









HGPflow: extending hypergraph particle flow to collider event reconstruction

Nilotpal Kakati^{1,a} , Etienne Dreyer¹ , Anna Ivina¹ , Francesco Armando Di Bello² , Lukas Heinrich³ ,
Marumi Kado⁴ , Eilam Gross¹ 

¹ Weizmann Institute of Science, Rehovot, Israel

² INFN and University of Genova, Genoa, Italy

³ Technical University of Munich, Munich, Germany

⁴ Max Planck Institute for Physics, Munich, Germany

Received: 16 January 2025 / Accepted: 17 June 2025
© The Author(s) 2025

Abstract In high energy physics, the ability to reconstruct particles based on their detector signatures is essential for downstream data analyses. A particle reconstruction algorithm based on learning hypergraphs (HGPflow) has previously been explored in the context of single jets. In this paper, we expand the scope to full proton–proton and electron–positron collision events and study reconstruction quality using metrics at the particle, jet, and event levels. Instead of passing entire events through HGPflow, we train it on smaller partitions for scalability and to avoid potential bias from long-range correlations related to the physics process. We demonstrate that this approach is feasible and that on most metrics, HGPflow outperforms both traditional particle flow algorithms and a machine learning-based benchmark model.

1 Introduction

Collider experiments have enabled historic breakthroughs in the field of particle physics and continue to drive progress on fundamental questions. To study the interactions underlying particle collisions, detectors measure outgoing particles via the energy they deposit in a series of “hits” as they traverse concentric arrays of sensors. For general-purpose detectors at the Large Hadron Collider (LHC) [1], tracker and calorimeter hits comprise the bulk of information for pattern recognition algorithms that reconstruct the particles produced in each collision. Low-level reconstruction algorithms are designed to cluster hits in the tracking volume [2,3] and calorimeter [4,5] separately to form tracks and calorime-

ter clusters. High-level *particle flow* algorithms [6–9] then attempt to combine the track and cluster information in an optimal way to form reconstructed objects. Ideally, these particle flow objects mimic the true particles, offering a universal starting point for higher-level reconstruction, including jets, physics observables, and event topology.

In a given detector, the difficulty of reconstructing particles depends on intrinsic factors such as their momentum and type, as well as their local environment. While charged particles are generally well-modeled by track reconstruction, at higher momenta, calorimeter clusters associated with the track can significantly improve the measurement. In contrast, reconstructing long-lived neutral hadrons and photons relies entirely on calorimetry, which is more straightforward for electromagnetic showers than for hadronic showers. Hadronic showers exhibit large spatial fluctuations and dissipate significant fractions of their energy in forms that are invisible to most calorimeters. Moreover, a simple one-to-one relationship between calorimeter clusters and particles is often degraded by limited calorimeter granularity and the effects of noise suppression.

These challenges are further exacerbated in dense environments like hadronic jets, where calorimeter signatures from multiple particles frequently overlap. A substantial component of jet energy resolution thus stems from *confusion* – the difficulty in correctly associating calorimeter clusters with tracks and distinguishing hadronic and electromagnetic showers. Traditional particle flow algorithms address this by using parameterized energy profiles for each track to associate nearby calorimeter cells or to identify their neutral particle origin. However, the high-dimensional nature of the input data and sensitivity to environmental factors motivate a multivariate approach as a solution.

^a e-mail: nilotpal.kakati@weizmann.ac.il (corresponding author)

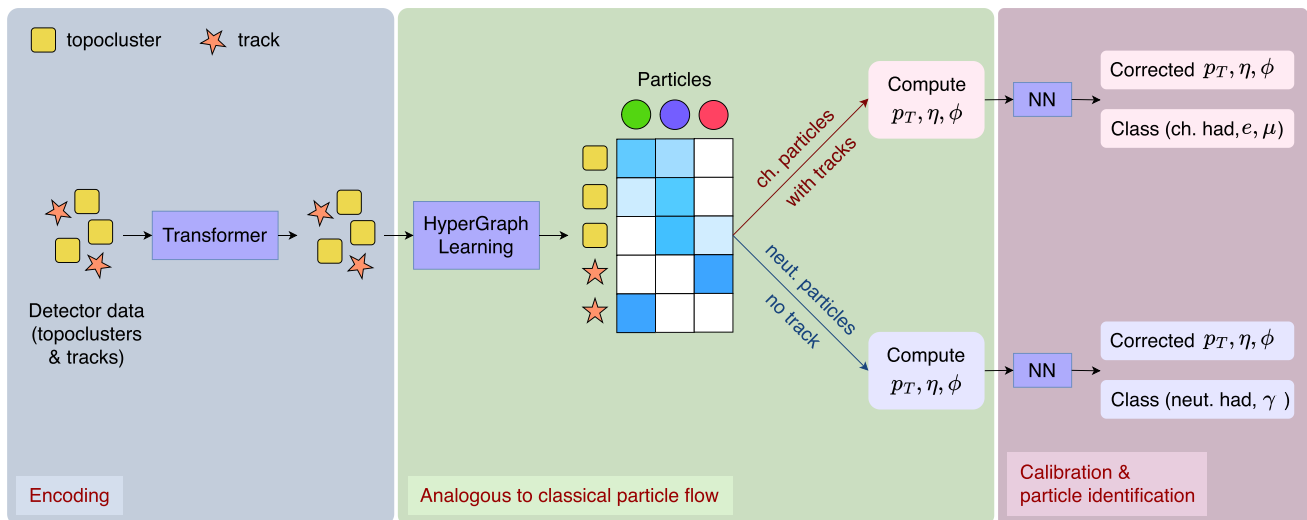


Fig. 1 Schematic showing the three main blocks of the HGPflow algorithm

In recent years, several efforts have explored modernizing the particle flow concept with the application of deep learning (DL) techniques [10–15]. From a DL perspective, particle reconstruction is a prediction task using the features associated with the set of detector hits¹ as an input to predict a ground-truth set of particles, including their class, direction, and momentum. The permutation invariance of the input and output sets motivates the use of graph neural networks [16] and transformers [17], as in various other applications to high energy physics [18–22]. The existing algorithms for particle reconstruction using DL can be compared based on how the set-to-set task is defined in each case.

In object condensation [11, 14], the network learns to cluster detector hits based on parent particle identity in a representation space guided by attractive and repulsive potentials as losses. Additional loss terms supervise the particle properties associated with each hit and encourage the promotion of some hits to serve as centroids in each cluster. The Machine Learning Particle Flow (MLPF) approach [12, 15] is similar in that particle properties are supervised for each hit, but differs in that the cardinality of particles is obtained by classifying hits as either belonging to a valid particle type or else belonging to the “reject” category. The first algorithm proposed in [13] – Transformer Set Prediction Network with Slot Attention (TSPN-SA) – links hits and particles through an attention matrix in an unsupervised manner. Finally, Hypergraph Particle Flow (HGPflow) [13] builds on the idea of a learnable matrix relating the input and output sets but defines this matrix in physical terms and promotes it to a supervised training objective.

¹ Depending on the approach, “hits” can refer to tracks, calorimeter cells, and calorimeter clusters.

In this paper, we extend the work of [13] by applying HGPflow to full collision events and evaluating a variety of performance metrics for different physics processes. Section 2 describes the HGPflow algorithm along with other algorithms used as benchmarks, including MLPF. In Sect. 3, we introduce event partitioning as a strategy to train DL algorithms on full collision events which is computationally efficient and avoids potential bias from long-range physical correlations. We test HGPflow in both an LHC-like scenario, simulated using COCOA [23], and a future e^+e^- collider scenario, simulated using the Compact Linear Collider (CLIC) detector model [24, 25]. The training and testing datasets are described in Sect. 4, the training target and procedure in Sect. 5, and the performance results in Sect. 6.

2 Algorithms

2.1 HGPflow algorithm

The problem of correctly assigning energy deposits in the calorimeter to particles can be naturally formulated as a hypergraph learning task, as first proposed in [13]. A hypergraph is defined by a set of nodes (vertices) $\mathcal{V} = \{V_i; i = 1 \dots N\}$, a set of hyperedges $\mathcal{E} = \{E_a; a = 1 \dots K\}$, and an incidence matrix $I^{N \times K}$ describing the relational structure between the two sets. The entry I_{ia} describes the weight with which node i is a member of hyperedge a . As shown in Fig. 1, the central objective of the HGPflow algorithm is to predict the incidence matrix relating the set of tracks and calorimeter clusters² (\mathcal{V}) to the set of particles, represented by hyperedges (\mathcal{E}). The relationship between tracks

² These relations could also be defined at the level of cells, as in [26].

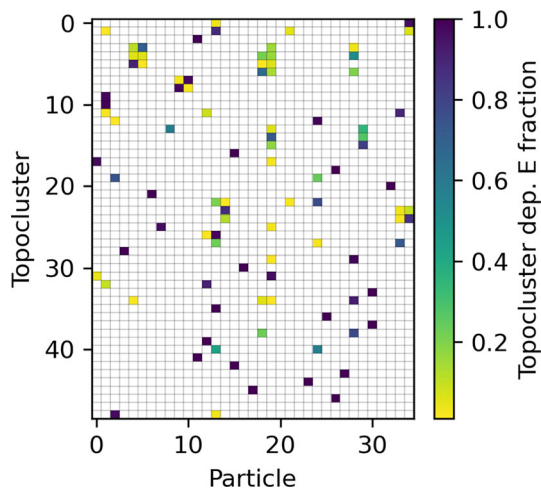


Fig. 2 An example of the energy-based incidence matrix relating calorimeter clusters (“topoclusters”) to particles for a dijet event from the test dataset. Entries for tracks are not shown

and particles is represented by binary entries in the incidence matrix. For calorimeter clusters, the incidence matrix entry I_{ia} denotes the fraction of a cluster i ’s true energy that it received from particle a :

$$I_{ia} = \frac{E_{ia}}{\sum_{\text{particles } b} E_{ib}} = \frac{E_{ia}}{E_i} \tag{1}$$

An example incidence matrix is shown in Fig. 2. The advantage of formulating particle reconstruction like this is that energy deposits in the calorimeter are assigned to particles in a manner that conserves energy by construction. If a fraction f of a given calorimeter cluster is associated with one particle, only $1 - f$ of its energy remains to be associated with other particles. In HGPflow, this is achieved by applying a SoftMax operation ($z_a \rightarrow e^{z_a} / \sum_b e^{z_b}$) to normalize the predicted incidence matrix entries for each cluster. The physical constraint thus serves as an inductive bias, enhancing the performance and interpretability of HGPflow predictions.

The first stage of HGPflow produces a hidden representation of tracks and clusters by first encoding their features in the same high-dimensional space and then updating them in the transformer encoder shown in Fig. 1. Since the true number of particles is unknown *a priori*, the number of candidate hyperedges, K , is set to a value large enough to serve as an upper bound on the number of particles per example. The incidence matrix is initialized such that tracks are uniquely assigned to hyperedges via fixed one-hot encodings. Then, for each cluster, the network predicts a vector of length K obtained through an iterative refinement sequence comprising 12 update blocks, following [27]. As in [13], we employ backpropagation through time, meaning that gradients are only computed for two sequences of 4 consecutive update

blocks each that are randomly selected from the 12 total iterations. The Kullback–Leibler divergence is used as the loss function, with the Hungarian algorithm [28] employed prior to backpropagation to determine the assignment between predicted and target hyperedges that minimizes the loss. Finally, to determine particle cardinality, HGPflow predicts an *indicator score* for each hyperedge determining whether it corresponds to a real particle or not. During inference, hyperedges are dropped unless they contain a track or have an indicator score above a certain threshold.

If the incidence matrix is learned properly, the energy and angular coordinates of each neutral particle can already be approximated based on its association with the energy deposits E_i in the calorimeter, as follows:

$$\hat{E}_a = \sum_i I_{ai} \cdot E_i \tag{2}$$

$$\{\hat{\eta}_a, \hat{\phi}_a\} = \frac{1}{\hat{E}_a} \sum_i I_{ai} \cdot E_i \cdot \{\eta_i, \phi_i\} = \sum_i \tilde{I}_{ai} \cdot \{\eta_i, \phi_i\}, \tag{3}$$

where $\tilde{I}_{ai} = E_{ia} / \hat{E}_a$ denotes a *dual* incidence matrix which is normalized column-wise rather than row-wise. We apply Eqs. 2, 3 to the predicted incidence matrix to construct a set of “proxy” particles. For hyperedges containing a track, we overwrite the proxy properties with those of the track. The creation of proxy particles is shown on the right-hand side of the second stage in Fig. 1.

The accuracy of proxy particles is mainly limited by track resolution and calorimeter characteristics such as granularity, upstream material interactions, electronic noise levels, and sampling fraction. The final objective of HGPflow, shown in the third stage of Fig. 1, is thus to recover the true particle properties as a learned correction to the proxy properties. Each particle is corrected using as input its proxy kinematics, node features weighted by the predicted incidence matrix, and the hidden representation of the corresponding hyperedge. For the datasets considered in this paper, correcting angular coordinates had negligible impact, so only transverse momentum and energy are corrected for charged particles and neutral particles, respectively. Furthermore, the CLIC calorimeter information proved good enough that no corrections were applied to the proxy particles obtained directly from the incidence matrix prediction. Classification of reconstructed particles is also performed at this stage. Hyperedges with an associated track are classified as either charged hadron, electron, or muon, while the rest are split into photon and neutral hadron categories.

Compared to the version in [13], the updated HGPflow model benefits from a more modern DL architecture. The node encoding network utilizes the diffusion transformer (DiT) architecture proposed in [29]. Compared to the original

hypergraph update sequence proposed in [27], we introduced a transformer in the hyperedge update to encode context with self-attention. Transformer operations are performed with FLASHATTENTION-2 [30, 31] for improved memory management and computational speedup. Refer to Sect. 5 for the target definition, training procedure, and hyperparameters for HGPflow.

2.2 MLPF algorithm

As a DL model for particle reconstruction trained on full events, the MLPF algorithm [12, 15] provides the closest comparison to our approach. Calorimeter clusters and tracks make up the set of inputs that MLPF uses to predict the set of target particles. Each target particle is associated with a unique input. A graph convolutional network [32] is used to update the hidden representation of each input based on its local context. This representation is used to predict the class, charge, and four-momentum of the target particles. The network is trained to classify each input to match the class of its assigned particle, while inputs without assignment are classified as unassigned and rejected. The latest version of MLPF uses Locality Sensitive Hashing (LSH) [33] to divide the inputs into bins based on their proximity in a learned feature space. Graph connectivity is restricted to the LSH bins, allowing information exchange that remains computationally scalable. The authors of [15] experimented with a model that operates on the level of calorimeter cells, but ultimately found that using calorimeter clusters is more feasible and leads to better performance. More details about the target definition and model configuration we used for MLPF can be found in Sect. 5.

2.3 PPflow algorithm

To represent traditional reconstruction approaches, we use a parameterized particle flow (PPflow) algorithm as a baseline. The PPflow algorithm is designed to improve jet momentum resolution by combining tracks and calorimeter clusters while avoiding double counting. This is accomplished by deriving templates of the energy distribution from single charged pions in the calorimeter. In jets, where hadronic showers from multiple charged and neutral sources overlap, these pion energy templates can then be subtracted from calorimeter deposits to infer the contributions from neutral particles. Summing the energy associated with the track with the energy of the residual neutral components provides a more optimal use of the calorimeter and tracker information.

2.4 Pandora algorithm

Pandora [9] is a highly sophisticated particle flow algorithm developed for future linear colliders and recently applied to

the CLIC detector [24]. Exploiting the high granularity of the CLIC calorimeter, Pandora clusters cells within cones aligned with the cluster direction, working outward from the innermost calorimeter layer, where track projections are used to seed clusters when present. Clusters without an associated track are identified as photons if they satisfy specific criteria; otherwise, they are considered for merging with neighboring track-associated clusters. Subsequently, clusters associated with a track are iteratively re-clustered to find a configuration where the track momentum is compatible with the associated calorimeter energy. Particle flow objects from Pandora are classified as either photon, electron, muon, charged hadron, or neutral hadron using a simple particle identification scheme. Software compensation is employed in Pandora to enhance jet energy resolution by identifying electromagnetic sub-showers embedded inside of hadronic showers and calibrating each separately [34]. These careful design choices complement the excellent CLIC detector performance to achieve a strong correspondence between particle flow objects from Pandora and truth particles. Performance studies of Pandora in the CLIC context demonstrated an impressive 3.5% jet energy resolution across a wide energy spectrum [35].

3 Event partitioning

Scaling up the task of particle reconstruction from single jet examples used in [13] to full collision events introduces two potential challenges. The first challenge is computational, given the increased dimensionality of the input space. A second challenge arises because training on full events opens the possibility of learning global features generated by the hard-scatter process. In Sects. 3.1 and 3.2, we argue that these challenges can be addressed by splitting events into smaller partitions and propose an event partitioning scheme in Sect. 3.3.

3.1 Memory scaling

Memory requirements for transformer architectures are sensitive to the maximal dimension N of the input nodes, which is in the hundreds for our training samples. Since the incidence matrix has the shape $(N \times K)$, it scales less than quadratically in N , because the upper bound K on the number of target particles is typically less than N . The main memory consumption in HGPflow arises from the transformer encoder (the first stage in Fig. 1), which scales quadratically with N . For relatively clean collision environments, this remains feasible to allocate for a full event when working at the level of tracks and calorimeter clusters. However, for high-multiplicity pp collisions including pileup, a more efficient approach is required. Operating at the level of cells

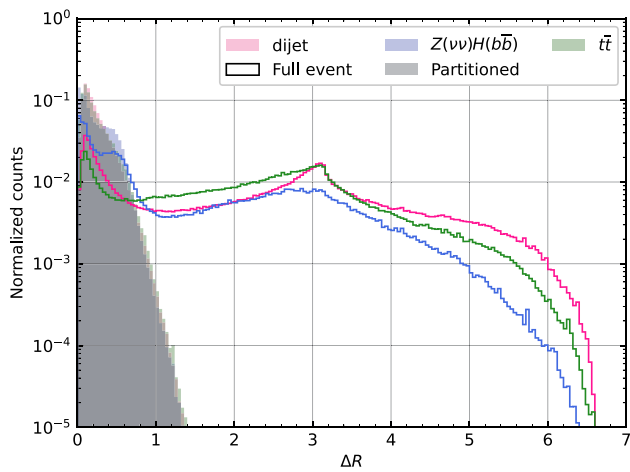


Fig. 3 Histograms of the angular separation between pairs of calorimeter clusters before (empty) and after (filled) partitioning for three different physics processes simulated in COCOA. Each pair enters the histogram weighted by the product of the two energies

instead of clusters can similarly increase N by one to two orders of magnitude, depending on calorimeter granularity.

Although the scalability of transformers has been thoroughly studied with promising alternative solutions such as [36–38], our approach takes a different direction by scaling down the problem instead of scaling up the solution. Partitioning the event and restricting training and inference to the smaller partitions makes the memory overhead tuneably small.

In [15], local sensitivity hashing (LSH) was employed for manageable memory scaling. Although LSH is highly efficient for retrieving similar items in large databases, it offers no guarantee that the resulting clusters are strictly local, which is a requirement for a meaningful incidence matrix. Therefore, we opt for a clustering approach that meets this requirement by operating directly in the angular space of the detector.

3.2 Long-range correlations

Correlations among the energy and spatial coordinates of outgoing particles carry insights about the hard scatter process, including its topology, momentum balance, and interaction type. Figure 3 shows correlations in the angular distribution of calorimeter clusters, weighted by energy, revealing a dependence on physics process. The peak at $\Delta R = \pi$ reveals back-to-back correlation, and is notably reduced for the boosted $Z(\nu\nu)H(b\bar{b})$ events, as expected. These events, on the other hand, feature a bump in the region $0 < \Delta R < 1$ consistent with the opening angle between the decay products.

For particle reconstruction to function universally, it is best approached as a local problem, agnostic to long-range corre-

lations. If reconstructed outputs provide information that is biased toward the physics process used to derive them, they no longer serve their purpose as generic inputs for high-level analyses. Since DL models naturally exploit all available features, they are free to use both local and global correlations present in the training data. Thus, when using DL for particle reconstruction, special care is needed to avoid inadvertently learning non-local, process-specific features.

Event partitioning is a straightforward way to remove long-range correlations. In this approach, the partition determines the local scope of the question and, hence, of the answer. Figure 3 demonstrates that correlations beyond $\Delta R \simeq 1$ are removed after event partitioning. In Sect. 6.6 we show that long-range correlations can get encoded in a DL model during training and influence its physics performance.

3.3 Mean shift clustering

Our event partitioning approach is illustrated in Fig. 4. In the first step, an algorithm divides the full set of tracks and calorimeter clusters into smaller partitions. Reconstruction is then performed for each partition in parallel. Finally, the predictions from each partition are merged, resulting in a set of reconstructed particles for the full event.

The choice of algorithm to perform the event partitioning in this paper is guided by a few desired properties. First, a unique assignment of *all* detector tracks and calorimeter clusters to disjoint partitions is necessary to avoid the need for overlap removal on the set of predicted particles. The algorithm also needs to be suitable for a variable number of input nodes and able to cope with significant variations in local density. While jet clustering algorithms based on sequential recombination are a natural choice for collision events, they are not designed to create clusters outside of jets. A handful of popular, generic algorithms also seem less than suitable: k -means clustering requires assuming the number (k) of clusters beforehand, DBScan [39] is intended for clusters of similar high density, and OPTICS [40] can leave some nodes clustered.

The Mean Shift (MS) [41,42] clustering algorithm is a straightforward choice that fulfills the above criteria. The MS algorithm works by iteratively shifting input nodes in the direction of higher data density. Given a distance metric, the process starts by defining a neighborhood around each node of size h called the *bandwidth*. After computing the centroid of each neighborhood as the mean over the nodes inside, the reference node is shifted in the direction of the centroid. This process repeats until convergence when centroids coalesce into stationary peaks of high density. Finally, the bandwidth is used to determine whether nodes belong to one cluster or another. This procedure requires minimal assumptions about the number of clusters or their density distribution. Although

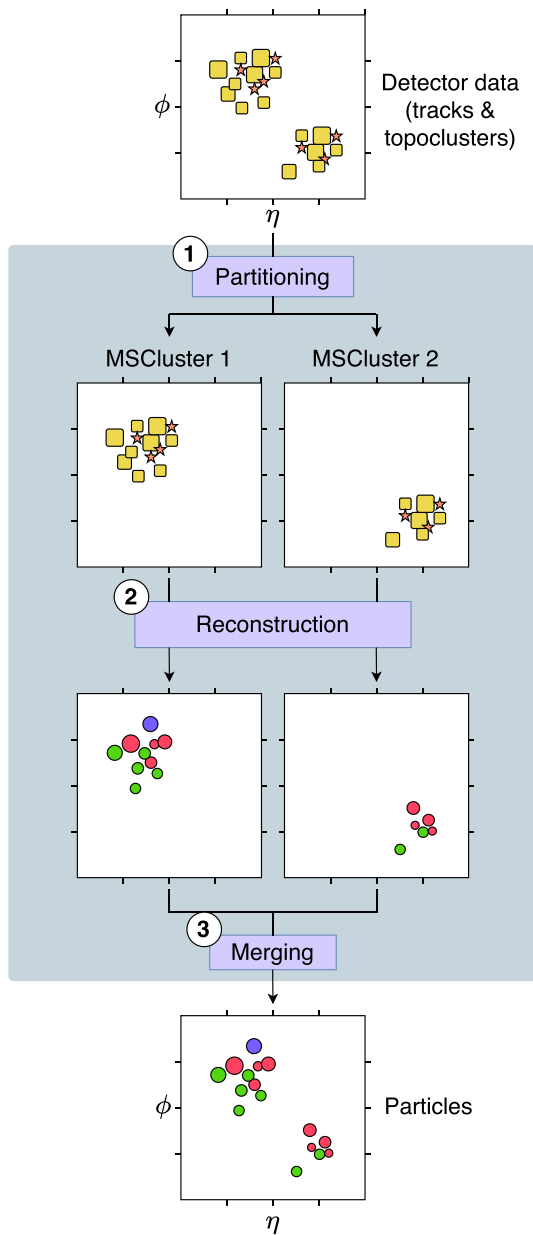


Fig. 4 The event partitioning procedure using Mean Shift clustering in the η - ϕ plane

it works best for spherical clusters, it behaves well for nodes in relatively sparse intermediate regions.

We used a modified version of the SCIKIT-LEARN [43] implementation of Mean Shift, with the default Cartesian distance metric replaced with ΔR to take into account the periodicity in ϕ . For clusters, the barycenter η and ϕ are used, whereas, for tracks the extrapolated position at the calorimeter face is used. The bandwidth h is set to 0.4 and 0.3 in our results on pp and e^+e^- events, respectively. This configuration gave good results in limited studies but should be properly optimized in future work.

The partitioning approach introduces edge effects when two or more energy deposits from a single particle get split

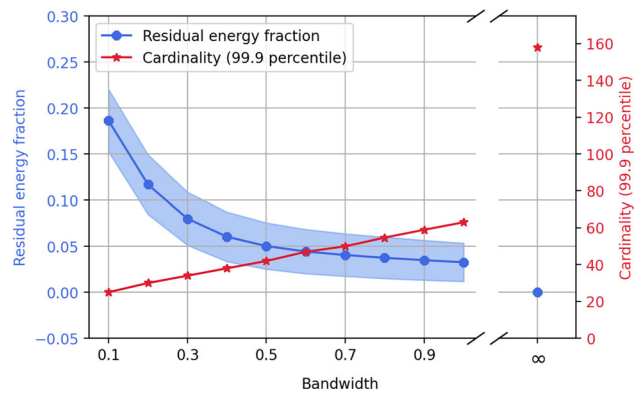


Fig. 5 The impact of bandwidth on residual energy fraction per event and particle cardinality per MS cluster. The band on residual energy indicates the interquartile range around the median, denoted by the markers

among multiple partitions. In such cases, each partition holds partial information about the same particle. If the target in each MS cluster is the original particle, double counting will arise once predictions are merged. In practice, this predominantly affects hadrons within jets. We deal with this effect in two separate cases, as follows. Case 1: if the split particle has no associated track, we create two separate target particles that have their kinematics recomputed based on the fraction of energy in each MS cluster. If both targets are successfully reconstructed, the original energy will be recovered upon merging, although there will be two neutral particles rather than one. Case 2: if the split particle has a track, we keep the original target particle only in the MS cluster that contains the track. Energy deposits from the original particle that end up in other MS clusters are then associated with a modified target particle called a *residual* particle. Since they have actual energy deposits, residual particles participate in the energy assignment of the target incidence matrix. However, residual particles are assigned a target indicator score of zero, training HGPflow to reject them along with non-existent hyperedges.

Figure 5 shows that the fraction of energy attributed to residual particles decreases for larger values of bandwidth. Conversely, the upper bound on the number of particles per MS cluster is directly proportional to the bandwidth. These trends demonstrate the interplay between edge effects and computational cost when deciding the characteristic size of event partitions.

4 Datasets

4.1 COCOA samples

Our main datasets are pp collisions at $\sqrt{s} = 13$ TeV generated and showered with PYTHIA8 [45]. The generated events include initial and final state radiation and underlying event, but no pileup. The detector response is simu-

lated using COCOA [23] – a nearly hermetic calorimeter model based on GEANT4 [46–48]. We use the default configuration of the COCOA calorimeter, comprising three electromagnetic calorimeter (ECAL) layers followed by three hadronic calorimeter (HCAL) layers. In each layer, cells are defined by segmenting both η and ϕ into D divisions, where $D = [256, 256, 128]$ for the three ECAL layers and $D = [64, 64, 32]$ for the HCAL.

Charged particle tracks are obtained by solving the helix equations in an axial 3.8T magnetic field, and material interactions are emulated through a parametric smearing applied to the η , ϕ , and q/p track parameters. Tracks are required to reach the calorimeter and to originate from a vertex with a transverse radius smaller than 75 mm and a longitudinal displacement less than 420 mm in order to be reconstructed. For our purposes, we remove all material from the tracking volume in the simulation, up until the iron solenoid layers preceding the calorimeter. Finally, a built-in topological clustering algorithm builds clusters (referred to as “topoclusters”) by considering the energy-over-noise values of cells for three successive thresholds.

Three different physics hard scatter processes are used for our results: inclusive dijet, inclusive $t\bar{t}$, and boosted $Z(\nu\nu)H(b\bar{b})$ production. The number of events in each sample is listed in Table 1. In the Higgs sample, the Z boson is forced to decay to neutrinos, and a transverse momentum of the Higgs boson of at least 400 GeV is required. Single-pion samples are also simulated in COCOA to evaluate performance in a standard particle flow setup. The training of the ML algorithms is performed solely on the dijet sample while the other processes are for performance evaluation. Figure 6 shows one event display for each of the three physics processes simulated in COCOA.

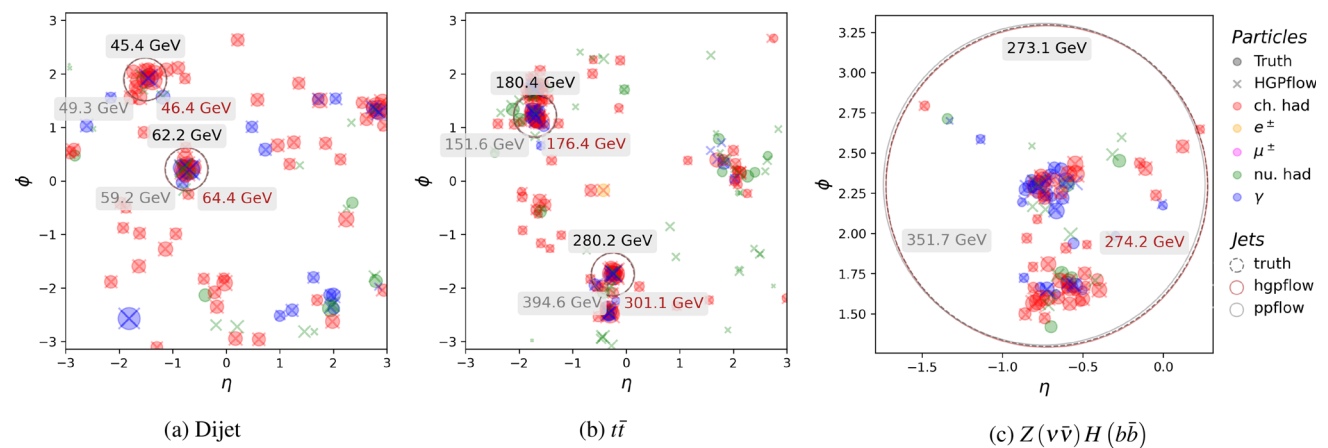


Fig. 6 Examples of full events showing angular coordinates and classes of truth and predicted particles for three different processes. The marker size is proportional to the logarithm of particle p_T . Circles indicate the location and p_T values of leading jets formed from truth

Table 1 Samples used for training and performance testing

Detector	Process	Statistics		
		Train	Val	Test
COCOA	$p^+p^+ \rightarrow q\bar{q}$	250k	10k	35k
	Single π^+	–	–	30k / p_T bin
	$p^+p^+ \rightarrow t\bar{t}$	–	–	20k
	$p^+p^+ \rightarrow Z(\nu\bar{\nu})H(b\bar{b})$	–	–	10k
CLIC	$e^+e^- \rightarrow q\bar{q}$	1M	5k	20k

Some key characteristics of the COCOA dijet events used to train HGPflow are shown in Fig. 7. First, Fig. 7a shows how many particles, tracks, and topoclusters are present in the events, as well as the number of partitions resulting from the MS clustering. The particles fulfill the requirements for target particles in COCOA defined in Table 2. Figures 7b and 7c can be interpreted as the number of entries per row and per column of the incidence matrix, respectively. The entries are separated into different histograms based on the fraction of total deposited energy they carry. While it is true that most topoclusters have at least one particle accounting for over 40% of its energy, long tails indicate that many additional particles are typically present, contributing smaller fractions of energy. The same is true for particles, with some particles dispersing their energy into as many as ten different topoclusters each containing 5–10% of the energy.

4.2 CLIC samples

To study the possible extension of HGPflow to high-granularity calorimeters, we use a sample of simulated dijet events produced in $\sqrt{s} = 380$ GeV e^+e^- collisions at the

particles, HGPflow predictions, and PPflow objects using the anti- k_T algorithm [44]. In **c** the jet radius parameter is $R = 1.0$ whereas **a**, **b** have $R = 0.4$

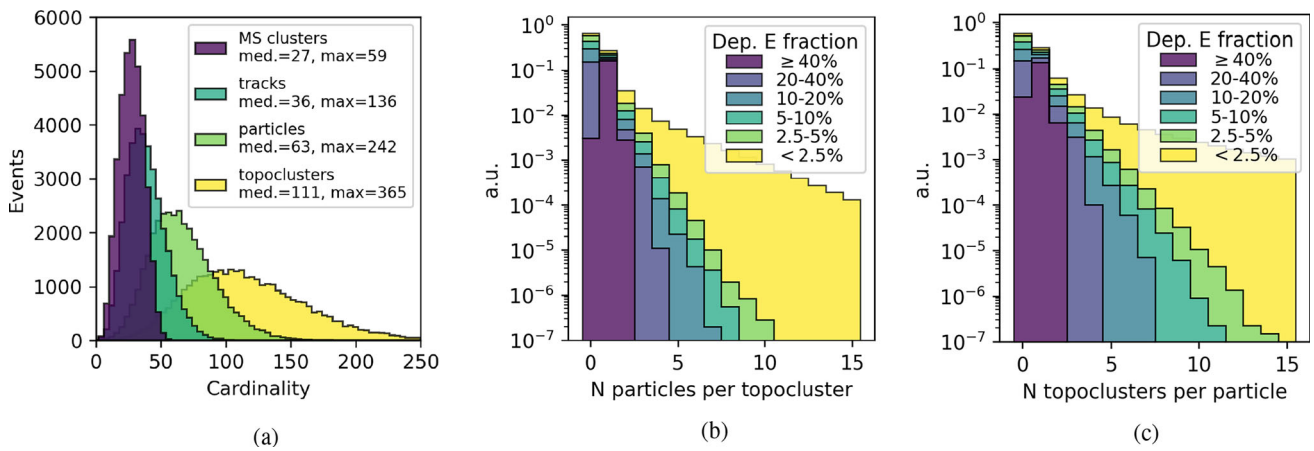


Fig. 7 Cardinality and energy sharing characteristics in dijet events simulated in COCOA. **a** The number of objects in each event, where particles follow the target definition for COCOA. **b, c** Show the num-

ber of particles associated with each topocluster and vice versa. The different histograms denote fractions of total deposited energy and are stacked

Table 2 Definitions of the ground truth and the training target for particles in the COCOA and CLIC datasets. Particles that interact with the detector are defined as those that are associated with a track or an energy deposit in the calorimeter

	COCOA Truth	Target	CLIC Truth	Target
ν veto	✓	✓	✓	✓
$p_T(E)$ [GeV]	> 1	> 1	(> 0.01)	(> 0.10)
$ \eta $	< 3	< 3	< 4	< 4
Generator status	1	1	1	0, 1, 2
Interacts with detector	–	✓	–	✓

Compact Linear Collider (CLIC). The sample is publicly available at [49] where it is part of the larger training sample used to train MLPF in [15]. The samples are produced using PYTHIA8 and the CLIC detector simulation [24,25], which is based on GEANT4 and the KEY4HEP software [50].

Compared to pp collisions at the LHC, e^+e^- collisions at CLIC are cleaner, having neither pileup nor underlying event. The calorimeter of the CLIC detector is also highly granular, comprising 40 ECAL layers with a cell pitch of $5 \times 5 \text{ mm}^2$ and 60 HCAL layers with a $30 \times 30 \text{ mm}^2$ cell pitch. The calorimeters are surrounded by a solenoid magnet providing a field strength of 4T. Unlike COCOA, the CLIC detector simulation includes tracking material and realistic track reconstruction. Each event contains particle flow objects reconstructed by the Pandora algorithm [51]. Calorimeter clusters are provided with a predefined calibration applied to their energy.

5 Target definition and training

5.1 Fiducial particle definition

Defining which particles comprise the target is a key design choice when approaching particle reconstruction as a DL task. In actual detectors, a significant amount of material lies upstream of the calorimeters in the tracking volume. Interactions with this material can cause non-trivial alterations to the set of stable particles leaving the interaction, particularly in the cases of photon conversions and electron bremsstrahlung. A fiducial electron definition in the form of angular criteria is necessary to decide when radiated energy should be merged with an electron and when it should be considered as a separate photon in the target set. Similarly, depending on the angular separation of two conversion electrons entering the calorimeter, a fiducial photon definition would distinguish cases where the resulting clusters should be considered merged or resolved, relative to detector resolution.

Given the detector dependence of designing fiducial criteria for electrons and photons, we have decided to avoid a full treatment in this paper, focusing instead on the feasibility of reconstructing full events. In COCOA, our choice is easily realized by the removal of tracking material, such that upstream interactions only occur in the iron layer immediately preceding the calorimeter. For CLIC, the situation is more nuanced, since the presence of tracking material causes significant rates of particles to be created upstream of the calorimeter. These particles have a generator status code of 0, easily distinguishing them from stable particles (status code 1) and unstable particles (status code 2) provided by PYTHIA. When an interaction marks the end of a neutral parent particle trajectory (for instance, in photon conversion), it will have no associated energy deposit in the detector. It

is problematic to train a physically-motivated model such as HGPflow to reconstruct particles that do not leave any trace in the detector. On the other hand, it is incorrect to simply remove such particles from the ground truth.

To resolve this conflict, we propose a practical compromise that separates the definition of the ground truth and the definition of the target used in the training objective, as defined in Table 2. This separation ensures that performance evaluation is based on a clear standard while allowing the freedom to design a training target suitable for each DL model.

The target definition we use to train MLPF is identical to the one in [15] except that we also include interacting particles with generator status 0 and 2, as this significantly improved alignment with our ground truth definition. Unlike HGPflow, the training objective in MLPF requires each particle to be associated with a unique track or calorimeter cluster. At low- p_T , neutral particles frequently fail to produce a leading energy contribution to any calorimeter cluster, making such an association ambiguous. For the COCOA dijet events we use for training, roughly 20% of photons and 10% of neutral hadrons fall into this category. In MLPF, such particles are merged kinematically with the nearest suitable target particle so that their energy can be accounted for in aggregate. Charged particles that absorb nearby neutral particles by this definition can thus have target properties that are modified with respect to their associated track and ground truth particle.

5.2 Training

The hyperparameters for the HGPflow model and training are shown in Table 3. HGPflow is written using the PYTORCH library [52]. The learning rate is scheduled using a single cosine decay cycle [53] with warmup. No hyperparameter optimization was performed for HGPflow, while for MLPF we used the model with hyperparameters obtained from the optimization performed in [15]. We use version 1.6.2 of the latest public MLPF code [54]. The first stage of HGPflow was trained for roughly 57 (100) hours on the COCOA (CLIC) sample, corresponding to 100 epochs on a single NVIDIA RTX A6000 graphics processing unit (GPU). The second stage of training progressed at roughly 10 (5) epochs per hour for 30 (18) epochs. During inference, the threshold for the indicator prediction in HGPflow is set to 0.5 (0.65) for COCOA (CLIC).

6 Results

We evaluate performance using particle-level, jet-level, and event-level metrics on the samples defined in Sect. 4.1. For each event, the sets of predicted particles from each partition

Table 3 Hyperparameters used for training HGPflow on the COCOA dataset

Optimizer	AdamW [55]
Learning rate (max)	0.001
Batch size -	
Stage 1	256
Stage 2	512
Number of epochs -	
Stage 1	100
Stage 2	30
Maximum number of hyperedges (K)	60
Embedded dimension	128
Number of iterative refinements	12
Number of trainable parameters	2.1M

are pooled together and compared with the set of truth particles in the full event. The predictions from the other reference algorithms – PPflow, Pandora, or MLPF – are passed through the same performance pipeline as HGPflow. Figure 6 shows truth particles and particles from HGPflow for example dijet, $t\bar{t}$, and boosted $Z(\nu\nu)H(b\bar{b})$ events.

6.1 Dijet sample (COCOA)

Since the HGPflow algorithm was trained exclusively on dijet events, we first investigate its performance on an independent test set of dijet events. To study particle-level performance, we use the Hungarian algorithm [28] to create pairs between the sets of predicted and truth particles. The matching is performed separately for charged and neutral particles using the following cost function:

$$c_{\text{Hung.}} = \sqrt{(\Delta p_T/p_T^{\text{targ}})^2 + \Delta\eta^2 + \Delta\phi^2} \quad (4)$$

In case the two sets have unequal sizes, a subset of particles from the larger set will be unpaired. For neutral particles, such cases of lower or higher cardinality with respect to the set of true particles are measured in terms of efficiency and fake rate, as follows:

$$\epsilon = N(\text{matched})/N(\text{true}) \quad (5)$$

$$f = N(\text{unmatched})/N(\text{pred.}) \quad (6)$$

The cardinality of charged particles, on the other hand, is determined in both HGPflow and MLPF by the number of tracks in the event.

Figure 8 shows reconstruction quality in terms of neutral particle properties. The histograms only include reconstructed neutral particles that are paired with truth neutral particles, so the normalization of the histograms reflects the

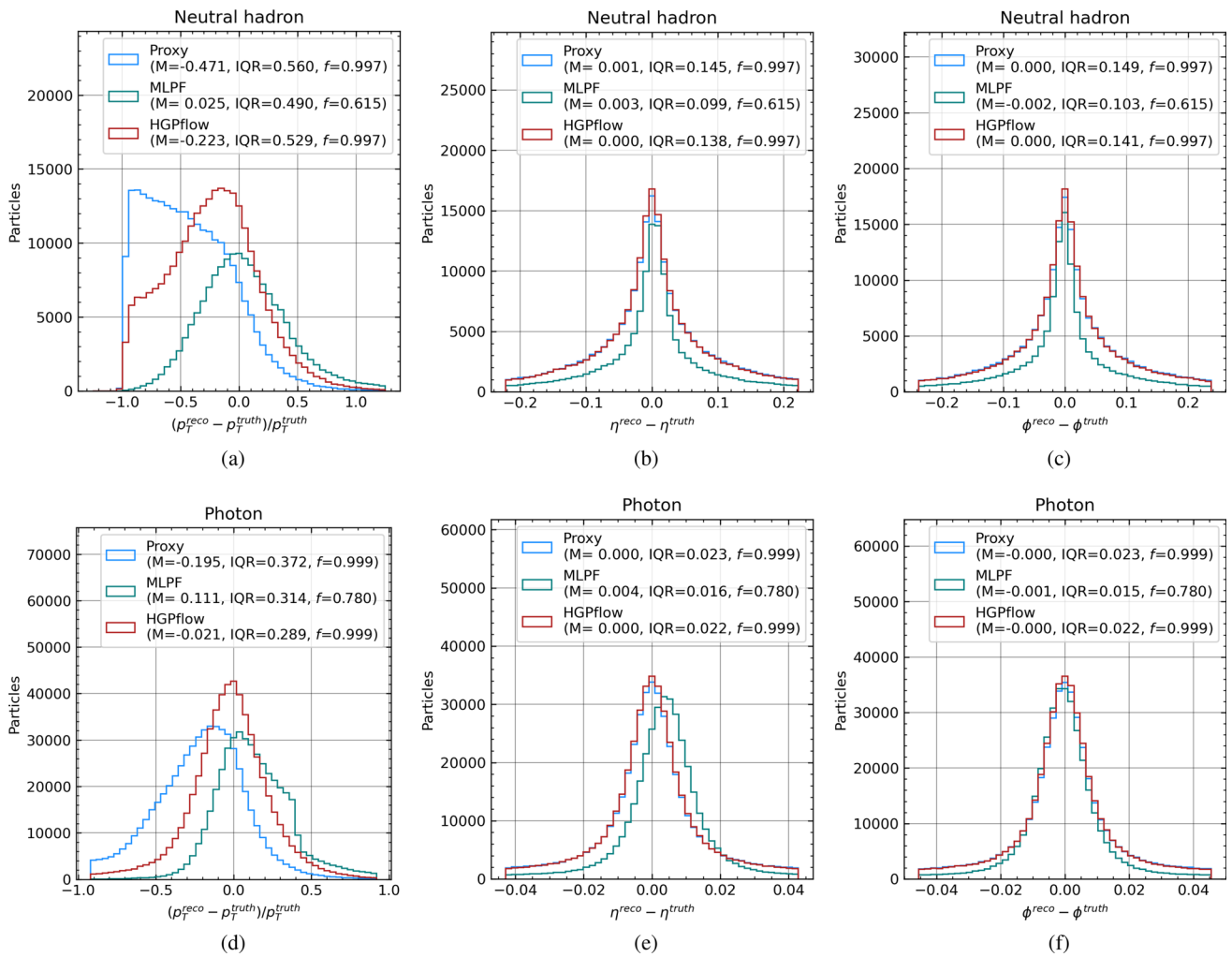


Fig. 8 Residual distributions for p_T , η , and ϕ of the neutral hadrons (a–c) and photons (d–f) matched to reconstructed particles from HGPflow and MLPF. The “Proxy” distribution refers to particles formed using only the incidence matrix and class predictions of HGPflow without

applying any learned correction. The metrics M and IQR refer to the median and interquartile range, respectively, while f is the efficiency of matching truth particles

matching efficiency. It is also important to note that the histograms are dominated by low- p_T particles, due to the steeply falling spectrum. Comparing the HGPflow and proxy (Eqs. 2, 3) histograms reveals the improvement obtained through the second stage of training. For both photons and neutral hadrons, the proxy quantities shown in blue already provide reasonable angular coordinates. For transverse momentum, on the other hand, the importance of a learned correction to the uncalibrated measurement from the calorimeter is clear. This is especially noticeable for neutral hadrons in Fig 8a because of their intrinsically low energy response and reduced topoclustering efficiency, which skew their reconstructed p_T toward zero. Compared to MLPF, HGPflow has a higher efficiency in matching true neutral particles, but it exhibits an overall worse resolution. This apparent trade-off can be understood as arising from the different target def-

initions used by each algorithm, as discussed in Sect. 5.1. While MLPF focuses on the subset of neutral particles that dominate one or more topoclusters, HGPflow also targets the more challenging neutral particles that contribute only sub-dominant energy fractions.

The top panel of Fig. 9 shows the matching efficiency (Eq. 5) for photons and neutral hadrons in bins of truth p_T . On the flip side, the middle of Fig. 9 shows the fake rate, i.e., the rate of unmatched particles predicted, as defined in Eq. 6. Finally, considering predicted particles matched to truth photons or neutral hadrons, the bottom panel in Fig. 9 reports the fraction of correct class prediction among the pairs. Overall, HGPflow reconstructs neutral particles with nearly 100% efficiency. On the other hand, MLPF has lower efficiency but a fake rate of nearly zero. As with Fig. 8, here it is important to note the interplay between these metrics

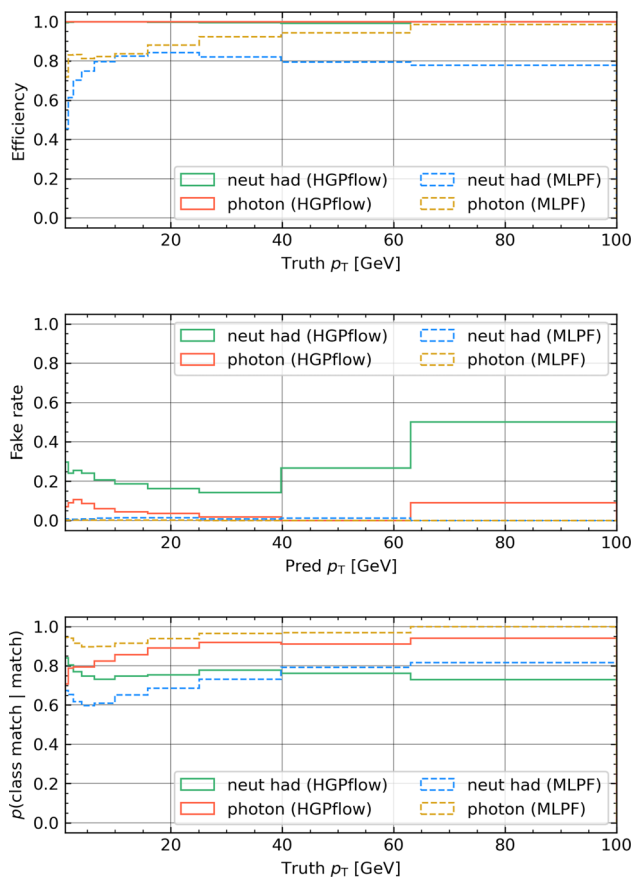


Fig. 9 Top and middle panels show neutral particle efficiency and fake rate for HGPflow and MLPF, as defined in Eqs. 5, 6. The bottom panel displays the fraction of pairs for which the reconstructed particle class matches that of its corresponding truth particle

and the cardinality prediction. For HGPflow, tightening the indicator threshold during inference would lower the fake rate at some expense to efficiency. However, for MLPF, we suspect that the efficiency is limited by the merging procedure described in Sect. 5.1, wherein some neutral particles are absorbed into neighboring particles when creating the training target. Both algorithms perform comparably well at distinguishing photons and neutral hadrons – a task that becomes considerably more difficult for low- p_T neutral particles.

Next, we evaluate the performance of HGPflow at the jet level. We form jets out of truth particles and out of predicted objects using the anti- k_T algorithm [44] implemented in FASTJET [56] with $R = 0.4$. In each event, truth jets are matched to reconstructed jets based on ΔR up to a maximal separation of 0.1.

The distributions of relative jet p_T residuals are shown in Fig. 10. Both DL-based algorithms outperform the parameterized benchmark algorithm PPflow, while HGPflow yields a narrower and less shifted p_T residual distribution compared to MLPF. Figure 10b also shows that HGPflow performs the

best in terms of the angular separation between the predicted and truth jet. Finally, HGPflow models the number of jet constituents significantly better than PPflow and MLPF, shown in Fig. 10c. This can be understood in light of how cardinality is handled in each reconstruction approach. By associating the input and output sets through the incidence matrix, HGPflow is equipped to learn an arbitrary mapping between topoclusters and particles. In PPflow, both topoclusters and tracks function as jet constituents, leading naturally to a cardinality larger than that of truth jets. This double counting can be suppressed in MLPF by assigning a subset of the topoclusters to the reject class. The underestimation observed for MLPF likely stems from its target definition, which is not designed for cases of multiple contributions to a given topocluster and merges neutral particles that are not leading in any topocluster.

6.2 Charged pion sample (COCOA)

The traditional approach to particle flow reconstruction, represented by PPflow, has at its core a model of calorimeter response for charged pions. Therefore, we compare the performance of the DL algorithms with that of PPflow on samples of single π^+ generated uniformly in $|\eta| < 3$ with fixed p_T values ranging from 2 GeV to 80 GeV. These are out-of-distribution examples, as the DL algorithms were not trained on isolated pions, and fewer than 1% of charged pions in the training dataset have $p_T > 20$ GeV. In these cases, the key metric is how efficiently the algorithm predicts a single charged pion without mistaking calorimeter energy fluctuations for additional neutral particles. In practice, this capability allows a particle flow algorithm to incorporate tracks without double-counting.

Figure 11 shows the fraction of single- π^+ examples for which additional neutral particles are predicted, binned in the ratio of total neutral energy to true π^+ energy. Both HGPflow and MLPF outperform the PPflow baseline, yielding significantly larger fractions in the first bin (0–2%). For pions below 10 GeV, HGPflow predicts neutral energy levels below 2% of the pion energy in over 85% of the cases. The task becomes more challenging with increasing p_T , with HGPflow predicting the same $\leq 2\%$ neutral energy fraction for 62% of pions with $p_T = 80$ GeV. Compared to MLPF, HGPflow predicts larger fractions of neutral particle energy for all the p_T bins. This can be understood as the same effect observed in Sect. 6.1, where HGPflow was characterized by higher efficiency and higher fake rate than MLPF in Fig. 9.

The neutral particle energy predicted by HGPflow exhibits a broad peak between 10% and 30% of the π^+ energy. In contrast to PPflow, where any fraction of a calorimeter cluster can be associated with neutral particles, HGPflow is trained to mimic energy assignments seen during training, leading to more discrete predictions. We observed that in over 90%

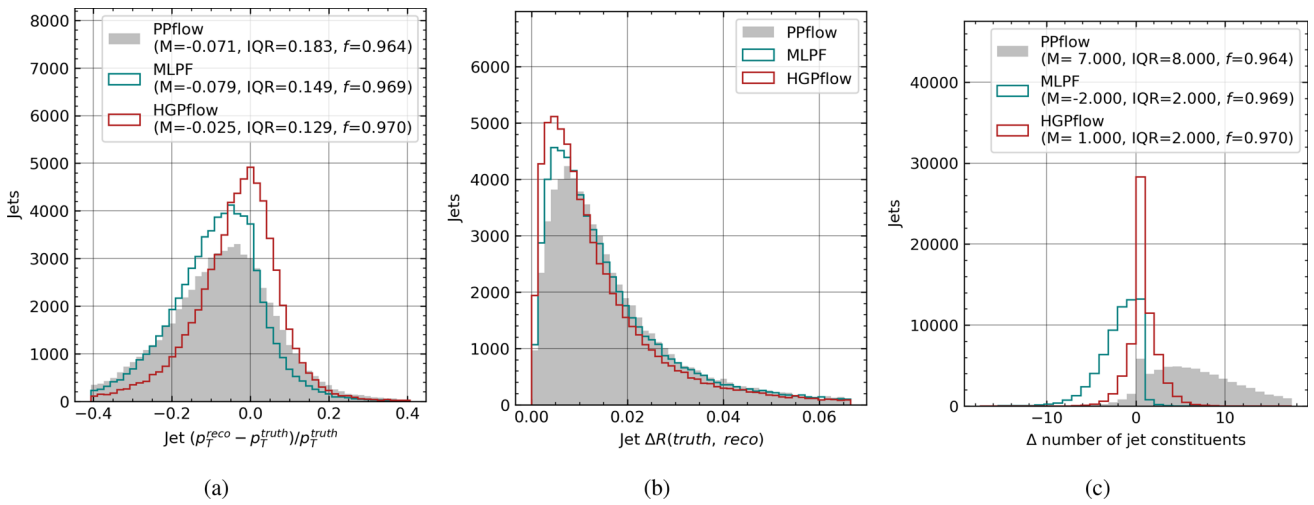


Fig. 10 Matching characteristics between truth and reconstructed jets in terms of **a** jet p_T , **b** angular separation ΔR , and **c** number of constituents N_{const} for dijet events simulated in the COCOA detector. Up to two leading truth jets with $p_T > 10$ GeV are taken from each event and matched to reconstructed jets based on ΔR up to a maximum of 0.1. The metrics M and IQR refer to the median and interquartile range, respectively, while f is the efficiency of matching truth jets

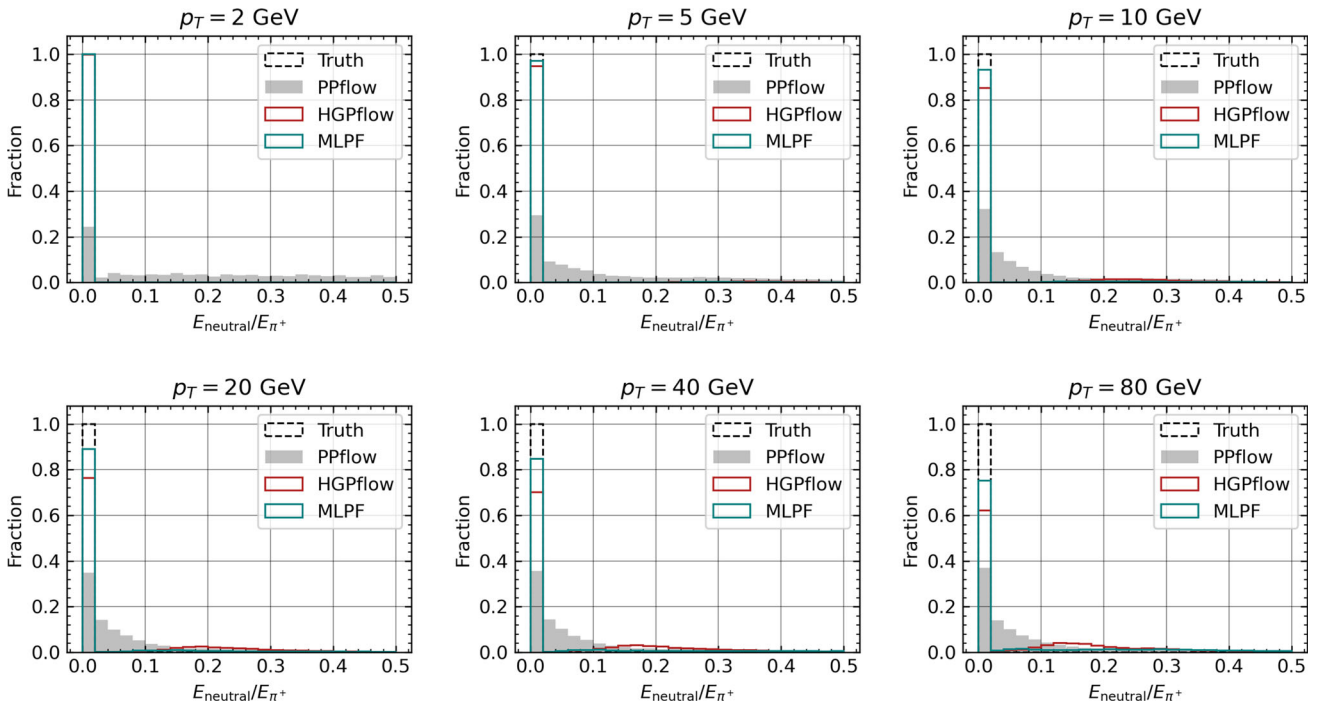


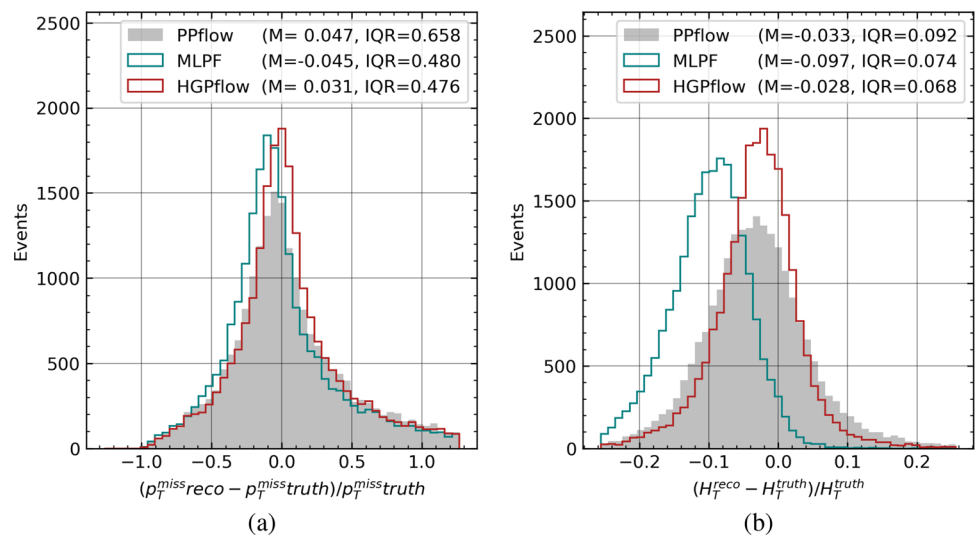
Fig. 11 The ratio of predicted neutral energy (E_{neutral}) relative to charged energy (E_{π^+}) in events containing a single truth π^+ . The subfigure title indicates the transverse momentum of the charged pion in each case. Each distribution is normalized, and the closer to the truth, the better

of the single pion examples, HGPflow predicts two or fewer neutral particles with energy fractions above 2%. Overall, Fig. 11 demonstrates that both DL algorithms successfully extrapolate to an energy association task typically used to tune parametric particle flow models.

6.3 Inclusive $t\bar{t}$ sample (COCOA)

Next, we evaluate performance for event-level observables and boosted topology. In Fig. 12 we check the performance of HGPflow for the $t\bar{t}$ sample in terms of the vector and scalar sums of particle transverse momenta: $p_T^{\text{miss}} = |\sum_i \vec{p}_T^i|$ and $H_T = \sum_i |\vec{p}_T^i|$. The superior residual distribution for

Fig. 12 Residual distributions of **a** the vector p_T^{miss} and **b** scalar H_T sums of transverse momenta in full $t\bar{t}$ events



HGPflow demonstrates not only that its partition-level training can extend to event-level inference, but that this also works for an event topology lying outside the distribution encountered during training.

6.4 Boosted $Z(\nu\nu)H(b\bar{b})$ sample (COCOA)

We test how well the HGPflow training extends to out-of-distribution events containing boosted two-prong Higgs decays. After pooling the partitions for each event, anti- k_T jet clustering is run with $R = 1.0$. Figure 13a shows the mass of the leading large-radius jet formed from PPflow and HGPflow predictions relative to the truth jet in each event. As before, the dijet-trained HGPflow outperforms the benchmark PPflow, demonstrating its ability to generalize to this boosted topology and reconstruct the Higgs mass. To further probe the ability of HGPflow to capture the two-prong nature of the system, we compute the substructure variables C_2 [57] and D_2 [58] using the constituents of the leading large- R jet. The results are shown in Fig. 13b, c, where it can be seen that the substructure distributions of HGPflow and MLPF resemble those of truth jets more closely than PPflow, without either having a clear lead.

6.5 Dijet sample (CLIC) – physics performance

While the various COCOA samples above demonstrate the potential of HGPflow to scenarios resembling that of the LHC (without pileup), here we explore the potential for application to e^+e^- collider experiments, represented by the CLIC detector simulation. Jets are created and matched in the same way as described in Sect. 6.1, but for CLIC we use the generalized k_T algorithm with a radius parameter of 0.7.

Figure 14 shows the performance of HGPflow on an independent set of dijet events in CLIC, with Pandora and MLPF also shown for comparison. In terms of inclusive jet energy

resolution, HGPflow performs slightly better than Pandora, whereas the MLPF residual distribution has a larger scale and width. HGPflow outperforms both alternatives in its predictions of jet angular coordinates and the number of jet constituents.

In Fig. 15, we show how the median and interquartile range (IQR) of the jet energy relative residual distribution vary with truth jet energy. All three algorithms produce distributions centered within roughly $\pm 2\%$, with Pandora tending to overestimate and HGPflow being the most centered overall. For IQR, HGPflow outperforms Pandora by 0.5–1.0% (absolute) for truth jets below 80 GeV, with the two performing similarly within error³ above that. The IQR values of MLPF are similar to HGPflow until 40 GeV but worsen by 0.5–1.5% (absolute) afterward.

6.6 Dijet sample (CLIC) – locality study

To investigate the locality of learned features in both models, we perturb one half of the dijet system and check the impact on the reconstruction of the other half. We start by selecting events from the test dataset with two leading central jets formed from calorimeter clusters using the generalized k_T algorithm with a radius parameter of 0.7. The ΔR between the jets is peaked at π and is required to be above 1.5. In each event, the energies of clusters associated with the subleading jet are reduced by 80%. This perturbation is clearly unphysical, serving merely as a stress test for potential long-range correlations in the model predictions. Finally, HGPflow and MLPF are used to reconstruct both the perturbed events and their original counterparts and compare the differences.⁴

³ The $\pm 1\sigma$ error bands are computed under the assumption of normal distributions using $\sigma_M = 0.93 \cdot \text{IQR} / \sqrt{N}$ and $\sigma_{\text{IQR}} = 1.16 \cdot \text{IQR} / \sqrt{N}$.

⁴ Note that the random seeds are equal in the two evaluations to rule out differences from stochastic elements.

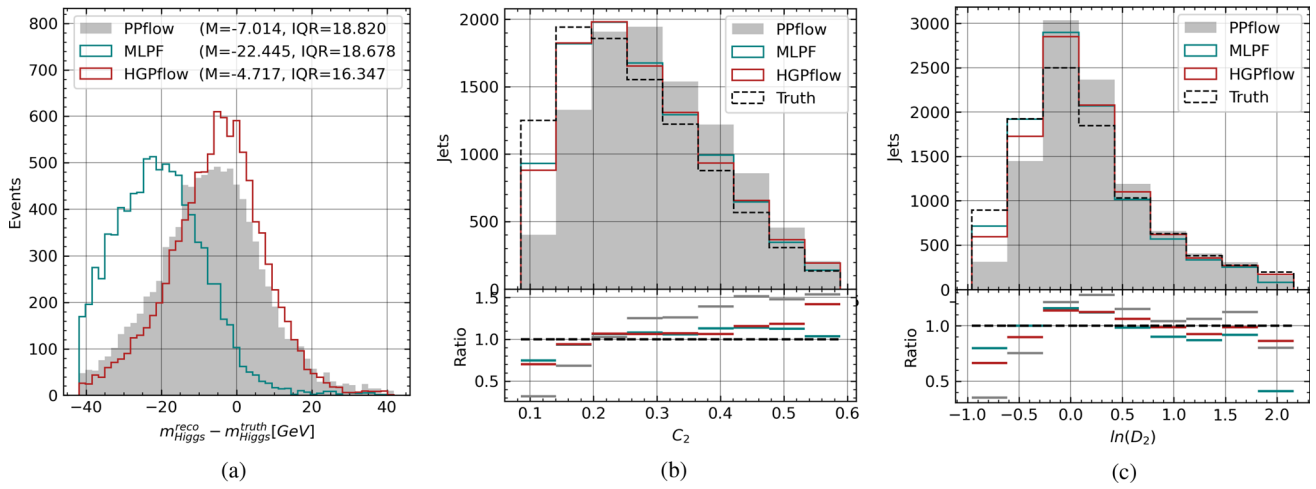


Fig. 13 **a** Residual distribution between the reconstructed Higgs mass, as the mass of the leading large-radius jet, and jet substructure variables, **b** C_2 and **c** $\ln(D_2)$ computed using truth particles and reconstructed particles in $Z(\nu\bar{\nu})H(b\bar{b})$ events

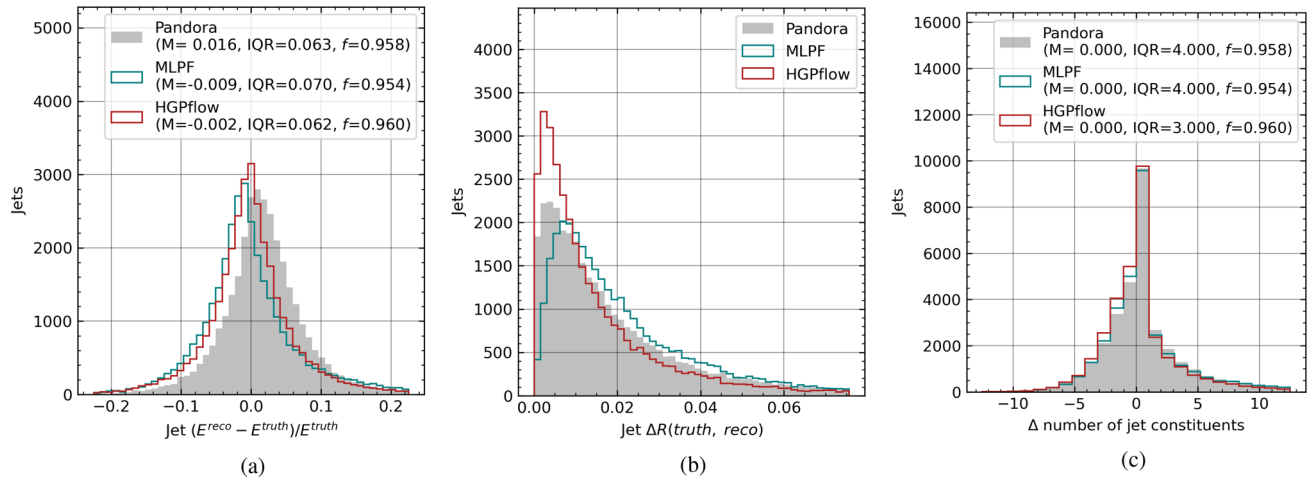


Fig. 14 Matching characteristics between truth and reconstructed jets in terms of **a** jet energy, **b** angular separation ΔR , and **c** number of constituents N_{const} . for dijet events simulated in the CLIC detector. Up to two leading truth jets with $p_T > 10$ GeV are taken from each event

and matched to reconstructed jets based on ΔR up to a maximum of 0.1. The metrics M and IQR refer to the median and interquartile range, respectively, while f is the efficiency of matching truth jets

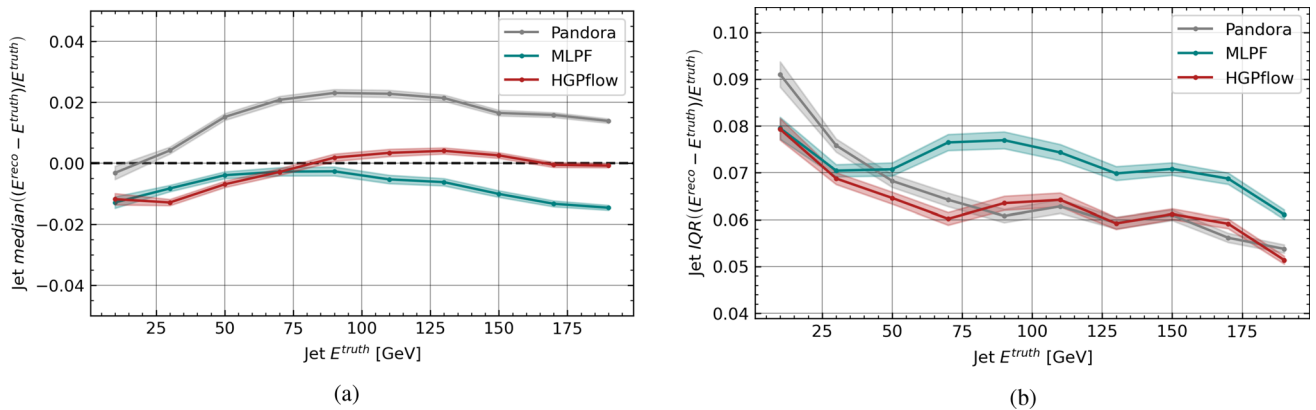


Fig. 15 **a** Median and **b** interquartile range of the jet energy relative residual distribution in bins of truth jet energy. Up to two leading truth jets with $p_T > 10$ GeV are taken from each event and matched to reconstructed jets based on ΔR up to a maximum of 0.1

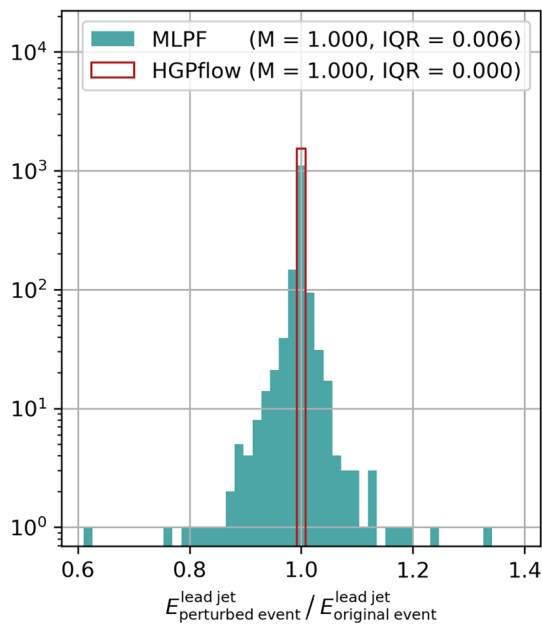


Fig. 16 Test of the locality of the model predictions in dijet events simulated in CLIC. The ratio quantifies the dependence of leading jet energy to subleading jet energy when a perturbation of -80% is applied to the latter

Figure 16 shows the ratio of leading jet energy in the perturbed event relative to the original one. The leading jet should not be affected by the perturbation applied to the subleading jet due to their separation in ΔR . This is realized for HGPflow, where event partitioning has broken this long-range correlation. In comparison, the test reveals a small but definite correlation between the jets reconstructed by MLPF. The limited impact of this correlation can potentially be credited to the approximate locality introduced by LSH.

7 Discussion

Our findings indicate that HGPflow coupled with event partitioning is a promising solution for particle reconstruction at existing LHC detectors and potential future detectors, represented by CLIC. Despite having access to only one partition at a time, HGPflow scales well to full events for different physics processes, as expected of a local reconstruction model. In the following discussion, we summarize the performance comparison with MLPF and note important ways in which our work in this paper can be extended in the future.

The performance of HGPflow and MLPF on the various samples considered is summarized in Table 4. Unlike MLPF, HGPflow is trained to disentangle cases where a single calorimeter cluster contains energy from more than one particle. This edge appears to be most relevant for lower granularity calorimeters, as suggested by comparing the relative performance of HGPflow and MLPF in modeling the num-

ber of jet constituents in COCOA (Fig. 10c) versus CLIC (Fig. 14c).

We have focused on a relatively simple definition of the region on which particle reconstruction is considered to be “local”. More detailed investigation is needed to better quantify this definition and how it impacts a model’s ability to function in a generic way. The potential for bias from other sources should also be considered. For example, the effect of fragmentation on jet energy distribution likely influences our current training by introducing sensitivity to a specific physics model, in this case, that of PYTHIA. This challenge is not unique to particle reconstruction, and may also benefit from domain adaptation techniques [59–61].

Another strategy to suppress dependence on specific physics information comes through the design of the training sample. Training on a dataset that contains diverse physics processes can discourage the network from over-optimizing on any single process, as was done in [15], for example. A potential shortcoming of this approach is that it is not obvious how to optimize the composition to minimize sample dependence, particularly when accounting for physics processes beyond the Standard Model.

Perhaps the most robust approach would be to train on artificial samples created by overlaying multiple single particles. In this case, the type, kinematics, and isolation of particles could be fully configurable while having no dependence on generator modeling or physics process. It would need to be shown, however, that a model trained in this way could generalize well to samples of real physics processes.

At the very least, we envision that a more optimized training dataset will be included in future work, covering a larger kinematic range and augmented with single-particle samples. This would enable the exploration of HGPflow performance for isolated leptons and photons, which have so far been omitted due to a lack of training examples. It would also present an opportunity to develop a fiducial particle definition for electrons and photons that accounts for bremsstrahlung and photon conversions (see Sect. 5.1). A more complete definition might also feature separate classes for nontrivial decay signatures such as neutral pions and tau leptons (see [62]).

The impact of pileup is similarly left to future work. This question is very important for gauging how well HGPflow could perform in realistic collision environments at future runs at the LHC. If the model trained without pileup struggles to generalize well, a dedicated approach to pileup suppression might be incorporated as a classification problem, or at least as a global conditioning variable.

The overall excellent performance of HGPflow suggests that edge effects from event partitioning are manageable. It was noted in Sect. 3.3 that less than 10% of the total energy in dijet events arise from hadronic showers that get split between two MS clusters, i.e., residual particles. However, the relatively large rate of unmatched neutral hadrons

Table 4 Performance summary of all the algorithms on both the COCOA and CLIC datasets in terms of median (M) and interquartile range (IQR) of the residual distributions and jet matching efficiency f

Model	#	Hyperparam	Jet $p_T(E)$ residual (dijet)			p_T^{miss} residual ($t\bar{t}$)		H_T residual ($t\bar{t}$)		m_{Higgs} residual		
			param	optimized	M	IQR	f	M	IQR	M	IQR	M
COCO A	PPFlow	–	–	–0.071	0.183	96.4%	0.047	0.658	–0.033	0.092	–7.014	18.820
	MLPF	6.0M	No	–0.0079	0.149	96.9%	–0.045	0.480	–0.097	0.074	–22.445	18.678
	HGPflow	2.0M	No	–0.025	0.129	97.0%	0.031	0.476	–0.028	0.068	–4.717	16.347
CLIC	Pandora	–	–	(0.016)	(0.063)	95.8%	–	–	–	–	–	–
	MLPF	6.0M	Yes ^a	(–0.009)	(0.070)	95.4%	–	–	–	–	–	–
	HGPflow	5.2M	No	(–0.002)	(0.062)	96.0%	–	–	–	–	–	–

^aPerformed using the ground truth definition of [15]

observed in Fig. 9 might be driven partly by residual particles. Future work could thus further quantify and reduce this impact by optimizing the bandwidth of MS clustering, refining the distance metric, and testing alternative algorithms. Provided a suitable clustering objective (e.g. minimizing cluster size and edge effects simultaneously), a partitioning approach based on DL may prove to be a versatile solution. Additionally, making the partitioning step GPU compatible could lead to significant speed-up and enable running it “on the fly” during model inference.

8 Conclusions

In this paper, we extended the paradigm of particle reconstruction with hypergraph learning (HGPflow) to full collision events. Instead of operating directly on the full event, we used a clustering algorithm to define smaller partitions on which to train and evaluate HGPflow. In addition to reducing memory usage, this approach imposes locality on the features extracted by the model, avoiding potential bias from long-range correlations. We demonstrate that HGPflow provides excellent modeling of particles, jets, and global observables that meets and, in most cases, exceeds that of its parametric and deep learning counterparts. This progress underscores the potential of deep learning to enhance particle reconstruction at current and future colliders while identifying key areas for further exploration in this domain.

Funding NK, ED, AI, and EG are supported by the BSF-NSF Grant No. 2020780 and the Weizmann Institute and MBZUAI Collaboration Grant. LH is supported by the Excellence Cluster ORIGINS, which is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC-2094-390783311.

Code and data availability The code for event partitioning, the HGPflow model, and performance evaluation are available at <https://github.com/nilotpal09/HGPflow> along with the trained model parameters. The code for training MLPF on CLIC data with our modified target and for training on COCOA events is available at <https://github.com/>

[annaivina/particleflow-fork](https://github.com/annaivina/particleflow-fork) along with the trained model parameters in each case. The COCOA samples are available at [63], and the CLIC samples were published previously in [49].

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.
Funded by SCOAP³.

References

1. L. Evans, P. Bryant, LHC machine. *JINST* **3**, S08001 (2008)
2. M. Aaboud et al., Performance of the ATLAS track reconstruction algorithms in dense environments in LHC run 2. *Eur. Phys. J. C* **77**(10), 673 (2017)
3. S. Chatrchyan et al., Description and performance of track and primary-vertex reconstruction with the CMS tracker. *JINST* **9**(10), P10009 (2014)
4. 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)
5. A. Hayrapetyan et al., Performance of the CMS electromagnetic calorimeter in pp collisions at $\sqrt{s} = 13$ TeV. *JINST* **19**(09), P09004 (2024)
6. D. Buskulic et al., Performance of the ALEPH detector at LEP. *Nucl. Instrum. Methods A* **360**, 481–506 (1995)
7. ATLAS collaboration, Jet reconstruction and performance using particle flow with the ATLAS Detector. *Eur. Phys. J. C* **77**(7), 1–47 (2017)
8. A.M. Sirunyan, CMS Collaboration et al., Particle-flow reconstruction and global event description with the CMS detector. *JINST* **12**(10), P10003 (2017)
9. M.A. Thomson, Particle flow calorimetry and the PandoraPFA algorithm. *Nucl. Instrum. Methods A* **611**, 25–40 (2009)

10. 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)
11. 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)
12. 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)
13. F.A. Di Bello, E. Dreyer, S. Ganguly, E. Gross, L. Heinrich et al., Reconstructing particles in jets using set transformer and hypergraph prediction networks. *Eur. Phys. J. C* **83**(7), 596 (2023)
14. S.R. Qasim, N. Chernyavskaya, J. Kieseler, K. Long, O. Viazlo, M. Pierini, R. Nawaz, End-to-end multi-particle reconstruction in high occupancy imaging calorimeters with graph neural networks. *Eur. Phys. J. C* **82**(8), 753 (2022)
15. J. Pata, E. Wulff, F. Mokhtar, D. Southwick, M. Zhang, M. Girone, J. Duarte, Improved particle-flow event reconstruction with scalable neural networks for current and future particle detectors. *Commun. Phys.* **7**(1), 124 (2024)
16. P.W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi et al., Relational inductive biases, deep learning, and graph networks. (2018). [arXiv:1806.01261](https://arxiv.org/abs/1806.01261)
17. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need. (2023). [arXiv:1706.03762](https://arxiv.org/abs/1706.03762)
18. J. Shlomi, P. Battaglia, J.R. Vlimant, Graph neural networks in particle physics. *Mach. Learn. Sci. Technol* **2**(2), 021001 (2020)
19. G. DeZoort, P. Battaglia, C. Biscarat, J.R. Vlimant, Graph neural networks at the large hadron collider. *Nat. Rev. Phys.* **5**(5), 281–303 (2023)
20. 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)
21. H. Qu, C. Li, S. Qian. Particle transformer for jet tagging. 2 (2022). [arXiv:2202.03772](https://arxiv.org/abs/2202.03772)
22. T. Finke, M. Krämer, A. Mück, J. Tönshoff, Learning the language of QCD jets with transformers. *JHEP* **06**, 184 (2023)
23. F.A. Di Bello, A. Charkin-Gorbulin, K. Cranmer, E. Dreyer, S. Ganguly et al., Configurable calorimeter simulation for AI applications. *Mach. Learn. Sci. Technol.* **4**(3), 035042 (2023)
24. CLIC Collaboration, CLICdet: the post-CDR CLIC detector model. (2017). <https://cds.cern.ch/record/2254048>
25. D. Arominski et al., A detector for CLIC: main parameters and performance. *12* (2018). [arXiv:1812.07337](https://arxiv.org/abs/1812.07337)
26. N. Kakati, E. Dreyer, E. Gross, Denoising graph super-resolution towards improved collider event reconstruction. 9 (2024). [arXiv:2409.16052](https://arxiv.org/abs/2409.16052)
27. D.W. Zhang, G.J. Burghouts, C.G.M. Snoek, Recurrently predicting hypergraphs. (2021). Preprint [arXiv:2106.13919](https://arxiv.org/abs/2106.13919)
28. H.W. Kuhn, The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **2**(1–2), 83–97 (1955)
29. W.S. Peebles, S. Xie, Scalable diffusion models with transformers, in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2022), pp. 4172–4182
30. T. Dao, D.Y. Fu, S. Ermon, A. Rudra, C. Ré, FlashAttention: fast and memory-efficient exact attention with IO-awareness. *Adv. Neural. Inf. Process. Syst.* **35**, 16344–16359 (2022). [arXiv:2205.14135](https://arxiv.org/abs/2205.14135)
31. T. Dao, Flashattention-2: faster attention with better parallelism and work partitioning. (2023). [arXiv:2307.08691](https://arxiv.org/abs/2307.08691)
32. T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in *International Conference on Learning Representations* (2017)
33. N. Kitaev, L. Kaiser, A. Levskaya, Reformer: the efficient transformer (2020)
34. H.L. Tran, K. Krüger, F. Sefkow, S. Green, J. Marshall, M. Thomson, F. Simon, Software compensation in particle flow reconstruction. *Eur. Phys. J. C* **77**(10), 698 (2017)
35. L. Linssen, A. Miyamoto, M. Stanitzki, H. Weerts, Physics and detectors at CLIC: CLIC conceptual design report. CERN yellow reports: monographs. CERN, Geneva, (CERN-2012-003). (2012). <https://cds.cern.ch/record/1425915>
36. Y. Tay, M. Dehghani, D. Bahri, D. Metzler, Efficient transformers: a survey. *ACM Comput. Surv.* **55**(6), 1–28 (2022)
37. Y. Sun, L. Dong, S. Huang, S. Ma, Y. Xia, J. Xue, J. Wang, F. Wei, Retentive network: a successor to transformer for large language models, July (2023)
38. J. Ding, S. Ma, L. Dong, X. Zhang, S. Huang, W. Wang, N. Zheng, F. Wei, Longnet: scaling transformers to 1,000,000,000 tokens (2023)
39. M. Ester, H.-P. Kriegel, J. Sander, X. Xu et al., A density-based algorithm for discovering clusters in large spatial databases with noise. *kdd* **96**, 226–231 (1996)
40. M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander, OPTICS: ordering points to identify the clustering structure, in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, SIGMOD '99* (Association for Computing Machinery, New York, NY, USA, 1999), pp. 49–60
41. K. Fukunaga, L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theory* **21**(1), 32–40 (1975)
42. D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 603–619 (2002)
43. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion et al., Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
44. M. Cacciari, G.P. Salam, G. Soyez, The anti- k_t jet clustering algorithm. *JHEP* **4**, 063 (2008)
45. C. Bierlich et al., A comprehensive guide to the physics and usage of PYTHIA 8.3. *SciPost Phys. Codebases*, 008 (2022)
46. Geant4 collaboration, Geant4—a simulation toolkit. *Nucl. Instrum. Methods A* **506**, 250–303 (2003)
47. Geant4 collaboration, Geant4 developments and applications. *IEEE Trans. Nucl. Sci.* **53**, 270 (2006)
48. Geant4 Collaboration, Recent developments in Geant4. *Nucl. Instrum. Methods A* **835**, 186–225 (2016)
49. J. Pata, E. Wulff, J. Duarte, F. Mokhtar, M. Zhang, M. Girone, D. Southwick, Simulated datasets for detector and particle flow reconstruction: CLIC detector, August (2023). <https://doi.org/10.5281/zenodo.8260741>
50. G. Ganis, C. Helsens, V. Völkl, Key4hep, a framework for future HEP experiments and its use in FCC. *Eur. Phys. J. Plus* **137**(1), 149 (2022)
51. J.S. Marshall, M.A. Thomson, The pandora software development kit for pattern recognition. *Eur. Phys. J. C* **75**(9), 439 (2015)
52. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan et al., PyTorch: an imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc., 2019), pp. 8024–8035
53. I. Loshchilov, F. Hutter, Sgdr: stochastic gradient descent with warm restarts. (2017). [arXiv:1608.03983](https://arxiv.org/abs/1608.03983)
54. J. Pata, E. Wulff, F. Mokhtar, J. Duarte, A. Tepper, L.S. Rlie, jzata/particleflow: v1.6.2, March (2024). <https://doi.org/10.5281/zenodo.10893930>
55. I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in *International Conference on Learning Representations* (2017)
56. M. Cacciari, G.P. Salam, G. Soyez, FastJet user manual. *Eur. Phys. J. C* **72**, 1896 (2012)
57. A.J. Larkoski, G.P. Salam, J. Thaler, Energy correlation functions for jet substructure. *JHEP* **06**, 108 (2013)

58. A.J. Larkoski, I. Moulton, D. Neill, Power counting to better jet observables. *JHEP* **12**, 009 (2014)
59. CMS Collaboration, Performance of the CNN-based tau identification algorithm with domain adaptation using adversarial machine learning for run 3 (2024)
60. L.R. Sheldon, D.S. Rankin, P. Harris, MACK: mismodeling addressed with contrastive knowledge. *10* (2024). [arXiv:2410.13947](https://arxiv.org/abs/2410.13947)
61. M. Baalouch, M. Defurne, J.-P. Poli, N. Cherrier, Sim-to-real domain adaptation for high energy physics, in *33rd Annual Conference on Neural Information Processing Systems*, 12 (2019). [arXiv:1912.08001](https://arxiv.org/abs/1912.08001)
62. L. Tani, N.-N. Seeba, H. Vanaveski, J. Pata, T. Lange, A unified machine learning approach for reconstructing hadronically decaying tau leptons. *Comput. Phys. Commun.* **307**, 109399 (2025)
63. N. Kakati, E. Dreyer, A. Ivina, F.A. Di Bello, L. Heinrich, M. Kado, E. Gross, Proton–proton collisions and single pions simulated with COCOA, April (2025). <https://doi.org/10.5281/zenodo.15181707>