

Finding unexpected nonhelical tracks

Levi Condren¹ and Daniel Whiteson¹

Department of Physics and Astronomy, University of California, Irvine, California, USA



(Received 30 September 2025; accepted 25 February 2026; published 29 May 2026)

Many theories of physics beyond the Standard Model predict particles with nonhelical trajectories in a uniform magnetic field, but standard tracking algorithms assume helical paths and so are incapable of discovering nonhelical tracks. While alternative algorithms have been developed for specific trajectories, unforeseen physics could lead to unanticipated behavior, and such unexpected tracks are largely invisible to current algorithms, despite being potentially striking to the naked eye. A model-agnostic tracking algorithm is presented, capable of reconstructing a broad class of smooth nonhelical tracks without requiring explicit specification of particle trajectories, instead defining the target trajectories implicitly in the training sample. The network exhibits strong performance, even outside of the trajectories defined by the training sample. This proof-of-principle study takes the first step towards searches for unexpected tracks which may await discovery in current data.

DOI: [10.1103/k9n2-d85w](https://doi.org/10.1103/k9n2-d85w)

I. INTRODUCTION

Particle colliders such as the Large Hadron Collider (LHC) have tremendous potential to discover new physics [1–8], but require computationally expensive analysis to take full advantage of their data. One of the most challenging tasks is to identify a set of detector responses due to an individual charged particle, a *track*, among the enormous number left by all particles produced in a collision. Considering every possible subset is computationally intractable task due to the vast combinatorics in the most general case [9]. Traditionally, simplifying assumptions are employed to make the task feasible [10]. Primary among these is restricting the search to tracks which follow the helical trajectory expected of electrically-charged Standard Model particles in a uniform magnetic field. This greatly simplifies the track identification task, but leaves experiments incapable of finding nonhelical tracks, shrinking the potential for discovery.

Several theories of physics beyond the Standard Model (BSM) have interactions which produce nonhelical trajectories, such as magnetic monopoles [11,12] or quirks [13–15]. Recent work on quirks [16] has demonstrated the capability to efficiently find nonhelical tracks when the predicted trajectories are specified in advance. However, the space of possible particle trajectories is much larger than what is described by the current set of BSM theories.

Model-agnostic tracking, capable of finding tracks without a pre-existing theory or prespecified trajectory, would broaden experimental capacity for unexpected discoveries, especially in low-background, high-fidelity environments such as tracking detectors [17]. Tracks with striking, surprising trajectories may exist in current experimental datasets, undiscovered only because they are invisible to current algorithms.

Most traditional track-finding algorithms [18] perform alternating track parameter fitting and iterative hit finding, which requires knowledge of the explicit parametric form of the track, and so are fundamentally unsuited to model-agnostic tracking. However, recent developments in machine learning such as graph neural networks (GNNs) [19] allow for track-finding algorithms where the form of the track is never specified explicitly, only implicitly via the sample used to train the algorithm [20,21]. While not the original goal of machine learning (ML) tracking, this new approach separates the finding and fitting steps [22,23], and therefore allows for finding nonhelical trajectories when trained on such examples [16]. Crucially, it may also open the door to model-agnostic tracking, if the broad space of possible trajectories can be described implicitly via a training sample, and the tracker learns to generalize beyond the specific training examples.

In this paper, a GNN-based physics-model-agnostic track finder is trained on a sample of smooth tracks generated without a specific trajectory model via selection of Fourier amplitudes in frequency space. Smoothness is guaranteed by imposing a Schwartz condition on the relationship among Fourier amplitudes. The performance of the track finder to reconstruct nonhelical smooth tracks which are similar to those in its training sample is measured, as well as its ability

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

to generalize to tracks dissimilar to those in its training sample.

In Sec. II, the simulation of tracks is described, including the detector geometry and model-agnostic track generation. In Sec. III, the tracking pipeline is described. Section III details the reconstruction performance on various sets of tracks, including background rejection and tests of generalization. Section IV discusses the results and next steps.

II. MODELING

Simulated tracks are used to train the track-finding pipeline, and to assess reconstruction efficiency, generalization performance, and rate of background tracks. Below, the simulation of the detector response, generation of helical standard model (SM) background tracks and generation of nonhelical tracks are described. Training and testing samples contain combinations of many helical SM tracks and at most one nonhelical track.

A. Detector geometry

A custom detector geometry is used, with 25 coaxial cylindrical layers with a length of 320 cm and even radial spacing ranging between 3.1 and 53 cm. Tracks are generated by producing trajectories which originate at the center of the detector.

Hits are recorded at every intersection of each trajectory with a detector layer until the particle's path leaves the detector, because the pipeline is trained to find sets of hits that correspond to continuous trajectories which remain within the volume. Many trajectories do return to the detector and leave additional hits, effectively producing several disconnected sets of hits from a single trajectory. While piecing together several such sets may give additional discovery power, it also raises a new and distinct set of reconstruction challenges, and so is left to future work.

Particle energy loss and hit location uncertainty are neglected, but similar work did not show strong sensitivity [16].

B. SM background

Madgraph5 [24] is used to generate $pp \rightarrow t\bar{t}$ events, which are showered and hadronized using Pythia8 [25]. Initial velocity and momenta are used to propagate Standard Model particles in helical trajectories under the assumption that there is no energy loss or secondary scattering.

All events use a pileup of $\mu = 1$. The impact of pileup on performance is not yet known, but in other studies the same GNN pipeline performed well in high pileup conditions similar to those expected at the high-luminosity LHC [26]. Such performance may extend to the case of generalized smooth tracks, or it may be more susceptible to confusion in such hit dense environments. A rigorous study is out of scope for this initial proof-of-principle.

C. Nonhelical track generation

The model-agnostic search aims to be sensitive to a very broad range of possible new particles with unexpected trajectories through the detector. However, it cannot accept every possible trajectory, or all sets of hits would be candidates. Instead, particle tracks are required to follow *smooth* paths, those which can be parametrized by a curve, $\vec{x}(t)$, where all orders of spatial derivatives with respect to time, $\frac{d^n \vec{x}}{dt^n}$, exist and are finite, which accepts all particles with physical paths, where the net force and its derivatives, $m \frac{d^n \vec{x}}{dt^n}$, are physical.¹ It rejects candidates whose paths are discontinuous or kinked. This method of track generation narrows our analysis by disregarding nonprompt decays, and particles that scatter off of detector material. While many particles exhibit nonsmooth behavior in detector environments, a smoothness assumption provides a theoretically sound foundation for demonstrating this proof-of-concept analysis.

Without assumption about the underlying physical theory, three-component spatial curves can be fully described in a Fourier basis by three sequences of Fourier amplitudes, (a_n, b_n, c_n) , defined in a Cartesian basis. Such paths, however, are not necessarily smooth unless each sequence approaches zero as $n \rightarrow \infty$ at least as fast as a Schwartz function $f(n)$ [28–30]. Explicitly, smoothness is guaranteed if $a_n, b_n, c_n \leq f(n)$ for all n .

Figure 1 shows example trajectories generated with a small number of frequencies, as well as one following Schwartz function 1 defined in Table V.

Figure 2 provides a visual example of generating a smooth amplitude by limiting amplitudes at higher frequencies. Note that a three dimensional path requires three such curves, one for each coordinate.

While all trajectories generated using a Schwartz function limit are smooth, and for any smooth trajectory there is a Schwartz function that would allow its generation (proofs in Appendix A 2), in principle describing the full space of smooth tracks may require an infinite number of frequency modes and many Schwartz functions. In practice, sampling from infinitely many frequencies is not computationally feasible, but Schwartz functions' rapid decay makes high-frequency modes nearly vanish. Sampling from a large number of frequencies, 25–30, therefore explores a significant fraction of the space. Particle displacement resulting from frequencies beyond this would contribute negligible oscillatory motion, much less than the radial spacing of detector layers.

Similarly, a set of Schwartz functions is selected to explore the space, which does not span the full, infinite

¹While this may be untrue for idealized or simplified systems which exhibit discontinuities in external forces, fully physical classical systems disregarding quantum effects such as decays are subject to classical field interactions, which are infinitely differentiable with respect to position and time [27].

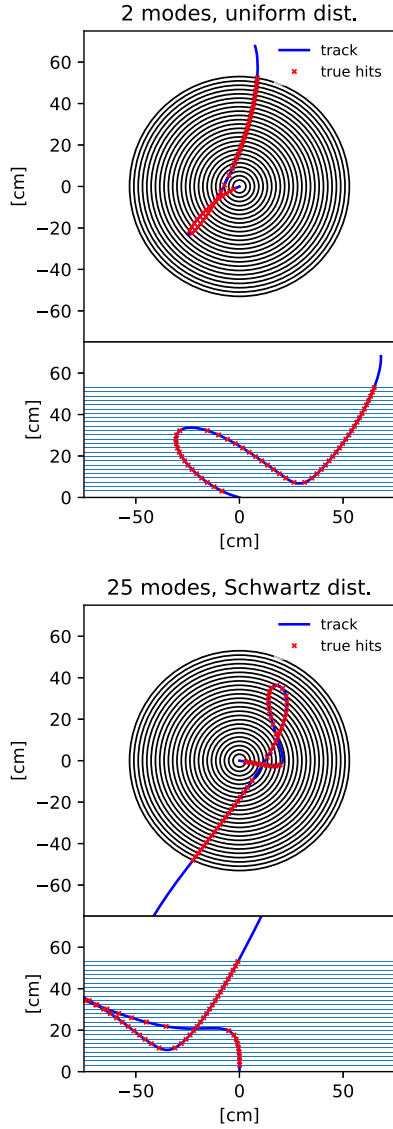


FIG. 1. Trajectories (blue) and hits (red) for tracks with 1 (top left), 2 (top right), 4 (bottom left), or 25 (bottom right) Fourier modes, where the 25-mode track follows a Schwartz function to ensure smoothness. Shown are the $x - y$ (top) and $r - z$ (bottom) projections.

space, but describes tracks significantly beyond the set of helical tracks. The chosen Schwartz functions include helices and explicit parametrization of helical tracks in a Fourier basis is shown in Appendix A 2.

For demonstration, nonhelical tracks with a small number of frequencies are also prepared, by sampling from finitely many frequencies in Fourier space, where each mode has a maximum amplitude of 60 cm. An N -frequency track is defined as the collection of points where a function $\vec{g}_N(t) = \sum_{i=1}^3 \sum_{n=-N}^N c_{ni} \exp(\frac{2\pi i n t}{\lambda}) \hat{x}_i$ intersects the detector layers, where $\hat{x}_i$ is a Cartesian basis, and c_{ni} are all randomly generated with a uniform distribution in the interval $[0, 60]$ cm, and λ is set equal to the radius of

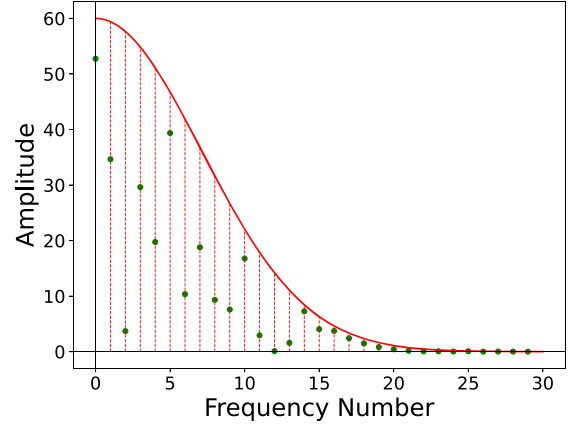


FIG. 2. Example of a Schwartz function (red), which enforces a falling upper limit on amplitude as frequency grows, guaranteeing smoothness if amplitudes are selected (green dots) below the upper limit.

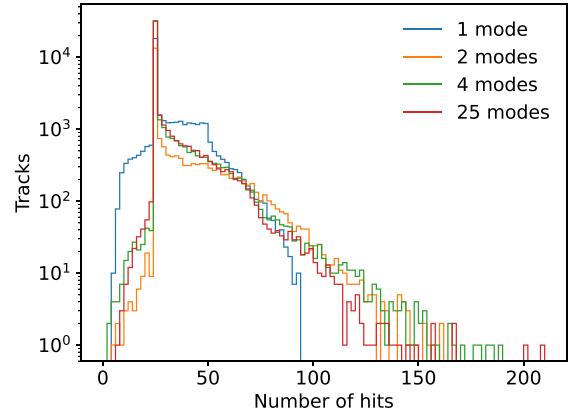


FIG. 3. Shown are the distributions of number of hits per generated track depending on the number of frequency modes used in the trajectory generation.

the detector which is a choice further discussed in Appendix A 3.

The distribution of the hit multiplicity for tracks with varying number of frequencies is shown in Fig. 3; while a significant number have as many hits as there are detector layers, a large fraction have many more hits. The data used for training this pipeline can be generated at the publicly available github [31].

III. TRACKING

Track reconstruction involves two distinct steps, *finding*, where a set of hits are identified as a track candidate, and *fitting*, which finds best values of parameters in an assumed form to minimize track-to-hit distance. Reconstructed tracks are crucial elements of data analysis in many areas, including triggers, vertex finding and jet flavor tagging.

Many tracking algorithms, such as a Kalman Filter, assume a parametric track path and ionization and

interaction properties typical of known particles. An initial set of hits allows for initial estimates of the track parameters, which in turn allows for propagation of the track to subsequent detector surfaces and a calculation of a window within which additional hits should be considered. This hit-by-hit approach naturally weaves together track finding and fitting. Machine learning can be used in a variety of ways to perform track finding [32–34]. For example, ML can be used to propagate initial tracks in a Kalman Filter [35], or more broadly to select the next hit [36,37].

The machine-learning-based approach used below does not require an explicit model of the trajectory. Instead, the relationships between hits on the same track are implicitly defined through the example tracks in the training sample. A track is represented as a directed graph, with hits as nodes. Track-finding pipelines, such as `Exa.TrkX`, which use graph neural networks achieve high track-finding efficiencies in a broad variety of particle experiments, even under high pileup conditions [38–49]. The `Exa.TrkX` pipeline first performs graph construction, where hits are linked into a graph, followed by edge classification, where graph edges are evaluated as their likelihood to connect two hits which belong to the same track, followed by graph segmentation, where edges are pruned and collected hit connections form track candidates. The `Exa.TrkX` pipeline employed here relaxes the requirement that hits are ordered in time according to their distance from the vertex, following Ref. [16]. Graph edges are evaluated by a metric learning model, which learns a mapping from the physical space of hits to an abstract latent space, such that hits on the same track are near each other and hits from other tracks are more distant.

The `Exa.TrkX` pipeline employed requires information about the hit ordering. Traditionally, the assumption is made that hit ordering follows the layers from inside out. As tracks may change direction, in this application the true hit ordering is used following Ref. [16]. For SM tracks, this has little impact as backscattering toward the production vertex occurs infrequently, and the fake rate remains low.

The metric learning model is trained for 5 epochs using on the order of 10^4 training events with a learning rate of 0.001. The edge classification model is trained for another 5 epochs, using early stopping and variable learning rate to avoid overfitting the data.

The primary performance metric is the tracking efficiency, the fraction of *reconstructable* particles (those with at least 22 hits) that are *reconstructed*, double-majority matched to a *matchable* track candidate (those with at least 15 hits). A track is double-majority matched to a track candidate when the majority of hits in the reconstructed track belong to the same truth track, and the majority of hits in the truth track belong to the reconstructed track. Tracks are labeled as *fake* if they do not double majority match any SM or nonhelical track. The pipeline may also be capable

TABLE I. Shown is track finding efficiency for nonhelical tracks in various testing and training configurations. SM refers to helical tracks, n Freq refers to tracks generated with n Fourier modes, and SS refers to tracks generated with 25 Fourier modes following a specific Schwartz function.

Train	Test	Efficiency	Fake rate
SM	SM + Signal		
SM	SM + 1 freq.	0.028 ± 0.008	0.383
SM	SM + 2 freq.	0.019 ± 0.006	0.019
SM	SM + 3 freq.	0.017 ± 0.004	0.391
SM	SM + 4 freq.	0.029 ± 0.002	0.200
SM	SM + SS 1	0.034 ± 0.009	0.423
Signal	Signal		
1 freq.	1 freq.	0.964 ± 0.006	0.001
2 freq.	2 freq.	0.962 ± 0.004	0.000
3 freq.	3 freq.	0.983 ± 0.002	0.001
4 freq.	4 freq.	0.993 ± 0.002	0.001
SS 1	SS 1	0.993 ± 0.001	0.000
SM + Signal	SM + Signal		
SM + 1 freq.	SM + 1 freq.	0.475 ± 0.020	0.002
SM + 2 freq.	SM + 2 freq.	0.561 ± 0.013	0.002
SM + 3 freq.	SM + 3 freq.	0.571 ± 0.011	0.001
SM + 4 freq.	SM + 4 freq.	0.626 ± 0.009	0.001
SM + SS 1	SM + SS 1	0.589 ± 0.009	0.001

of reconstructing shorter tracks; such studies are reserved for future work.

A. Baseline tests

The expected performance of the pipeline is verified on SM tracks, training on 40k SM $pp \rightarrow t\bar{t}$ events, and testing on 4k SM events. The reconstruction efficiency is 99.9% and the fake rate is consistent with zero.

To estimate of the difficulty of finding nonhelical tracks with traditional helical approaches, the ability of the SM-trained pipeline to reconstruct our nonhelical tracks is measured in events with one nonhelical track added to an SM event. Efficiencies on the nonhelical tracks are given in Table I, and are quite poor, indicating as expected that helical track finders are not well suited to finding nonhelical tracks.

The ML-based tracking pipeline has no inherent preference for helical tracks, and should allow for nonhelical track finding simply by preparing the appropriate training sample. A nonhelical track finder is trained with a sample of pure nonhelical tracks, one per event. Training and testing on such events achieves a very high efficiency (see Table I), demonstrating that the pipeline has the capacity in principle to learn the patterns of hits on a wide variety of nonhelical tracks.

In practice, nonhelical tracks are expected to be found among many SM tracks. In the next stage, SM tracks are added to the training sample and given the same

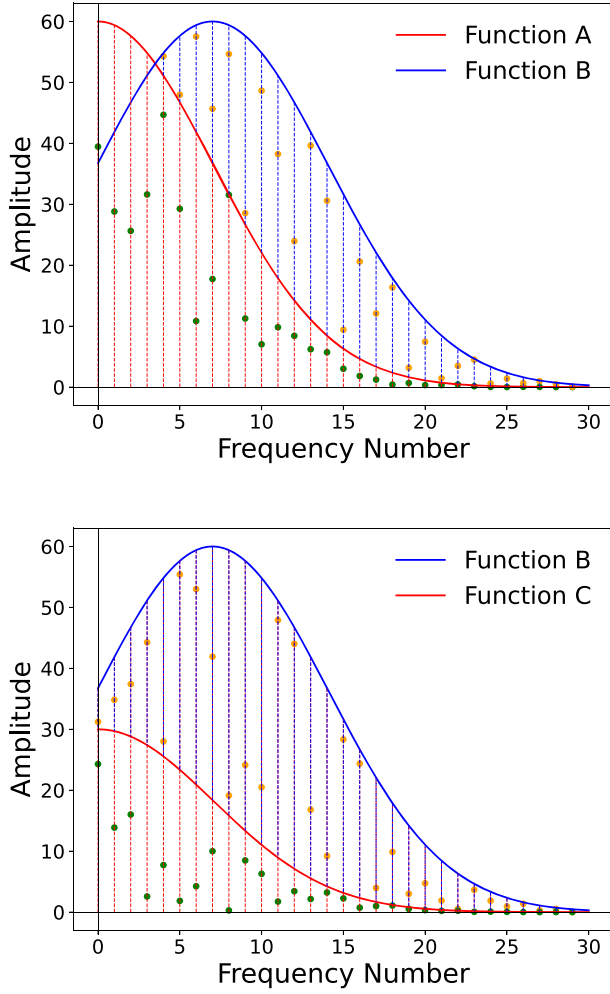


FIG. 4. Visualization of tracks generated in disjoint spaces defined by three example Schwartz functions A, B, and C. In green are Fourier coefficients generated under the red curve. In yellow are the coefficients generated between the blue and red curves. The two resulting tracks cannot have the same Fourier representation, and must be different.

labels.² Efficiency is lower in the presence of many background tracks, as expected, but remains strong; see Table I. Even in the presence of SM backgrounds, the tracking pipeline can efficiently find nonhelical tracks.

B. Generalization tests

The training sample is necessarily finite, and is constructed using a finite number of Fourier modes from one of many Schwartz functions. An important question is whether the tracker has learned the general properties of smooth tracks, allowing it to generalize outside of its

²In previous work which attempted to learn one specific nonhelical track pattern [16], SM tracks were labeled as fake to suppress their reconstruction. In contrast, here the aim is to learn all possible smooth tracks.

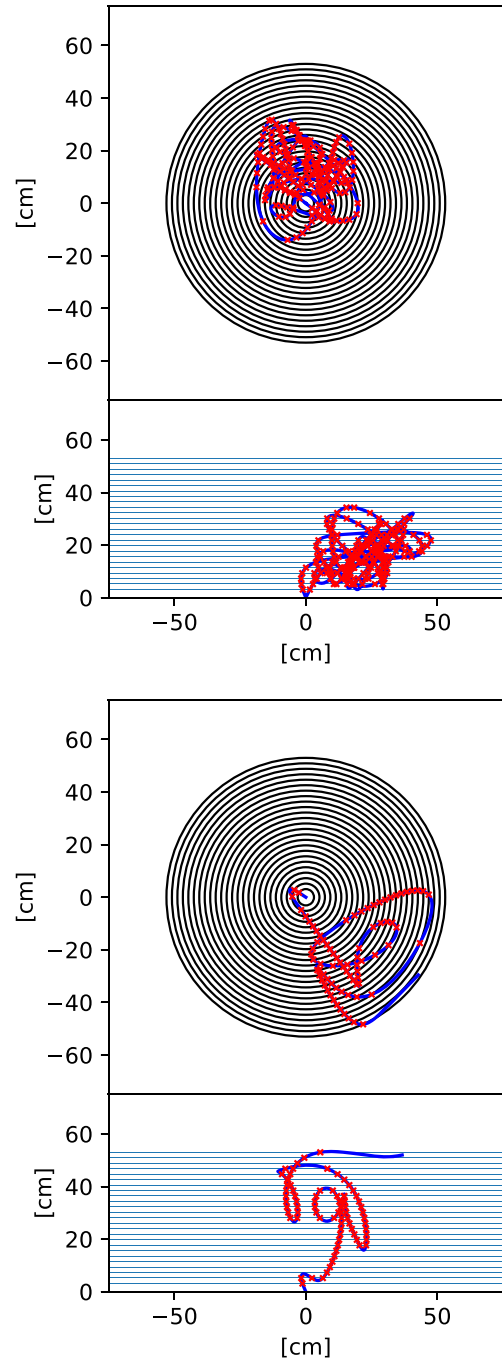


FIG. 5. Examples of tracks generated in the 6-5 space defined in the text.

training sample. Previous work [16] showed hints that the tracking pipeline may be capable of generalization, learning to reconstruct tracks which fall outside of the space described by the training set.

To measure the capacity for generalization, a different Schwartz function is chosen for the training and testing sets. The testing set includes only the disjoint subset, to ensure no overlap with the training set. This is a very strong condition in Fourier space, guaranteeing that any amplitude

TABLE II. Shown is the track finding efficiency for nonhelical tracks when training and testing are performed on disjoint spaces as described in Sec. III B. Training and testing both have nonhelical tracks and SM background.

Train	Test	Efficiency	Fake rate
SM + SS 1	SM + SS 2-1	0.775 ± 0.007	0.000
SM + SS 3	SM + SS 4-3	0.668 ± 0.008	0.001
SM + SS 1	SM + SS 8-1	0.667 ± 0.009	0.001
SM + SS 3	SM + SS 2-3	0.592 ± 0.008	0.008
SM + SS 1	SM + SS 9-1	0.590 ± 0.009	0.001
SM + SS 11	SM + SS 8-11	0.579 ± 0.009	0.001
SM + SS 4	SM + SS 2-4	0.546 ± 0.010	0.011
SM + SS 7	SM + SS 4-7	0.497 ± 0.014	0.001
SM + SS 1	SM + SS 4-1	0.414 ± 0.010	0.002
SM + SS 5	SM + SS 6-5	0.008 ± 0.002	0.002

of a particular mode cannot be generated in both the training and testing sets. Therefore, the pipeline cannot perform well in testing by simply learning a Fourier representation of the training tracks. Even if transformed to another basis, these spaces will still be disjoint, and crucially, the network would be unable to succeed by learning *any* particular basis representation of tracks. The tracker's learning must be basis-independent for successful reconstruction of tracks in a testing set disjoint from its training set. To succeed, the pipeline must learn more general properties of smoothness. Figure 4 gives a visual representation of disjoint spaces. Examples from the disjoint training set 6-5, and 4-3 are shown in Figs. 5 and 6. Examples from the disjoint training set 4-3 are shown.

Training and testing samples again consist of $pp \rightarrow t\bar{t}$ events with a single added nonhelical track. In every case, the training sample contains 40k events and the testing sample 4k events. The majority of these tests result in high efficiency, as shown in Table II, demonstrating generalization beyond the training set. The disjoint testing sets indicate that the network does not learn a basis-dependent representation of tracks but learns something more general about smooth tracks, and should therefore be able to generalize its training further to other spaces of smooth tracks.

In one case, training on Schwartz space 5 and testing on 6-5 gives poor performance. The testing set has very long tracks, as shown in Fig. 5, which makes reconstruction under the double matching criterion very difficult. Generalization is possible, but not universal or guaranteed.

C. Dependence on training functions

The selection of the Schwartz function clearly impacts the capacity of the network to find tracks, both within and beyond its training set. The form $f(x) = Ae^{-\alpha(x-c)^2}$ is chosen, where A is the maximum amplitude, α represents the width, and c is the location of the center.

TABLE III. Dependence of efficiency on maximum amplitude parameter of training function. Efficiency is for nonhelical tracks; training and testing samples include SM backgrounds.

Amplitude (cm)	Center	Width	Efficiency	Fake rate
30			0.718 ± 0.007	0.184
37.5			0.594 ± 0.008	0.001
45			0.771 ± 0.007	0.186
52.5			0.562 ± 0.008	0.001
60			0.562 ± 0.008	0.001
	0		0.562 ± 0.008	0.001
	1.75		0.553 ± 0.008	0.001
	3.5		0.546 ± 0.008	0.001
	5.25		0.546 ± 0.008	0.001
	7		0.536 ± 0.008	0.001
		0.01	0.562 ± 0.008	0.001
		0.015	0.562 ± 0.008	0.001
		0.02	0.568 ± 0.008	0.001
		0.025	0.594 ± 0.008	0.001
		0.03	0.581 ± 0.008	0.001
		0.035	0.577 ± 0.008	0.001

In this section, the dependence of the performance on the values of these parameters for the training function is explored; the values of the testing parameters are fixed at $A = 45$, $\alpha = 0.02$, and $c = 3.5$. These values are chosen because they each lie in the center of the ranges used for each parameter, making for straightforward comparison of training parameters that are either above or below the testing values.

The n th Fourier mode has an amplitude drawn uniformly from $[0, f(n)]$. A maximum amplitude larger than the detector radius is likely to take the track outside of the detector, preventing oscillatory motion that helps distinguish these tracks from the SM background. A range of amplitudes between 30 and 60 cm is therefore considered, for a detector with radius of 50 cm. Efficiencies are somewhat higher for smaller values, though fake rates also increase; see Table III. To balance these, $A = 52.5$ cm is selected.

A similar analysis is performed for the center and width of the training functions. The position of the center is varied in Table III. The efficiency decreases monotonically as the center shifts farther from zero. A center at the origin is chosen.

Table III shows the efficiency under variation of the width parameter. A local maximum of efficiency is reached at a width of 0.025. The final training function corresponds to Schwartz function 19 in Table V.

D. Results

The efficiency of the pipeline when trained using Schwartz function 19 is shown in Table IV for a selection of testing samples. The training sample size has been increased to 65K events, each with one nonhelical track and SM tracks from one SM event. This pipeline has a strong

TABLE IV. Tracking efficiency and fake rate of a pipeline trained on SM and Schwartz set 19, for several testing samples.

Test	Efficiency	Fake rate
SM + Schwartz set 1	0.764 ± 0.007	0.000
SM + Schwartz set 2	0.755 ± 0.007	0.000
SM + Schwartz set 4	0.489 ± 0.008	0.001
SM + Schwartz set 7	0.472 ± 0.008	0.001
SM + Schwartz set 8	0.662 ± 0.008	0.001
SM + Schwartz set 9	0.627 ± 0.008	0.001
SM + Schwartz set 10	0.491 ± 0.008	0.001
SM + Schwartz set 19	0.504 ± 0.008	0.001
SM + Schwartz set 21-19	0.647 ± 0.010	0.001
SM + Schwartz set 22-19	0.590 ± 0.010	0.001
SM + Schwartz set 23-19	0.372 ± 0.010	0.001

capacity to reconstruct many varieties of smooth tracks when correctly trained, while maintaining a low fake rate.

The ability of the pipeline to generalize when trained on Schwartz function 19 is also shown, by the performance reported in Table IV when testing on several spaces that are disjoint from set 19. Performance remains high; thus, it is expected that the pipeline has a strong ability to generalize when trained on this function.

E. Quirk reconstruction

The nonhelical track finding pipeline is applied in this section to a specific BSM theory, quirks [13] which leave nonhelical tracks. A measure of the pipeline's ability to recover physically motivated signals is provided by this section.

Training is again performed on 65 K events with a combination of nonhelical tracks from Schwartz function 19 and SM tracks as in Sec. III D, and testing is performed on 1 K events containing both SM and quirk tracks. Following the quirk generation scheme of [16], an overall efficiency of 32% on quirk tracks with 0 fakes is reported. The reconstruction efficiency versus the p_T of the quirk pair, the opening angle between pair produced quirks, the oscillation amplitude, and the oscillation period is displayed in Fig. 7. Examples of reconstructed quirk tracks are shown in Fig. 8.

F. Rejection of standard model tracks

To claim the discovery of a new particle from a small number of nonhelical tracks requires some technique to distinguish them from the large background of SM tracks, which are also reconstructed by the pipeline.

Helical tracks can be rejected by fitting their points to a helical trajectory and evaluating the goodness of fit, via χ^2 . For each track, an initial rapid estimation of the helical parameters is made [50], followed by an exploration of parameter values in the vicinity, and refinement with the

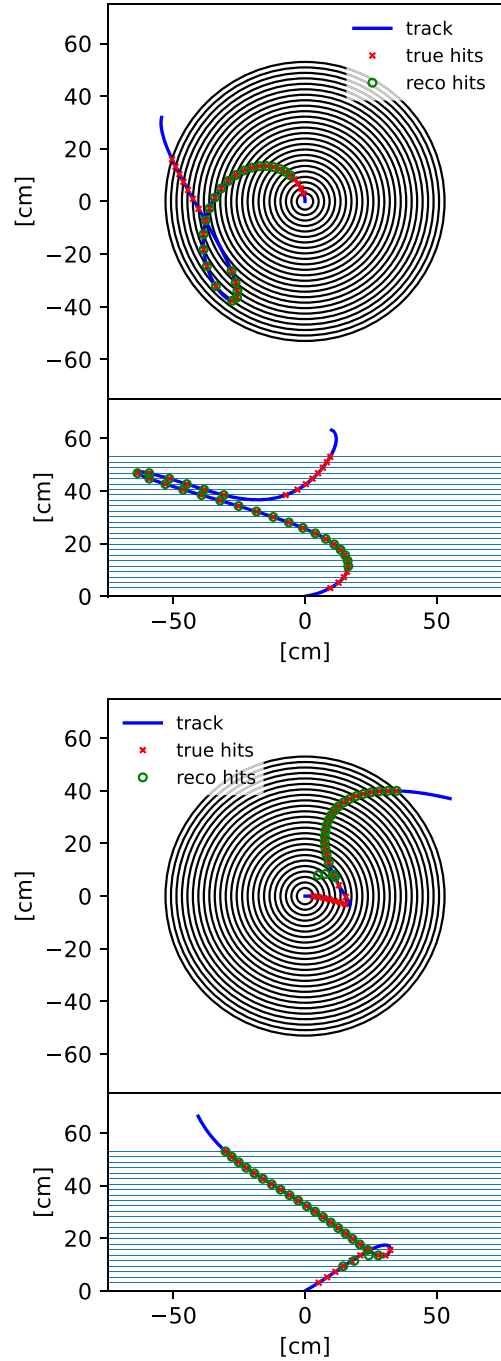


FIG. 6. Examples of tracks generated in the 4-3 space. The reconstructed (predicted) tracks are shown as well.

Nelder-Mead method [51] implemented in SCIPY.MINIMIZE [52]. The distribution of χ^2 for SM and nonhelical tracks is shown in Fig. 9. While some SM tracks will have larger χ^2 , and some nonhelical tracks have low χ^2 , there is a significant population of nonhelical tracks with large χ^2 and which are striking by eye. Several examples of fitting helices to hits left by nonhelical tracks are shown in Fig. 10.

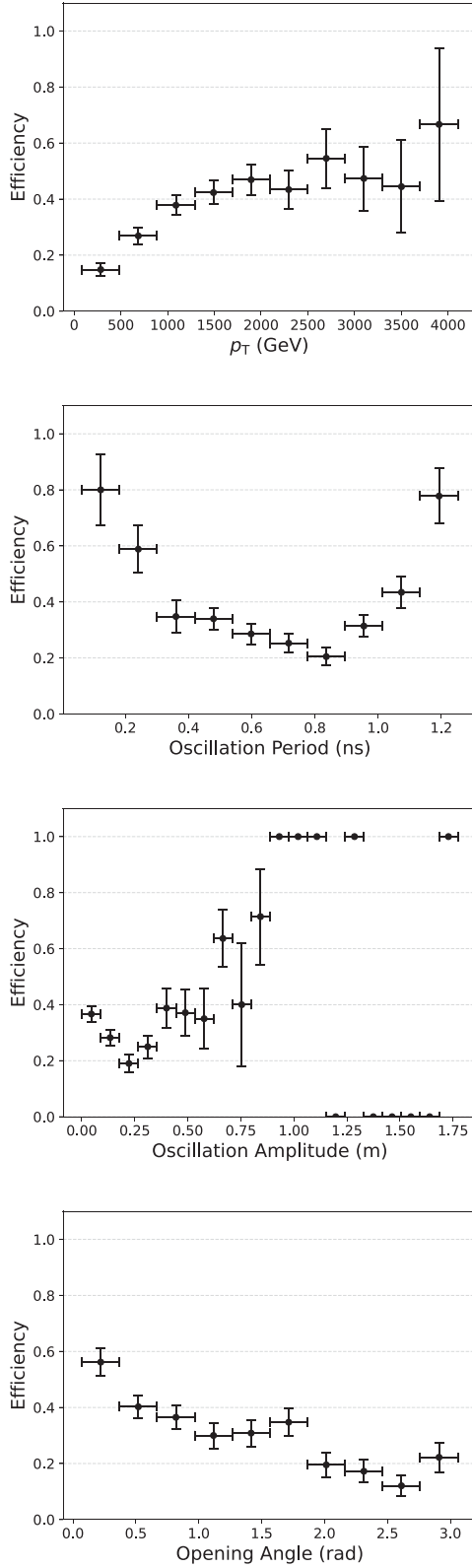


FIG. 7. Shown is the reconstruction efficiency on quirks with respect to p_T , oscillation period, amplitude, and opening angle.

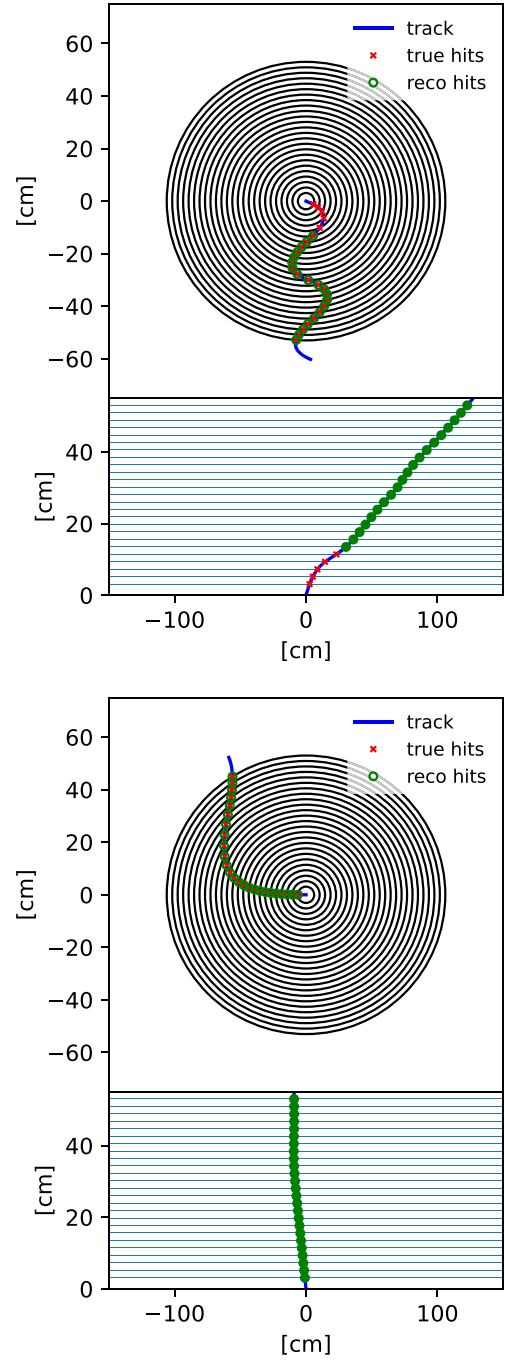


FIG. 8. Examples of reconstructed (predicted) quirk tracks.

IV. DISCUSSION

The methods presented above can, in principle, generate all possible smooth tracks. In practice, requirements that generation time be finite limit the space somewhat, but still encompass a great variety of trajectories beyond the traditional helix.

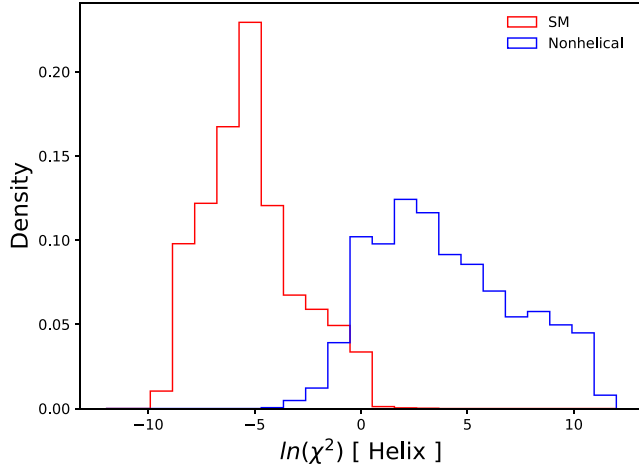


FIG. 9. Distribution of χ^2 values for helical fits to helical SM and nonhelical tracks.

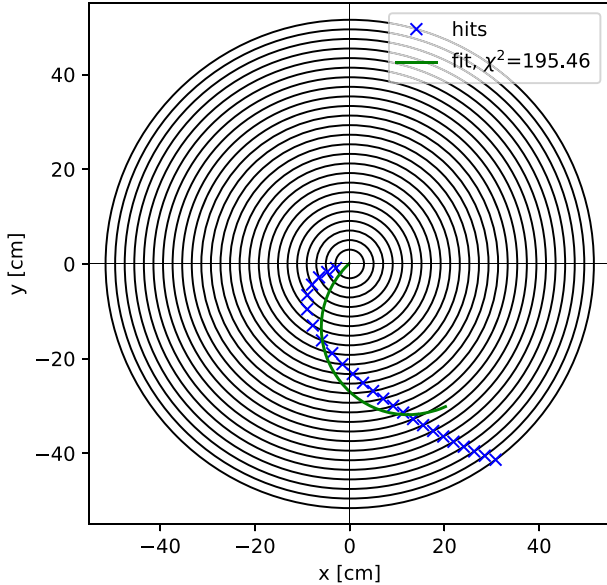


FIG. 10. Example helical fit to a nonhelical track.

Nonhelical tracks are extremely challenging for helical tracking codes, as expected. While some traditional algorithms can be modified for specific alternative trajectories, the machine learning pipeline described here has the capacity to learn a broad set of trajectories without the need for explicit parametric specification. The trajectories are specified implicitly via the training sample, and the power of ML to interpolate provides generalization power. Reconstruction performance for nonhelical tracks, among a background of helical tracks, is strong, even for tracks outside of the training sample.

A specific training set is identified which maximizes performance on tracks within that set while maintaining reasonable performance in tracks outside the set. The fake rate remains very low from hits not associated with real tracks.

Helical track fitting is a powerful discriminant, but the challenge of filtering helical tracks from nonhelical remains significant due to the expected low relative rate.

V. CONCLUSIONS

The advent of machine learning tools allows for the relaxation of simplifying assumptions in particle tracking that have long limited the power to discover unusual trajectories.

In a simplified setting, this study provides a proof of principle that nonhelical tracks can be found without the need to specify any particular trajectory. This is a first step toward development of a tracking pipeline which could, for the first time, reveal nonhelical surprises that may be lurking in current data samples and be striking to the naked eye.

Future work includes more realistic samples, including hit noise models, misalignments, and hard radiation among other effects. The use of physics-oriented constraints that restrict tracks beyond the smoothness assumption, and an investigation of the effects of the Standard Model background process on the performance of this pipeline should also be studied. Other work could include hybrid approaches where the helical tracks are fitted and removed, and the remaining hits are fitted to model agnostic tracks following the methods in this work. Additionally, further study of detector design could explore the capacity to which recording additional information with each hit, such as directional and timing information, could enhance discovery power when combined with works such as this.

ACKNOWLEDGMENTS

The authors are grateful to Makayla Vessella and Paata Ivanisvili for helpful discussion. L.C. and D.W. are supported by the DOE Office of Science.

DATA AVAILABILITY

The data that support the findings of this article are openly available [31].

APPENDIX

1. Schwartz functions

Selected Schwartz functions are shown in Table V.

TABLE V. Selected Schwartz functions.

Number	Expression
1	$f_1(n) = 60 \cdot \exp(-0.01n^2)$
2	$f_2(n) = 60 \cdot \exp(-0.01(n-7)^2)$
3	$f_3(n) = 30 \cdot \exp(-0.01n^2)$
4	$f_4(n) = 30 \cdot \exp(-0.01(n-7)^2)$
5	$f_5(n) = 10 \cdot \exp(-0.01n^2)$
6	$f_6(n) = 10 \cdot \exp(-0.01(n-7)^2)$
7	$f_7(n) = 45 \cdot \exp(-0.01n^2)$
8	$f_8(n) = 45 \cdot \exp(-0.01(n-7)^2)$
9	$f_9(n) = 45 \cdot \exp(-0.01(n-3.5)^2)$
10	$f_{10}(n) = 60 \cdot \exp(-0.03n^2)$
11	$f_{11}(n) = 45 \cdot \exp(-0.02(n-3.5)^2)$
12	$f_{12}(n) = 60 \cdot \exp(-0.01(n-3.5)^2)$
13	$f_{13}(n) = 60 \cdot \exp(-0.02n^2)$
14	$f_{14}(n) = 52.5 \cdot \exp(-0.01n^2)$
15	$f_{15}(n) = 37.5 \cdot \exp(-0.01n^2)$
16	$f_{16}(n) = 60 \cdot \exp(-0.01(n-1.75)^2)$
17	$f_{17}(n) = 60 \cdot \exp(-0.01(n-5.25)^2)$
18	$f_{18}(n) = 60 \cdot \exp(-0.015n^2)$
19	$f_{19}(n) = 60 \cdot \exp(-0.025n^2)$
20	$f_{20}(n) = 60 \cdot \exp(-0.035n^2)$
21	$f_{21}(n) = 60 \cdot \exp(-0.025(n-3.5)^2)$
22	$f_{22}(n) = 45 \cdot \exp(-0.025(n-7)^2)$
23	$f_{23}(n) = 30 \cdot \exp(-0.03(n-7)^2)$

2. Schwartz space properties

It is shown here that the method of smooth track generation detailed in Sec. II C is capable of generating every smooth track.

Curves are generated by choosing a Schwartz function, $s \in \mathcal{S}$, as an upper bound for 3 separate sequences of Fourier coefficients, where $\mathcal{S}$ is the set of Schwartz functions. A sequence is chosen for each spatial coordinate, so tracks can be generated that can be parametrized $(f(t), g(t), h(t))$ for $t \in [0, 2\pi]$, with periodic components with a period of 2π and $f(0) = g(0) = h(0) = 0$. For each mode of oscillation, n , we randomly choose a c_{n_x} , c_{n_y} , and c_{n_z} in $[0, s(n)]$. To maintain rotational invariance in our Fourier space, we reject all three and choose again if $c_{n_x}^2 + c_{n_y}^2 + c_{n_z}^2 > s^2(n)$, so we can think of the three coefficients as being a single point chosen with a uniform distribution inside of some sphere of radius $s(n)$. Thus, we can think of a single track as generated by a sequence of points, where the n th point is chosen to lie inside a sphere of radius $s(n)$. If $f(t)$, $g(t)$, and $h(t)$ are differentiable with respect to time to all orders, the track is called “smooth.”

To show that any smooth track can be generated by choosing the correct Schwartz function, suppose that $g(t) \in \mathbb{R}$ is smooth. We want to show that there exists a function $s \in \mathcal{S}$ for which the $C_n \leq s(n)$ for every n , where the sequence C_n is the Fourier representation of g . This can also be stated as

$$C_n = \frac{1}{2\pi} \int_0^{2\pi} g(x) e^{inx} dx.$$

We can then integrate by parts to show that

$$C_n = -\frac{1}{2\pi} \frac{1}{ni} \int_0^{2\pi} g'(x) e^{inx} dx,$$

and if we now integrate by parts k times,

$$C_n = (-1)^k \frac{1}{2\pi} \left(\frac{1}{ni} \right)^k \int_0^{2\pi} g^k(x) e^{inx} dx.$$

Therefore, it must be the case that

$$\left| \int_0^{2\pi} g^k(x) e^{inx} dx \right| \leq 2\pi \max(g^k(x))$$

for $x \in [0, 2\pi]$, and thus it can be immediately shown that

$$|C_n| = \left| \frac{1}{2\pi} \left(\frac{1}{ni} \right)^k \int_0^{2\pi} g^k(x) e^{inx} dx \right| \leq \left(\frac{1}{n} \right)^k \max(g^k(x)).$$

Therefore C_n is bounded from above by a Schwartz function, and every smooth track can be generated by careful choice of Schwartz function.

It is also important that this method *only* generates smooth tracks regardless of choice of Schwartz function. This is easy to show, as the Fourier transform is an isomorphism onto the Schwartz space. This means that choosing a sequence of Fourier coefficients to lie on the Schwartz function, $C_n = s(n)$, makes the resulting function Schwartz. If instead $C_n \leq s(n)$, then the resulting function is even more rapidly decaying, and is thus still Schwartz, and thus smooth.

It is illustrative to describe helical tracks in a Fourier basis and show that they are a subset of tracks that can be generated by the Schwartz function method described. In tracking, helices are typically fully described by the five parameters $(\phi_0, p_T, d_0, d_z, \tan(\lambda))$ where ϕ_0 is the initial angle of the particle's trajectory with respect to the z axis, p_T is the transverse momentum, d_0 is the distance of closest approach to the origin in the x - y plane, d_z is the distance of closest approach to the origin along the z direction, and $\tan(\lambda)$ is the slope of the helix in the r - z plane. The position of a particle for any time t can then be described:

$$\begin{cases} x = d_0 \cos(\phi_0) + \alpha p_T \{\cos(\phi_0) - \cos(\phi_0 - \omega t)\} \\ y = d_0 \sin(\phi_0) + \alpha p_T \{\sin(\phi_0) - \sin(\phi_0 - \omega t)\} \\ z = d_z - \alpha p_T \tan(\lambda) \omega t \end{cases},$$

where $\alpha = \frac{1}{qB}$ is the magnetic field constant.

This form can be expanded in a Fourier basis by first recognizing the form of the x and y components to already be trivially in a Fourier basis and expanding the z component,

$$z = d_z + \alpha p_T \tan(\lambda) \omega t - \sum_{n=1}^{\infty} \frac{2\alpha p_T \tan(\lambda)}{n} \sin(n\omega t).$$

For any value of t in a compact interval and any values of track parameters, there exists a Schwartz function that generates all the correct coefficients for this helical form. With the assumption that tracks start at the origin, the Schwartz functions chosen in this work will generate this form of helical track if $\alpha p_T \tan(\lambda)$ is not too large.

3. Signal track generation

Our signal track generation consists of producing a spatial curve via Fourier series, making a very large number of points on the curve which are evenly spaced by arc length, and recording the point closest to each layer that the curve passes through and labeling it a “hit.”

When producing the Fourier series, the fundamental wavelength is arbitrarily chosen to be equal in length to the detector radius. So long as sufficiently many frequencies are included in track generation, this will not limit the tracks that can be generated. However, this does put constraints on the tracks, such as those in Sec. III A, that are generated with a finite number of frequencies. When making this arbitrary choice it is only important that the fundamental wavelength is a similar length to the radius of the detector, so that the finite frequency tracks do not leave the detector too soon or too late.

-
- [1] G. Aad *et al.* (ATLAS Collaboration), *Phys. Lett. B* **716**, 1 (2012).
 - [2] S. Chatrchyan *et al.* (CMS Collaboration), *Phys. Lett. B* **716**, 30 (2012).
 - [3] T. Gehrmann and B. Malaescu, *Annu. Rev. Nucl. Part. Sci.* **72**, 233 (2022).
 - [4] P. Ferreira da Silva, *Annu. Rev. Nucl. Part. Sci.* **73**, 255 (2023).
 - [5] R. Aaij *et al.* (LHCb Collaboration), *Phys. Rev. Lett.* **115**, 072001 (2015).
 - [6] R. Aaij *et al.* (LHCb Collaboration), *Phys. Rev. Lett.* **113**, 151601 (2014).
 - [7] G. Aad *et al.* (ATLAS Collaboration), *Phys. Rev. Lett.* **105**, 252303 (2010).
 - [8] V. Khachatryan *et al.* (CMS Collaboration), *J. High Energy Phys.* **09** (2010) 091.
 - [9] A. Strandlie and R. Fruhwirth, *Rev. Mod. Phys.* **82**, 1419 (2010).
 - [10] G. Aad *et al.* (ATLAS Collaboration), *Comput. Software Big Sci.* **8**, 9 (2024).
 - [11] ATLAS Collaboration, *Phys. Rev. Lett.* **124**, 031802 (2020).
 - [12] A. Abulencia *et al.* (CDF Collaboration), *Phys. Rev. Lett.* **96**, 201801 (2006).
 - [13] J. Kang and M. A. Luty, *J. High Energy Phys.* **11** (2009) 065.
 - [14] J. A. Evans and M. A. Luty, *J. High Energy Phys.* **06** (2019) 090.
 - [15] S. Knapen, H. K. Lou, M. Papucci, and J. Setford, *Phys. Rev. D* **96**, 115015 (2017).
 - [16] Q. Sha, D. Murnane, M. Fieg, S. Tong, M. Zakharyan, Y. Fang, and D. Whiteson, *arXiv:2410.00269*.
 - [17] C. D. Anderson, *Phys. Rev.* **43**, 491 (1933).
 - [18] ATLAS Collaboration, *Eur. Phys. J. C* **77**, 673 (2017).
 - [19] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, *IEEE Signal Process. Mag.* **34**, 18 (2017).
 - [20] X. Ju *et al.* (Exa.TrkX Collaboration), in *33rd Annual Conference on Neural Information Processing Systems* (2020), *arXiv:2003.11603*.
 - [21] X. Ju, D. Murnane, P. Calafiura, N. Choma, S. Conlon, S. Farrell, Y. Xu, M. Spiropulu, J.-R. Vlimant, A. Aurisano *et al.*, *Eur. Phys. J. C* **81**, 876 (2021).
 - [22] S. Caillou, P. Calafiura, X. Ju, D. Murnane, T. Pham, C. Rougier, J. Stark, and A. Vallier (ATLAS Collaboration), *EPJ Web Conf.* **295**, 03030 (2024).
 - [23] S. Caillou, P. Calafiura, S. A. Farrell, X. Ju, D. T. Murnane, C. Rougier, J. Stark, and A. Vallier ATLAS ITk Track Reconstruction with a GNN-based pipeline (2022), *10.5281/zenodo.8119762*.
 - [24] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli, and M. Zaro, *J. High Energy Phys.* **07** (2014) 079.
 - [25] C. Bierlich *et al.*, *SciPost Phys. Codebases* **2022**, 8 (2022).
 - [26] X. Ju *et al.* (Exa.TrkX Collaboration), *Eur. Phys. J. C* **81**, 876 (2021).
 - [27] L. D. Landau and E. M. Lifschits, *The Classical Theory of Fields*, Volume 2 of Course of Theoretical Physics (Pergamon Press, Oxford, 1975), ISBN 978-0-08-018176-9.
 - [28] R. Shakarchi and Elias M. Stein, *Fourier Analysis: An Introduction (Princeton Lectures in Analysis I)* (Princeton University Press, Princeton, NJ, 2003).
 - [29] L. Hormander, *The Analysis of Linear Partial Differential Operators I* (Springer-Verlag, Berlin, 1990).
 - [30] M. Reed and B. Simon, *Methods of Modern Mathematical Physics: Functional Analysis I (Revised and enlarged ed.)* (Academic Press, San Diego, 1980).
 - [31] <https://github.com/LeviCondren/non-helical-tracks.git>.
 - [32] J. Duarte and J.-R. Vlimant, in *Artificial Intelligence for High Energy Physics* (World Scientific, Singapore, 2022), pp. 387–436.

- [33] G. DeZoort, P. W. Battaglia, C. Biscarat, and J.-R. Vlimant, *Nat. Rev. Phys.* **5**, 281 (2023).
- [34] S. Thais, P. Calafiura, G. Chachamis, G. DeZoort, J. Duarte, S. Ganguly, M. Kagan, D. Murnane, M. S. Neubauer, and K. Terao, [arXiv:2203.12852](#).
- [35] L. Heinrich, B. Huth, A. Salzburger, and T. Wettig, [arXiv:2401.16016](#).
- [36] L. H. Vaage, in *Connecting the Dots Workshop (CTD)* (Princeton, NJ, 2022).
- [37] T. Kortus, R. Keidel, N. R. Gauger *et al.* (B. pCT Collaboration), *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 15820 (2023).
- [38] S. Caillou, P. Calafiura, C. Rougier, J. Stark, D. T. Murnane, A. Vallier, X. Ju, and S. A. Farrell, CERN Technical Report No. ATL-COM-ITK-2022-057, 2022.
- [39] Y. Verma and S. Jena, [arXiv:2012.08515](#).
- [40] J. Hewes, A. Aurisano, G. Cerati, J. Kowalkowski, C. Lee, W.-k. Liao, A. Day, A. Agrawal, M. Spiropulu, J.-R. Vlimant *et al.*, in *EPJ Web of Conferences* (EDP Sciences, Les Ulis, France, 2021), Vol. 251, p. 03054.
- [41] F. Drielsma, K. Terao, L. Dominé, and D. H. Koh, [arXiv:2102.01033](#).
- [42] A. Akram and X. Ju, [arXiv:2208.12178](#).
- [43] X. Jia, X. Qin, T. Li, X. Huang, X. Zhang, N. Yin, Y. Zhang, and Y. Yuan, in *EPJ Web of Conferences* (EDP Sciences, Les Ulis, France, 2024), Vol. 295, p. 09006.
- [44] C. Biscarat, S. Caillou, C. Rougier, J. Stark, and J. Zahreddine, in *EPJ Web of Conferences* (EDP Sciences, Les Ulis, France, 2021), Vol. 251, p. 03047.
- [45] M. Zdybal, M. Kucharczyk, and M. Wolter, *Comput. Sci.* **25**, 25 (2024).
- [46] M. Andrews, B. Burkle, S. Chaudhari, D. Di Croce, S. Gleyzer, U. Heintz, M. Narain, M. Paulini, and E. Usai, in *EPJ Web of Conferences* (EDP Sciences, Les Ulis, France, 2021), Vol. 251, p. 03057.
- [47] G. DeZoort, S. Thais, J. Duarte, V. Razavimaleki, M. Atkinson, I. Ojalvo, M. Neubauer, and P. Elmer, *Comput. Software Big Sci.* **5**, 1 (2021).
- [48] A. Correia, F. I. Giasemis, N. Garroum, V. V. Gligorov, and B. Granado, *J. Instrum.* **19**, P12022 (2024).
- [49] 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.*, *Front. Big Data*, **5**, 828666 (2022).
- [50] M. Hansroul, H. Jeremie, and D. Savard, *Nucl. Instrum. Methods Phys. Res., Sect. A* **270**, 498 (1988).
- [51] J. Nelder and R. Mead, *Comput. J.* **7**, 308 (1965).
- [52] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, *Nat. Methods* **17**, 261 (2020).