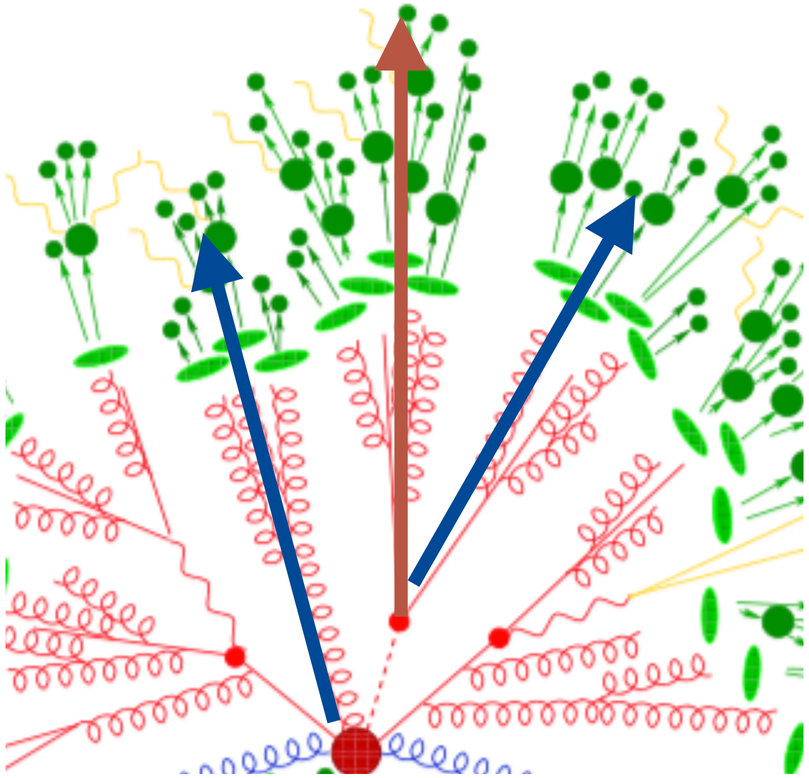
Iterate…

[hep-ph]
Cacciari, Salam, Soyez

Clustering

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$

$$d_{iB} = k_{ti}^{2p} \,,$$



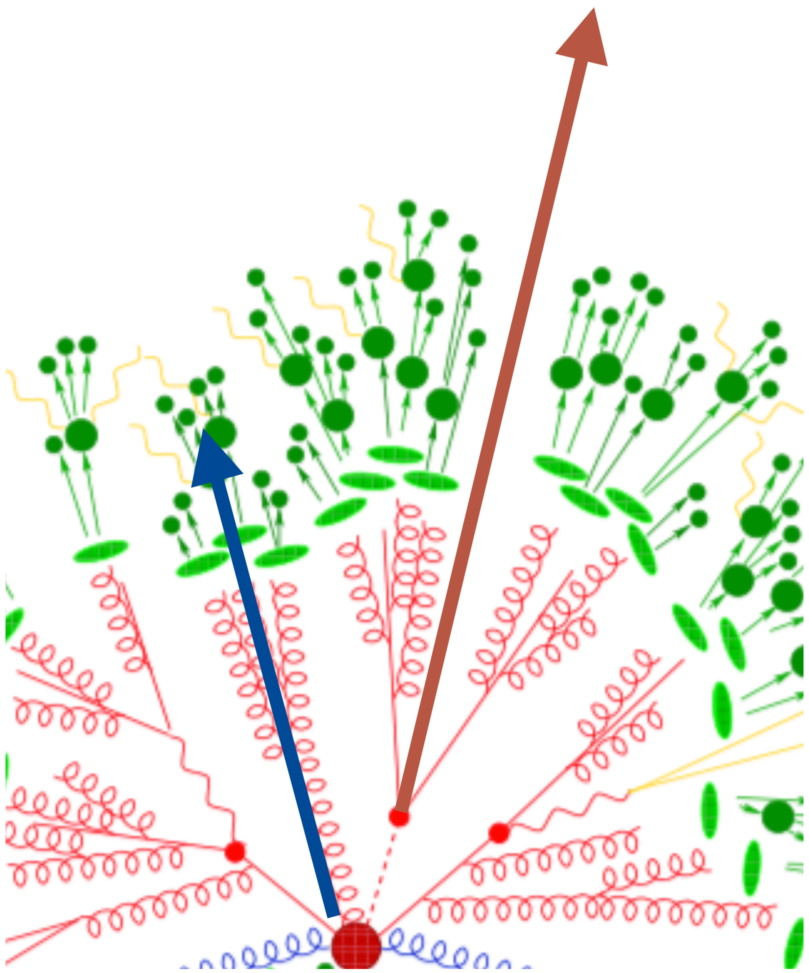
Iterate…

[hep-ph]
Cacciari, Salam, Soyez

Clustering

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$

$$d_{iB} = k_{ti}^{2p} \,,$$



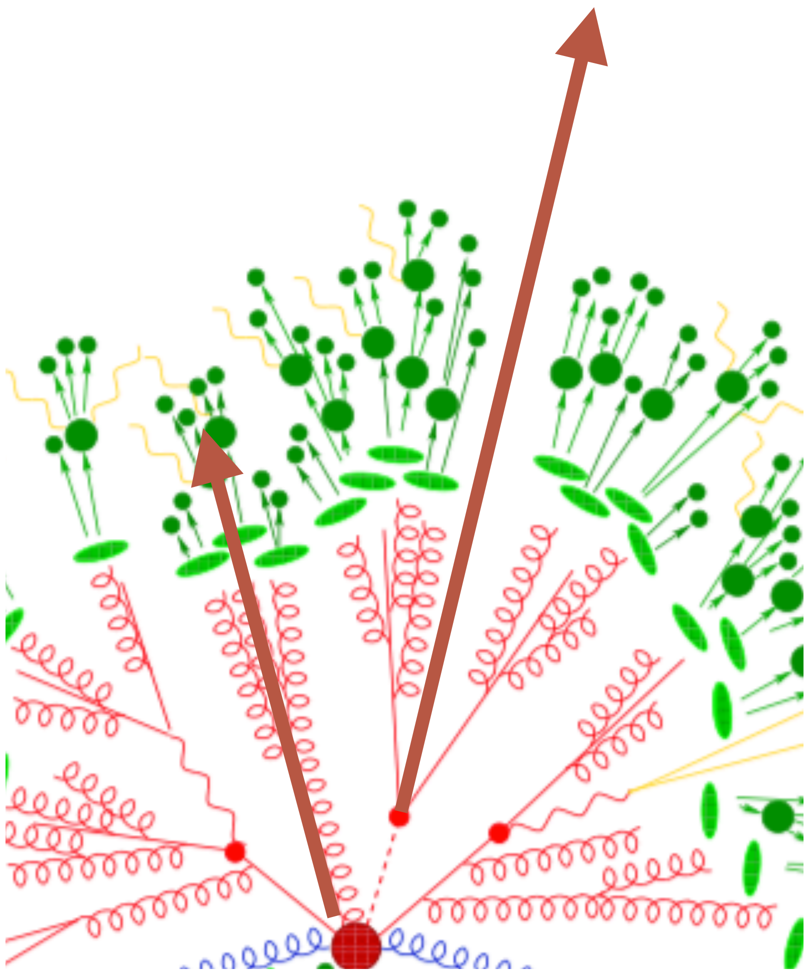
Iterate…

[hep-ph]
Cacciari, Salam, Soyez

Clustering

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$

$$d_{iB} = k_{ti}^{2p} \,,$$



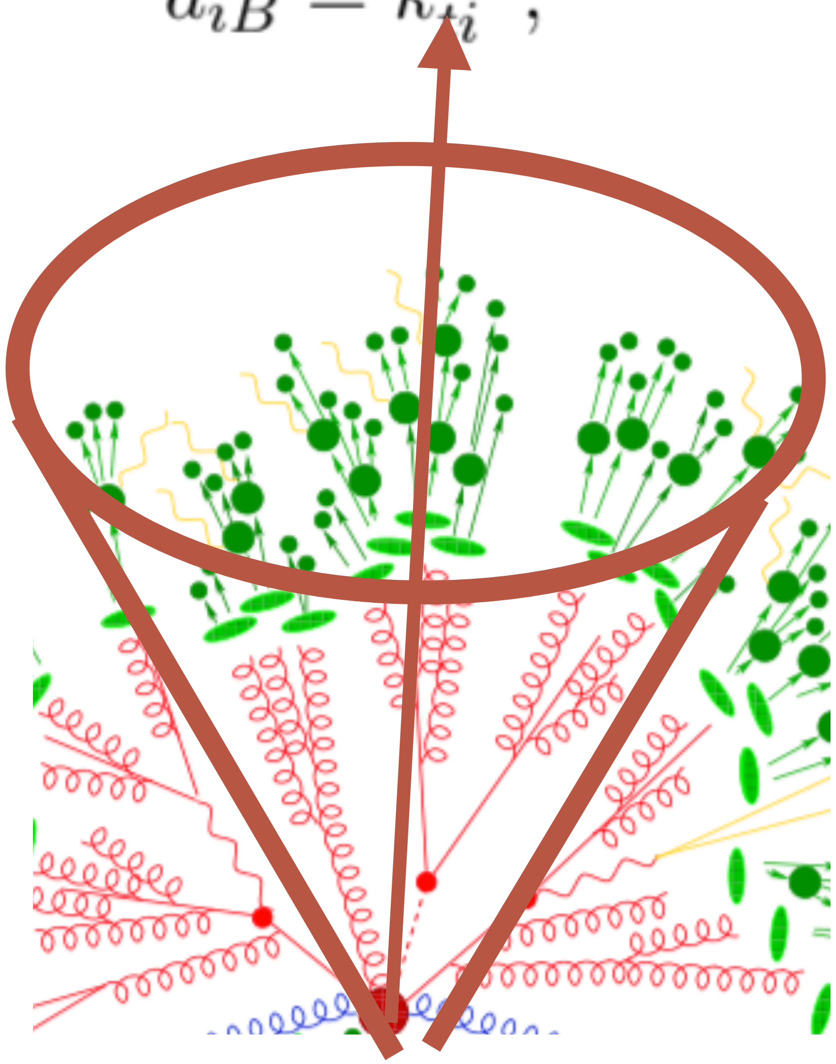
Iterate…

[hep-ph]
Cacciari, Salam, Soyez

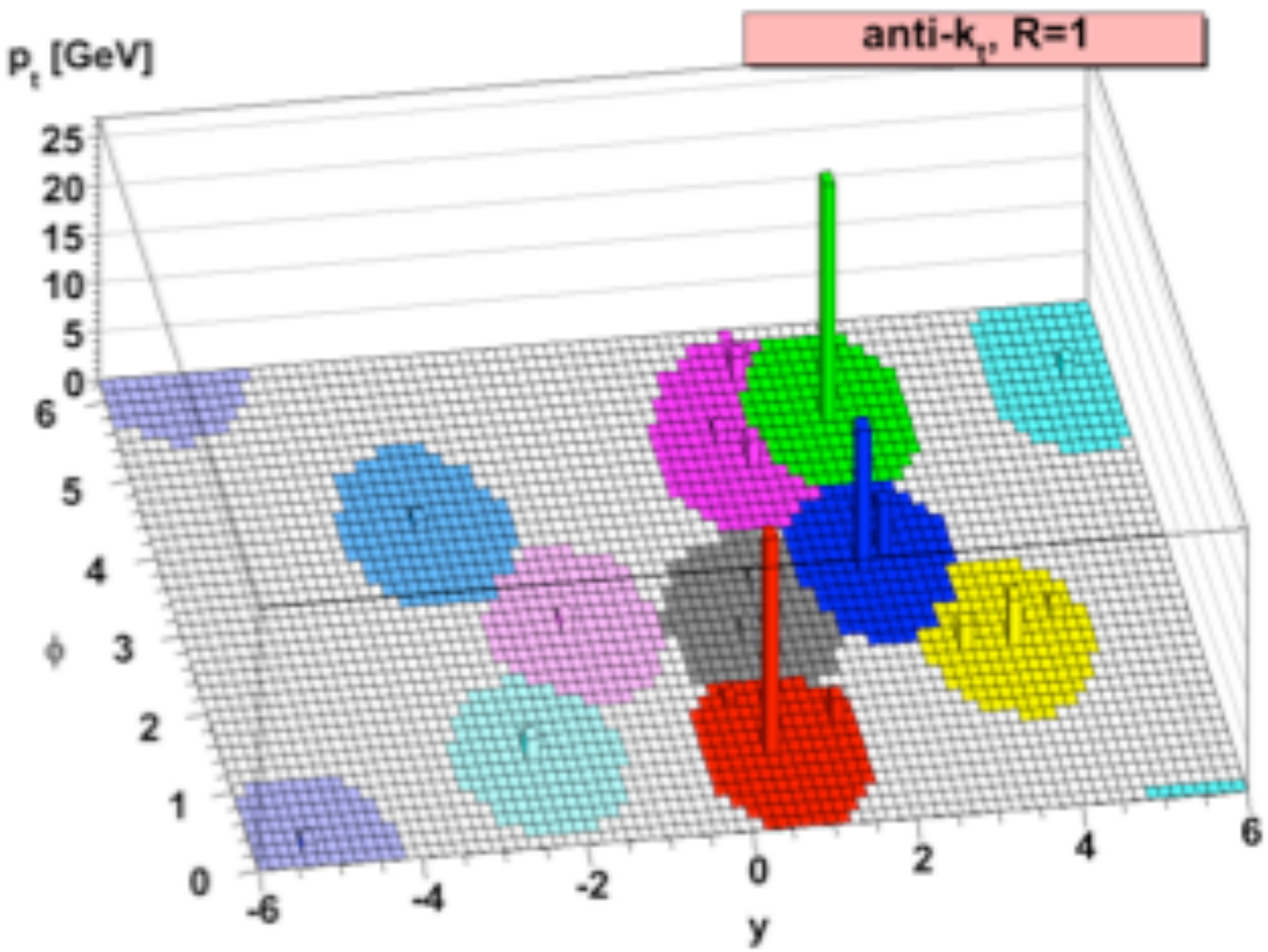
Clustering

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$

$$d_{iB} = k_{ti}^{2p} \,,$$



Final jet!



[hep-ph]
Cacciari, Salam, Soyez

Clustering

Any substructure?

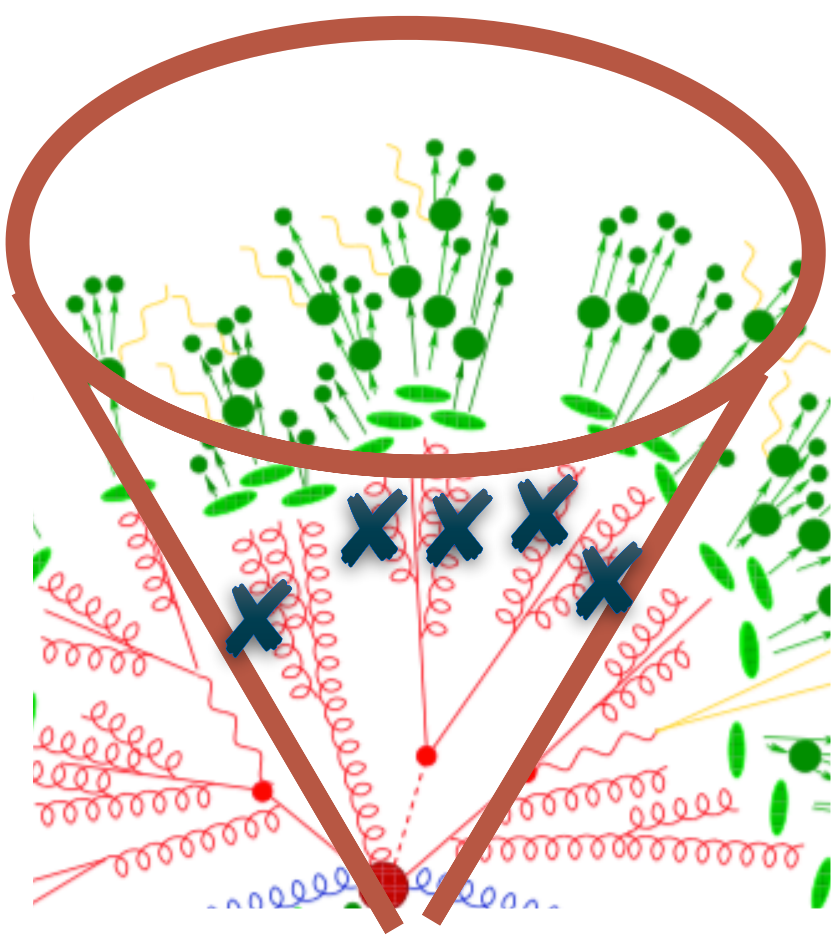
$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$

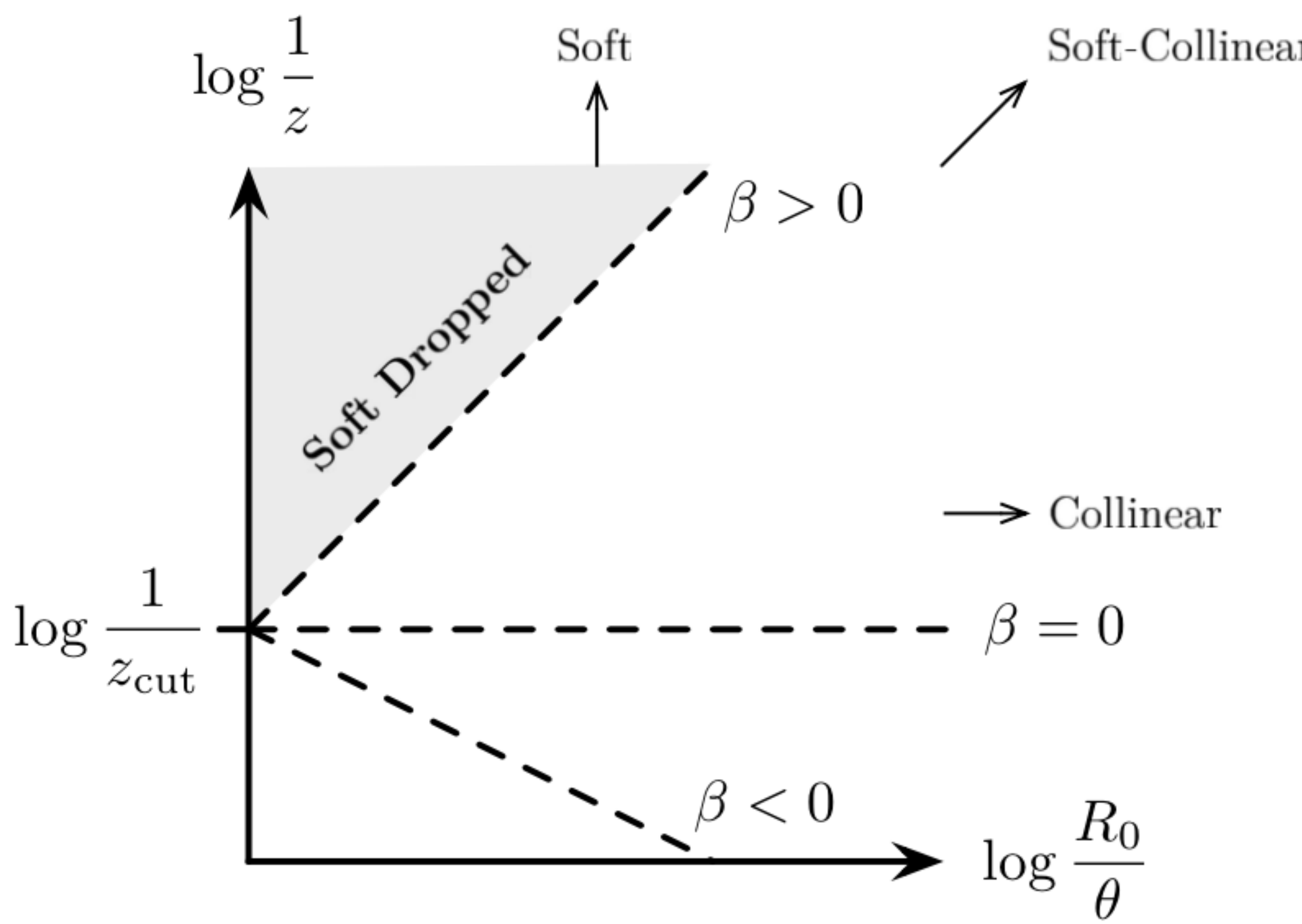
$$d_{iB} = k_{ti}^{2p} ,$$

Reversed!

+

$$\frac{\min(p_{T1}, p_{T2})}{p_{T1} + p_{T2}} > z_{cut} \left(\frac{\Delta R_{12}}{R} \right)^{\beta}$$





M. H. Seymour, Phys. C62 (1994) 127–138
J. M. Butterworth, A. R. Davison, M. Rubin and G. P. Salam, Phys. Rev. Lett. 100 (2008) 242001
D. E. Kaplan, K. Rehermann, M. D. Schwartz, and B. Tweedie, Phys.Rev.Lett. 101 (2008) 142001
S. D. Ellis, C. K. Vermilion and J. R. Walsh, Phys. Rev. D 81 (2010) 094023
S. D. Ellis, C. K. Vermilion and J. R. Walsh, Phys. Rev. D 80 (2009) 051501
D. Krohn, J. Thaler and L. -T. Wang, JHEP 1002 (2010) 084
Dasgupta, A. Fregoso, S. Marzani, and G. P. Salam, JHEP 1309 (2013) 029,
A. Larkoski, S. Marzani, G. Soyez, J. Thaler,JHEP 1405 (2014) 146

Clustering

Any substructure?

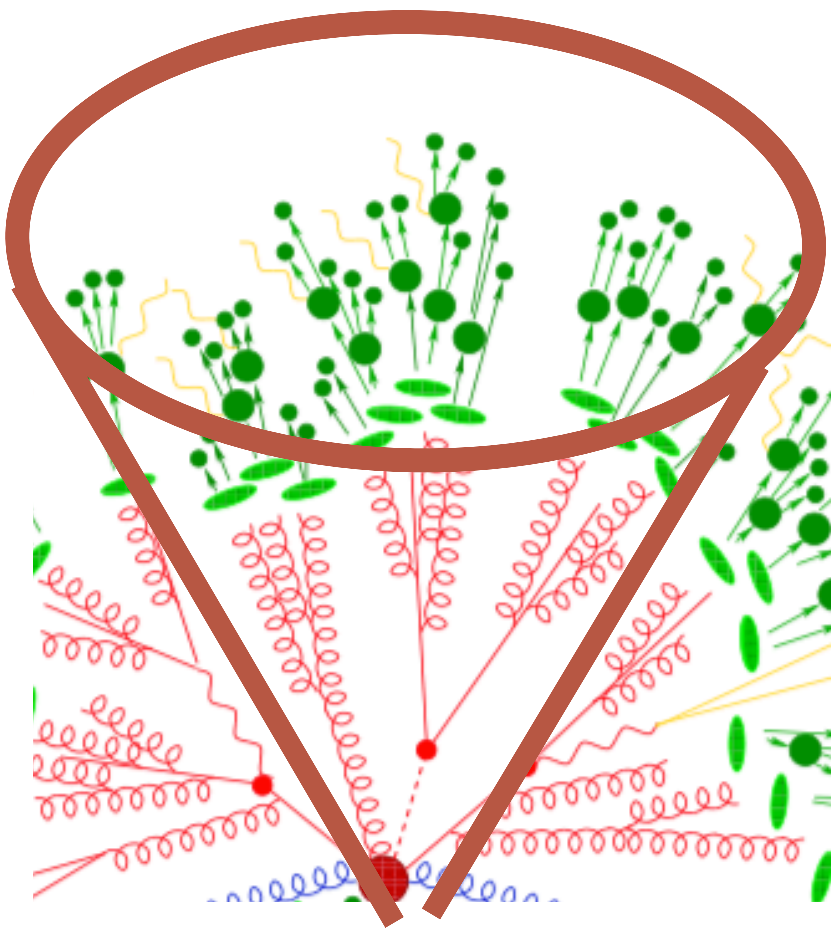
$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$

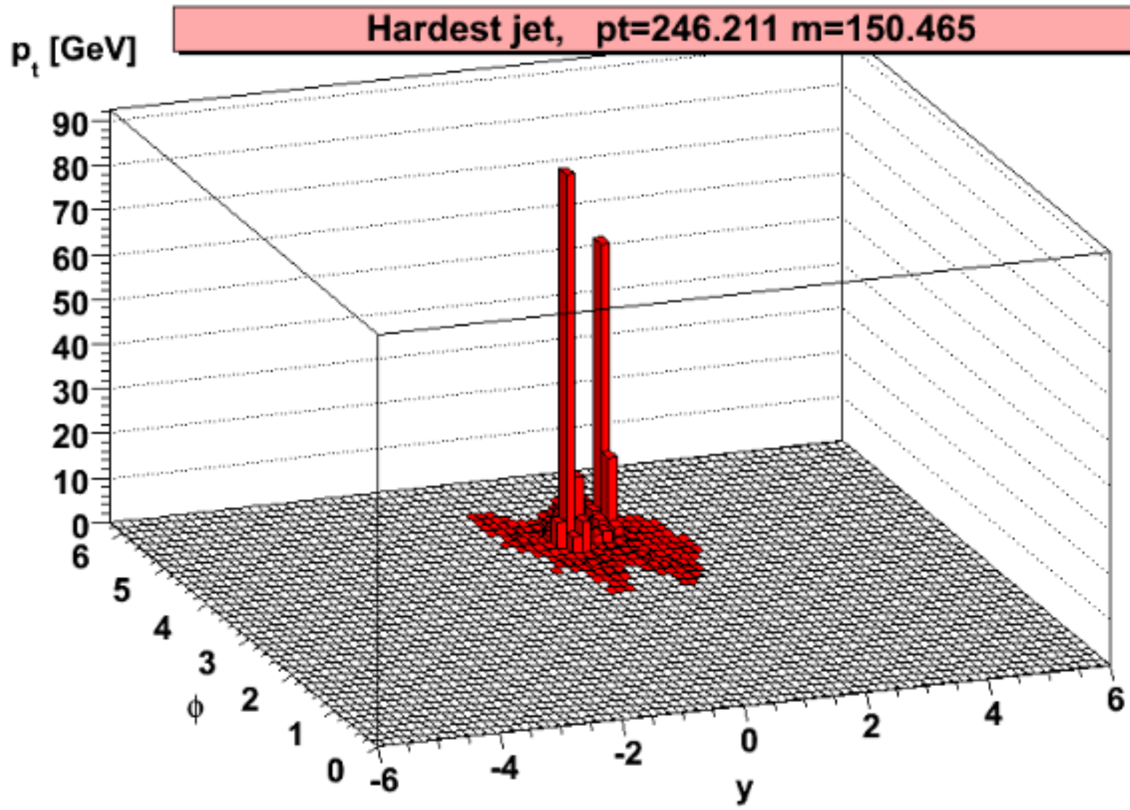
$$d_{iB} = k_{ti}^{2p} \,,$$

Reversed!

+

$$\frac{\min(p_{T1}, p_{T2})}{p_{T1} + p_{T2}} > z_{cut} \left(\frac{\Delta R_{12}}{R} \right)^{\beta}$$





Animation from https://gsalam.web.cern.ch/gsalam/talks/repo/2013-11-Oxford-substructure-NBI-jets.pdf

M. H. Seymour, Phys. C62 (1994) 127–138
J. M. Butterworth, A. R. Davison, M. Rubin and G. P. Salam, Phys. Rev. Lett. 100 (2008) 242001
D. E. Kaplan, K. Rehermann, M. D. Schwartz, and B. Tweedie, Phys.Rev.Lett. 101 (2008) 142001
S. D. Ellis, C. K. Vermilion and J. R. Walsh, Phys. Rev. D 81 (2010) 094023
S. D. Ellis, C. K. Vermilion and J. R. Walsh, Phys. Rev. D 80 (2009) 051501
D. Krohn, J. Thaler and L. -T. Wang, JHEP 1002 (2010) 084
Dasgupta, A. Fregoso, S. Marzani, and G. P. Salam, JHEP 1309 (2013) 029,
A. Larkoski, S. Marzani, G. Soyez, J. Thaler,JHEP 1405 (2014) 146

Clustering

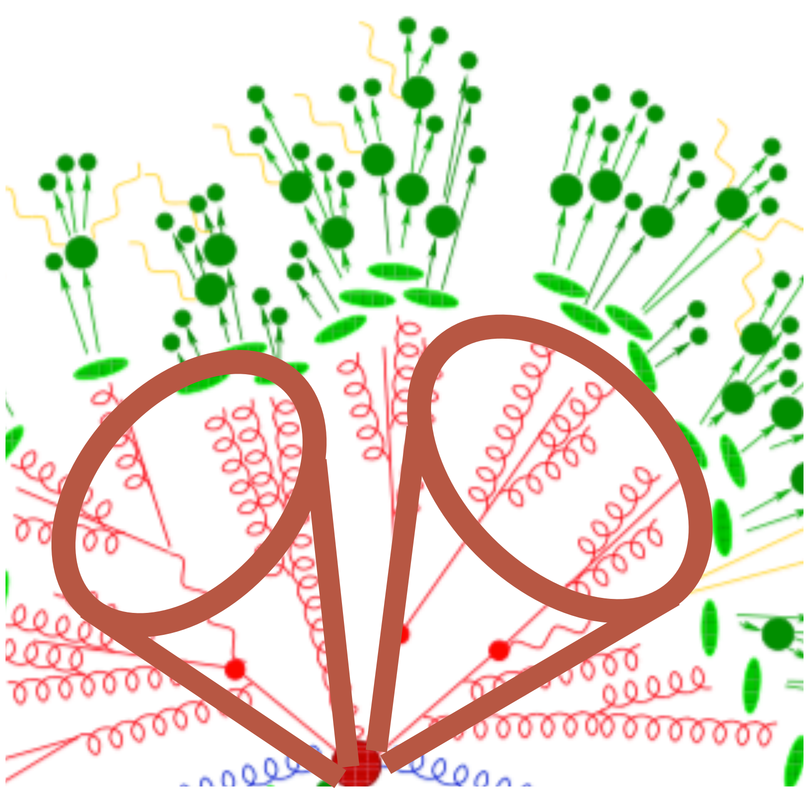
Check soft drop

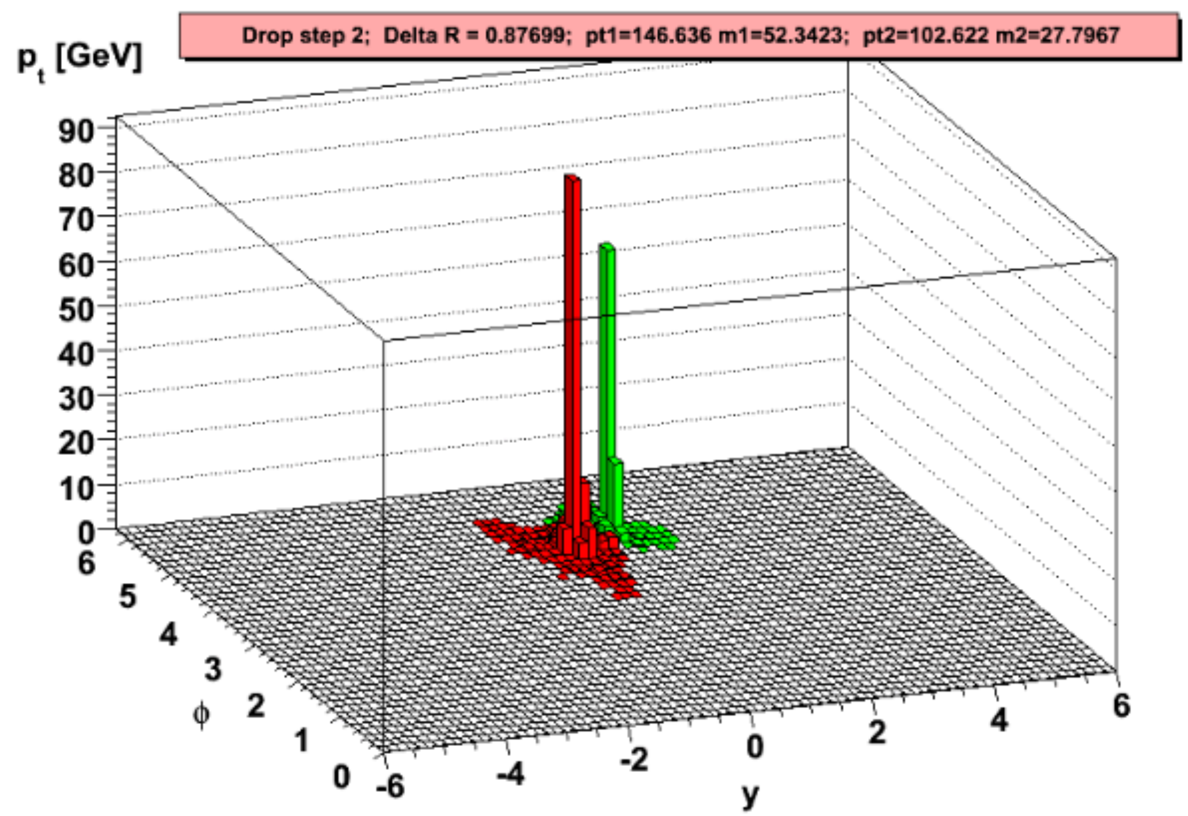
$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$

$$d_{iB} = k_{ti}^{2p},$$

Reversed!

$$+ \quad \frac{\min(p_{T1}, p_{T2})}{p_{T1} + p_{T2}} > z_{cut} \left(\frac{\Delta R_{12}}{R} \right)^{\beta}$$





M. H. Seymour, Phys. C62 (1994) 127–138
J. M. Butterworth, A. R. Davison, M. Rubin and G. P. Salam, Phys. Rev. Lett. 100 (2008) 242001
D. E. Kaplan, K. Rehermann, M. D. Schwartz, and B. Tweedie, Phys.Rev.Lett. 101 (2008) 142001
S. D. Ellis, C. K. Vermilion and J. R. Walsh, Phys. Rev. D 81 (2010) 094023
S. D. Ellis, C. K. Vermilion and J. R. Walsh, Phys. Rev. D 80 (2009) 051501
D. Krohn, J. Thaler and L. -T. Wang, JHEP 1002 (2010) 084
Dasgupta, A. Fregoso, S. Marzani, and G. P. Salam, JHEP 1309 (2013) 029,
A. Larkoski, S. Marzani, G. Soyez, J. Thaler,JHEP 1405 (2014) 146

Clustering

Final groomed jet

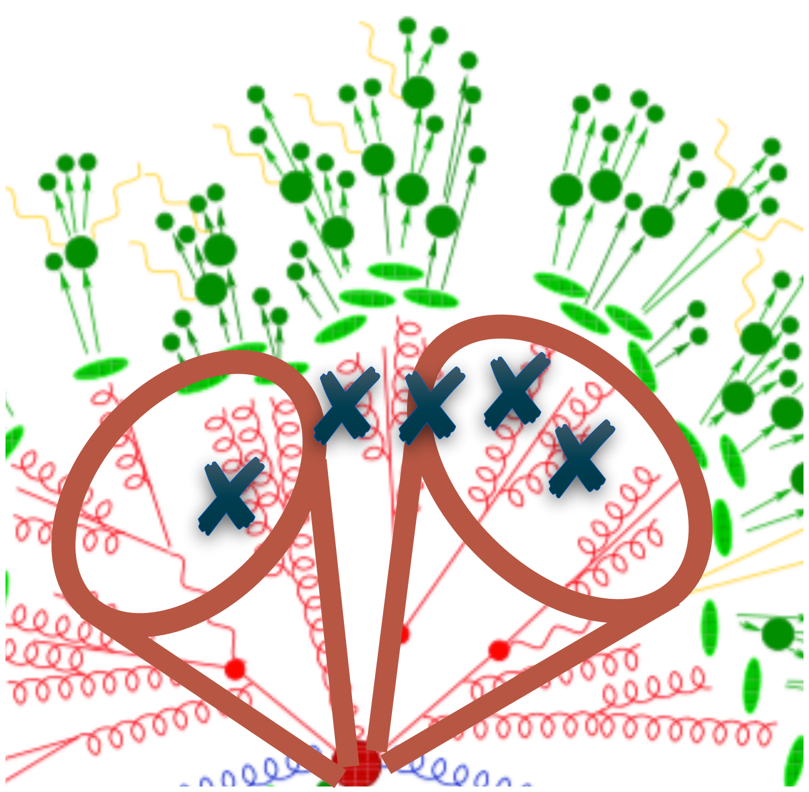
$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}$$

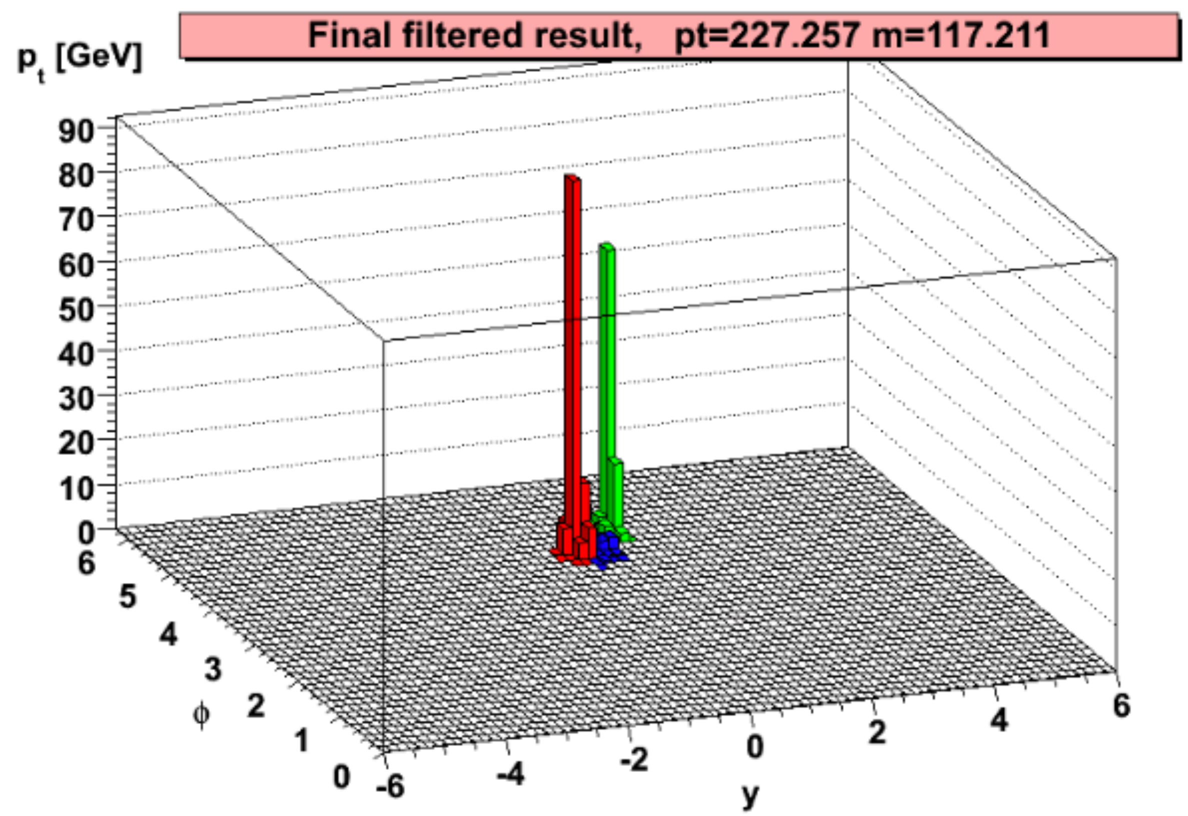
$$d_{iB} = k_{ti}^{2p},$$

Reversed!

+

$$\frac{\min(p_{T1}, p_{T2})}{p_{T1} + p_{T2}} > z_{cut} \left(\frac{\Delta R_{12}}{R} \right)^{\beta}$$



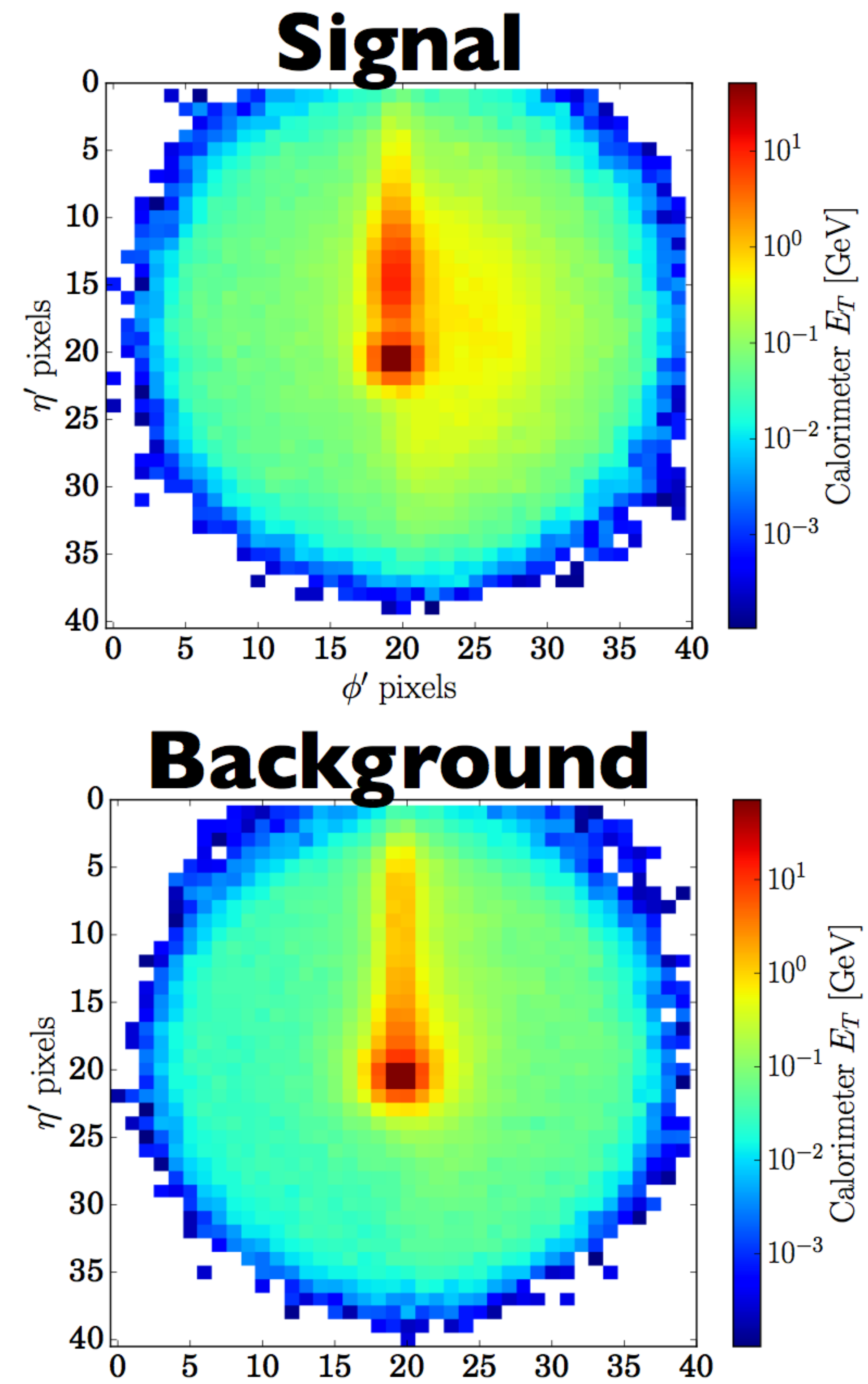


M. H. Seymour, Phys. C62 (1994) 127–138
J. M. Butterworth, A. R. Davison, M. Rubin and G. P. Salam, Phys. Rev. Lett. 100 (2008) 242001
D. E. Kaplan, K. Rehermann, M. D. Schwartz, and B. Tweedie, Phys.Rev.Lett. 101 (2008) 142001
S. D. Ellis, C. K. Vermilion and J. R. Walsh, Phys. Rev. D 81 (2010) 094023
S. D. Ellis, C. K. Vermilion and J. R. Walsh, Phys. Rev. D 80 (2009) 051501
D. Krohn, J. Thaler and L. -T. Wang, JHEP 1002 (2010) 084
Dasgupta, A. Fregoso, S. Marzani, and G. P. Salam, JHEP 1309 (2013) 029,
A. Larkoski, S. Marzani, G. Soyez, J. Thaler,JHEP 1405 (2014) 146

More advanced techniques... Machine Learning

- Plot energy of particles as a 2d projection
- Take average, put at center
- Take "next blob", put at top center
- Look at sum of many images

- Use image processing techniques!



Overlay of 10k images

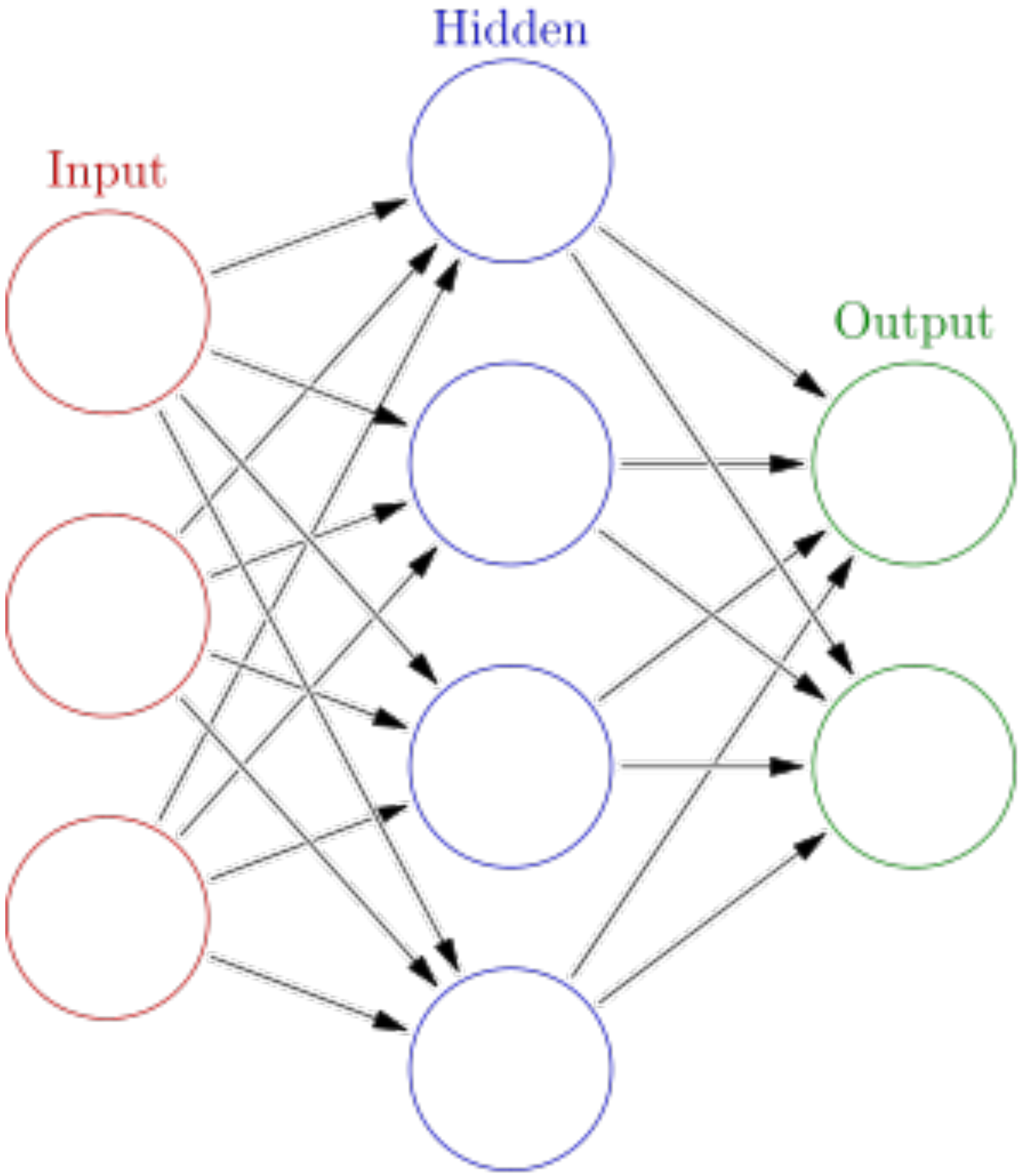
arXiv:1407.5675

JHEP 1607 069

JHEP 05 (2017) 006

Neural Networks

- The neural network is a group of neurons working together in such a persistent state:



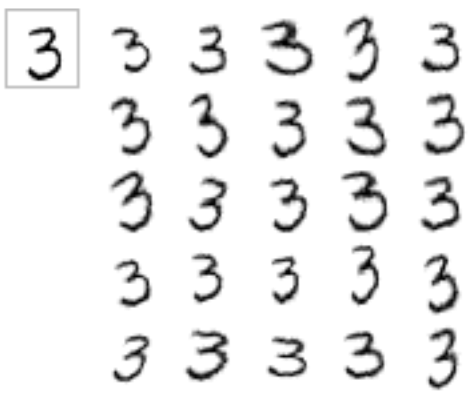
https://en.wikipedia.org/wiki/Artificial_neural_network

Neural Networks

- Training:
  - Feed a sample where you know the answer to the network
  - Tell the network the answer

Example to recognize handwriting:

"These are all the number 3":



- Discrimination:
  - Network takes inputs from UNKNOWN source
  - Outputs weights relative to known outcomes

"Is this a 3"?

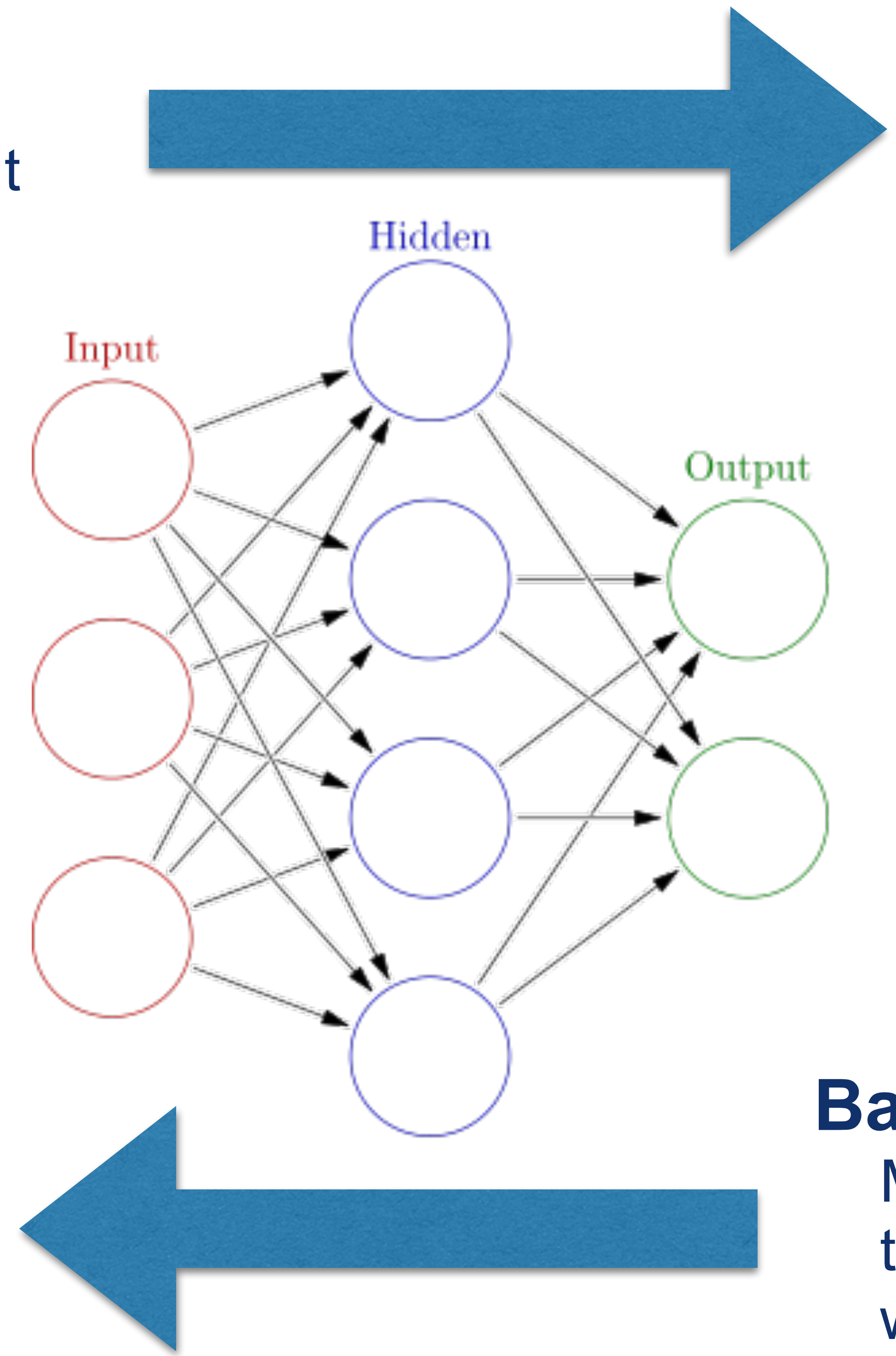
Outputs "yes"

Outputs "no"

https://www.codeproject.com/Articles/16650/Neural-Network-for-Recognition-of-Handwritten-Digi

Neural Networks

Feed forward:
Weight inputs, get output

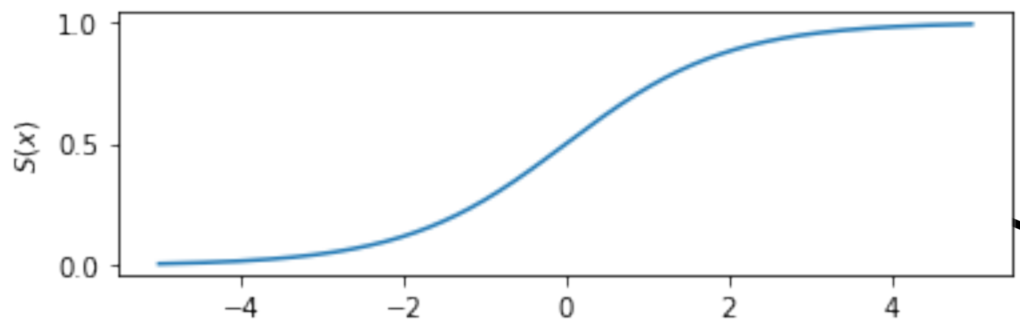


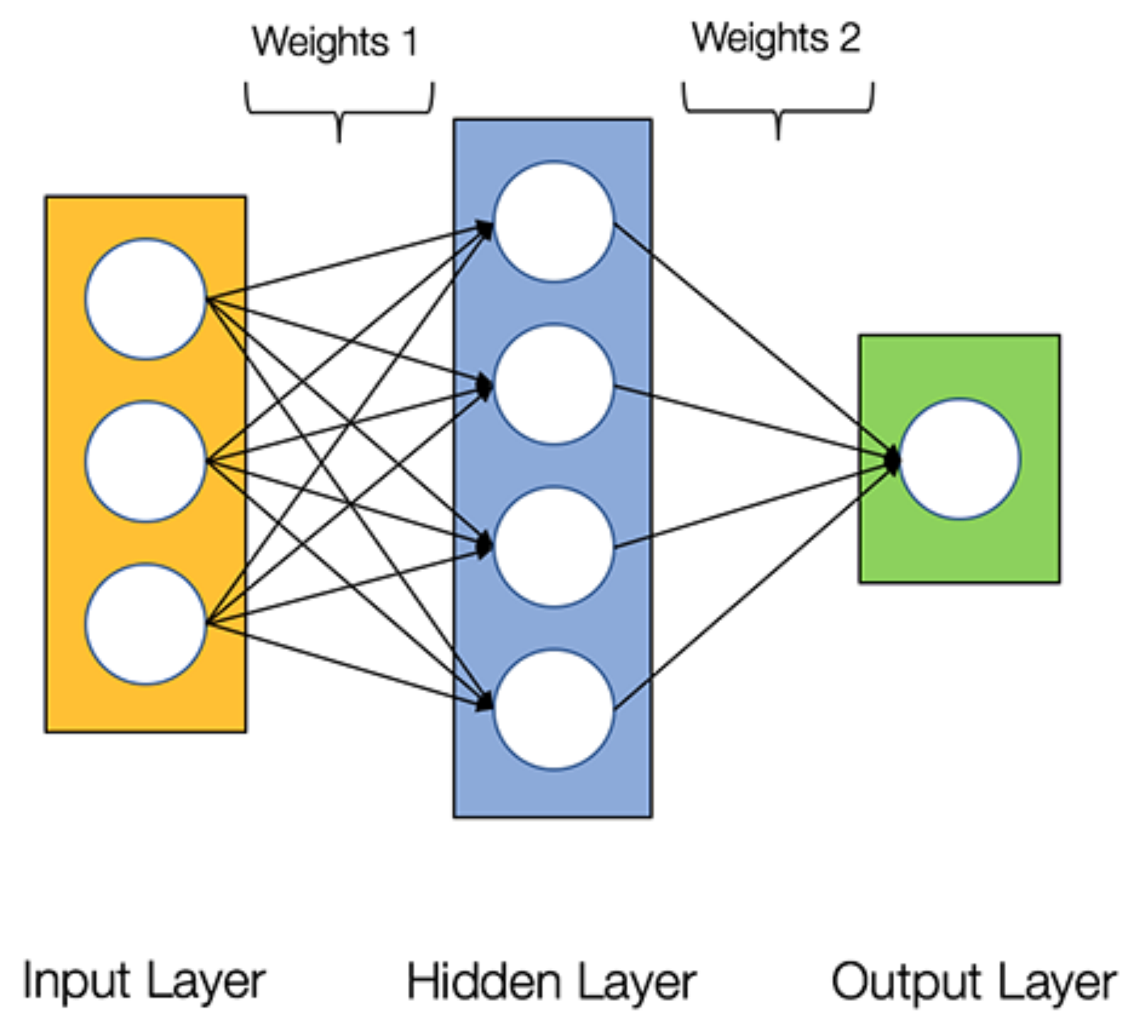
Back-propagate:
Minimize difference between target output and current weights

Neural Networks

- Consider a 2-layer network

- A bit more formally:
  - Input vector $\vec{x}$
  - Output vector $\vec{y}$
  - Weights and biases between layers W and b
  - Activation function σ





Usually rectified linear unit (ReLU), but we will use sigmoid because it is differentiable.

Adapted from https://towardsdatascience.com/how-to-build-your-own-neural-network-from-scratch-in-python-68998a08e4f6

Neural Networks

- For each training step:
  - Feed forward $\vec{x} \rightarrow \hat{y}$:
  - $\hat{y} = \sigma(W_2\sigma(W_1\vec{x} + b_1) + b_2)$
  - Compute loss function (least squares):
    $L(y, \hat{y}) = \sum_i (y - \hat{y})^2$
  - Compute gradient of loss function wrt weights
    $\frac{\partial L(y,\hat{y})}{\partial W}$
  - Back propagate :
    - Optimize using gradient descent (like BFGS!)

Caveats about optimization in multiple dimensions hold here!

Adapted from
https://towardsdatascience.com/how-to-build-your-own-neural-network-from-scratch-in-python-68998a08e4f6

Deep Learning and Neural Networks

- Next step: instead of teaching, let the neural network learn on its own
  - Example 1: Convolutional Neural Network
    - Extract features using convolution, pass features to neural network
    - Fixed-size inputs: Suitable for images
  - Example 2: Recurrent Neural Network
    - Instead of “feed forward” like last time, has possible recurrent links
    - Any size inputs: Suitable for text / speech / handwriting recognition
    - Can also be Recursive (don’t get confused!)

https://en.wikipedia.org/wiki/Deep_learning

https://en.wikipedia.org/wiki/Recurrent_neural_network

https://en.wikipedia.org/wiki/Recursive_neural_network

https://en.wikipedia.org/wiki/Convolutional_neural_network

https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-core-concepts/

Deep Learning and Neural Networks

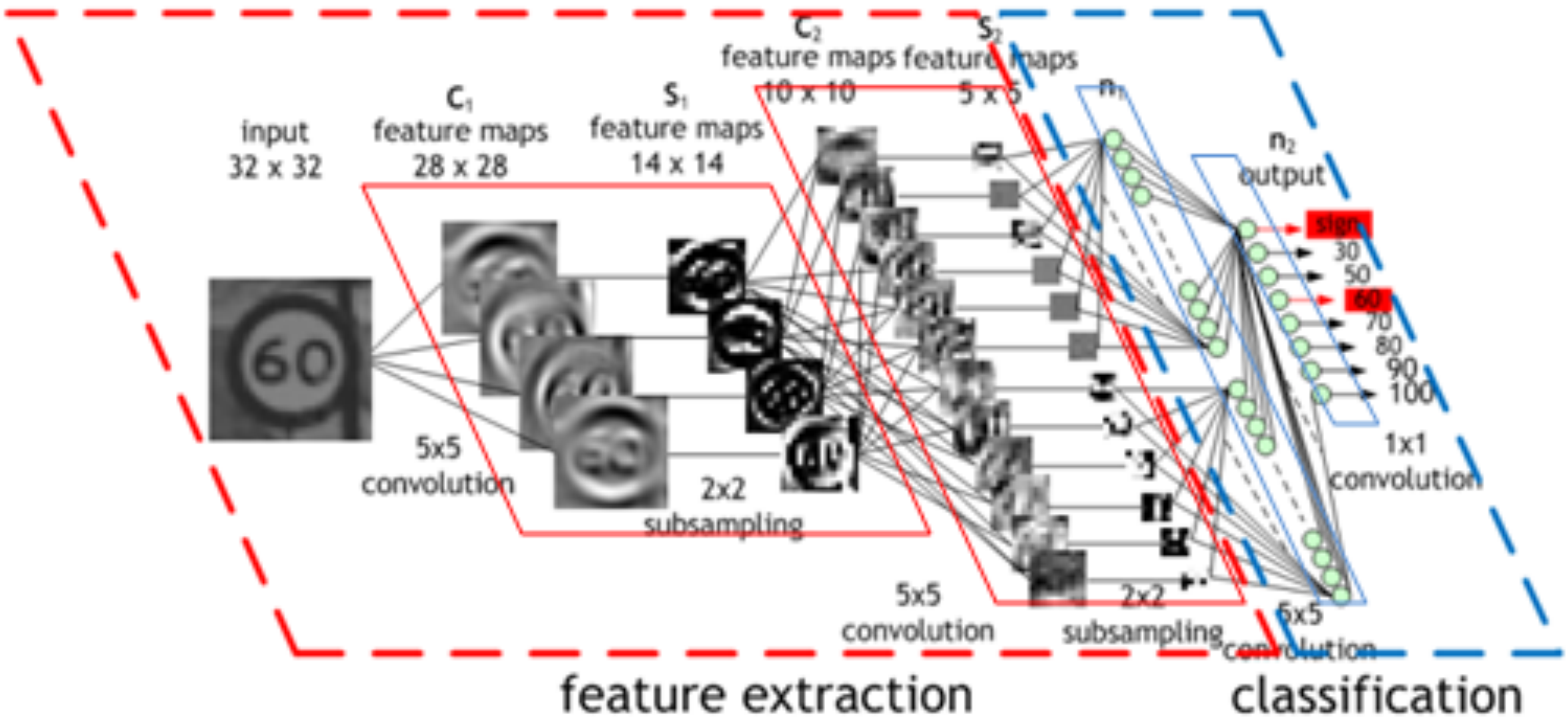
- Options to train:

- Supervised
  - Give inputs, tell it what the output is

- Unsupervised
  - Give inputs, tell it to optimize a function

Convolutional Neural Networks

- Preprocessing to perform feature extraction
  - Convolution or otherwise
- Classification
  - Standard neural network

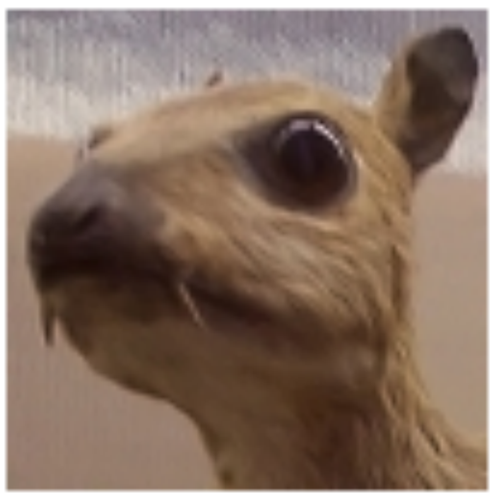


Convolutional Neural Networks

- Does “preprocessing” of the image by convoluting with kernels:

2-d Fourier transform!

Input image



Convolution Kernel

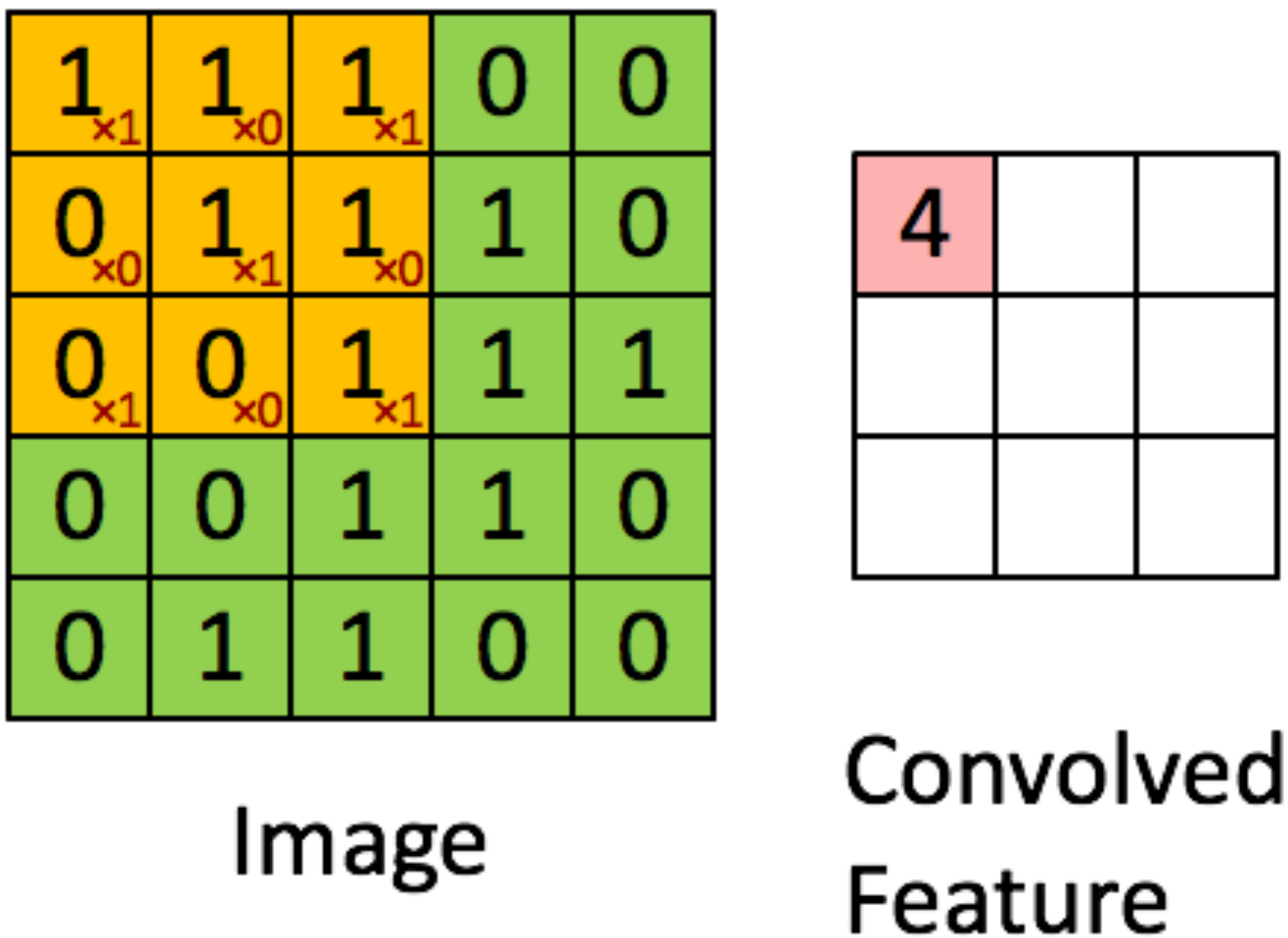
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Feature map

Convolutional Neural Networks

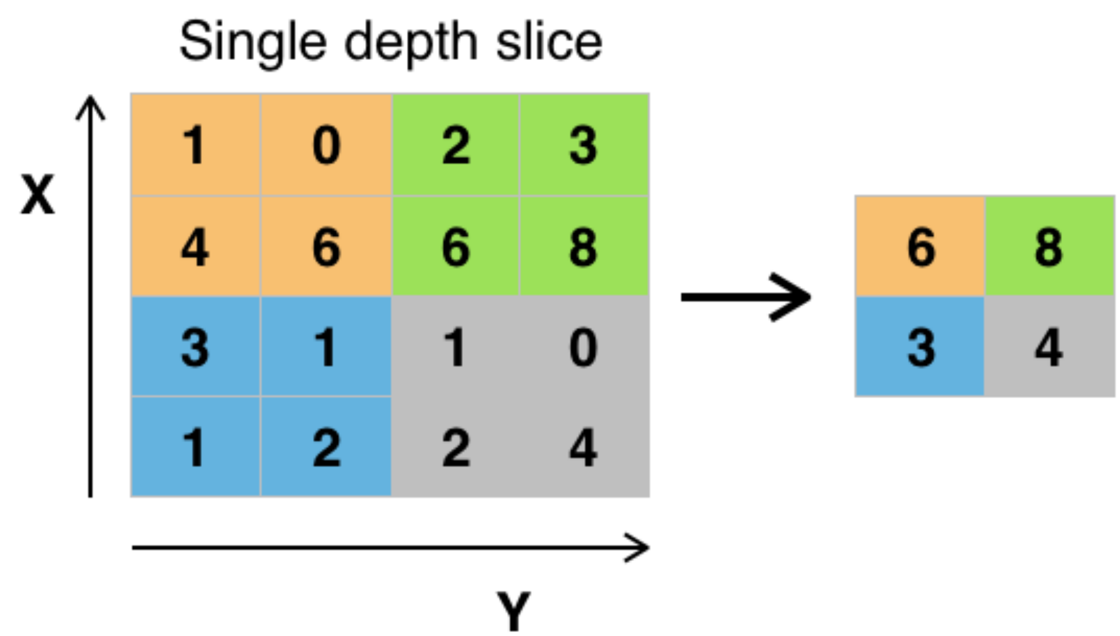
- Animation of convolution:
- Kernel:

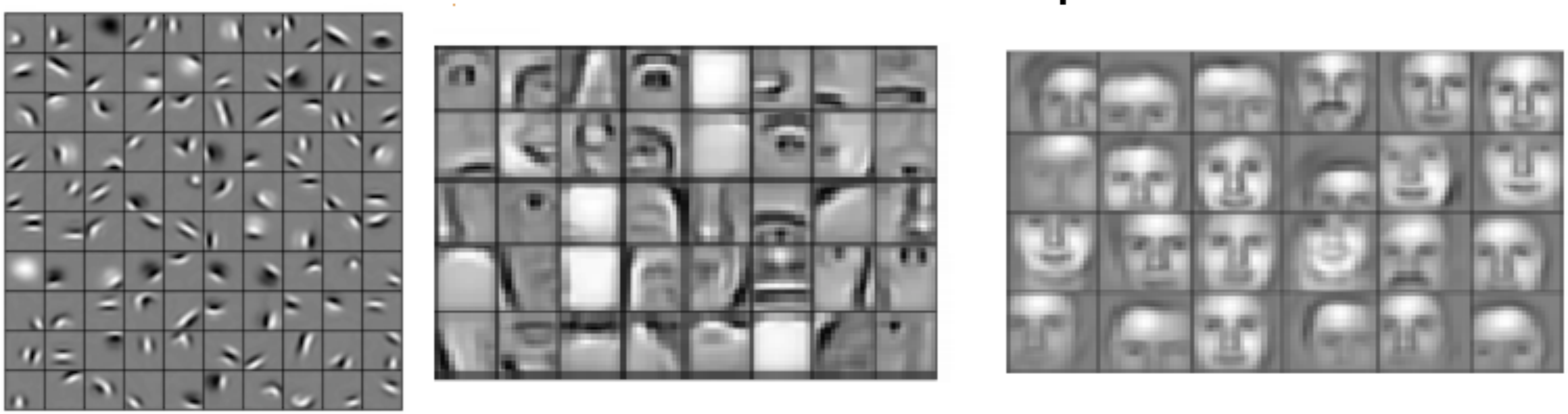
1	0	1
0	1	0
1	0	1



Convolutional Neural Networks

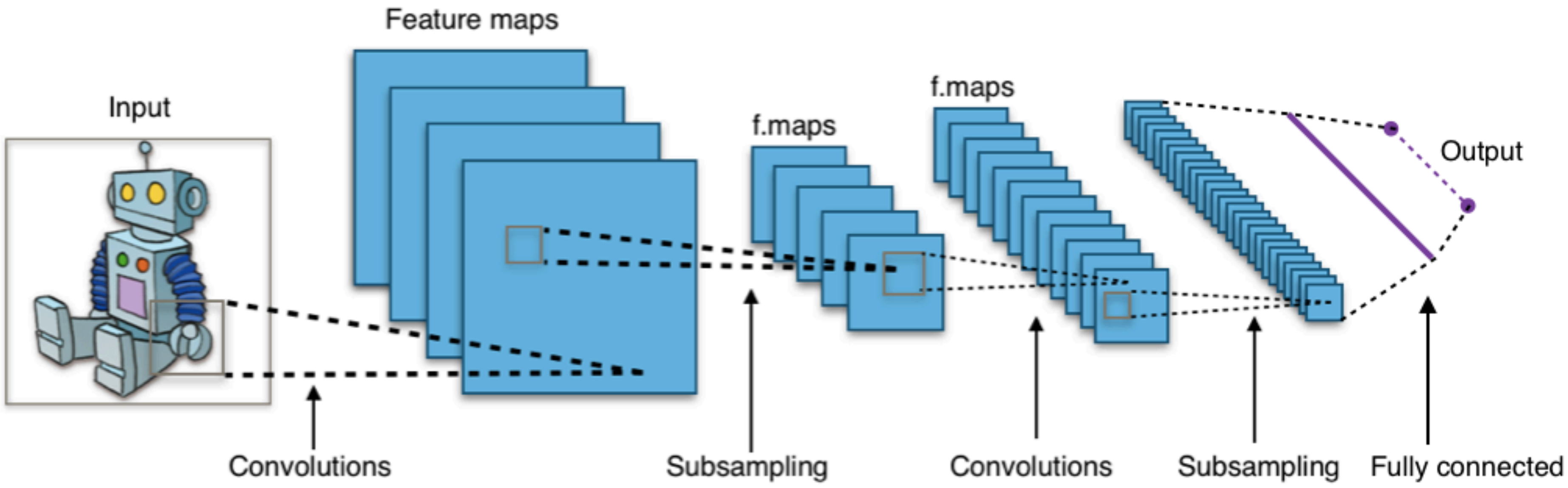
- Pooling layer : used to obtain very big features
  - Downsampled to extract gross features
  - Is it “pointy”?
  - Is it “round”?
  - Does it have a face?





Convolutional Neural Networks

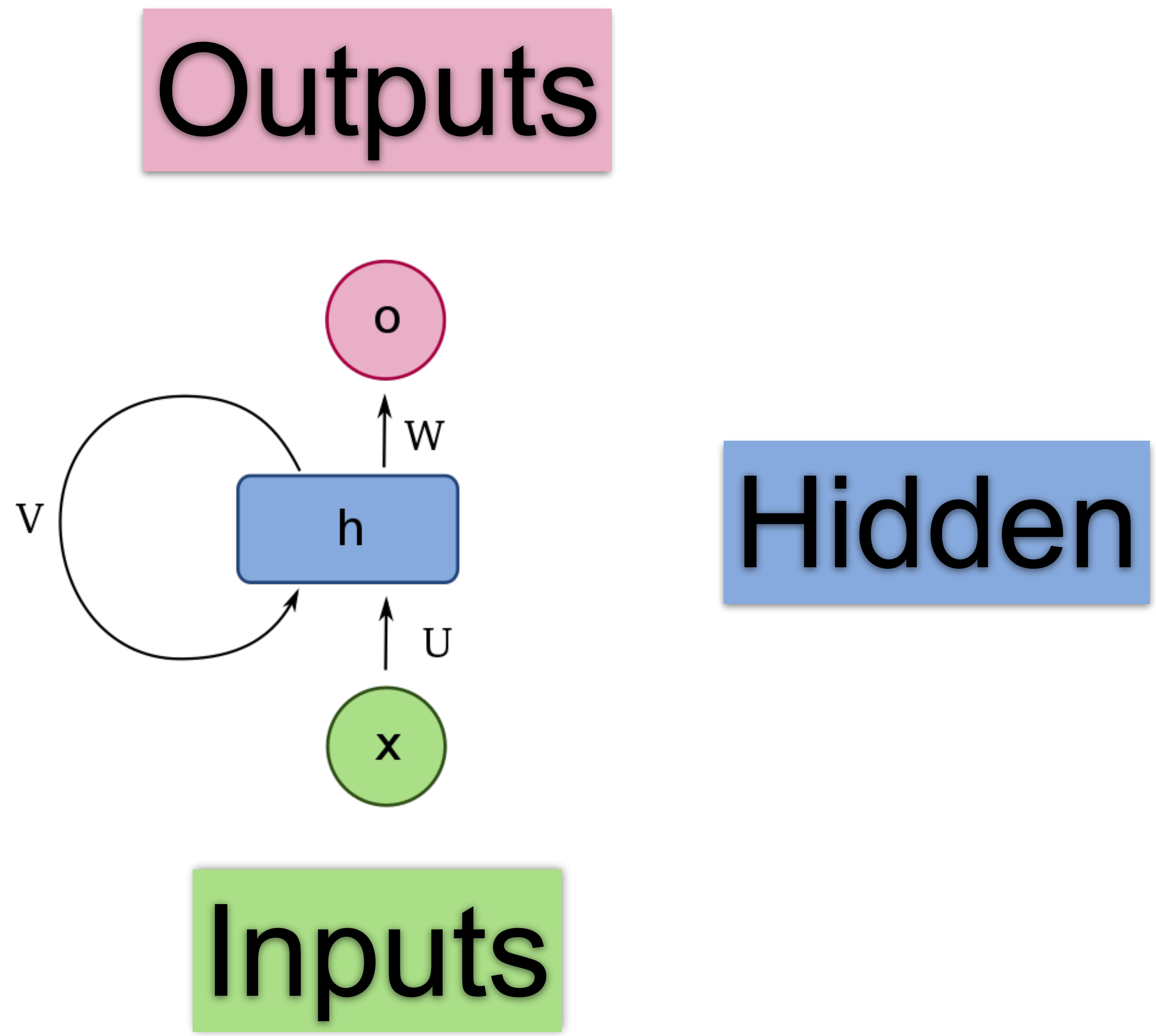
- Putting it together:
  - Convolutional layer
    - Discern features
  - Pooling layer
    - Extract global features
  - Activation function ("Rectified Linear Units")
    - Standard NN
  - Fully connected layer
    - Standard NN
  - Loss layer
    - How well did it



https://en.wikipedia.org/wiki/Convolutional_neural_network

Recurrent Neural Networks

Dynamic directed graph to process **sequences** of inputs

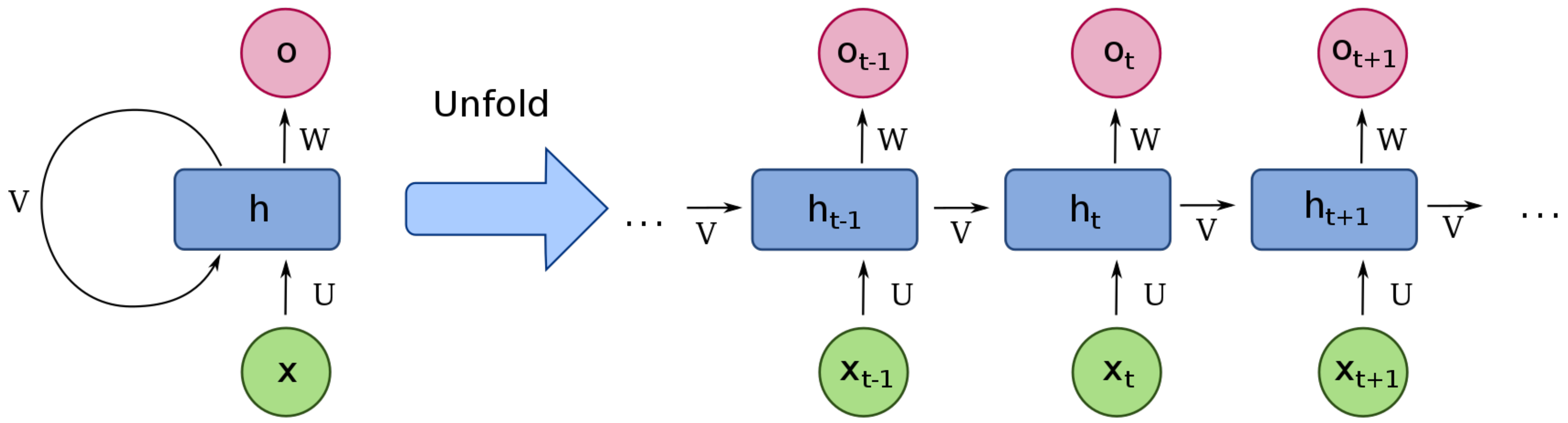


https://en.wikipedia.org/wiki/Recurrent_neural_network

Recurrent Neural Networks

Dynamic directed graph to process **sequences** of inputs

"Dynamic" in the sense that it changes over time



https://en.wikipedia.org/wiki/Recurrent_neural_network

Recurrent Neural Networks

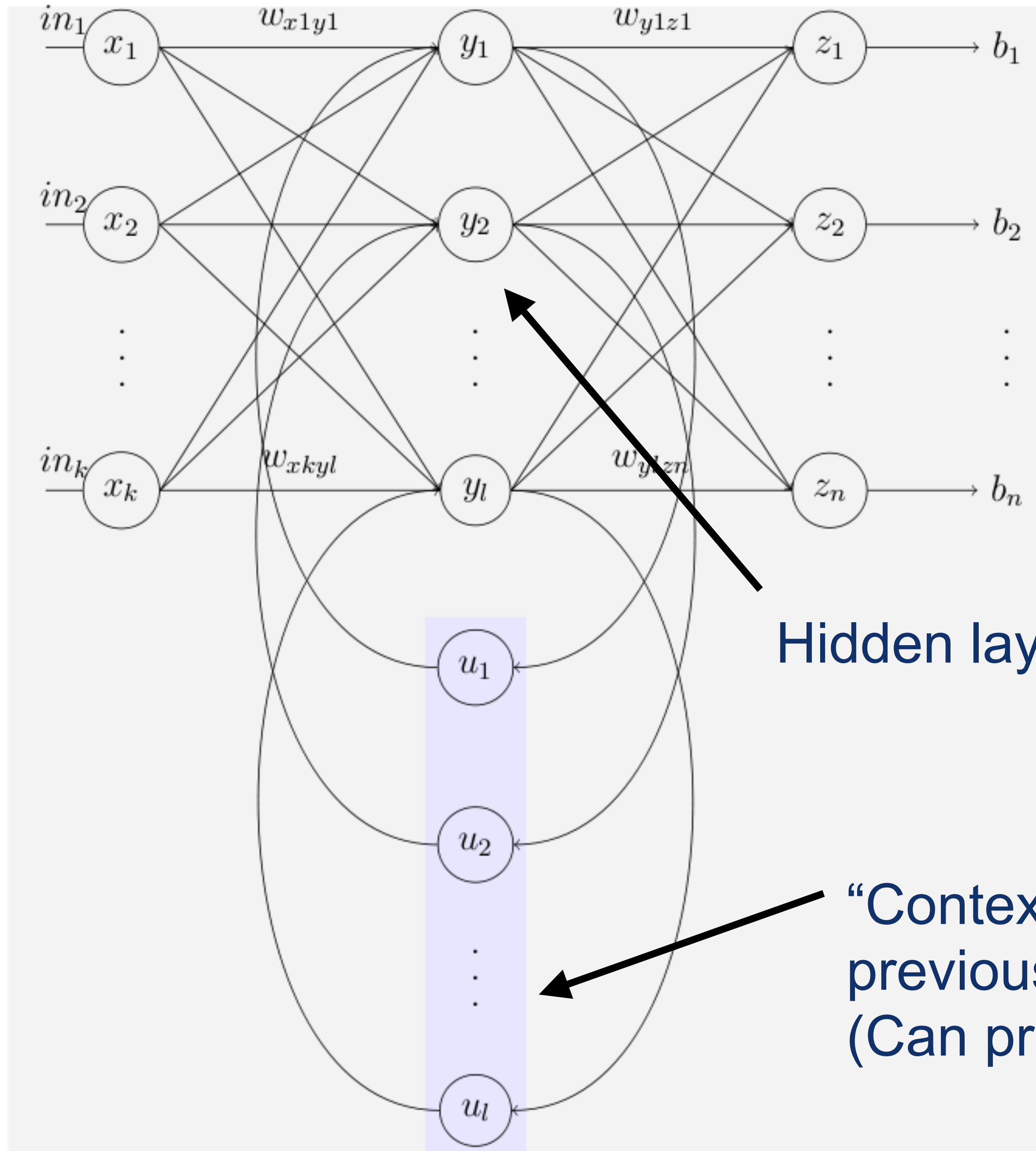
- “Recurrent” : Can “save state” of the inputs
  - (CNNs do not)

- Has use in predicting the next value in a sequence (speech, text recognition)

- Can also see how similar two sequences are

- Example: Elman network

Inputs

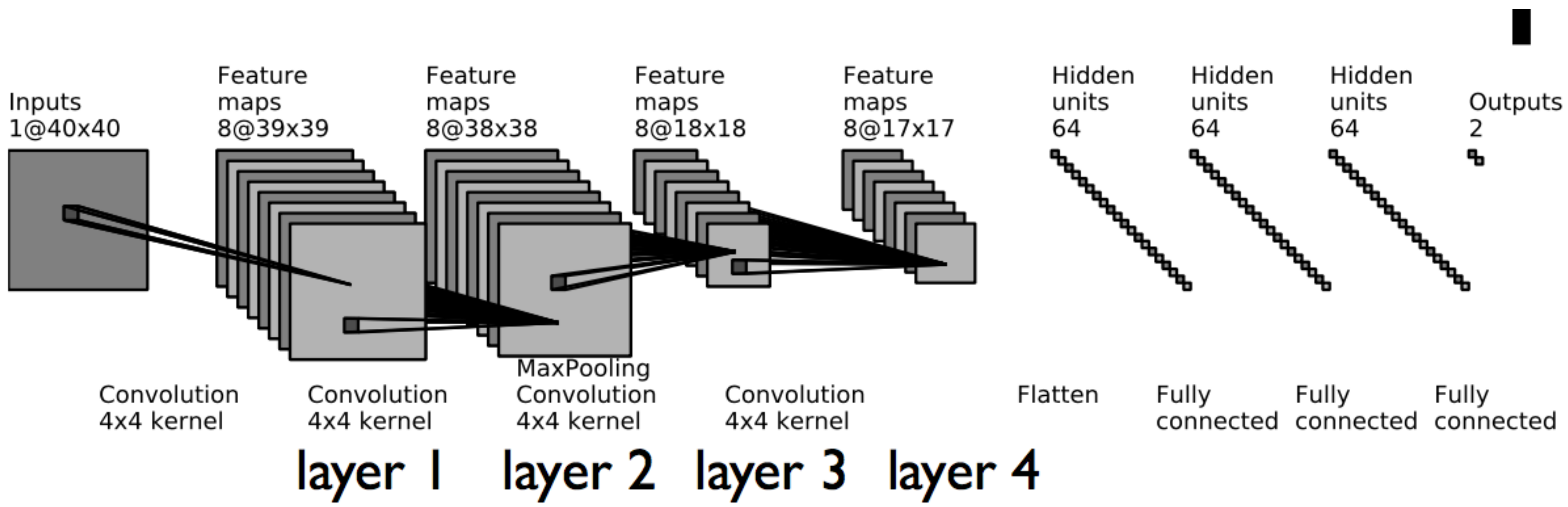


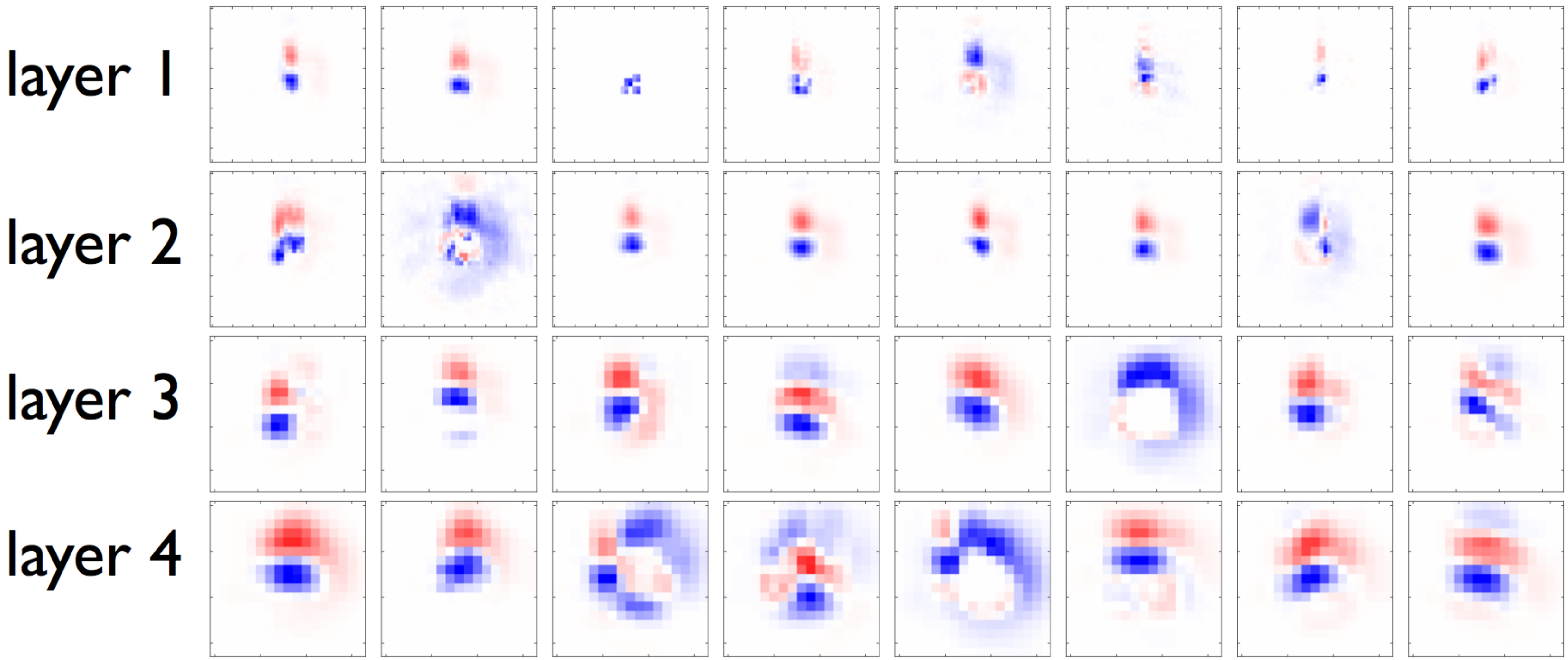
Outputs

Hidden layer

“Context” layer : Stores previous hidden layer. (Can predict!)

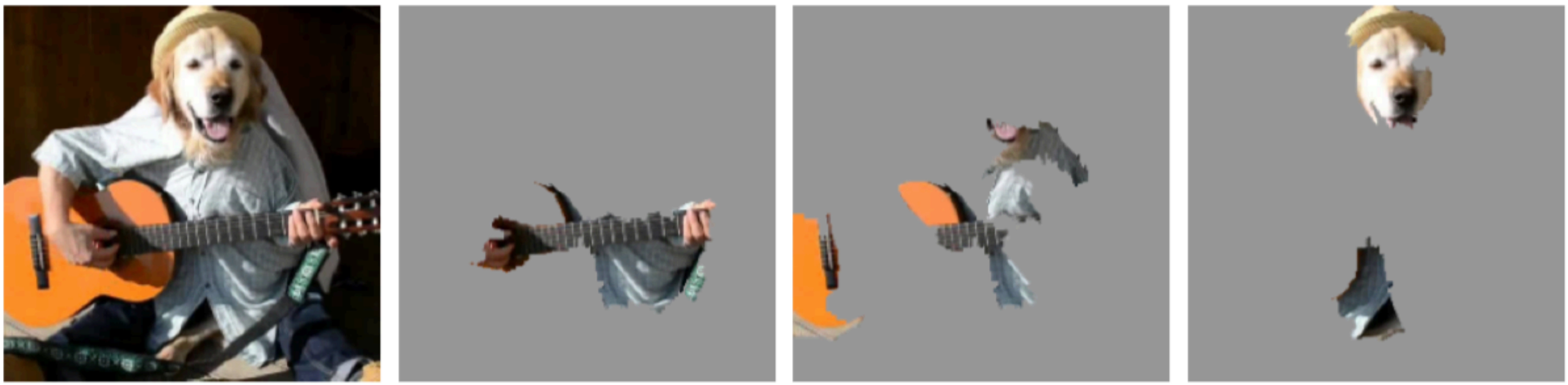
Boosted Jets





Explainable AI

- Use locally-interpretable models to understand what is happening:



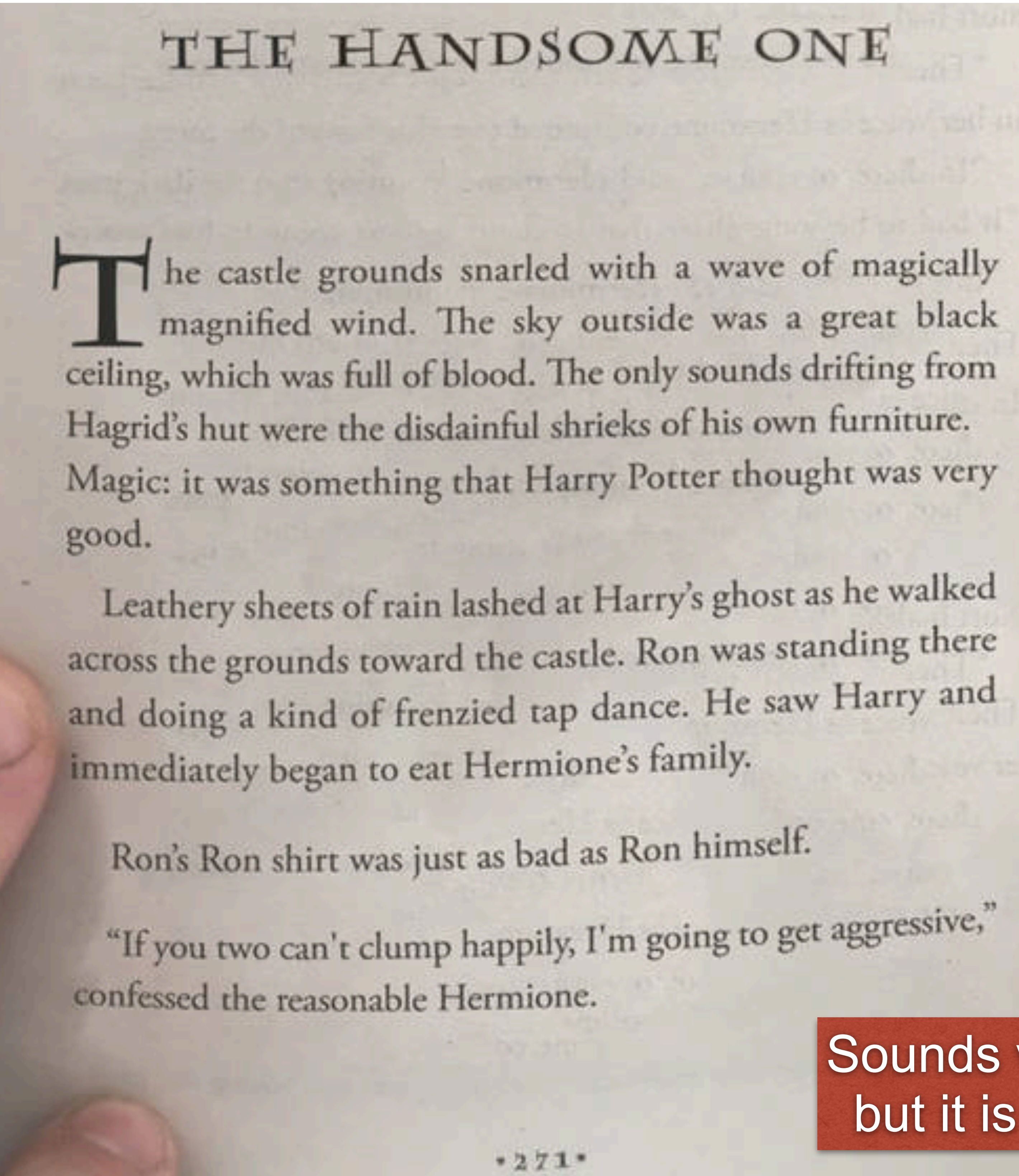
(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

https://arxiv.org/pdf/1602.04938.pdf

Just how good are we yet?

https://botnik.org/content/harry-potter.html



THE HANDSOME ONE

The castle grounds snarled with a wave of magically magnified wind. The sky outside was a great black ceiling, which was full of blood. The only sounds drifting from Hagrid's hut were the disdainful shrieks of his own furniture. Magic: it was something that Harry Potter thought was very good.

Leathery sheets of rain lashed at Harry's ghost as he walked across the grounds toward the castle. Ron was standing there and doing a kind of frenzied tap dance. He saw Harry and immediately began to eat Hermione's family.

Ron's Ron shirt was just as bad as Ron himself.

"If you two can't clump happily, I'm going to get aggressive," confessed the reasonable Hermione.

•271•

https://thenewstack.io/meet-ai-makes-absurd-inspirational-posters/

Sounds vaguely real, but it is clearly not.