THE EUROPEAN
PHYSICAL JOURNAL C

Reconstructing particles in jets using set transformer and hypergraph prediction networks

Francesco Armando Di Bello[1,a] , **Etienne Dreyer**[2,b] , **Sanmay Ganguly**[3], **Eilam Gross**[2], **Lukas Heinrich**[4], **Anna Ivina**[2], **Marumi Kado**[5,6], **Nilotpal Kakati**[2,c] , **Lorenzo Santi**[6], **Jonathan Shlomi**[2], **Matteo Tusoni**[6]

[1] INFN and University of Genova, Genoa, Italy
[2] Weizmann Institute of Science, Rehovot, Israel
[3] ICEPP, University of Tokyo, Tokyo, Japan
[4] Technical University of Munich, Munich, Germany
[5] Max Planck Institute for Physics, Munich, Germany
[6] INFN and Sapienza University of Rome, Rome, Italy

Abstract The task of reconstructing particles from low-level detector response data to predict the set of final state particles in collision events represents a set-to-set prediction task requiring the use of multiple features and their correlations in the input data. We deploy three separate set-to-set neural network architectures to reconstruct particles in events containing a single jet in a fully-simulated calorimeter. Performance is evaluated in terms of particle reconstruction quality, properties regression, and jet-level metrics. The results demonstrate that such a high-dimensional end-to-end approach succeeds in surpassing basic parametric approaches in disentangling individual neutral particles inside of jets and optimizing the use of complementary detector information. In particular, the performance comparison favors a novel architecture based on learning hypergraph structure, *HGPflow*, which benefits from a physically-interpretable approach to particle reconstruction.

1 Introduction

Testing theories in high energy physics rely on the ability to reconstruct high energy particle collision events using information recorded by particle detectors. General-purpose detectors enable this primarily through two sources of information: charged particle trajectories (tracks) measured in an inner tracking region and energy deposited by particle showers in a surrounding array of calorimeter cells.

Currently, experiments at the CERN Large Hadron Collider (LHC) employ parameterized "particle-flow" algorithms, which combine track and calorimeter information in a complementary way while avoiding double counting.

The performance of particle-flow algorithms is limited to an extent by detector design specifications, such as the precision and size of the inner tracking system, the magnetic field strength in the tracking volume, the granularity of the calorimeters, and their energy resolution. However, a number of intrinsic factors complicate the task of particle reconstruction in the LHC environment: the busy and often collimated signatures resulting from proton collisions, the presence of multiple simultaneous scattering events (pileup), and finally, the extensive and irregular array of sensitive elements required for granularity and angular coverage.

There are two main approaches to particle-flow algorithms. The approach used by the ATLAS collaboration [1] involves subtracting the expected shower profile for each track in an event from the calorimeter deposits to infer the energy contributed by nearby neutral particles. The CMS collaboration, on the other hand, employs a *global* particle-flow algorithm where final state particles of different types are reconstructed simultaneously [2]. Global particle-flow algorithms allow a high physics analysis flexibility and eliminate the need for overlap-removal algorithms while better exploiting the strengths of each sub-detector system.

In this paper, we approach the global particle-flow paradigm using machine learning (ML) models operating on graph data. As in other applications to particle physics, ML brings the advantage of replacing parameterized cuts (for example, in energy subtraction schemes) with fully differen-

[a] e-mail: francescoarmando.dibello@unige.it

[b] e-mail: etienne.dreyer@weizmann.ac.il

[c] e-mail: nilotpal.kakati@weizmann.ac.il (corresponding author)

tiable decision boundaries in the full space of relevant features in data. The expressiveness of ML models also opens new possibilities, such as reconstructing individual neutral particles inside of jets. Similarly, the choice to represent input data as graphs is motivated by several advantages: graphs more naturally capture the spatial correlations encoded in irregular detector geometry and also are well-suited for the sparsity and variable cardinality of the input set. Graph neural networks (GNN) have therefore emerged as an architecture of choice in recent particle reconstruction models, as they have in other particle physics tasks [3].

In a collision event, the true set of particles T upstream to the detector sensitive volume gives rise to a set of detector-level hits D. So the input set comprising the detector record is sampled from $p(D|T)$. Then global particle-flow reconstruction is the set-to-set task where the input set of detector-level hits D is transformed into a typically much smaller output set R comprising N_R predicted particles. The predictions of a successful reconstruction algorithm $R(D)$ will correctly model the cardinality N_T of T and the properties (class, momentum, and angular coordinates) of its members. Several ML approaches have been proposed in the literature to predict $R(D)$.

In [4] the object condensation (OC) approach was proposed, which clusters nodes or pixels in latent space to form candidate objects, in our case, particles. Recently, OC has been used to predict clusters in CMS data [5,6], where the authors focused on reconstruction efficiency and energy regression of showers from single particles embedded in pileup. We implement OC with modifications as explained in Sect. 3.4 for the purpose of establishing a performance baseline for an ML-based particle reconstruction.

The reduction in size from input to output set is handled in the MLPF [7] approach by assigning input nodes to particle classes in the output set or else to a dedicated "neglect class". This approach was also recently successfully tested using CMS data [8,9], where the model predictions were trained to match the output candidates from a standard particle-flow algorithm. For predicting true particles, MLPF is limited to cases where one or more clusters can be associated to each particle. It would therefore be required to define a fractional target definition in order to efficiently reconstruct particles that do not contribute a dominant fraction of energy in any single cluster (for example, a significant percentage of low-p_T photons).

In this paper, we contribute to the exploration of GNN-based particle reconstruction by proposing two new algorithms and comparing their performance alongside a modified OC implementation as a baseline and a parameterized particle flow algorithm. Rather than full proton-proton collision events, we focus on events comprising a single jet, which represent the typical domain over which inter-particle correlations are expected to play a significant role in recon-struction. Our dataset incorporates full GEANT4 [10] treatment of particle showers in a nearly-hermetic calorimeter simulation [11]. An example of a simulated single jet event is shown in Fig. 1. In the true particle-flow paradigm, our approach is built around the idea of combining low-level features from calorimeter showers with the complementary information provided by tracks.

We find that a novel application of recurrent hypergraph learning leads to the most accurate results and preserves a high degree of interpretability. This is achieved thanks to a physics-inspired approach which allows the network to exploit the relationships between properties of the target particles and their energy deposits in the detector.
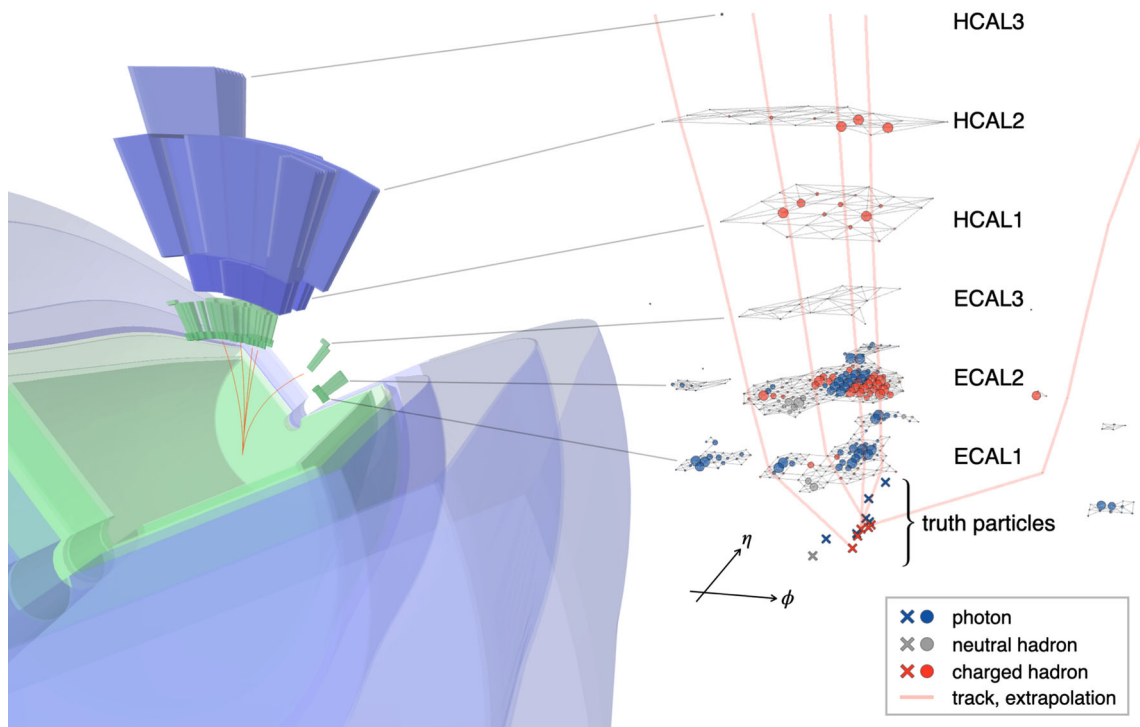
2 Dataset

2.1 Detector simulation

Unlike the full detector models used to simulate experiments at the LHC, publicly-available codes such as DELPHES [12] do not model particle interactions with sufficient complexity to enable training a network with the full calorimeter signature available at real detectors. This motivated the development of the Configurable Calorimeter simulatiOn for AI (COCOA) package [11], which we used to generate the datasets in this paper.

The geometric coverage of the COCOA calorimeter is split into a barrel ($0.0 < |\eta| < 1.5$) and two identical endcaps ($1.5 < |\eta| < 3.0$) regions. The endcap region is situated in a hermetic way such that there is no void in the transition region. In depth, the calorimeter has a total of six concentric layers: the first three layers comprising an electromagnetic calorimeter (ECAL) and the next three a hadronic calorimeter (HCAL). The calorimeters have uniform segmentation in η and ϕ enabling high spatial resolution, as listed in Table 1. The geometric depth of the cells is modulated as $1 / \cosh \eta$ in order to achieve a constant effective interaction depth with increasing η.

The inner region of COCOA is immersed in a uniform axial magnetic field of 3.8T that extends until a radius of 150 cm, where four 1.1 cm layers of iron immediately precede the ECAL. The ECAL is modeled as a homogeneous calorimeter by mixing lead and liquid argon, corresponding to ATLAS calorimeter materials, in volume proportion 1.2:4.6 leading to a radiation length of $X_0 = 2.5$ cm. For the HCAL, iron is used as the absorber material, and polyvinyl toluene plastic material as the scintillating active material. These are mixed with a volume proportion 1.1:1.0, yielding a nuclear interaction length of $\lambda_{int} = 26.6$ cm. The simulated energy deposits in each layer are smeared to reproduce the expected sampling energy resolution. For our dataset, the hadronic sampling term is 10%. The effect of pileup and elec-

Fig. 1 A depiction of a single-jet event from the test dataset in both the COCOA calorimeter layers (left) and as an input graph in $\eta-\phi$ space (right). On the left, the actual geometry of the calorimeter cells is shown, while on the right, they are represented by spheres with sizes proportional to their energy divided by noise threshold (up to a maxi-mum value). Lines represent tracks and their projected locations in η and ϕ in each calorimeter layer. Connections between calorimeter cells are the edges formed during graph construction (inter-layer edges and track-cell edges are not shown). The markers at the bottom right indicate the $\eta-\phi$ coordinates of the truth particles

Table 1 Characteristics of the six calorimeter layers: depths in radiation length (X_0) and nuclear interaction length (λ_{int}), granularity, and standard deviations of the simulated noise distributions

Layer	Depth	Granularity ($\eta \times \phi$)	Noise [MeV]
ECAL1	$4X_0$	256×256	13
ECAL2	$16X_0$	256×256	34
ECAL3	$2X_0$	128×128	41
HCAL1	$1.5\lambda_{int}$	64×64	75
HCAL2	$4.1\lambda_{int}$	64×64	50
HCAL3	$1.8\lambda_{int}$	32×32	25

tronic noise is mimicked using normal distributions centered at zero with widths varying according to the layer. The choice of material and smearing parameters is tuned to reproduce the ATLAS calorimeter system's single-particle response.

The effect of tracking is emulated by smearing truth charged particles with a resolution $\frac{\sigma(p)}{p} = a \times p$ with $a = 10^{-5}/\text{GeV}$. The smearing of the track direction is neglected as it is expected to have a subdominant effect in our problem of interest.

Charged particles produced from hadrons decaying-in-flight above a transverse radius $R > 75\,\text{mm}$ (250 mm) in the barrel (endcap) have no tracks associated to them. To focus on the reconstruction of particles as they appear at the calorimeter, the dataset simulates photon conversions only at the stage of the iron layer prior to the calorimeters, while material interactions within the tracker are emulated solely by the track q/p smearing.

2.2 Dataset generation

Event generation, followed by parton shower and hadronization is performed with PYTHIA8 [13] with a single initial state quark or gluon particle. The parton initial energy is sampled in the range 10–200 GeV, and angular coordinates are distributed uniformly in the range $\eta \in [-2.5, 2.5]$, $\phi \in [-\pi, \pi]$. Final state particles are interfaced with GEANT4 to simulate their interaction with material, both showering in the calorimeter and scattering and e.g. photon conversions in the iron layer preceding it. Additional pileup collisions were not simulated. The targets of the machine learning algorithms are final state stable particles with transverse momentum above 1 GeV, which reach the calorimeter.

In each event, a standard clustering algorithm is used to group calorimeter cells into "topoclusters" based on their proximity and deposited energy, following the algorithm

described in [14] with minor modifications. First, an energy over noise ratio of $(\frac{E}{\sigma}) > 4.6$ is used to identify cluster seeds. For each seed, a two-stage search is performed in its vicinity to group neighboring cells with nonzero energy. The first search collects neighboring cells with an energy-to-noise threshold ratio above 2, and the second search further extends the clusters with cells that have energy above 0. Finally, the algorithm applies a set of rules to merge topoclusters sharing seed cells and split topoclusters formed by particles in close proximity.

A record is kept of contributing particles and their energy contribution to each cell. Electronic noise is simulated in the calorimeter cells at realistic levels and dominates a fraction of the clustered cell. A small fraction of topoclusters, therefore, consist purely of cells where noise was the dominant contributor. One or more such topoclusters are present in 23% of the training events.

In summary, the data used for ML comprise the following object collections: cells which belong to a topocluster, all tracks that reached the calorimeter, and the set of particles which entered the calorimeter. An identical configuration is used to generate the dataset of 50,000 events for training and the independent dataset of 30,000 events for testing. In addition, a "gluon jet" dataset containing 30,000 events is generated by replacing the single incident quark by a gluon with the same initial energy and angular distributions. The quark and gluon jet datasets are provided in [15]. The results obtained with this gluon jet sample are discussed in Sect. 4.5. Figure 2 summarizes the number of various entities stored in both the single jet and gluon jet test datasets.

2.3 Fiducial particle definitions for reconstruction targets

In a collision event, not all particles produced can be reconstructed in the detector. When defining target truth particles it is important to account only for those that can be detected, i.e. those that are within the detector acceptance and have sufficiently high transverse momentum to be reconstructed. Beyond these simple criteria, particles produced in the collision can later decay or interact with the detector and convert, radiate or interact and produce other particles. The specific definition of the particles that are targets for the reconstruction algorithm, referred to as fiducial particles, is important to remove ambiguities during training and in assessing the performance of reconstruction. To qualify as fiducial particles, truth stable particles must have the following properties:

– $p_T > 1$ GeV
– be produced before the first calorimeter layer
– release a nonzero amount of energy in the calorimeter.

Additional consideration would be needed to achieve a more realistic environment where bremsstrahlung, pair pro-
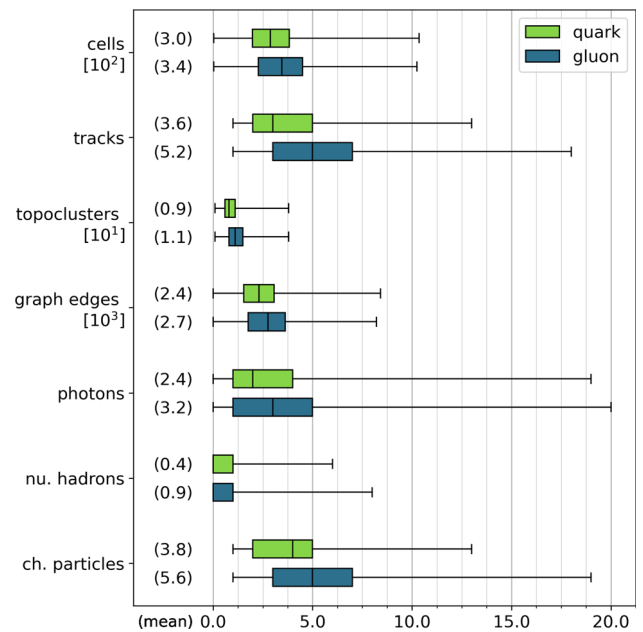

Fig. 2 Composition of the quark and gluon test samples in terms of the cardinality of different sets of entities contained. The mean value in each case is written in parentheses while the range and quartiles of the distributions over events are shown in the box plot

duction, and the presence of soft particles in general might result in highly collimated topologies, above the spatial reconstruction capabilities of the detector. In this work, the absence of pileup and the absence of material in the inner tracking region justify the use of the three fiducial criteria described above.

2.4 Input graph

We build a fixed heterogeneous graph out of each event by connecting calorimeter cells and tracks based on their proximity. Each cell is connected to the k nearest cells in the same calorimeter layer, where $k = 8$ in the ECAL and $k = 6$ in the HCAL. Additionally, each cell is connected to the single nearest cell in its immediately adjacent layer(s). A cell in layer l can only receive incoming edges from other cells if they are separated in ΔR by less than $\{d_{\max}^{c-c}\}_l = \{0.05, 0.07, 0.14, 0.30, 0.30, 0.60\}$ for the six calorimeter layers. A set of indices and weights is assigned per cell listing the true particles which contributed and their relative contribution to the total cell energy. An index of -1 is given to energy contributions from noise.

Tracks are likewise connected to cells based on closest separation in ΔR between the cell and the projected $\eta-\phi$ coordinate of the track in the corresponding calorimeter layer. A track is connected to a maximum number of $k = 4$ cells in each ECAL layer and $k = 3$ cells in each HCAL layer. For track-cell edges, a larger maximum ΔR is allowed: $\{d_{\max}^{t-c}\}_l =$
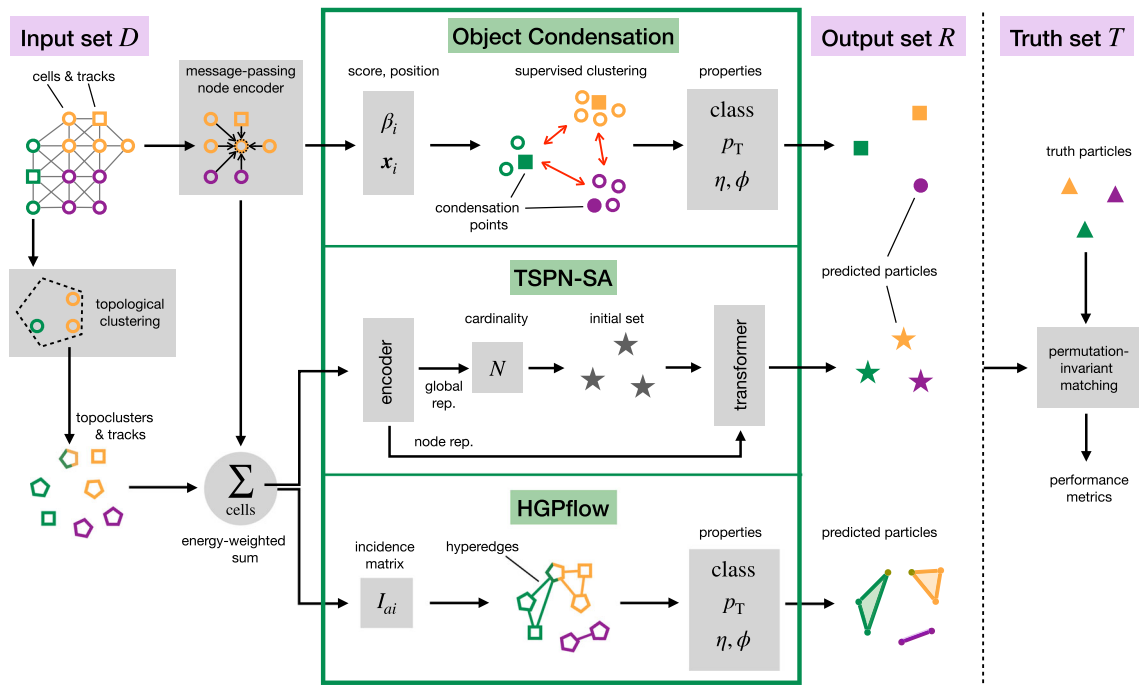
Fig. 3 Comparison of the different ways in which the three ML reconstruction algorithms map the input set of nodes in the form of a graph to the output set of predicted particles, to be compared with the set of truth particles. Colors indicate distinct particles and the nodes for which they are the dominant contributor

{0.15, 0.15, 0.40, 1.10, 1.10, 2.00}. A depiction of the graph connectivity for tracks and cells is shown in Fig. 1.

Topoclusters are represented in the input graph by a separate set of nodes with edges connecting each to the set of cells belonging to the topocluster. The angular coordinates of a topocluster are taken at its energy barycenter.

3 Particle reconstruction algorithms

3.1 Parameterized particle-flow algorithm

To compare the performance of the ML algorithms, we implemented a traditional parameterised particle-flow algorithm [1], which we refer to as PPflow. The algorithm aims at subtracting the energy deposited in the calorimeter from charged particles associated to tracks. To this end, shower templates are derived from single π^+ samples and parameterized as a function of the track p_T and the layer where the first nuclear interaction takes place. The energy subtraction is performed in concentric rings of radius equal to a single cell pitch built from the extrapolated track position in each calorimeter layer. The ring's energy is progressively subtracted from the topoclusters until the expected total energy determined in the single π^+ template is reached. The remaining energy in the topoclusters after this subtraction is considered as originating from neutral particles. The

PPflow algorithm does not aim at reconstructing the single particles composing the jets, but rather it is designed to estimate the overall neutral energy component for each topocluster.

3.2 Common description for the ML algorithms

We investigate three ML-based particle reconstruction models for the set-to-set prediction $R(D)$: object condensation (OC) as an existing ML baseline, transformer set prediction network with slot attention (TSPN-SA), and a hypergraph architecture (HGPflow). Descriptions of each algorithm is given in Sects. 3.4, 3.5, and 3.6. Section 4 compares their relative particle-level performance and a comparison to the PPflow baseline for jet reconstruction.

There are commonalities to all three algorithms. Predicted particles in each case are inferred from the node features (skip connections) concatenated with a node representation vector from a common encoder network, discussed in Sect. 3.3. Tracks are treated similarly in each case: the charged particle cardinality in an event is set by the number of tracks, and the charged particles' η and ϕ are determined directly from the tracks without regression. While the OC algorithm takes calorimeter cells as input nodes, the other two algorithms use topoclusters instead, to reduce the dimensionality of the input. Figure 3 illustrates the core differences between the

ways each algorithm maps the set of detector-level nodes D to the set of predicted particles R.

The choice to use calorimeter cells compared to using the coarser topoclusters can be compared in terms of an *injective condition*: the degree to which the energy deposit in a node can be mapped back to a single parent particle. In the case of cells, although contributions from more than one parent particle are present in general, the injective condition is more valid than in the case of topoclusters. Since the injective condition is an assumption of the OC algorithm (i.e. in the definition of the entries of I_{ki} in Eq. 1), this motivates the choice of cells as input nodes.

Having contributions to a node from more than one parent particle can be learned in the TSPN-SA architecture in an unsupervised way via node-particle attention. The HGPflow architecture, on the other hand, is fully equipped to disentangle multiple-particle contributions to a node thanks to supervised learning of the incidence matrix, discussed in Sect. 3.6. In both cases, computing gradients for predictions on edges becomes significantly more expensive for cell-level inputs compared to topocluster-level inputs, which was the main motivation for choosing the latter.

The loss associated with predicted particle properties is computed similarly in each algorithm. Particle class is trained using a categorical cross-entropy term between the predicted and the target class. A mean squared error loss term is used to regress continuous properties η_i and $p_{T,i}$. The ϕ prediction is trained using $1 - \cos(\phi^{\text{pred}} - \phi^{\text{targ}})$.

The total number of trainable parameters in the neural network blocks of the OC, TSPN-SA, and HGPflow algorithms is compared in Table 2 including the node encoding network in each. An estimate of their computational performance is also shown. For each of the three algorithms, hyperparameter optimization scans have not been performed, except on the threshold cuts used during inference for OC and HGPflow. The code for the algorithms is provided in [16].

3.3 Graph nodes encoding

Each event is represented as a heterogeneous graph comprising track, cell, and topocluster nodes connected by edges as defined in Sect. 2.4. The embedding model described in the following is shared among the different network architectures. Figure 4 illustrates the network components of the encoding model: input feature vectors associated with track and cell nodes are passed through separate networks to embed them in a common representation space of dimension 100. The cell features input to the embedding are (energy, position, ϕ, η, layer). Similarly, the track input features are the

Table 2 Comparison of the three particle reconstruction algorithms in terms of model size and computational resources. The number of trainable parameters belonging to the node encoding model is shown alongside the total. The time per event is averaged over 100 single jet events evaluated sequentially, and the memory is estimated as the peak memory over the same. Results are obtained on the same GPU (NVIDIA TITAN RTX)

Algorithm	# Parameters Total (Node enc.)	Speed* [ms/event]	Memory [MiB/event]
OC	1.8M (0.2M)	249	1480
TSPN-SA	1.5M (0.2M)	465	1448
HGPflow	1.8M (0.2M)	257	1394

*Algorithms not optimized for execution time

track parameters $(q/p, \theta, \phi, d_0, z_0)$[1] and the extrapolated η–ϕ coordinates of the track at each calorimeter layer. The latter are important features because a charged particle after exiting the magnetic field travels in a straight line which no longer points back to the origin (assuming no material interactions). In addition to their hidden representation, these nodes are also given an additional binary feature which flags whether they originate from cells or tracks.

The node encodings are then updated to incorporate the graph relational structure via 4 successive blocks of message passing along edges. In each block, a dedicated network is used with the following three inputs concatenated: current node representation, sum of representations from neighboring nodes, and a graph-level global representation (the mean of all current node representations). Following the message passing blocks, topocluster representations are computed by the energy-weighted mean of the cell representation vectors belonging to the topocluster.

3.4 Modified object condensation (OC)

The OC algorithm was proposed in [4] for tasks of segmenting a set of input nodes into a set of target objects and prediction of their properties, which it does simultaneously. In the particle reconstruction case, the input set comprises tracks and calorimeter cells and the output set of objects are the progenitor particles ("parents") with their classes and properties. The set-to-set procedure for OC is shown in the top row of Fig. 3. Our implementation follows the original OC approach with some modifications which are stated in the following description.

The OC algorithm is based on clustering nodes according to their parents in a learned few-dimensional space x. The clustering is supervised by adding to the loss potentials defined on this space: a repulsive potential $\hat{V}(x) \propto$

---

[1] The track impact parameters d_0 and z_0 measure the distance of closest approach of the track to the beam line in the transverse and longitudinal directions, respectively.
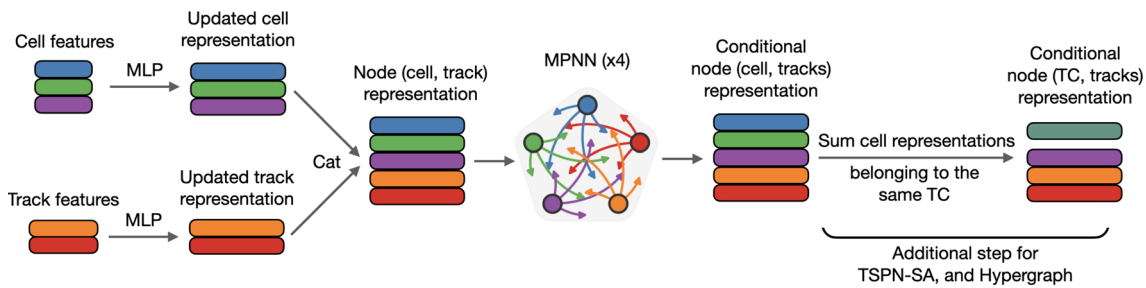
Fig. 4 The encoding model used to derive a learned node representation

$\max(2 - \Delta x, 0)$ between nodes that belong to different parent particles, and an attractive one $\check{V}(x) \propto \Delta x^2$ for nodes having a common parent. The goal is that after training the resulting clusters of nodes will correspond to the set of parent particles.

However, calculating the sum of N^2 pairwise potential terms during training becomes expensive for problems of even moderate N. This is addressed by designating a single representative node for each parent particle, called a *condensation point*, to impose the potentials on all other nodes during training. A separate network is trained to predict a score, $\beta \in [0, 1]$, with a target value of 1 for condensation points and 0 otherwise. An increasing function $q(\beta) = \mathrm{arctanh}^2 \beta + q_{\min}$ (with $q_{\min}$ a hyperparameter) is used in analogy to charge in the loss term responsible for the clustering potentials:

$$L_V = \frac{1}{N} \sum_{i=1}^{N} q(\beta_i) \sum_{k=1}^{K} \left[I_{ki} \check{V}_{c,k}(x_i) + (1 - I_{ki}) \hat{V}_{c,k}(x_i) \right] \tag{1}$$

where I_{ki} is an $N \times K$ matrix determining whether particle k is the parent of node i. The matrix will be revisited in Sect. 3.6.

For each node i, which is a cell, the properties loss is of the same form as discussed in Sect. 3.2, where target class is either photon, neutral hadron or charged particle. Similar to L_V above, the particle property loss is also weighted by $q(\beta)$ such that nodes with the highest β receive the most supervision during training. These nodes are ultimately selected for the output set during inference by requiring their predicted $\beta > t_\beta$ and that they be separated in the clustering space by $\Delta x > t_d$, where t_β and t_d are two threshold hyperparameters.

Compared to the original OC model, our implementation has two modifications connected to the condensation score β. The condensation points defined during training do not have a physical meaning and are learned in an unsupervised way. In our approach, we instead use the following physics-oriented definition:

$$\mathrm{CP}_k^T = \begin{cases} \text{track} \in k, & \text{if } k \text{ is charged particle} \\ \mathrm{argmax}_z(\text{cells} \in k), & \text{if } k \text{ is neutral particle} \end{cases} \tag{2}$$

where z is the energy over noise threshold ratio for each cell. This definition removes the need to identify a representative node for charged particles, assuming a 1–1 mapping to tracks in the event. For neutral particles, on the other hand, the β prediction is fully supervised and can be interpreted as the likelihood that a cell has maximal z in a given shower. Since this cell also serves as an approximate location of the shower center, the η and ϕ for neutral particles are regressed via a learned offset to the cell η–ϕ coordinates. During inference, condensation points passing the t_b and t_d thresholds are further required to be classified as either photon or neutral hadron, whereas cells classified as charged particles are discarded (since this role is fulfilled by tracks).

A second modification compared to the original OC approach is that instead of a $\sim (1 - \beta)$ regression-type loss computed on condensation points only, we train the β prediction using a binary cross-entropy loss evaluated for all nodes. The reasoning behind a classification-type loss is to directly penalize the network for predicting large β for nodes which are not condensation points, i.e. false positives. In an ablation study, each modification was seen to bring substantial improvement at essentially no additional model complexity.

Besides the two modifications above, our OC implementation differs from that of [6] in a few regards. Firstly, they propose an upgrade to the original OC algorithm where particles are represented in the clustering space not only by a singular condensation point but by the β-weighted average over a learned distance scale. The authors of [6] report that these and other modifications lead to improved training stability and reduced noise, so we recognize the potential for improving our implementation as well. Finally, we point out that our model has significantly more parameters (Table 2), with GravNet blocks [17] replaced by the node encoding model described in Sect. 3.3. Our choice of network block sizes has not been optimized for computational efficiency and allocates a large proportion of its parameters to the node prediction networks compared to the message-passing networks.
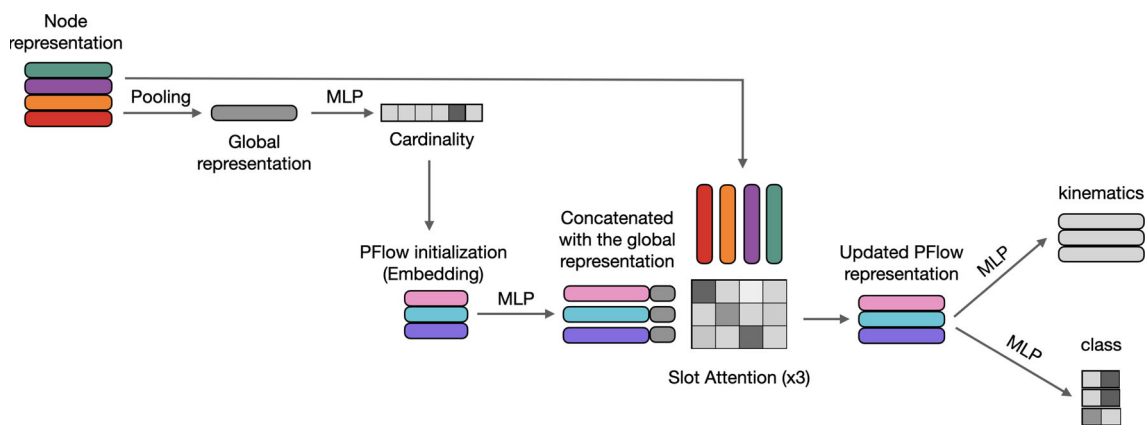
Fig. 5 The TSPN-SA architecture. The cardinality of the set of output particles is predicted from the global representation, while their properties are predicted from representation vectors resulting from successive slot attention blocks

3.5 Transformer set prediction network with slot attention (TSPN-SA)

The Transformer Set Prediction Network (TSPN) was initially developed for the permutation-invariant encoding and decoding of variable-size sets of feature vectors [18]. The utility of this for set-to-set problems in particle physics is clear: a model is needed that predicts an output set of entities (i.e. particles) based on an input set of different entities (i.e. calorimeter clusters, tracks), where both sets typically have different cardinality. The model is divided into two networks: the first for predicting neutral particles and the second for predicting charged particles (discussed later).

As shown in the top part of Fig. 5, the first architecture starts with a set encoder network whose output is used to predict the number of neutral particles, N^{pred} This prediction is trained using a categorical cross-entropy loss over 25 classes, which is an upper bound on the number of neutral particles per event. The cardinality prediction is used during test time. During training, the truth cardinality is enumerated to form a set of numbers that are passed through embedding layers to instantiate the initial set of random vectors. These vectors combined with the global representation vector function as queries for a series of 3 slot-attention (SA) blocks [19]. The SA blocks are not part of the original TSPN proposal but were found to be fundamental for performance in our application (abbreviated TSPN-SA). Each block contains 3 iterations where the particle hidden representation is updated using the attention-weighted representations of the topoclusters and tracks in the event. Finally, the updated hidden representations are inputs to two dedicated neural network blocks aimed at predicting the kinematics (p_T, η, ϕ) and the class of the particle candidates.

The neutral particle properties loss function for the TSPN-SA algorithm is defined as the sum of a categorical cross-entropy term for class t_i and a mean-squared error (MSE)

term for the continuous properties $\boldsymbol{p}_i = \{\eta_i, \phi_i, p_{T,i}\}$ computed for each particle candidate. The target particles are defined by matching to the set of predicted particles using the Hungarian assignment algorithm [20], with the loss itself being the distance metric.

The second part of the TSPN-SA algorithm, for predicting charged particles, makes explicit use of the prior knowledge originating from track-particle objects. Each track is promoted to a particle, such that the cardinality of the output set is fixed by the number of tracks. Similar to the neutral architecture, SA blocks are used to update the hidden representation of the charged particle candidates. Unlike the model for neutral particles, the only predicted quantity for charged particles is their transverse momentum in order to improve over the track resolution. For charged particles, the track index is used to match with the corresponding target particle. As in the OC algorithm, the η and ϕ of the charged particles are taken directly from their representative tracks. Finally, the total loss for the TSPN-SA algorithm is the sum of neutral particle cardinality loss, the neutral particle properties loss, and the charged particle p_T MSE loss. These loss terms are minimized simultaneously during training.

3.6 HGPflow: particles as hyperedges

A hypergraph is a generalization of a graph where hyperedges can each connect one, two, or multiple nodes (Fig. 6). While connectivity in a graph of N nodes is described by an $N \times N$ adjacency matrix, a hypergraph containing K hyperedges is described by an incidence matrix $I^{(N \times K)}$. In the context of particle reconstruction, calorimeter deposits and tracks can be represented as nodes in a hypergraph, while each particle is represented by a hyperedge connecting the set of nodes to which it contributed. We propose *HGPflow*: an algorithm that treats particle reconstruction as a task of learning hyperedges
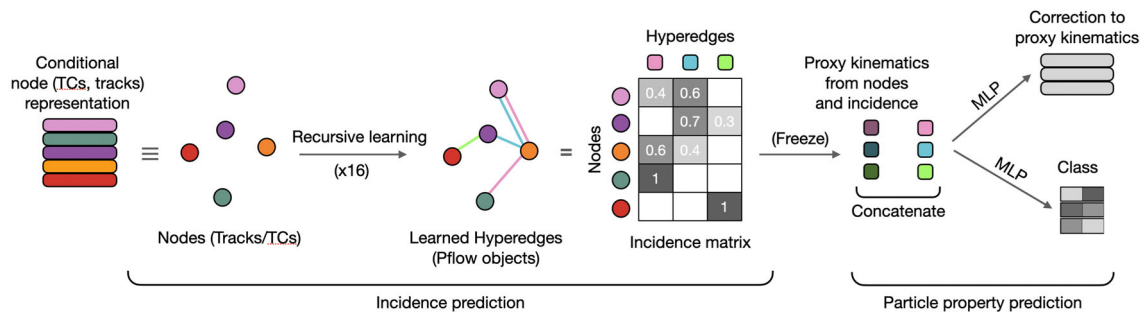
Fig. 6 The two stages of learning in the HGPflow algorithm. The objective of the first stage is to predict the fractional entries in the energy-weighted incidence matrix, where columns correspond to hyperedges (i.e. particles). This is done by accumulating the loss over a sequence of recursive updates. In the second stage, the incidence matrix is frozen and the network minimizes losses for particle property predictions, defined relative to proxy quantities

and their properties. There are two objectives in the training of HGPflow:

1. predict the incidence matrix defining the hyperedges
2. predict the hyperedge (i.e. particle) properties.

The first objective is similar to the task of separating overlapping charged and neutral showers which was the focus of [21]. In this first stage, the HGPflow network predicts $(N + 1) \times K$ entries comprising a zero-padded incidence matrix and an additional row of binary values that indicates whether the particle corresponding to a given column exists or not. Since the number of particles per event varies, the number of columns K is set to an upper bound on the number of particles estimated from the training set (in our case $K = 30$). To express a non-injective map from particles to nodes, we define a target incidence matrix which has fractional rather than binary-valued entries. The entry relating node i to particle a is the following:

$$[I]_{ia} = \frac{E_{ia}}{\sum_{\text{particles } b} E_{ib}} = \frac{E_{ia}}{E_i} \qquad (3)$$

where E_{ia} is the amount of energy that particle a contributes to the total energy E_i of node i. For nodes which are tracks, incidence entries are simply 1, whereas for topoclusters they compute the fraction of the topocluster's energy that came from a given particle. An example of target and predicted incidence matrix entries are shown in Fig. 7 for one event.

Predicted rows in the incidence matrix are normalized using Softmax (i.e. sum over all hyperedges for a given node is 1) before being compared to the target via Kullback–Leibler divergence loss.

$$L_{\text{inc}} = \sum_a \text{KL}_i \left(I_{ia}^{\text{targ}}, \text{Softmax}_i(I_{ia}^{\text{pred}}) \right). \qquad (4)$$

The predicted entries of the indicator row are passed through a sigmoid function and compared to the (binary) target entries
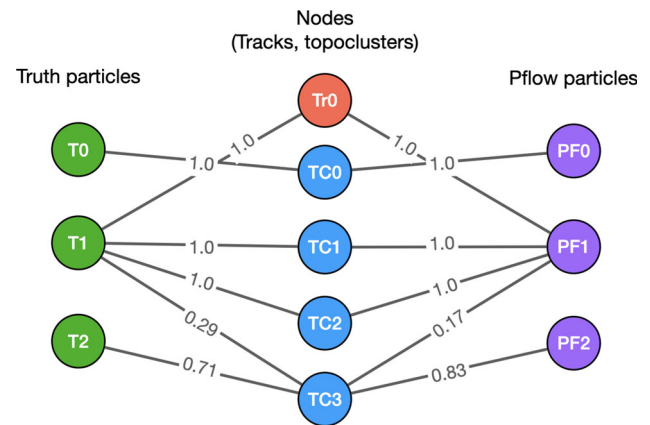
Fig. 7 Schematic representation of the truth and predicted incidence matrix in HGPflow for one event. The left part of the diagram shows the three truth particles in the event. One of them has a track (Tr) associated to it. The three particles deposit their energy into four topoclusters (TC). The links represent the fractional energy originated by a given particle in a given topocluster or track. The right part of the diagram shows the predicted values of the incidence matrix for each reconstructed particle

using a binary cross entropy loss function. Predicted columns are rearranged using the Hungarian algorithm to minimize the loss.

The incidence matrix prediction network is trained using the recurrent strategy proposed by [22], described briefly hereafter. The loss in Eq. 4 is calculated for a sequence of 16 refinement blocks each comprising an updated prediction of the incidence matrix followed by an update of node representation vectors V, and hyperedge representation vectors E. The iteration $t \to t + 1$ is performed with the following three successive steps:

$$I_{ia}^{t+1} = \phi_I \left(v_i^t, e_a^t, I_{ia}^t \right) \qquad (5)$$

$$V^{t+1} = \phi_V \left(\left\{ v_i^t, \rho_{E \to V}(i, t), v^0 | i = 1 \dots n \right\} \right) \qquad (6)$$

$$E^{t+1} = \phi_E \left(\left\{ e_a^t, \rho_{V \to E}(a, t) \ | a = 1 \dots k \right\} \right) \qquad (7)$$

where $\rho_{E \to V}(i, t) = \sum_a I_{ia}^{t+1} e_a^t$ and $\rho_{V \to E}(a, t) = \sum_i I_{ia}^{t+1} v_i^t$ are aggregations of node (v) and hyperedge (e) representation vectors weighted by the updated incidence matrix. The updates are performed at each step using the same networks ϕ_I, ϕ_V, and ϕ_E, where the latter two networks are DeepSets models [23].

To reduce computational cost, not every iteration of the backward pass is included in the gradient step. Two sequences of 4 adjacent iterations are randomly selected out of the 16 for which the incidence loss is computed and added to the loss from the prediction at the end of the sequence.

The second training objective of the HGPflow network (Fig. 6c) is to predict particle properties for each hyperedge. The corresponding loss function contains classification and regression terms evaluated by matching predicted and target particles using the Hungarian algorithm. Particles corresponding to hyperedges where the predicted indicator was below the threshold are matched to dummy targets and weighted by zero in the loss. Classification between photons and neutral hadrons is performed for hyperedges which do not contain a track and are thus identified as neutral particles. The regression task benefits from a unique advantage enabled by learning the incidence matrix (Eq. 3): particle kinematics can be approximated as weighted sums and averages over the input features of the topoclusters contained in the hyperedge. Proxy quantities (denoted ˆ) for energy and angular coordinates can be computed as:

$$\hat{E}_a = \sum_{\text{nodes } i} E_i I_{ia}, \qquad \{\hat{\eta}_a, \hat{\phi}_a\} = \sum_{\text{nodes } i} \{\eta_i, \phi_i\} \tilde{I}_{ia} \qquad (8)$$

where a dual incidence matrix $\tilde{I}$, normalized over node instead of particle indices, can be defined:

$$\tilde{I}_{ia} = \frac{E_{ia}}{\sum_{\text{nodes } j} E_{ja}} = \frac{E_{ia}}{E_a} = \frac{E_i \cdot I_{ia}}{\sum_{\text{nodes } j} (E_j \cdot I_{ja})}. \qquad (9)$$

The property prediction networks in HGPflow are therefore given the simpler objective of learning corrections to the approximate values from Eq. 8. The loss terms used for the property predictions follow the description in Sect. 3.2.

Therefore, neutral particle kinematics (p_T, η, ϕ) are regressed by predicting an offset to the proxy values in Eq. 8. For charged particles, an offset is likewise predicted for the p_T measured from the associated track. The properties loss is computed by matching predicted and target particles using the Hungarian algorithm [20].

4 Performance of particle reconstruction in jets

One of the most challenging tasks of global particle flow algorithms is the reconstruction of particles in dense environments, in particular jets. In this section, the performance of

the ML reconstruction algorithms will be assessed by quantifying the similarity between the set of predicted and set of target particles. The following four types of metrics are meant to evaluate the cardinality, class, and properties predictions:

– Efficiency and fake rate
– Classification purity
– Particle angular and momentum resolution
– Jet-level quantities.

The efficiency and fake rate are defined as follows:

$$\epsilon \equiv \frac{N(\text{matched pred})}{N(\text{targ})}, \quad f \equiv \frac{N(\text{unmatched pred})}{N(\text{pred})}. \qquad (10)$$

The quality of the regression tasks is evaluated from distributions of their residuals, defined as $(y_{\text{targ}} - y_{\text{pred}})/y_{\text{targ}}$ for particle property $y \in \{p_T, \eta, \phi\}$.

4.1 Particle matching

Predicted and target particles are matched using the Hungarian algorithm to find the pairings which minimize the distance between their properties, defined by the following metric:

$$d_{\text{match}} = \sqrt{(\Delta p_T / p_T^{\text{truth}})^2 + \Delta R^2} \qquad (11)$$

where Δ denotes the difference between a predicted and target property, and $\Delta R^2 = \Delta \eta^2 + \Delta \phi^2$. Matching is performed separately for neutral particles and charged particles since the latter are distinguished by the presence of a track. The coefficients c_{p_T} and $c_{\Delta R}$ are set to 1 and 5 for neutral particles while for charged, matching is based only on ΔR (i.e. $c_{p_T} = 0$). Prioritizing spatial matching helps decouple reconstruction efficiency from classification accuracy, which in particular will dominate at low-p_T because of the similarity of photon and neutral hadron signatures.

In each event, when the cardinality of the predicted set of particles is larger than that of the target, the non-matched predictions are labeled as "fake" particles. Conversely, inefficiency arises when not enough neutral or charged particles were predicted in order to match every target.

4.2 Charged particle performance

Charged particles include electrons, muons, and charged hadrons. In jets charged pions produced during hadronization account for around 90% of all charged particles. Leptons such as electrons and muons are present in less than 3% of the jets. Electrons are produced from photon conversions and hadrons decaying in flight while muons are mostly produced by the latter mechanism. Given the large class imbalance and the fact that no dedicated studies have been performed to improve the classification of electrons inside jets, the three
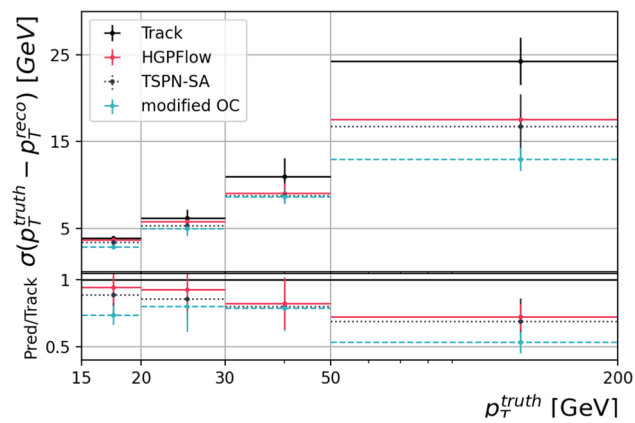
Fig. 8 Resolution of charged particle-flow candidates and tracks as a function of the associated particle transverse momentum. At high p_T the particle-flow candidates show improved resolutions over the tracks

classes are grouped together and characterized as a single class of charged particles.

Tracking efficiency presents an upper bound on the efficiency of charged particle reconstruction (see Sect. 2.1). Since fake tracks are not emulated in the data, charged particle fake rates are neglected in this study. In any case, the rate of fake tracks at 1 GeV is typically at the percent level for the ATLAS and CMS experiments, which is expected to have a small impact.

In cases where the track belonging to a charged particle is not reconstructed, the target particle is relabelled to avoid confusing the network during training. Charged hadrons without a track are relabelled as neutral hadrons, and electrons without a track as photons. Photon pairs from neutral pion decays prior to the calorimeter are treated as two distinct target particles.

A key characteristic of charged particle reconstruction is the resolution of p_T with respect to the true value. It is well known that at low transverse momentum tracks provide the best momentum estimate over the calorimeter resolutions. An opposite trend appears at high energies where the calorimeter systems provide the most accurate energy measurement. Figure 8 shows the resolutions of charged particles reconstructed with the three ML approaches and compared to the track resolutions for charged particles with $p_T > 15$ GeV. Below this value, the particle p_T regression is replaced with simply the track p_T since an improvement is not expected. An increasing improvement at high p_T is observed for all the ML algorithms demonstrating that indeed the complementarity between the calorimeter and tracking measurements has been learned during training.

4.3 Photon and neutral hadron performance

The presence of photons inside jets is mainly due to decays of neutral pions and to a lesser extent from bremsstrahlung

processes. Long-lived neutral hadrons on the other hand trace back to the shower of the initial partons. Disentangling these two components is not a trivial task and is detector-dependent – for COCOA it is observed that 70% of neutral hadrons below 5 GeV release all their energy in the ECAL layers, making it difficult to distinguish them from photons. This fraction steeply decreases with the energy of the initial hadrons to approach percent levels at around 20 GeV. Neutral particle reconstruction is further complicated inside the collimated environment because of the frequent overlaps between neutral showers. For this reason, efficiency and fake rate plots are computed by considering photon and neutral hadron predictions inclusively, without requiring a match between predicted and target class.

Efficiencies, fake rates, and class prediction accuracy for photons and neutral hadrons are shown in Fig. 9. The efficiency of reconstructing photons with $p_T > 2$ GeV is above 90% for HGPflow, rising to 98% for photons above $p_T > 30$ GeV. For TSPN-SA and OC, the efficiency for photons reaches 95% and 90%, respectively. Neutral hadrons above 5 GeV are reconstructed with efficiencies ranging from 76 to 86% for the three algorithms. Figure 9c shows that the rate of producing unmatched photon predictions drops from 16% (30%) at a predicted p_T of 2 GeV to 1.4% (11%) above 30 GeV for HGPflow (TSPN-SA). For neutral hadrons, the fake rate (Fig. 9d) is a factor of 2–4 times larger across the full p_T range for HGPflow, and 1.6–2.8 times larger for TSPN-SA.

For the TSPN-SA and HGPflow algorithms, reconstructing neutral particles at low-p_T is challenging because a large fraction of the target particles does not contribute a leading amount of energy to any topocluster in the event (33% of photons and 25% of neutral hadrons). In HGPflow, this is compensated in a supervised manner by learning subdominant contributions to topoclusters as fractional entries in the predicted incidence matrix. This limitation could be further overcome by using cell-level input nodes such as for the OC algorithm.

Efficiency and fake rate plots are complemented by studying the probability of misclassification between photons and neutral hadrons. In Fig. 9e and f, both HGPflow and TSPN-SA algorithms exhibit high accuracy of classification for predictions matched to photons, and for neutral hadrons an accuracy that rises with p_T: from 51 to 76% for HGPflow and from 16 to 63% for TSPN-SA. A lower accuracy is expected when considering that due to the class imbalance between photons and neutral hadrons of 5.8:1 (inclusive), a random classifier would have an accuracy of roughly 15% for neutral hadrons. Moreover, the class imbalance is p_T-dependent, with the proportion of photons dropping off faster than neutral hadrons.

The OC algorithm behaves similarly to the others for reconstruction efficiency of neutral particles, albeit with lower performance in most bins. For the fake rate, and in

(a)

(b)

(c)

(d)

(e)

(f)
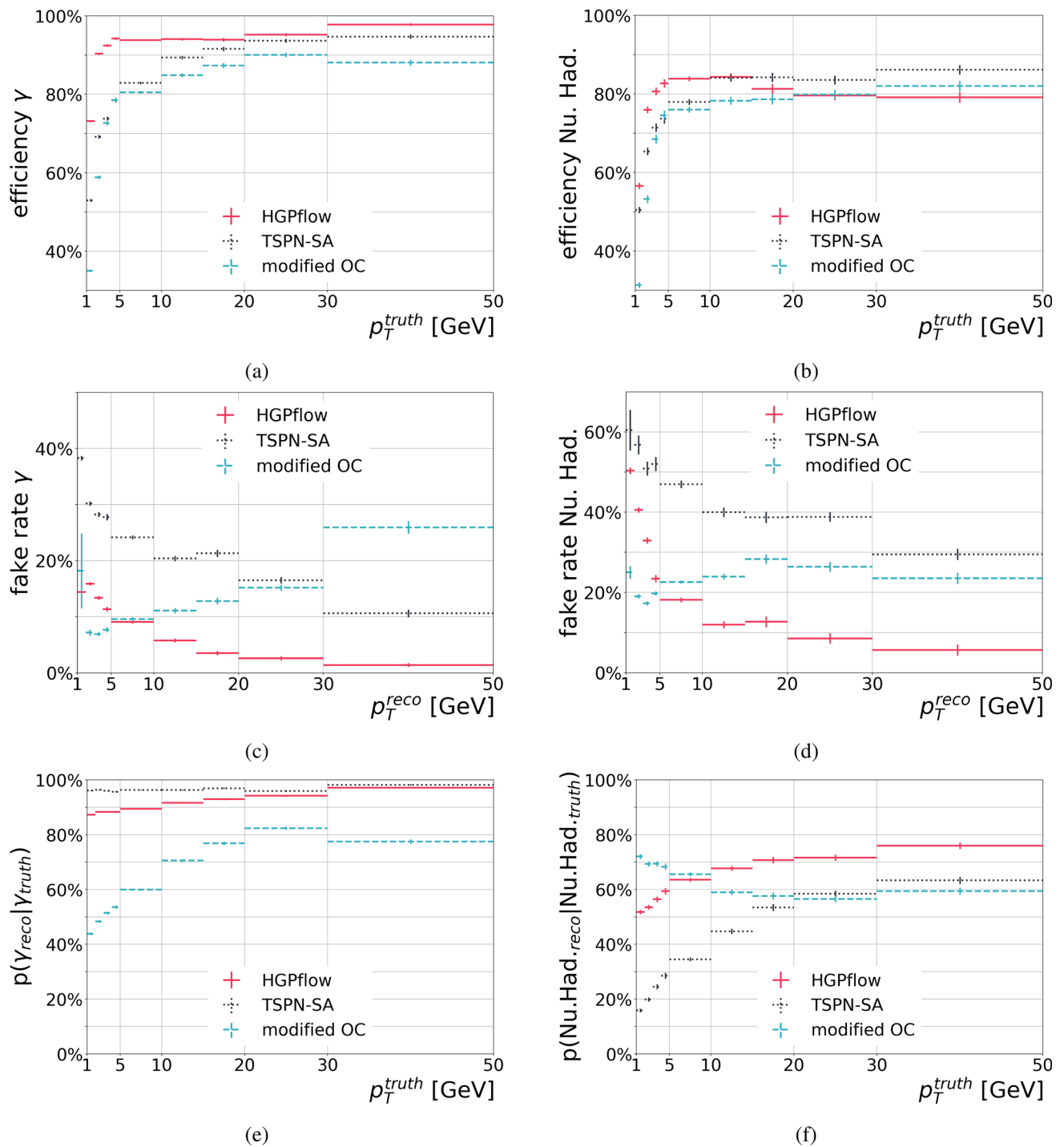
Fig. 9 Top: efficiency of matching predicted neutral particles to truth photons (**a**) and neutral hadrons (**b**) in a jet as a function of the associated truth particle p_T. Middle: fake rate, i.e. probability that predicted photons (**c**) and neutral hadrons (**d**) are not matched to any truth neutral particle, as a function of the predicted particle p_T. Bottom: the prob-ability that the predicted neutral particles which are matched to truth photons (**e**) and neutral hadrons (**f**) are assigned the correct class. For each curve, the misclassification probability is simply the difference of the curve from 1
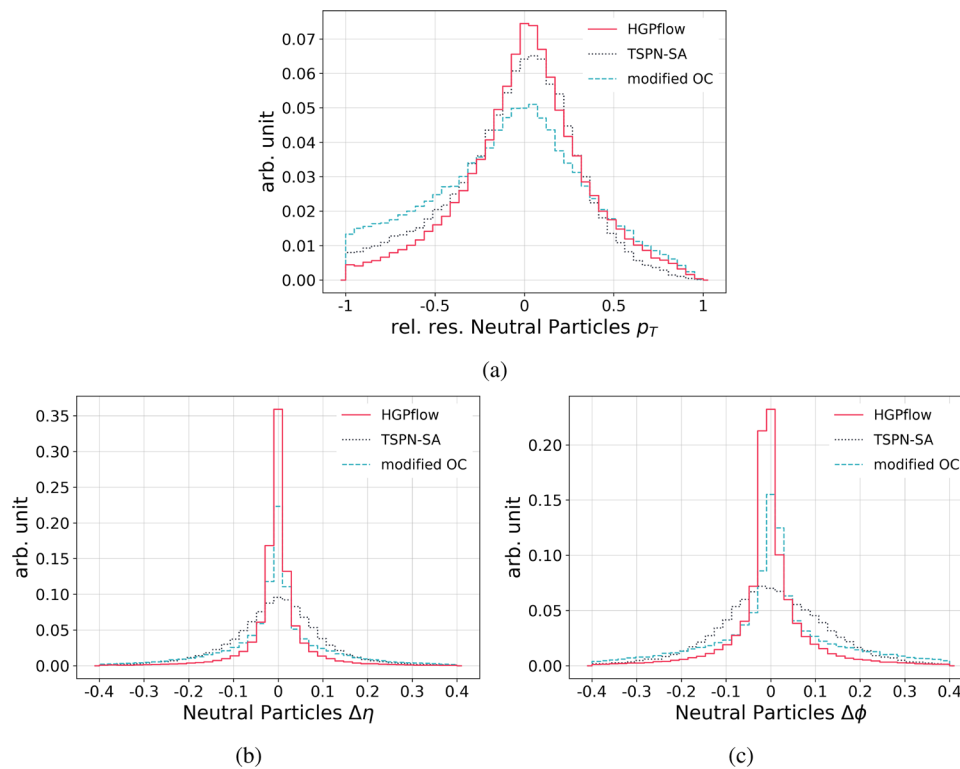
Fig. 10 Distributions of relative residuals between predicted and true p_T (**a**), η (**b**), and ϕ (**c**) of reconstructed neutral particles

particular, for photons, OC exhibits an increase with p_T. The classification accuracy for OC is also lower for photons and shows a different trend in p_T for neutral hadrons compared to HGPflow and TSPN-SA. These differences are related to the fact that in OC neutral particle predictions correspond to a subset of calorimeter cells passing the selection defined by the t_b and t_d threshold cuts. This introduces a sensitivity of the neutral particle cardinality to the number of cells per particle, which grows as a function of particle p_T. The trend in fake rate appears to reflect this. Furthermore, towards high p_T a growing majority of cells belong to showers of charged particles, which makes the classification task more challenging in the OC approach (which involves 3 classes, unlike HGPflow and TSPN-SA). Introducing p_T-dependent weights on the β and x prediction tasks could help counter this effect.

Finally, relative residuals are used to quantify the ability to correctly predict the p_T, η, and ϕ of the reconstructed neutral particles, shown in Fig. 10a, b, and c, respectively. The HGPflow algorithm shows the best performance at estimating accurately both angular variables and momentum. It is interesting to note that the TSPN-SA algorithm has the worst performance for the angular variables. This is related to the usage of topoclusters in a less supervised way compared to HGPflow, where it is known which topoclusters contributed to the formation of a particle. The OC algorithm instead uses the more granular calorimeter cells directly showing similar

performance to the HGPflow algorithm for angular regression.

The models also perform differently for neutral particle p_T regression, shown in Fig. 10a. The OC model has a tendency to overestimate the neutral particle p_T. A similar trend is less pronounced in the TSPN-SA regression, while the distributions of predictions from HGPflow exhibit the least skew, in addition to the smallest mean and variance of the three.

4.4 Jet-level performance

The ability to efficiently reconstruct jets and correctly predict their properties is a priority for experiments at the LHC. Jet performance depends on the overall efficiency, fake-rates and kinematic regression of the constituent particles, therefore being an important test for the ML reconstruction algorithms.

Following evaluation of the networks, jets are built using the anti-k_t algorithm [24] with a radius parameter of 0.4 and a minimum number of 2 constituents. We define three sets of jets with differing input constituents:

– **Truth jets**: jets built using the set of target particles
– **ML jets**: jets built using the sets of particles predicted by the OC, TSPN-SA, and HGPflow algorithms
– **PPflow jets**: jets built using tracks and topoclusters with the charged energy subtraction procedure of a parameterized particle-flow algorithm (see Sect. 3.1).
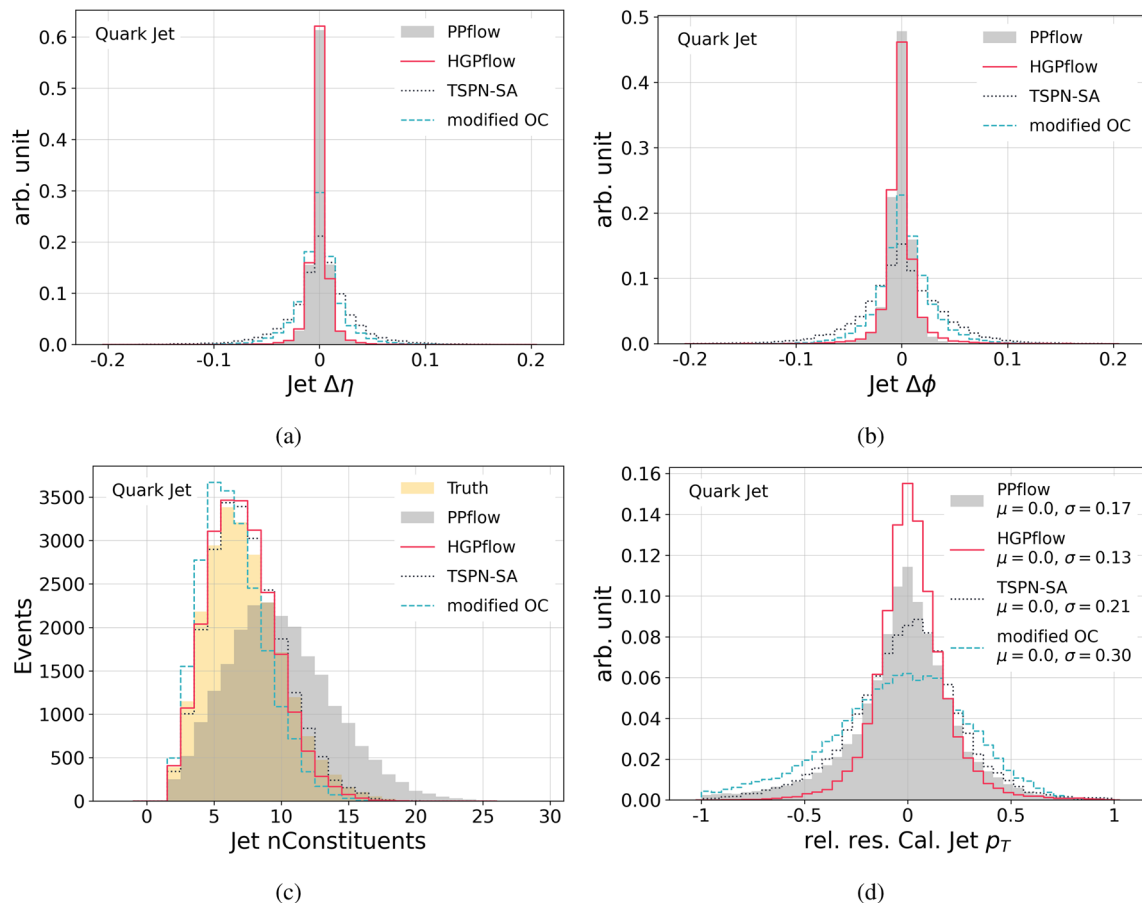
Fig. 11 Jet-level performance metrics shown for the three algorithms and the PPflow reference in comparison to true jets. Angular residuals between reconstructed and true jets (**a** and **b**), the number of jet-constituents in (**c**), and the calibrated p_T relative residuals in (**d**)

The number of constituents is shown in Fig. 11c for each algorithm and compared to that of the true jet. The ML algorithms, by accounting for neutral particles in the jet, are able to model this reasonably well. On the other hand, the PPflow distribution overestimates the truth distribution as expected, since its constituents are tracks and topoclusters rather than particles (Fig. 12).

In order to further optimize the jet-p_T and provide a more quantitative figure of the jet resolutions, a simplistic calibration is applied. First jet p_T residual distributions are computed in different p_T bins. For each, a dedicated scale factor is computed. A functional fit is performed and the corresponding scale factor is applied to reconstructed jets based on their p_T. This procedure is applied separately to each reconstruction algorithm.

Relative residuals are shown in Fig. 11. As observed for neutral particles, HGPflow shows the best performance at the jet-level. In terms of jet angular observables, HGPflow is comparable to the traditional PPflow approach while for jet p_T resolution it shows a 24% improvement relative to PPflow. The TSPN-SA jet p_T resolution is better than OC, while for angular observables OC performs slightly better.

To help visualize the jet reconstruction task, Fig. 12 displays an event from the test dataset showing predictions from the trained HGPflow algorithm. In this example, each of the four neutral particles at truth level can be matched to a predicted particle with the correct class and an η–ϕ prediction consistent within the cell granularity. The calorimeter panels illustrate the tight arrangement of topoclusters used as input nodes for the HGPflow prediction.

4.5 Performance on gluon jets

To study the ability of the particle reconstruction algorithms to generalize beyond the training data to a new physics process, we evaluate the trained models on the dataset of single gluon jet events described in Sect. 2.2. The difference arising at the parton shower for gluon-initiated jets reflects itself in the dataset feature distributions, for example, the larger multiplicity of cells, tracks, and particles shown in Fig. 2. Since the appropriate upper bound on the number of particles was determined based on the training dataset, a single gluon jet found to contain > 30 particles was excluded from the dataset in order to evaluate HGPflow. The results for the three
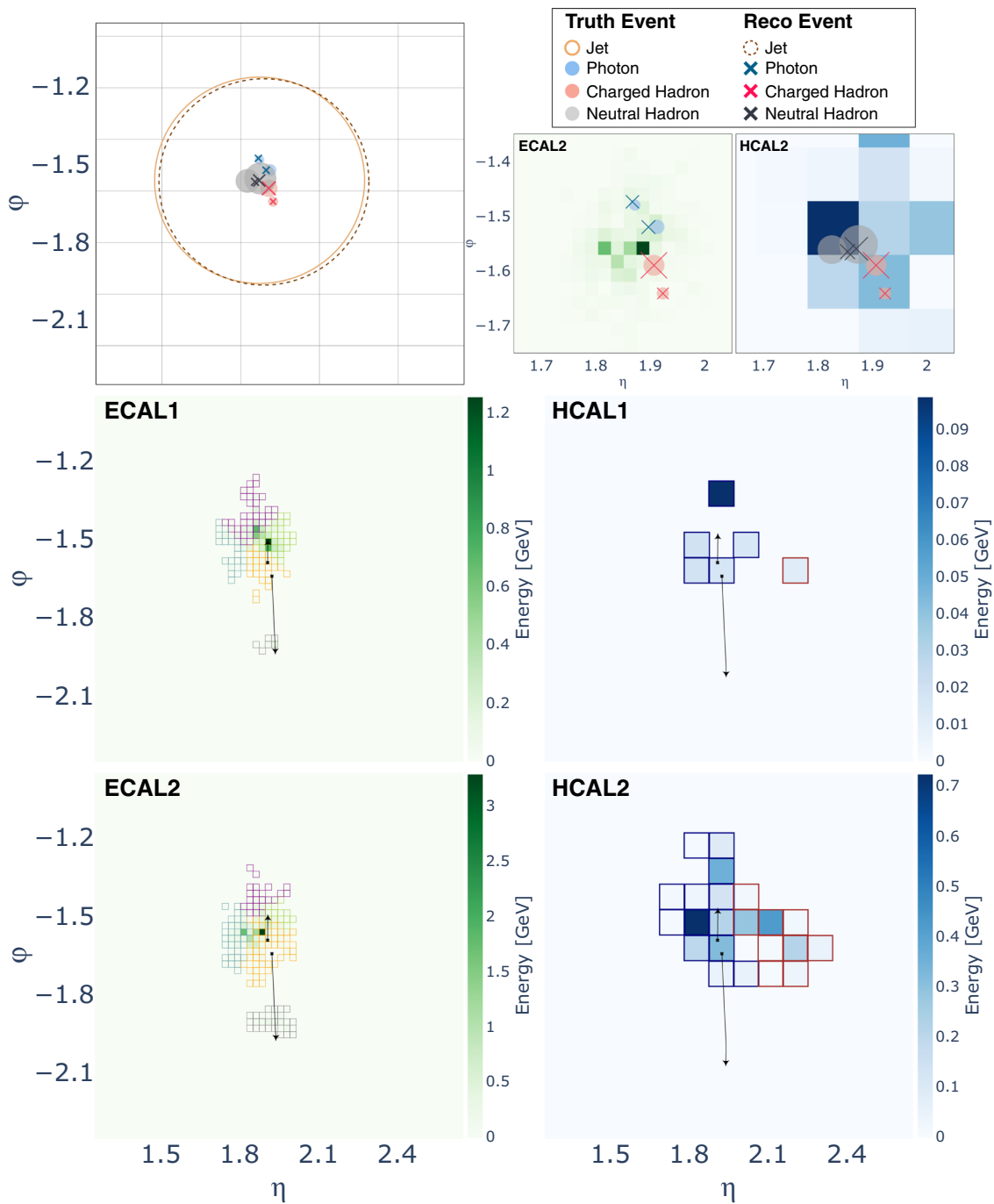
Fig. 12 Event display of a single jet event from the test dataset. In the top left panel, the $\eta-\phi$ coordinates of truth particles with momentum above 1 GeV are shown as circles along with the set of predicted particles from the HGPflow algorithm shown as crosses. The set of particles and their p_T at truth level are as follows: two photons in blue (1.8, 3.0 GeV), a pair of neutral K_L^0 mesons in grey (12.3, 22.3 GeV), and two charged pions in red (2.2, 6.5 GeV). The circles of $R = 0.4$ represent anti-k_t jets built from the truth (solid) and predicted (dashed) particle sets that nearly overlap in $\eta-\phi$ and have p_T agreement within 35%. In the top right panels, the truth and predicted particles are shown overlaid on a zoomed region of the ECAL2 and HCAL2 layers. In the bottom panels, the detector-level information serving as input to the reconstruction algorithms is shown for each of the first two layers of both ECAL and HCAL in the same $\eta-\phi$ plane. Cells that have the same border color belong to the same topocluster. Green and blue fill is used to indicate the energy of cells in the ECAL and HCAL layers, respectively. The arrows describe the tracks for charged particles from the interaction point with the arrowheads indicating the angular coordinate extrapolated at the given layer
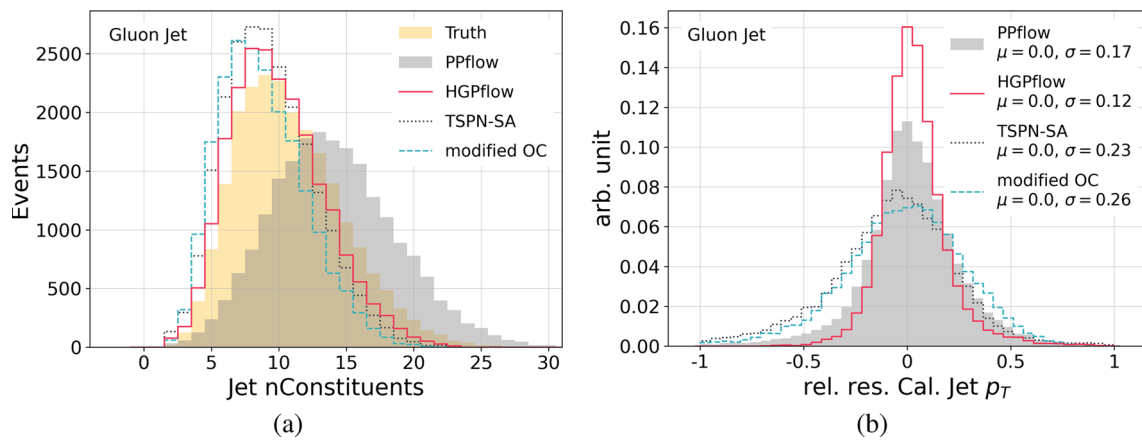
Fig. 13 Distributions of the number of constituents per jet (**a**), and distributions of relative residuals between predicted and true p_T (**b**) for the three ML particle reconstruction algorithms evaluated on the sample of gluon jets

ML algorithms are shown along with the PPflow comparison in Fig. 13. Overall, the algorithms demonstrate an ability to generalize: the number of predicted constituents is shifted slightly lower with respect to truth compared to the quark jet case (Fig. 11c), while the jet relative p_T residual distributions are comparable to Fig. 11d. The rank of the algorithms in terms of performance remains the same as before, with HGPflow again boasting narrower jet p_T resolution than the PPflow comparison.

5 Discussion

5.1 Perspective on the ML algorithms

Compared to the OC and TSPN-SA algorithms and the PPflow benchmark, HGPflow shows the best performance in terms of jet momentum resolution, which was not directly a training objective. This traces back to superior modeling of neutral particle momentum shown in Fig. 10a. Unlike the other ML models, which must learn implicitly that a given energy deposit in the calorimeter cannot be associated with more than one parent particle, HGPflow benefits from being structured around the concept of energy conservation. Successfully predicting an incidence matrix defined via Eq. 3 and the hyperedge indicator row entails knowing the energy contributions a given topocluster received from all particles (Fig. 7). Furthermore, the normalization ensures that energy attributed to a given particle candidate is not counted again in assignments to other particles. Since both the hyperedge representation and the proxy for neutral particle energy (Eq. 8) are weighted by entries of the incidence matrix, the property predictions which stem from these inputs inherit a bias towards energy conservation.

The hypergraph approach allows common elements of both the OC and TSPN-SA approaches to be handled in a more clear formalism. In the OC potential loss (Eq. 1), a binary-valued incidence matrix I_{ik} functions as a lookup table determining whether a node is repelled or attracted to the representation of a particle (i.e. condensation point). The clustering of nodes according to parent particle can thus be thought of as an indirect way of learning I_{ik}, limited by the extent to which the injective condition applies (discussed in Sect. 3.2). Likewise, the TSPN-SA algorithm is built around an attention matrix between particle candidates k and nodes i from the input set which resembles an incidence matrix, although it is normalized along columns rather than rows. The attention weights also have a latent rather than physical meaning and are learned in an unsupervised way. On the other hand, HGPflow not only explicitly predicts the incidence matrix, which is the key to unraveling overlapping particle showers, but expresses it in the physical basis of energy contributions with the advantage mentioned previously.

We anticipate that the structure of HGPflow can be extended in at least three ways. First, the input set granularity has been set without tuning to that of topoclusters from a standard ATLAS-like algorithm. This granularity can be increased to further enable the segmentation of overlapping energy deposits from nearby particles. Second, in the trainings for our results, the two objectives of incidence and properties prediction have been carried out nearly independently. However, a more powerful representation learning scheme could lead to a model which learns these two objectives in a synergistic way, allowing the incidence prediction to be informed by the properties prediction and vice versa. Finally, while Table 2 indicates an acceptable inference time of HGPflow, more optimization is needed to reduce its training duration. This could be achieved by hyperparameter optimization in the recurrent training configuration and by exploring alternatives to the Hungarian match-

ing, which the authors of [22] identified as a computational bottleneck.

Similarly, besides the modifications proposed in [5], we suspect that the OC algorithm can be substantially improved in future work. While the TSPN-SA and HGPflow algorithms both have neural network layers for information exchange following the node encoding (i.e. successive attention and incidence weighted-updates), the node predictions of our OC algorithm are likely limited by a comparatively narrow receptive field. One way to increase the receptive field in OC is to add additional message passing blocks in the node encoder model, although a limited study of this option did not lead to conclusive improvement. Another possibility is introducing an attention mechanism. The distance between nodes i and j in the latent clustering space entails a term $a_{ij} = x_i \cdot x_j$ with the form of attention, e.g. in the attractive potential, $\check{V} \propto \Delta x^2 \ni -2a_{ij}$. Therefore the clustering mechanism in OC seems a natural point to introduce transformer blocks for enhanced information exchange, at the cost of computing additional gradients for the set of edges.

5.2 Datasets for future work

Several opportunities emerge for future investigation on new datasets. The performance reported in this work for $R = 0.4$ quark and gluon jets suggests the application to substructure reconstruction in large-R jets from boosted boson decays. The goal of studying particle reconstruction on a single jet dataset was to focus on the local system of overlapping particle showers which represents the kernel of the problem at the full-event level. For this reason, we envision that reconstructing full events could proceed by mapping the same trained model onto spatial partitions of the detector hits D defined by topological and jet clustering algorithms, for instance. Given an effective scheme to deal with potential overlap, each partition could be treated as an approximately isolated set of input nodes, graph edges, and attention or incidence matrix weights. In this case the resource requirements reported in Table 2 would scale linearly with the number of partitions. Ignoring overlap, an upper bound on the number of $R = 0.4$ partitions of a full event could be estimated as $2\pi \cdot 6/0.4 \simeq 10^2$.

Studying the robustness of the models in the presence of pileup will also be an important follow-up task (see [6] for existing work in this direction). Likewise, the impact of interactions with material upstream of the calorimeter needs to be thoroughly addressed in a future dataset. In this case, electron pair production from early conversions and photons from bremsstrahlung will require a thoughtful definition of target particles to ensure that they can be feasibly distinguished during training. For example, photon conversions could be treated as a separate class, and electrons could be defined as

targets depending on the quality of their associated track (if present at all).

6 Conclusions

In this paper, we applied ML-based reconstruction algorithms to the dense environment of a jet. Single-jet datasets were generated using a realistic calorimeter model to mimic the complexity and input features of data from the LHC. Particle-flow reconstruction is inherently a set-to-set task suited for ML applications. Three ML algorithms – a modified version of Object Condensation, a set-transformer architecture with slot attention (TSPN-SA), and a novel hypergraph learning approach (HGPflow) – were compared by their ability to reconstruct particles in the jet and jet-level quantities using an input graph comprising tracks and calorimeter clusters. In particular, the algorithms were trained to reconstruct individual neutral particles in the dense environment, a task going beyond the scope of traditional particle flow algorithms.

For charged particles, the algorithms learned to exploit the complementary information provided by calorimeter activity to improve on the measured track momentum. The efficiency of reconstructing photons and neutral hadrons reached 90% and 80%, respectively, for $p_T > 10$ GeV. The neutral particle fake rates were more variable for each algorithm, with the best performance being 10% and lower for $p_T > 10$ GeV.

Jets formed from the predicted particles were compared to those from the true particles and also a parameterized particle flow baseline. HGPflow showed the best performance and surpassed the baseline in terms of both angular and momentum resolution of the jet. This can be explained from the fact that the hypergraph formalism is structured around energy conservation, which also makes its predictions more interpretable from a physics point of view. We anticipate that the suitability of the hypergraph formalism for the set-to-set task of particle reconstruction is yet to be fully leveraged. By demonstrating the potential of ML algorithms to disentangle the jet dense environment, our findings motivate the application to full collision events in future work.

Data Availability Statement This manuscript has associated data in a data repository. [Authors' comment: The manuscript contains associated data available in the data repository [15].]

References

1. ATLAS collaboration. Jet reconstruction and performance using particle flow with the ATLAS Detector. Eur. Phys. J. C, **77**(7), 1–47 (2017)
2. A.M. Sirunyan, CMS Collaboration et al., Particle-flow reconstruction and global event description with the CMS detector. JINST **12**(10), P10003 (2017)
3. J. Shlomi, P. Battaglia, J.R. Vlimant, Graph neural networks in particle physics. Mach. Learn. Sci. Technol. **2**(2), 021001 (2020)
4. J. Kieseler, Object condensation: one-stage grid-free multi-object reconstruction in physics detectors, graph, and image data. Eur. Phys. J. C **80**(9), 1–12 (2020)
5. S.R. Qasim, K. Long, J. Kieseler, M. Pierini, R. Nawaz, Multi-particle reconstruction in the High Granularity Calorimeter using object condensation and graph neural networks. EPJ Web Conf. **251**, 03072 (2021)
6. S.R. Qasim, N. Chernyavskaya, J. Kieseler, K. Long, O. Viazlo, M. Pierini, N. Nawaz, End-to-end multi-particle reconstruction in high occupancy imaging calorimeters with graph neural networks. Eur. Phys. J. C **82**(8), 753 (2022)
7. J. Pata, J. Duarte, J.R. Vlimant, M. Pierini, M. Spiropulu. MLPF: efficient machine-learned particle-flow reconstruction using graph neural networks. Eur. Phys. J. C **81**(5), 1–14 (2021)
8. J. Pata, J. Duarte, F. Mokhtar, E. Wulff, J. Yoo, J.R. Vlimant, M. Pierini, M. Girone. Machine learning for particle flow reconstruction at CMS. J. Phys. Conf. Ser. **2438**(1), 012100 (2022)
9. F. Mokhtar, J. Pata, J. Duarte, E. Wulff, M. Pierini, J.R. Vlimant, Progress towards an improved particle flow algorithm at CMS with machine learning, in *ACAT 2022* (2023). arXiv:2303.17657
10. S. Agostinelli et al., GEANT4: a simulation toolkit. Nucl. Instrum. Meth. **506**(3), 250–303 (2003)
11. F.A. Di Bello, A. Charkin-Gorbulin, K. Cranmer, E. Dreyer, S. Ganguly, E. Gross, L. Heinrich, L. Santi, M. Kado, N. Kakati, P. Rieck, M. Tusoni, Configurable calorimeter simulation for AI applications (2023). Preprint. arXiv:2303.02101
12. J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, M. Selvaggi, DELPHES 3: a modular framework for fast simulation of a generic collider experiment. JHEP **2**, 2014 (2014)
13. C. Bierlich et al., A comprehensive guide to the physics and usage of PYTHIA 8.3. SciPost Phys. Codebases, 008 (2022)
14. ATLAS Collaboration. Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1. Eur. Phys. J. C **77**(7), 1–73 (2017)
15. Training and test datasets. https://doi.org/10.5281/zenodo.7699681
16. Code for algorithms. https://github.com/nilotpal09/hg-tspn-pflow
17. 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**(7), 608 (2019)
18. A.R. Kosiorek, H. Kim, D.J. Rezende, Conditional set generation with transformers (2020). Preprint. arXiv:2006.16841
19. F. Locatello et al., Object-centric learning with slot attention. Adv. Neural Inf. Process. Syst. **33**, 11525–11538 (2020)
20. H.W. Kuhn, The Hungarian method for the assignment problem. Nav. Res. Logist. Q. **2**(1-2), 83–97 (1955)
21. F.A. Di Bello, S. Ganguly, E. Gross, M. Kado, M. Pitt, L. Santi, J. Shlomi, Towards a computer vision particle flow. Eur. Phys. J. C. **81**(2), 1–14 (2021)
22. D.W. Zhang, G.J. Burghouts, C.G.M. Snoek, Recurrently predicting hypergraphs (2021). Preprint. arXiv:2106.13919
23. M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R.R. Salakhutdinov, A.J. Smola, Deep sets. Adv. Neural Inf. Process. Syst. 30 (2017)
24. M. Cacciari, G.P. Salam, G. Soyez, The anti-k_t jet clustering algorithm. JHEP **4**, 063 (2008)