



End-to-end multi-particle reconstruction in high occupancy imaging calorimeters with graph neural networks

Shah Rukh Qasim^{1,2,a} , Nadezda Chernyavskaya^{1,b} , Jan Kieseler^{1,c} , Kenneth Long^{4,d} ,
Oleksandr Viazlo^{3,e} , Maurizio Pierini^{1,f} , Raheel Nawaz^{5,g}

¹ Experimental Physics Department, CERN, Geneva, Switzerland

² Manchester Metropolitan University, Manchester, UK

³ Florida State University, Tallahassee, USA

⁴ Massachusetts Institute of Technology, Cambridge, USA

⁵ Staffordshire University, Staffordshire, UK

Received: 14 April 2022 / Accepted: 1 August 2022 / Published online: 29 August 2022
© The Author(s) 2022

Abstract We present an end-to-end reconstruction algorithm to build particle candidates from detector hits in next-generation granular calorimeters similar to that foreseen for the high-luminosity upgrade of the CMS detector. The algorithm exploits a distance-weighted graph neural network, trained with object condensation, a graph segmentation technique. Through a single-shot approach, the reconstruction task is paired with energy regression. We describe the reconstruction performance in terms of efficiency as well as in terms of energy resolution. In addition, we show the jet reconstruction performance of our method and discuss its inference computational cost. To our knowledge, this work is the first-ever example of single-shot calorimetric reconstruction of $\mathcal{O}(1000)$ particles in high-luminosity conditions with 200 pileup.

1 Introduction

The high-luminosity upgrade of the Large Hadron Collider (HL-LHC) will present unprecedented computing challenges [1]. Because the processing complexity of LHC collision scales with the number of hits and energy deposits in the detectors from interacting particles, the computing resource

needs will increase significantly as a function of the number of simultaneous proton-pair collisions at each particle beam crossing (pileup). In addition, particle detectors with irregular geometries, motivated by a need to combat the high radiation environment of high-pileup events, prohibit algorithms that view the detector as a simple grid and encourage more sophisticated approaches.

One of the most problematic computing tasks is the so-called local event reconstruction, i.e., the task of clustering detector hits (energy deposits left by particles on various detector sensors) into particle candidates. Similar to how semantic segmentation works in computing vision, particle reconstruction is more than a clustering task. The reconstruction of a clustered object implies associating it to one particle category (a classification task) and determining the particle energy and flight direction (a regression task). In the future, the enforcement of 5D reconstruction (3D position, energy, and particle time of arrival) both in hardware and software would imply an additional regression task for the time to be considered.

In traditional approaches, particle reconstruction follows a two-step strategy: first, clusters are built, and then classification and regression tasks are performed on these clusters. In some cases, the regression and classification steps employ machine learning algorithms. Boosted decision trees have been used most extensively for this purpose at the LHC. An example of this strategy is discussed in Ref. [2], in the case of the CMS electromagnetic calorimeter.

In LHC proton-proton collisions, two bunches of $\sim 10^{11}$ protons are brought to collision; to increase the probability of rare and interesting interactions (e.g., the production of a Higgs boson) to occur. Because of this, a single collision

^a e-mail: shah.rukh.qasim@cern.ch (corresponding author)

^b e-mail: nadezda.chernyavskaya@cern.ch

^c e-mail: jan.kieseler@cern.ch

^d e-mail: kenneth.long@cern.ch

^e e-mail: oleksandr.viazlo@cern.ch

^f e-mail: maurizio.pierini@cern.ch

^g e-mail: raheel.nawaz@staffs.ac.uk

event contains the particles resulting from more than one collision (the primary particles). These particles travel through the detector components and, when crossing a calorimeter, produce showers of other particles (secondary particles). During Run II, an average LHC collision event consisted of ~ 40 pileup collisions, resulting in $\mathcal{O}(1000)$ primary particles. At the HL-LHC, up to 200 pileup collisions per event are expected. Traditional event reconstruction algorithms are challenged by the large primary-particle multiplicity due to the fast scaling of their memory utilization and execution time with the cardinality of the problem. Limiting the increase of computing resource consumption at large pileup is a necessary step for the success of the HL-LHC physics program [1]. The use of modern machine learning techniques to perform particle reconstruction has been advocated as a possible solution to this problem [3].

In this paper, we present an end-to-end reconstruction algorithm, which takes as input a collection of detector hits in a highly granular calorimeter, similar to that under construction by the CMS collaboration [4], in view of the HL-LHC upgrade [5]. The algorithm returns identified particle candidates and their momenta. The algorithm consists of a distance-weighted graph neural network [6, 7], trained using object condensation [8]. At training time, the loss function is built by combining the clustering and regression tasks. In this respect, the presented algorithm is an example of a single-shot graph segmentation model that could find applications beyond the domain of particle physics.

We discuss the algorithm performance in terms of accuracy and computational costs, considering in particular how its performance scales with increasing pileup. We consider both single-particle response and jet reconstruction to evaluate the reconstruction algorithm both at particle level and with higher-level objects. To our knowledge, this study represents the first demonstration of end-to-end calorimeter reconstruction at high pileup exploiting neural networks.

This paper is organized as follows: Sect. 2 discusses the existing literature related to this work; Sect. 3 describes the detector geometry and the data generation workflow. Section 4 describes the dataset format and its preprocessing. Sections 5 and 6 describe the model architecture and the optimization metric, respectively; Sect. 7 discusses the inference clustering algorithm and how it improves with respect to the original version of object condensation and the computational costs; Sect. 8 shows the physics performance of the algorithm, respectively; conclusions are given in Sect. 9.

2 Related work

In recent years, GPU acceleration has been investigated as a means to speed up traditional particle-reconstruction algorithms by parallelization [9–15]. In view of these promis-

ing results, the LHC experimental collaborations invested financial resources to migrate their computing infrastructure towards CPU+GPU heterogeneous computing [16–18]. This transition offers the possibility to exploit neural networks for the same task while benefiting from the impressive technological development in Artificial Intelligence (AI) inference on GPU+CPU heterogeneous platforms. Leveraging this technology trend, we are investigating the possibility of using solutions entirely based on AI algorithms to accomplish calorimetric reconstruction tasks [6, 19, 20]. A similar effort has already been established for particle tracking [21–24].

Neural networks have also been used in classification and regression tasks on portions of calorimeters [25–27]. The use of neural networks for end-to-end reconstruction goes beyond calorimeter reconstruction or tracking. Computer vision techniques based on convolutional and graph neural networks have been used for event-topology classification directly from the energy map of detector hits [28–31]. Deep neural networks have been exploited as a tool to cluster high-level objects, e.g., particle-flow candidates [32] and jets [33, 34]. All these studies demonstrate that modern AI algorithms give significant accuracy and computational efficiency advantages, maximally benefiting from highly parallelizable hardware. For this reason, and given the popularity of these techniques outside the domain of particle physics, it is natural to assume that AI algorithms will be at the center of future technological developments and, consequently, will play a crucial role in the evolution of the computing model at future collider experiments.

3 Detector

For this study, we use a dedicated simulation of a highly granular calorimeter based on GEANT4 [35]. The detector geometry is a simplified version of the CMS high-granularity calorimeter (HGCal) [36], expected to be integrated into the CMS detector for HL-LHC.

To describe the detector, we use a right-handed Cartesian coordinate system with the z -axis oriented along the beam axis, the x -axis toward the center of the LHC, and the y -axis oriented upward, as shown in Fig. 1. The x and y axes define the transverse plane, while the z -axis identifies the longitudinal direction. The azimuth angle ϕ is computed with respect to the x axis. Its value is given in radians, in the $[-\pi, \pi]$ range. The polar angle θ is measured from the positive z -axis and is used to compute the pseudorapidity $\eta = -\log(\tan(\theta/2))$. The transverse momentum (p_T) is the projection of the particle momentum on the (x, y) plane. We use natural units such that $c = \hbar = 1$, and we express energy in units of electronvolt (eV) and its prefix multipliers.



Fig. 1 Schematic representation of the right-handed Cartesian coordinate system adopted to describe the detector

Like the CMS HGICAL, our detector is positioned in the endcap region of a cylinder-shaped 4π detector. The endcap calorimeter covers the $1.5 < |\eta| < 3.0$ pseudorapidity range. Its longitudinal and transverse cross-section views are shown in Fig. 2. The detector design is based on a sampling geometry structured in three blocks:

1. $14\times$ Electromagnetic layers
2. $12\times$ Hadronic layers with thin absorbers
3. $16\times$ Hadronic layers with thick absorbers

A schematic representation of the longitudinal sampling structure is shown in Fig. 3, where the layer thickness in z and the materials of which each layer is composed are outlined. The electromagnetic section of the detector corresponds to 17 radiation and 1.3 nuclear interaction lengths, while the hadronic section corresponds to about 10 nuclear interaction lengths.

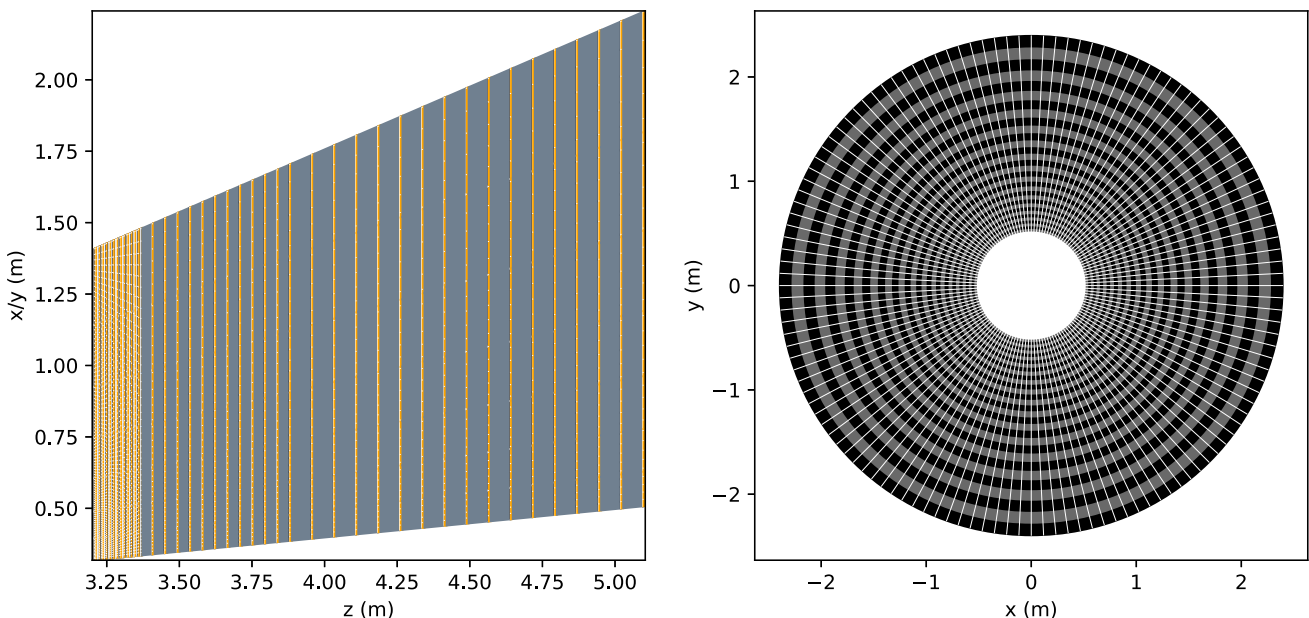


Fig. 2 Left: schematic representation of the detector longitudinal sampling structure. Right: transverse view of the last active layer. Different colors represent different materials: copper (orange), stainless steel and lead (gray), air (white) and active sensors made of silicon (black)

Since each electromagnetic layer consists of two silicon-sensor planes, there are, in total, 56 silicon layers. When projected on the (η, ϕ) plane, the first silicon layer consists of square-shaped 0.02×0.02 wide sensors. The sensor size linearly increases with the layer depth, reaching 0.07×0.07 for the last layer. In total, each endcap consists of 778,712 sensors. Due to cost-related considerations, the CMS HGICAL geometry is characterized by hexagon-shaped sensors. While being based on simpler square-shaped sensors, our detector has a comparable granularity and it is complex enough to faithfully represent the reconstruction challenges (e.g., in terms of image sparsity, average occupancy, and image resolution) of the HGICAL while making the GEANT4 simulation of the detector more tractable.

4 Simulation and event generation

Data generation starts with a GEANT4 simulation of individual particles or individual proton collisions that are later combined to form more complex events, e.g., events with pileup. In each simulation, one or more primary particles are produced at the interaction point $(0, 0, 0)$. These particles travel in empty space, since no simulation of an inner tracking detector or magnetic field is part of our simplified setup. Some of the primary particles reach the calorimeter and interact with its material, starting a showering process that creates secondary particles $P_{\text{secondary}}$. These secondary particles leave energy deposits on the sensors of the silicon

layer as hits. Each hit on a sensor is associated to a secondary particle. Four types of interactions are simulated:

1. Type A: Single-particle simulations for training. The particles are randomly chosen as e^- , γ , π^+ , π^0 or τ , with momentum coordinates uniformly distributed in $E \in [0.1, 200]$ GeV, $\eta \in [1.4, 3.1]$, and $\phi \in [0, 2\pi]$. $3.1 \cdot 10^5$ simulations are generated that are all used for training.
2. Type B: Stable single-particle simulations generated 1 mm away from the detector, as if they are coming from the interaction point in a straight line for testing performance of the models. The particles are randomly chosen as e^- , γ , or π^+ , with momentum coordinates uniformly distributed in $E \in [0.1, 200]$ GeV, $\eta \in [1.6, 2.9]$, and $\phi \in [0, 2\pi]$. 80,000 simulations are generated which are all used for testing.
3. Minimum bias: Synthetic minimum bias proton-proton interactions, generated at a center-of-mass energy of $\sqrt{s} = 13$ TeV (as during Run II of the LHC), using PYTHIA8 [37]. $3.1 \cdot 10^5$ and $2 \cdot 10^5$ simulations are generated for training and testing, respectively. These simulations are used to generate pileup for both training and testing.
4. $t\bar{t}$: Synthetic $q\bar{q} \rightarrow t\bar{t}$ events generated at $\sqrt{s} = 13$ TeV using PYTHIA8. This sample is used to study the jet reconstruction accuracy. The $q\bar{q}$ production mechanism is selected in order to maximize the fraction of events that produce primary particles in the endcap region. 40,000 simulations are generated, which are all used for testing.

As a pre-processing step, the raw energy deposit on a sensor returned by GEANT4 is calibrated by rescaling the hit energy according to a multiplicative factor c_i , defined as:

$$c_i = 1 + \frac{w_a(i)}{w_s(i)}, \tag{1}$$

where $w_s(i)$ is the width in mm of the relevant sensor and $w_a(i)$ is that of the absorber layer in front of it. Starting from c_i , we then apply a global calibration factor \hat{g} , which is computed by minimizing the squared difference between the energy of the incoming particle and the deposited hit energy, using a single-particle calibration dataset D_{calib} :

$$\hat{g} = \underset{g}{\operatorname{argmin}} \sum_{S \in D_{\text{calib}}} (E(p_S) - \sum_{i \in S} g c_i d_{\text{raw}}(i, p_S))^2, \tag{2}$$

where p_S labels the unique primary particle in the event S , $E(p_S)$ – its energy, i – the sensor in the ensemble \mathbb{S} of all sensors, and $d_{\text{raw}}(i, p_S)$ – the raw energy deposit on i by p_S . The D_{calib} sample, a subsample of the training dataset, is defined by requiring all particles to be photons with $1.8 < \eta(p_S) < 2.8$, and consists of the 240 single-photon events with energies between 4 and 200 GeV. Fig. 4 shows the calibration performance.

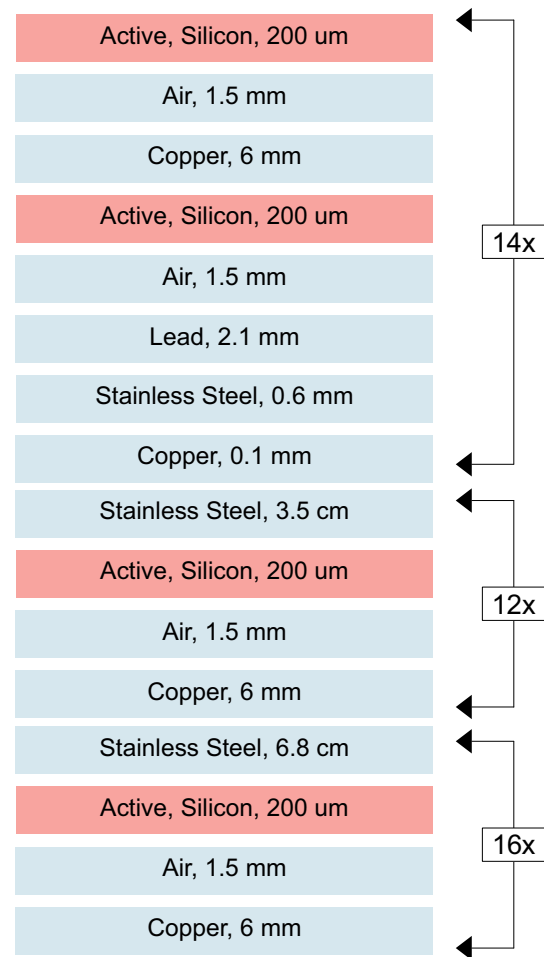


Fig. 3 Layers of the detector. Top 8 blocks in the diagram represent the electromagnetic section of the calorimeter, and the rest, the hadronic section

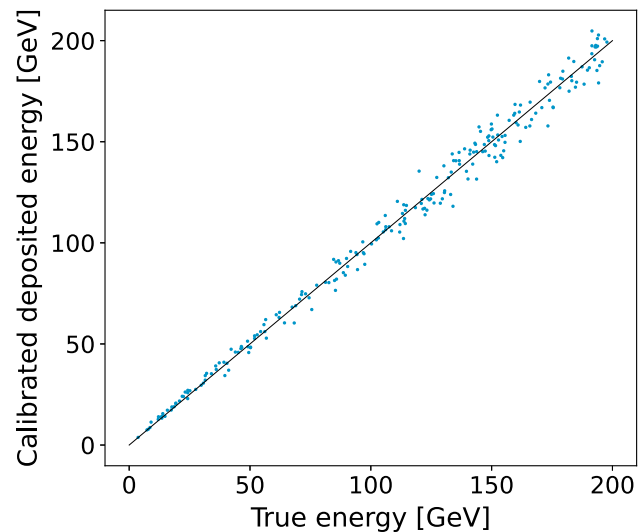


Fig. 4 A scatter plot showing calibrated deposited energy versus true energy of photons shot between $1.8 < \eta < 2.8$

Several simulations (S) are combined to form a full event as outlined in the algorithm Procedure 1. The ultimate task on this full event is to reconstruct every particle that enters the detector. For this purpose, the ground-truth clusters (T) that the network is trained to learn are built from all the secondary particles ($P_{\text{secondary}}$) from the combined simulations and not only from the primary ones. The raw deposits on the sensors from different simulations are added together. In order to emulate realistic detector conditions, detector noise is added to the deposited energy at generator-level, as specified in Procedure 1: 6. The detector noise model consists of a generation of spurious energy measurements in the detector sensors, distributed according to a Gaussian probability density function centered at 0 and with a variance of $5 \cdot 10^{-6}$. An example of detector noise is shown in Fig. 5 (left).

Procedure 1 Event Generation

```

Input  $S$ 
Output  $H, T$ 
1:  $T \leftarrow \bigcup_{S \in \mathbb{S}} P_{\text{secondary}}(S)$ 
2:  $G_{\text{close}} \leftarrow (T, \{f_{\text{close}}(p_1, p_2) \mid (p_1, p_2) \in T \times T\})^1$ 
3:  $T \leftarrow \text{merge}(\text{connected\_components}(G_{\text{close}}))$ 
4: for each  $i \in \mathbb{S}$  do
5:    $t(i) \leftarrow \text{undef}$ 
6:    $\bar{e} \leftarrow \text{sample}(N(0, 5 \times 10^{-6}))$ 
7:    $e_{\text{max}} \leftarrow \max_T(d_{\text{raw}}(i, p) \mid p \in T)$ 
8:   if  $e_{\text{max}} > \bar{e}$  then
9:      $t(i) \leftarrow \text{argmax}_T(d_{\text{raw}}(i, p) \mid p \in T)$ 
10:   $\bar{e} \leftarrow \bar{e} + \sum_{p \in T} d_{\text{raw}}(i, p)$ 
11:   $e_i \leftarrow \hat{g}c_i \bar{e}_i$ 
12:  $H \leftarrow \{i \mid \forall i \in \mathbb{S} \exists \bar{e}_i / A_i > \rho\}$ 
13:  $T \leftarrow \{p \mid \forall p \in T \exists |h \forall h \in H \exists t(h) = p| > 0\}$ 

```

All sensors with uncalibrated deposited energy per sensor area (\bar{e}/A) greater than ρ are considered as the reconstructed hits (rechits) in the event (Procedure 1: 12). We choose the constant $\rho = 1.3 \cdot 10^{-7} \text{ GeV/mm}^2$, which corresponds to an uncalibrated energy ranging from $5.5 \cdot 10^{-3} \text{ MeV}$ to 3 MeV between the smallest and the largest sensor. A set of rechits H in the event is given as an input to the network. It is represented as a feature vector $V_{\text{feat}}(h) = [r, \eta, \phi, x, y, z, A, e]_h$ defined $\forall h \in H$, where r, η, ϕ (x, y, z) are the boost-invariant cylindrical coordinates (Cartesian coordinates) of the sensor, A is its area, and e is the deposited energy. Multiple particles can leave an energy deposit in a single sensor, however the particle that leaves the highest deposit is considered as the true candidate ($t(s)$) for that hit (Procedure 1: 9).

We define the ground truth as a realistic target for the reconstruction algorithm, applying a selection to the generator-level hits in order to take into account their overlap. For instance, when two particles (p_1 and p_2) are max-

imally overlapping, we merge them into a single particle in the ground truth since disentangling such two clusters would be an impractical and imprecise task. The $f_{\text{close}}(p_1, p_2)$ in Procedure 1: 2 evaluates whether the two showers are maximally overlapping if the following three conditions are met: their incident angles are closer in η than $1.5w_\eta$, in ϕ than $1.5w_\phi$, and if the difference in their showering angles is less than 0.09. w_η and w_ϕ represent the max of width in η and ϕ , respectively, of the first sensors that p_1 and p_2 hit. The showering angle is taken from GEANT4 by selecting the momentum direction from when the particles start showering.

The set of true particles (including merged particles) T that have at least one hit assigned to them after the filtering process are taken as the reconstruction target (Procedure 1: 13).

As an example, the 3D view of a generated event is shown on Fig. 5 (right). Detector hits belonging to different incoming primary particles are shown with different colors.

In total, six datasets are generated. Table 1 shows the number of hits and true particles for different datasets as a proxy for event complexity.

1. Training set: 5000 events, where each event is created from 200 Minimum Bias simulations and 60 Type B simulations. However, due to computational constraints, it is impractical to train with 200 pileup. To overcome this, we augment the data. For each pileup simulation, we randomly picks a point in $\phi_0 \sim U(0, 2\pi)$ and only select $P_{\text{secondary}}$ with impact directions between ϕ_0 and $\phi_0 + 30^\circ$. The particles originating from Type A simulations are left intact. Figure 6 shows an example event from the training set.
2. Single-particle testing set: 20,000 events, where each event is created from Type B simulation only.
3. PU40+1 testing set: 6800 events, where each event is created from 40 minimum bias simulations and 1 Type B simulation.
4. PU200+1 testing set: 6800 events, where each event is created from 200 minimum bias simulations and 1 Type B simulation.
5. PU40+ $t\bar{t}$ testing set: 6800 events, where each event is created from 40 minimum bias simulations and 1 $t\bar{t}$ simulation.
6. PU200+ $t\bar{t}$ testing set: 6800 events, where each event is created from 200 Minimum Bias simulations and 1 $q\bar{q} \rightarrow t\bar{t}$ simulation.

The training set with 5000 events would require 10^6 minimum bias simulations, but we could only generate 3.2×10^5 . Therefore, we randomly sample from the simulation set without replacement. This strategy ensures minimum overlap of

¹ Defines a graph as (nodes, edges)

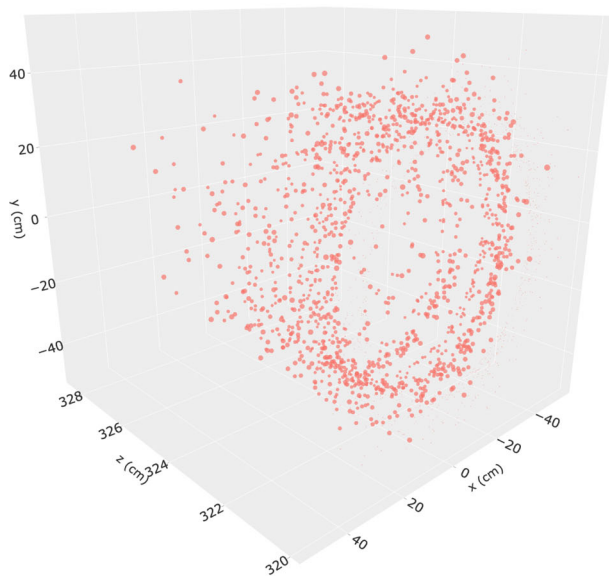
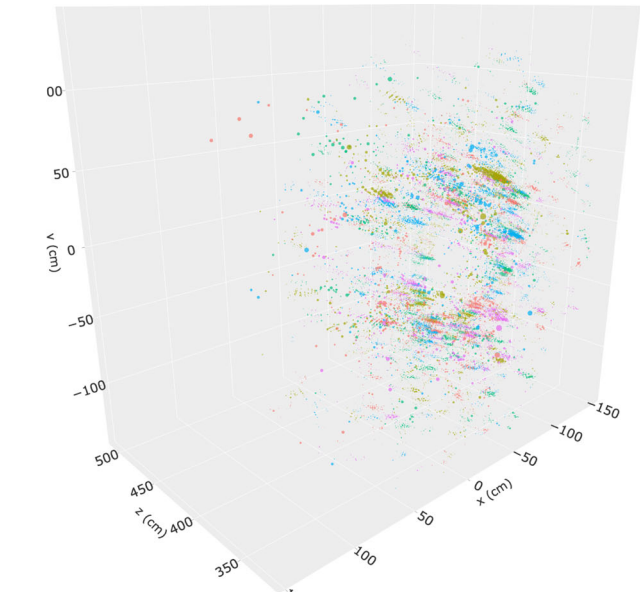


Fig. 5 Left: example of detector noise generated as described in the text. Right: example of a generated event, obtained from the overlap of 40 single-collision events and noise. The shower of secondary particles generated by an individual primary particle yields a set of energy deposits in the detector labeled through markers of different colors and



sizes. Different colors refer to different incoming primary particles. Marker sizes are proportional to $\log(e_h + 1)$ and are max normalized independently in both figures (the same method is used for choosing marker size for all the event displays in this article)

Table 1 Event complexity for different datasets

| Dataset | $ H $ | $ T $ |
|------------------------------|----------------------|----------------|
| Training set | $34,000 \pm 2000$ | 340 ± 18 |
| Single-particles testing set | 2600 ± 240 | 1.0 ± 0.0 |
| PU40 testing sets | $43,000 \pm 8000$ | 1000 ± 160 |
| PU200 testing sets | $160,000 \pm 12,000$ | 3200 ± 130 |

pileup between consecutive events and is used for generating all training and testing datasets.

5 Neural network and training

As discussed in Sect. 4, the input to the neural network is a set of hits. To create the edges needed to construct a graph, we use a dynamic graph neural network approach, GravNet [6]. GravNet dynamically computes the edges with the help of the k-nearest-neighbour algorithm, evaluated in a low dimensional learnable coordinate space. With each application of a GravNet layer, this coordinate space can change, and as a consequence, a different set of neighbours can be assigned to each vertex. Figure 7 visualises the model architecture, which is inspired by Ref. [20] with minor modifications.

Following the object condensation paradigm, the model makes a set of predictions per hit: the three-dimensional cluster space coordinates x_h , a condensation score β_h , and

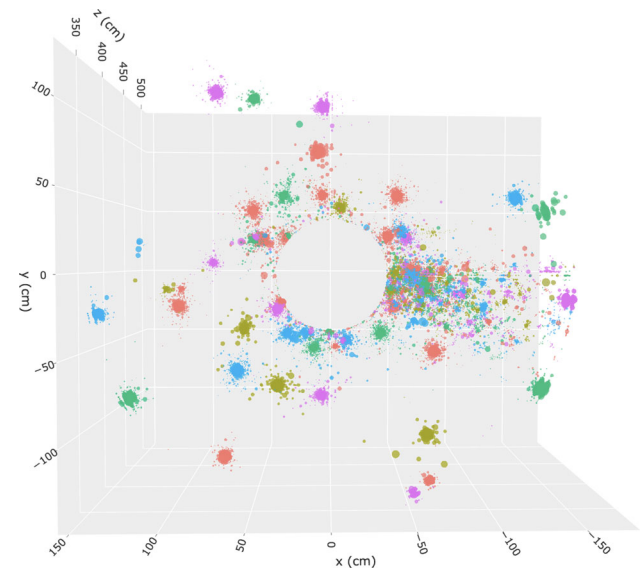
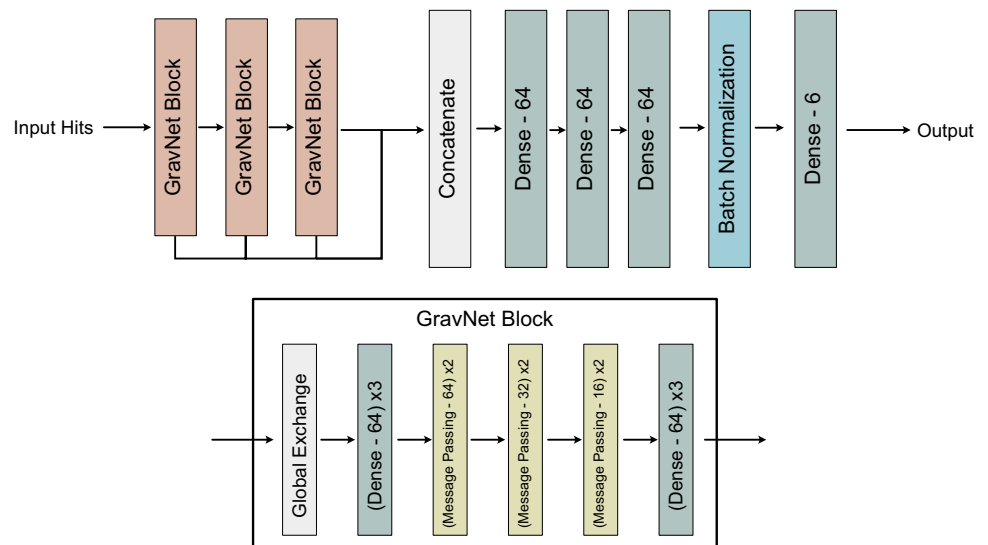


Fig. 6 Example of an event with pileup, part of the training set. No filter is applied to Type A simulations, but for pileup simulations, a filter, which selects particles (and their corresponding hits) between ϕ_0 and $\phi_0 + 30\sigma$, is applied. The high density of hits on the right side of the figure shows the slice in which pileup is present

other object properties, in this case an energy correction factor $\psi_h \approx 1$. This factor corrects the total deposited calibrated energy in the calorimeter cells, evaluated using truth, to match the impact energy of the particle the hit belongs to. The total calibrated deposited energy left by a particle on all

Fig. 7 Architecture of the model. Three GravNet blocks are used, each of them containing multiple message passing layers. In a ResNet-type architecture, the features output from these layers are concatenated and then passed through a dense net. For each hit, six outputs are produced. Three represent the clustering space, one β confidence, one distance threshold, and one energy correction factor



the sensors is defined as the deposited energy of the said particle. In addition, we introduce a distance measure φ_h , also per hit, that scales with the expected distance between one shower and hits from other showers or noise. This addition to object condensation allows us to keep the cluster coordinate space low dimensional, and therefore interpretable, while introducing another degree of freedom to adopt distances to locally dense environments. In total, the neural network has a 6-dimensional output per hit. These quantities are further explained in Sects. 6 and 7.

The model is trained for 68 epochs using the Adam optimizer [38] on the training set described in Sect. 4, at which point the loss becomes stable and does not further decrease with additional epochs. Each training batch consists of one event.¹ The training is performed within the DeepJetCore framework [39] and the models are implemented in Tensorflow [40] and Keras [41]. We run the training at the Flatiron GPU computing cluster using NVIDIA V100 GPUs.

6 Loss function

The object condensation method [8] aims at identifying a unique representative hit accumulating all shower properties for each shower, referred to as condensation points. Each hit is also embedded in the clustering space to resolve ambiguities and to assign the remaining hits to showers. The loss consists of three terms: the potential loss (L_V) is responsible for creating the clustering space and embedding hits in it, the condensation score loss (L_β) trains the network to identify the condensation points, and the payload loss L_P creates gradients for the other object properties, in our case the energy

correction factor. The relative contribution of the two loss terms is set by a factor s_C , which we take as $s_C = 1$ for this study.

$$L = L_V + s_C(L_\beta + L_P). \tag{3}$$

Especially for the hadronic showers, hits that are significantly displaced in position from the shower core are challenging to assign to their initiating incident particle. The mis-assignment of these hits to showers that are closer in space, but initiated by another incident particle is relatively common and difficult to avoid. Typically such hits are low energy, so their correct shower association is less critical to the task of estimating the total shower properties. With this in mind, we adapt the original object condensation method to reduce the impact of the mis-association of this class of hits in the network. Mathematically, this also serves to reduce the maximum fluctuations in the gradients. In comparison to Ref. [8], we adapt the calculation of the clustering charge q_h , and smoothen the potential terms. The clustering charge is calculated based on β_h with $0 \leq \beta_h \leq 1$. We rescale the calculation of q_h slightly to avoid strong gradients for $\beta_h \rightarrow 1$ as follows:

$$q_h = \operatorname{arctanh}^2(\beta_h/1.002) + q_{\min} \nu_h. \tag{4}$$

In addition, a new parameter ν_h that describes a spectator weight is introduced. Hits that are scattered far away from the shower core receive a smaller weight, in our case $\nu_h = 0.1$, while all other hits receive a weight of $\nu_h = 1$. To define whether a given hit should be considered a spectator, we first perform a principal component analysis (PCA) on the truth-assigned energy-weighted hit coordinates of the shower to identify the two principal components which act as the proxies for the shower axes. For this task we define the shower axes in two dimensions only, where the dimensionality is reduced by one due to the shower symmetry. The hits

¹ This choice is governed by the available GPU memory, not due to restrictions of the training framework.

belonging to the shower are then projected onto the defined shower axes. Using the projected coordinates, we compute the Mahalanobis distance [42] for each hit. We consider a hit a spectator if its Mahalanobis distance is larger than 3. For the attractive and repulsive potential losses, the hit α_t with the highest β score for each truth shower t , also taking into account the spectator weights, plays a special role. It is defined as:

$$\alpha_t = \operatorname{argmax}_{h \in H_t} (\beta_h v_h), \tag{5}$$

where H_t is the set of hits belonging to truth shower t . Furthermore, the β -weighted average learned distance scale $\bar{\varphi}_t$ for a truth shower t is calculated as:

$$\bar{\varphi}_t = \frac{\sum_{h \in H_t} \beta_h * \varphi_h}{\sum_{h \in H_t} \beta_h}. \tag{6}$$

Taking the weighted average over the shower as opposed to considering only the hit α_t has the advantage that it creates a more consistent gradient for φ_h to be learned for every hit. A similar approach is taken for the reference point in clustering space of the potentials that attract or repulse other hits. Here, the reference point for each truth shower t is calculated as

$$\bar{x}(t) = \frac{1}{2} \left(x_{\alpha_t} + \frac{\sum_{h \in H_t} (\beta_h q_h x_h)}{\sum_{h \in H_t} \beta_h q_h} \right). \tag{7}$$

This represents another modification of the original object condensation loss, which takes x_{α_t} only. The new term in the sum serves to remove noise from the training, while keeping a large impact of the hit α_t , which helps to resolve the degeneracy in the beginning of the training. Based on these ingredients, the attractive potential loss, \check{V}_h , is then re-defined as follows:

$$\check{V}_t(h) = q_{\alpha_t} w_t \ln \left(e \cdot \left(\frac{\|x_h - \bar{x}(t)\|^2}{2\bar{\varphi}_t^2 + \epsilon} \right) + 1 \right), \tag{8}$$

where w_t is the shower weight. For $E_{\text{true}} > 10$, $w_t = 1$. From 10 to 0.5 GeV, it linearly decreases from 1 to 0. ϵ is a small number that is added for numerical stability. The repulsive loss is modified accordingly, as

$$\hat{V}_t(h) = q_{\alpha_t} w_t \cdot \exp \left(-\frac{\|x_h - \bar{x}(t)\|^2}{2\bar{\varphi}_t^2 + \epsilon} \right). \tag{9}$$

The full potential loss function takes the form:

$$L_V = \frac{1}{|T|} \sum_{t \in T} \left(\frac{1}{|H_t|} \sum_{h \in H_t} q_h \check{V}_t(h) + \frac{1}{|H - H_t|} \sum_{h \in (H - H_t)} q_h \hat{V}_t(h) \right). \tag{10}$$

Here $H - H_t$ represents the set difference, i.e., all hits that are not assigned to shower t . The payload loss L_P is also

weighed by the object weight w_t to reduce the impact of low energy showers, that is

$$L_P = \sum_{t \in T} \frac{w_t}{\sum_{h \in H_t} \xi(h)} \sum_{h \in H_t} \xi(h) L_E, \tag{11}$$

with $\xi(h) = \operatorname{arctanh}^2(\beta_h/1.002)$. The energy loss contribution L_E is calculated as

$$L_E = \log \left(\left(\frac{E_{\text{true},t} - \psi_h E_{\text{dep},t}}{\sqrt{E_{\text{true},t}} + 0.003} \right)^2 + 1 \right), \tag{12}$$

where E_{dep} is the total energy collected in the calibrated calorimeter cells associated to the truth shower t .

The beta loss term consists of two parts and is identical to Ref. [8],

$$L_\beta = \frac{1}{|T|} \sum_{t \in T} (1 - \beta_{\alpha_t}) + s_B \frac{1}{|H_o|} \sum_{h \in H_o} \beta_h. \tag{13}$$

The first term ensures that at least one hit per truth shower is promoted to a condensation point. The second term suppresses noise. H_o represents the set of all noise hits. We choose the scaling factor $s_B = 1$.

7 Inference

We also extend the inference algorithm from Ref. [8] to reflect the introduction of the local distance scale ϕ_h . The algorithm is outlined in Procedure 2 and is applied to the learned clustering space. The algorithm starts with the hit with the highest β -score β_α and assigns all hits within a certain radius $t_d \cdot \phi_\alpha$ to it, with $t_d = 1.0$. These hits are removed for the next iteration. This procedure is repeated until the highest β -score is lower than the threshold $t_\beta = 0.3$. The remaining unassigned hits are considered noise. To determine the energy of the reconstructed cluster, we sum the energy of all hits assigned to a cluster collected around hit α and multiply this sum by the learned energy correction factor ψ_α .

Procedure 2 Clustering Inference

Input $H, \beta, x, \psi, \varphi, t_d, t_\beta$
Output P

- 1: $P \leftarrow \{\}$
- 2: $H_{\text{cand}} \leftarrow \{h \in H \forall h \in H \ni \beta_h > t_\beta\}$
- 3: $H_{\text{free}} \leftarrow H$
- 4: **while** $|H_{\text{cand}}| > 0$ **do**
- 5: $\alpha \leftarrow \operatorname{argmax}_h (\beta_h \forall h \in H_{\text{cand}})$
- 6: $p \leftarrow \text{NEW_PARTICLE}$
- 7: $H_p \leftarrow \{h \in H_{\text{free}}, \|x_h - x_\alpha\| < t_d \varphi_\alpha\}$
- 8: $E_{\text{pred}}(p) \leftarrow \psi_\alpha \sum_{h \in H_p} e_h$
- 9: $H_{\text{free}} \leftarrow H_{\text{free}} - H_p$
- 10: $H_{\text{cand}} \leftarrow H_{\text{cand}} - H_p$
- 11: $P \leftarrow P \cup \{p\}$

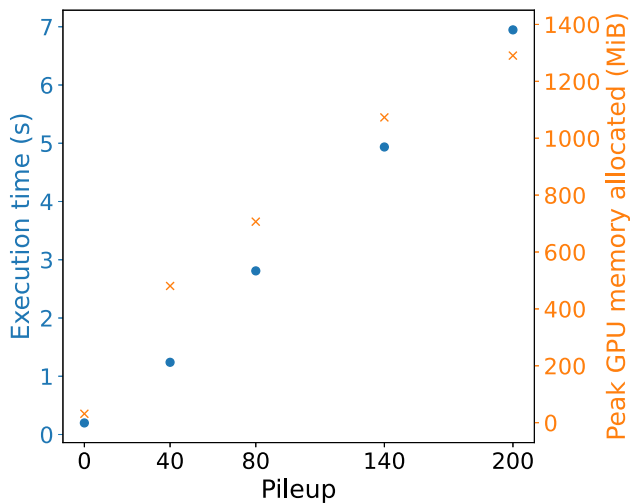


Fig. 8 Compute specifications of our model as a function of the amount of pileup to run inference with one event. The left axis (blue) shows average execution time, and the right (orange) shows the average peak memory allocated on the GPU

In Fig. 8, we show inference time and peak GPU memory required for single particle in different pileup conditions. The inference times are evaluated on a Nvidia V100 GPU. For single-particle events, inference takes about 200 ms. The inference time increases to 1.2 s or 7 s for 40 and 200 pileup, respectively. We expect an additional significant improvement of the overall inference time using edge-contraction methods to reduce the cardinality of the hits. These refinements will be considered in future work. These values should be compared to $\mathcal{O}(1000)$ s taken by currently adopted algorithms running on CPU, when scaled up to a 200 pileup environment.

Inference in 40 and 200 pileup allocates an average of only 500 MiB and 1300 MiB, respectively, on the GPU. This opens up the possibility that our method can be deployed on machines with less powerful GPUs with smaller VRAM. Note that a larger GPU is required for the training stage as memory can't be freed up after executing a neural network layer for backpropagation-related computations.

Here, the final inference algorithm was adapted to only consider close-by hits using a binning approach, making its contribution to the execution time negligible.

8 Physics performance

We evaluate physics performance in several ways by studying the reconstruction performance of the individual particles and jets. The individual particles, split in electromagnetic particles (e^- and γ) and hadronic particles (π^+), are studied separately as they exhibit different behaviors. Reconstruction efficiency, energy response, and resolution are studied

in different pileup environments, as well as the rate of reconstructed clusters that are either split off from the main shower (unmatched showers). For jets, we investigate the response and resolution in different pileup environments, assuming per-particle pileup removal procedures are in place.

The metrics are studied as a function of the p_T of the particles and jets. The neural network is regressing only the particles' energy, but for the computation of their p_T , we use energy-weighted mean hit positions to estimate the particles' direction. For consistency, we also use the same methodology to compute truth-level p_T .

Figure 9 visually shows the predictions of the neural network and compares them to the truth for both individual particle reconstruction and jet reconstruction.

8.1 Particle reconstruction performance

We begin by studying the performance in 0 pileup. These events contain only one probe truth particle and some detector noise. The probe particle is taken from Type B simulations as discussed in Sect. 4. We then overlay the probe particle with 40 and 200 pileup interactions to study performance in a more controlled fashion. While our method reconstructs all the particles in the event, including all the particles from the pileup, we only study the reconstruction performance of the probe particle.

First, we match the probe shower to one of the predicted showers by applying a hit-based matching procedure that we already introduced in Ref. [20]. The procedure calculates energy weighted hit-intersection over hit-union score (EIOU) of a reconstructed cluster p and a truth shower \hat{t} . The predicted shower that results in the highest overlap is taken as the matched shower (\hat{p}):

$$\hat{p} = \operatorname{argmax}_{p \in P} (\text{EIOU}(\hat{t}, p)). \quad (14)$$

We apply a lower threshold of 0.5 to the EIOU score to study reconstruction efficiency which is shown in Figs. 10a and 11a for electromagnetic and hadronic particles respectively.

The efficiency rises steeply with the increase in p_T in both electromagnetic and hadronic cases. In 0 pileup, the efficiency reaches a plateau of almost one at $p_T > 1$ GeV for electromagnetic particles while it remains slightly lower for the hadronic particles with $p_T < 15$ GeV. As expected, because of the dense environment, the performance drops as the pileup is increased. In 40 pileup, the reconstruction efficiency of the electromagnetic particles deteriorates to around 80% at 1-15 GeV with a significant drop for $p_T < 1$ GeV. For the hadronic particles the reconstruction efficiency drops to around 70% at 5-20 GeV.

The efficiency deterioration occurs when the neural network oversplits the showers and these split showers fail to satisfy the matching criterion. Unlike electromagnetic show-

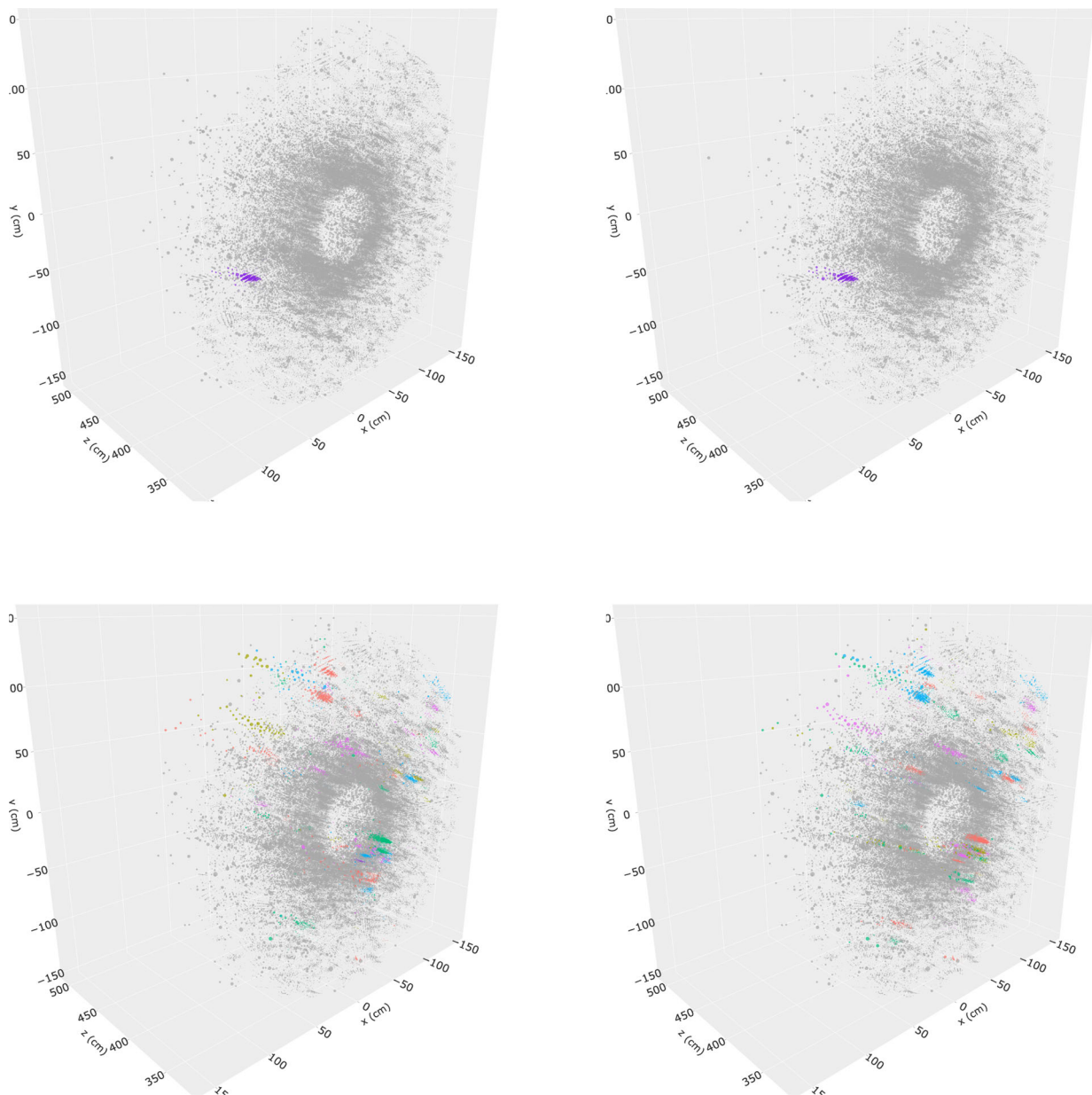


Fig. 9 True versus predicted cluster examples in 200 pileup. In the top two figures, a single particle is shot into the calorimeter, where the left and right figures show the true and the corresponding matched predicted cluster, respectively. The bottom row shows particles originating from

$q\bar{q} \rightarrow t\bar{t}$ collision in colors while the grey is 200 pileup. The predicted clusters (right) are matched to the true clusters (left). Jet reconstruction performance is studied on these true and matched clusters while the pileup is ignored

ers, the tendency to oversplit is inherent to the nature of hadrons which is why the hadronic efficiency drops also at high p_T when pileup is added. Therefore, to study the oversplits, we use energy-weighted intersection over minimum (EIOM), defined below:

$$\text{EIOM}(t, p) = \frac{\sum_{h \in (H_t \cap H_p)} e_h}{\min(\sum_{h \in H_t} e_h, \sum_{h \in H_p} e_h)}.$$

Unmatched showers are all the predicted clusters with $\text{EIOM} > 0.9$ with the truth-level probe particle but with EIOU less than 0.5 and these are shown in Fig. 12. The unmatched rate decreases steeply with the predicted p_T . This indicates that low p_T clusters are split off from higher- p_T showers while most of the energy is reconstructed properly. We note, that by adding tracking information and employing a suitable particle flow algorithm, these splits could be re-merged, increasing the efficiency. In addition to oversplit-

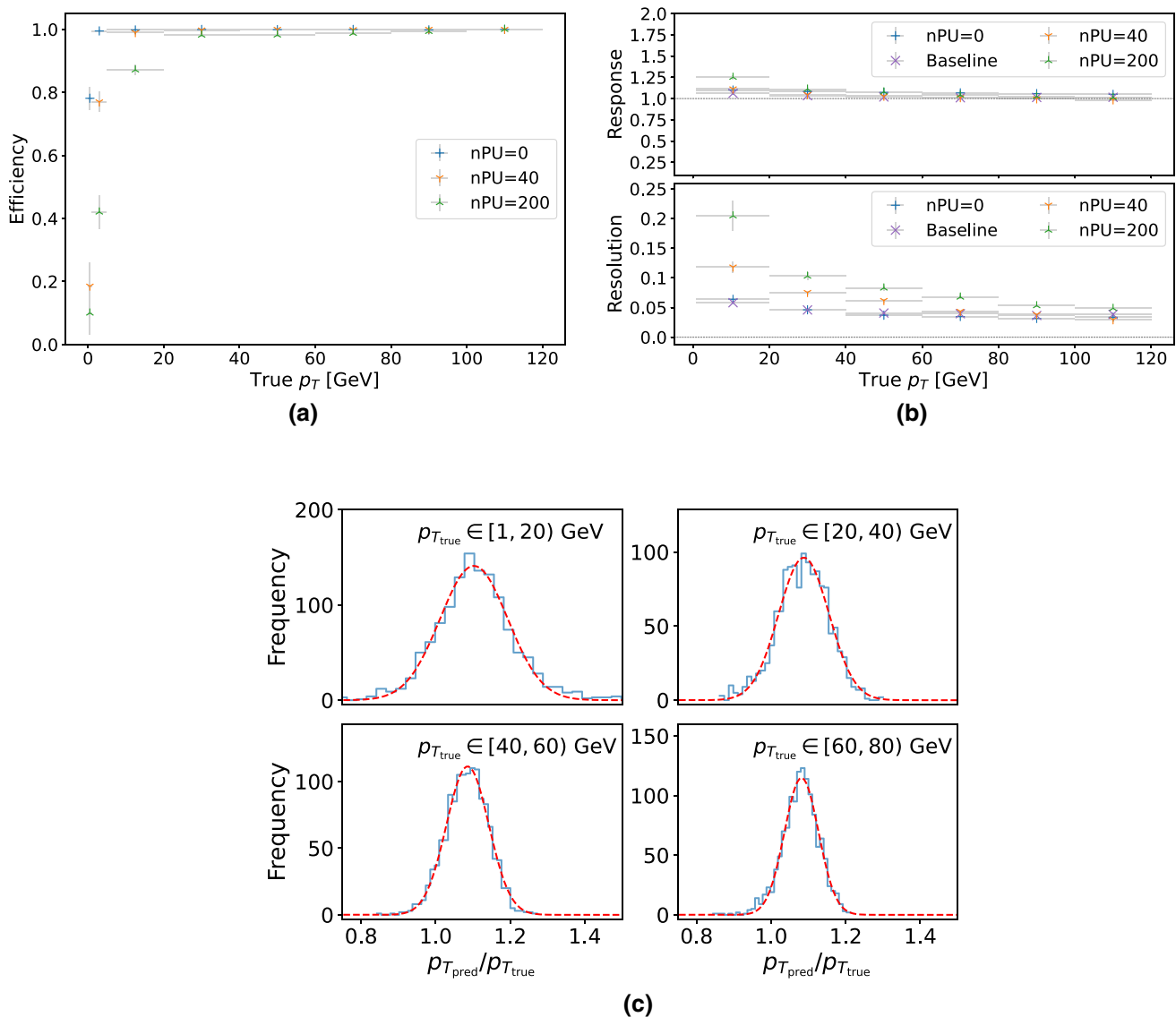


Fig. 10 Reconstruction performance of electromagnetic particles (photons and electrons) in different pileup environments. **a** Efficiency as a function of the true p_T . **b** Mean response (top) and resolution (bottom) as a function of the true p_T . The response and resolution are

computed as the mean and mean-corrected standard deviation of the Gaussian fit to the $p_{T, \text{pred}}/p_{T, \text{true}}$ distribution in individual p_T bins. **c** The distribution of p_T response in different p_T ranges corresponding to the first four bins in (a) and (b) in 0 pileup environment

ting, it could be possible that the neural network creates fake showers from the noise hits only. However, we observed that the fake rate is close to zero for p_T above 1 GeV. Additionally, only 0.5% of the total noise energy is assigned to a predicted cluster on average.

For the matched true showers, we compare the true p_T and, for the corresponding predicted shower, the regressed p_T . As a baseline, we define a truth-assisted p_T reconstruction that considers the true deposited energy as reconstructed energy which is also then compared to the true shower p_T . In Figs. 10b and 11b, we show the p_T regression performance scaling with the truth p_T for electromagnetic and hadronic

particles, respectively. To compute the response and resolution, we fit the distribution of the p_T response, $p_{T, \text{pred}}/p_{T, \text{true}}$, with a Gaussian function independently in each p_T bin. Figures 10c and 11c show the p_T response distributions and the Gaussian functions for the first four bins. Mean (μ) and mean-corrected standard deviation (σ/μ) of the fitted Gaussian functions are taken as the response and resolution. The response can be corrected a posteriori, although it serves as an important metric for the algorithm’s behavior, while the resolution directly reflects the p_T reconstruction performance.

Due to the nature of hadronic showers and because the cell energies are calibrated on electromagnetic showers, the base-

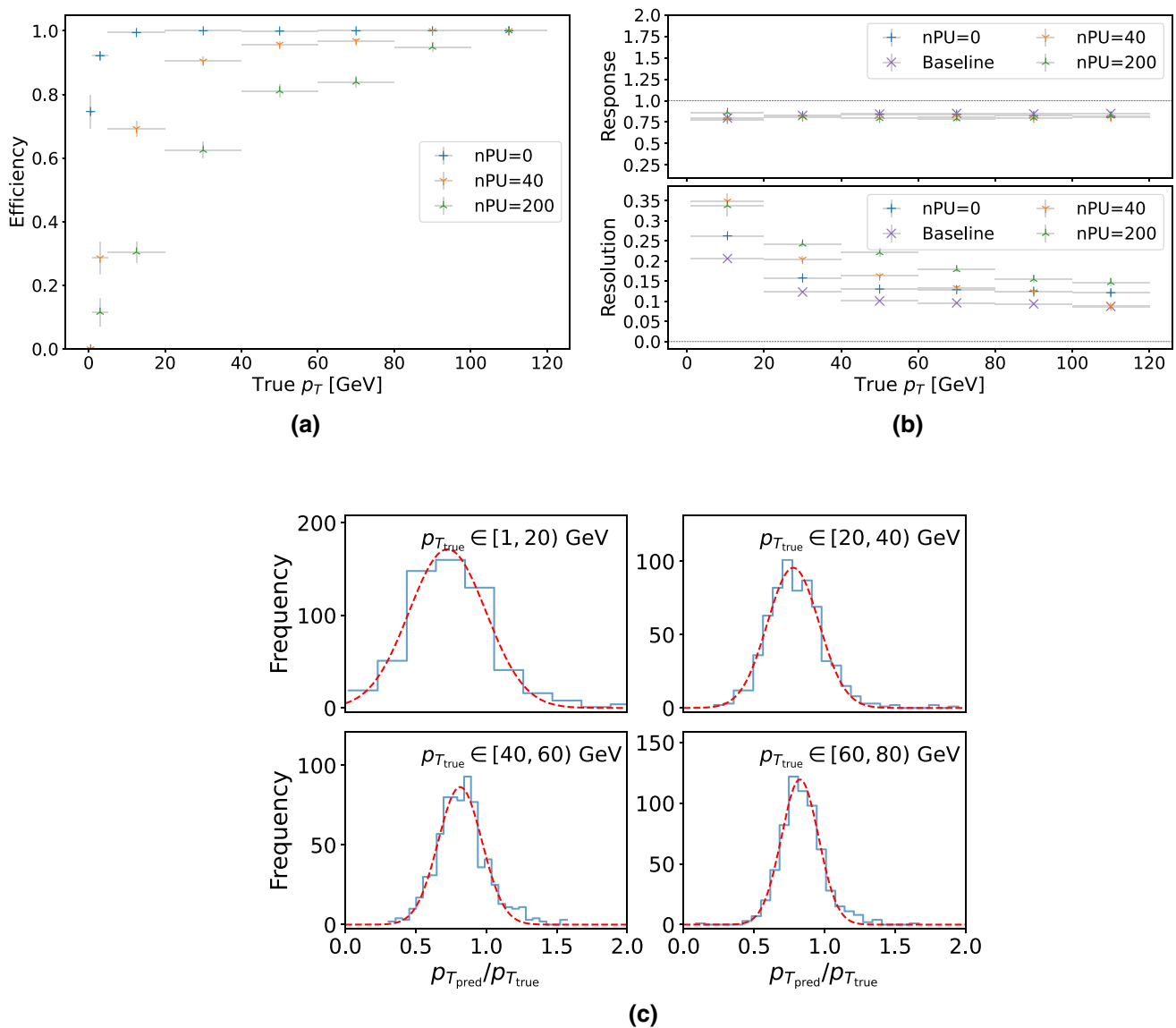


Fig. 11 Reconstruction performance of hadrons (π^+) in different pileup environments. **a** Efficiency as a function of the true p_T . **b** Mean response (top) and resolution (bottom) as functions of the true p_T . The response and resolution are computed as the mean and mean-corrected

standard deviation of the Gaussian fit to the $p_{T, \text{pred}}/p_{T, \text{true}}$ distribution in individual p_T bins. **c** The distribution of response in different p_T ranges corresponding to the first four bins in (a) and (b) in 0 pileup environment

line response for charged pions is below one while it is compatible with one for electromagnetic showers, in particular at high energies. The reconstructed p_T response provided by the network only differs mildly from the baseline response. The difference to unity response increases by about a factor of two for charged pions if the energy correction factor is not applied (not shown), indicating that the network is capable of distinguishing different shower types.

As expected, the resolution improves with true p_T and degrades with increase in pileup. Also, here, even the reconstructed hadronic shower resolution is close to the baseline reconstruction and converges to $\sim 15\%$ above 60 GeV,

even in high pileup. The reconstructed energy resolution in 0 pileup for electromagnetic showers is almost indistinguishable from the baseline and, therefore, close to the detector limitations. In 40 pileup, the electromagnetic resolution deviates from the baseline only at low p_T but approximates the baseline at high p_T and in 200 pileup, it deviates slightly more.

The neural network was trained on events with hundreds of true showers as shown in Table 1 however it offers an excellent generalization performance on a vastly different datasets, especially where only one particle is present (0 pileup environment). The fact that we do not observe the

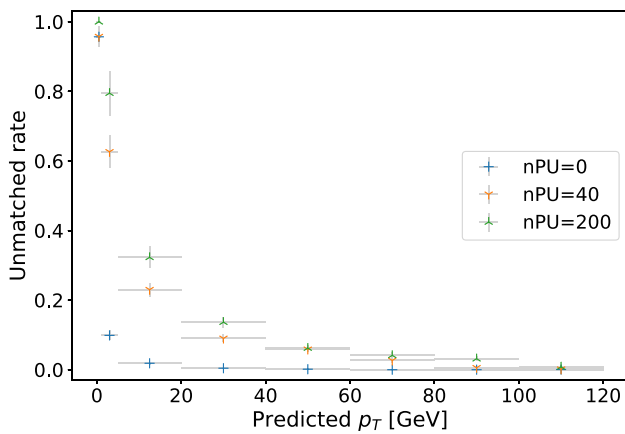
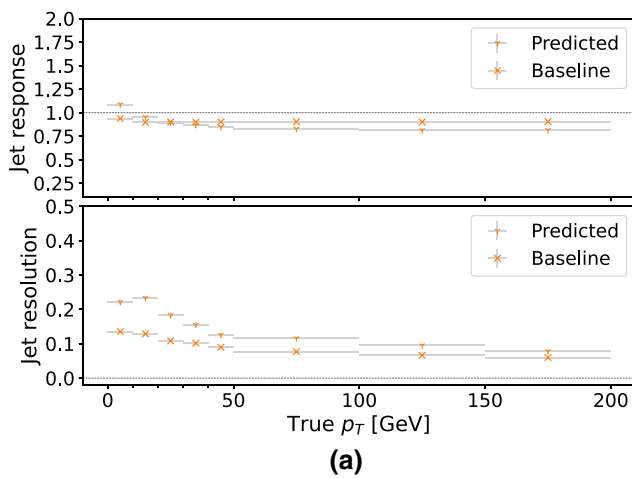
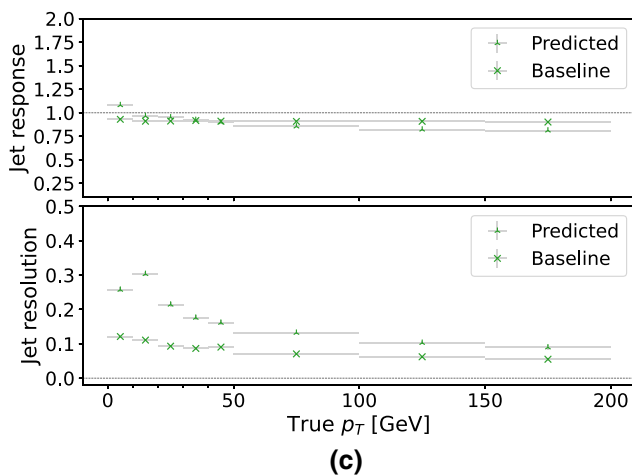


Fig. 12 Unmatched rate as a function of predicted p_T



(a)

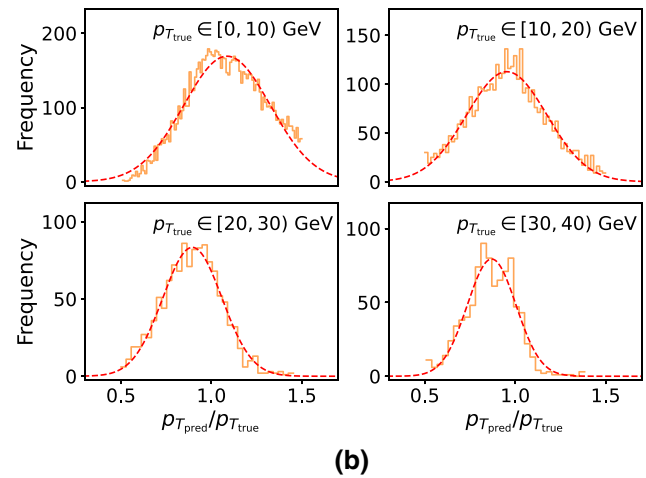


(c)

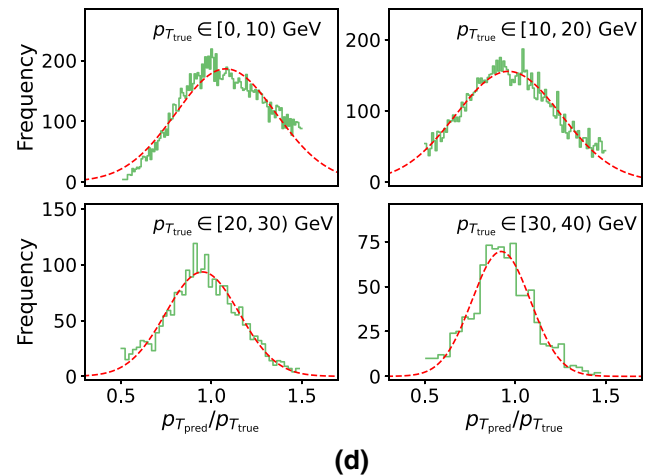
creation of fakes demonstrates that the neural network has correctly learned to cluster using only local information. This increases confidence in the extrapolation capabilities of the network and training method beyond training conditions in general.

8.2 Jet reconstruction performance

Jets and their substructure are an integral ingredient for the analysis of particle collisions. In particular in the forward region, well-resolved individual jet constituents are crucial for a successful pileup removal and for identifying e.g. quark jets over gluon jets in vector-boson-scattering or fusion processes. As modern jet clustering algorithms are infrared and collinear safe, jets also offer a way to gauge the



(b)



(d)

Fig. 13 Jet reconstruction performance. **a** Mean jet response (top) and resolution (bottom) in 40 pileup as functions of the true p_T . **b** Response distributions and the fitted Gaussian functions in different p_T ranges corresponding to **(a)**. **c** Mean jet response (top) and resolution (bottom) in 200 pileup as functions of the true p_T . **d** Response distributions and the fitted Gaussian functions in different p_T ranges corresponding to

(c). In both 40 and 200 pileup, the response and resolution are computed as the mean and mean-corrected standard deviation of the Gaussian fit to the $p_{T\text{pred}}/p_{T\text{true}}$ distribution in individual p_T bins. We note that the better resolution in the first p_T bin with the respect to the second bin is an artifact of the biased Gaussian fit due to the asymmetry of the response distribution, as shown in the top-left subplots of **(b)** and **(d)**

performance of the calorimeter clustering algorithm without strong dependencies of subtleties in the definition of the single-particle truth. Moreover, oversplitting and overmerging due to the reconstruction process have less impact on the cumulative jet quantities. Generally, a pileup removal algorithm is applied before jet clustering to remove contributions from particles not associated to the primary collision. As discussed in Sect. 4, we generate $q\bar{q} \rightarrow t\bar{t}$ events on top of either 40 or 200 minimum bias events, as these events provide significant hadronic activity in the forward region. After the single-particle reconstruction is performed, a pileup removal algorithm is simulated aided by truth information: We remove all showers that originate from pileup interactions unless they share more than 10% of their energy-weighted hits with a reconstructed non-pileup shower. All remaining reconstructed showers are considered for clustering the reconstructed jets. To form the truth jets, we only consider truth showers that stem from the non-pileup interaction. We also define a baseline reconstruction algorithm based on the true deposited energy of the incident non-pileup particles. Jets are then clustered using the anti-kt algorithm [43] with a distance parameter of $R = 0.4$. Reconstructed and truth jets are matched based on a $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$ matching. Among all jets with $|\Delta p_T|/p_{T\text{true}} < 0.5$ and $\Delta R < 0.3$, we select the best match by minimum ΔR .

Following the procedure performed for individual particle reconstruction, we fit Gaussian functions to the p_T response distribution in each p_T bin. The mean (μ) and the mean-corrected standard deviation (σ/μ) of the Gaussian function are taken as jet response and resolution. The distributions and the fitted functions are shown in Fig. 13a, and the response and resolution are shown in Fig. 13a, c for 40 and 200 pileup, respectively.

The response falling below one is a direct consequence of the single-particle responses shown in Fig. 11b. At higher energies, the resolution starts to approximate the baseline. At lower energies, the presence of pileup degrades the performance slightly, however much less than in the case of single particles. As jets are less affected by truth matching and splitting effects, the assumption that single-particle performance depends on the matching procedure and the splitting of showers is verified. As shown in Fig. 13c, the performance degrades in 200 pileup compared to 40 pileup.

9 Conclusion and further research

In this paper, we presented the first demonstration that end-to-end reconstruction in high granularity calorimeters using graph neural networks is a feasible method for event reconstruction in very dense particle physics collisions. We perform this task in a single-step approach, in which the particle hits in the detector are taken as inputs and clustered showers,

including their corrected energy, are output with no intermediate steps. Our model has been built using the GravNet graph neural network and the object condensation approach. We have evaluated the performance of our model on both single-particle events and physics events with pileup, and demonstrated promising performance of energy resolution and response for single particles, as well as for clustered jets. The method shows excellent generalisation properties from single-particle events to dense jets in 200 pileup.

The proposed method also provides computational advantages, allowing one to exploit GPU acceleration at inference hence reducing the average inference time by at least two orders of magnitude with respect to high-pileup projections of the currently employed algorithms designed for CPUs.

Currently, our model does not perform particle identification on the cluster object, which will be further studied in future work. In addition, measurements from additional detector subsystems, such as timing or tracking information, could be included to extend the one-shot reconstruction to a full particle flow prediction. The results presented here pave the way for such extensions and represent an initial demonstration that one-shot inference is an effective and efficient means for reconstruction in dense particle physics collisions.

Acknowledgements This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 772369). We thank Ian Fisk and the Flatiron Institute of the Simons Foundation for providing access to the GPU cluster used to perform these studies.

Data Availability Statement This manuscript has no associated data or the data will not be deposited. [Authors' comment: The data used in this paper is very large and it can be reproduced with GEANT4. We are also willing to share the GEANT4 addon software that was used to produce the dataset with individuals upon request.]

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Funded by SCOAP³. SCOAP³ supports the goals of the International Year of Basic Sciences for Sustainable Development.

References

1. J. Albrecht et al., A roadmap for HEP software and computing R&D for the 2020s. *Comput. Softw. Big Sci.* **3**(1), 7 (2019)

2. A.M. Sirunyan et al., Electron and photon reconstruction and identification with the CMS experiment at the CERN LHC. *JINST* **16**(05), P05014 (2021)
3. K. Albertsson et al., Machine learning in high energy physics community white paper. *J. Phys. Conf. Ser.* **1085**(2), 022008 (2018)
4. CMS Collaboration, The Phase-2 Upgrade of the CMS Endcap Calorimeter. Technical report (2017)
5. I. Zurbano Fernandez et al., High-Luminosity Large Hadron Collider (HL-LHC): Technical design report. 10/2020 (2020)
6. S.R. Qasim et al., Learning representations of irregular particle-detector geometry with distance-weighted graph networks. *Eur. Phys. J. C* **79**(7), 608 (2019)
7. Y. Iiyama et al., Distance-weighted graph neural networks on FPGAs for real-time particle reconstruction in high energy physics. *Front. Big Data* **3**, 598927 (2020)
8. J. Kieseler, Object condensation: one-stage grid-free multi-object reconstruction in physics detectors, graph and image data. *Eur. Phys. J. C* **80**(9), 886 (2020)
9. F. Pantaleo et al., Development of a phase-II track trigger based on GPUs for the CMS experiment, in *2015 IEEE nuclear science symposium and medical imaging conference*, p. 7581775 (2016)
10. D. Rohr et al., GPU-accelerated track reconstruction in the ALICE High Level Trigger. *J. Phys. Conf. Ser.* **898**(3), 032030 (2017)
11. M. Rovere et al., CLUE: a fast parallel clustering algorithm for high granularity calorimeters in high-energy physics. *Front. Big Data* **3**, 591315 (2020)
12. R. Aaij et al., Allen: a high level trigger on GPUs for LHCb. *Comput. Softw. Big Sci.* **4**(1), 7 (2020)
13. A. Bocci et al., Bringing heterogeneity to the CMS software framework. *EPJ Web Conf.* **245**, 05009 (2020)
14. R. Aaij et al., A comparison of CPU and GPU implementations for the LHCb Experiment Run 3 Trigger. *Comput. Softw. Big Sci.* **6**(1), 1 (2022)
15. D. Rohr, Usage of GPUs in ALICE online and offline processing during LHC Run 3. *EPJ Web Conf.* **251**, 04026 (2021)
16. LHCb Collaboration, LHCb Upgrade GPU High Level Trigger Technical Design Report. Technical report. CERN, Geneva (2020)
17. P. Buncic et al., Technical Design Report for the Upgrade of the Online-Offline Computing System. Technical report (2015)
18. CMS Collaboration, The Phase-2 Upgrade of the CMS Data Acquisition and High Level Trigger. Technical report. CERN, Geneva (2021) (**This is the final version of the document, approved by the LHCC**)
19. S. Bhattacharya et al., Gnn-based end-to-end reconstruction in the cms phase 2 high-granularity calorimeter (2022). [arXiv:2203.01189](https://arxiv.org/abs/2203.01189)
20. S.R. Qasim et al., Multi-particle reconstruction in the high granularity calorimeter using object condensation and graph neural networks, in *EPJ Web of Conferences*, EDP Sciences, vol. 251, p. 03072 (2021)
21. X. Ju et al., Graph neural networks for particle reconstruction in high energy physics detectors, in *33rd annual conference on neural information processing systems* (2020)
22. N. Choma et al., Track seeding and labelling with embedded-space graph neural networks. *FERMILAB-CONF-20-387-PPD-QIS-SCD* (2020)
23. X. Ju et al., Performance of a geometric deep learning pipeline for HL-LHC particle tracking. *Eur. Phys. J. C* **81**(10), 876 (2021)
24. G. DeZoort et al., Charged particle tracking via edge-classifying interaction networks. *Comput. Softw. Big Sci.* **5**(1), 26 (2021)
25. L. De Oliveira, B. Nachman, M. Paganini, Electromagnetic showers beyond shower shapes. *Nucl. Instrum. Methods A* **951**, 162879 (2020)
26. D. Belayneh et al., Calorimetry with deep learning: particle simulation and reconstruction for collider physics. *Eur. Phys. J. C* **80**(7), 688 (2020)
27. N. Akchurin et al., On the use of neural networks for energy reconstruction in high-granularity calorimeters. *J. Instrum.* **16**(12), P12036 (2021)
28. M. Andrews et al., End-to-end physics event classification with CMS open data: applying image-based deep learning to detector data for the direct classification of collision events at the LHC. *Comput. Softw. Big Sci.* **4**(1), 6 (2020)
29. M. Andrews et al., End-to-end jet classification of quarks and gluons with the CMS Open Data. *Nucl. Instrum. Methods A* **977**, 164304 (2020)
30. M. Andrews et al., Accelerating end-to-end deep learning for particle reconstruction using CMS open data. *EPJ Web Conf.* **251**, 03057 (2021)
31. S. Thais et al., Graph neural networks in particle physics: implementations, innovations, and challenges (2022). [arXiv:2203.12852](https://arxiv.org/abs/2203.12852)
32. J. Pata et al., MLPF: efficient machine-learned particle-flow reconstruction using graph neural networks. *Eur. Phys. J. C* **81**(5), 381 (2021)
33. A.A. Pol et al., Jet single shot detection. *EPJ Web Conf.* **251**, 04027 (2021)
34. A.A. Pol et al., Lightweight jet reconstruction and identification as an object detection task (2022)
35. S. Agostinelli et al., Geant4—a simulation toolkit. *Nucl. Instrum. Methods Phys. Res. Sect. A* **506**(3), 250–303 (2003)
36. CMS Collaboration, The Phase-2 Upgrade of the CMS Endcap Calorimeter. Technical report. CERN, Geneva (2017)
37. T. Sjöstrand et al., An introduction to PYTHIA 8.2. *Comput. Phys. Commun.* **191**, 159–177 (2015)
38. D.P. Kingma, J. Ba, Adam: a method for stochastic optimization (2014). [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
39. J. Kieseler et al., DeepJetCore (2020)
40. M. Abadi et al., TensorFlow: large-scale machine learning on heterogeneous systems (2015). <https://www.tensorflow.org/>
41. F. Chollet et al., Keras (2015). <https://keras.io>
42. P.C. Mahalanobis, On the generalized distance in statistics. National Institute of Science of India (1936)
43. M. Cacciari, G.P. Salam, G. Soyez, The anti-kt jet clustering algorithm. *J. High Energy Phys.* **2008**(04), 063 (2008)