AI-Aided Kalman Filtering for Scalable Track Finding

Eric Kim
University of California, Berkeley
Mentors: Louis-Guillaume Gagnon, Rocky Bala Garg

May 2026

1 Project Description

Charged-particle tracking is one of the most computationally demanding reconstruction tasks in high-energy physics. In collider events, tracking algorithms must identify which detector hits were produced by the same charged particle and estimate the corresponding trajectory. In high-occupancy environments such as the HL-LHC, increased pileup produces many possible hit combinations, making the track-finding step increasingly expensive.

Kalman-filter-based tracking remains a standard framework for track reconstruction because it provides a structured recursive method for propagating track states, incorporating detector measurements, and estimating uncertainties [1, 2]. In ACTS, track finding can be performed using a Combinatorial Kalman Filter (CKF), which propagates track candidates from seed estimates through detector surfaces and searches for compatible measurements [3]. When multiple compatible measurements are found, the trajectory can branch into multiple candidates. In dense events, this candidate branching can lead to rapid growth in the number of track candidates, increasing both runtime and memory usage.

This project will investigate how neural network models can be integrated into different stages of Kalman-filter-based tracking algorithms, with the primary goal of improving the computational scaling and total computational footprint of the track-finding task. The focus is not to replace Kalman tracking with a black-box neural network. Instead, the project will study whether learned components can help guide candidate-hit compatibility scoring, hit-association ranking, branch pruning, or candidate termination within a CKF-style workflow.

The project will build on techniques discussed in the AI-Aided Kalman Filters paper [5]. That work surveys approaches for combining learned models with Kalman-type filtering, including learned preprocessing, learned correction terms, learned Kalman gains, and state-space-model-oriented methods. This project will investigate selected techniques from that paper and evaluate whether they can be adapted to the track-finding setting, where the main bottleneck is the combinatorial growth of candidate tracks.

Improving the precision or statistical consistency of the final track fit is an important secondary goal. In particular, the project may also investigate whether AI-aided methods can improve uncertainty estimates or filtering behavior. However, the central objective is to determine whether learned components can reduce candidate growth and runtime while preserving reconstruction quality.

Evaluation will focus on both physics and computational metrics. Physics-oriented diagnostics may include tracking efficiency, fake rate, duplicate rate, track-parameter residuals, pull distributions, uncertainty estimates, and overall reconstruction quality. Computational diagnostics will include runtime, memory usage where available, number of propagated candidates, branching behavior, scaling with occupancy, and the inference cost of neural-network components relative to the classical tracking baseline. The final goal is to understand whether AI-aided Kalman-filter techniques provide a practical path toward more scalable track finding.

2 Software Deliverables

The expected software deliverables are:

- An evaluation pipeline for comparing classical CKF-style Kalman track finding with AI-aided Kalman-filter approaches.

- A study of selected techniques from the AI-Aided Kalman Filters paper, adapted to candidate scoring, hit-association ranking, branch pruning, or candidate termination in track finding.
- Performance comparisons against classical Kalman-filter tracking, including tracking efficiency, fake rate, duplicate rate, residuals, pulls, runtime, candidate counts, and branch-growth metrics.
- Reproducible scripts in a Git repository for running experiments, generating plots, and comparing model variants.
- A short final report summarizing the investigated methods, observed performance, computational tradeoffs, and limitations.

3 Timeline

The project is planned for the summer research period, beginning in early July 2026. The exact model scope will be finalized with the mentors based on the existing ACTS-based tools, available tracking workflows, and the techniques selected from the AI-Aided Kalman Filters paper.

- **Weeks 1–2: Project setup and literature review.** Review the AI-Aided Kalman Filters paper, study the relevant ACTS/CKF tracking workflow, and define evaluation metrics for both reconstruction quality and computational scaling. Deliverable: finalized project scope and evaluation plan.
- **Weeks 3–4: Baseline evaluation pipeline.** Set up scripts for running or interfacing with the classical Kalman-filter tracking baseline. Define output diagnostics such as tracking efficiency, fake rate, duplicate rate, residuals, pulls, runtime, candidate counts, and branching behavior. Deliverable: working baseline evaluation pipeline.
- **Weeks 5–7: AI-aided model investigation.** Implement or adapt selected AI-aided Kalman-filter techniques inspired by the AI-KF paper. Focus on learned components that can plausibly improve candidate scoring, hit-association ranking, branch pruning, or candidate termination. Deliverable: first AI-aided tracking comparison.
- **Weeks 8–9: Performance comparison.** Compare AI-aided and classical Kalman-filter tracking under controlled occupancy or ambiguity settings. Evaluate the tradeoff between computational savings and reconstruction quality. Deliverable: plots and quantitative comparison of model performance.
- **Weeks 10–12: Finalization.** Clean up scripts, document the repository, summarize results, and prepare the final report and presentation. Deliverable: reproducible Git repository, final report, and presentation.

References

[1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, 1960.

[2] R. Frühwirth, "Application of Kalman filtering to track and vertex fitting," *Nuclear Instruments and Methods in Physics Research Section A*, 1987.

[3] X. Ai et al., "A Common Tracking Software Project," *Computing and Software for Big Science*, vol. 6, no. 8, 2022.

[4] G. Revach et al., "KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics," *IEEE Transactions on Signal Processing*, 2022.

[5] N. Shlezinger et al., "AI-Aided Kalman Filters," arXiv:2410.12289v3, 2025.

[6] S. Shlezinger Lab, "AI-Aided Kalman Filters," GitHub repository. https://github.com/ShlezingerLab/AI_Aided_KFs