LIGHTWEIGHT EXPLAINABLE PHYSICS-INFORMED NEURAL NETWORKS BY LEARNABLE ACTIVATION FUNCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Physics-informed neural networks (PINNs) provide a data-efficient framework for solving partial differential equations, but improving their accuracy often requires larger multilayer perceptron (MLP) backbones, increasing parameter count and computational cost. We investigate whether learnable activation functions can improve PINN performance while preserving lightweight backbones and explainable nonlinearities. We adapt the Cannistraci-Muscoloni-Gu generalized logistic-logit function (CMG) to PINNs and introduce CMG-GELU, which replaces the tanh term in the GELU approximation with learnable CMG. To support stable PINN training, we reformulate CMG's implicit logit phase into an explicit differentiable approximation and adopt a layer-wise parameterization that adds only two trainable activation parameters per hidden layer. We compare CMG and CMG-GELU with nine representative activation and architecture baselines across 25 forward and inverse PINN tasks using five random seeds. Compared with representative PINN method studies, our 25-task suite covers substantially more tasks, enabling a broader cross-task evaluation. CMG-GELU delivers the strongest aggregate performance and the highest cross-task consistency among the 11 methods, achieving the best mean rank, rank standard deviation, and Robustness Performance Ranking Index (RPRI), while CMG ranks second by RPRI. Learned activation parameters reveal task- and layer-dependent activation modulation patterns, providing parameter-level explainability, while low-dimensional parameter-profile embeddings provide exploratory task-level evidence of local organization among tasks with related physical regimes. These results support CMG-based learnable activations as a lightweight and explainable approach to improving PINNs across heterogeneous problems.

1 INTRODUCTION

Partial differential equations (PDEs) are fundamental for describing physical phenomena across science and engineering. Physics-informed neural networks (PINNs) provide a scientific machine learning framework for solving forward and inverse differential-equation problems by jointly fitting observations and governing physical laws (Raissi et al., 2019; Cuomo et al., 2022; Karniadakis et al., 2021). By embedding the governing equations into the training loss, PINNs can approximate PDE solutions and incorporate sparse observations without relying entirely on grid-based numerical solvers. A common realization uses a multilayer perceptron (MLP) whose derivatives are evaluated through automatic differentiation. However, enlarging the MLP increases parameter count and computational cost, and it may also introduce additional optimization difficulties in PINNs, including vanishing gradients and initialization pathologies that may lead to performance degradation(Es'kin et al., 2025; Jiang et al., 2025). Recent studies further suggest that compact and shallow PINN architectures can provide favorable training efficiency and mitigate some optimization difficulties associated with deeper networks (Rishi et al., 2024; Hu et al., 2024; Es'kin et al., 2025). This motivates lightweight designs that improve the nonlinear representation without substantially enlarging the MLP backbone.

One way to improve PINN performance is through architectural and representational modifications. Representative examples include Fourier features (Wang et al., 2021b), PirateNet (Wang et al., 2024). A complementary direction concerns the choice of activation function. PINN studies have employed

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

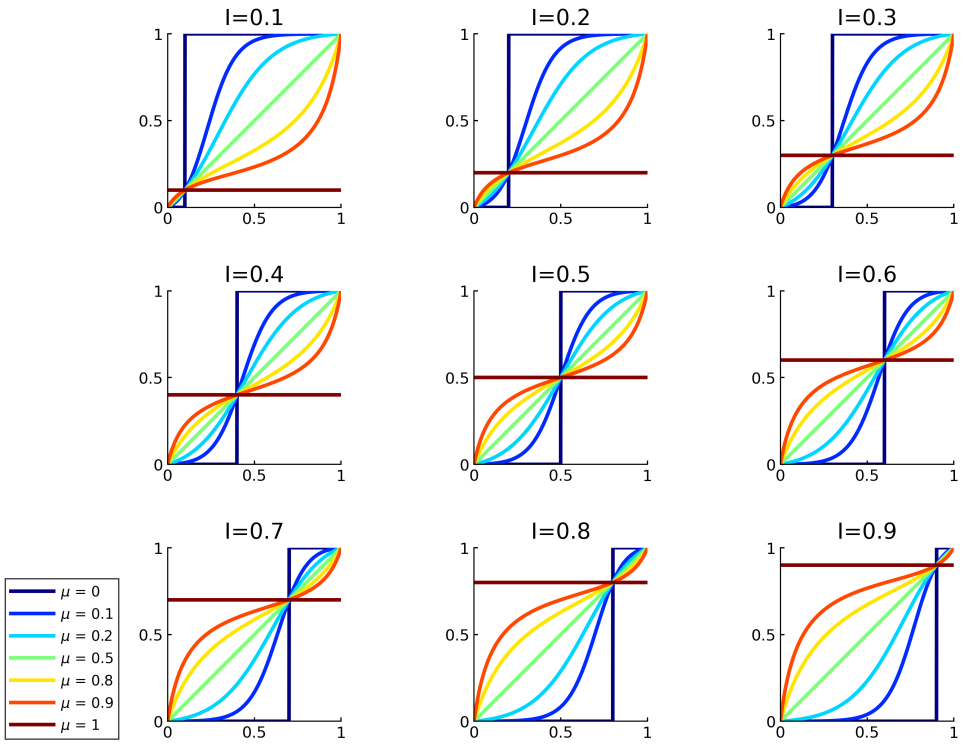


Figure 1: CMG-Explicit activation curves for various normalized deviate-inflection point I and inflection-rate parameter μ values. $0 < \mu < 0.5$ corresponds to the logistic phase, whereas $0.5 < \mu < 1$ corresponds to the logit phase; $\mu = 0$ and $\mu = 1$ are the step function and constant mappings. Announcement and suppression are defined relative to the identity transformation of the normalized activation input.

fixed nonlinearities such as tanh, GELU, Swish/SiLU, sine, and Sigmexd (Cuomo et al., 2022; Michaloglou et al., 2026; Gryt & Przystałka, 2026; Kurukulasuriya et al., 2026). Fixed activations provide simple nonlinear bases, but their functional form does not adapt to individual tasks or hidden layers. Learnable activation functions provide greater flexibility: for example, L-LAAF learns a layer-wise slope for tanh, while REAct uses a learnable rational-exponential parameterization (Jagtap et al., 2020; Mishra et al., 2025). However, increasing activation flexibility also raises questions about parameter efficiency and the interpretability of the learned nonlinear shape. It therefore remains important to determine whether a parameter-compact learnable activation can maintain competitive performance across heterogeneous physical problems while retaining explainable layer-wise nonlinear behavior.

We investigate this question using two learnable activation functions based on the Cannistraci-Muscoloni-Gu generalized logistic-logit function (CMG) (Gu et al., 2026). CMG is governed by two geometrically meaningful shape parameters: the inflection-rate parameter μ controls the transition between logistic and logit modulation regimes, while the normalized deviate-inflection-point parameter I governs the asymmetry (Figure 1). Relative to the identity transformation, CMG can selectively amplify or attenuate different regions of the normalized activation input, which we refer to as feature announcement and feature suppression, respectively. These geometrically interpretable parameters make the learned activation shapes directly inspectable and form the basis of our parameter-level explainability analysis. In our layer-wise parameterization, each hidden layer learns one μ and one I, adding only two trainable activation parameters per hidden layer. We further introduce CMG-GELU, which replaces the fixed tanh term in the standard GELU approximation with learnable CMG (Hendrycks & Gimpel, 2016). To support stable PINN training, we reformulate the implicit logit-phase approximation of CMG as an explicit differentiable expression using Newton's method with one approximation step (Kelley, 2003).

Table 1: Task categories. 1D and 2D denote spatial dimensionality; +T indicates an additional temporal coordinate; T-only denotes a time-only ODE; IDE denotes an integro-differential equation; inv denotes an inverse problem.

Category	Tasks
Linear diffusive systems	Heat equation [1D+T]; Diffusion–Reaction [1D+T]; Inverse Heat [1D+T, inv]
Nonlinear diffusive systems	Burgers [1D+T]; Allen–Cahn [1D+T]; Fisher–KPP [1D+T]; Gray–Scott [2D+T]
Dispersive-wave systems	Dynamic Beam [1D+T]; KdV [1D+T]; Schrödinger [1D+T]
Spatial-fractional systems	Fractional Poisson [1D]; Fractional Diffusion [1D+T]
Additional benchmark tasks	Underdamped vibration [T-only]; Nonlinear Telegraph [1D+T]; Klein–Gordon [1D+T]; Poisson L-shape [2D]; Laplace Disk [2D]; Helmholtz [2D]; Inverse Poisson [1D, inv]; Volterra IDE [IDE]; Darcy Flow [2D]; Kovasznay flow [2D]; Taylor–Green [2D+T]; Shallow Water [2D+T]; Inverse Brinkman [1D, inv]

A broad evaluation is important because PINN behavior can vary substantially across forward and inverse problems, dimensions, governing equations, and physical regimes. We therefore construct a benchmark of 25 forward and inverse PINN tasks. The benchmark includes four equation families whose members share common physical mechanisms—linear diffusive systems, nonlinear diffusive systems, dispersive-wave systems, and spatial-fractional systems—together with additional tasks that broaden its coverage across other equations, physical settings, and inverse problems (Table 1). The governing equations, domains, conditions, and evaluation metrics are provided in Appendix A.

The breadth of existing PINN evaluations also varies considerably across method studies. Table 2 summarizes the number of benchmark tasks reported in representative studies of activation and architecture methods. Relative to the studies summarized here, our 25-task benchmark provides a broader setting for assessing both average task-wise performance and cross-task consistency.

We compare CMG and CMG-GELU with nine representative activation and architecture baselines—Vanilla MLP, GELU, Swish, sine, Sigmexd, L-LAAF, REAct, Fourier, and PirateNet—across all 25 tasks using a controlled five-seed protocol. Models are approximately parameter-matched within each task while retaining their defining architectures or activation designs. We assess both average task-wise performance and cross-task consistency, and further analyze the learned CMG and CMG-GELU parameter profiles across hidden layers and tasks. At the task level, low-dimensional embeddings of these profiles are used to examine whether tasks with related physical regimes occupy neighboring regions of the learned-parameter space.

Across the 25-task benchmark, CMG-GELU achieves the lowest mean rank, rank standard deviation, and Robustness Performance Ranking Index (RPRI) among the 11 compared methods, indicating a favorable combination of average task-wise performance and cross-task consistency. CMG ranks second by RPRI. The learned activation profiles additionally reveal task- and layer-dependent nonlin-

Table 2: Benchmark coverage in representative PINN method studies

Method	Reference	Number of tasks
L-LAAF	Jagtap et al. (2020)	2
Fourier	Wang et al. (2021b)	4
PirateNet	Wang et al. (2024)	10
REAct	Mishra et al. (2025)	7
Sigmexd	Kurukulasuriya et al. (2026)	9
Current article	–	25

ear behavior, providing parameter-level explainability and exploratory task-level evidence of local organization among tasks with related physical regimes in the learned activation-parameter space.

Our main contributions are:

1. **Learnable CMG-based activations for PINNs.** We develop an explicit differentiable formulation of CMG suitable for stable PINN training and introduce CMG-GELU, a learnable CMG-based extension of GELU. Both designs require only two trainable activation parameters per hidden layer.
2. **Broad and controlled benchmark.** We evaluate the proposed activations against nine representative activation and architecture baselines in an approximately parameter-matched benchmark comprising 25 tasks, 11 methods, and five random seeds, spanning diverse forward and inverse PINN problems.
3. **Parameter- and task-level explainability.** We analyze learned layer-wise activation profiles to reveal task- and depth-dependent nonlinear behavior, including feature-announcement and feature-suppression patterns in CMG, and further examine their task-level organization in the learned-parameter space through low-dimensional embeddings.

2 METHODS

2.1 BENCHMARK AND COMPARATOR METHODS

We compare 11 methods across the 25 forward and inverse PINN tasks summarized in Table 1. The benchmark spans diverse governing equations, spatial and temporal dimensions, and both forward and inverse settings. Appendix A, Tables A1–A5, provides the governing equations, domains, boundary and initial conditions, and primary evaluation metrics for every task.

The proposed CMG and CMG-GELU formulations are introduced in Section 3.1. The controlled benchmark compares these two methods with nine representative activation and architecture baselines:

- **Vanilla MLP:** a three-hidden-layer MLP with tanh activation used as the reference architecture.
- ⋆ **Fixed activations:** GELU, Swish, Sine, and Sigmexd, implemented as the reference MLP with the corresponding activation (Hendrycks & Gimpel, 2016; Ramachandran et al., 2017; Gryt & Przystałka, 2026; Kurukulasuriya et al., 2026).
- ⋆ **Adaptive activations:** L-LAAF and REAct, using trainable-slope and rational-exponential activations, respectively (Jagtap et al., 2020; Mishra et al., 2025).
- ▲ **Architecture baselines:** Fourier, an MLP preceded by a Fourier-feature coordinate mapping, and PirateNet, a residual-adaptive architecture under the shared lightweight protocol (Wang et al., 2021b; 2024).

Throughout the method, parameter-count, and result tables, ⋆ denotes an activation-based method, ▲ an architecture/representation method, and • the baseline PINN.

For the Burgers formulation study in Table 3, we additionally compare node-wise and layer-wise CMG parameterizations. In the node-wise configuration, each hidden neuron learns an independent (μ, I) pair, whereas all neurons within a hidden layer share one pair in the layer-wise configuration. All configurations in Table 3 are evaluated under the same five-seed protocol with closely matched model sizes.

2.2 PARAMETER-MATCHED ARCHITECTURES

For each task, all methods are evaluated under a common lightweight parameter budget. Hidden widths are selected by integer search to minimize the parameter-count gap while preserving each method's defining architecture, including method-specific trainable modules and activation parameters. Because exact equality is not always feasible, the models are approximately parameter-matched. Appendix B reports the resulting task–method parameter counts.

2.3 CONTROLLED TRAINING PROTOCOL

All experiments use three trainable hidden layers, and five random seeds. Training consists of task-specific Muon optimization (Jordan et al., 2024) followed by 15,000 L-BFGS iterations (Nocedal, 1980). For each task–method pair, we select the learning-rate configuration with the lowest five-seed mean primary error from the search spaces detailed in Appendix C, which also reports the random seeds and task-specific Muon budgets.

2.4 PARAMETER-PROFILE ANALYSIS AND TASK EMBEDDINGS

For Figures 2 and 3, each trained state is represented by the six-dimensional layer-wise activation profile

$$\mathbf{p} = [\mu_1, \mu_2, \mu_3, I_1, I_2, I_3].$$

CMG and CMG-GELU profiles are standardized separately across the task–seed records. For each task, the medoid of the five standardized seed profiles is selected by Euclidean distance, yielding one representative observed state; the same medoids are used for the heatmaps and embeddings. Principal component analysis (PCA) (Jolliffe & Cadima, 2016) is fitted separately to the 25 CMG and 25 CMG-GELU medoids, and the first two principal components form the two-dimensional embeddings.

We use the four equation families in Table 1 to examine whether tasks sharing common physical mechanisms exhibit local organization in the learned-parameter space. We quantify the separation of the tasks in these equation families using TSPS-AUPR (Acevedo et al., 2024), which can capture nonlinear group separation. Definitions, score aggregation are reported in Appendix G.

2.5 EVALUATION AND RANKING

Results are reported as mean primary error $\pm$ SEM across five seeds using each task's primary metric (Appendix A). Methods are ranked independently within each task by mean primary error, with rank 1 denoting the lowest error. Let $\bar{r}$ and σ_r denote the mean and standard deviation of the 25 task-wise ranks. We define the Robustness Performance Ranking Index (RPRI) as

$$\mathrm{RPRI} = \bar{r} + \sigma_r.$$

Lower RPRI indicates better average ranking and cross-task consistency. We also report worst rank and top-rank count, defined as the maximum task-wise rank and number of rank-1 tasks, respectively.

Computational overhead is profiled on Burgers for all 11 approximately parameter-matched methods on one NVIDIA RTX A6000 using three fresh-process repeats. Training times are estimated by short-run extrapolation from 1,000 timed Muon iterations after 100 warm-up iterations and 300 timed L-BFGS iterations; peak allocated GPU memory is recorded during optimization. Appendix F reports the resulting estimates.

3 RESULTS

3.1 CMG AND CMG-GELU DESIGN

The Cannistraci-Muscoloni-Gu generalized logistic-logit function (CMG) (Gu et al., 2026) is a two-parameter activation family governed by the inflection-rate parameter μ and normalized deviate-inflection-point parameter $I \in [0, 1]$. Its learnable shape permits adaptive feature announcement and suppression. To make its logit phase practical for repeated PINN differentiation, we use Newton's method (Kelley, 2003) with one approximation step to obtain an explicit differentiable form. The detailed derivation is provided in Appendix D.

Let $t = (x - x_{\min})/(x_{\max} - x_{\min})$. With input bounds $[x_{\min}, x_{\max}]$ and output bounds $[y_L, y_R]$, the explicit CMG activation is

$$\mathrm{CMG}(x) = y_L + (y_R - y_L) \begin{cases} \dfrac{1}{1 + \frac{1-t}{t} \exp\left[-\left(\frac{1}{\mu} - 2\right)(t - I)\right]}, & 0 < \mu \leq 0.5, \\ \dfrac{t + \left(\frac{1}{1-\mu} - 2\right) It(1-t)}{1 + \left(\frac{1}{1-\mu} - 2\right) t(1-t)}, & 0.5 < \mu < 1. \end{cases} \tag{1}$$

Figure 1 illustrates how μ and I control the activation relative to the identity transformation. Hereafter, we refer to the original formulation with implicit inversion of the logit phase as CMG-Implicit, and to the explicit Newton formulation as CMG-Explicit.

CMG-GELU replaces the tanh term in the standard GELU approximation (Hendrycks & Gimpel, 2016) with CMG:

$$\text{CMG-GELU}(x) = \frac{x}{2}\left[1 + \mathrm{CMG}\left(\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right)\right]. \tag{2}$$

We next validate the explicit and layer-wise design choices before evaluating the two proposed activations against the full set of baselines.

3.2 Validation of the explicit CMG formulation

Taking Burgers' equation as a representative testbed, we evaluate the explicit CMG formulation and compare its node-wise and layer-wise parameterizations. Table 3 compares the node-wise and layer-wise CMG-Explicit configurations with node-wise CMG-Implicit and the Vanilla MLP under the same five-seed protocol. Both CMG-Explicit configurations achieve lower mean relative L_2 error than CMG-Implicit and the Vanilla MLP. We therefore use the layer-wise configuration in all subsequent experiments because it retains the improvement over the Vanilla MLP while adding only two trainable activation parameters per hidden layer. Hereafter, unless otherwise specified, CMG and the CMG component of CMG-GELU use the explicit layer-wise formulation.

Table 3: Empirical validation of CMG parameterizations on Burgers' equation

Method	**Relative L_2 error (mean $\pm$ SEM)**
CMG-Explicit node-wise	3.42e-03 $\pm$ 1.33e-03
CMG-Implicit node-wise	3.33e-02 $\pm$ 1.06e-02
CMG-Explicit layer-wise	4.85e-03 $\pm$ 1.93e-03
Vanilla MLP (tanh)	6.89e-03 $\pm$ 1.45e-03

3.3 Twenty-five-task comparison

With the explicit layer-wise CMG formulation selected, we next evaluate CMG and CMG-GELU together with the nine baselines across the full 25-task PINN benchmark.

Table 4 shows that CMG-GELU provides the strongest aggregate performance across the benchmark. Among the 11 methods, it achieves the best mean rank (3.72), the lowest rank standard deviation (rank SD; 2.03), and consequently the best RPRI (5.75). CMG-GELU ranks first on four tasks, places among the top three on 13 of the 25 tasks, and never falls below seventh place. Its advantage therefore lies primarily in maintaining consistently strong performance across heterogeneous PINN problems, rather than relying on a small number of exceptional results. CMG-GELU also improves upon both the Vanilla MLP with tanh, the standard activation used in many PINNs, and the fixed GELU activation, which ranks fourth by mean rank. By replacing the fixed tanh term in the GELU approximation with learnable CMG, CMG-GELU retains the smooth gated-linear structure of GELU while allowing the activation shape to vary across tasks and hidden layers. It also provides a more compact form of layer-wise adaptation than REAct, using two constrained shape parameters per

hidden layer instead of a four-parameter rational-exponential activation. The results for Sine further highlight the distinction between task-specific gains and cross-task robustness. Sine has the largest number of task-wise wins (six), but also has the highest rank SD and ranks tenth or eleventh on seven tasks. This pattern is consistent with a strong periodic inductive bias that can yield substantial gains on certain tasks but does not transfer uniformly across a heterogeneous benchmark suite. In contrast, CMG-GELU combines fewer task-wise wins with substantially greater ranking stability, demonstrating broader applicability under the shared training protocol.

CMG exhibits a similar advantage from learning the activation shape. It attains the second-best RPRI (7.63) and the third-best mean rank (5.12) in Table 4. Whereas L-LAAF introduces one trainable activation-slope parameter per hidden layer, CMG uses two geometrically interpretable parameters that control modulation regime and asymmetry. These parameters allow different hidden layers to learn distinct feature-announcement and feature-suppression patterns while adding only two trainable parameters per layer. PirateNet achieves a marginally better mean rank than CMG (5.08 versus 5.12), whereas CMG has a lower rank SD (2.51 versus 2.86) and a better worst rank (9 versus 11), resulting in a better RPRI. Thus, CMG-GELU offers the strongest overall balance between average ranking and cross-task consistency, while CMG remains competitive with a lightweight layer-wise parameterization.

Table 4: Twenty-five-task method summary. Parenthetical values indicate ranks among the 11 methods for each summary statistic; lower is better except for top-rank count.

Method	RPRI	Mean rank	Rank SD	Worst rank	Top-rank count
⋆ CMG-GELU	**5.75 (1)**	**3.72 (1)**	**2.03 (1)**	**7 (1)**	4
⋆ CMG	**7.63 (2)**	**5.12 (3)**	**2.51 (3)**	**9 (2)**	1
▲ PirateNet	**7.94 (3)**	**5.08 (2)**	2.86 (6)	11 (5)	2
⋆ GELU	8.10 (4)	5.44 (4)	2.66 (5)	10 (4)	2
⋆ REAct	8.18 (5)	6.00 (7)	**2.18 (2)**	**9 (2)**	1
⋆ L-LAAF	9.22 (6)	5.68 (5)	3.54 (9)	11 (5)	3
• Vanilla MLP	9.45 (7)	6.84 (9)	2.61 (4)	11 (5)	2
⋆ Sigmexd	9.63 (8)	6.72 (8)	2.91 (7)	11 (5)	0
⋆ Sine	9.64 (9)	5.76 (6)	3.88 (11)	11 (5)	6
⋆ Swish	10.09 (10)	6.96 (10)	3.13 (8)	11 (5)	1
▲ Fourier	12.45 (11)	8.68 (11)	3.77 (10)	11 (5)	3

Taken together, the advantage of CMG-GELU and CMG is not that they rank first on every task, but with only two layer-wise shape parameters per hidden layer, they achieve a favorable balance between average performance and cross-task stability. These results support CMG-based learnable activations as an effective approach to improving lightweight PINNs.

Detailed five-seed mean $\pm$ SEM values for every task are reported in Appendix E, Tables E1–E9.

3.4 Computational overhead

Introducing learnable activation functions, together with their additional trainable parameters and computations, naturally incurs computational overhead. As reported in Tables F1 and F2, the increases in training time and peak GPU memory for CMG and CMG-GELU are reasonable on the Burgers efficiency benchmark. Among the top-ranked methods in the 25-task benchmark, CMG and PirateNet achieve similarly strong aggregate performance, with CMG attaining a better RPRI and PirateNet a marginally better mean rank. Compared with PirateNet, CMG requires less training time, while the difference in peak GPU memory is small. CMG-GELU incurs additional overhead relative to CMG, but at the same time achieves the best mean rank and RPRI among the 11 methods. These results suggest a favorable trade-off between predictive performance and computational cost in CMG and CMG-GELU.

3.5 PARAMETER-LEVEL AND TASK-LEVEL EXPLAINABILITY

Beyond predictive performance, the learned activation parameters provide a means of examining how the nonlinear transformations vary across tasks and network depths. Figure 2A and Figure 3A show the layer-wise parameter profiles learned by CMG-GELU and CMG, respectively. CMG-GELU exhibits a broader distribution of learned μ and I parameters across tasks and layers, whereas CMG parameters are more concentrated, with μ values mainly in the logistic regime and I remaining close to 0.5. The layer-wise variation of μ and I reveals the activation shapes favored at different network depths and, for CMG, the corresponding feature-announcement and feature-suppression patterns. The distinct distributions observed for the two activation families show that they use their learnable nonlinearities differently across tasks and layers, providing parameter-level explainability for the trained networks.

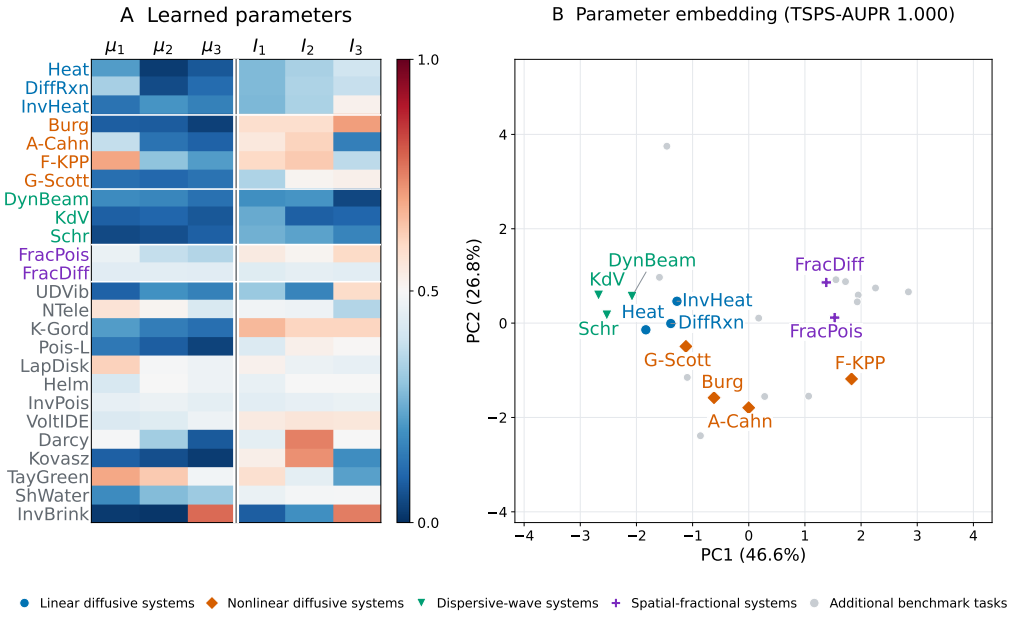


Figure 2: CMG-GELU parameter profiles and task embedding. Panel A shows the layer-wise μ and I parameters for all 25 tasks. Panel B shows the two-dimensional PCA embedding, with the 12 tasks in four equation families labeled and scored by TSPS-AUPR; the additional benchmark tasks are shown as background points. Task abbreviations are listed in Table A6.

To examine these profiles at the task level, we apply principal component analysis (PCA) separately to the layer-wise activation parameters learned by CMG-GELU and CMG. The resulting two-dimensional embeddings allow us to assess whether tasks with related physical regimes occupy neighboring regions of the learned-parameter space (Figures 2B and 3B). The embeddings are annotated using the four equation families in Table 1.

Appendix G reports six separability metrics. CMG-GELU scores 1.000 across all six, while CMG ranges from 0.729 to 0.902. We report TSPS-AUPR in the main text because its travelling-salesman projection can capture nonlinear separation; it yields 1.000 for CMG-GELU and 0.902 for CMG, the highest separability score for CMG among the six metrics.

In the CMG-GELU embedding, tasks from the same group tend to occupy neighboring regions (Figure 2B), yielding a TSPS-AUPR of 1.000. This organization suggests that shared physical regimes are reflected in the activation-parameter profiles learned by CMG-GELU. CMG exhibits a similar overall organization (Figure 3B), but its TSPS-AUPR is lower at 0.902, with greater within-group dispersion, particularly within the dispersive-wave family. These physically related group separability patterns in the PCA representation space provide evidence of physically related similarities in the learned activation-parameter space, supporting task-level explainability of the parameters learned via CMG-based activation functions.

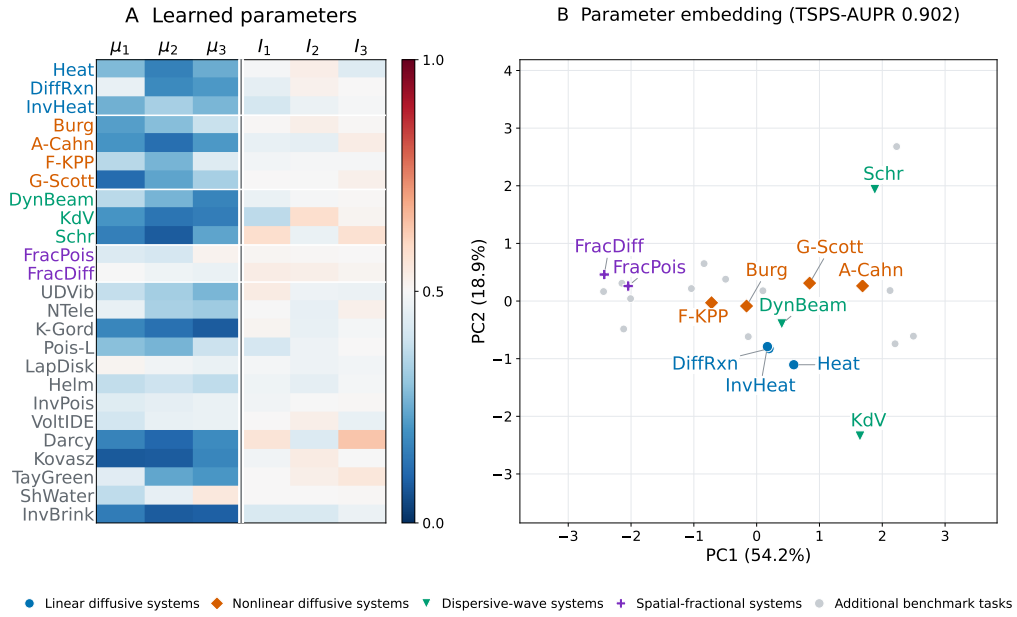


Figure 3: CMG parameter profiles and task embedding. Panel A shows the layer-wise μ and I parameters for all 25 tasks. Panel B shows the two-dimensional PCA embedding, with the 12 tasks in four equation families labeled and scored by TSPS-AUPR; the additional benchmark tasks are shown as background points. Task abbreviations are listed in Table A6.

4 Discussion

This work shows that learnable activation functions can improve lightweight PINNs without substantially enlarging the underlying MLP backbone. CMG and CMG-GELU add only two trainable activation parameters per hidden layer while allowing the nonlinear transformation to adapt across tasks and network depths. Across the controlled 25-task, 11-method benchmark, CMG-GELU delivers the strongest aggregate performance and the highest cross-task consistency, while CMG ranks second by RPRI. This improved performance is accompanied by reasonable computational overhead on the Burgers benchmark: CMG requires less training time than PirateNet with only a small difference in peak GPU memory, while CMG-GELU incurs higher overhead but achieves the strongest aggregate performance. Together, these results suggest that learnable activation functions provide a favorable performance–cost trade-off for lightweight PINNs.

Beyond predictive performance, the learned activation parameters provide parameter-level explainability by revealing task- and depth-dependent activation modulation patterns. The PCA and separability analyses further provide task-level evidence that tasks with related physical regimes exhibit local organization in the learned-parameter space, with particularly strong separability for CMG-GELU. These analyses show that the learned nonlinearities are not only adaptive, but also provide a means of examining how activation behavior varies across tasks and network depths.

However, these results are best interpreted in terms of aggregate performance and cross-task consistency rather than uniform task-wise superiority, as the best-performing method varies across individual tasks. In addition, the current CMG and CMG-GELU implementations are not specifically optimized with fused hardware-aware kernels; specialized Triton or CUDA implementations may further reduce their computational overhead (Tillet et al., 2019; Ansel et al., 2024). Finally, our controlled benchmark intentionally focuses on lightweight three-hidden-layer PINN backbones. Evaluating substantially deeper or larger architectures is outside the scope of this study and is left for future work.

AI USE STATEMENT

During the preparation of this manuscript, generative AI tools were used to assist with language editing. All AI-assisted edits were reviewed by the authors, who remain responsible for the final content of the manuscript.

REPRODUCIBILITY STATEMENT

The task definitions, training protocol, and evaluation procedures are described in the Methods and Appendix. The code is provided in the supplementary material.

REFERENCES

Aldo Acevedo, Claudio Duran, Ming-Ju Kuo, Sara Ciucci, Michael Schroeder, and Carlo Vittorio Cannistraci. Measuring Group Separability in Geometrical Space for Evaluation of Pattern Recognition and Dimension Reduction Algorithms. *IEEE Access*, 10:22441–22471, 2022. ISSN 2169-3536. doi: 10.1109/ACCESS.2022.3152789. URL https://ieeexplore.ieee.org/document/9716930/.

Aldo Acevedo, Yue Wu, Fabio Lorenzo Traversa, and Carlo Vittorio Cannistraci. Geometric separability of mesoscale patterns in embedding representation and visualization of multidimensional data and complex networks. *PLOS Complex Systems*, 1(2):e0000012, October 2024. ISSN 2837-8830. doi: 10.1371/journal.pcsy.0000012. URL https://dx.plos.org/10.1371/journal.pcsy.0000012.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pp. 929–947, La Jolla CA USA, April 2024. ACM. ISBN 979-8-4007-0385-0. doi: 10.1145/3620665.3640366. URL https://dl.acm.org/doi/10.1145/3620665.3640366.

Pi-Yueh Chuang and Lorena A. Barba. Experience report of physics-informed neural networks in fluid simulations: pitfalls and frustration, July 2022. URL http://arxiv.org/abs/2205.14249. arXiv:2205.14249 [physics.flu-dyn].

Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next. *Journal of Scientific Computing*, 92(3):88, September 2022. ISSN 0885-7474, 1573-7691. doi: 10.1007/s10915-022-01939-z. URL https://link.springer.com/10.1007/s10915-022-01939-z.

Alemayehu Tamirie Deresse and Alemu Senbeta Bekela. A deep learning approach: physics-informed neural networks for solving a nonlinear telegraph equation with different boundary conditions. *BMC Research Notes*, 18(1):77, February 2025. ISSN 1756-0500. doi: 10.1186/s13104-025-07142-1. URL https://bmcresnotes.biomedcentral.com/articles/10.1186/s13104-025-07142-1.

Vasiliy A. Es'kin, Alexey O. Malkhanov, and Mikhail E. Smorkalov. Are two hidden layers still enough for the physics-informed neural networks? *Journal of Computational Physics*, 537: 114085, September 2025. ISSN 00219991. doi: 10.1016/j.jcp.2025.114085. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999125003687.

Przemysław Gryt and Piotr Przystałka. Physics-Informed Neural Networks in Aerospace Engineering: A Systematic Review of Architectures, Training Strategies, and Open Challenges. *Applied Sciences*, 16(13):6282, June 2026. ISSN 2076-3417. doi: 10.3390/app16136282. URL https://www.mdpi.com/2076-3417/16/13/6282.

Wenqi Gu, Yingtao Zhang, Alessandro Muscoloni, and Carlo Vittorio Cannistraci. A generalized logistic-logit function and its application to multi-layer perceptron and neuron segmentation. *Frontiers in Artificial Intelligence*, 9:1785867, June 2026. ISSN 2624-8212. doi: 10.3389/frai.2026.1785867. URL https://www.frontiersin.org/articles/10.3389/frai.2026.1785867/full.

Zhongkai Hao, Jiachen Yao, Chang Su, Hang Su, Ziao Wang, Fanzhi Lu, Zeyu Xia, Yichi Zhang, Songming Liu, Lu Lu, and Jun Zhu. PINNacle: A Comprehensive Benchmark of Physics-Informed Neural Networks for Solving PDEs. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 76721–76774. Curran Associates, Inc., 2024. doi: 10.52202/079017-2442. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/8c63299fb2820ef41cb05e2ff11836f5-Paper-Datasets_and_Benchmarks_Track.pdf.

Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs), 2016. URL https://arxiv.org/abs/1606.08415. Version Number: 5.

Wei-Fan Hu, Yi-Jun Shih, Te-Sheng Lin, and Ming-Chih Lai. A shallow physics-informed neural network for solving partial differential equations on static and evolving surfaces. *Computer Methods in Applied Mechanics and Engineering*, 418:116486, January 2024. ISSN 00457825. doi: 10.1016/j.cma.2023.116486. URL https://linkinghub.elsevier.com/retrieve/pii/S0045782523006102.

Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2239):20200334, July 2020. ISSN 1364-5021, 1471-2946. doi: 10.1098/rspa.2020.0334. URL http://royalsocietypublishing.org/rspa/article/80756.

Feilong Jiang, Xiaonan Hou, Jianqiao Ye, and Min Xia. Deeper-PINNs: Unlocking the power of deep physics-informed neural networks. *Applied Soft Computing*, 185:114048, December 2025. ISSN 15684946. doi: 10.1016/j.asoc.2025.114048. URL https://linkinghub.elsevier.com/retrieve/pii/S1568494625013614.

Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, April 2016. ISSN 1364-503X, 1471-2962. doi: 10.1098/rsta.2015.0202. URL http://royalsocietypublishing.org/rsta/article/115142.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Taniya Kapoor, Hongrui Wang, Alfredo Nunez, and Rolf Dollevoet. Physics-informed neural networks for solving forward and inverse problems in complex beam systems. *IEEE Transactions on Neural Networks and Learning Systems*, 35(5):5981–5995, May 2024. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2023.3310585. URL http://arxiv.org/abs/2303.01055. arXiv:2303.01055 [cs.LG].

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, May 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00314-5. URL https://www.nature.com/articles/s42254-021-00314-5.

C. T. Kelley. *Solving Nonlinear Equations with Newton's Method*. Society for Industrial and Applied Mathematics, January 2003. ISBN 978-0-89871-546-0 978-0-89871-889-8. doi: 10.1137/1.9780898718898. URL https://epubs.siam.org/doi/book/10.1137/1.9780898718898.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael Mahoney. Characterizing possible failure modes in physics-informed neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 26548–26560. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/df438e5206f31600e6ae4af72f2725f1-Paper.pdf.

Manoj Kurukulasuriya, Chanaka Batuwatta-Gamage, Chaminda Karunasena, Hyogu Jeong, Kalani Ranathunga, Charith Rathnayaka, and YuanTong Gu. A novel Sigmexd activation function to enhance Physics-Informed Neural Networks (PINNs) for differential equations with a perspective on the role of activation functions in PINNs. *Neurocomputing*, 666:132364, February 2026. ISSN 09252312. doi: 10.1016/j.neucom.2025.132364. URL https://linkinghub.elsevier.com/retrieve/pii/S092523122503036X.

Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. DeepXDE: A Deep Learning Library for Solving Differential Equations. *SIAM Review*, 63(1):208–228, January 2021. ISSN 0036-1445, 1095-7200. doi: 10.1137/19M1274067. URL https://epubs.siam.org/doi/10.1137/19M1274067.

Alkmini Michaloglou, Ioannis Papadimitriou, Ilias Gialampoukidis, Stefanos Vrochidis, and Ioannis Kompatsiaris. Physics-Informed Neural Networks in Materials Modeling and Design: A Review. *Archives of Computational Methods in Engineering*, 33(4):5223–5260, May 2026. ISSN 1134-3060, 1886-1784. doi: 10.1007/s11831-025-10448-9. URL https://link.springer.com/10.1007/s11831-025-10448-9.

Sourav Mishra, Shreya Hallikeri, and Suresh Sundaram. REAct: Rational Exponential Activation for Better Learning and Generalization in PINNs. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, Hyderabad, India, April 2025. IEEE. ISBN 979-8-3503-6874-1. doi: 10.1109/ICASSP49660.2025.10887788. URL https://ieeexplore.ieee.org/document/10887788/.

Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980. ISSN 1088-6842, 0025-5718. doi: 10.1090/S0025-5718-1980-0572855-7. URL https://www.ams.org/mcom/1980-35-151/S0025-5718-1980-0572855-7/.

Guofei Pang, Lu Lu, and George Em Karniadakis. fPINNs: Fractional Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing*, 41(4):A2603–A2626, January 2019. ISSN 1064-8275, 1095-7197. doi: 10.1137/18M1229845. URL https://epubs.siam.org/doi/10.1137/18M1229845.

Michael Penwarden, Ameya D. Jagtap, Shandian Zhe, George Em Karniadakis, and Robert M. Kirby. A unified scalable framework for causal sweeping strategies for Physics-Informed Neural Networks (PINNs) and their temporal decompositions. *Journal of Computational Physics*, 493:112464, November 2023. ISSN 00219991. doi: 10.1016/j.jcp.2023.112464. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999123005594.

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*, 378:686–707, February 2019. ISSN 00219991. doi: 10.1016/j.jcp.2018.10.045. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999118307125.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for Activation Functions, October 2017. URL http://arxiv.org/abs/1710.05941. arXiv:1710.05941 [cs.NE].

J. Rishi, Azhar Gafoor, Sumanth Kumar, and Deepak Subramani. On the Training Efficiency of Shallow Architectures for Physics Informed Neural Networks. In Leonardo Franco, Clélia De Mulatier, Maciej Paszynski, Valeria V. Krzhizhanovskaya, Jack J. Dongarra, and Peter M. A. Sloot (eds.), *Computational Science – ICCS 2024*, volume 14834, pp. 363–377. Springer Nature Switzerland, Cham, 2024. ISBN 978-3-031-63758-2 978-3-031-63759-9. doi: 10.1007/978-3-031-63759-9_39. URL https://link.springer.com/10.1007/978-3-031-63759-9_39. Series Title: Lecture Notes in Computer Science.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench Datasets, 2022. URL https://darus.uni-stuttgart.de/citation?persistentId=doi:10.18419/darus-2986.

Philippe Tillet, H. T. Kung, and David Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pp. 10–19, Phoenix AZ USA, June 2019. ACM. ISBN 978-1-4503-6719-6. doi: 10.1145/3315508.3329973. URL https://dl.acm.org/doi/10.1145/3315508.3329973.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing*, 43(5): A3055–A3081, January 2021a. ISSN 1064-8275, 1095-7197. doi: 10.1137/20M1318043. URL https://epubs.siam.org/doi/10.1137/20M1318043.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, October 2021b. ISSN 00457825. doi: 10.1016/j.cma.2021.113938. URL https://linkinghub.elsevier.com/retrieve/pii/S0045782521002759.

Sifan Wang, Bowen Li, Yuhan Chen, and Paris Perdikaris. PirateNets: Physics-informed Deep Learning with Residual Adaptive Networks. *Journal of Machine Learning Research*, 25(402): 1–51, 2024. URL http://jmlr.org/papers/v25/24-0313.html.

Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, April 2022. ISSN 00457825. doi: 10.1016/j.cma.2022.114823. URL https://linkinghub.elsevier.com/retrieve/pii/S0045782522001438.

A TASK DEFINITIONS

Tables A1–A5 summarize the governing equations, computational domains, initial and boundary conditions, and primary evaluation metrics for the 25 benchmark tasks. Here, u^* denotes a reference solution.

Linear diffusive systems are governed by local diffusive transport with linear state dynamics.

Table A1: Linear diffusive systems task definitions

Task	Definition
Heat equation (Lu et al., 2021)	**Equation:** $u_t - 0.4u_{xx} = 0$ **Domain:** $x, t \in [0, 1]$ **Conditions:** $u(0,t) = u(1,t) = 0$; $u(x,0) = \sin(\pi x)$ **Primary metric:** Global relative L_2
Diffusion–Reaction (Yu et al., 2022)	**Equation:** $u_t - u_{xx} = e^{-t}\left[\frac{3}{2}\sin(2x) + \frac{8}{3}\sin(3x) + \frac{15}{4}\sin(4x) + \frac{63}{8}\sin(8x)\right]$ **Domain:** $x \in [-\pi, \pi]$, $t \in [0, 1]$ **Conditions:** $u(x,0) = \sin x + \frac{\sin(2x)}{2} + \frac{\sin(3x)}{3} + \frac{\sin(4x)}{4} + \frac{\sin(8x)}{8}$; $u(\pm\pi, t) = 0$ **Primary metric:** Global relative L_2
Inverse Heat (Lu et al., 2021)	**Equation:** $u_t - Du_{xx} + e^{-t}\big[\sin(\pi x) - \pi^2 \sin(\pi x)\big] = 0$; recover D, $D^* = 1$ **Domain:** $x \in [-1, 1]$, $t \in [0, 1]$ **Conditions:** $u(\pm 1, t) = 0$; $u(x,0) = \sin(\pi x)$ **Primary metric:** Relative error of D

Nonlinear diffusive systems couple local diffusion with state-dependent nonlinear dynamics.

Table A2: Nonlinear diffusive systems task definitions

Task	Definition
Burgers (Raissi et al., 2019)	**Equation:** $u_t + uu_x - \frac{0.01}{\pi}u_{xx} = 0$ **Domain:** $x \in [-1, 1]$, $t \in [0, 0.99]$ **Conditions:** $u(\pm 1, t) = 0$; $u(x,0) = -\sin(\pi x)$ **Primary metric:** Global relative L_2
Allen–Cahn (Yu et al., 2022)	**Equation:** $u_t - 0.001u_{xx} - 5(u - u^3) = 0$ **Domain:** $x \in [-1, 1]$, $t \in [0, 1]$ **Conditions:** $u(x,0) = x^2\cos(\pi x)$; $u(\pm 1, t) = -1$ **Primary metric:** Global relative L_2
Fisher–KPP (Krishnapriyan et al., 2021)	**Equation:** $u_t - 2u_{xx} - 5u(1 - u) = 0$ **Domain:** $x \in [0, 2\pi)$, $t \in [0, 1]$ **Conditions:** $u(x,0) = \exp\left[-\left(\frac{x - \pi}{\pi/4}\right)^2\right]$; $u(0,t) = u(2\pi, t)$, $u_x(0,t) = u_x(2\pi, t)$ **Primary metric:** Global relative L_2

Continued on the next page.

Table A2 (continued). Nonlinear diffusive systems task definitions

Task	Definition
Gray–Scott (Hao et al., 2024)	**Equation:** $u_t = D_u\Delta u - uv^2 + F(1-u)$, $v_t = D_v\Delta v + uv^2 - (F+k)v$, $(D_u, D_v, F, k) = (10^{-5}, 5\times10^{-6}, 0.04, 0.1)$ **Domain:** $(x, y) \in [-1, 1]^2$, $t \in [0, 20]$ **Conditions:** $u(x, y, 0) = 1 - e^{-80[(x+0.05)^2+(y+0.02)^2]}$; $v(x, y, 0) = e^{-80[(x-0.05)^2+(y-0.02)^2]}$; periodic u, v **Primary metric:** Global relative L_2 over u, v

Dispersive-wave systems exhibit wave dynamics with wavelength-dependent propagation and phase evolution.

Table A3: Dispersive-wave systems task definitions

Task	Definition
Dynamic Beam (Kapoor et al., 2024)	**Equation:** $u_{tt} + u_{xxxx} = 0$ **Domain:** $x, t \in [0, 1]$ **Conditions:** $u(x, 0) = \sin(\pi x)$; $u_t(x, 0) = 0$; $u = u_{xx} = 0$ at $x = 0, 1$ **Primary metric:** Global relative L_2
KdV (Penwarden et al., 2023)	**Equation:** $u_t + uu_x + 0.0025u_{xxx} = 0$ **Domain:** $x \in [-1, 1]$, $t \in [0, 1]$ **Conditions:** $u(x, 0) = \cos(\pi x)$; $u(-1, t) = u(1, t)$, $u_x(-1, t) = u_x(1, t)$ **Primary metric:** Global relative L_2
Schrödinger (Raissi et al., 2019)	**Equation:** $u_t + \frac{1}{2}v_{xx} + (u^2 + v^2)v = 0$, $v_t - \frac{1}{2}u_{xx} - (u^2 + v^2)u = 0$ **Domain:** $x \in [-5, 5]$, $t \in [0, \pi/2]$ **Conditions:** $u(x, 0) = 2\,\mathrm{sech}(x)$, $v(x, 0) = 0$; periodic u, v, u_x, v_x **Primary metric:** Relative L_2 error of $\sqrt{u^2 + v^2}$

Spatial-fractional systems represent long-range, non-local spatial interactions.

Table A4: Spatial-fractional systems task definitions

Task	Definition
Fractional Poisson (Pang et al., 2019)	**Equation:** $D^{1.5}u = f$, $u^*(x) = x^3(1-x)^3$, $f = D^{1.5}u^*$ **Domain:** $x \in [0, 1]$ **Conditions:** $u(0) = u(1) = 0$ **Primary metric:** Global relative L_2
Fractional Diffusion (Pang et al., 2019)	**Equation:** $u_t + Q_{1.8,52}u = f$, $u^*(x, t) = x(1-x)t^2$, $f = \partial_t u^* + Q_{1.8,52}u^*$ **Domain:** $x, t \in [0, 1]$ **Conditions:** $u(x, 0) = 0$; $u(0, t) = u(1, t) = 0$ **Primary metric:** Global relative L_2

Nonlocal operators. For Fractional Poisson, D^α denotes the unnormalized sum of the left- and right-sided Grünwald–Letnikov derivatives, with $\alpha = 1.5$ and zero exterior extension. For Fractional Diffusion, let $r_j = (j + \frac{1}{2})/52$, $j = 0, \ldots, 51$, and define

$$\widetilde{u}(\xi, t) = \begin{cases} u(\xi, t), & |\xi| \le 1, \\ 0, & |\xi| > 1. \end{cases}$$

The nonlocal operator is

$$Q_{\alpha,52}u(x,t) = \frac{1}{2\cdot 52^2}\sum_{d\in\{-1,1\}}\sum_{j=0}^{51}\frac{u(x,t)-\widetilde{u}(x+dr_j,t)}{r_j^{1+\alpha}}, \qquad \alpha = 1.8.$$

Residuals and evaluation errors are computed on $(x,t)\in[0,1]^2$. For nonlocal queries with $\xi\in[-1,0)$, the operator uses the same network extension $u_\theta(\xi,t)=\xi(1-\xi)tN_\theta(\xi,t)$, where N_θ denotes the network output before the output transform, without separately prescribed exterior values. The forcing is constructed using the polynomial reference extension $u^*(\xi,t)=\xi(1-\xi)t^2$.

Table A5: Additional benchmark task definitions

Task	Definition
Underdamped vibration (Mishra et al., 2025)	**Equation:** $u_{tt}+3u_t+9u=0$ **Domain:** $t\in[0,1]$ **Conditions:** $u(0)=1$; $u_t(0)=0$ **Primary metric:** Global relative L_2
Nonlinear Telegraph (Deresse & Bekela, 2025)	**Equation:** $u_{tt}+4u_t+u^3-u_{xx}=f$, $f=-2e^{-t}\sin x+e^{-3t}\sin^3 x$ **Domain:** $x\in[0,2\pi]$, $t\in[0,0.5]$ **Conditions:** $u(x,0)=\sin x$; $u_t(x,0)=-\sin x$; $u(0,t)=u(2\pi,t)=0$ **Primary metric:** Global relative L_2
Klein–Gordon (Wang et al., 2021a)	**Equation:** $u_{tt}-u_{xx}+u^3=f$, $u^*=x\cos(5\pi t)+(xt)^3$, $f=\partial_{tt}u^*-\partial_{xx}u^*+(u^*)^3$ **Domain:** $x,t\in[0,1]$ **Conditions:** $u(x,0)=x$; $u_t(x,0)=0$; $u(0,t)=0$; $u(1,t)=\cos(5\pi t)+t^3$ **Primary metric:** Global relative L_2
Poisson L-shape (Lu et al., 2021)	**Equation:** $-u_{xx}-u_{yy}=1$ **Domain:** $\Omega=\{(x,y)\in(-1,1)^2:\ x<0 \text{ or } y<0\}$ **Conditions:** $u=0$ on $\partial\Omega$ **Primary metric:** Global relative L_2
Laplace Disk (Lu et al., 2021)	**Equation:** $ru_r+r^2u_{rr}+u_{\theta\theta}=0$ **Domain:** $r\in[0,1]$, $\theta\in[0,2\pi]$ **Conditions:** $u(1,\theta)=\cos\theta$; angular periodicity enforced by $(r,\theta)\mapsto(r\sin\theta, r\cos\theta)$ **Primary metric:** Global relative L_2
Helmholtz (Lu et al., 2021)	**Equation:** $-\Delta u-(2\pi)^2u=f$, $u^*=\sin(2\pi x)\sin(2\pi y)$, $f=4\pi^2u^*$ **Domain:** $(x,y)\in[0,1]^2$ **Conditions:** $u=0$ on $\partial\Omega$ **Primary metric:** Global relative L_2
Inverse Poisson (Lu et al., 2021)	**Equation:** $u_{xx}-q=0$; recover q, $q^*(x)=-\pi^2\sin(\pi x)$ **Domain:** $x\in[0,1]$ **Conditions:** $u(0)=u(1)=0$ **Primary metric:** Relative L_2 error of q
Volterra IDE (Lu et al., 2021)	**Equation:** $u'(x)+u(x)-\int_0^x e^{t-x}u(t)\,dt=0$ **Domain:** $x\in[0,5]$ **Conditions:** $u(0)=1$ **Primary metric:** Global relative L_2

Continued on the next page.

Table A5 (continued). Additional benchmark task definitions

Task	Definition
Darcy Flow (Takamoto et al., 2022)	**Equation:** $\mathbf{q} + a(x,y)\nabla p = 0,\ \nabla\cdot\mathbf{q} = 1;$ $a(x,y)$ from the fixed first PDEBench Darcy sample **Domain:** $(x,y) \in [0,1]^2$ **Conditions:** $p = 0$ on $\partial\Omega$ **Primary metric:** Normalized global relative L_2 over p, q_x, q_y
Kovasznay flow (Lu et al., 2021)	**Equation:** $uu_x + vu_y + p_x - \nu\Delta u = 0,$ $uv_x + vv_y + p_y - \nu\Delta v = 0,$ $u_x + v_y = 0,$ $\nu = 1/20$ **Domain:** $(x,y) \in [-0.5,1] \times [-0.5,1.5]$ **Conditions:** $u = u^*,\ v = v^*$ on $\partial\Omega$; $p = p^*$ at $x = 1$, where $\lambda = (2\nu)^{-1} - \sqrt{(4\nu^2)^{-1} + 4\pi^2},$ $u^* = 1 - e^{\lambda x}\cos(2\pi y),$ $v^* = \frac{\lambda}{2\pi} e^{\lambda x}\sin(2\pi y),$ $p^* = \frac{1}{2}(1 - e^{2\lambda x})$ **Primary metric:** Mean relative L_2 over u, v, p
Taylor–Green (Chuang & Barba, 2022)	**Equation:** $\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - 0.01\Delta\mathbf{u} = 0,$ $\nabla\cdot\mathbf{u} = 0$ **Domain:** $(x,y) \in [0,2\pi]^2,\ t \in [0,1]$ **Conditions:** $u(x,y,0) = -\cos x \sin y;$ $v(x,y,0) = \sin x \cos y;$ $p(x,y,0) = -\frac{1}{4}(\cos 2x + \cos 2y)$; periodic u, v, p **Primary metric:** Normalized global relative L_2 over u, v, p
Shallow Water (Takamoto et al., 2022)	**Equation:** $h_t + (m_x)_x + (m_y)_y = 0,$ $(m_x)_t + \left(\frac{m_x^2}{h_{\text{safe}}} + \frac{1}{2}gh^2\right)_x$ $+ \left(\frac{m_x m_y}{h_{\text{safe}}}\right)_y = 0,$ $(m_y)_t + \left(\frac{m_x m_y}{h_{\text{safe}}}\right)_x$ $+ \left(\frac{m_y^2}{h_{\text{safe}}} + \frac{1}{2}gh^2\right)_y = 0,$ $m_x = hu,\ m_y = hv,\ g = 1,$ $h_{\text{safe}} = \max(h, 10^{-4})$ **Domain:** $(x,y) \in [-1,1]^2,\ t \in [0,0.25]$ **Conditions:** $h(x,y,0) = 1 + 0.2e^{-20(x^2+y^2)};$ $m_x(x,y,0) = m_y(x,y,0) = 0;$ $\partial_n h = \partial_n m_x = \partial_n m_y = 0$ on $\partial\Omega$ **Primary metric:** Normalized global relative L_2 over h, m_x, m_y
Inverse Brinkman (Yu et al., 2022)	**Equation:** $-\frac{\nu_e}{0.4}u_{xx} + \frac{10^{-3}}{K}u = 1;$ recover ν_e, K, $\nu_e^* = K^* = 10^{-3}$ **Domain:** $x \in [0,1]$ **Conditions:** $u(0) = u(1) = 0$ **Primary metric:** Mean relative error of ν_e and K

Table A6: Task abbreviations used in Figures 2 and 3

Abbrev.	Full task name
Heat	Heat equation
DiffRxn	Diffusion–Reaction
InvHeat	Inverse Heat
Burg	Burgers
A-Cahn	Allen–Cahn
F-KPP	Fisher–KPP
G-Scott	Gray–Scott
DynBeam	Dynamic Beam
KdV	KdV
Schr	Schrödinger
FracPois	Fractional Poisson
FracDiff	Fractional Diffusion
UDVib	Underdamped vibration
NTele	Nonlinear Telegraph
K-Gord	Klein–Gordon
Pois-L	Poisson L-shape
LapDisk	Laplace Disk
Helm	Helmholtz
InvPois	Inverse Poisson
VoltIDE	Volterra IDE
Darcy	Darcy Flow
Kovasz	Kovasznay flow
TayGreen	Taylor–Green
ShWater	Shallow Water
InvBrink	Inverse Brinkman

B TRAINABLE PARAMETER COUNTS

Tables B1–B7 report the total number of trainable parameters for every canonical task–method model.

Table B1: Trainable parameter counts

Method	Heat equation	Diffusion–Reaction	Inverse Heat	Burgers
⋆ CMG-GELU	2247	3787	4328	2247
⋆ CMG	2247	3787	4328	2247
▲ PirateNet	2202	3912	4133	2202
⋆ GELU	2241	3781	4322	2241
⋆ REAct	2253	3793	4334	2253
• Vanilla MLP	2241	3781	4322	2241
⋆ Sigmexd	2241	3781	4322	2241
⋆ Swish	2241	3781	4322	2241
⋆ L-LAAF	2244	3784	4325	2244
⋆ Sine	2241	3781	4322	2241
▲ Fourier	2211	3853	4244	2211

Table B2: Trainable parameter counts

Method	Allen–Cahn	Fisher–KPP	Gray–Scott	Dynamic Beam
⋆ CMG-GELU	2247	8323	7938	2815
⋆ CMG	2247	8323	7938	2815
▲ PirateNet	2202	8468	8039	2902
⋆ GELU	2241	8317	7932	2809
⋆ REAct	2253	8329	7944	2821
• Vanilla MLP	2241	8317	7932	2809
⋆ Sigmexd	2241	8317	7932	2809
⋆ Swish	2241	8317	7932	2809
⋆ L-LAAF	2244	8320	7935	2812
⋆ Sine	2241	8317	7932	2809
▲ Fourier	2211	8361	7952	2823

Table B3: Trainable parameter counts

Method	KdV	Schrödinger	Fractional Poisson	Fractional Diffusion
⋆ CMG-GELU	4327	26798	2215	2815
⋆ CMG	4327	26798	2215	2815
▲ PirateNet	4132	26683	2127	2902
⋆ GELU	4321	26792	2209	2809
⋆ REAct	4333	26804	2221	2821
• Vanilla MLP	4321	26792	2209	2809
⋆ Sigmexd	4321	26792	2209	2809
⋆ Swish	4321	26792	2209	2809
⋆ L-LAAF	4324	26795	2212	2812
⋆ Sine	4321	26792	2209	2809
▲ Fourier	4243	26714	2179	2823

Table B4: Trainable parameter counts

Method	Underdamped vibration	Nonlinear Telegraph	Klein–Gordon	Poisson L-shape
⋆ CMG-GELU	7754	12327	8323	8323
⋆ CMG	7754	12327	8323	8323
▲ PirateNet	7695	12340	8468	8468
⋆ GELU	7748	12321	8317	8317
⋆ REAct	7760	12333	8329	8329
• Vanilla MLP	7748	12321	8317	8317
⋆ Sigmexd	7748	12321	8317	8317
⋆ Swish	7748	12321	8317	8317
⋆ L-LAAF	7751	12324	8320	8320
⋆ Sine	7748	12321	8317	8317
▲ Fourier	7775	12463	8361	8361

Table B5: Trainable parameter counts

Method	Laplace Disk	Helmholtz	Inverse Poisson	Volterra IDE
⋆ CMG-GELU	2247	12327	4144	2215
⋆ CMG	2247	12327	4144	2215
▲ PirateNet	2202	12340	4287	2127
⋆ GELU	2241	12321	4138	2209
⋆ REAct	2253	12333	4150	2221
• Vanilla MLP	2241	12321	4138	2209
⋆ Sigmexd	2241	12321	4138	2209
⋆ Swish	2241	12321	4138	2209
⋆ L-LAAF	2244	12324	4141	2212
⋆ Sine	2241	12321	4138	2209
▲ Fourier	2211	12463	4220	2179

Table B6: Trainable parameter counts

Method	Darcy Flow	Kovasznay flow	Taylor–Green
⋆ CMG-GELU	4233	8451	4277
⋆ CMG	4233	8451	4277
▲ PirateNet	4204	8572	4309
⋆ GELU	4227	8445	4271
⋆ REAct	4239	8457	4283
• Vanilla MLP	4227	8445	4271
⋆ Sigmexd	4227	8445	4271
⋆ Swish	4227	8445	4271
⋆ L-LAAF	4230	8448	4274
⋆ Sine	4227	8445	4271
▲ Fourier	4297	8463	4353

Table B7: Trainable parameter counts

Method	Shallow Water	Inverse Brinkman
⋆ CMG-GELU	8000	2217
⋆ CMG	8000	2217
▲ PirateNet	8089	2129
⋆ GELU	7994	2211
⋆ REAct	8006	2223
• Vanilla MLP	7994	2211
⋆ Sigmexd	7994	2211
⋆ Swish	7994	2211
⋆ L-LAAF	7997	2214
⋆ Sine	7994	2211
▲ Fourier	8011	2181

C TRAINING PROTOCOLS

The tables in this section provide the complete controlled training protocol used for the 25-task, 11-method, five-seed comparison.

For 24 tasks, learning rate candidate set is $\{5\text{e}{-4}, 1\text{e}{-3}, 2\text{e}{-3}\}$. For Inverse Poisson, the candidate set is $\{5\text{e}{-5}, 1\text{e}{-4}, 2\text{e}{-4}\}$. Methods with trainable activation parameters (CMG, CMG-GELU, REAct, and L-LAAF) search a 3×3 grid of backbone and activation-parameter learning rates, whereas the remaining methods search three backbone learning rates.

Table C1: Baseline training protocol

Item	**Protocol**
Network depth	Three trainable hidden layers
Random seeds	$\{0, 1, 2, 3, 4\}$
Optimization	Muon (Jordan et al., 2024) followed by 15,000 L-BFGS iterations (Nocedal, 1980)

Table C2: Task-specific Muon training budgets

Muon iterations	**Tasks**
5,000	Helmholtz
10,000	Fractional Poisson; Fisher–KPP; Volterra IDE; Fractional Diffusion; Taylor–Green; Dynamic Beam; Schrödinger
15,000	Burgers; Laplace Disk
20,000	Diffusion–Reaction; Inverse Poisson; Inverse Brinkman; Nonlinear Telegraph; Gray–Scott; Darcy Flow; Shallow Water; Heat equation
30,000	Kovasznay flow
40,000	Allen–Cahn; KdV; Klein–Gordon
50,000	Poisson L-shape; Inverse Heat; Underdamped vibration

D CMG-EXPLICIT DERIVATION

When $0 < \mu \leq 0.5$, CMG is in the logistic phase and admits an explicit analytical expression. When $0.5 < \mu < 1$, CMG enters the logit phase, which is defined as the inverse of the corresponding logistic-phase mapping. Because an exact inverse in a simple closed form is not available, we derive an explicit approximation by applying Newton's method with one approximation step (Kelley, 2003).

Normalization. Let $x \in [x_{\min}, x_{\max}]$ be the input and $y \in [y_L, y_R]$ be the mapped output limits. We normalize both axes to $[0, 1]$:

$$t = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \qquad s = \frac{y - y_L}{y_R - y_L}. \tag{3}$$

Logistic-phase expression. In normalized coordinates, the logistic-phase CMG can be written as

$$s = \sigma(\text{logit}(t) + \alpha(t - I)), \tag{4}$$

where $\sigma(z) = 1/(1 + \exp(-z))$, $\text{logit}(t) = \log(t/(1 - t))$, and $I \in [0, 1]$ is the deviate inflection point. In the logit phase,

$$\alpha = \frac{1}{1 - \mu} - 2, \qquad 0.5 < \mu < 1. \tag{5}$$

Inverse problem. To derive the logit-phase expression, we invert the normalized function:

$$t = \sigma(\mathrm{logit}(s) + \alpha(s - I)). \tag{6}$$

Applying the logit transform gives

$$\mathrm{logit}(t) = \mathrm{logit}(s) + \alpha(s - I). \tag{7}$$

Define

$$F(s) = \mathrm{logit}(s) + \alpha(s - I) - \mathrm{logit}(t). \tag{8}$$

Solving $F(s) = 0$ gives the logit-phase output.

One-step Newton approximation. Using $s_0 = t$ as the initial guess, one Newton step gives

$$s_1 = s_0 - \frac{F(s_0)}{F'(s_0)}. \tag{9}$$

At $s_0 = t$,

$$F(t) = \alpha(t - I), \qquad F'(t) = \frac{1}{t(1-t)} + \alpha. \tag{10}$$

Therefore,

$$s_1 = t - \frac{\alpha(t - I)}{\frac{1}{t(1-t)} + \alpha} = \frac{t + \alpha I t(1-t)}{1 + \alpha t(1-t)}. \tag{11}$$

Final expression. Returning to the original coordinates yields the explicit logit-phase approximation:

$$\mathrm{CMG}_{\mathrm{logit}}(x) \approx y_L + (y_R - y_L)\frac{t + \left(\frac{1}{1-\mu} - 2\right) I t(1-t)}{1 + \left(\frac{1}{1-\mu} - 2\right) t(1-t)}. \tag{12}$$

This expression avoids grid-based implicit inversion and supports direct automatic differentiation in PINN training.

E Detailed task-wise results

Tables E1–E9 report the mean primary error ± SEM over five seeds. Bold entries are the lowest eligible mean error within each task.

Table E1: Detailed task-wise results

Method	Heat equation	Diffusion–Reaction	Inverse Heat
⋆ CMG-GELU	**3.55e-04 ± 6.84e-05**	**9.63e-05 ± 1.34e-05**	1.82e-03 ± 9.02e-04
⋆ CMG	3.85e-04 ± 6.61e-05	1.40e-04 ± 7.39e-06	1.78e-03 ± 7.31e-04
▲ PirateNet	4.13e-04 ± 4.13e-05	1.36e-04 ± 1.17e-05	**1.32e-03 ± 3.60e-04**
⋆ GELU	5.09e-04 ± 4.61e-05	1.65e-04 ± 1.58e-05	2.39e-03 ± 7.88e-04
⋆ REAct	4.24e-04 ± 8.17e-05	1.51e-04 ± 2.11e-05	2.29e-03 ± 5.25e-04
• Vanilla MLP	7.26e-04 ± 1.52e-04	1.93e-04 ± 1.69e-05	3.13e-03 ± 4.53e-04
⋆ Sigmexd	5.74e-04 ± 5.21e-05	2.82e-04 ± 4.75e-05	1.89e-03 ± 5.25e-04
⋆ Swish	7.69e-04 ± 9.38e-05	2.61e-04 ± 2.56e-05	2.48e-03 ± 1.79e-04
⋆ L-LAAF	3.82e-04 ± 4.29e-05	1.06e-04 ± 6.91e-06	3.89e-03 ± 9.36e-04
⋆ Sine	6.21e-04 ± 1.43e-04	2.24e-04 ± 4.16e-05	2.11e-03 ± 6.41e-04
▲ Fourier	4.53e-04 ± 5.91e-05	2.85e-02 ± 8.30e-03	6.48e-01 ± 5.71e-02

Table E2: Detailed task-wise results

Method	Burgers	Allen–Cahn	Fisher–KPP
⋆ CMG-GELU	5.54e-02 ± 1.84e-02	6.66e-03 ± 1.48e-03	7.48e-04 ± 1.28e-04
⋆ CMG	**4.85e-03 ± 1.93e-03**	3.87e-03 ± 7.81e-04	5.94e-04 ± 1.67e-04
▲ PirateNet	5.84e-03 ± 1.38e-03	3.23e-03 ± 9.30e-04	1.18e-03 ± 4.04e-04
⋆ GELU	7.19e-02 ± 9.31e-03	1.72e-02 ± 5.17e-03	1.53e-03 ± 6.21e-04
⋆ REAct	5.08e-03 ± 5.54e-04	8.84e-03 ± 5.10e-03	7.72e-04 ± 2.67e-04
• Vanilla MLP	6.89e-03 ± 1.45e-03	7.57e-03 ± 2.22e-03	8.88e-04 ± 2.11e-04
⋆ Sigmexd	4.61e-02 ± 1.49e-02	4.08e-02 ± 8.91e-03	6.92e-04 ± 3.29e-04
⋆ Swish	7.36e-02 ± 2.18e-02	3.88e-02 ± 3.69e-03	1.25e-03 ± 3.47e-04
⋆ L-LAAF	9.36e-03 ± 3.58e-03	**2.82e-03 ± 6.44e-04**	5.14e-04 ± 1.57e-04
⋆ Sine	7.51e-02 ± 3.64e-03	1.99e-02 ± 2.13e-03	**4.58e-04 ± 1.19e-04**
▲ Fourier	1.90e-01 ± 2.66e-02	3.38e-03 ± 1.05e-03	6.28e-01 ± 1.40e-02

Table E3: Detailed task-wise results

Method	Gray–Scott	Dynamic Beam	KdV
⋆ CMG-GELU	6.84e-02 ± 3.75e-03	1.49e-01 ± 5.46e-02	9.98e-03 ± 3.87e-03
⋆ CMG	1.21e-01 ± 1.20e-03	2.23e-01 ± 2.94e-02	5.08e-03 ± 4.21e-04
▲ PirateNet	9.76e-02 ± 6.36e-03	5.15e-02 ± 5.38e-03	4.08e-03 ± 3.95e-04
⋆ GELU	6.91e-02 ± 7.76e-03	3.50e-02 ± 8.86e-03	1.16e-02 ± 1.94e-03
⋆ REAct	1.19e-01 ± 3.06e-03	1.72e-01 ± 4.02e-02	7.16e-03 ± 1.25e-03
• Vanilla MLP	9.01e-02 ± 5.38e-03	1.57e-01 ± 5.27e-02	5.23e-03 ± 5.28e-04
⋆ Sigmexd	7.35e-02 ± 2.55e-03	5.39e-02 ± 7.81e-03	1.72e-02 ± 2.40e-03
⋆ Swish	6.83e-02 ± 1.03e-02	4.42e-02 ± 8.00e-03	1.99e-02 ± 1.30e-03
⋆ L-LAAF	1.26e-01 ± 2.32e-04	2.44e-01 ± 4.37e-02	**3.44e-03 ± 3.36e-04**
⋆ Sine	**4.80e-02 ± 1.07e-03**	**3.34e-02 ± 5.20e-03**	1.67e-02 ± 6.96e-03
▲ Fourier	1.24e-01 ± 1.29e-04	9.73e-01 ± 1.05e-01	1.68e-01 ± 9.87e-02

Table E4: Detailed task-wise results

Method	Schrödinger	Fractional Poisson	Fractional Diffusion
⋆ CMG-GELU	**6.46e-04 ± 3.51e-05**	1.09e-02 ± 2.47e-05	3.92e-05 ± 6.39e-06
⋆ CMG	1.10e-03 ± 2.99e-05	1.15e-02 ± 2.07e-04	8.57e-05 ± 1.23e-05
▲ PirateNet	1.04e-03 ± 1.26e-05	1.11e-02 ± 7.09e-05	3.77e-05 ± 8.48e-06
⋆ GELU	1.29e-03 ± 5.90e-05	1.10e-02 ± 4.38e-05	**3.42e-05 ± 3.47e-06**
⋆ REAct	9.95e-04 ± 3.82e-05	1.15e-02 ± 1.78e-04	6.96e-05 ± 9.31e-06
• Vanilla MLP	1.30e-03 ± 1.88e-05	1.20e-02 ± 4.57e-04	9.95e-05 ± 1.09e-05
⋆ Sigmexd	1.53e-03 ± 2.09e-04	1.11e-02 ± 4.20e-05	4.04e-05 ± 5.91e-06
⋆ Swish	1.51e-03 ± 1.06e-04	1.10e-02 ± 3.17e-05	4.43e-05 ± 6.92e-06
⋆ L-LAAF	1.08e-03 ± 2.76e-05	1.15e-02 ± 8.85e-05	1.03e-04 ± 1.67e-05
⋆ Sine	8.16e-04 ± 5.58e-05	1.19e-02 ± 1.85e-04	1.29e-04 ± 2.87e-05
▲ Fourier	2.85e-01 ± 2.30e-02	**1.06e-02 ± 1.56e-05**	3.32e-03 ± 4.28e-04

Table E5: Detailed task-wise results

Method	Underdamped vibration	Nonlinear Telegraph	Klein–Gordon
⋆ CMG-GELU	2.58e-06 ± 7.59e-07	3.24e-04 ± 5.99e-05	1.40e-02 ± 3.56e-03
⋆ CMG	1.86e-06 ± 4.40e-07	2.73e-04 ± 3.44e-05	9.82e-03 ± 1.08e-03
▲ PirateNet	3.30e-06 ± 7.86e-07	2.30e-04 ± 3.90e-05	1.43e-02 ± 2.78e-03
⋆ GELU	7.86e-06 ± 2.25e-06	3.82e-04 ± 7.20e-05	2.43e-02 ± 1.69e-03
⋆ REAct	3.20e-06 ± 1.30e-06	2.84e-04 ± 5.38e-05	1.53e-02 ± 3.17e-03
• Vanilla MLP	3.99e-06 ± 1.06e-06	3.56e-04 ± 5.10e-05	**8.10e-03 ± 1.81e-03**
⋆ Sigmexd	7.17e-06 ± 7.62e-07	4.26e-04 ± 4.03e-05	3.33e-02 ± 8.39e-03
⋆ Swish	8.68e-06 ± 1.61e-06	4.11e-04 ± 6.34e-05	3.59e-02 ± 4.55e-03
⋆ L-LAAF	1.97e-06 ± 6.36e-07	2.28e-04 ± 1.69e-05	1.07e-02 ± 2.55e-03
⋆ Sine	1.14e-05 ± 6.81e-06	**2.11e-04 ± 4.19e-05**	1.11e-02 ± 1.92e-03
▲ Fourier	**3.50e-07 ± 7.97e-08**	2.82e-04 ± 4.20e-05	8.39e-01 ± 7.06e-02

Table E6: Detailed task-wise results

Method	Poisson L-shape	Laplace Disk	Helmholtz
⋆ CMG-GELU	1.21e-01 ± 1.31e-03	7.27e-06 ± 1.14e-06	6.77e-06 ± 1.08e-06
⋆ CMG	1.22e-01 ± 1.01e-03	1.47e-05 ± 2.72e-06	1.12e-05 ± 2.49e-06
▲ PirateNet	1.22e-01 ± 1.37e-03	**1.73e-06 ± 5.13e-07**	8.41e-06 ± 1.39e-06
⋆ GELU	1.31e-01 ± 3.08e-03	9.56e-06 ± 2.92e-06	5.04e-06 ± 5.81e-07
⋆ REAct	1.23e-01 ± 1.06e-03	1.63e-05 ± 1.68e-06	**4.44e-06 ± 5.21e-07**
• Vanilla MLP	1.28e-01 ± 6.74e-04	1.69e-05 ± 4.28e-06	5.82e-06 ± 6.94e-07
⋆ Sigmexd	1.35e-01 ± 7.74e-04	9.49e-06 ± 2.30e-06	6.60e-06 ± 6.33e-07
⋆ Swish	1.39e-01 ± 4.09e-03	7.72e-06 ± 2.09e-06	9.96e-06 ± 9.88e-07
⋆ L-LAAF	1.21e-01 ± 6.02e-04	1.21e-05 ± 3.18e-06	2.56e-05 ± 2.82e-06
⋆ Sine	1.42e-01 ± 1.37e-03	2.69e-05 ± 5.36e-06	5.75e-06 ± 5.59e-07
▲ Fourier	**8.28e-02 ± 2.42e-02**	2.30e-04 ± 4.84e-05	5.85e-01 ± 3.01e-02

Table E7: Detailed task-wise results

Method	Inverse Poisson	Volterra IDE	Darcy Flow
⋆ CMG-GELU	6.80e-03 ± 1.53e-03	6.72e-04 ± 1.04e-04	9.89e-02 ± 1.38e-03
⋆ CMG	2.06e-02 ± 4.73e-03	4.53e-03 ± 4.39e-04	9.74e-02 ± 1.02e-03
▲ PirateNet	2.75e-02 ± 9.73e-03	7.13e-03 ± 6.68e-04	9.73e-02 ± 2.34e-03
⋆ GELU	**5.85e-03 ± 1.33e-03**	5.98e-04 ± 9.60e-05	9.83e-02 ± 8.83e-04
⋆ REAct	1.82e-02 ± 3.70e-03	4.96e-03 ± 5.16e-04	1.01e-01 ± 1.88e-03
• Vanilla MLP	1.90e-02 ± 4.08e-03	4.26e-03 ± 4.64e-04	**9.61e-02 ± 2.01e-04**
⋆ Sigmexd	1.53e-02 ± 2.34e-03	4.98e-04 ± 4.72e-05	1.02e-01 ± 1.25e-03
⋆ Swish	1.76e-02 ± 2.40e-03	**4.17e-04 ± 6.89e-05**	1.00e-01 ± 1.40e-03
⋆ L-LAAF	2.63e-02 ± 5.11e-03	1.06e-02 ± 6.49e-04	1.01e-01 ± 1.14e-03
⋆ Sine	1.14e-02 ± 1.57e-03	4.44e-03 ± 1.10e-04	9.65e-02 ± 6.97e-04
▲ Fourier	8.42e-02 ± 3.04e-02	3.59e-01 ± 6.83e-02	2.30e-01 ± 6.09e-03

Table E8: Detailed task-wise results

Method	Kovasznay flow	Taylor–Green
⋆ CMG-GELU	3.20e-04 ± 2.14e-05	5.90e-03 ± 2.59e-03
⋆ CMG	3.37e-04 ± 3.31e-05	2.33e-02 ± 6.92e-03
▲ PirateNet	3.42e-04 ± 3.66e-05	2.67e-02 ± 9.05e-03
⋆ GELU	3.20e-04 ± 1.14e-05	9.60e-03 ± 3.12e-03
⋆ REAct	4.06e-04 ± 6.06e-05	1.81e-02 ± 4.89e-03
• Vanilla MLP	3.71e-04 ± 1.83e-05	2.95e-02 ± 8.01e-03
⋆ Sigmexd	6.25e-04 ± 6.50e-05	1.96e-02 ± 1.22e-02
⋆ Swish	4.47e-04 ± 2.30e-05	9.82e-03 ± 2.06e-03
⋆ L-LAAF	**2.64e-04 ± 1.93e-05**	1.64e-02 ± 5.30e-03
⋆ Sine	3.85e-04 ± 3.61e-05	**4.06e-03 ± 1.03e-03**
▲ Fourier	5.84e-04 ± 7.77e-05	2.20e-01 ± 7.75e-02

Table E9: Detailed task-wise results

Method	Shallow Water	Inverse Brinkman
⋆ CMG-GELU	7.11e-03 ± 2.87e-05	**8.08e-03 ± 1.33e-03**
⋆ CMG	7.16e-03 ± 4.09e-05	9.27e-03 ± 1.87e-03
▲ PirateNet	7.20e-03 ± 4.10e-06	1.46e-02 ± 2.18e-03
⋆ GELU	6.96e-03 ± 2.63e-05	1.10e-02 ± 1.44e-03
⋆ REAct	7.17e-03 ± 3.76e-05	8.11e-03 ± 1.23e-03
• Vanilla MLP	7.18e-03 ± 1.86e-05	1.94e-02 ± 1.20e-03
⋆ Sigmexd	6.92e-03 ± 4.67e-05	1.11e-02 ± 2.60e-04
⋆ Swish	6.93e-03 ± 1.93e-05	1.32e-02 ± 1.52e-03
⋆ L-LAAF	7.17e-03 ± 3.86e-05	1.56e-02 ± 2.30e-03
⋆ Sine	**6.85e-03 ± 2.24e-05**	2.04e-02 ± 9.31e-04
▲ Fourier	6.95e-03 ± 5.89e-05	1.07e+00 ± 2.37e-01

F COMPUTATIONAL COST

The following tables report the computational costs measured on the Burgers benchmark for the 11 methods used in the main 25-task comparison. Values are 15,000-step training-time estimates obtained by short-run extrapolation; entries are mean ± SEM over three repeats. Peak allocated GPU memory was identical across the three fresh-process repeats for each method; therefore, the SEM was exactly zero, and thus not reported here.

Table F1: Estimated training time by method

Method	Muon time (min)	L-BFGS time (min)	Total time (min)
• Vanilla MLP	4.13 ± 0.03	10.51 ± 0.15	14.65 ± 0.17
⋆ Sine	4.23 ± 0.04	10.50 ± 0.15	14.73 ± 0.13
⋆ GELU	4.39 ± 0.03	10.71 ± 0.05	15.10 ± 0.05
⋆ L-LAAF	4.54 ± 0.04	10.61 ± 0.07	15.15 ± 0.09
⋆ Swish	4.57 ± 0.06	11.12 ± 0.22	15.69 ± 0.17
⋆ Sigmexd	4.58 ± 0.00	11.24 ± 0.14	15.83 ± 0.14
⋆ CMG	6.09 ± 0.03	12.29 ± 0.08	18.38 ± 0.11
▲ PirateNet	8.15 ± 0.01	12.77 ± 0.07	20.92 ± 0.08
⋆ CMG-GELU	7.16 ± 0.02	14.18 ± 0.25	21.34 ± 0.26
▲ Fourier	7.32 ± 0.23	14.31 ± 0.53	21.63 ± 0.63
⋆ REAct	8.45 ± 0.07	14.67 ± 0.15	23.12 ± 0.11

Table F2: Peak GPU memory by method

Method	Peak GPU memory (GiB)
• Vanilla MLP	0.075
⋆ Sine	0.077
⋆ L-LAAF	0.079
⋆ GELU	0.080
⋆ Sigmexd	0.083
▲ PirateNet	0.083
⋆ Swish	0.084
▲ Fourier	0.087
⋆ CMG	0.089
⋆ CMG-GELU	0.106
⋆ REAct	0.124

G SEPARABILITY METRIC SUMMARY

We evaluate the separability of the four equation families in the two-dimensional PCA embeddings using six separability metrics. The evaluation includes the 12 tasks belonging to these families. PSI-ROC, PSI-PR, and PSI-MCC follow the projection-separability framework of Acevedo et al. (2022). Here, each pair of families is projected onto the line connecting their coordinate-wise median centers, and separability is evaluated using ROC area, precision–recall area, or the Matthews correlation coefficient, respectively. CPS-AUPR and LDPS-AUPR evaluate precision–recall area after centroid and linear-discriminant projection, respectively. TSPS-AUPR instead uses cumulative distances along a travelling-salesman path, allowing nonlinear separation to be assessed (Acevedo et al., 2024).

For consistency across all six metrics, we aggregate the six pairwise scores using the maximizing-score aggregation of Acevedo et al. (2022), $\bar{s}/(1+\sigma_s)$, where $\bar{s}$ and σ_s denote their mean and sample standard deviation, respectively. Higher values favor stronger separation with less variation across family pairs. Under the median-centroid setting used here, PSI-PR and CPS-AUPR are equivalent and therefore numerically equivalent.

Table G1: Separability metric summary.

Metric	**CMG-GELU**	**CMG**
PSI-ROC	1.000	0.870
PSI-PR	1.000	0.851
PSI-MCC	1.000	0.729
CPS-AUPR	1.000	0.851
LDPS-AUPR	1.000	0.838
TSPS-AUPR	1.000	0.902