CKF Parameter Tuning Studies

Elyssa Hofgard, Rocky Bala Garg, Lauren Tompkins

Stanford University

Outline

- Problem summary/previous work
- Parameter tuning algorithms
  - Evolutionary algorithms
  - Lipschitz Optimization (LIPO)
  - Hyperparameter tuning framework (Optuna)
- Future work/discussion

Problem Summary

Combinatorial Kalman Filter (CKF) Track Reconstruction

- Kalman filter: discrete measurements to determine the internal state of a linear dynamical system
  - Combinatorial Kalman filter (CKF) technique combines track finding and track fitting
  - Starting track parameters estimated from reconstructed seeds used to steer the CKF

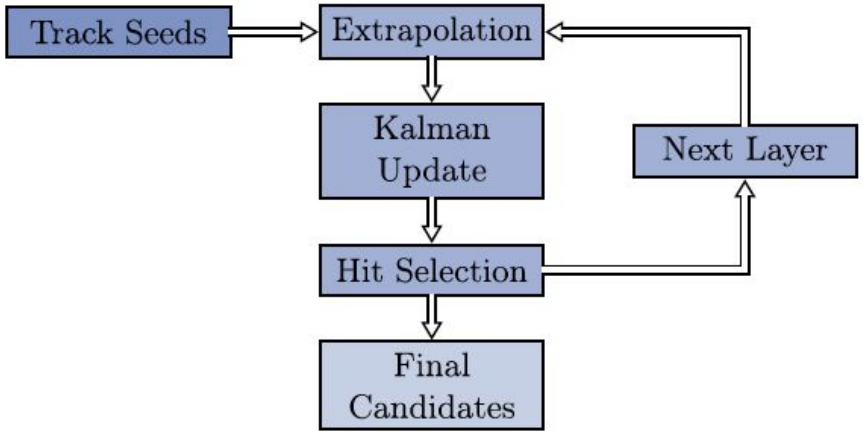


Figure from N.L. Braun, Combinatorial Kalman Filter and High Level Trigger Reconstruction for the Belle II Experiment. Ph.D. thesis, KIT, Karlsruhe (2019). https://publikationen.bibliothek.kit.edu/1000089317

Track Seeding

- Main performance bottleneck from track seeding step
- Seed comprised of three space points (x,y,z coordinates of detector hits) -> seeding algorithms loops over kinematically feasible combinations of hits
- Relies on configurable parameters to select potential space points -> these parameters can be optimized for algorithm performance
    - **Essentially a hyperparameter tuning problem**

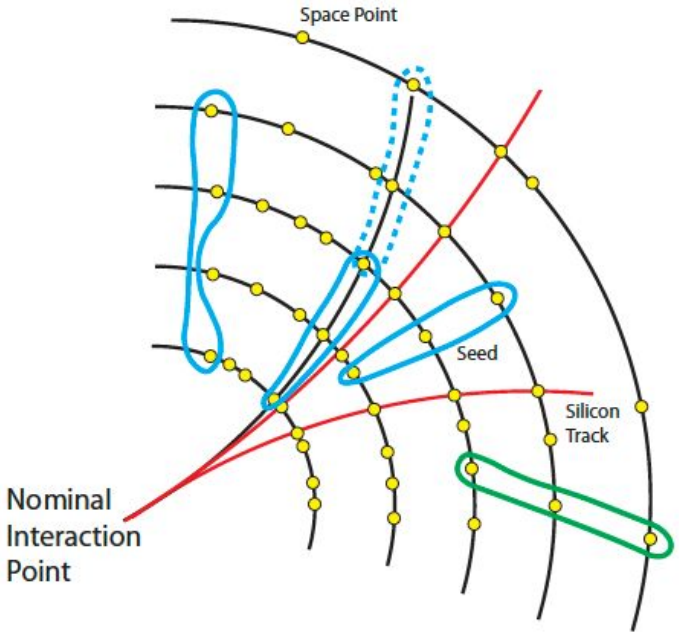


Figure from P. Chatain, R. Garg, L. Tompkins, Evolutionary Algorithms for Tracking Algorithm Parameter Optimization. (2021). https://doi.org/10.1051/epjconf/202125103071.

Previous Work and Goals

- Stanford group explored track seeding parameter optimization through evolutionary algorithms
- Identified most important parameters to optimize and good parameter configuration for seeding algorithm

Project Goals

1. **Extend previous work with evolutionary algorithms to CKF**
2. **Explore new algorithms for CKF parameter optimization and characterize the parameter space**

Configurable Parameters/Datasets

- Total of 22 configurable parameters in track seeding algorithm
- Currently [tuning parameters previously identified] from last study
- Generic Detector in ACTS
- Input dataset for tuning: sample of ttbar at √s = 14 TeV with 200 additional pile-up vertices per event (~2000 findable particles per event)
  - **Issue: need to run CKF for each parameter configuration, using 1 event for optimization algorithms**
  - 100 events for algorithm comparisons

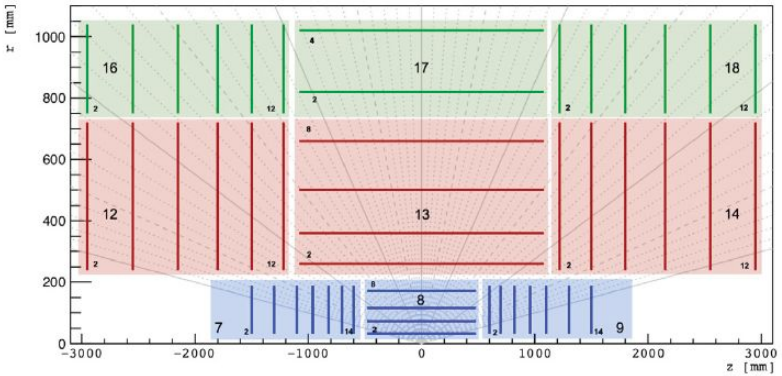


Figure from A Common Tracking Software Project. (2021). https://arxiv.org/abs/2106.13593.

Current Parameters Tuned

- **maxPtScattering**: maximum value of pT considered for multiple scattering consistency criterion
- **impactMax**: impact parameter = closest distance between interaction point and the path defined by the seed (discards seeds greater than impactMax)
- **deltaRMin**: min radius between two space points in a seed
- **deltaRMax**: max radius between two space points in a seed
- **sigmaScattering**: max standard deviations of multiple scattering angle when calculating compatibility of hits with trajectory
- **radLengthPerSeed**: average radiation length a particle traverses
- **maxSeedsPerSpM**: given middle space point, if multiple seeds exist, they are sorted based on how likely they are to be a good seed. Only the first n+1 seeds, n < maxSeedsPerSpM are chosen

Initial start, could explore parameter space further

Score Function*

Goal: high efficiency while keeping duplicate and fake rate low

- Track reconstruction efficiency: fraction of generated particles which have made at least 9 measurements on the traversed detectors and are associated with tracks
- Fake/duplicate rate: fraction of fake and duplicate tracks among all reconstructed tracks
- Want to control tradeoff between efficiency and duplicate/fake rate

*Note: used interchangeably with objective/cost function in this talk

Score Function for CKF

- **Previous score function** for seeding algorithm optimization (K = 1000)

$$\text{Efficiency} - \frac{\text{Fake} * \text{Duplicate}}{K}$$

- Need a **new score function** for CKF
  - Fake rate w/ CKF is extremely low (~0.05-0.33 %), doesn't make sense to multiply with duplicate rate
  - Adjust value of K for optimal tradeoff between efficiency and duplicate rate (K = 1,3,5,10), **optimal K = 3**

$$\text{Efficiency} - \frac{\text{Duplicate}}{3}$$

Optimization Algorithms

- Score function may not be differentiable or have a global maximum
  - Use information from evaluating score function at different parameter configurations to explore parameter space (**derivative-free optimization**)
- Extensive research done into potential derivative-free algorithms
  - For example: Rios and Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations", Journal of Global Optimization (2013).
- Explored 3 potential algorithms/frameworks
  - Evolutionary algorithms using Deap
  - LIPO Dlib package
  - Optuna

Evolutionary/Genetic Algorithms

Evolutionary Algorithms (EA)

- **Population**: each individual is one parameter set for the seeding algorithm
- **Each iteration (generation)**: run CKF on test dataset, calculate a score for each individual in population
- Selection function then chooses individuals such that better performing ones are more likely to stay
    - Some individuals selected for **mutation**, allows exploration of parameter space

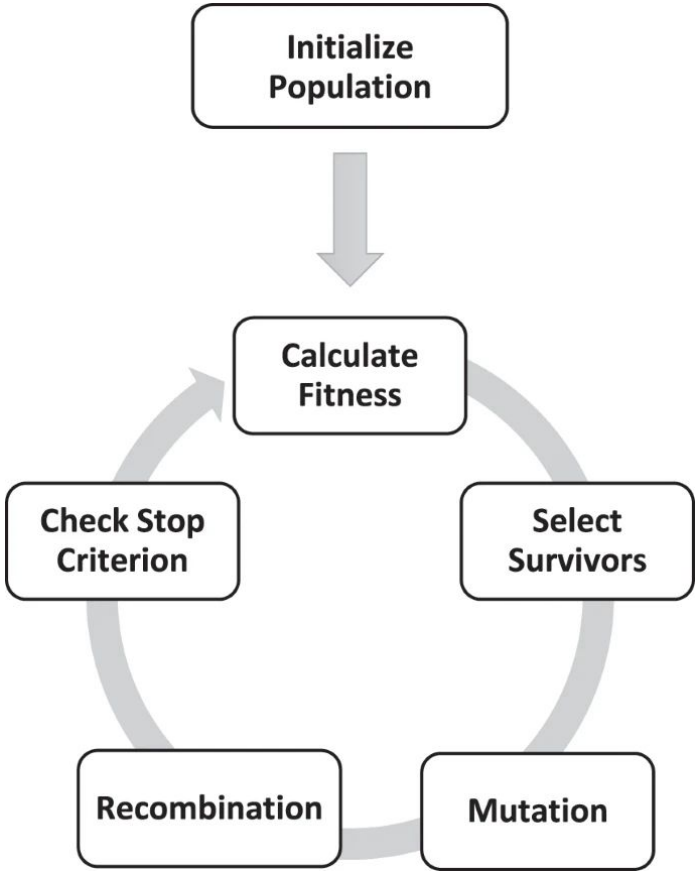
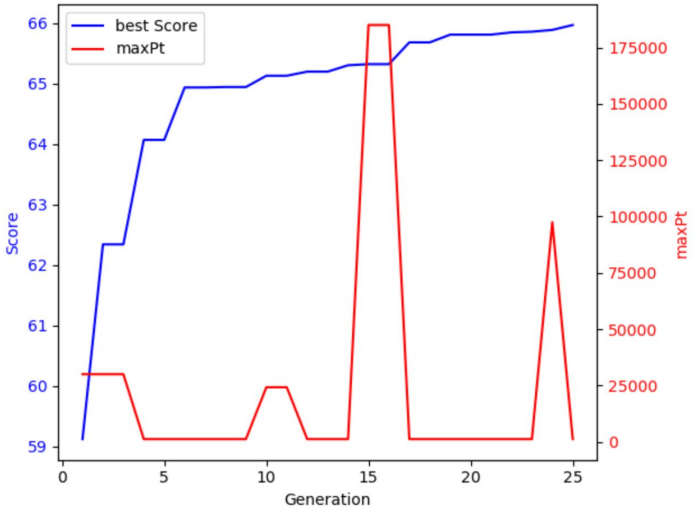


Figure from Distribution Optimization: An evolutionary algorithm to separate Gaussian mixtures. (2020). https://www.nature.com/articles/s41598-020-57432-w.

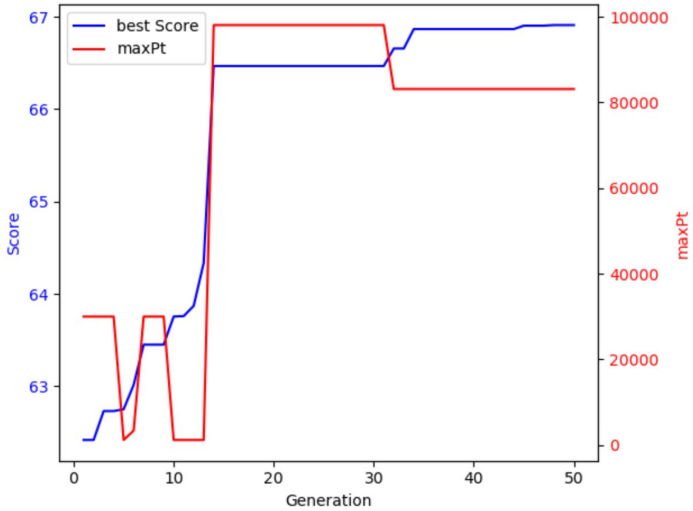
Previous research suggested that outcome of EA does not depend strongly on the parameters of the evolutionary algorithm itself (e.g., population size, number of generations, tournament size, mutation rate, etc.)

Not true in our case -> multiple optima

25 Generations



50 Generations



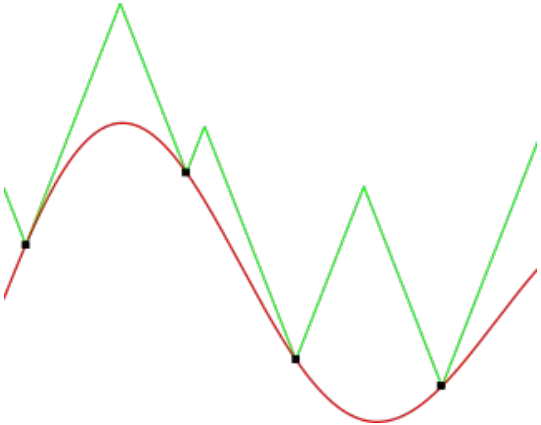
Lipschitz Optimization (LIPO)

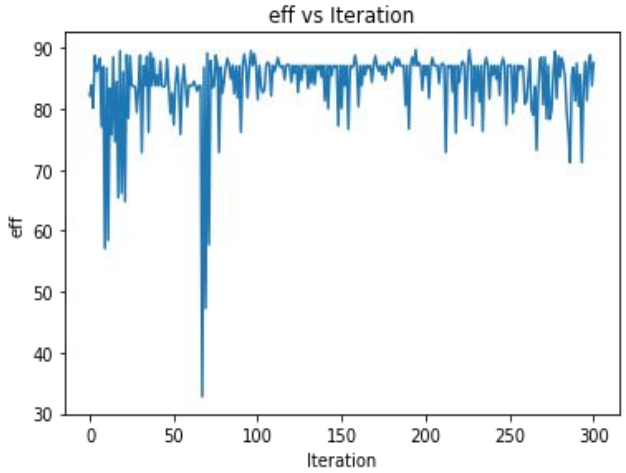
LIPO Overview

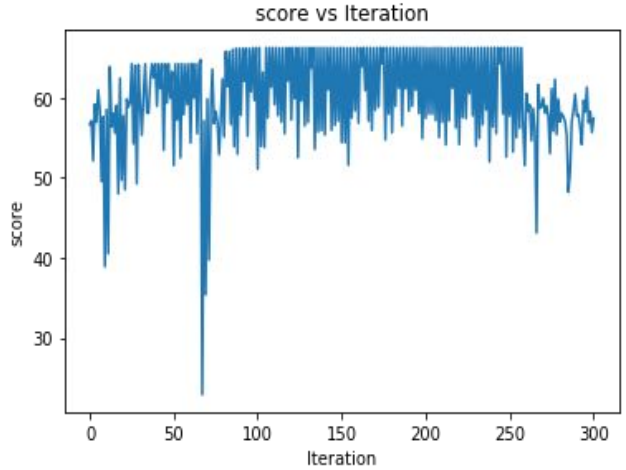
- Provably better than random search
- Maintain a piecewise linear upper bound of f(x) to decide which x to evaluate at each step
- Picks points at random, checks if the upper bound for the new point is better than the best point seen so far, and if so selects it as the next point to evaluate
- Relatively simple algorithm/doesn't have its own hyperparameters
- More in depth explanation

$$U(x) = \min_{i=1\ldots t} \left(f(x_i) + k \cdot \|x - x_i\|_2 \right)$$

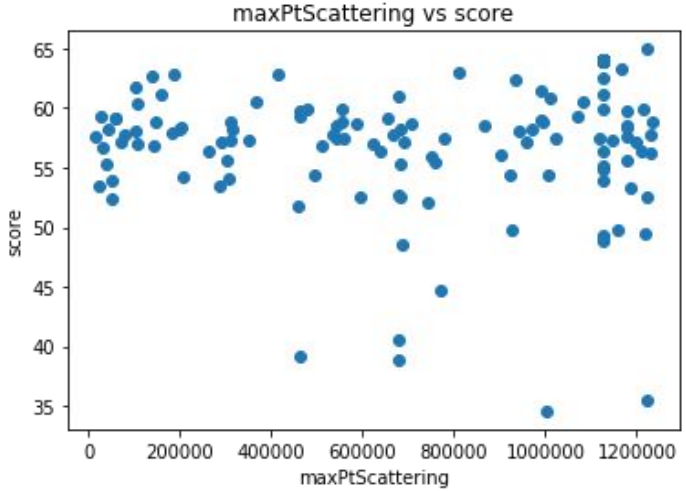
k=Lipschitz constant

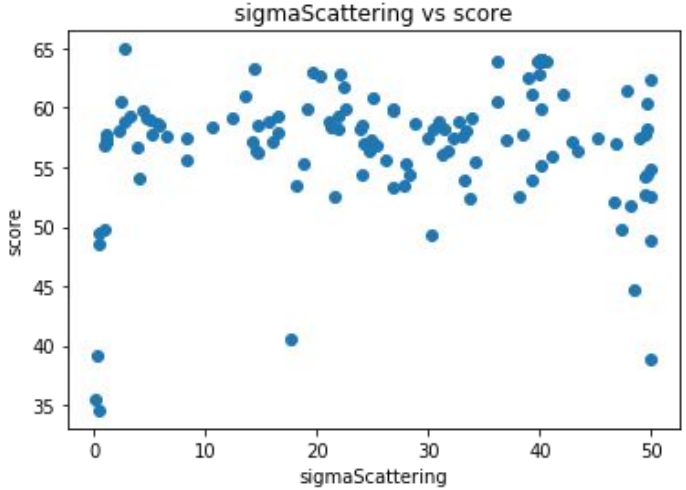






- Lack of convergence to “stable” score regardless of starting point or number of iterations
- Is this a problem with LIPO or indicative of ill-posed problem with many local optima?



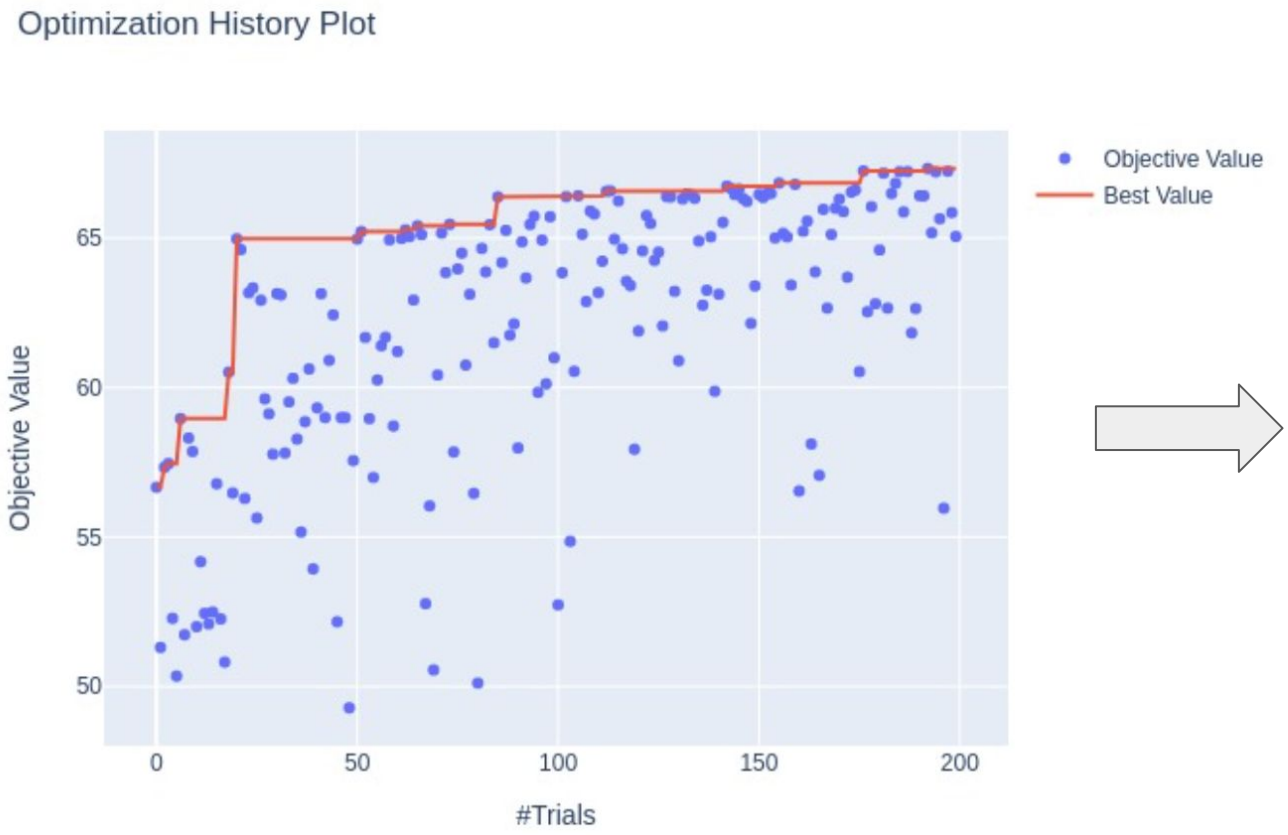


- No apparent pattern between high scores and variables of interest (held for other 5 variables)
- Motivated to try other optimization algorithms -> do we observe the same lack of convergence?

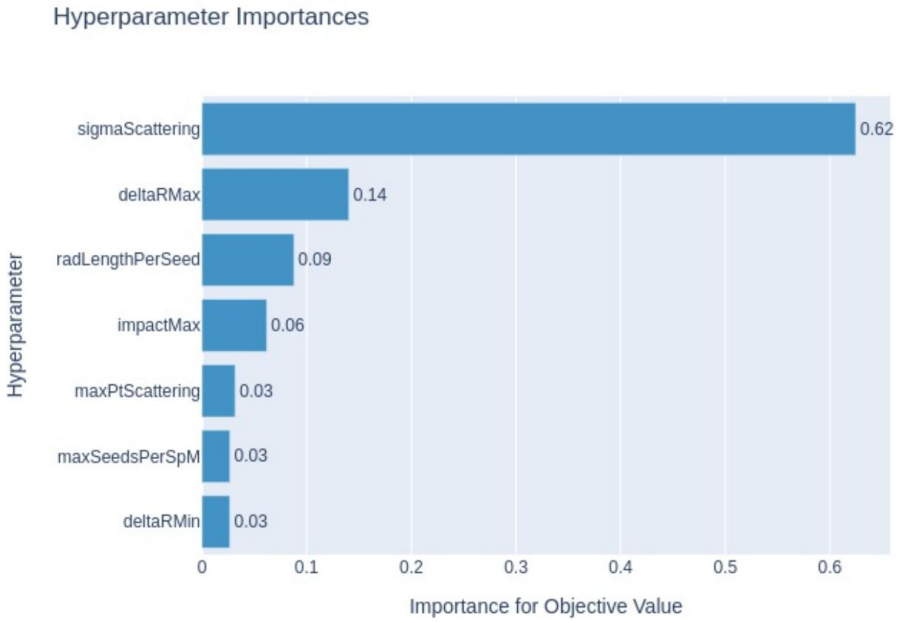
Optuna (Hyperparameter Tuning Framework)

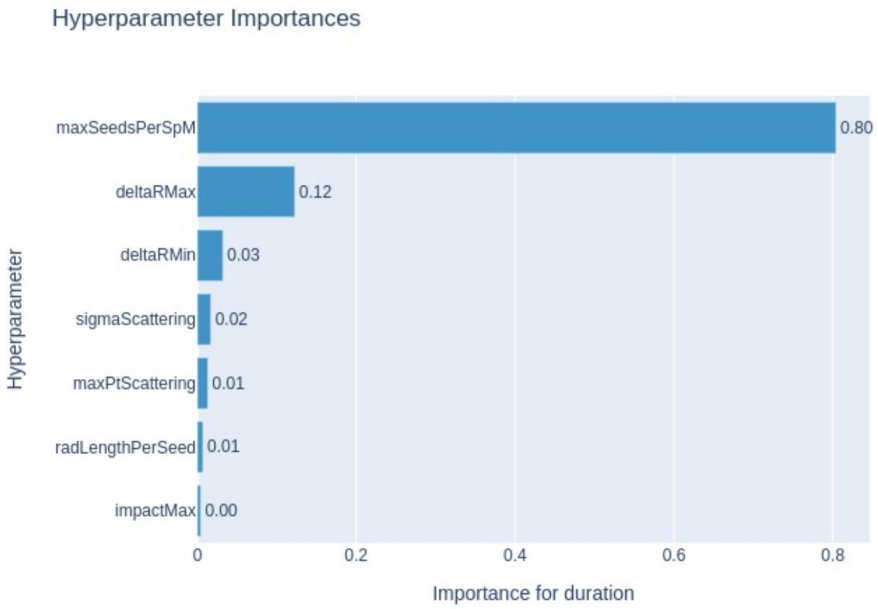
Optuna Overview

- A hyperparameter tuning problem, motivated to try prebuilt hyperparameter tuning frameworks
- Multiple approaches for sampling hyperparameters in Optuna, see [“Algorithms for Hyperparameter Optimization”]
- Default: Tree-structured Parzen Estimator (TPE)
  - On each iteration, for each hyperparameter
    - Divide previously seen hyperparameter values into subset associated with the best scores and subset of the remaining hyperparameter values
    - Estimate density $l(x)$ of best hyperparameter values and density $g(x)$ of others through fitting Gaussian Mixture Model (GMM) to each
    - Choose the hyperparameter value that maximizes $\frac{l(x)}{g(x)}$

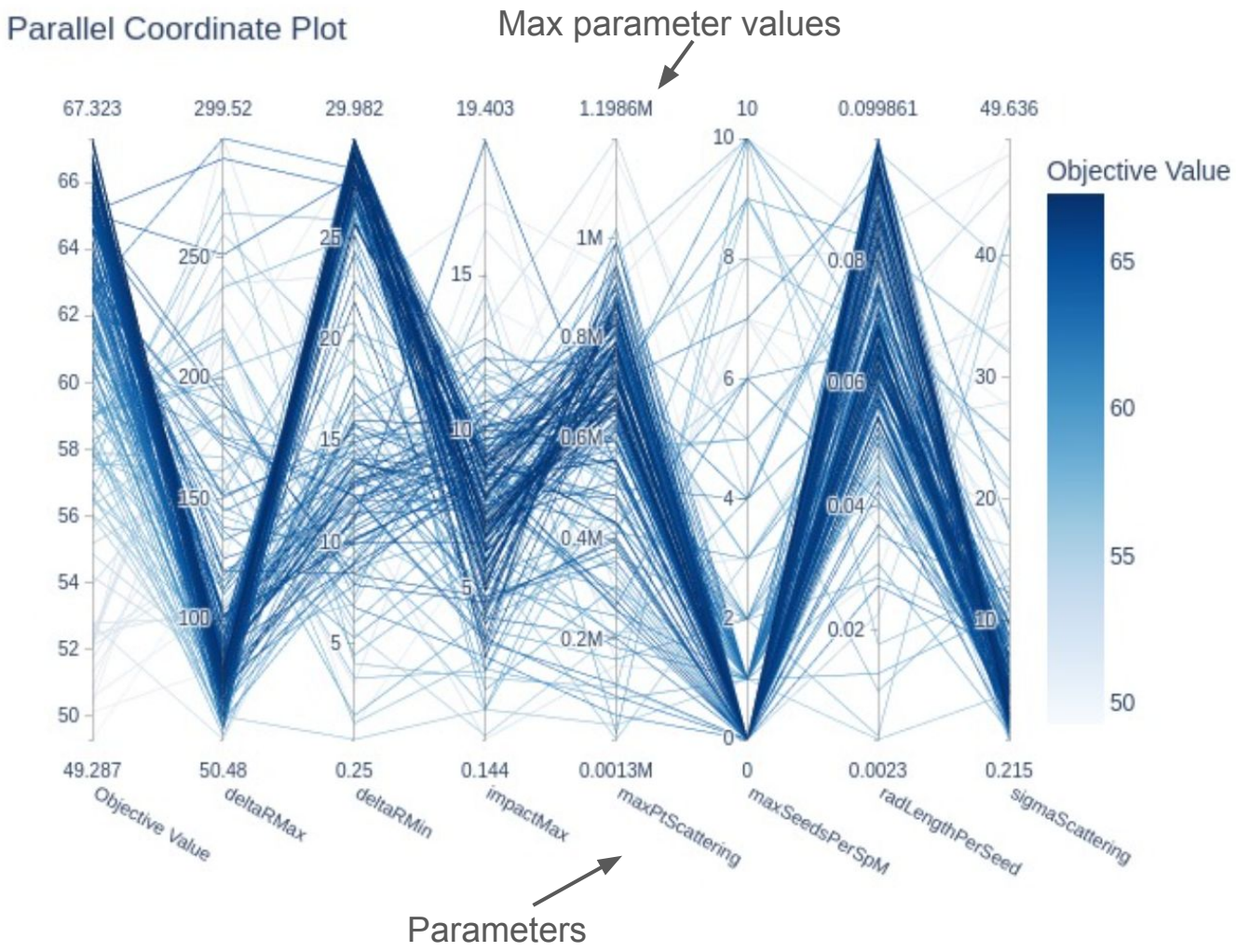


Clearer convergence than LIPO

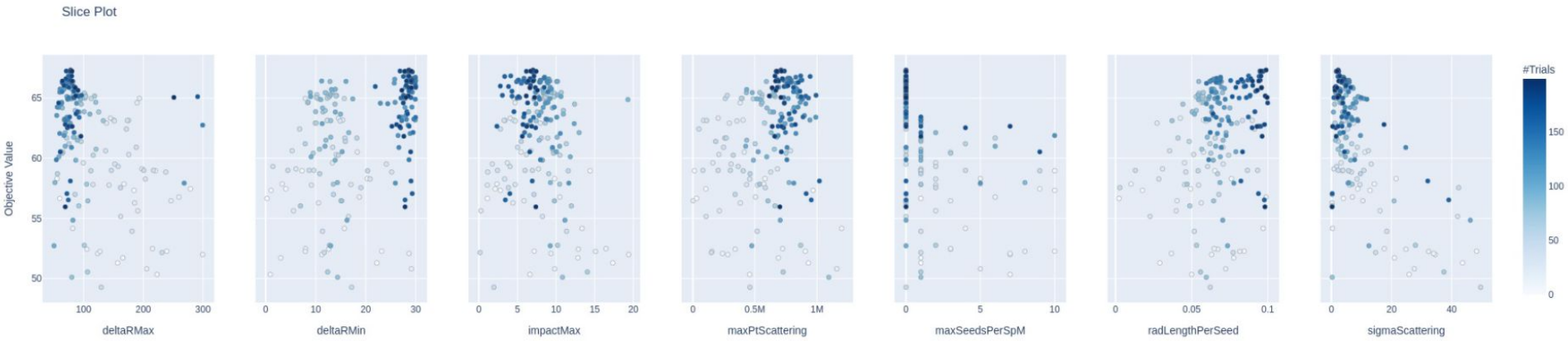




- Optuna uses fANOVA hyperparameter importance evaluation algorithm
  - Random forest regression model to predict the objective value given a parameter configuration, compute fraction of variance contributed by different hyperparameter subsets (details)



Trace out optimization path for different scores (objective values)



- Clearer convergence to higher score parameter values with increasing number of trials (quite different from LIPO)
- Optuna is easier to interpret/better convergence than LIPO in our case

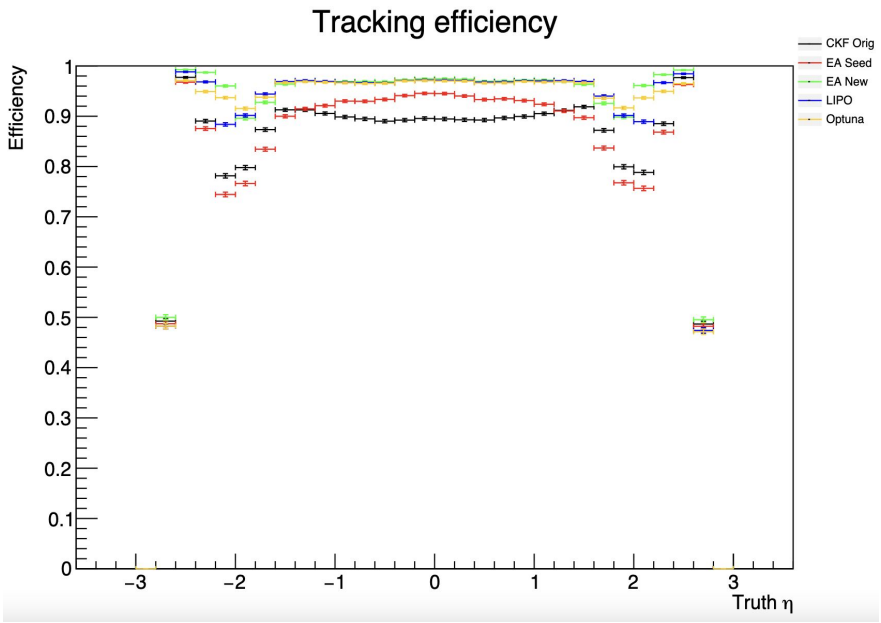
Algorithm Comparison

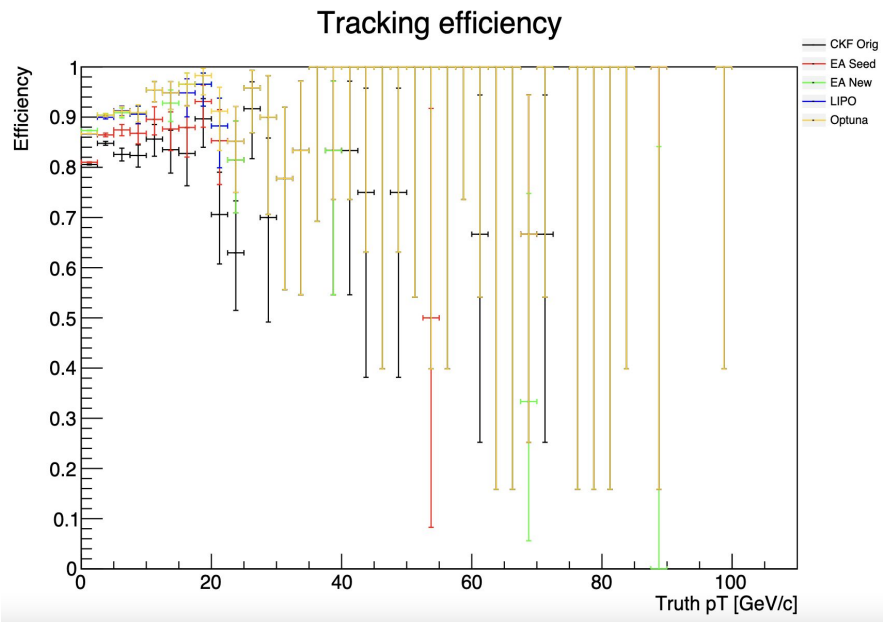
	CKF Default	EA Initial	EA New	LIPO	Optuna
Efficiency	0.807	0.812	0.875	0.868	0.869
Duplicate Rate	0.750	0.754	0.627	0.616	0.586
Fake Rate	0.000355	0.000183	0.000220	0.000257	0.000268
SeedingAlgorithm time/event (s)	0.116	0.082	0.085	0.064	0.093
TrackFinding time/event (s)	18.65	11.77	5.417	4.935	7.512

Similar efficiencies, Optuna reduced duplicate rate, difference in timing (?)

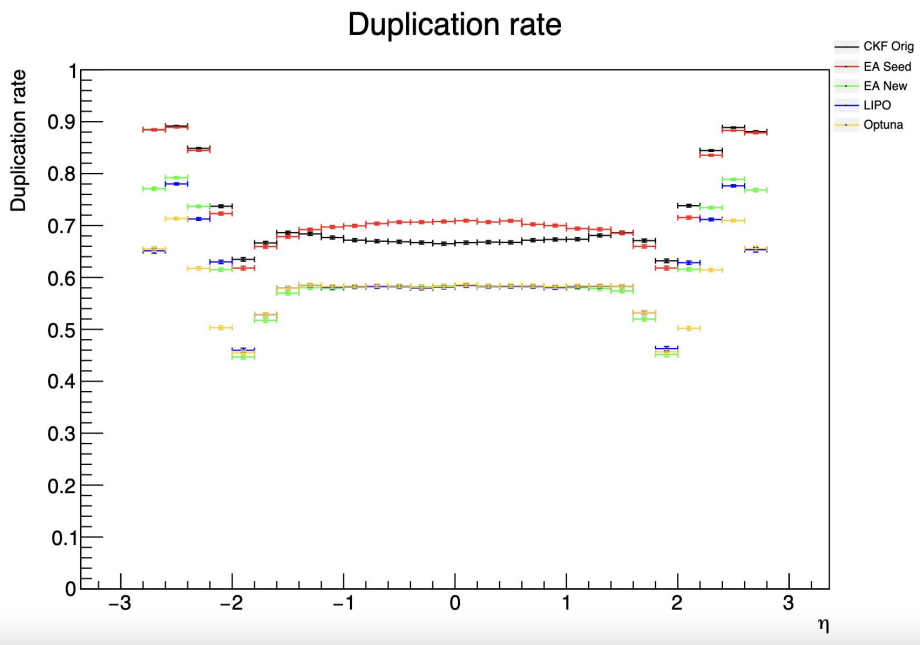
	CKF Orig	EA Seed	EA New	LIPO	Optuna
maxPtScattering (MeV)	10000	30000	83121.85	908873	710793.03
impactMax (mm)	3	1.1	12.43	19.69	7.01
deltaRMin (mm)	1	0.25	5.12	21.02	28.63
sigmaScattering	50	4.0	3.89	31.23	2.24
deltaRMax (mm)	60	60	78.39	66.24	78.16
maxSeedsPerSpM	1	1	0	0	0
radLengthPerSeed	0.1	.0025	0.0131	0.0010	0.0988

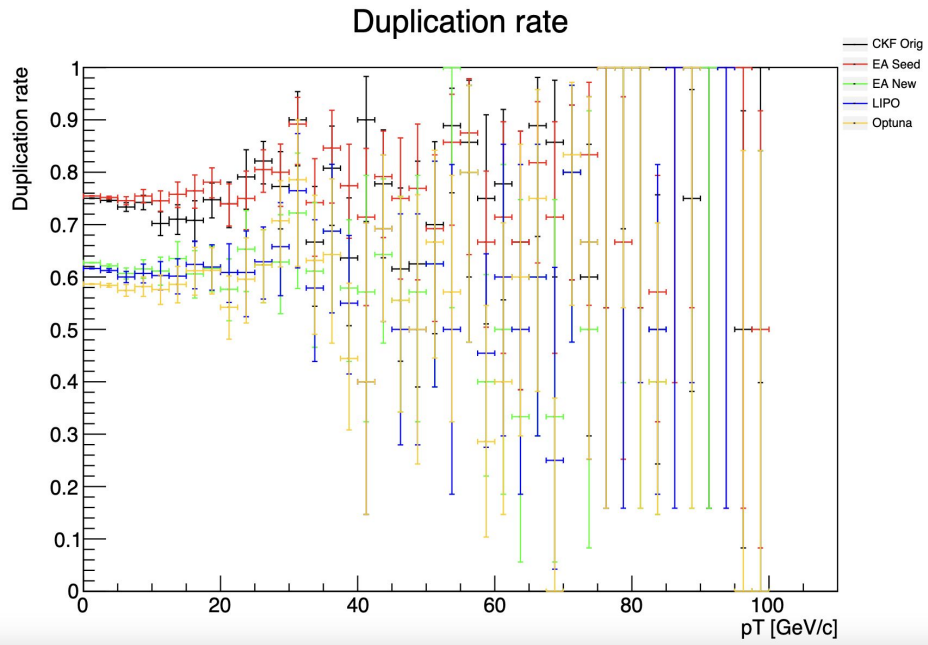
Each algorithm found quite different parameter configurations
Many possible configurations/how to choose?



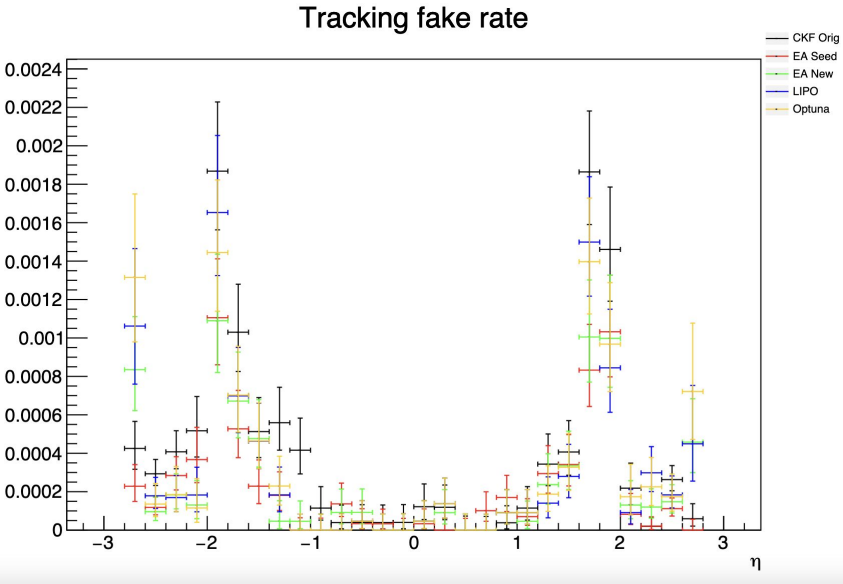


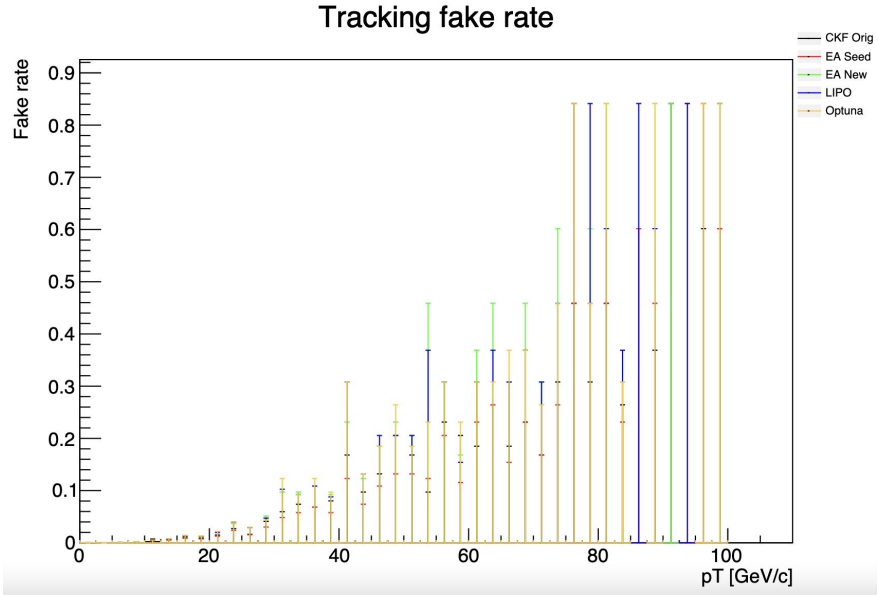
Similar improvements in efficiency across algorithms
Good improvement in forward region





Best reduction in duplicate rate in endcap with Optuna parameters





New parameters don’t have large impact on fake rate

Conclusion/Further Steps

Conclusions

- Compared three optimization algorithms for tuning seeding algorithm parameters for the CKF
  - Evolutionary algorithm
  - LIPO
  - Optuna
- Similar final performance with each despite differences in optimal parameter values
  - Multiple local optima
- Improvements in efficiency and duplicate rate for each
  - Best improvement in duplicate rate with Optuna

Further Steps

- Explore other optimization algorithms
  - nevergrad allows for simultaneous comparison of multiple derivative-free optimization algorithms
- Optimize parameters separately with η for barrel and encap of detector
- Switch to ITK detector geometry
- Questions/comments/suggestions for further steps?

Backup Slides

Previous Work

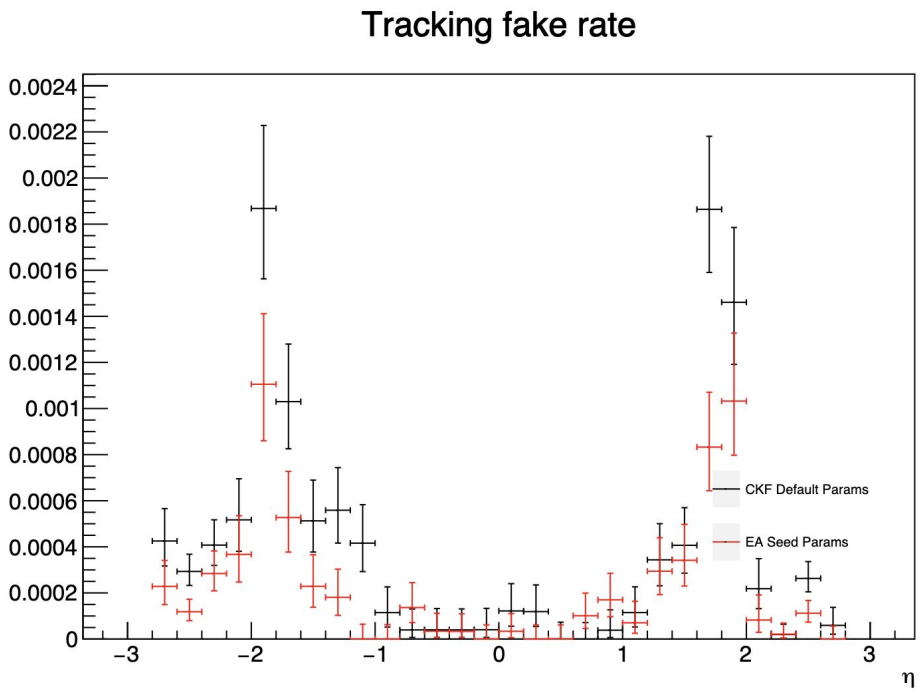
- Evolutionary algorithm used to optimize track seeding parameters in seeding algorithm

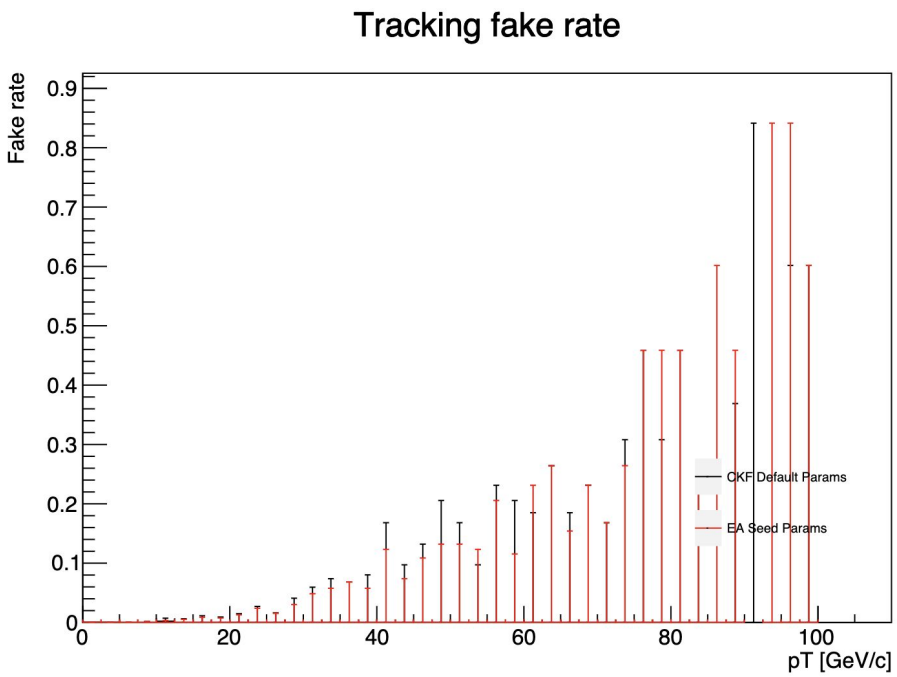
Variable	Generic Hand	Generic EA
maxPtScattering [GeV]	1200	30
impactMax [mm]	3	1.1
deltaRMin [mm]	1	0.25
sigmaScattering [σ]	4.0	4.0
deltaRMax [mm]	80.	60
maxSeedsPerSpM	2	1
radLengthPerSeem [λ]	0.005	0.0023

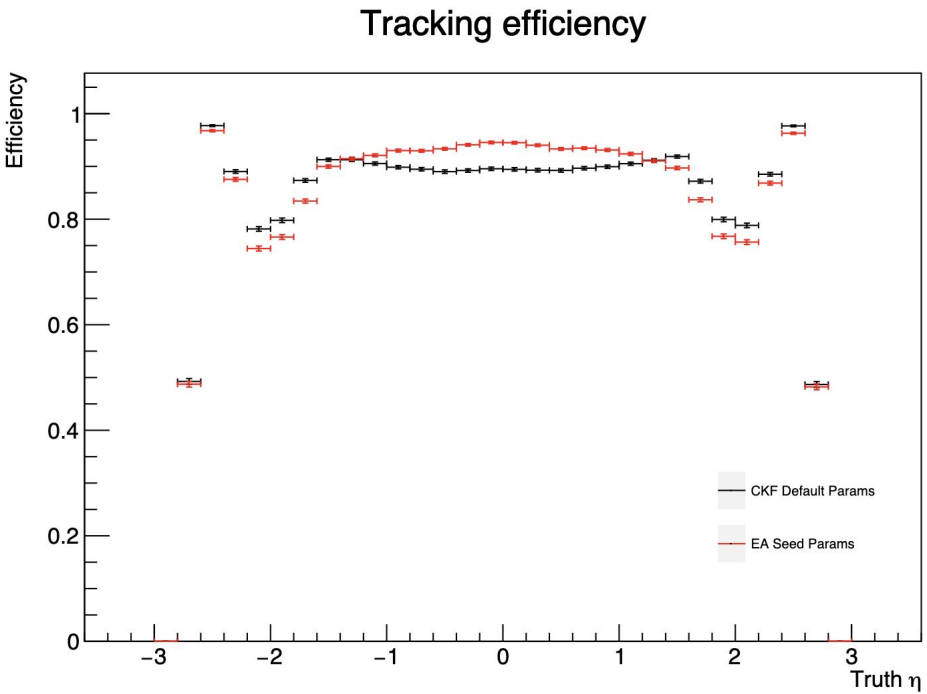
- **Q1: are the found EA parameters “good” parameters for the CKF?**

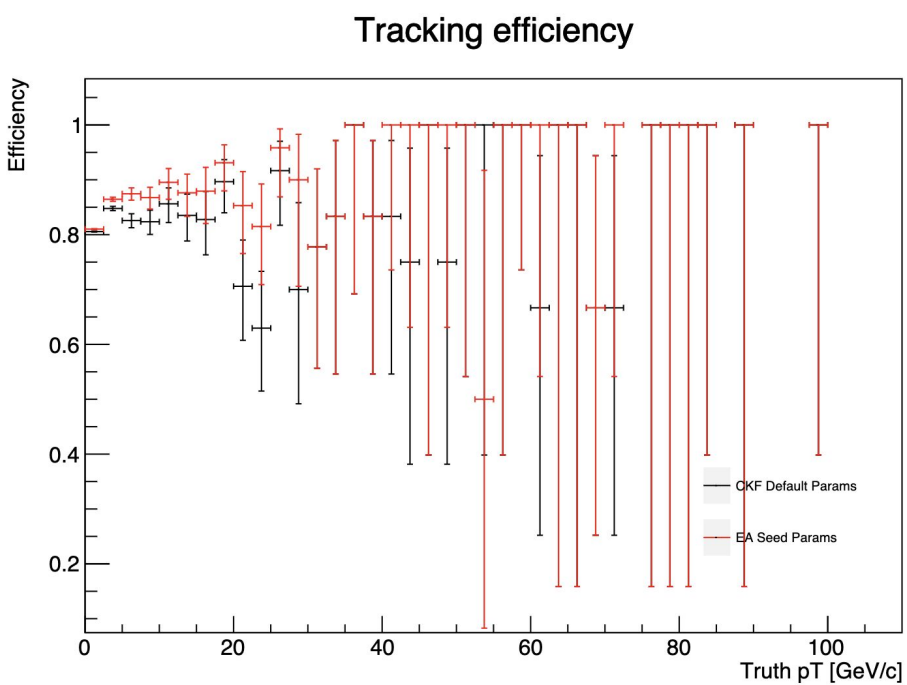
Default CKF Parameters Used

- rMax = 200 mm
- deltaRMin = 1 mm
- deltaRMax = 60 mm
- collisionRegionMin = -250 mm
- collisionRegionMax = 250 mm
- zMin = -2000 mm
- zMax = 2000 mm
- maxSeedsperSpM = 1
- cotThetaMax = 7.40627
- sigmaScattering = 50
- radLengthperSeed = 0.1
- minPt = 500 MeV
- bFieldInZ = 1.99724 T
- beamPos = {0_mm, 0_mm}
- impactMax = 3 mm
- maxPtScattering = 10000 MeV (default)

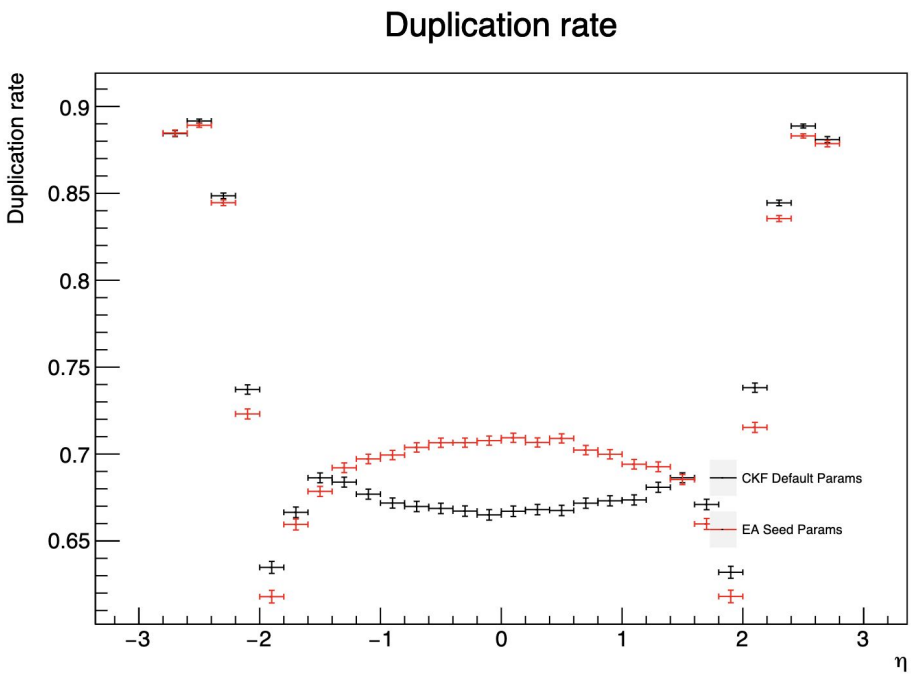

Tracking fake rate
0.0024
0.0022
0.002
0.0018
0.0016
0.0014
0.0012
0.001
0.0008
0.0006
0.0004
0.0002
0
−3
−2
−1
0
1
2
3
η
CKF Default Params
EA Seed Params

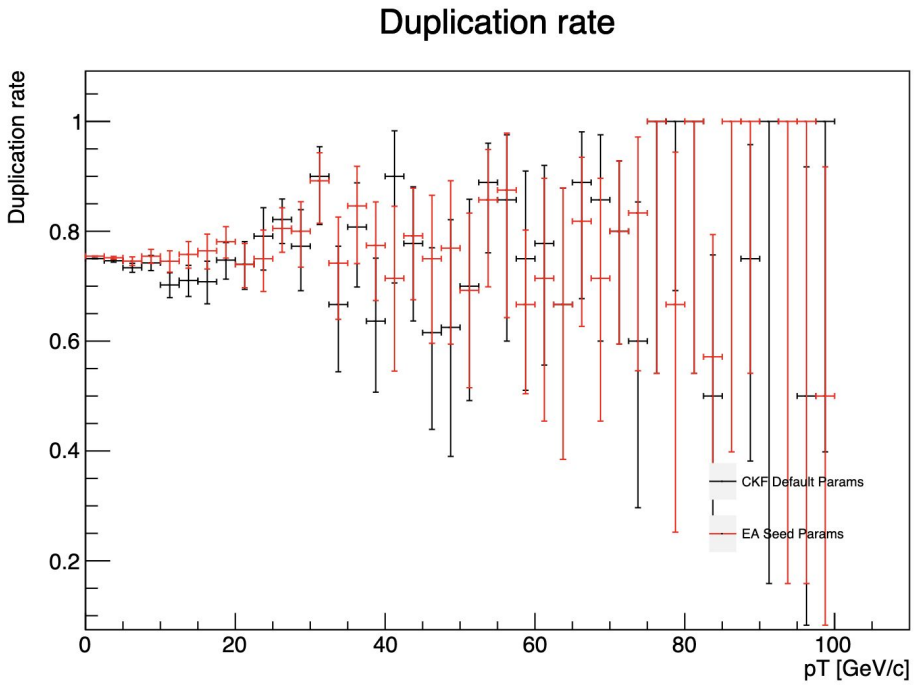

Tracking fake rate
Fake rate
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
pT [GeV/c]
CKF Default Params
EA Seed Params





Improvement in efficiency with previous EA parameters





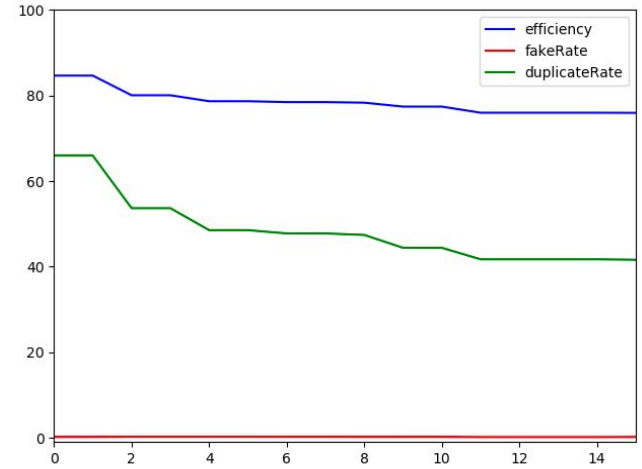
Trade-off between increasing efficiency and duplicate rate

Comparison of “Default CKF” to Old EA Parameters

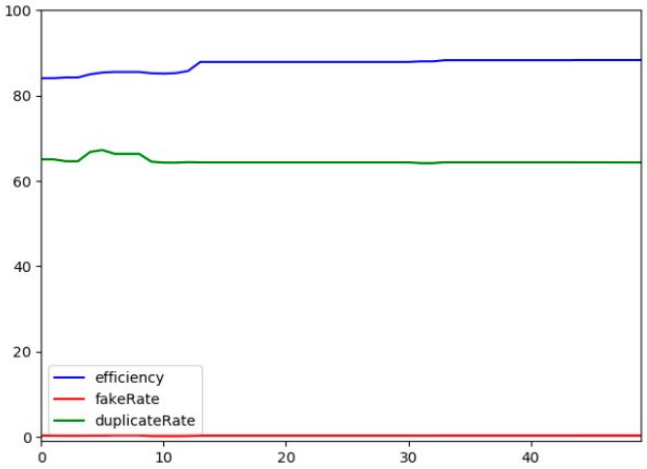
- Parameters currently in CKF for seeding are not tuned, use as a baseline for comparison, ttbar sample w/ 100 events

	Default CKF	Old EA Parameters
Efficiency	0.807102	0.812331
Duplicate Rate	0.749742	0.754407
Fake Rate	0.00035455	0.000183471
SeedingAlgorithm time/event (s)	0.1156	0.08229
TrackParamsEstimation time/event (s)	0.05605	0.04535
TrackFinding time/event (s)	18.65	11.77

K = 1



K = 3



Number of iterations

- Tradeoff between efficiency and duplicate rate with varying K (1, 3, 5, 10, 15)
- Best score found with K = 3
- Using K = 3 score function with other algorithms

$$\text{Efficiency} - \frac{\text{Duplicate}}{3}$$

EA Parameter Exploration