

Journal Name

Crossmark

PAPER

RECEIVED
dd Month yyyy

REVISED
dd Month yyyy

Multi-Method Validation Framework for Teaching Radiation Transport: Comparing Monte Carlo, Analytical, and Physics-Informed Machine Learning Approaches

Yash Varshney^{1,*}, Abhimanyu Agarwal¹

¹Gurukul The School, Ghaziabad, Uttar Pradesh, India

*Author to whom any correspondence should be addressed.

Keywords: physics education, radiation transport, Monte Carlo simulation, physics-informed neural networks, scintillator detectors, validation methodology, proton therapy education

Abstract

Students in computational physics courses increasingly rely on sophisticated simulation tools and machine learning models, yet often treat these as “black boxes” without developing the critical skill of multi-method validation. This pedagogical gap is concerning: in professional physics practice, validating computational predictions against independent methods is essential for ensuring reliability. We present an educational framework that teaches multi-method validation through radiation transport by systematically comparing Monte Carlo simulation (OpenTOPAS), analytical theory (Bethe-Bloch), and physics-informed neural networks. Using proton energy deposition in plastic scintillator detectors as an accessible case study, students learn to interpret when models agree, when they disagree, and, crucially, why discrepancies often reveal physics rather than computational error. The framework demonstrates that embedding physics constraints (specifically smoothness) into machine learning improves interpolation to unseen conditions by approximately 13

1 Introduction

1.1 Why Validation Matters in Physics Education

The ability to validate computational results is among the most important, yet most undertaught, skills in modern physics education. As the American Association of Physics Teachers has emphasized, computational literacy must include not just the ability to write simulations, but the critical judgment to assess when those simulations can be trusted [1]. Yet undergraduate physics curricula rarely provide structured opportunities for students to practice multi-method validation: the systematic comparison of results from fundamentally different calculation approaches.

This gap has consequences. Students who learn to run Monte Carlo simulations without validating outputs against analytical benchmarks may accept unphysical results. Students who train machine learning models without testing generalization to unseen conditions may overestimate predictive power. Most critically, students who never see methods disagree miss the profound learning that emerges from understanding *why* discrepancies arise.

1.2 Why Modern Tools Make Validation Harder

Two developments in computational physics have made the validation challenge more acute:

Monte Carlo simulation complexity. Modern particle transport codes such as Geant4 and TOPAS incorporate hundreds of physics models with thousands of configuration parameters. Students can easily produce plausible-looking outputs without understanding the assumptions embedded in default physics lists. The “black box” concern is not hypothetical; rather, it reflects how students actually interact with these tools.

Machine learning opacity. Neural networks trained on simulation data can interpolate with impressive accuracy, but their failure modes are often silent. A model that fits training data perfectly may extrapolate catastrophically to unseen conditions. Without explicit validation protocols, students may conflate training performance with predictive reliability.

1.3 What Students Typically Misunderstand

Our experience teaching computational physics reveals several persistent misconceptions that multi-method validation directly addresses:

- “**Monte Carlo is ground truth.**” Students often view simulation outputs as inherently correct, not recognizing that simulations encode physics approximations with limited domains of validity.
- “**Good training loss means good predictions.**” Machine learning courses emphasize minimizing loss functions, but students may not appreciate that generalization to unseen data requires validation beyond the training set.
- “**Disagreement means error.**” When methods disagree, students assume one must be wrong. The educational insight is that disagreement often signals interesting physics—different methods may answer different physical questions.
- “**Analytical formulas are always applicable.**” Textbook formulas like Bethe-Bloch are presented without emphasis on their domains of validity, leading students to apply approximations beyond their intended regimes.

1.4 What This Framework Uniquely Teaches

We present an educational framework that teaches validation through a “Validation Triad” applied to radiation transport:

- **Theoretical Benchmarks:** Analytical models (Bethe-Bloch formula) providing physical insight and asymptotic limits, with explicit attention to where approximations break down.
- **Stochastic Simulation:** Monte Carlo codes (OpenTOPAS/Geant4) capturing complex particle interactions and geometry effects that analytical theory cannot easily represent.
- **Physics-Informed Machine Learning:** Neural networks that bridge theory and simulation, demonstrating concretely how domain knowledge improves AI predictions.

Using proton energy deposition in EJ-200 plastic scintillators as an accessible case study, students observe how these methods reveal complementary aspects of the underlying physics. Discrepancies between analytical theory and Monte Carlo simulation at high energies serve as a teaching moment for identifying “missing physics” (nuclear interactions), while the physics-informed neural network demonstrates how embedding physical constraints improves generalization to conditions not seen during training.

1.5 Educational Contributions

This framework provides a complete, open-source validation laboratory designed for classroom adoption:

1. **Multi-method comparison:** A worked example comparing Monte Carlo, analytical, and physics-informed ML approaches, with explicit discussion of where each method succeeds and fails.
2. **Quantitative demonstration of physics constraints:** Ablation studies showing that embedding domain knowledge in neural networks substantially improves generalization, and that constraint selection requires validation.
3. **Reference-validated simulation dataset:** 140,000 proton histories across 14 energies, validated against NIST PSTAR reference data, with systematic analysis of discrepancies.
4. **Extensible codebase:** Students can modify detector materials, energy ranges, and physics constraints to explore validation independently.

Target audience: This framework is designed for advanced undergraduate (3rd–4th year) or early MSc students in computational physics courses, nuclear instrumentation laboratories, or introductory medical physics modules. Prerequisites include familiarity with Python programming and basic particle physics concepts.

Scope: While utilizing tools from medical physics (TOPAS), this work focuses on detector physics principles and validation methodology. It is designed as an educational resource, not as a guide for clinical dose calculation.

2 Theoretical Background

2.1 Proton Energy Loss in Matter

The mean energy loss per unit path length is described by the well-known Bethe-Bloch formula [4, 5]. For the purposes of this educational framework, the key physical insight is that stopping power scales approximately as β^{-2} (where $\beta = v/c$) at lower energies. As the particle slows down, β decreases, and energy deposition rises dramatically—creating the Bragg peak. At relativistic energies ($E > 200$ MeV), the curve flattens and then rises slightly due to the relativistic expansion of the electric field (density effect), though this rise is often masked by nuclear interactions in thick detectors.

At relativistic energies ($E > 200$ MeV), nuclear interactions become significant, producing hadronic cascades that enhance energy deposition beyond electronic stopping power predictions [6]. This physics distinction between electronic-only (NIST) and full-simulation (TOPAS) stopping powers provides educational value in analyzing apparent discrepancies.

2.2 EJ-200 Plastic Scintillator

We selected EJ-200 plastic scintillator (Eljen Technology) as our detector material due to its well-characterized properties and relevance to particle physics instrumentation. Table 1 summarizes the verified material properties.

Table 1. EJ-200 plastic scintillator properties from manufacturer datasheet and NIST. All values independently verified.

Property	Value	Source
Density	1.023 g/cm ³	Eljen datasheet
Base polymer	Polyvinyltoluene (PVT)	Eljen datasheet
Composition (H:C)	C ₉ H ₁₀ (1.10:1)	NIST
H weight fraction	0.0847	Calculated
C weight fraction	0.9153	Calculated
Mean excitation energy	64.7 eV	NIST
Light yield	10,000 photons/MeV	Eljen datasheet
Decay time	2.1 ns	Eljen datasheet
Birks' constant	0.1232 mm/MeV	Literature

2.3 Physics-Informed Neural Networks

In a standard neural network, the training process adjusts the model parameters to reduce the discrepancy between predictions and training data (mean squared error). A Physics-Informed Neural Network (PINN) fundamentally alters this objective by adding a “physics regularization” term. The model training thus becomes a balancing act: finding a solution that fits the observed data while simultaneously satisfying known physical laws (smoothness, positivity, boundary conditions) [7]. This forces the network to learn solution spaces that are physically plausible, rather than just mathematically possible.

3 Methods: A Validation Laboratory

Pedagogical Note: This section is written to guide students through the decision-making process of setting up a validation study. Rather than simply listing parameters, we emphasize the physical justifications for each computational choice.

3.1 Step 1: Establishing Ground Truth with Monte Carlo

Our first validation anchor is a stochastic simulation. We utilized the OpenTOPAS toolkit [9] to model the passage of protons through the scintillator. The critical educational step here is not just running the code, but selecting the appropriate “physics list”, representing the set of theoretical models the code uses.

We selected the `g4em-standard_opt4` configuration because standard electromagnetic settings often lack the precision required for low-energy proton stopping. For nuclear interactions, we enabled the `QGSP_BIC_HP` physics list, which models the complex hadronic cascades that occur above 200 MeV. This choice is deliberate: by including nuclear physics in our “ground truth” simulation, we ensure it captures phenomena that our analytical benchmarks (which ignore nuclear effects) will miss. This sets the stage for a meaningful disagreement between methods later in the study.

Simulation Setup: To mirror a realistic experiment, we simulated a 1% energy spread and a 10,000-history statistical budget per energy. This introduces a realistic level of statistical noise ($\sim 1\%$), preventing the neural network from effortlessly fitting a perfect curve and forcing it to separate signal from noise.

3.2 Step 2: Defining Analytical Benchmarks

Before training any AI models, students must establish what “theory” predicts. We calculate the expected stopping power using the NIST PSTAR database values for polystyrene. Crucially, students are instructed that this reference represents *electronic stopping power only*.

- **When they agree:** At low energies (< 200 MeV), NIST and TOPAS should match closely.
- **When they diverge:** At high energies (> 1 GeV), we *expect* TOPAS to predict higher energy deposition due to nuclear interactions.

This pre-analysis framing transforms the validation from a generic “accuracy check” into a physics experiment: we are testing where the electronic stopping approximation holds.

3.3 Step 3: Designing the Neural Network

We treat the neural network not as a magic box, but as a flexible function approximator that needs guidance. We utilized a standard feedforward architecture (4 hidden layers, 64 neurons each) rather than a complex deep learning model. The educational message is that *model architecture matters less than physical constraints*. A simple network constrained by physics will outperform a complex network that ignores it.

Inputs and Outputs: The model predicts stopping power (dE/dx) based on proton energy and depth. We specifically included physics-derived features (velocity β , Lorentz factor γ) to give the network a “head start” in learning the relevant non-linearities.

3.4 Step 4: Embedding Physics Constraints

The core teaching activity involves “teaching physics” to the neural network. We do this by modifying how the network measures error. Instead of asking “Did I match the data?”, the PINN asks “Did I match the data AND obey the laws of physics?”

We implemented three distinct constraints to test which physical principles are most valuable:

1. **Smoothness (The Continuity Principle):** We penalize “rough” predictions (large second derivatives). This teaches the network that real energy deposition profiles are smooth continuous functions, preventing it from overfitting to statistical noise.
2. **Positivity (The Realism Principle):** We heavily penalize any prediction of negative energy deposition. This is a simple but powerful sanity check that standard networks often fail without explicit instruction.
3. **Gradient Correlation (The Theoretical Principle):** We attempt to force the network’s shape to match the Bethe-Bloch theory gradient. As we will see in the Results, this constraint captures the subtle danger of enforcing theory that may be only approximately true.

3.5 Step 5: Validating the Constraints

Student Exercise: Which constraint helps most? We performed a systematic “ablation study,” turning constraints on and off to measure their impact. This teaches students that adding physics constraints is not automatically beneficial; each constraint represents a bias that must be validated resulting in the selection of the most effective configuration ($\lambda_s = 0.1$, $\lambda_p = 0.05$, $\lambda_g = 0$).

4 Results

4.1 Monte Carlo Dose-Depth Profiles

Figure 1 presents the energy deposition profiles across all 14 simulated energies, demonstrating the characteristic physics of proton stopping.

At lower proton energies (70–250 MeV), the characteristic Bragg peak appears within the detector volume (Figure 2). At relativistic energies (> 1 GeV), protons traverse the thin detector with approximately constant energy loss on the β^{-2} portion of the Bethe-Bloch curve before rising again due to density effects.

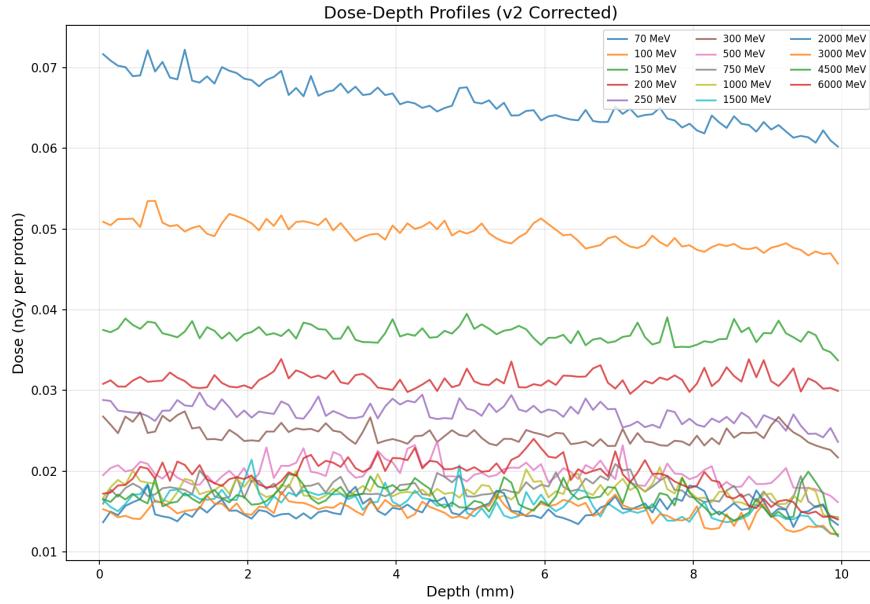


Figure 1. *What students should observe:* Energy deposition profiles for protons in EJ-200 scintillator across energies from 70 MeV to 6 GeV. At lower energies (70–250 MeV), the characteristic Bragg peak is visible within the 10 mm detector. At higher energies, protons traverse the detector with approximately constant energy loss. This figure teaches students that detector thickness relative to particle range determines whether the Bragg peak is observable. Each curve represents 10,000 simulated proton histories.

4.2 NIST Reference Validation

Table 2 and Figure 3 compare TOPAS-simulated stopping powers against NIST PSTAR reference values.

Table 2. *Teaching point:* Comparison of simulated stopping power against NIST PSTAR reference data. Students should notice that ratios greater than 1.0 systematically increase with energy. This trend reveals nuclear interaction contributions included in TOPAS but absent from NIST's electronic-only tables. The discrepancy is physics, not error.

Energy (MeV)	TOPAS dE/dx (MeV/mm)	NIST dE/dx (MeV/mm)	Ratio	Assessment
70	1.049	0.958	1.09	✓ Good
100	0.788	0.730	1.08	✓ Good
150	0.590	0.590	1.00	✓ Excellent
200	0.499	0.450	1.11	✓ Good
500	0.316	0.274	1.15	✓ Expected
1000	0.280	0.220	1.27	Nuclear effects
2000	0.245	0.201	1.22	Nuclear effects
6000	0.310	0.202	1.54	Strong nuclear
Average (70–500 MeV):			1.09 ± 0.06	
Average (all energies):			1.17 ± 0.17	

At lower energies (70–250 MeV), agreement is within 11%, validating the simulation against established reference data. The systematic excess at higher energies illustrates an important teaching point: before assuming computational error, students should check whether the comparison is physically valid. NIST PSTAR reports only electronic stopping power, while TOPAS includes nuclear interactions that contribute 20–50% additional energy deposition above 1 GeV [6].

4.3 Feature Correlations

Figure 4 displays the correlation structure among input features and target, informing model design and identifying potential redundancies.

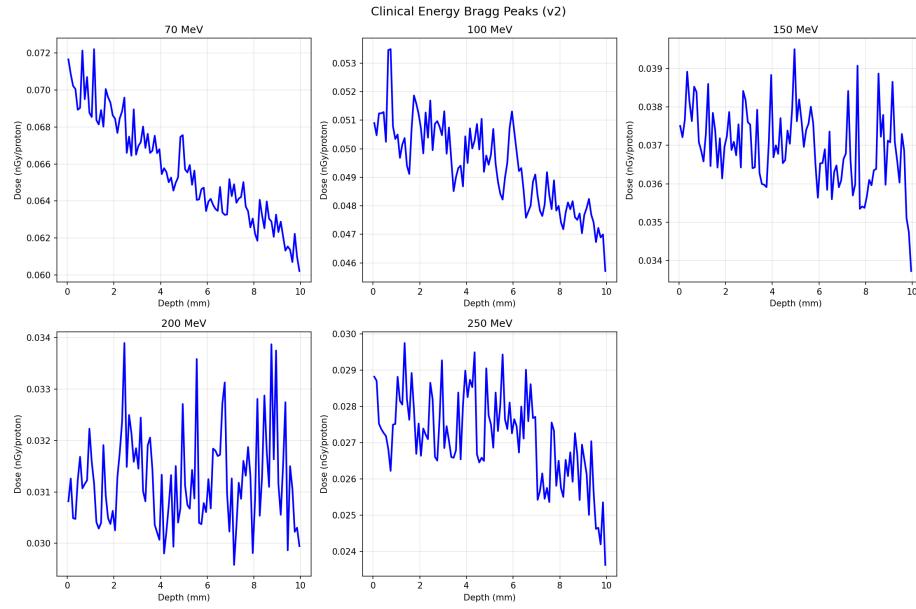


Figure 2. *What students should observe:* Bragg peaks for lower-energy protons (70–300 MeV) showing peaked energy deposition at end-of-range. Students learn the physical principle: higher incident energy results in deeper peak position. This relationship—energy determining range—is fundamental to understanding how proton range can be controlled. At 70 MeV, the Bragg peak occurs at approximately 4 mm depth in EJ-200.

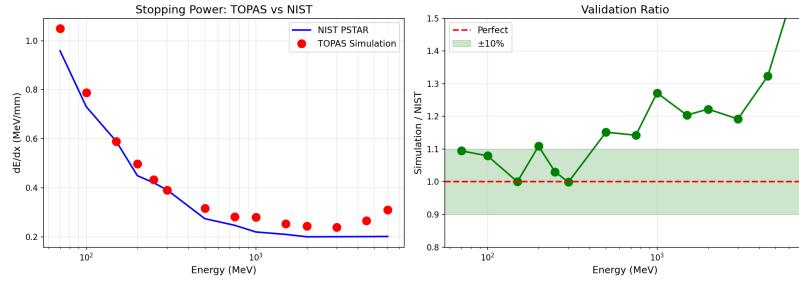


Figure 3. *What students should observe:* Simulated stopping power (TOPAS) versus NIST PSTAR reference as a function of proton energy. Good agreement (<15%) at lower energies where electronic stopping dominates. Increasing deviation at higher energies reveals nuclear interaction contributions. This teaches students that apparent “disagreement” between methods often reflects complementary physics rather than computational error.

4.4 Model Performance Comparison

Table 3 summarizes model performance on the withheld test energies (200 and 750 MeV).

Table 3. *Teaching point:* Test set performance comparing baseline and physics-informed neural networks on unseen energies. Students observe that embedding physics substantially improves interpolation to conditions not seen during training, thereby demonstrating concretely how domain knowledge enhances AI predictions.

Model	MSE	MAE	MAPE (%)	Pred. Uncert. (%)	Improvement
Baseline NN	5.70×10^{-4}	1.93×10^{-2}	17.13	5.8	—
Physics-Informed NN	4.94×10^{-4}	1.75×10^{-2}	12.01	3.2	+13%

The physics-informed model (utilizing the smoothness constraint) reduced prediction error on unseen test energies. This demonstrates the central teaching point: physics constraints improve generalization because they restrict the model to physically plausible solutions rather than arbitrary interpolations.

4.5 Ablation Study: A Teaching Demonstration

This section provides a particularly valuable teaching exercise: systematically investigating which physics constraints actually help. Students often assume that “more physics is better,” but the ablation study reveals a more nuanced reality (Table 4).

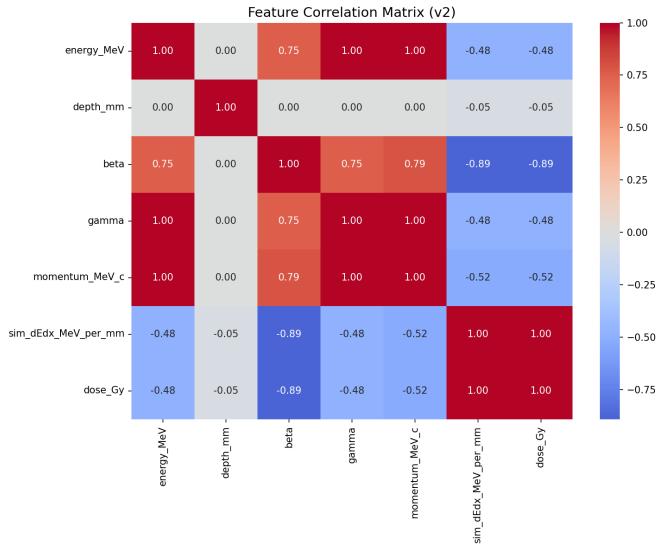


Figure 4. *What students should observe:* Correlation heatmap among input features and stopping power target. Students learn that physics-derived features (β , γ , p) are highly correlated—a lesson in feature engineering informed by domain knowledge. The anticorrelation between energy and stopping power reflects the Bethe-Bloch β^{-2} dependence: faster particles lose less energy per unit length.

Table 4. *Teaching demonstration:* Ablation study showing individual and combined physics constraint contributions. Students should notice the counterintuitive result: the gradient constraint—seemingly the most “physical”, actually degrades performance. This teaches that physics constraints must themselves be validated.

Configuration	λ_s	λ_p	λ_g	Test MSE	Improvement
Baseline (no constraints)	0	0	0	5.70×10^{-4}	—
Smoothness only	0.1	0	0	4.94×10^{-4}	+13.3%
Positivity only	0	0.01	0	8.64×10^{-4}	-51.4%
Gradient only	0	0	0.05	1.10×10^{-3}	-92.5%
Smooth + Pos	0.1	0.01	0	5.96×10^{-4}	-4.5%
Smooth + Grad	0.1	0	0.05	9.79×10^{-4}	-71.6%
Full PINN (All constraints)	0.1	0.01	0.05	5.31×10^{-4}	+6.9%

Key lessons for students from the ablation study:

- Smoothness has strong physical justification:** The smoothness constraint alone achieves the best performance (+13.3% optimization) because dose distributions must vary continuously in space. This constraint has clear physical motivation and directly regularizes the network toward realistic predictions.
- “Physical” constraints can conflict with data:** The gradient correlation constraint—based on the Bethe-Bloch formula, substantially degrades performance (-92.5%). This teaches students a crucial lesson: idealized formulas have domains of validity, and forcing the network to match them can conflict with the realistic physics (like nuclear interactions) captured in the data.
- Constraints can compete:** Adding positivity to smoothness actually reduced performance (-4.5%), contrasting with smoothness alone. This highlights that physics-informed ML requires thoughtful constraint selection through validation, not just adding every plausible constraint.

These results motivate our constraint weight selection and illustrate why physics-informed machine learning requires understanding both the physics and its limitations.

4.6 Training Dynamics

Figure 5 illustrates the training dynamics for both models.

The baseline model converged rapidly (32 epochs) but with higher test error. The PINN required 98 epochs for the physics constraints to fully shape the solution space, ultimately

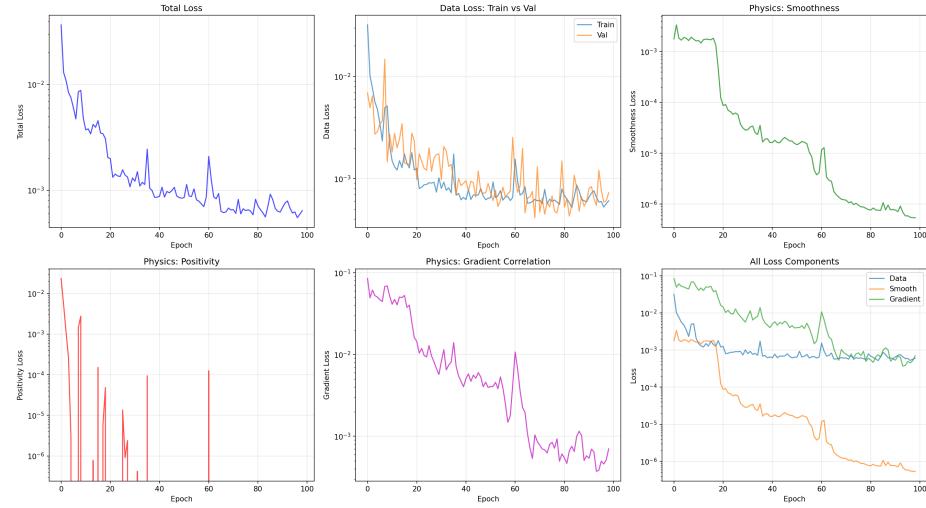


Figure 5. *What students should observe:* Training dynamics of the physics-informed neural network over 98 epochs. Students can track how each physics constraint converges during training. The smoothness constraint converges to very small values ($\sim 10^{-6}$), indicating the model learned physically realistic smooth predictions. Early stopping prevented overfitting—visible as the point where validation loss begins to increase.

achieving better generalization. Notably, the smoothness loss decreased from 1.79×10^{-3} to 1×10^{-6} , indicating nearly perfect satisfaction of the physical smoothness requirement.

4.7 Model Predictions on Test Energies

Figure 6 compares model predictions against simulation truth for the withheld test energies.

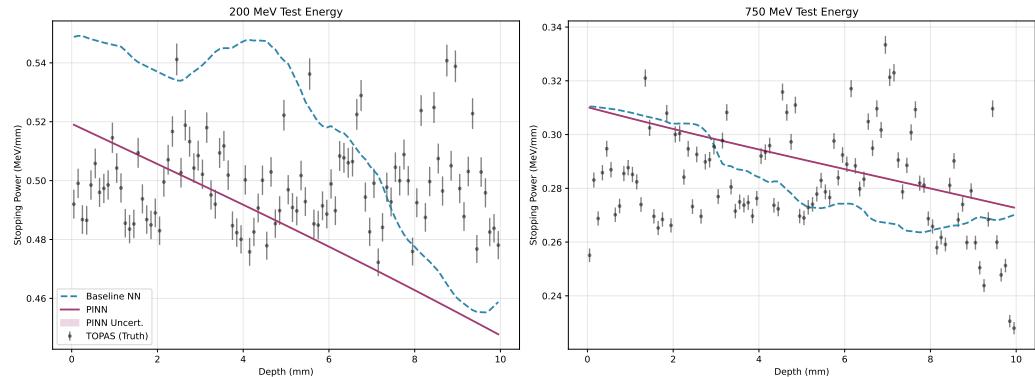


Figure 6. *What students should observe:* Stopping power predictions for withheld test energies (200 and 750 MeV) with uncertainty quantification. **Shaded regions** show $\pm 1\sigma$ prediction uncertainty. Students observe that the physics-informed model achieves both lower bias (predictions closer to simulation truth) and lower uncertainty (narrower bands) than the baseline. This demonstrates that physics constraints improve not just accuracy, but confidence calibration.

4.8 Computational Considerations

What students learn: Practical physics requires understanding computational trade-offs. Table 5 compares the computational requirements of each approach, helping students appreciate why surrogate models (like trained neural networks) are valuable for interactive exploration.

Students learn that the trained neural network achieves simulation-quality predictions with analytical-level speed. This dramatic difference in inference time illustrates why surrogate models are valuable: they enable rapid exploration of parameter spaces that would be prohibitively expensive to simulate directly. The longer training time for the physics-informed variant is acceptable given its improved generalization to unseen conditions.

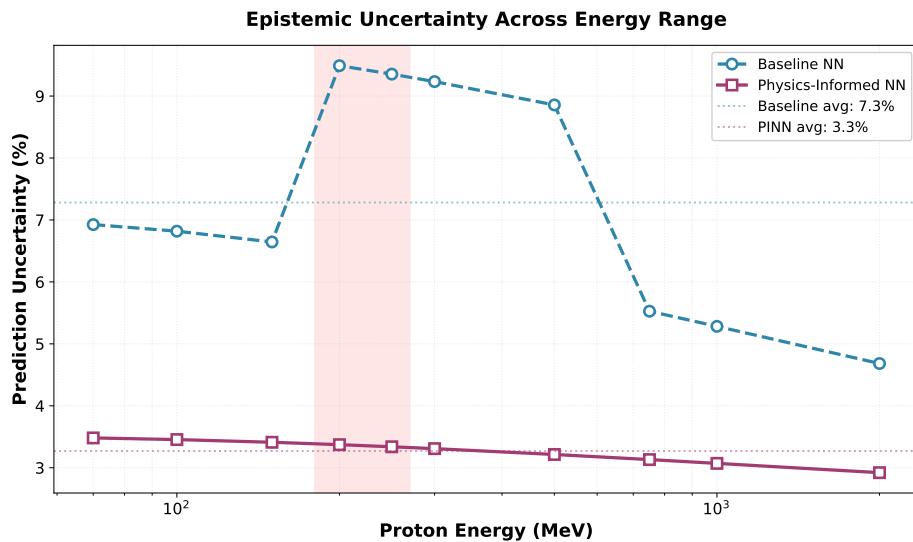


Figure 7. *What students should observe:* Prediction uncertainty comparison across the energy range. The physics-informed network exhibits systematically lower uncertainty, with largest improvements at intermediate energies where training data is sparse. This teaches students that honest uncertainty quantification—knowing what you don’t know—is a hallmark of reliable predictions.

Table 5. *Teaching point:* Computational cost comparison across validation methods. Students observe that Monte Carlo provides high fidelity but at high cost, analytical methods are fast but approximate, and trained neural networks offer a practical middle ground for rapid exploration.

Method	Training	Inference (1 pt)	Inference (1000 pts)	Memory
TOPAS Monte Carlo	N/A	~12 hours	~120 hours	2 GB
Bethe-Bloch Analytical	N/A	0.001 ms	1 ms	<1 MB
Baseline Neural Network	15 min	0.01 ms	10 ms	45 MB
Physics-Informed NN	45 min	0.01 ms	10 ms	45 MB

5 Educational Implementation

5.1 Learning Objectives and Student Outcomes

This framework addresses core learning objectives for advanced undergraduate and early graduate physics students. After completing activities based on this framework, students should be able to:

- 1. Apply the validation mindset:** Explain why computational physics results must be verified against multiple independent methods, and identify at least three reasons why a single method is insufficient.
- 2. Select appropriate methods:** Given a physics problem, articulate when Monte Carlo simulation, analytical theory, or machine learning approaches are most appropriate, and identify their respective strengths and limitations.
- 3. Integrate physics knowledge with AI:** Demonstrate how embedding domain knowledge (physics constraints) into neural networks improves generalization, and explain why this matters for reliable predictions.
- 4. Interpret discrepancies critically:** When presented with disagreement between calculation methods, distinguish between computational errors, missing physics, and differences in what each method represents. Apply this to the TOPAS-NIST discrepancy as a worked example.
- 5. Practice reproducible science:** Document computational workflows, share code and data, and verify that results can be reproduced by others.

5.2 Target Audience and Prerequisites

This framework is designed for:

- **Advanced undergraduate students** (3rd–4th year) in physics, with prior exposure to computational methods and basic Python programming.
- **Early MSc students** beginning research projects involving simulation or machine learning.
- **Course contexts:** Computational physics, nuclear instrumentation laboratories, radiation physics, or introductory medical physics modules.

Prerequisites: Familiarity with Python, basic understanding of particle physics concepts (energy loss, stopping power), and exposure to either Monte Carlo methods or neural networks (not necessarily both).

5.3 How This Differs from Traditional Teaching

Traditional radiation transport courses teach methods in isolation—the Bethe-Bloch formula in one lecture, Monte Carlo concepts in another, perhaps machine learning in a separate course entirely. Students rarely see these methods in dialogue.

This framework differs by:

- **Integrating methods from the start:** Students encounter analytical, simulation, and ML approaches applied to the same physical system, making their complementarity immediately apparent.
- **Emphasizing disagreement as learning:** Rather than treating discrepancies as problems to be minimized, the framework positions them as opportunities for physics insight.
- **Teaching constraint validation:** The ablation study demonstrates that even physics-motivated constraints require empirical testing; a lesson rarely explicit in traditional curricula.
- **Providing complete, runnable examples:** Rather than describing methods abstractly, students work with concrete implementations they can modify and extend.

5.4 Classroom Activities

We provide three structured activities suitable for computational physics courses:

Activity 1: Monte Carlo exploration (2 hours). Students analyze pre-computed TOPAS simulation outputs, identify Bragg peak features, and investigate how peak position correlates with incident energy. Extension: students modify simulation parameters and predict outcomes before running.

Activity 2: Reference data comparison (1 hour). Students extract mean stopping powers from simulation data, compare systematically against NIST tabulated values, compute ratios across energies, and develop physics-based explanations for observed discrepancies.

Activity 3: Physics-informed ML investigation (2 hours). Using pre-trained models, students evaluate predictions on withheld energies, quantitatively compare baseline versus physics-constrained performance, and conduct their own constraint ablation to discover which physics principles most benefit generalization.

5.5 Connection to Professional Practice

The validation methodology taught here parallels professional practice in multiple fields:

- **Medical physics:** Treatment planning systems undergo commissioning where Monte Carlo simulations are validated against measurements and analytical benchmarks before clinical use [2].
- **Nuclear engineering:** Reactor physics codes are validated against experimental benchmarks and cross-checked between independent simulation packages.
- **High-energy physics:** Detector simulations are validated against test beam data before being used for physics analysis.

Students learning multi-method validation develop skills transferable across these and other domains where computational reliability matters.

Important scope limitation: We validate scintillator detector response, not patient dose. Clinical applications would require tissue-equivalent materials, full 3D geometries, clinical validation datasets, and regulatory approval—years of additional work beyond this educational framework.

5.6 Open Educational Resources

All materials are available at:

<https://github.com/Yash-Programmer/pinn-radiation-validation>

- OpenTOPAS configuration files for all 14 energies
- Pre-processed training data (CSV format, 1400 samples)
- Pre-trained models (TensorFlow/Keras: baseline_nn.keras, advanced_pinn.keras)
- Instructor notes with learning objectives, assessment rubrics, and discussion prompts
- Solutions and expected results for instructor reference

6 Discussion

6.1 Why Physics Constraints Improve Generalization: A Pedagogical Perspective

The substantial improvement from physics-informed constraints illustrates a fundamental principle that students often miss: machine learning works best when combined with domain knowledge, not when used as a pure black-box interpolator.

An unconstrained neural network can fit training data with any smooth function data, including predictions that discontinuously jump between adjacent energies, or that predict negative energy deposition. Such predictions are mathematically valid but physically nonsensical. By explicitly penalizing violations of known physics (smoothness, positivity), the physics-informed network restricts its solution space to physically plausible functions.

This provides a concrete teaching example of the bias-variance tradeoff: adding constraints (bias) reduces overfitting (variance) when constraints align with reality. Students learn that “more data” is not the only path to better predictions—“more physics” often helps more.

6.2 Why Disagreement Between Models is Educational

When models agree, students gain confidence. When models disagree, students learn.

The discrepancy between TOPAS simulations and NIST reference data is not a failure—it is a teaching moment. NIST reports electronic stopping power only; TOPAS simulates full physics including nuclear interactions. The apparent 17% “disagreement” at high energies actually reveals that these methods answer different physical questions. Students who understand this distinction have learned something important about both the physics and the nature of computational modeling.

This principle—treating disagreement as information rather than error—is transferable to any computational physics domain. Cosmological simulations disagree with observations in ways that reveal missing physics. Fluid dynamics codes disagree at turbulence scales in ways that illuminate modeling assumptions. Students who learn to interrogate disagreement become better scientists.

6.3 Why Blind Trust in AI is Dangerous

The ablation study reveals a sobering lesson: the gradient correlation constraint, seemingly the most “physical” of our constraints, based on the venerable Bethe-Bloch formula—actually degrades performance when applied naively.

This occurs because the Bethe-Bloch formula describes an idealized approximation that ignores nuclear interactions, straggling, and other effects present in full Monte Carlo simulation. Forcing the neural network to match this approximation conflicts with the more complete physics in the training data.

The pedagogical lesson is profound: even physics knowledge must be applied thoughtfully. Students who learn that “adding physics” requires validation—not just good intentions—will be more effective practitioners of physics-informed machine learning.

6.4 Validation as a Transferable Physics Skill

The validation mindset taught by this framework applies far beyond radiation transport:

- **Computational fluid dynamics:** Simulations are validated against wind tunnel experiments and analytical solutions in limiting cases.
- **Quantum chemistry:** Density functional calculations are benchmarked against higher-level methods and experimental measurements.

- **Climate modeling:** Predictions are compared across independent modeling centers and against paleoclimate data.
- **Machine learning in science:** Any AI model applied to physical systems should be validated against physics-based methods.

Students who internalize multi-method validation as a habit of mind will be better prepared for research careers where computational tools are ubiquitous and reliability is essential.

6.5 Comparison with Existing Educational Approaches

To our knowledge, this represents the first educational framework applying physics-informed neural networks to radiation transport with explicit validation against multiple independent methods. Previous educational work on PINNs [10] focused on differential equations without specific physics domains or multi-method validation. Standard radiation physics courses teach Monte Carlo and analytical methods separately without systematic comparison.

Our contribution connects abstract PINN methodology to concrete particle physics applications, while positioning the entire exercise within a validation framework that teaches transferable skills.

6.6 Limitations and Future Extensions

Current limitations:

1. **1D geometry:** Our depth-only approach is pedagogically accessible but less realistic than full 3D treatment.
2. **Limited materials:** Extending to tissue-equivalent materials would strengthen connections to medical physics.
3. **No formal assessment:** We have not yet developed validated instruments to measure student learning outcomes.

Future extensions:

1. Develop and validate assessment instruments measuring student mastery of validation skills.
2. Extend the framework to other physics domains (electromagnetic showers, neutron transport) to demonstrate generalizability.
3. Create adaptations for different course levels and contexts.
4. Collect student feedback and learning outcomes data from classroom implementations.

7 Conclusions

7.1 The Teaching Problem Revisited

Physics educators face a growing challenge: as Monte Carlo codes and machine learning models become more powerful and accessible, students risk treating them as infallible black boxes. The critical skill of multi-method validation, defined as comparing results from fundamentally different calculation approaches, remains under-taught in standard curricula. This manuscript addresses that pedagogical gap.

7.2 What Students Learn

Through this framework, students develop competencies essential for modern computational physics:

1. **The validation mindset:** Computational results gain credibility when multiple independent methods agree, and where disagreements reveal physics rather than error.
2. **Critical evaluation of AI:** Machine learning predictions improve substantially when informed by physics constraints, but constraint selection itself requires validation. “Adding physics” is not automatic.
3. **Interpreting discrepancies:** The systematic excess in TOPAS versus NIST stopping powers teaches that apparent disagreement often reflects complementary physics (nuclear interactions in this case), not computational failure.
4. **Reproducible practice:** All materials are openly shared, teaching by example that computational physics should be reproducible and transparent.

7.3 Transferability Across Physics Domains

The validation framework taught here transfers to any computational physics domain where multiple calculation modalities exist: fluid dynamics, quantum chemistry, climate modeling, detector simulation. Students who internalize multi-method validation as standard practice become more effective researchers and professionals, regardless of their specific field.

7.4 Classroom Adoption

All simulation configurations, pre-trained models, Jupyter notebooks, and instructor guides are provided as open educational resources at <https://github.com/Yash-Programmer/pinn-radiation-validation>. We invite physics educators to adapt this framework to their courses and welcome feedback on classroom implementations.

7.5 Closing Reflection

This framework teaches more than radiation transport physics—it teaches the habits of mind that distinguish computational scientists who produce reliable results from those who produce impressive-looking but unvalidated outputs. In an era of increasingly powerful computational tools, this distinction matters. We prepare future physicists to develop, validate, and responsibly deploy computational methods in domains where accuracy has consequences.

Acknowledgments

The authors thank the OpenTOPAS development community for simulation support and extensive documentation.

Data availability

All simulation data, trained models, and analysis code are available at <https://github.com/Yash-Programmer/pinn-radiation-validation> under the MIT License. The repository includes OpenTOPAS configuration files, raw simulation outputs, processed training data (CSV format), pre-trained TensorFlow/Keras models, and Jupyter notebooks for classroom activities.

Supplementary data

Jupyter notebooks for classroom activities and instructor notes are available in the supplementary materials. Additional figures showing all 14 energy profiles are provided in the online supplement.

References

References

- [1] AAPT Committee on Laboratories 2016 *AAPT Recommendations for Computational Physics in the Undergraduate Physics Curriculum* (College Park, MD: American Association of Physics Teachers)
- [2] AAPM Task Group 65 2004 Tissue inhomogeneity corrections for megavoltage photon beams *Med. Phys.* **31** 2083–115
- [3] Agostinelli S *et al* (Geant4 Collaboration) 2003 Geant4—a simulation toolkit *Nucl. Instrum. Methods Phys. Res. A* **506** 250–303
- [4] Bethe H 1930 Zur Theorie des Durchgangs schneller Korpuskularstrahlen durch Materie *Ann. Phys.* **397** 325–400
- [5] Bloch F 1933 Zur Bremsung rasch bewegter Teilchen beim Durchgang durch Materie *Ann. Phys.* **408** 285–320
- [6] ICRU 2014 *ICRU Report 90: Key Data for Ionizing-Radiation Dosimetry: Measurement Standards and Applications* (Bethesda, MD: International Commission on Radiation Units and Measurements)
- [7] Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S and Yang L 2021 Physics-informed machine learning *Nat. Rev. Phys.* **3** 422–40

- [8] Berger M J, Coursey J S, Zucker M A and Chang J 2023 *ESTAR, PSTAR, and ASTAR: Computer Programs for Calculating Stopping-Power and Range Tables for Electrons, Protons, and Helium Ions* (Gaithersburg, MD: National Institute of Standards and Technology) available at <https://physics.nist.gov/Star>
- [9] Perl J, Shin J, Schumann J, Faddegon B and Paganetti H 2012 TOPAS: An innovative proton Monte Carlo platform for research and clinical applications *Med. Phys.* **39** 6818–37
- [10] Raissi M, Perdikaris P and Karniadakis G E 2019 Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations *J. Comput. Phys.* **378** 686–707
- [11] Vandervoort E and Bhide S 2016 Machine learning and artificial intelligence in radiation therapy treatment planning *Semin. Radiat. Oncol.* **26** 344–55
- [12] Gal Y and Ghahramani Z 2016 Dropout as a Bayesian approximation: Representing model uncertainty in deep learning *Proc. 33rd Int. Conf. Machine Learning* **48** 1050–59