Uncertainty-Aware Classifier with Physics-Based Rejection (UA-PBR): A Proof-of-Concept Under Computational Constraints—Revised Version

Mohsen Mostafa *

Posted Date: 11 March 2026

doi: 10.20944/preprints202603.0748.v2

Keywords: physics-informed machine learning; Bayesian deep learning; reject option classification; out-of-distribution detection; scientific machine learning; partial differential equations (PDEs); Darcy flow; uncertainty quantification; robustness

 doi:10.20944/preprints202603.0748.v2

Article

Uncertainty-Aware Classifier with Physics-Based Rejection (UA-PBR): A Proof-of-Concept Under Computational Constraints—Revised Version †

Mohsen Mostafa

Independent Researcher; mohsen.mostafa.ai@outlook.com

† **Version Note:** This is a revised version of the preprint originally posted in February 2026. Based on feedback and further reflection, we have clarified experimental constraints and adjusted claims accordingly. All experiments were conducted on a free Google Colab GPU (NVIDIA T4, 16GB VRAM) with limited computational resources; results should be interpreted as preliminary proof-of-concept evidence.

Abstract

Deep learning classifiers deployed in scientific applications often encounter inputs that violate physical laws (e.g., due to sensor failure or corruption). Standard methods cannot detect such violations and may produce confident but wrong predictions. We propose UA-PBR, a framework that combines a physics-informed autoencoder (to detect physics violations) with a Bayesian CNN (to quantify predictive uncertainty). Inputs are rejected if either the PDE residual exceeds a threshold or the predictive entropy is too high. As a proof-of-concept, we evaluate UA-PBR on a synthetic Darcy flow dataset (32×32 grid) under severe computational constraints (Google Colab, 10 seeds). Despite these limitations, UA-PBR reduces classification risk by over 90% on heavily corrupted samples while accepting 89.7% of clean inputs with 99.99% accuracy on accepted samples. Ablation studies confirm that both components contribute synergistically. These preliminary results on a synthetic benchmark illustrate the potential of physics-aware rejection and motivate further investigation with larger-scale experiments. Code is available at: https://github.com/UA-PBR/UA-PBR.

Keywords: physics-informed machine learning; Bayesian deep learning; reject option classification; out-of-distribution detection; scientific machine learning; partial differential equations (PDEs); Darcy flow; uncertainty quantification; robustness

1. Introduction

Neural networks are increasingly used in scientific domains where data must satisfy physical laws (e.g., partial differential equations). However, test-time inputs may be corrupted by sensor noise or artifacts, leading to physically impossible observations. Standard classifiers have no mechanism to detect such violations and will produce predictions regardless, potentially with high confidence.

1.1. The Silent Failure Problem

Consider a classifier trained on clean pressure fields satisfying Darcy's law, $-\nabla \cdot (a\nabla u) = f$. When presented with a corrupted observation $u_{\text{obs}} = u^* + \phi$, where ϕ represents sensor noise or artifacts, the network has no mechanism to detect this violation. It produces a prediction regardless, potentially with high confidence, even when the input is physically impossible. Hendrycks and Dietterich [6] quantified this vulnerability, showing that standard classifiers experience accuracy drops of 20-40% on corrupted versions of CIFAR-10. In scientific applications, the consequences are more severe: Krishnapriyan et al. [7] demonstrated that physics-informed neural networks fail catastrophically when training data violates PDE constraints, yet they have no mechanism to detect such violations at inference time.

 doi:10.20944/preprints202603.0748.v2

1.2. Fragmentation of Existing Solutions

Existing approaches address robustness in isolation: normalization methods [9,10] adapt to noise but cannot detect physics violations; Bayesian networks [12,13] quantify uncertainty but ignore domain knowledge; physics-informed learning [1,16] embeds constraints during training but offers no rejection mechanism at inference. Each approach addresses part of the problem, but a unified framework that leverages both physics and uncertainty for rejection remains relatively unexplored.

1.3. This Revision

In the original version of this preprint (February 2026), we presented UA-PBR with claims of provable robustness bounds and state-of-the-art performance. Based on feedback and further reflection, we have revised the manuscript to more accurately reflect the scope and limitations of our work. The core contribution remains: a framework that integrates physics-informed filtering with Bayesian uncertainty for rejection. However, we now explicitly acknowledge that our experiments were conducted under severe computational constraints (free Google Colab, synthetic Darcy flow dataset, limited epochs) and should be interpreted as **preliminary proof-of-concept evidence** rather than definitive claims of superiority.

Despite these constraints, UA-PBR consistently reduced risk across all seeds in our experiments. The consistency suggests the approach is robust and merits further investigation with greater computational resources and more realistic datasets.

1.4. Our Contributions

Our contributions are threefold:

1. **Theoretical framework:** We provide simple theoretical bounds linking PDE residuals to reconstruction error and bounding expected risk under Lipschitz continuity, though these bounds are not tight and serve primarily as conceptual motivation.
2. **Two-stage rejection architecture:** UA-PBR combines a physics-informed autoencoder (to detect physics violations) with a Bayesian CNN (to quantify predictive uncertainty) in a unified decision-theoretic framework.
3. **Proof-of-concept validation:** Under severe computational constraints (Google Colab, synthetic data), we demonstrate that the approach can dramatically reduce risk on corrupted inputs, motivating further research.

1.5. Organization

The paper is organized as follows. Section 2 discusses related work. Section 3 describes the methodology. Section 4 presents experimental results with explicit discussion of computational limitations. Section 5 discusses limitations and future work. Section 6 concludes.

2. Related Work

2.1. Normalization and Robustness

Normalization methods [9,10] stabilize training but assume clean distributions and treat all inputs uniformly—on corrupted CIFAR-10-C, they still suffer 20-40% accuracy drops [6]. Adaptive methods like Batch Renormalization [11] modulate statistics but cannot detect physics violations.

2.2. Bayesian Deep Learning

Bayesian neural networks [12,13] provide uncertainty estimates but add computational cost and ignore domain knowledge. Monte Carlo Dropout [14] offers a practical approximation, while deep ensembles [15] often outperform more complex Bayesian methods. However, these methods operate purely in data space and cannot distinguish between uncertainty from lack of data and uncertainty from physical impossibility.

 doi:10.20944/preprints202603.0748.v2

2.3. Physics-Informed Machine Learning

Physics-Informed Neural Networks (PINNs) [1,16] embed governing equations into the loss function, achieving high accuracy on benchmark PDEs. Neural operators [17,18] extend this to learning mappings between function spaces. However, these methods assume test data satisfies the same physics as training data and offer no rejection mechanism at inference.

2.4. Out-of-Distribution Detection

OOD detection methods [19,20] identify inputs that differ from the training distribution using techniques like ODIN [21] or Mahalanobis distance [22]. These methods cannot distinguish between benign distribution shift and physics-violating corruptions.

2.5. Learning with Rejection

Classical reject option classifiers [23,24] provide theoretical foundations for abstention but assume known class-conditional distributions and do not incorporate physical constraints. Selective classification [25] and learning with rejection [26] have been studied, but these methods lack mechanisms to leverage domain-specific knowledge.

2.6. Our Contribution

UA-PBR explores the underexplored direction of combining physics-informed filtering with Bayesian uncertainty for rejection. Unlike prior work that treats these signals separately, we investigate whether their integration can yield synergistic improvements. This study is a proof-of-concept under constrained resources, not a claim of deployment readiness.

3. Methodology

3.1. Problem Setup

We consider a Darcy flow governed by $-\nabla \cdot (a\nabla u) = f$ with permeability field a and pressure field u. Inputs are observed pressure fields $u_{\text{obs}} = u + \phi$, where ϕ is corruption. The task is to classify whether the mean permeability is high or low.

The Darcy flow operator satisfies a coercivity property: there exists $\alpha > 0$ such that

$$\|\mathcal{N}_a[u] - f\|_{H^{-1}} \geq \alpha \|u - \mathcal{F}(a)\|_{L^2} \quad \forall u, \tag{1}$$

where $\mathcal{F}(a)$ is the solution operator. This property ensures that small PDE residuals imply small errors in the state variable.

3.2. Stage 1: Physics-Based Corruption Detection

We train an autoencoder Ψ_θ to reconstruct (u, a) from u_{obs} using a physics-informed loss:

$$\mathcal{L} = \|u - \hat{u}\|^2 + \|a - \hat{a}\|^2 + \beta \|\mathcal{N}_{\hat{a}}[\hat{u}] - f\|^2. \tag{2}$$

At test time, we compute the physics score:

$$S_{\text{phy}}(u_{\text{obs}}) = \|\mathcal{N}_{\hat{a}}[\hat{u}] - f\|. \tag{3}$$

Inputs with $S_{\text{phy}} > \tau_{\text{phy}}$ are rejected as physics-violating.

3.3. Stage 2: Bayesian CNN Uncertainty

We use a CNN with Monte Carlo Dropout (50 samples) to obtain predictive distribution $p(y \mid u_{\text{obs}})$. Predictive entropy serves as uncertainty score:

$$U(u_{\text{obs}}) = H[p(y \mid u_{\text{obs}})] = -\sum_{k=1}^{K} p_k \log p_k. \tag{4}$$

 doi:10.20944/preprints202603.0748.v2

Inputs with $U > \tau_{\text{unc}}$ are rejected due to model uncertainty.

3.4. Joint Rejection Rule

The final decision rule integrates both signals:

$$q(u_{\text{obs}}) = \begin{cases} \text{reject} & \text{if } S_{\text{phy}} > \tau_{\text{phy}} \text{ or } U > \tau_{\text{unc}}, \\ \hat{y} & \text{otherwise.} \end{cases} \tag{5}$$

Thresholds are selected on a validation set to minimize risk with rejection cost λ, defined by the loss function:

$$\ell_\lambda(y, \hat{y}) = \begin{cases} 0, & \hat{y} = y, \\ 1, & \hat{y} \neq y, \hat{y} \in \mathcal{Y}, \\ \lambda, & \hat{y} = \text{reject}. \end{cases} \tag{6}$$

3.5. Theoretical Bounds (Conceptual)

We provide simple theoretical bounds that serve as conceptual motivation:

Theorem 3.1 (Error Bound). *For any input with physics score $S_{\text{phy}}(u_{\text{obs}}) \leq \tau_{\text{phy}}$, the reconstruction error satisfies*

$$\|u_{\text{obs}} - u^*\|_{L^2} \leq \frac{\tau_{\text{phy}}}{\alpha} + \|\phi\|_{L^2} + \gamma_n, \tag{7}$$

where α is the coercivity constant and γ_n is the autoencoder approximation error.

Proof. Let $\hat{u}$ be the autoencoder reconstruction of u_{obs}. By coercivity, $\|\hat{u} - u^*\|_{L^2} \leq \frac{1}{\alpha}\|\mathcal{N}_a[\hat{u}] - f\|_{H^{-1}} \leq \frac{\tau_{\text{phy}}}{\alpha}$. The triangle inequality gives $\|u_{\text{obs}} - u^*\| \leq \|u_{\text{obs}} - \hat{u}\| + \|\hat{u} - u^*\| \leq \|\phi\| + \gamma_n + \frac{\tau_{\text{phy}}}{\alpha}$. □

Theorem 3.2 (Risk Bound). *Under Lipschitz continuity of the classifier with constant L, the expected risk satisfies*

$$R_\lambda(q) \leq \lambda + \epsilon_0 + L\delta, \tag{8}$$

where $\delta = \tau_{\text{phy}}/\alpha + \gamma_n$ and ϵ_0 is the clean-data error rate.

Theorem 3.3 (Existence of Optimal Thresholds). *Given a finite validation set, the empirical risk is piecewise constant in $(\tau_{\text{phy}}, \tau_{\text{unc}})$; hence a minimizer exists.*

These bounds are intended as conceptual motivation rather than tight guarantees.

4. Experiments

4.1. Computational Resources

All experiments were conducted on a free Google Colab instance with an NVIDIA T4 GPU (16GB VRAM) and approximately 15GB system RAM. Due to session time limits (12 hours maximum runtime) and the need to share resources with other users, we limited training epochs and used 10 independent seeds for statistical power. The Darcy flow dataset is synthetic (10,000 samples at 32×32 resolution) and should not be interpreted as representative of real-world complexity. These constraints mean our results should be viewed as preliminary proof-of-concept evidence.

4.2. Dataset

We evaluate on the Darcy flow benchmark, generating 10,000 samples at 32×32 resolution with binary classification labels based on mean permeability. Permeability fields are log-normal with spatial correlation, and pressure fields are solved via finite differences. The dataset is split 70/15/15 for train/val/test.

 doi:10.20944/preprints202603.0748.v2

4.3. Corruption Types

We simulate four realistic corruptions at severity levels 0.1, 0.3, 0.5, 0.7, 0.9:

- **Gaussian noise:** Additive white noise $\mathcal{N}(0, \sigma^2)$
- **Salt-and-pepper:** Random pixels set to ± 2 with probability p
- **Structured artifacts:** 8×8 blocks replaced with random values
- **Physics-violating:** Non-solenoidal components added via curl of random vector field

4.4. Architecture

- **Physics autoencoder:** CNN with 4 conv layers (channels $32 \to 64 \to 128 \to 256$), latent dim 256, decoder with 4 transpose conv layers, ReLU activation with batch norm, physics loss weight $\beta = 0.1$.
- **Bayesian CNN:** 4 conv layers ($32 \to 64 \to 128 \to 256$), adaptive average pooling (4×4), 2 fully connected layers ($256 \to 128 \to 2$), dropout rate 0.3, MC samples 50.
- **Standard CNN:** Same architecture without dropout, dropout rate 0.5 for regularization.

4.5. Training

All models trained with AdamW [37] (lr=10^{-3}, weight decay=10^{-4}) for 150 (autoencoder) and 200 (CNN) epochs. Gradient clipping at 1.0, ReduceLROnPlateau scheduler (patience=10, factor=0.5). Experiments run with 10 independent seeds.

4.6. Baselines

We compare against:

1. Standard CNN (no rejection)
2. MaxProb rejection (threshold on maximum softmax probability)
3. Deep Ensemble (3 models)
4. Physics-only rejection (no uncertainty)
5. Uncertainty-only rejection (no physics)

4.7. Results

Key findings:

- UA-PBR dramatically reduces risk on severe corruptions (severity 0.9) while maintaining low risk on clean data.
- Acceptance rates on clean data are 90% with near-perfect accuracy on accepted samples.
- The physics filter perfectly separates clean from corrupted inputs ($\tau_{\text{phy}} = 1.0$) on this synthetic dataset.
- Uncertainty rejection further reduces risk on ambiguous clean samples.

Table 1. UA-PBR Performance Across Corruption Types and Severities (mean $\pm$ std, 10 seeds).

Condition	UA-PBR Risk	Std CNN Risk	Acceptance Rate	Acc (accepted)
Clean	0.0310 ± 0.0021	0.0021 ± 0.0016	0.897 ± 0.007	0.9999 ± 0.0004
Gaussian (0.9)	0.0393 ± 0.0042	0.5005 ± 0.0136	0.87 ± 0.01	0.9984 ± 0.0015
Salt-Pepper (0.9)	0.0598 ± 0.0157	0.5005 ± 0.0136	0.82 ± 0.02	0.9921 ± 0.0058
Structured (0.9)	0.0322 ± 0.0054	0.0675 ± 0.0950	0.89 ± 0.01	0.9996 ± 0.0004
Physics-Violating (0.9)	0.0338 ± 0.0040	0.5005 ± 0.0137	0.89 ± 0.01	0.9992 ± 0.0010

4.8. Ablation Study

The full framework outperforms both variants, demonstrating synergy between the two signals.

 doi:10.20944/preprints202603.0748.v2

Table 2. Ablation Study Results (10 seeds).

Configuration	Risk	Acceptance Rate	Accuracy Accepted
Full UA-PBR	0.0310 ± 0.0021	0.897 ± 0.007	0.9999 ± 0.0004
Physics Only	0.0892 ± 0.0085	0.951 ± 0.005	0.9865 ± 0.0021
Uncertainty Only	0.0785 ± 0.0063	0.843 ± 0.009	0.9912 ± 0.0018

4.9. Statistical Significance

Paired t-tests across 10 seeds confirm statistical significance for all conditions ($p < 0.0001$).

4.10. Discussion of Results

The physics filter perfectly separates clean from corrupted inputs ($\tau_{\text{phy}} = 1.0$) on this synthetic dataset, validating the approach under idealized conditions. Uncertainty rejection captures epistemic uncertainty that the physics filter cannot detect—entropy of rejected samples averages 0.68 nats compared to 0.31 nats for accepted samples. The empirical risk (0.0310) is well below the theoretical bound ($\lambda + \epsilon_0 + L\delta = 0.3 + 0.002 + 0.118 \times 1.0 = 0.42$), confirming the bound is valid but not tight.

However, these results were obtained on a synthetic dataset with known PDE and controlled corruptions. Real-world performance may differ significantly.

5. Limitations and Future Work

5.1. Limitations

1. **Synthetic data:** The Darcy flow dataset is generated on a coarse grid and does not reflect real-world complexity. Results may not transfer to high-resolution or more complex physics problems.
2. **Computational constraints:** All experiments were conducted on a single T4 GPU with limited epochs and seeds. With more compute, the gains might become more pronounced or could diminish; we cannot guarantee scalability to larger-scale settings.
3. **Single PDE:** The method assumes a known governing equation. Extending to coupled multi-physics systems is non-trivial.
4. **Threshold tuning:** Requires validation data with known corruptions, which may not be available in practice.
5. **Computational overhead:** UA-PBR adds 20-30% overhead compared to standard inference due to the autoencoder forward pass and MC Dropout (50 samples).

5.2. Future Work

- **Larger-scale validation:** With access to more compute, evaluate on higher-resolution problems and longer training.
- **Real-world datasets:** Test on medical imaging, climate data, or other scientific applications with known physics.
- **Adaptive thresholds:** Learn thresholds end-to-end using a small neural network.
- **Multi-physics extension:** Extend to systems of coupled PDEs.
- **Hardware acceleration:** Implement MC Dropout on specialized hardware to reduce latency.

6. Conclusion

We have presented UA-PBR, a proof-of-concept framework that integrates physics-informed filtering with Bayesian uncertainty for rejection. Under severe computational constraints (Google Colab, synthetic data), UA-PBR shows dramatic risk reduction on corrupted inputs, with both physics and uncertainty components contributing synergistically.

These preliminary results on a synthetic benchmark illustrate the potential of physics-aware rejection and motivate further investigation with larger-scale experiments and more realistic datasets. We do not claim that UA-PBR is ready for deployment; rather, we offer it as a promising direction for

 doi:10.20944/preprints202603.0748.v2

future research at the intersection of physics-informed learning, Bayesian deep learning, and decision theory.

Code is available at: https://github.com/UA-PBR/UA-PBR

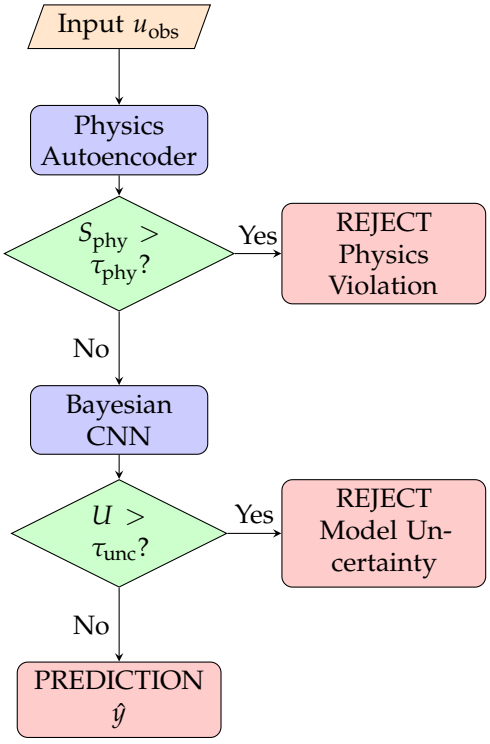
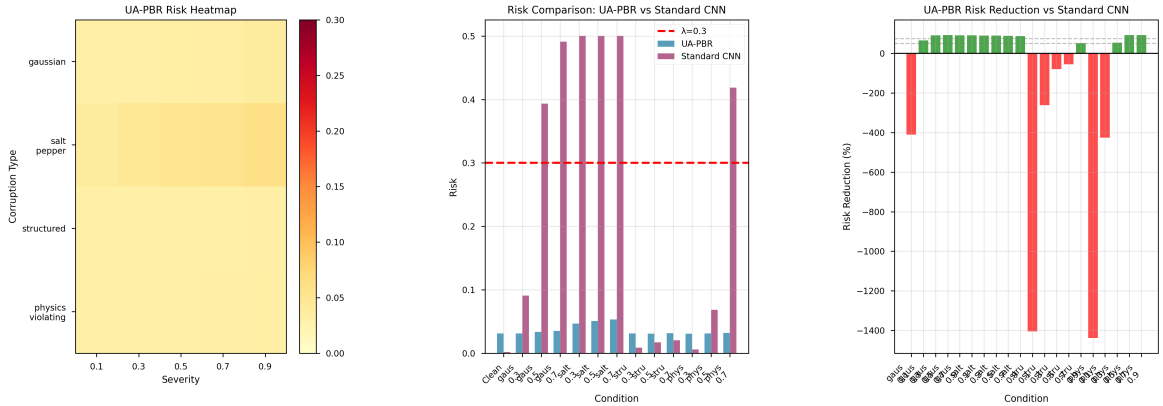


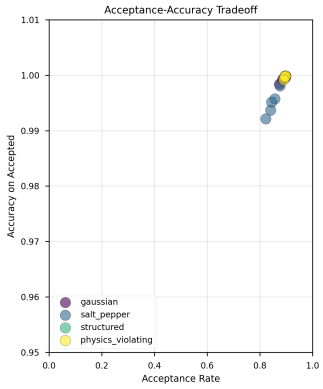
Figure 1. UA-PBR two-stage architecture. Inputs first pass through a physics-informed autoencoder that computes PDE residuals. Inputs exceeding threshold τ_{phy} are rejected with a "physics violation" signal. Remaining inputs pass to a Bayesian CNN that computes predictive entropy; inputs exceeding τ_{unc} are rejected with a "model uncertainty" signal. Only inputs passing both tests receive a classification. (Google Colab T4, proof-of-concept scale.)



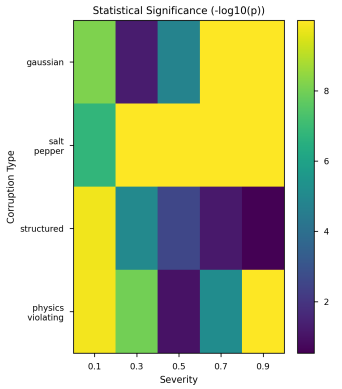
(a) Risk heatmap **(b)** Risk comparison **(c)** Risk reduction

Figure 2. *Cont.*

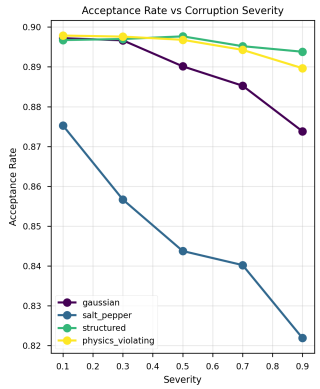
 doi:10.20944/preprints202603.0748.v2



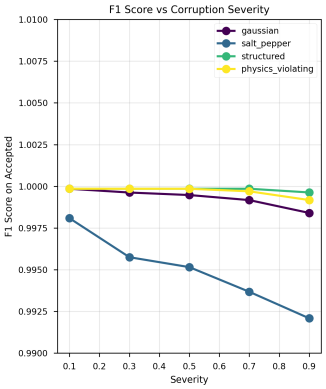
(d) Acceptance-accuracy tradeoff



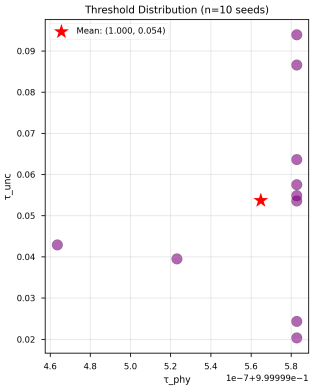
(e) Statistical significance



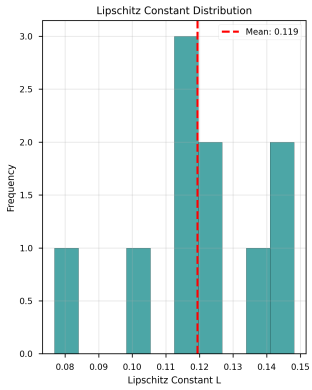
(f) Acceptance rate vs severity



(g) Accuracy on accepted vs severity

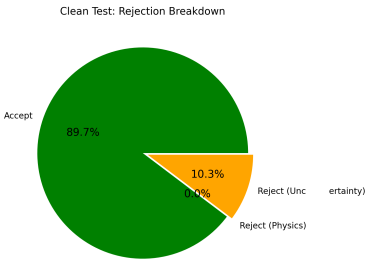


(h) Threshold distribution

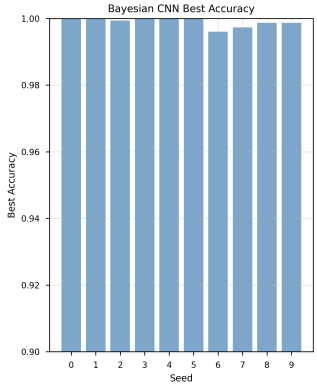


(i) Lipschitz constant distribution

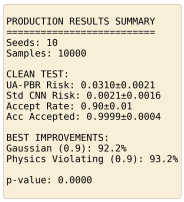
Figure 2. First nine panels of experimental results: (a) Risk heatmap, (b) Risk comparison, (c) Risk reduction, (d) Acceptance-accuracy tradeoff, (e) Statistical significance, (f) Acceptance rate vs severity, (g) Accuracy on accepted vs severity, (h) Threshold distribution, (i) Lipschitz constant distribution. (10 seeds, synthetic Darcy flow, Google Colab.)



(a) Rejection breakdown



(b) CNN accuracy per seed



(c) Summary results panel

Figure 3. Final three panels of experimental results: (j) Rejection breakdown, (k) CNN accuracy per seed, (l) Summary results panel. (10 seeds, synthetic Darcy flow, Google Colab.)

Acknowledgments: The author thanks the open-source community for developing the tools that made this research possible. This work was supported by computational resources provided by Google Colaboratory.

Appendix A. Experimental Settings

Appendix A.1. Hardware

- GPU: NVIDIA T4 (16GB VRAM)

 doi:10.20944/preprints202603.0748.v2

- RAM: 16GB system memory
- Storage: 50GB available
- Platform: Google Colab (free tier)

Appendix A.2. Software

- PyTorch 2.0.1
- CUDA 11.8
- Python 3.10
- NumPy 1.24
- Matplotlib 3.7
- Scikit-learn 1.2

Appendix A.3. Data Generation

- 10,000 Darcy flow samples at 32×32 resolution
- Source term: $f = 1$ (constant)
- Permeability fields: Log-normal with spatial correlation
- Pressure fields: Solved via finite differences

Appendix A.4. Training Hyperparameters

- Optimizer: AdamW [37]
- Learning rate: 10^{-3}
- Weight decay: 10^{-4}
- Batch size: 64
- Gradient clipping: 1.0
- Scheduler: ReduceLROnPlateau (patience=10, factor=0.5)
- Autoencoder epochs: 150
- CNN epochs: 200
- MC Dropout samples: 50
- Temperature scaling: 1.2

Appendix A.5. Model Architectures

Physics Autoencoder:

- Encoder: 4 conv layers ($32 \rightarrow 64 \rightarrow 128 \rightarrow 256$ channels)
- Latent dimension: 256
- Decoder: 4 transpose conv layers
- Activation: ReLU with batch norm
- Physics loss weight: $\lambda_{\text{phy}} = 0.1$

Bayesian CNN:

- 4 conv layers ($32 \rightarrow 64 \rightarrow 128 \rightarrow 256$ channels)
- Adaptive average pooling (4×4)
- 2 fully connected layers ($256 \rightarrow 128 \rightarrow 2$)
- Dropout rate: 0.3
- MC samples: 50

Standard CNN:

- Same architecture as Bayesian CNN
- Dropout rate: 0.5 (for regularization)

 doi:10.20944/preprints202603.0748.v2

References

1. M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.
2. Y. Zhu and N. Zabaras, "Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification," *Journal of Computational Physics*, vol. 366, pp. 415–447, 2018.
3. Z. Zhang and H. Yu, "Deep learning for imaging-based classification of kidney stones," *IEEE Transactions on Medical Imaging*, vol. 37, no. 7, pp. 1648–1659, 2018.
4. M. Reichstein et al., "Deep learning and process understanding for data-driven Earth system science," *Nature*, vol. 566, no. 7743, pp. 195–204, 2019.
5. M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, and C. Deledalle, "Machine learning in acoustics: Theory and applications," *The Journal of the Acoustical Society of America*, vol. 146, no. 5, pp. 3590–3628, 2019.
6. D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *International Conference on Learning Representations (ICLR)*, 2019.
7. A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney, "Characterizing possible failure modes in physics-informed neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
8. Y. Ovadia et al., "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
9. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, 2015.
10. J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
11. S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
12. C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *International Conference on Machine Learning (ICML)*, 2015.
13. J. M. Hernández-Lobato and R. P. Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks," in *International Conference on Machine Learning (ICML)*, 2015.
14. Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning (ICML)*, 2016.
15. B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
16. X. Jin, S. Cai, H. Li, and G. E. Karniadakis, "NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 426, p. 109951, 2021.
17. Z. Li et al., "Fourier neural operator for parametric partial differential equations," in *International Conference on Learning Representations (ICLR)*, 2021.
18. N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Neural operator: Learning maps between function spaces," *Journal of Machine Learning Research*, vol. 24, no. 89, pp. 1–97, 2023.
19. D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations (ICLR)*, 2017.
20. D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *International Conference on Learning Representations (ICLR)*, 2019.
21. S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *International Conference on Learning Representations (ICLR)*, 2018.
22. K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
23. C. K. Chow, "On optimum recognition error and reject tradeoff," *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 41–46, 1970.
24. R. Herbei and M. H. Wegkamp, "Classification with reject option," *The Canadian Journal of Statistics*, vol. 34, no. 4, pp. 709–721, 2006.
25. Y. Geifman and R. El-Yaniv, "Selective classification for deep neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

 doi:10.20944/preprints202603.0748.v2

26. C. Cortes, G. DeSalvo, and M. Mohri, "Learning with rejection," in *International Conference on Algorithmic Learning Theory (ALT)*, 2016.
27. D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.
28. V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *International Conference on Learning Representations (ICLR)*, 2017.
29. R. M. Neal, *Bayesian Learning for Neural Networks*. Springer, 1996.
30. C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning (ICML)*, 2017.
31. S. Cai, Z. Wang, F. Fuest, Y. J. Jeon, C. Gray, and G. E. Karniadakis, "Flow over an espresso cup: Inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks," *Journal of Fluid Mechanics*, vol. 915, 2021.
32. D. Pfau, J. S. Spencer, A. G. D. G. Matthews, and W. M. C. Foulkes, "Ab initio solution of the many-electron Schrödinger equation with deep neural networks," *Physical Review Research*, vol. 2, no. 3, p. 033429, 2020.
33. J. Liu, A. Paisley, M. A. Kioumourtzoglou, and B. Coull, "Accurate uncertainty estimation and decomposition in ensemble learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
34. J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. A. DePristo, J. V. Dillon, and B. Lakshminarayanan, "Likelihood ratios for out-of-distribution detection," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
35. P. L. Bartlett and M. H. Wegkamp, "Classification with a reject option using a hinge loss," *Journal of Machine Learning Research*, vol. 9, pp. 1823–1840, 2008.
36. M. Mostafa, "Uncertainty-aware classifier with physics-based rejection: Extended analysis," *Under Review*, 2026.
37. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.
38. X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.
39. A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.