أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Graph Neural Networks and Minimum Cost Flow for Particle Associations

Abdullah Saeed Asir, Aurora Perego and Felice Pantaleo

abdullah.asiri.1@kaust.edu.sa, aurora.perego@cern.ch, felice.pantaleo@cern.ch

Abstract

This project investigates the problem of linking reconstructed tracks and tracksters within the TICL reconstruction pipeline for the CMS Phase-2 detector. The work explores a combined machine-learning and optimization framework for identifying physically related detector objects and assembling them into consistent particle candidates. Particular attention is given to graph representation, feature preparation, supervised link prediction, global candidate reconstruction, and reproducible evaluation. The developed workflow provides a structured foundation for studying local associations and their effect on event-level reconstruction. It also highlights the challenges introduced by detector complexity, class imbalance, fragmented calorimeter showers, and ambiguous object associations. Overall, the project demonstrates a systematic approach to integrating graph-based learning with global reconstruction techniques and identifies directions for improving the reliability and practical applicability of track and trackster linking in future TICL studies.

Date: August 24, 2026

Code: https://github.com/AbdullahSaeed100/TracksAndTrackstersLinking

Contents

1 Introduction 2

2 Related works 3

2.1 Graph-Linking Reference Implementation . . . 3

2.2 Feature and Preprocessing Reference . . . 3

2.3 Minimum-Cost-Flow Reference . . . 4

3 Methodology 4

3.1 Pipeline Overview . . . 4

3.2 Event Inputs and Object Selection . . . 4

3.3 Joint Event-Graph Construction . . . 5

3.4 Event Split and Leakage-Safe Preprocessing . . . 6

3.5 Edge-Aware Graph Neural Network . . . 6

3.6 Minimum-Cost-Flow Candidate Reconstruction . . . 7

3.7 Frozen Test Protocol . . . 8

4 Results 8

4.1 Sealed Test Evaluation . . . 8

4.2 GNN Edge-Link Classification . . . 8

4.3 Minimum-Cost-Flow Reconstruction . . . 10

4.4 Dependence on Pseudorapidity and Energy . . . 10

5 Discussion 12

5.1 Interpretation of Local Link Classification . . . 12

5.2 From Edge Scores to Global Reconstruction . . . 12

5.3 Dependence on Detector Region and Energy . . . 13

5.4 Overall Assessment . . . 13

6 Limitations and Future Work 13

6.1 Data and Supervision Limitations . . . 14

6.2 Graph and GNN Limitations . . . 14

6.3 MCF and Evaluation Limitations . . . 15

6.4 Deployment and Reproducibility Scope . . . 15

6.5 Prioritized Next Steps . . . 15

7 Conclusion 16

References 17

1 Introduction

The High-Luminosity Large Hadron Collider will expose the CMS experiment to substantially more complex collision environments than those of the present detector. As part of the Phase-2 upgrade, the High-Granularity Calorimeter (HGCAL) will provide fine spatial, energy, and timing measurements of particle showers in the CMS endcaps [**cms_hgcal_tdr**]. Exploiting this information requires reconstruction algorithms that can separate nearby showers, combine their detector measurements, and associate calorimeter activity with charged particles reconstructed by the silicon tracker.

The Iterative CLustering (TICL) framework organizes HGCAL reconstruction into modular stages. It processes layer clusters and produces three-dimensional shower objects called *tracksters*, together with reconstructed particle properties and identification information [**cms_daq_hlt_tdr**]. A charged particle may also have a *track*, representing its trajectory through the tracking detector and its extrapolation to HGCAL. The reconstruction problem studied in this project is the linking stage: determining which tracks should be associated with which tracksters and which tracksters should be grouped as parts of the same physical shower. These two tasks are referred to as track-to-trackster (trk–ts) and trackster-to-trackster (ts–ts) linking.

This linking problem is difficult for both physical and computational reasons. One particle can produce multiple spatially related tracksters, showers from nearby particles can overlap, and timing or energy measurements may be absent or uncertain. Candidate generation must remain sufficiently inclusive to retain true associations, but this produces many more false candidate edges than true ones, especially for ts–ts linking. Moreover, independently accepting every locally plausible edge can produce an incompatible event-level assignment. An effective solution must therefore learn local compatibility while also enforcing global reconstruction constraints.

This project addresses the problem with a two-stage graph-based approach. First, tracks and tracksters are represented as nodes in a directed candidate graph. Physics-motivated geometric requirements define possible trk–ts and ts–ts edges, and node and edge attributes encode position, kinematics, energy, and timing information. A graph neural network (GNN) then uses edge-aware message passing to assign one raw compatibility logit to every candidate edge. Second, the logits are converted into integer link costs and supplied to a Minimum Cost Flow (MCF) network. The flow optimization selects a globally constrained combination of links and decodes the solution into charged and neutral reconstructed candidates.

The main contributions of the project are:

1. an event-level graph-construction and truth-labelling pipeline for joint trk–ts and ts–ts association;
2. a training-only feature-preprocessing procedure and a reproducible event-level train–validation–test protocol that prevents information leakage between collision events;
3. an edge-aware GNN trained with a focal objective to address the severe imbalance between true and false candidate links while retaining the complete candidate graph;
4. an MCF formulation that converts learned edge logits into global charged and neutral candidate reconstruction; and
5. a sealed test evaluation that applies the frozen model, classification thresholds, and MCF parameters without further training or tuning.

The remainder of this report presents the scientific and computational context of the linking problem, the construction and training of the GNN, the MCF formulation, and the final held-out test results. It then discusses the strengths and limitations of the combined approach and identifies

directions for improving global TICL reconstruction.

2 Related works

The implementation developed in this project builds on three code references that address different parts of the reconstruction chain. The `Graph_Linking` repository provides the principal graph-learning foundation, the `prepare_data.py` script provides a physics-oriented reference for constructing track and trackster features, and the `trackAndCandidatesLinking WithMCFandNN.ipynb` notebook provides the reference formulation of the Minimum Cost Flow reconstruction. The present work combines and extends these components into one reproducible pipeline for joint trk–ts and ts–ts linking.

2.1 Graph-Linking Reference Implementation

The `Graph_Linking` repository [1] is the main reference for the graph neural network stage. Its code demonstrates how CMS ROOT branches can be processed into PyTorch Geometric event graphs, with tracks and tracksters represented in a shared node table and geometrically compatible track-to-trackster pairs represented as directed edges. It also provides the original structure of the edge-aware attention block, the stacked graph-convolution architecture, the edge classifier, and the associated training and evaluation utilities.

Several parts of the submitted implementation retain this overall design. Node and edge inputs are embedded into a common hidden space, three edge-aware graph blocks update the node representations, and the final classifier combines the source node, target node, learned edge representation, and original edge attributes. This reference therefore established the architectural starting point and the basic interface in which the network assigns a score to every candidate link.

The present project extends that reference in several important ways. The event graph contains both trk–ts and directed inner-to-outer ts–ts candidates, rather than treating track-to-trackster association alone. Positive labels require the two endpoints to share the appropriate simulated-particle association, including a same-particle requirement for trk–ts edges. The network returns raw logits, the two edge types are masked and optimized as separate tasks, and the complete training lifecycle uses a frozen event-level split, training-only preprocessing statistics, structured checkpoints, and content fingerprints. These changes make the GNN output suitable for the later MCF cost construction and for a sealed final test evaluation.

2.2 Feature and Preprocessing Reference

The `prepare_data.py` script from the `trackLinking` repository [2] was used as the main reference for the physical quantities supplied to the learning problem. The script reads the track, trackster, simulation, and association branches needed for linking. Its track-to-trackster and trackster-to-trackster feature construction includes angular coordinates, momentum or energy, timing measurements, timing uncertainties, spatial distances, and time-of-flight-corrected timing differences. It also demonstrates the association-score and shared-energy requirements used to identify relevant reconstructed tracksters.

Those ingredients guided the choice of node and edge information in the submitted graph representation. The final implementation reorganizes them into a typed node table and a compact set of genuine pairwise edge quantities. The preprocessing is implemented separately and is fitted only on the training partition: selected variables are clipped or transformed, missing timing values receive explicit treatment, continuous quantities are standardized, and a timing-validity indicator is appended to the edge attributes. Consequently, the reference script supplies the physical feature

basis, while the submitted pipeline defines its own leakage-safe statistical transformations and unified GNN input schema.

2.3 Minimum-Cost-Flow Reference

The notebook `trackAndCandidatesLinkingWithMCFandNN.ipynb` [3] is the direct reference for the global reconstruction stage. It demonstrates how learned track-to-trackster and trackster-to-trackster scores can be converted into integer costs and inserted into an OR-Tools flow network. The network contains a source, track nodes, separate input and output nodes for each trackster, and a sink. Capacity, initialization, neutral, and multiple-use costs control how flow traverses the network. After optimization, positive-flow paths are decoded into charged and neutral candidates, and overlapping candidates are resolved using their trackster content, calorimeter energy, and track momentum. The notebook also provides the basis for truth matching and for efficiency, purity, fake-rate, merge-rate, and structural-consistency measurements.

The submitted implementation extracts this exploratory notebook logic into a reusable module. It maps graph-local indices back to the original ROOT objects, evaluates the frozen focal GNN once per selected event, converts its raw logits to edge-type-specific costs, and solves both detector endcaps. The chosen MCF parameters are frozen before the test evaluation, and the resulting candidates, metrics, curves, and reproducibility fingerprints are stored as a sealed test artifact.

Together, the three references define the starting points for data preparation, learned edge scoring, and global optimization. The contribution of the present project is their integration and revision into a single joint trk–ts/ts–ts workflow with consistent truth definitions, reproducible training, and a strictly separated final test stage.

3 Methodology

3.1 Pipeline Overview

The reconstruction method combines a local learned linking stage with a global optimization stage. Each detector event is first converted into a heterogeneous graph containing track and trackster nodes. Geometrically compatible trk–ts and ts–ts pairs form the directed candidate edges. An edge-aware graph neural network (GNN) assigns one raw logit to every candidate, and those logits are then converted into costs for a Minimum Cost Flow (MCF) network. Solving the flow problem assembles the individually scored links into mutually compatible charged and neutral particle candidates. Figure 1 summarizes this complete workflow.

The implementation is divided into two controlled phases. During model development, preprocessing statistics are fitted, the GNN checkpoint is selected, probability thresholds are obtained, and the MCF parameters are fixed using only the training and validation partitions. The test partition is then opened once for inference and evaluation; it is not used for fitting, threshold selection, or MCF tuning.

3.2 Event Inputs and Object Selection

The input is read from the TICL ROOT output with `uproot`. Four groups of information are required: reconstructed tracks, reconstructed tracksters, simulated tracksters, and the reconstructed-to-simulated association tables. The track branches provide the extrapolated HGCAL position and direction, momentum, hit count, quality, and MTD timing information. The trackster branches provide the barycenter position, energy, transverse momentum, electromagnetic energy, and timing quantities. The association branches are used only to form supervised labels and evaluation truth.

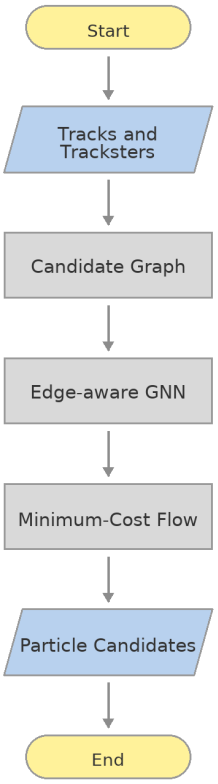


Figure 1 Final reconstruction pipeline. Tracks and tracksters are converted into a joint candidate graph, transformed using training-only statistics, scored by the edge-aware GNN, and assembled into charged and neutral candidates by the MCF optimizer. Model selection and parameter choices are frozen before the sealed test evaluation.

For the supervised sample, a track must be associated with a simulated object so that a simulation identifier can be assigned. Tracks are additionally required to have a valid nonzero MTD position, no more than four missing outer hits, $p_T > 1\,\text{GeV}$, and quality at least one. The MTD position is used in the time-of-flight calculation but is not placed directly in the GNN node feature table. Tracksters are matched to simulated particles when the association score is below 0.99 and the shared-energy fraction relative to the reconstructed raw energy is greater than 0.4. Because a reconstructed trackster may overlap more than one simulated particle, its target is stored as a set of simulation identifiers rather than a single class index.

3.3 Joint Event-Graph Construction

Candidate searches are accelerated with an η–ϕ tiling of each endcap. The final geometric requirement is applied with the wrapped azimuthal difference,

$$\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} < 0.2, \qquad \Delta\phi \in [-\pi, \pi]. \tag{1}$$

Only objects in the same detector endcap are linked. A track points to every compatible trackster. A trackster points only to compatible tracksters farther from the interaction point in $|z|$, producing an inner-to-outer directed graph and preventing the same ts–ts pair from appearing in both directions.

Tracks and tracksters occupy one shared, zero-padded node table with 21 columns. The ten track quantities are HGCAL η, ϕ, x, y, and z; momentum and transverse momentum; number of hits; MTD time; and MTD time uncertainty. The eleven trackster quantities are barycenter η, ϕ, x, y, and z; raw energy; time and time uncertainty; raw electromagnetic energy and transverse momentum; and total raw transverse momentum. Disjoint feature blocks preserve the object type without requiring separate graph objects.

Every candidate initially has eight pair-level attributes: $\Delta\eta$, wrapped $\Delta\phi$, ΔR, the target–source energy difference, the target/source energy ratio, a time-of-flight-corrected time difference, timing significance, and an edge-type flag. For a trk–ts pair, the source energy-like quantity is the track momentum and the flight distance starts at the track MTD position. For a ts–ts pair, both energy-like quantities are trackster raw energies and the flight distance is measured between the two barycenters. Preprocessing appends a ninth attribute that marks whether both endpoint timing measurements are valid.

Let $S(v)$ denote the set of simulated-particle identifiers assigned to a trackster v, and let $s(t)$ be the simulated-particle identifier assigned to a track t. Edge targets are defined by

$$y_{t\to v} = \mathbb{1}[s(t) \in S(v)]\,, \tag{2}$$

$$y_{u\to v} = \mathbb{1}[S(u) \cap S(v) \neq \varnothing]\,. \tag{3}$$

The same-particle condition in both definitions makes the learned task more specific than merely recognizing whether a trackster has any valid truth association.

3.4 Event Split and Leakage-Safe Preprocessing

Complete event graphs are the indivisible splitting unit. Stable identities are constructed from the source filename and the run, luminosity-block, and event numbers. With seed 42, the data are divided into 70% training, 15% validation, and 15% test partitions. From 5000 deterministic candidate assignments, the saved manifest selects the split that best balances total edge counts, positive edge counts, and the high-positive event tails for both edge types. The manifest also stores a dataset fingerprint so a different graph or label population cannot silently reuse the split.

All data-dependent preprocessing statistics are fitted exclusively on the training graphs. Continuous spatial and angular quantities are standardized. Large transverse momenta are clipped at a training-derived percentile and square-root transformed; trackster energy variables use logarithmic transformations. Energy differences and ratios on the edges use signed-log or log-ratio transformations as appropriate. Missing timing values are replaced with deliberately separated fill values, while the appended validity indicator allows the network to distinguish imputation from a physical measurement. Timing variables are clipped where required before standardization. The fitted statistics and their training-split fingerprint are saved and reused unchanged for validation and test graphs. The resulting model input dimensions are 21 node features and nine edge features.

3.5 Edge-Aware Graph Neural Network

The GNN uses a hidden dimension of 64. Separate multilayer encoders first map the node and edge attributes into the hidden space. Three edge-aware graph blocks then update the node states. Within each block, four attention heads combine projected source and target states with an edge-dependent contribution; attention is normalized over the candidate neighbours of each source node. An edge multilayer perceptron produces messages from the two endpoint states and the learned edge representation, and a residual node update with layer normalization completes the block.

After message passing, the classifier concatenates the final source state, final target state, learned edge representation, and original processed edge attributes. Its output is one raw logit z_e per edge rather than a post-sigmoid probability. The edge type remains part of the input, while masks separate the trk–ts and ts–ts objectives during optimization.

The final model is trained with a focal loss computed independently for each edge type. For a masked batch, the dynamic positive-class factor is

$$\alpha_{\text{dyn}} = \frac{N_-}{N_+ + N_-}, \qquad \alpha_{\text{eff}} = 0.5(0.6) + 0.5\alpha_{\text{dyn}}. \tag{4}$$

With label smoothing $\epsilon = 0.1$, focusing parameter $\gamma = 2$, smoothed target $\widetilde{y} = (1-\epsilon)y + \epsilon/2$, and $p_t = \exp[-\,\text{BCEWithLogits}(z, \widetilde{y})]$, the loss has the form

$$\mathcal{L}_{\text{focal}} = \frac{1}{N}\sum_{e=1}^{N} \alpha_{t,e}(1-p_{t,e})^{\gamma}\,\text{BCEWithLogits}(z_e, \widetilde{y}_e). \tag{5}$$

The two masked losses receive equal weight. Training uses AdamW with batch size 4, learning rate 10^{-4}, weight decay 10^{-5}, dropout 0.2, and seed 42. The epoch budget is 200, with early-stopping patience 15 and a minimum qualifying validation-loss improvement of 10^{-4}. The checkpoint with the lowest equally weighted validation focal loss is retained. Separate trk–ts and ts–ts probability thresholds are then selected on validation data by maximizing F1 and frozen for the final test.

3.6 Minimum-Cost-Flow Candidate Reconstruction

The GNN supplies local compatibility evidence, but independent edge decisions can produce conflicting or incomplete chains. The MCF stage therefore performs a global discrete optimization separately in the positive and negative endcaps. Graph-local node indices are first mapped back to the corresponding ROOT track and trackster indices. The frozen GNN is evaluated once per selected event, and its logits are cached for deterministic flow construction.

Each GNN logit is converted directly to an integer arc cost,

$$c_e = \text{round}(-wz_e + s). \tag{6}$$

The frozen selected transformations are

$$c_{\text{trk-ts}} = \text{round}(-100z + 131.9967), \tag{7}$$

$$c_{\text{ts-ts}} = \text{round}(-100z + 30.7473). \tag{8}$$

Thus, a more compatible edge receives a lower cost and is more attractive to the optimizer.

The flow network contains a source, one node per selected track, an input and output node for every trackster, and a sink. Source-to-track arcs initialize charged chains, source-to-trackster arcs initialize neutral chains, trk–ts and ts–ts arcs carry the learned costs, and trackster input-to-output arcs enforce usage capacity. Trackster-to-sink arcs terminate a chain. The selected fixed costs are -126 for track initialization, zero for trackster initialization, zero for neutral termination, and 500 for the additional-use penalty. At this stage, tracks are restricted to $p_T \geq 1\,\text{GeV}$, $p \geq 2\,\text{GeV}$, and $1.5 \leq |\eta| \leq 3.0$, and must already occur in the GNN graph.

OR-Tools solves the integer-capacity network with a total supply equal to the number of tracksters in the endcap. Positive-flow paths beginning at track nodes are decoded as charged candidates, whereas paths beginning directly at a trackster are decoded as neutral candidates. Reconstructed candidates that share tracksters are iteratively resolved using their overlap, calorimeter energy, and track momentum so that the final candidate collection has a consistent assignment of reconstructed objects.

3.7 Frozen Test Protocol

The development notebook performs training and validation without constructing the test subset. Before final evaluation, the split manifest, preprocessing statistics, model checkpoint, validation-derived thresholds, and MCF parameters are fingerprinted and frozen. The test notebook verifies these dependencies, performs no training or tuning, evaluates exactly one MCF configuration, and stores the outputs with the corresponding fingerprints. Only measurements produced by this sealed test procedure are reported in Section 4.

4 Results

4.1 Sealed Test Evaluation

This section reports only the results obtained from the sealed test partition. The GNN checkpoint, preprocessing statistics, edge-type probability thresholds, and MCF parameter set were fixed before the test data were loaded. No training, threshold selection, or MCF tuning was performed on these events. The GNN results therefore measure edge-link classification under the frozen model, while the MCF results measure the final reconstructed candidate collection produced from those edge scores.

4.2 GNN Edge-Link Classification

Figure 2 summarizes the three reported classification metrics for each edge type. The trk–ts classifier reached a test ROC-AUC of 0.980952, an F1 score of 0.746193, and an average precision of 0.831631. For ts–ts edges, the corresponding values were 0.961683, 0.368481, and 0.349867. Thus, both tasks retained strong ranking performance as measured by ROC-AUC, but the precision–recall-sensitive metrics were substantially lower for the more imbalanced ts–ts task.

Edge type	ROC-AUC	F1	Average precision
trk_ts	0.980952	0.746193	0.831631
ts_ts	0.961683	0.368481	0.349867

Figure 2 Final test metrics for trk–ts and ts–ts edge classification. ROC-AUC and average precision are threshold independent; F1 is evaluated at the edge-type threshold frozen on the validation partition.

The complete test ROC curves are shown in Figure 3. The trk–ts curve rises more sharply toward the upper-left corner, consistent with its larger ROC-AUC. The ts–ts curve also remains well above random ranking, although its lower average precision shows that its false-positive burden is more apparent under the test class distribution.

Figure 4 gives the decisions at the frozen probability thresholds. At the trk–ts threshold of 0.5801, the model produced 12,459 true negatives, 113 false positives, 187 false negatives, and 441 true positives. These counts correspond to a precision of 0.7960 and a recall of 0.7022. At the ts–ts threshold of 0.4485, it produced 74,743 true negatives, 756 false positives, 358 false negatives, and 325 true positives, corresponding to a precision of 0.3006 and a recall of 0.4758.

The matrices contain 13,200 trk–ts candidates and 76,182 ts–ts candidates. Positive links represent 4.76% of the trk–ts candidates but only 0.90% of the ts–ts candidates. This test-set composition explains why a high ts–ts ROC-AUC coexists with a much lower average precision

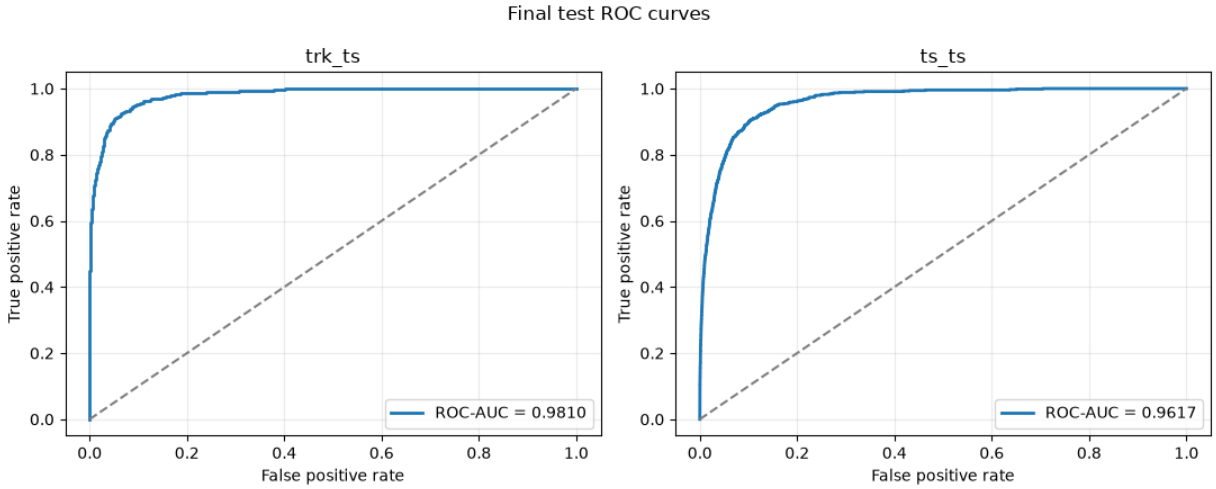


Figure 3 Final test ROC curves for trk–ts edges (left) and ts–ts edges (right). The dashed diagonal represents random ranking.

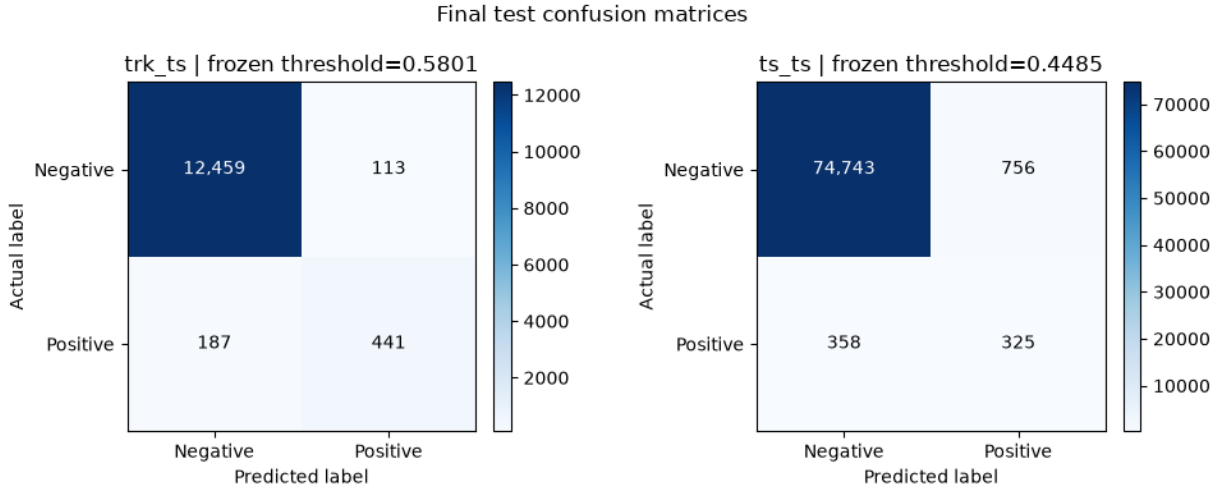


Figure 4 Final test confusion matrices at the validation-derived frozen thresholds. Rows are actual labels and columns are predicted labels.

and F1: even a small false-positive fraction produces many false links relative to the small positive population.

4.3 Minimum-Cost-Flow Reconstruction

The frozen selected MCF configuration produced 6,165 reconstructed candidates across the test events, consisting of 457 charged candidates and 5,708 neutral candidates. Its scalar and structural test measurements are shown in Figure 5. The event-averaged efficiency was 0.816698, the reported purity was 0.866325, the fake rate was 0.819858, and the merge rate was 0.012595.

	Efficiency	Purity	Fake rate	Merge rate
selected parameters	0.816698	0.866325	0.819858	0.012595

configuration	trk	chg	ts	Candidates	Charged candidates	Neutral candidates
selected parameters	1.0	0.761684	0.383262	6165	457	5708

Figure 5 Final test MCF measurements for the single frozen parameter set. The upper table reports scalar reconstruction metrics; the lower table reports structural consistency and candidate counts.

These scalar quantities use different denominators. Efficiency is the fraction of eligible, associated truth objects whose best reconstructed candidate contains more than half of the associated energy and, for a charged truth object, the correct track. The reported purity is the fraction of those truth objects associated with exactly one reconstructed candidate. Fake rate is instead normalized to the reconstructed candidate population, and merge rate is the fraction of reconstructed candidates associated with more than one truth object. The high purity and high fake rate are therefore not contradictory: the first measures truth-object splitting, whereas the second measures unmatched or incorrectly matched reconstructed candidates.

The structural measurements examine whether object combinations inside each candidate are consistent with one simulated particle or generator jet. The track–track value was 1.000000, the charged track–trackster value was 0.761684, and the trackster–trackster value was 0.383262. Track pairs within the same reconstructed candidate were therefore fully consistent in the evaluated sample, while the lower ts–ts value identifies trackster grouping as the weaker structural component of the final reconstruction.

4.4 Dependence on Pseudorapidity and Energy

Figure 6 shows the binned MCF efficiency. Across the displayed $|\eta|$ interval, the values remain approximately between 0.74 and 0.87 without a monotonic detector-wide trend. The energy dependence is stronger: efficiency is lowest in the first low-energy bin, rises through the intermediate-energy region, and reaches approximately 0.85–0.87 above about 100 GeV.

The corresponding fake-rate dependence is shown in Figure 7. The rate remains high throughout the displayed $|\eta|$ range, at approximately 0.78–0.93. Its energy dependence is more pronounced: the fake rate is largest at low energy, decreases rapidly across the intermediate-energy bins, and reaches its lowest values in the high-energy region. The binned behavior is consistent with the overall result and localizes the largest per-candidate fake probability to the low-energy region; the plotted rates do not by themselves determine how many candidates occupy each bin.

Overall, the sealed test demonstrates strong local ranking of candidate edges, particularly for trk–ts links, and an MCF reconstruction with high truth-object efficiency and a small merge rate.

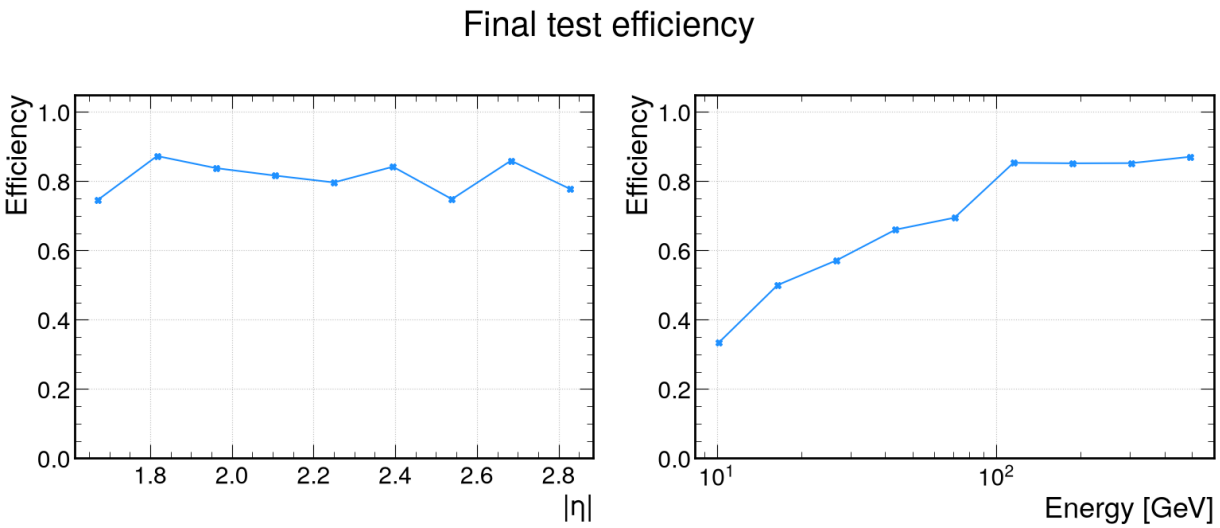


Figure 6 Final test MCF efficiency binned in $|\eta|$ (left) and truth energy in GeV (right). Each point is the mean success flag in its bin.

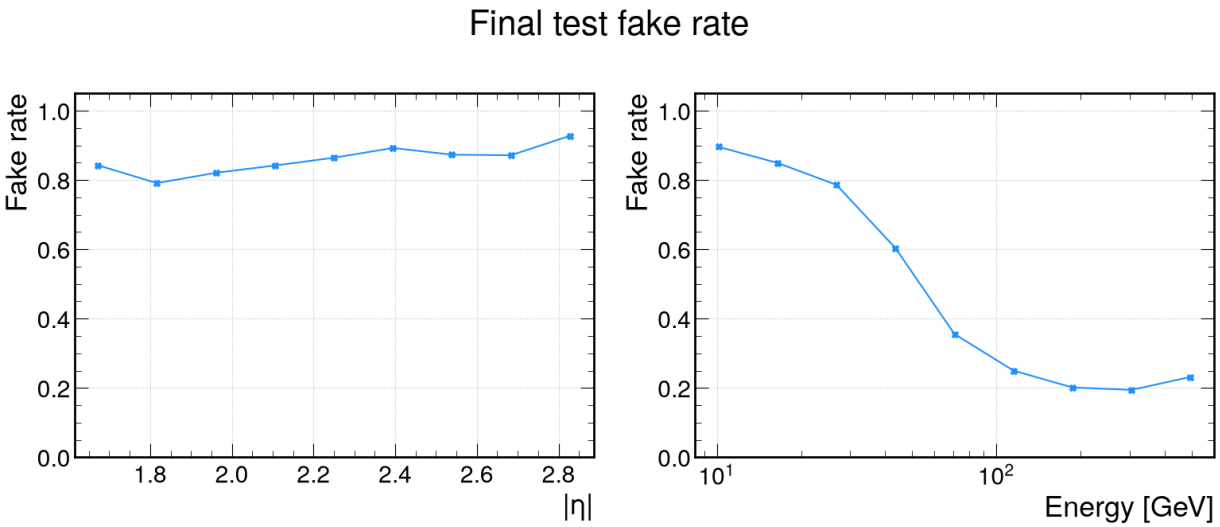


Figure 7 Final test MCF fake rate binned in $|\eta|$ (left) and reconstructed candidate energy in GeV (right). Each point is the mean fake flag in its bin.

The main observed weaknesses are the ts–ts precision, the trackster-pair structural consistency, and the large fake-candidate rate. Their interpretation and implications for further development are discussed in Section 5.

5 Discussion

5.1 Interpretation of Local Link Classification

The sealed test results show that the GNN learns useful local compatibility information for both edge types, but the two tasks are not equally resolved. The trk–ts classifier combines a track trajectory and momentum estimate with the position, energy, and timing of a calorimeter object. This provides a physically anchored extrapolation problem, and the resulting ROC-AUC of 0.980952 and average precision of 0.831631 indicate strong separation of compatible and incompatible candidates. At the frozen operating point, its precision of 0.7960 and recall of 0.7022 give a comparatively balanced classification result.

The ts–ts task is intrinsically more ambiguous. Both endpoints are fragments of calorimeter activity, and nearby tracksters may belong to the same shower, to different particles in a dense region, or to detector-level fragmentation with similar spatial and timing properties. The directed inner-to-outer construction reduces combinatorics, but it does not remove these physical ambiguities. The test ROC-AUC remains high at 0.961683, showing that the scores contain substantial ranking information, yet average precision falls to 0.349867. At the frozen threshold, the model retrieves 325 positive links while accepting 756 false links, giving precision 0.3006 and recall 0.4758.

This difference also illustrates why ROC-AUC alone is insufficient for this problem. Positive ts–ts links form only 0.90% of their test candidates. ROC-AUC measures the probability that a positive is ranked above a negative and is relatively insensitive to this class prior. Average precision, F1, and the confusion matrix expose the operational consequence of the same ranking under severe imbalance: a small false-positive fraction can still outnumber the true-positive population. The trk–ts positive fraction is larger at 4.76%, so an equally small false-positive fraction has a less severe effect on precision.

5.2 From Edge Scores to Global Reconstruction

The binary GNN operating points and the MCF serve different purposes. The confusion matrices evaluate probabilities at validation-derived frozen thresholds, whereas the MCF does not discard edges at those thresholds. It uses the continuous raw logits, transforms them into integer costs, and finds the minimum-cost set of paths subject to capacity and flow constraints. A classification false positive is therefore not automatically selected by the MCF, and a below-threshold edge can still contribute when its cost is favorable within the complete event solution.

The final reconstruction metrics demonstrate both the benefit and the limit of this global stage. The efficiency of 0.816698 shows that an eligible, associated truth object is often represented by a reconstructed candidate with sufficient shared energy and the correct track when one is required. The reported purity of 0.866325 and merge rate of 0.012595 further show that truth objects are usually not split across several candidates and that a single candidate rarely combines significant contributions from multiple truth objects. These measurements indicate that the global constraints can assemble many coherent paths while suppressing large-scale merging.

At the same time, the fake rate of 0.819858 shows that the reconstructed collection contains many additional candidates that do not satisfy the truth energy and track-matching requirements. This is not inconsistent with the efficiency or reported purity because the denominators differ:

efficiency and purity are truth-oriented, whereas fake rate is candidate-oriented. The result can therefore be summarized as high recovery of reconstructable truth objects accompanied by substantial overproduction of reconstructed candidates.

The structural measurements identify where the global chains lose consistency. The track–track score of 1.0 shows that multiple tracks grouped inside one candidate, when present, correspond to the same truth object or generator jet. Track–trackster consistency decreases to 0.761684, and trackster–trackster consistency falls further to 0.383262. This ordering matches the edge-classification evidence: the track provides a strong anchor, whereas grouping calorimeter fragments remains the more difficult part of the event interpretation.

The additive MCF objective also helps explain this boundary. It can enforce capacities and compare complete path costs, but it does not learn an additional whole-candidate representation. Its decision is the sum of initialization, termination, reuse, and pairwise GNN-derived costs. Consequently, it can resolve conflicts among informative local scores, but it cannot create missing discrimination when several ts–ts alternatives receive similar scores. Local ambiguity can then propagate through a chain even when the flow solution is globally optimal for the supplied costs.

5.3 Dependence on Detector Region and Energy

Efficiency is relatively stable across the displayed $|\eta|$ range, with no large monotonic degradation from the inner to the outer part of the studied acceptance. This suggests that the folded endcap representation and geometric candidate construction transfer reasonably across the selected pseudorapidity interval. The fake rate is also high throughout this range and rises toward its largest value in the highest displayed bin, so geometric coverage alone does not explain the candidate overproduction.

Energy produces a much clearer separation. The lowest-energy region has the smallest efficiency and the largest fake rate, while efficiency rises and fake rate falls as energy increases. This behavior is consistent with low-energy tracksters providing weaker and more easily confused spatial, energy, and timing evidence. Higher-energy showers offer a clearer signal and more stable energy sharing, making both the local links and the reconstructed chains easier to distinguish. The present binned curves establish this correlation, although they do not by themselves isolate which feature or candidate type causes it.

5.4 Overall Assessment

The project demonstrates a functioning integration of learned local linking and constrained global optimization. The GNN provides strong trk–ts ranking and useful, though less selective, ts–ts ranking. The MCF converts these scores into a candidate collection that recovers a large fraction of eligible truth objects with little merging. The central unresolved trade-off is that this recovery is achieved with a large fake-candidate population, alongside weak trackster-pair structural consistency.

The test results therefore support the two-stage design while showing that its quality is controlled by the least discriminating local relation. Global flow constraints improve event-level consistency, but the final reconstruction cannot be stronger than the information encoded in the edge costs. The specific limitations exposed by this evaluation and the corresponding routes for improvement are presented in Section 6.

6 Limitations and Future Work

The sealed test establishes the behavior of the submitted pipeline under its frozen configuration, but it does not make the method production ready. The following limitations define both the

interpretation of the reported results and the most important next steps.

6.1 Data and Supervision Limitations

Truth-dependent track selection.. The current graph-building code retains a track only when its identifier can be found in the simulated-trackster collection. This provides the simulation identifier required for supervised trk–ts labels, but it also uses truth to decide which reconstructed tracks enter the graph. Such a gate is unavailable for real detector data and removes unmatched or fake reconstructed tracks from the learning problem. The reported trk–ts performance therefore applies to a truth-qualified track population and may be optimistic relative to a fully reconstruction-level input. The first production-oriented change must separate sample selection from label construction: all tracks passing reconstructed quality cuts should enter the graph, while simulation information should be used only to assign positive or negative training targets.

Limited data-source coverage.. The final workflow expects one approved ROOT file and creates its training, validation, and test partitions from events within that source. The split is event-disjoint and leakage controlled, but it is not an independent-file or independent-sample generalization test. Robustness across other particle compositions, pileup conditions, detector configurations, and simulation productions has not been established. Future studies should train on multiple files and reserve complete files or simulation campaigns as external test domains.

Truth definitions depend on fixed association cuts.. Trackster labels require an association score below 0.99 and shared-energy fraction above 0.4; candidate metrics introduce additional 0.1 and 0.5 matching fractions. These definitions are consistent across the pipeline, but they determine which ambiguous fragments are called positive, efficient, pure, fake, or merged. Sensitivity to these choices has not been quantified. An expanded evaluation should report how conclusions change under alternative truth-matching definitions and should align the final metric definitions with the conventions used by the target TICL validation workflow.

6.2 Graph and GNN Limitations

Fixed candidate construction.. All candidates use $\Delta R < 0.2$, directed outward ts–ts edges, and fixed track-quality and timing requirements. A missing true edge cannot be recovered by the GNN or MCF, so the candidate graph places an upper bound on downstream efficiency. The project evaluates classification among generated candidates but does not separately report the candidate-generation recall or graph size as a function of ΔR, energy, and $|\eta|$. This coverage study should precede further classifier optimization and should identify the smallest geometric neighbourhood that retains the required truth links.

One model configuration and random seed.. The final result comes from one architecture, one focal-loss setting, and seed 42. The sealed test is valid for that choice, but it does not measure training variance or establish that the selected configuration is optimal. Repeated training with predeclared seeds and validation-only hyperparameter selection is needed to attach uncertainty to the learned model. Any configuration developed after observing the current test results must be evaluated on a new blind test sample rather than reused on the present test partition.

Shared representation for unequal edge tasks.. The GNN uses a common node encoder, message-passing stack, and edge classifier for trk–ts and ts–ts candidates. This supports information sharing, but the test results show very different class priors and operating behavior. A shared head may not provide enough capacity to specialize for calorimeter-fragment linking. Edge-type-specific output heads, heterogeneous message functions, or separate calibration maps should be

compared with the common architecture. The ts–ts task may also benefit from shower-depth, layer-overlap, shape, and energy-continuity features that describe a multi-trackster shower more directly than barycenter proximity alone.

Timing treatment has not been ablated.. Missing track and trackster timing values are imputed, and tracks with an unset MTD position are excluded. Although a validity indicator is provided for edge timing, the contribution and robustness of these choices have not been measured. Tests with and without timing, as well as controlled degradation of timing resolution and availability, are needed before attributing performance to the timing features or applying the method under different detector conditions.

6.3 MCF and Evaluation Limitations

Additive pairwise cost model.. The MCF minimizes a sum of transformed edge logits and fixed initialization, termination, and reuse costs. It enforces global capacities but has no learned whole-candidate representation or explicit shower-energy conservation term. The low trackster-pair structural consistency and high fake rate show the practical limit of this formulation. Future validation studies should examine logit calibration between the two edge types, neutral-chain priors, candidate-length penalties, and energy–momentum consistency constraints. The selection objective should combine truth efficiency with fake-candidate rejection rather than optimize either edge-level F1 or candidate efficiency in isolation.

Insufficient diagnostic breakdown of the fake rate.. Neutral candidates constitute 5,708 of the 6,165 reconstructed candidates, but the current output does not report fake rate separately for charged and neutral candidates. It also does not give candidate counts in each energy or $|\eta|$ bin. The binned curves identify where the conditional rate is high, but not which category contributes the largest number of fake candidates. Future artifacts should include charged/neutral, track multiplicity, chain length, energy, $|\eta|$, and event-occupancy breakdowns so the dominant failure mode can be isolated.

Statistical uncertainty and baselines.. The reported scalar metrics are event-averaged point estimates, and the curves do not contain statistical uncertainties or bin populations. Confidence intervals or bootstrap uncertainties are required to distinguish stable trends from finite-sample fluctuations. The integrated system should also be compared on the same events with standard TICL reconstruction, the reference MCF configuration, a GNN-only thresholded reconstruction, and simple geometric linking. These comparisons are necessary to quantify the contribution of each pipeline stage.

6.4 Deployment and Reproducibility Scope

The implementation is an offline Python prototype using `uproot`, PyTorch Geometric, and OR-Tools. It has not been integrated into the CMS Software framework (CMSSW) or benchmarked against online or offline reconstruction requirements. Runtime, memory, graph-size scaling, model export, numerical reproducibility across hardware, and failure handling for unusual events remain unmeasured. A deployment study should profile graph construction, GNN inference, and MCF solving separately before selecting a production representation.

6.5 Prioritized Next Steps

The immediate continuation of the project should proceed in the following order:

1. remove truth-dependent track admission and measure candidate-graph recall using reconstructed-only selections;
2. expand the dataset across independent files and conditions, then repeat training with multiple predeclared seeds;
3. improve and calibrate the ts–ts scorer using task-specific modelling and richer shower-continuity information;
4. retune the MCF on validation data with an explicit efficiency–fake-rate objective and detailed charged/neutral diagnostics;
5. compare against TICL and simpler reconstruction baselines with statistical uncertainties; and
6. evaluate CMSSW integration, latency, memory, and numerical stability.

Because the present test partition has already been examined, these changes must be developed on training and validation data and assessed on a newly reserved blind sample. This preserves the meaning of the sealed results reported in Section 4.

7 Conclusion

This project developed an end-to-end method for linking tracks to tracksters and tracksters to other tracksters in the TICL reconstruction pipeline. The central idea was to separate the problem into two complementary stages: an edge-aware graph neural network learns local object compatibility, and a Minimum Cost Flow optimizer combines those scores into globally constrained charged and neutral particle candidates.

The submitted implementation converts each event into a joint directed graph with trk–ts and outward ts–ts candidates inside $\Delta R < 0.2$. Tracks and tracksters share a typed 21-feature node table, while each link is represented by nine processed spatial, energy, timing, validity, and edge-type attributes. Simulation associations provide same-particle edge labels. A deterministic event-level split, training-only preprocessing, structured checkpoints, and content fingerprints make the training and evaluation procedure reproducible. The final GNN uses three edge-aware graph blocks, dynamic-alpha focal loss, and separate masked objectives for the two edge types. Its frozen logits are mapped to integer MCF costs and solved independently in both detector endcaps.

On the sealed test partition, trk–ts classification achieved ROC-AUC 0.980952, F1 0.746193, and average precision 0.831631. The ts–ts task reached ROC-AUC 0.961683, but its F1 of 0.368481 and average precision of 0.349867 showed the greater difficulty of selecting true calorimeter-fragment links under a positive fraction below one percent. These results establish that the GNN learns useful rankings for both relations while identifying ts–ts precision as the principal local-linking weakness.

The frozen MCF configuration produced 6,165 candidates, including 457 charged and 5,708 neutral candidates. It obtained event-averaged efficiency 0.816698, reported purity 0.866325, fake rate 0.819858, and merge rate 0.012595. Track–track structural consistency was 1.0, whereas track–trackster and trackster–trackster consistency were 0.761684 and 0.383262. The global optimization therefore recovered a large fraction of eligible associated truth objects with little merging, but it also produced a large population of unmatched or incorrectly matched candidates. The energy dependence further showed that the reconstruction is most reliable for higher-energy objects and most difficult in the low-energy region.

The main contribution of the work is a unified and reproducible demonstration that learned pairwise compatibility can be integrated with flow-based global reconstruction for joint trk–ts

and ts–ts linking. The sealed results support this two-stage design but do not yet establish production readiness. The truth-dependent admission of tracks must be removed, validation must be extended across independent samples and random seeds, ts–ts discrimination must be strengthened, and the MCF objective must reject fake candidates more effectively. Because the present test sample has already been examined, these improvements require development on training and validation data followed by a new blind evaluation. Within these boundaries, the project provides a concrete foundation for further graph-based global event interpretation in TICL.

References

[1] Muhammad Anwar. *Graph_Linking: linking_trk_trkst*. https://github.com/Moanwar/Graph_Linking/tree/main/linking_trk_trkst. 2024.

[2] Aurora Perego. *trackLinking: prepare_data.py*. https://github.com/AuroraPerego/trackLinking/blob/main/edgesNN/prepare_data.py. 2024.

[3] Aurora Perego. *trackLinking: trackAndCandidatesLinkingWithMCFandNN.ipynb*. https://github.com/AuroraPerego/trackLinking/blob/main/trackAndCandidatesLinkingWithMCFandNN.ipynb. 2024.