Accelerating Particle Physics Simulation with GPUs

Siddharth Sule with Michael H. Seymour

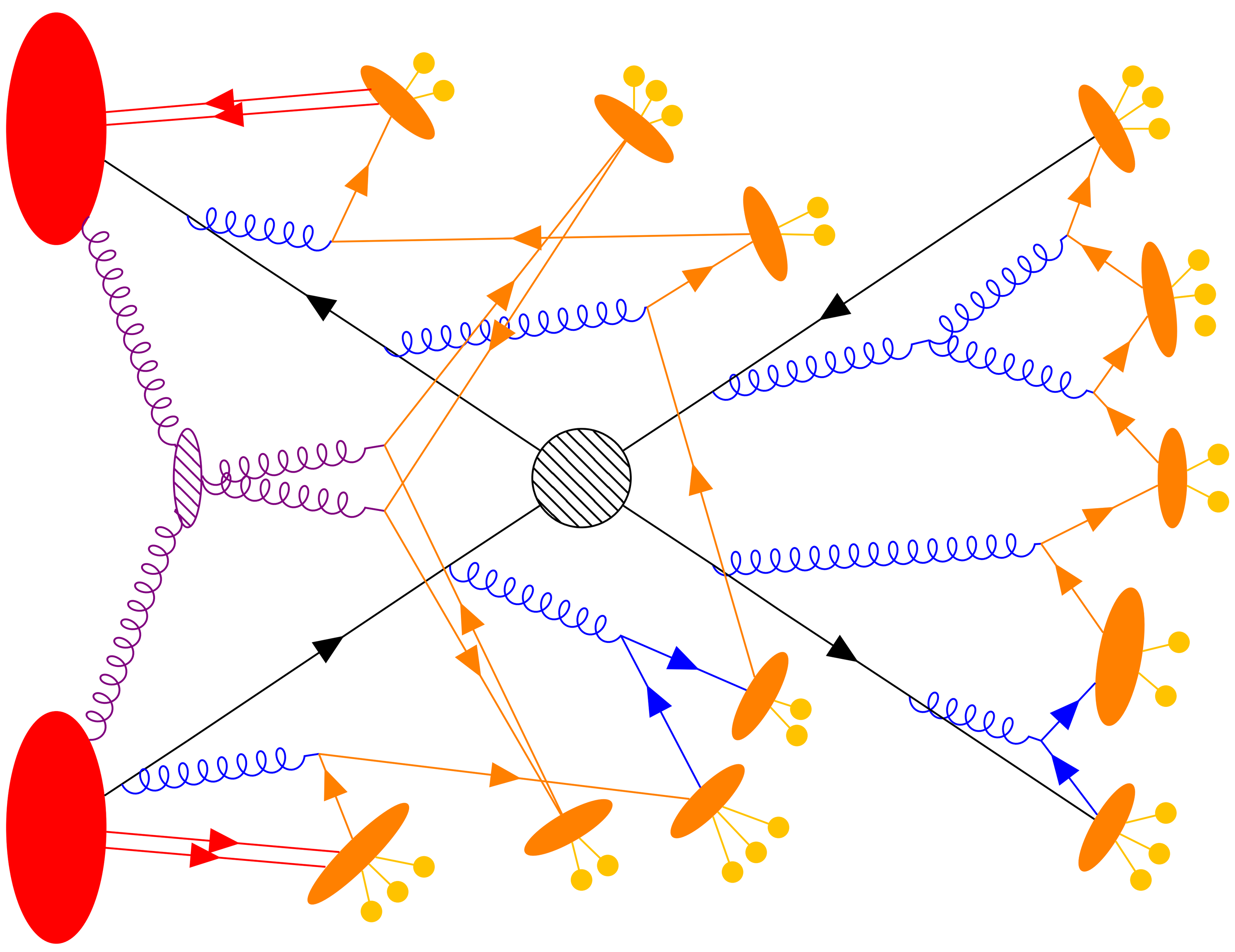
MANCHESTER 1824
The University of Manchester

Abstract

Particle Physics Simulation accounts for 17% of CPU Usage at CERN [CERN-LHCC-2022-005]. Recent progress has been made to reduce this usage by accelerated computing on GPUs [2311.06198, 2303.18244]. We contribute to this growing area of research with a GPU algorithm for simulating emissions from gluons and quarks.

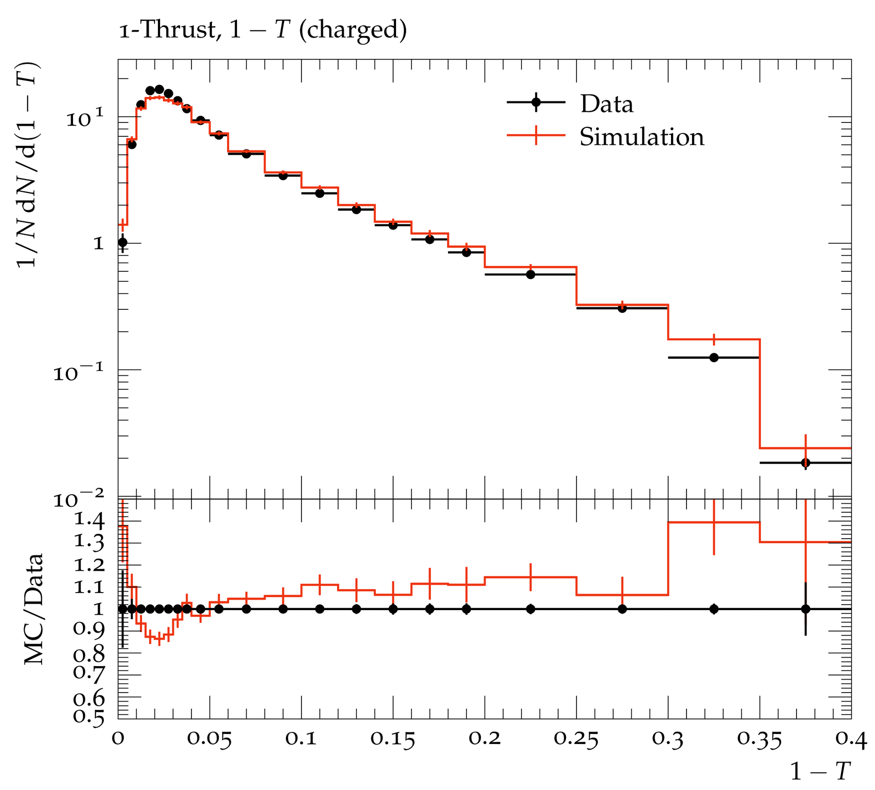
Monte Carlo Event Generators

A Particle Collision at the LHC involves several processes: the **fundamental interaction** is accompanied by emissions, underlying interactions and confinement of quark and gluons. These processes connect the colliding protons to the detected particles.



Particle Collisions are simulated using Monte Carlo Techniques and Object-Oriented Programming in a software known as an *Event Generator*. The simulation results are compared to detector data to:

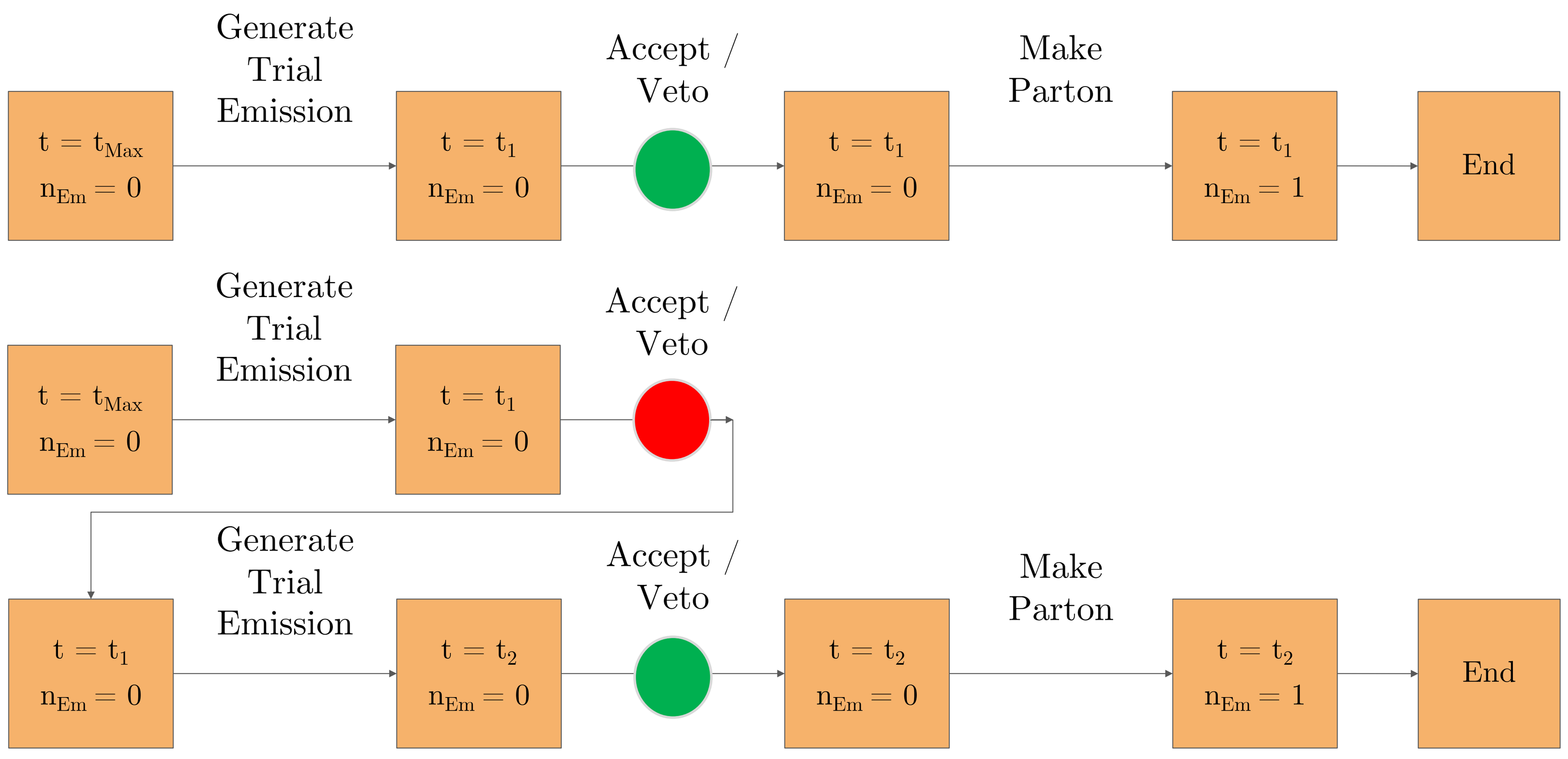
- **Identify** signs of new physics
- **Test** new theories and models
- **Forecast** future detectors



Simulating Emissions: Parton Showers

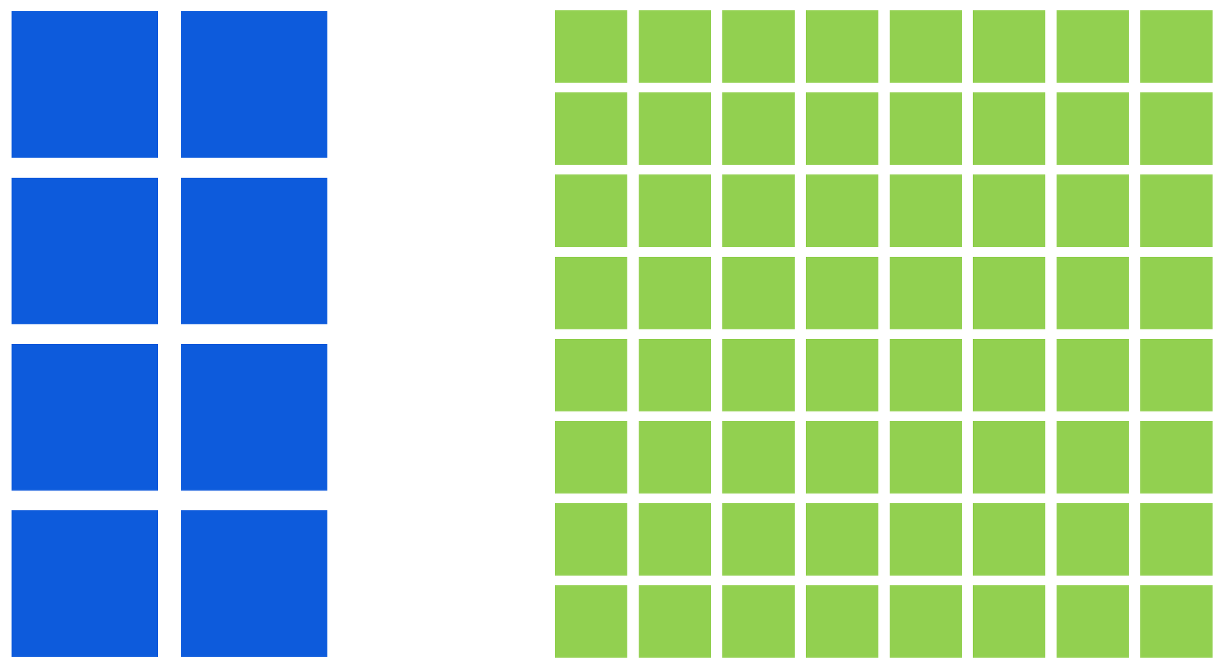
Quarks and gluons (collectively called *partons*) behave like free particles at high energies. Besides interactions, they also release energy by emitting additional quarks and gluons. This chain reaction of emissions is known as a *parton shower*. At such high energies, perturbation theory can be applied to calculate the probability of emission to occur, $\mathcal{P}$, depending on the scale of the momentum transferred in the process, t.

Parton Showers are simulated using the **Veto Algorithm**. Here, a trial emission at scale t is generated and accepted with the emission probability $\mathcal{P}(t)$. If the emission is rejected/vetoed, a new trial emission is generated at a lower scale ($t' < t$). Emissions occur until the generated value of t reaches the cutoff for perturbation theory (usually around the QCD confinement scale, Λ_{QCD}.



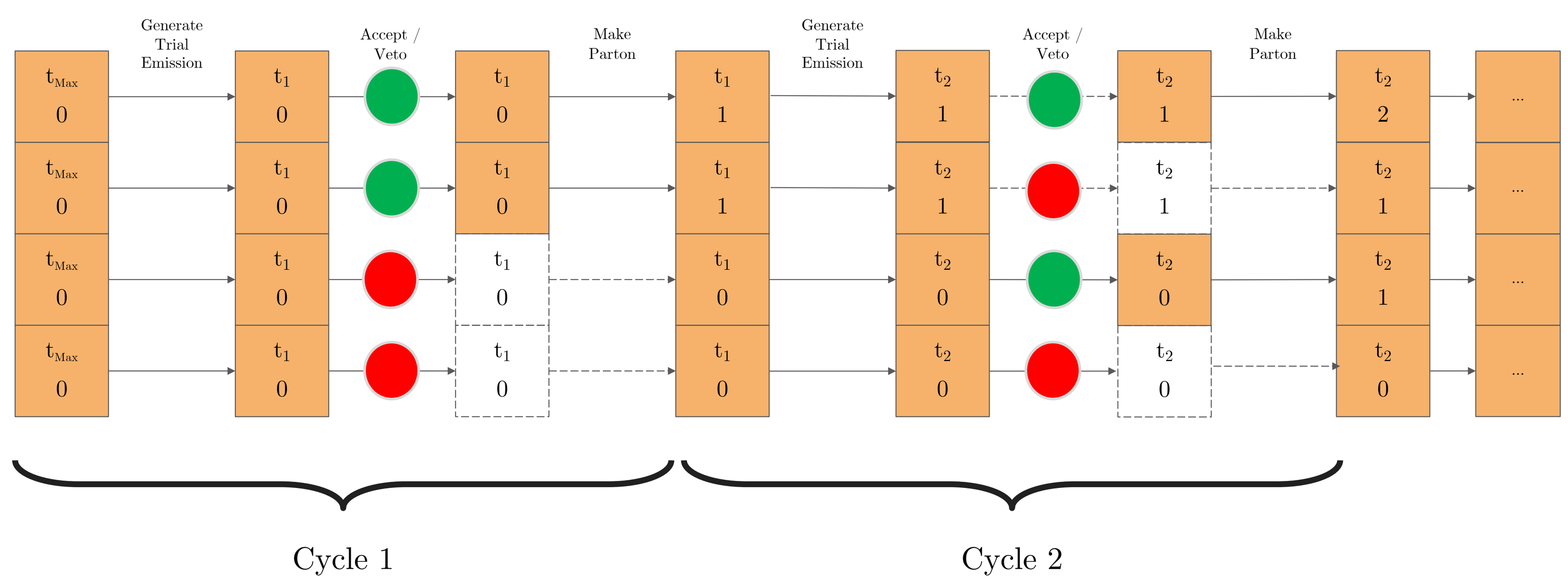
GPUs and Parallel Programming

While CPUs are built with a single or handful of large, powerful cores, GPUs opt for hundreds to thousands of small, minimal cores. This feature allows developers to execute the same function with different parameters in parallel, such as simultaneously manipulating elements of an array.



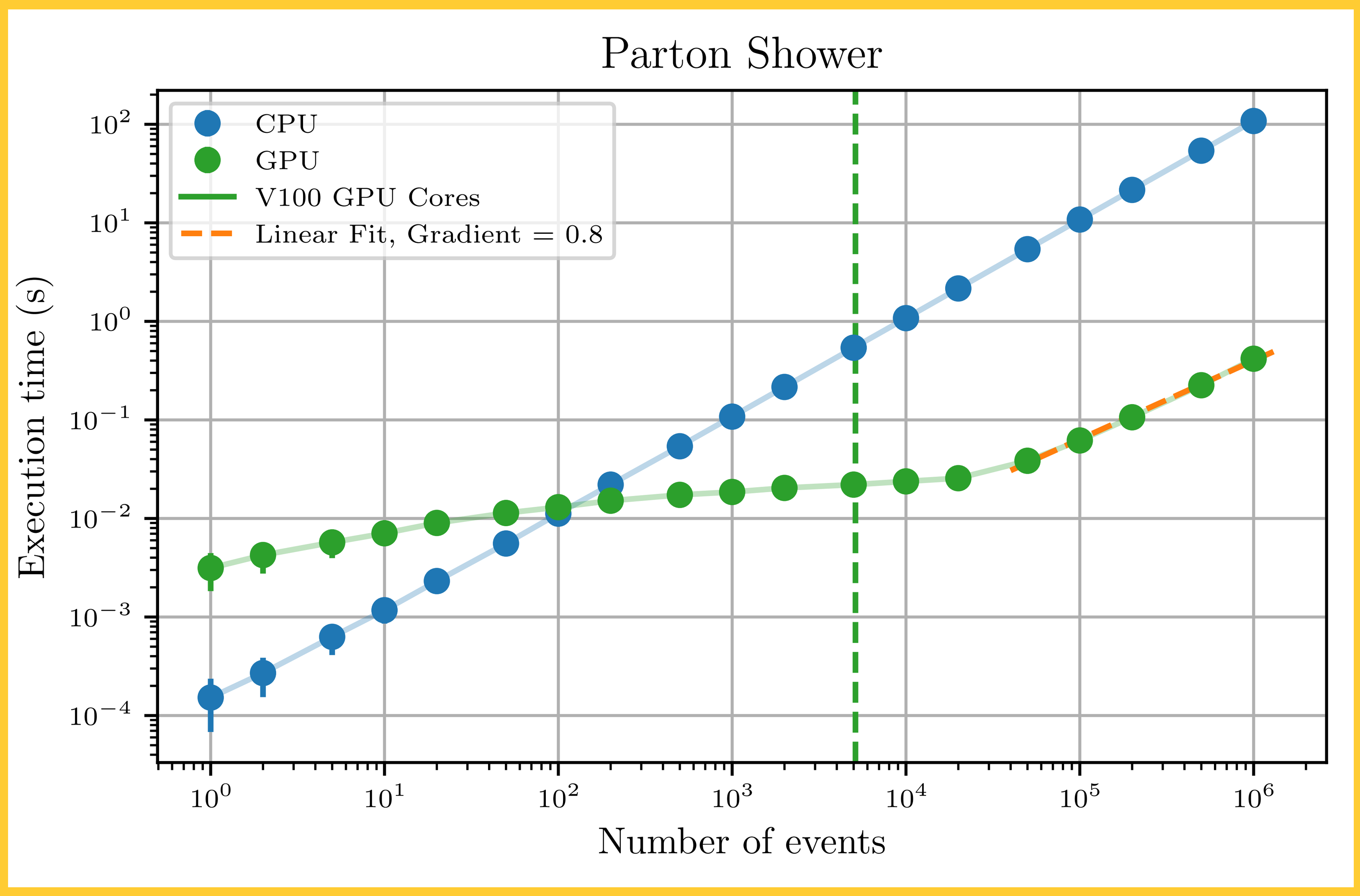
Paralellising the Parton Shower Algorithm

Although each event can take a different trajectory in the veto algorithm, the same steps are repeated every time. To effectively use the GPU for the parton shower, we execute each step of the veto algorithm in parallel, where threads with rejected emissions stay idle, whilst threads with accepted emissions generate the new particle.



Results: A CUDA C++ Parton Shower

We implemented this algorithm for NVIDIA GPUs using CUDA C++. The result was a GPU Event Generator that Simulates fundamental interactions and Emissions. Compared to a near identical CPU-Only Event Generator, the simulations were **275 times faster**.



These event generators are available as the open source program

GAPS: a GPU Amplified Parton Shower

Conclusions and Next Steps

The algorithm and the code are explained in our paper *An Algorithm to Parallelise Parton Showers on a GPU* [2403.08692].

Currently, the generator only simulates final state emissions (i.e., emissions after the fundamental interaction). To simulate initial state emissions, we need to add an extra step in the Parton Shower Algorithm, which uses Parton Distribution Functions (PDFs) to represent the probability of finding a given quark or gluon inside a hadron.