LORENTZ GROUP NETWORK

LORENTZ GROUP EQUIVARIANT NEURAL NETWORK

Alexander Bogatskiy, Brandon Anderson, Risi Kondor, David W. Miller, Jan T. Offermann, Marwah Roussi

Fermilab — October 15,2020

THE UNIVERSITY OF CHICAGO

Center for Data and Computing
AT THE UNIVERSITY OF CHICAGO

FLATIRON INSTITUTE

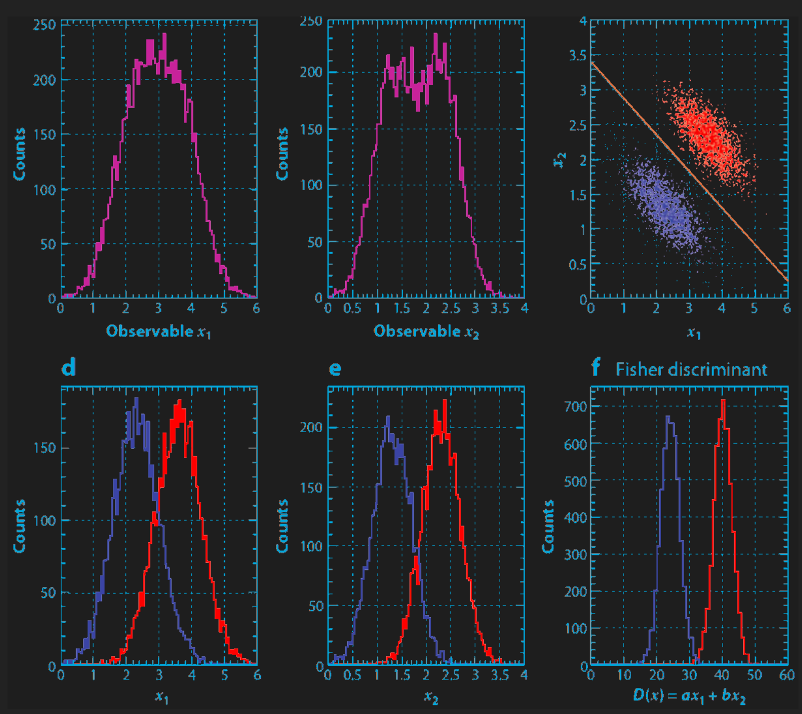
OVERVIEW

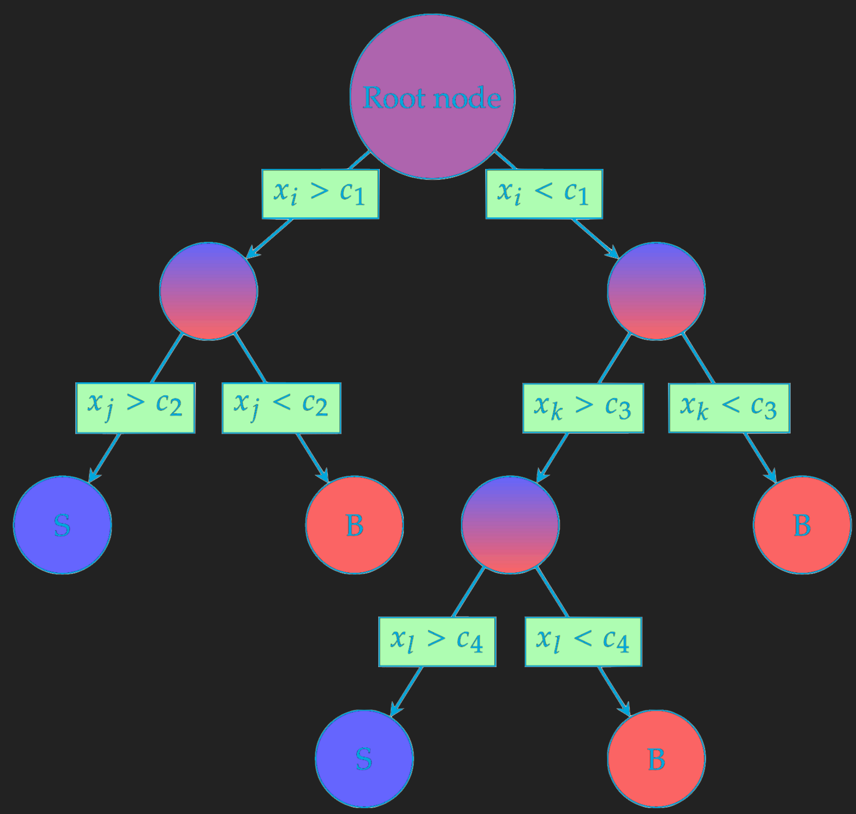

p_1^μ
p_{31}
p_3^μ
p_{12}
p_{23}
p_2^μ
Scal
W_{in}
p^2
f
$\mathcal{F}_1$
$\mathcal{F}_2$
$\mathcal{F}_3$
id
$\otimes 2$
$\sum$
CG
P_{cov}
P_{inv}
MLP
W_{irr}
$\mathcal{F}_1'$
$\mathcal{F}_2'$
$\mathcal{F}_3'$
W_{out}
(w_-, w_+)

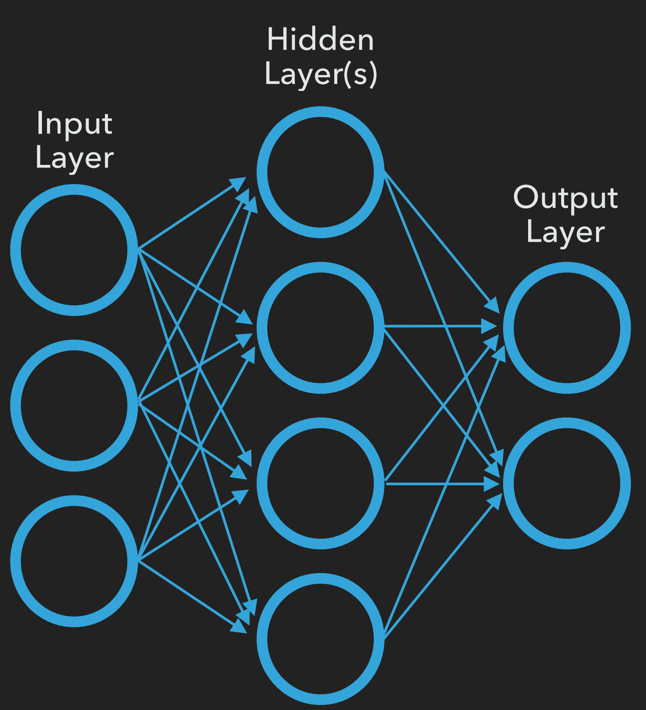
MACHINE LEARNING IN HIGH-ENERGY PHYSICS

- ML has a long history of use in HEP.

MULTIVARIATE METHODS → **BOOSTED DECISION TREES** → **NEURAL NETWORKS**







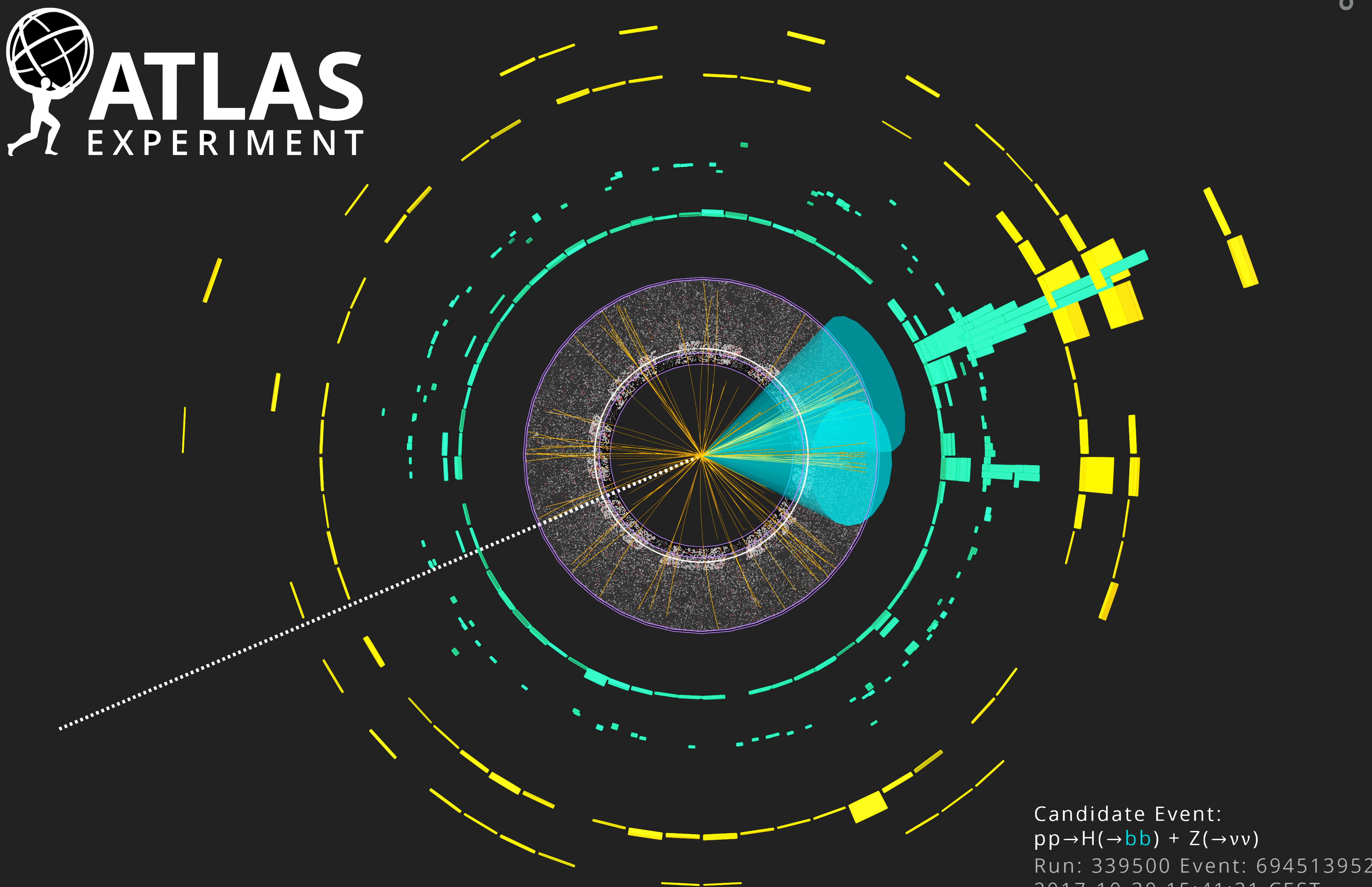
MACHINE LEARNING IN HIGH-ENERGY PHYSICS

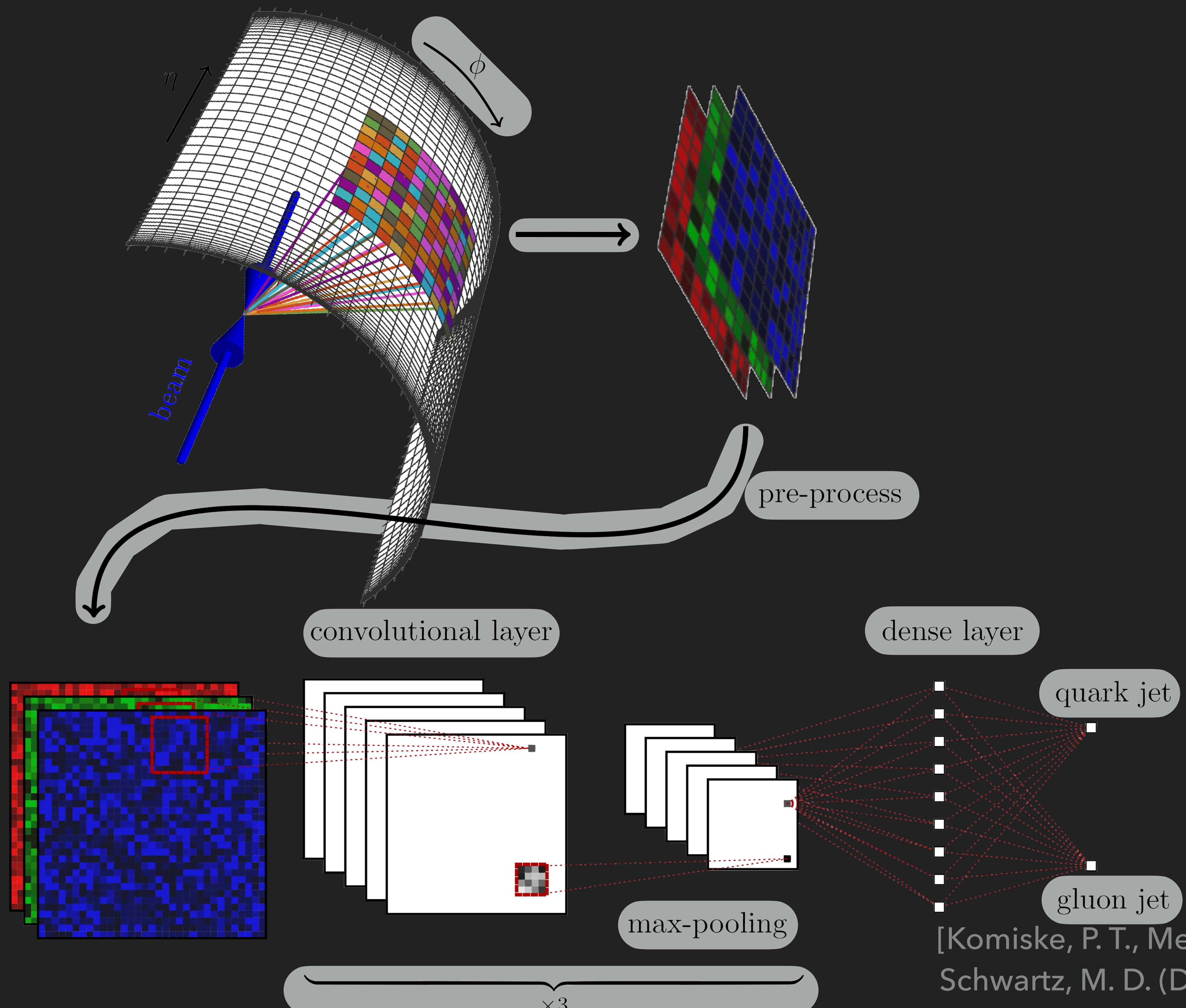
- ML has a long history of use in HEP.

- This story is over-simplified.
  - Neural networks have been used in HEP since the late 1980's (track-finding, classification).
  - Multivariate methods and BDT's are still in use.

[Bhat, P. (History of Machine Learning for High Energy Physics, 2018)]

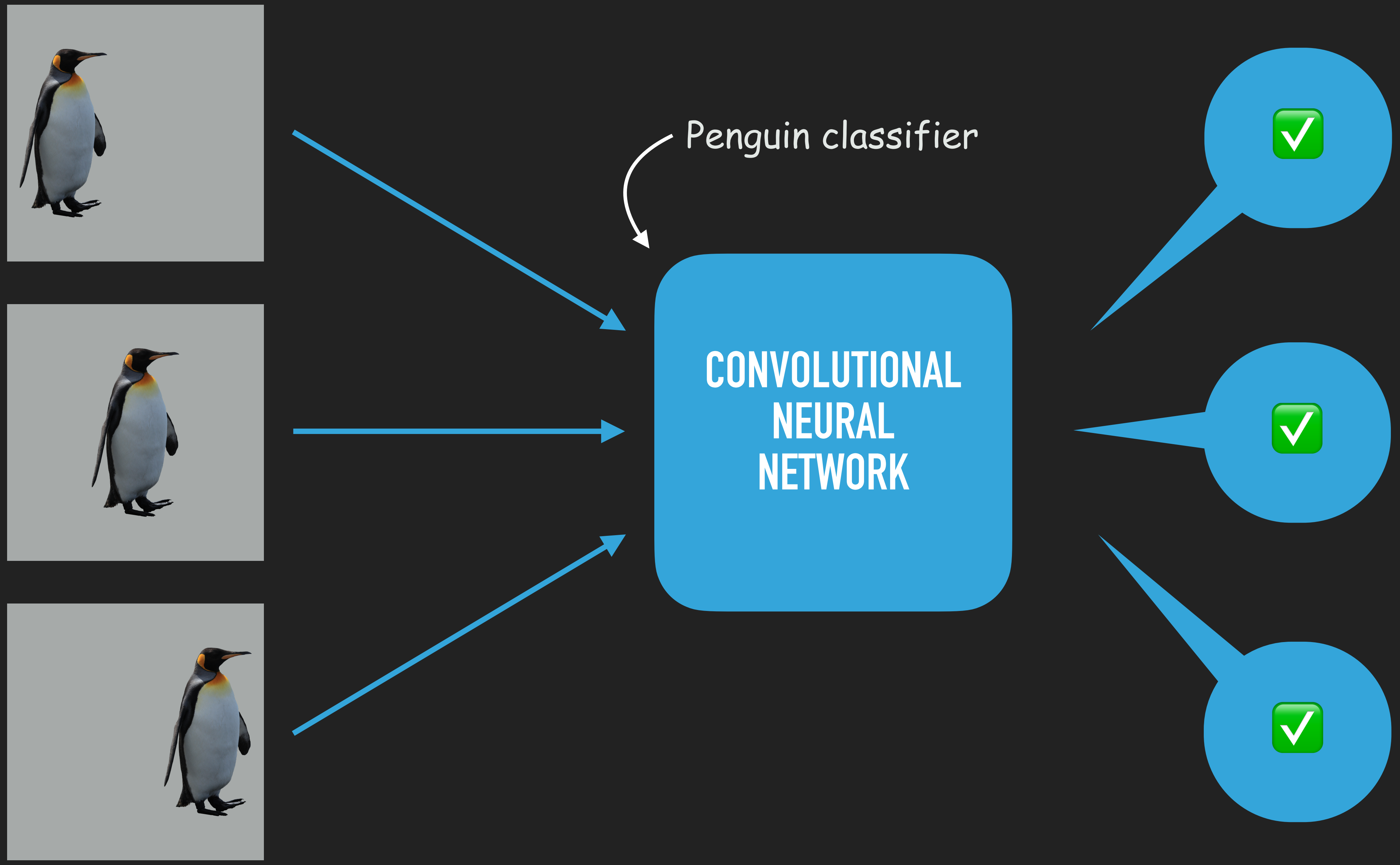
- HEP is replete with examples and uses of neural networks.
- One common task – our focus for today – is jet tagging.
  - Some approaches use out-of-the-box methods from other fields, e.g. image recognition.
  - Others use more *physics-inspired* architectures.


ATLAS
EXPERIMENT
Candidate Event:
pp→H(→bb) + Z(→νν)
Run: 339500 Event: 694513952
2017-10-30 15:41:21 CEST



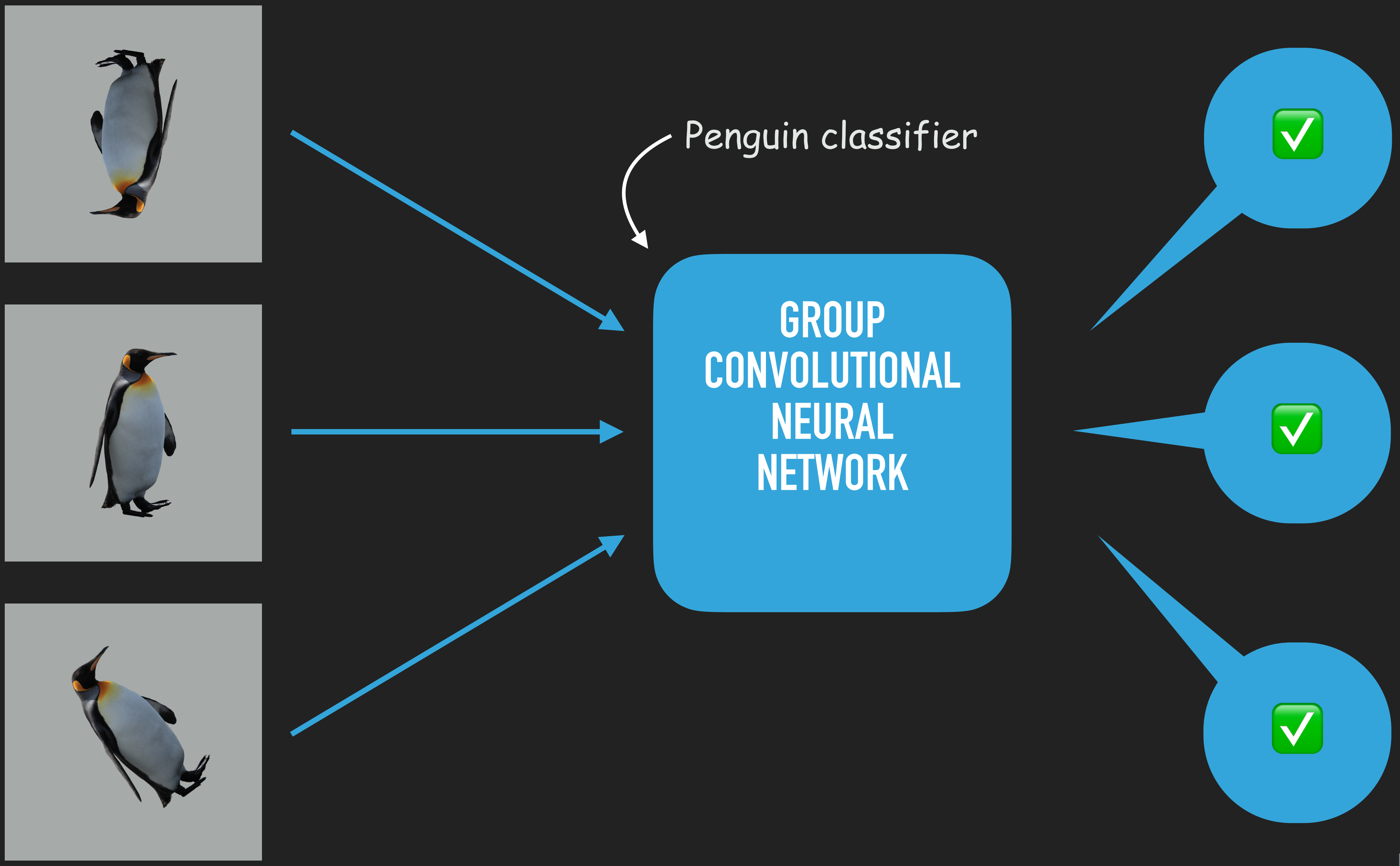
[Komiske, P. T., Metodiev, E. M., Schwartz, M. D. (Deep learning in color, 2016)]

- Convolutional neural networks allow one to take advantage of symmetries in image-recognition.



Translational Symmetry

- Convolutional neural networks allow one to take advantage of symmetries in image-recognition.

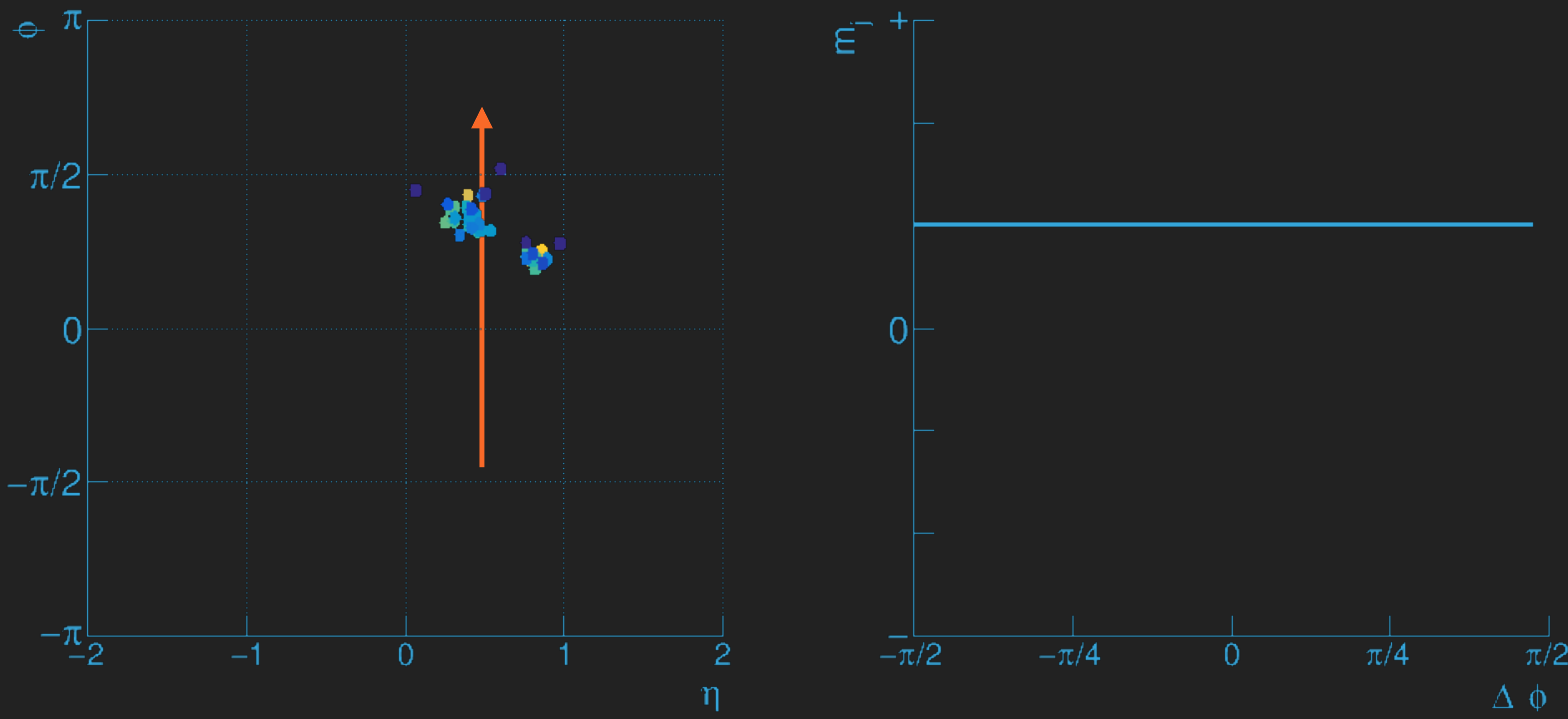


Rotational Symmetry

[Cohen, T., Welling, M. (Group Equivariant Convolutional Networks)]

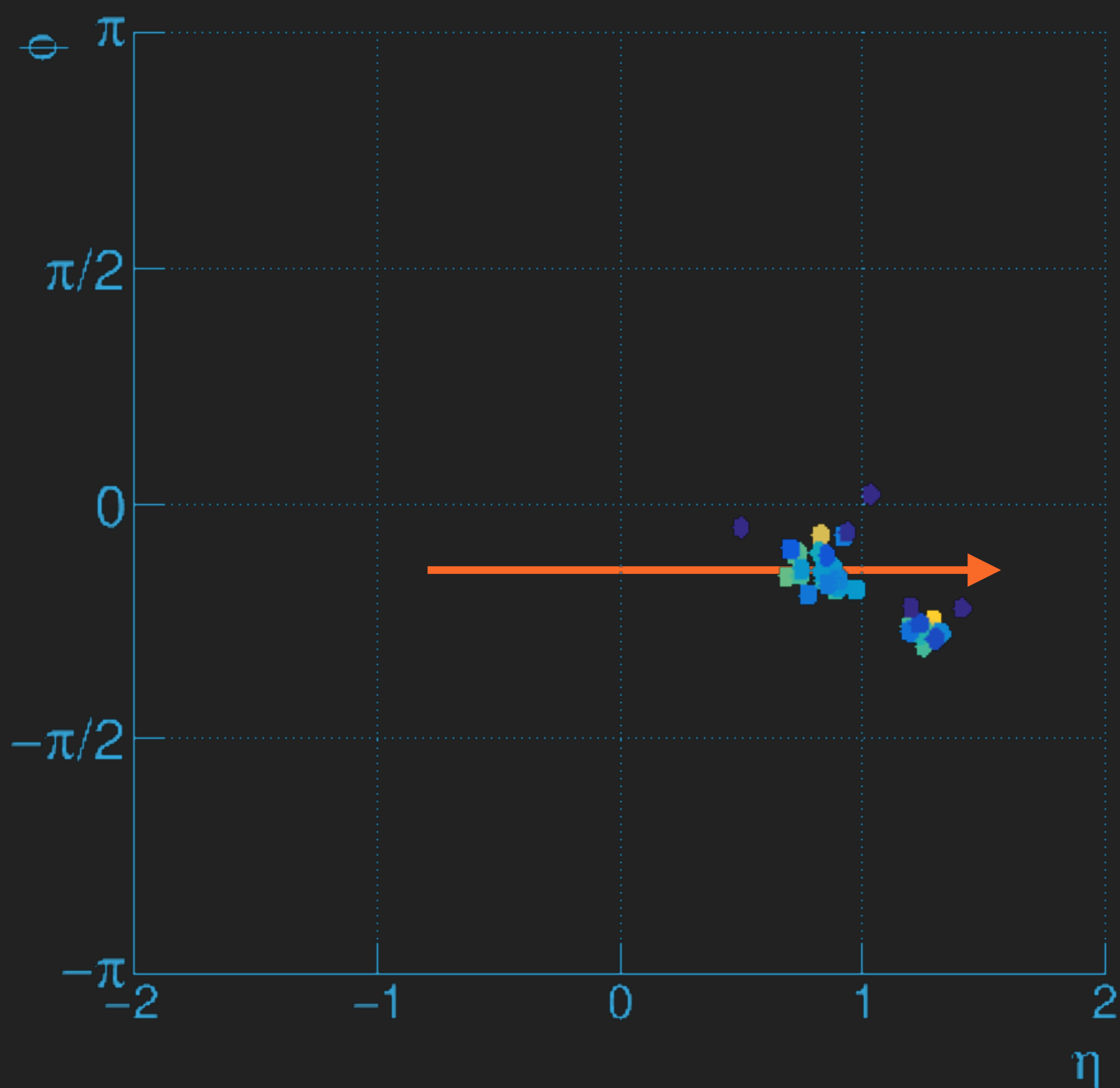
BUT WAIT…

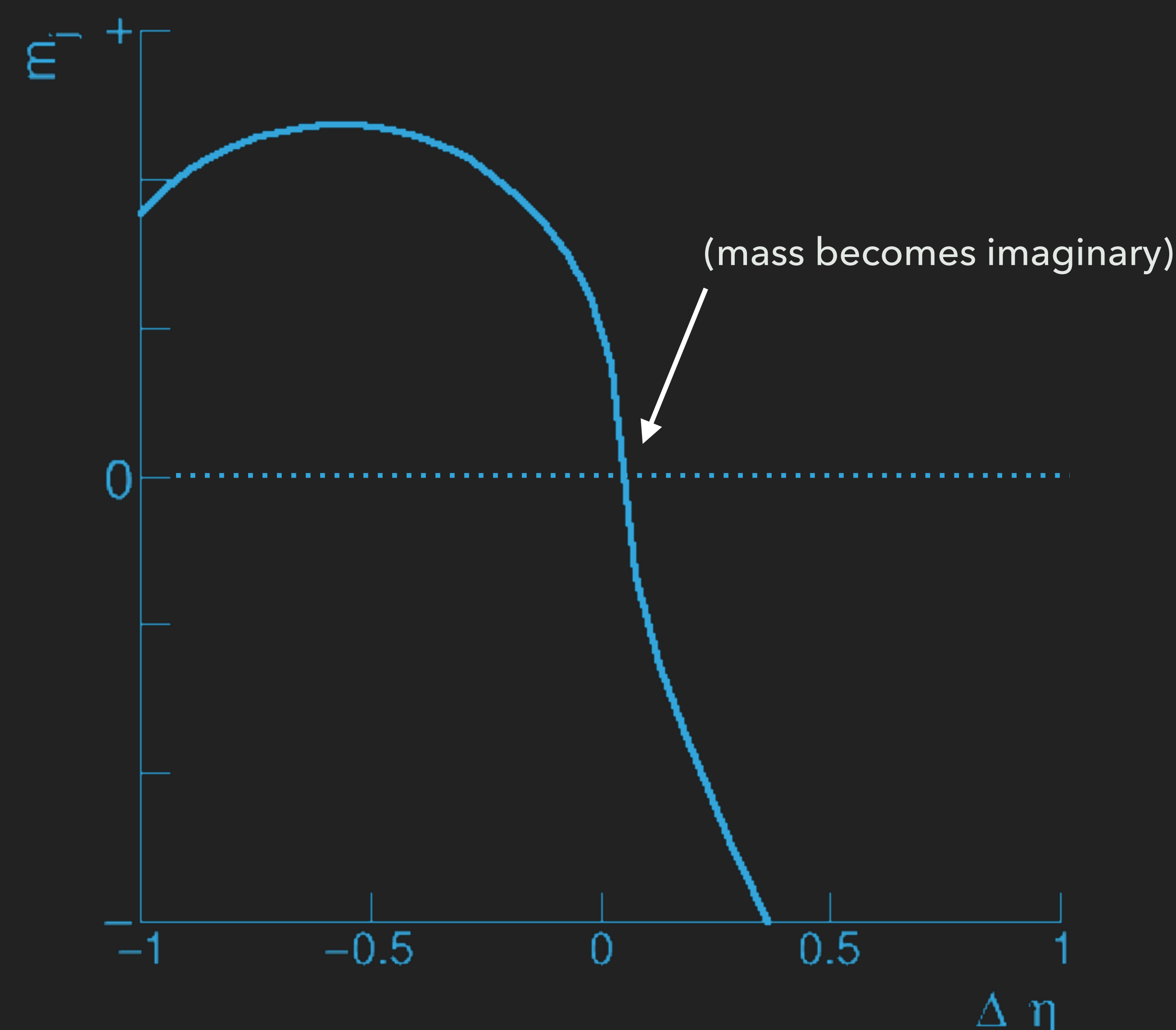
- What if the data does *not* exhibit these symmetries?
- Consider "jet images" – projections of jet constituents onto (η, ϕ).
  - Let's fix E and p_T, and transform the images in the (η, ϕ) plane.



BUT WAIT…

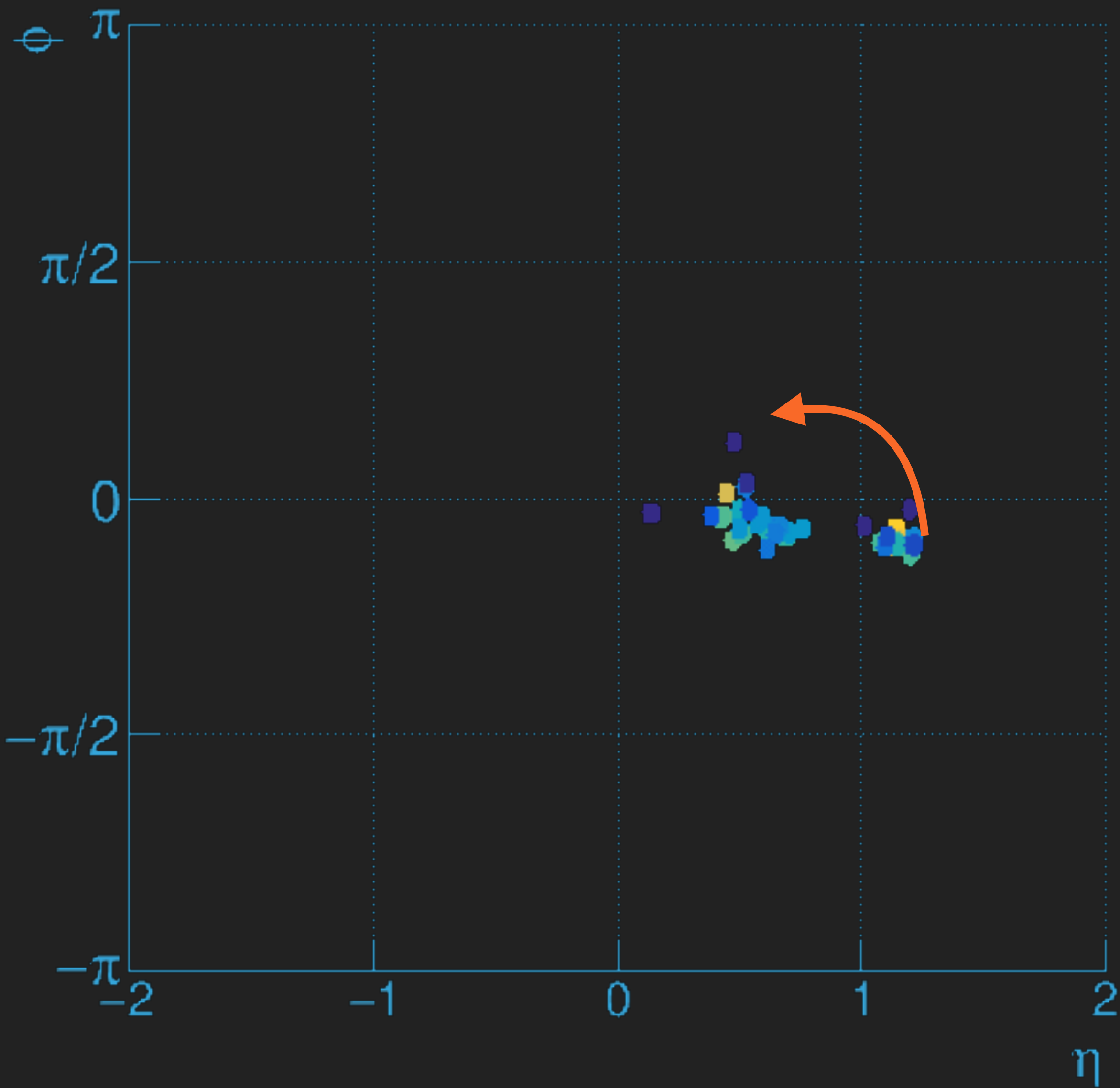
- What if the data does *not* exhibit these symmetries?
- Consider "jet images" – projections of jet constituents onto (η, ϕ).
  - Let's fix E and p_T, and transform the images in the (η, ϕ) plane.

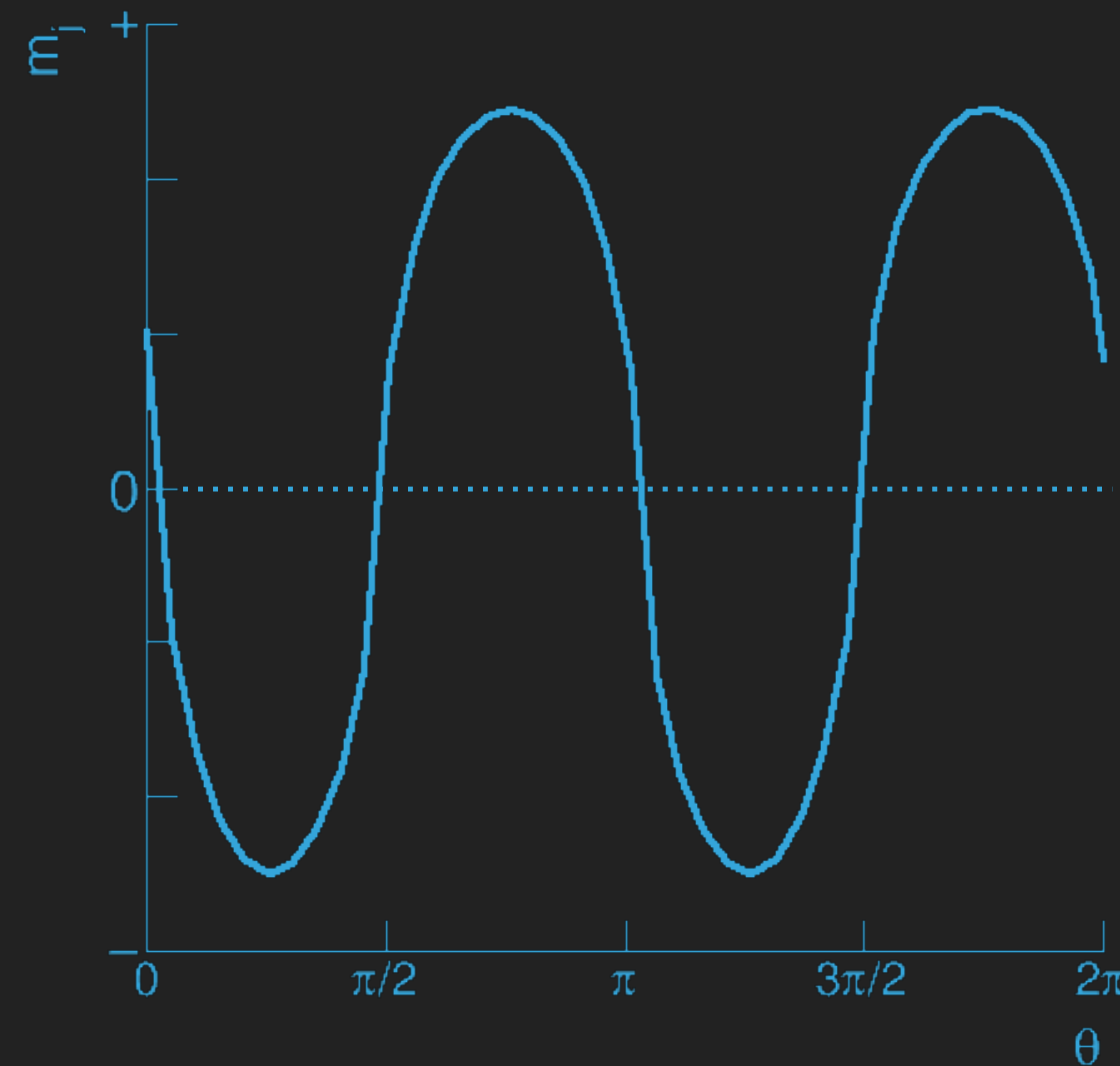




BUT WAIT…

- What if the data does *not* exhibit these symmetries?
- Consider "jet images" – projections of jet constituents onto (η, ϕ).
  - Let's fix E and p_T, and transform the images in the (η, ϕ) plane.





- HEP is replete with examples and uses of neural networks.
- One common task – our focus for today – is jet tagging.
  - Some approaches use out-of-the-box methods from other fields, e.g. image recognition.
  - Others use more *physics-inspired* architectures.

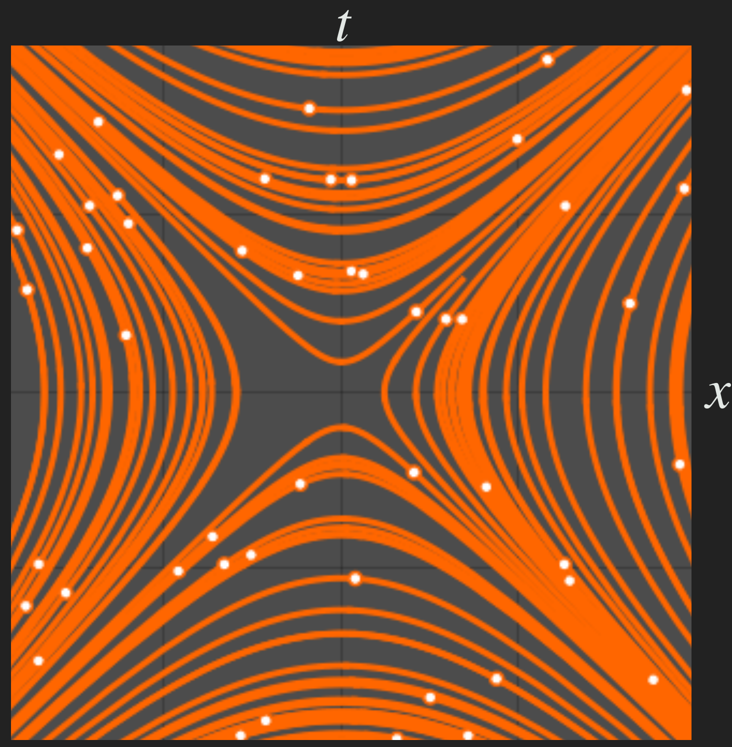
SOME NOTABLE EXAMPLES

- Lorentz Layer[1]
  - Network layer explicitly calculates *Lorentz invariants* m^2, p_T etc. from some input p^μ.
- Particle/Energy Flow Networks[2]
  - Construct observables as some $F\left(\sum_{i=1}^{M}(z_i)\Phi(\hat{p}_i)\right)$.
- Lorentz Boost Networks[3]
  - Lorentz-boosts into momenta's rest frames to extract features (to feed into a deep neural network).

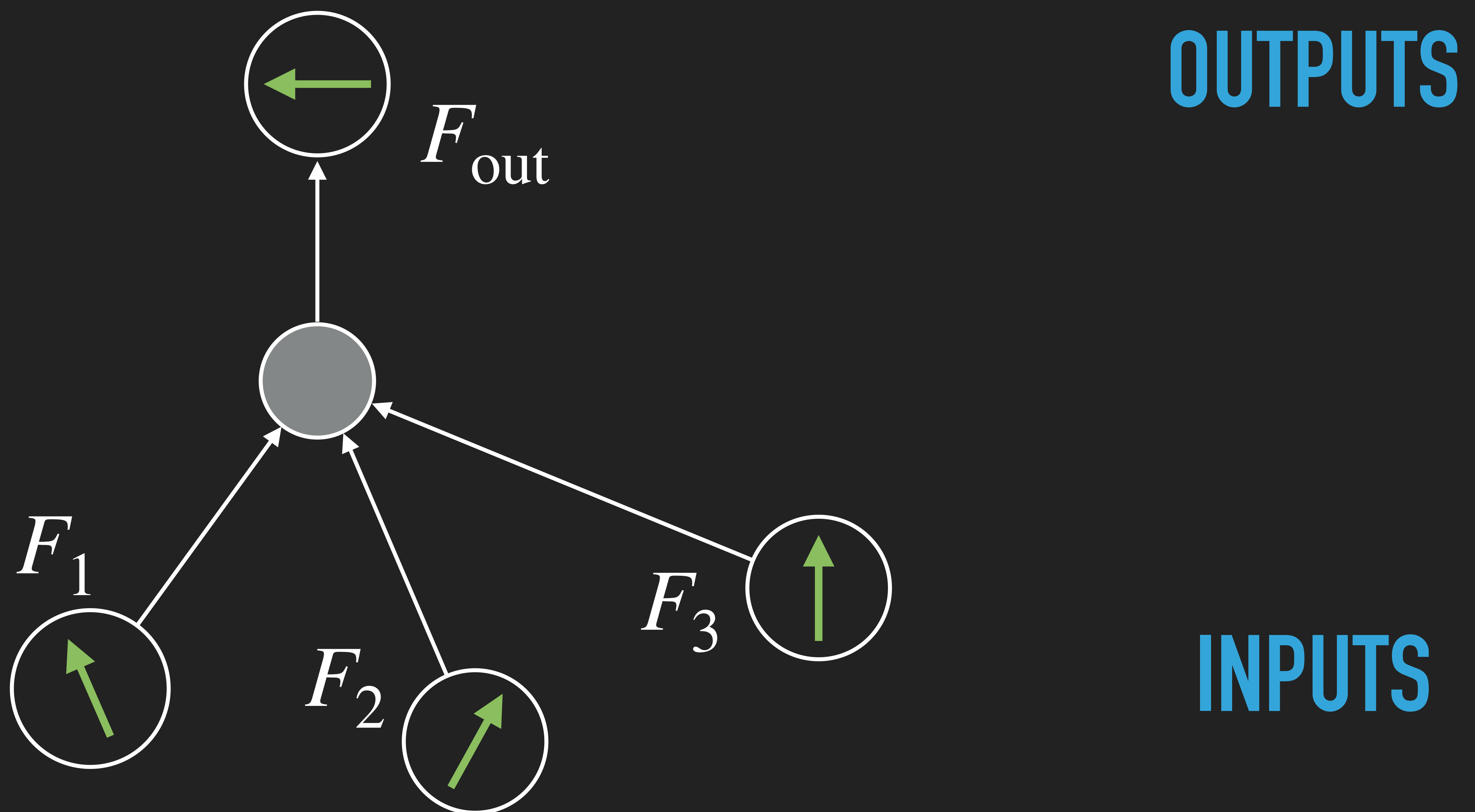
[Butter, A., Kasieczka, G., Plehn, T., Russell, M. (LoLa – Lorentz Layer, 2018)]
[Thaler, J., Komiske, P. T., Metodiev, E. M. (Energy/Particle Flow Networks, 2018)]
[Erdmann, M., Geiser, E., Rath, Y., Rieger, M. (Lorentz Boost Networks, 2018)]

LGN: THE MOTIVATING IDEA

- We wish to construct a network *equivariant* under action by members of the Lorentz group.
- Similar in spirit to image identification, but built from the correct symmetry group for the problem: $SO(1,3)^+$ vs $SO(3)$.



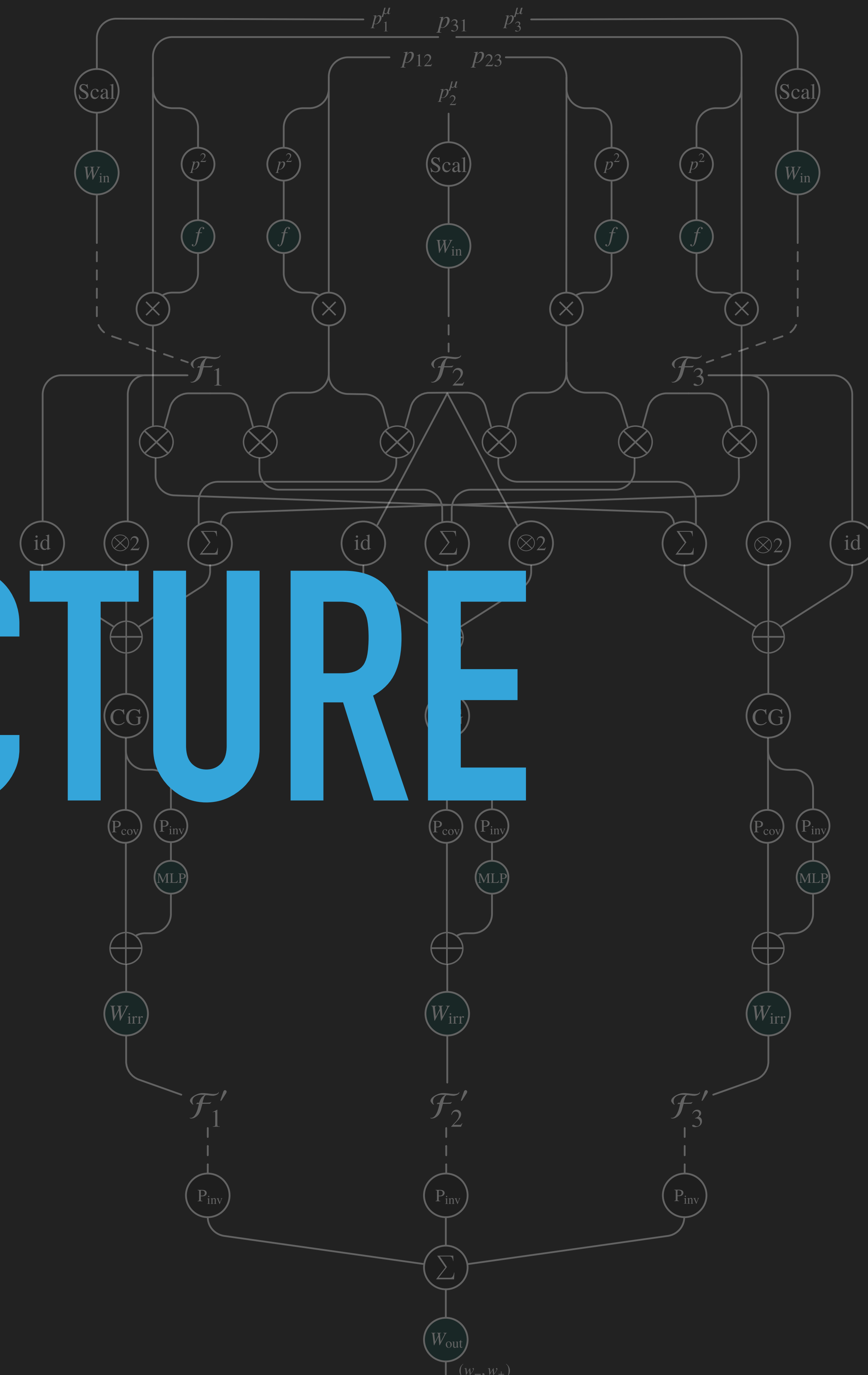
$$F_{\text{out}} \to \rho_{\text{out}}(g) F_{\text{out}}$$



$$F_1 \to \rho_1(g) F_1 \qquad F_2 \to \rho_2(g) F_2 \qquad F_3 \to \rho_3(g) F_3$$

[credit to Alex Bogatskiy]

ARCHITECTURE



- Input: N particles' 4-momenta p_i^μ.
- N activations $\mathscr{F}_i$ at each level live in representations of the Lorentz group.
- The update rule involves pair interactions.

permutation invariance

$p_{ij} \equiv p_i - p_j$

$$\mathscr{F}_i \mapsto W \cdot \left(\mathscr{F}_i \oplus \mathscr{F}_i^{\otimes 2} \oplus \sum_j f\left(p_{ij}^2\right) \cdot \mathscr{Y}_{ij}(p_{ij}) \otimes \mathscr{F}_j \right)$$

self-interaction

zonal harmonics

learnable scalar function

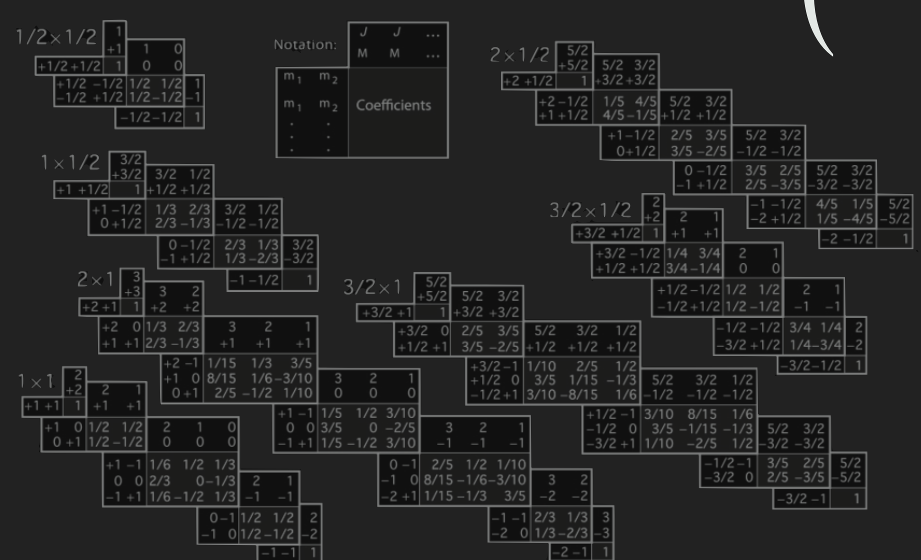
Level 0: $$p_i \mapsto W \cdot \left(p_i \oplus p_i^{\otimes 2} \oplus \sum_j f\left(p_{ij}^2\right) \cdot p_{ij} \otimes p_j \right)$$

- Arbitrary traditional sub-networks can be applied to Lorentz invariants.
- Output layer sums over i and projects onto invariants.(or other irrep).

- Input: N particles' 4-momenta p_i^μ.
- N activations $\mathscr{F}_i$ at each level live in representations of the Lorentz group.
- The update rule involves pair interactions.

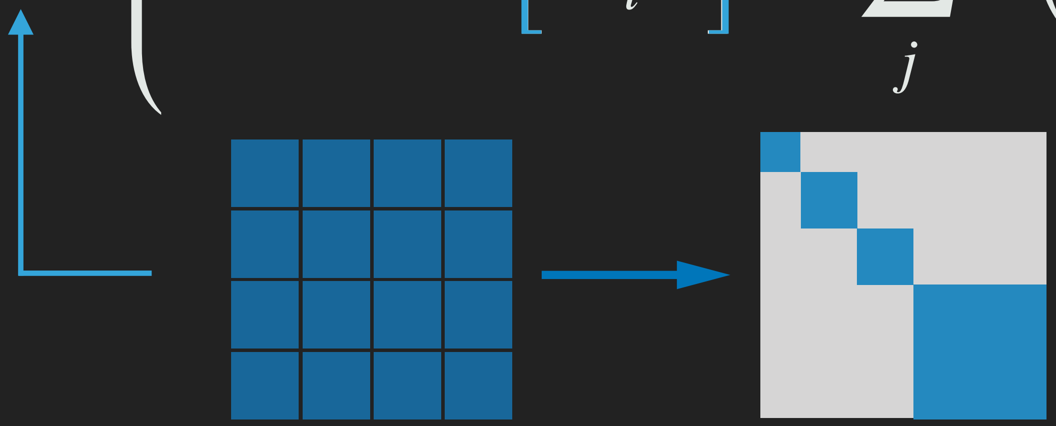
$$\mathscr{F}_i \mapsto W \cdot \left(\mathscr{F}_i \oplus \mathrm{CG}\left[\mathscr{F}_i^{\otimes 2}\right] \oplus \sum_j f\left(p_{ij}^2\right) \cdot \mathrm{CG}\left[p_{ij} \otimes \mathscr{F}_j\right] \right)$$

Clebsch-Gordan decompositions



- Finite-dimensional representations of the Lorentz group are decomposable.
- We decompose the activations via Clebsch-Gordan decompositions.

$$\mathscr{F}_i \mapsto W \cdot \left(\mathscr{F}_i \oplus \mathrm{CG}\left[\mathscr{F}_i^{\otimes 2}\right] \oplus \sum_j f\left(p_{ij}^2\right) \cdot \mathrm{CG}\left[p_{ij} \otimes \mathscr{F}_j\right] \right)$$



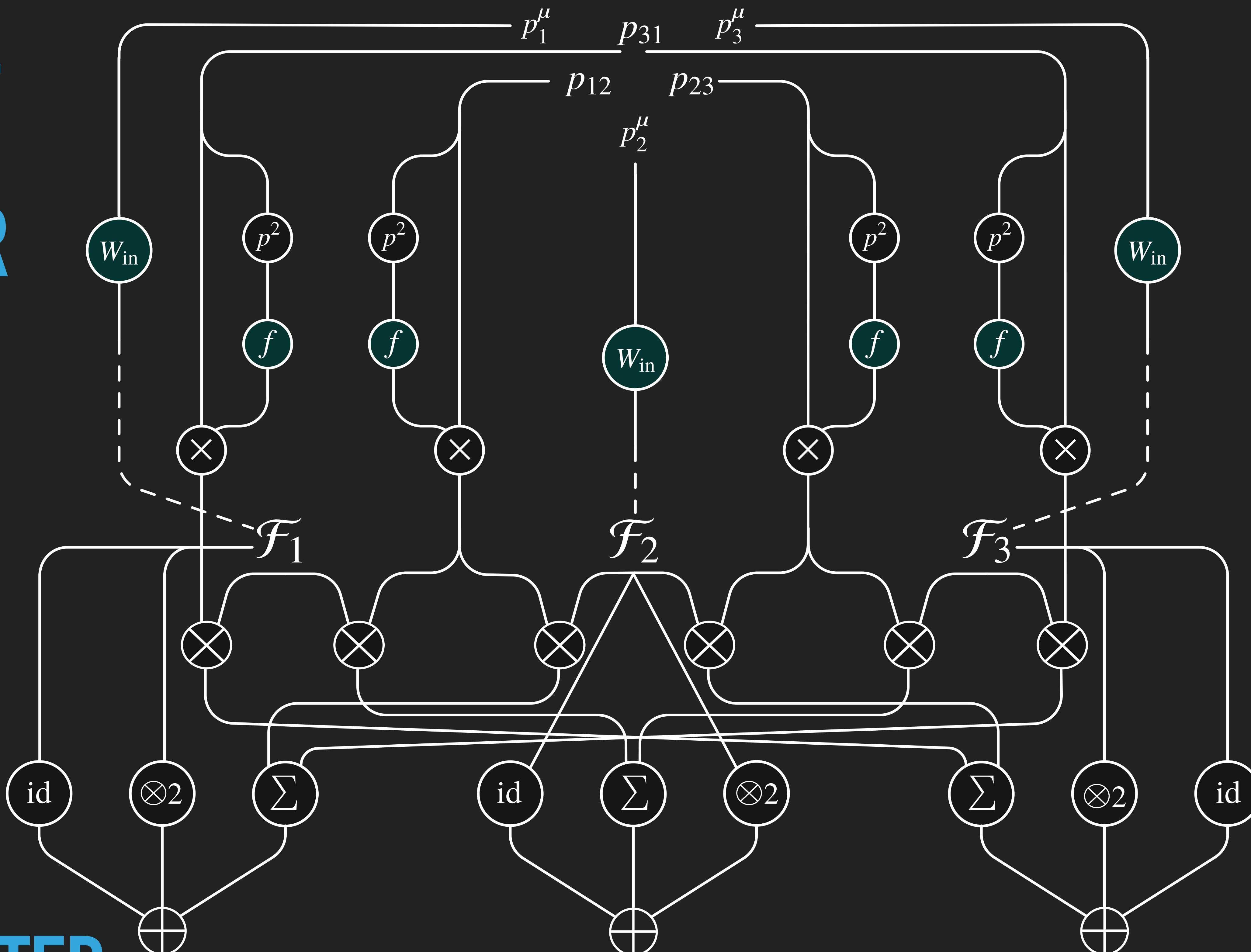
- We require W, our linear operation, to observe Lorentz equivariance.
- As a consequence of Schur's lemma, W acts as a scalar multiplication on each irrep, and only linearly combines vectors of the same weight.

> Let U and V be completely reducible representations of G:
> $$V = \oplus_\alpha R_\alpha^{\oplus \tau_\alpha}, \quad U = \oplus_\alpha R_\alpha^{\oplus \tau'_\alpha}.$$
> Then linear equivariant map $W : V \to U$ can be parametrized by
>
> $\{W_\alpha \in \mathrm{Mat}(\tau'_\alpha, \tau_\alpha)\}$, with W_α acting on the irreps within $R_\alpha^{\oplus \tau_\alpha}$.

[Kondor, R., Trivedi, S. (On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups, 2018)]

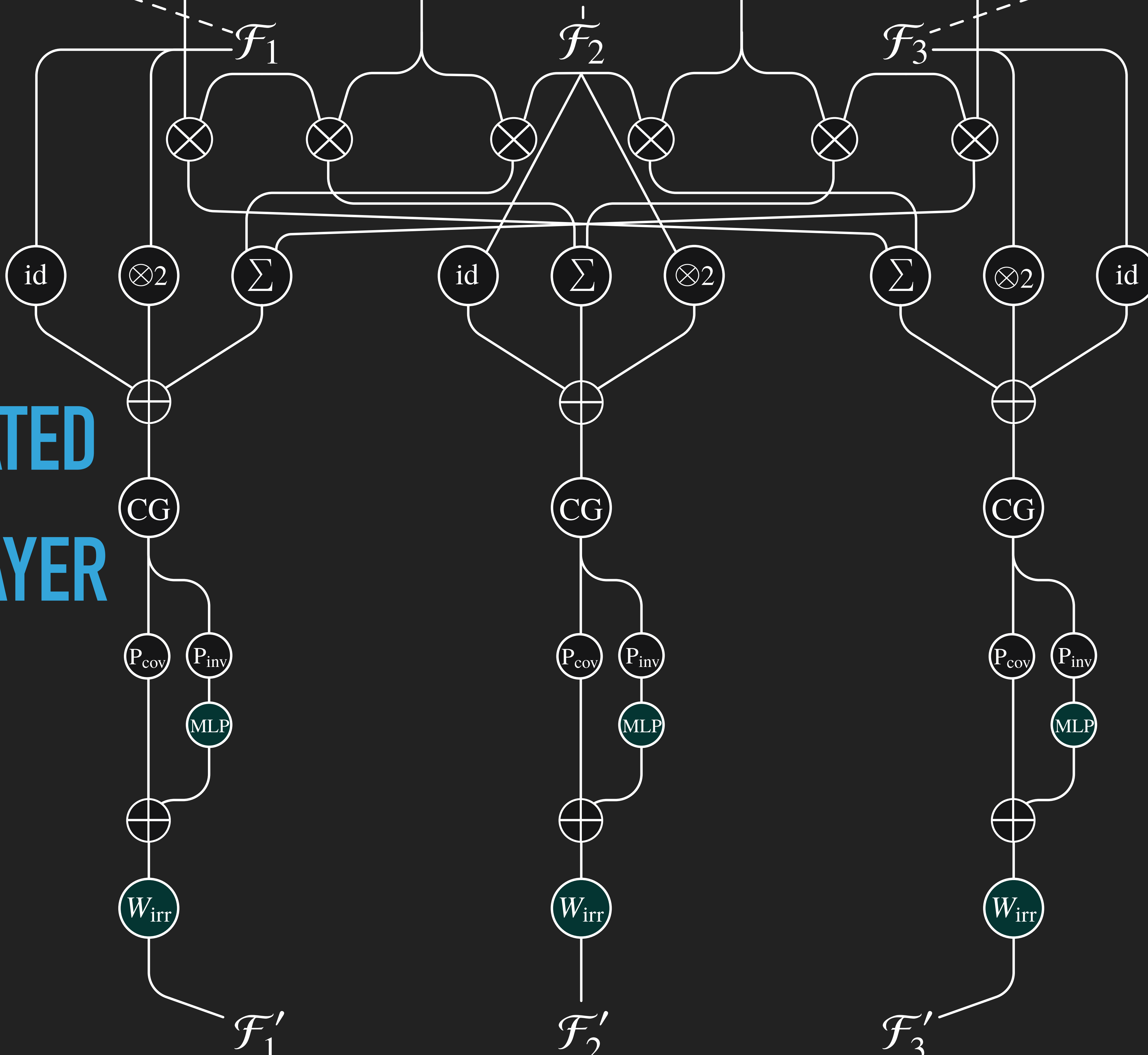
INPUT LAYER

ITERATED CG LAYER



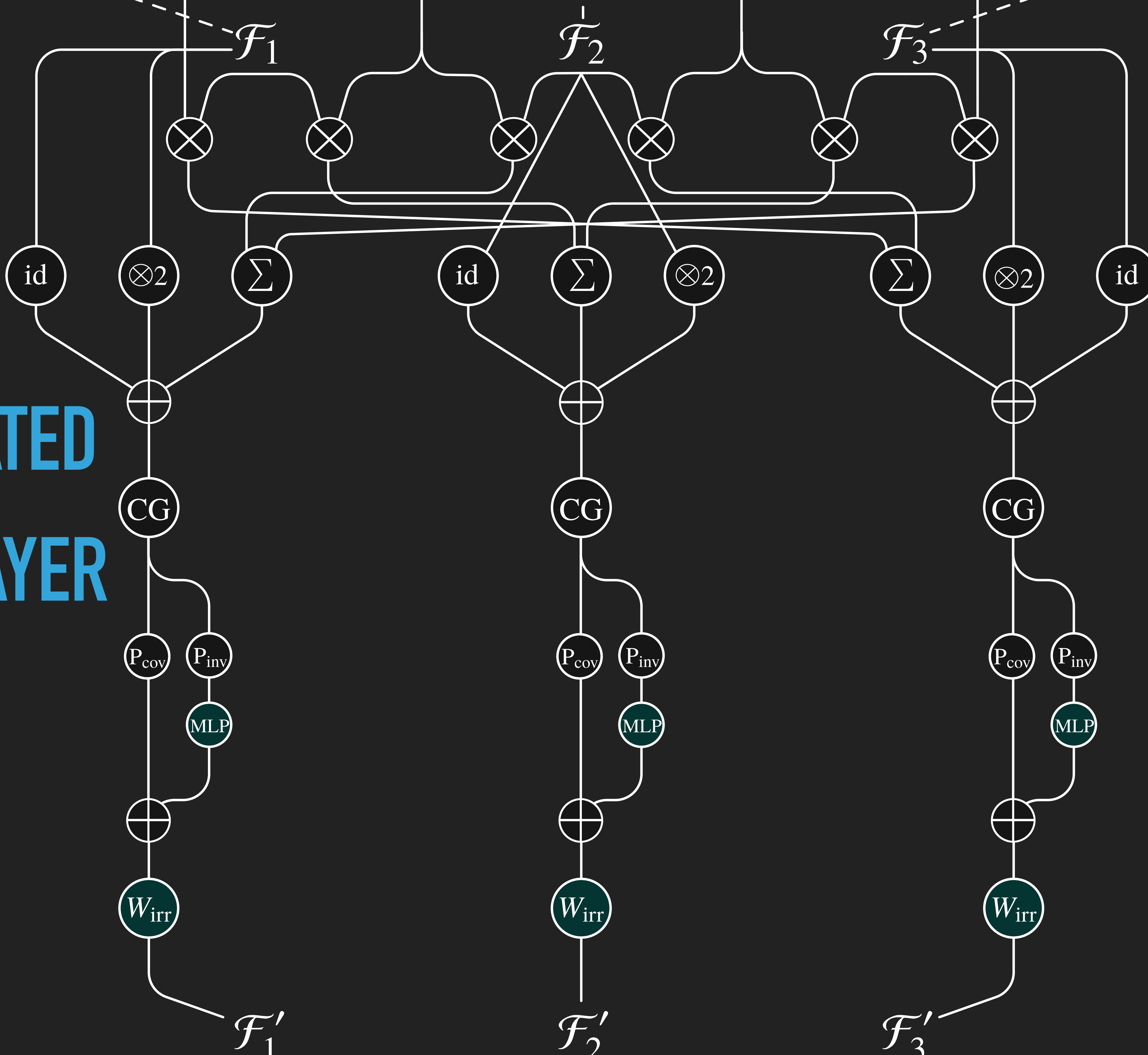
[credit to Alex Bogatskiy]

ITERATED CG LAYER



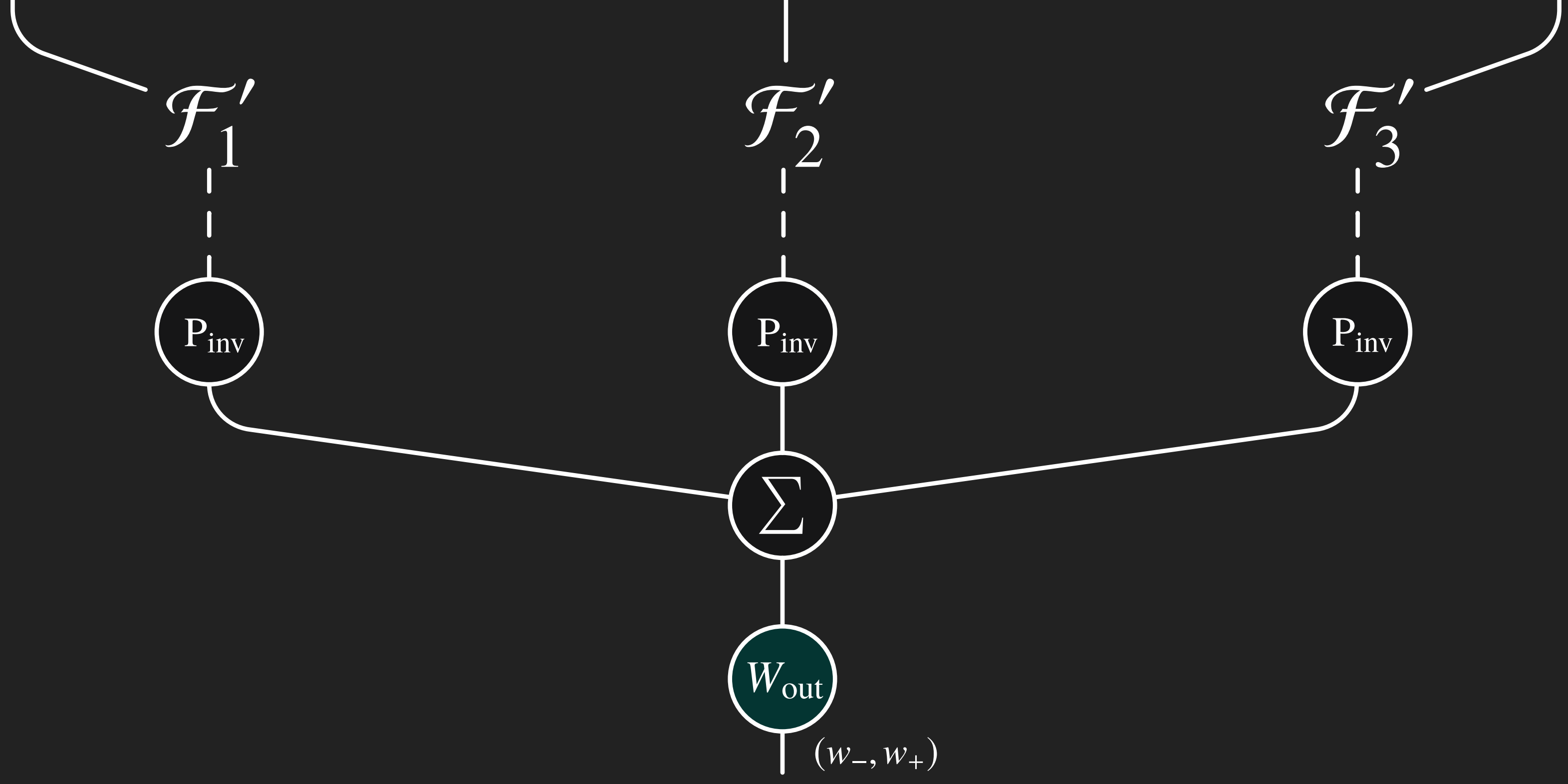
[credit to Alex Bogatskiy]

ITERATED CG LAYER



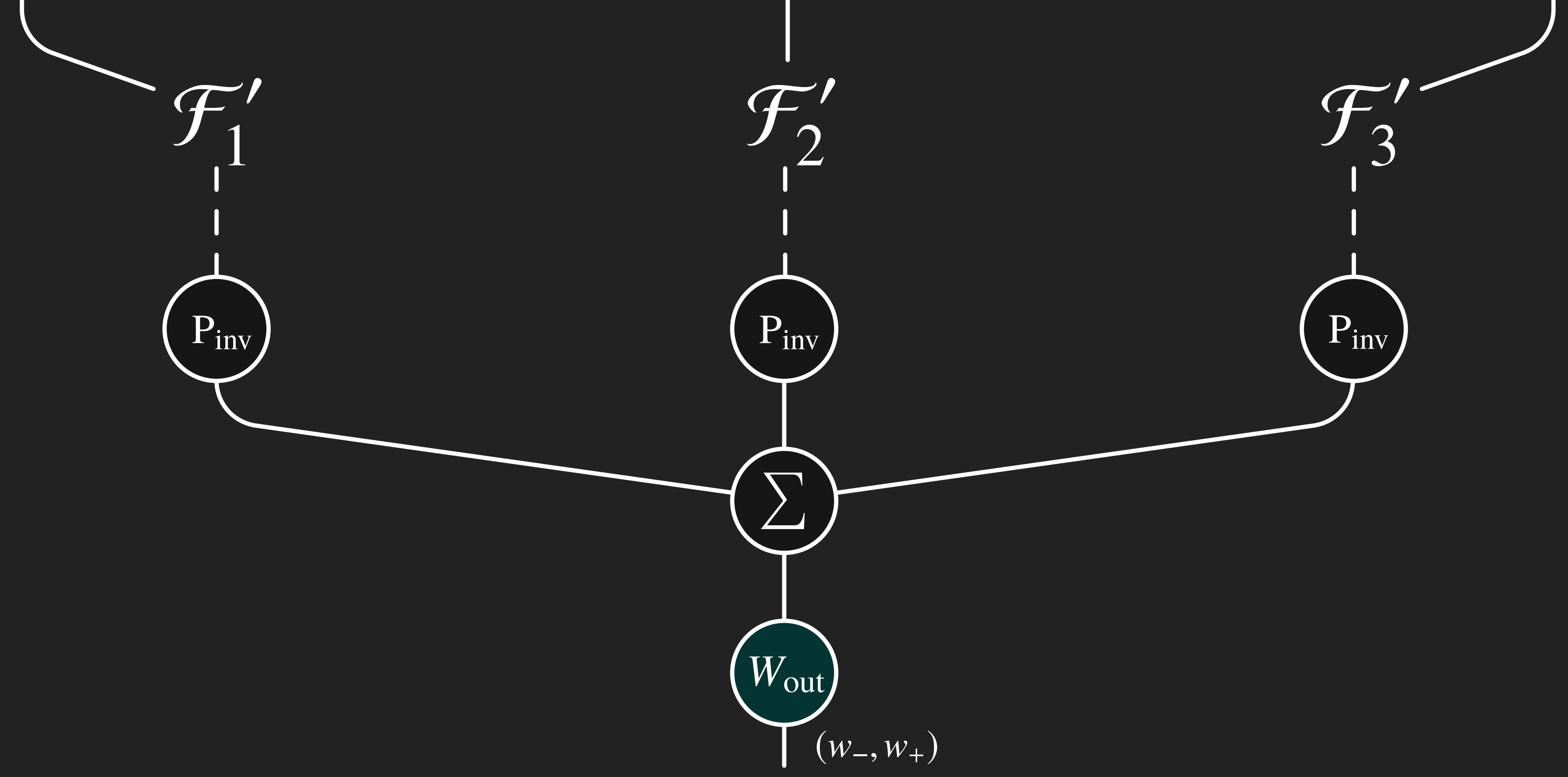
[credit to Alex Bogatskiy]

OUTPUT LAYER



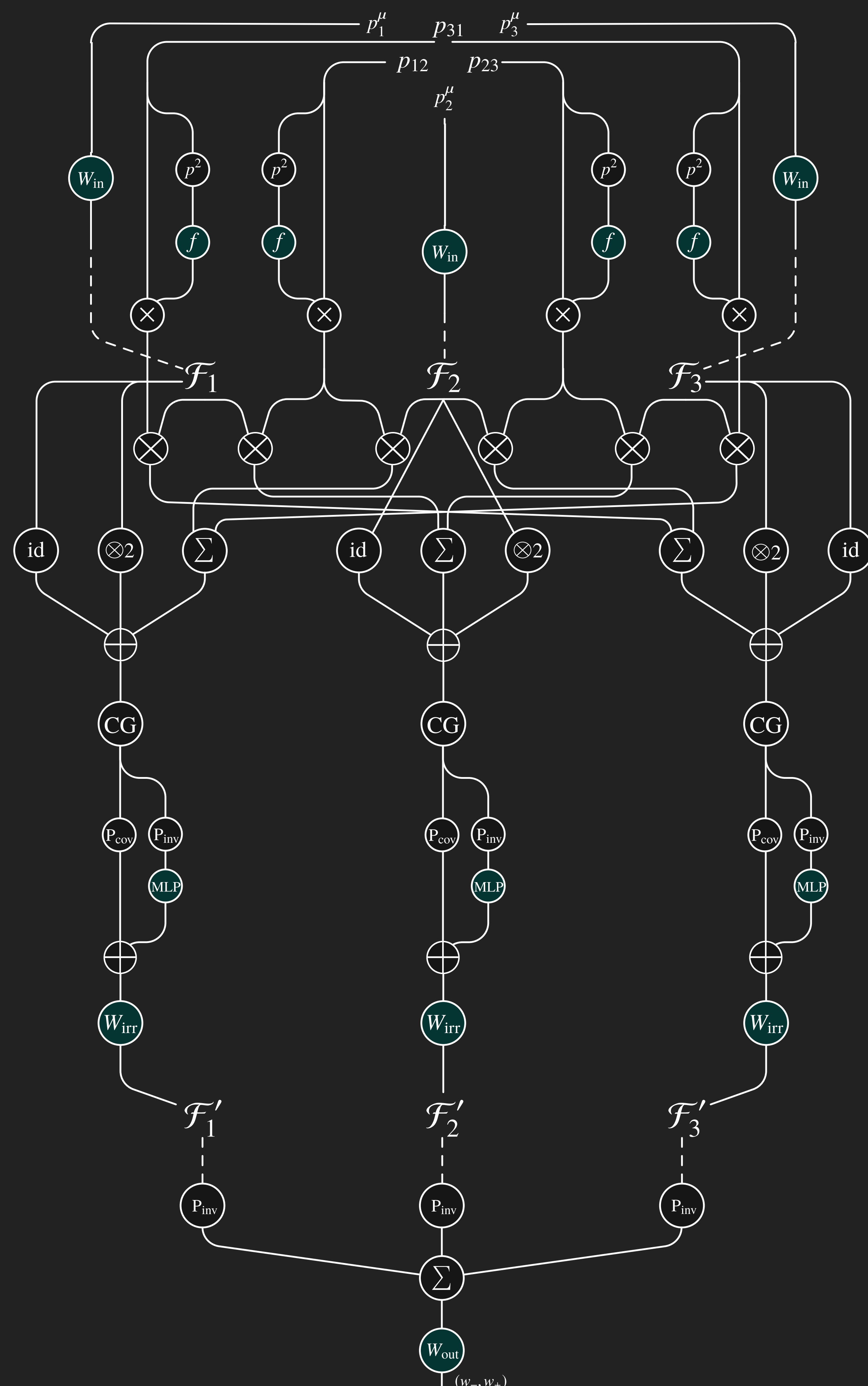
[credit to Alex Bogatskiy]

OUTPUT LAYER



[credit to Alex Bogatskiy]

LGN

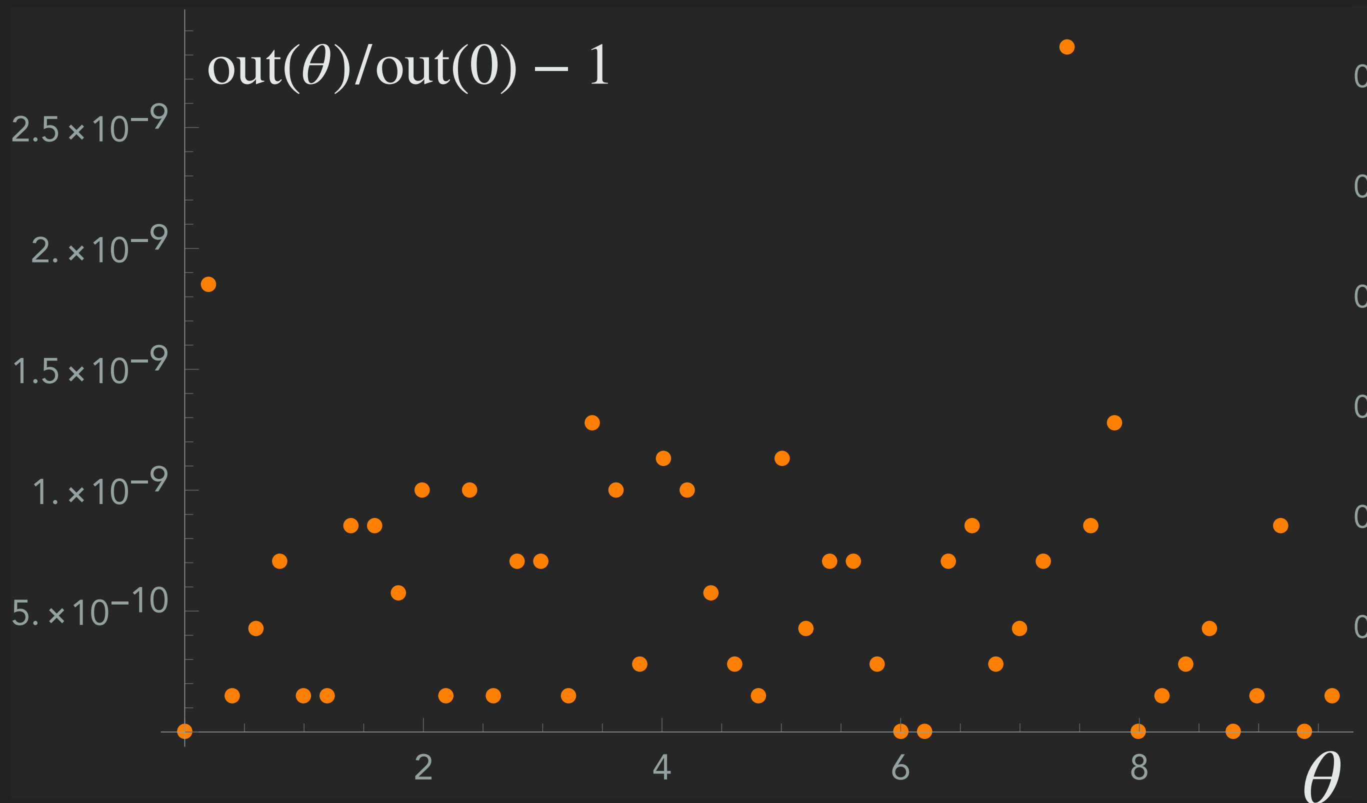


[credit to Alex Bogatskiy]

TESTING LORENTZ INVARIANCE

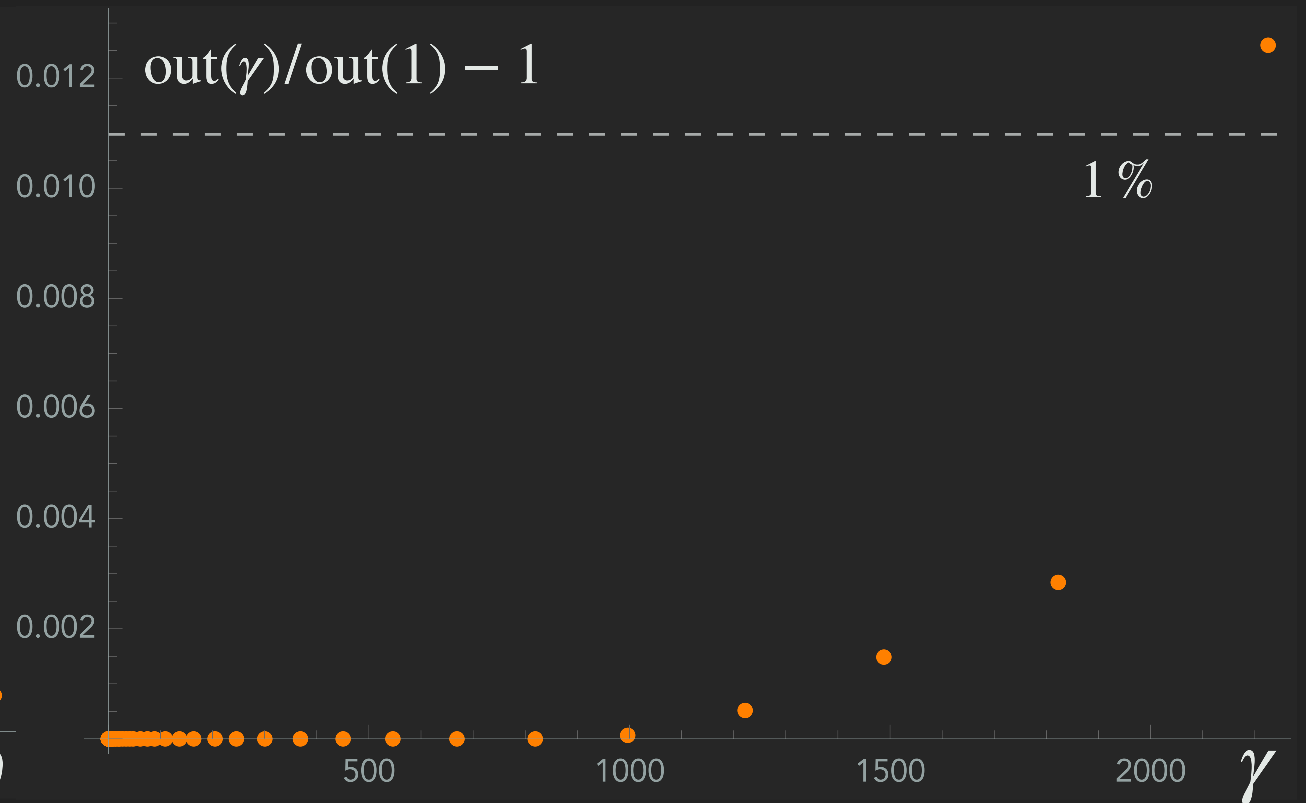
- We have set up LGN for top-tagging, a Lorentz-invariant task.
- Let's test network invariance.
  - Feed in some dummy p^μ twice – once with some Lorentz boost applied – and look for differences in network output.

Rotational invariance



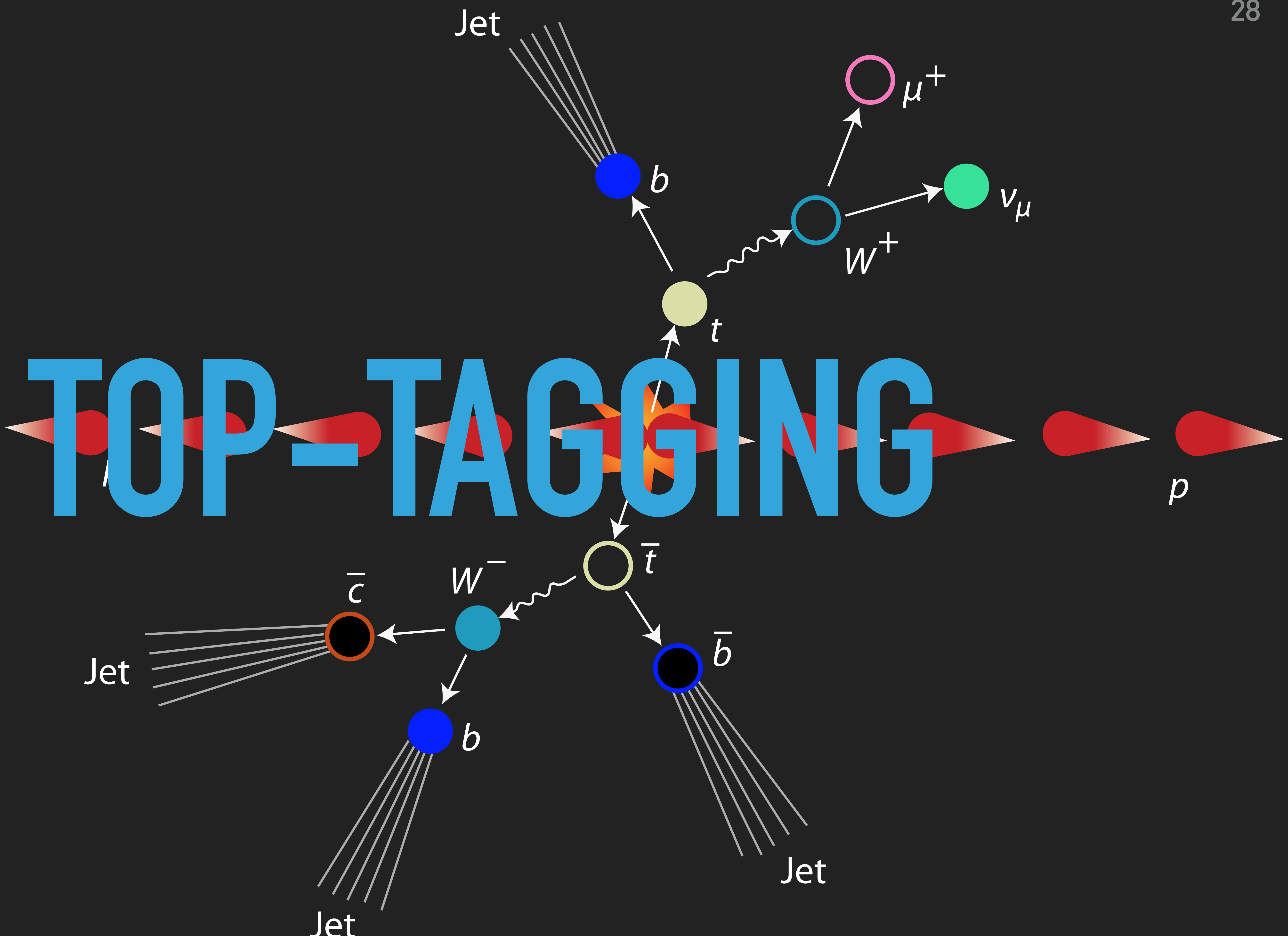
Relative invariance within 10^{-9} using double precision

Boost invariance

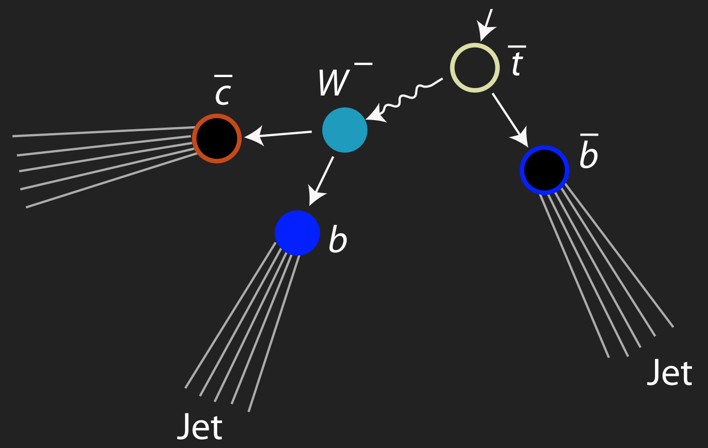


Relative invariance within 10^{-2} limited by floating point precision

TOP-TAGGING



- 2M jets (anti-$k_T\, R = 0.8$).
  - $p_T(j) \in [550{,}650]$ GeV, $|\eta(j)| < 2$.
  - 1M hadronic top decays (signal).
  - 1M leading jets from QCD dijet events (background).

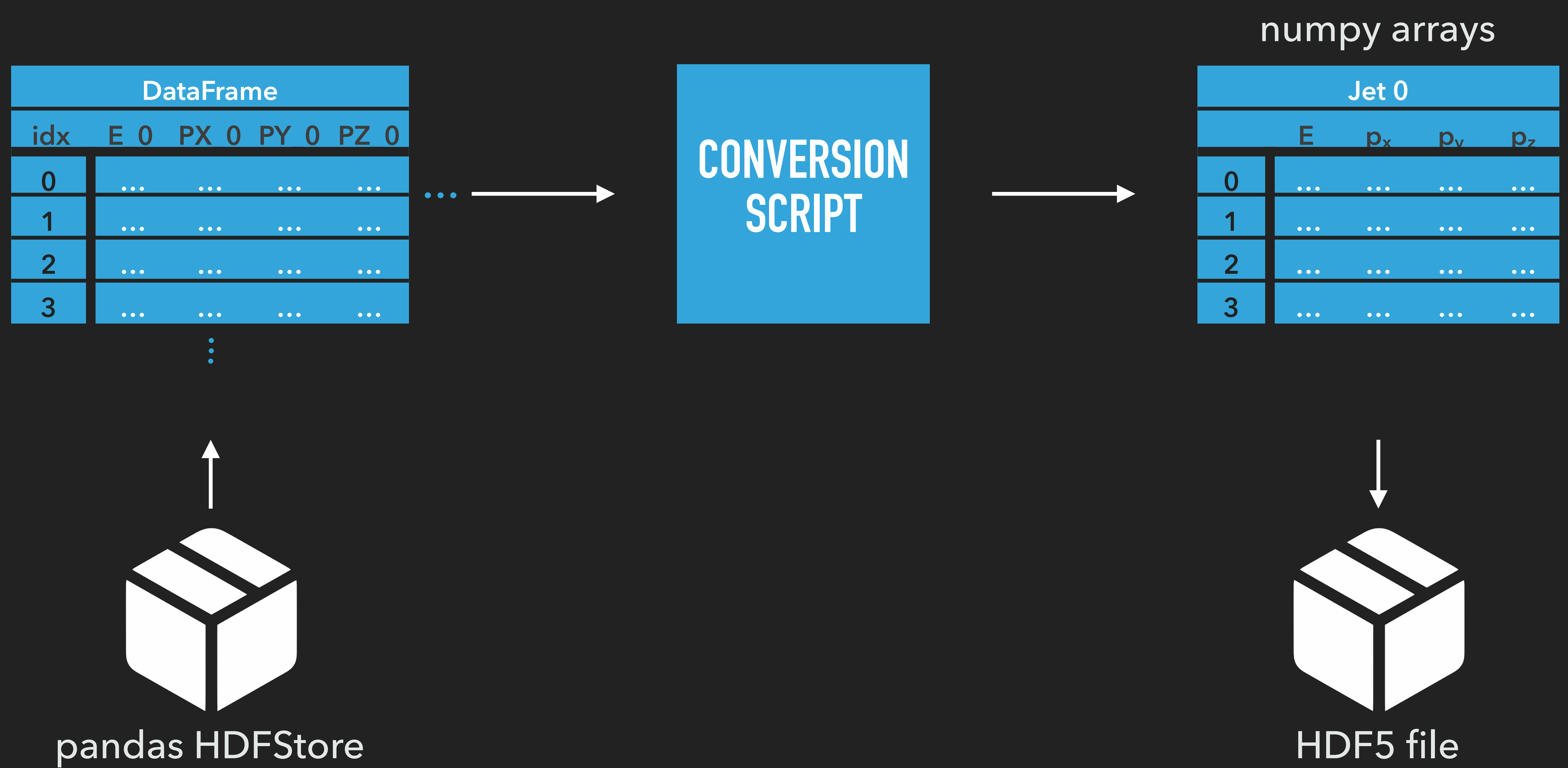


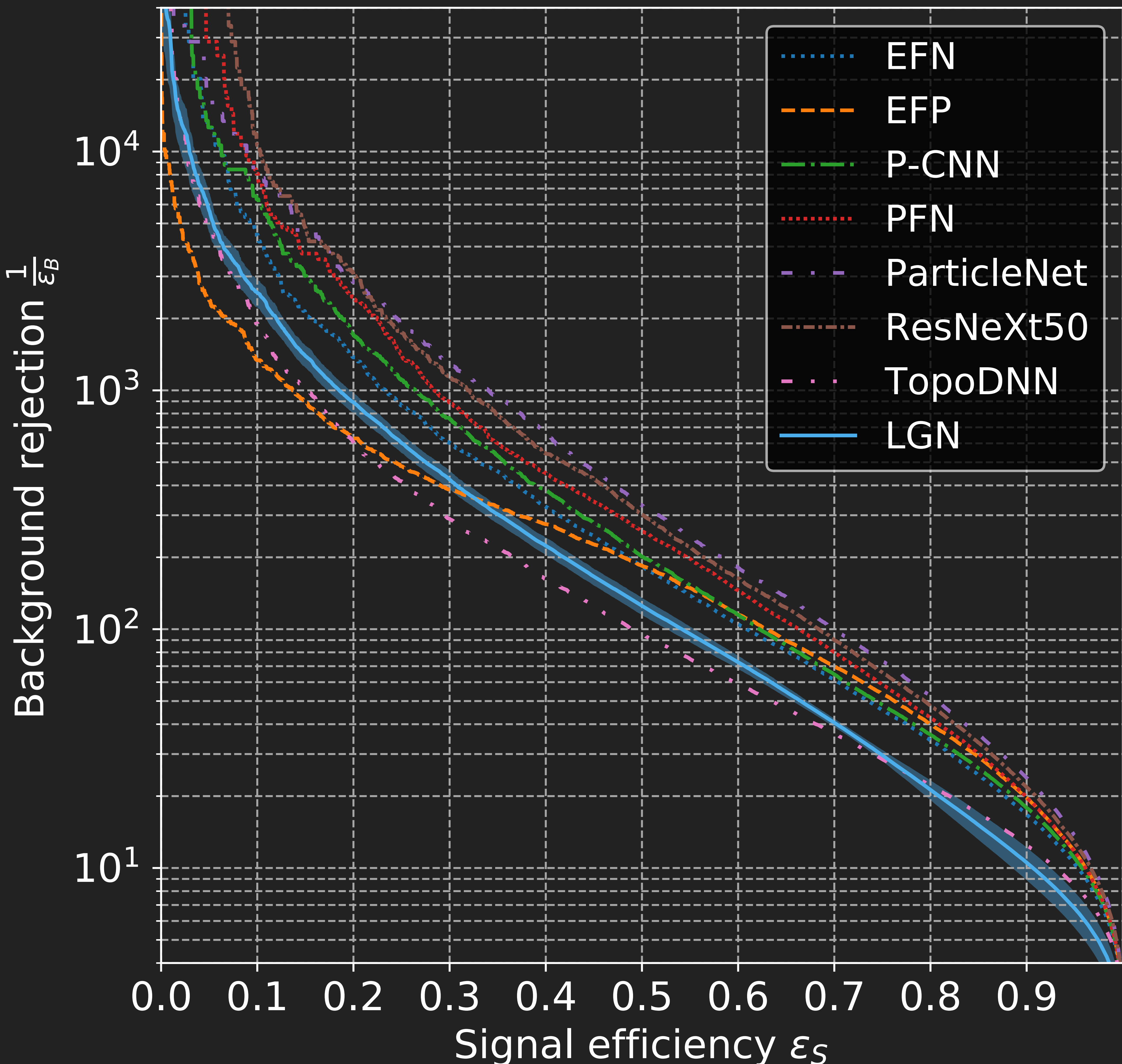
- Simulated with DELPHES + E-flow (fast detector sim).
- For each jet, the 200 leading jet constituents' p^μ stored in Cartesian coordinates, along with the truth top p^μ for signal.

[Kasieczka, G., Plehn, T., Thompson, J., Russel, M. (2019). Top Quark Tagging Reference Dataset (Version v0 (2018_03_27)) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.2603256]

PREPARING THE DATA

- LGN does not require any pre-processing of data.
- We *repackage* the dataset from a pandas DataFrame saved in an HDF5 file, to a native HDF5 format via h5py.



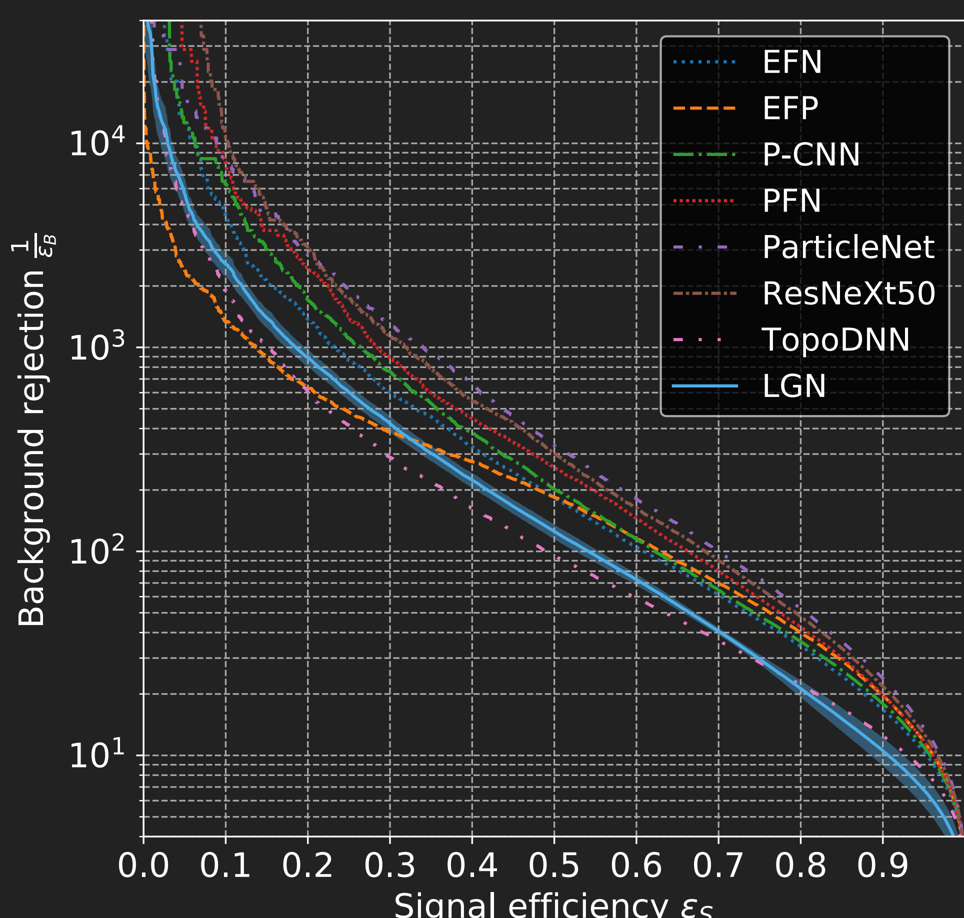


Error band given by $\pm 1/2$ standard deviation

	AUC	Acc	$1/\epsilon_B$ ($\epsilon_S = 0.3$) single	mean	median	#Param
CNN	0.981	0.930	914±14	995±15	975±18	610k
ResNeXt	0.984	0.936	1122±47	1270±28	1286±31	1.46M
TopoDNN	0.972	0.916	295±5	382± 5	378 ± 8	59k
Multi-body N-subjettiness 6	0.979	0.922	792±18	798±12	808±13	57k
Multi-body N-subjettiness 8	0.981	0.929	867±15	918±20	926±18	58k
TreeNiN	0.982	0.933	1025±11	1202±23	1188±24	34k
P-CNN	0.980	0.930	732±24	845±13	834±14	348k
ParticleNet	0.985	0.938	1298±46	1412±45	1393±41	498k
LBN	0.981	0.931	836±17	859±67	966±20	705k
LoLa	0.980	0.929	722±17	768±11	765±11	127k
LDA	0.955	0.892	151±0.4	151.5±0.5	151.7±0.4	184k
Energy Flow Polynomials	0.980	0.932	384			1k
Energy Flow Network	0.979	0.927	633±31	729±13	726±11	82k
Particle Flow Network	0.982	0.932	891±18	1063±21	1052±29	82k
LGN	0.964	0.929	424 ± 82			4.5k

average of 6 independently-trained instances → LGN

[Kasieczka, G., Plehn, T., et. al. (ML Landscape of top taggers, 2019)]



[Macaluso, S., Shih, D. (CNN, 2018)]

[Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K. (ResNeXt, 2016)]

[Pearkes, J., Fedorko, W., Lister, A., Gay, C. (TopoDNN, 2017)]

[Moore, L., Nordström, K., Varma, S., Fairbairn, M. (Multi-body N-subjettiness, 2018)]

[Macaluso, S., Cranmer, K (TreeNiN, 2019)]

[The CMS Collaboration (P-CNN, 2017)]

[Qu, H., Gouskos, L. (ParticleNet, 2019)]

[Erdmann, M., Geiser, E., Rath, Y., Rieger, M. (LBN, 2018)]

[Butter, A., Kasieczka, G., Plehn, T., Russell, M. (LoLa, 2018)]

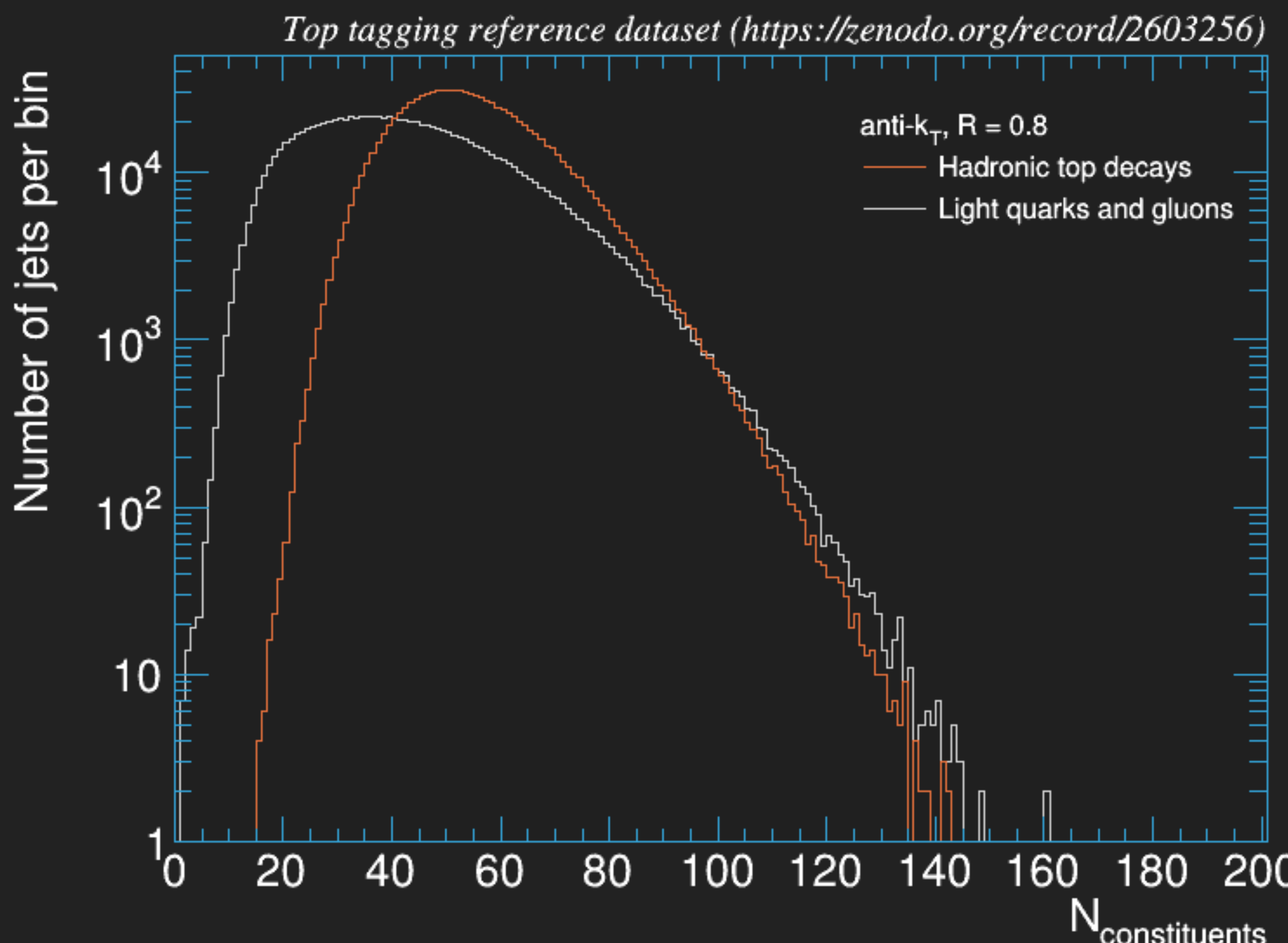
[Dillon, B. M., Faroughy, D. A., Kamenik, J, F. (LDA, 2019)]

[Komiske, P. T., Metodiev, E. M., Thaler, J. (Energy Flow Polynomials, 2017)]

[Komiske, P. T., Metodiev, E. M., Thaler, J. (Energy/Particle Flow Networks, 2018)]

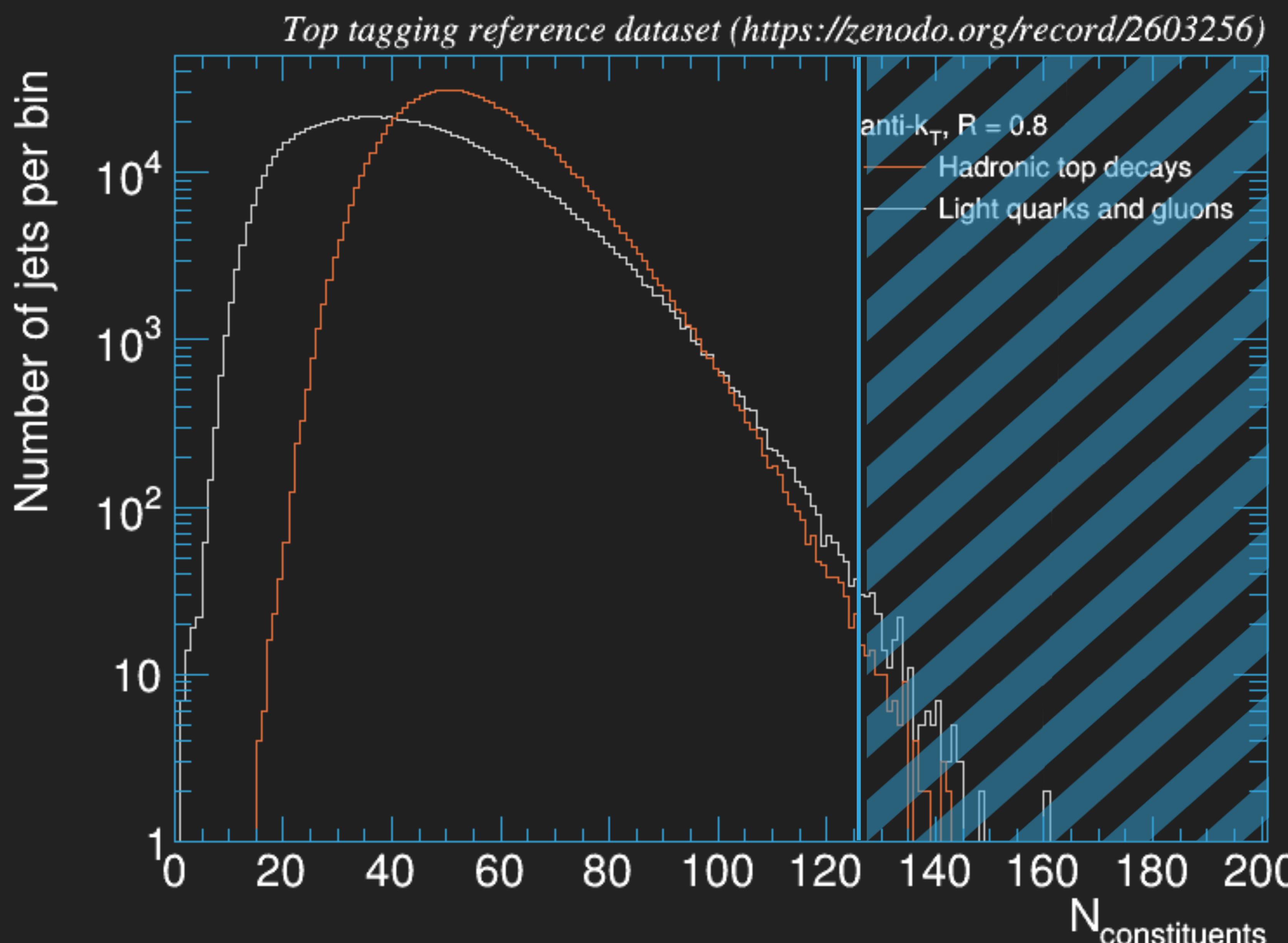
TRIMMING JET CONSTITUENTS

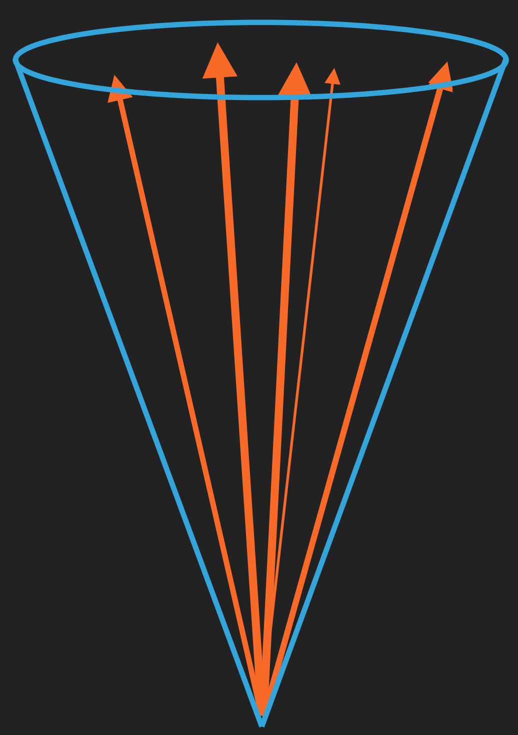
- All jets have notably fewer jet constituents than the maximum value of 200.



TRIMMING JET CONSTITUENTS

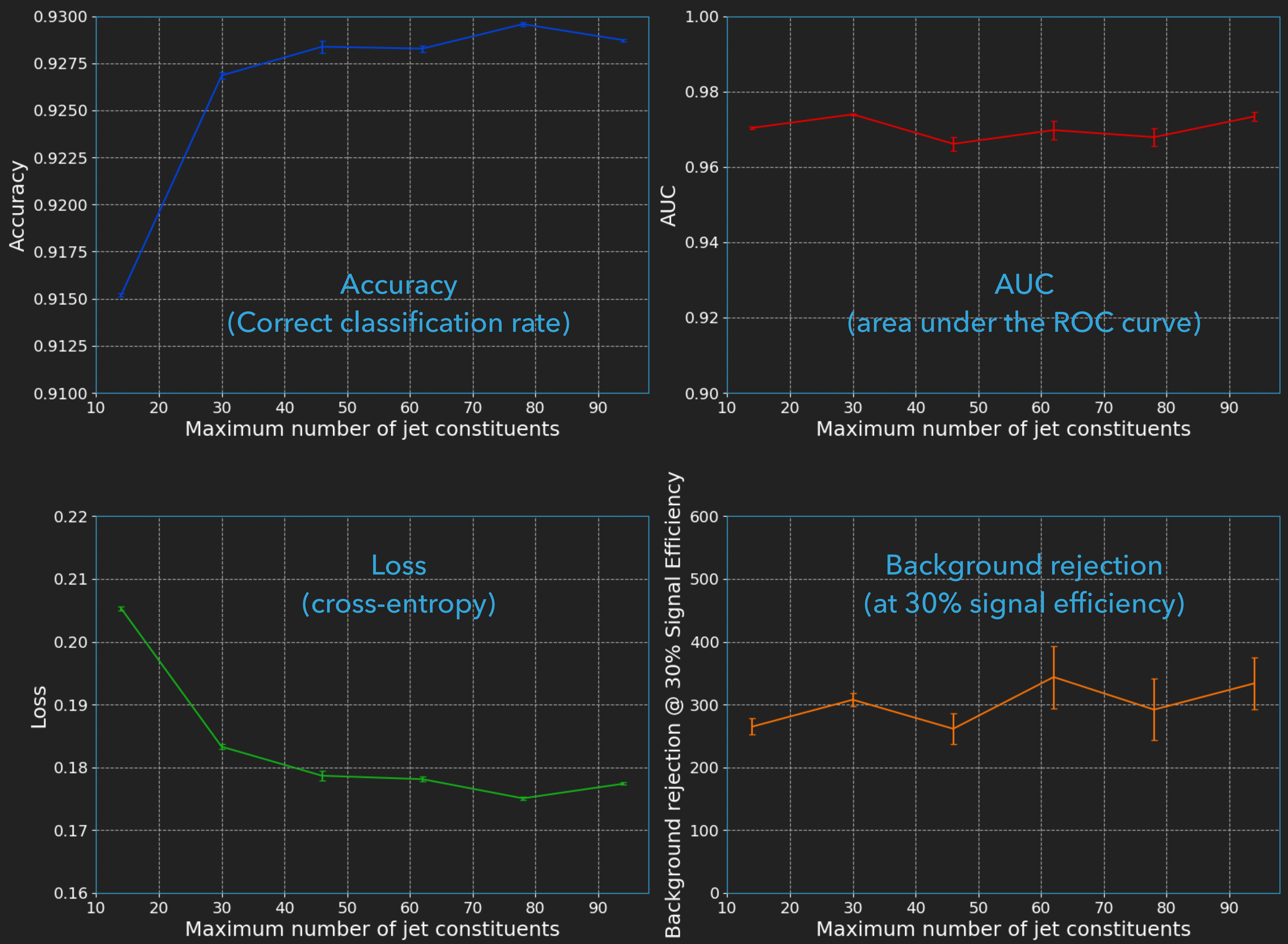
- By default, we use the 126 *leading* constituents of each jet as input. We can alter this cut to test network dependence.





TRIMMING JET CONSTITUENTS

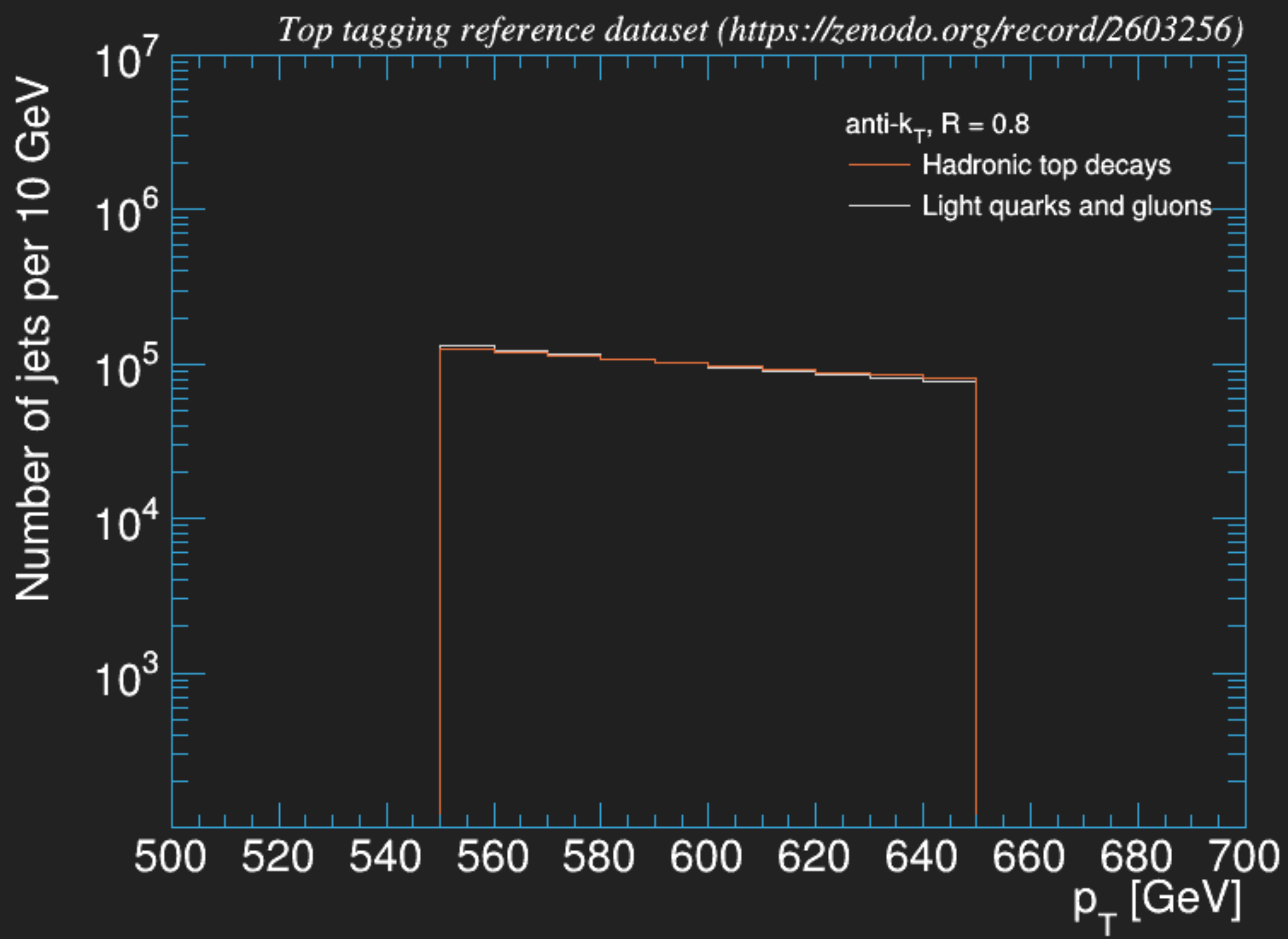
- We find that performance is quite stable across choices of the cut in number of jet constituents used as input.
- We can characterize performance by looking at the network accuracy, area under the ROC curve, loss (cross-entropy), and background rejection at 30% signal efficiency as performance benchmarks.


Accuracy
(Correct classification rate)
Accuracy
0.9300
0.9275
0.9250
0.9225
0.9200
0.9175
0.9150
0.9125
0.9100
10 20 30 40 50 60 70 80 90
Maximum number of jet constituents
AUC
(area under the ROC curve)
AUC
1.00
0.98
0.96
0.94
0.92
0.90
Maximum number of jet constituents
Loss
(cross-entropy)
Loss
0.22
0.21
0.20
0.19
0.18
0.17
0.16
Maximum number of jet constituents
Background rejection
(at 30% signal efficiency)
Background rejection @ 30% Signal Efficiency
600
500
400
300
200
100
0
Maximum number of jet constituents

TRANSFER LEARNING

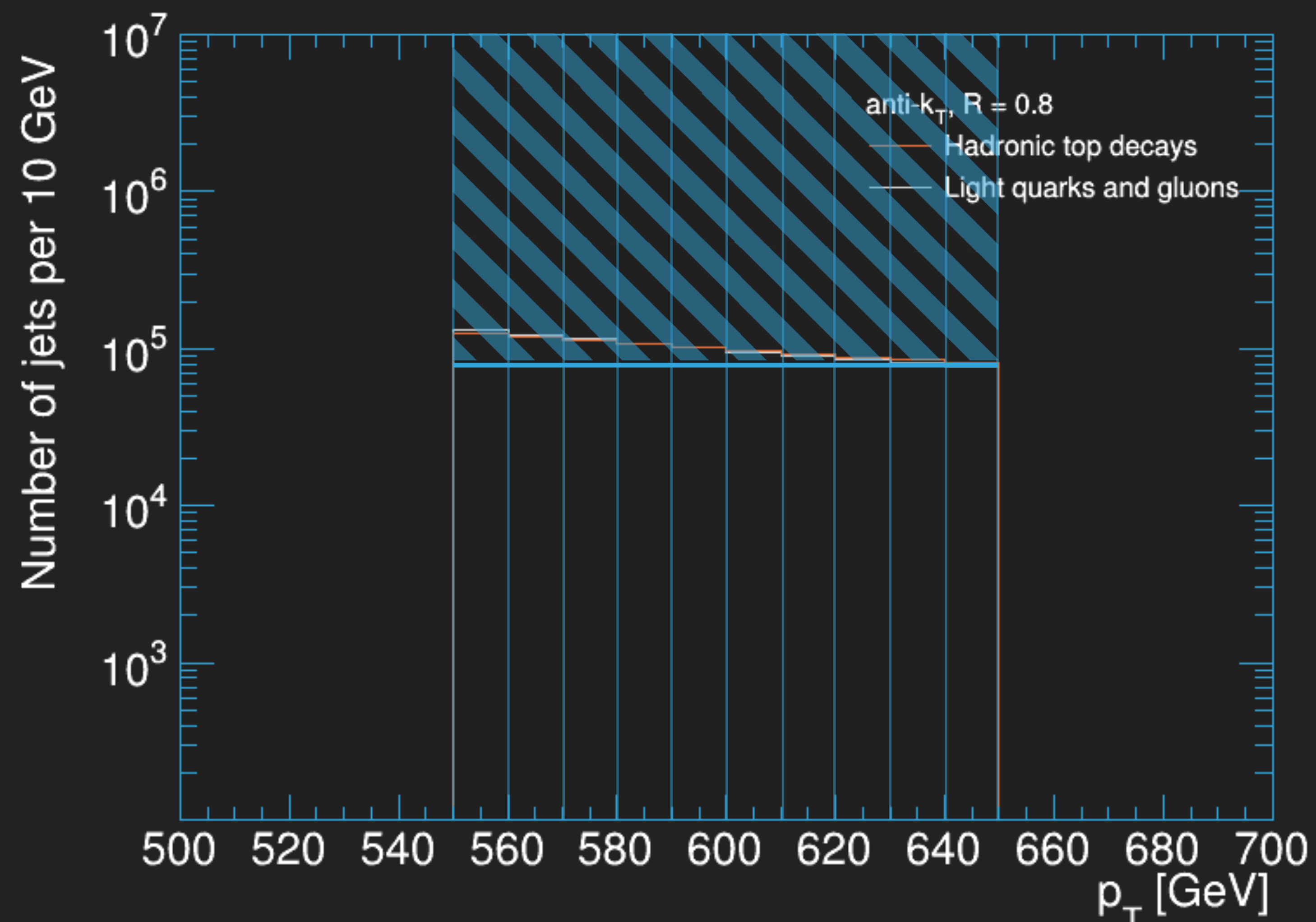
- We can also explore how well LGN can extrapolate results from one region of phase space to another.

- Consider the reconstruction-level jet p_T distribution.



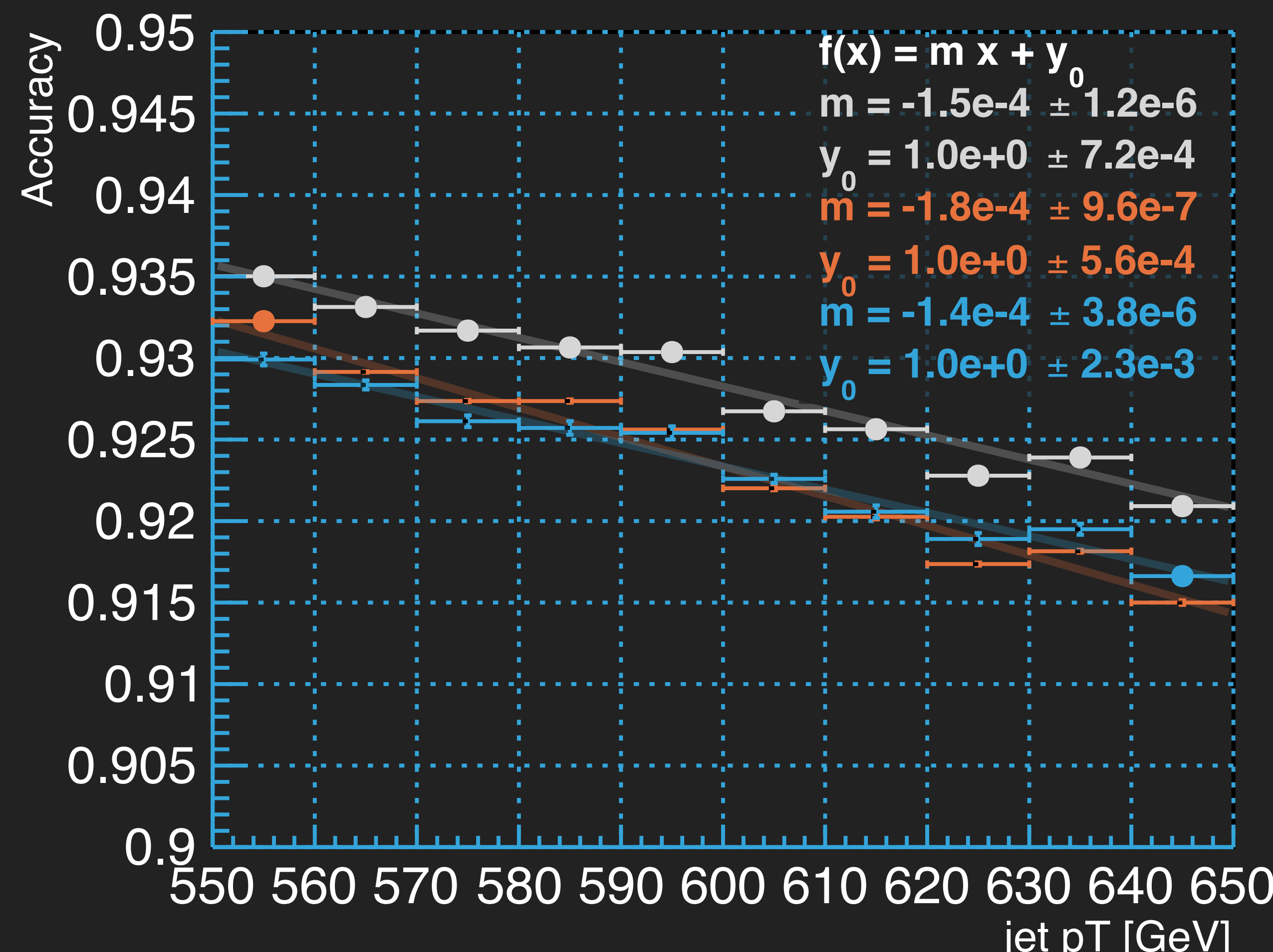
TRANSFER LEARNING: JET p_T

- We divide the data into ten 10 GeV reco jet p_T bins.
  - We discard events such that each bin has an even split of signal and background, and the same total number of training events.

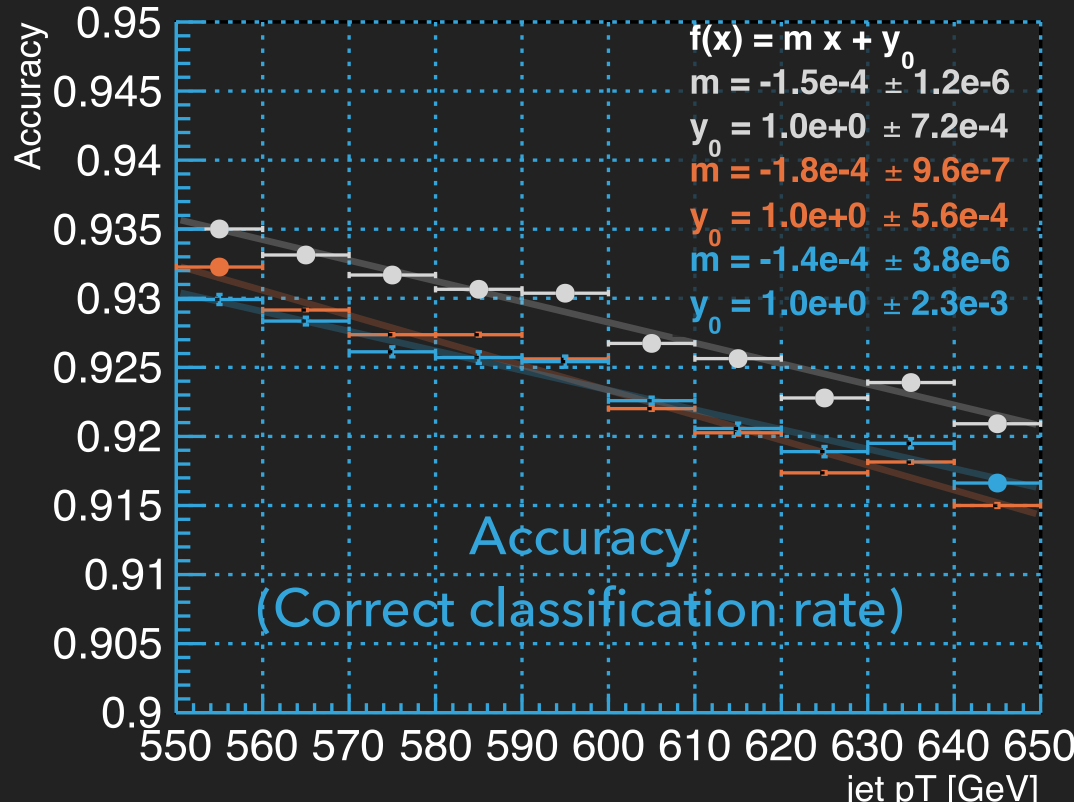


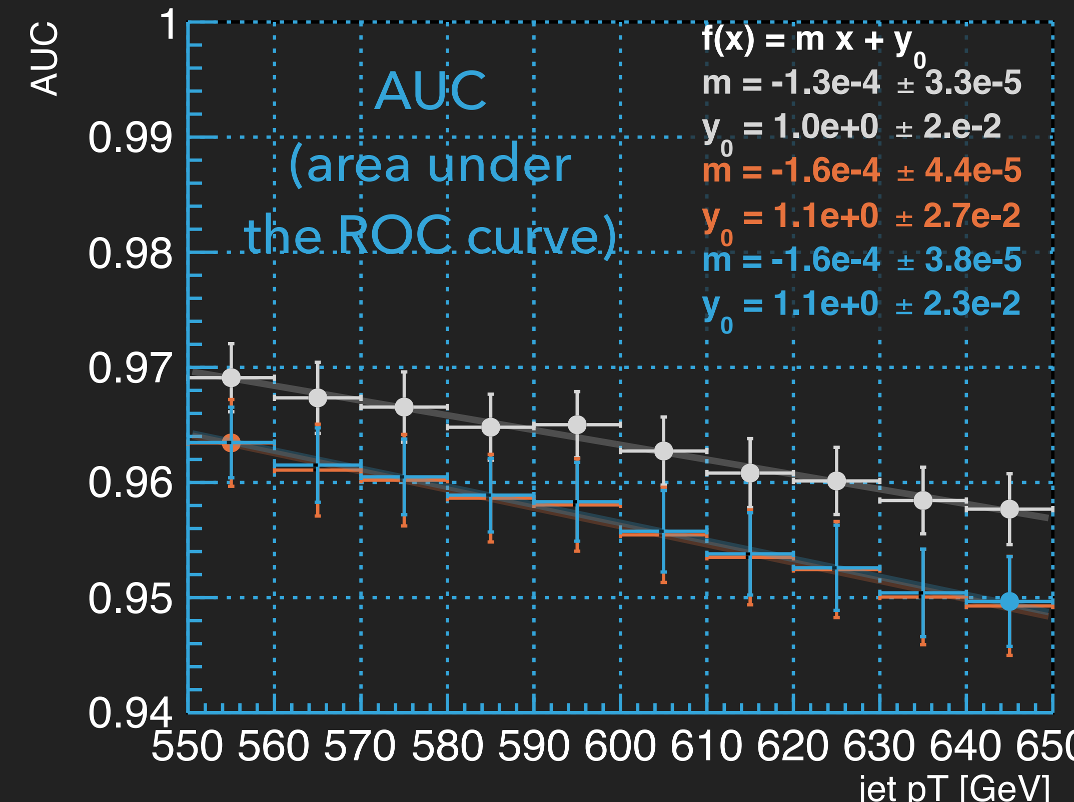
TRANSFER LEARNING: JET p_T

- LGN is relatively agnostic to the jet p_T bin used for training.
  - Performance is correlated with the jet p_T of the testing bin, but this correlation is consistent across training bins.

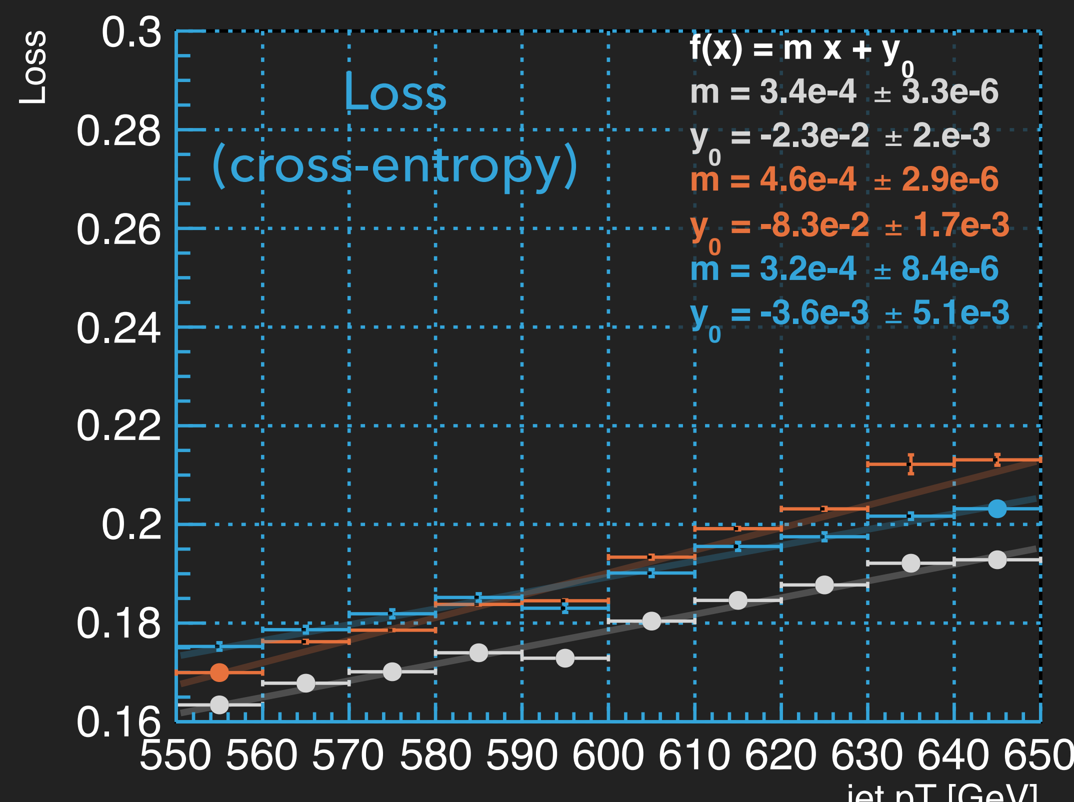


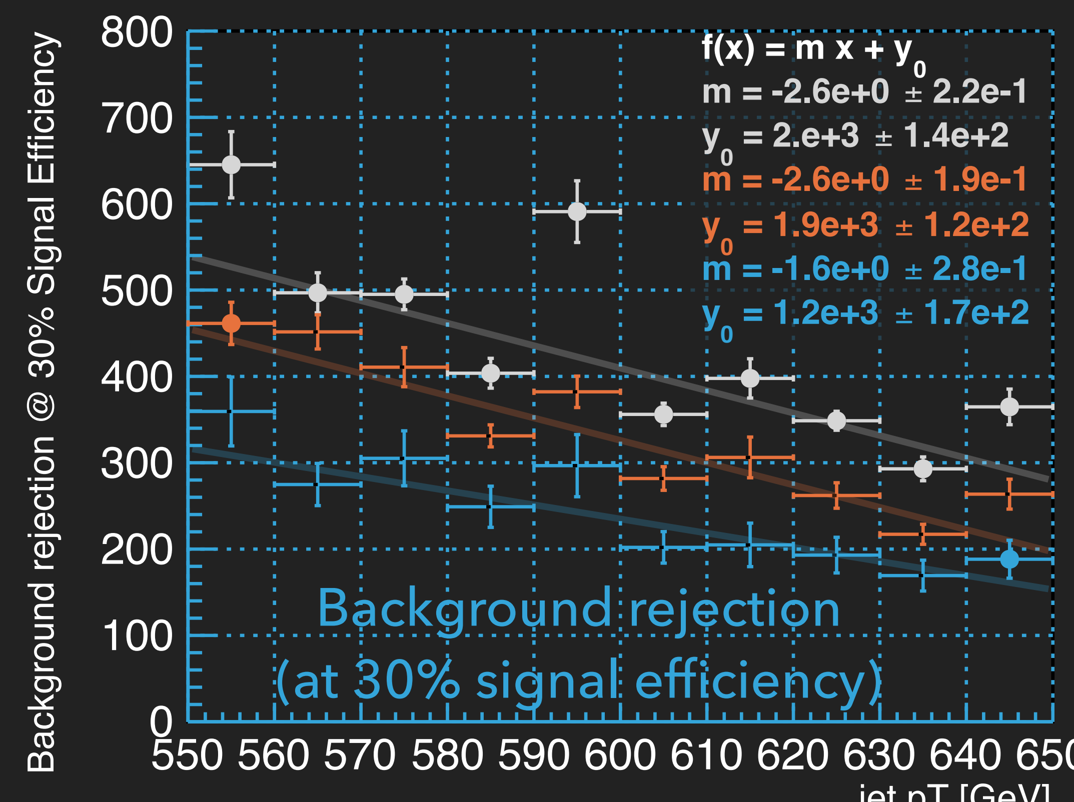
All metrics evaluated on jet p_T bins from the testing set. Circles indicate the p_T jet bin(s) used for training.





Each metric is averaged over 6-8 trained instances. Error bars are given by ± 1/2 standard error on the mean.





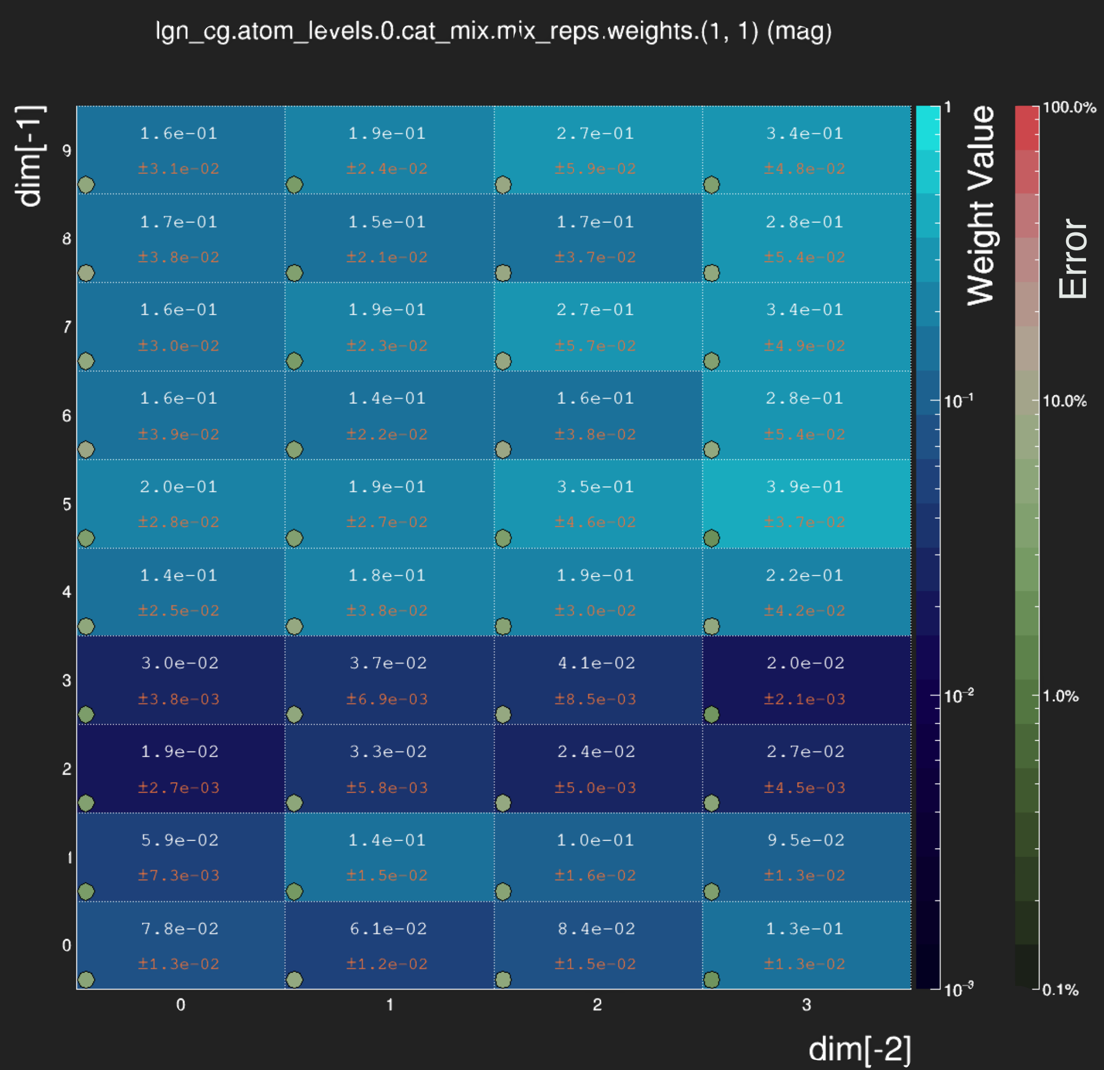
A FEW IMPORTANT DETAILS…

- LGN is currently slow to train.
  - ~8 hours/epoch, on a single Nvidia GeForce RTX 2080.
  - Could be improved by parallelization across GPU's, or a custom CUDA kernel.
- We have not performed a full hyper-parameter scan.
  - Better-performing configurations *may* exist.

INTERPRETABILITY

- LGN is not the highest performer, but trades a small amount of performance for prospects of interpretability.

	#Param
CNN	610k
ResNeXt	1.46M
TopoDNN	59k
Multi-body N-subjettiness 6	57k
Multi-body N-subjettiness 8	58k
TreeNiN	34k
P-CNN	348k
ParticleNet	498k
LBN	705k
LoLa	127k
LDA	184k
Energy Flow Polynomials	1k
Energy Flow Network	82k
Particle Flow Network	82k
LGN	4.5k

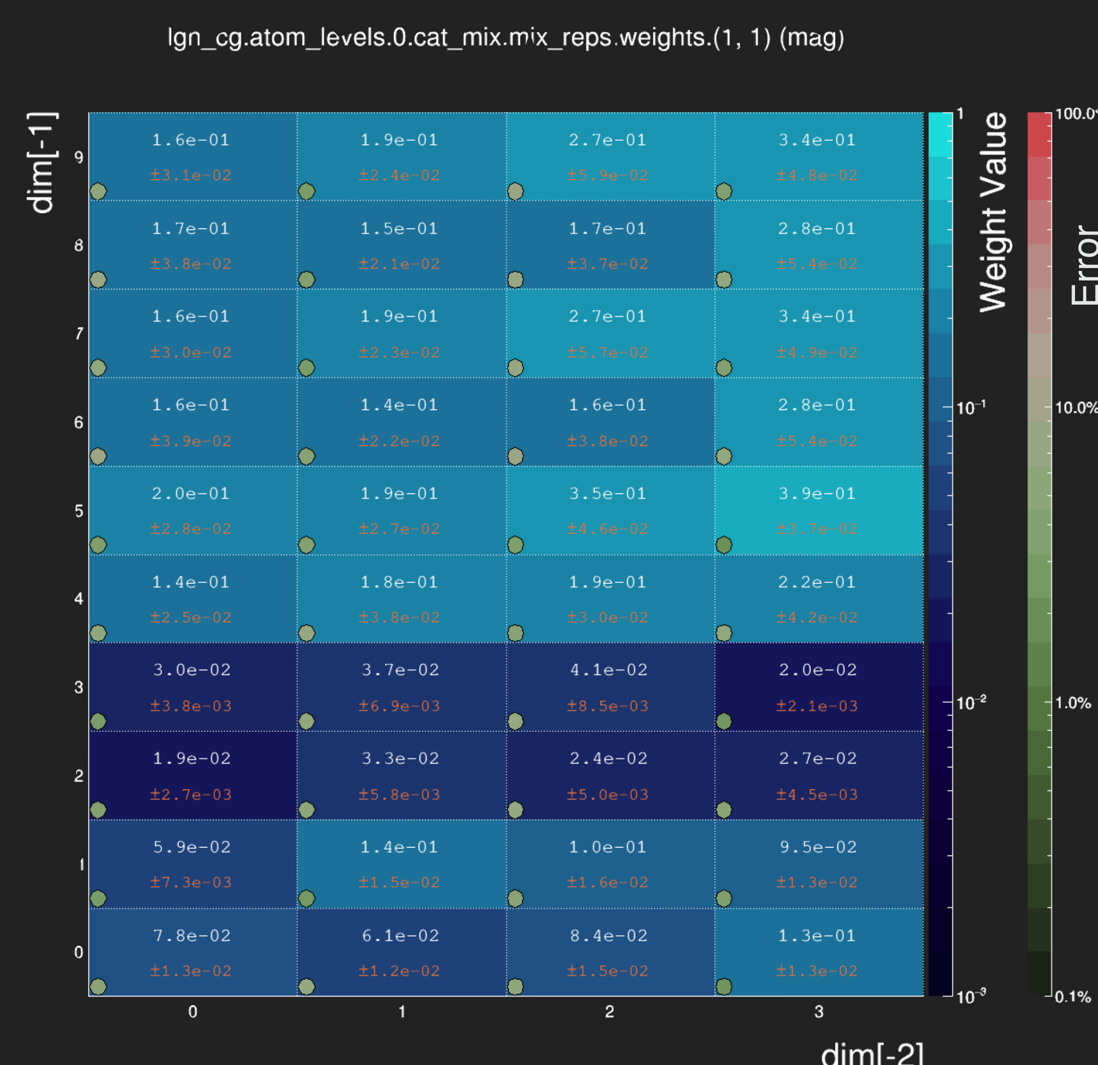


(Work in Progress)

INTERPRETABILITY

- LGN is not the highest performer, but trades a small amount of performance for prospects of interpretability.
- Among theory-inspired networks, it has far fewer learnable parameters than any others except for EFP.

	#Param
LBN	705k
LoLa	127k
LDA	184k
Energy Flow Polynomials	1k
Energy Flow Network	82k
Particle Flow Network	82k
LGN	4.5k



(Work in Progress)

INTERPRETABILITY

- Furthermore, these parameters correspond with physically-meaningful quantities – Lorentz-equivariant expressions formed by tensor products of momenta.

(Work in Progress)

CONCLUSION

- To the best of our knowledge, LGN is the first example of a neural network in particle physics with the symmetries of the Lorentz group fully embedded in the architecture.
- This architecture can be naturally extended for data containing additional particle information, such as charge.
- Furthermore, LGN can in principle be used for Lorentz-covariant tasks, such as four-momentum regression.

ONGOING AND NEAR-FUTURE WORK

- Studying irrep mixing weights.
  - Are there patterns among better-performing networks?
  - Correlations with training bin jet p_T?
- Covariant top quark four-momentum measurement.
  - Can we predict momenta in $t \to W(q\bar{q})b$?*

* requires a different dataset

REFERENCES

- Past talks:
  - ML4Jets 2020: https://indi.to/xmXL8
  - ICML 2020: https://icml.cc/virtual/2020/poster/5843
- Papers:
  - ICML 2020: https://arxiv.org/abs/2006.04780
  - A more HEP-oriented companion paper coming soon…
- GitHub: https://github.com/fizisist/LorentzGroupNetwork

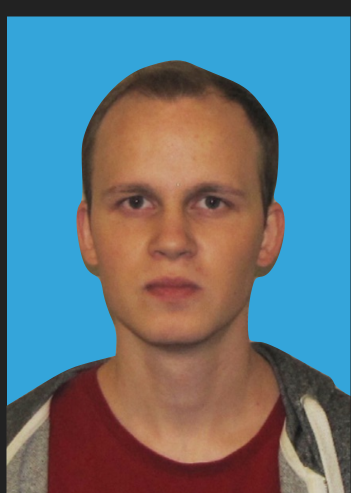
THE UNIVERSITY OF CHICAGO
Physics Department
Computer Science Department

ENRICO FERMI INSTITUTE

FLATIRON INSTITUTE

Center for Data and Computing
AT THE UNIVERSITY OF CHICAGO

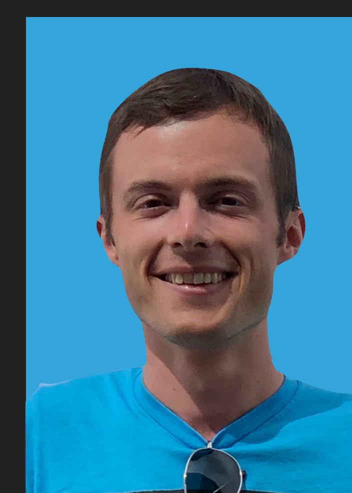
KADANOFF CENTER
FOR THEORETICAL PHYSICS

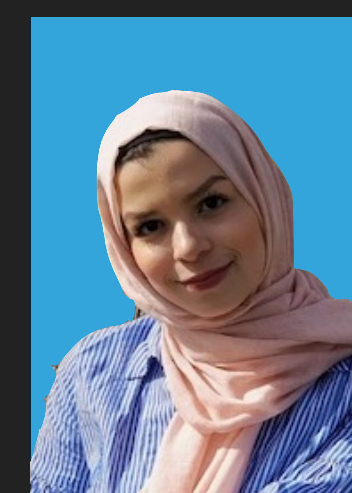

Alexander Bogatskiy


Brandon Anderson

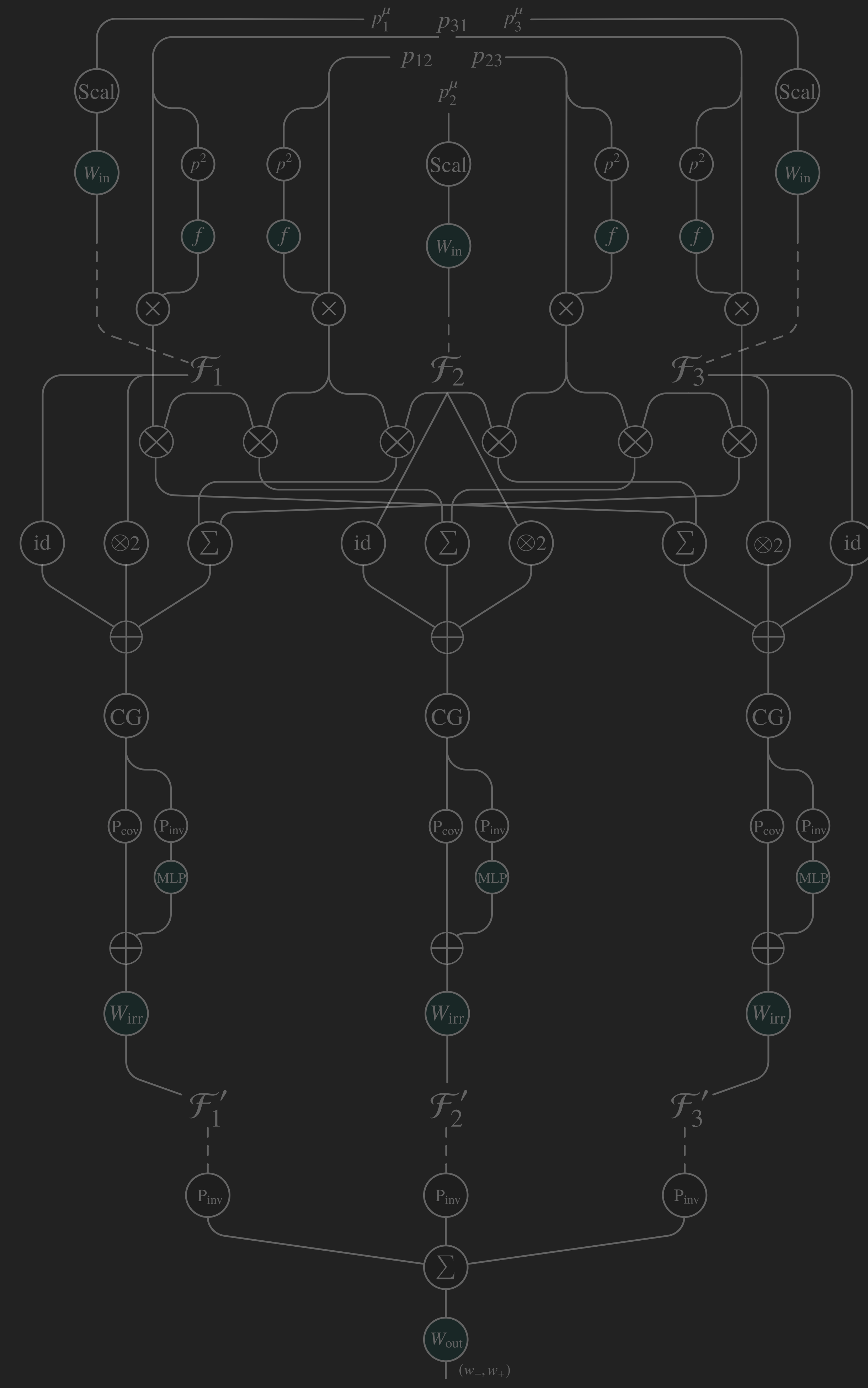

Risi Kondor


David Miller


Jan Offermann


Marwah Roussi

BACKUP


p_1^μ
p_{31}
p_3^μ
p_{12}
p_{23}
p_2^μ
Scal
W_{in}
p^2
f
$\mathcal{F}_1$
$\mathcal{F}_2$
$\mathcal{F}_3$
id
$\otimes 2$
$\sum$
CG
P_{cov}
P_{inv}
MLP
W_{irr}
$\mathcal{F}_1'$
$\mathcal{F}_2'$
$\mathcal{F}_3'$
W_{out}
(w_-, w_+)

1) DATASET KINEMATIC DISTRIBUTIONS

2) DATASET EVENT DISPLAYS

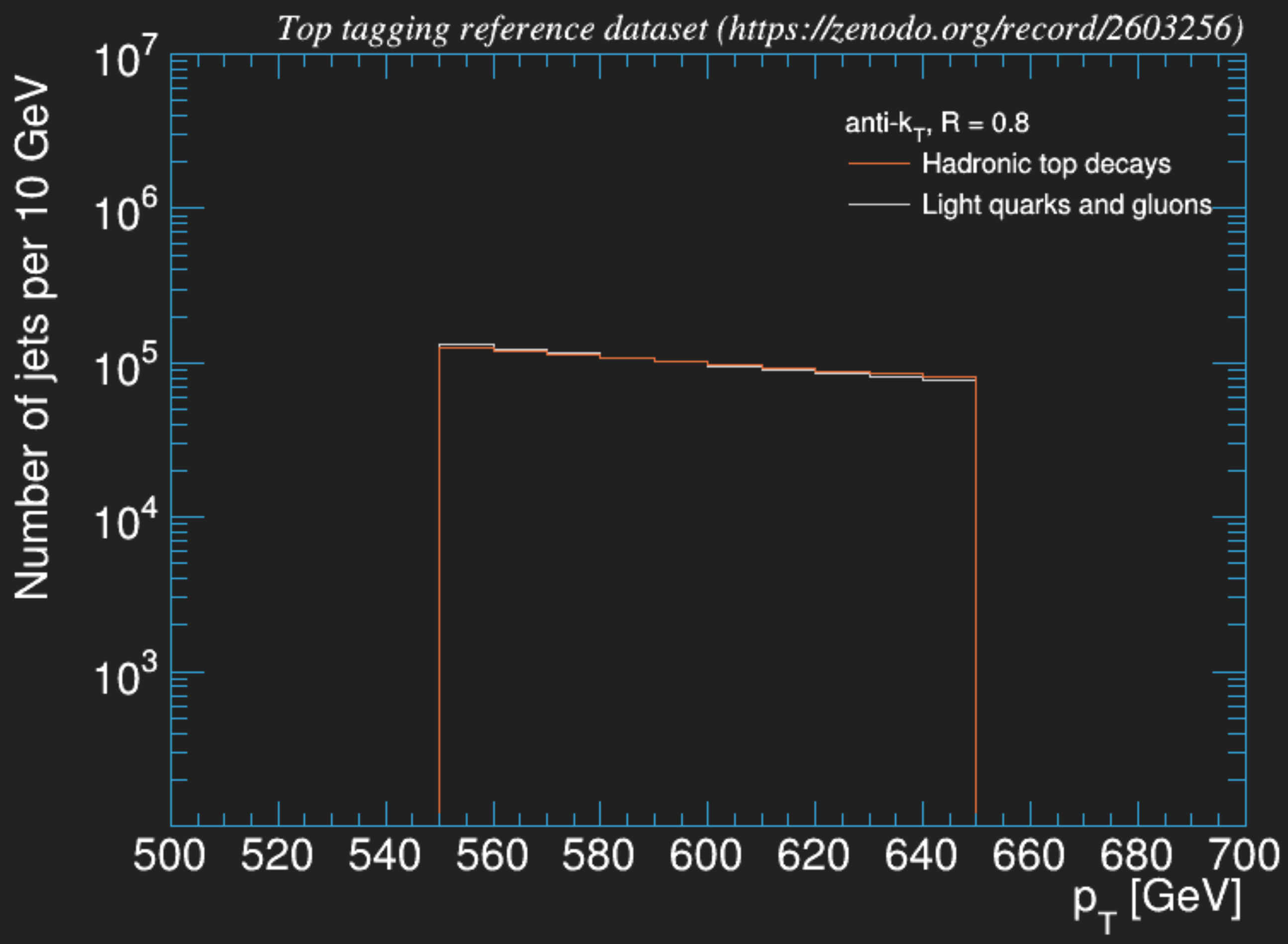
3) TRAINING STATS

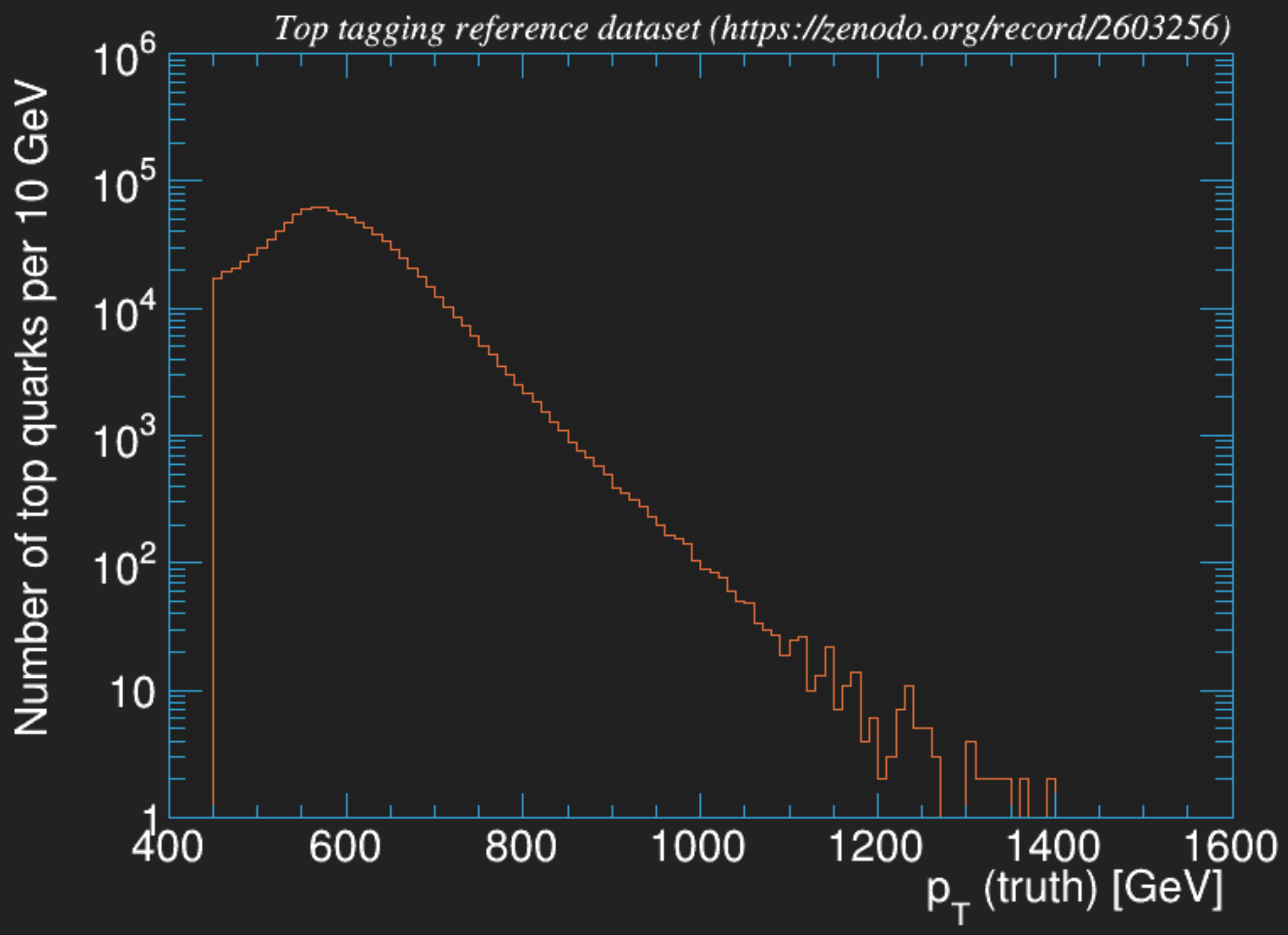
4) COMPARISON TOP TAGGERS

5) SCHUR'S LEMMA

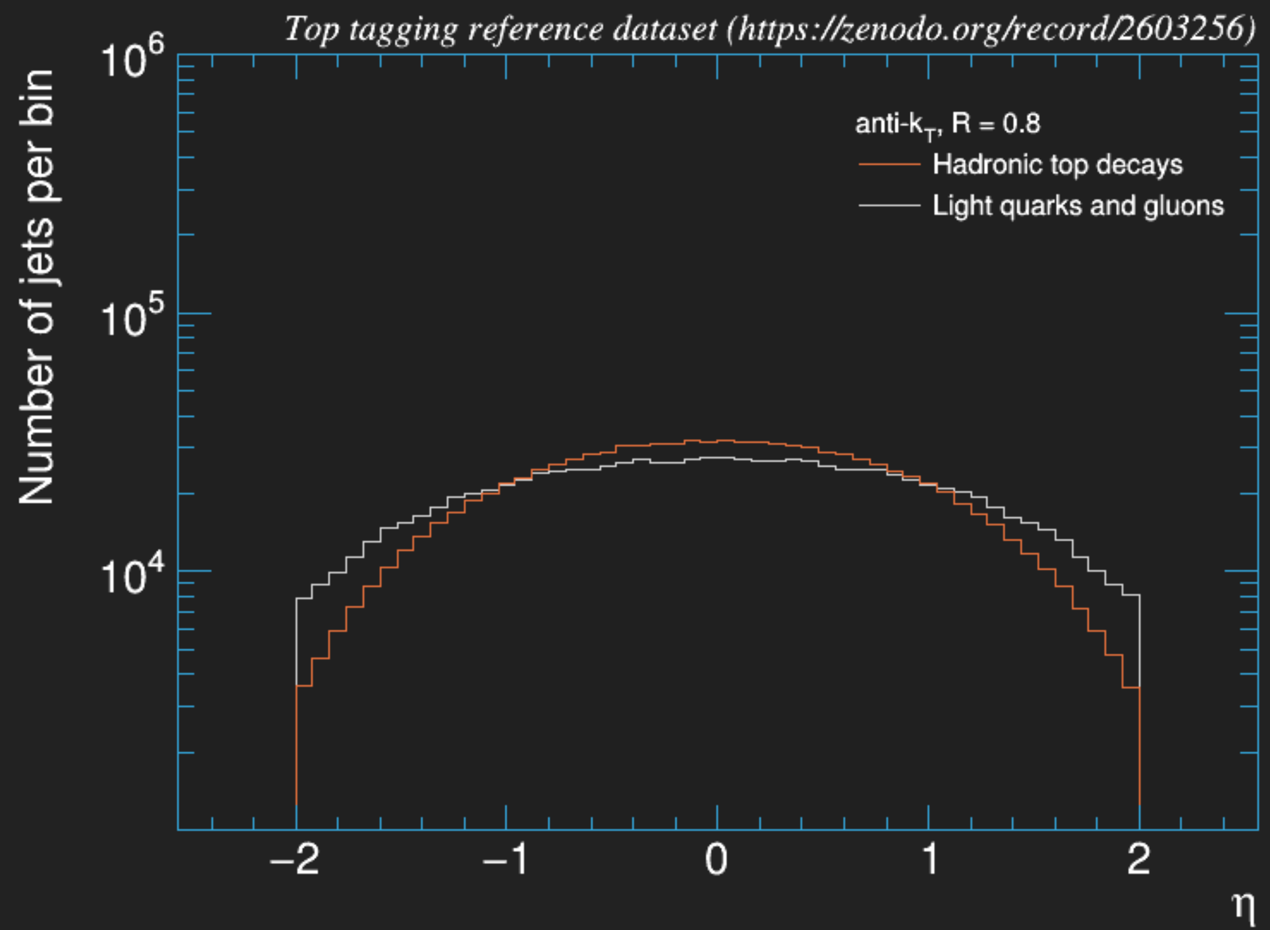
6) CLEBSCH-GORDAN COEFFICIENTS

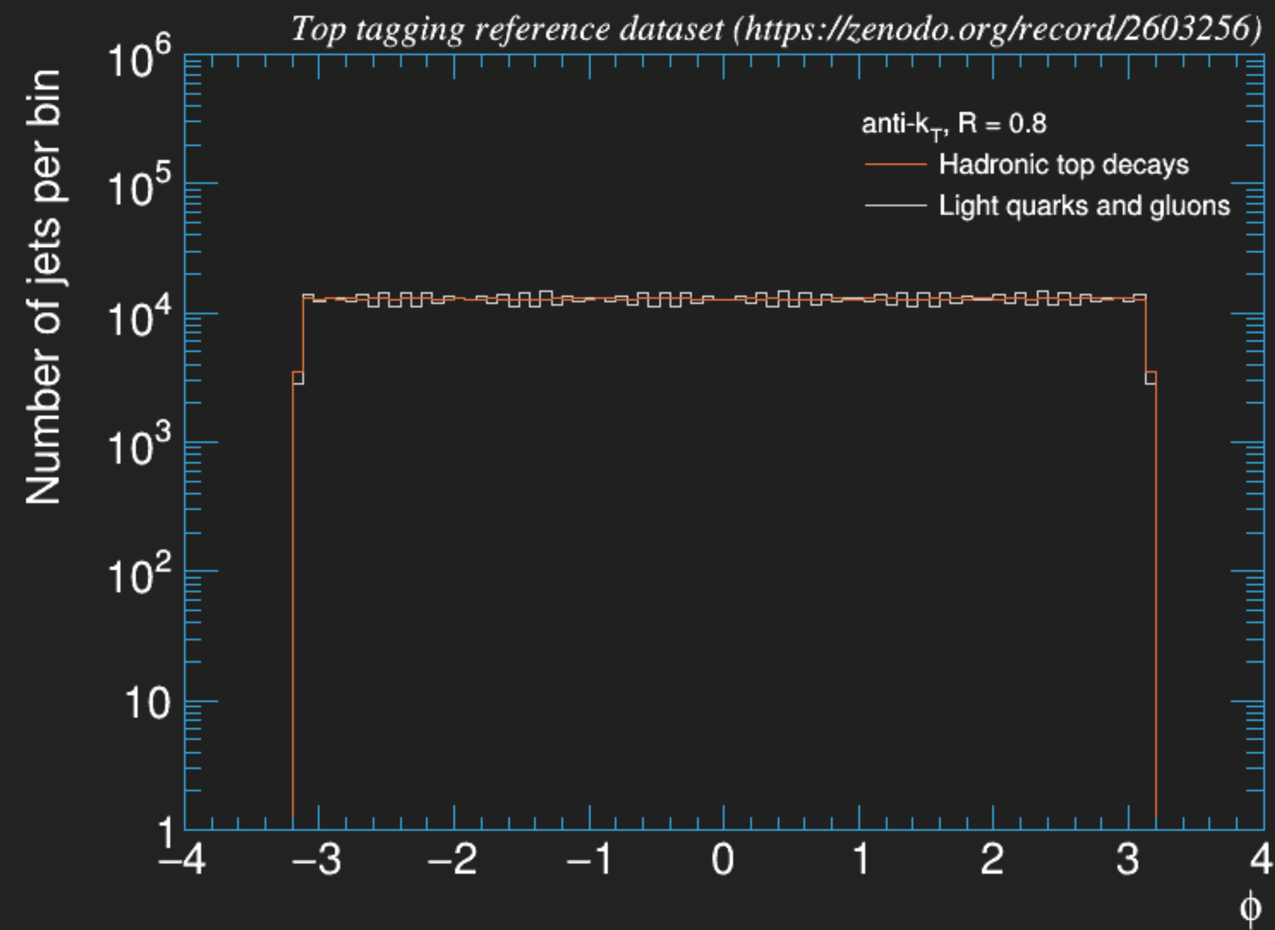
JET p_T


Top tagging reference dataset (https://zenodo.org/record/2603256)
Number of jets per 10 GeV
10^7
10^6
10^5
10^4
10^3
anti-k_T, R = 0.8
Hadronic top decays
Light quarks and gluons
500 520 540 560 580 600 620 640 660 680 700
p_T [GeV]

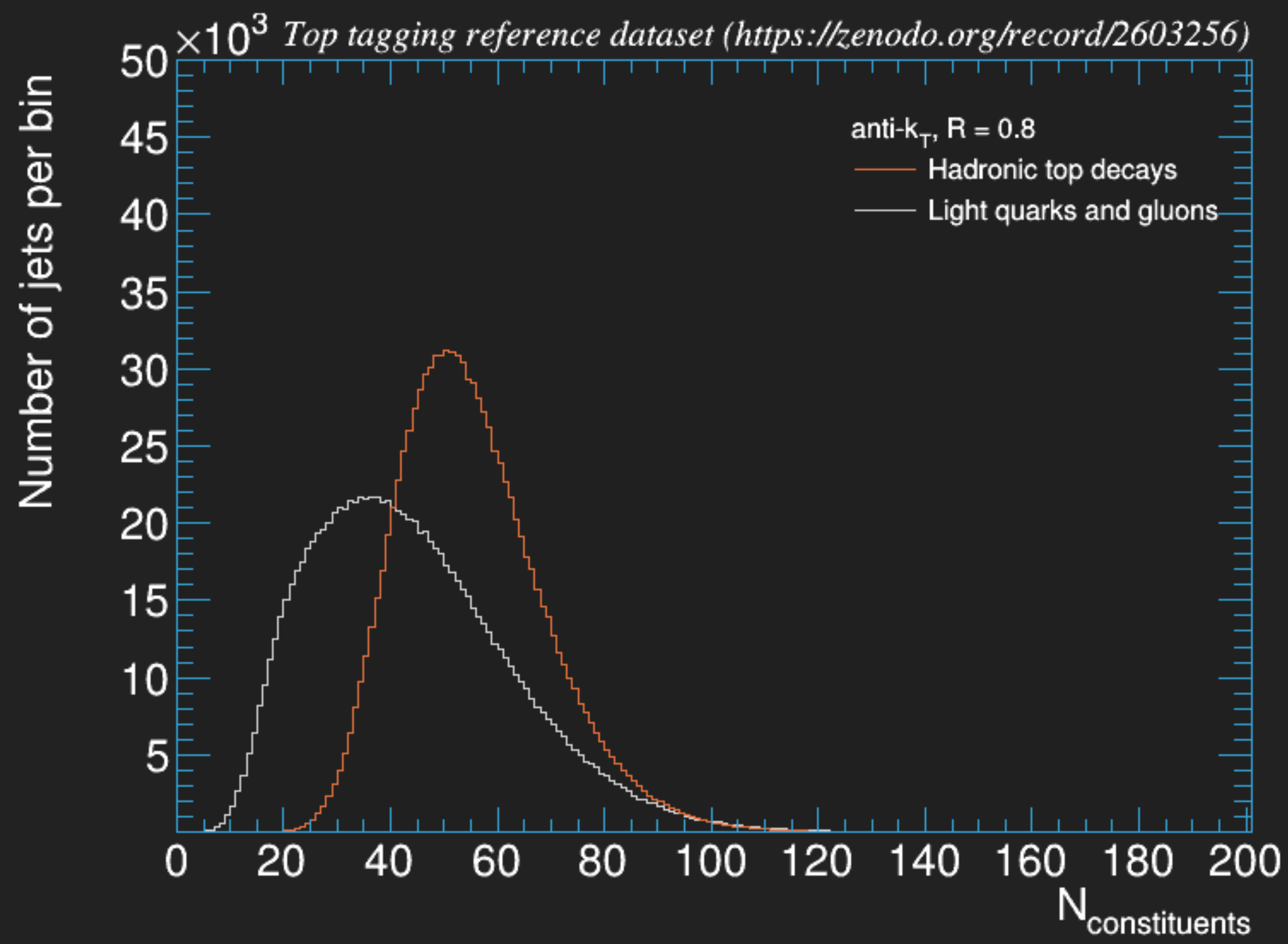

Top tagging reference dataset (https://zenodo.org/record/2603256)
Number of top quarks per 10 GeV
10^6
10^5
10^4
10^3
10^2
10
1
400 600 800 1000 1200 1400 1600
p_T (truth) [GeV]

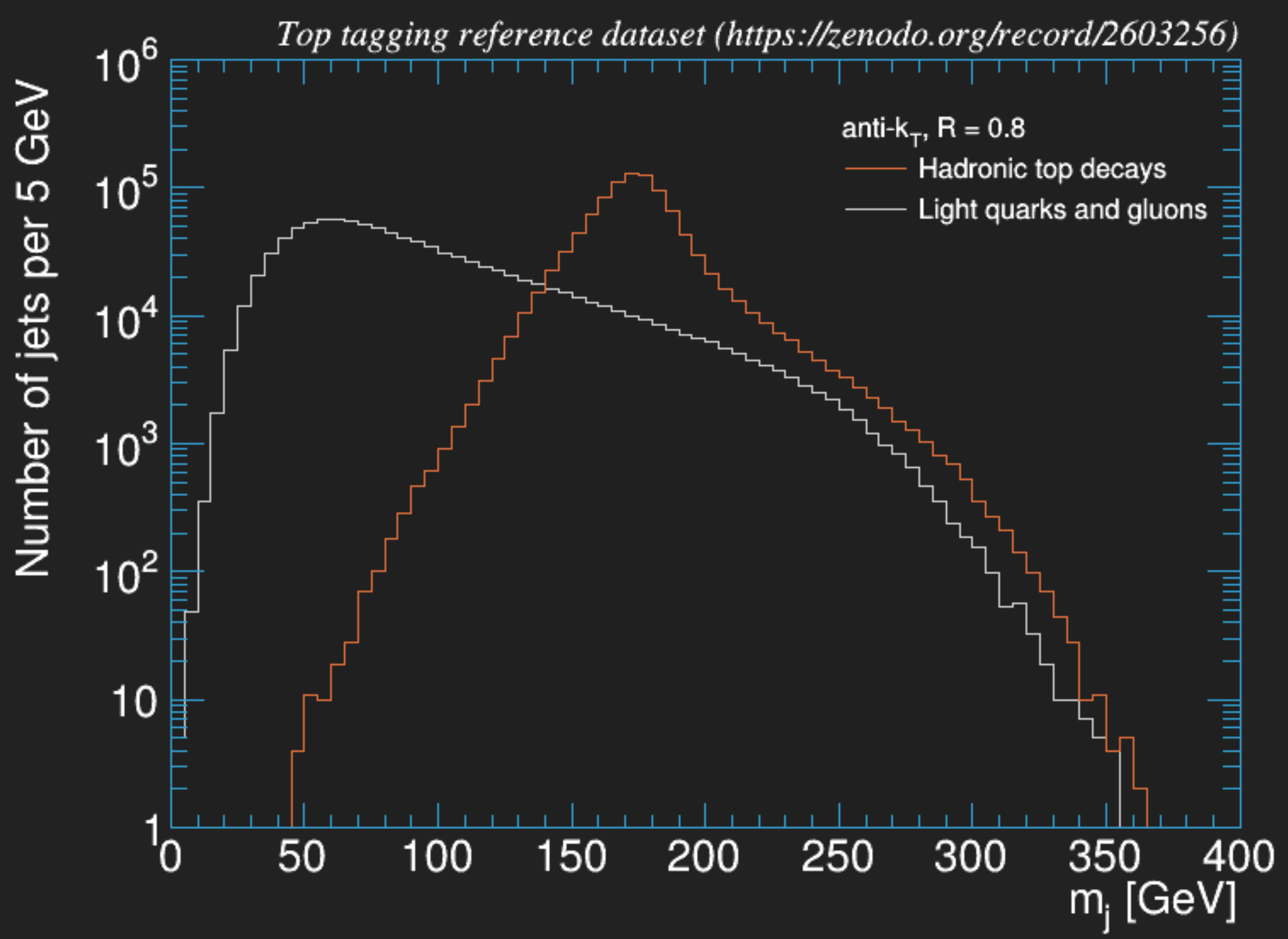
JET η & ϕ


Top tagging reference dataset (https://zenodo.org/record/2603256)
Number of jets per bin
anti-kT, R = 0.8
Hadronic top decays
Light quarks and gluons
η

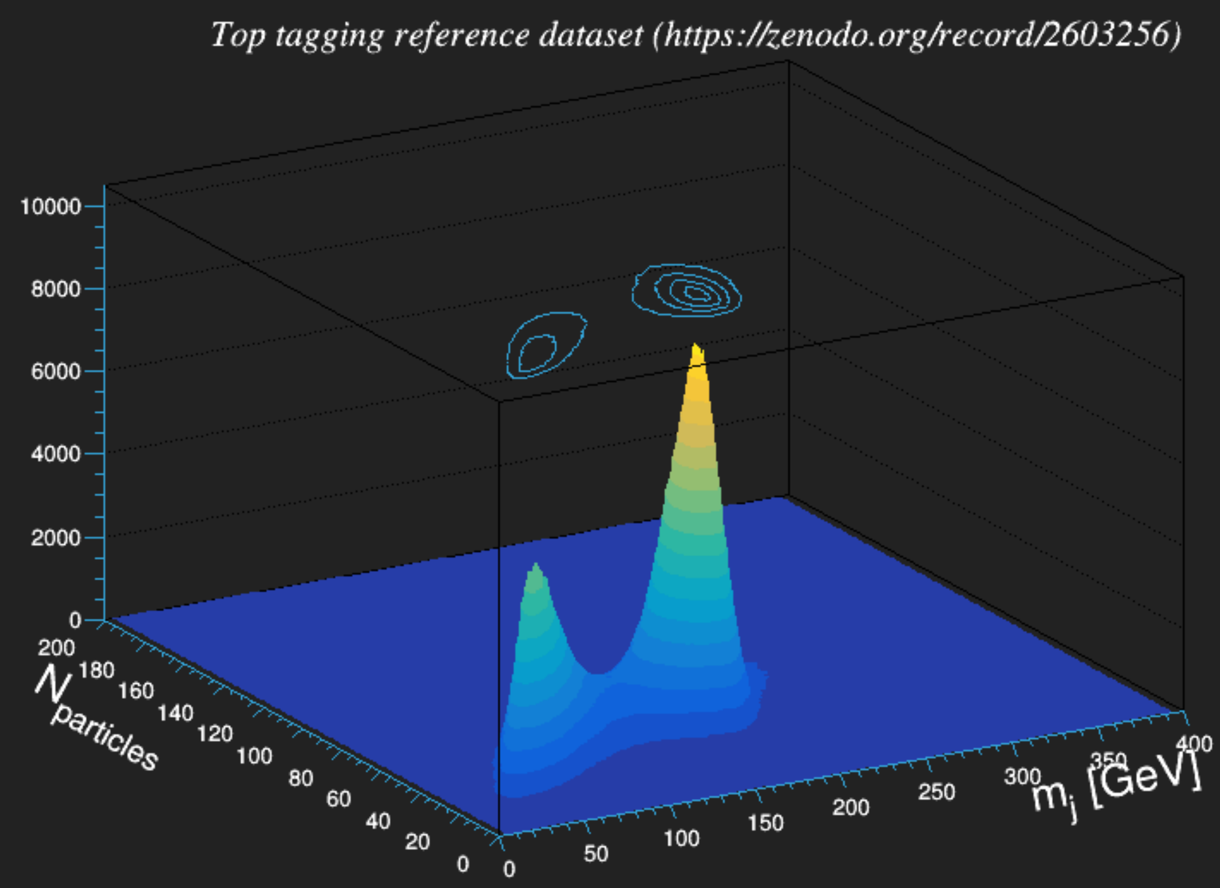

Top tagging reference dataset (https://zenodo.org/record/2603256)
Number of jets per bin
anti-kT, R = 0.8
Hadronic top decays
Light quarks and gluons
ϕ

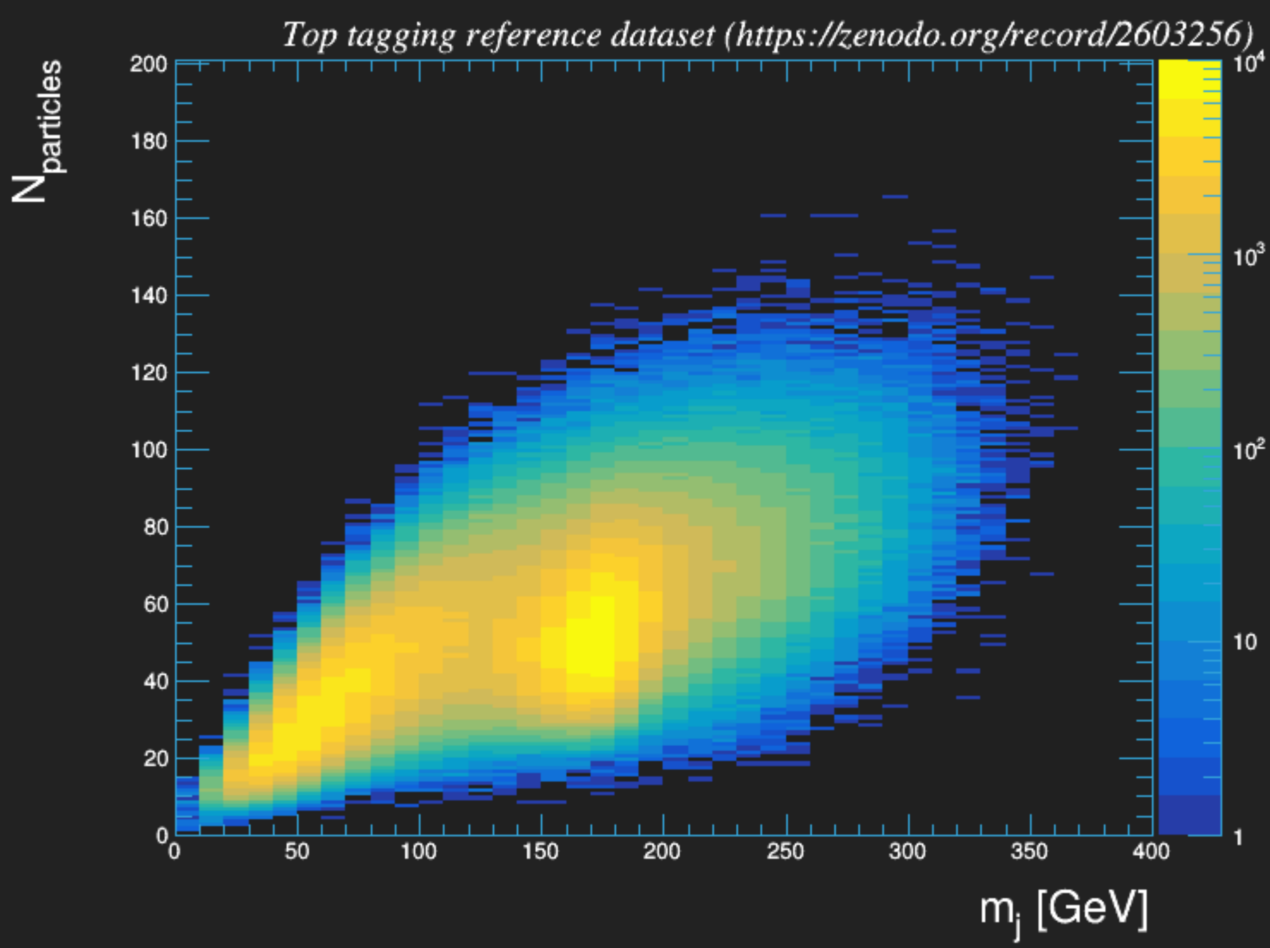
JET NUMBER OF CONSTITUENTS & m



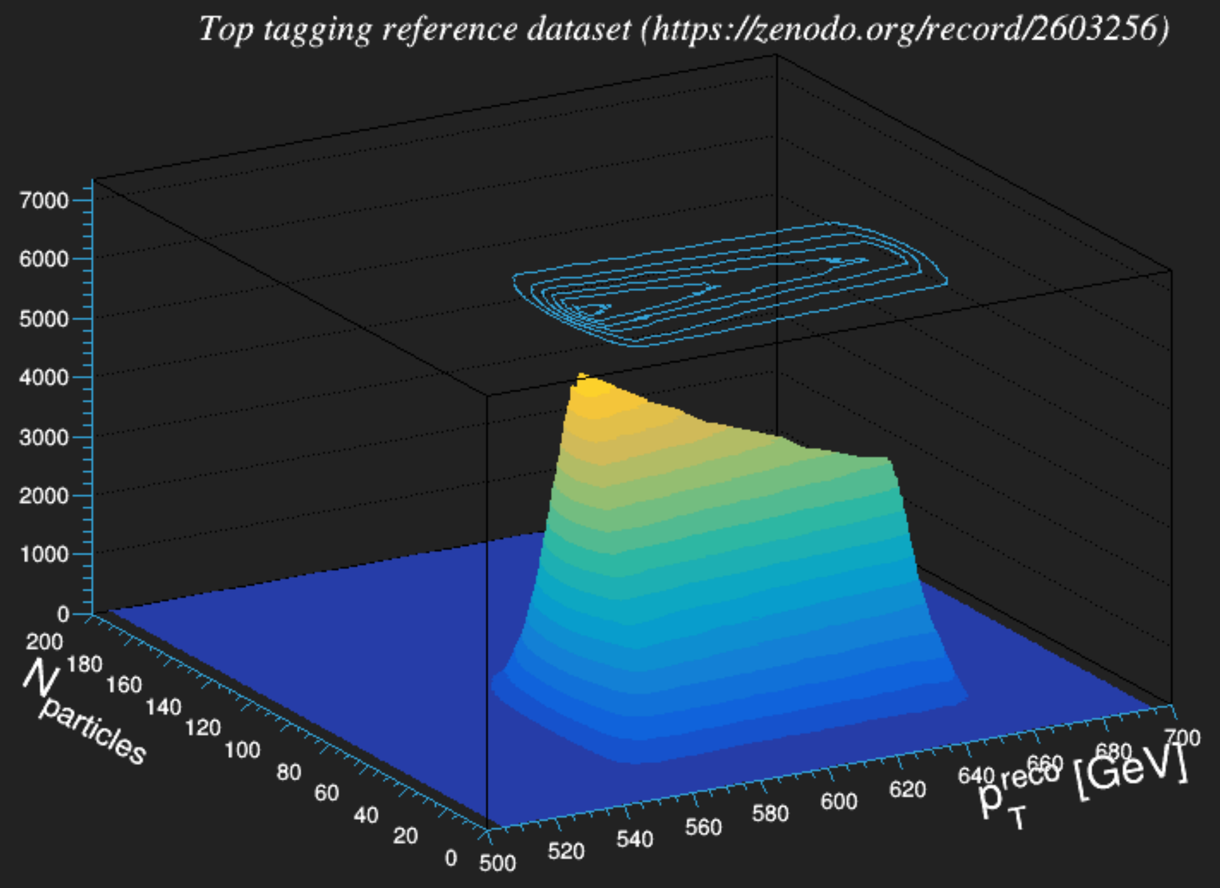


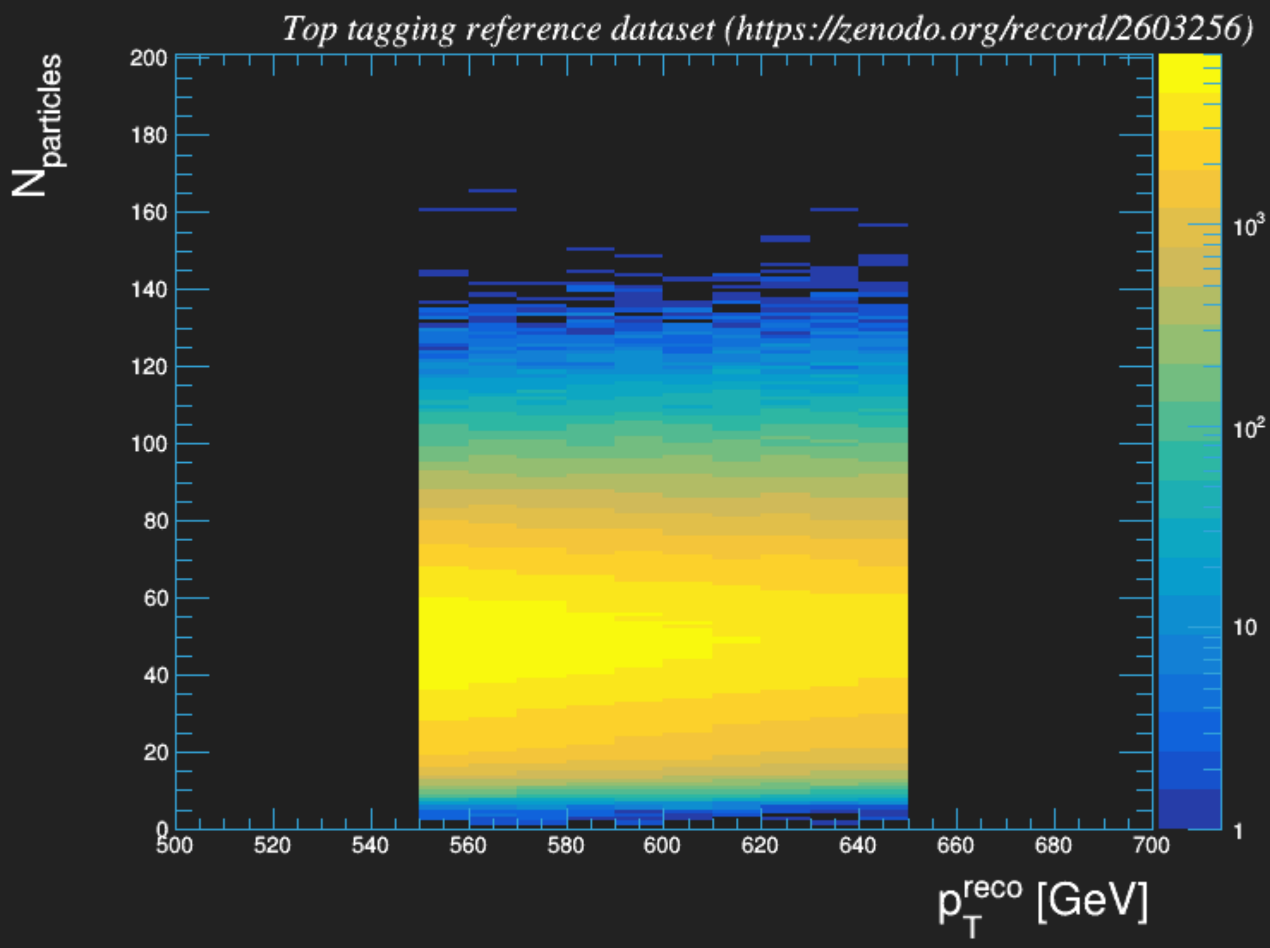
JET NUMBER OF CONSTITUENTS VS. m



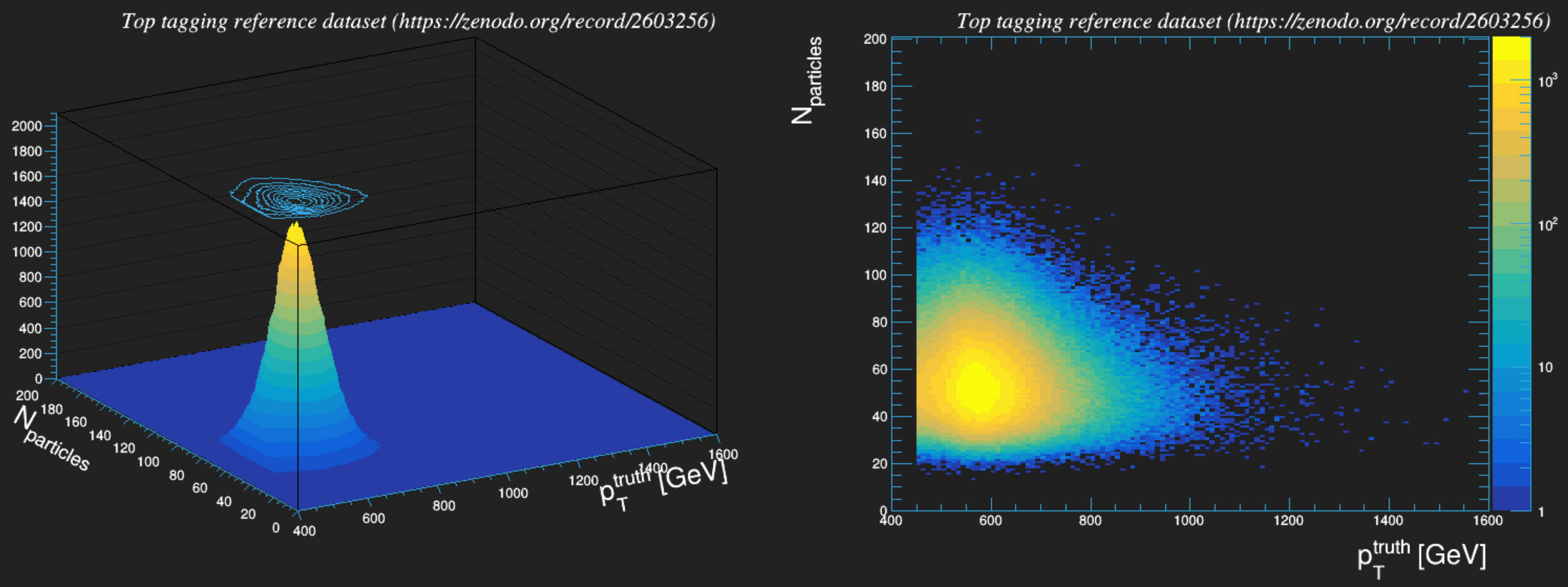


JET NUMBER OF CONSTITUENTS VS. p_T

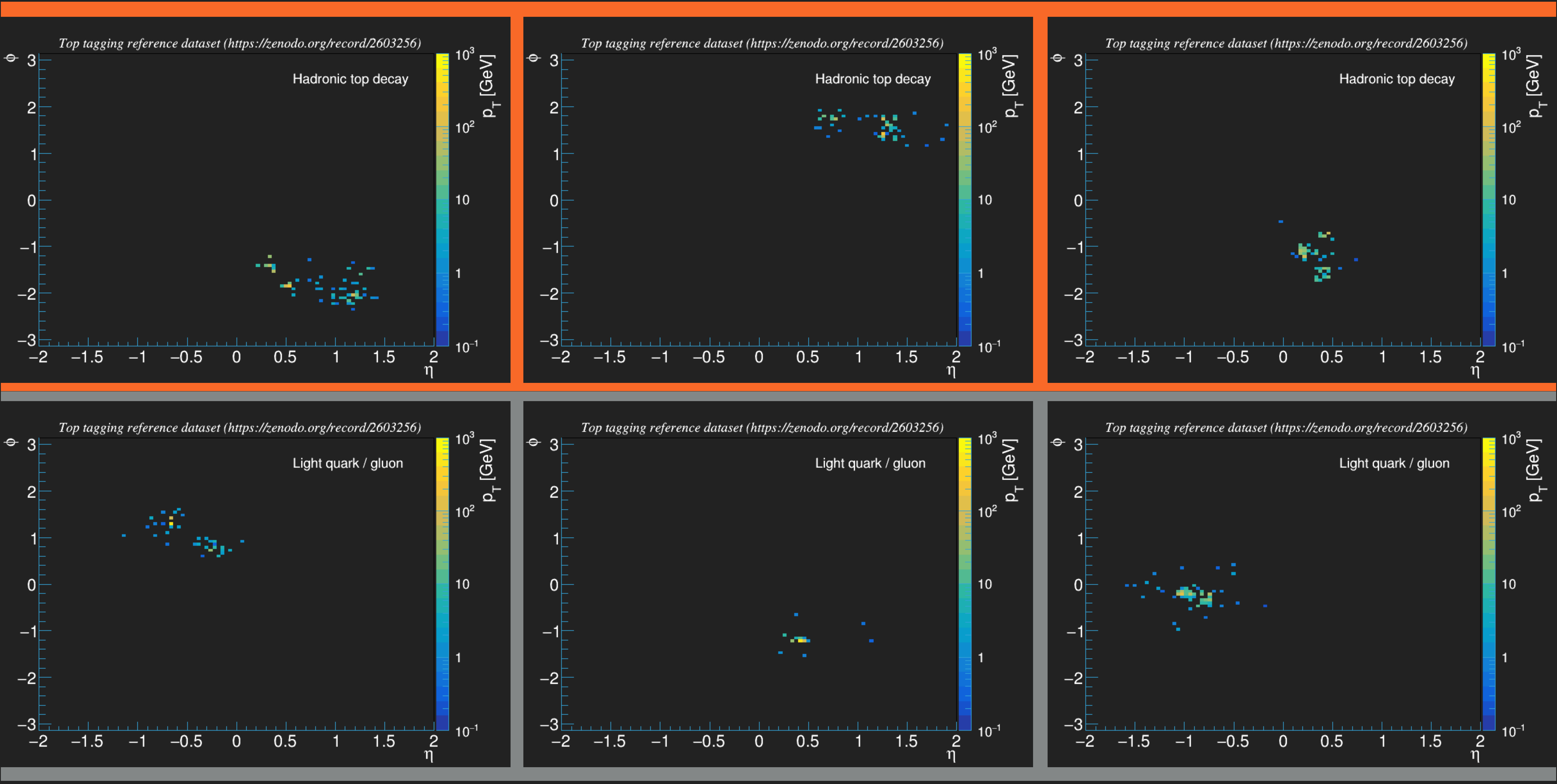




JET NUMBER OF CONSTITUENTS VS. p_T

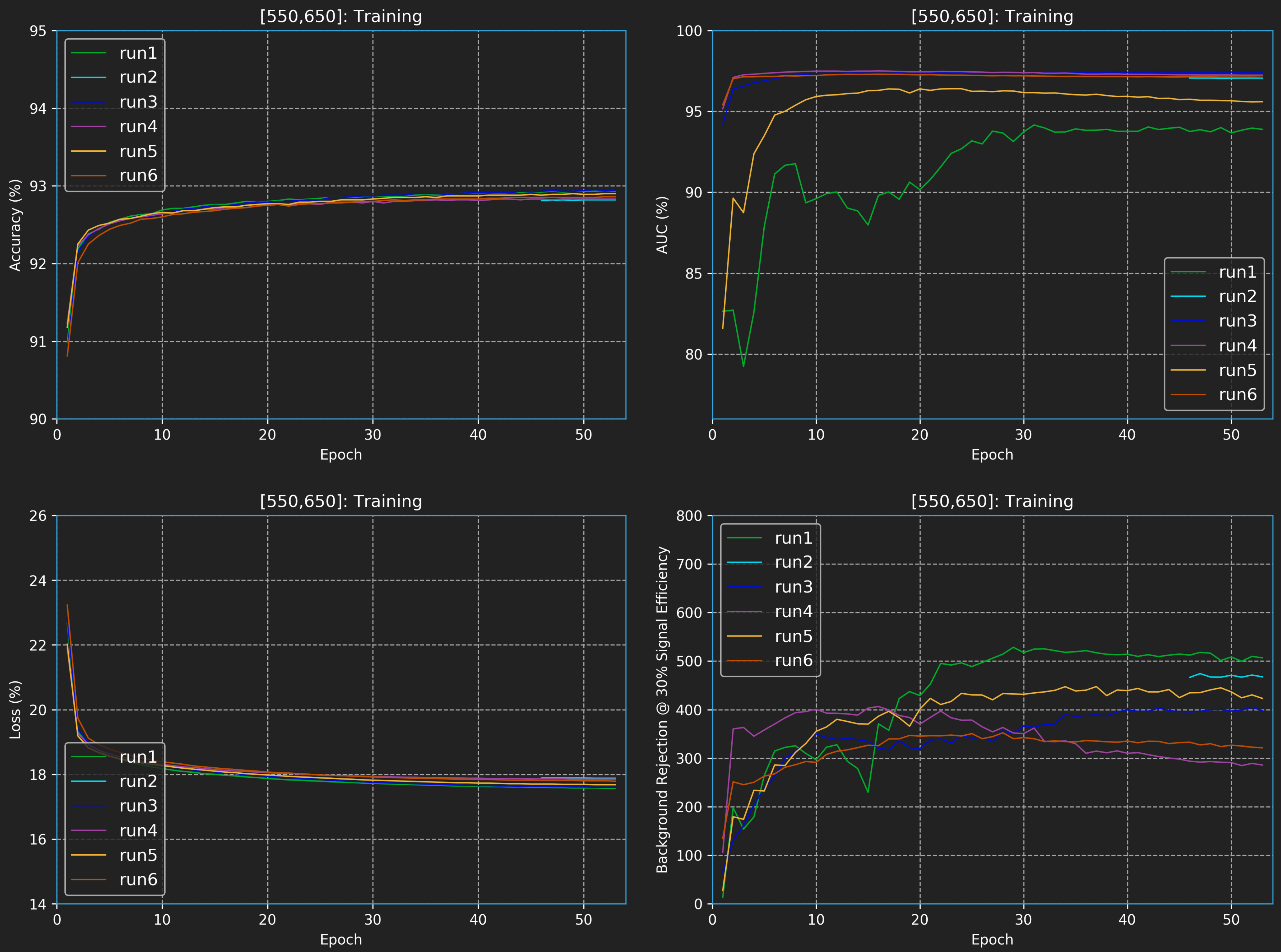


SAMPLE EVENT DISPLAYS (SIGNAL & BACKGROUND)

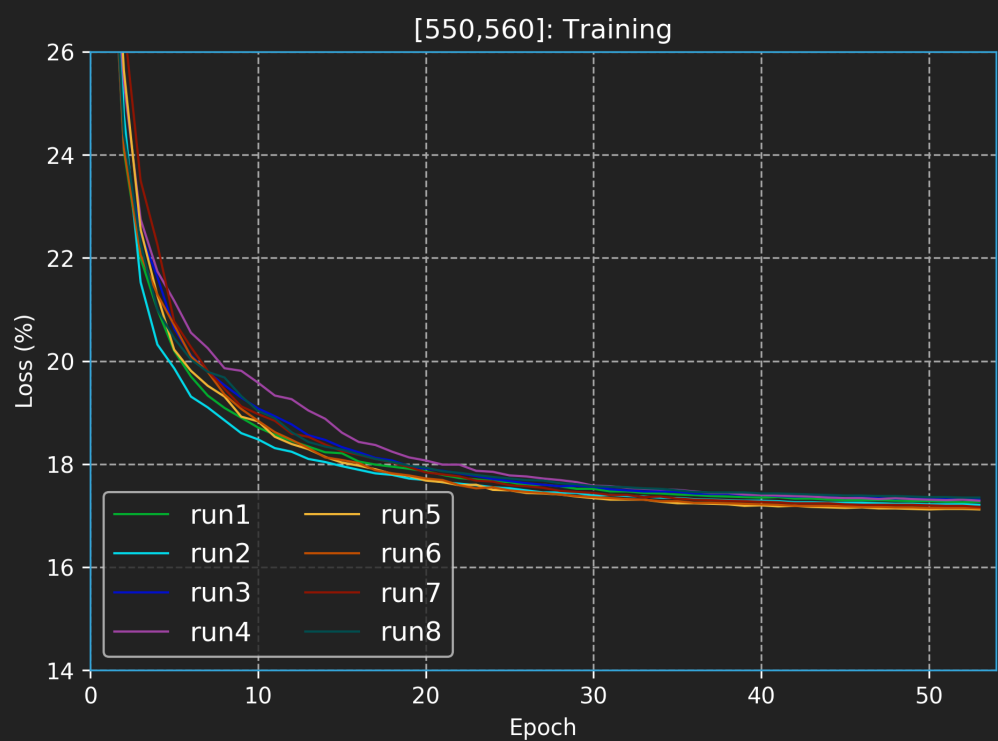


TRAINING ON THE FULL DATASET

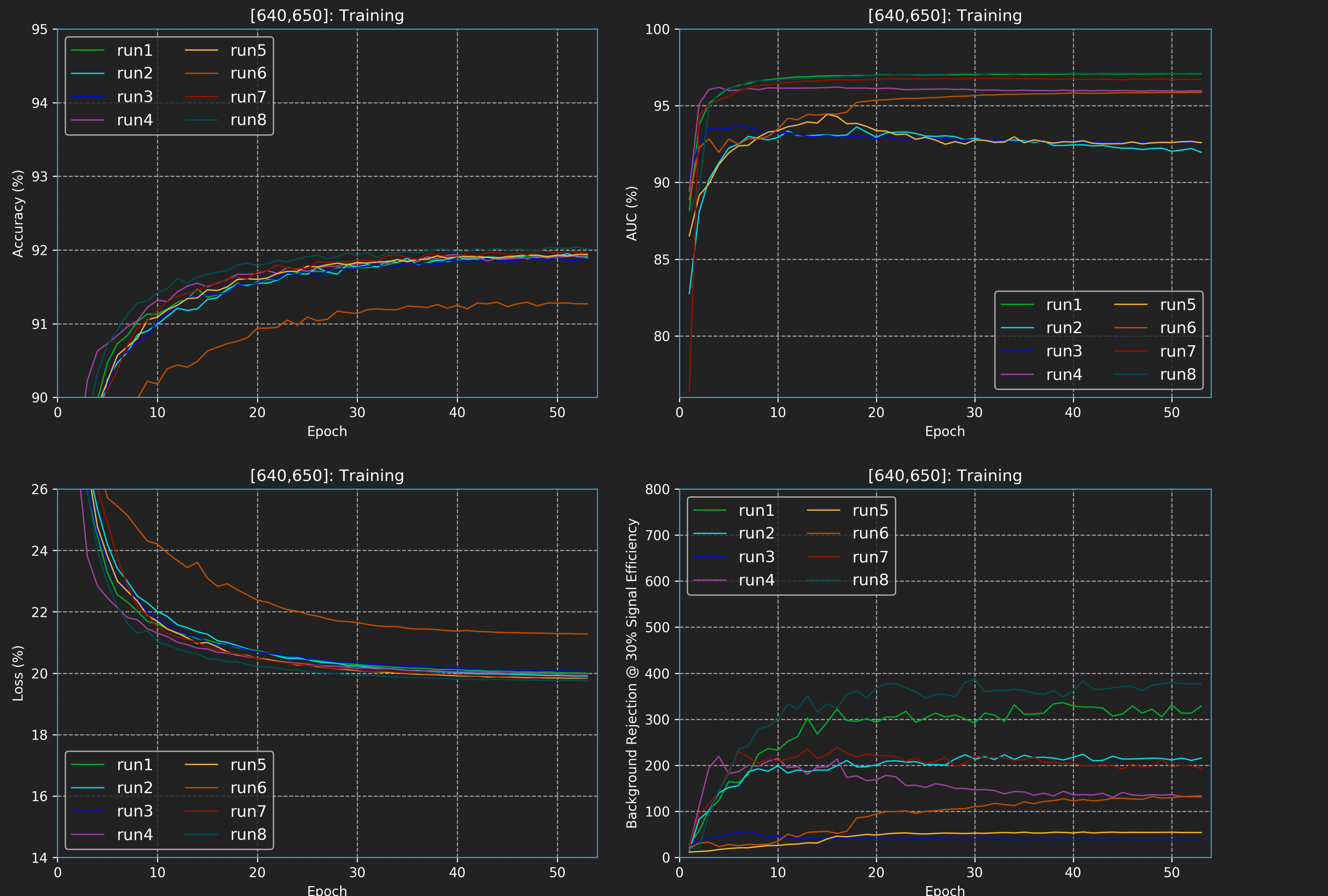
Note: Data from earlier epochs missing for run2.



TRAINING ON $p_T \in [550,560]$ GeV

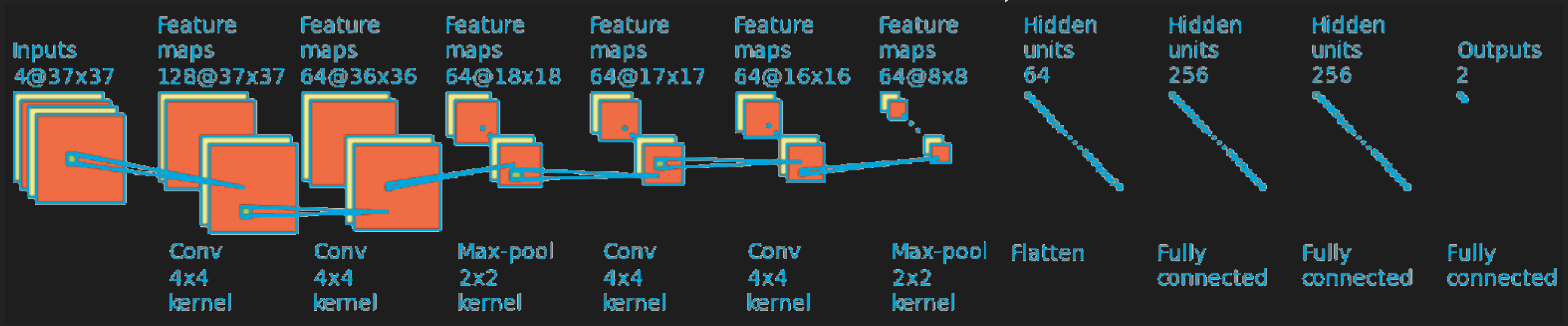


TRAINING ON $p_T \in [640,650]$ GeV



CNN

- Uses jet images (in (η, ϕ)).
  - Images are pre-processed by centering on the jet's p_T-weighted centroid, rotating so that the 2nd highest-intensity cluster is along the vertical axis.
- Takes advantage of multiple channels $\left(p_T^{\text{neutral}},\ p_T^{\text{track}},\ N^{\text{track}},\ N^{\text{muon}}\right)$.



[Macaluso, S., Shih, D. (Pulling Out All the Tops with Computer Vision and Deep Learning, 2018)]

RESNEXT

- Uses jet images (in (η, ϕ)).
  - Images are pre-processed by centering on the jet's p_T -weighted centroid.
- An "out-of-the-box" application of an image-identification.

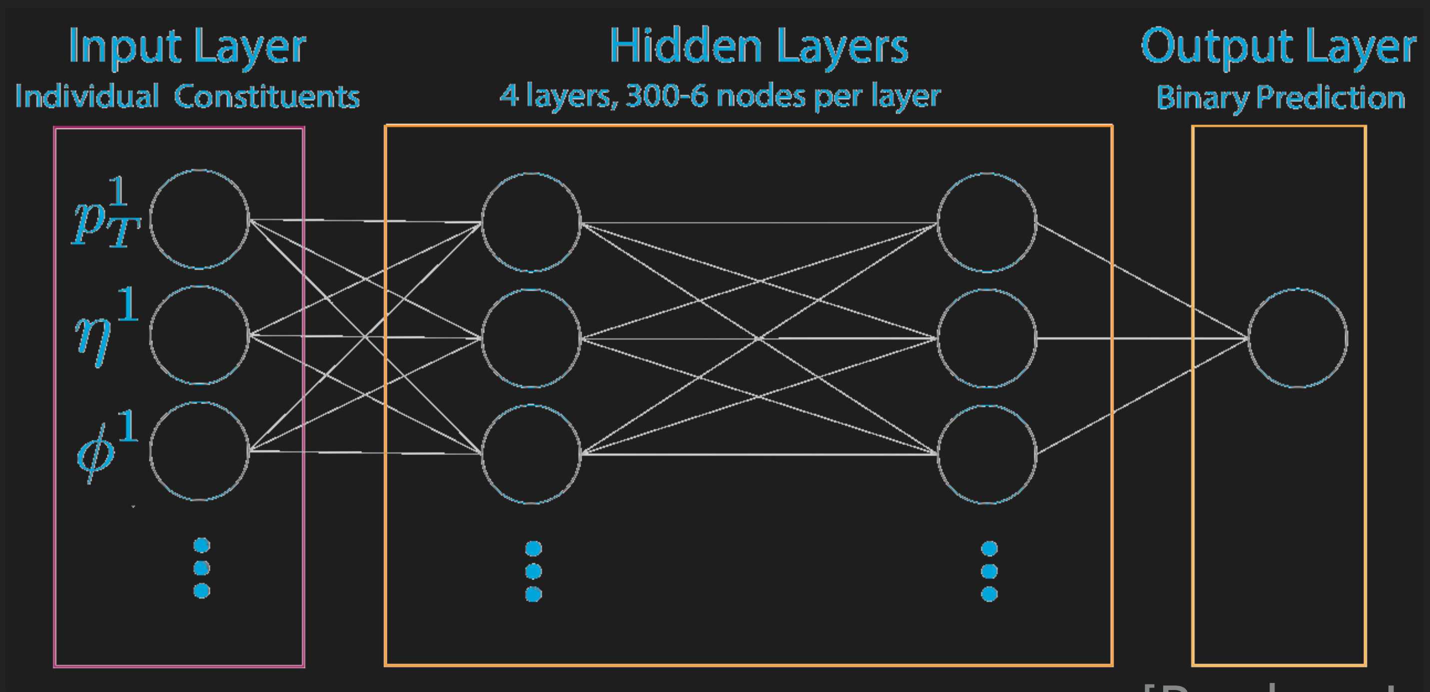
[Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K. (ResNeXt, 2016)]

TOPODNN

- Deep neural network using jet constituents' 4-momenta components (p_T, η, ϕ) as inputs.

- Jets are preprocessed by translation in (η, ϕ) to center the leading subjet. Then the momenta are transformed:

$$p'_{y,n} = p_{y,n} \cos\theta - p_{z,n} \sin\theta\ ,\ \ p'_{z,n} = p_{y,n} \sin\theta - p_{z,n} \cos\theta\ ,\ \ \text{with } \theta = \arctan\left(\frac{p_{y,2}}{p_{z,2}}\right) + \frac{\pi}{2}\ .$$



[Pearkes, J., Fedorko, W., Lister, A., Gay, C. (Jet Constituents for Deep Neural Network Based Top Quark Tagging, 2017)]

MULTI-BODY N-SUBJETINESS

- Dense neural network.
- Uses a family of N-subjetiness variables $\{\tau_i^{(\alpha)}\}$ as input.[*]

$$\tau_N^{(\beta)} = \frac{1}{p_{T,J}} \sum_{i \in J} p_{T,i} \min \left\{ R_{1i}^{\beta}, R_{2i}^{\beta} \ldots R_{Ni}^{\beta} \right\}$$

- No (unphysical) pre-processing of data necessary.
  - $\{\tau_i^{(\alpha)}\}$ may need to be calculated, but these are well-understood high-level variables.

* [Thaler, J., Van Tilburg, K. (Identifying Boosted Objects with N-subjettiness, 2010)]
[Moore, L., Nordström, K., Varma, S., Fairbairn, M. (Multi-body N-subjettiness, 2018)]

TREE NIN

- Uses a "tree neural network" structure.[*]
- Jets are restructured as binary trees, with each node carried a set of features $\left(|\vec{p}|, \eta, \phi, E, E_{\text{frac}}, p_T, \theta \right)$.
- The "Network in Network" structure allows for fully-connected layers in each binary tree node.

* [Roy, D., Priyadarshini, P., Roy K. (Tree-CNN: A Hierarchical Deep Convolutional Neural Network for Incremental Learning, 2018)]
[Macaluso, S., Cranmer, K (Tree Network in Network (TreeNiN) for Jet Physics, 2019)]

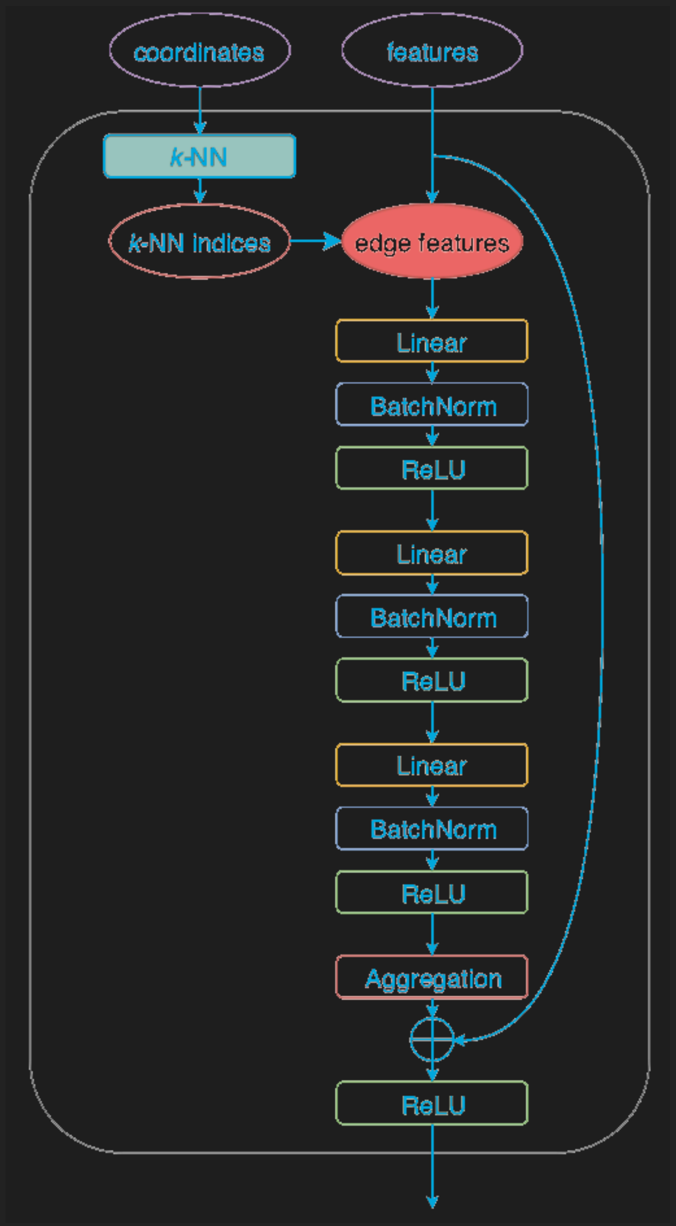
P-CNN

- One-dimensional CNN, based on a DNN used by CMS.[*] Similar to ResNet, but uses 1D convolutions.
- Jets are input as a p_T-ordered list of the 100 leading constituents.
- For each jet, P-CNN computes input features $\left\{ \log p_T^i,\, \log E^i,\, \log(p_T^i/p_t^{\text{jet}}),\, \Delta\eta^i,\, \Delta\phi^i,\, \Delta R^i \right\}$, with angular distances computed with respect to the jet axis.

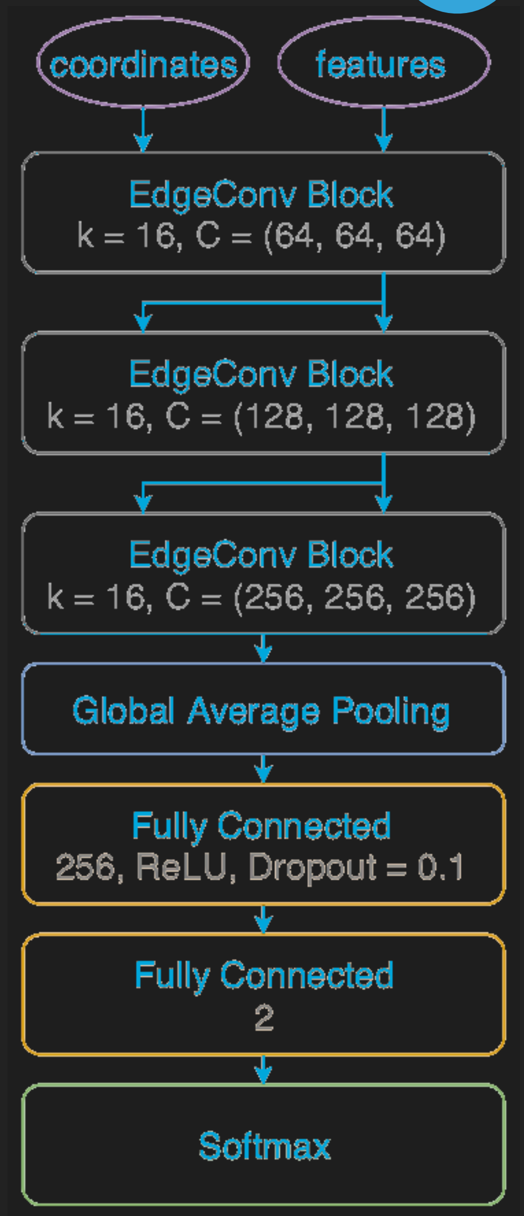
*[The CMS Collaboration (Boosted jet identification using particle candidates and deep neural networks, 2017)]
[Kasieczka, G., Plehn, T., et. al. (ML Landscape of top taggers, 2019)]

PARTICLENET

- Deep graph CNN.
- Jets are represented as unordered sets of particles.
- A graph is constructed for each jet, with particles as vertices. Edges connect each constituent to its k nearest neighbors in (η, ϕ).
- Edge convolutions[*] are applied to the graph, with graph distances updated after each convolution.



EdgeConv



ParticleNet

*[Wang, Y., Sun, Y., Liu, Z., Sarma, S., Bronstein, M. M., Solomon, J . M. (Dynamic Graph CNN for Learning on Point Clouds, 2018)]
[Qu, H., Gouskos, L. (ParticleNet, 2019)]

LORENTZ BOOST NETWORK

- Jet constituents are input as four-momenta.
  - The jets are pre-processed by anti-k_T reclustering, with $\Delta R = 0.2$, to provide a consistent constituent ordering.
- An intermediate layer treats half the inputs as constituents and the other half as rest frames, into which the constituents are boosted.
- An output layer computes a set of features from the boosted constituents, (E, m, p_T, ϕ, η), as well as cosine of the angles between all boosted constituents.

[Erdmann, M., Geiser, E., Rath, Y., Rieger, M. (Lorentz Boost Networks, 2018)]

LORENTZ LAYER

- Jet constituents are input as four-momenta.
- A combination layer (CoLa) linearly combines the momenta: $k_{\mu,i} \to \tilde{k}_{\mu,j} = k_{\mu,i} C_{ij}$.
- The Lorentz Layer (LoLa) calculates a set of Lorentz invariants: $k_j \to \hat{k}_j = \left\{ m^2(\tilde{k}_j),\ p_T(\tilde{k}_j),\ w_{jm}^{(E)} E(\tilde{k}_m),\ w_{jm}^{(m^2)} m^2(\tilde{k}_m),\ w_{jm}^{(d)} d_{jm}^2 \right\}$.

[Butter, A., Kasieczka, G., Plehn, T., Russell, M. (LoLa – Lorentz Layer, 2018)]

LATENT DIRICHLET ALLOCATION

- Jets are input as a series of subjets, which are produced by Cambrige-Aachen clustering, followed by sequential de-clustering $j_0 \to j_1 j_2$.
  - For each subjet j_0, inputs are given as the observables
$$\left\{ m_{j0},\ \frac{m_{j1}}{m_{j0}},\ \frac{m_{j2}}{m_{j1}},\ \frac{\min(p_{T,1}^2, p_{T,2}^2)}{m_{j0}^2}\Delta R_{1,2}^2 \right\}.$$
- The likelihood of generating jet $j = \{o_1, o_2, \ldots o_n\}$ is modeled by
$$p(j\,|\,\alpha,\beta) = \int_\omega p(\omega\,|\,\alpha) \prod_{o \in j} \left(\sum_t p(t\,|\,\omega) p(o\,|\,t,\beta) \right) \mathrm{d}\omega.$$

  theme (β) — *hyper-parameter* (α) — *theme proportion* (ω)

  - Model does not account for $p(o_i\,|\,o_{i-1})$.
- A neural network is trained to invert the above expression to find (β, ω).

[Dillon, B. M., Faroughy, D. A., Kamenik, J, F. (Uncovering latent jet substructure, 2019)]

ENERGY FLOW POLYNOMIALS

- For a jet with M constituents and a multigraph G with N vertices and edges $(k, l) \in G$, the corresponding EFP is

$$\text{EFP}_G = \sum_{i_1=1}^{M} \ldots \sum_{i_N=1}^{M} z_{i_1} \ldots z_{i_N} \prod_{(k,l)\in G} \theta_{i_k i_l}, \text{ with } z_i \equiv \frac{E_i}{\sum_{j=1}^{M} E_j}.$$

- EFP's form a *complete linear basis* for jet substructure, so that any IRC-safe observable can be computed as $S \simeq \sum_{G\in\mathcal{G}} s_G \, \text{EFP}_G$.
  - In practice, one truncates $\mathcal{G}$ via a max number of edges.
- EFP's can be used in linear regression or as DNN inputs.

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

ENERGY/PARTICLE FLOW NETWORKS

- IRC-safe observables can be approximated as $F\left(\sum_{i=1}^{M}(z_i)\Phi(\hat{p}_i)\right)$, where:
  - $z_i = \{E_i, p_{T,i}\}$ for EFN, $z_i = 1$ for PFN, $\hat{p}_i = \frac{\vec{p}}{|\vec{p}|}$.
  - $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^l$ is a per-particle mapping, $F : \mathbb{R}^l \rightarrow \mathbb{R}$ is a continuous function.
    - These can be parametrized as neural network layers for complicated observables.

[Komiske, P. T., Metodiev, E. M., Thaler, J.
(Energy Flow Networks: Deep Sets for Particle Jets, 2018)]

One formulation (from Kondor & Trivedi):

Let $\{\rho(g) : U \to U\}_{g \in G}$ and $\{\rho'(g) : V \to V\}_{g \in G}$ *be two irreducible representations of a compact group* G.

Let $\phi : U \to V$ *be an equivariant linear mapping for these reps, i.e.* $\phi\left(\rho(g)(u)\right) = \rho'(g)\big(\phi(u)\big)\, \forall\, u \in U$.

Then, unless ϕ *is the zero map,* ρ *and* ρ' *are equivalent representations.*

[Kondor, R., Trivedi, S. (On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups, 2018)]

SO(3), SU(2) : $B^{-1} : R_{l_1} \otimes R_{l_2} \to \bigoplus_{l=|l_1-l_2|}^{l_1+l_2} R_l$

$$\left|l_1, m_1\right\rangle \otimes \left|l_2, m_2\right\rangle = \sum_{l,m} B^{l,m}_{l_1,m_1;l_2,m_2} \left|l, m\right\rangle$$

$$(2l_1 + 1)(2l_2 + 1) \times (2l + 1)$$

SO(1,3)$^+$, SL(2,ℂ) : $T^{(k,n)} \simeq \bigoplus_{l=|k-n|/2}^{(k+n)/2} R_l$

$$H^{-1} : \quad T^{(k_1,n_1)} \otimes T^{(k_2,n_2)} \to \bigoplus_{k=|k_1-k_2|}^{k_1+k_2} \bigoplus_{n=|n_1-n_2|}^{n_1+n_2} T^{(k,n)}$$

$$H^{(k_1,n_1),l_1,m_1;(k_2,n_2),l_2,m_2}_{(k,n),l,m} = \sum_{m_1',m_2'} B^{\frac{k}{2},m_1'+m_2';\frac{n}{2},m-m_1'-m_2'}_{l,m} B^{\frac{k_1}{2},m_1';\frac{k_2}{2},m_2'}_{\frac{k}{2},m_1'+m_2'} B^{\frac{n_1}{2},m_1-m_1';\frac{n_2}{2},m_2-m_2'}_{\frac{n}{2},m-m_1'-m_2'} B^{\frac{k_1}{2},m_1';\frac{n_1}{2},m_1-m_1'}_{l_1,m_1} B^{\frac{k_2}{2},m_2';\frac{n_2}{2},m_2-m_2'}_{l_2,m_2}$$

$$(k_1 + 1)(n_1 + 1)(k_2 + 1)(n_2 + 1) \times (k + 1)(n + 1)$$

$$D_{(k,n)}(\alpha, \beta, \gamma) = \left(H^{(k,0),(0,n)}_{(k,n)}\right)^T \cdot \left(D^{k/2}(\alpha, \beta, \gamma) \otimes \overline{D^{n/2}(-\alpha, \beta, -\gamma)}\right) \cdot \left(H^{(k,0),(0,n)}_{(k,n)}\right).$$

[credit to Alex Bogatskiy]