Equivariant Architectures vs. Transformers: Or, What Gradient Overlaps Reveal About Memorization and Generalization

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

We introduce a diagnostic for symmetry learning in PDE surrogates: the similarity between parameter gradients computed on symmetry-related states. On compressible Euler flows, our diagnostic reveals that a UNet exhibits partial but unstable gradient coherence across square group actions and translations, whereas a ViT reaches lower prediction error yet shows largely orthogonal updates across orbits. This exposes an optimization-symmetry trade-off: stronger inductive biases promote data efficiency but can couple updates rigidly; flexible architectures optimize easily but ignore physical structure. Our diagnostic offers a reproducible test for whether training dynamics propagate information across symmetry orbits.

1 Introduction

Deep learning emulators for partial differential equation (PDE) solvers routinely achieve impressive in-distribution accuracy [1–6], yet they often fail to respect the fundamental symmetries of the governing equations [7–9]. This limitation undermines their ability to extrapolate or generalize, raising the question: are such models truly learning physics, or merely fitting correlations present in the training data? Addressing this gap requires probing not just the outputs, but also the learning dynamics [10, 11].

Symmetries of the Euler equations, namely translations, rotations, reflections, scalings, and Galilean boosts, organize the solution space into orbits whose members are physically equivalent [12]. A model that has internalized the solution operator should propagate information seamlessly across these orbits: gradients of the loss with respect to parameters, evaluated on symmetry-related inputs, should align. Measuring this *gradient overlap* offers a diagnostic beyond standard forward-pass equivariance checks, exposing the degree to which training updates are physically consistent.

If gradient overlap decays rapidly across group actions, the model is memorizing localized patterns rather than learning physical processes [13–15]. Conversely, persistent gradient coherence signals that the network has learned to couple symmetry-related states, consistent with the behavior of a true solution operator. Our *symmetry-aware gradient diagnostic* therefore quantifies a model's ability to generalize across orbits, providing a principled tool to assess how architectural choices, loss design, and inductive biases promote, or hinder, robust generalization.

2 Method

We compare a UNet (13M parameters, 4 down-sampling blocks, 24 embedding channels) and a Vision Transformer (ViT; 5M parameters, 6 layers, 256 channels) trained as emulators for two-dimensional compressible Euler flows from PDEGym [2]. For data, we selected three classes of Riemann-type initial conditions (CE-RP, CE-RPUI, CE-CRP), each with $5{,}000$ trajectories of 16 time steps. Each state snapshot is a 128×128 grid of mass density, Cartesian momentum densities, and energy density. Models were trained autoregressively to emulate the Euler evolution operator.

Optimization used Adam with learning rate 5×10^{-4} and weight decay $\lambda = 10^{-6}$ on mini-batches of $N = 64$ transitions. The cost function was a scaled mean-squared error (SMSE), that normalizes errors by channel RMS to balance large- and small-amplitude features, ensuring shocks and wavefronts are captured while retaining sensitivity to quiescent flows, in addition to rendering dimensionless the gradient overlap matrix of interest. Both models were trained in distributed mode on two 40GB A100 GPUs using Lux.jl [16, 17] and Zygote.jl [18], with three seeds controlling initialization and dataset splits. Results are reported with quantile range bars to capture variability across seeds and mini-batches. Despite having fewer parameters, the ViT consistently outperforms the UNet after 90 epochs. Further training details are reported in a previous manuscript that we omit here.

To evaluate our models, we compute the curvature–adjusted gradient overlap between a test input x and its g-transformed counterpart $g \cdot x$ as the normalized inner product

$$\Omega_g(x) \;=\; \frac{\langle \nabla_\theta \mathcal{L}(g \cdot x),\, \nabla_\theta \mathcal{L}(x) \rangle_\chi}{\|\nabla_\theta \mathcal{L}\|_\chi}, \qquad \langle u, v \rangle_\chi = u^\top \chi v, \tag{1}$$

where the metric is defined as $\chi = (\lambda + \eta)^{-1}$, with η the neural tangent kernel [19], and $\|\nabla_\theta \mathcal{L}\|_\chi$ denotes the Frobenius norm of the mini-batch gradient overlap matrix between group-transformed and untransformed inputs. In practice, χ is applied via a Krylov.jl [20] matrix-free solver, yielding a curvature-sensitive measure of gradient alignment that reflects the local geometry of the loss surface [21, 22]

A primary limitation of our analysis is computational: solving for a mini-batch of gradient overlaps in the χ-metric can require more than a day, making it infeasible to compute the full normalizing denominator needed to obtain the cosine angle between gradients. Also, we were limited to a relative error tolerance of 5×10^{-3} and 5×10^{-2} for our UNet and ViT gradient overlaps, respectively. An additional limitation is that our study considers only UNet and ViT baselines and does not explore the full symmetry group of the Euler equations, in particular omitting geometric convolutional and Lie-symmetry-aware architectures.

3 Results and Discussion

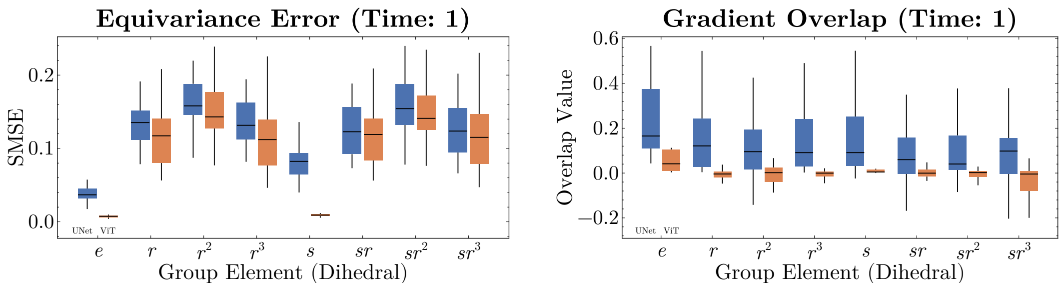
Our evaluation considers both equivariance error (forward-pass consistency under symmetry) and gradient overlap (alignment of parameter updates between symmetry-related inputs). Gradient overlaps reveal whether learning dynamics propagate information coherently across symmetry-related states, exposing whether a model is genuinely learning physics or merely fitting data. We find that forward error metrics alone are insufficient to characterize the extent to which our models have internalized symmetry: the UNet enforces coupling inconsistently, while the ViT converges to a symmetry-breaking solution despite superior predictive accuracy. Convergence to basins that respect the symmetries of the underlying problem is both essential for generalization and a persistent challenge for current architectures.

The results of our analysis underscores the trade-off between inductive bias and optimization ease. Our ViT is easy to train but symmetry-blind, whereas our UNet partially encodes symmetry at the cost of relatively frustrated training dynamics [23–25]. This contrast illustrates why hard equivariant constraints can hinder convergence, while unconstrained models converge rapidly but fail to generalize beyond the training distribution, emphasizing the utility of approximately constrainted modeling [26].

3.1 Dihedral Group

We analyze model behavior under the action of the dihedral group D_4 at the first step in the autoregressive evolution of compressible Euler flow. At this time, we expect a trained model to perform equivalently on D_4-rotated inputs because the data generating process for Riemann initial conditions is itself D_4 symmetric. Failure to capture the governing physics at the initial step is especially detrimental, as early inconsistencies propagate and amplify across subsequent rollout steps.

Figure 1a shows the SMSE of the UNet and ViT when their outputs are transformed along the D_4 orbit. Despite the ViT's superior performance on untransformed test data, both models exhibit comparable



(a) Tukey plot of the equivariance error under D_4 symmetry operations.

(b) Tukey plot of the gradient overlap between examples related by D_4 transformations.

Figure 1: Diagnostics under D_4 symmetry, which is generated by counter-clockwise $\pi/2$ rotations r and reflections s about the vertical axis; e is the identity transformation. The left (blue) candle represents our UNet and the right (orange) candle represents our ViT.

equivariance errors once inputs are rotated or reflected. The ViT does not show a substantial rise in error under reflections, but overall neither model demonstrates consistent dihedral symmetry. This failure reflects their lack of inductive biases; awareness of rotation and reflection must be inferred from data. Both models fail this test to a similar degree, even though their raw test accuracy differs markedly.

Figure 1b probes the learning dynamics by examining gradient overlap across dihedral transformations. The ViT shows consistently *minimal* overlap: gradients computed on x and $g \cdot x$, with $g \in D_4$, are nearly orthogonal. This suggests that ViT training dynamics treat symmetry-related states as unrelated problems. In other words, while the ViT fits the untransformed distribution well, its parameter updates fail to propagate information across the orbit, explaining its poor equivariance generalization.

By contrast, the UNet shows *larger but highly variable* overlap values. This variance indicates that gradients sometimes align across rotations and reflections, but in an inconsistent manner. Such instability reflects the rigidity of convolutional features. This unstable coupling slows convergence and frustrates optimization

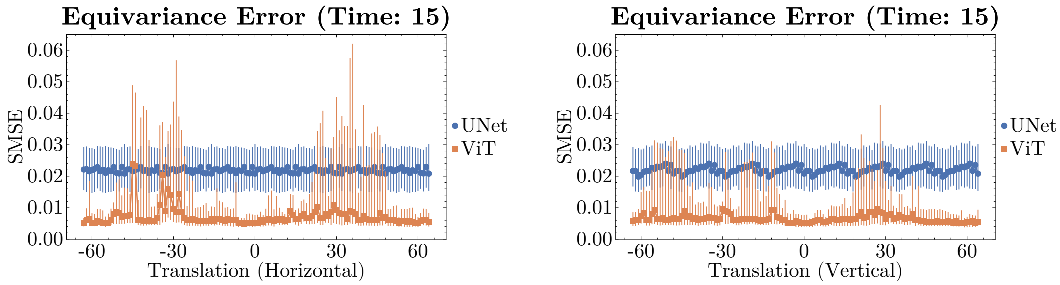
Neither architecture internalizes dihedral symmetry in its learning dynamics. For the ViT, positional encodings create a fundamental mismatch with rotation; for the UNet, convolutional filters confer only translation symmetry, leaving rotations to be learned opportunistically from local data.

3.2 Translation Group

For the translation group, we evaluate purely horizontal and vertical translations at the latest time that is not on the boundary of the time domain seen during training. At this stage, it is most reasonable to expect the model to have learned translation equivariance: prolonged mixing tends to render the dataset statistically homogeneous in feature space. Because the governing PDE applies uniformly across space, learned dynamics should be equivariant under translations. In any given flow snapshot, wave interactions occupy only a few localized regions, yet they may arise at arbitrary spatial locations. A model that learns translational equivariance will therefore treat such interactions consistently wherever they occur, ensuring that these rare but critical events are captured and enabling accurate long-time extrapolation.

Figure 2a and Figure 2b report equivariance error under horizontal and vertical translations. Across both directions, the ViT typically attains lower SMSE than the UNet, apart from exceptional inputs that produce isolated spikes. These spikes indicate sharp variations in the local loss landscape near the converged ViT parameters, indicating failure to learn translation equivariance.

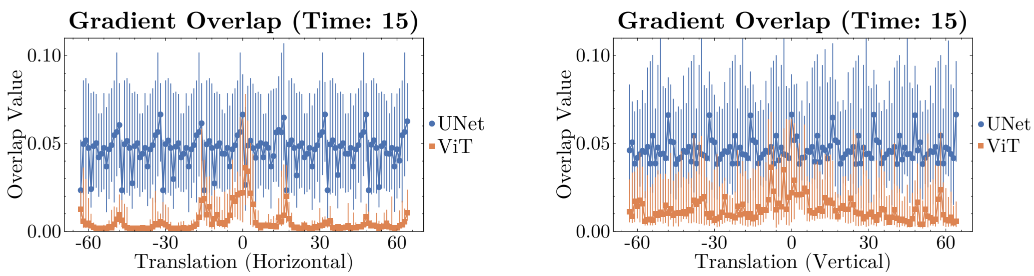
The gradient overlaps in Figure 3a and Figure 3b further distinguish the two architectures. The UNet exhibits consistently larger, though variable, overlap values, indicating partial but variable coupling of parameter updates across translated states. The ViT, by contrast, shows gradients that are largely orthogonal between shifted inputs, with a systematic asymmetry: vertical translations exhibit a greater typical overlap than horizontal translations, likely reflecting biases in patch embeddings and positional encodings. Notably, some equivariance error peaks coincide with gradient overlap



(a) Equivariance error under horizontal translations.

(b) Equivariance error under vertical translations.

Figure 2: Forward-pass equivariance error under horizontal (a) and vertical (b) translations. Markers represent median values and the whiskers indicate neighboring quantiles.



(a) Gradient overlap under horizontal translations.

(b) Gradient overlap under vertical translations.

Figure 3: Curvature-adjusted gradient overlaps between symmetry-related inputs under horizontal (a) and vertical (b) translations. Markers represent median values and the whiskers indicate neighboring quantiles.

resonances, consistent with the fact that the overlap matrix encodes the local geometry of the loss surface. Other error peaks, however, occur without gradient coherence, revealing cases where the model simply fails to propagate information provided by symmetry.

4 Conclusion

Our analysis complements recent [27] lessons on the role of inductive biases in deep learning. Hard constraints such as equivariance offer data efficiency and principled generalization, yet our gradient diagnostics reveal that they often impose optimization bottlenecks: parameter updates become rigidly coupled across symmetry orbits, slowing convergence and frustrating training. By contrast, unconstrained architectures such as transformers converge rapidly by freely specializing their gradients, even if this means disregarding underlying symmetries. The result is high predictive accuracy but limited physical consistency; powerful interpolators rather than genuine physics-aware models.

Furthermore, our gradient-overlap analysis reframes this trade-off by exposing whether a model propagates learning coherently across symmetry-related states or merely memorizes surface correlations. From this perspective, apparent accuracy without gradient coherence signals fragile generalization. Looking forward, these diagnostics motivate the development of approximate or relaxed symmetry methods that preserve enough structure to guide generalization while retaining the flexibility needed for efficient optimization, potentially reconciling the scalability of transformers with the principled construction of equivariant models.

Beyond technical contributions, our framework aims to strengthen trust in scientific machine learning by clarifying when models truly learn physics, while also underscoring the risks of misuse if surrogate predictions are deployed without such diagnostic safeguards.

References

[1] J. Brandstetter, D. Worrall, and M. Welling, Message passing neural pde solvers (2023), arXiv:2202.03376 [cs.LG] .

[2] M. Herde, B. Raonić, T. Rohner, R. Käppeli, R. Molinaro, E. de Bézenac, and S. Mishra, Poseidon: Efficient foundation models for pdes (2024), arXiv:2405.19101 [cs.LG] .

[3] M. Takamoto, T. Praditia, R. Leiteritz, D. MacKinlay, F. Alesiani, D. Pflüger, and M. Niepert, Pdebench: An extensive benchmark for scientific machine learning (2024), arXiv:2210.07182 [cs.LG] .

[4] P. Lippe, B. S. Veeling, P. Perdikaris, R. E. Turner, and J. Brandstetter, Pde-refiner: Achieving accurate long rollouts with neural pde solvers (2023), arXiv:2308.05732 [cs.LG] .

[5] J. K. Gupta and J. Brandstetter, Towards multi-spatiotemporal-scale generalized pde modeling (2022), arXiv:2209.15616 [cs.LG] .

[6] R. Ohana, M. McCabe, L. Meyer, R. Morel, F. J. Agocs, M. Beneitez, M. Berger, B. Burkhart, K. Burns, S. B. Dalziel, D. B. Fielding, D. Fortunato, J. A. Goldberg, K. Hirashima, Y.-F. Jiang, R. R. Kerswell, S. Maddu, J. Miller, P. Mukhopadhyay, S. S. Nixon, J. Shen, R. Watteaux, B. R.-S. Blancard, F. Rozet, L. H. Parker, M. Cranmer, and S. Ho, The well: a large-scale collection of diverse physics simulations for machine learning (2025), arXiv:2412.00568 [cs.LG] .

[7] T. Akhound-Sadegh, L. Perreault-Levasseur, J. Brandstetter, M. Welling, and S. Ravanbakhsh, Lie point symmetry and physics informed networks (2023), arXiv:2311.04293 [cs.LG] .

[8] W. G. Gregory, D. W. Hogg, B. Blum-Smith, M. T. Arias, K. W. K. Wong, and S. Villar, Equivariant geometric convolutions for emulation of dynamical systems (2024), arXiv:2305.12585 [cs.LG] .

[9] N. Gruver, M. Finzi, M. Goldblum, and A. G. Wilson, The lie derivative for measuring learned equivariance (2024), arXiv:2210.02984 [cs.LG] .

[10] S. Fort, P. K. Nowak, S. Jastrzebski, and S. Narayanan, Stiffness: A new perspective on generalization in neural networks (2020), arXiv:1901.09491 [cs.LG] .

[11] B. Zhao, R. M. Gower, R. Walters, and R. Yu, Improving convergence and generalization using parameter symmetries (2024), arXiv:2305.13404 [cs.LG] .

[12] J. Brandstetter, M. Welling, and D. E. Worrall, in *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 162, edited by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato (PMLR, 2022) pp. 2241–2256.

[13] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien, A closer look at memorization in deep networks (2017), arXiv:1706.05394 [stat.ML] .

[14] H. He and W. J. Su, The local elasticity of neural networks (2020), arXiv:1910.06943 [cs.LG] .

[15] S. Chatterjee, Coherent gradients: An approach to understanding generalization in gradient descent-based optimization (2020), arXiv:2002.10657 [cs.LG] .

[16] A. Pal, Lux: Explicit Parameterization of Deep Neural Networks in Julia (2023).

[17] A. Pal, On Efficient Training & Inference of Neural Differential Equations (2023).

[18] M. Innes, CoRR **abs/1810.07951** (2018), arXiv:1810.07951 .

[19] A. Jacot, F. Gabriel, and C. Hongler, Neural tangent kernel: Convergence and generalization in neural networks (2020), arXiv:1806.07572 [cs.LG] .

[20] A. Montoison and D. Orban, Journal of Open Source Software **8**, 5187 (2023).

[21] S. Fort and S. Ganguli, Emergent properties of the local geometry of neural loss landscapes (2019), arXiv:1910.05929 [cs.LG] .

[22] P. Zielinski, S. Krishnan, and S. Chatterjee, Weak and strong gradient directions: Explaining memorization, generalization, and hardness of examples at scale (2020), arXiv:2003.07422 [cs.LG] .

[23] R. Zhang, in *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 97, edited by K. Chaudhuri and R. Salakhutdinov (PMLR, 2019) pp. 7324–7334.

[24] A. Azulay and Y. Weiss, Journal of Machine Learning Research **20**, 1 (2019).

[25] O. S. Kayhan and J. C. van Gemert, On translation invariance in cnns: Convolutional layers can exploit absolute spatial location (2020), arXiv:2003.07064 [cs.CV] .

[26] R. Wang, R. Walters, and R. Yu, Approximately equivariant networks for imperfectly symmetric dynamics (2022), arXiv:2201.11969 [cs.LG] .

[27] D. Canez, N. Midavaine, T. Stessen, J. Fan, S. Arias, and A. Garcia, Effect of equivariance on training dynamics (2024).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our main contributions: We introduce a gradient-overlap diagnostic for symmetry, contrasting a UNet and a transformer, and demonstrating how this lens distinguishes memorization from generalization. These claims are supported by both our theoretical framing and empirical results.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss limitations, including the high computational cost of solving for mini-batch gradient overlaps in the χ–metric, which can take more than a day and prevents computing full cosine-normalized overlaps, as well as the restricted scope of models and symmetries considered (limited to UNet and ViT, without exploring geometric convolutional or Lie-symmetry–aware architectures).

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This work does not introduce new theoretical results. We rely only on established mathematical definitions of symmetry groups and coherence, without advancing new theorems or proofs.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes. We describe the dataset (PDEGym with three classes of Riemann initial conditions, $5{,}000$ trajectories each, 128×128 grids), the architectures (UNet and ViT with parameter counts, layer structure, and channels), and the full training setup (Adam optimizer, 5×10^{-4} learning rate, 10^{-6} weight decay, batch size 64, SMSE loss). We also report the number of seeds, precisely which will be made available on code release, mini-batches, hardware used, and the procedure for computing gradient overlaps via Krylov solvers. Together these details allow the main experimental results and conclusions to be reproduced even in the absence of released code.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We are currently working through the code release process required by our institution. While the materials cannot yet be made openly available, we intend to provide them pending institutional approval, and all experimental details are documented to ensure reproducibility in the interim.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

Justification: Yes. We fully specify the dataset (PDEGym with three classes of Riemann initial conditions, 5,000 trajectories each, 128×128 grids), the model architectures (UNet with 13M parameters, ViT with 5M parameters, including layer structure), and the training setup (Adam optimizer with learning rate 5×10^{-4}, weight decay 10^{-6}, batch size 64, scaled MSE loss). We also describe the number of epochs, seeds, and mini-batches used, as well as the hardware environment and Krylov solver tolerances for computing gradient overlaps. These details provide sufficient information to understand and reproduce the reported results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All experiments were repeated across multiple models, random seeds, and mini-batch samplings. To convey statistical significance, we report quantile range bars on all plots, exhibiting variability and ensuring that our claims are supported by consistent trends rather than single instances.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes. All models were trained in distributed mode on two NVIDIA A100 GPUs (40GB memory each). Each run of 90 training epochs required approximately six days, and gradient overlap computations in the χ–metric for a single mini-batch could also take more than a day when solved with the Krylov matrix-free method. We report that both UNet and ViT models were trained with three seeds, yielding a total of nine full training runs per architecture. These details provide sufficient information on compute type, memory, and time of execution to reproduce the experiments.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research involves only synthetic PDE datasets and standard deep learning models, with no human subjects, sensitive data, or dual-use concerns. All experiments and analyses were conducted in accordance with the NeurIPS Code of Ethics, and we have preserved anonymity throughout.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes. We discuss broader impacts in terms of both potential benefits and risks. On the positive side, improved diagnostic tools for distinguishing memorization from generalization in PDE surrogates may accelerate progress in scientific machine learning and help ensure physically reliable models in high-stakes domains such as climate or engineering simulations. On the negative side, any surrogate model trained without adequate physical guarantees could lead to misleading conclusions or unsafe deployment if trusted uncritically. We mitigate this by framing our contribution as a diagnostic rather than a deployable system and by highlighting the need for caution in scientific and engineering use cases.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks. Both the model architectures and the dataset used are in the open literature. The dataset involves academically oriented physics.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cite all literature sources describing the models and datasets used; all model code was written by us directly from these descriptions rather than reusing existing implementations.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The paper does not release new assets. We used model architectures and data that are openly available and described in the literature.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.