



A Flexible and Efficient Approach to Missing Transverse Momentum Reconstruction

William Balunas¹ · Donatella Cavalli² · Teng Jian Khoo³ · Matthew Klein⁴ · Peter Loch⁵ · Federica Piazza⁶ · Caterina Pizio² · Silvia Resconi² · Douglas Schaefer⁷ · Russell Smith⁸ · Sarah Williams¹

Received: 30 August 2023 / Accepted: 16 December 2023
© The Author(s) 2023

Abstract

Missing transverse momentum is a crucial observable for physics at hadron colliders, being the only constraint on the kinematics of “invisible” objects such as neutrinos and hypothetical dark matter particles. Computing missing transverse momentum at the highest possible precision, particularly in experiments at the energy frontier, can be a challenging procedure due to ambiguities in the distribution of energy and momentum between many reconstructed particle candidates. This paper describes a novel solution for efficiently encoding information required for the computation of missing transverse momentum given arbitrary selection criteria for the constituent reconstructed objects. Pileup suppression using information from both the calorimeter and the inner detector is an integral component of the reconstruction procedure. Energy calibration and systematic variations are naturally supported. Following this strategy, the ATLAS Collaboration has been able to optimise the use of missing transverse momentum in diverse analyses throughout Runs 2 and 3 of the Large Hadron Collider and for future analyses.

Keywords Missing transverse momentum · ETmiss · Reconstruction · Hadron collider

✉ Teng Jian Khoo
teng.jian.khoo@cern.ch

William Balunas
william.keaton.balunas@cern.ch

Donatella Cavalli
donatella.cavalli@mi.infn.it

Matthew Klein
matthew.henry.klein@cern.ch

Peter Loch
loch@physics.arizona.edu

Federica Piazza
federica.piazza@cern.ch

Caterina Pizio
Caterinapizio@gmail.com

Silvia Resconi
silvia.resconi@mi.infn.it

Douglas Schaefer
schae@cern.ch

Russell Smith
russell.smith@cern.ch

Sarah Williams
sarah.louise.williams@cern.ch

¹ Cavendish Laboratory, University of Cambridge, Cambridge, UK

² INFN Sezione di Milano & Dipartimento di Fisica, Università di Milano, Milano, Italy

³ Institut für Physik, Humboldt-Universität zu Berlin, Berlin, Germany

⁴ Physics Department, Southern Methodist University, Dallas, TX, USA

⁵ Physics Department, University of Arizona, Tucson, AZ, USA

⁶ Institute for Fundamental Science, University of Oregon, Eugene, OR, USA

⁷ Enrico Fermi Institute, University of Chicago, Chicago, IL, USA

⁸ Nevis Laboratory, Columbia University, Irvington, NY, USA

Introduction

Despite the highly developed state of particle detector design in the era of the Large Hadron Collider (LHC) [1], there exist particles which even the most sensitive instruments cannot reliably detect. Neutrinos and similar particles that are practically undetectable (or “invisible”) are the signifying features of numerous processes of interest, including Standard Model electroweak physics and more hypothetical Beyond-the-Standard Model (BSM) processes such as production of supersymmetric particles or dark matter. At hadron colliders, the main constraint on invisible particle kinematics in a given event is the missing transverse momentum, the negative vector sum of the transverse momenta of all objects associated with the recorded event. This two-dimensional vector is denoted \vec{p}_T^{miss} , with its magnitude as p_T^{miss} , E_T^{miss} , or where mathematical notation is infeasible (e.g. in software), “MET”.¹ This paper describes a novel approach for reconstructing p_T^{miss} that preserves a great deal of flexibility to meet the diverse needs of physics analyses. It was adopted by the ATLAS experiment [2] within the context of its software suite [3], and published in the open source Athena repository [4], for the simulation, reconstruction, and analysis of collision data and has been in use since LHC Run 2.

Due to its status as an inclusive event observable, the reconstruction of p_T^{miss} requires the imposition of a global event description: the selection of all objects associated with the hard interaction, and identification or classification of these objects in order to perform calibration. Hitherto, the definition of this event description has been effectively static [5], fixed at the point when the experimental data are reconstructed, which due to CPU constraints can only be performed occasionally. This static representation would function as follows:

1. Select particle candidates to be included in the p_T^{miss} calculation.
2. Perform an ambiguity resolution procedure between all selected particle candidates to prevent double-counting of detector signals.
3. Sum the transverse momentum vectors of all surviving particle candidates, optionally dividing this sum into terms representing the contributions of particular particle types.
4. Sum the transverse momentum vectors of basic detector signals not affiliated with particle candidates as a representation of diffuse soft hadronic recoil.

5. Combine the “hard” particle terms with the soft recoil term into the total vectorial transverse momentum of the event and invert the sign of this vector.

While this makes the p_T^{miss} computation straightforward to implement, this static definition is a limitation in several ways:

- Particle identities are fixed in the p_T^{miss} on long timescales and shared across all use cases, limiting options for optimising these for a given analysis.
- Particle selection must be made on the basis of calibrations derived at the time of reconstruction, which may be inconsistent with updated calibrations used in the final analysis.
- Systematic uncertainties cannot be fully accounted for in the particle selection, particularly where these uncertainties can affect reconstructed particle identities.

To partially address these issues, the ATLAS p_T^{miss} event data model (EDM) has historically incorporated a scheme for book-keeping, recording exactly which reconstructed objects were used in the p_T^{miss} calculation and any modifications to the kinematics of these objects applied in the process. This was realised by recording a set of persistent references to each object contributing to each term of the p_T^{miss} calculation, together with a weight for each transverse momentum vector component. An implementation of this static EDM (`MissingETComponentMap`) can be found in the 21.2 branch of the Athena repository [4].

In this paper, an improved p_T^{miss} EDM is described, which compactly records information necessary to freely recompute the p_T^{miss} given an arbitrary event description and any necessary four-momentum corrections applied to the constituent objects. This is accomplished by encoding information needed for resolving signal overlaps between different selected objects, using the constituents of hadronic jets [6] as a basis. The new “dynamic” EDM effectively addresses the limitations mentioned above, facilitating efficient optimisation of object selections and p_T^{miss} computations, as well as fully consistent treatment of systematic uncertainties. An intuitive user interface is critical to ensuring that customisation of the p_T^{miss} calculation can be performed by a wide user base.

The outline of the paper is as follows. Motivations for the dynamic p_T^{miss} EDM are given in the context of the ATLAS software and constraints on p_T^{miss} reconstruction in the “[Motivation](#)”. A full description of the EDM follows in the “[Event Data Model](#)”. Then, “[Reconstruction Implementation](#)” explains the algorithmic implementation of the p_T^{miss} reconstruction, showing how information is compiled into the EDM encoding. The user interface for analysis is detailed in “[Analysis Interface](#)”. “[Computational Performance](#)”

¹ This paper uses the notation p_T^{miss} as this represents a momentum rather than an energy, but “MET” or “Missing E_T ” remains ubiquitous for historical reasons. These all denote the same variable.

demonstrates the performance gains in CPU and disk usage from the new approach. Adaptation of the design to address recent and future challenges in the LHC computing environment are described in the “[Adaptability for Run 3 & Beyond](#)”. Finally, conclusions are presented in the “[Conclusion](#)”.

Motivation

The design of the ATLAS p_T^{miss} reconstruction is foremost specified by the need for a compact data structure that permits a flexible reconfiguration of the p_T^{miss} calculation, while ensuring that the calculated observable is robust against reconstruction errors. While full flexibility could be achieved by retaining in analysis data formats all objects needed for the p_T^{miss} computation, the disk cost would be prohibitive due to the extremely high multiplicity of low-energy signals. Furthermore, p_T^{miss} reconstruction necessitates non-trivial operations to disambiguate the calorimeter and tracker signals that may be shared between multiple reconstructed objects. This makes the procedure more complex than a simple summation over selected particles, and implies the need for a supporting infrastructure to facilitate the p_T^{miss} computation in the end stages of analysis event processing, when the final object calibrations are available. Below, a detailed description of the constraints on ATLAS p_T^{miss} reconstruction are given, which determine the specifications for the p_T^{miss} EDM and associated analysis tools. Despite the attention paid here to the specific case of ATLAS, these considerations can be taken as representative of a typical general purpose particle detector operating at a hadron or lepton collider.

Detector Structure

The ATLAS detector possesses a concentric cylindrical structure optimised both for particle identification and energy/momentum measurements [2]. The inner detector (ID), immersed in a solenoidal magnetic field, provides precision reconstruction of charged-particle trajectories. Electromagnetic and hadronic showers are captured in the calorimeter system, with coverage close to 4π in solid angle. The calorimeters possess longitudinal as well as transverse segmentation, to capture shower development in depth. These components are surrounded by a muon spectrometer (MS) integrated with ATLAS’s eponymous toroidal magnet system. The sub-detectors of ATLAS vary in their fiducial acceptance, with the ID coverage being similar to but more limited than that of the MS, while the calorimeter coverage extends significantly further into the forward region than the other components.

Overview of ATLAS Event Reconstruction

Reconstruction of analysis objects in ATLAS from the digital detector output, or from analogous simulated inputs, takes place in several steps. First, *basic constituents* are constructed from the raw digital signals:

- Hits in the inner detector are fitted to produce a set of tracks describing the trajectories of charged particles [7].
- Calorimeter cell energies are determined and calibrated based on the sampled calorimeter pulse shapes to the scale of electron and photon showers as measured in test beams (electromagnetic scale). Cells are grouped into noise-suppressed *topological clusters* (topoclusters) [8]. The resulting clusters may subsequently be calibrated to correct their energies to match the scale of hadrons using cell-level weights (hadronic scale).
- *Track segments* are formed from hits in the muon spectrometer, which may further be combined into muon spectrometer standalone tracks [9].

Subsequently, these basic constituents are combined to form particle candidates or hadronic jets. Of crucial importance is the fact that most higher-level reconstruction operations of this nature run independently, such that the outputs of different particle identification (PID) algorithms in most cases do not influence one another. The following *objects* are reconstructed:

- *Electron candidates* are identified based on the presence of narrow showers in the electromagnetic calorimeter [10]. At least one nearby track is associated to the electron candidate and used to refine the energy/momentum measurement. Quality criteria based on shower shape and track properties are defined to provide a balance between high efficiency and a low misidentification rate of fake electron candidates.
- *Photon candidates* are identified similarly to electron candidates, but no track is required; nearby tracks may be used to improve the four-momentum measurement and PID under the assumption that the photon has undergone a conversion in the inner detector material [10].
- *Muon candidates* are identified using inner detector and/or muon spectrometer tracks [9]. The most precise reconstruction and PID is achieved when extrapolated ID tracks can be matched to a MS track, however muon candidates may also be formed using a more limited set of hits in either system, to improve coverage and reconstruction efficiency. Calorimeter cells along the muon trajectory are identified and used to improve estimates of the energy deposited in the calorimeter by the muon.
- A particle flow algorithm is run over the tracks and topoclusters, to extract a better measurement of the kin-

ematics of charged hadrons and permit suppression of charged hadron pileup contributions to hadronic measurements [11]. Muon and electron candidate tracks are excluded from the particle flow algorithm, as the energy subtraction is optimised for pion showers. The outputs of the particle flow algorithm are termed *particle flow objects* (PFOs). Each PFO is called ‘charged’ if a track was used in its reconstruction or ‘neutral’ if only topoclusters were used.

- *Jets* are reconstructed using sequential clustering algorithms (usually anti- k_r [12], as implemented in FASTJET [13]) from topoclusters calibrated at either the hadronic or electromagnetic scales, or charged and neutral PFOs [11, 14]. The inputs provided to the jet clustering algorithm are henceforth designated *jet constituents*. Tracks in the catchment area of each jet are matched to the jet and used for calibration and pileup suppression. A sequence of calibrations is applied to subtract pileup contributions, match the average jet scale to that of simulated hadrons, compensate for response differences due to flavour and shower development and finally to correct differences between the response determined in simulation and in data [14].
- Hadronically decaying *tau lepton candidates* are seeded from jets reconstructed from hadronic scale topoclusters. Particle flow techniques are used to refine the tau lepton energy scale calibration and the description of the tau lepton decay and shower [15]. Multivariate classifiers are used to discriminate hadronic tau leptons from parton-initiated jets and electrons [16].

Computation of p_T^{miss} requires the selection of objects that represent a coherent description of the event, with basic constituents used no more than once in the summation. Hence objects that share tracks or clusters must undergo an overlap removal procedure. Fully reconstructed and calibrated objects account only for a limited fraction of the total momentum from the hard scattering process, and therefore the remainder must be estimated from basic constituents unassociated to the selected objects.

Analysis Requirements on p_T^{miss} Reconstruction

Analyses using ATLAS data vary widely in their requirements, with the target final state and dominant background processes heavily influencing exactly which objects are selected for analysis. To correctly handle systematic uncertainties and to ensure a coherent description of each event, the objects from which the p_T^{miss} is calculated must be consistent with the objects defining the event selection and other event reconstruction procedures. This leads to a set of specific requirements on the p_T^{miss} EDM.

First and foremost, the ability to freely choose exactly which leptons, photons, and jets are accepted is important for analysis optimisation. Optimisation of an analysis requires the ability to repeat the analysis procedure in approximately a day, if not faster, whereas data samples for analysis are reconstructed afresh at most a handful of times in a year. The use of additional intermediate dataset formats permits a higher frequency for reconstruction operations to be repeated, but is still limited to a timescale of weeks or (more often) months for update of configurations. This significantly disadvantages the use of static p_T^{miss} calculations, as even the provision of dozens of p_T^{miss} configurations fails to satisfy highly optimised analyses, and comes at a substantial cost of disk space and CPU. A non-negligible consideration is the occasional need to fix bugs in the p_T^{miss} calculation itself as well as optimisation of the p_T^{miss} computation.

PID decisions and four-momentum calibrations are applied at the time of reconstruction, but are typically re-applied during event processing for analysis, taking advantage of refinements derived during the course of data-taking. For this reason, a computation of p_T^{miss} solely based on information available in the initial reconstruction procedure will not correspond to the calibrations and PID used in analysis event selection. This implies that the p_T^{miss} calculation must be able to be updated with the final calibrations and input selection at analysis time. Even in the case where input selection and calibration could be frozen in advance, the application of systematic uncertainties on object four-vectors taking into account the overlap removal and other corrections applied during p_T^{miss} reconstruction would imply that additional information about the contribution of each object to the p_T^{miss} sums must be recorded. Furthermore, the correct handling of systematic uncertainties requires mutability of object selection, as object identities may themselves be subject to uncertainties.

Summary of Demands on EDM and Analysis Tools

To address the requirements previously described, the following were determined to be necessary features that must be provided by the p_T^{miss} EDM:

1. The EDM should record the full space of *possible overlaps* between reconstructed objects rather than forming a simple kinematic sum after overlap removal.
2. The EDM and supporting tools must permit users to specify lists of selected leptons and photons defining priorities for each class of object to be retained during overlap removal.
3. The EDM must be capable of determining the momentum sum of basic constituents not associated to selected objects.

4. The EDM and reconstruction procedure must support differing signal bases for jet reconstruction: topoclusters and tracks or PFOs.

In the next section, an implementation of an EDM that satisfies the criteria above is detailed.

Event Data Model

A C++ implementation of this dynamic p_T^{miss} EDM is carried out in the context of the `xAOD` EDM devised by ATLAS for LHC Run 2 [17] (2015–2018). It is made available in the `Event/xAOD/xAODMissingET` directory of the ATLAS Athena software repository [4]. As context for the p_T^{miss} EDM description, the `xAOD` data structure is first briefly described, together with relevant fundamental elements of the ATLAS EDM.

Overview of the `xAOD` EDM

The `xAOD` structures data in a tree, using the `TTree` class from the `ROOT` framework [18], but emulates the organisation of these data into objects (`AuxElement`), which may be grouped into *containers* (`AuxVectorData`). Classes deriving from `AuxElement` provide the object-oriented interface, whereas the underlying data are stored in an auxiliary store comprising a set of `std::vector` data members. For the i th element of a container, the corresponding data is held in the i th elements of each `std::vector` in the auxiliary container. Data on `xAOD` objects may be static (i.e. defined explicitly in the auxiliary container) or dynamic. Dynamic data may be recorded in the form of a mutable augmentation, or “decoration”, attachable even to immutable objects.

Containers serve not only as the receptacle for data content, but also as the vessel for information transfer between algorithmic components, being recorded in a *store* with access given by a corresponding *key*. To provide persistent references to individual objects, the `ElementLink` construct is used, which identifies a given object by the key of its container and the index of the object within the container. Concretely, the `ElementLink` is a template class, taking the target container type as template argument.

A common base class is shared by all reconstructed `xAOD` objects possessing four-momenta: the `IParticle` class, which itself derives from `AuxElement`. Apart from serving as a base type, the `IParticle` interface chiefly provides access to the basic four-vector, and a check of the type of the object via a C++ `enum` named `ObjectType`.

The derived classes of `IParticle` with relevance to the p_T^{miss} EDM at the time of writing include:

- `CaloCluster`,
- `TrackParticle`,
- `PFO` (superseded by `FlowElement` for Run 3),
- `TruthParticle`,
- `Jet`,
- `Electron`,
- `Photon`,
- `Muon`,
- `TauJet`.

Description of the p_T^{miss} Data Classes

Two sets of `xAOD` objects are defined to hold the p_T^{miss} data:

- `MissingET` object—This represents the kinematic properties of the p_T^{miss} itself, including values for each individual term in the sum used to compute it.
- `MissingETAssociationMap`—This is a representation of dynamic calculation, including signal-base (topocluster/track/PFO) four-momentum sums needed for overlap removal.

Each set of classes is described in the following sections. The efficiency and flexibility of this method originates from the structure of the `MissingETAssociationMap`, which encodes all information needed to apply any possible overlap removal choices downstream in a compact format.

`MissingET`

The ultimate goal of the p_T^{miss} reconstruction is to provide the p_T^{miss} two-vector, corresponding to the best estimate of the total vectorial transverse momentum carried by particles produced in the primary interaction which are non-interacting and stable on detector scales. As such, the main EDM object with direct physical significance is a representation of this two-vector, augmented with supporting information including identification of the contributing objects in the form of a bitmask (“source” tag) and the scalar transverse momentum sum, which captures information important to p_T^{miss} performance characterisation. The interface object is called the `MissingET`, which is ordinarily held in a `MissingETContainer`.

For additional information, the contributions to the total p_T^{miss} from different types of particle candidates are encoded in distinct `MissingET` objects, held in the same `MissingETContainer` as the total p_T^{miss} . These are distinguished and retrieved from the container primarily by name, via a fast hash comparison, but can also be extracted by the “source” tag. A qualitative sketch of the high-level structure is shown in Fig. 1.

Fig. 1 Illustration of the high-level structure of the `MissingETContainer` EDM and the typical basic constituents (tracks and calorimeter energy clusters) corresponding to each element. The “soft track term” in this example represents the contribution from tracks not associated to any particle candidate. The arrows represent the overall direction of each term

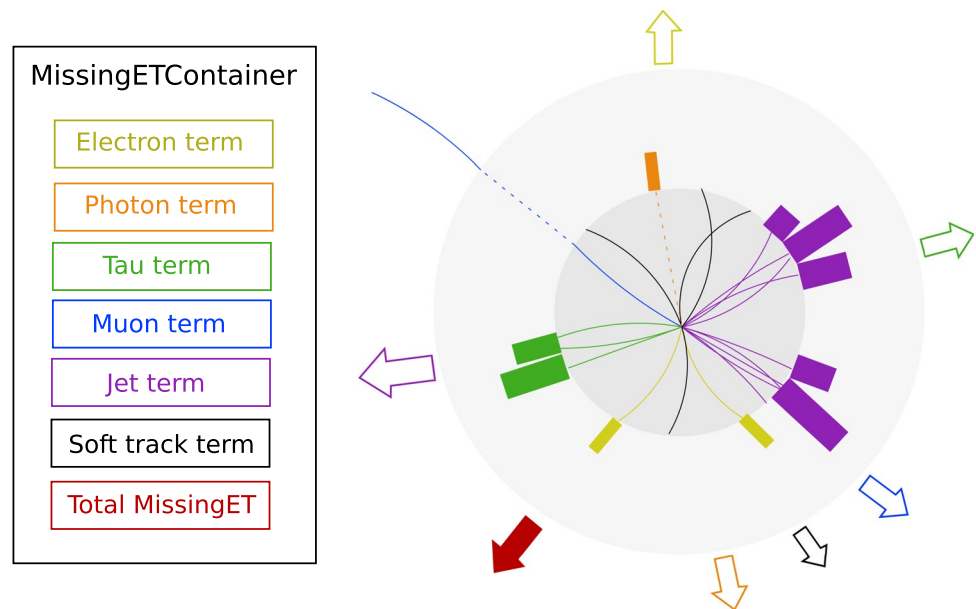
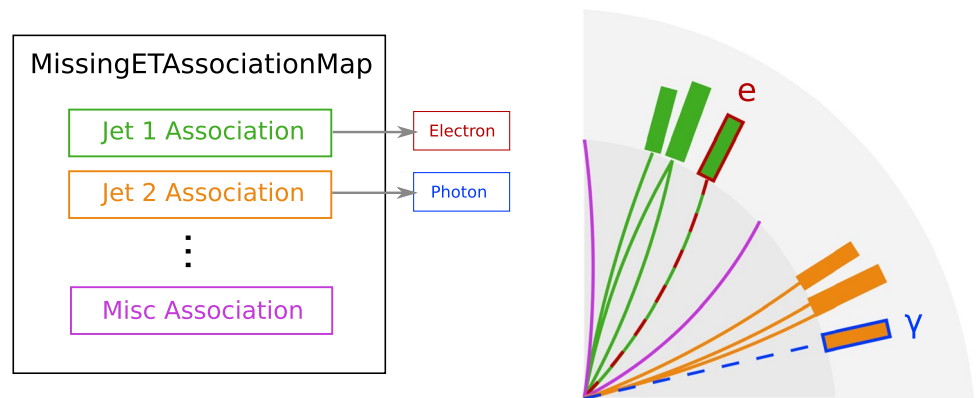


Fig. 2 Illustration of the high-level structure of the `MissingETAssociationMap` EDM and the typical detector signals (tracks and calorimeter energy clusters) corresponding to each element



`MissingETAssociationMap`

To satisfy the demands listed in the “[Summary of Demands on EDM and Analysis Tools](#)”, a compact representation is needed of the possible combinations of distinct objects whose transverse momenta should be summed to compute the total p_T^{miss} two-vector. The `MissingETAssociationMap` encodes this information efficiently, permitting overlap removal of the contributing energy/momentum measurements for an arbitrary choice of quality criteria to be applied to the lepton/photon candidates used in the calculation. This overlap removal is able to be carried out precisely, down to the level of the individual basic constituents (tracks/topoclusters/PFOs) used to reconstruct the contributing physics objects.

To condense the necessary information for the overlap removal procedure, jet constituents are used as a basis on which to represent the energy/momentum contributions of every lepton/photon candidate to the total energy measured

in each collision event. Leptons and photons are then matched to jets on the basis of shared basic constituents, limiting the search space needed to determine signal overlaps between leptons and photons. The `MissingETAssociationMap` can thus be constructed from a set of individual `MissingETAssociation` objects. For an event with N_{jet} reconstructed particle-jets, $N_{\text{jet}} + 1$ `MissingETAssociation` objects are required; one per jet, and one “miscellaneous” association, recording the signal contributions that were not matched to any jet. These exist because jet clustering algorithms such as anti- k_r include an energy or momentum cutoff below which a clustered object is not considered a jet. A simple example is illustrated in Fig. 2.

To mitigate sensitivity to pileup, an alternate representation of the event is formed using only information from charged particles matched to the nominal hard-scatter primary vertex (commonly identified based on associated track momenta), localised by ghost-associating [19, 20] inner

detector tracks to the jets. This is computed in the same way as the calorimeter-based/inclusive representation of the event, but requires additional information to be stored in each `MissingETAssociation` object. Concretely, each association holds the following information:

`jetLink` An `ElementLink` to the association’s *reference jet*, whose constituents make up the basis for all computations with this association object. For the miscellaneous association, an invalid link is recorded.

`isMisc` A `bool` indicating if this association object is the miscellaneous association.

`objectLinks` A `std::vector<ElementLink>`, identifying the leptons/photons sharing basic constituents with the reference jet. A lepton/photon may share constituents with multiple reference jets, and hence be represented in multiple association objects.

`overlapIndices` A `std::vector<std::vector<size_t>>`, holding a sparse representation of the overlaps between leptons/photons linked in the `objectLinks` vector. For each element in the `objectLinks`, the indices to any other elements of `objectLinks` that share constituents are stored, with the representation being symmetric.

`overlapTypes` A `std::vector<std::vector<unsigned char>>`, with entries corresponding to the indices in `overlapIndices`, for which each element functions as a bitmask identifying the types of basic constituents that were found to be shared between the overlapping objects. This representation is primarily used to indicate whether the overlaps are between contributing charged particle tracks or if calorimeter energy clusters are also shared. For example, it may suffice to perform overlap removal on muons solely on the basis of shared tracks.

`calpx/calpy/calpz/cale/calsumpt` Five `std::vector<float>` members recording the four-momentum and scalar p_T sum of each basis group of constituents needed to perform overlap removal on the reference jet and associated leptons/photons, as defined in the “[Overlap Removal Encoding](#)”.

`calkey` A `std::vector<bitmask_t>`², index parallel with the `calpx/py/pz/e/sumpt` vectors, encoding the associations of the constituent basis groups to the objects referred to by `objectLinks`.

`trkpx/trkpy/trkpz/trke/trksumpt/trkkey` Analogous information to the `calpx, . . .` vectors, computed from selected tracks ghost-associated to the reference jet, representing potential contributions of associated leptons/photons to the track-based soft term.

`jettrkpx/jettrkpy/jettrkpz/jettrke/jettrksumpt` The four-momentum and scalar p_T sum of the selected tracks ghost-associated to the reference jet, representing the reference jet’s maximal contribution to the track-based soft term.

Not all basic constituents in a given event are used to reconstruct higher level particle candidates. The p_T^{miss} corresponding to only these leftover basic constituents is recorded in a separate `MissingET` object, referred to as the “core soft term” for the event, which is effectively inert as far as overlap removal is concerned.

Overlap Removal Encoding

Within each `MissingETAssociation`, the possible overlaps between combinations of selected objects are decomposed according to the following process, in order to generate the *overlap removal basis*. The end result is to fill the association’s member variables described above, which fully encode all information required for an arbitrary overlap removal definition downstream.

1. For each of the N_{obj} lepton/photons matched to this association, extract the list of basic constituents amongst the reference jet’s constituents, and generate a constituent-to-particle-candidate map. This map is not necessarily one-to-one; a constituent may be associated with multiple objects. To aid human interpretation, it may be helpful to sort the particle candidates by any stable ordering principle, for example descending order in p_T .
2. Assign each particle candidate a boolean flag indicating whether this object was selected for the p_T^{miss} computation. The state of these flags can be represented by a binary string of length N_{obj} , with the maximal value of this string being $2^{N_{\text{obj}}} - 1$. The particle candidate with index i will be represented by the bit corresponding to 2^i .
3. Let \vec{C}_j be a 5-dimensional vector with elements representing the four-momentum coordinates p_x, p_y, p_z, E and the p_T of the jet constituent j . p_T is included here explicitly so that when many such vectors are summed, this component of the result will contain the scalar p_T sum. Let \mathcal{M} be a one-to-one map of binary strings o to 5-d vectors V_o where $o \in [0, 2^{N_{\text{obj}}} - 1]$. The binary strings created in step 2 will form a basis for constructing these o .
4. For each jet constituent, check whether it is associated to any particle candidates. If not, continue. If it is, then define $o_j = \sum_{i \in \text{matched particle candidates}} 2^i$, and add \vec{C}_j to $\mathcal{M}(o_j)$. Record the `overlapIndices` and set

² Typedef of unsigned long long.

the `overlapType` bit for each corresponding particle candidate i .

- For each o, V_o pair in \mathcal{M} with non-zero V_o , add an entry to `calkey` and `cal[px/py/pz/e/sumpt]`, filling in the corresponding components respectively.

The process is repeated substituting ghost-associated hard-scatter tracks for jet constituents in order to fill the `trk[key/px/py/pz/e/sumpt]` vectors.

Reconstruction Implementation

The central goal of the “reconstruction” step is to construct the objects needed at analysis level to compute the p_T^{miss} with the desired flexibility. These consist of the `MissingETAssociationMap` encoding all possible overlaps between hard objects (jets, leptons and photons satisfying particle identification criteria), as well as a representation of the “core soft term” (the constituents not associated to any hard object) in the form of a `MissingETContainer`. Finally, these objects can be used to build the final `MissingET` object according to any given analysis-level object selection definitions. Algorithms for constructing the `MissingETAssociationMap` and derived `MissingET` objects can be found respectively in the `Reconstruction/MET/METReconstruction` and `Reconstruction/MET/METUtilities` directories of the ATLAS Athena software repository [4].

Building the Association Map

The primary challenge in defining these objects is that the jets, leptons, and photons are created from different detector signals. Depending on the type of particle, these can include tracks, PFOs, topoclusters, or other specialised detector-level objects. Therefore, a method is required for associating jet constituents and tracks with leptons and photons that may have been built using different sets of low-level objects. For example, photons may be reconstructed using energy deposits in the calorimeter, but not using the same topocluster or PFO definition that jets are built with. This association of constituents to these hard objects is performed by a tool known as a `MissingETAssociator`. This is a base class which has a different specialization for each type of hard object, implementing the appropriate method for associating the constituents to that particular type.

The first step in the reconstruction procedure is the construction of the `MissingETAssociation` objects—one for each jet, plus the miscellaneous association. This is performed by a `METJetAssocTool`, which sets the `jetLink` and `isMisc` variables for each association. Additionally, this tool initialises a map from jet constituents

(represented in `ElementLink` form) to association indices. This map is kept as a member of the `MissingETAssociationMap` and is used for search functionality in the rest of the reconstruction procedure. It acts as a transient cache which is not written to disk in the persistent representation of the `MissingETAssociationMap`, and is used to keep track of which objects are selected for the p_T^{miss} calculation.

After creation, the associations are sequentially filled with information corresponding to each type of hard object that can overlap with the jets (and each other): muons, electrons, photons, and tau leptons. In each case, the corresponding `MissingETAssociator` is used to associate jet constituents (or tracks) with the hard objects and then fill the `MissingETAssociationMap` with the relevant information. After determining the associations for each hard object (as described below), the tool iterates over the associated constituents. For each constituent/track, the corresponding jet is used to select the correct `MissingETAssociation`. In the case where the constituent/track is not associated with any jet, the miscellaneous association is used. The member variables of this `MissingETAssociation` are filled or updated with the constituent/track information described in the Sect. “`MissingETAssociationMap`”. This is repeated until all of the constituents/tracks associated with all hard objects have been allocated to an `MissingETAssociation`.

The exact methods used for associating jet constituents and tracks to leptons and photons vary depending on the object type and are detailed below. In the cases of tracks and charged PFOs (which are constructed from tracks), the track in question must always pass quality requirements and be associated with the primary vertex to be considered. Whenever a jet constituent is a charged PFO, its track is used to determine its associations. In some instances, precise criteria are not listed for some aspects of these methods; in these cases the definitions are chosen or tuned by the user.

Muon reconstruction includes association of tracks with the muon. The same association is used for p_T^{miss} reconstruction. Only ID tracks are considered, such that “standalone” muons reconstructed using only the MS have no associated tracks. For the purposes of p_T^{miss} reconstruction, only the original track is used; any refitting from the muon reconstruction procedure is ignored.

Although muons are minimally ionising particles, the energy they deposit in the calorimeters is not entirely negligible, and may lead to measurable calorimeter signals exceeding noise thresholds. The calorimeter cells crossed by a muon track can be identified, in particular any cells in topoclusters or neutral PFOs that might contribute to the reconstruction of jets or other objects. Association of these cells with the muon can be used to avoid double-counting their energy in corrections to the muon momentum and overlapping jets or the p_T^{miss} soft term.

Electron, photon and hadronic tau lepton candidates are reconstructed using a combination of topoclusters and ID tracks. The associations between the particle candidates and their basic constituents, defined using angular proximity or other more complex selection criteria, are recorded prior to further manipulations refining the particle energy/momentum reconstruction, and can be recalled for the purposes of p_T^{miss} reconstruction. Were these associations not retained in the particle schema, they would need to be reproduced by repeating the corresponding matching procedures. A custom association procedure could also be followed for particle candidates built from basic constituents that do not directly map onto the $\text{jet}/p_T^{\text{miss}}$ constituents.

The Core Soft Term

The core soft term represents the contribution of all constituents/tracks that are not associated with any jet or other hard object. This is constructed by a tool called a `MET-SoftAssociator`. This functions by iterating over all constituents/tracks and searching the map for the `MissingETAssociation` in which it is represented. There are three possible outcomes for each constituent/track:

1. It is found in a `MissingETAssociation` corresponding to a jet, meaning it is associated with a jet and potentially with leptons/photons.
2. It is found in the miscellaneous association, meaning it is associated with one or more leptons/photons but not with a jet.
3. It is not found in any association, meaning it is not associated with any hard object at all.

All constituents/tracks which are not represented in any association (i.e. outcome 3) are included in the core soft term. This simply consists of a `MissingET` object containing the vector and scalar sums of the transverse momenta of these objects.

The core soft term is distinguished from a general soft term because the latter is defined only at the analysis level, where different selections on leptons and photons may be applied. For example, if a downstream selection removes an isolated lepton, its contribution to the p_T^{miss} will enter the soft term, despite not being included in the core soft term. The information necessary for this later redefinition of leptons/photons at analysis level is encoded in the miscellaneous association in this case.

Calculating the Missing Transverse Momentum

The final step of the reconstruction procedure is the actual computation of the missing transverse momentum from the association map, core soft term, and hard objects in the

event. The format of this output is a `MissingETContainer`, consisting of one `MissingET` for each term and one for the total sum. This is created by initializing an empty `MissingETContainer` and sequentially adding each set of hard objects to it via a tool called `METMaker`. The user is free to do this in any order they choose (or omit some objects if necessary), but the standard convention is electrons, photons, tau leptons, muons, and finally, jets. `METMaker` can be used directly in analysis to reconstruct the p_T^{miss} using customised object selections. Its main functionality is implemented in the function `rebuildMET()`, which takes as input the collection of hard objects to be added to the calculation. This function creates a new `MissingET` representing the corresponding term, computes its 2-dimensional momentum vector (excluding objects which fail overlap removal), and inserts it into the `MissingETContainer`. In the case of jets, a specialised function `rebuildJetMET()` is used instead. This constructs the jet and soft terms, ensuring that no overlapping momentum contributions are double-counted by checking against the objects that were included in the earlier terms. In this way, a consistent and correct calculation is achieved even if the user applied additional selection requirements to the other hard objects before adding them. The calculation is then completed using the `buildMETSum()` function of `METMaker`, which adds all of the terms together in a vector sum to compute the total p_T^{miss} , which is also inserted into the `MissingETContainer`. The overall procedure is represented in Fig. 3.

Overlap Removal

Each time `rebuildMET()` or `rebuildJetMET()` is called, it is necessary to add only momentum contributions which are not otherwise included as part of a higher-priority object. The definition of which objects receive priority can vary based on user choices; a typical example of standard usage by ATLAS is given in Ref. [21]. This defines the order as: electrons, photons, hadronic tau leptons, muons, jets. In general, when two hard objects overlap, one is removed entirely from the calculation. If the removed object contains signals that are not part of any retained object, these signals will then be captured in the soft term.

This method requires keeping a record of which signals are forming part of a hard object term, and which are not (and should therefore be included in the soft term). This is implemented in the form of a transient bitmask, `useObjectFlags`, which is index-parallel with `calkey/trkkey`. The value of each bit encodes whether the corresponding object has been selected for one of the hard object terms. When a collection of hard objects is added via `METMaker`, this is checked to determine if the given constituent basis group should be omitted to avoid

```

1 // Instantiate the helper class to record transient object selections
2 xAOD::MissingETAssociationHelper metHelper(metMap);
3 xAOD::MissingETContainer met;
4 // Create MET terms for each type of physics object
5 // Specify name of MET term and type of object,
6 // output container, input objects and helper
7 metMaker.rebuildMET(
8     "Electrons", xAOD::Type::Electron, met, selected_electrons, metHelper
9 );
10 // Ditto for photons, tau leptons, muons in overlap removal order
11 ...
12 // Complete jet and soft term reconstruction
13 // Specify name of jet term and soft term
14 // output container, input objects, core soft term and helper
15 metMaker.rebuildJetMET(
16     "Jets", "PVSoftTracks",
17     met, jets, core_soft_met, metHelper
18 );
19 // Build total MET, specifying type of soft term
20 met::buildMETSum("Total", met, MissingETBase::Source::Track );
21 // Access term by string key
22 std::cout << "Total MET px: " << met["Total"].mpx()
23     << ", py: " << met["Total"].mpy()
24     << ", magnitude: " << met["Total"].met()
25     << std::endl;
26

```

Fig. 3 Pseudocode illustrating p_T^{miss} reconstruction with object selections. Details such as pointer usage and status code checks are omitted

double-counting it. Subsequently, `useObjectFlags` is updated accordingly if any further objects are selected in each step. To satisfy thread safety requirements discussed in “[Adaptability for Run 3 & Beyond](#)”, this bitmask and procedure are implemented within a transient helper class `MissingETAssociationHelper` instantiated once for each complete invocation of `METMaker`. Earlier versions which did not require thread safety (e.g. during LHC Run 2) instead implemented this directly within `MissingETAssociationMap`.

Analysis Interface

This section discusses the interface used for applying analysis-specific (re-) calculations of p_T^{miss} , which include:

- Using analysis-specific object overlap and overlap removal procedures,
- Applying updated calibrations for selected hard objects,
- Propagating the impact of systematic uncertainties impacting hard objects through the p_T^{miss} calculation,
- Choosing objects used for calculating the soft p_T^{miss} term (cluster-based or track-based), and

- Applying additional systematic uncertainties to the soft p_T^{miss} term.

(Re-)Calculating Missing Transverse Momentum in Analyses

The first three above operations can be performed using the same workflow for the initial p_T^{miss} reconstruction described in “[Calculating the missing transverse momentum](#)”. That is to say the functions `rebuildMET()` or `rebuildJetMET()` can be called using analysis-specific selected and calibrated objects. This permits users to modify their object definitions/calibrations used in the p_T^{miss} calculation and even fix any potential bugs without requiring any large-scale reprocessing of data. As before, the order in which the p_T^{miss} terms are rebuilt using `rebuildMET()` defines a priority list, and objects overlapping with prior objects are omitted from inclusion in the term being calculated, whilst the energy/momentum associated with the selected object is subtracted from jets that contain them. At analysis level, `rebuildJetMET()` is called with all the (calibrated) jets in the event. However, `METMaker` can be configured using the argument of this function such that different centrally defined “work-points” for jets included in the p_T^{miss} calculation can be

applied. These working points impose additional selection criteria on the jets to reduce the impact of pileup on p_T^{miss} reconstruction. These can impact performance and the resulting systematic uncertainties in a non-trivial way so optimisation of the p_T^{miss} working points are also part of the analysis design. In addition to selecting the working point for jet selection, `rebuildJetMET()` also allows the user to choose the calculation used for the soft term. Since Run 2, the default soft term calculation is the “track soft term (TST)”, where only tracks associated to the primary vertex but not associated with prior hard objects are included in the soft term, however the calorimeter-based “cluster soft term (CST)” can also be used to account for soft neutral objects at the expense of increased sensitivity to pileup. Once these functions have been called the total p_T^{miss} can be calculated using the `buildMETSum()` function of `METMaker`. This procedure can be repeated to calculate the p_T^{miss} associated with systematic variations that impact the 4-momenta of the calibrated objects.

The implementation of `rebuildMET()` and `rebuildJetMET()` can include post-hoc corrections to incorrect reconstruction logic in the production of the `MissingETAssociationMap`, which would typically come in the form of incomplete overlap removal. When such corrections can be made, this is a major advantage, due to the computational cost and the lead times needed to rerun a corrected reconstruction campaign. Bugfixes can be provided in this way to analysis users effectively as soon as they are devised and validated. The scope for analysis-level corrections is significantly broader in the dynamic p_T^{miss} EDM.

When calculating the p_T^{miss} for new calibrations or systematic variations of hard objects such as electrons and muons, an object called `ShallowAuxContainer` is used to represent copies of the original objects with some modification applied. Rather than duplicating all data members in memory, this “shallow copy” points back to the original auxiliary container it was created from in order to read data which is the same between the original and the copy. However, variables that are set explicitly are stored locally as part of the copy. This allows, for example, the 4-vector of an object to be updated by a new calibration, without making a “deep copy” of the original object (i.e. copying *all* of its associated variables) which would be more memory and CPU intensive. Since the `MissingETAssociationMap` identifies these objects by reference to their container and index, a method is needed to match the calibrated object back to the original one for the p_T^{miss} re-calculation. This is achieved by decorating the copied object with an `ElementLink` to the original object.

As well as enabling the propagation of systematic uncertainties impacting hard objects through the p_T^{miss} reconstruction, the software also enables the evaluation of the

impact of systematic uncertainties on the soft p_T^{miss} term. These uncertainties are applied as variations on the corresponding p_T^{miss} object itself. They are implemented in a `METSystematicsTool`, which provides a function `applyCorrection()` taking as input the `MissingET` to be varied (usually a shallow copy of the original) and the corresponding `MissingETAssociationMap`. For each component of the systematic uncertainty, this function can be called to modify or vary the p_T^{miss} object accordingly. In practice, these generally take the form of adjustments to the scale or resolution of the soft term, decomposed into orthogonal components parallel or perpendicular to the total transverse momentum of all hard objects. In the case of a “resolution variation”, the soft term component is smeared by a factor randomly sampled from a Gaussian distribution with the appropriate width. Before `applyCorrection()` is called, the systematics tool must be “told” what kind of variation to apply, using another function `applySystematicVariation()`, which takes as an argument an ATLAS-common object specifying a systematic uncertainty definition. By iterating through all desired uncertainty components affecting the soft term, one can obtain a set of varied p_T^{miss} objects encapsulating their effects, which may then be used as inputs to statistical interpretations. The generality of this interface allows any form of variation to be applied to the soft term in principle.

Specialised Functionality

Occasionally it is beneficial for a physics analysis project to construct a customised variant of the p_T^{miss} . The flexibility of the p_T^{miss} reconstruction design makes it readily adaptable to these analysis-specific techniques that may not follow a standard prescription. For example, the “recoil” variable used in W boson mass measurements [22], which does not involve any jet definition, can be reconstructed with these tools. Two further examples are given below in more detail to demonstrate this principle: p_T^{miss} calculated purely from tracks, and p_T^{miss} computed as if some subset of the objects in the event are “invisible” to the detector. Note also that in general there are no restrictions on the objects used in the calculation, provided that the association map is constructed using the appropriate constituents. For example, the user may choose a different vertex or track selection when building the map, and this is supported transparently. Typical use cases include analyses using only photons in their final state, which cannot rely on tracking to identify a primary vertex (e.g. Refs. [23, 24]).

In addition to the choice of track-based or cluster-based soft term calculations, the analysis interface also enables the calculation of an entirely track-based p_T^{miss} to further reduce pileup contamination, at the expense of excluding neutral

particles from the calculation (which can degrade its accuracy) and limiting the acceptance of the calculation to that of the tracker. This calculation is handled by the `rebuildTrackMET()` function of `METMaker` which takes the same arguments as the `rebuildJetMET()` function. This functions identically (and calls `rebuildJetMET()` itself), except all tracks associated with jets are counted in the “soft track term”. When these tracks are added to the object ordinarily called the “track soft term”, the result is in fact the full track-based p_T^{miss} , as it then includes all tracks in the event which pass the quality requirements are associated with the primary vertex. No other objects such as leptons or photons are included in the calculation, so no overlap removal is required in this case. For a typical use case of this p_T^{miss} definition, see Ref. [25].

`METMaker` also includes functionality to mark a container of objects as “invisible”, excluding these objects and their overlaps from the p_T^{miss} calculation. The implementation is provided by a function `markInvisible()` in `METMaker`, which replaces the corresponding call to `rebuildMET()`. This functions much like `rebuildMET()` in that it creates a p_T^{miss} term corresponding to the given collection of objects, accounting for overlaps. The difference is that it sets that term’s `source` tag to a designated value that indicates that the term is to be ignored in the final, overall p_T^{miss} calculation. When `buildMETSum()` is called, it will skip any terms which are marked as having an invisible source. The result is what the detector would have seen if the given set of particles did not interact with it at all. This has use cases such as marking muons as invisible in $Z \rightarrow \mu\mu$ events to obtain a sample representative of $Z \rightarrow \nu\nu$ events, e.g. for estimating backgrounds in analyses using p_T^{miss} without relying on simulations.

Computational Performance

In an increasingly resource-limited computing environment, the compact energy overlap representation and dynamic p_T^{miss} computation permit significant gains in the CPU and disk cost of providing optimised p_T^{miss} quantities for analysis. These savings are achieved primarily by eliminating redundant operations and information that would otherwise be needed to adapt the computation to diverse physics object selections.

Algorithmic Efficiency

The CPU cost of the initial reconstruction step is dominated by the cost of associating p_T^{miss} and jet constituents (clusters, tracks or PFOs) to the physics objects. At worst, this scales as the product of the number of physics objects ($\mathcal{O}(10)$) and signal constituents ($\mathcal{O}(10)$), but can be accelerated by using

predefined links when the physics objects are built from common constituents, or by preemptively segmenting the object collections into bins in η and ϕ to minimise the number of required comparisons.

For a benchmark sample of 2018 data, with 50 interactions per event on average, reconstruction of raw data into the basic ‘AOD’ analysis data format (the standard file format for reconstructed data, structured using the `xAOD` model) in a 8-threaded job takes approximately 2 s per event [26], evaluated on a 16-core Intel®Xeon®CPU E5-2630 v3 at 2.40 GHz. In such a job, each instance of association map construction contributes less than 1% of the CPU cost. Three instances are run, producing output maps serving consumers of three jet collections,³ for which the constituent associations and overlap calculations need to be repeated. This time is comparable within 10% to the time taken for computing a single p_T^{miss} collection in a static p_T^{miss} event data model, as the required matching operations are for the most part identical. Any differences due to applying a stricter object selection in the static model are outweighed by the necessity of rerunning the associations for any change of the object selection. For comparison, running one jet clustering algorithm takes 1.5–2 times this per event, while building simple sums over calorimeter cells is up to 6 times more expensive due to requiring iteration over nearly 200,000 cells.

At analysis time, the additional computation needed to determine the final p_T^{miss} two-vectors is $\mathcal{O}(1\%)$ of the per-event processing time needed to prepare event collections, even when the operation needs to be redone for every systematic variation. In a static EDM, similar operations need to be carried out in order to update object calibrations and apply systematic variations.

Size on Disk

The output size for each of the p_T^{miss} collections is approximately 1.5 kB per event, two thirds of which is in the p_T^{miss} association maps, and the remainder constituted by the core soft term. Simulated events typically have a larger disk footprint, as large samples are produced of processes of interest, which are often biased towards higher centre-of-mass energies. Using top-quark pair events as a benchmark, the size of the p_T^{miss} collections is 3 kB per event, with the same breakdown between the association maps and core soft term container. The size and scaling of the core soft term content with the event activity may come as a surprise; this is because links to the soft signals contributing to the core soft terms are saved as decorations on the objects, permitting traceability and specialised

³ The three jet collections used here were $R = 0.4$ jets built from topoclusters, $R = 0.4$ jets built from PFOs, and $R = 1.0$ jets built from topoclusters, all using anti- k_r , where R is the clustering distance parameter.

studies of the soft term composition. For physics analysis purposes, this additional information may be dropped, saving an additional 0.5 kB per event, an important reduction for end-user analysis formats that are kept as lightweight as possible.

For LHC Run 3 (i.e. since 2022), the ATLAS analysis model [27] includes two data formats to cover the needs of most analysis use cases:

- DAOD_PHYS contains uncalibrated physics object containers holding all the content needed for analysis-ready calibrations to be applied, with a target event size of 50 kB/event. This is representative of the analysis formats used in Run 2.
- DAOD_PHYSLITE is a more streamlined format in which calibrations are applied in advance, minimising the disk footprint as well as the analysis-time CPU, with a target event size of 10 kB/event.

In DAOD_PHYS, two p_T^{miss} collections are retained, each making up 3% of the total event size. In DAOD_PHYSLITE, the truncated physics object collections require a recomputation of the p_T^{miss} associations, but this in fact permits two further size reductions. Firstly, only one standardised jet collection is retained, hence only one p_T^{miss} collection need be saved. Secondly, as fewer object overlaps need to be registered, the size of the p_T^{miss} association maps is also substantially reduced, resulting in a total size of 0.3 kB/event. For comparison, the total p_T^{miss} content in Run 1 consolidated analysis formats was 10% of the event content, which contained 27 custom p_T^{miss} definitions, in a file format that was also less optimised for efficient storage.

Overall, substantial savings are achieved in the computational resources needed to reconstruct p_T^{miss} with the added benefit of an increased flexibility for optimisation and refinements of the p_T^{miss} reconstruction strategy.

Adaptability for Run 3 & Beyond

The p_T^{miss} EDM and reconstruction algorithms described above were originally developed and used at ATLAS for Run 2 of the LHC. However, the flexibility of this design makes it easily adaptable to the evolving needs of physicists in Run 3 and beyond into the era of the High Luminosity LHC. This section discusses three particular elements which have been developed for usage in this scope: support for multithreaded processing, global particle flow reconstruction, and reconstruction of objects with large impact parameter tracks.

Multithreading

In order to best make use of limited hardware resources in the face of ever-increasing computational demands, all ATLAS

reconstruction software was adapted for multithreaded use from Run 3 onwards [28]. This introduced several complications into the design of all EDM and reconstruction software, including for p_T^{miss} . One requirement for the thread-safe design is that all of the p_T^{miss} EDM classes must not have their states modified after they have been recorded to the event store (otherwise, multiple threads may attempt to do this simultaneously, thus spoiling the information used by the other). Effectively, this means no mutable data members are permitted. In order to meet this requirement, the purely transient `MissingETAssociationHelper` class was introduced to handle the object selection flags discussed in Sect. “Overlap Removal”, which would otherwise require mutable members on the `MissingETAssociationMap`.

When the association map has been created and p_T^{miss} needs to be computed, the user initialises a `MissingETAssociationHelper`. As the only argument to its constructor, the user provides a pointer to the relevant `MissingETAssociationMap`. The `MissingETAssociationHelper` then internally initialises one bitmask per association, which can be freely modified since it is thread-local. After this, the user no longer needs to interact directly with the association map: the helper itself can be provided to `METMaker` instead. As each object is sequentially added to the calculation, the helper’s object selection flags are dynamically updated, ensuring a correct and consistent application of the overlap removal. This allows the state of the association map itself to remain unchanged throughout the procedure.

Global Particle Flow

In addition to multithreading, the p_T^{miss} reconstruction software is also extended to support “global particle flow” in Run 3. This is a method for considering a global event description, where each PFO is treated as a particle candidate [29]. It entails associating PFOs with reconstructed objects, including photons, electrons, muons, and tau leptons, when these objects share tracks or calorimeter cells with the PFOs. It aims to provide a more accurate ambiguity resolution than could be achieved by retrospectively determining associations downstream, thereby allowing more refined calibrations and improving the resolution of p_T^{miss} reconstruction. For example, PFOs that are associated with (or ‘labelled as’) electrons can be excluded from calibrations accounting for charged hadrons.

Photons, electrons, muons, and tau leptons are reconstructed from the same tracks that are used to build PFOs. Therefore, the association for these is straightforward: a charged PFO is associated with any of these objects if its track was also used to reconstruct that object. The situation is somewhat more complicated when associating these objects with neutral PFOs, since

all use calorimeter cells in their reconstruction but not precisely the same topoclusters that are used to construct PFOs. However, electron, photons, and tau leptons make use of the same topoclusters during their reconstruction, so the global particle flow method allows a direct association regardless. Associating neutral PFOs with muons is done by identifying the calorimeter cells crossed by the muon's track, as described in “[Building the Association Map](#)”.

After the PFOs and other objects are initially reconstructed, these associations are determined and saved to each object as a dynamic decoration. This takes the form of an `ElementLink` which points to the associated PFO. Links pointing from the PFO to the associated objects are created similarly. When the p_T^{miss} association map is then created, these links can simply be dereferenced to find the correct association. This allows reconstruction of the association map even from data formats which do not contain the detailed information necessary to initially determine the association. Global particle flow still allows for overlaps between reconstructed objects in general, so the association map is still required to encode them.

Large Impact Parameter Tracks

Improvements in the ATLAS inner detector tracking for Run 3 permit more widespread reconstruction of charged particles with a large transverse impact parameter [30]. These tracks can be used to build particle candidates, especially leptons, that may be produced at a large radius in the detector from the decays of metastable particles. Whereas these ‘large radius tracks’ (LRTs) are composed of ID hits left over after standard track reconstruction, and entirely disjoint from the standard ID tracks, for lepton reconstruction they may be combined with the same topoclusters or MS segments as used for standard lepton reconstruction. The p_T^{miss} soft term is constructed only of prompt tracks hence the LRTs can be ignored in overlap removal, but common calorimeter signals must still be correctly handled. To extend the p_T^{miss} EDM to correctly handle large radius leptons, it therefore suffices to run `MissingETAssociationMap` reconstruction appending additional `MissingETAssociators` for each LRT lepton collection, with no further algorithmic modifications.

Conclusion

This paper has presented the design and implementation of the event data model (EDM) used by the ATLAS Collaboration for reconstructing missing transverse momentum, p_T^{miss} ,

since its second data taking run (Run 2). While defined for the particular purposes of ATLAS, the principles underlying this design generalise to other collider experiments.

This improved EDM enables the flexible recalculation of p_T^{miss} at analysis level using analysis-specific physics object selections and choices of overlap removal. It also allows the p_T^{miss} to be recalculated using updated calibrations for selected “hard” objects, and enables systematic uncertainties associated with these objects to be propagated through the p_T^{miss} calculation. It supports analysis-specific choices for calculating the soft p_T^{miss} term (e.g. cluster-based or track-based) and applying systematic uncertainties associated with this calculation. This design also results in more efficient usage of computing resources, as the most performance-intensive part (associating constituents with hard objects) only needs to be performed once, even if multiple object definitions are used or subsequently changed. The structure of the EDM is also space-efficient, allowing full information for customizable p_T^{miss} calculation to be retained in even the most streamlined analysis data formats. The flexible and modular design also makes the EDM and reconstruction algorithms easily adaptable to the evolving needs of Run 3 and the future, facilitating fully multithreaded computing and adaptation or replacement of any part of the methodology to suit changes in requirements.

Acknowledgements This work was done as part of the offline software research and development programme of the ATLAS Collaboration, and we thank the collaboration for its support and cooperation. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 787331).

Author Contributions WB, TJK and SW wrote the main manuscript text and WB prepared Figs. 1, 2. All authors contributed to specific details of the design and implementation of the software, and reviewed the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability All source code pertaining to this article is open access (see Ref. [4]). CPU performance and disk usage statistics discussed under 'Computational Performance' were compiled on data restricted to the the ATLAS Collaboration.

Declarations

Competing Interests The authors declare no competing interests.

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/>.

References

- Evans L, Bryant P (2008) LHC machine. JINST 3:08001. <https://doi.org/10.1088/1748-0221/3/08/S08001>
- Collaboration ATLAS (2008) The ATLAS experiment at the CERN large hadron collider. JINST 3:08003. <https://doi.org/10.1088/1748-0221/3/08/S08003>
- ATLAS Collaboration (2021) The ATLAS Collaboration Software and Firmware. ATL-SOFT-PUB-2021-001. <https://cds.cern.ch/record/2767187>
- ATLAS Collaboration (2021) Athena. <https://doi.org/10.5281/zenodo.2641996>. <https://gitlab.cern.ch/atlas/athena>
- ATLAS Collaboration (2017) Performance of algorithms that reconstruct missing transverse momentum in $\sqrt{s} = 8$ TeV proton–proton collisions in the ATLAS detector. Eur Phys J C 77:241. <https://doi.org/10.1140/epjc/s10052-017-4780-2>. arXiv:1609.09324
- Salam GP (2010) Towards Jetography. Eur Phys J C 67:637–686. <https://doi.org/10.1140/epjc/s10052-010-1314-6>. arXiv:0906.1833
- Cornelissen T, Elsing M, Fleischmann S, Liebig W, Moysé E (2007) Concepts, design and implementation of the ATLAS New Tracking (NEWT). ATL-SOFT-PUB-2007-007. <https://cds.cern.ch/record/1020106>
- ATLAS Collaboration (2017) Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1. Eur Phys J C 77:490. <https://doi.org/10.1140/epjc/s10052-017-5004-5>. arXiv:1603.02934
- ATLAS Collaboration (2021) Muon reconstruction and identification efficiency in ATLAS using the full Run 2 pp collision data set at $\sqrt{s} = 13$ TeV. Eur Phys J C 81: 578. <https://doi.org/10.1140/epjc/s10052-021-09233-2>. arXiv:2012.00578
- ATLAS Collaboration (2019) Electron and photon performance measurements with the ATLAS detector using the 2015–2017 LHC proton–proton collision data. JINST 14:12006. <https://doi.org/10.1088/1748-0221/14/12/P12006>. arXiv:1908.00005
- ATLAS Collaboration (2017) Jet reconstruction and performance using particle flow with the ATLAS Detector. Eur Phys J C 77:466. <https://doi.org/10.1140/epjc/s10052-017-5031-2>. arXiv:1703.10485
- Cacciari M, Salam GP, Soyez G (2008) The anti- k_r jet clustering algorithm. JHEP 04:063. <https://doi.org/10.1088/1126-6708/2008/04/063>. arXiv:0802.1189
- Cacciari M, Salam GP, Soyez G (2012) FastJet user manual. Eur Phys J C 72:1896. <https://doi.org/10.1140/epjc/s10052-012-1896-2>. arXiv:1111.6097
- ATLAS Collaboration (2020) Jet energy scale and resolution measured in proton–proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector. Eur Phys J C 81:689. <https://doi.org/10.1140/epjc/s10052-021-09402-3>. arXiv:2007.02645
- ATLAS Collaboration (2016) Reconstruction of hadronic decay products of tau leptons with the ATLAS experiment. Eur Phys J C 76:295. <https://doi.org/10.1140/epjc/s10052-016-4110-0>. arXiv:1512.05955
- ATLAS Collaboration (2017) Measurement of the Tau Lepton Reconstruction and Identification Performance in the ATLAS Experiment Using pp Collisions at $\sqrt{s} = 13$ TeV. ATLAS-CONF-2017-029. <https://cds.cern.ch/record/2261772>
- Buckley A, Eifert T, Elsing M, Gillberg D, Koeneke K, Krasznahorkay A, Moysé E, Nowak M, Snyder S, Gemmeren P (2015) Implementation of the ATLAS Run 2 event data model. J Phys Conf Ser 664(7):072045. <https://doi.org/10.1088/1742-6596/664/7/072045>
- Antcheva I et al (2009) ROOT: A C++ framework for petabyte data storage, statistical analysis and visualization. Comput Phys Commun 180:2499–2512. <https://doi.org/10.1016/j.cpc.2009.08.005>. arXiv:1508.07749
- Cacciari M, Salam GP (2008) Pileup subtraction using jet areas. Phys Lett B 659:119–126. <https://doi.org/10.1016/j.physletb.2007.09.077>. arXiv: 0707.1378
- Cacciari M, Salam GP, Soyez G (2008) The catchment area of jets. JHEP 04:005. <https://doi.org/10.1088/1126-6708/2008/04/005>. arXiv: 0802.1188
- ATLAS Collaboration (2018) Performance of missing transverse momentum reconstruction with the ATLAS detector using proton–proton collisions at $\sqrt{s} = 13$ TeV. Eur Phys J C 78:903. <https://doi.org/10.1140/epjc/s10052-018-6288-9>. arXiv:1802.08168
- ATLAS Collaboration (2018) Measurement of the W -boson mass in pp collisions at $\sqrt{s} = 7$ TeV with the ATLAS detector. Eur Phys J C 78:110. <https://doi.org/10.1140/epjc/s10052-017-5475-4>. arXiv:1701.07240
- ATLAS Collaboration (2022) Measurements of the Higgs boson inclusive and differential fiducial cross-sections in the diphoton decay channel with pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector. JHEP 08:027. [https://doi.org/10.1007/JHEP08\(2022\)027](https://doi.org/10.1007/JHEP08(2022)027). arXiv:2202.00487
- ATLAS Collaboration (2021) Search for dark matter in events with missing transverse momentum and a Higgs boson decaying into two photons in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector. JHEP 10:013. [https://doi.org/10.1007/JHEP10\(2021\)013](https://doi.org/10.1007/JHEP10(2021)013). arXiv:2104.13240
- ATLAS Collaboration (2015) Observation and measurement of Higgs boson decays to WW^* with the ATLAS detector. Phys Rev D 92:012006. <https://doi.org/10.1103/PhysRevD.92.012006>. arXiv:1412.2641
- ATLAS Collaboration (2021) Performance of multi-threaded reconstruction in ATLAS. ATL-SOFT-PUB-2021-002. <https://cds.cern.ch/record/2771777>
- Elmsheuser J et al (2020) Evolution of the ATLAS analysis model for Run-3 and prospects for HL-LHC. EPJ Web Conf 245:06014. <https://doi.org/10.1051/epjconf/202024506014>
- Leggett C, Baines J, Bold T, Calafura P, Farrell S, Gemmeren P, Malon D, Ritsch E, Stewart G, Snyder S, Tsulaia V, Wynne B (2017) AthenaMT: upgrading the ATLAS software framework for the many-core world with multi-threading. J Phys Conf Ser 898:042009. <https://doi.org/10.1088/1742-6596/898/4/042009>
- CMS Collaboration (2017) Particle-flow reconstruction and global event description with the CMS detector. JINST 12:10003. <https://doi.org/10.1088/1748-0221/12/10/P10003>. arXiv:1706.04965
- ATLAS Collaboration. Performance of the reconstruction of large impact parameter tracks in the inner detector of ATLAS. submitted to EPIC <https://doi.org/10.48550/arXiv.2304.12867>. arXiv:2304.12867

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.