
Does Text Steer Neural PDE Surrogates? A Controlled Diagnostic with OperatorCLIP

Aadi Dash¹, Lennon J. Shikhman^{1,2,*}, Michael Galarnyk³

¹College of Computing, Georgia Institute of Technology

²Department of Mathematics and Systems Engineering, Florida Institute of Technology

³Wallace H. Coulter Department of Biomedical Engineering, Georgia Institute of Technology

*Corresponding author

{aadidash, lshikhman3, mgalarnyk3}@gatech.edu

Abstract

Lower error from a text-conditioned neural surrogate does not, by itself, show that the model uses the meaning of the text. We examine this attribution problem with *OperatorCLIP*, comparing an unconditioned FNO, a constant-sentence FiLM control, and a fixed task description trained with contrastive alignment. Three-seed experiments cover Darcy2D, ShallowWater2D, and three-dimensional compressible Navier–Stokes (CNS3D). Constant conditioning has lower mean test error on both 2D tasks. Relative to this control, task text plus alignment has a similar mean on ShallowWater2D and CNS3D and a higher mean on Darcy2D; these descriptive comparisons have substantial seed uncertainty. The latter comparison changes both prompt content and loss, so it isolates neither effect. The text encoder is trained from scratch, and each conditioned model sees only one description during training. In this regime, pairwise InfoNCE cannot identify matched pairs and has minimum $\log B$. Prompt interventions show no reliable semantic ordering. This methodological caution demonstrates why pathway controls are needed; it neither establishes semantic competence of the encoder nor tests the effectiveness of text under varying physical context.

1 Introduction

Neural PDE surrogates approximate solution operators and time-advance maps (12). Observed fields may underdetermine behavior when coefficients (14), boundary conditions (30), forcing or operator examples (33), geometry (15), or discretization (5) vary.

PDEBench exposes such context through varying physical and simulation settings (32; 29). Context may also appear in metadata, solver configurations, or scientific language, motivating models conditioned on text, equations, and operator examples (19; 16; 33).

Adapters, affine modulation, extra parameters, and auxiliary losses can alter optimization even when a prompt is constant. A task-specific model can therefore improve without using linguistic content. Establishing semantic steering requires matched pathway controls and prompt interventions, beyond comparison with an unconditioned model.

OperatorCLIP combines CLIP-style alignment (25) with FiLM under the same FNO backbone specification. Constant-prompt controls assess pathway and capacity changes together. Correct, missing, shuffled, corrupted, paraphrased, and numeric-metadata prompts probe sensitivity after training. Since each model trains on one description, these interventions cannot establish that it learned how changing physical context should change predictions.

Contributions. We document attribution limits of fixed-prompt comparisons, prove that identical deterministic text embeddings make pairwise alignment unidentifiable, and report all 27 runs across two 2D tasks and one 3D task. The controls expose confounds rather than fully separating their causal effects.

2 Related Work

Operators and context. Operator architectures include DeepONet (20), spectral FNO (14), geometry-aware models (15; 2), CNO and U-Net (27; 28), and attention-based OFormer and GNOT (13; 7). Physics-informed networks impose equations through the loss (26); meta-learning shares information across parametric families (4; 3). All our comparisons use FNO.

Benchmarks and failure modes. PDEBench spans eight PDE families in one to three dimensions (32); PDEArena varies parameters and scales (6); and CFDBench varies boundaries and geometry (21). The Well, BubbleML, and APEBench add further systems and rollout settings (23; 9; 11). Audits show dependence on boundary distributions (30), failure under parameter or boundary shifts (29), discretization mismatch error (5), and inconsistency under temporal composition (31). These settings motivate testing whether text supplies information beyond the numerical input.

Multimodal scientific learning. PROSE and PROSE-FD jointly process equations and fields (17; 16); language-conditioned surrogates use cross-attention (19); and in-context operators accept data-pair prompts (33), with multimodal extensions incorporating equations and natural-language descriptions (34). Unisolver and PDEformer encode PDE components or graphs (36; 35), while DPOT, Poseidon, and MPP pretrain across systems without text (8; 10; 22). OperatorCLIP combines contrastive alignment (25) with FiLM (24); PICL derives similarity from equation coefficients (18); and OmniArch aligns captions with spectral changes (1). Our diagnostic examines attribution limits of fixed-text conditioning.

3 Controlled Diagnostic Framework

For input field or history x_i and target y_i , an unconditioned surrogate predicts $\mathcal{G}_\theta(x_i)$. OperatorCLIP encodes operator context m_i , such as coefficients, boundaries, or geometry, in a structured prompt p_i and predicts

$$\hat{y}_i = \mathcal{G}_\theta(x_i; h_i^{\text{text}}), \quad (1)$$

where h_i^{text} is the text encoder’s conditioning vector. The projection heads below are used only for the auxiliary alignment loss.

Shared text-data embedding. A text encoder maps p_i to $h_i^{\text{text}} = E_{\text{text}}(p_i)$, while a data encoder maps numerical evidence to $h_i^{\text{data}} = E_{\text{data}}(x_i, y_i)$. For $q \in \{\text{text}, \text{data}\}$, projection heads produce normalized embeddings

$$z_i^q = \frac{P_q h_i^q}{\|P_q h_i^q\|_2}. \quad (2)$$

For a minibatch of B matched text-data pairs, with $S_{ij} = \langle z_i^{\text{text}}, z_j^{\text{data}} \rangle / \tau$, we use the symmetric contrastive loss

$$\mathcal{L}_{\text{align}} = \frac{1}{2B} \sum_{i=1}^B \left[-\log \frac{\exp(S_{ii})}{\sum_j \exp(S_{ij})} - \log \frac{\exp(S_{ii})}{\sum_j \exp(S_{ji})} \right]. \quad (3)$$

The training objective is

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}}, \quad (4)$$

where $\mathcal{L}_{\text{pred}}$ is the supervised field or trajectory loss.

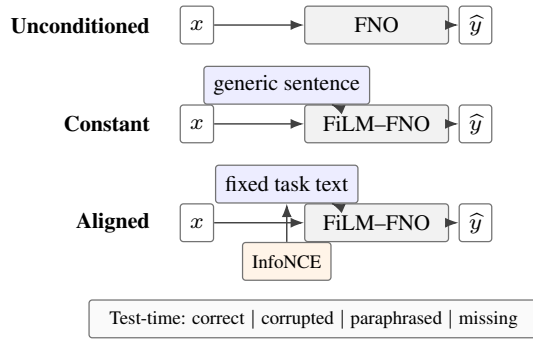


Figure 1: Diagnostic comparisons. Unconditioned-to-constant changes pathway and capacity together; constant-to-aligned changes both text and loss. Interventions probe prompt sensitivity.

Proposition 1 (fixed-prompt non-identifiability). *If every example produces the same deterministic text embedding, then $\mathcal{L}_{\text{align}} \geq \log B$ and the loss contains no information identifying matched text–data pairs.* In the data-to-text direction, all B candidates are identical and the loss is exactly $\log B$. In the text-to-data direction, every row shares one score distribution while each data item serves once as the positive; the batch average is minimized by a uniform distribution, again at $\log B$. Away from equal scores, alignment can still produce gradients; the result does not imply a constant loss throughout training or fix the learned embedding. Shuffling identical prompts cannot change the model inputs.

Text encoder and conditioning. Both conditioned configurations use a text encoder trained jointly with the FNO from random initialization, without language pretraining. Its output supplies the FiLM conditioning vector. Appendix A specifies the architecture, tokenizer, and initialization.

4 Results

Controlled setup. Each FNO is trained separately on one task. Darcy2D maps a 128×128 coefficient field to its solution using 8,000/1,000/1,000 train/validation/test examples; ShallowWater2D maps the first 128×128 state to the next using an 800/100/100 split. CNS3D maps five-channel 128^3 states one step forward using 80/10/10 examples. The 2D FNO has width 64 and 12 modes per axis and trains for 50 epochs with batch size four; the 3D FNO has width 48 and eight modes per axis and trains for 30 epochs with batch size two. All use four layers, channel normalization, AdamW (5×10^{-4}), optimization seeds 7, 17, and 27, and the same fixed data split. We report final-epoch test relative L^2 as mean $\pm$ sample standard deviation, with every seed in Appendix A.

The *unconditioned* FNO removes the text encoder and FiLM modules. The *constant* control retains them but always receives “Operator context unavailable. Predict the output from the numerical input,” with no alignment loss. The *aligned* condition uses a fixed task description listing dataset, dimension, channels, task type, input/target steps, filename, and data key, with $\lambda_{\text{align}} = 0.1$ and temperature 0.07. Its numerical-evidence encoder uses per-channel input/target statistics. Targets enter only the auxiliary training loss, never inference. Conditioning adds approximately 2.8–2.9% surrogate parameters in 2D and 0.6% in 3D. Constant-to-aligned changes the prompt, vocabulary size, and objective together; it is not an isolated alignment ablation.

For semantic steering, prompt dependence should be accompanied by an ordering consistent with meaning: correct and meaning-preserving descriptions should outperform missing or factually corrupted ones. Here that criterion is a diagnostic, not a capability learned under varying training prompts.

Table 1: Test relative L^2 (mean $\pm$ sample s.d., three seeds). Positive $\Delta_{U \rightarrow C}$ and $\Delta_{C \rightarrow A}$ denote lower mean error in constant versus unconditioned and aligned versus constant. Percentages are descriptive ratios of means, not isolated causal effects or significance claims.

Task	Unconditioned	Constant	Fixed prompt + align.	$\Delta_{U \rightarrow C}$	$\Delta_{C \rightarrow A}$
Darcy2D	.03714 $\pm$.00744	.03009 $\pm$.00064	.03903 $\pm$.00941	+19.0%	−29.7%
ShallowWater2D	.001694 $\pm$.000242	.001394 $\pm$.000272	.001373 $\pm$.000099	+17.7%	+1.5%
CNS3D	.40446 $\pm$.00485	.40675 $\pm$.00789	.40458 $\pm$.01458	−0.6%	+0.5%

The constant mean is lower than the unconditioned mean on both 2D tasks (Table 1). On ShallowWater2D, the aligned mean is only 1.5% lower than the constant mean; the 18.9% difference from unconditioned would conflate this small difference with the pathway/capacity change. Darcy2D favors constant in the observed means, while CNS3D means differ by less than 0.6%. Three seeds on one split do not establish stable rankings or variance effects. Alignment losses approach $\log B$ (1.386 for $B = 4$ in 2D and 0.693 for $B = 2$ in 3D), as permitted by Proposition 1.

The interventions do not establish semantic ordering (Figure 2). Shuffling is exactly neutral because prompts are identical within each task. Darcy2D is nearly invariant; CNS3D changes are smaller in magnitude than ShallowWater2D’s. On ShallowWater2D, corrupted facts yield a paired error change of $+2.3\% \pm 14.2\%$, compared with $+50.6\% \pm 62.3\%$ for paraphrases. These variable responses show embedding sensitivity without establishing a semantic explanation. Because these alternative

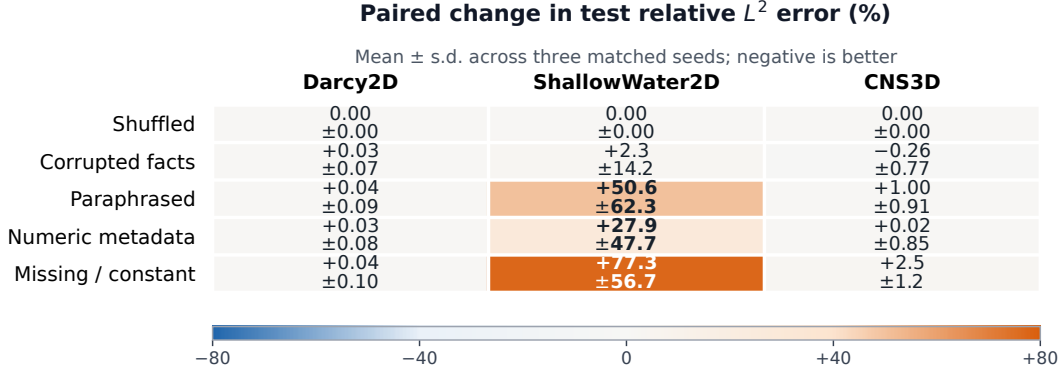


Figure 2: Prompt sensitivity of the aligned models. Cells show the paired change in test relative L^2 error from the correct description (mean $\pm$ sample standard deviation over the same three seeds); positive values indicate higher error. Missing and constant modes use the same sentence.

descriptions are unseen in training, their effects also depend on tokenization and representation quality.

5 Discussion

The constant control shows that lower observed mean error can accompany a fixed generic prompt. Its embedding drives a learned, dataset-wide affine modulation, but the experiment does not separate FiLM, added capacity, and optimization effects. Capacity-matched FNOs and learned task identifiers remain necessary controls. Differences in dataset size, dimension, and training budget also prevent attributing cross-task patterns to PDE family alone.

The encoder has no pretrained semantic competence, and one-description training supplies no supervision for how changes in text should affect predictions. We have not assessed representation quality independently, including distances between intervention embeddings. These experiments cannot distinguish inadequate representations, representations the surrogate has not learned to use, and context already implicit in a task-specific model. The expected non-identifiability under this design is a methodological caution, not evidence against text conditioning generally.

The aligned runs use a fixed $\lambda_{\text{align}} = 0.1$; no selection or sensitivity study is reported. A same-task-prompt control with $\lambda_{\text{align}} = 0$, followed by a weight sweep, is needed to isolate alignment and assess robustness. Proposition 1 holds independently of that weight, but empirical rankings need not. A natural next experiment jointly trains on varying coefficients, boundaries, forcing, geometry, or horizons, with held-out contexts, exact numerical metadata, and representation checks. More seeds and data splits would improve uncertainty assessment.

6 Conclusion

The constant-prompt results and fixed-prompt non-identifiability result show why improvement over an unconditioned baseline is insufficient evidence that text meaning helped. The observed means and prompt interventions do not isolate the causes of performance differences or establish semantic steering. Testing that capability requires varying physical context during training, matched capacity and loss controls, and evidence that the text representation preserves the relevant meaning.

Acknowledgements

Funding. This work was supported in part by Dell Technologies. Lennon J. Shikhman and Michael Galarnyk acknowledge financial support from Dell Technologies. Computational resources provided by Dell Technologies and used in this work include a Dell Pro Max T2 workstation equipped with an Intel Core Ultra 9 285K processor, 128 GB of DDR5 ECC memory, and an NVIDIA RTX PRO 6000 Blackwell GPU, and a Dell Pro Max 16 Plus laptop equipped with an Intel Core Ultra 9 285HX processor, 128 GB of DDR5-6400 memory, and an NVIDIA RTX PRO 5000 Blackwell GPU.

References

- [1] Tianyu Chen, Haoyi Zhou, Ying Li, Hao Wang, Chonghan Gao, Rongye Shi, Shanghang Zhang, and Jianxin Li. OmniArch: Building foundation model for scientific computing. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267, pages 9860–9887. PMLR, 2025. URL <https://proceedings.mlr.press/v267/chen25cp.html>.
- [2] Wenqian Chen, Zhi-Feng Wei, Yucheng Fu, Michael Penwarden, Pratanu Roy, and Panos Stinis. ArGenT: Arbitrary geometry-encoded transformer for operator learning. *arXiv preprint arXiv:2602.11626*, 2026. URL <https://arxiv.org/abs/2602.11626>.
- [3] Yanlai Chen and Shawn Koohy. GPT-PINN: Generative pre-trained physics-informed neural networks toward non-intrusive meta-learning of parametric PDEs. *Finite Elements in Analysis and Design*, 228: 104047, 2024. URL <https://arxiv.org/abs/2303.14878>.
- [4] Woojin Cho, Kookjin Lee, Donsub Rim, and Noseong Park. Hypernetwork-based meta-learning for low-rank physics-informed neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/24f8dd1b8f154f1ee0d7a59e368eccf3-Abstract-Conference.html.
- [5] Wenhan Gao, Ruichen Xu, Yuefan Deng, and Yi Liu. Discretization-invariance? on the discretization mismatch errors in neural operators. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=J9Fgrq00ni>.
- [6] Jayesh K. Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized PDE modeling. *Transactions on Machine Learning Research*, 2023. URL <https://arxiv.org/abs/2209.15616>.
- [7] Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. GNOT: A general neural operator transformer for operator learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202, pages 12556–12569. PMLR, 2023. URL <https://proceedings.mlr.press/v202/hao23c.html>.
- [8] Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. PMLR, 2024. URL <https://arxiv.org/abs/2403.03542>.
- [9] Sheikh Md Shakeel Hassan, Arthur Feeney, Akash Dhruv, Jihoon Kim, Youngjoon Suh, Jaiyoung Ryu, Yoonjin Won, and Aparna Chandramowlishwaran. BubbleML: A multiphase multiphysics dataset and benchmarks for machine learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, volume 36, 2023. URL <https://arxiv.org/abs/2307.14623>.
- [10] Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. URL <https://arxiv.org/abs/2405.19101>.
- [11] Felix Koehler, Simon Niedermayr, Rüdiger Westermann, and Nils Thuerey. APEBench: A benchmark for autoregressive neural emulators of PDEs. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, volume 37, 2024. URL <https://arxiv.org/abs/2411.00180>.
- [12] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023. URL <http://jmlr.org/papers/v24/21-1524.html>.

- [13] Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations’ operator learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL <https://openreview.net/forum?id=EPPqt3uERT>.
- [14] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=c8P9NQVtmn0>.
- [15] Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388): 1–26, 2023. URL <http://jmlr.org/papers/v24/23-0064.html>.
- [16] Yuxuan Liu, Jingmin Sun, Xinjie He, Griffin Pinney, Zecheng Zhang, and Hayden Schaeffer. PROSE-FD: A multimodal PDE foundation model for learning multiple operators for forecasting fluid dynamics. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*, 2024. URL <https://openreview.net/forum?id=7tEA7ENjLd>.
- [17] Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. PROSE: Predicting multiple operators and symbolic expressions using multimodal transformers. *Neural Networks*, 180:106707, 2024. URL <https://doi.org/10.1016/j.neunet.2024.106707>.
- [18] Cooper Lorsung and Amir Barati Farimani. PICL: Physics-informed contrastive learning for partial differential equations. *APL Machine Learning*, 2(4):046107, 2024. URL <https://doi.org/10.1063/5.0223651>.
- [19] Cooper Lorsung and Amir Barati Farimani. Explain like i’m five: Using LLMs to improve PDE surrogate models with text, 2024. URL <https://arxiv.org/abs/2410.01137>.
- [20] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deepoNet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, March 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5. URL <http://dx.doi.org/10.1038/s42256-021-00302-5>.
- [21] Yining Luo, Yingfa Chen, and Zhen Zhang. CFDBench: A large-scale benchmark for machine learning methods in fluid dynamics. *arXiv preprint arXiv:2310.05963*, 2023. URL <https://arxiv.org/abs/2310.05963>.
- [22] Michael McCabe, Bruno Régald-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, Mariel Pettee, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. Multiple physics pretraining for spatiotemporal surrogate models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 119301–119335, 2024. URL <https://openreview.net/forum?id=DKSI3bULiZ>.
- [23] Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régald-Saint Blancard, François Rozet, Liam Holden Parker, Miles Cranmer, and Shirley Ho. The well: A large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, volume 37, 2024. URL <https://openreview.net/forum?id=00Sx577BT3>.
- [24] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. URL <https://ojs.aaai.org/index.php/AAAI/article/view/11671>.
- [25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/radford21a.html>.
- [26] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045. URL <https://doi.org/10.1016/j.jcp.2018.10.045>.

- [27] Bogdan Raonic, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bezenac. Convolutional neural operators for robust and accurate learning of PDEs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=MtekhXRP4h>.
- [28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.
- [29] Lennon Shikhman. Diagnosing failure modes of neural operators across diverse PDE families. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL <https://openreview.net/forum?id=OS1LWZHQYn>.
- [30] Lennon Shikhman. One operator to rule them all? on boundary-indexed operator families in neural PDE solvers. In *AI&PDE: ICLR 2026 Workshop on AI and Partial Differential Equations*, 2026. URL <https://openreview.net/forum?id=1DjWQ9UxRy>.
- [31] Lennon J. Shikhman. Semigroup consistency as a diagnostic for learned physics simulators. In *ICML 2026 Workshop on AI for Physics*, 2026. URL <https://openreview.net/forum?id=MeAF0ZnrVM>.
- [32] Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 1596–1611. Curran Associates, Inc., 2022. URL <https://arxiv.org/abs/2210.07182>.
- [33] Liu Yang, Siting Liu, Tingwei Meng, and Stanley J. Osher. In-context operator learning with data prompts for differential equation problems. *Proceedings of the National Academy of Sciences*, 120(39): e2310142120, 2023. doi: 10.1073/pnas.2310142120. URL <https://www.pnas.org/doi/abs/10.1073/pnas.2310142120>.
- [34] Liu Yang, Siting Liu, and Stanley J. Osher. Fine-tune language models as multi-modal differential equation solvers, 2023. URL <https://arxiv.org/abs/2308.05061>.
- [35] Zhanhong Ye, Xiang Huang, Leheng Chen, Zining Liu, Bingyang Wu, Hongsheng Liu, Zidong Wang, and Bin Dong. PDEformer-1: A foundation model for one-dimensional partial differential equations. *arXiv preprint arXiv:2407.06664*, 2024. URL <https://arxiv.org/abs/2407.06664>.
- [36] Hang Zhou, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. Unisolver: PDE-conditional transformers towards universal neural PDE solvers. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*. PMLR, 2025. URL <https://arxiv.org/abs/2405.17527>.

A Run-level results and implementation details

Table 2 gives every final-epoch test error on the fixed split. Matching seed labels do not guarantee identical backbone initializations or minibatch orders because model construction changes random-number consumption. The runs measure optimization variation, not uncertainty over splits or PDE regimes. Table 1 reports sample standard deviations, not confidence intervals. Its percentages are $100(1 - \bar{e}_C/\bar{e}_U)$ and $100(1 - \bar{e}_A/\bar{e}_C)$, using unrounded means.

Table 2: All 27 existing runs: final-epoch test relative L^2 , lower is better. Each row contains three separately trained models.

Task	Seed	Unconditioned	Constant	Fixed prompt + align.
Darcy2D	7	0.032205	0.030741	0.049860
	17	0.033520	0.030068	0.034352
	27	0.045690	0.029452	0.032868
ShallowWater2D	7	0.00155104	0.00118240	0.00135158
	17	0.00155618	0.00129879	0.00128726
	27	0.00197349	0.00170078	0.00148120
CNS3D	7	0.409692	0.413856	0.391852
	17	0.403565	0.398259	0.401415
	27	0.400125	0.408121	0.420486

Text pathway. The encoder uses 128-dimensional token embeddings, masked mean pooling, a biased $128 \rightarrow 256 \rightarrow 128$ MLP with GELU, and L^2 normalization. It is randomly initialized and jointly trained, with no language pretraining, attention, positional encoding, or dropout. Pooling excludes padding and is invariant to token order. The lowercase tokenizer retains alphanumeric characters, underscores, periods, plus/minus signs, and slashes. Each run fits its vocabulary on at most 4,096 training examples and pads or truncates to 128 tokens; unseen tokens share one unknown token. Vocabulary sizes, including padding and unknown tokens, are 11 for constant, 32 for aligned 2D, and 37 for aligned CNS3D. The output conditions each FNO block through $(1 + \gamma)h + \beta$, with the linear map producing γ, β initialized to zero.

Optimization and alignment. The encoder, FiLM heads, and FNO are optimized jointly with normalized-field mean squared error; aligned runs additionally backpropagate the weighted contrastive loss through the text encoder and trainable alignment heads. All use AdamW with learning rate 5×10^{-4} , weight decay 10^{-5} , model gradient clipping at norm 1, and mixed precision disabled. The evidence encoder concatenates per-channel means, population standard deviations, minima, and maxima of normalized inputs and targets, then applies an MLP with hidden width 256, GELU, and output width 128. Biased linear $128 \rightarrow 128$ text and data projections precede normalization and InfoNCE with fixed temperature 0.07. Final-epoch losses are approximately 1.386294 in 2D and 0.693167–0.693221 in 3D. Neither convergence near $\log B$ nor Proposition 1 identifies the cause of predictive-error differences.

Parameter accounting. Surrogate parameter counts are 4,743,937 / 4,877,313 / 4,880,001 in 2D and 18,889,397 / 19,006,261 / 19,009,589 in 3D for unconditioned / constant / aligned, using the same implementation counting convention. Vocabulary sizes explain the difference between conditioned models. The training-only evidence encoder and projections add 68,224 parameters in 2D and 76,416 in 3D.

Interventions and reproducibility. Interventions reuse the final aligned model and training vocabulary. Shuffling permutes identical descriptions; missing/constant substitutes the generic sentence. Corruption replaces task, dimension, channels, target step, and other metadata values. Paraphrasing changes the template while retaining facts. Numeric metadata serializes labeled facts as text, not direct numerical conditioning. Figure 2 summarizes $100(e_{\text{intervention}} - e_{\text{correct}})/e_{\text{correct}}$ within each model across seeds. Tables are generated by `scripts/build_review_tables.py`; `generated/review_evidence.json` records source metrics and hashes.