## **Track Fitting and Vertexing** EDM

Joe Osborn BNL October 11, 2023

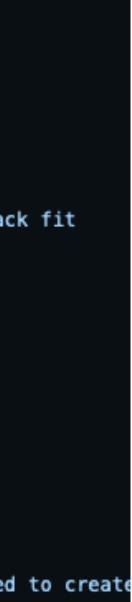
### Overview

- Some discussion last week about the output collection from the fitter
- Issue currently 3 outputs from the fitter, all of which contain (effectively) the same information
- Has downstream implications for all other reconstruction algorithms using tracks (electron finder, PID, vertexing...)
- Will propose a solution for the fitting output, and then a corresponding design for the vertexing EDM

- Current output:
  - 1. edm4eic::TrajectoryCollection contains all track states
  - 2. edm4eic::TrackParametersCollection contains track parameters target surface
  - 3. std::vector<ActsExamples::Trajectories> An Acts EDM object v packages parameters and states for use by other Acts algorithm
    - 1. Note: Acts development team moving away from this object
- Proposal:
  - Single output of edm4eic::TrackCollection
    - This will contain the track parameters at the target surface, all states and associated hits (eventually, measurements), and a v association
    - Missing a field for track position at target surface, which shoul added
    - Additional fields from edm4eic::Trajectory can be added, e.g. r nOutliers etc. (these are inspired from the Acts trajectory object

# CKF Output

	edm4eic::Track:		
	Description: "Track inf	formation at the v	vertex"
ers at the	Author: "S. Joosten"		
	Members:		
	- int32_t t	type	<pre>// Flag that defines the type of track</pre>
	– float d	chi2	<pre>// Total chi2 (sum) of the track fit</pre>
which	<pre>- int32_t r</pre>	ndf	// Numbers of degrees of freedom of the trac
ns	- edm4hep::Vector3f m	momentum	<pre>// Track 3-momentum at the vertex [GeV]</pre>
	- edm4eic::Cov3f m	momentumError	<pre>// Covariance matrix on the momentum</pre>
ject	– float t	time	<pre>// Track time at the vertex [ns]</pre>
<b>j</b>	- float t	timeError	<pre>// Error on the track vertex time</pre>
	– float d	charge	// Particle charge
	OneToOneRelations:		
	- edm4eic::Trajectory	y trajectory	<pre>// Trajectory of this track</pre>
	- edm4eic::Vertex v	vertex	<pre>// Track vertex of this track</pre>
	OneToManyRelations:		
ll track	- edm4eic::TrackerHit	t trackerHits	<pre>// Hits that were used for this track</pre>
vertex	- edm4eic::Track t	tracks	<pre>// Tracks (segments) that have been combined</pre>
uld be			
uld be			
nStates,			
ect)			



# Advantages/Disadvantages

- Advantages
  - Single output container that is defined within our EDM, so we maintain control and are not affected by external changes (e.g. Acts updates)
  - Contains all track information that will realistically be needed by any downstream algorithm or analysis
- Disadvantages
  - Have to use additional CPU time to swap in between edm4eic and Acts::EDM for other tracking algorithms (e.g. vertexing, track projections, whatever else comes along)
    - Ultimately a small price to pay to insulate ourselves from external changes

### Vertex EDM

## ====================================		===				
## Vertexing						
## ===================		====				
<pre>edm4eic::Vertex:</pre>						
Description: "EIC verte	ex"					
Author: "W. Armstrong,	S. Joosten, ba	sed	off EDM4hep"			
Members:						
- int32_t	primary	//	Boolean flag, i			
– float	chi2	//	Chi-squared of			
– float	probability	//	Probability of			
<pre>- edm4hep::Vector3f</pre>	position	//	[mm] position o			
<pre>## this is named "covMatrix" in EDM4hep, renamed for co</pre>						
<pre>- edm4eic::Cov3f</pre>	positionError	11	Covariance matr			
- int32_t	algorithmType		Type code for t			
<pre>## Additional parameter not in EDM4hep: vertex time</pre>						
– float	time	//	Vertex time			
VectorMembers:						
– float	parameters	//	Additional para			
OneToOneRelations:						
## @TODO: why one and	d not multiple	par	ticles?			
<pre>- edm4eic::Reconstruc</pre>	ctedParticle as	soc	iatedParticle //			

- Current vertex object is missing several notable fields
  - relations to many tracks (not a single particle), missing NDF of vertex fit

\_\_\_\_\_ \_\_\_\_\_ f vertex is the primary vertex of the event the vertex fit the vertex fit of the vertex. consistency with the rest of edm4eic rix of the position the algorithm that has been used to create the vertex — check/set the colle ameters related to this vertex — check/set the collection parameter "Vertex reconstructed particle associated to this vertex.

• Missing time covariance, primary is not clear in streaming context (what is "the" PV?), should have

- vertex evaluation
  - that could be evaluated
- Robust vertex evaluation will be a longer term task



• There is also currently no truth vertex object, or any robust way to do

 This will become a critical issue when doing studies with backgrounds, where (e.g.) one may have many truth vertices in a single time frame

# **Proposal for Reco Vertex Object**

edm4eic::Vertex: Description: "EIC vertex" Author: "J. Osborn" Members:

- uint32 t - float - float
- edm4eic::Cov4f

OneToManyRelations:

- chi2 ndf // NDF of the vertex fit

- Contains a complete 4D position and covariance
- 3)
- Contains chi2/ndf to evaluate goodness of vertex fit

type // Type flag, to identify what kind of vertex is identified // Chi-squared of the vertex fit - edm4hep::Vector4f fullPosition // [mm] position + time t0 of the vertex. fullPositionError // Covariance matrix of the position

edm4eic::Track associatedTracks // reconstructed tracks associated to this vertex.

• Contains relations to reconstructed tracks, which is what the CKF will point to (from slide

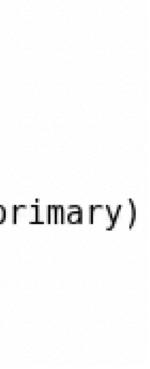
# **Proposal for Truth Vertex Object**

edm4eic::GeneratedVertex: Description: "Generated EIC vertex" Author: "J. Osborn" Members: - uint32\_t type // Type flag, to identify what kind of MC vertex - edm4hep::Vector4f position // [mm] position + time t0 of the MC vertex OneToManyRelations:

edm4eic::ReconstructedParticle sim // reference

- Generated vertices are the vertices from final state particles, not MCVertex which is going to be overkill for our purposes
- Only require a 3+1 position+time and a type to identify the vertex
- Note relation to ReconstructedParticle because currently in our reconstruction a ReconstructedParticleCollection is filled with the actual final state generated particles. See this <u>algorithm</u>
  - Some discussion about whether or not an edm4eic::Vertex object could be used as a GeneratedVertex object. However, we need a truth vertex to point back to truth particles, not edm4eic::tracks

- // Type flag, to identify what kind of MC vertex is identified (e.g. background vs. primary) // [mm] position + time t0 of the MC vertex
  - // reference to the Generated particles associated to the GeneratedVertex



# To-Do List

- Change CKF output collection to edm4eic::TrackCollection. This will have downstream implications for all other algorithms that use tracks
- Introduce new Vertex and GeneratedVertex objects
- Alter vertexing algorithm to use this new object
- Write a MC algorithm to fill GeneratedVertexCollection and output with PODIO
- Need an algorithm to relate reconstructed to generated vertices