Boosted Top Tagging with Long Short-Term Memory Networks

Shannon Egan
University of British Columbia

TOP2017

19 September 2017

Why boosted tops?

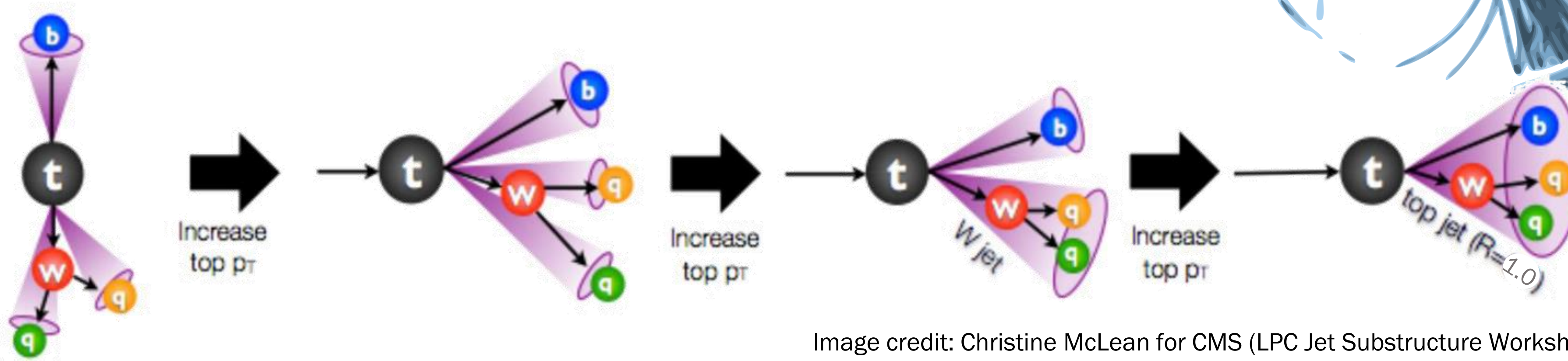
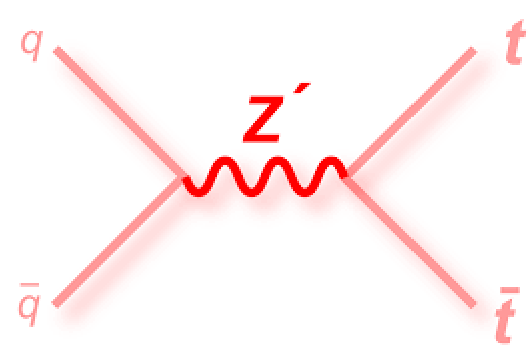


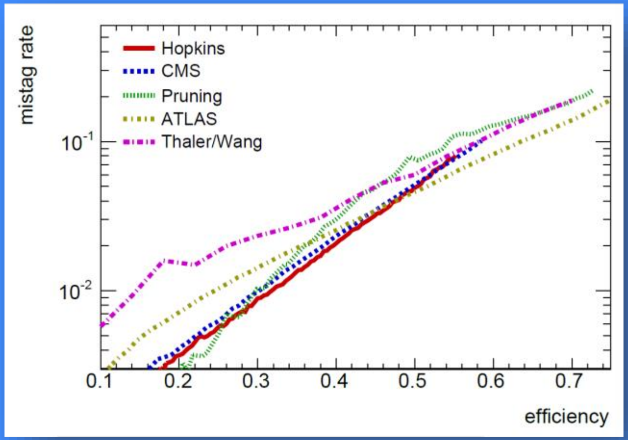
Image credit: Christine McLean for CMS (LPC Jet Substructure Workshop)

- Boosted tops in BSM physics: heavy resonances decaying to $t\bar{t}$ pairs, VLQ, SUSY
- Top jets are difficult to distinguish from background – hand-made taggers in use



Traditional top taggers

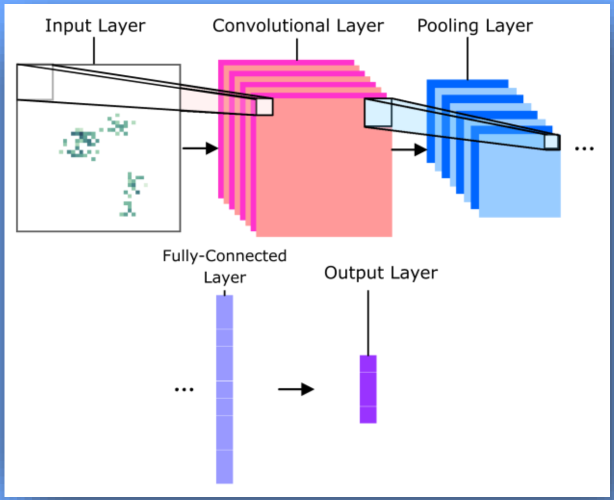
Jet mass, high level QCD variable inputs



- Use QCD motivated variables (τ_{32}, N-subjettiness, jet mass) and clustering history to identify top candidates
- Plehn and Spannowsky (2011, arXiv:1112.4441) show these methods reach background rejection at 0.5 signal efficiency of 10-15

Recent developments: Convolutional Neural Networks (CNN)

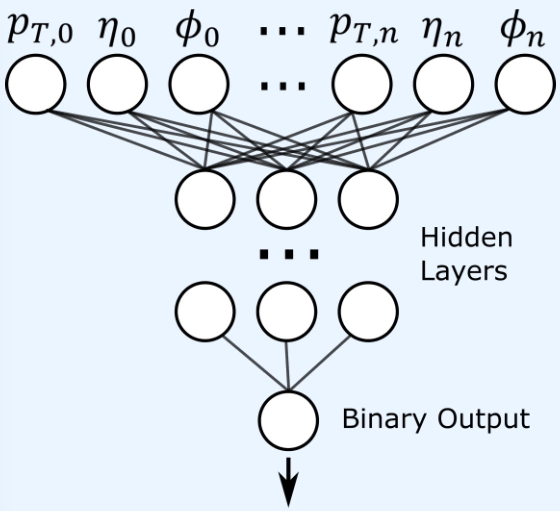
Jet image inputs



- Network alternates between convolution and pooling to progressively extract information from and down-sample jet images before fully-connected layers make a prediction

Our previous work: Deep Dense Neural Networks (DNN)

Particle 4-momenta inputs



arXiV:1704.02124v2

- Network receives flat list containing each particle's transverse momentum, pseudorapidity and azimuth as input and feeds information through a series of fully-connected (Dense) layers

Anatomy of an LSTM

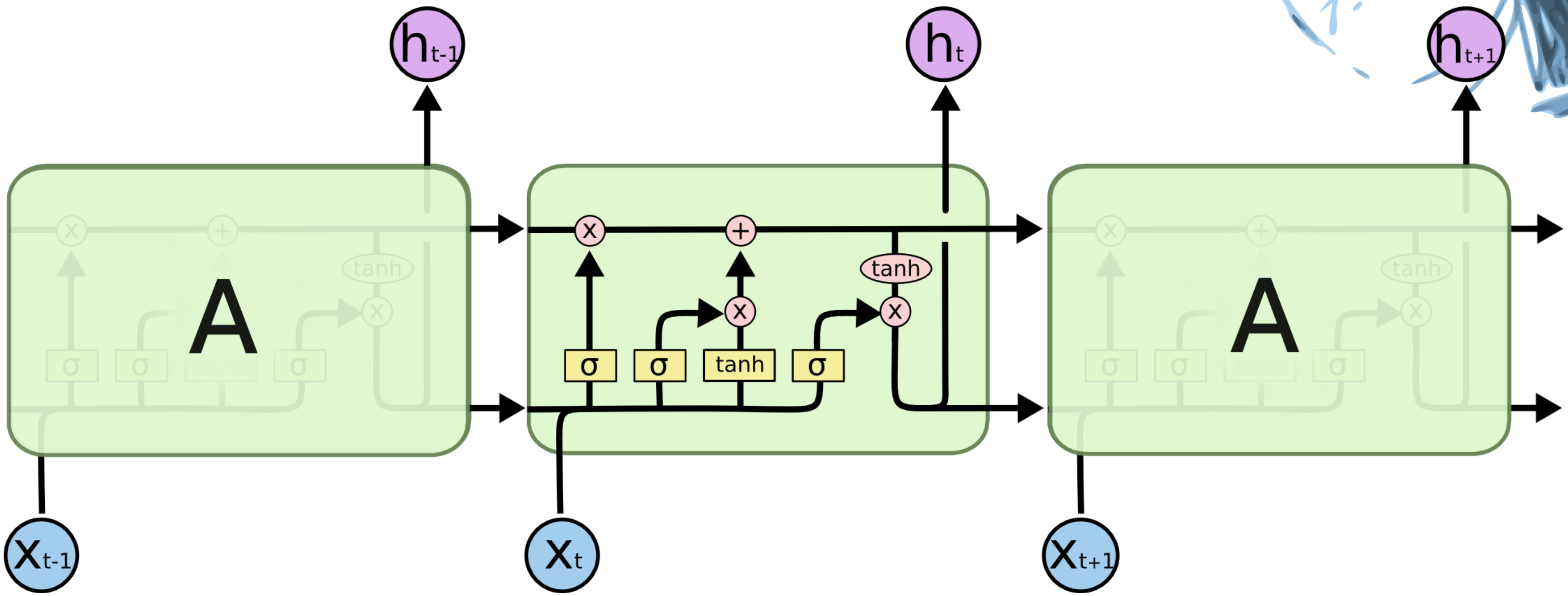


Image credit: colah's blog (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

3 factors affect the final output of an LSTM cell:

1. Input at given timestep (x_t)
2. Output at previous timestep (h_{t-1})
3. Current value of cell state (C_t)

LSTM inputs and outputs

$$\begin{bmatrix} p_{T,n} \\ \eta_n \\ \phi_n \end{bmatrix}, \dots, \begin{bmatrix} p_{T,1} \\ \eta_1 \\ \phi_1 \end{bmatrix}, \begin{bmatrix} p_{T,0} \\ \eta_0 \\ \phi_0 \end{bmatrix} \rightarrow \{LSTM\} \dashrightarrow \{Dense\}$$

$$\begin{bmatrix} p_{T,n} \\ \eta_n \\ \phi_n \end{bmatrix}, \dots, \begin{bmatrix} p_{T,1} \\ \eta_1 \\ \phi_1 \end{bmatrix} \rightarrow \{LSTM\} \dashrightarrow \{Dense\}$$

- The LSTM does not output to the Dense layer until the final timestep

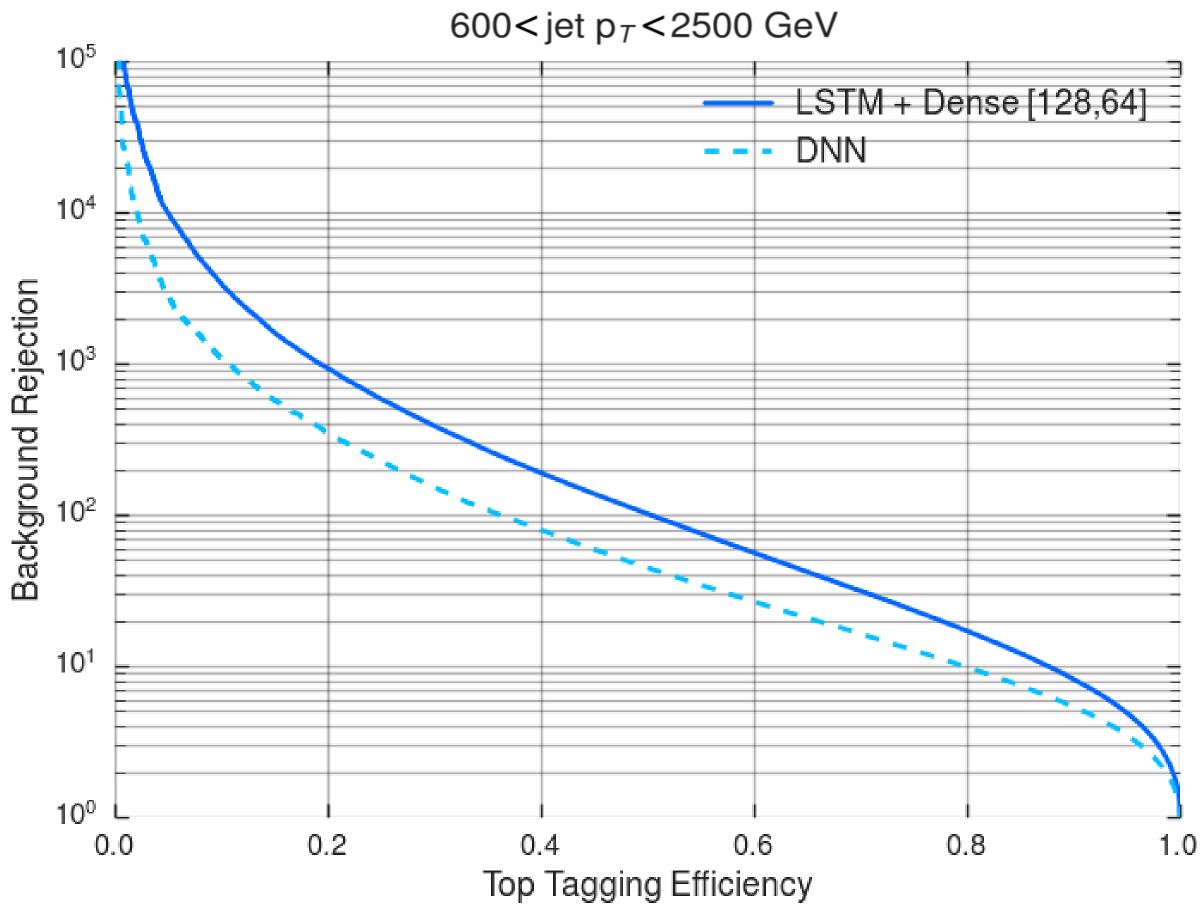
$$\begin{bmatrix} p_{T,n} \\ \eta_n \\ \phi_n \end{bmatrix} \rightarrow \{LSTM\} \rightarrow \{Dense\}$$

Simulation and jet preselection

- Signal: Z' → $t\bar{t}$
- Background: dijets
- Generated with PYTHIA v8.219 NNPDF23 LO AS 0130 QED PDF
- DELPHES v3.4.0 using default CMS card, particle-flow

- Selected jets are flat in p_T, signal matched in eta
- $600 \leq p_{T,jet} \leq 2500$ GeV
- ~ 4 million signal jets and ~4 million background jets
  - Sample divided into 80%, 10%, 10% for training, validation and testing
  - Network evaluated on an orthogonal set of ~8 million jets

Comparison to DNN



Model	BR @ 50% SE	BR @ 80% SE
DNN [300, 150, 50, 10, 5, 1]	45.4	9.8
LSTM + Dense [128,64,1]	101	17

Key Metrics

- Signal efficiency (SE)

$$SE = \frac{s}{S}$$

- Background rejection (BR)

$$BR = \frac{B}{b}$$

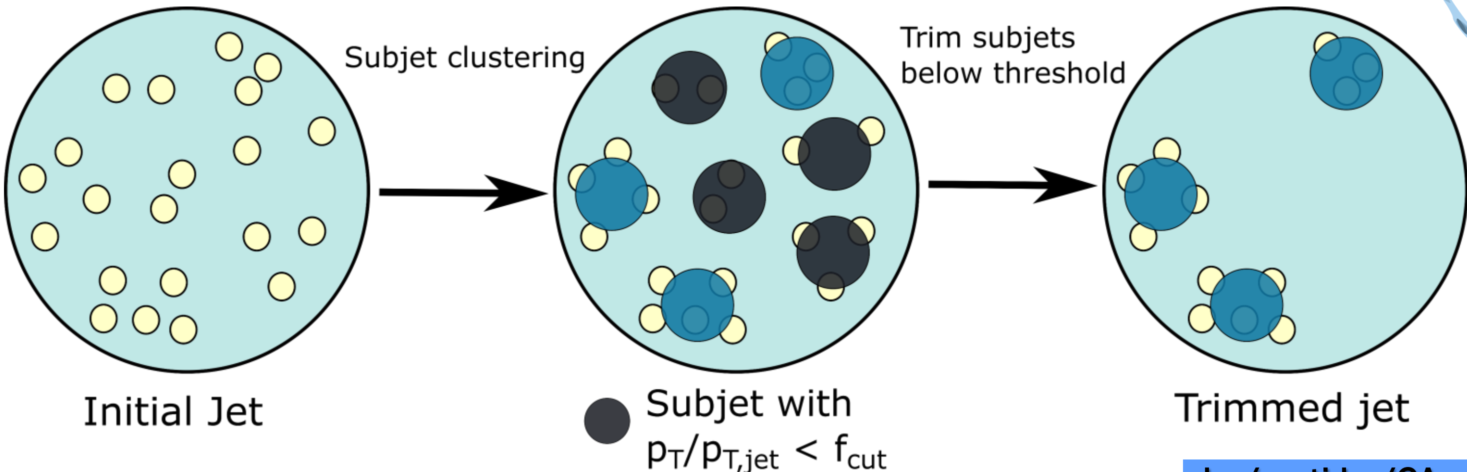
s - tagged signal jets

S - true signal jets

b – background jets tagged as signal

B – true background jets

Trimming and subjet sorting



Trimming algorithm

for jet in list of jets:

Recluster particles into *subjets* using k_T algorithm

Compute the transverse momentum ($p_{T,\,subjet}$) of each subjet

If $p_{T,\,subjet}/p_{T,\,jet} < f_{cut}$

Remove subjet constituents from list of jet particles

k_T / anti-k_T /CA algorithm

while # unclustered particles > 0***:***

Compute distance between all pairs of particles (d_{ij}) and from each particle to beam (d_{iB})

if minimum distance is d_{ij}***:***

Sum 4-momenta of *i* and *j* and add to list of particles. Remove *i* and *j* from list

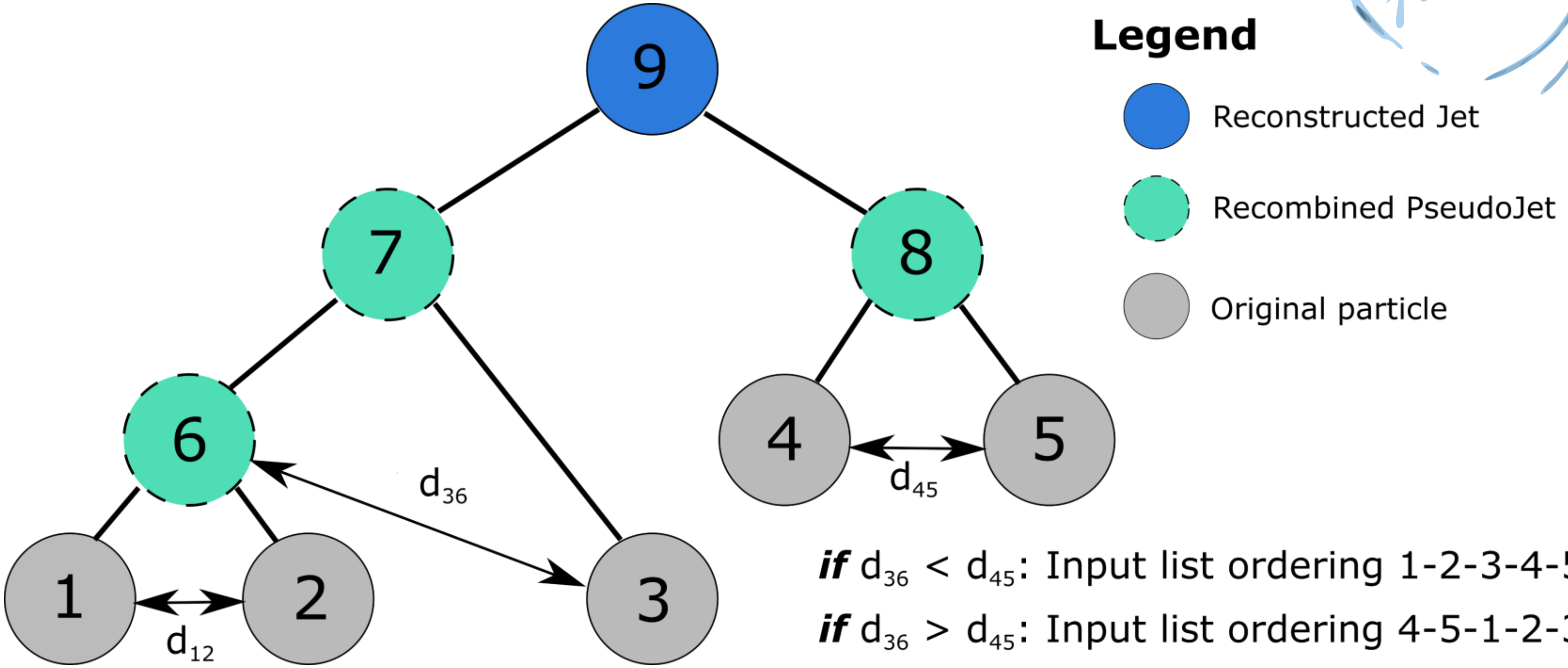
if smallest distance is d_{iB}***:***

Label *i* as a jet and remove from list

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

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

Jet structure sorting

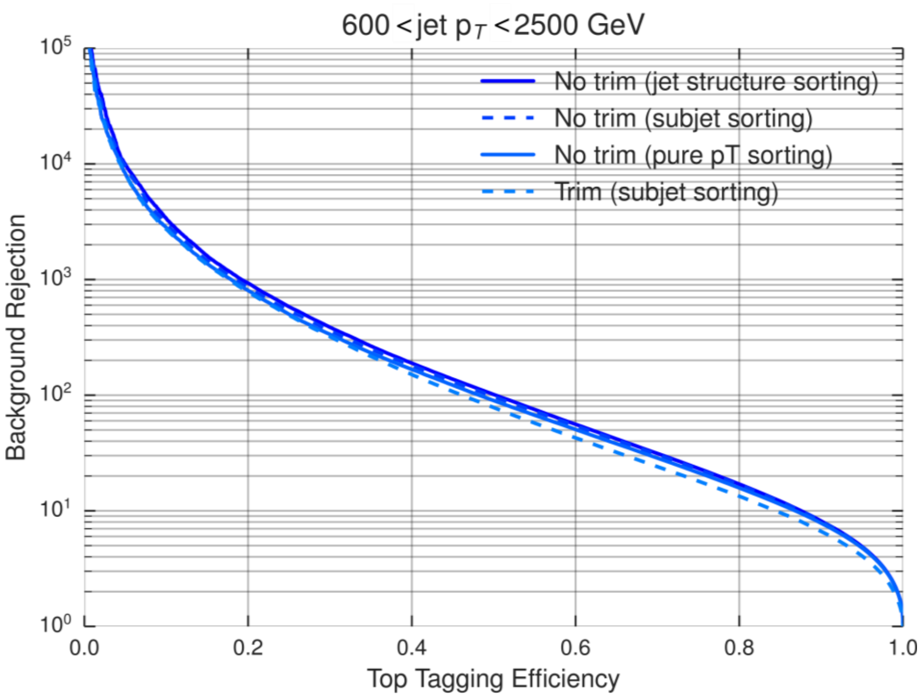


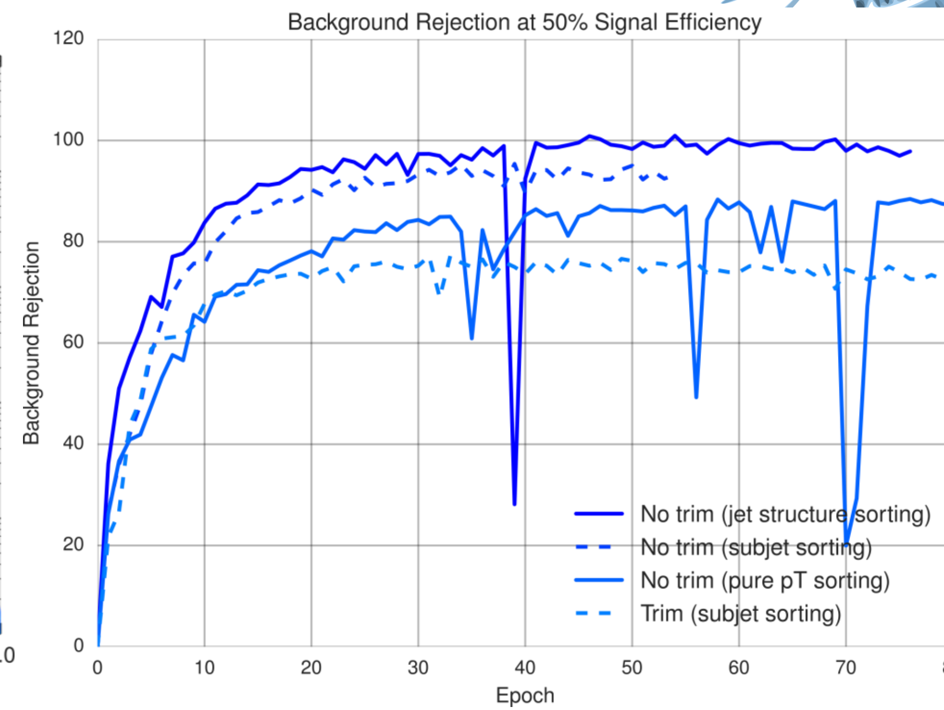
if $d_{36} < d_{45}$: Input list ordering 1-2-3-4-5

if $d_{36} > d_{45}$: Input list ordering 4-5-1-2-3

- We developed a recursive algorithm that performs a depth first traversal of the clustering tree
- **<u>Goal:</u> add particles to the input list in an order that reflects their closeness in the jet substructure**

Jet structure sorting - Results





- Only ~1% of background jets mistagged as signal by best performing network

Conclusions and next steps

What we learned:

- Implementing a boosted top tagger using LSTMs yields greater than factor of 2 improvement over our DNN model, which already improves on existing methods
- Constituent ordering carries important information; modest effects on network performance

Next steps:

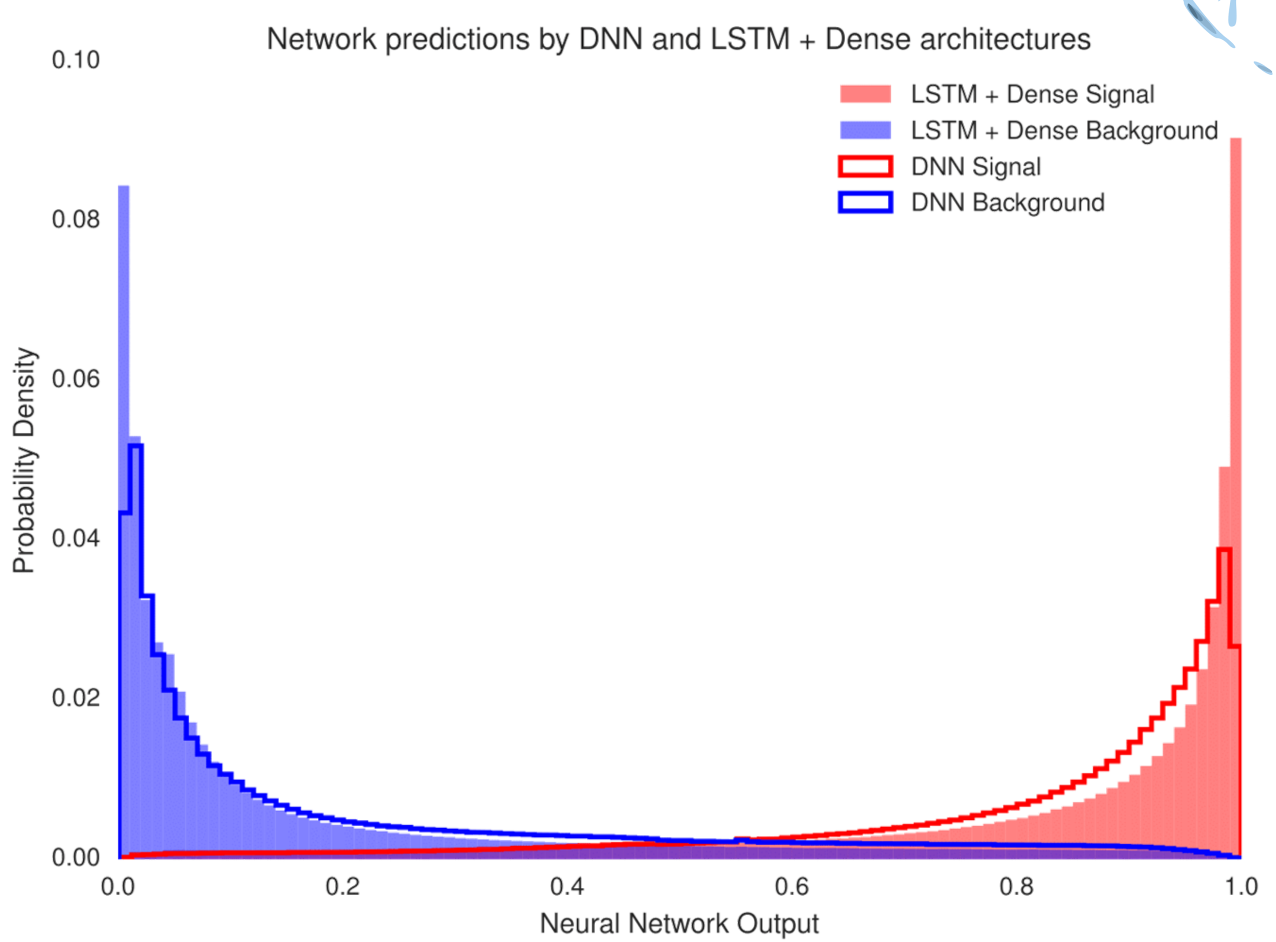
- Test additional sorting methods, e.g. chronological clustering order
- Further analyze effects of pileup, p_T dependence, trimming etc. and try to improve resilience
- Modelling uncertainties
- Look at performance on data

Thanks for listening!

shannon.monica.egan@cern.ch

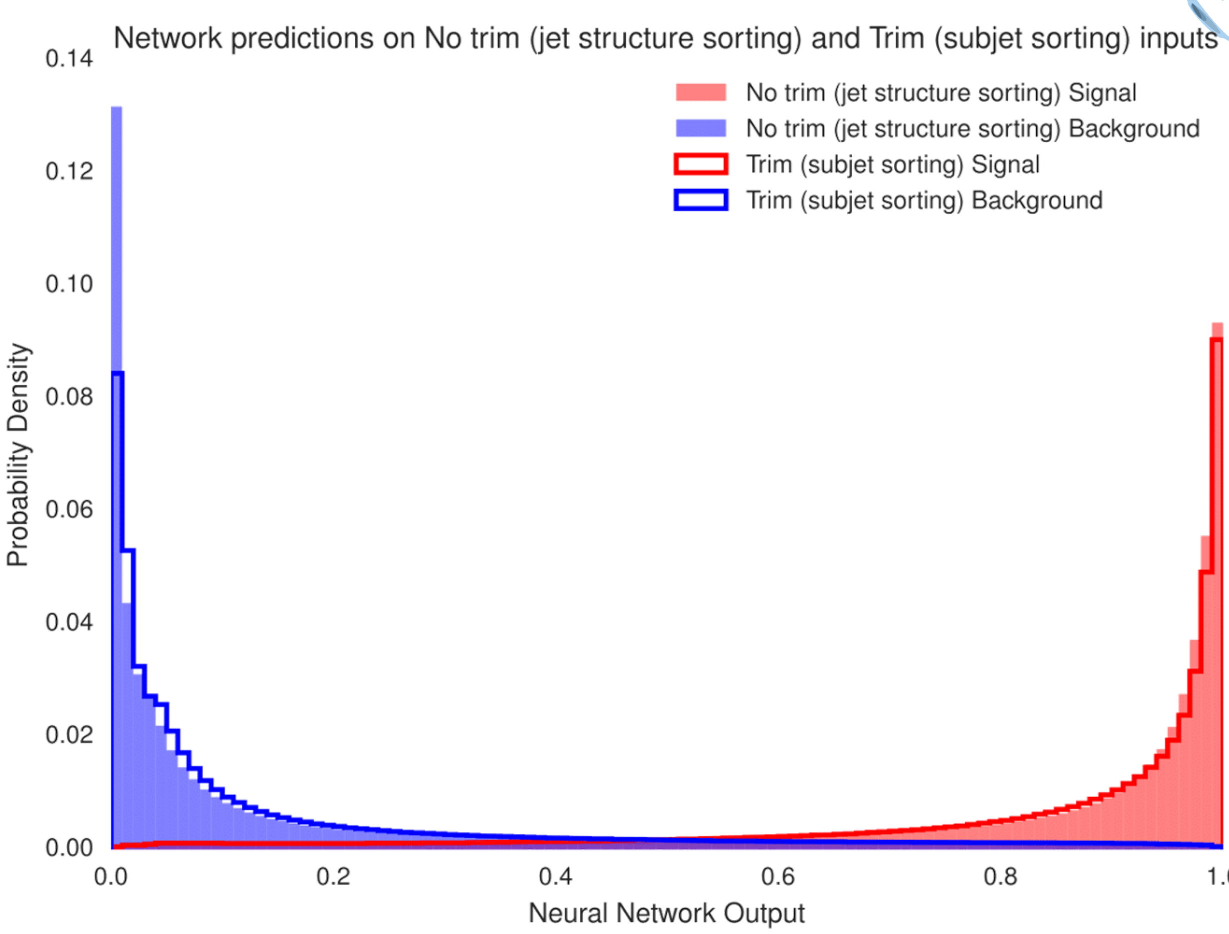
Backup

Prediction histograms



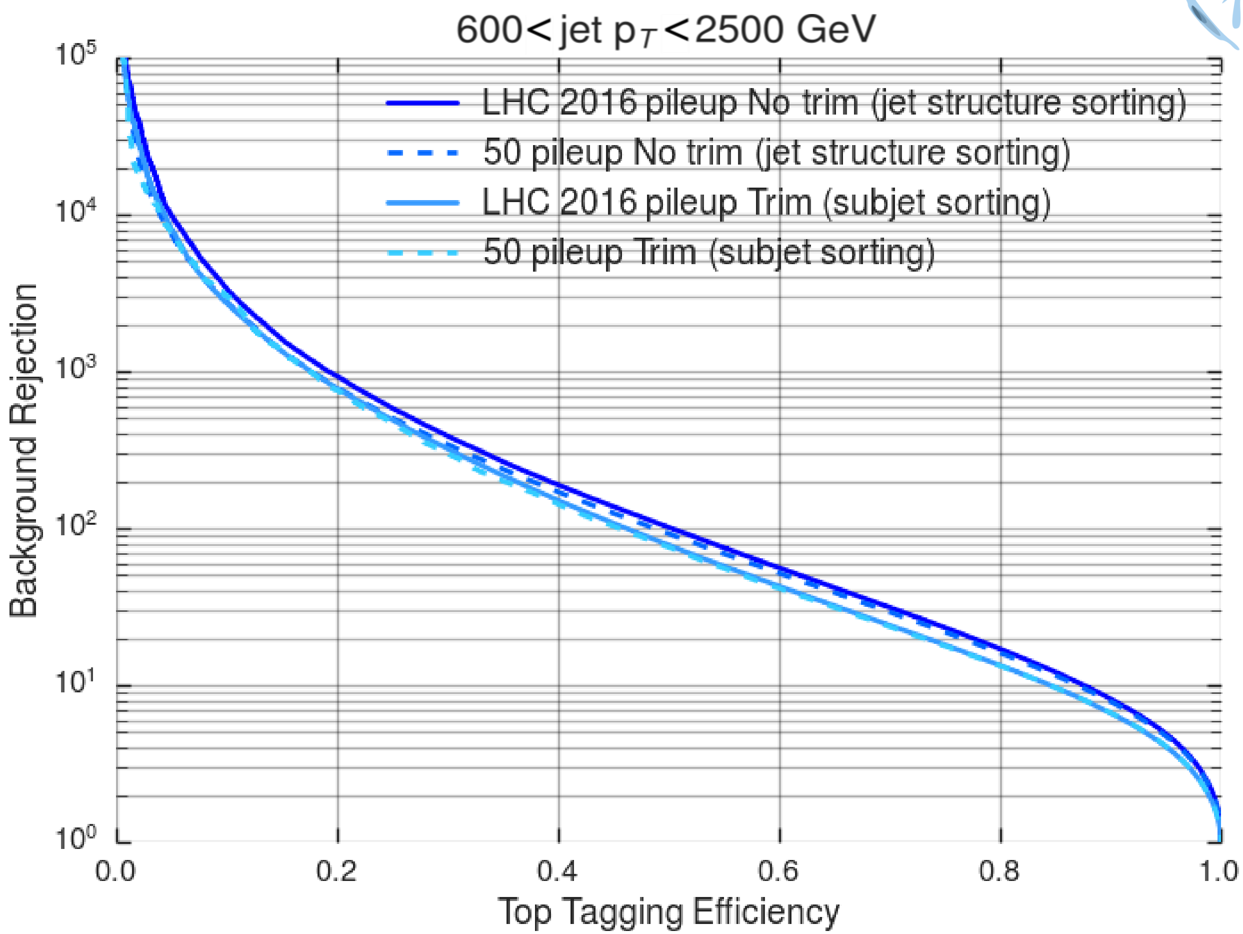
- Evaluated on trimmed inputs with subjet sorting
- Gains in background rejection largely come from better identifying signal

Prediction histograms



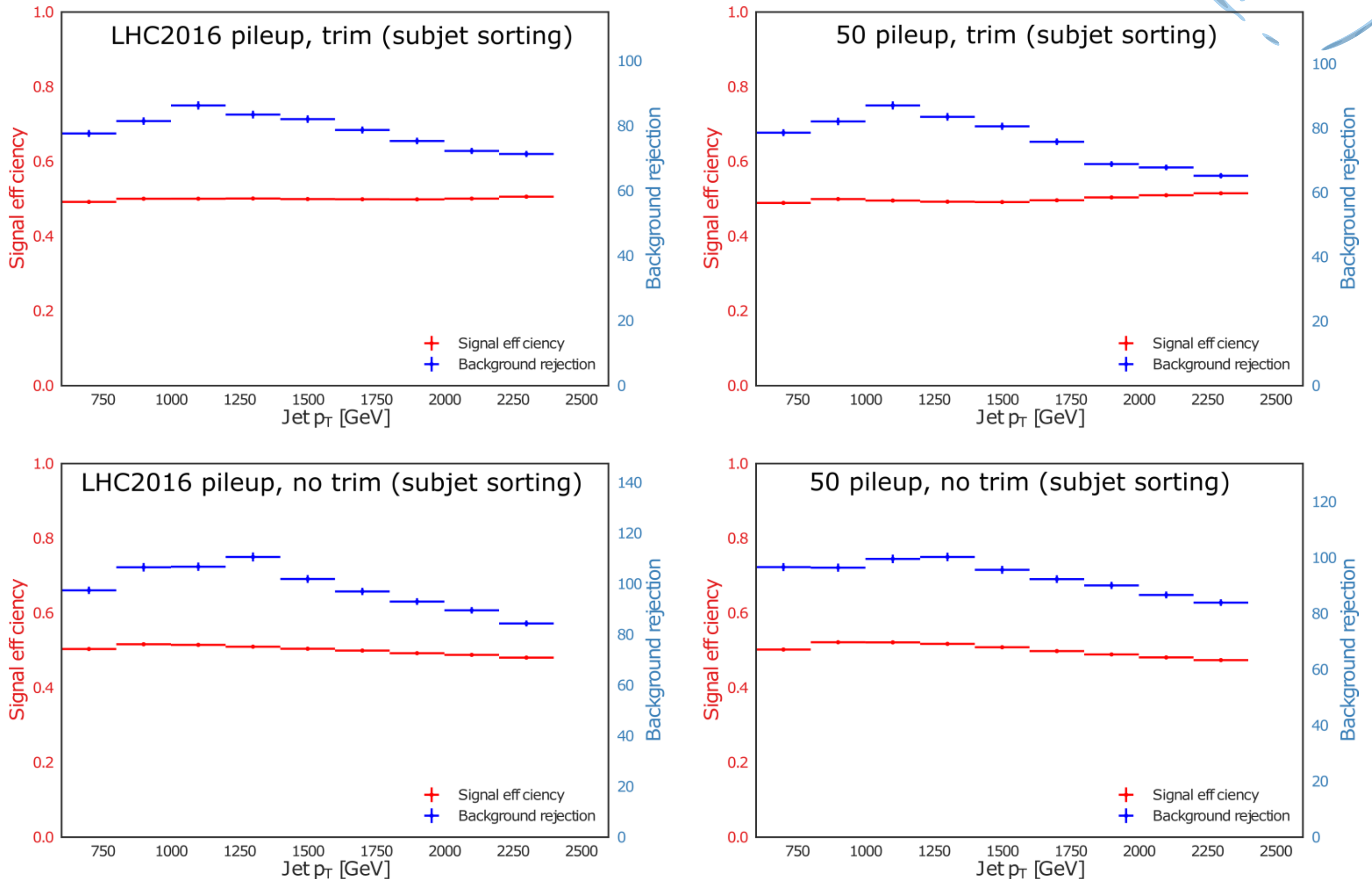
- Evaluated on LSTM + Dense [128, 64]
- Gains in background rejection largely come from better classifying background

Pileup and trimming effects



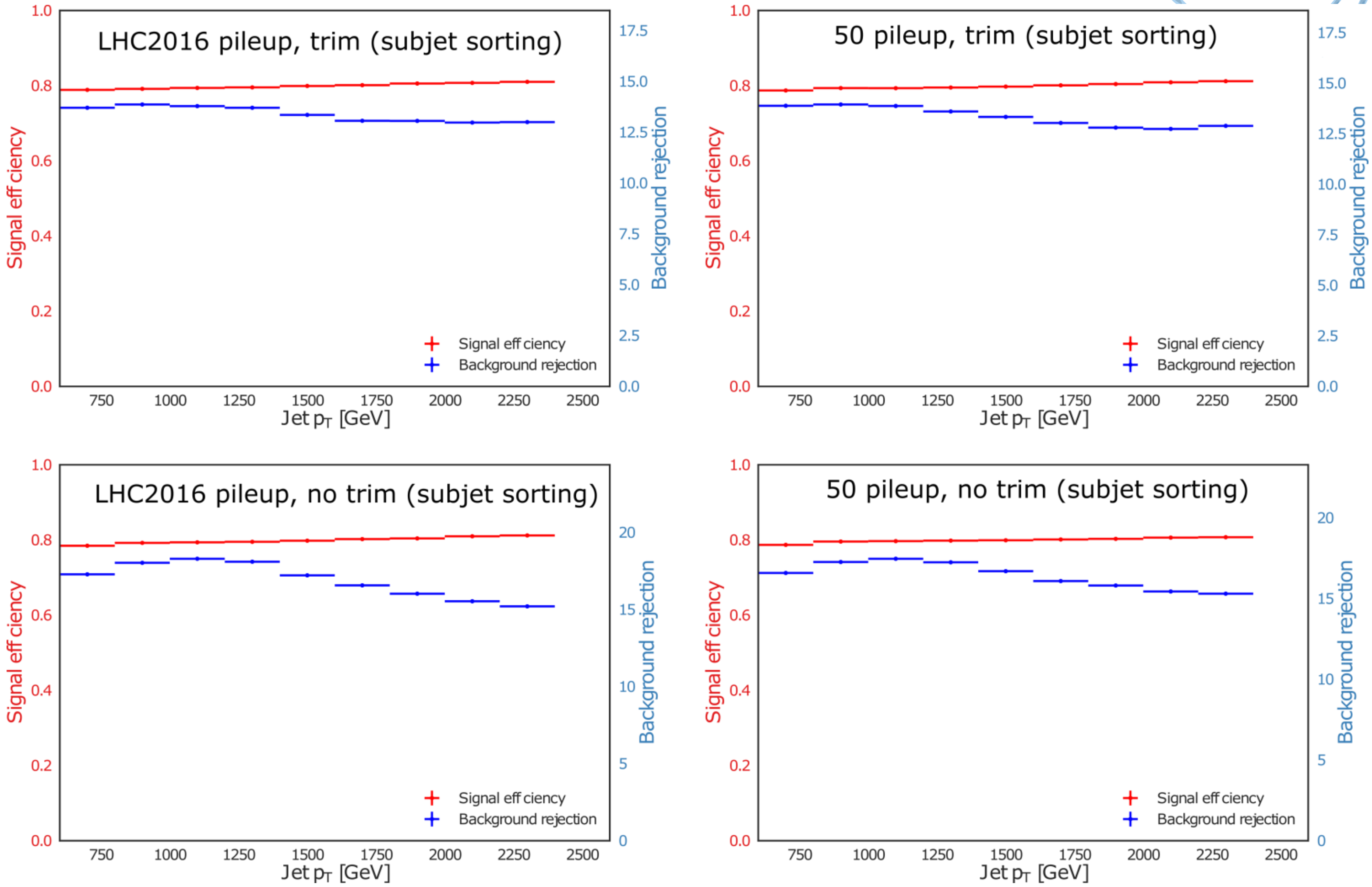
- No trim results in better performance in either pileup case
- Network trained on trimmed inputs largely resilient to pileup, performance decreases slightly at higher pileup when inputs are trimmed

p_T dependence under pileup and trimming effects

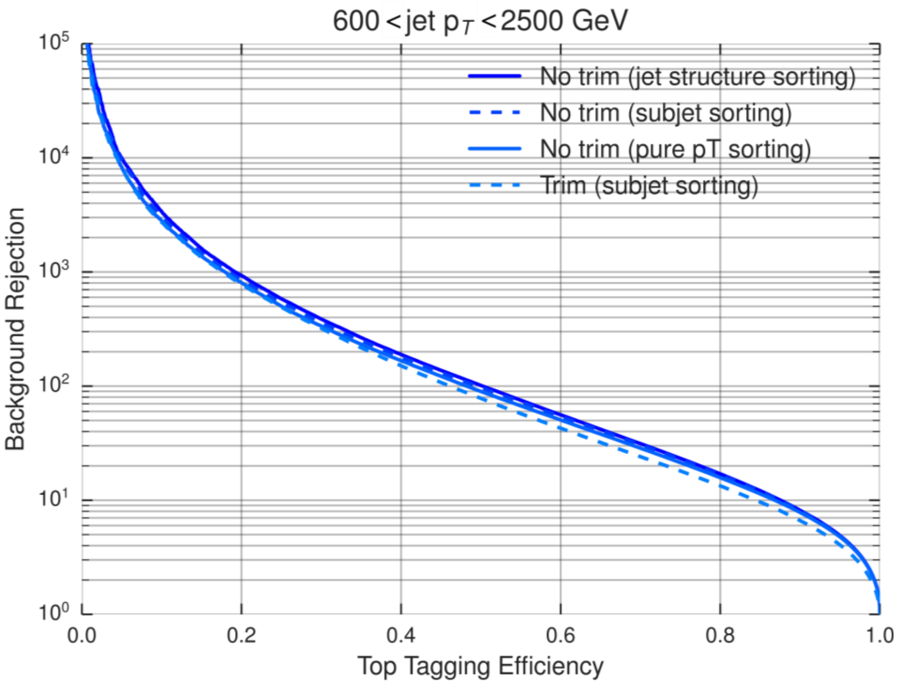


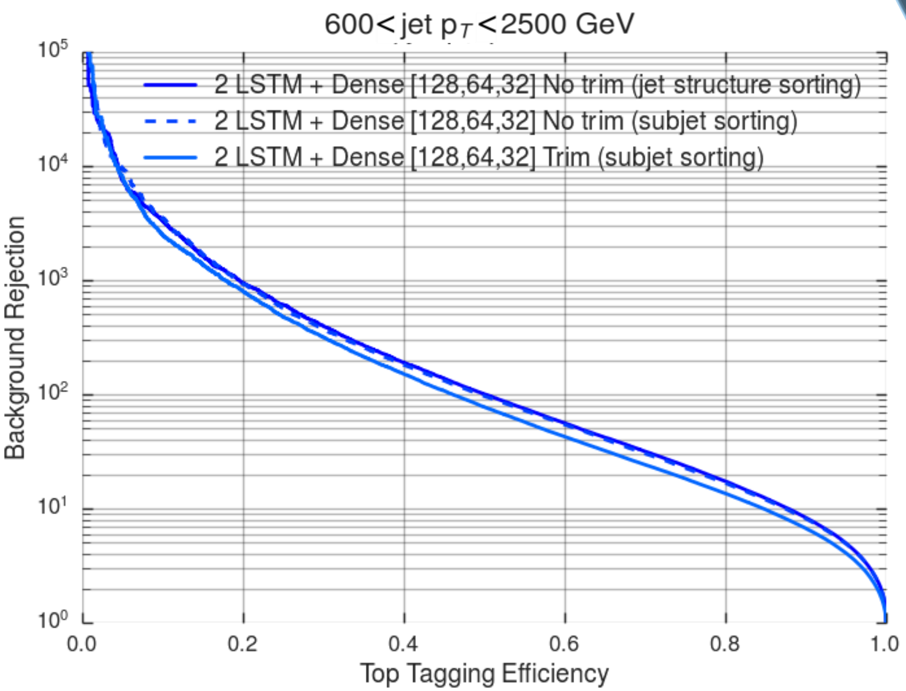
- Trimming increases resiliency to performance decrease at high p_T

p_T dependence under pileup and trimming effects



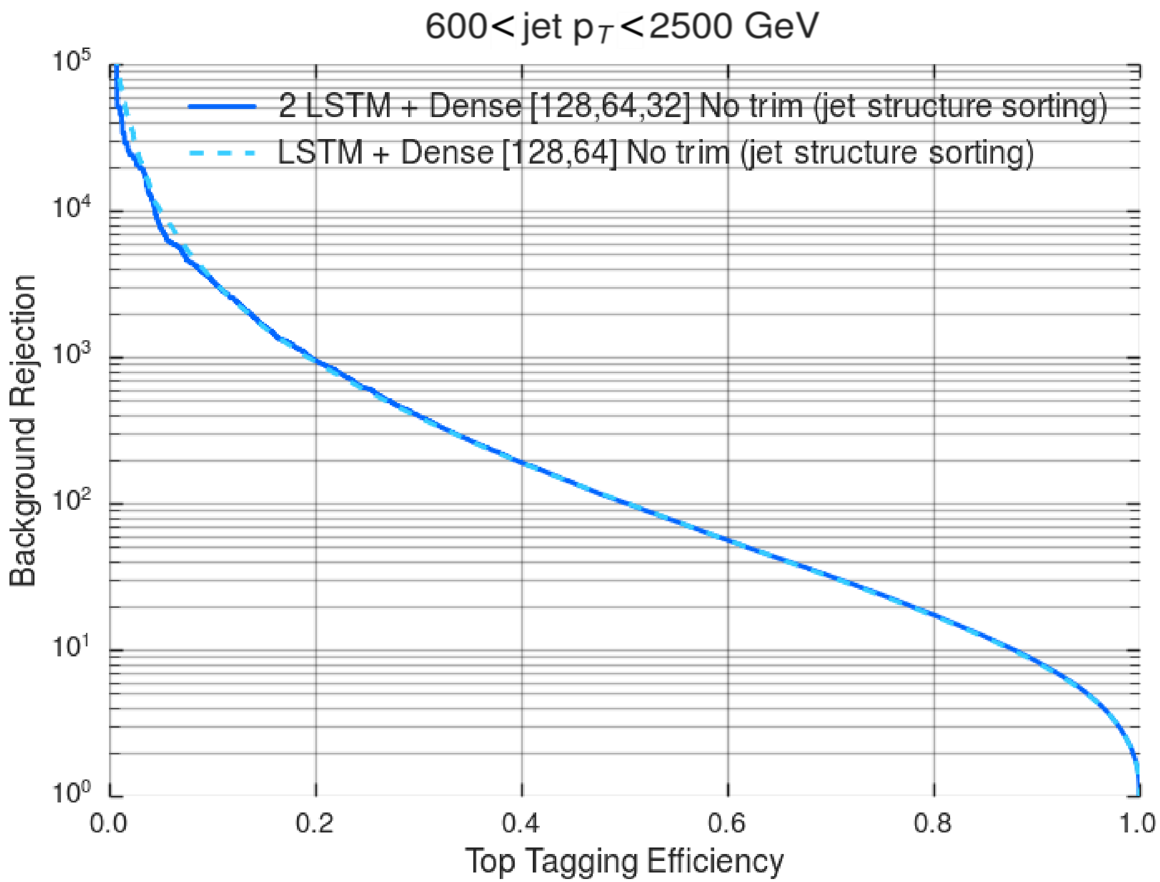
Other LSTM Architectures





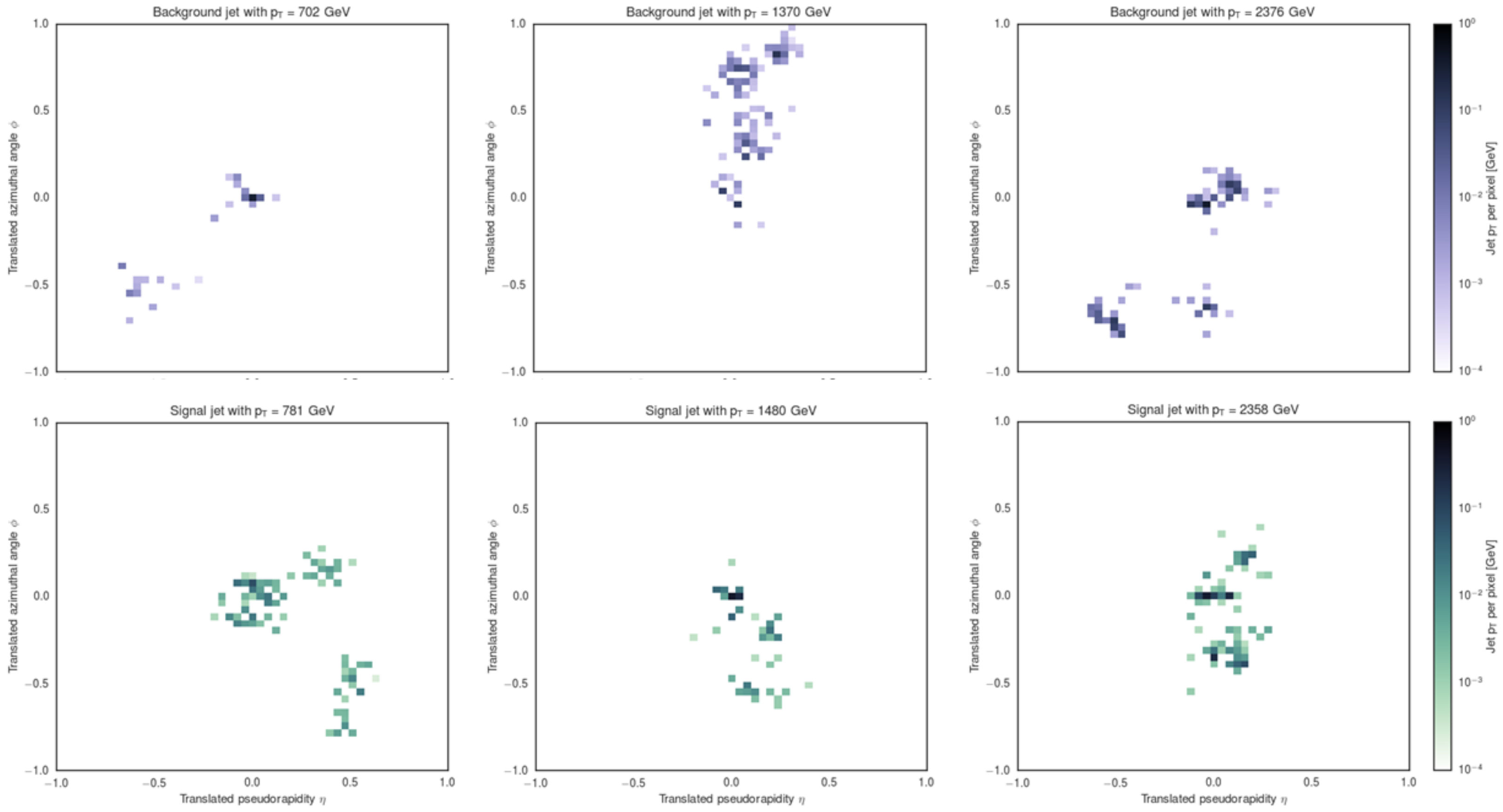
- Both architectures show very similar trends with respect to sorting methods

Other LSTM Architectures



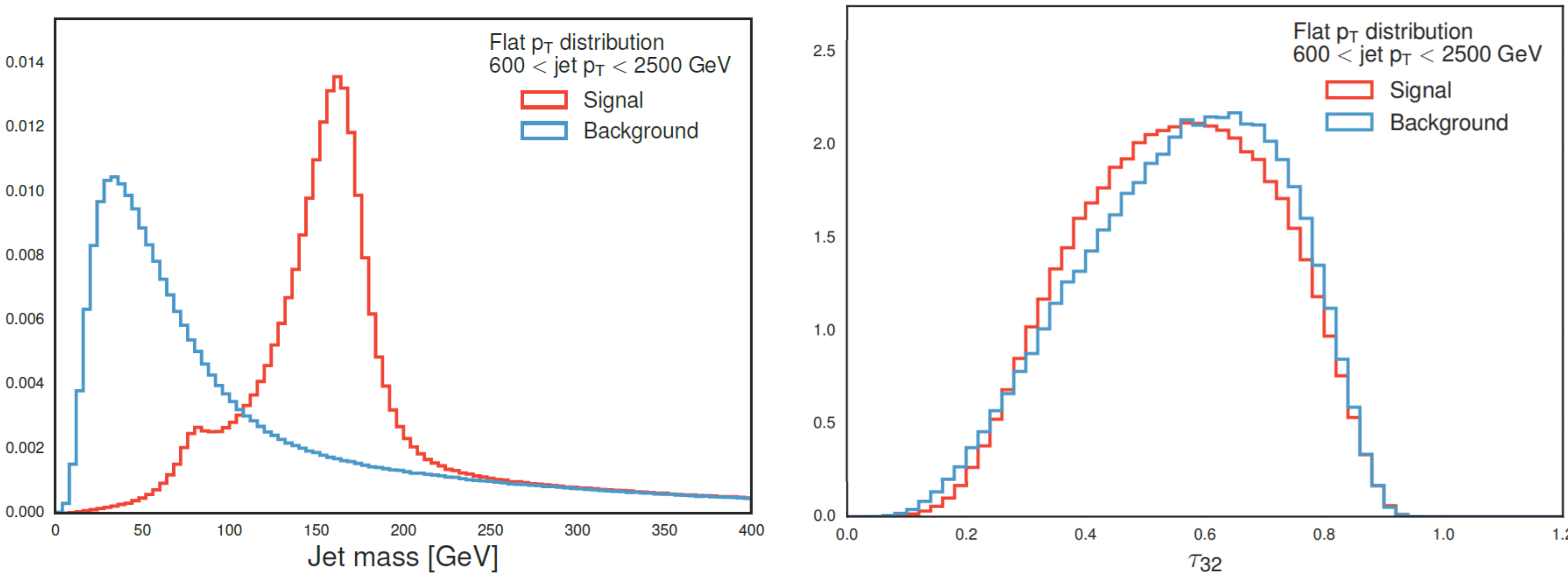
- Adding a second LSTM layer has minimal effect on performance, but makes training much more time-consuming

Drawbacks of jet images



- Jet images are largely sparse in eta-phi space and are not easily distinguishable by eye

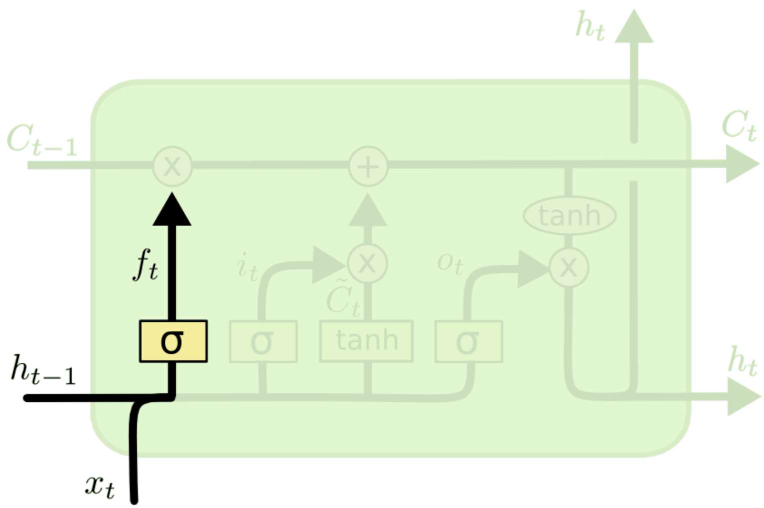
Learned features



- Jet mass (left) and τ_{32} (right) distributions for signal and background tagged jets (DNN)

LSTM Walkthrough

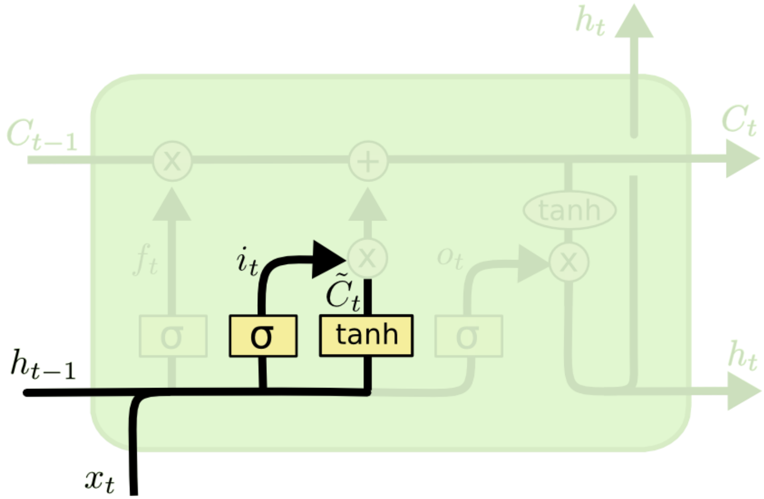
Forget gate



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

LSTM Walkthrough

Input gate, cell gate

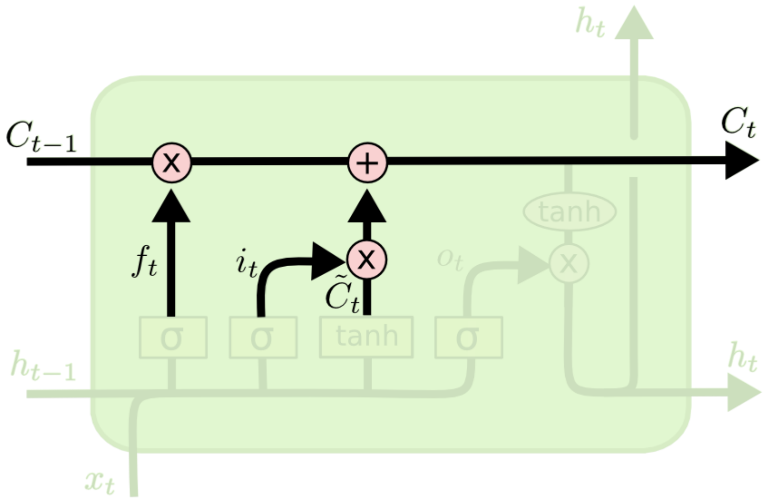


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

LSTM Walkthrough

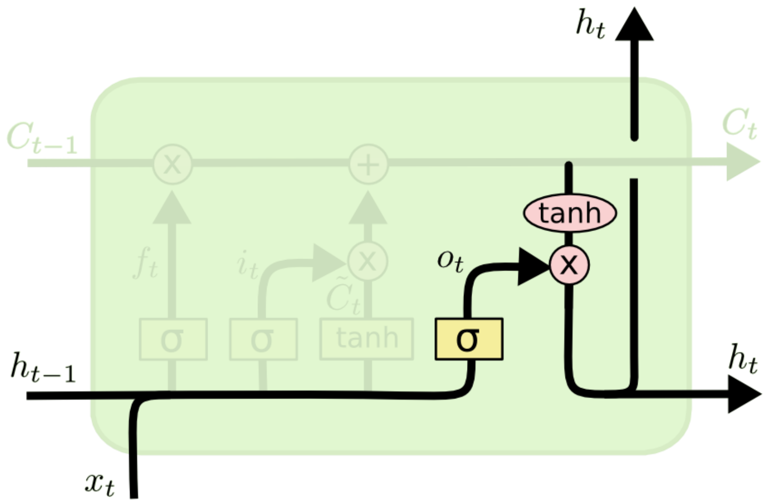
Cell state



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

LSTM Walkthrough

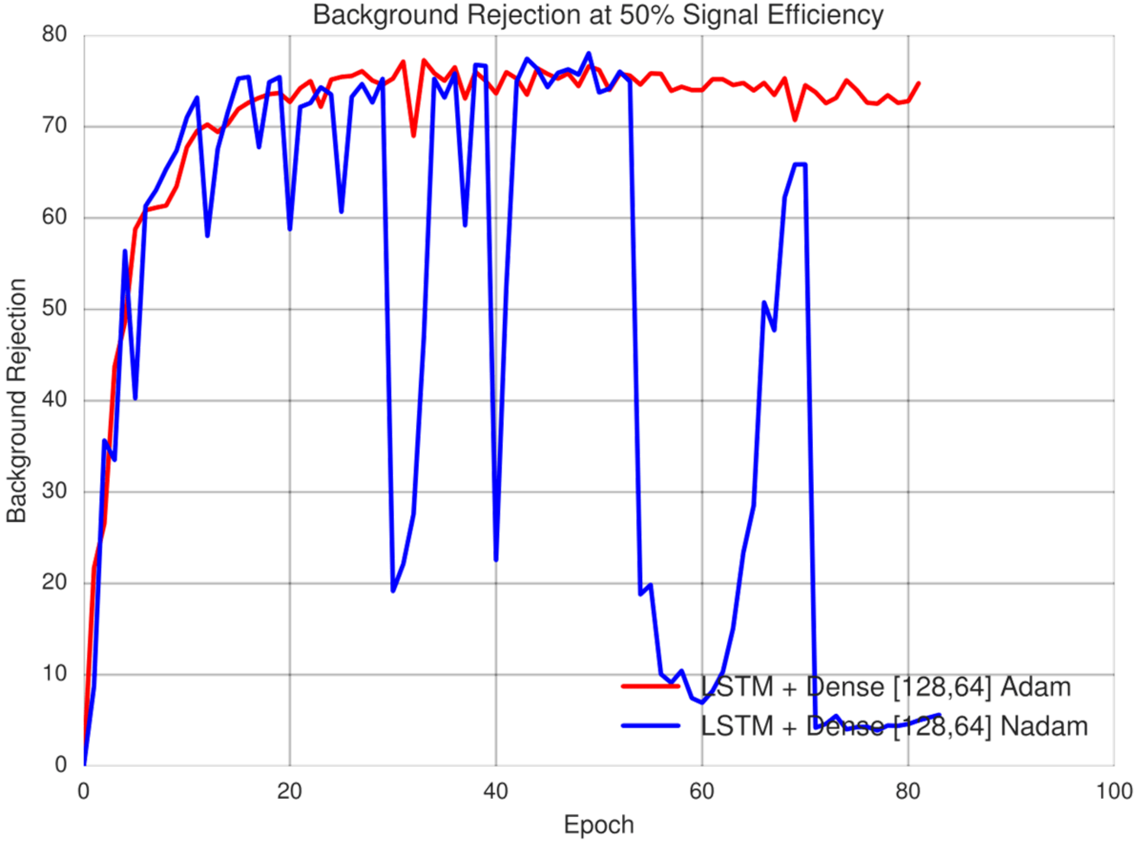
Output gate, output



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Training instability



- The chosen optimizer can have major impacts on learning stability