COORDINATES ARE LABELS, NOT PHYSICS: BREAKING THE COORDINATE SHORTCUT IN NEURAL SURROGATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural surrogates approximate physical fields at a lower computational cost than numerical simulation. On aligned vehicle benchmarks, a shared pose and fixed inflow direction allow models to predict field values from absolute coordinates, a strategy we call the *coordinate shortcut*. Low error on these benchmarks therefore does not show whether a model has learned how the field responds to geometry and boundary conditions. We evaluate surrogates beyond this setting, under unseen inflow directions and on geometries from unrelated shape families. Removing coordinate inputs prevents the shortcut but also discards spatial information about the arrangement of points within a geometry. We propose **GraFINS** (Graph-Encoded Frame-Invariant Neural Surrogate), which replaces coordinate inputs with frame-invariant features encoding local geometry and its relation to boundary conditions, and a graph embedding encoding spatial information. Trained only on vehicles with flow from the front, GraFINS predicts pressure under side and rear inflow more accurately than equivariant and non-equivariant baselines given the same features, and without retraining it also outperforms them on non-vehicle geometries. It preserves in-distribution accuracy on automotive and patient-specific blood-vessel benchmarks and remains stable under scene rotations.

1 INTRODUCTION

Data-driven neural surrogates promise to replace hours of numerical simulation of physical systems with a single forward pass. Transformer- and operator-based models report low prediction errors on vehicle simulations (Wu et al., 2024; Luo et al., 2025; Alkin et al., 2025; Ranade et al., 2025). However, their success raises a question about prediction beyond those benchmarks. Numerical solvers assemble fluxes from face normals, cell volumes, distances between neighbouring cell centres and boundary conditions (Jasak, 1996; Moukalled et al., 2015). Many neural surrogates instead take absolute coordinates, the raw (x, y, z) positions in the dataset's frame, to learn the spatial relationships needed for prediction. However, coordinates also provide a shortcut, allowing models to predict field values from coordinates alone. Low error therefore does not show whether a model has learned how the field responds to changes in geometry and boundary conditions.

Standard evaluations on aligned benchmarks allow this shortcut. DrivAerML (Ashton et al., 2024b), AhmedML (Ashton et al., 2024a) and ShapeNet-Car (Umetani & Bickel, 2018; Chang et al., 2015) each place geometries in a shared pose with a fixed inflow direction, making coordinates predictive of local field values across cases. A surrogate can therefore predict field values partly from coordinates rather than from geometry and boundary conditions. We call this the *coordinate shortcut*, an instance of shortcut learning (Geirhos et al., 2020). Our probe shows that coordinates alone are predictive of field values on aligned automotive data, but not on patient-specific blood vessels (Li et al., 2025), which vary in shape across patients and do not share a common pose.

We therefore ask whether a surrogate can predict beyond the aligned vehicle setting that supports the coordinate shortcut. *Under unseen inflow, can it predict the field when only the inflow direction changes relative to a fixed geometry? Under arbitrary geometries, can a model trained only on vehicles predict on non-vehicle geometries from entirely different shape families without retraining?*

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

Unseen inflow holds the geometry fixed but changes the boundary condition, so the field at the same point may change. Arbitrary geometries replace vehicles with unrelated shape families, so coordinates that were predictive on vehicles need not remain predictive. Low prediction error in the presence of the coordinate shortcut is not, by itself, sufficient evidence of accurate prediction under unseen inflow or on arbitrary geometries. To ensure that the model cannot rely on this shortcut, we exclude coordinates from its learnable inputs.

Removing coordinates, however, also discards useful *spatial information*, since coordinate differences encode how points are arranged relative to one another. Baseline accuracy on aligned benchmarks decreases substantially when this information is not preserved. To retain the spatial information required for accurate prediction without supplying coordinates to learnable layers, we introduce **GraFINS** (Graph-Encoded Frame-Invariant Neural Surrogate), which combines graph-encoded spatial information with frame-invariant features.

Spatial information with a graph. We construct a graph to represent how points are arranged within the geometry rather than where they lie in the dataset's frame. Features pooled at sampled anchors are updated through message passing between neighbouring anchors and interpolated to each query point. These *graph embeddings* replace coordinate inputs to the surrogate; no learnable layer receives coordinates. **Frame-invariant features.** At each point, we encode local geometry and its relation to boundary conditions through geometric properties, scalar distances, and alignments with the inflow and boundaries. These features are invariant under joint rotations of geometry and boundary conditions, but sensitive to changes in boundary conditions relative to the geometry.

Breaking the coordinate shortcut in this way changes what the model can learn. **Unseen inflow.** Trained on a single inflow direction, GraFINS predicts fields under unseen inflow directions more accurately than equivariant and non-equivariant baselines given the same frame-invariant features. **Arbitrary geometries.** Without retraining, it outperforms these baselines on synthetic geometries from shape families entirely unrelated to the training vehicles, both when only the geometry changes and when the inflow changes as well. It also achieves high in-distribution accuracy on automotive and vascular benchmarks. Under a *rotated frame*, where geometry and inflow rotate together, GraFINS remains stable without retraining or augmentation.

Our contributions are summarized as follows.

1. We identify the *coordinate shortcut* by measuring how well coordinates alone predict the field and testing surrogate accuracy beyond the geometries and inflow directions seen during training.
2. We introduce GraFINS, which replaces coordinate inputs without discarding their spatial role, through frame-invariant features and a graph embedding that supplies spatial information.
3. We show that GraFINS preserves in-distribution accuracy, improves generalization to unseen inflow and arbitrary geometries without retraining, and remains stable under scene rotations.

Related work. Neural operators, mesh graph networks, and transformer surrogates provide different representations for field prediction on complex geometries (Li et al., 2023b; Pfaff et al., 2021; Wu et al., 2024), while equivariant surrogates incorporate rotational symmetries into their architectures (Kim et al., 2026; Rygiel et al., 2026). Our study examines the predictive role of coordinate inputs and generalization under unseen inflow and on arbitrary geometries. Further discussion of surrogate architectures, geometry representations, and invariance is provided in App. A.1.

2 The Coordinate Shortcut

The automotive benchmarks share a common pose and inflow direction, whereas Aneumo contains patient-specific vascular geometries without a shared pose (Table 1, Fig. 15). We examine the *coordinate shortcut* by quantifying the predictive value of coordinates at the dataset and model levels. We first evaluate field prediction from coordinates alone without training a neural surrogate, then compare surrogates trained with different inputs to assess the contribution of coordinate inputs to prediction accuracy.

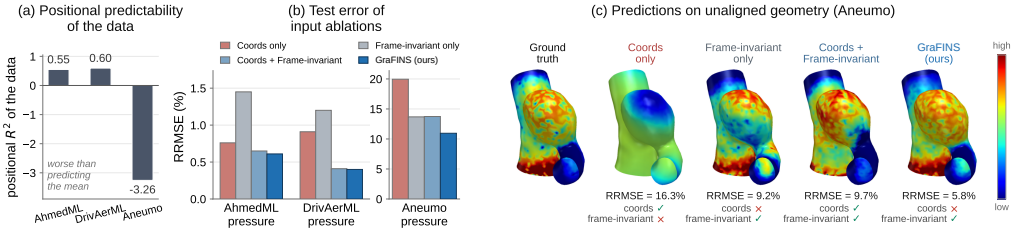


Figure 1: **Diagnosing the coordinate shortcut.** **(a)** Case-averaged R^2 of pressure prediction from other cases using coordinates alone (kNN). **(b)** Test-set mean RRMSE (%, ↓) in the native frame: ■ Coords only, ■ Frame-invariant features only, ■ Coords + frame-invariant features, and ■ GraFINS. **(c)** Predictions and ground truth for an Aneumo case.

How predictive are coordinates alone? Before training any surrogate, we assess whether field values at a given position can be predicted from nearby coordinates in other cases. For each case, we predict pressure using the k=8 nearest sampled points from all other cases, with neighbours selected by coordinate distance and pressure values averaged using inverse-distance weights. This procedure is repeated for all 48 cases. The probe uses coordinates alone, without geometry or boundary-condition features, to predict the pressure coefficient on AhmedML, surface pressure on DrivAerML, and non-dimensional pressure on Aneumo. We compute R^2 for each case and report its mean across cases (App. B). The mean R^2 is 0.55 on AhmedML and 0.60 on DrivAerML, compared with -3.26 on Aneumo (Fig. 1a). Coordinates alone therefore explain more than half of the field variance on average across cases in the two aligned datasets, whereas the probe performs worse than predicting each case's mean on Aneumo. Coordinates thus provide predictive information across cases on the aligned automotive benchmarks.

Table 1: Pose and inflow conventions.

Dataset	Domain	Shared pose	Inflow variation
DrivAerML	aerodynamics	yes	none
AhmedML	aerodynamics	yes	none
ShapeNet-Car	aerodynamics	yes	none
Aneumo	hemodynamics	**no**	flow rate

Do trained models show the same pattern? To assess how coordinate inputs affect prediction accuracy, we train three Transolver variants under identical training settings, varying only their per-point inputs. *Coords only* uses coordinates. *Frame-invariant features only* uses scalars that describe the local geometry and its relation to the boundary conditions, such as curvature and alignment with the inflow, and that are unchanged under a joint rotation of the geometry and the boundary conditions. *Coords + frame-invariant features* uses both. In Fig. 1b, *Coords only* yields lower error than *Frame-invariant features only* on the aligned datasets, but higher error on Aneumo. Relative to *Frame-invariant features only*, *Coords + frame-invariant features* reduces error on AhmedML and DrivAerML, but slightly increases it on Aneumo. With the same features and backbone, GraFINS yields the lowest error on all three.

Figure 1c illustrates the Aneumo comparison for one case: *Coords only* captures less spatial detail than *Frame-invariant features only* in this case. This agrees with the weak positional predictability measured by the data probe on Aneumo. These results are consistent with coordinates being useful under shared alignment, but less useful when shapes and orientations vary across cases.

3 Methodology

Overview. Given query points $p_i \in \mathbb{R}^3$ on a surface or in a volume, GraFINS predicts a field value $\hat{y}_i$ at each point from the geometry and boundary conditions (Fig. 2). We construct a frame-invariant feature vector φ_i at each point and sample anchor points at L scales. Features pooled at the anchors are updated through message passing between neighbouring anchors and interpolated to the query points, producing a graph embedding g_i that provides spatial information in place of coordinate inputs. The backbone f receives the frame-invariant features fused with g_i, together with scalar conditions b. A prediction head combines the backbone output with g_i and φ_i to predict $\hat{y}_i$. Coordinates are used only in fixed computations; no learnable layer receives them. App. A.2 summarizes the notation.

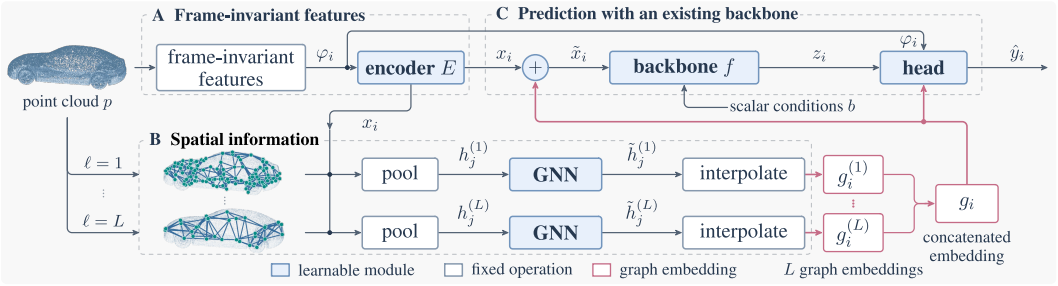


Figure 2: **GraFINS architecture.** (A) A frame-invariant feature vector φ_i at every query point, encoded to the point feature x_i. (B) Anchors at L scales, of which the first and the last are shown. Each scale pools, runs its own graph network and interpolates, and the scales are concatenated into the graph embedding g_i. (C) A gated residual adds g_i before the backbone, and the head reads it again together with φ_i.

3.1 Frame-invariant features from geometry and boundary conditions

Shape alone does not determine the field. Changing the boundary conditions can alter the field even when the geometry remains fixed. We therefore construct φ_i from scalar features describing local geometric properties and their relations to boundary conditions. Each channel is expressed as

$$\varphi_i^{(c)} = \psi_c\big(A_c(p_i),\, B_c\big), \tag{1}$$

where i indexes the query points and c the channels of φ_i. The channels are collected into $\varphi_i \in \mathbb{R}^{C_\varphi}$. The geometric quantity A_c is local, such as the distance d to the body, whose surface is the no-slip wall, or the unit gradient of that distance. B_c specifies a boundary reference, such as the inflow direction, ground normal, or inlet and outlet locations. The map ψ_c converts these quantities into a scalar feature using the channel-specific transformation and normalization. Geometry-only channels, such as proximity and curvature, do not depend on B_c. Each channel is invariant under joint rotations of the geometry and boundary conditions. Channels such as the streamwise position locate a point relative to boundary references, not the dataset frame. Scalar conditions such as inflow speed or mass-flow rate carry no direction or position to pair with, so they are collected in b, one channel per condition, and passed to the backbone instead. App. C.1 lists channels per domain.

3.2 Spatial information from an anchor graph

Equation 1 describes each point independently and does not encode the spatial relationships between points. Two points in distant regions of the geometry can therefore have identical φ_i. The field at each point, however, also depends on where the point lies on the geometry and on the geometry around it, so a network that receives only these features produces the same prediction at both points although the true values differ. Coordinates distinguish such points, so removing them eliminates the shortcut but also this distinction. GraFINS supplements these features with spatial information from an anchor graph, a sparse set of anchors drawn from the query points and connected to their nearest neighbours. We build this graph at several scales, pass messages between neighbouring anchors, and interpolate the anchor states back to the query points, so points with identical φ_i but different neighbourhoods can receive different embeddings.

A shared linear encoder $E\colon \mathbb{R}^{C_\varphi} \to \mathbb{R}^C$ maps every query point to a point feature $x_i = E(\varphi_i) \in \mathbb{R}^C$, where C_φ is the number of frame-invariant channels the domain defines and C the width of the network. At each of the L scales, the graph embedding is then computed in three steps.

(1) Multi-scale anchor selection. The spatial information that anchors provide depends on the number of anchors. A fine scale distinguishes nearby points, whereas a coarse scale indicates where a point lies within the geometry as a whole. Combining scales locates a point within the geometry more precisely than any single scale (App. D.6). We therefore draw S_ℓ anchors $a_1^{(\ell)}, \dots, a_{S_\ell}^{(\ell)} \in \mathbb{R}^3$ from the N query points at each of L scales, with $S_1 > S_2 > \cdots > S_L$. The anchor count decreases exponentially with ℓ, so the neighbourhood size grows from a few points at the finest scale $\ell = 1$ to a large part of the geometry at the coarsest scale $\ell = L$, where S_L is a few dozen.

Within each scale, the query points are partitioned into regions of the geometry that contain equal numbers of points, and the same number of anchors is drawn from each region, so that the anchors cover all regions evenly, even at the coarsest scale. Each scale draws its anchors from the full point cloud independently of the other scales, so a coarse scale is not a subset of a fine one. The anchors are redrawn at every training step, so a case is seen with different anchor sets.

(2) Graph construction. Every query point is assigned to its nearest anchor, which partitions the points into Voronoi cells $\mathcal{V}_j^{(\ell)}$, and the features in each cell are combined by max and mean pooling into the anchor state $h_j^{(\ell)} \in \mathbb{R}^C$. An anchor state thus summarizes the points of its cell rather than one point. An anchor then receives an edge from each of its k nearest anchors at the same scale, and the edge carries the distance between the two anchors.

(3) Graph embedding. The graph embedding of a query point should describe the geometry around it at the neighbourhood size of each scale, so that points with identical φ_i but different surroundings receive different embeddings. At each scale, a graph network passes messages between neighbouring anchors, so the updated state $\tilde{h}_j^{(\ell)}$ of an anchor describes not only its own cell but also the surrounding cells and their distances from it. Each query point interpolates the updated states of its nearest anchors with inverse-distance weights, which gives $g_i^{(\ell)}$. The L scales are concatenated into $g_i = \left[g_i^{(1)}; \ldots; g_i^{(L)}\right] \in \mathbb{R}^{LC}$, which describes the surroundings of a point from its nearest neighbours to the geometry as a whole. The coordinates of a point and the distance between two points can be recovered from a representation containing g_i about as accurately as from models that receive coordinates (Section 4.5, App. D.6).

Each scale has its own weights, and the scales do not exchange messages. App. C.2 gives the implementation of each step, the values of C, L, S_ℓ and k, and the layer settings. Figure 2 shows the first and the last scale.

3.3 Prediction with an existing backbone

GraFINS keeps an existing backbone f unchanged and gives it the frame-invariant features and the graph embedding in place of coordinate inputs. Before the backbone, the graph embedding is added to the point feature x_i as a gated residual,

$$\tilde{x}_i = x_i + \alpha \, \mathrm{LN}(W_g \, g_i) \in \mathbb{R}^C, \tag{2}$$

where $W_g \in \mathbb{R}^{C \times LC}$ is a learned projection, LN is layer normalization and $\alpha \in \mathbb{R}$ is a learned scalar gate. The backbone receives $\tilde{x}_i$ together with the scalar conditions b and returns a state $z_i \in \mathbb{R}^C$ at every point. Because the backbone mixes information across points, the prediction head receives g_i and φ_i again and predicts $\hat{y}_i = \mathrm{Head}([\, z_i;\, g_i;\, \varphi_i \,])$.

The only requirement on f is that it uses no coordinates. Transolver (Wu et al., 2024) and Transolver-3 (Zhou et al., 2026) meet it once coordinates are removed from their input, because they group points into slices by their features rather than by their positions. We use both with their layers unchanged and the output width set to C, and all components are trained jointly.

Test-time averaging. Anchor sampling is stochastic, so at inference we average K predictions from resampled anchor sets. We use $K = 8$ unless stated otherwise. AB-UPT and AB-GATr also sample their anchors at random, and we average their predictions in the same way (App. C.2).

4 Experiments

Benchmarks. On DrivAerML we predict volume pressure, surface pressure and surface wall shear stress (WSS). On AhmedML we predict the surface pressure coefficient C_p and the surface skin-friction coefficient C_f. Both are vehicle aerodynamics benchmarks with every case in one shared pose. On Aneumo we predict volume pressure in cerebral aneurysms, normalized by the peak dynamic pressure of each case, and the cases have no shared pose. On ShapeNet-Car, also in a shared pose, we predict surface pressure with a separate ground truth per inflow direction from our own OpenFOAM runs (App. D.3.1) and train the models for the unseen-inflow and arbitrary-geometry studies (Sections 4.2 and 4.3). The latter evaluates them on surface pressure over fifty geometries

Table 2: **In-distribution accuracy**, grouped by whether the benchmark has a shared pose (rel-L_2 %, ↓). The p and φ columns mark which inputs a model receives, and g in the GraFINS rows is the graph embedding. External baselines also receive φ, for a fair comparison.

			Shared pose					No shared pose
	Inputs		DrivAerML			AhmedML		Aneumo
Method	p	φ	vol. press.	surf. press.	surf. WSS	surf. C_p	surf. C_f	vol. press.
Non-equivariant baselines								
GINO (Li et al., 2023b)	✓	✓	36.15	21.07	40.59	10.50	13.21	49.04
DeepONet (Lu et al., 2021)	✓	✓	41.66	38.06	53.51	18.04	15.66	50.64
AB-UPT (Alkin et al., 2025)	✓	✓	4.32	4.74	9.17	4.80	6.31	43.81
Equivariant baselines								
EqGINO (Kim et al., 2026)	×	✓	28.29	18.56	37.30	8.15	8.79	44.90
AB-GATr (Rygiel et al., 2026)	✓	✓	11.87	8.39	14.48	5.06	6.85	42.75
Backbone input variants (Transolver)								
Transolver(p)	✓	×	10.04	10.12	19.51	5.95	6.93	67.95
Transolver(φ)	×	✓	24.23	13.26	23.31	10.75	17.21	44.92
Transolver(p, φ)	✓	✓	4.07	4.53	10.06	5.14	6.04	46.34
GraFINS = Transolver(φ, g)	×	✓	**3.85**	**4.45**	**9.12**	4.73	**5.97**	**36.39**
Backbone input variants (Transolver-3)								
Transolver-3(p)	✓	×	14.78	10.97	22.71	5.28	6.69	66.82
Transolver-3(φ)	×	✓	25.77	13.80	24.54	11.55	17.41	45.31
Transolver-3(p, φ)	✓	✓	4.40	4.89	10.45	**4.55**	6.48	47.23
GraFINS = Transolver-3(φ, g)	×	✓	4.39	4.67	9.73	4.58	6.12	36.89

that we generated and solved with the same OpenFOAM setup (App. D.4.1). Target definitions, solvers, sizes and splits are in App. A.3, and cross-dataset transfer results are in App. E.

Comparisons. To attribute a result to the per-point input rather than to the backbone, we train four variants that differ only in that input. With f the backbone that GraFINS uses, p the coordinates of a point, φ its frame-invariant features and g its graph embedding (Section 3.2), the variants are **(i)** $f(p)$, **(ii)** $f(\varphi)$, **(iii)** $f(p, \varphi)$ and **(iv)** GraFINS $= f(\varphi, g)$. The gap between $f(p, \varphi)$ and $f(\varphi)$ measures what a coordinate input contributes, and the gap between $f(\varphi, g)$ and $f(\varphi)$ measures what the graph embedding provides in its place. We run the four variants on Transolver (Wu et al., 2024) and Transolver-3 (Zhou et al., 2026), and GraFINS uses Transolver unless stated otherwise. As external baselines we use the non-equivariant GINO (Li et al., 2023b), DeepONet (Lu et al., 2021) and AB-UPT (Alkin et al., 2025) and the equivariant EqGINO (Kim et al., 2026) and AB-GATr (Rygiel et al., 2026). Each keeps its own inputs, including coordinates, and also receives φ, so none lacks the features we add. They differ from the four variants in backbone, capacity and training recipe, so they compare architectures rather than inputs.

Metrics. We report the relative L_2 error (rel-L_2) in percent, computed per case in the original scale of each target and averaged over the test cases, and lower is better. The unseen-inflow and arbitrary-geometry studies report R^2, averaged in the same way, and higher is better (App. A.4).

4.1 In-distribution accuracy: Can a surrogate predict without coordinates?

This experiment asks whether removing coordinates costs accuracy where coordinates work well. On DrivAerML and AhmedML every case sits in one shared pose, so coordinates are a reliable input and the shortcut is at its strongest. Coordinates alone, Transolver(p), already reach a rel-L_2 of 10.04% on DrivAerML volume pressure. Table 2 answers with three comparisons. First, removing coordinates without a replacement degrades accuracy. Transolver(φ) is far worse than Transolver(p, φ) on every DrivAerML and AhmedML column, for example 24.23 against 4.07 on DrivAerML volume pressure. Second, the graph embedding restores the accuracy lost by removing coordinates. GraFINS is more accurate than Transolver(p, φ) on all six columns and is the most accurate model on five of them. The exception is AhmedML C_p, where GraFINS is close to Transolver-3(p, φ) at 4.73 against 4.55. Third, the same ordering holds with Transolver-3 as the backbone, with the same single exception, where the two are nearly equal at 4.58 against 4.55, so the result does not depend on one backbone.

Aneumo has no shared pose (Fig. 15), and there the coordinate input turns from help to harm. Adding coordinates to Transolver(φ) raises the error from 44.92 to 46.34, whereas the graph embedding lowers it to 36.39, the largest gain of GraFINS over Transolver(p, φ) among the six columns. The same holds with Transolver-3, at 45.31, 47.23 and 36.89. Each vessel lies in its own frame, so

Table 3: **Unseen inflow and arbitrary geometries** (per-case R^2, ↑). All models train on ShapeNet-Car at $0°$ only, and every column is scored on test shapes. (a) evaluates the 100 ShapeNet-Car test shapes at the trained $0°$ and at $30°$, $90°$ and $180°$. (b) applies the same checkpoints frozen to 50 geometries unrelated to any vehicle, generated as described in App. D.4.1.

	Inputs		(a) ShapeNet-Car geometries					(b) Arbitrary geometries			
			Trained	Inflow directions				Inflow directions			
Method	p	φ	$0°$	$30°$	$90°$	$180°$	*Mean*	$0°$	$30°$	$90°$	*Mean*
Non-equivariant baselines											
GINO (Li et al., 2023b)	✓	✓	0.460	−0.026	−0.291	0.021	*−0.099*	−0.696	−0.373	−0.436	*−0.502*
DeepONet (Lu et al., 2021)	✓	✓	0.482	0.102	0.003	−0.148	*−0.014*	−5.204	−2.492	−1.878	*−3.191*
AB-UPT (Alkin et al., 2025)	✓	✓	0.683	0.097	−0.456	−0.420	*−0.260*	−0.131	−0.271	−0.922	*−0.441*
Equivariant baselines											
EqGINO (Kim et al., 2026)	×	✓	0.538	0.185	0.044	0.373	*0.201*	−0.335	−0.247	−0.262	*−0.281*
AB-GATr (Rygiel et al., 2026)	✓	✓	0.641	0.211	0.382	0.422	*0.338*	−0.099	−0.110	−0.128	*−0.112*
Backbone input variants											
Transolver(p)	✓	×	0.634	0.047	−0.423	−0.115	*−0.164*	−0.114	−0.114	−0.332	*−0.187*
Transolver(φ)	×	✓	0.648	0.045	0.274	−0.012	*0.102*	0.061	−0.034	0.047	*0.024*
Transolver(p, φ)	✓	✓	**0.692**	0.062	−0.081	−0.432	*−0.151*	0.012	−0.151	−0.283	*−0.141*
GraFINS = Transolver(φ, g)	×	✓	0.670	**0.406**	**0.488**	**0.544**	***0.480***	**0.421**	**0.301**	**0.348**	***0.356***

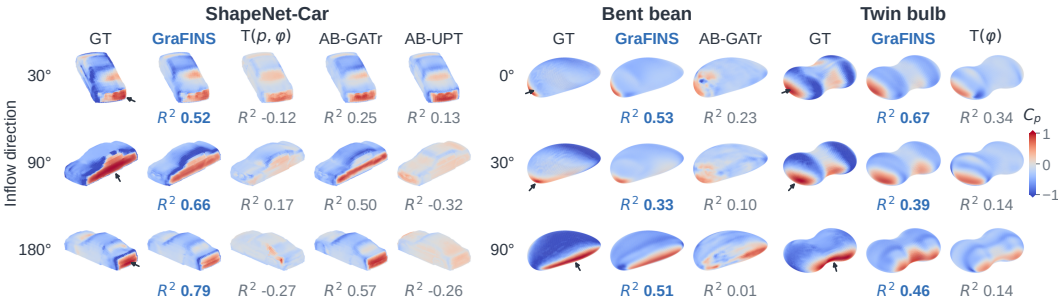


Figure 3: **Surface pressure under unseen inflow.** Rows are inflow directions, each marked by the arrow. T is Transolver. All nine models and directions are in Fig. 14.

Figure 4: **Surface pressure on two arbitrary geometries.** Rows are inflow directions, each marked by the arrow. T is Transolver. All nine models on all sixteen shape families are in App. F.3.

the same coordinates refer to different parts of the anatomy in different cases. The graph embedding supplies spatial information independent of the frame, so it lowers the error here as well.

Ablations. App. D.1 varies the number k of graph neighbours, the message passing depth, the number of scales, the pooling and the number K of test-time averages on AhmedML C_p, and removes each frame-invariant channel in turn.

4.2 Unseen inflow: Can a surrogate predict an unseen boundary condition?

This experiment asks whether a surrogate responds to a boundary condition it has not seen. The vehicle benchmarks fix the inflow direction, so we re-solved the ShapeNet-Car flows at each tested inflow direction. Each geometry is held fixed while the inflow direction changes, so each direction gives a different flow with its own solution. A model that relies on the dataset frame tends to keep the high-pressure region where it was during training, whereas a model that relates the geometry to the inflow moves it to the new windward face. All models train at $0°$ only and are tested on the 100 test shapes at $0°$, $30°$, $90°$ and $180°$ (App. D.3).

Table 3(a) shows that GraFINS has the highest R^2 in every unseen direction and the highest mean over them at 0.480. Accuracy at the trained direction does not predict this result. Transolver(p, φ) is the most accurate model at $0°$ with 0.692 but falls to -0.432 at $180°$, whereas the R^2 of GraFINS stays between 0.406 and 0.544, against 0.670 at $0°$. The input variants separate the two contributions. Adding coordinates to Transolver(φ) lowers the mean from 0.102 to -0.151, and adding the graph embedding raises it to 0.480. The sign of the mean splits the nine models. The five whose prediction is tied to the dataset frame are all negative, which is worse than predicting the mean field, and the four that are free of it are all positive, the two equivariant baselines, Transolver(φ) and GraFINS. Equivariance guarantees the correct response when the geometry and the inflow rotate together, but

a new inflow direction on a fixed geometry is a new boundary condition, and GraFINS leads the equivariant AB-GATr by 0.14. Figure 3 shows the response on one shape. GraFINS moves the high-pressure region to the newly windward face, including the rear at $180°$, while Transolver(p, φ) and AB-UPT keep it at the front bumper. A surrogate therefore responds to an unseen inflow direction only when nothing in its input is tied to the dataset frame.

4.3 Arbitrary geometries: Can a surrogate predict on an unrelated shape?

This experiment asks whether a surrogate responds to a geometry unlike any it was trained on. The governing equations do not refer to a shape family, so a model that has learned the flow response should transfer to such geometries, whereas one that has learned a shape prior should not. We generate fifty geometries without vehicle-like structure, grouped into sixteen families such as domes, cross blocks, prisms, cylinders and T-junctions, build each surface on the ShapeNet-Car template of $3{,}682$ points, and solve each with the OpenFOAM setup of our ShapeNet-Car runs (App. D.4.1). The checkpoints of Table 3(a) are applied frozen, and nothing is trained or tuned on the new shapes.

Table 3(b) separates two changes. Its first column keeps the trained inflow direction and changes only the geometry, and its other two columns change the geometry and the inflow together. With only the geometry new, GraFINS reaches 0.421, whereas no other model exceeds 0.061. With the inflow changed as well, GraFINS leads every column, and it and Transolver(φ) are the only models whose mean over the three columns stays positive. The input variants again separate the two contributions. Adding coordinates to Transolver(φ) lowers the mean from 0.024 to -0.141, and adding the graph embedding raises it to 0.356. Both equivariant baselines are negative in every column, so equivariance alone does not make a model transfer to an unseen shape.

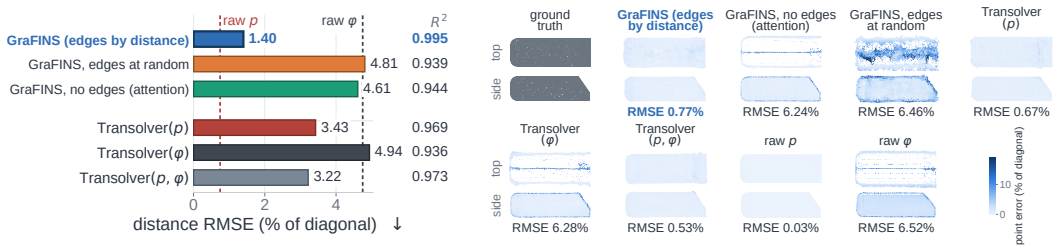
Figure 4 shows the same behaviour on two of these geometries. GraFINS places the high-pressure region on the surface facing the oncoming flow and moves it as the inflow turns, whereas the most accurate baseline on each geometry predicts it weakly or in the wrong place. On the *twin bulb* it also recovers the raised pressure along the narrow waist between the two lobes, which Transolver(φ) misses. The waist is shielded from the oncoming flow, and its pressure rises because the lobes turn the flow inward and slow it in the concave corner. Predicting this requires knowing that the point lies between two lobes, which its local features do not encode, whereas the graph embedding describes the surrounding geometry at the scale of the lobes. GraFINS is the only model with a positive R^2 in all sixteen families (App. D.4.2). A positive R^2 is far from an accurate prediction, but it shows that GraFINS responds to a geometry and an inflow it has never seen, which no baseline does.

4.4 Rotated frame: Can a surrogate predict without a shared frame?

This experiment asks whether a surrogate needs the training frame at all. We rotate the geometry and the inflow together by $45°$ and $90°$, angles never seen in training, recompute every geometry-derived input from the rotated scene and retrain nothing. The physical scene is unchanged and only its description in coordinates moves, so a correct surrogate returns the same field. GraFINS does at every angle (Table 4) and on all three datasets (App. D.5). Every input it receives is unchanged by the rotation. The models that receive coordinates do not. Transolver(p, φ) rises from 4.53 to 91.55 although it receives the same frame-invariant features as GraFINS, and GINO, DeepONet and AB-UPT rise above 80, where 100 is the error of predicting zero. The two equivariant baselines behave as their construction predicts. AB-GATr is stable at every angle but starts from 8.39 against our 4.45, and EqGINO returns to 21.43 at $90°$ from 18.56 unrotated while rising to 84.01 at $45°$, because its symmetry is a discrete subgroup. Transolver(φ) is stable as well but less accurate than GraFINS on every dataset. Removing coordinates gives the invariance, and the graph embedding restores the accuracy.

Table 4: **Rotated frame** on DrivAerML surface pressure (rel-L_2 %, ↓), full test split of 50 cases. The $0°$ column restates Table 2. AhmedML and Aneumo are in App. D.5.

	DrivAerML (surf. p)		
	Scene rotation		
Method	$0°$	$45°$	$90°$
Non-equivariant baselines			
GINO (Li et al., 2023b)	21.07	103.5	98.45
DeepONet (Lu et al., 2021)	38.06	130.6	132.7
AB-UPT (Alkin et al., 2025)	4.74	82.68	91.21
Equivariant baselines			
EqGINO (Kim et al., 2026)	18.56	84.01	21.43
AB-GATr (Rygiel et al., 2026)	8.39	8.42	8.40
Backbone input variants			
Transolver(p)	10.12	113.8	123.0
Transolver(φ)	13.26	13.26	13.26
Transolver(p, φ)	4.53	89.03	91.55
GraFINS = Transolver(φ, g)	**4.45**	**4.45**	**4.45**



(a) Distance probe **(b) Coordinate probe**

Figure 5: **Probing frozen features for spatial information** on AhmedML. **(a)** Distance between two points, with the same probe on the raw inputs p and φ dashed. **(b)** Coordinates of each point on one test case, in top and side views. Errors are RMSE in percent of the bounding-box diagonal (↓).

4.5 Spatial probes: Does the graph embedding encode spatial information?

This experiment asks whether the graph makes spatial information more accessible in the learned representation, which the results above are consistent with but do not measure. We freeze each trained surrogate and train a small MLP to predict the distance between two points from their representations at the input of the prediction head. Two variants of GraFINS replace the nearest-anchor graph. One keeps the anchors but links each to a fixed random set of them, and the other computes the embedding with induced set attention, in which learned tokens attend to all points, with no anchors or graph. Figure 5(a) gives a distance error of 1.40% for GraFINS, far below either variant and below the models that receive coordinates, supporting the role of the anchor graph in encoding spatial information. Figure 5(b) turns to a separate probe, which predicts each point's coordinates. GraFINS never receives a coordinate, yet it matches the models that do, while both variants stay at the level of the raw features. Every AhmedML case is in the same pose, so a point's place in the geometry already fixes its coordinate. The probe therefore supports recoverable spatial information, not retention of the dataset frame. App. D.6 repeats the distance probe on all four benchmarks, where GraFINS is the most accurate coordinate-free representation.

5 Conclusion

On aligned benchmarks much of the usefulness of coordinates comes from the shared pose rather than from the physics, so low in-distribution error does not show that a surrogate has learned how the field responds to the geometry and the boundary conditions. GraFINS removes coordinates from every learnable layer and replaces them with frame-invariant features, which encode local geometry and its relation to the boundary conditions, and a graph embedding, which supplies the spatial information that coordinates provided. The replacement costs no in-distribution accuracy. On all six prediction targets of DrivAerML, AhmedML and Aneumo, GraFINS is more accurate than the same backbone given coordinates. Beyond the training distribution the difference grows. The most accurate model at the trained inflow direction loses its accuracy when the inflow turns, whereas GraFINS predicts every unseen direction better than all baselines, is the only model with a positive R^2 on all sixteen families of arbitrary geometries, and remains stable when the scene is rotated.

Limitations. GraFINS responds to unseen inflow directions and unseen geometries, but its accuracy there remains below its accuracy at the trained direction. Training on several inflow directions is a natural way to close this gap. The ground truth for both studies comes from our own OpenFOAM runs with one solver setup, but every model is scored against the same fields, so the comparison between models does not rest on the absolute accuracy of that setup. One pass of GraFINS costs about 2.5 times that of the Transolver backbone, comparable to the external baselines (App. D.2).

Outlook. The construction relies on no fluid-specific component beyond the choice of feature channels, and we expect it to apply to other field-prediction problems on 3D geometries. We hope that results under unseen inflow, on arbitrary geometries and under rotation are reported alongside in-distribution error, since they are what separates evaluating a surrogate for a dataset from evaluating it for the physics.

Reproducibility Statement

Anonymized source code is available at `https://anonymous.4open.science/r/GraFINS_ICLR27`. It implements the frame-invariant features, the anchor graph, the four input variants on both backbones, training, evaluation under joint rotation and the frozen probes, and it includes sample cases from DrivAerML, AhmedML and the arbitrary geometries. Section 3 gives the architecture, App. C.1 the feature definitions and App. C.2 the implementation and hyperparameters. App. A.3 gives the benchmarks and our splits, including the base-anatomy-disjoint Aneumo protocol. App. D.3.1 and App. D.4.1 describe the OpenFOAM setup behind the ShapeNet-Car ground truth and the arbitrary geometries. ShapeNet-Car and Aneumo are not redistributed and are obtained from their original releases.

Ethics Statement

This work targets engineering and biomedical simulation surrogates. Hemodynamic predictions from our models are research artifacts and must not be used for clinical decision-making without regulatory validation. The vascular datasets are de-identified public releases, and we conducted no study with human subjects. DrivAerML and AhmedML are released under CC BY-SA and CMHA under CC BY, Aneumo is non-commercial and no-derivatives, and ShapeNet-Car geometries follow the ShapeNet terms of use for non-commercial research. We use all of them under these terms and redistribute none of the restricted data.

AI Use Statement

The authors led every part of this work, including the problem formulation, the method, the design and execution of the experiments, the interpretation of the results and the writing. Large language models assisted under the authors' direction with implementing and running experiment code, including data processing, the OpenFOAM case generation for the ShapeNet-Car and arbitrary-geometry ground truth, the probes and the figures, and with drafting and editing text. The authors reviewed all code, results and text and take full responsibility for the content.

References

Corey Adams, Rishikesh Ranade, Ram Cherukuri, and Sanjay Choudhry. GeoTransolver: Learning physics on irregular domains using multi-scale geometry aware physics attention transformer. *arXiv preprint arXiv:2512.20399*, 2025.

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Benedikt Alkin, Maurits Bleeker, Richard Kurle, Tobias Kronlachner, Reinhard Sonnleitner, Matthias Dorfer, and Johannes Brandstetter. AB-UPT: Scaling neural CFD surrogates for high-fidelity automotive aerodynamics simulations via anchored-branched universal physics transformers. *Transactions on Machine Learning Research*, 2025.

Neil Ashton, Danielle C Maddix, Samuel Gundry, and Parisa M Shabestari. AhmedML: High-fidelity computational fluid dynamics dataset for incompressible, low-speed bluff body aerodynamics. *arXiv preprint arXiv:2407.20801*, 2024a.

Neil Ashton, Charles Mockett, Marian Fuchs, Louis Fliessbach, Hendrik Hetmann, Thilo Knacke, Norbert Schonwald, Vangelis Skaperdas, Grigoris Fotiadis, Astrid Walle, Burkhard Hupertz, and Danielle Maddix. DrivAerML: High-fidelity computational fluid dynamics dataset for road-car external aerodynamics. *arXiv preprint arXiv:2408.11969*, 2024b.

Johann Brehmer, Pim de Haan, Sönke Behrends, and Taco S. Cohen. Geometric algebra transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations (ICLR)*, 2022.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. doi: 10.1038/s42256-020-00257-z.

Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow approximation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 481–490, 2016.

Hrvoje Jasak. *Error analysis and estimation for the finite volume method with applications to fluid flows*. PhD thesis, Imperial College London, 1996.

Sungwon Kim, Namkyeong Lee, Yunyoung Doh, Seungmin Shin, Guimok Cho, Seung-Won Jeon, Sangkook Kim, and Chanyoung Park. Thickness-aware E(3)-equivariant 3D mesh neural networks. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, 2025.

Sungwon Kim, Juho Song, Seungmin Shin, Guimok Cho, Sangkook Kim, and Chanyoung Park. EqGINO: Equivariant geometry-informed Fourier neural operators for 3D PDEs. *arXiv preprint arXiv:2606.03260*, 2026.

Xigui Li, Yuanye Zhou, Feiyang Xiao, Xin Guo, Chen Jiang, Tan Pan, Xingmeng Zhang, Cenyu Liu, Zeyun Miao, Jianchao Ge, Xiansheng Wang, Qimeng Wang, Yichi Zhang, Wenbo Zhang, Fengping Zhu, Limei Han, Yuan Qi, Chensen Lin, and Yuan Cheng. Aneumo: A large-scale multimodal aneurysm dataset with computational fluid dynamics simulations and deep learning benchmarks. *arXiv preprint arXiv:2505.14717*, 2025.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

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

Zongyi Li, Nikola B. Kovachki, Christopher B. Choy, Boyi Li, Jean Kossaifi, Shourya Prakash Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Animashree Anandkumar. Geometry-informed neural operator for large-scale 3D PDEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

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

Huakun Luo, Haixu Wu, Hang Zhou, Lanxiang Xing, Yichen Di, Jianmin Wang, and Mingsheng Long. Transolver++: An accurate neural solver for PDEs on million-scale geometries. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, 2025.

Fadl Moukalled, Luca Mangani, and Marwan Darwish. The finite volume method. In *The finite volume method in computational fluid dynamics: An advanced introduction with OpenFOAM® and Matlab*, pp. 103–135. Springer, 2015.

Jeong Joon Park, Peter R. Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 165–174, 2019.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Rishikesh Ranade, Mohammad Amin Nabian, Kaustubh Tangsali, Alexey Kamenev, Oliver Hennigh, Ram Cherukuri, and Sanjay Choudhry. DoMINO: A decomposable multi-scale iterative neural operator for modeling large scale engineering simulations. *arXiv preprint arXiv:2501.13350*, 2025.

Patryk Rygiel, Julian Suk, Kak Khee Yeung, Christoph Brune, and Jelmer M Wolterink. Symmetry in the wild: The role of equivariance in neural fluid surrogates. *arXiv preprint arXiv:2605.18816*, 2026.

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

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 991–1001, 2017.

Sangwoo Seo, Hyunsung Kim, Jiwan Kim, and Chanyoung Park. Adaptive graph rewiring to mitigate over-squashing in mesh-based GNNs for fluid dynamics simulations. *arXiv preprint arXiv:2511.12709*, 2025.

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. DiffusionNet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics*, 41(3):27:1–27:16, 2022. doi: 10.1145/3507905.

Miao Song, Simin Wang, Qian Qian, Yuan Zhou, Yi Luo, and Xijun Gong. Intracranial aneurysm CTA images and 3D models dataset with clinical morphological and hemodynamic data. *Scientific Data*, 11(1):1213, 2024. doi: 10.1038/s41597-024-04056-8.

Jian Sun, Maks Ovsjanikov, and Leonidas J. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009. doi: 10.1111/j.1467-8659.2009.01515.x.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Nobuyuki Umetani and Bernd Bickel. Learning three-dimensional flow for interactive aerodynamic design. *ACM Transactions on Graphics*, 37(4):89, 2018. doi: 10.1145/3197517.3201325.

Shizheng Wen, Arsh Kumbhat, Levi E. Lingsch, Sepehr Mousavi, Yizhou Zhao, Praveen Chandrashekar, and Siddhartha Mishra. 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.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53681–53705, 2024.

Hang Zhou, Haixu Wu, Haonan Shangguan, Yuezhou Ma, Huikun Weng, Jianmin Wang, and Mingsheng Long. Transolver-3: Scaling up transformer solvers to industrial-scale geometries. *arXiv preprint arXiv:2602.04940*, 2026.

COORDINATES ARE LABELS, NOT PHYSICS: BREAKING THE COORDINATE SHORTCUT IN NEURAL SURROGATES

APPENDIX CONTENTS

A General information **14**

A.1 Related work . . . 14

A.2 Notation for the method . . . 15

A.3 Benchmark specifications and splits . . . 15

A.4 Error metrics . . . 16

B Positional predictability probe **17**

C Method and implementation **17**

C.1 Frame-invariant feature details per domain . . . 17

C.2 Implementation and training details . . . 18

D Additional experiments **21**

D.1 Ablation study . . . 21

D.2 Computational cost . . . 22

D.3 Unseen inflow . . . 23

D.4 Arbitrary geometries . . . 26

D.5 Rotated-frame evaluation across benchmarks . . . 28

D.6 Spatial probes on all benchmarks . . . 29

E Cross-dataset transfer **32**

E.1 Transfer protocol . . . 32

E.2 External aerodynamics: DrivAerML, ShapeNet-Car and AhmedML . . . 34

E.3 Vascular flow: Aneumo to CMHA . . . 35

E.4 Summary of the seven settings . . . 36

F Qualitative results **37**

F.1 Predictions under unseen inflow . . . 37

F.2 Predictions on Aneumo . . . 38

F.3 Predictions on arbitrary geometries . . . 40

A General information

A.1 Related work

Neural surrogates for PDEs on 3D geometry. Existing surrogates process a 3D geometry in one of four ways. (i) *Grid-based* operators compute spectral convolutions on a regular grid, which gives them a global receptive field through Fourier modes (Li et al., 2021) but no direct way to take an irregular geometry. Geo-FNO (Li et al., 2023a) learns a deformation of the domain onto a uniform latent grid, and GINO (Li et al., 2023b) encodes the point cloud onto a latent grid, applies the spectral operator there, and decodes back to the points. (ii) *Point-based* models operate on the sampled points themselves. Branch-trunk operators feed the query coordinate to a network conditioned on an encoding of the geometry (Lu et al., 2021), and local-neighbourhood operators aggregate features in a ball around each point, so the per-point cost is independent of mesh size (Ranade et al., 2025). (iii) *Mesh-based* graph networks pass messages along the edges of the simulation mesh, whose graded cells represent the local geometry faithfully (Pfaff et al., 2021). Each round crosses one edge, so the receptive field is tied to the mesh resolution, and later work widens it by adaptively rewiring the graph (Seo et al., 2025) or adding edges between opposite surfaces (Kim et al., 2025). (iv) *Token-based* transformers compress the points into far fewer tokens so that global attention becomes affordable. Transolver learns slices that group points sharing a physical state (Wu et al., 2024), and later work scales them to larger meshes (Luo et al., 2025) and to over a hundred million cells by tiling the computation of physical states (Zhou et al., 2026), or adds multi-scale geometry cross-attention (Adams et al., 2025) and attentional graph encoders (Wen et al., 2025). UPT compresses the input into a resolution-free latent space (Alkin et al., 2024), and AB-UPT reaches industrial mesh sizes through anchored, branched attention (Alkin et al., 2025).

Across these representations, position typically enters as coordinates in the frame in which the dataset was generated, with normals, distance fields, and design parameters as optional extras (e.g., Alkin et al., 2025). On aligned benchmarks, low error alone does not reveal how much prediction depends on the shared pose. The sensitivity of coordinate-based surrogates to rotation has motivated equivariant designs (Kim et al., 2026; 2025; Rygiel et al., 2026), and reliance on dataset-specific cues is a known failure mode termed shortcut learning (Geirhos et al., 2020).

Geometry representations. Signed-distance fields have long served as a network's shape input. They were the geometry input of the early CNN flow surrogates (Guo et al., 2016), became the learned representation itself in DeepSDF (Park et al., 2019), and are now a common *auxiliary* channel alongside coordinates (e.g., Li et al., 2023b; Ranade et al., 2025). Invariant descriptors in place of coordinates are likewise common outside PDE surrogates, where each surface point is characterized by signatures computed from the surface alone, such as the heat kernel signature (Sun et al., 2009), which DiffusionNet (Sharp et al., 2022) uses to learn on surfaces independently of pose and discretization. Because such signatures are computed from the surface alone, however, they carry nothing about the inflow direction or the ground, the relations the physics depends on. The frame-invariant features of GraFINS add exactly these relations to the boundary conditions.

Invariance and equivariance in neural surrogates. In the molecular and point-cloud literature, where no canonical orientation exists, symmetry is obtained in two ways. SchNet (Schütt et al., 2017) feeds the network invariant scalars (interatomic distances), while tensor field networks (Thomas et al., 2018) and EGNN (Satorras et al., 2021) constrain the architecture to be equivariant. The design question is the same for PDE surrogates on industrial geometry. T-EMNN (Kim et al., 2025) predicts node-level fields on industrial meshes from data-driven coordinates made E(3)-invariant by aligning each shape to its own principal axes. GATr (Brehmer et al., 2023) builds equivariant attention in geometric algebra, and AB-GATr (Rygiel et al., 2026) scales it to fluid meshes. AB-GATr reports lower in-distribution accuracy than its non-equivariant counterpart on aligned automotive data but higher accuracy on hemodynamics, which it attributes to the alignment of the automotive data. EqGINO (Kim et al., 2026) guarantees equivariance to discrete rotation subgroups within a neural-operator backbone. Invariant inputs and equivariant architectures are well-studied approaches, and neither alone guarantees accuracy under new boundary conditions or on new shape families. Our study examines whether replacing coordinate inputs can retain in-distribution accuracy while improving prediction under unseen inflow and on arbitrary geometries.

A.2 NOTATION FOR THE METHOD

Table 5: **Main-text notation for GraFINS.**

Symbol	Meaning
Query points and physical inputs	
p_i	Position of query point i. In input-variant labels, p denotes the coordinates.
φ_i	Frame-invariant local feature vector, with channels indexed by c.
$A_c, B_c;\ \psi_c$	Local geometric quantity and boundary reference of channel c, and the map that turns them into a scalar.
b	Scalar conditions supplied through the backbone's conditioning input.
d	Distance to the body surface, the no-slip wall.
Anchor graph and learned representations	
$L;\ \ell$	Number of anchor scales and the scale index, with $\ell = 1$ the finest and $\ell = L$ the coarsest.
$C;\ C_\varphi$	Width of the network and the number of frame-invariant channels.
$N;\ S_\ell;\ a_j^{(\ell)}$	Number of query points, number of anchors at scale ℓ, and the position of anchor j.
$\mathcal{V}_j^{(\ell)}$	Voronoi cell of anchor j, the set of query points assigned to it.
$h_j^{(\ell)};\ \tilde{h}_j^{(\ell)}$	Anchor state after pooling and after message passing.
k	Number of nearest anchors joined to an anchor by an edge.
$g_i^{(\ell)};\ g_i$	Graph embedding interpolated from scale ℓ to point i, and its concatenation over scales.
$x_i, \tilde{x}_i$	Encoded local feature, not a coordinate, and its graph-enriched version entering the backbone.
E, W_g	Learned linear maps for point encoding and projection of concatenated graph embeddings.
α	Learned scalar gate on the graph residual added before the backbone.
$\mathrm{LN}, [\cdot;\cdot]$	Layer normalization and feature concatenation.
$f;\ z_i$	Backbone and its state at point i.
$\mathrm{Head};\ \hat{y}_i$	Pointwise prediction head and the field value it predicts.
K	Number of predictions from resampled anchor sets averaged at inference.

Table 5 summarizes the main-text method notation. Point indices are written as i, anchor indices as j, scale indices as ℓ and channel indices as c. An omitted point index denotes the array over all query points. Appendix-only quantities are defined where they appear.

A.3 BENCHMARK SPECIFICATIONS AND SPLITS

Table 6: **Benchmark specifications.** Mesh sizes are median numbers of cells per case. A dash marks a mesh the release does not distribute. Split counts the cases we use, and for Aneumo a case is one geometry at one mass-flow rate.

		Mesh size			
Dataset	Cases	surface	volume	Simulation	Split
DrivAerML	500	8.43M	158M	OpenFOAM, DDES	400/34/50
AhmedML	500	1.07M	21.2M	OpenFOAM, hybrid RANS-LES	400/50/50
ShapeNet-Car	889	31.5k	424k	OpenFOAM, RANS	711/78/100
Aneumo	85,280	4.85k	61.0k	OpenFOAM, steady	1440/120/540
CMHA	149	7.24k	–	Fluent, steady	16/8/20
Arbitrary geoms. (ours)	50	30.7k	405k	OpenFOAM, RANS	eval only

Table 6 lists the six benchmarks used in this paper. Table 1 in Section 2 gives the frame conventions of four public benchmarks. Every model reported in this paper, ours and the baselines alike, is trained and evaluated on the same split of each benchmark.

DrivAerML (Ashton et al., 2024b)**.** Five hundred variants of the DrivAer notchback, morphed through sixteen parameters, all at one condition, $U_\infty = 38.889$ m/s, giving a Reynolds number of 7.19×10^6 on the 2.786 m wheelbase. Each case is a scale-resolving simulation at industrial resolution. Sixteen of the five hundred cases ship without simulation results, leaving 484, which we split $400/34/50$, the split sizes of Transolver-3 (Zhou et al., 2026). Surface and volume tasks use the same split. Its targets use the scaling constants of the public `physicsnemo-curator` pipeline rather than the case's own free stream, so they are not pressure coefficients.

AhmedML (Ashton et al., 2024a)**.** Five hundred variants of the Ahmed body over six shape parameters. The release holds everything but the geometry fixed, at $U_\infty = 1$ m/s and unit reference density, giving a Reynolds number of 7.68×10^5 on the 0.288 m body height. The rear slant drives pressure-induced separation and three-dimensional vortices (Ashton et al., 2024a). Nothing but the shape changes between cases, so the benchmark isolates what a model learns from shape while the frame is fixed. We split $400/50/50$ at random. We take C_p from the release's pressure-coefficient array and obtain C_f by doubling the released wall shear stress, since the free stream is unity.

ShapeNet-Car (Umetani & Bickel, 2018)**.** Cars from the ShapeNet category, edited to remove side mirrors, spoilers and tyres, so the shapes are independent vehicle designs rather than variants of one body. Its cases share one pose, as in DrivAerML and AhmedML. The release provides fields at one inflow direction only, so we re-solve every geometry in OpenFOAM, the training and validation shapes at the trained 0° and the test shapes at every evaluated direction, and map the surface pressure onto the $3{,}682$ points of the release's shared surface mesh (App. D.3.1). All ShapeNet-Car fields in this paper come from these runs. We hold out one of the nine folds as the test set, the same 100 cases as in earlier work on this benchmark, and split the rest $711/78$.

Aneumo (Li et al., 2025)**.** The release deforms the 427 real aneurysm geometries of the AneuX database into $10{,}660$ shapes and solves each under eight steady mass-flow conditions, giving $85{,}280$ cases. Each vessel is given in its own scanner frame, so no pose is shared, and the sac grows at a different place in every patient, so the shape variety is wide. The target is the pressure in the vessel volume in the release's dimensionless form $p^* = p/(\frac{1}{2}\rho V_{\max}^2)$, with $V_{\max}$ the peak velocity magnitude of the case, which removes the difference in pressure scale between mass-flow conditions. Many shapes are deformations of the same anatomy, so a split over shapes would put near-copies of a test shape into training. Ours is disjoint in base anatomy instead. We take 195 of the 427 anatomies with two deformations each and assign every anatomy to a single split. Training uses 240 geometries and the test set 90 geometries on unseen anatomies, both at six of the eight mass-flow rates.

CMHA (Song et al., 2024)**.** Clinical data rather than a simulation benchmark. The release reconstructs patient models from CTA, 105 middle cerebral artery aneurysms in 99 patients and 44 aneurysm-free controls, and reports per-case scalar summaries but no point-wise field. We use the 44 controls, split $16/8/20$. The transfer study of App. E.3 therefore predicts a case-level target, a dimensionless pressure formed from the release's mean internal pressure and peak velocity. The vessels are real patients from another institution rather than deformations of a shared set, so transfer from Aneumo changes the geometry distribution and the provenance of the data at once.

A.4 Error metrics

Both metrics are computed on the sampled points of one case and then averaged over the cases of the test split. Writing y for the reference field on those points, $\hat{y}$ for the prediction and $\bar{y}$ for the mean of y over the case,

$$\varepsilon = 100\,\frac{\|\hat{y} - y\|_2}{\|y\|_2}, \qquad R^2 = 1 - \frac{\|\hat{y} - y\|_2^2}{\|y - \bar{y}\|_2^2}. \tag{3}$$

The tables report ε as rel-L_2 in percent. It divides by the norm of the reference field, so a constant rescaling of a target leaves it unchanged. For the vector targets, wall shear stress and C_f, the three components are averaged. The reference of R^2 is the mean of the case's own field, so a negative value means the prediction is worse than predicting that mean field. Means over inflow directions weight each direction equally.

B POSITIONAL PREDICTABILITY PROBE

The probe uses the same number of cases and points on every benchmark, so the three values of R^2 are computed at equal sample size. The cases are 48 of the 50 test cases of DrivAerML and AhmedML and 48 Aneumo geometries drawn at random from the 10,660 of the release at a mass-flow rate of 0.003 kg/s. Each case contributes 3,682 points drawn uniformly from its surface cell centres, or from its interior points on Aneumo. Neighbours are searched in the dataset's own coordinates, and the left-out case supplies only query positions.

C METHOD AND IMPLEMENTATION

C.1 FRAME-INVARIANT FEATURE DETAILS PER DOMAIN

Table 7: **Frame-invariant channels per domain.** d is the distance to the wall and Δd its Laplacian, $\hat{u}$ the inflow direction, $\hat{g}$ the ground normal and $\hat{n}_{\mathrm{in}}$ the inlet-plane normal. d_{in} and d_{out} are the distances to the inlet centre and to the nearest outlet point, p_0 is the extreme sampled point along B_c, $z = (p - p_0) \cdot \hat{g}$ and r is the maximum vessel radius. A_c and B_c give the pair of Equation 1 behind each channel. A dash marks a channel built from the geometry alone (B_c) or one that the domain does not define.

	A_c	B_c	Surface (DrivAerML, AhmedML, ShapeNet-Car)	Volume (DrivAerML)	Internal (Aneumo)
proximity	d	–	–	$\lvert d\rvert / d_{99}$	$\tanh(d/0.5)$
streamwise position	$p - p_0$	$\hat{u}$	$(p - p_0) \cdot \hat{u}/\Lambda$	–	$d_{\mathrm{in}}/(d_{\mathrm{in}} + d_{\mathrm{out}})$
curvature	Δd	–	$\operatorname{arcsinh}(\Lambda\, \Delta d)$	$\operatorname{arcsinh}(d_{99}\, \Delta d)$	$\operatorname{arcsinh}(\Delta d)$
flow alignment	$\widehat{\nabla d}$	$\hat{u}$	$\widehat{\nabla d} \cdot \hat{u}$	$\widehat{\nabla d} \cdot \hat{u}$	$\widehat{\nabla d} \cdot \hat{n}_{\mathrm{in}}$
confinement	$p - p_0$	$\hat{g}$	$z/(z + 0.1\Lambda)$	$z/(z + 0.1\, d_{99})$	–
handedness χ	$\widehat{\nabla d}$	$\hat{u} \times \hat{g}$	$\widehat{\nabla d} \cdot (\hat{u} \times \hat{g})$	$\widehat{\nabla d} \cdot (\hat{u} \times \hat{g})$	–
distance field			signed, BVH	signed, BVH	unsigned, BVH
stencil axes			inflow–ground	inflow–ground	inlet–mesh
step h			0.01	0.05	$0.02\, r$
scalar BCs (separate input)			3 global parameters	3 global parameters	mass-flow rate

All channels are computed from a distance field $d(\cdot)$ of the geometry, queried per point by closest-point search on a bounding volume hierarchy (BVH) built with Open3D, and from its gradient ∇d and Laplacian Δd, both taken by central finite differences of step h over a six-point stencil without smoothing. The stencil lies along three orthonormal axes. The exact gradient and Laplacian do not depend on the axes, and the finite-difference values depend on them only through the truncation error. We build the axes from the boundary references and the geometry, so they turn with the scene, and express the mesh and the query points in them before the BVH is built. The distance values and finite differences of a rotated case then equal those of the original case to floating-point precision. The inflow–ground frame of external flow consists of $\hat{u}$, the component of $\hat{g}$ normal to $\hat{u}$ and their cross product. The inlet–mesh frame of internal flow consists of the inlet-plane normal $\hat{n}_{\mathrm{in}}$, the unit vector normal to $\hat{n}_{\mathrm{in}}$ that points toward the mesh vertex farthest from the line through the vertex centroid along $\hat{n}_{\mathrm{in}}$, and their cross product. Any other orthonormal axes that turn with the scene give the same invariance. The gradient enters only through dot products, and the vector itself is never an input. Length scales are per case. Λ is the extent of the point cloud along $\hat{u}$, and d_{99} the 99th percentile of d over the sampled points. Table 7 lists the instantiation used by each benchmark, and Equation 4 is the surface instantiation used by DrivAerML, AhmedML and ShapeNet-Car.

$$\varphi_{\mathrm{surf}}(p) \;=\; \Big[\; \frac{(p - p_0) \cdot \hat{u}}{\Lambda},\;\; \operatorname{arcsinh}\big(\Lambda\, \Delta d\big),\;\; \cos\angle\big(\nabla d,\, \hat{u}\big),\;\; \frac{z}{z + 0.1\Lambda},\;\; \widehat{\nabla d} \cdot (\hat{u} \times \hat{g}) \;\Big], \quad (4)$$

where p_0 is the most upstream sampled point, so the first channel is a streamwise position in $[0, 1]$, and z is the height above the lowest sampled point along $\hat{g}$.

The six channels come from the pairs of Equation 1. Projecting the unit gradient on the imposed directions gives *flow alignment* and *handedness*, and projecting the position gives *streamwise position*

and *confinement*. Two channels, *proximity* and *curvature*, use d and its Laplacian alone and refer to no condition. A surface has no proximity channel because the wall distance is identically zero there, and internal flow has no handedness channel because no second reference direction is defined, so the surface and volume domains carry five channels and internal flow four.

External aerodynamics. The distance field is signed (positive outside the body) and is computed exactly from BVH closest-point queries on the surface mesh. Both derivatives are taken on this signed field, with $h = 0.01$ on the surfaces and $h = 0.05$ in the DrivAerML volume, in the mesh units of each dataset (metres for DrivAerML and AhmedML). The volume proximity channel uses $|d|$. The confinement channel uses the height above the lowest point of the sampled cloud rather than the true ground gap, which coincides with it when the geometry rests on the ground plane. The handedness channel χ is included on all three vehicle benchmarks. It is odd under reflection, so a mirrored car receives mirrored χ and the model can tell left from right without a frame.

Hemodynamics (Aneumo). Vessel surfaces are open meshes (inlet and outlet caps are missing), so a ray-parity sign is orientation-dependent. We therefore use the unsigned wall distance. Its finite-difference gradient is ill-conditioned at the wall, so $\widehat{\nabla d}$ is the unit vector from the closest wall point to p, which is the gradient of the unsigned distance away from the medial axis. The Laplacian is the finite difference above, clipped to $[-50, 50]$ before the arcsinh. The alignment channel uses the inlet-plane normal, obtained as the smallest principal axis of the inlet point set and oriented into the lumen. The streamwise channel $d_{\mathrm{in}}/(d_{\mathrm{in}} + d_{\mathrm{out}})$ uses Euclidean distances to the inlet centre and to the nearest outlet point. It is independent of the number of outlets and encodes the zero-pressure outlet as the reference of the pressure field. Distances are computed after centring the geometry at the centroid of the point cloud and dividing it by the maximum radius r about that centroid, so that d and Δd are of order one across vessels of different size.

Boundary conditions. Scalar boundary conditions do not enter the frame-invariant channels. They are broadcast per point and passed to the backbone's conditioning input. On DrivAerML and AhmedML surfaces they are the three global parameters normalized by their reference values (the volume runs pass the same three parameters), on ShapeNet-Car the min–max normalized inflow speed, on Aneumo the mass-flow rate in scaled units.

Invariance under rotation. Every channel is a dot product or a distance between two quantities that rotate with the scene. The unit gradient $\widehat{\nabla d}$, the offset $p - p_0$ and the boundary references $\hat{u}$, $\hat{g}$ and $\hat{n}_{\mathrm{in}}$ all turn together when the geometry and the boundary conditions are rotated, so their dot products do not change. On the vehicle benchmarks the rotations of Section 4.4 turn the geometry and the inflow about $\hat{g}$, which leaves the ground in place. Tilting the vehicle relative to a fixed ground, like turning the inflow on a fixed geometry, changes the boundary conditions and with them the channels. Vector targets such as wall shear stress are predicted as components along the streamwise, lateral and vertical axes of the inflow–ground frame and mapped back to Cartesian components, so the predicted vector turns with the scene.

C.2 Implementation and training details

Point sampling. Each training and evaluation pass samples a fixed number of points per case, $200{,}000$ surface cells (area-weighted) on DrivAerML and AhmedML and $32{,}000$ volume points on Aneumo. GINO and EqGINO on the DrivAerML volume, GINO on the DrivAerML surface and AB-GATr on DrivAerML and AhmedML use $50{,}000$ cells per case to fit within the available GPU capacity. On the DrivAerML surface, $200{,}000$ cells would limit the latent grid of GINO to 20^3 nodes on a 40 GB GPU, whereas $50{,}000$ cells allow 32^3. The experiments use NVIDIA L40S (48 GB), A100 (40 GB) and H100 (80 GB) GPUs.

External baseline inputs. Every external baseline receives φ in addition to its own inputs. DeepONet receives p and φ together in its trunk network. AB-UPT receives p as the coordinates of its anchors and query points and φ as their features. GINO receives φ as its point features and p as the geometry of its kernel integrals and of its latent grid, and its kernels act on the coordinate difference between a point and a grid node. EqGINO receives φ as its point features and p as the geometry of its two kernel integrals and of its latent grid. The kernel inputs are the distance between a point and a grid node and the distance of the grid node from the body centre, and the grid is fixed to the body. AB-GATr receives φ as scalar channels and p as geometric-algebra vectors on which its attention acts equivariantly. In both equivariant models the inflow direction enters only through φ.

Its speed enters as a global scalar, and the models see no direction vector. AB-UPT and AB-GATr sample their anchor and geometry points at random, and AB-GATr starts the farthest-point sampling of its pooling centres at a random point. Like GraFINS, both are evaluated in every table by averaging $K = 8$ predictions from independently sampled anchor sets, and the other baselines are deterministic.

Coordinates inside the graph constructor. Coordinates are used in four non-learned computations, the nearest-anchor assignment, the k-nearest-neighbour edges among anchors, the inverse-distance weights of the broadcast and the edge scalar. No learnable layer receives coordinates, and the only coordinate-derived quantity that reaches one is the edge scalar, a ratio of distances that a rotation leaves unchanged. All four measure Euclidean distance in per-case normalized coordinates

$$p' = (p - c) \,/\, r, \tag{5}$$

where c is the centroid of the point cloud and r its maximum radius about c. This normalization is isotropic, so a rotation of the scene is an isometry of p'. Pairwise distances are preserved, and the assignment, the edges and the weights built for a rotated case are those of the original case up to the randomness of anchor sampling and to ties in neighbour selection. The anchor ranking itself uses a frame-invariant channel, so the sampling is pose-independent by construction. The graph of every benchmark uses Equation 5. Under a joint rotation of the geometry and the boundary conditions, then, three things hold. Every channel of φ agrees with its unrotated value to floating-point precision, which rests on each channel being a product of co-rotating quantities and on a finite-difference stencil that turns with the scene. For the same anchor seed the assignment π_ℓ, the edge lists and the broadcast weights are identical, which rests on the isotropic normalization and on coordinates reaching no learnable layer. With the same evaluation seeds at every angle the prediction is identical, which is the quantity Table 4 reports.

Anchors. Anchor counts per scale are $30{,}000 \to 4{,}000 \to 500 \to 80$ on the DrivAerML surface and volume and on the AhmedML surface, $1{,}024 \to 256 \to 64 \to 16$ on ShapeNet-Car, and $6{,}000 \to 1{,}000 \to 200 \to 50$ on Aneumo (all four-scale). On the AhmedML and DrivAerML surfaces, the mean distance between neighbouring anchors, measured with the trained sampler on five test cases, is $0.32/0.86/2.39/6.09\%$ and $0.31/0.80/2.10/4.68\%$ of the bounding-box diagonal from the finest to the coarsest scale, an increase by a factor of 2.2 to 2.8 per scale. At every scale the points are ranked by their first frame-invariant channel, the streamwise position on a surface and the wall distance in a volume, split into eight equal-count bands, and $\lfloor S_\ell/8 \rfloor$ anchors are drawn uniformly from each band. Because the bands hold equal numbers of points, the anchors follow the point density as uniform sampling would, and the stratification guarantees every band its share at every scale. The anchors of each scale are sampled independently from the full cloud. They are sampled at random during training. A single evaluation pass uses a fixed seed per scale, the same for every case, and test-time averaging instead runs K passes with different seeds and averages their predictions. Each point is assigned to its nearest anchor (a Voronoi cell), and the anchor state pools the encoded point features in its cell. The pooling operator is set per benchmark among the channel-wise maximum, the mean and their combination, the concatenated maximum and mean followed by a linear layer. The models of Table 2 use the combination on the AhmedML and DrivAerML surfaces and the maximum on the DrivAerML volume and Aneumo. App. D.1 compares the three operators on AhmedML.

Edges and message passing. Each anchor keeps directed edges from its $k = 13$ nearest anchors, excluding itself, and each GATv2 layer adds one self-loop per anchor whose edge scalar is the mean of the incoming ones. An edge carries one scalar, its length divided by the mean edge length of that scale. Each scale has its own three-layer GATv2 network (Brody et al., 2022) of width 256, and the scales do not exchange messages. Each layer is a GATv2 convolution with four heads of width 64 whose outputs are concatenated, with the edge scalar given to the attention as an edge feature. The first two layers are followed by layer normalization and GELU, the last by neither, and there are no residual connections and no dropout. Anchor states are broadcast back to every point by inverse-distance weighting, $w = 1/\|p'_i - a'_j\|$, over the point's 8 nearest anchors of that scale.

Fusion and prediction head. Equation 2 and the head expression beside it in Section 3.3 give the early residual and the late concatenation. The encoder E maps the local features to width $C = 256$, and W_g projects the concatenated graph embeddings back to that width. The scalar gate α is initialized at 0.1. The backbone receives $\tilde{x}$ and the scalar conditions b, and its output layer emits

a C-dimensional state z_i at each point rather than the target. The prediction head is LayerNorm–Linear–GELU–Linear, with one output per target component.

Backbones. Transolver (PhysicsNeMo implementation) with 8 layers, width 256, 8 heads and 32 slices, and Transolver-3 with the same depth and width and its default MLP ratio of 4, wrapped so that its position input receives $\tilde{x}$ rather than coordinates. The graph embedding, fusion, and prediction head are identical across the two. In both cases the per-point input of the backbone is the concatenation of $\tilde{x}_i$ and the broadcast scalar conditions b, passed through the backbone's own input layer. Its slice attention therefore operates on features that carry spatial information through g_i and no coordinate. The backbone is used with dropout 0 and its output layer is widened to C so that it returns z_i.

Optimization. Surface pressure models on DrivAerML and AhmedML use AdamW, learning rate 10^{-4}, weight decay 0.05, 500 epochs with linear warm-up over the first 25 and cosine decay over the rest, gradient clipping at 1, bfloat16 autocast, batch size 1 case, relative-L_2 loss. Aneumo uses AdamW, learning rate 5×10^{-4}, weight decay 10^{-5}, 400 epochs with linear warm-up over the first 20 and cosine decay over the rest. The DrivAerML volume models use AdamW, peak learning rate 10^{-4}, weight decay 0.05, and a one-cycle schedule that rises from one fifth of the peak over the first 30% of the steps and then decays by cosine annealing, with $200{,}000$ points per case, gradient clipping at 1 and bfloat16 autocast. An exponential moving average of the weights with decay 0.999 is kept during training and used for validation and testing. All four input variants of each backbone on the volume use this schedule and averaging. Checkpoints are selected on the validation split.

D ADDITIONAL EXPERIMENTS

D.1 ABLATION STUDY

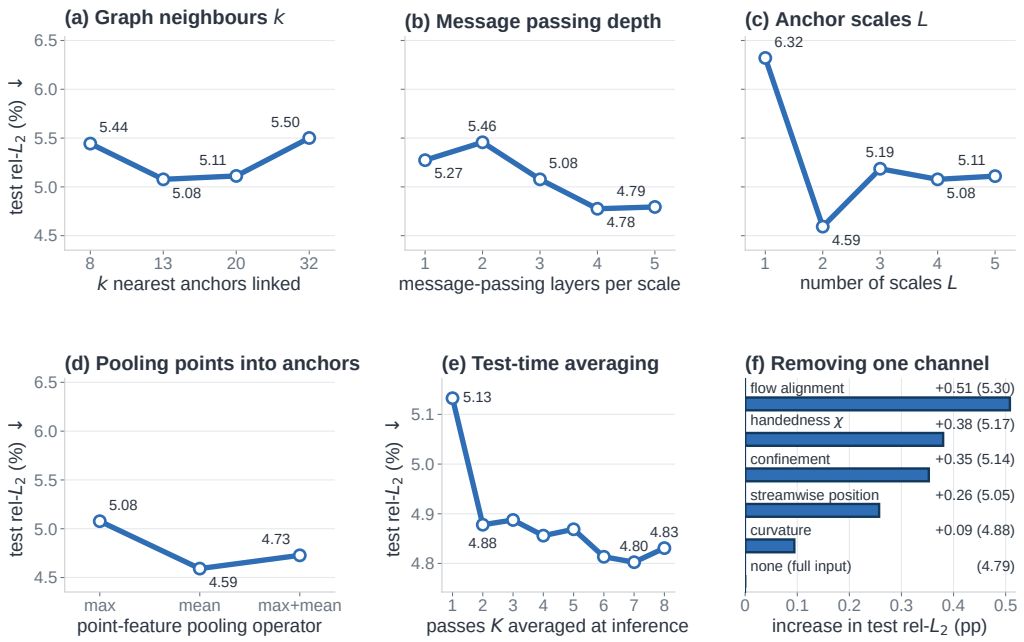


Figure 6: **Architecture and test-time sensitivity on AhmedML surface pressure** (rel-L_2 %, $K = 8$, 50 test cases). (a) Number of nearest anchors k linked by graph edges, (b) message-passing layers per scale, (c) number of anchor scales L, (d) pooling operator, (e) number of anchor-resampling passes K averaged at inference, and (f) removing one frame-invariant channel, with the increase over the full-input model and the absolute error in parentheses. Panels (a)–(c) vary the max-pooling configuration, and the max+mean point in (d) is the model of Table 2.

Table 8: **Test-time averaging of the stochastic baselines** (rel-L_2 %, ↓). $K = 8$ is the setting of Table 2.

		DrivAerML			AhmedML		Aneumo
Method	K	vol. press.	surf. press.	surf. WSS	surf. C_p	surf. C_f	vol. press.
AB-UPT (Alkin et al., 2025)	1	4.45	4.96	9.40	5.00	6.39	43.85
	8	4.32	4.74	9.17	4.80	6.31	43.81
AB-GATr (Rygiel et al., 2026)	1	13.07	8.90	15.66	5.43	7.24	43.65
	8	11.87	8.39	14.48	5.06	6.85	42.75

Protocol and reference model. We study architecture sensitivity on AhmedML surface pressure (C_p), reporting mean per-case relative L_2 error on 50 test cases. The graph-edge, depth, and anchor-scale sweeps use a five-channel frame-invariant input model with max pooling, $k = 13$, three message-passing layers, and four anchor scales with $(30{,}000, 4{,}000, 500, 80)$ anchors as their shared configuration. The pooling panel also includes the max+mean model of Table 2. The feature ablation in panel (f) uses the configuration of Table 2 rather than the max-pooling one. Each of the five frame-invariant channels is clamped to zero in training, validation and test, and the model is retrained from scratch for 500 epochs under that recipe. Width, parameter count, graph construction and the scalar boundary-condition input are unchanged, so a removal also propagates to the anchor sampling, which depends on φ.

Graph edges and depth. Figure 6(a) shows errors of $5.44/5.08/5.11/5.50\%$ for $k = 8/13/20/32$ nearest anchors, respectively. Linking more than 13 neighbours gives no consistent improvement. In panel (b), $1/2/3/4/5$ message-passing layers per scale yield $5.27/5.46/5.08/4.78/4.79\%$. Four and five layers have similar observed errors, and greater depth does not improve accuracy monotonically.

Anchor scales. The one- through five-scale configurations in panel (c) yield $6.32/4.59/5.19/5.08/5.11\%$.

Scale count also changes the anchor budget, and the five-scale configuration changes intermediate-scale counts. From one to five scales the anchor counts are $(30{,}000)$, $(30{,}000, 4{,}000)$, $(30{,}000, 4{,}000, 500)$, $(30{,}000, 4{,}000, 500, 80)$ and $(30{,}000, 8{,}000, 2{,}000, 500, 80)$.

Pooling. Max, mean, and max+mean pooling of point features into anchors (panel (d)) yield errors of $5.08/4.59/4.73\%$, respectively. Max+mean is the configuration used in Table 2.

Test-time averaging. Panel (e) evaluates the checkpoint of Table 2 for every averaging count $K = 1, \ldots, 8$ on one fixed set of sampled surface points, using cumulative prefixes of three independent sequences of anchor sets. The mean error is 5.13% at $K = 1$, drops to 4.88% at $K = 2$, and then changes little ($4.89/4.86/4.87/4.81/4.80/4.83\%$ for $K = 3, \ldots, 8$). Most of the gain comes from the first additional pass.

Test-time averaging of the baselines. AB-UPT and AB-GATr also sample their anchors at random, and averaging $K = 8$ passes lowers their errors in every column of Table 8. The largest reduction is for AB-GATr on DrivAerML volume pressure, from 13.07 to 11.87%.

Frame-invariant channels. Panel (f) removes one channel at a time from the configuration of Table 2. Every removal raises the error above the 4.79% of the full input, by 0.51 percentage points for flow alignment, 0.38 for handedness χ, 0.35 for confinement, 0.26 for streamwise position and 0.09 for curvature. No remaining channel recovers what a removed one carried, so the five are not redundant with each other on this benchmark. Removing flow alignment costs the most, consistent with it being the only channel that relates the surface orientation to the inflow.

D.2 Computational cost

Table 9: **Computational cost** on one AhmedML surface case on one L40S GPU, each model at its own precision and point count. Inference is one forward pass at $K = 1$, and the training step includes the optimizer update. Times are medians over repeated runs without data loading or compilation, and memory is the peak allocated.

			Inference ($K = 1$)		Training step	
Method	Points	Params	Time	Mem. (GiB)	Time	Mem. (GiB)
Non-equivariant baselines						
GINO (Li et al., 2023b)	200k	37.87 M	405 ms	19.7	480 ms	27.4
DeepONet (Lu et al., 2021)	200k	0.04 M	0.6 ms	0.1	5 ms	0.3
AB-UPT (Alkin et al., 2025)	200k	8.86 M	248 ms	1.1	796 ms	19.4
Equivariant baselines						
EqGINO (Kim et al., 2026)	200k	4.15 M	289 ms	4.3	653 ms	6.3
AB-GATr (Rygiel et al., 2026)	50k	1.62 M	293 ms	2.5	1300 ms	7.1
Backbone input variants						
Transolver(p)	200k	5.96 M	116 ms	1.2	369 ms	18.8
Transolver(φ)	200k	5.96 M	116 ms	1.2	369 ms	18.8
Transolver(p, φ)	200k	5.96 M	116 ms	1.2	369 ms	18.8
Transolver-3(p, φ)	200k	5.46 M	215 ms	2.1	718 ms	9.4
GraFINS = Transolver(φ, g)	200k	8.87 M	294.0 ms	4.1	744.9 ms	25.6

Table 9 reports implementation-level timings on one AhmedML surface-pressure test case. GraFINS takes 294.0 ms per inference pass and 744.9 ms per training step, against 116 and 369 ms for the backbone input variants, and its training step needs 25.6 against 18.8 GiB. A stage-wise profile of the inference pass attributes 39% of the time to the backbone and 53% to the anchor branch. Within the

branch, neighbour search and edge construction take 16% of the pass, the inverse-distance broadcast back to the $200{,}000$ points 15%, message passing 11% and pooling into the anchors 9%. Three quarters of the branch time thus lies in searches and scatter operations over all points rather than in learned layers, so it grows with the numbers of points and anchors rather than with the width of the model. The searches run on a uniform grid that returns the same neighbours as a direct search and shortens the inference pass by 24%. The three models that sample anchors at random, GraFINS, AB-UPT and AB-GATr, are evaluated with $K = 8$ in every accuracy table, which executes eight such passes each.

D.3 Unseen inflow

D.3.1 ShapeNet-Car ground-truth regeneration

For the inflow study, each vehicle remains in its dataset frame while the imposed velocity changes as $\mathbf{u}_\infty(\theta) = 20(\sin\theta, 0, \cos\theta)$ m/s, with $+y$ vertical. Evaluation uses the 100-shape test split and angle-specific pressure arrays on the ShapeNet-Car surface points.

CFD setup. Every ShapeNet-Car solution in this paper and every geometry of App. D.4.1 is solved with one OpenFOAM v2412 setup. The flow is steady incompressible RANS, solved with simpleFoam and the k-ω SST model with wall functions for $1{,}500$ SIMPLE iterations, at $U_\infty = 20$ m/s and a kinematic viscosity of 1.5×10^{-5} m^2/s. The domain spans 50×50 m in the horizontal plane and 15 m in height, with the body near its centre above a slip ground plane. Every far-field boundary carries a freestream condition, so the same domain serves every inflow direction. A 0.6 m background mesh is refined to levels 4 to 5 at the body surface with snappyHexMesh, with additional refinement around the body and in its wake, giving a median of 0.42 million cells per ShapeNet-Car case. Pressure is kinematic with a zero far-field reference, and the surface pressure is mapped onto the $3{,}682$ ShapeNet-Car points by nearest neighbour.

The flow-dependent features are recomputed for each angle.

D.3.2 Training recipes for the unseen-inflow study

The rows of Table 3 come from two sets of training runs. The five external baselines were trained with AdamW at 10^{-4}, weight decay 0.05, 25 warmup epochs and 225 cosine epochs, and were evaluated in fp32 with $K = 8$. The Transolver backbone and GraFINS were trained with Lion at 10^{-4} under the same weight decay, for 250 epochs including 25 warmup epochs, and were evaluated with $K = 8$ for GraFINS and $K = 1$ for the deterministic Transolver variants, where the two are numerically identical. Both sets share the same 711 training, 78 validation and 100 test geometries, the same case identifiers, the same nearest-mapped OpenFOAM ground truth, the same training-split min-max normalization, the same $3{,}682$ surface points and the same 250-epoch budget.

D.3.3 Unseen-inflow evaluation protocol

How each external baseline receives p and φ is given in App. C.2.

Every cell of Table 3(a) is a mean of per-case R^2 over the 100 ShapeNet-Car test shapes evaluated at that inflow direction. Stochastic models are evaluated with $K = 8$ test-time averaging and deterministic models with $K = 1$, where the two are numerically identical. This covers GraFINS, AB-UPT and AB-GATr at $K = 8$ and the remaining models at $K = 1$.

Single-pass comparison. With $K = 1$, GraFINS reaches R^2 of $0.398/0.469/0.533$ at $30/90/180°$, above every other model of Table 3 in each direction, including AB-UPT and AB-GATr at $K = 8$.

Pressure-maximum position. The median distance between predicted and ground-truth pressure maxima, normalized by shape diagonal, is $1.2/11.8/2.3/1.5\%$ at $30/90/180/330°$ for GraFINS, versus $7.8/30.0/52.9/7.8\%$ for a prediction that keeps the maximum of the $0°$ solution, as a model that ignores the inflow would. These are $K = 1$ predictions, and the pressure maximum stands in for the stagnation point.

D.3.4 RESULTS FOR EACH INFLOW DIRECTION

Table 10: **All five inflow directions** on ShapeNet-Car. Left block is per-case R^2 (↑), right block is per-case rel-L_2 in percent (↓), over the same 100 test shapes at every direction. 0° is the trained direction.

	$R^2\uparrow$					rel-$L_2\downarrow$				
Method	0°	30°	90°	180°	330°	0°	30°	90°	180°	330°
Non-equivariant baselines										
GINO	0.460	−0.026	−0.291	0.021	−0.033	58.5	82.8	90.7	82.3	82.1
DeepONet	0.482	0.102	0.003	−0.148	0.082	56.9	77.5	78.3	89.3	77.4
AB-UPT	0.683	0.097	−0.456	−0.420	0.102	40.1	77.5	96.7	98.9	76.3
Equivariant baselines										
EqGINO	0.538	0.185	0.044	0.373	0.155	52.8	73.5	78.6	64.7	73.7
AB-GATr	0.641	0.211	0.382	0.422	0.196	43.3	72.5	61.2	61.6	72.1
Backbone input variants										
Transolver(p)	0.634	0.047	−0.423	−0.115	0.036	43.7	79.8	96.1	87.9	79.3
Transolver(φ)	0.648	0.045	0.274	−0.012	0.037	43.9	79.4	67.5	82.5	78.9
Transolver(p, φ)	0.692	0.062	−0.081	−0.432	0.024	40.6	78.4	83.0	100.1	78.4
GraFINS	0.670	0.406	0.488	0.544	0.395	41.2	61.9	54.7	53.5	62.0

Table 10 extends Table 3(a) to all five evaluation directions and reports rel-L_2 as well. The oblique directions 30° and 330° are mirror images of one another and agree to within 0.038 in R^2 for every model, with a cross-model correlation of 0.995, which is why Table 3 reports only 30°.

D.3.5 PRESSURE PROFILES ALONG THE INFLOW AXIS

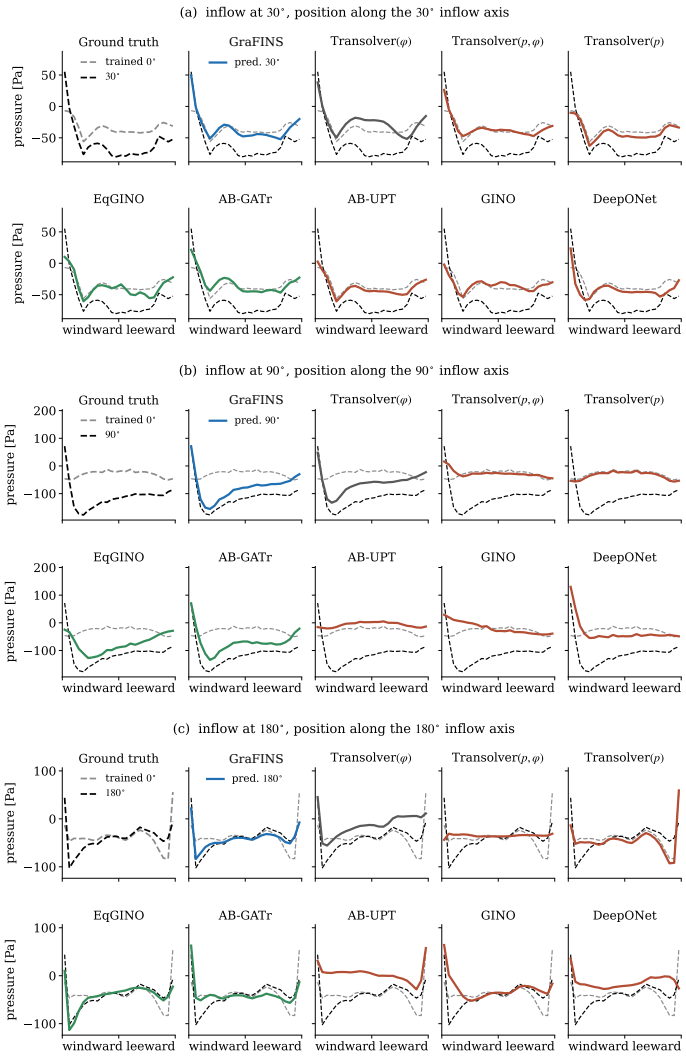


Figure 7: **Mean surface pressure along the inflow axis under unseen directions**, averaged over the 100 test shapes. Black dashed is the ground truth, grey dashed is the $0°$ ground truth on the same axis, and solid lines are the predictions of Table 3(a).

For each unseen direction, the 3,682 surface points of every test shape are binned by their coordinate along that direction's inflow axis, from the windward end to the leeward end, and the bin means are averaged over the 100 test shapes (Fig. 7). The grey profile is the one a model reproduces when it ignores the inflow. The models that receive coordinates stay near it at $30°$ and $90°$, and Transolver(p) and AB-UPT stay near it at $180°$ as well. The models that receive φ without coordinates follow the ground truth, and GraFINS follows it most closely. The same response measured as a position, the centre of the highest 1% of the surface pressure divided by the body length along the inflow axis, gives at $180°$ a median distance from the solution of 0.012 for GraFINS, 0.019 for AB-GATr, 0.023 for EqGINO and 0.024 for Transolver(φ), against 0.246 for Transolver(p, φ), 0.977 for AB-UPT and 0.978 for Transolver(p). A prediction that keeps the $0°$ field scores 0.976. GINO and DeepONet place the region correctly, at 0.052 and 0.022, and are inaccurate elsewhere on the surface.

D.4 ARBITRARY GEOMETRIES

D.4.1 GENERATION OF THE ARBITRARY GEOMETRIES

The fifty geometries of Section 4.3 are generated for this paper. They test whether a model trained on cars responds to shapes that no vehicle dataset contains, while every other part of the problem stays that of ShapeNet-Car.

Shapes. The fifty geometries belong to sixteen shape families, including domes, blunt wedges, stepped blocks, hexagonal and triangular prisms, vertical cylinders, thickened T and cross forms, bent beans and twin bulbs. The families differ from cars and from each other in silhouette, aspect ratio, edge sharpness and concavity. Each is an analytic deformation of a closed surface whose shape parameters are drawn by Latin hypercube sampling, and none contains a vehicle part.

Surfaces. Every geometry is a closed mesh of $7{,}360$ triangles on the $3{,}682$ points of the ShapeNet-Car surface template, so a checkpoint trained there runs on it without resampling. Each surface is connected, closed and of genus zero, sits 0.08 m above the ground plane, and carries no thin wing or small hole.

Solutions. Each geometry is solved with the CFD setup of App. D.3.1 at 20 m/s, with the inflow at $0°$, $30°$ and $90°$, giving 150 solutions. The surface pressure is mapped onto the $3{,}682$ template points. Pressure is kinematic with a zero reference, so $C_p = p/(\frac{1}{2}U_\infty^2)$.

D.4.2 RESULTS BY SHAPE FAMILY

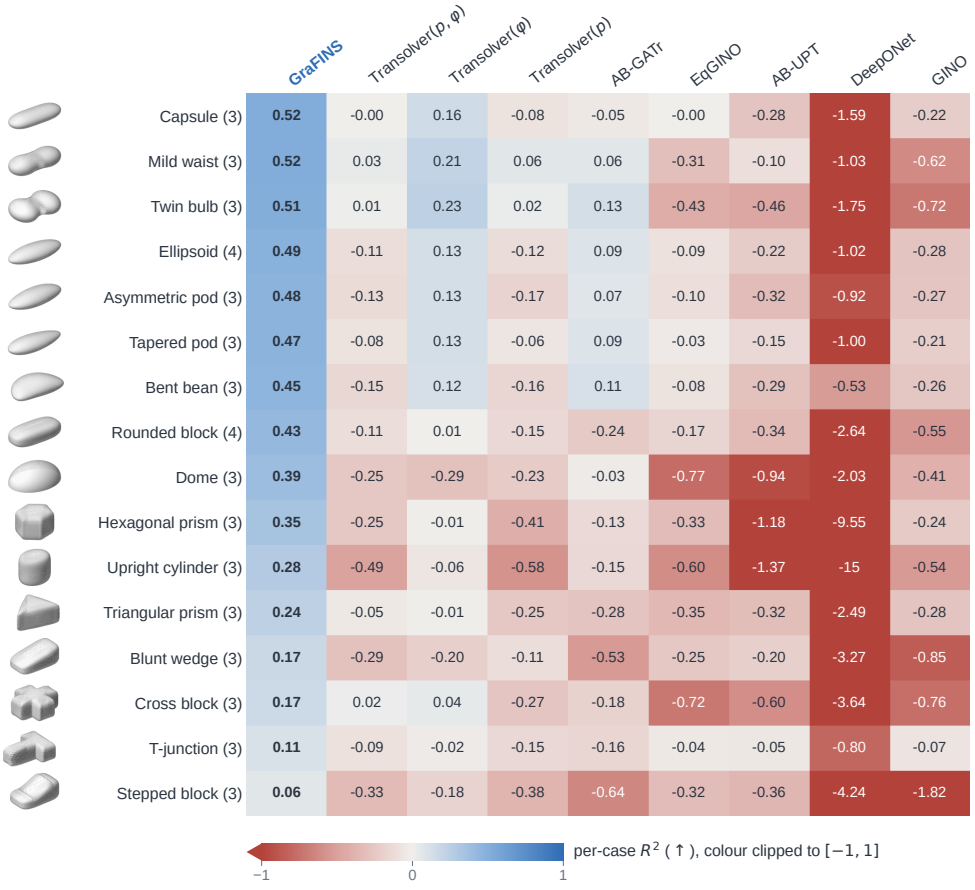


Figure 8: **Arbitrary geometries by shape family** (per-case R^2, ↑), averaged over the geometries of each family and over the 0°, 30° and 90° inflow. Each row label gives the number of geometries beside the family's representative shape (Figs. 17–32). Colour is clipped to $[-1, 1]$, and the printed values are not.

Figure 8 reports the R^2 of each shape family, and App. F.3 gives one representative per family.

D.5 ROTATED-FRAME EVALUATION ACROSS BENCHMARKS

Table 11: **Rotated frame, every benchmark** (rel-L_2 %, ↓). Conventions follow Table 4, whose DrivAerML block this table repeats beside AhmedML and Aneumo. Each benchmark is evaluated on its full test split, 50, 50 and 540 cases.

	DrivAerML (surf. p)			AhmedML (C_p)			Aneumo (p^*)		
				Scene rotation					
Method	0°	45°	90°	0°	45°	90°	0°	45°	90°
Non-equivariant baselines									
GINO (Li et al., 2023b)	21.07	103.5	98.45	10.50	76.71	87.83	49.04	54.27	52.46
DeepONet (Lu et al., 2021)	38.06	130.6	132.7	18.04	59.92	91.30	50.64	51.35	51.46
AB-UPT (Alkin et al., 2025)	4.74	82.68	91.21	4.80	64.18	90.56	43.81	46.80	47.45
Equivariant baselines									
EqGINO (Kim et al., 2026)	18.56	84.01	21.43	8.15	34.97	8.03	44.90	48.32	44.87
AB-GATr (Rygiel et al., 2026)	8.39	8.42	8.40	5.06	5.15	5.09	42.75	42.81	42.82
Backbone input variants									
Transolver(p)	10.12	113.8	123.0	5.95	93.42	101.1	67.95	68.25	67.43
Transolver(φ)	13.26	13.26	13.26	10.75	10.75	10.75	44.92	44.92	44.92
Transolver(p, φ)	4.53	89.03	91.55	5.14	65.33	78.28	46.34	51.56	52.73
GraFINS = Transolver(φ, g)	**4.45**	**4.45**	**4.45**	**4.73**	**4.73**	**4.73**	**36.39**	**36.39**	**36.39**

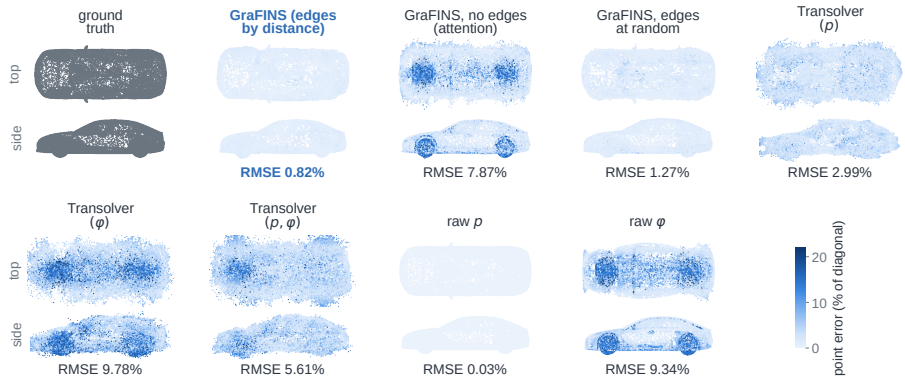
On Aneumo, rotating the scene changes the error of the non-equivariant models that receive coordinates by at most 6.4 percentage points, whereas the same models reach errors of 60% to 133% at 45° and 90° on the vehicle benchmarks. This small change does not mean that they handle Aneumo well. At 0° they are already less accurate than GraFINS, with errors of 43.8% to 68.0% against 36.4%, and they stay less accurate at every angle. Each Aneumo vessel lies in its own scanner frame, so the training cases share no pose that the coordinates could encode, and these models gain no pose-specific accuracy that a rotation could remove. On the vehicle benchmarks they do gain it from the shared pose, and a rotation removes it. The shortcut is therefore a property of aligned data rather than of the architectures.

D.6 SPATIAL PROBES ON ALL BENCHMARKS

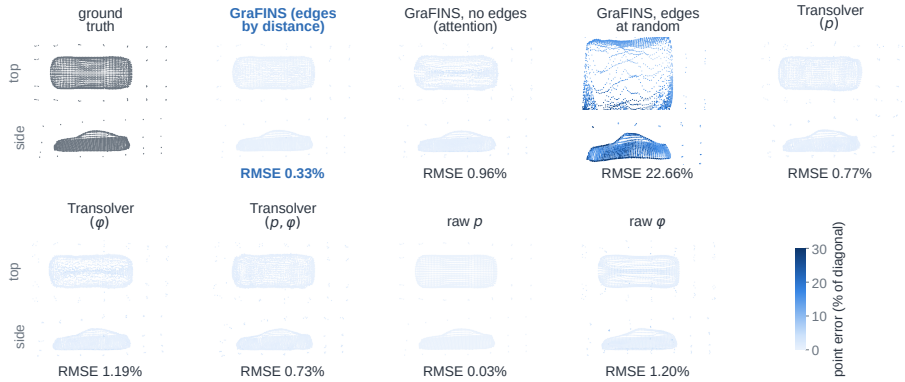
Table 12: **Coordinate and distance probes on frozen features.** RMSE is in percent of the bounding-box diagonal, averaged over test cases. On Aneumo, which has no shared pose, only the distance probe is informative. Raw p is an oracle that receives the coordinates. The AhmedML distance probe samples its point pairs differently from Fig. 5. ‡Trained at learning rate $5 \cdot 10^{-5}$.

	coordinate probe			distance probe	
representation	RMSE %	Kabsch %	lateral R^2	RMSE %	R^2
AhmedML (50 test cases)					
GraFINS (edges by distance)	**0.52**	0.49	1.00	**1.47**	**0.994**
GraFINS, edges at random	4.38	4.32	0.69	4.87	0.938
GraFINS, no edges (attention)	4.30	4.30	0.69	4.70	0.942
Transolver(p)	0.49	0.43	1.00	3.55	0.967
Transolver(φ)	4.33	4.33	0.69	4.97	0.935
Transolver(p, φ)	0.43	0.40	1.00	3.17	0.974
raw φ	4.57	4.51	0.67	4.83	0.939
raw p (oracle)	0.02	0.02	1.00	0.75	0.999
DrivAerML (50 test cases)					
GraFINS (edges by distance)	**0.48**	0.44	1.00	**1.62**	**0.993**
GraFINS, edges at random	0.70	0.68	0.99	1.92	0.990
GraFINS, no edges (attention)	4.59	4.57	0.55	5.76	0.907
Transolver(p)	1.71	1.68	0.97	12.60	0.556
Transolver(φ)	5.70	5.69	0.41	13.97	0.454
Transolver(p, φ)	3.16	3.13	0.84	13.57	0.485
raw φ	5.48	5.47	0.37	5.25	0.923
raw p (oracle)	0.02	0.02	1.00	0.63	0.999
ShapeNet-Car (100 test cases)					
GraFINS (edges by distance)	**0.38**	0.32	1.00	**0.92**	**0.994**
GraFINS, edges at random‡	12.78	8.79	−4.73	1.21	0.990
GraFINS, no edges (attention)‡	0.85	0.81	0.97	1.58	0.983
Transolver(p)	0.57	0.50	0.99	3.48	0.917
Transolver(φ)	0.97	0.93	0.96	3.33	0.924
Transolver(p, φ)	0.81	0.72	0.98	4.47	0.863
raw φ	1.13	1.03	0.97	1.31	0.988
raw p (oracle)	0.02	0.01	1.00	0.65	0.997
Aneumo (540 test cases; volumes without a shared pose)					
GraFINS (edges by distance)	14.53	10.19	–	**6.21**	**0.779**
GraFINS, edges at random	14.17	10.95	–	8.67	0.570
GraFINS, no edges (attention)	13.94	11.72	–	10.50	0.369
Transolver(p)	0.22	0.19	–	1.31	0.990
Transolver(φ)	14.20	11.56	–	10.30	0.393
Transolver(p, φ)	1.18	1.01	–	4.63	0.877
raw φ	13.58	12.55	–	10.58	0.360
raw p (oracle)	0.00	0.00	–	0.14	1.000

Both probes of Section 4.5 are repeated on DrivAerML, ShapeNet-Car and Aneumo with the same protocol. The representation is the input to the final prediction-head LayerNorm of a model trained for pressure only, frozen. A probe (feature standardization, Linear–GELU–Linear with 128 hidden units, AdamW, 100 epochs, selection on the validation split) is fitted on the training cases and scored on the test cases. The coordinate probe maps one point's feature to its normalized coordinates and is scored as pointwise RMSE in percent of the case's bounding-box diagonal, before and after optimal rigid alignment (Kabsch), and as R^2 of the axis that fails first, the lateral one (y for the two automotive surfaces, x for ShapeNet-Car). The distance probe reads $[\,|h_i - h_j|,\ h_i \odot h_j\,]$ of 1,024 point pairs per case and is scored as RMSE in percent of the diagonal and R^2. The ShapeNet-Car models use the same five frame-invariant channels as the other two surfaces. Table 12 lists every number, and Figs. 9 and 10 show the decoded clouds.



(a) DrivAerML



(b) ShapeNet-Car

Figure 9: **Decoded point clouds on DrivAerML and ShapeNet-Car**, one test case each, in the layout of Fig. 5. The RMSE under each panel is for the case shown.

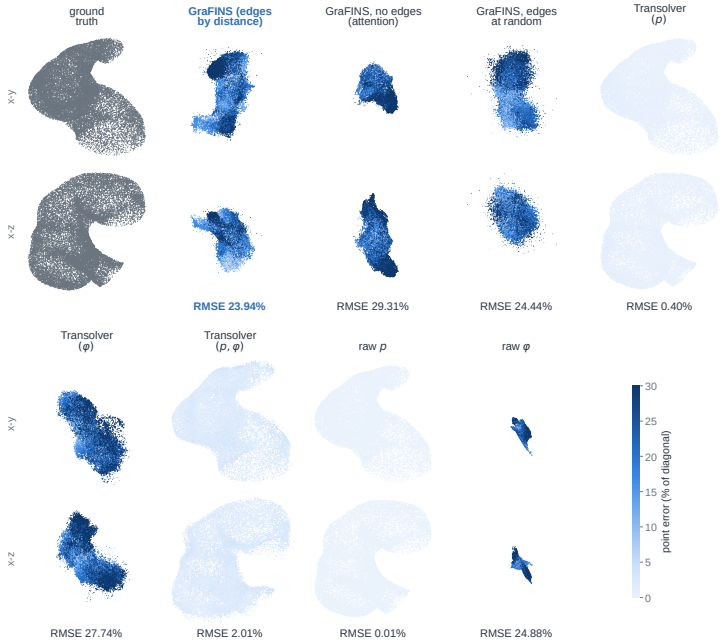


Figure 10: **Decoded point clouds on Aneumo**, one test case in x–y and x–z views.

Graph variants. The random-edge variant keeps the anchor sampling, pooling, message-passing depth, interpolation, and fusion paths, but uses k=13 fixed random neighbours. The set-attention variant replaces the whole anchor graph, including its sampling, pooling, and interpolation. On AhmedML, the set-attention model's pressure error is 11.1% versus 4.9% for GraFINS at $K = 1$.

Three observations carry over from AhmedML and one does not. First, among the representations that receive no coordinates, distance-based edges read relative distance best on every benchmark (R^2 0.994, 0.993, 0.994, 0.779), and removing the edges altogether (GraFINS, no edges) gives the lowest distance R^2 of the graph variants on DrivAerML, ShapeNet-Car and Aneumo. Second, the Transolvers that receive coordinates are not a ceiling. On the three surfaces GraFINS reads distance better than Transolver(p) and Transolver(p, φ) (on DrivAerML 0.993 against 0.556 and 0.485), whereas on Aneumo, where a coordinate is directly a metric quantity because cases share no pose, they read it better than the graph (0.990 against 0.779). Third, wherever absolute coordinates are recoverable at all, the axis that fails for the coordinate-free representations is the lateral one, and the failure sits where the local invariants are constant along that axis, the flat sides, roof and floor of the Ahmed body and the roof and underbody of the DrivAerML cars (lateral R^2 0.37–0.67 for raw φ against 1.00 for GraFINS). ShapeNet-Car has no such regions. Its small, everywhere-curved meshes let the raw frame-invariant features alone place a point to 1.13, so the coordinate probe cannot separate architectures there. What does not carry over is the random-edge ablation. It fails on AhmedML, nearly matches GraFINS on DrivAerML (0.70 against 0.48), and on ShapeNet-Car ends far below its own input (12.78 against 1.13 for raw φ) while its distance R^2 stays at 0.990. The lateral coordinate is blurred within each case (per-case correlation 0.80 against 0.998 for GraFINS, no sign flips) and scaled inconsistently across cases, which a pooled R^2 penalises and a within-case distance probe does not. On every benchmark, connecting points by distance yields the coordinate-free representation from which pairwise distance is read most accurately.

Table 13: **Coordinate probe on subsets of the per-scale graph embeddings**, with z_i and φ_i always in the probe input. Errors are RMSE in percent of the bounding-box diagonal. The shaded row is the full representation, and bold marks the lowest error per benchmark.

graph embeddings in the probe	AhmedML	DrivAerML
none	3.57	2.16
$g^{(1)}$	3.00	1.23
$g^{(2)}$	1.55	0.98
$g^{(3)}$	0.52	0.54
$g^{(4)}$	1.82	0.72
$g^{(1)}, g^{(2)}$	1.57	0.89
$g^{(1)}, g^{(3)}$	0.52	0.45
$g^{(1)}, g^{(4)}$	1.65	0.57
$g^{(2)}, g^{(3)}$	**0.47**	0.45
$g^{(2)}, g^{(4)}$	1.22	0.53
$g^{(3)}, g^{(4)}$	0.58	0.60
$g^{(1)}, g^{(2)}, g^{(3)}$	0.48	**0.42**
$g^{(1)}, g^{(2)}, g^{(4)}$	1.21	0.50
$g^{(1)}, g^{(3)}, g^{(4)}$	0.57	0.52
$g^{(2)}, g^{(3)}, g^{(4)}$	0.52	0.51
$g^{(1)}, g^{(2)}, g^{(3)}, g^{(4)}$	0.52	0.48

Scale subsets of the graph embedding. The coordinate probe is repeated on AhmedML and DrivAerML with the same GraFINS checkpoints, points and probe, but the probe input keeps the backbone state z_i and the features φ_i together with only a subset of the per-scale graph embeddings $g_i^{(\ell)}$, all taken from the same frozen head input. Scale $\ell = 1$ has the most anchors (30,000) and $\ell = 4$ the fewest (80). Table 13 lists all 16 subsets. Every subset lowers the error below that of z_i and φ_i alone (3.57 on AhmedML and 2.16 on DrivAerML). Among single scales, $\ell = 3$ gives the lowest error (0.52 and 0.54) and the finest scale the highest (3.00 and 1.23). The lowest error on

each benchmark comes from a combination of scales, $g_i^{(2)}$ and $g_i^{(3)}$ on AhmedML (0.47) and $g_i^{(1)}$, $g_i^{(2)}$ and $g_i^{(3)}$ on DrivAerML (0.42), below the best single scale. Adding the coarsest scale to these combinations raises the error slightly, to 0.52 and 0.48 with all four scales, which reproduces the values in Table 12. The probe decodes absolute coordinates, so it measures how well each subset places a point within the geometry and not how well it separates neighbouring points.

E CROSS-DATASET TRANSFER

If coordinates carry the pose of one dataset rather than the physics, a model that does not read them should carry over to a new dataset more readily. We test this by transferring a trained surface-pressure model between datasets in seven source–target pairs, both directions among the three external-aerodynamics benchmarks (DrivAerML, ShapeNet-Car, AhmedML) and the vascular pair Aneumo to CMHA. Every dataset keeps its native coordinate convention. DrivAerML and AhmedML share one, with the flow along $+x$ and the ground normal along $+z$. ShapeNet-Car has the flow along $+z$ and the ground normal along $+y$, and every Aneumo and CMHA vessel lies in its own scanner frame. A model that relies on its source frame therefore sees a different frame on the target in every pair except DrivAerML and AhmedML.

Each pair is run with four variants, differing only in what enters the per-point input: GraFINS (frame-invariant features plus the graph), Transolver(φ) (the same features, no graph), Transolver(p, φ) and Transolver(p). For each variant we evaluate the source model on the target with no fine-tuning (*zero-shot*), then fine-tune it on only N target cases (*few-shot fine-tuning*) and compare against the same architecture trained from scratch on the same N cases. The budget N is at most 64 on the automotive pairs, where a full training split has 400 to 711 cases, and at most 16 on the vascular pair. Within one budget the transferred and from-scratch runs differ *only* in initialization. Target cases, epoch budget, normalization, optimizer settings and evaluation are identical.

E.1 TRANSFER PROTOCOL

Prediction target. The automotive models predict a dimensionless surface pressure. AhmedML releases the pressure coefficient C_p. ShapeNet-Car stores kinematic pressure at an inflow speed of 20 m/s with zero reference pressure, which we divide by $\frac{1}{2}U_\infty^2 = 200\,\mathrm{m^2/s^2}$ to obtain C_p. DrivAerML keeps the curator scaling of App. A.3, which equals $0.697\,C_p$. A zero-shot model keeps the output units of its source dataset.

Input channels. The three variants that read features use the five surface channels of Table 7, streamwise position, curvature, flow alignment, confinement and handedness. Each channel is dimensionless and has one definition on all three datasets, so it measures the same quantity on the source and on the target. The channels are computed in the native frame of the dataset being evaluated, from that dataset's own inflow direction and ground normal. For the two variants that read coordinates, each geometry is centred at its centroid and divided by its maximum radius, which leaves the axis directions of the native frame unchanged. Transolver(p) has no other input that gives the flow direction, so it also receives the inflow unit vector as three constant channels. GraFINS uses the anchor counts of the target dataset (App. C.2), which do not change the number of weights.

Boundary conditions. The scalar boundary inputs of App. C.1 differ in number and meaning between the automotive datasets, and each takes one value for all cases of its dataset, so it carries no information about the flow of a case. We set them to zero on every dataset. For a transferred model, the first-layer weights applied to the constant boundary input of the source are folded into the first-layer bias, which leaves the source predictions unchanged.

Output scale and initialization. Each model predicts the target standardized with its mean and standard deviation over the N target training cases, and no validation or test case enters these statistics. A transferred model starts from all weights of the source model. Its last linear layer is rewritten so that, before fine-tuning, it predicts the same values as the source model. With source statistics (μ_s, σ_s) and target statistics (μ_t, σ_t), the weights and bias of that layer become

$$W_t = \frac{\sigma_s}{\sigma_t} W_s, \qquad b_t = \frac{\sigma_s b_s + \mu_s - \mu_t}{\sigma_t}.$$

Input channels that the source model lacks, the inflow vector of Transolver(p) transferred from DrivAerML or AhmedML, start with zero weights. A model trained from scratch has the same architecture and the same initialization seed. The zero-shot model is the source model with its own output statistics, applied to the target without any target data.

Target data and training. The N target cases, $N \in \{4, 16, 64\}$, are nested subsets $4 \subset 16 \subset 64$ of the target training split, the same for every model and initialization. Checkpoints are selected on the full validation split and tested on the full test split, 34 and 50 cases on DrivAerML, 50 and 50 on AhmedML and 78 and 100 on ShapeNet-Car. Each DrivAerML and AhmedML case contributes $200{,}000$ area-weighted surface cells and each ShapeNet-Car case all $3{,}682$ surface points, and the evaluated cells of a case are the same for every model. Few-shot fine-tuning and training from scratch run 600, 300 and 150 epochs for $N = 4$, 16 and 64, with AdamW at learning rate 10^{-4} and weight decay 0.05, a linear warm-up over the first 5% of the epochs, cosine decay to 10^{-6}, gradient clipping at 1 and bfloat16 autocast. The loss is the relative L_2 error of the target, and all models draw the same training points in each epoch. The validation error is computed every 5 epochs, and the checkpoint with the lowest validation error is tested in float32. GraFINS averages $K = 8$ passes in validation and test, and the other variants, which are deterministic, use one pass. Each condition is trained once.

Source models. The DrivAerML and AhmedML source models are trained on the full training split of their dataset. On ShapeNet-Car we train the four source models on its 711 training cases for 500 epochs with the same optimizer and schedule, so that their layers have the same shapes as those of the DrivAerML and AhmedML models.

Vascular pair. CMHA reports per-case scalar summaries rather than a pressure field (App. A.3), so this pair predicts one value per case. The Aneumo source models predict the point-wise dimensionless pressure $p^* = p/(\frac{1}{2}\rho V_{\max}^2)$, with zero pressure at the outlets. The matching CMHA target is $P^* = (\bar{p} - p_{\mathrm{ref}})/(\frac{1}{2}\rho V_{\max}^2)$, formed from the mean internal pressure $\bar{p}$ and the peak velocity $V_{\max}$ of the release, with $\rho = 1{,}060\,\mathrm{kg/m^3}$ and the reference pressure $p_{\mathrm{ref}} = 13{,}332\,\mathrm{Pa}$ ($100\,\mathrm{mmHg}$) of the CMHA simulations. $V_{\max}$ enters only this target and never the input. The prediction of a model for a case is the mean of its point-wise output over the $32{,}000$ sampled interior points. Both datasets are internal flows, so the four internal-flow channels of Table 7 and the inlet mass-flow rate have the same meaning on both, and the mass-flow rate stays an input. The channels and coordinates follow the preprocessing with which the Aneumo source models were trained. It normalizes each geometry by its axis-aligned bounding box instead of the centroid and maximum radius of App. C.1, and it takes the finite differences along the coordinate axes. Transolver(p) again receives the inlet unit direction of each vessel. Output statistics, initialization and test evaluation follow the automotive pairs.

Vascular training. Every run trains for 400 epochs at $N \in \{4, 8, 16\}$, with AdamW at learning rate 10^{-4} and weight decay 10^{-5}, a 20-epoch warm-up, cosine decay to 10^{-6}, gradient clipping at 1 and bfloat16 autocast. The loss is the squared error of the standardized case prediction. The N training cases are nested subsets $4 \subset 8 \subset 16$ of the 16 training controls. Checkpoints are selected every 10 epochs by the root mean squared error of P^* over the 8 validation controls, and the remaining 20 controls form the test split.

Error metric. On the automotive targets, the RRMSE of a case is its root mean squared error divided by the range of its target values, and the reported error is the mean over the test cases with equal weight. On CMHA each case has one value, so the root mean squared error is taken over the 20 test cases and divided by the range of their targets.

E.2 External aerodynamics: DrivAerML, ShapeNet-Car and AhmedML

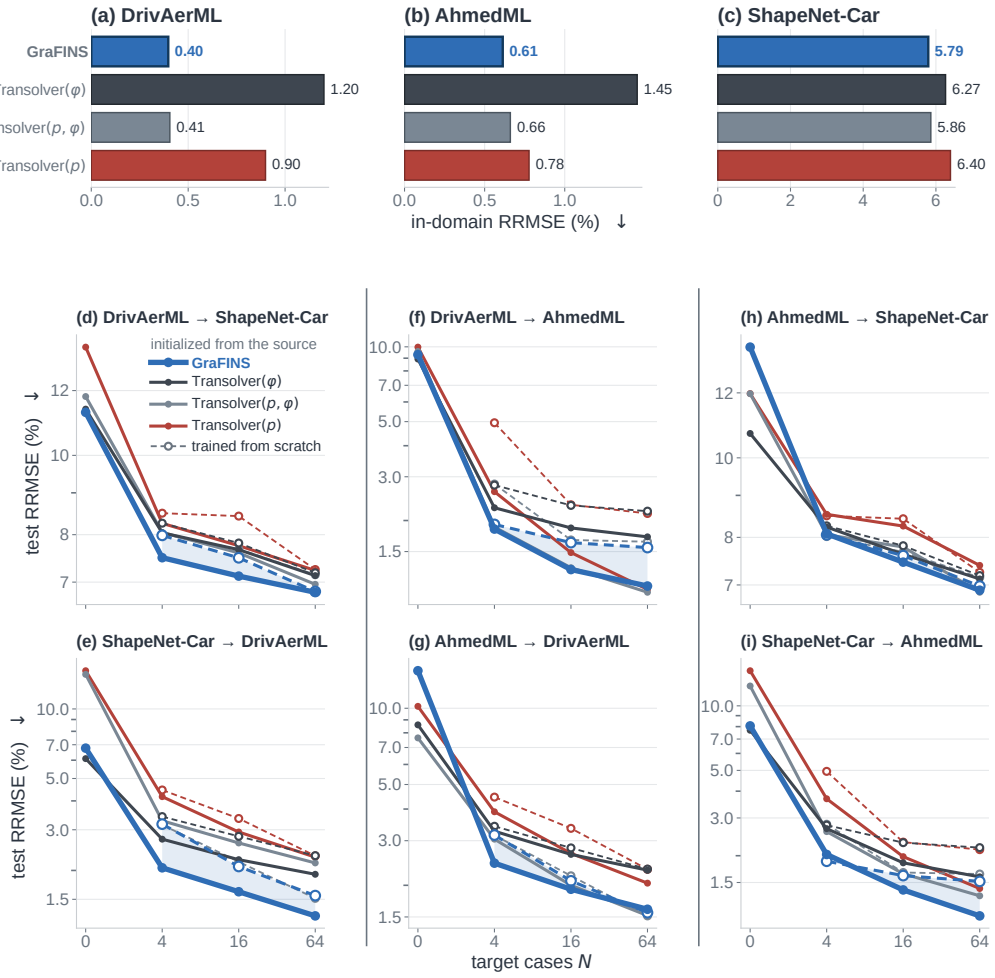


Figure 11: **Cross-dataset transfer among the three automotive benchmarks. (a–c)** In-distribution error of the four source models on each dataset's own test split. **(d–i)** The same models on the target dataset, from zero-shot ($N{=}0$) to few-shot fine-tuning on $N{=}4$ to 64 target cases, with each column holding one dataset pair in both directions. Solid curves start from the source model and dashed curves are trained from scratch, and the band marks GraFINS. The y scales differ between panels.

Removing coordinates helps transfer. Pooling the two coordinate-free variants against the two variants that receive coordinates, the coordinate-free side is more accurate in 19 of the 24 direction–budget conditions of Fig. 11, and in all six directions with only four target cases.

The graph improves accuracy in few-shot fine-tuning. GraFINS and Transolver(φ) see the same frame-invariant features and differ only in the graph. After few-shot fine-tuning GraFINS is more accurate in all 18 direction–budget conditions, by 25% to 37% at $N{\geq}16$ in four of the six directions. Without target cases, Transolver(φ) is more accurate in five of the six directions.

E.3 VASCULAR FLOW: ANEUMO TO CMHA

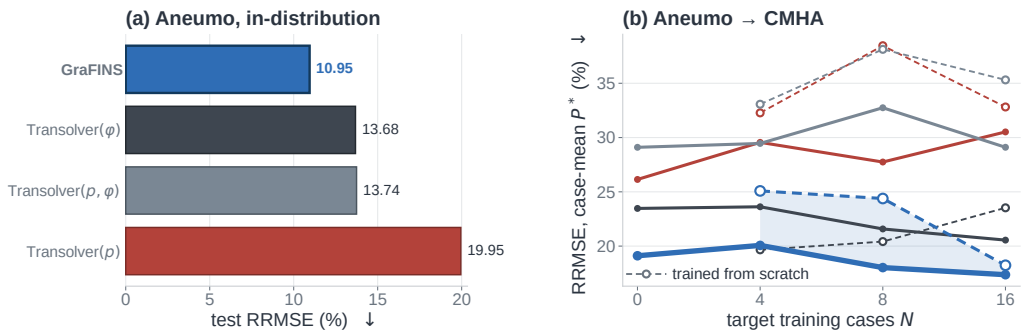


Figure 12: **Transfer between two aneurysm datasets. (a)** In-distribution pressure error on the 540 Aneumo test cases. **(b)** Transfer to CMHA, from zero-shot to few-shot fine-tuning on $N{=}4$ to 16 target cases, with the curve styles of Fig. 11. Panel (b) scores one case-mean scalar per case on a 20-case test split, so only the ordering within each panel should be compared.

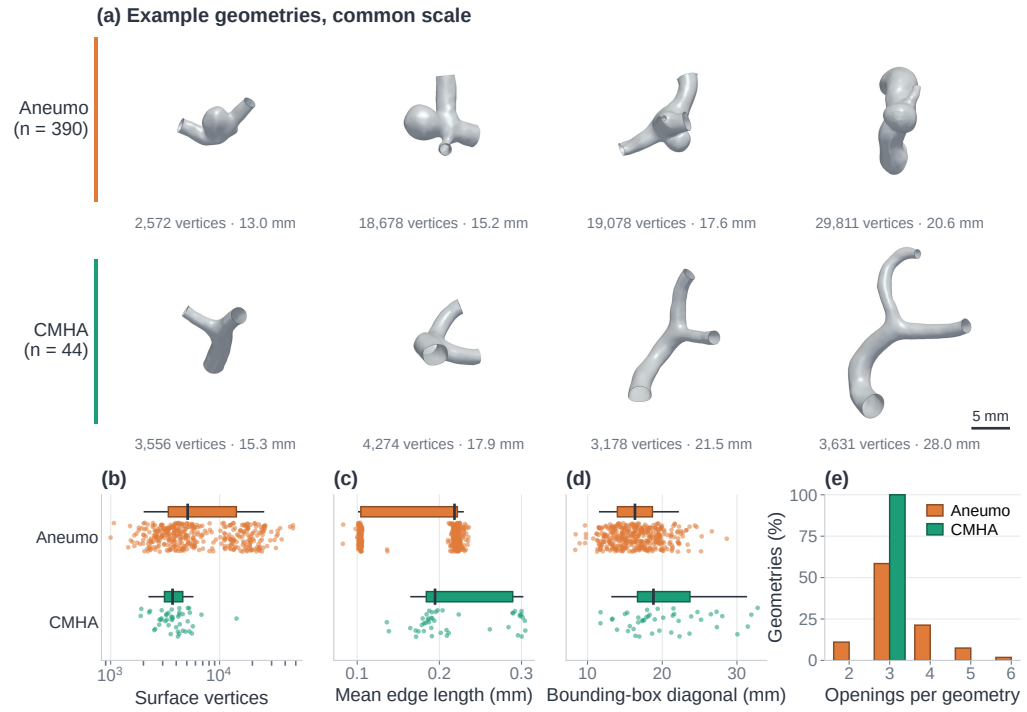


Figure 13: **Aneumo and CMHA geometries. (a)** Four geometries per dataset at the 15th, 40th, 65th and 90th percentiles of the bounding-box diagonal, in their scanner frames at one common scale. Dark loops mark the openings. **(b)–(d)** Surface vertices, mean edge length and bounding-box diagonal of every geometry. Boxes span the quartiles and whiskers the 5th to 95th percentiles. **(e)** Openings per geometry. Aneumo covers the 390 geometries of our split and CMHA the 44 controls.

On the vascular pair (Fig. 12), GraFINS is the most accurate transferred model at zero shot and at every budget, and the zero-shot ordering follows the coordinate content of the input. Without target cases GraFINS reaches 19.1, against 23.5 for Transolver(φ) and 26.1 and 29.1 for the two variants that receive coordinates. After few-shot fine-tuning it reaches 20.1, 18.0 and 17.4 for $N{=}4, 8, 16$,

against 23.6, 21.6 and 20.6 for the next-best transferred model. Transfer lowers its error below training from scratch by 20% at $N{=}4$ and 26% at $N{=}8$, and by 5% at $N{=}16$, where the target data alone becomes sufficient.

The two datasets also differ in geometry (Fig. 13). The Aneumo shapes are deformations of aneurysm geometries and carry a sac, with two to six openings, whereas the CMHA controls are aneurysm-free bifurcations with three openings in every case. The CMHA vessels are slightly larger, with a median bounding-box diagonal of 18.8 against 16.3 mm. Both datasets enter the model as $32{,}000$ interior points per case.

E.4 Summary of the seven settings

Across the seven settings, a model carries over to a new dataset better when it does not read coordinates. The coordinate-free inputs are more accurate after transfer in most direction–budget conditions, and in few-shot fine-tuning the graph adds a large and consistent gain on top of those features, on the automotive benchmarks and on the vascular pair. On the vascular pair GraFINS is also the most accurate model without target cases. On the automotive benchmarks the graph does not improve on the features alone without target cases in five of the six directions. The two largest gaps take AhmedML as the source, whose curvature channel has about a third of the standard deviation seen on the other two datasets, so a model trained there extrapolates on them.

F QUALITATIVE RESULTS

F.1 PREDICTIONS UNDER UNSEEN INFLOW

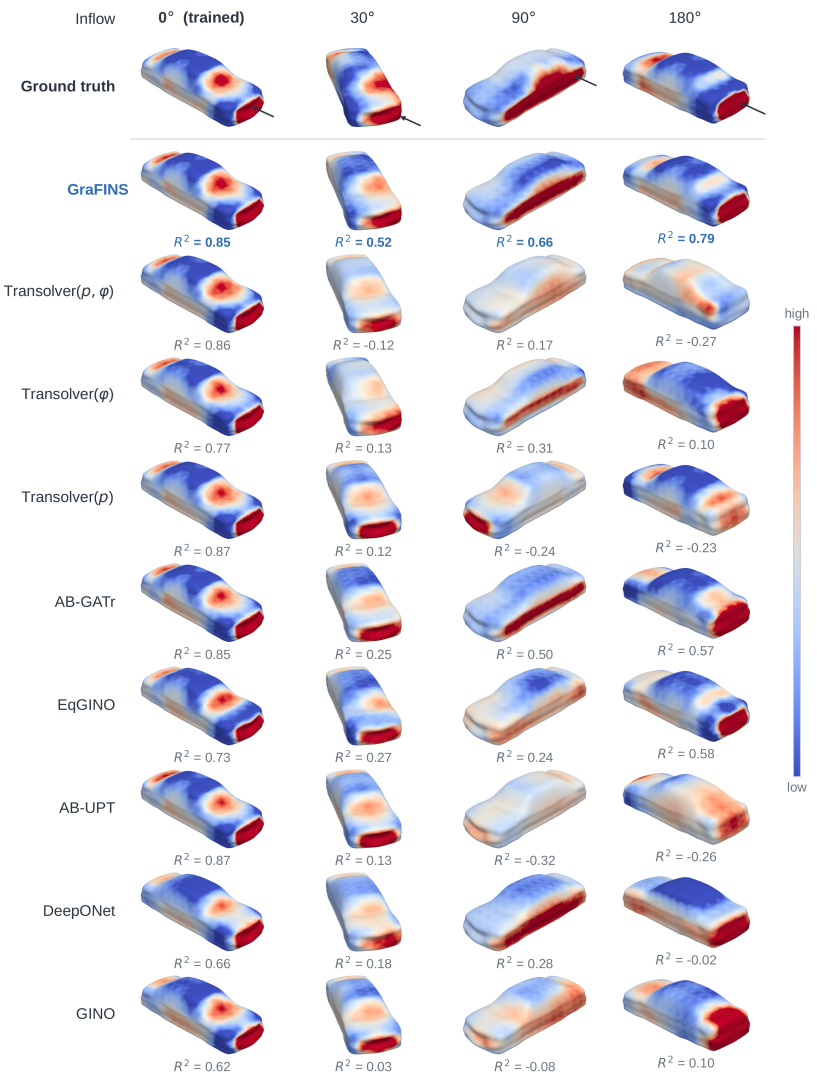


Figure 14: **All nine models under changed inflow** for the shape of Fig. 3, rendered as in that figure. Rows are the ground truth and the models of Table 3 in reverse order, and columns are the trained direction and three unseen directions. The 330° direction mirrors 30° and is reported in Table 10. Each R^2 is for this shape alone.

F.2 PREDICTIONS ON ANEUMO

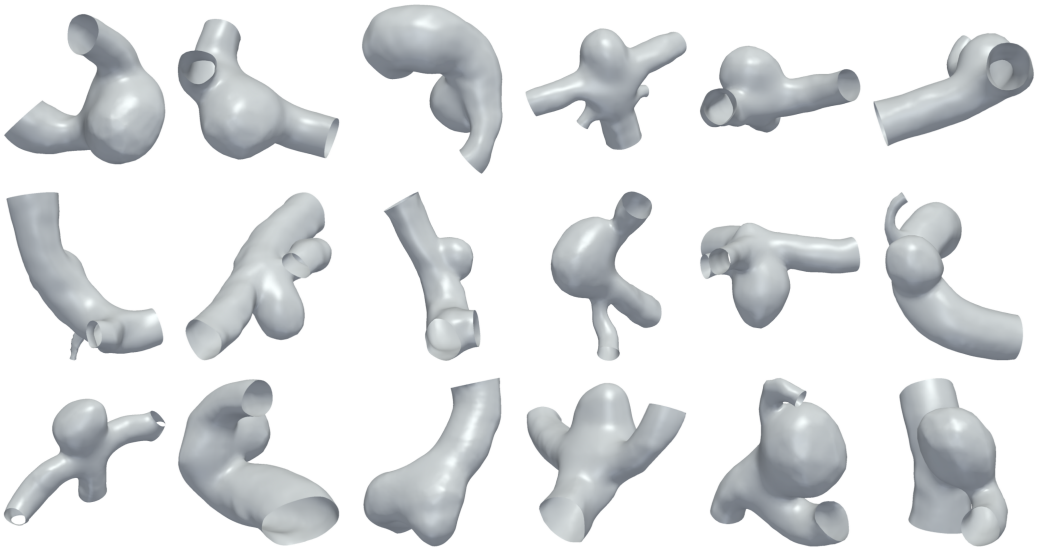


Figure 15: **Aneumo test geometries**, one from each of eighteen of the forty-five test anatomies, drawn in their native scan frames from one common view direction and each scaled to its cell.

Aneumo contains patient-specific cerebral aneurysms, each in the frame of its own scan, so the vessels differ in shape, branching and orientation and share no pose (Fig. 15). Figure 16 shows the pressure p^* predicted by every model of Table 2 on three test cases from distinct anatomies, chosen among the cases on which GraFINS has the lowest error of all nine models.

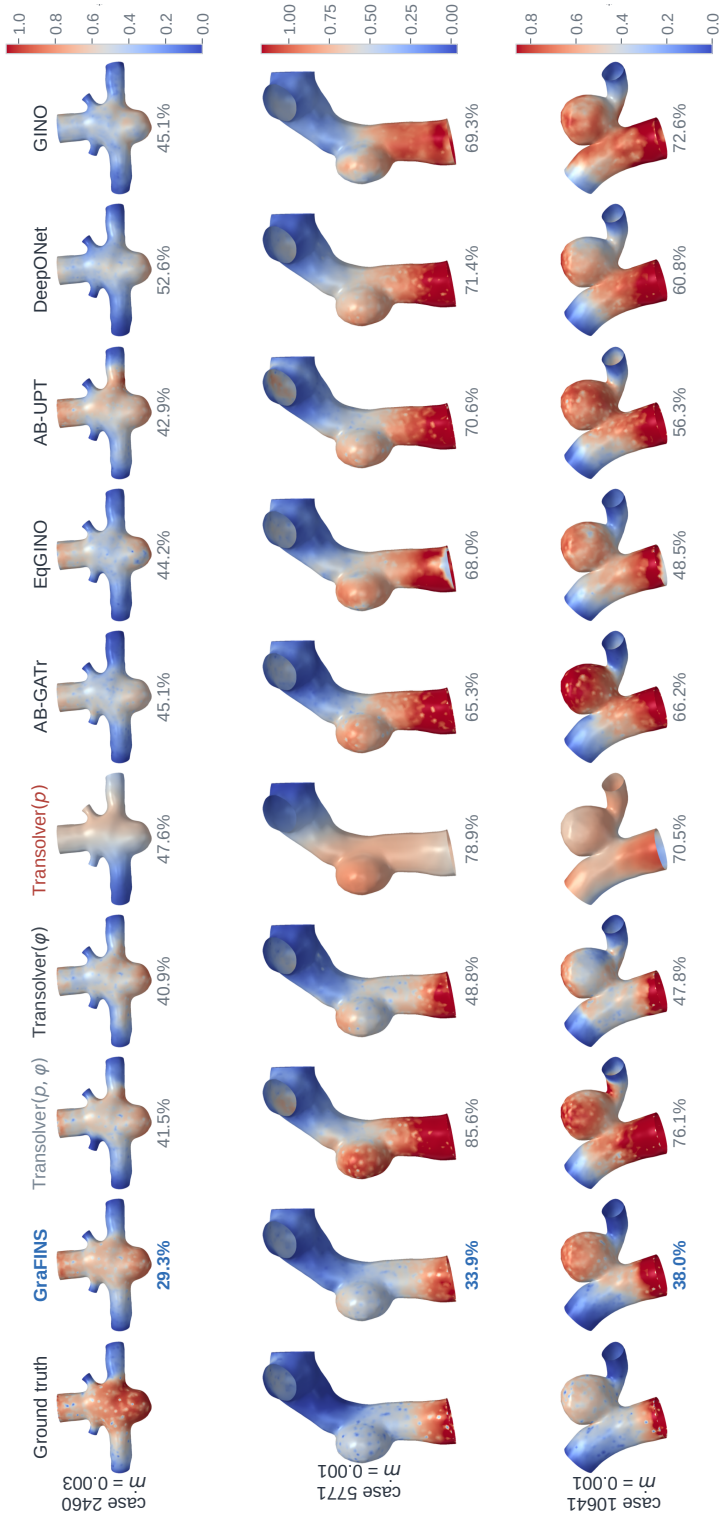


Figure 16: **Aneumo pressure p^*, all models of Table 2**, on three test cases with their mass-flow rate $\dot{m}$. Volume predictions are mapped to the vessel wall by inverse-distance weighting, and each row shares the colour range of its ground truth. The value under each panel is the relative L_2 error on the sampled volume points of that case.

F.3 PREDICTIONS ON ARBITRARY GEOMETRIES

The fifty arbitrary geometries of Section 4.3 fall into sixteen shape families, defined in App. D.4.1. The figures that follow show one geometry from each family, named by its family and ordered alphabetically. The representative is the geometry whose R^2 under GraFINS is closest to the median of its family, so the choice does not favour the cases where the surrogate does well.

Aggregate results by family are in App. D.4.2.

Each page carries one geometry at the trained 0° inflow and at the unseen 30° and 90°. The upper block is the predicted surface pressure with its R^2, the lower block is the absolute error against the ground truth with its relative L_2. Columns run from the ground truth through GraFINS to the weakest baseline, the reverse of the order in Table 3, so that our prediction sits beside the solution it is meant to reproduce. The ground-truth column repeats C_p in both blocks and is not an error map. Both colour ranges are set by the ground truth, GraFINS, Transolver(φ) and AB-GATr, so the remaining baselines can saturate, and the error range is the largest error on that geometry. Every metric is computed on the original $3{,}682$ surface nodes and each value is that single geometry, so it differs from the fifty-shape means of Table 3.

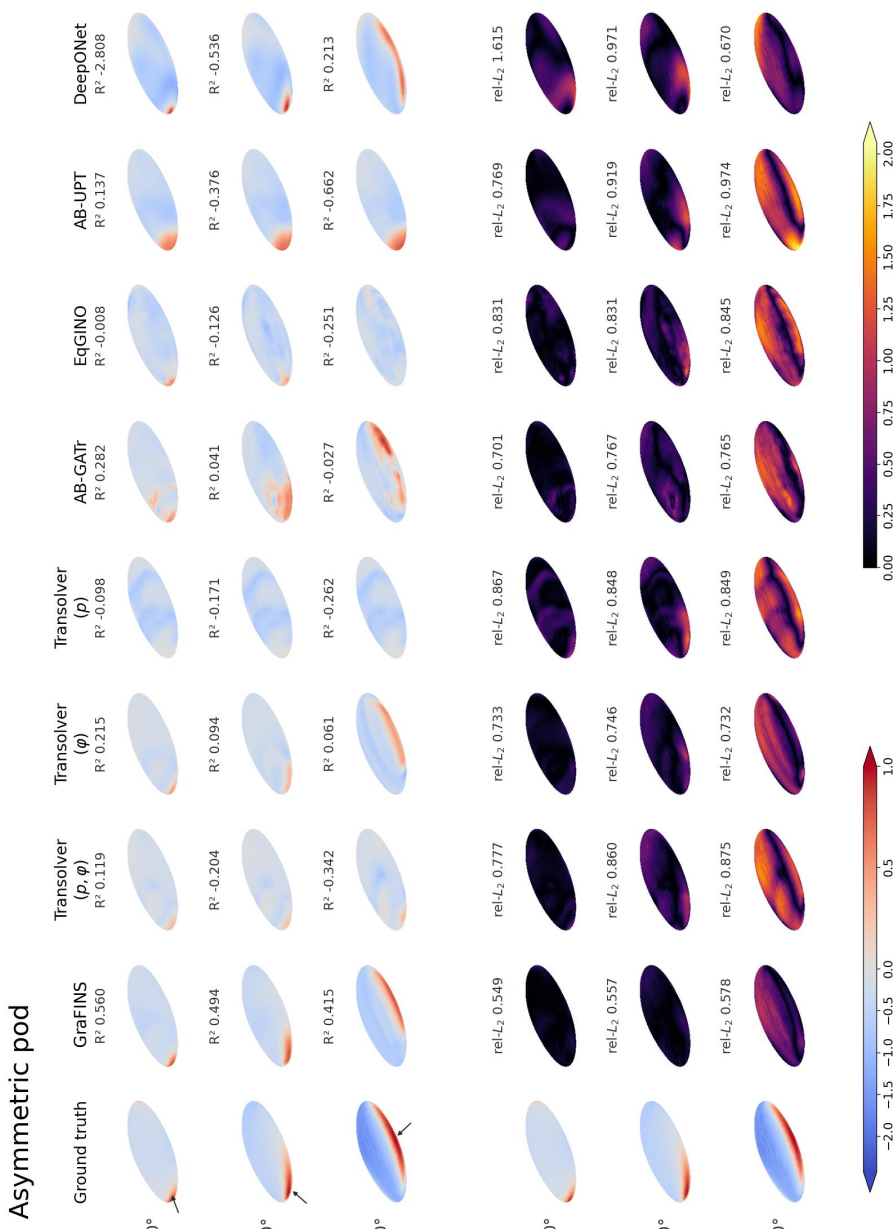


Figure 17: **Arbitrary geometries, asymmetric pod**, in the conventions described above. The arrow on each ground-truth panel marks the inflow, and the fifteen remaining families follow through Fig. 32.

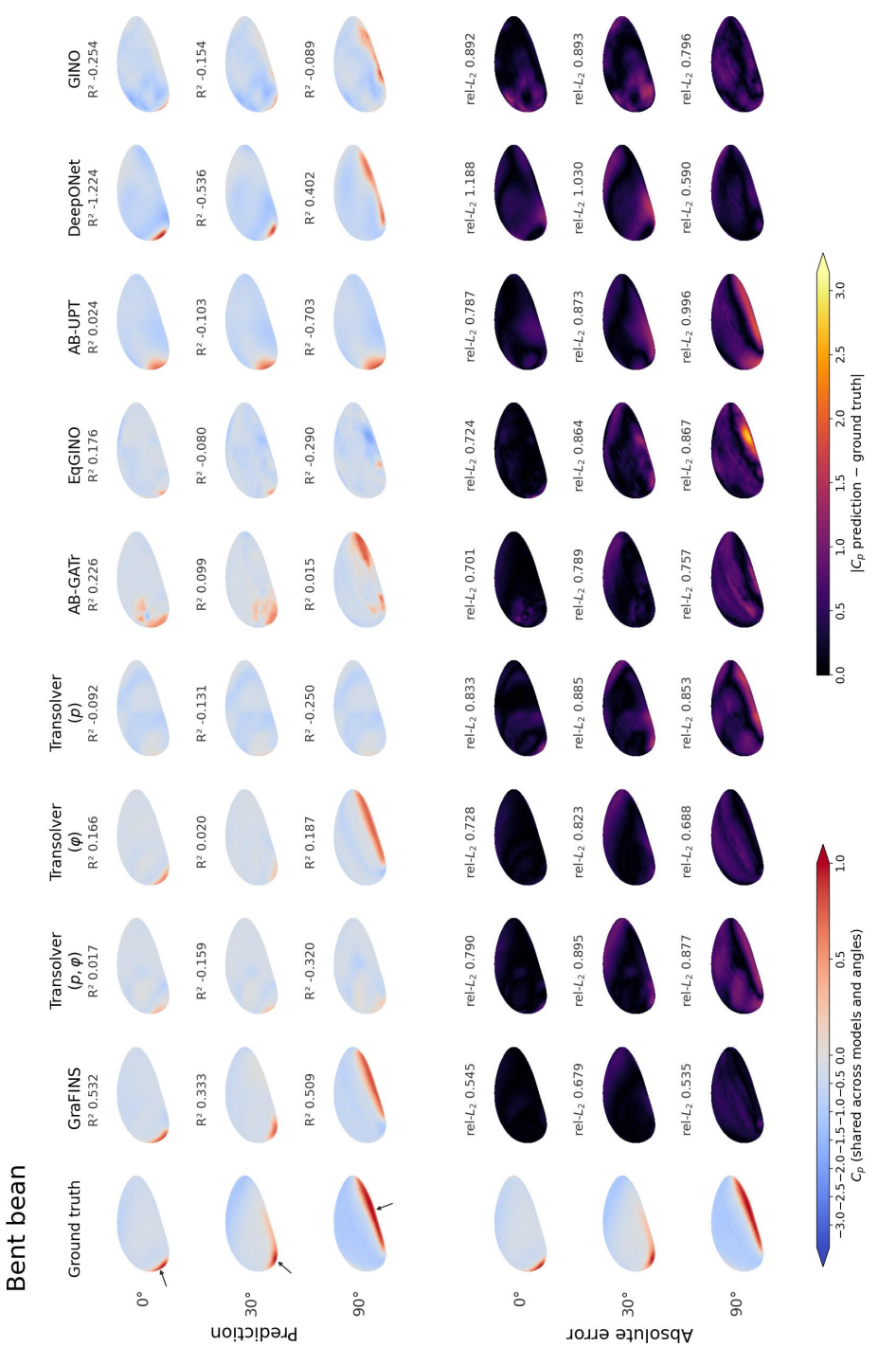


Figure 18: **Arbitrary geometries, bent bean.** Conventions follow Fig. 17.

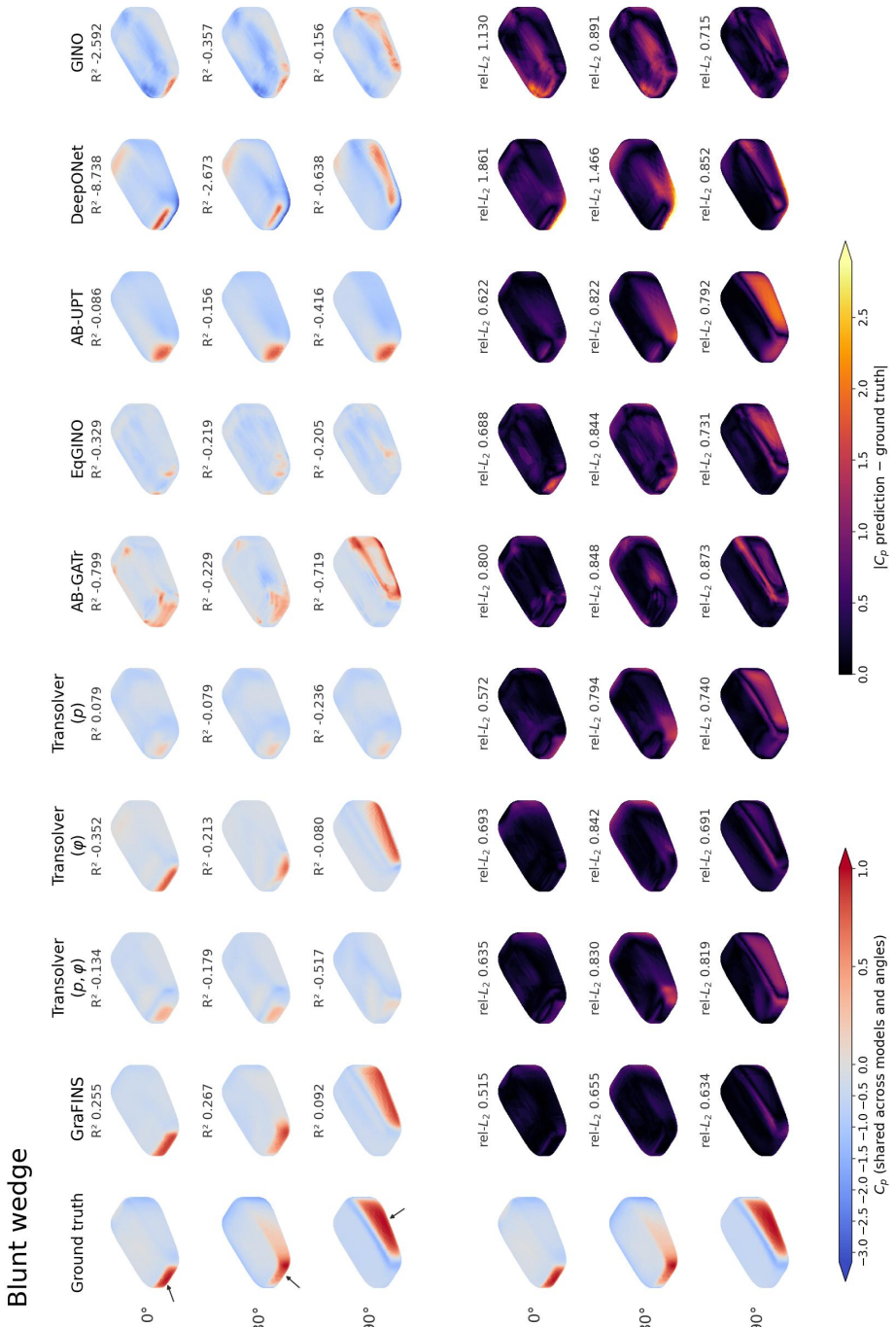


Figure 19: **Arbitrary geometries, blunt wedge.** Conventions follow Fig. 17.

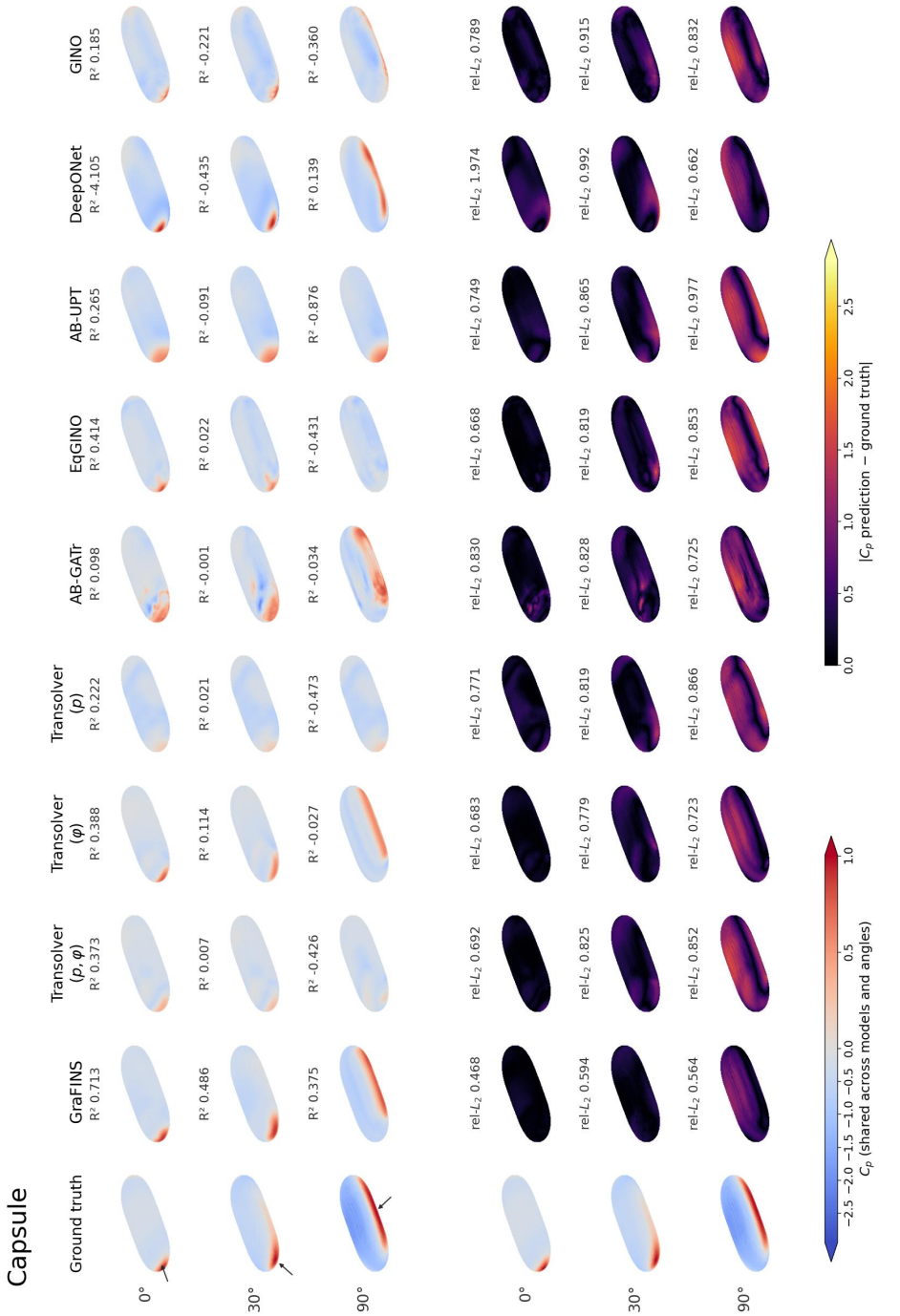


Figure 20: **Arbitrary geometries, capsule.** Conventions follow Fig. 17.

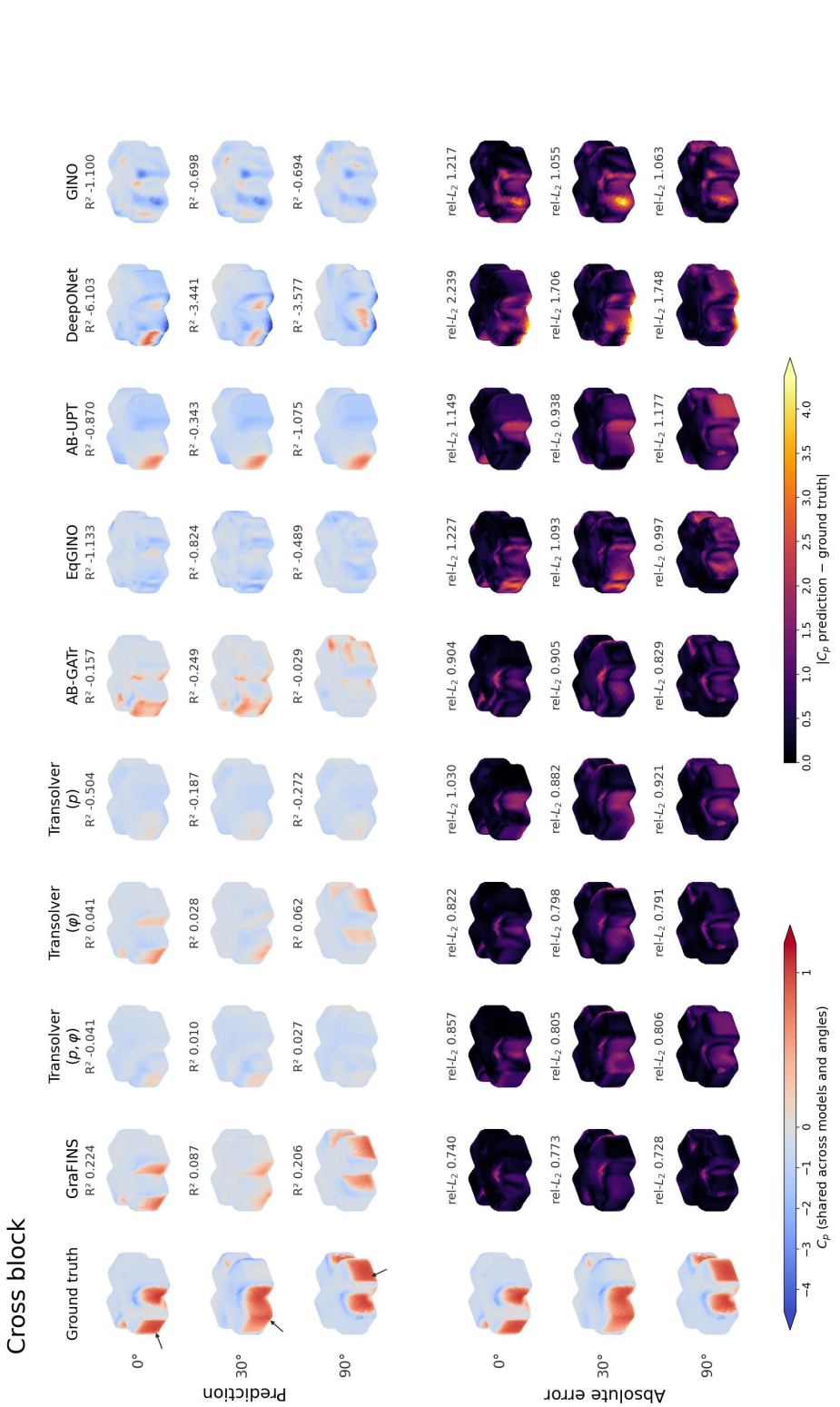


Figure 21: **Arbitrary geometries, cross block.** Conventions follow Fig. 17.

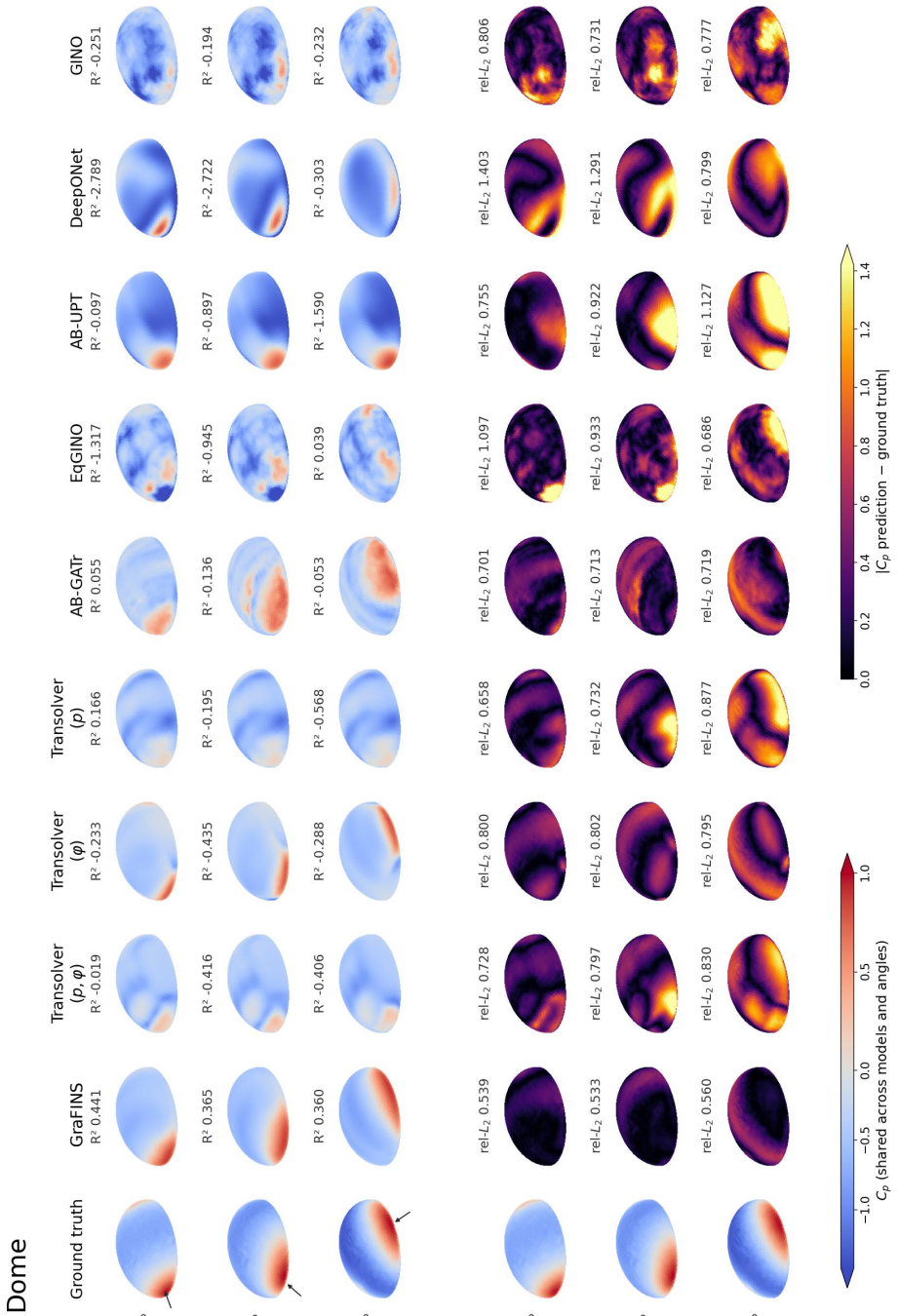


Figure 22: **Arbitrary geometries, dome.** Conventions follow Fig. 17.

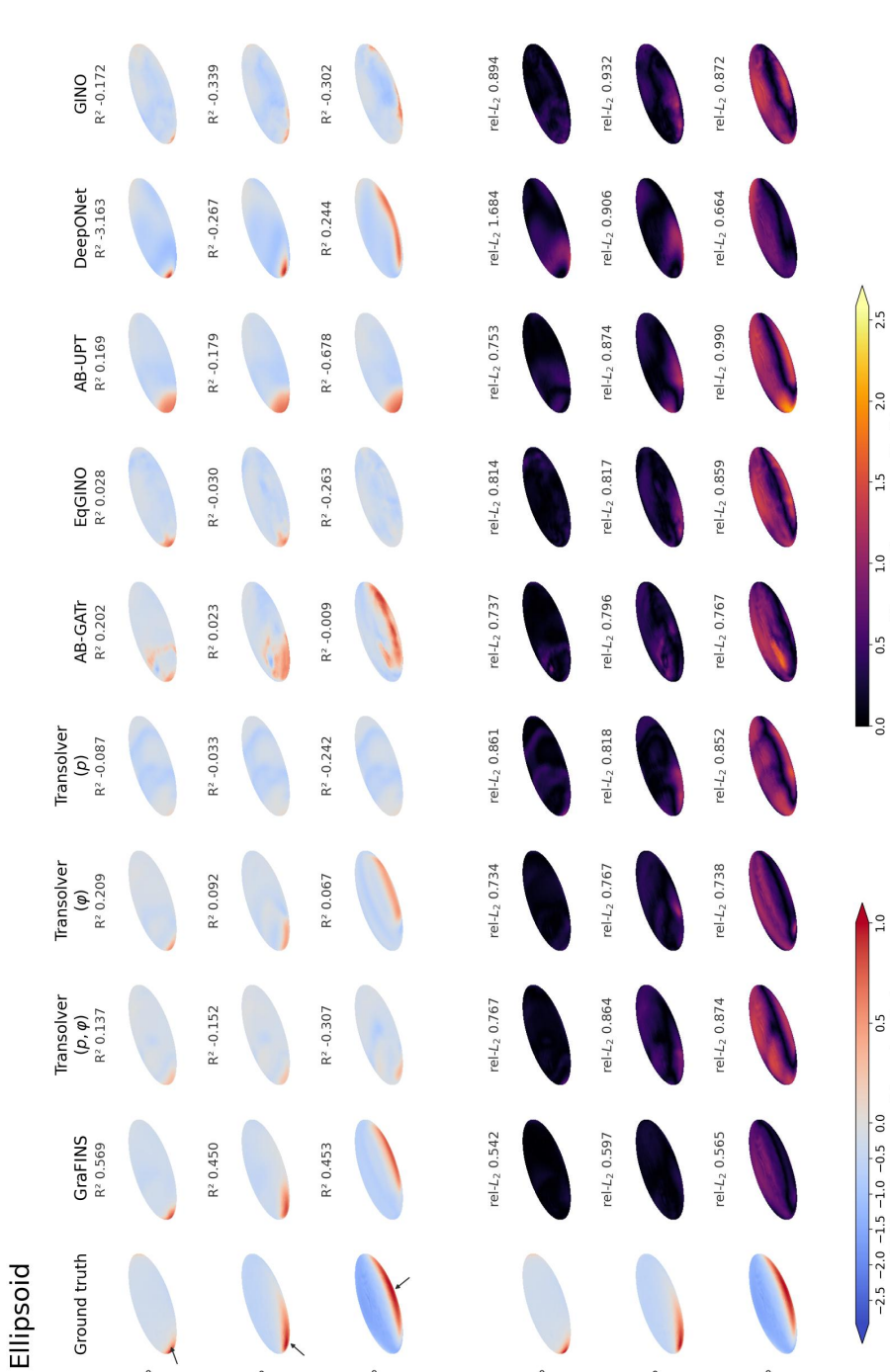


Figure 23: **Arbitrary geometries, ellipsoid.** Conventions follow Fig. 17.

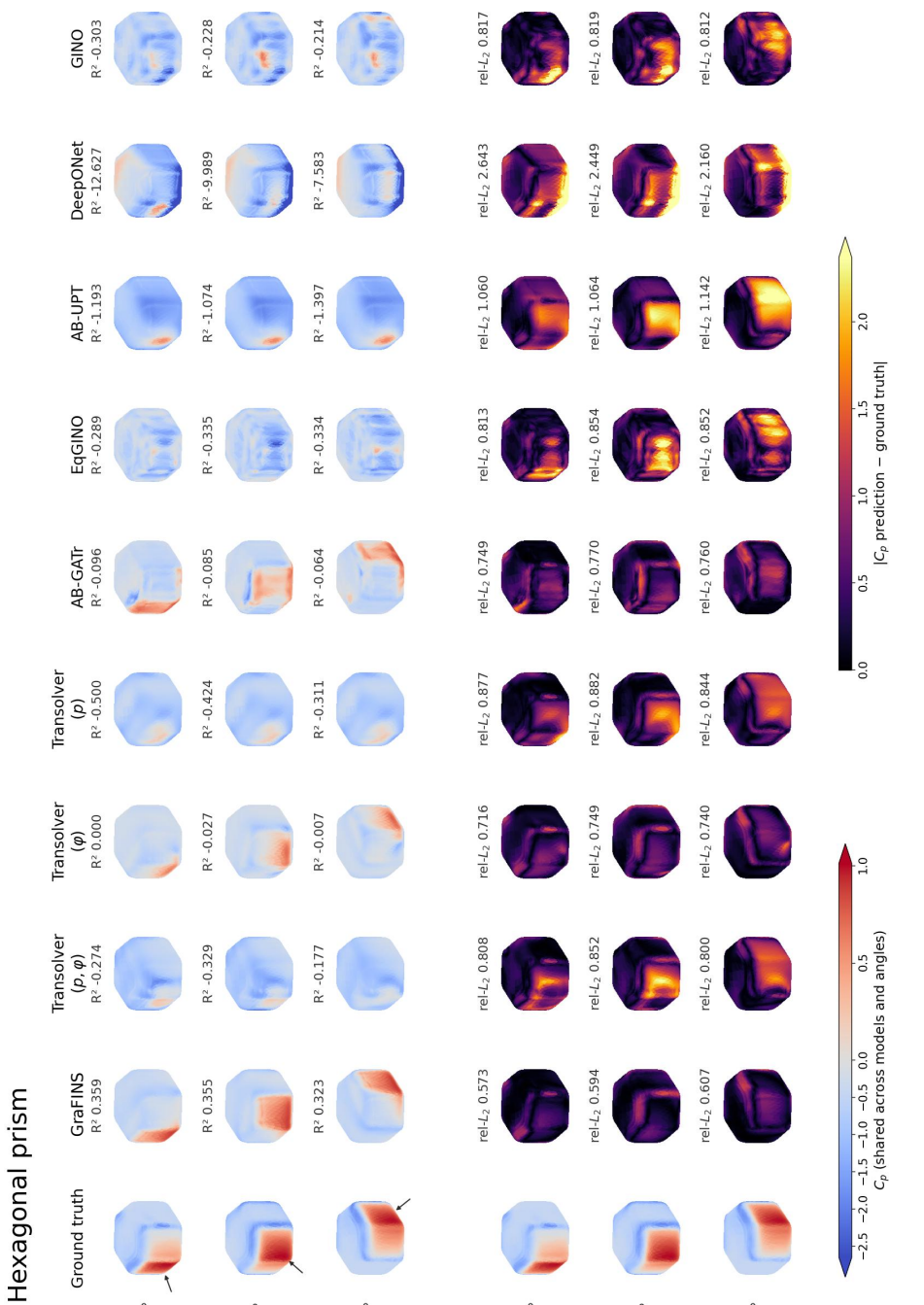


Figure 24: **Arbitrary geometries, hexagonal prism.** Conventions follow Fig. 17.

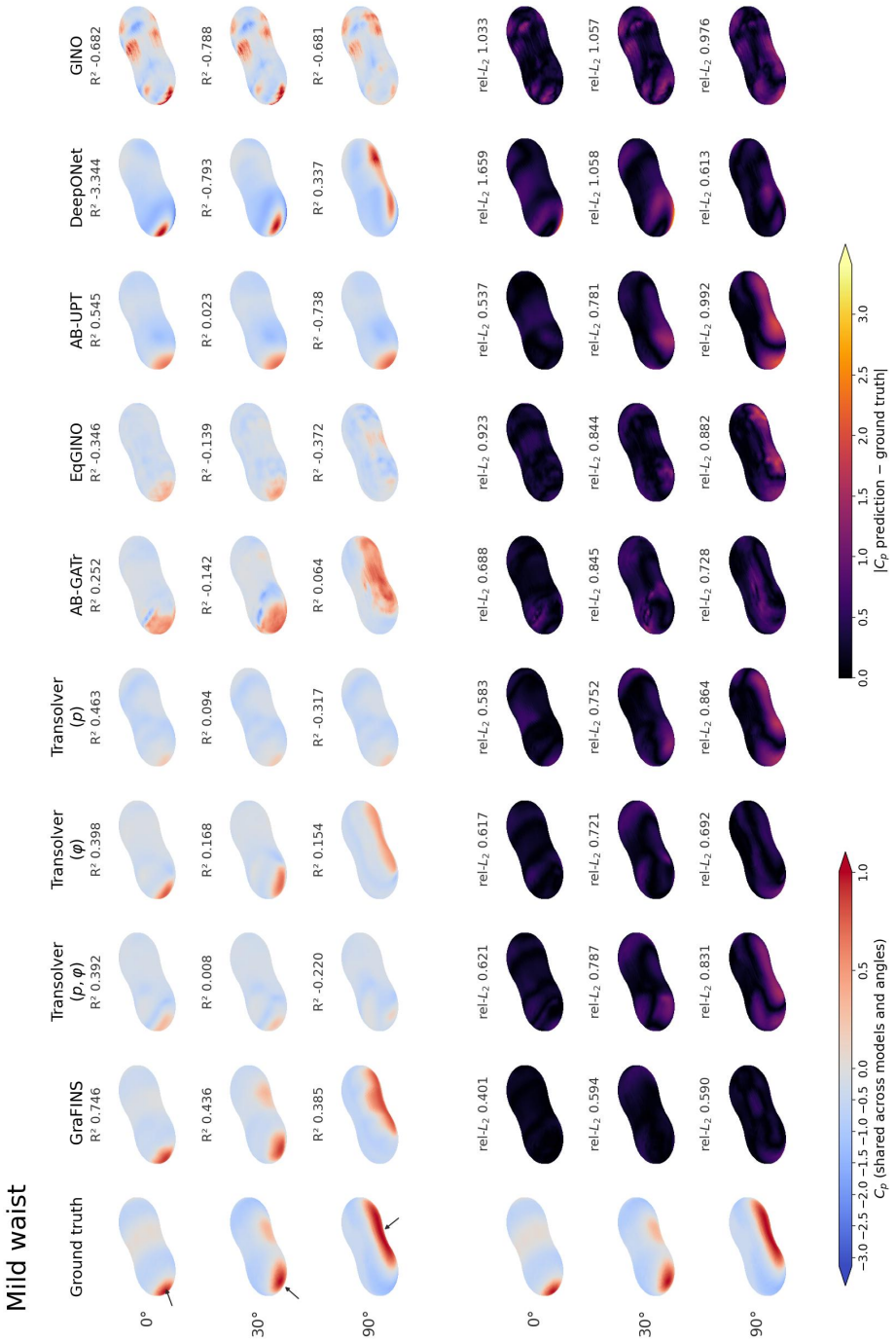


Figure 25: **Arbitrary geometries, mild waist.** Conventions follow Fig. 17.

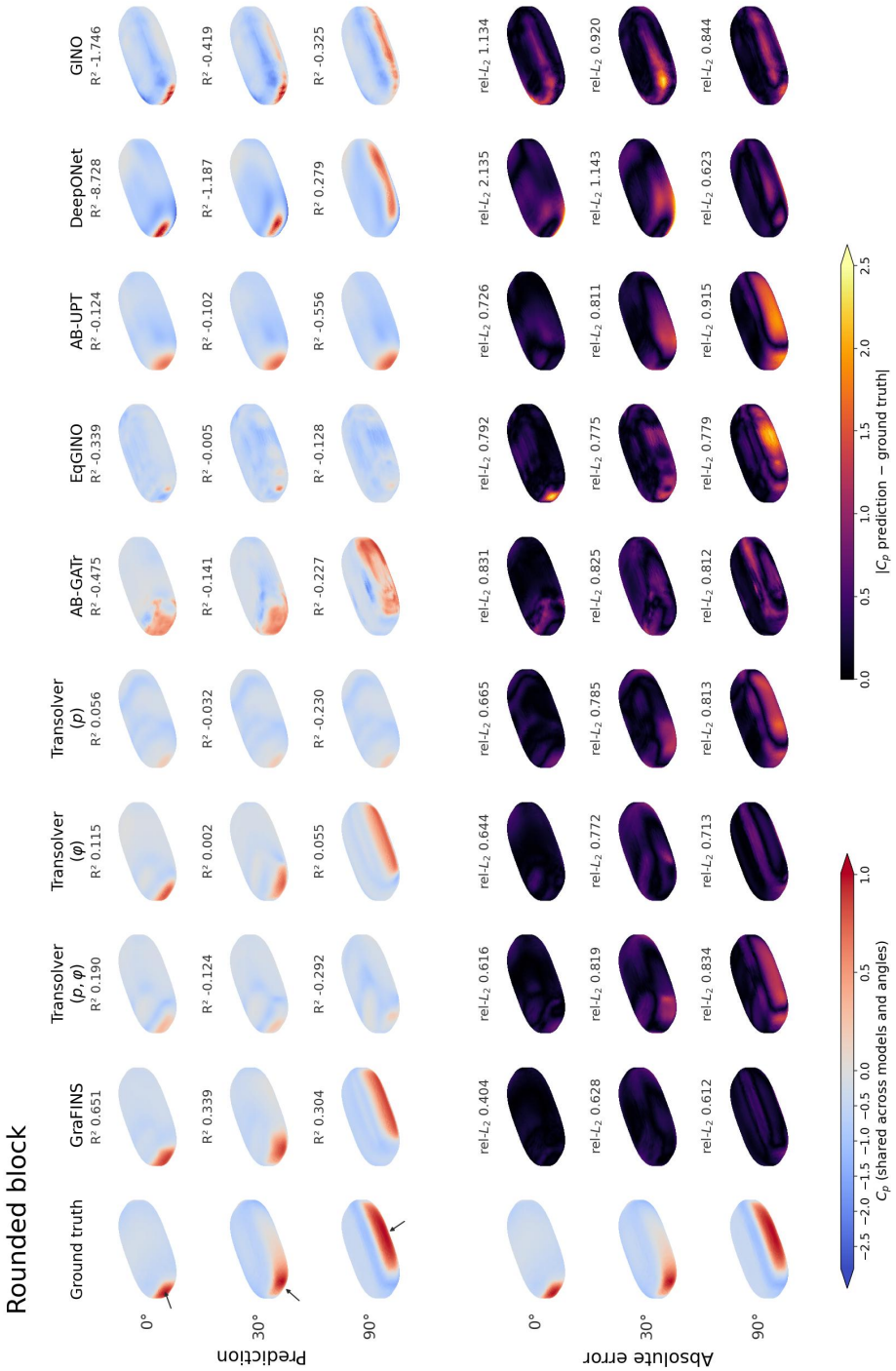


Figure 26: **Arbitrary geometries, rounded block.** Conventions follow Fig. 17.

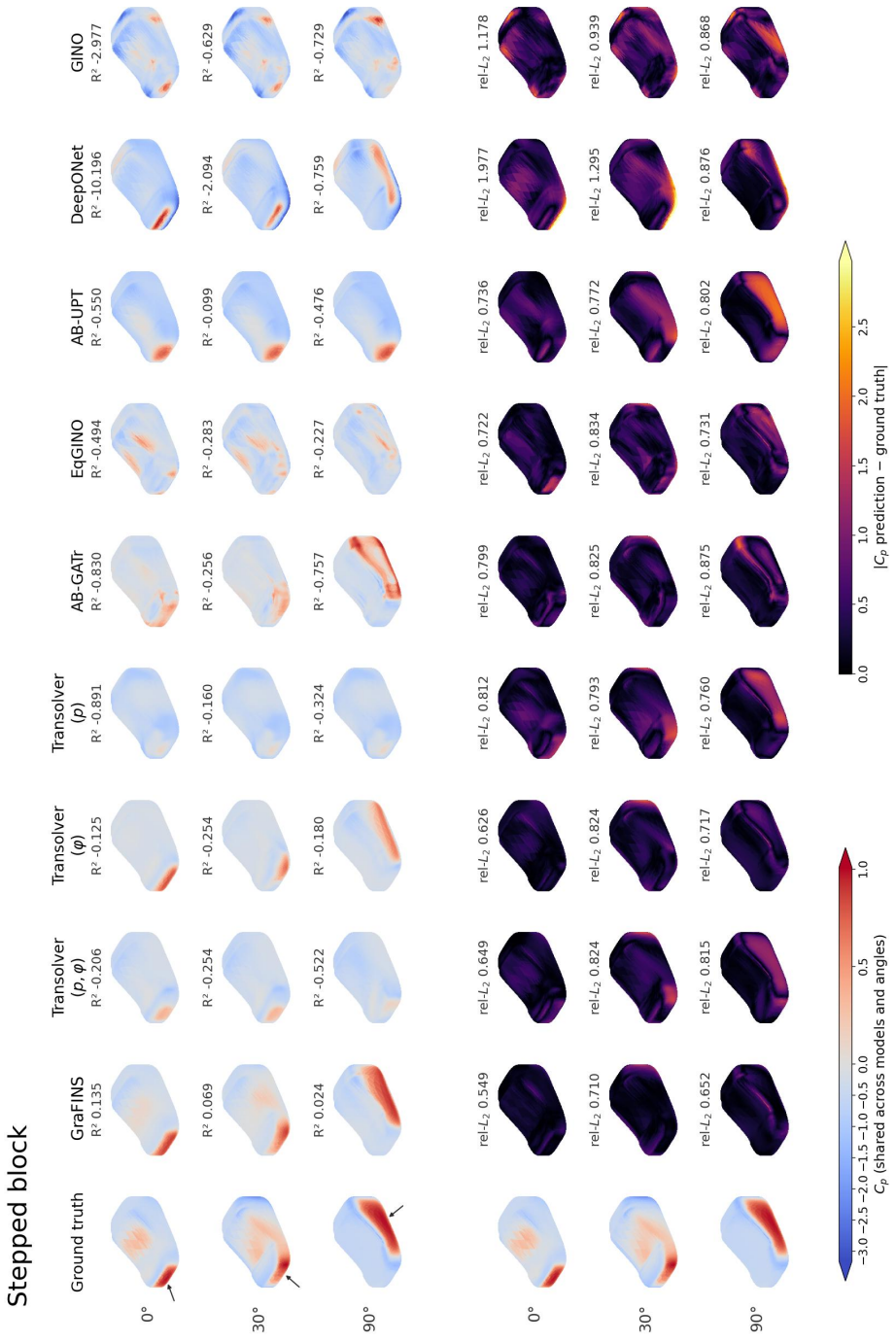


Figure 27: **Arbitrary geometries, stepped block.** Conventions follow Fig. 17.

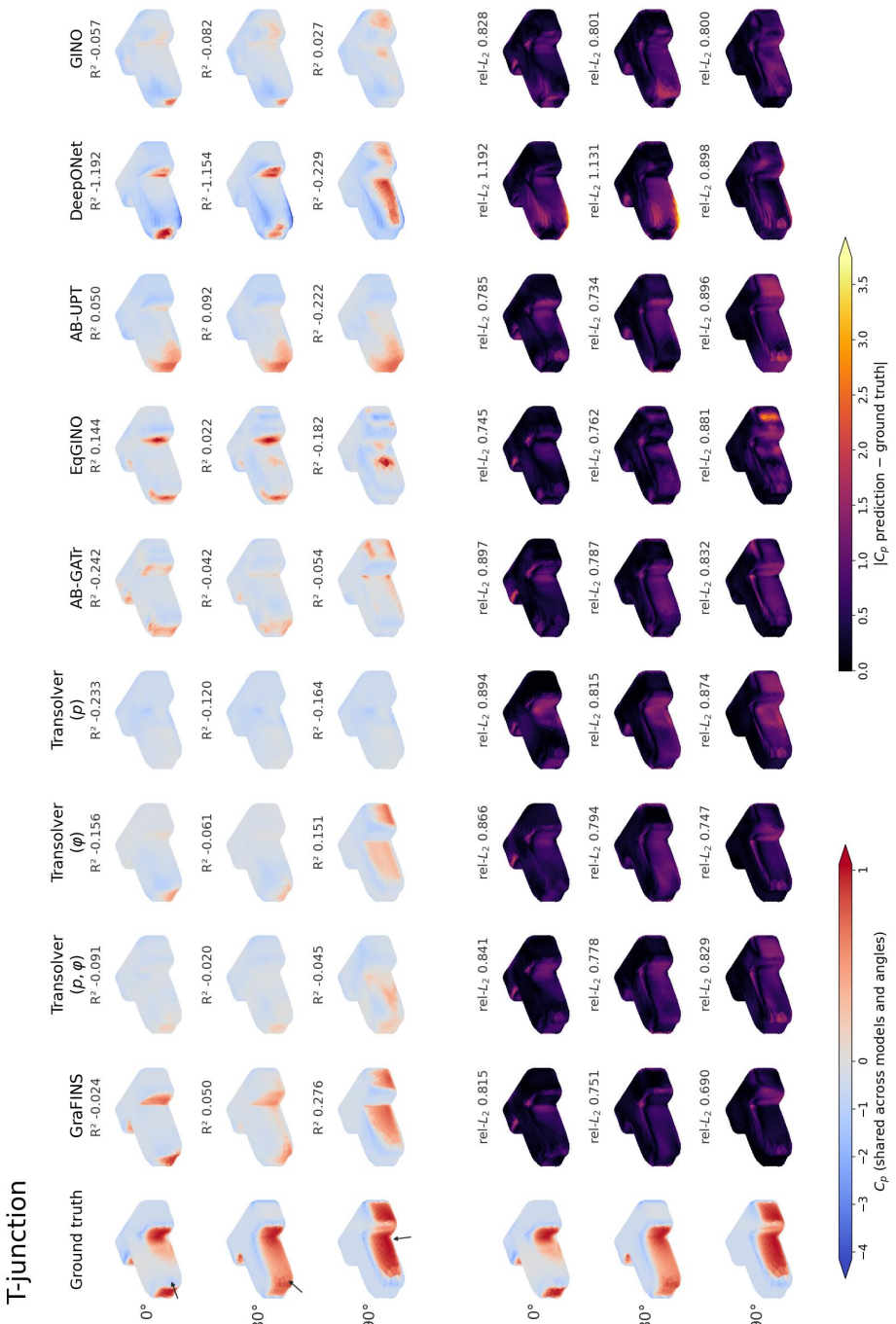


Figure 28: **Arbitrary geometries, t-junction.** Conventions follow Fig. 17.

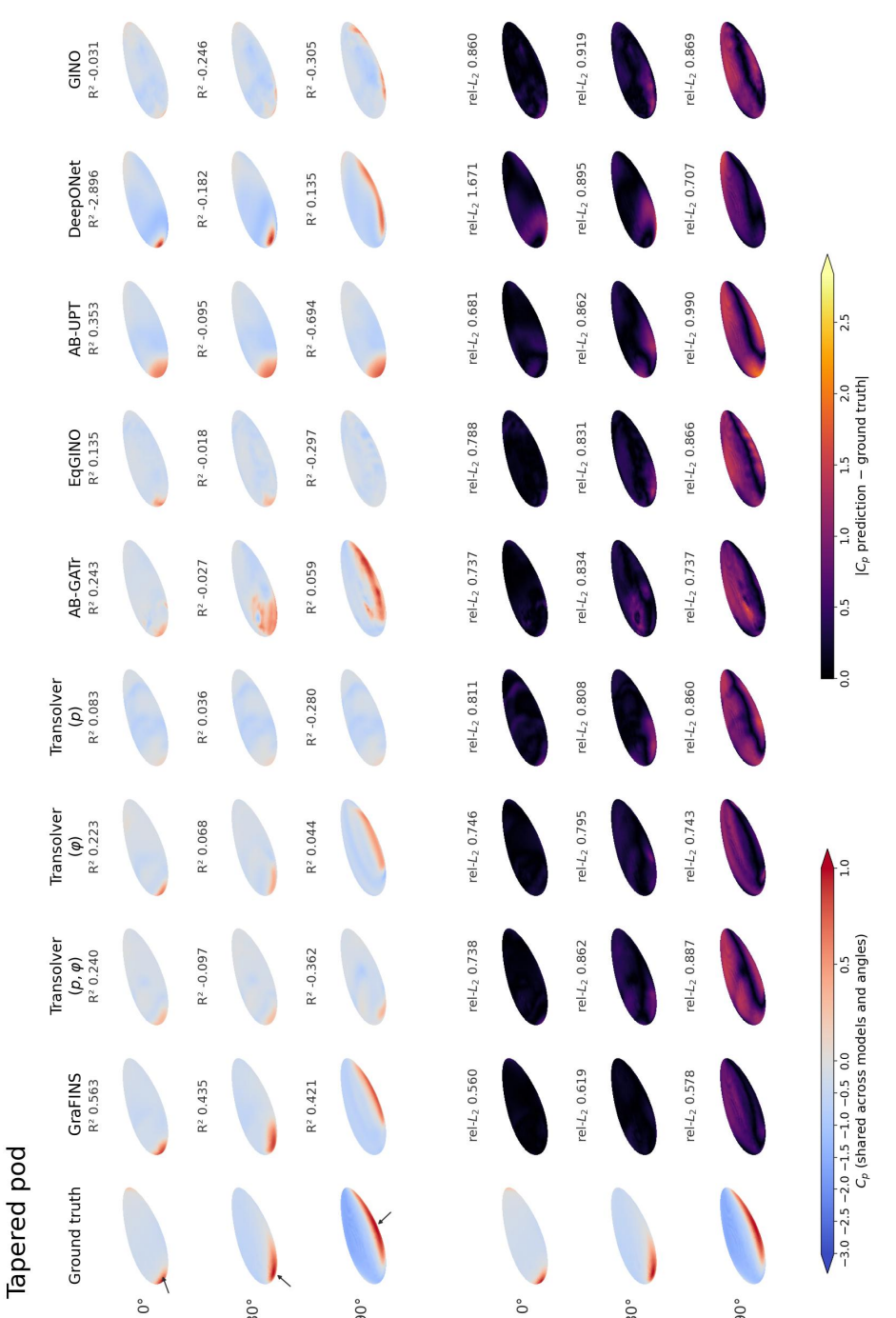


Figure 29: **Arbitrary geometries, tapered pod.** Conventions follow Fig. 17.

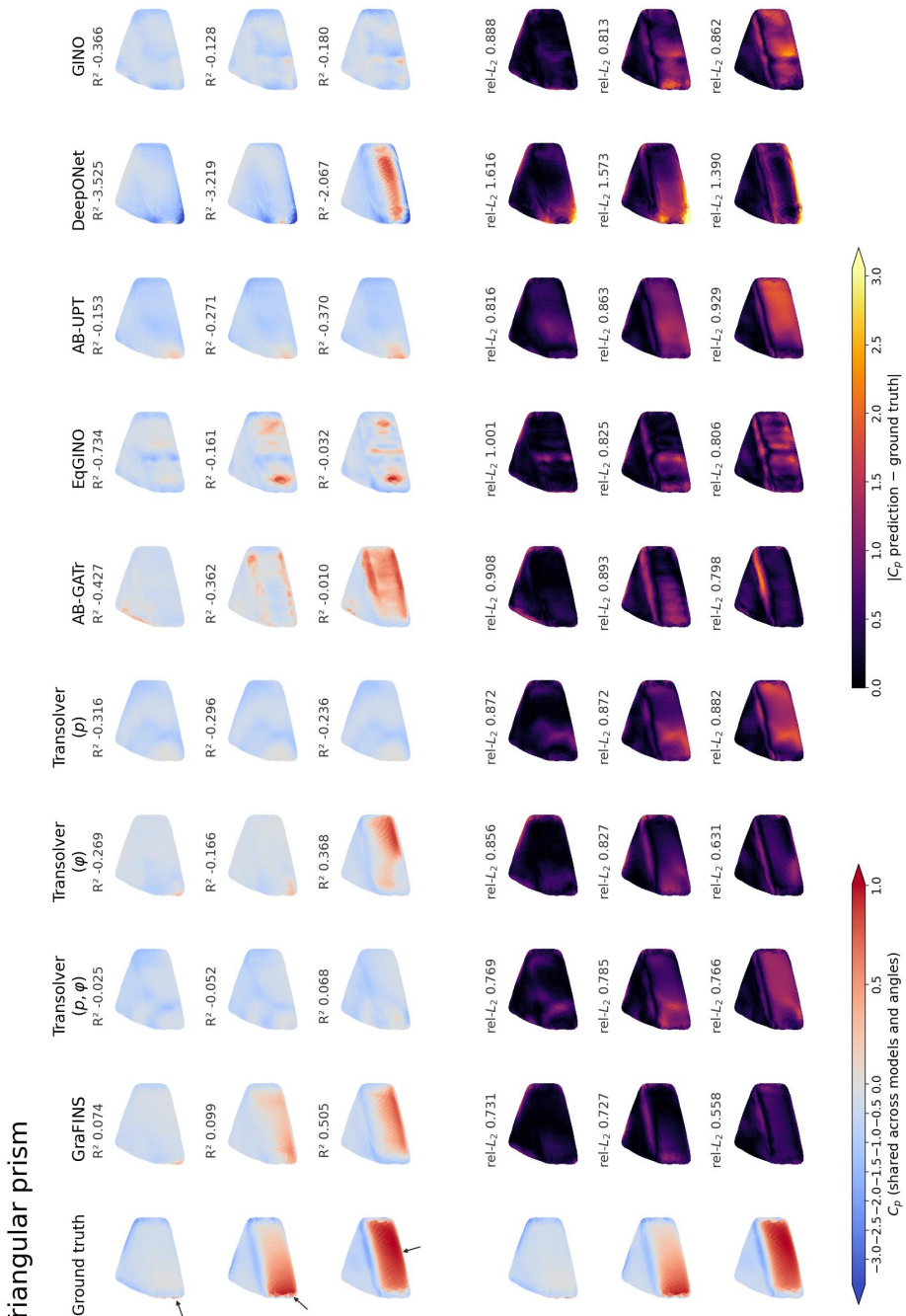


Figure 30: **Arbitrary geometries, triangular prism.** Conventions follow Fig. 17.

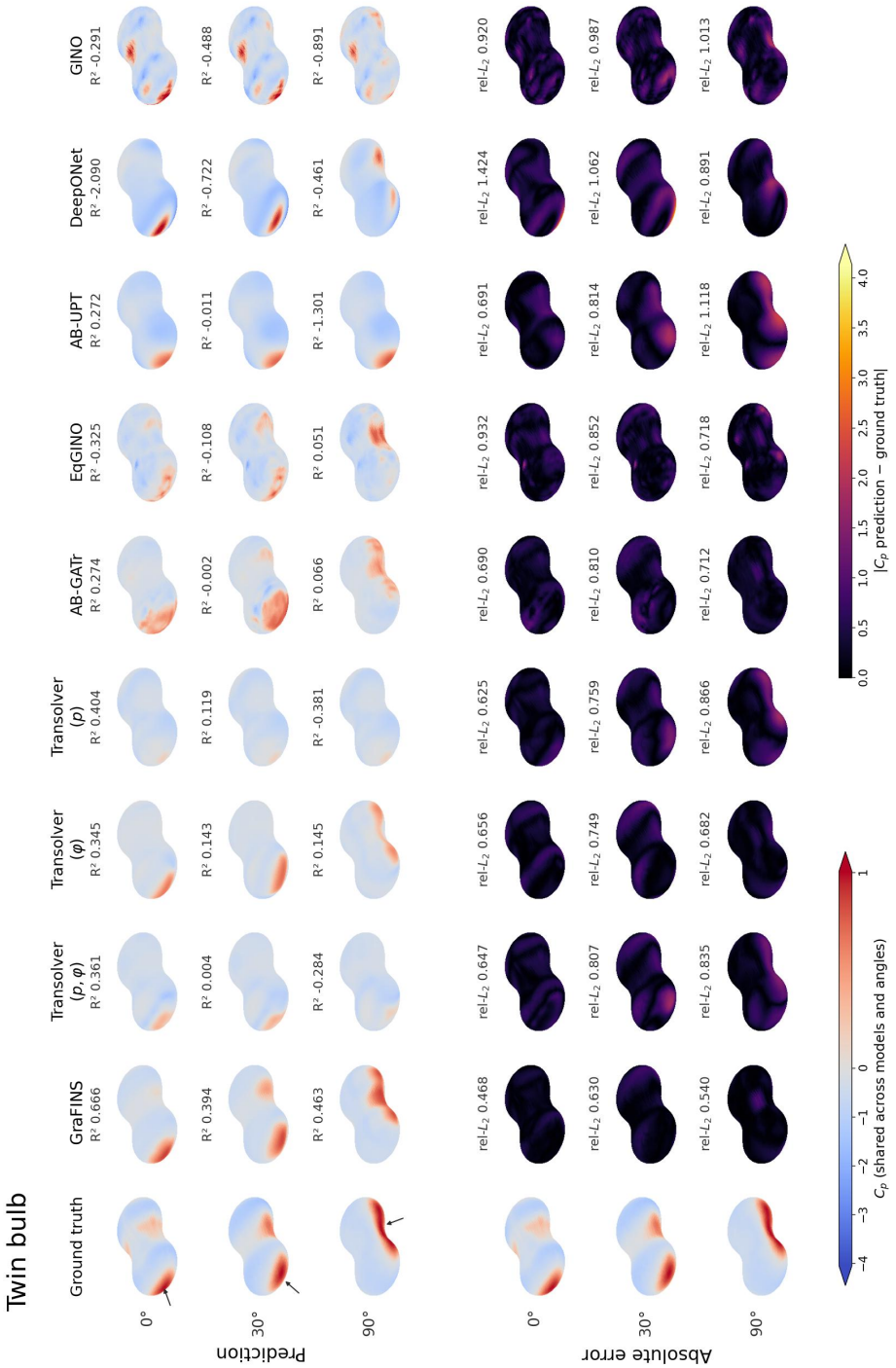


Figure 31: **Arbitrary geometries, twin bulb.** Conventions follow Fig. 17.

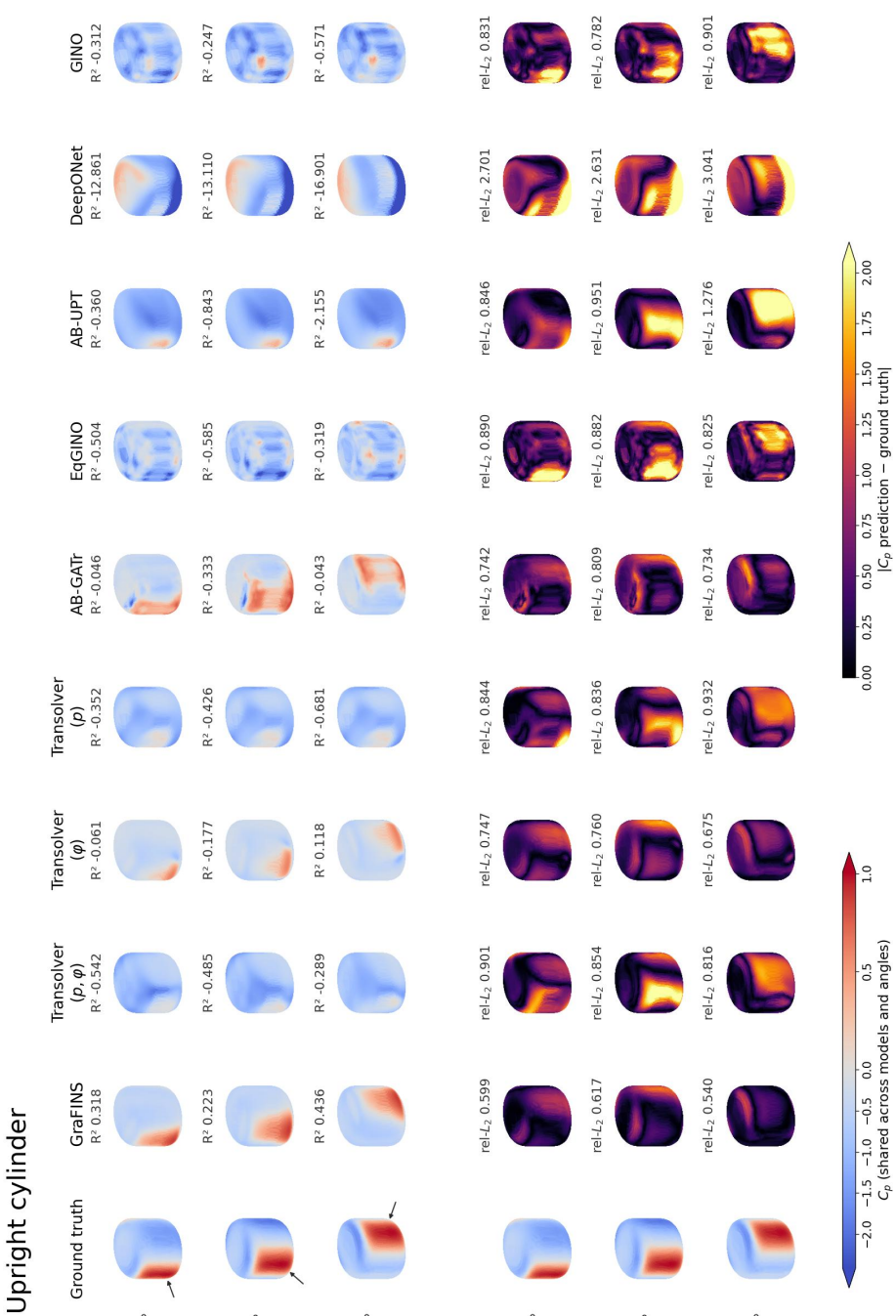


Figure 32: **Arbitrary geometries, upright cylinder.** Conventions follow Fig. 17.