IRIS-HEP Fellowship Proposal

Packaging Jet Substructure Observable Tools

Jordan Ashley
University of Tennessee, Knoxville

May 2024

Project Details

Introduction

Historically, the particle physics community has forged the knife's edge of data science and its applications. Maintaining that edge, however, requires a separate, dedicated effort. Two widely applicable jet substructure observable tools—EnergyFlow and Wasserstein—have lost much of their value due to a lack of maintenance. This project will modernize the packaging and distribution of these tools, pulling their efficiency and insights in line with the advances in modern computing architectures, packaging tools, and sustainable software developments.

Review

The Wasserstein library applies a modified version of the network simplex algorithm to efficiently calculate Wasserstein distances—a metric for measuring space between probability distributions [2]. The library is written primarily in C++, with a SWIG-enabled, NumPy-based Python wrapper. Despite the library's efficient and valuable modified implementations of algorithms from the Python Optimal Transport library, it is no longer actively maintained. The original authors have left the project in the care of a community organization, but the complexities of upgrading the Python packaging for modern CPU architectures (e.g. aarch_64), combined with nonexistent release procedures, have blunted recent attempts at much-needed fixes [3], compounding problems and hindering progress.

EnergyFlow, an even larger library that integrates numerous tools for analyzing and visualizing the substructure of particle jets [1], suffers from a similar situation. The EnergyFlow library is pure-Python, avoiding complications with packaging compiled extensions, but its dependency requirements and lack of packaging infrastructure introduce their own complications.

By taking this on as a single large project, I will be able to overcome many of the obstructions that have prevented progress so far through consistent, dedicated attention. The original authors constructed many useful features into these packages, which have been fostered and—to the best their availability has allowed—faithfully maintained by community caretakers. I intend to study their

work intently, learning from their bug fixes and organizational choices so that my upgrades maintain consistency and integrate smoothly with prior efforts.

Deliverables

The primary goal of this project is to modernize the build systems and Python packaging infrastructure of the Wasserstein and EnergyFlow libraries, resulting in new releases of the libraries to the Python package index (PyPI) and conda-forge. Those updates will include modernizing and changing build systems to implement compatibility upgrades, designing and initiating new continuous-integration and continuous-delivery (CI/CD) systems, and releasing upgraded Python distributions to public package indexes.

This research will be completed under the mentorship of Dr. Matthew Feickert (University of Wisconsin-Madison) and Dr. Henry Schreiner (Princeton University).

Timeline

Tentative start date: May 20th, 2024

Weeks 1-2

Research essential project tools, including the organization of both project repositories, Scientific Python development best practices, scikit-build-core documentation, and CI/CD workflows. Set up PyPI account.

Week 3

Attain fluency in project tool usage by assembling sample builds and implementing fundamental pieces of the prior stage's knowledge base to structurally mimic Wasserstein and EnergyFlow.

Weeks 4-6

Develop, document, and implement a well-defined plan to upgrade Wasserstein to achieve the primary goals of designing and implementing a scikit-build-core/pybind11 build, modernizing the CI/CD system, and releasing updates to PyPI.

Week 7

Finalize release and update the Wasserstein documentation. Review and analyze the EnergyFlow repository to identify potential pain points based on my experience with Wasserstein.

Weeks 8-9

Refocus on EnergyFlow, transitioning to a similar build and CI/CD system as with Wasserstein, and preparing releases for PyPI.

Week 10

Finalize the EnergyFlow release and update any associated documentation. Gain a nuanced understanding of conda-forge best practices.

Week 11

Package and distribute both Wasserstein and EnergyFlow via conda-forge.

Week 12

Finalize documentation updates for both Wasserstein and EnergyFlow, including demo updates and Read the Docs sites. Condense findings and results into a final presentation.

References

[1] Patrick T. Komiske and Eric M. Metodiev and Jesse Thaler. *EnergyFlow.network*. 2020. Retrieved April 24, 2024, from https://energyflow.network/.

[2] Patrick T. Komiske and Eric M. Metodiev and Jesse Thaler. *The Metric Space of Collider Events*. 2019. arXiv: 1902.02346 [hep-ph].

[3] Release procedure for Wasserstein. 2023. Retrieved April 24, 2024, from https://github.com/thaler-lab/Wasserstein/issues/7