The TrackML challenge

David Rousseau, Sabrina Amrouche, Paolo Calafiura, Steven Farrell, Cécile Germain, Vladimir Gligorov, Tobias Golling, Heather Gray, Isabelle Guyon, Mikhail Hushchyn, et al.

NIPS 2018 Competition proposal:
The TrackML challenge

David Rousseau *[1], Sabrina Amrouche[2], Paolo Calafiura[3], Steven Farrell[3], Cécile Germain[4], Vladimir Vava Gligorov[5], Tobias Golling[2], Heather Gray[3], Isabelle Guyon[4], Mikhail Hushchyn[6], Vincenzo Innocente[7], Moritz Kiehn[2], Andreas Salzburger[7], Andrey Ustyuzhanin[8], Jean-Roch Vlimant[9], and Yetkin Yilmaz[1]

[1]LAL, Orsay
[2]University of Geneva
[3]Lawrence Berkeley National Laboratory
[4]UPSud, INRIA, University Paris-Saclay
[5]LPNHE-Paris
[6]Yandex, MIPT
[7]CERN
[8]Yandex, HSE
[9]CalTech

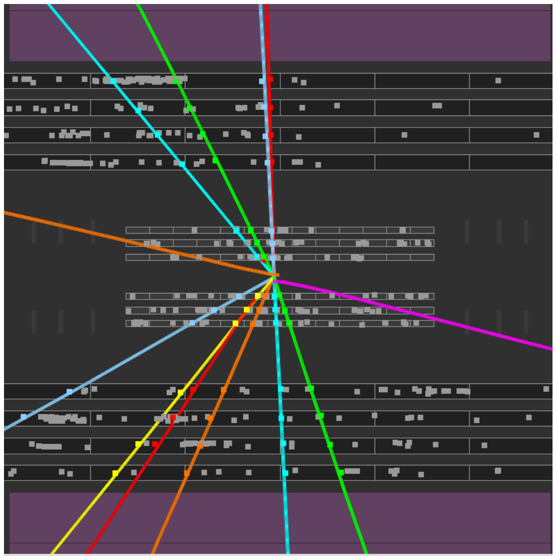
March 3, 2018

0.1 Overview of the competition

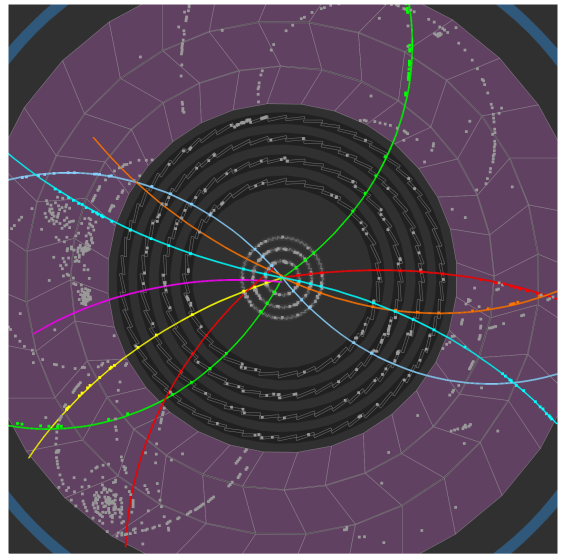
Challenges in machine learning are increasingly used not only to solve problems of economic of industrial interest or to advance machine learning and data science, but as a tool in basic and applied science research. Recently, several challenges in climate science, astrophysics, high energy physics, and chemistry have been organized as reported at the NIPS 2017 "challenges in machine learning" workshop[1] dedicated to the use of challenges as a research tool.

In this proposal we describe the second challenge in particle physics that we are organizing. Following the huge success of our first Kaggle challenge, the "Higgs Boson challenge" [1, 2], which attracted 1785 teams (an all time record as a Kaggle challenge at the time), we have been preparing a new challenge over the course of two years, with a team composed of machine learning scientists and particle physicist. The new problem chosen is of great importance to improve the quality of novel particle detection in the Large Hadron

*trackml.contact@gmail.com
[1]`http://ciml.chalearn.org`



(a)



(b)

Figure 1: **The TrackML problem.** Projection of particle tracks in the longitudinal and transverse planes, for low multiplicity events in the current detector (the challenge data will be 100 times more busy). The detector is made of concentric cylinders densely populated with silicon sensors. The dots correspond to the impact of particles on such sensors. The colored lines correspond to reconstructed trajectories. The trajectories are helices for charged particles due to the presence of a high magnetic field.

Collider at CERN. At the same time, the proposed challenge poses new and interesting machine learning problems that should attract the interest of the NIPS community.

We call our new challenge "TrackML", for **Tracking trajectories of particles with Machine Learning**. This refers to recognizing trajectories in 3D images of proton collisions obtained at the Large Hadron Collider (LHC) at CERN. Trajectories are particle "footprints": Each particle has its own distinctive type of trajectory. Unfortunately, all we have are "dots", corresponding to the impact of particles on silicon sensors organized in concentric cylinders. The problem is to "connect the dots", or rather the points, i.e. return all sets of points belonging to alleged particle trajectories. A schematic representation of the problem is shown in Figure 1. Think of this as the picture of a fireworks with some latency: the time information is lost because this is only a snapshot, but all particle trajectories have roughly the same origin and therefore there is a correspondence between arc length and time ordering.

From the Machine Learning point of view, the problem can be treated as a **latent variable problem** similar to a clustering, in which particle trajectory "memberships" must be inferred, or a **tracking problem** considering trajectories as time series, or a **pattern de-noising problem** considering that the dotted trajectories are noisy versions of continuous traces. Therefore this challenge offers an interesting new puzzle to the Computational Intelligence community, while addressing pressing needs of the Physics community.

Indeed, current methods employed for tracking particles in the LHC experiments at CERN will be soon outdated: By 2025, there will be a major upgrade of the LHC to fulfill its rich physics program: understanding the characteristics of the Higgs boson, searching for the elusive dark matter, or elucidating the dominance of matter over anti-matter in the observable Universe. The number of proton collisions will be increased 10-fold progressively until 2025 so that the number of particles per proton bunch collision will also increase from about 1000 to 10,000. In addition, the ATLAS and CMS experiments plan a 10-fold increase of the readout rate. The explosion in combinatorial complexity is mainly due to the increase of the probability of confusion between tracks. It will have to be dealt with with a flat budget at best. The projection of CPU computing power gain with the already highly optimized production software leaves at least a 10-fold gap.

The HEP (High Energy Physics) experiments have embraced Machine Learning, originally for supervised classification as a routine tool in the final analysis stage, and in the past few years for exploring more diverse applications. The preliminary attempts of applying Machine Learning to particle physics pattern recognition-tracking indicate a strong potential [3]. Considering the success of the Higgs Boson ML Challenge [1, 2], the HEP-ML collaboration for this challenge can be expected to produce high impact results. The algorithms exposed during the challenge, if promising, will be reused within the LHC experiments.

The proposed challenge is the culmination of a series of events in which we have been ramping up the sophistication of our evaluation system and the difficulty of the problem. First we organized a one day hackathon[4] limited to two-dimensional problems (similarly to Figure 1-a). We are opening up this month a first round of competition with result submission, not imposing to participants any computional constraints (which has been accepted a Kaggle competition and is part of the official selection of the WCCI conference). Finally, the present proposal is to run as a NIPS challenge the final round with code submission, in which realistic computational constraints will be given to the participants.

Although solving the proposed problem will be highly useful to the physics community, no knowledge of physics will be required to participate.

0.2 Keywords

Machine learning, time series prediction, clustering, tracking, latent variable problems

0.3 Novelty

The problem of tracking is pervasive in many application domains and is common in video processing. Several challenges in computer vision involving tracking have been orgenized[2]. However, the setting of such challenges is very different, and to our knowledge, **no competition about particle tracking in physics has ever been run** (outside of our program, which includes the two first rounds previously mentioned). It is worth noting that the problem of particle tracking from still images is quite different from the problem of tracking from video data. The high speed of particles does not allow us to sample trajectories, hence reconstituting the lost time information is part of the problem.

This challenge inherits from the experience we have with orgnizing the Higgs Boson challenge[1, 2] and the Flavor of Physics challenge[3](representatives of the organizers of both challenges are in our committee). However, these two challenges were on topics quite different from the new proposed challenge: the first one asked participants to single out novel "signal" events from rather similar looking "background" events, and the second asked them to compensate for systematic bias introduced by simulators of events.

We are not aware of other challenge efforts in high energy physics.

1 Competition description

1.1 Background and impact

Importance of the problem and anticipated impact

Our challenge program inserts itself in a bigger effort of the Atlas collaboration (one of the three experiments analyzing data collected at CERN on the Large Hadron Collider– LHC) to use Machine Learning to assist high energy physicists in discovering and characterizing new particles.

In the LHC, proton bunches (beams) circulates and collide at high energy. Each beam-beam collision (further called an *event*) produces a firework of new particles (figure 2). To identify the types and measure the kinematic properties of these particles, a complex apparatus, the detector records the small energy deposited by the particles when they impact well-defined locations in the detector.

The tracking problem refers to reconstructing the trajectories of the particles from the information recorded by the detector. The augmentation of data throughput creates a major scaling bottleneck for the associated pattern recognition-tracking task. Thus, current methods [5] will soon become obsolete and there is an urgent need for novel algorithms.

[2]Tracking competitions in ISMAR 2017 `https://sites.google.com/view/ismartc2017`, Visual Object Tracking `http://www.votchallenge.net/vot2017/`, Multi-camera object tracking `http://mct.idealtest.org/`

[3]`https://www.kaggle.com/c/flavours-of-physics`

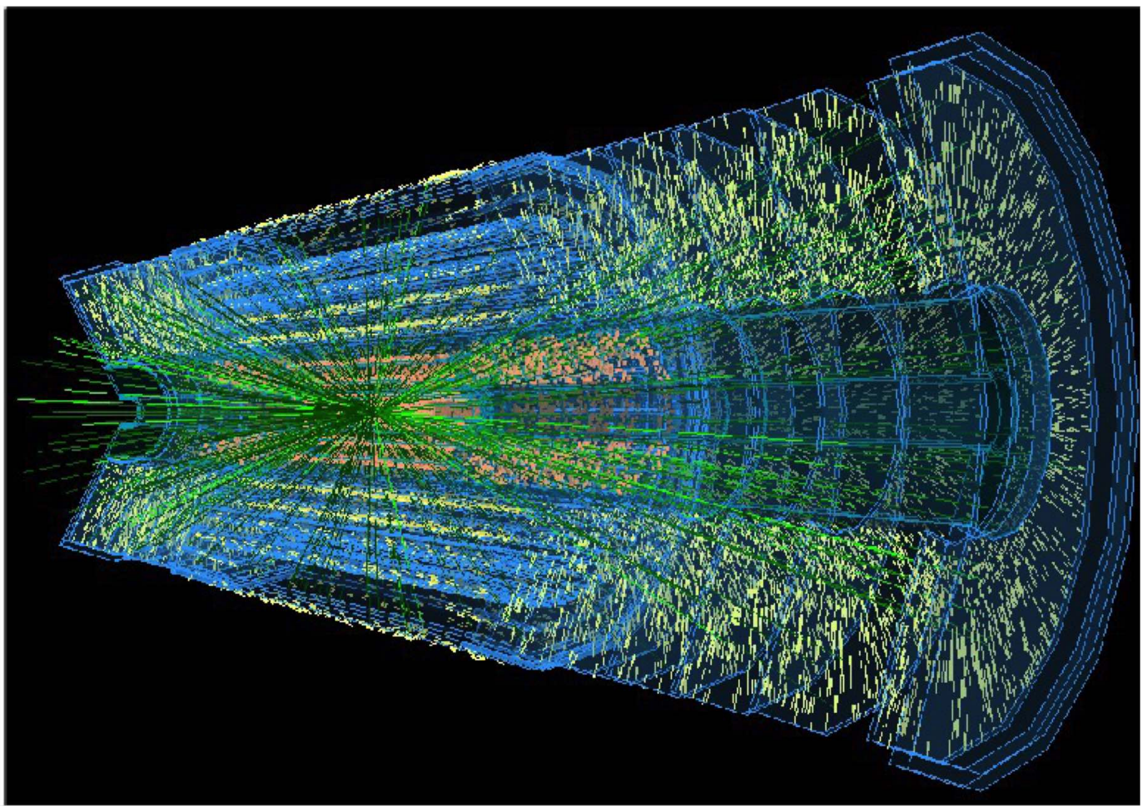


Figure 2: An etched-out high-multiplicity collision image in the future detector, measurements are in yellow, trajectories are green.

The whole chain of analysis of particles relies on tracking, hence to a large extent, the success of the whole endeavor rests upon algorithmic advances on tracking.

Relevance of the problem to the NIPS audience

Although at first sight, recovering particle trajectories may look like an optimization problem, in fact, it is not treated as such by methods currently deployed. It is treated as a **pattern recognition** problem, after adequate transformations like th Hough transform, or as a **tracking problem** using Kalman filters (see [5] for a review). This gives several promising avenues to the machine learning community.

We envision that the problem could be treated by machine learning scientists in at least three different ways:

- **A latent variable problem:** A data generating process first drew at random particles with given characteristics (momentum, charge, mass), then drew points along a trajectory originating near a collision focus (with uncertainties including diffusion/scattering and imperfections of the detector sensors). "Particle memberships" are the latent variables to be inferred. This is similar to a clustering problem.

- **A tracking problem:** Since all particle trajectories originate roughly from the same collision point and follow roughly linear of circular trajectories, using the correspondence between arc length and time ordering, one can treat the trajectories as time series and use tracking techniques.

- **A pattern de-noising problem:** Considering the collision snapshot as a 3D image, through the data acquisition process, the original trajectory lines were degraded into dotted lines with just a dozen points per line (the human eye cannot see the lines); the problem can therefore be thought of as signal enhancement of an "in-painting" problem (filling in missing data).

Therefore this challenge should appeal to a large fraction of the NIPS community working either on **unsupervised learning** (latent variable models, clustering), or on **recurrent networks** (such as LSTMs, which can suitably enhance and replace Kalman filters, as already demonstrated in the one-day hackathon we organized), or on **convolutional auto-encoders** (as a means of denoising trajectories). We also hope that other ideas we have not anticipated will emerge.

Application advances

A major upgrade of the LHC is planned for 2025. The specific interest of this competition is coping with the associated explosion in combinatorial complexity of the tracking task.

With the upgrade, the number of particles per proton bunch collision will go from about 1000 to 10.000, and the readout rate also incur a ten-fold increase. The tracking step, which is the bottleneck of reconstruction, must sustain this data rate. The present methods, based on combinatorial iterations combined with Kalman filters, are already highly optimized but they scale poorly: each event is analyzed in roughly 100 seconds CPU time on one processor core, while we are aiming at bringing this down to less than 10 seconds. With the expected increase in the number of cores, this will ensure to sustain the required throughput for the LHC upgrade. We are hoping the Machine Learning can bring us there.

1.2 Data

We used the fast (10s per event) and accurate simulation engine ACTS[4] [6] to generate the challenge data. It allowed us to generate **realistic data** emulating a full Silicon LHC detector (see Fig 3), while providing us with the **ground truth of particle trajectory membership**. Thus, for each event we obtained the “detected” 3D points coordinates (and additional features), and, as ground truth, the list of points associated to each track. There is a one to one relationship between the true 3D points and the reconstructed ones.

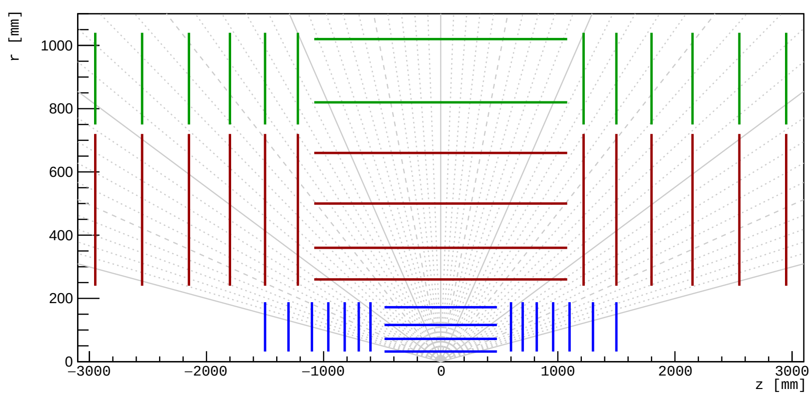


Figure 3: Simplified illustration of the detector, which has revolution geometry : cylinders appear as horizontal lines and disks as vertical lines. The different detector technologies (with different intrinsic precision) are shown with different colors. Innermost pixel detector (blue), followed by a short (red) and long strip (green) detector.

Realistic collisions yielding 10.000 tracks per event have been simulated with a sufficient level of details to make the task almost as difficult as for real events: points are measured with a precision from 10 to 100 microns, some tracks are grouped in dense ”jets” (increasing

[4]https://gitlab.cern.ch/acts/acts-core

the possibility of confusion), multiple scattering distorts the tracks, points are some times missing, some tracks stop early.

The data set is large in order to allow the training of data intensive methods : $\sim 10^4$ events, with each $\sim 10^4$ tracks, for a total dataset size of 100 GBytes. The events are independent and equivalent.

The public and private evaluation datasets need to be much smaller, for feasibility reasons, about 100 events (1 GBytes), but are large enough to evaluate the metrics within a per mille of statistical uncertainty.

The dataset has been generated for the purpose of the challenge, and can be publicly released; all copyrights and privacy of experimental data and software have been respected.

Although the simulation software is public, its configuration for producing the challenge dataset is private and only a few people have access to the ground truth of the test dataset.

1.3 Tasks and application scenarios

For the purpose of this proposal, we give some details on the actual experimental setting. However, the participants will not need to understand the physics of the problem and will be able to treat is as an abstract machine learning problem. Because a fraction of the participants will actually be motivated by contributing a solution to a real problem, we will provide a document explaining in some details the application, for educational purposes and to satisfy the curiosity of the participants.

The task

From an abstract point of view, the detector is simply an apparatus that records the impacts, called the *hits*, of the particles traversing the detector. The hits locations are a list of 3D coordinates. For each particle, the number of hits is on average 12, but as low as 1 and up to 20 . Each point is a 3D measurement in euclidean coordinates (x, y, z), with some non isotropic measurement error. The participants should associate the 3D points together to form tracks.

The tracks are slightly distorted arc of helices with axes parallel to a fixed direction, and pointing approximately to the interaction center. On figure 1, the arcs appear as lines on the longitudinal projection and circles on the transverse one. Robustness with respect to these distortions and approximate pointing are enforced by the metric and are a de facto requirement.

While the task can be formally stated as a clustering problem, the ratio between the number of clusters ($\sim$ 10K) and their size ($\sim$ 10 points), is highly unusual, and drastically limits the performance of off the shelf clustering algorithms. Typically, at least 90% of the true tracks should be recovered.

Scientific context

The challenge focuses on tracking in an original context. Tracking is an important subfield of computer vision [7, 8]. In computer vision, the usual goal is to predict future position of multiple moving objects based on their previous positions, with numerous applications such as video surveillance, vehicle navigation, and autonomous robot navigation. These applications focus on identifying a few target objects in complex environments.

The challenge shares the basic setting of the classical computer vision problem: reconstructing a trajectory based on low-level data with no metadata. However, it departs from classical tracking on two majors features: the considerable multiplicity of objects to track, in the order of 10^4, while the objects as much simpler, in the order of a ten of points; and the fact that there is no hierarchy of objects: all, or at least most of, the points must be associated with a track.

The problem relates to representation learning [9] as in [10], to combinatorial optimization as in [11], neural-network based clustering [12], and even to time series prediction [13] (even though the time information is lost, it can safely be assumed that particles were coming from the center of the detector and have successively crossed the nested layers of the detector).

A possible approach is to efficiently exploit the a priori knowledge about geometrical constraints [14]. Indeed, trajectories are close to segments of helices, as shown in figure 1. The generative approach [15], [16], in particular with introduction of supervision in variational autoencoders [17, 18], as well as the discriminative approaches [19] could be exploited for combining structural priors and nonlinear state estimation with deep neural networks.

In physics, the field of particle tracking is well developed with a specialized conference [20, 21]. While early methods included mathematical transformations such as the Hough transform, the methods offering the best speed/accuracy tradeoff have concentrated on variants of Kalman filters in recent years, combined with various local pattern recognition methods. For an in depth review of the pre-Machine Learning state of the art, see [5].

The preliminary attempts of applying Machine Learning to particle physics pattern recognition-tracking indicate a strong potential [3]. [4] analyzes a simplified and smaller 2D version of the problem. Several promising machine learning and neural network solutions have emerged, including LSTM (Long Short-Term Memory) [22, 23]. Optimization methods such as MCTS (Monte Carlo Tree Search) were also successfully used.

Related applications

Although this problem is relatively specific of particle physics, it is not limited to our particular setting of proton collisions in the LHC. Solutions to this problem can generalize to other particle tracking in Physics.

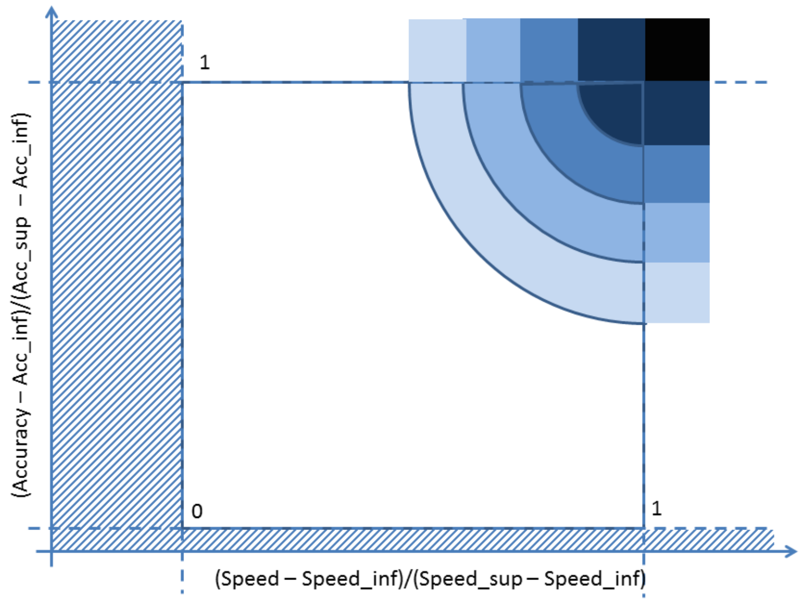


Figure 4: **Multi-objective metric.** We represent as shaded areas of decreasing intensity zones of decreasing scores according to our evaluation scheme, which considers the tradeoff speed/accuracy. The hashed areas represent zone of algorithm rejection.

1.4 Metrics

Our challenge is to obtain at least as good track reconstruction accuracy as existing algorithms while considerably improving speed. To evaluate the participants' entries we therefore devised a multi-objective metric balancing speed and accuracy. We first describe the multi-objective metric, then explain how we evaluate accuracy.

Multi-objective metric

To constrain the participants in a meaningful way, the organizers will calibrate the evaluation protocol with two baseline methods:

- **The "Accuracy" baseline**: the most accurate tracking method available to us at the start of the code submission round, regardless of how computationally expensive it is. Its performance will provide us with accuracy Acc_sup and speed Speed_inf).
- **The "Throughput" baseline**: A "quick and dirty" method providing accuracy Acc_inf and speed Speed_sup.

Inferior entries with accuracy lower then Acc_inf or speed lower than Speed_inf will not be considered. The challenge is to get the best compromise between speed and accuracy.

We schematically represent in Figure 4 our multi-objective criterion. The Accuracy metric is defined in the next section. If we define $Acc = (Accuracy - Acc\_inf)/(Acc\_sup - Acc\_inf)$ and $Spe = (Speed - Speed\_inf)/(Speed\_sup - Speed\_inf)$, our ranking criterion will be:

$$\max(Acc, Spe) \quad \texttt{if} \quad Acc > 1 \;\; \texttt{or} \;\; Spe > 1 \tag{1}$$

$$1 - \sqrt{(1 - Acc)^2 + (1 - Spe)^2} \quad \texttt{otherwise} \tag{2}$$

Accuracy metric

Although many clustering metrics could potentially be used, because of the direct potential applicability of the algorithms of the winners, the metric chosen stick closely to the requirements of the physics community. We describe is briefly. Other standard clustering metrics will be used for comparison and analyses in post-challenge evaluations.

A perfect algorithm will uniquely and correctly associate each point to the track it belongs to. An imperfect algorithm will miss some tracks altogether, miss one or more points for an otherwise valid track, associate wrong points to an otherwise valid track, find tracks from random association of points, find multiple variants of the same track.

Because the data come from simulation, we know which particle created each hit (point), in other words the ground truth. For brevity, we note R-tracks the proposed solutions and T-tracks the ground truth. A point must belong to at most one R-track, but it is not required to list all points. The score is defined as follows.

- R-Tracks with 3 points or less have a zero score, as they do not allow to compute any meaningful physical quantity in further analysis.
- R-tracks and T-tracks are uniquely matched by the combination of the following rules.
  - For each R-Track, the matching T-track is the one to which the majority of the R-track points belong; if there is no such particle, the score for this track is zero.
  - The R-Track should have the majority of the points of the matching T-track, otherwise the score of this track is zero

  These two requirements guaranty a one to one relation M between all remaining R-tracks and T-tracks.
- The score of a R-track r is the weighted count of the points in the intersection $r \cap M(r)$.
- The score of the event is the sum of the scores of the R-tracks, normalized by the sum of the weights of all points. This actually normalizes the score to the $[0, 1]$ range, with 1 being the score of a perfect reconstruction.

- Finally, the overall score is the average over the 100 test events. We have evaluated the statistical uncertainty to be less than 10^{-3}.

The weights are incentives to get physically meaningful reconstructions, along two directions: the weight of a point is the product of two independent quantities *weight_order* and *weight_pt*.

- *weight_order* The points at the beginning of the track, close to the collision, and at the end, are more important than the ones in the middle. The weights reflect this hierarchy and are normalized so that the sum of weights for one T-track is 1.

- *weight_pt* The high energy particles (large transverse momentum p_T) are the most interesting ones. As the bulk of the tracks have low p_T, we have to explicitly favor high p_T. *weight_pt* is 0.2 if $p_T < 0.5$GeV and 1. for $p_T > 3$GeV, with a linear interpolation in between. Note that the lower the p_T, the larger the geometrical curvature; large p_T tracks appear as straight lines.

- Particles which generates 3 hits (points) or less are considered spurious, the weights of the associated points are set to zero.

As the weights disclose important information, they are provided along with the points in the training data, but will be kept hidden for the test data.

It is a combination of the Jaccard version of counting pairs [24] and set matching [25]. With set matching, it shares the one-to-one assignment of reconstructed clusters (R-Tracks) to true clusters (T-tracks). However, thanks to the majority rule, it does not suffer from the "problem of matching" [25]. With respect to counting pairs, the Jaccard index is more appropriate than the Rand index [26], as the result of the later would be dominated by the true negatives (pair of points that agree to be in different clusters), which are not taken into account in the Jaccard counting points index.

1.5 Baselines and code available

The participants will be provided with sample code providing an example of code submission, following the prescribed API, which will be thoroughly documented. They will also be given a Jupyter-notebook loading sample data, providing some data visualization, and calling the sample code.

At this stage, we already have a Jupyter-notebook ready for the preliminary code submission round. We will revise it as needed using the feed-back of the participants. For this first round, we provide two baselines (and codes) (see their score on Fig. 5, which can be seen to degrade when the track multiplicity increases, calling for more sophisticated methods).

- DBScan: this clustering algorithm demonstrates non-trivial performance, although far from the requested ones. The main goal is to provide in a few lines a method to demonstrate the workflow.

- Hough transform; where the 3D hit space is mapped onto a track parameter space, where maxima (corresponding to tracks) are found, and then moved back to the original 3D space to associate the points. This technique has a linear complexity; however it does not allow to reach the maximum efficiency.

At the end of the result submission round, we will decide on the Accuracy and Throughput baselines to be used in the code submission round to calibrate the multi-objective metric (Equations 1 and 2). At this stage, we think that "DBScan" could be a good candidate for the Throughput baseline because it is very fast (with a mediocre accuracy). The Accuracy baseline will be the winning method of the first round or the method of tracking expert physicists, whichever is best.

1.6 Tutorial and documentation

The documentation will be available on the Kaggle website[5]. Its goal is to make the challenge fully unbiased with respect to physics knowledge. A white paper in preparation[6] will be released.

2 Organizational aspects

2.1 Protocol

We propose to organize for NIPS 2018 the final round of our TrackML competition program. We started with a one day hackathon a year ago to probe feasibility and interest, we are this month opening up a competition with result submission only to calibrate task difficulty, and for NIPS we want to organize a **full fledged competition with code submission**.

Platform

Kaggle[7] has offered to host both the result submission round and the final code submission round. Kaggle is a world leader in challenge organization and has recently enhanced their platform to allow participants to submit code, as exemplified by the NIPS 2017 challenge on adversarial attacks and defenses.

Phases (within the final round organized for NIPS 2018)

The competition will take place is two phases: a development phase and a test phase.

[5] `https://www.kaggle.com/c/trackml-particle-identification/host` (private at the time of writing)
[6] `https://www.overleaf.com/13743218dgmhcdhtyhsg#/53192542/`
[7] `http://kaggle.com`

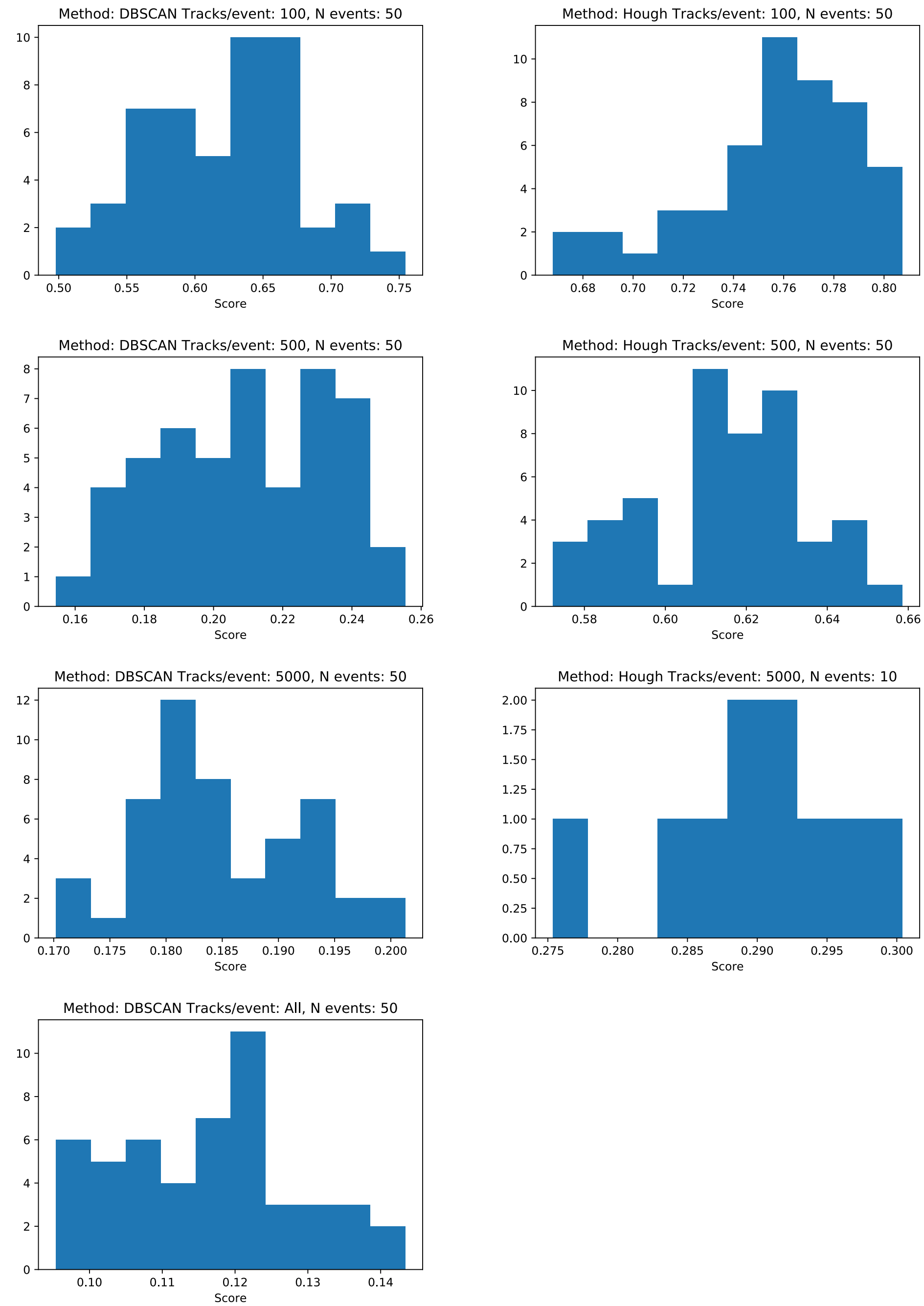


Figure 5: **Baseline scores** Score obtained from DBscan baseline (left) and Hough baseline (right). The number of tracks considered increases from top to bottom ("All" refer to the complete event which correspond to about 10000 tracks); the score is normalised so that the score knowing the ground truth would be 1.

- **Development phase:** During the development phase the participants will have access to "public" data consisting solely of **labeled training data**, which they will be able to use to develop their method. The code that they will submit to the platform will be exposed to **development unlabeled test data**. The resulting prediction score will be revealed on a "public" leaderboard, for immediate feed-back.
- **Final test phase:** During the final test phase the participants will only be allowed to make one final code submission. Their submitted code will be exposed to **final unlabeled test data**. Their prediction score will be kept secret on a "private" leaderboard and used for the final ranking, which will be revealed only at the end of the challenge.

Code execution

The participants will have to deliver algorithms, which can be executed within execution time constraints to demonstrate the practical viability of their solution. The Kaggle competition platform will make possible to evaluate all code submissions in a well defined and reproducible computational environment with CPU virtual machines emulating computation nodes of the supercomputing center at CERN (which are standard CPU cores). Docker images will be used to encapsulate the software environment needed to run the evaluation and will be both made available to the participants and used on the evaluation platform. Sample code will be provided to facilitate following a prescribed API, which will be published and documented.

The participants will be limited to two submissions per day during the development phase and only one submission during the final phase. They will be allowed to submit docker images including all necessary libraries to run their code, as long as their code respects the API.

During post-challenge evaluations, we will benchmark the code of top-ranking participants in other computational environments, and may encourage them to modify their code to take advantage of GPUs. We will motivate them to collaborate with us by proposing to work on a joint publication.

Beta testing, cheating

By the time the NIPS TrackML round with code submission starts, the competition with result submission will have ended and the soundness of the task and data thoroughly validated. We will still have time (a few weeks) to make adjustments to the difficulty of the task and to the baselines, if necessary. This is the benefit of ramping up difficulty by running first an "accuracy only" competition round.

We anticipate that cheating will not be an issue in this challenge. Uniqueness of accounts (to prevent exceeding the maximum number of submissions per day) is well controlled in the Kaggle platform. Gaining access to truth values will be very difficult since only a few trustworthy people have access to them and treat the data will a great level of care. On the platform, the code of the participants and the scoring program are executed in tight compartments, preventing the code of the participants to gain access to the truth values.

2.2 Rules

The rules will respect the general policy of the Kaggle platform and the privacy conditions of CERN.

In addition, we will enforce the following rules specific to this challenge:

- **Registration and announcements:** Participants may register as teams of individuals. There is no limitation in the number of team members. Team are mutually exclusive. To receive announcements and be informed of any change in rules, the participants must provide a valid email. They may otherwise choose a pseudonym and remain anonymous during the challenge. Teams are forbidden to use multiple accounts.

- **Conditions of participation:** Participation requires complying with the rules of the challenge. The organizers, sponsors, their students, close family members (parents, sibling, spouse or children) and household members, as well as any person having had access to the truth values or to any information about the data or the challenge design giving him (or her) an unfair advantage, are excluded from participation. A disqualified person may submit one or several entries in the challenge and request to have them evaluated, provided that they notify the organizers of their conflict of interest. If a disqualified person submits an entry, this entry will not be part of the final ranking and does not qualify for prizes. The participants should be aware that the organizers reserve the right to evaluate for scientific purposes any entry made in the challenge, whether or not it qualifies for prizes.

- **Submission method:** The results must be submitted through the Kaggle competition site. The participants can make up to 2 submissions per day in the development phase and a single submission (in total) in the final phase. The entries must be formatted following the instructions. The organizers decline any responsibility for entries which fail to execute.

- **Ranking:** Ranking will be performed by the performance metric described on the challenge website. In case of ties, the first submitted entry will prevail. The final ranking determining the prizes will be performed according to the private leaderboard scores. The top three participants will be declared winners and may be eligible for prizes (see prize claiming).

- **Prize claiming:** The participants must disclose their real identity to the organizers to claim any prize they might win. If a participant provides his real name, it will be publicly disclosed at the competition workshop. In addition, the winners who claim their prize will have to make their code publicly available under en open-source OSI-approved license and fill out fact sheets describing their method. Both requirements will have to be completed within a week of the deadline for submitting the final code. Non winners or entrants who decline their prize retain all their rights on their entries and are not obliged to publicly release their code. However, Kaggle has specific conditions on their use of entries that must be reviewed and accepted by the participants and are out of the control of the organizers.

- **Dissemination:** The participants will be invited to attend a workshop organized in conjunction with a major machine learning conference and contribute to the proceedings (NIPS if the competition is selected as part of the NIPS 2018competition program.)

- **Travel awards:** Travel awards may given (depending on fund availability) to attend the workshop organized in conjunction with the challenge. The award money will be granted in reimbursement of expenses including airfare, ground transportation, hotel, or workshop registration. Reimbursement is conditioned on (i) attending the workshop, (ii) making an oral presentation of the methods used in the challenge, and (iii) presenting original receipts and boarding passes. The reimbursements will be made after the workshop.

- **Conflicts:** Conflicts will be resolved amicably by writing to the challenge committee chair person David Rousseau <rousseau@lal.in2p3.fr> and reviewed by the organizing committee and Kaggle whose decision will be binding and final.

2.3 Schedule

We are under no time pressure. We have devised a schedule, which will leave plenty of time to participants to perform in the challenge and to the organizers to assess the results.

- **2016-2017: challenge design.** This challenge took a lot of preparation time. We already ran a small size hackathon event on toy data a year ago to probe interest and assess feasibility.

- **January-February 2018: dataset finalization.** The data are already prepared and the preliminary “result submission” round is scheduled to start by the end of February and last for 40 days.

- **June 1, 2018: Start of competition** The NIPS code submission final competition round will be launched mid June 2018, and will run until September 2018.

- **September 15, 2018: End of competition.**

- **September 16-September 30, 2018: Verification and examination of the submissions.** The organizers will review submissions and the winners will be announced early October to give them the opportunity to register to NIPS and make travel plans.

- **October-November 2018: Post challenge analyses.** The winners will be requested to fill our fact sheets and open-source their code. They will be invited to submit extended abstracts to the workshop. The organizers wll run systematic post-challenge benchmarks to compare metrics and computational platforms.

- **December 2018: NIPS workshop.** If accepted at NIPS, the results of the challenge will be revealed in December at NIPS.

- **Spring 2019: CERN workshop.** A final workshop will be organized at CERN in spring 2019, where winners of both rounds of the challenge will be invited.

2.4 Competition promotion

The team organising this competition has experience with the HiggsML challenge [1] in 2014 on Kaggle `https://www.kaggle.com/c/Higgs-boson`, which, with close to 2000 participants, was the most successful Kaggle competition at the time, as well as with the Flavour of Physics challenge in 2015 (`https://www.kaggle.com/c/flavours-of-physics`), with close to 800 participants. Furthermore, many collaboration and workshops between Machine Learning and High Energy Physics have taken place meanwhile (for example the DS@HEP series, with first edition at CERN in October 2016 `https://www.nature.com/news/artificial-intelligence-called-in-to-tackle-lhc-data-deluge-1.18922` or Hammers and Nails at Weizmann in 2017 `https://www.weizmann.ac.il/conferences/SRitp/Summer2017/hammers-and-nails-machine-learning-and-hep`), many of them co-organised by members of the team. It is foreseen to advertise the competition through these channels, and in addition relay the promotion of the challenge through CERN and LHC experiments social media, with hundred of thousands followers. The call for participation will also be distributed on machine learning and data science challenge mailing lists, including that of Kaggle (tens of thousands of subscribers). In addition, several team members are former challenge participants and organizers with their own mailing list and contacts. We also count on the snow ball effect of the first round (with result submission only). So, in spite of the difficulty of the challenge and the fact that it addresses a problem new to the machine learning community, we expect several hundred participants.

2.5 Organizing team

The team includes many particle physics tracking experts working on the ATLAS (SA, PC, SF, TG, HG, MK, DR, AS), CMS (VI, JRV) and LHCb (VG, AU) experiments at CERN who have been working on providing quality data and baselines, and several machine learning specialists (CG,IG,AU,MH) who have been advising on how to cast the problem as a machine learning problem and design the challenge protocol. The roles of the individual team members are as follows:

Coordinator	David Rousseau
Data provider	Andreas Salzburger and Moritz Kiehn
Platform administrators	Kaggle. Interaction with the organizers will go through David Rousseau
Baseline methods providers	Sabrina Amrouche and Mikhail Hushchyn
Beta testers	Paolo Calafiura (leader)
Evaluators	The co-organizers and the scientific advisory committee, with David Rousseau on the HEP side and Isabelle Guyon on the ML side as co-leaders.

David Rousseau, senior physicist at LAL-Orsay, developed and coordinated software (including tracking software) for the ATLAS experiment from 2000 and 2012, co-organized HiggsML challenge in 2014, currently co-coordinator of the ATLAS Machine Learning group.

Sabrina Amrouche, PhD student at University of Geneva, data scientist in pervasive computing, currently working on data evaluation and reference solutions involving machine learning models for track reconstruction.

Paolo Calafiura, Computer Scientist at Lawrence Berkeley National Lab, US ATLAS Computing and Software Manager, Principal Investigator of the HEP.TrkX project that investigates novel algorithmic paradigms for particle tracking at the High Luminosity LHC.

Steven Farrell, postdoctoral fellow in physics at Lawrence Berkeley National Lab and software developer for ATLAS simulation and analysis code, currently developing machine learning solutions for HEP as part of the HEP.TrkX project.

Cécile Germain, professor of Computer Science at University Paris Sud ; co-organized HiggsML challenge in 2014 ; she is working on supervised and unsupervised machine learning applications to modeling and optimizing complex systems.

Vladimir Vava Gligorov, scientist at LPNHE-Paris, the deputy physics coordinator of the LHCb experiment and formerly in charge of its real-time data processing. He proposed and coauthored the first BDT to be used for real-time data processing in LHCb, and subsequently coordinated the introduction of machine learning techniques throughout the real-time reconstruction and analysis of LHCb data.

Tobias Golling, physicist, associate professor at University of Geneva, experienced with tracking in ATLAS since 2005 and developing machine learning techniques in particle physics.

Heather Gray, physicist, scientist Lawrence Berkeley National Laboratory, detailed studies and responsibilities in ATLAS tracking, improved the performance of ATLAS clustering algorithms in dense environments by introducing a neural-network based algorithm. Exploits machine learning techniques in Higgs analyses.

Isabelle Guyon, professor of informatics at UPSud Paris-Saclay and machine learning researcher. She has extensive experience with organizing machine learning challenges.

Mikhail Hushchyn, PhD student at Moscow Institute of Physics and Technology, data scientist at Yandex-CERN research group. Currently working on track pattern recognition for the SHiP experiment at CERN.

Vincenzo Innocente, Principal Software Scientist at CERN. Responsible for simulation and reconstruction software in three generations of collider experiments at CERN. Currently leader of the Tracking Physics Group in the CMS experiment at the LHC. In machine learning since 1990.

Moritz Kiehn, post-doc at University of Geneva, ACTS developer, also involved in the Silicon hardware R&D

Andreas Salzburger, physicist at CERN, ACTS project coordinator, co-leader of the design team of the ATLAS Inner Tracker for the high luminosity LHC upgrade, former ATLAS tracking software group convener and reconstruction software group convener.

Andrey Ustyuzhanin, head of Yandex-CERN research group focused on applying Machine Learning to solving Physics problems, organizer of "Flavour of Physics" challenge on Kaggle, head of laboratory at Higher School of Economics, Russia.

Jean-Roch Vlimant, Associate physicist at the California Institute of Technology. Former CMS tracking software coordinator and reconstruction project coordinator. Advisor and organizer of several HEP-ML events. Principal investigator of several super-computing allocations for deep learning projects. Member of the HEP.TrkX project that investigates novel algorithmic paradigms for particle tracking at the High Luminosity LHC.

Yetkin Yilmaz, physicist, post-doc research assistant in the Applied Statistics and Machine Learning Group of LAL. Developed the code for the TrackMLRamp hackathon at CTD/WIT 2017 workshop in Orsay.

3 Resources

Kaggle will sponsor the challenge monetary prizes (25k$ at the time of writing). To cover invitations of winners to NIPS and to CERN workshop, academic sponsorship from the organisers' institutions is being gathered reaching already 10k at the time of writing. Application for industry sponsorship have been or are being submitted in parallel to Azure and Amazon WS, as well as to partners of CERN Openlab Intel, Nvidia and IBM. This additional sponsorship will allow us to propose significant prize money, and to sponsor training resources for participants. The organisers' institutions computing resources is sufficient for all necessary activities: data set generation and detailed analysis of participants submission. The technical support of the competition platform will be done by Kaggle, while the organisers will have organised rotating shifts to monitor the leaderboard and the forum.

References

[1] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau, "The Higgs boson machine learning challenge," in *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, ser. JMLR: Workshop and Conference Proceedings, vol. 42, Montreal, Canada, 2014. [Online]. Available: http://proceedings.mlr.press/v42/cowa14.html

[2] ——, "How Machine Learning won the Higgs Boson Challenge," in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, Belgium, Apr. 2016. [Online]. Available: https://hal.inria.fr/hal-01423097

[3] S. Farrell *et al.*, "The HEP.TrkX Project: deep neural networks for HL-LHC online and offline tracking," *EPJ Web Conf.*, vol. 150, p. 00003, 2017.

[4] S. Amrouche *et al.*, "Track reconstruction at LHC as a collaborative data challenge use case with RAMP," *EPJ Web Conf.*, vol. 150, p. 00015, 2017.

[5] T. Cornelissen, M. Elsing, S. Fleischmann, W. Liebig, and E. Moyse, "Concepts, Design and Implementation of the ATLAS New Tracking (NEWT), ATL-SOFT-PUB-2007-007," 2007.

[6] C. Gumpert, A. Salzburger, M. Kiehn, J. Hrdinka, and N. Calace, "ACTS: from ATLAS software towards a common track reconstruction software," *J. Phys. Conf. Ser.*, vol. 898, no. 4, p. 042011, 2017.

[7] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, Dec. 2006.

[8] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, pp. 58:1–58:48, Oct. 2013.

[9] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[10] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *26th NIPS*, 2013, pp. 809–817.

[11] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," 2016. [Online]. Available: https://arxiv.org/abs/1611.09940

[12] G. Aad *et al.*, "A neural network clustering algorithm for the ATLAS silicon pixel detector," *Journal of Instrumentation*, vol. 9, no. 09, 2014.

[13] P. Ondruska and I. Posner, "Deep tracking: Seeing beyond seeing using recurrent neural networks," in *13th AAAI Conference on Artificial Intelligence*, 2016, pp. 3361–3368.

[14] R. Urtasun, "Incorporating structure in deep learning," in *4th ICLR*, 2016, keynote. [Online]. Available: http://videolectures.net/iclr2016_urtasun_incoporating_structure/

[15] R. G. Krishnan, U. Shalit, and D. Sontag, "Deep kalman filters," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1511.05121

[16] M. Johnson *et al.*, "Composing graphical models with neural networks for structured representations and fast inference," in *29th NIPS*, 2016.

[17] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 3581–3589.

[18] S. Narayanaswamy, B. Paige, J. van de Meent, A. Desmaison, N. D. Goodman, P. Kohli, F. D. Wood, and P. H. S. Torr, "Learning disentangled representations with semi-supervised deep generative models," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5927–5937.

[19] T. Haarnoja *et al.*, "Backprop kf: Learning discriminative deterministic state estimators," in *29th NIPS*, 2016, pp. 4376–4384.

[20] R. Fruhwirth, E. Brondolin, B. Kolbinger, and W. Waltenberger, Eds., *Proceedings, Connecting The Dots 2016*, vol. 127, 2016. [Online]. Available: https://www.epj-conferences.org/articles/epjconf/abs/2016/22/contents/contents.html

[21] C. Germain, H. Grasland, A. Lounis, D. Rousseau, and D. Varouchas, Eds., *Proceedings, Connecting The Dots / Intelligent Tracker (CTD/WIT 2017)*, vol. 150, 2017. [Online]. Available: https://www.epj-conferences.org/articles/epjconf/abs/2017/19/contents/contents.html

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[23] Z. C. Lipton, "A critical review of recurrent neural networks for sequence learning," *CoRR*, vol. abs/1506.00019, 2015.

[24] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data," in *Pacific Symposium on Biocomputing*, 2002, pp. pp. 6–17.

[25] M. Meilă, "Comparing clusterings—an information based distance," *J. Multivar. Anal.*, vol. 98, no. 5, pp. 873–895, 2007.

[26] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.