Simultaneous multi-vertex reconstruction with a minimum-cost lifted multicut graph partitioning algorithm

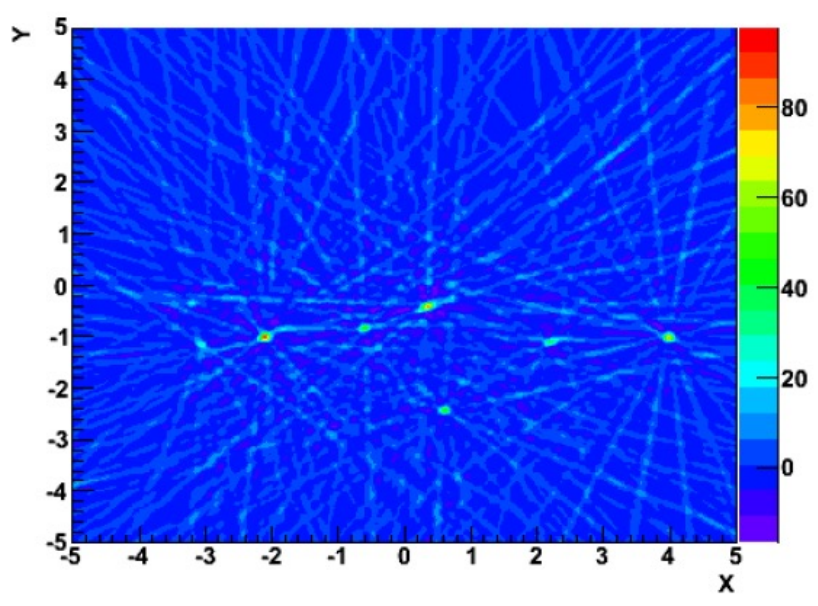
V. Kostyukhin
Siegen university

Introduction

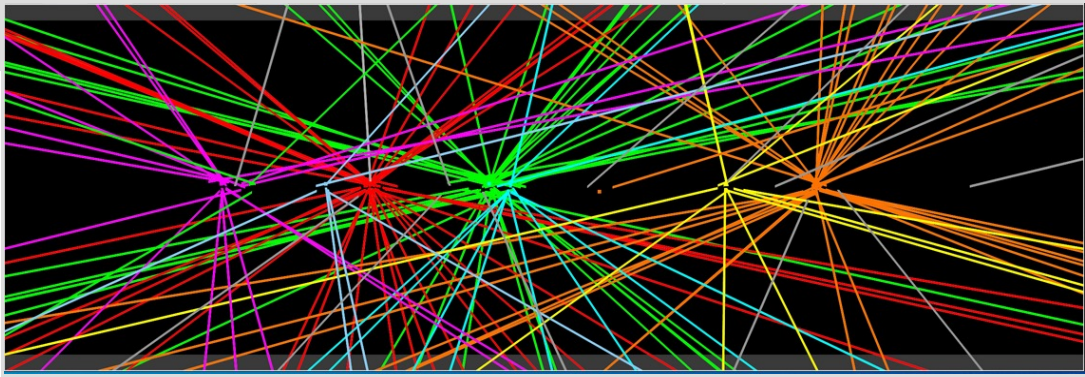
UNIVERSITÄT SIEGEN

- Inclusive vertex reconstruction problem, namely the necessity to find all vertices in a given set of reconstructed tracks, is common in the current LHC and future HL-LHC and FCC experiments.
- Increasing complexity of this problem (more explanations later) suggests application of ML approach.
- As all tracks are produced in vertices, it seems natural to solve this problem by partitioning a set of tracks into clusters representing vertices. Then the multi-vertex finding problem can be formulated as a dedicated clustering problem.
- In my presentation, I will discuss an application of the recently proposed minimum-cost lifted multicut graph partitioning algorithm to the multiple vertices finding in (HL-)LHC environment.
- Current work addresses the primary vertex reconstruction (effectively 1D problem) mainly because of the relative simplicity of getting realistic test data with varying complexity in this case. Extension to the 3D vertex finding is relatively straightforward and won't be discussed here.
- More details can be found in JINST 18 (2023) P07013

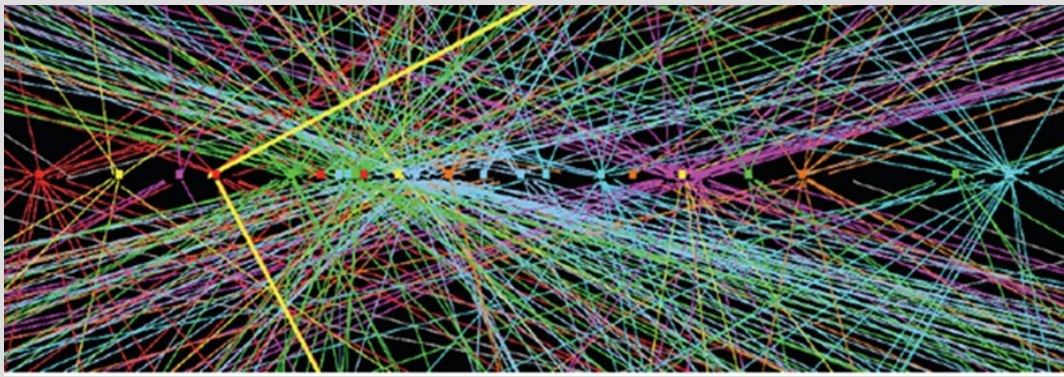
Some examples



Medical Imaging Inspired Vertex Reconstruction at LHC
Journal of Physics: Conference Series 396 (2012) 022021

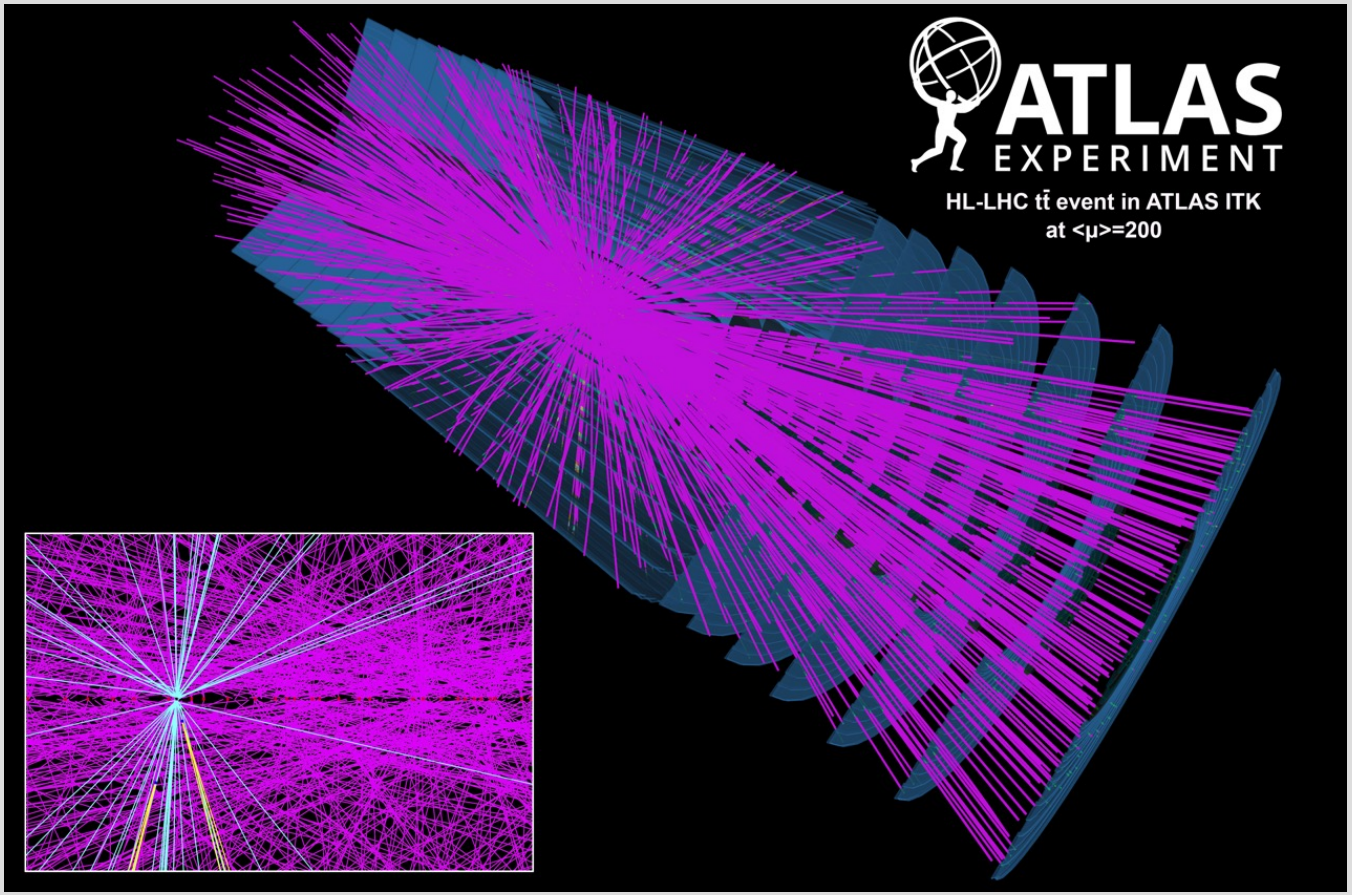


ATLAS collision event from the first 2011 LHC fill with 7 vertices.



A visual example of pile-up in the ATLAS tracker
Proc.Comp.Science v66 (2015) 540-545

Foreseen complexity



A simulated tt̄bar event at average pile-up of 200 collisions per bunch crossing, with an ITk layout including the very forward extension. The bottom-left inset is a 2D r-z view of the interaction region. The vertical scale is 2.5mm and the horizontal one 12cm. All reconstructed tracks have pT>1 GeV. The tracks coming from the ttbar vertex are coloured in cyan. Two secondary vertices can be reconstructed and the tracks coming from them are highlighted in yellow.

Vertexing problem definition

UNIVERSITÄT SIEGEN

Given a set of reconstructed tracks, one needs to find all physics vertices in it.

A physics-motivated way to solve this problem is to partition the track set into a collection of isolated, non-overlapping clusters. Each cluster represents a vertex, the parameters of which can be computed from the assigned tracks.

Then, this is an optimal clustering problem that can be solved by ML methods

1) Convert a track set into a track adjacency graph with the track-track "closeness" information encoded as edge weights;
2) Partition this adjacency graph in the most optimal way with respect to the edge weights;
3) Convert the found clusters into physics vertices.

Looks pretty straightforward, but….

Challenges

- A priory unknown (big) amount of truth vertices with unlimited track multiplicity ([2,∞]). Number of clusters needs to be decided by the clustering itself;
- Reconstructed track position displacement due to limited resolution is comparable with a typical vertex-vertex distance;
- Order of magnitude difference in reconstruction accuracy (and corresponding displacement) for various tracks;
- High track density resulted in minimum track-track distances also comparable with the vertex-vertex distances;
- High density of the tracks and vertices and significant track position displacements caused by the reconstruction errors result in a strong overlap of the tracks from different truth vertices;

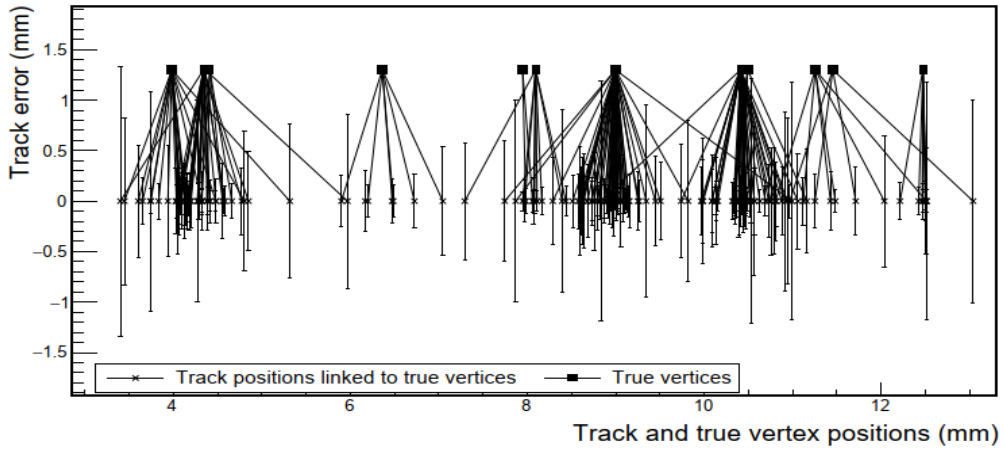


Figure 3. Example display of overlapping tracks from different vertices caused by measurement errors (zoom of a simulated DELPHES event with $\mu = 150$). The crosses at the ordinate value of 0 represent the track positions, and the vertical error bars represent the corresponding position measurement errors. Squares at ordinate values of 1.3 represent the truth vertex positions. The connecting lines show the origin vertex for every track.

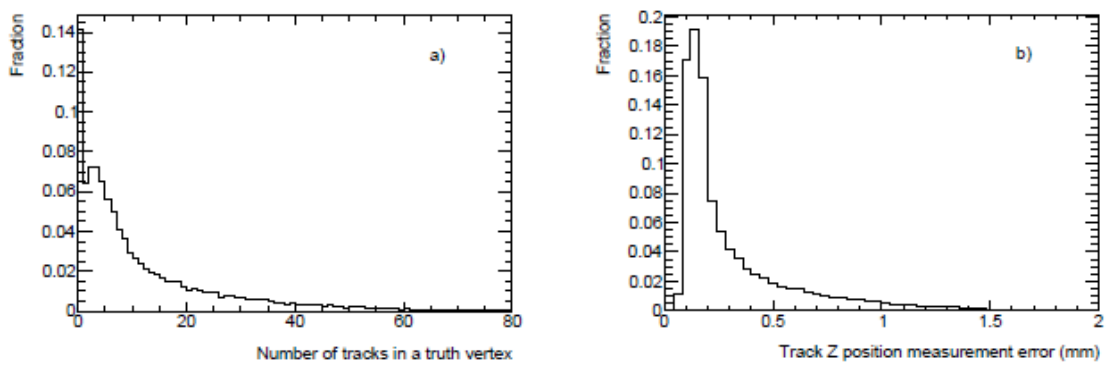
JINST 18 (2023) P07013

Test data samples

Vertex finding should be tested on data reproducing all “real life” features. DELPHES framework, arXiv:1307.6346 tuned to the typical ATLAS track reconstruction acceptance and resolution, was used for the test data production. 4 data samples, representing LHC and HL-LHC pileup conditions with increasing complexity, were generated

Energy	$\langle\mu\rangle$	Interaction region σ_z	$\langle N_{\text{trk}}^{\text{event}}\rangle$	$\langle N_{\text{trk=0}}^{\text{vrt}}\rangle$	$\langle N_{\text{trk=1}}^{\text{vrt}}\rangle$	$\langle N_{\text{trk>1}}^{\text{vrt}}\rangle$
13 TeV	63	35 mm	718	9	4	50
14 TeV	150	42 mm	1674	22	9	119
14 TeV	200	42 mm	2227	28	12	160
14 TeV	250	42 mm	2771	35	16	199

Simulation parameters. Only vertices with 2 and more reconstructed tracks are considered as reconstructable in this work.



Number of the reconstructed tracks in a pileup vertex and the track resolution. It can be compared with the interaction region size.

$\langle\mu\rangle$	63	150	200	250
Truth track overlap fraction	20%	41%	53%	66%

Truth track overlap fraction is the fraction of reconstructed tracks in an event that are closer in space to another truth vertex than to the truth vertex of origin. See previous slide.

Minimum-cost Multicut

UNIVERSITÄT SIEGEN

The most crucial part of the proposed approach defining an overall performance is the graph partitioning part. This is done by the *Minimal Cost Lifted Multicut* algorithm, originally proposed in arXiv:1505.06973.

First the *Minimum-Cost Multicut* problem definition:

The minimum cost multicut problem is a grouping problem defined for a graph $G = (V, E)$ and a cost function $c : E \rightarrow \mathrm{R}$ which assigns to all edges $e \in E$ a real-valued cost or reward for being cut. Then, the minimum cost multicut problem is to find a binary edge labelling y according to

$$\min_{y \in \{0,1\}^E} \sum_{e \in E} c_e y_e \tag{2.1}$$

subject to

$$\forall C \in \mathrm{cycles}(G) \quad \forall e \in C : y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'} \,. \tag{2.2}$$

The constraints on the feasible set of labellings y given in equation (2.2) ensure that the solution of the multicut problem relates one-to-one to the decompositions of graph G, by ensuring for every cycle in G that if an edge is cut within the cycle (y_e = 1), so needs to be at least one other. Trivial optimal solutions are avoided by assigning positive (attractive) costs c_e to edges between nodes $v, w \in V$ that likely belong to the same component, while negative (repulsive) costs are assigned to edges that likely belong to different components.

y_e=1↔ *edge is cut*, y_e=0 ↔ *edge is kept*

Minimum-cost Lifted Multicut

UNIVERSITÄT SIEGEN

The Minimum-cost Multicut problem can be extended (lifted) by adding constraints:

The minimum cost *lifted* multicut problem (LMC) generalizes over the problem defined in equation (2.1)–equation (2.2) by adding a second set of edges that defines additional, potentially long-range costs without altering the set of feasible solutions. It thus defines a second set of edges F between the nodes V of G, resulting in a lifted graph $G' = (V, E \cup F)$, on which we can define a cost function $c' : E \cup F \rightarrow \mathbb{R}$. Then, equation (2.1) and equation (2.2) are optimized over all edges in $E \cup F$ and two additional sets of constraints are defined according to [9]

$$\forall v, w \in F \quad \forall P \in v, w - \text{paths}(G) : y_{vw} \leq \sum_{e \in P} y_e \tag{2.3}$$

$$\forall v, w \in F \quad \forall C \in v, w - \text{cuts}(G) : 1 - y_{vw} \leq \sum_{e \in C} (1 - y_e) \tag{2.4}$$

to ensure that the feasible solutions to the LMC problem still relate one-to-one to the decompositions of the original graph G.

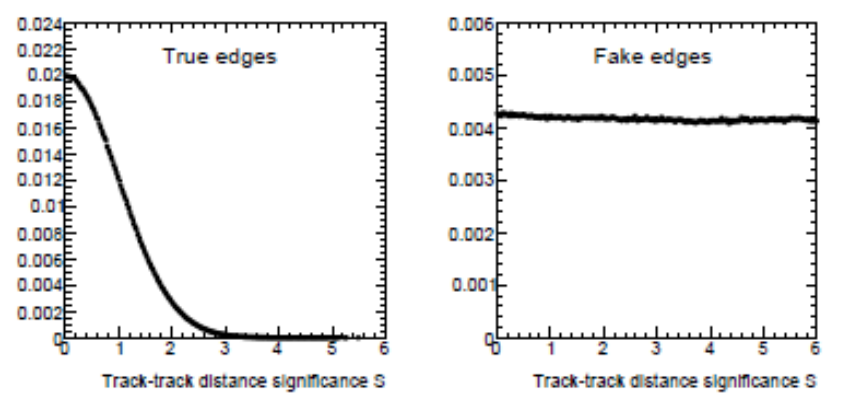
For the vertex-finding problem, this formulation allows encoding Euclidean distance constraints in the structure of graph G (e.g. point observations that are spatially distant cannot originate from the same vertex), while the cost function can be naturally defined in the distance significance space to take into account the measurement errors. The Euclidean distance and its significance can be very different in case of significant reconstruction errors, the lifted multicut approach encodes both metrics in the same graph.

Edge score options

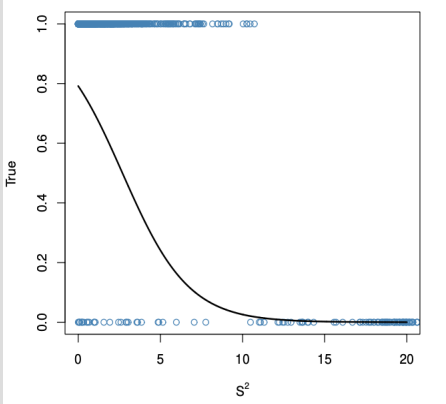
The edge weights are typically chosen to be negative for edges that should be cut and positive for those connecting nodes that should be joined.

1. Probability distribution ratio $w = \log(\frac{p_{true}}{p_{false}})$ of the minimal track-track distance significance $S = \sqrt{\frac{(x_i - x_j)^2}{\sigma_i^2 + \sigma_j^2}}$;
2. Logistic regression $p = \frac{1}{1+e^{-z}}$ where $z = \beta_0 + \beta_1 \cdot S$. Then $w = \log\left(\frac{1}{1-p}\right)$ weight has necessary features.
3. BDT edge classification (7 variables) score in [-1,1] range

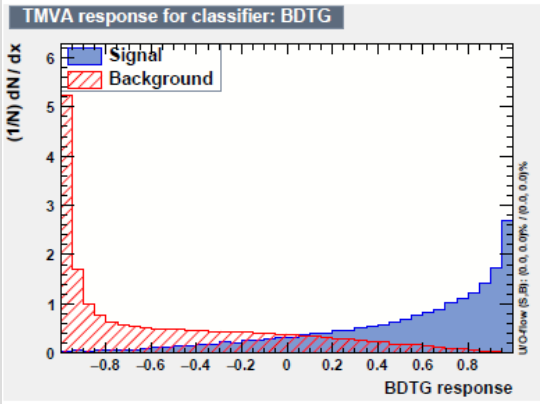
Track-track significance S



Logistic regression



BDT classification



Performance metrics

UNIVERSITÄT SIEGEN

Some metrics are needed to estimate the performance of the inclusive multi-vertex finding algorithms, maximise it, compare various options, etc...

Many metrics for clustering performance estimation are available in the statistical literature: Rand index, Silhouette score, Mutual information, Fowlkes–Mallows index, Jaccard index, Mirkin metric, Calinski-Harabasz Index, etc. They either compare the obtained cluster set with the truth or favour partitioning producing well-separated clusters.

In the multi-vertex finding problem exact truth reproduction is not possible due to significant track overlap caused by the reconstruction errors. Obtained clusters are very close as well. This can affect the features of the standard metrics and make problematic the corresponding scores interpretation.

Something adapted more for multi-vertex problem would be useful.

Performance metrics

UNIVERSITÄT SIEGEN

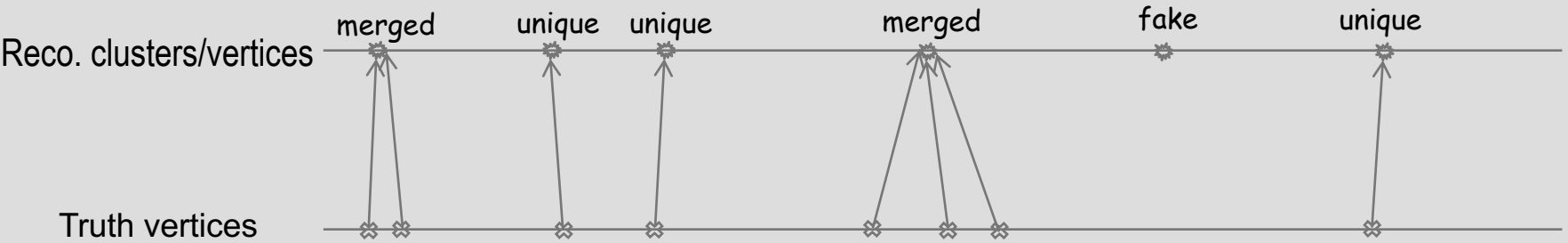
1. Statistical solution:
   Modify the existing metrics by weighting individual contributions with reconstruction errors $w = \frac{1}{\sigma^2}$. Then metrics get dominated by well-measured / least displaced tracks, while the influence of significantly displaced tracks is reduced. *Variation of Information* (VI) J.Multivar.Anal. 98(2007),873 and *Silhouette* J.Comput.App.Math. 20(1987),53 metrics are modified in this way.

2. Physics-inspired solution:
   Usually, the total amount of vertices and their positions are of primary importance in inclusive vertex-finding problems, the track-to-vertex assignment can be addressed afterwards, if needed. Therefore, a metric based on the "closeness" of the truth vertex positions to the cluster centers of mass, not relying on track-to-vertex counting, is needed. For universality, this metric should not be based on any explicit scale (typical resolution, averaged vertex-vertex distance, etc)

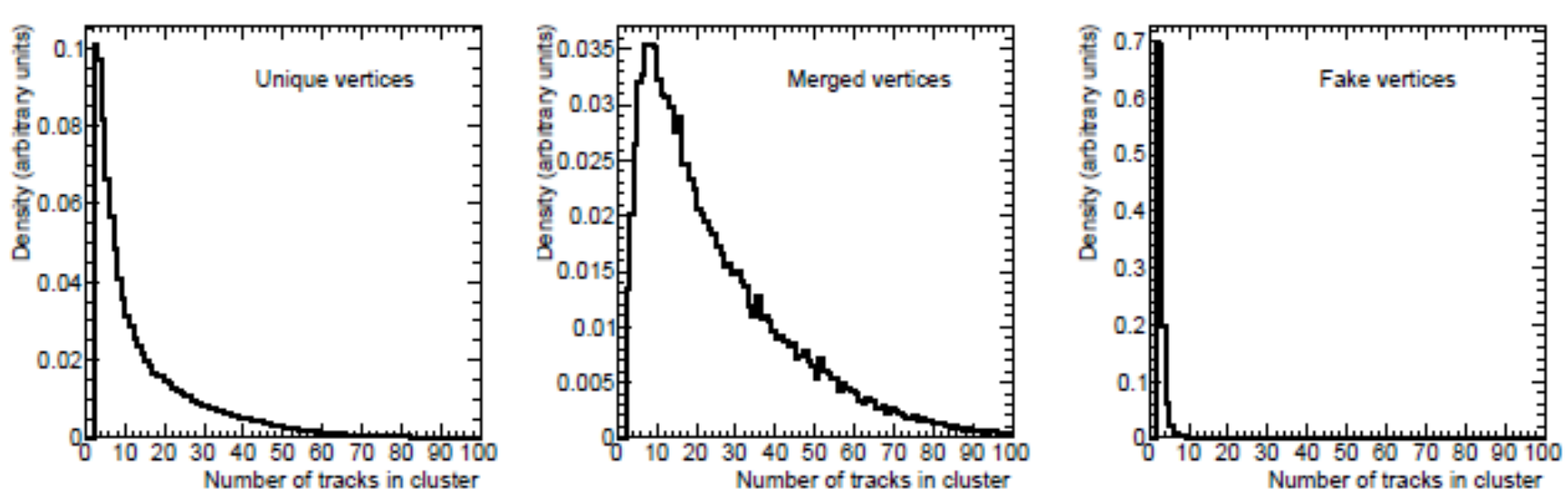
Physics-inspired performance estimator

Every reconstructable truth vertex is linked to the closest reconstructed cluster in the Cartesian space. Thus, a list of linked reconstructed clusters is obtained. Then, every reconstructed cluster is classified depending on how many times it enters into this list. If a cluster enters this list only once, there is just a single truth vertex referencing this cluster. Therefore, it can be called *unique*, which means that a truth vertex is unambiguously reconstructed as a cluster. If a cluster enters several times into the list, it is referenced by several truth vertices, and therefore it combines tracks from these vertices: this cluster can be called *merged*. Also, some clusters may not appear in this list at all: such clusters are not referenced by any truth vertex and are thus *fake*.

The total number of obtained clusters and their classification as *unique*, *merged*, *fake* are scale-independent and can be used as a metric to compare various clustering options.



Physics-inspired performance estimator



The *unique*, *merged* and *fake* clusters behave as expected.

Performance: preliminaries

UNIVERSITÄT SIEGEN

1) *Variation of Information* (VI) metric estimates the information difference between the truth (true vertices and true track-to-vertex association) and reconstructed clusters. A smaller VI value corresponds to a better agreement between truth and reconstruction.

2) *Silhouette* metric estimates the compactness of the obtained clusters. A bigger *Silhouette* value (≤1.0) corresponds to a smaller cluster size in Cartesian space what means a better solution.

3) N_{trk}^{wrong} is a fraction of the tracks which are fully surrounded by tracks assigned to a different cluster. This is a consequence of a mismatch between the track-track Cartesian distance and this distance significance based on track reconstruction errors.

Performance: Results μ=63

UNIVERSITÄT SIEGEN

Edge weight		VI	VI weighted	Silhouette	Silhouette weighted	Unique	Merged	Fake	N_{trk}^{wrong}	CPU
PDF ratio	base	0.839	0.407	0.615	0.646	33.3	8.2	2.4	15%	0.25s
	cnst	0.782	0.362	0.649	0.660	33.9	7.9	2.3	8%	0.18s
Regression	base	0.860	0.416	0.589	0.623	34.7	7.6	4.1	14%	0.27s
	cnst	0.829	0.387	0.614	0.633	35.0	7.5	3.9	8%	0.18s
BDT	base	0.945	0.399	0.478	0.230	35.0	7.5	7.1	5%	0.23s
	cnst	0.937	0.377	0.487	0.234	35.2	7.4	7.0	4%	0.14s

1) ~70% of the true reconstructable truth vertices are reconstructed as the *unique* clusters;
2) The weighting of the statistical metrics reduces the influence of the widely spread tracks with big reconstruction errors, thus improving their informativeness;
3) Clustering with constraints (lifted adjacency graph) provides better solutions according to all metrics and is approximately 30% faster;
4) Clustering with the BDT-based weights seems the best (*unique*, N_{trk}^{wrong}, VI). But this solution has more fakes and bad Silhouette value.

Performance: Results μ=250

UNIVERSITÄT SIEGEN

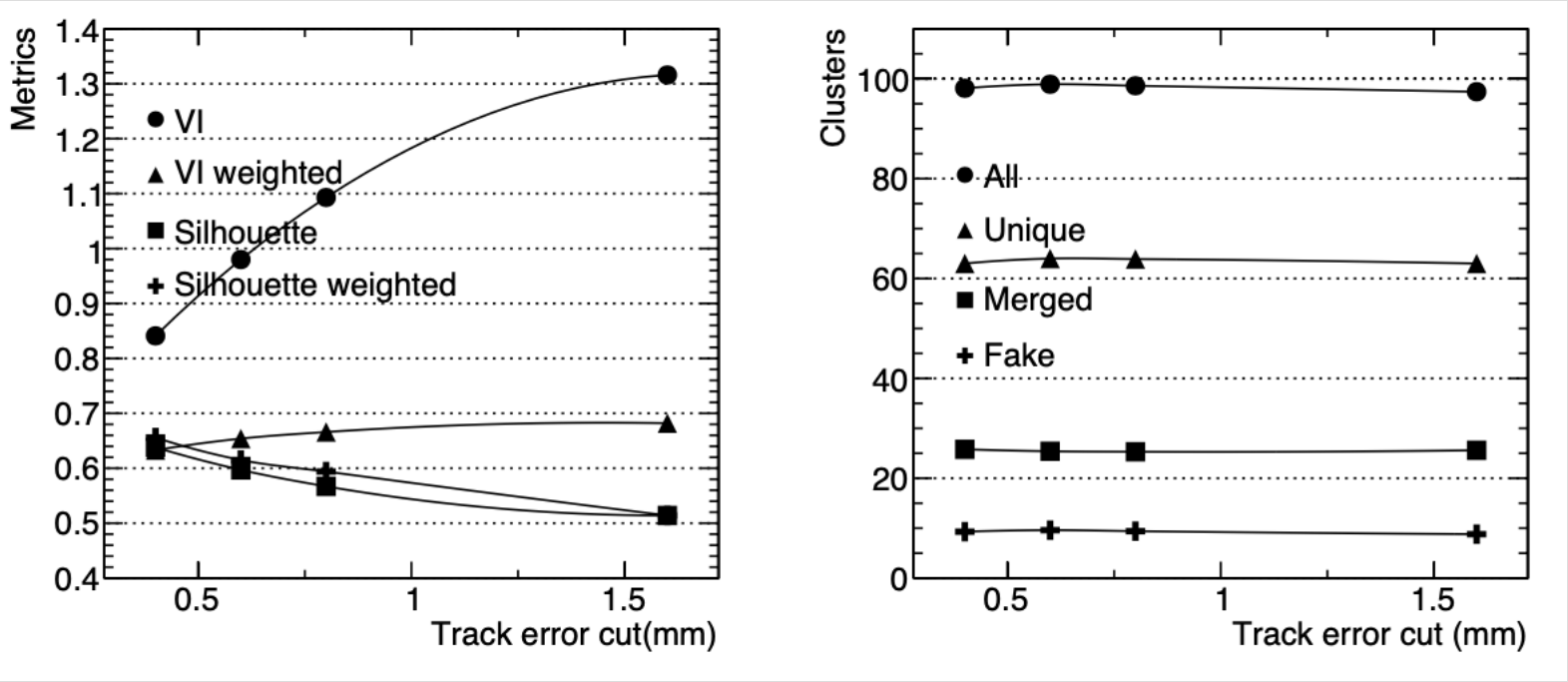
Edge weight		VI	VI weighted	Silhouette	Silhouette weighted	Unique	Merged	Fake	N_{trk}^{wrong}	CPU
PDF ratio	base	1.782	0.990	0.477	0.526	68.7	53.2	6.4	42%	3.0s
	cnst	1.638	0.887	0.531	0.569	71.0	52.7	5.3	21%	1.7s
Regression	base	1.753	0.961	0.467	0.517	77.1	51.2	11.	38%	3.2s
	cnst	1.672	0.895	0.505	0.547	77.8	51.1	9.9	21%	1.7s
BDT	base	1.691	0.941	0.307	0.040	72.8	52.4	15.	12%	3.0s
	cnst	1.651	0.882	0.330	0.055	74.5	52.0	14.	9%	1.2s

1) Solution quality is degraded according to all metrics, only ~38% of the true reconstructable vertices are reconstructed as *unique* clusters;
2) Biggest number of the *unique* clusters is obtained with the logistic regression-based weighting, not BDT-based one;
3) Clustering with constraints provides better solutions according to all metrics and is ~50% faster;
4) Clustering with the BDT-based weights provides very bad Silhouette metric, but the lowest fraction of badly associated tracks.

Performance: Resolution check

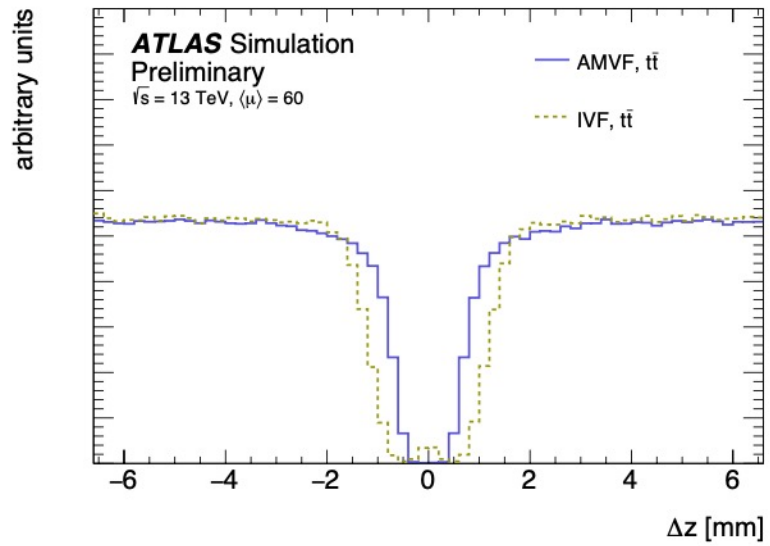
UNIVERSITÄT SIEGEN

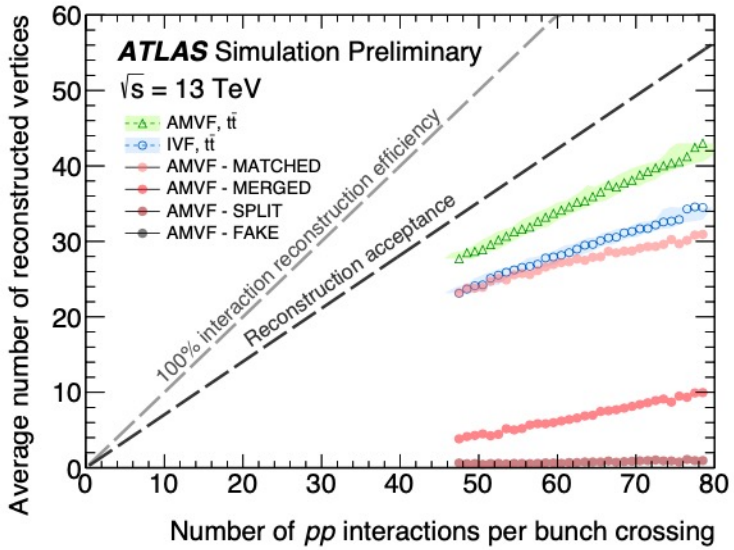
As the origin of the problems in the vertex finding is significant track reconstruction errors, one can try to remove the badly reconstructed tracks before clustering.

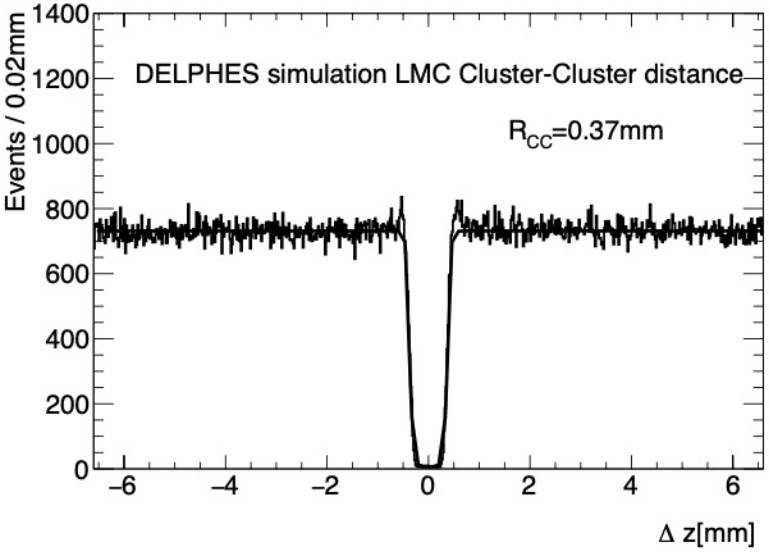


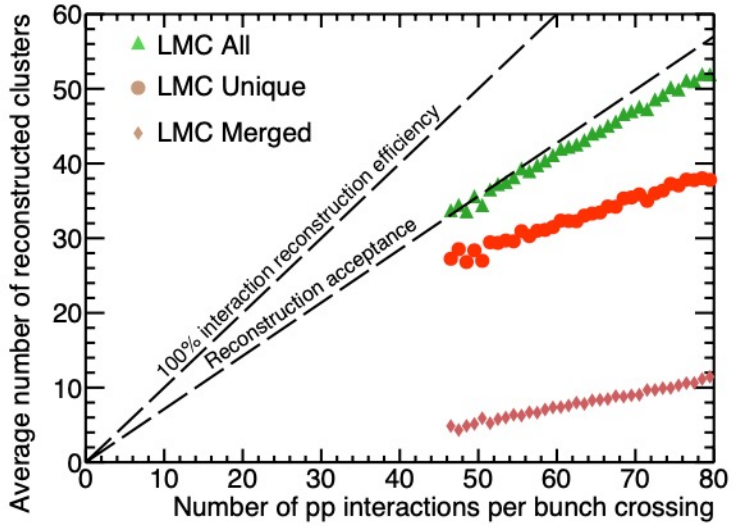
1. Statistical metrics demonstrate strong dependence on track error cut.
2. Physics-inspired metric value is practically independent on this cut

Performance: Comparison





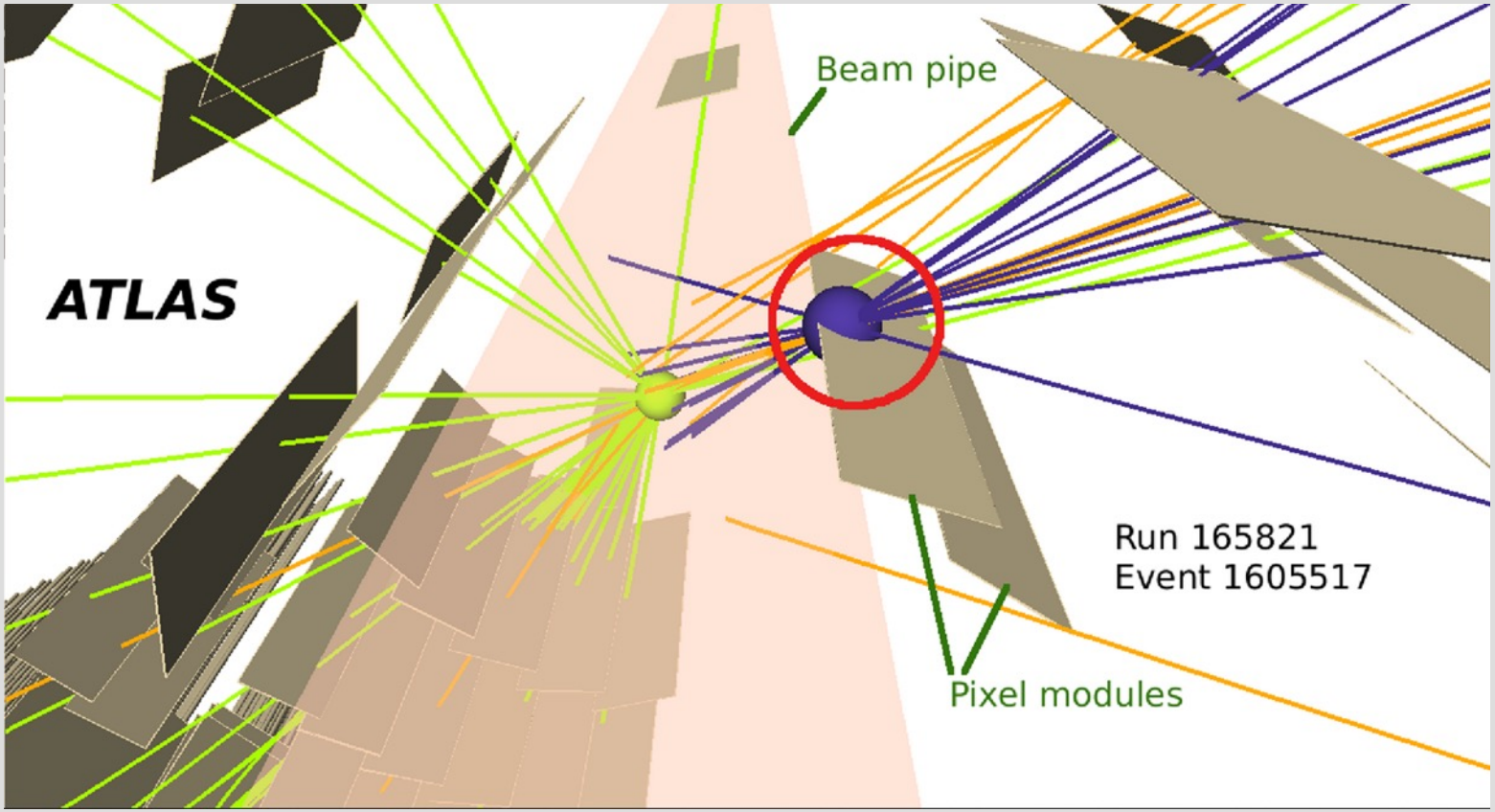




Conclusions

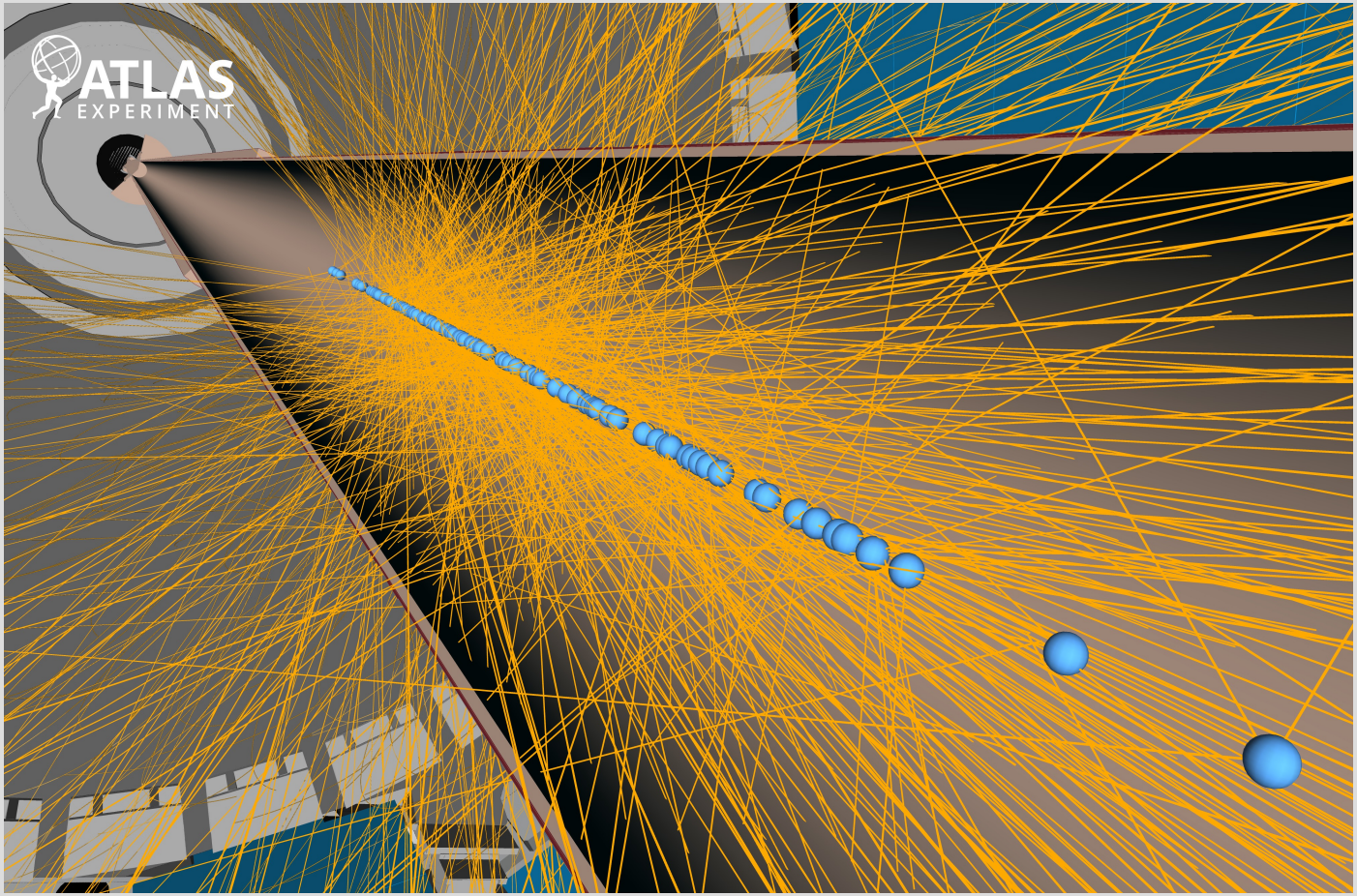
1. Big primary vertex density and important track reconstruction errors lead to significant overlap of the reconstructed tracks from different vertices in a typical (HL-)LHC environment. Such overlap reaches 66% for μ=250 pileup;
2. The overlap complicates the multi-vertex finding problem and bias standard performance metrics;
3. Alternative physics-inspired metric based on cluster/vertex type (*unique/merged/fake*) counting seems better suited for the multiple primary vertices finding problem;
4. The best understanding of the multi-vertex clustering features can be gained by simultaneous usage of several metrics covering different aspects of performance;
5. Minimum-cost lifted multicut graph partitioning algorithm efficiently solves the multi-vertex finding problem;
6. Constraints in the partitioning algorithm do improve the clustering performance;
7. More complex edge score estimations (GNN?) and more elaborated constraints may boost the overall performance further;

3D example



An event from a jet-trigger data sample, where a high-mass vertex (circled) is the result of an apparently random, large-angle intersection between a track and alow-mass hadronic-interaction vertex produced in a pixel module. Tracks originating from this vertex are shown in blue, those from the primary vertex are green, and other tracks are orrange. The beampipe and pixel modules with track hits are shown.

Backup: Foreseen HL-LHC complexity

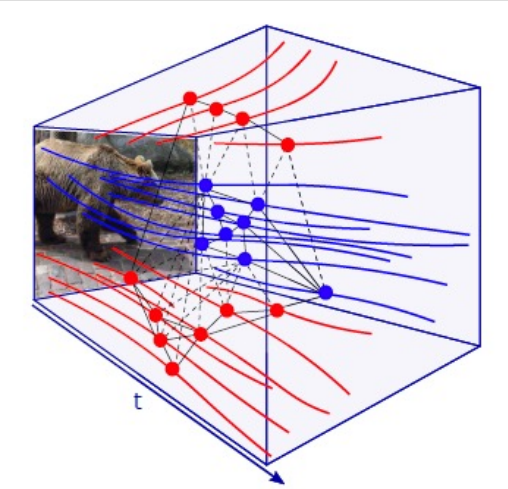


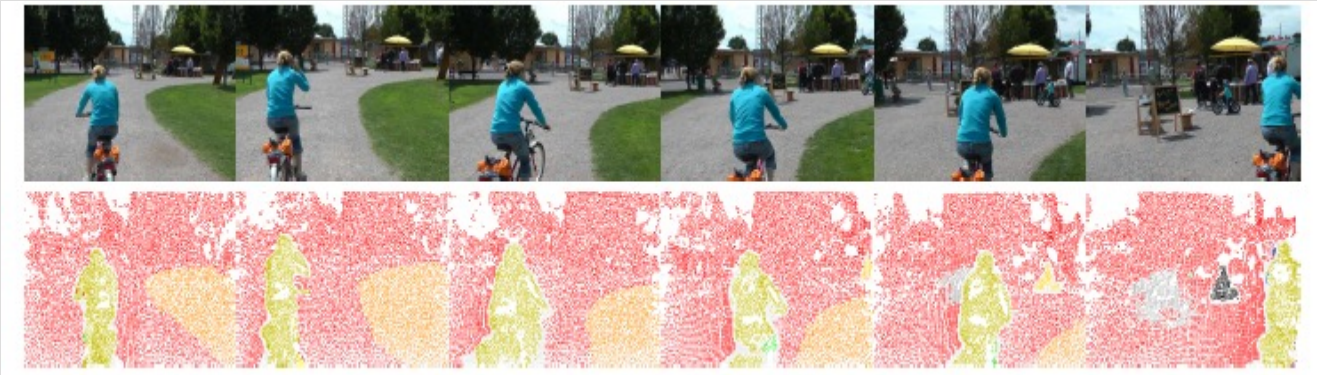
A visualisation of simulated top-antitop quark pair production in a proton-proton collision at 14 TeV center-of-mass energy at the future High-Luminosity LHC (HL-LHC). The simulated event includes approximately 200 pileup interactions in the same bunch crossing, and the image shows 88 primary vertices (blue balls) reconstructed along the beam line. From each primary vertex, many different particles are produced, whose tracks are visualised here in light orange. In total, more than 2000 tracks are reconstructed in the proton-proton interaction. The image shows a cutout of the LHC beam pipe at the center of the ATLAS detector; in the background, a part of the muon spectrometer and the shielding are visible; the other parts of the detector have been hidden for a better view of the interaction point. (HL-LHC ttbar mu 200 with 88 vertices)

Backup: Motion trajectories

UNIVERSITÄT SIEGEN

Identify different object and trace their evolution through all frames in a movie





From: 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 3271-3279

Minimum Cost Multicut algorithm demonstrates high efficiency in this problem

Backup: Resolution

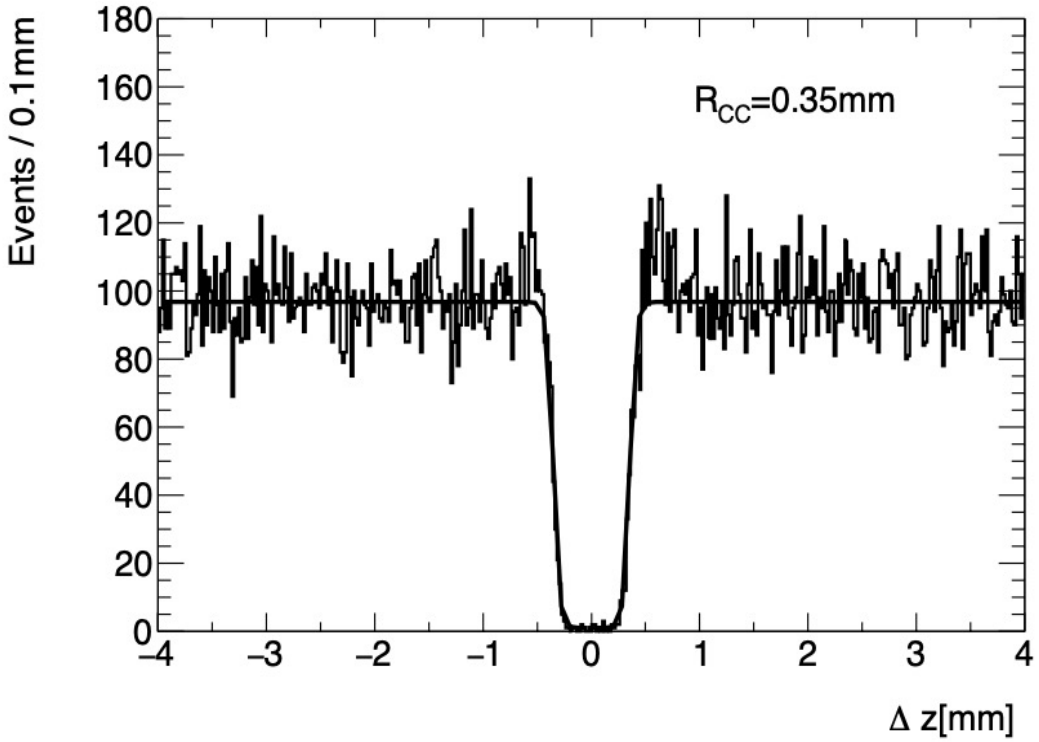


Figure 7. Example of a fit to the cluster-cluster distance to determine the resolution. The used fitting function is $a/\{1+\exp[b\cdot(R_{cc}-|x|)]\}+c$ where a, b, c are free fitting parameters and R_{cc} is the cluster-cluster resolution, defined as the half-width at the half-depth of the dip in the centre of the cluster-cluster weighted centre distances, averaged over all clusters.

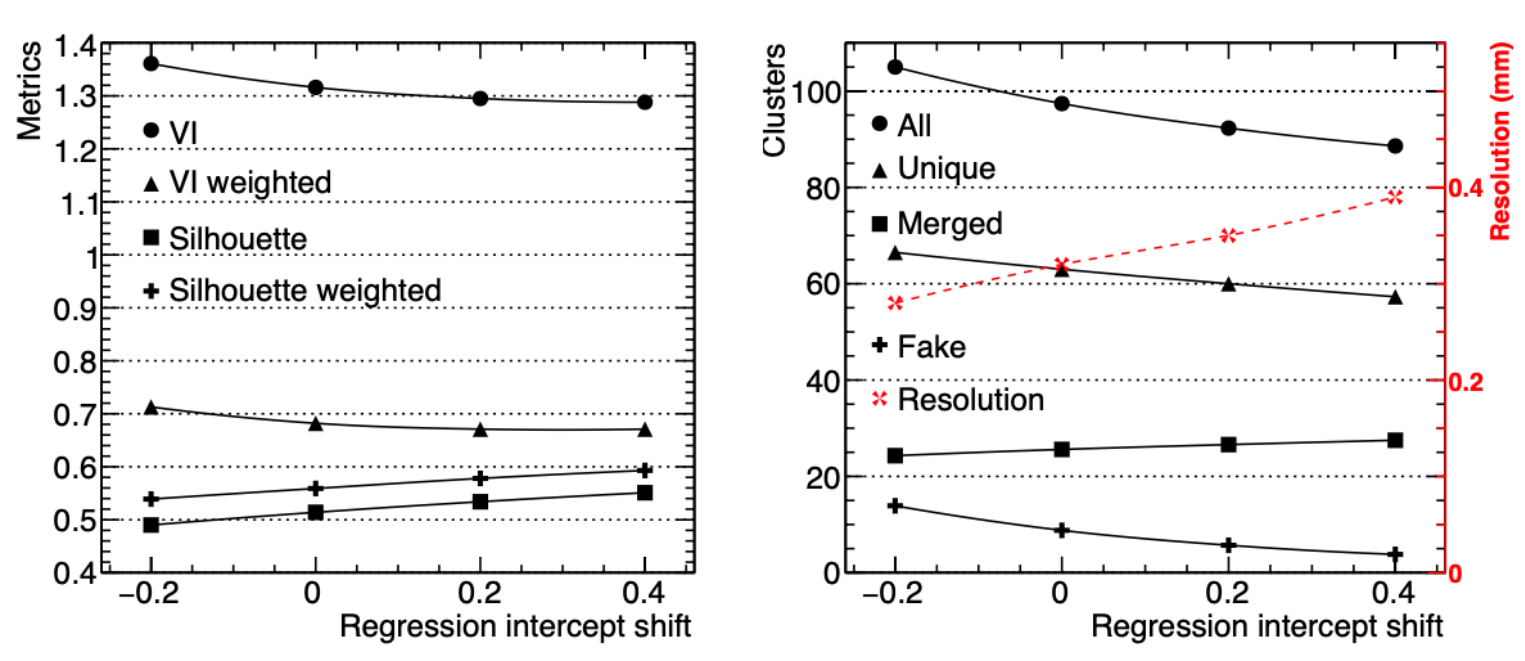
Backup: BDT variables

UNIVERSITÄT SIEGEN

1. Squared significance S^2 (or $\chi^{2)}$ of track-track distance along beamline
2. Average position of the track pair along beamline
3. Position measurement uncertainty σ_{z0} of track 1
4. Position measurement uncertainty σ_{z0} of track 2
5. Pseudorapidity η of track 1
6. Pseudorapidity η of track 2
7. Number of other tracks crossing the beamline between tracks 1 and 2

Backup: Improvements?

Performance of the clustering can be modified by changing the edge score, e.g., by introducing a prior probability that a given edge in the adjacency graph should be cut . In case of regression this is equivalent to the modification of the intercept term.



1. Limited performance modification, e.g. tuning of *unique* vs *fake* ratio, is possible,
2. None of the performance metrics demonstrates an optimum.
   There is no “optimal” clustering for the inclusive vertex-finding problem, additional tuning for any concrete problem is recommended.