Geometric Graph Neural Network based track finding

Dolores Garcia[1,*], *Brieuc* Francois[1], and *Michele* Selvaggi[1]

[1]CERN

Abstract. An essential component of event reconstruction in particle physics experiments is identifying the trajectory of charged particles in the detector. Traditional methods for track finding are often complex, and tailored to specific detectors and input geometries, limiting their adaptability to new detector designs and optimization processes. To overcome these limitations, we present a novel, end-to-end track finding algorithm that is detector-agnostic and can take into account multiple input types. To achieve this, our approach unifies inputs from multiple sub-detectors and detector types into a single geometric algebra representation, simplifying data handling compared to traditional methods. Then, we leverage an equivariant graph neural network, GATr, to perform track finding across all data from an event simultaneously. We validate the effectiveness of our pipeline on various detector concepts with different technologies for the FCC-ee at CERN, the IDEA, and CLD detectors. This work generalizes track finding across diverse types of input geometric data and tracking technologies, facilitating the development of innovative detector concepts and enabling comprehensive detector optimization.

1 Introduction

The Future Circular Collider is one of the proposed future colliders that could follow from the Large Hadron Collider (LHC) at CERN. In the first stage, FCC-ee, an electron and a positron beam are accelerated to nearly the speed of light to collide in multiple interaction points. These collisions generate sprays of outgoing particles. Sophisticated detector systems, comprising hundreds of millions of sensors, are utilized to measure information about these particles. One of the main challenges is to identify and reconstruct the 3D trajectory and kinematic properties of charged particles propagating inside the detector's magnetic field (see Figure 1), which is referred to as tracking. Classic approaches solve this problem with combinatorial optimization methods such as Kalman filters [1]. However, these methods are detector-dependent and have a long development cycle, reducing their adaptability to new detector concepts. Novel approaches to tracking are needed to maximize the potential of future detectors. Tracking is performed in two stages: first, track finding, and then track fitting. Track finding is essentially an object instantiation task, which requires identifying groups of hits that form a track. Where the hits are the collection of measurements resulting from the particles interacting with the detector. Hits are usually positions, but they can be more complex geometries, such as shapes or directions. Track finding is a challenging problem: tracks can have different geometries and varying number of hits (10-100's), a track can have missing

*e-mail: dolores.garcia@cern.ch

hits in the trajectory, they can have hits in one or multiple sub-detectors, or an in-flight decay of a particle can produce an abrupt change in direction (kinked tracks) or disappear. Track finding has been approached in multiple ways. Global methods cluster hits by transforming the coordinates to a feature space in which relevant patterns are easier to detect [2]. Another approach is seeding and track following, usually implemented with a Combinatorial Kalman Filter [3]. This approach evaluates combinations of hits and iteratively builds tracks [4]. However, it is complex due to the large combinatorics, and it requires a detailed description of the geometry and materials of the experiment. In addition, when dealing with various types of input geometric data in each sub-detector, some works perform track finding for each sub-detector and then combine the track sections [5]. For example, the Belle II experiment has two tracking detectors, and its track finding approach is composed of eight different algorithms, including multiple algorithms for each tracking detector and for the combination [5]. Overall, these algorithms are detector-dependent and have a long development cycle.

Currently, there are new collider proposals with novel developments in detector technologies. For example, for the FCC-ee, there are several detector concepts under investigation [6, 7]. Each different tracking component can result in multiple input geometric types, depending on the measurement. As these new detector concepts are envisioned, it is important to have adaptable tracking pipelines with fast turnaround that can cope with different sub-detector concepts, geometric input data, variations, and optimizations thereof. Figure 1 shows two of the detectors that produce different track representations, requiring usually different algorithms.

Novel Graph Neural Network (GNN) based methods have been recently shown to perform competitively to classic pattern recognition approaches for a generic silicon detector while remaining scalable and not depending on the geometry implementation of the detector [8, 9]. However, so far they have only been applied to silicon tracking systems, which have pointcloud-like inputs. This work presents the detector agnostic Geometric Graph Track Finding (GGTF) method. This method is a generalized geometric track finding approach that: 1) can cope with multiple sub-detectors with different input geometries from multiple tracking technologies. 2) Does not require the geometry and materials specifications of the detector. 3) Does not rely on an analytical parametrization of the trajectories. Specifically, our end-to-end pipeline considers the hits from all tracking components and outputs a set of tracks. A critical new component is a geometric algebra representation of the data that allows multiple geometric types and a GNN, GATr [10], that exploits the symmetries of the detector through equivariance. The performance of the algorithm is assessed using two of the FCC-ee baseline detector concepts, the IDEA detector [7] and the CLD detector [6].

2 Preliminaries

Problem statement An event is a collection of measurements in a detector. The detector registers the signals deposited by the interaction of particles with its sensitive components, referred to as hits. The role of a tracking algorithm is to identify sets of hits produced by single particles and to derive information on the particle's trajectory based on those. For a set of hits in a given event, $\mathcal{X}$, composed of hits from different tracking components $\mathcal{X} = \{X_v, X_i, X_o, ...\}$, (vertex, inner tracker, outer tracker, endcap inner tracker,...) we want to obtain a set of tracks that approximates the target set of charged particles. The geometry of the detector is cylindrical, with a magnetic field along the z direction. We define the (x, y) as the transverse plane, and the z as the beam direction. Figure 1 shows an example set of hits for different detectors. In this work, this is framed as an object instantiation problem.

Trajectories In a detector with a homogeneous magnetic field, the trajectory of a particle forms a helix, with varying curvature depending on its momentum. Particles with higher

https://doi.org/10.1051/epjconf/202533701125

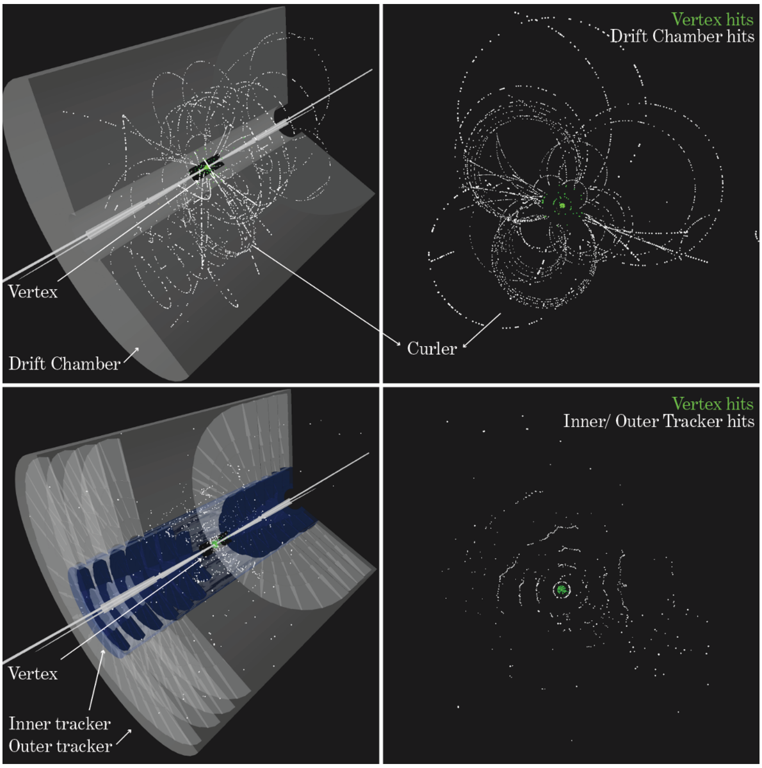

Figure 1. Top left: Simulated collision event showing the hits (in white) in the tracker of the IDEA detector. In the drift chamber volume, each hit is composed of two position coordinates due to the ambiguity in the measurement (more details are given in Section 3). Top right: front view of reconstructed hits. The circular shrinking pattern on the left corresponds to a low-energy particle curling inside the detector while losing energy. The straight tracks correspond to higher energy particles. Bottom left: reconstructed hits in the CLD detector of the same event, the vertex detector is shown in black, the inner tracker layers are shown in blue, and the outer tracker disks are shown in grey. Bottom right: front view of the hits in the CLD detector.

momentum result in an arc, while lower momentum particles can curl inside the detector. As they lose energy, the curvature of the helix increases, generating the shrinking displaced circles observed in Figure 1 top.

Representations The inputs from the different tracking detectors, as well as the geometries of the tracks, are detector-dependent. For a silicon detector, each hit $h = (\mathbf{x}, s)$ consists of a set of 3D coordinates $\mathbf{x} \in \mathbb{R}^3$ and a scalar s corresponding to the subdetector index. For other detector technologies, other information might be available. For example, in a drift chamber, a gaseous detector filled with wires that register ionization signals, each hit leads to two sets of coordinates, $(\mathbf{x}_{\text{left}}, \mathbf{x}_{\text{right}})$, due to inherent left-right ambiguity in the measurement. This richer data structure, which steps away from point-like track representations, makes the application of classical approaches complex.

In addition, tracks might have very different geometries in different detectors. For a silicon detector like CLD, a particle will leave around 10-15 hits per track [6], while for a drift chamber, each particle can generate O(100) hits. Not only is the number of hits different, but also the presented geometries. A drift chamber-like detector allows to capture particles of very low energy looping through the detector due to the magnetic field, as can be seen in Figure 1 left. Lower momentum particles have smaller curvature radius, leading to spiral trajectories, which we will refer to as curlers. These curlers represent a complex reconstruction task for classical algorithms, which usually focus on the high momentum regime (higher curvature radius tracks).

3 Geometric-based generalized track finding

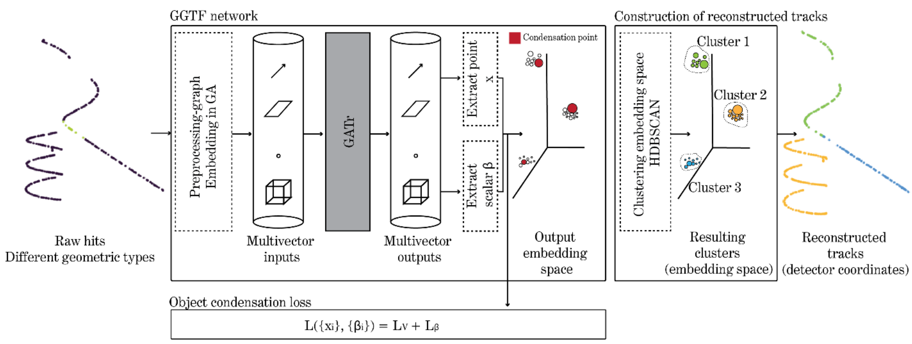

Figure 2. Overview of the GGTF end-to-end approach for track finding. The raw hits are represented as multivectors in the geometric algebra. The multivectors are transformed using the GATr. The output of the network is a coordinate in 3d space and a scalar, β, obtained from the output multivector by extracting the point and scalar components. Each ground truth track (object) should result in a high β node or condensation point in the embedding space, pictured filled in red. The outputs of the network are used by the object condensation loss to determine attractive and repulsive potentials between nodes to further allow a separable output embedding space as described in Section 3. Finally, a clustering step determines the track candidates by forming subsets of hits.

Our main contribution is the GGTF pipeline, which transitions from the hits of the different tracking detectors to a set of tracks. Our pipeline uses a novel geometric GNN, the Geometric Algebra Transformer (GATr), [10], which represents the inputs in a projective geometric algebra respecting equivariance, and the object condensation loss [11] to cluster hits belonging to the same object close in an embedding space, as in [8]. Unlike [8, 12], our architecture does not require edge filtering, which reduces the memory constraints. Also, our pipeline allows for more types of geometric inputs, unlocking this new approach for more tracking detectors such as the drift chamber.

Representation of hits in the graph As an initial step, the set of hits for each event is converted into a graph. However, unlike traditional approaches, the GATr architecture allows to represent the raw hits as different geometric objects and transformations in the projective geometric algebra $\mathcal{G}_{3,0,1}$. The elements of the algebra are 16-dimensional multivectors, which have direction, sign, and can be linearly combined [10]. This method allows for having as input planes, lines, points, or scalars. For example, we choose to represent the hits from the vertex detector as points (trivectors) and the hits from the drift chamber as vectors. This vector goes from the left to the right hit coordinates, therefore accounting for the ambiguity in the measurement.

Geometric Algebra Transformer The hits from the different systems are then unified in the multivector format, which represents the different geometrical objects in 16 dimensions.

Then, we use the GATr architecture as shown in Figure 2. It consists of multiple transformer blocks. Each block has a *LayerNorm*, an *equivariant multivector self-attention* layer, a second *LayerNorm*, an *equivariant multivector MLP* with *geometric bilinear interactions*, and another residual connection. This architecture is $E(3)$ equivariant. The $E(3)$-equivariant linear layers map multivectors using grade projections and multiplication with a homogeneous basis vector. The geometric bilinears allow for building more expressive networks by allowing grade mixing. The network outputs multivectors, which are used by the object condensation loss, defined below.

Loss function Modern approaches to reconstructing multiple objects focus on predicting a bounding box per object as an output. However, due to the helical trajectory of the tracks, it would be hard to parameterize these boxes. On the other hand, the object condensation approach for multi-object detection [11] allows to reconstruct an unknown number of objects through a dedicated loss function without the use of bounding boxes. The intuition behind this approach is as follows: the loss function requires the output of hits belonging to the same track to be close in a low-dimensional embedding space and those belonging to different tracks to be far away. In this way, in the embedding space, the tracks are easily separable. We consider the object condensation loss as presented in [11]. Starting from the output multivectors of the GATr network, we define the coordinates of each node as the point component extracted from the multivector and the β for each node as the scalar component. This is represented on the right side of Figure 2.

Final track reconstruction During inference, the extracted point component of each multivector, the output embedding, is clustered using the HDBSCAN clustering algorithm, as this algorithm is able to reconstruct clusters of different densities. Finally, the hits inside each output cluster define a reconstructed track as shown in Figure 2 right.

4 Experiments

Metrics In order to measure the performance of the various algorithms, each reconstructed track is matched to a ground truth particle to which it shares the largest number of hits.

- *The track hit purity* is the ratio of the number of hits in the track that belong to the MC particle and the total number of hits of the reconstructed track.
- The *track hit efficiency* is the ratio of the number of hits in the track that belong to the MC particle and the total number of hits this particle left in the detector.

The *tracking efficiency* is the probability to reconstruct a track. It can be defined in multiple ways, we will consider the following variants:

1. The percentage of reconstructable charged particles that lead to a reconstructed track with track hit purity greater than 75%, as in [6] .

2. The percentage of reconstructable charged particles with both ratios, the track hit purity and the track hit efficiency, above 50%, as defined in the accuracy tracking challenge [9]. Tracks are further separated into four categories. *Good*: both ratios above 50%, reconstructed in this definition. *Split*: track hit efficiency below 50%, track hit purity above 50%, so only a fraction of the track is reconstructed. *Multiple*: track hit efficiency above 50%, but track hit purity below 50%, typically due to the aggregation of multiple particles. *Bad*: both below 50%.

3. For the CLD detector, as there are only 10-15 hits per track, we consider the percentage of reconstructable charged particles that match with a reconstructed track with at least four hits (the minimum number of hits for track fitting, the second tracking step).

We evaluate the performance of our algorithm by showing the efficiency vs particle proximity and track displacement. *Particle proximity* is the smallest angular distance, Δ_{MC}, between the Monte Carlo particle associated with the track and any other MC particle. The angular distance between two particles can be defined as a function of the azimuthal angle ϕ and the pseudorapidity η as $\Delta_{MC} = \sqrt{\Delta\phi^2 + \Delta\eta^2}$. Reconstruction of *displaced tracks* is critical for the identification of long-lived particles. These tracks originate from a secondary vertex rather than from the primary interaction point, e.g, B and D mesons, which can travel a measurable distance before decaying. The displacement of a track can be measured by looking at the production vertex radius, i.e. $R = \sqrt{x^2 + y^2}$.

4.1 CLD detector

We start the demonstration of the GGTF pipeline and performance with the CLD detector. This detector is an all-silicon tracker; therefore, an easier case scenario as all hits are of the same type, and can be compared to a heuristic-based algorithm: the conformal tracking [2, 6]. The inputs are the hits left by the particles in the different subdetectors: vertex, inner tracker, and outer tracker. We generate a training dataset with 320k events resulting from a $e^+e^- \to Z \to q\bar{q}, q = u, d$ decay, at a center of mass energy of 91.2 GeV; a validation dataset with 10k events; and a regular evaluation set with 10k events for each detector concept.

In the top left and center, Figure 3, we show the track hit purity and track hit efficiency, respectively. Conformal tracking has a higher track hit purity but captures up to 40% fewer hits than GGTF in the low p_T regime. This shows in the different efficiency definitions. Using the definition based only on purity, definition 1, the tracking efficiency is slightly lower than the baseline for low p_T, shown in the bottom left of Figure 3. If we consider the second definition, based on track hit purity and efficiency, GGTF has an overall higher tracking efficiency, shown in the bottom center of Figure 3. The discontinuity observed at 700 MeV can be explained by the fact that this is the minimal transverse momentum needed to reach the outside of the tracker. Therefore, this threshold sets the limit for curlers. We use the tracking efficiency definition 3 to study the performance on displaced tracks and particle proximity. This definition allows for comparing both algorithms on equal grounds as it uses a looser definition of reconstructed track. GGTF outperforms the baseline for displaced tracks reconstruction, as shown in the top right of Figure 3. This is due to the conformal tracking algorithm assuming that all tracks originate from a common point. Therefore, these displaced tracks do not align with straight lines in the conformal plane, which leads to the failure of the conformal tracking algorithm. The percentage of fake tracks is very similar for both algorithms: 4.8% for GGTF and 5.2% for conformal tracking. Since the track fitting stage is not performed, the distribution per p_T is not available. Finally, the bottom right of Figure 3 shows the efficiency with the most inclusive definition 3 as a function of particle proximity. Overall, this follows the expected trend; if the closest track is further, it is easier to detect the tracks, and GGTF improves the performance for really close tracks over the baseline.

4.2 IDEA detector

Next, we turn towards a more complex detector involving multiple tracking technologies and producing more complex geometries. We study the IDEA detector, which has a silicon inner vertex detector and a drift chamber. The drift chamber is filled with gas and wires that collect the ionization signal of particles. Each particle can leave O(100) hits in the detector, which is an order of magnitude larger than for CLD. The IDEA detector does not have a baseline algorithm available. We therefore use these results to illustrate the algorithm's versatility in

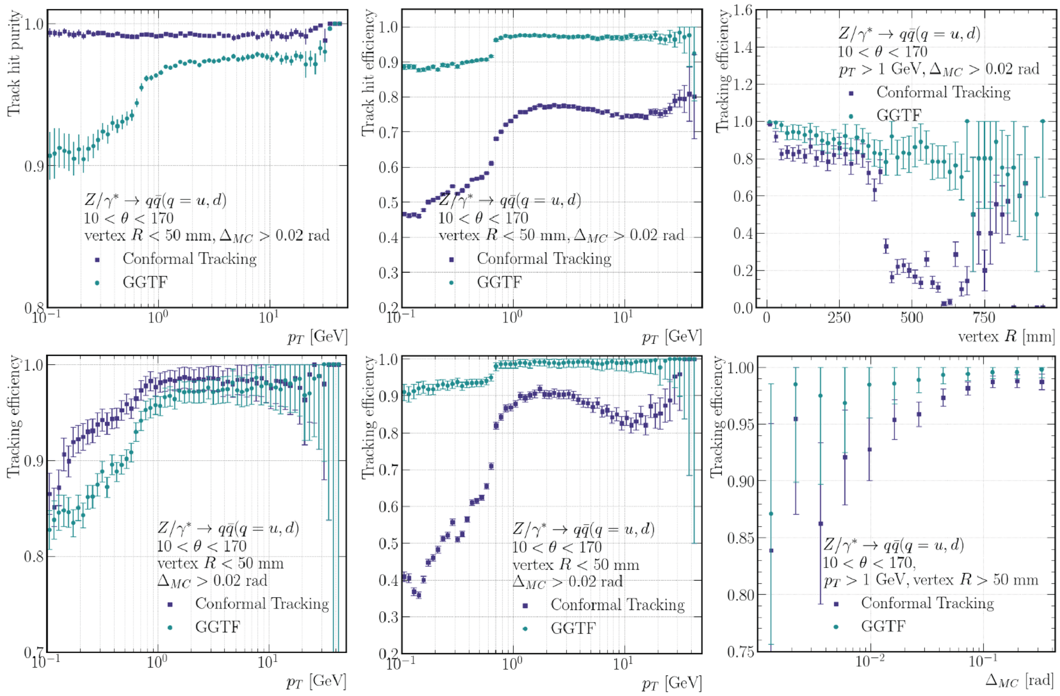

Figure 3. Results for the CLD detector. Top, left: track hit purity as a function of p_T. Center: track hit efficiency as a function of p_T. Right: tracking efficiency as a function of the production vertex radius (def. 3). Bottom left: tracking efficiency (def. 1) as a function of p_T. Center: tracking efficiency (def. 2) as a function of p_T. Right: tracking efficiency as a function of particle proximity.

terms of detector technology. The dataset decay and size are the same as those described in Section 4.1 for the CLD detector. The results are shown in Figure 4. Figure 4 shows that GGTF provides good tracking efficiency (definition 2). The curling limit of p_T is shown with a blue vertical line. As expected, the tracking efficiency is lower for lower p_T particles as these are curling in the detector, leaving very long tracks which are hard to recover fully. This is shown in the center of Figure 4. These results are comparable to the tracking efficiency shown for the Belle II detector [5].

5 Conclusions

GGTF is a high-efficiency general track finding method for multiple sub-detectors with different input geometries. It is able to find varied track geometries via an equivariant GNN. The success of GGTF indicates that highly parameterized algorithms are likely unnecessary and can be replaced with suitable GNNs at least in a low multiplicity environment in e+e- collisions. An important future step is to evaluate the algorithm's impact on the track fitting stage, as its higher track hit efficiency could improve the accuracy of the track parameters.

References

[1] K. Fujii, Extended kalman filter, Refernce Manual **14**, 41 (2013).

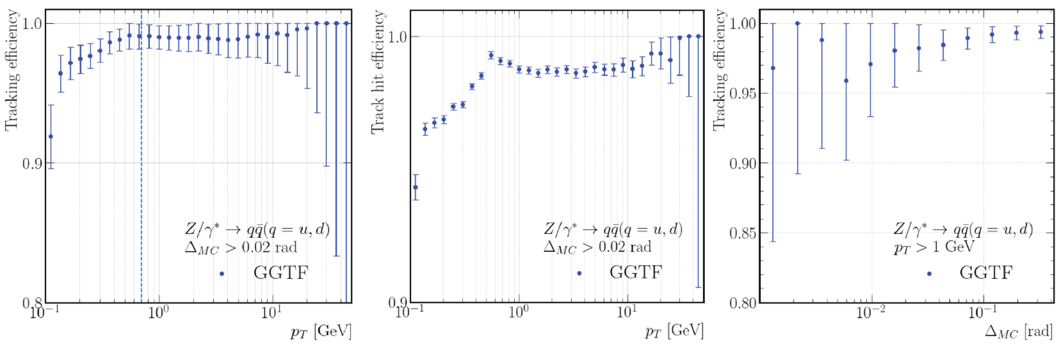

Figure 4. Results for the IDEA detector. Left: tracking efficiency (def. 2) as a function of p_T. Center: percentage of hits of the MC particle captured by the reconstructed track as a function of p_T. Right: tracking efficiency as a function of particle proximity.

[2] E. Brondolin, E. Leogrande, D. Hynds, F. Gaede, M. Petrič, A. Sailer, R. Simoniello, Conformal tracking for all-silicon trackers at future electron–positron colliders, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **956**, 163304 (2020).

[3] N. Braun, N. Braun, Combinatorial Kalman Filter, Combinatorial Kalman Filter and High Level Trigger Reconstruction for the Belle II Experiment pp. 117–174 (2019).

[4] X. Ai, C. Allaire, N. Calace, A. Czirkos, M. Elsing, I. Ene, R. Farkas, L.G. Gagnon, R. Garg, P. Gessinger et al., A common tracking software project, Computing and Software for Big Science **6**, 8 (2022).

[5] V. Bertacchi, T. Bilka, N. Braun, G. Casarosa, L. Corona, S. Cunliffe, F. Dattola, G. De Marino, M. De Nuccio, G. De Pietro et al., Track finding at Belle II, Computer Physics Communications **259** (2021).

[6] N. Bacchetta, J.J. Blaising, E. Brondolin, M. Dam, D. Dannheim, K. Elsener, D. Hynds, P. Janot, A. Kolano, E. Leogrande et al., CLD - A Detector Concept for the FCC-ee, https://arxiv.org/abs/1911.12230 (2019).

[7] F. Bedeschi, A detector concept proposal for a circular e+ e-collider, in *40th International Conference on High Energy Physics* (2021), p. 819

[8] X. Ju, D. Murnane, P. Calafiura, N. Choma, S. Conlon, S. Farrell, Y. Xu, M. Spiropulu, J.R. Vlimant, A. Aurisano et al., Performance of a geometric deep learning pipeline for HL-LHC particle tracking, The European Physical Journal C **81**, 1 (2021).

[9] S. Amrouche, L. Basara, P. Calafiura, V. Estrade, S. Farrell, D.R. Ferreira, L. Finnie, N. Finnie, C. Germain, V.V. Gligorov et al., The tracking machine learning challenge: accuracy phase (Springer, 2020)

[10] J. Brehmer, P. De Haan, S. Behrends, T.S. Cohen, Geometric algebra transformer, Advances in Neural Information Processing Systems **36** (2024).

[11] J. Kieseler, Object condensation: one-stage grid-free multi-object reconstruction in physics detectors, graph, and image data, The European Physical Journal C **80**, 1 (2020).

[12] K. Lieret, G. DeZoort, An Object Condensation Pipeline for Charged Particle Tracking at the High Luminosity LHC, in *EPJ Web of Conferences* (EDP Sciences, 2024), Vol. 295, p. 09004