WHERE TO COMPUTE AND HOW TO INTERACT: OPERATOR-READABLE ADAPTATION WITH GAUGE-AWARE TRANSPORT

Anonymous authors
Paper under double-blind review

ABSTRACT

Adaptive meshes enable neural operators for partial differential equations (PDEs) to allocate spatial samples and computational resources according to local physical structures. Existing approaches, however, primarily focus on where to compute, while paying less attention to how to interact after node relocation. Mesh adaptation changes local sampling scales, neighborhood structures, and geometric contexts, making representations formed at different nodes not necessarily directly comparable. Direct aggregation may therefore entangle genuine physical variation with discretization-induced representation variation. Moreover, because allocation and interaction are jointly optimized through the same output objective, their individual roles are difficult to distinguish from final errors alone. We introduce operator readability, which requires an adaptive operator to explicitly account for and test why computation is allocated to particular locations and how representations interact under the resulting nonuniform discretization. Based on this principle, we propose the Gauge-Aware Adaptive Mesh Neural Operator (GA-AMNO). Physics-informed adaptive allocation answers where to compute, while geometry-conditioned low-rank Gauge transport maps source features into target representation contexts before aggregation, answering how to interact. This design turns the otherwise implicit mesh-to-solver information exchange into an inspectable and intervenable computational process. We further establish sufficient conditions for representation-consistent aggregation and analyze approximate transport errors and continuity under topology-preserving mesh deformations. Experiments across five PDE benchmarks demonstrate improved predictive accuracy, while controlled interventions and geometric-mismatch analyses verify the computational roles of allocation and interaction and show that Gauge transport improves cross-discretization representation compatibility under strong geometric mismatch.

1 INTRODUCTION

Neural operators aim to learn solution mappings for partial differential equations (PDEs) between function spaces, while their numerical realizations operate on physical fields sampled over grids or meshes (Huang et al., 2025; Kovachki et al., 2023). Adaptive discretization makes this computational representation state dependent by relocating nodes toward dynamically informative regions (Berger & Oliger, 1984). Existing adaptive operators are therefore commonly organized around one question: where should computation be allocated? However, node relocation changes not only the spatial distribution of resolution, but also the local sampling scales, edge directions, neighborhood structures, and mesh Jacobians under which features are formed. The resulting representations must subsequently be exchanged by the operator under these unequal local geometric conditions. Adaptive computation must therefore answer a second question: how should these representations interact? These two decisions are connected through the learned geometry. Allocation determines the local discretization contexts presented to the interaction mechanism, while interaction determines whether the representations created by that allocation can be combined effectively. Making the mesh adaptive without explicitly accounting for the resulting information exchange consequently leaves an important part of adaptive computation unspecified.

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.

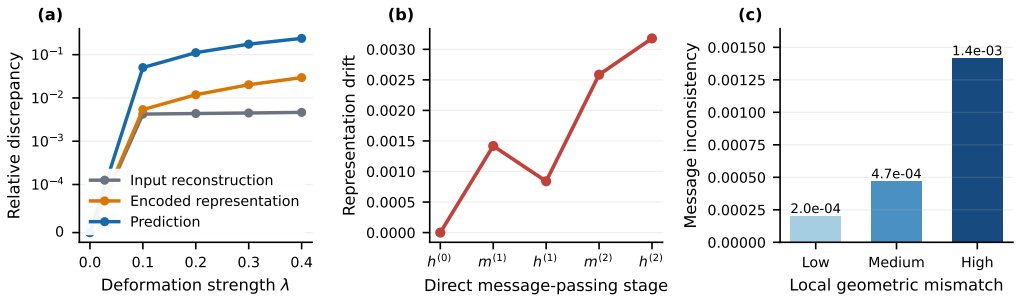

Figure 1: Preliminary diagnosis of the interaction effects induced by adaptive discretization on Navier–Stokes. (a) For the same physical state, increasing the topology-preserving deformation strength λ produces substantially larger representation and prediction discrepancies than input reconstruction discrepancy. (b) Representation drift emerges at direct aggregation and accumulates through subsequent message passing, where $\mathbf{h}^{(\ell)}$ denotes the node representation at layer ℓ and $\mathbf{m}^{(\ell)}$ the aggregated message. (c) Message inconsistency increases with local geometric mismatch. Together, these results show that direct aggregation amplifies differences induced by unequal local discretization contexts.

This missing account of information exchange is difficult to expose under standard end-to-end training, because allocation and interaction are jointly optimized through the same prediction objective. An expressive solver may compensate for an uninformative mesh, while a strong allocation mechanism may reduce the apparent need for geometry-aware interaction. Different allocation–interaction configurations may therefore achieve similar predictive performance, even though they use the adaptive geometry in substantially different ways. To determine whether this concern has observable consequences, we conduct a controlled diagnosis in Figure 1. We keep the physical state, model parameters, node identities, and neighborhood topology fixed, while progressively deforming only the sampling geometry. A small input reconstruction discrepancy develops into substantially larger representation and prediction discrepancies. Layerwise measurements further locate the onset of this drift at direct message aggregation, while message inconsistency increases with local geometric mismatch. These results show that the problem is not irregular sampling alone; it emerges when representations formed under unequal local discretization contexts are directly combined.

These observations motivate operator readability as an operational and experimentally testable computational criterion. An operator-readable adaptive model should make the roles of its adaptive decisions explicit and allow each role to be examined through controlled interventions. In this work, we use the following conditions as sufficient operational criteria for the two aspects of readability. *Allocation readability* is established when node movement is explicitly linked to declared physical quantities through a clear computational path and responds predictably to controlled modifications of the allocation signal. *Interaction readability* is established when source-to-target information transformation is explicitly conditioned on local geometric relations, can be examined at the edge level, and exhibits measurable responses to geometric mismatch and controlled interventions. Readability therefore does not refer to whether the resulting mesh is visually intuitive. It refers to whether the model can account for both why computational resolution is allocated to particular regions and how information is exchanged under the resulting discretization.

Based on this formulation, we propose the Gauge-Aware Adaptive Mesh Neural Operator (GA-AMNO) as one realization of operator-readable adaptation. As summarized in Figure 2, physics-informed indicators construct an importance map that determines where nodes are relocated, while the resulting relative coordinates, local mesh Jacobians, and state descriptors condition an edge-wise low-rank transport that determines how their features interact. We use Gauge transport as a design principle rather than assume that every discretization change induces an exact gauge transformation. By mapping each source feature into its target representation context before aggregation, it explicitly accounts for differences in local sampling scale, orientation, and neighborhood geometry, making the otherwise implicit mesh-to-solver interaction inspectable. Gauge transport therefore provides a structured realization of interaction readability, while the predictive capability is supplied by the complete GA-AMNO architecture. We further establish sufficient conditions for representation-

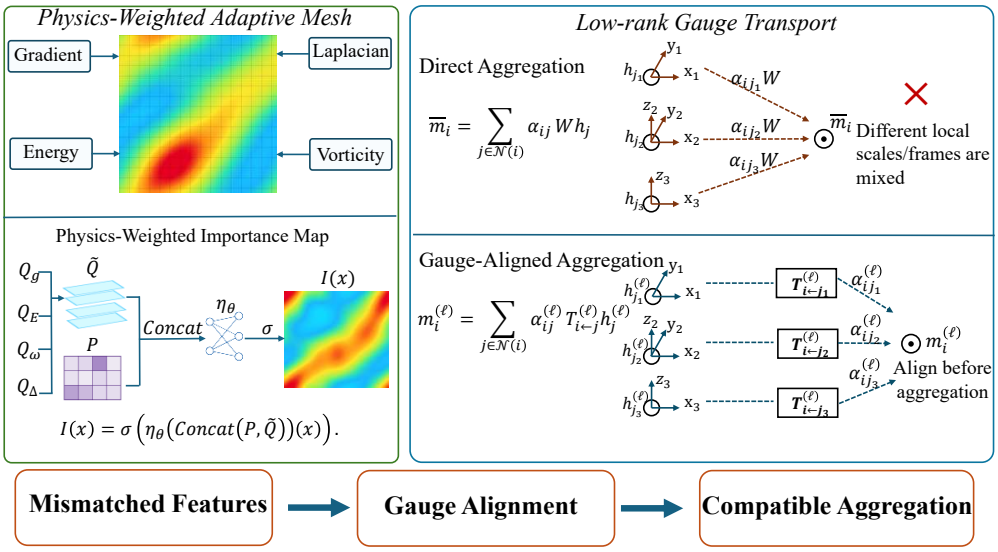

Figure 2: Conceptual overview of operator-readable adaptive computation. Physics-informed allocation explains where adaptive resolution is placed. The resulting nonuniform mesh produces unequal local discretization contexts, under which direct aggregation leaves the effects of geometric mismatch on cross-node interaction implicit. Gauge-aware transport instead maps each source feature into a target-conditioned representation before aggregation, making explicit how information is exchanged after the discretization changes.

consistent aggregation, characterize the propagation of approximate transport error, and analyze continuity under topology-preserving mesh deformation.

The main contributions are summarized as follows:

- **Conceptual contribution.** We reformulate adaptive neural operator learning as two connected problems: resource allocation and cross-discretization interaction. Based on this formulation, we introduce an operational notion of operator readability and identify, through controlled re-discretization, a reproducible failure mode in which direct aggregation amplifies discretization-induced representation drift.
- **Architectural and theoretical contribution.** We develop GA-AMNO as a concrete realization of operator-readable adaptation. Physics-informed allocation determines where computational resolution is placed, while geometry-conditioned low-rank transport makes explicit how representations formed under unequal local discretizations interact. We additionally provide consistency conditions, an approximate-transport error bound, and a continuity analysis under topology-preserving mesh deformation.
- **Evaluation contribution.** We evaluate both predictive utility and mechanism readability across multiple PDE families. Beyond standard prediction benchmarks, allocation counterfactuals, edge interventions, geometric-mismatch stratification, and controlled mesh deformations test whether the allocation and interaction mechanisms fulfill their declared computational roles.

2 Related Work

We briefly review the three lines of research most relevant to our work. A more comprehensive discussion is provided in Appendix B.

Neural operators. Neural operators learn mappings between function spaces using spectral transformations, coordinate-conditioned kernels, graph interactions, or physics-aware attention (Li et al., 2020a;b; Wu et al., 2024; Bie et al., 2025). While these methods support regular or irregular dis-

cretizations, they do not primarily examine how input-dependent mesh adaptation changes the representation contexts in which features interact.

Adaptive mesh methods. Classical and learned adaptive methods allocate spatial resolution through error indicators, local refinement, node movement, or learned policies (Berger & Oliger, 1984; Berger & Colella, 1989; Rudd et al., 2014; Pfaff et al., 2020). They mainly determine where computation should be concentrated, whereas the interaction between features formed under the resulting nonuniform local discretizations is typically delegated to the downstream solver.

Gauge-aware representations. Gauge-aware geometric networks transform features between local reference frames to support consistent information exchange (Bronstein et al., 2021; Cohen et al., 2019). Motivated by this principle, we use geometry-conditioned transport without assuming exact Gauge equivariance, making feature interaction across adaptive local discretizations explicit before aggregation.

3 METHOD

3.1 PROBLEM FORMULATION

Let $\Omega \subset \mathbb{R}^d$ be the spatial domain and let $\mathbf{u}$ denote a C-channel physical field. We consider PDE systems that can be written in the generic form

$$\mathcal{F}_{\mathrm{PDE}}\left(\mathbf{u}, \partial_t \mathbf{u}, \nabla \mathbf{u}, \nabla^2 \mathbf{u}; \boldsymbol{\mu}\right) = 0. \tag{1}$$

Here, $\mathcal{F}_{\mathrm{PDE}}$ is the governing differential operator, ∂_t is the temporal derivative, ∇ and ∇^2 denote first- and second-order spatial derivatives, and $\boldsymbol{\mu}$ collects physical parameters, coefficient fields, or forcing terms. For steady-state PDEs, the temporal derivative is omitted.

The corresponding solution operator is

$$\mathcal{G}^\dagger : \mathcal{A} \to \mathcal{U}, \qquad \mathbf{u} = \mathcal{G}^\dagger(\mathbf{a}), \tag{2}$$

where $\mathcal{A}$ is the input function space, $\mathcal{U}$ is the solution function space, $\mathbf{a}$ denotes the input condition, and $\mathcal{G}^\dagger$ is the unknown exact operator Kovachki et al. (2023). For temporal prediction, $\mathbf{a}$ contains a short history of states; for steady-state prediction, it may contain coefficients, forcing terms, or boundary information.

GA-AMNO learns an approximation

$$\widehat{\mathbf{u}} = \mathcal{G}_\theta(\mathbf{a}) \approx \mathcal{G}^\dagger(\mathbf{a}), \tag{3}$$

where $\widehat{\mathbf{u}}$ is the predicted field and $\mathcal{G}_\theta$ is the neural operator with trainable parameters θ.

The input field is first represented on a reference discretization $\mathcal{X}^r = \{\mathbf{x}_i^r\}_{i=1}^N$. GA-AMNO then constructs an input-dependent adaptive mesh

$$\mathcal{X}^a = \left\{\mathbf{x}_i^a = \Pi_\Omega\left(\mathbf{x}_i^r + \Delta\mathbf{x}_i\right)\right\}_{i=1}^N. \tag{4}$$

Here, $\mathcal{X}^a$ is the adaptive mesh, $\mathbf{x}_i^r$ and $\mathbf{x}_i^a$ are the reference and adaptive coordinates of node i, N is the number of nodes, $\Delta\mathbf{x}_i \in \mathbb{R}^d$ is the learned displacement, and Π_Ω keeps the displaced node inside the computational domain.

The adaptive coordinates define an input-dependent local discretization environment for each node. Once constructed for an input $\mathbf{a}$, the adaptive mesh $\mathcal{X}^a(\mathbf{a})$ remains fixed during the subsequent operator computation but may vary across inputs, leading to changes in local spacing, neighborhood geometry, and mesh Jacobians. Consequently, nodal features describing related physical content may be formed under different local discretization environments and need not be directly comparable. To address this mismatch, GA-AMNO does not assume an analytically specified coordinate transformation. Instead, it learns local feature-transport mappings that are conditioned on mesh geometry and input descriptors and are applied before aggregation to mitigate the resulting representation offsets.

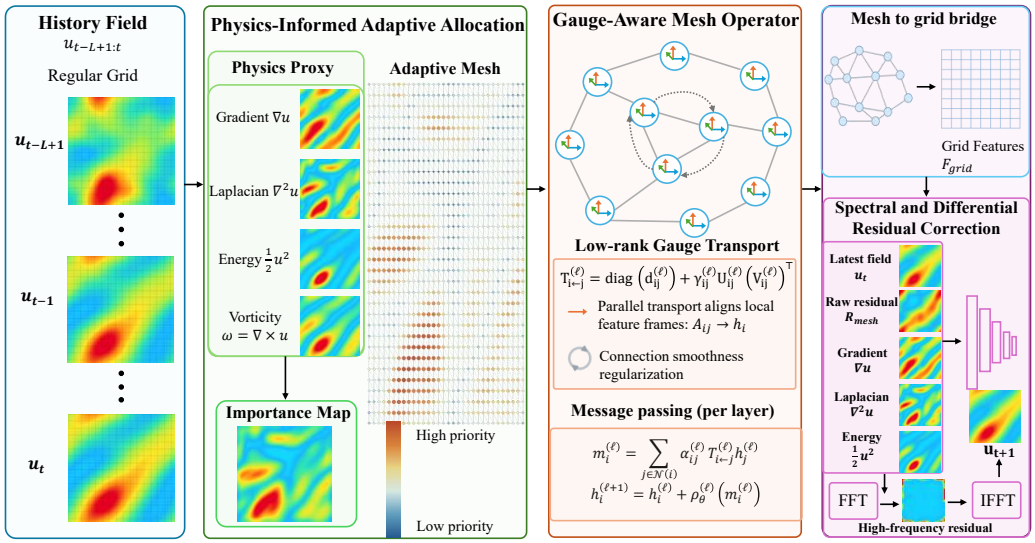

Figure 3: Overall architecture of GA-AMNO. Physics-informed indicators guide adaptive allocation, low-rank gauge transport aligns features before aggregation on the resulting mesh, and the reconstructed grid features are refined by spectral and differential residual corrections.

3.2 Overall Architecture

Given an input $\mathbf{a}$ on $\mathcal{X}^r$, GA-AMNO encodes local state descriptors and physics-informed indicators, predicts an importance map, and constructs $\mathcal{X}^a(\mathbf{a})$. Gauge-aware operator layers then use adaptive coordinates, local mesh Jacobians, and state descriptors to transport each source feature into its target context before aggregation. After the nodal updates, a mesh-to-grid bridge reconstructs the adaptive features on the reference grid, where spectral and differential residual corrections produce the final output, as summarized in Figure 3. The complete forward procedure is given in Appendix A, while the reconstruction details are provided in Section D.3.

3.3 Physics-Informed Adaptive Allocation

This module determines where representation capacity is allocated. Given the latest input slice $\mathbf{v}$, we compute four physically meaningful indicators: the gradient magnitude Q_g, Laplacian magnitude Q_Δ, local energy Q_E, and vorticity-related response Q_ω. Their complete definitions and scalar-field treatment are provided in Appendix D.1. The indicators are combined with the learned spatial state descriptor $\mathbf{P} = \mathrm{Encoder}(\mathbf{a})$ to predict the importance map:

$$\begin{aligned} \widetilde{\mathbf{Q}}(\mathbf{x}) &= \mathrm{Concat}(\mathcal{N}(Q_g), \mathcal{N}(Q_\Delta), \mathcal{N}(Q_E), \mathcal{N}(Q_\omega))\,(\mathbf{x}), \\ I(\mathbf{x}) &= \sigma\Big(\eta_\theta\Big(\mathrm{Concat}(\mathbf{P}, \widetilde{\mathbf{Q}})\Big)\,(\mathbf{x})\Big)\,. \end{aligned} \tag{5}$$

Here, $\mathbf{x}$ is a spatial location, $\widetilde{\mathbf{Q}}(\mathbf{x})$ is the concatenated physical-indicator vector at $\mathbf{x}$, $\mathcal{N}(\cdot)$ denotes per-sample channel normalization, and $\mathrm{Concat}(\cdot)$ denotes channel-wise concatenation. The quantity $I(\mathbf{x})$ is the scalar importance value, η_θ is a convolutional predictor parameterized by θ, and σ is the sigmoid function. The input $\mathbf{a}$ and its encoded descriptor $\mathbf{P}$ have been defined above.

The predicted importance modulates a bounded node displacement:

$$\begin{aligned} \mathbf{o}_i &= \tanh(\psi_\theta(\mathbf{P})_i)\,, \\ w_i &= \frac{I_{\min} + s_I I_i}{N^{-1}\sum_{k=1}^{N}(I_{\min} + s_I I_k) + \epsilon}, \\ \Delta\mathbf{x}_i &= \delta_{\max} w_i \mathbf{o}_i. \end{aligned} \tag{6}$$

Here, $\mathbf{o}_i$ is the bounded raw displacement of node i, ψ_θ is the convolutional displacement head, and $I_i = I(\mathbf{x}_i^r)$ is the importance value at that node. The quantity w_i is its positive mean-normalized importance weight, $I_{\min} > 0$ prevents complete suppression of a node, s_I controls the modulation range, k indexes the N nodes, and $\epsilon > 0$ ensures numerical stability. Finally, $\delta_{\max}$ bounds the displacement scale and $\Delta\mathbf{x}_i$ is the resulting displacement. The reference coordinate $\mathbf{x}_i^r$ and the number of nodes N have been defined in Section 3.1.

The resulting path $Q_g, Q_\Delta, Q_E, Q_\omega \to I(\mathbf{x}) \to w_i \to \Delta\mathbf{x}_i$ explicitly connects physical structure to node relocation, thereby supporting allocation readability. The nonuniform local discretizations produced by this allocation motivate the interaction mechanism introduced next.

3.4 Gauge-Aware Mesh Operator

Adaptive allocation changes local spacing, orientation, neighborhood geometry, and mesh Jacobians. GA-AMNO therefore applies an edge-conditioned feature transport before aggregating information from neighboring adaptive nodes. For each directed edge $j \to i$, the edge descriptor is

$$\mathbf{e}_{ij} = \mathrm{Concat}\left(\mathbf{x}_j^a - \mathbf{x}_i^a, \mathrm{vec}(\mathbf{J}_j^a), \mathrm{vec}(\mathbf{J}_i^a), \mathbf{p}_j, \mathbf{p}_i\right). \tag{7}$$

Here, $\mathbf{e}_{ij}$ is the descriptor of the edge from source node j to target node i, and $\mathbf{x}_j^a - \mathbf{x}_i^a$ is their relative adaptive coordinate. The matrices $\mathbf{J}_j^a$ and $\mathbf{J}_i^a$ are the estimated local mesh Jacobians at the source and target nodes, respectively, $\mathrm{vec}(\cdot)$ flattens a matrix into a vector, and $\mathbf{p}_j$ and $\mathbf{p}_i$ are their learned state descriptors. The adaptive coordinates $\mathbf{x}_i^a$ have been defined in Section 3.1.

The transport uses a diagonal map with a gated low-rank correction:

$$\mathbf{T}_{i\leftarrow j}^{(\ell)} = \mathrm{diag}(\mathbf{d}_{ij}^{(\ell)}) + \frac{\gamma_{ij}^{(\ell)}}{\sqrt{C_h r}} \mathbf{U}_{ij}^{(\ell)} (\mathbf{V}_{ij}^{(\ell)})^\top. \tag{8}$$

Here, $\mathbf{T}_{i\leftarrow j}^{(\ell)} \in \mathbb{R}^{C_h \times C_h}$ is the source-to-target transport matrix in layer ℓ, C_h is the hidden feature width, and r is the transport rank. The vector $\mathbf{d}_{ij}^{(\ell)} \in \mathbb{R}^{C_h}$ determines diagonal channel scaling, $\mathbf{U}_{ij}^{(\ell)}, \mathbf{V}_{ij}^{(\ell)} \in \mathbb{R}^{C_h \times r}$ determine the low-rank correction, and $\gamma_{ij}^{(\ell)}$ is an edge-conditioned sigmoid gate. The superscript $\top$ denotes matrix transpose. These quantities are predicted from $\mathbf{e}_{ij}$ by separate learned heads.

Each source feature is transported into its target-conditioned representation before weighted aggregation:

$$\begin{aligned} \alpha_{ij}^{(\ell)} &= \frac{\exp(a_{ij}^{(\ell)})}{\sum_{k\in\mathcal{N}(i)} \exp(a_{ik}^{(\ell)})}, \\ \mathbf{m}_i^{(\ell)} &= \sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(\ell)} \mathbf{T}_{i\leftarrow j}^{(\ell)} \mathbf{h}_j^{(\ell)}. \end{aligned} \tag{9}$$

Here, $a_{ij}^{(\ell)}$ is the learned compatibility score of edge $j \to i$, and $\alpha_{ij}^{(\ell)}$ is its normalized attention weight. The set $\mathcal{N}(i)$ contains the neighbors of target node i, while k indexes these neighbors in the softmax denominator. The vector $\mathbf{h}_j^{(\ell)}$ is the source-node feature in layer ℓ, and $\mathbf{m}_i^{(\ell)}$ is the transported message aggregated at target node i.

The target feature is then updated through

$$\mathbf{h}_i^{(\ell+1)} = \mathbf{h}_i^{(\ell)} + \rho_\theta^{(\ell)}\left(\mathbf{m}_i^{(\ell)}\right). \tag{10}$$

Here, $\mathbf{h}_i^{(\ell)}$ and $\mathbf{h}_i^{(\ell+1)}$ are the target-node features before and after the update, respectively, and $\rho_\theta^{(\ell)}$ is the layer-specific update network. The residual form preserves the current target feature while incorporating transported neighborhood information.

The transport is interpreted as a learned, geometry-conditioned feature correction rather than an analytically prescribed coordinate transformation. Its representation-level motivation and complete implementation details are provided in Appendix D.2, while its theoretical analysis is given in Appendix E. The subsequent residual prediction is described in Appendix D.4.

Table 1: Relative ℓ_2 error (lower is better). All the models report mean $\pm$ standard deviation over three seeds. Classic denotes adaptive mesh learning with direct aggregation and the shared base solver. Best results are bold.

Dataset	GA-AMNO	Classic	MMPDE	Transolver	FNO	GNO	WNO	UNet
Navier-Stokes	$\mathbf{0.003794 \pm 0.000624}$	0.011175 ± 0.000065	0.005300 ± 0.000535	0.019768	0.006781	0.033800	0.016903	0.005054
Rayleigh-Bénard	$\mathbf{0.000927 \pm 0.000363}$	0.001472 ± 0.000213	0.003032 ± 0.000477	0.002664	0.006327	0.003954	0.007910	0.010840
KS	$\mathbf{0.000380 \pm 0.000061}$	0.000891 ± 0.000128	0.003988 ± 0.001184	0.013923	0.009289	0.010016	0.009549	0.010877
Kolmogorov	$\mathbf{0.020221 \pm 0.001529}$	0.081671 ± 0.001860	0.069614 ± 0.003031	0.070090	0.030034	0.090519	0.121220	0.028476
Darcy	$\mathbf{0.028157 \pm 0.001701}$	0.341614 ± 0.004773	0.030082 ± 0.001907	0.046329	0.028557	0.254866	0.118636	0.030238

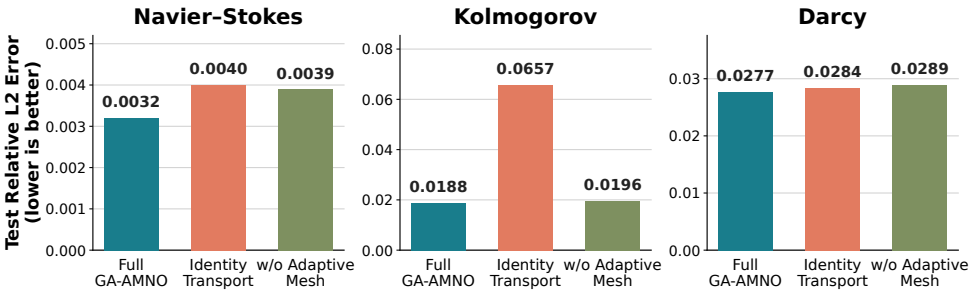

Figure 4: Ablation of the two core mechanisms. Identity Transport retains the adaptive mesh but removes learned representation transport, while w/o Adaptive Mesh retains the remaining solver without learned node relocation. Lower test relative ℓ_2 error is better.

4 Experimental Results

4.1 Overall Predictive Performance

Table 1 compares GA-AMNO with seven representative baselines covering spectral operators, graph operators, convolutional architectures, transformer-based operators, and adaptive-mesh solvers. GA-AMNO achieves the lowest central relative ℓ_2 error on all five benchmarks. This consistent advantage across substantially different computational paradigms demonstrates that the proposed architecture is not specialized to outperform only one particular operator family. Instead, physics-informed adaptive allocation, low-rank gauge transport, and residual field refinement form an effective operator-learning framework that improves solution approximation across regular-grid, graph-based, and adaptive-mesh alternatives.

The comparisons with Classic and MMPDE are particularly important to the central claim of this work. Classic uses an adaptive mesh with direct feature aggregation, while MMPDE introduces a dedicated moving-mesh mechanism. GA-AMNO consistently outperforms both, showing that determining where nodes should be allocated is only part of successful adaptive operator learning. Explicitly transporting source features into the target representation context before aggregation provides an additional and practically useful capability for processing heterogeneous local discretizations. The strong results across Navier-Stokes, Rayleigh-Bénard, KS, Kolmogorov, and Darcy further demonstrate that this benefit extends from evolutionary and chaotic dynamics to static coefficient-to-solution mappings. Overall, the main experiment supports the central design principle of operator readability: the predictive benefit emerges when adaptive discretization is operator-readable, while broad applicability is retained across distinct PDE regimes.

4.2 Component Ablation

We isolate the two mechanisms central to operator-readable adaptation while retaining the remaining architecture and evaluation protocol. Identity Transport replaces every learned source-to-target transport with the identity map, whereas w/o Adaptive Mesh removes the learned node relocation. Figure 4 reports the resulting test relative ℓ_2 errors on Navier-Stokes, Kolmogorov, and Darcy.

Full GA-AMNO achieves the lowest error on all three datasets. Replacing learned transport with the identity increases error, most substantially on Kolmogorov, while removing adaptive allocation also consistently degrades performance. The comparison directly verifies that both where representations

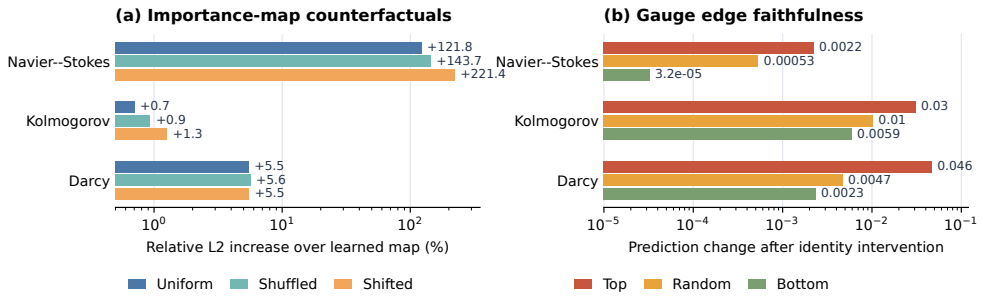

Figure 5: Causal tests of allocation and interaction readability. (a) Relative ℓ_2 increase after replacing the learned importance map with uniform, shuffled, or shifted alternatives. (b) Prediction change after replacing transport with the identity on top-ranked, random, or bottom-ranked edge subsets.

are allocated and how they are aligned contribute to the complete model. Targeted ablations of the remaining components are provided in Figure 8 in the extended results.

4.3 Interpreting Adaptive Computation

This subsection examines whether the learned importance map is used by the predictor, whether the learned connection identifies influential edges, and whether transport converts source messages into target-compatible representations under strong local geometric mismatch. All experiments use frozen checkpoints and held-out samples, with no parameter updates during evaluation. Prediction effects in high-adaptation-demand regions are reported in the targeted component ablation of Figure 8, avoiding duplicate tabulation of the same quantities.

4.3.1 Importance Allocation Faithfulness

We intervene on the learned importance map while leaving the remaining network unchanged. The learned map is replaced by a spatially uniform map, randomly shuffled within each sample, or cyclically shifted. If the map merely visualized activity without controlling useful computation, these counterfactuals would leave the prediction nearly unchanged.

As shown in Figure 5(a), the learned map is best in all three cases, but the impact varies across PDEs. Navier-Stokes exhibits a strong counterfactual gap, Darcy exhibits a moderate gap, and Kolmogorov is only weakly sensitive. The counterfactual design is important because it distinguishes a visually plausible importance map from one that actually controls prediction: preserving its values while destroying their spatial correspondence degrades performance. This experiment therefore supports allocation readability by showing that the learned spatial allocation is functionally used, although it does not prove that every highlighted region has a unique physical interpretation or that importance identifiability is equally strong for every PDE.

4.3.2 Gauge Edge Faithfulness

Using the correction magnitude and prediction-change metric defined in Appendix C, we replace the learned transport on a selected 10% of edges by the identity map. Top-ranked, random, and bottom-ranked edge subsets are evaluated under the same intervention budget.

As shown in Figure 5(b), top-ranked interventions produce 2.99–9.92 times the prediction change of random interventions, while bottom-ranked edges have little influence. The correction magnitude is therefore functionally faithful to the trained operator: edges assigned a large representation repair are also the edges on which the output causally depends. The purpose of this intervention is to establish a causal link between an interpretable internal quantity and the final prediction, which cannot be obtained from feature visualization alone. It supports interaction readability by showing that the learned connection identifies consequential information transfers rather than acting as a decorative latent variable. The correction magnitude is not itself claimed to be a physical observable.

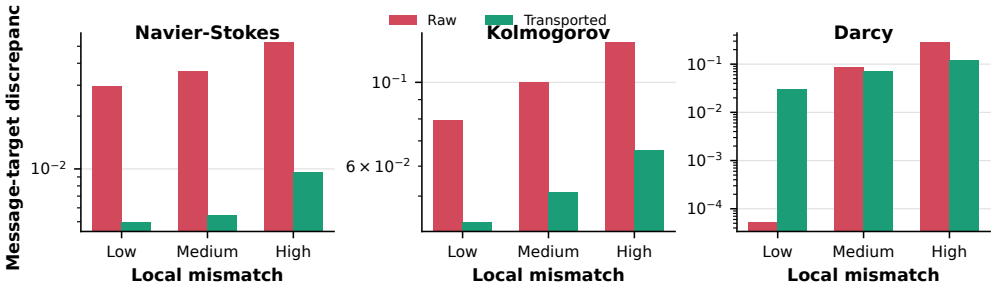

Figure 6: Source–target and transported-message–target discrepancy stratified by local geometric mismatch. The reduction measures message compatibility before aggregation, not the covariance defect of the learned connection.

4.3.3 Message Compatibility by Local Geometric Mismatch

We use the geometric-mismatch groups and compatibility metrics defined in Appendix C to compare source–target discrepancy with transported-message–target discrepancy. This evaluation asks whether the learned correction is most active where the adaptive geometry is most nonuniform; it does not by itself separate physical feature variation from discretization-induced variation.

As summarized in Table 4, transport reduces message–target discrepancy by 48.32%–81.93% in the high-geometric-mismatch group. The full stratification in Figure 6 shows that this compatibility gain is concentrated where the adaptive geometry is most nonuniform. This concentration is the key mechanism-level result: the learned correction is strongest where direct comparison is least justified by the local discretization, which is the regime targeted by interaction readability. Together with the edge intervention and regional prediction ablation, the result connects geometric mismatch, feature correction, and output utility. Because the metric also contains actual source–target physical variation, it remains a compatibility diagnostic rather than a standalone proof of gauge covariance.

5 Conclusion

We introduced GA-AMNO to address a structural issue in adaptive neural operators: state-dependent discretization changes not only where information is sampled, but also the local representation context in which information is encoded and exchanged. GA-AMNO therefore treats adaptive computation as two coupled problems, using physics-informed adaptive allocation to determine where resolution should be allocated and low-rank gauge transport to determine how features associated with unequal local discretizations should interact before aggregation. The theoretical analysis explains why shared direct aggregation cannot generally remove discretization-induced representation ambiguity, establishes sufficient conditions for consistent transported aggregation, and bounds the effect of approximate learned connections without claiming exact gauge equivariance. The counterfactual, intervention, mismatch-stratified, and deformation experiments further show that the learned allocation and transport are functionally used rather than merely producing visually plausible meshes or latent variables. The broader significance of this work is to reposition adaptive discretization from an opaque preprocessing or resource-allocation mechanism into a mechanism that operationalizes operator readability, with allocation decisions and information transfers that can both be inspected and tested. Future work should extend this formulation to explicitly equivariant connection parameterizations, topology-changing and anisotropic mesh adaptation, three-dimensional unstructured domains, and scalable sparse transport, while developing stronger links between representation consistency, numerical approximation error, and the convergence properties of learned PDE operators.

References

Ferran Alet, Adarsh K. Jeewajee, Maria Bauza Villalonga, Alberto Rodriguez, Tomas Lozano-Perez, and Leslie P. Kaelbling. Graph element networks: Adaptive, structured computation and memory. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 212–222. PMLR, 2019.

Francesca Bartolucci, Emmanuel de Bezenac, Bogdan Raonic, Roberto Molinaro, Siddhartha Mishra, and Rima Alaifari. Representation equivalent neural operators: a framework for alias-free operator learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=7LSEkvEGCM.

Marsha J Berger and Phillip Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of computational Physics*, 82(1):64–84, 1989.

Marsha J Berger and Joseph Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of computational Physics*, 53(3):484–512, 1984.

Pengfei Bie, Ning Song, Nuoqing Zhang, Jie Nie, Min Ye, Xinyue Liang, and Qi Wen. Space–frequency cross-attention node feature optimization graph neural operator for partial differential equations. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Keaton J Burns, Geoffrey M Vasil, Jeffrey S Oishi, Daniel Lecoanet, and Benjamin P Brown. Dedalus: A flexible framework for numerical simulations with spectral methods. *Physical Review Research*, 2(2):023068, 2020.

Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International conference on Machine learning*, pp. 1321–1330. PMLR, 2019.

Filipe de Avila Belbute-Peres, Thomas Economon, and Zico Kolter. Combining differentiable PDE solvers and graph neural networks for fluid flow prediction. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2402–2411. PMLR, 2020.

Pim de Haan, Maurice Weiler, Taco Cohen, and Max Welling. Gauge equivariant mesh CNNs: Anisotropic convolutions on geometric graphs. In *International Conference on Learning Representations*, 2021.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3d roto-translation equivariant attention networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Efficient token mixing for transformers via adaptive fourier neural operators. In *International Conference on Learning Representations*, 2022.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. GNOT: A general neural operator transformer for operator learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12556–12569. PMLR, 2023.

Lingshen He, Yiming Dong, Yisen Wang, Dacheng Tao, and Zhouchen Lin. Gauge equivariant transformer. *Advances in Neural Information Processing Systems*, 34:27331–27343, 2021.

Jacob Helwig, Xuan Zhang, Cong Fu, Jerry Kurtin, Stephan Wojtowytsch, and Shuiwang Ji. Group equivariant fourier neural operators for partial differential equations. *arXiv preprint arXiv:2306.05697*, 2023.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel De Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

Peiyan Hu, Yue Wang, and Zhi-Ming Ma. Better neural pde solvers through data-free mesh movers. In *International Conference on Learning Representations*, volume 2024, pp. 4550–4576, 2024.

Shudong Huang, Wentao Feng, Chenwei Tang, Zhenan He, Caiyang Yu, and Jiancheng Lv. Partial differential equations meet deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 36(8):13649–13669, 2025. doi: 10.1109/TNNLS.2025.3545967.

Erik Jenner and Maurice Weiler. Steerable partial differential operators for equivariant neural networks. *arXiv preprint arXiv:2106.10163*, 2021.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Zijie Li, Saurabh Patil, Francis Ogoke, Dule Shu, Wilson Zhen, Michael Schneier, John R Buchanan Jr, and Amir Barati Farimani. Latent neural pde solver: A reduced-order modeling framework for partial differential equations. *Journal of Computational Physics*, 524:113705, 2025.

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

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for PDEs on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023a.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36:35836–35854, 2023b.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/IMS Journal of Data Science*, 1(3):1–27, 2024. doi: 10.1145/3648506.

Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. Pde-refiner: Achieving accurate long rollouts with neural pde solvers. *Advances in Neural Information Processing Systems*, 36:67398–67433, 2023.

Shouyi Liu, Xiaokang Yang, and Yuntian Chen. Riesz neural operator for solving partial differential equations. In *The Fourteenth International Conference on Learning Representations*, 2026.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5.

Leon Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960.

Jung Yeon Park, Lawson Wong, and Robin Walters. Modeling dynamics over meshes with gauge equivariant nonlinear message passing. *Advances in Neural Information Processing Systems*, 36: 15277–15302, 2023.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

Md Ashiqur Rahman, Zachary E. Ross, and Kamyar Azizzadenesheli. U-NO: U-shaped neural operators. *Transactions on Machine Learning Research*, 2023.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

François Rozet and Gilles Louppe. Score-based data assimilation. *Advances in Neural Information Processing Systems*, 36:40521–40541, 2023.

Keith Rudd, Gianluca Di Muro, and Silvia Ferrari. A constrained backpropagation approach for the adaptive solution of partial differential equations. *IEEE Transactions on Neural Networks and Learning Systems*, 25(3):571–584, 2014. doi: 10.1109/TNNLS.2013.2277601.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 2020.

Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9323–9332. PMLR, 2021.

Aliaksandra Shysheya, Cristiana Diaconu, Federico Bergamin, Paris Perdikaris, José M Hernández-Lobato, Richard E Turner, and Emile Mathieu. On conditional diffusion models for PDE simulations. *Advances in Neural Information Processing Systems*, 37:23246–23300, 2024.

Wenbin Song, Mingrui Zhang, Joseph G Wallwork, Junpeng Gao, Zheng Tian, Fanglei Sun, Matthew Piggott, Junqing Chen, Zuoqiang Shi, Xiang Chen, et al. M2n: Mesh movement networks for pde solvers. *Advances in Neural Information Processing Systems*, 35:7199–7210, 2022.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. In *International Conference on Learning Representations*, 2023.

Tapas Tripura and Souvik Chakraborty. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 404:115783, 2023. doi: 10.1016/j.cma.2022.115783.

Zhichao Wang, Xinhai Chen, Qinglin Wang, Xiang Gao, Qingyang Zhang, Menghan Jia, Xiang Zhang, and Jie Liu. Ugm2n: An unsupervised and generalizable mesh movement network via m-uniform loss. *arXiv preprint arXiv:2508.08615*, 2025.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S. Cohen. 3d steerable CNNs: Learning rotationally equivariant features in volumetric data. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. *arXiv preprint arXiv:2402.02366*, 2024.

Tailin Wu, Takashi Maruyama, Qingqing Zhao, Gordon Wetzstein, and Jure Leskovec. Learning controllable adaptive simulation for multi-resolution physics. *arXiv preprint arXiv:2305.01122*, 2023.

A Algorithmic Summary

Table 2 summarizes the forward procedure of GA-AMNO. All learnable components are optimized jointly during training.

B Expanded Related Work

B.1 Neural Operators for PDE Solving

Neural operators learn mappings between function spaces and provide a practical route to surrogate PDE solvers that can share parameters across discretization resolutions Huang et al. (2025); Kovachki et al. (2023). Deep Operator Network (DeepONet) represents an input function and its query

Table 2: Forward procedure of the proposed GA-AMNO.

Step	Operation
Input	Receive the temporal history or steady conditioning field $\mathbf{a}$ on the reference discretization $\mathcal{X}^r$.
1	Encode the input into a state descriptor: $\mathbf{P} = \mathrm{Encoder}(\mathbf{a})$.
2	Compute gradient, Laplacian, energy, and vorticity-related indicators from the latest or conditioning field $\mathbf{v}$.
3	Predict the importance map from the encoded state and normalized physical indicators: $I = \sigma\Big(\eta_\theta(\mathrm{Concat}(\mathbf{P}, \widetilde{\mathbf{Q}}))\Big)$.
4	Predict bounded node directions and construct importance-modulated adaptive coordinates: $\mathbf{x}_i^a = \Pi_\Omega(\mathbf{x}_i^r + \delta_{\max} w_i \mathbf{o}_i)$.
5	Estimate local mesh Jacobians $\mathbf{J}_i^a$ and form each directed edge descriptor $\mathbf{e}_{ij}$ from relative coordinates, source and target Jacobians, and source and target state descriptors.
6	Predict the edge attention score and diagonal-plus-low-rank transport: $\mathbf{T}_{i\leftarrow j}^{(\ell)} = \mathrm{diag}(\mathbf{d}_{ij}^{(\ell)}) + \gamma_{ij}^{(\ell)} \mathbf{U}_{ij}^{(\ell)} (\mathbf{V}_{ij}^{(\ell)})^\top / \sqrt{C_h r}$.
7	Transport every source feature into its target-conditioned representation: $\widetilde{\mathbf{h}}_{i\leftarrow j}^{(\ell)} = \mathbf{T}_{i\leftarrow j}^{(\ell)} \mathbf{h}_j^{(\ell)}$.
8	Aggregate transported messages and update the target feature: $\mathbf{m}_i^{(\ell)} = \sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(\ell)} \widetilde{\mathbf{h}}_{i\leftarrow j}^{(\ell)}$ and $\mathbf{h}_i^{(\ell+1)} = \mathbf{h}_i^{(\ell)} + \rho_\theta^{(\ell)}(\mathbf{m}_i^{(\ell)})$.
9	Repeat Steps 6–8 for all L_g low-rank gauge transport layers and modulate the final nodal features by the learned importance weights.
10	Reconstruct the adaptive nodal features on the canonical grid using coarse-grid upsampling or distance-weighted interpolation, producing $\mathbf{F}_{\mathrm{grid}}$.
11	Predict the local residual $\boldsymbol{\delta}_{\mathrm{raw}}$ and the truncated spectral correction $\boldsymbol{\delta}_{\mathrm{spec}}$, and set $\boldsymbol{\delta}_1 = \boldsymbol{\delta}_{\mathrm{raw}} + \boldsymbol{\delta}_{\mathrm{spec}}$.
12	Construct the provisional field $\mathbf{u}_{\mathrm{base}}$, compute its finite-difference features, and obtain the differential correction and final residual $\boldsymbol{\delta}$.
Output	Return $\widehat{\mathbf{u}} = \mathbf{u}_t + \boldsymbol{\delta}$ for temporal residual prediction, or $\widehat{\mathbf{u}} = \boldsymbol{\delta}$ for steady or direct-output problems.

coordinates through branch and trunk networks Lu et al. (2021), whereas Fourier Neural Operator (FNO) parameterizes global integral operators in the spectral domain Li et al. (2020a). This basic formulation has subsequently been extended through factorized spectral layers Tran et al. (2023), adaptive Fourier token mixing Guibas et al. (2022), U-shaped multiscale architectures Rahman et al. (2023), wavelet decompositions Tripura & Chakraborty (2023), and physics-informed objectives Li et al. (2024). Graph Neural Operator (GNO) Li et al. (2020b) and Message Passing Neural PDE Solvers Brandstetter et al. (2022) support local interactions on irregular samples, while Geo-FNO Li et al. (2023a) learns a deformation between irregular physical domains and a regular latent grid. Geometry-Informed Neural Operator further combines graph-based lifting with spectral processing for large-scale PDEs on varying geometries Li et al. (2023b). Transformer-based operators, including GNOT Hao et al. (2023) and Transolver Wu et al. (2024), model long-range physical interactions, while explicit space–frequency coupling improves the recovery of multiscale solution features Bie et al. (2025). More recent work has addressed stable long-horizon prediction through iterative spectral refinement Lippe et al. (2023), general-purpose PDE pretraining through the Poseidon foundation model Herde et al. (2024), and the integration of global spectra with local derivative information through the Riesz Neural Operator Liu et al. (2026). Despite these advances, the numerical realization of a neural operator still acts on discretized field representations. Representation Equivalent Neural Operators show that changes in sampling and reconstruction can introduce operator aliasing and break consistency between continuous operators and their discrete realizations Bartolucci et al. (2023). This observation is closely related to our motivation, but our focus is different: rather than addressing global continuous–discrete equivalence alone, we study how features produced under unequal, input-dependent local discretizations should interact inside an adaptive-mesh operator.

B.2 Data-driven Adaptive Mesh Methods

Adaptive meshing concentrates computational resolution near sharp gradients, discontinuities, and dynamically complex structures. Classical adaptive mesh refinement uses equation-dependent er-

ror indicators and hierarchical refinement rules Berger & Oliger (1984); Berger & Colella (1989), while constrained neural adaptation provides an early example of learning spatial resolution for PDE approximation Rudd et al. (2014). Graph Element Networks introduce adaptive structured computation over spatial elements Alet et al. (2019), and graph-based simulators subsequently support physical prediction on mesh and particle representations Sanchez-Gonzalez et al. (2020). MeshGraphNets learn dynamics directly on simulation meshes and adapt connectivity during rollout Pfaff et al. (2020), whereas differentiable combinations of PDE solvers and graph networks preserve a stronger connection to conventional numerical computation de Avila Belbute-Peres et al. (2020). More explicit learned-mesh approaches treat node placement itself as an optimization problem. M2N learns end-to-end mesh movement for PDE solvers Song et al. (2022), and LAMP jointly learns physical evolution with a controllable refinement and coarsening policy for multiresolution simulation Wu et al. (2023). Data-Free Mesh Movers construct an r-adaptive mesh through a Monge–Ampère-based objective and integrate the resulting mesh into a learned PDE solver Hu et al. (2024). UGM2N further introduces an unsupervised mesh-movement objective based on local equidistribution Wang et al. (2025). Related operator-learning methods handle geometric variation through learned global coordinate transformations: Geo-FNO maps irregular geometries to a uniform latent domain Li et al. (2023a), while GINO transfers information between irregular point sets and a regular latent grid Li et al. (2023b). These methods substantially improve where computation is allocated or how irregular data are embedded. However, the adapted mesh is still commonly treated as a sampling, interpolation, or message-passing structure once it has been generated. Node relocation changes local density, orientation, spatial support, and neighborhood geometry, but conventional aggregation does not explicitly account for how these changes affect the representation convention of each node feature. Consequently, physical variation and discretization-induced representation variation may be mixed during information exchange. GA-AMNO addresses this complementary problem by coupling physics-informed adaptive allocation with source-to-target feature transport before aggregation.

B.3 Equivariant and Gauge-Aware Representation Learning

Equivariant representation learning provides a principled framework for processing features under transformations of coordinates, orientations, or local reference frames. Geometric deep learning unifies many of these constructions across grids, groups, graphs, and manifolds Bronstein et al. (2021). Steerable three-dimensional convolutions Weiler et al. (2018), $\mathrm{SE}(3)$-Transformers Fuchs et al. (2020), and $E(n)$-equivariant graph networks Satorras et al. (2021) preserve prescribed transformation laws under global rotations and translations. In operator learning, Group Equivariant Fourier Neural Operators extend rotational, translational, and reflectional symmetries to spectral operator layers Helwig et al. (2023). Steerable partial differential operators provide a complementary characterization of when differential mappings between feature fields are equivariant Jenner & Weiler (2021). These global symmetry constructions improve physical consistency, but they do not directly resolve the comparison of features expressed in independently varying local frames. Gauge-equivariant convolutional networks instead associate each location with a local representation frame and require feature transformations to follow a prescribed gauge action Cohen et al. (2019). Gauge Equivariant Mesh CNNs extend this principle to manifold meshes by parallel-transporting neighboring features before applying anisotropic convolution de Haan et al. (2021). Gauge Equivariant Transformer incorporates parallel transport into self-attention on triangular meshes He et al. (2021), while gauge-equivariant nonlinear message passing applies the same geometric principle to nonlinear dynamics and PDE evolution on surfaces Park et al. (2023). These studies establish the central principle that features expressed under different local conventions should be transported into compatible representations before interaction. However, they generally assume a fixed geometric domain, explicitly constructed local frames, and predefined transformation laws. GA-AMNO considers a different setting in which the discretization itself depends on the input. The resulting local representation changes are not known a priori and are not claimed to obey exact gauge equivariance. Instead, relative coordinates, adaptive-mesh Jacobians, and state descriptors condition a learned diagonal-plus-low-rank transport that maps each source feature into a target-conditioned representation before aggregation. The gauge interpretation therefore provides a model for interaction readability under input-dependent discretization, while the learned connection is treated as an approximate, functionally testable representation-alignment mechanism.

C Experimental Setting

C.1 Datasets and Data Preparation

We evaluate GA-AMNO on five PDE benchmarks: Navier-Stokes Li et al. (2025), Rayleigh-Bénard Burns et al. (2020), KS Shysheya et al. (2024), Kolmogorov Rozet & Louppe (2023), and Darcy Li et al. (2025). The benchmarks cover scalar and multi-channel temporal prediction as well as a steady-state coefficient-to-solution mapping.

The Navier-Stokes dataset contains $5{,}000$ trajectories of two-dimensional incompressible flow at viscosity 10^{-3}. Each trajectory contains 50 scalar flow-state snapshots on a 64×64 grid. We use trajectories 1–1,000 for training, 1,001–1,100 for validation, and the final 100 trajectories for testing. The Rayleigh-Bénard dataset contains vorticity and temperature-anomaly fields on a 64×64 grid. Its training, validation, and test splits contain 256, 64, and 128 trajectories, respectively, with 41 snapshots per trajectory. The KS dataset contains one-dimensional scalar trajectories with 256 spatial points and 641 temporal snapshots. The training, validation, and test splits contain 800, 100, and 50 trajectories, respectively. The Kolmogorov dataset contains two-channel forced-flow states on a 64×64 grid. It provides 819 training trajectories, 102 validation trajectories, and 103 test trajectories, with 64 snapshots per trajectory. The Darcy benchmark is a steady-state operator-learning problem on a 64×64 grid. Its training, validation, and test splits contain 900, 124, and $1{,}024$ samples, respectively. Each sample stores a coefficient field followed by its normalized solution field.

For the four temporal benchmarks, the input is a history of four consecutive snapshots, $\mathcal{U}_t = \{u_{t-3}, u_{t-2}, u_{t-1}, u_t\}$, and the target is the immediately following state u_{t+1}. The starting indices of successive windows are separated by 8 snapshots for Navier-Stokes, Rayleigh-Bénard, and KS, and by 4 snapshots for Kolmogorov. Thus, the stride controls window extraction and does not introduce gaps between the four states inside a history window. For Darcy, the coefficient field is used as a single conditioning input and the corresponding solution is the target. No additional dataset-level normalization is applied during training; the pre-normalized Darcy files are loaded directly.

C.2 Baselines and Comparison Protocol

We compare GA-AMNO with seven representative baselines. **1)** FNO Li et al. (2020a) applies global spectral convolution on a regular grid. **2)** Transolver Wu et al. (2024) uses physics-aware slice attention to model long-range spatial interactions. **3)** GNO Li et al. (2020b) performs coordinate-conditioned local message passing. **4)** WNO Tripura & Chakraborty (2023) uses Haar-wavelet operator blocks to capture localized multiresolution components. **5)** UNet Ronneberger et al. (2015) is a convolutional encoder–decoder with multiscale skip connections. **6)** Classic is a controlled adaptive-mesh baseline that learns node displacements but directly aggregates node features using the shared base solver, without physics-informed adaptive allocation, low-rank gauge transport, spectral residual correction, or PDE-aware correction. **7)** MMPDE Hu et al. (2024) combines a data-free differentiable mesh mover with regular-mesh and moving-mesh solution branches to perform adaptive PDE prediction. Our implementations of Transolver, GNO, WNO, and MMPDE follow the core mechanisms of the cited architectures and are adapted to the unified input–output protocol, spatial resolution, and training configuration used in this study. Classic shares the corresponding base prediction protocol with GA-AMNO and serves as a controlled comparison for evaluating the benefit of operator-readable adaptive discretization beyond adaptive node movement alone.

All methods use the same dataset files, trajectory splits, history windows, prediction targets, spatial resolutions, learning rate, epoch count, and number of updates per epoch. Models that use coordinates receive the same regular physical-coordinate encoding. Checkpoints are selected by the same validation relative ℓ_2 criterion. The comparison controls the data protocol and optimization-step budget but does not enforce parameter- or FLOP-matched architectures. Parameter counts and counted FLOPs are therefore reported separately when computational cost is discussed.

C.3 Evaluation Metrics

We use relative ℓ_2 error as the primary metric for predictive comparison. RMSE and spectral-energy error are additionally reported for rollout evaluation, while gradient, differential, and high-band

spectral errors are used for targeted mechanism analyses.

$$\mathrm{RelL2}(\widehat{u}, u) = \frac{\|\widehat{u} - u\|_2}{\|u\|_2 + \epsilon},$$

$$\mathrm{RMSE}(\widehat{u}, u) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widehat{u}_i - u_i)^2},$$

$$\mathrm{SpectrumError}(\widehat{u}, u) = \frac{\|\overline{|\mathcal{F}_{\mathrm{DFT}}(\widehat{u})|^2}_c - \overline{|\mathcal{F}_{\mathrm{DFT}}(u)|^2}_c\|_2}{\|\overline{|\mathcal{F}_{\mathrm{DFT}}(u)|^2}_c\|_2 + \epsilon}. \tag{11}$$

Here, $\widehat{u}$ and u are the prediction and ground truth, N is the number of scalar field entries, $\mathcal{F}_{\mathrm{DFT}}$ is the orthonormal discrete Fourier transform, $\overline{(\cdot)}_c$ denotes averaging over channels, and $\epsilon = 10^{-8}$. The overall $\mathrm{SpectrumError}$ measures spectral-energy discrepancy over all represented frequencies and is used for rollout evaluation. For the Spectral Residual targeted ablation, we separately report high-band spectral error. Let

$$\mathcal{K}_{\mathrm{hi}} = \left\{ k : \frac{\nu(k)}{\nu_{\max}} \geq 0.6 \right\},$$

where $\nu(k)$ is the radial frequency of coefficient k and $\nu_{\max}$ is the maximum represented radius. The high-band spectral error is the same channel-averaged spectral-energy discrepancy restricted to $\mathcal{K}_{\mathrm{hi}}$:

$$\mathrm{HighBandError}(\widehat{u}, u) = \frac{\left\| \overline{|\mathcal{F}_{\mathrm{DFT}}(\widehat{u})|^2}_{c,\mathcal{K}_{\mathrm{hi}}} - \overline{|\mathcal{F}_{\mathrm{DFT}}(u)|^2}_{c,\mathcal{K}_{\mathrm{hi}}} \right\|_2}{\left\| \overline{|\mathcal{F}_{\mathrm{DFT}}(u)|^2}_{c,\mathcal{K}_{\mathrm{hi}}} \right\|_2 + \epsilon}.$$

Here, the subscript $c, \mathcal{K}_{\mathrm{hi}}$ denotes channel averaging followed by restriction to the high-band coefficients. Gradient error is used in mechanism and regional analyses:

$$\mathrm{GradError}(\widehat{u}, u) = \frac{\|[D_x\widehat{u} - D_x u, D_y\widehat{u} - D_y u]\|_2}{\|[D_x u, D_y u]\|_2 + \epsilon}, \tag{12}$$

where D_x and D_y are centered finite differences in grid-index units with one-sided boundary differences. For one-dimensional KS data, the inactive vertical derivative is zero.

For Darcy, the current auxiliary differential diagnostic is reported as Laplacian discrepancy,

$$E_\Delta = \frac{\|\Delta\widehat{u} - \Delta u\|_2}{\|\Delta u\|_2 + \epsilon}, \tag{13}$$

rather than as an equation residual, because it does not explicitly evaluate $-\nabla \cdot (a\nabla\widehat{u}) - f$. For scalar Navier-Stokes and KS fields, divergence, vector vorticity, and enstrophy are not reported because they are not defined by the stored single-channel representation. Physics-specific vector-field diagnostics are used only when the stored channels have the required physical meaning.

To study input-dependent adaptation demand, we use

$$\begin{aligned} d(\mathbf{x}) = {} & 0.4\mathcal{N}(g(\mathbf{x})) + 0.3\mathcal{N}(c(\mathbf{x})) \\ & + 0.2\mathcal{N}(m(\mathbf{x})) + 0.1\mathcal{N}(I(\mathbf{x})), \end{aligned} \tag{14}$$

where $g = \|\nabla u_t\|$ is input-gradient magnitude, $c = |\Delta u_t|$ is input-curvature magnitude, m is the norm of the local finite-difference variation of the learned mesh offset, and I is the learned importance map. $\mathcal{N}$ denotes per-sample robust normalization using the 5th and 95th percentiles. The lower, middle, and upper terciles of d define low-, medium-, and high-demand masks. The masks are constructed once from the input and the full GA-AMNO geometry, without using targets or predictions, and the same masks are applied to every compared model.

Importance allocation faithfulness is evaluated by replacing the learned importance map with a spatially uniform, randomly shuffled, or cyclically shifted map and then reporting the resulting relative ℓ_2 error. Gauge edge faithfulness is measured using the transport-correction magnitude

$$c_{ij} = \left\| \mathbf{T}_{i \leftarrow j}\mathbf{h}_j - \mathbf{h}_j \right\|_2, \tag{15}$$

where $\mathbf{h}_j$ is the source feature and $\mathbf{T}_{i\leftarrow j}$ is the learned transport from node j to node i. After replacing the transport on an edge subset S by the identity, its causal influence is quantified as

$$\Delta_{\mathrm{pred}}(S) = \frac{\|\widehat{u}_S - \widehat{u}_{\mathrm{full}}\|_2}{\|\widehat{u}_{\mathrm{full}}\|_2 + \epsilon}, \tag{16}$$

where $\widehat{u}_S$ is the intervened prediction and $\widehat{u}_{\mathrm{full}}$ is the original prediction.

Message–target compatibility is evaluated after dividing directed edges into low-, medium-, and high-geometric-mismatch groups using local edge-scale ratio, mesh-Jacobian variation, and importance difference. For each group $\mathcal{E}_b$, we compute

$$\begin{aligned} E^{(b)}_{\mathrm{src}} &= \frac{1}{|\mathcal{E}_b|} \sum_{(i,j)\in\mathcal{E}_b} \frac{\|\mathbf{h}_i - \mathbf{h}_j\|_2}{\|\mathbf{h}_i\|_2 + \epsilon}, \\ E^{(b)}_{\mathrm{msg}} &= \frac{1}{|\mathcal{E}_b|} \sum_{(i,j)\in\mathcal{E}_b} \frac{\|\mathbf{h}_i - \mathbf{T}_{i\leftarrow j}\mathbf{h}_j\|_2}{\|\mathbf{h}_i\|_2 + \epsilon}, \\ G^{(b)}_{\mathrm{comp}} &= \frac{E^{(b)}_{\mathrm{src}} - E^{(b)}_{\mathrm{msg}}}{E^{(b)}_{\mathrm{src}} + \epsilon}. \end{aligned} \tag{17}$$

Here, b indexes the geometric-mismatch group, $E^{(b)}_{\mathrm{src}}$ is the source–target feature discrepancy, $E^{(b)}_{\mathrm{msg}}$ is the transported-message–target discrepancy, and $G^{(b)}_{\mathrm{comp}}$ is their relative reduction. A positive value means that the message supplied to aggregation is closer to the current target feature. Because neighboring nodes may contain actually different physical content, this quantity is a message-compatibility diagnostic rather than a direct estimate of the covariance defect in Equation 36. The training-time loop diagnostic constrains only the diagonal transport component around sampled local triangular cycles and is therefore termed diagonal loop consistency, not exact holonomy of the complete low-rank transport. Mesh inversion rate, minimum cell angle, and aspect ratio are reported only as mesh-health diagnostics.

C.4 Implementation Details

GA-AMNO uses a hidden width of 32, two low-rank gauge transport layers, six indexed neighbors per node, and a low-rank transport rank of $r = 4$. The operator stride is 2 for the two-dimensional datasets and 1 for KS. The maximum latent-mesh displacement is 0.08 for the temporal benchmarks and is reduced to 0.02 for Darcy. Dataset-specific spectral modes and auxiliary-loss weights are selected using only the validation split. The reference connectivity is fixed, while latent coordinates, local Jacobians, edge descriptors, and transport matrices depend on the input state.

All methods are optimized with AdamW for 300 fixed-budget epochs using an initial learning rate of 10^{-3}. Each epoch consists of 100 mini-batch updates, giving 30,000 parameter updates per run rather than 300 complete passes through each windowed dataset. The default training batch size is 8. Owing to GPU-memory constraints, GA-AMNO uses a batch size of 4 on Rayleigh-Bénard, whereas the corresponding baseline runs use batch size 8. Validation is performed every 10 epochs, in addition to the first and final epochs, and the checkpoint with the lowest validation relative ℓ_2 error is selected for testing. GA-AMNO is evaluated using three random seeds, 42, 43, and 44, and its main results are reported as the mean and standard deviation across these runs. Unless otherwise specified, the baseline and mechanism-analysis experiments use random seed 42.

The GA-AMNO objective combines field MSE with supervised gradient, Laplacian, and spectral losses, together with diagonal loop consistency, transport alignment, mesh-quality, physics-informed allocation, frame-consistency, cross-mesh, and mesh-perturbation regularizers. The weights are tuned separately for each dataset on its validation split and remain fixed during test evaluation. These auxiliary terms regularize the learned latent geometry and feature transport; they are not treated as exact gauge-equivariance constraints.

C.5 Efficiency and Memory Scalability

We evaluate computational scalability using the Navier-Stokes GA-AMNO checkpoint on an NVIDIA GeForce RTX 4070 Laptop GPU. The model contains 177.4K trainable parameters. At

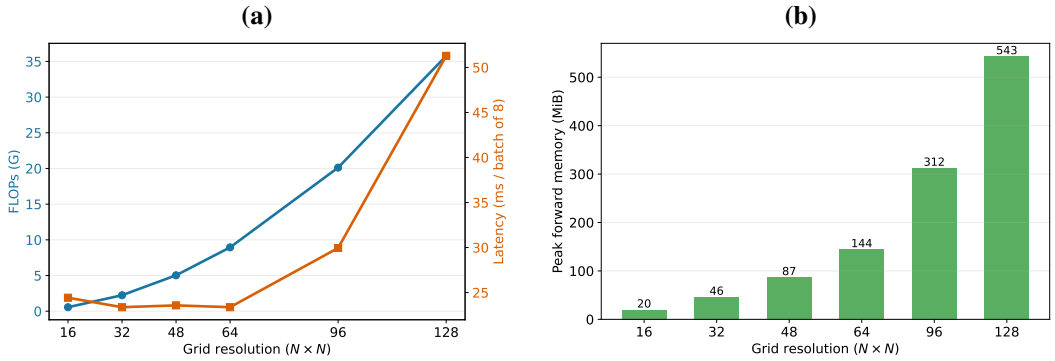

Figure 7: Computational scalability of GA-AMNO. (a) Counted FLOPs and synchronized forward latency at different grid resolutions. (b) Peak allocated forward memory. The same Navier-Stokes checkpoint and batch size 8 are used throughout.

64×64 resolution with batch size 8, the hook-based estimator reports 8.943G counted FLOPs for convolutional and linear operations. The measured forward latency is 23.38 ms per batch and peak allocated forward memory is 144.3 MiB. Latency is measured after 15 warm-up passes and averaged over 30 synchronized inference passes. The FLOP estimate counts a multiply–add as two operations and does not include every FFT, distance, interpolation, normalization, softmax, or tensor-contraction operation; it is therefore reported as an implementation-level estimate rather than an exact hardware-independent operation count.

As shown in Figure 7, the same checkpoint is evaluated from 16×16 to 128×128 without changing model parameters. Inputs are bilinearly resized solely for this computational-scaling study. The counted FLOPs increase from 0.559G at 16×16 to 35.771G at 128×128, while peak forward memory increases from 19.7 MiB to 542.8 MiB. Latency is dominated by launch overhead at resolutions below 64×64 and increases to 51.28 ms per batch at 128×128. Figure 7 is a computational scalability analysis and should not be interpreted as a cross-resolution accuracy experiment.

D Additional Method Details

D.1 Physics-Informed Adaptive Allocation

For the latest input slice $\mathbf{v}$, the physical indicators used by the importance predictor are

$$
\begin{aligned}
Q_g(\mathbf{x}) &= \left[\frac{1}{C}\sum_{c=1}^{C}\left(|D_x v^{(c)}(\mathbf{x})|^2 + |D_y v^{(c)}(\mathbf{x})|^2\right)\right]^{1/2}, \\
Q_\Delta(\mathbf{x}) &= \frac{1}{C}\sum_{c=1}^{C}|D_x D_x v^{(c)}(\mathbf{x}) + D_y D_y v^{(c)}(\mathbf{x})|, \\
Q_E(\mathbf{x}) &= \frac{1}{C}\sum_{c=1}^{C}|v^{(c)}(\mathbf{x})|^2, \\
Q_\omega(\mathbf{x}) &= |D_x v^{(2)}(\mathbf{x}) - D_y v^{(1)}(\mathbf{x})|.
\end{aligned} \tag{18}
$$

Here, $\mathbf{x}$ is a spatial location, $\mathbf{v}$ is the latest input field, C is its number of physical channels, c is the channel index, and $v^{(c)}(\mathbf{x})$ is the value of channel c at $\mathbf{x}$. The operators D_x and D_y are finite-difference derivatives along the two spatial directions. The quantities Q_g, Q_Δ, Q_E, and Q_ω denote the gradient magnitude, Laplacian magnitude, local energy, and vorticity-related response, respectively.

The four indicators provide complementary local information. Specifically, Q_g measures first-order variation, Q_Δ captures second-order local structure, Q_E measures field strength or activity, and Q_ω provides a cue related to rotation or shear. For scalar fields, where the two-component response required by Q_ω is unavailable, the implementation uses the Laplacian-magnitude channel instead.

As defined in Equation 5, the indicators are normalized and concatenated with the learned descriptor $\mathbf{P}$. They serve as explicit conditioning variables rather than an assumed unique or complete description of the governing dynamics. The nonlinear predictor may therefore adjust their relative contributions according to the current input state.

In Equation 6, the mean normalization makes w_i a relative positive importance weight. The lower bound $I_{\min}$ prevents any node from being completely suppressed, while s_I and $\delta_{\max}$ control the importance-modulation range and maximum displacement scale, respectively. Consequently, the indicators directly condition node relocation rather than serving only as auxiliary prediction targets.

D.2 Gauge-Aware Mesh Operator

To motivate source-to-target transport, let $\mathbf{g}_i$ denote the local discretization context at node i and let $\mathbf{z}_i$ denote its latent physical content. We model the encoded feature as

$$\mathbf{h}_i = \mathbf{R}(\mathbf{g}_i)\mathbf{z}_i, \tag{19}$$

where $\mathbf{h}_i$ is the encoded feature at node i, $\mathbf{g}_i$ collects its local discretization variables, $\mathbf{z}_i$ represents the underlying latent physical content, and $\mathbf{R}(\mathbf{g}_i)$ is an invertible representation map associated with that local context.

Under this representation model, the ideal source-to-target transport is

$$\mathbf{T}^{\star}_{i\leftarrow j} = \mathbf{R}(\mathbf{g}_i)\mathbf{R}(\mathbf{g}_j)^{-1}. \tag{20}$$

Here, $\mathbf{T}^{\star}_{i\leftarrow j}$ denotes the ideal transport from source node j to target node i, the superscript $\star$ distinguishes the ideal transport from its learned approximation, and $\mathbf{R}(\mathbf{g}_j)^{-1}$ is the inverse representation map of the source context. The representation maps and local contexts have been defined in Equation 19.

Applying this ideal transport to $\mathbf{h}_j$ gives $\mathbf{R}(\mathbf{g}_i)\mathbf{z}_j$, which expresses the source content under the target representation context. This relation motivates the learned transport but is not imposed as an exact equality on the implemented network.

The learned edge descriptor in Equation 7 includes relative adaptive coordinates, source and target mesh Jacobians, and their state descriptors. It therefore conditions information exchange on both the geometry produced by adaptive allocation and the current encoded state.

The diagonal component in Equation 8 supports efficient channel-wise adjustment, while the gated low-rank component permits cross-channel correction without predicting an unrestricted $C_h \times C_h$ matrix for every edge. Separate prediction heads process $\mathbf{e}_{ij}$ to generate $\mathbf{d}_{ij}^{(\ell)}$, $\mathbf{U}_{ij}^{(\ell)}$, $\mathbf{V}_{ij}^{(\ell)}$, and $\gamma_{ij}^{(\ell)}$.

In Equation 9, transport is applied before the weighted sum. The operator therefore exposes how the source–target geometry conditions each incoming message instead of leaving this dependence implicit in direct aggregation. The residual update in Equation 10 then combines the transported message with the existing target representation.

The learned transformation is intended to reduce representation incompatibility within the considered adaptive-mesh regime. It is not claimed to recover a unique physical frame, represent an exact analytic coordinate transformation, or impose exact Gauge equivariance. The corresponding sufficient conditions, approximation analysis, and deformation-continuity results are presented in Appendix E.

D.3 Mesh-to-Grid Bridge

The gauge-aware operator produces features on adaptive nodes, whereas the output head expects a regular spatial tensor. When the operator and output grids have the same resolution, the bridge assigns distance-based weights to nearby adaptive nodes for each target location $\mathbf{y}_q^r$:

$$\kappa_{qi} = \frac{\exp(-\|\mathbf{x}_i^a - \mathbf{y}_q^r\|_2/\tau_b)}{\sum_{j\in\mathcal{N}_b(q)} \exp(-\|\mathbf{x}_j^a - \mathbf{y}_q^r\|_2/\tau_b)}. \tag{21}$$

Here, κ_{qi} is the reconstruction weight from adaptive node i to target grid point q, $\mathbf{y}_q^r$ is the qth regular-grid coordinate, $\mathcal{N}_b(q)$ is the bridge neighborhood, and τ_b is the temperature controlling distance sensitivity.

The reconstructed feature at grid point q is

$$\mathbf{z}_q = \sum_{i \in \mathcal{N}_b(q)} \kappa_{qi} \widehat{\mathbf{h}}_i, \qquad \widehat{\mathbf{h}}_i = w_i \mathbf{h}_i^{(L_g)}. \tag{22}$$

Here, $\mathbf{z}_q$ is the interpolated grid feature, $\widehat{\mathbf{h}}_i$ is the mean-normalized importance-modulated final nodal feature, w_i is defined in the adaptive-mesh module, L_g is the number of gauge-aware layers, and $\mathbf{h}_i^{(L_g)}$ is the final feature of adaptive node i.

The regular tensor is then obtained by projection and reshaping:

$$\mathbf{F}_{\text{grid}} = \text{Reshape}_{H,W}\left([\rho_b(\mathbf{z}_1), \ldots, \rho_b(\mathbf{z}_M)]\right). \tag{23}$$

Here, ρ_b is a learnable channel projection, H and W are the target grid height and width, and $M = HW$ is the number of regular-grid locations. In the default two-dimensional configuration, the gauge operator uses stride 2 and therefore follows the coarse-grid path: its feature map is bilinearly upsampled and then passed through a channel projection. The distance-weighted bridge above is used when the operator and output grids have equal resolution. We consequently use *mesh-to-grid reconstruction* as the general name for these two implementation paths.

D.4 Spectral and Differential Residual Correction

The prediction head combines local convolutional prediction with spectral and physical refinement. The local solver input is

$$\mathbf{Z}_{\text{sol}} = \text{Concat}(\mathbf{F}_{\text{grid}}, \mathbf{v}), \tag{24}$$

where $\mathbf{Z}_{\text{sol}}$ is the solver input and $\mathbf{v}$ denotes the latest input field for temporal prediction or the conditioning field for steady-state prediction. The local solver first produces $\boldsymbol{\delta}_{\text{raw}} = \mathcal{H}_{\text{local}}(\mathbf{Z}_{\text{sol}})$. The spectral branch receives the complete input history, reconstructed grid features, and canonical coordinates:

$$\begin{aligned} \mathbf{X}_{\text{spec}} &= \text{Concat}(\text{Flatten}_t(\mathbf{a}), \mathbf{F}_{\text{grid}}, \mathbf{X}^r), \\ \mathbf{Z}_{\text{spec}} &= \mathcal{L}_{\text{spec}}(\mathbf{X}_{\text{spec}}), \\ \boldsymbol{\delta}_{\text{spec}} &= s_{\text{spec}} \mathcal{P}_{\text{spec}}(\varphi(\mathcal{N}_{\text{spec}}(\mathcal{C}_{\text{F}}(\mathbf{Z}_{\text{spec}}) + \mathcal{C}_{1\times1}(\mathbf{Z}_{\text{spec}})))). \end{aligned} \tag{25}$$

Here, $\mathcal{L}_{\text{spec}}$ and $\mathcal{P}_{\text{spec}}$ are the spectral lift and output projection, $\mathcal{C}_{\text{F}}$ is a truncated Fourier convolution retaining a prescribed set of low-order modes, $\mathcal{C}_{1\times1}$ is a pointwise spatial path, $\mathcal{N}_{\text{spec}}$ is group normalization, φ is GELU, and $s_{\text{spec}} = 0.2\tanh(\beta_{\text{spec}})$ is a learned bounded scale. This module is therefore a global spectral correction, not an explicit high-pass filter.

The intermediate residual $\boldsymbol{\delta}_1 = \boldsymbol{\delta}_{\text{raw}} + \boldsymbol{\delta}_{\text{spec}}$ is then refined using finite-difference features of its provisional field:

$$\begin{aligned} \mathbf{u}_{\text{base}} &= \mathbf{v} + \boldsymbol{\delta}_1, \\ \mathbf{Z}_{\text{phy}} &= \text{Concat}(\mathbf{v}, \boldsymbol{\delta}_1, D_x \mathbf{u}_{\text{base}}, D_y \mathbf{u}_{\text{base}}, \\ &\qquad\qquad \Delta \mathbf{u}_{\text{base}}, E(\mathbf{u}_{\text{base}})), \\ \boldsymbol{\delta} &= \boldsymbol{\delta}_1 + s_{\text{phy}} \mathcal{H}_{\text{phy}}(\mathbf{Z}_{\text{phy}}). \end{aligned} \tag{26}$$

Here, $\mathcal{H}_{\text{phy}}$ is the convolutional differential-correction network, $s_{\text{phy}} = 0.2\tanh(\beta_{\text{phy}})$ is its learned bounded scale, Δ is the discrete Laplacian, and E is the channel-averaged squared amplitude. For direct-output tasks in which the conditioning and output channel counts differ, $\mathbf{u}_{\text{base}}$ is set to $\boldsymbol{\delta}_1$.

The final prediction is

$$\widehat{\mathbf{u}} = \begin{cases} \mathbf{u}_t + \boldsymbol{\delta}, & \text{for temporal residual prediction,} \\ \boldsymbol{\delta}, & \text{for steady-state prediction or direct output.} \end{cases} \tag{27}$$

In this equation, $\mathbf{u}_t$ is the latest observed state and $\widehat{\mathbf{u}}$ is the predicted future or steady-state field. All modules are trained end to end, with mesh-quality and transport-consistency terms used as auxiliary regularization during training.

E THEORETICAL ANALYSIS OF OPERATOR READABILITY

We formalize *operator readability* relative to a declared family of nondegenerate adaptive discretizations. It has two requirements. First, allocation readability requires the node displacement to be an explicit function of specified input and geometry variables, so that their influence can be inspected or intervened on. Second, interaction readability requires a source message to have the same physical meaning after an admissible change of its local representation convention. The first requirement concerns where computation is allocated; the analysis below focuses on the second requirement, which concerns how the resulting representations are exchanged.

Let $\mathbf{g}_i$ collect the local discretization variables at node i, including local coordinates, mesh Jacobians, and state descriptors. The representation model in Equation 19 adopts the standard gauge-equivariant viewpoint that the same underlying content may have different coordinate representations under changes of local frame (Cohen et al., 2019; de Haan et al., 2021). Here, we adapt this viewpoint to input-dependent discretizations without assuming that the implemented GA-AMNO satisfies exact gauge equivariance. Specifically, $\mathbf{z}_i \in \mathbb{R}^{C_h}$ is fixed latent physical content and $\mathbf{R}(\mathbf{g}_i) \in \mathfrak{G} \subseteq \mathrm{GL}(C_h)$ is invertible. The admissible group $\mathfrak{G}$ contains only representation changes induced by the considered nondegenerate mesh family; it is not identified with every element of $\mathrm{GL}(C_h)$. We assume that $\mathbf{R}$ and $\mathbf{R}^{-1}$ are bounded on this family. Unless the Frobenius norm is explicitly indicated, matrix norms below are induced ℓ_2 norms. This representation model is our formalization for the stated adaptive-discretization setting; it is sufficient for the analysis and does not claim that the frame is unique or directly observable.

Definition 1 (Discretization-induced representation ambiguity). For fixed latent physical content $\mathbf{z}_i$, define its admissible representation set as

$$\mathcal{O}(\mathbf{z}_i) = \{\mathbf{R}(\mathbf{g})\mathbf{z}_i : \mathbf{g} \in \mathfrak{D}\}, \tag{28}$$

where $\mathfrak{D}$ is the declared family of nondegenerate local discretization contexts. Discretization-induced representation ambiguity occurs when $\mathcal{O}(\mathbf{z}_i)$ contains more than one element, so that the same physical content admits different encoded features under different admissible local discretizations. For two contexts $\mathbf{g}_i, \mathbf{g}_i' \in \mathfrak{D}$, the corresponding representations satisfy

$$\mathbf{h}_i' = \mathbf{S}_i\mathbf{h}_i, \qquad \mathbf{S}_i = \mathbf{R}(\mathbf{g}_i')\mathbf{R}(\mathbf{g}_i)^{-1}. \tag{29}$$

We refer to this local nonuniqueness of feature coordinates, with $\mathbf{z}_i$ unchanged, as a hidden gauge ambiguity. This is a modeling description of admissible representation changes, not a claim that the implemented network is exactly gauge equivariant.

Definition 2 (Interaction readability). A directed interaction $j \to i$ satisfies interaction readability if its message depends on the source physical content $\mathbf{z}_j$ and the target context $\mathbf{g}_i$, but not on the chosen admissible source-frame convention used to encode $\mathbf{z}_j$. Equivalently, changing local representation conventions must transform the aggregated message according to the target action $\mathbf{S}_i$, rather than according to the independently varying source actions. An adaptive mesh is operator-readable relative to the declared discretization family when this interaction property is combined with allocation readability.

Theorem 1 (Limitation of shared direct aggregation). Consider the uncorrected linear message

$$\overline{\mathbf{m}}_i = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}\mathbf{W}\mathbf{h}_j, \tag{30}$$

where $\mathbf{W}$ is shared across edges. Consider the subset of local reparameterizations $\mathbf{h}_j' = \mathbf{S}_j\mathbf{h}_j$ that leaves the scalar weights unchanged. If $\overline{\mathbf{m}}_i' = \mathbf{S}_i\overline{\mathbf{m}}_i$ must hold for all source features, then every edge with $\alpha_{ij} \neq 0$ must satisfy

$$\mathbf{W}\mathbf{S}_j = \mathbf{S}_i\mathbf{W}. \tag{31}$$

Consequently, a single shared map cannot guarantee operator-readable aggregation for independently varying local frames.

Proof: The required equality is

$$\sum_j \alpha_{ij}\mathbf{W}\mathbf{S}_j\mathbf{h}_j = \sum_j \alpha_{ij}\mathbf{S}_i\mathbf{W}\mathbf{h}_j. \tag{32}$$

Because the source features can vary independently, each nonzero edge must satisfy Equation 31. For $\mathbf{W} = \mathbf{I}$, this reduces to $\mathbf{S}_j = \mathbf{S}_i$; for invertible $\mathbf{W}$, it imposes the same fixed conjugacy relation on every neighbor. Neither condition accommodates independently changing local discretization frames in general.

Theorem 1 concerns shared uncorrected aggregation, not every irregular-mesh operator. A geometry-conditioned edge map may avoid this limitation; such a map already acts as a form of representation transport.

Theorem 2 (Sufficient condition for operator-readable transport). Let $\alpha_{ij} \geq 0$ and $\sum_j \alpha_{ij} = 1$. If $\mathbf{T}_{i\leftarrow j} = \mathbf{T}^{\star}_{i\leftarrow j}$, with the ideal transport defined in Equation 20, then the transported aggregation satisfies

$$\mathbf{m}_i = \sum_j \alpha_{ij} \mathbf{T}_{i\leftarrow j} \mathbf{h}_j = \mathbf{R}(\mathbf{g}_i) \sum_j \alpha_{ij} \mathbf{z}_j. \tag{33}$$

Moreover, under a local coordinate change $\mathbf{h}'_i = \mathbf{S}_i \mathbf{h}_i$, if $\alpha'_{ij} = \alpha_{ij}$ and

$$\mathbf{T}'_{i\leftarrow j} = \mathbf{S}_i \mathbf{T}_{i\leftarrow j} \mathbf{S}_j^{-1}, \tag{34}$$

then $\mathbf{m}'_i = \mathbf{S}_i \mathbf{m}_i$.

Proof: Substituting Equation 19 and Equation 20 cancels the source-frame factors before summation and gives Equation 33. Under the coordinate change,

$$\mathbf{m}'_i = \sum_j \alpha_{ij} \mathbf{S}_i \mathbf{T}_{i\leftarrow j} \mathbf{S}_j^{-1} \mathbf{S}_j \mathbf{h}_j = \mathbf{S}_i \mathbf{m}_i. \tag{35}$$

Thus, all incoming physical contents are represented in one target frame, and changing local coordinates changes only the coordinates of the target message.

Theorem 2 is an exact sufficient condition. The learned diagonal-plus-low-rank map is not asserted to satisfy it identically. To cover the learned case, define the attention defect and connection defect as

$$\begin{aligned} \delta_{\alpha,i} &= \sum_j |\alpha'_{ij} - \alpha_{ij}|, \\ \varepsilon_{ij} &= \left\| \mathbf{T}'_{i\leftarrow j} \mathbf{S}_j - \mathbf{S}_i \mathbf{T}_{i\leftarrow j} \right\|_2 . \end{aligned} \tag{36}$$

Theorem 3 (Approximate interaction-consistency bound). Assume $\|\mathbf{h}_j\|_2 \leq H$ and $\|\mathbf{T}'_{i\leftarrow j} \mathbf{S}_j\|_2 \leq B$. Then

$$\|\mathbf{m}'_i - \mathbf{S}_i \mathbf{m}_i\|_2 \leq BH\delta_{\alpha,i} + H \sum_j \alpha_{ij} \varepsilon_{ij}. \tag{37}$$

Proof: Add and subtract $\sum_j \alpha_{ij} \mathbf{T}'_{i\leftarrow j} \mathbf{S}_j \mathbf{h}_j$ to obtain

$$\begin{aligned} \mathbf{m}'_i - \mathbf{S}_i \mathbf{m}_i = &\sum_j (\alpha'_{ij} - \alpha_{ij}) \mathbf{T}'_{i\leftarrow j} \mathbf{S}_j \mathbf{h}_j \\ &+ \sum_j \alpha_{ij} (\mathbf{T}'_{i\leftarrow j} \mathbf{S}_j - \mathbf{S}_i \mathbf{T}_{i\leftarrow j}) \mathbf{h}_j. \end{aligned} \tag{38}$$

The triangle inequality and submultiplicativity bound the first sum by $BH\delta_{\alpha,i}$ and the second by $H \sum_j \alpha_{ij} \varepsilon_{ij}$.

The bound separates two failure modes of a learned layer: the scalar neighbor weighting may change, and the feature transport may deviate from connection covariance. It therefore applies even when attention is produced from learned state descriptors rather than assumed to be exactly gauge invariant.

Proposition 1 (Approximation capacity of diagonal-plus-low-rank transport). Let $\mathbf{T}^{\star} \in \mathbb{R}^{C_h \times C_h}$ denote an ideal transport matrix for one edge, and let $\mathbf{D}$ be any admissible diagonal matrix.

Define $\mathbf{E} = \mathbf{T}^\star - \mathbf{D}$, with singular values $\sigma_1(\mathbf{E}) \geq \cdots \geq \sigma_{C_h}(\mathbf{E})$. Then the best rank-$r$ correction satisfies

$$\min_{\mathrm{rank}(\mathbf{L})\leq r} \|\mathbf{T}^\star - (\mathbf{D} + \mathbf{L})\|_F^2 = \sum_{k>r} \sigma_k^2(\mathbf{E}). \tag{39}$$

Consequently, the parameterization $\mathbf{D} + \mathbf{U}\mathbf{V}^\top$ can represent the optimal rank-r approximation of the non-diagonal remainder for the selected $\mathbf{D}$.

Proof: For fixed $\mathbf{D}$, the problem is equivalent to finding the best rank-r approximation to $\mathbf{E}$. Equation 39 follows from the Eckart–Young–Mirsky theorem (Eckart & Young, 1936; Mirsky, 1960), and a truncated singular value decomposition supplies factors $\mathbf{U}$ and $\mathbf{V}$. The edgewise scalar gate and normalization used in the implementation can be absorbed into these unconstrained factors whenever the gate is nonzero.

Proposition 1 is an expressivity statement, not an optimization guarantee. It explains why the diagonal term can model channelwise rescaling while a small rank captures dominant cross-channel frame mixing. The residual tail in Equation 39 contributes to the connection defect ε_{ij} in Theorem 3; the learned edge network is not assumed to attain the best approximation.

Proposition 2 (Topology-preserving deformation stability). Consider two discretizations $\mathcal{X}$ and $\mathcal{X}'$ of the same physical input, with the same node identities and neighborhood topology, and let $\delta_{\mathcal{X}} = \max_i \|\mathbf{x}_i' - \mathbf{x}_i\|_2$. If all geometry-dependent learned features, normalized edge weights, and induced transport matrices are locally Lipschitz functions of the coordinates on a compact family of nondegenerate meshes, then there is a finite L_m such that

$$\|\mathbf{m}_i' - \mathbf{m}_i\|_2 \leq L_m \delta_{\mathcal{X}}. \tag{40}$$

Justification: For the same physical sample, decompose the message difference into feature, weight, and transport changes and apply their local Lipschitz bounds. This gives Equation 40. The statement applies only while connectivity is fixed and cells remain nondegenerate; it does not cover folding or topology changes.

Corollary 1 (Propagation to the operator output). Let Φ_ℓ and Φ_ℓ' denote the ℓth complete feature-update maps under two admissible descriptions of the same physical input, and let $\mathbf{S}_\ell$ denote the corresponding action on layer-ℓ features. Assume

$$\|\Phi_\ell'(\mathbf{S}_{\ell-1}\mathbf{h}) - \mathbf{S}_\ell \Phi_\ell(\mathbf{h})\|_2 \leq b_\ell \tag{41}$$

on the considered compact domain, where b_ℓ can be bounded using Theorem 3 together with the deformation contribution in Proposition 2. Suppose Φ_q' is K_q-Lipschitz. For the complete reconstruction and prediction maps $\mathcal{D}$ and $\mathcal{D}'$, assume that $\mathcal{D}'$ is K_{out}-Lipschitz and

$$\left\|\mathcal{D}'(\mathbf{S}_{L_g}\mathbf{h}) - \mathbf{S}_{\mathrm{out}}\mathcal{D}(\mathbf{h})\right\|_2 \leq b_{\mathrm{out}}. \tag{42}$$

Then the final predictions satisfy

$$\|\widehat{\mathbf{u}}' - \mathbf{S}_{\mathrm{out}}\widehat{\mathbf{u}}\|_2 \leq K_{\mathrm{out}} \sum_{\ell=1}^{L_g} \left(\prod_{q=\ell+1}^{L_g} K_q \right) b_\ell + b_{\mathrm{out}}. \tag{43}$$

Here, $\mathbf{S}_{\mathrm{out}}$ is the prescribed output action; it is the identity after reconstruction to the same canonical grid. An empty product in Equation 43 is defined as one.

Proof: Apply Equation 41 at the first layer and propagate its defect through each later Lipschitz map. Repeating this argument for every layer and using the triangle inequality gives the weighted sum in Equation 43. Applying Equation 42 contributes the final term b_{out}.

Corollary 1 connects local transport quality to final operator stability, but it also exposes the remaining requirements: small edgewise defects alone are insufficient if later mappings have large Lipschitz constants or are incompatible with the canonical output representation. In the implemented architecture, the transport-continuity and loop terms are auxiliary regularizers rather than exact constraints. Moreover, for the low-rank mode the loop term regularizes the diagonal channel-scaling component only; no exact holonomy claim is made for the full diagonal-plus-low-rank transport.

This analysis is performed on the induced matrix $\mathbf{T}_{i\leftarrow j}$, not on the individual diagonal and low-rank factors. It establishes the limitation of shared direct aggregation, an exact sufficient condition for

operator-readable transport, an approximation-capacity result for the implemented parameterization, and local-to-output bounds for the learned approximate case. It does not claim unique recovery of a physical frame or exact invariance of the implemented network. The intervention and deformation studies in Section 4 examine whether the learned connection is functionally used and remains stable, while the message–target discrepancy reported there is treated as a compatibility diagnostic rather than a direct measurement of the latent covariance defect in Equation 36.

F Expanded Results

F.1 Targeted Component Ablations

We remove one component at a time while retaining the remaining training protocol and evaluate each ablation through the failure mode that the removed component is intended to address. All comparisons use frozen checkpoints. To make quantities with different scales visually comparable, Figure 8 reports the ratio between the Full GA-AMNO error and the corresponding w/o-module error. The dashed line at one is the ablated reference; a value below one means that the complete model is better.

1) Adaptive Mesh: difficult-region allocation. We use the target-free high-demand masks defined in Appendix C and measure regional relative ℓ_2 and gradient errors. This test asks whether adaptive allocation improves the regions that exhibit strong input variation and nonuniform geometric demand, rather than whether it uniformly changes the whole field.

2) Spectral Residual: spectral correction. We jointly measure full-field relative ℓ_2 error and high-band spectral-energy error. The high band contains Fourier coefficients whose normalized radial frequency is at least 0.6 of the maximum represented radius. Although the implemented Fourier convolution retains truncated low-order modes rather than applying an explicit high-pass mask, this diagnostic tests whether the complete spectral-plus-pointwise correction empirically improves frequency content that is difficult for the mesh-domain prediction alone.

3) Differential Correction: local-structure repair. We measure gradient error and the dataset-specific differential residual used in the evaluation pipeline. For Darcy, the latter is the Laplacian discrepancy defined in Appendix C, not a claim of evaluating the complete coefficient-weighted equation residual. These metrics match the derivative features supplied to the correction head.

4) Low-rank Gauge Transport: geometry-conditioned message correction. We measure high-demand regional relative ℓ_2 error together with residual message–target discrepancy after transport. For the compatibility test, Full GA-AMNO and w/o Low-rank Gauge Transport receive the same GA-AMNO adaptive mesh and identical smooth perturbations, isolating the low-rank correction from mesh allocation.

All bars in Figure 8 are below one. Adaptive allocation reduces difficult-region errors by approximately 1%–12%. The spectral correction gives the largest targeted effect, reducing relative ℓ_2 and high-band errors by 45%–99%; this is an empirical ablation result rather than a claim that the branch is structurally restricted to high frequencies. Differential correction reduces its two local-structure diagnostics by 10%–15%. Low-rank Gauge Transport reduces high-demand prediction error by 4%–13% and residual message–target discrepancy by 39%–53%. The significance of this experiment is the observed module–failure-mode correspondence: removing a component primarily degrades the quantity that the component was designed to control. This correspondence shows that the architecture is not merely an undifferentiated stack of accuracy modules and links Physics-Informed Adaptive Allocation and Low-rank Gauge Transport to their intended computational roles. It does not imply that every component lowers every global metric on every dataset.

F.2 Smooth Adaptive-Mesh Perturbations

We next test whether the learned message-correction mechanism remains stable after controlled changes to the internally learned mesh. Starting from mesh G, we construct

$$G_{k,\ell} = G + \ell \eta d_k(G), \qquad \ell \in \{0, 0.25, 0.50, 0.75, 1.00\}, \tag{44}$$

where k indexes one of six smooth boundary-tapered transformations, ℓ is the perturbation strength, d_k is the displacement field, and η equals 12% of the mean edge length. The transformations com-

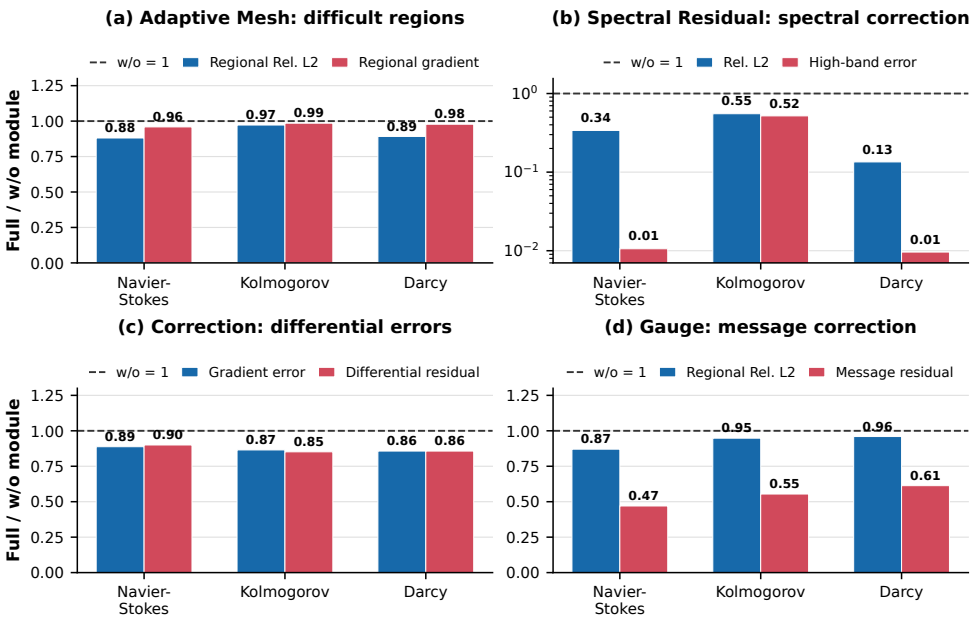

Figure 8: Targeted component ablations using frozen checkpoints. Each bar is the Full GA-AMNO metric divided by the corresponding w/o-module metric; lower is better and the dashed line denotes the ablated reference. Adaptive Mesh is evaluated in high-demand regions, Spectral Residual through field and high-band spectral errors, Differential Correction through gradient and differential errors, and Low-rank Gauge Transport through regional prediction and residual message–target discrepancy. Panel (b) uses a logarithmic vertical axis.

Table 3: Mean message-compatibility gain over six smooth transformations and four nonzero strengths.

Dataset	GA-AMNO (%)	Adaptive Mesh (%)	Difference
Navier-Stokes	**83.15**	64.13	+19.02
Kolmogorov	**48.05**	6.24	+41.81
Darcy	**39.52**	1.35	+38.17

prise smooth horizontal translation, diagonal translation, rotation, sine-modulated rotation, cosine-modulated rotation, and a sine–cosine twist. Connectivity is fixed and every evaluated mesh has a positive cell Jacobian.

Full GA-AMNO is compared with the trained adaptive-mesh ablation without Low-rank Gauge Transport. Both models receive the same GA-AMNO adaptive mesh and the same perturbation, isolating the effect of Low-rank Gauge Transport from resource allocation.

As shown in Table 3, GA-AMNO has higher message-compatibility gain in all 72 dataset–transformation–strength combinations. It also has lower prediction error than the adaptive-mesh ablation in all 24 perturbation conditions on Navier-Stokes and Kolmogorov, but not on Darcy. Moreover, it is not uniformly less sensitive according to a target-free output-change metric. Unlike the previous experiment, which analyzes mismatch already present in the learned mesh, this controlled perturbation test changes the geometry while holding the physical sample and mesh allocation shared between the two models. It therefore verifies that the advantage of low-rank transport persists under multiple smooth, topology-preserving changes of the discretization and is not only a correlation observed on the original mesh. The result supports deformation-robust interaction correction, rather than universal invariance to arbitrary geometry changes.

Table 4: Message–target compatibility on high-geometric-mismatch edges. Lower E_{src} and E_{msg} indicate smaller discrepancy, whereas a larger positive Gain indicates a stronger improvement produced by low-rank Gauge transport.

Dataset	E_{src}	E_{msg}	Gain (%)
Navier–Stokes	0.052823	0.009543	81.93
Kolmogorov	0.127806	0.066045	48.32
Darcy	0.287031	0.123934	56.82

Table 5: Geometric quality of the learned adaptive mesh.

Dataset	Inversion Rate	Min. Angle	Aspect Ratio
Navier-Stokes	0.0000	89.019°	1.0036
Rayleigh-Bénard	0.0000	84.308°	1.0360
Kolmogorov	0.0000	89.716°	1.0038
Darcy	0.0000	89.669°	1.0014

F.3 High-Mismatch Message Compatibility

To complement the mismatch-stratified results in Figure 6, we report the absolute compatibility measurements for the high-geometric-mismatch group in Table 4. Here, E_{src} measures the discrepancy between the untransported source feature and the target representation, while E_{msg} measures the discrepancy after source-to-target transport. The compatibility gain is computed as

$$\mathrm{Gain} = \frac{E_{\mathrm{src}} - E_{\mathrm{msg}}}{E_{\mathrm{src}}} \times 100\%.$$

Transport consistently lowers message–target discrepancy, with gains ranging from 48.32% to 81.93%. These results provide the absolute values underlying the stratified comparison in the main text.

F.4 Mesh Quality

Finally, Table 5 verifies that the learned allocation is realized on valid geometries. KS is omitted because it is one-dimensional. All evaluated two-dimensional meshes have zero inversion rate, minimum cell angles above 84°, and aspect ratios close to one. This experiment rules out a degenerate explanation in which prediction gains are obtained through folded cells, invalid neighborhoods, or extreme element anisotropy. These quantities are geometric feasibility controls rather than evidence of operator readability by themselves; the counterfactual and transport experiments above establish whether the valid adaptive geometry and its interactions are functionally meaningful.

Taken together, the experiments support a layered conclusion rather than relying on prediction error alone. The main comparison establishes operator utility; component ablation assigns improvements to intended failure modes; importance counterfactuals test allocation readability; edge intervention and mismatch stratification test whether representation correction is both consequential and geometrically targeted; controlled deformation evaluates whether that mechanism persists when the discretization changes; and mesh diagnostics exclude invalid geometry as an explanation. This chain of evidence supports the central claim that operator readability increases when both node allocation and cross-node interaction can be inspected and functionally tested. It does not establish exact gauge covariance, unique recovery of latent physical frames, or invariance to arbitrary external meshes.

F.5 Cross-Resolution Generalization

Table 6 evaluates the proposed model under resolution changes. The model is trained at the canonical 64×64 resolution and directly evaluated at resolutions from 16 to 128. Only relative ℓ_2 error is reported to focus on the overall reconstruction robustness under discretization changes.

The lowest error is consistently obtained at the native resolution. Nevertheless, the model remains accurate at nearby resolutions. For example, the Navier-Stokes error is 0.013443 at resolution 48 and 0.009215 at resolution 96, compared with 0.003219 at the native resolution. Similar trends are observed for Rayleigh-Bénard convection, Kuramoto–Sivashinsky, and Kolmogorov flow.

Table 6: Cross-resolution generalization of GA-AMNO. The model is trained at 64×64. Relative ℓ_2 error is reported; lower is better.

Dataset	16	32	48	64	96	128
Navier-Stokes	0.175885	0.041609	0.013443	**0.003219**	0.009215	0.012195
Rayleigh-Bénard	0.202487	0.068442	0.033256	**0.000680**	0.019584	0.020235
KS	0.172811	0.077675	0.045117	**0.000387**	0.013415	0.007378
Kolmogorov	0.431518	0.169014	0.064861	**0.018796**	0.058290	0.076400
Darcy	0.351837	0.132896	0.061258	**0.027743**	0.083502	0.113239

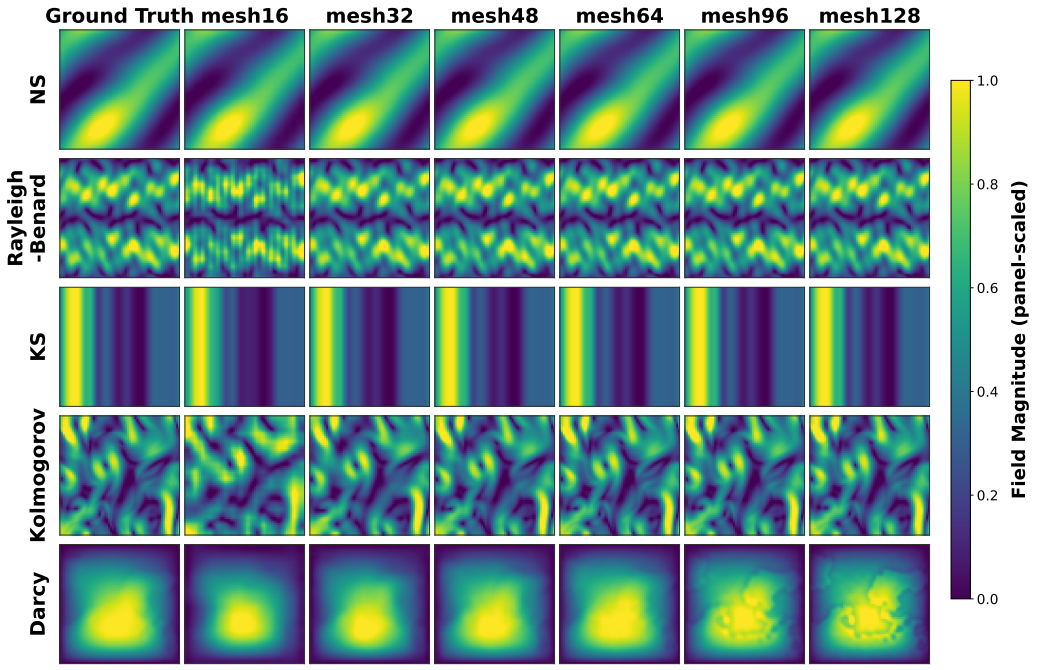

Figure 9: GA-AMNO predictions under cross-resolution evaluation. Columns show the ground truth and predictions at mesh resolutions 16, 32, 48, 64, 96, and 128; rows correspond to the five PDE benchmarks. The native training resolution is 64, and colors are independently scaled within each panel.

The error increases more noticeably at extremely coarse resolutions, especially for Kolmogorov flow and Darcy flow. This behavior is expected because 16×16 sampling cannot represent localized vortices, sharp gradients, or heterogeneous permeability patterns adequately. The results indicate that the learned gauge-aware representation is stable under moderate resolution shifts, while its accuracy remains fundamentally tied to the information available in the input discretization.

To complement the quantitative results in Table 6, Figure 9 shows representative predictions across the evaluated resolutions. The visualizations confirm the same resolution-dependent pattern: severe downsampling removes localized and high-frequency structures, whereas resolutions close to or finer than the training resolution preserve the dominant field morphology. Because each panel is scaled independently, the figure is intended to illustrate structural preservation rather than absolute-amplitude agreement or resolution invariance.

F.6 Rollout Stability

Table 7 reports autoregressive rollout results of GA-AMNO. Darcy flow is not included because it is a static PDE operator-learning problem. We report prediction errors at one, five, and ten rollout steps.

For Navier-Stokes, the relative ℓ_2 error increases from 0.003213 at one step to 0.030052 after ten steps. Rayleigh-Bénard convection remains particularly stable, with a ten-step error of only

Table 7: Autoregressive rollout results of GA-AMNO. Lower is better. 'Spec.' denotes the overall spectral-energy error $\mathrm{SpectrumError}$.

Dataset	Step	Rel. L2	RMSE	Spec.
Navier-Stokes	1	0.003213	0.002604	0.001566
Navier-Stokes	5	0.016251	0.014324	0.011963
Navier-Stokes	10	0.030052	0.027607	0.024469
Rayleigh-Bénard	1	0.000692	0.000085	0.000195
Rayleigh-Bénard	5	0.004747	0.000612	0.001478
Rayleigh-Bénard	10	0.012066	0.001595	0.004401
KS	1	0.000407	0.000223	0.000339
KS	5	0.002140	0.001146	0.001753
KS	10	0.004134	0.002108	0.003505
Kolmogorov	1	0.018522	0.018065	0.006738
Kolmogorov	5	0.100077	0.097531	0.064080
Kolmogorov	10	0.253117	0.247677	0.165009

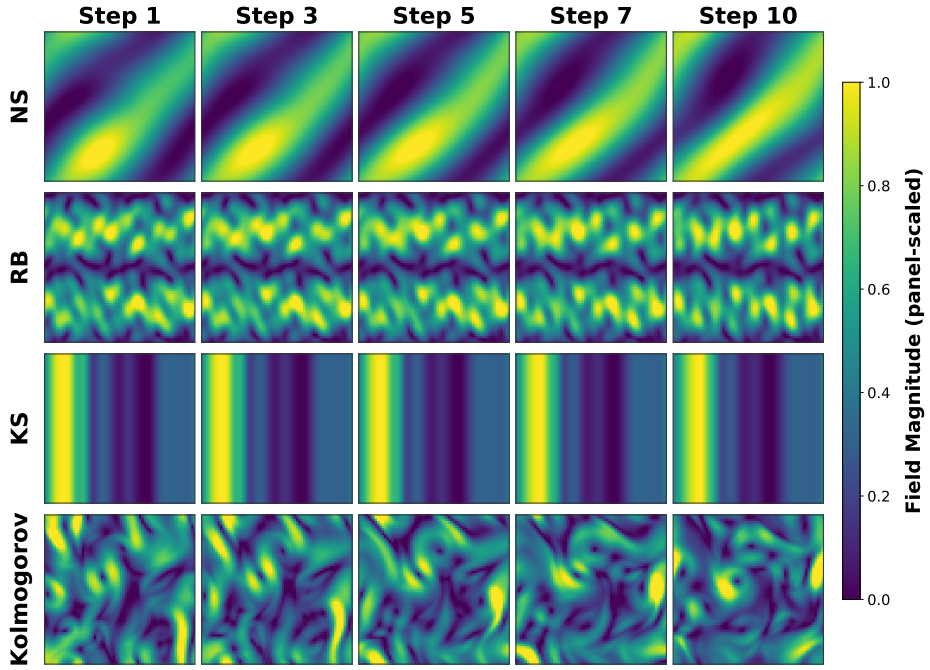

Figure 10: Selected GA-AMNO autoregressive predictions at rollout steps 1, 3, 5, 7, and 10 for Navier-Stokes, Rayleigh-Bénard, KS, and Kolmogorov. Colors are independently scaled within each panel, and the one-dimensional KS profile is repeated vertically for visualization.

0.012066. Kuramoto–Sivashinsky also maintains a low ten-step error of 0.004134, despite its chaotic temporal evolution. Kolmogorov flow has the largest long-horizon error, increasing from 0.018522 to 0.253117, which reflects the accumulation of phase and vortex-location errors in forced turbulent dynamics.

These results suggest that the proposed model preserves stable temporal evolution for moderate rollout horizons. The differential residual correction helps suppress local high-frequency drift, while Physics-Informed Adaptive Allocation and Low-rank Gauge Transport reduce the propagation of representation mismatch across time. The remaining degradation on Kolmogorov flow highlights that strongly forced turbulence remains challenging for purely autoregressive neural operators.

The qualitative rollouts in Figure 10 provide a visual counterpart to the error growth reported in Table 7. Navier-Stokes, Rayleigh-Bénard, and KS retain their dominant structures over the displayed horizon, whereas Kolmogorov flow exhibits increasingly visible phase and vortex-location deviations. This behavior is consistent with its faster quantitative error accumulation. Darcy is excluded because it defines a steady coefficient-to-solution mapping rather than a temporal rollout task.

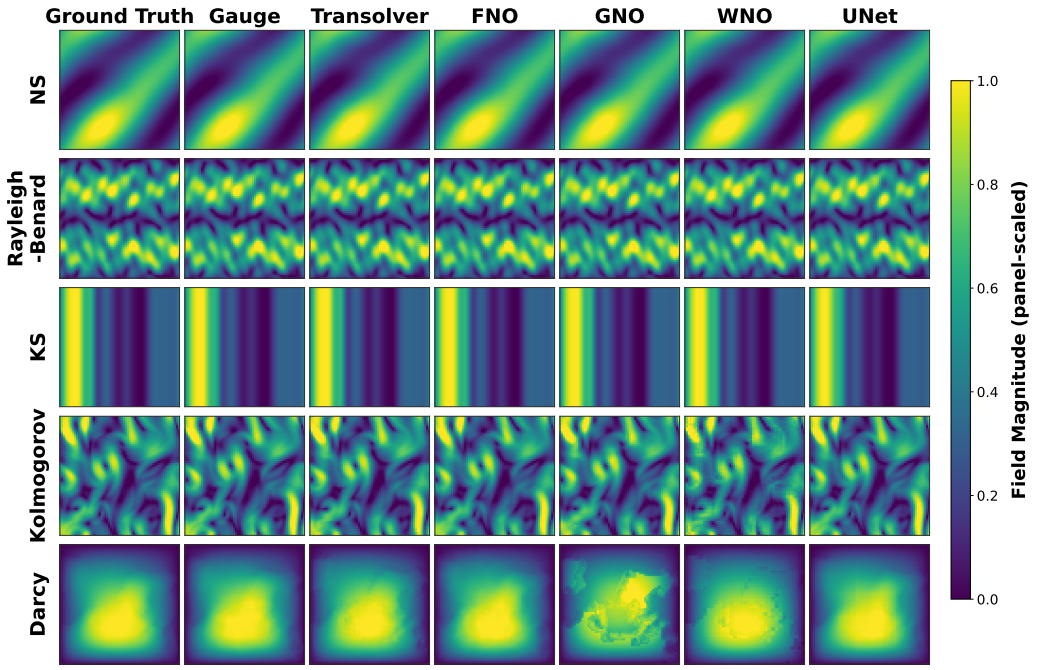

Figure 11: Representative native-resolution predictions. Columns compare the ground truth, GA-AMNO, Transolver, FNO, GNO, WNO, and UNet; rows correspond to Navier-Stokes, Rayleigh-Bénard, KS, Kolmogorov, and Darcy. Colors are independently scaled within each panel to emphasize spatial structure.

F.7 Qualitative Comparison at Native Resolution

Figure 11 complements the aggregate errors in Table 1 with one representative prediction at the native evaluation resolution. GA-AMNO preserves the dominant spatial structures across all five PDE families, including localized flow patterns and heterogeneous Darcy responses. Each panel is scaled independently to $[0, 1]$ for structural comparison, so color should not be interpreted as a shared physical amplitude across models or datasets. The one-dimensional KS profile is repeated vertically only to maintain a common field-map layout; quantitative conclusions remain those of Table 1.

G Notation

The following notation summarizes the symbols used in the method, evaluation, and theoretical analysis. Superscripts r and a denote reference and adaptive discretizations, respectively. A prime denotes a quantity after an admissible change of local representation convention.

PDE fields and operator mappings

Ω	Spatial domain.
d	Spatial dimension.
$\mathcal{F}_{\mathrm{PDE}}$	Governing differential operator.
$\boldsymbol{\mu}$	Physical parameters, coefficient fields, or forcing terms.
$\mathcal{A}$	Input function space of the PDE solution operator.
$\mathcal{U}$	Output solution function space of the PDE solution operator.
$\mathbf{a}$	Temporal input history or steady conditioning field.
$\mathcal{U}_t$	Input history ending at time t.
$\mathbf{v}$	Latest input state or conditioning field.
$\mathbf{u}$	Ground-truth physical field.
$\widehat{\mathbf{u}}$	Predicted physical field.
$\mathcal{G}^\dagger$	Ground-truth PDE solution operator.
$\mathcal{G}_\theta$	Learned neural operator parameterized by θ.
	Adaptive discretization and state encoding
$\mathcal{X}^r$	Reference discretization or canonical grid.
$\mathbf{X}^r$	Canonical coordinate tensor used by the grid-based branches.
$\mathcal{X}^a$	Input-dependent adaptive discretization.
$\mathbf{x}_i^r$	Coordinate of reference node i.
$\mathbf{x}_i^a$	Coordinate of adaptive node i.
$\mathbf{P}$	Encoded state descriptor or physics-proxy feature map.
$\mathbf{p}_i$	State descriptor at node i.
Q_g	Gradient-magnitude physical indicator.
Q_Δ	Laplacian-magnitude physical indicator.
Q_E	Local energy indicator.
Q_ω	Vorticity-related indicator.
$\widetilde{\mathbf{Q}}(\mathbf{x})$	Concatenated normalized physical-indicator vector at location $\mathbf{x}$.
$I(\mathbf{x})$	Predicted scalar importance map at spatial location $\mathbf{x}$.
$\mathcal{N}(\cdot)$	Per-sample channel-normalization operator.
η_θ	Convolutional importance-map predictor.
σ	Sigmoid activation used for importance prediction.
	Mesh dimensions and adaptive movement

N	Number of nodes or scalar field entries, according to context.
C	Number of physical channels.
C_h	Hidden feature width.
I_i	Learned importance value at node i.
w_i	Positive mean-normalized importance weight.
$\mathbf{o}_i$	Bounded raw displacement direction.
$\Delta\mathbf{x}_i$	Importance-modulated node displacement.
$\delta_{\max}$	Maximum allowed displacement magnitude.
ψ_θ	Convolutional head that predicts the bounded raw node displacement.
$I_{\min}$	Positive lower bound used in importance modulation.
s_I	Scale controlling the range of importance modulation.
Π_Ω	Projection that keeps displaced nodes inside the spatial domain Ω.
$\mathbf{J}_i^a$	Local adaptive-mesh Jacobian.
	Gauge-connection parameterization
$\mathbf{e}_{ij}$	Edge descriptor for directed edge $j \to i$.
r	Rank of the low-rank transport correction.
$\mathbf{d}_{ij}^{(\ell)}$	Diagonal transport factors at layer ℓ.
$\mathbf{U}_{ij}^{(\ell)}$	Left low-rank transport factor.
$\mathbf{V}_{ij}^{(\ell)}$	Right low-rank transport factor.
$\gamma_{ij}^{(\ell)}$	Gate controlling the low-rank correction.
$\mathbf{T}_{i\leftarrow j}^{(\ell)}$	Learned source-to-target feature transport.
	Attention and transported aggregation
$a_{ij}^{(\ell)}$	Unnormalized edge-attention score.
$\alpha_{ij}^{(\ell)}$	Softmax-normalized edge-attention weight.
$\mathbf{h}_i^{(\ell)}$	Feature of node i at layer ℓ.
$\mathbf{m}_i^{(\ell)}$	Aggregated transported message at node i.
$\mathcal{N}(i)$	Neighborhood of node i.
$\rho_\theta^{(\ell)}$	Learned residual feature-update network at layer ℓ.
L_g	Number of low-rank gauge transport layers.
ϵ	Positive numerical-stabilization constant.
	Mesh-to-grid reconstruction

$\mathbf{y}_q^r$	Coordinate of canonical output point q.
$\mathcal{N}_b(q)$	Adaptive-node neighborhood used for reconstruction.
κ_{qi}	Interpolation weight from adaptive node i to output point q.
τ_b	Distance temperature of the mesh-to-grid bridge.
$\mathbf{z}_q$	Reconstructed feature at canonical point q.
$\mathbf{F}_{\mathrm{grid}}$	Canonical grid feature tensor.
	Residual prediction and differential operators
$\boldsymbol{\delta}_{\mathrm{raw}}$	Residual predicted by the local branch.
$\boldsymbol{\delta}_{\mathrm{spec}}$	Residual predicted by the spectral branch.
$\boldsymbol{\delta}_1$	Sum of local and spectral residuals.
$\mathbf{u}_{\mathrm{base}}$	Provisional physical field used for differential correction.
$\mathbf{Z}_{\mathrm{phy}}$	Input tensor of the differential correction head.
$\boldsymbol{\delta}$	Final corrected residual.
D_x	Discrete derivative in the first spatial direction.
D_y	Discrete derivative in the second spatial direction.
Δ	Discrete Laplacian operator.
$\mathcal{F}_{\mathrm{DFT}}$	Orthonormal discrete Fourier transform.
	Feature-compatibility evaluation
$\mathcal{E}_b$	Directed-edge set in geometric-mismatch group b.
$E_{\mathrm{src}}^{(b)}$	Source-to-target feature discrepancy in group b.
$E_{\mathrm{msg}}^{(b)}$	Transported-message-to-target discrepancy in group b.
$G_{\mathrm{comp}}^{(b)}$	Relative compatibility gain produced by transport.
$\mathbf{g}_i$	Local discretization context at node i.
$\mathbf{z}_i$	Latent physical content at node i.
$\mathbf{R}(\mathbf{g}_i)$	Representation map induced by local context $\mathbf{g}_i$.
	Local representation analysis
$\mathfrak{G}$	Admissible group of local representation changes.
$\mathfrak{D}$	Family of admissible nondegenerate discretization contexts.
$\mathcal{O}(\mathbf{z}_i)$	Admissible representation set of physical content $\mathbf{z}_i$.
$\mathbf{S}_i$	Local representation action at node i.
$\overline{\mathbf{m}}_i$	Message produced by uncorrected direct aggregation.
$\mathbf{W}$	Shared linear map in direct aggregation.
	Connection consistency and ideal transport

$\delta_{\alpha,i}$	Attention-consistency defect at node i.
ε_{ij}	Connection-consistency defect on edge $j \to i$.
$\mathbf{T}^{\star}_{i\leftarrow j}$	Ideal source-to-target transport matrix.
$\mathbf{T}^{\star}$	Ideal transport matrix for a representative edge in the approximation analysis.
$\mathbf{D}$	Diagonal component of an ideal transport.
$\mathbf{E}$	Non-diagonal remainder $\mathbf{T}^{\star} - \mathbf{D}$.
$\sigma_k(\mathbf{E})$	The kth singular value of $\mathbf{E}$.
	Mesh and operator stability constants
$\delta_{\mathcal{X}}$	Maximum coordinate change between two meshes.
L_m	Local Lipschitz constant for message variation.
Φ_ℓ	Complete feature-update map at layer ℓ.
b_ℓ	Representation-consistency defect at layer ℓ.
K_ℓ	Lipschitz constant of layer ℓ.
$\mathcal{D}$	Canonical reconstruction and prediction map.
K_{out}	Lipschitz constant of the output map.
$\mathbf{S}_{\text{out}}$	Prescribed representation action on the output.
b_{out}	Representation-consistency defect of the output map.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing and limited programming support. All scientific ideas, methodology, experimental design, implementation decisions, analysis, and conclusions were developed and verified by the authors. All AI-assisted content was carefully reviewed and revised by the authors. We take full responsibility for the final content of this work.