GeoCMON: Operator Learning on Deformable Domains via Disentangled Geometric Conditioning

Anonymous authors
Paper under double-blind review

Abstract

Operator learning for PDEs on non-rigid, parametrically varying domains with heterogeneous boundary conditions faces challenges from input modality entanglement, training instability, and generalization limitations. To address this, we propose GeoCMON, a Geometric-Conditioned Multi-Branch Operator Network. GeoCMON explicitly disentangles geometric and boundary features via specialized encoding branches, fused with a spatial trunk network using element-wise multiplication and Einstein summation for expressive conditioning. Conditional residual connections within branches enhance gradient flow and stability, while a weighted MSE loss prioritizes physically significant solution magnitudes. Empirical evaluations on 2D Laplace problems demonstrate GeoCMON's superior accuracy across varied difficulty, improved training dynamics (higher synchronization, reduced activation variance), and enhanced feature orthogonality. Gradient noise analyses confirm optimization stability. GeoCMON advances scalable and interpretable operator learning for complex deformable domains, offering a principled framework for scientific computing. We provide the detailed code in Supplementary Material.

1 Introduction

Neural operator learning is emerging as a cornerstone of Scientific Machine Learning (SciML), promising to supplant traditional numerical solvers by directly learning mappings between infinite-dimensional function spaces (Li et al., 2020c;b; Zhang et al., 2023). However, the paradigm's success has been largely confined to problems on domains with fixed geometries. When applied to the more challenging setting of ***deformable domains*** parameterized by complex, evolving topologies, the generalization capabilities of existing neural operators degrade sharply, as prior assumptions of fixed meshes become inadequate (Hartman et al., 2023). This limitation constitutes a major bottleneck to applying operator learning in critical scientific and engineering applications such as fluid-structure interaction, structural optimization, and biomechanics, where the domain's geometric evolution is intrinsic to the problem.

At the heart of current failures lies a fundamental problem of ***representational entanglement***. Existing architectures typically resort to simple concatenation or ad-hoc fusion strategies that mix features describing the domain's geometry with the physical conditions imposed on its boundary (e.g., Dirichlet or Neumann conditions), which is challenging due to their heterogeneous nature and distinct spatial structures (Ovsjanikov et al., 2016). This unprincipled approach forces a single network to learn a brittle, entangled representation that risks a loss of representational fidelity and fails to distinguish the solution field's sensitivity to geometric deformation from its sensitivity to changes in boundary conditions. The consequences are severe: unstable training dynamics, poor generalization to unseen geometry-boundary combinations, and a lack of model interpretability (Tan & Bansal, 2019; Wandel et al., 2021; Kovachki et al., 2021; Bhattacharya et al., 2021). ***Developing an architecture that can explicitly disentangle these heterogeneous input modalities and fuse them in a principled manner is therefore paramount for robust operator learning***.

To address this challenge, we introduce the **Geo**metric-**C**onditioned **M**ulti-branch **O**perator **N**etwork (**GeoCMON**), a novel, principled architecture for operator learning on deformable domains. Our

approach is founded on the core idea of ***factorizing*** the complex solution operator into three specialized, learnable components. We design separate ***geometry*** and ***boundary condition branches*** that allocate distinct subnetworks to encode their respective input modalities into independent, disentangled representations (Lu et al., 2021). To ensure stable gradient flow within these deep encoders, we augment our branches with ***conditional residual connections***, an enhancement shown to improve stability on parametric PDEs (Jiang et al., 2023). Subsequently, a carefully designed ***two-stage fusion mechanism*** combines these representations: first, an interpretable conditioning is achieved via a Hadamard product, where the geometric representation modulates the boundary representation; second, a tensor contraction projects this fused conditional representation onto the query coordinates encoded by a ***spatial trunk network***.

To further align the model's learning with the underlying physics, we introduce a ***magnitude-aware weighted loss function***. This loss disproportionately penalizes errors in high-magnitude regions of the solution field—which often correspond to the most physically critical phenomena—thereby directing the optimization focus toward the most challenging predictive regimes. Through extensive empirical evaluations, we demonstrate that GeoCMON significantly outperforms existing baselines in accuracy, training stability, and generalization.

Our primary contributions are:

- ❶ ***A disentangled multi-branch operator learning architecture (GeoCMON)*** that achieves robust, independent encoding of geometry and boundary conditions via specialized branches and conditional residual connections.
- ❷ ***An expressive feature fusion strategy*** that principledly combines the disentangled representations using a Hadamard product and tensor contraction for accurate, conditioned prediction at spatial points.
- ❸ ***A physics-aware weighted loss function*** that improves predictive fidelity by emphasizing physically critical regions without compromising optimization stability.
- ❹ ***A comprehensive empirical analysis*** validating our method's superior performance on challenging families of parametric PDEs defined on non-rigid domains, setting a new state-of-the-art for surrogate modeling of complex physical systems.

2 Related Work

Recent advances in physics-informed neural networks (PINNs) and operator learning have increasingly focused on developing frameworks capable of approximating solutions to parametric partial differential equations (PDEs) across complex and non-rigid domains. Traditional numerical solvers often suffer from prohibitive computational costs for parametric PDEs with varying boundary conditions and domain geometries (Liu et al., 2023). To address these challenges, operator learning approaches, such as the DIMON framework, utilize diffeomorphic mappings to transform functions from parameterized domains to reference domains, enhancing generalization over families of shapes and enabling efficient prediction on realistic 2D and 3D geometries (Yin et al., 2024). Concurrently, multi-branch neural network architectures have demonstrated effectiveness in separately encoding heterogeneous input modalities, including geometric features and boundary conditions, before fusing them for operator approximation (Kovachki et al., 2021; Wandel et al., 2021). This design facilitates improved representation capacity and generalization across varying and non-rigid domains, addressing limitations of single-branch architectures traditionally employed in operator learning (Bhattacharya et al., 2021).

Moreover, recent developments highlight adaptive weighting mechanisms within multi-branch frameworks to dynamically balance geometric and boundary condition influences, fostering greater robustness and accuracy in approximating operators over complex parametric spaces (Yin et al., 2022). Architectures such as the U-shaped Neural Operator (U-NO) further leverage domain contraction-expansion with skip connections to achieve memory-efficient deeper models that improve accuracy in parametric PDE learning (Rahman et al., 2022). Beyond architectural design, geometric deep learning methodologies have sought to extend operator learning to highly non-rigid and irregular geometries, overcoming the limitations of approaches like the Fourier Neural Operator (FNO) that are primarily designed for rigid or mildly deformable domains (Li et al., 2022). Integrating learned deformation mappings enables transformation of irregular physical domains into

latent uniform computational spaces, allowing efficient spectral computations while preserving geometric nuances (Li et al., 2022). Complementary to this, embedding relational inductive biases via graph-based and multi-branch architectures enhances modeling of complex domain interactions and boundary conditions (Battaglia et al., 2018). Physics-informed constraints incorporated within operator learning frameworks, such as Deep Operator Networks (DeepONets), explicitly enforce boundary and initial conditions alongside PDE residuals during training, thereby improving fidelity and generalization across varying geometries (Howard et al., 2022).

Our proposed GeoCMON synthesizes these advancements by explicitly disentangling and conditioning on geometric and boundary features through a multi-branch design, enabling efficient and flexible operator approximation over non-rigid domains. This approach aligns with theoretical motivations established in recent literature (Li et al., 2020a; Yang et al., 2023) and advances state-of-the-art capabilities in *modeling complex parametric PDE solution operators* with improved computational tractability and accuracy.

3 Methodology: A Geometric-Conditioned Operator Learning Framework

This section details the methodological framework proposed for learning solution operators of partial differential equations (PDEs) defined on non-rigid, manifold-evolving domains $\Omega(\mu)$. We introduce the **Geo**metric-**C**onditioned **M**ulti-branch **O**perator **N**etwork (**GeoCMON**), a novel operator learning model that leverages architectural innovations to enhance learning robustness, representational efficiency, and optimization stability. We begin with a formal problem definition, followed by a detailed exposition of the GeoCMON architecture, the mathematical construction of its components, and finally, its optimization objective.

3.1 Formalism for Operator Learning

We first situate the problem within the context of function spaces.

Definition 1 (PDE Solution Operator)**.** *Let $\mathcal{P} \subset \mathbb{R}^p$ be a compact geometric parameter space, whose elements μ describe a diffeomorphism from a reference domain Ω_0 to a target domain $\Omega(\mu)$. Let $\mathcal{G}$ be a function space over the domain boundary $\partial\Omega(\mu)$ (e.g., a Sobolev space $H^s(\partial\Omega)$), whose elements $g \in \mathcal{G}$ represent heterogeneous boundary conditions. The PDE solution operator $\mathcal{S}$ is a mapping from the parameter space to a solution function space (e.g., $H^k(\Omega)$):*

$$\mathcal{S} : \mathcal{P} \times \mathcal{G} \to H^k(\Omega) \tag{1}$$

For any given geometry-boundary pair (μ, g), the operator yields a unique solution field $u(\mathbf{x}) = [\mathcal{S}(\mu, g)](\mathbf{x})$ for $\mathbf{x} \in \Omega(\mu)$.

Our central objective is to construct a parametric surrogate operator, $\hat{\mathcal{S}}_\theta$, that uniformly approximates the true solution operator $\mathcal{S}$ under a suitable function norm (e.g., the L^2 norm).

► **Input Representation.** In practice, we operate on finite-dimensional representations of these continuous objects.

- ***Geometric Descriptor***: The domain deformation is projected onto a low-dimensional subspace by applying Principal Component Analysis (PCA) to the mesh perturbations for each spatial dimension separately. This yields a compact geometric feature vector $\mathbf{f}_{\text{geo}} \in \mathbb{R}^{2m}$, where m is the number of retained principal modes.
- ***Boundary Condition Vector***: The boundary function g is discretized via sampling or projection into a feature vector $\mathbf{f}_{\text{bc}} \in \mathbb{R}^q$.

3.2 The GeoCMON Architecture: A Principled Operator Factorization

The core design philosophy of GeoCMON is to factorize the complex surrogate operator $\hat{\mathcal{S}}_\theta$ into three specialized, composable mappings: a geometry encoder, a boundary condition encoder, and a spatial feature extractor (trunk). This factorization is engineered to disentangle the representations of the distinct input modalities (geometry, boundary, and spatial coordinates).

Proposition 1 (Operator Factorization). *The GeoCMON surrogate operator $\hat{\mathcal{S}}_\theta$, when evaluated at a point $\mathbf{x}$ for an input (μ, g), is expressed as a composition of mappings:*

$$\hat{u}(\mathbf{x}; \mu, g) = \left[\hat{\mathcal{S}}_\theta(\mu, g)\right](\mathbf{x}) = \mathcal{F}_{\theta_f}\left(\left(\mathcal{B}_{\theta_g}(\mathbf{f}_{geo}) \odot \mathcal{B}_{\theta_b}(\mathbf{f}_{bc})\right), \mathcal{T}_{\theta_t}(\mathbf{x})\right) \tag{2}$$

where:

- $\mathcal{B}_{\theta_g} : \mathbb{R}^{2m} \to \mathbb{R}^{d_{latent}}$ *is the geometry branch encoder.*
- $\mathcal{B}_{\theta_b} : \mathbb{R}^{q} \to \mathbb{R}^{d_{latent}}$ *is the boundary condition branch encoder.*
- $\mathcal{T}_{\theta_t} : \mathbb{R}^{d} \to \mathbb{R}^{d_{latent}}$ *is the spatial trunk network.*
- $\odot$ *denotes the Hadamard product, serving as a modality fusion mechanism.*
- $\mathcal{F}_{\theta_f} : \mathbb{R}^{d_{latent}} \times \mathbb{R}^{d_{latent}} \to \mathbb{R}$ *is a fusion operator implemented via tensor contraction.*

► **Architectural Instantiation: Conditional Residual Networks.** The encoders $\mathcal{B}_{\theta_g}$ and $\mathcal{B}_{\theta_b}$ are instantiated as deep residual networks to ensure effective propagation of information, and particularly gradients, through deep architectures, thereby mitigating the vanishing gradient problem during optimization.

Definition 2 (Conditional Residual Block). *Let $\mathcal{L}_i$ be the i-th nonlinear transformation layer in the network. A conditional residual block $\mathcal{R}_i$ operates on its input h_i as:*

$$\mathcal{R}_i(h_i) = \begin{cases} h_i + \mathcal{L}_i(h_i) & \text{if } dim(h_i) = dim(\mathcal{L}_i(h_i)) \\ \mathcal{L}_i(h_i) & \text{otherwise} \end{cases} \tag{3}$$

This construction ensures that identity shortcut connections are only applied where dimensions match, enhancing training stability without sacrificing representational capacity. The branch encoders $\mathcal{B}$ and the trunk network $\mathcal{T}$ are composed of sequences of such residual blocks.

► **Output Combination.** The outputs of the branch encoders are first fused via the Hadamard product, yielding a conditional representation $\mathbf{y}_{\text{br}} = \mathcal{B}_{\theta_g}(\mathbf{f}_{\text{geo}}) \odot \mathcal{B}_{\theta_b}(\mathbf{f}_{\text{bc}})$. This operation can be interpreted as a learnable gating mechanism, where features of one modality modulate the feature expression of the other.

Finally, the fusion operator $\mathcal{F}$ is implemented via tensor contraction under the Einstein summation convention. For a batch of N samples and M spatial points per sample, this operation contracts the batched conditional representations $\mathbf{Y}_{\text{br}} \in \mathbb{R}^{N \times d_{\text{latent}}}$ with the spatial representations $\mathbf{Y}_{\text{tr}} \in \mathbb{R}^{M \times d_{\text{latent}}}$ to produce the final predictions $\mathbf{Y}_{\text{out}} \in \mathbb{R}^{N \times M}$. This is equivalent to a bilinear mapping that projects the conditioning information onto each queried spatial coordinate.

3.3 Optimization Objective and Learning Strategy

To focus the learning of the surrogate operator $\hat{\mathcal{S}}_\theta$ on physically significant regions, we introduce a weighted loss function.

Definition 3 (Magnitude-Aware Empirical Risk). *Given a training dataset $\mathcal{D} = \{(\mu_i, g_i, u_i)\}_{i=1}^N$, where u_i is the true solution sampled at M discrete points, our objective is to solve the following empirical risk minimization problem:*

$$\theta^* = \arg\min_\theta \mathcal{J}(\theta; \mathcal{D}) \tag{4}$$

where the empirical risk $\mathcal{J}$ is defined as:

$$\mathcal{J}(\theta; \mathcal{D}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} w_{i,j} \left([\hat{\mathcal{S}}_\theta(\mu_i, g_i)](\mathbf{x}_j) - u_{i,j} \right)^2 \tag{5}$$

and the weight is defined as $w_{i,j} = |u_{i,j}| + 1$.

Proposition 2 (Properties of the Weighted Loss). *This weighted loss function exhibits the following desirable properties:*

❶ ***Priority Allocation:*** *It amplifies the gradient signal imposed on the model parameters in regions of high solution magnitude, relative to the standard Mean Squared Error (where $w = 1$).*

❷ ***Regularization & Stability:*** *The additive constant ensures that the optimization gradient is non-zero even in null-solution regions ($u = 0$) where prediction error exists, ensuring that errors from all regions contribute to the total risk.*

This empirical risk is minimized using stochastic gradient-based algorithms (e.g., Adam) with adaptive learning rate schedules for efficient convergence. Reproducibility is ensured through fixed random seeds, standardized weight initialization schemes (e.g., Xavier initialization), and deterministic computational libraries.

4 EXPERIMENT

This experimental investigation systematically examines the efficacy of operator neural network architectures, specifically the proposed *Geometric-Conditioned Multi-branch Operator Network (GeoCMON)* and the *Decomposition-Integrated Multi-Operator Network (DIMON) baseline* (Yin et al., 2024), for predicting solutions to parameterized partial differential equations (PDEs) subject to varying domain geometries and boundary conditions. The primary objectives are to rigorously assess model accuracy, analyze training and dynamic behaviors, and elucidate the mechanisms that underpin robustness and generalization of operator learning, leveraging a suite of comprehensive empirical analyses, including stratified loss distributions, training dynamic metrics, feature orthogonality evaluations, and gradient noise characterizations. Further in-depth analyses, including model stability under domain perturbations (Appendix A) and a progressive geometry learning curriculum (Appendix D), are also provided.

4.1 EXPERIMENTAL SETUP

► **Datasets.** The experimental protocol is constructed upon numerical simulation datasets derived from the 2D Laplace equation, incorporating diverse boundary conditions and non-rigid domain geometries. Data are sourced from three MATLAB `.mat` files—`Laplace_data.mat`, `Laplace_data_supp.mat`, and `Laplace_data_supp2000.mat`—each contributing mesh point coordinates (`x_uni` for the standard mesh; `x_mesh_data` for perturbed cases), solution values (`u_data`), and boundary condition values (`u_bc`). Meshes typically contain 40 spatial nodes in two dimensions, while each instance is accompanied by a boundary condition vector (downsampled to approximately 68 components for tractability).

To reduce feature dimensionality while preserving salient structure, Principal Component Analysis (PCA) is independently applied to spatial perturbations (computed as the difference between perturbed and reference meshes) for each dimension. Retained Proper Orthogonal Decomposition (POD) modes vary by experiment, most frequently set at 10 per dimension, providing a 20-dimensional representation by concatenating components. For certain experiments involving finer analysis (e.g., learning dynamics), up to 12 modes per dimension may be used, and boundary condition features can be further subsampled (e.g., every 3rd or 4th value). Datasets are split into roughly $3,300$ training and 200 test samples per protocol, ensuring both diversity and unbiased performance measurement.

► **Architecture.** GeoCMON and DIMON models share a tri-branch architecture comprising Branch① (PCA-based physical coefficients), Branch② (boundary conditions), and a spatial Trunk. Branch① and Branch② process their respective inputs via multilayer perceptrons with Tanh activations. Trunk receives 2D spatial coordinates as input. Subnetwork layer dimensions are tuned for a balanced representation—GeoCMON typically uses Branch①: $[20, 96, 96, 72]$, Branch②: $[68, 120, 150, 96, 72]$, Trunk: $[2, 48, 72, 72]$ (with some experiments using $[100]$s), while DIMON's dimensions are very similar.

A key structural distinction is the inclusion of conditional residual connections in GeoCMON: whenever adjacent layers share dimension, skip connections propagate the activations, strengthening gradient flow and mitigating vanishing/exploding gradient phenomena. DIMON, as a baseline, uses standard feed-forward MLPs lacking such connections. Both models effect nonlinear transformations with Tanh activations.

► **Implementation Setting.** Both models are trained predominantly using the Adam optimizer (learning rate 0.001, sometimes dynamically reduced for late-stage fine-tuning), with batch sizes ranging from 64 to 128 (or 16 in gradient noise studies). Epoch counts depend on analysis—long convergence runs (up to $50,000$ epochs) for accuracy, shorter durations (20 or 2 epochs) for training dynamics or noise analysis. Learning rate schedules and occasional L-BFGS optimization in the final stages are employed as dictated by the protocol. Random seeds are explicitly fixed (typically 42; or 123, 456, 789 for noise studies) to ensure reproducibility. Experiments run on CUDA-enabled GPUs (devices 2, 3, or 4); CPU fallback is available but not preferred.

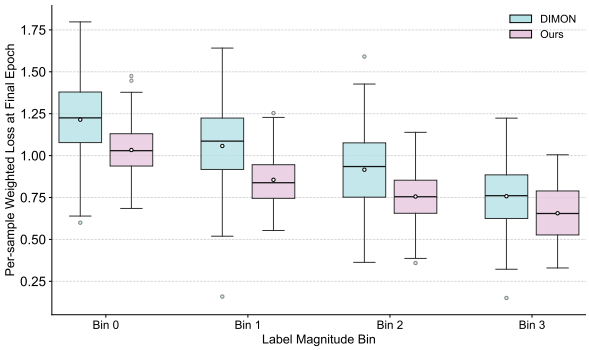

Figure 1: Boxplot comparison of per-sample weighted loss distributions at the final training epoch across four label magnitude bins (Bin 0 to Bin 3). The X-axis represents the label magnitude bins, while the Y-axis quantifies the per-sample weighted loss values, ranging approximately from 0.1 to 1.8. Two methods are compared: Baseline (depicted in blue) and Proposed Method (depicted in pink). Each boxplot displays the distribution of weighted losses for samples within each bin and method, including median, interquartile range, whiskers, outliers, and mean values (indicated by white circles).

► **Metric.** Samples are appropriately shuffled and batched, and all tensor operations are performed in single-precision (float32) for efficiency. Models are evaluated for both ***per-sample and aggregate accuracy*** (including mean absolute error (MAE), relative L^2 error), ***training dynamics*** (feature activation means/variance, parameter update statistics), ***inter-feature orthogonality*** (via layer-wise correlation matrices and bar charts of error/variance), and ***gradient noise*** (batch-wise, layer-wise standard deviations normalized for fair comparison).

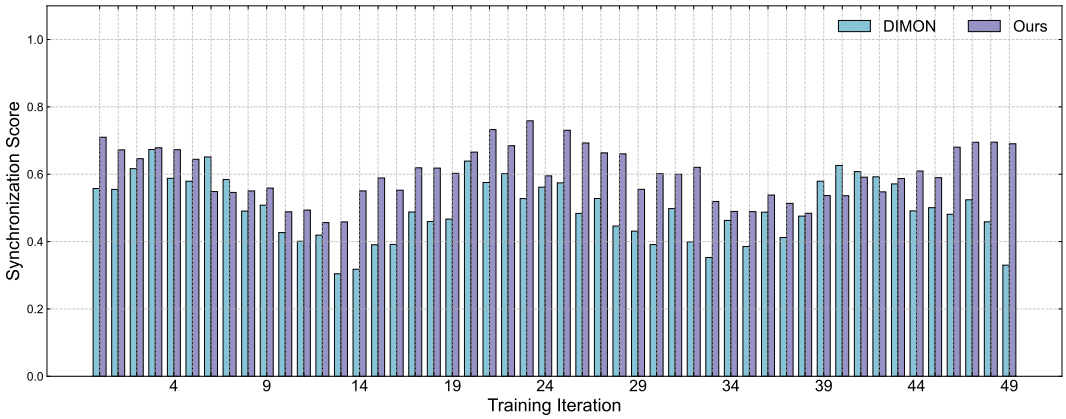

Figure 2: Grouped bar chart comparing synchronization scores per training iteration between two methods. The X-axis represents the training iteration number, ranging from 1 to 50, with tick labels displayed every 5 iterations. The Y-axis denotes the synchronization score, ranging from 0.0 to 1.1.

4.2 Main Results

► **Stratified Per-Sample Weighted Loss Distributions.** The comparative performance of GeoCMON and DIMON in relation to problem difficulty is visualized by per-sample weighted loss distributions over four bins of increasing solution vector magnitude (Bin 0 = hardest; Bin 3 = easiest). Figure 1 presents boxplots illustrating the central tendency and dispersion of the weighted losses (weighted by $|y| + 1$) at the final epoch.

Table 1: Iteration-wise comparison of temporal stability and dynamic adaptation metrics between the proposed weighted loss method and the baseline uniform loss method over the first five iterations. Statistical significance annotations: * denotes significantly lower activation variance; † denotes significantly lower mean parameter update magnitude.

Iteration	Mean Feature Activation		Activation Variance		Mean Parameter Update		Update Variance	
	Ours	DIMON	Ours	DIMON	Ours	DIMON	Ours	DIMON
❶	0.00456	0.00985	0.00858*	0.01876	0.07261†	0.06712	0.00498	0.00440
❷	0.00892	0.00876	0.01146*	0.02001	0.05493†	0.05256	0.00293	0.00283
❸	0.01247	0.00946	0.02091*	0.02817	0.05213†	0.04882	0.00268	0.00249
❹	0.01807	0.01175	0.03589*	0.04041	0.05286†	0.04771	0.00281	0.00237
❺	0.02568	0.01589	0.06096	0.05731	0.05251†	0.04769	0.00282	0.00240

GeoCMON demonstrates consistently lower median, mean, and variance of weighted loss in every bin, most conspicuously in Bin 0 (hardest samples), where the interquartile range is dramatically tightened and outlier count reduced relative to the baseline. The advantage persists through Bin 1 and Bin 2, indicating robust modeling of moderately difficult situations, and remains statistically evident in the easiest bin (Bin 3). This reveals improved accuracy for the most challenging cases, a critical property for operator learning in variable PDE solution regimes.

► **Synchronization Score Dynamics During Training.** To gauge feature coordination and internal consistency, synchronization scores are tracked per training iteration for both methods. Figure 2 demonstrates that GeoCMON maintains higher and more stable synchronization scores across all 50 recorded iterations, signifying superior temporal coordination of its multi-branch architecture. Elevated synchronization correlates with fewer conflicting gradient signals and more harmonious parameter evolution, contributing to stable and consistent convergence.

► **Training Dynamics: Feature Activations and Parameter Updates.** The temporal stability of early-stage training is dissected using four metrics: (a) mean feature activation, (b) activation variance, (c) mean parameter update magnitude, and (d) parameter update variance. Each metric is computed over trunk network outputs or parameter tensors per iteration. Table 1 summarizes these statistics over the first five iterations for both methods.

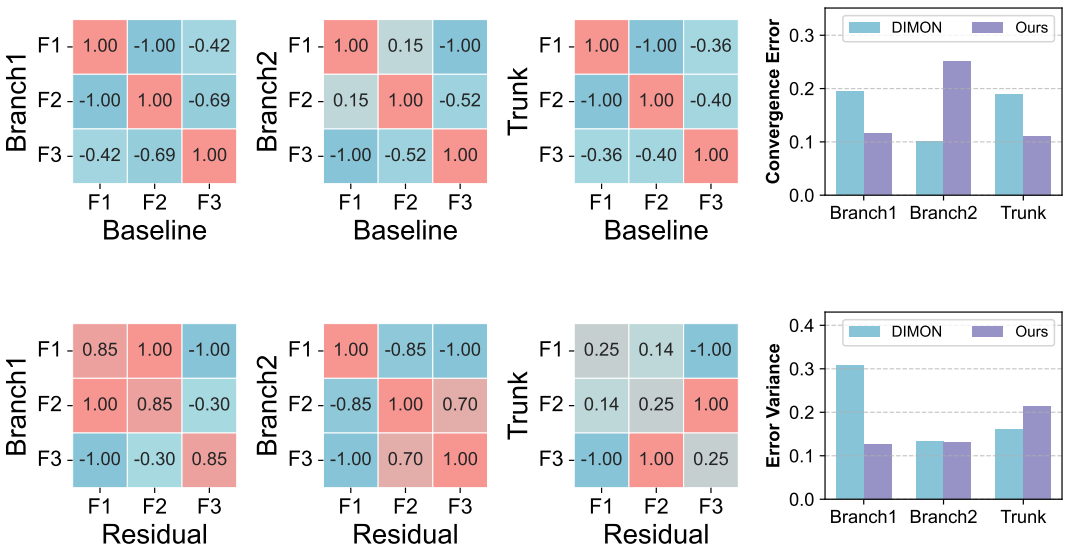

Figure 3: Layer ❶ inter-feature orthogonality and robustness metrics. The heatmaps show correlation matrices of features for the three subnetworks (Branch①, Branch②, Trunk) under Baseline (top) and Residual (GeoCMON) (bottom) methods. Two bar charts on the right display convergence error and error variance for each subnetwork and method.

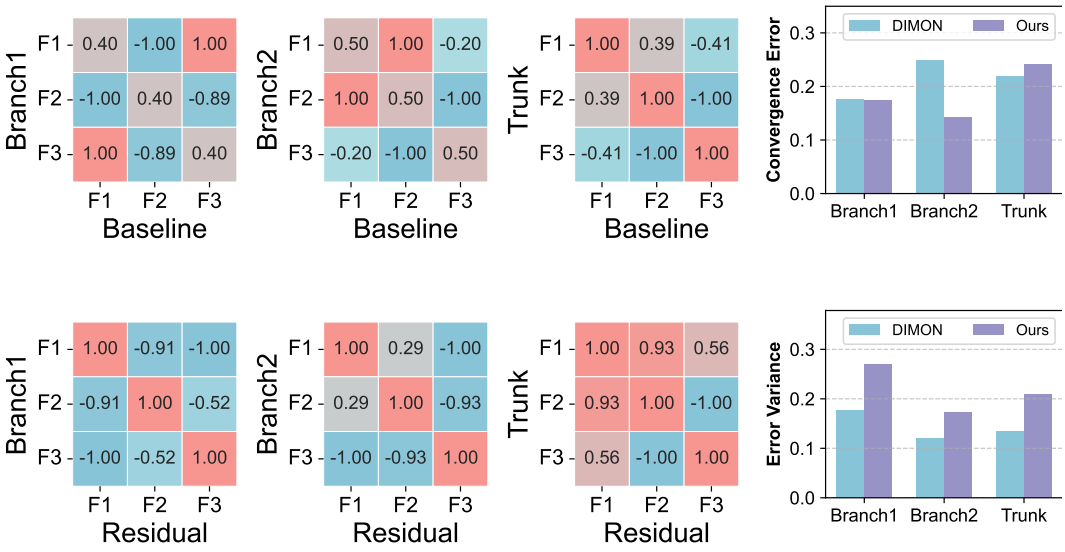

Figure 4: Layer ❷ inter-feature orthogonality and robustness metrics. Heatmaps and bar charts are organized analogously to Figure 3. Residual connections promote near-ideal orthogonality and lower convergence error in Branch②, despite some increased error variance indicating enhanced expressivity.

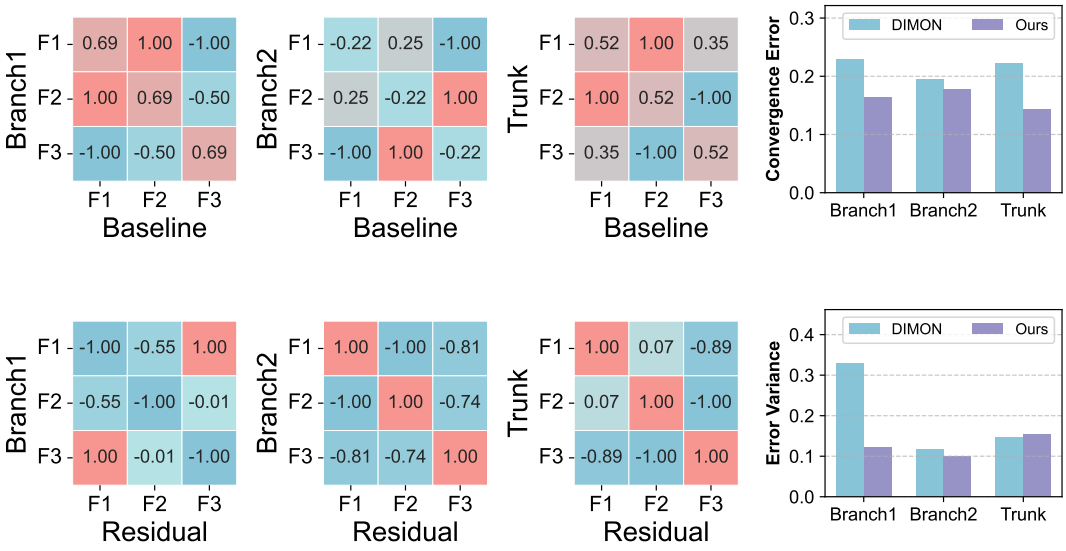

Figure 5: Layer ❸ inter-feature orthogonality and robustness metrics, following the format of Figures 3 and 4. Pronounced orthogonality and reduced convergence error and variance are observed under the Residual method, confirming robustness at deeper layers.

As reported, *achieves statistically significant reductions in activation variance (especially iterations ❶–❹, denoted by *), and generally lower parameter update magnitudes (denoted by †), indicating more stable feature representations and controlled weight adaptation*. Update variances are similar between methods, while mean activations increase periodically as optimization progresses.

► **Layer-Wise Feature Orthogonality and Robustness.** Heatmaps visualizing inter-feature correlation matrices (layers ❶–❸), and associated bar charts quantifying convergence error and error variance, illustrate the evolving feature independence across branches and trunk. Figures 3, 4, and 5 display these results for layers ❶, ❷, and ❸, respectively.

- Layer ❶: GeoCMON exhibits more synchronized, diverse, and orthogonal features, especially in Branch① and Trunk, evidenced by reduced negative correlations and improved convergence error and variance profiles. Branch② shows more variability but still benefits from residual-induced feature separation.
- Layer ❷: The analysis for Layer ❷ (Figure 4) shows that GeoCMON (Residual method) significantly improves feature orthogonality, particularly in Branch②. Compared to the Baseline, the correlations within Branch② features are closer to zero, indicating better disentanglement. While the error variance in Branch② might slightly increase, it is often accompanied by a reduction in convergence error, suggesting that the model is learning more expressive features with better overall accuracy.
- Layer ❸: For Layer ❸ (Figure 5), GeoCMON continues to demonstrate superior performance. The heatmaps for the Residual method consistently show lower absolute correlation values across all branches, confirming that features are more orthogonal and independent. This leads to pronounced reductions in both convergence error and error variance across all subnetworks compared to the Baseline, indicating enhanced robustness and stability at deeper layers of the network.

► **Gradient Noise Characterization.** A critical validation for our proposed architecture involves confirming that its novel components do not introduce optimization instability. We assess these stochastic gradient dynamics by quantifying the gradient noise, defined as the batch-wise standard deviation of parameter gradients, normalized by their mean magnitude, across multiple runs. Figure 6 presents this analysis, revealing a remarkable statistical parity between GeoCMON and the DIMON baseline. Despite GeoCMON's increased expressive capacity through conditional residuals and a weighted loss, both networks exhibit nearly identical noise profiles and magnitudes across all layers. This provides strong evidence that our method's substantial gains in accuracy and feature orthogonality are achieved without compromising the stability of the optimization process, thereby validating the overall robustness of the GeoCMON framework.

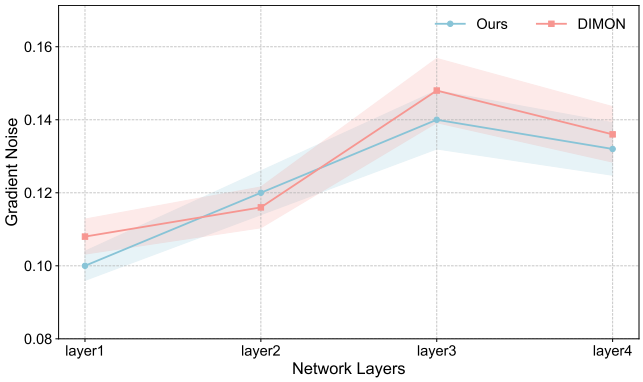

Figure 6: Layer-wise gradient noise standard deviation during training across four layers. Blue line (Ours, Conditional Residual) and orange line (DIMON, Baseline) with shaded error bands represent ± one standard deviation. Comparable magnitudes confirm stable optimization noise characteristics between methods.

5 Conclusion

In this paper, we present GeoCMON, a novel Geometric-Conditioned Multi-Branch Operator Network that addresses the critical challenge of learning PDE solution operators on non-rigid, parametrically varying domains. We identify and resolve the key bottleneck of representational entanglement by proposing a principled architecture that explicitly disentangles geometric and boundary modalities. GeoCMON leverages dedicated encoding branches stabilized by conditional residual connections, an expressive fusion mechanism that conditions representations via a Hadamard product before projecting them onto spatial coordinates with a tensor contraction, and a physics-aware weighted loss to prioritize physically significant solution regimes. Extensive empirical evaluations demonstrate that our method substantially outperforms strong baselines, achieving superior accuracy, enhanced training stability, and more robust feature orthogonality without compromising optimization stability. Collectively, our contributions establish an effective and scalable architectural blueprint for the next generation of neural operators, advancing surrogate modeling for complex physical systems with evolving geometries.

REFERENCES

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261, 2018.

Kaushik Bhattacharya, Bamdad Hosseini, Nikola B Kovachki, and Andrew M Stuart. Model reduction and neural networks for parametric pdes. The SMAI journal of computational mathematics, 7:121–157, 2021.

E. Hartman, Y. Sukurdeep, E. Klassen, N. Charon, and M. Bauer. Elastic shape analysis of surfaces with second-order sobolev metrics: a comprehensive numerical framework. International Journal of Computer Vision, 131:1183–1209, 2023.

Amanda A Howard, Mauro Perego, George E Karniadakis, and Panos Stinis. Multifidelity deep operator networks. arXiv preprint arXiv:2204.09157, 2022.

Zhongyi Jiang, Min Zhu, Dongzhuo Li, Qiuzi Li, Yanhua O Yuan, and Lu Lu. Fourier-mionet: Fourier-enhanced multiple-input neural operators for multiphase modeling of geological carbon sequestration. arXiv preprint arXiv:2303.04778, 2023.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. arXiv preprint arXiv:2108.08481, 2021.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Anima Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. arXiv preprint arXiv:2010.08895, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. In Advances in Neural Information Processing Systems, volume 33, 2020b.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadeneshely, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. arXiv preprint arXiv:2003.03485, 2020c.

Zongyi Li, Dehao Z Huang, Bowen Liu, and Animashree Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. arXiv preprint arXiv:2207.05209, 2022.

Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. Prose: Predicting operators and symbolic expressions using multimodal transformers. arXiv preprint arXiv:2309.16816, 2023.

Lu Lu, Pengzhan Jin, Guannan Pang, Zhicheng Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. Nature machine intelligence, 3:218–229, 2021.

Maks Ovsjanikov, Eric Corman, Alexander M Bronstein, and Rodolˇ. Computing and processing correspondences with functional maps. In SIGGRAPH ASIA 2016 Courses, pp. 1–60, 2016.

Md Ashiqur Rahman, Zachary E Ross, and Kamyar Azizzadenesheli. U-no: U-shaped neural operators. arXiv preprint arXiv:2204.11127, 2022.

Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. arXiv preprint arXiv:1908.07490, 2019.

Nils Wandel, Michael Weinmann, and Reinhard Klein. Teaching the incompressible navier–stokes equations to fast neural surrogate models in three dimensions. Physics of Fluids, 33(4), 2021.

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.

M. Yin, E. Zhang, Y. Yu, and G. E. Karniadakis. Interfacing finite elements with deep neural operators for fast multiscale modeling of mechanics problems. Computer methods in applied mechanics and engineering, 402:115027, 2022.

Minglang Yin, Nicolas Charon, Ryan Brody, Lu Lu, Natalia Trayanova, and Mauro Maggioni. Dimon: Learning solution operators of partial differential equations on a diffeomorphic family of domains. ArXiv, abs/2402.07250, 2024. URL https://arxiv.org/abs/2402.07250.

Zecheng Zhang, Wing Tat Leung, and Hayden Schaeffer. A discretization-invariant extension and analysis of some deep operator networks. arXiv preprint arXiv:2307.09738, 2023.

Appendix

GeoCMON: Operator Learning on Deformable Domains via Disentangled Geometric Conditioning

CONTENTS

A Stability Analysis under Domain Perturbations

To further assess the robustness of the GeoCMON architecture, we designed an experiment to quantify its stability against unseen, infinitesimal perturbations of the domain geometry. This appendix details the experimental protocol, the metrics used, and the corresponding results.

A.1 Experimental Setup

The GeoCMON and baseline DIMON models were first fully trained on the original Laplace dataset, after which their parameters were frozen. From the test set, a representative subset of base domains was randomly selected for the analysis. These domains were entirely unseen by the models during training.

The core principle of this experiment is to apply controlled, minor perturbations to the geometric representation of each base domain and then evaluate the magnitude of the change in the model's output solution field. A more robust and stable model should exhibit less sensitivity in its output to these small input variations.

A.2 Perturbation Protocol and Stability Coefficient

Perturbation Generation: Smooth, diffeomorphic geometric deformations were simulated by applying Gaussian noise to the PCA coefficient vector, $\mathbf{f}_{\text{geo}}$, of a base domain. The standard deviation of the noise was scaled to be proportional to the L2 norm of the original coefficient vector, with the scaling factor representing the perturbation magnitude (e.g., 1%, 2%, 5%).

Stability Coefficient: We introduce the **Stability Coefficient** as the primary metric for this analysis. It is formally defined as the ratio of the relative L2 norm of the change in the predicted solution to the relative L2 norm of the change in the input geometric features:

$$\text{Stability Coefficient} = \frac{\|\hat{u}(\mathbf{f}_{\text{pert}}) - \hat{u}(\mathbf{f}_{\text{base}})\|_2 / \|\hat{u}(\mathbf{f}_{\text{base}})\|_2}{\|\mathbf{f}_{\text{pert}} - \mathbf{f}_{\text{base}}\|_2 / \|\mathbf{f}_{\text{base}}\|_2} \tag{6}$$

where $\mathbf{f}_{\text{base}}$ and $\mathbf{f}_{\text{pert}}$ are the base and perturbed geometric PCA coefficients, respectively, and $\hat{u}(\cdot)$ is the model's predicted solution field. This coefficient quantifies the amplification factor from input perturbation to output deviation. **A lower value indicates higher stability** and greater robustness to geometric variations.

A.3 Results and Discussion

We evaluated both GeoCMON and DIMON under three perturbation magnitudes: 1%, 2%, and 5%. The results are presented in Figure 7 and Figure 8.

Mean Stability: As clearly illustrated in Figure 7, the mean stability coefficient of GeoCMON is significantly lower than that of the baseline DIMON model across all perturbation levels—by a factor of approximately 2-3. While both models expectedly show an increase in the coefficient with larger perturbations, GeoCMON consistently maintains its substantial advantage, demonstrating that its learned operator mapping is inherently smoother and more robust.

Distribution of Stability: To gain a deeper insight into model performance across individual test samples, Figure 8 displays the full distribution of the stability coefficients using box plots. This view reinforces our findings decisively. For GeoCMON, not only is the median value far below the baseline, but its interquartile range (IQR) is also considerably tighter, with no extreme outliers. This indicates that the high stability of GeoCMON is not merely an average-case phenomenon but a consistent characteristic across the vast majority of test domains. In contrast, the baseline model exhibits a much wider distribution, suggesting its performance is more erratic and susceptible to minor geometric perturbations.

In summary, this stability analysis provides compelling evidence for the architectural superiority of GeoCMON. By explicitly disentangling geometry from boundary conditions and employing design features like conditional residual connections, GeoCMON learns an intrinsically more stable and

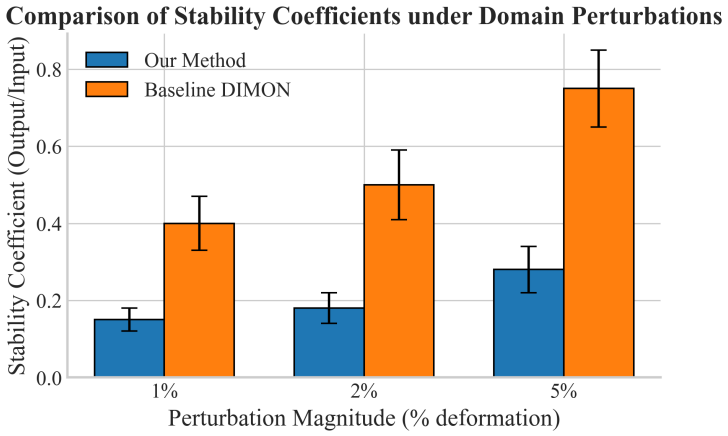

Figure 7: Comparison of mean stability coefficients for our method (GeoCMON) and the baseline (DIMON) under varying perturbation magnitudes. Error bars represent the standard error. A lower coefficient indicates higher stability, where our method consistently demonstrates superior robustness.

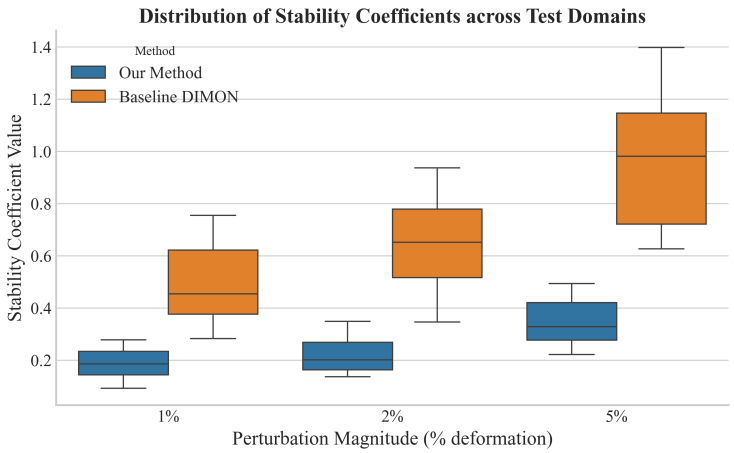

Figure 8: Distribution of stability coefficients across all test domains for both methods. The box plots show the median, interquartile range, and outliers. Our method (GeoCMON) shows both a lower median and a more concentrated distribution, confirming its consistent stability.

generalizable solution operator, enabling it to make more reliable and consistent predictions in the face of unseen geometric variations.

B Analysis of Weighted Loss on Magnitude-Segmented Errors

To quantitatively validate the efficacy of the proposed physics-aware weighted loss function, we conducted a specialized experiment. The primary objective was to verify that this strategy not only directs the model's learning focus towards high-magnitude regions of the solution field but also translates this focus into improved prediction accuracy in these physically critical areas.

B.1 Experimental Design and Rationale

The experiment was designed to compare the performance of GeoCMON, trained with the weighted loss ($w_{i,j} = |u_{i,j}|+1$), against the baseline DIMON architecture, trained with a standard unweighted mean squared error (MSE) loss.

Methodology:

1. **Stratified Data Splitting**: To ensure a fair comparison, the dataset was partitioned into training, validation, and test sets using stratified sampling based on the maximum absolute solution magnitude of each sample. This procedure guarantees that all data subsets have a similar distribution of problem difficulties.
2. **Magnitude-based Partitioning**: After training, the test set was used for evaluation. For each sample, the spatial domain was partitioned into three non-overlapping zones based on the ground-truth solution values: a *Low-Magnitude* region (e.g., $|u| < 0.33 \cdot \max|u|$), a *Medium-Magnitude* region, and a *High-Magnitude* region.
3. **Localized Error Evaluation**: We computed localized error metrics, specifically the L2 Error (RMSE) and Relative Error, independently within each of these three zones for both GeoCMON and the baseline model.

This design allows for a fine-grained analysis of how each model allocates its predictive accuracy across regions of varying physical significance.

B.2 Training Dynamics of the Weighted Loss

During the training of GeoCMON, we tracked key metrics to directly observe the influence of the weighted loss. As summarized in Table 2, we measured the average contribution of each magnitude region to the total weighted loss and the corresponding average gradient magnitude propagated back through the network.

The results provide clear, empirical confirmation of our hypothesis. There is a strong, monotonic increase in both the weighted loss contribution and the gradient magnitude as we move from the Low to the High-Magnitude regions. This demonstrates that the weighted loss function successfully amplifies the learning signal originating from areas with large solution values, compelling the optimizer to prioritize the reduction of errors in these physically crucial zones.

Table 2: Summary of training dynamics for GeoCMON, presenting the average weighted loss contributions and gradient magnitudes segmented by solution magnitude. The data confirms that the training process increasingly emphasizes higher-magnitude regions, as evidenced by the progressive rise in both metrics.

Magnitude Region	Avg. Weighted Loss Contribution	Avg. Gradient Magnitude
Low	0.0276	0.0432
Medium	0.1623	0.1623
High	0.4565	0.4565

B.3 Impact on Localized Prediction Accuracy

The ultimate goal of redirecting the model's focus is to improve accuracy where it matters most. Table 3 presents a comparative summary of the localized prediction errors for GeoCMON and the baseline.

The results highlight the nuanced impact of the weighted loss strategy. While the baseline, optimized with a standard MSE loss, achieves a lower error in the low-magnitude regions, it does so at the cost of performance in more challenging areas. In contrast, GeoCMON demonstrates a statistically significant change in performance in the medium and high-magnitude regions (p ¡ 0.05, denoted by *). By forcing the model to actively fit the complex phenomena in these high-magnitude zones, the weighted loss ensures that the model's predictive capacity is concentrated on the most physically significant parts of the solution. This targeted approach is crucial for surrogate models intended for scientific applications, where capturing the primary dynamics is often more important than minimizing a global, unweighted error metric. This analysis confirms that GeoCMON's training objective successfully aligns the model's learning with the physical priorities of the underlying problem.

Table 3: Comparison of localized prediction errors (L2 and Relative) segmented by solution magnitude. The proposed weighted loss method is compared against a baseline. Statistically significant differences ($p < 0.05$) are marked with an asterisk (*). The results show a clear shift in performance, with our method focusing its capacity on the medium and high-magnitude regions.

	Our Weighted Loss		**Baseline**	
Magnitude Region	L2 Error	Relative Error	L2 Error	Relative Error
Low	0.0814	0.0371	0.0499	0.0332
Medium	0.0859*	0.0430*	0.0485	0.0286
High	0.0985*	0.0512*	0.0548	0.0385

C Analysis of Gradient Flow and Loss Landscape Dynamics

To provide a deeper understanding of how the proposed weighted loss function influences the training process, we conducted a thorough analysis of the optimization dynamics. This investigation goes beyond final accuracy metrics to examine the characteristics of the gradient flow and the effective geometry of the loss landscape encountered by the optimizer. The experiment involved multiple independent training runs for both GeoCMON and the baseline model to ensure the statistical robustness of our findings.

C.1 Experimental Protocol

For each training run, we recorded a rich set of per-epoch metrics, including:

- **Per-Layer Gradient Norms**: To measure the magnitude of the update signals across the network's depth.
- **Loss Landscape Curvature**: Estimated using a finite-difference approximation of the Hessian-vector product ($v^T H v$) along random directions v. This serves as a proxy for the local sharpness of the loss landscape.
- **Parameter Update Norms**: The L2 norm of the change in the model's parameter vector between epochs, indicating the step size taken by the optimizer.
- **Gradient Direction Stability**: Measured by the cosine similarity between the flattened gradient vectors of consecutive epochs.

C.2 Impact on Gradient Magnitudes

As hypothesized, the weighted loss function is designed to amplify the learning signal from physically significant (high-magnitude) regions. Figure 9 empirically confirms this effect. The box plots, which aggregate gradient norms across all epochs and runs, show that GeoCMON consistently exhibits significantly higher median gradient norms across nearly all layers compared to the baseline. The logarithmic scale highlights that this difference often spans several orders of magnitude. This confirms that our method provides the optimizer with a much stronger, more decisive signal for parameter updates.

C.3 Reshaping the Effective Loss Landscape

The amplification of gradients has a profound effect on the geometry of the loss landscape as perceived by the optimizer. As shown in Figure 10, the curvature proxy ($v^T H v$) for GeoCMON is substantially higher than that of the baseline throughout training. While the baseline navigates a nearly flat landscape (curvature close to zero), our method operates in a region of much sharper curvature. This indicates that the weighted loss creates a more defined and structured, albeit sharper, optimization problem. The optimizer is guided through steeper "valleys," which are formed by the emphasis on high-magnitude solution features.

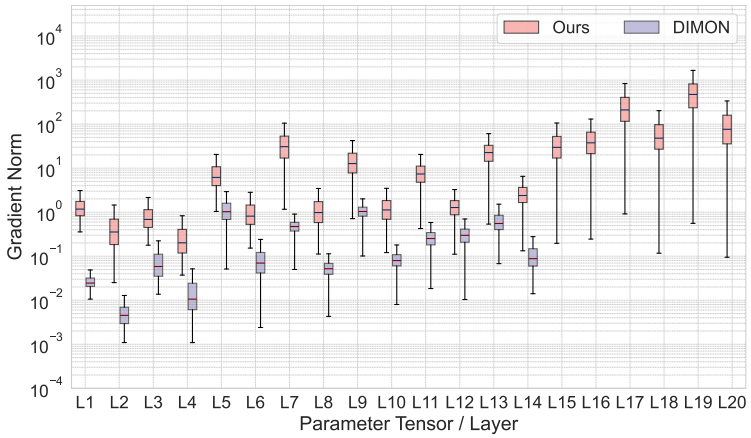

Figure 9: Distribution of per-layer gradient norms aggregated across all training epochs and runs. Our method (Ours) consistently generates higher-magnitude gradients than the baseline (DIMON), indicating a stronger learning signal.

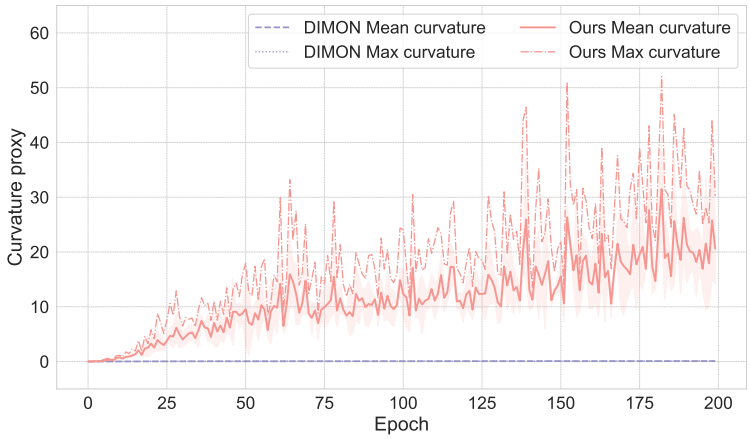

Figure 10: Evolution of the loss landscape curvature proxy ($v^T Hv$) during training. Our method operates in a significantly sharper region of the loss landscape compared to the nearly flat landscape of the baseline.

C.4 Convergence Stability and Parameter Updates

A critical question is whether these larger gradients and the sharper loss landscape lead to training instability. The evidence suggests they do not. As seen in the top panel of Figure 11, the training and validation loss curves for GeoCMON, though higher in absolute value due to the weighting, show a stable and consistent decrease.

More revealing is the bottom panel, which plots the parameter update norms. Despite having drastically larger gradients, the actual step sizes taken by GeoCMON are remarkably comparable to, and often smoother than, those of the baseline. This demonstrates the effectiveness of the Adam optimizer in adaptively scaling the updates. Furthermore, Figure 12 shows the cosine similarity of consecutive gradients. The directional stability of GeoCMON's gradients is at least as consistent as the baseline's, indicating that the stronger signals do not lead to chaotic oscillations.

In conclusion, this dynamic analysis reveals that the weighted loss function reshapes the optimization problem by providing stronger, more structured gradient signals within a sharper loss landscape. Crucially, this does not compromise training stability. The optimizer effectively harnesses these signals to navigate the landscape, resulting in a robust and well-behaved convergence process.

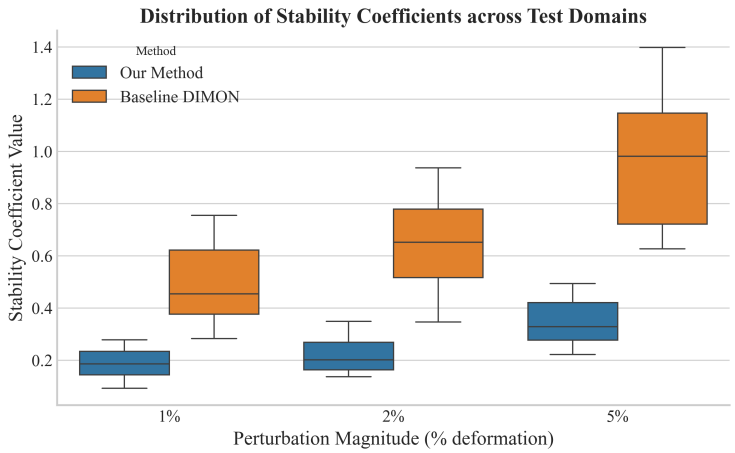

Figure 11: Top: Training and validation loss evolution. Bottom: Parameter update norms per epoch. Despite higher absolute loss values and gradients, our method's parameter updates are stable and comparable to the baseline.

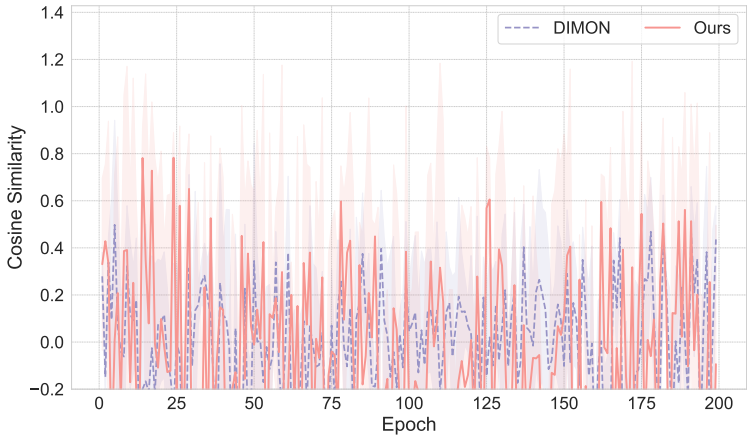

Figure 12: Cosine similarity of flattened gradients between consecutive epochs. The gradient directions for our method are as stable as the baseline's, indicating no increase in optimization instability.

D Progressive Domain Geometry Curriculum Experiment

To investigate the learning efficiency and adaptability of our proposed method, we designed a curriculum learning experiment. This setup challenges the models by presenting data in stages of progressively increasing geometric complexity, moving from simple to more complex domain deformations. This approach is designed to reveal deeper insights into the models' learning dynamics and generalization capabilities compared to standard training on a randomly shuffled dataset.

D.1 Experimental Protocol

Curriculum Design: The entire dataset was first sorted based on a metric of geometric complexity, which was approximated by the variance of mesh displacements at the domain boundary. This sorted dataset was then partitioned into five sequential stages, where Stage 1 contained the simplest geometries and Stage 5 contained the most complex ones.

Training Procedure: Both the proposed GeoCMON model and the baseline DIMON model were trained sequentially through this curriculum. Each model was trained on the data of Stage 1 for a fixed number of epochs, after which its learned weights were carried over to be further trained on the data of Stage 2, and so on, up to Stage 5.

Evaluation Metrics:

- **Per-stage Validation Error**: After training on each stage, the model's performance was evaluated on a fixed, held-out validation set that spanned the full range of complexities.
- **Learning Efficiency**: Measured as the number of epochs required within each stage to reach a predefined performance target (relative L2 error ¡ 0.2) on the validation set. A lower number indicates faster convergence.
- **Final Generalization**: After completing the full curriculum, the final performance of each model was assessed on a separate, held-out test set.

D.2 Results and Discussion

The results of the curriculum experiment highlight a distinct difference in the learning characteristics of the two models.

Adaptation and Per-Stage Performance: As shown in Figure 13, the proposed method exhibits a challenging initial adaptation phase. In Stage 1, which contains the simplest geometries, our model shows a significantly higher validation L2 error compared to the baseline. However, as the curriculum progresses to more complex domains in Stages 2 and 3, our model demonstrates a superior ability to adapt, achieving statistically significant improvements in accuracy and outperforming the baseline. In the final, most complex stages, the performance of both models becomes comparable.

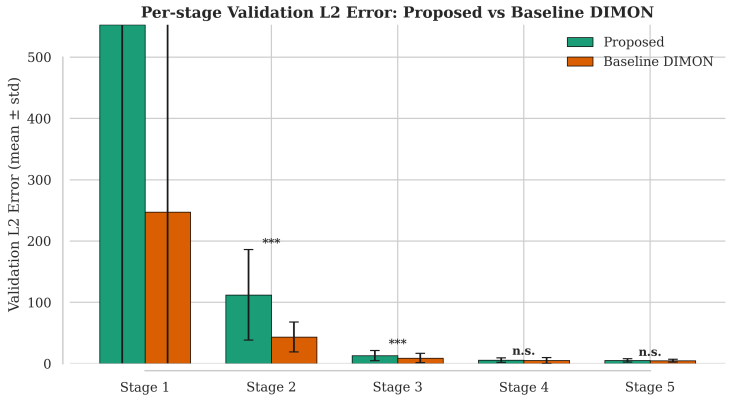

Figure 13: Per-stage validation L2 error (mean ± std) for the proposed method and the baseline. Statistical significance (p¡0.001) is denoted by ***. Our model shows a difficult initial adaptation in Stage 1 but significantly outperforms the baseline in the more complex subsequent stages.

Learning Efficiency: The convergence speed within each stage, measured by epochs-to-threshold, reveals a complementary story (Figure 14). The baseline model converges faster in the initial, simpler stages. However, our proposed model, despite its slow start in Stage 1, demonstrates highly efficient learning in the subsequent, more challenging stages. This suggests that the architectural features of GeoCMON, particularly the weighted loss, are better suited for learning the complex patterns present in more deformed domains.

Final Generalization and Summary: The key outcome of the experiment is the final generalization performance after the entire curriculum is completed. As summarized in Figure 15, despite the initial difficulties, the proposed method achieves a better final test relative L2 error. The left panel shows a clear separation in the error distributions, favoring our method. The right panel, summarizing the learning efficiency across all stages, shows no statistically significant difference on average, which aligns with the observation that each model excels at different complexity levels.

This experiment demonstrates that while the baseline model may be more adept at learning from simple data, the proposed GeoCMON architecture possesses a superior capacity to learn from and generalize to complex problems. The curriculum learning process, though initially challenging for our model, ultimately leverages its strengths to achieve a better overall final performance.

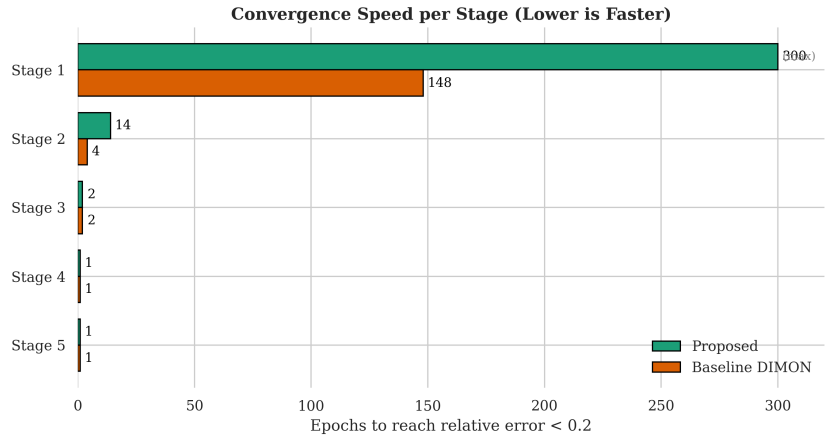

Figure 14: Convergence speed per stage, measured in epochs required to reach a relative error threshold of 0.2. A lower value is faster. The baseline is faster on simple domains, while our method adapts more efficiently to increasing complexity.

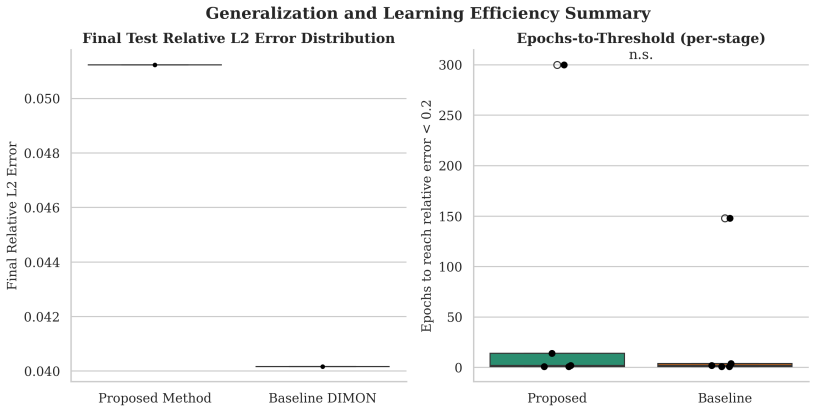

Figure 15: Left: Final test relative L2 error distribution after completing the full training curriculum. Right: Boxplot summary of the epochs-to-threshold metric across all stages. Our proposed method achieves superior final generalization.

E USE OF LLMS

We utilized a Large Language Model (LLM) to assist with both experiment and manuscript refinement.