POLITECNICO DI TORINO
Repository ISTITUZIONALE

OpenMC-Fusion-Benchmarks, a CAD-Based Automated Framework for Fusion Neutronics Validation

Original
OpenMC-Fusion-Benchmarks, a CAD-Based Automated Framework for Fusion Neutronics Validation / Segantin, S., Pettinari, D., Shimwell, J., Peterson, E.. - In: FUSION SCIENCE AND TECHNOLOGY. - ISSN 1943-7641. - (2026), pp. 1-10. [10.1080/15361055.2026.2664345]

Availability:
This version is available at: 11583/3012642 since: 2026-07-02T14:14:15Z

Publisher:
TAYLOR & FRANCIS INC

Published
DOI:10.1080/15361055.2026.2664345

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

21 September 2026

Taylor & Francis
Taylor & Francis Group

Fusion Science and Technology

ISSN: 1536-1055 (Print) 1943-7641 (Online) Journal homepage: www.tandfonline.com/journals/ufst20

OpenMC-Fusion-Benchmarks, a CAD-Based Automated Framework for Fusion Neutronics Validation

Stefano Segantin, Davide Pettinari , Jonathan Shimwell & Ethan Peterson

To cite this article: Stefano Segantin, Davide Pettinari , Jonathan Shimwell & Ethan Peterson (14 May 2026): OpenMC-Fusion-Benchmarks, a CAD-Based Automated Framework for Fusion Neutronics Validation, Fusion Science and Technology, DOI: 10.1080/15361055.2026.2664345

To link to this article: https://doi.org/10.1080/15361055.2026.2664345

© 2026 The Author(s). Published with license by Taylor & Francis Group, LLC.

Published online: 14 May 2026.

Submit your article to this journal

Article views: 395

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=ufst20

FUSION SCIENCE AND TECHNOLOGY
© 2026 The Author(s). Published with license by Taylor & Francis Group, LLC.
DOI: https://doi.org/10.1080/15361055.2026.2664345

Check for updates

OpenMC-Fusion-Benchmarks, a CAD-Based Automated Framework for Fusion Neutronics Validation

Stefano Segantin[a], Davide Pettinari[a,b], Jonathan Shimwell[c], and Ethan Peterson[a]

[a]*Plasma Science and Fusion Center, Massachusetts Institute of Technology, Cambridge, MA;* [b]*DENERG, Politecnico di Torino, Torino, Italy;* [c]*Proxima Fusion GmbH, Munich, Germany*

Received February 3, 2026
Accepted for Publication April 19, 2026

Abstract — *Currently, verification and validation (V&V) for fusion neutronics are largely performed manually: analysts convert benchmark models between code input formats and reformat the results for comparison, making the process time consuming, error prone, and hard to reproduce. Existing benchmark suites also rely almost exclusively on constructive solid geometry, thereby excluding from validation much of the practical neutronics workflow, including computer-aided design–based geometries, meshing, and transport on unstructured meshes.*

We propose a new approach to address these gaps, implemented in the open-source OpenMC-Fusion-Benchmarks (OFB) repository. The OFB provides a code-agnostic, standardized, and human-readable benchmark specification, with geometries supplied as Standard for the Exchange of Product (STEP) model data files and an automated meshing pipeline. The framework is designed to enable the near-fully automated V&V of neutronics codes and methods, from benchmark definition to simulation execution and results comparison. We illustrate the workflow on a representative benchmark and outline the remaining challenges and the future developments aimed at further automating and strengthening the validation of fusion neutronics tools.

Keywords — *Fusion neutronics, verification and validation (V&V), benchmark, uncertainty quantification, OpenMC.*

I. INTRODUCTION

As private fusion energy companies move closer to licensing applications for their machines, the demand for precise and robust neutronics analyses has become increasingly critical. Designers and analysts must assess key reactor parameters, such as the tritium breeding ratio (TBR), radiation damage, shielding performance, operational dose, component activation, and shutdown dose rates (SDDRs) with high confidence. In this context, a best-estimate plus uncertainty (BEPU) [1] approach is likely to be well received, provided that the methods offer sufficient accuracy and transparent uncertainty quantification (UQ). The validation pipeline of all the methods and models, hence, becomes key.

At present, fusion analysts often validate their computational tools by manually reproducing selected integral benchmarks from legacy benchmark collections like SINBAD and CONDERC [2,3], building internal trust case by case. However, this process is hindered by several limitations:

1. Existing benchmarks are limited in number and often incomplete.

2. Benchmark information is dispersed across multiple input files, formats, and publications.

3. Data structures and conventions are frequently inconsistent.

CONTACT Segantin Stefano segantin@psfc.mit.edu
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

4. Manual remodeling of benchmarks introduces ambiguity and potential errors.

5. These issues hinder reproducibility and validation efforts.

In addition, current workflows are time intensive and poorly integrated with modern automated tool chains, especially those relying on computer-aided design (CAD)–based modeling and UQ.

As nuclear analysis increasingly becomes a bottleneck in iterative reactor design processes, there is a growing need for CAD-native, reproducible, and extensible benchmark workflows. In this work, we introduce the OpenMC-Fusion-Benchmarks (OFB) platform, which addresses many of these challenges through a centralized and automated pipeline. It supports benchmark definition, CAD-based geometry specification, and embedded uncertainty analysis. This streamlines validation, enhances trust in simulations, and enables tighter integration of neutronics in the fusion design cycle.

The OFB supports a modular and extensible architecture that facilitates the inclusion of new benchmarks and interoperability with state-of-the-art simulation tools. By standardizing data formats and workflows, the platform reduces the manual effort required for model preparation and minimizes sources of user-dependent error, promoting reproducibility and transparency in neutronics validation activities.

We introduce the first CAD-based, open-source repository for fusion neutronics verification and validation (V&V), providing a unified platform for code and methods testing. In doing so, we promote a new paradigm for V&V centered on standardization, methodological rigor, and workflow automation.

II. STRUCTURE AND FEATURES

II.A. Specifications and Schema

The OFB is a V&V repository of new generation, designed to support and automate the rigorous and reproducible validation of fusion neutronics codes. It features a collection of integral benchmarks and a database of experimental and code results. The OFB provides a code-agnostic, sufficient, and consistent way to define the data necessary to model a benchmark. Specifically, each benchmark is described by a single human-readable file, `specifications.yaml`, that contains the minimum amount of technical data required to unambiguously define the problem and obtain consistent results.

Each benchmark is defined through a structured `specifications` file, which consists of multiple modular sections. Among these, the sections `metadata`, `materials`, `geometry`, `sources`, `settings`, and `tallies` are mandatory and provide the fundamental information needed for model definition and simulation.

Additional sections, such as `uncertainty_quantification` and `irradiation_schedule`, are optional and used only when relevant to the specific benchmark. For example, the `irradiation_schedule` section is only required for SDDR benchmarks.

The modular structure allows for the seamless addition of new sections in the future to accommodate evolving validation needs. This standardization not only applies to the benchmark inputs, but also to the expected outputs. In particular, the `tallies` section includes a structured definition of the expected results, including shape and dimensionality, which supports automated post-processing and validation workflows.

For scalability and readability, tallies and results are indeed processed as `Xarray DataArrays` and stored in `netCDF` files [4,5]. All `specifications.yaml` files share a common structure and syntax, enforced by a centralized schema file, `benchmark_schema.yaml`, which unambiguously defines the allowed fields, types, and organization of the benchmark data. Each benchmark specifications is automatically validated against the schema during instantiation.

This structure enables the development of application programming interfaces (APIs) capable of reading the specifications, building the model, running the simulation, and postprocessing the results for a given code. While the framework is general and code independent, the OFB currently includes a fully operational Python API supporting OpenMC-based workflows [6,7].

With a few lines of code and no additional input files required, the OFB API allows for

1. Retrieving the benchmark specifications and validating them against the schema.

2. Fetching and meshing the CAD geometry.

3. Building the model and running the simulation.

4. Postprocessing the results.

The workflow and example usage are shown in Fig. 1.

The rigorous and unified benchmark definition directly addresses typical issues found in legacy collections, such as inconsistent data, scattered documentation, and code-specific formats. Moreover, automating

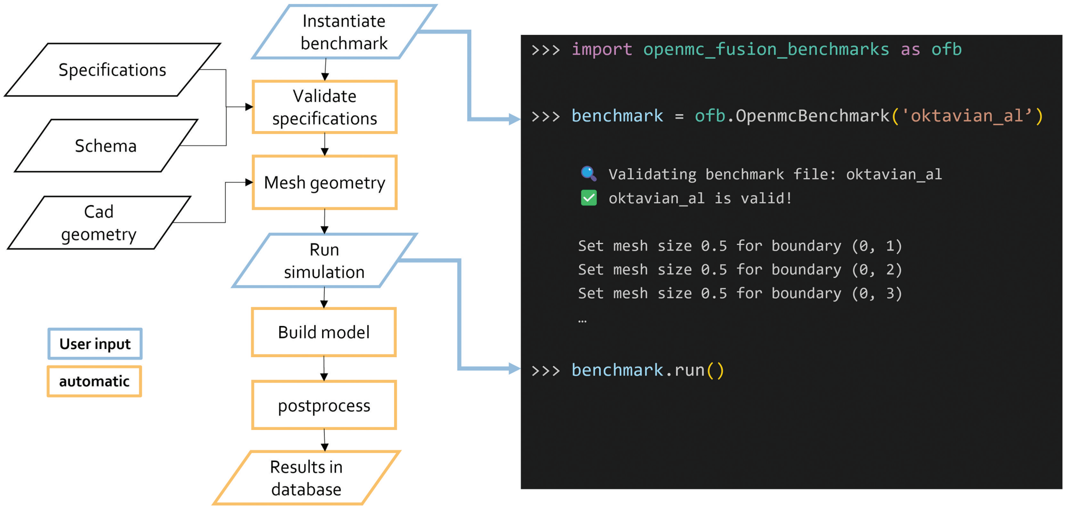

Fig. 1. Standard workflow for benchmark instantiation and simulation, indicating which steps are performed by the user and which are handled automatically by the program in the background.

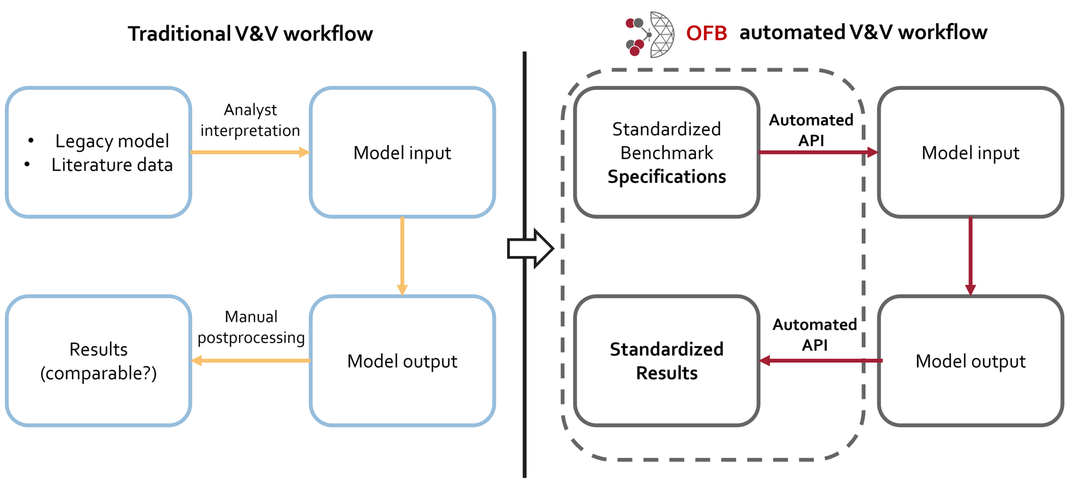

Fig. 2. Overview of simplified neutronics code testing and V&V workflows. The figure contrasts traditional validation using legacy benchmark collections with the automated, standardized, and integrated approach enabled by the OFB framework.

the modeling-postprocessing workflow removes the manual interpretation and modeling steps, reducing subjectivity and human error while increasing reproducibility and efficiency. Fig. 2 displays the mentioned improvements to the benchmark testing and neutronics V&V workflow.

II.B. CAD Geometry

As fusion machine designers and analysts increasingly adopt CAD-based workflows for neutronics modeling and simulation, the OFB accommodates this shift by providing benchmark geometries in CAD format, specifically using the widely supported STEP file standard. For legacy benchmarks derived from the literature, such as those from the CONDERC suite [3], the geometry definitions were originally constructed using constructive solid geometry (CSG). These CSG models have been automatically converted to a CAD format using the GEOUNED tool [8], and have undergone careful inspection and verification to ensure fidelity.

Each benchmark's `specifications.yaml` file references the corresponding CAD file and includes suggested parameters for meshing, such as the maximum mesh element size for each component, enabling reproducible meshing across tools.

To support automated workflows, the OFB is designed to interface with a variety of meshing tools via pluggable APIs. Well-established packages like gmsh, and its Python-based wrappers (including cad-to-dagmc, stellarmesh, and others) [9,10], as well as proprietary tools, such as Coreform Cubit [11], provide scripting interfaces that can be integrated into the benchmark pipeline. The OFB currently includes an API for automatic meshing using the cad-to-dagmc package [9], which facilitates generation of mesh-compatible DAGMC geometries directly from the CAD input.

Furthermore, the availability of benchmark geometries in CAD format enables the systematic testing and validation of CAD-to-CSG conversion workflows, such as those implemented by GEOUNED [8]. This bidirectional capability enhances the utility of the OFB for validating neutronics workflows that rely on either CSG or CAD models, contributing to broader interoperability and reproducibility within the fusion neutronics community.

II.C. Uncertainty Quantification

The implementation of a Total Monte Carlo (TMC) engine within the OFB framework serves several critical objectives. First and foremost, it enhances the overall credibility of the neutronics simulation and validation process by providing a robust methodology for UQ, thereby increasing confidence in both the benchmark models and the neutronics codes and methods used.

Through the repeated sampling and propagation of uncertainties, the TMC approach enables the systematic identification of the dominant sources of uncertainty in key figures of merit, such as TBR. This capability aligns well with the objectives of the BEPU methodology, representing a meaningful step toward its integration into fusion device design and analysis workflows.

At present, the engine supports the automated perturbation of nuclear data. Work is underway to generalize the workflow so that perturbations can be applied to all other input parameters as well. The nuclear data perturbation procedure utilizes the NJOY and SANDY codes [12,13] to process covariance data and generate randomized nuclear data libraries.

NJOY is employed to access and convert evaluated nuclear data files into a usable format, while SANDY enables the sampling of cross-section covariances to produce perturbed nuclear data consistent with the underlying uncertainty structure.

Architecturally, the TMC engine is implemented as a for-loop that embeds a perturbed neutronics model at every iteration. In order to run a benchmark simulation in UQ mode it is just necessary to set the argument `uq = True` in the `run()` method (Listing 1).

The output from the TMC simulations is fully compatible with standalone OFB benchmark runs, maintaining the same tally structure. However, in TMC mode, one or more additional dimensions are appended to the resulting `xarray.DataArray`, corresponding to the perturbed realizations of the model (see Fig. 3).

These new dimensions are clearly labeled and indexed to facilitate statistical analysis and visualization of the propagated uncertainties. To support this, the OFB API includes built-in tools for distribution analysis, convergence checking, and basic statistical diagnostics, which help to ensure that the uncertainty estimates are meaningful and robust.

Importantly, the TMC engine is not limited to predefined OFB benchmarks; it can be applied to any user-defined OpenMC model that conforms to the OFB's simulation interface. This generality makes it a powerful and reusable tool for UQ and validation across a wide range of fusion neutronics applications.

II.D. Database of the Results

The OFB repository includes an integrated database of the simulation and experimental results to support model validation, code-to-code comparison, and method-to-

Listing 1: Benchmark Instantiation and Simulation.

```python
import openmc_fusion_benchmarks as ofb

# Instantiate oktavian-aluminum benchmark
benchmark = ofb.OpenmcBenchmark(name='oktavian_al')
# Build model, run UQ, postprocess to Xarray
benchmark.run(uq=True)
```

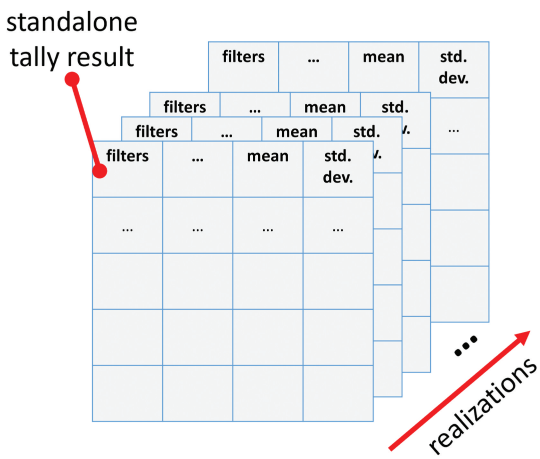

Fig. 3. Array shape schematic of a TMC run result. The same tally is repeated on the third dimension as many times as the number of perturbations/realizations specified in the TMC engine.

method comparison. Legacy benchmark results, both from experiments and from historical neutronics code calculations, have been systematically restructured and reformatted to match the standardized `xarray.DataArray` structure adopted by the OFB. This ensures consistent syntax and data access across all models and workflows.

The database is continuously expanded to include new simulation results, categorized by key parameters such as the simulation code, code version, nuclear data library, meshing tool, and CAD preprocessing pipeline. This rich metadata tagging supports the transparent traceability and reproducibility of results.

A dedicated Python API is provided to interact with the database, enabling users to retrieve, inspect, visualize, and compare results in a programmatic and reproducible way. This functionality not only facilitates side-by-side validation against experimental data or alternate codes, but also supports automated regression testing and statistical post processing across large result sets.

II.E. Documentation

The OFB repository is fully documented to facilitate ease of use, reproducibility, and community adoption. Comprehensive documentation is hosted online via `ReadTheDocs` [14], providing detailed information on the repository structure, available benchmarks, implemented methods, API usage, data inspection routines, and automated reporting tools.

To guide users through common workflows and facilitate onboarding, a suite of Jupyter tutorial notebooks is also provided. These notebooks demonstrate typical usage scenarios, including model setup, results analysis, uncertainty propagation, and benchmarking comparisons, making the repository accessible to both new and experienced users.

III. IMPLEMENTED EXAMPLE

Currently, the OFB repository, which is in a proof-of-concept phase, provides the complete workflow for the Oktavian aluminum benchmark. In the following, we present the main results obtained for this benchmark.

Fig. 4 shows the CAD-based geometry of the benchmark, as available as step file in the OFB, visualized via SolidWorks and the relative unstructured mesh generated automatically during the benchmark instantiation with `gmsh` engine deployment [15,16].

Figs. 5 and 6 display, for the two main results of the Oktavian aluminum benchmark (i.e., neutron and photon leakage spectra), a comparison between the experimental results, the OpenMC-CSG results, and the MCNP-CSG results. The experimental and MCNP-CSG are legacy results available in the OFB results database.

Considering the results available in the literature [2], a slight deviation in the computed data from the experimental (C/E) results was to be expected. The deviation in the OpenMC CAD results is of similar magnitude of the CSG results included here. The slight discrepancy between the OpenMC CAD and CSG is associated with the geometry approximation caused by the meshing process. Future works will focus on a sensitivity analysis on the mesh element size and quality.

Figs. 7 and 8 display, for the Oktavian aluminum neutron and photon leakage spectra, a comparison of different nuclear data libraries results obtained by running the OpenMC CAD model available in the OFB. The

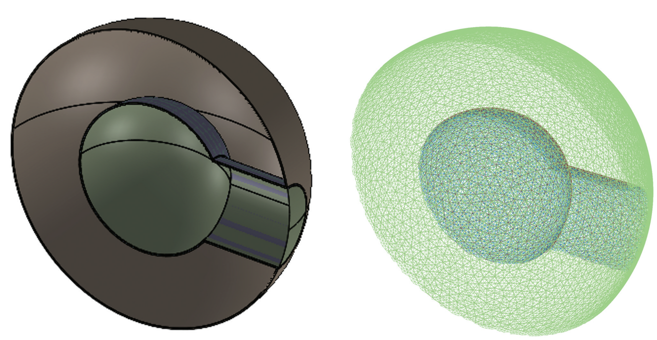

Fig. 4. Sectioned geometry of the Oktavian aluminum and relative mesh used for unstructured mesh transport.

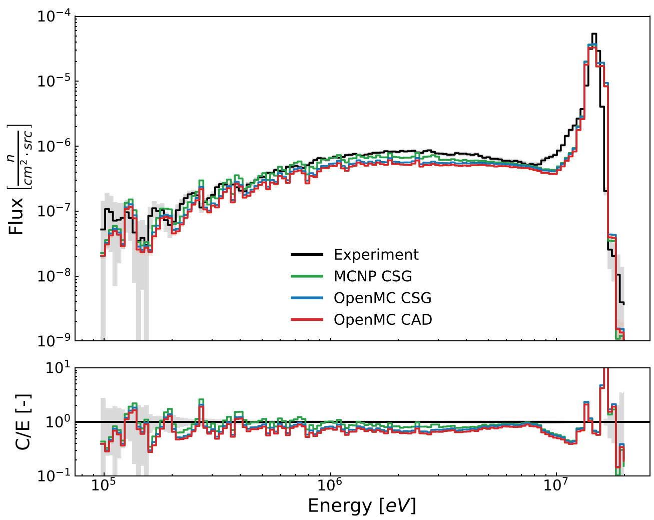

Fig. 5. Neutron leakage results for the (top) energy spectrum and (bottom) C/E values of the spectrum: comparison of the OpenMC CAD, OpenMC-CSG, and MCNP-CSG results. The MCNP results were retrieved from SINBAD [2]. All the computational simulations were run with the ENDF/B-8.0 library. The shaded regions account for three standard deviations.

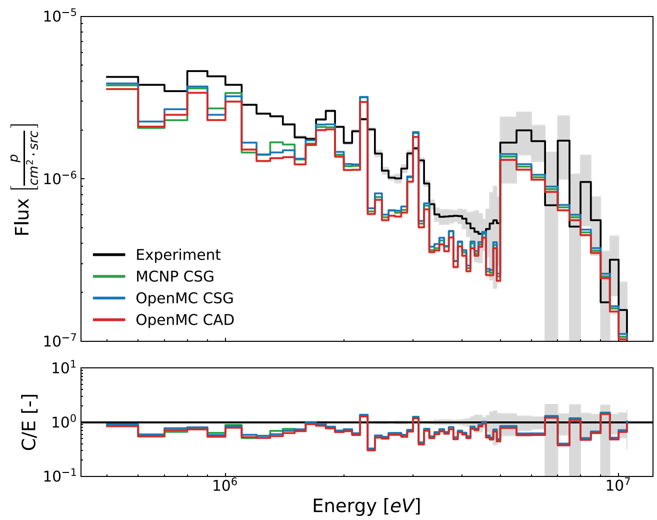

Fig. 6. Photon leakage results for the (top) energy spectrum and (bottom) C/E values of the spectrum: comparison of the OpenMC CAD, OpenMC-CSG, and MCNP-CSG results. The MCNP results were retrieved from SINBAD [2]. All the computational simulations were run with the ENDF/B-8.0 library. The shaded regions account for three standard deviations.

nuclear data libraries selected were ENDF/B-8.0, ENDF/B-8.1, FENDL-3.2c, and JEFF-3.3 [17–20].

Figs. 9 and 10 depict the the Oktavian aluminum results for the UQ run performed by perturbing Al27 nuclear data in 500 realizations. Only the neutron cross sections were perturbed. The perturbations were performed on the ENDF/B-8.0 nuclear data library [17]. The figures include a statistical study of convergence through the evolution of the means, relative standard deviation, and relative standard deviation of the standard deviations with the TMC realizations, and the distribution of the means.

IV. DISCUSSION

In the course of our work on OpenMC validation and the testing of advanced fusion neutronics methods, we observed that existing benchmark collections are often not openly accessible and lack standardized benchmark definitions. More critically, to date there exists no benchmark collection that enables CAD-based neutronics workflows by providing model geometry definitions directly in CAD format.

These limitations have forced V&V and methods testing workflows, ours included, to rely on manual geometry conversion and data manipulation, both for newly generated results and for incorporating legacy results into the comparisons. Such processes are inherently error prone and can compromise reproducibility.

We addressed this issue through the introduction of benchmark specifications, which, when combined with CAD-based geometries, provide a human-readable, transport code–agnostic, and unambiguous definition of a fusion

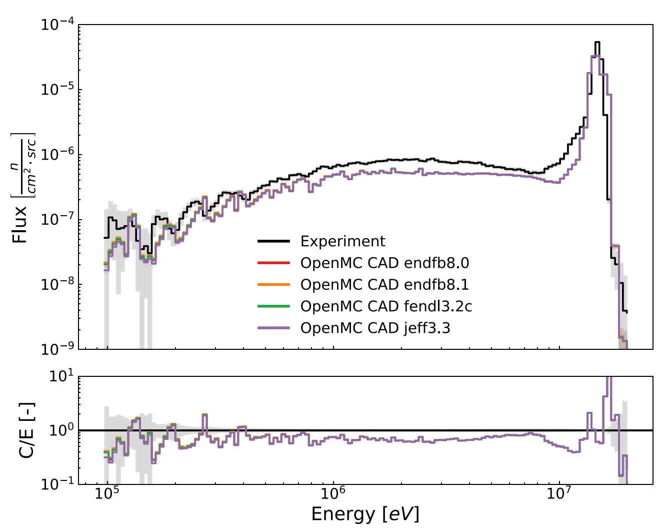

Fig. 7. Neutron leakage results for the (top) energy spectrum and (bottom) C/E values of the spectrum: comparison of ENDF/B-8.0, ENDF/B-8.1, FENDL-3.2c, and JEFF-3.3 nuclear data libraries with the OpenMC CAD model available in the OFB. The shaded regions account for three standard deviations.

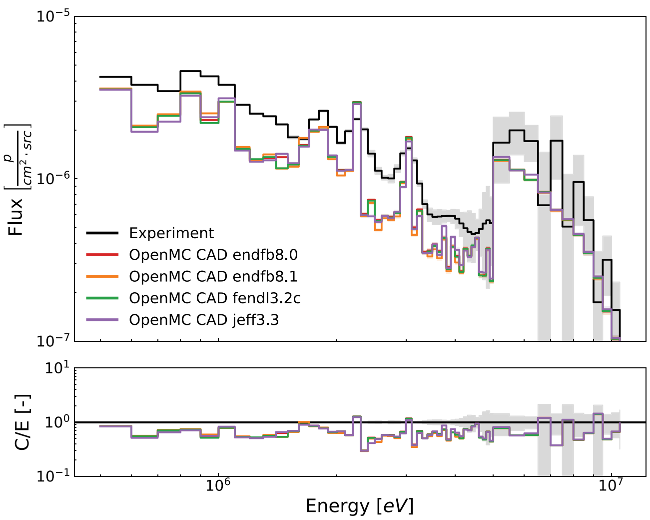

Fig. 8. Photon leakage results for the (top) energy spectrum and (bottom) C/E values of the spectrum: comparison of the ENDF/B-8.0, ENDF/B-8.1, FENDL-3.2c, and JEFF-3.3 nuclear data libraries with the OpenMC CAD model available in the OFB. The shaded regions account for three standard deviations.

neutronics benchmark. This approach enables rigorous modeling practices and opens the door to workflow automation. In addition to promoting standardization, it reduces the human error associated with manual conversion, remodeling, and data reshaping.

We recognize, however, that while this methodology reduces sporadic and varied user-side errors, it also introduces the potential for systematic errors originating from the developer or contributor side, particularly in the drafting of the `specifications` file, the design of the CAD geometries, or the development and maintenance of the API. Many of these risks can be mitigated through robust code testing, continuous integration (CI), and the adoption of a benchmark schema. Overall, we expect the benefits of this approach, including standardization, automation, and the reduction in user error, to outweigh the risks. Systematic errors, once identified, are typically straightforward to correct and do not reoccur once resolved.

A particularly important novelty of this work is the provision of benchmark geometry definitions directly via CAD files. To our knowledge, this is the first benchmark collection to offer this feature, which we regard as a critical requirement for the fusion energy sector. Fusion device concepts vary significantly, and their geometries are often highly complex (e.g., tokamaks and stellarators). While CSG definitions may be suitable during early, preconceptual design phases, they are

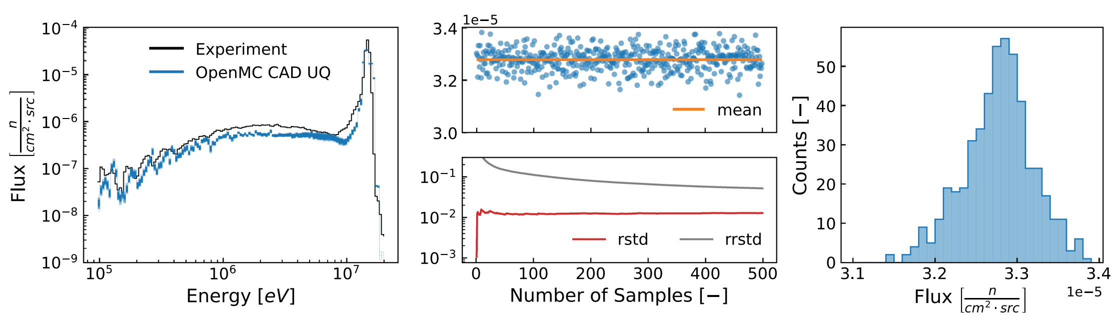

Fig. 9. Results for the TMC-UQ run perturbing the Al27 neutron cross sections: neutron leakage energy spectrum results. The center panels and right panel give an insight into the statistics for the 14-MeV energy group.

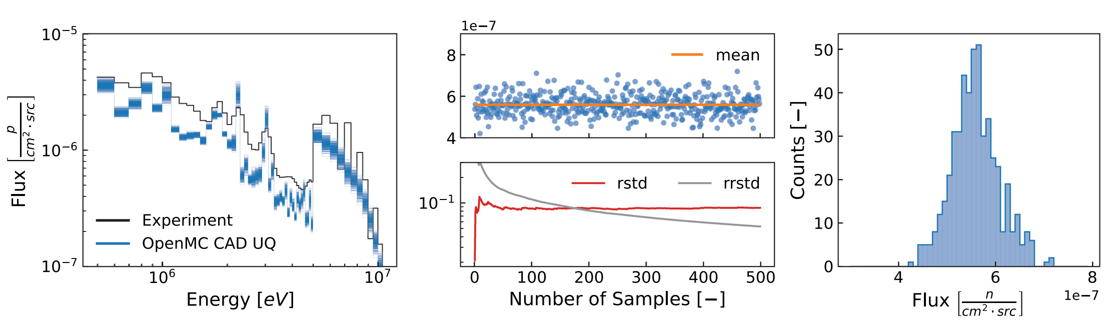

Fig. 10. Results for the TMC-UQ run perturbing the Al27 neutron cross sections: photon leakage energy spectrum results. The center panels and right panel give an insight into the statistics for the 7.5-MeV energy group.

inadequate for advanced stages of development where ultra-high precision in quantities such as the TBR, magnet heating, material damage, and SDDR is essential for design approval, licensing, and eventual operation.

In practical terms, nuclear analysts in fusion energy projects typically receive CAD files directly from the machine designers. Thus, it is natural to provide a validation framework that operates from CAD-based geometries as a starting point. Manual conversion from CAD to CSG is both time consuming and error prone, and even when performed without mistakes, introduces unavoidable geometric approximations. The uncertainty in the neutronics results arising from such geometric approximations is widely recognized within the community as being among the largest contributors to overall simulation uncertainty.

The most reliable current workflows either transport particles in unstructured mesh geometries or employ automatic CAD-to-CSG conversion (e.g., the GEOUNED tool [8]). In our experience, automatic CAD-to-CSG conversion can significantly speed up particle transport, but is limited to relatively simple geometries. Unstructured mesh transport, while potentially slower in terms of the transport phase, offers higher geometric fidelity and can simplify certain workflows, such as the rigorous two-step SDDR method [21].

Moreover, for very large and complex models, such as full-facility simulations, where computationally intensive tallies (e.g., mesh based) dominate, the fraction of the total run time spent on transport decreases. This reduces the performance gap between CSG and unstructured mesh simulations when using identical geometries [22].

Such workflows require rigorous testing and validation, which is why we designed the OFB to support both approaches starting from the same CAD geometry and associated `specifications` file. Nonetheless, in implementing CAD-based workflows, especially in an automated and open-source context, we have encountered several technical challenges. Even standardized CAD formats (e.g., STEP) are not fully unambiguous in syntax, meaning that the same geometry processed in different CAD tools (e.g., SolidWorks, FreeCAD, SpaceClaim) can lead to variations in the final CAD model, and more critically, in its meshing.

Volume identification is generally straightforward, but surface identification can be more challenging, particularly when assigning materials or defining tallies. Surface identifiers may vary depending on the CAD engine and meshing tool used, which can complicate reproducibility. The OFB currently relies on surface identifiers for tally definition, but we are developing a more generic approach based on identifying surfaces via specified Cartesian coordinates.

Regarding meshing, the OFB currently uses the `cad_to_dagmc` package, which wraps the open-source `gmsh` meshing engine, a tool already widely adopted in fusion neutronics analysis [9,16]. While this ensures a fully open-source default workflow, the meshing step is highly sensitive to geometry complexity, CAD model quality, and the CAD engine used.

In particular, cad_to_dagmc performs well with OpenCascade-based CAD kernels [23]. However, there remains a lack of highly resilient and efficient open-source meshing tools capable of handling the full range of fusion geometries. To address this, we are working to incorporate alternative meshing backends, such as Coreform's Cubit Python API [11], into the OFB.

As the repository matures and user experience grows, future development priorities will be guided by community needs. This may involve balancing versatility and model personalization that are critical for testing new methods and analyzing their outcomes against the demands of rigorous, automated validation, which would require expanded CI integration, workflow automation, and advanced quality evaluation techniques for the results.

V. CONCLUSIONS AND FUTURE WORKS

This work introduced the OFB repository, a new-generation V&V platform designed to support and automate the rigorous, reproducible validation of fusion neutronics codes. The OFB, which provides a curated collection of integral benchmarks together with a repository of comparable results, explicitly targets three major shortcomings of legacy benchmark collections and current fusion neutronics V&V workflows: (1) limited rigor and standardization in the benchmark definitions, (2) the absence of CAD-based workflows, and (3) the lack of automated testing and validation pipelines.

The OFB reduces human error and improves the reproducibility and comparability of results through a unified `specifications` file that standardizes the benchmark definitions. For the first time in this context, geometries are distributed as CAD models, reflecting the workflows and geometry representations typically used by fusion designers and neutronics analysts.

A Python API automates the testing and validation workflows. A further key novelty is the implementation and automation of a UQ pipeline based on a TMC engine, enabling more comprehensive testing of codes and methods and moving toward a BEPU design approach.

The framework is modular. Alongside a default open-source CAD-to-mesh workflow, it allows for alternative meshing engines and CAD-to-CSG conversion tools to be plugged in for comparison. The benchmark models are easily customizable, facilitating the development and testing of new neutronics methods. New benchmarks can be added, and a database of the results can be populated for broad, systematic comparison. The OFB has already been used internally for testing newly developed neutronics methods, which has helped to refine its usability.

We recognize that further steps are needed to minimize errors in the V&V workflow. While the OFB substantially reduces analyst/user error, it is not yet immune to systematic errors that may arise in drafting the `specifications` or in the API implementation. We expect such issues to be detectable and correctable, and are introducing mitigation measures (e.g., schema-based validation, CI checks).

Future work will focus on extending the benchmark suite, with particular emphasis on fusion-relevant quantities that are often underrepresented in existing collections, such as the TBR, shielding responses, and the SDDR. Additional meshing pipelines (e.g., for Cubit) are a priority, as is the implementation of more advanced statistical diagnostics within the UQ pipeline. In this regard, we are currently generalizing the TMC engine to support the automatic perturbation of input parameters beyond nuclear data.

The OFB is intended as a community resource. We explicitly invite contributions that (1) add new code pipelines (for different transport codes and workflows), (2) introduce additional benchmarks, (3) integrate new UQ and sensitivity analysis methods, (4) extend the set of meshing and CAD-to-geometry workflows, and (5) generally improve and harden the V&V pipelines.

In the longer term, we aim to integrate the OFB into workflows analogous to CI for simulation codes, where events such as new code releases or the introduction of new methods automatically trigger the execution of the full benchmark suite, the application of a grading system, and automatic report generation for code quality assurance.

Author Contributions

CRediT: **Stefano Segantin:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing; **Davide Pettinari:** Formal analysis, Investigation, Methodology, Software, Validation; **Jonathan Shimwell:** Methodology, Software; **Ethan Peterson:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by Commonwealth Fusion Systems under RPP036.

ORCID

Stefano Segantin http://orcid.org/0000-0002-5745-7293

References

1. F. D’Auria, C. Camargo, and O. Mazzantini, “The Best Estimate Plus Uncertainty (BEPU) Approach in Licensing of Current Nuclear Reactors,” *Nucl. Eng. Des.*, **248**, 317 (2012); https://doi.org/10.1016/j.nucengdes.2012.04.002.
2. I. A. Kodeli and E. Sartori, “Sinbad–Radiation Shielding Benchmark Experiments,” *Ann. Nucl. Energy*, **159**, 108254 (2021); https://doi.org/10.1016/j.anucene.2021.108254.
3. L. Giot et al., “Compilation of Nuclear Data Experiments for Radiation Characterization. Summary Report of a Technical Meeting,” Technical Report, International Atomic Energy Agency, Vienna, Austria (2023).
4. S. Hoyer and J. Hamman, “Xarray: N-D Labeled Arrays and Datasets in Python,” *J. Open Res. Softw.*, **5**, *1*, 10 (2017); https://doi.org/10.5334/jors.148.10.5334/jors.148.
5. R. Rew and G. Davis, “NetCDF: An Interface for Scientific Data Access,” *IEEE Comput. Grap. Appl.*, **10**, *4*, 76 (1990); https://doi.org/10.1109/38.56302.
6. P. K. Romano and B. Forget, “The OpenMC Monte Carlo Particle Transport Code,” *Ann. Nucl. Energy*, **51**, 274 (2013); https://doi.org/10.1016/j.anucene.2012.06.040.
7. P. K. Romano et al., “OpenMC: A State-of-the-Art Monte Carlo Code for Research and Development,” *Ann. Nucl. Energy*, **82**, 90 (2015); https://doi.org/10.1016/j.anucene.2014.07.048.
8. J. Catalán et al., “GEOUNED: A New Conversion Tool from CAD to Monte Carlo Geometry,” *Nucl. Eng. Technol.*, **56**, *6*, 2404 (2024); https://doi.org/10.1016/j.net.2024.01.052.
9. “Cad_to_dagmc: Cad to DAGMC Conversion Workflow” GitHub (2025); https://github.com/fusion-energy/cad_to_dagmc.
10. “Stellarmesh,” GitHub (2025); https://github.com/stellarmesh/stellarmesh.

11. L. L. C. Coreform, "Coreform Cubit Meshing Software," https://coreform.com/products/coreform-cubit/ (accessed Oct. 08, 2024).
12. R. Macfarlane et al., "The NJOY Nuclear Data Processing System, Version 2016," Technical Report, Los Alamos National Laboratory, Los Alamos, NM (2017).
13. L. Fiorito et al., "Nuclear Data Uncertainty Propagation to Integral Responses Using SANDY," *Ann. Nucl. Energy*, **101**, 359 (2017); https://doi.org/10.1016/j.anucene.2016.11.026.
14. OpenMC Fusion Benchmarks Contributors, "OpenMC Fusion Benchmarks: User and Developer Documentation" (2025); https://openmc-fusion-benchmarks.readthedocs.io/en/latest/ (accessed Dec. 23, 2025).
15. "MySolidWorks" MySolidworks (2023); https://my.solidworks.com/.
16. C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D Finite Element Mesh Generator with Built-In Pre- and Post-Processing Facilities," *Int. J. Numer. Methods Eng.*, **79**, *11*, 1309 (2009).
17. D. A. Brown et al., "ENDF/B-VIIIThe 8th Major Release of the Nuclear Reaction Data Library with CIELO-Project Cross Sections, New Standards and Thermal Scattering Data," *Nucl. Data Sheets*, **148**, 1 (2018); https://doi.org/10.1016/j.nds.2018.02.001.
18. G. Nobre et al., "ENDF/B-VIII.1: Updated Nuclear Reaction Data Library for Science and Applications," *arXiv Prepr. arXiv:2511.03564* (2025).
19. R. Forrest et al., "FENDL-3 Library-Summary Documentation," Technical Report, International Atomic Energy Agency, Vienna, Austria (2012).
20. A. J. Plompen et al., "The Joint Evaluated Fission and Fusion Nuclear Data Library, JEFF-3.3," *Eur. Phys. J. A*, **56**, *7*, 181 (2020); https://doi.org/10.1140/epja/s10050-020-00141-9.
21. E. E. Peterson et al., "Development and Validation of Fully Open-Source R2S Shutdown Dose Rate Capabilities in OpenMC," *Nucl. Fusion*, **64**, *5*, 056011 (2024); https://doi.org/10.1088/1741-4326/ad32dd.
22. K. Du, "CAD-Based Geometry Representations for Monte Carlo Fusion Neutronics Methods and CSG vs. DAGMC Performance Tradeoffs in OpenMC" MS Thesis, Department of Nuclear Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA (2024).
23. "Open Cascade Technology" Open Cascade S.A.S. Guyancourt, France (2025); https://www.opencascade.com.