Parameter Optimization and Layer Linking in ACTS Track Seeding

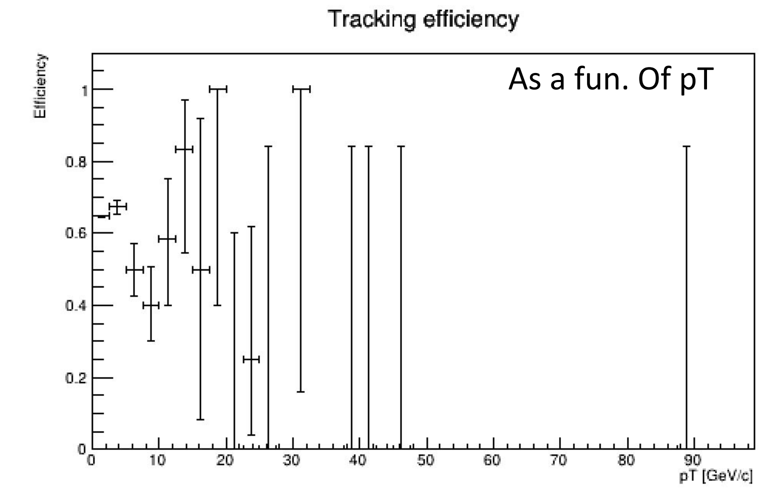
Peter Chatain, Rocky B Garg, Lauren Tompkins

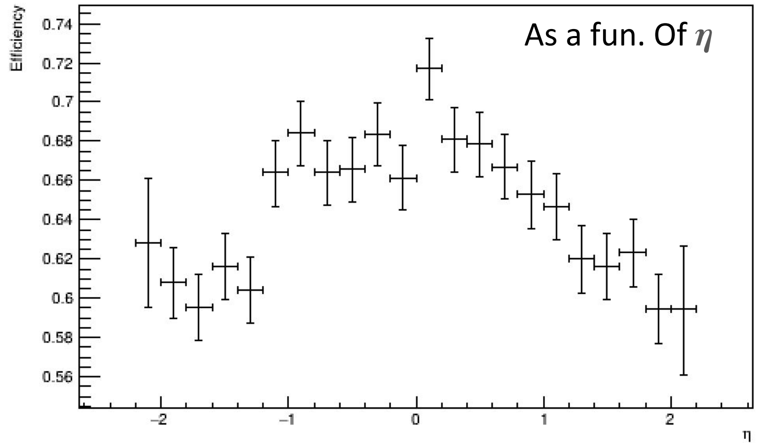
Stanford University

ACTS Machine Learning Meeting
February 26th, 2021

1. Parameter Optimization : Background

- Motivation: Originally found very poor performance of track seeding ~50% efficiency on ttbar sample with generic detector
  - 200 pileup, generic detector
  - Efficiency = fraction of true particles with a matched seed
- Tried filtering out particles that don’t have 3 hits in the pixel detector
  - Only small improvement seen ~65% efficiency
  - Implied cuts needed to be tuned, *hand tuning was not very efficient* (see back-up)

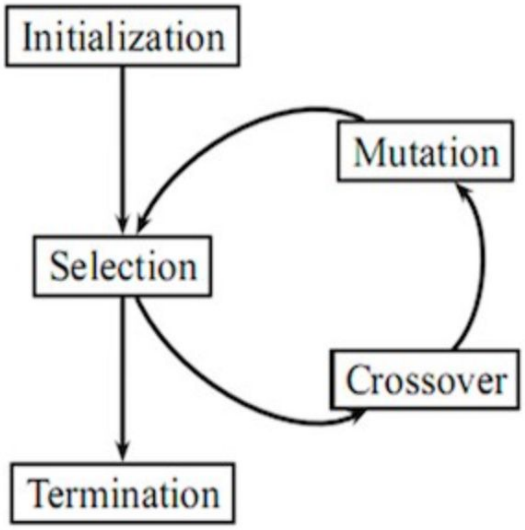




Alternative: Evolutionary Algorithm for parameter optimization

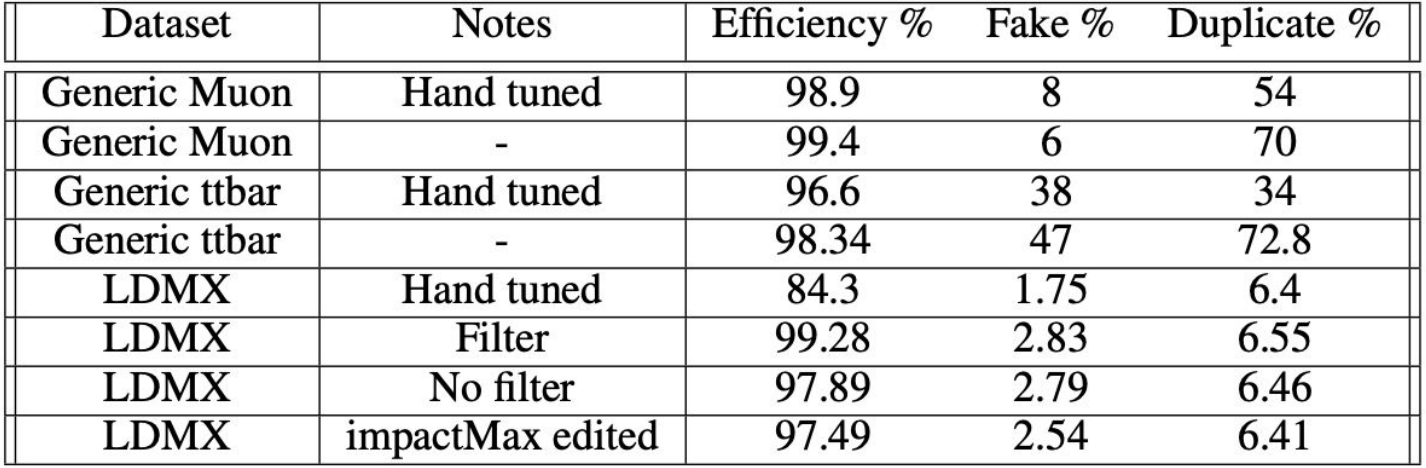
DISTRIBUTED EVOLUTIONARY ALGORITHMS IN PYTHON

- Initialization
    - Provide a good guess, create N copies of it
    - **Individual = one seedfinder configuration**
- Selection
    - Evaluate the population with scoring function:
        - Score = Efficiency - fake rate * duplicate rate/1000
    - randomly delete poor performing individuals
    - replicate good performing individuals to keep pop size constant
- Mutation
    - Each individual has a 0.3 chance of being mutated
    - If mutated, each value in an individual has a 0.2 chance of being mutated
    - Mutation is drawn from gaussian distribution centered at 0
    - Numbers hand chosen before running the algorithm
- Termination
    - Either max gen reached or ideal (> 99.4% efficiency, < 10% fake rate, < 60% duplicate rate)



Achieved good results

Required particles to have 3 hits in pixel layers, and 2 hits in the outer detectors



Dataset	Notes	Efficiency %	Fake %	Duplicate %
Generic Muon	Hand tuned	98.9	8	54
Generic Muon	-	99.4	6	70
Generic ttbar	Hand tuned	96.6	38	34
Generic ttbar	-	98.34	47	72.8
LDMX	Hand tuned	84.3	1.75	6.4
LDMX	Filter	99.28	2.83	6.55
LDMX	No filter	97.89	2.79	6.46
LDMX	impactMax edited	97.49	2.54	6.41

Table 1: Evolutionary Algorithm Results

Next Steps for Parameter Optimization

- Optimizing the seedfinder parameters for the ITk Geometry
- Integrating within ACTS to reduce computational overhead from reading in space points and particle files repeatedly
- Apply to tracking algorithm parameters:
    - All that’s required is a way to run the algorithm in parallel, and read in results to judge which configuration performed best.
- Writing up for CHEP paper (almost complete!)

2. Layer Linking: Idea behind this approach

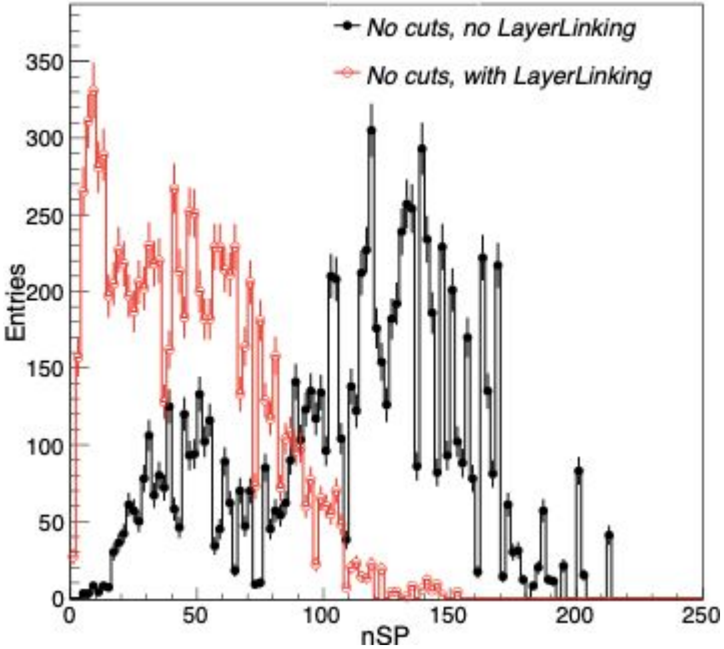
- Based on simulated data, probability of a track to go from one layer to another layer is computed
- The layer pairs sharing more tracks get high probability
- Each layer pair within the tracking detector is assigned a value which signifies how frequently these layers share the same track
- Now, the track is searched within these connected layers only
- This approach was originally implemented by Dmitry Emeliyanov, 2nd winner of TrackML challenge, in his solution
- This approach helped a lot in reducing combinatorics

Pixel layer linking

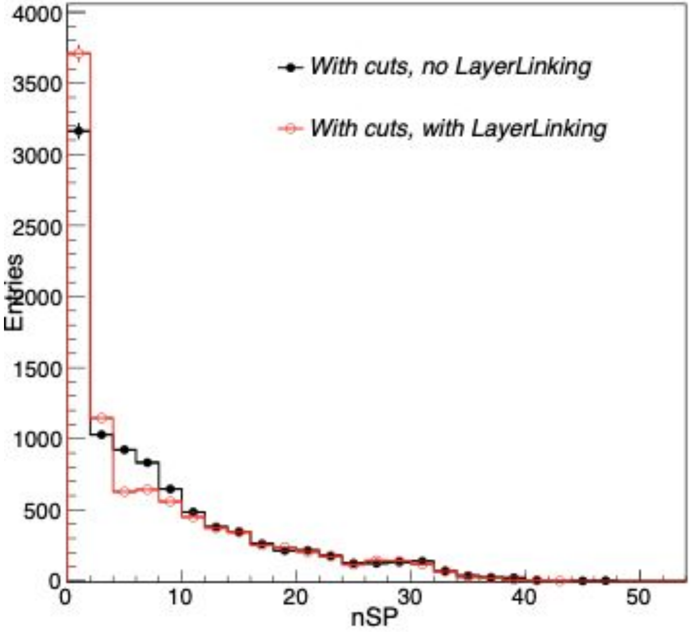
- Since track seeding use only innermost pixel detector, we have tried to implement layer linking just on pixel layers
- Only the connected layers are considered while choosing compatible bottom and top SPs for a given middle SP
- The resulting algorithm is found to be almost equally efficient as the original one (~ 1% decrease in efficiency)
- Fake rate remain same while duplicate rate reduced by 2% upon layer linking implementation
- CPU time consumption is also found to be almost similar for both cases

Effect of adding layer linking in track seeding

Number of top/bottom space points to be studied for each middle space point



No ΔR, Δ cotθ and z cuts



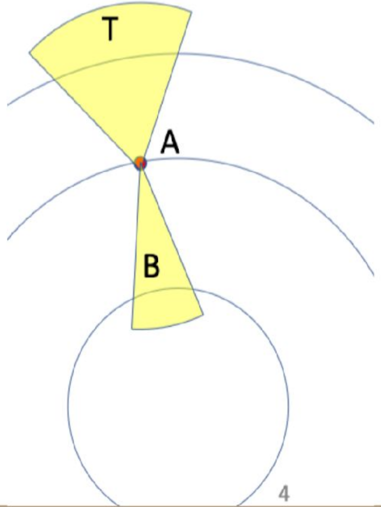
with ΔR, Δ cotθ and z cuts

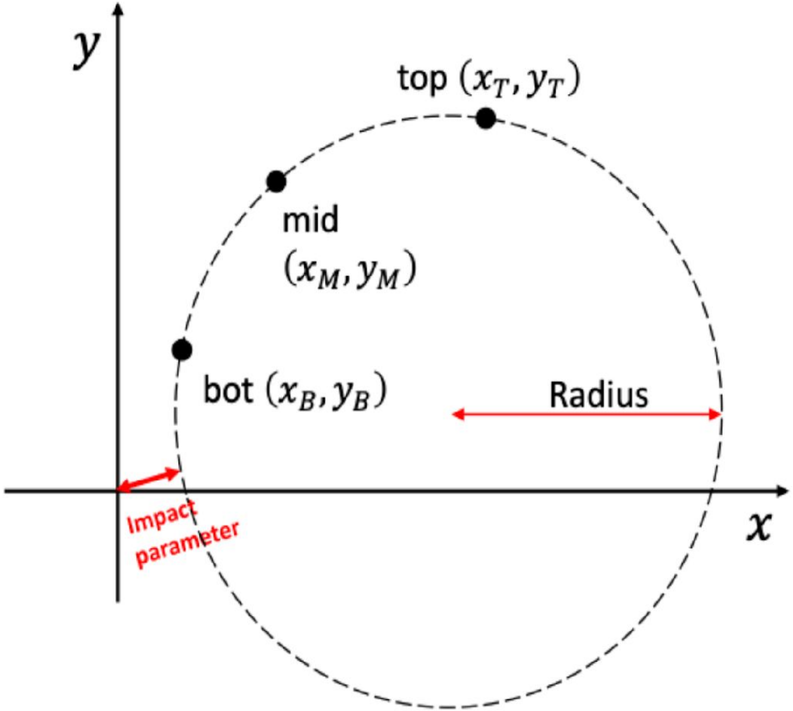
Back-up

How does the Seed Finding Algorithm Work?

- **Groups hits into 3 bins – Top, Middle, and Bottom**
- **Create seeds within search window**
  - Filter Seeds according to parameters (e.g. Impact Parameter)
- **Search complexity grows exponentially with more data points**

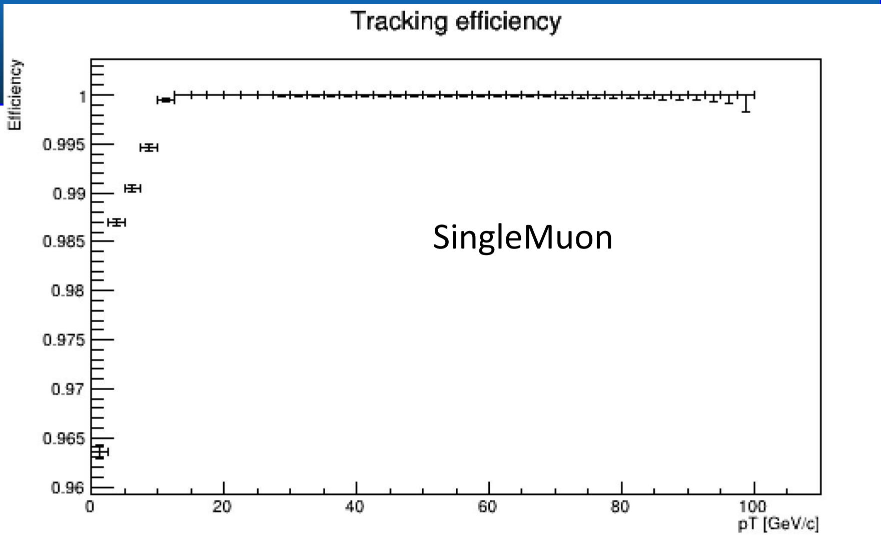
Search window per middle space point “A”

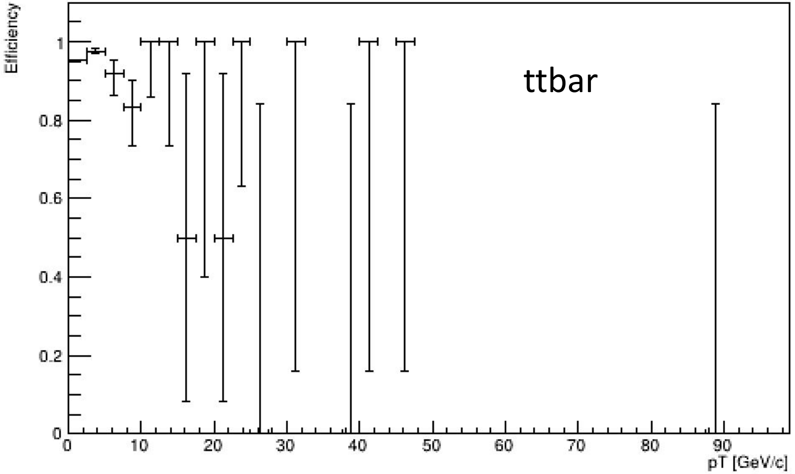




Hand Tuning

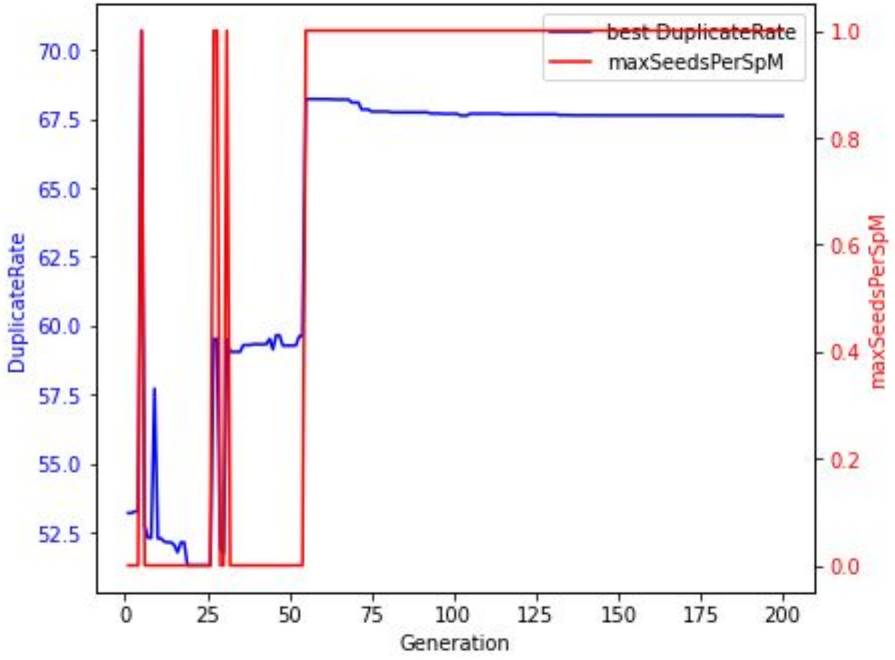
- Wrote a script using multi-processing to analyze which configuration to use
- Removed one parameter (maxPtScattering) which was behaving weirdly
- Downsides:
  - Parameters depend on each other so takes many iterations
  - Inefficient exploration of high dimensional space
  - Unclear whether configuration is optimal

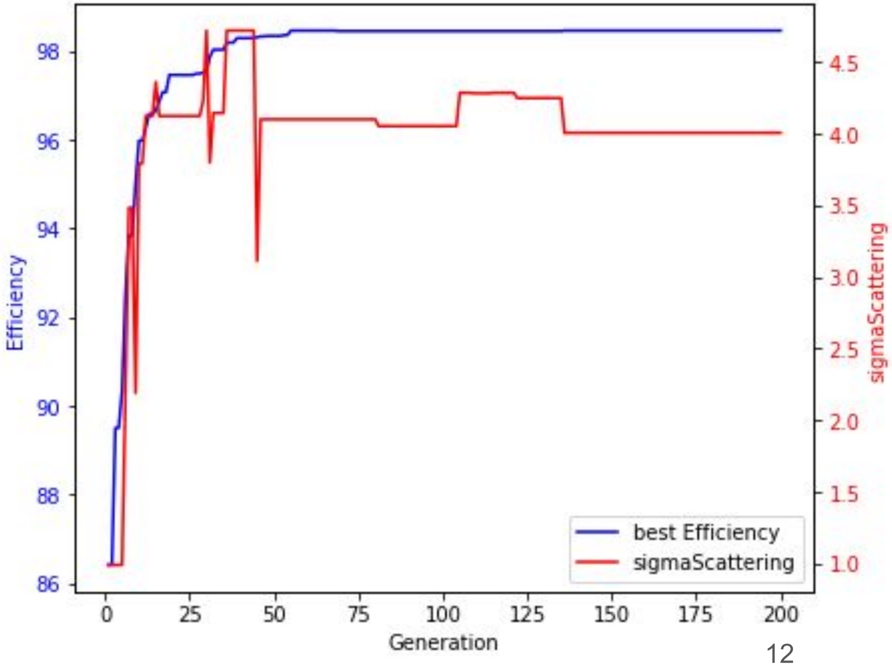




Evolutionary Algs Performance

Parameter values, efficiency and fake rate as a function of generation number


best DuplicateRate
maxSeedsPerSpM
DuplicateRate
maxSeedsPerSpM
Generation


best Efficiency
sigmaScattering
Efficiency
sigmaScattering
Generation