🌲 DeepTreeGAN 🌲

Fast Generation of High Dimensional Point Clouds

Moritz Scham[1,2,3], Dirk Krücker[1], Benno Käch[1,4], Simon Schnake[1,3], Kerstin Borras[1,3]

Generative models for particle showers

Motivation: Computing Challenge

- Particle physics: simulate detector response and physical processes to test theories
  ⟶ Immense computational effort
- LHC High-Luminosity phase ⟶More particles to simulate
- Future CMS high granularity calorimeter (HGCAL): more then 6M channels
  ⟶ Time-consuming simulations
  ⟶ Projected compute budget insufficient

⇒ **Save CPU time by using a Neural Network to simulate HGCAL data**

ML Challenges

Future high granularity calorimeters:

- High number of Channels
- Irregular Geometry
- Sparse data

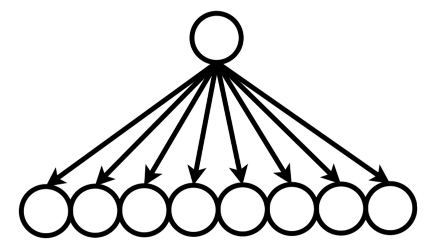
⇒No ML model powerful enough yet

⇒**Point Clouds** (PCs)

Target PC size: 2k Hits × 5 [E,t,x,y,z]

Core problem: How do we upsample PCs?

Naive approach:
Latent Vector ⟶ FFN ⟶ PCs

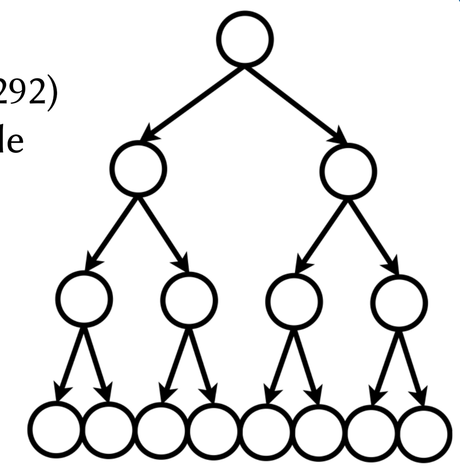


Number of parameters explodes
⇒ Not trainable

DeepTreeGAN:

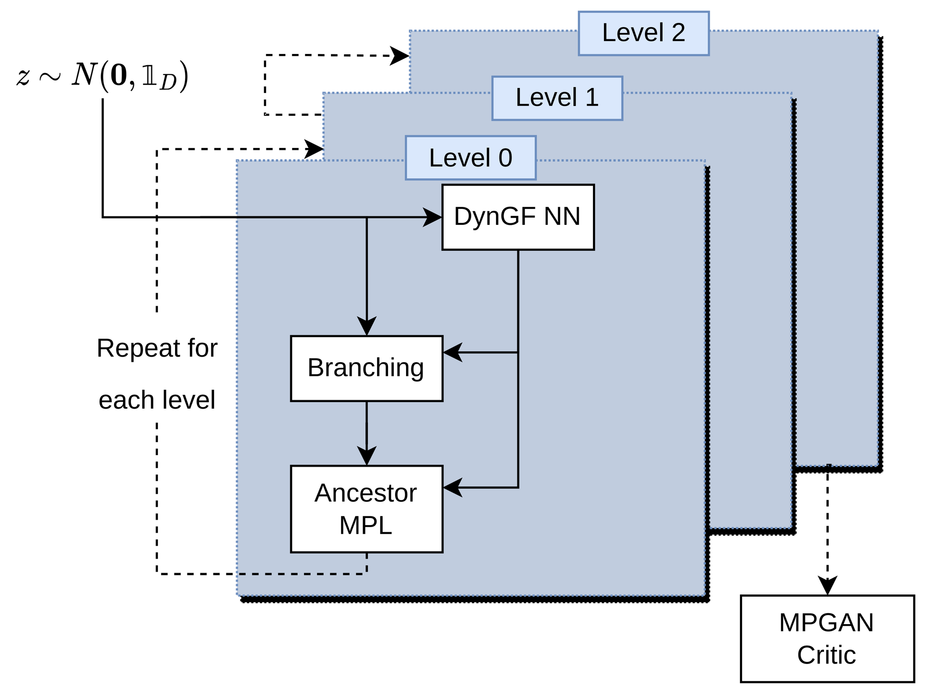
- Inspired by TreeGAN (arXiv:1905.06292)
- FFN projects each particle to multiple
- Repeat to grow a tree
- After k projections: $\prod_i^k k_i$ particles

⟹ Small output space for each FFN
⟹ Small number of parameters
⟹ Sparse representation, no padding



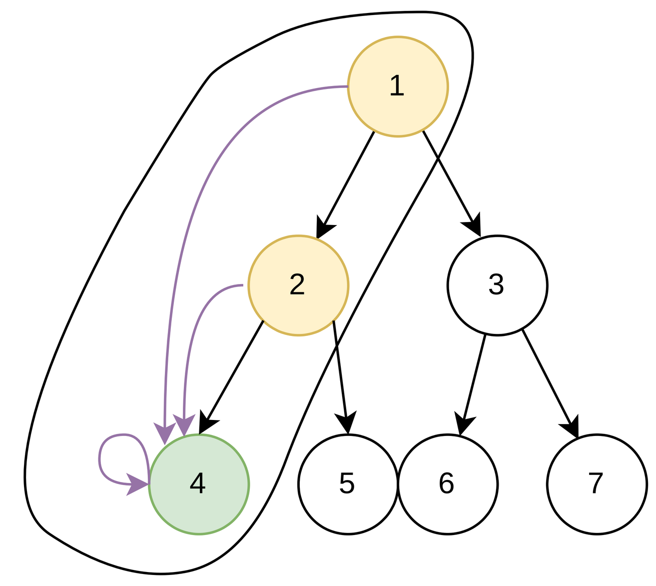
The Generator

Model Overview



- Start with random vector z
- Repeat for each level:
  1) **DynGF**: Encode the global state of the leaves (FFN ⟶ Sum ⟶ FFN)
  2) **Branching**: Split each of the leaves
  3) **Ancestor MPL**: Pass information down from ancestors to their children
- Last level: Take the first n points, to match the simulation

Ancestor Message Passing Layer



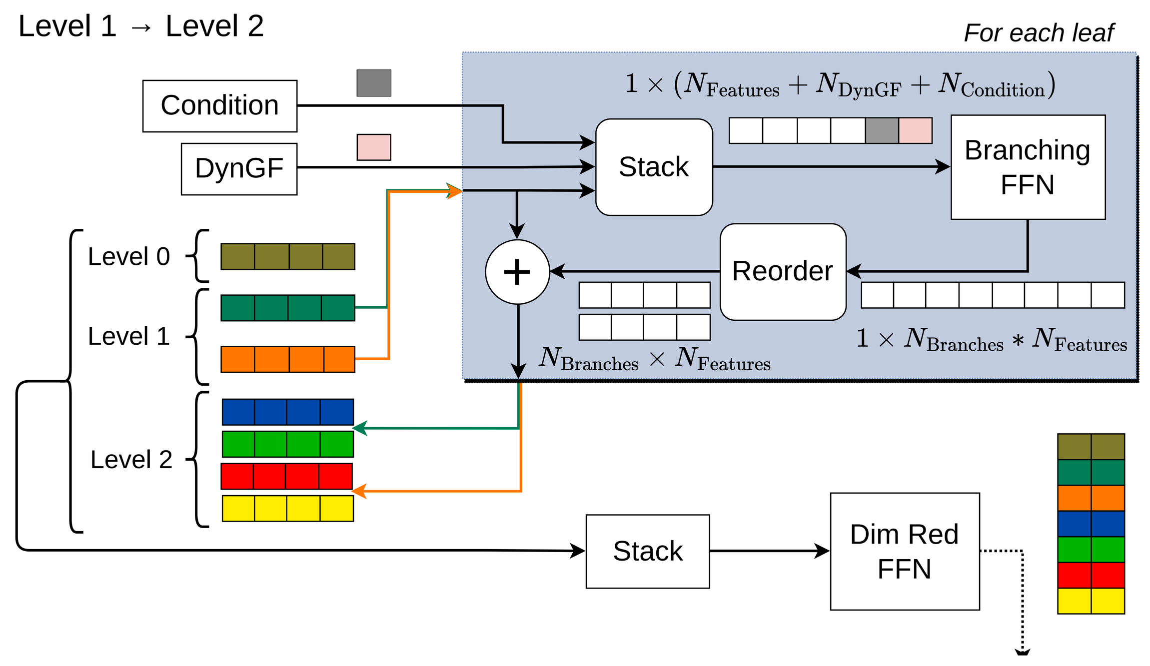
Selected MPL: GINConv (arXiv: 1810.00826)

1) Message: $\mathrm{Msg}_{j \to i} = x_j$

2) Aggregate: $\mathbf{Aggr}_i = \sum_{i \in N(j)} \mathrm{Msg}_{j \to i}$

3) Update: $x_i \leftarrow \mathrm{NN}\left((1+\epsilon)\, x_i + \mathbf{Aggr}_i\right) + x_i$

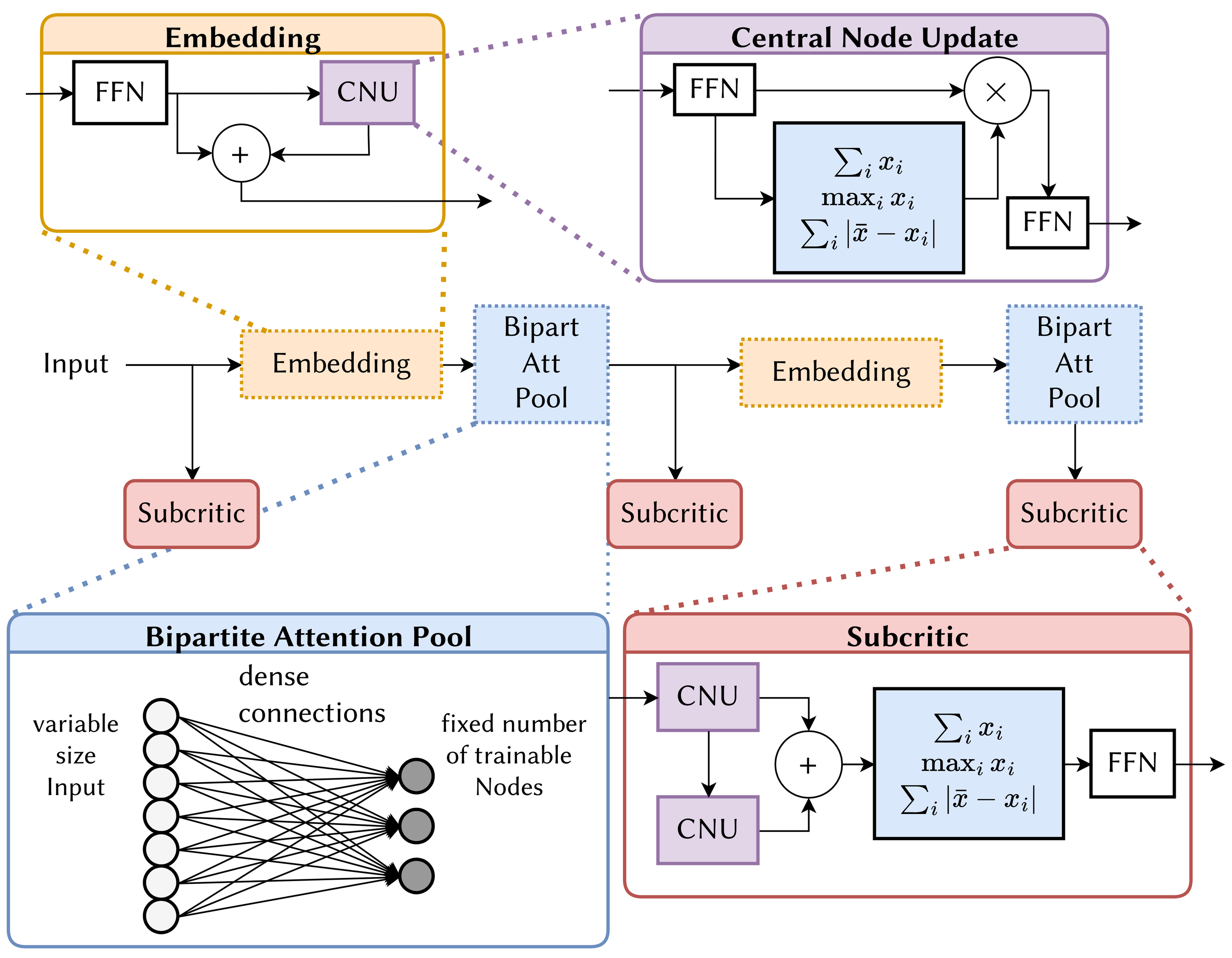
Branching



Branch from Level 1 to Level 2

- For each leaf separately:
  1) **DynGF**: Stack leaf with condition and global feature vector
  2) Branching FFN: Project to target dimension
  3) Split into the number of branches
  4) Add the parent ontop
- Stack up all levels of the tree
- Reduce the number of features with a FFN

Critic



Central Node Update (CNU)

- Pass points though feed-forward network (FFN)
- Aggregate points (sum+max+width)
- Pass aggregate & points through 2nd FFN

Embedding

- Map points in higher dim. space with FFN
- Pass though CNU (+residual connection)

Subcritic

- Pass though 2 CNU (+residual connection)
- Aggregate points (sum+max+width)
- Map aggregate to dim 1 with FFN
- Hinge Loss

Bipartite Attention Pool

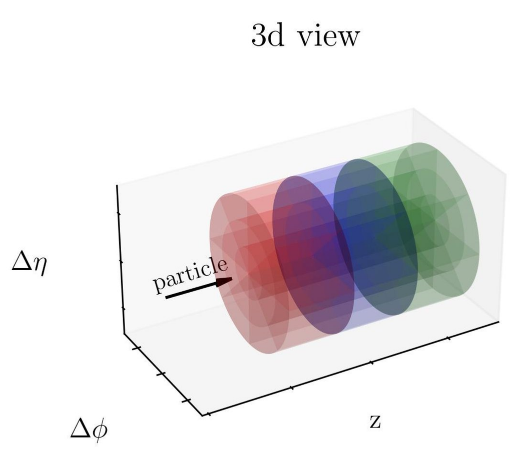
1) Construct dense, bipartite graph from input to trainable nodes
2) Apply Gatv2Conv Message Passing Layer

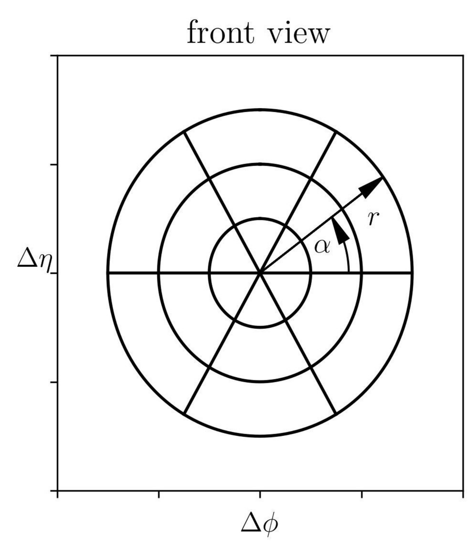
- differentiable
- permutation invariant
- variable sized input
- fixed size output

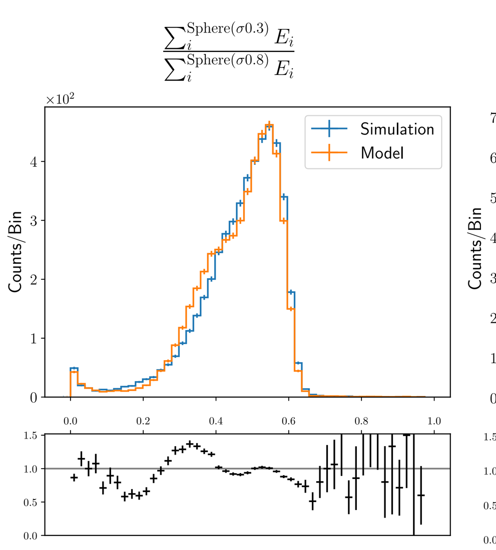
Benchmark II: CaloChallange Dataset 2

Public dataset: DOI 10.5281/ `calochallenge.github.io/homepage/`

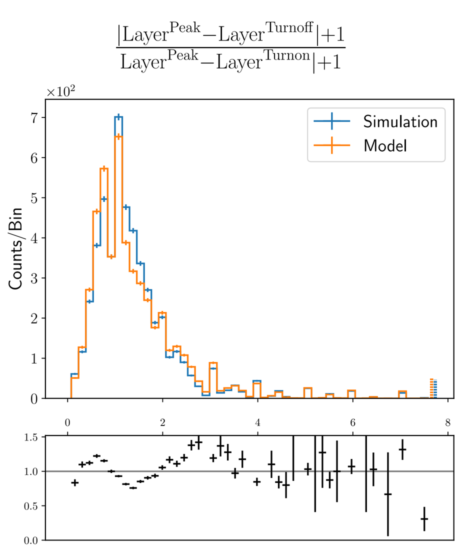
- 100k GEANT4-simulated electrons showers for training/testing
- Energies with log-uniform distribution [1 GeV,1 TeV]
- Concentric cylinder detector geometry
- 45 layers(z) x 9 radial segments (r) x 16 angular segments (α) = 6480 voxels



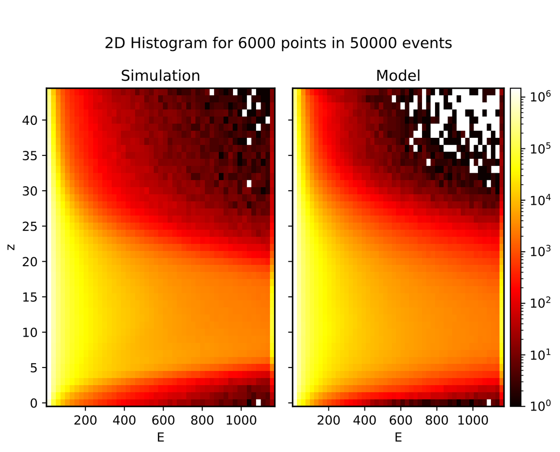




Check on the density: Energy contained in Sphere around center with radius 0.3*std vs 0.8*std



Check on showershape: Distance turnon (½ maximum) ⟶ peak vs turnoff ⟶ peak



Distribution of the hits in energy (E)[GeV] and layer (z)

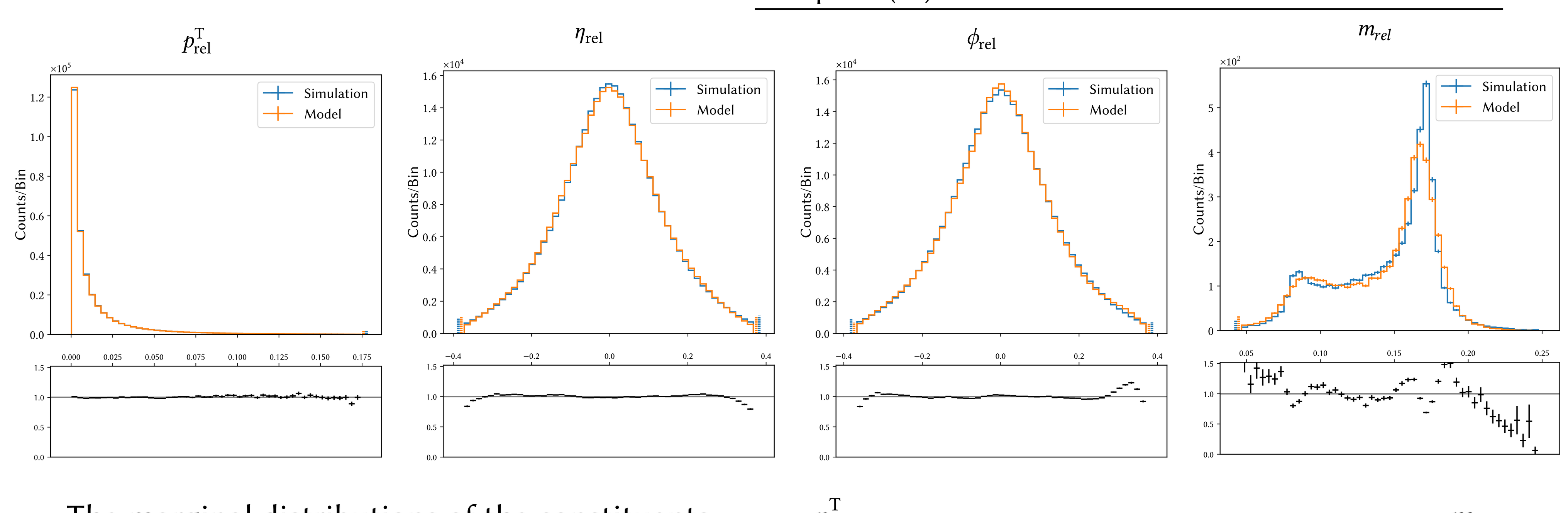
Benchmark I: JetNet 150 Top-Quarks

JetNet (arXiv:2106.11535) dataset:

- Hadronized jets, anti-kT (R=0.8)
- 170k gluon/light quark/top jets
- Leading 150 constituents by p^T
- Metrics for benchmarking

Metrics compared to other models, lower is better, best in bold
arxiv: 2305.15254 (MDAM) 2301:08128 (EPiC-GAN)

Model	$W_1^M \times 10^3$	$W_1^P \times 10^3$	$W_1^{EFP} \times 10^5$	FPD×10^4
In Sample	0.42±0.09	0.12±0.04	1.22±0.32	1.2±0.6
EPiC-GAN	0.69±0.08	0.65±0.03	2.67±0.39	22±1
MDMA	**0.57±0.09**	0.10±0.02	**2.12±0.64**	5.3±0.9
DeepTree(v2)	1.49±0.04	0.13±0.02	5.01±0.08	**3.4±0.7**



The marginal distributions of the constituents and the mass of the top-initiated jets

$$p^T_{\text{rel}} = \frac{p^T_{\text{particle}}}{p^T_{\text{jet}}},\ \eta_{\text{rel}} = \eta_{\text{particle}} - \eta_{\text{jet}},\ \phi_{\text{rel}} = \phi_{\text{particle}} - \phi_{\text{jet}},\ m_{\text{rel}} = \frac{m_{\text{jet}}}{p^T_{\text{jet}}}$$

⇒ **Almost identical architecture for jets and calo. showers**

⇒ **New, differentiable up/downscaling methods for point clouds**

⇒ **Promising model to scale to even larger point clouds (e.g. HGCAL)**

(1) DESY, Hamburg; (2) FZJ and Jülich Supercomputer Center; (3) RWTH University of Aachen; (4) Hamburg University

MS/BK funded through Helmholtz AI - grant number: ZT-I-PF-5-3/ZT-I-PF-5-064

HELMHOLTZ AI ARTIFICIAL INTELLIGENCE COOPERATION UNIT · DESY. · JÜLICH Forschungszentrum · RWTH AACHEN UNIVERSITY · UHH