ML-QTRACK: LEARNING-GUIDED GLOBAL OPTIMIZATION FOR FULL-DETECTOR PARTICLE TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Charged-particle track reconstruction is a fundamental yet challenging problem in high-energy physics, with full-detector tracking posing a particularly difficult trade-off between reconstruction quality and computational cost. Learning-based methods can efficiently process large candidate graphs but may produce substantial fake trajectories, while global optimization enforces stronger trajectory-level consistency at a higher computational cost. We propose **ML-QTrack**, a learning-guided global optimization framework that combines graph purification with trajectory-level QUBO optimization. On the full-event, full-detector TrackML benchmark, ML-QTrack achieves 98.1% efficiency and reduces the fake rate from 36.71% to 6.87% compared with an HGNN-based method using the same candidate graph, while completing reconstruction in 1.68 s per event. These results show that learning-guided hypothesis reduction can make trajectory-level global optimization computationally practical for large-scale full-detector tracking, while substantially improving reconstruction quality.

1 INTRODUCTION

Charged-particle track reconstruction is a fundamental task in modern high-energy physics Foundation et al. (2019). In collider experiments such as the Large Hadron Collider (LHC), charged particles traverse highly granular tracking detectors and leave sequences of localized measurements, referred to as *hits* Evans & Bryant (2008); Collaboration (2012). The goal of track reconstruction is to infer the underlying particle trajectories from these detector measurements, making it a large-scale structured inference problem involving highly ambiguous hit associations. The increasing collision rates expected at the forthcoming High-Luminosity LHC (HL-LHC) will further increase detector occupancy and reconstruction complexity, motivating the development of scalable and accurate tracking algorithms Alonso et al. (2020); Collaboration (2020); Software & Computing (2022).

Traditional reconstruction pipelines incrementally extend particle trajectories through local geometric compatibility using techniques such as combinatorial seeding, Cellular Automata, and Kalman filter Cornelissen et al. (2008); Cucciarelli et al. (2006); Funke et al. (2014). More recently, learning-based approaches have substantially advanced graph-based particle tracking through high-recall candidate graphs and learned edge-level prediction, enabled by techniques such as metric learning, graph neural networks (GNNs), object condensation, and Transformer architectures Cerati et al. (2018); Farrell et al. (2018); Van Stroud et al. (2024); Ju et al. (2020); Liu et al. (2023). Despite their methodological differences, both paradigms ultimately reconstruct complete trajectories by aggregating local inferred hit associations and therefore fundamentally follow the same local inference paradigm. Under dense full-event reconstruction, however, multiple locally plausible associations may coexist, resulting in duplicate trajectories, mixed-particle assignments, and unresolved ambiguities Collaboration (2020); Software & Computing (2022). Consequently, local inference alone cannot explicitly enforce global trajectory consistency, making globally consistent reconstruction increasingly difficult under realistic full-detector conditions.

To overcome these limitations, a complementary perspective formulates tracking as a global combinatorial optimization problem. Quadratic Unconstrained Binary Optimization (QUBO) provides a natural framework for jointly optimizing candidate trajectory hypotheses while encoding global consistency through penalty terms in the objective function Bapst et al. (2020). Existing studies have demonstrated the effectiveness of QUBO-based optimization for improving global trajectory consis-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tency Bapst et al. (2020); Okawa et al. (2024b; 2025); Pastorello & Blanzieri (2019). However, these methods have largely been evaluated under simplified reconstruction settings, typically relying on detector-specific formulations, handcrafted candidate generation procedures, or restricted detector regions. Consequently, it remains unclear whether QUBO-based global optimization can effectively complement modern local inference under realistic full-detector reconstruction conditions.

These observations suggest that learning and optimization address complementary aspects of the tracking problem. Learning-based methods excel at constructing high-recall candidate graphs, whereas combinatorial optimization provides principled mechanisms for enforcing global trajectory consistency. Realizing this complementary paradigm requires overcoming two key challenges: (1) dense candidate graphs generated for high recall contain optimization-irrelevant associations that severely limit optimization scalability; and (2) existing QUBO formulations rely on detector-specific assumptions and therefore cannot be directly applied to realistic full-detector reconstruction.

To address these challenges, we propose **ML-QTrack**, a learning-guided global optimization framework for realistic full-detector particle tracking. Starting from the high-recall metric-learning candidate graph generated by the GNN4ITk pipeline Caillou et al. (2022), ML-QTrack first employs a LightGBM-based graph purification module to remove optimization-irrelevant candidate edges while preserving reconstruction recall, substantially reducing downstream optimization complexity. We further develop a topology-independent QUBO formulation based on local geometric consistency rather than detector-specific layer-order assumptions, enabling unified trajectory optimization across both barrel and endcap detector regions. Finally, we introduce a truth-free trajectory recovery procedure to resolve detector-induced ambiguities under realistic reconstruction conditions.

Extensive experiments on the TrackML benchmark demonstrate that ML-QTrack achieves 98.1% reconstruction efficiency with a 6.87% fake rate for tracks with transverse momentum above $1\,\mathrm{GeV}$ and at least five associated hits. Compared with a representative HGNN-based tracking pipeline operating on the same metric-learning candidate graph, ML-QTrack reduces the fake rate by over 80% while maintaining comparable reconstruction efficiency. The complete reconstruction takes 1.68 ± 0.10 s per event on a 6-core CPU with 32 GB memory, demonstrating the practical scalability of trajectory-level global optimization to realistic full-detector events. These results provide empirical evidence that learning-guided global trajectory optimization can extend QUBO-based particle tracking from simplified settings to realistic full-detector reconstruction, combining learned hypothesis reduction with explicit combinatorial reasoning for large-scale scientific inference. Our contributions are summarized as follows:

- We establish learning-guided global optimization as a practical paradigm for full-detector particle tracking, demonstrating that trajectory-level global optimization can complement local inference for globally consistent reconstruction.
- We develop a topology-independent QUBO formulation based on local geometric consistency and introduce a learning-guided graph purification strategy that preserves high reconstruction recall while substantially reducing the combinatorial search space.
- We validate ML-QTrack on realistic full-detector TrackML events, achieving high reconstruction efficiency and low fake rates with second-scale end-to-end runtime.

2 Trajectory Representation and Problem Formulation

Building upon the structured inference problem introduced in Section 1, we formulate full-event particle tracking as a global trajectory hypothesis selection problem.

The TrackML detector comprises a central barrel and forward/backward endcap regions (Fig. 1), providing nearly full angular coverage for charged particles. Under realistic HL-LHC conditions, dense trajectory overlap near the interaction point creates substantial ambiguity in hit associations. Consequently, the goal is not merely to identify locally compatible hit associations, but to recover a globally consistent set of trajectories from a large number of competing trajectory hypotheses.

Hits and Trajectory Representation. Formally, a detector hit is represented as:

$$h_i = (x_i, y_i, z_i, \ell_i), \tag{1}$$

where (x_i, y_i, z_i) denote the hit coordinates and ℓ_i denotes the detector layer identifier. Let $\mathcal{V} = \{h_1, h_2, \ldots, h_N\}$ denote the hits in a collision event. To facilitate trajectory-level reason-

ing, higher-order trajectory structures are constructed from hits. A candidate *doublet* is defined as $d_{ij} = (h_i, h_j)$, where h_i and h_j belong to different detector layers and are ordered by increasing radial distance. Building upon doublets, a candidate *triplet* is defined as $t_{ijk} = (h_i, h_j, h_k)$, formed by two consecutive doublets, representing a locally consistent trajectory segment over three hits.

Trajectory Hypotheses. Compared with pairwise hit associations, triplets encode richer geometric information, including local trajectory continuity, directional consistency, and curvature characteristics. We therefore view each triplet as a *local trajectory hypothesis* and a complete particle trajectory as a chain of mutually compatible triplets. Let $\mathcal{T} = \{T_1, \ldots, T_M\}$ denote the set of candidate triplets, where each T_m corresponds to a specific triplet t_{ijk}. Neighboring trajectory hypotheses may either support the continuation of the same physical trajectory or provide competing explanations for the detector measurements.

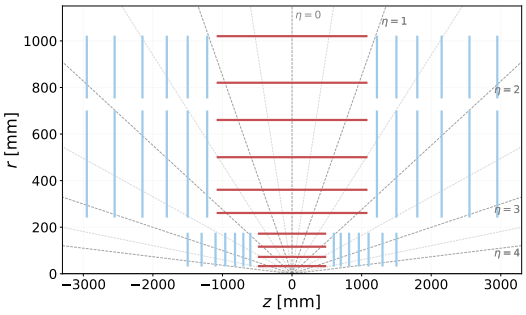


Figure 1: TrackML detector geometry in the r–z plane, showing barrel (red) and endcap (blue) detector layers, together with lines of constant pseudorapidity η.

Track Reconstruction as Combinatorial Optimization. Under the trajectory-hypothesis view, particle tracking can be formulated as selecting a subset $\mathcal{T}^* \subseteq \mathcal{T}$ that provides the most plausible explanation of the detector measurements. This selection process can be naturally represented using binary decision variables, indicating whether each trajectory hypothesis is selected in the final reconstruction. A valid reconstruction should simultaneously satisfy two fundamental requirements: (1) selected trajectory hypotheses should be geometrically compatible and collectively form smooth charged-particle trajectories; and (2) incompatible trajectory hypotheses should not coexist in the final solution, thereby preventing conflicting assignments of local trajectory structures.

Consequently, the full-event track reconstruction is cast as a large-scale combinatorial optimization problem over mutually interacting trajectory hypotheses. This formulation provides the conceptual foundation for the learning-guided global optimization framework developed in the next section.

3 Learning-Guided Structured Optimization

As illustrated in Fig. 2, ML-QTrack formulates full-detector particle tracking as a learning-guided structured inference framework. Starting from full-detector hits in a TrackML event, ML-QTrack first constructs a graph through learning-guided graph construction and hypothesis pruning. The pruned graph is then converted into a triplet-based QUBO formulation to select globally consistent trajectory hypotheses. A lightweight trajectory-level refinement step is subsequently applied to merge duplicate trajectories and reconnect fragmented segments, producing the final tracks.

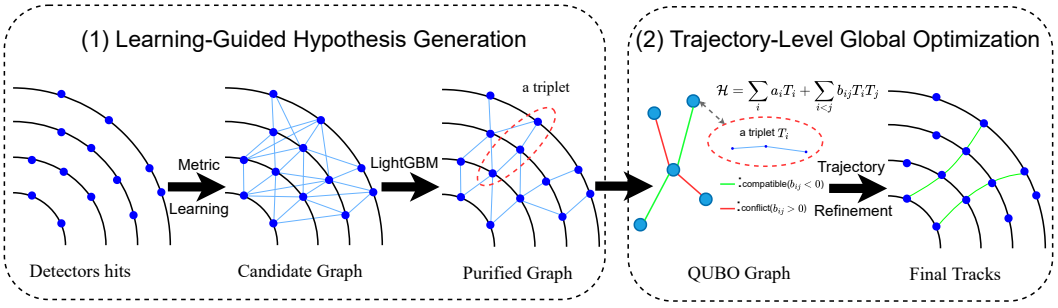


Figure 2: Overview of ML-QTrack for full-detector particle tracking. The framework consists of two main stages: (1) learning-guided hypothesis space construction to generate a high-recall, optimization-friendly trajectory hypothesis space; and (2) trajectory-level global optimization based on a triplet QUBO formulation for globally consistent selection. A lightweight trajectory refinement step is then applied for duplicate merging and fragment reconnection to produce the final tracks.

3.1 LEARNING-GUIDED HYPOTHESIS SPACE CONSTRUCTION

The first stage of ML-QTrack aims to construct a compact yet high-recall hypothesis space for downstream trajectory optimization. Directly optimizing over all detector hits in a realistic full TrackML event would lead to a prohibitively large combinatorial search space. We adopt a two-step learning-guided hypothesis generation strategy: an initial metric-learning stage is used to produce a high-recall candidate graph, and a subsequent graph-purification stage removes candidate associations that would otherwise interfere with higher-order trajectory construction and QUBO optimization.

Metric-learning candidate graph construction. Following the standard TrackML reconstruction pipeline used in the CommonFramework benchmark Caillou et al. (2022), we first construct an initial candidate graph using the publicly available metric-learning model. Given a full event of detector hits, the metric-learning stage produces a sparse graph in which detector hits are treated as nodes and candidate hit associations are represented as edges. The model is intentionally optimized for high recall and therefore preserves most physically valid local trajectory segments, making it a suitable front-end for full-event track reconstruction.

However, high recall alone is insufficient for downstream optimization. Residual candidate edges, although often physically plausible, may introduce excessive triplet candidates and QUBO couplings, resulting in an unnecessarily large and ambiguous optimization space.

Optimization-aware graph purification. To control the combinatorial complexity of downstream trajectory optimization, we introduce an optimization-aware graph purification stage implemented using LightGBM. For each candidate edge generated by the metric-learning stage, the classifier estimates its retention probability for subsequent trajectory construction. Edges with confidence below a predefined threshold are removed before triplet generation.

The role of graph purification is not merely to improve edge-level classification quality, but to control the combinatorial complexity of the downstream optimization problem. Many residual candidate edges have limited impact on edge metrics, yet they can dramatically increase the number of admissible triplets, quadruplets, and pairwise QUBO couplings. Removing these interfering edges yields a cleaner optimization graph while preserving the majority of physical trajectory hypotheses.

Since candidate edges within and across detector regions exhibit different geometric characteristics, we train separate classifiers for different edge categories. The geometric features used by the classifiers are summarized in Table 5 in Appendix A.2. The resulting purified graph defines a compact trajectory hypothesis space for the trajectory-level QUBO optimization described next.

3.2 TRAJECTORY-LEVEL GLOBAL OPTIMIZATION

Although graph purification reduces the number of spurious candidate edges, track reconstruction remains a global structured inference problem, where multiple locally plausible trajectory hypotheses may coexist. Following the trajectory representation introduced in Section 2, we formulate trajectory reconstruction as a global selection problem over candidate hypotheses.

Existing QUBO-based approaches have demonstrated the potential of global optimization for charged-particle tracking Bapst et al. (2020); Zlokapa et al. (2021). However, doublet-based formulations provide limited geometric constraints for trajectory-level reasoning Zlokapa et al. (2021), while triplet-based formulations introduce a rapidly growing number of optimization variables and pairwise couplings as the hypothesis space expands, previous triplet-based formulations developed for barrel-dominated detector geometries also rely on detector-specific normalization terms based on missing detector layers (holes) Bapst et al. (2020). Such quantities become difficult to define consistently for trajectories traversing heterogeneous barrel–endcap layouts.

To address these limitations, we design the QUBO formulation using local geometric consistency rather than detector-specific layer statistics. The trajectory selection problem over candidate triplets is formulated as:

$$\min_{\mathbf{x}} H(\mathbf{x}) = \sum_{i=1}^{M} a_i x_i + \sum_{i<j}^{M} b_{ij} x_i x_j, \tag{2}$$

where $\mathbf{x} = (x_1, \cdots, x_M)$ is the vector of binary variables with $x_i \in \{0, 1\}$ indicating whether triplet T_i is selected in the final reconstruction. The unary coefficient a_i evaluates the intrinsic geometric

consistency of an individual triplet, while the pairwise coefficient b_{ij} models the compatibility or conflict between two triplets. The two components are described in detail below.

Unary triplet score. The unary coefficient a_i captures the intrinsic consistency of triplet T_i based on its intra-triplet longitudinal smoothness in the r–z plane. Specifically, let $\Delta L_i^{\text{int}} = \left|\theta_{L,i}^{(2)} - \theta_{L,i}^{(1)}\right|$, where $\theta_{L,i}^{(1)}$ and $\theta_{L,i}^{(2)}$ denote the local polar angles in the r–z plane of the two constituent doublets forming triplet T_i. The unary coefficient is then defined as:

$$a_i = \frac{\Delta L_i^{\text{int}}}{\Delta L_i^{\text{int}} + \tau_a}, \tag{3}$$

where $\tau_a > 0$ is a regularization constant. Triplets with smoother longitudinal evolution exhibit smaller ΔL_i^{int} values and lower unary penalties under QUBO minimization. This term provides a lightweight geometric prior for individual triplets, whereas transverse curvature consistency is deferred to the pairwise interaction term, where it is evaluated between trajectory continuations.

Pairwise triplet interaction. The pairwise coupling coefficient b_{ij} captures the interaction between two triplets and enforces both trajectory continuity and hit assignment exclusivity. For compatible triplets that form a valid quadruplet, an attractive coupling encourages trajectory continuation. The strength of this coupling is determined by a compatibility score S_{ij}, which measures the local geometric consistency between the triplets T_i and T_j. In contrast, conflicting triplets receive repulsive penalties to prevent incompatible triplets from being selected simultaneously.

For triplets forming a valid quadruplet, we define the compatibility score S_{ij} based on local geometric consistency in both the longitudinal (r–z) and transverse (x–y) planes:

$$S_{ij} = \alpha \left(1 - \min\left(\frac{\Delta L_{ij}}{\delta_L^{\max}}, 1\right)\right) + (1-\alpha)\left(1 - \min\left(\frac{\Delta C_{ij}}{\delta_C^{\max}}, 1\right)\right), \tag{4}$$

where ΔL_{ij} denotes the maximum longitudinal mismatch of triplets T_i and T_j in the r–z plane when forming the quadruplet continuation, and $\Delta C_{ij} = \left|\delta(q/p)_{T_i} - \delta(q/p)_{T_j}\right|$ measures the inconsistency of transverse curvature between the two triplets. Here, $\delta(q/p)_{T_i}$ denotes the transverse-curvature quantity estimated from the three hit coordinates of triplet T_i in the (x)–(y) plane. The normalization thresholds $\delta_L^{\max}$ and $\delta_C^{\max}$ represent the maximum admissible longitudinal and transverse-curvature mismatch, respectively. The parameter $\alpha \in [0, 1]$ balances the contribution from the two geometric components. We use $\alpha = 0.5$; its sensitivity is discussed in Appendix E.

Based on the compatibility between triplets, we define the pairwise coupling coefficient as:

$$b_{ij} = \begin{cases} -S_{ij}, & \text{if } T_i \text{ and } T_j \text{ form a valid quadruplet,} \\ \zeta, & \text{if } T_i \text{ and } T_j \text{ are in conflict,} \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where $\zeta > 0$ is a repulsive penalty coefficient. Here, two triplets are considered in conflict if they share at least one hit but do not form a valid quadruplet. The negative sign for compatible triplets converts high geometric consistency into attractive couplings under QUBO minimization, while the positive penalty suppresses the simultaneous selection of conflicting trajectory hypotheses. These three interaction modes are illustrated in Fig. 3.

The resulting QUBO optimization problem is solved using simulated bifurcation (SB), a quantum-inspired Ising optimization algorithm for large-scale binary optimization problems Goto et al. (2019); Okawa et al. (2024b). SB is adopted as the optimization backend due to its favorable scalability for large-scale QUBO instances, offering a practical balance between solution quality and computational efficiency on classical hardware. This scalability is particularly important for future high-occupancy particle tracking scenarios, where the number of candidate trajectories and the corresponding binary variable can grow substantially. All ML-QTrack experiments use the same SB configuration, with details on the QUBO-to-Ising transformation, coefficient normalization, dynamical evolution, and solution selection provided in Appendix D.

3.3 Trajectory-Level Refinement

The QUBO optimization stage selects a set of globally consistent triplets, which must be further assembled into complete trajectories. Unlike previous QUBO-based tracking studies that relied on

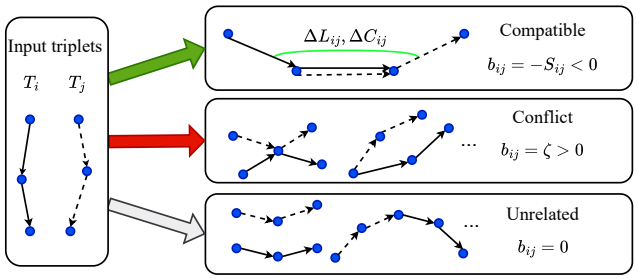


Figure 3: Illustration of pairwise interaction construction in the trajectory-level QUBO formulation. Candidate triplets are treated as binary variables, and the quadratic coefficient b_{ij} encodes their geometric relationship. Compatible triplets forming valid quadruplet continuations receive attractive couplings weighted by the compatibility score S_{ij}, while conflicting triplets are assigned repulsive penalties ζ. Triplet pairs without continuation or conflict relationships have zero coupling.

truth-based processing to remove same-layer duplicate hits before optimization Bapst et al. (2020); Zlokapa et al. (2021), ML-QTrack operates on the full detector event without such preprocessing. Consequently, the optimized solution may still contain residual trajectory-level ambiguities, including duplicate reconstructions and fragmented trajectory segments. To address these issues, we introduce a lightweight post-optimization refinement stage that assembles trajectory candidates by linking compatible triplet continuations. It further resolves duplicate trajectories from ambiguous hit assignments and fragmented trajectories from missing connections in the hypothesis graph.

Duplicate trajectories are identified through detector-layer overlap and hit correspondence consistency, while fragmented trajectories are recovered by detecting compatible segments based on directional consistency and curvature continuity. The identified relationships are subsequently merged to produce the final reconstructed tracks. This refinement operates only on the QUBO-selected trajectories and does not alter the global optimization objective, enabling ML-QTrack to operate under realistic full-event conditions without relying on truth-based ambiguity removal. Implementation details of the refinement criteria and merging procedure are provided in Appendix F.

4 Experiments

4.1 Experimental Setup and Evaluation

We evaluate ML-QTrack on the TrackML dataset Amrouche et al. (2019). All results are evaluated on held-out test events excluded from training. Candidate graphs are generated using the publicly available metric-learning model from the CommonFramework benchmark Caillou et al. (2022). The generated graphs are then refined by the proposed LightGBM-based purification stage before trajectory-level QUBO optimization. All ML-QTrack QUBO instances are solved using the same SB optimizer configuration unless otherwise specified.

We use the standard TrackML double-majority matching criterion Amrouche et al. (2019). A reconstructed trajectory is matched to a true particle if (1) more than 50% of the particle's hits are recovered by the trajectory and (2) more than 50% of the trajectory hits originate from that particle. Let $N_{\text{reconstructed}}$ denote the number of reconstructed trajectories, and let N_{matched} and $N_{\text{matched tracks}}$ denote the numbers of matched particles and matched reconstructed trajectories, respectively. The target-particle selection is specified for each evaluation setting, with $p_T \geq 1$ GeV and the minimum hit requirement determined by the corresponding evaluation protocol. Let N_{target} denote the number of target particles. Reconstruction performance is evaluated using efficiency and fake rate:

$$\text{Efficiency} = \frac{N_{\text{matched}}}{N_{\text{target}}}, \qquad \text{Fake Rate} = 1 - \frac{N_{\text{matched tracks}}}{N_{\text{reconstructed}}}. \tag{6}$$

For ML-QTrack, uncertainties denote the sample standard deviation across 100 test events.

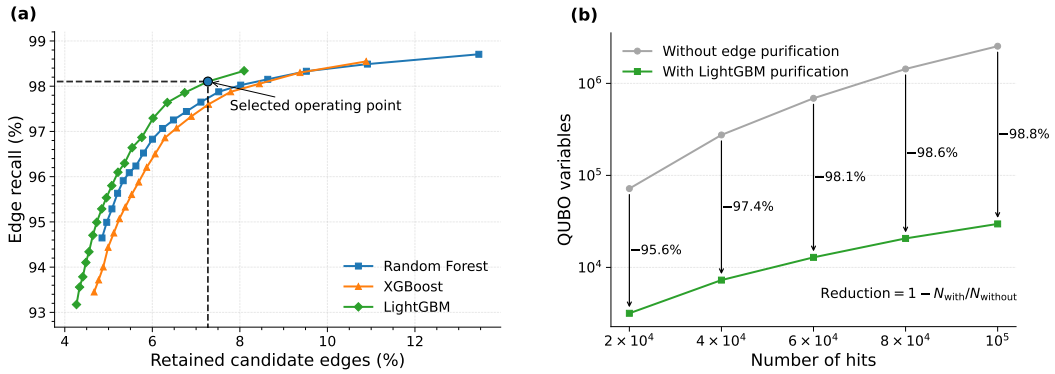


Figure 4: Effect of edge purification on hypothesis-space compression and downstream QUBO complexity. (a) Retained candidate-edge ratio versus edge recall for Random Forest, XGBoost, and LightGBM across the threshold sweep, with the selected LightGBM operating point indicated. (b) QUBO variable count versus hit number with and without LightGBM-based edge purification, showing a 95.6–98.8% reduction in QUBO variables across the tested hit counts.

4.2 Does Learning Improve the Better Hypotheses?

We investigate whether learning-guided edge purification can reduce the candidate space while preserving the high recall required for downstream reconstruction. The metric-learning stage achieves over 99% edge recall but only about 2% edge precision, resulting in a dense graph with many spurious hypotheses. Figure 4(a) shows that LightGBM provides the most favorable compression–recall trade-off. We therefore select $\tau = 0.02$, reducing the candidate-edge space from 436.3k to 26.4k edges (93.9% reduction) while retaining 98.16% of the true edges.

This compression becomes increasingly important for the higher-order QUBO formulation. As shown in Figure 4(b), at 10^5 hits, purification reduces the number of QUBO variables from approximately 2.53×10^6 to 2.97×10^4, demonstrating that edge purification prevents the combinatorial growth of the downstream optimization problem.

4.3 How Do Learning and Global Optimization Complement Each Other?

We next examine the complementary roles of hypothesis purification and global trajectory optimization. Table 1 compares the two components individually and in combination. Without purification, QUBO optimization achieves an efficiency of 93.50% but has a high fake rate of 18.75%. Applying greedy selection to the purified graph reduces the fake rate to 3.51%, while global optimization on the same purified graph further improves efficiency to 98.48% and reduces the fake rate to 1.12%.

Table 1: Ablation study of hypothesis purification and trajectory-level global optimization, showing their complementary effects on track reconstruction.

Method	Purification	Global optimization	Efficiency (%) ↑	Fake Rate (%) ↓
QUBO only	×	✓	93.50 ± 1.2	18.75 ± 2.5
Greedy selection	✓	×	95.72 ± 0.9	3.51 ± 0.4
ML-QTrack (full)	✓	✓	$\mathbf{98.44 \pm 0.2}$	$\mathbf{1.12 \pm 0.1}$

These results show that the two components address different reconstruction aspects: purification reduces the hypothesis space and suppresses spurious candidates, whereas global optimization selects globally consistent trajectory combinations. Their combination thus provides the best reconstruction performance. The details of the greedy selection procedure are provided in Appendix G.

4.4 Comparison with Existing Track Reconstruction Methods

Existing TrackML studies adopt different detector configurations and evaluation protocols, substantially affect tracking difficulty. We therefore compare ML-QTrack under three settings: (1) the duplicate-hit removal protocol used by prior QUBO-based methods, (2) the pixel-only configuration adopted by several recent learning-based methods, and (3) the full-event full-detector setting.

Setting 1: Duplicate-hit removal. For direct comparison with previous QUBO-based approaches Bapst et al. (2020); Zlokapa et al. (2021), we follow their protocol by removing same-particle same-layer duplicate hits before reconstruction. This preprocessing removes ambiguities caused by overlapping sensor modules and simplifies the QUBO formulation by making adjacent-hit relations more regular. Bapst et al. (2020) consider only the barrel detector, while Zlokapa et al. (2021) include both barrel and endcap but evaluate sectorized events with 500 particles per event rather than full TrackML events. As shown in Table 2, ML-QTrack achieves 98.4% efficiency and 1.1% fake rate under this protocol, while reconstructing a complete Barrel + Endcap event in 1.68 s. In comparison, the implementation in Bapst et al. (2020) requires more than 1 hour per complete Barrel-only event in our reproduction, consistent with their reported runtime. Thus, ML-QTrack achieves competitive reconstruction performance with broader detector coverage and second-scale event reconstruction.

Table 2: Comparison with previous QUBO-based methods under the duplicate-hit removal protocol.

Method	Detector Geometry	Efficiency (%)	Fake Rate (%)	Runtime / event
Bapst et al. (2020)	Barrel only	97.2	2.7	> 1 h
Zlokapa et al. (2021)	Barrel + Endcap	95.5	2.0	–
ML-QTrack	Barrel + Endcap	**98.4** $\pm$ **0.2**	**1.1** $\pm$ **0.1**	**1.68** $\pm$ **0.10 s**

Setting 2: Pixel-only reconstruction. We next retain same-layer duplicate hits to match the evaluation setting used by learning-based tracking methods. The trajectory-level refinement is therefore enabled to resolve these detector-induced ambiguities; details are given in Section 3.3. Recent approaches, including OC Lieret et al. (2023) and the Transformer-based method Van Stroud et al. (2025), restrict reconstruction to the innermost pixel (P) layers. This limited detector coverage reduces trajectory length and combinatorial complexity, with the pixel-only setting using only 18 of the 48 detector layers. We evaluate ML-QTrack under the same configuration. Although the triplet-based QUBO formulation requires $N_{\text{hits}}^{\min} = 4$, versus 3 for the baseline methods, this difference reduces the target trajectory population by only about 5ML-QTrack achieves 96.2to those methods (Table 3), while remaining extensible to the outer strip layers and full-detector reconstruction.

Table 3: Comparison with recent pixel-only learning-based tracking methods. All are evaluated at $|\eta| < 4$. The minimum hit requirement for ML-QTrack is $N_{\text{hits}}^{\min} = 4$, as the QUBO variables represent overlapping hit triplets, and at least two consecutive triplets form a reconstructed track.

Method	$p_T^{\min}$ (GeV)	$N_{\text{hits}}^{\min}$	Efficiency (%)	Fake Rate (%)
OC Lieret et al. (2023)	0.9	3	96.4	0.9
Transformer Van Stroud et al. (2025)	1.0	3	94.1 $\pm$ 0.1	0.7 $\pm$ 0.1
ML-QTrack	1.0	4	**96.2** $\pm$ **0.2**	**0.6** $\pm$ **0.1**

Setting 3: Full-event, full-detector reconstruction. This represents the most challenging setting, combining full event complexity with the complete detector geometry. HGNN Liu et al. (2026) provides the closest comparison, as both methods use full events and metric-learning-based candidate graphs, while adopting different trajectory reconstruction strategies. HGNN uses hierarchical graph representations for track reconstruction, whereas ML-QTrack performs trajectory-level global optimization. As shown in Table 4, ML-QTrack achieves a comparable efficiency of 98.1% while reducing the fake rate from 36.71% to 6.87%, corresponding to an $\sim$81% relative reduction. This result highlights the effectiveness of trajectory-level global optimization in suppressing fake trajectories while maintaining high reconstruction efficiency.

Overall, ML-QTrack performs competitively against both QUBO-based methods and recent learning-based approaches across different reconstruction settings, while extending from restricted

Table 4: Comparison with HGNN under the full-event, full-detector TrackML setting. Both methods use same metric-learning-based candidate graph and target particles with $p_T \geq 1$ GeV and ≥ 5 hits.

Method	Efficiency (%) ↑	Fake Rate (%) ↓
HGNN Liu et al. (2026)	97.9	36.71
ML-QTrack	**98.1** $\pm$ **0.3**	**6.87** $\pm$ **0.4**

pixel-only reconstruction to full-detector, full-event reconstruction. In the full-event comparison with HGNN, ML-QTrack achieves comparable efficiency with a substantially lower fake rate.

4.5 Runtime and Scalability

We finally evaluate the computational scalability of ML-QTrack across different event sizes. The measured runtime covers the complete reconstruction pipeline, including LightGBM purification, QUBO construction, and trajectory-level global optimization. The preceding metric-learning candidate generation requires less than 10 ms per event and contributes negligibly to the overall runtime.

As shown in Figure 5, increasing the event size from 2×10^4 to 10^5 hits increases the total runtime from 0.55 s to 1.67 s, corresponding to a fivefold increase in input size but only a roughly threefold increase in end-to-end runtime. The runtime growth is dominated by LightGBM inference, while QUBO construction remains approximately constant at 0.45–0.48 s.

Importantly, the global optimization stage remains computationally inexpensive. Its runtime increases from only 8.7 ms to 21.9 ms across the tested event sizes. This efficiency results from the combination of edge purification, which substantially reduces the QUBO problem size, and the highly parallelizable dynamics of the quantum-annealing-inspired Ballistic Simulated Bifurcation (BSB) optimizer. A detailed study of BSB-based acceleration for track reconstruction is provided in Okawa et al. (2024a). Overall, ML-QTrack maintains second-scale full-event reconstruction without making trajectory-level optimization a computational bottleneck.

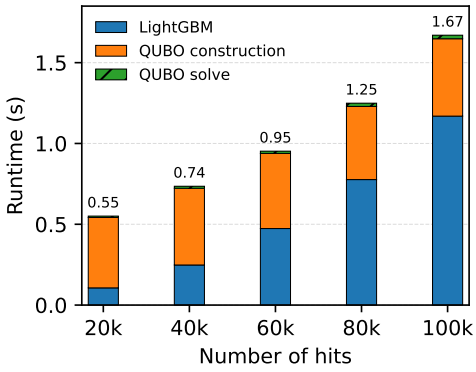


Figure 5: Runtime scaling of ML-QTrack across event sizes, showing the total runtime and contributions from LightGBM purification, QUBO construction, and QUBO solving. Measurements use a 6-core, 12-thread CPU with 32 GB RAM.

5 Conclusion and Discussion

We presented ML-QTrack, a learning-guided global optimization framework for particle track reconstruction. By combining high-recall candidate generation, LightGBM-based hypothesis purification, and trajectory-level QUBO optimization, ML-QTrack reduces the combinatorial search space while enforcing global trajectory consistency. Experiments on TrackML demonstrate that this approach achieve high reconstruction efficiency and low fake rates across a range of reconstruction settings.

More broadly, our results suggest that learned hypothesis compression can make global combinatorial optimization tractable for large-scale tracking. By reducing the hypothesis space before trajectory-level optimization, ML-QTrack enables global consistency while remaining compatible with different candidate generators and QUBO optimizers. A remaining challenge is scalability at higher track densities, where candidate hypotheses may grow rapidly, as well as the recall–complexity trade-off introduced by purification. Future work will explore higher-occupancy and more realistic detector conditions, including missing hits and detector imperfections, together with hardware-aware implementations of global optimization.

REFERENCES

I. Béjar Alonso, O. Brüning, P. Fessia, M. Lamont, L. Rossi, L. Tavian, and M. Zerlauth (eds.). *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report*, volume 10 of *CERN Yellow Reports: Monographs*. CERN, 2020. doi: 10.23731/CYRM-2020-0010. URL https://e-publishing.cern.ch/index.php/CYRM/issue/view/127. ISBN 978-92-9083-586-8 (paperback), 978-92-9083-587-5 (PDF).

Sabrina Amrouche, Laurent Basara, Paolo Calafiura, Victor Estrade, Steven Farrell, Diogo R. Ferreira, Liam Finnie, Nicole Finnie, Cécile Germain, Vladimir Vava Gligorov, Tobias Golling, Sergey Gorbunov, Heather Gray, Isabelle Guyon, Mikhail Hushchyn, Vincenzo Innocente, Moritz Kiehn, Edward Moyse, Jean-François Puget, Yuval Reina, David Rousseau, Andreas Salzburger, Andrey Ustyuzhanin, Jean-Roch Vlimant, Johan Sokrates Wind, Trian Xylouris, and Yetkin Yilmaz. *The Tracking Machine Learning Challenge: Accuracy Phase*, pp. 231–264. Springer International Publishing, November 2019. ISBN 9783030291358. doi: 10.1007/978-3-030-29135-8_9. URL http://dx.doi.org/10.1007/978-3-030-29135-8_9.

Frédéric Bapst, Wahid Bhimji, Paolo Calafiura, Heather Gray, Wim Lavrijsen, Lucy Linder, and Alex Smith. A pattern recognition algorithm for quantum annealers. *Computing and Software for Big Science*, 4(1):1, 2020. doi: 0.1007/s41781-019-0032-5. URL https://link.springer.com/article/10.1007/s41781-019-0032-5.

Sylvain Caillou, Paolo Calafiura, Steven Andrew Farrell, Xiangyang Ju, Daniel Thomas Murnane, Charline Rougier, Jan Stark, and Alexis Vallier. ATLAS ITk Track Reconstruction with a GNN-based pipeline. In *Connecting The Dots*, Princeton, United States, May 2022. URL https://hal.archives-ouvertes.fr/hal-03793565.

G Cerati, P Elmer, S Krutelyov, S Lantz, M Lefebvre, M Masciovecchio, K McDermott, D Riley, M Tadel, P Wittich, F Wurthwein, and A Yagil. Parallelized kalman-filter-based reconstruction of particle tracks on many-core architectures. *Journal of Physics: Conference Series*, 1085:042016, sep 2018. doi: 10.1088/1742-6596/1085/4/042016. URL https://doi.org/10.1088%2F1742-6596%2F1085%2F4%2F042016.

ATLAS Collaboration. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1–29, September 2012. doi: https://doi.org/10.1016/j.physletb.2012.08.020. URL https://www.sciencedirect.com/science/article/pii/S037026931200857X.

ATLAS Collaboration. ATLAS HL-LHC Computing Conceptual Design Report. Technical report, CERN, 2020. URL https://cds.cern.ch/record/2729668.

T Cornelissen, M Elsing, I Gavrilenko, W Liebig, E Moyse, and A Salzburger. The new atlas track reconstruction (newt). *Journal of Physics: Conference Series*, 119(3):032014, jul 2008. doi: 10.1088/1742-6596/119/3/032014. URL https://doi.org/10.1088/1742-6596/119/3/032014.

S. Cucciarelli, M. Konecki, D. Kotlinski, and T. Todorov. Track reconstruction, primary vertex finding and seed generation with the pixel detector. 1 2006.

Lyndon Evans and Philip Bryant. LHC Machine. *JINST*, 3(0):S08001, 2008. doi: 10.1088/1748-0221/3/08/S08001.

Steven Farrell et al. Novel deep learning methods for track reconstruction. In *4th International Workshop Connecting The Dots 2018 (CTD2018) Seattle, Washington, USA, March 20-22, 2018*, 2018. URL http://lss.fnal.gov/archive/2018/conf/fermilab-conf-18-598-cd.pdf.

The HEP Software Foundation, J. Albrecht, A. A. Alves, et al. A roadmap for hep software and computing r&d for the 2020s. *Computing and Software for Big Science*, 3(1):7, 2019. doi: 10.1007/s41781-018-0018-8. URL https://doi.org/10.1007/s41781-018-0018-8.

Daniel Funke, Thomas Hauth, Vincenzo Innocente, Günter Quast, P Sanders, and D Schieferdecker. Parallel track reconstruction in cms using the cellular automaton approach. In *Journal of Physics: Conference Series*, volume 513, pp. 052010. IOP Publishing, 2014. doi: 10.1088/1742-6596/513/5/052010. URL https://iopscience.iop.org/article/10.1088/1742-6596/513/5/052010.

Hayato Goto, Kosuke Tatsumura, and Alexander R. Dixon. Combinatorial optimization by simulating adiabatic bifurcations in nonlinear hamiltonian systems. *Science Advances*, 5(4):eaav2372, 2019. doi: 10.1126/sciadv.aav2372. URL https://www.science.org/doi/abs/10.1126/sciadv.aav2372.

Xiangyang Ju, Steven Farrell, Paolo Calafiura, Daniel Murnane, Prabhat, Lindsey Gray, Thomas Klijnsma, Kevin Pedro, Giuseppe Cerati, Jim Kowalkowski, Gabriel Perdue, Panagiotis Spentzouris, Nhan Tran, Jean-Roch Vlimant, Alexander Zlokapa, Joosep Pata, Maria Spiropulu, Sitong An, Adam Aurisano, V Hewes, Aristeidis Tsaris, Kazuhiro Terao, and Tracy Usher. Graph neural networks for particle reconstruction in high energy physics detectors, 2020. URL https://arxiv.org/abs/2003.11603.

Kilian Lieret, Gage DeZoort, Devdoot Chatterjee, Jian Park, Siqi Miao, and Pan Li. High pileup particle tracking with object condensation, 2023. URL https://arxiv.org/abs/2312.03823.

Ryan Liu, Paolo Calafiura, Steven Farrell, Xiangyang Ju, Daniel Thomas Murnane, and Tuan Minh Pham. Hierarchical graph neural networks for particle track reconstruction, 2023. URL https://arxiv.org/abs/2303.01640.

Ryan Liu, Paolo Calafiura, Steven Farrell, Xiangyang Ju, Daniel Thomas Murnane, and Tuan Minh Pham. Hierarchical graph neural networks for particle track reconstruction. *Journal of Physics: Conference Series*, 3206(1):012084, apr 2026. doi: 10.1088/1742-6596/3206/1/012084. URL https://doi.org/10.1088/1742-6596/3206/1/012084.

Hideki Okawa, Qing-Guo Zeng, Xian-Zhe Tao, and Man-Hong Yung. Quantum-annealing-inspired algorithms for track reconstruction at high-energy colliders. *Computing and Software for Big Science*, 8(1):16, 2024a. ISSN 2510-2044. doi: 10.1007/s41781-024-00126-z. URL https://doi.org/10.1007/s41781-024-00126-z.

Hideki Okawa, Qing-Guo Zeng, Xian-Zhe Tao, and Man-Hong Yung. Quantum-annealing-inspired algorithms for track reconstruction at high-energy colliders. *Computing and Software for Big Science*, 8(1):16, 2024b. URL https://link.springer.com/article/10.1007/s41781-024-00126-z.

Hideki Okawa, Xian-Zhe Tao, Qing-Guo Zeng, and Man-Hong Yung. Quantum-annealing-inspired algorithms for multijet clustering. *Physics Letters B*, 864:139393, 2025. URL https://www.sciencedirect.com/science/article/pii/S0370269325001534.

Davide Pastorello and Enrico Blanzieri. Quantum annealing learning search for solving qubo problems. *Quantum Information Processing*, 18(10):303, 2019. URL https://link.springer.com/article/10.1007/s11128-019-2418-z.

CMS Offline Software and Computing. CMS Phase-2 Computing Model: Update Document. Technical report, CERN, Geneva, 2022. URL https://cds.cern.ch/record/2815292.

Samuel Van Stroud, Nikita Pond, Max Hart, Jackson Barr, Sébastien Rettie, Gabriel Facini, and Timothy Scanlon. Secondary vertex reconstruction with MaskFormers. *Eur. Phys. J. C*, 84(10):1020, 2024. doi: 10.1140/epjc/s10052-024-13374-5.

Samuel Van Stroud, Philippa Duckett, Max Hart, Nikita Pond, Sébastien Rettie, Gabriel Facini, and Tim Scanlon. Transformers for charged particle track reconstruction in high-energy physics. *Phys. Rev. X*, 15:041046, Dec 2025. doi: 10.1103/md46-yqgd. URL https://link.aps.org/doi/10.1103/md46-yqgd.

Alexander Zlokapa, Abhishek Anand, Jean-Roch Vlimant, Javier M. Duarte, Joshua Job, Daniel Lidar, and Maria Spiropulu. Charged particle tracking with quantum annealing optimization. *Quantum Machine Intelligence*, 3(2), November 2021. ISSN 2524-4914. doi: 10.1007/s42484-021-00054-w. URL http://dx.doi.org/10.1007/s42484-021-00054-w.

A LightGBM Edge Classification

The LightGBM classifier is trained to distinguish true particle connections from spurious candidate edges generated by the metric-learning stage. The training samples are constructed directly from the candidate edges of the training events and the corresponding hit-level truth information.

A.1 Training Sample Construction

For each candidate edge connecting two hits, we retrieve the spatial coordinates and detector-volume identifiers of the source and target hits from the corresponding truth table. Candidate edges are divided into two categories according to their detector volumes. Edges whose source and target hits belong to the same detector volume are assigned to the same-volume classifier, while edges connecting different volumes are assigned to the cross-volume classifier. Separate classifiers are used because the geometric characteristics of within-volume and cross-volume connections differ.

The binary training label is defined from the edge truth label provided by the candidate graph. An edge is treated as a positive example only when it corresponds to a true particle connection and the associated particle has transverse momentum

$$p_T = \sqrt{tpx^2 + tpy^2} > 1 \text{ GeV}.$$

All remaining candidate edges are treated as negative examples. This labeling procedure aligns the purification task with the target-particle definition used in the tracking evaluation.

A.2 Geometric Features

For each candidate edge, we construct geometric features from the coordinates of its two endpoint hits. Let the source and target hits be denoted by (x_1, y_1, z_1) and (x_2, y_2, z_2), respectively. The Cartesian displacements are defined as

$$dx = x_2 - x_1, \qquad dy = y_2 - y_1, \qquad dz = z_2 - z_1.$$

The radial coordinates are

$$r_1 = \sqrt{x_1^2 + y_1^2}, \qquad r_2 = \sqrt{x_2^2 + y_2^2},$$

from which the relative radial growth is computed as

$$\text{length_ratio} = \frac{r_2}{r_1}.$$

The azimuthal and polar-angle differences are calculated as

$$d\phi = \phi_2 - \phi_1, \qquad d\theta = \theta_2 - \theta_1,$$

with $d\phi$ wrapped to the interval $[-\pi, \pi]$. We additionally use the longitudinal slope

$$drz = \frac{z_2 - z_1}{r_2 - r_1},$$

which provides a direct geometric description of the hit displacement in the longitudinal direction.

To expose both signed and magnitude-based geometric information, the absolute values of the signed displacement and angular features are included as additional inputs. For cross-volume edges, the source and target volume identifiers and an encoded volume-pair feature are also provided to the classifier.

The complete feature set is summarized in Table 5.

Table 5: Geometric features used for LightGBM edge classification.

Feature	Description
dx, dy, dz	Cartesian coordinate differences between two hits
$d\phi$	Wrapped azimuthal angle difference
$d\theta$	Polar-angle difference
drz	Longitudinal slope, $(z_2 - z_1)/(r_2 - r_1)$
r_1, r_2	Radial coordinates of the two hits
r_2/r_1	Relative radial growth ratio
$\|dx\|, \|dy\|, \|dz\|$	Absolute Cartesian displacements
$\|drz\|$	Absolute longitudinal slope
$\|d\phi\|, \|d\theta\|$	Absolute angular differences
$V_{\mathrm{src}}, V_{\mathrm{dst}}$	Source and target volume IDs
V_{pair}	Encoded source–target volume pair

A.3 Classifier Training

The same LightGBM configuration is used for both the same-volume and cross-volume classifiers. Each classifier consists of 300 boosting iterations with a learning rate of 0.05 and 64 leaves per tree. The maximum tree depth is left unconstrained, and all available CPU threads are used for training. The complete configuration is listed in Table 6.

Table 6: LightGBM configuration used for edge purification.

Parameter	Value
Number of estimators	300
Learning rate	0.05
Number of leaves	64
Maximum depth	Unlimited
Parallel threads	All available CPU threads
Validation fraction	20%
Random seed	42

For each classifier, the available edge samples are randomly divided into training and validation subsets using an 80/20 split with a fixed random seed. The validation set is used to evaluate the classifier through the area under the receiver operating characteristic curve (AUC) and to study the operating threshold for edge retention.

A.4 Edge Retention Threshold

After training, the classifier assigns each candidate edge a probability score corresponding to the predicted positive class. An edge is retained when its predicted score exceeds a threshold τ. The threshold determines the trade-off between retaining true particle connections and suppressing spurious candidates.

For a given threshold, we evaluate the edge-level recall, precision, and retained-edge ratio as

$$R_{\mathrm{edge}} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{positive}}}, \qquad P_{\mathrm{edge}} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{retained}}}, \qquad \rho_{\mathrm{keep}} = \frac{N_{\mathrm{retained}}}{N_{\mathrm{candidate}}}.$$

The threshold is selected to retain a high fraction of true edges while substantially reducing the size of the candidate graph. Unless otherwise specified, the main experiments use $\tau = 0.02$.

The resulting purified graph is then passed to the trajectory-level QUBO construction described in the main text. In this way, the LightGBM stage acts as a learned hypothesis-space reduction step rather than performing the final trajectory assignment.

A.5 GRAPH PURIFICATION

Figure 6 illustrates the graph purification process used in ML-QTrack. The metric-learning model first generates a high-recall candidate graph containing both true and spurious edge hypotheses. The LightGBM classifier then filters these candidates to produce a compact graph while retaining the trajectory structure needed for downstream reconstruction. The resulting graph is used as the input to the trajectory-level QUBO optimization.

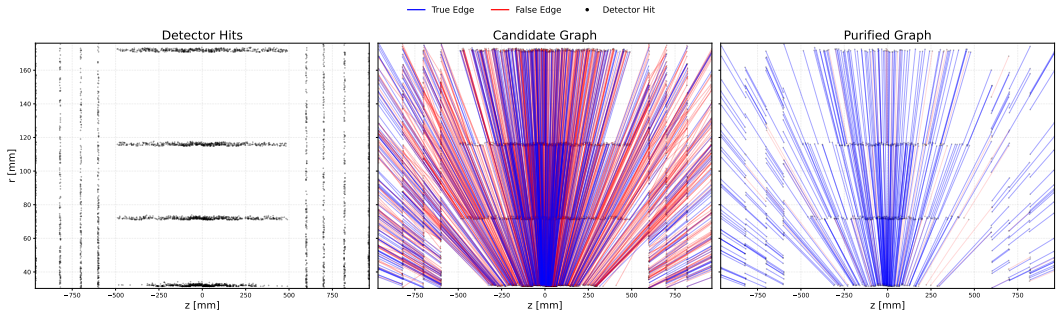


Figure 6: Visualization of the proposed learning-guided hypothesis generation. From left to right, the panels show detector hits, the candidate graph produced by metric learning, and the purified graph after learning-guided edge filtering. For visualization purposes, only a subset of detector layers and edge hypotheses are displayed. True edges are shown in blue, false edges in red, and detector hits in black. The proposed learning stage removes the majority of spurious edge hypotheses while preserving the underlying trajectory structure.

B CHARACTERIZATION OF LOCAL GEOMETRIC CONSISTENCY

To characterize the local geometric structure of true trajectories, we analyze two quantities used in the trajectory-level QUBO: the absolute transverse curvature $|\delta(q/p)|$ and the longitudinal mismatch ΔL^{int}. The distributions are evaluated for truth-level triplets formed by three consecutive hits belonging to the same particle, restricted to particles with $p_T > 1$ GeV. The resulting distributions are shown in Fig. 7.

The distribution of $|\delta(q/p)|$ is relatively concentrated, with a median of 4.08×10^{-4}, while the 95th and 99th percentiles reach 1.25×10^{-3} and 2.01×10^{-3}, respectively. This indicates that most true triplets exhibit similar local transverse curvature, while a small fraction shows substantially larger curvature.

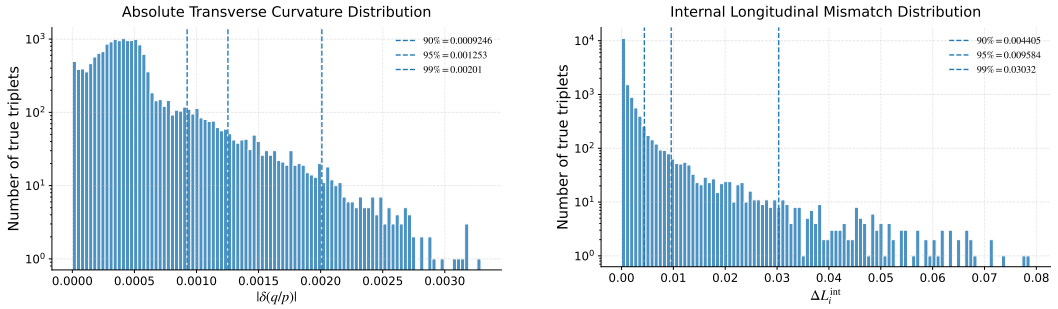
The longitudinal mismatch ΔL^{int} exhibits a more pronounced right-skewed distribution. Its median is 3.00×10^{-4}, compared with 9.58×10^{-3} and 3.03×10^{-2} at the 95th and 99th percentiles, respectively. Thus, although most true triplets have small longitudinal deviations, a non-negligible tail extends to larger values.

These empirical distributions motivate the use of continuous geometric penalties in the QUBO objective. Rather than imposing hard thresholds on individual triplets, the formulation assigns larger penalties to geometrically less consistent hypotheses while retaining the possibility of valid trajectories with larger local deviations.

C DETAILED HYPOTHESIS GENERATION PROCEDURE

The trajectory-level QUBO operates on triplet hypotheses generated from the purified candidate graph. This section summarizes the implementation details of the hypothesis-generation procedure used in ML-QTrack.

Triplet construction. Given a directed candidate graph, each doublet represents a directed connection between two detector hits. Triplets are constructed by joining an incoming doublet and



(a) Absolute transverse curvature

(b) Longitudinal mismatch

Figure 7: Distributions of local geometric quantities for truth-level triplets with $p_T > 1$ GeV. (a) Absolute transverse curvature $|\delta(q/p)|$ and (b) longitudinal mismatch ΔL^{int}. The dashed lines indicate the 90th, 95th, and 99th percentiles of the corresponding distributions. The results characterize the local geometric variations of true trajectories and motivate the continuous geometric penalties used in the trajectory-level QUBO.

an outgoing doublet that share a common hit. Specifically, for two doublets $d_1 = (h_0, h_1)$ and $d_2 = (h_1, h_2)$, we construct the triplet

$$T = (h_0, h_1, h_2). \tag{7}$$

For each triplet, the indices of its two constituent doublets and the three associated hits are retained, allowing subsequent trajectory continuations to be identified through shared doublets.

To efficiently enumerate triplets, the incoming and outgoing doublets of each hit are stored using compressed sparse row (CSR) representations. For a hit h, all combinations of its incoming and outgoing doublets are enumerated locally. The number of candidate triplets generated at a hit is therefore

$$N_T(h) = N_{\text{in}}(h) N_{\text{out}}(h). \tag{8}$$

The complete triplet set is obtained by processing all detector hits.

Triplet geometric quantities and weights. Each generated triplet is assigned a longitudinal mismatch and a transverse curvature. The longitudinal mismatch is computed from the difference between the r–z angular quantities of its two constituent doublets,

$$\Delta L_i^{\text{int}} = \left| \theta_{L,i}^{(2)} - \theta_{L,i}^{(1)} \right|. \tag{9}$$

The transverse curvature of the three hits is evaluated using the Menger curvature. For consistency with the notation used in the main text, we define $\kappa_i \equiv \delta(q/p) T_i$, where κ_i denotes the transverse-curvature quantity associated with triplet T_i. It is given by

$$\kappa_i = \frac{4A_i}{\ell_{01}\ell_{12}\ell_{02}}, \tag{10}$$

where A_i is the area of the triangle formed by the three hits, and ℓ_{01}, ℓ_{12}, and ℓ_{02} are the corresponding pairwise distances. The sign of κ_i is determined by the orientation of the three hits in the (x)–(y) plane.

The longitudinal mismatch is converted into the unary triplet weight used in the QUBO as

$$a_i = \frac{\Delta L_i^{\text{int}}}{\Delta L_i^{\text{int}} + \tau_a}, \tag{11}$$

where $\tau_a = 0.02$ is a fixed regularization parameter. Thus, triplets with smaller longitudinal mismatch receive smaller unary penalties.

Quadruplet construction and compatibility. Trajectory continuations are generated by joining two triplets that share the intermediate doublet. For

$$T_i = (h_0, h_1, h_2), \qquad T_j = (h_1, h_2, h_3), \tag{12}$$

the pair forms a four-hit trajectory hypothesis. Candidate triplet pairs are identified using a CSR mapping from doublets to triplets.

For each candidate pair, the absolute transverse-curvature difference and the maximum longitudinal mismatch of the two triplets are evaluated. These quantities are combined using the compatibility score defined in Eq. 4. Only pairs satisfying the predefined compatibility threshold are retained as valid quadruplet hypotheses.

In the implementation, the quadruplet strength is evaluated using $\delta_C^{\max} = 10^{-4}$ and $\delta_L^{\max} = 0.1$. The relative contribution of transverse curvature and longitudinal consistency is controlled by the parameter α defined in Eq. 4, with the default value $\alpha = 0.5$.

Hypothesis-space reduction. After valid quadruplets have been constructed, triplets that do not participate in any valid quadruplet are discarded. The remaining triplets are then re-indexed before QUBO construction. This reduction removes hypotheses that cannot participate in a trajectory continuation and hence do not contribute to the trajectory-level optimization. The mapping between the original triplet indices and the compact QUBO indices is retained throughout the reconstruction pipeline, allowing the binary solution returned by the optimizer to be mapped back to the corresponding physical triplet hypotheses during trajectory decoding.

Implementation and sparse data structures. The implementation uses CSR representations for the doublet-to-triplet and hit-to-triplet mappings, providing efficient access to local neighborhoods during quadruplet construction, conflict generation, and trajectory decoding without constructing dense adjacency matrices. Triplet and quadruplet arrays are allocated using a two-pass construction strategy: the first pass determines the required storage size, while the second pass fills the preallocated arrays. Geometric quantities such as the transverse curvature are evaluated using JIT-compiled numerical kernels. Together, these choices reduce dynamic memory allocation and enable the same hypothesis-generation procedure to be applied efficiently across different event occupancies.

D Simulated Bifurcation Solver Configuration

All QUBO instances in this work are solved using a *simulated bifurcation* (SB)–based solver, a quantum-inspired classical heuristic designed to approximate low-energy solutions of Ising and QUBO Hamiltonians through continuous-time dynamical evolution of coupled nonlinear oscillators.

QUBO to Ising mapping. Given a QUBO objective of the form

$$\mathcal{H}_{\mathrm{QUBO}} = \sum_i a_i x_i + \sum_{i<j} b_{ij} x_i x_j, \qquad x_i \in \{0, 1\}, \tag{13}$$

the problem is converted into an equivalent Ising formulation

$$\mathcal{H}_{\mathrm{Ising}} = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j, \qquad s_i \in \{-1, +1\}, \tag{14}$$

using the standard variable transformation $x_i = (s_i + 1)/2$. Linear fields h_i and couplings J_{ij} are constructed accordingly from the QUBO bias vector and quadratic coefficient matrix. To improve numerical stability, all coefficients are rescaled by a global normalization factor equal to the maximum absolute value of $\{J_{ij}, h_i\}$.

Solver implementation. We employ a deterministic, batch-based bSB solver, which evolves a set of continuous dynamical variables corresponding to Ising spins under a discretized time update. For each QUBO instance, the solver is configured with the following fixed hyperparameters:

- Number of iterations: $N_{\mathrm{iter}} = 300$

- Integration step size: $\Delta t = 0.3$
- Batch size (parallel trajectories): $N_{\text{batch}} = 10$

All solver parameters are kept identical across events and occupancy settings to ensure a fair comparison between pipelines.

Solution selection. Each batch trajectory yields a candidate spin configuration at the final iteration. The corresponding binary solution is recovered via $x_i = (s_i + 1)/2$, and the original QUBO energy is evaluated for each candidate. The solution with the minimum QUBO energy across the batch is selected as the final optimizer output and passed to the post-processing stage for track reconstruction.

Remarks. The simulated bifurcation solver is used here as a practical, CPU-friendly substitute for simulated annealing employed in earlier QUBO-based tracking studies. Importantly, the underlying QUBO formulation and constraint structure remain unchanged; the solver choice affects only the optimization procedure, not the definition of the objective.

E Sensitivity to the Geometric Compatibility Weight α

The compatibility score in Eq. 4 balances longitudinal and transverse-curvature consistency through the parameter α. To assess the sensitivity to this weighting, we vary α while keeping all other reconstruction settings fixed. The results are shown in Fig. 8.

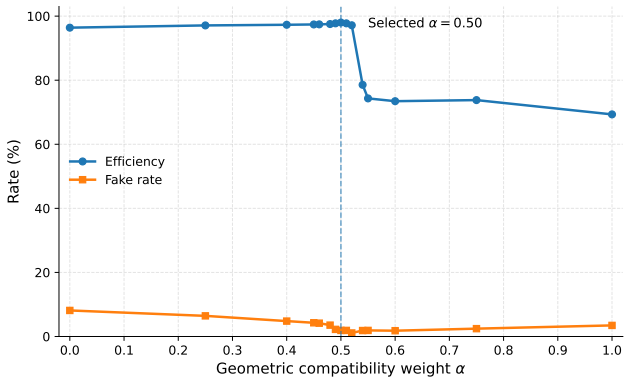


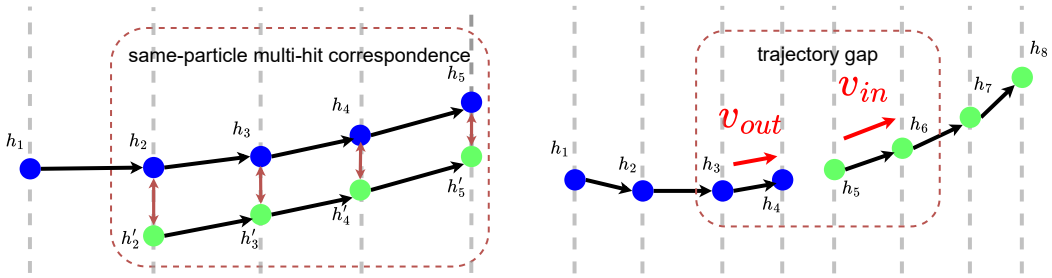
Figure 8: Reconstruction efficiency and fake rate as a function of the geometric compatibility weight α. The dashed vertical line indicates the value $\alpha = 0.50$ used in the main experiments.

The two geometric components exhibit a clear trade-off as α varies. Increasing α from 0 to 0.50 improves both reconstruction efficiency and fake-rate suppression, reaching 98.0% efficiency and a 1.9% fake rate at $\alpha = 0.50$. The efficiency remains above 97% up to $\alpha = 0.52$, while a further increase to $\alpha = 0.54$ causes a sharp drop to 78.6%. This indicates that overly emphasizing longitudinal consistency can exclude valid trajectory hypotheses. Based on this sensitivity study, we use $\alpha = 0.50$ in the main experiments.

F Detailed Trajectory Recovery Procedure

After QUBO optimization, the selected triplets are converted into preliminary trajectory candidates by connecting triplets with compatible continuations. Since ML-QTrack preserves all detector measurements and does not remove ambiguous hits using ground-truth information before optimization, the resulting trajectories may still contain trajectory-level ambiguities, including duplicate trajectories and fragmented segments.

The trajectory recovery stage resolves these ambiguities through two trajectory-level operations: duplicate-pair identification and fragment-pair identification. Candidate trajectory pairs are first preselected using coarse kinematic descriptors, after which geometric consistency tests are applied to identify duplicate or fragmented reconstructions.



(a) Duplicate-pair identification.

(b) Fragment-pair identification.

Figure 9: Trajectory-level recovery operations in ML-QTrack. (a) Duplicate-pair identification based on consistent same-particle multi-hit correspondence in overlapping detector layers. (b) Fragment-pair identification based on geometric continuity between disconnected trajectory segments.

Preliminary trajectory assembly. The selected triplets from the QUBO solution are first assembled into trajectory candidates by recursively linking triplets that form geometrically consistent quadruplet continuations. This step converts the optimized triplet selection into a set of preliminary reconstructed trajectories, which serve as the input to the subsequent recovery operations.

Trajectory-pair preselection. To reduce the number of trajectory pairs requiring detailed geometric comparison, candidate pairs are first preselected using their global pseudorapidity and azimuthal direction. For two trajectories i and j, a pair is retained when

$$|\eta_i - \eta_j| < 0.025, \qquad \Delta\phi_{ij} < 0.4, \tag{15}$$

where $\Delta\phi_{ij}$ denotes the periodic azimuthal separation. These loose requirements are used only for candidate-pair generation and are followed by trajectory-level geometric consistency tests.

Duplicate-pair identification. Duplicate trajectories are identified among candidate pairs with overlapping detector-layer coverage. For each shared detector layer, the spatial separation between the corresponding hits is evaluated. Pairs with a maximum layer-wise separation exceeding 20 mm are rejected.

For the remaining pairs, local consistency is evaluated by comparing the original and alternative hit assignments within the common detector-layer range. The mean curvature variation, maximum spatial separation, and local angular variation are used to assess whether the two trajectories can be explained by same-particle multi-hit ambiguity. In particular, the maximum spatial separation is required to remain below 20 mm, the mean Menger-curvature variation below 0.01, and the maximum local angular variation below 0.001 rad. Pairs satisfying all consistency requirements are identified as duplicate reconstructions of the same underlying trajectory.

Fragment-pair identification. Fragment recovery targets candidate trajectory pairs with no shared detector layers that may nevertheless represent disconnected segments of the same physical trajectory. The trajectory with the smaller innermost transverse radius is treated as the inner segment, while the other is treated as the outer segment.

Let $\mathbf{v}_{\text{in}}$ denote the outgoing direction of the inner trajectory segment and $\mathbf{v}_{\text{out}}$ denote the incoming direction of the outer segment. The directions are normalized before comparison. A candidate pair is accepted only when

$$\mathbf{v}_{\text{in}} \cdot \mathbf{v}_{\text{out}} > \tau_{\text{dir}}, \tag{16}$$

where the directional consistency threshold is

$$\tau_{\text{dir}} = 0.99. \tag{17}$$

In addition to directional agreement, curvature continuity is required at the potential connection point. Let κ_1 and κ_2 denote the Menger curvatures evaluated from the three consecutive points

spanning the candidate connection. The pair is accepted only when

$$|\kappa_1 - \kappa_2| < \tau_\kappa, \tag{18}$$

with

$$\tau_\kappa = 0.001. \tag{19}$$

Trajectory pairs satisfying both directional and curvature consistency requirements are treated as fragmented segments of the same particle trajectory.

Trajectory merging. The identified duplicate and fragment relationships are represented as trajectory-level connections and consolidated using a union–find procedure. Trajectories belonging to the same connected group are merged into a single reconstructed track.

For duplicate trajectories, merging removes redundant reconstructions caused by same-particle multi-hit ambiguity. For fragmented trajectories, merging reconnects disconnected segments across missing detector-layer connections. When trajectories are merged, duplicate hits are removed and the remaining hits are ordered by increasing transverse radius and longitudinal position. This recovery stage improves trajectory completeness while retaining the full-event reconstruction setting and does not use ground-truth particle identities.

G GREEDY SELECTION

Algorithm 1 Greedy Track Selection

Require: Triplets $\mathcal{T}$, quadruplets $\mathcal{Q}$
Ensure: Tracks $\mathcal{R}$
1: Sort $\mathcal{T}$ by triplet quality score
2: $\mathcal{R} \leftarrow \emptyset$
3: **for** each unremoved triplet $t \in \mathcal{T}$ **do**
4: $\quad r \leftarrow \textsc{Grow}(t, \mathcal{Q})$
5: $\quad \mathcal{R} \leftarrow \mathcal{R} \cup r$
6: $\quad$ Remove all triplets sharing hits with r
7: **end for**
8: Decode triplet chains in $\mathcal{R}$ into hit sequences
9: **return** $\mathcal{R}$

GROW extends the seed in both forward and backward directions. At each step, it selects the compatible available triplet with the smallest quadruplet strength, using triplet quality as a tie-breaker.

H PERFORMANCE VERSUS TRANSVERSE MOMENTUM

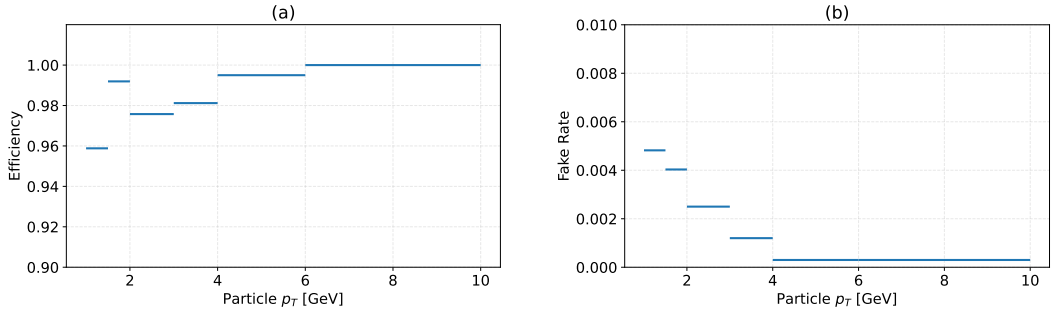


Figure 10: Reconstruction performance as a function of particle transverse momentum p_T. (a) Reconstruction efficiency and (b) fake rate across the tested p_T bins. The evaluation covers target particles with $p_T \geq 1$ GeV and $N_{\text{hits}}^{\text{min}} = 5$.

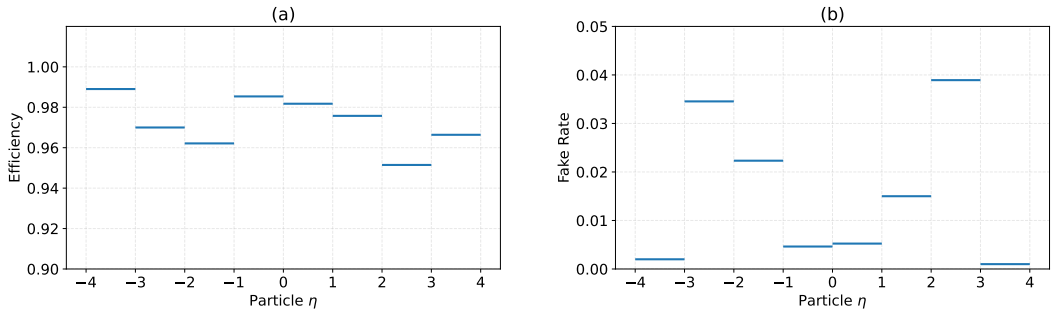


Figure 11: Reconstruction performance as a function of particle pseudorapidity η. (a) Reconstruction efficiency and (b) fake rate across the tested range $-4 < \eta < 4$.

To further characterize the reconstruction performance, we evaluate the efficiency and fake rate as a function of the transverse momentum p_T of the target particles. The results are shown in Fig. 10. The evaluation covers the range $p_T \geq 1$ GeV used in the main experiments.

The reconstruction efficiency remains above 95% across all tested momentum bins and increases to nearly 100% at higher p_T. The fake rate remains below 0.5% throughout the full range and decreases to approximately 0.03% for $p_T > 4$ GeV. The slightly lower efficiency in the lowest-p_T bin is consistent with the increased transverse curvature of low-momentum trajectories, while the overall results indicate robust reconstruction across the tested momentum range.

I Performance versus Pseudorapidity

We further evaluate the reconstruction performance as a function of particle pseudorapidity η to assess its dependence on detector geometry. The results are shown in Fig. 11, covering the full range $-4 < \eta < 4$ considered in the experimental study.

The reconstruction efficiency remains above 95% across all pseudorapidity bins, ranging from 95.1% to 98.9%. The fake rate exhibits larger variations across different η regions, reaching at most 3.9%, while remaining below 2% in five of the eight bins. Despite these variations, no pronounced efficiency degradation is observed toward the forward regions, indicating that the trajectory-level formulation remains effective across the heterogeneous detector geometry.

J Additional Statement

J.1 The Use of Large Language Model

Large language models were used to polish and proofread the manuscript for grammar and readability. All research design, mathematical derivations, experiments, and scientific claims were designed, executed, and verified independently by the authors. The authors take full responsibility for all content.