

# Requirements:

## Dark Matter Physics Simulation and Analysis:

### Team Members:

Jacob Woods: [jwoods2022@my.fit.edu](mailto:jwoods2022@my.fit.edu)

Nikhil Chaba: [nchaba2021@my.fit.edu](mailto:nchaba2021@my.fit.edu)

Samuel Rock: [srock2023@my.fit.edu](mailto:srock2023@my.fit.edu)

### Client:

Dr. Marcus Hohlmann

Pietro Iapozzuto

Other Graduate Physics Students

### Faculty Advisor:

Pietro Iapozzuto, [piapozzuto2015@my.fit.edu](mailto:piapozzuto2015@my.fit.edu)

Dr. Marcus Hohlmann [hohlmann@fit.edu](mailto:hohlmann@fit.edu)

# Table of contents

<b>1. Introduction:</b>	<b>2</b>
1.1. Purpose:	2
1.2. Scope:	2
1.3. Definitions, Acronyms, and Abbreviations:	2
1.4. References:	2
1.5. Overview:	3
<b>2. Overall Description:</b>	<b>3</b>
2.1. Product Perspective:	3
2.4. Operating Environment:	3
2.5. Design and Implementation constraints:	4
2.6. Assumptions and Dependencies:	4
<b>3. Specific Requirements:</b>	<b>4</b>
3.1. External Interface Requirements:	4
3.2. Functional Requirements:	5
<b>4. Other Non-Functional Requirements</b>	<b>6</b>
4.1. Performance Requirements:	6
4.2. Software Quality Attributes:	7

# 1. Introduction:

## 1.1. Purpose:

- 1.1.1. The purpose of this document is to specify the functional and non-functional requirements for the Analysis and Dark Matter Physics Simulation for the Dark photon project.

## 1.2. Scope:

- 1.2.1. The primary goal of this project is to support accurate comparisons and reproducible analysis while also streamlining the whole workflow so studies can be easily repeated. The project will provide scripts and tools to:
  - 1.2.2. Organise HepMC event files and reconstructed event data files.
  - 1.2.3. Produce analysis plots and derived physics quantities.
  - 1.2.4. Reproduce the same analysis across different simulation settings, including scans over candidate masses (such as particle ID 1023).

## 1.3. Definitions, Acronyms, and Abbreviations:

- 1.3.1. **EIC:** Electron-ion Collider
- 1.3.2. **ePIC:** Electron-Proton/Ion Collider
- 1.3.3. **EIC Shell:** A software environment for working on the Electron-Ion Collider software stack.
- 1.3.4. **MadGraph:** Is a physics software tool that generates particle collision events from theory.
- 1.3.5. **HepMC:** Is a standard file format for storing simulated high-energy physics events.
- 1.3.6. **ROOT:** Is a standard data analysis framework for high-energy physics made by CERN.
- 1.3.7. **CERN:** European Organisation for Nuclear Research.
- 1.3.8. **Invariant mass:** Invariant mass is the mass of a particle or system computed from its total energy and momentum in a way that is the same in all reference frames.
- 1.3.9. **Pseudorapidity:** is an angular coordinate used in collider physics, defined from a particle's polar *angle*  $\theta$ /*theta* $\theta$
- 1.3.10. **DDSIM:** an executable that runs a simulation (it simulates particles passing through the EIC shell detector and records hits).

## 1.4. References:

- 1.4.1. Dark Matter Project Plan
- 1.4.2. Project Design Document;
- 1.4.3. Project Plan Presentation

1.4.4. The requirements document will follow the guidelines set in the IEEE Standard for software requirements specifications.

## 1.5. Overview:

1.5.1. The remainder of the document will give an overall description of the Analysis and Dark Matter Physics Simulation for the Dark Photon Project. Section 2 focuses on product context and constraints, and Section 3 focuses on detailed descriptions of the functional and nonfunctional requirements.

## 2. Overall Description:

### 2.1. Product Perspective:

2.1.1. The expected workflow is:

- Generate events in a HepMC file using MadGraph.
- Run the user-created scripts to create histograms, plots, and metrics on potential parent particles and their children based on the HepMC file created.
- Run DDSIM to simulate particle response.
- Run visualisation;
- Run the reconstruction executable.
- Run the user-created scripts to create histograms, plots, and metrics on potential parent particles and their children based on the reconstruction executable.

### 2.2. Product Functions:

2.2.1. Ingest user-created HepMC files from MadGraph;

2.2.2. Ingest and organise simulated and reconstructed event outputs;

2.2.3. Produce histograms and graph plots for analysis of different particles;

2.2.4. Batch run samples with different simulation configurations;

2.2.5. Log configurations, seed, environment and results;

### 2.3. User Characteristics:

2.3.1. Primary users are:

- Physics faculty who are focusing on the study of Dark Matter.
- Other students/ faculty who plan to maintain the scripts.

### 2.4. Operating Environment

2.4.1. The operating system that will be used to create the HepMC files within MadGrah will be a Linux-based operating system.

- 2.4.2. The operating system that will run the EIC-shell that will be used to create the reconstructed Root files. As well as being responsible for visualisations of the EIC-shell showing the particles, it will again be a Linux-based operating system.
- 2.4.3. As some of the Python scripts need the PyRoot library, a Linux-based operating system is once again chosen for ease and compatibility with ROOT in C scripts.
- 2.4.4. The use of either WSL or Docker container is recommended for any users running either a Windows or Mac operating system to maintain uniformity.

## 2.5. Design and Implementation constraints:

- 2.5.1. Simulation, event generation, and reconstruction are provided by external tools and are thus out of scope for direct development.
- 2.5.2. MadGraph, EIC-shell and Root are required dependencies.
- 2.5.3. Event generation seeds must be controllable to support reproducible comparisons.

## 2.6. Assumptions and Dependencies:

- 2.6.1. Users have access to the environment that can run EIC-shell and Root.
- 2.6.2. Users have access to scripts involved with data organisation.
- 2.6.3. Users have the space available on their workspace to store HepMC files and reconstructed root files.

# 3. Specific Requirements:

## 3.1. External Interface Requirements:

- 3.1.1. User Interface:
  - The system shall provide a command-line interface (CLI) with clear subcommands in order to create HepMC files, create reconstructed root files and run the user-created Python/C scripts so that non-programmers can run without needing to modify source code.
- 3.1.2. File Interface:
  - **Input:** HepMC event File.
  - **Input:** Reconstructed Root output files produced by the external reconstruction pipeline.
  - **Output:** Organised intermediate dataset (e.g., histograms and other graphs).
  - **Output:** logs capturing configuration and seeds.

- 3.1.3. Hardware Interfaces:
  - No special hardware interfaces are required beyond standard compute resources. As a virtual machine is being used, the only cost is the amount of time it takes to compute.
- 3.1.4. Communication Interfaces:
  - No network communications are required for core functionality. A repository is being used for hosting, collaboration and distribution purposes.
- 3.1.5. Software Interfaces:
  - **MadGraph:** Event generation; produces events for the selected dark photon process.
  - **Fortran Compiler:** A dependency of MadGraph.
  - **EIC-Shell:** Detector simulation using ePIC geometry. Produces reconstructed ROOT outputs for analysis.
  - **Root:** Histogramming, plotting, file I/O, and analysis scripting.
  - **GitHub:** Version control and GitHub Pages for a documentation site.

## 3.2. Functional Requirements:

- 3.2.1. Data Generation and Input:
  - **Event Generation:** The system shall generate events for the selected dark photon process using MadGraph and produce event files for downstream processing.
  - **Input Event Format:** The system shall accept generated events in HepMC format as the standard input for detector simulation.
- 3.2.2. Detector Simulation and Reconstruction:
  - **Simulation Execution:** The system shall run detector simulation using EIC-Shell with ePIC geometry to process HepMC input events.
  - **Simulation Output:** The system shall produce simulated ROOT output files from the simulation stage for later comparison and validation.
  - **Reconstruction Execution:** The system shall run the EPIC/EIC reconstruction chain on simulated outputs to generate reconstructed event data.
  - **Reconstruction Output:** The system shall produce reconstructed ROOT output files for use as the primary

inputs to analysis.

3.2.3. Physics Object Identification and Extraction:

- **Dark Photon Identification:** The system shall identify the potential dark photon parent particles in the generated event record (e.g., ID 1023) and extract its decay products.
- **Decay Product Selection:** The system shall select the relevant final-state particles from the dark photon decay (e.g.,  $e^+/e^-$  pair) for analysis.

3.2.4. Visualisation and Reporting:

- **ROOT-Based Plotting:** The system shall generate histograms and plots using ROOT.
- **Track Visualisation Artefacts:** The system shall support the production of visualisation of the outputs for the trajectories/tracks of gamma rays, positive charge rays, and negative charge rays, showing where those respective particles are headed.

## 4. Other Non-Functional Requirements

### 4.1. Performance Requirements:

4.1.1. File Creation:

- Creating the HepMC file through MadGraph should take the system approximately 5-10 minutes to complete. This initial simulation produces the data needed, so a quicker completion time can be expected.
- Taking this HepMC file through DDSIM to obtain a root file took our personal system about four hours. This can be expected due to the reconstruction of the particles within the HepMC file.

4.1.2. Data Analyzation:

- The Python scripts written to skim through the HepMC file will take around 20 seconds to completely find their instructed data. The HepMC file is a text file, so reading through it is relatively quick.
- The C scripts written to follow root analysis will take 30-60 seconds to complete. This is due to the fact that we are analysing the same particles twice, once in the MC particles branch and once in the reconstructed particle branch within the root file.

## 4.2. Software Quality Attributes

### 4.2.1. Software versions

- MadGraph - MG5\_aMC\_v3.6.7
- ROOT - v6.36.08
- DDSIM - DDSIM 1.24.0
- EIC shell - GitHub version
- Python - Python 3.14
- C - C23