Visualization

This notebook provides a short guide on how to use the visualization tool VISTAS, developed by the MLhad group for Pythia. VISTAS is still in a beta stage, and so bugs are to be expected.

Requirements

Before running this notebook, we need to set up our environment. First, we install and import the wurlitzer module. This allows programs that have C-like backends to write their output to the Python console. In short, this allows the output of Pythia to be displayed in this notebook.

```
# Redirect the C output of Pythia to the notebook.
!pip install wurlitzer
from wurlitzer import sys_pipes_forever

sys_pipes_forever()
```

```
Requirement already satisfied: wurlitzer in /usr/local/lib/python3.11/dist-packages
```

Next, we need to install the Pythia module.

```
# Install and import the Pythia module.
!pip install pythia8mc
import pythia8mc as pythia8
```

```
Collecting pythia8mc
  Downloading pythia8mc-8.315.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86
Downloading pythia8mc-8.315.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_6
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 29.9/29.9 MB 32.1 MB/s eta 0:00:00
Installing collected packages: pythia8mc
Successfully installed pythia8mc-8.315.0
```

Finally, we need to download and import the visuzation module.

```
# Download the visualization module.
!wget -q -N https://gitlab.com/mcgen-ct/tutorials/-/raw/main/vistas/visualization.py
!wget -q -N https://gitlab.com/mcgen-ct/tutorials/-/raw/main/vistas/status_meaning.py

# Import `math` and `visualization` modules.
import math
import visualization
```

Introduction

The idea behind VISTAS, a Visualization Interface for Particle Collision Simulations, is to provide a more intuitive way to visualize high energy particle collisions as produced by Monte Carlo event generators. In this initial version of VISTAS, the focus has been to convert the output of Pythia into an interactive visual representation of the event generation process, where the different steps of high energy physics MC event generation are shown, including hard process, parton shower, and hadronization. This is different from experimental event displays, where events are displayed as reconstructed by the detector from real data.

VISTAS uses the Phoenix event display framework, which is focused on experimental event displays. The output of Pythia events is parsed into a format which can be used in this framework and then displayed interactively.

Running Vistas

Displaying Pythia events in a way that provides some intuition can be challenging, and so there are a number of options that we have explored on how the visualization can be produced. Below is a dictionary of the possible settings that are available. Documentation is still in the process of being written up, but hopefully some of the options names are relatively straight forward.

```
# Define settings for the visualization.
settings = {
    # Removes carbon copies.
    "remove_copy": True,
    # Includes beam remnants.
    "beam_remnant": True,
    # Scale factor for visualization.
    "scale_factor": 1,
    # Boost mode: None, "cm_incoming", "cm_outgoing".
    "boost_mode": None,
    # Scaling type: "unit", "energy", "log_energy".
    "scaling_type": "unit",
    # Rescaling: "none", "total_distance_based", "category_distance_based".
    "rescaling_type": "category_distance_based",
    # Base length added to each track.
    "base_length": 40,
    # Shows color connections.
    "color_connection": True,
    # Includes multi-parton interactions.
    "mpi": True,
    # Highlight certain category, all others are grayed out: "hard_process",
    # "beam_remnants", "MPI", "parton_shower", "hadronization",
    # "color_connection".
```

```
    "highlight_category": None,
    # Define where the MPI locations are shifted from the hard process.
    "mpi_location": [
        # Shift in x, cos(25 degrees).
        1 / math.sqrt(2) * math.cos(math.radians(25)) * 30,
        # Shift in y, sin(25 degrees).
        1 / math.sqrt(2) * math.sin(math.radians(25)) * 30,
        # Shift in z.
        0.0,
    ],
}
```

After creating the settings, we create a Pythia instance, configure it, and generate event. We then take that event, pass it to VISTAS, and generate a JSON file.

```
# Create and configure a Pythia instance.
pythia = pythia8.Pythia()

# Configure Pythia here. We can look at top production for example.
pythia.readString("Top:gg2ttbar = on")

# Initialize Pythia and generate an event.
pythia.init()
pythia.next()

# Initialize the Visualization tool.
vistas = visualization.Visualization(pythia, settings)

# Write the output to a JSON file.
vistas.write_json("ttbar.json");
```

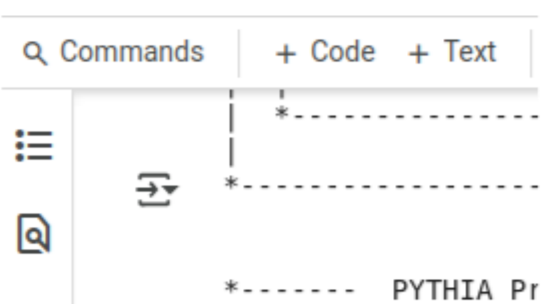
```
*------------------------------------------------------------------------------------
|
|  *---------------------------------------------------------------------------------
|  |
|  |
|  |   PPP   Y   Y  TTTTT  H   H  III    A      Welcome to the Lund Monte Carlo!
|  |   P  P   Y Y    T    H   H   I    A A     This is PYTHIA version 8.315
|  |   PPP     Y     T    HHHHH   I   AAAAA    Last date of change: 27 May 2025
|  |   P       Y     T    H   H   I   A   A
|  |   P       Y     T    H   H  III  A   A    Now is 03 Jul 2025 at 17:46:06
|  |
|  |   Program documentation and an archive of historic versions is found on:
|  |
|  |                               https://pythia.org/
|  |
|  |   PYTHIA is authored by a collaboration consisting of:
|  |
|  |   Javira Altmann, Christian Bierlich, Naomi Cooke, Nishita Desai,
|  |   Ilkka Helenius, Philip Ilten, Leif Lonnblad, Stephen Mrenna,
|  |   Christian Preuss, Torbjorn Sjostrand, and Peter Skands.
|  |
|  |   The complete list of authors, including contact information and
```

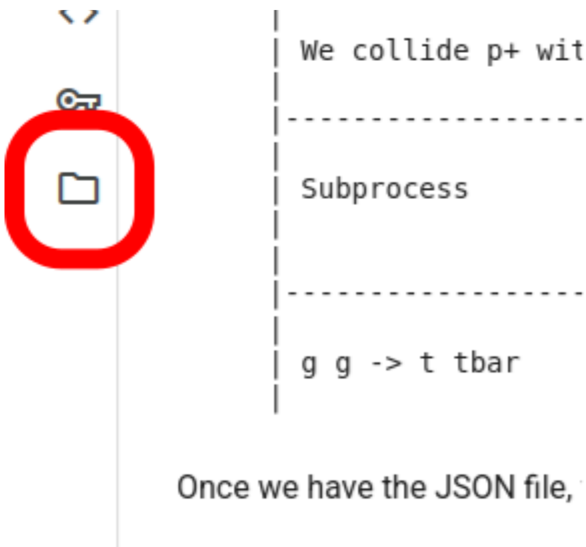
```
 |  |   The complete list of authors, including contact information and
 |  |   affiliations, can be found on https://pythia.org/.
 |  |   Problems or bugs should be reported on email at authors@pythia.org.
 |  |
 |  |   The main program reference is C. Bierlich et al,
 |  |   'A comprehensive guide to the physics and usage of Pythia 8.3',
 |  |   SciPost Phys. Codebases 8-r8.3 (2022) [arXiv:2203.11601 [hep-ph]]
 |  |
 |  |   PYTHIA is released under the GNU General Public Licence version 2
 |  |   or later. Please respect the MCnet Guidelines for Generator Authors
 |  |   and Users.
 |  |
 |  |   Disclaimer: this program comes without any guarantees.
 |  |   Beware of errors and use common sense when interpreting results.
 |  |
 |  |   Copyright (C) 2025 Torbjorn Sjostrand
 |  |
 |  |
 |  *------------------------------------------------------------------------------
 |
 *----------------------------------------------------------------------------------


 *-------  PYTHIA Process Initialization  --------------------------*
 |                                                                  |
 | We collide p+ with p+ at a CM energy of 1.400e+04 GeV            |
 |                                                                  |
 |------------------------------------------------------------------|
 |                                                    |             |
 | Subprocess                                    Code |   Estimated |
 |                                                    |    max (mb) |
 |                                                    |             |
 |------------------------------------------------------------------|
 |                                                    |             |
 | g g -> t tbar                                  601 |   9.089e-06 |
 |                                                                  |
```

Using Phoenix

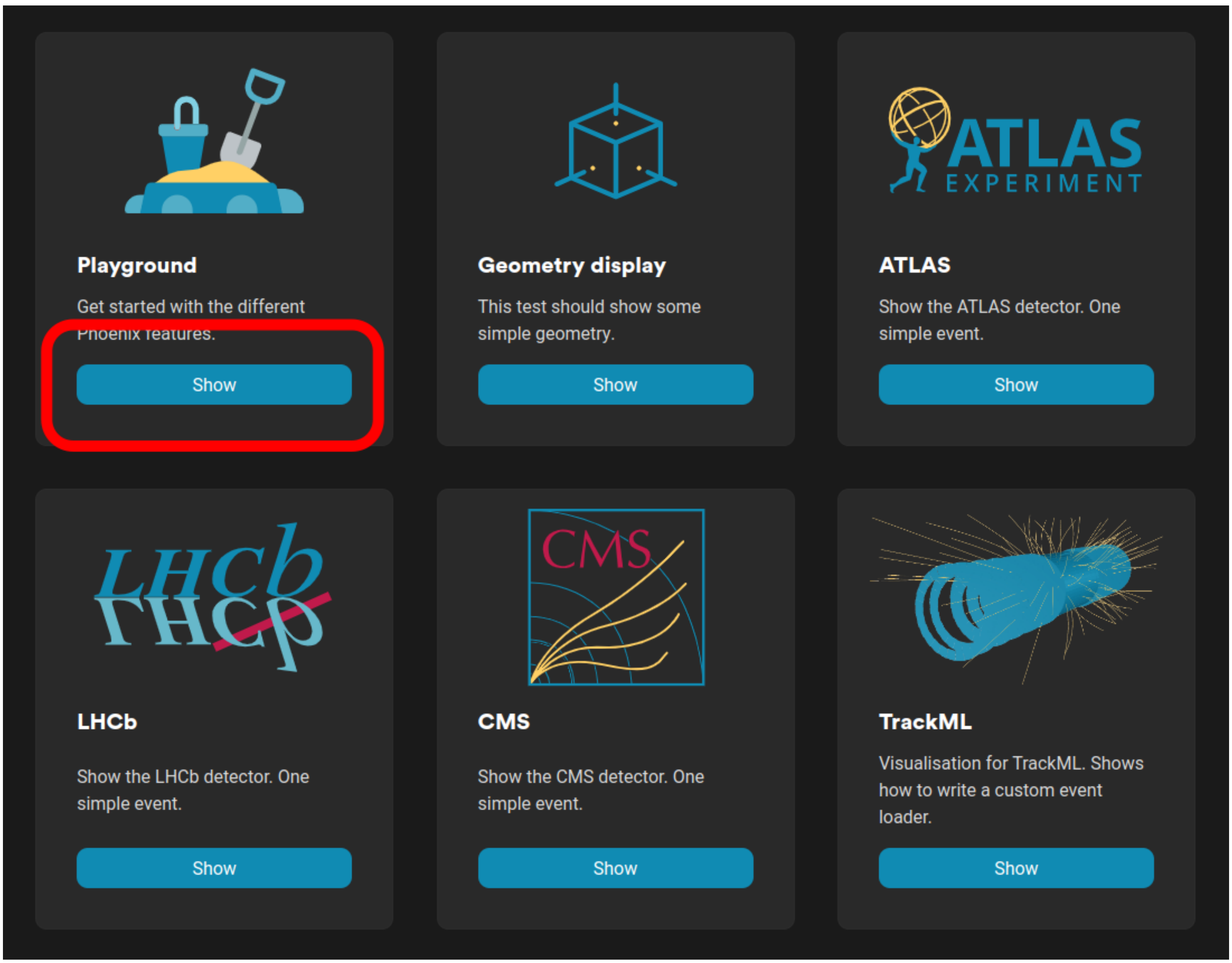
Once we have the JSON file, we can download it from this notebook (if running on Colab) and upload it to [Phoenix](#).

1. Open the Colab file explorer indicated by the folder icon on the left-hand menu of the screen (assuming a default Colab configuration). Select the file generated with VISTAS, the default in this example is `ttbar.json`, and download it locally.

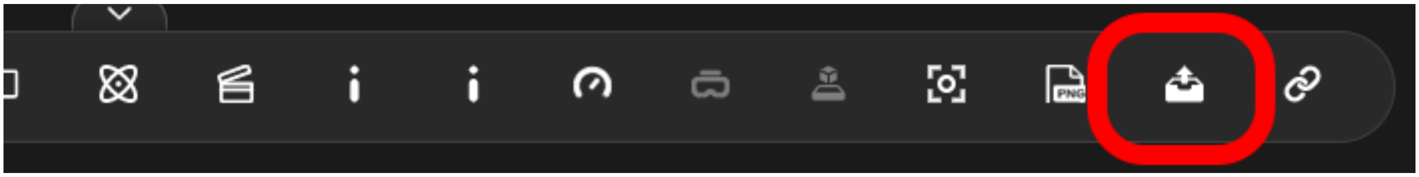




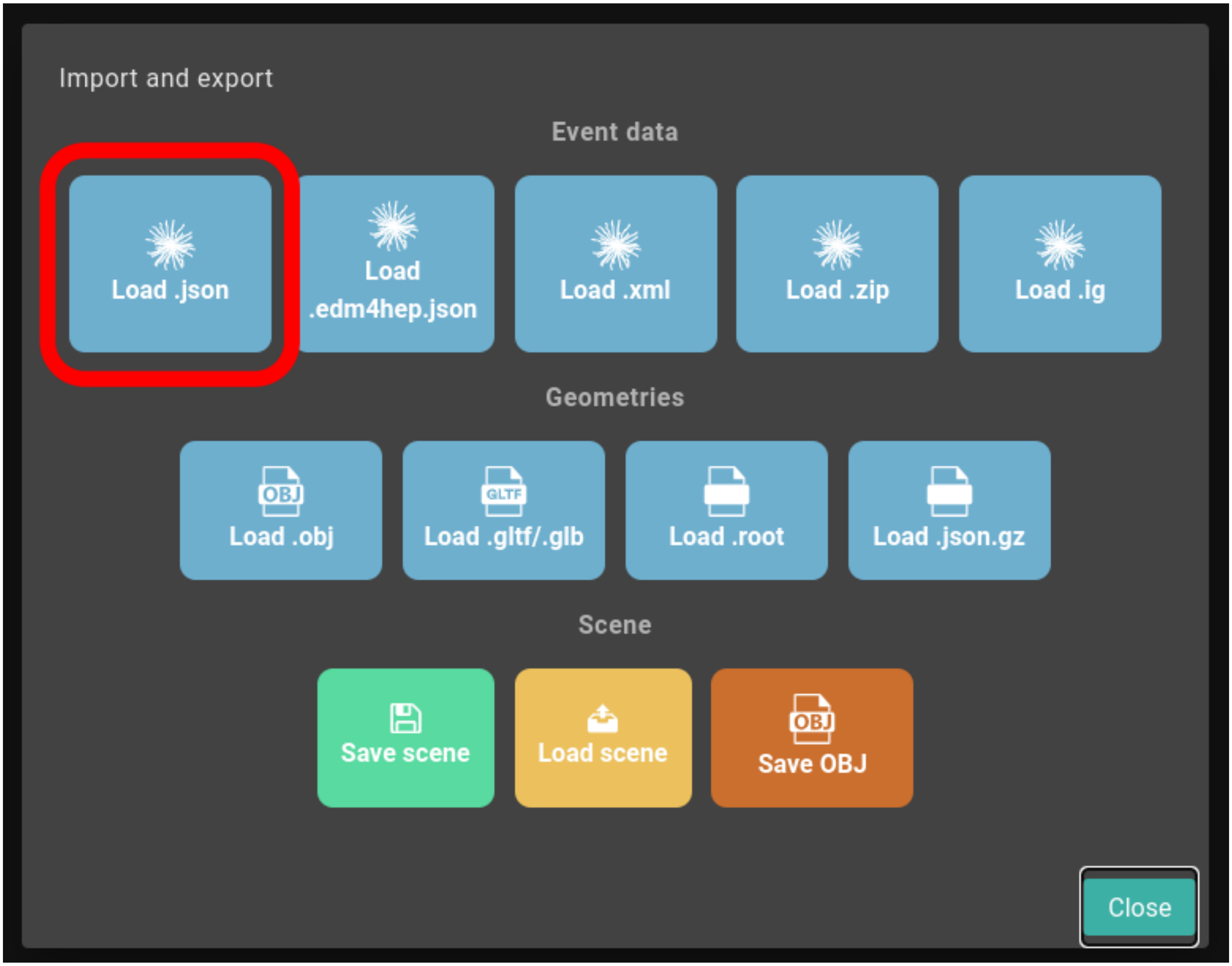
2. Go to the [Phoenix playground](). Alternatively, go to the top level [Phoenix](), and select the playground.



3. Select the `Import and export options` icon from the menu.



4. Import the JSON file by selecting `Load .json`.



5. Explore the event!