Parallel reweighting with teawREX

Extending the MG5aMC CUDACPP plugin for leading order event reweighting

Zenny Wettersten (CERN, TU Wien)

Quickstart guide

Repo available in my fork of madgraph4gpu

- git clone –recurse-submodules -b rexDev git@github.com:zeniheisser/madgraph4gpu.git DIR
- cd DIR/MG5_aMC/mg5amcnlo
- ./bin/mg5_aMC
- generate `PROC`
- output madevent_simd OR output madevent_gpu
- launch

Include event reweighting, and make sure to add the line `change gpucpp True` to the reweight_card. With this, the Python interface will automatically generate, output, and run the reweighting executable.

Alternative quickstart guide

If you already have generated events you want to reweight,

- git clone –recurse-submodules -b rexDev git@github.com:zeniheisser/madgraph4gpu.git DIR
- cd DIR/MG5_aMC/mg5amcnlo
- ./bin/mg5_aMC
- generate PROC
- output standalone_rwgtcpp RWGT_DIR
- cd RWGT_DIR/SubProcesses
- make
- ./rwgt_driver_cpp.exe -lhe=/PATH/TO/LHEFILE -rwgt=/PATH/TO/REWEIGHT_CARD -slha=../Cards/param_card.dat -out=/PATH/TO/OUTPUT
- ./rwgt_driver_gpu.exe -lhe=/PATH/TO/LHEFILE -rwgt=/PATH/TO/REWEIGHT_CARD -slha=../Cards/param_card.dat -out=/PATH/TO/OUTPUT

Standard “make” automatically detects whether you have a GPU backend available and try to compile using it, otherwise it will resort to vectorised C++.

Caveats

- Only supports single-model reweighting
- Only supports helicity-summed reweighting
- Does not amend kinematics for mass reweighting
- Necessitates "`set BLOCKNAME paramNumber value`" structure in reweight_card
- Possibly unstable(?)
  - Possible race condition in the subprocess compilation
    - All subprocesses rely on src compilation — seen a compilation failure for the subprocesses once based on this
  - Possible exporter issue
    - `TypeError: PLUGIN_ProcessExporter.finalize() got an unexpected keyword argument 'second_exporter'`

How it works

Overview

- REX parses LHE file and SLHA parameter card
- teawREX parses reweight card and keeps track of data pulled from LHE file
- rwgt_instance sorts out details for interfacing with CUDACPP functions
- rwgt_runner pipes parton configurations and momenta between executable and amplitude evaluations
- rwgt_driver is the main executable: Calls rwgt_runners to sort parton configs between different subprocesses and gives a list of them to teawREX

REX

- **R**apid **E**vent e**X**traction
- C++ library for parsing LHE files and extracting data from them
- Can "rotate" LHE files from AoS layout to SoA layouts
- Not (yet) capable of back-and-forth treatment, i.e. can only read LHE files and modify, not write completely new ones
- Very sizable: Even after some cleanup, 3700 lines of code

teawREX

- **t**ensorial **e**vent **a**daption **with** **REX**
- Extension to REX that handles logistics for reweighting
- Extends REX classes to modify SLHA parameter cards
- Takes as input LHE-file, SLHA parameter card, reweight card, and amplitude functions
- Iterates reweight parameters, starting with original parameters, calls amplitude functions with relevant event data in SoA format
- Outputs LHE file with new event weights

rwgt_instance

- Simple wrapper to keep track of CUDACPP format amplitude functions and parton configurations
- Single class `fBridge` that ensures memory alignment etc
- Used to map parton configurations to specific functions

rwgt_runner

- Interface to compile and call amplitudes as a library
- Generated from templates by MG
- Creates std::function to call amplitude evaluation
- Wraps parton-level event comparison to sort LHE file
- Creates `fBridge` object for specific subprocess
- Allows sending SoA directly from C++ into CUDACPP

rwgt_driver

- Main executable for the reweighting
- Takes input: paths to LHE-file, SLHA parameter card, MG-format reweight card, output path
- Gets `fBridge` objects from rwgt_runners
- Loads LHE file, parameter card, reweight parameter sets
  - teawREX calls REX functions to parse and rotate LHE file
  - Uses event comparison operators from rwgt_runners to sort events
- Checks which (possibly all) amplitude functions are relevant
- Sends vector of relevant functions back to teawREX
- teawREX calls amp functions to run reweighting
- Exports rwgt names, cross sections, propagated errors into csv file to be read by MG for terminal output

CERN