METRO: Metric-Enhanced Token Routing Operator

Nodens Koren
Department of Computer Science
ETH Zürich
Zürich, Switzerland
nodens.koren@inf.ethz.ch

Thomas Hofmann
Department of Computer Science
ETH Zürich
Zürich, Switzerland
thomas.hofmann@inf.ethz.ch

Georgios Kissas
Swiss Data Science Center
ETH Zürich
Zürich, Switzerland
georgios.kissas@sdsc.ethz.ch

Abstract

State-of-the-art neural operators scale to complex meshes via *slice-and-process* architectures, yet many rely on linear compatibility scores for latent tokenization. Under common feature normalization, such scores are equivalent to isotropic Euclidean clustering, while without normalization they induce unbounded linear decision regions. In both cases, they lack slice-specific anisotropic locality, which can lead to redundant and entangled latent slices. To address this, we propose **Metric-Enhanced Token Routing Operator (METRO)**, a geometry-aware routing mechanism that replaces linear projection with a learnable Mahalanobis metric. By enabling each latent slice to learn a local anisotropic tensor, METRO shapes receptive fields into exponentially localized, oriented ellipsoids that naturally align with flow features like boundary layers and wakes. As a drop-in replacement, METRO yields consistent improvements across both Transformer and Mamba backbones. Empirically, our method achieves substantial performance gains on irregular domains, outperforming baselines on both standard PDE benchmarks and complex industrial design tasks. Finally, METRO exhibits enhanced robustness in out-of-distribution regimes across varying Reynolds numbers and geometric configurations.

1 Introduction

Industrial design workflows in sectors such as aerospace, automotive, energy, and turbomachinery rely on solving Partial Differential Equations (PDEs) repeatedly for varying system configurations. While traditional solvers offer high accuracy, they are computationally slow, making them impractical for rapid prototyping, uncertainty quantification, and real-time control (Lu et al., 2021; Li et al., 2021). In light of these bottlenecks, deep learning algorithms (surrogate models) have recently been explored to predict the behavior of physical systems, with a specific focus on neural operators that learn the solution operator of PDEs directly.

Neural operators (Kovachki et al., 2023; Li et al., 2023a; Azizzadenesheli et al., 2024; Kissas et al., 2022; Seidman et al., 2022; Lingsch et al., 2024a; Lu et al., 2021) have evolved from solving proof-of-concept problems to serving as accurate, scalable surrogates increasingly adopted in complex engineering workflows, from climate modeling to structural design (Bonev et al., 2023; Wu et al., 2024; Azizzadenesheli et al., 2024; Tiwari et al., 2025). To achieve broader adoption in large-scale industrial applications, such as aerodynamic shape optimization, structural analysis, and turbomachinery, they need to: (i) process complex geometries without relying on naive interpolation

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

to regular grids, which disregards the underlying geometric structure and the physical nature of the problem; (ii) scale to a massive number of degrees of freedom to accommodate the high-resolution discretizations required for industrial fidelity; (iii) capture long-range interactions, as complex physical phenomena like wakes often span the entire domain; and (iv) capture anisotropic information propagation, where features such as boundary layers and shocks are thin in one direction yet distinctively elongated in another.

A common strategy to address these challenges is to *compress* the input field to a much smaller set of latent tokens, perform global mixing in this latent space, and *decode* back to the original resolution. Existing methods differ mainly in how the mesh-to-latent coupling is defined (e.g., local neighborhood integration (Wen et al., 2025; Li et al., 2023b) versus global soft routing (Hu et al., 2025; Luo et al., 2025)). One prominent approach in this compress-process-decode family is the so-called "slice-and-process" method. Despite their success, current slice-based operators rely on simplified geometric assumptions that limit their expressivity. For instance, the original Transolver employs dot-product attention, which defines linear decision boundaries. As we derive in Section 3.2, under standard feature normalization, dot-product routing is equivalent, up to constants, to nearest-prototype assignment under an isotropic Euclidean distance: it clusters points based on Euclidean proximity to a prototype, effectively favoring spherical neighborhoods in the learned feature space. While subsequent works (Luo et al., 2025) have improved computational efficiency, this limitation in anisotropic local tokenization has yet to be addressed. Consequently, there remains a critical gap for a mechanism that is *locally adaptive*, allowing different latent slices to learn distinct anisotropic metric tensors in the learned feature space.

Motivated by these insights, we propose **Metric-Enhanced Token Routing Operator (METRO)**, a principled generalization of the latent-attention paradigm. METRO endows neural operators with *learnable anisotropic metrics* that shape each attention head's receptive field into an exponentially localized ellipsoid, rather than an unbounded hyperplane or a rigid sphere. Specifically, we replace the standard isotropic dot-product in the attention routing with a Mahalanobis distance tailored to the underlying data. This yields *geometry-aware attention*: each latent slice learns its own local metric tensor, allowing it to attend to points along specific orientations while suppressing off-axis or outlier features. The metric is learned end-to-end, providing the model with the flexibility to discover the optimal anisotropy for each physical region.

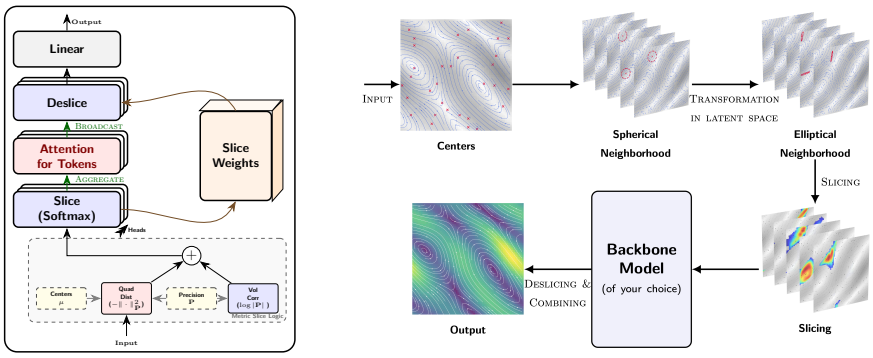
In summary, our key contributions are:

- Task-Specialized Local Routing: We introduce a Mahalanobis-based routing mechanism where each latent slice learns its own distinct anisotropic metric. Unlike global feature reweighting approaches, this enables the model to create a structured latent partition where different slices can independently specialize to capture varying physical modes—such as highly anisotropic boundary layers or isotropic diffusion—within the same system.
- High-Efficiency Anisotropic Tokenization: By effectively shaping the receptive fields to match complex geometries, our method captures physical phenomena using significantly fewer latent slices. Under strict parameter-matched conditions, this compact representation yields both lower error rates and faster training times compared to isotropic baselines.
- Model-Agnostic Framework: Our approach is a modular upgrade compatible with both Transformer and state-space model (SSM) backbones, providing richer geometric awareness without sacrificing linear scalability.

Beyond these technical contributions, METRO aims to broaden the applicability of latent operator networks. Our empirical results highlight substantial accuracy gains across diverse PDE systems, a nearly 50% reduction in volumetric prediction error on the AirfRANS (Bonnet et al., 2022) benchmark, and superior generalization in aerodynamic design tasks.

2 Literature review

Neural operators and PDE solvers. Surrogate models for physics simulations on irregular domains can be broadly categorized based on their strategy for handling geometric discretization. *Transformation-based approaches*, including Geo-FNO (Li et al., 2023a), Diffeomorphic Neural Operators (Ahmad et al., 2024), Optimal Transport Neural Operators (Li et al., 2025), and DIMON (Yin et al., 2024), map non-uniform meshes into regular latent grids to leverage scalable backbones



(a) Pipeline (b) Workflow Visualization

Figure 1: **Overview of the METRO Architecture.** (**a**) The processing pipeline: Input features are mapped to centers, transformed by the metric, sliced, and processed. (**b**) Visual demonstration: Comparing standard spherical neighborhoods (Iso) to flow-aligned elliptical neighborhoods (Aniso), showing how the model explicitly attends to local physical structures.

such as FNO (Li et al., 2021). However, these methods rely on diffeomorphisms that are often too topologically restrictive for complex geometries, or on optimal transport maps that may yield non-smooth, non-invertible coordinates across varying instances. *Graph-based methods*, such as GNO (Li et al., 2020), GINO (Li et al., 2023b), and MeshGraphNet (Pfaff et al., 2021), instead utilize graph embeddings to process irregular meshes directly or map them to global representations. Despite their utility, these approaches typically incur high computational costs due to multiple message-passing rounds, the need for multi-scale receptive fields, and the hardware inefficiencies associated with unstructured memory access. *Direct methods*, such as GNOT (Hao et al., 2023) and arbitrary-grid FNOs (Lingsch et al., 2024b), operate on the physical domain without intermediate mappings. While mathematically robust, operating on the full physical discretization often hinders efficient scaling to high-resolution three-dimensional problems. Finally, *embedding approaches*—such as the slicing mechanism in Transolver (Wu et al., 2024; Luo et al., 2025; Hu et al., 2025) or continuous neural representations (Serrano et al., 2023; Yin et al., 2023)—offer high flexibility and scalability by treating geometry as unstructured tokens. While they bypass rigid topological constraints, existing methods fail to explicitly preserve intrinsic geometric properties.

Anisotropic Routing. Anisotropic metrics have also been explored in attention and mixture-of-experts architectures. Nielsen et al. (2025) propose an adaptive clustering router that uses cluster-specific feature weights derived from within-cluster dispersion to select specialized experts. While sharing the motivation of incorporating anisotropic geometry into routing, our work addresses a different setting: constructing latent representations for neural operators on irregular discretizations. Rather than relying on empirical cluster statistics, METRO learns slice-specific centers and diagonal precision parameters end-to-end through the operator prediction objective. These parameters define dense, geometry-aware assignments for normalized feature aggregation and reconstruction, rather than sparse expert selection.

3 Methodology

In this section, we formalize latent operator learning, analyze the geometric limitations of previously proposed approaches, and introduce our core contribution: *Metric-Enhanced Token Routing Operator (METRO)*. More specifically, we demonstrate that linear routing in slicing operations induces an isotropic partition of the latent space under the standard normalization assumptions. To address this, we propose a learnable anisotropic metric. We define the resulting geometry by interpreting the learned latent metric as inducing a pullback pseudometric on the physical domain, a formulation that simplifies computation.

3.1 Problem Setup and Notation

Neural operators learn an approximate map between infinite-dimensional function spaces $\mathcal{A} \to \mathcal{U}$. We consider a physical domain Ω discretized by a (non-)uniform mesh with N points, and an input embedding $\mathbf{x}_n \in \mathbb{R}^D$ at point n, obtained from an encoder $\phi : \Omega \times \mathbb{R}^{d_{in}} \to \mathcal{Z} \subset \mathbb{R}^D$ acting on an input field and geometry. Crucially, all routing operations are performed in this learned feature space rather than directly in physical coordinates. The "Slice-and-Process" paradigm (Wu et al., 2024) compresses these N points into G latent tokens (slices), where $G \ll N$. These tokens are processed via a neural backbone and then projected back to the original resolution:

$$\begin{aligned}
&\text{Slicing} \qquad \mathbf{z}_g = \frac{\sum_{n=1}^{N} A_{n,g}\mathbf{x}_n}{\sum_{n=1}^{N} A_{n,g}} \in \mathbb{R}^D, \quad g = 1, ..., G \\
&\text{Process} \qquad (\mathbf{z}'_1, ..., \mathbf{z}'_G) = \mathcal{G}(\mathbf{z}_1, ..., \mathbf{z}_G), \\
&\text{de-Slicing} \qquad \mathbf{x}'_n = \sum_{g=1}^{G} A_{n,g}\mathbf{z}'_g \in \mathbb{R}^D, \quad n = 1, ..., N.
\end{aligned}$$

Here $A_{n,g} \geq 0$ represents the routing weights, satisfying $\sum_g A_{n,g} = 1$ for each n, and $\mathcal{G}$ denotes an arbitrary backbone, such as Transformer (Hu et al., 2025; Luo et al., 2025) or Mamba (Tiwari et al., 2025). METRO modifies the routing mechanism as detailed in Section 3.3.

3.2 Revisiting Linear Slicing: The Isotropy Equivalence

Wu et al. (2024) proposed a slicing mechanism where routing weights are derived from a scalar compatibility score $S_{n,g}$ via a Softmax operator:

$$A_{n,g} = \text{Softmax}_g\left(S_{n,g}\right), \tag{1}$$

In the original Transolver, this score is a dot product $S_{n,g} = \frac{\mathbf{x}_n^\top \boldsymbol{\mu}_g}{\tau}$, where $\tau > 0$ is the temperature, utilizing learnable prototypes $\boldsymbol{\mu}_g \in \mathbb{R}^D$. This scoring induces linear pairwise assignment boundaries. Under the normalized/equal-norm interpretation below, these boundaries correspond to an isotropic Euclidean Voronoi partition.

Under standard feature normalization—such as LayerNorm (Ba et al., 2016) or an explicit ℓ_2 normalization—maximizing this dot-product is equivalent, up to a constant, to minimizing the isotropic Euclidean distance: $\|\mathbf{x} - \boldsymbol{\mu}\|_2^2 = \|\mathbf{x}\|_2^2 + \|\boldsymbol{\mu}\|_2^2 - 2\mathbf{x}^\top\boldsymbol{\mu}$. Given the normalization regime where $\|\mathbf{x}\|_2^2 \approx \text{const}$ and $\|\boldsymbol{\mu}\|_2^2 \approx \text{const}$, using the score $\mathbf{x}^\top\boldsymbol{\mu}$ is functionally equivalent to using the distance $\|\mathbf{x} - \boldsymbol{\mu}\|_2^2$.

This reveals two distinct views of the original routing mechanism. Without feature normalization, the linear dot-product score induces linear decision boundaries (i.e., unbounded hyperplanes). Under standard normalization, maximizing the dot-product becomes equivalent to minimizing an isotropic Euclidean distance, leading to spherical clustering. What is common to both views is that linear routing does not induce a localized, anisotropic interior region. In contrast, the proposed Mahalanobis score defines ellipsoidal level sets with exponential locality. Since information flow in physical systems is often highly preferential—such as in boundary layers or wakes—this motivates the need for an anisotropic routing geometry.

3.3 Geometry-Aware Anisotropic Slicing

To induce anisotropic similarity within the slice-process-deslice paradigm, we replace the dot-product score with a log-Gaussian density score. Each slice g is parameterized by a center $\boldsymbol{\mu}_g \in \mathbb{R}^D$ and a positive-definite precision matrix $\mathbf{P}_g \in \mathbb{R}^{D\times D}$. We define the geometry-aware score as:

$$S_{n,g} = -\frac{1}{2}\|\mathbf{x}_n - \boldsymbol{\mu}_g\|_{\mathbf{P}_g}^2 + \frac{1}{2}\log\det(\mathbf{P}_g), \tag{2}$$

where $\|\mathbf{x}\|_{\mathbf{P}}^2 = \mathbf{x}^\top\mathbf{P}\mathbf{x}$ denotes the squared Mahalanobis norm. Substituting this into Eq. 1 yields weights proportional to (up to a per-n normalization):

$$A_{n,g} \propto \sqrt{\det(\mathbf{P}_g)} \cdot \exp\left(-\frac{1}{2}\|\mathbf{x}_n - \boldsymbol{\mu}_g\|_{\mathbf{P}_g}^2\right). \tag{3}$$

For a fixed g, the quadratic form $(\mathbf{x}_n - \boldsymbol{\mu}_g)^\top \mathbf{P}_g (\mathbf{x}_n - \boldsymbol{\mu}_g)$ defines ellipsoidal level sets determined by the eigenspectrum of $\mathbf{P}_g$. This allows the router to *stretch and compress* the receptive field around a point in an explicitly learned, anisotropic manner. The ellipsoids are defined over the latent coordinates, and the encoder decides how these correspond to coordinates in the physical space. While $\mathbf{P}_g$ may be full rank, we adopt a diagonal parameterization:

$$\mathbf{P}_g = \mathrm{diag}(p_{g,1}, \ldots, p_{g,D}), \quad p_{g,i} > 0. \tag{4}$$

We ensure positive definiteness by parameterizing $p_{g,i} = \mathrm{softplus}(\hat{p}_{g,i}) + \varepsilon$, where $\varepsilon > 0$. While this choice implicitly forces the ellipsoids to be axis-oriented in the latent coordinates, we show next that these axis-aligned ellipsoids induce receptive fields in the physical coordinates through the pullback that are oriented via the encoder map.

The term $\frac{1}{2} \log\det(\mathbf{P}_g)$ acts as the Gaussian normalization constant; algebraically, it balances the softmax logits to ensure that slices with higher variance (lower precision) do not artificially dominate the routing weights.

3.4 Theoretical Connections: Pullback Pseudometrics and Oriented Receptive Fields

METRO can be interpreted as learning localized kernels in the feature space. Let $\mathbf{u} \in \mathbb{R}^m$ denote the input coordinates and $\mathbf{x} = \phi(\mathbf{u}) \in \mathbb{R}^D$ their learned embedding. For each slice g, the latent Mahalanobis geometry induces a pullback pseudometric on the physical coordinate space through the encoder ϕ. We define the pullback bilinear form

$$g_{\mathrm{pb}}^{(g)}(\mathbf{u})(\mathbf{v}, \mathbf{w}) = \langle J_\phi(\mathbf{u})\mathbf{v}, \mathbf{P}_g J_\phi(\mathbf{u})\mathbf{w} \rangle \,, \tag{5}$$

where $J_\phi(\mathbf{u})$ is the Jacobian of ϕ, $\mathbf{P}_g$ is the slice precision matrix, and $\langle \cdot, \cdot \rangle$ is the Euclidean inner product in $\mathbb{R}^D$. This form is always positive semidefinite, and becomes positive definite when $J_\phi(\mathbf{u})$ has full column rank. Thus, even when $\mathbf{P}_g$ is diagonal in latent coordinates, the induced local geometry in physical coordinates can be non-diagonal through the encoder Jacobian, producing oriented receptive fields. We formalize this local statement below.

Proposition 3.1 (Local orientation via pullback pseudometric)**.** *Let $\phi : \mathbb{R}^m \to \mathbb{R}^D$ be twice differentiable, and let $\mathbf{P} \in \mathbb{R}^{D \times D}$ be a diagonal matrix with strictly positive diagonal entries. Define*

$$q(\mathbf{u}) = \mathbf{r}(\mathbf{u})^\top \mathbf{P} \mathbf{r}(\mathbf{u}), \qquad \mathbf{r}(\mathbf{u}) = \phi(\mathbf{u}) - \boldsymbol{\mu}. \tag{6}$$

Then, for any $\mathbf{u}_0 \in \mathbb{R}^m$, the Hessian of q evaluated at $\mathbf{u}_0$ satisfies

$$\nabla^2 q(\mathbf{u}_0) = 2 J_\phi(\mathbf{u}_0)^\top \mathbf{P} J_\phi(\mathbf{u}_0) + 2 \sum_{k=1}^{D} P_{kk} r_k(\mathbf{u}_0) \nabla^2 \phi_k(\mathbf{u}_0). \tag{7}$$

In particular, if $\phi(\mathbf{u}_0) = \boldsymbol{\mu}$, i.e. $\mathbf{r}(\mathbf{u}_0) = \mathbf{0}$, then

$$\nabla^2 q(\mathbf{u}_0) = 2 J_\phi(\mathbf{u}_0)^\top \mathbf{P} J_\phi(\mathbf{u}_0). \tag{8}$$

In Appendix B, we provide a continuous-kernel interpretation of the Mahalanobis slicing score and its discrete approximation. Proposition 3.1 shows that the local second-order geometry of the latent Mahalanobis distance is governed by the pullback form $J_\phi^\top \mathbf{P}_g J_\phi$ at points mapped to the slice center. Away from the center, the additional Hessian term in Eq. (7) captures higher-order distortion from the nonlinear encoder. Therefore, axis-aligned ellipsoids in latent coordinates can induce locally oriented receptive fields in physical coordinates. We empirically measure the alignment between the principal axis of this induced local geometry and the local streamline direction in Appendix G.3; the alignment increases systematically with anisotropy, supporting the interpretation that elongated slices adapt to physically meaningful directional features.

4 Experiments

To evaluate the proposed METRO framework, we conduct extensive experiments across diverse physical domains, ranging from standard PDE benchmarks to large-scale industrial design tasks. We demonstrate that integrating our geometry-aware anisotropic slicing into Transolver and LaMO backbones (denoted as **M-Transolver** and **M-LaMO**) consistently enhances performance across both structured and unstructured discretizations.

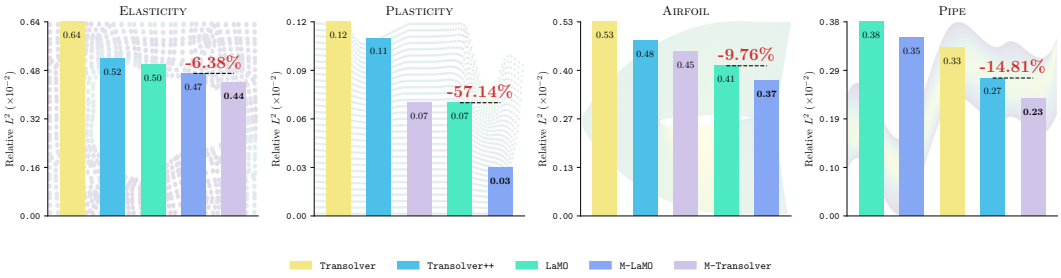


Figure 2: M-Transolver consistently outperforms baselines across complex geometries (Elasticity, Plasticity, Airfoil Pipe), achieving an average **23.4% relative error reduction**.

Table 1: Quantitative results on industrial benchmarks AirfRANS (unstructured RANS) and AirCraft (3D external aerodynamics). We report relative L_2 error (↓ lower is better) and Spearman's rank correlation ρ_L (↑ higher is better).

Model	AirfRANS				AirCraft					
	Volume ↓	Surf ↓	C_L ↓	ρ_L ↑	C_p ↓	ρ ↓	U ↓	V ↓	W ↓	Pressure ↓
GraphSAGE (Hamilton et al., 2017)	0.0087	0.0184	0.1476	0.9964	0.1442	0.0803	0.0148	0.2154	0.2395	0.0923
PointNet (Qi et al., 2017)	0.0253	0.0996	0.1973	0.9919	0.1472	0.0783	0.0147	0.2104	0.2528	0.0937
Graph U-Net (Gao & Ji, 2019)	0.0076	0.0144	0.1677	0.9949	0.1475	0.0784	0.0152	0.2225	0.2477	0.0947
MeshGraphNet (Pfaff et al., 2021)	0.0214	0.0387	0.2252	0.9945	0.1372	0.0682	0.0164	0.1835	0.2175	0.0893
GNO (Li et al., 2020)	0.0269	0.0405	0.2016	0.9938	0.1556	0.0785	0.0167	0.2012	0.2318	0.1071
Galerkin (Cao, 2021)	0.0074	0.0159	0.2336	0.9951	0.0990	0.0429	0.0110	0.1304	0.1077	0.0480
Geo-FNO (Li et al., 2023a)	0.0361	0.0301	0.6161	0.9257	0.2586	0.1227	0.0300	0.3014	0.2757	0.1432
GNOT (Hao et al., 2023)	0.0049	0.0152	0.1992	0.9942	0.0824	0.0407	0.0092	0.1141	0.1164	0.0436
GINO (Li et al., 2023b)	0.0297	0.0482	0.1821	0.9958	0.1513	0.0636	0.0160	0.1501	0.1431	0.0853
Transolver (Wu et al., 2024)	0.0037	0.0142	0.1030	0.9978	*0.0748*	*0.0379*	*0.0091*	*0.1107*	*0.1016*	0.0439
Transolver++ (Luo et al., 2025)	0.0025	0.0084	0.1228	0.9976	0.0784	0.0390	0.0122	0.1318	0.1048	0.0445
LaMO (Tiwari et al., 2025)	*0.0020*	*0.0066*	*0.0977*	*0.9983*	0.0786	0.0400	0.0099	0.1234	0.1094	*0.0438*
M-Transolver (Ours)	0.0014	0.0061	0.0528	0.9991	**0.0692**	**0.0353**	**0.0060**	**0.0919**	**0.0899**	**0.0389**
M-LaMO (Ours)	**0.0010**	**0.0053**	**0.0495**	**0.9994**	0.0734	0.0358	0.0075	0.1029	0.0953	0.0405
Relative Promotion	**+50.0%**	**+19.7%**	**+49.3%**	/	**+7.5%**	**+6.9%**	**+34.1%**	**+17.0%**	**+11.5%**	**+11.2%**

4.1 Experimental Setup

Benchmarks. We evaluate performance on two categories of datasets. **Standard PDE Benchmarks:** Following (Wu et al., 2024; Tiwari et al., 2025), we evaluate on six standard problems: *Airfoil*, *Darcy Flow*, *Navier-Stokes (NS)*, *Plasticity*, *Elasticity*, and *Pipe*. **Industrial Design Benchmarks:** We employ *AirfRANS* (Bonnet et al., 2022), *AirCraft* (Luo et al., 2025), and *ShapeNet-Car* (Chang et al., 2015). For AirfRANS and ShapeNet-Car, we additionally measure the Spearman's rank correlation of lift (C_L) and drag (C_D) coefficients to assess design utility. Full quantitative results for ShapeNet-Car and standard PDE benchmarks are provided in Appendix A. Detailed dataset specifications are provided in Appendix F.

Baselines. We compare METRO against a comprehensive suite of more than ten baseline methods, including frequency-based operators (e.g., FNO (Li et al., 2021)) and state-of-the-art Transformer and State-Space Model operators such as Transolver (Wu et al., 2024), Transolver++ (Luo et al., 2025), and LaMO (Tiwari et al., 2025). For the complete enumeration and configuration of all baselines, we refer readers to Appendix F.5.

4.2 Results on Standard Benchmarks

We first verify the generality of our method by substituting linear slicing with Mahalanobis metrics across standard benchmarks. **Full quantitative results are reported in Table 5 (Appendix).** Two key trends emerge from this evaluation. *Enhancement on Irregular Domains:* M-Transolver achieves consistent gains over the original Transolver baseline. Notably, on problems defined over irregular domains (Airfoil, Elasticity, Plasticity, Pipe), our method outperforms all baselines, achieving an average 23.4% relative error reduction. *Backbone Agnosticism:* Improvements are observed for both M-Transolver and M-LaMO, indicating that anisotropic tokenization complements both Transformer and SSM architectures.

Table 2: Generalization performance on Out-of-Distribution (OOD) tasks for the AirfRANS dataset. We evaluate extrapolation capabilities on unseen Reynolds numbers and Angles of Attack. ↓ indicates lower is better; ↑ indicates higher is better (Spearman's rank correlation).

Model	OOD Reynolds		OOD Angles	
	$C_L \downarrow$	$\rho_L \uparrow$	$C_L \downarrow$	$\rho_L \uparrow$
Simple MLP	0.6205	0.9578	0.4128	0.9572
GraphSAGE (Hamilton et al., 2017)	0.4333	0.9707	0.2538	0.9894
PointNet (Qi et al., 2017)	0.3836	0.9806	0.4425	0.9784
Graph U-Net (Gao & Ji, 2019)	0.4664	0.9645	0.3756	0.9816
MeshGraphNet (Pfaff et al., 2021)	1.7718	0.7631	0.6525	0.8927
GNO (Li et al., 2020)	0.4408	0.9878	0.3038	0.9884
Galerkin (Cao, 2021)	0.4615	0.9826	0.3814	0.9821
GNOT (Hao et al., 2023)	0.3268	0.9865	0.3497	0.9868
GINO (Li et al., 2023b)	0.4180	0.9645	0.2583	0.9923
Transolver (Wu et al., 2024)	0.2996	*0.9896*	*0.1500*	0.9950
Transolver++ (Luo et al., 2025)	0.2897	0.9856	0.2269	0.9914
LaMO (Tiwari et al., 2025)	*0.2713*	0.9887	0.1505	*0.9958*
M-Transolver (Ours)	**0.1761**	0.9950	**0.0960**	0.9977
M-LaMO (Ours)	0.1988	**0.9969**	0.1015	**0.9980**

4.3 Results on Industrial Benchmarks

The results on industrial benchmarks (Table 1) demonstrate the distinct advantage of our method in handling complex geometries.

1. **Complex Wake Resolution via Anisotropy (Volume Pressure).** M-LaMO reduces the Volume Pressure error on AirfRANS by nearly 50% compared to the best baseline. We attribute this improvement to the model's learnable anisotropy. In external aerodynamics, the volumetric domain is dominated by elongated, highly directional structures such as turbulent wakes. Isotropic spherical clusters struggle to cover these features efficiently, often capturing irrelevant free-stream background data. By allowing latent slices to stretch into flow-aligned ellipsoids, METRO efficiently tracks these directional wakes, significantly improving volumetric accuracy.

Boundary Layer Isolation via Exponential Locality (Surface & Lift). The Surface error and Lift Coefficient (C_L) serve as proxies for resolving the microscopically thin boundary layer. M-Transolver reduces Surface error by 32.2% and C_L error by 48.7%. We hypothesize that this is driven by the exponential metric decay induced by the quadratic Mahalanobis distance. In high-dimensional latent spaces, isotropic routing can allocate diffuse neighborhoods that mix near-wall features with less relevant surrounding flow. The quadratic term introduces an exponential metric-locality bias, encouraging surface-focused attention heads to assign lower weight to metric-distant regions and helping preserve the high-gradient signals required for accurate lift prediction.

3. Vector Field Sensitivity (AirCraft). On the 3D AirCraft dataset, we observe substantial gains in the vertical (V, $+17.0\%$) and spanwise (W, $+11.5\%$) velocity components. These fields often represent subtle, directional secondary flows (e.g., wingtip vortices). The learnable anisotropy appears to allow the model to align receptive fields with these specific flow directions, rather than averaging them out with a rotationally invariant dot product.

4. Practical Utility for Design Optimization. Beyond reconstruction, the *Spearman's rank correlation* (ρ_L) for lift reaches 0.9994 on AirfRANS. In industrial design loops, the ability to correctly *rank* design candidates is often more valuable than absolute numerical precision. This suggests M-Transolver effectively captures the geometric sensitivities required for gradient-based shape optimization.

4.4 Generalization to Out-of-Distribution Physics

To assess robustness, we adopt the Out-of-Distribution (OOD) protocol from (Wu et al., 2024), evaluating performance on AirfRANS under unseen Reynolds numbers and Angles of Attack.

1. OOD Reynolds Numbers: Robustness via Metric Locality. We evaluate flow regimes with Reynolds numbers outside the training distribution. M-Transolver achieves a relative L_2 error of 0.1761, a substantial improvement over Transolver (0.2996) and Transolver++ (0.2897). We

hypothesize that this robustness is partly due to the *exponential decay* in learned feature distance. Physically, increasing the Reynolds number reduces the thickness of the boundary layer and introduces high-frequency local turbulence (Schlichting & Gersten, 2016). Isotropic routing mechanisms can more easily mix localized high-frequency structures with broader flow patterns seen during training. By introducing a metric-locality bias, our method encourages the model to rely on feature-near interactions that appear more stable across regimes, rather than on dataset-specific long-range correlations that may fail at higher Reynolds numbers.

2. OOD Angles of Attack: Alignment via Anisotropy. For extrapolation to unseen geometric orientations, M-Transolver achieves a relative error of 0.0960, significantly outperforming Transolver (0.1500) and LaMO (0.1505). We attribute this to the geometric adaptability of the precision matrix $\mathbf{P}_g$. While isotropic kernels are rotationally invariant, they are inefficient at tracking directional features (like wakes) that rotate relative to the grid. To cover a rotated wake, an isotropic kernel must indiscriminately expand its radius. In contrast, our results suggest that the anisotropic mechanism successfully learns to rotate its principal axes in tandem with the flow features. This allows the model to maintain a tightly aligned receptive field around the wake structure, regardless of its global orientation.

4.5 Discussion and Ablation Analysis

Our ablation studies (Table 3) reveal a consistent trend across both datasets: while removing either the quadratic distance term or anisotropy leads to moderate degradation, the model's performance remains significantly stronger than the Transolver baseline. However, removing *both* components causes the error to regress close to the baseline, indicating that each geometric mechanism contributes a distinct but complementary inductive bias. We interpret these results through the lens of attention geometry and physical structure.

1. Complementary Geometry on AirfRANS. AirfRANS features two-dimensional flows dominated by sharp boundary layers and surface-driven pressure distributions. Here, we observe that removing the quadratic term increases *surface* error disproportionately (from 0.0061 to 0.0090), while removing anisotropy increases *volume* error (from 0.0014 to 0.0023). These trends reflect the dual roles of our metric components. *Anisotropy* enables elliptical slicing aligned with thin shear layers and streamline directions, improving accuracy in modeling internal flow structure. *Quadratic Decay* biases attention toward metric-near features, reducing long-range interference—especially important for accurately predicting near-wall scalar fields like pressure. Notably, the full model achieves better-than-additive gains: combining both mechanisms yields over 60% improvement in volume and surface errors compared to Transolver. Removing both components collapses this advantage, supporting the importance of their joint use for high-aspect-ratio physics.

2. Locality Dominance in AirCraft. The AirCraft dataset involves fully three-dimensional geometries with complex topologies (e.g., separate wings, tail, fuselage). In this setting, the quadratic term emerges as crucial for predicting velocity components (U, V, W). This suggests that in irregular 3D domains, a *metric-locality bias* is useful not just for resolution, but for preventing *geometric cross-talk*—ensuring that tokens on the wing do not erroneously attend to the tail simply due to linear alignment. Meanwhile, anisotropy provides incremental benefits by capturing global shape effects such as wing sweep. Together, the two mechanisms contribute orthogonal benefits: directionality shapes the receptive field, while the Mahalanobis norm enforces a learned locality bias.

4.6 Case Study: Visualizing Learned Metric Routing

Having established the quantitative superiority of M-Transolver, we now revisit the geometric hypotheses from Section 1. To examine whether these gains are associated with spatial disentanglement and anisotropic alignment, we visualize the internal representations under three distinct physical regimes.

4.6.1 Hypothesis 1: From Isotropic Entanglement to Spatial Partitioning

Our first premise was that isotropic attention forces *mode collapse*, where multiple slices redundantly attend to the same global features. We test this on the Darcy Flow benchmark (Fig. 3).

Visualizing Disentanglement. As hypothesized, the standard Transolver exhibits severe spatial redundancy, with the majority of slices collapsing into identical spherical blobs centered on the domain's highest-energy region. In contrast, M-Transolver breaks this symmetry (Fig. 3b). The

Table 3: Unified Ablation Study on Geometric Slicing Components. We evaluate the contribution of the Quadratic term (Exponential Locality) and Anisotropy (Directionality) across both the **AirfRANS** (2D) and **AirCraft** (3D) datasets. The "w/o All" variant removes both the quadratic distance term and anisotropic precision structure within our metric-routing implementation, yielding a linear isotropic routing variant; the Transolver reference is reported separately.

Model Variant	AirfRANS				AirCraft (Relative L_2 Error ↓)					
	Volume ↓	Surf ↓	C_L ↓	ρ_L ↑	C_p	ρ	U	V	W	P
M-Transolver (Full)	**0.0014**	**0.0061**	**0.0528**	**0.9991**	**0.0692**	**0.0353**	**0.0060**	**0.0919**	**0.0899**	**0.0389**
w/o Anisotropy	0.0023	0.0066	0.0752	0.9988	0.0702	0.0358	0.0071	0.0987	0.0984	0.0402
w/o Quadratic Term	0.0018	0.0090	0.0780	0.9989	0.0718	0.0363	0.0073	0.1024	0.0984	0.0411
w/o All Components	0.0049	0.0105	0.1159	0.9974	0.0791	0.0386	0.0093	0.1125	0.0997	0.0448
Transolver (Ref.)	*0.0037*	*0.0142*	*0.1030*	*0.9978*	*0.0748*	*0.0379*	*0.0091*	*0.1107*	*0.1016*	*0.0439*

learned slices effectively partition the physical domain with minimal overlap, evolving from rigid spheres into flexible geometric primitives—elongated ellipsoids and boundary-hugging contours. This supports the interpretation that the Mahalanobis metric helps the model align with the porous medium structure rather than simply imposing a global grid.

4.6.2 Hypothesis 2: Robustness in Chaotic Regimes

We further argued that a metric-locality bias is useful to prevent feature entanglement in high-entropy systems. We examine this using the Navier-Stokes benchmark at the chaotic timestep $T = 20$.

Separation vs. Noise. As fluid dynamics evolve into chaos, the baseline representation degrades into global noise, covering the domain indiscriminately (Fig. 3c). M-Transolver, however, maintains *local separation* even in this highly non-linear regime. The learned metric-locality bias acts as a stabilizer, preventing attention heads from "leaking" into the chaotic background and ensuring that individual slices track distinct vortices without entanglement.

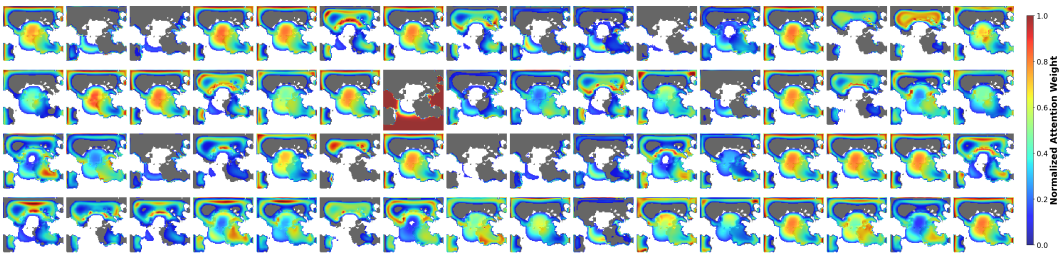
4.6.3 Hypothesis 3: The Reality of Learned Anisotropy

Finally, we investigate the geometric mechanism by visualizing the learned precision matrix $\mathbf{P}_s$ on the AirCraft dataset. We compare M-Transolver against an ablated variant where the quadratic term ($\mathbf{x}^\top \mathbf{P}\mathbf{x}$) is removed.

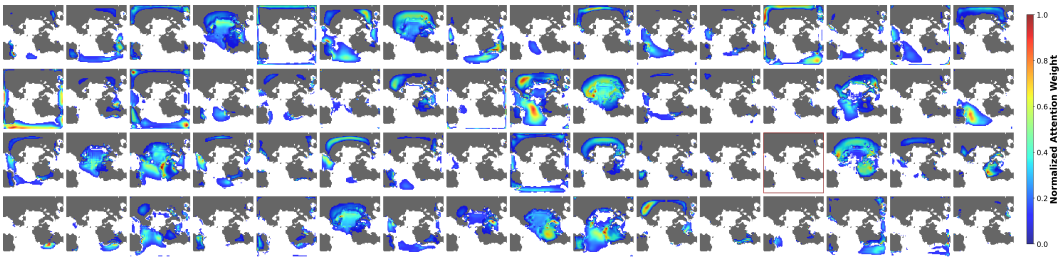
Heatmap Analysis. The heatmap for the ablated model (Fig. 4) displays distinct horizontal banding. This indicates that while the learned precision scales vary *between* slices, they remain uniform *within* each slice ($\mathbf{P}_s \approx \alpha_s \mathbf{I}$).

Mathematical Interpretation: The Hierarchy of Expressivity. This ablation study establishes a clear hierarchy of geometric adaptation:

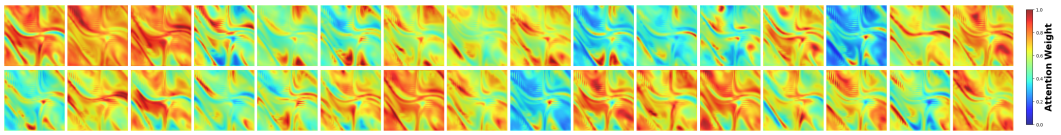
- **Baseline (Fixed Linear):** Standard Transolver uses linear attention ($\mathbf{q}^\top \mathbf{k}$) with a fixed global temperature ($1/\sqrt{d}$), forcing all slices to operate at the same scale, limiting flexibility.
- **Ablated (Adaptive Linear):** By removing the quadratic term but keeping the learnable $\mathbf{P}_s$, the model reverts to linear attention ($\mathbf{x}^\top \mathbf{P}_s \boldsymbol{\mu}$), but with a crucial advantage. The "banding" behavior confirms it functions as an *Adaptive Temperature* mechanism ($\alpha_s \mathbf{x}^\top \boldsymbol{\mu}$), where distinct slices learn different sensitivities—sharpening focus for fine details or flattening it for global trends. This conceptually aligns with the motivation behind the explicit local adaptive temperature in Transolver++ (Luo et al., 2025), though our method achieves this adaptivity implicitly through the precision matrix rather than a separate heuristic.
- **Full Model (Anisotropic Metric-Local):** The full model (Fig. 4) restores the quadratic term ($\mathbf{x}^\top \mathbf{P}_s \mathbf{x}$), closing the geometry from infinite hyperplanes into exponentially localized *ellipsoids*. The high-variance texture in the heatmap indicates that the metric deforms along specific axes. This adds the critical missing capability: *Metric Locality* (suppressing metric-distant noise) and *Directional Alignment* (hugging flow streamlines), which neither the fixed linear baseline nor the adaptive linear ablation can achieve.



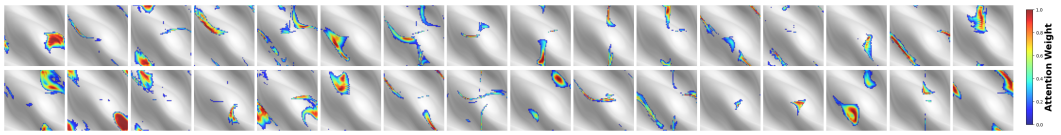
(a) Darcy Flow: Standard Transolver (Baseline)



(b) Darcy Flow: M-Transolver (Ours)



(c) Navier-Stokes: Standard Transolver at $T = 20$



(d) Navier-Stokes: M-Transolver at $T = 20$

Figure 3: **Learned Attention Topology on Darcy and Navier-Stokes Flows. Darcy Flow (a–b):** Standard Transolver exhibits substantial slice redundancy, with many attention regions concentrated near the domain center. In contrast, M-Transolver learns diverse anisotropic regions that better cover the spatial domain. **Navier-Stokes (c–d):** At $T = 20$, standard Transolver exhibits broadly overlapping attention regions, while M-Transolver maintains more localized and spatially differentiated assignments.

5 Conclusion

In this work, we identified and addressed a geometric mismatch in neural operators: the disparity between predominantly isotropic tokenization mechanisms and the highly anisotropic, irregular structures that arise in many physical systems. We proposed **Metric-Enhanced Token Routing Operator (METRO)**, a modular framework that reformulates the tokenization process as a learnable anisotropic latent clustering problem. By encouraging *metric locality* via quadratic constraints and *directionality* via learnable anisotropy, METRO generates receptive fields that naturally align with physical structures such as boundary layers and wakes. Furthermore, by effectively shaping the receptive fields to match underlying physical structures, METRO achieves a highly compact latent representation. This token efficiency allows the model to capture complex geometries using significantly fewer latent slices, resulting in both lower error rates and faster training times under strict parameter-matched conditions. Our extensive evaluation on industrial benchmarks demonstrates that these contributions are not merely additive but synergistic: METRO achieves a 43% error reduction averaged across physical metrics on AirfRANS, sets a new state-of-the-art in OOD extrapolation, and consistently enhances performance across both Transformer and State-Space Model backbones.

References

Ahmad, Z., Chen, S., Yin, M., Kumar, A., Charon, N., Trayanova, N., and Maggioni, M. Diffeomorphic latent neural operators for data-efficient learning of solutions to partial differential equations. *arXiv preprint arXiv:2411.18014*, 2024.

Azizzadenesheli, K., Kovachki, N., Li, Z., Liu-Schiaffini, M., Kossaifi, J., and Anandkumar, A. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 6(5): 320–328, 2024.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Bonev, B., Kurth, T., Hundt, C., Pathak, J., Baust, M., Kashinath, K., and Anandkumar, A. Spherical Fourier neural operators: Learning stable dynamics on the sphere. In *International Conference on Machine Learning (ICML)*, 2023.

Bonnet, F., Mazari, J., Cinnella, P., and Gallinari, P. AirfRANS: High fidelity computational fluid dynamics dataset for approximating Reynolds-averaged Navier–Stokes solutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Cao, S. Choose a transformer: Fourier or Galerkin. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Deng, J., Li, X., Xiong, H., Hu, X., and Ma, J. Geometry-guided conditional adaptation for surrogate models of large-scale 3D PDEs on arbitrary geometries. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2024.

Gao, H. and Ji, S. Graph U-Nets. In *International Conference on Machine Learning (ICML)*, 2019.

Gupta, G., Xiao, X., and Bogdan, P. Multiwavelet-based operator learning for differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Hao, Z., Wang, Z., Su, H., Ying, C., Dong, Y., Liu, S., Cheng, Z., Song, J., and Zhu, J. GNOT: A general neural operator transformer for operator learning. In *International Conference on Machine Learning (ICML)*, 2023.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

Hu, P., Wang, Y., and Ma, Z.-M. Better neural pde solvers through data-free mesh movers. In *International Conference on Learning Representations (ICLR)*, 2024.

Hu, W., Liu, S., Qiao, P., Sun, Z., and Dou, Y. Transolver is a linear transformer: Revisiting physics-attention through the lens of linear attention. *arXiv preprint arXiv:2511.06294*, 2025.

Kissas, G., Seidman, J. H., Guilhoto, L. F., Preciado, V. M., Pappas, G. J., and Perdikaris, P. Learning operators with coupled attention. *Journal of Machine Learning Research*, 23(215):1–63, 2022.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Li, X., Li, Z., Kovachki, N., and Anandkumar, A. Geometric operator learning with optimal transport. *arXiv preprint arXiv:2507.20065*, 2025.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., Anandkumar, A., et al. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

Li, Z., Huang, D. Z., Liu, B., and Anandkumar, A. Fourier neural operator with learned deformations for PDEs on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023a.

Li, Z., Kovachki, N., Choy, C., Li, B., Kossaifi, J., Otta, S., Nabian, M. A., Stadler, M., Hundt, C., Azizzadenesheli, K., et al. Geometry-informed neural operator for large-scale 3D PDEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

Li, Z., Meidani, K., and Farimani, A. B. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023c.

Li, Z., Shu, D., and Barati Farimani, A. Scalable transformer for PDE surrogate modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023d.

Lingsch, L. E., Grund, D., Mishra, S., and Kissas, G. FUSE: Fast unified simulation and estimation for PDEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Lingsch, L. E., Michelis, M. Y., De Bezenac, E., Perera, S. M., Katzschmann, R. K., and Mishra, S. Beyond regular grids: Fourier-based neural operators on arbitrary domains. In *International Conference on Machine Learning (ICML)*, 2024b.

Liu, X., Xu, B., and Zhang, L. HT-Net: Hierarchical transformer based operator learning model for multiscale PDEs. *arXiv preprint arXiv:2210.10890*, 2022.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision (CVPR)*, 2021.

Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3): 218–229, 2021.

Luo, H., Wu, H., Zhou, H., Xing, L., Di, Y., Wang, J., and Long, M. Transolver++: An accurate neural solver for PDEs on million-scale geometries. In *International Conference on Machine Learning (ICML)*, 2025.

Nielsen, S. K., Teo, R. S. Y., Abdullaev, L. U., and Nguyen, T. M. Tight clusters make specialized experts. In *International Conference on Learning Representations (ICLR)*, 2025.

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., and Battaglia, P. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017.

Rahman, M. A., Ross, Z. E., and Azizzadenesheli, K. U-NO: U-shaped neural operators. *Transactions on Machine Learning Research*, 2022.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention (MICCAI)*, 2015.

Rowbottom, J., Maierhofer, G., Deveney, T., Müller, E. H., Paganini, A., Schratz, K., Lio, P., Schönlieb, C.-B., and Budd, C. G-adaptivity: Optimised graph-based mesh relocation for finite element methods. In *International Conference on Machine Learning (ICML)*, 2025.

Schlichting, H. and Gersten, K. *Boundary-layer theory*. Springer, 2016.

Seidman, J., Kissas, G., Perdikaris, P., and Pappas, G. J. NOMAD: Nonlinear manifold decoders for operator learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Seitzer, M., Tavakoli, A., Antic, D., and Martius, G. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Serrano, L., Le Boudec, L., Kassaï Koupaï, A., Wang, T. X., Yin, Y., Vittaut, J.-N., and Gallinari, P. Operator learning with neural fields: Tackling PDEs on general geometries. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Song, W., Zhang, M., Wallwork, J. G., Gao, J., Tian, Z., Sun, F., Piggott, M., Chen, J., Shi, Z., Chen, X., and Wang, J. M2n: Mesh movement networks for pde solvers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Tiwari, K., Dutta, N., Krishnan, N. A., and AP, P. Latent Mamba operator for partial differential equations. In *International Conference on Machine Learning (ICML)*, 2025.

Tran, A., Mathews, A., Xie, L., and Ong, C. S. Factorized Fourier neural operators. In *International Conference on Learning Representations (ICLR)*, 2023.

Wen, G., Li, Z., Azizzadenesheli, K., Anandkumar, A., and Benson, S. M. U-FNO—an enhanced Fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022.

Wen, S., Kumbhat, A., Lingsch, L., Mousavi, S., Zhao, Y., Chandrashekar, P., and Mishra, S. Geometry aware operator transformer as an efficient and accurate neural surrogate for PDEs on arbitrary domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Wu, H., Hu, T., Luo, H., Wang, J., and Long, M. Solving high-dimensional PDEs with latent spectral models. In *International Conference on Machine Learning (ICML)*, 2023.

Wu, H., Luo, H., Wang, H., Wang, J., and Long, M. Transolver: A fast transformer solver for PDEs on general geometries. In *International Conference on Machine Learning (ICML)*, 2024.

Xiao, Z., Hao, Z., Lin, B., Deng, Z., and Su, H. Improved operator learning by orthogonal attention. In *International Conference on Machine Learning (ICML)*, 2024.

Yang, J., Dzanic, T., Petersen, B., Kudo, J., Mittal, K., Tomov, V., Camier, J.-S., Zhao, T., Zha, H., Kolev, R., Anderson, R., and Faissol, D. Reinforcement learning for adaptive mesh refinement. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Yin, M., Charon, N., Brody, R., Lu, L., Trayanova, N., and Maggioni, M. DIMON: Learning solution operators of partial differential equations on a diffeomorphic family of domains. *arXiv preprint arXiv:2402.07250*, 2024.

Yin, Y., Kirchmeyer, M., Franceschi, J.-Y., Rakotomamonjy, A., and Gallinari, P. Continuous PDE dynamics forecasting with implicit neural representations. In *International Conference on Learning Representations (ICLR)*, 2023.

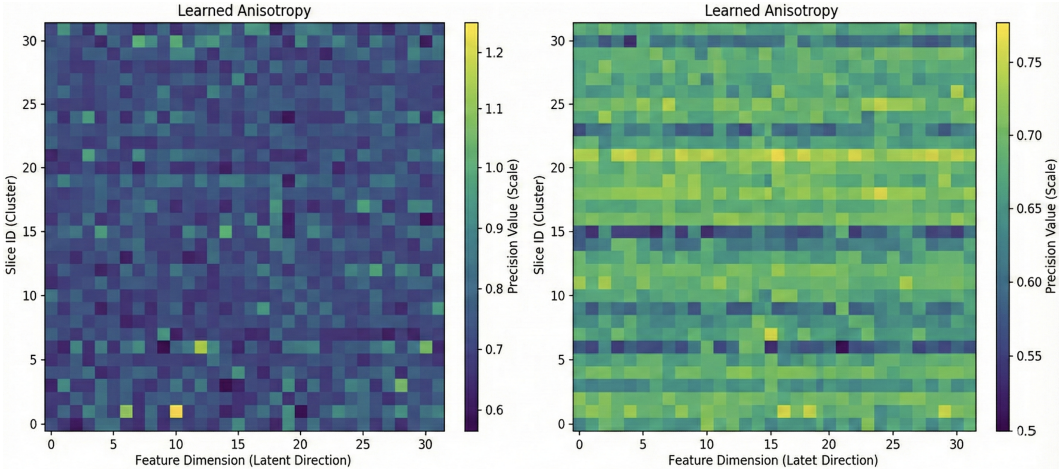


Figure 4: **Learned Anisotropy ($\mathbf{P}_s$).** (L) The Full M-Transolver captures complex directional variance. (R) Without the quadratic term, the model collapses into isotropic horizontal banding.

A Other Main Experiments

In this section, we provide extended quantitative results and detailed analyses that complement the main text. We first extend our evaluation to the domain of automotive aerodynamics using the ShapeNet-Car benchmark, validating the model's robustness on bluff-body flows. We then provide a comprehensive breakdown of the standard PDE benchmarks, offering a granular analysis of the performance trade-offs between architectural paradigms across different geometric topologies.

Table 4: Quantitative results on the **ShapeNet-Car** benchmark (3D automotive aerodynamics). We compare against geometric baselines and state-of-the-art neural operators. We report relative L_2 error (↓) for Volume and Surface fields, and the error/correlation for the Drag Coefficient (C_D, ρ_D). The best baseline results are *italicized*, and our results are **bolded**.

Model	ShapeNet-Car			
	Volume ↓	Surf ↓	C_D ↓	ρ_D ↑
GraphSAGE (Hamilton et al., 2017)	0.0461	0.1050	0.0270	0.9695
PointNet (Qi et al., 2017)	0.0494	0.1104	0.0298	0.9583
Graph U-Net (Gao & Ji, 2019)	0.0471	0.1102	0.0226	0.9725
MeshGraphNet (Pfaff et al., 2021)	0.0354	0.0781	0.0168	0.9840
GNO (Li et al., 2020)	0.0383	0.0815	0.0172	0.9834
Geo-FNO (Li et al., 2023a)	0.1670	0.2378	0.0664	0.8280
Galerkin (Cao, 2021)	0.0339	0.0878	0.0179	0.9764
GNOT (Hao et al., 2023)	0.0329	0.0798	0.0178	0.9833
GINO (Li et al., 2023b)	0.0386	0.0810	0.0184	0.9826
3D-GeoCA (Deng et al., 2024)	0.0319	0.0779	0.0159	0.9842
Transolver (Wu et al., 2024)	0.0225	0.0789	0.0123	*0.9915*
Transolver++ (Luo et al., 2025)	0.0226	0.0800	0.0132	0.9914
LaMO (Tiwari et al., 2025)	*0.0212*	*0.0763*	*0.0119*	0.9907
M-Transolver (Ours)	**0.0173**	**0.0741**	**0.0103**	**0.9938**
Relative Promotion	**+18.4%**	**+2.9%**	**+13.4%**	/

A.1 Generalization to Automotive Aerodynamics (ShapeNet-Car)

To validate the robustness of M-Transolver on bluff-body aerodynamics—characterized by massive flow separation and turbulent wakes—we evaluate performance on the *ShapeNet-Car* dataset. Unlike the streamlined flows of aviation airfoils, this task is dominated by drag forces rather than lift.

1. Accurate Drag Estimation. Estimating the Drag Coefficient (C_D) is the primary objective in automotive design optimization. As shown in Table 4, M-Transolver achieves a relative error of

0.0103, reducing the error of the best baseline (LaMO, 0.0119) by 13.4%. Furthermore, it attains a Spearman's rank correlation of 0.9938, surpassing Transolver 0.9915. This high ranking accuracy is critical for design optimization loops, ensuring the model can reliably distinguish between subtle geometric modifications that reduce drag.

2. Wake Resolution via Volume Error. We observe a substantial improvement in *Volume* error ($+18.4\%$ over LaMO). Automotive simulations involve a large computational domain where the wake region is spatially extensive and highly turbulent. The significant gain in volume accuracy suggests that the Mahalanobis metric effectively focuses attention on these complex wake structures, likely filtering out the uninformative free-stream regions that often dilute the performance of isotropic baselines.

A.2 Extended Analysis on Standard PDE Benchmarks

In this section, we present the comprehensive quantitative results and specific experimental configurations for all standard PDE benchmarks evaluated in this study.

As detailed in Table 5, our results demonstrate that the proposed anisotropic routing provides a consistent performance gains across the evaluated benchmark suite. On highly unstructured and irregular datasets, such as *Pipe* (which requires precise modeling of variable boundaries) and the *Elasticity* point cloud, M-Transolver achieves state-of-the-art errors of 0.0023 and 0.0044, respectively. This supports the view that global attention, when augmented with a local anisotropic metric, can capture complex pairwise interactions in disordered systems. Furthermore, the strong performance of M-LaMO on datasets like Plasticity and Airfoil indicates that our tokenization provides backbone-agnostic benefits.

Crucially, this representational advantage extends seamlessly to perfectly regular grids. To ensure a fair architectural evaluation of latent tokenization capacity, we evaluated the state-of-the-art SSM baseline LaMO using the identical isotropic "slice-and-process" embedding utilized by Transolver. Under this unified architectural formulation, replacing the isotropic clustering with our Mahalanobis metric allows M-Transolver to significantly outperform all baselines, setting a new state-of-the-art with a *Navier-Stokes* error of 0.0678 and a *Darcy* error of 0.0045. Ultimately, these results demonstrate that introducing a geometry-aware, anisotropic tokenization strategy allows models to efficiently compress both highly structured grids and heavily disordered point clouds into physically meaningful latent representations.

Table 5: Comparative performance (Relative L2 Error) on standard PDE benchmarks. Datasets are grouped by geometry type.

	Operator	Point Cloud	Regular Grid		Structured Mesh		
		Elasticity	Navier-Stokes	Darcy	Plasticity	Airfoil	Pipe
Classic	U-Net (Ronneberger et al., 2015)	0.0235	0.1982	0.0080	0.0051	0.0079	0.0065
	ResNet (He et al., 2016)	0.0262	0.2753	0.0587	0.0233	0.0391	0.0120
	Swin (Liu et al., 2021)	0.0283	0.2248	0.0397	0.0170	0.0270	0.0109
	DeepONet (Lu et al., 2021)	0.0965	0.2972	0.0588	0.0135	0.0385	0.0097
Frequency	WMT (Gupta et al., 2021)	0.0359	0.1541	0.0082	0.0076	0.0075	0.0077
	U-FNO (Wen et al., 2022)	0.0239	0.2231	0.0183	0.0039	0.0269	0.0056
	FNO (Li et al., 2021)	0.0229	0.1556	0.0108	0.0074	0.0138	0.0067
	U-NO (Rahman et al., 2022)	0.0258	0.1713	0.0113	0.0034	0.0078	0.0100
	F-FNO (Tran et al., 2023)	0.0263	0.2322	0.0077	0.0047	0.0078	0.0070
	LSM (Wu et al., 2023)	0.0218	0.1535	0.0065	0.0025	0.0059	0.0050
Transformer	Galerkin (Cao, 2021)	0.0240	0.1401	0.0084	0.0120	0.0118	0.0098
	HT-Net (Liu et al., 2022)	/	0.1847	0.0079	0.0333	0.0065	0.0059
	OFormer (Li et al., 2023c)	0.0183	0.1705	0.0124	0.0017	0.0183	0.0168
	GNOT (Hao et al., 2023)	0.0086	0.1380	0.0105	0.0336	0.0076	0.0047
	FactFormer (Li et al., 2023d)	/	0.1214	0.0109	0.0312	0.0071	0.0060
	ONO (Xiao et al., 2024)	0.0118	0.1195	0.0076	0.0013	0.0053	0.0046
	Transolver (Wu et al., 2024)	0.0064	0.0900	0.0057	0.0012	0.0053	0.0033
	Transolver++ (Luo et al., 2025)	0.0052	0.0719	0.0049	0.0011	0.0048	0.0027
	M-Transolver (ours)	**0.0044**	**0.0678**	**0.0045**	0.0007	0.0045	**0.0023**
SSM	LaMO (Tiwari et al., 2025)	0.0050	0.1010	0.0064	0.0007	0.0041	0.0038
	M-LaMO (ours)	0.0047	0.1027	0.0071	**0.0003**	**0.0037**	0.0035

B Continuous Interpretation of Metric Routing

We give a continuous interpretation of METRO's Mahalanobis metric routing. Rather than treating the full nonlinear architecture as a single linear integral operator, we separate the construction into four parts: (i) metric slice aggregation as a normalized weighted measurement, (ii) the finite-dimensional encoder–processor–decoder form of the full layer, (iii) the linear special case, where the construction reduces to a finite-rank integral operator, and (iv) the local pullback pseudometric induced by the latent Mahalanobis score.

Let $\Omega \subset \mathbb{R}^m$ be compact and let ν be the sampling measure on Ω. When ν has density ρ with respect to Lebesgue measure, we write $d\nu(\xi) = \rho(\xi)d\xi$. The distinction is important: without explicit quadrature weights, the continuum limit corresponds to the mesh sampling measure, not necessarily to physical volume measure.

The central object is the metric logit

$$\ell_g(\xi) = -\frac{1}{2}\|\phi(\xi) - \mu_g\|^2_{P_g} + \frac{1}{2}\log\det P_g,$$

where $\phi : \Omega \to \mathbb{R}^D$ is the learned geometry encoder, $\mu_g \in \mathbb{R}^D$, and $P_g \succ 0$. This logit defines a slice-specific anisotropic notion of proximity in the learned feature space. After converting logits into nonnegative assignment weights, each latent slice computes a normalized metric-weighted measurement of the input field. This corresponds to the column-normalized aggregation weights used for slicing in Algorithm 1.

For the default METRO routing rule,

$$w_g(\xi) = \frac{\exp(\ell_g(\xi))}{\sum_{r=1}^{G}\exp(\ell_r(\xi))},$$

the weights form a partition of unity across slices:

$$w_g(\xi) \geq 0, \qquad \sum_{g=1}^{G} w_g(\xi) = 1.$$

The aggregation result below does not depend on the softmax form specifically. It applies to any bounded, measurable, nonnegative metric assignment w_g with positive finite mass, including unnormalized assignments such as $\exp(\ell_g)$, $\mathrm{softplus}(\ell_g)$, or $\sigma(\ell_g)$.

All results in this section should be read conditionally on fixed learned parameters and fixed assignment functions. If the assignments depend on the input field, the resulting operator is generally nonlinear even before applying the latent processor.

Proposition B.1 (Continuous limit of metric slice aggregation)**.** *Let $\xi_1, \ldots, \xi_N$ be i.i.d. samples from ν. Let $f : \Omega \to \mathbb{R}^C$ satisfy $f \in L^1(\Omega, \nu; \mathbb{R}^C)$. Assume $w_g : \Omega \to \mathbb{R}_{\geq 0}$ is bounded, measurable, and has positive mass*

$$\alpha_g = \int_\Omega w_g(\xi)\, d\nu(\xi) > 0.$$

Define the empirical metric slice coefficient

$$z_{g,N}(f) = \frac{\sum_{i=1}^{N} w_g(\xi_i) f(\xi_i)}{\sum_{i=1}^{N} w_g(\xi_i)}.$$

Then

$$z_{g,N}(f) \to z_g(f) = \frac{\int_\Omega w_g(\xi) f(\xi)\, d\nu(\xi)}{\int_\Omega w_g(\xi)\, d\nu(\xi)}$$

almost surely as $N \to \infty$, componentwise in $\mathbb{R}^C$.

Proof. Since w_g is bounded and $f \in L^1(\Omega, \nu; \mathbb{R}^C)$, each component of $w_g f$ is integrable. By the strong law of large numbers, for each component,

$$\frac{1}{N}\sum_{i=1}^{N} w_g(\xi_i) f(\xi_i) \to \int_\Omega w_g(\xi) f(\xi)\, d\nu(\xi)$$

almost surely. Similarly,

$$\frac{1}{N}\sum_{i=1}^{N} w_g(\xi_i) \to \int_\Omega w_g(\xi)\, d\nu(\xi) = \alpha_g$$

almost surely. Since $\alpha_g > 0$, the ratio converges componentwise to the stated limit. □

Deterministic meshes. The i.i.d. assumption is used only to state a clean Monte Carlo limit. For deterministic meshes, the same interpretation applies whenever the empirical quadrature rule converges to integration against ν. With quadrature weights q_i, the corresponding discrete slice coefficient is

$$z^q_{g,N}(f) = \frac{\sum_{i=1}^N q_i w_g(\xi_i) f(\xi_i)}{\sum_{i=1}^N q_i w_g(\xi_i)}.$$

If no quadrature weights are used, the limiting measure is the empirical mesh-node distribution.

Proposition B.2 (Metric encoder–processor–decoder form)**.** *For fixed metric assignment functions w_g and reconstruction weights $\widetilde{w}_g$, the continuous METRO layer can be written as*

$$\widehat{f} = D_{\widetilde{w}} \circ F_\theta \circ E_w(f),$$

where the metric encoder is

$$(E_w f)_g = \frac{\int_\Omega w_g(\xi) f(\xi)\, d\nu(\xi)}{\int_\Omega w_g(\xi)\, d\nu(\xi)},$$

the latent processor is $F_\theta : \mathbb{R}^{G\times C} \to \mathbb{R}^{G\times C'}$, and the decoder is

$$(D_{\widetilde{w}} a)(y) = \sum_{g=1}^{G} \widetilde{w}_g(y) a_g.$$

Thus, METRO acts through a finite-dimensional set of metric-weighted latent measurements before reconstructing in the output domain.

Proof. By Proposition B.1, the empirical slice aggregation converges componentwise to the continuous metric-weighted coefficient $(E_w f)_g$. The latent backbone maps the resulting coefficients through F_θ, and reconstruction applies the weighted decoder $D_{\widetilde{w}}$. Composing these three maps gives the stated form. □

If F_θ is nonlinear, this is generally not a linear integral operator with an input-independent kernel. Moreover, if the assignment functions themselves depend on the input field, the resulting operator is nonlinear even when F_θ is linear.

Proposition B.3 (Linear special case as a finite-rank integral operator)**.** *Assume the metric assignments are fixed and the latent processor is linear. In the scalar-channel case, let*

$$[F_\theta(z)]_g = \sum_{h=1}^{G} B_{gh} z_h.$$

Then METRO reduces to the finite-rank integral operator

$$\widehat{f}(y) = \int_\Omega K_B(y,\xi) f(\xi)\, d\nu(\xi),$$

with kernel

$$K_B(y,\xi) = \sum_{g=1}^{G}\sum_{h=1}^{G} \widetilde{w}_g(y) B_{gh} \frac{w_h(\xi)}{\int_\Omega w_h(s)\, d\nu(s)}.$$

For vector-valued channels, the same formula holds with B_{gh} replaced by the corresponding channel-mixing matrix.

Proof. Substituting the linear form of F_θ into Proposition B.2 gives

$$\widehat{f}(y) = \sum_{g=1}^{G} \widetilde{w}_g(y) \sum_{h=1}^{G} B_{gh} \frac{\int_\Omega w_h(\xi) f(\xi)\, d\nu(\xi)}{\int_\Omega w_h(s)\, d\nu(s)}.$$

Since the sums are finite, we exchange summation and integration to obtain the stated kernel. The vector-valued case is identical with matrix-valued channel maps. □

This finite-rank kernel is a linear special case of the metric-routing formulation, not an assumption on the full nonlinear METRO model.

Proposition B.4 (Role of the log-determinant term)**.** *For the metric logit*

$$\ell_g(\xi) = -\frac{1}{2}\|\phi(\xi) - \mu_g\|_{P_g}^2 + \frac{1}{2}\log\det P_g,$$

we have

$$\exp(\ell_g(\xi)) = \det(P_g)^{1/2} \exp\left(-\frac{1}{2}\|\phi(\xi) - \mu_g\|_{P_g}^2\right).$$

Thus, before normalization, the log-determinant term contributes the standard latent Gaussian volume factor associated with the precision P_g.

Proof. Exponentiating the definition of ℓ_g gives

$$\exp(\ell_g(\xi)) = \exp\left(-\frac{1}{2}\|\phi(\xi) - \mu_g\|_{P_g}^2\right) \exp\left(\frac{1}{2}\log\det P_g\right).$$

Since $\exp(\frac{1}{2}\log\det P_g) = \det(P_g)^{1/2}$, the identity follows. □

Under softmax routing across slices, this term changes the relative competition among slices and therefore affects the row-wise assignments $w_g(\xi)$. For independently normalized non-softmax slice aggregation, however, multiplying a single slice weight w_g by a positive constant does not change the normalized slice coefficient in Proposition B.1, because the same constant appears in both numerator and denominator. Therefore, the log-determinant term should be understood primarily as a cross-slice competition and latent-volume factor in the default METRO softmax routing. It should not be interpreted as a physical-domain Jacobian correction unless the encoder has additional structure, such as $D = m$ and a local diffeomorphism.

Proposition B.5 (Local pullback geometry of metric routing)**.** *Let $\phi : \mathbb{R}^m \to \mathbb{R}^D$ be twice continuously differentiable and let $P \succ 0$. Define*

$$q(x) = \|\phi(x) - \mu\|_P^2.$$

Then

$$\nabla^2 q(x_0) = 2J_\phi(x_0)^\top P J_\phi(x_0) + 2\sum_{k=1}^{D} [P(\phi(x_0) - \mu)]_k \nabla^2 \phi_k(x_0).$$

In particular, if $\phi(x_0) = \mu$, then

$$\nabla^2 q(x_0) = 2J_\phi(x_0)^\top P J_\phi(x_0).$$

Thus, near a slice center and under local linearization of the encoder, the physical shape of a latent Mahalanobis slice is governed by the pullback quadratic form $J_\phi^\top P J_\phi$.

Proof. Let $r(x) = \phi(x) - \mu$, so $q(x) = r(x)^\top P r(x)$. Differentiating once gives

$$\nabla q(x) = 2J_\phi(x)^\top P r(x).$$

Differentiating again gives

$$\nabla^2 q(x_0) = 2J_\phi(x_0)^\top P J_\phi(x_0) + 2\sum_{k=1}^{D} [Pr(x_0)]_k \nabla^2 \phi_k(x_0).$$

Substituting $r(x_0) = 0$ gives the centered expression. □

Since $P \succ 0$, the pullback form

$$Q_g(x) = J_\phi(x)^\top P_g J_\phi(x)$$

is positive semidefinite. If $J_\phi(x)$ has full column rank, then $Q_g(x)$ is positive definite and defines a local Riemannian metric. If $J_\phi(x)$ is rank deficient, $Q_g(x)$ is a positive semidefinite pullback quadratic form, i.e. a local pseudometric. This explains how diagonal latent precision matrices can induce locally oriented anisotropic receptive fields in physical coordinates, while making clear that the statement is local and conditional on the encoder geometry.

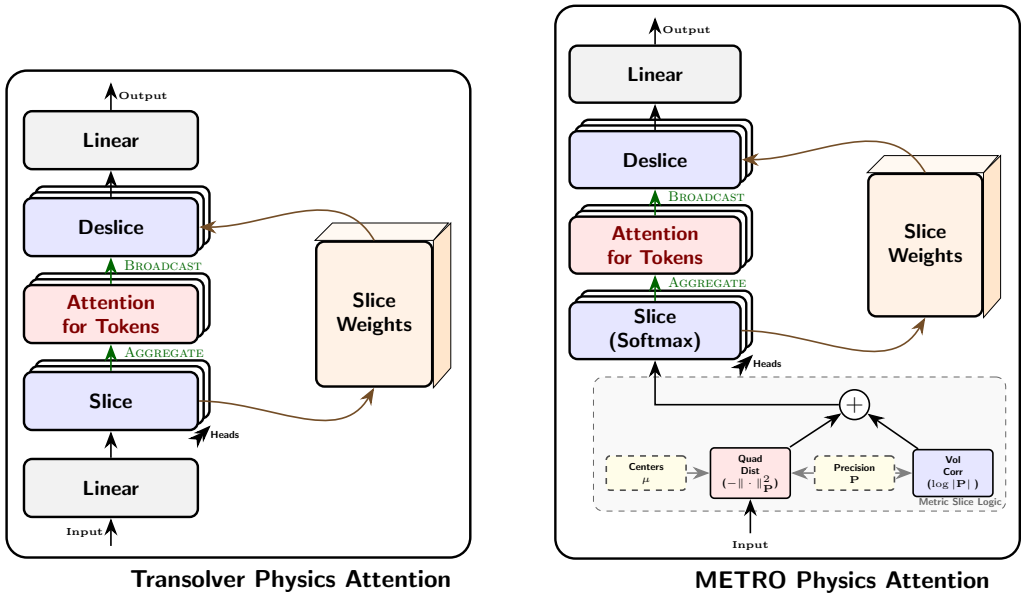
C Algorithmic Formulation and Visual Analysis

In this section, we provide the formal procedural details of the M-Transolver layer and visually compare its attention mechanism against the baseline.

Algorithm 1 M-Transolver: Metric-Aware Slicing Mechanism

Require: Input features $\mathbf{X} \in \mathbb{R}^{N \times D_{\text{in}}}$, number of slices G.
Ensure: Output features $\mathbf{Y}_{\text{out}} \in \mathbb{R}^{N \times D_{\text{out}}}$.

1. Projection
$\mathbf{X}_{\text{geo}} \leftarrow \text{Linear}_{\text{geo}}(\mathbf{X})$ ▷ $\mathbf{X}_{\text{geo}} \in \mathbb{R}^{N \times D}$
$\mathbf{V} \leftarrow \text{Linear}_{\text{val}}(\mathbf{X})$ ▷ $\mathbf{V} \in \mathbb{R}^{N \times C}$
2. Metric Parameters
Retrieve learnable centers $\boldsymbol{\mu} \in \mathbb{R}^{G \times D}$ and raw diagonal precisions $\mathbf{P}_{\text{raw}} \in \mathbb{R}^{G \times D}$.
$\mathbf{P} \leftarrow \text{softplus}(\mathbf{P}_{\text{raw}}) + \epsilon_{\text{pd}}$ ▷ Positive diagonal precisions
3. Metric Slice Assignment
for $g = 1, \ldots, G$ **do**
  for $n = 1, \ldots, N$ **do**
    $d^2_{n,g} \leftarrow \sum_{k=1}^{D} P_{g,k} \left(X_{\text{geo},n,k} - \mu_{g,k}\right)^2$ ▷ Diagonal Mahalanobis distance
    $L_{n,g} \leftarrow -\frac{1}{2} d^2_{n,g} + \frac{1}{2} \sum_{k=1}^{D} \log P_{g,k}$ ▷ Metric logit
  end for
end for
$\mathbf{A}^{\text{row}} \leftarrow \text{softmax}_g(\mathbf{L})$ ▷ Row-wise slice assignments, $\sum_g A^{\text{row}}_{n,g} = 1$
4. Aggregate into Slice Tokens
$m_g \leftarrow \sum_{n=1}^{N} A^{\text{row}}_{n,g}$ for each g ▷ Slice masses
$A^{\text{slice}}_{n,g} \leftarrow A^{\text{row}}_{n,g} / m_g$ ▷ Column-normalized aggregation weights
$\mathbf{T}_{\text{slice}} \leftarrow (\mathbf{A}^{\text{slice}})^\top \mathbf{V}$ ▷ $\mathbf{T}_{\text{slice}} \in \mathbb{R}^{G \times C}$
5. Process Slice Tokens
$\mathbf{T}_{\text{out}} \leftarrow \text{TransformerBlock}(\mathbf{T}_{\text{slice}})$ ▷ Inter-slice processing
6. Broadcast Back to Points
$\mathbf{Y} \leftarrow \mathbf{A}^{\text{row}} \mathbf{T}_{\text{out}}$ ▷ Point-wise reconstruction
$\mathbf{Y}_{\text{out}} \leftarrow \text{Linear}_{\text{out}}(\mathbf{Y})$
return $\mathbf{Y}_{\text{out}}$



(a) Baseline Transolver (Isotropic)

(b) M-Transolver (Anisotropic)

Figure 5: **Impact of Metric Learning on Attention.** (a) The standard Transolver uses isotropic Euclidean distances, treating all directions equally. This often results in attention leaking into irrelevant regions (e.g., across boundary layers). (b) **M-Transolver** adapts the metric tensor $\mathbf{P}$ to the local physics. The attention mechanism naturally elongates along flow streamlines and compresses across normal gradients, resulting in cleaner, physics-aligned feature aggregation as described in Algorithm 1.

D Computational Complexity Analysis

A critical requirement for neural PDE solvers is scalability. Here, we formally prove that the Anisotropic Physics-Attention mechanism maintains linear complexity with respect to the input resolution N, regardless of whether a full or diagonal precision matrix is employed.

Proposition D.1 (Asymptotic Complexity)**.** *Let N be the number of mesh points, G be the number of latent slices, and D be the dimension of the feature space. The computational complexity of the Anisotropic Physics-Attention is $\mathcal{O}(N \cdot G \cdot D^k)$, where $k = 1$ for the diagonal implementation and $k = 2$ for the full matrix formulation. In both cases, the complexity remains linear with respect to the resolution N.*

Proof. The computational cost is dominated by the scoring function between latent features $\mathbf{z}_j \in \mathbb{R}^D$ and slice centers $\boldsymbol{\mu}_g \in \mathbb{R}^D$. We analyze the complexity for the general and optimized cases.

1. Baseline: Standard Linear Attention. The standard Transolver employs a dot-product attention mechanism ($\mathbf{q}^\top \mathbf{k}$). For feature vectors of dimension D, this requires D multiplications and summations per pair.

$$\mathcal{C}_{\text{linear}} = \mathcal{O}(N \cdot G \cdot D). \tag{9}$$

2. Theoretical Generalization: Full Anisotropy. In the general case, the Mahalanobis distance is computed using a dense precision matrix $P_g \in \mathbb{R}^{D \times D}$, capturing correlations between all feature dimensions:

$$\text{Score}_{\text{full}} = -\frac{1}{2}(z_j - \mu_g)^\top P_g (z_j - \mu_g) + \frac{1}{2}\log\det(P_g).$$

The matrix-vector multiplication $P_g(z_j - \mu_g)$ requires $O(D^2)$ operations per point-slice pair, giving $O(NGD^2)$ pairwise scoring cost. Computing or parameterizing $\log\det(P_g)$ may add an additional

Table 6: Summary of experiment benchmarks. The first six datasets are from FNO (Li et al., 2021) and geo-FNO (Li et al., 2023a), AirfRANS is from (Bonnet et al., 2022), and AirCraft is a high-fidelity industrial dataset. #Mesh records the size of discretized meshes. #Dataset is organized as the number of samples in training and test sets.

GEOMETRY	BENCHMARKS	#DIM	#MESH	#INPUT	#OUTPUT	#DATASET
POINT CLOUD	ELASTICITY	2D	972	STRUCTURE	INNER STRESS	(1000, 200)
STRUCTURED MESH	PLASTICITY	2D+TIME	3,131	EXTERNAL FORCE	MESH DISPLACEMENT	(900, 80)
	AIRFOIL	2D	11,271	STRUCTURE	MACH NUMBER	(1000, 200)
	PIPE	2D	16,641	STRUCTURE	FLUID VELOCITY	(1000, 200)
REGULAR GRID	NAVIER-STOKES	2D+TIME	4,096	PAST VELOCITY	FUTURE VELOCITY	(1000, 200)
	DARCY	2D	7,225	POROUS MEDIUM	FLUID PRESSURE	(1000, 200)
UNSTRUCTURED MESH	AIRFRANS	2D	32,000	STRUCTURE	VELOCITY & PRESSURE	(800, 200)
	SHAPENET-CAR	3D	32,186	STRUCTURE	VELOCITY & PRESSURE	(789, 100)
	AIRCRAFT	3D	∼300,000	STRUCTURE	6 QUANTITIES	(110, 10)

per-slice cost, e.g. $O(GD^3)$ for a naive dense determinant, but this term is independent of the mesh resolution N. Therefore, the full formulation remains linear with respect to N, with dominant pairwise scoring cost $O(NGD^2)$ in the large-mesh regime.

Remark: For tasks operating directly on low-dimensional physical coordinates ($D \in \{2, 3\}$), the D^2 pairwise cost and dense log-determinant cost are small constants. In this setting, the full formulation is practically close to the linear baseline, although our high-dimensional latent implementation uses the diagonal formulation for efficiency and stability.

3. Efficient Implementation: Diagonal Anisotropy. To scale our method to high-dimensional latent spaces (e.g., $D = 64$ or 128), we constrain $\mathbf{P}_g$ to be diagonal, $\mathbf{P}_g = \text{diag}(p_{g,1}, \ldots, p_{g,D})$. This models axis-aligned scaling without cross-correlations.

$$\text{Score}_{\text{diag}} = \sum_{k=1}^{D} -\frac{1}{2} p_{g,k}(z_{j,k} - \mu_{g,k})^2 + \frac{1}{2}\log(p_{g,k}). \tag{10}$$

- The term $p_{g,k}(z_{j,k} - \mu_{g,k})^2$ involves element-wise operations only.
- This reduces the per-pair cost from $\mathcal{O}(D^2)$ to $\mathcal{O}(D)$.
- Total Complexity: $\mathcal{O}(N \cdot G \cdot D)$.

Conclusion. Our implementation (Case 3) achieves strict asymptotic equivalence to the standard linear attention (Case 1). Even in the theoretically richer Case 2, the complexity remains linear with respect to the mesh size N, ensuring the model scales efficiently to massive physical simulations. □

D.1 Parameter-Matched Efficiency.

To verify that METRO's improvements stem from superior routing rather than added capacity, we evaluated a strict parameter-matched setting. The standard Transolver ($G = 64$ slices) contains $\approx$ 2.82M parameters and achieves an Elasticity error of 0.0064 at $\sim$ 10s/epoch. By increasing Transolver to $G = 96$ slices and reducing M-Transolver to $G = 16$ slices, both models are matched at $\approx$ 2.83M parameters. Under these constraints, M-Transolver (0.0048 error, $\sim$ 8s/epoch) simultaneously outperforms the baseline (0.0061 error, $\sim$ 13s/epoch) in both accuracy and computational speed. The anisotropic routing efficiently covers the physical domain using a fraction of the latent tokens required by isotropic routing.

E Latent versus Physical Partition of Unity

Our proposed METRO framework fundamentally operates by constructing a partition of unity over the *latent* feature space $\mathcal{Z}$. This algebraic domain of operation yields a distinct representational advantage over methods that construct localized weighting functions directly in the physical domain Ω.

The Constraint of Physical Contiguity. Standard physical partition of unity methods define weighting functions $\{\psi_k(\xi)\}_{k=1}^K$ over spatial coordinates $\xi \in \Omega$. To maintain operator locality, the support of each function, $\text{supp}(\psi_k)$, is bounded to a contiguous spatial neighborhood defined by a bandwidth ϵ. Algebraically, if two points ξ_1 and ξ_2 are separated by a Euclidean distance $\|\xi_1 - \xi_2\| > \epsilon$, the condition $\psi_k(\xi_1) > 0 \implies \psi_k(\xi_2) = 0$ is rigidly enforced. Consequently, a physical partition of unity cannot aggregate spatially separated regions exhibiting analogous physical phenomena without expanding its bandwidth to encompass the intervening, potentially dissimilar domain. This spatial constraint forces the network to allocate separate, redundant clusters for repeating physical states across the mesh.

Latent Partitioning and Semantic Grouping. METRO bypasses this limitation by computing the assignment score via the Mahalanobis metric over the learned embeddings $x_n = \phi(\xi_n)$ (Eq. 2).

To illustrate this, consider the aerodynamic simulation of a vehicle. A physical partition of unity method tasked with resolving localized turbulence, such as the flow separating around the side mirrors, typically requires independent, redundant clusters for the left and right mirrors. Because the physical distance between the mirrors is much larger than the local flow features, a contiguous spatial kernel cannot group them together without also absorbing the unrelated flow fields in between, such as the laminar flow over the windshield.

In contrast, METRO's non-linear encoder $\phi : \Omega \to \mathcal{Z}$ projects regions governed by analogous local dynamics to proximal coordinates in the latent space. Because the flow behavior around both mirrors shares identical semantic features, they are embedded into the same neighborhood in $\mathcal{Z}$. Consequently, a single latent ellipsoid parameterized by P_g encapsulates the features of both structures simultaneously. When projected back to the physical domain via the preimage ϕ^{-1}, this single latent slice naturally manifests as a union of spatially disjoint subsets, accurately resolving both regions.

This property ensures that a single task-specialized metric P_g can process spatially separated but physically identical structures simultaneously. As illustrated in Figure 6, METRO efficiently aggregates repeating semantic features across the domain, achieving global feature localization while eliminating the parameter redundancy inherent to physically contiguous formulations.

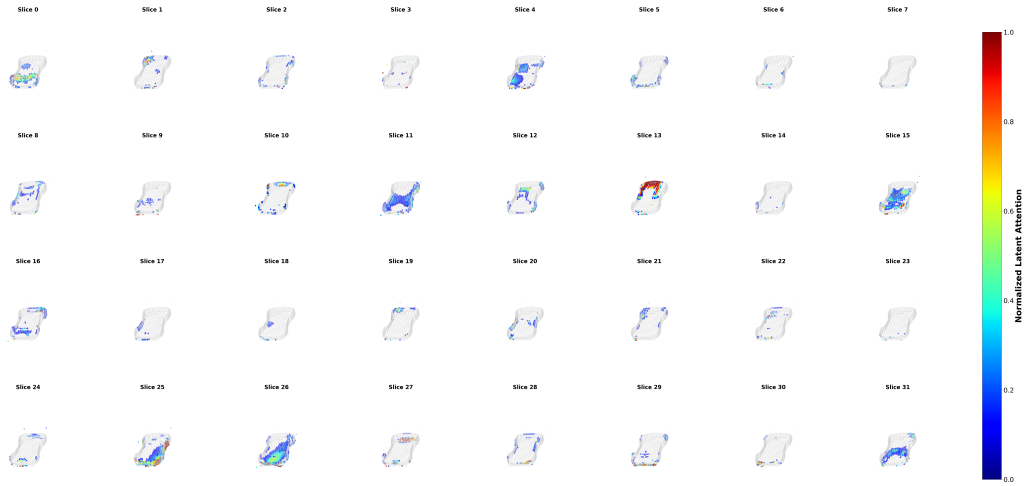


Figure 6: Latent versus physical domain partitioning in vehicle aerodynamics. While standard physical partition of unity methods must allocate independent, redundant spatial clusters to capture spatially disconnected structures—such as the separated flow around bilateral side mirrors—METRO's latent Mahalanobis routing embeds these semantically identical regions into the same latent neighborhood. When projected back to the physical domain via the preimage ϕ^{-1}, a single latent slice naturally encompasses both mirrors. This demonstrates METRO's ability to achieve efficient global feature localization without being constrained by physical contiguity.

F Benchmarks

To evaluate the effectiveness and generalization capabilities of our model, we conduct experiments across nine distinct benchmarks. These datasets encompass a wide spectrum of physical phenomena, ranging from solid mechanics (Elasticity, Plasticity) to fluid dynamics (Airfoil, Pipe, Navier-Stokes, Darcy, AirfRANS, AirCraft, and ShapeNet-Car). They vary significantly in geometry, dimensionality, and resolution, spanning both standard academic baselines and complex industrial-grade simulations.

F.1 Solid Mechanics

Elasticity. This benchmark focuses on estimating the internal stress distribution within hyper-elastic materials governed by the Rivlin-Saunders material model. Each case simulates a unit cell containing a central void, clamped at the bottom and subjected to tension at the top. The geometry is discretized into 972 points. The input is a tensor of shape 972×2 containing the 2D spatial coordinates, and the target output is a 972×1 tensor representing the von Mises stress at each point. The dataset comprises 1,000 samples for training and 200 samples for testing.

Plasticity. This task involves predicting the deformation dynamics of a plastic material subjected to impact. The simulation models a scenario where a rigid, frictionless die of arbitrary shape strikes a block at a constant velocity, described by an elastoplastic constitutive model. The input is the geometry of the die, discretized into a 101×31 mesh. The objective is to predict the material deformation over 20 future time steps, resulting in an output tensor of shape $20 \times 101 \times 31 \times 4$ (representing deformation in four directions). The dataset includes 900 samples for training and 80 for testing.

F.2 Fluid Dynamics

Airfoil. Designed to test the modeling of transonic flow, this benchmark utilizes the Euler equations to simulate flow over an airfoil. The domain is discretized into a structured mesh of 221×51, with shapes derived from deformations of the baseline NACA-0012 airfoil. The model takes the mesh node positions as input and predicts the Mach number field for each mesh point. The dataset consists of 1,000 training samples and 200 testing samples.

Pipe. This benchmark simulates incompressible fluid flow through a pipe channel, governed by the Navier-Stokes equations with a fixed viscosity of $\nu = 0.005$. The geometry involves a pipe of length 10 and width 1, with varying centerline shapes controlled by cubic polynomials. The fluid field is represented by a 129×129 structured mesh. The input is a tensor of shape $129 \times 129 \times 2$ (spatial coordinates), and the goal is to predict the horizontal velocity component ($129 \times 129 \times 1$). The split is 1,000 samples for training and 200 for testing.

Navier-Stokes. This classic benchmark models viscous, incompressible flow on a 2D unit torus $(0, 1)^2$ using the vorticity formulation. With a fixed viscosity of 10^{-5}, the data is discretized on a 64×64 regular grid. The task is a temporal forecasting problem: given the fluid velocity field for the past 10 time steps, the model must predict the velocity for the subsequent 10 steps. The dataset provides 1,000 simulations with varied initial conditions for training and 200 for testing.

Darcy Flow. This dataset simulates steady-state fluid flow through a porous medium on a unit square, governed by Darcy's law (a second-order elliptic equation). The input is the permeability field representing the porous structure, and the output is the resulting fluid pressure field. While originally generated on a 421×421 grid, it is downsampled to 85×85 for experimentation. The dataset contains 1,000 training samples and 200 testing samples.

F.3 Industrial Applications

AirfRANS. Providing higher fidelity than the standard Airfoil benchmark, AirfRANS utilizes Reynolds-Averaged Navier–Stokes (RANS) simulations. It includes airfoils from the NACA 4-digit and 5-digit series, discretized into finer meshes of approximately 32,000 points. The dataset captures variations in airfoil shape, angle of attack, and Reynolds number. The primary evaluation targets are the pressure quantities on the volume and surface, as well as the lift coefficients. The dataset includes 800 training samples and 200 testing samples.

AirCraft. This benchmark represents a significant increase in complexity, involving 3D high-fidelity CFD simulations of over 30 distinct aircraft designs. Simulations cover diverse flow conditions, including variations in Mach number and angle of attack. Unlike 2D benchmarks, each aircraft is discretized into a massive unstructured mesh of approximately 300,000 points, where each point is characterized by spatial coordinates (x, y, z) and surface normals. The model predicts six key physical quantities: pressure coefficient C_p, fluid density ρ, velocity components (U, V, W), and pressure p. Due to dataset availability, we utilize 110 samples for training and 10 samples for testing.

ShapeNet-Car. Focused on automotive design, this benchmark estimates the drag coefficient for vehicles moving at 72 km/h. The dataset contains 889 samples derived from the "car" category of ShapeNet, with 789 used for training and 100 for testing. Each sample is discretized into an unstructured mesh of 32,186 points, where the input features combine mesh position, signed distance function (SDF), and surface normal vectors. The model is tasked with predicting the velocity and pressure fields at every point, which are subsequently used to calculate the vehicle's aerodynamic drag coefficient.

F.4 Evaluation Metrics

To comprehensively assess the performance of our model, we employ three distinct metrics. For all benchmarks, we utilize the standard Relative L2 Error to evaluate field reconstruction accuracy. Additionally, for the AirfRANS and ShapeNet-Car benchmarks, we introduce design-oriented metrics—specifically the error in Aerodynamic Coefficients (Lift and Drag) and Spearman's Rank Correlation—to evaluate the model's utility in aerodynamic optimization tasks.

Relative L2 Error (Field). For all datasets (Elasticity, Plasticity, Airfoil, Pipe, Navier-Stokes, Darcy, AirfRANS, AirCraft, and ShapeNet-Car), we measure the accuracy of the predicted physical fields using the relative root mean square error. Given the ground truth field u and the model prediction $\hat{u}$, the Relative L2 error (L2RE) is defined as:

$$\text{L2RE} = \frac{\|\hat{u} - u\|_2}{\|u\|_2}, \tag{11}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. This metric provides a direct, normalized quantification of the deviation between the predicted physics states and the high-fidelity simulations.

Relative L2 Error (Aerodynamic Coefficients). For industrial benchmarks, the utility of a solver is often determined by its ability to predict scalar quantities derived from the fields. We specifically focus on the Lift Coefficient (C_L) for AirfRANS and the Drag Coefficient (C_D) for ShapeNet-Car. These coefficients are computed by integrating the pressure p and wall shear stress τ over the object surface $\partial\Omega$:

$$C_{(\cdot)} = \frac{2}{\rho v^2 A} \int_{\partial\Omega} (p(\xi)\mathbf{n}(\xi) \cdot \mathbf{d} + \tau(\xi) \cdot \mathbf{d}) d\xi, \tag{12}$$

where ρ is the fluid density, v is the inlet velocity, A is the reference area, $\mathbf{n}$ is the surface normal, and $\mathbf{d}$ is the projection direction. For Lift (C_L), $\mathbf{d}$ aligns with the vertical axis (e.g., $(0, 0, 1)$); for Drag (C_D), $\mathbf{d}$ aligns with the flow direction. We calculate the relative error between the predicted coefficient $\hat{C}$ and the ground truth C:

$$\text{L2RE}_{coef} = \frac{\|\hat{C} - C\|_2}{\|C\|_2}. \tag{13}$$

Spearman's Rank Correlation. In industrial design optimization, the relative ranking of different geometric designs is often more critical than absolute numerical precision. To evaluate how well our model preserves the ordering of designs based on their aerodynamic performance, we utilize Spearman's rank correlation coefficient (ρ) on the AirfRANS (Lift) and ShapeNet-Car (Drag) test sets. Given K test samples, we compare the rankings of the ground truth coefficients $R(C)$ against the rankings of the predicted coefficients $R(\hat{C})$:

$$\rho = \frac{\text{cov}(R(C), R(\hat{C}))}{\sigma_{R(C)}\sigma_{R(\hat{C})}}, \tag{14}$$

where $\text{cov}(\cdot)$ denotes covariance and σ represents standard deviation. A higher correlation $\rho \in [-1, 1]$ indicates that the model correctly identifies better-performing designs (e.g., higher lift or lower drag), making it a reliable surrogate for design optimization loops.

F.5 Implementations

As shown in Table 7, all baselines and our proposed models are trained and tested using consistent strategies to ensure fair comparison.

METRO Models Configuration. For our proposed METRO models (M-Transolver and M-LaMO), we adopt the same architectural hyperparameters as the original Transolver (Wu et al., 2024) to ensure that performance gains are attributable to the learned Mahalanobis physics attention rather than increased model capacity. Specifically, the number of channels D is set to 256 for high-dimensional or complex inputs (Navier-Stokes, AirfRANS, ShapeNet-Car, and AirCraft) and 128 for the remaining benchmarks. To balance computational efficiency with expressivity, we set the number of slices G to 32 when $D = 256$ and 64 when $D = 128$.

Baselines and Reproduction Protocol. For the top three leading baselines (Transolver, LaMO, and Transolver++), we executed a comprehensive reproduction study following the strict grid search strategy and hyperparameter settings detailed in the Transolver paper. In all cases, we were able to reproduce the results within the reported standard deviation; consequently, we report the numbers from the original authors to maintain consistency with the literature. For other established baselines, we reuse the results directly from the Transolver, Transolver++, LNO, and LaMO papers.

AirfRANS and ShapeNet-Car Settings. For the AirfRANS and ShapeNet-Car datasets, we note that the Transolver++ and LaMO papers did not originally report results on these benchmarks. To ensure a fair comparison, we trained these models (along with our METRO models) using the identical settings adopted by Transolver.

AirCraft Settings and Baseline Adaptations. For the AirCraft dataset, we conducted a grid search for all models, sweeping the number of layers in $\{2, 4, 6, 8\}$, number of channels in $\{64, 128, 256, 512\}$, and number of heads in $\{1, 2, 4, 8\}$. We selected the best-performing configuration provided that the total memory consumption remained under a strict budget of 48GB.

- *Baselines Implementation:* We implemented GraphSAGE, PointNet, and Graph U-Net following the codebase of Transolver. For GNO, 3D-GeoCA, and GINO, we adopted the official codebases. All models were trained to jointly predict the six available physical fields: pressure coefficient (C_p), density (ρ), velocity components (U, V, W), and static pressure (P).

- *Efficiency:* The Transolver family and our METRO variants proved highly efficient, utilizing less than 12GB of memory under optimal settings. Conversely, graph-based baselines such as GNO and MeshGraphNet exceeded the 48GB budget even with a reduced hidden size of 64; thus, for these specific baselines, we allowed the memory usage to exceed the 48GB limit to enable comparison.

The training configuration for AirCraft (loss functions, epochs, and optimizer) is identical to that of AirfRANS.

Table 7: Training and model configurations for M-Transolver and M-LaMO. Our settings strictly follow the baselines (Wu et al., 2024) to ensure fair comparison. L_{rL2} denotes the relative L2 loss, L_v and L_s represent volume and surface field losses, and L_g is the spatial gradient regularization term. Note that AirCraft adopts the same training configuration as AirfRANS.

	Training Configuration						Model Configuration			
Benchmarks	Loss	Epochs	Init LR	Optimizer	Batch	Scheduler	Layers L	Heads	Channels D	Slices G
Elasticity	Relative L2	500	10^{-3}	AdamW	1	CosineAnnealing	8	8	128	64
Plasticity	Relative L2	500	10^{-3}	AdamW	4	OneCycle	8	8	128	64
Airfoil	Relative L2	500	10^{-3}	AdamW	4	OneCycle	8	8	128	64
Pipe	Relative L2	500	10^{-3}	AdamW	4	OneCycle	8	8	128	64
Navier-Stokes	Relative L2	500	10^{-3}	AdamW	2	OneCycle	8	8	256	32
Darcy	$L_{rL2} + 0.1L_g$	500	10^{-3}	AdamW	8	OneCycle	8	8	128	64
AirfRANS	$L_v + L_s$	400	10^{-3}	Adam	1	OneCycle	8	8	256	32
AirCraft	$L_v + L_s$	400	10^{-3}	Adam	1	OneCycle	8	8	256	32
ShapeNet-Car	$L_v + 0.5L_s$	200	10^{-3}	Adam	1	OneCycle	8	8	256	32

F.6 Experimental Settings for Out-of-Distribution Generalization

To evaluate the generalizability of our model, we conducted experiments on unseen physical parameters using the AirfRANS dataset. We established two distinct Out-of-Distribution (OOD) settings: one for **Reynolds numbers** and one for **Angles of Attack**. In both cases, the training and test sets are constructed with non-overlapping parameter ranges to ensure the model cannot rely on interpolation. Table 8 details the specific ranges and sample sizes used for these experiments.

Table 8: Settings for OOD generalization experiments on AirfRANS. The training and test sets contain disjoint ranges for Reynolds numbers and angles of attack.

	OOD Reynolds		OOD Angles	
Dataset	Range	Samples	Range	Samples
Training Set	$[3 \times 10^6, 5 \times 10^6]$	500	$[-2.5°, 12.5°]$	800
Test Set	$[2 \times 10^6, 3 \times 10^6] \cup [5 \times 10^6, 6 \times 10^6]$	500	$[-5°, -2.5°] \cup [12.5°, 15°]$	200

G Additional Experiments

G.1 Ablation on Precision Matrix Structure

Finally, we investigate the structural parameterization of the precision matrix $\mathbf{P}_g$. While our standard model employs a diagonal precision matrix, theoretical expressivity can be increased by adding low-rank updates (e.g., $\mathbf{P}_g = \mathbf{D}_g + \mathbf{u}\mathbf{u}^\top$) to explicitly model off-diagonal correlations in the latent space. Table 9 summarizes the results on the Airfoil and Darcy datasets.

Trade-off: Expressivity vs. Optimization. Contrary to the expectation that higher theoretical expressivity yields better performance, we observe a degradation when introducing off-diagonal terms: the standard Diagonal model (0.0045) outperforms both Rank-1 (0.0047) and Rank-2 (0.0048) updates. We attribute this to two factors derived from our pullback analysis:

- **Effectiveness of Diagonal Latent Metrics:** Proposition 3.1 shows that the local pullback geometry in physical coordinates is governed by $J_\phi^\top P_g J_\phi$ near a slice center. Thus, even when P_g is diagonal in latent coordinates, composition with the encoder can induce oriented receptive fields in physical coordinates. This suggests that diagonal latent precision matrices can already provide useful anisotropic geometry, while explicit off-diagonal updates may offer less practical benefit on these benchmarks than their nominal expressivity suggests.
- **Optimization Complexity:** Introducing low-rank updates creates parameter redundancy. Since rotations can be represented either by modifying the encoder weights (changing J_ϕ) or by the precision matrix update ($\mathbf{u}\mathbf{u}^\top$), the model faces a non-unique solution space. This likely creates a more complex optimization landscape with additional saddle points, hindering convergence compared to the strictly constrained diagonal formulation.

Table 9: Ablation study on the precision matrix structure. We compare our standard Diagonal formulation against adding low-rank non-diagonal updates on the **Airfoil** and **Darcy** datasets. The results show that increasing rank complexity yields no performance gain (and even degradation on Airfoil), suggesting that axis-aligned scaling is sufficient and easier to optimize for these standard benchmarks.

Precision Structure	Airfoil	Darcy
Diagonal (Standard)	**0.0045**	**0.0045**
Diagonal + Rank-1 Update	0.0047	**0.0045**
Diagonal + Rank-2 Update	0.0048	0.0046

G.2 Scalability and Robustness Analysis

To evaluate the efficiency and stability of the proposed M-Transolver, we conduct a series of scalability experiments. We analyze the model's sensitivity to three critical hyper-parameters: the resolution of the physical slicing (G), the size of the training dataset (N), and the network depth (L).

G.2.1 Impact of Slice Resolution (G)

We first investigate the impact of the number of slices G, which governs the granularity of the geometric prior. Table 10 compares the Relative L_2 Error (L2RE) of the baseline Transolver against our M-Transolver on the Elasticity and Darcy benchmarks.

Table 10: Ablation study on the scalability with respect to the number of slices G. We compare the performance (Relative L_2) of the baseline Transolver and our M-Transolver on the Elasticity (Point Cloud) and Darcy (Regular Grid) datasets.

Ablations		Transolver L2RE		M-Transolver L2RE	
Number of Slices		Elasticity	Darcy	Elasticity	Darcy
	1	0.0148	0.0386	0.0221	0.0379
	8	0.0071	0.0096	0.0052	0.0054
	16	0.0067	0.0067	0.0048	0.0048
	32	0.0067	0.0063	0.0049	0.0047
	64	0.0064	0.0059	0.0044	0.0045
	96	0.0061	0.0055	0.0044	0.0045
	128	0.0058	0.0054	0.0043	0.0043
	256	0.0054	0.0050	0.0044	0.0043
	512	0.0059	0.0056	0.0041	0.0044

As shown in Table 10, setting $G = 1$ (global attention without slicing) results in high errors for both models, confirming the necessity of the slice-based attention mechanism for physical simulations. As G increases, the error drops significantly. Notably, *M-Transolver consistently outperforms the baseline across all resolutions*, stabilizing at a lower error floor ($\approx 0.43\%$) compared to the baseline ($\approx 0.54\%$) on Elasticity. Furthermore, M-Transolver achieves optimal performance with fewer slices (around $G = 64$), suggesting that the metric-aware mechanism captures critical geometry more efficiently than the isotropic baseline.

G.2.2 Scalability with Training Samples

Data efficiency is a crucial metric for neural operators, as generating high-fidelity simulation data is computationally expensive. We evaluate model performance across training sizes ranging from 200 to 1000 samples.

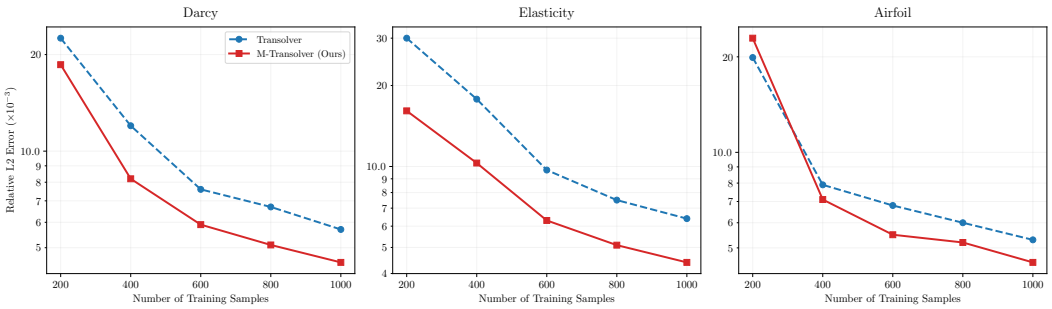


Figure 7: **Data Efficiency Analysis.** Convergence of Relative L_2 Error as the number of training samples increases. M-Transolver (red) exhibits superior data efficiency, achieving significantly lower error rates in the low-data regime (200-400 samples) across all three benchmarks.

The results, visualized in Figure 7, demonstrate that M-Transolver exhibits a steeper convergence rate. In the data-scarce regime (200 samples), our method reduces the error by approximately 17% on

Darcy (0.0186 vs 0.0225) and 46% on Elasticity (0.0161 vs 0.0300). This indicates that the learned anisotropic geometric prior acts as an effective regularizer, allowing the model to generalize well even with limited supervision.

G.2.3 Scalability with Network Depth

Finally, we examine the scalability of the architecture with respect to network depth (number of layers), varying from 8 to 40 layers. This tests the model's ability to propagate physical information without suffering from degradation or over-smoothing.

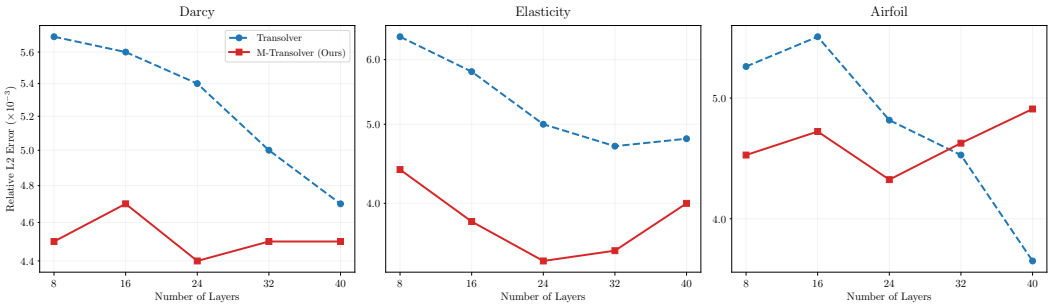


Figure 8: **Depth Efficiency Analysis.** Performance comparison across varying network depths (8 to 40 layers). M-Transolver demonstrates high stability and robustness, maintaining low error rates even with shallower networks, whereas the baseline often requires greater depth to match performance.

As illustrated in Figure 8, M-Transolver is remarkably depth-efficient. On the Elasticity benchmark, our 8-layer model achieves an error of 0.0044, which outperforms the 40-layer baseline model (0.0048). While the baseline Transolver exhibits performance fluctuations at intermediate depths, M-Transolver maintains a stable and consistent error profile. This stability implies that the anisotropic attention mechanism effectively disentangles physical features early in the network, reducing the need for excessive depth to resolve complex flow features.

G.3 Eccentricity and Alignment of Pullback Metrics

To quantitatively analyze the geometric structure of the learned metrics, we examine the shape and orientation of the attention mechanisms induced by M-Transolver. We define the **eccentricity** (e) of an attention slice based on the eigenvalues ($\lambda_1 \geq \lambda_2$) of its spatial covariance matrix:

$$e = \sqrt{1 - \frac{\lambda_2}{\lambda_1}}. \tag{15}$$

An eccentricity close to 0 indicates an isotropic (spherical) receptive field, while a value close to 1 implies a highly anisotropic (needle-like) structure. We further measure the *alignment* as the absolute cosine similarity between the principal axis of the ellipsoid and the local flow velocity vector, where 1.0 indicates perfect alignment with the streamline.

As shown in Table 11, M-Transolver learns highly anisotropic structures, with the top 10% of slices exhibiting a mean eccentricity of 0.933. Crucially, these anisotropic slices are not random; they show a high degree of alignment (0.80) with the physical flow direction. This suggests that the model naturally discovers the underlying physics, learning to aggregate information along streamlines (where advection dominates) rather than isotropically.

Table 11: Geometric analysis of attention slices in the Navier-Stokes dataset. M-Transolver learns highly anisotropic receptive fields that are strongly aligned with the fluid flow velocity.

Tier (by Eccentricity)	Mean Ecc.	Mean Alignment	Count
Top 10% Most Aniso.	0.9330	0.8000	636
Top 25% Most Aniso.	0.8952	0.7567	1591
Top 50% Most Aniso.	0.8454	0.7265	3182
Bottom 10% (Least)	0.4347	0.6613	636

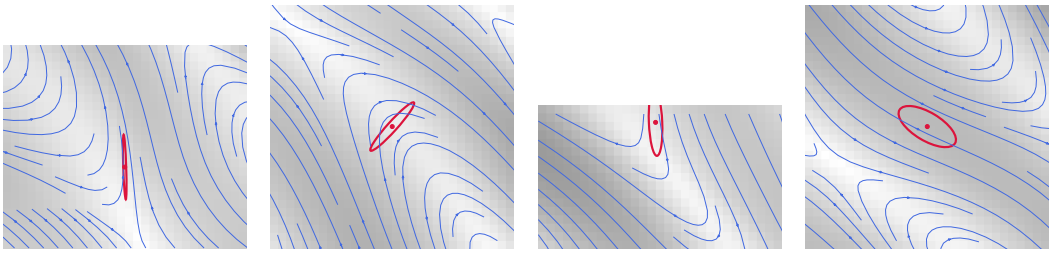


Figure 9: Showcases of the learned pullback metric alignment of M-Transolver. The visualizations demonstrate how the model learns anisotropic receptive fields that align with the physical flow structures.

G.4 Standard Deviations and Robustness

To ensure statistical rigor, we repeated all experiments across four independent runs and report the mean and standard deviations in Table 12. Our proposed method demonstrates robust performance, surpassing the previous state-of-the-art on unstructured domains (Point Cloud and Unstructured Mesh benchmarks) with high confidence. Notably, in complex geometry tasks like ShapeNet-Car and AirfRANS, our method consistently achieves superior correlation coefficients with clear statistical separation from the best baselines.

Table 12: Standard deviations of our model compared to the best previous baselines across all experiments. We report the mean ± standard deviation over 4 independent runs. Spearman's rank correlations for AirfRANS and ShapeNet-Car are presented as percentages (scaled by 100). All other values are relative errors ($\times 10^{-2}$). The best baseline model for each task is indicated in parentheses.

	Point Cloud			Structured Mesh		Reg. Grid	Unstructured Mesh	
Model	Elasticity	Plasticity	Airfoil	Pipe	Navier-Stokes	Darcy	ShapeNet-Car	AirfRANS
Best Baseline	$0.50_{\pm 0.021}$ (LaMO)	$0.07_{\pm 0.008}$ (LaMO)	$0.41_{\pm 0.013}$ (LaMO)	$0.27_{\pm 0.026}$ (Transolver++)	$7.19_{\pm 0.072}$ (Transolver++)	$0.49_{\pm 0.025}$ (Transolver++)	$99.15_{\pm 0.01}$ (Transolver)	$99.83_{\pm 0.010}$ (LaMO)
Ours	$\mathbf{0.44}_{\pm 0.015}$ (M-Transolver)	$\mathbf{0.03}_{\pm 0.006}$ (M-LaMO)	$\mathbf{0.37}_{\pm 0.017}$ (M-LaMO)	$\mathbf{0.23}_{\pm 0.028}$ (M-Transolver)	$\mathbf{6.78}_{\pm 0.17}$ (M-Transolver)	$\mathbf{0.45}_{\pm 0.01}$ (M-Transolver)	$\mathbf{99.38}_{\pm 0.025}$ (M-Transolver)	$\mathbf{99.94}_{\pm 0.010}$ (M-LaMO)

G.5 Probabilistic Interpretation and Limitations of Diagonal Precision

Gaussian-mixture interpretation. METRO employs a diagonal precision matrix for each latent slice, which admits a probabilistic interpretation conditioned on the learned feature representation. Let $z_i = \phi(x_i) \in \mathbb{R}^d$ denote the learned embedding of input point i. For each slice g, let $\mu_g \in \mathbb{R}^d$ be its learned center and $P_g = \mathrm{diag}(p_{g,1}, \ldots, p_{g,d}) \succ 0$ its diagonal precision matrix. The routing logit is

$$\ell_{ig} = -\frac{1}{2}(z_i - \mu_g)^\top P_g (z_i - \mu_g) + \frac{1}{2}\log\det P_g. \tag{16}$$

Under equal mixture priors, this is the Gaussian log-density of component g, up to a constant shared across components. Consequently, the row-normalized assignment weights

$$\psi_{ig} = \frac{\exp(\ell_{ig})}{\sum_{h=1}^{G}\exp(\ell_{ih})} = p(g \mid z_i) \tag{17}$$

coincide exactly with posterior responsibilities of a Gaussian mixture with means μ_g and covariance matrices P_g^{-1}.

The slice representation

$$Z_g = \frac{\sum_i \psi_{ig} v_i}{\sum_i \psi_{ig}} \tag{18}$$

is therefore the empirical responsibility-weighted mean of the value features v_i. Reconstruction through the corresponding assignment weights can likewise be viewed as mixture-based interpolation of the transformed slice representations.

This interpretation is conditional rather than generative. METRO is optimized discriminatively through the downstream operator-learning objective; it does not assume that the learned embeddings are generated by a Gaussian mixture, nor does it maximize a Gaussian-mixture likelihood.

Axis alignment and expressivity. Restricting P_g to be diagonal introduces an axis-alignment inductive bias. All slices measure distances using diagonal metrics in the same learned coordinate system, although their diagonal entries may differ. Thus, the embedding network must learn a representation in which the routing-relevant correlations within different slices are approximately aligned with the coordinate axes.

For multiple slices, this requirement is analogous to approximate joint diagonalization of their local covariance structures. A nonlinear embedding can substantially reshape the input geometry and may adapt its local orientation across the domain. Nevertheless, a single shared embedding need not simultaneously diagonalize incompatible local covariance structures, particularly where slices overlap. In such cases, diagonal precision may be less expressive than a full-covariance metric.

The restriction is deliberate: diagonal precision reduces the number of metric parameters from $O(Gd^2)$ to $O(Gd)$ and avoids the additional parameterization and optimization complexity of full positive-definite matrices.

Precision behavior and potential failure modes. The Gaussian interpretation also provides insight into the behavior of the learned precision parameters. Expanding Eq. (16) gives

$$\ell_{ig} = -\frac{1}{2}\sum_{k=1}^{d} p_{g,k}(z_{i,k} - \mu_{g,k})^2 + \frac{1}{2}\sum_{k=1}^{d} \log p_{g,k}. \tag{19}$$

For comparison, under conventional Gaussian maximum-likelihood estimation with fixed responsibilities, the stationary diagonal variance estimate satisfies

$$p_{g,k}^{-1} = \frac{\sum_i \psi_{ig}(z_{i,k} - \mu_{g,k})^2}{\sum_i \psi_{ig}}, \tag{20}$$

where μ_g is the corresponding responsibility-weighted mean. Although METRO does not optimize this likelihood, the expression illustrates the geometric bias of Gaussian routing: higher precision corresponds to tighter localization along a coordinate direction.

If the learned embedding cannot provide a suitable axis-aligned representation, several failure modes are possible. The model may approximate incompatible orientations using overlapping axis-aligned components, distort the embedding to accommodate the routing metric, or produce highly concentrated components covering only a small number of input points. Components receiving little assignment mass may also contribute weakly to the downstream prediction and receive limited task gradients, resulting in underutilized slices.

The log-determinant term mitigates a specific degeneracy of the distance-only formulation. Without this term, taking $P_g \to 0$ eliminates the quadratic distance penalty, allowing an arbitrarily diffuse component to obtain a relatively favorable routing score. With Gaussian normalization,

$$\ell_{ig} = -\frac{1}{2}(z_i - \mu_g)^\top P_g (z_i - \mu_g) + \frac{1}{2}\log\det P_g \longrightarrow -\infty \quad \text{as } P_g \to 0, \tag{21}$$

for fixed z_i and μ_g. Thus, increasingly diffuse components are penalized rather than rewarded. However, this normalization does not guarantee balanced slice utilization, prevent highly concentrated assignments, or eliminate poorly utilized components.

Relation to heteroscedastic Gaussian optimization. These failure modes should be distinguished from those identified by Seitzer et al. (2022) for predictive heteroscedastic Gaussian negative log-likelihood. In that setting, predicted variance directly reweights the mean-prediction gradient, potentially causing poorly fitted observations to receive insufficient emphasis. METRO does not predict output uncertainty and is not trained using Gaussian negative log-likelihood. Therefore, that particular gradient pathology does not directly apply. The analogous concern in METRO is *responsibility starvation*: a slice receiving little assignment mass may have limited influence on the operator output and consequently receive weak downstream gradients.

Practical implications. Diagonal precision therefore trades metric expressivity for parameter efficiency and optimization simplicity. Our existing low-rank precision ablation on Airfoil and Darcy indicates that diagonal metrics are sufficient and easier to optimize in these settings. However, these results do not establish that diagonal precision is optimal across all geometries, particularly complex three-dimensional flows with strongly varying local orientations. More expressive precision parameterizations and explicit mechanisms for mitigating slice underutilization remain possible directions for future investigation.

H Limitations and Future Work.

While effective, METRO presents limitations that motivate future work. First, the diagonal parameterization of the precision matrix restricts expressivity in complex 3D turbulent flows requiring full off-diagonal correlations. Stabilizing the training of full covariance matrices (e.g., via Riemannian gradient descent) remains an open challenge. Second, METRO learns static spatial metrics $\mathbf{P}_g$, which may struggle to track transient structures in highly unsteady flows, highlighting a need for time-dependent matrices $\mathbf{P}_g(t)$. Finally, METRO operates at a single spatial scale. Integrating it as a boundary-aware frontend for $O(N)$ architectures like LinearNO (Hu et al., 2025), or into hierarchical pooling operations like GAOT (Wen et al., 2025), is necessary to achieve true multi-scale geometric adaptivity.

H.1 Relation to Adaptive-Mesh Methods

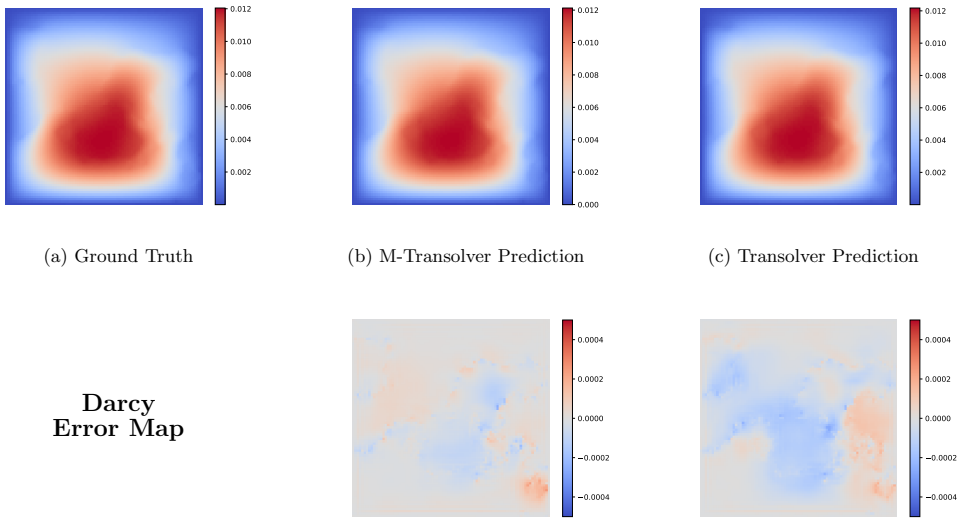
Adaptive-mesh methods and METRO address complementary aspects of PDE learning. Adaptive meshing modifies the physical discretization to better resolve solution features. In particular, r-adaptivity relocates mesh nodes, as in M2N Song et al. (2022), MM-PDE Hu et al. (2024), and G-Adaptivity Rowbottom et al. (2025), whereas h-adaptivity changes local resolution through refinement or coarsening, as in learned adaptive mesh refinement Yang et al. (2023).

In contrast, METRO operates on the supplied discretization without modifying physical node positions, connectivity, or mesh resolution. Its learned anisotropic routing determines how existing samples are aggregated into a smaller set of latent tokens. Thus, adaptive meshing addresses *where the physical domain is sampled*, while METRO addresses *how the sampled information is represented and processed*.

These approaches are therefore complementary. For example, an adaptive mesh could concentrate physical samples near shocks or interfaces, after which METRO could organize these samples through geometry-aware latent routing. METRO is also applicable when remeshing is unavailable or the discretization is fixed. The graph-based baselines considered in our experiments operate on supplied discretizations and are not intended to represent the broader adaptive-mesh literature. Since a direct comparison would conflate discretization adaptation with latent representation learning, evaluating their combination is an interesting direction for future work.

I Extended Qualitative Results

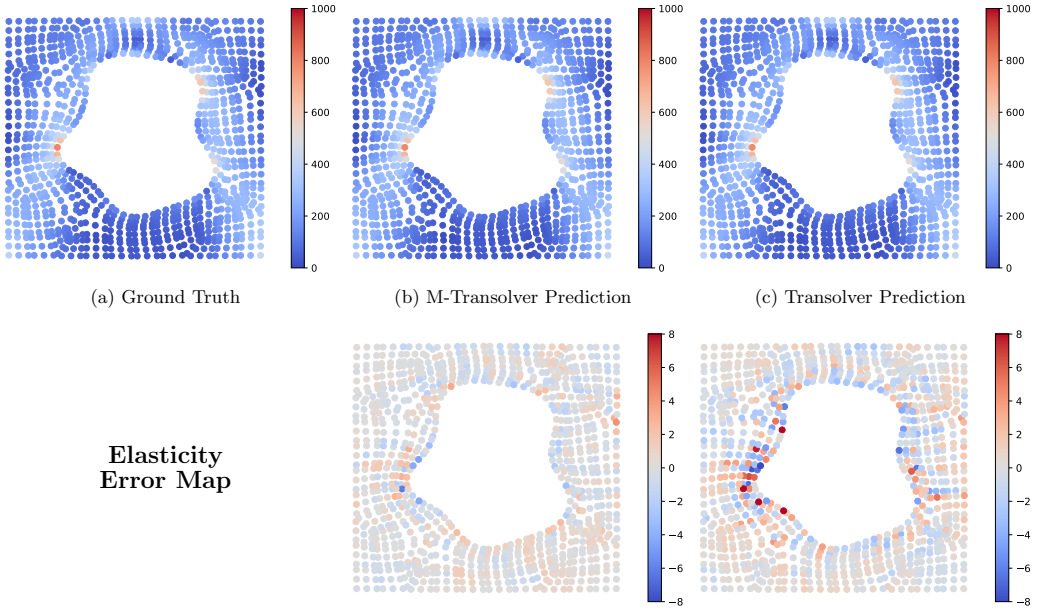
When interpreting the learned attention topologies across different physical systems, it is important to note that the degree of visual anisotropy depends directly on the governing partial differential equations. Problems dominated by advection and strong directional transport, such as fluid dynamics (Airfoil, AirCraft, Navier-Stokes), naturally exhibit highly elongated, streamline-aligned features. Consequently, the slices learned by M-Transolver in these regimes are starkly anisotropic. Conversely, problems governed by more diffusive or structural mechanics behaviors, such as Elasticity and



(a) Ground Truth (b) M-Transolver Prediction (c) Transolver Prediction

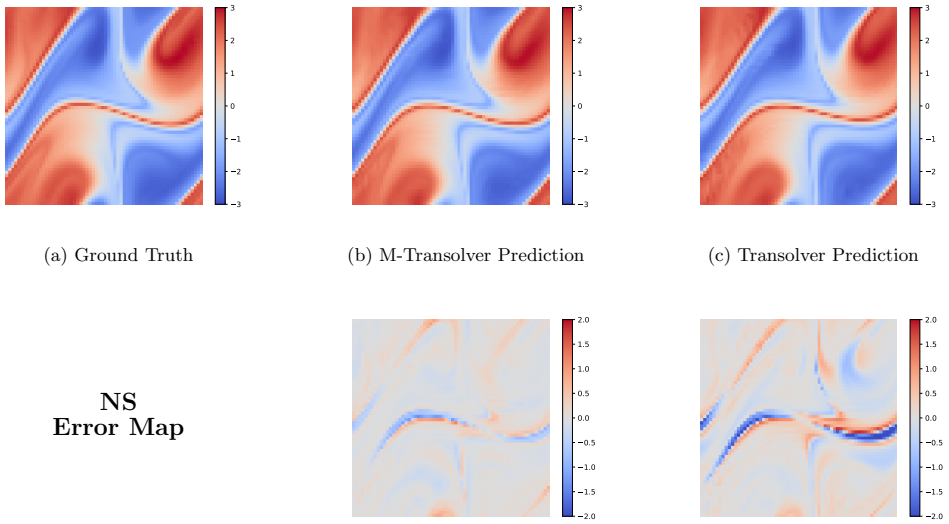
Figure 11: Prediction and error visualization of M-Transolver vs. Transolver on the Darcy flow problem.

Plasticity, lack these sharp directional gradients. In these domains, physical information propagates more uniformly, and M-Transolver correctly adapts by learning less eccentric, more structurally distributed slices. This variability suggests that the routing mechanism does not simply impose an artificial geometric prior, but adapts flexibly to the intrinsic physical nature of each dataset. To ensure a fair and representative comparison, all visualizations provided in this section display randomly selected samples from the test set, confirming that these geometric adaptations are consistent global behaviors rather than cherry-picked anomalies.



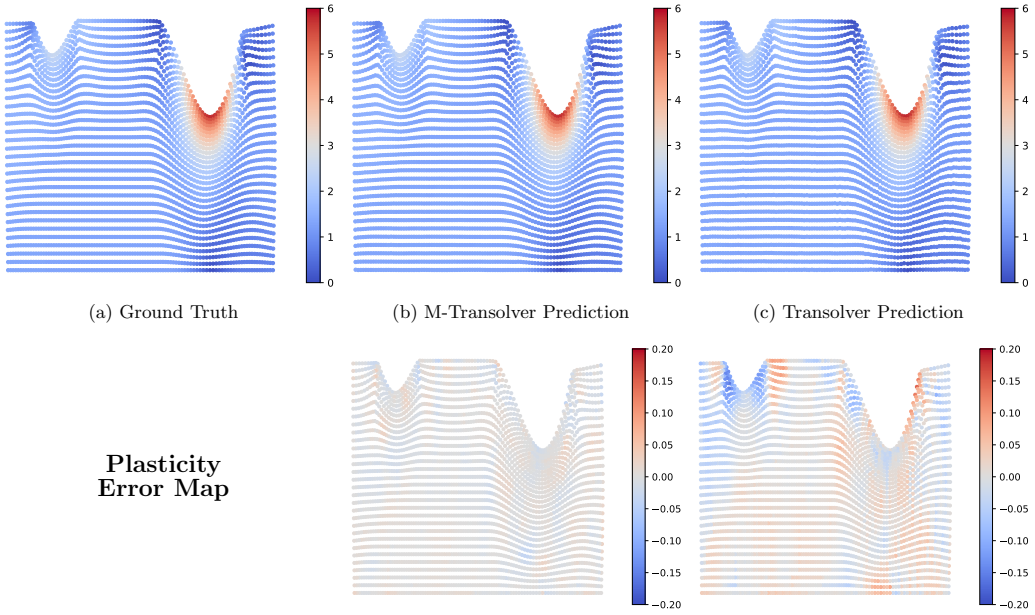
(a) Ground Truth (b) M-Transolver Prediction (c) Transolver Prediction

Figure 10: Prediction and error visualization of M-Transolver vs. Transolver on the elasticity problem.



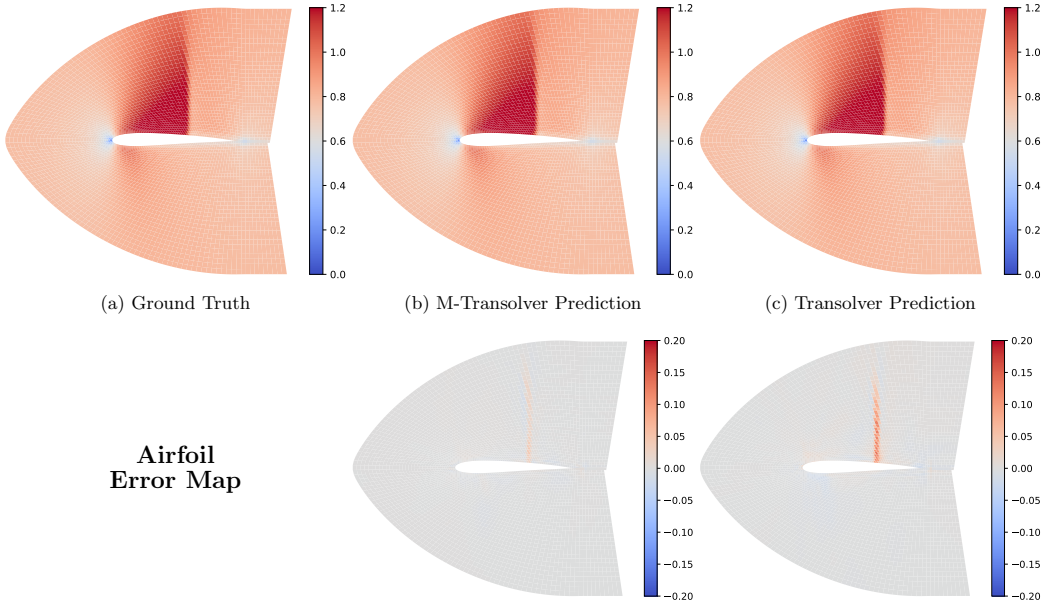
(a) Ground Truth (b) M-Transolver Prediction (c) Transolver Prediction

Figure 12: Prediction and error visualization of M-Transolver vs. Transolver on the Navier-Stokes problem.



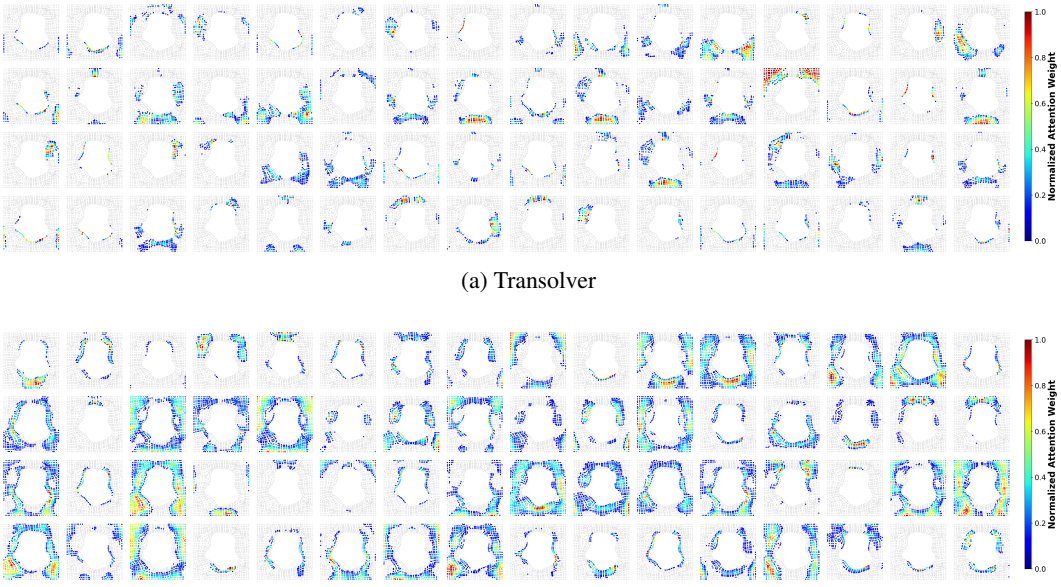
(a) Ground Truth (b) M-Transolver Prediction (c) Transolver Prediction

Figure 13: Prediction and error visualization of M-Transolver vs. Transolver on the plasticity problem.



(a) Ground Truth (b) M-Transolver Prediction (c) Transolver Prediction

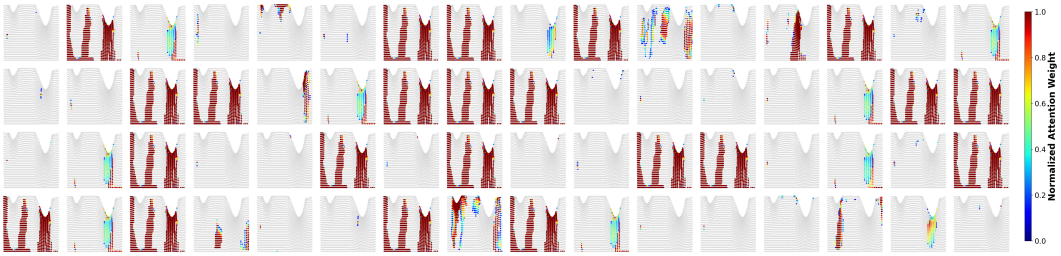
Figure 14: Prediction and error visualization of M-Transolver vs. Transolver on the airfoil problem.



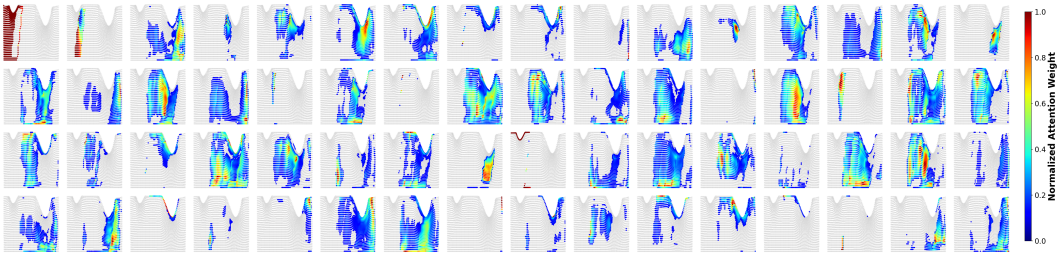
(a) Transolver

(b) M-Transolver

Figure 15: Comparison of slices learned by Transolver (top) and M-Transolver (bottom) for the elasticity problem.

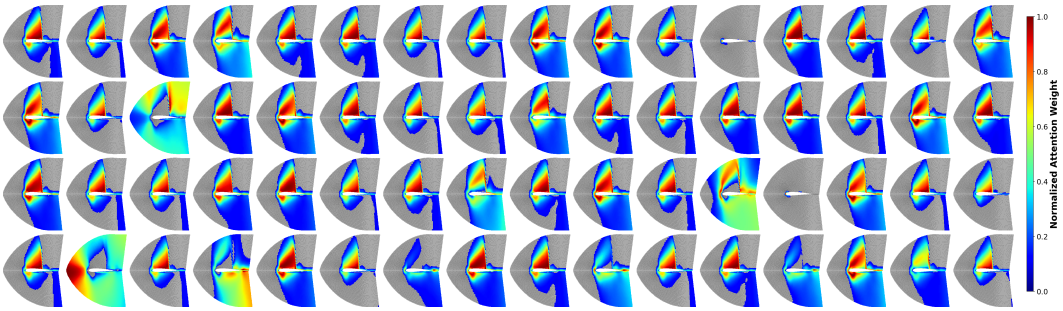


(a) Transolver

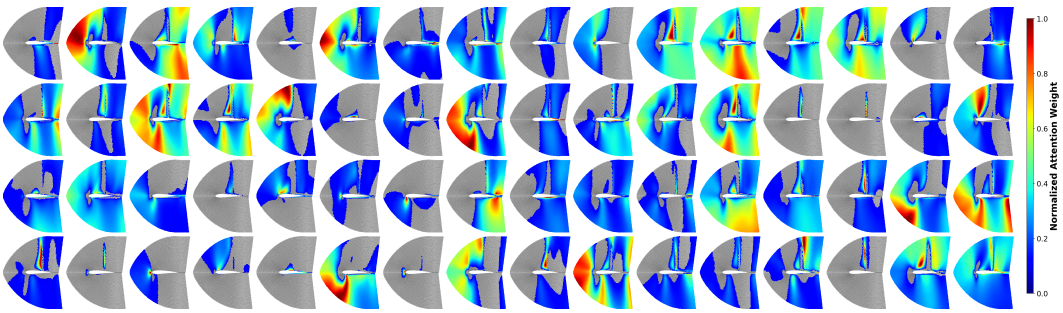


(b) M-Transolver

Figure 16: Comparison of slices learned by Transolver (top) and M-Transolver (bottom) for the plasticity problem.

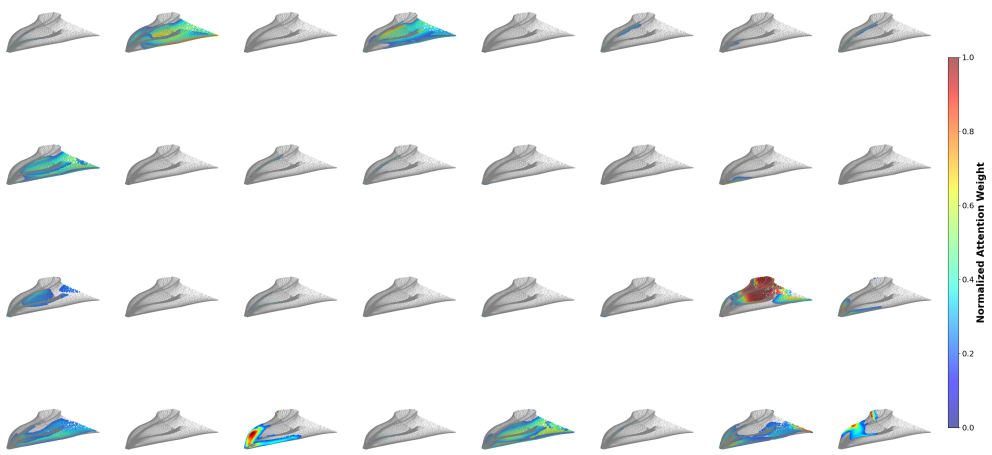


(a) Transolver

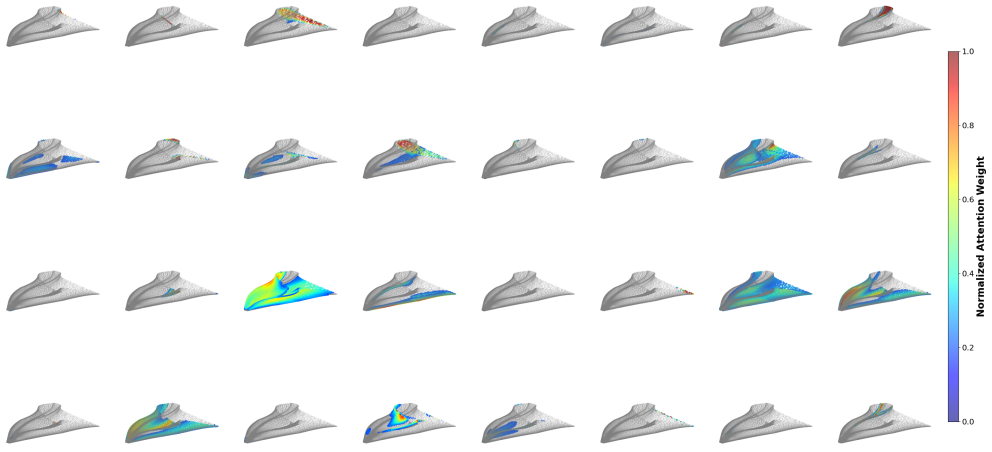


(b) M-Transolver

Figure 17: Comparison of slices learned by Transolver (top) and M-Transolver (bottom) for the airfoil problem.



(a) Transolver



(b) M-Transolver

Figure 18: Comparison of slices learned by Transolver (top) and M-Transolver (bottom) for the AirCraft problem.

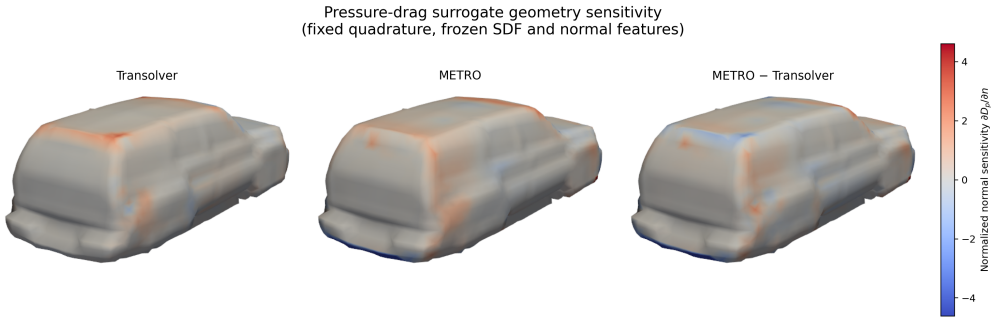
I.1 Qualitative Predictions on ShapeNet-Car

To complement the quantitative evaluation, we visualize predictions from Transolver and M-Transolver (METRO) on two randomly selected held-out ShapeNet-Car geometries. Figures 19 and 20 present geometry sensitivity, surface-pressure predictions, and center-plane streamwise-velocity predictions. The visualizations include ground-truth and predicted fields, absolute-error maps, and direct error-reduction maps. Both models use identical geometries, inputs, normalization, postprocessing, and plotting scales. Reported RMSE values are compared only within the same physical field.

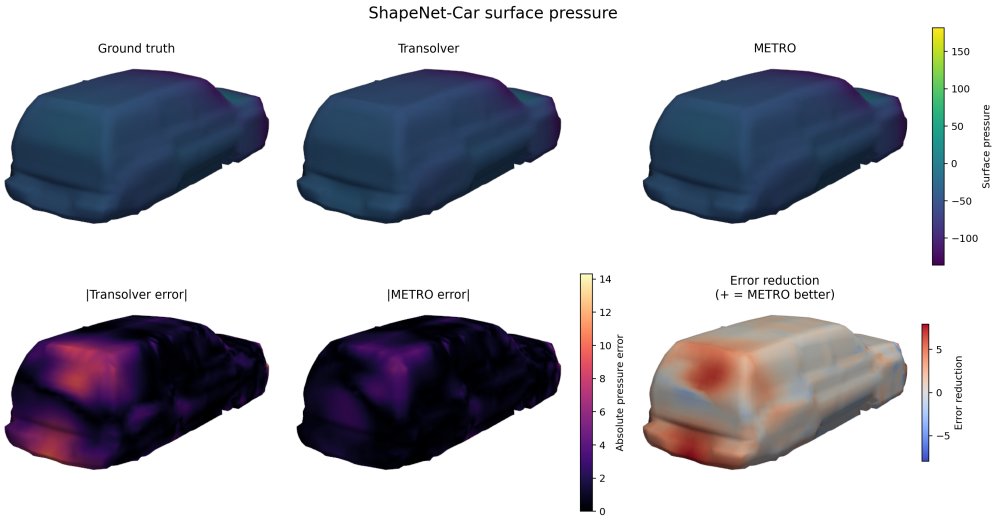
For Example A (Fig. 19), surface-pressure RMSE decreases from 4.358 for Transolver to 3.805 for METRO, corresponding to a 12.7% reduction. The center-plane streamwise-velocity RMSE decreases from 0.692 to 0.252, a 63.6% reduction. The largest visible velocity improvements occur downstream and near the vehicle, while pressure errors are reduced across several spatially coherent surface regions.

For Example B (Fig. 20), surface-pressure RMSE decreases from 3.370 to 2.609, a 22.6% reduction. The center-plane streamwise-velocity RMSE decreases from 0.319 to 0.213, a 33.3% reduction. The error-reduction maps also identify regions where Transolver performs better locally, demonstrating that the improvements are not uniform pointwise.

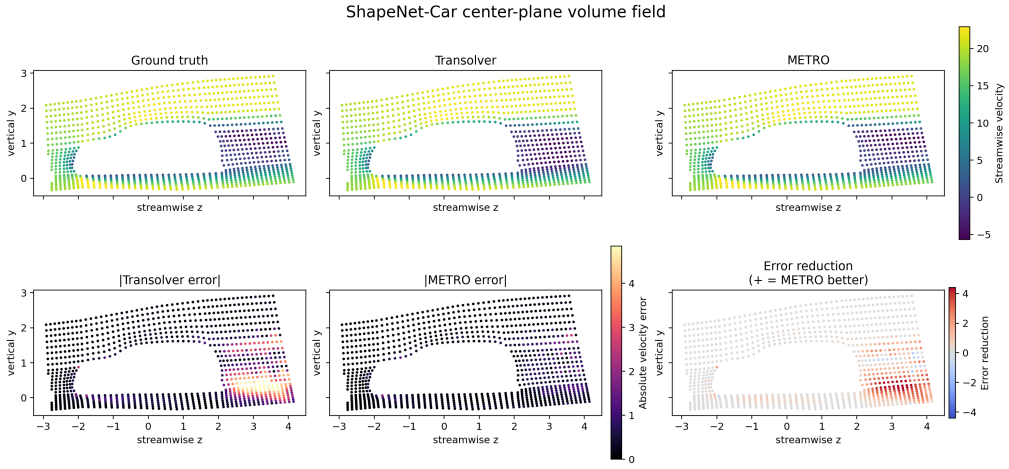
Across both examples, METRO preserves the principal near-body and downstream flow structures while reducing prediction errors in several coherent spatial regions. These qualitative results complement the aggregate metrics by showing where the improvements occur, rather than attributing them to a global rescaling of the predictions.



(a) Geometry sensitivity.

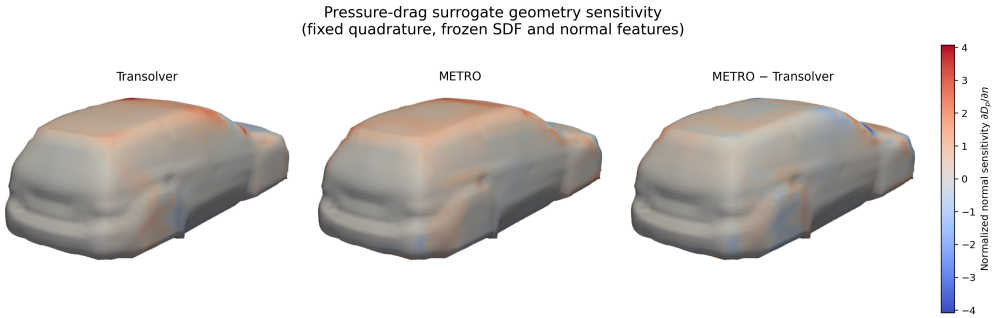


(b) Surface-pressure predictions and errors.

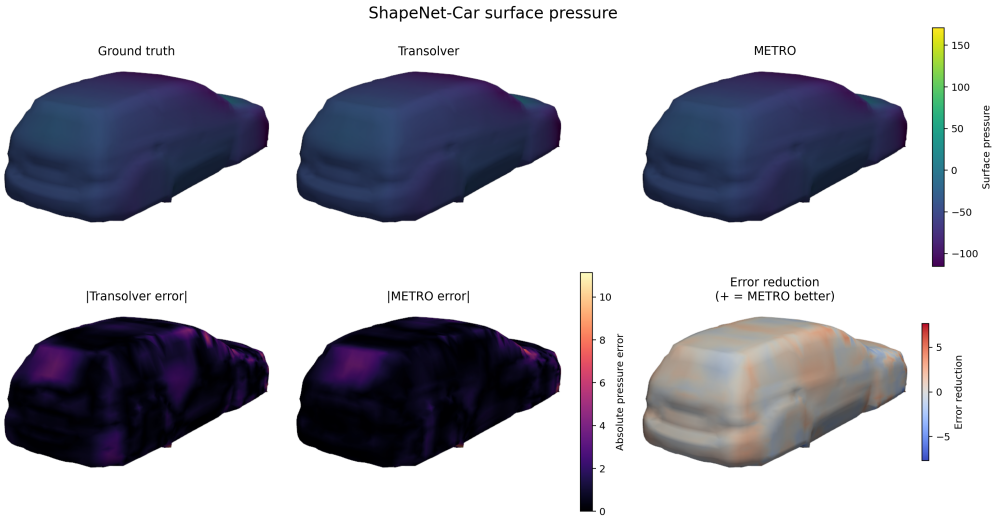


(c) Center-plane streamwise-velocity predictions and errors.

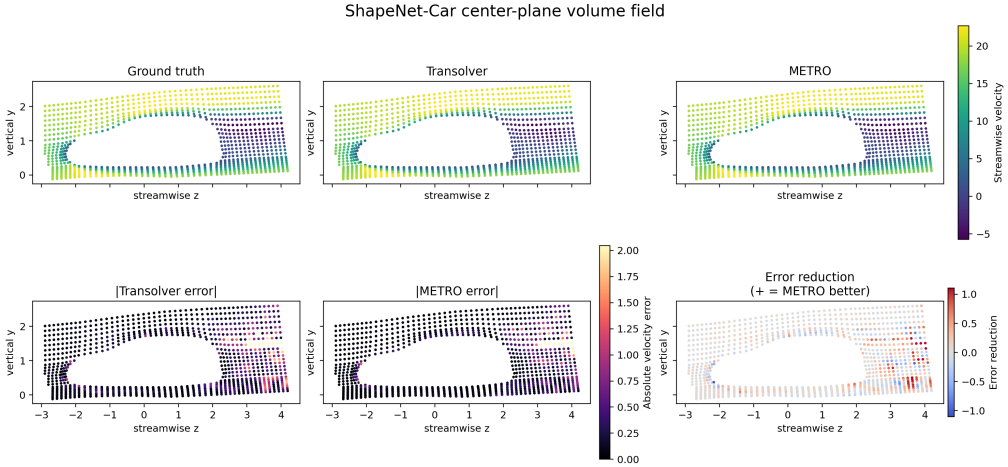
Figure 19: **Qualitative comparison on ShapeNet-Car (Example A).** Comparison of Transolver and METRO on a randomly selected held-out vehicle geometry. Visualizations include geometry sensitivity, surface-pressure predictions, and center-plane streamwise velocity, together with absolute-error and error-reduction maps. Surface-pressure RMSE decreases by 12.7%, while streamwise-velocity RMSE decreases by 63.6%. All corresponding fields use identical plotting scales across models.



(a) Geometry sensitivity.



(b) Surface-pressure predictions and errors.



(c) Center-plane streamwise-velocity predictions and errors.

Figure 20: **Qualitative comparison on ShapeNet-Car (Example B).** Comparison of Transolver and METRO on a second randomly selected held-out vehicle geometry. Surface-pressure RMSE decreases by 22.6%, while streamwise-velocity RMSE decreases by 33.3%. The error-reduction maps show both positive and negative regions, indicating local improvements and regressions rather than uniformly improved predictions. Corresponding fields use identical plotting scales across models.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract faithfully summarizes the core theoretical contributions of the paper. It clearly reflects the paper's scope, avoids overstatements, and aligns with the formal results presented in the main body.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: We explicitly discuss the limitations of our work and outline directions for future research in the Limitations and Future Work section in Appendix I.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: All theoretical results are accompanied by a complete set of assumptions and formal proofs, which are included in the appendix. In the main text, we provide intuitive sketches to guide the reader and connect the theory to its implications.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: While code and data are not released at submission time due to anonymity constraints, we document all necessary experimental details in the paper and appendix to support reproducibility. Code will be released upon acceptance to ensure full transparency and verifiability.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: We plan to release the code and instructions to reproduce the main results upon acceptance. At submission time, the code and data are not publicly available to preserve anonymity.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide comprehensive information on datasets, training/test splits, model architecture, hyperparameters (e.g., hidden size, optimizer choice, learning rate, training epochs), and all baselines in the technical appendix to facilitate understanding and replication.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All primary experimental results are reported as the mean $\pm$ standard deviation calculated across 4 independent trials using different random seeds.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We specify the type of GPUs we used, where the memory size can be directly inferred in the technical appendix. We also conducted parameter-matched experiments to compare time of execution.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our research focuses on applied mathematical methods for solving partial differential equations (PDEs) and does not involve human subjects, sensitive data, or applications with foreseeable misuse potential. We have reviewed the NeurIPS Code of Ethics and confirm that our work aligns with its principles, including considerations related to research ethics, data usage, and potential societal impact.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

Answer: [Yes]

Justification: We discuss the societal implications of METRO in the appendix. On the positive side, our highly efficient surrogate model can significantly reduce the computational energy footprint of industrial engineering simulations (e.g., vehicle aerodynamics) and accelerate the development of sustainable technologies. Regarding negative impacts, we acknowledge the risks of deploying deep learning-based PDE solvers in safety-critical applications; inaccurate predictions on out-of-distribution geometries could lead to flawed physical designs if used as a complete replacement for traditional solvers without formal error bounds.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: Our research develops neural operators for solving partial differential equations (PDEs) in physical simulations, such as fluid dynamics and structural mechanics. These models and the associated scientific datasets do not possess the high risk for malicious misuse (e.g., disinformation, toxic generation, privacy violations) associated with large language models or image generators, rendering specific release safeguards for those risks inapplicable.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We properly credit the creators of all datasets (e.g., AirfRANS, ShapeNet) and baseline models by citing their original publications. Details regarding the dataset origins, references, and their respective terms of use/licenses are provided in Appendix G and the supplementary material.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: NA

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: Our research does not involve human subjects or crowdsourcing. It is entirely theoretical and computational, focused on applied mathematics and machine learning methods.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: Our research does not involve human subjects or crowdsourcing. It is entirely theoretical and computational, focused on applied mathematics and machine learning methods.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: We did not use LLMs in the design or implementation of the core methods in this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.