A High-Performance Implementation of GNN-Based Trajectory Reconstruction on FPGA

Yun-Chen Yang*, Hao-Chun Liang†, and Bo-Cheng Lai*
*Institute of Electronics, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
†Institute of Pioneer Semiconductor Innovation, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

ABSTRACT

Field-programmable gate arrays (FPGAs) equipped with graph neural networks (GNNs) are a compelling platform for real-time charged-particle tracking at the High-Luminosity Large Hadron Collider (HL-LHC). Each event must be processed within $4\,\mu$s while sustaining more than 2.22 MHz throughput. Existing prototypes fall short of these targets and accelerate only the GNN inference stage, omitting the remainder of the reconstruction pipeline.

In this paper, we propose the first FPGA accelerator that performs complete GNN-based trajectory reconstruction. Our work unifies geometry-aware graph construction, low-latency GNN inference, and sequential track building in a single AXI-Stream pipeline. Its core contributions are: (i) a geometry-guided graph constructor that prunes spurious edges, (ii) a scheduling-optimized GNN engine that reduces inference latency by 52.3% through fine-grained pipelining and resource sharing, and (iii) a lightweight track builder that removes clustering overhead while improving resource efficiency and tracking accuracy. Design parameters allow the pipeline to be tuned for diverse HL-LHC scenarios.

Implemented on a Xilinx® Virtex UltraScale VU9P, our work achieves $2.36\,\mu$s end-to-end latency and 2.35 MHz sustained throughput, exceeding the HL-LHC Level-1 trigger requirements. Comprehensive benchmarks confirm scalability and position our work as a template for next-generation AI acceleration in data-intensive science.

1. INTRODUCTION

Charged–particle trajectory reconstruction underpins modern collider physics [1]. Silicon trackers at the Large Hadron Collider (LHC) [2] record spatial hits that must be associated online to form complete tracks. In the CMS Level-1 trigger (L1T) each event must be processed within $4\,\mu$s while sustaining 2.22 MHz per FPGA node in the time-multiplexed architecture [3]–[5]. The High-Luminosity upgrade (HL-LHC) [6] will further raise occupancy and data volume, pushing CPU and GPU solutions beyond their latency and power limits [7]–[9]. Deterministic-latency field-programmable gate arrays (FPGAs) therefore offer an attractive platform for end-to-end real-time tracking.

Graph neural networks (GNNs) capture hit relationships and already deliver excellent physics performance [7]–[19]. The *Interaction Network* (IN) [20] is a prominent example. A GNN tracker typically comprises:

1) **Graph construction** — map hits to nodes and create candidate edges between nearby layers,
2) **Edge classification** — assign to each edge the probability of belonging to a true track,
3) **Track building** — assemble high-confidence edges into trajectories.

Most prior work ignores the strict L1T budget [10]–[13], [15], [18], [19] or evaluates on CPUs/GPUs [7]–[9]. FPGA prototypes [14], [17] process only edge classification, leaving graph construction and track building off-chip and incurring communication overhead and extra latency. A fully integrated on-chip solution is therefore urgent.

HiGTR (High-Performance GNN-based Trajectory Reconstructor) executes the complete pipeline on a single FPGA. Its main contributions are:

- Hardware-optimized graph construction and track building: lightweight parallel units expand the geometric pruning of [7] to eliminate infeasible edges early,
- Low-latency GNN edge classification: a streaming scheduler derived from [17] removes redundant memory traffic, yielding a 2.09× speedup and reduced resource usage,
- End-to-end integration and scalability: three balanced stages are offered in scalable variants that partition workloads across multiple FPGAs for HL-LHC deployment.

Implemented on an AMD Xilinx Virtex UltraScale™ VU9P at 200 MHz, HiGTR sustains 2.35 MHz event throughput with $2.36\,\mu$s end-to-end latency, comfortably meeting HL-LHC L1T targets.

The remainder of this paper is organized as follows. Section 2. details the algorithms and optimizations. Section 3. presents the HiGTR architecture and FPGA implementation. Section 4. reports performance. Section 5. concludes and outlines future directions.

2. HARDWARE-EFFICIENT ALGORITHM

In this section, we introduce a three-stage, hardware-optimized pipeline engineered to satisfy rigorous resource and latency requirements. As illustrated in Fig. 1 (right), the proposed framework comprises:

1) Geometry-aware graph construction that prunes infeasible hit connections early,
2) Streaming edge classification using an Interaction Network (IN) tailored for low-latency inference, and
3) Deterministic track building that assembles validated edges into complete trajectories.

Together, these enhancements deliver substantial throughput improvements with minimal hardware overhead, achieving physics performance comparable to software baselines.

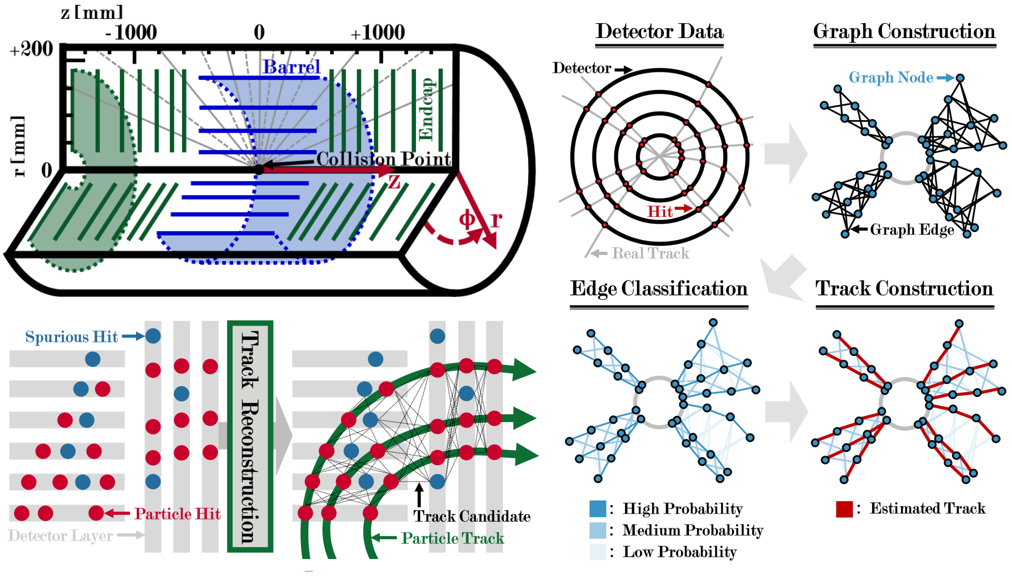

Fig. 1. Top-Left: Cross-section of a cylindrical collider detector (adapted from [14]). The interaction point is at the origin; the radial coordinate r and longitudinal axis z define geometry, and the azimuthal angle ϕ is measured around the beam axis. The barrel and endcaps together form the layered architecture required for precision tracking. Bottom-Left: Charged-particle track reconstruction in a multi-layer detector. Genuine hits (red) are interspersed with spurious measurements (blue). The pipeline links hits across layers, producing a validated candidate (black) and finally the reconstructed trajectory (green). Right: End-to-end pipeline comprising graph construction, edge classification, and track building.

2.1. Graph Construction

During the graph construction stage, detector measurements (*hits*) are transformed from Cartesian coordinates (x, y, z) to cylindrical coordinates (r, ϕ, z) using on-chip lookup tables (LUTs), each producing one tuple per clock cycle. Subsequently, hits are partitioned into multiple equally spaced azimuthal sectors. To constrain the search space, edges are formed only between hits that lie within the same sector or an adjacent sector on consecutive detector layers (see Fig. 2, upper right). We further apply radial and longitudinal cuts to remove edges that violate simple geometric constraints.

For each remaining candidate edge (i, j), we compute

$$\Delta r = r_j - r_i, \qquad \Delta\phi = \phi_j - \phi_i, \tag{1}$$

$$\Delta z = z_j - z_i, \qquad \Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}, \tag{2}$$

$$\eta = -\ln\tan\left(\tfrac{\theta}{2}\right), \qquad \theta = \operatorname{atan2}(r, z), \tag{3}$$

We intentionally omit the longitudinal-intercept constraint $|z_0| = |z_i - r_i\,(\Delta z/\Delta r)| < 15\,\text{cm}$ because detailed simulation shows it rejects only 0.003% of valid edges while saving one subtraction, one multiplication, and one division per edge. Overall, this pruning strategy reduces the number of candidate edges by factors of 9.13, 18.26, and 36.53 for total angular ranges of $\pi/4$, $\pi/2$, and π, respectively, compared to a naive all-to-all construction between adjacent layers [7].

2.2. Edge Classification

Following graph construction and pruning, the remaining edges are evaluated by an optimized Interaction Network (IN) [20]. This network consists of two relational modules, an edge-to-node aggregation step, and a node update module, formalized as

$$e'_{ij} = \phi_{R1}\big(v_i,\, v_j,\, e_{ij}\big), \qquad (i, j) \in \mathcal{E}, \tag{4}$$

$$a_j = \sum_{(i,j)\in\mathcal{I}(v_j)} e'_{ij}, \qquad v_j \in \mathcal{V}, \tag{5}$$

$$v'_j = \phi_O\big(v_j,\, a_j\big), \qquad v_j \in \mathcal{V}, \tag{6}$$

$$p_{ij} = \phi_{R2}\big(v'_i,\, v'_j,\, e'_{ij}\big), \qquad (i, j) \in \mathcal{E}, \tag{7}$$

where

$$v_i = \big(r_i,\, \phi_i,\, z_i\big), \quad e_{ij} = \big(\Delta r_{ij},\, \Delta\phi_{ij},\, \Delta z_{ij},\, \Delta R_{ij}\big).$$

Each of the functions ϕ_{R1}, ϕ_O, and ϕ_{R2} is implemented as an on-chip multilayer perceptron (MLP) using pipelined matrix–vector units. To sustain maximum streaming throughput, large RAM buffers between modules are replaced by lightweight data-level FIFOs, and redundant streams are collapsed via shared replication. Compared to the LL-GNN baseline [17], these enhancements achieve over 52% lower end-to-end latency and reduced overall resource utilization.

2.3. Track Building

In the final stage, HiGTR replaces costly quadratic clustering with a deterministic, sequential traversal that assembles full tracks in linear time. Each hit in layer L selects its highest-probability outgoing edge to layer $L+1$ and records the target index in a compact lookup table (see Fig. 2, mid right). To initiate track propagation, we introduce three seed categories:

1) *Barrel seeds*: Hits in the innermost barrel layer B_0;
2) *Endcap seeds*: Hits in endcap layer E_0 without any upstream connections;
3) *Mid-layer seeds*: Hits in intermediate layers that have no incoming edges.

By following these precomputed links, the algorithm extends each track in $O(N)$ time with negligible additional

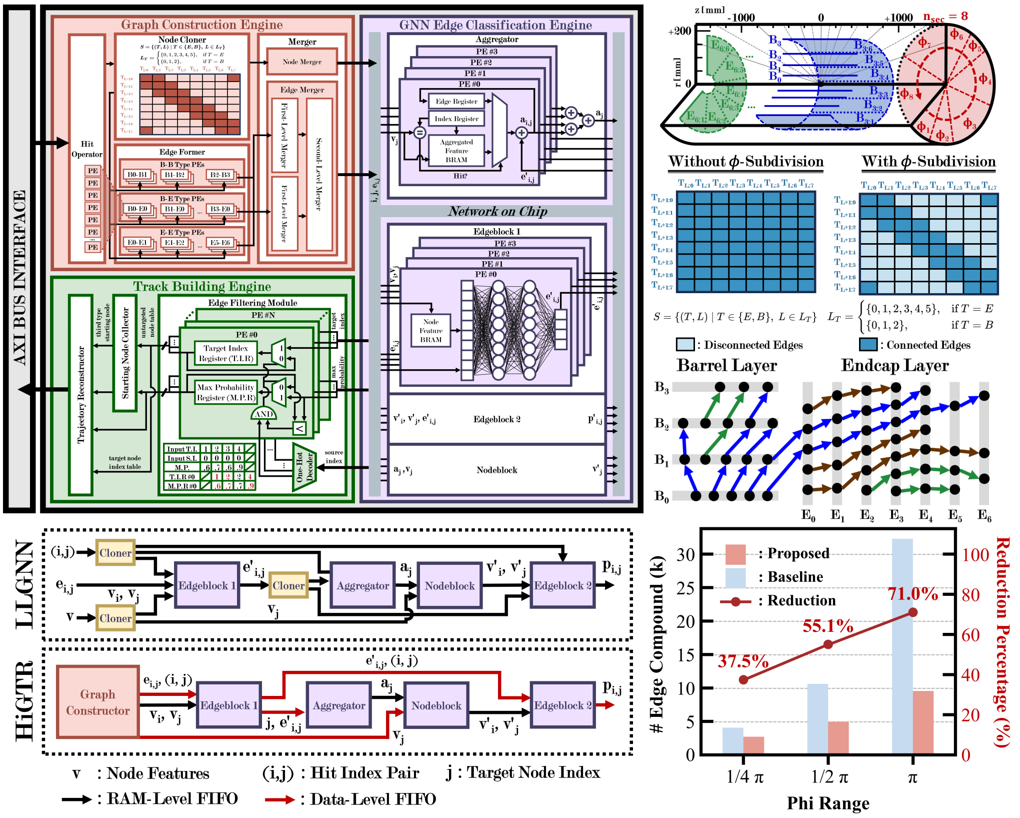

Fig. 2. Top-Left: HiGTR system architecture with three streaming stages connected by AXI interfaces. Bottom-Left: LL-GNN baseline (top) versus optimized HiGTR GNN implementation (bottom). Symbols follow standard IN notation (see Sec. II-B). HiGTR removes several *Cloner* modules and replaces RAM-level FIFOs with smaller data-level FIFOs, reducing storage and latency. Top-Right: Partitioning the detector into eight ϕ sectors sharply limits candidate edges. The left plot shows unconstrained connectivity; the right plot retains only connections within neighboring sectors, eliminating most geometrically implausible edges. Middle-Right: Sequential track-building seeds—barrel-first (blue), endcap-first (brown), and intermediate-layer (green)—covering different track origins. Bottom-Right: Edge-count reduction from ϕ subdivision. Blue bars: baseline; red bars: proposed; red line: percentage reduction. Wider ϕ ranges yield larger savings.

memory. This deterministic, low-latency design meets the stringent timing and resource constraints of Level-1 trigger pipelines.

3. HiGTR ARCHITECTURE

HiGTR adopts an event-level pipeline in which the Graph Construction Engine, the GNN Engine, and the Track Building Engine concurrently process events N, $N-1$, and $N-2$, respectively, to maximize throughput. Data and control signals between stages and the host system flow over standard AXI-Stream interfaces. To satisfy the HL-LHC Level-1 Trigger requirement of 2.22 MHz event rate—equivalent to an initiation interval below 450 ns (90 cycles at 200 MHz)—HiGTR balances workload across its three stages through architectural optimizations: fine-grained dataflow tuning using small FIFOs and ping-pong buffers for nonsequential access patterns, and stage-level balancing by allocating parallel processing elements (PEs) to each computational task based on workload profiling, thereby avoiding bottlenecks.

Figure 2(top-left) depicts the hardware implementation. The **Graph Construction Engine** comprises parallel *Hit Operators* that perform Cartesian-to-cylindrical coordinate transformation using on-chip lookup tables (LUTs) and partition hits into ϕ-sectors. *Node Cloner* modules replicate hit data for parallel processing by the *Edge Former* PEs, which compute geometric features and apply layer-pair–dependent cuts (B-B, B-E, E-E). An *Edge Merger* then consolidates valid edges into an ordered stream. The **GNN Engine** implements the optimized Interaction Network (Sec. 2.-B; Fig. 2 bottom-left), overlapping execution between blocks (e.g., streaming Edge-Block1 output to the Aggregator) and sharing data efficiently via lightweight FIFOs. The **Track Building Engine** includes an *Edge Filter* composed of parallel units that select the highest-probability outgoing edge per source node to build the target-index table, and a *Trajectory Reconstructor* that executes the sequential seeding and extension logic outlined in Sec. 2.-C. By tuning PE counts and schedule, all stages achieve equal latency such that a new event launches every 85 cycles (425 ns), corresponding to a sustained throughput of 2.35 MHz at a 200 MHz clock frequency.

4. EXPERIMENTAL RESULTS

We evaluate HiGTR using the TrackML dataset [21], [22], which emulates HL-LHC conditions with emphasis on the high-granularity inner pixel layers following the setup in [7]. The detector geometry, illustrated in Figure 1 (top-left), is partitioned into azimuthal (ϕ) sectors [14]. We explore three sector widths ($\pi/4$, $\pi/2$, and π), corresponding to increasing data volumes per FPGA. In all cases, we apply physics selection cuts requiring transverse momentum $p_T > 2\,\text{GeV}$ to match CMS Level-1 trigger criteria [4], [5].

Table I reports resource utilization (LUTs and BRAMs) alongside measured latency and throughput for each sector width on the Xilinx VU9P. Processing elements (PEs) were allocated to satisfy the 90-cycle initiation-interval target derived from the Level-1 trigger throughput requirement (see Section 3.), based on 95th-percentile hit and edge counts observed in TrackML for each subdivision.

The $\pi/4$ configuration achieves a latency of $2.36\,\mu$s and a sustained throughput of 2.35 MHz, thereby satisfying both the sub-$4\,\mu$s latency and the > 2.22 MHz throughput requirements. After minor PE rebalancing to accommodate implementation overheads, the $\pi/2$ variant attains $2.60\,\mu$s latency and 2.24 MHz throughput, also meeting the Level-1 trigger targets. The coarsest partition (π) processes a substantially larger data volume and achieves $3.80\,\mu$s latency; however, its throughput is limited to 1.53 MHz due to resource-constrained PE parallelism. This shortfall indicates that larger FPGAs or additional time-multiplexing stages are necessary to sustain target throughput under this scenario.

Figure 2 (bottom-right) illustrates the impact of azimuthal subdivision on reducing the initial candidate-edge count, highlighting its importance for computational load management across all configurations.

TABLE I
Measured performance across sectorizations on VU9P.

Sectors	LUT	BRAM	Latency (μs)	Throughput (MHz)
$\pi/4$	382 k	780	2.36	2.35
$\pi/2$	411 k	812	2.60	2.24
π	463 k	921	3.80	1.53

5. CONCLUSION

This paper presented *HiGTR*, the first end-to-end FPGA accelerator for GNN-based trajectory reconstruction in high-energy physics. HiGTR fuses geometry-guided graph construction, latency-balanced GNN inference, and sequential track building in a deeply pipelined, hierarchically parallel architecture. Implemented on an AMD Xilinx VU9P, it sustains 2.35 MHz at an end-to-end latency of 2.36 μs, satisfying the Level-1 trigger targets foreseen for the HL-LHC. Geometry-driven edge pruning, a streamlined GNN pipeline, and an efficient track builder remove redundant computation and overcome the scalability limits of earlier FPGA solutions.

This work shows that neural-network tracking is practical for future HL-LHC triggers. Next steps include (i) scaling to larger or multi-FPGA systems, (ii) adopting richer network architectures to enhance physics performance, and (iii) retargeting the accelerator to alternative detector geometries. These efforts will evolve HiGTR into a next-generation trigger platform that combines state-of-the-art AI with sub-microsecond timing guarantees for modern high-energy-physics experiments.

6. REFERENCES

[1] A. Ryd and L. Skinnari, “Tracking triggers for the hl-lhc,” *Annual Review of Nuclear and Particle Science*, vol. 70, no. 1, p. 171–195, Oct. 2020. [Online]. Available: http://dx.doi.org/10.1146/annurev-nucl-020420-093547

[2] L. Evans, “The large hadron collider,” *New Journal of Physics*, vol. 9, no. 9, p. 335, 2007.

[3] F. Pastore, “Level-1 trigger systems for lhc experiments,” in *IFAE 2007: Incontri di Fisica delle Alte Energie Italian Meeting on High Energy Physics Napoli, 11–13 April 2007*. Springer, 2008, pp. 303–307.

[4] “The Phase-2 Upgrade of the CMS Tracker,” CERN, Geneva, Tech. Rep., 2017. [Online]. Available: https://cds.cern.ch/record/2272264

[5] “The Phase-2 Upgrade of the CMS Level-1 Trigger,” CERN, Geneva, Tech. Rep., 2020, final version. [Online]. Available: https://cds.cern.ch/record/2714892

[6] O. Aberle, C. Adorisio, A. Adraktas, M. Ady, J. Albertone, L. Alberty, M. Alcaide Leon, A. Alekou, D. Alesini, B. Almeida Ferreira *et al.*, “High-luminosity large hadron collider (hl-lhc): Technical design report,” 2020.

[7] G. DeZoort, S. Thais, J. Duarte, V. Razavimaleki, M. Atkinson, I. Ojalvo, M. Neubauer, and P. Elmer, “Charged particle tracking via edge-classifying interaction networks,” *Comput. Softw. Big Sci.*, vol. 5, no. 1, pp. 1–13, 2021.

[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*, vol. 81, pp. 1–14, 2021.

[9] J. Pata, J. Duarte, J.-R. Vlimant, M. Pierini, and M. Spiropulu, “Mlpf: efficient machine-learned particle-flow reconstruction using graph neural networks,” *The European Physical Journal C*, vol. 81, pp. 1–14, 2021.

[10] C. Biscarat, S. Caillou, C. Rougier, J. Stark, and J. Zahreddine, “Towards a realistic track reconstruction algorithm based on graph neural networks for the hl-lhc,” in *EPJ Web of Conferences*, vol. 251. EDP Sciences, 2021, p. 03047.

[11] J. D. Burleson, J. Chan, P. Calafiura, H. Torres, C. Rougier, S. Caillou, M. T. Pham, J. Stark, D. T. Murnane, M. Neubauer *et al.*, “Physics performance of the atlas gnn4itk track reconstruction chain,” ATL-COM-SOFT-2023-138, Tech. Rep., 2023.

[12] S. Caillou, P. Calafiura, C. Rougier, J. Stark, D. T. Murnane, A. Vallier, X. Ju, and S. A. Farrell, “Atlas itk track reconstruction with a gnn-based pipeline,” ATL-COM-ITK-2022-057, Tech. Rep., 2022.

[13] N. Choma, D. Murnane, X. Ju, P. Calafiura, S. Conlon, S. Farrell, G. Cerati, L. Gray, T. Klijnsma, J. Kowalkowski *et al.*, “Track seeding and labelling with embedded-space graph neural networks,” *arXiv preprint arXiv:2007.00149*, 2020.

[14] A. Elabd, V. Razavimaleki, S.-Y. Huang, J. Duarte, M. Atkinson, G. DeZoort, P. Elmer, S. Hauck, J.-X. Hu, S.-C. Hsu *et al.*, “Graph neural networks for charged particle tracking on fpgas,” *Frontiers in big Data*, vol. 5, p. 828666, 2022.

[15] S. Farrell, P. Calafiura, M. Mudigonda, D. Anderson, J.-R. Vlimant, S. Zheng, J. Bendavid, M. Spiropulu, G. Cerati, L. Gray *et al.*, “Novel deep learning methods for track reconstruction,” *arXiv preprint arXiv:1810.06111*, 2018.

[16] A. Heintz, V. Razavimaleki, J. Duarte, G. DeZoort, I. Ojalvo, S. Thais, M. Atkinson, M. Neubauer, L. Gray, S. Jindariani *et al.*, “Accelerated charged particle tracking with graph neural networks on fpgas,” *arXiv preprint arXiv:2012.01563*, 2020.

[17] S. Huang, Y. Yang, Y. Su, B. Lai, J. Duarte, S. Hauck, S. Hsu, J. Hu, and M. S. Neubauer, “Low latency edge classification gnn for particle trajectory tracking on fpgas,” in *2023 33rd International Conference on Field-Programmable Logic and Applications (FPL)*. Los Alamitos, CA, USA: IEEE Computer Society, sep 2023, pp. 294–298. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/FPL60245.2023.00050

[18] X. Ju, S. Farrell, P. Calafiura, D. Murnane, L. Gray, T. Klijnsma, K. Pedro, G. Cerati, J. Kowalkowski, G. Perdue *et al.*, “Graph neural networks for particle reconstruction in high energy physics detectors,” *arXiv preprint arXiv:2003.11603*, 2020.

[19] C.-Y. Wang, X. Ju, S.-C. Hsu, D. Murnane, P. Calafiura, S. Farrell, M. Spiropulu, J.-R. Vlimant, A. Aurisano, J. Hewes *et al.*, “Reconstruction of large radius tracks with the exa. trkx pipeline,” in *Journal of Physics: Conference Series*, vol. 2438, no. 1. IOP Publishing, 2023, p. 012117.

[20] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende *et al.*, “Interaction networks for learning about objects, relations and physics,” *Advances in neural information processing systems*, vol. 29, 2016.

[21] M. Kiehn, S. Amrouche, P. Calafiura, V. Estrade, S. Farrell, C. Germain, V. Gligorov, T. Golling, H. Gray, I. Guyon *et al.*, “The trackml high-energy physics tracking challenge on kaggle,” in *EPJ Web of Conferences*, vol. 214. EDP Sciences, 2019, p. 06037.

[22] 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.