Eur. Phys. J. C (2021) 81:381
https://doi.org/10.1140/epjc/s10052-021-09158-w

THE EUROPEAN PHYSICAL JOURNAL C

Regular Article - Experimental Physics

MLPF: efficient machine-learned particle-flow reconstruction using graph neural networks

Joosep Pata[1,2,a], **Javier Duarte**[3,b], **Jean-Roch Vlimant**[2,c], **Maurizio Pierini**[4,d], **Maria Spiropulu**[2,e]

[1] National Institute of Chemical Physics and Biophysics (NICPB), Rävala pst 10, 10143 Tallinn, Estonia
[2] California Institute of Technology, Pasadena, CA 91125, USA
[3] University of California San Diego, La Jolla, CA 92093, USA
[4] European Center for Nuclear Research (CERN), 1211 Geneva 23, Switzerland

Received: 5 February 2021 / Accepted: 19 April 2021 / Published online: 2 May 2021

Abstract In general-purpose particle detectors, the particle-flow algorithm may be used to reconstruct a comprehensive particle-level view of the event by combining information from the calorimeters and the trackers, significantly improving the detector resolution for jets and the missing transverse momentum. In view of the planned high-luminosity upgrade of the CERN Large Hadron Collider (LHC), it is necessary to revisit existing reconstruction algorithms and ensure that both the physics and computational performance are sufficient in an environment with many simultaneous proton–proton interactions (pileup). Machine learning may offer a prospect for computationally efficient event reconstruction that is well-suited to heterogeneous computing platforms, while significantly improving the reconstruction quality over rule-based algorithms for granular detectors. We introduce MLPF, a novel, end-to-end trainable, machine-learned particle-flow algorithm based on parallelizable, computationally efficient, and scalable graph neural network optimized using a multi-task objective on simulated events. We report the physics and computational performance of the MLPF algorithm on a Monte Carlo dataset of top quark–antiquark pairs produced in proton–proton collisions in conditions similar to those expected for the high-luminosity LHC. The MLPF algorithm improves the physics response with respect to a rule-based benchmark algorithm and demonstrates computationally scalable particle-flow reconstruction in a high-pileup environment.

J. Pata: This work was partially carried out at Caltech.

[a] e-mail: joosep.pata@cern.ch (corresponding author)
[b] e-mail: jduarte@ucsd.edu
[c] e-mail: jvlimant@caltech.edu
[d] e-mail: maurizio.pierini@cern.ch
[e] e-mail: smaria@caltech.edu

1 Introduction

Reconstruction algorithms at general-purpose high-energy particle detectors aim to provide a holistic, well-calibrated physics interpretation of the collision event. Variants of the particle-flow (PF) algorithm have been used at the CELLO [1], ALEPH [2], H1 [3], ZEUS [4,5], DELPHI [6], CDF [7–9], D0 [10], CMS [11] and ATLAS [12] experiments to reconstruct a particle-level interpretation of high-multiplicity hadron collision events, given individual detector elements such as tracks and calorimeter clusters from a multi-layered, heterogeneous, irregular-geometry detector. The PF algorithm generally correlates tracks and calorimeter clusters from detector layers such as the electromagnetic calorimeter (ECAL), hadron calorimeter (HCAL) and others to reconstruct charged and neutral hadron candidates as well as photons, electrons, and muons with an optimized efficiency and resolution. Existing PF reconstruction implementations are tuned using simulation for each specific experiment because detailed detector characteristics and geometry are critical for the best possible physics performance.

Recently, there has been significant interest in adapting the PF reconstruction approach for future high-luminosity experimental conditions at the CERN Large Hadron Collider (LHC) [13], as well as for proposed future collider experiments such as the Future Circular Collider (FCC) [14,15]. PF reconstruction is also a key driver in the detector design for future lepton colliders [16–18]. While reconstruction algorithms are often based on an imperative, rule-based approach, the use of supervised machine learning (ML) to define reconstruction parametrically based on data and simulation samples may improve the physics reach of the experiments by allowing a more detailed reconstruction to be deployed given a fixed computing budget. Reconstruction algorithms based on ML may be well-suited to irregular, high-granularity

detector geometries and for novel signal models, where it may not be feasible to encode the necessary granularity in the ruleset. A fully probabilistic particle-level interpretation of the event from an ML-based reconstruction may also improve the physics performance of downstream algorithms such as jet tagging with more granular inputs. At the same time, ML-solutions for computationally intensive problems may offer a modern computing solution that may scale better with the expected progress on ML-specific computing infrastructures, e.g., at high-performance computing centers.

ML-based reconstruction approaches using GNNs [19–23] have been proposed for various tasks in particle physics [24], including tracking [25–29], jet finding [30–32] and tagging [33–36], calorimeter reconstruction [37], pileup mitigation [38], and PF reconstruction [39–41]. The clustering of energy deposits in detectors with a realistic, irregular-geometry detector using GNNs has been first proposed in Ref. [37]. The ML-based reconstruction of overlapping signals without a regular grid was further developed in Ref. [39], where an optimization scheme for reconstructing a variable number of particles based on a potential function using an object condensation approach was proposed. The clustering of energy deposits from particle decays with potential overlaps is an essential input to PF reconstruction. In Ref. [40], various ML models including GNNs and computer-vision models have been studied for reconstructing neutral hadrons from multi-layered granular calorimeter images and tracking information. In particle gun samples, the ML-based approaches achieved a significant improvement in neutral hadron energy resolution over the default algorithm, which is an important step towards a fully parametric, simulation-driven reconstruction using ML.

In this paper, we build on the previous ML-based reconstruction approaches by extending the ML-based PF algorithm to reconstruct particle candidates in events with a large number of simultaneous pileup (PU) collisions. In Sect. 2, we propose a benchmark dataset that has the main components for a particle-level reconstruction of charged and neutral hadrons with PU. In Sect. 3, we propose a GNN-based machine-learned particle-flow (MLPF) algorithm where the runtime scales approximately linearly with the input size. Furthermore, in Sect. 4, we characterize the performance of the MLPF model on the benchmark dataset in terms of hadron reconstruction efficiency, fake rate and resolution, comparing it to the baseline PF reconstruction, while also demonstrating using synthetic data that MLPF reconstruction can be computationally efficient and scalable. Finally, in Sect. 5 we discuss some potential issues and next steps for ML-based PF reconstruction.

2 Physics simulation

We use PYTHIA 8 [42,43] and DELPHES 3 [44] from the HepSim software repository [45] to generate a particle-level dataset of 50,000 top quark–antiquark ($t\bar{t}$) events produced in proton–proton collisions at 14 Te V, overlaid with minimum bias events corresponding to a PU of 200 on average. The $t\bar{t}$ dataset is used for training the MLPF model. We additionally generate 5000 events composed uniquely of jets produced through the strong interaction, referred to as quantum chromodynamics (QCD) multijet events, with the same PU conditions for validation to evaluate the model in a different physics regime from the training dataset. The dataset consists of detector hits as the input, generator particles as the ground truth and reconstructed particles from DELPHES for additional validation. The QCD sample uses a minimum invariant p_T of 20 Ge V, otherwise, the same generator settings are used as for the $t\bar{t}$ sample. The DELPHES model corresponds to a CMS-like detector with a multi-layered charged particle tracker, an electromagnetic and hadron calorimeter. The full PYTHIA 8 and DELPHES data cards are available on Zenodo along with the dataset [46].

Although this simplified simulation does not include important physics effects such as pair production, Brehmsstrahlung, nuclear interactions, electromagnetic showering or a detailed detector simulation, it allows the study of overall per-particle reconstruction properties for charged and neutral hadrons in a high-PU environment. Different reconstruction approaches can be developed and compared on this simplified dataset, where the expected performance is straightforward to assess, including from the aspect of computational complexity.

The inputs to PF are charged particle tracks and calorimeter clusters. We use these high-level detector inputs (elements), rather than low-level tracker hits or unclustered calorimeter hits to closely follow how PF is implemented in existing reconstruction chains, where successive reconstruction steps are decoupled, such that each step can be optimized and characterized individually. In this toy dataset, tracks are characterized by transverse momentum (p_T),[1] charge, and the pseudorapidity and azimuthal angle coordinates (η, ϕ), including extrapolations to the tracker edge ($\eta_{\text{outer}}, \phi_{\text{outer}}$).

The track η and ϕ coordinates are additionally smeared with a 1% Gaussian resolution to model a finite tracker resolution. Calorimeter clusters are characterized by electro-

[1] As common for collider physics, we use a Cartesian coordinate system with the z axis oriented along the beam axis, the x axis on the horizontal plane, and the y axis oriented upward. The x and y axes define the transverse plane, while the z axis identifies the longitudinal direction. The azimuthal angle ϕ is computed with respect to the x axis. The polar angle θ 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 fix units such that $c = \hbar = 1$.

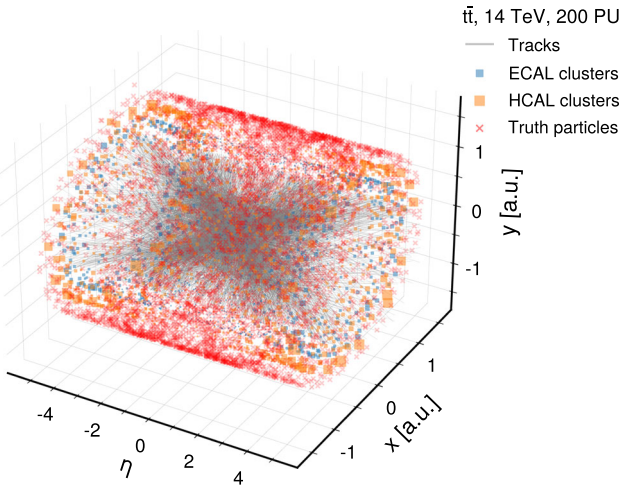


Fig. 1 A simulated $t\bar{t}$ event from the MLPF dataset with 200 PU interactions. The input tracks are shown in gray, with the trajectory curvature being defined by the inner and outer η, ϕ coordinates. Electromagnetic (hadron) calorimeter clusters are shown in blue (orange), with the size corresponding to cluster energy for visualization purposes. We also show the locations of the generator particles (all types) with red cross markers. The radii and thus the x, y-coordinates of the tracker, ECAL and HCAL surfaces are arbitrary for visualization purposes

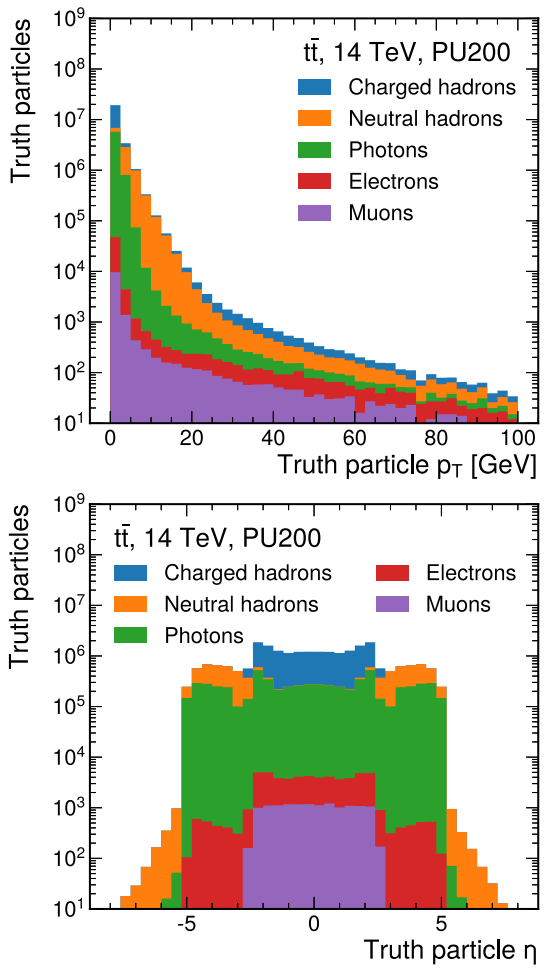


Fig. 2 The p_T (upper) and η (lower) distributions of the generator particles in the simulated $t\bar{t}$ dataset with PU, split by particle type

magnetic or hadron energy E and η, ϕ coordinates. In this simulation, an event has $N = (4.9 \pm 0.3) \times 10^3$ detector inputs on average.

The targets for PF reconstruction are stable generator-level particles that are associated to at least one detector element, as particles that leave no detector hits are generally not reconstructable. Generator particles are characterized by a particle identification (PID) which may take one of the following categorical values: charged hadron, neutral hadron, photon, electron, or muon. In case multiple generator particles all deposit their energy completely to a single calorimeter cluster, we treat them as reconstructable only in aggregate. In this case, the generator particles are merged by adding the momenta and assigning it the PID of the highest-energy subparticle. In addition, charged hadrons are indistinguishable outside the tracker acceptance from neutral hadrons, therefore we label generated charged hadrons with $|\eta| > 2.5$ to neutral hadrons. We also set a lower energy threshold on reconstructable neutral hadrons to $E > 9.0\,\text{Ge V}$ based on the DELPHES rule-based PF reconstruction, ignoring neutral hadrons that do not pass this threshold. A single event from the dataset is visualized in Fig. 1, demonstrating the input multiplicity and particle distribution in the event. The differential distributions of the generator-level particles in the simulated dataset are shown in Fig. 2.

We also store the PF candidates reconstructed by DELPHES for comparison purposes. The DELPHES rule-based PF algorithm is described in detail in Ref. [44]. Charged and neutral hadrons are identified based on track and hadron calorimeter cluster overlaps and energy subtraction. Photons are identified based on electromagnetic calorimeter clusters not matched to tracks. In addition, we note that electrons and muons are identified by DELPHES based on the generator particle associated to the corresponding track, therefore, for electron and muon tracks we add the corresponding generator-level identification as an input feature to the MLPF training to demonstrate that given the appropriate detector inputs, these less common particles can also be identified by the algorithm.

Each event is now fully characterized by the set of generator particles $Y = \{y_j\}$ (target vectors), the set of detector inputs $X = \{x_i\}$ (input vectors), with

$$y_j = [\text{PID}, p_\text{T}, E, \eta, \phi, q]\,, \tag{1}$$

$$x_i = [\text{type}, p_\text{T}, E_\text{ECAL}, E_\text{HCAL}, \eta, \phi, \eta_\text{outer}, \phi_\text{outer}, q]\,, \tag{2}$$

$$\text{PID} \in \{\text{charged hadron}, \text{neutral hadron}, \gamma, \text{e}^\pm, \mu^\pm\} \tag{3}$$

$$\text{type} \in \{\text{track}, \text{cluster}\}\,. \tag{4}$$

For input tracks, only the type, p_T, η, ϕ, η_{outer}, ϕ_{outer}, and q features are filled. Similarly, for input clusters, only the type, E_{ECAL}, E_{HCAL}, η and ϕ entries are filled. Unfilled features for both tracks and clusters are set to zero. In future iterations of MLPF, it may be beneficial to represent input elements of different types with separate data matrices to improve the computational efficiency of the model. Precomputing additional features such as track trajectory intersection points with the calorimeters may further improve the performance of PF reconstruction based on machine learning.

Functionally, the detector is modelled in simulation by a function $S(Y) = X$ that produces a set of detector signals from the generator-level inputs for an event. Reconstruction imperfectly approximates the inverse of that function $R \simeq S^{-1}(X) = Y$. In the following section, we approximate the reconstruction as set-to-set translation and implement a baseline MLPF reconstruction using GNNs.

3 ML-based PF reconstruction

For a given set of detector inputs X, we want to predict a set of particle candidates Y' that closely approximates the target generator particle set Y. The target and predicted sets may have a different number of elements, depending on the quality of the prediction. For use in ML using gradient descent, this requires a computationally efficient, differentiable set-to-set metric $||Y - Y'|| \in \mathbb{R}$ to be used as the loss function.

We simplify the problem numerically by first zero-padding the target set Y such that $|Y| = |X|$. This turns the problem of predicting a variable number of particles into a multi-classification prediction by adding an additional "no particle" to the classes already defined by the target PID and is based on Ref. [39]. Furthermore, for PF reconstruction, the target generator particles are often geometrically and energetically close to well-identifiable detector inputs. In physics terms, a charged hadron is reconstructed based on a track, while a neutral hadron candidate can always be associated to at least one primary source cluster, with additional corrections taken from other nearby detector inputs. Therefore, we choose to preprocess the inputs such that for a given arbitrary ordering of the detector inputs $X = [\dots, x_i, \dots]$ (sets of vectors are represented as matrices with some arbitrary ordering for ML training), the target set Y is arranged such that if a target particle can be associated to a detector input, it is arranged to be in the same location in the sequence. This data preprocessing step speeds up model convergence, but does not introduce any additional assumptions to the model. Since the target set now has a predefined size, we may compute the loss function which approximates reconstruction quality element-by-element:

$$||Y - Y'|| \equiv \sum_{j \in \text{event}} L(y_j, y'_j)\,, \tag{5}$$

$$L(y_j, y'_j) \equiv \text{CLS}(c_j, c'_j) + \alpha \text{REG}(p_j, p'_j)\,, \tag{6}$$

where the target values and predictions $y_j = [c_j; p_j]$ are decomposed such that the multi-classification is encapsulated in the scores and one-hot encoded classes c_j, while the momentum and charge regression values in p_j. We use CLS to denote the multi-classification loss, while REG denotes the regression loss for the momentum components weighted appropriately by a coefficient α. This combined per-particle loss function serves as a baseline optimization target for the ML training. Further physics improvements may be reached by extending the loss to take into account event-level quantities, either by using an energy flow distance as proposed in Refs. [47–49], or using a particle-based [50–53] generative adversarial network (GAN) [54] to optimize the reconstruction network in tandem with an adversarial classifier that is trained to distinguish between the target and reconstructed events, given the detector inputs.

3.1 Graph neural network implementation

Given the set of detector inputs for the event $X = \{x_i\}$, we adopt a message passing approach for reconstructing the PF candidates $Y = \{y_j\}$. First, we need to construct a trainable graph adjacency matrix $\mathcal{F}(X|w) = A$ for the given set of input elements, represented with the graph building block in Fig. 3. The input set is heterogeneous, containing elements of different type (tracks, ECAL clusters, HCAL clusters) in different feature spaces. Therefore, defining a static neighborhood graph in the feature space in advance is not straightforward. A generic approach to learnable graph construction using kNN in an embedding space, known as GravNet, has been proposed in Ref. [37], where the authors demonstrated that a learnable, dynamically-generated graph structure significantly improves the physics performance of an ML-based reconstruction algorithm for calorimeter clustering. Similar dynamic graph approaches have also been proposed in Ref. [23].

However, naive kNN graph implementations in common ML packages such as TensorFlow or PyTorch-Geometric have $\mathcal{O}(n^2)$ time complexity: for each set element out of $n = |X|$, we must order the other $n - 1$ elements by distance and pick the k closest. More efficient kNN graph construction is possible with, for example, k-dimensional trees [55], but so far, we are not aware of an implementation that interfaces with common, differentiable ML tools. For reconstruction, given equivalent physics performance, both computational efficiency (a low overall runtime) and

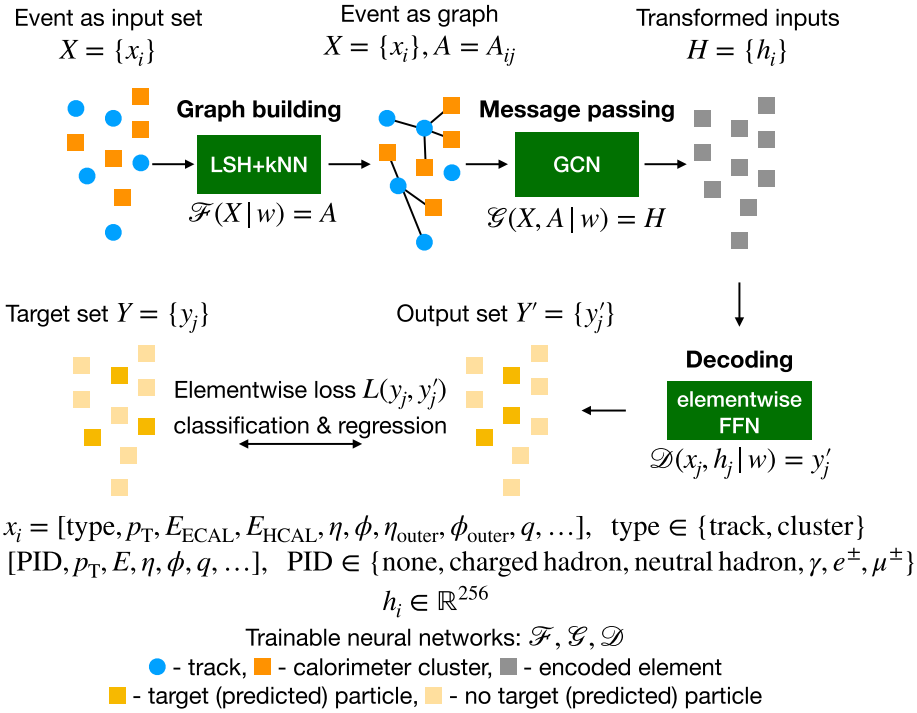


Fig. 3 Functional overview of the end-to-end trainable MLPF setup with GNNs. The event is represented as a set of detector elements x_i. The set is transformed into a graph by the graph building step, which is implemented here using an locality sensitive hashing (LSH) approximation of kNN. The graph nodes are then encoded using a message passing step, implemented using graph convolutional nets. The encoded elements are decoded to the output feature vectors y_j using elementwise feedforward networks

scalability (subquadratic time and memory scaling with the input size) are desirable.

We build on the GravNet approach [37] by using an approximate kNN graph construction algorithm based on locality sensitive hashing (LSH) to improve the time complexity of the graph building algorithm. The LSH approach has been recently proposed [56] for approximating and thus speeding up ML models that take into account element-to-element relations using an optimizable $n \times n$ matrix known as self-attention [57]. The method divides the input into bins using a hash function, such that nearby elements are likely to be assigned to the same bin. The bins contain only a small number of elements, such that constructing a kNN graph in the bin is significantly faster than for the full set of elements, and thus not strongly affected by the quadratic scaling of the kNN algorithm.

In the kNN+LSH approach, the n input elements x_i are projected into a d_K-dimensional embedding space by a trainable, elementwise feed-forward network $\text{FFN}(x_i|w) = z_i \in \mathbb{R}^{d_K}$. As in Ref. [56], we now assign each element into one of d_B bins indexed by integers b_i using $h(z_i) = b_i \in [1, \ldots, d_B]$, where $h(x)$ is a hash function that assigns nearby x to the same bin with a high probability. We define the hash function as $h(x) = arg\,max[xP; -xP]$ where $[u; v]$ denotes the concatenation of two vectors u and v and P is a random projection matrix of size $[d_K, d_B/2]$ drawn from the normal distribution at initialization.

We now build d_B kNN graphs based on the embedded elements z_i in each of the LSH bins, such that the full sparse graph adjacency A_{ij} in the inputs set X is defined by the sum of the subgraphs. The embedding function can be optimized with backpropagation and gradient descent using the values of the nonzero elements of A_{ij}. Overall, this graph building approach has $\mathcal{O}(n \log n)$ time complexity and does not require the allocation of an n^2 matrix at any point. The LSH step generates d_B disjoint subgraphs in the full event graph. This is motivated by physics, as we expect subregions of the detector to be reconstructable approximately independently. The existing PF algorithm in the CMS detector employs a similar approach by producing disjoint PF blocks as an intermediate step of the algorithm [11].

Having built the graph dynamically, we now use a variant of message passing [20,22,58,59] to create hidden encoded states $\mathcal{G}(x_i, A_{ij}|w) = h_i$ of the input elements taking into account the graph structure. As a first baseline, we use a variant of graph convolutional network (GCN) that combines local and global node-level information [60–62]. This choice is motivated by implementation and evaluation efficiency in establishing a baseline. This message passing step is represented in Fig. 3 by the GCN block. Finally, we decode the encoded nodes $H = \{h_i\}$ to the target outputs with an elementwise feed-forward network that combines the hidden state with the original input element $\mathcal{D}(x_i, h_i|w) = y'_i$ using a skip connection.

We have a joint graph building, but separate graph convolution and decoding layers for the multi-classification and the momentum and charge regression subtasks. This allows each subtask to be retrained separately in addition to a combined end-to-end training should the need arise. The classification and regression losses are combined with constant empirical weights such that they have an approximately equal contribution to the full training loss. We use categorical cross-

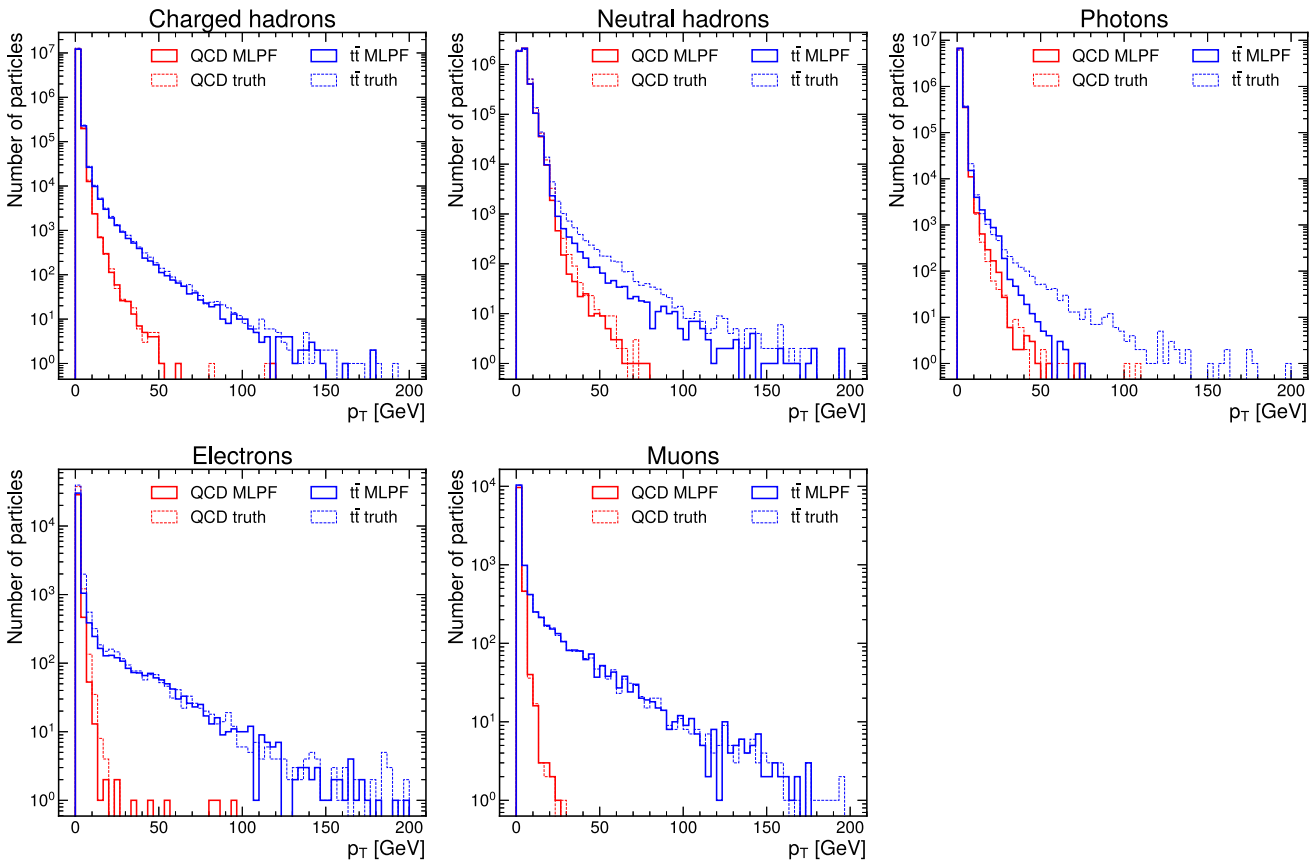


Fig. 4 The MLPF reconstruction compared to the truth-level p_T distribution for the QCD validation sample and the $t\bar{t}$ sample used for training. The differences between the MLPF and truth distributions are a measure of the prediction error. Charged hadrons, electrons, and muons are identified based on tracks with no misidentification or loss of efficiency, hence the prediction error is negligible for both samples. For neutral hadrons and photons, the tail is reconstructed at a lower efficiency for $t\bar{t}$ as compared to QCD, which could arise from overrepresentation of low-p_T particles in the unweighted $t\bar{t}$ training sample

entropy for the classification loss, which measures the similarity between the true label distribution c_j and the predicted labels c'_j. For the regression loss, we use componentwise mean-squared error between the true and predicted momenta, where the losses for the individual momentum components $(p_T, \eta, \sin\phi, \cos\phi, E)$ are scaled by normalization factors such that the components have approximately equal contributions to the total loss. It may be beneficial to use specific multi-task training strategies such as gradient surgery [63] to further improve the performance across all subtasks and to reduce the reliance on ad-hoc scale factors between the losses in a multi-task setup.

The multi-classification prediction outputs for each node are converted to particle probabilities with the softmax operation. We choose the PID with the highest probability for the reconstructed particle candidate, while ensuring that the probability meets a threshold that matches a fake rate working point defined by the baseline DELPHESPF reconstruction algorithm.

The predicted graph structure is an intermediate step in the model and is not used in the loss function explicitly – we only optimize the model with respect to reconstruction quality. However, using the graph structure in the loss function when a known ground truth is available may further improve the optimization process. In addition, access to the predicted graph structure may be helpful in evaluating the interpretability of the model.

The set of networks for graph building, message passing and decoding has been implemented with TENSORFLOW 2.3 and can be trained end-to-end using gradient descent. The inputs are zero-padded to $n = 6400$ elements. Additional elements beyond 6400 are truncated for efficient training and performance evaluation, amounting to about 0.007% of the total number of elements in the $t\bar{t}$ simulation sample. The truncated elements are always calorimeter towers as the order of the elements is set by the DELPHES simulation. For inference during data taking, truncation should be avoided. The LSH bin size chosen to be 128 such that the number of bins $d_B = 50$ and the number of nearest neighbors $k = 16$. We

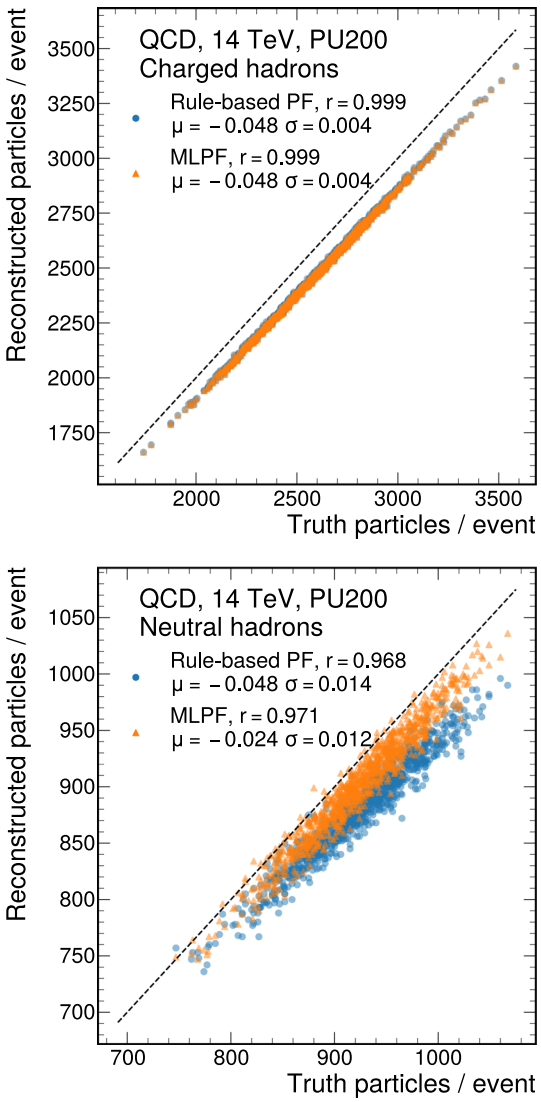


Fig. 5 True and predicted particle multiplicity for MLPF and DELPHESPF for charged (upper) and neutral hadrons (lower) in simulated QCD multijet events with PU. Both models show a high degree of correlation (r) between the generated and predicted particle multiplicity, with the MLPF model reconstructing the neutral particle multiplicities with improved resolution (σ) and a lower bias (μ)

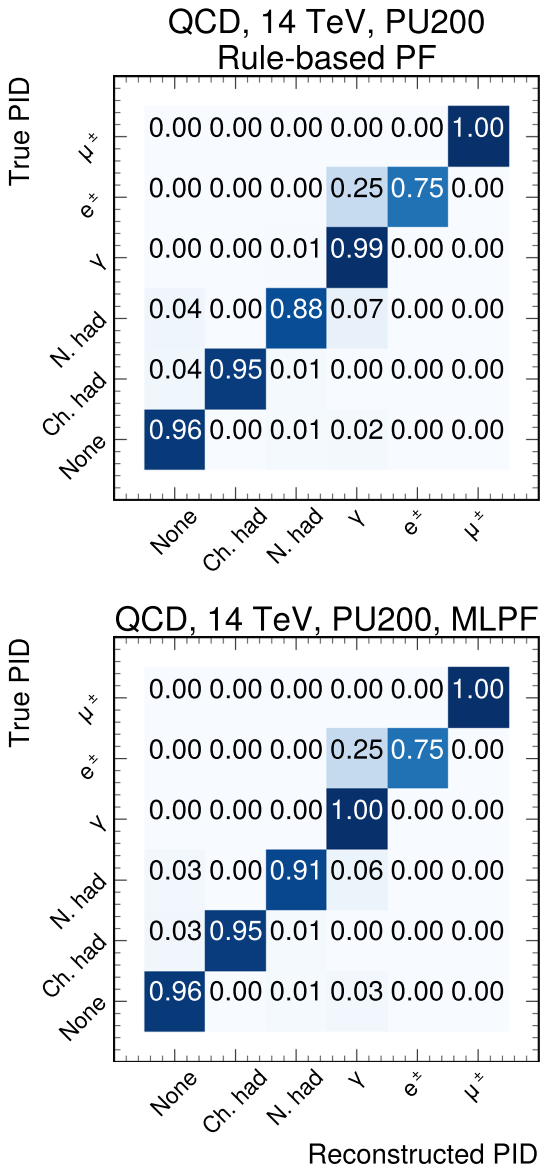


Fig. 6 Particle identification confusion matrices in simulated QCD multijet events with PU, with gen-level particles as the ground truth, showing the baseline rule-based DELPHESPF (upper) and the MLPF (lower) outputs. The rows have been normalized to unit probability, corresponding to normalizing the dataset according to the generated PID

use two hidden layers for each encoding and decoding net with 256 units each, with two successive graph convolutions between the encoding and decoding steps. Exponential linear activations (ELU) [64] are used for the hidden layers and linear activations are used for the outputs. Overall, the model has approximately 1.5 million trainable weights and 25,000 constant weights for the random projections. For optimization, we use the Adam [65] algorithm with a learning rate of 5×10^{-6} for 300 epochs, training over 4×10^4 events, with 10^4 events used for testing. The events are processed in mini-batches of five simultaneous events per graphics processing unit (GPU), we train for approximately 48 h using five RTX 2070S GPUs using data parallelism on 40,000 simulated $t\bar{t}$ events. We report the results of the multi-task learning problem in the next section. The code and dataset to reproduce the training are made available on the Zenodo platform [46,66].

4 Results

In Fig. 4, we show the p_T distributions for the MLPF reconstruction and generator-level truth for both simulated QCD multijet and $t\bar{t}$ events. Although the MLPF model was trained

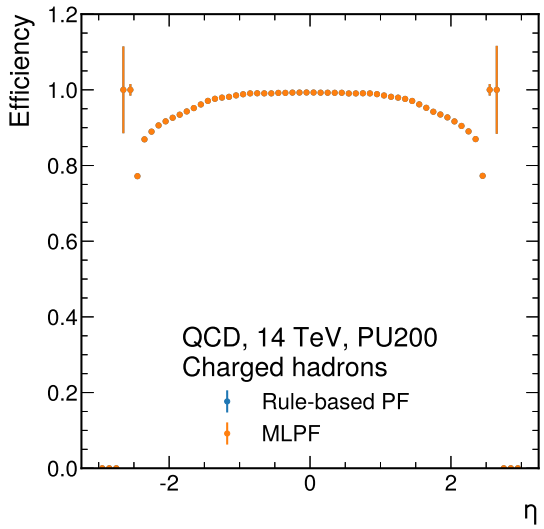


Fig. 7 The efficiency of reconstructing charged hadron candidates as a function of the generator particle pseudorapidity η in simulated QCD multijet events with PU. Since the simulation does not contain fake tracks, the charged hadron reconstruction is driven entirely by tracking efficiency and is the same for MLPF and the rule-based PF

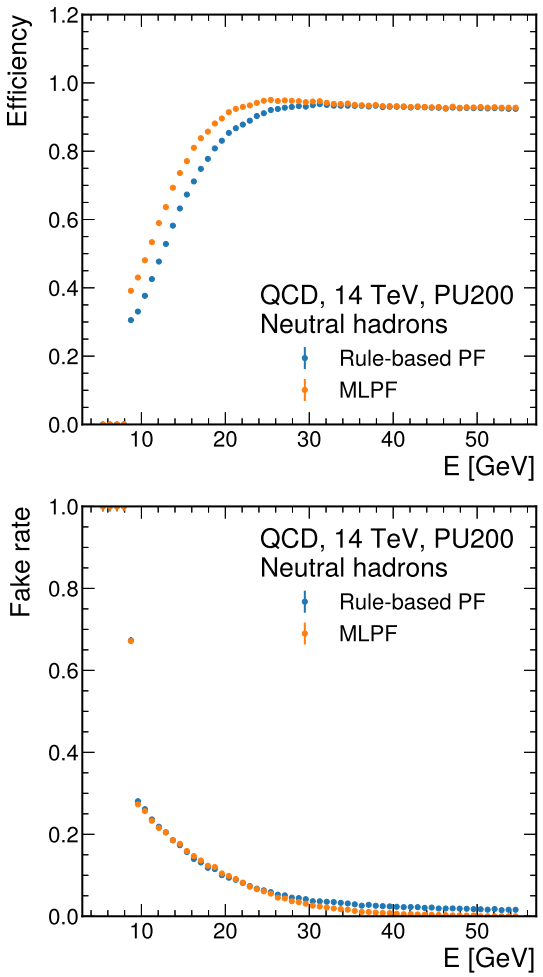


Fig. 8 The efficiency (upper) and fake rate (lower) of reconstructing neutral hadron candidates as a function of the generator particle energy in simulated QCD multijet events with PU. The MLPF model shows comparable performance to the DELPHESPF benchmark, with a somewhat lower fake rate at a similar efficiency

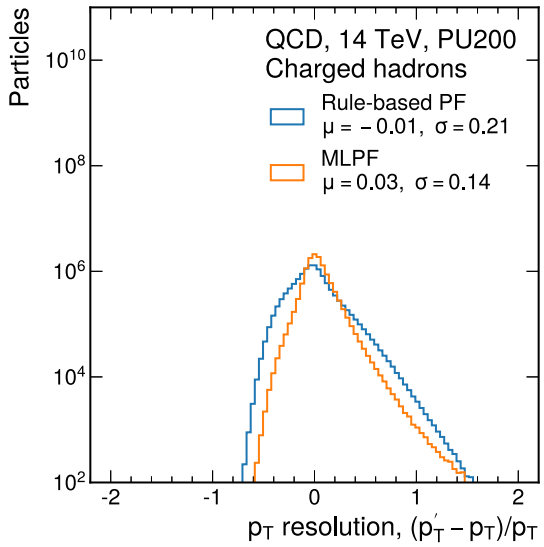
on $t\bar{t}$, we observe a slight underprediction at high transverse momentum for photons and neutral hadrons, which could arise from the much greater numbers of low-p_T particles relative to high-p_T particles in this unweighted sample. Further work is needed to improve the performance in the high-p_T tail of the distribution. We find that the model generalizes well to the QCD sample that was not used in the training, demonstrating that the MLPF-based reconstruction is transferable across different physics samples.

For the following results, we focus on the charged and neutral hadron performance in QCD events, as hadrons make up the bulk of the energy content of the jets and thus are the primary target for PF reconstruction. We do not report detailed performance characteristics for photons, electrons, and muons at this time because of the limitations of the DELPHES dataset and the rule-based PF algorithm. A realistic study of photon and electron disambiguation, in particular, requires a more detailed dataset that includes additional physics effects, as discussed in Sect. 2. In Fig. 5, we present the charged and neutral hadron multiplicities from both the baseline rule-based PF and MLPF algorithms as a function of the target multiplicities. The particle multiplicities from the MLPF model correlate better with the generator-level target than the rule-based PF algorithm, demonstrating that the multi-classification model successfully reconstructs variable-multiplicity events. In general, we do not observe significant differences in the physics performance of the MLPF algorithm between the QCD and $t\bar{t}$ samples in the phase space where we have validated it.

In Fig. 6, we compare the per-particle multi-classification confusion matrix for both reconstruction methods. We see overall a similar classification performance for both approaches. The charged hadron identification performance is driven by track efficiency and is the same for MLPF and the rule-based PF. The neutral hadron identification efficiency is slightly higher for MLPF (0.91 vs 0.88), since hadron calorimeter cluster energies that are not matched to tracks must be determined algorithmically for neutral hadron reconstruction. The electron–photon misidentification is driven by the parametrized tracking efficiency, as electromagnetic calorimeter clusters without an associated track are reconstructed as photons. Electron and muon identification performance is shown simply for completeness, as it is driven by the use of generator-level PID values for those tracks. Improved Monte Carlo generation, subsampling, or weighting may further improve reconstruction performance for particles or kinematic configurations that occur rarely in a phys-

Table 1 Particle reconstruction efficiency and fake rate, multiplicity N, p_T (E) and η resolutions for charged (neutral) hadrons, comparing the rule-based PF baseline and the proposed MLPF method. Bolded values indicate better performance

Metric	Charged hadrons		Neutral hadrons	
	Rule-based PF	MLPF	Rule-based PF	MLPF
Efficiency	0.953	0.953	0.883	**0.908**
Fake rate	0.000	0.000	0.071	**0.068**
p_T (E) resolution	0.213	**0.137**	0.350	**0.323**
η resolution	**0.240**	0.245	**0.050**	0.058
N resolution	0.004	0.004	0.014	**0.013**



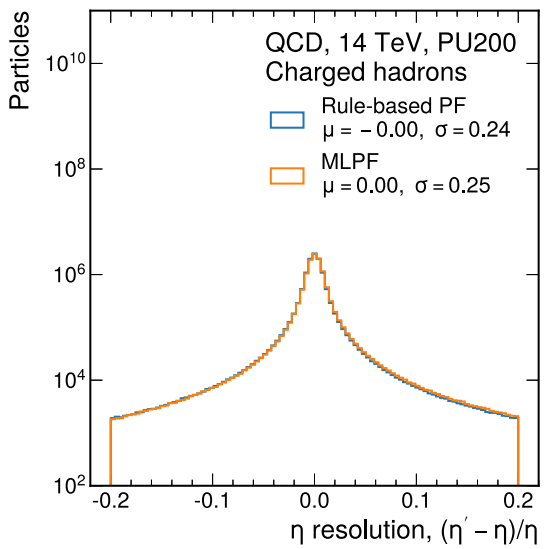
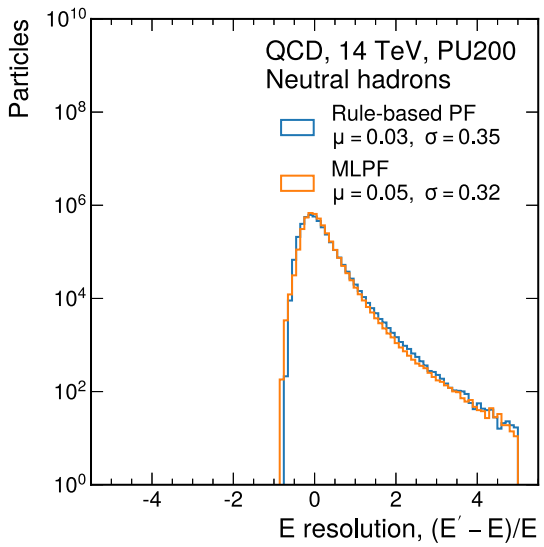


Fig. 9 The p_T and η resolution of the DELPHESPF benchmark and the MLPF model for charged hadrons in simulated QCD multijet events with PU. The p_T resolution is comparable for both algorithms, with the angular resolution being driven by the smearing of the track (η, ϕ) coordinates



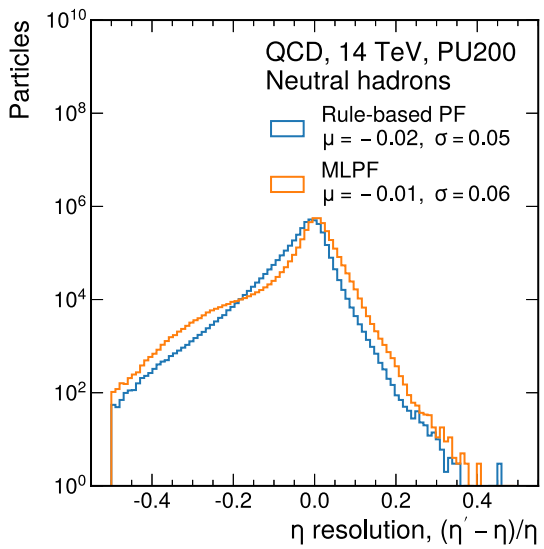


Fig. 10 The energy and η resolution of the DELPHESPF benchmark and the MLPF model for neutral hadrons in simulated QCD multijet events with PU. Both reconstruction algorithms show comparable performance

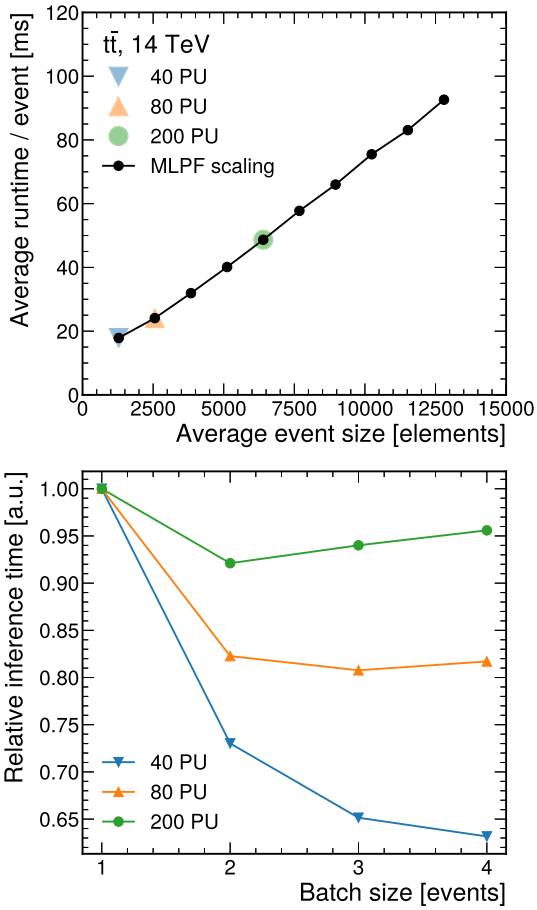


Fig. 11 Average runtime of the MLPF GNN model with a varying input event size (upper) and the relative inference time when varying the number of events evaluated simultaneously, i.e. batch size (lower), normalized to batch size 1. For a simulated event equivalent to 200 PU collisions, we see a runtime of around 50 ms, which scales approximately linearly with respect to the input event size. We see a weak dependence on batch size, with batching having a minor positive effect for low-pileup events. The runtime for each event size is averaged over 100 randomly generated events over three independent runs. The timing tests were done using an Nvidia RTX 2060S GPU and an Intel i7-10700@2.9GHz CPU. We assume a linear scaling between PU and the number of detector elements

ical simulation. In this set of results, we apply no weighting on the events or particles in the event.

In Fig. 7, we see that the η-dependent charged hadron efficiency (true positive rate) for the MLPF model is somewhat higher than for the rule-based PF baseline, while the fake rate (false positive rate) is equivalently zero, as the DELPHES simulation includes no fake tracks. From Fig. 8, we observe a similar result for the energy-dependent efficiency and fake rate of neutral hadrons. Both algorithms exhibit a turn-on at low energies and show a constant behaviour at high energies, with MLPF being comparable or slightly better than the rule-based PF baseline.

Furthermore, we see on Figs. 9 and 10 that the energy, energy (p_T) and angular resolution of the MLPF algorithm are generally comparable to the baseline for neutral (charged) hadrons.

Overall, these results demonstrate that formulating PF reconstruction as a multi-task ML problem of simultaneously identifying charged and neutral hadrons in a high-PU environment and predicting their momentum may offer comparable or improved physics performance over hand-written algorithms in the presence of sufficient simulation samples and careful optimization. The performance characteristics for the baseline and the proposed MLPF model are summarized in Table 1.

We also characterize the computational performance of the GNN-based MLPF algorithm. In Fig. 11, we see that the average inference time scales roughly linearly with the input size, which is necessary for scalable reconstruction at high PU. We also note that the GNN-based MLPF algorithm runs natively on a GPU, with the current runtime at around 50 ms/event on a consumer-grade GPU for a full 200 PU event. The algorithm is simple to port to computing architectures that support common ML frameworks like TENSORFLOW without significant investment. This includes GPUs and potentially even field-programmable gate arrays (FPGAs) or ML-specific processors such as the GraphCore intelligence processing units (IPUs) [67] through specialized ML compilers [68–70]. These coprocessing accelerators can be integrated into existing CPU-based experimental software frameworks as a scalable service that grows to meet the transient demand [71–73].

5 Discussion and outlook

We have developed a ML algorithm for PF reconstruction in a high-pileup environment for a general-purpose multilayered particle detector based on transforming input sets of detector elements to the output set of reconstructed particles. The MLPF implementation with GNNs is based on graph building with a LSH approximation for kNN, dubbed LSH+kNN, and message passing using graph convolutions. Based on benchmark particle-level tt̄ and QCD multijet datasets generated using PYTHIA 8 and DELPHES 3, the MLPF GNN reconstruction offers comparable performance to the baseline rule-based PF algorithm in DELPHES, demonstrating that a purely parametric ML-based PF reconstruction can reach or exceed the physics performance of existing reconstruction algorithms, while allowing for greater portability across various computing architectures at a possibly reduced cost. The inference time empirically scales approximately linearly with the input size, which is useful for efficient evaluation in the high-luminosity phase of the LHC. In addition, the ML-based reconstruction model may offer useful features for downstream physics analysis like per-particle probabilities for different reconstruction interpretations, uncertainty

estimates, and optimizable particle-level reconstruction for rare processes including displaced signatures.

The MLPF model can be further improved with a more physics-motivated optimization criterion, i.e. a loss function that takes into account event-level, in addition to particle-level differences. While we have shown that a per-particle loss function already converges to an adequate physics performance overall, improved event-based losses such as the object condensation approach or energy flow may be useful. In addition, an event-based loss may be defined using an adversarial classifier that is trained to distinguish the target particles from the reconstructed particles.

Reconstruction algorithms need to adapt to changing experimental conditions – this may be addressed in MLPF by a periodic retraining on simulation that includes up-to-date running condition data such as the beam-spot location, dead channels, and latest calibrations. In a realistic MLPF training, care must be taken that the reconstruction qualities of rare particles and particles in the low-probability tails of distributions are not adversely affected and that the reconstruction performance remains uniform. This may be addressed with detailed simulations and weighting schemes. In addition, for a reliable physics result, the interpretability of the reconstruction is essential. The reconstructed graph structure can provide information about causal relations between the input detector elements and the reconstructed particle candidates.

In order to develop a usable ML-based PF reconstruction algorithm, a realistic high-pileup simulated dataset that includes detailed interactions with the detector material needs to be used for the ML model optimization. The model should be optimized and validated on a mix of realistic high-PU events to learn global properties of reconstruction, as well as on a set of particle gun samples to ensure that local properties of particle reconstruction are learned in a generalizable way. To evaluate the reconstruction performance, efficiencies, fake rates, and resolutions for all particle types need to be studied in detail as a function of particle kinematics and detector conditions. Furthermore, high-level derived quantities such as pileup-dependent jet and missing transverse momentum resolutions must be assessed for a more complete characterization of the reconstruction performance. With ongoing work in ML-based track and calorimeter cluster reconstruction upstream of PF [26,29,52,74–76] and ML-based reconstruction of high-level objects including jets and jet classification probabilities downstream of PF [33–35,77–81], care must be taken that the various steps are optimized and interfaced coherently.

Finally, the MLPF algorithm is inherently parallelizable and can take advantage of hardware acceleration of GNNs via graphics processing units (GPUs), field-programmable gate arrays (FPGAs) or emerging ML-specific processors. Current experimental software frameworks can easily integrate coprocessing accelerators as a scalable service. By harnessing heterogeneous computing and parallelizable, efficient ML, the burgeoning computing demand for event reconstruction tasks in the high-luminosity LHC era can be met while maintaining or even surpassing the current physics performance.

Acknowledgements We would like to thank Guenther Dissertori for suggesting the idea of ML-driven PF reconstruction several years ago in private discussions. We thank our colleagues in the CMS Collaboration, especially in the Particle Flow, Physics Performance and Datasets, Offline and Computing, and Machine Learning groups, in particular Josh Bendavid, Kenichi Hatakeyama, Lindsey Gray, Jan Kieseler, Danilo Piparo, Gregor Kasieczka, Laurits Tani, and Juska Pekkanen, for helpful feedback in the course of this work.
J. P. was supported by the Prime National Science Foundation (NSF) Tier2 award 1624356 and the U.S. Department of Energy (DOE), Office of Science, Office of High Energy Physics under Award No. DE-SC0011925 while at Caltech, and is currently supported by the Mobilitas Pluss Grant no. MOBTP187 of the Estonian Research Council. J. D. is supported by the DOE, Office of Science, Office of High Energy Physics Early Career Research program under Award No. DE-SC0021187 and by the DOE, Office of Advanced Scientific Computing Research under Award No. DE-SC0021396 (FAIR4HEP). M. P. is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 772369). J-R. V. and M. S. are supported by the DOE, Office of Science, Office of High Energy Physics under Award No. DE-SC0011925, DE-SC0019227, and DE-AC02-07CH11359. J-R. V. was additionally partially supported the same ERC Grant as M. P.
We are grateful to Caltech and the Kavli Foundation for their support of undergraduate student research in cross-cutting areas of machine learning and domain sciences. This work was conducted at "*iBanks*," the AI GPU cluster at Caltech, and on the NICPB GPU resources, supported by European Regional Development Fund through the CoE program Grant TK133. We acknowledge Nvidia, SuperMicro and the Kavli Foundation for their support of *iBanks*. Part of this work was also performed using the Pacific Research Platform Nautilus HyperCluster supported by NSF awards CNS-1730158, ACI-1540112, ACI-1541349, OAC-1826967, the University of California Office of the President, and the University of California San Diego's California Institute for Telecommunications and Information Technology/Qualcomm Institute. Thanks to CENIC for the 100 Gpbs networks.

Data Availability Statement This manuscript has associated data in a data repository. [Authors' comment: The simulated datasets used in this paper are available at https://doi.org/10.5281/zenodo.4559324 and at https://zenodo.org/communities/mpp-hep/.]

Funded by SCOAP[3].

References

1. CELLO Collaboration, An analysis of the charged and neutral energy flow in e^+e^- hadronic annihilation at 34 GeV, and a determination of the QCD effective coupling constant. Phys. Lett. B **113**, 427 (1982). https://doi.org/10.1016/0370-2693(82)90778-X
2. ALEPH Collaboration, Performance of the ALEPH detector at LEP. Nucl. Instrum. Methods A **360**, 481 (1995). https://doi.org/10.1016/0168-9002(95)00138-7
3. H1 Collaboration, Measurement of charged particle multiplicity distributions in DIS at HERA and its implication to entanglement entropy of partons. arXiv:2011.01812
4. ZEUS Collaboration, Measurement of the diffractive structure function F2(D(4)) at HERA. Eur. Phys. J. C **1**, 81–96 (1998). https://doi.org/10.1007/s100520050063. arXiv:hep-ex/9709021
5. ZEUS Collaboration, Measurement of the diffractive cross-section in deep inelastic scattering using ZEUS, data. Eur. Phys. J. C **6**(1999), 43–66 (1994). https://doi.org/10.1007/PL00021606. arXiv:hep-ex/9807010
6. DELPHI Collaboration, Performance of the DELPHI detector. Nucl. Instrum. Methods A **378**, 57 (1996). https://doi.org/10.1016/0168-9002(96)00463-9
7. A. Bocci, S. Lami, S. Kuhlmann, G. Latino, Study of jet energy resolution at CDF. Int. J. Mod. Phys. A **16S1A**, 255 (2001). https://doi.org/10.1142/S0217751X01006632
8. A.L. Connolly, A Search for Supersymmetric Higgs Bosons in the Di-tau Decay Mode in $p\bar{p}$ Collisions at 1.8 TeV. Ph.D. thesis, UC Berkeley, 2003. https://doi.org/10.2172/15017134
9. CDF Collaboration, Measurement of $\sigma(p\bar{p} \to Z)$.Br$(Z \to 2\tau)$ in $p\bar{p}$ collisions at $\sqrt{s} = 1.96$ TeV. Phys. Rev. D **75**, 092004 (2007). https://doi.org/10.1103/PhysRevD.75.092004
10. D0 Collaboration, Measurement of $\sigma(p\bar{p} \to Z + X)$ Br$(Z \to \tau^+\tau^-)$ at $\sqrt{s} = 1.96$ TeV. Phys. Lett. B **670**, 292 (2009). https://doi.org/10.1016/j.physletb.2008.11.010. arXiv:0808.1306
11. CMS Collaboration, Particle-flow reconstruction and global event description with the CMS detector. JINST **12**, P10003 (2017). https://doi.org/10.1088/1748-0221/12/10/P10003. arXiv:1706.04965
12. ATLAS Collaboration, Jet reconstruction and performance using particle flow with the ATLAS Detector. Eur. Phys. J. C **77**, 466 (2017). https://doi.org/10.1140/epjc/s10052-017-5031-2. arXiv:1703.10485
13. CMS Collaboration Collaboration, "Challenges of particle flow reconstruction in the CMS High-Granularity Calorimeter at the High-Luminosity LHC", Technical Report CMS-CR-2016-151. 1, CERN, Geneva, Jul, 2016. https://doi.org/10.1088/1742-6596/928/1/012027
14. FCC-hh Collaboration, Physics requirements for the FCC-hh calorimeter system. J. Phys. Conf. Ser. **1162**, 012010 (2019). https://doi.org/10.1088/1742-6596/1162/1/012010
15. FCC Collaboration, FCC-hh: the hadron collider. Eur. Phys. J. ST **228**, 755 (2019). https://doi.org/10.1140/epjst/e2019-900087-0
16. FCC Collaboration, FCC-ee: the lepton collider: future circular collider conceptual design report volume 2. Eur. Phys. J. ST **228**, 261 (2019). https://doi.org/10.1140/epjst/e2019-900045-4
17. T. Behnke et al., The International Linear Collider Technical Design Report—volume 1: Executive Summary. arXiv:1306.6327
18. CEPC Study Group Collaboration, CEPC Conceptual Design Report: Volume 2—Physics & Detector. arXiv:1811.10545
19. F. Scarselli et al., The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61 (2009). https://doi.org/10.1109/TNN.2008.2005605
20. J. Gilmer et al., "Neural message passing for quantum chemistry", in *Proceedings of the 34th International Conference on Machine Learning*, volume 70 ed. by D. Precup, Y.W. Teh (PMLR, 2017), p. 1263. arXiv:1704.01212
21. C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). https://doi.org/10.1109/CVPR.2017.16. arXiv:1612.00593
22. P.W. Battaglia et al., Interaction Networks for Learning about Objects, Relations and Physics, in *Advances in Neural Information Processing Systems*, volume 29 ed. by D. Lee et al (Curran Associates, Inc., 2016), p. 4502. arXiv:1612.00222
23. Y. Wang et al., Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. (2019). https://doi.org/10.1145/3326362. arXiv:1801.07829
24. J. Shlomi, P. Battaglia, J.-R. Vlimant, Graph neural networks in particle physics. Mach. Learn. Sci. Technol. **2**, 021001 (2021). https://doi.org/10.1088/2632-2153/abbf9a. arXiv:2007.13681
25. S. Farrell et al., Novel deep learning methods for track reconstruction, in *4th International Workshop Connecting the Dots* (2018). arXiv:1810.06111
26. X. Ju et al., Graph neural networks for particle reconstruction in high energy physics detectors, in *2nd Machine Learning and the Physical Sciences Workshop at the 33rd Conference on Neural Information Processing Systems* 3 (2020). arXiv:2003.11603
27. S. Amrouche et al., The tracking machine learning challenge : accuracy phase, in *The NeurIPS '18 Competition* (2020), p. 231. https://doi.org/10.1007/978-3-030-29135-8_9. arXiv:1904.06778
28. S. Amrouche et al., Similarity hashing for charged particle tracking, in *IEEE International Conference on Big Data 2019* (2019), p. 1595. https://doi.org/10.1109/BigData47090.2019.9006316
29. N. Choma et al., Track Seeding and Labelling with Embedded-space Graph Neural Networks, in *6th International Workshop Connecting the Dots* (2020). arXiv:2007.00149
30. X. Ju, B. Nachman, Supervised jet clustering with graph neural networks for Lorentz boosted bosons. Phys. Rev. D **102**, 075014 (2020). https://doi.org/10.1103/PhysRevD.102.075014. arXiv:2008.06064
31. J. Li, T. Li, F.-Z. Xu, Reconstructing boosted Higgs jets from event image segmentation. arXiv:2008.13529
32. J. Guo, J. Li, T. Li, The boosted Higgs jet reconstruction via graph neural network. arXiv:2010.05464
33. E.A. Moreno et al., JEDI-net: a jet identification algorithm based on interaction networks. Eur. Phys. J. C **80**, 58 (2020). https://doi.org/10.1140/epjc/s10052-020-7608-4. arXiv:1908.05318
34. E.A. Moreno et al., Interaction networks for the identification of boosted $H \to b\bar{b}$ decays. Phys. Rev. D **102**, 012010 (2020). https://doi.org/10.1103/PhysRevD.102.012010. arXiv:1909.12285
35. H. Qu, L. Gouskos, ParticleNet: jet tagging via particle clouds. Phys. Rev. D **101**, 056019 (2020). https://doi.org/10.1103/PhysRevD.101.056019. arXiv:1902.08570
36. V. Mikuni, F. Canelli, ABCNet: An attention-based method for particle tagging. Eur. Phys. J. Plus **135**(6), 463 (2020). https://doi.org/10.1140/epjp/s13360-020-00497-3. arXiv:2001.05311
37. S.R. Qasim, J. Kieseler, Y. Iiyama, M. Pierini, Learning representations of irregular particle-detector geometry with distance-weighted graph networks. Eur. Phys. J. C **79**, 608 (2019). https://doi.org/10.1140/epjc/s10052-019-7113-9. arXiv:1902.07987
38. J.A. Martínez et al., Pileup mitigation at the large hadron collider with graph neural networks. Eur. Phys. J. Plus **134**, 333 (2019). https://doi.org/10.1140/epjp/i2019-12710-3. arXiv:1810.07988
39. J. Kieseler, Object condensation: one-stage grid-free multi-object reconstruction in physics detectors, graph and image data. Eur. Phys. J. C **80**, 886 (2020). https://doi.org/10.1140/epjc/s10052-020-08461-2. arXiv:2002.03605
40. F.A. Di Bello et al., Towards a Computer Vision Particle Flow. arXiv:2003.08863

41. J. Duarte, J.-R. Vlimant, Graph neural networks for particle tracking and reconstruction, in *Artificial Intelligence for Particle Physics* (World Scientific Publishing, 2020). Submitted to Int. J. Mod. Phys. A. https://doi.org/10.1142/12200. arXiv:2012.01249
42. T. Sjöstrand, S. Mrenna, P.Z. Skands, PYTHIA 6.4 physics and manual. JHEP **05**, 026 (2006). https://doi.org/10.1088/1126-6708/2006/05/026. arXiv:hep-ph/0603175
43. T. Sjöstrand, S. Mrenna, P.Z. Skands, A brief introduction to PYTHIA8.1. Comput. Phys. Commun. **178**, 852 (2008). https://doi.org/10.1016/j.cpc.2008.01.036. arXiv:0710.3820
44. DELPHES 3 Collaboration, DELPHES3, a modular framework for fast simulation of a generic collider experiment. JHEP **02**, 057 (2014). https://doi.org/10.1007/JHEP02(2014)057. arXiv:1307.6346
45. S. Chekanov, HepSim: a repository with predictions for high-energy physics experiments. Adv. High Energy Phys. **2015**, 136093 (2015). https://doi.org/10.1155/2015/136093. arXiv:1403.1886
46. J. Pata et al., Simulated particle-level events of $t\bar{t}$ and QCD with PU200 using PYTHIA8+DELPHES3 for machine learned particle flow (MLPF) (2021). https://doi.org/10.5281/zenodo.4559324
47. P.T. Komiske, E.M. Metodiev, J. Thaler, Energy flow networks: deep sets for particle jets. JHEP **01**, 121 (2019). https://doi.org/10.1007/JHEP01(2019)121. arXiv:1810.05165
48. P.T. Komiske, E.M. Metodiev, J. Thaler, Metric space of collider events. Phys. Rev. Lett. **123**, 041801 (2019). https://doi.org/10.1103/PhysRevLett.123.041801. arXiv:1902.02346
49. M.C. Romao et al., Use of a generalized energy mover's distance in the search for rare phenomena at colliders. arXiv:2004.09360
50. R. Kansal et al., Graph Generative Adversarial Networks for Sparse Data Generation in High Energy Physics, in *3rd Machine Learning and the Physical Sciences Workshop at the 34th Conference on Neural Information Processing Systems* (2020). arXiv:2012.00173
51. M. Bellagente et al., How to GAN away detector effects. Sci. Post Phys. **8**, 070 (2020). https://doi.org/10.21468/SciPostPhys.8.4.070. arXiv:1912.00477
52. D. Belayneh et al., Calorimetry with deep learning: particle simulation and reconstruction for collider physics. Eur. Phys. J. C **80**, 688 (2020). https://doi.org/10.1140/epjc/s10052-020-8251-9. arXiv:1912.06794
53. A. Butter, T. Plehn, R. Winterhalder, How to GAN LHC events. Sci. Post Phys. **7**, 075 (2019). https://doi.org/10.21468/SciPostPhys.7.6.075. arXiv:1907.03764
54. I.J. Goodfellow et al., Generative adversarial nets, in *Advances in Neural Information Processing Systems*, volume 27 ed. by Z. Ghahramani et al. (Curran Associates, Inc., 2014). arXiv:1406.2661
55. N. Rajani, K. McArdle, I.S. Dhillon, Parallel k nearest neighbor graph construction using tree-based data structures, in *1st High Performance Graph Mining workshop*, volume 1 (2015), p. 3–11
56. N. Kitaev, Ł. Kaiser, A. Levskaya, Reformer: the efficient transformer, in *8th International Conference on Learning Representations* (2020). arXiv:2001.04451
57. A. Vaswani et al., Attention is all you need, in *Advances in Neural Information Processing Systems*, volume 30 ed. by I. Guyon et al. (Curran Associates, Inc., 2017), p. 5998. arXiv:1706.03762
58. F. Scarselli et al., The graph neural network model. IEEE Trans. Neural Netw. **20**, 61 (2009). https://doi.org/10.1109/TNN.2008.2005605
59. P.W. Battaglia et al., Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261
60. T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in *5th International Conference on Learning Representations*. 2017. arXiv:1609.02907
61. F. Wu et al., Simplifying graph convolutional networks, in *Proceedings of the 36th International Conference on Machine Learning*, volume 97 ed. by K. Chaudhuri, R. Salakhutdinov (PMLR, 2019), p. 6861. arXiv:1902.07153
62. X. Xin, A. Karatzoglou, I. Arapakis, J.M. Jose, Graph highway networks. arXiv:2004.04635
63. T. Yu et al., Gradient surgery for multi-task learning, in *Advances in Neural Information Processing Systems*, volume 33 ed. by H. Larochelle et al. (2020). arXiv:2001.06782
64. D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs), in *4th International Conference on Learning Representations* (2016). arXiv:1511.07289
65. D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in *3rd International Conference on Learning Representations*, ed. by Y. Bengio, Y. LeCun (2015). arXiv:1412.6980
66. J. Pata, J.M. Duarte, A. Tepper, jpata/particleflow: MLPF DELPHES paper software release. https://github.com/jpata/particleflow (2021). https://doi.org/10.5281/zenodo.4559587
67. L.R.M. Mohan et al., Studying the potential of Graphcore IPUs for applications in particle physics. arXiv:2008.09210
68. J. Duarte et al., Fast inference of deep neural networks in FPGAs for particle physics. JINST **13**, P07027 (2018). https://doi.org/10.1088/1748-0221/13/07/P07027. arXiv:1804.06913
69. Y. Iiyama et al., Distance-weighted graph neural networks on FPGAs for real-time particle reconstruction in high energy physics. Front. Big Data **3**, 44 (2021). https://doi.org/10.3389/fdata.2020.598927. arXiv:2008.03601
70. A. Heintz et al., Accelerated charged particle tracking with graph neural networks on FPGAs, in *3rd Machine Learning and the Physical Sciences Workshop at the 34th Conference on Neural Information Processing Systems* (2020). arXiv:2012.01563
71. J. Duarte et al., FPGA-accelerated machine learning inference as a service for particle physics computing. Comput. Softw. Big Sci. **3**, 13 (2019). https://doi.org/10.1007/s41781-019-0027-2. arXiv:1904.08986
72. J. Krupa et al., GPU coprocessors as a service for deep learning inference in high energy physics. https://doi.org/10.1088/2632-2153/abec21. arXiv:2007.10359 (**Accepted by Mach. Learn.: Sci. Technol.**)
73. D.S. Rankin et al., FPGAs-as-a-Service Toolkit (FaaST), in *2020 IEEE/ACM International Workshop on Heterogeneous High-performance Reconfigurable Computing (H2RC)* (2020). https://doi.org/10.1109/H2RC51942.2020.00010 arXiv:2010.08556
74. CMS Collaboration, "The Phase-2 Upgrade of the CMS Endcap Calorimeter", CMS Technical Design Report CERN-LHCC-2017-023. CMS-TDR-019, CERN (2017)
75. ATLAS Collaboration, "Deep Learning for Pion Identification and Energy Calibration with the ATLAS Detector", ATLAS Public Note ATL-PHYS-PUB-2020-018, CERN (2020)
76. L. De Oliveira, B. Nachman, M. Paganini, Electromagnetic showers beyond shower shapes. Nucl. Instrum. Methods A **951**, 162879 (2020). https://doi.org/10.1016/j.nima.2019.162879. arXiv:1806.05667
77. C.M.S. Collaboration, Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV. JINST **13**, P05011 (2018). https://doi.org/10.1088/1748-0221/13/05/P05011. arXiv:1712.07158
78. ATLAS Collaboration, Identification of boosted Higgs bosons decaying into b-quark pairs with the ATLAS detector at 13 TeV. Eur. Phys. J. C **79**, 836 (2019). https://doi.org/10.1140/epjc/s10052-019-7335-x. arXiv:1906.11005
79. ATLAS Collaboration, ATLAS b-jet identification performance and efficiency measurement with $t\bar{t}$ events in pp collisions at $\sqrt{s} = 13$ TeV. Eur. Phys. J. C **79**, 970 (2019). https://doi.org/10.1140/epjc/s10052-019-7450-8. arXiv:1907.05120

80. E. Bols et al., Jet flavour classification using deepjet. JINST **15**, P12012 (2020). https://doi.org/10.1088/1748-0221/15/12/P12012. arXiv:2008.10519

81. CMS Collaboration, Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques. JINST **15**, P06005 (2020). https://doi.org/10.1088/1748-0221/15/06/P06005 arXiv:2004.08262