BAYLOR
UNIVERSITY

Research Proposal for the 2023 U.S. CMS HL-LHC S&C Program

PI: **Kenichi Hatakeyama**, Associate Professor
Department of Physics, Baylor University

1 Introduction

This is my proposal for the 2023 U.S. CMS HL-LHC S&C program. The goal of the proposal is to **develop heterogeneous particle flow reconstruction for the CMS Phase 2 detector**. This goal would be well aligned with one of the priority areas of this program, *i.e.*, **algorithm development**.

The particle flow (PF) algorithm [1] plays a central role in CMS event reconstruction. The present PF reconstruction uses information from the tracker, ECAL, HCAL, and muon spectrometer to build a comprehensive list of final-state particles, called PF candidates. This list of PF candidates provides a global event description, which yields excellent performance for measuring jets, missing E_{T}, hadronic τ decays, electrons, photons, and muons.

Although PF reconstruction in CMS was established during Run 1, it needs significant updates as we upgrade our detectors. PF reconstruction for the Phase 2 detector needs to integrate high granularity calorimeter (HGCAL) reconstruction in the endcap region, incorporate forward tracks up to $|\eta| \sim 4$, and perform 4-dimensional reconstruction by incorporating the precision timing provided by e.g. the MIP timing detector (MTD) and HGCAL. It also needs to be able to cope with the increased data rates without loss of physics performance by efficiently exploiting the growing heterogeneous computing resources offered by GPUs and/or FPGAs.

The PI has spent the past few years on various aspects of PF developments for Run 3 and Phase 2, together with Dr. M. Saunders, a postdoc supported by this program from February 2021 to May 2022, his student (J. Samudio), and his collaborators. The recent developments include the PF rechit and cluster producer acceleration using GPU [2], the development of PF for the forward region ($3 \lesssim |\eta| \lesssim 4$) for Phase 2, and developments of "The Iterative CLustering" (TICL) for Phase 2 for the HGCAL region [3]. The CUDA-based software development for PF modules is basically complete, and now the transition to a portability library "Alpaka" is well underway. In the meantime, the PI has been a part of the team developing TICL reconstruction for PF reconstruction in the HGCAL region. TICL has been used as the default PF reconstruction software for the HGCAL region since the time of HLT TDR [4], and it has been improved over time [5]. The TICL is a modular framework developed for heterogeneous infrastructure that provides the particle shower reconstruction and particle flow candidate reconstruction, primarily developed for HGCAL, but it could work well for other calorimeter regions, and thus can provide the coherent reconstruction across all calorimeter regions.

With support by this program, the selected postdoc would **complete the transition of the CUDA-implemented PF modules to Alpaka**, and deploy it not only for use at the high level trigger (HLT) also for offline reconstruction. In addition, he/she will work on **further development of PF reconstruction for Phase 2, using TICL as a baseline, improve physics performance, and establish a coherent PF reconstruction across all calorimeters**.

2 Heterogeneous particle flow reconstruction

A simplified workflow diagram of the current PF reconstruction chain is shown in Fig. 1. First, a list of PF-specific rechits is created from rechits of each calorimeter subsystem, and it is used to produce calorimeter clusters. Then, a list of PF clusters as well as PF tracks and several other sets of information, called PF elements, are sent to the PFBlockProducer to put all linked elements in a PFBlock. Then each block is processed by PFAlgo to produce a list of PF candidates.

Fig. 2 shows the breakdown of processing time at HLT for data taking in Fall 2023 and for the Phase 2 offline processing of $t\bar{t}$ simulated events and 200 pileup interactions. Fig. 2(a) is not an estimate for HL-LHC operation, which is the main theme of this program; however, it illustrates current progress on the use of heterogeneous computing

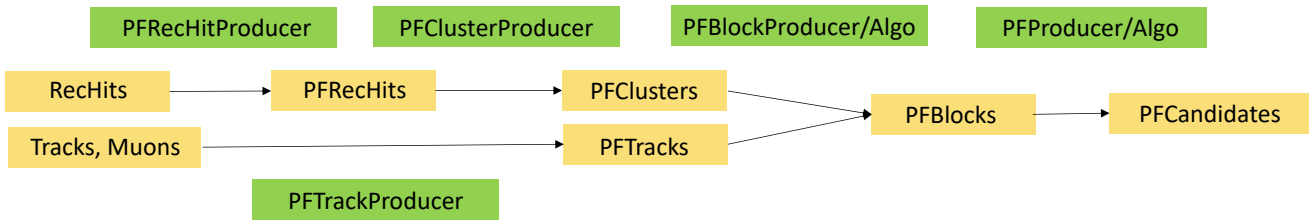

Figure 1: Simplified diagram of the particle flow reconstruction workflow.

resources for HLT. CMS introduced the heterogeneous HLT farm starting in Run 3 by equipping all $\sim$ 200 HLT nodes with 256 CPU threads and two GPUs, and pixel tracking and ECAL and HCAL local reconstruction software written in NVIDIA CUDA have been run on GPUs. The natural next target to be ported to GPU is PF reconstruction, as many inputs to PF are reconstructed on GPU.

For the Phase 2 PF reconstruction, we currently use a hybrid approach in which TICL is used for the HGCAL region and the traditional PFAlgo in the other regions. For Phase 2 offline, the PF reconstruction (RecoParticle-Flow+RecoHGCal) takes about 12% of the total processing time as shown in Fig. 2(b), which is not a major portion; however, this number could easily increase as we try to improve physics performance further. We need a conscious effort to keep this percentage well under control while improving physics performance.

2.1 Porting the early steps of PF to GPU

Among several steps of PF reconstruction shown in Fig. 1, one of the most time consuming steps is **PFCluster-Producer**, which groups calorimeter hits to form clusters corresponding to showers created by incident particles. Clustering is a common exercise in HEP, but it is often not trivial to parallelize.

A Baylor ex-postdoc Dr. Mark Saunders worked on this project from February 2021 until May 2022 and produced an alpha version of the GPU accelerated algorithm for PFRecHitProducer and PFClusterProducer. He had to leave the position due to a health reason, but my Ph.D. student (J. Samudio) and another student from DESY (F. Lorkowski) took over the project with support from myself and scientists from CERN and PSI (F. Pantaleo, M. Rovere, M. Missiroli). The recent improvement to the implementation includes adaptation of parallel-operation-friendly connected-component labeling algorithm, ECL-CC, and optimization in Structure-of-arrays (SoA) data structure. The preliminary results were presented at ACAT2022, and the GPU based PF clustering has shown a

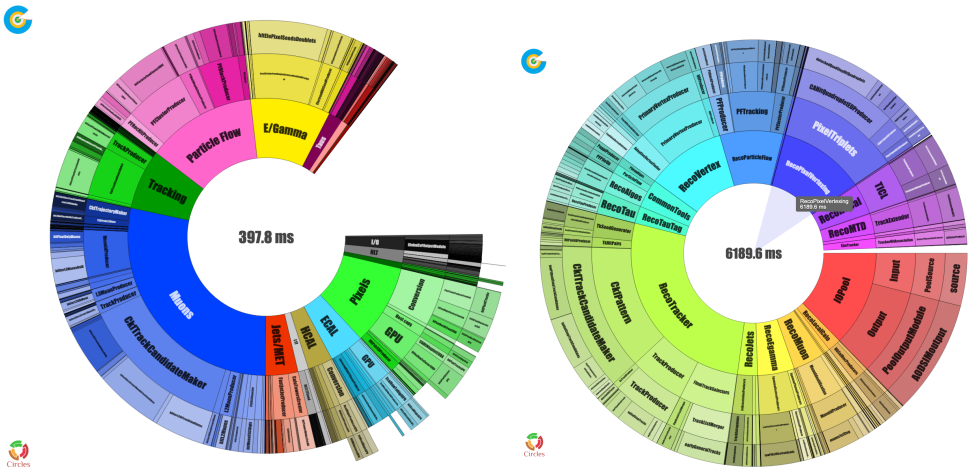

Figure 2: (a) Processing time profile of . (b) Processing time profile of the Phase 2 offline reconstruction scenario with $t\bar{t}$ events with 200 pileup interactions.

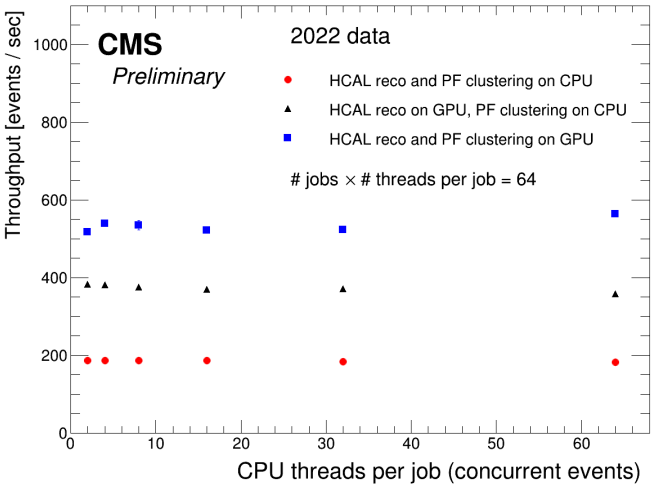

Figure 3: The throughput in events per second for three workflows running the HCAL local reconstruction and PF reconstruction, either on CPU or on GPU, measured using Run 3 2022 data: (red) both HCAL local reconstruction and PF clustering on CPU, (black) HCAL local reconstruction on GPU and PF clustering on CPU, and (blue) both HCAL local reconstruction and PF clustering on GPU. Measurement details are described in Ref. [2].

factor 3 improvement over the CPU implementation, when 1 GPU and 64 CPU cores are used for comparisons.

Since then, we solved some data hazard in an earlier implementation, optimized the memory consumption, and then focused on migration of this implementation from NVIDIA CUDA to the portability library Alpaka, as Alpaka was chosen as a portability library choice by CMS at least for Run 3 operation. Our team worked on this development in February Patatrack Hackason, and will be finishing it up in Summer 2023. So, when the postdoc to be supported by this program starts, it's likely that this part of the project is at a fairly advanced stage, but some work could be still left for commissioning for both HLT use and offline use. The selected postdoc will be able to help in that phase and get familiar with PF reconstruction, CMS HLT and offline workflow, and various testing and validation.

2.2 TICL development for the HGCAL region

For Phase 2 PF reconstruction, we use TICL as a baseline approach for the HGCAL region since the time of HLT TDR, and it has been optimized over time. My group has been a part of this TICL development team. More details of TICL can be found in e.g. Refs. [3, 5]. In the most recent major release TICLv4 [5], one of the major changes is in the "Pattern Recognition" module, which is the core of the TICL framework, and it reconstructs 3D clusters (called tracksters) from the 2D clusters. In TICLv4, the Pattern Recognition module moved to a CLUE3D algorithm which tracks the energy flow of showers using the capabilities of CLUE [6] in a more detailed manner than the earlier Cellular Automation version.

The TICLv4 shows good performance for photon reconstruction with pileup up to 200; however, the performance for charged hadrons are still marginal as shown in Fig. 5. The efficiency drops toward lower energies, and the pileup has a significant impact on the reconstruction performance of hadrons, especially in the high η region where the detector occupancy is very high. Optimizing the pattern recognition and linking algorithm more specifically for hadrons is one of the priority items for TICL development, and it has been the area the Baylor group ahs been asked to look into. It would be a good area for the selected postdoc to look into. In addition to algorithm approach, machine

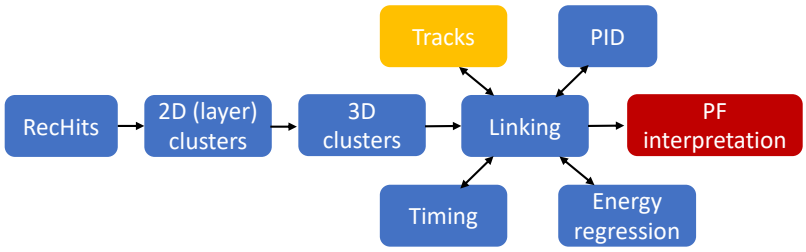

Figure 4: Illustration of the building blocks of the HGCAL TICL reconstruction workflow.

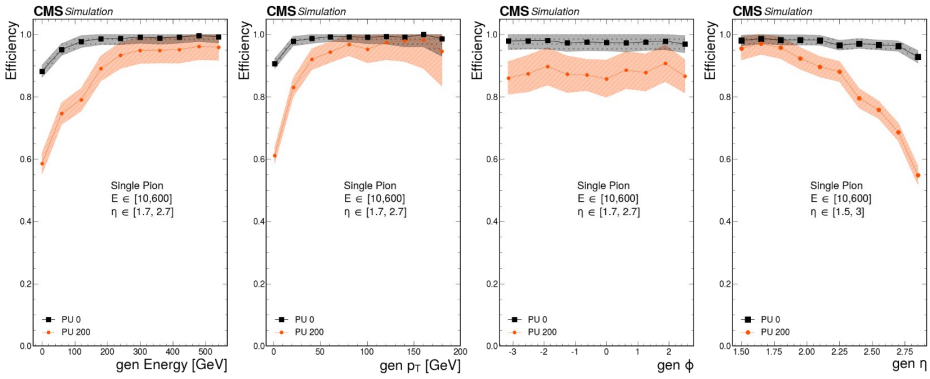

Figure 5: The reconstruction efficiency of the merged trackster collection obtained after the linking procedure compared between 0PU and 200PU, for energy, p_{T} , ϕ and η, for charged pions.

Learning methods, in particular Graph Neural Networks, can be exploited for this task, building up the experience from MLPF [7].

2.3 Porting TICL modules to Alpaka

As already written earlier, the TICL framework has been developed targetting the heterogeneous infrastructure. In fact, for some components of TICL such as CLUE and CLUE3D, there are standalone implementations in both CUDA and Alpaka; however, they are not integrated into CMSSW. The first step of this integration would be to define the proper SoA data format for layer clusters and tracksters, and then the remaining integration should be straightforward. This is another area where the selected candidate can work on.

2.4 Coherent PF for the entire Phase 2 detector

As discussed earlier, the current Phase 2 PF reconstruction consists of two almost separate paths for inside and outside the HGCAL detector acceptance. This hybrid approach has been fulfilling what we need now for Phase 2 R&D studies, and performs ok as shown in e.g. Fig. 6. However, for real Run 4 data reconstruction, there has been a strong desire to establish a more coherent PF reconstruction across all calorimeters when the HGCAL local reconstruction TICL becomes mature using TICL as a baseline. We aim to develop new pattern recognition and linking algorithms for the barrel reconstruction, in order to extend the TICL framework, exploiting barrel information together with the information coming from the tracker and timing detectors. In addition, we will need to develop separate patter recognition and linking algorithms for the hardon forward (HF) calorimeter region as well, as we need to link tracks in the extended forward region to HF clusters. Data structure's design and code maintainability will benefit from the use of the same framework within barrel, endcap, and HF regions.

The project already started with a Ph.D. student Alessandro Brusamolino from U. Hamburg. As an initial step, he is studying layer clusters in the barrel region using the CLUE algorithm, which is used for HGCAL reconstruction. The main limitation of CLUE is that it creates binary clusters without allowing cluster overlaps, which is not a significant limitation for finely-segmented HGCAL; however, in more coarsely segmented ECAL and HCAL barrel and HF, particle showers naturally overlap more significantly, and extension may be necessary for CLUE, or we may directly adopt the current PF clustering as inputs to the TICL's linking algorithm.

For the HF region, as an initial step, the traditonal PFAlgo was extended so that it can link tracks to HF clusters. However, this linking and track propagators to HF need more refinement, and we aim to make the planned refinement within TICL framework by incorporating HF clusters to TICL.

The plan for the postdoc to be supported by this program is to priotize the hadronic shower reconstruction improvements, linking optimization for charged hadrons, and their PF interpretations; however, he/she would help Alessandro and other people in this developmental work as well.

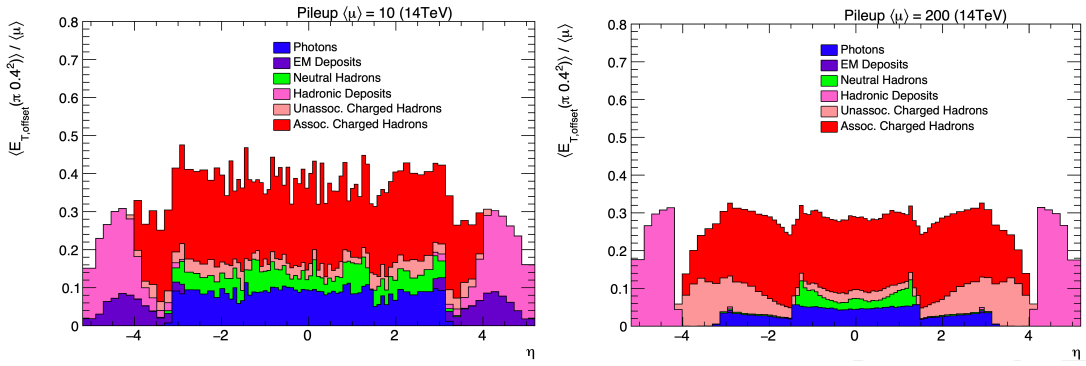

Figure 6: The normalized PF candidate energy density distributions vs detector-η with 0PU (left) and 200PU (right) in the Phase 2 scenarios.

3 Timeline and mentorship plan

The tentative timeline and the milestones of the planned work for the postdoc are presented below:

December 2023: finalize commissioning of Alpaka PF rechic/cluster producers for HLT and offline workflows

February 2024: identify main contributing factors for suboptimal performance of hadron reconstruction in TICL

June 2024: first iteration of optimization of hadron shower reconstruction in TICL

October 2024: first iteration of optimization of linking and energy regression for charged hadrons in TICL

The postdoctoral candidate for the proposed project is not identified yet. I plan to hire a postdoc who is proficient in programming in C++ and python, and in addition some experience with CMS reconstruction software developments and with CUDA and/or Alpaka will be preferred (but not required) given the planned work. I can directly supervise him/her, together with my Ph.D. student (Samudio) and my collaborators from CERN and PSI. The selected postdoc will be suggested to attend GPU coding camp opportunities such as the Patatrack Hackathons and/or parallel programming courses offered elsewhere (e.g. the course like this one).

4 Summary

In summary, the particle flow algorithm plays a central role in CMS event reconstruction, and its upgrade for Phase 2 is of paramount importance for the CMS experiment's success during HL-LHC. I think the proposed work on the development of the heterogeneous particle flow reconstruction for Phase 2 has significant impacts on the U.S. CMS HL-LHC S&C program and the CMS Collaboration as a whole. Thank you for your consideration.

References

[1] A. M. Sirunyan *et al.* (CMS Collaboration), JINST **12**, no. 10, P10003 (2017), arXiv:1706.04965.

[2] F. Pantaleo *et al.* (CMS Collaboration), CMS-CR-2023-043, ACAT2022.

[3] A. Di Pilato *et al.* (CMS Collaboration), JINST **15**, no.06, C06023 (2020), arXiv:2004.10027.

[4] CMS Collaboration, CERN-LHCC-2021-007; CMS-TDR-022.

[5] F. Pantaleo *et al.* (CMS Collaboration), CMS-CR-2023-036, ACAT2022.

[6] Z. Chen *et al*, arXiv:2001.09761.

[7] F. Mokhtar *et al*, arXiv:2303.17657, ACAT2022.