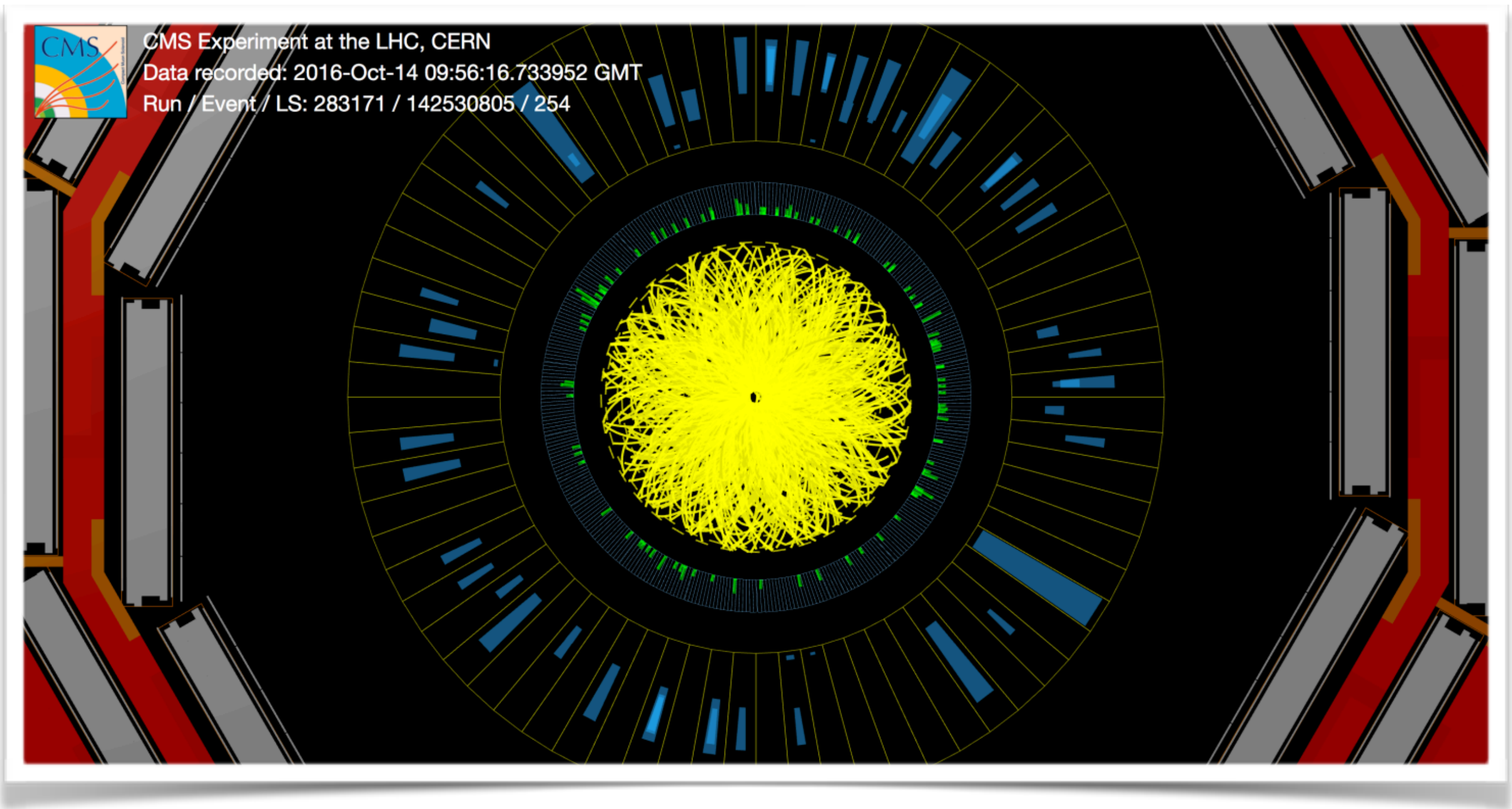
IML Workshop 2018 Challenge Solution
(+ Automatic Feature Engineering)

Marcel Rieger, David Schmidt,
Martin Erdmann, Erik Geiser, Yannik Rath

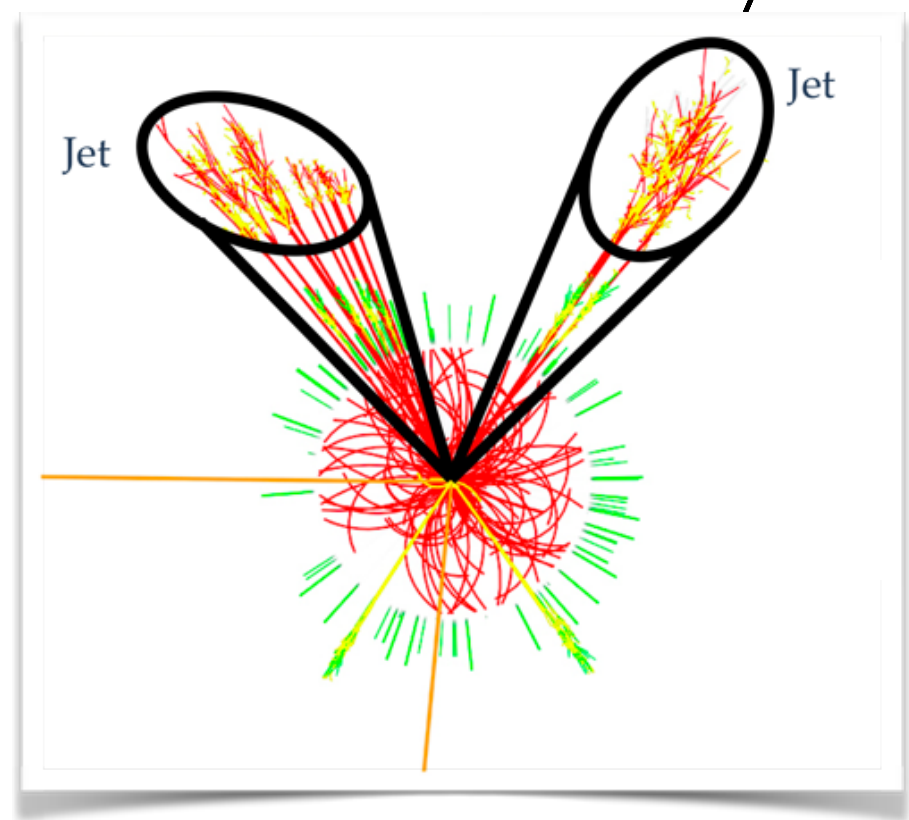
IML Meeting

4 June 2018

- Challenge:
  - “**Regress** the **generator-level** soft drop mass of **reconstructed** high-p_T jets (O(TeV))”

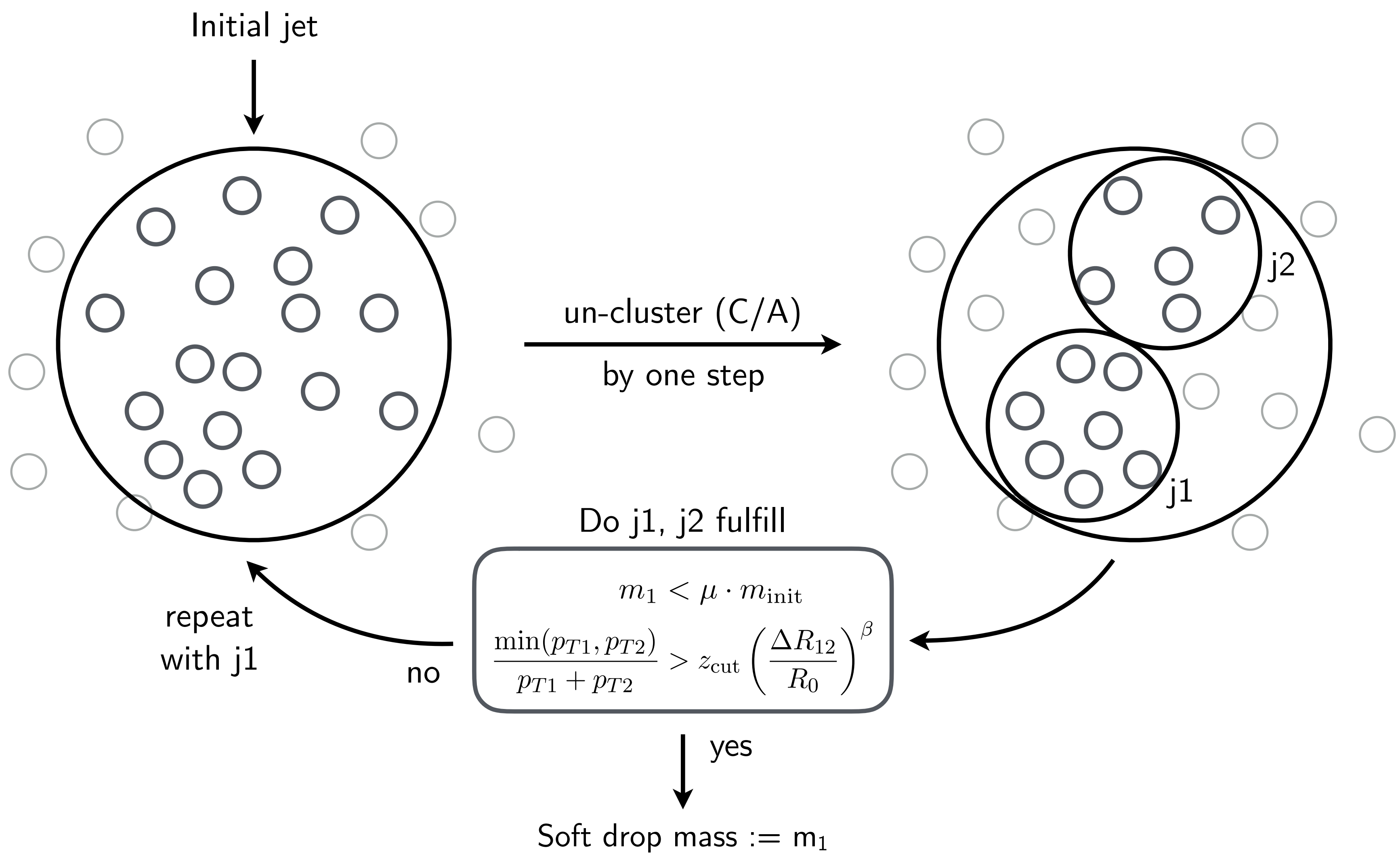


$m_{\text{softdrop,GEN}}$?



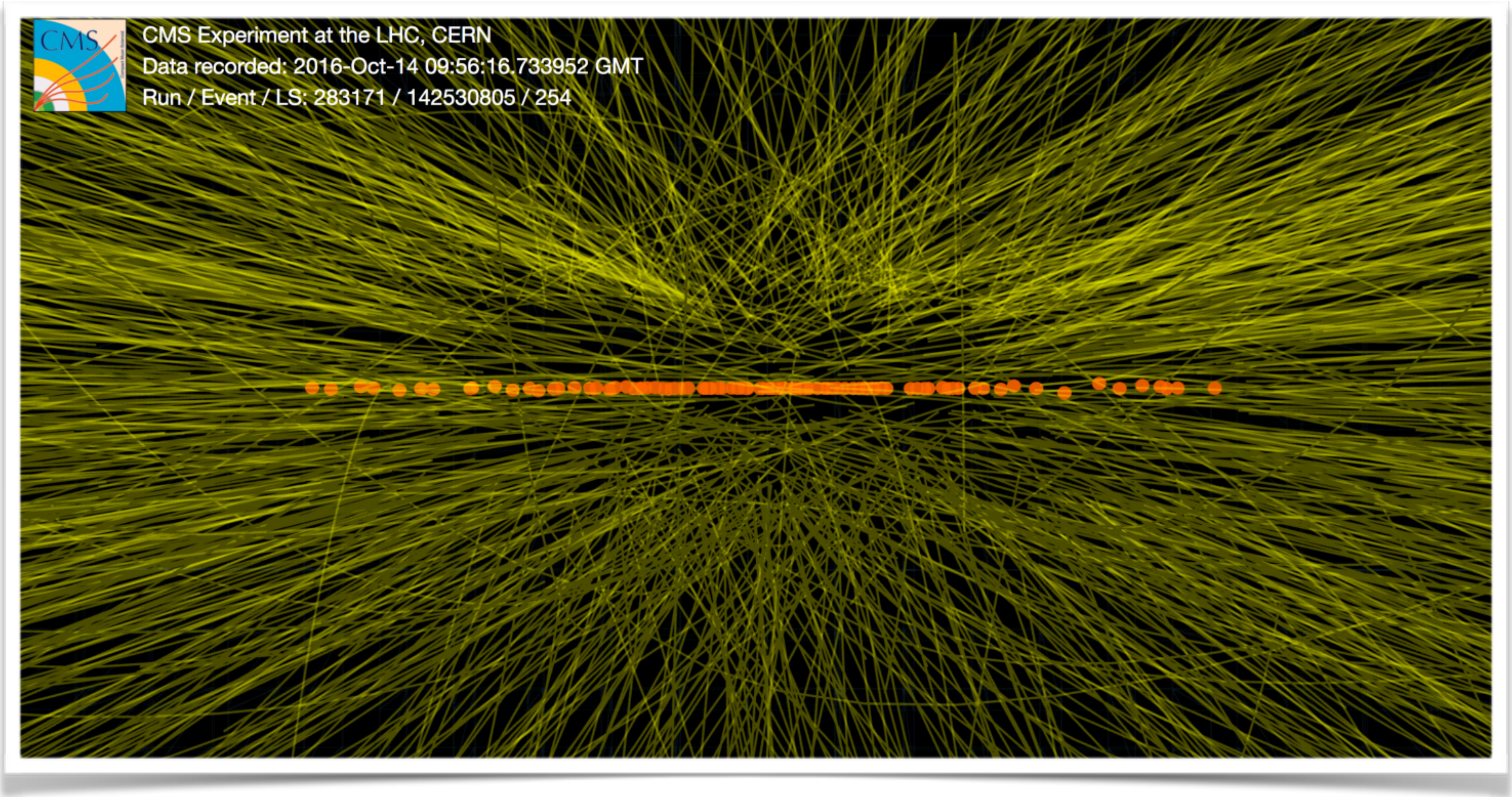
- Conditions:
  - Simulated Future Circular Collider (FCC) conditions
  - Detector simulation via Delphes with CMS phase II config
  - 1000 additional (pile-up) interactions

- Challenge:
  - "**Regress** the **generator-level** soft drop mass of **reconstructed** high-p_T jets (O(TeV))"



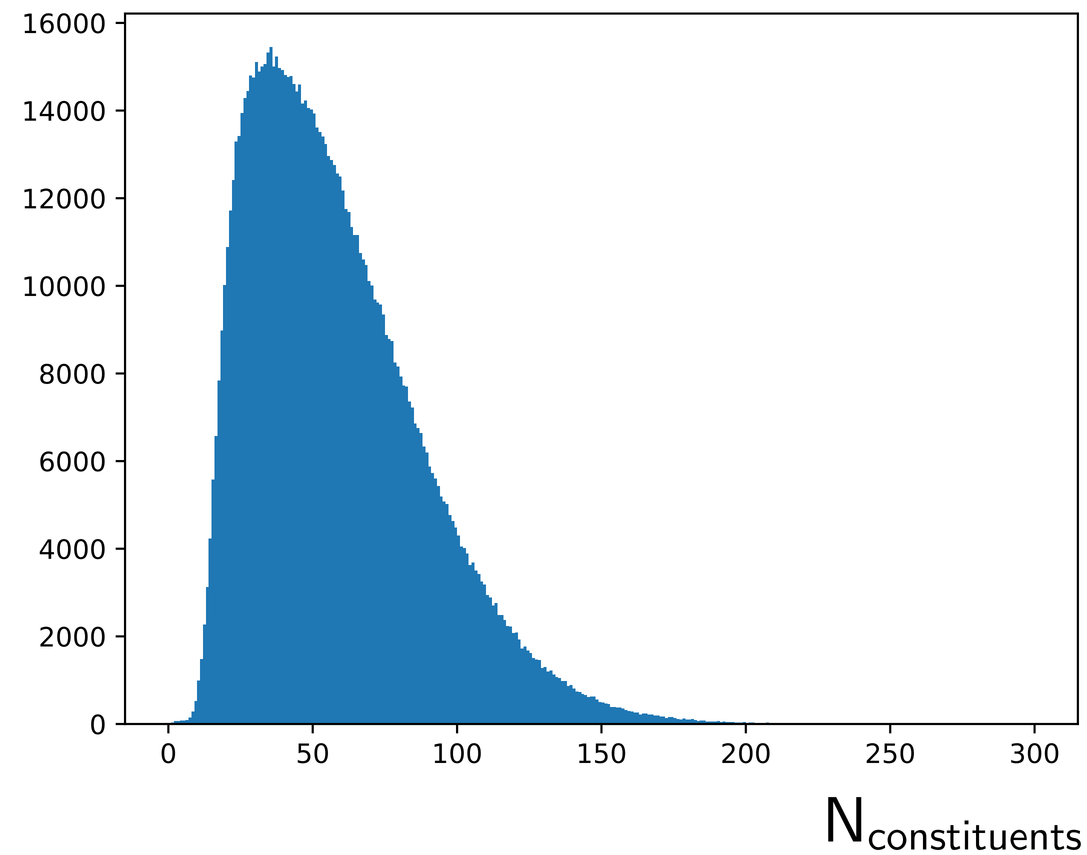
- Conditions:
  - Simulated Future Circular Collider (FCC) conditions
  - Detector simulation via Delphes with CMS phase II config
  - 1000 additional (pile-up) interactions

- Challenge:
  - "**Regress** the **generator-level** soft drop mass of **reconstructed** high-p_T jets (O(TeV))"

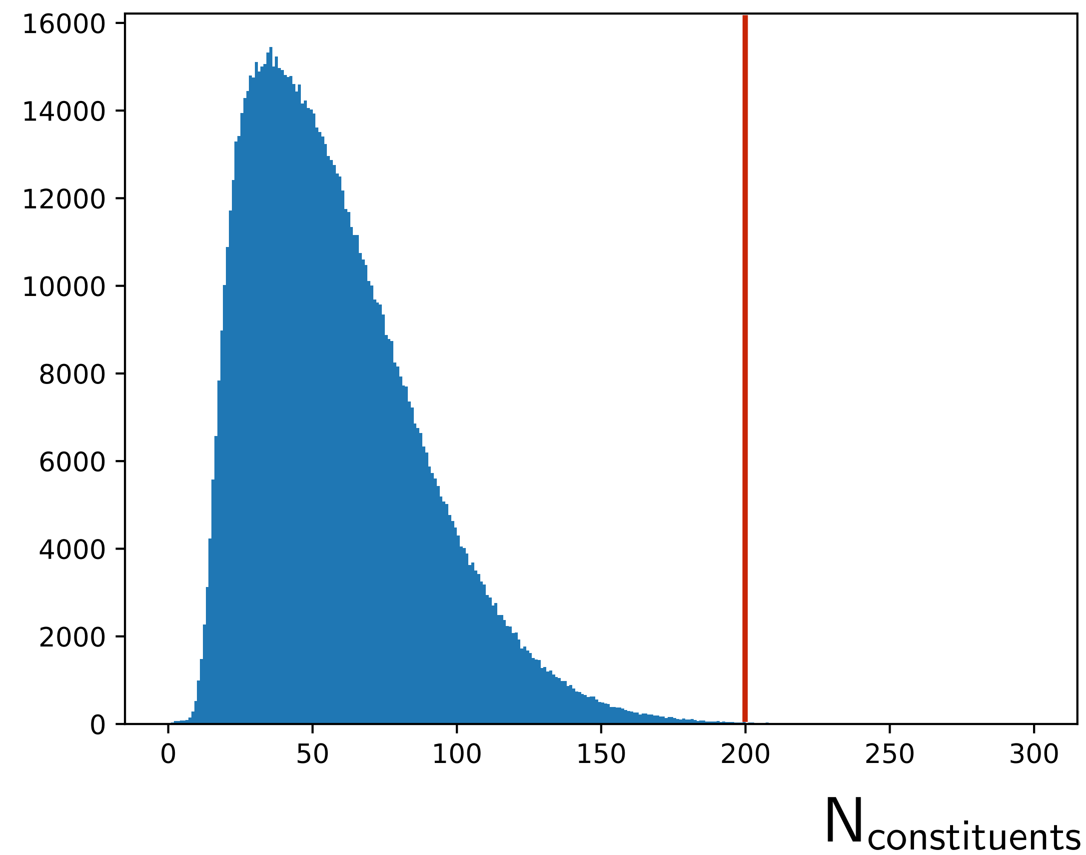


- Conditions:
  - Simulated Future Circular Collider (FCC) conditions
  - Detector simulation via Delphes with CMS phase II config
  - 1000 additional (pile-up) interactions

- 1M jets for training (20% used for validation)
- Input features:
  - Jet + soft drop jet
    - p_T, eta, phi, mass
  - Jet constituents (up to ~300):
    - p_T, eta, phi
    - Charge, d_{xy} + d_z impact parameters
    - Energy deposit in ECAL + HCAL

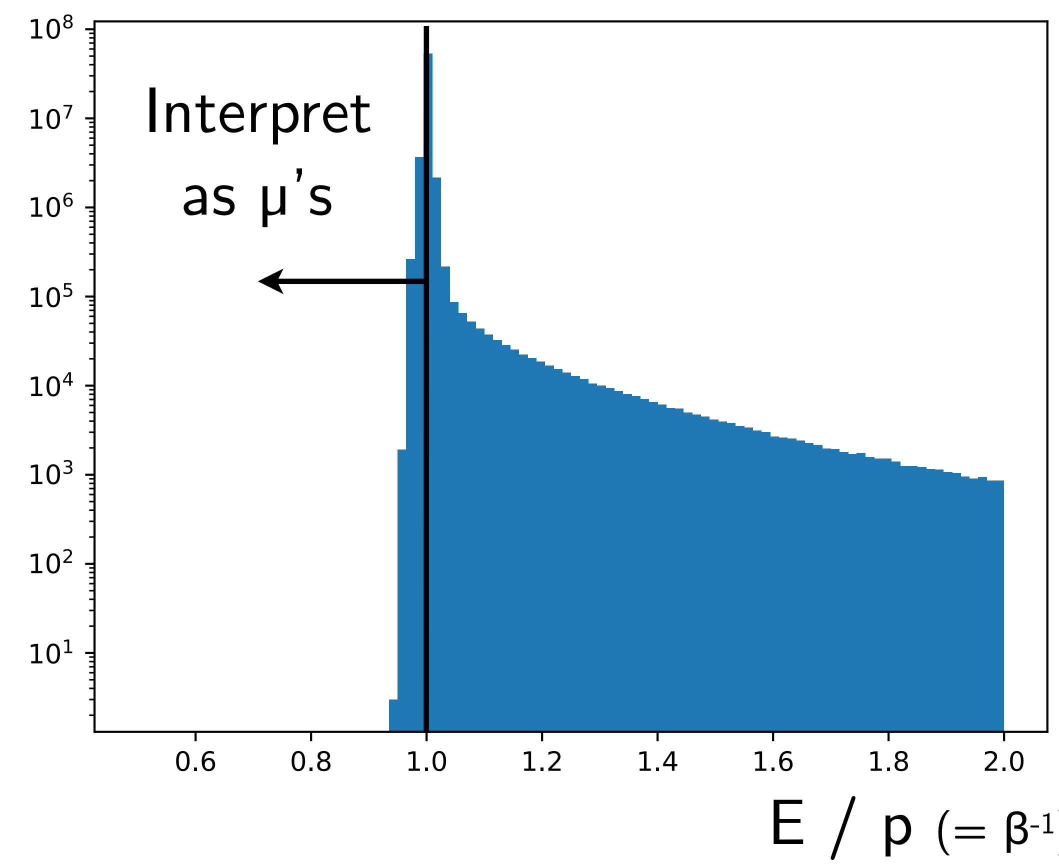


- 1M jets for training (20% used for validation)
- Input features:
  - Jet + soft drop jet
    - p_T, eta, phi, mass
  - Jet constituents (up to ~300):
    - p_T, eta, phi
    - Charge, d_{xy} + d_z impact parameters
    - Energy deposit in ECAL + HCAL



- **Pre-processing of constituents**:
  - Use first N = 200 constituents sorted by p_T
  - Zero-padding when N < 200
  - Determine full 4-vector:
    - $E = E_{ECAL} + E_{HCAL}$
    - Identify μ's via E/p < 1 → use m_μ ($O(m_\pi)$)

!! Only keep 4-vectors, discard charge, etc.



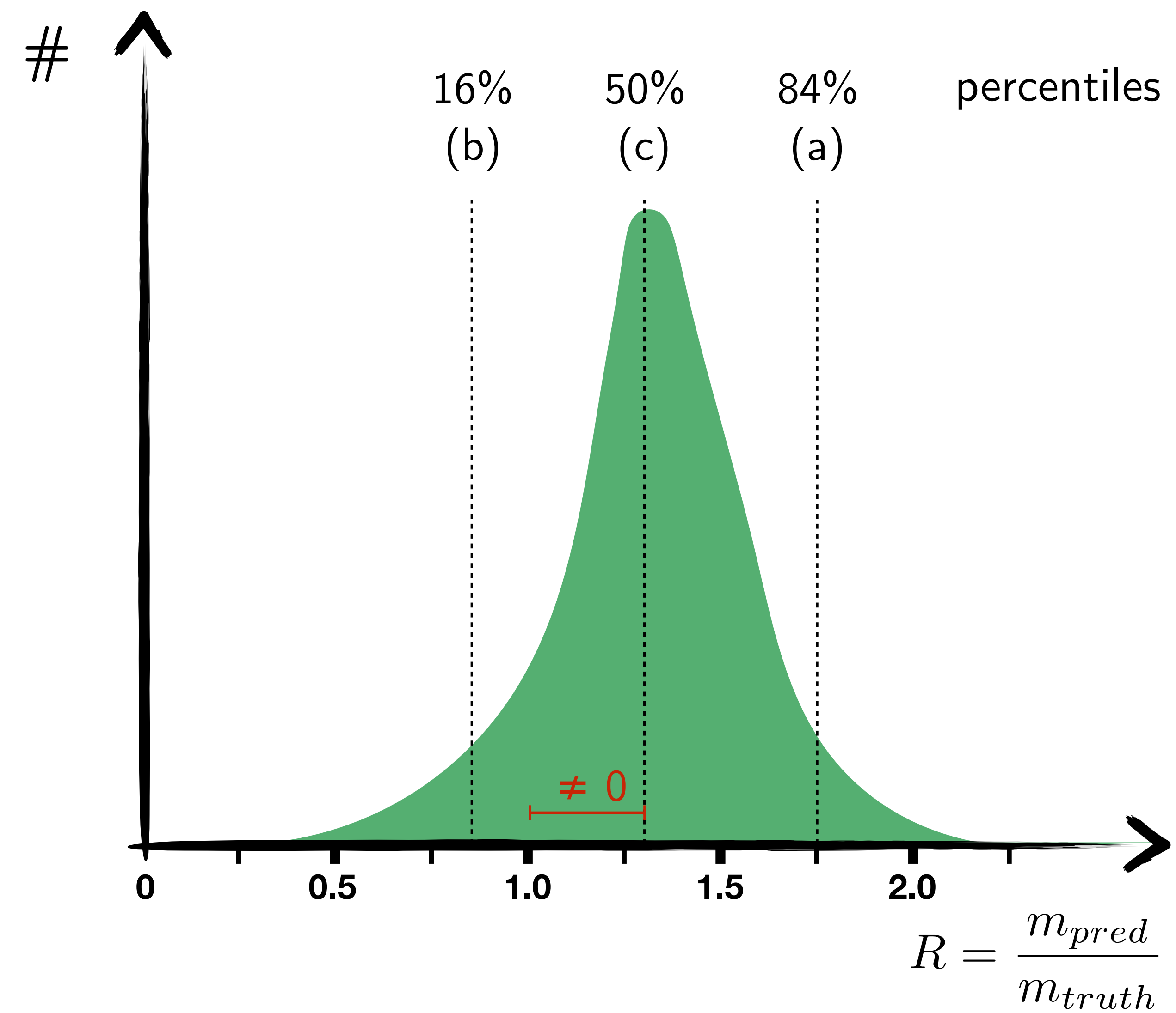
- Network output:
  - Predicted soft drop mass m_{pred}

- Metric:
  - Resolution of truth-normalized masses

$$M = \frac{a-b}{2c}$$

  - Practical definition, overall offset could be subject to calibration

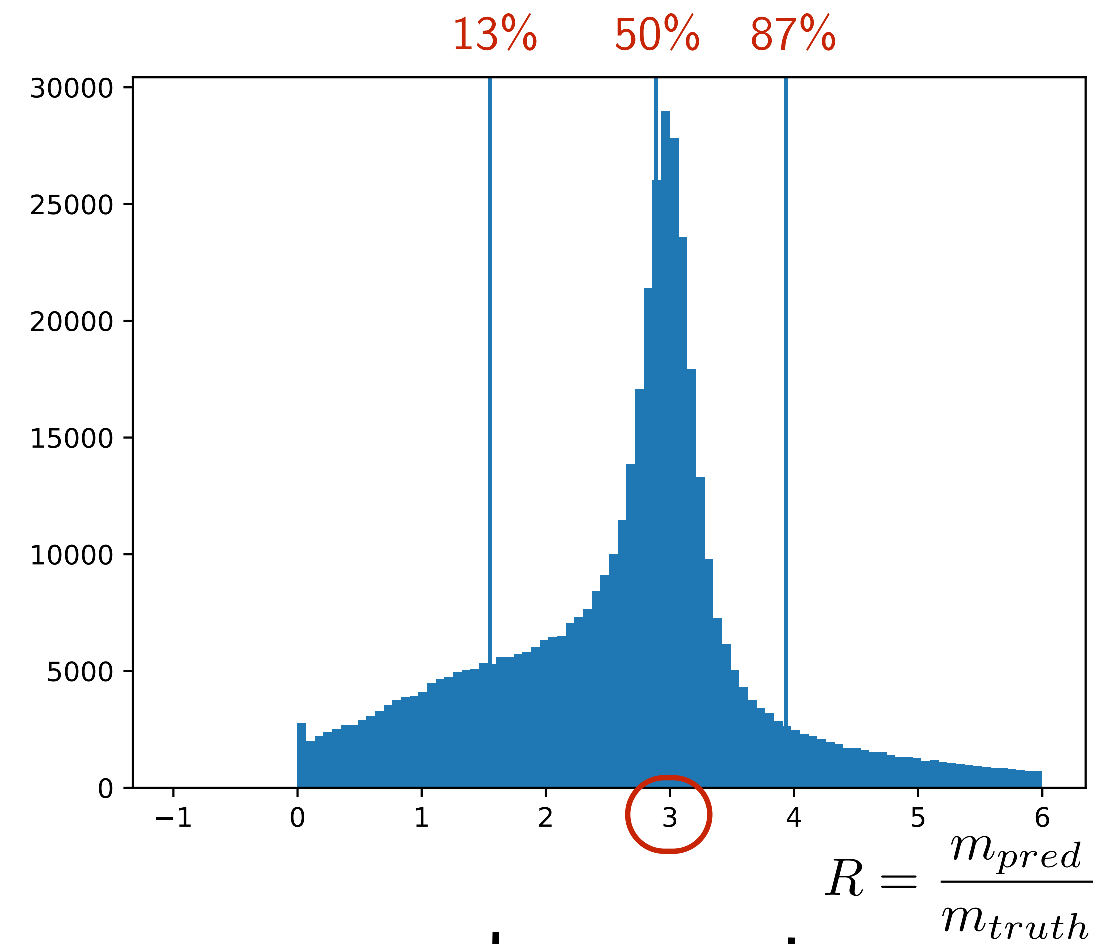
!! Insensitive to tails, i.e. 32% of all jets, take into account in loss function



Loss function

- Using the challenge metric itself as loss does not yield strong performance
  - !! Back-prop. through percentiles ...

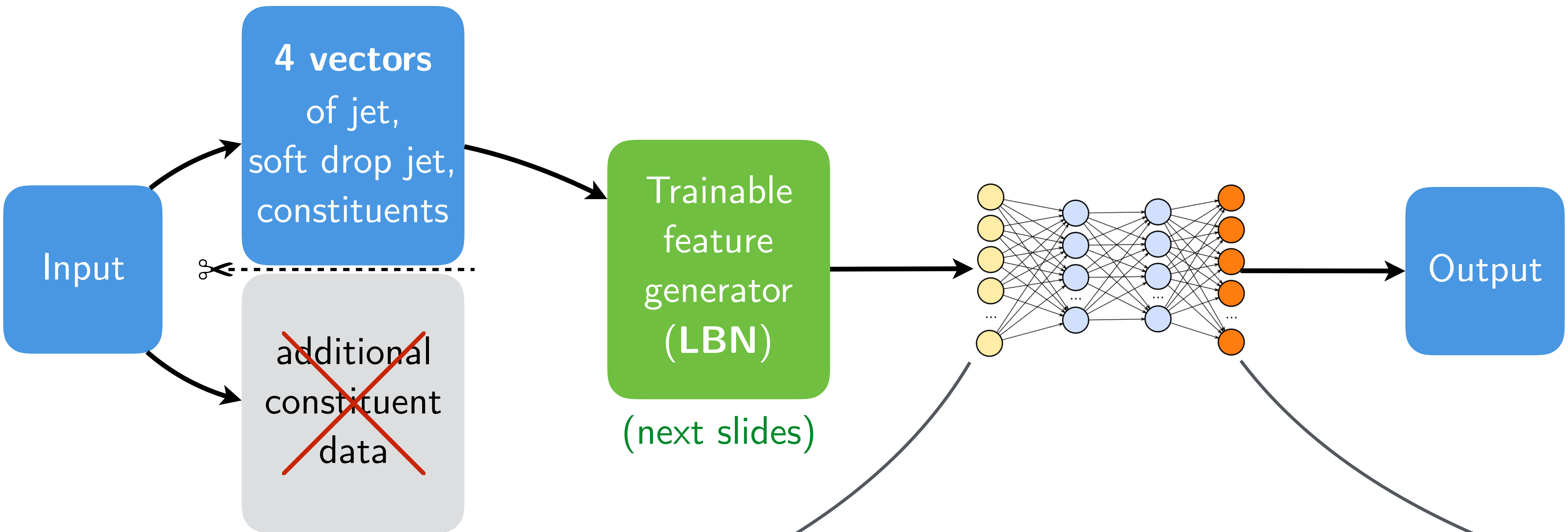
- Custom loss (per batch):
  - Select predictions in central **74%** range of prediction / truth
    - ▹ Increased range w.r.t. metric to be robust against boundary effects
  - Compute **MSE**, scaled to arbitrary value that is subject to tuning
    - ▹ t = 3 found to result in best training (interferes with weight init., batch norm., etc)



$$\mathrm{MSE}_t = \frac{1}{n_{\mathrm{batch}} \cdot t} \sqrt{\sum_i (t - R_i)^2}$$

→ Final resolution: 41.3%

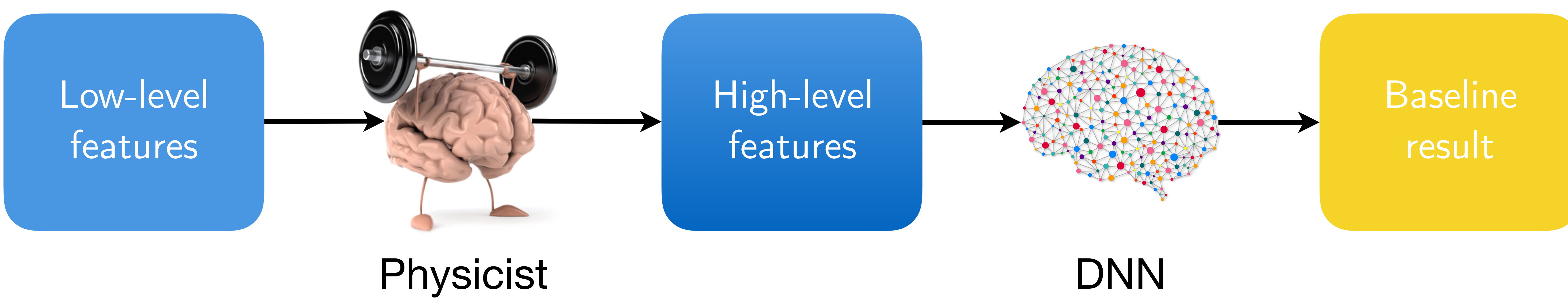
Network architecture



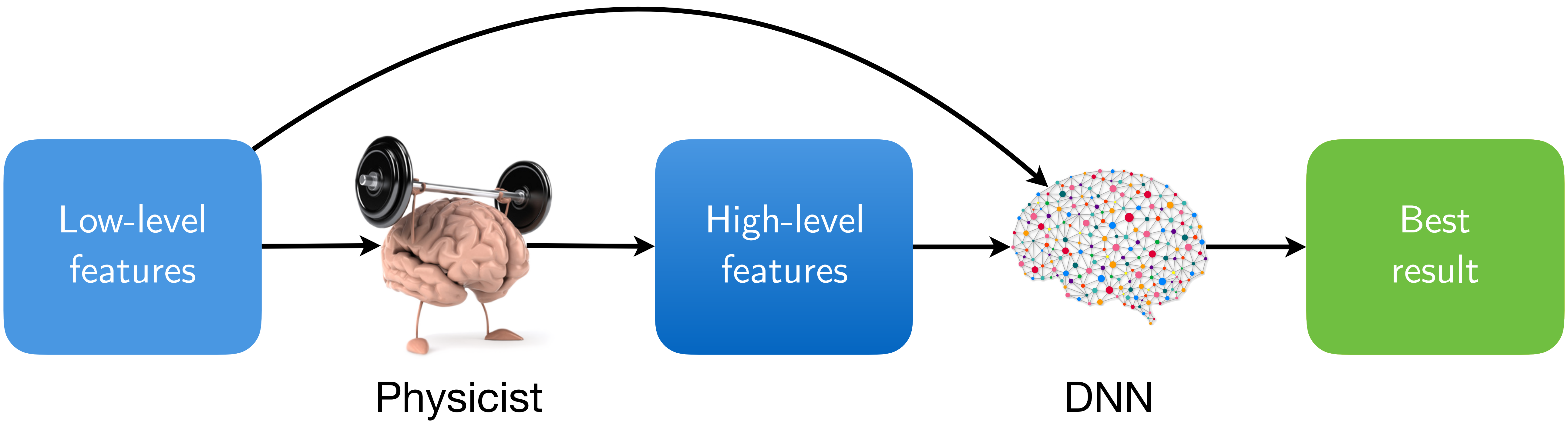
- Fully connected (FCN) architecture
- 5 hidden layers, 2048→1024→512→256→128
- ELU activation
- ADAM, r = 2.0e-4 @ batch size 4096
- L2 regularization, λ = 1.5e-5
- Batch normalization

→ Final resolution: 41.3%

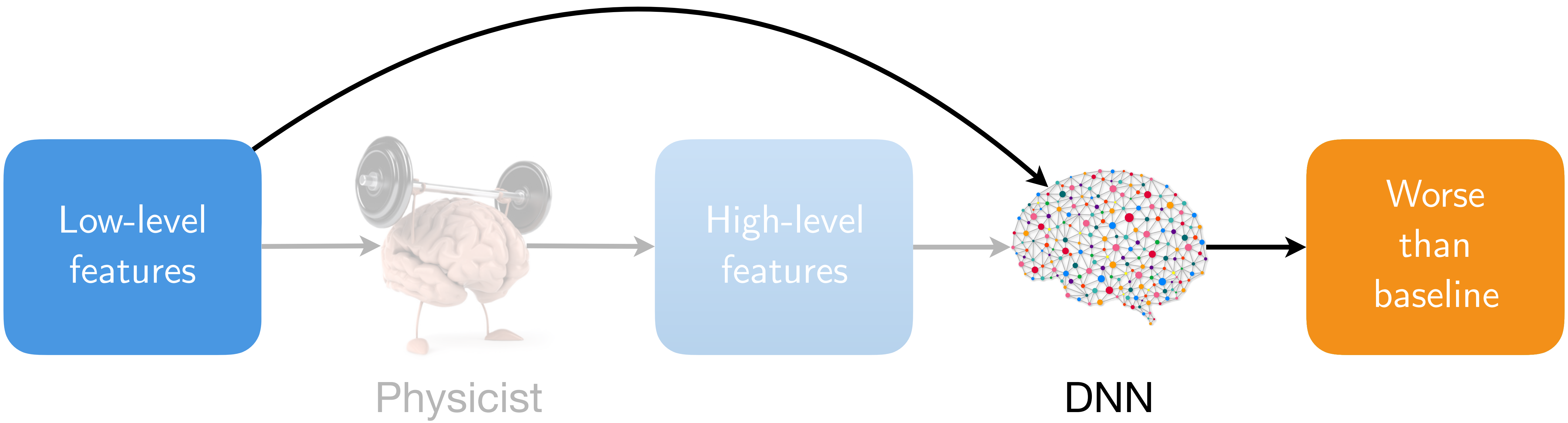
Feature engineering in HEP



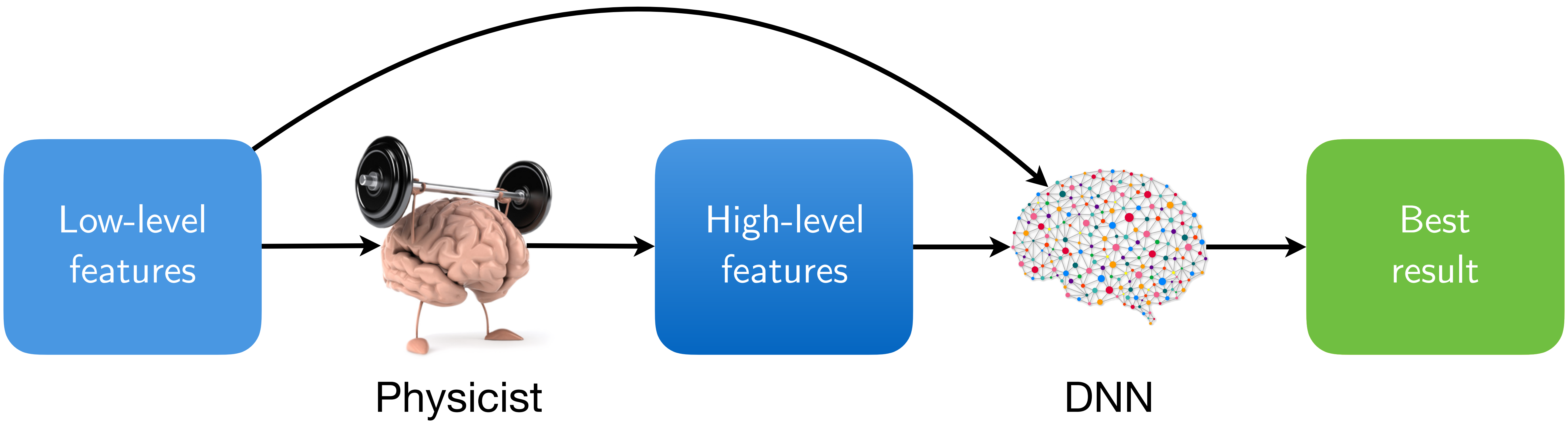
Feature engineering in HEP



Feature engineering in HEP

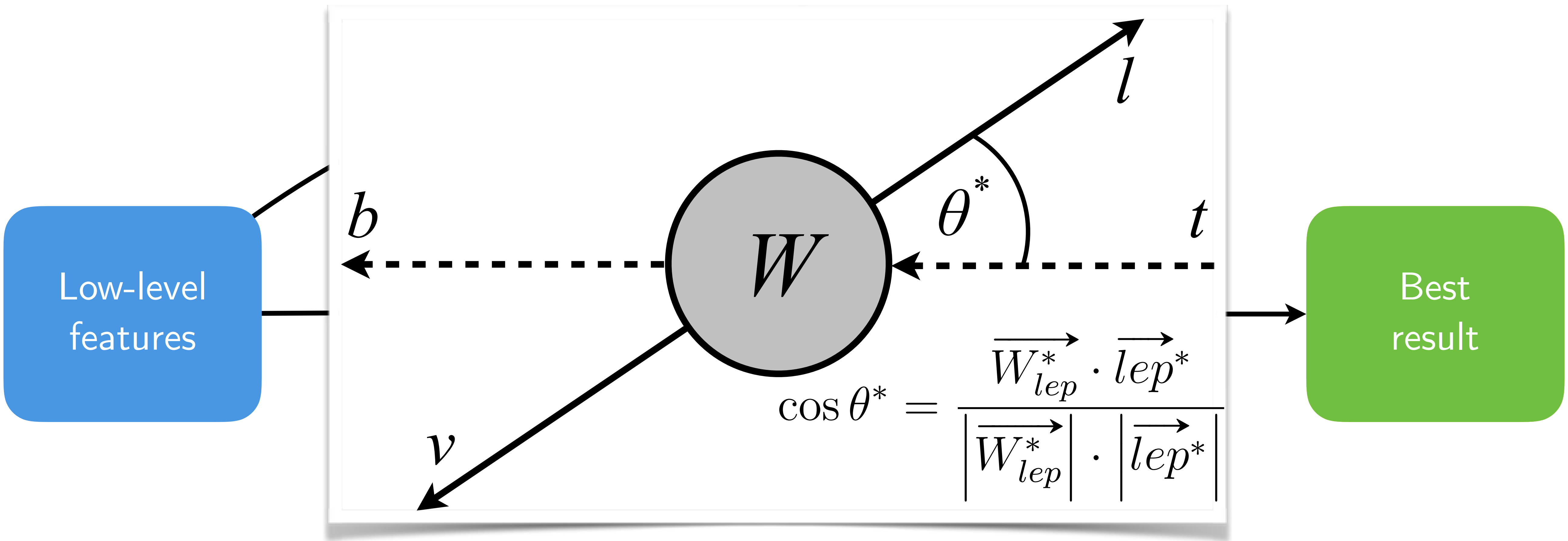


Feature engineering in HEP



- Problem statement

    1. Physicists' crafted high-level features might not exploit all available information
    2. In practice, it is hard for "standard" DNNs to learn representations of complex features



- Problem statement

1. Physicists' crafted high-level features might not exploit all available information
2. In practice, it is hard for "standard" DNNs to learn representations of complex features

- Aim: “”Encode first-principles of domain (physics) into network structure”
- Similar situation to FCNs → CNNs:
  - Images contain information in translation invariant adjacency of pixels

  → Exploit by changing the network structure!
- Already successful applications out there (selection):

arXiv 1707.08966

Deep-learned Top Tagging with a Lorentz Layer

Anja Butter[1], Gregor Kasieczka[2], Tilman Plehn[1], and Michael Russell[1,3]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Institute for Particle Physics, ETH Zürich, Switzerland
3 School of Physics and Astronomy, University of Glasgow, Scotland
plehn@uni-heidelberg.de

We introduce a new and highly efficient tagger for hadronically decaying top quarks, based on a deep neural network working with Lorentz vectors and the Minkowski metric. With its novel machine learning setup and architecture it allows us to identify boosted top quarks not only from calorimeter towers, but also including tracking information. We show how the performance of our tagger compares with QCD-inspired and image-recognition approaches and find that it significantly increases the performance for strongly boosted top quarks.

arXiv 1702.00748

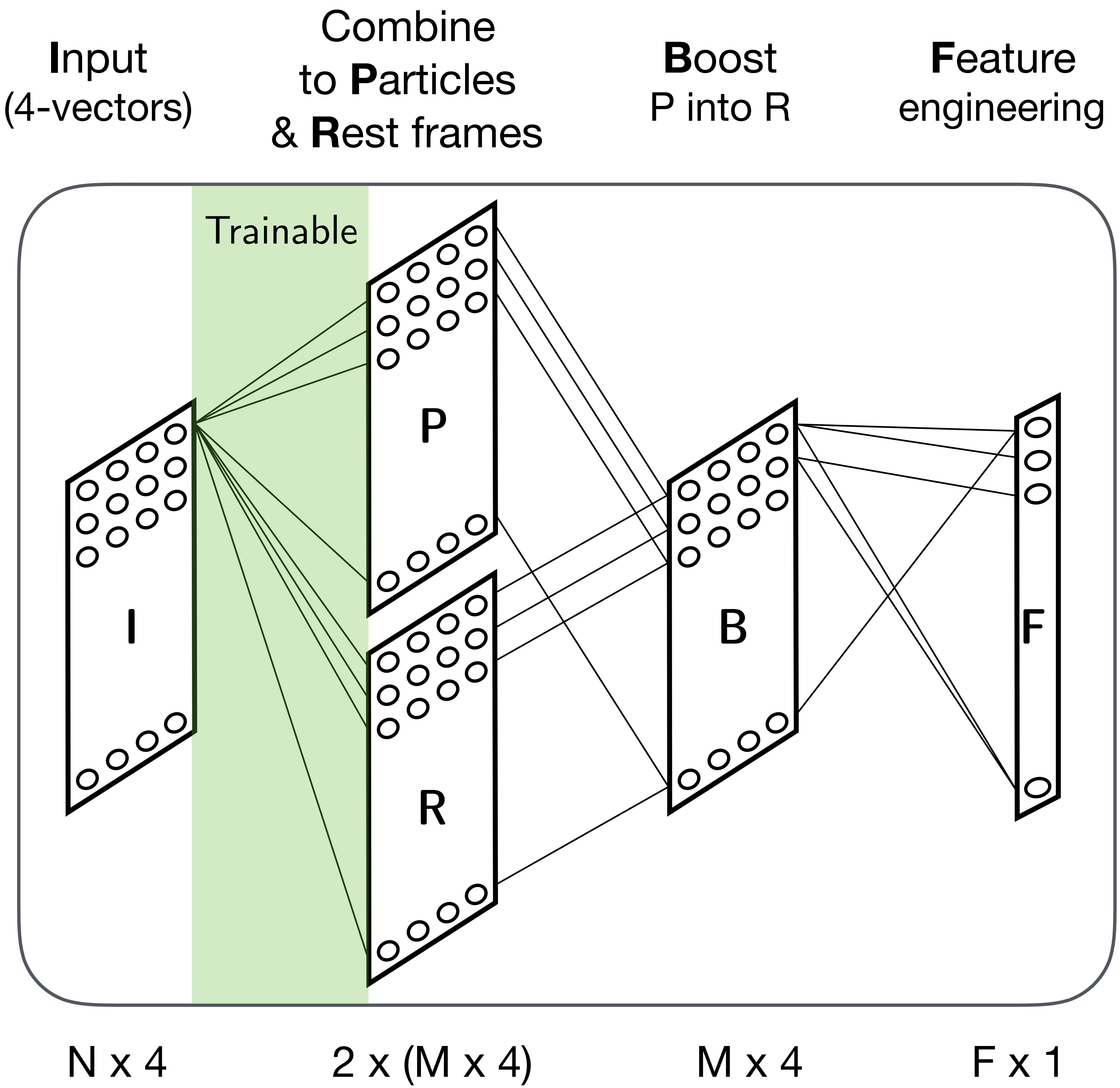
QCD-Aware Recursive Neural Networks for Jet Physics

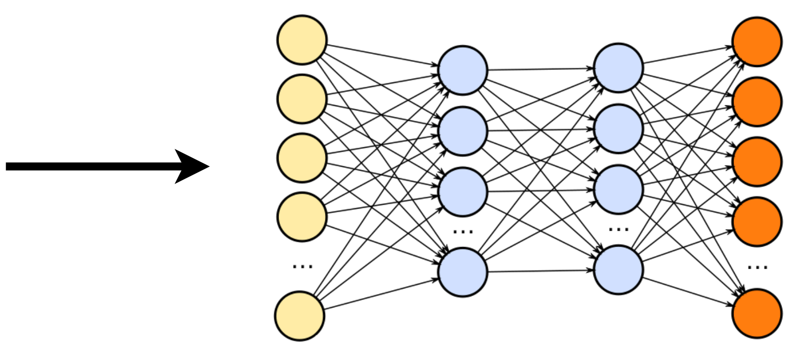
Gilles Louppe,[1] **Kyunghyun Cho,**[1] **Cyril Becot,**[1] and **Kyle Cranmer**[1]
[1] *New York University*

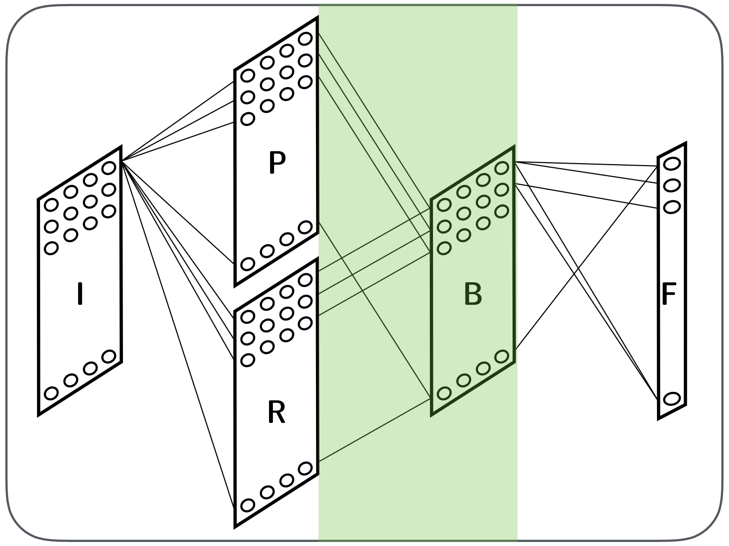
Recent progress in applying machine learning for jet physics has been built upon an analogy between calorimeters and images. In this work, we present a novel class of recursive neural networks built instead upon an analogy between QCD and natural languages. In the analogy, four-momenta are like words and the clustering history of sequential recombination jet algorithms is like the parsing of a sentence. Our approach works directly with the four-momenta of a variable-length set of particles, and the jet-based tree structure varies on an event-by-event basis. Our experiments highlight the flexibility of our method for building task-specific jet embeddings and show that recursive architectures are significantly more accurate and data efficient than previous image-based networks. We extend the analogy from individual jets (sentences) to full events (paragraphs), and show for the first time an event-level classifier operating on all the stable particles produced in an LHC event.

- Event classification by learning a rest frame with a “Lorentz Layer” (J.-R. Vlimant, link, link)

Paper public soon







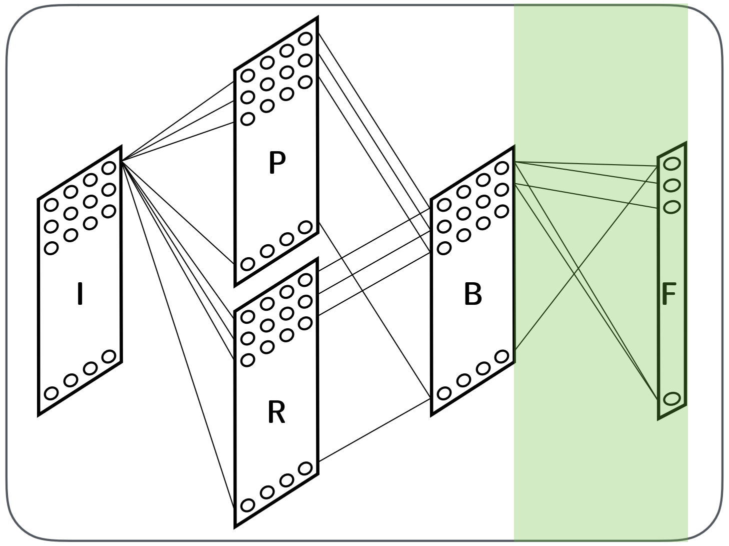
- Lorentz transformation $B_m = \Lambda(R_m) \cdot P_m$ with boost matrix

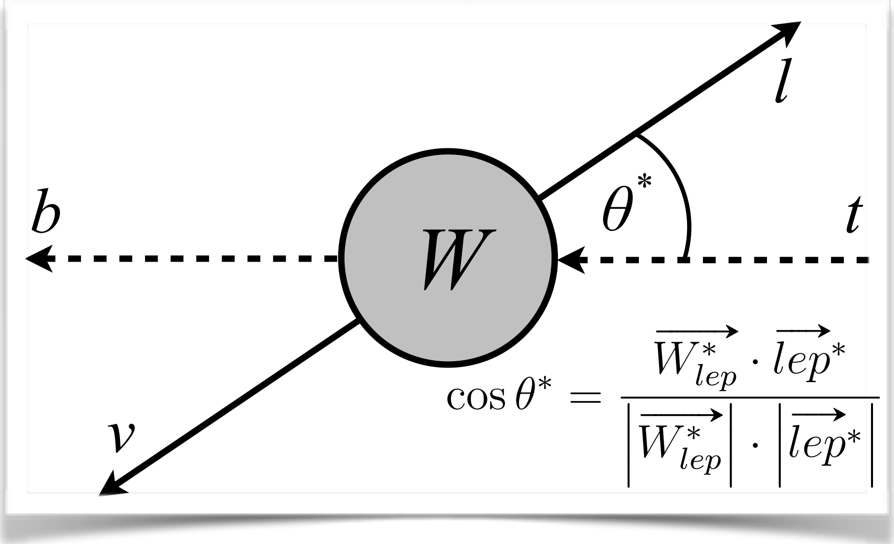
$$\Lambda = \begin{bmatrix} \gamma & -\gamma\beta n_x & -\gamma\beta n_y & -\gamma\beta n_z \\ -\gamma\beta n_x & 1 + (\gamma - 1)n_x^2 & (\gamma - 1)n_x n_y & (\gamma - 1)n_x n_z \\ -\gamma\beta n_y & (\gamma - 1)n_y n_x & 1 + (\gamma - 1)n_y^2 & (\gamma - 1)n_y n_z \\ -\gamma\beta n_z & (\gamma - 1)n_z n_x & (\gamma - 1)n_z n_y & 1 + (\gamma - 1)n_z^2 \end{bmatrix}$$

with $\vec{n} = \vec{\beta}/\beta$

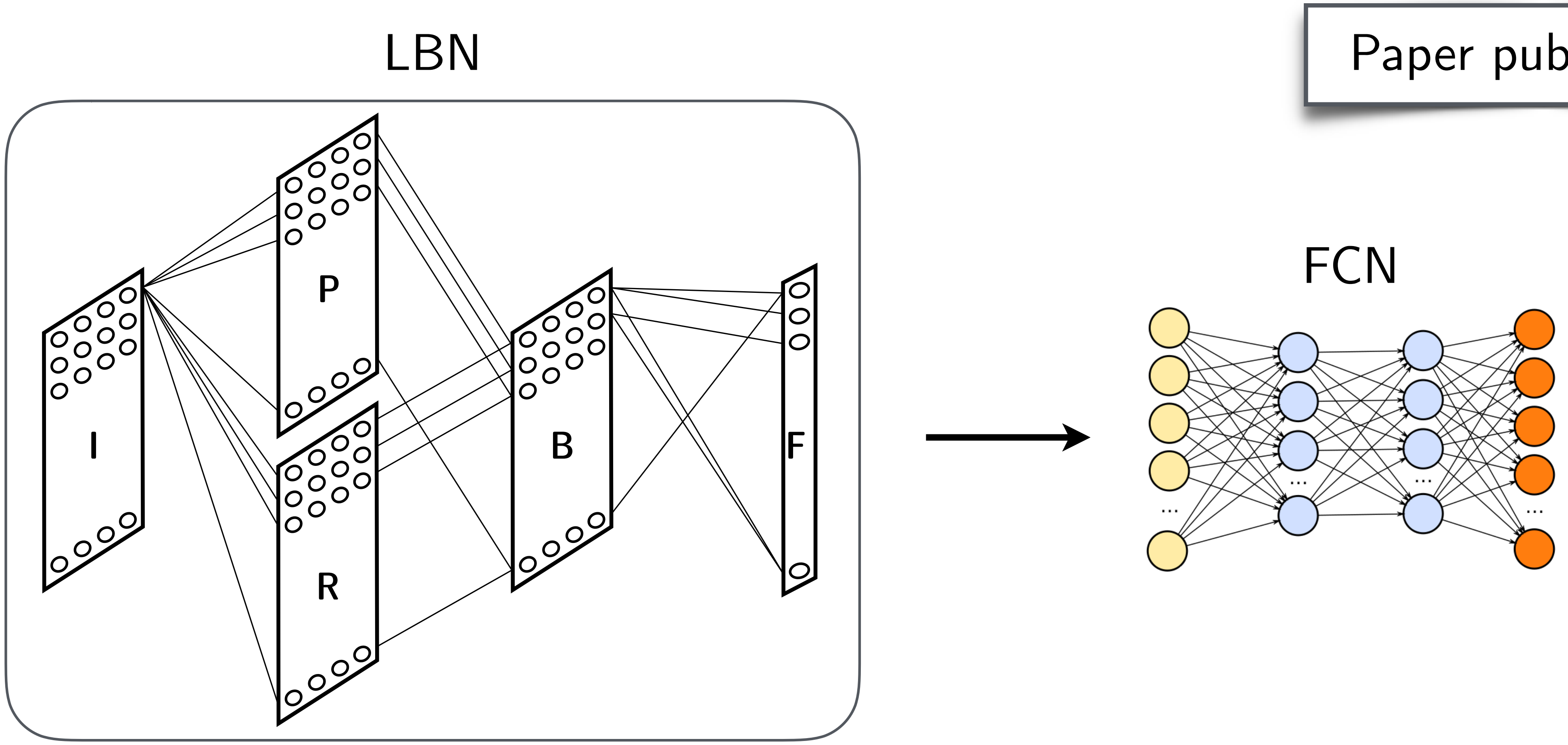
- Vectorized formulation to run efficiently on GPUs
  - 4D tensor (batch x particle x 4 x 4)

- Project features from M boosted 4-vectors:
  - Features per vector:
    - E, pt, eta, phi, mass
  - Pairwise features:
    - cos(ϕ) between vectors
  - More features possible, but not necessarily required
  - We used M = 50 → 1475 features in total
- Input feature scaling / normalization not applicable
  → Batch normalization applied after feature layer





Paper public soon



1. LBN is **not a black box**
   - ▹ Extract which (combined) particles and features are important from combinations
2. FCN behind LBN can be **rather shallow**
   - ▹ Feature representation moved to LBN
   - ▹ FCN can focus on transformation to output

Top tagging reference dataset

- Developed along arXiv:1707.08966 (A. Butter, G. Kasieczka, T. Plehn, M. Russell)
- Data:
  - 1.2M + 400k + 400k jets (top + QCD)
  - Only 4-vectors of up to 160 constituents
- Documentation:

Idea

Provide a simple set of training/testing MC simulation for the evaluation of top tagging architectures.

This is work in progress. Please let us know about any issues you encounter and share the performance you achieve on the test sample.

[...]

Collection of reference results

Approach	AUC	Acc.	1/eB (@ eS=0.3)	Training Time	Contact	Comments
LoLa	0.979	0.928			GK / SImon Leiss	Preliminary number, based on LoLa
LBN	0.979	0.928	60.6		Marcel Rieger	Preliminary number

HEP community needs method comparisons!

- IML challenges & public reference datasets:
  - ▹ Opportunities to **compare methods** and to open fruitful discussions

- *"Which separating variable could increase my network performance?"*

  should become

  "How can I design my network to (even better) work with raw features?"
  - ▹ Encode physics knowledge right into network **rather than** into variables

- Lorentz Boost Network possible candidate for many use cases